

## THESIS / THÈSE

### MASTER EN SCIENCES INFORMATIQUES

Initiation à la programmation à l'aide d'un jeu multijoueur destiné à l'adulte débutant

TOUNE, Fabien

*Award date:*  
2022

*Awarding institution:*  
Universite de Namur

[Link to publication](#)

#### General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

#### Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

UNIVERSITÉ DE NAMUR  
Faculté d'informatique  
Année académique 2021-2022

**Initiation à la programmation à l'aide  
d'un jeu multijoueur destiné à l'adulte  
débutant**

Fabien Toune



Promoteur : \_\_\_\_\_ (Signature pour approbation du dépôt - REE art. 40)  
Benoît Vanderose

Co-promotrice : Julie Henry

Mémoire présenté en vue de l'obtention du grade de  
Master en Sciences Informatiques.

## Résumé

L'objet de ce travail est d'évaluer l'utilité d'un jeu vidéo multijoueur dans un cours de promotion sociale portant sur l'initiation à la programmation informatique. Plus spécifiquement, un nombre restreint de concepts clés est sélectionné et un « serious game » est élaboré afin de pouvoir conduire une expérience auprès d'une classe d'étudiants adultes au niveau secondaire. Pour ce faire, cinq séances de cours de quatre heures sont d'abord dispensées afin d'introduire les concepts sélectionnés de manière classique. Les étudiants sont ensuite invités à se confronter les uns aux autres lors de plusieurs parties de jeu. À l'issue de quoi un ensemble de résultats est collecté via des entretiens semi-directifs réalisés auprès de sept élèves puis analysé de manière qualitative. Au travers des perceptions du jeu selon trois dimensions (andragogie, gameplay et efficacité didactique), nous critiquons les éléments de design de notre prototype et évaluons leurs impacts sur l'apprentissage des rudiments de la programmation. Notre conclusion est qu'il est possible de mêler efficacement pratique ludique et finalité sérieuse même si l'étendue réelle des bénéfices obtenus reste discutable.

**Mots clés :** *initiation à la programmation, pensée informatique, jeux sérieux éducatifs, andragogie, apprentissage par le jeu.*

## REMERCIEMENTS

Tout d'abord, je tiens ici à remercier sincèrement le promoteur de ce mémoire, Monsieur Benoît Vanderose, qui a accepté sans réserve de s'engager à me faire confiance.

Ensuite, je n'aurai jamais assez de mots pour témoigner de ma gratitude envers la co-promotrice de ce travail, Madame Julie Henry. Elle a cru en moi contre vents et marées (ce qui reste un mystère à mes yeux) pour me soutenir jusqu'à la ligne d'arrivée. Si l'abnégation, la gentillesse et l'efficacité avaient un visage, ça serait sûrement le sien !

Merci aussi au professeur qui a initié ce projet, a manifesté tant de patience à mon égard et m'a fait découvrir les joies de l'ingénierie des architectures logicielles, Monsieur Vincent Englebert.

Merci à tout le corps professoral de l'Université, qui a nourri et entretenu ma passion pour l'informatique durant toutes ces années.

Merci à tout le personnel administratif, qui est toujours là, avec sa sympathie, pour huiler les rouages et faire que ça roule.

Merci à ma famille, ma femme Corine et mes enfants Chani et Pierre, qui ont été d'un soutien et d'une patience indéfectibles. Il n'ont cessé de m'apporter leur amour, supportant tantôt mes absences, tantôt ma présence !

Merci à mon ami Francis. Je sais à quel point ce travail lui tient à cœur, et c'est aussi grâce à lui et à nos rêveries mathématiques que je n'ai pas abandonné.

Merci enfin à tous ceux qui m'ont apporté leur aide, chacun à sa manière. Je ne peux tous les énumérer : ils sont vraiment trop nombreux. J'ai une affection particulière pour chacun d'entre eux — je sais qu'ils le savent et se reconnaîtront.

Ce mémoire témoigne d'efforts et de qualités humaines incroyables fournies par tellement de personnes que j'ai l'impression que ce travail leur appartient pour beaucoup. Alors encore merci à tous, du fond du cœur !

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>5</b>
1.1	Culture numérique . . . . .	5
1.2	Programmer pour penser ou penser pour programmer . . . . .	6
1.3	La place de la formation pour adultes . . . . .	6
1.4	Questionnement et objet de notre recherche . . . . .	7
1.5	Structure de ce mémoire . . . . .	8
<b>2</b>	<b>État de l’art</b>	<b>9</b>
2.1	Initiation à la programmation informatique . . . . .	9
2.1.1	Contours et structure du domaine . . . . .	9
2.1.2	L’approche fondamentaliste de Peter Denning . . . . .	10
2.1.3	Le niveau secondaire supérieur de Gilles Dowek . . . . .	12
2.1.4	L’approche anglo-saxonne . . . . .	13
2.1.5	En Belgique . . . . .	15
2.1.6	La pensée informatique de Jeanette Wing . . . . .	18
2.1.7	En résumé . . . . .	18
2.2	Enseigner aux adultes - L’andragogie . . . . .	19
2.2.1	Enseigner aux adultes . . . . .	20
2.2.2	La reprise d’études . . . . .	21
2.2.3	Le modèle « expectancy value » . . . . .	21
2.2.4	Les caractéristiques des motifs d’engagement . . . . .	22
2.2.5	Moteurs d’apprentissage chez les adultes peu qualifiés . . . . .	22
2.2.6	Étymologie et principes de l’andragogie . . . . .	22
2.3	L’enseignement par le jeu . . . . .	25
2.3.1	Étymologie . . . . .	25
2.3.2	Définition . . . . .	25
2.3.3	L’industrie du SG . . . . .	26
2.3.4	Classification des jeux sérieux . . . . .	27
2.3.5	Le modèle MDA . . . . .	28
2.3.6	La conception des SG . . . . .	32
2.3.7	Les jeux sérieux éducatifs . . . . .	32
<b>3</b>	<b>Méthodologie</b>	<b>40</b>
3.1	Le design expérimental d’environnements d’apprentissage . . . . .	40
3.2	Étapes de la recherche . . . . .	41
3.2.1	Identification du problème à résoudre . . . . .	41
3.2.2	Analyse de la situation . . . . .	42
3.2.3	Prospections préliminaires . . . . .	42

---

3.2.4	Proposition d'une solution innovante . . . . .	46
3.3	Déroulement de l'expérience . . . . .	46
3.3.1	Introduction à l'informatique . . . . .	47
3.3.2	Algorithmique de base . . . . .	47
3.3.3	Premiers programmes rédigés textuellement . . . . .	48
3.3.4	Pratique du jeu . . . . .	49
3.4	Collecte et traitement des résultats . . . . .	49
<b>4</b>	<b>Résultats</b>	<b>53</b>
4.1	Description du jeu . . . . .	53
4.1.1	Exigences techniques . . . . .	53
4.1.2	Exigences conceptuelles . . . . .	53
4.1.3	Implémentation . . . . .	54
4.2	Données recueillies . . . . .	60
4.2.1	Compétition, collaboration et savoirs échangés . . . . .	61
4.2.2	Caractéristiques du jeu . . . . .	61
4.2.3	Appréciation des apprentissages techniques . . . . .	62
4.3	Discussion des résultats . . . . .	63
4.3.1	Le public . . . . .	63
4.3.2	Concernant la conception du jeu . . . . .	64
4.3.3	Efficacité didactique . . . . .	65
<b>5</b>	<b>Conclusion</b>	<b>66</b>
<b>A</b>	<b>Appendices</b>	<b>68</b>
A.1	Questionnaire . . . . .	68
A.2	Niveaux de jeu . . . . .	70
A.2.1	Niveau 1 . . . . .	70
A.2.2	Niveau 2 . . . . .	70
A.2.3	Niveau 3 . . . . .	71
A.2.4	Niveau 4 . . . . .	72
A.2.5	Niveau 5 . . . . .	73
A.2.6	Niveau 6 . . . . .	74
A.2.7	Niveau 7 . . . . .	75
A.2.8	Niveau 8 . . . . .	76
A.2.9	Niveau 9 . . . . .	77
A.2.10	Niveau 10 . . . . .	78
<b>B</b>	<b>Bibliographie</b>	<b>79</b>

# Chapitre 1

## Introduction

### 1.1 Culture numérique

Bien que le terme de « culture numérique » soit largement véhiculé de nos jours, la notion qu'il recouvre reste ambiguë et manque de définition rigoureuse, probablement en raison des nombreuses perspectives selon lesquelles il est possible de l'évoquer. Se focalisant sur sa diffusion au sein du grand public [Proulx, 2002] définit la culture numérique comme « *un ensemble de valeurs, de connaissances et de pratiques qui impliquent l'usage d'outils informatisés, mais aussi des pratiques de consommation médiatique et culturelle, de communication et d'expression de soi* ». C'est une évidence : la culture numérique imprègne la société actuelle.

Matériellement, tout d'abord, au travers des ordinateurs, mais aussi via une pléthore croissante de dispositifs, connectés ou non, réputés « intelligents ». Spirituellement, ensuite, pourrait-on presque dire. Internet et les réseaux sociaux influencent fortement l'image que les individus ont d'eux-mêmes, la manière dont ils interagissent et construisent leurs repères et leurs connaissances.

Il s'agit aussi d'une évolution sociétale : il est aujourd'hui impossible d'interagir avec les institutions gouvernementales et privées sans une maîtrise minimale des technologies de l'information et des communications. Les informations personnelles font l'objet d'intenses convoitises. La manière dont les générations présentes et futures vont s'intégrer dans cette culture sera probablement déterminante quant à l'orientation que prendra notre société.

Reflet de celle-ci, le marché de l'emploi évolue rapidement et fondamentalement : l'informatique s'immisce et s'impose dans de nombreux domaines, de nouvelles activités voient le jour. L'apparition et l'évolution de nouvelles technologies induit le remplacement de nombreux travailleurs par des automates, des logiciels. Ce phénomène concernant essentiellement des tâches exigeant moins de qualifications, il participe indéniablement à l'accroissement des clivages sociaux. Parallèlement, la demande en main d'œuvre spécialisée augmente régulièrement alors que le nombre de diplômés sur le marché du travail a tendance à reculer.

Pourtant, en 2017, 61 millions d'adultes âgés de 25 à 64 ans dans l'Union Européenne avaient interrompu leurs études avant d'avoir terminé l'enseignement secondaire supérieur. Approximativement 20 % de la population adulte des pays participant à l'Enquête de l'OCDE sur les compétences des adultes (PIAAC) sont confrontés à des problèmes de lecture, d'écriture et en mathématique. 43 % de la population de l'Union a un niveau insuffisant de compétences numériques et 17 % n'en a pas

du tout. Les données suggèrent également que dès 2022, 54 % de tous les employés nécessiteront une amélioration significative ou une requalification de leurs compétences.

Face à tous ces constats, aider les adultes à s’immerger dans un environnement numérique paraît nécessaire, à fortiori si cela peut aider l’individu à s’épanouir humainement parlant.

## 1.2 Programmer pour penser ou penser pour programmer

Alors qu’il y a quelques années encore l’utilisation du numérique exigeait des connaissances techniques conséquentes, les immenses progrès de l’ergonomie, des interfaces et de leur convivialité a profondément changé la donne. L’informatique en tant que discipline scientifique reste au cœur de la sphère du monde digital, mais le profane n’a plus besoin de savoir « ce qui se passe sous le capot ». On peut dès lors s’interroger sur la pertinence d’un apprentissage de la programmation qui soutiendrait la culture numérique.

Le débat à ce niveau n’est pas clos, nous adhérons cependant au point de vue de Périne Brotcorne selon qui « l’acquisition d’une véritable culture en la matière [de culture numérique] ne peut faire l’économie d’une formation portant sur une compréhension des systèmes informatique qui la rendent possible » [Henry et al., 2017].

Nous en voulons pour preuve l’émergence de la notion de « Pensée Informatique » (CT pour Computational Thinking en anglais). Concept séminal introduit par la chercheuse Jeanette Wing [Wing, 2006], il fait notamment référence à la capacité d’abstraction, à la l’identification de schémas récurrents, à la généralisation de ceux-ci et donc à la capacité de mettre au point des algorithmes [McNicholl, 2018]. Dans la pensée informatique, ces compétences sont considérées comme dépassant le cadre de l’informatique et transposables à de nombreuses situations de la vie quotidienne, professionnelle ou non. Selon [González, 2015], les logiques conditionnelles, itératives ou encore récursive font aussi partie de la pensée informatique.

Bien que distincts, programmation et pensée informatique apparaissent comme fortement liées et susceptibles de se nourrir mutuellement pour améliorer à la fois la pratique du développement et l’adoption des technologies numériques.

## 1.3 La place de la formation pour adultes

L’initiation précoce à la programmation est devenue un enjeu majeur dans l’enseignement scolaire destiné aux jeunes, et nombreuses sont les recherches consacrées à ce sujet. Concernant la formation des adultes, surtout lorsqu’il s’agit d’un public moins qualifié et sorti des cursus classiques, il n’en va pas de même.

En Belgique, l’enseignement officiel de promotion sociale (EPS) s’applique, depuis le décret de 1991 [Communauté Française de Belgique, 1991], à dispenser des cours qui ont pour « finalités générales de :

- concourir à l’épanouissement individuel en promouvant une meilleure insertion professionnelle, sociale et culturelle,



- répondre aux besoins et demandes en formation émanant des entreprises, des administrations, de l’enseignement et d’une manière générale, des milieux socio-économiques et culturels. »

Il vise un public qui n’est en principe plus soumis à l’obligation scolaire (minimum 18 ans) et est donc au service d’adultes de générations, de formations, de disciplines, d’expériences de vie ou professionnelles diverses qui souhaitent acquérir, développer ou actualiser leurs compétences. Les filières d’enseignement pour adultes développées par les établissements organisant l’EPS répondent ainsi à des besoins individuels et collectifs d’initiation, de rattrapage, de qualification, de perfectionnement, de recyclage, de reconversion, de spécialisation et d’épanouissement personnel.

Le public de l’enseignement de promotion sociale peut ainsi être considéré comme un public adulte hétérogène en reprise d’études. En termes de niveau scolaire, et toujours selon la publication ci-dessus, 78 % des unités d’enseignement dispensées le sont au niveau secondaire contre 22 % au niveau supérieur. Bien que l’informatique soit présente depuis les débuts de l’EPS, ce n’est que très récemment, en 2017, qu’une section complète consacrée à l’apprentissage de la programmation est apparue au niveau secondaire. Les prérequis sont volontairement très faibles, n’incluant ni expérience préalable dans le domaine, ni connaissances particulières en mathématiques par exemple.

## 1.4 Questionnement et objet de notre recherche

Dans un monde où la fracture numérique et éducationnelle participe à l’augmentation du clivage social, l’apprentissage de la programmation et le développement de la pensée informatique chez un public d’adultes en situation de réinsertion socio-professionnelle fait sens.

Malheureusement apprendre la programmation est foncièrement ardu, ce qui mène à de forts taux d’échecs et d’abandons, surtout lorsque l’on ne parle plus d’enseignement obligatoire. Confrontés à cette situation, les acteurs du monde de la formation explorent de nouvelles méthodes pour aider leurs étudiants à surmonter les obstacles qu’il rencontrent. L’utilisation de « *Jeux Sérieux* » (*SG pour Serious Game en anglais*) utilisés pour sensibiliser les élèves au « codage » dès leurs plus jeune âge nous amène à questionner la pertinence d’une démarche similaire dans des cours destinés à l’EPS.

Ce qui nous amène à nous poser la question suivante :

« *Quels sont les impacts d’un jeu vidéo multijoueur sur un cours d’initiation à la programmation informatique suivi par des étudiants de promotion sociale ?* »

Plus spécifiquement, nous nous intéressons à ces incidences en matière de didactique, d’andragogie et de gameplay.

Nos recherches ne nous ayant pas permis de trouver de jeu présentant les caractéristiques adéquates pour mener notre recherche, nous avons conçu et implémenté un prototype pour le mettre en œuvre dans une situation d’apprentissage concret. Nous nous proposons de récolter des données issues d’entretiens semi-directifs menés auprès des participant et d’analyser celles-ci pour tenter de valider nos hypothèses et de répondre à notre question de recherche.

Notre jeu consiste en une *bataille d’algorithmes*. Deux protagonistes s’affrontent en réseau, en concevant chacun un programme destiné à piloter un avatar dans une série de labyrinthes. Une fois les programmes rédigés, chaque personnage commence

à exécuter le sien, en tentant d'atteindre son objectif : trouver l'issue du labyrinthe, récolter un maximum d'items, etc. Afin de permettre aux joueurs débutant en programmation de se focaliser sur la logique de leurs procédures, le langage utilisé est de type « blocs ». Un système de communication en cours de partie est implémenté afin d'encourager la collaboration entre joueurs.

Pour tenter de répondre à notre question de recherche, nous partons des trois postulats suivants :

- Un Serious Game (SG) peut se révéler complémentaire aux moyens pédagogiques traditionnels,
- Les dimensions sociales de compétition et de collaboration dans un SG favorisent l'échange de connaissances entre participants,
- La programmation par bloc dans un SG aide à maîtriser les structures de base de la programmation.

## 1.5 Structure de ce mémoire

Alors que cette introduction constitue le **premier chapitre**, le **second chapitre** est dédié à l'état de l'art. Nous avons décidé de découper celui-ci en trois parties distinctes, consacrées aux trois facettes selon lesquelles nous désirons mener notre recherche :

- **L'initiation à la programmation informatique** : où, après avoir délimité et précisé le contour du domaine, nous nous attachons à dégager les concepts clés reconnus comme cruciaux lors des premiers pas en la matière,
- **L'enseignement aux adultes** : où nous commençons par définir celui-ci et à le positionner par rapport à l'apprentissage. Nous observons ensuite les éléments qui mènent à la reprise d'études. Nous terminons cette section en introduisant le concept d'andragogie.
- Le sujet clé de notre mémoire, à savoir **l'enseignement par le jeu** constitue le dernier volet de notre exploration littéraire. Il y sera question des Serious Game, de leur origine, de leurs caractéristiques, de leur classification ou encore de leur développement. La mise en œuvre de tels jeux dans un contexte éducatif sera finalement abordée.

Le **troisième chapitre** développe d'abord la méthodologie utilisée pour mener notre recherche. Il y est question du **design expérimental d'environnement d'apprentissage** et de la manière de concevoir et de conduire une expérimentation basée sur la méthodologie RDE. Nous y décrivons la manière dont des règles de design guident la réalisation d'artefacts expérimentaux et dont ceux-ci sont évalués *in situ* puis discutés afin de faire émerger où de préciser certaines théories.

S'ensuit une description plus précise du cadre et du déroulement de notre propre expérimentation. Les modalités pratiques précises pour collecter et traiter les résultats y sont précisées.

Le **quatrième chapitre** est consacré aux résultats : la **description** de ceux-ci d'une part et leur **interprétation** d'autre part.

Le **cinquième chapitre** conclut ce travail sur la réponse à la question de recherche, les réserves à apporter et les pistes de développement suscitées par notre contribution.

# Chapitre 2

## État de l'art

### 2.1 Initiation à la programmation informatique

Nous allons dans cette section aborder les concepts informatiques impliqués dans l'enseignement de la programmation aux débutants. Après un bref passage sur l'informatique en tant que science et ses principes, nous aborderons les différents courants sous-jacents aux cursus actuels. En analysant ces courants, nous tenterons de dégager des pistes quant à la direction à suivre pour notre recherche, en tenant compte des spécificités de notre public.

#### 2.1.1 Contours et structure du domaine

Les définitions de la discipline informatique se sont succédées au fil du temps et continuerons de le faire tant que la discipline continuera d'évoluer. Selon [Denning, 2007] on pourrait résumer cette évolution comme ceci :

- L'usage de dispositifs mécaniques pour aider à la réalisation de calcul remonte à plusieurs siècles, des premiers abaqués à la machine différentielle de Babbage. En 1939, John Atanasoff construit le premier ordinateur conçu à partir de l'électronique digitale. D'autres machines comme celle de Turing voient le jour durant les années 30, avec en commun une finalité mathématique, parfois appliquée au domaine militaire. Denning considère ainsi que la naissance de l'informatique a lieu à ce moment, avec la fusion de trois disciplines : les mathématiques, l'ingénierie et la science. C'est aussi à cette époque qu'est apparue l'Association for Computing Machinery (ACM).
- De 1950 à 1970, les définitions essentiellement mathématiques utilisées jusqu'à se sont faites contraignantes. Des champs d'application émergents comme l'intelligence artificielle n'apparaissent plus comme fondamentalement mathématiques. C'est aussi à cette époque que la question de savoir si l'informatique peut être considérée comme une science à part entière a vu le jour. En réponse, la définition s'est élargie pour englober l'ensemble des phénomènes au sein mais aussi autour des ordinateurs. C'est aussi durant ces années que l'on voit l'informatique prendre une place réellement significative dans les institutions académiques, avec les premiers programmes de cours, essentiellement basés sur les fondements mathématiques de l'informatique et élaborés par l'ACM.
- Durant les années 80, les informaticiens ainsi que plusieurs scientifiques de

différents domaines commencent à voir l'informatique comme une nouvelle manière de pratiquer la science. Dépassant son cadre d'origine, il s'agit d'un mode de pensée innovant ouvrant sur de nouvelles découvertes : la pensée informatique.

- À partir des années 90, les pratiquants des sciences naturelles comme la biologie ou la physique commencent à établir des correspondances entre les mécanismes observés dans leurs domaines respectifs et ceux présents dans les processus de traitement de l'information propres à l'informatique.

Selon Peter J. Denning, ancien directeur de l'ACM et acteur majeur dans l'enseignement de l'informatique, « La discipline de l'informatique est l'étude systématique des processus algorithmiques qui décrivent et transforment l'information : leur théorie, leur analyse, leur conception, leur efficacité, leur mise en œuvre et leur application. La question fondamentale qui sous-tend l'ensemble de l'informatique est la suivante : que peut-on (efficacement) automatiser ? » ([Denning, 2005]). Ainsi, la science, les mathématiques et l'ingénierie se combinent-elles pour former le vaste champ d'application de l'informatique. Certaines activités comme l'étude des algorithmes expérimentaux sont essentiellement de nature scientifique. D'autres comme par exemple les calculs de complexité des algorithmes sont apparentées aux mathématiques. La conception de microprocesseurs ou d'architectures logicielles relèvent du domaine de l'ingénierie.

L'informatique est-elle une science ? Pour être reconnu comme tel, un champ de recherche doit répondre à différents critères ([Denning, 2003] se basant sur Francis Bacon) :

1. Un corpus de connaissances systématiquement organisé,
2. Une méthode expérimentale,
3. Des résultats expérimentaux reproductibles,
4. Des hypothèses, vérifiables ou infirmables,
5. Des prévisions surprenantes,
6. Des objets d'étude naturels.

Après avoir débattu ces questions de manière détaillée, il conclut que malgré quelques dissensions internes et externes, on peut bel et bien assimiler l'informatique à une science, humaine qui plus est ([Denning, 2005, Denning, 2007]).

### 2.1.2 L'approche fondamentaliste de Peter Denning

Le cadre d'analyse séminal « Great Principles » ([Denning, 2003]) énumère un certain nombre de grands principes de la science et de l'ingénierie informatique. Un principe est une déclaration qui guide ou contraint l'action. Trois critères ont été utilisés pour sélectionner ces principes : l'universalité, la récurrence et l'influence majeure. Ces principes ne sont cependant pas invariants : avec le temps, certains peuvent être réinterprétés ou encore devenir obsolètes.

Sont d'abord rassemblés les grands principes du corpus théorique de l'informatique, qu'il rassemble sous les termes de mécanique de l'informatique (« computing mechanics ») par analogie avec d'autres sciences comme avec la physique. Sept thèmes englobent ces principes et s'entrelacent pour former une seule trame globale :

- **Computation** qui se traduit par traitement, calcul. Il s'agit de considérer l'information, sa représentation, la manière dont elle est traitée, avec quelle

efficacité (temps, complexité). C'est dans cette thématique qu'on retrouve par exemple les algorithmes, la logique des prédicats ou encore la calculabilité.

- **Communication** ou comment les informations sont transformées en messages transmis sur des médias à l'aide d'émetteurs, de récepteurs et de diverses techniques d'encodage et de décodage. Les réseaux TCP/IP appartiennent à cette thématique.
- **Coordination** Un système de coordination est caractérisé ensemble d'agents interagissant avec un objectif commun. La synchronisation des threads d'application ou les interblocages relèvent du domaine de la coordination.
- **Recollection** se traduit par *mémorisation*. Les principes du stockage, de la récupération des informations afin de procéder à leur traitement dépendent du concept de mémorisation. Par exemple : Les schémas d'adressage ou l'optimisation à l'aide de cache.
- **Automation** concerne l'automatisation physique qui met en correspondance les tâches de calcul complexes avec les systèmes physiques, qui les exécutent de manière acceptable. Ainsi, l'intelligence artificielle établit une correspondance entre les tâches cognitives humaines et les systèmes physiques qui les exécutent de manière acceptable. On parle ici notamment de deep learning ou encore de systèmes experts.
- **Evaluation** Ces principes concernent la manière dont les systèmes informatiques fonctionnent sous diverses charges de calcul et la capacité dont ils ont besoin pour fournir leurs résultats à temps.
- **Design** Ces principes concernent la manière de concevoir des logiciels et des systèmes informatiques qui soient fiables, sûrs, ergonomiques et sécurisés.

Ces catégories résultent d'une analyse fonctionnelle de plusieurs technologies de l'informatique et de leurs applications :

- Les systèmes informatiques sont bâtis à partir d'unités de traitement qui transforment et mémorisent l'information (computation, mémorisation) ;
- les unités de traitement échangent des informations (communication) ;
- les unités de traitement coopèrent avec un objectif commun (coordination) ;
- les êtres humains délèguent des tâches à des systèmes informatiques (automation) ;
- les êtres humains prédisent la vitesse et les capacités des systèmes (évaluation) ;
- les êtres humains décomposent les systèmes en unités de traitements et organisent leur architecture (design)

La mécanique informatique n'épuise pas tous les principes du domaine. Les professionnels de l'informatique suivent des principes de conception (« Design ») qui leur permettent de mettre la mécanique au service des utilisateurs et des clients. Cinq préoccupations animent les principes de conception :

- La simplicité : Diverses formes d'abstraction et de structure qui permettent de surmonter l'apparente complexité des applications.
- Performance : prédire le débit, le temps de réponse, les goulots d'étranglement, la planification des capacités.
- Fiabilité : redondance, récupération, point de contrôle, intégrité, confiance dans le système.
- Évolutivité : adaptation aux changements de fonction et d'échelle.
- Sécurité : contrôle d'accès, secret, vie privée, authentification, intégrité, sé-

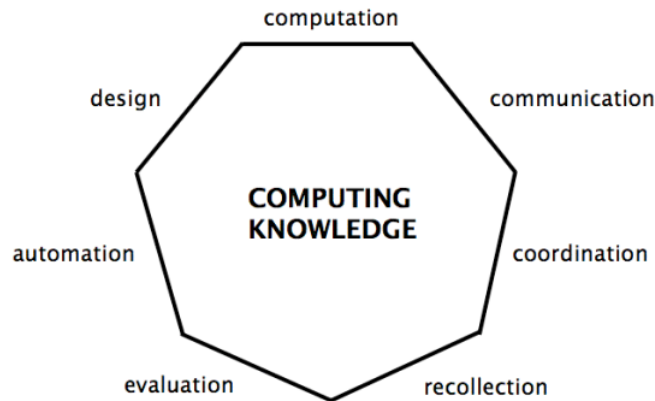


FIGURE 2.1 – Catégories du cadre d'analyse « Great Principles (GP) » ( [Denning, 2003])

curité.

