University of Namur

THESIS / THÈSE

MASTER IN COMPUTER SCIENCE

Generic PDA: based museum guide with sound enhancement

Ramdoyal, Ravi

Award date:
2003

Facultés Universitaires Notre-Dame de la Paix, Namur
Institut d'Informatique
Année Académique 2002-2003

Generic PDA Based Museum Guide With Sound Enhancement

Ravi R.C. Ramdoyal

Mémoire présenté en vue de l'obtention
du grade de Maître en Informatique

Abstract

This dissertation focuses on the possibilities of implementing a museum guide with sound features on a PDA. Several research projects are being led in this field, and more generally in the mobile guiding domain. PDAs do suffer limitations, especially when it comes to user interaction and display of information. That's why bringing sound enhancement can improve the usability and user-friendliness of PDA-based applications.

We will give a rough idea of the kind of generic museum guiding application that we would like to develop for PDA, then we will analyse which kind of sound pattern could suit it best. We will then examine the existing programming languages available, before refining the definition and features of our application. After exposing a few implementation problems and issues, we will try to rate our application by presenting an evaluation questionnaire, and we shall finish by considering possible improvements and opportunities for our application.

Keywords : PDA, Sound, Museum Guide.

Résumé

Ce mémoire se concentre sur les possibilités d'implémentation d'un guide de musée avec fonctionnalités sonores sur PDA. Il existe déjà divers projets de recherche dans ce domaine, et de manière plus générale dans le guidage mobile. Les PDAs souffrent de limitations, en particulier au niveau de l'interaction avec l'utilisateur et de l'affichage d'informations. Un enrichissement sonore des interfaces pourrait dès lors améliorer l'utilisabilité et l'intérêt des applications sur PDA.

Nous donnerons une idée globale du type de guide générique de musée que nous souhaitons implémenter sur PDA, puis nous analyserons le type de modèle sonore le plus approprié. Nous examinerons les différents langages de programmation disponibles avant de raffiner la définition et les fonctionnalités de notre application. Après avoir exposé quelques problèmes et astuces d'implémentation, nous essayerons d'évaluer notre application sur base du questionnaire que nous présenterons. Et pour terminer, nous envisagerons des améliorations et des perspectives pour notre application.

Mots-clés : PDA, Son, Guide de Musée.

Acknowledgement

This Master Degree work demanded a fair amount of personal investment, but it couldn't have been achieved without the help of several persons. That's why, first of all, I would like to thank my Professor, Mrs Noirhomme, for her support and guidance during the whole process of my work.

I would also like to thank my supervisor in Pisa, Dr Fabio Paternò, for all his help during my stay in Italy. Although he was very busy, especially with the organization of the HCI Mobility 2002, he managed to find some time to orientate and follow my research.

My gratitude also goes to my landlady and all the HCI Group from ISTI-CNR in Pisa, for their kindness and support during my stay : Signora Siclari, Barbara, Carmen, Carmine, Franca, Francesco, Giulio, Luisa, Silvia, Vincenzo and Willy. Their presence really meant a lot to me.

I would also like to thank all the persons who answered the evaluation questionnaire of my application. It required quite some time and energy, but every answer was truly constructive and critical for my work. Thanks again.

Besides, I would like to say a special thanks to Giz for his mathematical support, and to Etienne Cuvelier, who helped me analyse the results of my investigation questionnaire.

And last but not least, I would like to thank all my family and my friends for their unconditional support. Never underestimate the boons of the ones you cherish.

Content

Content i

List of Tables v

List of Figures vi

1 Introduction 1

2 State Of Art Review 3

2.1 The Actual Situation in Museums 3
2.1.1 Traditionnal Visiting Behaviors 3
2.1.2 Considering Users Expectations 3
2.2 The Existing Solutions and Research Projects 4
2.2.1 The Louvre Museum Audio Guides 4
2.2.2 ILEX 4
2.2.3 Cyberguide 5
2.2.4 The HIPS Project 6
2.2.5 GUIDE 7
2.2.6 An Unexpensive And Easy-to-install Museum Guide 7
2.2.7 LoL@ 8
2.2.8 The Cooltown Museum 9
2.2.9 Rememberer 9
2.2.10 The Musical Instruments Museum 11
2.3 Conclusion 11

3 A First Approach For Using Sound In PDA Interfaces 12

3.1 The Limitations Of The PDAs 12
3.2 The Various Enhancements Provided By Sound 13
3.2.1 Why Should We Use Sound? 13
3.2.2 How Can We Use Sound In General? 14
3.2.3 How Can We Use Sound and Speech In Mobile Museum Applications? 15
3.3 The Use Of Vocal Input 16
3.4 Conclusion 17

4 Exposing The Main Objectives Of A Generic PDA Based Museum Guide 18

4.1 Context 18
4.2 Scenarios 18
4.2.1 The Wandering Visitor 18
4.2.2 Specific Art Domain Visitor 19
4.2.3 Specific Object Visitor 19
4.3 Task Analysis 19
4.3.1 Orientation Within The Museum 20
4.3.2 Access To Museum Information 20
4.3.3 Control Of The User Interface 20
4.3.4 Control Of The Tour Mode 20
4.4 Use Of Sound In The Interface 20
4.5 Task Models 20
4.5.1 The ConcurTaskTree Notation 20
4.5.2 The ConcurTaskTree For Our Application 23
4.6 Conclusion 23

5 **Defining the appropriate sound pattern** **25**

5.1 Possibilities 25
5.1.1 Auditoy Icons 25
5.1.2 Earcons 26
5.1.3 Comparing Auditory icons ans Earcons 27
5.1.4 Conclusion : choosing between auditory icons and icons 28
5.2 Generic Sonification guidelines 28
5.2.1 Warning guidelines 28
5.2.2 Spatial Sonication Guidelines 28
5.2.3 Prefering Auditory Displays to Visual Displays Guidelines 29
5.3 Auditory Icons Guidelines 29
5.3.1 Issues 29
5.3.2 Mapping Sounds to Events 29
5.3.3 The Vocabulary of Auditory Icons 31
5.3.4 Annoyance 33
5.4 Earcons guidelines 34
5.4.1 Where to begin 35
5.4.2 Timbre 35
5.4.3 Register 35
5.4.4 Pitch 35
5.4.5 Rhythm, tempo and duration 35
5.4.6 Intensity 36
5.4.7 Spatial location 36
5.4.8 Making earcons attention-grabbing 36
5.4.9 Combinations 36
5.5 The sound pattern for our application 36
5.5.1 Interacting with common dialogs 36
5.5.2 Interacting with the maps with auditory icons 37
5.5.3 Interacting with the maps with earcons 37
5.5.4 Localisation Issues 38
5.6 Conclusion 38

6 **Choosing The Appropriate Programming Language.** **39**

6.1 Introduction 39
6.2 Java for PDAs. 39
6.2.1 PersonalJava™ 39
6.2.2 CLDC and MIDP, for low level PDAs 39
6.2.3 CDC and Foundation Profile, for High Level PDAs 40
6.2.4 Virtual Machines for Java 41
6.2.5 Java Speech Solutions. 42
6.3 C++ for PDAs. 43
6.3.1 eMbedded Visual C++ 43
6.3.2 Sound and Speech Libraries 44
6.4 Conclusion 44

7 **Refining And Planning The Implementation Of Our Generic PDA Based Museum Guide** **45**

7.1 Introduction 45
7.2 Requirements 45
7.2.1 Features and Limitations Of The Application 45
7.2.2 Class Diagram 45
7.2.3 Non functionnal requirements 46
7.3 Logical Conception 46
7.3.1 Abstract Interface 46
7.3.2 Data Storage 48
7.3.3 Logical Components Definition 48

7.4 Physical Conception 48
7.5 Conclusion 50

8 Implementation And Usability Concerns 51

8.1 Issues 51
8.2 Customizing The Environment 51
8.2.1 Format of the specification file 51
8.2.2 Using Customized Images And Sounds 52
8.2.3 Determining The Coordinates Of The Environment 52
8.2.4 Example : the Room 19 specification 53
8.3 Determining The Target Of The Stylus Input 55
8.3.1 The Problem 55
8.3.2 A Few Definitions 55
8.3.3 Establishing the algorithm 55
8.3.4 The algorithm itself 56
8.4 Limitations 57
8.4.1 The Connections Between Rooms 57
8.4.2 The Art Objects 58
8.4.3 The "Inner Rooms" 58
8.5 Conclusion 60

9 Rating the interface 61

9.1 Introduction 61
9.2 Presentation Of The Interface 61
9.3 Adapting Our Application For The Evaluation 63
9.4 Presentation Of The Questionnaire 64
9.4.1 Personal Information 64
9.4.2 Previous Experience in Sound and Mobility 64
9.4.3 Scenarios 64
9.4.4 Interface General Rating 65
9.4.5 Interface Sound Rating 65
9.4.6 General Comments 65
9.5 Results and Analysis 65
9.5.1 The subjects 65
9.5.2 The Tool and the Methods Used 66
9.5.3 The Impact of the Personnal Background on the Apprehension of the Prototype 67
9.5.4 The Impact of the Personnal Background on the Sound Appreciation 70
9.5.5 More General Issues of Fact 72
9.6 Possible Improvements And Opportunities For The Application 73
9.6.1 Immediate Improvements 73
9.6.2 Further Improvements 74
9.6.3 Opportunities 75
9.7 Conclusion 75

10 Conclusion 76

11 Bibliography 78

Appendix A : Detailed ConcurTask Trees 85

A.1 The First ConcurTaskTree Approach For Our Application 86
A.2 The Refined ConcurTaskTree Of Our Application 90

Appendix B : Environment Specification Examples 92

B.1 Specification File Of A Small And Simple Environment 93
B.2 Specification File Of The Fourth Floor Of The Computer Science Institute 97

Appendix C : Questionnaire Answers 119

Appendix D : Questionnaire Analysis **149**
D.1 Critical Values Table For the t Distribution of Student 150
D.2 Analysis of the Apprehension of the Prototype 151
D.2.1 Scenario 1 151
D.2.2 Scenario 2 154
D.2.3 Scenario 3 160
D.3 Analysis of the Sound Appreciation 169
D.3.1 Auditory Icons 169
D.3.3 Sound Utility 171

Appendix E : Source Code **172**
E.1 ravi-guide.h 173
E.2 ravi-guide.cpp 177
E.3 ravi-guide.rc 225
E.4 multimedia.h 228
E.5 multimedia.cpp 229
E.6 init.h 233
E.7 init.cpp 234
E.8 attributelist.h 263
E.9 attributelist.cpp 264
E.10 attribute.h 268
E.11 attribute.cpp 269
E.12 roomlist.h 270
E.13 roomlist.cpp 271
E.14 room.h 276
E.15 room.cpp 278
E.16 artobject.h 285
E.17 artobject.cpp 286

List of Tables

Table 4-1 - The ConcurTaskTree Task Classification 21
Table 4-2 - The ConcurTaskTree Temporal operators 22
Table 6-1 - Java Virtual Machines For PDAs 41
Table 8-1 - Format of the specification file 52
Table 8-2 - Excerpt of the definitions of the Art Object class header 52
Table 8-3 - The Room 19 specification example 54
Table 8-4 - The algorithm used to determine a given point is inside a polygon or not 56
Table 8-5 - The Polygon and the Rectangle methods in eMbedded C++ 57
Table 8-6 - Excerpt of the Room class header 57
Table 8-7 - Definition of the maximum of connections allowed in the ArtObject class header 58
Table 9-1 - Personal Information 64
Table 9-2 - Previous Experience in Sound and Mobile Devices 64
Table 9-3 - Scenarios 65
Table 9-4 - Interface Overall Evaluation 65
Table 9-5 - Sound Pattern Evaluation 65
Table 9-6 - Ordinal Values for The Questionnaire Analysis 66
Table 9-7- Influence of the Musical Formation on the Earcon Appreciation 71
Table 9-8 Sound Pattern Appreciation Summary 72
Table 9-9 - Excerpt of the Specification File for the Prototype Evaluation 73

List of Figures

Figure 2-1 - Screenshot of the Cyberguide Indoor Prototype....5
Figure 2-2 - The Cyberguide Outdoor Prototype, with its GPS unit....6
Figure 2-3 - The GUIDE Interface....7
Figure 2-4 - Modality of navigation of the Carrara Marble Museum Guide....8
Figure 2-5 - The LoL@ Interface....9
Figure 2-6 - The Rememberer toolkit....10
Figure 2-7 - The Rememberer webpage....10
Figure 3-1 - A few PDAs and PalmTops....12
Figure 4-1 - Hotel Reservation ConcurTaskTree....22
Figure 4-2 - Museum Information Access ConcurTaskTree....22
Figure 4-3 - Flight Reservation ConcurTaskTree....23
Figure 4-4 - The First ConcurTaskTree Approach For Our Application....24
Figure 7-1 - The Class Diagram For Our Application....46
Figure 7-2 - The Refined ConcurTaskTree Of Our Application....47
Figure 7-3 – Logical Components Definition For Our Application....48
Figure 7-4 – Class / File Hierarchy of Our Application....49
Figure 8-1 - Absolute origin of an environment....53
Figure 8-2 - Circular Enumeration of the Coordinates of a Room....53
Figure 8-3 - The Room 19....54
Figure 8-4 - Intersections between a convex polygon and a semi-strait line....55
Figure 8-5 - Intersections between a non convex polygon and a semi-strait line....56
Figure 8-6 – Polygon configurations for which the rules fails....56
Figure 8-7 - The Connection Problem with the Rectangle Method....57
Figure 8-8 – Representation of an Art Object in the Room Map....58
Figure 8-9 - The Inner Room Problem....58
Figure 8-10 - Specification limits for the inner room problem....59
Figure 8-11 - The imaginary wall alternative....59
Figure 9-1 - Start Tour Screenshots....61
Figure 9-2 - Customize Tour Mode Screenshots....61
Figure 9-3 - Overview and Room Map....62
Figure 9-4 - The Art Object Information Screen....62
Figure 9-5 - The InfraRed Simulation Screenshots....63
Figure 9-6 – The Fourth Floor of the Computer Science Institute....63

1 Introduction

Nowadays, mobile devices are becoming more and more powerful in terms of computational and interactive resources and capacity. Mobile business applications, like online banking and trading, file access, mail, or calendar, which are quite similar to the well known Internet/Intranet applications, are commonly used with Laptops or powerful Personal Digital Assistants (PDAs). Moreover, mobile leisure applications, like mobile multiplayer games, video messaging, shopping and other tools for leisure time use, provide entirely new possibilities to mobile users.

In the meantime, especially at the research level, an increasing interest in exploiting the massive potential of mobile technologies has arisen, to provide dynamic multimedia information to people on the move. This raises interesting issues on how to exploit them to support users in various contexts of use, for instance for guiding assistance.

This dissertation precisely focuses on the possibilities of implementing a generic and re-usable museum guide on a PDA. As mobile devices aren't always so intuitive and easy to use, we will also explore different sonification methods that could enable a better interaction with the users. Our dissertation will be organized as follow.

In Chapter 2, we will take a look at several research projects being led in this field of museum guiding application, and more generally in the mobile guiding domain.

In Chapter 3, we will give a first approach for using sound in PDA interfaces. PDAs do suffer limitations, especially when it comes to user interaction and display of information. We will see how sound enhancement can improve the usability and user-friendliness of PDA-based applications.

In Chapter 4, we will give a rough idea of the kind of generic museum guiding application that we would like to develop for PDA. For this purpose, we will present the ConcurTask Notation that was used for exposing the main objectives of our generic PDA based museum guide.

In Chapter 5, we will try to define the most appropriate sound pattern for our application. As we will see, choosing between Auditory icons and Earcons is not that simple. Guidelines will be given for these two sonification methods, as well as more generic guidelines.

Then, in Chapter 6, we will examine the existing languages available for mobile applications programming. We will especially try to find out what sound solutions exist, and if speech technologies are available.

We will therefore be able to refine the definition and features of our application in Chapter 7. It will enable us to explain the main entities that our application will have to manipulate, and to expose the architecture that we will use.

Afterwards, in Chapter 8, we will expose a few implementation problems and issues. We will explain how the text input file specifying an environment for our application should be formatted, which implementation limitations we had to set and how we had to manage the stylus input.

Then, in Chapter 9, we will try to rate our application by presenting its interface and an evaluation questionnaire. We will present and analyze the answers we collected in order to draw some teachings and envisage possible improvements.

We will therefore be able to conclude our dissertation with Chapter 10. Examples of specification files will be given in Appendix B, the answers to the questionnaire in Appendix C, their analysis in Appendix D, and finally, the source code of our application in Appendix E.

2 State Of Art Review

2.1 The Actual Situation in Museums

Museums play a key part in our actual cultural and social environment. When you visit a city, you will most probably want to admire the masterpieces exposed in the local museums. The reasons can vary from a person to the other : viewing a specific artist's work, understanding art from a particular era, or just wandering around and enjoying some free time. Moreover, visitors have different points of view when it comes to organizing their museum visit.

2.1.1 Traditional Visiting Behaviors

Based on [ActiveArt99] and [HIPS97B], we can point out the traditional guiding supports used by visitors.

First of all, museums usually provide guided tours for groups, supervised by museum-trained employees. The visitors move as a group, visiting the same art piece at the same time, listening to the human guide's explanation about each piece of art, and then moving on, as a group. The visit is rather linear, and depending on the good will of the human guide, the tour and the explanations can be more or less personalized. But in the end, the visit remains by group and thus cannot be tailored to the individual. The discourse of the human guide can disturb other visitors.

It is also possible to use museum-provided cassettes to explore the museum. These tours are similar to the human-guided tours in many ways, except that individuals are able to choose the pace of the tour by controlling the cassette tape device. The degree of personalization seems to be lower, in the sense that you can't modify the content of the cassette, sort the objects of interest, or decide to directly skip part of the speech: the discourse remains sequential.

Other visitors may come with their own agendas and documentation, in order to view a specific subset of the exhibit. These visitors will ask for a map of the museum, and directly go to their point of interest. However, reading interferes with the visit, and depending on the guide book, the information can be limited, or extremely specific.

Finally, some visitors just wander through the museum, viewing art without any particular pattern. A visitor may see an art piece, consider it interesting, search for similar work, view it, and continue exploring. He (or she) must rely on the labels available next to the object he is looking at, which can be troublesome, for instance when the museum is crowded.

2.1.2 Considering Users Expectations

i3 is a European Programme which aims at promoting human centered methodologies for the design of Intelligent Information Interfaces, i.e. tools supporting people in their everyday work or leisure activities [i3-WEB]. During a workshop organized by the consortium in October 1997, a study on the museum visitors' point of view was led [HIPS97B]. The results showed that the visitors expected better delivery systems, i.e. better ways of accessing information in museums.

Indeed, the various labels, panels and signs installed by museums are certainly a simple way of delivering information, but their placement and legibility can be problematic. And as we already mentioned, human guides and guide books aren't always an ideal solution either.

The study also pointed out the deficiencies of the information content. Visitors have a need for better orientation indications (where are located specific items, the toilets, the bar, etc.), up to date and practical information. They would enjoy more explanations on the context and the links between the presented items. And special attention should be given to presenting information to children, in a fashion which is engaging and stimulating.

All these considerations lead us to conclude that museum visitors expect a richer and more accessible information structure, which shouldn't be restraining or difficult to use. Consequently, it would allow them to freely plan and enjoy their tour in the best conditions.

2.2 The Existing Solutions and Research Projects

In order to improve the visitors touring conditions, different solutions have been proposed, and several research projects are being led. We will take an overview of some of these researches, and related works that are relevant for our matter (such as mobile city guides).

2.2.1 The Louvre Museum Audio Guides

Back in 1997, the Louvre Museum provided audio guides to the visitors, as described in [HIPS97A]. These quite simple devices, looking like a large telephone handset, had a speaker, a numeric keypad, a small LCD display, and 4 buttons for play/pause, stop/cancel, rewind/fast forward and volume control. The handset was handed out together with a leaflet listing the items that had an audio presentation.

The audio guide provided comments for a limited number of items displayed in the museum. To hear an audio commentary, the user had to enter the 3 digit code related to the object on the keypad, then press play. Hence the audio guide enabled the visitors to browse to the museum as they desire. Indeed, this is a pull oriented system : the users can pull the information when and where they want. We can note that an example of a push oriented system would be the tape tours. We can view the tv/web comparison as an illustration of the push/pull opposition

But as we can see, the audio guide truly consisted of 2 parts, which were the handset and the leaflet. It is obvious that this didn't increase the handiness of the guide. Moreover, the content wasn't flexible or customizable, and it was limited by the small storage features of the device.

This former audioguide has now been replaced by an individual CD ROM player which includes commentaries on 180 works and major locations within the museum, without the need to follow a specific route. It is available in 6 languages (French, English, German, Spanish, Italian, Japanese), and can be obtained from the entrances of the museum [Louvre-WEB].

2.2.2 ILEX

Rather then providing static information, it could be interesting to adapt it in an appropriate way. The Intelligent Labeling Explorer (ILEX) is precisely a system that dynamically generates text labels for exhibits in museums, such as the 20th Century Jewellery Exhibit in the Royal Museum of Scotland [ILEX99]. The description of the objects encountered during the tour are generated to reflect the interests of the user and to take in account the discourse history. That means that the description of an object being currently viewed will introduce comparisons and contrasts to the previously viewed objects, and will omit all background information already delivered.

ILEX is an hypermedia system, that is an hypertext system including graphics, diagrams, photographs, movies, animations, etc. Hence it uses a browser based interface which allows a non linear access to the various multimedia resources. The information generation process has four stages, which are content selection, content structuration, sentence realization and finally, text presentation. The aim is to reproduce the kind of descriptions that a real curator might give to a visitor [ILEX-WEB]. Unfortunately, ILEX is not implemented on mobile devices.

2.2.3 Cyberguide

Cyberguide is a location-aware tourist guide that provides information to tourists based on knowledge of position and orientation [Cyberguide96]. What should retain our attention is that the system is designed as a group of interacting service components, in order to reflect the diverse needs of the user. Hence there are a mapping, an information, a positioning and a communication component, which enables an extensive and modular approach to system development.

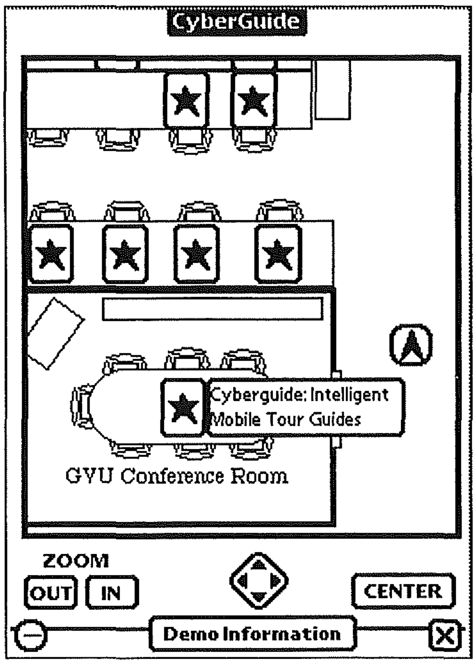


Figure 2-1 - Screenshot of the Cyberguide Indoor Prototype

Cyberguide has been implemented on the Apple MessagePad in two versions. In the "indoor" version, the map module contains the map of the place to visit with scrolling and zooming features, while the information module contains information about the objects visible on the map. The communication module, based on a wired Internet Appletalk Gateway, allows facilities such as email. Finally, the positioning module is based on infrared technology (IR). The "outdoor" guide required adapting the mapping and the positioning module. The map was modified to represent the wider environment and was just replaced in the first module, while the positioning system was replaced by GPS.

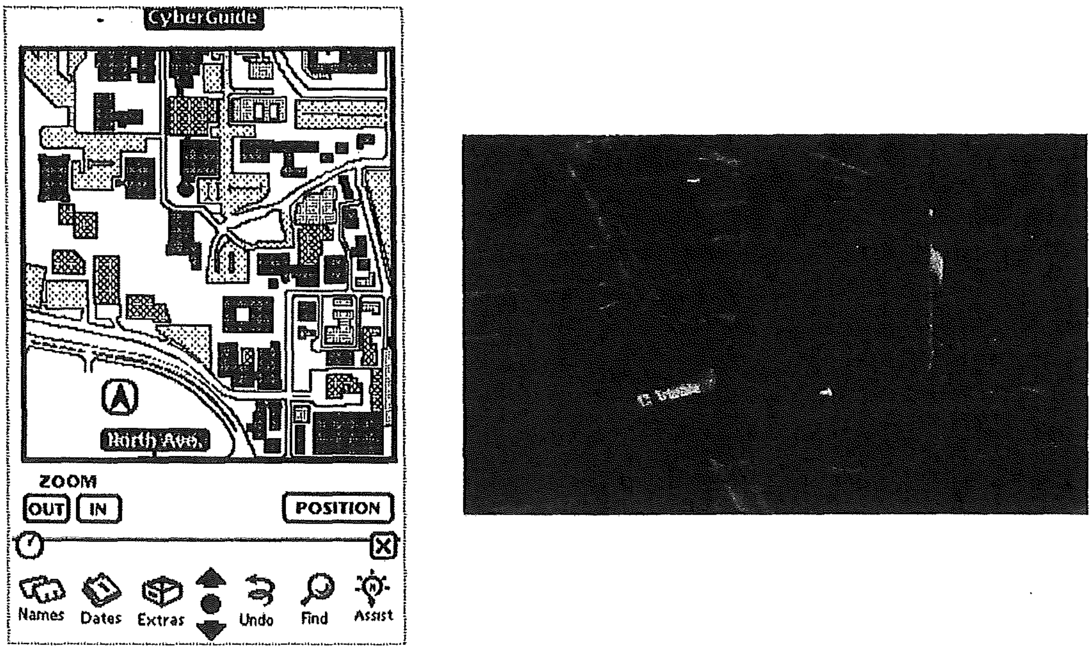


Figure 2-2 - The Cyberguide Outdoor Prototype, with its GPS unit

After evaluating the system, it appeared that it is possible to create cost-effective context-aware applications with equipment that is readily available. Secondly, it seems that absolute positioning information throughout an entire space is not so important: knowing what someone is looking at is more useful than knowing the person's exact physical position and orientation. Finally, it is better to separate the positioning system from the communications system, because positioning information doesn't need to be uniformly distributed, while communication services need global coverage of the physical space. Which is even more relevant when passing from an indoor to an outdoor environment [Cyberguide95].

2.2.4 The HIPS Project

The Hyper-Interaction within Physical Space (HIPS) project is funded by the European Commission's i3 consortium [HIPS97A]. It aims at developing a hand-held electronic tour guide allowing tourists to navigate both the physical space and a related information space when visiting a museum or a city.

One of the interesting particularities of HIPS is its location awareness feature: the system is able to detect the location of the user and to personalize the contextual information. It is based on a client-server model, where the clients (hand-held computers) are wirelessly linked to the server. The positioning is performed by using various means such as infrared, radio, GPS.

HIPS is a push and pull system. Indeed, the users pull information while standing in front of an exhibit and walking around, while information can be pushed by guiding the user along predetermined routes; like in most of the location aware systems, the chosen information is selected depending on the user position and the length of stay in that position, which can be considered as manifestation of the user's interests. But this assumption isn't necessarily well-founded, as we can see in [Ciavarella&Paterno]. Nevertheless, it still encounters to reducing the gap between the physical space and the related information space [HIPS-WEB].

2.2.5 GUIDE

GUIDE is a closely related but more recent project, which consists in an intelligent electronic tourist guide for Lancaster, developed by the Distributed Multimedia Research Group of Lancaster University [GUIDE2000]. The system combines mobile computing technologies with a wireless infrastructure to present personalized information to the visitors. It also uses context-awareness and adaptive hypermedia to support the information and navigation needs of the visitors.

The user interface is based around a modified browser metaphor. It enables the user to perform tasks such as retrieving information, navigating in the city using a map, creating and following a tour of the city, communicating with other visitors (via text messages), or even booking accommodations.

The analysis of GUIDE showed that interaction with a context/location-aware system is not affected only by the design of the user interface, but also by the design of the infrastructure, i.e. the strategic placement of cells in order to ensure location resolution and network connectivity. It also showed how difficult it can be to evaluate the information requirements of users based on current context. For example, restricting the information available to visitors, in such a way that they can only access information on the attractions at their current location, can cause frustration and disappointment : users may want to access information relative to an object they can see, but that isn't close enough according to the system.

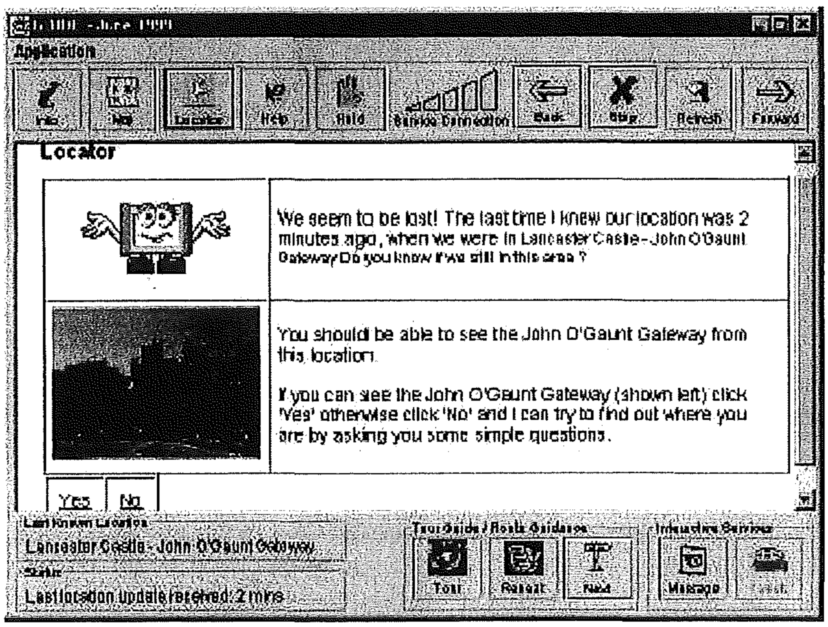


Figure 2-3 - The GUIDE Interface

2.2.6 An Unexpensive And Easy-to-install Museum Guide

CNR-ISTI's HCI Group designed an application aiming to overcome limitations such as technical difficulties and expenses [Ciavarella&Paterno]. Indeed, systems based on automatic generation of location-aware information suffer from the limitations of adaptive systems, where adapting the user interface can, as a matter of fact, cause disorientation. In addition, the location-aware technology can still be quite expensive, inefficient and difficult to install in wide environments. Infrared, for instance, suffer short range communication possibilities, and therefore need to be installed for each work. The emitters and the receivers must be lined-up in order to communicate.

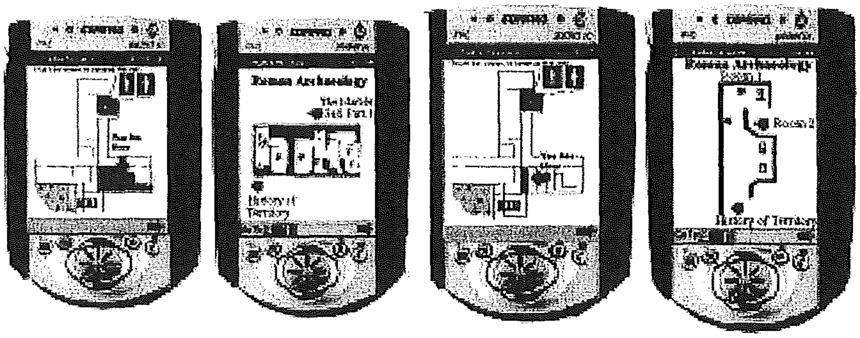


Figure 2-4 - Modality of navigation of the Carrara Marble Museum Guide

The study was led with the Carrara Marble Museum. This museum is precisely a museum with hundreds of works of art and limited budget. The proposed solution is based on a multimedia PDA whose main purpose is to support a user which can move freely in a museum. Location-awareness isn't available, but the interface is structured as to allow visitors to easily orient themselves and to provide adapted and relevant information.

The system enables orientation within the museum, with three levels of spatial information : a museum map, a section map, and, for each physical environment composing the section, a map with icons indicating the main pieces of work available in the room and their location. The user can select the icons to access basic information and corresponding audio description. The user can control the interface and media, e.g. he can change the volume of the audio comments, stop and start them, and move through the various levels of detail of the museum description. Various museum information is provided at different levels : museum, section, physical environment and single work.

2.2.7 LoL@

The Local Location Assistant (LoL@) is a prototype of a mobile application for GPRS/UMTS designed by the Forschungszentrum Telekommunikation Wien (ftw) [LoL@2002]. It implements a tourist guide for the users in the City of Vienna on PDA and cell phone, and a laptop is currently used as terminal to test the application.

LoL@ provides a map of the inner city of Vienna in two zoom levels (overview and detail), plus textual and multimedia information screens. It can be used from a static location, for example to plan a sight seeing tour. It can also be used while walking in the city : the user is aware of his position in the city, and can ask for various information on points of interest and how to reach them. After completing a tour, the user can access a tour diary that is made available on the provider's server. The detail level of the map is still somehow low, so the location isn't always very precise : the user has to select or confirm his/her location in some cases, which makes LoL@'s routing an hybrid concept, combining automatic user positioning with GPS and user interaction.

Figure 2-5 - The LoL@ Interface

As the project is very recent, the field trials haven't been performed yet, but we can already see that map presentation on smart phones with small displays is feasible. Even routing systems can be implemented on such devices. Considering the varying positioning accuracy, the interactive routing concept will give the best user satisfaction.

2.2.8 The Cooltown Museum

The Cooltown Museum is part of HP's wide Cooltown project, as we can see in [Cooltown-WEB] & [Cooltown2000]. The idea is to offer visitors a Web-enhanced experience : as they tour the museum, their PDA receives Web URL from wireless beacons [Cooltown2002]. Using their PDA's Web browser, the visitors can therefore access various multimedia resources related to the artwork in the museum. What is truly ingenious is that the system isn't self-centered : it is possible to gather information from on-line resources available elsewhere.

This Cooltown technology is aiming to a widely spread development in our daily working and private life. For instance, it can be used in conference rooms to control or find information about electronic devices available in the room, such as projectors, printers, etc. It can even be used to save lives by providing a vital link to emergency medical services, using smart, connected appliances [Cooltown-WEB]. Although this work is still at an experimental level, it truly shows all the potential behind mobile technologies.

2.2.9 Rememberer

Rememberer isn't a guiding and information system like the previously exposed research projects. It is a tool designed for capturing personal experiences during a museum visit, so that the user can "remember" them during or after the visit [Rememberer]. The goal is to imply the visitor in the tour, and make him a real actor of the visit.

Rememberer consists of a "remember-this" technology, with which the user selects exhibits during his visit : RFIDS tags, mounted on watches or credit-card shaped artifact, as well as PDAs wirelessly connected to a 802.11 network, using Cooltown infrared beacons [Cooltown2002]. Secondly, the record consists of Web Pages about the visited exhibits, including real-time photographs and typed notes. Finally, a physical artifact (e.g. a fridge magnet) reminds the user of the visit and contains the URL of the visit record, so that he can access his souvenirs and share them with his relatives.

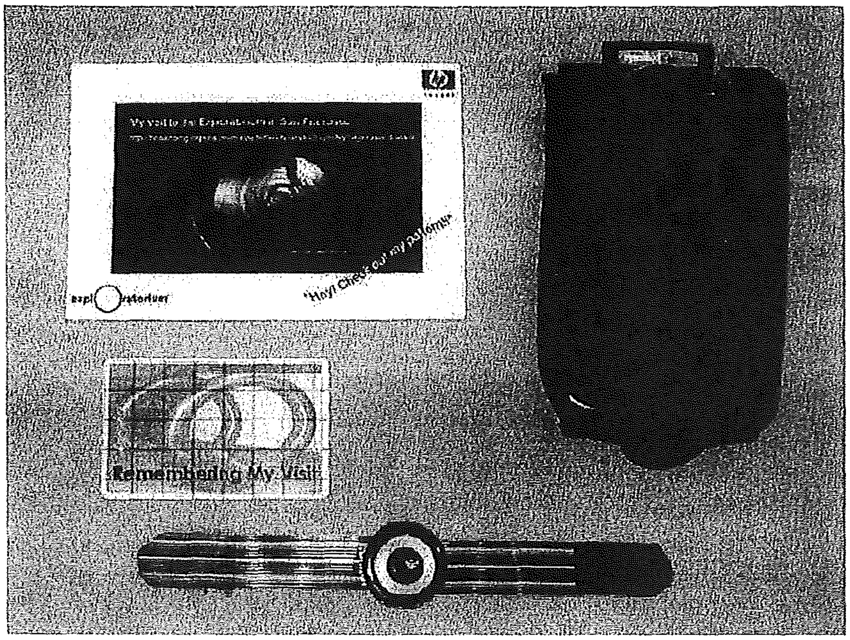


Figure 2-6 - The Rememberer toolkit

A 'reminder' fridge magnet (top left) and 'remember-this' technologies: an RFID card and 'wristwatch', and a PDA in a case that receives and invokes beaconed URLs.

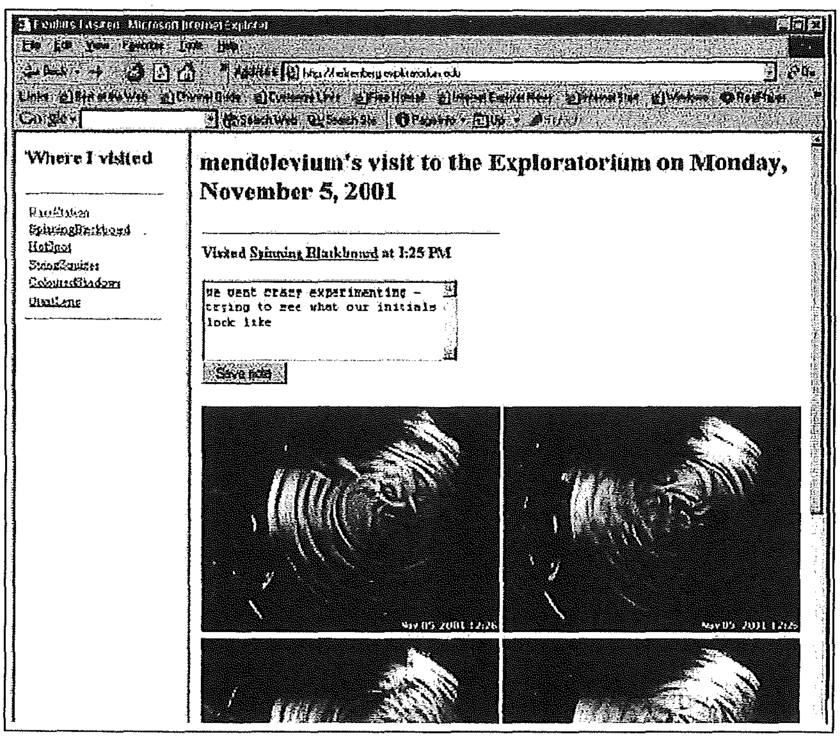


Figure 2-7 - The Rememberer webpage

A Rememberer page showing a user's visit to the "Spinning Blackboard" exhibit.

2.2.10 The Musical Instruments Museum

The Musical Instruments Museum [MIM] of Brussels offers another kind of interactive experience. Some of the instruments that are exposed have a little headphone sign near them. This informs the visitor that an audio track corresponding to the instrument can be heard, if equipped. Indeed, infrared headphones are given to the visitors at the entrance of the museum, which allows them to have a richer experience by listening to the musical context of the instruments exposed in the various exhibits. All the audio elements are musical melodies, there is no speech.

What is very exciting in this kind of approach is that it somehow involves more intimately the visitor in his tour. The counterpart is that the use of the infrared system can be problematic : once a good positioning is found, the visitor has a limited scope of movement. As a matter of fact, the infrared connection can easily be lost and the audio track will hence be terminated at once. Another problem is that it is sometimes difficult to find out to which element the audio track relates to. Indeed, identifying the correct glass-case of instruments can be difficult, but moreover, given a specified one, there is only one melody for all the instruments, so one cannot tell what instrument it relates to.

2.3 Conclusion

As we can see, the research in the field of human guiding assistance is really flourishing. Research groups are taking advantage of the different hand-held supports (PDAs, smart phones, etc.) and communication technologies available (infrared, GPS, etc.) to provide the best mobile information infrastructure and guidance assistance possible. Location and context awareness improvements help tailoring the applications to the needs and desires of the users, which are expressing a certain interest in taking advantage of these new possibilities.

Unfortunately, most of all these technologies remain quite expensive and difficult to install for a widely spread use. And as we will see, the communication technologies and mobile devices are still limited by various factors. The future studies and researches will therefore have to focus on improving the interactions between the users and their mobile device in order to compensate these limitations. One of the interesting ideas is to use sound as a *de facto* part of the applications interface, and this is what we will discuss in the following part of our work.

3 A First Approach For Using Sound In PDA Interfaces

3.1 The Limitations Of The PDAs

Even though the computational and interactive resources of small mobile devices are just getting better and better, PDA do still suffer technical constraints, as shown in [LoL@] or [MicrosoftPDA].

The processors aren't really a source of dissatisfaction. Indeed, they are still way far from their desktop cousins, but they have no reason to be ashamed of their performances. For instance, the iPAQ Pocket Series go from 206 MHz Intel® StrongARM 32-bit RISC Processors to 400 MHz Intel® PXA250 Application Processors [HP]. Combined with the typically 2MB to 64MB of built-in memory, it is sufficient for basic applications, such as address books, calendars, notepads, and other small programs. But it is true that more memory can be necessary to store large files like digital photos or audio recordings, as well as to run large software programs.

Concerning the display, "old" PDA screens are still monochrome or gray-scaled, like some of the current Palmtops [Palm], but the more recent ones can display from 256 to 64,000 colors. Screen size varies among PDAs, but undoubtedly a larger screen enables more readable space. An average screen could correspond to a 320*240 pixels display.

PDA screen displays are either active matrix or passive matrix. Active matrix displays generally are easier to see, more responsive, faster, and can be viewed at larger angles than passive matrix displays, although passive matrix displays will generally use less battery power.

Regarding the interaction with the device, you often have to use a pen-like stylus to enter information into your handheld computer, and to substitute for a mouse for navigating through on-screen menus and controls. Depending on the PDA, you can tap on an on-screen keyboard or enter data by writing on the screen, which requires learning a new way of printing the alphabet. Some PDAs have built-in keyboards, but it is usually an optional peripheric.

Finally, the data transfer rate in mobile networks is still low. GSM allows switched data connection at 9.6 kbits/s, while GPRS goes up to 57.6 kbits/s. UMTS claims 384 kbits/s, but typically it goes from 64 kbits/s to 144 kbits/s. Moreover, network interworking functions and the mobile radio channel causes varying transmission bandwidth and delays.

Consequently, user interaction is still restrained by the PDAs technical constraints. In particular, given the size of the screen, it is not possible to show all the relevant information simultaneously and to allow an easy stylus-input interaction.

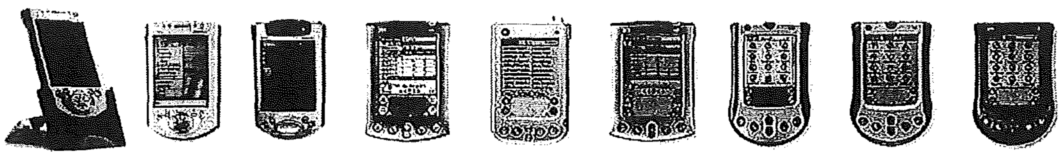


Figure 3-1 - A few PDAs and PalmTops

3.2 The Various Enhancements Provided By Sound

Most of the actual interfaces are graphical and visually very demanding. The information to transmit is more and more voluminous and complex, which implies a higher degree of concentration from the users [Noirhomme2000], and the limitations induced by the size of the PDA screen are permanent. Indeed, a PDA will always have a small screen because of its inherent mobile nature, even if the resolution is increased.

Our concern is that museum guiding applications must precisely be able to provide a lot of information to the visitors, and this in the best delays. Let's see how the use of sound could be an alternative to provide feedback and extra information to the users.

3.2.1 Why Should We Use Sound?

Visual Channel Limitations

We all daily experience the fact that our visual channel is limited, and this, at different levels. First of all, we have, more or less, a 180° vision capability. Per contra, hearing is non-directional [Demoulin&Schöller2001] : we can't see what is behind us, but we can hear it, and we are able lo locate its position quite precisely [Brewster94] and [Brewster&Crease99]. Secondly, when you look at your computer screen for example, even though you perceive loads of details, most of them are blurred by the particular area you are focusing on. The visual channel is indeed very selective. If a detail suddenly changes (a system tray icon starting to blink for instance), you won't necessary notice it. But if a little alarm signal is played, it will more probably catch your attention [Brewster94] and [Brewster&Crease99].

So, as we can see, persons need to use their eyes to perform tasks, or should we say, to perform "one" task at "a" time. But thanks to their hearing, they can do several things simultaneously, because hearing doesn't overburden sight [Noirhomme2000]. In particular, in a museum, they can look at a piece of art, and listen to the comment of the docent at the same time. And if a strange noise suddenly occurs, they will spot it immediately (given that the sound is loud enough of course).

Given its characteristics, adding sound to visual representations offers additional dimensionality : when a complex information is presented simultaneously in a visual and auditory form, it requires significantly less effort to interpret [Demoulin&Schöller2001]. It also offers new means to detect trends and relationships in an information structure. Moreover, short term auditive memory is more efficient than short term visual memory [Noirhomme2000]. Especially with PDAs, sound is therefore an interesting way of receiving information and feedback on various things, while concentrating on a specific task which requires all visual capabilities, or the use of both hands. How could someone truly admire a piece of art if her eyes keep going up and down from the text on her PDA screen to the piece of art itself?

We must clearly distinguish the two categories of sound that can be used in a graphical interface [Noirhomme2000] : speech and non-speech sound. **Speech** is the easiest way of communication between humans. Vocal monitoring can be more efficient than written messages, as the appeal is stronger while the attention can still focus on the other visual sources of information. Multimedia is precisely a domain where such a technique is relevant [Dutoit97]. **Non speech sounds** can be used to represent a concept in a very concise way [Noirhomme 2000]. Several sounds can be heard in parallel, and a complex non speech sound can handle more parameters that speech.

However, human hearing doesn't easily perceive absolute values such as height and intensity, it rather distinguishes changes in the parameters. And when it comes to representing data, non speech sounds are limited by the bounds of auditive perception. The resolution of non speech sounds will depend on the range of the values to represent, which does not apply to speech [Noirhomme2000].

Still, sound is transient, while graphics are persistent. Once it occurred, the sound is gone, while the graphics remain. Another thing to keep in mind is the regulation of the sound volume. It is necessary to monitor the ambient sound level around the device and adapt its output volume appropriately. Reducing the problems of annoyance due to excessive or insufficient volume of sound is a very important step in increasing the acceptability of auditory interfaces [Brewster&al.2000].

Common Problems in Graphical Interfaces

Brewster and Crease pointed out some of the problems when trying to select an item on a menu bar in [Brewster&Crease99]. There can be :

- **mis-selection** : the user chooses a wrong item,
- **item slip** : the user accidentally selects an item while releasing a button,
- **menu slip** : the cursor leaves the menu while the button is being released, which is emphasized when the screen becomes smaller, like for PDAs.

Action slips occur with expert users who perform automatically many simple operations, such as button clicks and menu selections, and do not explicitly monitor the feedback from each interaction. Indeed, as users become familiar with a simple task, they don't feel obliged to monitor the feedback meticulously, especially when they perform a largely automatic task in familiar surroundings.

Closure occurs when a user perceives a task as being completed. In some cases the task may appear to be completed when it is not. The user may experience closure and carry on to do something else and cause an error. As we can see, the graphical feedback can also be insufficient in this case.

Finally, we must mention the problem of the **legibility** of screen text output. If it is possible to adjust font size on desktop computers, it becomes more problematic with PDAs, because of the size of their screen. Moreover, as a PDA is a mobile hand held device, it is not stable, and therefore, reading a text on the screen becomes similar as reading a book while walking : it is possible, but it is not that simple!

3.2.2 How Can We Use Sound In General?

In [NFS97], sonification is defined as the use of non speech audio to convey information. It is a transformation of data relations into perceived relations in an acoustic signal for the purposes of facilitating communication or interpretation. [Demoulin&Schöller2001] took a look at some of the sonification works that have been led.

First of all, we can realize that using sound as part of computers interface isn't that recent : it has been quite some time since our computers took advantage of using **alarm signals**. These signals can reach the user even if the person is occupied with other things. The simplest data sonification can be done with a regular PC speaker, used by the BIOS to announce the state of the hardware acoustically. But alarm signals are rather poor for complete presentation of complex data [Hermann99].

Gaver led significant early researches on the use of non-speech sounds in human-computer interfaces, creating auditory icons to sonify the Macintosh Finder interface in order to produce the SonicFinder [Gaver89]. **Auditory icons** are based around natural, everyday sounds that have an intuitive meaning in the interface. The SonicFinder added sounds for selecting different types of items, dragging and copying, but did not use sound for its own menus. The extra sonic feedback gave users information that they could not see, for example file size and type when selecting. Gaver did not formally experiment his sonic enhancements, but related works show that they are relevant.

[Brewster&al.95A], for instance, successfully investigated the use of sound to improve interaction with graphical buttons. To solve the slip off problems of buttons, they added sounds when the mouse button was pressed down in a graphical button and when a correct selection was made. Analysis of user response to this enhancement showed that the usability of the buttons was increased, and that error recovery was pretty faster.

Other works consisted in sonifying widgets such as scrollbars, like in [Brewster&al.94] and [Beaudouin&Conversy96]. Study of theses implementations also showed improvements in usability. For instance, the time taken to complete tasks, and the overall workload were reduced.

In [Brewster&Crease99], Brewster and Crease enhanced a standard graphical interface by using earcons [Blattner&al.89]. **Earcons** are abstract, musical tones that can be used in structured combinations to create sound messages to represent parts of an interface. Guidelines for creating earcons will be exposed in Chapter 5.

The results have shown that usability of menus could be improved by indicating menu and item slip errors in a salient way. Sound was shown to be a very effective method of providing this feedback. Once again, the overall workload was reduced significantly when using the sonically-enhanced menus, because the sonic enhancements meant that participants needed to expend less effort to notice and recover from menu and item slips.

As for error recovery, users could tell they had made a mistake and correct it significantly faster than in the visual condition. The reason is that users' visual focus move to the next menu before the previous selection is completed. In the visual condition, the avoidable graphical menu feedback is missed, whereas in the auditory condition the demanding sonic feedback ensures alerting the user in case of problem.

3.2.3 How Can We Use Sound and Speech In Mobile Museum Applications?

First of all, speech can be used to give explanations on pieces of art. The visitor can select the specific object on his PDA, then listen to the comment while looking at the object, and not at the screen. As we have seen in the State of Art Review Chapter, some application already took advantage of this possibility. The problem is that usually, they use pre-recorded audio tracks, and therefore, it is not possible to tailor the discourse for the users, in function of what they have already seen for instance.

In this context, the Spoken Output Labelling Explorer Project (SOLE) project is quite interesting : it researches ways to couple natural language generation (NLG) systems with Text-to-Speech (TTS) systems [SOLE]. This should precisely allow to provide dynamic Text-to-Speech and tailor the discourse to the user. A major aspect of SOLE is to couple two existing state-of-the-art systems to form a single speech generation system. The ILEX system, developed jointly by HCRC and AI, is the NLG system operating that we mentioned in Chapter 2. Festival is the speech synthesis system developed at CSTR. The researchers expect that coupling these two systems will have a positive effect on intonation : ILEX would be able to tell Festival when it's comparing or contrasting two objects, when it's referring to old or new information, when it's using a parenthetical or starting a new paragraph, etc. Festival would decide, based on this information, when it needs to pause, to emphasize or deemphasize, to modify its pitch range, etc. Eventually the system will be contained in a portable unit so that visitors can receive spoken information during their tour of a museum.

From a more general point of view, speech can be used when users search for a practical information, such as locating specific places or items. The path to follow can be shown on the screen, but it can also be explained vocally, so that, once again, the user doesn't have to look at the screen. Another use can be the providing of on-line and real-time information. For instance, a vocal message can inform visitors that there are few people in a special exhibit room at the present time, which could be interesting especially if the application enables users to choose a "to see" exhibit list at the beginning of their visit. A few minutes before closing, it can also be used to inform visitors that they should prepare themselves to exit the museum.

Non-speech sound can be used in different manners. For example, it can be used to provide an ambience, an atmosphere for the pieces of art. While playing a commentary on a specific object, a background music can also be played to project the visitor in the context : specific era, specific country, etc. And if we consider once again the idea of a "to see" exhibit list, and offer the possibility to automatically extend the list to related works of art, sound can be used to grab visitor's attention while wandering in the museum. The user would be informed that a piece of art related to his interests is just next to him. Or, if the user specifies on his PDA that he would like to access objects similar to the one he is looking at, speech could indicate him, as for the practical information, where to go.

Finally, [SottoVoce2002] explores the use of sound in an electronic guidebook that supports technologically mediated sharing of informational audio content. The idea is to enhance visitors' engagement with copresent by providing awareness and context from their activity. They hence consider groups of visitors as "paired users". The application runs on PDA with a headset (microphone and speaker) and uses WLAN using UDP/IP to communicate. The users obtain information about objects in their environment by selecting them on the visual interface, which uses a Web browser imagemaps metaphor. As they select an image map target, the guidebook plays an audio clip that describes the object. Paired users can listen to what their peer is listening at in a real-time and synchronized way : a person A can listen to her own clip, then switch to her peer B, but then A will have to listen to the other clip as from the part B has reached. This feature is called **eavesdropping**.

The analysis of users behavior showed eavesdropping integrates into, rather than supplants, visitors conversational interactions. The mutual use of the eavesdropping mechanism result in a more cohesive social experience than the one resulting from use of speakers in open air. The interaction between companions was increased, encouraging more natural conversations, as well as visitors awareness of the room and its contents. We could therefore also consider allowing users to listen to audio commentary and appreciations left by previous visitors of the museum...

3.3 The Use Of Vocal Input

Although this is not the true purpose of this work, vocal input is a theme that deserves a minimum of attention when considering multimodal access applications. Multimodal access enables users to interact with an application in a variety of ways:

- input with speech, keyboard, keypad, mouse and/or stylus;
- output as synthesized speech, audio, plain text, motion video and/or graphics.

Each of these modes could be used independently or concurrently [SALT]. In particular, speech recognition is the ability of a computer to understand and interpret speech. Given the late progress in software and hardware, this technology is becoming an efficient and affordable alternative to traditional input systems. Speech recognition is already used in various domains such as :

- The association of vocal dictation and word processing
- Remote command and control of electronic objects
- Phone information servers
- Mail services
- Vocal portals
- Etc.

Researchers are also looking at speech processing techniques as an extension of speech recognition. For instance, the Speech Application Language Tags (SALT) are a lightweight set of extensions to existing markup languages, in particular HTML and XHTML that enable multimodal and telephony access to information, applications and Web services from PCs, telephones, tablet PCs and particularly wireless personal digital assistants [SALT]. Wireless devices are becoming pervasive, but lack of a natural input mechanism interferes with adoption as well as with application development on these devices. Therefore, the main possibilities of speech are to offer :

- a more natural and intuitive interface, for inexperienced users (interfaces would also become lighter) ,
- an easier and more adapted navigation capacity,
- more accessible services,
- a better access to novice users, as well as for handicapped and blind persons.

The lack of time has compelled us to forsake any use and analysis of speech recognition for our matter. However, it is definitely an option to keep in mind for designing multimodal access application for wireless devices.

3.4 Conclusion

As we have seen, PDAs are somehow limited by their actual interaction capacities with the users. Sound can reduce this problem by enhancing program interface with auditory icons, earcons, or by sonifying widgets of the interface. But more specifically, for museum applications, speech and sound can be used to offer a richer experience to the visitors, as well as more liberty, autonomy and support. And further than the simple usability improvement, it seems that it can also have a positive social impact on the visitors of the museum, which could make visiting museums even more attractive. We will now focus on defining what kind of museum application we want to design on PDA.

4 Exposing The Main Objectives Of A Generic PDA Based Museum Guide

4.1 Context

Now that we have set the bases, we would like conceive a simple but generic PDA based guiding system for museums. It should be simple, because we have no guarantee on the future users profiles. Mobile technologies, such as PDAs, are quite recent and currently used by a minority of persons. The handiness of these devices can be problematic, so we should try to simplify as much as possible the steps necessary to perform a task.

It should also be generic, in order to specify any kind of indoor museum. The application will take a specially formatted text file as input to generate the environment. One of the main issues will be to use voice and sound in the programme to improve usability and interaction.

We assume that the user is not familiar with the museum that he is visiting, and that his art knowledge is not predetermined. When the person arrives at the museum, she will receive a PDA with the guiding application to assist him during his visit. Let us consider a few scenarios of what might just happen.

4.2 Scenarios

4.2.1 The Wandering Visitor

Today is Miss A.'s weekly day off. Like she didn't plan anything particular for her afternoon, she decides to go and visit the newly open nearby museum. Miss A. isn't really an art fanatic and she doesn't know exactly what there is to see in this museum. But she doesn't like following predefined tours, and she prefers to wander around, and discover the art pieces as they present themselves.

Therefore, she will select a free tour mode, which will enable her to view the overview map of the museum on the PDA. Whenever she wants, she will be able to switch to a particular room map by selecting the room from the museum overview map. A more precise map will be displayed, along with all the art objects available in the room. If there is an object that catches her attention, she will be able to access relative information by selecting it on the room map.

Miss A. will then access basic information on the object, such as its name, a picture of it, the author, the style, the period, the country and the material used. A little textual comment could also be displayed, and if she wants, Miss A. would be able to listen to the audio comment instead of reading it on the screen, and stop it when she wants. On the same information screen, she would also be able to ask for more comments on the object, as well as for related type of works.

If she asks for related type of works, she will access a screen where she will be asked to precise the type of similarity relation that she desires : author, style, period, country or material. She will then have to select the detail of the corresponding category, then continue with the free enhanced tour mode.

When she will be done with an object, she will go back to the room map, and at any time, she will be able to exit the program or switch back to the museum overview map.

4.2.2 Specific Art Domain Visitor

Mr B. is a fanatic of impressionist painters. So when he arrives at the museum, he will select a customized tour mode, which will allow him to choose a Style category. He will then select the Impressionism subcategory, where, by default, all the paintings of the Impressionism subcategory will be selected. Mr B. will be able to deselect the paintings he isn't interested with.

Once he validates his choice, he will start the visit. The museum overview map will be displayed, with the rooms to see highlighted. Mr B. will therefore be able to walk through the museum at will, and to access the detailed room map of his choice by selecting it from the museum overview map.

In the room map, all the impressionist paintings will be highlighted for his convenience, as well as the connections to adjacent rooms also containing impressionist paintings. Like Miss A., he will be able ask for information on any object he finds interesting, even those that aren't impressionist paintings.

If he finds one of these objects interesting/uninteresting, he should be able to add/remove it from the list of objects to be seen. And like Miss A. he will be able switch back to the museum overview map or quit the application whenever he wants.

4.2.3 Specific Object Visitor

Mrs C. is a very busy woman, but she has managed a little break to come and admire the exceptional last painting of Mr P., the locally famous new age artist. As she arrives at the museum, she will select the customized tour mode. First, she will select the Style category, then the New Age Painting subcategory. By default, all the paintings of the subcategory will be selected, so she will deselect all of them, then reselect the only painting she wants to see.

Like she is quite in a rush, she will start the tour at once. The museum overview map will be displayed, with the room containing the targeted painting highlighted. Mrs C. will directly go to the room containing the desired painting, and once she arrives there, she will select the room from the overview map, and hence switch to the room map.

On the room map, she will select the highlighted object, and access the information screen of the famous painting, just like Miss A. and Mr B. would have done it. After seeing it all about the painting, Mrs C. may realize that she still has a few minutes left before leaving. So she will be able to switch to the free tour mode, and just wander around like Miss A, asking for information on the items that attract her.

4.3 Task Analysis

Based on our scenarios and [Ciavarella&Paterno], we can precise 3 of the main types of tasks that the user will manage with the application. Because of the tour mode and the ToSee list, we will add a fourth type of task.

4.3.1 Orientation Within The Museum

For the convenience of the user, a museum map and a room map will be provided, with objects indicating the main pieces of art and their location. By selecting the objects, a basic information screen will be displayed : name and picture of the object, author, style, date, country, material as well as a short comment. Using a picture will allow the user to be sure that he is looking at the correct art object. The user will be able to ask for vocal output of the comment, or for a longer description of the object.

4.3.2 Access To Museum Information

The application should provide easy access to general and practical information on the museum: information on rooms (content, subject of exhibit), toilets and exit location, etc. This information will be available from the overview and the room detailed map.

4.3.3 Control Of The User Interface

The application should provide control of the audio comments, as well stop and start buttons. It should allow the user to browse through the different levels of the museum description and information screens, and to start and exit a tour.

4.3.4 Control Of The Tour Mode

Finally, we can add a fourth type of task, which consists in controlling the tour mode and the content of the objects to see list. Indeed, the user should be able to dynamically change the tour mode : free or customized. He should also be able to dynamically edit his ToSee list, by adding or removing objects... or even by bluntly resetting the list.

4.4 Use Of Sound In The Interface

In addition to the vocal output for comments on artwork, sound could be used when passing from the museum overview map to the room/section detailed map. A different sound could be played when passing from the room/section detailed map to the museum overview map. Sounds (chosen in function of the type of object for example) could also be played when selecting an object on the map. When a user enters a new room/section, a sound could also be played. Finally, sound could be used to enhance general interface buttons such as the OK button, the Cancel button, etc. We will determine the sound pattern to use in the next chapter.

4.5 Task Models

The main difference between a scenario and a task model is that a scenario indicates only one specific sequence of occurrences of one specific activity, while the task model should indicate all the main activities and their possible temporal relationship. We used the ConcurTaskTree notation [Paternò2000] to specify the task models of our application.

4.5.1 The ConcurTaskTree Notation

The ConcurTaskTree notation has been developped and exposed in [Paternò2000]. The task models allow the designer to specify hierarchically the activities which should be performed by the user, the application and their interactions. The specification indicates the temporal relationships among these activities and the objects (both the presentation object composing the user interface and the application objects) which need to be manipulated.

Such a notation offers quite some advantages. First of all, you can focus on activities. It allows designers to concentrate on the most relevant aspects when designing interactive applications that encompass both user and system-related aspects avoiding low levels implementation details that at the design stage would only obscure the decisions to take. In ConcurTaskTrees, tasks are of a certain type which is reflected in the icon that represents the task, as shown in Table 4-1.

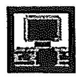
Task	Classification	Comment
	Abstract	All tasks that can be decomposed in subtasks of different categories.
	Human	All tasks independent of the application, and requiring human action : personal decision, hand-writing on a sheet of paper, ...
	Interactive	All tasks requiring an interaction between the user and the application : selection, edition, control, ...
	Automatic	All automated tasks : computing, comparison, printing, ...

Table 4-1 - The ConcurTaskTree Task Classification

Secondly, you can visualize more easily the hierarchical structure. Actually, when people have to solve a problem, they often tend to decompose it into smaller problems, while still maintaining the relationships among the smaller parts of the solution; the hierarchical structure of this specification has two advantages : it provides a large range of granularity allowing large and small task structures to be reused, it enables reusable task structures to be defined at both a low and a high semantic level. It also allows you to use a graphical syntax to structure the tasks. A graphical syntax is often more easy to interpret, in this case it should reflect the logical structure, so it should have a tree-like form.

Moreover, it offers a rich set of temporal operators. Operators for temporal ordering are used to link subtasks at the same abstraction level, as shown in Table 4-2. This sort of aspect is usually implicit, expressed informally in the outputs of task analysis. Making the analyst use these operators is a substantial change to normal practice. The reason for this innovation is that after an informal task analysis, designers should clearly express the logical temporal relationships. This is because such ordering should be taken into account in the user interface implementation to allow the user to perform at any time the tasks that should be active from a semantic point of view.

This notation has shown two positive results:

- It is an expressive and flexible notation able to represent concurrent and interactive activities, also with the possibility to support cooperation among multiple users and possible interruptions;
- It offers a compact and understandable representation; indeed, task trees are generally easy to understand and to build, while providing many information in an intuitive way without requiring excessive efforts from the users of the notation.

To illustrate the notation, we can consider a few simple examples to explicit :

- The inheritance of relationships with an hotel reservation (Figure 4-1);
- The relationships between tasks and subtasks with a simple interface to access information in a museum (Figure 4-2);

- The use of optional tasks in a flight reservation procedure (Figure 4-3).

Operator	**Representation**	**Comment**
Enabling 1	T1 >> T2	T2 becomes active when T1 terminates.
Enabling 2	T1 [ ]>> T2	When T1 terminates, it provides information to T2 and activates it.
Disabling	T1 [> T2	When T2 is activated, it disables T1.
Interruption	T1 |> T2	When T2 is activated, it interrupts T1.
Choice	T1 [ ] T2	One must choose between T1 and T2.
Iteration	T1*	T1 can be repeated any number of time to achieve the goal.
Iteration	$T1^n$	T1 must be repeated n times to achieve the goal.
Concurrency 1	T1 ||| T2	T1 and T2 can be done in any order.
Concurrency 2	T1 |[]| T2	T1 and T2 must be synchronized on certain events to exchange information.
Optionality	[T1]	T1 is an optional task.

Table 4-2 - The ConcurTaskTree Temporal operators

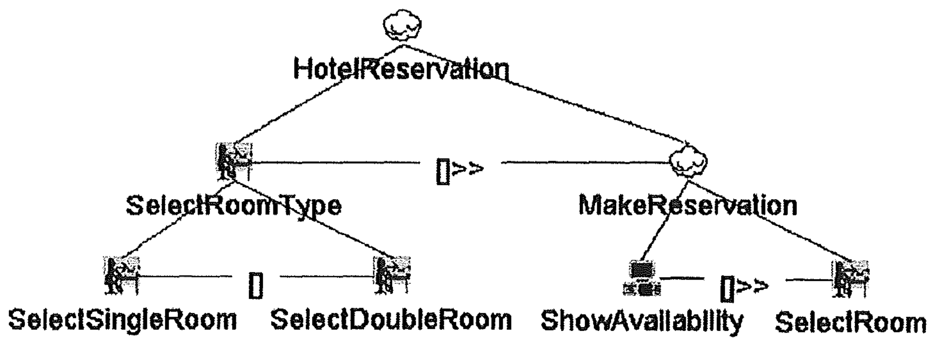


Figure 4-1 - Hotel Reservation ConcurTaskTree

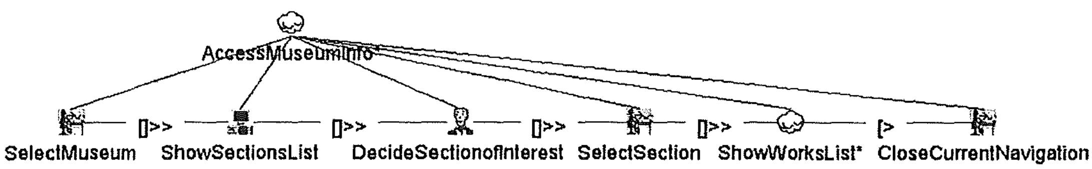


Figure 4-2 - Museum Information Access ConcurTaskTree

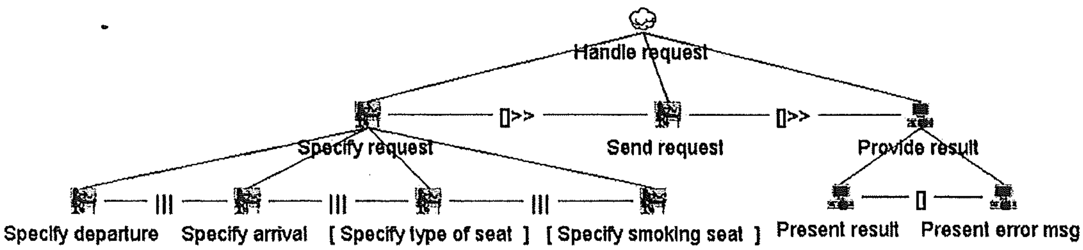


Figure 4-3 - Flight Reservation ConcurTaskTree

4.5.2 The ConcurTaskTree For Our Application

A first analysis, based on the scenarios we considered, led us to design the ConcurTaskTree given at Figure 4-4 and detailed in Appendix A, section A.1. We used the ConcurTaskTree Environment tool available at [CTTE] to build the task tree.

We clearly distinguish the different levels of tasks. First of all, the user would have to specify what kind of tour he desires, that is free or customized. He would then be able to perform interactive actions based on an overview map of the museum : ask for room information, edit his ToSee list, or switch to the room map. From the room map, he would be able to ask for object and room information. He should also be able to edit his ToSee list, switch to another room or go back to the overview map. These are the main features that our application should support.

4.6 Conclusion

We have rough-drawn the main objectives of our application, but we must still refine them. For this purpose, we will now analyse more precisely the kind of sound pattern we need for our application, then we will look at the implementation languages available.

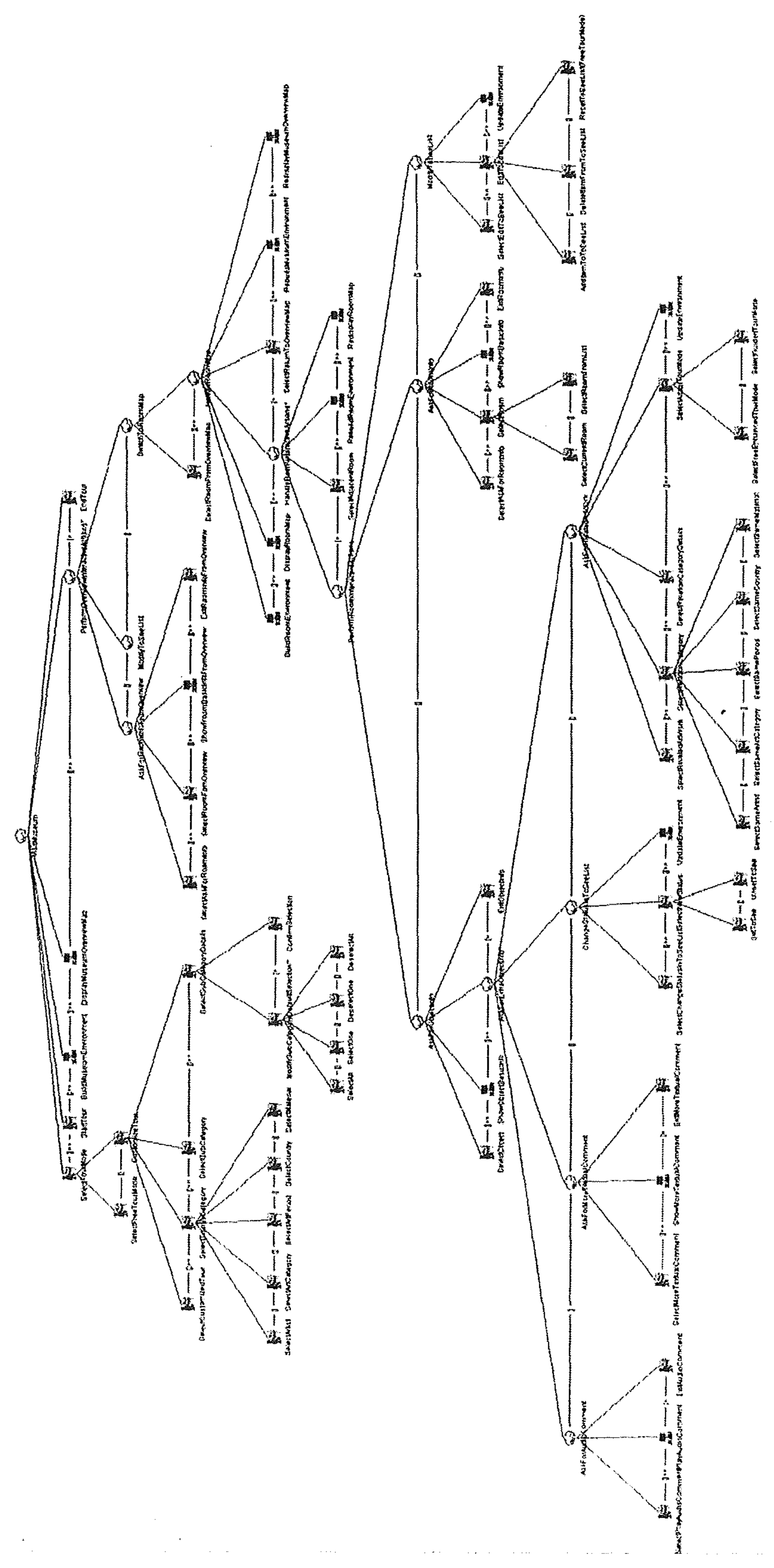


Figure 4-4 - The First ConcurTaskTree Approach For Our Application

5 Defining the appropriate sound pattern

5.1 Possibilities

Sounds can be used to indicate errors or supply redundant feedback for what is already graphically displayed on the screen. But they should also be used to present information that is not currently displayed so that users can be aware of it and deal with it [BresterPhD]. As we have mentioned earlier, there are three main methods when it comes to adding sound in an interface : auditory icons, earcons and speech synthesis. Given the actual limitations of speech synthesis on PDA, let's focus on the two former methods and take a closer look to them.

5.1.1 Auditoy Icons

As [Brewster94] reportedly stated, it is Gaver who first developed the idea of auditory icons, in [Gaver86]. Auditory icons are natural, everyday sounds which are used to represent actions and objects within an interface. The sounds have an intuitive link to the thing they represent, which allows them to provide information about many different things within the environment [Mountford&Gaver90]. For example, they can represent :

- Physical events: a dropped glass will make a different sound whether it broke or bounced.
- Invisible structures: by tapping on a wall, you will know if there is a hollow behind it.
- Dynamic changes: when you fill a glass with liquid, you can hear when it is full.
- Abnormal structures: a malfunctioning engine sounds different to a normal one.
- Events in space: if you hear the sound of footsteps, you know that someone is approaching, and you might even tell from which direction and at which distance.

Gaver used sounds of events that were recorded from the natural environment to create auditory icons. From this, he built the SonicFinder [Gaver89], which is an interface that runs on the Apple Macintosh alongside the ordinary Finder and provides auditory representations of some objects and actions within the interface. We must precise that this system was not designed for blind users but as an aid for sighted Mac users : it mainly added redundant information to the Macintosh interface. Most of the actions and objects that were sonified still had a graphical representation.

Gaver chose a "wooden" sound for files, a metallic sound for applications and a "papery" sound for folders. The larger the object the deeper the sound it made. So, when selecting an application by clicking on it, a metallic sound was played, which confirmed that it was an application. The deepness of the sound indicated the size of the application. Copying used the idea of pouring liquid into a receptacle : the rising of the pitch indicated that the receptacle was getting fuller, until completion of the copy. When deleting a file by dragging and dropping it into the wastebasket, the sound of smashing dishes was played to indicate destruction.

Gaver extended his ideas of auditory icons into the area of large-scale collaborative environments with SharedARK, the Shared Alternative Reality Kit [SharedARK90]. In this system a virtual physics laboratory was modelled, where multiple users could perform virtual experiments on objects, in an environment that extended beyond the limitations of their screens. Three groups of sounds were used : confirmatory sounds, process and state sounds as well as navigation aids.

The confirmatory sounds were based on the principles of the SonicFinder. Clicking on a button made a tapping sound (the same as for selecting a file or folder in the SonicFinder), putting an object on top of another made a wooden sound (similar to moving an icon over a folder or the wastebasket), and so on. To justify the redundancy between these sounds and the visual feedback already provided by the system, Gaver and Smith explained that the sounds seemed to provide feedback about actions in a way that was immediate, intuitive and engrossing.

Indeed, traditionally, sound is used to indicate that something is happening, but not what is happening, while auditory icons seem able to convey semantic information on events. One of the biggest advantages of auditory icons is the ability to communicate meanings which listeners can easily learn and remember, as Gaver & al. observed when studying the deployment of ARKola, the implementation of SharedARK for a soft drink factory [ARKola91]. Other methods like earcons use abstract sounds where the meanings are harder to learn.

Another system, called EAR (for Environmental Audio Reminders), demonstrated that auditory icons were also helpful for supporting collaboration in the office environment itself. This system played a variety of non speech audio cues to offices and common areas inside EuroPARC to keep the users informed about a variety of events around the building. [Buxton&al.94]

We can see that problems start to occur when we have to represent some abstract interface actions and objects which don't have an obvious representation in everyday sound. For the SonicFinder, Gaver used a pouring sound to indicate copying because there was no natural equivalent. He also suggested the use of movie like sound effects to create sounds for things with no easy representation, as for the sounds used in EAR. These sounds might have seemed frivolous because they were cartoon-like stereotypes of naturally-occurring sounds.

Brewster argues that this "deviation" may cause problems, because the sounds become more abstract than representational, and forces the advantages of auditory icons to be lost [Brewster94]. But Gaver & al. explained that it is precisely because theses sounds are stereotyped that they are effective. More "serious" sounds, such as electronic beeps or sequences of tones, would be likely to be less easily remembered than these.

Another problem was pointed out in [Blattner&al89]. It is that having a large number of auditory icons creates the problem of memorising each one as a distinct entity because there is no structure linking them together.

5.1.2 Earcons

Earcons differ from auditory icons in the types of sounds they use, which are abstract, synthetic tones that can be combined in different ways [Brewster94]. The sound messages created are used to represent parts of an interface and provide information to the user about some computer object, operation or interaction.

Unlike Gaver's auditory icons, earcons have a strong structure to link them together, and their built-in structure can easily be manipulated, by changing pitch or intensity for example. Nevertheless, there is no intuitive link between the sound and what it represents; the link must be learned by the listener.

Because they rely on musical parameters of sound, achieving the correspondence between sound and graphics is indeed more difficult using earcons, even though they can allow the representation of complex hierarchical structures [Brewster98]. For example, the earcon for deleting a file bears little relation to the graphic equivalent of dropping a file icon over the trashcan-icon. A different set of rules must be learned for the auditory and graphic components of such systems, while with auditory icons, a single set of metaphors can guide both aspects of the interface [Buxton&al.94].

5.1.3 Comparing Auditory icons and Earcons

There have been two main comparisons of earcons and auditory icons. The first was carried out by Jones & Furner and consisted of two experiments [Jones&Furner89]. In the first they tried to find out which types of sounds listeners preferred. Subjects were given typical interface commands such as delete or copy, and were played a sample earcon, an auditory icon and some synthesised speech. In the experiment, earcons were preferred to auditory icons. A second experiment was carried out to see if subjects were able to associate sounds with commands. The subjects were played a sound and had to match it to a command in a list. This time auditory icons proved to be more popular than earcons. This may have been because family earcons would initially be harder to associate to commands, as they have no inherent meaning and take time to learn, whereas auditory icons have a semantic relationship with the function they represent.

A further comparison experiment was undertaken by Lucas [Lucas94]. He conducted a more detailed analysis of earcons, auditory icons and synthetic speech. He presented subjects with sound stimuli of the three types, and they had to choose which command was related to which sound. He conducted a second trial a week after the first. His results showed that there was no difference in response time between earcons and auditory icons, while speech was significantly faster. There were no differences in response between trial one and two, however, subjects did make fewer errors on trial two. This would indicate that with more training auditory cues could easily be learned. There were no significant differences in error rates between earcons and auditory icons and as you might guess, there were no errors in the speech condition, as the stimuli were self-explanatory. After the first trial, half of the subjects were given an explanation of the sounds used and the design methods. These subjects showed a decrease in error rates on trial two, which indicates that a knowledge of the auditory cue design does improve the accuracy of cue recognition.

These two comparisons have shown little difference between earcons and auditory icons. It may be that each has advantages over the other in certain circumstances and that a combination of both might be best. In some situations the intuitive nature of auditory icons may make them favourable. In other situations earcons might be more suitable because of the powerful structure they contain, especially if there is no real-world equivalent of what the sounds are representing. Indeed, there may be some middle ground where the natural sounds of auditory icons can be manipulated to give the structure of earcons. Cohen proposes that there is a continuum of sound from the literal everyday sounds of auditory icons to the abstract sounds of earcons [Cohen93]. Objects or actions within an interface that do not have an auditory equivalent must have an auditory icon made for them.

This then has no semantic link to what it represents : its meaning must be learned, which means that the auditory icon becomes more abstract. When hearing an earcon, the listener may hear and recognise a piano timbre, rhythm and pitch structure as a kind of 'catch-phrase'; he/she will not hear all the separate parts of the earcon and work out the meaning from them. The earcon will be heard more as a whole source and thus the perception of the earcon moves more towards the representational side of the continuum. Therefore, earcons and icons are not necessarily as far apart as they might appear. Both have advantages and disadvantages but these can be maximised/minimised by looking at the properties of each.

5.1.4 Conclusion : choosing between auditory icons and icons

Obviously, both sonification methods could be valuable for our project. Our application is not a daily application, that is, it will not be used on a daily basis : most of the users will be occasional visitors that are not familiar with the application. Therefore, the sound pattern used should be meaningful and immediately comprehensive, because the users won't be able to familiarize themselves with the application by using it regularly.

We will hence develop two sound patterns, in order to analyse which one suits the best the users:

- an earcons set, with specially designed synthetic sounds;
- an auditory icons set whose meaning should be more immediate than earcons. The problem of representing unnatural events with auditory icons, will be discussed in the guidelines proposed by Gaver, Buxton & al.

5.2 Generic Sonification guidelines

In this section we present general guidelines developed by Patterson for warning purpose, as well as other guidelines put together by Warin and McCormick & Sanders. These guidelines emphasize the form of the signifiers. The set of signifiers needs to be discriminably different to represent different signifieds and to prevent ambiguous combinations [Warin2002].

5.2.1 Warning guidelines

Patterson produced a set of guidelines covering all aspects of warning design (oriented for aircrafts), in [Patterson82] & [Patterson89]. For example, the warnings he suggests use much lower intensities and slower onsets/offsets to avoid startling the pilot. The main points addressed by his guidelines are:

- **Overall level :** The lowest intensity level of a warning sound should be 15dB above the threshold imposed by background noise. The upper limit is 25dB above the threshold.
- **Temporal characteristics :** Component pulses of a warning sound should have onsets and offsets 20-30 ms in duration. These avoid a startle response in the listener. Pulses should be 100-150 ms in duration with a 150 ms inter-pulse gap for urgent sounds and a 300 ms gap for non-urgent sounds. Distinctive rhythms of five or more pulses should be used.
- **Spectral characteristics :** The fundamental frequency of a warning should be within the range 150-1000 Hz. There should be four or more component harmonics to help avoid masking. The overall spectral range of warnings should be 500-5000 Hz.
- **Ergonomics :** Manual volume control should be avoided and automatic control restricted to a range of 10-15 dB variation. There should be no more than six immediate action warnings.
- **Voice warnings :** These should be brief and use a key-word format. They should not be repeated in a background version of the warning. Voice warnings used as immediate awareness warnings should use a full-phrase format and be repeated after a short pause.

5.2.2 Spatial Sonification Guidelines

In the pragmatic approach presented in [Warin2002], guidelines are given for spatial information and orientation :

- **Eye relief :** Auditory displays should be used to relieve the eyes.
- **Presentation of restricted information :** Auditory displays should be used to present restricted information such as :

    - **Yes / No information** and indications of amount or degree, which allows to spot deviations from normal behaviour.
    - **Continuous information**, such as radio-range signals for aircraft courses.
    - **Automatic information**, such as recorded word signals as from automatic enunciators.
- **Speech channel saturation :** Auditory displays of tonal or noise signals should be used when speech channels are already fully employed.
- **Single representativity :** Representation should be confined to a single dimension, because multidimensional displays are less effective than their visual counterparts.
- **Abnormal behaviour emphasis :** A standard stimulus should be used in normal conditions, while abrupt changes would indicate abnormality. Human listeners are sensitive to frequency or intensity change, but poor at identifying a unique signal.
- **Intensity rather than frequency :** Changes in intensity should be preferred to changes in frequency, because they are more detectable by human hearing.

