

THESIS / THÈSE

MASTER EN SCIENCES INFORMATIQUES

Réalisation d'un simulateur d'éclipses à usage didactique

Van Hemelrijck, Alain

Award date:
1992

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.

**FACULTES
UNIVERSITAIRES
N.D. DE LA PAIX
NAMUR**

INSTITUT D'INFORMATIQUE

**REALISATION D'UN SIMULATEUR
D'ECLIPSES
A USAGE DIDACTIQUE**

Promoteur : C. CHERTON

**Mémoire présenté par :
Alain VAN HEMELRIJCK**

RUE GRANGAGNAGE, 21, B - 5000 NAMUR (BELGIUM)

Facultés Universitaires Notre-Dame de la Paix

Institut d'informatique

Rue Grangagnage, 21, B - 5000 NAMUR (BELGIUM)

Réalisation d'un simulateur d'éclipses à usage didactique

Résumé

Le projet de ce mémoire était de développer un outil didactique sur l'astronomie. Nous avons dû nous limiter à l'étude du phénomène des éclipses et à la réalisation d'un simulateur d'éclipses sur PC. Nous proposons en plus des pistes de réflexions pour améliorer ce simulateur et le compléter par l'intégration d'autres sujets d'astronomie.

Abstract

The project of this memoir was to develop a didactic tool on astronomy. We had to limit us to the study of eclipses' phenomenon and to the realization of an eclipses' simulator on PC. We propose reflections' tracks to improve this simulator and complete it with the integration of other subjects.

Mémoire de licence et maîtrise en informatique

Juin 1992

Promoteur : Prof. C. CHERTON

Mémorisant : A. VAN HEMELRIJCK

Je tiens à remercier ici tous ceux qui
ont permis la réalisation de ce mémoire,
et tout particulièrement Monsieur C. Cherton

A. Van Hemelryck

Table des matières

Introduction	p 1
Chapitre 1 : Les éclipses	p 4
1. Les éclipses de Lune	p 4
2. Les éclipses de Soleil	p 8
3. Périodicité des éclipses sur Terre	p 10
4. Justification de l'intérêt scientifique de l'observation des éclipse	p 11
Chapitre 2 : Le simulateur	p 13
1. Le projet	p13
2. Modélisation du problème en trois dimensions	p 13
3. Représentation du phénomène à l'écran : projections en perspective	p 16
4. Réalisation du programme	p 19
4.1. Les structures de données	p 19
4.2. La structure du programme	p 22
4.3. Réalisation de l'interface utilisateur	p 23
4.3.1. Réalisation de l'interface traditionnelle	p 24
4.3.2. Réalisation de l'interface visuelle	p 24
4.4. Réalisation de l'application	p 25
4.4.1. Calcul des paramètres de la matrice de changement de repère	p 25
4.4.2. Calcul de l'orbite des corps dans le repère observateur	p 26

4.4.3. Calcul et affichage de la position des corps au	
temps t	p 27
4.4.3.1. Réalisation de la procédure générale d'affichage	p 27
4.4.3.2. Réalisation de la procédure particulière	
d'affichage	p 30
a) Calcul du triangle du cône d'ombre	p 31
b) Calcul de la zone d'ombre sur le corps	p 32

Chapitre 3 : Manuel d'utilisation du programme p 36

1. But du programme	p 36
2. Matériel nécessaire	p 36
3. Modélisation du système	p 36
4. Structure des fichiers de paramètres	p 37
5. Chargement du programme	p 40
6. Utilisation du programme	p 41
6.1. Initialisation des paramètres	p 41
6.2. Activation d'un cadre	p 42
6.3. Interface traditionnelle	p 42
6.3.1. Explication des paramètres que l'utilisateur	
peut modifier interactivement	p 43
6.4. Interface visuelle	p 45
6.5. Chargement et sauvegarde des paramètres	p 49
6.6. Quitter le programme	p 49
6.7. La simulation	p 49
7. L'usage didactique	p 51

Chapitre 4 : Les améliorations possibles du programme de simulation	p 53
1. Plusieurs vues simultanément affichées à l'écran	p 54
2. Modifications d'autres paramètres lors de la simulation	p 55
3. Pause automatique lors de phénomènes intéressants	p 56
4. Affichage des zones de pénombre	p 56
5. Affichage d'ombres qui ne sont plus des cônes	p 56
6. Affichage des différentes orbites à tout moment	p 57
7. Affichage de la projection de l'ombre d'un corps sur un autre	p 58
8. Observateur sur un corps qui tourne sur lui-même	p 58
9. Observateur sur un corps qui ne tourne pas sur lui-même	p 59
10. Modification plus aisée des positions du centre de l'orbite de chaque corps	p 59
 Chapitre 5 : Projet d'avenir	 p 61
1. Un programme d'interface général	p 61
2. Sujets à développer	p 63
 Annexe	 p 64
 Bibliographie	 p 66