Bien qu'elles ne soient pas des principes, il est utile d'identifier les pratiques centrales des professionnels de l'informatique, afin de les distinguer des principes. Les pratiques sont les « savoir-faire » qui constituent l'ensemble des compétences d'une personne. Elles sont distinctes du « savoir-quoi » représenté dans les descriptions des principes. Les individus peuvent être évalués à différents niveaux de compétence dans une pratique, comme débutant, compétent et expert. Les pratiques de base de l'informatique sont les suivantes :

- Programmation (y compris la pratique de plusieurs langages)
- Systèmes et pensée systémique
- Modélisation, validation, test et mesure
- Innover
- Appliquer

Dening propose donc une structuration de l'informatique selon plusieurs axes. Les grands principes sont de deux ordres : l'un, académique, correspond à la mécanique informatique avec sept thèmes correspondant à des perspectives différentes d'observation, et l'autre concernant le design. Les différentes technologies du domaine sont bâties sur ces grands principes. Parallèlement à ce découpage, les savoir-faire des artisans du domaine sont regroupés en plusieurs catégories correspondant à des qualités spécifiques.

Cette structuration a pour objectif d'aider à la connaissance du domaine de l'informatique.

### 2.1.3 Le niveau secondaire supérieur de Gilles Dowek

En 2012, la France a introduit l'informatique dans son programme de cours pour les classes de terminales scientifique, pour faire face à la numérisation croissante de la société autant que pour préparer les étudiants à des études spécialisées. Dans le but d'élaborer un support pédagogique efficace [Dowek et al., 2013], le parti est pris de structurer les cours en se reposant sur quatre notions fondamentales qu'il s'agit de représenter de la manière la plus équitable possible :

- L'algorithme : la possibilité de généraliser la notion à d'autres domaines que

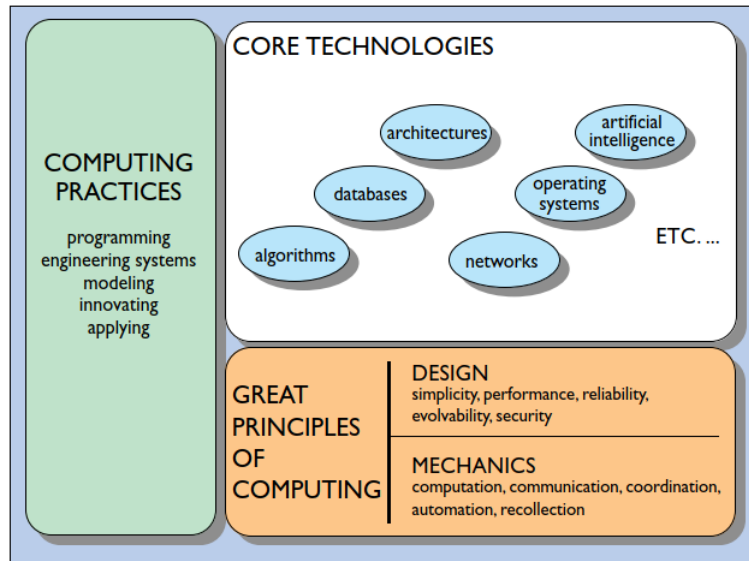


FIGURE 2.2 – Portrait de l’informatique basés sur les grands principes « Great Principles (GP) » ( [Denning, 2003])

l’informatique, l’importance d’utiliser un langage symbolique, le (non) déterminisme sont notamment abordés.

- L’information : la numérisation en tant que représentation symbolique est mise en évidence. D’autres notions comme le codage, la compression, le cryptage la sauvegarde et la récupération sont également vues.
- Les langages informatiques : en partant de la notion générale et de son découpage en langages informels et formels, de la postériorité des ordinateurs relativement à ces derniers, les langages en tant que moyen de décrire les algorithmes et de les implémenter sont vus en premier.
- La machine, comme support d’exécution des algorithmes : sont autant considérés l’architecture d’un ordinateur, avec ses portes logiques, son langage machine et ses limitations que les réseaux et les robots.

Deux points sont jugés comme important par les auteurs : la généralité des concepts ainsi que leur pérennité. Même si les technologies et applications évoluent rapidement, le cœur du domaine est vu comme relativement stable, tout au moins au niveau scolaire abordé. En pratique, la mise en application de ces notions au sein du cursus élaboré suppose que l’étudiant est déjà doté de capacités d’abstraction conséquentes ainsi de que solides notions en mathématiques.

#### 2.1.4 L’approche anglo-saxonne

Aux États-Unis, quatre organisations coordonnent leurs efforts depuis près d’un demi- siècle ans pour dicter des lignes de conduite relatives à l’enseignement de l’informatique au niveau supérieur : l’ACM, l’AIS (Association for Information Systems), l’AITP (Association of Information Technology Professionals) et l’IEEE-CS (Computer Society of the Institute for Electrical and Electronic Engineers). Le rapport [Shackelford et al., 2006] rédigé conjointement par ces organisations distingue cinq disciplines majeures :

- *Computer Engineering* (CE) où l'accent est mis sur l'ingénierie d'ordinateurs ou de dispositifs en contenant, ainsi que sur l'intégration de logiciels dédiés. La tendance actuelle dans ce domaine est aux systèmes embarqués, tels les objets connectés, les systèmes de guidages, etc.
- *Computer Science* (CS), discipline qui couvre un large spectre allant des fondements théoriques et algorithmiques aux tous derniers développement tels la bio-informatique et l'intelligence artificielle. Trois grandes optiques se dégagent : le développement de logiciels tenant compte des dernières avancées, l'utilisation innovante des technologies, avec par exemple les ordinateurs quantiques, et l'étude des nouvelles manières de résoudre des problèmes à l'aide d'algorithmes de plus en plus évolués, de protocoles réseaux améliorés, etc. Un travail ancré dans de solides fondations théoriques permet une optimisation des performances, mais il est parfois reproché à cette discipline d'être moins axée sur la productivité professionnelle.
- *Information System* (IS) : avec le traitement des informations comme moyen de répondre au besoin spécifiques des entreprises en terme de productivité. La conception des systèmes d'informations demande une compréhension aigüe des mécanismes et des structures d'entreprise. Un exemple est donné par le système Alladin, conçu et exploité par la plus grande société de gestion d'actifs mondiale Blackrock. Ce système fait usage des dernières avancées en intelligence artificielle pour optimiser de manière optimale et automatique les opérations financières réalisées par l'entreprise.
- *Information Technology* (IT). Cette discipline possède deux acceptations. Dans le langage courant, elle fait référence aux usages génériques du numériques. Au niveau académique, elle vient compléter l'IS. Alors que les systèmes d'informations sont concentrés sur l'information et son traitement, cette branche aborde principalement la technologie en tant qu'outil au service des autres activités, plus spécifiques. Par exemple, un service IT est responsable d'une intégration efficace des outils, autant matériels que matériels au sein d'une organisation.
- *Software Engineering* (SE), où l'on se concentre sur la conception, l'implémentation et la maintenance à long terme d'applications informatiques fiables et financièrement viables, répondant au mieux aux exigences des clients. Le caractère intangible d'un software est confronté à son évolution discontinue, c'est ce qui donne un caractère particulier à cette discipline.

Le même rapport énumère quarante sujets présents, dans des mesures variables, dans chacune des disciplines évoquées plus haut. Au travers d'un tableau, deux valeurs numériques *min* et *max*, sont attribuées à chaque sujet, pour chaque discipline. La valeur minimum traduit le niveau d'exigence le plus bas attendus dans une discipline donnée par rapport aux autres. Les étudiants peuvent se spécialiser dans des domaines particuliers, la valeur maximum exprime le niveau généralement attendu dans ce cas, en tenant compte des limites définies par rapport aux autres disciplines. Les pondérations sont exprimées sous la forme de valeurs entières comprise entre 0 (faible) et 5 (fort). Un exemple est donné par le tableau 2.1.



Sujet	CS		IS	
	Min	Max	Max	Min
Théorie des langages de programmation	3	5	0	1
E-Business	0	0	4	5

TABLE 2.1 – Exemple de pondération de sujet par rapport aux disciplines, extrait de [Shackelford et al., 2006]

En se basant sur les exigences minimales, il est alors possible de déterminer les sujets les plus fondamentaux, à l'aide de la formule suivante où  $I_{suj}$  représente l'importance d'un sujet,  $Min_{i,suj}$  l'exigence minimale pour le sujet  $suj$  au sein de la discipline  $i$  et  $nbDisciplines$  le nombre de disciplines :

$$I_{suj} = \frac{\sum_{i=1}^{nbDisciplines} Min_{i,suj}}{nbDisciplines}$$

Les quatre sujets les plus importants sont donc : « Fondamentaux de la programmation » ( $I_{suj} = 3.4$ ), suivi de « Interaction homme machine » ( $I_{suj} = 2.6$ ), « analyse des besoins techniques » ( $I_{suj} = 2.4$ ) et « conception de logiciels » ( $I_{suj} = 2.4$ ).

Un autre tableau extrait du même rapport décrit les aptitudes sensées être acquises au terme des diverses activités d'enseignement. Elles sont au nombre de soixante, ventilées sur onze catégories. Une pondération semblable à celle décrite précédemment est utilisée, n'utilisant cette fois qu'une seule valeur, toujours comprise entre 0 et 5. En utilisant une formule semblable, il est possible de déterminer les aptitudes considérées comme fondamentales, à savoir « programmer à petite échelle » ( $I_{apt} = 4.2$ ), « créer une interface utilisateur » ( $I_{apt} = 4.0$ ) et « utiliser correctement les fonctionnalités d'un traitement de texte » ( $I_{apt} = 4.2$ ).

$$I_{apt} = \frac{\sum_{i=1}^{nbDisciplines} I_{i,apt}}{nbDisciplines}$$

L'analyse réalisée ci-dessus nous a permis de mettre en évidence deux éléments omniprésents et fondamentaux dans l'ensemble des cursus informatiques : **la maîtrise des concepts fondamentaux de la programmation** et **l'aptitude à concevoir un programme simple**. Cette information est importante : elle repose sur un large consensus au sein des communautés académiques, professionnelles et éducatives, elle reflète le côté transversal des éléments dégagés et s'applique au niveau des études supérieures. Ce dernier point est intéressant dans la mesure où ce mémoire s'attache au secondaire supérieur,

### 2.1.5 En Belgique

Au niveau abordé dans le cadre de ce mémoire (enseignement secondaire supérieur de promotion sociale), il existe principalement trois sections dans le domaine informatique : technicien en bureautique, technicien en informatique et technicien en programmation. En consultant les dossiers pédagogiques d'application pour les deux premières sections, on peut lire :

*Cette section a pour finalités de permettre à l'étudiant :*

- de prendre en charge les problèmes relatifs à l'utilisation et l'entretien de PC;

- d'être l'élément d'appoint d'une équipe de maintenance de systèmes informatiques ;
- d'assurer l'exécution et le suivi de procédures de maintenance logicielle [et matérielle dans « technicien en informatique »], de contrôle et de sécurité des systèmes informatiques et des réseaux locaux.

Au niveau détaillé des contenus, outre un tronc commun, l'accent est mis sur les logiciels (traitement de texte, tableur, présentation, etc.) d'un côté et sur le matériel de l'autre. En ce sens, ces deux sections se rapprochent de la discipline IT du modèle anglo-saxon. La dernière, consacrée à la programmation se rapproche, elle, de la discipline CS, notamment en raison du large spectre des disciplines abordées (programmation, architecture des bases de données, réseaux, etc.)

C'est dans cette section que notre recherche prend place, plus précisément au sein du module de cours « Programmation - Niveau 1 ». Voici ce qui est mentionné dans le dossier pédagogique officiel de ce module.

**Programme officiel :** « Face à un système informatique opérationnel et connecté à internet, sur base des consignes données par le chargé de cours, en respectant les règles d'utilisation et de sécurité de l'équipement et du matériel informatique, en utilisant les commandes appropriées, en utilisant le vocabulaire technique adéquat, l'étudiant sera capable :

- de définir la notion de variable ;
- d'identifier et d'associer les différents types de variables et leurs contenus ;
- d'utiliser les variables dans des opérations (affectation, déclaration, opérations mathématiques, concaténation ...)
- de représenter sous forme schématique les différentes structures nécessaires à la résolution d'un problème (structures conditionnelles, itératives ...)
- de définir les notions de procédures et de fonctions ;
- d'identifier et d'utiliser les procédures et les fonctions ;
- de résoudre des situations élémentaires proposées par le chargé de cours
  - en réalisant sous forme schématique la structure du programme ;
  - en la transposant dans un langage structuré ;
  - en la traduisant en langage de programmation en respectant sa syntaxe spécifique ;
- de recourir à bon escient à la documentation disponible. »

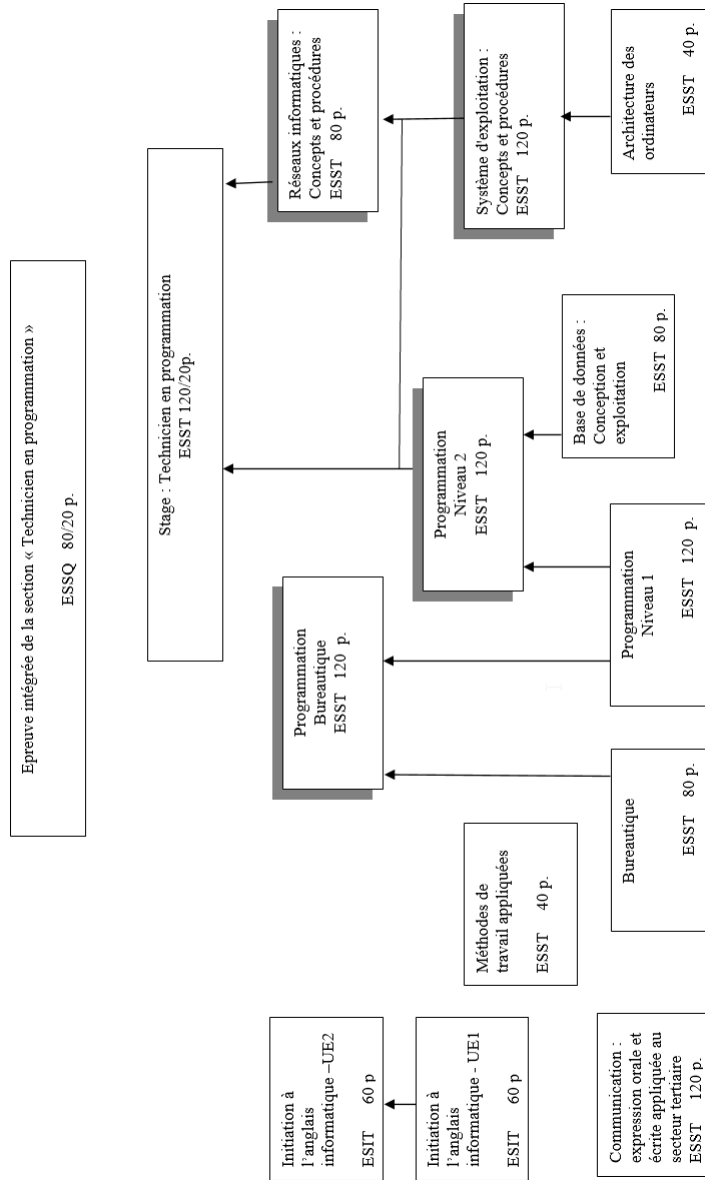


FIGURE 2.3 – Organigramme de la section « Technicien en programmation »

### 2.1.6 La pensée informatique de Jeanette Wing

La pensée informatique ou computationnelle est introduite par Jeanette Wing en 2006, qui la décrit comme :

*« un ensemble d'attitudes et d'acquis universellement applicables que tous, et pas seulement les informaticiens, devraient apprendre et maîtriser. [...] [elle] conduit à résoudre des problèmes, à concevoir des systèmes et à comprendre le comportement humain, en s'appuyant sur les concepts fondamentaux de la discipline et en y incluant une large collection d'outils intellectuels qui reflètent l'étendue de la science qu'est l'informatique. [...] Penser en informaticien signifie beaucoup plus que savoir programmer, cela requiert de penser à plusieurs niveaux d'abstraction » [Wing, 2006].*

Le concept a été par la suite été repris par le monde académique, plusieurs chercheurs établissant des définitions plus ou moins concordantes [Denning, 2006, Yadav et al., 2014, Hemmendinger, 2010, Furber, 2012]. Après plusieurs révisions de son propre travail (la publication originale ne comportait que trois pages et n'explorait qu'une partie des sujets impliqués), Wing elle-même propose une autre définition selon laquelle « La pensée informatique est le processus de pensée impliqué dans la formulation des problèmes et de leurs solutions afin que les solutions soient représentées sous une forme pouvant être efficacement mise en œuvre par un agent de traitement de l'information. » [Wing, 2011].

Elle défend l'idée que la pensée informatique devrait faire partie des compétences intellectuelle de tout un chacun, et pour ce faire être introduite systématiquement dans les cours, le plus tôt possible. Selon elle, la pensée informatique appliquée à la programmation est caractérisée par :

- L'abstraction, qui en constitue l'essence. Différents outils sont impliqués comme la reformulation, la modélisation ou la décomposition de problèmes, le dégagement d'invariants, la modularisation, etc.
- Des méthodes de construction générales, où sont inclus les boucles, les structures conditionnelles, la récursivité, les traitements concurrents. La vue d'ensemble d'un programme permet aussi d'en percevoir l'esthétique.
- La prise en compte des contraintes physiques, les programmes étant in fine destinés à interagir avec le monde concret. Il convient dès lors de prendre en compte les situations extrêmes (worst case scenario) en mettant en œuvre la redondance, la détection ou la correction d'erreurs, les contrats d'interface, etc.
- Un autre niveau de réflexion concerne les données. L'omniprésence des technologies permettant leur collecte ouvre la voie à de nouvelles idées. Il existe une boucle de rétroaction : ces nouvelles idées amèneront à poser de nouvelles questions qui nécessiteront la collecte de plus de données, faisant appel à de nouvelles technologies. Les abstractions, représentations et traitements des données selon les points de vue de la pensée informatique permettront d'extraire les connaissances enfouies dans les données ou diffusées à travers elles.

### 2.1.7 En résumé

En ne citant qu'un nombre restreint d'auteurs et d'organismes, il apparaît une importante variété d'approches pour définir le cursus idéal. Le modèle anglo-saxon possède sans doute les racines les plus anciennes et décline le domaine informatique

en une multitude de sujets répartis sur cinq disciplines principales. L'analyse de ces sujets dégage la maîtrise des concepts fondamentaux de la programmation et l'aptitude à concevoir un programme simple comme les éléments les plus importants parmi plus d'une centaine. En réponse à la complexité de ce découpage, Dening élabore le modèle des « Great Principles ». Ces concepts sont fondamentaux, faciles à énoncer, organisés selon sept grands axes et possèdent chacun une histoire les décrivant précisément. Parallèlement à ce corpus théorique, Dening décrit un ensemble de compétences à acquérir dont le degré de maîtrise révèle la qualité du praticien : la programmation, la pensée systémique, la modélisation, l'innovation et la mise en application. En France, l'éducation nationale, dans sa volonté de réintroduire l'informatique dans l'enseignement secondaire utilise un découpage proposé par Gilles Dowek selon quatre domaines : algorithmes, informations, langages et machine. En raison de la filière scientifique où est introduite l'informatique, le cursus est fortement teinté de concepts et d'applications mathématiques. En Belgique, au niveau considéré dans ce travail, le premier module de cours dédié à la programmation attend de l'étudiant, pour atteindre le seuil de réussite, qu'il soit capable de résoudre une situation élémentaire en réalisant sous forme schématique la structure d'un programme. Celle-ci doit ensuite être transposée dans un langage schématique puis implémentée dans un langage de programmation donné, en respectant grammaire et syntaxe.

Pour résumer, quel que soit le niveau de complexité ou d'innovation où l'on se place dans le vaste domaine des sciences de l'informatique, on se rend compte qu'existe une fondation construite sur des concepts pérennes, relativement simples et peu nombreux. Pour s'engager dans un apprentissage fructueux, le plus important semble d'assimiler profondément ces concepts, de les appréhender sous toutes leurs formes et dans toutes leur subtilité. Il est ainsi crucial d'apprendre à combiner les briques élémentaires de l'algorithmique, d'expérimenter, d'innover, d'identifier les schémas, de généraliser les applications, d'identifier et de renforcer les liens profonds entre l'abstrait et le concret. C'est en persévérant dans cette démarche que l'apprenant pourra finalement réaliser tout le potentiel de la « pensée informatique ».

## 2.2 Enseigner aux adultes - L'andragogie

Il est, dans cette seconde partie de notre état de l'art, question d'aborder le concept d'andragogie. Pour introduire celui-ci, nous nous proposons de parcourir un certain nombre de considérations relatives à l'adulte en reprise de formation. Avant toute chose, il nous a semblé pertinent de nous pencher plus avant sur la définition de l'adulte. Dans son acceptation la plus simple, le vocable signifie « Qui est parvenu au terme de sa croissance, à son plein développement. Qui fait preuve d'équilibre, de maturité (par opposition à infantile) » (Dictionnaire Larousse en ligne).

Il est toutefois à noter que le statut d'adulte peut s'apprécier selon différentes perspectives. Ainsi [Petit, 2011] énumère que «

- d'un point de vue biologique, l'adulte est celui qui est en âge de se reproduire,
- d'un point de vue économique, c'est celui qui travaille et qui est en mesure de subvenir à ses besoins,
- d'un point de vue juridique, c'est celui qui devient responsable de ses actes,
- d'un point de vue psychologique, c'est celui qui a conscience de sa responsabilité. »

L'adulte est « une personne qui décide, agit et gère des situations complexes, jouissant d'une liberté de choisir et d'agir. C'est aussi un être en quête d'identité, marquant et affirmant sa singularité au travers de ses choix et ses actions. Une quête d'identité qui fait de lui un *sujet* » [Hubrecht, 2016]

« Un sujet en développement et en quête de reconnaissance, qui souhaite valoriser son action sur son environnement et être reconnu dans ses désirs, ses aspirations et ses compétences. Un sujet qui veut que soient pris en compte son expérience, ses affects, sa motivation et son image de lui » [Averlant, 2007]. L'andragogie s'adresse ainsi aux personnes qui ont atteint la maturité. Ces dernières ne sont pas à confondre avec celles qui ont atteint l'âge de la majorité.

Il existe de multiples définitions de l'adulte. Du point de vue biologique, on devient adulte lorsque l'on est capable de procréer. Légalement, lorsque l'on arrive à l'âge où l'on est autorisé à voter, à posséder un permis de conduire, à se marier, etc. Socialement, c'est plus une question de rôles que l'on endosse dans la société comme celui de parent, travailleur, époux ou citoyen. Dans le cadre de l'apprentissage, une autre facette, psychologique, paraît fondamentale. Il s'agit du moment où nous nous voyons nous-même comme adulte, ou nous nous considérons responsable de notre vie, de la direction que nous voulons lui donner et de nos choix. Il ne s'agit évidemment pas d'un cap que l'on peut définir avec précision, dans la mesure où cette évolution est progressive, depuis l'enfance et au travers de l'adolescence, influencée par les éléments cités avant. Généralement toutefois, quitter l'école, trouver du travail, fonder une famille concourt à forger cette image d'adulte que l'on a de soi.

### 2.2.1 Enseigner aux adultes

Pour savoir comment enseigner, il peut être utile de changer de point de vue et de se mettre à la place de l'apprenant.

[Perrenoud, Ph., 2004] pose la question « Qu'est-ce qu'apprendre ? » et énumère en guise de réponse un certain nombre de réponses :

- *Désirer* : Selon Philippe Meirieux, si le désir de savoir est naturel, celui d'apprendre ne l'est pas forcément. « *faire apprendre, c'est aussi, c'est d'abord créer et maintenir le désir d'apprendre et de savoir, et neutraliser des désirs antagonistes.* »
- *Persévérer* : Il ne suffit pas s'engager sur la voie de l'apprentissage pour arriver à son terme. Cette progression nécessite des efforts, parfois conséquents, qui risquent d'être source de découragement, ou même d'abandon. Persévérer face à des obstacles que l'on n'arrive pas à surmonter, à des idées que l'on n'arrive pas à saisir ou à retenir demande volonté, discipline et une capacité à résister à la frustration et à une forme de souffrance. Être capable de se projeter dans l'avenir pour imaginer les avantages du succès aide, prendre du plaisir dans des activités d'apprentissage stimulantes et excitantes également.
- *Construire* : Apprendre, ce n'est pas accumuler une suite de savoirs, qui seraient transmis comme des objets que l'on empilerait. Il s'agit plutôt de mettre en relation, d'organiser des notions, des concepts pour en construire d'autres. Dans ce contexte, le rôle de l'enseignant est de mettre en œuvre son expertise pour communiquer des briques de base, mais surtout créer des situations où l'apprenant, pour surmonter des obstacles, devra construire lui-

même de nouveaux savoirs.

- *Interagir* : Parfois, on travaille seul pour apprendre un cours, pour rédiger un document. Dans d'autres cas, on se confronte à des outils comme des logiciels informatiques. Mais apprendre, c'est au moins autant interagir avec autrui, dans une relation de coopération ou de compétition.
- *Prendre des risques* : Selon Jean-Pierre Astolfi, l'erreur est un outil pour enseigner. Essayer, échouer, analyser ses erreurs et remettre cent fois sur le métier son ouvrage est probablement l'une des manières les plus efficaces d'ancrer à long terme ses acquis. Cela implique de se mettre en porte à faux, au moins temporairement, et de faire face à une situation d'échec avant de repousser ses limites.
- *Mobiliser et faire évoluer un rapport au savoir* : Les didacticiens l'affirment : nul n'aborde un savoir sans une idée préconçue, un sentiment à l'égard de celui-ci. Combien d'étudiants ont ainsi une aversion marquée pour la trigonométrie par exemple ? Dans ce cas, un rapport défensif peut constituer un obstacle important à l'apprentissage qu'il convient de prendre en compte.

### 2.2.2 La reprise d'études

La reprise d'étude demande une importante motivation de la part des apprenants. Plusieurs auteurs tentent de comprendre les motivations qui peuvent amener les adultes à s'inscrire dans un projet d'études. Par exemple, [Demol, 1995] distingue trois types de motifs permettant d'expliquer la décision d'entamer une formation :

- « les besoins personnels, [...] les aspirations d'ordre personnel ou d'ordre psychologique au sens large du terme : désir d'apprendre, de connaître « autre chose », quête d'autonomie et/ou de réalisation de soi ; » .
- « les possibilités offertes [...] : réalisation d'un projet professionnel, changement de statut professionnel et/ou social, et, en moindre proportion, changement de milieu professionnel ; ».
- « les influences extérieures, [...] les influences subies par le stagiaire au moment d'entrer en formation. La décision de formation est alors le plus souvent liée à un événement d'ordre familial ou professionnel ».

### 2.2.3 Le modèle « expectancy value »

Etienne Bourgeois associe l'entrée en formation des adultes au paradigme « expectancy value ». Selon ce dernier, la décision de s'engager en formation constitue le résultat « d'une interaction positive entre deux facteurs fondamentaux, à savoir » [Bourgeois, 1998] :

- « La valeur (value) attribuée par le sujet à la formation envisagée », c'est-à-dire la croyance selon laquelle la formation envisagée « lui apportera effectivement des bénéfices (et que ceux-ci sont suffisamment importants, pour lui) »,
- « L'espérance de réussite dans cette tâche (expectancy) », c'est-à-dire la croyance selon laquelle « ses chances de succès dans l'entreprise sont suffisamment élevées ».

Dans ce modèle, la valeur attribuée à la formation et l'espérance de réussite sont considérées comme interactives. Ainsi en cas d'absence de l'un d'entre eux la décision

de s'engager est compromise.