5.2.3 Preferring Auditory Displays to Visual Displays Guidelines

McCormick & Sanders gave some guidelines explaining where auditory displays are preferable to visual displays [Warin2002]. They should prevail when :
- The origin of the signal is itself a sound.
- The message is short and simple.
- The message will not be referred later.
- The message refers to event in time.
- The message calls for immediate action.
- The information is continuously changing.
- The visual system is overburdened.
- The receiver moves from one place to another.

5.3 Auditory Icons Guidelines

These guidelines are based on [Buxton&al.94].

5.3.1 Issues

A number of issues become clear in considering the type of interfaces we want to implement :
- **Mapping :** How should sounds be mapped to events?
- **Vocabulary :** What sounds should be used in a recognizable and discriminable way?
- **Annoyance :** How can we use sounds without disturbing users?

5.3.2 Mapping Sounds to Events

The two defining features of auditory icons are the use of environmental sounds controlled along parameters of the events which cause them, and the use of intuitive mappings between the sounds and the computer events they indicate. It is easy to say that the mapping between an auditory icon and the event it represents should be intuitive, but more difficult to know what makes a particular mapping intuitive.

What is being mapped to what?

If we consider what happens when we throw a computer file away, there are several ways to consider what is going on. At the level of the display, we are simply throwing the file icon into that of the trashcan, knowing the file icon will disappear when the trash is "emptied."

But of course the purpose is not just to get rid of the icon, but also the file itself. Throwing its icon becomes equivalent to throwing a real file into the trash; not only does its apparition go away, but the file itself goes away with all the attendant consequences : it is no longer accessible for reading, copying, etc. The display is the outward appearance of the model world of files and wastebaskets set up by the computer, a model world which not only determines the current display, but also future possibilities for action.

Still, in reality we aren't throwing away a file at all, but manipulating a graphical output, which is linked to data structures in the computer. However, what is really happening behind the scene is not crucial for the user, he only needs to know that the deletion will be processed. In fact, the reality of the computer is simultaneously hidden and made accessible to us by the conceptual mapping that allows electronics to be thought of as information, information to be thought of as variables, variables to be grouped into structures, and structures to be called "files".

Types of mapping

Conceptual mappings between the reality of the hardware structure and the model world of files, trashcans and so on, are metaphorical. But there are three different kinds of mappings that may allow the model world to be expressed as a display : symbolic, metaphorical, and iconic mappings, which are distinguished by the degree to which they are arbitrary or lawful; this also has profound effects on the ease with which they are learned and remembered. Let's consider the deletion example once more.

The first kind of mapping is **symbolic** : an arbitrary sign is used to mark the file, for example an X for the visual display, and a simple beep for the auditory one. As we can imagine, symbolic mappings are difficult to learn because they are entirely arbitrary and rely on social convention for their establishment.

The second kind of mapping is **metaphorical** : the displays form an analogy to the real-world metaphor of deletion. For the visual display, this is accomplished by fading the icon, as if it were in the process of disappearing. For the auditory display, an earcon is used in which a motive standing for a file is diminished in amplitude. The mapping between the file and the motive in this example is wholly symbolic in that the choice of motive is arbitrary. The metaphor here is between the deletion of the file and the decreasing volume of the motive. Metaphorical mappings are easier to learn because they rely on similarities between the representation and the thing to be represented.

Finally, the third kind of mapping is **iconic** : the display resembles the real-world event in a lawful way. In the graphical version, an icon of the file is dropped over the icon of the trashcan. For the auditory display, the sound of an object dropping into a (full) trashcan is played. Iconic mappings are easiest to learn because they rely on the similarity between a representation and the thing it represents : in the case of perceptual mappings, this means that the representation must look (or sound) like the event it represents. Iconic mappings allow learners to apply the skills they already possess in the everyday world to the new model worlds created by computer technology.

From this point of view, designers of auditory interfaces should be concerned with using mappings between sounds and events that are as lawful as possible. In other words, they should avoid creating new symbolic languages for sounds, and instead seek iconic or metaphorical mappings. This is one of the key concerns behind the concept of auditory icons.

5.3.3 The Vocabulary of Auditory Icons

The mapping between sound and event is crucial for auditory icons. As auditory icons could be constructed from musical tones or motives if they were mapped in an iconic (or strongly metaphorical) way to their intended message, the use of everyday sounds for auditory icons may seem superfluous.

But the fact that most auditory icons use everyday sounds is no accident. Instead, it is precisely because everyday sounds are defined in terms of events that lawful mappings can be created between auditory icons and the events they convey. Instead of trying to map dimensions of a computer event to dimensions of sound, one can now map events to events.

Thus, most auditory icons should use everyday sounds. If a file is selected by clicking the mouse over it, then a tapping sound should be produced. If it is moved by dragging it over a background, then a scraping sound provides appropriate feedback. Whenever possible, an iconic mapping between the event in the model world and the sound that is heard should be used.

Metaphors, Sound-effects, Clichés, and Genre Sounds

As we have seen, literal iconic mappings between sounds and computer events are sometimes impossible to achieve. Take the example of copying a file. What sound should this action make? It might be tempting to use the real-world analogy of photocopying a document. But this seems misleading in using the sound made by a repetitive, step-by-step operation to indicate a more continuous one : we can stop copying a paper document after one page, whereas electronic file copies are typically all-or-nothing. In addition, such a sound doesn't indicate some information that might be of value to users, such as the progress of the job. In this case, it seems useful to go beyond a literal mapping to create a sound that more fully indicates the attributes of copying in the model world of the computer.

One way to go beyond literal mappings is to use **metaphors**. Remember the sound of pouring a liquid into a container to indicate the copying of a file in the SonicFinder for instance. As the operation neared completion, the sound rose in pitch to indicate that the container was getting full. Thus the sound conveyed useful attributes of the copy operation:

- that it involves transferring information into a new container,
- that the operation is relatively continuous,
- and that it is possible to predict how close it is to completion.

However, metaphors such as these may also have unwanted implications : for instance, since the liquid must come from somewhere, the sound might be taken as indicating a move rather than a copy. The sound also does not differentiate reading and writing stages of a copy operation, which might be useful information in some circumstances. Finally, using the sound of a container being filled implies a constrained space waiting for the new information. In the case of the copy operation, this space is determined by the size of the information to be transferred, as if bottles changed size to automatically hold just the amount of liquid to be poured. As this example indicates, there is a trade-off in designing and using metaphors such as this : they provide a lawful structure that can guide users' expectations, but may sometimes be misleading.

Another solution is to use **sound-effects**. Although sound-effects are not naturally-occurring everyday sounds, if they are well designed, they seem to be heard in many of the same ways. Think of the sound made by a ray-gun in a science fiction movie, for example. If well designed, it will indicate many of the properties of the event : that it is electronically produced, quick and powerful. Such sounds have never existed, but they are not arbitrary : they seem to extend the laws relating events to sounds.

Sound effects can be very useful when creating auditory icons. For example, in designing the SonicFinder it became clear that sounds should accompany opening and closing windows. Many attempts were made to use, literally, the sounds made by opening and closing glass windows, but none sounded appropriate. Finally it became clear that opening and closing windows in a computer is only metaphorically related to the real-world event. In the graphic interface, windows do not slide open, but instead expand into existence from their associated icon. They appear from nowhere in a way that has no literal analog in the real world – though such appearances are common in science fiction movies. Thus it was more appropriate to use a kind of "whooshing" noise to indicate this event, as if air rushed around the window as it expanded into existence.

Sound effects such as these are closely related to the **genre sounds** used by Cohen [Cohen&Koizumi92] & [Cohen&Ludwig91]. As he points out, sounds from popular TV shows, movies and the like can be incorporated successfully into auditory interfaces. In some cases, they may provide information that is difficult to convey using more literal sounds. For instance, designing a sound for the teleporter in SharedARK might have a challenge because, clearly, no analogous mechanism exists in the everyday world, but it does in science fiction. Thus the sound used to convey this event, basically a slowly growing, inharmonic sound, was based on the sounds used to indicate teleportation in popular TV shows.

One of the drawbacks of genre sounds, as Cohen mentions, is that they may not be interpretable by people who are not familiar with their sources. This problem may be overcome to some degree by creating sound effects that are custom-made for particular applications. As they rely on extensions of physics, rather than familiarity with particular movies or television shows, such sounds can be expected to be more readily recognized even by users unfamiliar with the specific reference.

Finally, both sound-effects and genre sounds are closely related to another class of sounds that can be used to extend auditory icons. These are **auditory clichés**, that is, sounds which are arbitrarily related to their meaning, even in the everyday world, but which are so strongly part of the culture, that they can be used as everyday sounds are. For instance, from the point of view of everyday listening, the sound of a telephone bell simply conveys information about a hard mallet repetitively striking a small metal one. But because in many cultures this sound is strongly linked with the request to make an electronic connection to a remote colleague for the purposes of communication, it can be used to indicate analogous computer connections (as it is, for instance, in EAR). Similarly, from the point of view of everyday listening, the sound of somebody knocking on a door simply indicates a somewhat softer mallet striking a large wooden surface. Again, because the symbolic message it conveys is so widely recognized, auditory icons can be created which map to this message rather than the more literal meaning of the sound.

In sum, the vocabulary for auditory icons is driven more by the efficacy of the mappings allowed than by the sounds themselves. In general this means using everyday sounds that indicate their real-world meaning due to the laws of physics. But it is also possible and useful to expand the vocabulary for auditory icons to include metaphors, sound-effects, genre sounds and clichés. Such sounds can vastly expand the repertoire of auditory icons and the information they convey. Still, it must be recognized that the mapping from sound to meaning becomes more arbitrary the greater the move from everyday sounds and iconic mappings. In general, mapping a file selection to a tapping sound can be expected to be more readily guessed, learned, and remembered than mapping, for instance, an instantaneous move to the sound of a television transporter.

5.3.4 Annoyance

A final issue for auditory icons and for auditory interfaces in general is how to design sounds that will not prove annoying to users. Indeed, one of the first responses most people have to the idea of auditory interfaces is to say "I like to work in peace and quiet so why would I possibly want a computer that makes noise?"

In truth, even in quiet places, we are surrounded by sounds. They can be subtle, the can be almost inaudible, but they are there. In an office for example, you are likely to hear sounds such as the whoosh of a ventilator, the hum of a computer, the shuffle of papers on nearby desks, footsteps as somebody walks by, even if it is a very quiet office. Most of these sounds are hardly noticed, most of them are not annoying, and many of them are useful in maintaining a background awareness of ongoing events. And when you think about it, would you really be able to work in an anechoic office?

The point is hence to provide a model of how sounds can be used without being obtrusive: in general, the goal is to create sounds that are as subtle, unobtrusive, and informative as the sounds we already hear in the office. Therefore, the sound shouldn't be too loud or too complex.

Experience

We can take advantage of results in psychoacoustic studies of urgency and annoyance [Edworthy&al.90] to develop guidelines for the creation of auditory interfaces. Experiential factors also affect the annoyance of sounds and thus must also be taken into account. Indeed, we often find that the melody that was so instantly hummable becomes sickening after repeated hearing. Clearly experiential affects play a large role in determining what we will find annoying.

Experimental and theoretical studies of emotion and aesthetics are useful in understanding the role of experience. Two simple factors have emerged as most relevant: familiarity and complexity. Both highly familiar and unfamiliar things are liked less than some moderately familiar item; both highly complex and simple things are less preferable than intermediate values. These findings can be summarized succinctly in the **principle of optimal psychological complexity**:

> *Moderately complex sounds are less annoying than very simple or very complicated ones; however, perceived complexity decreases with familiarity, highly complex and simple things are less preferable than intermediate values.*

This principle implies a variety of possibilities for the design of informative sounds. Complicated sounds may be used if they are heard relatively frequently. Although initially likely to be annoying, they will usually become acceptable over time. Conversely, very simple sounds should not be repeated too often: though initially acceptable, they become irritating if heard to often.

Once again, the ideal solution is maybe to choose a median approach, by using moderately complex sounds, but making subtle variations fairly often. This approach works well with the strategy behind parameterized auditory icons, in that it encourages more information to be conveyed by sounds, thus producing richer auditory icons. For example, in the SonicFinder the size of objects usually affected the sounds they made, with large objects making lower sounds than small ones. This had the effect not only of conveying potentially valuable information, but also of preventing the impact and scraping sounds from becoming too annoying. It is fortunate that the least annoying auditory interfaces may also be the most informative.

Semantic effects

To acoustic and experiential factors that influence annoyance must be added semantic factors. For instance, people regularly find highway sounds more annoying than birdsong, despite the fact that birdsongs have more of the acoustic parameters corresponding to annoyance, because of their high pitch, abrupt envelopes, fast changes, etc. It appears that the semantic connotations of the sounds can override acoustic and even experiential considerations in determining annoyance.

Even everyday listening does not directly address higher-level semantics of many sounds, as can be seen, for instance, by the fact that a literal account of everyday listening would describe a telephone bell in terms of a piece of metal being struck repeatedly by a hard clapper. Still less can the emotional reactions to the semantics of sounds be predicted from theory. Nonetheless, designers should be sensitive to issues surrounding the semantics of sound in developing auditory interfaces.

Clarity versus obtrusiveness

The principles guiding the design of intuitively-obvious and unobtrusive auditory icons may often be at odds with one another. Everyday sounds tend to have many of the acoustic features that characterize annoying sounds. For instance, consider a metallic impact sound, used often in the auditory interfaces described above. For many object mappings, the sound will be high-pitched, inharmonic, abrupt, and atonal. This is the recipe for an urgent sound, according to our discussion above. On the other hand, if we want the sound to be unobtrusive, we expect it to be low-pitched, harmonic, have smooth envelopes, and have tonal relations with one another!

There are several heuristics for reducing this tension between identifiability and obtrusiveness. First, everyday sounds may be shaped acoustically to reduce their annoyance. For instance, the attack of percussive sounds can be slowed slightly to reduce the tendency for abrupt sounds to demand attention. Most sounds can be low-pass filtered, reducing the amplitude of annoying high-frequency partials. Many long sounds can be shortened considerably, and thus made to convey information concisely. In all these cases, effective variations can be made without reducing sounds' identification if the changes are kept small.

In addition, the tension between the acoustic factors leading to identifiability and those that produce annoying sounds can be mitigated by taking advantage of experiential and semantic factors. This is, in fact, one of the major advantages of auditory icons: They are designed to be similar to the sorts of sounds one hears in the everyday world, and thus less distracting than introducing a new vocabulary (e.g. music). Because they fit with the existing auditory ambience, they benefit from familiarity and semantic congruity. Moreover, they can be designed to reach an optimal level of familiarity, as described above, and with the aim of using semantically pleasing – or better, neutral – sounds.

5.4 Earcons guidelines

The creation of a set of earcons to sonify an interface depends on the interface and what the application behind it does. However, Brewster developed some general guidelines for use when creating earcons [Brewster&al95B]. These guidelines form part of the structured method Brewster developed for adding sound to human-computer interfaces [Brewster94]; they suggest what sounds should be used. A designer could use the guidelines to create earcons that could effectively communicate complex information in sound.

5.4.1 Where to begin

When designing a family of earcons, one should start with timbre, register and rhythm, which can be used to create the basic structure. For example, each family of earcons might have a different timbre and default register. This would differentiate it from other families of earcons. Each family could also be given a different spatial location. Rhythm can then be used to create the major sub-groups within each family. To further differentiate the sub-groups pitch, intensity, chords or effects such as chorus or delay can be used. Care must be taken to make sure that the earcons are recognisably different, because small, subtle changes between earcons are unlikely to be noticed by anyone but skilled musicians.

5.4.2 Timbre

One should use musical instrument timbres, because simple tones such as sinewaves or square waves are not effective [ANPT73]. Where possible timbres with multiple harmonics should be privileged, as this helps perception and can avoid masking. Timbres that are subjectively easy to tell apart should be used. For example, on a musical instrument synthesiser use 'brass' and 'organ' rather than 'brass1' and 'brass2'. However, instruments that sound different in real life may not when played on a synthesiser, so care should be taken when choosing timbres. Using multiple timbres per earcon may confer advantages when using compound earcons

5.4.3 Register

If listeners are to make absolute rather than relative judgements of earcons then pitch/register should not be used. A combination of pitch and another parameter would give better rates of recall. If register alone must be used, then there should be large differences between earcons, although even then it might not be the most effective method. Two or three octaves difference should be used. This is not a problem if relative judgements are to be made.

5.4.4 Pitch

Pitch should not be used on its own unless there are large differences between those used. Complex intra-earcon pitch structures are effective in differentiating earcons if used along with rhythm and another parameter. Some suggested ranges for pitch are from a maximum of 5kHz (four octaves above C3) to a minimum of 125Hz to 150Hz (the octave of C4).

5.4.5 Rhythm, tempo and duration

Rhythms should be as different as possible. Putting different numbers of notes in each rhythm is very effective. Small note lengths might not be noticed so do not use notes less than sixteenth notes or semi-quavers. This depends on the tempo. If 180 bpm is used then sixteenth notes last 0.0825 sec. Earcons should be kept as short as possible so that they can keep up with interactions in the interface being sonified. Two earcons can be played in parallel to speed up presentation. Earcons with up to six notes played in one second have been shown to be usable. In order to make each earcon sound like a complete rhythmic unit, the first note should be accented (played slightly louder) and the last note should be slightly longer. Changing the tempo, speeding up or slowing down the sounds, is another effective method for differentiating earcons.

5.4.6 Intensity

Great care should be taken over the use of intensity because it is the main cause of annoyance due to sound. Suggested ranges should be kept between a maximum of 20dB above threshold and a minimum of 10dB above threshold. The overall sound level should be under the control of the user of the system. Earcons should all be kept within a close range so that if the user changes the volume of the system no sound will be lost.

5.4.7 Spatial location

Stereo position or full three-dimensions could be used if extra spatialisation hardware is available. This is very useful for locating the sound source, and, for example, differentiating parallel earcons playing simultaneously. It can also be used with serial earcons, for example each family of earcons might have a different location.

5.4.8 Making earcons attention-grabbing

Capture the listener's attention can be achieved in different ways. It can be done by using intensity, which is crude but effective and very common. However, it is potentially annoying for the primary user and people nearby so other methods should be privileged. Rhythm or pitch can be used (perhaps combined with lower intensity), for example, because the human auditory system is very good at detecting dynamic stimuli. If a new sound is played, even at a low intensity, it is likely to grab a listener's attention without disturbing nearby persons. As another example, if the rhythm of an earcon is changed (perhaps speeding up or slowing down) this will also demand attention. High pitch, a wide pitch range, rapid onset and offset times, irregular harmonics and atonal or arrhythmic can also be used. The opposites of most of these can be used to make sounds avoidable but in this case the main parameters are low intensity and regular rhythm.

5.4.9 Combinations

When playing earcons one after another use a gap between them so that users can tell where one finishes and the other starts. A delay of 0.1 seconds is adequate. If the above guidelines are followed for each of the earcons to be combined, then recognition rates should be similar to that of individual earcons.

5.5 The sound pattern for our application

As mentioned previously, we will define two distinct sound patterns : an auditory icon based one, and an earcon based one. But first of all, we should define the generic sounds for interacting with common dialogs.

5.5.1 Interacting with common dialogs

Common dialog regroup all the dialog boxes, menu items, etc. For interacting with such dialogs, we choose a single sound pattern.

Yes / Confirm / Next buttons

Clicking on a confirmation button has no literal counterpart in everyday sounds. The idea is therefore to use a single tone which suggests a "positive" click.

No / Cancel / Back buttons

Inversely, for annulation buttons, we should use a sound that suggests a "negative" click. The true value of the chosen sounds should be tested to see whether users truly make the difference between these arbitrary "positive" and "negative" sound.

Select / Deselect buttons

When it comes to selecting an object, the interface actually uses 2 types of dialog controls, which are radio buttons and list boxes. The problem is that, depending on the implementation language, the selection process can by automatically managed without the programmer's intervention. Hence, when possible, we will choose a "neutral" click for selecting items, by comparison with the two previous sounds.

5.5.2 Interacting with the maps with auditory icons

The real difference between the auditory icon pattern and the earcon pattern will occur when interacting with the maps. To define our auditory pattern, the idea is to use an auditory icon metaphor to illustrate the level switching.

Switching from the Overview to the Room level

As this action basically consists of zooming in a portion of the overview map, we could use a sound suggesting we are getting closer to a room. It was not easy finding a sound suggesting this drawing nearer, but we finally chose an excerpt of the sound of a landing plane.

Switching from the Room to the Overview level

Now that we are zooming out, there is an underlying idea of going up, or stepping back. That is why we used a sound similar to the previous one : an excerpt of the sound of a plane taking off.

Switching from a Room to another

Although there is not always a door between the rooms in a museum, we chose to use the sound of a closing wooden door to indicate room switching. This sound is used when the user is navigating from a room to another from the screen, or when an IR message is received from a new room (which then means that the person has *physically* changed room).

Switching between a Room and an Object

For switching between a room and the information screen of an object, we decided to use a book metaphor : when accessing the information screen of an object, it would be as if the user was turning the pages of a book. When returning to the room map, the same sound will be played.

5.5.3 Interacting with the maps with earcons

The choice of the earcon pattern was somehow more problematic because of it's inherent abstract nature. However, we managed to distinguish three classes of earcons, distinguished mainly by the intensity, the duration and the timbre. The sounds we used were based on the ones available on [DeHaan].

Switching between the Overview to the Room level

Switching from the overview map to the room map has the same semantic orientation than switching from the room map to the overview map. That's why we chose two earcons based on a sequence of three notes, having similar intensity, timbre and duration.

Switching from a Room to another

The earcon chosen for switching from a room to another consists of a sequence of two notes, which implies a shorter duration than the two previous earcons. The timbre is also different.

Switching between a Room and an Object

Finally, the earcon chosen for switching between a room and an object also consists of a sequence of two notes, but with another timbre (rather like a little bell).

5.5.4 Localisation Issues

As we have already mentioned, we should play the switching room sound when an infra-red signal is received from a new room. This feature is very important to convey undisplayed information, if the user is looking at an art object information screen on the PDA for example. As he didn't manually switch room, he will be informed of the physical change in the environment.

We should mention the **soundholder,** one the most interesting concepts in SharedARK, which basically was an auditory landmark [SharedARK90]. A soundholder can be placed anywhere in the system and will constantly emit a sound whose volume decreases as the user moves away from it or increases as the user gets closer. Gaver and Smith suggested using environmental sounds such as bird calls or burbling streams for these, as they are very distinct and easy to remember.

This concept could be very suitable to our application if we had a real localisation system : a soundholder could be associated with each object selected by the user (a "ToSee" object). When a user would come near an object that he had selected in his ToSee list, the soundholder would enable him to locate the object and the distance to it.

5.6 Conclusion

In this chapter, we focused on the two main sound patterns used in interfaces. Now that we defined the sound pattern we should use based on sonification guidelines, we will focus on technologies and programming languages available to provide sound and speech features for a museum application on PDA.

6 Choosing The Appropriate Programming Language.

6.1 Introduction

In the field of mobile programming, two main implementations languages are emerging : Java and C++. Although quite similar from the code point of view, their usability and distribution are quite different. Let us take a look at what is currently available for us.

6.2 Java for PDAs.

Java™ 2 Platform, Micro Edition (J2ME™) is a highly optimized Java runtime environment, specifically addressing the vast consumer space, which covers the range of extremely tiny commodities such as smart cards or a pager all the way up to the set-top box, an appliance almost as powerful as a computer. Different J2ME™ technologies are available for PDAs, and the choice of the technology should be driven by the Operative System and the hardware of the device.

6.2.1 PersonalJava™

The PersonalJava™ application environment (PJAE) is a Java™ application environment that executes software written in the Java programming language [Java1]. The PJAE addresses the software needs of networked applications running on personal consumer devices such as set-top boxes, PDAs and smart phones rather than desktop computers. The PJAE provides a virtual machine based on *The Java™ Virtual Machine Specification, First Edition.* It uses JDK 1.1.8 as its base and adds security as specified in Java 2 SDK, Standard Edition, v 1.2.

The free PersonalJavaTM Runtime Environment available on Sun's website is a binary version of the PJAE for Windows CE devices. This code should be able to run on devices running version 2.11 of the Windows CE operating system with one of the following processors: MIPS (R4000 Compatible) / SH3. The device should have at minimum 16 MB of storage and a 16 colour or grayscale display. At present time, there are no SH1, SH2, x86 or StrongARM implementations available.

6.2.2 CLDC and MIDP, for low level PDAs

The Connected Limited Device Configuration (CLDC) is one of two configurations defined through the Java Community Process to be part of the J2ME Platform [Java2]. CLDC is the foundation of the Java runtime environment targeting small, resource-constrained devices, such as mobile phones, mainstream personal digital assistants, and small retail payment terminals.

Sun offers two virtual machines to support the CLDC. The K virtual machine (KVM) is a virtual machine designed from the ground up with the constraints of inexpensive mobile devices in mind. It is named to reflect that its size is measured in the tens of kilobytes. CLDC with KVM is suitable for devices with 16/32-bit RISC/CISC microprocessors/controllers, and with as little as 160 KB of total memory available for the Java technology stack. 128 KB of this is for the storage of the actual virtual machine and libraries, and the remainder is for Java applications.

The CLDC HotSpot virtual machine is targeted for newer generation devices with larger available memory. The HotSpot Implementation is suitable for devices with 32-bit RISC/CISC microprocessors/controllers, and with 512KB to 1MB of total memory available for the Java technology stack, including applications. This Virtual Machine adds adaptive compilation for a dramatic increase in performance on ARM processors, with minimal increase in memory footprint. Version 1.0 of CLDC HotSpot Implementation now being offered by Sun Microsystems is integrated with CLDC. This initial offering conforms to the CLDC Specification version 1.0 and Technology Compatibility Kit (TCK) 1.0. To complete the Java technology stack, a compatible implementation of the MIDP 1.0 Specification is also offered.

Precisely, combined with a set of device specific Java APIs, such as the Mobile Information Device Profile (MIDP), CLDC provides a complete J2ME runtime environment for small resource-constrained devices [Java3]. The MIDP is a set of Java APIs which provides (together with CDLC) a complete J2ME application runtime environment targeted for mobile information devices, such as mobile phones and entry level PDAs. The MIDP specification addresses issues such as user interface, persistence storage, networking, and application life cycle.

Regarding multimedia features of CDLC, the Mobile Media API (JSR 135) is an optional package for CDLC that specifies a small multimedia API for Java enabled devices, from simple cellular phones to more sophisticated, multimedia devices. This API allows simple access and control of audio and video time-based media. It is both scalable and extensible to support more sophisticated multimedia features.

MMAPI is designed to be protocol and format indifferent. For example, it does not specify that transport protocols such as HTTP or RTP, or media formats such as MP3, MIDI or MPEG-4 have to be supported. However, it contains all of the functionality needed to support these protocols and many more : single tone, MIDI (.mid), Wave (.wav), PCM, MPEG-1 (.mpg). It allows API implementors and Java profile creators to choose which formats they will support.

The Mobile Media API Reference Implementation (MMAPI RI) is based on MMAPI [Java4]. It is an optional package that can be implemented and used on most Java language-enabled devices. Special care has been taken to make sure it will run on CLDC/MIDP platforms, with upward scalability for CDC and J2SE platforms.

6.2.3 CDC and Foundation Profile, for High Level PDAs

The Connected Device Configuration (CDC) is the second configuration defined through the Java Community Process [Java5]. It provides a virtual machine and basic class libraries to support Java language applications on consumer electronic and embedded devices such as smart communicators, pagers, personal digital assistants (PDAs), and interactive, digital television set-top boxes. Typically, these devices run a 32-bit microprocessor/controller and have more than 2.0 MB of total memory for the storage of the virtual machine and libraries. CDC contains the CVM virtual machine. CVM is a full-featured virtual machine designed for devices needing the functionality of the Java 2 virtual machine feature set, but with a smaller footprint.

The proposed specification shouldn't have any dependencies on specific operating systems, CPUs, or I/O devices, but should at least meet the following requirements :

- 512K minimum ROM available
- 256K minimum RAM available
- Connectivity to some type of network.
- Supporting a complete implementation of the Java Virtual Machine as defined in the *Java Virtual Machine Specification, 2nd Edition.*

- User-interfaces with varying degrees of sophistication down to and including none may be supported by this configuration specification.
- TV set-top boxes, web enabled phones, and car entertainment/navigation systems are some, but not all, of the devices that may be supported by this configuration specification.

The J2ME Connected Device Configuration will define the minimum required complement of Java Technology components and API's for connected devices. Supported APIs, application life-cycle, security model, and code installation are the primary topics to be addressed by this specification.

For a complete J2ME application runtime environment, CDC uses the Foundation Profile, which is a set of Java™ APIs that provide functionality common to all target devices [Java6]. It can be combined with the J2ME™ Personal Profile specification, which provides a platform for the development of applications and higher-level APIs for consumer electronic devices. Personal Profile is highly suitable for the high-end PDA market and contains the full set of AWT APIs for support of a graphical user interface (GUI), including support for applets and Xlets, as well as a complete toolkit.

6.2.4 Virtual Machines for Java

One of the problems with Java for PDAs is that there is no standard virtual machine available for quick use. As shown in Table 6-1, many distributions were available when we led our investigation, but they were not all compatible. Even worse, most of them were not freely available, whereas Java is supposed to be a free development language!

Name of the JVM	**PDA's Operating System**	**Libraries bundled with the JVM**	**Comments**
IBM J9 VM	WindowsCE, PalmOS, and several other non-PDA OS	CLDC and MIDP	J9 was based on JDK 1.2.2 technology, targeted for embedded devices. IBM had also made available accompanying class libraries that bestow CLDC/MIDP and CDC compatibility. Note that J9 has now been replaced by the WebSphere Micro Environment [IBMWebSphere].
microJBlend kit for CE	Windows CE	CLDC and MIDP	See [MicroJBlend].
Kada VM	WindowsCE/ PocketPC, PalmOS	Personal Java, CLDC, CDC	See [KadaVM].
Jeode EVM	WindowsCE 2.12 and 3.0 and other non-PDA OS	Personal Java, Embedded Java	See [JeodeEVM]. Note that Jeode has now been replaced by Jbed™ Micro Edition [EsmertecJbed], since Insignia was bought by Esmertec [Esmertec].
CrEme	WindowsCE	unknown	See [CrEme].
ChaiVM	WindowsCE and other non-PDA OS	CLDC	ChaiVM is a subset of J2SE that is not certified, microChai is the version that is based on the CLDC-MIDP specification, but it is not certified either. See [ChaiVM]
J2SE, integrated in OS	SavaJe XE Operating System	Java 2, for JDBC, Jini, RMI, CORBA	Full, "big", Java for small devices, especially iPaq hardware from Compaq and other hardware with StrongARM processor [SavaJe].

Table 6-1 - Java Virtual Machines For PDAs

Finally, we should mention Waba, which is a language and platform based on Java, but not entirely compatible. More details on Waba can be found on [WabaSoft] and [SuperWaba].

6.2.5 Java Speech Solutions.

The Java™ Speech API allows Java applications to incorporate speech technology into their user interfaces. It defines a cross-platform API to support command and control recognizers, dictation systems and speech synthesizers. JSAPI 1.0. is not part of the JDK and Sun does not ship an implementation of JSAPI [JSAPI1]. Instead, they work with third party speech companies to encourage the availability of multiple implementations.

The JSAPI 2.0 is actually in development. It should allow developers to incorporate speech technology into user interfaces for Java technology enabled applets and applications. This API will provide access to speech technology on the growing number of devices running the mobile environment for Java 2 Platform, Micro Edition (J2ME), enhancing the usability and functionality of mobile devices [JSAPI2].

The Java Speech 1.0 API isn't officially supported by the J2ME platforms, hence compatibility tests with the following JSAPI implementations should be led.

FreeTTS 1.1.1 is a free speech synthesis system written entirely in the Java™ programming language. It is based upon Flite 1.1: a small run-time speech synthesis engine developed at Carnegie Mellon University. Flite is derived from the Festival Speech Synthesis System from the University of Edinburgh and the FestVox project from Carnegie Mellón University. It requires the JDK 1.4, and provides partial support for JSAPI 1.0 and the jsapi.jar [FreeTTS].

Festival is a free general multi-lingual (British English, American English, Spanish and Welsh) speech synthesis system developed by the Centre for Speech Technology Research at the University of Edinburgh [Festival]. It offers a full Text-To-Speech system with various APIs, as well an environment for development and research of speech synthesis techniques. It is written in C++ with a Scheme-based command interpreter for general control and provides a binding to the Java Speech API : initial support for talking to a Festival server from java is included from version 1.3.0 and initial JSAPI support is included from 1.4.0. At present the JSAPI talks to a Festival server elsewhere rather than as part of the Java process itself [Festival99A].

Festival runs on Suns (SunOS and Solaris), FreeBSD, Linux, SGIs, HPs and DEC Alphas and is portable to other Unix machines. Preliminary support is available for Windows 95 and NT, but Windows CE isn't mentioned. It requires a Java compiler and development kit (currently jdk1.1.7), as well as an implementation of the JSAPI framework (jsapi.jar). Currently a few classes from the Java Speech API Utilities from the Speech Group at Sun Microsystems are used.

The Free Schooling Freeplay is a free multimedia authoring program especially oriented to writing lessons [Freeplay]. It provides jsynth, which is a pure java Text-To-Speech library. The quality is not quite as good as current commercial TTS products but seems to be understandable. The library is lightweight, with only 59KB for the version without a dictionary and 1.4MB for the version with the Carnegie Mellon pronunciation dictionary. The dictionary somewhat improves the quality. The interface to the library is a very minimal subset of the JSAPI (Java Speech API). It implements plain text speech and the 'wordStarted' callback of SpeakableListener. There is a test program in the source code zip which shows how the library can be used.

With an implementation based on IBM's ViaVoice product, IBM's **Speech for Java** supports continuous dictation, command and control and speech synthesis. It supports all the European language versions of ViaVoice (US & UK English, French, German, Italian and Spanish) plus Japanese [SpeechForJava]. It requires JDK 1.1.7 or later or JDK 1.2 on Windows 95 with 32MB, or Windows NT with 48MB. Both platforms also require an installation of IBM's ViaVoice 98. A 90-day trial version can be downloaded from [SpeechForJava].

The Cloud Garden is an implementation for use with any recognition/TTS speech engine compliant with Microsoft's SAPI5 (with SAPI4 support for TTS engines only) [CloudGarden]. An additional package allows redirection of audio data to/from Files, Lines and remote clients (using the javax.sound.sampled package). Some examples demonstrate its use in applets in NetscapeTM and IE browsers. It requires JDKTM 1.1 or better, Windows 98, Me, 2000 or NT, and any SAPI 5.1, 5.0 or 4.0 compliant speech engine (some of which can be downloaded from Microsoft's web site).

Elan Speech Cube is a Multilingual, multichannel, cross-operating system Text-To-Speech software component for client-server architecture [Elan]. Speech Cube is available with 2 TTS technologies (Elan Tempo : diphone concatenation and Elan Sayso : unit selection), covering 11 languages. Speech Cube native Java client supports JSAPI/JSML. It requires JDK 1.3 or later on Windows NT/2000/XP, Linux or Solaris 2.7/2.8, Speech Cube V4.2 and higher.

Lernout & Hauspie's TTS for Java Speech API implementations is based upon the ASR1600 and TTS3000 engines, which support command and control and speech synthesis [L&H]. It supports 10 different voices and associated whispering voices for the English language, and provides control for pitch, pitch range, speaking rate, and volume. It requires Sun Solaris OS version 2.4 or later, and JDK 1.1.5. Sun Swing package for graphical Type-n-Talk demo.

Finally, **Conversa Web 3.0** is a voice-enabled Web browser that provides a range of facilities for voice-navigation of the web by speech recognition and Text-To-Speech [Conversa]. The developers of Conversa Web chose to write a JSAPI implementation for the speech support. It requires Windows 95/98 or NT 4.0 running on Intel Pentium 166 MHz processor or faster (or equivalent). A minimum of 32 MB RAM is necessary, but 64 MB is recommended. Microsoft Internet Explorer 4.0 or higher is required.

As we can see, Text-To-Speech features seems to be quite limited. The main efforts are targeted for desktop computers, and we have no guarantee on their compatibility with PDA devices.

6.3 C++ for PDAs.

6.3.1 eMbedded Visual C++

The Microsoft eMbedded Visual Tools 3.0 deliver a free and complete desktop development environment for creating applications and system components for Windows powered devices, including the Pocket PC and Handheld PC [Vstudio]. The eMbedded Visual Tools particularly include eMbedded Visual C++ as well as SDKs for the Pocket PC, Palm-size PC, and Handheld PC. In our case, we should point out that the Pocket PC SDK allows to simulate an x86 PDA environment on a desktop computer. This version is standalone and does not require Visual Studio. Another point to mention is the fact that, C++ enables to compile executable files immediately usable, and doesn't require any virtual machine, unlike Java.

6.3.2 Sound and Speech Libraries

Microsoft eMbedded Visual C++ enables playing sounds on Pocket PCs. Functions such as PlaySound, MessageBeep, waveOutWrite and other waveOut functions are indeed available [Philippov2001]. Even though WinCE supports the powerful DirectSound library, Microsoft elected to leave it out of the Pocket PC platform, and preferred referring to the old MMIO API. But to complicate the problem, Pocket PC does not include the entire MMIO API, and the hardware support for the functions that are included are somehow limited. Nevertheless, using waveOut functions, it is possible to create simple audio player [Wolraich2001].

Regarding Text-To-Speech libraries, we have already mentioned **Festival**, whose free distribution include full C++ source for modules, SIOD interpreter, Scheme library, but also low level C++ library. It provides the Edinburgh Speech Tools Library, which is a collection of C++ class, functions and related programs for manipulating the sorts of objects used in speech processing. It includes support for reading and writing waveforms and parameter files in various formats and converting between them.

It also includes support for linguistic type objects and support for various label files and ngrams (with smoothing), as well as a number of programs in addition to the library. An intonation library which includes a pitch tracker, smoother and labelling system, a classification and regression tree building program called wagon. Also there is growing support for various speech recognition classes such as decoders and HMMs. The documentation for the Edinburg Speech Tools library is available at [Festival99B]. Unfortunately, it requires Visual C++ 6.0 to operate.

We can't talk of TTS without mentioning the **Mbrola Project**. Central to the project is MBROLA, a speech synthesizer based on the concatenation of diphones [MBROLA]. It takes a list of phonemes as input, together with prosodic information (duration of phonemes and a piecewise linear description of pitch), and produces speech samples on 16 bits (linear), at the sampling frequency of the diphone database used. Mbrola is therefore not a Text-To-Speech synthesizer, since it does not accept raw text as input. This synthesizer is provided for free, for non commercial, non military applications only. Even if Mbrola isn't a TTS synthesizer, C++ programs based on it are trying to make TTS available for C++ programmers, like TexTalk or Freespeech [MBROLA].

6.4 Conclusion

As we can see, both Java and C++ are limited from the musical and TTS aspect, and therefore seem equivalent when it comes to consider what is feasible. That's why we finally choose the eMbedded C++ solution. As we had not much time to get started, this choice seemed more reasonable, given that it offered a ready-to-use and easy-to-test implementation environment. The visual environment and the built-in emulator were crucial points that we couldn't ignore.

7 Refining And Planning The Implementation Of Our Generic PDA Based Museum Guide

7.1 Introduction

Now that we have surveyed the different possibilities and constraints, it is time to refine our objectives and to define more precisely the features of our application. The idea is not explain in detail each step of the implementation, but to give a general idea of how we went through it.

7.2 Requirements

7.2.1 Features and Limitations Of The Application

Our idea is to keep as much as possible the tasks and the objectives stated in Chapter 4. First of all, the user should be able to load an environment file that will dynamically build the museum environment in which he will evolve. Once the environment is set, the user should be able to visualize an overview map, from which he should be able to see all the rooms of the environment. From the overview map, the user should be able to switch to a room map by selecting it on the overview map. Note that we consider only one floor level.

Once in a room map, all the art objects located in the room should be displayed. By selecting an objects, the user should access a basic information screen : name and picture of the object, author, style, date, country, material. Text-To-Speech will not be supported by the application, but we will enable a sound file to be associated with each art object. This sound file will automatically be played when accessing the information screen. Therefore, we could associate a sound file with vocal content to an object. From the information screen, the user should be able to switch back to the room map, and from the room map to the overview map. An immediate limitation appears regarding the type of the art objects : we are working as if the art objects were all "grounded". This means that our application will rather suit modern art museums than classical museums with paintings. Indeed, in our application, we cannot have an object vertically located under another (which is often the case of paintings hung on a wall).

Because of the lack of time and as our application remains a prototype, several features won't be available. In particular, real localisation will not be ensured. Instead, an infrared simulation will be offered, which will allow users to explicitly mention the reception of an infrared signal. Another limitation will be that the To See List will not be dynamically editable, although the implementation should give the tools to allow it. Instead, the tour will have to be restarted. This implies that it will not be possible to ask for similar types of work from the information screen of an object. These features will be rediscussed when we will envisage the perspectives of our work, and more practical limitations (from the implementation point of view) will be exposed in the following chapter.

7.2.2 Class Diagram

From the task analysis of Chapter 4 and the refinements we just made, we can finally deduce the class diagram for the main entities that our application will have to manipulate (Figure 7-1).

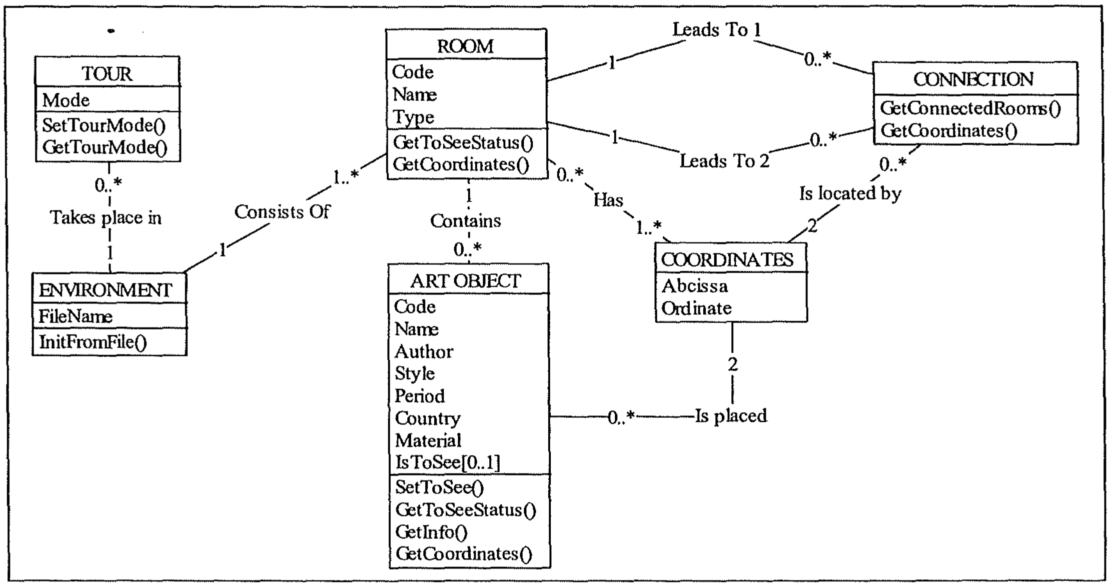


Figure 7-1 - The Class Diagram For Our Application

The relationships between the entities are as follow :

- The **Tour** is a visit of a museum environment. It is characterized by the mode the user selects to visit the museum, that is free or customized.
- The **Environment** is the museum environment itself. All the information necessary to build a museum environment is contained in a file that can be loaded. It will allow the application to build all the rooms of the environment.
- The **Room** is a subset of the environment. It contains **Art Objects** and **Connections** to other rooms.
- An Art Object can be part of the objects that a user wants to see in his customized tour mode. In this case, the **IsToSee** flag will be set to true, and the room containing the art object will also be considered as to be seen.
- Finally, each room, connection between rooms and art object has coordinates that allow to locate them.

7.2.3 Non functionnal requirements

As we mentioned earlier, the future users will probably be neophytes of mobile technologies. Thus, the interface of the application should be as intuitive as possible. To ease interaction, the number of steps necessary to perform an action should be minimal.

7.3 Logical Conception

7.3.1 Abstract Interface

The ConcurTaskTree Notation allows to simultaneously express the task structures and hierarchy, instead of using a Task Tree or a sequencement diagram That's why, on basis of our new requirements, we decided to refine the Tree given in Figure 4-4 in order to obtain the one given in Figure 7-2 and detailed in Appendix A, section A.2. It enables us to have a first idea of what the application interface will look like.

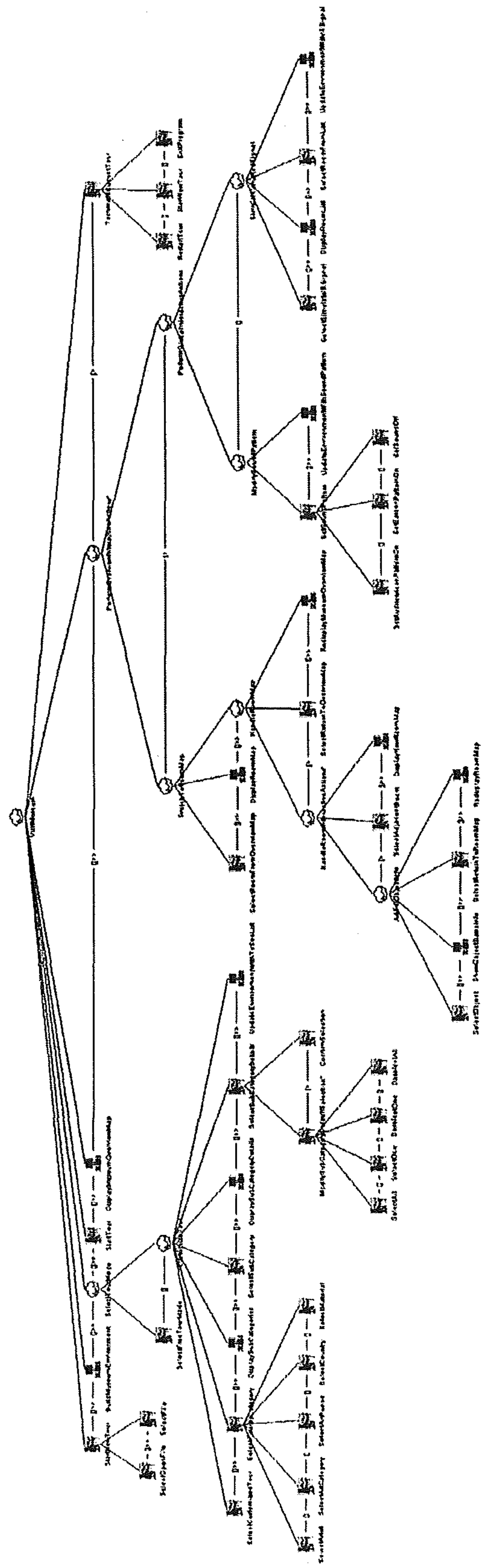


Figure 7-2 - The Refined ConcurTaskTree Of Our Application

7.3.2 Data Storage

Like we stated, all the information necessary to build an environment will be put in a file. In order to ease the specification of the environment, we decided to use our own specification format. All the sound and picture files will be loaded directly from the device, using special naming conventions. All these specification rules will be exposed in the following chapter.

7.3.3 Logical Components Definition

Before starting the physical conception phase, it is important to summarize the definition of the logical components we will manipulate within our application. These components and the main features they should provide are given in Figure 7-3.

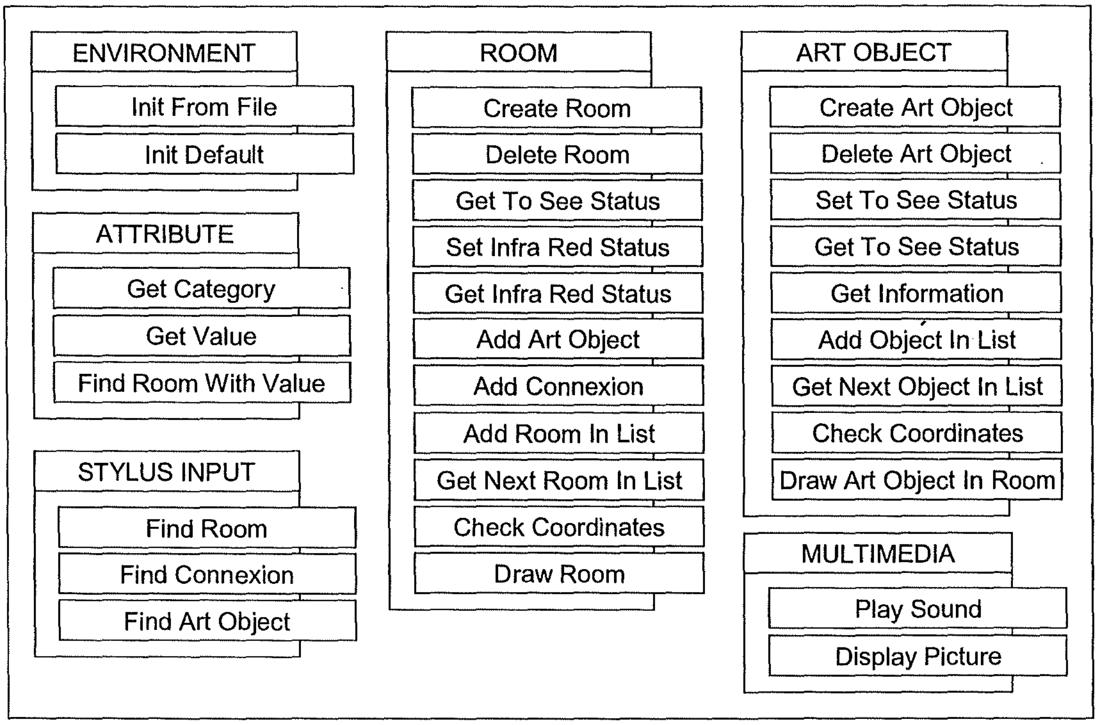


Figure 7-3 – Logical Components Definition For Our Application

7.4 Physical Conception

This part of the dissertation presents the structure of our application, in order to understand more easily the code listings available in Appendix E. As we chose eMbedded C++ to implement the application, we decided to organize our classes as shown in Figure 7-4. Note that the arrows indicate inclusion. The structure of the classes were highly influenced by the Windows programming style described in [Boling2001].

The **Ravi-guide** class is the heart and soul of our application. It manages all the aspects of initialization of the application, initialization of an instance of the application and termination of the program. As Windows (and in particular Windows CE) works by exchanging and dealing with messages and events, this class translates, dispatches and handles all the messages generated and received by the application.

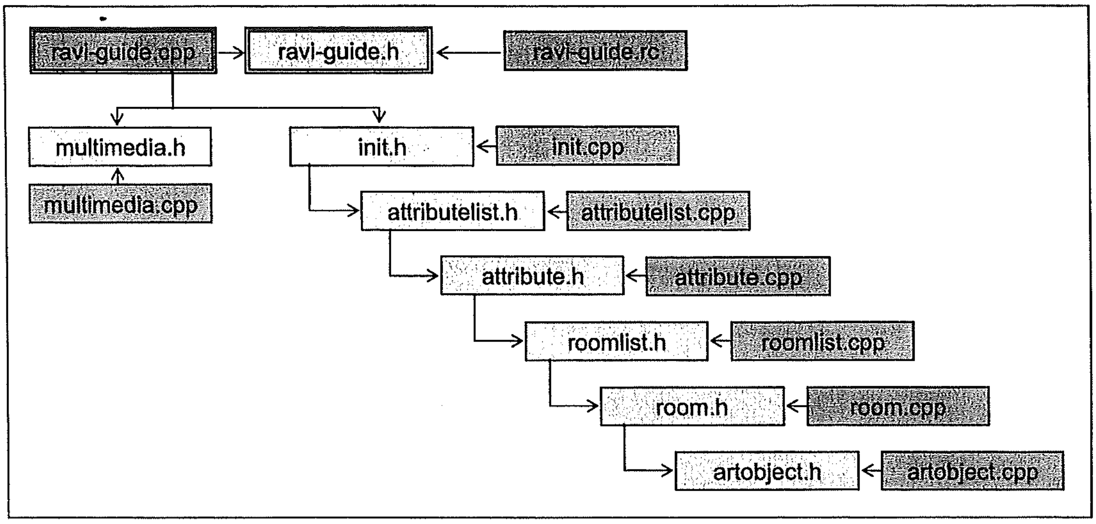


Figure 7-4 – Class / File Hierarchy of Our Application

It especially handles the application window's painting, that is everything that is displayed on the screen. It manages the whole user interface, as well as the navigation between the menus. In particular, it manages the stylus input (which will be discussed more in details in the next chapter). Finally, it handles the infrared simulation part.

The **Multimedia** class provides tools to play sound files and to display images available on the PDA's hard disk. These tools are used by various methods of the Main class, especially the painting one.

The **Init** class provides the method to initialize a museum environment by using a specification input file available on the PDA's hard disk. It parses the file and initializes the global variables of the Main class if the file is correctly formatted.

The **AttributeList** class provides a list structure to handle the **Attribute** class. The combination of both classes allows to build the list of all the existing values for each art object attribute category (name, author, style, period, country, material), and links each pair of "attribute category" and "attribute value" to a set of rooms containing pertinent objects. This is a key mechanism for building the menus where the user wants to customize his tour in the museum.

The **RoomList** class provides a list structure to handle the **Room** class. All the rooms of the environment are stored in a List structure. Each room contains the information necessary to be drawn on the PDA screen, as well as links to the contained art objects and the adjacent rooms. It also confirms if a given pair of coordinates (x,y) is a connection to an adjacent room, an area of the room, or an external point.

The **ArtObject** class is a class that allows a list structure. The class should had been decomposed into ArtObjectList and ArtObject classes, but changing the structure would have been too time-demanding. The class contains all the information on a given object (name, author, style, period, country, material), as well as a flag for the To See status. The header file contains all the constants definition, in particular the restraining values for the specification input file.

Finally, the resource file **Ravi-guide.rc** provides resources for handling the dialog boxes of the application, and the links to other resources such as the default sounds and pictures used in the prototype.

7.5 Conclusion

After refining the requirements of our objectives, we had to formalize them to conceptually conceive the application. After translating this to a physical hierarchy to plan and clarify the structure of the application, we were able to start the implementation phase. The code listings that resulted are available in Appendix E. It is interesting to spot a few problems we encountered during this implementation period, and how we overcame them.

8 Implementation And Usability Concerns

8.1 Issues

We will now analyse a few implementation problems and issues that we had to face during the implementation period. We will first discuss the application customization problems, then the stylus input mechanism, and we will finally mention a few limitations of our application.

8.2 Customizing The Environment

8.2.1 Format of the specification file

Like our program needs to be generic, we allow users to specify the environment of the museum by writing a specially formatted text file, using the following structure (Table 8-1). Note that a line beginning by // will be ignored : this allows the user to insert comment lines in his specification file.

```
#newroom
#code=<Code of the Room>
#name=<Name of the Room>
#type=<Type of the Room>
#numberofcoordinates=<Ncrd>
#coordinates=
<x1>,<y1>
...
<xNcrd>,<yNcrd>

#numberofconnexions=<Ncnx>

  #connexioncode=<Code of the 1st Connected Room>
  #coordinates=
  <x1>,<y1>
  <x2>,<y2>

  ...

  #connexioncode=<Code of the Ncnxth Connected Room>
  #coordinates=
  <x1>,<y1>
  <x2>,<y2>

#numberofartobjects=<Nao>

  #code=    <Code of the 1st Object>
  #name=    <Name of the 1st Object>
  #attrib1= <Author of the 1st Object>
  #attrib2= <Style of the 1st Object>
  #attrib3= <Period of the 1st Object>
  #attrib4= <Country of the 1st Object>
  #attrib5= <Material of the 1st Object>
  #coordinates=
  <x1>,<y1>
  <x2>,<y2>

  ...
```

```
#code= <Code of the Nao^th Object>
#name= <Name of the Nao^th Object>
#attrib1= <Author of the Nao^th Object>
#attrib2= <Style of the Nao^th Object>
#attrib3= <Period of the Nao^th Object>
#attrib4= <Country of the Nao^th Object>
#attrib5= <Material of the Nao^th Object>
#coordinates=
<x1>,<y1>
<x2>,<y2>
```

Table 8-1 - Format of the specification file

Note that the art object attributes that we suggest (author, style, period, country and material) could be replaced by any other type of attribute. Still, the semantic of the attribute should still be defined *in the code* of the application, more precisely in the definitions of the header file of the ArtObject class defined in the previous chapter (Table 8-2). The detail of this class can be found in Appendix E.

```
#define ARTOBJECT_ATTRIBUTE1      "Artist"
#define ARTOBJECT_ATTRIBUTE2      "Style"
#define ARTOBJECT_ATTRIBUTE3      "Period"
#define ARTOBJECT_ATTRIBUTE4      "Country"
#define ARTOBJECT_ATTRIBUTE5      "Material"
```

Table 8-2 - Excerpt of the definitions of the Art Object class header

We must mention the fact that the unit of the coordinates is somehow abstract. This means that the only constraint is that all the coordinates must ensure the correct proportion of all the rooms and objects in the environment.

8.2.2 Using Customized Images And Sounds

The specification file should be placed in a folder containing two sub directories named "images" and "sounds". Typically, the specification folder should be put in the MyDocument directory of the PDA in order to be accessed from the Open File menu of the application (see next chapter). The images directory should contain 256 colours bitmap pictures, with an ideal size of 200x125 pixels, and named by the code of the object they are representing. When generating the object information screen, if no corresponding bitmap is found in the folder, a default image will be displayed. The sound directory should contain all the wave files related to the objects, and as for the pictures, these files should be named by the code of the related object. When generating the object information screen, if no corresponding wave file is found in the folder, no audio comment will be available for this object.

So, if in the specification file there is an artobject declared with the code RAVI2411, the program will try to access RAVI2411.bmp from the images subfolder, and RAVI2411.wav from the sounds subfolder of the specification folder.

8.2.3 Determining The Coordinates Of The Environment

The absolute origin

All the coordinates have to be given referring to an absolute origin, as shown in Figure 8-1. This origin should be chosen so that the coordinates of all the rooms of the environment are positive or null. At least one of these coordinates will have its abscissa null, and another one its ordinate null (it can be the same point in both cases).

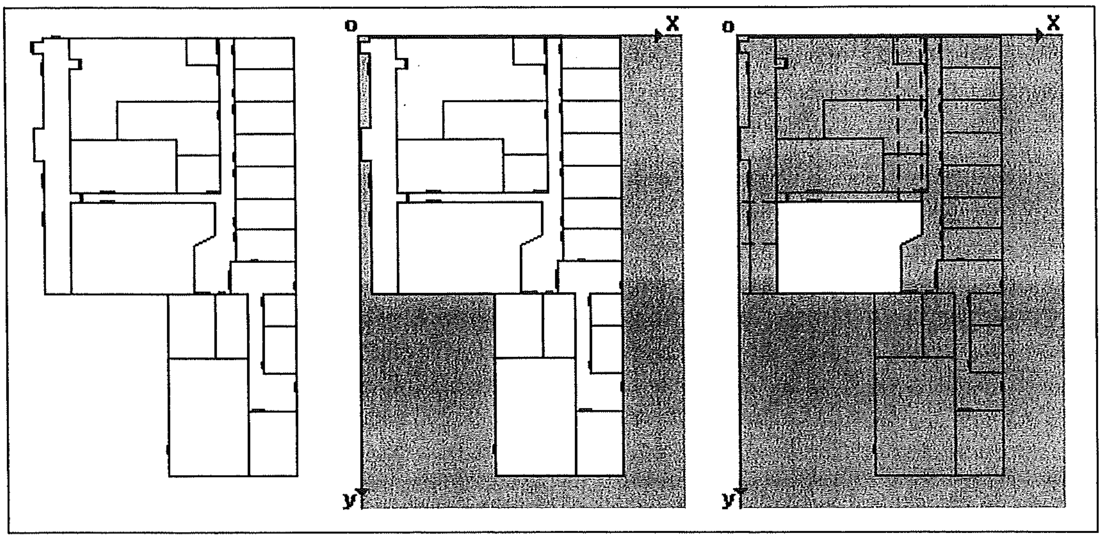


Figure 8-1 - Absolute origin of an environment

Choosing the order to enumerate the coordinates

The coordinates must be enumerated in a "circular way", because the drawing method used to draw a room as a polygon basically draws a succession of lines point to point. As shown in Figure 8-2, the enumeration can begin with any point, but once started, you should always examine the left (or right) vertex of the polygon.

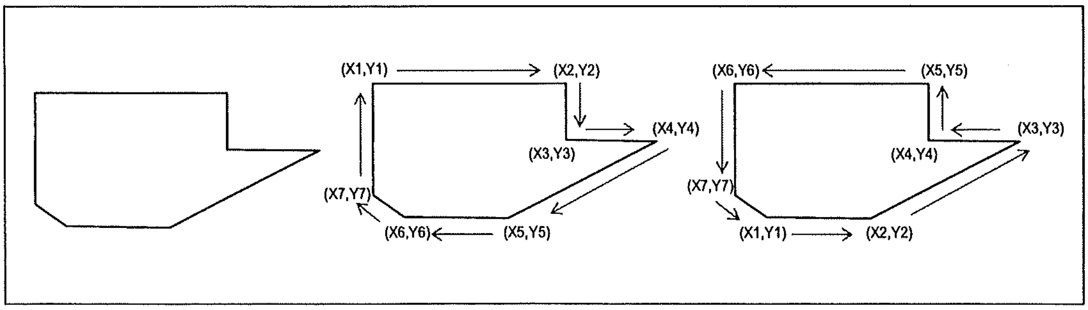


Figure 8-2 - Circular Enumeration of the Coordinates of a Room

8.2.4 Example : the Room 19 specification

An example should allow to understand these constraints more easily. Let us consider a rectangle room called "Room 19", connected to a room whose code is "codeHall2", and containing 2 objects called Art Object A and Art Object B (Figure 8-3). The specification code for this room would therefore look like the one of Table 8-3.

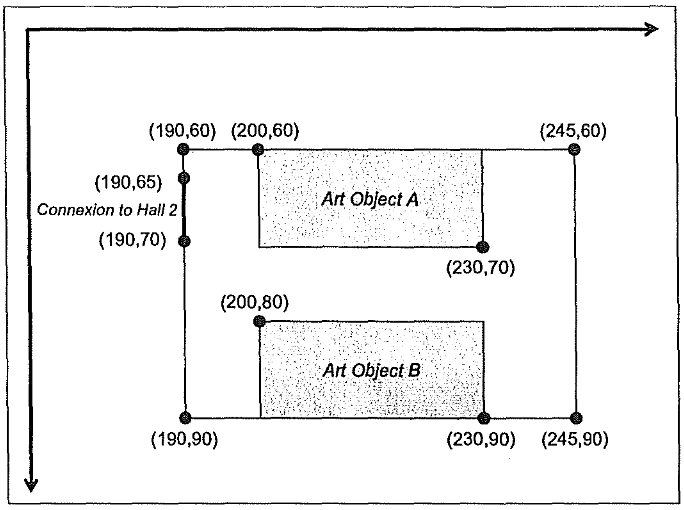


Figure 8-3 - The Room 19

```
#newroom
#code= code19
#name= Room 19
#type= Exhibit Room
#numberofcoordinates= 4
#coordinates=
190,90
190,60
245,60
245,90

#numberofconnexions= 1

#connexioncode=codeHall2
#coordinates=
190,65
190,70

#numberofartobjects=2

#code=    code19A
#name=    Art Object A
#attrib1= Author A
#attrib2= Style A
#attrib3= 2002
#attrib4= Country A
#attrib5= Material A
#coordinates=
200,60
230,70

#code=    code19B
#name=    Art Object B
#attrib1= Author B
#attrib2= Style B
#attrib3= 2002
#attrib4= Country B
#attrib5= Material B
#coordinates=
200,80
230,90
```

Table 8-3 - The Room 19 specification example

8.3 Determining The Target Of The Stylus Input

8.3.1 The Problem

Determining if a point on the PDA screen is in a room or an object is a crucial issue for the interaction between the user and our application : the level switching is done by taping on a room. This problem can be considered as determining if a given point is inside a polygon or not. The algorithm we use is an adaptation of the one presented in [Faivre2001].

8.3.2 A Few Definitions

Before looking at the algorithm, let us introduce a few definitions, as given in [Borowski&Borwein91].

A *polygon* is a "closed plane figure bounded by three or more straight line segments that terminate in pairs at the same number of vertices, and do not intersect other than at their vertices. The sum of the interior angles is (n-3)*180° where n is the number of sides ; the sum of exterior angles is always 360°".

A *convex polygon* is a polygon "having no interior angle greater than 180°, so that all lines joining any pair of points on the boundary of the figure lie wholly inside it".

These definitions are crucial, because they are implicitly used in our definition of a room : we consider all the rooms of an environment as a convex polygon.

8.3.3 Establishing the algorithm

Let (X,Y) be a point and $\{(X_1,Y_1),\ldots,(X_n,Y_n)\}$ be a connected polygon. A first graphical observation shows us that a vertical semi-strait line passing through a point inside the polygon intersects this one in another point. Likewise, a vertical strait line passing trough a point outside a convex polygon intersects the boundary of this polygon on zero or two points (Figure 8-4).

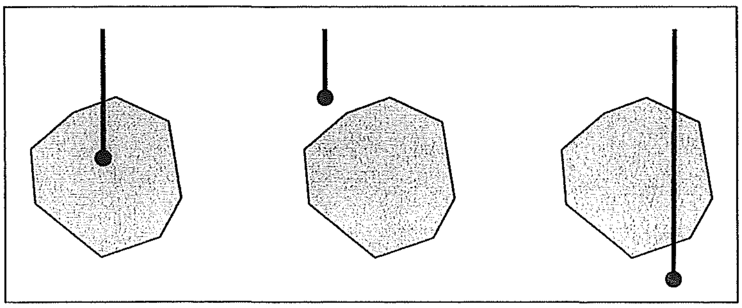


Figure 8-4 - Intersections between a convex polygon and a semi-strait line

The same method applied to a connected but non-convex polygon allow us to state a more general rule : a point is inside (respectively outside) a polygon if and only if the vertical semi-strait line passing through the point intersects the boundary of the polygon on an odd (respectively even) number of points (Figure 8-5).

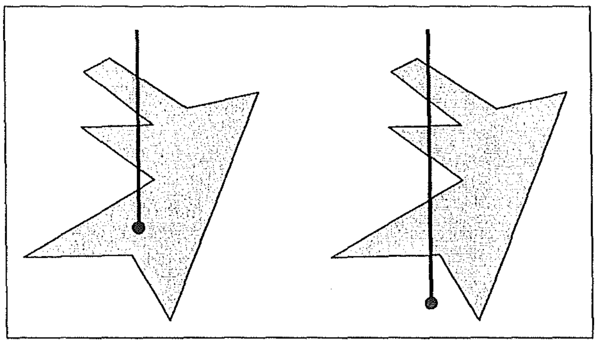


Figure 8-5 - Intersections between a non convex polygon and a semi-strait line

We should notice that it can become difficult to enumerate the number of points intersecting the boundary of the polygon if one of the sides of the polygon is vertical. Furthermore, the rule we stated above fails when there is at least one double intersection, that is an intersection point that coincide with one vertex of the polygon (Figure 8-6). As a consequence, we have to choose the slope of the strait line different from each slope of the sides of the polygon.

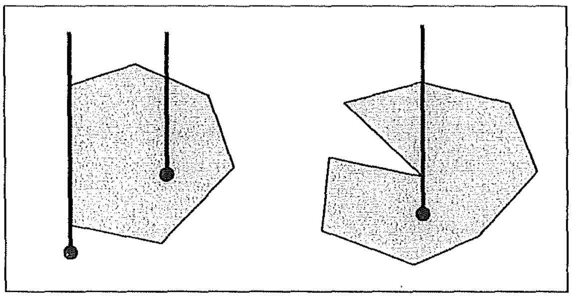


Figure 8-6 – Polygon configurations for which the rules fails

Finally, we can discard the case of an intersection of the point with one side of the polygon to distinguish a strict intersection from an intersection with the boundary of the polygon. All these properties arise from intuitive arguments and allow us to build an efficient algorithm based on visual observations. However, this algorithm isn't optimized for computing treatment purpose.

8.3.4 The algorithm itself

Let (X,Y) be a point and $\{(X_1,Y_1), \ldots, (X_n,Y_n)\}$ be a connected polygon with n distinct summits.
If (X,Y) belongs to one of the segments $[(X_i,Y_i),(X_{i+1},Y_{i+1})]$, conclude.
Otherwise, calculate the n-1 slopes of the segments $[(X_i,Y_i),(X_{i+1},Y_{i+1})]$ and place in an ordered list (let us say an increasing one).
Then calculate the n slopes of the segments $[(X,Y),(X_i,Y_i)]$, and add them to the list.
Let us define the slope p as the arithmetic mean of the first two distinct slopes of the ordered list.
The segment S, passing through (X,Y) with the slope p, and for which the other extremity has an abscissa equal to the maximum of $\{X_1, \ldots, X_n\}$, meets the conditions necessary to continue.
By using an iteration, check if S intersects with each segment of the polygon, and increment a counter (initially set to null), if it is the case.
Finally, testing the parity value of the counter allows to conclude.

Table 8-4 - The algorithm used to determine a given point is inside a polygon or not

The implementation of the algorithm can be found in the room class available in Appendix E.

8.4 Limitations

A few aspects had to be left aside during the implementation phase, which led to a few limitations of our generic application.

8.4.1 The Connections Between Rooms

In the implementation, we assumed that all the connections between rooms would be either horizontal or vertical. When drawn on a room map, the connection is represented by a horizontal/vertical enlarged rectangle based only on two coordinates (Figure 8-7). This simplification was mainly motivated by the lack of time. It could be overcome by calculating the slope of the connection, and by using the built-in Polygon method of eMbedded C++ instead of the Rectangle method in order to draw the connection (Table 8-5). Indeed, the Polygon method allows to draw any kind of polygon, while the Rectangle method only allows horizontal/vertical rectangles.

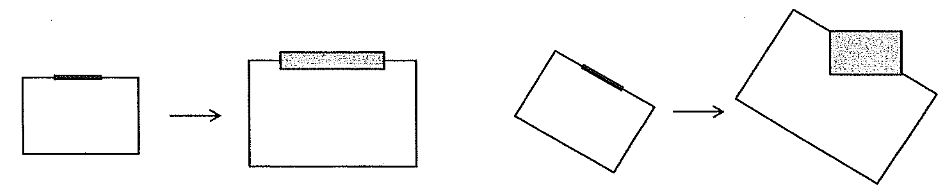


Figure 8-7 - The Connection Problem with the Rectangle Method

```
BOOL Polygon (HDC hdc, const POINT *lpPoints, int nCount );
BOOL Rectangle  (HDC hdc, int nLeftRect, int nTopRect, int nRightRect, int
nBottomRect );
```

Table 8-5 - The Polygon and the Rectangle methods in eMbedded C++

Another issue concerns the number of connections possible. As for each room structure, the connections information are placed in arrays (Table 8-6), and as we cannot dynamically change the size of an array in C++, we had to specify a maximum size for those arrays (Table 8-7). Hence, the maximum number of connections is also limited.

```
char   code[MAX_STRINGSIZE];
char   name[MAX_STRINGSIZE];
char   type[MAX_STRINGSIZE];
int    numberOfCoordinates;
POINT  coordinates[MAX_COORDINATES;
int    numberOfAdjacentRooms;
char   adjacentRoomsCode[MAX_CONNEXIONS][MAX_STRINGSIZE];
POINT  connexionsA[MAX_CONNEXIONS];
POINT  connexionsB[MAX_CONNEXIONS];
ArtObject * firstObjectInRoom;
```

Table 8-6 - Excerpt of the Room class header

```
#define MAX_CONNEXIONS     50
```

Table 8-7 - Definition of the maximum of connections allowed in the ArtObject class header

8.4.2 The Art Objects

The problems we had with the art objects are somehow similar to the ones we had with the room connections. This time, we also used two coordinates to locate the art objects in a room : the upper left and the lower right coordinates, assuming all the objects would be represented by a horizontal/vertical rectangle (Figure 8-8). Once again, the problem could be overcome by using the built-in Polygon method of eMbedded C++ instead of the Rectangle method in order to draw the art object in the room map.

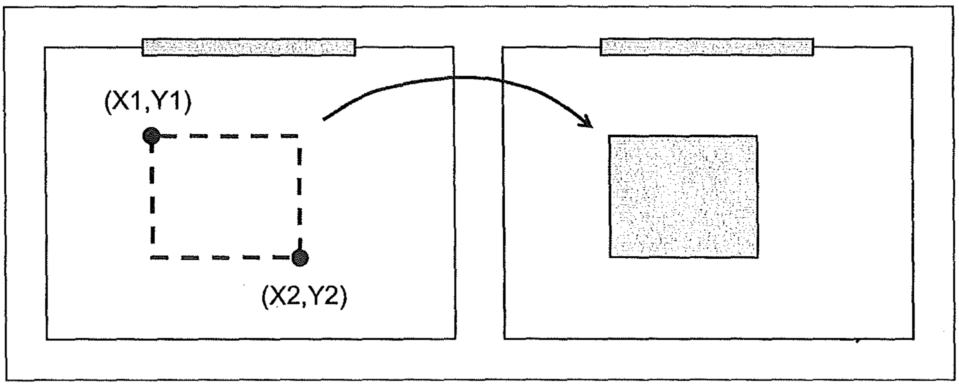


Figure 8-8 – Representation of an Art Object in the Room Map

The other issue concerns the number of art objects possible in a room. In one of the methods we used to handle the dialog boxes of our application (SelectSubCatDetailsDlgProc, in the main class), we had to use an array whose size depended on the maximum of art objects available. Once again, we therefore had to specify a maximum size for those arrays. Hence, the maximum number of artobjects is also limited.

8.4.3 The "Inner Rooms"

Let us consider the environment presented in Figure 8-9. There is a corridor that we can see as a room containing other rooms, which induces two main problems. First, the specification file format doesn't allow to specify "non plain" room (Figure 8-10). Indeed, there is no way to specify that there is a portion of the room that should be ignored, and if we only consider it as a "plain" room, we lose all the information about the connections to the inner rooms.

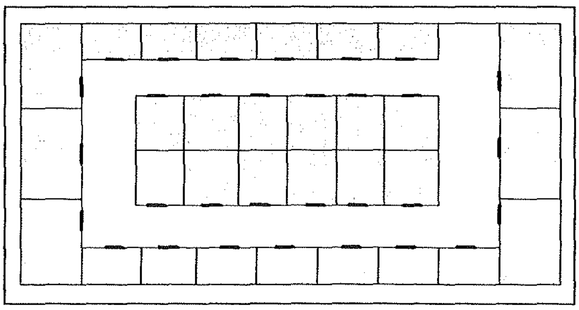


Figure 8-9 - The Inner Room Problem

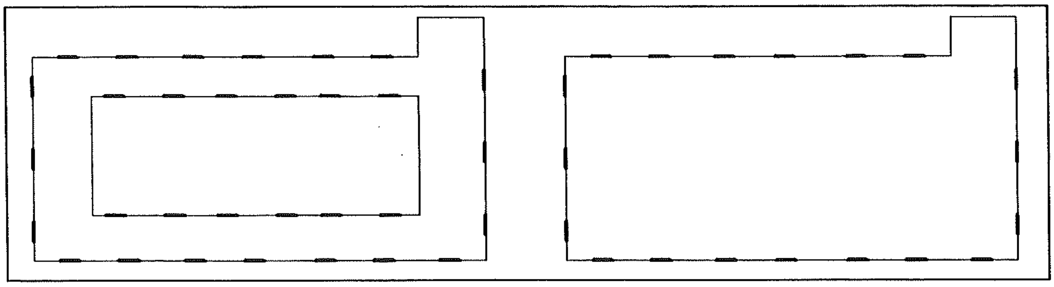


Figure 8-10 - Specification limits for the inner room problem

However, let us admit that we could specify the corridor as a "plain" room without losing any connection information. The second problem is that until now we only considered rooms as an entity containing only art objects, and this is the logic we used for the implementation of our application :

- the rooms of the environment are inserted in a list;
- the order of insertion in the list (which depends on the order of specification in the specification file) determines the order in which the rooms will be drawn on the overview map, and the order in which we will test the stylus input from this overview map.

Therefore, if the corridor is specified in the specification file of the environment before its "inner rooms", it will be drawn first on the overview map, and the inner rooms will be drawn over it. From the visual point of view, there will be no problem, all the rooms will be visible. Yet, when the user will tap the screen with the stylus to select one of the inner rooms of the corridor, the localisation of the stylus input will first be tested with the corridor : as the coordinates will be inside the corridor, the input point will be recognized as part of it and the room map of the corridor will be displayed instead of the one of the intended inner room.

Specifying the inner rooms before the corridor will just have the opposite effect : the inner rooms will be hidden on the screen by the corridor, while the stylus input will still detect these rooms. That's why, unless rethinking the whole semantic of a room and the way to represent it, the alternative solution is to specify the corridor with an imaginary wall (Figure 8-11).

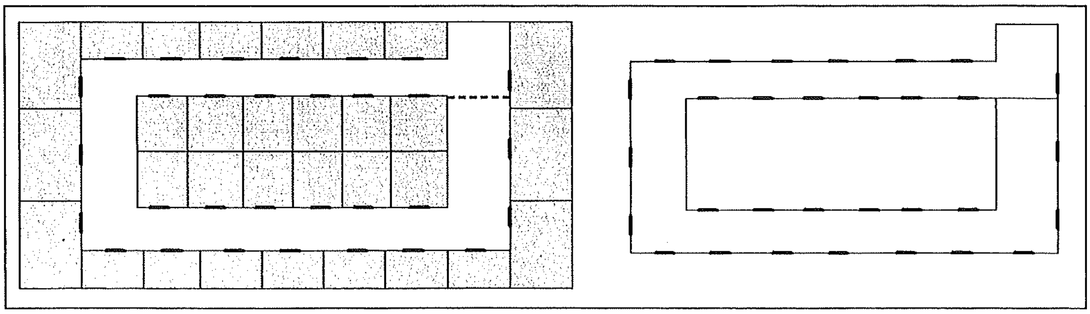


Figure 8-11 - The imaginary wall alternative

This solution is not very aesthetic, but at least it allows to put aside the inner room problem. The choice of the location of the imaginary wall is not really crucial. Nevertheless, it should be chosen in order to limit the annoyance of the users.

8.5 Conclusion

As we have seen, implementing our guiding application compelled us to analyse parallel problems that could have seemed somehow anodyne. Nevertheless, now that we analyzed a little more closely some of these implementation issues, it is finally time to take a look at what our prototype looks like.

9 Rating the interface

9.1 Introduction

In this chapter, we will finally present our prototype and we will try to rate it by using an evaluation questionnaire. By analyzing the answers we collected, we will draw some teachings and envisage possible improvements.

9.2 Presentation Of The Interface

In our application, the user can start a new tour by using the New Tour command from the File menu. He will be able to choose a specification file from the hard disk of the PDA (Figure 9-1).

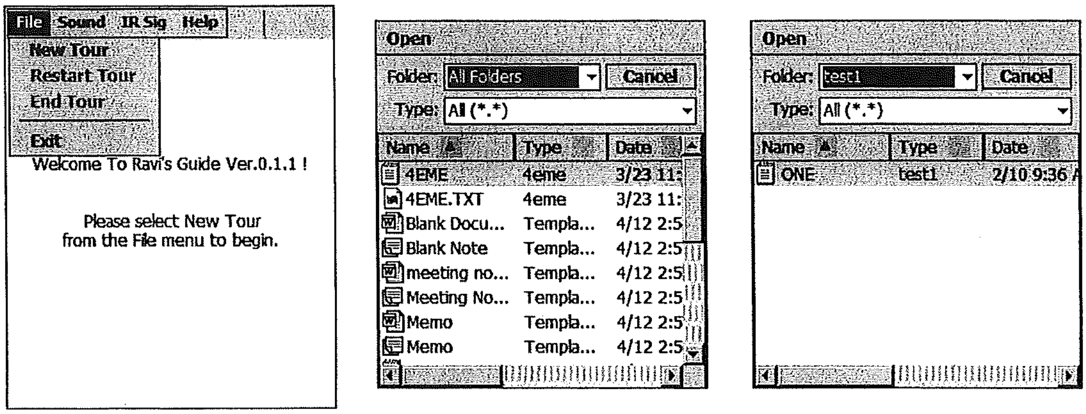


Figure 9-1 - Start Tour Screenshots

If he chooses the Free Tour Mode, he accesses directly the overview map of the museum. If he chooses to customize his tour, he can select the objects he wants to see by choosing a category, a sub category and finally the details of the sub category (Figure 9-2). The Customized Tour Mode will highlight the rooms and objects that the user has selected.

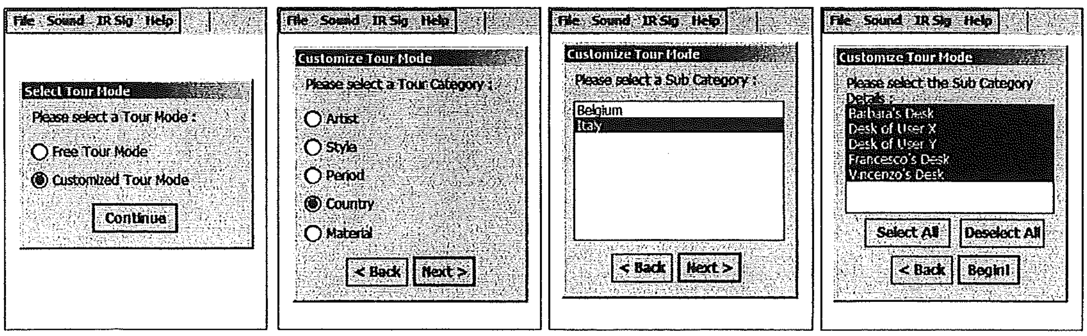


Figure 9-2 - Customize Tour Mode Screenshots

On the overview map, the map of the museum is displayed. If the customized tour mode is activated, the rooms containing objects selected by the user will be highlighted. By taping on a room with the stylus, the user will access the room map. On the room map, the map of the room is displayed, as well as its objects and the connections to adjacent rooms. Once again, if the customized tour mode is on, the selected objects and the connections to selected rooms will be highlighted (Figure 9-3).

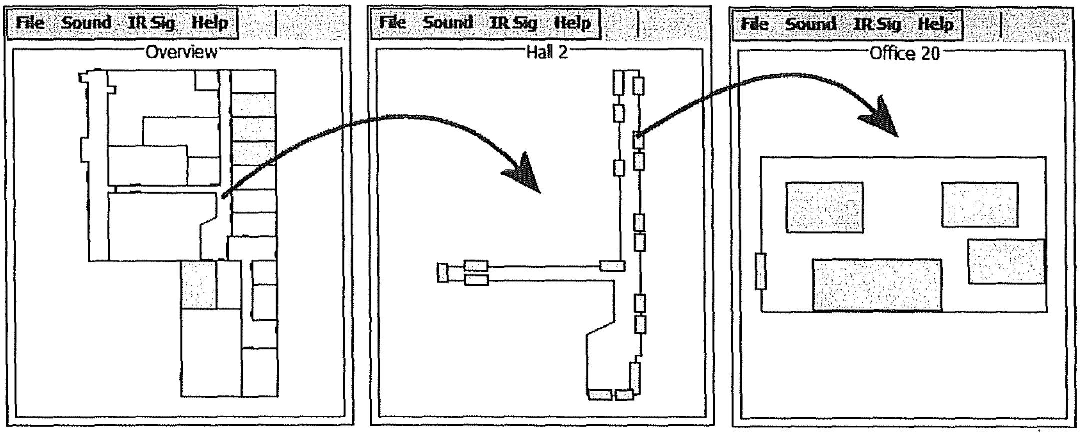


Figure 9-3 - Overview and Room Map

By taping on an object, the user will access the Object Information screen. By taping on a room connection, he will switch to the connected room. By taping outside the room, he will return to the overview map. In the Figure 9-3, the Hall 2 is connected to the Office 20 which contains some objects selected by the user. By clicking on the highlighted connection, the user will access the Office 20 map. The Object Information screen contains a picture of the object, as well as basic information. The picture will be displayed only if it follows the conventions given in the previous chapter. Otherwise, a default image will be displayed (Figure 9-4). By taping on the screen, the user will return to the room map.