Liste des figures

Figure 1 : Eclipse totale de Lune	p 5
Figure 2 : Eclipse partielle de Lune	p 6
Figure 3 : Eclipse par la pénombre	p 7
Figure 4 : éclipse totale de Lune avec réfraction des rayons du Soleil dans l'atmosphère	p 8
Figure 5 : éclipse totale de Soleil	p 9
Figure 6 : éclipse annulaire de Soleil	p 9
Figure 7 : orbite de la Lune : saison où une éclipse est impossible	p 10
Figure 8 : orbite de la Lune : saison où une éclipse est possible	p 11
Figure 9 : modélisation pour le corps c_i	p 14
Figure 10 : projection en perspective	p 17
Figure 11: volume de vision	p 19
Figure 12 : structure du programme	p 23
Figure 13 : projection du cône d'ombre	p 31
Figure 14 : rappel : le triangle du cône d'ombre	p 31
Figure 15 : vue à l'écran	p 33
Figure 16 : distance entre extrémité de la nuit et centre du corps	p 33
Figure 17 : construction dans l'espace	p 34
Figure 18 : affichage lors de l'appel à VUE	p 47
Figure 19 : écran de la simulation	p 50
Figure 20 : cas de non éclipse alors que la vue unique depuis le plan de l'écliptique pourrait laisser supposer le contraire	p 54
Figure 21 : cas d'éclipse : la même projection sur l'écliptique peut signifier qu'il y a une éclipse	p 55
Figure 22 : grandeur de la zone de pénombre	p 57
Figure 23 : ombre et pénombre dans le cas d'un corps lumineux plus petit que le corps non lumineux	p 57

Introduction

De tous temps, des hommes se sont interrogés sur la structure de l'univers. Copernic, Galilée et bien d'autres encore ont collaboré à la vision que nous en avons aujourd'hui. Par exemple, Copernic a élaboré une nouvelle théorie des mouvements planétaires en passant du géocentrisme à l'héliocentrisme, Galilée a montré que la Terre était ronde, Kepler a énoncé les lois qui régissent le mouvement des corps sur leur orbite, C'est ainsi que ces personnes ont contribué à la naissance d'une science, l'astronomie, dont l'importance n'a cessé de croître au cours des siècles.

En nous référant au Larousse, nous définirons l'astronomie comme une science qui étudie la position, les mouvements et la constitution des corps célestes. Mais historiquement, elle se préoccupa d'abord exclusivement de l'étude de la Terre et de son proche voisinage, au point que longtemps, en Europe, on a pensé que la Terre était le centre de l'univers. Ce n'est qu'aux XVI et XVIIèmes siècles que l'on se rendit compte que cette conception géocentrique était trop simpliste. Dès lors, le sujet d'étude de l'astronomie s'élargit pour prendre en considération le système solaire dans un premier temps avant d'étudier, vers le début de ce siècle, les autres galaxies et arriver ainsi à l'univers tel qu'on le connaît aujourd'hui.

Aujourd'hui, l'astronomie est devenue une science très populaire : ne fut-ce que par l'intermédiaire de la télévision lors de la diffusion des images de la sonde Voyager, le suivi de la trajectoire d'une comète, ou encore plus récemment lors de la retransmission d'un lancement d'une navette spatiale. Mais notre connaissance est souvent passive et superficielle, au point que nous ne sommes pas tous capables de préciser le domaine d'étude de l'astronomie, et encore moins d'expliquer certains phénomènes qu'elle décrit, même s'ils nous sont très familiers : pensons au phénomène des marées, aux phases de la Lune, aux éclipses, au raccourcissement du jour et de la nuit selon les saisons,

Pourtant, aujourd'hui, la littérature sur le sujet s'accompagne d'outils très performants permettant d'aborder facilement l'étude des corps célestes. Notamment, il existe un peu partout dans le monde des Planétariums. Un planétarium est, rappelons le, un système qui permet de projeter sur un dôme la représentation de la voûte céleste à une époque donnée. Le point fort de cet outil est qu'il permet de simuler, en accéléré, le mouvement de tous les corps qu'il représente, ce qui est impossible à faire sur un tableau ou sur une feuille de papier. On peut ainsi observer, en une après-midi, les différentes phases de la lune, la position des planètes et leurs mouvements relatifs au cours d'une année.

Cependant, même si cet outil est très performant et d'une bonne valeur didactique, il n'est pas disponible à tout moment. Un professeur voulant expliquer les phases de la lune à ses élèves pourrait visiter un planétarium avec eux. Mais si, par la suite, il désire prolonger l'explication en classe, il perdrait le bénéfice de l'animation du planétarium. Ainsi, pour ce professeur et pour tous ceux pour qui une telle visite n'est pas possible, il serait intéressant d'avoir un outil à la portée de tous, capable de produire des animations, en un mot un bon support pour des cours d'astronomie. Or, actuellement, la micro-informatique est implantée dans la plupart des écoles. De plus, à partir d'un micro-ordinateur on peut exécuter de nombreux calculs ce qui est nécessaire pour produire des séquences animées sur un écran. Pourquoi, dès lors ne pas utiliser un PC pour comprendre et étudier l'astronomie ?