### 2.2.4 Les caractéristiques des motifs d'engagement

Selon Joseph et Mailhot les motifs d'engagement en formation des adultes sont pluriels, contingents et évolutifs [Joseph M., 2018].

**Des motifs pluriels** Différents types de motifs peuvent amener les adultes à s'engager dans un projet de formation. Les auteurs en distinguent dix, à savoir épistémiques, socio-affectifs, hédoniques, économiques, prescrits, dérivatifs, opératoires professionnels, opératoires personnels, identitaires et vocationnels. Toujours selon les auteurs, « on s'engage rarement en formation pour une seule raison, plusieurs motifs entrant en général dans une combinaison singulière ». D'ailleurs, selon [Broekemier, 2002], cité par [Doray et al., 2012], les motifs d'engagements « peuvent se combiner », sans pour autant « s'exclure mutuellement ».

**Des motifs contingents** Toujours selon Joseph et Mailhot, les motifs d'engagement en formation des adultes sont contingents dans la mesure où « ils sont relatifs à un certain projet de formation ». Un même apprenant peut dès lors « avoir deux tableaux motivationnels différents selon qu'il s'engage dans telle ou telle autre formation ».

**Des motifs évolutifs** Les même auteurs soulignent finalement que les motifs d'engagement en formation des adultes peuvent changer dans le temps en fonction des « conditions de vie ou de l'histoire de vie des individus ». En d'autres mots, un adulte en reprise d'études peut ressentir différents types de motivation pour une même formation « à des moments différents de sa vie ».

### 2.2.5 Moteurs d'apprentissage chez les adultes peu qualifiés

[Brown and Bimrose, 2018] analysent le parcours professionnel et d'éducation de 105 adultes répartis en Europe, âgés de 25 à 40 ans. Les résultats dégagent cinq moteurs jugés essentiels dans le cadre de l'apprentissage :

- Amélioration personnelle
- Amélioration de ses capacités
- Orientation professionnelle
- Pratique professionnelle
- Relation aux autres

### 2.2.6 Étymologie et principes de l'andragogie

Étymologiquement, le terme andragogie est issu de deux racines grecques : andros (άνδρός), qui signifie « l'homme », dans le sens d'homme mâle et agogos (άγωγός), qui signifie « le guide ».

Le concept d'andragogie apparaît pour la première fois au XIXème siècle dans un ouvrage d'Alexander Kapp [Loeng, 2017] conçu à partir des travaux de Platon sur l'éducation [Kapp et al., 1833]. Ce n'est que bien plus tard, en 1921, qu'Eugène Rosenstock, Historien du droit et sociologue, fait à nouveau référence au concept



d'andragogie. Au lendemain de la Première Guerre mondiale, il voit ainsi la nécessité d'établir un enseignement propre à l'adulte, avec des structures, des enseignants et des techniques spécialement dédiés. Malcolm Shepherd Knowles (1913 - 1997) pionnier de l'éducation pour adulte, développe un modèle andragogique qui considère les caractéristiques spécifiques de l'adulte dans le processus d'apprentissage [Knowles et al., 2012]. Selon Knowles l'andragogie repose sur les six hypothèses principales suivantes :

1. « Les adultes ont besoin de savoir pourquoi ils doivent apprendre quelque chose.
2. Les adultes ont besoin d'être traités comme des individus capables de s'auto-gérer et admettent mal que les autres leur imposent leur volonté.
3. Les adultes arrivent avec une expérience propre, rendant les groupes très hétérogènes et nécessitant une personnalisation des stratégies d'enseignement et d'apprentissage. En revanche, cela favorise les méthodes expérimentales et activités d'échange.
4. Les adultes sont prêts à apprendre si les connaissances permettent d'affronter des situations réelles. L'apprentissage ne peut pas être dissocié du besoin de développement.
5. Les adultes assimilent d'autant mieux que les connaissances, les compétences, les attitudes sont présentées dans le contexte de leur mise en application sur des situations réelles.
6. Les adultes sont motivés intérieurement par le désir d'accroître leur satisfaction professionnelle, leur estime de soi... » [Knowles et al., 2012].

Dans son travail, Knowles se base lui-même sur [Lindeman, 1926] qui établit un certain nombre d'éléments clés caractérisant l'apprenant adulte. Ainsi, les adultes apprendraient le mieux lorsque :

- « L'apprentissage correspond à ce qu'ils veulent : les adultes connaissent leurs besoins et sont toujours heureux de partager la responsabilité de leur propre apprentissage.
- L'apprentissage est participatif : la participation active au processus d'apprentissage rend la formation plus efficace et a un impact plus marqué parce qu'elle est plus facile à appliquer.
- L'apprentissage repose sur des expériences vécues : Les adultes apportent un bagage d'expérience à la formation qu'ils désirent partager.
- L'apprentissage s'accompagne de réflexion : les adultes aiment réfléchir sur les expériences passées, tirer des conclusions et tirer des principes qu'ils pourront appliquer à l'avenir.
- L'apprentissage s'appuie sur les réactions : pour être efficace, l'apprentissage doit tenir compte des réactions qui sont favorables et qui proposent des modifications.
- L'apprentissage repose sur le respect du participant : le respect et la confiance mutuel entre le formateur et le participant mettent en valeur le processus d'apprentissage.
- L'apprentissage se déroule dans une atmosphère de sécurité : L'adulte peut arriver en formation avec son orgueil, et ses anxiétés face à un retour en classe. Il est primordial que le formateur établisse une atmosphère propice à des échanges respectueux » [Boutinet, 2013].

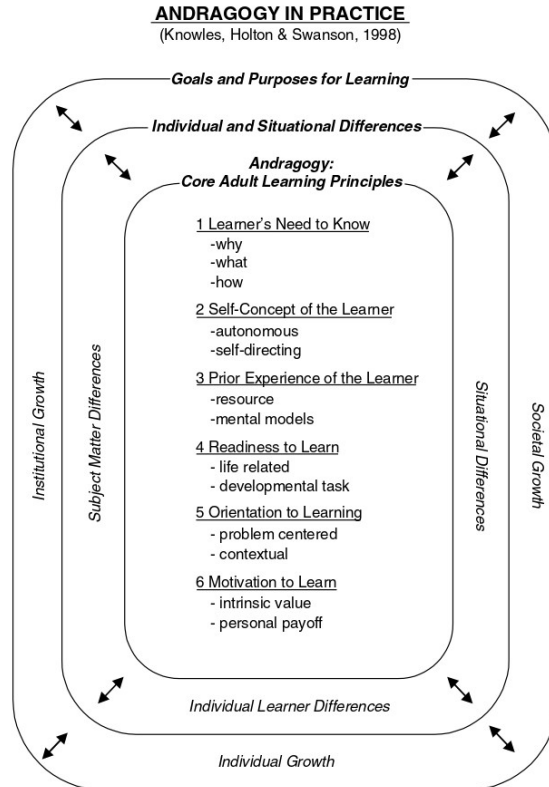


FIGURE 2.4 – Le modèle andragogique [Knowles et al., 2015]

**Une science imprégnée d’humanisme.** L’andragogie s’inscrit dans une démarche scientifique en ce qui concerne « la planification, l’application et l’évaluation des interventions éducatives ». Elle constitue toutefois également un art. Ceci parce qu’elle est nécessairement imprégnée d’humanisme et se base sur le climat relationnel qui existe entre :

- l’apprenant
- le facilitateur
- les autres apprenants

Il s’agit, en andragogie de favoriser un climat d’apprentissage détendu, convivial, informel, égalitaire, centré sur :

- « l’estime de soi de l’apprenant.
- le désir de collaboration,
- les besoins des apprenants » [Boutinet, 2013]

**Une approche spécifique** L’approche de l’andragogie se distingue à différents niveaux de la pédagogie pour enfants. Ceci notamment parce qu’elle s’adapte nécessairement aux caractéristiques du public auquel elle s’adresse, c’est-à-dire les adultes. Or ces derniers n’ont pas la même capacité de mémorisation que les enfants. Ils n’ont d’ailleurs plus l’habitude d’apprendre par cœur. Ils acceptent d’ailleurs moins facilement les idées toutes faites et ont besoin d’être convaincu.

De plus, « l’apprentissage est une remise en cause de ses certitudes, ce qui est parfois mal perçu » [Boutinet, 2013]. Par contre, « l’adulte dispose d’une expérience, sur laquelle on peut s’appuyer, et d’un esprit critique plus développé ».

## 2.3 L'enseignement par le jeu

Il est, dans cette troisième section de notre état de l'art, question d'aborder le sujet central de notre mémoire : les jeux sérieux ou « Serious Games » (que nous désignerons aussi par SG).

Pour comprendre au mieux le concept dont il est question ainsi que l'utilité qu'il peut présenter dans l'enseignement, il peut être utile de se pencher brièvement sur la notion de jeu elle-même. Le Larousse donne plusieurs plusieurs définitions du terme « jeu ». Parmi celles-ci, les deux principales nous semblent être :

- *une activité d'ordre physique ou mental, non imposée, ne visant à aucune fin utilitaire, et à laquelle on s'adonne pour se divertir, en tirer un plaisir*
- *une activité de loisir soumise à des règles conventionnelles, comportant gagnant(s) et perdant(s) et où interviennent, de façon variable, les qualités physiques ou intellectuelles, l'adresse, l'habileté et le hasard*

La première de ces définitions souligne le côté spontané, sans contraintes de l'activité ludique ainsi que le plaisir qu'elle procure. La seconde fait apparaître les notions de règles, de compétition, d'enjeu et de qualités requises.

### 2.3.1 Étymologie

L'idée même de jeu sérieux est très ancienne, Aristote par exemple annonçait son intention de traiter, dans sa Poétique, des vertus du rire et du jeu pour libérer l'esprit de l'erreur. Au XVI<sup>e</sup> siècle, dans le courant humaniste est apparu le concept de *serio ludere*, juxtaposition formelle des termes « jeu » et « sérieux », qui connut à l'époque un certain succès. Il s'agissait d'un mode reconnu d'expression de la pensée reposant sur l'oxymore, consistant à traiter un sujet sérieux en utilisant le jeu, le rire

Selon [Léotard-Sommer, 2019] il « confrontait les normes du serio à la licence du ludere, mais une licence qui devait être autorisée, c'est-à-dire *soumise à des règles* ». Parmi ses fonctions, le *serio ludere* était notamment utilisé dans un contexte pédagogique et préconisé par Erasme et Juan Luis Vivès dans leurs traités sur le sujet. Le moine Franciscain Jean Thenaud en aurait également fait usage en tant que précepteur de François 1<sup>er</sup>. Bien plus tard, au XIX<sup>e</sup>, en Autriche, apparaît un « jeu de guerre » ayant pour objectif d'entraîner les officiers.

Plus récemment, La première approche scientifique formelle des serious games est généralement attribuée à [Abt, 1970] dans son livre éponyme, où il décrit des simulations et des jeux, informatiques ou non (jeux de cartes, de sociétés), pour améliorer l'apprentissage. Les jeux vidéo, bien que mentionnés, n'en sont encore, comme l'informatique, qu'à leur balbutiements.

Actuellement, les jeux sérieux sont le plus souvent associés à l'informatique. La plupart prennent la forme de logiciels. Ils offrent généralement une certaine interactivité. À l'instar de toute application informatique, « le jeu soulage l'utilisateur des manipulations répétitives et fastidieuses, de la mémorisation des règles » [Loisier, 2015, p.17].

### 2.3.2 Définition

Il existe différentes définitions des SG. Dans une réflexion sur la sémantique des termes, [Djaouti, 2014] n'en dénombre pas moins de vingt-trois originales, antérieures

à 2010.

Ainsi, par exemple, selon le Grand dictionnaire terminologique, le jeu sérieux peut être défini comme une « Application informatique qui combine une intention sérieuse de type pédagogique, informatif, communicationnel, idéologique ou autre, avec un environnement d'apprentissage prenant la forme d'un jeu vidéo, afin de transmettre un savoir pratique ou de sensibiliser à un enjeu social ». (Grand dictionnaire terminologique de la langue française) Selon [Sauvé, 2008], professeure à la TELUQ, le jeu sérieux désigne « un jeu vidéo (avec un environnement réaliste ou artificiel) auquel les auteurs rattachent une composante pédagogique ».

Selon [Abt, 1970], *on peut jouer à des jeux sérieusement ou avec désinvolture. Nous entendons « jeux sérieux » dans le sens où ceux-ci ont un but éducationnel avéré et soigneusement étudié. Leur pratique n'a pas pour finalité première le divertissement. Ceci ne veut pas dire que des jeux sérieux ne sont pas, ou ne devraient pas être amusant.*

[Djaouti, 2011] mentionne *tout logiciel associant un objectif non divertissant (sérieux) et une structure de jeu vidéo (jeu)*

### 2.3.3 L'industrie du SG

L'industrie du SG est de caractère polymorphe. Elle regroupe effectivement l'ensemble des niches et des marchés qui « instrumentalisent le jeu vidéo pour viser des objectifs s'écartant du seul divertissement » [Michaud and Alvarez, 2008, p.31]. Elle se compose de trois types d'acteurs principaux, c'est-à-dire des développeurs, des éditeurs et des diffuseurs.

**Les développeurs** Les développeurs créent le contenu des SG. En Europe, ce groupe d'acteurs est essentiellement composé de PME, de TPE ou de freelances. Il s'agit, de manière générale, des « des pure players du SG ». Ils ne sont pas issus du monde du jeu vidéo mais plutôt « de la communication, du développement et de l'édition de logiciels métiers » [Michaud and Alvarez, 2008, p.32].

**Les éditeurs** Les éditeurs sont les acteurs qui se consacrent aux « frais de publication, de marketing et de conditionnement des SG » [Michaud and Alvarez, 2008, p.33]. Et ceci tant en ce qui concerne la vente dématérialisée que physique.

**Les diffuseurs** Dans le domaine des SG, les diffuseurs ne sont pas toujours liés à l'activité de distributeur. Ainsi « l'armée américaine ou bien la NSA diffusent au grand public leurs SG » [Michaud and Alvarez, 2008, p.33]. La diffusion s'effectue essentiellement par Internet.

**Les premiers SG** Les toutes premières utilisations « sérieuses » des jeux vidéo ont été mises à profit dans le secteur militaire. Leurs objectifs se situaient essentiellement dans l'entraînement et dans la simulation de stratégies.

« America's Army » est l'un des premiers SG marketing. Ce SG, conçu par l'armée américaine et sorti en 2000, permet aux joueurs de découvrir « la réalité des opérations militaires » [Méli, 2017]. Grâce à ce jeu, l'armée a souhaité susciter l'intérêt des jeunes tout en valorisant son l'image.

Les jeux sérieux ont ensuite également été investis par de grandes entreprises pour former et recruter leurs employés. Aujourd'hui l'utilisation des jeux sérieux concerne nombre de domaines et de secteurs, dont la publicité, les sports, la santé et l'éducation. Ils sont d'ailleurs également utilisés pour communiquer de l'information dans de nombreux secteurs, dont « le marché de l'emploi, l'écologie, la politique, l'humanitaire, la religion, l'art, etc. » [Loisier, 2015, p.17]. Le marché des jeux sérieux constitue en tout cas un marché en croissance constante.

### 2.3.4 Classification des jeux sérieux

En raison de l'engouement et du nombre croissant de jeux sérieux, il est important de définir une taxonomie qui aide à contextualiser le travail effectué dans ce domaine, notamment lors de la conception d'un jeu sérieux. En se fondant sur les informations extraites de diverses études [Laamarti et al., 2014] propose une classification selon diverses caractéristiques :

- *le type d'activité* demandée au joueur pour interagir avec le jeu. Cela va de l'activité physique pour lutter contre l'obésité par exemple à des stimulations physiologiques pour détecter certaines pathologies. Dans le cas de l'enseignement de la programmation qui nous occupe, il s'agit naturellement d'activités intellectuelles.
- *la modalité*, à savoir les canaux sensoriels au travers desquels transitent les informations du jeu (vue, ouïe, toucher, etc.). Cet aspect est important, il permet de soutenir l'expérience de jeu. Par exemple, [Consolvo et al., 2008] a montré qu'un soutien visuel au travers d'une application ludique mobile, consultée plusieurs fois par jour, permettait au participant d'une enquête de maintenir plus efficacement une activité physique journalière. Les modalités peuvent être utilisées comme renforcement positif ou négatif selon les normes du behaviorisme.
- *le style d'interaction* va de l'usage classique d'une souris, d'un clavier ou d'une manette de jeu à la mise en œuvre de technologies plus avancées comme la détection de mouvement. Dans les jeux immersifs les plus récents, le corps entier est utilisé, à l'aide de capteurs placés sur les articulations principales des bras, des jambes, etc. Même si de telles technologies ne sont pas d'application pour nous, une réflexion à ce niveau peut être importante : le fait d'utiliser la souris au lieu du clavier peut par exemple se révéler plus agréable, améliorant ainsi l'expérience de jeu.
- *l'environnement* fait référence à plusieurs critères d'importance :
  - visuellement, utilise-t-on la 2D, la 3D ou un mélange des deux ?
  - le jeu prend-il place dans un univers réel ou complètement virtuel ? Certains jeux font maintenant usage, avec l'utilisation de réalité augmentée, d'un mélange des deux. Un prolongement intéressant de ce travail consisterait par exemple à utiliser des modèles réduits de char réels, programmé au travers d'une interface de jeu.
  - la localisation physique. Par exemple, le jeu exige-t-il la présence sur un même site des différents joueurs ?
  - le jeu est-il mobile ou pas ?
  - le jeu se pratique-t-il en réseau ou pas ?
  - quelle est la dimension sociale du jeu ? Dans un contexte de jeu, il a été dé-

montré que des exercices collaboratifs et multijoueur sont plus engageants et motivants que ceux réalisés en solitaire [Yim, 2008].

- *le domaine d'application*. Ils sont nombreux, et ne cessent de se multiplier. On trouve par exemple les soins de santé, le bien-être, la culture, les interactions sociales. En 2009, les domaines les plus représentés étaient l'éducation (26%) et la publicité (31%). Même s'il s'agit d'un indicateur anecdotique, ces chiffres laisse suggérer une certaine pertinence de l'usage des jeux dans le domaine de l'éducation.

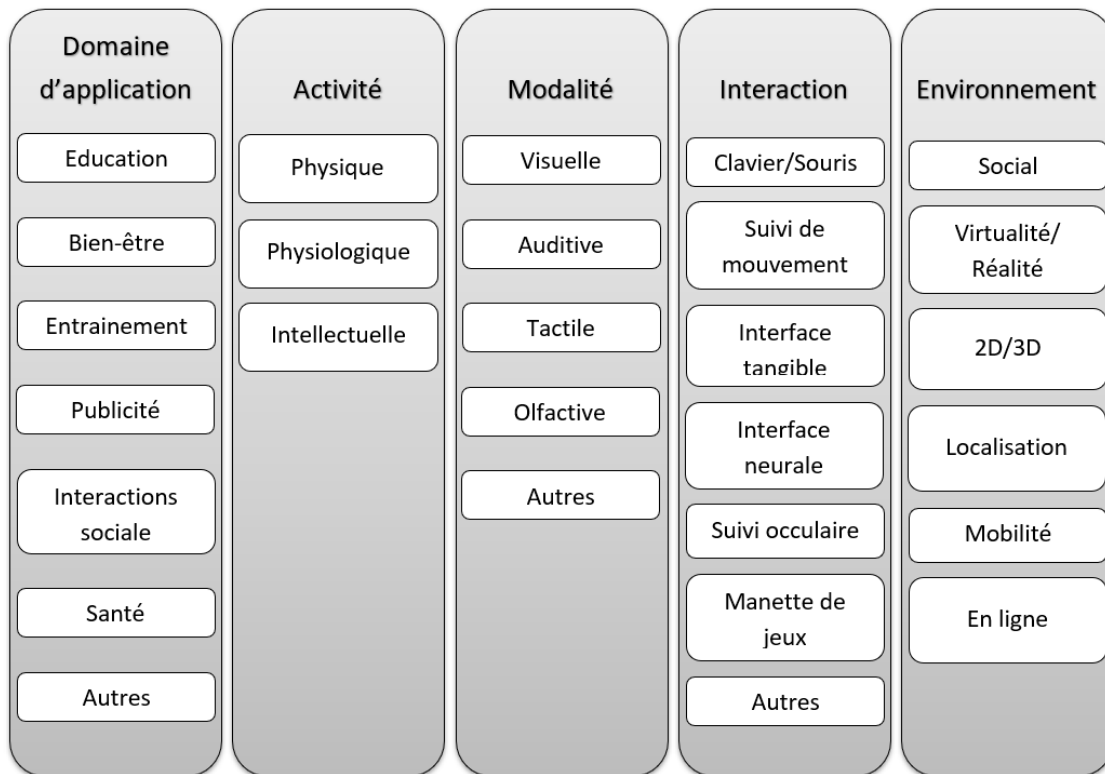


FIGURE 2.5 – Taxonomie des jeux sérieux selon [Laamarti et al., 2014]

### 2.3.5 Le modèle MDA

Le modèle MDA s'applique au développement et à l'analyse formelle des jeux vidéo. La notion de « jeu vidéo » est probablement familière à tous. Elle a beaucoup évolué au fil du temps, des endroits et continue de le faire. Avant l'avènement des ordinateurs personnels, il s'agissait de jeux pratiqués sur bornes d'arcade puis sur des consoles de jeu personnelles. Avec la généralisation des ordinateurs, le développement de nombreuses technologies applicables au domaine et l'expansion croissante des marchés concernés, le monde des jeux vidéo est devenu extrêmement vaste et complexe.

Au regard de la dimension ludique, on peut s'attendre à retrouver les éléments suivants en accord avec la définition d'un jeu, présents d'une manière ou d'une autre et à des degrés divers. :

- du conflit : le joueur doit être confronté à une forme d'adversité. Des joueurs uniques ou multiples, réels ou virtuels peuvent tenir ce rôle. Il peut aussi s'agir de circonstances spécifiques, aléatoires par exemple.

- des règles définissant avec précision ce qui peut être fait ou non, à quel moment, par qui, etc.
- l'exploitation de certaines qualités des joueurs comme l'habileté, la dextérité, la mémoire ou encore des aptitudes plus complexes comme la stratégie par exemple.
- une valorisation des résultats obtenus. Les modalités de cette valorisation peuvent être très différentes : victoire et défaite, gain d'argent, amélioration de performances personnelles, etc.

Du côté technique, la mise en application des règles, la gestion éventuelle des points et l'affichage des éléments visuel devraient être délégués entièrement à l'ordinateur. Celui-ci peut également incarner l'un ou plusieurs de protagonistes du jeu.

Les multiples interactions entre les sous-systèmes conçus peuvent laisser apparaître de nombreux comportements complexes, le plus souvent dynamiques et imprévisibles. Les concepteurs doivent prendre conscience de ces interdépendances durant la conception des jeux, et les chercheurs en tenir compte lorsqu'il s'agit de tirer des conclusions scientifiques quant aux expériences étudiées.

C'est partant de cette réflexion qu'a été établi le cadre de travail MDA, pour Mechanics, Dynamics and Aesthetics [Hunicke et al., 2004]. MDA se veut une approche formelle de la compréhension des jeux comblant les fossés entre la conception et le développement du jeu, leur analyse critique de la part des utilisateurs et leur utilisation comme outils de recherche.

Le postulat de base de l'approche MDA est qu'un jeu est un objet de consommation comme un autre qui est produit par ses concepteurs, consommé par ses utilisateurs, et éventuellement même jeté au final. À la différence d'autres produits de divertissement comme des films ou des livres, la manière dont les jeux sont consommés est relativement imprévisible : les événements survenant, leurs enchaînements et la manière dont ils sont vécus ne sont certainement pas maîtrisés lorsque la conception du jeu est terminée.

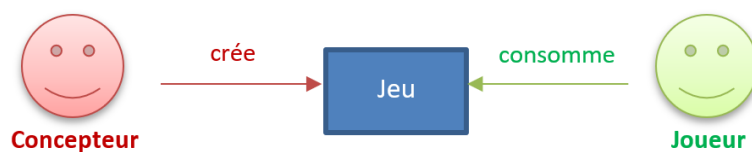
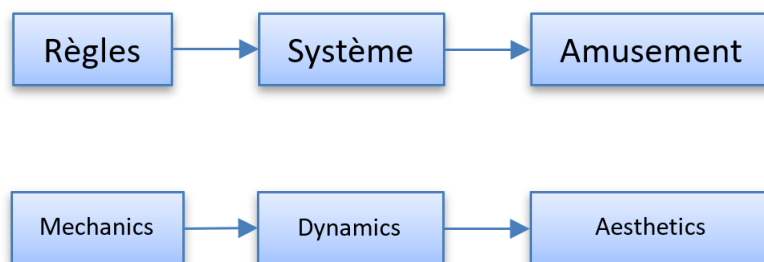


FIGURE 2.6 – Le jeu comme produit de consommation selon [Hunicke et al., 2004]

Le modèle MDA formalise la consommation des jeux à l'aide d'un découpage en trois éléments et en établissant les éléments de conception correspondants :



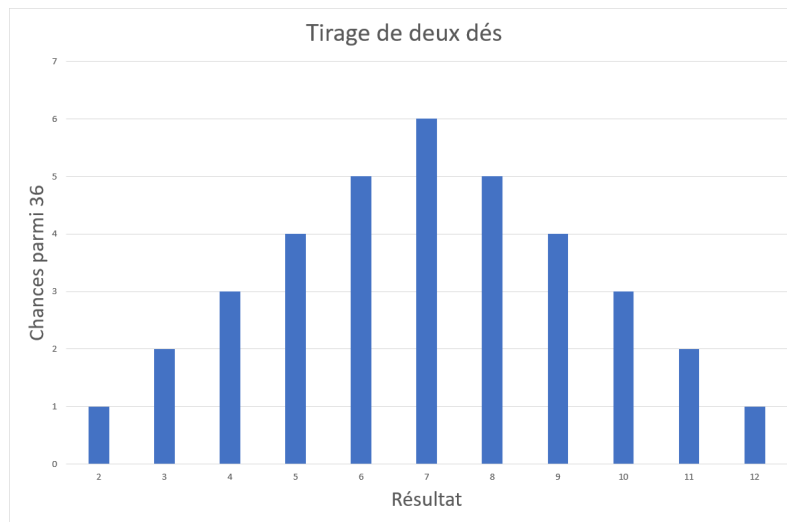
**L'esthétique** d'un jeu fait référence aux sensations éprouvées par le joueur, au plaisir de jouer, généralement défini comme « gameplay » ou « jouabilité ». Au-delà de ces termes, il est intéressant d'établir une terminologie plus étoffée afin de poser un cadre d'analyse plus formel. À cette fin, [Hunicke et al., 2004] propose une taxonomie de base fondée sur les notions majeures suivantes :

- Sensation. C'est une composante essentielle d'un jeu qui cherche à susciter des émotions chez le joueur. La conception d'un jeu d'horreur comme Resident Evil fait appel à la peur, à l'angoisse.
- Fantasy. Il s'agit de l'immersion dans l'imaginaire. Dans la série des Final Fantasy, les personnages incarnés sont en quête de cristaux magiques leur permettant d'acquérir des pouvoirs afin de dominer le monde.
- Narratif. Le joueur prend part au déroulement d'une histoire plus ou moins scénarisée. Dans Age of Empires, l'évolution des civilisations au cours des siècles sert de cadre où le joueur revit de grandes conquêtes, l'évolution des technologies, etc.
- Challenge. Des obstacles surviennent, qu'il faut surmonter. Résoudre un challenge peut être satisfaisant en soi, ou déboucher sur une forme de récompense, de renforcement positif stimulant le joueur face à des difficultés croissantes. Dans LightBot, il s'agit par exemple de programmer un robot pour le faire évoluer au cœur de labyrinthe de plus en plus évolués.
- Camaraderie. Une dimension sociale est intégrée. Des jeux de rôle en ligne massivement multijoueur comme World of Warcraft reposent entièrement sur des milliers de joueurs qui s'entraident où se combattent.
- Découverte. Un espace de jeu inconnu est exploré. Il peut s'agir de révéler la géographie d'un monde imaginaire, mais aussi de voir apparaître de nouvelles actions, de nouvelles capacités. Dans un jeu de stratégie comme Command and Conquer, il faut établir des bases militaires en ne connaissant à priori ni la configuration du terrain, ni la position de l'adversaire. Au fil d'une, il est aussi possible de construire de nouvelles armes ou de nouveaux véhicules.
- Expression. C'est la part de créativité laissée à l'utilisateur. Dans le jeu de blocs Minecraft, des briques de base servent à bâtir des édifices d'une variété extraordinaire. En allant encore plus loin, certains joueurs ont même détourné les éléments de construction pour reproduire l'architecture et des prototypes fonctionnels d'ordinateurs.
- Soumission. Cette dimension s'oppose quelque part à celle du challenge. Le but poursuivi est le délasserment, la récréation. Dans un jeu comme Farmville, le joueur exploite des fermes virtuelles en cultivant ses champs, en élevant des animaux, etc. Il n'y a pas de réelle difficulté à surmonter. L'esprit se met en quelque sorte au repos, et c'est de là que le joueur tire son plaisir.