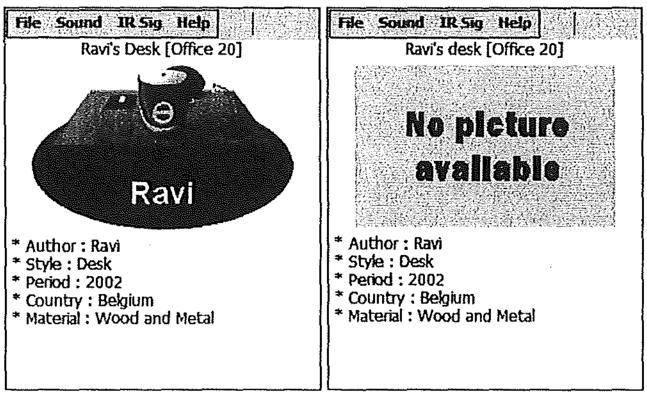


Figure 9-4 - The Art Object Information Screen

At the beginning of the implementation, we considered implementing an InfraRed module. Alas, we didn't have infrared beacons at disposal, which was quite troublesome. Still, to simulate the behaviour of our application in "infrared conditions", we implemented an InfraRed simulation as illustrated in Figure 9-5.

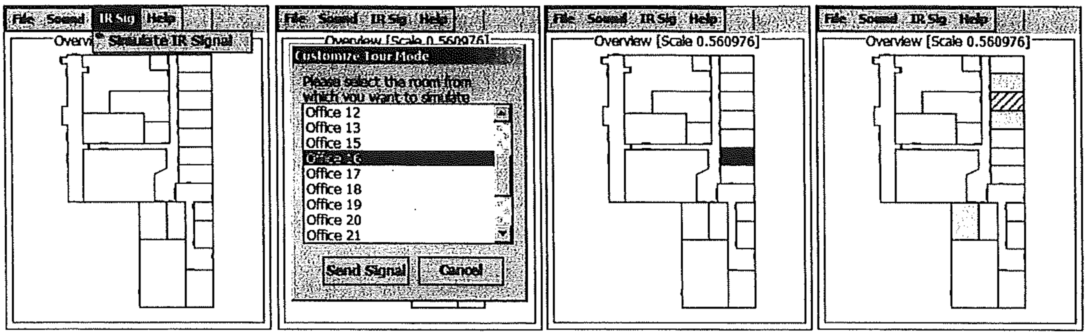


Figure 9-5 - The InfraRed Simulation Screenshots

To use the simulation, the user must manually specify the signal by using the IR Sig menu. He must then choose the room from which the IR signal is coming. After a delay of 1 sec, the infrared selected room will appear in red on the overview map, and from the room maps, all the connections leading to this room will also be highlighted in red. In the case where the room signalled by infrared is also a room to be seen, the chosen colour will be yellow (the "to see" colour) with additional red hatchings. Note that we imposed the delay of 1 sec so that the user can visualize and hear the change of location. Also note that the sound pattern can be changed by using the Sound menu.

9.3 Adapting Our Application For The Evaluation

As we were unable to test our application in a real museum, we had to adapt it. We chose to use it as a indoor guide for the fourth floor of the Computer Science Institute of the Facultés Universitaires Notre-Dame de la Paix, Namur. For this purpose, we changed the names of the Art Object attributes (in the ArtObject class header file), as follow :

- attribute1 : from "Artist" to "Person"
- attribute2 : from "Style" to "Title"
- attribute3 : from "Period" to "Domain"
- attribute4 : from "Country" to "Team"
- attribute5 : from "Material" to "Comment"

As we will see, this categorization was not the most judicious one. The complete specification file of the environment is given in Appendix B and a part of the result can be seen in Figure 9-6.

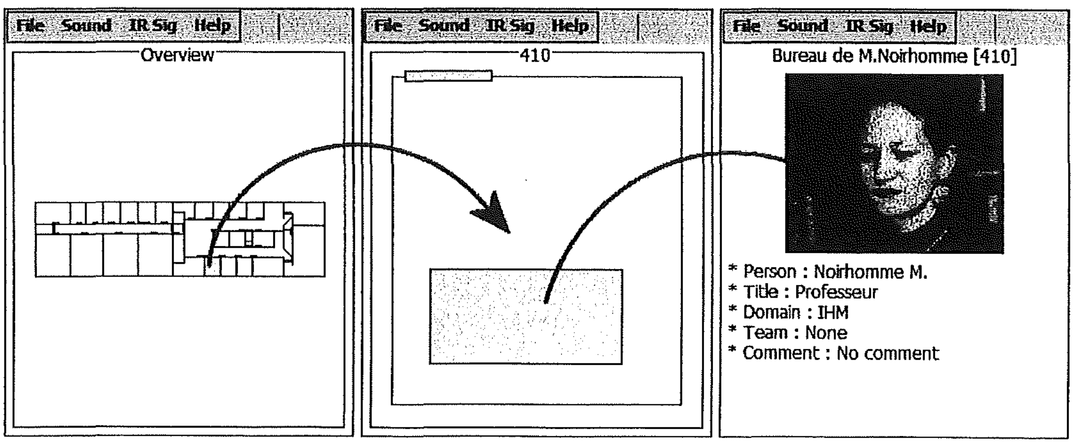


Figure 9-6 – The Fourth Floor of the Computer Science Institute

For the pictures, we used bitmap pictures created from photos available from the Computer Science website [FUNDP], and for the audio files, we used Text-To-Speech wave files generated with AT&T's Interactive Multi-Lingual Demo [AT&T].

9.4 Presentation Of The Questionnaire

9.4.1 Personal Information

First, the volunteers are asked to give some personal information (name, age, gender, title/position, field of activity and e-mail address) which will be used to identify the subjects and validate their answers (Table 9-1).

Name	
Age	
Gender	
Title/Position	
Field of Activity/Study	
E-mail Address	

Table 9-1 - Personal Information

9.4.2 Previous Experience in Sound and Mobility

The second step consists in analyzing more precisely the musical and mobilé profile of the subjects. We will ask them what is their musical background (did they follow any musical formation, do they play any music instrument, ...), if they are used to cellular phones, PDAs (or similar handheld devices) and if they play video games using maps for orientation within the game (Table 9-2).

Musical Background (formation, etc.)	None	Basic	Average	Upper	Expert
Previous Use of Mobile Phone	Never	Rare	Occasional	Often	Daily Basis
Previous Use of PDA (or similar)	Never	Rare	Occasional	Often	Daily Basis
Previous Use of Electronic Games	Never	Rare	Occasional	Often	Daily Basis
If it is the case, are there some of them using maps?	...				

Table 9-2 - Previous Experience in Sound and Mobile Devices

9.4.3 Scenarios

First of all, we will briefly explain the subjects the functioning of the prototype. Usually in museum, when an interactive support is provided to visitors, a brief explanation is *de facto* given to the users. We will then ask the subjects to perform a few scenarios in order to make them manipulate the application. We will time their performance to see if it is easy to use, and set one sound pattern for the first two scenarios, and the other one for the third scenario.

The first scenario will simply consist in searching and accessing the MAI team. The second scenario will have two phases. First, the subjects will have to find all the persons working in the "Ingénierie des Systèmes d'Information" domain, then they will have to find the Office of

Professor Noirhomme. Finally, in the third scenario, the subjects will have to find the office of all the professors, except the one of the two "new professors", then the projector, and to finish, the men's room (Table 9-3).

Sound Pattern	Auditory Icons / Earcon	Sound Pattern	Auditory Icons / Earcon
Scenario 1 (in sec)	...	Scenario 3a (in sec)	...
Scenario 2a (in sec)	...	Scenario 3b (in sec)	...
Scenario 2b (in sec)	...	Scenario 3c (in sec)	...

Table 9-3 - Scenarios

9.4.4 Interface General Rating

After manipulating the interface, we will ask the subjects to rate the lisibility of the interface, the navigation with the map and the menus, especially for searching specific objects (Table 9-4).

Lisibility	Difficult	Not so good	Average	Good	Excellent
Map Navigation	Difficult	Not so good	Average	Good	Excellent
Menus	Difficult	Not so good	Average	Good	Excellent

Table 9-4 - Interface Overall Evaluation

9.4.5 Interface Sound Rating

After the interaction with the device, we will ask the subjects their impressions about the sound in the application. We will try to see which sound pattern they preferred, and if they found the use of sound in the interface somehow helpful or not (Table 9-5).

Auditory Icons	Horrible	Not so good	Average	Good	Excellent
Earcons	Horrible	Not so good	Average	Good	Excellent
Sound Utility	Useless	Not so useful	Neutral	Useful	Essential

Table 9-5 - Sound Pattern Evaluation

9.4.6 General Comments

Finally, we will collect all their comments on the application, its handiness, its sound features, the hiatus and the possible improvements.

9.5 Results and Analysis

9.5.1 The subjects

We questioned 28 persons, half of which did the first two scenarios with the auditory icon sound pattern and the last scenario with the earcon sound pattern. The other half started with the earcon pattern, then switched to the auditory icon pattern.

The results will somehow be biased because the subjects contain :

- Only persons between 19 and 24 years old, mainly students;
- Only 5 woman;
- 17 persons from computer science, 4 from mathematics, 3 from geography, 1 from biology, 1 from philosophy, 1 from secretariat, 1 from mathematics with computer science orientation;

However, as 20 of them never used a PDA before, it will anyway be interesting analyzing how younger persons reacted to the prototype, because we know that young people are more at ease with new technologies. Once again, it is the lack of time that compelled us to restrain our investigations.

9.5.2 The Tool and the Methods Used

To analyze our results, we used the S-Plus data mining environment [S-Plus]. After converting all our data into a single input file, we proceeded to a **clustering** in order to determine which factors influenced the apprehension of the prototype (the time of execution for each scenario) and the appreciation of the sound pattern. The explicative factors were chosen from the personal information and background, and some of the qualitative answers we received were replaced by an ordinal value, as shown in Table 9-6. This was possible only because these qualitative answers expressed a gradation in the appreciation or the use of a specific matter.

Musical Background (formation, etc.)	None	Basic	Average	Upper	Expert
Previous Use of Mobile Phone	Never	Rare	Occasional	Often	Daily Basis
Previous Use of PDA (or similar)	Never	Rare	Occasional	Often	Daily Basis
Previous Use of Electronic Games	Never	Rare	Occasional	Often	Daily Basis
Auditory Icons	Horrible	Not so good	Average	Good	Excellent
Earcons	Horrible	Not so good	Average	Good	Excellent
Sound Utility	Useless	Not so useful	Neutral	Useful	Essential
Associated ordinal value	**1**	**2**	**3**	**4**	**5**

Table 9-6 - Ordinal Values for The Questionnaire Analysis

For those who are not familiar with data mining techniques, clustering is the process of grouping physical or abstract objects into classes of similar objects. It's principle is to maximize the intraclass similarity and minimize the interclass similarity.

After identifying the two main classes for each scenario, we should calculate the **critical probability** in order to determine if our results are significative or not. To do this, a certain number of steps must be done in order to first obtain the interval of confidence at 95% for our 2 classes population, which have the same unknown variance [Wonnacott&Wonnacott91] :

- Calculate the degree of liberty *ddl*, using the formula :

$$\boxed{ddl = (n_1 - 1) + (n_2 - 1)}$$

Where n_1 is the number of elements in the first class,
And n_2 the number of elements in the second class.

- Calculate s_p, the variance that the two classes have in common, using the formula :

$$\boxed{s_p^2 = \frac{\sum(X_1 - \overline{X_1})^2 + \sum(X_2 - \overline{X_2})^2}{ddl}}$$

Where X_i is a value of the class i, and $\overline{X_i}$ the mean of the class i.

- Calculate the interval of confidence at 95% (α = 0.050), using the formula based on the difference of means under the t distribution of Student (because of the small size of the population) :

$$(\mu_1 - \mu_2) = (\overline{X_1} - \overline{X_2}) \pm t_{.025} \cdot s_p \cdot \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}$$

Note that if we had a population with more than 100 persons, and a well known variance σ, we should have used the formula based on the classic normal distribution to calculate the interval of confidence at 95% :

$$(\mu_1 - \mu_2) = (\overline{X_1} - \overline{X_2}) \pm z_{.025} \cdot \sigma \cdot \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}$$

Once this is done, we can calculate the critical probability *pc* by calculating :

$$t = \frac{\overline{X_1} - \overline{X_2}}{sp \cdot \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}$$

Using the ddl[th] line of the Critical Values Table For the t Distribution of Student, available in Appendix D, section D.1, we can determine the $t_{.p}$ value for which $t > t_{.p}$, with $0 < .p < 1$.

We will therefore be able to conclude that pc < .p . In our case, we will consider that our findings are relevant and significative if $pc < 0.05$.

As a first step, we will systematically analyze for each scenario the critical probability for the class of subjects that never used a PDA before and the class of subjects that already used a PDA at least once before. More analysis will be done in completion (impact of musical background and video games) as an appendix that will be added later on to the current work.

The detailed results of our analysis can be found in Appendix D.

9.5.3 The Impact of the Personal Background on the Apprehension of the Prototype

For each scenario, we tried to explain the time necessary to accomplish the scenario based on factors that were chosen from the personal information and background : sex, field of activity, musical formation, previous use of cellular phone, PDA and video games. We did not use the age because all our subjects were between 19 and 24 years old, which was not so significative.

Scenario1

When trying to explain the first scenario, the first factor to come out was the previous use of a PDA. Indeed, the 8 persons who already used a PDA at least once before the test (corresponding ordinal value is 2 or higher) only needed 8.875 seconds in average, while the 20 others needed 18.05 seconds.

Among these 20 persons, the field of activity was the second determining factor. The 15 persons from biology, computer-science, geography, maths / computer science and philosophy needed

16.07 seconds, while the 5 remaining persons from mathematics and secretariat needed 24 seconds.

Finally among the 15 persons from biology, computer-science, geography, maths / computer science and philosophy, it was the musical formation that influenced the swiftness of execution. The 6 persons with at least basic musical notions (corresponding ordinal value is 2 or higher) needed 14.33 seconds, while the 9 others with no musical formation needed 17.22 seconds.

Now let's calculate the critical probability between the two main groups, namely the persons that never used a PDA before, and the ones who already used one at least once. In our case, n_1 is equal to 20, while n_2 equals to 8.

From that point, we obtain :

- ddl = (20-1) + (8-1) =26
- s_p^2 = (1378,95 + 30,875) / 26 = 54,22403846
- (μ1-μ2) = 9,175 ± 2,06 * 7,363697336 * SquareRoot(1/20 + 1/8)
  = 9,175 ± 2,06 * 7,363697336 * 0,418330013
  = 9,175 ± 6,345738545

We finally obtain :
t = 9,175 / (7,363697336 * SquareRoot(1/20 + 1/8))
t = 2,97845552

Using the 26th line of the Critical Values Table For the t Distribution of Student, available in Appendix D, section D.1, we find that $t > t_{.005}$, which means that pc < .005. Our conclusions are hence significative : the previous use of PDAs does have a real impact on the time of execution of the scenario.

Scenario2a

As for the first scenario, the first factor to come out was the previous use of a PDA. Once again, the 8 persons who already used a PDA at least once before the test were the fastest, needing only 8.75 seconds in average, while the 20 others needed 14.5 seconds.

Among these 20 persons, it was also the field of activity which turned out to be the second determining factor. It must be said that the 9 persons that came from biology, geography, maths, maths / computer science, philosophy and secretariat needed 12 seconds, while the 11 remaining persons, which all came from computer science, needed 16.55 seconds! This is quite surprising because one would expect computer science persons to obtain better results.

These findings should however be moderated, as the critical probability pc is < .10. This is confirmed by the fact that the interval of confidence is between -2,577064844 an 14,27706484. As 0 is part of the interval of confidence, no conclusion can be drawn.

Scenario2b

As for this scenario, the impact of the previous used of PDAs started to vanish, as the first determining factor was the field of activity. The 20 persons who came from computer science (17 persons) or geography (3 persons) only needed 9.95 seconds, while the other ones (coming from biology, maths, maths / computer science, philosophy and secretariat) needed 15.5 seconds.

If we look at the first 20 persons, we can however see that the influence of the previous use of a PDA is still there, because the 8 persons who already used a PDA at least once before the test were faster, needing only 8 seconds in average, while the 12 others needed 11.25 seconds.

If we focus on the two classes of persons who never used a PDA before (20 persons), and the ones that already did before (8 persons), we can see that the critical probability pc is < .025, which means that the previous use of a PDA is still significative.

Scenario3a

For this scenario, the previous use of PDAs clearly disapeared as an explicative factor. As a matter of fact, the first determining factor was the field of activity. The 23 persons from biology, computer science, mathematics and maths / computer science only needed 23 seconds, while the 5 others (from geography, philosophy and secretariat) needed 32.8 seconds.

In the first subgroup of 23 persons, the fastest were the 10 persons with at least basic musical notions, especially if they had a basic or an average musical formation (corresponding ordinal value is 2 or 3). In that case, the 5 persons concerned needed 18.2 seconds, while the 5 other persons, with an upper or expert musical formation (corresponding ordinal value is 4 or 5), needed 21 seconds. The 13 persons with no musical formation needed 25.62 seconds.

If we focus on the two classes of persons who never used a PDA before (20 persons), and the ones that already did before (8 persons), we can see that the critical probability pc is not smaller that .25, which means that the previous use of a PDA is absolutely not significative.

Scenario3b

As previously, the field of activity was once again the key factor, but this time, the groups were almost inverted. Indeed the 5 persons from biology, geography and secretariat only needed 13.4 seconds, while the 23 others from computer science, maths, maths / computer science and philosophy needed nearly twice as much time : 26.61 seconds.

Among these 23 persons, it is the 18 who did not play video games daily (corresponding ordinal value is 4 or lower) who were the fastest, needing 24.56 seconds, while the 5 daily players (corresponding ordinal value is 5) needed 34 seconds.

Finally, if we look at these 18 persons, it comes out that the 10 of them that had no musical formation needed 23.5 seconds, while the 8 others that had at least a basic musical formation needed 25.88 seconds.

If we focus on the two classes of persons who never used a PDA before (20 persons), and the ones that already did before (8 persons), we can see that the critical probability pc is < .25, which means that the previous use of a PDA is not so significative.

Scenario3c

As it now became usual, the determinant factor for the last scenario was the field of activity. The 23 persons from biology, computer science, geography, maths / computer science and philosophy needed 15.57 seconds, while the 5 remaining from mathematics and secretariat needed 28.8 seconds.

Among these 23 persons, the PDA factor reappeared, but in an unexpected manner. Indeed, the 15 persons who had never used a PDA before only needed 12.60 seconds, while the 8 persons which had used a PDA at least once before needed 21.13 seconds.

If we look at these 15 persons, we can see that the fastest were the 6 persons who had at least basic musical notion, with 10.5 seconds, while the 9 persons with no musical formation needed 14 seconds.

If we focus on the two classes of persons who never used a PDA before (20 persons), and the ones that already did before (8 persons), we can see that the critical probability pc is not smaller that .25, which means that the previous use of a PDA is absolutely not significative.

Conclusion

A first conclusion concerns the impact of the previous use of a PDA. For the first scenarios, it seems obvious that all the persons that had already used a PDA before were faster than the other ones. As from the first of the 3 last scenarios, the users had been able to become familiar with the application, but we must also say that the set of scenarios in the third part of the tests were slightly more difficult than the first ones.

It also seems that the musical formation had a positive impact, but this must still be analyzed using the critical probability. The field of activity also often appeared as a key factor, but no particular field distinguished itself from the others. The other explicative factors, especially the use of video games, did not seem to have any particular influence.

9.5.4 The Impact of the Personal Background on the Sound Appreciation

To explain the sound appreciation given by our subjects (for Auditory Icons, Earcons and Sound Utility), we used the same explicative factors as previously, but we added the first Sound Pattern used for the scenarios. We wanted to see how the fact of using a sound pattern before another could influence acceptance, but we also wanted to see if the Musical Formation and the use of cell phones could lead to a preference for earcons while the use of video games could lead to a preference for auditory icons. Please refer to Table 9-6 for the ordinal values associated to the appreciation answers we received.

The Auditory Icon Pattern Appreciation

We received the following answers for the Auditory Icon pattern appreciation:

- 1 horrible,
- 4 not so good,
- 3 average,
- 15 good,
- 5 excellent.

When analyzing the results, it is clearly the use of video games that influenced the rating of the pattern. Indeed, the 6 persons who played video games rarely or never (corresponding ordinal value is 2 or less) rated the pattern with an ordinal value of 2.667, which corresponds to a rather "negative average" appreciation. The 22 others, playing at least occasionally (corresponding ordinal value is 3 or higher) rated the pattern with an ordinal value of 3.955, which can fairly be considered as "good".

Among these 22 players, the best appreciation (4.118) came from 17 players that played occasionally or often. The 5 daily players rated the pattern as "positively average" (3.400).

Finally, amongst these 17 players, the best appreciation (4.300, that is "positively good") came from the 10 persons who started the scenarios with the earcon pattern before switching to the auditory pattern. The 7 other persons who used inverted patterns rated the auditory pattern as rather "good" (3.857).

The Earcon Pattern Appreciation

On the other hand, we received the following answers for the Auditory Icon pattern appreciation:
- 2 horrible,
- 4 not so good,
- 18 average,
- 4 good,
- 0 excellent.

As for the auditory icons, it is clearly the use of video games that influenced the rating of the pattern. Indeed, the 6 persons who played video games rarely or never rated the pattern with an ordinal value of 2.167, which corresponds to a rather "not so good" appreciation. The 22 others playing at least occasionally rated the pattern with an ordinal value of 3.045, which is "average".

Among these 22 players, the determining factor was the previous use of PDAs. The 6 persons who had already used a PDA at least once rated the pattern as "negative average" (2.667), while the 16 persons that had never used a PDA before were, if we can say, a little more enthusiastic, rating it as "average" (3.188).

And if we look at these 16 persons, it is the 6 persons with at least a musical formation that gave the lower appreciation, which was clearly "average" (3.000), while the 10 other persons with no musical background gave a "positive average" rating (3.300).

It is interesting to see if the earcons were directly influenced by the musical formation level, which we did in Table 9-7, by aggregating the negative answers (horrible, not so good) together, as well as the positive answers (good, excellent).

	Negative	Neutral	Positive	Total
No Musical Background	2	8	4	14
Musical Background	4	10	0	14

Table 9-7- Influence of the Musical Formation on the Earcon Appreciation

What clearly comes out is that the two groups (musicians and none musicians) were both rather neutral, while the persons with a musical background were definitely not positive.

The Relation between Auditory Icons and Earcons

If once again we aggregate the negative and the positive answers, we can summarize the sound patterns appreciation in Table 9-8.

Appreciation	Negative	Neutral	Positive	Total
Auditory Icons	5	3	**20**	28
Earcons	6	**18**	4	28

Table 9-8 Sound Pattern Appreciation Summary

It is somehow obvious that the auditory icon pattern was more successful than the earcon pattern. It would definitely be interesting leading this survey again, using different auditory icon and earcon sets, to see how the results would differ.

The Sound Utility Appreciation

Independently from the sound pattern preference, we received the following answers for the Sound Utility appreciation:

- 5 useless,
- 6 not so useful,
- 7 neutral,
- 9 useful,
- 1 essential.

At this point, a new crucial information can lighten the two previous considerations. In this case, we can only record that the 6 persons who played video games rarely or never rated the sound utility with an ordinal value of 1.833, which clearly corresponds to a "not,so useful at all" appreciation. The 22 others playing at least occasionally rated the pattern with an ordinal value of 3.091, which is "neutral".

Among these 22 players, the 16 persons from computer science, maths / computer science and secretariat were still rather sceptic, giving a rather "negative neutral" appreciation (2.688), while the 6 persons from biology, geography and mathematics were clearly enthusiastic, giving a "useful" appreciation (4.167).

If we look at these 16 persons, it comes out that the 5 persons playing video games daily were more severe towards the sound pattern (2.200, which means "not so useful"), while the 11 others remained rather "neutral" (2.909).

It becomes obvious that the 6 persons who did not appreciate the two sound patterns did not find the sound useful anyway. These persons do not usually play video games, which can explain a certain annoyance to the repeated use of sound in the application. But as a matter of fact, we must realize that no real trend comes out of the analysis, as 11 persons are rather negative, 10 are rather positive, and 7 remain neutral.

9.5.5 More General Issues of Fact

Now we come to more general considerations about the handiness of the application : the lisibility, the navigation and the menus.

For the lisibility at the screen, the size of the characters and the maps, we received quite positive answers :

- 1 average,
- 19 good,
- 8 excellent.

For the navigation through the various map levels, we also received rather enthusiastic answers :
- 2 average,
- 16 good,
- 10 excellent.

In contrary, we received more moderate answers for the menus:
- 1 not so good,
- 15 average,
- 10 good,
- 2 excellent.

In general, the users found the interface quite easy to use and to navigate into. They appreciated the fact that the information were rather clear and simple. From the navigation point of view, three problems were mentioned a fair number of times :
- The lack of landmarks when zooming at the room level of a corridor,
- The lack of a "back" button on the room map and art object information screen,
- The size of the connections in the room maps.

But the major problem came from the Restart Tour menu that the users had to use to perform a search for the scenarios. The users would have expected a real Search menu, but above all, they were disoriented by the classification of the objects. It is not the application itself that was pointed out, but the specification file used to create the environment for the evaluation.

Indeed, for the evaluation, the art objects we were referring to were the desks of the staff as well as other utility objects. It was easy to establish a link between a desk and its owner, but it was more troublesome to associate a person to a utility object. The problem is that because of the prototype's implementation, it was mandatory to fill each of the five attributes (Person, Title, Domain, Team, Comment) of the "art objects" contained in each room. That's why some of the "art" objects had quite exotic attribute values, such as the projector in Table 9-9. This shows how important it is to correctly specify the environment one wishes to represent. Some ideas to improve this issue will be given in the next part of this chapter.

```
#code= projecteur-reunion
#name= Projecteur
#attrib1= Projecteur              // Person
#attrib2= Objet Utilitaire        // Title
#attrib3= Utilitaire              // Domain
#attrib4= None                    // Team
#attrib5= No comment              // Comment
#coordinates=
250,60
260,65
```

Table 9-9 - Excerpt of the Specification File for the Prototype Evaluation

9.6 Possible Improvements And Opportunities For The Application

9.6.1 Immediate Improvements

First of all, following the comments gathered from the evaluation questionnaire, some immediate improvements should be brought to our application, especially at the menu and navigation levels.

For the menus, we should start by creating a real "Search" menu, instead of using the "Restart Tour" command, which would be more intuitive. In this search section, the name of the "Begin" button should be changed, using an "OK" or "Continue" button. In the final selection screen, the objects should not all be selected by default, because we noticed that users thought they had to explicitly reselect the objects they wanted to see.

Another modification should be made regarding the handling of the art objects attribute. As we saw, adapting the application to the fourth floor of the Computer Institute was not so trivial. Therefore, if we really want a generic indoor guide, we should allow to specify a variable number of categories for each object, under the constraint that giving a value to each category should not be mandatory. We should also allow to modify the name of the categories without modifying the code of the application itself (as we did when testing our application).

Finally, for the room maps, we should manage the context of each room. This means that we should also represent the adjacent rooms, which will enable the users to orientate themselves more easily, in particular in the corridors. We could also consider handling different zoom levels, but this would rather be a further improvement. Another important improvement should be done by adding a "Back" button to the room maps and the art object information screen. This lack has indeed been mentioned several times.

These modifications should allow to improve the handiness and user-friendliness of the application, and should constitute a first serie of changes. Afterwards, the specification file of the fourth floor of the Computer Institute could be rewritten in more appropriate way, and we could consider implementing secondary improvements.

9.6.2 Further Improvements

One of the aspects that we left aside during this work was the handling of multi-level environments, that is environments with several floors. For our application, we only considered three levels of maps : overview, room and art object. A solution would be to manage a fourth level, which would be a "Section" map. The overview would contain several sections, each section would contain several rooms and links to adjacent sections, and each room would contain several art objects and links to adjacent rooms. We could even consider that a section could be recursive, which means that a section could itself contain other sections. Switching from a floor to another will therefore be considered as switching from a section to another.

Another enhancement could consist in managing the art objects located on the walls of the rooms. This would be relevant for the museums with paintings, because there are usually many paintings on a same wall, and they are often vertically located under another painting. The idea would be to allow a user to switch from a room map to a "wall map" for each wall of the room. The user would therefore be able to select directly the art objects from the images on the corresponding wall.

Other improvements should be made to the maps handling. First, we should try to solve the "inner room problem" that we mentioned in Chapter 8. A solution would be to explicitly mention this property in the specification file. We could also consider allowing to explicitly specify the entrances and exits of a section, which could for instance be represented on the screen by incoming or outgoing arrows. We could also consider allowing the environment creators to use personalized icons or pictures to represent the different categories of objects on the room maps.

All this leads us to another key improvement : the specification of an environment. It would really be a good thing to create a specification application which would allow to easily specify an environment, and automatically generate the corresponding specification file. This intermediate application could even provide a graphical window to directly draw the sections, the rooms and the art objects without having to know how the graphical visualization of their environment will be converted into a specification file. The format of the specification file could also be modified, for instance by using the XML standard. By using special conventions, it could allow to have portable files that could be exchanged from our application to another, like web-based museum managing applications.

Also, the way to manage the ToSee list should be rethought. Users should be able to edit and reinitialize the list of art objects highlighted on the maps. They should also be able to make searches not only on art objects attributes, but also on Room properties.

And finally, all the default messages boxes of the application should be reimplemented to prevent the language clash problem. Indeed, using default messages boxes can lead to unexpected surprises, such as having the text of the message box in English, while the buttons are in French! This is definitely annoying.

9.6.3 Opportunities

Used as a canvas, our application could progressively integrate several new technologies. As it is a mobile application, it could definitely take advantage of speech recognition and speech synthesis to interact with the users. Using real-time localisation, the application could become even more attractive, and provide interesting functionalities such as the Soundholder mentioned in Chapter 5. Using the latest communication technologies, all the data could be moved on-line instead of being installed directly on the mobile device. All these enhancements could lead this application to be used in distant surveillance, maintenance and domotic systems, with connections to video-cameras and electric/electronic appliances that could be controlled remotely. This would require a whole abstract specification system that would allow to specify the different commands and controls of any remotely manageable appliance.

9.7 Conclusion

As we have seen, our prototype received a rather enthusiastic welcome. The subjects found it quite easy to use, although a few problems (especially specification ones) occurred. It was the persons who had already used a PDA before that got along faster with the application, at least for the first scenarios. Apart from that, no real preferences could be derived from their appreciation of the sound patterns and their utility. Based on their comments, we then proposed possible improvements to be done prioritarily, as well as some "bonus" improvements to be done later. As for the future, the opportunities are alluring, which means that a lot remains to do in this mobile guiding field.

10 Conclusion

Throughout this dissertation, our objective was to set good foundations to create a generic museum guide for PDA with sound enhancement. The field of mobile guiding is actually flourishing, and we took advantage of previous works to implement a prototype, which turned out to be a generic indoor guide.

We started by analyzing the traditional behavior of museum visitors, which led us to conclude that they expected richer and more accessible information and guiding systems. By taking a look at the state of art in the mobile guiding domain, we saw that there were many ways to tackle the problem and to take advantage of the available communication technologies. Alas, we also saw that these technologies remained quite expensive and difficult to install for a widely spread use.

Moreover, handheld devices such as PDAs still suffer limitations, especially when it comes to displaying information and managing user interaction. We hence gave a first idea of the enhancements that sound and speech could bring to mobile interfaces. For this purpose, we mentioned the visual channel limitations, as well as common problems in graphic interfaces. It turned out that speech and sound could be used to offer a richer experience to users, as well as more liberty, autonomy and support.

We were therefore ready to give the outline of the kind of generic museum guiding application that we wanted to implement for PDA. Our approach consisted in considering a few scenarios, which helped us to point out the main tasks we would have to deal with. Using the ConcurTask Trees Notation, we were then able to elaborate a first task tree for our application.

As the use of sound was one of the main concerns of this work, we decided to analyze more in details the different sonification methods commonly used, namely the Auditory Icons and the Earcons. Choosing a single sound pattern was not so trivial, and we finally had to define two distinct sound patterns for our application, based on several guidelines.

It was then necessary to examine the existing programming languages available for mobile applications implementation. Although quite similar from the coding techniques and the (rather limited) sound features, we saw that the two main languages available, that is Java and C++, differed by their usability and distribution. We finally settled for the eMbedded C++, notably because it offered a visual environment and a built-in Pocket PC emulator, while Java did not seem so "programmer-friendly" for designing mobile applications.

From then, we had surveyed the different possibilities and constraints, which led us to refine the definition and features of our application. As time was limited, we had to set several limitations to our initial project. We exposed the main steps of the conception phase, by establishing the class diagram and the new task tree of our application before defining the data storage method and logical components. We then got more practical, and gave an overview of the class hierarchy of the intended architecture.

As it could be expected, the implementation process brought his share of problems. That's why we also discussed some of the issues we had to face during the coding phase. For instance, we explained how to format the specification file of a given environment, how to use custom sounds and images in the application, and how to give the coordinates of each element of the environment. We also mentioned the non trivial problem of determining the target of the stylus input for the navigation in the different map levels of the application. Finally, we presented some limitations inherent to the implementation language we used.

We then finally unveiled our application and presented its interface. In order to appraise the prototype, we tested it with 28 young persons, using the specification of the fourth floor of the Computer Science Institute of Namur University. We submitted an evaluation questionnaire to the volunteers in order to collect their impressions, and to consequently draw some teachings and envisage possible improvements.

The analysis of the results showed that the subjects were rather enthusiastic about the prototype, although they spotted a few problems, especially concerning the object search and the specification of the environment. As we could have guessed, the persons who had already used a PDA before got along faster with the application, at least for the first scenarios. It is interesting to see that although most of the subjects came from the Computer Science field, they were not particularly faster than the other subjects. Apart from that, we were not able to derive a real trend regarding the sound patterns and their utility. This means that there is still much to do in daily applications sonification. Then, based on the comments of our volunteers, we were able to suggest prioritary improvements as well as further improvements to be done later.

We shall now conclude by saying that this work was only a beginning. At our humble level, although we can see that some improvements still have to be made, we can above all consider some quite interesting opportunities. So when you think that the mobile guiding domain has still so much to offer, the perspectives are simply breathtaking. The people are on the move, so are the communication and mobile technologies. It would be sad not to take our chance.

11 Bibliography

[ActiveArt99] The ActiveArt Project, "Part 1: Defining the requirements, Version 2.0", October 5, 1999. Available from : http://www.cc.gatech.edu/classes/AY2000/cs6750a_fall/projects/Team5/part1_v2.html

[ANPT73] American National Standards Institute, *American National Psychoacoustic Terminology* (No. S3.20), New York, 1973.

[ARKola91] W. Gaver, R. Smith, & T. O'Shea, "Effective sounds in complex systems: The ARKola simulation", In S. Robertson, G. Olson, & J. Olson (Eds.), *Proceedings of CHI'91*, New Orleans: ACM Press, Addison-Wesley, pp. 85-90, 1991.

[AT&T] AT&T Labs-Research Interactive Multi-Lingual Demo. Available from : http://www.research.att.com/projects/tts/demo.html

[Beaudouin&Conversy96] M. Beaudouin-Lafon, and S. Conversy, "Auditory illusions for audio feedback", In *ACM CHI'96 Conference Comp*anion, ACM Press, Addison-Wesley, 1996, pp. 299-300.

[Blattner&al.89] M. Blattner, D. Sumikawa and R. Greenberg, "Earcons and icons: Their structure and common design principles", *Human Computer Interaction, 4*, 1989, pp 11-44.

[Boling2001] Douglas Boling, "Programming Windows CE (Second Edition)", Microsoft Press, June 2001.

[Borowski&Borwein91] E.J Borowski & J.M. Borwein, "Collins Dictionnary Of Mathematics", Third Printing, Harper Collins Publishers, 1991.

[Brewster94] S. A. Brewster, "Providing a structured method for integrating non-speech audio into human-computer interfaces", PhD Thesis, University of York, UK, 1994.

[Brewster&al.94] S.A. Brewster, P.C. Wright, and A.D.N. Edwards, "The design and evaluation of an auditory-enhanced scrollbar", In *Proceedings of ACM CHI'94*, ACM Press, Addison-Wesley, 1994, pp. 173-179.

[Brewster&al.95A] S.A. Brewster, P.C. Wright, A.J. Dix and A.D.N. Edwards, "The sonic enhancement of graphical buttons, In *Proceedings of IFIP Interact'95*, Chapman & Hall, 1995, pp. 43-48.

[Brewster&al.95B]S.A. Brewster, P.C. Wright, & A.D.N. Edwards, "Experimentally derived guidelines for the creation of earcons", In *Adjunct Proceedings of BCS HCI'95*, Huddersfield, UK, 1995, pp. 155-159.

[Brewster98] S.A. Brewster, "Using Non-Speech Sounds to Provide Navigation Cues", In *ACM Transactions on Computer-Human Interaction*, 1998; 5, 3: pp 224-259.

[Brewster&Crease99] S.A. Brewster, and M.G. Crease, "Correcting Menu Usability Problems With Sound", Accepted for publication in *Behaviour and Information Technology 17*, 1999.

[Brewster&al.2000] S.A. Brewster, A. Crossan, and M.G. Crease, "Automatic volume control for auditory interfaces", In *Volume II Proceedings of BCS HCI 2000*, Sunderland, UK, 2000, pp 17-18.

[Buxton&al.94] William Buxton, William Gaver, Sarah Bly, "Auditory Interfaces : The Use of Non-Speech Audio at the Interface", work stopped in progress in 1994. Available from : http://www.billbuxton.com/Audio.TOC.html

[ChaiVM] Hewlett Packard's Chai VM Homepage. Available from : http://www.hp.com/emso/products/chaivm/index.html

[Ciavarella & Paterno] C. Ciavarella, F. Paterno, "Design of Handheld Interactive Suport".

[CloudGarden] The Cloud Garden Homepage. Available from : http://www.cloudgarden.com/

[Cohen&Ludwig91] M. Cohen & L. Ludwig, "Multidimensional audio window management", In *International Journal of Man-Machine Studies*, 34(3), pp319-336, 1991.

[Cohen&Koizumi92] M. Cohen & N. Koizumi, "Exocentric control of audio imaging in binaural telecommnication", In *IEICE Trans. Fundamentals.* E75-A(2), pp164-192, 1992.

[Cohen93] M. Cohen, "Throwing, pitching and catching sound: Audio windowing models and modes", In *International Journal of Man-Machine Studies*, 39, pp. 269-304, 1993.

[Conversa] The Conversa Web Homepage. Available from : http://www.conversa.com/

[Cooltown2000] T. Kindberg et al., "People, Places, Things: Web Presence for the Real World". In *Proceedings of 3rd Workshop on Mobile Computer Systems and Applications*, December 2000, pp. 19-28.

[Cooltown2002] The Cooltown Beacons Homepage. Available from : http://cooltown.hp.com/beacon_full.htm

[Cooltown-WEB] The Cooltown Project Homepage. Available from : http://www.cooltown.hp.com

[CrEme] Creme's Java Enabler for Windows CE Homepage. Available from : http://www.nsicom.com/products/creme.asp

[CTTE] The ConcurTaskTree Environment Homepage. Available from : http://giove.cnuce.cnr.it/ctte.html

[Cyberguide95] S. Long, D. Aust, G.D. Abowd and C. Atkeson, "Cyberguide: Prototyping Context-Aware Mobile Applications", in *CHI'96 Short Papers*, December 1995.

[Cyberguide96] S. Long, R. Kooper, G.D. Abowd, C.G. Atkeson, "Rapid Prototyping of Mobile Context-Aware Applications: The Cyberguide Case Study", in *Proceedings of 2nd ACM International Conference on Mobile Computing*, Rye NY, 1996, ACM Press.

[Cyberguide-WEB] The Cyberguide Project Home Page. Available from :
http://www.cc.gatech.edu/fce/cyberguide/

[DeHaan] The Homepage of Ab De Haan. Available from :
http://www.socsci.kun.nl/~dehaan
http://www.socsci.kun.nl/~dehaan/earcons/earcons.html

[Demoulin&Schöller2001] C. Demoulin, O. Schöller, "Sonification of Time-dependant Data", Master Degree Thesis, Facultés Universitaires Notre-Dame de la Paix, Namur, 2001.

[Dutoit97] T. Dutoit, 1997, "High-Quality Text-to-Speech Synthesis : an Overview", Journal of Electrical & Electronics Engineering, *Australia: Special Issue on Speech Recognition and Synthesis*, vol. 17 n°1, pp. 25-37.

[Elan] The Elan Speech Cube Homepage. Available from :
http://www.elantts.com/products/fp_speechcube.html

[Esmertec] Anonymous, "Esmertec acquires Insignia JVM business assets and becomes master distributor of Insignia's Java Virtual Machine solution", November 12, 2002. Available from :
http://www.esmertec.com/press/announces_support.shtm

[EsmertecJbed] Esmertec's Jbed ME (Micro Edition) Homepage. Available from :
http://www.esmertec.com/technology/jbed_me.shtm

[Faivre2001] Christophe Faivre Duboz, "Présentation d'algorithmes permettant de contrôler l'intersection de deux objets géographiques", November 2001. Available from :
http://www.iag.asso.fr/articles/intersection.htm

[Festival] The Festival Project Homepage. Available from :
http://www.cstr.ed.ac.uk/projects/festival/

[Festival99A] Alan W Black, Paul Taylor and Richard Caley, "The Festival Speech Synthesis System Documentation, Edition 1.4, for Festival Version 1.4.0", 17th June 1999. Available from :
http://www.cstr.ed.ac.uk/projects/festival/manual/festival_28.html#SEC134

[Festival99B] Paul Taylor, Richard Caley, Alan W. Black, Simon King, "The Edinburg Speech Tools Library System Documentation Edition 1.2, for 1.2.0", 15th June 1999. Available from :
http://www.cstr.ed.ac.uk/projects/speech_tools/manual-1.2.0

[Freeplay] The Free Schooling Freeplay Homepage. Available from :
http://chronogears.sourceforge.net/html/download.html

[FreeTTS] The FreeTTS Homepage. Available from :
http://freetts.sourceforge.net/

[FUNDP] The Facultés Universitaires Notre-Dame de la Paix Computer Science Institute's Homepage. Available from :
http://info.fundp.ac.be/

[Gaver86] William Gaver, "Auditory Icons: Using sound in computer interfaces.", In *Human Computer Interaction*, 1986 , pp. 167-177.

[Gaver89] William Gaver, "The SonicFinder: An interface that uses auditory icons", In *Human Computer Interaction, 4*, 1989, pp 67-94.

[GUIDE2000] K. Cheverst, N. Davies, K. Mitchell, A. Friday,C. Efstratiou, "Developing a Context-aware Electronic Tourist Guide: Some Issues and Experiences", in *Proceedings of CHI 2000*, Netherlands, April 2000, pp 17-24.

[Hermann99] T. Hermann, "Data Exploration by sonification", 1999. Available from : http://www.techfak.uni-bielefeld.de/ags/ni/projects/datamining/datason/datason_e.html

[HIPS97A] J. Broadbent, P. Marti, "Location aware mobile interactive guides : usability issues", in ICHIM'97, Paris, September 1997.

[HIPS97B] J. Broadbent, P. Marti, "Envisionning HIPS with the users", in i3magazine n.1, 1997.

[HIPS-WEB] The HIPS Home Page. Available from : http://marconi.ltt.dii.unisi.it/progetti/HIPS/

[HP] The Hewlett-Packard / Compaq Home Page for Handheld Computers. Available from : http://www.compaq.com/products/handhelds/pocketpc/

[i3-WEB] The European Network for Intelligent Information Interfaces. Available from : http://www.i3net.org/

[ICAD] The International Community for Auditory Display Web Site. Available from : http://www.icad.org/

[IBMWebSphere] IBM's WebSphere Micro Environment Homepage. Available from : http://www-3.ibm.com/software/wireless/wme/

[ILEX99] R. Cox, M. O'Donnell, J. Oberlander, "Dynamic versus static hypermedia in museum education: an evaluation of ILEX, the intelligent labelling explorer" in *Proceedings of the Artificial Intelligence in Education conference*, Le Mans, July 1999.

[ILEX2002] The ILEX Project Homepage. Available from : http://www.hcrc.ed.ac.uk/ilex/

[Java1] The PersonalJava Homepage. Available from : http://java.sun.com/products/personaljava

[Java2] The J2ME CDLC Homepage. Available from : http://java.sun.com/products/cldc

[Java3] The J2ME MIDP Homepage. Available from : http://java.sun.com/products/midp

[Java4] The J2ME MMAPI Homepage. Available from : http://java.sun.com/j2me/docs

[Java5] The J2ME CDC Homepage. Available from : http://java.sun.com/products/cdc

[Java6] The J2ME Foundation Profile Homepage. Available from :
http://java.sun.com/products/foundation

[JeodeEVM] Insignia's JVM Products. Available from :
http://www.insignia.com/content/products/jvmProducts.shtml

[JSAPI1] The Java Speech 1.0 API specification, October 26, 1998. Available from :
http://java.sun.com/products/java-media/speech/

[JSAPI2] Anonymous, "Conversay Hosts IBM, Motorola, Nokia and Sun in Java Speech API Expert Group Meeting", in *Speech Technology Magazine*, August 6, 2002. Available from :
http://www.speechtechmag.com/pub/industry/1032-1.html

[Jones&Furner89] S.D. Jones & S.M. Furner, "The construction of audio icons and information cues for human-computer dialogues", In T. Megaw (Ed.), *Contemporary Ergonomics: Proceedings of the Ergonomics Society's 1989 Annual Conference*, Reading, UK: Taylor & Francis, pp. 436-441, 1989.

[KadaVM] The Kada Mobile Platform Homepage. Available from :
http://www.kadasystems.com/resources/mobile_platform.jsp

[L&H] Lernout & Hauspie's TTS for Java Speech API Homepage. Available from :
http://www.lhs.com/

[LoL@2002] G. Pospischil, M. Umlauft, E. Michlmayr, "Designing LoL@, a Mobile Tourist Guide for UMTS", in Proceedings of 4th International Symposium, Mobile HCI 2002, Pisa, September 2002, pp 140-154.

[Louvre-WEB] The Louvre Museum Official Website. Available from :
www.louvre.fr/louvrea.htm.

[Lucas94] P. Lucas, "An evaluation of the communicative ability of auditory icons and earcons", In G. Kramer (Ed.), Accepted for publication in the *Proceedings of ICAD'94*, Santa Fé Institute, Santa Fé, NM.: Addison-Wesley, 1994.

[MBROLA] The MBROLA Project Homepage. Available from :
http://tcts.fpms.ac.be/synthesis

[MicroJBlend] The microJBlend Homepage. Available from :
http://www.jblend.com/en/overview/microJBlend.html

[MicrosoftPDA] Microsoft, "Helpful facts about Personal Digital Assistants". Available from :
http://www.microsoft.com/mobile/pocketpc/pdainfo.asp

[MIM] The Musical Instruments Museum Website. Available from :
http://www.mim.fgov.be/home_fr.htm

[Mountford&Gaver90] S.J. Mountford & William Gaver, "Talking and listening to computers", In B. Laurel (Ed.), *The art of human-computer interface design*, pp. 319-334. Reading, Massachusetts: Addison-Wesley, 1990

[NFS97]-"Sonification Report: Status of the Field and Research Agenda", Prepared for the National Science Foundation by Members of the International Community for Auditory Display, Fall 1997. Available from :
http://www.icad.org/websiteV2.0/References/nsf.html

[Noirhomme2000] M. Noirhomme-Fraiture, "Le son dans les interfaces IHM : Application à la représentation de données multivariées complexes", in *2èmes Journées Multimédia*, Namur, September 2000.

[Noirhomme&al.2002] M. Noirhomme-Fraiture, O.Schöller, C. Demoulin, S. Simoff, "Sonification of time dependant data", In S. Simoff, M. Noirhomme-Fraiture, M. Böhlen (Eds.), *Proceedings of ECML / PKDD-2002Workshop, Second International Workshop on Visual Data Mining,* Helsinki, August 2002.

[Palm] The Palm Handheld Family. Available from :
http://www.palm.com/products/handhelds/

[Paternò2000] Fabio Paternò, "Model-based Design And Evaluation Of Interactive Application", Springer Verlag 2000.

[Patterson82] R.D. Patterson, "Guidelines for auditory warning systems on civil aircraft", CAA Paper No. 82017, Civil Aviation Authority, London, 1982.

[Patterson89] R.D. Patterson, "Guidelines for the design of auditory warning sounds", In *Proceeding of the Institute of Acoustics*, Spring Conference, 11(5), pp. 17-24, 1989.

[Philippov2001] Vassili Philippov, "Play wav sound on Pocket PC", August 23, 2001. Available from :
http://www.pocketpcdn.com/articles/playsound.html

[Rememberer] M. Fleck, M. Frid, T. Kindberg, E. O'Brien-Strain, R. Rajani, M. Spasojevic, "Rememberer: A Tool for Capturing Museum Visits".

[SharedARK90] W. Gaver & R. Smith, "Auditory icons in large-scale collaborative Environments", In D. Diaper, D. Gilmore, G. Cockton, & B. Shackel (Eds.), *Human Computer Interaction: Interact'90*, Cambridge, UK: Elsevier Science Publishers B.V. (North Holland), pp. 735-740, 1990.

[SALT] The Speech Application Language Tags Frequently Asked Questions Homepage. Available from :
http://www.saltforum.org/faq/faq.asp

[SavaJe] The SavaJe OS Homepage. Available from :
http://www.savaje.com/products/index.html

[SOLE] The SOLE Project Homepage. Available from :
http://www.cstr.ed.ac.uk/projects/sole

[SottoVoce2002] Paul M. Aoki, Rebecca E. Grinter, Amy Hurst, Margaret H. Szymanski, James D. Thornton, Allison Woodruff, "Sotto voce: exploring the interplay of conversation and mobile audio spaces" in *Proceedings of the SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves*, Minneapolis, Minnesota, April 2002.

[SpeechForJava] IBM's Speech for Java Homepage. Available from :
http://www.alphaworks.ibm.com/tech/speech

[S-Plus] Insightful's S-PLUS for Windows Homepage. Available from :
http://www.insightful.com/products/product.asp?PID=3

[SuperWaba] SuperWaba's Java VM for PDAs Homepage. Available from :
http://www.superwaba.com.br/

[Vstudio] The Microsoft eMbedded Visual Tools Product Information on Microsoft eMbedded Visual C++ 3.0. Available from :
http://msdn.microsoft.com/vstudio/device/embedded/prodinfo.asp#section3

[Warin2002] Cedric Warin, "Sound As Means for Data Representation for Data Mining", Master Degree Thesis, Facultés Universitaires Notre-Dame de la Paix, Namur, 2002.

[WabaSoft] WabaSoft's Waba Homepage. Available from :
http://www.wabasoft.com

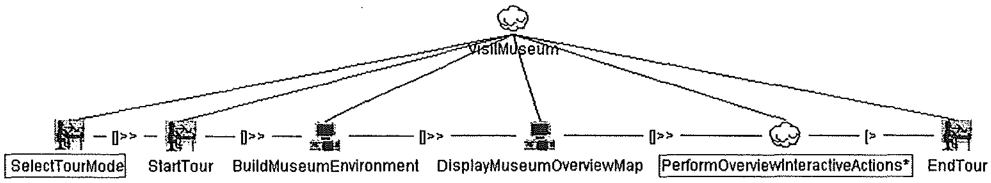
[Wolraich2001] Michael Wolraich, "Playing audio on the Pocket PC", September 20, 2001. Available from :
http://www.pocketpcdn.com/articles/pocketpcaudio.html

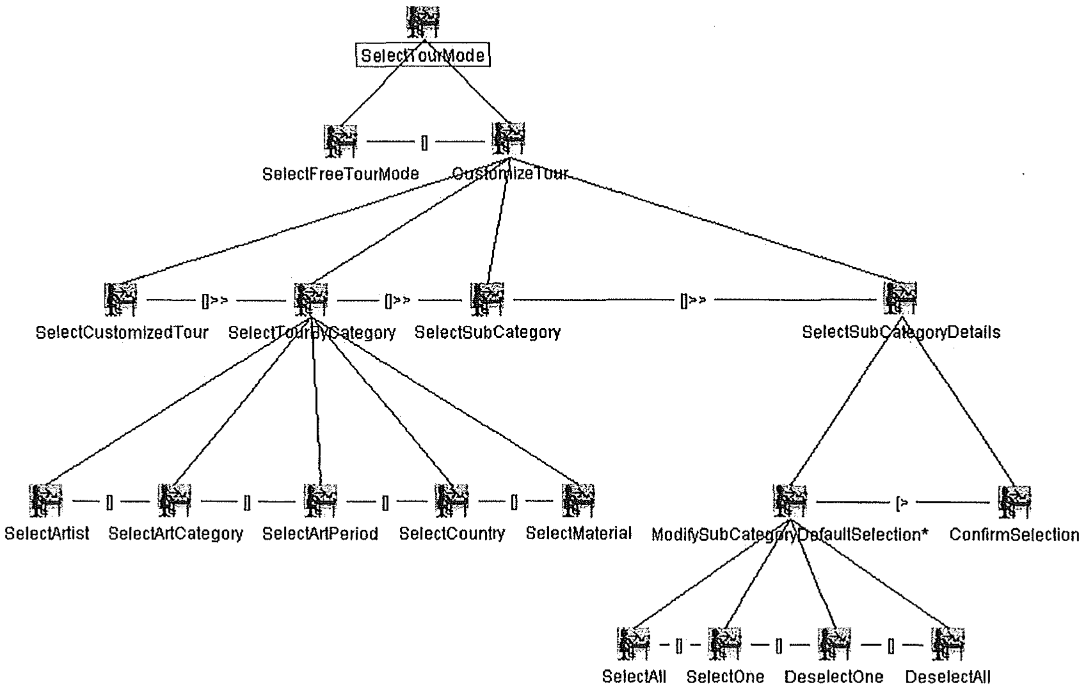
[Wonnacott&Wonnacott91] Thomas H. Wonnacott, Ronald J. Wonnacotf, "Statistique : Economie – Gestion – Sciences – Médecine (avec exercices d'application)", 4ème Edition, Economica, Paris, France, 1991.

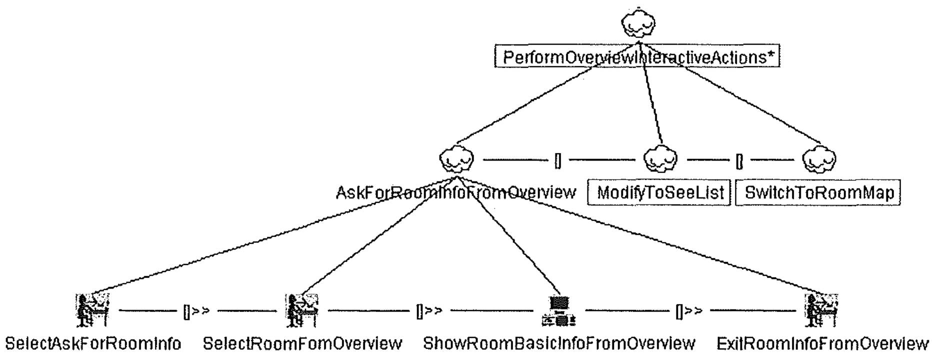
Appendix A : Detailed ConcurTask Trees

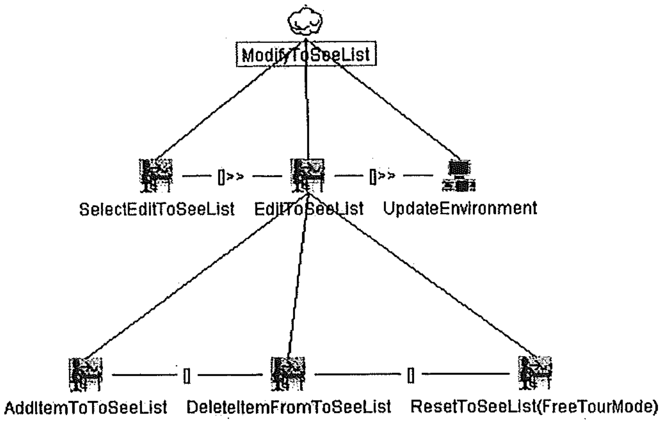
The following pages give the detailed ConcurTask Tree for the first approach to our application exposed in Figure 4-4 of Chapter 4, as well as the refined Tree presented in Figure 7-2 of Chapter 7.

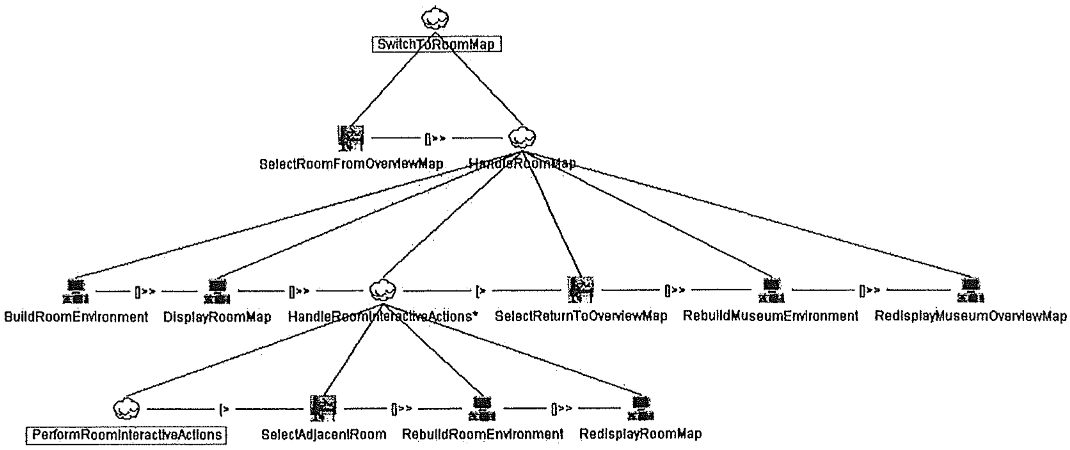
A.1 The First ConcurTaskTree Approach For Our Application

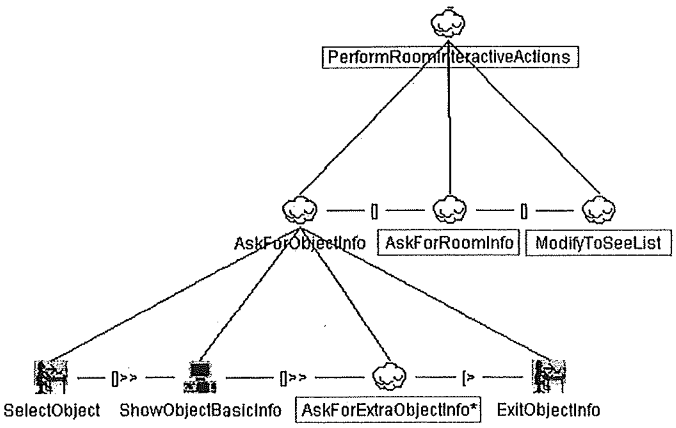


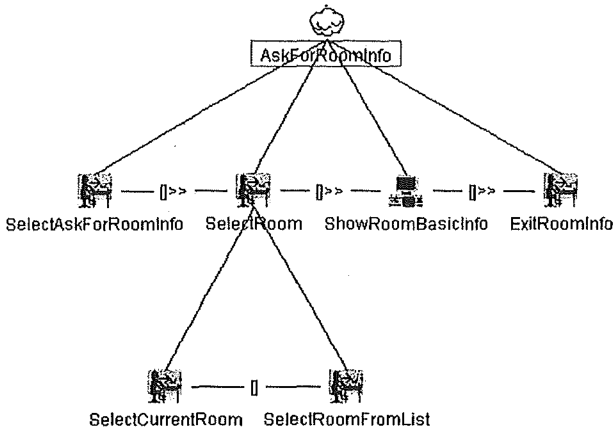


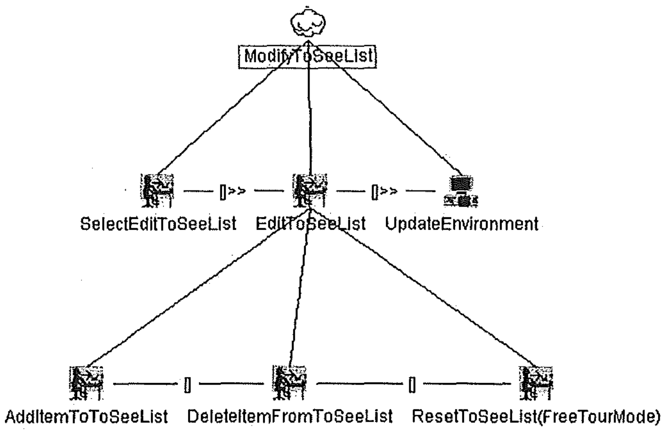


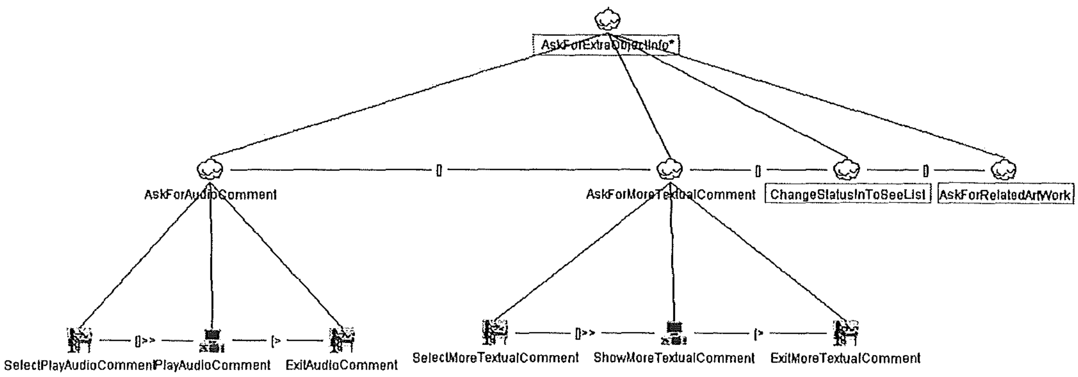

ModifyToSeeList
SelectEditToSeeList
EditToSeeList
UpdateEnvironment
AddItemToToSeeList
DeleteItemFromToSeeList
ResetToSeeList(FreeTourMode)

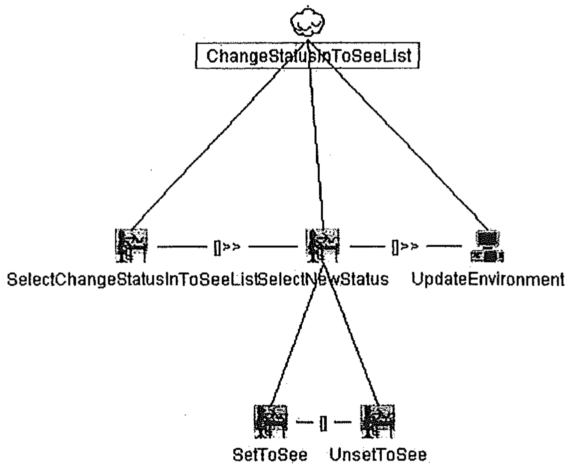

SwitchToRoomMap
SelectRoomFromOverviewMap
HandleRoomMap
BuildRoomEnvironment
DisplayRoomMap
HandleRoomInteractiveActions*
SelectReturnToOverviewMap
RebuildMuseumEnvironment
RedisplayMuseumOverviewMap
PerformRoomInteractiveActions
SelectAdjacentRoom
RebuildRoomEnvironment
RedisplayRoomMap

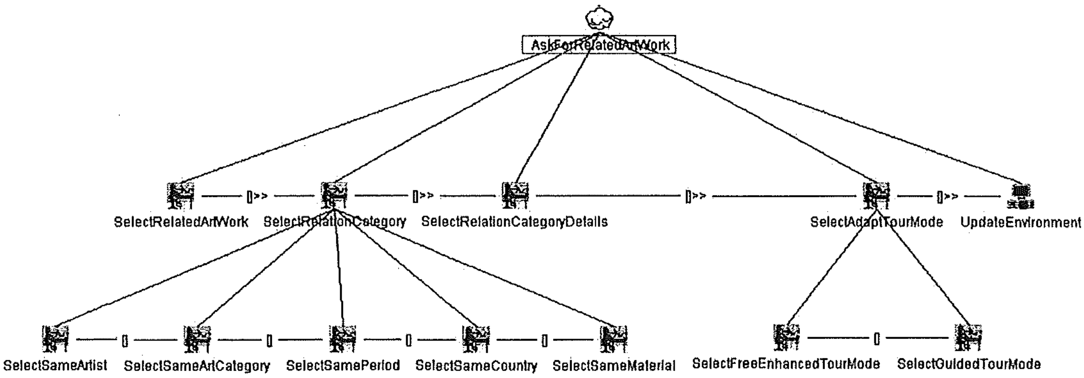

PerformRoomInteractiveActions
AskForObjectInfo
AskForRoomInfo
ModifyToSeeList
SelectObject
ShowObjectBasicInfo
AskForExtraObjectInfo*
ExitObjectInfo


AskForRoomInfo
[]>>
SelectAskForRoomInfo
SelectRoom
ShowRoomBasicInfo
ExitRoomInfo
[]
SelectCurrentRoom
SelectRoomFromList

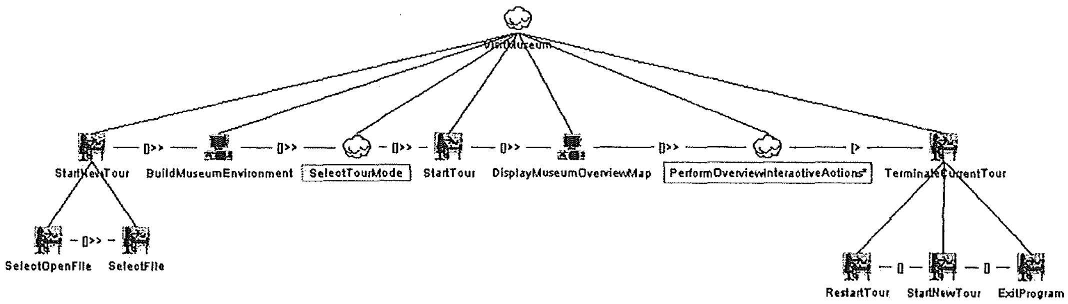

ModifyToSeeList
[]>>
SelectEditToSeeList
EditToSeeList
UpdateEnvironment
[]
AddItemToToSeeList
DeleteItemFromToSeeList
ResetToSeeList(FreeTourMode)

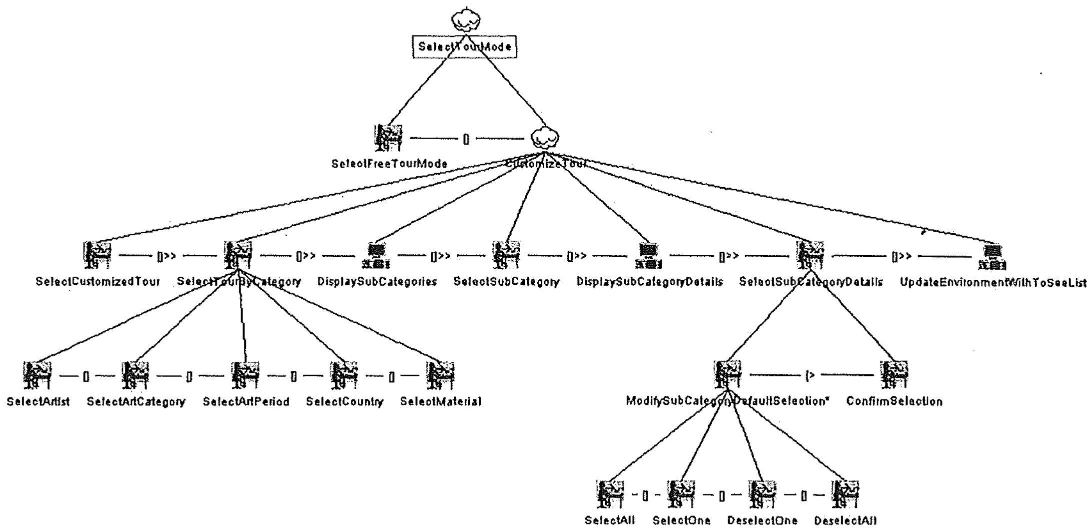

AskForExtraObjectInfo*
[]
AskForAudioComment
AskForMoreTextualComment
ChangeStatusInToSeeList
AskForRelatedArtWork
[]>>
[>
SelectPlayAudioComment
PlayAudioComment
ExitAudioComment
SelectMoreTextualComment
ShowMoreTextualComment
ExitMoreTextualComment

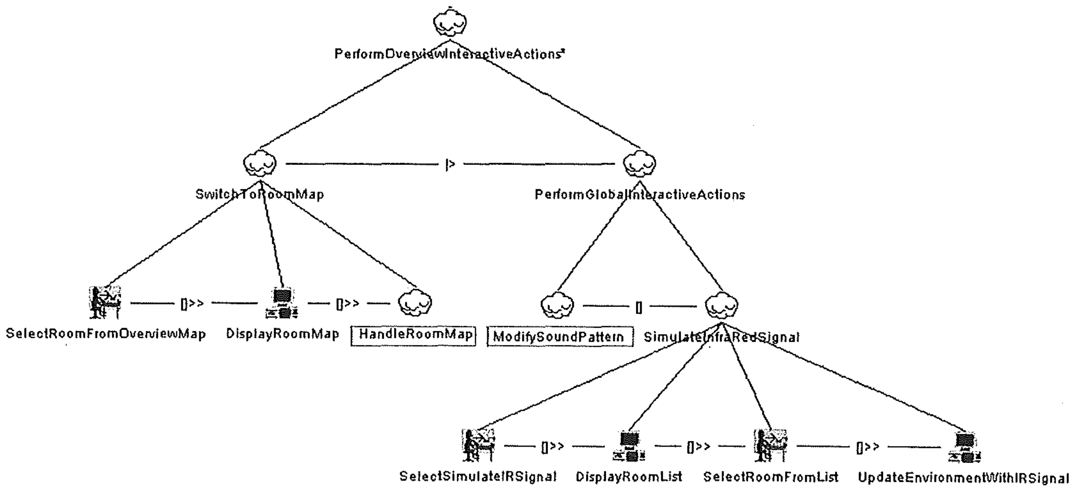

ChangeStatusInToSeeList
SelectChangeStatusInToSeeList
SelectNewStatus
UpdateEnvironment
SetToSee
UnsetToSee

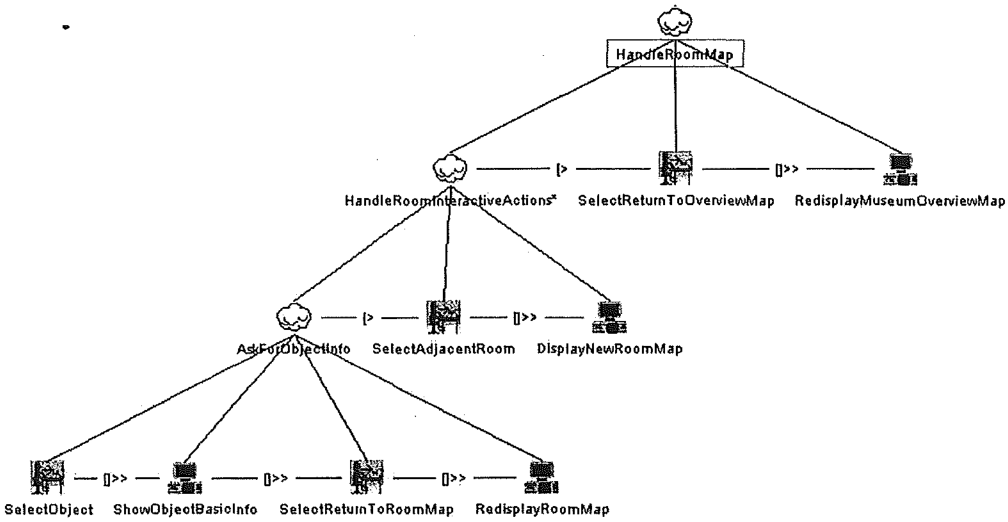

AskForRelatedArtWork
SelectRelatedArtWork
SelectRelationCategory
SelectRelationCategoryDetails
SelectAdaptTourMode
UpdateEnvironment
SelectSameArtist
SelectSameArtCategory
SelectSamePeriod
SelectSameCountry
SelectSameMaterial
SelectFreeEnhancedTourMode
SelectGuidedTourMode

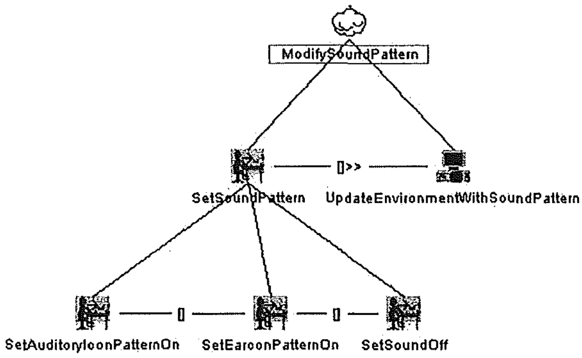
A.2 The Refined ConcurTaskTree Of Our Application








HandleRoomMap
HandleRoomInteractiveActions*
[>
SelectReturnToOverviewMap
[]>>
RedisplayMuseumOverviewMap
AskForObjectInfo
[>
SelectAdjacentRoom
[]>>
DisplayNewRoomMap
SelectObject
[]>>
ShowObjectBasicInfo
[]>>
SelectReturnToRoomMap
[]>>
RedisplayRoomMap


ModifySoundPattern
SetSoundPattern
[]>>
UpdateEnvironmentWithSoundPattern
SetAuditoryIconPatternOn
[]
SetEarconPatternOn
[]
SetSoundOff

Appendix B : Environment Specification Examples

The following pages give two environment specification examples. The input text file and screenshots of the corresponding results will be provided. The first example is a small environment composed of a corridor and two rooms. The second is the complete specification of the fourth floor of the Computer Institute, which was used for the evaluation of the application.