C'est dans cette optique que nous avons choisi le sujet de ce mémoire : nous voudrions simuler, grâce à un ordinateur, une sorte de planétarium. Cependant l'étendue des phénomènes représentés par un planétarium est très vaste et il serait utopique d'imaginer que l'on pourrait, lors d'un mémoire et grâce à un PC, simuler tous ceux-ci. En effet, à l'occasion d'un mémoire, le temps est limité et il serait par conséquent impossible d'étudier et de programmer tous les phénomènes abordés par le planétarium. Quant au PC, malgré ses bonnes performances de calcul et la possibilité de produire des animations à l'écran, il a ses limites :

- Le temps de calcul entre deux affichages successifs ne peut pas être trop long sous peine de ne plus avoir une animation, mais un mouvement discontinu, haché Il faudra donc veiller à ne pas consommer trop de temps CPU.
- La précision de calcul sur un PC est nettement moins fine que sur un "super calculateur", et il est donc impossible de fournir, par exemple, les mêmes prévisions que sur un planétarium.
- L'écran est très petit par rapport au dôme du planétarium, et on ne sait donc pas représenter autant à l'écran que sur le dôme.
- L'écran n'occupe qu'une partie plane de l'espace qui l'entoure, ce qui change complètement la perception du phénomène par l'observateur. Dans un planétarium l'observateur est entouré par la projection du ciel, le phénomène qu'il observe se déroule tout autour de lui sur une surface sphérique, tandis que sur l'ordinateur et donc sur l'écran qu'il regarde, le phénomène est mis à plat sur une toute petite portion de l'espace qui l'entoure.

Puisqu'il est impossible de redévelopper complètement un planétarium sur un ordinateur de type PC, il faut donc limiter le sujet d'étude de ce mémoire. C'est ainsi que nous avons établi une liste de phénomènes susceptibles (voir chapitre sur les projets d'avenir) d'être traités sur PC. Nous en avons retenu un pour ce travail. Les

autres sujets pourront être développés par la suite dans d'autres mémoires et intégrés ensuite dans un programme général d'étude de l'astronomie.

Dans ce mémoire, nous avons décidé de simuler le phénomène des éclipses, et plus particulièrement nous visualiserons le mouvement des trois corps qui les produisent. Car la difficulté dans le problème des éclipses, et d'ailleurs pour tous les problèmes qui se déroulent dans l'espace, est de pouvoir visualiser un phénomène tridimensionnel en mouvement.

Dans un premier temps, nous rappellerons les conditions indispensables pour observer une éclipse, puis, nous expliquerons la programmation de la simulation du phénomène et nous décrirons l'utilisation du programme et l'usage que l'on pourrait en faire comme outil didactique. Ensuite, nous commenterons les limitations que nous avons dû y apporter, avant de conclure par une synthèse des opportunités de développement dans le futur. Ceci pour compléter l'approche de l'astronomie grâce à l'ordinateur.

Les éclipses

Ce chapitre a pour double but de définir les éclipses de manière générale et d'en commenter de manière détaillée deux cas particuliers : les éclipses de Soleil et les éclipses de Lune. En conclusion de ce chapitre, nous essayerons de justifier l'importance, d'un point de vue scientifique, de l'observation des éclipses.

Le terme éclipse signifie occultation ou obscurcissement d'un corps céleste par un autre. De nombreux cas de ce phénomène sont observables depuis la Terre : citons par exemple, les éclipses des satellites de Jupiter, ou encore celles d'étoiles doubles comme Algol, et bien sûr les éclipses de Lune et les éclipses de Soleil.

1. Les éclipses de Lune

Pour qu'une éclipse de Lune puisse être observée sur Terre, il suffit que la Lune soit occultée entièrement ou en partie par la Terre. Tel est le cas chaque fois que l'ombre (*1) ou la pénombre (*2) de la Terre atteint la Lune. On peut en outre observer que la Lune se trouve alors nécessairement dans sa phase dite de *Pleine Lune*. En effet, la phase de Pleine Lune a lieu chaque fois que le Soleil, la Terre et la Lune se trouvent tous trois dans un plan perpendiculaire au plan de l'écliptique (plan de l'orbite terrestre) alignés dans l'ordre précité (pour les phases de la Lune : voir référence bibliographique 1). Or on observe une éclipse de Lune chaque fois que le Soleil, la Terre et la Lune sont quasi alignés dans cet ordre, ce qui est une condition suffisante pour se trouver dans la phase de Pleine Lune.