Cette liste n'est pas exhaustive, et la majorité des jeux incluent à divers degrés plusieurs des éléments décrits. Le vocabulaire ainsi établi permet de concevoir un modèle esthétique qui servira à mieux percevoir la manière dont vont émerger la dynamique et la mécanique des jeux : un jeu comme World of Warcraft, à forte connotation compétitive, a du succès car les participants sont *émotionnellement impliqués*. On peut dès lors envisager qu'un feedback clair sur qui est en train de gagner, un scénario où les chances de victoire restent présentes le plus longtemps possible sont des facteurs qui garantiront l'attrait de l'ensemble.



**La modélisation dynamique** intervient pour susciter l'esthétique désirée. L'utilisation d'un minuteur lors de résolution d'un puzzle, augmentera sans doute la perception de challenge. L'intégration d'un système de communication vocale destiné aux équipes dans un jeu en ligne massivement multijoueur mettra l'accent sur l'élément de camaraderie. Développer une modélisation concrète qui prédit et décrit le gameplay permet d'éviter des erreurs de conception. Un exemple est donné par l'analyse probabiliste des résultats d'un tirage de deux dés, qui permet la création judicieuse d'un plateau où évolueront les pions.



De manière similaire, dans un jeu comme le Monopoly, l'observation de la dynamique du jeu permet de noter que l'enrichissement des joueurs dominant une partie rend ceux-ci encore plus forts, induisant un cercle vicieux qui risque de créer le désintérêt des joueurs les plus faibles.

**La mécanique** décrit les constituants monolithiques d'un jeu, ainsi que les règles qui régissent leurs comportements et interactions. La mécanique vient supporter la création de la dynamique et de l'esthétique. Dans l'exemple du Monopoly évoqué plus haut, le plateau de jeu, les cartes de chance, les billets, etc. forment avec les règles les mécanismes. Pour lutter contre l'appauvrissement inexorable des joueurs dominés, l'adjonction d'une règle laissant à ces derniers une chance de gagner une grosse somme d'argent pour les relancer dans la partie pourrait lutter contre cet effet. Avec des parties de poker en ligne, le sabot des cartes, le mélange de celles-ci, le système de donne et de mise permet l'émergence de comportements dynamiques comme le bluff.

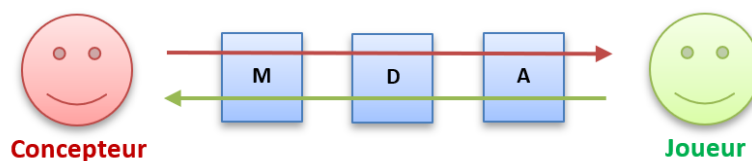


FIGURE 2.7 – Vue synthétique du modèle MDA [Hunicke et al., 2004]

### 2.3.6 La conception des SG

La conception des SG prend nécessairement en compte différents éléments, c'est-à-dire :

- « des scénarios d'apprentissage [Howell, 2005, Raybourn, 2007],
- les caractéristiques individuelles des apprenants telles que leur âge [Zyda, 2005],
- leurs connaissances antérieures sur le sujet de formation [Ratan and Ritterfeld, 2009] ».
- Des facteurs liés au produit SG portant sur la dimension ludique et ses propriétés (flow, engagement, immersion, interactivité, plaisir du jeu, etc.) »

**Le flow** Le terme flow désigne « un sentiment de satisfaction et de plénitude dans la réalisation d'une activité pour laquelle toute l'attention se porte sur la tâche en cours » [Cohard, 2015, p.11]. Les expériences en termes de flow sont caractérisées par « une forte concentration sur une tâche possédant des objectifs clairs et fournissant un feedback immédiat, donnant un sentiment de contrôle de ses actes et produisant une perte de la conscience de soi et du temps » [Cohard, 2015, p.11]. Le flow est ainsi associé à un haut niveau d'engagement « qui nécessite de la part des concepteurs, la capacité de transmettre via le jeu vidéo un message appréhendable par l'utilisateur à travers ses propres caractéristiques cognitives et sociales » [Sherry, 2004, 27].

**L'engagement** L'engagement quant à lui est une forte implication dans l'utilisation du jeu, qui au contraire de l'addiction, a des conséquences positives [Charlton and Danforth, 2010]. Enfin, la présence est le sentiment d'être dans un environnement donné, même si en réalité, la personne reste physiquement dans un autre environnement. La présence est liée à l'engagement, plus particulièrement à l'implication, mais aussi à l'immersion qui est l'état psychologique caractérisé par le fait de se percevoir comme enveloppé, inclus dans et en interaction avec un environnement qui fournit continuellement un flux de stimuli et d'expérience [Witmer and Singer, 1998]. Ainsi les qualités multimédia d'un SG ne peuvent suffire; elles sont des conditions nécessaires, mais pas suffisantes de leur réussite. [Cohard, 2015]

### 2.3.7 Les jeux sérieux éducatifs

#### Avantages pédagogiques des SG

Les avantages des SG pour l'éducation (SGE) sont nombreux. Selon [Pivec and Pivec, 2008] les principaux avantages des SG concernent :

- La motivation des apprenants
- L'apprentissage par essais et erreurs
- la prise en compte des différents rythmes d'apprentissage
- La stimulation d'interactions pédagogiques entre élèves

**La motivation des apprenants** Un des avantages principaux de l'usage du SG dans l'enseignement réside dans son « impact globalement positif sur la motivation des élèves ». Il est à noter que la motivation de l'apprenant ne s'explique pas seulement par l'aspect « nouveauté » de l'introduction du jeu en classe. En effet, « les quelques études sur l'utilisation de jeux sur le long terme montrent bien que l'objet

SG en tant que tel peut apporter un gain de motivation pour les élèves » ([Malone, 1981, Wastiau et al., 2009, Wix, 2012]).

**L'apprentissage par essais et erreurs** Le deuxième avantage des SG concerne « la mise à disposition de l'apprenant d'un espace d'expérimentation dans lequel il est invité à exercer ses capacités à réfléchir » [Djaouti, 2016, p.51]. La grande majorité des SG se basent effectivement sur un mode d'apprentissage par essais et erreurs. Dans celui-ci, l'apprenant « construit mentalement une « hypothèse » avant de la tester dans le jeu, qui lui donne alors un retour positif ou négatif » [Djaouti, 2016, p.51]. Il revient dès lors au joueur d'affiner continuellement son hypothèse jusqu'à trouver la solution et ainsi gagner la partie ou le jeu. Dans un bon SG, les joueurs disposent ainsi des informations nécessaires pour leur permettre de construire une hypothèse juste par eux-mêmes. [Sanchez, 2011].

**La prise en compte des différences de rythmes d'apprentissage** Il est à noter que le recours au SG offre également certains avantages aux enseignants. Il leur permet effectivement d'identifier et de considérer les différences en termes de rythme d'apprentissage entre élèves d'un même groupe [Kafai et al., 1995]. Les SG dans l'enseignement offrent effectivement la possibilité à chaque élève de progresser à son rythme. Les élèves peuvent ici recommencer autant de fois que nécessaire avant de trouver la solution sans avoir peur d'un jugement de ses pairs. L'élève qui réussit plus rapidement peut, quant à lui, poursuivre le jeu sans être obligé d'attendre ses camarades. [Djaouti, 2016, p.51].

**La stimulation d'interactions pédagogiques entre élèves** Certains SG favorisent la stimulation des interactions pédagogiques entre élèves. En effet « le travail d'équipe et la collaboration peuvent être encouragés par les SG [Allal-Cherif and Bajard, 2011], notamment à travers la création d'une dynamique de groupe autour du jeu. Ils peuvent ainsi encourager « la communication et la collaboration entre élèves, aussi bien en ligne qu'en personne » [Djaouti, 2016, p.51]. Dans ce type de jeux « les élèves s'expliquent mutuellement des concepts, se félicitent lorsqu'ils réussissent ou échouent et se soutiennent lorsqu'ils prennent des risques » [Djaouti, 2016, p.52].

### **Le SG et tant qu'outil pédagogique**

De manière tout à fait générale peut être considéré tel un outil supplémentaire dans la trousse du formateur. Lorsqu'il est utilisé à bon escient, le SG peut sensiblement améliorer l'acquisition de connaissances et des de compétences des apprenants. Une bonne maîtrise de l'outil de la part du formateur s'avère en tout cas essentielle pour concevoir une utilisation efficace de ce « nouveau » support pédagogique.

Il est, pour l'enseignant, par ailleurs, l'essentiel définir des thématiques à aborder les thématiques et d'identifier préalablement les connaissances pour lesquelles le SG peut s'avérer pertinent. Ceci lui permettra d'intégrer intelligemment le SG dans son cours.

L'intégration d'un SG dans une pratique pédagogique peut prendre trois formes distinctes. Le formateur peut ainsi effectivement [Djaouti, 2016, p.52] :

- « Utiliser des SG existants avec ses étudiants.
- Créer des SG « sur mesure » pour ses étudiants.

- Faire créer des SG directement par ses étudiants »

**Utiliser un SG existant** La première approche pouvant être mis en place par le formateur consiste à utiliser des SG existants avec ses étudiants. Il peut ici s'agir d'un SG existant, commercial ou gratuit. Dans ce premier cas, le formateur ne dispose pas de la possibilité de modifier ce support pédagogique et doit dès lors prendre le parti d'adapter son cours en fonction de l'utilisation de l'outil.

La sélection du SG est ici de très grande importance. L'outil doit idéalement :

- être pertinent pour l'objectif pédagogique visé
- être accessible aux apprenants

Il est à noter que le choix d'un SG peut s'avérer particulièrement ardu pour différentes raisons. Ceci en premier lieu par ce que « l'éducation est plutôt pauvre en matière de jeux pédagogiques, en particulier pour les jeux en français » [Michaud and Alvarez, 2008]. Il faut dire que le cours de réalisation d'un SG par une entreprise spécialisée avoisine les 150.000 € [Michaud and Alvarez, 2008]. Le financement de ce type de jeu par les systèmes éducatifs nationaux francophones constitue rarement une priorité.

Il est toutefois à noter que de nombreux SG ont été pensés pour l'éducation francophones existent, à l'image du jeu « Les Eonaves » sorti en 2012 émis à l'arrêt le 15 avril 2020, initialement conçu pour la formation au Français Langue Etrangère (FLE).

Outre les SG pensés pour l'éducation, d'autres SG destinés à d'autres marchés peuvent toutefois également être intégrés dans une activité pédagogique. C'est notamment le cas pour Stop Disasters (Playerthree, 2007).

Il s'agit d'un jeu diffusé gratuitement par l'ONU qui a pour objectif de « sensibiliser le grand public aux catastrophes naturelles, et de lui transmettre les rudiments de leur prévention de gestion « dans lequel le joueur prend les commandes d'un village menacé par une catastrophe naturelle imminente (tsunami, tremblement de terre. . .) » [Michaud and Alvarez, 2008].

Les informations contenues dans Stop Disasters « rendent ce jeu potentiellement intéressant pour l'éducation, par exemple dans un cours de Géographie » [Michaud and Alvarez, 2008].

**Créer un SG « sur mesure »** Il n'existe actuellement pas encore de SG pour couvrir toutes les thématiques pédagogiques. Il est pour ces dernières, dès lors envisageables, de créer un SG sur mesure pleinement adapté à la pratique pédagogique du formateur. Deux possibilités sont, le cas échéant, envisageable : Dans la première situation, le formateur arrive regrouper une somme suffisamment importante pour lui permettre de construire son projet avec des professionnels du secteur. Dans ce cas, il est loisible de

- laissant le travail technique de réalisation du jeu à des spécialistes
- définir les objectifs pédagogiques du jeu
- proposer des idées de jeu
- participer à l'évaluation du SG

La conception d'un tel projet peut fortement varier en fonction de différents paramètres :

- Complexité technique : Le coût du SG peut ainsi varier en fonction de sa

complexité technique, notamment en fonction qu'il s'agisse d'un « jeu mono-joueur ou multijoueur en réseau, en 2D ou en 3D.

- Richesse du contenu : Le coût du SG peut ainsi varier en fonction de la richesse de son contenu, notamment en fonction « du nombre de niveaux, de quêtes, de dialogues, de personnages, d'objets... » [Michaud and Alvarez, 2008, p.35].
- La plate-forme de destination : Le coût du SG peut enfin varier en fonction de la plate-forme de destination, notamment, s'il s'agit d'un « ordinateur, smartphone, console... » [Michaud and Alvarez, 2008, p.35].

La réalisation d'un tel projet varie « de quelques dizaines de milliers d'euros pour un SG « simple », par exemple un petit jeu en 2D sur téléphone mobile, à plus de 500.000 € pour un jeu complexe, par exemple une simulation réaliste en 3D » [Michaud and Alvarez, 2008, p.35]. Le formateur est dans la plupart des situations amené à travailler avec un budget bien trop restreint pour pouvoir faire appel à des professionnels. Une autre option consiste à rendre lui-même en charge la réalisation du SG. Il est à noter que cette option se veut beaucoup plus chronophage. Le résultat s'en veut d'ailleurs nécessairement plus modeste d'un point de vue technique.

Pour créer le SG, les formateurs peuvent soit :

- s'appuyer sur leurs propres connaissances
- « se tourner vers des logiciels de création vidéoludique simple d'accès ».

**Les usines à jeux** Le choix d'un logiciel de création vidéoludique, également appelée usine à jeux, n'est pas toujours chose évidente. Les logiciels de création, ou usines à jeux, permettent déjà après « une séance de deux heures de présentation d'une usine à jeux à des enseignants ou à des étudiants de commencer à l'utiliser pour créer leur propre SG » [Djaouti, 2016, p.60].

Il est toutefois à noter que « Les nombreuses « usines à jeux » existantes possèdent toutes leurs contraintes propres » [Djaouti, 2016, p.60]. Ainsi, par exemple, certains de ces logiciels ne permettent que la création « de certains types de jeux particuliers, tandis que d'autres posent des limites fortes sur la dimension technique du jeu (pas de 3D, pas de multijoueur, etc.) » [Djaouti, 2016, p.60]. L'usine à jeux doit en tout cas être adaptée :

- au niveau de maîtrise informatique de l'enseignant
- à la complexité de son projet
- au temps dont il dispose.

**Le modding** La création d'un SG peut également se concevoir par la modification d'un jeu vidéo déjà existant. Cette modification de jeu est appelée « modding ». Nombre de nombreux jeux vidéo de divertissement sont à ce propos « fournis avec des outils permettant au joueur de les transformer, tant qu'il reste dans un cadre défini par les créateurs originels du jeu » [Djaouti, 2016, p.61]. Dans ce cas de figure, le formateur gagne un temps considérable étant donné qu'il ne part plus d'une page blanche. Cette solution lui permet, par ailleurs, également de « bénéficier de l'expérience des créateurs du jeu original en matière de « game design » » [Djaouti, 2016, p.61]. Il est à noter que le modding est, de manière générale, « destinée à des fins purement récréatives » [Djaouti, 2016, p.62]. Il peut néanmoins tout à fait « être utilisée pour transformer un jeu vidéo de divertissement en SG » [Monterrat et al., 2012, p.25].

**Faire Créer des SG par les apprenants** Le formateur peut utiliser les « usines à jeux » et « jeux modifiables » pour imaginer des activités pédagogiques « dans lesquelles les apprenants créent par eux-mêmes un SG » [Djaouti, 2016, p.63]. Le formateur peut, ici, par exemple demander à ses étudiants de créer SG, seul ou en petits groupes.

### Les effets des SG éducatifs sur l'apprentissage et la motivation

Les SG éducatifs (SGE) constituent des supports numériques particulièrement intéressants pour l'apprentissage en général, et plus particulièrement pour l'apprentissage à distance. Ce média peut effectivement « amener les apprenants à s'engager des processus cognitifs plus complexes grâce à son caractère divertissant et motivant » [Erhel and Jamet, 2012, p.248]. Les apports des SGE en matière d'apprentissage restent encore très discutables.

Il n'existe effectivement à ce jour pas de preuve qui permette d'affirmer que les SG constituent une méthode à privilégier dans l'ensemble des situations d'apprentissage. [Hays, 2005].

Cette position est également partagée par différentes revues consacrées aux simulations interactives aux jeux éducatifs. Ainsi, en 1992, [Randel et al., 1992] ont recensé 68 études dans lesquelles étaient comparés les SGE à des documents conventionnels.

Seules 22 études d'entre elles concluaient que les SGE étaient plus bénéfiques en matière d'apprentissage. Malgré ces conclusions mitigées les auteurs de l'enquête « observent un réel intérêt des jeux éducatifs / simulations en termes de motivation ».

Dans une enquête similaire, [Hays, 2005] parvient à des conclusions similaires. Ainsi, l'analyse de 48 études portant sur les bénéfices des jeux éducatifs et des simulations, « cet auteur remarque que ces media ne favorisent pas systématiquement le transfert de connaissances comparés à des cours ou des supports conventionnels ». [Hays, 2005] remarque néanmoins, comme [Randel et al., 1992], que les SGE peuvent être particulièrement bénéfiques sur au niveau de l'intérêt et la motivation de l'étudiant.

Enfin, l'analyse de de [Kebritchi et al., 2010], mesurant 16 études concernant l'efficacité pédagogique des jeux éducatifs confirme les résultats des travaux précédents qui attestent les bénéfices mitigés des SGE en matière d'apprentissage. Ainsi, seules 9 sur 16 des études analysées « permettent de conclure à une amélioration de la qualité des apprentissages comparée à des cours conventionnels ».

L'étude concernée démontre, par ailleurs, contrairement aux études précédentes « un faible bénéfice des SGE en matière de motivation ».

Selon d'autres revues, les SGE s'avèrent toutefois plus bénéfiques en termes d'apprentissage. Ainsi, une méta-analyse de [Vogel et al., 2006] qui regroupe 32 études tend à démontrer que les simulations interactives et les jeux éducatifs ont un effet particulièrement positif en termes d'apprentissages comparativement à des méthodologies d'enseignements plus traditionnelles.

Selon [Vogel et al., 2006] l'attitude des apprenants serait d'ailleurs également plus positive vis-à-vis des méthodes d'apprentissages « basées sur les jeux éducatifs / simulations que sur les méthodes d'apprentissage plus conventionnelles ».

Toujours selon les mêmes auteurs, les bénéfices des SGE peuvent varier en fonction de deux facteurs, c'est-à-dire :

- les caractéristiques des individus

— la situation d'apprentissage.

Les résultats des SGE en termes d'apprentissage semblent ainsi effectivement pouvoir varier en fonction du contexte dans lesquelles elles prennent place. Ainsi, dans le cadre d'un cours sur le fonctionnement de systèmes mécaniques [Coller and Scott, 2009] observent que les jeux vidéo éducatifs permettent d'aboutir à des performances significativement supérieures par rapport à des cours traditionnels.

[Kebritchi et al., 2010] obtiennent des résultats identiques dans une étude longitudinale « sur un jeu éducatif permettant l'apprentissage de notions de mathématique ». Les résultats de l'étude tendent ici encore à démontrer que les « jeu éducatif » permettent d'améliorer de manière significative les « compétences en mathématiques comparées à ceux de la condition « cours traditionnels ». Selon l'étude, le « jeu éducatif » permet de favoriser un investissement motivationnel plus important par rapport à celui observé pendant les cours traditionnels.

De la même manière, [Liu et al., 2011] « démontrent que les apprenants soumis à une simulation expérimentent plus volontiers une expérience de flow ce qui les amène à s'engager dans des stratégies plus pertinentes pour résoudre les problèmes ».

Ces dernières études peuvent ainsi laisser penser que les SGE ont un effet bénéfique sur l'apprentissage et la motivation. Ici encore, d'autres études viennent contredire ces résultats.

Ainsi, selon l'étude de [Annetta et al., 2009] un cours sur la génétique à l'aide d'un SGE permet nettement d'améliorer « le niveau d'engagement des apprenants dans la tâche comparée à un cours présentiel ». Toutefois, selon l'étude, « ces bénéfices ne se répercutent pas sur la qualité des apprentissages ». Les études comme celles de [Gredler, 2004] ou [Papastergiou, 2009] tendent à confirmer ces mêmes résultats.

Il semble ainsi, particulièrement ardu de conclure à une relation positive entre les SGE et la motivation. Cette même motivation, lorsqu'elle est présente, ne permettrait d'ailleurs pas nécessairement d'améliorer la qualité de l'apprentissage.

Les études sur les SGE donnent en tout cas lieu à des résultats pour le moins controversés. Les contradictions relevées dans les études présentées peuvent, selon certains auteurs s'expliquer par différents éléments, et plus particulièrement par :

- les caractéristiques des individus ( [Vogel et al., 2006]),
- la situation d'apprentissage [Ke, 2011, Vogel et al., 2006],
- les thématiques abordées au cours de l'apprentissage ( [Hays, 2005, Ke, 2011, Ke, 2011, Randel et al., 1992]).

Les études précédentes ne permettent donc pas d'affirmer la supériorité des SGE aux moyens pédagogiques traditionnels. La littérature permet toutefois d'identifier différents facteurs favorisant les bénéfices réels des SGE.

## Facteurs de performance des SGE

En ce qui concerne les SGE, différents facteurs tendent ainsi effectivement à favoriser les performances des apprenants. Il s'avère ainsi, par exemple que [Erhel and Jamet, 2012, p.251] :

- « la présence d'explications multimédia entre les phases d'un jeu permet d'améliorer les performances des apprenants en termes d'apprentissage ».
- « la mise à disposition d'informations sur le domaine abordé avant et pendant la simulation favorise, de même, un traitement plus pertinent du contenu pédagogique ».

- « l'utilisation de questions pertinentes par rapport aux objectifs d'apprentissage » favorise également l'amélioration de la qualité des acquisitions ».
- « l'utilisation d'un style conversationnel peut améliorer la rétention et le transfert de connaissances ».
- « l'ajout d'un agent pédagogique dans un jeu portant sur la botanique améliore significativement les performances des apprenants en matière de transfert de connaissances ».
- « les individus apprennent mieux les informations dans un SGE lorsqu'ils sont incités à expliquer leurs réponses ».

**Feedback** La présence de feedback constitue l'un des facteurs les plus étudiés en matière de SGE . Parmi les différentes études qui y sont consacrées, « l'une des plus connues est celle de Leutner (1993) qui démontre que la présence des feedbacks dans un SGE consacré aux principes écologiques améliore la qualité des apprentissages » [Erhel and Jamet, 2012, p.251]. Il est toutefois à noter que tous les feedbacks n'engendrent pas le même type de résultats.

**Feedbacks informatifs et correctifs** Les résultats d'une étude de [Moreno, 2004] ont permis de démontrer que les feedbacks explicatifs sont efficaces que les feedbacks correctifs lorsqu'il s'agit favoriser le transfert de connaissances. Ainsi, en utilisant le SGE « Design a plant » [Moreno and Mayer, 2000] ont réussi à démontrer que, comparativement aux feedbacks correctifs, les feedbacks explicatifs permettent d'obtenir :

- « des scores plus élevés en matière de transfert,
- moins de réponses incorrectes
- moins d'idées fausses au cours de la résolution de problème » [Erhel and Jamet, 2012, p.252].

De la même manière, une étude de [Mayer and Johnson, 2010] basée sur un cours portant sur la résolution de problèmes avec des circuits électriques permet d'attester « qu'une version incorporant des feedbacks explicatifs conduit à de meilleures performances en matière de transfert et à une efficacité plus importante qu'une version comportant des feedback correctifs ».

### Limites du SG pour l'éducation

L'utilisation des SGE est efficace pour la transmission de connaissances. Leur mise en pratique ne s'avère pourtant pas toujours « plus efficace que d'autres approches pédagogiques » [Djaouti, 2016, p.52]. Ils peuvent dans certaines situations d'ailleurs même s'avérer contre-productifs, ceci pour plusieurs raisons :

- L'utilisation de SG non pertinents
- L'absence d'intégration du SG au travail de l'enseignant
- Des contraintes matérielles et logistiques

**L'utilisation de SG non pertinents** La limite la plus importante du SG à visée pédagogique réside dans la pertinence du choix du jeu. [Djaouti, 2016, p.55] Il est d'ailleurs à noter que les SG peuvent présenter des qualités très variables en fonction des concepteurs. La sélection du SG le plus pertinent doit être réalisé par l'enseignant en fonction :



- « de ses étudiants
- de ses objectifs pédagogiques
- de ses méthodes de travail » [Djaouti, 2016, p.54].

**L'absence d'intégration du SG au travail de l'enseignant** La majorité des études réalisées sur l'utilisation efficace des SG en classe mettent en lumière l'importance du rôle de l'enseignant. En effet, « La simple mise à disposition d'un jeu ne semble pas changer grand-chose à ce que les élèves apprennent. [...] La médiation de l'enseignant, qui pense la place du jeu dans une démarche d'apprentissage, reste donc indispensable » [Djaouti, 2016, p.55]. Il s'avère ainsi d'après des expérimentations menées par [Habgood, 2007], qu'un SG, s'avère bien plus efficace en termes d'acquisition de connaissances si l'enseignant effectue un « débriefing » après la séance avec les étudiants.

**Des contraintes matérielles et logistiques** Notons enfin que l'implication logistique des SG peut également constituer une contrainte des jeux vidéo en classe [Wix, 2012]. En effet, les limites de la mise en pratique de ce type de jeu peut également résider dans la « disponibilité des ordinateurs ou des consoles capables de faire tourner le jeu ou l'autorisation de la direction de l'établissement pour l'achat éventuel de logiciel ou matériel » [Djaouti, 2016, p.55].

# Chapitre 3

## Méthodologie

### 3.1 Le design expérimental d’environnements d’apprentissage

Pour rappel la question qui anime notre recherche concerne « *les impacts d’un jeu vidéo multijoueur sur un cours d’initiation à la programmation informatique suivi par des étudiants de promotion sociale ?* ». Nous sommes particulièrement intéressés par les bénéfices didactiques obtenus par le public adulte au travers de la conception du jeu.

Pour mener notre recherche, nous avons choisi une approche méthodologique bien accueillie dans la recherche sur les technologies éducatives [Class and Schneider, 2013] qui s’enracine dans un postulat défendu par Stokes [Stokes, 1997] selon lequel il existe une « recherche fondamentale inspirée par de l’utilisation ». Elle se rencontre sous diverses dénominations anglaises, comme *design experiments*, *educational design research* ou *design based research*. La traduction française la plus pertinente à notre sens est celle proposée par [Class and Schneider, 2013], que nous utiliserons : Recherche Design en Éducation (RDE).