B.1 Specification File Of A Small And Simple Environment

```
#newroom
#code= codeHall2
#name= Hall 2
#type= Hall
#numberofcoordinates=12
#coordinates=
185,240
150,240
150,195
170,185
170,155
45,155
45,145
175,145
175,0
190,0
190,210
185,210

#numberofconnexions=17

#connexioncode= codeHall1
#coordinates=
185,220
185,235

#connexioncode= codeHall3
#coordinates=
45,147
45,153

#connexioncode= codeOffice11
#coordinates=
155,240
165,240

#connexioncode= codeOffice11b
#coordinates=
175,240
180,240

#connexioncode= codeOffice15
#coordinates=
190,185
190,190

#connexioncode= codeOffice16
#coordinates=
190,170
190,175

#connexioncode= codeOffice17
#coordinates=
190,125
190,130

#connexioncode= codeOffice18
#coordinates=
190,110
190,115

#connexioncode= code190
```

```
#coordinates=
190,65
190,70

#connexioncode= codeOffice20
#coordinates=
190,50
190,55

#connexioncode= codeOffice21
#coordinates=
190,10
190,15

#connexioncode= codeOfficeIAT
#coordinates=
65,155
75,155

#connexioncode= codeOfficeIEI
#coordinates=
65,145
75,145

#connexioncode= codeOfficeCED1
#coordinates=
165,145
175,145

#connexioncode= codeOfficeCED2
#coordinates=
175,70
175,75

#connexioncode= codeOfficeCED3
#coordinates=
175,30
175,35

#connexioncode= codeOfficeCED4
#coordinates=
175,5
175,15

#numberofartobjects=0

#newroom
#code= code190
#name= Room 19
#type= Exhibit Room
#numberofcoordinates= 4
#coordinates=
190,90
190,60
245,60
245,90

#numberofconnexions= 1

#connexioncode= codeHall2
#coordinates=
190,65
190,70

#numberofartobjects= 2
```

```
#code= code19A
#name= Art Object A
#attrib1= Author A
#attrib2= Style A
#attrib3= 2002
#attrib4= Country A
#attrib5= Material A
#coordinates=
200,60
230,70

#code= code19B
#name= Art Object B
#attrib1= Author B
#attrib2= Style B
#attrib3= 2002
#attrib4= Country B
#attrib5= Material B
#coordinates=
200,80
230,90

#newroom
#code= codeOffice20
#name= Office 20
#type= Office
#numberofcoordinates=4
#coordinates=
190,60
190,30
245,30
245,60

#numberofconnexions=1

#connexioncode=codeHall2
#coordinates=
190,50
190,55

#numberofartobjects=4

#code= code200
#name= Barbara's Desk
#attrib1= Barbara
#attrib2= Desk
#attrib3= 2002
#attrib4= Italy
#attrib5= Wood and Metal
#coordinates=
230,46
245,55

#code= code201
#name= Francesco's Desk
#attrib1= Francesco
#attrib2= Desk
#attrib3= 2002
#attrib4= Italy
#attrib5= Wood and Metal
#coordinates=
225,35
240,44
```

```
#code= code202
#name= Ravi's Desk
#attrib1= Ravi
#attrib2= Desk
#attrib3= 2002
#attrib4= Belgium
#attrib5= Wood and Metal
#coordinates=
195,35
210,45

#code= code203
#name= Vincenzo's Desk
#attrib1= Vincenzo
#attrib2= Double Desk
#attrib3= 2002
#attrib4= Italy
#attrib5= Wood and Metal
#coordinates=
200,50
225,60
```

B.2 Specification File Of The Fourth Floor Of The Computer Science Institute

```
//Couloir Principal==========================================

#newroom
#code= Couloir0
#name= Couloir Principal
#type= Couloir
#numberofcoordinates= 21
#coordinates=
216,0
236,0
236,16
236,12
244,12
244,15
236,23
236,28
168,28
168,43
226,43
226,28
236,28
236,45
244,53
244,56
236,56
236,53
141,53
141,17
216,17
#numberofconnexions= 19

#connexioncode= 402
#coordinates=
238,12
242,12

#connexioncode= 404
#coordinates=
238,56
242,56

#connexioncode= 405
#coordinates=
206,53
210,53

#connexioncode= 406
#coordinates=
210,28
214,28

#connexioncode= 406
#coordinates=
210,43
214,43

#connexioncode= 407
#coordinates=
200,43
204,43
```

```
#connexioncode= 408
#coordinates=
191,53
195,53

#connexioncode= 409
#coordinates=
176,53
180,53

#connexioncode= 410
#coordinates=
161,53
165,53

#connexioncode= 411
#coordinates=
155,53
159,53

#connexioncode= 414
#coordinates=
147,17
151,17

#connexioncode= 415
#coordinates=
163,17
167,17

#connexioncode= 416
#coordinates=
169,28
173,28

#connexioncode= 417
#coordinates=
180,17
184,17

#connexioncode= 418
#coordinates=
195,17
199,17

#connexioncode= 419
#coordinates=
194,28
198,28

#connexioncode= 420
#coordinates=
201,28
205,28

#connexioncode= 421
#coordinates=
211,17
215,17

#connexioncode= Couloir2
#coordinates=
141,24
141,28
```

```
#numberofartobjects= 0

//=====================================================================

#newroom
#code= 401
#name= 401 Remise
#type= Office
#numberofcoordinates= 4
#coordinates=
236,23
244,15
244,26
236,26
#numberofconnexions= 0
#numberofartobjects= 0

//=====================================================================

#newroom
#code= 402
#name= 402 Salle Academique
#type= Office
#numberofcoordinates= 7
#coordinates=
236,0
273,0
273,23
244,23
244,12
236,12
236,16
#numberofconnexions= 1

#connexioncode= Couloir0
#coordinates=
238,12
242,12

#numberofartobjects= 1

#code= table-reunion
#name= Table de Reunion
#attrib1= Table de Reunion
#attrib2= Objet Utilitaire
#attrib3= Utilitaire
#attrib4= None
#attrib5= No comment
#coordinates=
250,10
260,15

//=====================================================================

#newroom
#code= 403
#name= 403 Kitchenette
#type= Kitchenette
#numberofcoordinates= 4
#coordinates=
236,26
244,26
244,53
236,45
#numberofconnexions= 1
```

```
#connexioncode= 404
#coordinates=
244,27
244,31

#numberofartobjects= 0

//=====================================================================

#newroom
#code= 404
#name= 404 Salle Academique
#type= Office
#numberofcoordinates= 7
#coordinates=
236,56
236,70
273,70
273,23
244,23
244,60
244,56
#numberofconnexions= 2

#connexioncode= 403
#coordinates=
244,27
244,31

#connexioncode= Couloir0
#coordinates=
238,56
242,56

#numberofartobjects= 1

#code= projecteur-reunion
#name= Projecteur
#attrib1= Projecteur
#attrib2= Objet Utilitaire
#attrib3= Utilitaire
#attrib4= None
#attrib5= No comment
#coordinates=
250,60
260,65

//=====================================================================

#newroom
#code= 405
#name= 405
#type= Office
#numberofcoordinates= 4
#coordinates=
205,53
236,53
236,70
205,70
#numberofconnexions= 1

#connexioncode= Couloir0
#coordinates=
206,53
```

```
210,53 .

#numberofartobjects= 1

#code= F.Bodart
#name= Bureau de F.Bodart
#attrib1= Bodart F.
#attrib2= Professeur
#attrib3= Ingenierie des systemes d'information
#attrib4= Vesale
#attrib5= No comment
#coordinates=
210,63
220,68

//==========================================================================

#newroom
#code= 406
#name= 406 Archives
#type= Office
#numberofcoordinates= 4
#coordinates=
205,28
226,28
226,43
205,43
#numberofconnexions= 2

#connexioncode= Couloir0
#coordinates=
210,28
214,28

#connexioncode= Couloir0
#coordinates=
210,43
214,43

#numberofartobjects= 0

//==========================================================================

#newroom
#code= 407
#name= 407
#type= Office
#numberofcoordinates= 4
#coordinates=
199,37
205,37
205,43
199,43
#numberofconnexions= 1

#connexioncode= Couloir0
#coordinates=
200,43
204,43

#numberofartobjects= 0


//==========================================================================
```

```
#newroom
#code= 408
#name= 408 Secretariat de Direction
#type= Office
#numberofcoordinates= 4
#coordinates=
190,53
205,53
205,70
190,70
#numberofconnexions= 1

#connexioncode= Couloir0
#coordinates=
191,53
195,53

#numberofartobjects= 1

#code= B.Di Guardia
#name= Bureau de B.Di Guardia
#attrib1= Di Guardia B.
#attrib2= Secretaire
#attrib3= Secretariat de Direction
#attrib4= None
#attrib5= Secretariat
#coordinates=
192,63
202,68

//=============================================================

#newroom
#code= 409
#name= 409
#type= Office
#numberofcoordinates= 4
#coordinates=
175,53
190,53
190,70
175,70
#numberofconnexions= 1

#connexioncode= Couloir0
#coordinates=
176,53
180,53

#numberofartobjects= 1

#code= P.Y.Schobbens
#name= Bureau de P.Y.Schobbens
#attrib1= Schobbens P.Y.
#attrib2= Professeur
#attrib3= Semantique, logique et calcul
#attrib4= None
#attrib5= No comment
#coordinates=
177,63
187,68

//=============================================================

#newroom
```

```
#code= 410
#name= 410
#type= Office
#numberofcoordinates= 4
#coordinates=
160,53
175,53
175,70
160,70
#numberofconnexions= 1

#connexioncode= Couloir0
#coordinates=
161,53
165,53

#numberofartobjects= 1

#code= M.Noirhomme
#name= Bureau de M.Noirhomme
#attrib1= Noirhomme M.
#attrib2= Professeur
#attrib3= IHM
#attrib4= None
#attrib5= No comment
#coordinates=
162,63
172,68

//======================================================================

#newroom
#code= 411
#name= 411
#type= Office
#numberofcoordinates= 6
#coordinates=
130,70
130,58
141,58
141,53
160,53
160,70
#numberofconnexions= 1

#connexioncode= Couloir0
#coordinates=
155,53
159,53

#numberofartobjects= 3

#code= A.de Baenst
#name= Bureau de A.de Baenst
#attrib1= De Baenst A.
#attrib2= Chercheur
#attrib3= IHM
#attrib4= None
#attrib5= No comment
#coordinates=
143,55
153,60

#code= F.Collot
#name= Bureau de F.Collot
```

```
#attrib1= Collot F.
#attrib2= Chercheur
#attrib3= IHM
#attrib4= None
#attrib5= No comment
#coordinates=
148,63
158,68

#code= A.Rousseau
#name= Bureau de A.Rousseau
#attrib1= Rousseau A.
#attrib2= Chercheur
#attrib3= IHM
#attrib4= None
#attrib5= No comment
#coordinates=
132,63
142,68

//===================================================================

#newroom
#code= 414
#name= 414
#type= Office
#numberofcoordinates= 6
#coordinates=
130,0
151,0
151,17
141,17
141,10
130,10
#numberofconnexions= 1

#connexioncode= Couloir0
#coordinates=
147,17
151,17

#numberofartobjects= 2

#code= Ch.le Roy
#name= Bureau de Ch.le Roy
#attrib1= Le Roy Ch.
#attrib2= Gestionnaire
#attrib3= Gestion des Contrats
#attrib4= None
#attrib5= Gestion
#coordinates=
132,2
142,7

#code= L.Oger
#name= Bureau de L.Oger
#attrib1= Oger L.
#attrib2= Gestionnaire
#attrib3= Gestion des Contrats
#attrib4= None
#attrib5= Gestion
#coordinates=
145,2
150,12
```

```
//=================================================================

#newroom
#code= 415
#name= 415 Direction
#type= Office
#numberofcoordinates= 4
#coordinates=
151,0
168,0
168,17
151,17
#numberofconnexions= 1

#connexioncode= Couloir0
#coordinates=
163,17
167,17

#numberofartobjects= 1

#code= J.Fichefet
#name= Bureau de J.Fichefet
#attrib1= Fichefet J.
#attrib2= Professeur
#attrib3= Aide a la decision
#attrib4= None
#attrib5= Direction
#coordinates=
153,2
163,7

//=================================================================

#newroom
#code= 416
#name= 416 WC Messieurs
#type= Office
#numberofcoordinates= 4
#coordinates=
168,28
184,28
184,43
168,43
#numberofconnexions= 1

#connexioncode= Couloir0
#coordinates=
169,28
173,28

#numberofartobjects= 1

#code= WC Messieurs
#name= WC Messieurs
#attrib1= WC Messieurs
#attrib2= Objet Utilitaire
#attrib3= Utilitaire
#attrib4= None
#attrib5= No comment
#coordinates=
170,39
182,42

//=================================================================
```

```
#newroom
#code= 417
#name= 417
#type= Office
#numberofcoordinates= 4
#coordinates=
168,0
184,0
184,17
168,17
#numberofconnexions= 1

#connexioncode= Couloir0
#coordinates=
179,17
183,17

#numberofartobjects= 1

#code= Nouveau Professeur 2
#name= Bureau du Nouveau Professeur
#attrib1= Nouveau Professeur
#attrib2= Professeur
#attrib3= A determiner
#attrib4= A determiner
#attrib5= No comment
#coordinates=
170,2
180,7

//=========================================================================

#newroom
#code= 418
#name= 418 Direction ff
#type= Office
#numberofcoordinates= 4
#coordinates=
184,0
200,0
200,17
184,17
#numberofconnexions= 1

#connexioncode= Couloir0
#coordinates=
195,17
199,17

#numberofartobjects= 1

#code= C.Lobet
#name= Bureau de C.Lobet
#attrib1= Lobet C.
#attrib2= Professeur
#attrib3= Info, Organisation & Societe
#attrib4= None
#attrib5= Direction ff
#coordinates=
186,2
196,7

//=========================================================================
```

```
#newroom
#code= 419
#name= 419 WC Dames
#type= Office
#numberofcoordinates= 4
#coordinates=
184,28
199,28
199,43
184,43
#numberofconnexions= 1

#connexioncode= Couloir0
#coordinates=
194,28
198,28

#numberofartobjects= 1

#code= WC Dames
#name= WC Dames
#attrib1= WC Dames
#attrib2= Objet Utilitaire
#attrib3= Utilitaire
#attrib4= None
#attrib5= No comment
#coordinates=
186,30
194,41

//=====================================================================

#newroom
#code= 420
#name= 420 WC Handicapes
#type= Office
#numberofcoordinates= 4
#coordinates=
199,28
205,28
205,37
199,37
#numberofconnexions= 1

#connexioncode= Couloir0
#coordinates=
201,28
205,28

#numberofartobjects= 0

//=====================================================================

#newroom
#code= 421
#name= 421
#type= Office
#numberofcoordinates= 4
#coordinates=
200,0
216,0
216,17
200,17
#numberofconnexions= 1
```

```
#connexioncode= Couloir0
#coordinates=
211,17
215,17

#numberofartobjects= 1

#code= N.Habra
#name= Bureau de N.Habra
#attrib1= Habra N.
#attrib2= Professeur
#attrib3= Ingenierie des systemes d'information
#attrib4= None
#attrib5= No comment
#coordinates=
202,2
212,7

//=====================================================================

#newroom
#code= Couloir2
#name= Couloir Intermediaire
#type= Couloir
#numberofcoordinates= 4
#coordinates=
130,21
141,21
141,31
130,31
#numberofconnexions= 3

#connexioncode= Couloir0
#coordinates=
141,24
141,28

#connexioncode= Couloir1
#coordinates=
130,24
130,28

#connexioncode= 314
#coordinates=
132,31
136,31

#numberofartobjects= 0

//=====================================================================

#newroom
#code= Piece001
#name= Piece Vide
#type= Couloir
#numberofcoordinates= 4
#coordinates=
130,10
141,10
141,21
130,21
#numberofconnexions= 0
#numberofartobjects= 0

//=====================================================================
```

```
#newroom
#code= 314
#name= 314 Escalier Secondaire
#type= Escalier
#numberofcoordinates= 4
#coordinates=
130,31
141,31
141,58
130,58
#numberofconnexions= 1

#connexioncode= Couloir2
#coordinates=
132,31
136,31

#numberofartobjects= 0

//=============================================================================

#newroom
#code= Couloir1
#name= Couloir Secondaire
#type= Couloir
#numberofcoordinates= 4
#coordinates=
17,21
130,21
130,31
17,31

#numberofconnexions= 13

#connexioncode= Musee
#coordinates=
18,24
18,28

#connexioncode= 426
#coordinates=
111,31
115,31

#connexioncode= 428
#coordinates=
79,31
83,31

#connexioncode= 432
#coordinates=
46,31
50,31

#connexioncode= 435
#coordinates=
19,31
23,31

#connexioncode= 434
#coordinates=
19,21
23,21
```

```
#connexioncode= 433
#coordinates=
41,21
45,21

#connexioncode= 431
#coordinates=
56,21
60,21

#connexioncode= 430
#coordinates=
72,21
76,21

#connexioncode= 429
#coordinates=
89,21
93,21

#connexioncode= 427
#coordinates=
101,21
105,21

#connexioncode= 425
#coordinates=
121,21
125,21

#connexioncode= Couloir2
#coordinates=
130,24
130,28

#numberofartobjects= 0

//=========================================================================

#newroom
#code= Musee
#name= Musee
#type= Musee
#numberofcoordinates= 4
#coordinates=
0,21
18,21
18,31
0,31

#numberofconnexions= 1

#connexioncode= Couloir1
#coordinates=
18,24
18,28

#numberofartobjects= 1

#code= Musee1
#name= Objet du Musee
#attrib1= Objet du Musee
#attrib2= Objet
#attrib3= Patrimoine
#attrib4= None
```

```
#attrib5= No comment
#coordinates=
5,25
10,28

//==========================================================================

#newroom
#code= 426
#name= 426
#type= Office
#numberofcoordinates= 4
#coordinates=
98,31
130,31
130,70
98,70

#numberofconnexions= 1

#connexioncode= Couloir1
#coordinates=
111,31
115,31

#numberofartobjects= 4

#code= M.Ndaye
#name= Bureau de M.Ndaye
#attrib1= Ndaye M.
#attrib2= Chercheur
#attrib3= IHM
#attrib4= MAI
#attrib5= No comment
#coordinates=
100,33
110,48

#code= C.Pirotte
#name= Bureau de C.Pirotte
#attrib1= Pirotte C.
#attrib2= Chercheur
#attrib3= IHM
#attrib4= MAI
#attrib5= No comment
#coordinates=
118,33
128,48

#code= A.Nahimana
#name= Bureau de A.Nahimana
#attrib1= Nahimana A.
#attrib2= Chercheur
#attrib3= IHM
#attrib4= ASSO
#attrib5= No comment
#coordinates=
100,50
110,65

#code= E.Leconte
#name= Bureau de E.Leconte
#attrib1= Leconte E.
#attrib2= Chercheur
#attrib3= IHM
```

```
#attrib4= Encadrement
#attrib5= No comment
#coordinates=
118,50
128,65

#newroom
#code= 428
#name= 428
#type= Office
#numberofcoordinates= 4
#coordinates=
65,31
98,31
98,70
65,70

#numberofconnexions= 1

#connexioncode= Couloir1
#coordinates=
79,31
83,31

#numberofartobjects= 4

#code= V.Dumont
#name= Bureau de V.Dumont
#attrib1= Dumont V.
#attrib2= Chercheur
#attrib3= Info, Organisation & Societe
#attrib4= CITA
#attrib5= No comment
#coordinates=
67,33
77,48

#code= J.M.Jacques
#name= Bureau de J.M.Jacques
#attrib1= Jacques J.M.
#attrib2= Chercheur
#attrib3= Info, Organisation & Societe
#attrib4= CITA
#attrib5= No comment
#coordinates=
86,33
96,48

#code= E.Koch
#name= Bureau de E.Koch
#attrib1= Koch E.
#attrib2= Chercheur
#attrib3= Info, Organisation & Societe
#attrib4= CITA
#attrib5= No comment
#coordinates=
67,50
77,65

#code= J.Tirtiaux
#name= Bureau de J.Tirtiaux
#attrib1= Tirtiaux J.
#attrib2= Chercheur
#attrib3= Info, Organisation & Societe
#attrib4= CITA
```

```
#attrib5= No comment
#coordinates=
86,50
96,65

//==========================================================================

#newroom
#code= 432
#name= 432
#type= Office
#numberofcoordinates= 4
#coordinates=
33,31
65,31
65,70
33,70

#numberofconnexions= 1

#connexioncode= Couloir1
#coordinates=
46,31
50,31

#numberofartobjects= 5

#code= L.Bahali
#name= Bureau de L.Bahali
#attrib1= Bahali L.
#attrib2= Chercheur
#attrib3= Aide a la decision
#attrib4= EMIM
#attrib5= No comment
#coordinates=
35,33
45,48

#code= B.Georges
#name= Bureau de B.Georges
#attrib1= Georges B.
#attrib2= Chercheur
#attrib3= Aide a la decision
#attrib4= EMIM
#attrib5= No comment
#coordinates=
53,33
63,48

#code= C.Hayez
#name= Bureau de C.Hayez
#attrib1= Hayez C.
#attrib2= Chercheur
#attrib3= Aide a la decision
#attrib4= EMIM
#attrib5= No comment
#coordinates=
35,50
45,65

#code= H.Meurisse
#name= Bureau de H.Meurisse
#attrib1= Meurisse H.
#attrib2= Chercheur
#attrib3= Aide a la decision
```

```
#attrib4= EMIM
#attrib5= No comment
#coordinates=
53,50
63,65

#code= A.Bereikdar
#name= Bureau de A.Bereikdar
#attrib1= Bereikdar A.
#attrib2= Chercheur
#attrib3= Aide a la decision
#attrib4= EMIM
#attrib5= No comment
#coordinates=
46,66
51,68

//====================================================================

#newroom
#code= 435
#name= 435
#type= Office
#numberofcoordinates= 4
#coordinates=
0,31
33,31
33,70
0,70

#numberofconnexions= 1

#connexioncode= Couloir1
#coordinates=
19,31
23,31

#numberofartobjects= 5

#code= A.F.Brogneau
#name= Bureau de A.F.Brogneau
#attrib1= Brogneau A.F.
#attrib2= Chercheur
#attrib3= Ingenierie des bases de donnees
#attrib4= DB-Main
#attrib5= No comment
#coordinates=
2,33
12,48

#code= V.Detienne
#name= Bureau de V.Detienne
#attrib1= Detienne V.
#attrib2= Chercheur
#attrib3= Ingenierie des bases de donnees
#attrib4= DB-Main
#attrib5= No comment
#coordinates=
25,33
31,48

#code= J.Henrard
#name= Bureau de J.Henrard
#attrib1= Henrard J.
#attrib2= Chercheur
```

```
#attrib3= Ingenierie des bases de donnees
#attrib4= DB-Main
#attrib5= No comment
#coordinates=
2,50
12,65

#code= J.M.Hick
#name= Bureau de J.M.Hick
#attrib1= Hick J.M.
#attrib2= Chercheur
#attrib3= Ingenierie des bases de donnees
#attrib4= DB-Main
#attrib5= No comment
#coordinates=
25,50
31,65

#code= D.Roland
#name= Bureau de D.Roland
#attrib1= Roland D.
#attrib2= Chercheur
#attrib3= Ingenierie des bases de donnees
#attrib4= DB-Main
#attrib5= No comment
#coordinates=
13,66
20,68

//==========================================================================

#newroom
#code= 434
#name= 434 Salle Brunin
#type= Salle de Reunion
#numberofcoordinates= 4
#coordinates=
0,0
33,0
33,21
0,21

#numberofconnexions= 1

#connexioncode= Couloir1
#coordinates=
19,21
23,21

#numberofartobjects= 0

//==========================================================================

#newroom
#code= 433
#name= 433
#type= Office
#numberofcoordinates= 4
#coordinates=
33,0
49,0
49,21
33,21

#numberofconnexions= 1
```

```
#connexioncode= Couloir1
#coordinates=
41,21
45,21

#numberofartobjects= 1

#code= J.L.Hainaut
#name= Bureau de J.L.Hainaut
#attrib1= Hainaut J.L.
#attrib2= Professeur
#attrib3= Ingenierie des bases de donnees
#attrib4= DB-Main
#attrib5= No comment
#coordinates=
36,2
46,7

//==========================================================================

#newroom
#code= 431
#name= 431
#type= Office
#numberofcoordinates= 4
#coordinates=
49,0
65,0
65,21
49,21

#numberofconnexions= 1

#connexioncode= Couloir1
#coordinates=
56,21
60,21

#numberofartobjects= 1

#code= Ph.Thiran
#name= Bureau de Ph.Thiran
#attrib1= Thiran Ph.
#attrib2= Chercheur
#attrib3= Ingenierie des bases de donnees
#attrib4= DB-Main
#attrib5= No comment
#coordinates=
52,2
63,7

//==========================================================================

#newroom
#code= 430
#name= 430
#type= Office
#numberofcoordinates= 4
#coordinates=
65,0
81,0
81,21
65,21
```

```
#numberofconnexions= 1

#connexioncode= Couloir1
#coordinates=
72,21
76,21

#numberofartobjects= 1

#code= Nouveau Professeur 1
#name= Bureau du Nouveau Professeur
#attrib1= Nouveau Professeur
#attrib2= Professeur
#attrib3= A determiner
#attrib4= A determiner
#attrib5= No comment
#coordinates=
68,2
78,7

//=====================================================================

#newroom
#code= 429
#name= 429
#type= Office
#numberofcoordinates= 4
#coordinates=
81,0
98,0
98,21
81,21

#numberofconnexions= 1

#connexioncode= Couloir1
#coordinates=
89,21
93,21

#numberofartobjects= 1

#code= F.Malaya
#name= Bureau de F.Malaya
#attrib1= Malaya F.
#attrib2= Professeur
#attrib3= A determiner
#attrib4= A determiner
#attrib5= No comment
#coordinates=
84,2
95,7

//=====================================================================

#newroom
#code= 427
#name= 427
#type= Office
#numberofcoordinates= 4
#coordinates=
98,0
114,0
114,21
98,21
```

```
#numberofconnexions= 1

#connexioncode= Couloir1
#coordinates=
105,21
109,21

#numberofartobjects= 1

#code= V.Englebert
#name= Bureau de V.Englebert
#attrib1= Englebert V.
#attrib2= Professeur
#attrib3= Ingenierie des systemes d'information
#attrib4= None
#attrib5= No comment
#coordinates=
101,2
111,7

//=========================================================================

#newroom
#code= 425
#name= 425
#type= Office
#numberofcoordinates= 4
#coordinates=
114,0
130,0
130,21
114,21

#numberofconnexions= 1

#connexioncode= Couloir1
#coordinates=
121,21
125,21

#numberofartobjects= 1

#code= R.Cotet
#name= Bureau de R.Cotet
#attrib1= Cotet R.
#attrib2= Administrateur
#attrib3= Reseau et securite
#attrib4= None
#attrib5= No comment
#coordinates=
117,2
127,7
```

Appendix C : Questionnaire Answers

The following pages show the charts containing the answers to the interface evaluation questionnaire. As stated in Chapter 9, we received 28 answers for the evaluation, with 14 persons testing the Auditory Icons for the two first scenarios, and as many persons testing the Earcon first.

Part 1 : Personnal Information

Name	Quentin Dallons
Age	24
Gender	Male
Title/Position	Student
Field of Activity/Study	Computer Science
E-mail Address	qdallons@info.fundp.ac.be

Part 2 : Previous Experience in Sound and Mobile Devices

Musical Background (formation, etc.)	None	Basic	**Average**	Upper	Expert
Previous Use of Mobile Phone	Never	Rare	Occasionnal	Often	**Daily Basis**
Previous Use of PDA (or similar)	Never	Rare	Occasionnal	Often	**Daily Basis**
Previous Use of Electronic Games	Never	Rare	Occasionnal	Often	**Daily Basis**
If it is the case, are there some of them using maps?	Quake like, ...				

Part 3 : Scenarios

Sound Pattern	Auditory Icons	Sound Pattern	Earcons
Scenario 1 (in sec)	11	Scenario 3a (in sec)	24
Scenario 2a (in sec)	8	Scenario 3b (in sec)	12
Scenario 2b (in sec)	8	Scenario 3c (in sec)	8

Part 4 : Interface Overall Evaluation

Lisibility	Difficult	Not so good	Average	**Good**	Excellent
Map Navigation	Difficult	Not so good	Average	Good	**Excellent**
Menus	Difficult	Not so good	Average	**Good**	Excellent

Part 5 : Sound Pattern Evaluation

Auditory Icons	Horrible	**Not so good**	Average	Good	Excellent
Earcons	Horrible	Not so good	**Average**	Good	Excellent
Sound Utility	**Useless**	Not so usefull	Neutral	Usefull	Essential

Part 6 : General Comments

- Pour la lisibilité, il y a un problème de reflets, mais c'est inhérent au PDA.
- Pour les auditory icons, on ne distingue pas trop ce que représentent les sons à cause de la qualité du sample
- Pour ce qui est du son, cela ne sert pas dans la navigation.

Part 1 : Personnal Information

Name	Michael De Gols
Age	23
Gender	Male
Title/Position	Student
Field of Activity/Study	Computer Science
E-mail Address	mdegols@info.fundp.ac.be

Part 2 : Previous Experience in Sound and Mobile Devices

Musical Background (formation, etc.)	**None**	Basic	Average	Upper	Expert
Previous Use of Mobile Phone	Never	Rare	Occasionnal	Often	**Daily Basis**
Previous Use of PDA (or similar)	**Never**	Rare	Occasionnal	Often	Daily Basis
Previous Use of Electronic Games	Never	Rare	Occasionnal	Often	**Daily Basis**
If it is the case, are there some of them using maps?	Counter Strike, Role Playing Games ...				

Part 3 : Scenarios

Sound Pattern	Earcons	Sound Pattern	Auditory Icons
Scenario 1 (in sec)	10	Scenario 3a (in sec)	21
Scenario 2a (in sec)	15	Scenario 3b (in sec)	17
Scenario 2b (in sec)	9	Scenario 3c (in sec)	7

Part 4 : Interface Overall Evaluation

Lisibility	Difficult	Not so good	Average	Good	**Excellent**
Map Navigation	Difficult	Not so good	Average	Good	**Excellent**
Menus	Difficult	**Not so good**	Average	Good	Excellent

Part 5 : Sound Pattern Evaluation

Auditory Icons	Horrible	Not so good	Average	**Good**	Excellent
Earcons	Horrible	**Not so good**	Average	Good	Excellent
Sound Utility	Useless	**Not so usefull**	Neutral	Usefull	Essential

Part 6 : General Comments

- Du point de vue de la lisibilité, c'est clair, on a juste les informations nécessaires.
- Le classement est mal fait pour les reccherches, il est peu intuitif.
- Pour les auditory icons, le bruit de la porte est bien, mais pour les objets, le son choisi est un peu long. Un bruit de tiroir aurait été mieux.
- Les earcons, c'est juste énervant.
- Pour la navigation dans le plan, je m'attendais à revenir au plan général lorsque je quittais une pièce. Il faudrait des niveaux de zoom pour mieux se repérer, et des petits messages d'information pour mieux identifier les connexions. Il manque des boutons pour revenir en arrière (sur l'écran d'information des objets notamment).

Part 1 : Personnal Information

Name	Frédéric Foulard
Age	23
Gender	Male
Title/Position	Student
Field of Activity/Study	Computer Science
E-mail Address	ffoulard@hotmail.com

Part 2 : Previous Experience in Sound and Mobile Devices

Musical Background (formation, etc.)	None	Basic	Average	Upper	Expert
Previous Use of Mobile Phone	Never	Rare	Occasionnal	Often	Daily Basis
Previous Use of PDA (or similar)	Never	Rare	Occasionnal	Often	Daily Basis
Previous Use of Electronic Games	Never	Rare	Occasionnal	Often	Daily Basis
If it is the case, are there some of them using maps?	Half Life, Role Playing Games…				

Part 3 : Scenarios

Sound Pattern	Auditory Icons	Sound Pattern	Earcons
Scenario 1 (in sec)	15	Scenario 3a (in sec)	33
Scenario 2a (in sec)	10	Scenario 3b (in sec)	25
Scenario 2b (in sec)	11	Scenario 3c (in sec)	12

Part 4 : Interface Overall Evaluation

Lisibility	Difficult	Not so good	Average	Good	Excellent
Map Navigation	Difficult	Not so good	Average	Good	Excellent
Menus	Difficult	Not so good	Average	Good	Excellent

Part 5 : Sound Pattern Evaluation

Auditory Icons	Horrible	Not so good	Average	Good	Excellent
Earcons	Horrible	Not so good	Average	Good	Excellent
Sound Utility	Useless	Not so usefull	Neutral	Usefull	Essential

Part 6 : General Comments

- Dans les recherches, le choix des attributs n'est pas très parlant (« title » par exemple).
- Pour le scénario 3a, je m'attendais à pouvoir faire une recherche par attribut pour le caractère « nouveau » des profs.
- Dans la navigation, il y a des problèmes pour revenir en arrière.
- J'aurais préféré voir un zoom du plan plutôt que la pièce toute seule.
- Le bruit de la porte est bien, mais le bruit des objets n'est pas top, c'est énervant.

Part 1 : Personnal Information

Name	Benoit Loffer
Age	23
Gender	Male
Title/Position	Student
Field of Activity/Study	Computer Science
E-mail Address	bloffet@info.fundp.ac.be

Part 2 : Previous Experience in Sound and Mobile Devices

Musical Background (formation, etc.)	None	Basic	Average	Upper	Expert
Previous Use of Mobile Phone	Never	Rare	Occasionnal	Often	Daily Basis
Previous Use of PDA (or similar)	Never	Rare	Occasionnal	Often	Daily Basis
Previous Use of Electronic Games	Never	Rare	Occasionnal	Often	Daily Basis
If it is the case, are there some of them using maps?	Diablo, Splinter Cell, Baldur's Gate…				

Part 3 : Scenarios

Sound Pattern	Earcons	Sound Pattern	Auditory Icons
Scenario 1 (in sec)	11	Scenario 3a (in sec)	27
Scenario 2a (in sec)	11	Scenario 3b (in sec)	17
Scenario 2b (in sec)	7	Scenario 3c (in sec)	45

Part 4 : Interface Overall Evaluation

Lisibility	Difficult	Not so good	Average	Good	Excellent
Map Navigation	Difficult	Not so good	Average	Good	Excellent
Menus	Difficult	Not so good	Average	Good	Excellent

Part 5 : Sound Pattern Evaluation

Auditory Icons	Horrible	Not so good	Average	Good	Excellent
Earcons	Horrible	Not so good	Average	Good	Excellent
Sound Utility	Useless	Not so usefull	Neutral	Usefull	Essential

Part 6 : General Comments

- Les classements sont un peu difficiles : les domaines de cours sont mélangés avec le secrétariat, les utilitaires… Il aurait mieux valu les séparer.
- Les auditory icons sont plus proches de la réalité, les bruits sont plus sympas. Le bruit des porte est bien, c'est clair.
- Les earcons me cassent les oreilles, j'aime vraiment pas !
- On manque un peu de repères dans le plan : ça aurait été bien de garder le plan autour de la pièce pour pouvoir mieux se situer.

Part 1 : Personnal Information

Name	François Gruselin
Age	24
Gender	Male
Title/Position	Student
Field of Activity/Study	Computer Science
E-mail Address	fgruseli@info.fundp.ac.be

Part 2 : Previous Experience in Sound and Mobile Devices

Musical Background (formation, etc.)	None	Basic	Average	Upper	Expert
Previous Use of Mobile Phone	Never	Rare	Occasionnal	Often	Daily Basis
Previous Use of PDA (or similar)	Never	Rare	Occasionnal	Often	Daily Basis
Previous Use of Electronic Games	Never	Rare	Occasionnal	Often	Daily Basis
If it is the case, are there some of them using maps?	Heroes of Might & Magic, Counter Strike, …				

Part 3 : Scenarios

Sound Pattern	Auditory Icons	Sound Pattern	Earcons
Scenario 1 (in sec)	25	Scenario 3a (in sec)	21
Scenario 2a (in sec)	60	Scenario 3b (in sec)	30
Scenario 2b (in sec)	17	Scenario 3c (in sec)	17

Part 4 : Interface Overall Evaluation

Lisibility	Difficult	Not so good	Average	Good	Excellent
Map Navigation	Difficult	Not so good	Average	Good	Excellent
Menus	Difficult	Not so good	Average	Good	Excellent

Part 5 : Sound Pattern Evaluation

Auditory Icons	Horrible	Not so good	Average	Good	Excellent
Earcons	Horrible	Not so good	Average	Good	Excellent
Sound Utility	Useless	Not so usefull	Neutral	Usefull	Essential

Part 6 : General Comments

- On manque de repères pour switcher de niveau (comme pour retourner au niveau précédent) : un bouton pour revenir au départ aurait été le bienvenu.
- La recherche est peu intuitive.
- Les auditory icons me tapent un peu sur le système (surtout le son pour les objets). Le bruit de la porte n'est pas trop mal.
- Le son n'est pas vraiment utile pour la navigation.

Part 1 : Personnal Information

Name	Stéphane Valk
Age	24
Gender	Male
Title/Position	Assistant, Researcher
Field of Activity/Study	Dynamic Systems (Mathematics)
E-mail Address	sval@math.fundp.ac.be

Part 2 : Previous Experience in Sound and Mobile Devices

Musical Background (formation, etc.)	None	Basic	**Average**	Upper	Expert
Previous Use of Mobile Phone	Never	Rare	Occasionnal	Often	**Daily Basis**
Previous Use of PDA (or similar)	**Never**	Rare	Occasionnal	Often	Daily Basis
Previous Use of Electronic Games	Never	Rare	Occasionnal	**Often**	Daily Basis
If it is the case, are there some of them using maps?	Quake like, …				

Part 3 : Scenarios

Sound Pattern	Earcons	Sound Pattern	Auditory Icons
Scenario 1 (in sec)	16	Scenario 3a (in sec)	11
Scenario 2a (in sec)	8	Scenario 3b (in sec)	30
Scenario 2b (in sec)	8	Scenario 3c (in sec)	11

Part 4 : Interface Overall Evaluation

Lisibility	Difficult	Not so good	Average	Good	**Excellent**
Map Navigation	Difficult	Not so good	**Average**	Good	Excellent
Menus	Difficult	Not so good	Average	Good	**Excellent**

Part 5 : Sound Pattern Evaluation

Auditory Icons	Horrible	Not so good	Average	**Good**	Excellent
Earcons	Horrible	Not so good	**Average**	Good	Excellent
Sound Utility	Useless	Not so usefull	Neutral	**Usefull**	Essential

Part 6 : General Comments

- La navigation est un peu difficile, à cause du stylet et de la petite taille des connexions.
- Ce serait bien de faire tourner les pièces pour les adapter.
- Le bouton Begin a un titre mal choisi.
- Les earcons me font penser à des bruits de vieux PDA !
- Le son permet de confirmer nos actions.

Part 1 : Personnal Information

Name	Caroline Sainvitu
Age	22
Gender	Female
Title/Position	Assistant, Researcher
Field of Activity/Study	Numeric Analysis (Mathematics)
E-mail Address	csai@math.fundp.ac.be

Part 2 : Previous Experience in Sound and Mobile Devices

Musical Background (formation, etc.)	None	Basic	Average	Upper	Expert
Previous Use of Mobile Phone	Never	Rare	Occasionnal	Often	Daily Basis
Previous Use of PDA (or similar)	Never	Rare	Occasionnal	Often	Daily Basis
Previous Use of Electronic Games	Never	Rare	Occasionnal	Often	Daily Basis
If it is the case, are there some of them using maps?	Age of Empire, …				

Part 3 : Scenarios

Sound Pattern	Auditory Icons	Sound Pattern	Earcons
Scenario 1 (in sec)	26	Scenario 3a (in sec)	45
Scenario 2a (in sec)	20	Scenario 3b (in sec)	23
Scenario 2b (in sec)	10	Scenario 3c (in sec)	18

Part 4 : Interface Overall Evaluation

Lisibility	Difficult	Not so good	Average	Good	Excellent
Map Navigation	Difficult	Not so good	Average	Good	Excellent
Menus	Difficult	Not so good	Average	Good	Excellent

Part 5 : Sound Pattern Evaluation

Auditory Icons	Horrible	Not so good	Average	Good	Excellent
Earcons	Horrible	Not so good	Average	Good	Excellent
Sound Utility	Useless	Not so usefull	Neutral	Usefull	Essential

Part 6 : General Comments

- Il manque des boutons pour revenir en arrière.
- Je préfère les auditory icons, mais j'aime bien quand même le bruit pour changer de pièce dans les earcons !

Part 1 : Personnal Information

Name	Johann Mignolet
Age	20
Gender	Male
Title/Position	Student
Field of Activity/Study	Biology
E-mail Address	Inquisitor1@caramail.com

Part 2 : Previous Experience in Sound and Mobile Devices

Musical Background (formation, etc.)	None	Basic	Average	Upper	Expert
Previous Use of Mobile Phone	Never	Rare	Occasionnal	Often	Daily Basis
Previous Use of PDA (or similar)	Never	Rare	Occasionnal	Often	Daily Basis
Previous Use of Electronic Games	Never	Rare	Occasionnal	Often	Daily Basis
If it is the case, are there some of them using maps?	Baphomet, …				

Part 3 : Scenarios

Sound Pattern	Earcons	Sound Pattern	Auditory Icons
Scenario 1 (in sec)	18	Scenario 3a (in sec)	25
Scenario 2a (in sec)	10	Scenario 3b (in sec)	15
Scenario 2b (in sec)	14	Scenario 3c (in sec)	12

Part 4 : Interface Overall Evaluation

Lisibility	Difficult	Not so good	Average	Good	Excellent
Map Navigation	Difficult	Not so good	Average	Good	Excellent
Menus	Difficult	Not so good	Average	Good	Excellent

Part 5 : Sound Pattern Evaluation

Auditory Icons	Horrible	Not so good	Average	Good	Excellent
Earcons	Horrible	Not so good	Average	Good	Excellent
Sound Utility	Useless	Not so usefull	Neutral	Usefull	Essential

Part 6 : General Comments

- Les titres des menus ne sont pas assez explicites.
- Il aurait été bien d'utiliser des niveaux de zoom, avec les pièces voisines représentées en grisé.
- On aurait pu utiliser des couleurs différentes pour les toilettes, les escaliers, etc.
- Le son est important, mais il est encore plus important d'avoir une information visuelle sur la pièce.
- Avec les auditory icons, on allie l'image au son, ce qui est mieux.
- Les earcons sont bien aussi, mais… il n'y a pas de correspondance.

Part 1 : Personnal Information

Name	Benoit Donnet
Age	22
Gender	Male
Title/Position	Student
Field of Activity/Study	Computer Science
E-mail Address	bdonnet@info.fundp.ac.be

Part 2 : Previous Experience in Sound and Mobile Devices

Musical Background (formation, etc.)	None	Basic	Average	Upper	**Expert**
Previous Use of Mobile Phone	**Never**	Rare	Occasionnal	Often	Daily Basis
Previous Use of PDA (or similar)	**Never**	Rare	Occasionnal	Often	Daily Basis
Previous Use of Electronic Games	Never	**Rare**	Occasionnal	Often	Daily Basis
If it is the case, are there some of them using maps?	Warcraft, ...				

Part 3 : Scenarios

Sound Pattern	Auditory Icons	Sound Pattern	Earcons
Scenario 1 (in sec)	14	Scenario 3a (in sec)	12
Scenario 2a (in sec)	21	Scenario 3b (in sec)	46
Scenario 2b (in sec)	11	Scenario 3c (in sec)	16

Part 4 : Interface Overall Evaluation

Lisibility	Difficult	Not so good	Average	**Good**	Excellent
Map Navigation	Difficult	Not so good	Average	**Good**	Excellent
Menus	Difficult	Not so good	**Average**	Good	Excellent

Part 5 : Sound Pattern Evaluation

Auditory Icons	**Horrible**	Not so good	Average	Good	Excellent
Earcons	**Horrible**	Not so good	Average	Good	Excellent
Sound Utility	**Useless**	Not so usefull	Neutral	Usefull	Essential

Part 6 : General Comments

- Il y a problème de classement des objets.

Part 1 : Personnal Information

Name	Xavier Cuvellier
Age	22
Gender	Male
Title/Position	Student
Field of Activity/Study	Computer Science
E-mail Address	xcuvelli@info.fundp.ac.be

Part 2 : Previous Experience in Sound and Mobile Devices

Musical Background (formation, etc.)	None	Basic	Average	Upper	Expert
Previous Use of Mobile Phone	Never	Rare	Occasionnal	Often	Daily Basis
Previous Use of PDA (or similar)	Never	Rare	Occasionnal	Often	Daily Basis
Previous Use of Electronic Games	Never	Rare	Occasionnal	Often	Daily Basis
If it is the case, are there some of them using maps?	Baldur's Gate, ...				

Part 3 : Scenarios

Sound Pattern	Earcons	Sound Pattern	Auditory Icons
Scenario 1 (in sec)	10	Scenario 3a (in sec)	21
Scenario 2a (in sec)	9	Scenario 3b (in sec)	58
Scenario 2b (in sec)	11	Scenario 3c (in sec)	66

Part 4 : Interface Overall Evaluation

Lisibility	Difficult	Not so good	Average	Good	Excellent
Map Navigation	Difficult	Not so good	Average	Good	Excellent
Menus	Difficult	Not so good	Average	Good	Excellent

Part 5 : Sound Pattern Evaluation

Auditory Icons	Horrible	Not so good	Average	Good	Excellent
Earcons	Horrible	Not so good	Average	Good	Excellent
Sound Utility	Useless	Not so usefull	Neutral	Usefull	Essential

Part 6 : General Comments

- Il y a un problème de classification des objets.
- Les earcons sont peu agaçantes à la fin.
- Le son peut aider, mais moi ça m'est égal.

Part 1 : Personnal Information

Name	Christophe Grégoire
Age	23
Gender	Male
Title/Position	Student
Field of Activity/Study	Computer Science
E-mail Address	cgregoir@info.fundp.ac.be

Part 2 : Previous Experience in Sound and Mobile Devices

Musical Background (formation, etc.)	None	Basic	Average	Upper	Expert
Previous Use of Mobile Phone	Never	Rare	Occasionnal	Often	Daily Basis
Previous Use of PDA (or similar)	Never	Rare	Occasionnal	Often	Daily Basis
Previous Use of Electronic Games	Never	Rare	Occasionnal	Often	Daily Basis
If it is the case, are there some of them using maps?	/				

Part 3 : Scenarios

Sound Pattern	Auditory Icons	Sound Pattern	Earcons
Scenario 1 (in sec)	18	Scenario 3a (in sec)	29
Scenario 2a (in sec)	15	Scenario 3b (in sec)	16
Scenario 2b (in sec)	9	Scenario 3c (in sec)	15

Part 4 : Interface Overall Evaluation

Lisibility	Difficult	Not so good	Average	Good	Excellent
Map Navigation	Difficult	Not so good	Average	Good	Excellent
Menus	Difficult	Not so good	Average	Good	Excellent

Part 5 : Sound Pattern Evaluation

Auditory Icons	Horrible	Not so good	Average	Good	Excellent
Earcons	Horrible	Not so good	Average	Good	Excellent
Sound Utility	Useless	Not so usefull	Neutral	Usefull	Essential

Part 6 : General Comments

- Pour une meilleure lisibilité, il aurait été mieux d'utiliser des couleurs différentes pour distinguer les couloirs des pièces.
- Les portes sont bien représentées, mais il aurait été mieux de représenter les objets par autre chose que de simples carrés (avec des icones par exemple).
- Quitter une pièce est un peu difficile.
- Pour les menus, il y a un problème de classification.
- Les auditory icons ont une durée plus longue, le son est plus présent.
- Les earcons sont plus énervantes, mais néanmoins plus discrètes.
- L'usage du pattern sonore dépend de l'environnement. Probablement que dans un musée, les auditory icons seraient plus appropriées.

Part 1 : Personnal Information

Name	Samuel Pierard
Age	23
Gender	Male
Title/Position	Student, Employed
Field of Activity/Study	Secretariat
E-mail Address	/

Part 2 : Previous Experience in Sound and Mobile Devices

Musical Background (formation, etc.)	None	**Basic**	Average	Upper	Expert
Previous Use of Mobile Phone	Never	Rare	Occasionnal	Often	**Daily Basis**
Previous Use of PDA (or similar)	**Never**	Rare	Occasionnal	Often	Daily Basis
Previous Use of Electronic Games	Never	Rare	Occasionnal	**Often**	Daily Basis
If it is the case, are there some of them using maps?	Age of Empire, …				

Part 3 : Scenarios

Sound Pattern	Earcons	Sound Pattern	Auditory Icons
Scenario 1 (in sec)	47	Scenario 3a (in sec)	29
Scenario 2a (in sec)	14	Scenario 3b (in sec)	6
Scenario 2b (in sec)	34	Scenario 3c (in sec)	56

Part 4 : Interface Overall Evaluation

Lisibility	Difficult	Not so good	Average	**Good**	Excellent
Map Navigation	Difficult	Not so good	Average	Good	**Excellent**
Menus	Difficult	Not so good	**Average**	Good	Excellent

Part 5 : Sound Pattern Evaluation

Auditory Icons	Horrible	Not so good	Average	Good	**Excellent**
Earcons	Horrible	Not so good	**Average**	Good	Excellent
Sound Utility	Useless	Not so usefull	**Neutral**	Usefull	Essential

Part 6 : General Comments

- Pour la navigation, on sait déjà comment ça marche au bout de 10 secondes!
- Par contre, on se perd un peu dans les menus au début.
- Pour les auditory icons, le bruit pour changer de pièce est bien, on sait qu'on passe une porte. Quand au bruit des objets, on sait que l'on « tourne une page ».
- Les earcons sont ok, le son est bien.
- Le son dans le programme ne me dérange pas. J'aime bien, au début en tout cas !

Part 1 : Personnal Information

Name	Amandine Schuurman
Age	21
Gender	Female
Title/Position	Student
Field of Activity/Study	Computer Science
E-mail Address	aschuurm@info.fundp.ac.be

Part 2 : Previous Experience in Sound and Mobile Devices

Musical Background (formation, etc.)	None	Basic	Average	Upper	Expert
Previous Use of Mobile Phone	Never	Rare	Occasionnal	Often	Daily Basis
Previous Use of PDA (or similar)	Never	Rare	Occasionnal	Often	Daily Basis
Previous Use of Electronic Games	Never	Rare	Occasionnal	Often	Daily Basis
If it is the case, are there some of them using maps?	The Sims, ...				

Part 3 : Scenarios

Sound Pattern	Auditory Icons	Sound Pattern	Earcons
Scenario 1 (in sec)	8	Scenario 3a (in sec)	21
Scenario 2a (in sec)	6	Scenario 3b (in sec)	26
Scenario 2b (in sec)	9	Scenario 3c (in sec)	12

Part 4 : Interface Overall Evaluation

Lisibility	Difficult	Not so good	Average	Good	Excellent
Map Navigation	Difficult	Not so good	Average	Good	Excellent
Menus	Difficult	Not so good	Average	Good	Excellent

Part 5 : Sound Pattern Evaluation

Auditory Icons	Horrible	Not so good	Average	Good	Excellent
Earcons	Horrible	Not so good	Average	Good	Excellent
Sound Utility	Useless	Not so usefull	Neutral	Usefull	Essential

Part 6 : General Comments

Il faudrait mieux représenter les entrées, par des petites flèches par exemple.

- Quand on switche de pièce, un petit plan dans le coin de l'écran avec la pièce courante éclairée permettrait de mieux se repérer (notamment pour les longs couloirs tout droits).
- Pour les auditory icons :
  - le bruit pour zoomer est bien, mais celui pour dézoomer l'est moins. Le bruit des earcons pour revenir au plan d'ensemble est mieux !
  - le son des portes est super !
  - le bruit des objets ressemble au son d'un appareil photo, il est trop long.
- Les earcons sont bien, mais ca me tape sur le système ! Des sons plus courts auraient été mieux !
- Le son est utile au début pour se repérer, mais à la longue… C'est surtout utile pour les débutants.

Part 1 : Personnal Information

Name	David Leonet
Age	22
Gender	Male
Title/Position	Student
Field of Activity/Study	Mathematics
E-mail Address	/

Part 2 : Previous Experience in Sound and Mobile Devices

Musical Background (formation, etc.)	None	Basic	Average	Upper	Expert
Previous Use of Mobile Phone	Never	Rare	Occasionnal	Often	Daily Basis
Previous Use of PDA (or similar)	Never	Rare	Occasionnal	Often	Daily Basis
Previous Use of Electronic Games	Never	Rare	Occasionnal	Often	Daily Basis
If it is the case, are there some of them using maps?	Quake like, …				

Part 3 : Scenarios

Sound Pattern	Earcons	Sound Pattern	Auditory Icons
Scenario 1 (in sec)	15	Scenario 3a (in sec)	16
Scenario 2a (in sec)	8	Scenario 3b (in sec)	23
Scenario 2b (in sec)	14	Scenario 3c (in sec)	50

Part 4 : Interface Overall Evaluation

Lisibility	Difficult	Not so good	Average	Good	Excellent
Map Navigation	Difficult	Not so good	Average	Good	Excellent
Menus	Difficult	Not so good	Average	Good	Excellent

Part 5 : Sound Pattern Evaluation

Auditory Icons	Horrible	Not so good	Average	Good	Excellent
Earcons	Horrible	Not so good	Average	Good	Excellent
Sound Utility	Useless	Not so usefull	Neutral	Usefull	Essential

Part 6 : General Comments

- Les portes sont un peu petites sur le plan des pièces.
- Il y a un problème de classification.
- Le son permet de confirmer nos actions.
- Ce serait bien de rajouter plus d'objets, comme les plantes dans le bâtiment, etc. pour mieux se repérer.

Part 1 : Personnal Information

Name	Katia Demaseure
Age	24
Gender	Female
Title/Position	Assistant, Researcher
Field of Activity/Study	Mathematics
E-mail Address	kdem@math.fundp.ac.be

Part 2 : Previous Experience in Sound and Mobile Devices

Musical Background (formation, etc.)	None	Basic	Average	Upper	Expert
Previous Use of Mobile Phone	Never	Rare	Occasionnal	Often	Daily Basis
Previous Use of PDA (or similar)	Never	Rare	Occasionnal	Often	Daily Basis
Previous Use of Electronic Games	Never	Rare	Occasionnal	Often	Daily Basis
If it is the case, are there some of them using maps?	/				

Part 3 : Scenarios

Sound Pattern	Auditory Icons	Sound Pattern	Earcons
Scenario 1 (in sec)	16	Scenario 3a (in sec)	14
Scenario 2a (in sec)	14	Scenario 3b (in sec)	31
Scenario 2b (in sec)	10	Scenario 3c (in sec)	9

Part 4 : Interface Overall Evaluation

Lisibility	Difficult	Not so good	Average	Good	Excellent
Map Navigation	Difficult	Not so good	Average	Good	Excellent
Menus	Difficult	Not so good	Average	Good	Excellent

Part 5 : Sound Pattern Evaluation

Auditory Icons	Horrible	Not so good	Average	Good	Excellent
Earcons	Horrible	Not so good	Average	Good	Excellent
Sound Utility	Useless	Not so usefull	Neutral	Usefull	Essential

Part 6 : General Comments

- Il y a un problème de classification.
- Ce serait bien de garder en mémoire les pièces que l'on a déjà vues.
- Ce serait bien de mettre les points cardinaux pour mieux se repérer.

Part 1 : Personnal Information

Name	Philippe Keutgen
Age	19
Gender	Male
Title/Position	Student
Field of Activity/Study	Geography
E-mail Address	philippe_keutgen@hotmail.com

Part 2 : Previous Experience in Sound and Mobile Devices

Musical Background (formation, etc.)	None	Basic	Average	**Upper**	Expert
Previous Use of Mobile Phone	Never	Rare	Occasionnal	Often	**Daily Basis**
Previous Use of PDA (or similar)	**Never**	Rare	Occasionnal	Often	Daily Basis
Previous Use of Electronic Games	Never	Rare	Occasionnal	**Often**	Daily Basis
If it is the case, are there some of them using maps?	/				

Part 3 : Scenarios

Sound Pattern	Earcons	Sound Pattern	Auditory Icons
Scenario 1 (in sec)	17	Scenario 3a (in sec)	32
Scenario 2a (in sec)	12	Scenario 3b (in sec)	13
Scenario 2b (in sec)	6	Scenario 3c (in sec)	7

Part 4 : Interface Overall Evaluation

Lisibility	Difficult	Not so good	Average	Good	**Excellent**
Map Navigation	Difficult	Not so good	Average	**Good**	Excellent
Menus	Difficult	Not so good	Average	Good	**Excellent**

Part 5 : Sound Pattern Evaluation

Auditory Icons	Horrible	Not so good	Average	**Good**	Excellent
Earcons	Horrible	Not so good	**Average**	Good	Excellent
Sound Utility	Useless	Not so usefull	Neutral	Usefull	**Essential**

Part 6 : General Comments

- Le plan est un peu petit.
- Il faut faire attention à la classification.
- Le son est utile pour confirmer ses actions.

Part 1 : Personnal Information

Name	Sylvain Antoine
Age	21
Gender	Male
Title/Position	Student
Field of Activity/Study	Geography
E-mail Address	maoboy3@hotmail.com

Part 2 : Previous Experience in Sound and Mobile Devices

Musical Background (formation, etc.)	**None**	Basic	Average	Upper	Expert
Previous Use of Mobile Phone	Never	Rare	Occasionnal	**Often**	Daily Basis
Previous Use of PDA (or similar)	Never	Rare	**Occasionnal**	Often	Daily Basis
Previous Use of Electronic Games	Never	Rare	**Occasionnal**	Often	Daily Basis
If it is the case, are there some of them using maps?	Quake like, …				

Part 3 : Scenarios

Sound Pattern	Auditory Icons	Sound Pattern	Earcons
Scenario 1 (in sec)	5	Scenario 3a (in sec)	28
Scenario 2a (in sec)	7	Scenario 3b (in sec)	11
Scenario 2b (in sec)	5	Scenario 3c (in sec)	13

Part 4 : Interface Overall Evaluation

Lisibility	Difficult	Not so good	Average	**Good**	Excellent
Map Navigation	Difficult	Not so good	Average	**Good**	Excellent
Menus	Difficult	Not so good	Average	**Good**	Excellent

Part 5 : Sound Pattern Evaluation

Auditory Icons	Horrible	Not so good	Average	**Good**	Excellent
Earcons	Horrible	Not so good	**Average**	Good	Excellent
Sound Utility	Useless	Not so usefull	Neutral	**Usefull**	Essential

Part 6 : General Comments

- Dans les menus de recherche (select subcategory detail), il serait plus intuitif de ne pas avoir de sélection par défaut.
- A la fin les earcons deviennent un peu irritantes.

Part 1 : Personnal Information

Name	Barbara Van Eetvelde
Age	24
Gender	Female
Title/Position	Assistant, Researcher
Field of Activity/Study	Geography
E-mail Address	barbara.denil@fundp.ac.be

Part 2 : Previous Experience in Sound and Mobile Devices

Musical Background (formation, etc.)	None	Basic	Average	Upper	Expert
Previous Use of Mobile Phone	Never	Rare	Occasionnal	Often	Daily Basis
Previous Use of PDA (or similar)	Never	Rare	Occasionnal	Often	Daily Basis
Previous Use of Electronic Games	Never	Rare	Occasionnal	Often	Daily Basis
If it is the case, are there some of them using maps?	/				

Part 3 : Scenarios

Sound Pattern	Earcons	Sound Pattern	Auditory Icons
Scenario 1 (in sec)	10	Scenario 3a (in sec)	16
Scenario 2a (in sec)	13	Scenario 3b (in sec)	22
Scenario 2b (in sec)	11	Scenario 3c (in sec)	8

Part 4 : Interface Overall Evaluation

Lisibility	Difficult	Not so good	Average	Good	Excellent
Map Navigation	Difficult	Not so good	Average	Good	Excellent
Menus	Difficult	Not so good	Average	Good	Excellent

Part 5 : Sound Pattern Evaluation

Auditory Icons	Horrible	Not so good	Average	Good	Excellent
Earcons	Horrible	Not so good	Average	Good	Excellent
Sound Utility	Useless	Not so usefull	Neutral	Usefull	Essential

Part 6 : General Comments

- Il faudrait que tous les objets soient désélectionnés par défaut dans la recherche (au niveau du Select subcategory detail).
- Les auditory icons ne sont pas top. On reconnaît quand même la page qui se tourne pour les objets.
- Il faut aussi penser au problème de langue : les menus sont en anglais, alors que les objets sont en français !

Part 1 : Personnal Information

Name	Octave Detaille
Age	20
Gender	Male
Title/Position	Student
Field of Activity/Study	Mathematics, Computer Science orientation
E-mail Address	hsq125@hotmail.com

Part 2 : Previous Experience in Sound and Mobile Devices

Musical Background (formation, etc.)	None	Basic	Average	Upper	Expert
Previous Use of Mobile Phone	Never	Rare	Occasionnal	Often	Daily Basis
Previous Use of PDA (or similar)	Never	Rare	Occasionnal	Often	Daily Basis
Previous Use of Electronic Games	Never	Rare	Occasionnal	Often	Daily Basis
If it is the case, are there some of them using maps?	Quake like, …				

Part 3 : Scenarios

Sound Pattern	Auditory Icons	Sound Pattern	Earcons
Scenario 1 (in sec)	11	Scenario 3a (in sec)	21
Scenario 2a (in sec)	10	Scenario 3b (in sec)	34
Scenario 2b (in sec)	12	Scenario 3c (in sec)	15

Part 4 : Interface Overall Evaluation

Lisibility	Difficult	Not so good	Average	Good	Excellent
Map Navigation	Difficult	Not so good	Average	Good	Excellent
Menus	Difficult	Not so good	Average	Good	Excellent

Part 5 : Sound Pattern Evaluation

Auditory Icons	Horrible	Not so good	Average	Good	Excellent
Earcons	Horrible	Not so good	Average	Good	Excellent
Sound Utility	Useless	Not so usefull	Neutral	Usefull	Essential

Part 6 : General Comments

- L'interface est simple et facile à utiliser.
- Les portes sont difficiles à attraper.
- Le niveau de détail est suffisant pour un plan.
- Ce serait mieux de remettre les pièces dans leur contexte.
- Il y a un problème de classification : il aurait fallu plus de critères de recherche dès le départ.
- Pour les auditory icons, le son des portes est ok, et le son pour les objets est intuitif (bruit d'une feuille de papier pour la documentation).
- Pour les earcons, il faut le temps de se faire aux sons.

Part 1 : Personnal Information

Name	David Voogd
Age	23
Gender	Male
Title/Position	Student
Field of Activity/Study	Computer Science
E-mail Address	dvoogd@info.fundp.ac.be

Part 2 : Previous Experience in Sound and Mobile Devices

Musical Background (formation, etc.)	None	Basic	Average	Upper	Expert
Previous Use of Mobile Phone	Never	Rare	Occasionnal	Often	Daily Basis
Previous Use of PDA (or similar)	Never	Rare	Occasionnal	Often	Daily Basis
Previous Use of Electronic Games	Never	Rare	Occasionnal	Often	Daily Basis
If it is the case, are there some of them using maps?	Counter Strike, Mafia, …				

Part 3 : Scenarios

Sound Pattern	Earcons	Sound Pattern	Auditory Icons
Scenario 1 (in sec)	19	Scenario 3a (in sec)	49
Scenario 2a (in sec)	9	Scenario 3b (in sec)	32
Scenario 2b (in sec)	12	Scenario 3c (in sec)	5

Part 4 : Interface Overall Evaluation

Lisibility	Difficult	Not so good	Average	Good	Excellent
Map Navigation	Difficult	Not so good	Average	Good	Excellent
Menus	Difficult	Not so good	Average	Good	Excellent

Part 5 : Sound Pattern Evaluation

Auditory Icons	Horrible	Not so good	Average	Good	Excellent
Earcons	Horrible	Not so good	Average	Good	Excellent
Sound Utility	Useless	Not so usefull	Neutral	Usefull	Essential

Part 6 : General Comments

- Il y a un problème de classification.
- Pour les auditory icons, les portes j'aime bien, la page c'est bien. Les zooms, oui mais…
- Les earcons, ça c'est chouette.

Part 1 : Personnal Information

Name	Laura O'Connor
Age	21
Gender	Female
Title/Position	Student
Field of Activity/Study	Philosophy
E-mail Address	laura_o_connor@hotmail.com

Part 2 : Previous Experience in Sound and Mobile Devices

Musical Background (formation, etc.)	None	Basic	Average	Upper	Expert
Previous Use of Mobile Phone	Never	Rare	Occasionnal	Often	Daily Basis
Previous Use of PDA (or similar)	Never	Rare	Occasionnal	Often	Daily Basis
Previous Use of Electronic Games	Never	Rare	Occasionnal	Often	Daily Basis
If it is the case, are there some of them using maps?	/				

Part 3 : Scenarios

Sound Pattern	Auditory Icons	Sound Pattern	Earcons
Scenario 1 (in sec)	18	Scenario 3a (in sec)	59
Scenario 2a (in sec)	12	Scenario 3b (in sec)	24
Scenario 2b (in sec)	22	Scenario 3c (in sec)	7

Part 4 : Interface Overall Evaluation

Lisibility	Difficult	Not so good	Average	Good	Excellent
Map Navigation	Difficult	Not so good	Average	Good	Excellent
Menus	Difficult	Not so good	Average	Good	Excellent

Part 5 : Sound Pattern Evaluation

Auditory Icons	Horrible	Not so good	Average	Good	Excellent
Earcons	Horrible	Not so good	Average	Good	Excellent
Sound Utility	Useless	Not so usefull	Neutral	Usefull	Essential

Part 6 : General Comments

- Le programme est facile à utiliser, mais quelques explications sont quand même nécessaires.
- Le bouton begin à la fin de la phase de customization manque de clarté.
- Le son m'énerve ! Les earcons sont pires que les auditory icons.

Part 1 : Personnal Information

Name	Nicolas Riquet
Age	20
Gender	Male
Title/Position	Student
Field of Activity/Study	Computer Science
E-mail Address	nriquet@info.fundp.ac.be

Part 2 : Previous Experience in Sound and Mobile Devices

Musical Background (formation, etc.)	None	Basic	Average	Upper	Expert
Previous Use of Mobile Phone	Never	Rare	Occasionnal	Often	Daily Basis
Previous Use of PDA (or similar)	Never	Rare	Occasionnal	Often	Daily Basis
Previous Use of Electronic Games	Never	Rare	Occasionnal	Often	Daily Basis
If it is the case, are there some of them using maps?	Quake like, …				

Part 3 : Scenarios

Sound Pattern	Earcons	Sound Pattern	Auditory Icons
Scenario 1 (in sec)	22	Scenario 3a (in sec)	10
Scenario 2a (in sec)	10	Scenario 3b (in sec)	75
Scenario 2b (in sec)	10	Scenario 3c (in sec)	20

Part 4 : Interface Overall Evaluation

Lisibility	Difficult	Not so good	Average	Good	Excellent
Map Navigation	Difficult	Not so good	Average	Good	Excellent
Menus	Difficult	Not so good	Average	Good	Excellent

Part 5 : Sound Pattern Evaluation

Auditory Icons	Horrible	Not so good	Average	Good	Excellent
Earcons	Horrible	Not so good	Average	Good	Excellent
Sound Utility	Useless	Not so usefull	Neutral	Usefull	Essential

Part 6 : General Comments

- La lisibilité et la navigation sont asseez intuitive.
- Il y a un problème de classificatino, il aurait fallu gérer plus de catégories (exemple : matériel pour les projecteurs, ou avoir un champ objet). Les équipes et les personnes sont ok, mais les objets posent problèmes.
- Les earcons sont plus originaux et mettent « plus de bonne humeur ».
- Les auditory icons ne sont pas mal non plus, mais elles sont plus sérieuses et plus classiques.
- Un menu Recherche aurait été mieux que le menu Restart Tour !
- Il y a un problème de langues !

Part 1 : Personnal Information

Name	Benoit Vanderose
Age	20
Gender	Male
Title/Position	Student
Field of Activity/Study	Computer Science
E-mail Address	bvandero@info.fundp.ac.be

Part 2 : Previous Experience in Sound and Mobile Devices

Musical Background (formation, etc.)	None	Basic	Average	Upper	Expert
Previous Use of Mobile Phone	Never	Rare	Occasionnal	Often	Daily Basis
Previous Use of PDA (or similar)	Never	Rare	Occasionnal	Often	Daily Basis
Previous Use of Electronic Games	Never	Rare	Occasionnal	Often	Daily Basis
If it is the case, are there some of them using maps?	Fallout, Warcraft, …				

Part 3 : Scenarios

Sound Pattern	Auditory Icons	Sound Pattern	Earcons
Scenario 1 (in sec)	15	Scenario 3a (in sec)	16
Scenario 2a (in sec)	10	Scenario 3b (in sec)	115
Scenario 2b (in sec)	12	Scenario 3c (in sec)	17

Part 4 : Interface Overall Evaluation

Lisibility	Difficult	Not so good	Average	Good	Excellent
Map Navigation	Difficult	Not so good	Average	Good	Excellent
Menus	Difficult	Not so good	Average	Good	Excellent

Part 5 : Sound Pattern Evaluation

Auditory Icons	Horrible	Not so good	Average	Good	Excellent
Earcons	Horrible	Not so good	Average	Good	Excellent
Sound Utility	Useless	Not so usefull	Neutral	Usefull	Essential

Part 6 : General Comments

- La lisibilité est assez claire, et la navigation assez naturelle.
- Pour les menus de recherche, à part la classification des objets utilitaires, ça va. Le select all par défaut pose problème dans la partie Select subcategory detail (ce serait plus naturel de n'avoir rien de sélectionné par défaut dans ce dernier écran). L'existence du bouton Back est une bonne idée, c'est pratique.
- Les earcons sont bien dans leur genre, mais je préfère les sons naturels.
- A la fin, on pourrait se lasser du son.
- Néanmoins, le son est utile sur un PDA, car on sait quand on clique, on est certain d'avoir cliqué. C'est surtout pratique pour la validation.

Part 1 : Personnal Information

Name	Etienne Reygaerdts
Age	23
Gender	Male
Title/Position	Student
Field of Activity/Study	Computer Science
E-mail Address	ereygaer@info.fundp.ac.be

Part 2 : Previous Experience in Sound and Mobile Devices

Musical Background (formation, etc.)	None	Basic	Average	Upper	Expert
Previous Use of Mobile Phone	Never	Rare	Occasionnal	Often	Daily Basis
Previous Use of PDA (or similar)	Never	Rare	Occasionnal	Often	Daily Basis
Previous Use of Electronic Games	Never	Rare	Occasionnal	Often	Daily Basis
If it is the case, are there some of them using maps?	Counter Strike, Day of Defeat, Heroes of Might & Magic, …				

Part 3 : Scenarios

Sound Pattern	Earcons	Sound Pattern	Auditory Icons
Scenario 1 (in sec)	11	Scenario 3a (in sec)	14
Scenario 2a (in sec)	11	Scenario 3b (in sec)	32
Scenario 2b (in sec)	11	Scenario 3c (in sec)	6

Part 4 : Interface Overall Evaluation

Lisibility	Difficult	Not so good	Average	Good	Excellent
Map Navigation	Difficult	Not so good	Average	Good	Excellent
Menus	Difficult	Not so good	Average	Good	Excellent

Part 5 : Sound Pattern Evaluation

Auditory Icons	Horrible	Not so good	Average	Good	Excellent
Earcons	Horrible	Not so good	Average	Good	Excellent
Sound Utility	Useless	Not so usefull	Neutral	Usefull	Essential

Part 6 : General Comments

- Du point de vue de la lisibilité, l'affichage est un peu petit. Les connexions sont un peu petites.
- Heureusement que les pièces ont été découpées. Il ne faut pas trop descendre dans la découpe.
- Il aurait été plus judicieux de remettre chaque pièce dans son contexte, en particulier pour les couloirs.
- Quand on débouche dans un couloir, on ne sait pas d'où l'on vient et ce que l'on a déjà visité (sauf avec la simulation infrarouge).
- Le Begin du Select subcategory detail n'est pas approprié.
- Il y a un problème de classification.
- Il aurait fallu utiliser un menu Search plutôt que le Restart Tour.
- Pour les auditory icons, la porte c'est bien, le bruit des objets est un peu long et le zoom n'est pas top.
- Les earcons ne sont pas assez représentatifs, et à la longue ça m'énerve. Ils sont un peu longs.

Part 1 : Personnal Information

Name	Louis Zuyderhoff
Age	22
Gender	Male
Title/Position	Student
Field of Activity/Study	Computer Science
E-mail Address	lzuiderh@info.fundp.ac.be

Part 2 : Previous Experience in Sound and Mobile Devices

Musical Background (formation, etc.)	**None**	Basic	Average	Upper	Expert
Previous Use of Mobile Phone	Never	Rare	Occasionnal	Often	**Daily Basis**
Previous Use of PDA (or similar)	**Never**	Rare	Occasionnal	Often	Daily Basis
Previous Use of Electronic Games	Never	Rare	Occasionnal	**Often**	Daily Basis
If it is the case, are there some of them using maps?	Counter Strike, Baldur's Gate, Diablo, …				

Part 3 : Scenarios

Sound Pattern	Auditory Icons	Sound Pattern	Earcons
Scenario 1 (in sec)	23	Scenario 3a (in sec)	23
Scenario 2a (in sec)	15	Scenario 3b (in sec)	23
Scenario 2b (in sec)	18	Scenario 3c (in sec)	20

Part 4 : Interface Overall Evaluation

Lisibility	Difficult	Not so good	Average	**Good**	Excellent
Map Navigation	Difficult	Not so good	Average	**Good**	Excellent
Menus	Difficult	Not so good	**Average**	Good	Excellent

Part 5 : Sound Pattern Evaluation

Auditory Icons	Horrible	Not so good	Average	**Good**	Excellent
Earcons	Horrible	Not so good	**Average**	Good	Excellent
Sound Utility	Useless	**Not so usefull**	Neutral	Usefull	Essential

Part 6 : General Comments

- La navigation est facile.
- Pour les recherches, le critère "fonction" aurait été plus judicieux que le critère "titre". Il aurait fallu avoir un critère "objet" !
- Pour l'écran Select subcategory detail, ce serait mieux de n'avoir qu'un objet sélectionné par défaut.
- Pour un musée, ce serait mieux d'avoir des icones différentes pour chaque type d'objet. Pour faire la différence entre des statues et des tableaux par exemple.
- Les auditory icons me font un peu penser aux vieux jeux vidéo.
- Le son, c'est du gadget, ça m'énerve, mais l'aspect ludique est important pour un musée.

Part 1 : Personnal Information

Name	Stéphane Sandron
Age	20
Gender	Male
Title/Position	Student
Field of Activity/Study	Computer Science
E-mail Address	ssandron@info.fundp.ac.be

Part 2 : Previous Experience in Sound and Mobile Devices

Musical Background (formation, etc.)	None	Basic	Average	Upper	Expert
Previous Use of Mobile Phone	Never	Rare	Occasionnal	Often	Daily Basis
Previous Use of PDA (or similar)	Never	Rare	Occasionnal	Often	Daily Basis
Previous Use of Electronic Games	Never	Rare	Occasionnal	Often	Daily Basis
If it is the case, are there some of them using maps?	Baldur's Gate, Counter Strike, …				

Part 3 : Scenarios

Sound Pattern	Earcons	Sound Pattern	Auditory Icons
Scenario 1 (in sec)	9	Scenario 3a (in sec)	33
Scenario 2a (in sec)	6	Scenario 3b (in sec)	7
Scenario 2b (in sec)	7	Scenario 3c (in sec)	6

Part 4 : Interface Overall Evaluation

Lisibility	Difficult	Not so good	Average	Good	Excellent
Map Navigation	Difficult	Not so good	Average	Good	Excellent
Menus	Difficult	Not so good	Average	Good	Excellent

Part 5 : Sound Pattern Evaluation

Auditory Icons	Horrible	Not so good	Average	Good	Excellent
Earcons	Horrible	Not so good	Average	Good	Excellent
Sound Utility	Useless	Not so usefull	Neutral	Usefull	Essential

Part 6 : General Comments

- Ce serait bien d'avoir les pièces autour d'un couloir au niveau du Room Map.
- Ce serait mieux d'avoir un bouton Search.
- Le bruit des objets, ça peut encore aller.

Part 1 : Personnal Information

Name	Cédric Jadoul
Age	20
Gender	Male
Title/Position	Student
Field of Activity/Study	Computer Science
E-mail Address	cjadoul@info.fundp.ac.be

Part 2 : Previous Experience in Sound and Mobile Devices

Musical Background (formation, etc.)	None	Basic	Average	Upper	**Expert**
Previous Use of Mobile Phone	Never	Rare	Occasionnal	Often	**Daily Basis**
Previous Use of PDA (or similar)	Never	Rare	**Occasionnal**	Often	Daily Basis
Previous Use of Electronic Games	Never	**Rare**	Occasionnal	Often	Daily Basis
If it is the case, are there some of them using maps?	Sim City, …				

Part 3 : Scenarios

Sound Pattern	Auditory Icons	Sound Pattern	Earcons
Scenario 1 (in sec)	7	Scenario 3a (in sec)	19
Scenario 2a (in sec)	10	Scenario 3b (in sec)	8
Scenario 2b (in sec)	6	Scenario 3c (in sec)	11

Part 4 : Interface Overall Evaluation

Lisibility	Difficult	Not so good	Average	Good	**Excellent**
Map Navigation	Difficult	Not so good	Average	Good	**Excellent**
Menus	Difficult	Not so good	Average	**Good**	Excellent

Part 5 : Sound Pattern Evaluation

Auditory Icons	Horrible	Not so good	Average	Good	**Excellent**
Earcons	Horrible	Not so good	**Average**	Good	Excellent
Sound Utility	Useless	Not so usefull	Neutral	**Usefull**	Essential

Part 6 : General Comments

- Pour le plan des couloirs, ce serait mieux d'avoir plus d'indications sur les pièces environnantes.
- Il y a le problème du Search. Les titres et les objets sont mélangés : il aura mieux valu définit un critère objet et un critère personne.
- Pour les auditory icons, on reconnaît bien le bruit de porte et le bruit de fichier qu'on ouvre.
- Les earcons ne sont pas mal, mais les auditory icons sont mieux !
- Le son permet de confirmer les clics. J'aime moins sans le son.

Part 1 : Personnal Information

Name	Stéphane Génicot
Age	20
Gender	Male
Title/Position	Student
Field of Activity/Study	Computer Science
E-mail Address	sgenicot@info.fundp.ac.be

Part 2 : Previous Experience in Sound and Mobile Devices

Musical Background (formation, etc.)	None	Basic	Average	Upper	Expert
Previous Use of Mobile Phone	Never	Rare	Occasionnal	Often	Daily Basis
Previous Use of PDA (or similar)	Never	Rare	Occasionnal	Often	Daily Basis
Previous Use of Electronic Games	Never	Rare	Occasionnal	Often	Daily Basis
If it is the case, are there some of them using maps?	Hitman 2, GTA 3, ...				

Part 3 : Scenarios

Sound Pattern	Earcons	Sound Pattern	Auditory Icons
Scenario 1 (in sec)	5	Scenario 3a (in sec)	24
Scenario 2a (in sec)	6	Scenario 3b (in sec)	8
Scenario 2b (in sec)	9	Scenario 3c (in sec)	13

Part 4 : Interface Overall Evaluation

Lisibility	Difficult	Not so good	Average	Good	Excellent
Map Navigation	Difficult	Not so good	Average	Good	Excellent
Menus	Difficult	Not so good	Average	Good	Excellent

Part 5 : Sound Pattern Evaluation

Auditory Icons	Horrible	Not so good	Average	Good	Excellent
Earcons	Horrible	Not so good	Average	Good	Excellent
Sound Utility	Useless	Not so usefull	Neutral	Usefull	Essential

Part 6 : General Comments

- Pour la recherche, il y a toujours moyen de s'y retrouver à partir d'une caractéristique.
- On manque d'information pour switcher de level (comment sortir d'une pièce, etc.).
- Les auditory icons se ressemblent plus. Le bruit de la porte, ça va.
- Les earcons sont plus clairs et plus agréables.
- Le son est utile, car avec les PDA, on n'est jamais sûr de cliquer au bon endroit. Avec les sons, nos actions sont confirmées.

Appendix D : Questionnaire Analysis

The following pages present the questionnaire analysis generated by S-Plus. First, we analyze the influence of the background on the apprehension of the application (that is, the time to achieve the several scenarios), then the influence of the background on the appreciation of the sound patterns, and finally the relation between auditory icons and earcons. The method used is the discrimination tree method of S-Plus, which corresponds to a clustering method.

For this purpose, we remind you that we transformed some of our qualitative answers into ordinal values, given the qualitative answers had an appreciation gradation :

Musical Background (formation, etc.)	None	Basic	Average	Upper	Expert
Previous Use of Mobile Phone	Never	Rare	Occasionnal	Often	Daily Basis
Previous Use of PDA (or similar)	Never	Rare	Occasionnal	Often	Daily Basis
Previous Use of Electronic Games	Never	Rare	Occasionnal	Often	Daily Basis
Auditory Icons	Horrible	Not so good	Average	Good	Excellent
Earcons	Horrible	Not so good	Average	Good	Excellent
Sound Utility	Useless	Not so usefull	Neutral	Usefull	Essential
Associated ordinal value	**1**	**2**	**3**	**4**	**5**

We used S-Plus' discrimination trees with the following parameters :

- Mincut = 5 (minimum of observation before splitting)
- Minsize = 10 (minimum size of a node)
- Mindev = 0,010 (minimum deviance)

Thes values seemed reasonnable, given the small population we analyzed.
Note that S-Plus required choosing Mincut >= Minsize/2.

D.1 Critical Values Table For the t Distribution of Student

	$\alpha = 0.50$	$\alpha = 0.20$	$\alpha = 0.10$	$\alpha = 0.050$	$\alpha = 0.02$	$\alpha = 0.01$	$\alpha = 0.002$	$\alpha = 0.001$
ddl	$t_{0.25}$	$t_{0.10}$	$t_{0.05}$	$t_{0.025}$	$t_{0.01}$	$t_{0.005}$	$t_{0.001}$	$t_{0.0005}$
1	1.000	3.078	6.314	12.71	31.82	63.66	318.3	636.6
2	0.816	1.886	2.920	4.303	6.695	9.925	22.33	31.60
3	0.765	1.638	2.353	3.182	4.541	5.841	10.21	12.92
4	0.741	1.533	2.132	2.776	3.747	4.604	7.173	8.610
5	0.727	1.476	2.015	2.571	3.365	4.032	5.893	6.869
6	0.718	1.440	1.943	2.447	3.143	3.707	5.208	5.959
7	0.711	1.415	1.895	2.365	2.998	3.499	4.785	5.408
8	0.706	1.397	1.860	2.306	2.896	3.355	4.501	5.041
9	0.703	1.383	1.833	2.262	2.821	3.250	4.297	4.781
10	0.700	1.372	1.812	2.228	2.764	3.169	4.144	4.587
11	0.697	1.363	1.796	2.201	2.718	3.106	4.025	4.437
12	0.695	1.356	1.782	2.179	2.681	3.055	3.930	4.318
13	0.694	1.350	1.771	2.160	2.650	3.012	3.852	4.221
14	0.692	1.345	1.761	2.145	2.624	2.977	3.787	4.140
15	0.691	1.341	1.753	2.131	2.602	2.947	3.733	4.073
16	0.690	1.337	1.746	2.120	2.583	2.921	3.686	4.015
17	0.689	1.333	1.740	2.110	2.567	2.898	3.646	3.965
18	0.688	1.330	1.734	2.101	2.552	2.878	3.610	3.922
19	0.688	1.328	1.729	2.093	2.539	2.861	3.579	3.883
20	0.687	1.325	1.725	2.086	2.528	2.845	3.552	3.850
21	0.686	1.323	1.721	2.080	2.518	2.831	3.527	3.819
22	0.686	1.321	1.717	2.074	2.508	2.819	3.505	3.792
23	0.685	1.319	1.714	2.069	2.500	2.807	3.485	3.767
24	0.685	1.318	1.711	2.064	2.492	2.797	3.467	3.745
25	0.684	1.316	1.708	2.060	2.405	2.787	3.450	3.725
26	0.684	1.315	1.706	2.056	2.479	2.779	3.435	3.707
27	0.684	1.314	1.703	2.052	2.473	2.771	3.057	3.421
28	0.683	1.313	1.701	2.048	2.467	2.763	3.408	3.674
29	0.683	1.311	1.699	2.045	2.462	2.756	3.396	3.659
30	0.683	1.310	1.697	2.042	2.457	2.750	3.385	3.646
40	0.681	1.303	1.684	2.021	2.423	2.704	3.307	3.551
60	0.679	1.296	1.671	2.000	2.390	2.660	3.232	3.460
120	0.677	1.289	1.658	1.980	2.358	2.617	3.160	3.373
∞	0.674	1.282	1.645	1.960	2.326	2.576	3.090	3.291
	z.25	z.10	z.05	z.025	z.010	z.005	z.0010	z.0005

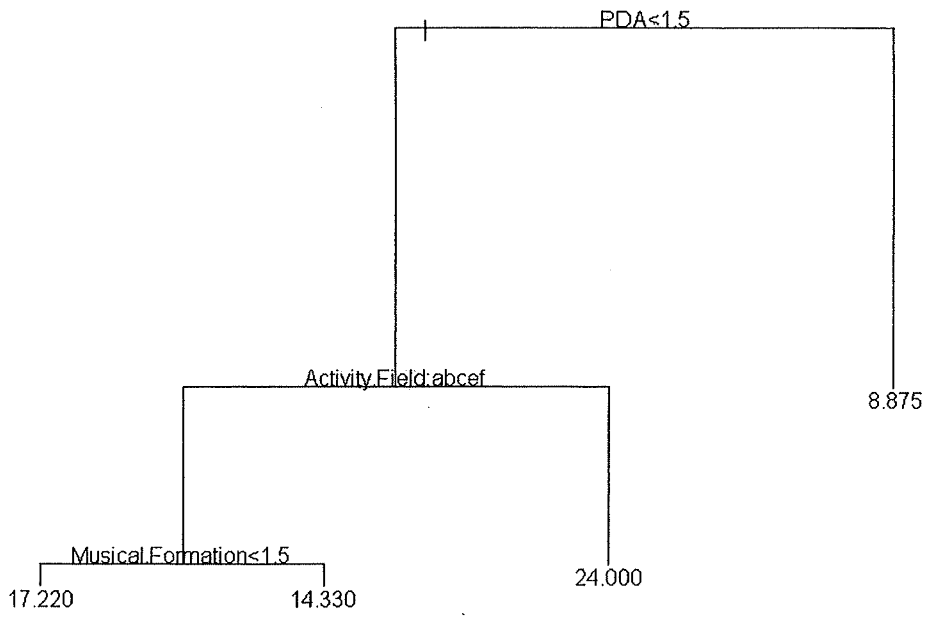
D.2 Analysis of the Apprehension of the Prototype

D.2.1 Scenario 1

Explanation of the time necessary to accomplish the Scenario1, based on the use of Cell Phone, PDA and Video Games, as well as the Musical Formation, the Sex and the Field of activity :

```
Regression tree:
tree(formula = Scenario1.time.in.sec ~ Sex + Activity.Field +
      Musical.Formation + GSM + PDA + Video.Games, data =
      resultats.questionnaire.ravi...all.quantitative, na.action =
      na.exclude, mincut = 5, minsize = 10, mindev = 0.01)
Variables actually used in tree construction:
[1] "PDA"              "Activity.Field"    "Musical.Formation"
Number of terminal nodes:  4
Residual mean deviance:  47.66 = 1144 / 24
Distribution of residuals:
     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
 -12.2200  -3.3330   0.7222   0.0000   2.1250  23.0000
node), split, n, deviance, yval
      * denotes terminal node

1) root 28 1891.00 15.430
  2) PDA<1.5 20 1379.00 18.050
    4)        Activity.Field:biology,computer-science,geography,maths/computer-
science,philosophy 15  400.90 16.070
      8) Musical.Formation<1.5 9  327.60 17.220 *
      9) Musical.Formation>1.5 6   43.33 14.330 *
    5) Activity.Field:maths,secretariat 5  742.00 24.000 *
  3) PDA>1.5 8   30.88  8.875 *
```

The derived regression tree:



Critical Probability Analysis :

X1 ≡ persons who never used a PDA before
X2 ≡ persons who already used a PDA at least once before

X1	(X1-Mean(X1))	$(X1\text{-}Mean(X1))^2$	X2	(X2-Mean(X2))	$(X2\text{-}Mean(X2))^2$
18	-0,05	0,0025	10	1,125	1,265625
17	-1,05	1,1025	8	-0,875	0,765625
14	-4,05	16,4025	11	2,125	4,515625
15	-3,05	9,3025	9	0,125	0,015625
18	-0,05	0,0025	10	1,125	1,265625
19	0,95	0,9025	5	-3,875	15,015625
11	-7,05	49,7025	7	-1,875	3,515625
25	6,95	48,3025	11	2,125	4,515625
15	-3,05	9,3025			
23	4,95	24,5025			
10	-8,05	64,8025			
22	3,95	15,6025			
5	-13,05	170,3025			
26	7,95	63,2025			
16	-2,05	4,2025			
16	-2,05	4,2025			
15	-3,05	9,3025			
11	-7,05	49,7025			
18	-0,05	0,0025			
47	28,95	838,1025			

n1 = 20
n2 = 8
Mean (X1) = 18,05
Mean (X1) = 8,875

Calculation of sp

$\Sigma\ (X1\text{-}Mean(X1))^2 =$ 1378,95
$\Sigma\ (X2\text{-}Mean(X2))^2 =$ 30,875
ddl = ((n1-1)+(n2-1)) = 26

$(\Sigma\ (X1\text{-}Mean(X1))^2 + \Sigma\ (X2\text{-}Mean(X2))^2) =$ 1409,825
$sp^2 = (\Sigma\ (X1\text{-}Mean(X1))^2 + \Sigma\ (X1\text{-}Mean(X1))^2 / ddl =$ 54,22403846

Calculation of the Interval of Confidence

Mean(X1) - Mean(X2) = 9,175
t.025 (for current ddl) = 2,06
sp = 7,363697336
(1/n1) = 0,05
(1/n2) = 0,125
(1/n1)+(1/n2) = 0,175
SquareRoot((1/n1)+(1/n2)) = 0,418330013
sp*SquareRoot((1/n1)+(1/n2)) = 3,080455604
t.025*sp*SquareRoot((1/n1)+(1/n2)) = 6,345738545

Hence ($\mu 1$-$\mu 2$) is between 2,829261455 and 15,52073855.