Différents types d'éclipses de Lune sont observables selon les positions relatives de la Terre, de la Lune et du Soleil.

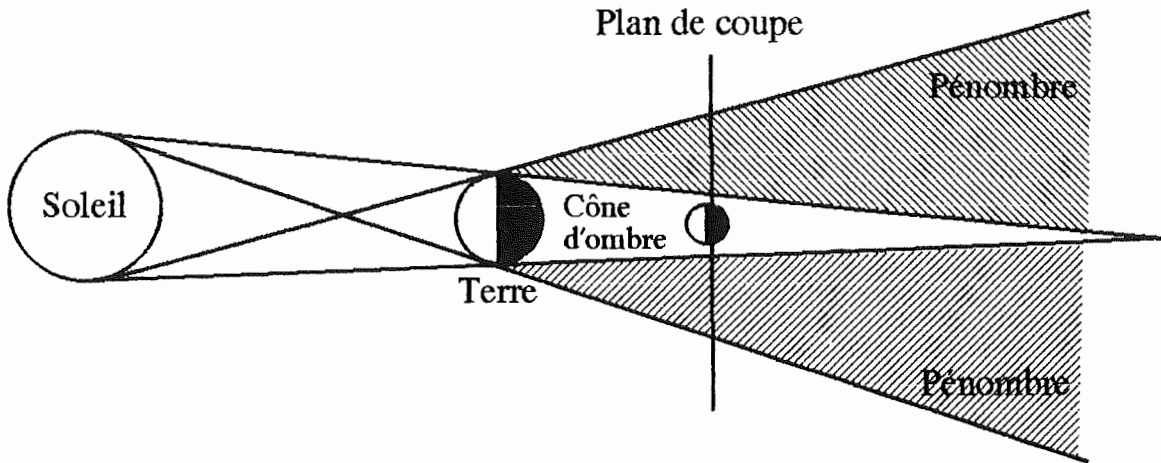
- Eclipse totale :

Etant donné le diamètre et la distance de la Lune par rapport à la Terre, le diamètre de la section du cône d'ombre de la Terre au niveau de la Lune est plus important que le diamètre de la Lune (2,6 fois plus grand 9000 km pour 3452 km). Par conséquent, il est possible qu'au cours de son déplacement la Lune se trouve totalement dans le cône d'ombre de la Terre : on parle alors d'éclipse totale.

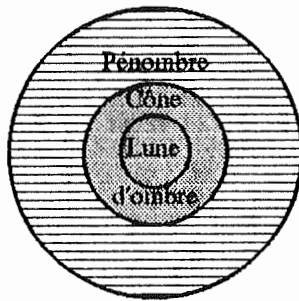
*1 L'ombre de la Terre est une partie du cône défini par l'ensemble des droites tangentes extérieures aux deux corps (voir figure 1)

*2 La pénombre de la Terre est une partie du cône défini par l'ensemble des droites tangentes extérieures intérieures aux deux corps (voir figure 1)

Vue de profil



Vue en coupe

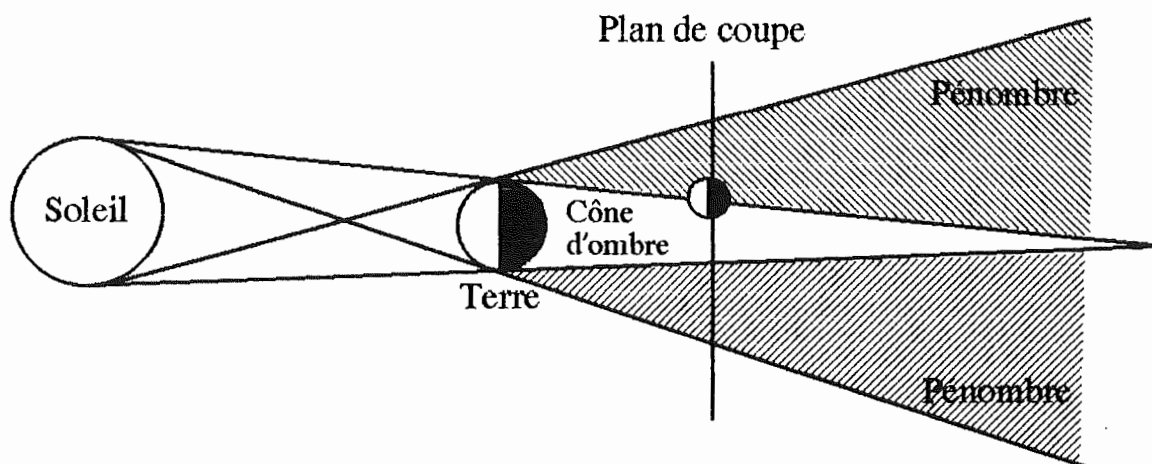


*Eclipse totale de Lune
figure 1*

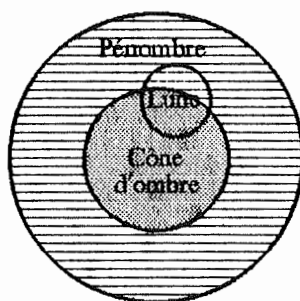
- Eclipse partielle :

Si à un moment donné la position de la Lune est telle qu'une partie seulement de sa surface est recouverte par l'ombre de la Terre, on parle d'éclipse partielle. On peut noter qu'il y a nécessairement une éclipse partielle de Lune avant et après chaque éclipse totale de Lune.