La RDE est une méthodologie itérative reposant sur l’intégration de dispositifs innovants au sein de situation concrètes et complexes d’apprentissage. Les principes de design de ces dispositifs sont l’objet de la recherche, et leur impact sur les processus éducatifs au cœur desquels ils sont mis en œuvre est évalué. Les dispositifs étudiés étant souvent novateurs, comme dans le cas d’artefacts logiciels, Il s’agit plutôt de tenter d’élaborer de nouvelles théories plutôt que d’en évaluer d’existantes. L’analyse des liens entre les principes de design et leur influence mène à une amélioration du dispositif étudié, d’où la nature itérative de la méthodologie.

La méthode emprunte à l’ingénierie du design comme décrit à la figure 3.1, mais aussi à la recherche fondamentale pour produire des résultats théoriques.

Allan Collins [Collins, 2010] énumère une liste détaillée des différentes étapes constituant un cycle.

- Identification d’un problème par un chercheur ou un praticien,
- Analyse de la situation pour comprendre les sources du problème,
- Revue de la littérature théorique relative au problème,
- Exploration de solutions potentielles et innovantes,
- Prise en considération de plusieurs options possibles,
- Création d’un design à mettre en œuvre en situation réelle,
- Développement d’un prototype et mise en service du design,

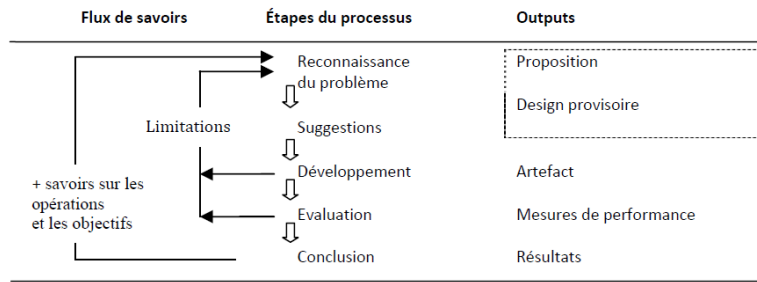


FIGURE 3.1 – Science du design, méthodologie générale [Järvinen, 2007], traduit par [Class and Schneider, 2013]

- Évaluation, avec un double but : l’affinement du design et des conjectures théoriques sur lesquelles ce dernier était fondé.

Suite à l’observation de nombreuse recherche en RDE, [McKenney and Reeves, 2014] ont élaboré un modèle générique pour guider la conduite d’une telle recherche. (figure3.2).

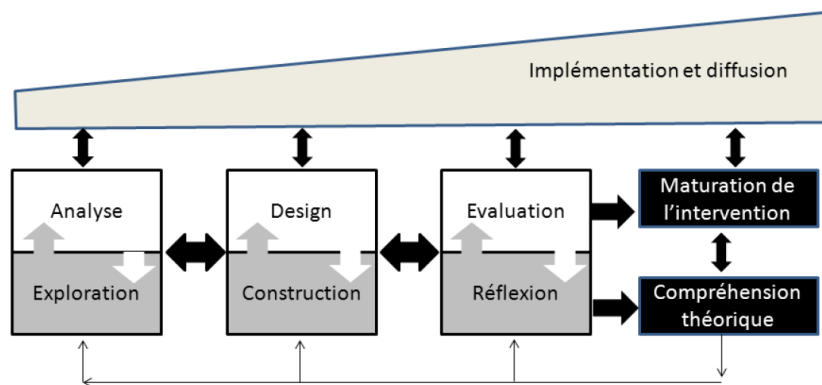


FIGURE 3.2 – Déroulement général d’une recherche RDE [McKenney and Reeves, 2014]

Nous décidons de suivre les grandes lignes de ce qui précède pour définir les étapes de la partie pratique de ce mémoire.

## 3.2 Étapes de la recherche

### 3.2.1 Identification du problème à résoudre

L’identification du problème que l’on souhaite adresser est du ressort du praticien ou du chercheur. Dans notre cas, il s’agit de la même personne. Elle constate que le public suivant les cours d’initiation à la programmation rencontre de nettes difficultés à acquérir la maîtrise de concepts fondamentaux. S’agissant d’un public adulte en reprise de formation, il est important d’en tenir compte pour éviter dévalorisation, découragement ou échec.

### 3.2.2 Analyse de la situation

La situation mise en évidence est la suivante : lors des premiers contacts avec la programmation, la compréhension des concepts d'instruction et de séquentialité ne présente généralement pas de difficulté majeure. En revanche, l'apparition des structures de contrôle, conditionnelles et itérative est nettement plus problématique, générant par ailleurs découragement et même renoncement.

L'observation de la dynamique du cours met en évidence plusieurs facteurs pouvant expliquer ce constat :

- l'étudiant comprend les structures itératives et conditionnelles selon le langage naturel mais n'appréhende pas leur acceptation algorithmique, plus rigoureuse et formelle. Il a notamment du mal à voir le test logique comme une opération mathématique fournissant un résultat binaire,
- la difficulté à appréhender l'essence de cette brique de base l'empêche de formuler le problème à résoudre sous une forme compréhensible par l'automate que constitue l'ordinateur,
- le public considéré dans cette recherche perçoit cette difficulté comme une lacune dévalorisante, souvent attachée à un manque de logique ou une faiblesse en mathématiques,
- au final, ces éléments, combinés aux rigueurs pragmatiques du développement comme l'environnement ou la syntaxe d'un langage textuel découragent l'apprenant, le mettent dans une situation de détresse menant souvent à l'abandon en cours de formation ou à l'échec.

Il est à noter que ces différentes observations sont corroborées par l'état de l'art établi précédemment dans ce travail.

### 3.2.3 Prospections préliminaires

Notre recherche littéraire, dans la partie consacrée à la programmation, définit et met en évidence le rôle fondamental des compétences de base. Elle souligne également la transversalité de ces compétences dans la pensée informatique. La seconde partie consacrée à l'apprentissage chez l'adulte montre la nécessité de tenir compte de ses spécificités et introduit divers éléments représentatifs issus de l'andragogie. Enfin, la troisième partie fournit des indications sur la conception des jeux sérieux, notamment éducatifs.

En questionnant cet état de l'art, nous procédons à une recherche d'un SG existant qui serait adapté à notre recherche. Plusieurs pistes s'offrent à nous, mais leur exploration montre assez rapidement leurs limitations. Citons notamment :

- **Lightbot** : Le but est de programmer un petit robot afin qu'il se déplace dans un labyrinthe et allume des lumières (figure 3.3). Les joueurs disposent de symboles de base à combiner, constituant les instructions pour piloter le robot (marcher, tourner, sauter, allumer une lumière, etc.) Les labyrinthes se compliquent et la liste des symboles s'étoffe au fur et à mesure que les leçons progressent. Les joueurs apprennent ainsi les concepts de programmation tels que les boucles ou les procédures, sans avoir à rédiger de code dans un langage de programmation textuel. Ce jeu ne possède hélas aucune dimension multijoueur intrinsèque.
- **Robocode** : ce jeu est destiné à l'apprentissage du langage Java. Il s'agit de programmer des tanks pour qu'ils agissent comme des intelligences artificielles

lors de combat d'arène (figure 3.4). Il est possible de déployer ce jeu sur un réseau local, mais en raison du langage utilisé et des nombreux concepts techniques mis en œuvre, la complexité apparaît très vite et rend cette solution difficilement viable.

- **JSBattle** : c'est une version plus abordable et conviviale de Robocode, où le langage utilisé est Javascript et les données de jeu moins nombreuses. Il est possible de jouer en ligne ou de déployer le jeu localement. Malgré un niveau plus accessible, l'usage d'un langage textuel et la nécessité de comprendre des concepts de trigonométrie sont aussi rédhibitoires.
- **Codingames** : il s'agit d'une plateforme, également accessible à d'éventuels recruteurs, entièrement dédiée à l'apprentissage de la programmation au travers de jeux vidéo (figure 3.5). La variété des jeux et des modalités est énorme. Il existe de nombreux modes multijoueur (challenges de type *blitz*, combats en ligne, etc.). Des canaux « discord » sont intégrés à la plateforme permettant une communication aisée entre joueurs. L'obligation de passer par un des nombreux langages textuels supporté et le niveau élevé de nombreux concurrents sont les seuls obstacles qui ont empêché l'usage de cette plateforme pour mener notre recherche. Il est à noter que des « *escape games* », collaboratifs et sans expérience requise en programmation sont en cours de développement.

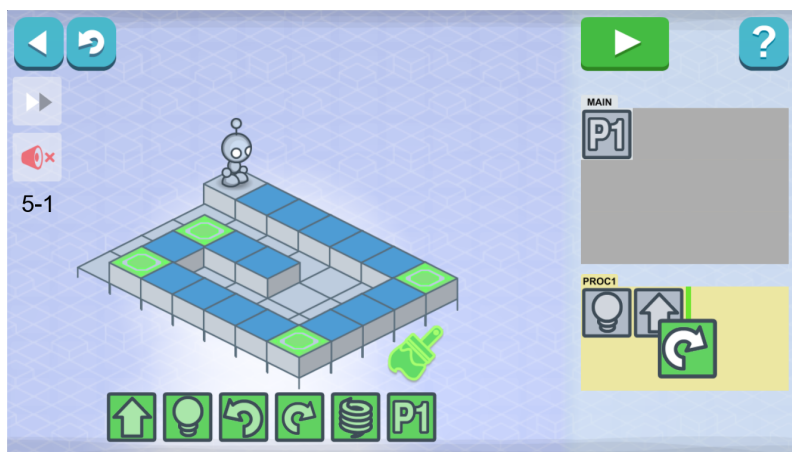
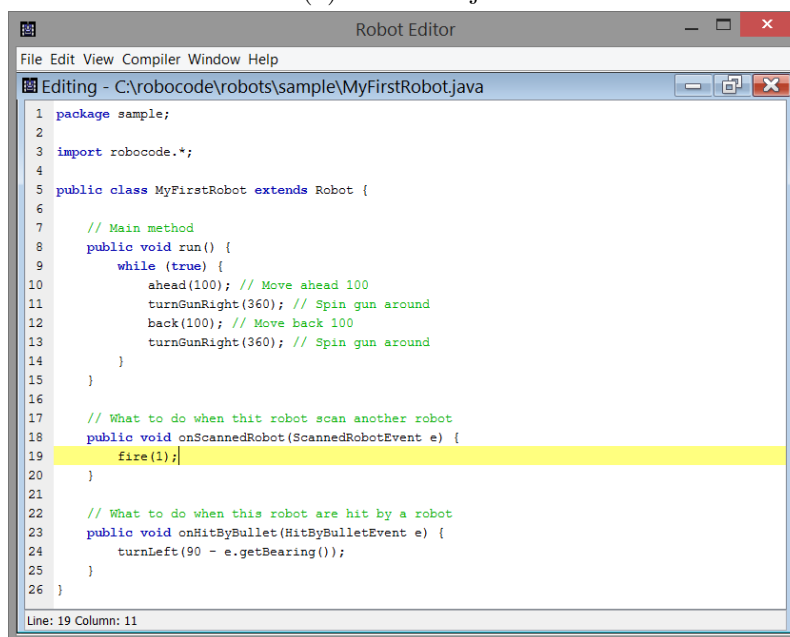


FIGURE 3.3 – Lightbot

Plusieurs autres solutions sont apparues, que nous ne détaillerons pas ici car elles présentent fondamentalement les mêmes types de problèmes que ceux décrits ci-dessus.

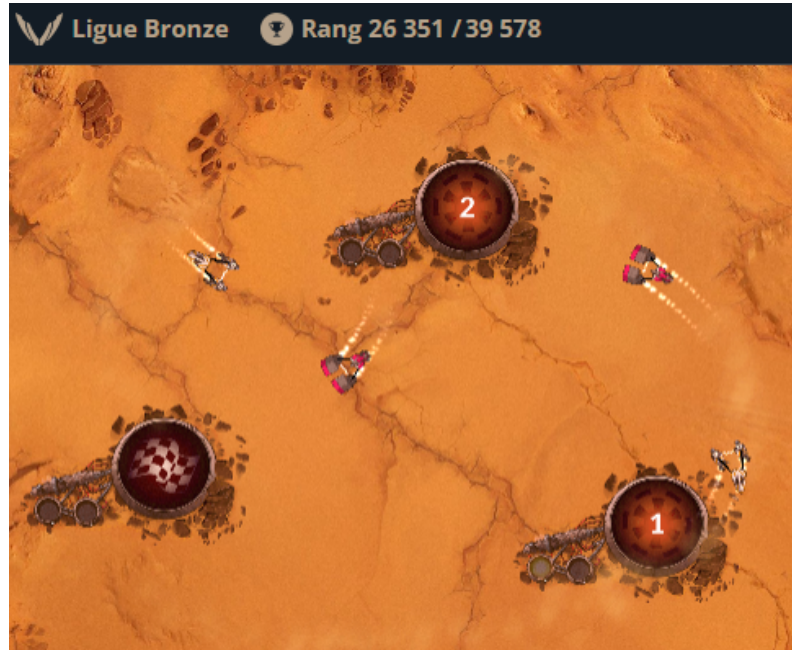


(a) Arène de jeu



(b) Programme en Java

FIGURE 3.4 – Robocode



(a) Course de « pods »

```
Python 3
1 import sys
2 import math
3
4 # Auto-generated code below aims at helping you parse
5 # the standard input according to the problem statement.
6
7
8 # game loop
9 while True:
10     # next_checkpoint_x: x position of the next check point
11     # next_checkpoint_y: y position of the next check point
12     # next_checkpoint_dist: distance to the next checkpoint
13     # next_checkpoint_angle: angle between your pod orientation and the direction
14     x, y, next_checkpoint_x, next_checkpoint_y, next_checkpoint_dist, next_checkpoint_angle = [int(i) for i in input().split()]
15     opponent_x, opponent_y = [int(i) for i in input().split()]
16
17     # Write an action using print
18     # To debug: print("Debug messages...", file=sys.stderr)
19     print("Debug messages. " + str(next_checkpoint_dist), file=sys.stderr)
20     if next_checkpoint_angle > 90 or next_checkpoint_angle < -90:
21         thrust = 0
22     else:
23         thrust = 100
24     if next_checkpoint_angle < 10 or next_checkpoint_angle > -10:
25         if next_checkpoint_dist > 800:
```

(b) Programme de pilotage en Python

FIGURE 3.5 – Codingame

### 3.2.4 Proposition d'une solution innovante

Nous pensons qu'un Serious Game for Education peut être propice à l'apprentissage des rudiments de la programmation et supposons que les caractéristiques fondamentales suivantes constituent des facteurs positifs clés :

- simplicité d'utilisation,
- simplicité extrême des concepts mis en application et progressivité de leur intégration,
- pratique d'une compétition douce entre plusieurs apprenants de la même classe,
- communication possible via le jeu.

Nous n'avons pas trouvé de SGE qui, présentant ces caractéristiques, nous aurait permis de vérifier nos suppositions. Nous avons donc décidé d'en créer un prototype pour tenter d'établir une « démonstration de principe ». Ce premier résultat de recherche n'étant à ce stade pas validé, nous espérons que sa mise en application et son évaluation dans une situation de cours effective y parviendra.

Les règles précises que nous avons élaborées, ainsi que la manière dont nous les avons suivies pour réaliser notre logiciel sont détaillées dans le quatrième chapitre de ce travail.

## 3.3 Déroulement de l'expérience

L'expérimentation destinée à fournir les résultats permettant d'apporter des éléments de réponse à notre question de recherche prend place au sein d'un cours d'introduction à la programmation donné dans la section de niveau secondaire « technicien en programmation ». Les élèves suivant ce cours sont en principe des novices complets en programmation.

L'idée générale est la suivante : mettre les étudiants en situation de relative difficulté par rapport à certains problèmes concrets de la programmation lors des premières séances de cours, pour ensuite leur faire utiliser un jeu sérieux multijoueur censé les aider dans leur apprentissage. Un maximum d'informations relatives aux impacts perçus par les protagonistes de l'expérience est ensuite recueilli. L'analyse et l'interprétation de ces résultats fournit des éléments de réponse à notre question de recherche.

La section « technicien en programmation » se déroule sur deux années de cours donnés à temps plein en journée. Elle se donne au niveau secondaire, il n'y a d'autres prérequis que le certificat de l'enseignement secondaire du deuxième degré (C2D). Dans cette section, le cours d'initiation à la programmation comporte 120 périodes de cours de 50 minutes. Selon le dossier pédagogique, « cette unité d'enseignement vise à permettre à l'étudiant de réaliser un programme traduisant une situation élémentaire

- en représentant sous forme schématique sa structure ;
- en la transposant dans un langage structuré puis dans un langage de programmation. »

Les cours sont donnés par séquences de cours de 4 périodes (une matinée, ou une après-midi). Les 5 premières demi-journées sont classiquement dédiées à une introduction aux premières notions à acquérir dans le cadre de ce cours. Après, pendant 2 demi-journées, les étudiants sont invités à pratiquer le jeu, en se confrontant les



uns aux autres. Ensuite les cours reprennent normalement.

### 3.3.1 Introduction à l'informatique

La première séance de cours est consacrée à une introduction générale à l'informatique dans laquelle est elle-même abordée la programmation. Les notions évoquées vont de l'histoire de l'informatique à la représentation des informations en passant par de la terminologie et un peu d'électronique. Le but est de contextualiser et de démystifier au maximum le domaine.

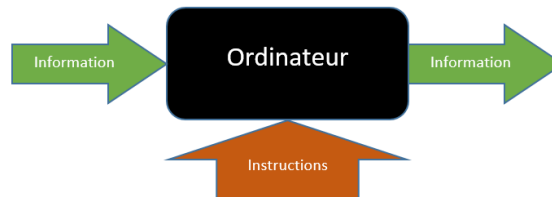


FIGURE 3.6 – L'ordinateur vu comme une simple boîte noire

La programmation informatique est quant à elle présentée de la manière suivante :

- Présentation de l'informatique comme « traitement automatique de l'information », avec une brève réflexion sur les trois termes utilisés.
- Présentation d'un modèle abstrait simplifié d'ordinateur mettant en œuvre les notions abordées plus tôt (figure 3.6).
- Affinage du premier modèle, avec l'introduction des notions de donnée, d'instruction et de séquence d'instructions, amenant à une première représentation naïve d'un programme (figure 3.7).

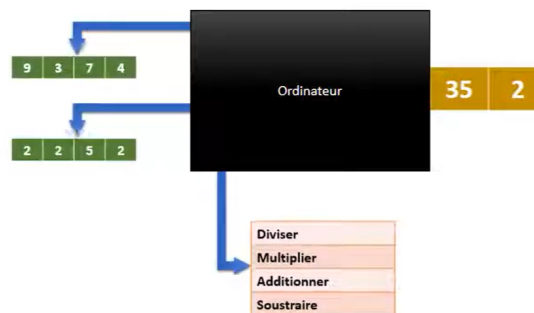


FIGURE 3.7 – Une vue très simplifiée de la programmation

### 3.3.2 Algorithmique de base

Les deux séances suivantes sont consacrées à la notion d'algorithme. Après un rappel de la leçon précédente et une formalisation des termes d'entrées/sorties, d'exécutant, d'instructions et d'algorithme, le cours se base sur l'utilisation de « Flowgorithm »<sup>1</sup>. Ce logiciel destiné au débutant repose sur l'utilisation « d'organigrammes graphiques simples »<sup>3.8a</sup>.

1. <http://www.flowgorithm.org/>

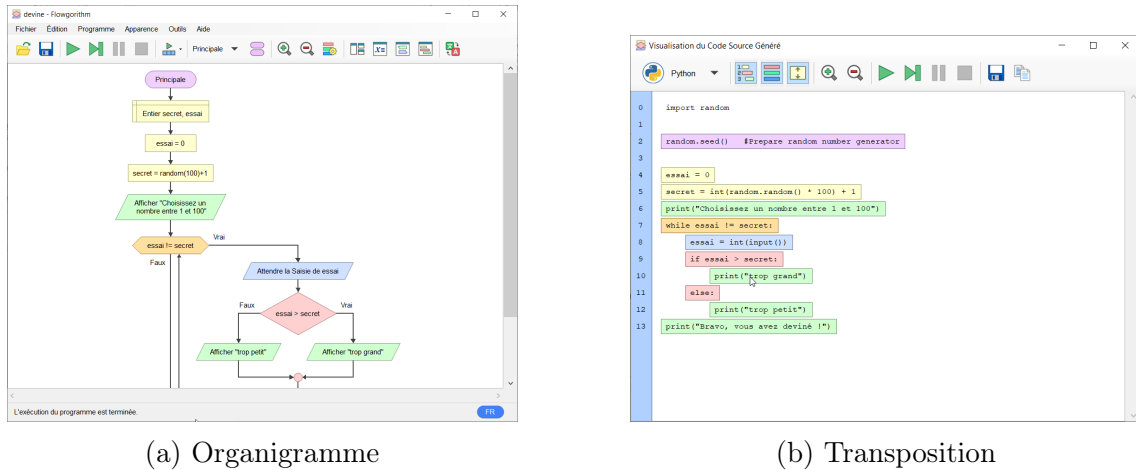


FIGURE 3.8 – Flowgorithm

La mise en œuvre du programme ne nécessite que quelques explications sur le fonctionnement de l'interface, qui est particulièrement ergonomique et traduisible en français.

Après quelques démonstrations de programmes simples par l'enseignant, l'étudiant tente à son tour quelques exercices portant sur des problèmes élémentaires de difficulté croissante. Le dernier de ceux-ci consiste généralement à implémenter le jeu du nombre à deviner sur base d'indications « trop grand, trop petit ». La solution n'est pas donnée car le but est de faire percevoir à l'élève la difficulté à formuler et résoudre un problème de manière formelle (pensée informatique). Cette question n'est que très rarement résolue par les vrais débutants.

### 3.3.3 Premiers programmes rédigés textuellement

Les deux séances suivantes sont consacrées à un premier contact avec le langage Python et les contraintes de programmation attachées. L'objectif ici est de faire percevoir à l'apprenant la charge cognitive induite par un langage textuel (mémorisation, syntaxe, rigueur, environnement de développement). Le lien est tout d'abord établi entre les programmes graphiques conçus avec Flowgorithm (figure 3.8a) et les langages plus formels et rédigés (figure 3.8b).

L'élève est ensuite guidé dans la mise en place de son environnement de développement : interpréteur Python<sup>2</sup> et Visual Studio Code<sup>3</sup>. Sur base de la prise de notes « papier » à partir des transpositions de Flowgorithm, le but est de rédiger soi-même des scripts Python pour résoudre des problèmes similaires à ceux rencontrés précédemment.

A ce stade, les élèves ont une conscience plus nette de l'objet du cours et des obstacles inhérents à la programmation. Les exercices choisis mettent en exergue les difficultés liées aux boucles, structures conditionnelles et tests logiques.

2. <https://www.python.org/>

3. <https://code.visualstudio.com/>

### 3.3.4 Pratique du jeu

Suite au début classique du cours, deux demi-journées de jeu sont alors organisées, où les élèves sont amenés à se confronter les uns aux autres. Le jeu ne fait pas l'objet d'explications préalables, à part pour les modalités d'accès (navigateur, adresse du serveur de jeu et comptes individuels). Le seul rôle de l'enseignant/chercheur consiste à vérifier le bon fonctionnement du dispositif et à expliquer aux participants le contexte où se dérouleront les deux cours à venir.

Pour la première séance, dix niveaux de jeu de difficultés progressives sont implémentés pour introduire de manière progressive les concepts abordés au cours. Les participants ont été libres de créer leurs parties pour attendre leur adversaire et/ou de rejoindre une ou des parties en attente. Tous ont pratiqué au moins une fois chacun des niveaux.

Pour la seconde séance, le mode « création » du jeu est expliqué et liberté est laissée aux étudiants intéressés de concevoir et de tester leurs propres personnages et leurs propres niveaux. Cette séance libre a manqué d'un cadre méthodologique qui aurait permis de récolter des données à analyser. La démarche créative sollicitée a semblé rencontrer un succès qui mériterait probablement une investigation plus poussée.

## 3.4 Collecte et traitement des résultats

La collecte des résultats dans notre cas est composée de deux phases complémentaires : la conception du jeu et son évaluation. Cette dernière se fonde sur l'analyse qualitative de données issues d'entretiens semi-directifs, réalisés au moyen d'un questionnaire élaboré ainsi :

- les questions sont préparées à l'avance, ouvertes et formulées de manière assez générale,
- elles sont organisées selon une structure logique, par thématique. Ainsi, nous commençons par des questions pour établir un profil de l'étudiant. Ces questions portent sur son sexe, son âge, ses motivations à s'inscrire à la formation et son expérience de vie, scolaire et professionnelle. Nous y incluons aussi une question ouverte sur son attitude face aux jeux vidéo. Nous structurons la suite de l'entretien selon trois angles.
  1. **L'andragogie** commence par deux questions complètement ouvertes sur l'opinion et le ressenti général, et est suivie de quatre autres, plus en lien avec des concepts de la théorie andragogique (réponses aux attentes, efficacité perçue, prise en compte de l'expérience et atmosphère de jeu).
  2. **la jouabilité** est explorée de manière assez détaillée. La compétition et la collaboration, points névralgiques, sont évoqués en premier. Trois points de design clés suivent : les labyrinthes conçus, le chat et le système de feedback. Une question relative à la réactivité du jeu est ajoutée.
  3. **La didactique** est abordée au regard de l'attitude (motivation et persévérance), puis selon l'assimilation des concepts intégrés dans le jeu (structures conditionnelles, capacité à combiner celles-ci et transposition en langage textuel).
- la plupart des questions ouvrent vers un enrichissement des réponses et/ou d'autres questions.

Les interviews se déroulent par téléphone, en « voice over IP » avec le logiciel Zoiper<sup>4</sup>, afin de pouvoir enregistrer les conversations. Celles-ci sont ensuite retranscrites mot à mot dans le traitement de texte Microsoft Word.

Les données brutes ainsi récoltées sont nombreuses, et la nature même de ce que l'on va en extraire est à l'origine mal défini. C'est la raison pour laquelle il est utile de les organiser. Pour cela, nous procédons en plusieurs étapes :

1. **Lecture complète** et attentive des transcriptions, pour s'imprégner globalement de la matière collectée,
2. **Codage ouvert descriptif** : il s'agit ici d'étiqueter des fragments de texte avec leur thème sous-jacent, relativement précisément.
3. **Codage axial** : en observant les codes détaillés obtenus à l'étape précédente, nous essayons de les regrouper d'autres codes plus généraux, potentiellement utiles pour notre étude. Les codes détaillés peuvent-être inclus dans plusieurs codes généraux.