Calculation of the Critical Probability

t = (Mean(X1)-Mean(X2)) / (sp*SquareRoot((1/n1)+(1/n2))) = 2,97845552

For the current ddl, the Critical Values Table For the t Distribution of Student gives us :
$t > t.005$, which means $pc < .005$

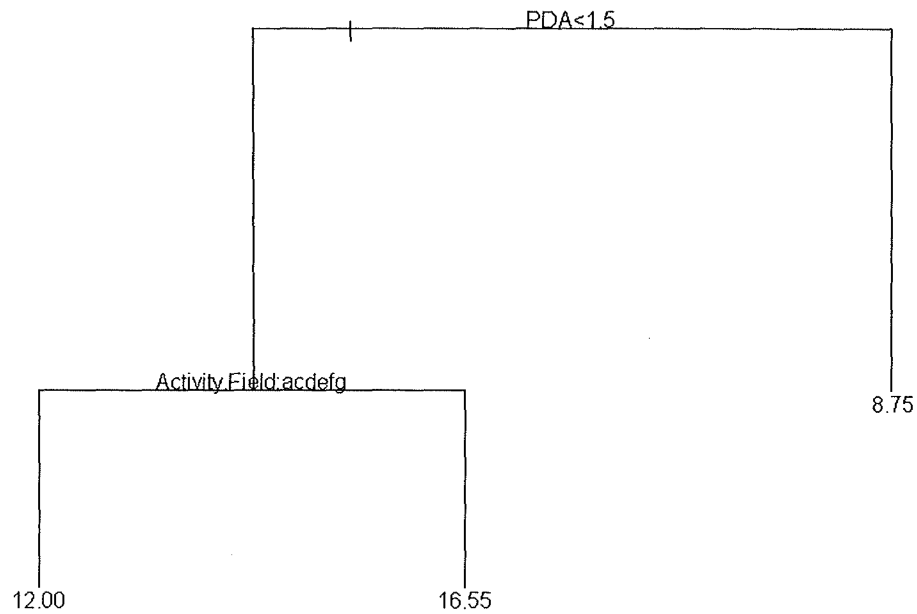
D.2.2 Scenario 2

Scenario 2a

Explanation of the time necessary to accomplish the Scenario2a based on the use of Cell Phone, PDA and Video Games, as well as the Musical Formation, the Sex and the Field of activity :

```
Regression tree:
tree(formula = Scenario2a.time.in.sec ~ Sex + Activity.Field +
      Musical.Formation + GSM + PDA + Video.Games, data =
      resultats.questionnaire.ravi...all.quantitative, na.action =
      na.exclude, mincut = 5, minsize = 10, mindev = 0.01)
Variables actually used in tree construction:
[1] "PDA"           "Activity.Field"
Number of terminal nodes:  3
Residual mean deviance:  95.93 = 2398 / 25
Distribution of residuals:
    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 -10.550  -4.000  -1.545   0.000   1.438  43.450
node), split, n, deviance, yval
      * denotes terminal node

1) root 28 2689.0 12.86
  2) PDA<1.5 20 2457.0 14.50
    4) Activity.Field:biology,geography,maths,maths/computer
         science,philosophy,secretariat 9  112.0 12.00 *
    5) Activity.Field:computer-science 11 2243.0 16.55 *
  3) PDA>1.5 8   43.5  8.75 *
```

The derived regression tree:



Critical Probability Analysis :

X1 ≡ persons who never used a PDA before
X2 ≡ persons who already used a PDA at least once before

X1	(X1-Mean(X1))	$(X1\text{-Mean}(X1))^2$	X2	(X2-Mean(X2))	$(X2\text{-Mean}(X2))^2$
10	-4,6	21,16	13	4,25	18,0625
12	-2,6	6,76	6	-2,75	7,5625
21	6,4	40,96	11	2,25	5,0625
10	-4,6	21,16	6	-2,75	7,5625
15	0,4	0,16	9	0,25	0,0625
9	-5,6	31,36	7	-1,75	3,0625
11	-3,6	12,96	10	1,25	1,5625
12	-2,6	6,76	8	-0,75	0,5625
60	45,4	2061,16			
15	0,4	0,16			
15	0,4	0,16			
10	-4,6	21,16			
6	-8,6	73,96			
20	5,4	29,16			
14	-0,6	0,36			
8	-6,6	43,56			
8	-6,6	43,56			
10	-4,6	21,16			
12	-2,6	6,76			
14	-0,6	0,36			

n1 = 20
n2 = 8
Mean(X1) = 14,6
Mean(X2) = 8,75

Calculation of sp

Σ $(X1\text{-Mean}(X1))^2$ = 2442,8
Σ $(X2\text{-Mean}(X2))^2$ = 43,5
ddl = ((n1-1)+(n2-1)) = 26

(Σ $(X1\text{-Mean}(X1))^2$ + Σ $(X2\text{-Mean}(X2))^2$) = 2486,3
sp^2 = (Σ $(X1\text{-Mean}(X1))^2$ + Σ $(X1\text{-Mean}(X1))^2$ / ddl = 95,62692308

Calculation of the Interval of Confidence

Mean(X1) - Mean(X2) = 5,85
t.025 (for current ddl) = 2,06
sp = 9,778901936
(1/n1) = 0,05
(1/n2) = 0,125
(1/n1)+(1/n2) = 0,175
SquareRoot((1/n1)+(1/n2)) = 0,418330013
sp*SquareRoot((1/n1)+(1/n2)) = 4,090808177
t.025*sp*SquareRoot((1/n1)+(1/n2)) = 8,427064844

Hence (μ1-μ2) is between -2,577064844 and 14,27706484.

Calculation of the Critical Probability

t = (Mean(X1)-Mean(X2)) / (sp*SquareRoot((1/n1)+(1/n2))) = 1,430035276

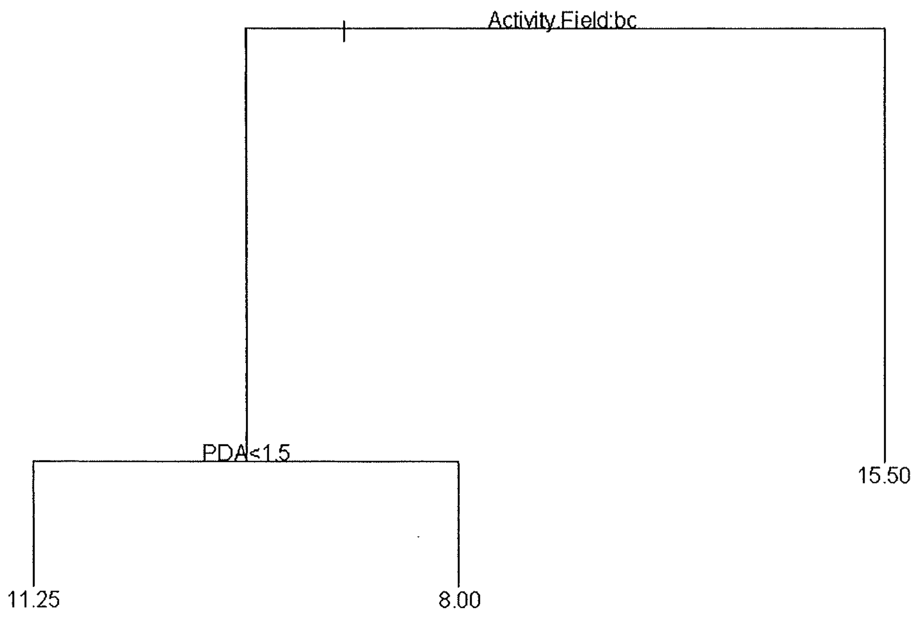
For the current ddl, the Critical Values Table For the t Distribution of Student gives us :
t> t.10, which means pc < .10

Scenario 2b

Explanation of the time necessary to accomplish the Scenario2b based on the use of Cell Phone, PDA and Video Games, as well as the Musical Formation, the Sex and the Field of activity :

```
Regression tree:
tree(formula = Scenario2b.time.in.sec ~ Sex + Activity.Field +
       Musical.Formation + GSM + PDA + Video.Games, data =
       resultats.questionnaire.ravi...all.quantitative, na.action =
       na.exclude, mincut = 5, minsize = 10, mindev = 0.01)
Variables actually used in tree construction:
[1] "Activity.Field" "PDA"
Number of terminal nodes:  3
Residual mean deviance:  27.05 = 676.3 / 25
Distribution of residuals:
     Min.   1st Qu.    Median     Mean   3rd Qu.      Max.
  -7.5000  -2.2500  -1.0000   0.0000   0.8125  18.5000
node), split, n, deviance, yval
      * denotes terminal node

1) root 28 903.0 11.54
  2) Activity.Field:computer-science,geography 20 208.9  9.95
    4) PDA<1.5 12 124.3 11.25 *
    5) PDA>1.5 8  34.0  8.00 *
  3) Activity.Field:biology,maths,maths/computer-
                 science,philosophy,secretariat 8 518.0 15.50 *
```

The derived regression tree:



Critical Probability Analysis :

X1 ≡ persons who never used a PDA before
X2 ≡ persons who already used a PDA at least once before

X1	(X1-Mean(X1))	$(X1\text{-Mean}(X1))^2$	X2	(X2-Mean(X2))	$(X2\text{-Mean}(X2))^2$
14	1,05	1,1025	11	3	9
6	-6,95	48,3025	9	1	1
11	-1,95	3,8025	7	-1	1
12	-0,95	0,9025	7	-1	1
9	-3,95	15,6025	11	3	9
12	-0,95	0,9025	5	-3	9
11	-1,95	3,8025	6	-2	4
17	4,05	16,4025	8	0	0
11	-1,95	3,8025			
18	5,05	25,5025			
9	-3,95	15,6025			
10	-2,95	8,7025			
9	-3,95	15,6025			
10	-2,95	8,7025			
10	-2,95	8,7025			
8	-4,95	24,5025			
14	1,05	1,1025			
12	-0,95	0,9025			
22	9,05	81,9025			
34	21,05	443,1025			

n1 = 20
n2 = 8
Mean(X1) = 12,95
Mean(X2) = 8

Calculation of sp

Σ $(X1\text{-Mean}(X1))^2$ = 728,95
Σ $(X2\text{-Mean}(X2))^2$ = 34
ddl = ((n1-1)+(n2-1)) = 26

(Σ $(X1\text{-Mean}(X1))^2$ + Σ $(X2\text{-Mean}(X2))^2$) = 762,95
sp^2 = (Σ $(X1\text{-Mean}(X1))^2$ + Σ $(X1\text{-Mean}(X1))^2$ / ddl = 29,34423077

Calculation of the Interval of Confidence

Mean(X1) - Mean(X2) = 4,95
t.025 (for current ddl) = 2,06
sp = 5,417031546
(1/n1) = 0,05
(1/n2) = 0,125
(1/n1)+(1/n2) = 0,175
SquareRoot((1/n1)+(1/n2)) = 0,418330013
sp*SquareRoot((1/n1)+(1/n2)) = 2,266106878
t.025*sp*SquareRoot((1/n1)+(1/n2)) = 4,66818017

Hence ($\mu 1$-$\mu 2$) is between 0,28181983 and 9,61818017.

Calculation of the Critical Probability

t = (Mean(X1)-Mean(X2)) / (sp*SquareRoot((1/n1)+(1/n2))) = 2,184362991

For the current ddl, the Critical Values Table For the t Distribution of Student gives us :
$t > t_{.025}$, which means $pc < .025$

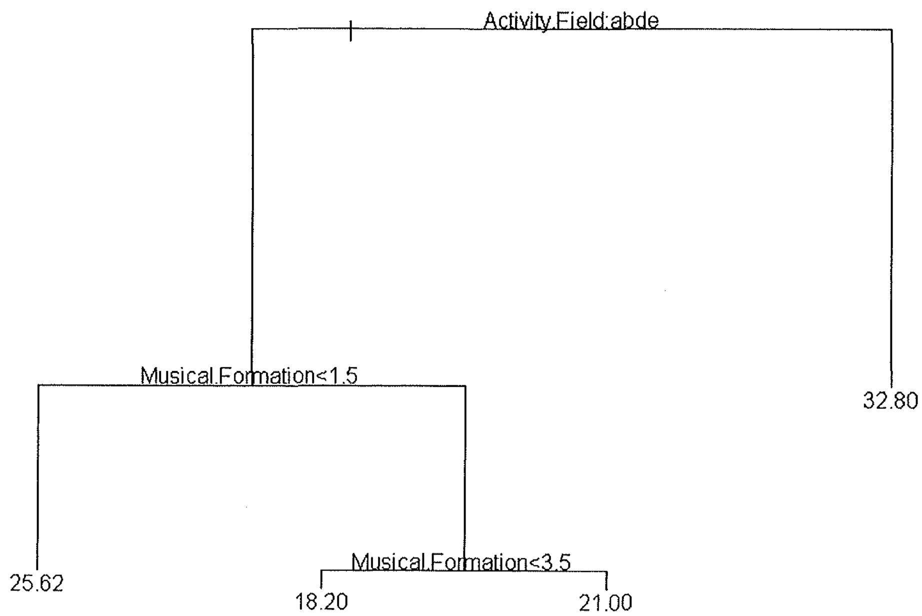
D.2.3 Scenario 3

Scenario 3a

Explanation of the time necessary to accomplish the Scenario3a based on the use of Cell Phone, PDA and Video Games, as well as the Musical Formation, the Sex and the field of activity :

```
Regression tree:
tree(formula = Scenario3a.time.in.sec ~ Sex + Activity.Field +
      Musical.Formation + GSM + PDA + Video.Games, data =
      resultats.questionnaire.ravi...all.quantitative, na.action =
      na.exclude, mincut = 5, minsize = 10, mindev = 0.01)
Variables actually used in tree construction:
[1] "Activity.Field"    "Musical.Formation"
Number of terminal nodes:  4
Residual mean deviance:  122.1 = 2931 / 24
Distribution of residuals:
    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 -16.800  -5.350  -2.308   0.000   3.988  26.200
node), split, n, deviance, yval
      * denotes terminal node

 1) root 28 3549.0 24.75
   2) Activity.Field:biology,computer-science,maths,maths/computer-science
                                                        23 2148.0 23.00
     4) Musical.Formation<1.5 13 1491.0 25.62 *
     5) Musical.Formation>1.5 10  452.4 19.60
      10) Musical.Formation<3.5 5  118.8 18.20 *
      11) Musical.Formation>3.5 5  314.0 21.00 *
   3) Activity.Field:geography,philosophy,secretariat 5 1007.0 32.80 *
```

The derived regression tree :



Critical Probability Analysis :

X1 ≡ persons who never used a PDA before
X2 ≡ persons who already used a PDA at least once before

X1	(X1-Mean(X1))	$(X1-Mean(X1))^2$	X2	(X2-Mean(X2))	$(X2-Mean(X2))^2$
25	-0,2	0,04	16	-7,625	58,140625
32	6,8	46,24	21	-2,625	6,890625
12	-13,2	174,24	27	3,375	11,390625
16	-9,2	84,64	33	9,375	87,890625
29	3,8	14,44	21	-2,625	6,890625
49	23,8	566,44	28	4,375	19,140625
14	-11,2	125,44	19	-4,625	21,390625
21	-4,2	17,64	24	0,375	0,140625
33	7,8	60,84			
23	-2,2	4,84			
21	-4,2	17,64			
10	-15,2	231,04			
24	-1,2	1,44			
45	19,8	392,04			
14	-11,2	125,44			
11	-14,2	201,64			
16	-9,2	84,64			
21	-4,2	17,64			
59	33,8	1142,44			
29	3,8	14,44			

n1 = 20
n2 = 8
Mean(X1) = 25,2
Mean(X2) = 23,625

Calculation of sp

Σ $(X1-Mean(X1))^2$ = 3323,2
Σ $(X2-Mean(X2))^2$ = 211,875
ddl = ((n1-1)+(n2-1)) = 26

(Σ $(X1-Mean(X1))^2$ + Σ $(X2-Mean(X2))^2$) = 3535,075
sp^2 = (Σ $(X1-Mean(X1))^2$ + Σ $(X1-Mean(X1))^2$ / ddl = 135,9644231

Calculation of the Interval of Confidence

Mean(X1) - Mean(X2) = 1,575
t.025 (for current ddl) = 2,06
sp = 11,66037834
(1/n1) = 0,05
(1/n2) = 0,125
(1/n1)+(1/n2) = 0,175
SquareRoot((1/n1)+(1/n2)) = 0,418330013
sp*SquareRoot((1/n1)+(1/n2)) = 4,877886226
t.025*sp*SquareRoot((1/n1)+(1/n2)) = 10,04844563

Hence (μ1-μ2) is between -8,473445627 and 11,62344563.

Calculation of the Critical Probability

t = (Mean(X1)-Mean(X2)) / (sp*SquareRoot((1/n1)+(1/n2))) = 0,32288576

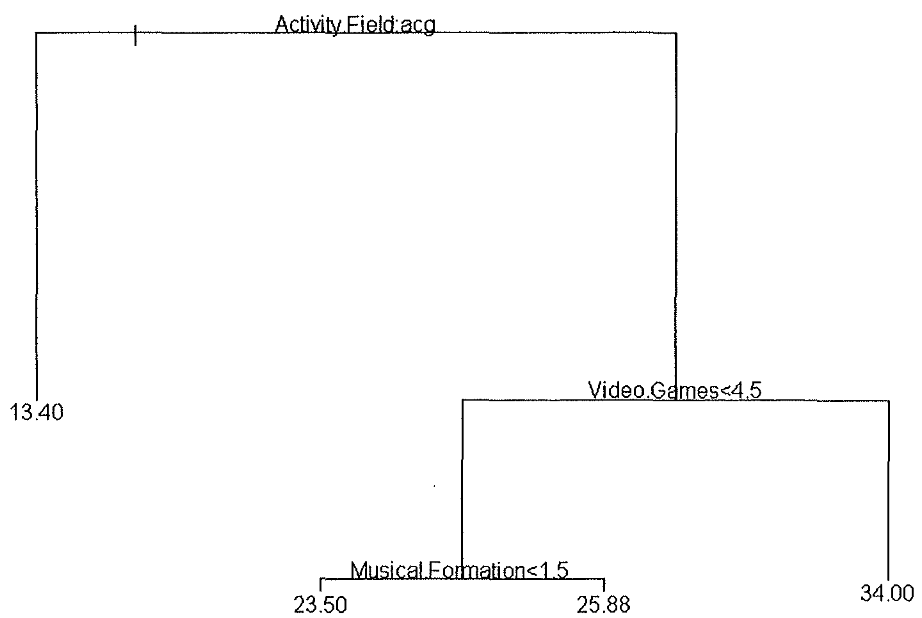
For the current ddl, the Critical Values Table For the t Distribution of Student gives us :
t < t.25, which means that pc is not smaller than .25

Scenario 3b

Explanation of the time necessary to accomplish the Scenario3b based on the use of Cell Phone, PDA and Video Games, as well as the Musical Formation, the Sex and the field of activity :

```
Regression tree:
tree(formula = Scenario3b.time.in.sec ~ Sex + Activity.Field +
       Musical.Formation + GSM + PDA + Video.Games, data =
       resultats.questionnaire.ravi...all.quantitative, na.action =
       na.exclude, mincut = 5, minsize = 10, mindev = 0.01)
Variables actually used in tree construction:
[1] "Activity.Field"     "Video.Games"        "Musical.Formation"
Number of terminal nodes:  4
Residual mean deviance:  224.7 = 5393 / 24
Distribution of residuals:
    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 -22.000  -7.750  -0.500   0.000   4.375  41.000
node), split, n, deviance, yval
      * denotes terminal node

 1) root 28 6483.0 24.25
   2) Activity.Field:biology,geography,secretariat  5  137.2 13.40 *
   3) Activity.Field:computer-science,maths,maths/computer-science,philosophy
                                                         23 5629.0 26.61
     6) Video.Games<4.5 18 2822.0 24.56
      12) Musical.Formation<1.5 10 1947.0 23.50 *
      13) Musical.Formation>1.5 8  850.9 25.88 *
     7) Video.Games>4.5 5 2458.0 34.00 *
```

The derived regression tree:



Critical Probability Analysis :

X1 ≡ persons who never used a PDA before
X2 ≡ persons who already used a PDA at least once before

X1	(X1-Mean(X1))	(X1-Mean(X1))²	X2	(X2-Mean(X2))	(X2-Mean(X2))²
15	-10,9	118,81	22	1,875	3,515625
13	-12,9	166,41	26	5,875	34,515625
46	20,1	404,01	17	-3,125	9,765625
15	-10,9	118,81	7	-13,125	172,265625
16	-9,9	98,01	58	37,875	1434,515625
32	6,1	37,21	11	-9,125	83,265625
32	6,1	37,21	8	-12,125	147,015625
30	4,1	16,81	12	-8,125	66,015625
25	-0,9	0,81			
23	-2,9	8,41			
17	-8,9	79,21			
75	49,1	2410,81			
8	-17,9	320,41			
23	-2,9	8,41			
31	5,1	26,01			
30	4,1	16,81			
23	-2,9	8,41			
34	8,1	65,61			
24	-1,9	3,61			
6	-19,9	396,01			

n1 = 20
n2 = 8
Mean(X1) = 25,9
Mean(X2) = 20,125

Calculation of sp

Σ (X1-Mean(X1))² = 4341,8
Σ (X2-Mean(X2))² = 1950,875
ddl = ((n1-1)+(n2-1)) = 26

(Σ (X1-Mean(X1))² + Σ (X2-Mean(X2))²) = 6292,675
sp² = (Σ (X1-Mean(X1))² + Σ (X1-Mean(X1))² / ddl = 242,0259615

Calculation of the Interval of Confidence

Mean(X1) - Mean(X2) = 5,775
t.025 (for current ddl) = 2,06
sp = 15,5571836
(1/n1) = 0,05
(1/n2) = 0,125
(1/n1)+(1/n2) = 0,175
SquareRoot((1/n1)+(1/n2)) = 0,418330013
sp*SquareRoot((1/n1)+(1/n2)) = 6,508036821
t.025*sp*SquareRoot((1/n1)+(1/n2)) = 13,40655585

Hence (μ1-μ2) is between -7,631555852 and 19,18155585

Calculation of the Critical Probability

t = (Mean(X1)-Mean(X2)) / (sp*SquareRoot((1/n1)+(1/n2))) = 0,887364371

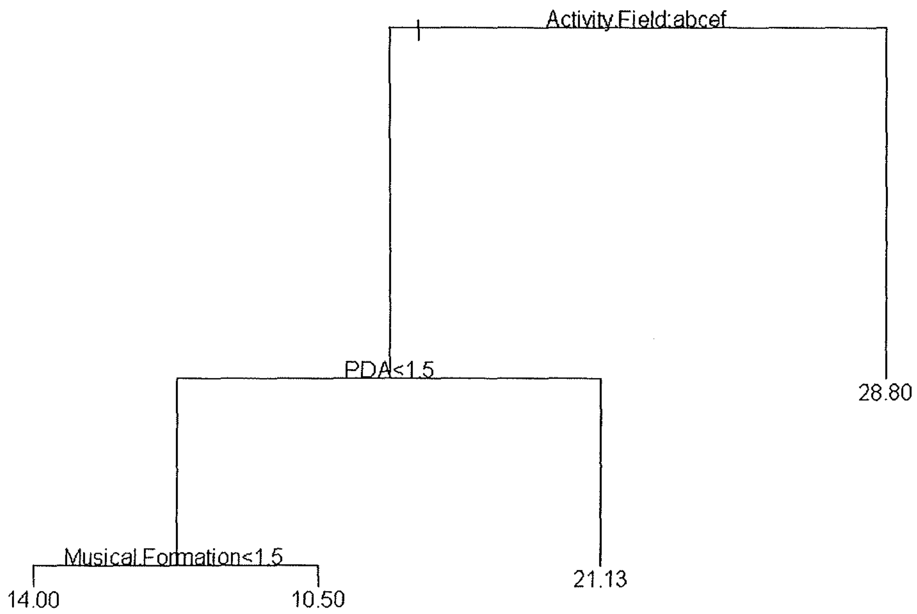
For the current ddl, the Critical Values Table For the t Distribution of Student gives us :
t > t.25, which means pc < .25

Scenario 3c explained with all the explicative variables

Explanation of the time necessary to accomplish the Scenario3c based on the use of Cell Phone, PDA and Video Games, as well as the Musical Formation, the Sex and the field of activity :

```
Regression tree:
tree(formula = Scenario3c.time.in.sec ~ Sex + Activity.Field +
      Musical.Formation + GSM + PDA + Video.Games, data =
      resultats.questionnaire.ravi...all.quantitative, na.action =
      na.exclude, mincut = 5, minsize = 10, mindev = 0.01)
Variables actually used in tree construction:
[1] "Activity.Field"    "PDA"              "Musical.Formation"
Number of terminal nodes:  4
Residual mean deviance:  239.5 = 5747 / 24
Distribution of residuals:
    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 -19.800  -9.375  -2.750   0.000   4.750  44.880
node), split, n, deviance, yval
      * denotes terminal node

1) root 28 6890.0 17.93
  2) Activity.Field:biology,computer-science,geography,maths/computer-
               science,philosophy 23 4156.0 15.57
    4) PDA<1.5 15  367.6 12.60
      8) Musical.Formation<1.5 9  226.0 14.00 *
      9) Musical.Formation>1.5 6   97.5 10.50 *
    5) PDA>1.5 8 3409.0 21.13 *
  3) Activity.Field:maths,secretariat 5 2015.0 28.80 *
```

The derived regression tree:



Critical Probability Analysis :

X1 ≡ persons who never used a PDA before
X2 ≡ persons who already used a PDA at least once before

X1	(X1-Mean(X1))	$(X1-Mean(X1))^2$	X2	(X2-Mean(X2))	$(X2-Mean(X2))^2$
15	-10,9	118,81	22	1,875	3,515625
13	-12,9	166,41	26	5,875	34,515625
46	20,1	404,01	17	-3,125	9,765625
15	-10,9	118,81	7	-13,125	172,265625
16	-9,9	98,01	58	37,875	1434,515625
32	6,1	37,21	11	-9,125	83,265625
32	6,1	37,21	8	-12,125	147,015625
30	4,1	16,81	12	-8,125	66,015625
25	-0,9	0,81			
23	-2,9	8,41			
17	-8,9	79,21			
75	49,1	2410,81			
8	-17,9	320,41			
23	-2,9	8,41			
31	5,1	26,01			
30	4,1	16,81			
23	-2,9	8,41			
34	8,1	65,61			
24	-1,9	3,61			
6	-19,9	396,01			

n1 = 20
n2 = 8
Mean(X1) = 16,65
Mean(X2) = 21,125

Calculation of sp

Σ $(X1-Mean(X1))^2$ = 3366,55
Σ $(X2-Mean(X2))^2$ = 3408,875
ddl = ((n1-1)+(n2-1)) = 26

(Σ $(X1-Mean(X1))^2$ + Σ $(X2-Mean(X2))^2$) = 6775,425
sp^2 = (Σ $(X1-Mean(X1))^2$ + Σ $(X1-Mean(X1))^2$ / ddl = 260,5932692

Calculation of the Interval of Confidence

Mean(X1) - Mean(X2) = -4,475
t.025 (for current ddl) = 2,06
sp = 16,14290151
(1/n1) = 0,05
(1/n2) = 0,125
(1/n1)+(1/n2) = 0,175
SquareRoot((1/n1)+(1/n2)) = 0,418330013
sp*SquareRoot((1/n1)+(1/n2)) = 6,753060204
t.025*sp*SquareRoot((1/n1)+(1/n2)) = 13,91130402

Hence (μ1-μ2) is between -18,38630402 and 9,43630402

Calculation of the Critical Probability

t = (Mean(X1)-Mean(X2)) / (sp*SquareRoot((1/n1)+(1/n2))) = -0,662662536

For the current ddl, the Critical Values Table For the t Distribution of Student gives us : t < t.25, which means that pc is not smaller that .25.

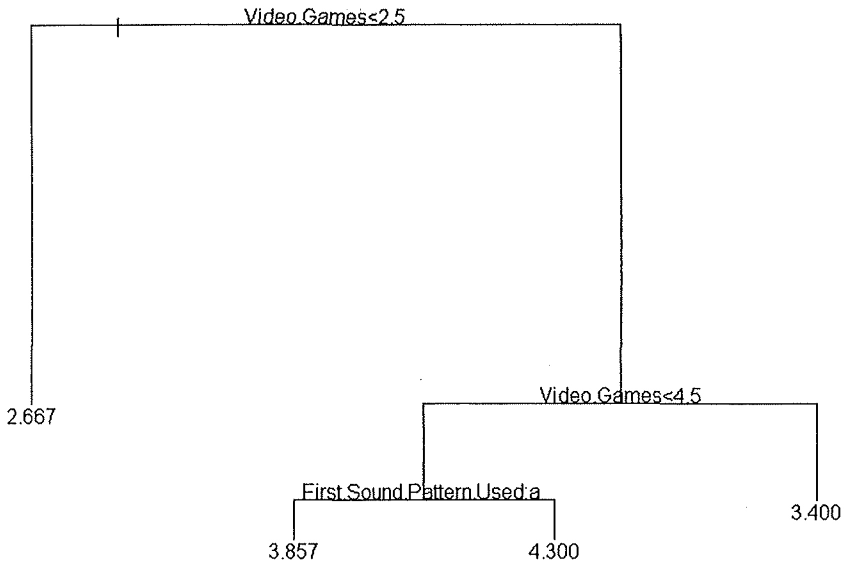
D.3 Analysis of the Sound Appreciation

D.3.1 Auditory Icons

Explanation of the Auditory Icon pattern appreciation based on the use of Cell Phone, PDA and Video Games, as well as the Musical Formation, the Sex and the field of activity as well as the First Sound Pattern used for the scenarios :

```
Regression tree:
tree(formula = Auditory.Icons ~ Sex + Activity.Field + Musical.Formation +
       GSM + PDA + Video.Games + First.Sound.Pattern.Used, data =
       resultats.questionnaire.ravi...all.quantitative, na.action =
       na.exclude, mincut = 5, minsize = 10, mindev = 0.01)
Variables actually used in tree construction:
[1] "Video.Games"              "First.Sound.Pattern.Used"
Number of terminal nodes:  4
Residual mean deviance:  0.8121 = 19.49 / 24
Distribution of residuals:
    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 -1.6670 -0.4667  0.1429  0.0000  0.6000  2.3330
node), split, n, deviance, yval
      * denotes terminal node

 1) root 28 30.1100 3.679
   2) Video.Games<2.5 6 11.3300 2.667 *
   3) Video.Games>2.5 22 10.9500 3.955
     6) Video.Games<4.5 17  5.7650 4.118
      12) First.Sound.Pattern.Used:auditory icons 7  0.8571 3.857 *
      13) First.Sound.Pattern.Used:earcons 10  4.1000 4.300 *
     7) Video.Games>4.5 5  3.2000 3.400 *
```

The derived classification tree :

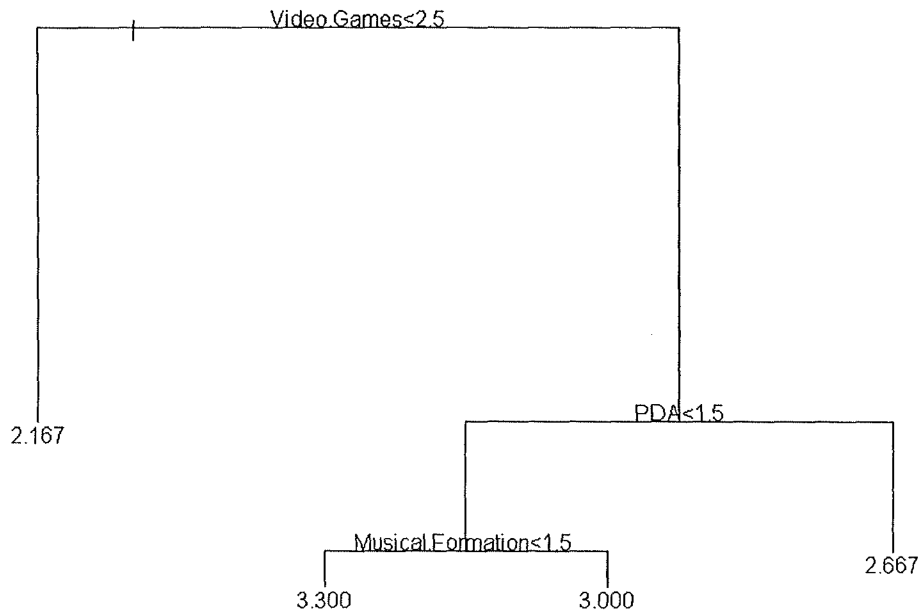


D.3.2 Earcons

Explanation of the Earcon pattern appreciation based on the use of Cell Phone, PDA and Video Games, as well as the Musical Formation, the Sex and the field of activity as well as the First Sound Pattern used for the scenarios :

```
Regression tree:
tree(formula = Earcons ~ Sex + Activity.Field + Musical.Formation + GSM + PDA
       + Video.Games + First.Sound.Pattern.Used, data =
       resultats.questionnaire.ravi...all.quantitative, na.action =
       na.exclude, mincut = 5, minsize = 10, mindev = 0.01)
Variables actually used in tree construction:
[1] "Video.Games"       "PDA"                  "Musical.Formation"
Number of terminal nodes:  4
Residual mean deviance:  0.4278 = 10.27 / 24
Distribution of residuals:
    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 -1.3000 -0.3000  0.0000  0.0000  0.4250  0.8333
node), split, n, deviance, yval
      * denotes terminal node

 1) root 28 15.430 2.857
   2) Video.Games<2.5 6  4.833 2.167 *
   3) Video.Games>2.5 22  6.955 3.045
     6) PDA<1.5 16  4.437 3.188
      12) Musical.Formation<1.5 10  4.100 3.300 *
      13) Musical.Formation>1.5 6  0.000 3.000 *
     7) PDA>1.5 6  1.333 2.667 *
```

The derived classification tree :

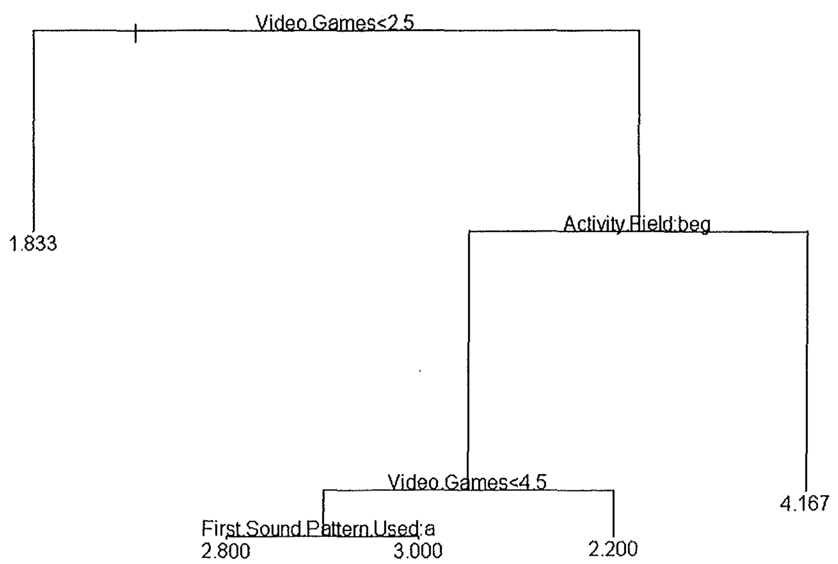


D.3.3 Sound Utility

Explanation of the Sound Utility appreciation based on the use of Cell Phone, PDA and Video Games, as well as the Musical Formation, the Sex and the field of activity as well as the First Sound Pattern used for the scenarios :

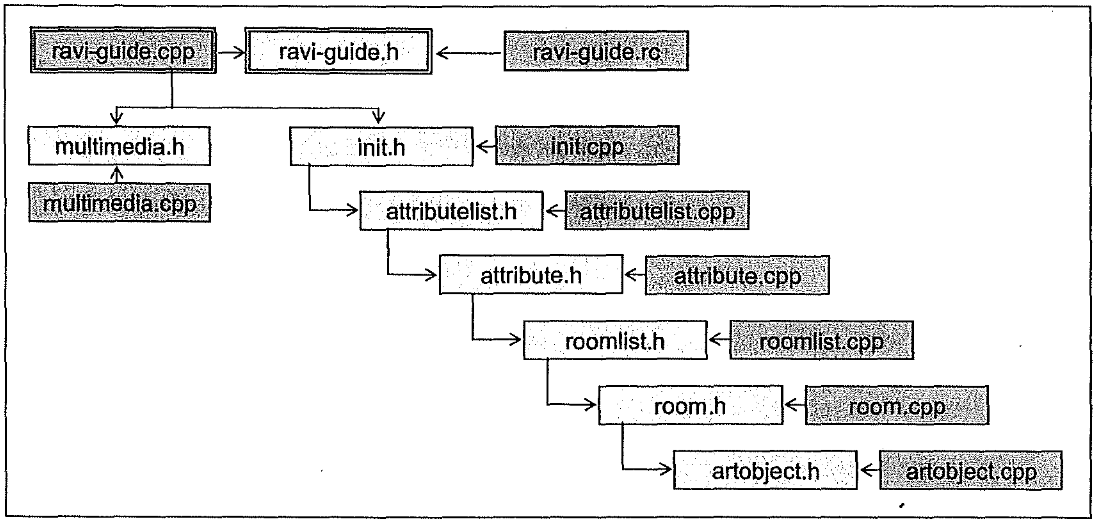
```
Regression tree:
tree(formula = Sound.Utility ~ Sex + Activity.Field + Musical.Formation + GSM
+
      PDA + Video.Games + First.Sound.Pattern.Used, data =
      resultats.questionnaire.ravi...all.quantitative, na.action =
      na.exclude, mincut = 5, minsize = 10, mindev = 0.01)
Variables actually used in tree construction:
[1] "Video.Games"             "Activity.Field"
[3] "First.Sound.Pattern.Used"
Number of terminal nodes:  5
Residual mean deviance:  0.8377 = 19.27 / 23
Distribution of residuals:
    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 -1.2000 -0.8000 -0.1667  0.0000  0.8000  2.1670
node), split, n, deviance, yval
      * denotes terminal node

 1) root 28 38.1100 2.821
   2) Video.Games<2.5 6  8.8330 1.833 *
   3) Video.Games>2.5 22 21.8200 3.091
     6)  Activity.Field:computer-science,maths/computer-science,secretariat  16
11.4400 2.688
      12) Video.Games<4.5 11  6.9090 2.909
        24) First.Sound.Pattern.Used:auditory icons 5  4.8000 2.800 *
        25) First.Sound.Pattern.Used:earcons 6  2.0000 3.000 *
      13) Video.Games>4.5 5  2.8000 2.200 *
     7) Activity.Field:biology,geography,maths 6  0.8333 4.167 *
```

The derived classification tree :



Appendix E : Source Code

The following pages present the code listings for each class used in our application. We remind you that the application used the following class hierarchy :



The main functionnalities of each has been exposed in Chapter 7. Nevertheless, you will find many comments to understand more easily specific parts of the code.

E.1 ravi-guide.h

```
// ==================================================================
// HEADER FILE for ravi-guide.cpp
// ==================================================================

#include "init.h"
// "init.h"              includes "attributelist.h"
// "attributelist.h" includes "attribute.h"
// "attribute.h"     includes "room.h"
// "room.h"              includes "artobject.h"

// ----------------------------------------------------------------

// Returns number of elements :
#define dim(x) (sizeof(x) / sizeof(x[0]))

// ----------------------------------------------------------------
// Generic definitions and data types
//

// This structure associates messages with a function
struct decodeUINT
{
	UINT Code;
	LRESULT (*Fxn)(HWND,UINT,WPARAM,LPARAM);
};

// This structure associates menu IDs with a function
struct decodeCMD
{
	UINT Code;
	LRESULT (*Fxn)(HWND,WORD,HWND,WORD);
};

typedef struct
{
	TCHAR *szClass;
	INT nID;
	TCHAR *szTitle;
	INT x;
	INT y;
	INT cx;
	INT cy;
	DWORD lStyle;
} CTLWNDSTRUCT, *PCTLWNDSTRUCT;

typedef struct
{
	TCHAR *pszLabel;
	WORD wNotification;
} NOTELABELS, *PNOTELABELS;

// ----------------------------------------------------------------
// Generic definitions used by the application
// ----------------------------------------------------------------

// Command bar ID
#define IDC_CMDBAR                              1

//Main Menu Resources and Item ID -> see ravi-guide.rc
#define ID_MENU                                 11
#define ID_ACCEL                                12
```

```
//Command Bar Menus ID
#define IDM_NEWTOUR                              100
#define IDM_RESTART_TOUR                         101
#define IDM_END_TOUR                             102
#define IDM_EXIT                                 103
#define IDM_SOUND_ON                             104
#define IDM_AUDIT_PATTERN_ON                     105
#define IDM_EARCON_PATTERN_ON                    106
#define IDM_SOUND_OFF                            107
#define IDM_SIMULATE_IR                          108
#define IDM_ABOUT                                199

//Select Tour Mode Menu ID -> see ravi-guide.rc
#define IDD_PLEASESELECTTOURMODE                 220
#define IDD_RADIO_FREE                           221
#define IDD_RADIO_CUSTOMIZED                     222

#define BTN_SELECTTOURMODE_CONTINUE              223

//[CustomizedTourMode] Select Tour By Category
#define IDD_PLEASESELECTTOURBYCAT                230
#define IDD_ARTOBJECT_ATTRIBUTE1                 231
#define IDD_ARTOBJECT_ATTRIBUTE2                 232
#define IDD_ARTOBJECT_ATTRIBUTE3                 233
#define IDD_ARTOBJECT_ATTRIBUTE4                 234
#define IDD_ARTOBJECT_ATTRIBUTE5                 235

#define BTN_SELECTTOURBYCATEGORY_BACK            236
#define BTN_SELECTTOURBYCATEGORY_CONTINUE        237

//[CustomizedTourMode] Select Sub Category
#define IDD_PLEASESELECTSUBCAT                   240
#define IDC_SUBCATEGORYLIST                      241

#define BTN_SELECTSUBCATEGORY_BACK               242
#define BTN_SELECTSUBCATEGORY_CONTINUE           243

//[CustomizedTourMode] Select Sub Category Details
#define IDD_PLEASESELECTSUBCATDETAILS            250
#define IDC_SUBCATDETAILSLIST                    251

#define BTN_SELECTSUBCATDETAILS_SELECTALL        252
#define BTN_SELECTSUBCATDETAILS_DESELECTALL      253

#define BTN_SELECTSUBCATDETAILS_BACK             254
#define BTN_SELECTSUBCATDETAILS_CONTINUE         255

//Simulate IR signal
#define IDD_PLEASESELECTROOMIR                   260
#define IDC_ROOMIRYLIST                          261
#define IDD_SELECTROOMIR_SEND                    262
#define IDD_SELECTROOMIR_CANCEL                  263

// Wave ID
#define IDR_AUDIT_WAVE_CONFIRM                   300
#define IDR_AUDIT_WAVE_CANCEL                    301
#define IDR_AUDIT_WAVE_RADIOBTN                  302
#define IDR_AUDIT_WAVE_ROOM2OVERVIEW             303
#define IDR_AUDIT_WAVE_OVERVIEW2ROOM             304
#define IDR_AUDIT_WAVE_ROOM2ROOM                 305
#define IDR_AUDIT_WAVE_ARTOBJECT                 306

#define IDR_EARCON_WAVE_CONFIRM                  400
#define IDR_EARCON_WAVE_CANCEL                   401
#define IDR_EARCON_WAVE_RADIOBTN                 402
```

```
#define IDR_EARCON_WAVE_ROOM2OVERVIEW                    403
#define IDR_EARCON_WAVE_OVERVIEW2ROOM                    404
#define IDR_EARCON_WAVE_ROOM2ROOM                        405
#define IDR_EARCON_WAVE_ARTOBJECT                        406

// Bitmaps ID

#define IDR_DEFAULT_BMP                                  500

// Definitions for te MyCreateHatchBrush (used to paint the rooms)

HBRUSH CreateCustomHatchBrush(INT fnStyle, COLORREF clrrefbackgrnd, COLORREF
clrrefforegrnd);

typedef struct
{
      BITMAPINFOHEADER bmi;
      COLORREF dwPal[2];
      BYTE bBits[64];
} BRUSHBMP;

#define HS_HORIZONTAL           0             /* ---- */
#define HS_VERTICAL             1             /* |||| */
#define HS_FDIAGONAL            2             /* \\\\ */
#define HS_BDIAGONAL            3             /* //// */
#define HS_CROSS                4             /* ++++ */
#define HS_DIAGCROSS            5             /* xxxx */

// ----------------------------------------------------------------
// Function prototypes for main window
// ----------------------------------------------------------------

int InitApp (HINSTANCE);
HWND InitInstance (HINSTANCE,LPWSTR,int);
int TermInstance (HINSTANCE, int);

int ListenIrThread(PVOID pArg);

// Window procedures
LRESULT CALLBACK MainWndProc (HWND,UINT,WPARAM,LPARAM);

// Message handlers
LRESULT DoCreateMain (HWND,UINT,WPARAM,LPARAM);
LRESULT DoPaintMain (HWND,UINT,WPARAM,LPARAM);
LRESULT DoHibernateMain (HWND,UINT,WPARAM,LPARAM);
LRESULT DoActivateMain (HWND,UINT,WPARAM,LPARAM);
LRESULT DoDestroyMain (HWND,UINT,WPARAM,LPARAM);

LRESULT DoManageStylusInput (HWND,UINT,WPARAM,LPARAM);

LRESULT DoCommandMain (HWND,UINT,WPARAM,LPARAM);
LRESULT DoInitMenuPopMain (HWND,UINT,WPARAM,LPARAM);

// Command functions
LPARAM DoMainCommandNewTour(HWND,WORD,HWND,WORD);
LPARAM DoMainCommandRestartTour(HWND,WORD,HWND,WORD);
LPARAM DoMainCommandEndTour(HWND,WORD,HWND,WORD);
LPARAM DoMainCommandExit(HWND,WORD,HWND,WORD);
LPARAM DoMainCommandAuditPatternOn(HWND,WORD,HWND,WORD);
LPARAM DoMainCommandEarconPatternOn(HWND,WORD,HWND,WORD);
LPARAM DoMainCommandSoundOff(HWND,WORD,HWND,WORD);
LPARAM DoMainCommandSimulateIrSignal(HWND,WORD,HWND,WORD);
LPARAM DoMainCommandAbout(HWND,WORD,HWND,WORD);

// Dialog Box Procedures
```

```
BOOL CALLBACK AboutDlgProc (HWND, UINT, WPARAM, LPARAM);
BOOL CALLBACK SimulateIrSignalDlgProc (HWND, UINT, WPARAM, LPARAM);
BOOL CALLBACK SelectTourModeDlgProc (HWND, UINT, WPARAM, LPARAM);
BOOL CALLBACK SelectTourByCategoryDlgProc (HWND, UINT, WPARAM, LPARAM);
BOOL CALLBACK SelectSubCategoryDlgProc(HWND, UINT, WPARAM, LPARAM);
BOOL CALLBACK SelectSubCatDetailsDlgProc(HWND, UINT, WPARAM, LPARAM);

// Dialog "Wrapping" procedures
// -> used to manage the [<Back] and [Next>] buttons of each dialog box
BOOL CALLBACK SelectTourModeWrapProc (HWND hWnd);
BOOL CALLBACK SelectTourByCategoryWrapProc (HWND hWnd);
BOOL CALLBACK SelectSubCategoryWrapProc(HWND hWnd);
BOOL CALLBACK SelectSubCatDetailsWrapProc(HWND hWnd);
```

E.2 ravi-guide.cpp

```
// ================================================================
// ravi-guide.cpp
// ================================================================
// Note : all page references are done to :
// [Boling2001] Douglas Boling, "Programming Windows CE (Second Edition)",
//  Microsoft Press, June 2001.
// ----------------------------------------------------------------

#include <windows.h>      // for all the windows stuff
#include <commctrl.h>     // to include command bar
#include <stdio.h>        // for string manipulation

#include "tchar.h"        // for TCHAR routing mapping
                          // -> enables to draw dynmaic text with DrawText

#include "Commdlg.h"      // to include the OPENFILENAME structure
                          // -> enables to create an open file dialog box!

#include "ravi-guide.h"   // to include program-specific stuff
                          //(and all custon class headers!)
#include "multimedia.h"   // to handle (play, display, ...)
                          //multimedia resources

// ----------------------------------------------------------------
//
// Global Data
//
// ----------------------------------------------------------------

TCHAR TszTopFolder[MAX_PATH];
TCHAR TszFileName[100];

RoomList      *pFirstRoomInList=new RoomList(); //first room from the room list
Room          *pActualRoom;                    //room actually displayed
Room          *pIrRoom;                        //room actually signaled by IR
ArtObject     *pActualArtObject;
AttributeList* pAttributeList=new AttributeList();
Attribute     * pActualAttribute;

float lOverviewScale;
float lRoomScale;
POINT mapCanevasTopLeft;
POINT mapCanevasBottomRight;

const TCHAR szAppName[] = TEXT("Ravi's Guide");

int iTaskLevel;
//0==Pre-tour task level
//1==Overview map task level
//2==Room map task level
//3==Object info

int iPreviousTaskLevel;
//0==Pre-tour task level
//1==Overview map task level
//2==Room map task level
//3==Object info

int iReOpen;
//0==The user is not retrying to open a file
//1==The user is retrying to open a file
```

```
int iAskUserResult;
//0==User answered No or cancelled
//1==User answeres Yes

int iTourMode;
//0==FreeTourMode
//1==CustmizedTourMode

int iSoundOn;
//0==Sound is off
//1==Auditory Icon Sound Pattern is on
//2==Earcon Sound Pattern is on

int iDrawNew;
// Used to avoid playing a sound in the room when redrawing the same room, or
// the same object info screen, which occurs when we receive an IR signal from
// the room the last room IR signaled.
// 0==we must draw a new room
// 1==we are redrawing the same room

// program instance handle
HINSTANCE hInst;

//application window's client rectangle
RECT clientRect;

//thread values (note that the ListenIrThread is initialized in InitInstance)
BOOL fContinue=TRUE;
HANDLE hListenIrEvent=INVALID_HANDLE_VALUE;
TCHAR szIrMsg[MAX_STRINGSIZE];

// Message dispatch table for MainWindowProc
const struct decodeUINT MainMessages[] =
{
        WM_CREATE,          DoCreateMain,
        WM_PAINT,           DoPaintMain,
        WM_HIBERNATE,       DoHibernateMain,
        WM_ACTIVATE,        DoActivateMain,
        WM_DESTROY,         DoDestroyMain,

        WM_LBUTTONUP,       DoManageStylusInput,

        WM_COMMAND,         DoCommandMain,
        WM_INITMENUPOPUP,   DoInitMenuPopMain,
};

//Command Message dispatch for MainWindowProc
const struct decodeCMD MainCommandItems[] =
{
        IDM_NEWTOUR,                DoMainCommandNewTour,
        IDM_RESTART_TOUR,           DoMainCommandRestartTour,
        IDM_END_TOUR,               DoMainCommandEndTour,
        IDM_EXIT,                   DoMainCommandExit,
        IDM_AUDIT_PATTERN_ON,       DoMainCommandAuditPatternOn,
        IDM_EARCON_PATTERN_ON,      DoMainCommandEarconPatternOn,
        IDM_SOUND_OFF,              DoMainCommandSoundOff,
        IDM_SIMULATE_IR,            DoMainCommandSimulateIrSignal,
        IDM_ABOUT,                  DoMainCommandAbout,
};

// ============================================================
//
// PROGRAM ENTRY POINT
//
// ============================================================
```

```
int WINAPI WinMain( HINSTANCE hInstance,
                                 HINSTANCE hPrevInstance,
                                 LPTSTR    lpCmdLine,
                                 int       nCmdShow)
{
       MSG msg;
       int rc=0;
       HWND hwndMain;
       HACCEL hAccel;
       iTaskLevel=0;
       iPreviousTaskLevel=0;
       iSoundOn=0;
       iDrawNew=1;
       iReOpen=0;

       //Initialize the application.
       rc = InitApp(hInstance);
       if (rc) return rc;

       //Initialize this instance.
       hwndMain = InitInstance(hInstance,lpCmdLine,nCmdShow);
       if (hwndMain==0) return 0x10;

       //Load Accelerator table
       hAccel = LoadAccelerators(hInst,MAKEINTRESOURCE(ID_ACCEL));

       //Application message loop
       while (GetMessage (&msg,NULL,0,0))
       {
              //translate accelerators
              if (!TranslateAccelerator (hwndMain,hAccel,&msg))
              {
                     TranslateMessage(&msg);
                     DispatchMessage(&msg);
              }
       }

       //Instance cleanup
       return TermInstance (hInstance, msg.wParam);
}

// ------------------------------------------------------------------
//
// InitApp : Application Initialization [STANDARD PROCEDURE]
//
// ------------------------------------------------------------------

int InitApp(HINSTANCE hInstance)
{
       WNDCLASS wc;

       #if defined(WIN32_PLATFORM_PSPC)
              //if Pocket PC, allow only one instance of the application
              HWND hWnd = FindWindow(szAppName,NULL);
              if (hWnd)
              {
                     SetForegroundWindow((HWND)(((DWORD)hWnd)|0x01));
                     return -1;
              }
       #endif

       //Register application main window class
       wc.style=0;                              //window style
       wc.lpfnWndProc=MainWndProc;              //callback function
```

```
       wc.cbClsExtra=0;                       //extra class data
       wc.cbWndExtra=0;                       //extra window data
       wc.hInstance=hInstance;                //owner handle
       wc.hIcon=NULL;                         //application icon
       wc.hCursor=LoadCursor(NULL,IDC_ARROW); //default cursor
       wc.hbrBackground=(HBRUSH)GetStockObject(WHITE_BRUSH);
       wc.lpszMenuName=NULL;                  //menu name
       wc.lpszClassName=szAppName;            //window class name

       if (RegisterClass (&wc) ==0) return 1;

       return 0;
}

// ------------------------------------------------------------------
//
// InitInstance : Instance Initialization [STANDARD PROCEDURE]
//
// ------------------------------------------------------------------

HWND InitInstance (HINSTANCE hInstance,LPWSTR lpCmdLine,int nCmdShow)
{
       HWND hWnd;
       HANDLE hListenIrThread;

       //Save program instance handle in global variable
       hInst = hInstance;

       //Create main window
       hWnd = CreateWindow(szAppName,          //window class
                    TEXT("Ravi's Guide"),//window title
                    WS_VISIBLE,          //style flag
                    CW_USEDEFAULT,       //x position
                    CW_USEDEFAULT,       //y position
                    CW_USEDEFAULT,       //initial width
                    CW_USEDEFAULT,       //initial height
                    NULL,                //parent
                    NULL,                //Menu -> must be null
                    hInstance,           //Application instance
                    NULL);               //pointer to create parameters

       if (!IsWindow(hWnd)) return 0;         //fail if not created

//////////////
//////////////
//////////////

       // ============================================
       // Launch the ListenIrThread
       // ============================================

              //create unnamed auto-reset event initially false for the
              // ListenIrThread
              hListenIrEvent = CreateEvent(NULL,FALSE,FALSE,NULL);

              //create the ListenIrThread
              hListenIrThread
               =CreateThread(NULL,0,(LPTHREAD_START_ROUTINE)ListenIrThread,
                           hWnd,0,NULL);
              if (hListenIrThread)
              {
                     CloseHandle(hListenIrThread);
              }
              else
              {
```

```
                    DestroyWindow (hWnd);
                    return 0;
            }
        // ============================================

//////////////
//////////////
//////////////

        //Standard show and update calls
        ShowWindow(hWnd,nCmdShow);
        UpdateWindow(hWnd);

        return hWnd;
}

// ---------------------------------------------------------------
//
// TermInstance : Program cleanup [STANDARD PROCEDURE]
//
// ---------------------------------------------------------------

int TermInstance (HINSTANCE hInstance, int nDefRC)
{
        fContinue = FALSE;

        if (hListenIrEvent!=INVALID_HANDLE_VALUE)
        {
                PulseEvent(hListenIrEvent);
                Sleep(100);
                CloseHandle(hListenIrEvent);
        }

        return nDefRC;
}


// ===============================================================
//
// MESSAGE HANDLING PROCEDURES FOR MAIN WINDOW
//
// ===============================================================

// ---------------------------------------------------------------
//
// MainWndProc : callback function for application window [STANDARD PROCEDURE]
//
// ---------------------------------------------------------------

LRESULT CALLBACK MainWndProc (HWND hWnd,UINT wMsg,WPARAM wParam,LPARAM lParam)
{
        INT i;

        // Search message list to see if we need to handle this message.
        //If in list, call procedure
        for (i=0;i<dim(MainMessages);i++)
        {
                if (wMsg==MainMessages[i].Code)
                        return (*MainMessages[i].Fxn)(hWnd,wMsg,wParam,lParam);
        }

        return DefWindowProc(hWnd,wMsg,wParam,lParam);
}
```

```
// ----------------------------------------------------------------
//
// DoCreateMain : pocess WM_CREATE message for window
//
// ----------------------------------------------------------------
// Modifications should only be made for changing the menus.
// ----------------------------------------------------------------

LRESULT DoCreateMain (HWND hWnd,UINT wMsg,WPARAM wParam,LPARAM lParam)
{
	HWND hwndCB;

	//Create a new command bar.
	//Note that IDC_CMDBAR is a generic definition of test1.h
	hwndCB = CommandBar_Create(hInst,hWnd,IDC_CMDBAR);

	//Add the menu
	CommandBar_InsertMenubar(hwndCB,hInst,ID_MENU,0);

	//Add exit button to command bar
	CommandBar_AddAdornments(hwndCB,0,0);

  return 0;
}

// ----------------------------------------------------------------
//
// DoPaintMain : process WM_PAINT message for window
//
// ----------------------------------------------------------------
// This procedure paints the program's window
// ----------------------------------------------------------------
// To force the repainting of the window, another procedure can
// use : InvalidateRect(hWnd,NULL,TRUE)
// ----------------------------------------------------------------

LRESULT DoPaintMain (HWND hWnd,UINT wMsg,WPARAM wParam,LPARAM lParam)
{
	//drawing objects
	PAINTSTRUCT ps;
	HDC hdc;
	HBRUSH hBr,hOldBr;

	//scaling objects
	float lLengthCoeff,lHeigthCoeff;
	POINT mapMinPoint; //used for calculating map scale coeff
	POINT mapMaxPoint; //used for calculating map scale coeff

	//various objects
	int i,j;
	TCHAR TszPath[MAX_PATH];
	TCHAR TszCode[200];
	TCHAR string[250];//must be this size for displaying art object info !

	//room pointers
	RoomList *pTempRoomList;
	pTempRoomList=pFirstRoomInList;
	POINT tempRoom[MAX_COORDINATES];
	//temporary point array used to resize the rooms to fit them in
	//the clientRect
	POINT tempCnx [2];
	//temporary point array used to resize the connexions of a room
	ArtObject *pTempArtObject;
```

```
// Adjust the size of the client rectangle to take into account the
// command bar height
GetClientRect (hWnd,&clientRect);
clientRect.top +=CommandBar_Height (GetDlgItem(hWnd,IDC_CMDBAR));

hdc = BeginPaint(hWnd,&ps);

//define the canevas were the map will be drawn
mapCanevasTopLeft.x         =clientRect.left+20;
mapCanevasTopLeft.y         =clientRect.top+20;
mapCanevasBottomRight.x     =clientRect.right-20;
mapCanevasBottomRight.y     =clientRect.bottom-20;

// The information to be displayed depends on the level of the task
// currently in course.
// This task level is stored in the variable iTaskLevel.

switch (iTaskLevel)
{
        case 0: //pre-tour Task Level

                _stprintf(string,L"\n\n\n\n\n\nWelcome To %S !\n\n\n
                         Please select New Tour\nfrom the File menu to
                          begin.",PRG_VERSION);
                DrawText(hdc, string,-1,&clientRect,
                         DT_CENTER | DT_WORDBREAK);

                EndPaint(hWnd,&ps);
                break;

        case 1: // map-overview level

                // play sound only if we are not coming from :
                // * the select tour mode section
                // * the map overview (this case occurs when we need to
                // redraw the overview map because of the reception of
                //an IR signal)

                if ((iSoundOn==1)
                        && (iPreviousTaskLevel!=0)
                        && (iPreviousTaskLevel!=1))
                        PlayWaveResource (IDR_AUDIT_WAVE_ROOM2OVERVIEW,
                                          hInst,0);

                if ((iSoundOn==2)
                        && (iPreviousTaskLevel!=0)
                        && (iPreviousTaskLevel!=1))
                        PlayWaveResource (IDR_EARCON_WAVE_ROOM2OVERVIEW,
                                          hInst,0);

                if (iDrawNew==0)
                {
                        iDrawNew=1;
                }

                iPreviousTaskLevel=1;

                //1) Define Drawing Canevas

                // draw back ground canevas
                hBr = (HBRUSH) GetStockObject(WHITE_BRUSH);
                hOldBr = (HBRUSH) SelectObject(hdc,hBr);
                Rectangle(hdc,clientRect.left+5,clientRect.top+5,
                          clientRect.right-5,clientRect.bottom-5);
                SelectObject(hdc,hOldBr);
```

```
// 2)Define Scale of the Map and Write it on the screen

// Get extremal coordinates of the overview map
// By default, the mapMinPoint is (0,0) !!!
// -> see document on How to specify a building map
mapMinPoint.x=0;
mapMinPoint.y=0;
mapMaxPoint=(*pFirstRoomInList).getMaxCoordinatesInList();

// Calculate overview scale coeff
lLengthCoeff
  =(float)(mapCanevasBottomRight.x
       -mapCanevasTopLeft.x)
       /(float)(mapMaxPoint.x-mapMinPoint.x);
lHeigthCoeff
  =(float)(mapCanevasBottomRight.y
       -mapCanevasTopLeft.y)
       /(float)(mapMaxPoint.y-mapMinPoint.y);

// Calculate overview scale coeff by choosing the smallest
// value between Length and Height Coeff
if (lLengthCoeff<lHeigthCoeff)
       lOverviewScale=lLengthCoeff;
else
       lOverviewScale=lHeigthCoeff;

// Center overview map on the screen
mapCanevasTopLeft.x
  = mapCanevasTopLeft.x
       + ((mapCanevasBottomRight.x-mapCanevasTopLeft.x)
       -(LONG)((mapMaxPoint.x
              -mapMinPoint.x)*lOverviewScale))/2;
mapCanevasTopLeft.y
  = mapCanevasTopLeft.y
       + ((mapCanevasBottomRight.y-mapCanevasTopLeft.y)
       -(LONG)((mapMaxPoint.y
              -mapMinPoint.y)*lOverviewScale))/2;

//Indicate Overview Level [THANK YOU CARMINE :-)]
// * Note that "??(" == "[" and "??)" == "]"
//(see Trigraphs in the Help section of eMbedded C++)
// * Note that "\n" == new line (see Escape Sequence)
//_stprintf(string,
              L"Overview ??(Scale %f??)",lOverviewScale);
_stprintf(string,L"Overview");

SetBkMode(hdc,OPAQUE); //see p.45
DrawText(hdc,string,-1, &clientRect,
              DT_CENTER|DT_SINGLELINE);

// 3) Draw the Overview Map

while (pTempRoomList!=NULL)
{
       //test if we received an IR signal
       if (pIrRoom!=NULL)
       {
              //test if this room is signaled by the IR
              if (strcmp(
                     (*(*pTempRoomList).roomToSee).name,
                     (*pIrRoom).name)==0)
              {
                     if ((*pIrRoom).isToSee())
                     {
```

```
                        //if the room signaeld by the IR
                        //is to see, we highlight it in
                        //red/yellow pattern :
                        // * yellow for "to see"
                        // * red for IR signaled
                        hBr
                         =(HBRUSH) CreateCustomHatchBrush
                              (HS_FDIAGONAL,
                              PALETTERGB(255,230,0),
                              PALETTERGB(255,0,0));
                }

                else
                {
                        //if the room signlaed by the IR
                        //is not to see?
                        //we highlight it in red
                        hBr
                         = (HBRUSH) CreateSolidBrush
                                   (RGB(255,0,0));
                }
        }

        else //this isn't the room signaled by the IR
        {
                if
        ((*(*pTempRoomList).roomToSee).isToSee())
                {
                        // we choose yellow color for
                        // "ToSee" Rooms
                        hBr
                         =(HBRUSH)CreateSolidBrush
                              (RGB(255,230,0));
                }
                else
                {
                        hBr
                         = (HBRUSH) GetStockObject
                              (WHITE_BRUSH);
                }
        }
}

else //there is no room currently signaled by the IR
{
        if ((*(*pTempRoomList).roomToSee).isToSee())
        {
                // we choose yellow color for "ToSee"
                // Rooms
                hBr
                 =(HBRUSH)CreateSolidBrush
                              (RGB(255,230,0));
        }
        else
        {
                hBr
                 = (HBRUSH) GetStockObject
                              (WHITE_BRUSH);
        }
}

hOldBr = (HBRUSH) SelectObject(hdc,hBr);

//resize room
for (j=0;
```

```
            j<(*(*pTempRoomList).roomToSee).numberOfCoordinates;
            j++)
            {
              tempRoom[j].x
              = mapCanevasTopLeft.x +(LONG)(
                ((*(*pTempRoomList).roomToSee).coordinates[j].x)
                *lOverviewScale);

              tempRoom[j].y
              = mapCanevasTopLeft.y + (LONG)(
                ((*(*pTempRoomList).roomToSee).coordinates[j].y)
                *lOverviewScale);
            }

            //draw room - see p70
            Polygon(hdc,tempRoom,
             (*(*pTempRoomList).roomToSee).numberOfCoordinates);
            hOldBr = (HBRUSH) SelectObject(hdc,hBr);
            SelectObject(hdc,hOldBr);

            pTempRoomList=(*pTempRoomList).nextRoomInList;
    }

    // 4) Draw Connexions over the Overview Map
    // NOTE : we represent the connexions by
    // horizontal/vertical rectangles

    pTempRoomList=pFirstRoomInList;
    while (pTempRoomList!=NULL)
    {
          //resize connexion
          for (j=0;
          j<(*(*pTempRoomList).roomToSee).numberOfAdjacentRooms;
          j++)
          {

              tempCnx[0].x
                   =mapCanevasTopLeft.x+(LONG)(
                 ((*(*pTempRoomList).roomToSee).connexionsA[j].x)
                   *lOverviewScale);
              tempCnx[0].y
                   =mapCanevasTopLeft.y+(LONG)(
                 ((*(*pTempRoomList).roomToSee).connexionsA[j].y)
                   *lOverviewScale);

              tempCnx[1].x
                   =mapCanevasTopLeft.x+(LONG)(
                 ((*(*pTempRoomList).roomToSee).connexionsB[j].x)
                   *lOverviewScale);
              tempCnx[1].y
                   =mapCanevasTopLeft.y+(LONG)(
                 ((*(*pTempRoomList).roomToSee).connexionsB[j].y)
                   *lOverviewScale);

                   //draw connexion
                   hBr = (HBRUSH) GetStockObject(WHITE_BRUSH);
                   hOldBr = (HBRUSH) SelectObject(hdc,hBr);

                   if (tempCnx[0].x<=tempCnx[1].x
                          && tempCnx[0].y<=tempCnx[1].y)
                   {
                          Rectangle(hdc,
                             tempCnx[0].x-DEFAULT_CNX_DRW_OVERVW,
                             tempCnx[0].y-DEFAULT_CNX_DRW_OVERVW,
```

```
                                        tempCnx[1].x+DEFAULT_CNX_DRW_OVERVW,
                                        tempCnx[1].y+DEFAULT_CNX_DRW_OVERVW
                                        );
                                SelectObject(hdc,hOldBr);
                        }
                        else
                        {
                                if (tempCnx[0].x<=tempCnx[1].x
                                        && tempCnx[0].y>=tempCnx[1].y)
                                {
                                  Rectangle(hdc,
                                   tempCnx[0].x-DEFAULT_CNX_DRW_OVERVW,
                                   tempCnx[1].y-DEFAULT_CNX_DRW_OVERVW,
                                   tempCnx[1].x+DEFAULT_CNX_DRW_OVERVW,
                                   tempCnx[0].y+DEFAULT_CNX_DRW_OVERVW
                                   );
                                        SelectObject(hdc,hOldBr);
                                }
                                else
                                {
                                        if (tempCnx[0].x>=tempCnx[1].x
                                          && tempCnx[0].y<=tempCnx[1].y)
                                        {
                                                Rectangle(hdc,
                                   tempCnx[1].x-DEFAULT_CNX_DRW_OVERVW,
                                   tempCnx[0].y-DEFAULT_CNX_DRW_OVERVW,
                                   tempCnx[0].x+DEFAULT_CNX_DRW_OVERVW,
                                   tempCnx[1].y+DEFAULT_CNX_DRW_OVERVW
                                                );
                                                SelectObject(hdc,hOldBr);
                                        }
                                        else
                                        {
                                                if
                                              (tempCnx[0].x>=tempCnx[1].x
                                          && tempCnx[0].y>=tempCnx[1].y)
                                                {
                                                 Rectangle(hdc,
                                   tempCnx[1].x-DEFAULT_CNX_DRW_OVERVW,
                                   tempCnx[1].y-DEFAULT_CNX_DRW_OVERVW,
                                   tempCnx[0].x+DEFAULT_CNX_DRW_OVERVW,
                                   tempCnx[0].y+DEFAULT_CNX_DRW_OVERVW
                                                );

SelectObject(hdc,hOldBr);
                                                }
                                        }
                                }
                        }
                }

                pTempRoomList=(*pTempRoomList).nextRoomInList;
        }

        EndPaint(hWnd,&ps);
        break;

case 2: // room map

        if (iSoundOn==1)
        {
                if (iPreviousTaskLevel==1)
                        PlayWaveResource
                                (IDR_AUDIT_WAVE_OVERVIEW2ROOM,hInst,0);
                else
```

```
                {
                        // play the switch room sound if we are
                        // switching from a room to another
                        if ((iDrawNew==1) && (iPreviousTaskLevel==2))
                        {
                                PlayWaveResource
                                  (IDR_AUDIT_WAVE_ROOM2ROOM,hInst,0);
                        }
                        else
                        {
                                if (iPreviousTaskLevel==3)
                                {
                                   PlayWaveResource
                                    (IDR_AUDIT_WAVE_ARTOBJECT,hInst,0);
                                }
                        }
                }
        }

        if (iSoundOn==2)
        {
                if (iPreviousTaskLevel==1)
                        PlayWaveResource
                           (IDR_EARCON_WAVE_OVERVIEW2ROOM,hInst,0);
                else
                {
                        // play the switch room sound if we are
                        // switching from a room to another
                        if ((iDrawNew==1) && (iPreviousTaskLevel==2))
                        {
                                PlayWaveResource
                                   (IDR_EARCON_WAVE_ROOM2ROOM,hInst,0);
                        }
                        else
                        {
                                if (iPreviousTaskLevel==3)
                                {
                                  PlayWaveResource
                                   (IDR_EARCON_WAVE_ARTOBJECT,hInst,0);
                                }
                        }
                }
        }

        iDrawNew=1;
        iPreviousTaskLevel=2;

        //1) Define Drawing Canevas

        //draw back ground canevas
        hBr = (HBRUSH) GetStockObject(WHITE_BRUSH);
        hOldBr = (HBRUSH) SelectObject(hdc,hBr);
        Rectangle(hdc,clientRect.left+5,clientRect.top+5,
                clientRect.right-5,clientRect.bottom-5);
        SelectObject(hdc,hOldBr);

        // 2)Define Scale of the Map and Write it on the screen

        //Get extremal coordinates of the room map
        mapMinPoint=(*pActualRoom).getMinCoordinates();
        mapMaxPoint=(*pActualRoom).getMaxCoordinates();

        //Calculate room scale coeff
        lLengthCoeff
          =(float)(mapCanevasBottomRight.x-mapCanevasTopLeft.x)
```

```
    /(float)(mapMaxPoint.x-mapMinPoint.x);
lHeigthCoeff
  =(float)(mapCanevasBottomRight.y-mapCanevasTopLeft.y)
    /(float)(mapMaxPoint.y-mapMinPoint.y);

//Calculate room scale coeff by choosing the smallest
//value between Length and Height Coeff
if (lLengthCoeff<lHeigthCoeff)
      lRoomScale=lLengthCoeff;
else
      lRoomScale=lHeigthCoeff;

// Center overview map on the screen
mapCanevasTopLeft.x
  = mapCanevasTopLeft.x
    + ((mapCanevasBottomRight.x-mapCanevasTopLeft.x)
    -(LONG)((mapMaxPoint.x-mapMinPoint.x)*lRoomScale))/2;

mapCanevasTopLeft.y
  = mapCanevasTopLeft.y
    + ((mapCanevasBottomRight.y-mapCanevasTopLeft.y)
    -(LONG)((mapMaxPoint.y-mapMinPoint.y)*lRoomScale))/2;

//Indicate Room Name and Scale Level
_stprintf(string,L"%S",(*pActualRoom).name);

SetBkMode(hdc,OPAQUE); //see p.45
DrawText(hdc,string,-1,&clientRect,
      DT_CENTER|DT_SINGLELINE);

// 3) Draw the Room Map

//resize room to fit in clientRect and align it top left
for (j=0;j<(*pActualRoom).numberOfCoordinates;j++)
{
      tempRoom[j].x
        = mapCanevasTopLeft.x + (LONG)(
          ((*pActualRoom).coordinates[j].x-mapMinPoint.x)
          *lRoomScale);
      tempRoom[j].y
        = mapCanevasTopLeft.y+(LONG)(
          ((*pActualRoom).coordinates[j].y-mapMinPoint.y)
          *lRoomScale);
}

//draw room in clientRect
hBr = (HBRUSH) GetStockObject(WHITE_BRUSH);
hOldBr = (HBRUSH) SelectObject(hdc,hBr);
Polygon(hdc,tempRoom,(*pActualRoom).numberOfCoordinates);
SelectObject(hdc,hOldBr);

// 4) Draw Connexions over the Room Map
// NOTE : we represent the connexions by
// horizontal/vertical rectangles

for (j=0;j<(*pActualRoom).numberOfAdjacentRooms;j++)
{
      //resize connexion

      tempCnx[0].x
        = mapCanevasTopLeft.x+(LONG)(
          ((*pActualRoom).connexionsA[j].x-mapMinPoint.x)
          *lRoomScale);
      tempCnx[0].y
        = mapCanevasTopLeft.y+(LONG)(
```

```
                        ((*pActualRoom).connexionsA[j].y-mapMinPoint.y)
                        *lRoomScale);

                    tempCnx[1].x
                      = mapCanevasTopLeft.x+(LONG)(
                        ((*pActualRoom).connexionsB[j].x-mapMinPoint.x)
                        *lRoomScale);
                    tempCnx[1].y
                      = mapCanevasTopLeft.y+(LONG)(
                        ((*pActualRoom).connexionsB[j].y-mapMinPoint.y)
                        *lRoomScale);


                    //define color: see p71-73
                    if (strcmp((*pActualRoom).adjacentRoomsCode[j],
                            "Exit")==0)
                    {
                            //if exit connexion -> green color
                            hBr
                              =(HBRUSH)CreateSolidBrush
                                   (RGB(0,175,0));
                            hOldBr = (HBRUSH) SelectObject(hdc,hBr);
                    }
                    else
                    {
                            hBr = (HBRUSH) GetStockObject(WHITE_BRUSH);

                            //see if the connexion leads to a "ToSee"
                            // Room
                            pTempRoomList=pFirstRoomInList;
                            while (pTempRoomList!=NULL)
                            {
if (strcmp((*pActualRoom).adjacentRoomsCode[j],
    (*(*pTempRoomList).roomToSee).code)==0)
{
        if (pIrRoom!=NULL)
        {
                //test if this connected room is signaled by the IR signal
                if (strcmp((*(*pTempRoomList).roomToSee).name,
                        (*pIrRoom).name)==0)
                {
                        if ((*(*pTempRoomList).roomToSee).isToSee())
                        {
                                // if the cnx leads to the room signaled by
                                // the IR and is "to see", we highlight it in
                                //red/yellow pattern :
                                // * yellow for "to see"
                                // * red for IR signaled
                                hBr = (HBRUSH) CreateCustomHatchBrush
                                        (HS_FDIAGONAL,PALETTERGB(255,230,0),
                                        PALETTERGB(255,0,0));
                                break;
                        }
                        else
                        {
                                // the cnx leads to the room currently
                                // signaled by the IR, but rhis room is not
                                //"to see"-> we just highlight the cnx in red
                                hBr = (HBRUSH) CreateSolidBrush
                                                (RGB(255,0,0));
                                break;
                        }
                }
                else //this connected room is NOT signaled by the IR
                {
```

```
                    if ((*(*pTempRoomList).roomToSee).isToSee())
                    {
                            // choose yellow color
                            hBr = (HBRUSH) CreateSolidBrush
                                            (RGB(255,230,0));
                            break;
                    }
                    else
                    {
                            //if normal connexion -> blue color
                            hBr = (HBRUSH) CreateSolidBrush
                                            (RGB(185,240,245));
                            break;
                    }
            }
    }
    else //pIrRoom==NULL
    {
            if ((*(*pTempRoomList).roomToSee).isToSee())
            {
                    // choose yellow color
                    hBr = (HBRUSH) CreateSolidBrush(RGB(255,230,0));
                    break;
            }
            else
            {
                    //if normal connexion -> blue color
                    hBr = (HBRUSH) CreateSolidBrush(RGB(185,240,245));
                    break;
            }
    }
}
else
{
        pTempRoomList=(*pTempRoomList).nextRoomInList;
}
```

```
                                        }
                                }

                                hOldBr = (HBRUSH) SelectObject(hdc,hBr);

                                //draw connexion
                                if (tempCnx[0].x<=tempCnx[1].x
                                        && tempCnx[0].y<=tempCnx[1].y)
                                {
                                        Rectangle(hdc,
                                                tempCnx[0].x-DEFAULT_CNX_DRW_ROOM,
                                                tempCnx[0].y-DEFAULT_CNX_DRW_ROOM,
                                                tempCnx[1].x+DEFAULT_CNX_DRW_ROOM,
                                                tempCnx[1].y+DEFAULT_CNX_DRW_ROOM);
                                        SelectObject(hdc,hOldBr);
                                }
                                else
                                {
                                        if (tempCnx[0].x<=tempCnx[1].x
                                                && tempCnx[0].y>=tempCnx[1].y)
                                        {
                                                Rectangle(hdc,
                                                   tempCnx[0].x-DEFAULT_CNX_DRW_ROOM,
                                                   tempCnx[1].y-DEFAULT_CNX_DRW_ROOM,
                                                   tempCnx[1].x+DEFAULT_CNX_DRW_ROOM,
                                                   tempCnx[0].y+DEFAULT_CNX_DRW_ROOM);
                                                SelectObject(hdc,hOldBr);
                                        }
                                        else
```

```
                                {
if (tempCnx[0].x>=tempCnx[1].x && tempCnx[0].y<=tempCnx[1].y)
{
        Rectangle(hdc,tempCnx[1].x-DEFAULT_CNX_DRW_ROOM,
                        tempCnx[0].y-DEFAULT_CNX_DRW_ROOM,
                        tempCnx[0].x+DEFAULT_CNX_DRW_ROOM,
                        tempCnx[1].y+DEFAULT_CNX_DRW_ROOM);

        SelectObject(hdc,hOldBr);
}
else
{
        if (tempCnx[0].x>=tempCnx[1].x && tempCnx[0].y>=tempCnx[1].y)
        {
                Rectangle(hdc,tempCnx[1].x-DEFAULT_CNX_DRW_ROOM,
                                tempCnx[1].y-DEFAULT_CNX_DRW_ROOM,
                                tempCnx[0].x+DEFAULT_CNX_DRW_ROOM,
                                tempCnx[0].y+DEFAULT_CNX_DRW_ROOM);

        SelectObject(hdc,hOldBr);
        }
}
                                }
                        }
                }

                // 5) Draw the ArtObjects over the Room Map
                // TEMP : we assume that art objects are
                // horizontal/vertical rectangles

                pTempArtObject=(*pActualRoom).firstObjectInRoom;
                while (pTempArtObject!=NULL)
                {
                        if ((*pTempArtObject).isToSee)
                        {
                                // we choose yellow color for "ToSee" Objects
                                // the color is the same as for the "ToSee"
                                // Rooms
                                hBr
                                 =(HBRUSH) CreateSolidBrush(RGB(255,230,0));
                                hOldBr = (HBRUSH) SelectObject(hdc,hBr);
                        }
                        else
                        {
                                // we choose light gray color for "ToSee"
                                //Objects
                                hBr
                                 =(HBRUSH)CreateSolidBrush(RGB(225,225,225));
                                hOldBr = (HBRUSH) SelectObject(hdc,hBr);
                        }

                        Rectangle(hdc,
                                mapCanevasTopLeft.x+(LONG)(
                                ((*pTempArtObject).positionA.x-mapMinPoint.x)
                                *lRoomScale),
                                mapCanevasTopLeft.y+(LONG)(
                                ((*pTempArtObject).positionA.y-mapMinPoint.y)
                                *lRoomScale),
                                mapCanevasTopLeft.x+(LONG)(
                                ((*pTempArtObject).positionB.x-mapMinPoint.x)
                                *lRoomScale),
                                mapCanevasTopLeft.y+(LONG)(
                                ((*pTempArtObject).positionB.y-mapMinPoint.y)
                                *lRoomScale));
                        SelectObject(hdc,hOldBr);
```

```
                pTempArtObject=(*pTempArtObject).nextObjectInRoom;
        }

        EndPaint(hWnd,&ps);
        break;

case 3: //object info

        if ((iSoundOn==1) && (iDrawNew==1))
                PlayWaveResource (IDR_AUDIT_WAVE_ARTOBJECT,hInst,0);

        if ((iSoundOn==2) && (iDrawNew==1))
                PlayWaveResource(IDR_EARCON_WAVE_ARTOBJECT,hInst,0);

        if (iDrawNew==0)
        {
                iDrawNew=1;
        }

        iPreviousTaskLevel=3;

        // Indicate Object Info header

        _stprintf(string,L"%S ??(%S??)",
                        (*pActualArtObject).name,
                        (*pActualRoom).name);
        DrawText(hdc,string,-1,&clientRect,
                DT_TOP|DT_CENTER|DT_SINGLELINE);

        clientRect.left  = clientRect.left+5;
        clientRect.right = clientRect.right-5;
        clientRect.top   = clientRect.top+20;

        // Draw bitmap picture corresponding to the object

        _stprintf(TszCode,L"%S",(*pActualArtObject).code);
        _stprintf(TszPath,L"");
        _tcscat(TszPath,TszTopFolder);
        _tcscat(TszPath,L"images\\");
        _tcscat(TszPath,TszCode);
        _tcscat(TszPath,L".bmp");

        i
        = DrawBitmapFromFileWithTCHAR (TszPath,hInst,hdc,
                clientRect)+5;

        clientRect.left  = clientRect.left-5;
        clientRect.right = clientRect.right+5;
        clientRect.top = clientRect.top + i;

        // Play wave file corresponding to the object
        _stprintf(TszPath,L"");
        _tcscat(TszPath,TszTopFolder);
        _tcscat(TszPath,L"sounds\\");
        _tcscat(TszPath,TszCode);
        _tcscat(TszPath,L".wav");

        //if there is a sound pattern activated
        //(either Auditory Icon or Earcon),
        //we must play the sound file available on the disk
        if (iSoundOn>0)
        {
                PlayWaveFromFileWithTCHAR(TszPath);
        }
```

```
                        _stprintf(string,L" * %S : %S\n * %S : %S\n * %S : %S\n
                                    * %S : %S\n * %S : %S",
                                    ARTOBJECT_ATTRIBUTE1,
                                    (*pActualArtObject).author,
                                    ARTOBJECT_ATTRIBUTE2,
                                    (*pActualArtObject).style,
                                    ARTOBJECT_ATTRIBUTE3,
                                    (*pActualArtObject).period,
                                    ARTOBJECT_ATTRIBUTE4,
                                    (*pActualArtObject).country,
                                    ARTOBJECT_ATTRIBUTE5,
                                    (*pActualArtObject).material
                                    );

                        DrawText(hdc, string,-1,&clientRect,
                              DT_LEFT | DT_WORDBREAK);

                        clientRect.top=clientRect.top - i -20;

                        EndPaint(hWnd,&ps);
                        break;
        }
        return 0;
}

// ----------------------------------------------------------------
//
// DoHibernateMain : process WM_HIBERNATE message for window
//
// ----------------------------------------------------------------
// This procedure reduces the program's use of the memory to the
// absolute minimum, for example if another program is being used.
// ----------------------------------------------------------------

LRESULT DoHibernateMain (HWND hWnd,UINT wMsg,WPARAM wParam,LPARAM lParam)
{
        //if not the active window, destroy the command bar to save memory
        if (GetActiveWindow() != hWnd)
              CommandBar_Destroy (GetDlgItem(hWnd,IDC_CMDBAR));

        return 0;
}

// ----------------------------------------------------------------
//
// DoActiveMain : processes the WM_ACTIVATE message for window
//
// ----------------------------------------------------------------
// This procedure is used to reactivate all the data structures and
// window controls that were freed by the WM_HIBERNATE message.
// ----------------------------------------------------------------

LRESULT DoActivateMain (HWND hWnd,UINT wMsg,WPARAM wParam,LPARAM lParam)
{
        HWND hwndCB;

        //if activating and no command bar, create it.
        if ((LOWORD (wParam) != WA_INACTIVE)
              && (GetDlgItem(hWnd,IDC_CMDBAR) == 0))
        {
              //Create a command bar.
              hwndCB = CommandBar_Create(hInst,hWnd,IDC_CMDBAR);

              //Add exit button to command bar.
```

```
•       CommandBar_AddAdornments(hwndCB,0,0);
      }

      return 0;
}

// -----------------------------------------------------------------
//
// DoDestroyMain : processes the WM_DESTROY message for window
//
// -----------------------------------------------------------------
// This procedure is used to send the WM_QUIT message, which means
// that the program is being quit.
// -----------------------------------------------------------------

LRESULT DoDestroyMain (HWND hWnd,UINT wMsg,WPARAM wParam,LPARAM lParam)
{
      PostQuitMessage (0);
      return 0;
}

// -----------------------------------------------------------------
//
// DoManageStylusInput : processes the WM_LBUTTONUP message for window
//
// -----------------------------------------------------------------
// This procedure is used to manage the stylus input in the program.
// -----------------------------------------------------------------

LRESULT DoManageStylusInput (HWND hWnd,UINT wMsg,WPARAM wParam,LPARAM lParam)
{
      POINT pt;
      float ptx,pty;
      RoomList *pTempRoomList;
      char adjRoomCode[MAX_STRINGSIZE];
      ArtObject *pTempArtObject;
      RECT tempRect;
      POINT mapMinPoint;

      pt.x = LOWORD(lParam);
      pt.y = HIWORD(lParam);

      //See if pen is on board, and if so, determine which cell
      if (PtInRect (&clientRect,pt))
      {
            switch (iTaskLevel)
            {
                  case 0:
                        break;

                  case 1 : //overview map is actually displayed

                        pTempRoomList=pFirstRoomInList;
                        ptx
                          = ((float)(pt.x-mapCanevasTopLeft.x)
                            /(float)lOverviewScale);
                        pty
                          =((float)(pt.y-mapCanevasTopLeft.y)
                            /(float)lOverviewScale);

                        //find room being pointed by the stylus to set
                        // pActualRoom
                        while (pTempRoomList!=NULL)
                        {
                              if
```

```
                    ((*(*pTempRoomList).roomToSee).isInRoom(ptx,pty))
                    {
                            pActualRoom
                             = (*pTempRoomList).roomToSee;

                            //indicate to switch room map
                            iTaskLevel=2;

                            //force to repaint the clientRect
                            InvalidateRect(hWnd,NULL,TRUE);
                            return 0;
                    }
                    else
                            pTempRoomList
                             = (*pTempRoomList).nextRoomInList;
            }

            break;

    case 2 : //the room map corresponding to pActualRoom
               //is actually displayed

            mapMinPoint=(*pActualRoom).getMinCoordinates();

            ptx
             = ((float)(pt.x-mapCanevasTopLeft.x)
               /(float)lRoomScale+mapMinPoint.x);
            pty
             = ((float)(pt.y-mapCanevasTopLeft.y)
               /(float)lRoomScale+mapMinPoint.y);

            strcpy(adjRoomCode,
                  (*pActualRoom).isCnx
                  (pt,lRoomScale,mapCanevasTopLeft,mapMinPoint));

            if (strcmp(adjRoomCode,"Exit")==0)
            {
                    //switch back to overview map
                    iTaskLevel=1;

                    //force to repaint the clientRect
                    InvalidateRect(hWnd,NULL,TRUE);
                    return 0;
            }

            if ((strcmp(adjRoomCode,"")>0)
                    || (strcmp(adjRoomCode,"")<0))

            // Remember that strcmp( string1, string2 ) returns:
            // *  < 0    if string1 less than string2
            // * == 0    if string1 identical to string2
            // *  > 0    if string1 greater than string2

            {
pTempRoomList=pFirstRoomInList;

//find room being pointed by the stylus to set pActualRoom
while (pTempRoomList!=NULL)
{
        if (strcmp(adjRoomCode,(*(*pTempRoomList).roomToSee).code)==0)
        {
                pActualRoom = (*pTempRoomList).roomToSee;

                //we stay at room map level : iTaskLevel=2;
```

```
            //force to repaint the clientRect
            InvalidateRect(hWnd,NULL,TRUE);
            return 0;
      }
else
{
      pTempRoomList=(*pTempRoomList).nextRoomInList;
}
```

```
                  }

                  pTempArtObject=(*pActualRoom).firstObjectInRoom;


                  //(TEMPORARY) METHOD TO SEE IF IT IS AN OBJECT
                  while (pTempArtObject!=NULL)
                  {
                      tempRect.left
                       = mapCanevasTopLeft.x+(LONG)(
                         ((*pTempArtObject).positionA.x-mapMinPoint.x)
                         *lRoomScale);
                      tempRect.top
                       = mapCanevasTopLeft.y+(LONG)(
                         ((*pTempArtObject).positionA.y-mapMinPoint.y)
                         *lRoomScale);
                      tempRect.right
                       = mapCanevasTopLeft.x+(LONG)(
                         ((*pTempArtObject).positionB.x-mapMinPoint.x)
                         *lRoomScale);
                      tempRect.bottom
                       = mapCanevasTopLeft.y+(LONG)(
                         ((*pTempArtObject).positionB.y-mapMinPoint.y)
                         *lRoomScale);

                         if (PtInRect (&tempRect,pt))
                         {
                                pActualArtObject=pTempArtObject;

                                //indicate to object info
                                iTaskLevel=3;

                                //force to repaint the clientRect
                                InvalidateRect(hWnd,NULL,TRUE);
                                return 0;
                                break;
                         }
                         else
                         {
                                pTempArtObject
                                 =(*pTempArtObject).nextObjectInRoom;
                         }
                  }

                  // If the inputPoint is not in the room
                  // indicate to switch back to overview map

                  if (!(*pActualRoom).isInRoom(ptx,pty))
                  {
                         iTaskLevel=1;

                         //force to repaint the clientRect
                         InvalidateRect(hWnd,NULL,TRUE);
                  }
                  break;

            case 3 :
```

```
                        // Switch back to room map
                        iTaskLevel=2;

                        //force to repaint the clientRect
                        InvalidateRect(hWnd,NULL,TRUE);

                        break;
                }
        }

        return 0;
}

// -----------------------------------------------------------------
//
// DoCommandMain : process WM_COMMAND message for window
//
// -----------------------------------------------------------------

LRESULT DoCommandMain (HWND hWnd,UINT wMsg,WPARAM wParam,LPARAM lParam)
{
        WORD idItem,wNotifyCode;
        HWND hwndCtl;
        INT i;

        //Parse the parameters
        idItem = (WORD) LOWORD (wParam);
        wNotifyCode = (WORD) HIWORD (wParam);
        hwndCtl = (HWND) lParam;

        //Call routine to handle control nessage
        for (i=0;i<dim(MainCommandItems);i++)
        {
                if (idItem==MainCommandItems[i].Code)
                        return
(*MainCommandItems[i].Fxn)(hWnd,idItem,hwndCtl,wNotifyCode);
        }

        return 0;
}

// -----------------------------------------------------------------
//
// DoInitMenuPopMain : process WM_INITMENUPOPUP message for window
//
// -----------------------------------------------------------------

LRESULT DoInitMenuPopMain (HWND hWnd,UINT wMsg,WPARAM wParam,LPARAM lParam)
{
        HMENU hMenu;

        hMenu = CommandBar_GetMenu(GetDlgItem(hWnd,IDC_CMDBAR),0);

        switch (iSoundOn)
        {
                case 0 : //sound is off
                        EnableMenuItem(hMenu, IDM_AUDIT_PATTERN_ON,
                                MF_BYCOMMAND | MF_ENABLED);
                        EnableMenuItem(hMenu, IDM_EARCON_PATTERN_ON,
                                MF_BYCOMMAND | MF_ENABLED);
                        EnableMenuItem(hMenu, IDM_SOUND_OFF,
                                MF_BYCOMMAND | MF_GRAYED);
                        break;
                case 1 : // auditory icon pattern is on
                        EnableMenuItem(hMenu, IDM_AUDIT_PATTERN_ON,
```

```
                        MF_BYCOMMAND | MF_GRAYED);
                EnableMenuItem(hMenu, IDM_EARCON_PATTERN_ON,
                        MF_BYCOMMAND | MF_ENABLED);
                EnableMenuItem(hMenu, IDM_SOUND_OFF,
                        MF_BYCOMMAND | MF_ENABLED);
                break;
        case 2 : // earcon pattern is on
                EnableMenuItem(hMenu, IDM_AUDIT_PATTERN_ON,
                        MF_BYCOMMAND | MF_ENABLED);
                EnableMenuItem(hMenu, IDM_EARCON_PATTERN_ON,
                        MF_BYCOMMAND | MF_GRAYED);
                EnableMenuItem(hMenu, IDM_SOUND_OFF,
                        MF_BYCOMMAND | MF_ENABLED);
                break;
    }

    switch (iTaskLevel)
    {
        case 0 : //pre-tour task level
                EnableMenuItem(hMenu, IDM_RESTART_TOUR,
                        MF_BYCOMMAND | MF_GRAYED);
                EnableMenuItem(hMenu, IDM_END_TOUR,
                        MF_BYCOMMAND | MF_GRAYED);
                EnableMenuItem(hMenu, IDM_SIMULATE_IR,
                        MF_BYCOMMAND | MF_GRAYED);
                break;
        case 1 : //overview map task level
                EnableMenuItem(hMenu, IDM_RESTART_TOUR,
                        MF_BYCOMMAND | MF_ENABLED);
                EnableMenuItem(hMenu, IDM_END_TOUR,
                        MF_BYCOMMAND | MF_ENABLED);
                EnableMenuItem(hMenu, IDM_SIMULATE_IR,
                        MF_BYCOMMAND | MF_ENABLED);
                break;
        case 2 : //room map task level
                EnableMenuItem(hMenu, IDM_RESTART_TOUR,
                        MF_BYCOMMAND | MF_ENABLED);
                EnableMenuItem(hMenu, IDM_END_TOUR,
                        MF_BYCOMMAND | MF_ENABLED);
                EnableMenuItem(hMenu, IDM_SIMULATE_IR,
                        MF_BYCOMMAND | MF_ENABLED);
                break;
        case 3 : //art object task level
                EnableMenuItem(hMenu, IDM_RESTART_TOUR,
                        MF_BYCOMMAND | MF_ENABLED);
                EnableMenuItem(hMenu, IDM_END_TOUR,
                        MF_BYCOMMAND | MF_ENABLED);
                EnableMenuItem(hMenu, IDM_SIMULATE_IR,
                        MF_BYCOMMAND | MF_ENABLED);
                break;
    }

    return 0;
}

// ===============================================================
//
// MENU COMMAND HANDLER ROUTINES
//
// ===============================================================

// ---------------------------------------------------------------
//
// DoMainCommandNewTour : process "new tour" command
//
```

```
// ----------------------------------------------------------------

LPARAM DoMainCommandNewTour(HWND hWnd,WORD idItem,HWND hwndCtl,WORD wNotify)
{
	RoomList *pTempFirstRoomInList;
	INT rc=IDYES;
	TCHAR   szFile[MAX_PATH] = TEXT("\0");
	OPENFILENAME      ofn;

	if (iSoundOn==1) PlayWaveResource (IDR_AUDIT_WAVE_CONFIRM,hInst,0);
	if (iSoundOn==2) PlayWaveResource (IDR_EARCON_WAVE_CONFIRM,hInst,0);

	memset( &(ofn), 0, sizeof(ofn));
	ofn.lStructSize   = sizeof(ofn);
	ofn.hwndOwner = hWnd;
	ofn.lpstrFile = szFile;
	ofn.nMaxFile = MAX_PATH;

	int iFullPathLength;
	int iFileNameLength;
	BOOL initOK,done;

	if (iTaskLevel!=0 && iReOpen==0)
		// user wants to start a new tour, but he is currently in a tour
		// and is not retrying to open a file
	{
		rc = MessageBox(hWnd,
			TEXT("By starting a new tour, you will lose all current
				informations! Do you wish to continue?"),
			TEXT("Start New Tour"),
			MB_YESNO | MB_ICONQUESTION);
		if (rc==IDYES)
		{
			if (iSoundOn==1)
			PlayWaveResource (IDR_AUDIT_WAVE_CONFIRM,hInst,0);

			if (iSoundOn==2)
			PlayWaveResource (IDR_EARCON_WAVE_CONFIRM,hInst,0);
		}
		else
		{
			if (iSoundOn==1)
			PlayWaveResource (IDR_AUDIT_WAVE_CANCEL,hInst,0);

			if (iSoundOn==2)
			PlayWaveResource (IDR_EARCON_WAVE_CANCEL,hInst,0);
		}
	}

	//reset iReOpen
	iReOpen=0;

	if (rc==IDYES)
	{

	////////////////////////////////////////////////////////////////////////
	// You can have multiple subdirectories for your applications; indeed,
	// many directory structures require subdirectories, such as links in
	// the Communications directory at \Windows\Start Menu\Programs\Communications.
	// However, a user does not have access to these folders through applications.
	// Further, Windows CE Services exposes the file system. Using Windows CE
	// Services, a user can create a multilevel directory on a Pocket PC through
	// a desktop computer. However, a user can only see that directory using a
	// Pocket PC if the directory begins in the My Documents folder. Further,
	// a Pocket PC user cannot access more than one level of subdirectories;
	// the shell ignores any subdirectories beyond the first.
```

```
//Open File Dialog Box
ofn.lpstrFilter = TEXT("All (*.*)\0*.*\0");
ofn.lpstrTitle = TEXT("Open File");
ofn.Flags = OFN_EXPLORER;
if (GetOpenFileName(&ofn))
{
      //MessageBox(NULL, ofn.lpstrFile, TEXT("Info"), MB_OK);

      // lpstrFile : long pointer to a buffer that contains the
      //full path of the file

      iFullPathLength=_tcslen(ofn.lpstrFile);
      iFileNameLength=0;
      done=FALSE;

      // Set path variables : directory and file name
      while ((iFileNameLength<iFullPathLength)&&(done==FALSE))
      {
            if (_tcsncicmp
                 (ofn.lpstrFile+iFullPathLength-iFileNameLength,
                 L"\\",1)==0)
            {
                  // set directory
                  _tcscpy(TszTopFolder,L"");
                  _tcsncat(TszTopFolder,
                        ofn.lpstrFile,
                        iFullPathLength-iFileNameLength+1);

                  // set file name
                  _tcscpy(TszFileName,L"");
                  _tcsncat(TszFileName,
                ofn.lpstrFile+iFullPathLength-iFileNameLength+1,
                        iFileNameLength);

                  done=TRUE;
            }
            else
            {
                  iFileNameLength++;
            }
      }

      // initEnv
      TCHAR TszCompleteFileName[MAX_PATH];
      _stprintf(TszCompleteFileName,
            L"%s%s",TszTopFolder,TszFileName);

      pTempFirstRoomInList=new RoomList();

      initOK=InitFromFileWithTCHAR(TszCompleteFileName,
                  pTempFirstRoomInList);

      if (initOK==TRUE)
      {
            iTaskLevel=0;

            // rebuild room list
            delete pFirstRoomInList;
            pFirstRoomInList=new RoomList();
            (* pFirstRoomInList).copy(pTempFirstRoomInList);

            delete pTempFirstRoomInList;

            // rebuild attribute list
```

```
                        delete pAttributeList;
                        pAttributeList
                              = new AttributeList(pFirstRoomInList);
                        pActualAttribute = NULL;
                        pIrRoom = NULL;

                        //Call the SelectTourMode WRAPPING procedure
                        SelectTourModeWrapProc (hWnd);
                  }

                  else //initialization somehow failed
                  {
                        /*
                        // It could be possible to propose a default tour !
                        rc = MessageBox(hWnd,
                              TEXT("This file is not well formated! Do you
                              wish to continue with the default tour?"),
                              TEXT("Start New Tour"),
                              MB_YESNO | MB_ICONQUESTION);

                        if (rc==IDYES)
                        {
                            initOK
                             = InitDefaultEnvironment(pTempFirstRoomInList);
                        }
                        */
                        if (iTaskLevel!=0)
                        {
                              rc = MessageBox(hWnd,
                                    TEXT("This file is not well formated !
                                    Do you wish to try another file?"),
                                    TEXT("Start New Tour"),
                                    MB_YESNOCANCEL | MB_ICONERROR);
                        }
                        else
                        {
                              rc = MessageBox(hWnd,
                                    TEXT("This file is not well formated !
                                    Do you wish to try another file?"),
                                    TEXT("Start New Tour"),
                                    MB_OKCANCEL | MB_ICONERROR);
                        }

                        if (rc==IDNO) // return to initial menu
                        {
                              iTaskLevel=0;
                        }
                        if (rc==IDYES || rc==IDOK) // re-open file
                        {
                              iReOpen=1;
                              DoMainCommandNewTour
                                    (hWnd,idItem,hwndCtl,wNotify);
                        }
                  }
            }

            // Invalidate clientRect, otherwise the clientRect's dimension
            // will be the equal to the last window displayed, that is the
            // last Message box !
            InvalidateRect(hWnd,NULL,TRUE);
      }

      //Do not play sound when redrawing the screen
      iDrawNew=0;
```

```
	return 0;
}

// --------------------------------------------------------------
//
// DoMainCommandRestartTour : process the "restart tour" command
//
// --------------------------------------------------------------

LPARAM   DoMainCommandRestartTour(HWND   hWnd,WORD   idItem,HWND   hwndCtl,WORD
wNotify)
{
	INT rc=IDYES;
	if (iTaskLevel!=0 && iTourMode!=0)
	// the tour has already begun and this is not a free tour mode !
	{
		rc = MessageBox(hWnd,
			TEXT("By restarting the tour, you will lose all current
			informations! Do you wish to continue?"),
			TEXT("Start New Tour"),
			MB_YESNO | MB_ICONQUESTION);
		if (rc==IDYES)
		{
			if (iSoundOn==1)
				PlayWaveResource (IDR_AUDIT_WAVE_CONFIRM,hInst,0);
			if (iSoundOn==2)
				PlayWaveResource (IDR_EARCON_WAVE_CONFIRM,hInst,0);

			//reset all the objects with ToSee = 0
			RoomList * pTempRoomList;
			pTempRoomList = pFirstRoomInList;
			ArtObject * pTempArtObject;
			while (pTempRoomList!=NULL)
			{
				pTempArtObject
				 =(*(*pTempRoomList).roomToSee).firstObjectInRoom;
				while(pTempArtObject!=NULL)
				{
					(*pTempArtObject).isToSee=0;
					pTempArtObject
					 =(*pTempArtObject).nextObjectInRoom;
				}
				pTempRoomList=(*pTempRoomList).nextRoomInList;
			}
		}
		else
		{
			if (iSoundOn==1)
				PlayWaveResource (IDR_AUDIT_WAVE_CANCEL,hInst,0);
			if (iSoundOn==2)
				PlayWaveResource (IDR_EARCON_WAVE_CANCEL,hInst,0);
		}
	}

	//we must reset global pointers in SelectTourModeWrapProc
	pActualAttribute = NULL;
	pIrRoom = NULL;

	if (rc==IDYES)
	{
		//Call the SelectTourMode WRAPPING procedure
		SelectTourModeWrapProc (hWnd);

		// Invalidate clientRect, otherwise the clientRect's dimension
		//will be the equal to the last window displayed, that is the
```

```
        •    //last Message box !
             InvalidateRect(hWnd,NULL,TRUE);
        }

        //Do not play sound when redrawing the screen
        iDrawNew=0;

        return 0;
}

// ---------------------------------------------------------------
//
// DoMainCommandRestartTour : process the "restart tour" command
//
// ---------------------------------------------------------------

LPARAM DoMainCommandEndTour(HWND hWnd,WORD idItem,HWND hwndCtl,WORD wNotify)
{
        INT rc=IDYES;
        if (iTaskLevel!=0)
        // the tour has already begun and this is not a free tour mode !
        {
                rc = MessageBox(hWnd,
                        TEXT("Are you sure you want to exit the current
                        tour?"),
                        TEXT("Start New Tour"),
                        MB_YESNO | MB_ICONQUESTION);
                if (rc==IDYES)
                {
                        iTourMode=0;
                        iTaskLevel=0;

                        pIrRoom = NULL;
                }
                else
                {
                        if (iSoundOn==1)
                                PlayWaveResource (IDR_AUDIT_WAVE_CANCEL,hInst,0);
                        if (iSoundOn==2)
                                PlayWaveResource (IDR_EARCON_WAVE_CANCEL,hInst,0);
                }
        }

        // Invalidate clientRect, otherwise the clientRect's dimension will be
        // the equal to the last window displayed, in our case the last Message
        //box !
        InvalidateRect(hWnd,NULL,TRUE);

        //Do not play sound when redrawing the screen
        iDrawNew=0;

        return 0;
}

// ---------------------------------------------------------------
//
// DoMainCommandExit : process the "exit" command
//
// ---------------------------------------------------------------

LPARAM DoMainCommandExit(HWND hWnd,WORD idItem,HWND hwndCtl,WORD wNotify)
{
        INT rc;

        if (iSoundOn==1) PlayWaveResource (IDR_AUDIT_WAVE_CONFIRM,hInst,0);
```

```
      if (iSoundOn==2) PlayWaveResource (IDR_EARCON_WAVE_CONFIRM,hInst,0);

      //Ask before leaving
      rc = MessageBox (hWnd,
            TEXT ("Are you sure you want to exit the program?"),
            TEXT ("Exit"),
            MB_YESNO | MB_ICONQUESTION);
      if (rc==IDNO)
      {
            if (iSoundOn==1)
                  PlayWaveResource (IDR_AUDIT_WAVE_CANCEL,hInst,0);
            if (iSoundOn==2)
                  PlayWaveResource (IDR_EARCON_WAVE_CANCEL,hInst,0);

            //Do not play sound when redrawing the screen
            iDrawNew=0;

            return 0;
      }
      //play sound synchronously before exiting the app
      if (iSoundOn==1) PlayWaveResource (IDR_AUDIT_WAVE_CONFIRM,hInst,1);
      if (iSoundOn==2) PlayWaveResource (IDR_EARCON_WAVE_CONFIRM,hInst,1);

      //we must delete global pointers
      delete pFirstRoomInList;
      delete pAttributeList;

      SendMessage (hWnd,WM_CLOSE,0,0);
      return 0;
}

// ------------------------------------------------------------------
//
// DoMainCommandAuditPatternOn : process the "Auditory Icon Pattern On"
// command
//
// ------------------------------------------------------------------

LPARAM DoMainCommandAuditPatternOn(HWND hWnd,WORD idItem,HWND hwndCtl,WORD
wNotify)
{
      iSoundOn=1;
      PlayWaveResource (IDR_AUDIT_WAVE_CONFIRM,hInst,0);
      return 0;
}

// ------------------------------------------------------------------
//
// DoMainCommandEarconPatternOn : process the "Earcon Pattern On" command
//
// ------------------------------------------------------------------

LPARAM DoMainCommandEarconPatternOn(HWND hWnd,WORD idItem,HWND hwndCtl,WORD
wNotify)
{
      iSoundOn=2;
      PlayWaveResource (IDR_EARCON_WAVE_CONFIRM,hInst,0);
      return 0;
}

// ------------------------------------------------------------------
//
// DoMainCommandSoundOff : process the "Sound Enhancement Off" command
//
// ------------------------------------------------------------------
```

```
LPARAM DoMainCommandSoundOff(HWND hWnd,WORD idItem,HWND hwndCtl,WORD wNotify)
{
       iSoundOn=0;
       return 0;
}

// ----------------------------------------------------------------
//
// DoMainSimulateIrSignal : process the "Simulate IR Signal" command
//
// ----------------------------------------------------------------

LPARAM DoMainCommandSimulateIrSignal(HWND hWnd,WORD idItem,HWND hwndCtl,WORD
wNotify)
{
       if (iSoundOn==1) PlayWaveResource (IDR_AUDIT_WAVE_CONFIRM,hInst,0);
       if (iSoundOn==2) PlayWaveResource (IDR_EARCON_WAVE_CONFIRM,hInst,0);

       //Call the dialog box
       DialogBox(hInst, TEXT("SelectRoomIR"),hWnd,SimulateIrSignalDlgProc);

       //Do not play sound when redrawing the screen
       iDrawNew=0;

       return 0;
}

// ----------------------------------------------------------------
//
// DoMainCommandAbout : process the "about" command
//
// ----------------------------------------------------------------

LPARAM DoMainCommandAbout(HWND hWnd,WORD idItem,HWND hwndCtl,WORD wNotify)
{
       if (iSoundOn==1) PlayWaveResource (IDR_AUDIT_WAVE_CONFIRM,hInst,0);
       if (iSoundOn==2) PlayWaveResource (IDR_EARCON_WAVE_CONFIRM,hInst,0);

       //Use DialogBox to create a modal dialog
       DialogBox(hInst, TEXT("aboutbox"),hWnd,AboutDlgProc);

       //Do not play sound when redrawing the screen
       iDrawNew=0;

       return 0;
}

// ================================================================
//
// DIALOG BOXES ROUTINES
//
// ================================================================

// ----------------------------------------------------------------
//
// About box Dialog Procedure
//
// ----------------------------------------------------------------

BOOL CALLBACK AboutDlgProc (HWND hWnd, UINT wMsg, WPARAM wParam, LPARAM
lParam)
{
       switch (wMsg)
       {
```

```
        case WM_COMMAND:

                switch (LOWORD(wParam))
                {
case IDOK:

        if (iSoundOn==1)
                PlayWaveResource (IDR_AUDIT_WAVE_CONFIRM,hInst,0);
        if (iSoundOn==2)
                PlayWaveResource (IDR_EARCON_WAVE_CONFIRM,hInst,0);
        return TRUE;

case IDCANCEL:

        if (iSoundOn==1)
                PlayWaveResource (IDR_AUDIT_WAVE_CANCEL,hInst,0);
        if (iSoundOn==2)
                PlayWaveResource (IDR_EARCON_WAVE_CANCEL,hInst,0);
        EndDialog(hWnd,0);
        return TRUE;
                }
        break;
        }

        return FALSE;
}

// ----------------------------------------------------------------
//
// SimulateIrSignal Dialog Procedure
//
// ----------------------------------------------------------------

BOOL CALLBACK SimulateIrSignalDlgProc(HWND hWnd, UINT wMsg, WPARAM wParam,
LPARAM lParam)
{
        TCHAR szOut[MAX_STRINGSIZE];
        INT lpnItems;
        RoomList * pTempRoomList;

        switch (wMsg)
        {
                case WM_INITDIALOG:

                        // Fill the list
                        pTempRoomList=pFirstRoomInList;
                        while (pTempRoomList!=NULL)
                        {
if (strcmp((*(*pTempRoomList).roomToSee).name,"")!=0)
{
        wsprintf(szOut, TEXT("%S"),(*(*pTempRoomList).roomToSee).name);

        SendDlgItemMessage(hWnd,IDC_ROOMIRYLIST,LB_ADDSTRING,0,
                (LPARAM)szOut);
}

pTempRoomList=(*pTempRoomList).nextRoomInList;
                        }

                        //by default, select first room in list
                        SendDlgItemMessage(hWnd,IDC_ROOMIRYLIST,
                                LB_SETCURSEL,(WPARAM)0,0);
                        return TRUE;
```

```
        case WM_COMMAND:
                switch (LOWORD (wParam))
                {
                        case IDD_SELECTROOMIR_SEND:
```

```
// like we force default selection, we are sure that there is at least
// one item selected.

lpnItems=SendDlgItemMessage(hWnd,IDC_ROOMIRYLIST,LB_GETCURSEL,0,0);

SendDlgItemMessage(hWnd,IDC_ROOMIRYLIST,LB_GETTEXT,
        (WPARAM) lpnItems,(LPARAM)(LPCTSTR) szOut);

// The name of the selected room must be put in a global variable
// that will be accessed by the ListenIrThread
_tcscpy(szIrMsg,szOut);

if (iSoundOn==1) PlayWaveResource (IDR_AUDIT_WAVE_CONFIRM,hInst,0);
if (iSoundOn==2) PlayWaveResource (IDR_EARCON_WAVE_CONFIRM,hInst,0);

EndDialog (hWnd,0);

// Finally, set the event for the ListeningIrThread
SetEvent(hListenIrEvent);

return TRUE;
```

```
                        case IDD_SELECTROOMIR_CANCEL:

                                if (iSoundOn==1)
                                        PlayWaveResource
                                          (IDR_AUDIT_WAVE_CANCEL,hInst,0);
                                if (iSoundOn==2)
                                        PlayWaveResource
                                        (IDR_EARCON_WAVE_CANCEL,hInst,0);
                                EndDialog (hWnd,0);
                                return TRUE;
                }
                break;
        }
        return FALSE;
}

// ----------------------------------------------------------------
//
// SelectTourMode Dialog Procedure
//
// ----------------------------------------------------------------

BOOL CALLBACK SelectTourModeDlgProc (HWND hWnd, UINT wMsg, WPARAM wParam,
LPARAM lParam)
{
        switch (wMsg)
        {
                case WM_INITDIALOG:
                //initializes the dialog box with free tour mode by default
                //see p212 [in case of edit text box]
                        iTourMode=0;
                        //0 is the value for FreeTourMode (default tour mode)
                        CheckRadioButton(hWnd,
                                IDD_RADIO_FREE,//First radio button in the group.
                                IDD_RADIO_CUSTOMIZED,
                                IDD_RADIO_FREE);//Radio button to select.
                        return TRUE;
```

```
        case WM_COMMAND:

                switch (LOWORD (wParam))
                {
                        case IDD_RADIO_FREE:
iTourMode=0;            //0 is the value for FreeTourMode

//see if free tour is already checked
if (SendDlgItemMessage (hWnd, IDD_RADIO_FREE,
        BM_GETSTATE,0,0)==BST_CHECKED)
{}
else
{
        if (iSoundOn==1)
                PlayWaveResource (IDR_AUDIT_WAVE_RADIOBTN,hInst,0);
        if (iSoundOn==2)
                PlayWaveResource (IDR_EARCON_WAVE_RADIOBTN,hInst,0);

        CheckRadioButton(hWnd,
                IDD_RADIO_FREE,
                IDD_RADIO_CUSTOMIZED,
                IDD_RADIO_FREE);
};

return TRUE;
                        case IDD_RADIO_CUSTOMIZED:
iTourMode=1;            //1 is the value for CustomizedTourMode

//see if customized tour mode is already checked
if (SendDlgItemMessage (hWnd, IDD_RADIO_CUSTOMIZED,
        BM_GETSTATE,0,0)==BST_CHECKED)
{}
else
{
        if (iSoundOn==1)
                PlayWaveResource (IDR_AUDIT_WAVE_RADIOBTN,hInst,0);
        if (iSoundOn==2)
                PlayWaveResource (IDR_EARCON_WAVE_RADIOBTN,hInst,0);
        CheckRadioButton(hWnd,
                IDD_RADIO_FREE,
                IDD_RADIO_CUSTOMIZED,
                IDD_RADIO_CUSTOMIZED);
};
return TRUE;
                        case BTN_SELECTTOURMODE_CONTINUE:

                                if (iSoundOn==1)
                                        PlayWaveResource
                                            (IDR_AUDIT_WAVE_CONFIRM,hInst,0);
                                if (iSoundOn==2)
                                        PlayWaveResource
                                            (IDR_EARCON_WAVE_CONFIRM,hInst,0);
                                iAskUserResult=1;
                                //the task level is 0 because we are starting
                                //a new tour.
                                iTaskLevel=0;
                                EndDialog (hWnd,0);
                                return TRUE;
```

```
•               }
                break;
        }
        return FALSE;
}

// ---------------------------------------------------------------
//
// [CustomizedTourMode] SelectTourByCategory Dialog Procedure
//
// ---------------------------------------------------------------

BOOL CALLBACK SelectTourByCategoryDlgProc (HWND hWnd, UINT wMsg, WPARAM
wParam, LPARAM lParam)
{
        switch (wMsg)
        {
                case WM_INITDIALOG: //initializes the dialog box

                        CheckRadioButton(hWnd,
                                IDD_ARTOBJECT_ATTRIBUTE1,
                                IDD_ARTOBJECT_ATTRIBUTE5,
                                IDD_ARTOBJECT_ATTRIBUTE1)
                        return TRUE;


                case WM_COMMAND:

                        switch (LOWORD (wParam))
                        {
                                case IDD_ARTOBJECT_ATTRIBUTE1:
```

```
//see if free tour is already checked
if (SendDlgItemMessage (hWnd,IDD_ARTOBJECT_ATTRIBUTE1,
        BM_GETSTATE,0,0)==BST_CHECKED)
{}
else
{
        if (iSoundOn==1)
                PlayWaveResource (IDR_AUDIT_WAVE_RADIOBTN,hInst,0);
        if (iSoundOn==2)
                PlayWaveResource (IDR_EARCON_WAVE_RADIOBTN,hInst,0);

        CheckRadioButton(hWnd,
                IDD_ARTOBJECT_ATTRIBUTE1,
                IDD_ARTOBJECT_ATTRIBUTE5,
                IDD_ARTOBJECT_ATTRIBUTE1);
};
iAskUserResult=1;
return TRUE;
```

```
                                case IDD_ARTOBJECT_ATTRIBUTE2:
```

```
//see if free tour is already checked
if (SendDlgItemMessage (hWnd, IDD_ARTOBJECT_ATTRIBUTE2,
         BM_GETSTATE,0,0)==BST_CHECKED)
        {}
else
{
        if (iSoundOn==1)
                PlayWaveResource (IDR_AUDIT_WAVE_RADIOBTN,hInst,0);
        if (iSoundOn==2)
                PlayWaveResource (IDR_EARCON_WAVE_RADIOBTN,hInst,0);
```

```
•       CheckRadioButton(hWnd,
                IDD_ARTOBJECT_ATTRIBUTE1,
                IDD_ARTOBJECT_ATTRIBUTE5,
};
iAskUserResult=2;
return TRUE;
```