Vue de profil



Vue en coupe



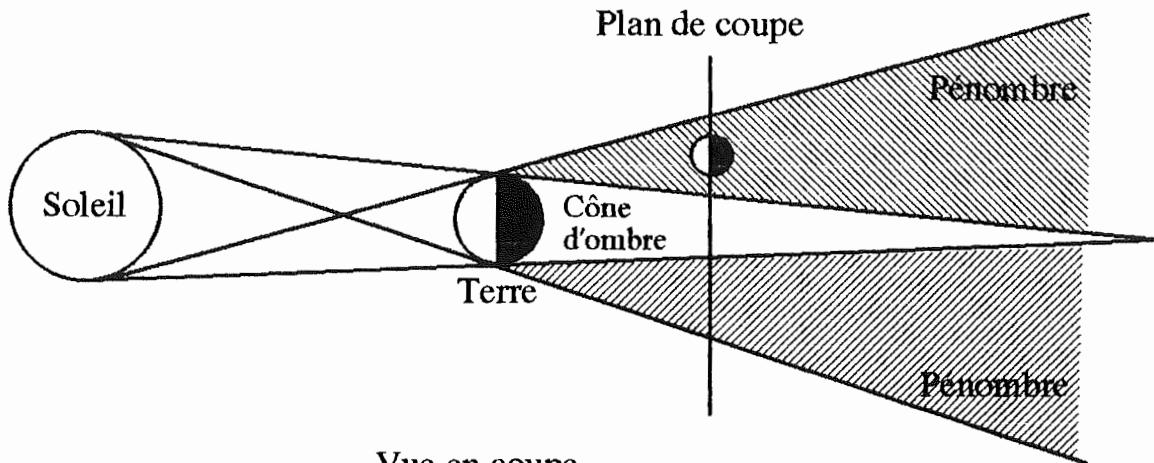
Eclipse partielle de Lune

Figure 2

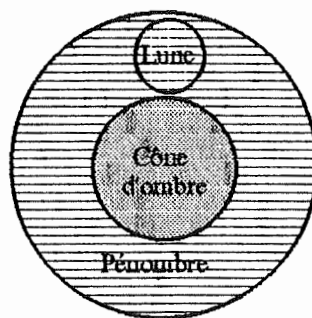
- Eclipse par la pénombre :

Si la Lune ne pénètre que dans la région de pénombre, le phénomène prend alors le nom d'éclipse par la pénombre. On peut noter également qu'il y a une éclipse par la pénombre, avant et après chaque éclipse partielle de Lune.

Vue de profil



Vue en coupe

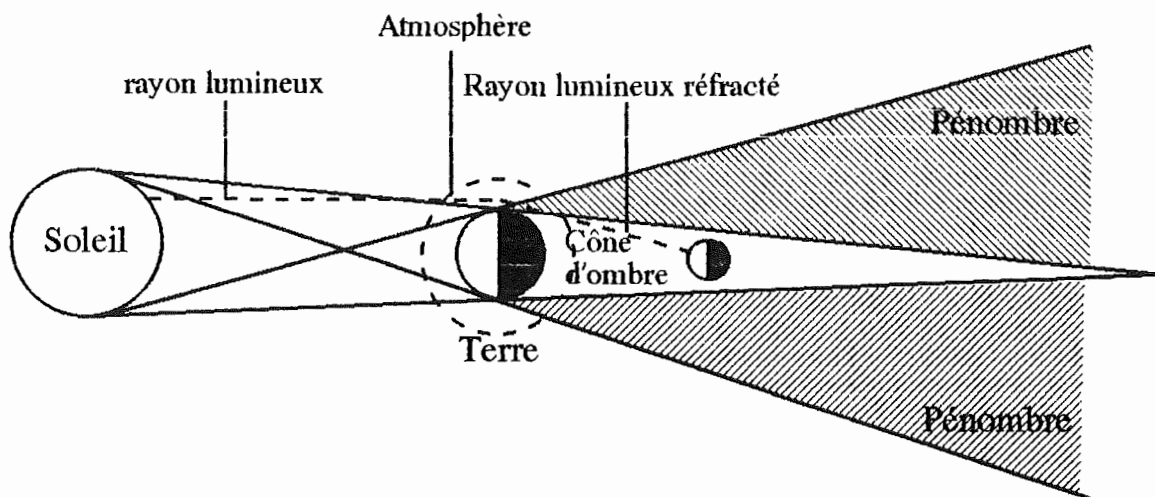


Eclipse par la pénombre

Figure 3

Il est important de souligner les deux caractéristiques suivantes d'une éclipse lunaire :