	A	B	C	D	E
1	Question				
2	Qu'est-ce qui vous a motivé à vous inscrire dans le cours de programmation informatique ?				
3		Apprendre de nouvelles choses sur l'informatique, avoir de nouveaux horizons, avoir d'autres débouchés vu que la restauration pour moi, ce n'est plus possible on va dire.	Evolution professionnelle	Evolution personnelle	Goût pour l'informatique
4		Et comme c'est un truc qui m'a toujours passionné, je me suis dit pourquoi pas me lancer là-dessus et voir où ça m'amène. Vers des projets qui arrivent petit à petit.	Evolution professionnelle	Attrait pour la programmation	
5	Aviez-vous une expérience antérieure en programmation informatique ?				
6		Non, aucune		Débutant complet	
7	Quelle est votre attitude par rapport aux jeux vidéo en général ?				
8		Très forte. Depuis mes dix ans, je dirais. Je suis ce que l'on pourrait appeler un joueur compulsif.	Beaucoup d'expérience	Passion jeux informatiques	Dépendance aux jeux
9	De manière tout à fait générale avez-vous apprécié le SG proposé lors du cours de programmation informatique ? (oui/non, pourquoi ?)				
10		Oui, c'était pas mal.		Avis positif sur le jeu proposé	
11		Après, au niveau graphisme etc., vu ce qu'on connaît maintenant, ça reste basique, je vais dire ça comme ça	Jeu peu évolué	Jeu basique	
12		mais au niveau ambiance générale du jeu, c'était sympa.		ambiance agréable	
13	De manière tout à fait générale, qu'avez-vous pensé du SG proposé lors du cours de programmation informatique ?				
14		C'est très bien, j'ai bien aimé le partage, enfin la coopération je vais dire ça comme ça. Le fait d'être à deux, je trouve ça vraiment sympa à faire sur le jeu.	Multijoueur positif	Coopération positive	
15	Le jeu correspondait à vos besoins ? (oui/non, pourquoi ?)				
16		Ça nous apprend, oui, ça nous apprend les marches en avant à suivre pour après.	Décomposition d'étapes	Apprentissage ressenti	
17		Ça nous apprend, comment je pourrais dire, les étapes pour pouvoir faire, si je peux dire ça comme ça.		Décomposition d'étapes	
18		Comment établir un programme, on le voit bien avec les blocs.	Les blocs aident à comprendre	Construction de programme	

FIGURE 3.9 – Première tentative de codage ouvert

Concrètement, nous suivons plusieurs itérations pour ces codages, pour affiner nos résultats. Une première tentative consiste à transposer dans un tableau chaque portion de texte et à tenter d'y trouver une description en un ou quelques mots (figure 3.9). Cette méthode, en plus d'être chronophage, s'est rapidement montrée peu efficace. Certaines phrases possèdent plusieurs connotations intéressantes, et il faut pouvoir y associer plusieurs codes, alors que ce n'est pas le cas pour d'autres. Il est difficile de se souvenir si d'autres phrases correspondent à un code donné, et il est donc très difficile d'avoir une vue d'ensemble, et cela même au sein d'un même document. Procéder ainsi sur plusieurs documents et commencer à réfléchir à un traitement axial de ces codes se révèle trop compliqué.

Cette itération nous permet cependant d'affiner nos codes et notre vue d'ensemble des données. Nous décidons d'utiliser un autre outil, dédié à l'analyse qualitative des données : Atlasti<sup>5</sup>.

4. <https://www.zoiper.com/>

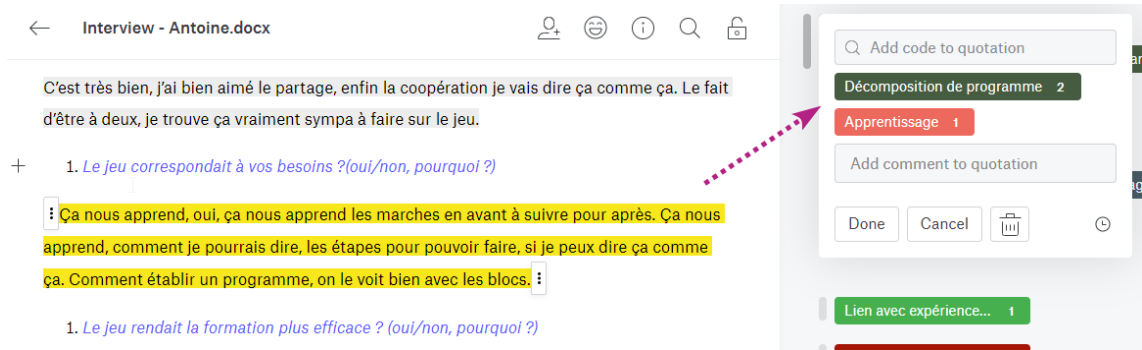
5. <https://atlasti.com>

Ce logiciel beaucoup plus souple et intuitif peut être utilisé de plusieurs manières, voici comment nous avons procédé.

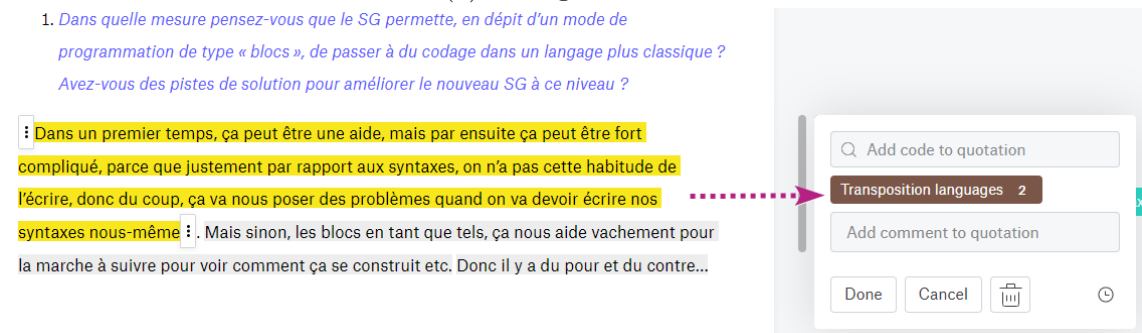
1. Nous avons importé les transcriptions de nos sept entretiens en leur donnant le nom de l'interlocuteur concerné,
2. Les contenus ayant déjà été parcourus en détail, chaque fragment de discours significatif a été annoté d'un ou plusieurs codes descriptifs. Un même fragment a pu être annoté de diverses manières et une portion d'un fragment se voir elle-même étiquetée.
3. La relecture de tous les codes générés a permis la fusion de certains d'entre eux et l'affinage des termes utilisés,
4. L'état de l'art a été relu en détail ainsi que la question de recherche et ses hypothèses, pour aider à contextualiser l'analyse axiale,
5. Les divers codes ont été regroupés sous la forme de thèmes jugés pertinents.

Par exemple, comme décrit dans la figure 3.10, une portion de texte a été associée à la notion de « décomposition de programme », et une autre, chez un autre interlocuteur, à celle de « transposition entre langages ». Ces deux notions, comme nous l'avons vu, sont assimilées à la Pensée Informatique, ce qui est décrit à la figure 3.10c. Ensuite, une analyse est réalisée en filtrant les citations selon ce code axial, ce qui permet par exemple de faire émerger le code « d'efficacité pédagogique » pour lequel nous pouvons filtrer les extraits de discussions. On pourrait ainsi lire que l'efficacité didactique pour comprendre la décomposition semble associée à la programmation par bloc, mais aussi au niveau débutant de l'apprenant.

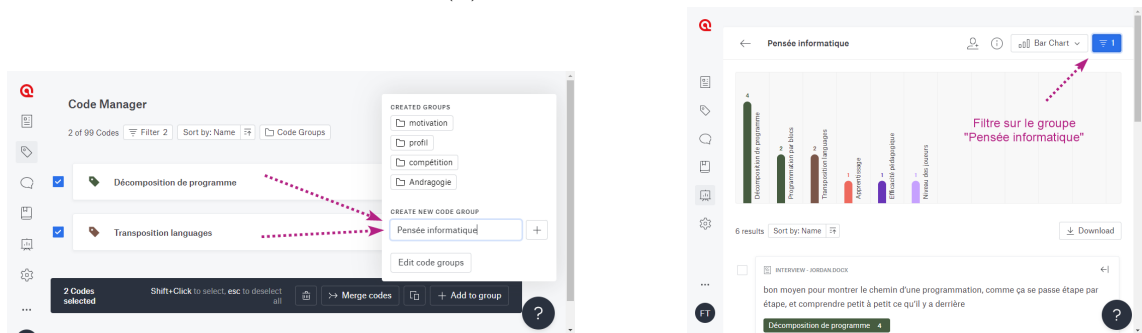
En réalité le cycle décrit ci-dessus a été effectué de nombreuses fois pour tenter d'objectiver et de structurer au mieux les informations extraites.



(a) Codage ouvert A



(b) Codage ouvert B



(c) Codage axial

(d) Analyse

FIGURE 3.10 – Codage d'entretien avec Atlasti

# Chapitre 4

## Résultats

### 4.1 Description du jeu

L'idée originale de ce jeu a dès le début été de confronter plusieurs adversaires dans des labyrinthes, au travers d'avatars pilotés par des procédures préétablies. Chaque protagoniste est ainsi invité à rédiger à l'avance un programme qui fera évoluer son personnage en fonction de son environnement, si possible mieux que ses adversaires.

Cette approche est destinée à illustrer la notion de programmation informatique en partant d'entrées/senseurs qu'un automate traite en suivant une procédure établie et en agissant conséquemment sur des sorties/actionneurs. On peut également voir que même si plusieurs programmes mènent au « bon » résultat, les performances de ceux-ci peuvent différer, introduisant le concept d'efficacité d'un algorithme.

On espère que l'interaction entre joueurs, autant en termes de compétition que de collaboration, sera stimulante et propice à une certaine forme de socioconstructivisme.

L'application a été conçue en respectant un certain nombre de contraintes, certaines d'ordre purement technique et d'autres plutôt conceptuelles, dictées par notre exploration de l'état de l'art et notre expérience du domaine.

#### 4.1.1 Exigences techniques

- le coût en temps et en argent nécessaire à la création d'un SG « grandeur nature » sortent clairement du cadre de ce travail et notre prototype ne devrait pas être trop ambitieux, tout en essayant d'être évolutif.
- s'agissant d'un jeu multijoueur devant être testé aisément, son déploiement sur le réseau et sur les terminaux des concurrents devrait être simple.
- La RDE s'appuie sur l'affinage cyclique d'un prototype. C'est pourquoi celui-ci devrait être modulable pour éventuellement permettre l'ajout d'éléments propres au jeu où à la recherche qu'il nourrit.

#### 4.1.2 Exigences conceptuelles

- Notre prototype est avant tout un Serious Game for Education, il doit donc, tout en restant ludique, être efficace pédagogiquement et favoriser motivation et persévérance.

- Le jeu doit s'intégrer dans le processus éducatif normal du cours. Le lien doit être établi entre ce qui précède et ce qui suit sa pratique ; la difficulté correspondre aux exigences du moment où il est utilisé.
- Un de nos postulats stipule que la compétition et la collaboration peuvent se combiner pour favoriser l'échange des connaissances. Une attention particulière doit être portée à l'équilibrage des deux dimensions et des dispositifs implémentés pour favoriser l'échange concret d'informations.
- Comme vu dans l'état de l'art, les apports réels des jeux sérieux éducatifs restent contestés. Il est donc important de suivre les prescriptions semblant, selon la littérature, favoriser ces bénéfiques. Retenons particulièrement :
  - la possibilité d'apprendre par essais/erreurs,
  - la présence de feedbacks informatifs et correctifs,
  - la prise en compte de différents rythmes d'apprentissage,
  - le fait d'avoir à expliquer ses réponses.
- Concrètement, le jeu doit pouvoir se focaliser sur un nombre restreint de concepts et faciliter leur intégration.
- La charge cognitive pesant sur le débutant doit être minimisée tant que possible pour permettre au joueur de se focaliser sur cette activité d'assimilation.

### 4.1.3 Implémentation

#### Fonctionnement général

Le jeu est pensé comme un jeu de plateaux sur lesquels s'opposent plusieurs joueurs à partir de leurs postes de travail. Chacun de ces plateaux est constitué d'un labyrinthe où l'on trouve des pièces à récolter. En début de partie, chaque protagoniste est représenté par un avatar positionné à un endroit précis du labyrinthe. Chaque concurrent dispose aussi d'une « boîte à outils » où des concepts fondamentaux de la programmation sont matérialisés par des blocs visuels à combiner comme des pièces de puzzle sur une zone dédiée. Un chat textuel est aussi disponible, permettant au joueur de converser en cours de partie. Une fois les programmes créés et validés par chacun des participants, les personnages les exécutent simultanément, bloc après bloc. Pour fournir un retour à l'utilisateur et aider au débogage, ces blocs sont mis en surbrillance au fil de leur exécution.

Pour la conception nous avons décidé de déployer notre logiciel sur un réseau local et une architecture client/serveur mettant en œuvre les protocoles et langages du Web. Outre la facilité offerte par ce choix, en centralisant l'échange des données de jeux (comme les coups joués ou les échanges écrits entre joueurs), il serait possible d'enregistrer des données utiles pour la recherche. Cette fonctionnalité n'a toutefois pas été implémentée à ce jour.

Le développement s'est appuyé sur :

- trois langages : Python, HTML et Javascript,
- l'environnement de développement Visual Studio Code,
- le système de contrôle de version GIT (notre projet est hébergé sur Github<sup>1</sup>).

---

1. [https://github.com/Lapin-Blanc/maze\\_project](https://github.com/Lapin-Blanc/maze_project)



## Fonctionnalités annexes

Le jeu dispose de deux éditeurs, l'un pour le design des plateaux et l'autre pour les avatars. L'enseignant peut ainsi concevoir des niveaux adaptés à des besoins spécifiques et les élèves des plateaux destinés à des parties sur mesure, changeant ainsi le paradigme de jeu originel.

Le prototype utilisé dans le cadre de ce travail possède, à des fins de recherche, dix niveaux de difficulté croissante identique pour chaque concurrent (les labyrinthes sont symétriques). Ces niveaux sont aussi conçus pour introduire progressivement les concepts évoqués plus haut : boucles et structures conditionnelles.

## Le serveur d'application

Notre application repose principalement sur le framework open-source Django<sup>2</sup>. Celui-ci est développé en Python, ce qui permet son utilisation sur la plupart des systèmes d'exploitation. Il permet le développement d'applications Web complètes déployables selon les spécifications WSGI ou ASGI. La norme WSGI décrit une interface entre les applications développées en Python et les serveurs Web. Sa successeurice ASGI permet le déploiement d'application asynchrones.

Django met en œuvre les paradigme MVT (Model View Template) et ORM (object-relational mapping) comme décrit par la figure 4.1.

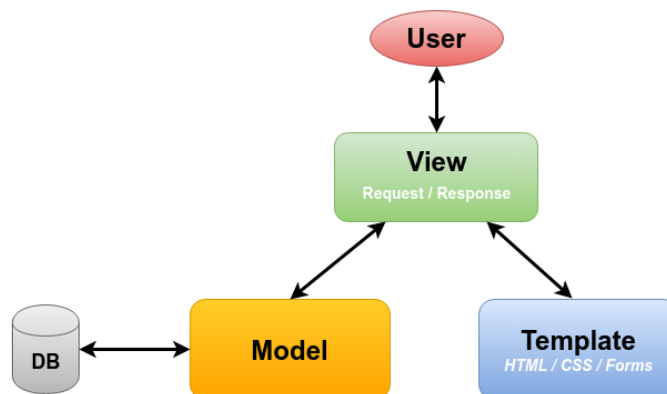


FIGURE 4.1 – Django MVT et ORM

Une entité globale (le projet) est créée puis dotée d'un certain nombre de modules (les apps). Dans notre cas, le projet principal *maze\_project* comprend les applications suivantes *accounts*, *maze*, *chat* et *game*.

**accounts** est le module dédié à la gestion centralisée des utilisateurs/joueurs. Ceux-ci doivent d'abord être encodés par l'administrateur du site pour pouvoir ensuite se connecter et jouer. L'interface d'administration est un module fournit d'origine par Django (figure 4.2) qui s'occupe de stocker et de protéger les informations de connexions dans la base de données du projet.

2. <https://www.djangoproject.com/>



FIGURE 4.2 – Gestion des joueurs

**maze** est destiné à la gestion des labyrinthes (blocs, niveaux et personnages). Le module fournit une interface conviviale pour créer, éditer et tester les personnages du jeu à l'aide de « sprites » (figures 4.3 et 4.4).



FIGURE 4.3 – Un « sprite » pour le personnage Astro

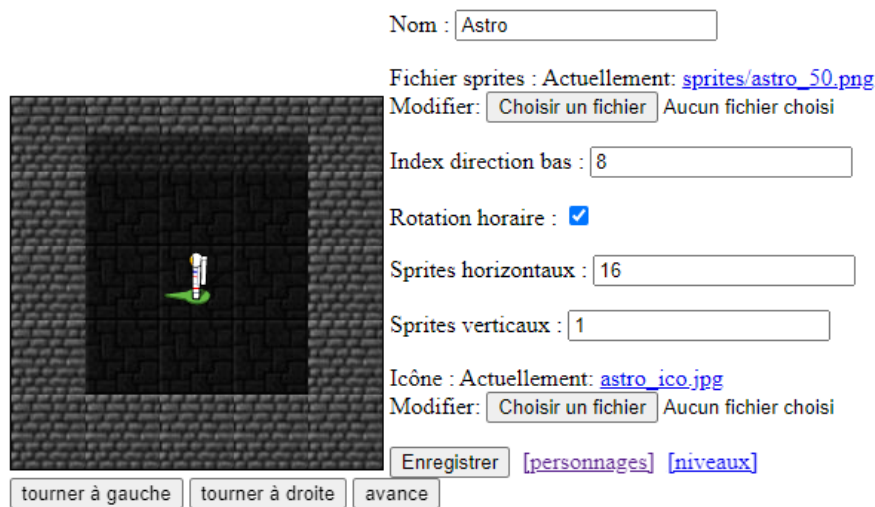


FIGURE 4.4 – Création et test d'un personnage

L'éditeur de niveau fonctionne sur le même principe et permet de choisir différentes tailles de labyrinthe de placer des murs, des pièces et les personnages dans ceux-ci.

**chat** implémente les fonctionnalités d'échange d'informations durant le jeu, aussi bien pour le chat que pour les communications systèmes (échange des programmes, listing en temps réels des salles disponibles, etc.). D'un point de vue technique, le module met en œuvre le serveur ASGI Daphne<sup>3</sup> pour gérer les communications asynchrones initiées côté client par le protocole WebSocket.

3. <https://pypi.org/project/daphne/>

```

HTTP GET /admin/maze/pixel/ 200 [0.06, 127.0.0.1:56650]
HTTP GET /admin/jsi18n/ 200 [0.01, 127.0.0.1:56650]
WebSocket DISCONNECT /ws/chat/ [127.0.0.1:56308]
WebSocket HANDSHAKING /ws/chat/15/ [127.0.0.1:56726]
WebSocket CONNECT /ws/chat/15/ [127.0.0.1:56726]
CHANNEL LAYER GROUPS : {'15': {'specific..inmemory!VabliZxqdvgl': 1653388206.089882}}
CHANNEL LAYER NAME : specific..inmemory!VabliZxqdvgl
POSITION : 0
PLAYERS : [['0', 'fabien'], ['0', 'corine']]
WebSocket DISCONNECT /ws/chat/15/ [127.0.0.1:56726]
    
```

FIGURE 4.5 – Log serveur Daphne

**game** La partie frontale est une application monopage mobilisant plusieurs technologies et bibliothèques côté client. L’implémentation du mécanisme de jeu est décrite à la figure 4.6. Une fois la page rendue par le moteur de gabarit Django, elle est envoyée au navigateur qui procède au chargement des bibliothèques Javascript tierce-partie nécessaires, finalement nos propres scripts implémentent la logique de jeu, la mécanique, l’esthétique, etc.

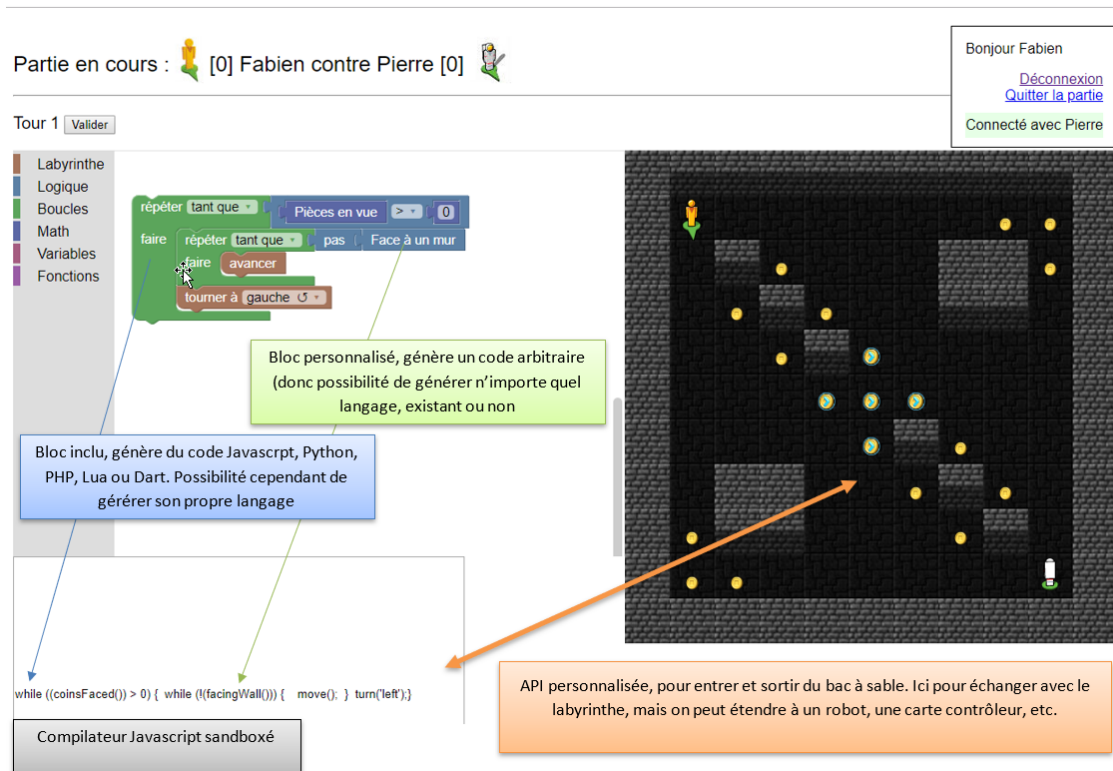


FIGURE 4.6 – Principe d’implémentation du frontend

- **Blockly**<sup>4</sup>, proposé par Google sert de fondation à notre système de programmation par blocs. La bibliothèque fournit de base un ensemble de briques correspondant aux structures que l’on peut retrouver dans la majorité des langages de programmation (figure 4.7). Elle permet de combiner ces blocs pour former des programmes « visuels » qui sont traductibles dans plusieurs langages parmi lesquels nous choisissons Javascript.

L’extensibilité est le point qui nous a fait choisir cette bibliothèque. Il est en effet possible de créer aisément des blocs personnalisés et de leur faire générer du code arbitraire. Ainsi, le bloc *moveForward* que nous avons créé

4. <https://developers.google.com/blockly>

4.8 va produire le code Javascript `move()`. C'est notamment via l'API fournie par cette librairie que nous pouvons mettre en surbrillance les blocs en cours d'exécution.

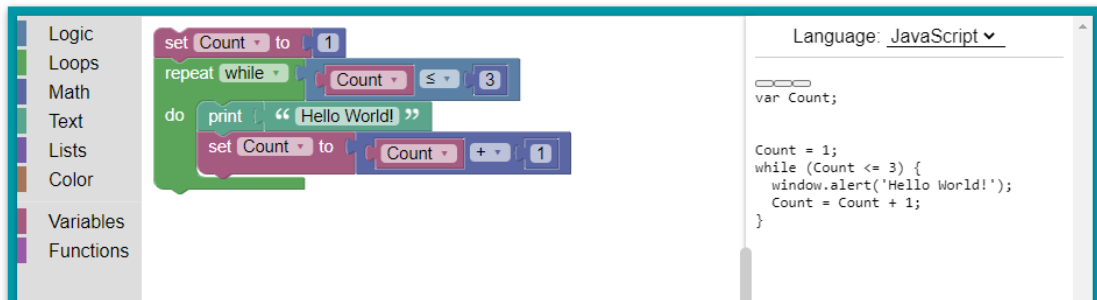


FIGURE 4.7 – Blockly

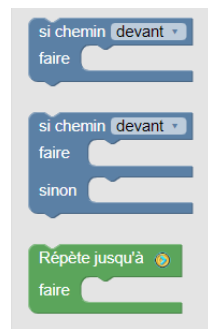


FIGURE 4.8 – Blocs personnalisées

- **JS-Interpreter**<sup>5</sup> : comme expliqué par son auteur, « JS-Interpreter est un interpréteur JavaScript sandboxé écrit en JavaScript. Il permet l'exécution de code arbitraire ligne par ligne. L'exécution est complètement isolée de l'environnement principal. Les instances multiples de l'interpréteur JS-Interpreter permettent l'exécution simultanée de JavaScript en mode multithread sans l'utilisation de Web Workers. »

### JS-Interpreter Demo

Enter JavaScript code below, then click `Parse`. To execute, either click `Step` repeatedly, or click `Run` once. Open your browser's console for errors.

```
var result = [];
function fibonacci(n, output) {
  var a = 1, b = 1, sum;
  for (var i = 0; i < n; i++) {
    output.push(a);
    sum = a + b;
    a = b;
    b = sum;
  }
}
fibonacci(16, result);
alert(result.join(', '));
```

Parse Step Run

FIGURE 4.9 – JS-Interpreter

5. <https://neil.fraser.name/software/JS-Interpreter/docs.html>

- **P5.js**<sup>6</sup> : il s'agit d'une bibliothèque graphique Javascript qui permet de créer des espaces de dessin interactifs 2D dans des pages Web, nommés canevas.

Home

# Flocking

Editor < Back to Examples

Download Demonstration of Craig Reynolds' "Flocking" behavior. See:  
http://www.red3d.com/cwr/ Rules: Cohesion, Separation, Alignment (from  
natureofcode.com). Drag mouse to add boids into the system.

Donate

Get Started

Reference

Libraries

Learn

Teach

Examples

Books

Community

Showcase

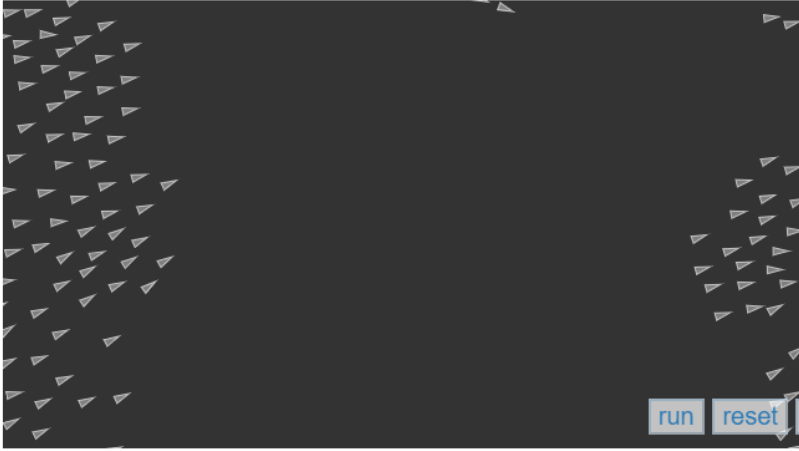
Forum

GitHub

Twitter

Instagram

Discord



```
let flock;  
  
function setup() {  
  createCanvas(640, 360);  
  createP("Drag the mouse to generate new boids.");  
  
  flock = new Flock();  
  // Add an initial set of boids into the system  
  for (let i = 0; i < 100; i++) {  
    let b = new Boid(width / 2,height / 2);  
  }  
}
```

FIGURE 4.10 – exemple p5\*js

---

6. <https://p5js.org/>

## 4.2 Données recueillies

Il s'agit dans cette section de décrire les données recueillies au travers de nos entretiens ainsi que l'analyse qualitative qui a suivi. Une fois ces résultats synthétisés et organisés, nous pourrions les interpréter et les articuler au regard de notre état de l'art, de notre question de recherche et de nos hypothèses.