case IDD_ARTOBJECT_ATTRIBUTE3:

```
//see if free tour is already checked
if (SendDlgItemMessage (hWnd, IDD_ARTOBJECT_ATTRIBUTE3,
        BM_GETSTATE,0,0)==BST_CHECKED)
{}
else
{
        if (iSoundOn==1)
                PlayWaveResource (IDR_AUDIT_WAVE_RADIOBTN,hInst,0);
        if (iSoundOn==2)
                PlayWaveResource (IDR_EARCON_WAVE_RADIOBTN,hInst,0);

        CheckRadioButton(hWnd,
                IDD_ARTOBJECT_ATTRIBUTE1,
                IDD_ARTOBJECT_ATTRIBUTE5,
                IDD_ARTOBJECT_ATTRIBUTE3);
};
iAskUserResult=3;
return TRUE;
```

case IDD_ARTOBJECT_ATTRIBUTE4:

```
//see if free tour is already checked
if (SendDlgItemMessage (hWnd, IDD_ARTOBJECT_ATTRIBUTE4,
        BM_GETSTATE,0,0)==BST_CHECKED)
{}
else
{
        if (iSoundOn==1)
                PlayWaveResource (IDR_AUDIT_WAVE_RADIOBTN,hInst,0);
        if (iSoundOn==2)
                PlayWaveResource (IDR_EARCON_WAVE_RADIOBTN,hInst,0);

        CheckRadioButton(hWnd,
                IDD_ARTOBJECT_ATTRIBUTE1,
                IDD_ARTOBJECT_ATTRIBUTE5,
                IDD_ARTOBJECT_ATTRIBUTE4);
};
iAskUserResult=4;
return TRUE;
```

case IDD_ARTOBJECT_ATTRIBUTE5:

```
//see if free tour is already checked
if (SendDlgItemMessage (hWnd, IDD_ARTOBJECT_ATTRIBUTE5,
        BM_GETSTATE,0,0)==BST_CHECKED)
{}
else
{
        if (iSoundOn==1)
                PlayWaveResource (IDR_AUDIT_WAVE_RADIOBTN,hInst,0);
        if (iSoundOn==2)
                PlayWaveResource (IDR_EARCON_WAVE_RADIOBTN,hInst,0);

        CheckRadioButton(hWnd,
```

```
                IDD_ARTOBJECT_ATTRIBUTE1,
                IDD_ARTOBJECT_ATTRIBUTE5,
                IDD_ARTOBJECT_ATTRIBUTE5);
};
iAskUserResult=5;
return TRUE;
```

```
                        case BTN_SELECTTOURBYCATEGORY_BACK:

                                if (iSoundOn==1)
                                        PlayWaveResource
                                                (IDR_AUDIT_WAVE_CANCEL,hInst,0);
                                if (iSoundOn==2)
                                        PlayWaveResource
                                              (IDR_EARCON_WAVE_CANCEL,hInst,0);
                                iAskUserResult=0-iAskUserResult;
                                EndDialog (hWnd,0);
                                return TRUE;

                        case BTN_SELECTTOURBYCATEGORY_CONTINUE:

                                if (iSoundOn==1)
                                        PlayWaveResource
                                            (IDR_AUDIT_WAVE_CONFIRM,hInst,0);
                                if (iSoundOn==2)
                                        PlayWaveResource
                                            (IDR_EARCON_WAVE_CONFIRM,hInst,0);
                                EndDialog (hWnd,0);
                                return TRUE;
                }
                break;
        }
        return FALSE;
}

// ---------------------------------------------------------------
//
// [CustomizedTourMode] SelectSubCategory Dialog Procedure
//
// ---------------------------------------------------------------

BOOL CALLBACK SelectSubCategoryDlgProc (HWND hWnd, UINT wMsg, WPARAM wParam,
LPARAM lParam)
{
        TCHAR szOut[MAX_STRINGSIZE],szTemp[MAX_STRINGSIZE];
        INT lpnItems;
        Attribute * pTempAttribute;

        switch (wMsg)
        {
                case WM_INITDIALOG:

                        // Fill the list
                        switch (iAskUserResult)
                        {
                                case 1 : //Select Artist Sub Cat

                                        //put the pActualAttribute on pAuthor
                                        pActualAttribute=(*pAttributeList).pAuthor;

                                        pTempAttribute=(*pAttributeList).pAuthor;
                                        while (pTempAttribute!=NULL)
                                        {
                                                if (strcmp((*pTempAttribute).value,
                                                        "")!=0)
```

```
                        {
                                wsprintf(szOut, TEXT("%S"),
                                        (*pTempAttribute).value);
                                SendDlgItemMessage(hWnd,
                                        IDC_SUBCATEGORYLIST,
                                        LB_ADDSTRING,0,
                                        (LPARAM)szOut);
                        }
                        pTempAttribute
                          =(*pTempAttribute).nextAttribute;
                }
                break;

        case 2: //style

                //put the pActualAttribute on pStyle
                pActualAttribute=(*pAttributeList).pStyle;

                pTempAttribute=(*pAttributeList).pStyle;
                while (pTempAttribute!=NULL)
                {
                        if (strcmp((*pTempAttribute).value,
                                "")!=0)
                        {
                                wsprintf(szOut, TEXT("%S"),
                                        (*pTempAttribute).value);
                                SendDlgItemMessage(hWnd,
                                        IDC_SUBCATEGORYLIST,
                                        LB_ADDSTRING,0,
                                        (LPARAM)szOut);
                        }
                        pTempAttribute
                          =(*pTempAttribute).nextAttribute;
                }
                break;

        case 3: //period

                //put the pActualAttribute on pPeriod
                pActualAttribute=(*pAttributeList).pPeriod;

                pTempAttribute=(*pAttributeList).pPeriod;
                while (pTempAttribute!=NULL)
                {
                        if (strcmp((*pTempAttribute).value,
                                "")!=0)
                        {
                                wsprintf(szOut, TEXT("%S"),
                                        (*pTempAttribute).value);
                                SendDlgItemMessage(hWnd,
                                        IDC_SUBCATEGORYLIST,
                                        LB_ADDSTRING,0,
                                        (LPARAM)szOut);
                        }
                        pTempAttribute
                          =(*pTempAttribute).nextAttribute;
                }
                break;

        case 4: //country

                //put the pActualAttribute on pCountry
                pActualAttribute=(*pAttributeList).pCountry;

                pTempAttribute=(*pAttributeList).pCountry;
```

```
                        while (pTempAttribute!=NULL)
                        {
                                if (strcmp((*pTempAttribute).value,
                                        "")!=0)
                                {
                                        wsprintf(szOut, TEXT("%S"),
                                                (*pTempAttribute).value);
                                        SendDlgItemMessage(hWnd,
                                                IDC_SUBCATEGORYLIST,
                                                LB_ADDSTRING,0,
                                                (LPARAM)szOut);
                                }
                                pTempAttribute
                                  =(*pTempAttribute).nextAttribute;
                        }
                        break;

                case 5: //material

                        //put the pActualAttribute on pMaterial
                        pActualAttribute=(*pAttributeList).pMaterial;

                        pTempAttribute=(*pAttributeList).pMaterial;
                        while (pTempAttribute!=NULL)
                        {
                                if (strcmp((*pTempAttribute).value,
                                        "")!=0)
                                {
                                        wsprintf(szOut, TEXT("%S"),
                                                (*pTempAttribute).value);
                                        SendDlgItemMessage(hWnd,
                                                IDC_SUBCATEGORYLIST,
                                                LB_ADDSTRING,0,
                                                (LPARAM)szOut);
                                }
                                pTempAttribute
                                  =(*pTempAttribute).nextAttribute;
                        }
                        break;
        }

SendDlgItemMessage(hWnd,IDC_SUBCATEGORYLIST,LB_SETCURSEL,(WPARAM)0,0);
        return TRUE;

    case WM_COMMAND:
        switch (LOWORD (wParam))
        {
                case BTN_SELECTSUBCATEGORY_BACK:

                        if (iSoundOn==1)
                                PlayWaveResource
                                        (IDR_AUDIT_WAVE_CANCEL,hInst,0);
                        if (iSoundOn==2)
                                PlayWaveResource
                                       (IDR_EARCON_WAVE_CANCEL,hInst,0);
                        iAskUserResult=0-iAskUserResult;
                        EndDialog (hWnd,0);
                        return TRUE;

                case BTN_SELECTSUBCATEGORY_CONTINUE:

                        // like we force default selection, we are
                        // sure that there is at least
                        // one item selected.
                        lpnItems
```

```
                                        =SendDlgItemMessage(hWnd,
                                             IDC_SUBCATEGORYLIST,
                                             LB_GETCURSEL,0,0);

                                   SendDlgItemMessage(hWnd,
                                        IDC_SUBCATEGORYLIST,LB_GETTEXT,
                                        (WPARAM)lpnItems,(LPARAM)(LPCTSTR)szOut);

                                   while (pActualAttribute!=NULL)
                                   {
                                          _stprintf(szTemp,L"%S",
                                                 (*pActualAttribute).value);
                                          if (_tcscmp(szTemp,szOut)==0)
                                          {
                                                 break;
                                          }
                                          else
                                          {
                                             pActualAttribute
                                              =(*pActualAttribute).nextAttribute;
                                          }
                                   }

                                   if (iSoundOn==1)
                                          PlayWaveResource
                                              (IDR_AUDIT_WAVE_CONFIRM,hInst,0);
                                   if (iSoundOn==2)
                                          PlayWaveResource
                                              (IDR_EARCON_WAVE_CONFIRM,hInst,0);
                                   //iAskUserResult stays the same
                                   EndDialog (hWnd,0);
                                   return TRUE;
                     }
                     break;
       }
       return FALSE;
}

// ------------------------------------------------------------------
//
// [CustomizedTourMode] SelectSubCatDetails Dialog Procedure
//
// ------------------------------------------------------------------

BOOL CALLBACK SelectSubCatDetailsDlgProc (HWND hWnd, UINT wMsg, WPARAM wParam,
LPARAM lParam)
{
       INT i,cItems;
       TCHAR szOut[MAX_STRINGSIZE],szTemp[MAX_STRINGSIZE];
       INT lpnItems[MAX_ARTOBJECTS];
       RoomList * pTempRoomList;
       ArtObject * pTempArtObject;
       int count,rc;

       switch (wMsg)
       {
              case WM_INITDIALOG:

              // Fill the list, knowing that :
              // iAskUserResult==1 -> pActualAttribute relates to an Author
              // * iAskUserResult==2 -> pActualAttribute relates to a  Style
              // * iAskUserResult==3 -> pActualAttribute relates to a  Period
              // * iAskUserResult==4 -> pActualAttribute relates to a  Country
              // * iAskUserResult==5 -> pActualAttribute relates to a  Material
```

```
count=0;
if (pActualAttribute!=NULL)
{
        // seek the ActualAttribute's roomlist to find
        // corresponding attribute values

        pTempRoomList
         =(*pActualAttribute).firstCorrespondingRoom;
        while (pTempRoomList!=NULL)
        {
                pTempArtObject
           =(*(*pTempRoomList).roomToSee).firstObjectInRoom;

                while (pTempArtObject!=NULL)
                {
                        switch (iAskUserResult)
                        {
```

```
case 1: //author
        if(strcmp((*pActualAttribute).value,(*pTempArtObject).author)==0)
        {
                //we must reset isToSee in case of [<Back]
                (*pTempArtObject).isToSee=0;

                wsprintf(szOut, TEXT("%S"),(*pTempArtObject).name);

                SendDlgItemMessage(hWnd,IDC_SUBCATDETAILSLIST,
                        LB_ADDSTRING,0,(LPARAM)szOut);

                count++;
        }
        break;

case 2: //style
        if (strcmp((*pActualAttribute).value,(*pTempArtObject).style)==0)
        {
                //we must reset isToSee in case of [<Back]
                (*pTempArtObject).isToSee=0;

                wsprintf(szOut, TEXT("%S"),(*pTempArtObject).name);

                SendDlgItemMessage(hWnd,IDC_SUBCATDETAILSLIST,
                        LB_ADDSTRING,0,(LPARAM)szOut);

                count++;
        }
        break;

case 3: //period
        if(strcmp((*pActualAttribute).value,(*pTempArtObject).period)==0)
        {
                //we must reset isToSee in case of [<Back]
                (*pTempArtObject).isToSee=0;

                wsprintf(szOut, TEXT("%S"),(*pTempArtObject).name);

                SendDlgItemMessage(hWnd,IDC_SUBCATDETAILSLIST,
                        LB_ADDSTRING,0,(LPARAM)szOut);

                count++;
        }
        break;

case 4: //country
        if (strcmp((*pActualAttribute).value,
                (*pTempArtObject).country)==0)
```

```
{
        //we must reset isToSee in case of [<Back]
        (*pTempArtObject).isToSee=0;

        wsprintf(szOut, TEXT("%S"),(*pTempArtObject).name);

        SendDlgItemMessage(hWnd,IDC_SUBCATDETAILSLIST,
              LB_ADDSTRING,0,(LPARAM)szOut);

        count++;
}
break;

case 5: //material
      if (strcmp((*pActualAttribute).value,
            (*pTempArtObject).material)==0)
                                                                {
        //we must reset isToSee in case of [<Back]
        (*pTempArtObject).isToSee=0;

        wsprintf(szOut, TEXT("%S"),(*pTempArtObject).name);

        SendDlgItemMessage(hWnd,IDC_SUBCATDETAILSLIST,
              LB_ADDSTRING,0,(LPARAM)szOut);

        count++;
}
break;
                                          }

pTempArtObject=(*pTempArtObject).nextObjectInRoom;
                                    }

pTempRoomList=(*pTempRoomList).nextRoomInList;
                           }
                  }

                  //select all by default

SendDlgItemMessage(hWnd,IDC_SUBCATDETAILSLIST,LB_SELITEMRANGE,

(WPARAM)(BOOL)TRUE,MAKELPARAM(0,count-1));
                  return TRUE;

        case WM_COMMAND:

                  switch (LOWORD (wParam))
                  {
                           // warning C4700: local variable 'count' used
                           // without having been initialized…
                           // => don't worry, count will be initialized by
                           // WM_INITDIALOG ! ;-)

                           case BTN_SELECTSUBCATDETAILS_SELECTALL:

                                    if (iSoundOn==1)
                                             PlayWaveResource
                                               (IDR_AUDIT_WAVE_RADIOBTN,hInst,0);
                                    if (iSoundOn==2)
                                             PlayWaveResource
                                               (IDR_EARCON_WAVE_RADIOBTN,hInst,0);
                                    SendDlgItemMessage(hWnd,
                                             IDC_SUBCATDETAILSLIST,
                                             LB_SELITEMRANGE,
                                             (WPARAM)(BOOL)TRUE,
```

```
                                MAKELPARAM(0,count-1));
                        break;

                case BTN_SELECTSUBCATDETAILS_DESELECTALL:

                        if (iSoundOn==1)
                                PlayWaveResource
                                  (IDR_AUDIT_WAVE_RADIOBTN,hInst,0);
                        if (iSoundOn==2)
                                PlayWaveResource
                                  (IDR_EARCON_WAVE_RADIOBTN,hInst,0);
                        SendDlgItemMessage(hWnd,
                                IDC_SUBCATDETAILSLIST,
                                LB_SELITEMRANGE,
                                (WPARAM)(BOOL)FALSE,
                                MAKELPARAM(0,count-1));
                        break;

                case BTN_SELECTSUBCATDETAILS_BACK:

                        if (iSoundOn==1)
                                PlayWaveResource
                                  (IDR_AUDIT_WAVE_CANCEL,hInst,0);
                        if (iSoundOn==2)
                                PlayWaveResource
                                  (IDR_EARCON_WAVE_CANCEL,hInst,0);
                        iAskUserResult=0-iAskUserResult;
                        EndDialog (hWnd,0);
                        return TRUE;

                case BTN_SELECTSUBCATDETAILS_CONTINUE:
                //[Begin!] button
```

```
// retrieve number of selections
cItems
 = SendDlgItemMessage(hWnd,IDC_SUBCATDETAILSLIST,LB_GETSELCOUNT,0,0);

// retrieve selected items identifier (in the listbox)
SendDlgItemMessage(hWnd,IDC_SUBCATDETAILSLIST,LB_GETSELITEMS,
        (WPARAM) cItems ,(LPARAM)(LPINT) lpnItems);

for (i=0;i<cItems;i++)
{
        // retrieve the selected items from (from the listbox)

        SendDlgItemMessage(hWnd,IDC_SUBCATDETAILSLIST,LB_GETTEXT,
                (WPARAM) lpnItems[i],(LPARAM)(LPCTSTR) szOut);

        //set the selected items to "ToSee"

        pTempRoomList=(*pActualAttribute).firstCorrespondingRoom;

        while (pTempRoomList!=NULL)
        {
                pTempArtObject
                  =(*(*pTempRoomList).roomToSee).firstObjectInRoom;

                while (pTempArtObject!=NULL)
                {
                        // we must convert (*pTempArtObject).name to TCHAR[]
                        // because szOut, which contains the selected
                        // artobject's name, is a TCHAR[]
                        _stprintf(szTemp,L"%S",(*pTempArtObject).name);

                        if (_tcscmp(szTemp,szOut)==0)
```

```
                        {
                                (*pTempArtObject).isToSee=1;
                        }

                        pTempArtObject=(*pTempArtObject).nextObjectInRoom;
                }

                pTempRoomList=(*pTempRoomList).nextRoomInList;
        }
}

if (cItems!=0)
{
        if (iSoundOn==1)
                PlayWaveResource (IDR_AUDIT_WAVE_CONFIRM,hInst,0);
        if (iSoundOn==2)
                PlayWaveResource (IDR_EARCON_WAVE_CONFIRM,hInst,0);
        EndDialog (hWnd,0);
}
else
{
        rc = MessageBox(hWnd,
                TEXT("You must select at least one item!"),
                TEXT("Sub Category Selection"),
                MB_ICONEXCLAMATION | MB_OKCANCEL);
        if (rc==IDOK)
        {
                if (iSoundOn==1)
                        PlayWaveResource (IDR_AUDIT_WAVE_CONFIRM,hInst,0);
                if (iSoundOn==2)
                        PlayWaveResource (IDR_EARCON_WAVE_CONFIRM,hInst,0);
        }
        else // user cancels
        {
                if (iSoundOn==1)
                        PlayWaveResource (IDR_AUDIT_WAVE_CONFIRM,hInst,0);
                if (iSoundOn==2)
                        PlayWaveResource (IDR_EARCON_WAVE_CONFIRM,hInst,0);
                iAskUserResult=0;
                EndDialog (hWnd,0);
        }
}
return TRUE;
```

```
                        }
                        break;
        }
        return FALSE;
}

// ==================================================================
//
// DIALOG BOXES "WRAPPING" Procedures
//
// ==================================================================
// used to manage the [<Back] and [Next>] buttons of each dialog box
// ==================================================================

BOOL SelectTourModeWrapProc (HWND hWnd)
{
        //INT rc;

        //Call the SelectTourMode dialog procedure
        DialogBox(hInst, TEXT("SelectTourMode"),hWnd,SelectTourModeDlgProc);

        if (iAskUserResult>0) //user didn't cancel
```

```
    {
        iTaskLevel=0;
        //the task level is 0 because we are starting a new tour.
        if (iTourMode ==0) //if user selected FreeTourMode
        {
            iTaskLevel=1;
        }
        else  //if user selected CustomizedTourMode
        {
            //Call the SelectTourByCategory WRAPPING procedure
            SelectTourByCategoryWrapProc(hWnd);
        }
    }
    // else : the user cancelled -> do nothing !
    return TRUE;
}

BOOL SelectTourByCategoryWrapProc (HWND hWnd)
{
    //Call the SelectTourByCategory dialog procedure
    DialogBox(hInst, TEXT("SelectTourByCategory"),
        hWnd,SelectTourByCategoryDlgProc);

    if (iAskUserResult>0)
    //user didn't cancel the selection -> iAskUserResult==Category Selection
    {
        //Call the SelectSubCatDetails WRAPPING procedure
        SelectSubCategoryWrapProc(hWnd);
    }
    else
    {
        if (iAskUserResult<0) // user wants to go back!
        {
            iAskUserResult=0-iAskUserResult;
            //Call the previous WRAPPING procedure
            SelectTourModeWrapProc(hWnd);
        }

        //else : the user cancelled -> do nothing !
    }
    return TRUE;
}

BOOL SelectSubCategoryWrapProc(HWND hWnd)
{
    //Call the SelectSubCategory dialog procedure
    DialogBox(hInst, TEXT("SelectSubCategory"),
        hWnd,SelectSubCategoryDlgProc);

    if (iAskUserResult>0)
    //user didn't cancel the selection -> iAskUserResult==Category Selection
    {
        //Call the SelectSubCatDetails WRAPPING procedure
        SelectSubCatDetailsWrapProc(hWnd);
    }
    else
    {
        if (iAskUserResult<0) // user wants to go back!
        {
            iAskUserResult=0-iAskUserResult;
            //Call the previous WRAPPING procedure
            SelectTourByCategoryWrapProc(hWnd);
        }

        //else : the user cancelled -> do nothing !
```

```
        }
        return TRUE;
}

BOOL SelectSubCatDetailsWrapProc(HWND hWnd)
{
        //Call the SelectSubCatDetails dialog procedure
        DialogBox(hInst, TEXT("SelectSubCatDetails"),
                hWnd,SelectSubCatDetailsDlgProc);

        if (iAskUserResult>0)
        //user didn't cancel the selection -> iAskUserResult==Category Selection
        {
                //Start Tour!
                iTaskLevel=1;
        }
        else
        {
                if (iAskUserResult<0) // user wants to go back!
                {
                        iAskUserResult=0-iAskUserResult;
                        //Call the previous WRAPPING procedure
                        SelectSubCategoryWrapProc(hWnd);
                }

                //else : the user cancelled -> do nothing !
        }       return TRUE;
}

// ==============================================================
//
// THREAD HANDLER ROUTINES
//
// ==============================================================

// --------------------------------------------------------------
//
// ListenIrThread : listens for incoming IR messages
//
// --------------------------------------------------------------
// for our test, the incomming msg are supplied by the "Simulate IR
// Signal" command of the menu
// --------------------------------------------------------------

int ListenIrThread(PVOID pArg) //see p550-574
{
        HWND hWnd;
        INT rc;
        RoomList * pTempRoomList;
        TCHAR szTempl[MAX_STRINGSIZE];

        hWnd = (HWND)pArg;

        // while the application in running, listen for IR messages
        while (1)
        {
                rc=WaitForSingleObject(hListenIrEvent,INFINITE);
                if (rc==WAIT_OBJECT_0)
                {
                        if (!fContinue)
                        {
                                break;
                        }
                        //read message
```

```
// First, check if the signal comes from the currently IR
// signaled room

if (pIrRoom!=NULL)
{
        _stprintf(szTemp1,L"%S",(*pIrRoom).name);
        if (_tcscmp(szTemp1,szIrMsg)==0)
        {
                // we are still in the same IR signaled room
                // Do not play sound when redrawing the
                //current screen,
                // except if we are in overview map
                iDrawNew=0;
        }
        else // the signal comes from another room
        {
                //find the corresponding room
                pTempRoomList=pFirstRoomInList;
                while (pTempRoomList!=NULL)
                {
                        _stprintf(szTemp1,L"%S",
                          (*(*pTempRoomList).roomToSee).name);
                        if (_tcscmp(szTemp1,szIrMsg)==0)
                        {
                                //Wait 1 sec
                                Sleep(1000);

                                if (iSoundOn==1)
                                        PlayWaveResource
                            (IDR_AUDIT_WAVE_ROOM2ROOM,hInst,0);
                                if (iSoundOn==2)
                                        PlayWaveResource
                            (IDR_EARCON_WAVE_ROOM2ROOM,hInst,0);

                                pIrRoom
                                 =(*pTempRoomList).roomToSee;

                                //Do not play sound when
                                //redrawing the current screen,
                                //except if we are in overview
                                //map
                                iDrawNew=0;

                                //force to redraw the screen
                                InvalidateRect(hWnd,NULL,TRUE);

                                break;
                        }
                        else
                        {
                             pTempRoomList
                               =(*pTempRoomList).nextRoomInList;
                        }
                }
        }
}
else
//pIrRoom is NULL, i.e. this is the first IR
//signal received.
{
        // find the corresponding room
        pTempRoomList=pFirstRoomInList;
        while (pTempRoomList!=NULL)
        {
                _stprintf(szTemp1,L"%S",
```

```
                                        (*(*pTempRoomList).roomToSee).name);
                                if (_tcscmp(szTemp1,szIrMsg)==0)
                                {
                                        //Wait 1 sec
                                        Sleep(1000);


                                        if (iSoundOn==1)
                                                PlayWaveResource
                                           (IDR_AUDIT_WAVE_ROOM2ROOM,hInst,0);
                                        if (iSoundOn==2)
                                                PlayWaveResource
                                           (IDR_EARCON_WAVE_ROOM2ROOM,hInst,0);
                                        pIrRoom=(*pTempRoomList).roomToSee;

                                        //Do not play sound when redrawing the
                                        //current screen,
                                        //except if we are in overview map
                                        iDrawNew=0;

                                        //force to redraw the screen
                                        InvalidateRect(hWnd,NULL,TRUE);

                                        break;
                                }
                                else
                                {
                                        pTempRoomList
                                         =(*pTempRoomList).nextRoomInList;
                                }
                        }
                }
            }
        }

        return 0;
}

// ---------------------------------------------------------------
//
// CreateCustomHatchBrush : creates custom hatched brushes
//
// ---------------------------------------------------------------
// The following fn Styles are defined in "test1.h" :
// * HS_HORIZONTAL  ----
// * HS_VERTICAL    ||||
// * HS_FDIAGONAL   \\\\
// * HS_FDIAGONAL   ////
// * HS_FDIAGONAL          ++++
// * HS_DIAGCROSS   xxxx
// ---------------------------------------------------------------

HBRUSH CreateCustomHatchBrush(INT fnStyle, COLORREF clrrefbackgrnd, COLORREF
clrrefforegrnd)
{
        BRUSHBMP brbmp;
        BYTE * pBytes;
        int i;

        DWORD dwBits[6][2] =
                {       {0x000000ff,0x00000000},
                        {0x10101010,0x10101010},
                        {0x01020408,0x10204080},
                        {0x80402010,0x08040201},
                        {0x101010ff,0x10101010},
```

```
                {0x81422418,0x18244281},
        };

    if ((fnStyle<0)||(fnStyle>dim(dwBits)))
    {
        return 0;
    }

    memset(&brbmp,0,sizeof(brbmp));

    brbmp.bmi.biSize          = sizeof(BITMAPINFOHEADER);
    brbmp.bmi.biWidth         = 8;
    brbmp.bmi.biHeight        = 8;
    brbmp.bmi.biPlanes        = 1;
    brbmp.bmi.biBitCount      = 1;
    brbmp.bmi.biClrUsed       = 2;
    brbmp.bmi.biClrImportant= 2;

    // initialize the palette of the bitmap
    brbmp.dwPal[0] = PALETTERGB((BYTE)((clrrefbackgrnd >> 16) & 0xff),
                               (BYTE)((clrrefbackgrnd >> 8) & 0xff),
                               (BYTE)(clrrefbackgrnd & 0xff));
    brbmp.dwPal[1] = PALETTERGB((BYTE)((clrrefforegrnd >> 16) & 0xff),
                               (BYTE)((clrrefforegrnd >> 8) & 0xff),
                               (BYTE)(clrrefforegrnd & 0xff));

    // write the hatch data to the bitmap
    pBytes = (BYTE *)&dwBits[fnStyle];
    for (i=0;i<8;i++)
    {
        brbmp.bBits[i*4] = *pBytes++;
    }

    // return the handle of the brush created
    return CreateDIBPatternBrushPt(&brbmp,DIB_RGB_COLORS);
}
```

E.3 ravi-guide.rc

```
// PROGRAM RESOURCE FILE

#include "windows.h"
#include "commctrl.h"
#include "ravi-guide.h"

// ============================================================================
// Menubar
// ============================================================================

ID_MENU MENU DISCARDABLE
BEGIN
    POPUP "File"
    BEGIN
      MENUITEM "New Tour",                    IDM_NEWTOUR
      MENUITEM "Restart Tour",                IDM_RESTART_TOUR
      MENUITEM "End Tour",                    IDM_END_TOUR
      MENUITEM SEPARATOR
      MENUITEM "Exit",                        IDM_EXIT
    END
    POPUP "Sound"
      BEGIN
      MENUITEM "Auditory Pattern On",         IDM_AUDIT_PATTERN_ON
      MENUITEM "Earcon Pattern On",           IDM_EARCON_PATTERN_ON
      MENUITEM "Sound Enhancement Off",       IDM_SOUND_OFF
    END
    POPUP "IR Sig"
    BEGIN
        MENUITEM "Simulate IR Signal",        IDM_SIMULATE_IR
    END
      POPUP "Help"
    BEGIN
        MENUITEM "About...",                  IDM_ABOUT
    END
END

// ============================================================================
// Dialog Boxes
// ============================================================================

SELECTTOURMODE DIALOG 10, 10, 120, 75
STYLE DS_CENTER | WS_POPUP | WS_VISIBLE | WS_CAPTION
// | WS_SYSMENU // WS_SYSMENU==cancel button on title bar
// | EXSTYLE WS_EX_CAPTIONOKBTN //WS_EX_CAPTIONOKBTN==ok button on title bar
CAPTION "Select Tour Mode"
BEGIN
    LTEXT           "Please select a Tour Mode :",IDD_PLEASESELECTTOURMODE,
                                       5,5,100,15
    CONTROL         "Free Tour Mode",IDD_RADIO_FREE,
                                "Button",BS_AUTORADIOBUTTON,
                                       5,20,100,15
    CONTROL         "Customized Tour Mode",IDD_RADIO_CUSTOMIZED,
                                "Button",BS_AUTORADIOBUTTON,
                                       5,35,100,15
    DEFPUSHBUTTON   "Continue",BTN_SELECTTOURMODE_CONTINUE,
                                       40,55,40,15
END

SELECTTOURBYCATEGORY DIALOG DISCARDABLE  10, 10, 120, 120
STYLE DS_CENTER | WS_POPUP | WS_VISIBLE | WS_CAPTION
CAPTION "Customize Tour Mode"
BEGIN
```

```
    LTEXT          "Please select a Tour Category :",IDD_PLEASESELECTTOURBYCAT,
                                     5,5,100,15
    CONTROL          ARTOBJECT_ATTRIBUTE1,IDD_ARTOBJECT_ATTRIBUTE1,
                                 "Button",BS_AUTORADIOBUTTON,
                                     5,20,100,15
    CONTROL          ARTOBJECT_ATTRIBUTE2,IDD_ARTOBJECT_ATTRIBUTE2,
                                 "Button",BS_AUTORADIOBUTTON,
                                     5,35,100,15
    CONTROL          ARTOBJECT_ATTRIBUTE3,IDD_ARTOBJECT_ATTRIBUTE3,
                                 "Button",BS_AUTORADIOBUTTON,
                                     5,50,100,15
    CONTROL          ARTOBJECT_ATTRIBUTE4,IDD_ARTOBJECT_ATTRIBUTE4,
                                 "Button",BS_AUTORADIOBUTTON,
                                     5,65,100,15
    CONTROL          ARTOBJECT_ATTRIBUTE5,IDD_ARTOBJECT_ATTRIBUTE5,
                                 "Button",BS_AUTORADIOBUTTON,
                                     5,80,100,15
    DEFPUSHBUTTON    "Next >",BTN_SELECTTOURBYCATEGORY_CONTINUE,
                                     65,100,40,15
    PUSHBUTTON       "< Back",BTN_SELECTTOURBYCATEGORY_BACK,
                                     15,100,40,15
END

SELECTSUBCATEGORY DIALOG DISCARDABLE  10, 10, 120, 120
STYLE DS_CENTER | WS_POPUP | WS_VISIBLE | WS_CAPTION
CAPTION "Customize Tour Mode"
BEGIN
    LTEXT            "Please select a Sub Category :",IDD_PLEASESELECTSUBCAT,
                                     5,5,100,15
    LISTBOX                 IDC_SUBCATEGORYLIST,
                                     5,20,110,75,
                                     LBS_SORT | WS_VSCROLL,
    DEFPUSHBUTTON    "Next >",BTN_SELECTSUBCATEGORY_CONTINUE,
                                     65,100,40,15
    PUSHBUTTON       "< Back",BTN_SELECTSUBCATEGORY_BACK,
                                     15,100,40,15
END

SELECTSUBCATDETAILS DIALOG DISCARDABLE  10, 10, 120, 120
STYLE DS_CENTER | WS_POPUP | WS_VISIBLE | WS_CAPTION
CAPTION "Customize Tour Mode"
BEGIN
    LTEXT            "Please select the Sub Category Details :",
                      IDD_PLEASESELECTSUBCATDETAILS,
                                     5,5,100,15
    LISTBOX          IDC_SUBCATDETAILSLIST,
                                     5,20,110,58,
                     LBS_SORT | LBS_MULTIPLESEL | WS_VSCROLL,

    DEFPUSHBUTTON    "Begin!",BTN_SELECTSUBCATDETAILS_CONTINUE,
                                     65,100,40,15
    PUSHBUTTON       "< Back",BTN_SELECTSUBCATDETAILS_BACK,
                                     15,100,40,15

    PUSHBUTTON       "Select All",BTN_SELECTSUBCATDETAILS_SELECTALL,
                                     10,80,45,15
    PUSHBUTTON       "Deselect All",BTN_SELECTSUBCATDETAILS_DESELECTALL,
                                     65,80,45,15
END

SelectRoomIR DIALOG DISCARDABLE  10, 10, 120, 120
STYLE DS_CENTER | WS_POPUP | WS_VISIBLE | WS_CAPTION
CAPTION "Simulate InfraRed Signal"
BEGIN
    LTEXT           "Please select the room from which you want to simulate the
```

```
        .           IR signal :",
                    IDD_PLEASESELECTROOMIR,
                                        5,5,100,15
    LISTBOX         IDC_ROOMIRYLIST,
                                        5,20,110,75,
                                        WS_VSCROLL,
    PUSHBUTTON      "Send Signal",IDD_SELECTROOMIR_SEND,
                                        15,100,45,15
    PUSHBUTTON      "Cancel",IDD_SELECTROOMIR_CANCEL,
                                        65,100,45,15
END

ABOUTBOX DIALOG DISCARDABLE  10, 10, 120, 45
STYLE DS_MODALFRAME | DS_CENTER | WS_POPUP | WS_VISIBLE | WS_CAPTION |
    WS_SYSMENU
CAPTION "About"
BEGIN
        LTEXT           PRG_VERSION,
                        -1,10,5,100,15
        LTEXT           "Written for Ravi Ramdoyal's Master Thesis in 2002-2003.",
                        -1,10,20,100,25
END

// ========================================================================
// Wave files
// ========================================================================

// Auditory Icon Pattern

IDR_AUDIT_WAVE_CONFIRM            WAVE   sounds\confirm.wav
IDR_AUDIT_WAVE_CANCEL             WAVE   sounds\cancel.wav
IDR_AUDIT_WAVE_RADIOBTN           WAVE   sounds\radiobutton1.wav

IDR_AUDIT_WAVE_ROOM2OVERVIEW      WAVE   sounds\audit_room2overview.wav
IDR_AUDIT_WAVE_OVERVIEW2ROOM      WAVE   sounds\audit_overview2room.wav
IDR_AUDIT_WAVE_ROOM2ROOM          WAVE   sounds\audit_room2room.wav
IDR_AUDIT_WAVE_ARTOBJECT          WAVE   sounds\audit_artobject.wav

// Earcon Pattern

IDR_EARCON_WAVE_CONFIRM           WAVE   sounds\confirm.wav
IDR_EARCON_WAVE_CANCEL            WAVE   sounds\cancel.wav
IDR_EARCON_WAVE_RADIOBTN          WAVE   sounds\radiobutton1.wav

IDR_EARCON_WAVE_ROOM2OVERVIEW     WAVE   sounds\earcon_room2overview.wav
IDR_EARCON_WAVE_OVERVIEW2ROOM     WAVE   sounds\earcon_overview2room.wav
IDR_EARCON_WAVE_ROOM2ROOM         WAVE   sounds\earcon_room2room.wav
IDR_EARCON_WAVE_ARTOBJECT         WAVE   sounds\earcon_artobject.wav

// ========================================================================
// Bitmap files
// ========================================================================

IDR_DEFAULT_BMP            BITMAP images\defaultdesk.bmp
```

E.4 multimedia.h

```
// ================================================================
// HEADER FILE for multimedia.cpp
// ================================================================

BOOL    PlayWaveResource    (const    int    iResourceWaveID,HINSTANCE    hInst,int
syncMode);
// Plays the wave sound identified by iResourceWaveID in "test1.h"
// * hInst should be the instance of the program
// * syncMode == 1 --> synchronous playback (otherwise async playback)
// returns TRUE if played correctly
// returns FALSE if the resource could not be found or loaded

BOOL PlayWaveFromFileWithTCHAR (TCHAR TszBitmapFileID[]);
// Plays the wave file specified by TszBitmapFileID
// returns TRUE if played correctly
// returns FALSE if the resource could not be found or loaded

int   DrawBitmapResource   (const   int   iResourceBitmapID,HINSTANCE   myhInst,HDC
myhdc,RECT myClientRect);
// Draws a resource bitmap on the screen, horizontally centered
// returns the height of the displayed image (which can be shrinked)

int      DrawBitmapFromFileWithTCHAR      (TCHAR      TszBitmapFileID[],HINSTANCE
myhInst2,HDC myhdc2,RECT myClientRect2);
// Draws a resource bitmap on the screen, horizontally centered
// returns the height of the displayed image (eventually shrinked)
```

E.5 multimedia.cpp

```
// =================================================================
// Multimedia handling methods for the program
// =================================================================

#include <windows.h>        // for all the windows stuff
#include <stdio.h>          // for string manipulation
#include "ravi-guide.h"

BOOL   PlayWaveResource   (const   int   iResourceWaveID,HINSTANCE   myhInst,int
syncMode)
// Plays the wave sound identified by iResourceWaveID in "test1.h"
// * hInst should be the instance of the program
// * syncMode == 1 --> synchronous playback (otherwise async playback)
// returns TRUE if played correctly
// returns FALSE if the resource could not be found or loaded
{
  BOOL bRtn;
  LPTSTR lpRes;
  HRSRC hResInfo;
  HANDLE hRes;

  // Find the wave resource.
  //hResInfo = FindResource (myhInst, lpName, TEXT("WAVE"));
  hResInfo = FindResource(myhInst,
                  MAKEINTRESOURCE(iResourceWaveID),TEXT("WAVE"));

  if (hResInfo == NULL)
    return FALSE;

  // Load the wave resource.
  hRes = LoadResource (myhInst, hResInfo);

  if (hRes == NULL)
    return FALSE;

  // Lock the wave resource and play it.
  lpRes = (LPTSTR) LockResource (hRes);

  if (lpRes != NULL)
    {
    if (syncMode)
            bRtn = sndPlaySound (lpRes,
                        SND_MEMORY | SND_SYNC | SND_NODEFAULT);
      else //asyn mode
            bRtn = sndPlaySound (lpRes,
                        SND_MEMORY | SND_ASYNC | SND_NODEFAULT);
    }
  else
    bRtn = 0;

  return bRtn;
}

BOOL PlayWaveFromFileWithTCHAR (TCHAR TszBitmapFileID[])
// Plays the wave file specified by TszBitmapFileID
// returns TRUE if played correctly
// returns FALSE if the resource could not be found or loaded
{
      return PlaySound( TszBitmapFileID,NULL,
                  SND_FILENAME | SND_NODEFAULT | SND_ASYNC);
      //SND_NODEFAULT : If the sound cannot be found, the function
      //returns silently without playing the default sound.
```

```
}

int  DrawBitmapResource  (const  int  iResourceBitmapID,HINSTANCE  myhInst,HDC
myhdc,RECT myClientRect)
// Draws a resource bitmap on the screen, horizontally centered
// returns the height of the displayed image (which can be shrinked)
{
	HGDIOBJ hOldSel;
	BITMAP bmp;

	int iShrinkCoeff=1;

	// Get the bitmap
	HBITMAP hBitmap = LoadBitmap(myhInst,
	                    MAKEINTRESOURCE(iResourceBitmapID));

	// Create a memory DC (device context) that matches the device
	HDC hdcMem = CreateCompatibleDC(myhdc);

	// Select the bitmap into the compatible device context
	hOldSel = SelectObject(hdcMem, hBitmap);

	// Get the bitmap dimensions from the bitmap
	GetObject(hBitmap, sizeof (BITMAP),&bmp);

	//Copy the bitmap image from the memory DC to the screen DC
	if ((myClientRect.right-myClientRect.left)>bmp.bmWidth
		&& (myClientRect.bottom-myClientRect.top)>(bmp.bmHeight))
	{
		BitBlt(myhdc,myClientRect.left
			+(((myClientRect.right-myClientRect.left)-bmp.bmWidth)/2),

	myClientRect.top,bmp.bmWidth,bmp.bmHeight,hdcMem,0,0,SRCCOPY);
	}
	else //the image doesn't fit in the client rectangle -> shrink it !
	{
		int iLengthCoeff
			= bmp.bmWidth/(myClientRect.right-myClientRect.left);
		int iHeightCoeff
			= bmp.bmHeight/(myClientRect.bottom-myClientRect.top);

		if (iLengthCoeff<=iHeightCoeff)
		{
			iShrinkCoeff=iHeightCoeff+1;
		}
		else //(iLengthCoeff>iHeigthCoeff)
		{
			iShrinkCoeff=iLengthCoeff+1;
		}

		StretchBlt(myhdc,myClientRect.left
				+(((myClientRect.right-myClientRect.left)
				-(bmp.bmWidth/iShrinkCoeff))/2),
				myClientRect.top,
				bmp.bmWidth/iShrinkCoeff,
				bmp.bmHeight/iShrinkCoeff,
				hdcMem,0,0,
				bmp.bmWidth,bmp.bmHeight,SRCCOPY);
	}

	//Restore original bitmap selection and destroy the memory DC
	SelectObject (hdcMem,hOldSel);
	DeleteDC(hdcMem);

	return bmp.bmHeight/iShrinkCoeff;
```

```
}

int      DrawBitmapFromFileWithTCHAR      (TCHAR      TszBitmapFileID[],HINSTANCE
myhInst2,HDC myhdc2,RECT myClientRect2)
// Draws a resource bitmap on the screen, horizontally centered
// returns the height of the displayed image (eventually shrinked)
{
	HGDIOBJ hOldSel;
	BITMAP bmp;

	int iShrinkCoeff=1;

	//Get the bitmap (SHLoadDIBitmap loads a bitmap from the specified file)
	HBITMAP hBitmap = SHLoadDIBitmap(TszBitmapFileID);

	if (hBitmap!=NULL)
	{
		// Create a memory DC (device context) that matches the device
		HDC hdcMem = CreateCompatibleDC(myhdc2);

		// Select the bitmap into the compatible device context
		hOldSel = SelectObject(hdcMem, hBitmap);

		// Get the bitmap dimensions from the bitmap
		GetObject(hBitmap, sizeof (BITMAP),&bmp);

		//Copy the bitmap image from the memory DC to the screen DC
		if ((myClientRect2.right-myClientRect2.left)>bmp.bmWidth
			&& (myClientRect2.bottom-myClientRect2.top)>(bmp.bmHeight))
		{
			BitBlt(myhdc2,myClientRect2.left
				+(((myClientRect2.right-myClientRect2.left)
				-bmp.bmWidth)/2),
				myClientRect2.top,
				bmp.bmWidth,bmp.bmHeight,hdcMem,0,0,SRCCOPY);
		}
		else //the image doesn't fit in the client rectangle->shrink it!
		{
			int iLengthCoeff
			  =bmp.bmWidth/(myClientRect2.right-myClientRect2.left);
			int iHeightCoeff
			  =bmp.bmHeight/(myClientRect2.bottom-myClientRect2.top);

			if (iLengthCoeff<=iHeightCoeff)
			{
				iShrinkCoeff=iHeightCoeff+1;
			}
			else //(iLengthCoeff>iHeigthCoeff)
			{
				iShrinkCoeff=iLengthCoeff+1;
			}

			StretchBlt(myhdc2,myClientRect2.left
					+(((myClientRect2.right-myClientRect2.left)
					-(bmp.bmWidth/iShrinkCoeff))/2),
					myClientRect2.top,
					bmp.bmWidth/iShrinkCoeff,
					bmp.bmHeight/iShrinkCoeff,
					hdcMem,0,0,
					bmp.bmWidth,bmp.bmHeight,SRCCOPY);
		}

		//Restore original bitmap selection and destroy the memory DC
		SelectObject (hdcMem,hOldSel);
		DeleteDC(hdcMem);
```

```
        return bmp.bmHeight/iShrinkCoeff;
    }

    else
    {
        //display the default resource bitmap
        return DrawBitmapResource (IDR_DEFAULT_BMP,
                    myhInst2,myhdc2,myClientRect2);
    }
}
```

E.6 init.h

```
// ===============================================================
// HEADER FILE for init.cpp
// ===============================================================

#include "attributelist.h"
// "attributelist.h" includes "attribute.h"
// "attribute.h"     includes "room.h"
// "room.h"          includes "artobject.h"

BOOL     InitFromFileWithTCHAR     (TCHAR     TszFileName[],     RoomList     *
pFirstRoomInTheList);
//returns TRUE if the file was read correctly
//returns FALSE otherwise

BOOL InitDefaultEnvironment(RoomList * pFirstRoomInTheList);
//returns TRUE if the initialization was done correctly
//returns FALSE otherwise
```