- Les éclipses de Lune sont visibles de tous points de la surface de la Terre pour lesquels la Lune se trouve au-dessus de l'horizon au moment où le phénomène se produit.
- La durée d'une éclipse totale de Lune peut atteindre un maximum de une heure quarante minutes. Pendant cette phase, la Lune ne disparaît pas complètement, car une certaine quantité de lumière solaire, réfractée par l'atmosphère terrestre, entre dans le cône d'ombre et vient frapper sur la surface lunaire. L'atmosphère terrestre absorbant principalement les rayons de petite longueur d'onde, les rayons qui parviennent à pénétrer dans le cône d'ombre et qui atteignent la Lune sont principalement ceux de grande longueur d'onde, de sorte que la Lune, pendant la phase de totalité, sera de couleur orangée.



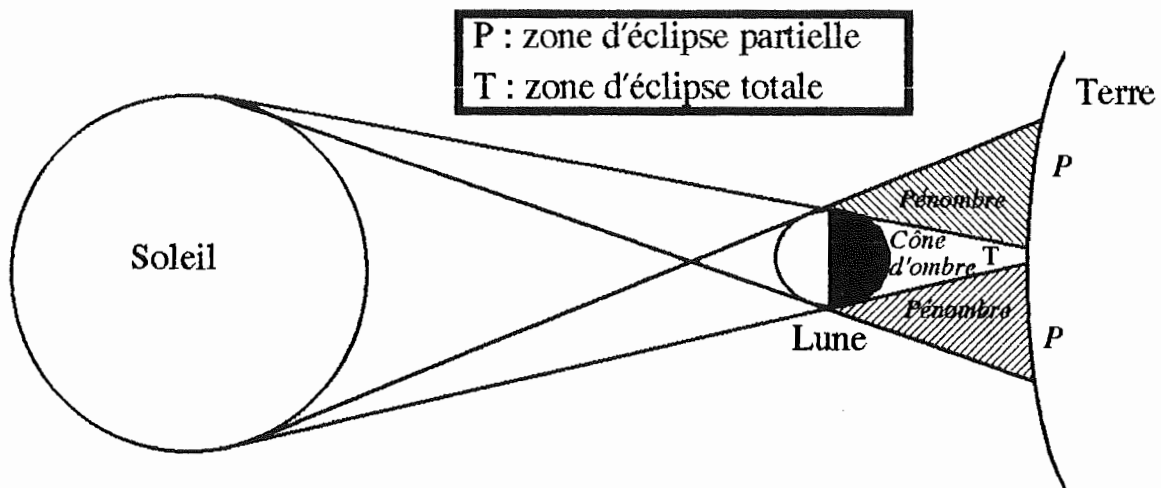
*Eclipse totale de Lune avec réfraction des rayons
du Soleil dans l'atmosphère*

Figure 4

2. Les éclipses de Soleil

Pour qu'une éclipse de Soleil puisse être observée sur Terre, il suffit que le Soleil soit occulté entièrement ou en partie par la Lune. Tel est le cas chaque fois que l'ombre ou la pénombre de la Lune atteint la Terre. On peut en outre observer que la Lune se trouve alors nécessairement dans sa phase dite de *Nouvelle Lune*. En effet, la phase de Nouvelle Lune a lieu chaque fois que le Soleil, la Lune et la Terre se trouvent tous trois dans un plan perpendiculaire au plan de l'écliptique alignés dans l'ordre précité (pour les phases de la Lune : voir référence bibliographique 1). Or, on observe une éclipse de Soleil chaque fois que le Soleil, la Lune et la Terre sont quasi alignés dans cet ordre, ce qui est une condition suffisante pour se trouver dans la phase de Nouvelle Lune.