Tout d'abord, sur les sept personnes interrogées :

- deux sont des femmes et cinq sont des hommes,
- les âges s'étendent de 22 à 60 ans, avec : 22(H), 23(H), 24(H), 26(H), 38(F), 58(F) et 60(H).
- professionnellement parlant, cinq sont sans emploi et deux sont employées à temps partiel et/ou en soirée.
- au niveau de la scolarité, deux ont un certificat de qualification, trois un certificat de l'enseignement secondaire supérieur, une est de niveau bachelier et une dernière a réussi une candidature en ingénieur civil.
- bien que la formation soit censément destinée aux débutants complets, deux des étudiants ont une expérience ancienne de la programmation (plus de vingt ans). Ces dernières déclarent cependant avoir « tout oublié » ou encore considèrent que leurs compétences sont désuètes.

Les *motivations pour s'inscrire* à la formation les plus souvent mentionnées sont les perspectives d'évolutions professionnelles suivies des personnelles. Viennent ensuite l'attrait pour la programmation et l'envie de progresser en informatique. Certains motifs plus précis sont aussi évoqués, comme l'envie de pratiquer le « hacking », l'envie de faire de la « programmation robotique » ou encore l'attrait pour le développement de jeux vidéo.

Les *attitudes envers ces jeux vidéo* sont particulières : les quatre profils les plus jeunes, tous dans la vingtaine, sont masculins et témoignent tous d'un attrait conséquent pour les jeux vidéo, avec même des connotations d'addiction. Cette passion est couplée à une expérience conséquente dans ce domaine. L'étudiante de 38 ans quant à elle décrit un attrait pour les « jeux Facebook », sans plus. Celle de 58 ans mentionne aussi attrait, mais ne pratique plus pour des raisons d'équilibre personnel. Le plus âgé des élèves, à 60 ans n'est pas intéressé par les jeux vidéo, pour lesquels il manifeste même une certaine aversion.

Lorsqu'il s'agit d'*apprécier le jeu éducatif dans sa globalité* en début d'entretien, l'ensemble des sondés décrivent une expérience positive, certains mettant en avant le côté sympathique de jouer à plusieurs ainsi que la dimension « réflexion » du jeu. L'ensemble des questionnés considère que le jeu est adapté à des débutants complets, et qu'il permet de se faire une idée générale de ce qu'est la programmation. L'un d'entre eux considère qu'il peut continuer à être utile à un « niveau intermédiaire ».

Lorsqu'il est question de *complémentarité avec les cours classiques*, les choses sont un peu plus nuancées. Plusieurs personnes pointent un réel apport, par exemple : « ça donne un équilibre, ça fait avancer », « avec le jeu ça se combine assez bien. » ou « on est incités nous même à produire ces actions au lieu de regarder des vidéos, d'écouter et d'essayer de reproduire ». D'autres regrettent l'absence d'interaction directe avec un enseignant : « Dans le cours on a une explication que le jeu ne donne pas », « J'ai vraiment besoin d'un professeur » ou encore « des explications orales en plus, [pour] tester le niveau dix ». L'une d'entre elle est même catégorique, « Je préfère un cours classique ». Finalement, deux des élèves suggèrent d'établir une

correspondance plus claire entre les phases de jeu et « les chapitres » du cours.

### 4.2.1 Compétition, collaboration et savoirs échangés

Concernant la *perception de la compétition*, elle est globalement positive : « plaisant », « sympa », « positif », « chouette » et même « vraiment fantastique » sont les termes utilisés. L'intensité de la compétition est jugée comme relativement faible. Une personne d'ailleurs ne la perçoit pas vraiment, d'autant que : « parfois on reçoit les solutions de l'adversaire ». Il est aussi intéressant de constater qu'une étudiante perd sans cesse sans en être dérangée, jugeant qu'elle apprend quand même et persévère dans ses efforts. Lorsqu'il est question des *apports en motivation* liés à la compétition, ils sont unanimement reconnus. Certains suggèrent que cela pourrait être lié à la camaraderie régnant entre les joueurs, ceux-ci se connaissant et étant dans la même classe. En cas de confrontation avec des joueurs « extérieurs », il y aurait même une crainte « d'être jugée ». Pour finir, deux élèves suggèrent des améliorations pour renforcer l'esprit de compétition dans le jeu (comptage du nombre d'essais sur plusieurs parties ou encore changement de la nature du jeu pour en faire un jeu de combat de chars d'assaut).

Lorsque l'on observe la facette collaborative du jeu, il est intéressant de constater que personne ne signale d'*incompatibilité entre la compétition et la collaboration*. L'esprit général de cette dimension est bien décrit dans cet extrait : « La compétition et la collaboration ne s'excluent pas. C'est comme une compétition en équipe parce qu'on veut atteindre tous le même objectif ». De nouveau, le faible degré de compétition est pointé. Pour beaucoup, la notion de collaboration est associée à celle d'*entraide* et une seule personne déclare ne l'avoir pas exploitée.

Les références à la nature et à la richesse des échanges dans le cadre de cette entraide sont quant à elles peu nombreuses et assez floues : « Fais comme ça, encourage-le... », « discuter [...] de la méthode qu'il a utilisée pour résoudre », « Si il donne la solution, ça va... ». Une élève soulève un soucis pratique relatif au chat et aux échanges textuels à ce niveau.

### 4.2.2 Caractéristiques du jeu

La *conception générale du jeu* est appréciée pour sa simplicité, autant au niveau de la présentation que de la prise en main. Les objectifs sont clairement perçus, la manière d'utiliser l'interface aussi. le fait de pouvoir commencer sans explications préalables est apprécié. Le fait de ne pas avoir de comptage de points ou de décompte de temps est mentionné comme facteur diminuant le stress.

Le *principe du jeu* qui met en jeu le concept sérieux d'algorithme est clairement perçu et nettement apprécié :

- « c'est excellent, au moins, on voit le déroulement du programme »
- « le fait du labyrinthe, ça colle très bien avec le délire du fait de se déplacer »
- « On comprend qu'il faut sortir en mettant une combinaison de blocs, et que quand c'est bon, ça marche »
- « pratique pour voir où on se trompe dans le labyrinthe »

Dans le même ordre d'idée, le feedback implémenté par la surbrillance des blocs est jugé utile par plusieurs. Exemple : « Je trouve que c'est pratique parce que ça permet de comprendre le fait que les commandes, même dans une boucle, ça recommence.

Ça permet de voir quelle commande s'exécute quand. Ça peut être pratique pour voir où on se trompe dans le labyrinthe. » Quelques réserves sont à nouveau émises concernant le dernier niveau, avec des difficultés à « tester les programmes », ou la nécessité de prendre « une feuille de papier en faisant le chemin, en faisant là je tourne à gauche, là je tourne à droite et ça n'allait pas. »

La *conception des labyrinthes* relève d'une progression perçue par tous les participants, chacun trouvant sa limite à tel ou tel niveau. Cette progression est appréciée, jugée globalement « bien faite », surtout avec le fait d'introduire de nouveaux blocs progressivement. En revanche, tout le monde s'accorde pour dire que le dernier plateau présente une complexité disproportionnée. Certains le voient d'ailleurs comme un « niveau bonus ». Plusieurs remarques suggèrent un intérêt pour le développement de plus de niveaux, de diverses complexités.

Au niveau de la *conception graphique*, une certaine esthétique est ressentie même si le côté « basique » du prototype est plusieurs fois mentionné : « en tout cas visuellement ça n'a rien de désagréable du tout ». Le dépouillement de l'interface de jeu est perçu comme ergonomique et propice aux débutants. Les simulations assez lentes sont appréciées par certains alors que d'autres les déplorent. Finalement, la jouabilité globale est bonne, avec même un certain « fun ».

Alors que la compétition repose sur le scénario de jeu, la collaboration dépend de la communication permise par *le chat*. Celui-ci est relativement bien perçu, surtout au niveau du principe et associé avec la notion d'entraide, de support. Par contre son implémentation fait l'objet de plusieurs remarques : le mode textuel est jugé trop simple, contraignant et peu pratique. Plusieurs pistes d'amélioration sont proposées, comme de l'agrémenter d'émoticônes, de couleurs, etc. Une adjonction ou même un remplacement par l'audio sont plusieurs fois mentionnés. La suggestion de pouvoir échanger graphiquement au sujet du jeu est faite plusieurs fois, notamment pour faciliter une aide plus concrète en cours de partie.

### 4.2.3 Appréciation des apprentissages techniques

Les *concepts clés* de boucles et de structures conditionnelles sont au cœur de la pratique du jeu. C'est leur maîtrise que nous espérons voire renforcée via celle-ci. Les joueurs ressentent généralement une aide à ce niveau, l'exprimant de manière plus ou moins précise : « ça m'a vachement aidé », « les si, sinon [...] c'est des outils qui nous servent beaucoup en programmation », « ça m'a aidé pour [...] les imbrications de si alors, sinon ». Certains soulignent la nécessité qu'ils ont perçue de devoir comprendre ces concepts, simples et peu nombreux, avant d'aller plus loin. Un des élèves estime cependant qu'il faut une explication préalable au principe d'une boucle avant de pouvoir en faire usage dans le jeu.

Alors que le côté visuel de la *programmation par bloc* soit perçu comme très utile pour comprendre et *décomposer la structure* d'un programme, la procédure inverse de *combiner les blocs* pour résoudre un problème est nettement moins évidente. Alors qu'un des questionnés relève que l'introduction progressive d'un concept à la fois est un indice sur la manière de procéder, plusieurs autres suggèrent l'utilisation de structures préparées, par exemple : « On montrerait comment les blocs se mettent ensemble, pour faire des trucs habituels. » ou encore « pour montrer : voilà, si vous imbriquez vos blocs comme ça, ça va fonctionner comme ça ».

Il est à noter, concernant les caractéristiques développées ci-dessus, que la pro-



grammation visuelle par blocs est largement mentionnée, de manière implicite ou explicite. Y est d'ailleurs associée une simplification due à l'absence de difficulté syntaxique ou linguistique. Deux suggestions concernent l'amélioration de ceux-ci via l'adjonction d'information ou d'aide complémentaire, « Genre une petite marque « détail » ».

Dans la mesure où la pratique du jeu doit s'intégrer dans un cours classique, il est crucial de voir dans quelle mesure les élèves pensent qu'il pourront *transposer dans un langage classique* les notions abordées durant le jeu. Là, même si sont relevées deux analogies, l'une avec Excel, l'autre avec Wordpress, beaucoup d'autres remarques expriment des préoccupations :

- « ça aide à comprendre, mais après, ça ne montre pas ce que veulent dire certains langages »
- « Pour moi, les deux sont fort séparés »
- « je ne vois pas comment l'un peut aider l'autre, justement, c'est fort différent »
- « la personne a peut-être besoin d'une explication en plus pour pouvoir faire le lien avec Python ou avec un autre langage »,
- « Dans un premier temps, ça peut être une aide, mais par ensuite ça peut être fort compliqué, parce que justement par rapport aux syntaxes, on n'a pas cette habitude de l'écrire »
- « on doit plus se concentrer sur l'utilisation d'un outil [interface de développement], et ça, on ne le retrouve pas vraiment dans le jeu. »

## 4.3 Discussion des résultats

Nous comptons interpréter nos résultats en commençant par discuter des personnes : qui étaient nos intervenants, quelles étaient leurs motivations, leurs caractéristiques et en quoi correspondaient-ils à ce que l'on pouvait attendre au regard de notre état de l'art ? Nous continuerons en analysant les interactions humaines, leurs natures et leurs apports. Suivra une réflexion sur la manière dont a été perçue notre prototype. Finalement, nous établiront un bilan pédagogique en lien avec nos raisonnements précédents.

### 4.3.1 Le public

Sans surprise, les personnes participant à notre étude forment un public adulte, hétérogène en termes d'âge, de scolarité et d'expérience professionnelle, conformément à ce que l'on peut attendre de la promotion sociale. Les deux sexes sont représentés, même si femmes le sont moins, ce qui correspond aussi à une tendance générale, qui semble cependant s'inverser au cours des années.

Les raisons qui poussent les élèves à s'inscrire à la formation sont variées et conformes à ce que nous dit notre état de l'art (besoins personnels, possibilités offertes, influences extérieures). Il est cela dit intéressant de constater que les novices partagent une certaine méconnaissance de la réalité du domaine et réfléchissent plutôt en termes d'image. L'exemple le plus représentatif est la personne attirée par le « hacking », grâce ou à cause de l'image du « pirate » véhiculée par les médias.

Knowles pensait que la diversité inhérente aux cours pour adultes était propice à des méthodes expérimentales et à des activités d'échange. Cela semble avoir été

le cas, les candidats décrivant généralement une expérience positive, reposant justement sur une dimension sociale. Les échanges provoqués via le jeu dépassent le cadre de celui-ci et créent une dynamique qu'il aurait été plus difficile d'atteindre autrement.

L'expérience menée mobilise une forme de socioconstructivisme, en ce sens que l'apprenant interagit avec son environnement, symbolisé par le jeu, et ses pairs, les autres élèves. L'interaction est ici composée d'un mélange de compétition et de collaboration, la question est de voir si cette association est viable et profitable. Il semblerait que cela soit le cas : l'association des deux ne pose pas de soucis et le concept de compétition paraît attractif et source de motivation et de persévérance. Nous pensons d'ailleurs, en accord avec certains des intervenants, qu'il serait intéressant de renforcer un peu ce sentiment de compétition, pourvu que l'on reste dans le cadre de la classe, sans « ouvrir » les parties au monde extérieur. La collaboration et la communication jouent leur rôle en permettant d'établir un climat sécurisant, en favorisant l'aide en cours de jeu. Le principe d'essais/erreurs est bien vécu car il permet de se faire aider par le meilleur joueur, qui apprend lui-même en devant guider son opposant. Il serait d'ailleurs intéressant de « valoriser » cette aide d'une manière ou d'une autre dans le jeu.

Pour finir, Il semble exister, dans notre expérience une corrélation entre l'attitude des étudiants face aux vidéos et la perception de notre outil. Ainsi plus l'élève est passionné par les jeux vidéo, plus il a trouvé l'approche défendue ici productive. Évidemment, cette corrélation n'est absolument pas représentative en raison de la taille de notre échantillon et ne décrit pas forcément un lien de cause à effet. Il semblait cependant intéressant de la signaler, car elle pourrait renforcer l'hypothèse de Knowles qui décrit le lien primordial entre l'expérience de vie personnelle et l'apprentissage chez l'adulte et ce même lorsqu'il est question de jeux.

### 4.3.2 Concernant la conception du jeu

Les qualités ludiques de notre jeu vidéo devaient être grandes pour soutenir son côté sérieux. Or, l'état de l'art souligne à quel point la perception d'un SG dépend de facteurs différents : jouabilité, modélisation dynamique, mécanique, flow ou encore engagement. N'étant pas concepteur de jeu vidéo et ne disposant que de peu de ressources, nous avons conçu un programme implémentant certes les fonctionnalités nécessaires à notre étude, mais de manière minimale et assez dépouillée.

Compte tenu de ces considérations, nous avons été heureusement surpris par les retours. C'est le principe de jeu qui a été principalement retenu, avec en son cœur le lien entre les personnages, les labyrinthes, les blocs et les feedbacks. Là où nous pensions que l'interface de jeu, sans fioritures et ne comportant que les commandes nécessaires serait pointée du doigt, elle a été perçue comme ergonomique et facilitant la concentration. D'autres caractéristiques, comme les déplacements assez lents ont aussi été du goût des joueurs débutants.

La conception des labyrinthes convenait bien, même si les remarques nous amènent à certaines réflexions : le dixième et dernier niveau a été pointé comme particulièrement difficile, voire insoluble pour beaucoup. Or, quasiment tout le monde est parvenu au neuvième. Nous en tirons deux conclusions : il faudrait plusieurs plateaux intermédiaires entre ces deux étapes, mais plus important, le jeu pourrait montrer là ses limites, deux personnes ayant mentionné la nécessité de « sortir » du

jeu, par exemple avec une feuille et un crayon pour continuer à réfléchir.

Finalement, la partie communication mériterait un remaniement en profondeur, voire une refonte complète. Ce que nous retenons de la situation actuelle, c'est que les possibilités offertes la communication sont attractives, mais que les modalités de fonctionnement actuelles rendent celle-ci compliquée. L'idée de développer des échanges graphiques est séduisante, mais nous émettons des réserves, car les joueurs seraient tentés d'échanger directement les solutions. Ce serait doublement pénalisant, d'une part pour le joueur qui recevrait cette solution sans avoir à réfléchir et d'autre part pour celui qui la donne sans avoir à l'expliquer. Or, c'est aussi en devant expliquer sa démarche que l'étudiant avance. L'audio nous paraît donc plus judicieux même si cela risque d'introduire des problèmes d'infrastructure à mobiliser.

### 4.3.3 Efficacité didactique

La question à se poser ici est : le jeu a-t-il réellement apporté quelque chose d'inédit au niveau des apprentissages, parallèlement aux cours classiques. Or, pour répondre à cette question et vu la méthodologie utilisée, nous ne disposons que de l'avis subjectif des élèves. Qui plus est, l'échange des connaissances sur lequel repose une partie de notre expérience dépend de l'implémentation de la communication dans notre prototype, qui n'est pas comme nous l'avons vu son point fort. Avec ces réserves à l'esprit, nous pouvons quand même tirer des conclusions.

Les étudiants ont perçu une aide réelle quant à leur esprit d'abstraction, grâce à la programmation visuelle couplée au déplacement des personnages. Ils ont mieux appréhendé les programmes comme des suites d'étapes. Il semble aussi qu'ils ont aussi mieux *compris* le rôle et le fonctionnement basique des structures itératives et conditionnelles. En revanche, les résultats paraissent problématiques pour la suite des cours : l'activité créatrice, la formulation et la résolution des problèmes, fondamentales dans la pensée informatique, ne semble pas facilitées. Peut-être la multiplication des niveaux et une meilleure progressivité pourraient-elles aider ? Assez préoccupant aussi, les élèves pensent que la transposition des acquis dans les langages plus formels de la suite du cours ne sera pas facilitée du tout. De nouveau, cette réflexion se base sur une opinion subjective et une expérience limitée de ces programmes. L'introduction progressive de portions de texte rédigé dans les blocs vient à l'esprit, mais impliquerait un développement logiciel conséquent.

# Chapitre 5

## Conclusion

Pour mettre en évidence les apports de notre recherche, il est important de garder à l'esprit le public que nous avons voulu considérer : des étudiants de promotion sociale, en reprise de formation au niveau secondaire et novices en développement informatique. En raison de spécificités cognitives, notamment liées aux capacités d'abstraction, de mémorisation ou de plasticité cérébrale, l'apprentissage de la programmation dans ce contexte est un réel challenge.

Il n'existait pas à notre connaissance d'outils mettant à profit les nombreuses qualités des adultes pour s'adapter à leurs besoins spécifiques et rendre l'enseignement plus efficace. Nous avons donc décidé d'explorer les perspectives ouvertes par l'enseignement de la programmation au travers de jeux sérieux en développant un prototype de SGE sur mesure. Nous avons aussi tenté de valider ce résultat de nos réflexions au travers d'une expérience menée dans le cadre concret d'un cours d'introduction à la programmation informatique.

Les grands axes que nous avons intégrés à notre prototypes sont :

- se focaliser sur l'intégration de quelques notions simples, en s'aidant d'un langage visuel de type « blocs »,
- stimuler la motivation et la persévérance à l'aide de la compétition,
- favoriser l'échange des connaissances entre élèves en favorisant la communication et la collaboration.

En réponse à notre question de recherche et au terme de ce travail, nous pensons pouvoir affirmer que nous sommes parvenus à concilier les concepts antagonistes de plaisir ludique et d'effort d'apprentissage dans un jeux sérieux éducatif. Nous suggérons aussi que l'utilisation de celui-ci dans un cours d'initiation à la programmation pour des adultes débutants peut se montrer bénéfique.

La pratique du jeu a eu un impact qui semble indéniablement positif sur l'ambiance et la dynamique de la classe, développant un climat propice à l'apprentissage. Il existe une cohérence certaine avec les théories de l'apprentissage pour adultes développées dans l'état de l'art : plusieurs éléments établis dans notre méthodologie semblent avoir porté les fruits attendus. Cependant, concernant les apports relatifs à l'acquisition des notions, les choses doivent être nuancées. Une certaine amélioration est certes perceptible, mais clairement limitée. La compréhension du fonctionnement des boucles et des structures conditionnelles semble s'être améliorée, mais l'extrapolation nécessaire à la construction de programmes destinés à résoudre des problèmes plus compliqués semble très vite poser soucis. Qui plus est, la transition du jeu vers le reste du cours semble aussi hasardeuse, les élèves semblant avoir du mal à faire

la jonction entre les deux manières de programmer (visuelle et textuelle). Ces deux derniers points suggèrent l'utilisation du jeu dans le cadre du cours, mais en appoint d'une approche classique.

Ces conclusions étant présentées, il convient maintenant de souligner les limitations de notre travail, et les pistes d'exploration ouvertes par celles-ci.

Tout d'abord, comme tout travail de recherche, celui-ci comporte plusieurs biais. En matière de méthodologie, nous avons un soucis de sélection de notre échantillon. Des circonstances malheureuses ont impacté le déroulement de notre expérience, la mise en quarantaine de l'établissement scolaire ayant directement suivi les séances de jeux. Alors que vingt-trois élèves étaient inscrits à la formation, nous n'avons pu compter que sur sept d'entre eux pour répondre à nos entretiens téléphoniques. Outre leur faible nombre, nous ne pouvons évidemment pas considérer nos participants comme représentatifs de la population sur laquelle porte notre étude.

Les entretiens que nous avons réalisés étaient satisfaisants, mais avec le recul, nous aurions aimé développer plus avant les réponses des interlocuteurs, en clarifiant d'avantage le sens de nos questions et avec plus de relances. Notre manque d'expérience laisse aussi une certaine marge d'amélioration de notre questionnaire.

Jouant à la fois le rôle de chercheur et d'enseignant, nous introduisons un biais conséquent : Les personnes interrogées savent qu'elles continueront à suivre les cours et seront évaluées dans le cadre de ceux-ci. Certains estiment donc probablement qu'il est dans leur intérêt de se montrer flatteur dans leurs réponses, même si nous avons essayé de contrer au maximum cet effet à l'aide d'explications préalables précises.

Nous ne pensons pas avoir révolutionné la recherche sur les jeux sérieux dans ce travail, mais bien avoir ouvert une réflexion qui mériterait d'être développée : comment transmettre le plaisir que l'on peut ressentir à programmer à des adultes qui n'ont pas eu la chance de faire cette découverte plus tôt dans leur vie. Nous pensons sincèrement que le jeu et la communication sont des pierres angulaires sur lesquelles bâtir.

Les pistes à suivre dans ce sens sont nombreuses : le jeu pourrait être développé et amélioré en fonction des constations faites dans ce mémoire, il pourrait alors faire l'objet d'une étude portant sur un panel plus large et varié. Il serait aussi intéressant de développer une analyse quantitative portant sur l'acquisition effective des concepts, mesurée après plusieurs séances de cours classiques.

# Annexe A

## Appendices

### A.1 Questionnaire

#### Profil de l'interlocuteur

- Sexe :
- Âge :
- Diplôme :
- Situation professionnelle :
- Expérience professionnelle précédente :
- Qu'est-ce qui vous a motivé à vous inscrire dans le cours de programmation informatique ?

#### La première dimension (andragogie)

- De manière tout à fait générale, avez-vous apprécié (ressenti) le SG proposé lors du cours de programmation informatique ?
- De manière tout à fait générale, qu'avez-vous pensé (opinion) du SG proposé lors du cours de programmation informatique ?
- Le jeu correspondait à vos besoins ? (oui/non, pourquoi ?)
- Le jeu rendait la formation plus efficace ? (oui/non, pourquoi ?)
- Le jeu reposait sur des expériences vécues vous permettait d'en tirer connaissances à appliquer à l'avenir ? (oui/non, pourquoi ?)
  - concernant les expériences vécues.
  - concernant les expériences à appliquer à l'avenir.
- Le jeu se déroule dans une atmosphère agréable ? (oui/non, pourquoi ?)

#### La deuxième dimension (gameplay/jouabilité)

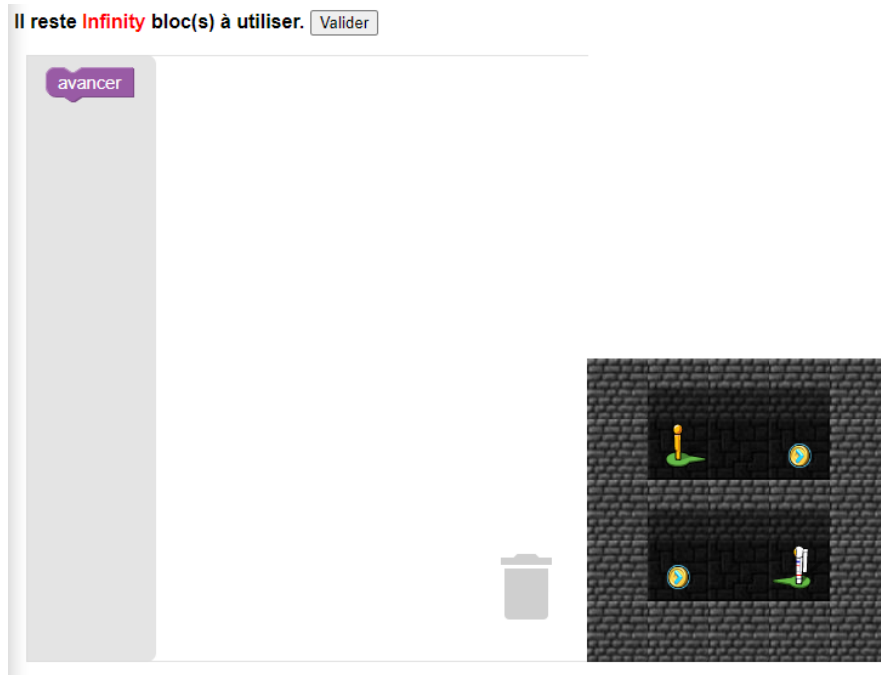
- Qu'avez-vous pensé de la dimension compétitive du SG ?
- Qu'avez-vous pensé de la dimension collaborative du SG ?
- Disposiez-vous de suffisamment d'informations sur le domaine abordé avant et pendant le jeu ?
- Qu'avez-vous pensé des différents labyrinthes présentés dans le jeu ?
- Qu'avez-vous pensé du chat présent dans le jeu ?
- Qu'avez-vous pensé des feedbacks automatiques dans le jeu ?
- Le jeu était réactif ? (oui/non, pourquoi ?)