E.7 init.cpp

```
// ================================================================
// init core file
// ================================================================

#include <windows.h>        // for all the windows stuff
#include <stdio.h>          // for string manipulation
#include "tchar.h"          // for TCHAR routing mapping
#include "init.h"

BOOL    InitFromFileWithTCHAR    (TCHAR    TszFileName[],    RoomList    *
pFirstRoomInTheList)
//returns TRUE if the file was read correctly
//returns FALSE otherwise
{
       HANDLE hFile;

       // Open the existing file.
       hFile = CreateFile (TszFileName,         // Open szFileName
                      GENERIC_READ,             // Open for reading
                      0,                        // Do not share
                      NULL,                     // No security
                      OPEN_EXISTING,            // Existing file only
                      FILE_ATTRIBUTE_NORMAL,    // Normal file
                      NULL);                    // No template file

       if (hFile == INVALID_HANDLE_VALUE)
       {
              return FALSE;
       }

       //Parsing variables
       DWORD dwBytesRead;
       char *kaBuff=(char*) malloc(1*sizeof(char));
       char lineBuffer[4096];
       int iLineCount;
       int iCharCount;
       int iRoomCount;
       int iCounter,iSubCounter;

       BOOL stringStartsWithSpace;
       char *pSubstring;
       int  iPosSubstring;

       int iParsingLevel;
       //ROOM HEADER
       //0 == #newroom
       //1 == #code=<Code of the Room>
       //2 == #name=<Name of the Room>
       //3 == #type=<Type of the Room>
       //40 == #numberofcoordinates=<Nrd>
       //41 == #coordinates=
       //                  <xi>,<yi>

       //CONNEXIONS
       //50 == #numberofconnexions=<Ncnx>
       //51 == #connexioncode=<Code of the Ncnxth Connected Room>
       //52 == #coordinates=
       //                  <x1>,<y1>
       //                  <x2>,<y2>

       //ARTOBJECTS
       //60 == #numberofartobjects=<Nao>
```

```
//61 == #code=<Code of the 1st Object>
//62 == #name=<Name of the 1st Object>
//63 == #attrib1=<Author of the 1st Object>
//64 == #attrib2=<Style of the 1st Object>
//65 == #attrib3=<Period of the 1st Object>
//66 == #attrib4=<Country of the 1st Object>
//67 == #attrib5=<Material of the 1st Object>
//68 == #coordinates=
//                  <x1>,<y1>
//                  <x2>,<y2>

//Room variables (see "rooom.h")

Room *myRoom;
char   szRoomCode          [MAX_STRINGSIZE];
char   szRoomName          [MAX_STRINGSIZE];
char   szRoomType          [MAX_STRINGSIZE];
int    iRoomNumberOfCoordinates;
POINT  roomCoordinatesTable[MAX_COORDINATES];
int    iRoomNumberOfAdjacentRooms;
char   roomAdjacentRoomsCodeTable[MAX_CONNEXIONS][MAX_STRINGSIZE];
POINT  roomConnexionsA     [MAX_CONNEXIONS];
POINT  roomConnexionsB     [MAX_CONNEXIONS];

char   szRoomNumberOf[MAX_STRINGSIZE];
int    iNumberOfArtObjects;

//ArtObject variables

ArtObject * firstArtObject=NULL;
ArtObject * lastArtObject=NULL;
ArtObject * currentArtObject=NULL;
char szArtObjectCode       [MAX_STRINGSIZE];
char szArtObjectName       [MAX_STRINGSIZE];
char szArtObjectAuthor     [MAX_STRINGSIZE];
char szArtObjectStyle      [MAX_STRINGSIZE];
char szArtObjectPeriod     [MAX_STRINGSIZE];
char szArtObjectCountry    [MAX_STRINGSIZE];
char szArtObjectMaterial   [MAX_STRINGSIZE];
POINT artObjectPositionA={0,0};
POINT artObjectPositionB={0,0};

//init variables
iParsingLevel=0;
strcpy(lineBuffer,"");
dwBytesRead=1;
iLineCount=0;
iCharCount=0;
iRoomCount=0;
iCounter=0;

while (dwBytesRead == 1)
{
      //read line char by char (1 char "=" 1 Byte)
      do
      {
            ReadFile (hFile, kaBuff, 1, &dwBytesRead, NULL);

            if (dwBytesRead == 1)
            {
                  if (((strncmp(kaBuff," ",1)==0)&&(iCharCount==0))
                        ||(strncmp(kaBuff,"\r",1)==0)
                        ||(strncmp(kaBuff,"\n",1)==0))
                  {
                        //do nothing!
```

```
				}
				else
				{
					strncat(lineBuffer,kaBuff,1);
				}
			}
		}
		while ((strncmp(kaBuff,"\r",1)!=0)
			&&(strncmp(kaBuff,"\n",1)!=0)
			&&(dwBytesRead == 1));

		//analyse line
		if (strcmp(lineBuffer,"")!=0)
		{
			//see if line is a comment
			pSubstring = strstr(lineBuffer,"//");
			iPosSubstring = pSubstring - lineBuffer + 1;
			if( pSubstring == NULL )
			{
				switch (iParsingLevel)
				{
					case 0: //#newroom
					{
//find if we just read the header of a newroom
pSubstring = strstr(lineBuffer,"#newroom");
iPosSubstring = pSubstring - lineBuffer + 1;
if( pSubstring == NULL )
{
	//the file is not well formatted!!

	// Close file.
	CloseHandle (hFile);
	return FALSE;
}
else
{
	iLineCount++;
	iParsingLevel=1;
	strcpy(lineBuffer,"");
}
break;
					}

					case 1: //#code
					{
pSubstring = strstr(lineBuffer,"#code=");
iPosSubstring = pSubstring - lineBuffer + 1;
if( pSubstring == NULL )
{
	//the file is not well formatted!!

	// Close file.
	CloseHandle (hFile);
	return FALSE;
}
else
{
	iLineCount++;
	strcpy(szRoomCode,"");

	strncat(szRoomCode,lineBuffer+strlen("#code="),
		strlen(lineBuffer)-strlen("#code="));
```

```
        stringStartsWithSpace=TRUE;
        while (stringStartsWithSpace==TRUE)
        {

                if(strncmp(szRoomCode," ",1)==0)
                {
                        strcpy(szRoomCode,szRoomCode+1);
                }
                else
                {
                        stringStartsWithSpace=FALSE;
                }
        }

        iParsingLevel=2;
        strcpy(lineBuffer,"");
}
break;
```

```
                                }

                                case 2: //#name
                                {
```

```
pSubstring = strstr(lineBuffer,"#name=");
iPosSubstring = pSubstring - lineBuffer + 1;
if( pSubstring == NULL )
{
        //the file is not well formatted!!

        // Close file.
        CloseHandle (hFile);
        return FALSE;
}
else
{
        iLineCount++;
        strcpy(szRoomName,"");

        strncat(szRoomName,lineBuffer+strlen("#name="),
                strlen(lineBuffer)-strlen("#name="));

        stringStartsWithSpace=TRUE;
        while (stringStartsWithSpace==TRUE)
        {
                if(strncmp(szRoomName," ",1)==0)
                {
                        strcpy(szRoomName,szRoomName+1);
                }
                else
                {
                        stringStartsWithSpace=FALSE;
                }
        }

        iParsingLevel=3;
        strcpy(lineBuffer,"");
}
break;
```

```
                                }

                                case 3: //#type
                                {
```

```
pSubstring = strstr(lineBuffer,"#type=");
iPosSubstring = pSubstring - lineBuffer + 1;
```

```
if( pSubstring == NULL )
{
	//the file is not well formatted!!

	// Close file.
	CloseHandle (hFile);
	return FALSE;
}
else
{
	iLineCount++;
	strcpy(szRoomType,"");

	strncat(szRoomType,lineBuffer+strlen("#type="),
		strlen(lineBuffer)-strlen("#type="));

	stringStartsWithSpace=TRUE;
	while (stringStartsWithSpace==TRUE)
	{
		if(strncmp(szRoomType," ",1)==0)
		{
			strcpy(szRoomType,szRoomType+1);
		}
		else
		{
			stringStartsWithSpace=FALSE;
		}
	}

	iParsingLevel=40;
	strcpy(lineBuffer,"");
}
break;
```

```
				}

				case 40: //#numberofcoordinates
				{
```

```
pSubstring = strstr(lineBuffer,"#numberofcoordinates=");
iPosSubstring = pSubstring - lineBuffer + 1;
if( pSubstring == NULL )
{
	//the file is not well formatted!!

	// Close file.
	CloseHandle (hFile);
	return FALSE;
}
else
{
	iLineCount++;
	strcpy(szRoomNumberOf,"");
	strncat(szRoomNumberOf,lineBuffer+strlen("#numberofcoordinates="),
strlen(lineBuffer)-strlen("#numberofcoordinates="));

	stringStartsWithSpace=TRUE;
	while (stringStartsWithSpace==TRUE)
	{
		if(strncmp(szRoomNumberOf," ",1)==0)
		{
			strcpy(szRoomNumberOf,szRoomNumberOf+1);
		}
		else
		{
			stringStartsWithSpace=FALSE;
			iRoomNumberOfCoordinates=atoi(szRoomNumberOf);
```

```
•           }
        }

        //a room should have at leeat 3 coordinates
        if (iRoomNumberOfCoordinates>=3)
        {
                iParsingLevel=41;
        }
        else
        {
                //the file is not well formatted!!

                // Close file.
                CloseHandle (hFile);
                return FALSE;
        }
        iCounter=0;
        strcpy(lineBuffer,"");
}
break;
```

```
                                }

                                case 41: //#coordinates
                                {
```

```
if (iCounter==0)
{
        pSubstring = strstr(lineBuffer,"#coordinates=");
        iPosSubstring = pSubstring - lineBuffer + 1;
        if( pSubstring == NULL )
        {
                //the file is not well formatted!!

                // Close file.
                CloseHandle (hFile);
                return FALSE;
        }
        else
        {
                iCounter++;
        }
}
else
{
        strcpy(szRoomNumberOf,"");
        iCharCount=strlen(lineBuffer);
        iPosSubstring=0;
        while ((iPosSubstring<iCharCount)
                &&(strncmp(lineBuffer+iPosSubstring,",",1)!=0))
        {
                strncat(szRoomNumberOf,lineBuffer+iPosSubstring,1);
                iPosSubstring++;
        }

        roomCoordinatesTable[iCounter-1].x=atoi(szRoomNumberOf);
        strcpy(szRoomNumberOf,"");

        strncat(szRoomNumberOf,lineBuffer+iPosSubstring+1,
                iCharCount-iPosSubstring-1);

        roomCoordinatesTable[iCounter-1].y=atoi(szRoomNumberOf);
        iCounter++;
}

if (iCounter>iRoomNumberOfCoordinates)
{
```

```
-       iParsingLevel=50;
}

strcpy(lineBuffer,"");
break;
                                      }

                                      case 50: //#numberofconnexions
                                      {
pSubstring = strstr(lineBuffer,"#numberofconnexions=");
iPosSubstring = pSubstring - lineBuffer + 1;
if( pSubstring == NULL )
{
        //the file is not well formatted!!

        // Close file.
        CloseHandle (hFile);
        return FALSE;
}
else
{
        iLineCount++;
        strcpy(szRoomNumberOf,"");

        strncat(szRoomNumberOf,lineBuffer+strlen("#numberofconnexions="),
                strlen(lineBuffer)-strlen("#numberofconnexions="));

        stringStartsWithSpace=TRUE;
        while (stringStartsWithSpace==TRUE)
        {
                if(strncmp(szRoomNumberOf," ",1)==0)
                {
                        strcpy(szRoomNumberOf,szRoomNumberOf+1);
                }
                else
                {
                        stringStartsWithSpace=FALSE;
                        iRoomNumberOfAdjacentRooms=atoi(szRoomNumberOf);
                }
        }

        if (iRoomNumberOfAdjacentRooms>0)
        {
                iParsingLevel=51;
        }
        else
        {
                iParsingLevel=60;
        }
        iCounter=0;
        strcpy(lineBuffer,"");
}
break;
                                      }

                                      case 51: //#connexioncode
                                      {
pSubstring = strstr(lineBuffer,"#connexioncode=");
iPosSubstring = pSubstring - lineBuffer + 1;
if( pSubstring == NULL )
{
        //the file is not well formatted!!

        // Close file.
```

```
        CloseHandle (hFile);
        return FALSE;
}
else
{
        iLineCount++;
        strcpy(roomAdjacentRoomsCodeTable[iCounter],"");

        strncat(roomAdjacentRoomsCodeTable[iCounter],
                lineBuffer+strlen("#connexioncode="),
                strlen(lineBuffer)-strlen("#connexioncode="));

        stringStartsWithSpace=TRUE;
        while (stringStartsWithSpace==TRUE)
        {
                if(strncmp(roomAdjacentRoomsCodeTable[iCounter]," ",1)==0)
                {
                        strcpy(roomAdjacentRoomsCodeTable[iCounter],
                                roomAdjacentRoomsCodeTable[iCounter]+1);
                }
                else
                {
                        stringStartsWithSpace=FALSE;
                }
        }

        iSubCounter=0;
        iParsingLevel=52;
        strcpy(lineBuffer,"");
}
break;
```

```
                                        }

                                        case 52: //#coordinates
                                        {
```

```
if (iSubCounter==0)
{
        pSubstring = strstr(lineBuffer,"#coordinates=");
        iPosSubstring = pSubstring - lineBuffer + 1;
        if( pSubstring == NULL )
        {
                //the file is not well formatted!!

                // Close file.
                CloseHandle (hFile);
                return FALSE;
        }
        else
        {
                iSubCounter++;
        }
}
else
{
        if (iSubCounter==1)
        {
                strcpy(szRoomNumberOf,"");
                iCharCount=strlen(lineBuffer);
                iPosSubstring=0;
                while ((iPosSubstring<iCharCount)
                        &&(strncmp(lineBuffer+iPosSubstring,",",1)!=0))
                {
                        strncat(szRoomNumberOf,lineBuffer+iPosSubstring,1);
                        iPosSubstring++;
                }
```

```
                roomConnexionsA[iCounter].x=atoi(szRoomNumberOf);
                strcpy(szRoomNumberOf,"");

                strncat(szRoomNumberOf,lineBuffer+iPosSubstring+1,
                        iCharCount-iPosSubstring-1);

                roomConnexionsA[iCounter].y=atoi(szRoomNumberOf);
                iSubCounter++;
        }
        else //(iSubCounter==2)
        {
                strcpy(szRoomNumberOf,"");
                iCharCount=strlen(lineBuffer);
                iPosSubstring=0;
                while ((iPosSubstring<iCharCount)
                        &&(strncmp(lineBuffer+iPosSubstring,",",1)!=0))
                {
                        strncat(szRoomNumberOf,lineBuffer+iPosSubstring,1);
                        iPosSubstring++;
                }

                roomConnexionsB[iCounter].x=atoi(szRoomNumberOf);
                strcpy(szRoomNumberOf,"");

                strncat(szRoomNumberOf,lineBuffer+iPosSubstring+1,
                        iCharCount-iPosSubstring-1);

                roomConnexionsB[iCounter].y=atoi(szRoomNumberOf);
                iSubCounter++;

                //don't forget to increase the number of connexions read!
                iCounter++;
        }
}

if (iSubCounter==3)
{
        if (iCounter<iRoomNumberOfAdjacentRooms)
        {
                iParsingLevel=51;
        }
        else
        {
                iParsingLevel=60;
        }
}
strcpy(lineBuffer,"");
break;
                                }

                                case 60: //#numberofartobjects
                                {
pSubstring = strstr(lineBuffer,"#numberofartobjects=");
iPosSubstring = pSubstring - lineBuffer + 1;
if( pSubstring == NULL )
{
        //the file is not well formatted!!

        // Close file.
        CloseHandle (hFile);
        return FALSE;
}
else
{
```

```
        iLineCount++;
        strcpy(szRoomNumberOf,"");

        strncat(szRoomNumberOf,lineBuffer+strlen("#numberofartobjects="),
                strlen(lineBuffer)-strlen("#numberofartobjects="));

        stringStartsWithSpace=TRUE;
        while (stringStartsWithSpace==TRUE)
        {
                if(strncmp(szRoomNumberOf," ",1)==0)
                {
                        strcpy(szRoomNumberOf,szRoomNumberOf+1);
                }
                else
                {
                        stringStartsWithSpace=FALSE;
                        iNumberOfArtObjects=atoi(szRoomNumberOf);
                }
        }

        if (iNumberOfArtObjects>0)
        {
                iParsingLevel=61;
        }
        else
        {
                //end of room declaration -> create room
                myRoom=new Room(szRoomCode,szRoomName,szRoomType,
                                iRoomNumberOfCoordinates,
                                roomCoordinatesTable,
                                iRoomNumberOfAdjacentRooms,
                                roomAdjacentRoomsCodeTable,
                                roomConnexionsA,roomConnexionsB);

                //add room to roomlist
                (*pFirstRoomInTheList).addRoom(myRoom);
                iRoomCount++;
                iParsingLevel=0;
        }
        iCounter=0;
        strcpy(lineBuffer,"");
}
break;
```

```
                                }
                                case 61: //#code
                                {
```

```
pSubstring = strstr(lineBuffer,"#code=");
iPosSubstring = pSubstring - lineBuffer + 1;
if( pSubstring == NULL )
{
        //the file is not well formatted!!

        // Close file.
        CloseHandle (hFile);
        return FALSE;
}
else
{
        iLineCount++;
        strcpy(szArtObjectCode,"");
        strncat(szArtObjectCode,lineBuffer+strlen("#code="),
                strlen(lineBuffer)-strlen("#code="));

        stringStartsWithSpace=TRUE;
        while (stringStartsWithSpace==TRUE)
```

```
	{
		if(strncmp(szArtObjectCode," ",1)==0)
		{
			strcpy(szArtObjectCode,szArtObjectCode+1);
		}
		else
		{
			stringStartsWithSpace=FALSE;
		}
	}

	iParsingLevel=62;
	strcpy(lineBuffer,"");
}
break;
				}

				case 62: //#name
				{
pSubstring = strstr(lineBuffer,"#name=");
iPosSubstring = pSubstring - lineBuffer + 1;
if( pSubstring == NULL )
{
	//the file is not well formatted!!

	// Close file.
	CloseHandle (hFile);
	return FALSE;
}
else
{
	iLineCount++;
	strcpy(szArtObjectName,"");
	strncat(szArtObjectName,lineBuffer+strlen("#name="),
		len(lineBuffer)-strlen("#name="));

	stringStartsWithSpace=TRUE;
	while (stringStartsWithSpace==TRUE)
	{
		if(strncmp(szArtObjectName," ",1)==0)
		{
			strcpy(szArtObjectName,szArtObjectName+1);
		}
		else
		{
			stringStartsWithSpace=FALSE;
		}
	}

	iParsingLevel=63;
	strcpy(lineBuffer,"");
}
break;
				}

				case 63: //#attrib1
				{
pSubstring = strstr(lineBuffer,"#attrib1=");
iPosSubstring = pSubstring - lineBuffer + 1;
if( pSubstring == NULL )
{
	//the file is not well formatted!!

	// Close file.
```

```
•       CloseHandle (hFile);
        return FALSE;
}
else
{
        iLineCount++;
        strcpy(szArtObjectAuthor,"");
        strncat(szArtObjectAuthor,lineBuffer+strlen("#attrib1="),
                strlen(lineBuffer)-strlen("#attrib1="));

        stringStartsWithSpace=TRUE;
        while (stringStartsWithSpace==TRUE)
        {
                if(strncmp(szArtObjectAuthor," ",1)==0)
                {
                        strcpy(szArtObjectAuthor,szArtObjectAuthor+1);
                }
                else
                {
                        stringStartsWithSpace=FALSE;
                }
        }

        iParsingLevel=64;
        strcpy(lineBuffer,"");
}
break;
```

```
                                        }

                                        case 64: //#attrib2
                                        {
```

```
pSubstring = strstr(lineBuffer,"#attrib2=");
iPosSubstring = pSubstring - lineBuffer + 1;
if( pSubstring == NULL )
{
        //the file is not well formatted!!

        // Close file.
        CloseHandle (hFile);
        return FALSE;
}
else
{
        iLineCount++;
        strcpy(szArtObjectStyle,"");
        strncat(szArtObjectStyle,lineBuffer+strlen("#attrib2="),
                strlen(lineBuffer)-strlen("#attrib2="));

        stringStartsWithSpace=TRUE;
        while (stringStartsWithSpace==TRUE)
        {
                if(strncmp(szArtObjectStyle," ",1)==0)
                {
                        strcpy(szArtObjectStyle,szArtObjectStyle+1);
                }
                else
                {
                        stringStartsWithSpace=FALSE;
                }
        }

        iParsingLevel=65;
        strcpy(lineBuffer,"");
}
break;
```

```
                                }

                                case 65: //#attrib3
                                {
pSubstring = strstr(lineBuffer,"#attrib3=");
iPosSubstring = pSubstring - lineBuffer + 1;
if( pSubstring == NULL )
{
        //the file is not well formatted!!

        // Close file.
        CloseHandle (hFile);
        return FALSE;
}
else
{
        iLineCount++;
        strcpy(szArtObjectPeriod,"");
        strncat(szArtObjectPeriod,lineBuffer+strlen("#attrib3="),
                strlen(lineBuffer)-strlen("#attrib3="));

        stringStartsWithSpace=TRUE;
        while (stringStartsWithSpace==TRUE)
        {
                if(strncmp(szArtObjectPeriod," ",1)==0)
                {
                        strcpy(szArtObjectPeriod,szArtObjectPeriod+1);
                }
                else
                {
                        stringStartsWithSpace=FALSE;
                }
        }

        iParsingLevel=66;
        strcpy(lineBuffer,"");
}
break;
                                }

                                case 66: //#attrib4
                                {
pSubstring = strstr(lineBuffer,"#attrib4=");
iPosSubstring = pSubstring - lineBuffer + 1;
if( pSubstring == NULL )
{
        //the file is not well formatted!!

        // Close file.
        CloseHandle (hFile);
        return FALSE;
}
else
{
        iLineCount++;
        strcpy(szArtObjectCountry,"");
        strncat(szArtObjectCountry,lineBuffer+strlen("#attrib4="),
                strlen(lineBuffer)-strlen("#attrib4="));

        stringStartsWithSpace=TRUE;
        while (stringStartsWithSpace==TRUE)
        {
                if(strncmp(szArtObjectCountry," ",1)==0)
                {
```

```
                        strcpy(szArtObjectCountry,szArtObjectCountry+1);
                }
                else
                {
                        stringStartsWithSpace=FALSE;
                }
        }

        iParsingLevel=67;
        strcpy(lineBuffer,"");
}
break;
```

```
                                        }

                                        case 67: //#attrib5
                                        {
```

```
pSubstring = strstr(lineBuffer,"#attrib5=");
iPosSubstring = pSubstring - lineBuffer + 1;
if( pSubstring == NULL )
{
        //the file is not well formatted!!

        // Close file.
        CloseHandle (hFile);
        return FALSE;
}
else
{
        iLineCount++;
        strcpy(szArtObjectMaterial,"");
        strncat(szArtObjectMaterial,lineBuffer+strlen("#attrib5="),
                strlen(lineBuffer)-strlen("#attrib5="));

        stringStartsWithSpace=TRUE;
        while (stringStartsWithSpace==TRUE)
        {
                if(strncmp(szArtObjectMaterial," ",1)==0)
                {
                        strcpy(szArtObjectMaterial,szArtObjectMaterial+1);
                }
                else
                {
                        stringStartsWithSpace=FALSE;
                }
        }
        iParsingLevel=68;
        iSubCounter=0;
        strcpy(lineBuffer,"");
}
break;
```

```
                                        }

                                        case 68: //#coordinates
                                        {
```

```
if (iSubCounter==0)
{
        pSubstring = strstr(lineBuffer,"#coordinates=");
        iPosSubstring = pSubstring - lineBuffer + 1;
        if( pSubstring == NULL )
        {
                //the file is not well formatted!!

                // Close file.
                CloseHandle (hFile);
```

```
			return FALSE;
		}
		else
		{
			iSubCounter++;
		}
}
else
{
		if (iSubCounter==1)
		{
			strcpy(szRoomNumberOf,"");
			iCharCount=strlen(lineBuffer);
			iPosSubstring=0;
			while ((iPosSubstring<iCharCount)
				&&(strncmp(lineBuffer+iPosSubstring,",",1)!=0))
			{
				strncat(szRoomNumberOf,lineBuffer+iPosSubstring,1);
				iPosSubstring++;
			}

			artObjectPositionA.x=atoi(szRoomNumberOf);
			strcpy(szRoomNumberOf,"");
			strncat(szRoomNumberOf,lineBuffer+iPosSubstring+1,
				iCharCount-iPosSubstring-1);

			artObjectPositionA.y=atoi(szRoomNumberOf);
			iSubCounter++;
		}
		else //(iSubCounter==2)
		{
			strcpy(szRoomNumberOf,"");
			iCharCount=strlen(lineBuffer);
			iPosSubstring=0;
			while ((iPosSubstring<iCharCount)
				&&(strncmp(lineBuffer+iPosSubstring,",",1)!=0))
			{
				strncat(szRoomNumberOf,lineBuffer+iPosSubstring,1);
				iPosSubstring++;
			}

			artObjectPositionB.x=atoi(szRoomNumberOf);
			strcpy(szRoomNumberOf,"");
			strncat(szRoomNumberOf,lineBuffer+iPosSubstring+1,
				iCharCount-iPosSubstring-1);

			artObjectPositionB.y=atoi(szRoomNumberOf);
			iSubCounter++;

			//create art object

			if(firstArtObject==NULL)
			{
				firstArtObject=new ArtObject(szArtObjectCode,
							szArtObjectName,
							szArtObjectAuthor,
							szArtObjectStyle,
							szArtObjectPeriod,
							szArtObjectCountry,
							szArtObjectMaterial,
							artObjectPositionA,
							ArtObjectPositionB,
							0,0);
```

```
                lastArtObject=firstArtObject;
            }
            else
            {

                currentArtObject= new ArtObject(szArtObjectCode,
                                    szArtObjectName,
                                    szArtObjectAuthor,
                                    szArtObjectStyle,
                                    szArtObjectPeriod,
                                    szArtObjectCountry,
                                    szArtObjectMaterial,
                                    artObjectPositionA,
                                    ArtObjectPositionB,
                                    0,0);

                (*lastArtObject).nextObjectInRoom
                    =currentArtObject;

                lastArtObject=currentArtObject;
            }

            //don't forget to increase the number of artobjects read!
            iCounter++;
        }
}

if (iSubCounter==3)
{
        if (iCounter<iNumberOfArtObjects)
        {
            iParsingLevel=61;
            iSubCounter=0;
        }
        else
        {
            //create room
            myRoom=new Room(szRoomCode,szRoomName,szRoomType,
                iRoomNumberOfCoordinates,roomCoordinatesTable,
                iRoomNumberOfAdjacentRooms,
                roomAdjacentRoomsCodeTable,
                roomConnexionsA,roomConnexionsB);

            (*myRoom).firstObjectInRoom=firstArtObject;

            //add room to roomlist
            (*pFirstRoomInTheList).addRoom(myRoom);

            //re-initialise artobject pointers
            firstArtObject=NULL;
            lastArtObject=NULL;
            currentArtObject=NULL;
            iRoomCount++;
            iParsingLevel=0;
        }
}

//don't forget to reinitialize lineBuffer !
strcpy(lineBuffer,"");
break;
```

```
                        }
                    }
            }
            //if line is a comment, don't forget to reinitialize
            //lineBuffer !
```

```
                strcpy(lineBuffer,"");
            }
        }

        // Close file.
        CloseHandle (hFile);

        if (iRoomCount>0)
        {
            return TRUE;
        }
        else
        {
            return FALSE;
        }
}

BOOL InitDefaultEnvironment(RoomList * pFirstRoomInTheList)
//returns TRUE if the initialization was done correctly
//returns FALSE otherwise

// NOTA BENE : The default environment is a section of the CNR of Pisa,
// where I had my training period :-)

{
        POINT ptemp1,ptemp2; //for temp artobject

        // ----------------------------------------------
        // Hall1
        // ----------------------------------------------

        Room *hall1;
        POINT hall1coord [MAX_COORDINATES];
        char  hall1cnx   [MAX_CONNEXIONS][MAX_STRINGSIZE];
        POINT hall1cnxA  [MAX_CONNEXIONS];
        POINT hall1cnxB  [MAX_CONNEXIONS];

        hall1coord[0].x=245;        hall1coord[0].y=350;
        hall1coord[1].x=200;        hall1coord[1].y=350;
        hall1coord[2].x=200;        hall1coord[2].y=240;
        hall1coord[3].x=185;        hall1coord[3].y=240;
        hall1coord[4].x=185;        hall1coord[4].y=210;
        hall1coord[5].x=245;        hall1coord[5].y=210;
        hall1coord[6].x=245;        hall1coord[6].y=240;
        hall1coord[7].x=215;        hall1coord[7].y=240;
        hall1coord[8].x=215;        hall1coord[8].y=315;
        hall1coord[9].x=245;        hall1coord[9].y=315;

        strcpy(hall1cnx[0],"Office 9");
        hall1cnxA[0].x =205;        hall1cnxA[0].y =350;
        hall1cnxB[0].x =215;        hall1cnxB[0].y =350;

        strcpy(hall1cnx[1],"Office 12");
        hall1cnxA[1].x =215;        hall1cnxA[1].y =305;
        hall1cnxB[1].x =215;        hall1cnxB[1].y =310;

        strcpy(hall1cnx[2],"Office 13");
        hall1cnxA[2].x =215;        hall1cnxA[2].y =245;
        hall1cnxB[2].x =215;        hall1cnxB[2].y =250;

        strcpy(hall1cnx[3],"Office 15");
        hall1cnxA[3].x =205;        hall1cnxA[3].y =210;
        hall1cnxB[3].x =210;        hall1cnxB[3].y =210;

        strcpy(hall1cnx[4],"Hall 2");
```

```
hall1cnxA[4].x =185;      hall1cnxA[4].y =220;
hall1cnxB[4].x =185;      hall1cnxB[4].y =235;

strcpy(hall1cnx[5],"Exit");
hall1cnxA[5].x =245;      hall1cnxA[5].y =230;
hall1cnxB[5].x =245;      hall1cnxB[5].y =235;

strcpy(hall1cnx[6],"Exit");
hall1cnxA[6].x =245;      hall1cnxA[6].y =325;
hall1cnxB[6].x =245;      hall1cnxB[6].y =330;

hall1 = new Room("Hall 1","Hall 1", "Hall",
             10,hall1coord,7,hall1cnx,hall1cnxA,hall1cnxB);

(*pFirstRoomInTheList).addRoom(hall1);

// ---------------------------------------------
// Hall2
// ---------------------------------------------

Room *hall2;
POINT hall2coord [MAX_COORDINATES];
char hall2cnx    [MAX_CONNEXIONS][MAX_STRINGSIZE];
POINT hall2cnxA  [MAX_CONNEXIONS];
POINT hall2cnxB  [MAX_CONNEXIONS];

hall2coord[0].x =185;     hall2coord[0].y =240;
hall2coord[1].x =150;     hall2coord[1].y =240;
hall2coord[2].x =150;     hall2coord[2].y =195;
hall2coord[3].x =170;     hall2coord[3].y =185;
hall2coord[4].x =170;     hall2coord[4].y =155;
hall2coord[5].x =45;      hall2coord[5].y =155;
hall2coord[6].x =45;      hall2coord[6].y =145;
hall2coord[7].x =175;     hall2coord[7].y =145;
hall2coord[8].x =175;     hall2coord[8].y =0;
hall2coord[9].x =190;     hall2coord[9].y =0;
hall2coord[10].x=190;     hall2coord[10].y=210;
hall2coord[11].x=185;     hall2coord[11].y=210;

strcpy(hall2cnx[0],"Hall 1");
hall2cnxA[0].x =185;      hall2cnxA[0].y =220;
hall2cnxB[0].x =185;      hall2cnxB[0].y =235;

strcpy(hall2cnx[1],"Office 11b");
hall2cnxA[1].x =175;      hall2cnxA[1].y =240;
hall2cnxB[1].x =180;      hall2cnxB[1].y =240;

strcpy(hall2cnx[2],"Office 15");
hall2cnxA[2].x =190;      hall2cnxA[2].y =185;
hall2cnxB[2].x =190;      hall2cnxB[2].y =190;

strcpy(hall2cnx[3],"Office 16");
hall2cnxA[3].x =190;      hall2cnxA[3].y =170;
hall2cnxB[3].x =190;      hall2cnxB[3].y =175;

strcpy(hall2cnx[4],"Office 17");
hall2cnxA[4].x =190;      hall2cnxA[4].y =125;
hall2cnxB[4].x =190;      hall2cnxB[4].y =130;

strcpy(hall2cnx[5],"Office 18");
hall2cnxA[5].x =190;      hall2cnxA[5].y =110;
hall2cnxB[5].x =190;      hall2cnxB[5].y =115;

strcpy(hall2cnx[6],"Office 19");
hall2cnxA[6].x =190;      hall2cnxA[6].y =65;
```

```
hall2cnxB[6].x =190;       hall2cnxB[6].y =70;

strcpy(hall2cnx[7],"Office 20");
hall2cnxA[7].x =190;       hall2cnxA[7].y =50;
hall2cnxB[7].x =190;       hall2cnxB[7].y =55;

strcpy(hall2cnx[8],"Office 21");
hall2cnxA[8].x =190;       hall2cnxA[8].y =10;
hall2cnxB[8].x =190;       hall2cnxB[8].y =15;

strcpy(hall2cnx[9],"Office IAT");
hall2cnxA[9].x =65;        hall2cnxA[9].y =155;
hall2cnxB[9].x =75;        hall2cnxB[9].y =155;

strcpy(hall2cnx[10],"Office IEI");
hall2cnxA[10].x =65;       hall2cnxA[10].y =145;
hall2cnxB[10].x =75;       hall2cnxB[10].y =145;

strcpy(hall2cnx[11],"Office CED1");
hall2cnxA[11].x =165;      hall2cnxA[11].y =145;
hall2cnxB[11].x =175;      hall2cnxB[11].y =145;

strcpy(hall2cnx[12],"Office CED2");
hall2cnxA[12].x =175;      hall2cnxA[12].y =70;
hall2cnxB[12].x =175;      hall2cnxB[12].y =75;

strcpy(hall2cnx[13],"Office CED3");
hall2cnxA[13].x =175;      hall2cnxA[13].y =30;
hall2cnxB[13].x =175;      hall2cnxB[13].y =35;

strcpy(hall2cnx[14],"Office CED4");
hall2cnxA[14].x =175;      hall2cnxA[14].y =5;
hall2cnxB[14].x =175;      hall2cnxB[14].y =15;

strcpy(hall2cnx[15],"Office 11");
hall2cnxA[15].x =155;      hall2cnxA[15].y =240;
hall2cnxB[15].x =165;      hall2cnxB[15].y =240;

strcpy(hall2cnx[16],"Hall 3");
hall2cnxA[16].x =45;       hall2cnxA[16].y =147;
hall2cnxB[16].x =45;       hall2cnxB[16].y =153;

hall2 = new Room("Hall 2","Hall 2", "Hall",12,
             hall2coord,17,hall2cnx,hall2cnxA,hall2cnxB);

(*pFirstRoomInTheList).addRoom(hall2);

// ---------------------------------------------
// Hall3
// ---------------------------------------------

Room *hall3;
POINT hall3coord  [MAX_COORDINATES];
char  hall3cnx    [MAX_CONNEXIONS][MAX_STRINGSIZE];
POINT hall3cnxA   [MAX_CONNEXIONS];
POINT hall3cnxB   [MAX_CONNEXIONS];

hall3coord[0].x =35;       hall3coord[0].y =240;
hall3coord[1].x =10;              hall3coord[1].y =240;
hall3coord[2].x =10;              hall3coord[2].y =115;
hall3coord[3].x =0;        hall3coord[3].y =115;
hall3coord[4].x =0;        hall3coord[4].y =85;
hall3coord[5].x =10;       hall3coord[5].y =85;
hall3coord[6].x =10;       hall3coord[6].y =15;
hall3coord[7].x =0;        hall3coord[7].y =15;
```

```
hall3coord[8].x =0;         hall3coord[8].y =5;
hall3coord[9].x =10;        hall3coord[9].y =5;
hall3coord[10].x=10;        hall3coord[10].y=0;
hall3coord[11].x=35;        hall3coord[11].y=0;
hall3coord[12].x =35;       hall3coord[12].y =20;
hall3coord[13].x =45;       hall3coord[13].y =20;
hall3coord[14].x =45;       hall3coord[14].y =30;
hall3coord[15].x =35;       hall3coord[15].y =30;
hall3coord[16].x =35;       hall3coord[16].y =145;
hall3coord[17].x =45;       hall3coord[17].y =145;
hall3coord[18].x =45;       hall3coord[18].y =155;
hall3coord[19].x =35;       hall3coord[19].y =155;

strcpy(hall3cnx[0],"Hall 2");
hall3cnxA[0].x =45; hall3cnxA[0].y =147;
hall3cnxB[0].x =45; hall3cnxB[0].y =153;

strcpy(hall3cnx[1],"Office CED3");
hall3cnxA[1].x =45; hall3cnxA[1].y =22;
hall3cnxB[1].x =45; hall3cnxB[1].y =28;

strcpy(hall3cnx[2],"WC 1");
hall3cnxA[2].x =10; hall3cnxA[2].y =165;
hall3cnxB[2].x =10; hall3cnxB[2].y =175;

strcpy(hall3cnx[3],"WC 2");
hall3cnxA[3].x =10; hall3cnxA[3].y =130;
hall3cnxB[3].x =10; hall3cnxB[3].y =140;

strcpy(hall3cnx[4],"WC 3");
hall3cnxA[4].x =10; hall3cnxA[4].y =25;
hall3cnxB[4].x =10; hall3cnxB[4].y =35;

strcpy(hall3cnx[5],"Office ??");
hall3cnxA[5].x =0;  hall3cnxA[5].y =7;
hall3cnxB[5].x =0;  hall3cnxB[5].y =13;

strcpy(hall3cnx[6],"Exit");
hall3cnxA[6].x =20; hall3cnxA[6].y =0;
hall3cnxB[6].x =25; hall3cnxB[6].y =0;

hall3 = new Room("Hall 3","Hall 3", "Hall",20,
             hall3coord,7,hall3cnx,hall3cnxA,hall3cnxB);

(*pFirstRoomInTheList).addRoom(hall3);

// -------------------------------------------
// Office 8
// -------------------------------------------

Room *office8;
POINT office8coord [MAX_COORDINATES];
char  office8cnx   [MAX_CONNEXIONS][MAX_STRINGSIZE];
POINT office8cnxA  [MAX_CONNEXIONS];
POINT office8cnxB  [MAX_CONNEXIONS];

office8coord[0].x =125;   office8coord[0].y =410;
office8coord[1].x =125;         office8coord[1].y =300;
office8coord[2].x =200;         office8coord[2].y =300;
office8coord[3].x =200;         office8coord[3].y =410;

strcpy(office8cnx[0],"Office ???");
office8cnxA[0].x =125;    office8cnxA[0].y =385;
office8cnxB[0].x =125;    office8cnxB[0].y =390;
```

```
office8 = new Room("Office 8","Office 8", "Office",4,
             office8coord,1,office8cnx,office8cnxA,office8cnxB);

(*pFirstRoomInTheList).addRoom(office8);

// ---------------------------------------------
// Office 9
// ---------------------------------------------

Room *office9;
POINT office9coord [MAX_COORDINATES];
char  office9cnx   [MAX_CONNEXIONS][MAX_STRINGSIZE];
POINT office9cnxA  [MAX_CONNEXIONS];
POINT office9cnxB  [MAX_CONNEXIONS];

office9coord[0].x =200;          office9coord[0].y =410;
office9coord[1].x =200;          office9coord[1].y =350;
office9coord[2].x =245;          office9coord[2].y =350;
office9coord[3].x =245;          office9coord[3].y =410;

strcpy(office9cnx[0],"Hall 1");
office9cnxA[0].x =205;     office9cnxA[0].y =350;
office9cnxB[0].x =215;     office9cnxB[0].y =350;

office9 = new Room("Office 9","Office 9", "Office",4,
             office9coord,1,office9cnx,office9cnxA,office9cnxB);

(*pFirstRoomInTheList).addRoom(office9);

// ---------------------------------------------
// Office 11
// ---------------------------------------------

Room *office11;
POINT office11coord [MAX_COORDINATES];
char  office11cnx   [MAX_CONNEXIONS][MAX_STRINGSIZE];
POINT office11cnxA  [MAX_CONNEXIONS];
POINT office11cnxB  [MAX_CONNEXIONS];

office11coord[0].x =125;         office11coord[0].y =240;
office11coord[1].x =125;         office11coord[1].y =300;
office11coord[2].x =170;         office11coord[2].y =300;
office11coord[3].x =170;         office11coord[3].y =240;

strcpy(office11cnx[0],"Hall 2");
office11cnxA[0].x =155;    office11cnxA[0].y =240;
office11cnxB[0].x =165;    office11cnxB[0].y =240;

office11 = new Room("Office 11","Office 11", "Office",4,
             office11coord,1,office11cnx,office11cnxA,office11cnxB);

(*pFirstRoomInTheList).addRoom(office11);

ArtObject *office11Object0;
ArtObject *office11Object1;
ArtObject *office11Object2;

ptemp1.x=125;ptemp1.y =250;
ptemp2.x=135;ptemp2.y =280;
office11Object0 = new ArtObject("Code110","Temp Desk 0","Temp User",
             "Temp Desk","2002","England","Wood",ptemp1,ptemp2,0,0);

ptemp1.x=160;ptemp1.y =250;
ptemp2.x=170;ptemp2.y =280;
office11Object1 = new ArtObject("Code111","Temp Desk 1","Temp User",
```

```
        "Temp Desk","2002","Italy","Wood and Metal",ptemp1,ptemp2,0,0);

ptemp1.x=135;ptemp1.y =280;
ptemp2.x=160;ptemp2.y =300;
office11Object2 = new ArtObject("Code112","Temp Desk 2","Temp User",
        "Desk","2002","France","Wood and Metal",ptemp1,ptemp2,0,0);

(*office11).firstObjectInRoom = office11Object0;
(*office11Object0).nextObjectInRoom=office11Object1;
(*office11Object1).nextObjectInRoom=office11Object2;

// ---------------------------------------------
// Office 11b
// ---------------------------------------------

Room *office11b;
POINT office11bcoord [MAX_COORDINATES];
char  office11bcnx   [MAX_CONNEXIONS][MAX_STRINGSIZE];
POINT office11bcnxA  [MAX_CONNEXIONS];
POINT office11bcnxB  [MAX_CONNEXIONS];

office11bcoord[0].x =170;        office11bcoord[0].y =240;
office11bcoord[1].x =170;        office11bcoord[1].y =300;
office11bcoord[2].x =200;        office11bcoord[2].y =300;
office11bcoord[3].x =200;        office11bcoord[3].y =240;

strcpy(office11bcnx[0],"Hall 2");
office11bcnxA[0].x =175;  office11bcnxA[0].y =240;
office11bcnxB[0].x =180;  office11bcnxB[0].y =240;

office11b = new Room("Office 11b","Office 11b", "Office",4,
        office11bcoord,1,office11bcnx,office11bcnxA,office11bcnxB);

(*pFirstRoomInTheList).addRoom(office11b);

// ---------------------------------------------
// Office 12
// ---------------------------------------------

Room *office12;
POINT office12coord [MAX_COORDINATES];
char  office12cnx   [MAX_CONNEXIONS][MAX_STRINGSIZE];
POINT office12cnxA  [MAX_CONNEXIONS];
POINT office12cnxB  [MAX_CONNEXIONS];

office12coord[0].x =215;        office12coord[0].y =315;
office12coord[1].x =215;        office12coord[1].y =270;
office12coord[2].x =245;        office12coord[2].y =270;
office12coord[3].x =245;        office12coord[3].y =315;

strcpy(office12cnx[0],"Hall 1");
office12cnxA[0].x =215;   office12cnxA[0].y =305;
office12cnxB[0].x =215;   office12cnxB[0].y =310;

office12 = new Room("Office 12","Office 12", "Office",4,
             office12coord,1,office12cnx,office12cnxA,office12cnxB);

(*pFirstRoomInTheList).addRoom(office12);

// ---------------------------------------------
// Office 13
// ---------------------------------------------

Room *office13;
POINT office13coord [MAX_COORDINATES];
```

```
char  office13cnx   [MAX_CONNEXIONS][MAX_STRINGSIZE];
POINT office13cnxA  [MAX_CONNEXIONS];
POINT office13cnxB  [MAX_CONNEXIONS];

office13coord[0].x =215;        office13coord[0].y =270;
office13coord[1].x =215;        office13coord[1].y =240;
office13coord[2].x =245;        office13coord[2].y =240;
office13coord[3].x =245;        office13coord[3].y =270;

strcpy(office13cnx[0],"Hall 1");
office13cnxA[0].x =215;   office13cnxA[0].y =245;
office13cnxB[0].x =215;   office13cnxB[0].y =250;

office13 = new Room("Office 13","Office 13", "Office",4,
             office13coord,1,office13cnx,office13cnxA,office13cnxB);

(*pFirstRoomInTheList).addRoom(office13);

// ---------------------------------------------
// Office 15
// ---------------------------------------------

Room *office15;
POINT office15coord [MAX_COORDINATES];
char  office15cnx   [MAX_CONNEXIONS][MAX_STRINGSIZE];
POINT office15cnxA  [MAX_CONNEXIONS];
POINT office15cnxB  [MAX_CONNEXIONS];

office15coord[0].x =190;        office15coord[0].y =210;
office15coord[1].x =190;        office15coord[1].y =180;
office15coord[2].x =245;        office15coord[2].y =180;
office15coord[3].x =245;        office15coord[3].y =210;

strcpy(office15cnx[0],"Hall 1");
office15cnxA[0].x =205;   office15cnxA[0].y =210;
office15cnxB[0].x =210;   office15cnxB[0].y =210;

strcpy(office15cnx[1],"Hall 2");
office15cnxA[1].x =190;   office15cnxA[1].y =190;
office15cnxB[1].x =190;   office15cnxB[1].y =185;

office15 = new Room("Office 15","Office 15", "Office",4,
             office15coord,2,office15cnx,office15cnxA,office15cnxB);

(*pFirstRoomInTheList).addRoom(office15);

// ---------------------------------------------
// Office 16
// ---------------------------------------------

Room *office16;
POINT office16coord [MAX_COORDINATES];
char  office16cnx   [MAX_CONNEXIONS][MAX_STRINGSIZE];
POINT office16cnxA  [MAX_CONNEXIONS];
POINT office16cnxB  [MAX_CONNEXIONS];

office16coord[0].x =190;        office16coord[0].y =180;
office16coord[1].x =190;        office16coord[1].y =150;
office16coord[2].x =245;        office16coord[2].y =150;
office16coord[3].x =245;        office16coord[3].y =180;

strcpy(office16cnx[0],"Hall 1");
office16cnxA[0].x =190;   office16cnxA[0].y =170;
office16cnxB[0].x =190;   office16cnxB[0].y =175;
```

```
office16 = new Room("Office 16","Office 16", "Office",4,
            office16coord,1,office16cnx,office16cnxA,office16cnxB);

(*pFirstRoomInTheList).addRoom(office16);

// ---------------------------------------------
// Office 17
// ---------------------------------------------

Room *office17;
POINT office17coord [MAX_COORDINATES];
char  office17cnx   [MAX_CONNEXIONS][MAX_STRINGSIZE];
POINT office17cnxA  [MAX_CONNEXIONS];
POINT office17cnxB  [MAX_CONNEXIONS];

office17coord[0].x =190;        office17coord[0].y =150;
office17coord[1].x =190;        office17coord[1].y =120;
office17coord[2].x =245;        office17coord[2].y =120;
office17coord[3].x =245;        office17coord[3].y =150;

strcpy(office17cnx[0],"Hall 2");
office17cnxA[0].x =190;   office17cnxA[0].y =125;
office17cnxB[0].x =190;   office17cnxB[0].y =130;

office17 = new Room("Office 17","Office 17", "Office",4,
            office17coord,1,office17cnx,office17cnxA,office17cnxB);

(*pFirstRoomInTheList).addRoom(office17);

// ---------------------------------------------
// Office 18
// ---------------------------------------------

Room *office18;
POINT office18coord [MAX_COORDINATES];
char  office18cnx   [MAX_CONNEXIONS][MAX_STRINGSIZE];
POINT office18cnxA  [MAX_CONNEXIONS];
POINT office18cnxB  [MAX_CONNEXIONS];

office18coord[0].x =190;        office18coord[0].y =120;
office18coord[1].x =190;        office18coord[1].y =90;
office18coord[2].x =245;        office18coord[2].y =90;
office18coord[3].x =245;        office18coord[3].y =120;

strcpy(office18cnx[0],"Hall 2");
office18cnxA[0].x =190;   office18cnxA[0].y =110;
office18cnxB[0].x =190;   office18cnxB[0].y =115;

office18 = new Room("Office 18","Office 18", "Office",4,
            office18coord,1,office18cnx,office18cnxA,office18cnxB);

(*pFirstRoomInTheList).addRoom(office18);

ArtObject *office18Object0;
ArtObject *office18Object1;

ptemp1.x=200;ptemp1.y =90;
ptemp2.x=230;ptemp2.y =100;
office18Object0 = new ArtObject("Code180","Wooden Desk of User X",
            "User X","Long Desk","2002","Italy",
            "Wood",ptemp1,ptemp2,0,0);

ptemp1.x=200;ptemp1.y =110;
ptemp2.x=230;ptemp2.y =120;
office18Object1 = new ArtObject("Code181","Desk of User Z",
```

```
            "User Z","Long Desk","2002","Italy",
            "Wood and Metal",ptemp1,ptemp2,0,0);

(*office18).firstObjectInRoom = office18Object0;
(*office18Object0).nextObjectInRoom=office18Object1;

// ----------------------------------------------
// Office 19
// ----------------------------------------------

Room *office19;
POINT office19coord [MAX_COORDINATES];
char  office19cnx   [MAX_CONNEXIONS][MAX_STRINGSIZE];
POINT office19cnxA  [MAX_CONNEXIONS];
POINT office19cnxB  [MAX_CONNEXIONS];

office19coord[0].x =190;        office19coord[0].y =90;
office19coord[1].x =190;        office19coord[1].y =60;
office19coord[2].x =245;        office19coord[2].y =60;
office19coord[3].x =245;        office19coord[3].y =90;

strcpy(office19cnx[0],"Hall 2");
office19cnxA[0].x =190;   office19cnxA[0].y =65;
office19cnxB[0].x =190;   office19cnxB[0].y =70;

office19 = new Room("Office 19","Office 19", "Office",4,
            office19coord,1,office19cnx,office19cnxA,office19cnxB);

(*pFirstRoomInTheList).addRoom(office19);

ArtObject *office19Object0;
ArtObject *office19Object1;

ptemp1.x=200;ptemp1.y =60;
ptemp2.x=230;ptemp2.y =70;
office19Object0 = new ArtObject("Code190","Desk of User X",
                  "User X","Long Desk","2002","Italy",
                  "Wood and Metal",ptemp1,ptemp2,0,0);

ptemp1.x=200;ptemp1.y =80;
ptemp2.x=230;ptemp2.y =90;
office19Object1 = new ArtObject("Code191","Desk of User Y",
                  "User Y","Long Desk","2002","Italy",
                  "Wood and Metal",ptemp1,ptemp2,0,0);

(*office19).firstObjectInRoom = office19Object0;
(*office19Object0).nextObjectInRoom=office19Object1;

// ----------------------------------------------
// Office 20
// ----------------------------------------------

Room *office20;
POINT office20coord [MAX_COORDINATES];
char  office20cnx   [MAX_CONNEXIONS][MAX_STRINGSIZE];
POINT office20cnxA  [MAX_CONNEXIONS];
POINT office20cnxB  [MAX_CONNEXIONS];

office20coord[0].x =190;        office20coord[0].y =60;
office20coord[1].x =190;        office20coord[1].y =30;
office20coord[2].x =245;        office20coord[2].y =30;
office20coord[3].x =245;        office20coord[3].y =60;

strcpy(office20cnx[0],"Hall 2");
office20cnxA[0].x =190;   office20cnxA[0].y =50;
```

```
office20cnxB[0].x =190;    office20cnxB[0].y =55;

office20 = new Room("Office 20","Office 20", "Office",4,
             office20coord,1,office20cnx,office20cnxA,office20cnxB);

(*pFirstRoomInTheList).addRoom(office20);

//art objects definition for the test
//BEWARE !!! the value for myImage Code must be the same than the one
//defined for the Bitmaps ID in "guide-ravi.h" !!!

ArtObject *office20Object0;
ArtObject *office20Object1;
ArtObject *office20Object2;
ArtObject *office20Object3;

ptemp1.x=230;ptemp1.y =46;
ptemp2.x=245;ptemp2.y =55;
office20Object0 = new ArtObject("Code200","Barbara's desk","Barbara",
                    "Desk","2002","Italy",
                    "Wood and Metal",ptemp1,ptemp2,0,0);

ptemp1.x=225;ptemp1.y =35;
ptemp2.x=240;ptemp2.y =44;
office20Object1 = new ArtObject("Code201","Francesco's desk",
                    "Francesco","Desk","2002","Italy",
                    "Wood and Metal",ptemp1,ptemp2,0,0);

ptemp1.x=195;ptemp1.y =35;
ptemp2.x=210;ptemp2.y =45;
office20Object2 = new ArtObject("Code202","Ravi's desk",
                    "Ravi","Desk","2002","Belgium",
                    "Wood and Metal",ptemp1,ptemp2,0,0);

ptemp1.x=200;ptemp1.y=50;
ptemp2.x=225;ptemp2.y=60;
office20Object3 = new ArtObject("Code203","Vincenzo's desk",
                    "Vincenzo","Double Desk","2002","Italy",
                    "Wood and Metal",ptemp1,ptemp2,0,0);

(*office20).firstObjectInRoom = office20Object0;
(*office20Object0).nextObjectInRoom=office20Object1;
(*office20Object1).nextObjectInRoom=office20Object2;
(*office20Object2).nextObjectInRoom=office20Object3;


// ---------------------------------------------
// Office 21
// ---------------------------------------------

Room *office21;
POINT office21coord [MAX_COORDINATES];
char  office21cnx   [MAX_CONNEXIONS][MAX_STRINGSIZE];
POINT office21cnxA  [MAX_CONNEXIONS];
POINT office21cnxB  [MAX_CONNEXIONS];

office21coord[0].x =190;        office21coord[0].y =30;
office21coord[1].x =190;        office21coord[1].y =0;
office21coord[2].x =245;        office21coord[2].y =0;
office21coord[3].x =245;        office21coord[3].y =30;

strcpy(office21cnx[0],"Hall 2");
office21cnxA[0].x =190;   office21cnxA[0].y =10;
office21cnxB[0].x =190;   office21cnxB[0].y =15;
```

```
office21 = new Room("Office 21","Office 21", "Office",4,
            office21coord,1,office21cnx,office21cnxA,office21cnxB);

(*pFirstRoomInTheList).addRoom(office21);

// ---------------------------------------------
// officeIAT
// ---------------------------------------------

Room *officeIAT;
POINT officeIATcoord [MAX_COORDINATES];
char  officeIATcnx   [MAX_CONNEXIONS][MAX_STRINGSIZE];
POINT officeIATcnxA  [MAX_CONNEXIONS];
POINT officeIATcnxB  [MAX_CONNEXIONS];

officeIATcoord[0].x =35;  officeIATcoord[0].y =240;
officeIATcoord[1].x =35;  officeIATcoord[1].y =155;
officeIATcoord[2].x =170; officeIATcoord[2].y =155;
officeIATcoord[3].x =170; officeIATcoord[3].y =185;
officeIATcoord[4].x =150; officeIATcoord[4].y =195;
officeIATcoord[5].x =150; officeIATcoord[5].y =240;

strcpy(officeIATcnx[0],"Hall 2");
officeIATcnxA[0].x =65;   officeIATcnxA[0].y =155;
officeIATcnxB[0].x =75;   officeIATcnxB[0].y =155;

officeIAT = new Room("Office IAT","Office IAT", "Office",6,
            officeIATcoord,1,officeIATcnx,officeIATcnxA,officeIATcnxB);

(*pFirstRoomInTheList).addRoom(officeIAT);

// ---------------------------------------------
// officeIEI
// ---------------------------------------------

Room *officeIEI;
POINT officeIEIcoord [MAX_COORDINATES];
char  officeIEIcnx   [MAX_CONNEXIONS][MAX_STRINGSIZE];
POINT officeIEIcnxA  [MAX_CONNEXIONS];
POINT officeIEIcnxB  [MAX_CONNEXIONS];

officeIEIcoord[0].x =35;  officeIEIcoord[0].y =145;
officeIEIcoord[1].x =35;  officeIEIcoord[1].y =95;
officeIEIcoord[2].x =135; officeIEIcoord[2].y =95;
officeIEIcoord[3].x =135; officeIEIcoord[3].y =145;

strcpy(officeIEIcnx[0],"Hall 2");
officeIEIcnxA[0].x =65;   officeIEIcnxA[0].y =145;
officeIEIcnxB[0].x =75;   officeIEIcnxB[0].y =145;

officeIEI = new Room("Office IEI","Office IEI", "Office",4,
            officeIEIcoord,1,officeIEIcnx,officeIEIcnxA,officeIEIcnxB);

(*pFirstRoomInTheList).addRoom(officeIEI);

// ---------------------------------------------
// officeCED1
// ---------------------------------------------

Room *officeCED1;
POINT officeCED1coord [MAX_COORDINATES];
char  officeCED1cnx   [MAX_CONNEXIONS][MAX_STRINGSIZE];
POINT officeCED1cnxA  [MAX_CONNEXIONS];
POINT officeCED1cnxB  [MAX_CONNEXIONS];
```

```
officeCED1coord[0].x =135;officeCED1coord[0].y =145;
officeCED1coord[1].x =135;officeCED1coord[1].y =110;
officeCED1coord[2].x =175;officeCED1coord[2].y =110;
officeCED1coord[3].x =175;officeCED1coord[3].y =145;

strcpy(officeCED1cnx[0],"Hall 2");
officeCED1cnxA[0].x =165; officeCED1cnxA[0].y =145;
officeCED1cnxB[0].x =170; officeCED1cnxB[0].y =145;

officeCED1 = new Room("Office CED1","Office CED1", "Office",4,
      officeCED1coord,1,officeCED1cnx,officeCED1cnxA,officeCED1cnxB);

(*pFirstRoomInTheList).addRoom(officeCED1);

// ---------------------------------------------
// officeCED2
// ---------------------------------------------

Room *officeCED2;
POINT officeCED2coord [MAX_COORDINATES];
char  officeCED2cnx   [MAX_CONNEXIONS][MAX_STRINGSIZE];
POINT officeCED2cnxA  [MAX_CONNEXIONS];
POINT officeCED2cnxB  [MAX_CONNEXIONS];

officeCED2coord[0].x =175;officeCED2coord[0].y =110;
officeCED2coord[1].x =135;officeCED2coord[1].y =110;
officeCED2coord[2].x =135;officeCED2coord[2].y =95;
officeCED2coord[3].x =80; officeCED2coord[3].y =95;
officeCED2coord[4].x =80; officeCED2coord[4].y =60;
officeCED2coord[5].x =175;officeCED2coord[5].y =60;

strcpy(officeCED2cnx[0],"Hall 2");
officeCED2cnxA[0].x =175; officeCED2cnxA[0].y =70;
officeCED2cnxB[0].x =175; officeCED2cnxB[0].y =75;

officeCED2 = new Room("Office CED2","Office CED2", "Office",6,
      officeCED2coord,1,officeCED2cnx,officeCED2cnxA,officeCED2cnxB);

(*pFirstRoomInTheList).addRoom(officeCED2);

// ---------------------------------------------
// officeCED3
// ---------------------------------------------

Room *officeCED3;
POINT officeCED3coord [MAX_COORDINATES];
char  officeCED3cnx   [MAX_CONNEXIONS][MAX_STRINGSIZE];
POINT officeCED3cnxA  [MAX_CONNEXIONS];
POINT officeCED3cnxB  [MAX_CONNEXIONS];

officeCED3coord[0].x =80; officeCED3coord[0].y =95;
officeCED3coord[1].x =35; officeCED3coord[1].y =95;
officeCED3coord[2].x =35; officeCED3coord[2].y =30;
officeCED3coord[3].x =45; officeCED3coord[3].y =30;
officeCED3coord[4].x =45; officeCED3coord[4].y =20;
officeCED3coord[5].x =35; officeCED3coord[5].y =20;
officeCED3coord[6].x =35; officeCED3coord[6].y =0;
officeCED3coord[7].x =145;officeCED3coord[7].y =0;
officeCED3coord[8].x =145;officeCED3coord[8].y =25;
officeCED3coord[9].x =175;officeCED3coord[9].y =25;
officeCED3coord[10].x=175;officeCED3coord[10].y=60;
officeCED3coord[11].x=80; officeCED3coord[11].y=60;

strcpy(officeCED3cnx[0],"Hall 2");
officeCED3cnxA[0].x =175; officeCED3cnxA[0].y =30;
```

```
    officeCED3cnxB[0].x =175; officeCED3cnxB[0].y =35;

    strcpy(officeCED3cnx[1],"Hall 3");
    officeCED3cnxA[1].x =45;  officeCED3cnxA[1].y =22;
    officeCED3cnxB[1].x =45;  officeCED3cnxB[1].y =28;

    officeCED3 = new Room("Office CED3","Office CED3", "Office",12,
          officeCED3coord,2,officeCED3cnx,officeCED3cnxA,officeCED3cnxB);

    (*pFirstRoomInTheList).addRoom(officeCED3);

    // --------------------------------------------
    // officeCED4
    // --------------------------------------------

    Room *officeCED4;
    POINT officeCED4coord [MAX_COORDINATES];
    char  officeCED4cnx   [MAX_CONNEXIONS][MAX_STRINGSIZE];
    POINT officeCED4cnxA  [MAX_CONNEXIONS];
    POINT officeCED4cnxB  [MAX_CONNEXIONS];

    officeCED4coord[0].x =175;officeCED4coord[0].y =25;
    officeCED4coord[1].x =145;officeCED4coord[1].y =25;
    officeCED4coord[2].x =145;officeCED4coord[2].y =0;
    officeCED4coord[3].x =175;officeCED4coord[3].y =0;

    strcpy(officeCED4cnx[0],"Hall 2");
    officeCED4cnxA[0].x =175; officeCED4cnxA[0].y =5;
    officeCED4cnxB[0].x =175; officeCED4cnxB[0].y =15;

    officeCED4 = new Room("Office CED4","Office CED4", "Office",4,
          officeCED4coord,1,officeCED4cnx,officeCED4cnxA,officeCED4cnxB);

    (*pFirstRoomInTheList).addRoom(officeCED4);

    return TRUE;
}
```

E.8 attributelist.h

```
// ================================================================
// HEADER FILE for attributelist.cpp
// ================================================================

#include "attribute.h"
// "attribute.h" includes "roomlist.h"
// "roomlist.h"      includes "room.h"
// "room.h"          includes "artobject.h"

class AttributeList
{
public :
       Attribute * pName;
       Attribute * pAuthor;
       Attribute * pStyle;
       Attribute * pPeriod;
       Attribute * pCountry;
       Attribute * pMaterial;

       //Default Constructor
       AttributeList(void);

       //Personnalized Constructor
       AttributeList(RoomList * pFirstRoomInList);

       //Default Destructor
       ~AttributeList(void);
};
```

E.9 attributelist.cpp

```
// ================================================================
// attributelist core file
// ================================================================

#include <windows.h>      // for all the windows stuff
#include <stdio.h>        // for string manipulation

#include "attributelist.h"

//Default Constructor
AttributeList::AttributeList(void)
{
	pName		= new Attribute ("");
	pAuthor		= new Attribute ("");
	pStyle		= new Attribute ("");
	pPeriod		= new Attribute ("");
	pCountry	= new Attribute ("");
	pMaterial	= new Attribute ("");
}

AttributeList::AttributeList(RoomList * pFirstRoomInList)
//PRE : must be at leat one room in list
{
	BOOL newAttributeValue =FALSE;
	ArtObject *pTempArtObject;
	RoomList *pTempRoomList;
	pTempRoomList=pFirstRoomInList;

	pName		= new Attribute ("");
	pAuthor		= new Attribute ("");
	pStyle		= new Attribute ("");
	pPeriod		= new Attribute ("");
	pCountry	= new Attribute ("");
	pMaterial	= new Attribute ("");

	Attribute * pTempAttr0=pName;
	Attribute * pTempAttr1;

	while (pTempRoomList!=NULL)
	{
		pTempArtObject=(*(*pTempRoomList).roomToSee).firstObjectInRoom;
		while (pTempArtObject!=NULL)
		{
			//see if actual art object has an unlisted name
			pTempAttr0=pName;
			while (pTempAttr0!=NULL)
			{
				if (strcmp((* pTempAttr0).value,
					(*pTempArtObject).name)==0)
				{
					(*pTempAttr0).addRoom(
						(*pTempRoomList).roomToSee);
					newAttributeValue=FALSE;
					break;
				}
				else
				{
					newAttributeValue=TRUE;
					pTempAttr1=pTempAttr0;
					pTempAttr0=(*pTempAttr0).nextAttribute;
				}
			}
```

```
if (newAttributeValue)
{
	Attribute * pTempAttr2
	  =new Attribute((*pTempArtObject).name,
	                   (*pTempRoomList).roomToSee);
	(*pTempAttr1).nextAttribute=pTempAttr2;
}

//see if actual art object has an unlisted author
pTempAttr0=pAuthor;
while (pTempAttr0!=NULL)
{
	if (strcmp((* pTempAttr0).value,
		(*pTempArtObject).author)==0)
	{
		(*pTempAttr0).addRoom(
		        (*pTempRoomList).roomToSee);
		newAttributeValue=FALSE;
		break;
	}
	else
	{
		newAttributeValue=TRUE;
		pTempAttr1=pTempAttr0;
		pTempAttr0=(*pTempAttr0).nextAttribute;
	}
}
if (newAttributeValue)
{
	Attribute * pTempAttr2
	  =new Attribute((*pTempArtObject).author,
	                   (*pTempRoomList).roomToSee);
	(*pTempAttr1).nextAttribute=pTempAttr2;
}

//see if actual art object has an unlisted style
pTempAttr0=pStyle;
while (pTempAttr0!=NULL)
{
	if (strcmp((* pTempAttr0).value,
		      (*pTempArtObject).style)==0)
	{
		(*pTempAttr0).addRoom(
		      (*pTempRoomList).roomToSee);
		newAttributeValue=FALSE;
		break;
	}
	else
	{
		newAttributeValue=TRUE;
		pTempAttr1=pTempAttr0;
		pTempAttr0=(*pTempAttr0).nextAttribute;
	}
}
if (newAttributeValue)
{
	Attribute * pTempAttr2
	  =new Attribute((*pTempArtObject).style,
	                   (*pTempRoomList).roomToSee);
	(*pTempAttr1).nextAttribute=pTempAttr2;
}

//see if actual art object has an unlisted period
pTempAttr0=pPeriod;
while (pTempAttr0!=NULL)
```

```
{
        if (strcmp((* pTempAttr0).value,
                    (*pTempArtObject).period)==0)
        {
                (*pTempAttr0).addRoom(
                    (*pTempRoomList).roomToSee);
                newAttributeValue=FALSE;
                break;
        }
        else
        {
                newAttributeValue=TRUE;
                pTempAttr1=pTempAttr0;
                pTempAttr0=(*pTempAttr0).nextAttribute;
        }
}
if (newAttributeValue)
{
        Attribute * pTempAttr2
          =new Attribute((*pTempArtObject).period,
                        (*pTempRoomList).roomToSee);
        (*pTempAttr1).nextAttribute=pTempAttr2;
}

//see if actual art object has an unlisted country
pTempAttr0=pCountry;
while (pTempAttr0!=NULL)
{
        if (strcmp((* pTempAttr0).value,
                    (*pTempArtObject).country)==0)
        {
                (*pTempAttr0).addRoom(
                    (*pTempRoomList).roomToSee);
                newAttributeValue=FALSE;
                break;
        }
        else
        {
                newAttributeValue=TRUE;
                pTempAttr1=pTempAttr0;
                pTempAttr0=(*pTempAttr0).nextAttribute;
        }
}
if (newAttributeValue)
{
        Attribute * pTempAttr2
          =new Attribute((*pTempArtObject).country,
                        (*pTempRoomList).roomToSee);
        (*pTempAttr1).nextAttribute=pTempAttr2;
}

//see if actual art object has an unlisted material
pTempAttr0=pMaterial;
while (pTempAttr0!=NULL)
{
        if (strcmp((* pTempAttr0).value,
                (*pTempArtObject).material)==0)
        {
                (*pTempAttr0).addRoom(
                    (*pTempRoomList).roomToSee);
                newAttributeValue=FALSE;
                break;
        }
        else
        {
```

```
                        newAttributeValue=TRUE;
                        pTempAttr1=pTempAttr0;
                        pTempAttr0=(*pTempAttr0).nextAttribute;
                    }
                }
                if (newAttributeValue)
                {
                    Attribute * pTempAttr2
                      =new Attribute((*pTempArtObject).material,
                                    (*pTempRoomList).roomToSee);
                    (*pTempAttr1).nextAttribute=pTempAttr2;
                }
                pTempArtObject=(*pTempArtObject).nextObjectInRoom;
            }
            pTempRoomList=(*pTempRoomList).nextRoomInList;
    }
}


//Default Destructor
AttributeList::~AttributeList(void)
{
    delete pName;
    delete pAuthor;
    delete pStyle;
    delete pPeriod;
    delete pCountry;
    delete pMaterial;
}
```

E.10 attribute.h

```
// ============================================================
// HEADER FILE for attribute.cpp
// ============================================================

#include "roomlist.h"
// "roomlist.h"      includes "room.h"
// "room.h"          includes "artobject.h"

class Attribute
{
public :
        char            value[MAX_STRINGSIZE];
        RoomList        * firstCorrespondingRoom;
        Attribute       * nextAttribute;

        //Default Constructor
        Attribute(void);

        //Personnalized Constructor 1
        Attribute(char myValue[MAX_STRINGSIZE]);

        //Personnalized Constructor 2
        Attribute(char myValue[MAX_STRINGSIZE], Room * pRoom);

        //Default Destructor
        ~Attribute(void);

        // Adds a room to the RoomList corresponding to the attribute's value
        // PRE : the check on the value has already been made
        BOOL addRoom(Room * pRoom);
};
```

E.11 attribute.cpp

```
// ==================================================================
// attributelist core file
// ==================================================================

#include <windows.h>       // for all the windows stuff
#include <stdio.h>         // for string manipulation

#include "attribute.h"


//Default Constructor
Attribute::Attribute()
{
       strcpy(value,"");
       firstCorrespondingRoom=new RoomList();
       nextAttribute=NULL;
}

//Personnalized Constructor 1
Attribute::Attribute(char myValue[MAX_STRINGSIZE])
{
       strcpy(value,myValue);
       firstCorrespondingRoom=new RoomList();
       nextAttribute=NULL;
}

//Personnalized Constructor 2
Attribute::Attribute(char myValue[MAX_STRINGSIZE], Room * pRoom)
{
       strcpy(value,myValue);
       firstCorrespondingRoom=new RoomList(pRoom);
       nextAttribute=NULL;
}

//Default Destructor
Attribute::~Attribute(void)
{
       strcpy(value,"");
       delete firstCorrespondingRoom;
       nextAttribute=NULL;
}

// Adds a room to the RoomList corresponding to the attribute's value
// PRE : the check on the value has already been made
BOOL Attribute::addRoom(Room * pRoom)
{
       return (*firstCorrespondingRoom).addRoom(pRoom);
}
```

E.12 roomlist.h

```
// ================================================================
// HEADER FILE for roomlist.cpp
// ================================================================

#include "room.h"
// "room.h"  includes "artobject.h"

class RoomList
{
public :
       Room          * roomToSee;
       RoomList      * nextRoomInList;

       //Default Constructor
       RoomList(void);

       //Personnalized Constructor
       RoomList(Room * pRoom);

       //Default Destructor
       ~RoomList(void);

       BOOL copy(RoomList  * pRoomListToCopy);
       //copies a room list in another
       //returns TRUE if the destination list is initially empty
       //             and if the copy was sucessfull
       //returns FALSE otherwise

       BOOL addRoom(Room * pRoom);
       // adds the Room pointed by pRoom in the list beginning by "this"
       // if it isn't already listed
       // returns TRUE  if the room has been added
       // returns FALSE if the room was already listed

       POINT getMinCoordinatesInList(void);
       // PRE : this.RoomToSee!=NULL && this.RoomToSee.numberOfCoordinates>0
       // returns the minimal x and the minimal y from the list containing
       // this room and the ones following in nextRoomInRoomList

       POINT getMaxCoordinatesInList(void);
       // PRE : this.RoomToSee!=NULL && this.RoomToSee.numberOfCoordinates>0
       // returns the maximal x and the maximal y from the list containing
       // this room and the ones following in nextRoomInRoomList
};
```

E.13 roomlist.cpp

```
// ==================================================================
// RoomList core file
// ==================================================================

#include <windows.h>       // for all the windows stuff

#include "roomlist.h"

//Default Constructor
RoomList::RoomList(void)
{
	roomToSee=NULL;
	nextRoomInList=NULL;
}

//Personnalized Constructor
RoomList::RoomList(Room * pRoom)
{
	roomToSee=pRoom;
	nextRoomInList=NULL;
}


//Default Destructor
RoomList::~RoomList(void)
{
	if (nextRoomInList!=NULL)
	{
		delete nextRoomInList;
		//nextRoomInList=NULL;
	}

	if (roomToSee!=NULL)
	{
		delete roomToSee;
		//roomToSee=NULL;
	}

}

BOOL RoomList::copy(RoomList     * pRoomListToCopy)
//copies a room list in a another
//returns TRUE if the destination list is initially empty
//             and if the copy was sucessfull
//returns FALSE otherwise
{
	Room         * pTempRoom;
	ArtObject    * pTempArtObject;
	ArtObject    * pFirstArtObject;
	ArtObject    * pCurrentArtObject;
	ArtObject    * pLastArtObject;
	RoomList     * pTempRoomList = pRoomListToCopy;

	if (roomToSee!=NULL && nextRoomInList!=NULL)
	{
		return FALSE;
	}

	if ((* pTempRoomList).roomToSee!=NULL)
	{
		pTempRoom=new Room(
			(*(*pTempRoomList).roomToSee).code,
```

```
                    (*(*pTempRoomList).roomToSee).name,
                    (*(*pTempRoomList).roomToSee).type,
                    (*(*pTempRoomList).roomToSee).numberOfCoordinates,
                    (*(*pTempRoomList).roomToSee).coordinates,
                    (*(*pTempRoomList).roomToSee).numberOfAdjacentRooms,
                    (*(*pTempRoomList).roomToSee).adjacentRoomsCode,
                    (*(*pTempRoomList).roomToSee).connexionsA,
                    (*(*pTempRoomList).roomToSee).connexionsB
                    );

        pFirstArtObject=NULL;
        pTempArtObject=(*(* pTempRoomList).roomToSee).firstObjectInRoom;

        while (pTempArtObject!=NULL)
        {
                //create art object
                if(pFirstArtObject==NULL)
                {
                        pFirstArtObject=new ArtObject(
                                (* pTempArtObject).code,
                                (* pTempArtObject).name,
                                (* pTempArtObject).author,
                                (* pTempArtObject).style,
                                (* pTempArtObject).period,
                                (* pTempArtObject).country,
                                (* pTempArtObject).material,
                                (* pTempArtObject).positionA,
                                (* pTempArtObject).positionB,
                                0,0);
                        pLastArtObject=pFirstArtObject;
                }
                else
                {
                        pCurrentArtObject=new ArtObject(
                                (* pTempArtObject).code,
                                (* pTempArtObject).name,
                                (* pTempArtObject).author,
                                (* pTempArtObject).style,
                                (* pTempArtObject).period,
                                (* pTempArtObject).country,
                                (* pTempArtObject).material,
                                (* pTempArtObject).positionA,
                                (* pTempArtObject).positionB,
                                0,0);

(*pLastArtObject).nextObjectInRoom=pCurrentArtObject;
                        pLastArtObject=pCurrentArtObject;
                }
                pTempArtObject=(*pTempArtObject).nextObjectInRoom;
        }
        (*pTempRoom).firstObjectInRoom=pFirstArtObject;

        //add room to roomlist
        addRoom(pTempRoom);
        if ((* pTempRoomList).nextRoomInList!=NULL)
        {
                pTempRoomList=(* pTempRoomList).nextRoomInList;
        }

        //re-initialise artobject pointers
        pFirstArtObject=NULL;
        pLastArtObject=NULL;
        pCurrentArtObject=NULL;
}
```

```
while (pTempRoomList!=NULL)
{
	if ((* pTempRoomList).roomToSee!=NULL)
	{
		pTempRoom=new Room(
			(*(*pTempRoomList).roomToSee).code,
			(*(*pTempRoomList).roomToSee).name,
			(*(*pTempRoomList).roomToSee).type,
			(*(*pTempRoomList).roomToSee).numberOfCoordinates,
			(*(*pTempRoomList).roomToSee).coordinates,
			(*(*pTempRoomList).roomToSee).numberOfAdjacentRooms,
			(*(*pTempRoomList).roomToSee).adjacentRoomsCode,
			(*(*pTempRoomList).roomToSee).connexionsA,
			(*(*pTempRoomList).roomToSee).connexionsB
			);

		pFirstArtObject=NULL;
		pTempArtObject
		 =(*(*pTempRoomList).roomToSee).firstObjectInRoom;

		while (pTempArtObject!=NULL)
		{
			//create art object
			if(pFirstArtObject==NULL)
			{
				pFirstArtObject=new ArtObject(
					(* pTempArtObject).code,
					(* pTempArtObject).name,
					(* pTempArtObject).author,
					(* pTempArtObject).style,
					(* pTempArtObject).period,
					(* pTempArtObject).country,
					(* pTempArtObject).material,
					(* pTempArtObject).positionA,
					(* pTempArtObject).positionB,
					0,0);
				pLastArtObject=pFirstArtObject;
			}
			else
			{
				pCurrentArtObject=new ArtObject(
					(* pTempArtObject).code,
					(* pTempArtObject).name,
					(* pTempArtObject).author,
					(* pTempArtObject).style,
					(* pTempArtObject).period,
					(* pTempArtObject).country,
					(* pTempArtObject).material,
					(* pTempArtObject).positionA,
					(* pTempArtObject).positionB,
					0,0);
				(*pLastArtObject).nextObjectInRoom
				  =pCurrentArtObject;
				pLastArtObject=pCurrentArtObject;
			}
			pTempArtObject=(*pTempArtObject).nextObjectInRoom;
		}
		(*pTempRoom).firstObjectInRoom=pFirstArtObject;

		//add room to roomlist
		addRoom(pTempRoom);

		//re-initialise artobject pointers
		pFirstArtObject=NULL;
		pLastArtObject=NULL;
```

```
                        pCurrentArtObject=NULL;
                }

                pTempRoomList=(* pTempRoomList).nextRoomInList;
        }

        return TRUE;
}

BOOL RoomList::addRoom(Room * pRoom)
// adds the Room pointed by pRoom in the list beginning by "this"
// if it isn't already listed
// returns TRUE  if the room has been added
// returns FALSE if the room was already listed
{
        if (roomToSee==NULL)
        {
                roomToSee=pRoom;
                return TRUE;
        }

        if (roomToSee==pRoom)
                return FALSE;
        else
        {
                RoomList * pTempRoomListPrev;
                RoomList * pTempRoomList;
                pTempRoomListPrev=this;
                pTempRoomList=nextRoomInList;
                while (pTempRoomList!=NULL)
                {
                        if ((*pTempRoomList).roomToSee==pRoom)
                                return FALSE;
                        else
                        {
                                pTempRoomListPrev=pTempRoomList;
                                pTempRoomList=(*pTempRoomList).nextRoomInList;
                        }
                }
                RoomList * pNewRoomList = new RoomList(pRoom);
                (*pTempRoomListPrev).nextRoomInList=pNewRoomList;
                return TRUE;
        }
}

POINT RoomList::getMinCoordinatesInList(void)
// PRE : this.RoomToSee!=NULL && this.RoomToSee.numberOfCoordinates>0
// returns the minimal x and the minimal y from the list containing
// this room and the ones following in nextRoomInList
{
        POINT resPointTEMP;

        POINT resPoint=(*roomToSee).getMinCoordinates();
        RoomList *pTempRoomList;
        pTempRoomList=nextRoomInList;

        while (pTempRoomList!=NULL)
        {

        resPointTEMP.x=((*(*pTempRoomList).roomToSee).getMinCoordinates()).x;

        resPointTEMP.y=((*(*pTempRoomList).roomToSee).getMinCoordinates()).y;
                if (resPoint.x>resPointTEMP.x) resPoint.x=resPointTEMP.x;
                if (resPoint.y>resPointTEMP.y) resPoint.y=resPointTEMP.y;
```

```
            pTempRoomList=(*pTempRoomList).nextRoomInList;
        }

        return resPoint;
}

POINT RoomList::getMaxCoordinatesInList(void)
//PRE : numberOfCoordinates>0
// returns the maximal x and the maximal y from the list containing
// this room and the ones following in nextRoomInList
{
        POINT resPointTEMP;

        POINT resPoint=(*roomToSee).getMaxCoordinates();
        RoomList *pTempRoomList;
        pTempRoomList=nextRoomInList;

        while (pTempRoomList!=NULL)
        {

        resPointTEMP.x=((*(*pTempRoomList).roomToSee).getMaxCoordinates()).x;

        resPointTEMP.y=((*(*pTempRoomList).roomToSee).getMaxCoordinates()).y;
            if (resPoint.x<resPointTEMP.x) resPoint.x=resPointTEMP.x;
            if (resPoint.y<resPointTEMP.y) resPoint.y=resPointTEMP.y;

            pTempRoomList=(*pTempRoomList).nextRoomInList;
        }

        return resPoint;
}
```

E.14 room.h

```
// ================================================================
// HEADER FILE for room.cpp
// ================================================================

#include "artobject.h"

class Room
{
public :
	//code name
	char	code[MAX_STRINGSIZE];
	//room name
	char	name[MAX_STRINGSIZE];
	//room type
	char	type[MAX_STRINGSIZE];
	int		numberOfCoordinates;
	//the room is seen as a concave polygone : each point of the table is an
	// extremal point following the other
	POINT	coordinates[MAX_COORDINATES];
	int		numberOfAdjacentRooms;
	//room code table for the adjacent rooms
	char	adjacentRoomsCode[MAX_CONNEXIONS][MAX_STRINGSIZE];
	//first coordinate of the connexion to another room
	POINT	connexionsA[MAX_CONNEXIONS];
	//second coordinate of the connexion to another room
	POINT	connexionsB[MAX_CONNEXIONS];
	// IMPORTANT : connexionsA & connexionsB must have the same
	// number of elements, which is numberOfAdjacentRooms.
	ArtObject * firstObjectInRoom;

	//Default Constructor
	Room(void);

	//Personnalized Constructor
	Room(char	myCode[MAX_STRINGSIZE],
		char	myName[MAX_STRINGSIZE],
		char	myType[MAX_STRINGSIZE],
		int		myNumberOfCoordinates,
		POINT	myCoordinates[MAX_COORDINATES],
		int		myNumberOfAdjacentRooms,
		char	myAdjacentRoomsCode[MAX_CONNEXIONS][MAX_STRINGSIZE],
		POINT	myConnexionsA[MAX_CONNEXIONS],
		POINT	myConnexionsB[MAX_CONNEXIONS]
		);

	//Default Destructor
	~Room(void);

	char	*	Room::isCnx(POINT	inputPoint,	float	lMyRoomScale,	POINT
myCanevasTopLeft,POINT myMapMinPoint);
	//returns "" if inputPoint is not a connexion to another room
	//returns the "code" of the connected room otherwise

	int Room::isInRoom(float inputPointx,float inputPointy);
	//returns 0 if inputPoint is not in the room
	//returns 1 if inputPoint is in the room

	int isToSee(void);
	//returns 0 if no objects in the room is to see
	//returns 1 if at least one object is too see

	POINT getMinCoordinates(void);
```

```
    // returns the minimal x and the minimal y of all the room coordinates

    POINT getMaxCoordinates(void);
    // returns the maximal x and the maximal y of all the room coordinates
};
```