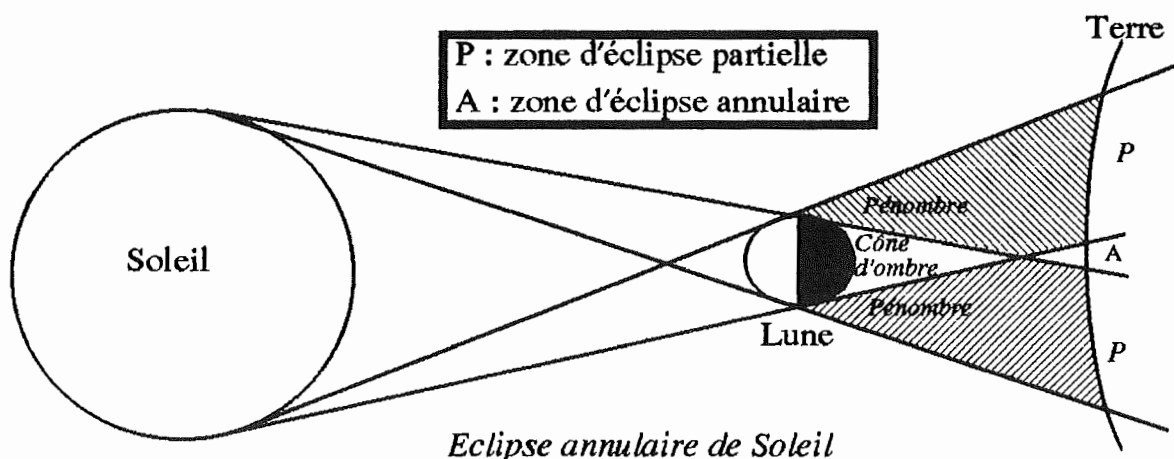
Différents types d'éclipses de Soleil sont observables selon les positions relatives de la Terre, de la Lune et du Soleil. Cependant, et contrairement aux éclipses de Lune, une éclipse de Soleil est un phénomène local. Ceci est dû au fait que le Soleil est un corps lumineux et qu'il n'y a donc une éclipse que pour certains points bien précis de l'espace : ceux dont la vue du Soleil est cachée par un autre corps. De plus, le diamètre de la section du cône d'ombre de la Lune au niveau de la Terre est de taille restreinte par rapport à celui de la Terre (< 270 Km pour un diamètre terrestre de plus ou moins 12700 km), par conséquent seule une petite partie de la Terre peut se trouver à un moment donné dans le cône d'ombre de la Lune, ce qui donne lieu à une éclipse totale locale. Quant aux régions voisines de la zone d'intersection de la surface de la Terre et du cône d'ombre, située dans la pénombre, elles subiront une éclipse de Soleil partielle comme on peut le voir sur la figure 5 qui suit.



Eclipse totale de Soleil

Figure 5

Par suite de l'excentricité de l'orbite lunaire (*1) et de la taille relative des corps en présence, il arrive que le cône d'ombre n'atteigne pas la Terre bien que les conditions pour avoir une éclipse de Soleil sont remplies. On parle alors d'éclipse Annulaire :



Eclipse annulaire de Soleil

Figure 6

Il faut enfin remarquer que seule une éclipse partielle se produira si la Terre au cours de son déplacement ne passe que dans la pénombre.

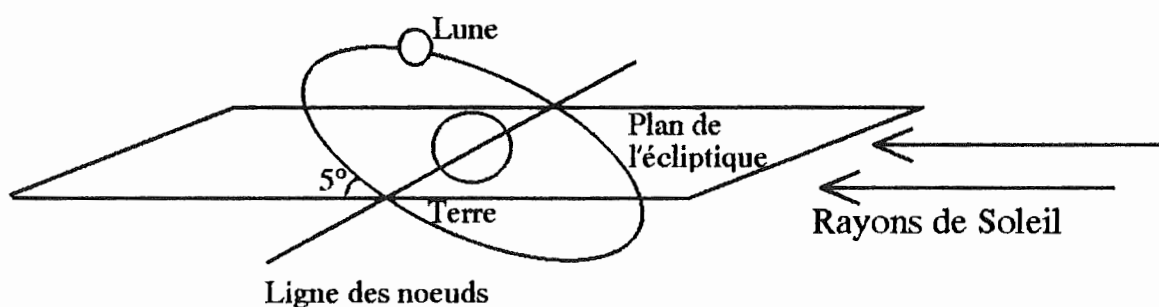
*1 La distance Lune-Terre varie par rapport à sa distance moyenne d'environ 10% (384400 Km (distance moyenne) - 356400 Km (distance au périégée), 384400 Km - 406700 Km (distance à l'apogée))

Notons encore une caractéristique des éclipses de Soleil :

- une éclipse totale de Soleil en un endroit de la Terre ne dure qu'au maximum sept minutes et demie. Il faut cependant remarquer que comme la Terre tourne sur elle-même, la projection du cône d'ombre se déplace également sur Terre mais en sens opposé et à la vitesse d'environ 1800 Km/h à proximité de l'équateur. On pourrait donc suivre une éclipse de Soleil pendant plus d'une heure, si on se déplaçait assez vite d'ouest en est avec un avion.