### La troisième dimension (efficacité didactique)

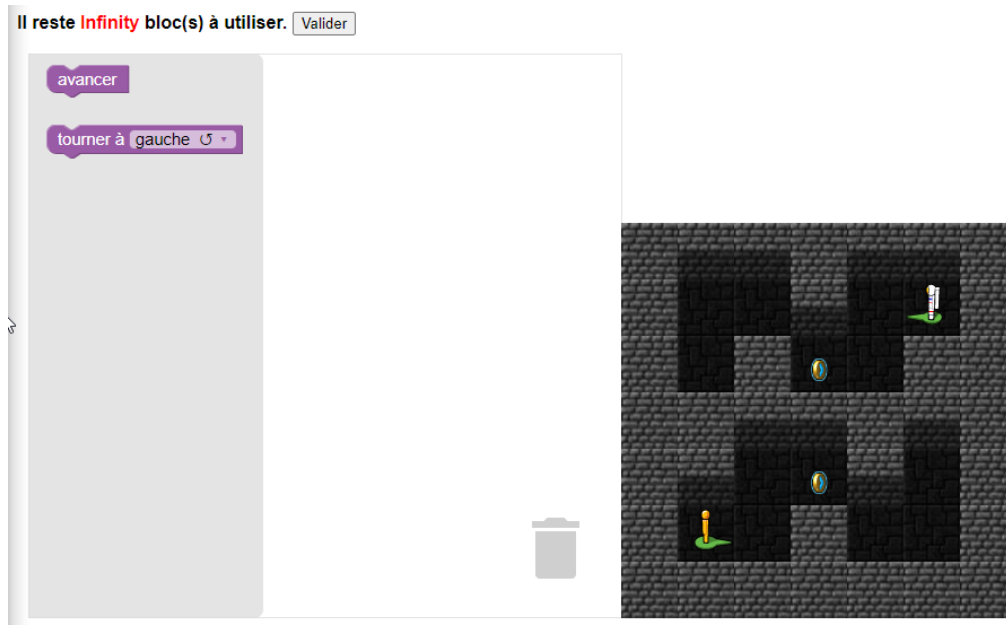
- Dans quelle mesure le jeu a-t-il influencé votre attitude face à un problème à résoudre en termes d'attrait (motivation à résoudre le problème) ? Avez-vous des pistes de solution pour améliorer le nouveau SG à ce niveau ? Argumentez, expliquez, exemplifiez.
- Dans quelle mesure le jeu a-t-il influencé votre attitude face à un problème à résoudre en termes de patience (persévérance) ? Avez-vous des pistes de solution pour améliorer le nouveau SG à ce niveau ?
- Dans quelle mesure le jeu vous a-t-il permis d'acquérir une meilleure compréhension des tests logiques (conditions) ? Avez-vous des pistes de solution pour améliorer le nouveau SG à ce niveau ?
- Dans quelle mesure le jeu vous a-t-il permis d'acquérir une meilleure compréhension des structures de contrôle conditionnelles (si, alors, sinon) ? Avez-vous des pistes de solution pour améliorer le nouveau SG à ce niveau ?
- Dans quelle mesure le jeu vous a-t-il permis d'acquérir une meilleure compréhension des structures de contrôle de type boucles conditionnelles (tant que, faire) ? Avez-vous des pistes de solution pour améliorer le nouveau SG à ce niveau ?
- Dans quelle mesure pensez-vous que le jeu vous a permis de développer la capacité à combiner structures de contrôle et tests logiques afin de résoudre des problèmes concrets ? Avez-vous des pistes de solution pour améliorer le nouveau SG à ce niveau ?

## A.2 Niveaux de jeu

### A.2.1 Niveau 1



### A.2.2 Niveau 2

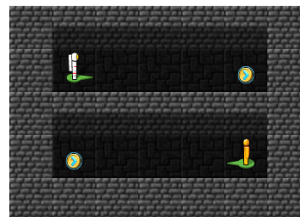




### A.2.3 Niveau 3

Il reste 2 bloc(s) à utiliser.

The image shows a Scratch script editor with three blocks in a vertical stack. The top block is a purple 'move' block labeled 'avancer'. The middle block is a purple 'turn' block labeled 'tourner à gauche' with a dropdown arrow. The bottom block is a green 'repeat' block labeled 'Répète jusqu'à faire' with a small circular icon and a blank input field.



## A.2.4 Niveau 4

Il reste 5 bloc(s) à utiliser. Valider

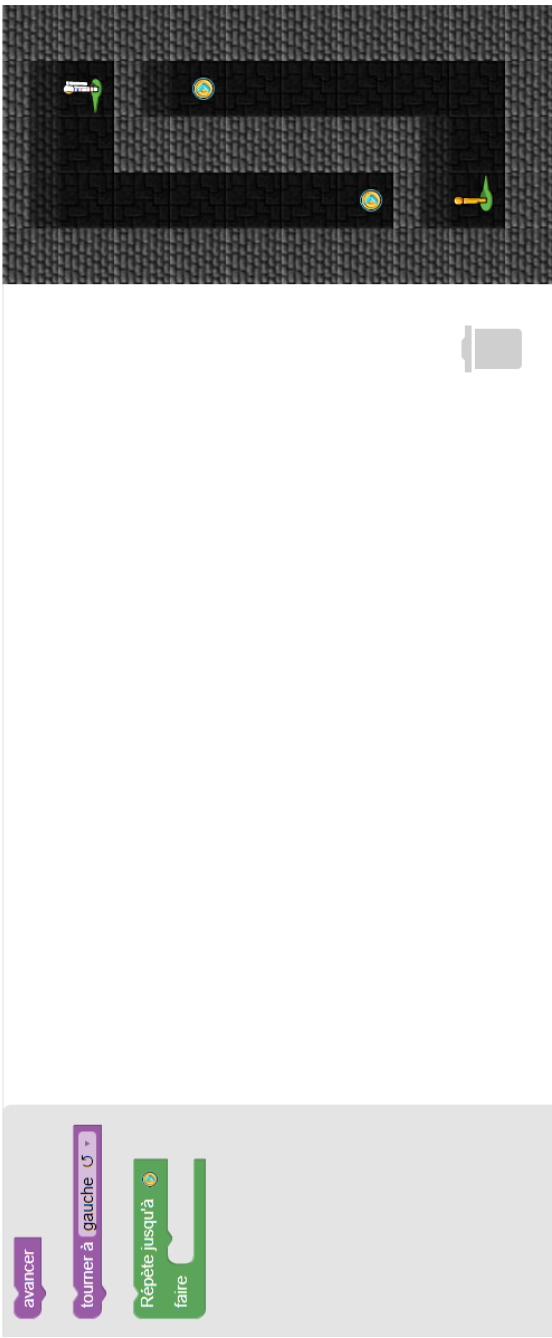
avancer

tourner à gauche ↶

Répète jusqu'à faire

## A.2.5 Niveau 5

Il reste 5 bloc(s) à utiliser. Valider



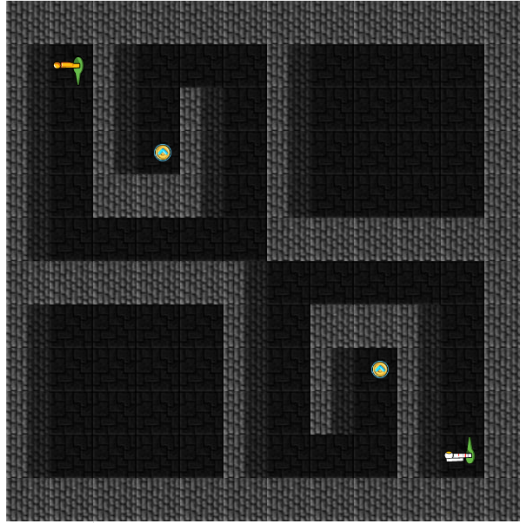
The image shows a Scratch script on the left and a maze on the right. The script consists of three blocks: a purple 'avancer' block, a purple 'tourner à gauche' block with a right arrow icon, and a green 'Répète jusqu'à faire' block containing a yellow 'Répète jusqu'à' block with a blue and yellow target icon. The maze is a 10x10 grid with a black path and grey walls. A green arrow icon is at the top-left of the path, and a yellow target icon is at the bottom-right. A grey trash can icon is in the top-right corner of the maze area.

avancer

tourner à gauche ↻

Répète jusqu'à faire

## A.2.6 Niveau 6

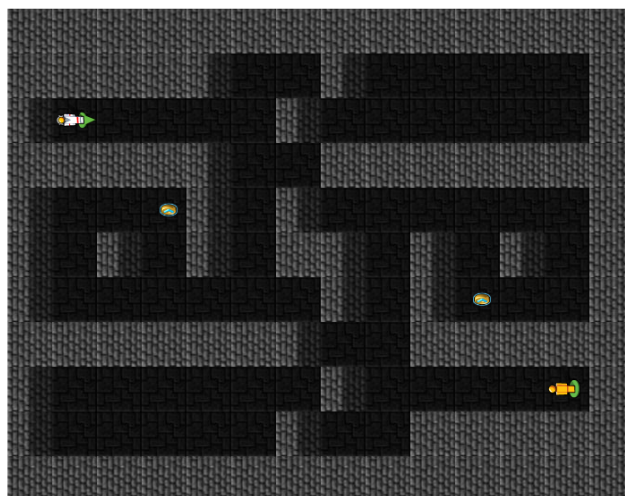


Il reste 5 bloc(s) à utiliser. Valider

A block palette containing four Scratch-like blocks:

- A purple block labeled "avancer".
- A purple block labeled "tourner à gauche" with a right-pointing arrow.
- A blue block labeled "si chemin devant" with a "faire" tab.
- A green block labeled "Répète jusqu'à" with a "faire" tab and a "Répète jusqu'à" icon.

## A.2.7 Niveau 7



Il reste 5 bloc(s) à utiliser.

avancer

tourner à gauche

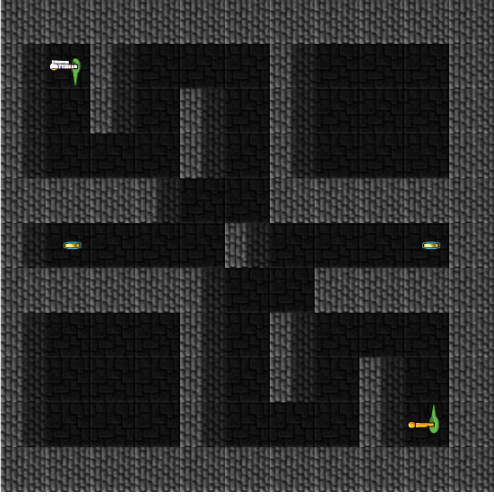
si chemin devant

faire

Répète jusqu'à


faire

### A.2.8 Niveau 8




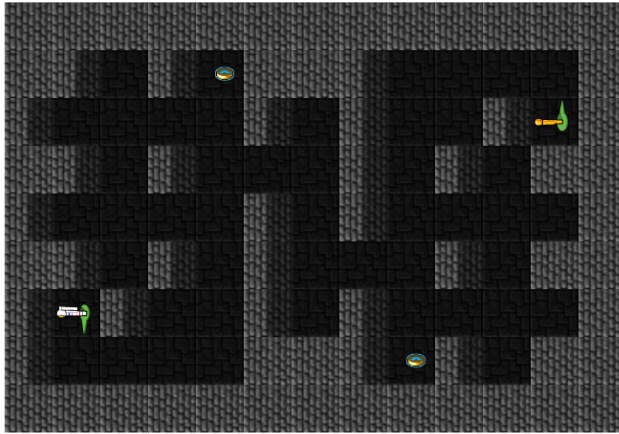
Il reste **10 bloc(s)** à utiliser.

- avancer
- tourner à gauche ↶
- si chemin devant  
faire
- Répète jusqu'à  
faire

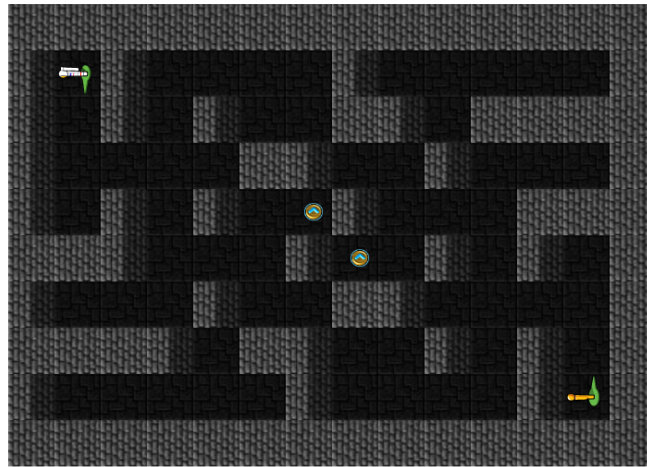


### A.2.9 Niveau 9

Il reste **7** bloc(s) à utiliser.



### A.2.10 Niveau 10



Il reste **10 bloc(s)** à utiliser.

avancer

tourner à gauche ↶

si chemin devant  
faire

si chemin devant  
faire  
sinon

Répète jusqu'à  
faire



# Annexe B

## Bibliographie

- [Abt, 1970] Abt, C. C. (1970). *Serious games*. New York : Viking Press.
- [Allal-Cherif and Bajard, 2011] Allal-Cherif, O. and Bajard, A. (2011). Serious games as a basis for human resource management : examples from the banking sector. *MCIS 2011 Proceedings*, 53 :1–13.
- [Annetta et al., 2009] Annetta, L. A., Minogue, J., Holmes, S. Y., and Cheng, M.-T. (2009). Investigating the impact of video games on high school students' engagement and learning about genetics. *Computers & Education*, 53(1) :74–85.
- [Averlant, 2007] Averlant, P. (2007). Sur la formation pour adultes. *VST-Vie sociale et traitements*, (4) :101–103.
- [Bourgeois, 1998] Bourgeois, E. (1998). Apprentissage, motivation et engagement en formation. *Éducation permanente*, 136(3) :101–109.
- [Boutinet, 2013] Boutinet, J.-P. (2013). L'andragogie, une préoccupation équivoque pour aborder des problèmes nouveaux. *Que sais-je ?*, 5 :81–95.
- [Broekemier, 2002] Broekemier, G. M. (2002). A comparison of two-year and four-year adult students : Motivations to attend college and the importance of choice criteria. *Journal of Marketing for Higher Education*, 12(1) :31–48.
- [Brown and Bimrose, 2018] Brown, A. and Bimrose, J. (2018). Drivers of learning for the low skilled. *International Journal of Lifelong Education*, 37(2) :151–167.
- [Charlton and Danforth, 2010] Charlton, J. and Danforth, I. (2010). Validating the distinction between computer addiction and engagement : Online game playing and personality. *Behaviour and Information Technology*.
- [Class and Schneider, 2013] Class, B. and Schneider, D. (2013). La recherche design en education : vers une nouvelle approche? *Frantice.net*, 7 :5–16. ID : unige :34459.
- [Cohard, 2015] Cohard, P. (2015). L'apprentissage dans les serious games : proposition d'une typologie. *@ GRH*, (3) :11–40.
- [Coller and Scott, 2009] Coller, B. D. and Scott, M. J. (2009). Effectiveness of using a video game to teach a course in mechanical engineering. *Computers & Education*, 53(3) :900–912.
- [Collins, 2010] Collins, A. (2010). Design experiments. *International encyclopedia of education*. Evanston, IL, USA.
- [Communauté Française de Belgique, 1991] Communauté Française de Belgique (1991). Décret organisant l'enseignement de promotion sociale. Article 7.

- [Consolvo et al., 2008] Consolvo, S., Klasnja, P., McDonald, D. W., Avrahami, D., Froehlich, J., LeGrand, L., Libby, R., Mosher, K., and Landay, J. A. (2008). Flowers or a robot army? : encouraging awareness & activity with personal, mobile displays. In *Proceedings of the 10th international conference on Ubiquitous computing*, pages 54–63. ACM.
- [Demol, 1995] Demol, J. N. (1995). L’entrée en formation. *Education permanente*, (125) :27–38.
- [Denning, 2003] Denning, P. J. (2003). Great principles of computing. *Communications of the ACM*, 46(11) :15–20.
- [Denning, 2005] Denning, P. J. (2005). Is computer science science? *Communications of the ACM*, 48(4) :27–31.
- [Denning, 2006] Denning, P. J. (2006). The profession of it beyond computational thinking. *Communications of the ACM*, 52(6) :28–30.
- [Denning, 2007] Denning, P. J. (2007). Computing is a natural science. *Communications of the ACM*, 50(7) :13–18.
- [Djaouti, 2011] Djaouti, D. (2011). *Serious Game Design : considérations théoriques et techniques sur la création de jeux vidéo à vocation utilitaire*. PhD thesis, Université de Toulouse, Université Toulouse III-Paul Sabatier.
- [Djaouti, 2014] Djaouti, D. (2014). De l’utilité de l’appellation « serious game ». le jeu est-il l’apanage du divertissement? *Interfaces numériques*, 3 :409–429.
- [Djaouti, 2016] Djaouti, D. (2016). Serious games pour l’éducation : utiliser, créer, faire créer? *Tréma*, (44) :51–64.
- [Doray et al., 2012] Doray, P., Kamanzi, P. C., Laplante, B., and Street, M. C. (2012). Les retours aux études postsecondaires : une expression de l’éducation tout au long de la vie? *Formation emploi. Revue française de sciences sociales*, (120) :75–100.
- [Dowek et al., 2013] Dowek, G., Archambault, J.-P., Baccelli, E., Cimelli, C., Cohen, A., Eisenbeis, C., Viéville, T., Wack, B., Bersini, H., and Le Blanc, G. (2013). *Informatique et sciences du numérique : Édition spéciale Python*. Eyrolles.
- [Erhel and Jamet, 2012] Erhel, S. and Jamet, E. (2012). Comprendre les effets des serious game éducatifs sur l’apprentissage et la motivation.
- [Furber, 2012] Furber, S. (2012). *Shut down or restart? : The way forward for computing in UK schools*. The Royal Society.
- [González, 2015] González, M. R. (2015). Computational thinking test : Design guidelines and content validation. In *Proceedings of EDULEARN15 conference*, pages 2436–2444.
- [Gredler, 2004] Gredler, M. E. (2004). Games and simulations and their relationships to learning. *Handbook of research on educational communications and technology*, 2 :571–581.
- [Habgood, 2007] Habgood, M. P. J. (2007). *The effective integration of digital games and learning content*. PhD thesis, University of Nottingham.
- [Hays, 2005] Hays, R. T. (2005). The effectiveness of instructional games : A literature review and discussion. Technical report, Naval Air Warfare Center Training Systems Div Orlando Fl.

- [Hemmendinger, 2010] Hemmendinger, D. (2010). A plea for modesty. *Acm Inroads*, 1(2) :4–7.
- [Henry et al., 2017] Henry, J., Nguyen, A., and Vandeput, E. (2017). L’informatique et le numérique dans la classe. *Qui, quoi, comment*, page 14.
- [Howell, 2005] Howell, K. (2005). Games for health conference 2004 : Issues, trends, and needs unique to games for health. *Cyberpsychology and behavior : the impact of the Internet, multimedia and virtual reality on behavior and society*, 8 :103–9.
- [Hubrecht, 2016] Hubrecht, Vinciane & Flament, N. (2016). Andra-quoi? gogie! ah... *Zoom 2.0*, pages 8–10.
- [Hunicke et al., 2004] Hunicke, R., LeBlanc, M., and Zubek, R. (2004). Mda : A formal approach to game design and game research. In *Proceedings of the AAAI Workshop on Challenges in Game AI*, volume 4, page 1722.
- [Järvinen, 2007] Järvinen, P. (2007). Action research is similar to design science. *Quality & Quantity*, 41(1) :37–54.
- [Joseph M., 2018] Joseph M., M. J. (2018). Les motifs d’engagement en formation d’alphabétisation [ressource électronique] : une enquête quali-quantitative auprès d’apprenants de lire et écrire bruxelles.
- [Kafai et al., 1995] Kafai, Y. B., Kafai, Y. B., and Kafai, Y. B. (1995). *Minds in play : Computer game design as a context for children’s learning*. Routledge.
- [Ke, 2011] Ke, F. (2011). A qualitative meta-analysis of computer games as learning tools. In *Gaming and simulations : Concepts, methodologies, tools and applications*, pages 1619–1665. IGI Global.
- [Kebritchi et al., 2010] Kebritchi, M., Hirumi, A., and Bai, H. (2010). The effects of modern mathematics computer games on mathematics achievement and class motivation. *Computers & education*, 55(2) :427–443.
- [Knowles et al., 2012] Knowles, M. S., Holton III, E. F., and Swanson, R. A. (2012). *The adult learner*. Routledge.
- [Knowles et al., 2015] Knowles, M. S., III, E. F. H., and Swanson, R. A. (2015). *The adult learner : the definitive classic in adult education and human resource development (6th)*.
- [Laamarti et al., 2014] Laamarti, F., Eid, M., and Saddik, A. E. (2014). An overview of serious games. *International Journal of Computer Games Technology*, 2014 :11.
- [Lindeman, 1926] Lindeman, E. C. (1926). The meaning of adult education. new york : New republic. *Nottingham Andragogy Group (1983) Towards a developmental theory of andragogy. University of Nottingham : Nottingham*.
- [Liu et al., 2011] Liu, C.-C., Cheng, Y.-B., and Huang, C.-W. (2011). The effect of simulation games on the learning of computational problem solving. *Computers & Education*, 57(3) :1907–1918.
- [Loisier, 2015] Loisier, J. (2015). Qu’est-ce qu’apprendre? en ligne : [http://www.refad.ca/documents/Etude\\_Jeux\\_serieux\\_en\\_FAD.pdf](http://www.refad.ca/documents/Etude_Jeux_serieux_en_FAD.pdf).
- [Léotard-Sommer, 2019] Léotard-Sommer, C. d. (2019). *Sans Bacchus et Vénus, la Galerie se refroidit : dispositif libérant le programme de l’intégralité du décor de la Galerie du Roi de Fontainebleau*. PhD thesis. Thèse de doctorat dirigée par Bertrand, Pascal-François Histoire de l’Art Bordeaux 3 2019.

- [Malone, 1981] Malone, T. W. (1981). Toward a theory of intrinsically motivating instruction. *Cognitive science*, 5(4) :333–369.
- [Mayer and Johnson, 2010] Mayer, R. E. and Johnson, C. I. (2010). Adding instructional features that promote learning in a game-like environment. *Journal of Educational Computing Research*, 42(3) :241–265.
- [McKenney and Reeves, 2014] McKenney, S. and Reeves, T. C. (2014). Educational design research. In *Handbook of research on educational communications and technology*, pages 131–140. Springer.
- [McNicholl, 2018] McNicholl, R. (2018). Computational thinking using code.org. *Hello World*, 4 :37.
- [Michaud and Alvarez, 2008] Michaud, L. and Alvarez, J. (2008). Serious games. *Advergaming, edugaming, training... IDATE Consulting & Research*.
- [Monterrat et al., 2012] Monterrat, B., Lavoué, E., and George, S. (2012). Learning game 2.0 : Support for game modding as a learning activity. In *Proceedings of the 6th European Conference on Games Based Learning*, pages 340–347.
- [Moreno, 2004] Moreno, R. (2004). Decreasing cognitive load for novice students : Effects of explanatory versus corrective feedback in discovery-based multimedia. *Instructional science*, 32(1-2) :99–113.
- [Moreno and Mayer, 2000] Moreno, R. and Mayer, R. E. (2000). Engaging students in active learning : The case for personalized multimedia messages. *Journal of educational psychology*, 92(4) :724.
- [Méli, 2017] Méli, B. (2017). « america’s army » : le jeu comme levier de recrutement. en ligne : [www.journaldunet.com/ebusiness/crm-marketing/selection/dix-Serious-games-marketing-originaux/america-s-army-le-jeu-comme-levier-de-recrutement.shtml](http://www.journaldunet.com/ebusiness/crm-marketing/selection/dix-Serious-games-marketing-originaux/america-s-army-le-jeu-comme-levier-de-recrutement.shtml).
- [Papastergiou, 2009] Papastergiou, M. (2009). Exploring the potential of computer and video games for health and physical education : A literature review. *Computers & Education*, 53(3) :603–622.
- [Perrenoud, Ph., 2004] Perrenoud, Ph. (2004). Qu’est-ce qu’apprendre ? en ligne : [https://www.unige.ch/fapse/SSE/teachers/perrenoud/php\\_main/php\\_2004/2004\\_08.html](https://www.unige.ch/fapse/SSE/teachers/perrenoud/php_main/php_2004/2004_08.html).
- [Petit, 2011] Petit, L. (2011). Les sujets en formation des adultes et la dimension enfant-élève-apprenant. *Recherches en didactiques*, (1) :85–96.
- [Pivec and Pivec, 2008] Pivec, M. and Pivec, P. (2008). Games in schools. *ISFE-EUN Partnership*.
- [Proulx, 2002] Proulx, S. (2002). Trajectoires d’usages des technologies de communication : les formes d’appropriation d’une culture numérique comme enjeu d’une « société du savoir ». In *Annales des télécommunications*, volume 57, pages 180–189. Springer.
- [Randel et al., 1992] Randel, J. M., Morris, B. A., Wetzell, C. D., and Whitehill, B. V. (1992). The effectiveness of games for educational purposes : A review of recent research. *Simulation & gaming*, 23(3) :261–276.
- [Ratan and Ritterfeld, 2009] Ratan, R. and Ritterfeld, U. (2009). Classifying serious games. *Serious games : Mechanisms and effects*, pages 10–24.

- [Raybourn, 2007] Raybourn, E. (2007). Applying simulation experience design methods to creating serious game-based adaptive training systems. *Interacting with Computers*, 19 :206–214.
- [Sanchez, 2011] Sanchez, E. (2011). Usage d’un jeu sérieux dans l’enseignement secondaire. modélisation comportementale et épistémique de l’apprenant. *Revue d’Intelligence Artificielle*, 25 :203–222.
- [Sauvé, 2008] Sauvé, L. (2008). Concevoir des jeux éducatifs en ligne. en ligne : [http://www.savie.ca/SAVIE/Publications/Jeux/060\\_Sauve%20Ludovia\\_08.pdf](http://www.savie.ca/SAVIE/Publications/Jeux/060_Sauve%20Ludovia_08.pdf). Accédé le 15-05-2022.
- [Shackelford et al., 2006] Shackelford, R., McGettrick, A., Sloan, R., Topi, H., Davies, G., Kamali, R., Cross, J., Impagliazzo, J., LeBlanc, R., and Lunt, B. (2006). Computing curricula 2005 : The overview report. *SIGCSE Bull.*, 38(1) :456–457.
- [Sherry, 2004] Sherry, J. L. (2004). Flow and media enjoyment. *Communication theory*, 14(4) :328–347.
- [Stokes, 1997] Stokes, D. (1997). Pasteur’s quadrant : Basic science and technological innovation (the brookings institution, washington dc).
- [Vogel et al., 2006] Vogel, J. J., Vogel, D. S., Cannon-Bowers, J., Bowers, C. A., Muse, K., and Wright, M. (2006). Computer gaming and interactive simulations for learning : A meta-analysis. *Journal of Educational Computing Research*, 34(3) :229–243.
- [Wastiau et al., 2009] Wastiau, P., Kearney, C., Van den Berghe, W., Joyce, A., and Gerhard, P. (2009). *Quels usages pour les jeux électroniques en classe ? : principaux résultats de l’étude : rapport de synthèse*. European Schoolnet.
- [Wing, 2006] Wing, J. M. (2006). La pensée informatique. *Communications of the ACM*, 49(3).
- [Wing, 2011] Wing, J. M. (2011). Research notebook : Computational thinking what and why? en ligne : <https://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why>. Accédé le 05-05-2022.
- [Witmer and Singer, 1998] Witmer, B. G. and Singer, M. J. (1998). Measuring presence in virtual environments : A presence questionnaire. *Presence*, 7(3) :225–240.
- [Wix, 2012] Wix, A. (2012). Jouer en classe, est-ce bien sérieux? bilan de l’expérimentation académique sur les usages de jeux sérieux au collège et au lycée. *Académie d’Aix-Marseille*.
- [Yadav et al., 2014] Yadav, A., Mayfield, C., Zhou, N., Hambruch, S., and Korb, J. T. (2014). Computational thinking in elementary and secondary teacher education. *ACM Transactions on Computing Education (TOCE)*, 14(1) :5.
- [Yim, 2008] Yim, J. W. (2008). *Computer-aided exercise*. PhD thesis.
- [Zyda, 2005] Zyda, M. (2005). From visual simulation to virtual reality to games. *Computer*, 38 :25 – 32.

# Acronymes

**ACM** Association for Computing Machinery. 9, 10

**CT** Computational Thinking. 6

**EPS** Enseignement de Promotion Sociale. 6, 7

**RDE** Recherche Designe en Education. 40, 53

**SG** Serious Game. 7, 8, 25, 32, 33, 38, 42, 53, 64

**SGE** Serious Game for Education. 36–38, 46, 53, 66