E.15 room.cpp

```
// ==================================================================
// Room core file
// ==================================================================

#include <windows.h>      // for all the windows stuff

#include "room.h"

//Default Constructor
Room::Room(void)
{
	strcpy(code,"");
	strcpy(name,"");
	strcpy(type,"");
	numberOfCoordinates=0;
	int i;
	for (i=0;i<MAX_COORDINATES;i++)
	{
		coordinates[i].x=0;
		coordinates[i].y=0;
	}
	numberOfAdjacentRooms=0;
	for (i=0;i<MAX_CONNEXIONS;i++)
	{
		connexionsA[i].x=0;
		connexionsA[i].y=0;
		connexionsB[i].x=0;
		connexionsB[i].y=0;
	}
	firstObjectInRoom=NULL;
}

//Personnalized Constructor
Room::Room(char    myCode[MAX_STRINGSIZE],
		char   myName[MAX_STRINGSIZE],
		char   myType[MAX_STRINGSIZE],
		int    myNumberOfCoordinates,
		        //PRE : 0 < myNumberOfCoordinates < MAX_COORDINATES
		POINT  myCoordinates[MAX_COORDINATES],
		int    myNumberOfAdjacentRooms,
		        //PRE : 0 <= myNumberOfAdjacentRooms < MAX_CONNEXIONS
		char   myAdjacentRoomsCode[MAX_CONNEXIONS][MAX_STRINGSIZE],
		POINT  myConnexionsA[MAX_CONNEXIONS],
		POINT  myConnexionsB[MAX_CONNEXIONS]
	)
{
	int i;
	strcpy(code,myCode);
	strcpy(name,myName);
	strcpy(type,myType);
	numberOfCoordinates=myNumberOfCoordinates;
	for (i=0;i<MAX_COORDINATES;i++)
	{
		if (i<=myNumberOfCoordinates)
		{
			coordinates[i].x=myCoordinates[i].x;
			coordinates[i].y=myCoordinates[i].y;
		}
		else
		{
			coordinates[i].x=0;
			coordinates[i].y=0;
```

```
	}
	}
	numberOfAdjacentRooms=myNumberOfAdjacentRooms;
	for (i=0;i<myNumberOfAdjacentRooms;i++)
	{
		strcpy(adjacentRoomsCode[i],myAdjacentRoomsCode[i]);
		connexionsA[i].x=myConnexionsA[i].x;
		connexionsA[i].y=myConnexionsA[i].y;
		connexionsB[i].x=myConnexionsB[i].x;
		connexionsB[i].y=myConnexionsB[i].y;
	}
	firstObjectInRoom=NULL;
	return;
}

//Default Destructor
Room::~Room(void)
{
}

char * Room::isCnx(POINT inputPoint, float lMyRoomScale, POINT myCanevasTopLeft, POINT myMapMinPoint)
//returns "" if inputPoint is not a connexion to another room
//returns the "code" of the connected room otherwise
//HYPOTHESIS : we assume that the connexion is either vertical
//or horizontal ONLY !!!
{
	int i;
	POINT tempCnx [2];//temporary point array used to resize the connexions of a room

	for (i=0;i<numberOfAdjacentRooms;i++)
	{
		tempCnx[0].x=(long)(myCanevasTopLeft.x
				+(connexionsA[i].x-myMapMinPoint.x)*lMyRoomScale);
		tempCnx[0].y=(long)(myCanevasTopLeft.y+(connexionsA[i].y
				-myMapMinPoint.y)*lMyRoomScale);

		tempCnx[1].x=(long)(myCanevasTopLeft.x+(LONG)(connexionsB[i].x
				-myMapMinPoint.x)*lMyRoomScale);
		tempCnx[1].y=(long)(myCanevasTopLeft.y+(LONG)(connexionsB[i].y
				-myMapMinPoint.y)*lMyRoomScale);

		if( tempCnx[0].x-DEFAULT_CNX_DRW_ROOM<=inputPoint.x &&
			tempCnx[0].y-DEFAULT_CNX_DRW_ROOM<=inputPoint.y &&
			tempCnx[1].x+DEFAULT_CNX_DRW_ROOM>=inputPoint.x &&
			tempCnx[1].y+DEFAULT_CNX_DRW_ROOM>=inputPoint.y)
		{
			return adjacentRoomsCode[i];
		}
	}
	return "";
}

int Room::isInRoom(float inputPointx,float inputPointy)
//returns 0 if inputPoint is not in the room
//returns 2 if inputPoint is a part of a connexion to another room
//returns 1 if inputPoint is in the room
{
	if (inputPointx>=(float)getMinCoordinates().x &&
		inputPointy>=(float)getMinCoordinates().y &&
		inputPointx<=(float)getMaxCoordinates().x &&
		inputPointy<=(float)getMaxCoordinates().y)
	{
		float penteTable[MAX_COORDINATES];
```

```
int verticalPenteTable[MAX_COORDINATES];
float maxminx[MAX_COORDINATES],
      minmaxx[MAX_COORDINATES],
      maxminy[MAX_COORDINATES],
      minmaxy[MAX_COORDINATES];
float iPente=0;
float iPenteTemp;
float helpPointx,helpPointy;
float intersectionPointx,intersectionPointy;
int numberOfIntersections=0;
int i;

//get the max iPente value from the slope of
//every segment of the room
for (i=0;i<numberOfCoordinates;i++)
{
      if (i!=numberOfCoordinates-1)
      {
            if (coordinates[i+1].x!=coordinates[i].x)
            //if the segment is not vertical
            {
                  iPenteTemp=((float)(coordinates[i+1].y
                  -coordinates[i].y))
                  /((float)(coordinates[i+1].x
                  -coordinates[i].x));
                  penteTable[i]=iPenteTemp;
                  verticalPenteTable[i]=0;
            }
            else
            {
                  penteTable[i]=0;
                  verticalPenteTable[i]=1;
            }

            //set min and max x of the segment
            if (coordinates[i+1].x>coordinates[i].x)
            {
                  maxminx[i]=(float)coordinates[i].x;
                  minmaxx[i]=(float)coordinates[i+1].x;
            }
            else
            {
                  minmaxx[i]=(float)coordinates[i].x;
                  maxminx[i]=(float)coordinates[i+1].x;
            }

            //set min and max y of the segment
            if (coordinates[i+1].y>coordinates[i].y)
            {
                  maxminy[i]=(float)coordinates[i].y;
                  minmaxy[i]=(float)coordinates[i+1].y;
            }
            else
            {
                  minmaxy[i]=(float)coordinates[i].y;
                  maxminy[i]=(float)coordinates[i+1].y;
            }
      }
      else //i==numberOfCoordinates-1
      {
            if (coordinates[0].x
               !=coordinates[numberOfCoordinates-1].x)
                 //if the segment is not vertical
            {
                  iPenteTemp=((float)(coordinates[0].y
```

```
                        -coordinates[numberOfCoordinates-1].y))
                        /((float)(coordinates[0].x
                        -coordinates[numberOfCoordinates-1].x));
                    penteTable[numberOfCoordinates-1]=iPenteTemp;
                    verticalPenteTable[i]=0;
                }
                else
                {
                    penteTable[numberOfCoordinates-1]=0;
                    verticalPenteTable[i]=1;
                }

                //set min and max x of the segment
                if (coordinates[0].x
                    >coordinates[numberOfCoordinates-1].x)
                {
                    minmaxx[numberOfCoordinates-1]
                     =(float)coordinates[0].x;
                    maxminx[numberOfCoordinates-1]
                     =(float)coordinates[numberOfCoordinates-1].x;
                }
                else
                {
                    maxminx[numberOfCoordinates-1]
                     =(float)coordinates[0].x;
                    minmaxx[numberOfCoordinates-1]
                     =(float)coordinates[numberOfCoordinates-1].x;
                }

                //set min and max y of the segment
                if (coordinates[0].y
                    >coordinates[numberOfCoordinates-1].y)
                {
                    minmaxy[numberOfCoordinates-1]
                     =(float)coordinates[0].y;
                    maxminy[numberOfCoordinates-1]
                     =(float)coordinates[numberOfCoordinates-1].y;
                }
                else
                {
                    maxminy[numberOfCoordinates-1]
                     =(float)coordinates[0].y;
                    minmaxy[numberOfCoordinates-1]
                     =(float)coordinates[numberOfCoordinates-1].y;
                }
            }

            if (iPente<iPenteTemp)
            {
                iPente=iPenteTemp;
            }
            else
            {
                if (iPente<-iPenteTemp)
                        iPente=-iPenteTemp;
            }
        }

        //make iPente value different from the slope of every segment
        // of the room
        iPente=iPente+1;

        //define help point for creating a segment from inputPoint
        helpPointx=(float)(getMaxCoordinates()).x;
        helpPointy=(helpPointx-inputPointx)*iPente + inputPointy;
```

```
//test if the segment [inputPoint,helpPoint] intersects
//the segments of the room
for (i=0;i<numberOfCoordinates;i++)
{

	if (verticalPenteTable[i]==0)
	{
		//can be done because we forced iPente to be <> from
		// all the pente of the room segment

		intersectionPointx
		=((penteTable[i]*((float)coordinates[i].x)
		 -iPente*(inputPointx))-((float)coordinates[i].y
		 -inputPointy))/(penteTable[i]-iPente);

		intersectionPointy
		=penteTable[i]*(intersectionPointx
		 -(float)coordinates[i].x)+(float)coordinates[i].y;
	}

	else //verticalPenteTable[i]==1
	{
		//can be done because we forced iPente to be <> from
		// all the pente of the room segment

		intersectionPointx=(float)coordinates[i].x;
		intersectionPointy
		=iPente*(intersectionPointx
		 -(float)helpPointx)+(float)helpPointy;
	}

	// set minmaxx amd maxminx, ie values between which the
	// intersection of the 2 lines should occur to be
	// intersection of the segments
	if (helpPointx<inputPointx)
	{
		if (helpPointx>maxminx[i])
			maxminx[i]=helpPointx;
		if (inputPointx<minmaxx[i])
			minmaxx[i]=inputPointx;
	}
	else
	{
		if (inputPointx>maxminx[i])
			maxminx[i]=inputPointx;
		if (helpPointx<minmaxx[i])
			minmaxx[i]=helpPointx;
	}

	// set minmaxy amd maxminy, ie values between which the
	// intersection of the 2 lines should occur to be
	// intersection of the segments
	if (helpPointy<inputPointy)
	{
		if (helpPointy>maxminy[i])
			maxminy[i]=helpPointy;
		if (inputPointy<minmaxy[i])
			minmaxy[i]=inputPointy;
	}
	else
	{
		if (inputPointy>maxminy[i])
			maxminy[i]=inputPointy;
		if (helpPointy<minmaxy[i])
```

```
                                        minmaxy[i]=helpPointy;
                        }

                        if ((maxminx[i]<=intersectionPointx)
                                &&(intersectionPointx<=minmaxx[i])
                                &&(maxminy[i]<=intersectionPointy)
                                &&(intersectionPointy<=minmaxy[i]))
                        {
                                numberOfIntersections++;
                        }
                }

                if ((numberOfIntersections % 2)==0)
                {
                        return 0;
                }
                else
                {
                        return 1;
                }
        }
        else
        {
                return 0;
        }
}

int Room::isToSee(void)
//returns 0 if no objects in the room is to see
//returns 1 if at least one object is too see
{
        ArtObject * pTempArtObject;
        pTempArtObject = firstObjectInRoom;
        while (pTempArtObject!=NULL)
        {
                if ((*pTempArtObject).isToSee==1)
                {
                        return 1;
                }
                pTempArtObject=(*pTempArtObject).nextObjectInRoom;
        }
        return 0;
}

POINT Room::getMinCoordinates(void)
//PRE : numberOfCoordinates>0
// returns the minimal x and the minimal y of all the room coordinates
{
        POINT resMinPoint;
        int i;

        resMinPoint.x=coordinates[0].x; //Can be done because of PRE
        resMinPoint.y=coordinates[0].y; //Can be done because of PRE

        for (i=0;i<numberOfCoordinates;i++)
        {
                if (resMinPoint.x>coordinates[i].x)
                {
                        resMinPoint.x=coordinates[i].x;
                }
                if (resMinPoint.y>coordinates[i].y)
                {
                        resMinPoint.y=coordinates[i].y;
                }
        }
```

```
        return resMinPoint;
}

POINT Room::getMaxCoordinates(void)
//PRE : numberOfCoordinates>0
// returns the maximal x and the maximal y of all the room coordinates
{
        POINT resPoint;
        int i;

        resPoint.x=coordinates[0].x; //Can be done because of PRE
        resPoint.y=coordinates[0].y; //Can be done because of PRE

        for (i=0;i<numberOfCoordinates;i++)
        {
                if (resPoint.x<coordinates[i].x) resPoint.x=coordinates[i].x;
                if (resPoint.y<coordinates[i].y) resPoint.y=coordinates[i].y;
        }

        return resPoint;
}
```

E.16 artobject.h

```
// ==================================================================
// HEADER FILE for artobject.cpp
// ==================================================================

#define PRG_VERSION              "Ravi's Guide v.0.1.1"
#define MAX_STRINGSIZE           50
#define MAX_COORDINATES          50
#define MAX_CONNEXIONS           50
#define MAX_ARTOBJECTS           500
#define DEFAULT_CNX_DRW_ROOM     4
#define DEFAULT_CNX_DRW_OVERVW   1

#define ARTOBJECT_ATTRIBUTE1     "Person"      //"Artist"
#define ARTOBJECT_ATTRIBUTE2     "Title"       //"Style"
#define ARTOBJECT_ATTRIBUTE3     "Domain"      //"Period"
#define ARTOBJECT_ATTRIBUTE4     "Team"        //"Country"
#define ARTOBJECT_ATTRIBUTE5     "Comment"     //"Material"

class ArtObject
{
public :
      char code     [MAX_STRINGSIZE];
      char name     [MAX_STRINGSIZE];
      char author   [MAX_STRINGSIZE];
      char style    [MAX_STRINGSIZE];
      char period   [MAX_STRINGSIZE];
      char country  [MAX_STRINGSIZE];
      char material [MAX_STRINGSIZE];

      //HYPOTHESIS : the art object is represented by a
      //vertical/horizontal rectangle
      POINT positionA;          //top left     coordinate of the object
      POINT positionB;          //bottom right coordinate of the object

      int isToSee;
      int hasBeenSeen;

      ArtObject * nextObjectInRoom;

      //Default Constructor
      ArtObject(void);

      //Personnalized Constructor
      ArtObject(char myCode            [MAX_STRINGSIZE],
                char myName            [MAX_STRINGSIZE],
                char myAuthor          [MAX_STRINGSIZE],
                char myStyle           [MAX_STRINGSIZE],
                char myPeriod          [MAX_STRINGSIZE],
                char myCountry         [MAX_STRINGSIZE],
                char myMaterial        [MAX_STRINGSIZE],

                POINT myPositionA,
                POINT myPositionB,

                int myToSee,
                int mySeen
            );

      //Default Destructor
      ~ArtObject(void);
};
```

E.17 artobject.cpp

```
// ================================================================
// ArtObject core file
// ================================================================

#include <windows.h>      // for all the windows stuff
#include <stdio.h>        // for string manipulation
#include "artobject.h"

//Default Constructor
ArtObject::ArtObject(void)
{
       strcpy(code,"");
       strcpy(name,"");
       strcpy(author,"");
       strcpy(style,"");
       strcpy(period,"");
       strcpy(country,"");
       strcpy(material,"");
       positionA.x=0;
       positionA.y=0;
       positionB.x=0;
       positionB.y=0;
       isToSee=0;
       hasBeenSeen=0;
       nextObjectInRoom=NULL;
       return;
}

//Personnalized Constructor
ArtObject::ArtObject(char myCode [MAX_STRINGSIZE],
             char myName       [MAX_STRINGSIZE],
             char myAuthor     [MAX_STRINGSIZE],
             char myStyle      [MAX_STRINGSIZE],
             char myPeriod     [MAX_STRINGSIZE],
             char myCountry    [MAX_STRINGSIZE],
             char myMaterial   [MAX_STRINGSIZE],
             POINT myPositionA,
             POINT myPositionB,
             int myToSee,
             int mySeen)
{
       strcpy(code,myCode);
       strcpy(name,myName);
       strcpy(author,myAuthor);
       strcpy(style,myStyle);
       strcpy(period,myPeriod);
       strcpy(country,myCountry);
       strcpy(material,myMaterial);
       positionA.x=myPositionA.x;
       positionA.y=myPositionA.y;
       positionB.x=myPositionB.x;
       positionB.y=myPositionB.y;
       isToSee=myToSee;
       hasBeenSeen=mySeen;
       nextObjectInRoom=NULL;
}

//Default Destructor
ArtObject::~ArtObject(void)
{
}
```

Facultés Universitaires Notre-Dame de la Paix, Namur
Institut d'Informatique
Année Académique 2002-2003

Addenda To
Generic PDA
Based Museum Guide
With Sound Enhancement

Ravi R.C. Ramdoyal

Mémoire présenté en vue de l'obtention
du grade de Maître en Informatique

Note

This addenda contains the screenshots omitted in Appendix B as well as the further analysis on the effective impact of the various explicative factors on the apprehension of our prototype, as mentionned in Chapter 9 and Appendix D.

Note

Cet addenda contient les captures d'écran omises dans l'Appendice B ainsi que les analyses complémentaires sur l'impact réel des divers facteurs explicatifs sur l'appréhension de notre prototype, comme mentionnées dans le Chapitre 9 et l'Appendice D.

Content

Content i

Addendum To Appendix B : Screenshots 1

B'.1 Screenshots Of The Small And Simple Environment 2
B'.2 Screenshots Of The Fourth Floor Of The Computer Science Institute 3

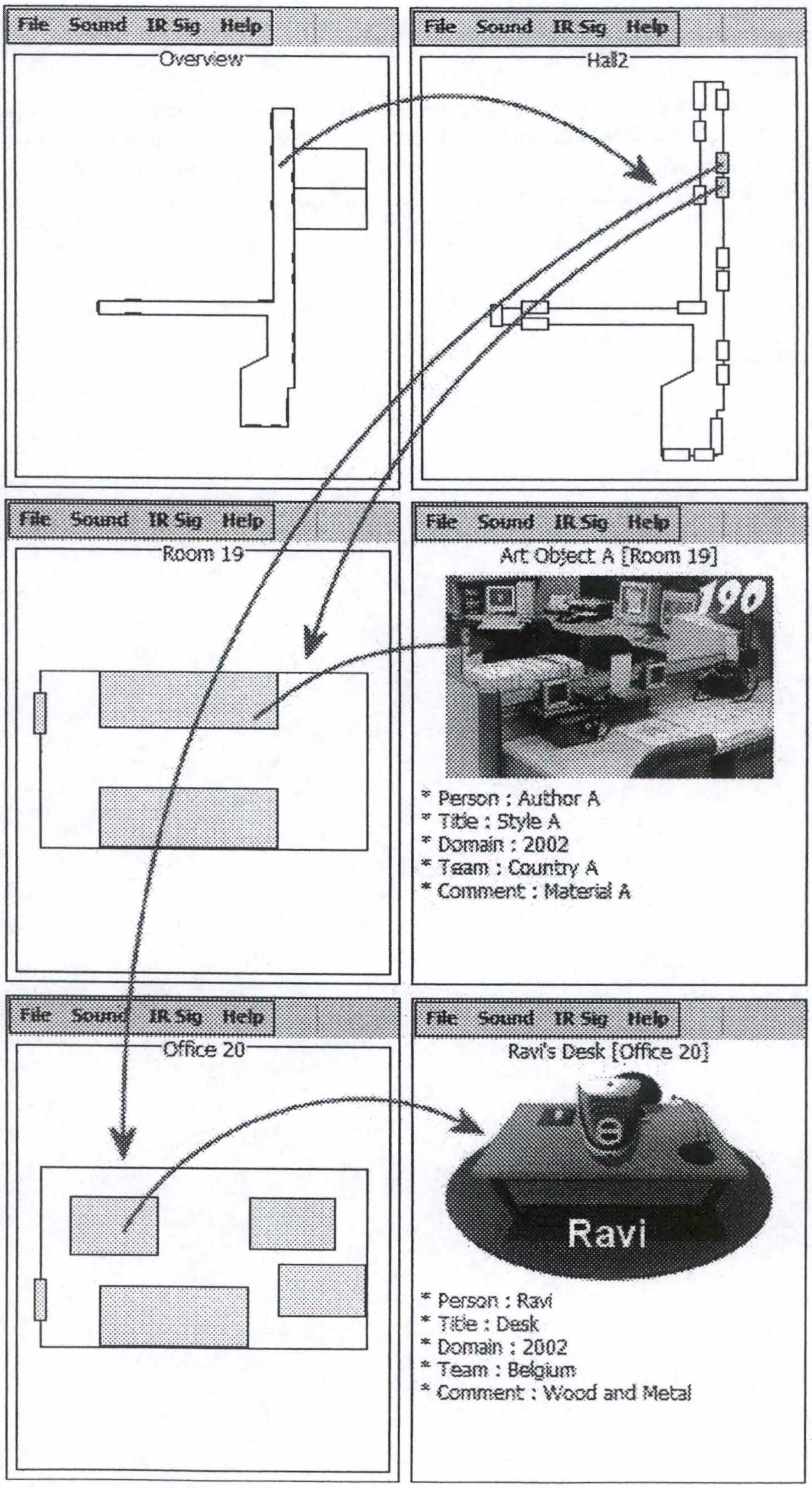
Addendum To Appendix D : Apprehension Of The Prototype 4

D'.1 Case By Case Analysis 5
D'.1.1 Scenario 1 5
D'.1.2 Scenario 2 7
D'.1.3 Scenario 3 11
D'.2 Analysis Of The Impact Of The Musical Formation 17
D'.2.1 Scenario 1 17
D'.2.2 Scenario 2 19
D'.2.3 Scenario 3 23
D'.3 Analysis Of The Impact Of The Use Of Video Games 29
D'.3.1 Scenario 1 29
D'.3.2 Scenario 2 31
D'.3.3 Scenario 3 35
D'.4 Analysis Of The Impact Of Video Games For Novice PDA Users 41
D'.4.1 Scenario 1 41
D'.4.2 Scenario 2 43
D'.4.3 Scenario 3 47

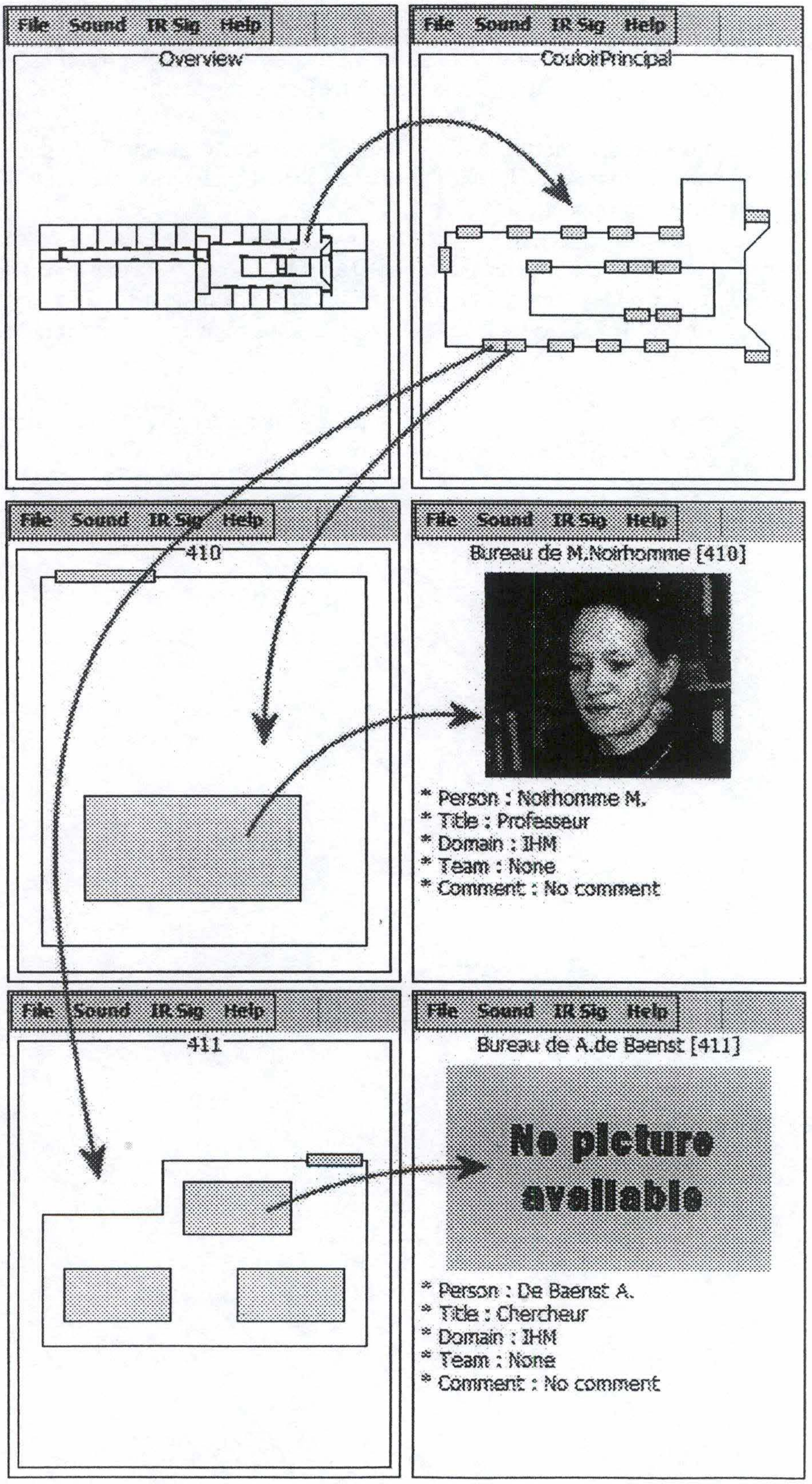
Addendum To Appendix B : Screenshots

The following pages give some screenshots corresponding to the two environment specification examples given in Appendix B. Let us remind that the first example is a small environment composed of a corridor and two rooms, while the second one is the fourth floor of the Computer Institute, which was used for the evaluation of the application.

B'.1 Screenshots Of The Small And Simple Environment



B'.2 Screenshots Of The Fourth Floor Of The Computer Science Institute



Addendum To Appendix D : Apprehension Of The Prototype

The following pages present more analysis regarding the relevance of our findings for the apprehension of the prototype (see Chapter 9 and Appendix D, section D.2).

As you will see, we are forced to notice that the trends we assumed about the impact of the musical background were ill-founded. The regular use of video games (whether being used to PDAs or not) had definitely no impact.

As a matter of fact, except for the previous use of PDAs, all the others classes we analyzed (musical background, use of video games with and without the persons that had already used a PDA) always had a critical probability > 0.05, which means no relevant conclusion could be drawn.

This implies that in the future, we should push our inquiry more deeply, and question more subjects, in order to get a more representative population.

D'.1 Case By Case Analysis

D'.1.1 Scenario 1

Critical Probability Analysis

X1 ≡ persons who never used a PDA before
X2 ≡ persons who already used a PDA at least once before

X1	(X1-Mean(X1))	(X1-Mean(X1))²	X2	(X2-Mean(X2))	(X2-Mean(X2))²
18	-0,05	0,0025	10	1,125	1,265625
17	-1,05	1,1025	8	-0,875	0,765625
14	-4,05	16,4025	11	2,125	4,515625
15	-3,05	9,3025	9	0,125	0,015625
18	-0,05	0,0025	10	1,125	1,265625
19	0,95	0,9025	5	-3,875	15,015625
11	-7,05	49,7025	7	-1,875	3,515625
25	6,95	48,3025	11	2,125	4,515625
15	-3,05	9,3025			
23	4,95	24,5025			
10	-8,05	64,8025			
22	3,95	15,6025			
5	-13,05	170,3025			
26	7,95	63,2025			
16	-2,05	4,2025			
16	-2,05	4,2025			
15	-3,05	9,3025			
11	-7,05	49,7025			
18	-0,05	0,0025			
47	28,95	838,1025			

n1 = 20
n2 = 8
Mean (X1) = 18,05
Mean (X1) = 8,875

Calculation of sp

Σ (X1-Mean(X1))² = 1378,95
Σ (X2-Mean(X2))² = 30,875
ddl = ((n1-1)+(n2-1)) = 26

(Σ (X1-Mean(X1))² + Σ (X2-Mean(X2))²) = 1409,825
sp² = (Σ (X1-Mean(X1))² + Σ (X1-Mean(X1))² / ddl = 54,22403846

Calculation of the Interval of Confidence

Mean(X1) - Mean(X2) = 9,175
t.025 (for current ddl) = 2,06
sp = 7,363697336
(1/n1) = 0,05

(1/n2) =	0,125
(1/n1)+(1/n2) =	0,175
SquareRoot((1/n1)+(1/n2)) =	0,418330013
sp*SquareRoot((1/n1)+(1/n2)) =	3,080455604
t.025*sp*SquareRoot((1/n1)+(1/n2)) =	6,345738545

Hence (μ1-μ2) is between 2,829261455 and 15,52073855.

Calculation of the Critical Probability

t = (Mean(X1)-Mean(X2)) / (sp*SquareRoot((1/n1)+(1/n2))) = 2,97845552

For the current ddl, the Critical Values Table For the t Distribution of Student gives us :
t > t.005, which means pc < .005

D'.1.2 Scenario 2

Scenario 2a : Critical Probability Analysis

X1 ≡ persons who never used a PDA before
X2 ≡ persons who already used a PDA at least once before

X1	(X1-Mean(X1))	(X1-Mean(X1))²	X2	(X2-Mean(X2))	(X2-Mean(X2))²
10	-4,6	21,16	13	4,25	18,0625
12	-2,6	6,76	6	-2,75	7,5625
21	6,4	40,96	11	2,25	5,0625
10	-4,6	21,16	6	-2,75	7,5625
15	0,4	0,16	9	0,25	0,0625
9	-5,6	31,36	7	-1,75	3,0625
11	-3,6	12,96	10	1,25	1,5625
12	-2,6	6,76	8	-0,75	0,5625
60	45,4	2061,16			
15	0,4	0,16			
15	0,4	0,16			
10	-4,6	21,16			
6	-8,6	73,96			
20	5,4	29,16			
14	-0,6	0,36			
8	-6,6	43,56			
8	-6,6	43,56			
10	-4,6	21,16			
12	-2,6	6,76			
14	-0,6	0,36			

n1 = 20
n2 = 8
Mean(X1) = 14,6
Mean(X2) = 8,75

Calculation of sp

Σ (X1-Mean(X1))² = 2442,8
Σ (X2-Mean(X2))² = 43,5
ddl = ((n1-1)+(n2-1)) = 26

(Σ (X1-Mean(X1))² + Σ (X2-Mean(X2))²) = 2486,3
sp² = (Σ (X1-Mean(X1))² + Σ (X1-Mean(X1))² / ddl = 95,62692308

Calculation of the Interval of Confidence

Mean(X1) - Mean(X2) = 5,85
t.025 (for current ddl) = 2,06
sp = 9,778901936
(1/n1) = 0,05
(1/n2) = 0,125
(1/n1)+(1/n2) = 0,175

SquareRoot((1/n1)+(1/n2)) = 0,418330013
sp*SquareRoot((1/n1)+(1/n2)) = 4,090808177
t.025*sp*SquareRoot((1/n1)+(1/n2)) = 8,427064844

Hence (μ1-μ2) is between -2,577064844 and 14,27706484.

Calculation of the Critical Probability

t = (Mean(X1)-Mean(X2)) / (sp*SquareRoot((1/n1)+(1/n2))) = 1,430035276

For the current ddl, the Critical Values Table For the t Distribution of Student gives us :
t> t.10, which means pc < .10

Scenario 2b : Critical Probability Analysis

X1 ≡ persons who never used a PDA before
X2 ≡ persons who already used a PDA at least once before

X1	(X1-Mean(X1))	$(X1\text{-Mean}(X1))^2$	X2	(X2-Mean(X2))	$(X2\text{-Mean}(X2))^2$
14	1,05	1,1025	11	3	9
6	-6,95	48,3025	9	1	1
11	-1,95	3,8025	7	-1	1
12	-0,95	0,9025	7	-1	1
9	-3,95	15,6025	11	3	9
12	-0,95	0,9025	5	-3	9
11	-1,95	3,8025	6	-2	4
17	4,05	16,4025	8	0	0
11	-1,95	3,8025			
18	5,05	25,5025			
9	-3,95	15,6025			
10	-2,95	8,7025			
9	-3,95	15,6025			
10	-2,95	8,7025			
10	-2,95	8,7025			
8	-4,95	24,5025			
14	1,05	1,1025			
12	-0,95	0,9025			
22	9,05	81,9025			
34	21,05	443,1025			

n1 = 20
n2 = 8
Mean(X1) = 12,95
Mean(X2) = 8

Calculation of sp

Σ (X1-Mean(X1))² = 728,95
Σ (X2-Mean(X2))² = 34
ddl = ((n1-1)+(n2-1)) = 26

(Σ (X1-Mean(X1))² + Σ (X2-Mean(X2))²) = 762,95
sp² = (Σ (X1-Mean(X1))² + Σ (X1-Mean(X1))² / ddl = 29,34423077

Calculation of the Interval of Confidence

Mean(X1) - Mean(X2) = 4,95
t.025 (for current ddl) = 2,06
sp = 5,417031546
(1/n1) = 0,05
(1/n2) = 0,125
(1/n1)+(1/n2) = 0,175
SquareRoot((1/n1)+(1/n2)) = 0,418330013
sp*SquareRoot((1/n1)+(1/n2)) = 2,266106878
t.025*sp*SquareRoot((1/n1)+(1/n2)) = 4,66818017

Hence (μ1-μ2) is between 0,28181983 and 9,61818017.

Calculation of the Critical Probability

t = (Mean(X1)-Mean(X2)) / (sp*SquareRoot((1/n1)+(1/n2))) = 2,184362991

For the current ddl, the Critical Values Table For the t Distribution of Student gives us :
t > t.025, which means pc < .025

D'.1.3 Scenario 3

Scenario 3a : Critical Probability Analysis

This analysis concerns the 23 persons from biology, computer science, mathematics and maths/computer-science, which were the fastest persons for the Scenario 3a.

X1 ≡ persons with no musical formation
X2 ≡ with at least a basic musical formation

X1	(X1-Mean(X1))	$(X1\text{-Mean}(X1))^2$	X2	(X2-Mean(X2))	$(X2\text{-Mean}(X2))^2$
25	-0,615384615	0,378698225	21	1,4	1,96
16	-9,615384615	92,4556213	14	-5,6	31,36
29	3,384615385	11,4556213	21	1,4	1,96
49	23,38461538	546,8402367	24	4,4	19,36
21	-4,615384615	21,30177515	11	-8,6	73,96
23	-2,615384615	6,840236686	27	7,4	54,76
21	-4,615384615	21,30177515	14	-5,6	31,36
10	-15,61538462	243,8402367	12	-7,6	57,76
24	-1,615384615	2,609467456	19	-0,6	0,36
33	7,384615385	54,53254438	33	13,4	179,56
21	-4,615384615	21,30177515			
45	19,38461538	375,7633136			
16	-9,615384615	92,4556213			

n1 = 13
n2 = 10
Mean(X1) = 25,61538462
Mean(X2) = 19,6

Calculation of sp

$\Sigma\,(X1\text{-Mean}(X1))^2 =$ 1491,076923
$\Sigma\,(X2\text{-Mean}(X2))^2 =$ 452,4
ddl = ((n1-1)+(n2-1)) = 21

$(\Sigma\,(X1\text{-Mean}(X1))^2 + \Sigma\,(X2\text{-Mean}(X2))^2) =$ 1943,476923
$sp^2 = (\Sigma\,(X1\text{-Mean}(X1))^2 + \Sigma\,(X1\text{-Mean}(X1))^2\,/\,ddl =$ 92,54652015

Calculation of the Interval of Confidence

Mean(X1) - Mean(X2) = 6,015384615
t.025 (for current ddl) = 2,08
sp = 9,620110194
(1/n1) = 0,076923077
(1/n2) = 0,1
(1/n1)+(1/n2) = 0,176923077
SquareRoot((1/n1)+(1/n2)) = 0,42062225
sp*SquareRoot((1/n1)+(1/n2)) = 4,046432392
t.025*sp*SquareRoot((1/n1)+(1/n2)) = 8,416579375

Hence ($\mu1$-$\mu2$) is between -2,40119476 and 14,43196399

Calculation of the Critical Probability

t = (Mean(X1)-Mean(X2)) / (sp*SquareRoot((1/n1)+(1/n2))) = 1,486589675

For the current ddl, the Critical Values Table For the t Distribution of Student gives us :
t < t0.10, which means that pc is < 0.10

Scenario 3b : Critical Probability Analysis

This analysis concerns the 23 persons from philosophy, computer science, mathematics and maths/computer-science, which were the slowest persons for the Scenario 3b.

X1 ≡ persons who do not play video games daily
X2 ≡ persons who already play video games daily

X1	(X1-Mean(X1))	$(X1\text{-Mean}(X1))^2$	X2	(X2-Mean(X2))	$(X2\text{-Mean}(X2))^2$
31	6,444444444	41,5308642	32	-2	4
24	-0,555555556	0,308641975	17	-17	289
46	21,44444444	459,8641975	75	41	1681
8	-16,55555556	274,0864198	12	-22	484
16	-8,555555556	73,19753086	34	0	0
26	1,444444444	2,086419753			
15	-9,555555556	91,30864198			
32	7,444444444	55,41975309			
23	-1,555555556	2,419753086			
17	-7,555555556	57,08641975			
30	5,444444444	29,64197531			
25	0,444444444	0,197530864			
23	-1,555555556	2,419753086			
8	-16,55555556	274,0864198			
7	-17,55555556	308,1975309			
58	33,44444444	1118,530864			
30	5,444444444	29,64197531			
23	-1,555555556	2,419753086			

n1 = 18
n2 = 5
Mean(X1) = 24,55555556
Mean(X2) = 34

Calculation of sp

Σ $(X1\text{-Mean}(X1))^2$ = 2822,444444
Σ $(X2\text{-Mean}(X2))^2$ = 2458
ddl = ((n1-1)+(n2-1)) = 21

(Σ $(X1\text{-Mean}(X1))^2$ + Σ $(X2\text{-Mean}(X2))^2$) = 5280,444444
sp^2 = (Σ $(X1\text{-Mean}(X1))^2$ + Σ $(X1\text{-Mean}(X1))^2$ / ddl = 251,4497354

Calculation of the Interval of Confidence

Mean(X1) - Mean(X2) = 9,444444444
t.025 (for current ddl) = 2,08
sp = 15,85716669
(1/n1) = 0,055555556
(1/n2) = 0,2
(1/n1)+(1/n2) = 0,255555556
SquareRoot((1/n1)+(1/n2)) = 0,50552503
sp*SquareRoot((1/n1)+(1/n2)) = 8,016194661

t.025*sp*SquareRoot((1/n1)+(1/n2)) = 16,67368489

Hence (μ1-μ2) is between -7,22924045 and 26,11812934

Calculation of the Critical Probability

t = (Mean(X1)-Mean(X2)) / (sp*SquareRoot((1/n1)+(1/n2))) =1,178170547

For the current ddl, the Critical Values Table For the t Distribution of Student gives us :
t > t.25, which means that pc is < 0.25

Scenario 3c : Critical Probability Analysis

This analysis concerns the 23 persons from biology, philosophy, geography, computer science and maths/computer-science, which were the fastest persons for the Scenario 3c.

X1 ≡ persons who never used a PDA before
X2 ≡ persons who already used a PDA at least once before

X1	(X1-Mean(X1))	$(X1\text{-}Mean(X1))^2$	X2	(X2-Mean(X2))	$(X2\text{-}Mean(X2))^2$
12	-0,6	0,36	8	-13,125	172,265625
7	-5,6	31,36	12	-9,125	83,265625
16	3,4	11,56	45	23,875	570,015625
17	4,4	19,36	6	-15,125	228,765625
15	2,4	5,76	66	44,875	2013,765625
5	-7,6	57,76	13	-8,125	66,015625
6	-6,6	43,56	11	-10,125	102,515625
17	4,4	19,36	8	-13,125	172,265625
12	-0,6	0,36			
20	7,4	54,76			
7	-5,6	31,36			
20	7,4	54,76			
13	0,4	0,16			
15	2,4	5,76			
7	-5,6	31,36			

n1 = 15
n2 = 8
Mean(X1) = 12,6
Mean(X2) = 21,125

Calculation of sp

$\Sigma\ (X1\text{-}Mean(X1))^2 =$ 367,6
$\Sigma\ (X2\text{-}Mean(X2))^2 =$ 3408,875
ddl = ((n1-1)+(n2-1)) = 21

$(\Sigma\ (X1\text{-}Mean(X1))^2 + \Sigma\ (X2\text{-}Mean(X2))^2) =$ 3776,475
$sp^2 = (\Sigma\ (X1\text{-}Mean(X1))^2 + \Sigma\ (X1\text{-}Mean(X1))^2\ /\ ddl =$ 179,8321429

Calculation of the Interval of Confidence

Mean(X1) - Mean(X2) =	8,525
t.025 (for current ddl) =	2,1
sp =	13,41015074
(1/n1) =	0,066666667
(1/n2) =	0,125
(1/n1)+(1/n2) =	0,191666667
SquareRoot((1/n1)+(1/n2)) =	0,437797518
sp*SquareRoot((1/n1)+(1/n2)) =	5,870930708
t.025*sp*SquareRoot((1/n1)+(1/n2)) =	12,32895449

Hence ($\mu 1$-$\mu 2$) is between -3,803954487 and 20,85395449

Calculation of the Critical Probability

t = (Mean(X1)-Mean(X2)) / (sp*SquareRoot((1/n1)+(1/n2))) = 1,452069599

For the current ddl, the Critical Values Table For the t Distribution of Student gives us :
t > t.25, which means that pc is < 0.25

D'.2 Analysis Of The Impact Of The Musical Formation

D'.2.1 Scenario 1

Critical Probability Analysis

X1 ≡ persons with no musical formation
X2 ≡ with at least a basic musical formation

X1	(X1-Mean(X1))	$(X1\text{-Mean}(X1))^2$	X2	(X2-Mean(X2))	$(X2\text{-Mean}(X2))^2$
18	2,285714286	5,224489796	11	-4,142857143	17,16326531
15	-0,714285714	0,510204082	11	-4,142857143	17,16326531
18	2,285714286	5,224489796	18	2,857142857	8,163265306
19	3,285714286	10,79591837	47	31,85714286	1014,877551
25	9,285714286	86,2244898	8	-7,142857143	51,02040816
23	7,285714286	53,08163265	16	0,857142857	0,734693878
10	-5,714285714	32,65306122	11	-4,142857143	17,16326531
22	6,285714286	39,51020408	17	1,857142857	3,448979592
5	-10,71428571	114,7959184	16	0,857142857	0,734693878
26	10,28571429	105,7959184	11	-4,142857143	17,16326531
15	-0,714285714	0,510204082	14	-1,142857143	1,306122449
9	-6,714285714	45,08163265	15	-0,142857143	0,020408163
10	-5,714285714	32,65306122	10	-5,142857143	26,44897959
5	-10,71428571	114,7959184	7	-8,142857143	66,30612245

n1 = 14
n2 = 14
Mean(X1) = 10,85714286
Mean(X2) = 15

Calculation of sp

$\Sigma\ (X1\text{-Mean}(X1))^2 =$ 215,7142857
$\Sigma\ (X2\text{-Mean}(X2))^2 =$ 2346
ddl = ((n1-1)+(n2-1)) = 26

$(\Sigma\ (X1\text{-Mean}(X1))^2 + \Sigma\ (X2\text{-Mean}(X2))^2) =$ 2561,714286
$sp^2 = (\Sigma\ (X1\text{-Mean}(X1))^2 + \Sigma\ (X1\text{-Mean}(X1))^2 / ddl =$ 98,52747253

Calculation of the Interval of Confidence

Mean(X1) - Mean(X2) = 4,142857143
t.025 (for current ddl) = 2,06
sp = 9,92610057
(1/n1) = 0,071428571
(1/n2) = 0,071428571
(1/n1)+(1/n2) = 0,142857143
SquareRoot((1/n1)+(1/n2)) = 0,377964473
sp*SquareRoot((1/n1)+(1/n2)) = 3,751713371
t.025*sp*SquareRoot((1/n1)+(1/n2)) = 7,728529544

Hence (μ1-μ2) is between -3,585672401 and 11,87138669

Calculation of the Critical Probability

t = (Mean(X1)-Mean(X2)) / (sp*SquareRoot((1/n1)+(1/n2))) = 1,104257371

For the current ddl, the Critical Values Table For the t Distribution of Student gives us :
t > t.25, which means that pc is < .25

D'.2.2 Scenario 2

Scenario 2a : Critical Probability Analysis

X1 ≡ persons with no musical formation
X2 ≡ with at least a basic musical formation

X1	(X1-Mean(X1))	(X1-Mean(X1))²	X2	(X2-Mean(X2))	(X2-Mean(X2))²
10	-0,857142857	0,734693878	11	-4	16
10	-0,857142857	0,734693878	10	-5	25
15	4,142857143	17,16326531	12	-3	9
9	-1,857142857	3,448979592	14	-1	1
12	1,142857143	1,306122449	6	-9	81
15	4,142857143	17,16326531	8	-7	49
15	4,142857143	17,16326531	8	-7	49
10	-0,857142857	0,734693878	12	-3	9
6	-4,857142857	23,59183673	14	-1	1
20	9,142857143	83,59183673	11	-4	16
8	-2,857142857	8,163265306	21	6	36
6	-4,857142857	23,59183673	60	45	2025
9	-1,857142857	3,448979592	13	-2	4
7	-3,857142857	14,87755102	10	-5	25

n1 = 14
n2 = 14
Mean(X1) = 10,85714286
Mean(X2) = 15

Calculation of sp

Σ (X1-Mean(X1))² = 215,7142857
Σ (X2-Mean(X2))² = 2346
ddl = ((n1-1)+(n2-1)) = 26

(Σ (X1-Mean(X1))² + Σ (X2-Mean(X2))²) = 2561,714286
sp² = (Σ (X1-Mean(X1))² + Σ (X1-Mean(X1))² / ddl = 98,52747253

Calculation of the Interval of Confidence

Mean(X1) - Mean(X2) = 4,142857143
t.025 (for current ddl) = 2,06
sp = 9,92610057
(1/n1) = 0,071428571
(1/n2) = 0,071428571
(1/n1)+(1/n2) = 0,142857143
SquareRoot((1/n1)+(1/n2)) = 0,377964473
sp*SquareRoot((1/n1)+(1/n2)) = 3,751713371
t.025*sp*SquareRoot((1/n1)+(1/n2)) = 7,728529544

Hence (μ1-μ2) is between -3,585672401 and 11,87138669

Calculation of the Critical Probability

t = (Mean(X1)-Mean(X2)) / (sp*SquareRoot((1/n1)+(1/n2))) = 1,104257371

For the current ddl, the Critical Values Table For the t Distribution of Student gives us :
t > t.25, which means that pc is < .25

Scenario 2b : Critical Probability Analysis

X1 ≡ persons with no musical formation
X2 ≡ with at least a basic musical formation

X1	(X1-Mean(X1))	$(X1\text{-Mean}(X1))^2$	X2	(X2-Mean(X2))	$(X2\text{-Mean}(X2))^2$
14	2,785714286	7,760204082	11	-0,857142857	0,734693878
12	0,785714286	0,617346939	12	0,142857143	0,020408163
9	-2,214285714	4,903061224	22	10,14285714	102,877551
12	0,785714286	0,617346939	34	22,14285714	490,3061224
17	5,785714286	33,4744898	9	-2,857142857	8,163265306
18	6,785714286	46,04591837	8	-3,857142857	14,87755102
9	-2,214285714	4,903061224	8	-3,857142857	14,87755102
10	-1,214285714	1,474489796	6	-5,857142857	34,30612245
9	-2,214285714	4,903061224	10	-1,857142857	3,448979592
10	-1,214285714	1,474489796	7	-4,857142857	23,59183673
14	2,785714286	7,760204082	11	-0,857142857	0,734693878
7	-4,214285714	17,76020408	11	-0,857142857	0,734693878
11	-0,214285714	0,045918367	11	-0,857142857	0,734693878
5	-6,214285714	38,61734694	6	-5,857142857	34,30612245

n1 = 14
n2 = 14
Mean(X1) = 11,21428571
Mean(X2) = 11,85714286

Calculation of sp

$\Sigma\,(X1\text{-Mean}(X1))^2 =$ 170,3571429
$\Sigma\,(X2\text{-Mean}(X2))^2 =$ 729,7142857
ddl = ((n1-1)+(n2-1)) = 26

$(\Sigma\,(X1\text{-Mean}(X1))^2 + \Sigma\,(X2\text{-Mean}(X2))^2) =$ 900,0714286
$sp^2 = (\Sigma\,(X1\text{-Mean}(X1))^2 + \Sigma\,(X1\text{-Mean}(X1))^2 / \text{ddl} =$ 34,61813187

Calculation of the Interval of Confidence

Mean(X1) - Mean(X2) =	0,642857143
t.025 (for current ddl) =	2,06
sp =	5,883717521
(1/n1) =	0,071428571
(1/n2) =	0,071428571
(1/n1)+(1/n2) =	0,142857143
SquareRoot((1/n1)+(1/n2)) =	0,377964473
sp*SquareRoot((1/n1)+(1/n2)) =	2,223836192
t.025*sp*SquareRoot((1/n1)+(1/n2)) =	4,581102556

Hence (μ1-μ2) is between -3,938245413 and 5,223959699

Calculation of the Critical Probability

t = (Mean(X1)-Mean(X2)) / (sp*SquareRoot((1/n1)+(1/n2))) = 0,289075762

For the current ddl, the Critical Values Table For the t Distribution of Student gives us :
$t < t.25$, which means that pc is not $< .25$

D'.2.3 Scenario 3

Scenario 3a : Critical Probability Analysis

X1 ≡ persons with no musical formation
X2 ≡ with at least a basic musical formation

X1	(X1-Mean(X1))	(X1-Mean(X1))²	X2	(X2-Mean(X2))	(X2-Mean(X2))²
25	-0,785714286	0,617346939	14	-9,714285714	94,36734694
16	-9,785714286	95,76020408	21	-2,714285714	7,367346939
29	3,214285714	10,33163265	59	35,28571429	1245,081633
49	23,21428571	538,9030612	29	5,285714286	27,93877551
21	-4,785714286	22,90306122	21	-2,714285714	7,367346939
23	-2,785714286	7,760204082	11	-12,71428571	161,6530612
21	-4,785714286	22,90306122	24	0,285714286	0,081632653
10	-15,78571429	249,1887755	32	8,285714286	68,65306122
24	-1,785714286	3,18877551	14	-9,714285714	94,36734694
45	19,21428571	369,1887755	27	3,285714286	10,79591837
16	-9,785714286	95,76020408	12	-11,71428571	137,2244898
33	7,214285714	52,04591837	33	9,285714286	86,2244898
21	-4,785714286	22,90306122	16	-7,714285714	59,51020408
28	2,214285714	4,903061224	19	-4,714285714	22,2244898

n1 =	14
n2 =	14
Mean(X1) =	25,78571429
Mean(X2) =	23,71428571

Calculation of sp

Σ (X1-Mean(X1))² =	1496,357143
Σ (X2-Mean(X2))² =	2022,857143
ddl = ((n1-1)+(n2-1)) =	26

(Σ (X1-Mean(X1))² + Σ (X2-Mean(X2))²) =	3519,214286
sp² = (Σ (X1-Mean(X1))² + Σ (X1-Mean(X1))² / ddl =	135,3543956

Calculation of the Interval of Confidence

Mean(X1) - Mean(X2) =	2,071428571
t.025 (for current ddl) =	2,06
sp =	11,6341908
(1/n1) =	0,071428571
(1/n2) =	0,071428571
(1/n1)+(1/n2) =	0,142857143
SquareRoot((1/n1)+(1/n2)) =	0,377964473
sp*SquareRoot((1/n1)+(1/n2)) =	4,397310795
t.025*sp*SquareRoot((1/n1)+(1/n2)) =	9,058460238

Hence (μ1-μ2) is between -6,987031667 and 11,12988881

Calculation of the Critical Probability

t = (Mean(X1)-Mean(X2)) / (sp*SquareRoot((1/n1)+(1/n2))) = 0,471067129

For the current ddl, the Critical Values Table For the t Distribution of Student gives us :
t < t.25, which means that pc is not < .25

Scenario 3b : Critical Probability Analysis

X1 ≡ persons with no musical formation
X2 ≡ with at least a basic musical formation

X1	(X1-Mean(X1))	(X1-Mean(X1))²	X2	(X2-Mean(X2))	(X2-Mean(X2))²
15	-10,21428571	104,3316327	32	8,714285714	75,93877551
15	-10,21428571	104,3316327	34	10,71428571	114,7959184
16	-9,214285714	84,90306122	24	0,714285714	0,510204082
32	6,785714286	46,04591837	6	-17,28571429	298,7959184
30	4,785714286	22,90306122	26	2,714285714	7,367346939
23	-2,214285714	4,903061224	30	6,714285714	45,08163265
17	-8,214285714	67,4744898	12	-11,28571429	127,3673469
75	49,78571429	2478,617347	13	-10,28571429	105,7959184
8	-17,21428571	296,3316327	31	7,714285714	59,51020408
23	-2,214285714	4,903061224	17	-6,285714286	39,51020408
23	-2,214285714	4,903061224	46	22,71428571	515,9387755
7	-18,21428571	331,7602041	25	1,714285714	2,93877551
58	32,78571429	1074,903061	22	-1,285714286	1,653061224
11	-14,21428571	202,0459184	8	-15,28571429	233,6530612

n1 = 14
n2 = 14
Mean(X1) = 25,21428571
Mean(X2) = 23,28571429

Calculation of sp

Σ (X1-Mean(X1))² = 4828,357143
Σ (X2-Mean(X2))² = 1628,857143
ddl = ((n1-1)+(n2-1)) = 26

(Σ (X1-Mean(X1))² + Σ (X2-Mean(X2))²) = 6457,214286
sp² = (Σ (X1-Mean(X1))² + Σ (X1-Mean(X1))² / ddl = 248,3543956

Calculation of the Interval of Confidence

Mean(X1) - Mean(X2) = 1,928571429
t.025 (for current ddl) = 2,06
sp = 15,7592638
(1/n1) = 0,071428571
(1/n2) = 0,071428571
(1/n1)+(1/n2) = 0,142857143
SquareRoot((1/n1)+(1/n2)) = 0,377964473
sp*SquareRoot((1/n1)+(1/n2)) = 5,956441838
t.025*sp*SquareRoot((1/n1)+(1/n2)) = 12,27027019

Hence (μ1-μ2) is between -10,34169876 and 14,19884162

Calculation of the Critical Probability

t = (Mean(X1)-Mean(X2)) / (sp*SquareRoot((1/n1)+(1/n2))) = 0,323779109

For the current ddl, the Critical Values Table For the t Distribution of Student gives us :
$t < t_{.25}$, which means that pc is not $< .25$

Scenario 3c : Critical Probability Analysis

X1 ≡ persons with no musical formation
X2 ≡ with at least a basic musical formation

X1	(X1-Mean(X1))	(X1-Mean(X1))²	X2	(X2-Mean(X2))	(X2-Mean(X2))²
12	-7,928571429	62,8622449	6	-9,928571429	98,57653061
17	-2,928571429	8,576530612	15	-0,928571429	0,862244898
15	-4,928571429	24,29081633	7	-8,928571429	79,71938776
5	-14,92857143	222,8622449	56	40,07142857	1605,719388
17	-2,928571429	8,576530612	12	-3,928571429	15,43367347
20	0,071428571	0,005102041	11	-4,928571429	24,29081633
7	-12,92857143	167,1479592	8	-7,928571429	62,8622449
20	0,071428571	0,005102041	7	-8,928571429	79,71938776
13	-6,928571429	48,00510204	9	-6,928571429	48,00510204
18	-1,928571429	3,719387755	45	29,07142857	845,1479592
50	30,07142857	904,2908163	16	0,071428571	0,005102041
6	-13,92857143	194,005102	12	-3,928571429	15,43367347
66	46,07142857	2122,576531	8	-7,928571429	62,8622449
13	-6,928571429	48,00510204	11	-4,928571429	24,29081633

n1 = 14
n2 = 14
Mean(X1) = 19,92857143
Mean(X2) = 15,92857143

Calculation of sp

Σ (X1-Mean(X1))² = 3814,928571
Σ (X2-Mean(X2))² = 2962,928571
ddl = ((n1-1)+(n2-1)) = 26

(Σ (X1-Mean(X1))² + Σ (X2-Mean(X2))²) = 6777,857143
sp² = (Σ (X1-Mean(X1))² + Σ (X1-Mean(X1))² / ddl = 260,6868132

Calculation of the Interval of Confidence

Mean(X1) - Mean(X2) = 4
t.025 (for current ddl) = 2,06
sp = 16,14579862
(1/n1) = 0,071428571
(1/n2) = 0,071428571
(1/n1)+(1/n2) = 0,142857143
SquareRoot((1/n1)+(1/n2)) = 0,377964473
sp*SquareRoot((1/n1)+(1/n2)) = 6,102538268
t.025*sp*SquareRoot((1/n1)+(1/n2)) = 12,57122883

Hence (μ1-μ2) is between -8,571228832 and 16,57122883

Calculation of the Critical Probability

t = (Mean(X1)-Mean(X2)) / (sp*SquareRoot((1/n1)+(1/n2))) = 0,65546496

For the current ddl, the Critical Values Table For the t Distribution of Student gives us :
t < t.25, which means that pc is not < .25

D'.3 Analysis Of The Impact Of The Use Of Video Games

D'.3.1 Scenario 1

Critical Probability Analysis

X1 ≡ persons who never play video games, or who play rarely
X2 ≡ persons who play video games occasionnally, often or daily

X1	(X1-Mean(X1))	(X1-Mean(X1))²	X2	(X2-Mean(X2))	(X2-Mean(X2))²
18	4,166666667	17,36111111	15	-0,863636364	0,745867769
16	2,166666667	4,694444444	19	3,136363636	9,83677686
18	4,166666667	17,36111111	26	10,13636364	102,7458678
14	0,166666667	0,027777778	5	-10,86363636	118,018595
10	-3,833333333	14,69444444	8	-7,863636364	61,83677686
7	-6,833333333	46,69444444	18	2,136363636	4,564049587
			25	9,136363636	83,4731405
			23	7,136363636	50,92768595
			5	-10,86363636	118,018595
			15	-0,863636364	0,745867769
			9	-6,863636364	47,10950413
			10	-5,863636364	34,3822314
			47	31,13636364	969,4731405
			16	0,136363636	0,018595041
			17	1,136363636	1,291322314
			11	-4,863636364	23,65495868
			15	-0,863636364	0,745867769
			10	-5,863636364	34,3822314
			22	6,136363636	37,65495868
			11	-4,863636364	23,65495868
			11	-4,863636364	23,65495868
			11	-4,863636364	23,65495868

n1 = 6
n2 = 22
Mean(X1) = 13,83333333
Mean(X2) = 15,86363636

Calculation of sp

Σ (X1-Mean(X1))² = 100,8333333
Σ (X2-Mean(X2))² = 1770,590909
ddl = ((n1-1)+(n2-1)) = 26

(Σ (X1-Mean(X1))² + Σ (X2-Mean(X2))²) = 1871,424242
sp² = (Σ (X1-Mean(X1))² + Σ (X1-Mean(X1))² / ddl = 71,97785548

Calculation of the Interval of Confidence

Mean(X1) - Mean(X2) = 2,03030303
t.025 (for current ddl) = 2,06

sp =	8,483976395
(1/n1) =	0,166666667
(1/n2) =	0,045454545
(1/n1)+(1/n2) =	0,212121212
SquareRoot((1/n1)+(1/n2)) =	0,460566186
sp*SquareRoot((1/n1)+(1/n2)) =	3,907432655
t.025*sp*SquareRoot((1/n1)+(1/n2)) =	8,049311268

Hence ($\mu 1-\mu 2$) is between	-6,019008238 and	10,0796143

Calculation of the Critical Probability

t = (Mean(X1)-Mean(X2)) / (sp*SquareRoot((1/n1)+(1/n2))) = 0,519600262

For the current ddl, the Critical Values Table For the t Distribution of Student gives us :
t < t.25, which means that pc is not < .25

D'.3.2 Scenario 2

Scenario 2a : Critical Probability Analysis

X1 ≡ persons who never play video games, or who play rarely
X2 ≡ persons who play video games occasionnally, often or daily

X1	(X1-Mean(X1))	$(X1\text{-Mean}(X1))^2$	X2	(X2-Mean(X2))	$(X2\text{-Mean}(X2))^2$
12	-2,166666667	4,694444444	10	-2,590909091	6,712809917
14	-0,166666667	0,027777778	9	-3,590909091	12,8946281
15	0,833333333	0,694444444	20	7,409090909	54,8946281
21	6,833333333	46,69444444	7	-5,590909091	31,25826446
13	-1,166666667	1,361111111	6	-6,590909091	43,44008264
10	-4,166666667	17,36111111	10	-2,590909091	6,712809917
			12	-0,590909091	0,349173554
			15	2,409090909	5,803719008
			6	-6,590909091	43,44008264
			8	-4,590909091	21,07644628
			6	-6,590909091	43,44008264
			9	-3,590909091	12,8946281
			14	1,409090909	1,98553719
			8	-4,590909091	21,07644628
			12	-0,590909091	0,349173554
			11	-1,590909091	2,530991736
			60	47,40909091	2247,621901
			15	2,409090909	5,803719008
			10	-2,590909091	6,712809917
			11	-1,590909091	2,530991736
			10	-2,590909091	6,712809917
			8	-4,590909091	21,07644628

n1 = 6
n2 = 22
Mean(X1) = 14,16666667
Mean(X2) = 12,59090909

Calculation of sp

$\Sigma\, (X1\text{-Mean}(X1))^2 =$ 70,83333333
$\Sigma\, (X2\text{-Mean}(X2))^2 =$ 2599,318182
ddl = ((n1-1)+(n2-1)) = 26

$(\Sigma\, (X1\text{-Mean}(X1))^2 + \Sigma\, (X2\text{-Mean}(X2))^2) =$ 2670,151515
$sp^2 = (\Sigma\, (X1\text{-Mean}(X1))^2 + \Sigma\, (X1\text{-Mean}(X1))^2 / ddl =$ 102,6981352

Calculation of the Interval of Confidence

Mean(X1) - Mean(X2) = 1,575757576
t.025 (for current ddl) = 2,06
sp = 10,13400884
(1/n1) = 0,166666667

(1/n2) =	0,045454545
(1/n1)+(1/n2) =	0,212121212
SquareRoot((1/n1)+(1/n2)) =	0,460566186
sp*SquareRoot((1/n1)+(1/n2)) =	4,667381806
t.025*sp*SquareRoot((1/n1)+(1/n2)) =	9,61480652

Hence ($\mu 1$-$\mu 2$) is between -8,039048944 and 11,1905641

Calculation of the Critical Probability

t = (Mean(X1)-Mean(X2)) / (sp*SquareRoot((1/n1)+(1/n2))) = 0,337610601

For the current ddl, the Critical Values Table For the t Distribution of Student gives us :
$t < t.25$, which means that pc is not $< .25$

Scenario 2b : Critical Probability Analysis

X1 ≡ persons who never play video games, or who play rarely
X2 ≡ persons who play video games occasionnally, often or daily

X1	(X1-Mean(X1))	(X1-Mean(X1))²	X2	(X2-Mean(X2))	(X2-Mean(X2))²
22	10,5	110,25	12	0,454545455	0,20661157
10	-1,5	2,25	12	0,454545455	0,20661157
9	-2,5	6,25	10	-1,545454545	2,388429752
11	-0,5	0,25	5	-6,545454545	42,84297521
11	-0,5	0,25	9	-2,545454545	6,479338843
6	-5,5	30,25	14	2,454545455	6,024793388
			17	5,454545455	29,75206612
			18	6,454545455	41,66115702
			9	-2,545454545	6,479338843
			14	2,454545455	6,024793388
			7	-4,545454545	20,66115702
			11	-0,545454545	0,297520661
			34	22,45454545	504,2066116
			8	-3,545454545	12,57024793
			6	-5,545454545	30,75206612
			7	-4,545454545	20,66115702
			11	-0,545454545	0,297520661
			9	-2,545454545	6,479338843
			10	-1,545454545	2,388429752
			11	-0,545454545	0,297520661
			12	0,454545455	0,20661157
			8	-3,545454545	12,57024793

n1 = 6
n2 = 22
Mean(X1) = 11,5
Mean(X2) = 11,54545455

Calculation of sp

Σ (X1-Mean(X1))² = 149,5
Σ (X2-Mean(X2))² = 753,4545455
ddl = ((n1-1)+(n2-1)) = 26

(Σ (X1-Mean(X1))² + Σ (X2-Mean(X2))²) = 902,9545455
sp² = (Σ (X1-Mean(X1))² + Σ (X1-Mean(X1))² / ddl = 34,72902098

Calculation of the Interval of Confidence

Mean(X1) - Mean(X2) = 0,045454545
t.025 (for current ddl) = 2,06
sp = 5,893133375
(1/n1) = 0,166666667
(1/n2) = 0,045454545
(1/n1)+(1/n2) = 0,212121212

SquareRoot((1/n1)+(1/n2)) = 0,460566186
sp*SquareRoot((1/n1)+(1/n2)) = 2,714177965
t.025*sp*SquareRoot((1/n1)+(1/n2)) = 5,591206608

Hence (μ1-μ2) is between -5,545752063 and 5,636661153

Calculation of the Critical Probability

t = (Mean(X1)-Mean(X2)) / (sp*SquareRoot((1/n1)+(1/n2))) = 0,016747076

For the current ddl, the Critical Values Table For the t Distribution of Student gives us :
t < t.25, which means that pc is not < .25

D'.3.3 Scenario 3

Scenario 3a : Critical Probability Analysis

X1 ≡ persons who never play video games, or who play rarely
X2 ≡ persons who play video games occasionnally, often or daily

X1	(X1-Mean(X1))	(X1-Mean(X1))²	X2	(X2-Mean(X2))	(X2-Mean(X2))²
59	34,16666667	1167,361111	16	-8,727272727	76,16528926
14	-10,83333333	117,3611111	49	24,27272727	589,1652893
29	4,166666667	17,36111111	45	20,27272727	410,9834711
12	-12,83333333	164,6944444	28	3,272727273	10,7107438
16	-8,833333333	78,02777778	21	-3,727272727	13,89256198
19	-5,833333333	34,02777778	25	0,272727273	0,074380165
			21	-3,727272727	13,89256198
			23	-1,727272727	2,983471074
			24	-0,727272727	0,52892562
			16	-8,727272727	76,16528926
			33	8,272727273	68,43801653
			21	-3,727272727	13,89256198
			29	4,272727273	18,25619835
			11	-13,72727273	188,4380165
			32	7,272727273	52,89256198
			27	2,272727273	5,165289256
			33	8,272727273	68,43801653
			21	-3,727272727	13,89256198
			10	-14,72727273	216,892562
			14	-10,72727273	115,0743802
			21	-3,727272727	13,89256198
			24	-0,727272727	0,52892562

n1 =	6
n2 =	22
Mean(X1) =	24,83333333
Mean(X2) =	24,72727273

Calculation of sp

Σ (X1-Mean(X1))² =	1578,833333
Σ (X2-Mean(X2))² =	1970,363636
ddl = ((n1-1)+(n2-1)) =	26

(Σ (X1-Mean(X1))² + Σ (X2-Mean(X2))²) =	3549,19697
sp² = (Σ (X1-Mean(X1))² + Σ (X1-Mean(X1))² / ddl =	136,5075758

Calculation of the Interval of Confidence

Mean(X1) - Mean(X2) =	0,106060606
t.025 (for current ddl) =	2,06
sp =	11,68364565
(1/n1) =	0,166666667

(1/n2) =	0,045454545
(1/n1)+(1/n2) =	0,212121212
SquareRoot((1/n1)+(1/n2)) =	0,460566186
sp*SquareRoot((1/n1)+(1/n2)) =	5,381092123
t.025*sp*SquareRoot((1/n1)+(1/n2)) =	11,08504977

Hence (μ1-μ2) is between -10,97898917 and 11,19111038

Calculation of the Critical Probability

t = (Mean(X1)-Mean(X2)) / (sp*SquareRoot((1/n1)+(1/n2))) = 0,019709866

For the current ddl, the Critical Values Table For the t Distribution of Student gives us :
t < t.25, which means that pc is not < .25

Scenario 3b : Critical Probability Analysis

X1 ≡ persons who never play video games, or who play rarely
X2 ≡ persons who play video games occasionnally, often or daily

X1	(X1-Mean(X1))	$(X1\text{-}Mean(X1))^2$	X2	(X2-Mean(X2))	$(X2\text{-}Mean(X2))^2$
24	-0,5	0,25	15	-9,181818182	84,30578512
31	6,5	42,25	32	7,818181818	61,12396694
16	-8,5	72,25	23	-1,181818182	1,396694215
46	21,5	462,25	11	-13,18181818	173,7603306
22	-2,5	6,25	26	1,818181818	3,305785124
8	-16,5	272,25	15	-9,181818182	84,30578512
			30	5,818181818	33,85123967
			23	-1,181818182	1,396694215
			8	-16,18181818	261,8512397
			23	-1,181818182	1,396694215
			7	-17,18181818	295,214876
			58	33,81818182	1143,669421
			6	-18,18181818	330,5785124
			30	5,818181818	33,85123967
			13	-11,18181818	125,0330579
			17	-7,181818182	51,5785124
			25	0,818181818	0,669421488
			17	-7,181818182	51,5785124
			75	50,81818182	2582,487603
			32	7,818181818	61,12396694
			34	9,818181818	96,39669421
			12	-12,18181818	148,3966942

n1 = 6
n2 = 22
Mean(X1) = 24,5
Mean(X2) = 24,18181818

Calculation of sp

$\Sigma (X1\text{-}Mean(X1))^2 =$ 855,5
$\Sigma (X2\text{-}Mean(X2))^2 =$ 5627,272727
ddl = ((n1-1)+(n2-1)) = 26

$(\Sigma (X1\text{-}Mean(X1))^2 + \Sigma (X2\text{-}Mean(X2))^2) =$ 6482,772727
$sp^2 = (\Sigma (X1\text{-}Mean(X1))^2 + \Sigma (X1\text{-}Mean(X1))^2 / ddl =$ 249,3374126

Calculation of the Interval of Confidence

Mean(X1) - Mean(X2) = 0,318181818
t.025 (for current ddl) = 2,06
sp = 15,79042155
(1/n1) = 0,166666667
(1/n2) = 0,045454545
(1/n1)+(1/n2) = 0,212121212

SquareRoot((1/n1)+(1/n2)) =	0,460566186
sp*SquareRoot((1/n1)+(1/n2)) =	7,272534234
t.025*sp*SquareRoot((1/n1)+(1/n2)) =	14,98142052

Hence (μ1-μ2) is between -14,6632387 and 15,29960234

Calculation of the Critical Probability

t = (Mean(X1)-Mean(X2)) / (sp*SquareRoot((1/n1)+(1/n2))) = 0,043751161

For the current ddl, the Critical Values Table For the t Distribution of Student gives us :
t < t.25, which means that pc is not < .25

Scenario 3c : Critical Probability Analysis

X1 ≡ persons who never play video games, or who play rarely
X2 ≡ persons who play video games occasionnally, often or daily

X1	(X1-Mean(X1))	(X1-Mean(X1))²	X2	(X2-Mean(X2))	(X2-Mean(X2))²
7	-4	16	17	-2,818181818	7,94214876
9	-2	4	5	-14,81818182	219,5785124
15	4	16	18	-1,818181818	3,305785124
16	5	25	13	-6,818181818	46,48760331
8	-3	9	12	-7,818181818	61,12396694
11	0	0	12	-7,818181818	61,12396694
			17	-2,818181818	7,94214876
			20	0,181818182	0,033057851
			13	-6,818181818	46,48760331
			50	30,18181818	910,9421488
			6	-13,81818182	190,9421488
			66	46,18181818	2132,760331
			56	36,18181818	1309,123967
			11	-8,818181818	77,76033058
			7	-12,81818182	164,3057851
			45	25,18181818	634,1239669
			12	-7,818181818	61,12396694
			7	-12,81818182	164,3057851
			20	0,181818182	0,033057851
			6	-13,81818182	190,9421488
			15	-4,818181818	23,21487603
			8	-11,81818182	139,6694215

n1 = 6
n2 = 22
Mean(X1) = 11
Mean(X2) = 19,81818182

Calculation of sp

Σ (X1-Mean(X1))² = 70
Σ (X2-Mean(X2))² = 6453,272727
ddl = ((n1-1)+(n2-1)) = 26

(Σ (X1-Mean(X1))² + Σ (X2-Mean(X2))²) = 6523,272727
sp² = (Σ (X1-Mean(X1))² + Σ (X1-Mean(X1))² / ddl = 250,8951049

Calculation of the Interval of Confidence

Mean(X1) - Mean(X2) = 8,818181818
t.025 (for current ddl) = 2,06
sp = 15,83966871
(1/n1) = 0,166666667
(1/n2) = 0,045454545

(1/n1)+(1/n2) =	0,212121212
SquareRoot((1/n1)+(1/n2)) =	0,460566186
sp*SquareRoot((1/n1)+(1/n2)) =	7,295215814
t.025*sp*SquareRoot((1/n1)+(1/n2)) =	15,02814458

Hence (μ1-μ2) is between -6,209962758 and 23,84632639

Calculation of the Critical Probability

t = (Mean(X1)-Mean(X2)) / (sp*SquareRoot((1/n1)+(1/n2))) = 1,208762296

For the current ddl, the Critical Values Table For the t Distribution of Student gives us :
t > t.25, which means that pc is < .25

D'.4 Analysis Of The Impact Of Video Games For Novice PDA Users

D'.4.1 Scenario 1

Critical Probability Analysis

This analysis concerns the 20 persons who never used a PDA before.

X1 ≡ persons who never play video games, or who play rarely
X2 ≡ persons who play video games occasionnally, often or daily

X1	(X1-Mean(X1))	$(X1\text{-Mean}(X1))^2$	X2	(X2-Mean(X2))	$(X2\text{-Mean}(X2))^2$
18	1,5	2,25	15	-3,4375	11,81640625
16	-0,5	0,25	19	0,5625	0,31640625
18	1,5	2,25	26	7,5625	57,19140625
14	-2,5	6,25	18	-0,4375	0,19140625
			25	6,5625	43,06640625
			23	4,5625	20,81640625
			5	-13,4375	180,5664063
			15	-3,4375	11,81640625
			47	28,5625	815,8164063
			16	-2,4375	5,94140625
			17	-1,4375	2,06640625
			15	-3,4375	11,81640625
			10	-8,4375	71,19140625
			22	3,5625	12,69140625
			11	-7,4375	55,31640625
			11	-7,4375	55,31640625

n1 = 4
n2 = 16
Mean(X1) = 16,5
Mean(X2) = 18,4375

Calculation of sp

$\Sigma (X1\text{-Mean}(X1))^2 =$ 11
$\Sigma (X2\text{-Mean}(X2))^2 =$ 1355,9375
ddl = ((n1-1)+(n2-1)) = 18

$(\Sigma (X1\text{-Mean}(X1))^2 + \Sigma (X2\text{-Mean}(X2))^2) =$ 1366,9375
$sp^2 = (\Sigma (X1\text{-Mean}(X1))^2 + \Sigma (X1\text{-Mean}(X1))^2 / ddl =$ 75,94097222

Calculation of the Interval of Confidence

Mean(X1) - Mean(X2) = 1,9375
t.025 (for current ddl) = 2,1
sp = 8,714411754
(1/n1) = 0,25
(1/n2) = 0,0625
(1/n1)+(1/n2) = 0,3125

SquareRoot((1/n1)+(1/n2)) =	0,559016994
sp*SquareRoot((1/n1)+(1/n2)) =	4,871504267
t.025*sp*SquareRoot((1/n1)+(1/n2)) =	10,23015896

Hence (μ1-μ2) is between -8,29265896 and 12,16765896

Calculation of the Critical Probability

t = (Mean(X1)-Mean(X2)) / (sp*SquareRoot((1/n1)+(1/n2))) = 0,397721093

For the current ddl, the Critical Values Table For the t Distribution of Student gives us :
t < t.25, which means that pc is not < .25

D'.4.2 Scenario 2

Scenario 2a : Critical Probability Analysis

This analysis concerns the 20 persons who never used a PDA before.

X1 ≡ persons who never play video games, or who play rarely
X2 ≡ persons who play video games occasionnally, often or daily

X1	(X1-Mean(X1))	$(X1\text{-}Mean(X1))^2$	X2	(X2-Mean(X2))	$(X2\text{-}Mean(X2))^2$
12	-3,5	12,25	10	-4,375	19,140625
14	-1,5	2,25	9	-5,375	28,890625
15	-0,5	0,25	20	5,625	31,640625
21	5,5	30,25	10	-4,375	19,140625
			12	-2,375	5,640625
			15	0,625	0,390625
			6	-8,375	70,140625
			8	-6,375	40,640625
			14	-0,375	0,140625
			8	-6,375	40,640625
			12	-2,375	5,640625
			60	45,625	2081,640625
			15	0,625	0,390625
			10	-4,375	19,140625
			11	-3,375	11,390625
			10	-4,375	19,140625

n1 = 4
n2 = 16
Mean(X1) = 15,5
Mean(X2) = 14,375

Calculation of sp

$\Sigma (X1\text{-}Mean(X1))^2 =$ 45
$\Sigma (X2\text{-}Mean(X2))^2 =$ 2393,75
ddl = ((n1-1)+(n2-1)) = 18

$(\Sigma (X1\text{-}Mean(X1))^2 + \Sigma (X2\text{-}Mean(X2))^2) =$ 2438,75
$sp^2 = (\Sigma (X1\text{-}Mean(X1))^2 + \Sigma (X1\text{-}Mean(X1))^2 / ddl =$ 135,4861111

Calculation of the Interval of Confidence

Mean(X1) - Mean(X2) = 1,125
t.025 (for current ddl) = 2,1
sp = 11,63985013
(1/n1) = 0,25
(1/n2) = 0,0625
(1/n1)+(1/n2) = 0,3125
SquareRoot((1/n1)+(1/n2)) = 0,559016994
sp*SquareRoot((1/n1)+(1/n2)) = 6,506874036
t.025*sp*SquareRoot((1/n1)+(1/n2)) = 13,66443548

Hence ($\mu 1$-$\mu 2$) is between -12,53943548 and 14,78943548

Calculation of the Critical Probability

t = (Mean(X1)-Mean(X2)) / (sp*SquareRoot((1/n1)+(1/n2))) = 0,17289408

For the current ddl, the Critical Values Table For the t Distribution of Student gives us :
$t < t.25$, which means that pc is not < .25

Scenario 2b : Critical Probability Analysis

This analysis concerns the 20 persons who never used a PDA before.

X1 ≡ persons who never play video games, or who play rarely
X2 ≡ persons who play video games occasionnally, often or daily

X1	(X1-Mean(X1))	$(X1-Mean(X1))^2$	X2	(X2-Mean(X2))	$(X2-Mean(X2))^2$
22	9	81	12	-0,9375	0,87890625
10	-3	9	12	-0,9375	0,87890625
9	-4	16	10	-2,9375	8,62890625
11	-2	4	14	1,0625	1,12890625
			17	4,0625	16,50390625
			18	5,0625	25,62890625
			9	-3,9375	15,50390625
			14	1,0625	1,12890625
			34	21,0625	443,6289063
			8	-4,9375	24,37890625
			6	-6,9375	48,12890625
			11	-1,9375	3,75390625
			9	-3,9375	15,50390625
			10	-2,9375	8,62890625
			11	-1,9375	3,75390625
			12	-0,9375	0,87890625

n1 = 4
n2 = 16
Mean(X1) = 13
Mean(X2) = 12,9375

Calculation of sp

Σ $(X1-Mean(X1))^2$ = 110
Σ $(X2-Mean(X2))^2$ = 618,9375
ddl = ((n1-1)+(n2-1)) = 18

(Σ $(X1-Mean(X1))^2$ + Σ $(X2-Mean(X2))^2$) = 728,9375
sp^2 = (Σ $(X1-Mean(X1))^2$ + Σ $(X1-Mean(X1))^2$ / ddl = 40,49652778

Calculation of the Interval of Confidence

Mean(X1) - Mean(X2) = 0,0625
t.025 (for current ddl) = 2,1
sp = 6,363688221
(1/n1) = 0,25
(1/n2) = 0,0625
(1/n1)+(1/n2) = 0,3125
SquareRoot((1/n1)+(1/n2)) = 0,559016994
sp*SquareRoot((1/n1)+(1/n2)) = 3,557409863
t.025*sp*SquareRoot((1/n1)+(1/n2)) = 7,470560711

Hence (μ1-μ2) is between -7,408060711 and 7,533060711

Calculation of the Critical Probability

t = (Mean(X1)-Mean(X2)) / (sp*SquareRoot((1/n1)+(1/n2))) = 0,017568962

For the current ddl, the Critical Values Table For the t Distribution of Student gives us :
t < t.25, which means that pc is not < .25

D'.4.3 Scenario 3

Scenario 3a : Critical Probability Analysis

This analysis concerns the 20 persons who never used a PDA before.

X1 ≡ persons who never play video games, or who play rarely
X2 ≡ persons who play video games occasionnally, often or daily

X1	(X1-Mean(X1))	(X1-Mean(X1))²	X2	(X2-Mean(X2))	(X2-Mean(X2))²
59	30,5	930,25	16	-8,375	70,140625
14	-14,5	210,25	49	24,625	606,390625
29	0,5	0,25	45	20,625	425,390625
12	-16,5	272,25	25	0,625	0,390625
			21	-3,375	11,390625
			23	-1,375	1,890625
			24	-0,375	0,140625
			16	-8,375	70,140625
			29	4,625	21,390625
			11	-13,375	178,890625
			32	7,625	58,140625
			33	8,625	74,390625
			21	-3,375	11,390625
			10	-14,375	206,640625
			14	-10,375	107,640625
			21	-3,375	11,390625

n1 = 4
n2 = 16
Mean(X1) = 28,5
Mean(X2) = 24,375

Calculation of sp

Σ (X1-Mean(X1))² = 1413
Σ (X2-Mean(X2))² = 1855,75
ddl = ((n1-1)+(n2-1)) = 18

(Σ (X1-Mean(X1))² + Σ (X2-Mean(X2))²) = 3268,75
sp² = (Σ (X1-Mean(X1))² + Σ (X1-Mean(X1))² / ddl = 181,5972222

Calculation of the Interval of Confidence

Mean(X1) - Mean(X2) = 4,125
t.025 (for current ddl) = 2,1
sp = 13,47580136
(1/n1) = 0,25
(1/n2) = 0,0625
(1/n1)+(1/n2) = 0,3125
SquareRoot((1/n1)+(1/n2)) = 0,559016994
sp*SquareRoot((1/n1)+(1/n2)) = 7,533201972
t.025*sp*SquareRoot((1/n1)+(1/n2)) = 15,81972414

Hence ($\mu 1$-$\mu 2$) is between -11,69472414 and 19,94472414

Calculation of the Critical Probability

t = (Mean(X1)-Mean(X2)) / (sp*SquareRoot((1/n1)+(1/n2))) = 0,54757592

For the current ddl, the Critical Values Table For the t Distribution of Student gives us :
t < t.25, which means that pc is not < .25

Scenario 3b : Critical Probability Analysis

This analysis concerns the 20 persons who never used a PDA before.

X1 ≡ persons who never play video games, or who play rarely
X2 ≡ persons who play video games occasionnally, often or daily

X1	(X1-Mean(X1))	$(X1\text{-}Mean(X1))^2$	X2	(X2-Mean(X2))	$(X2\text{-}Mean(X2))^2$
24	-5,25	27,5625	15	-10,0625	101,2539063
31	1,75	3,0625	32	6,9375	48,12890625
16	-13,25	175,5625	23	-2,0625	4,25390625
46	16,75	280,5625	15	-10,0625	101,2539063
			30	4,9375	24,37890625
			23	-2,0625	4,25390625
			8	-17,0625	291,1289063
			23	-2,0625	4,25390625
			6	-19,0625	363,3789063
			30	4,9375	24,37890625
			13	-12,0625	145,5039063
			25	-0,0625	0,00390625
			17	-8,0625	65,00390625
			75	49,9375	2493,753906
			32	6,9375	48,12890625
			34	8,9375	79,87890625

n1 = 4
n2 = 16
Mean(X1) = 29,25
Mean(X2) = 25,0625

Calculation of sp

Σ (X1-Mean(X1))² = 486,75
Σ (X2-Mean(X2))² = 3798,9375
ddl = ((n1-1)+(n2-1)) = 18

(Σ (X1-Mean(X1))² + Σ (X2-Mean(X2))²) = 4285,6875
sp² = (Σ (X1-Mean(X1))² + Σ (X1-Mean(X1))² / ddl = 238,09375

Calculation of the Interval of Confidence

Mean(X1) - Mean(X2) = 4,1875
t.025 (for current ddl) = 2,1
sp = 15,43028678
(1/n1) = 0,25
(1/n2) = 0,0625
(1/n1)+(1/n2) = 0,3125
SquareRoot((1/n1)+(1/n2)) = 0,559016994
sp*SquareRoot((1/n1)+(1/n2)) = 8,625792536
t.025*sp*SquareRoot((1/n1)+(1/n2)) = 18,11416433

Hence (μ1-μ2) is between -13,92666433 and 22,30166433

Calculation of the Critical Probability

t = (Mean(X1)-Mean(X2)) / (sp*SquareRoot((1/n1)+(1/n2))) = 0,485462638

For the current ddl, the Critical Values Table For the t Distribution of Student gives us :
$t < t.25$, which means that pc is not $< .25$

Scenario 3c : Critical Probability Analysis

This analysis concerns the 20 persons who never used a PDA before.

X1 ≡ persons who never play video games, or who play rarely
X2 ≡ persons who play video games occasionnally, often or daily

X1	(X1-Mean(X1))	(X1-Mean(X1))²	X2	(X2-Mean(X2))	(X2-Mean(X2))²
7	-4,75	22,5625	17	-0,875	0,765625
9	-2,75	7,5625	5	-12,875	165,765625
15	3,25	10,5625	18	0,125	0,015625
16	4,25	18,0625	12	-5,875	34,515625
			17	-0,875	0,765625
			20	2,125	4,515625
			13	-4,875	23,765625
			50	32,125	1032,015625
			56	38,125	1453,515625
			11	-6,875	47,265625
			7	-10,875	118,265625
			12	-5,875	34,515625
			7	-10,875	118,265625
			20	2,125	4,515625
			6	-11,875	141,015625
			15	-2,875	8,265625

n1 = 4
n2 = 16
Mean(X1) = 11,75
Mean(X2) = 17,875

Calculation of sp

Σ (X1-Mean(X1))² = 58,75
Σ (X2-Mean(X2))² = 3187,75
ddl = ((n1-1)+(n2-1)) = 18

(Σ (X1-Mean(X1))² + Σ (X2-Mean(X2))²) = 3246,5
sp² = (Σ (X1-Mean(X1))² + Σ (X1-Mean(X1))² / ddl = 180,3611111

Calculation of the Interval of Confidence

Mean(X1) - Mean(X2) = 6,125
t.025 (for current ddl) = 2,1
sp = 13,42985894
(1/n1) = 0,25
(1/n2) = 0,0625
(1/n1)+(1/n2) = 0,3125
SquareRoot((1/n1)+(1/n2)) = 0,559016994
sp*SquareRoot((1/n1)+(1/n2)) = 7,507519379
t.025*sp*SquareRoot((1/n1)+(1/n2)) = 15,7657907

Hence (μ1-μ2) is between -9,640790695 and 21,8907907

Calculation of the Critical Probability

t = (Mean(X1)-Mean(X2)) / (sp*SquareRoot((1/n1)+(1/n2))) = 0,81584871

For the current ddl, the Critical Values Table For the t Distribution of Student gives us :
t > t.25, which means that pc is < .25