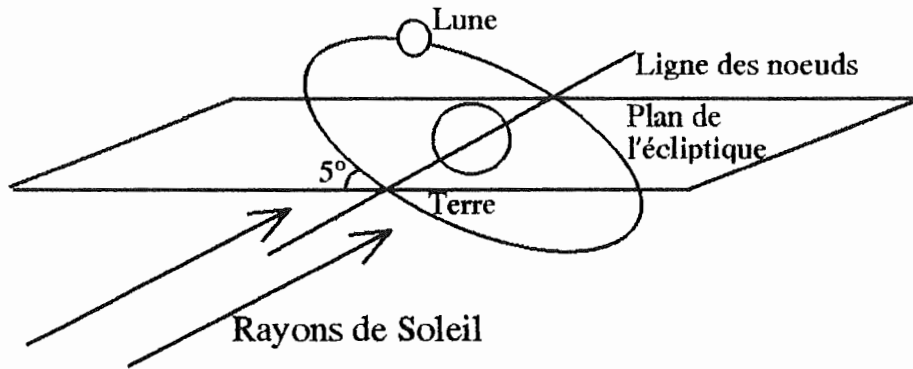
3. Périodicité des éclipses sur Terre

Si la Lune gravitait exactement dans le plan de l'orbite terrestre (plan de l'écliptique), il y aurait une éclipse de Soleil à chaque nouvelle Lune et une éclipse de Lune à chaque pleine Lune. Mais l'orbite lunaire est inclinée d'environ cinq degrés par rapport au plan de l'écliptique. Il n'existe donc que deux points d'intersections entre l'orbite de la Lune et le plan de l'écliptique. Ces points sont appelés noeuds. Pour avoir une éclipse, nous avons vu que le Soleil, la Terre et la Lune doivent être quasi alignés. Il est donc nécessaire pour avoir une éclipse que le Soleil se trouve suffisamment proche de la ligne des noeuds. Ce qui se produit environ deux fois par an pendant une période de 38 jours.



Orbite de la Lune : saison où une éclipse est impossible

Figure 7



Orbite de la Lune : saison où une éclipse est possible

Figure 8

Comme le phénomène des éclipses est produit par trois corps qui ont leur périodicité propre, on peut retrouver les mêmes conditions d'éclipse à intervalles réguliers, il suffit d'avoir un multiple de ces périodicités. Ce multiple est pour le système Terre, Lune, Soleil de 18 ans et 11 jours. Tous les 18 ans et 11 jours, on retrouve donc une configuration géométrique presque identique du système Terre-Lune-Soleil.

A intervalles de 18 ans et 11 jours, les éclipses de Soleil, comme de Lune, se reproduiront donc d'une façon presque identique (temps d'occultation égal, type identique, ...). Cette période est appelée cycle de Saros, elle était déjà connue 2000 ans avant JC.

A titre indicatif, il se produit au moins deux éclipses par an et il s'agit d'éclipses de Soleil. Le nombre maximal d'éclipses par an est de 7 dont 4 de Soleil et 3 de Lune, ou 5 de Soleil et 2 de Lune.

4. Justification de l'intérêt scientifique de l'observation des éclipses

L'observation des éclipses a joué un rôle de premier plan dans le progrès des connaissances astronomiques depuis l'antiquité jusqu'à nos jours.

On a, par exemple, découvert en comparant les instants auxquels avaient eu lieu certaines éclipses dans l'antiquité avec les instants auxquels elles auraient du avoir lieu si les conditions étaient restées les mêmes jusqu'aujourd'hui, que la vitesse de rotation de la Terre diminuait. En effet les éclipses de l'antiquité se produisaient systématiquement avec une avance de quelques heures par rapport aux résultats des calculs.

On a aussi pu grâce aux éclipses des satellites de Jupiter, constater que la vitesse de la lumière était une grandeur finie, et l'évaluer.

Comme on l'a vu précédemment, lors d'une éclipse totale de Lune, certains rayons réfractés arrivent sur la surface de la Lune. La couleur de ces rayons a permis une première étude de la composition de l'atmosphère terrestre.

Bien que tous ces sujets soient fort intéressants, et qu'ils justifient à eux seuls une étude des éclipses par les profanes, il ne s'agit pas du propos de ce mémoire et nous ne nous étendrons donc pas plus sur le sujet.

Le simulateur

1. Le projet

Le phénomène des éclipses est difficile à appréhender car il est produit par le mouvement de trois corps dans l'espace, mouvement souvent difficile de se représenter correctement. C'est pourquoi il nous a semblé intéressant de programmer un simulateur répondant aux critères suivants :

- représenter à l'écran le mouvement de trois corps sphériques quelconques c_1 , c_2 , c_3 dans l'espace suivant différents angles de vue.
- considérer qu'un de ces trois corps est lumineux et, par conséquent, visualiser, suivant l'angle de vue, les zones éclairées et non éclairées des deux autres corps (nuit et jour sur les corps).
- prévoir l'intégration facile de ce simulateur dans un programme didactique sur l'astronomie.

Dans les paragraphes suivants, nous allons expliquer comment nous avons modélisé le problème du mouvement des trois corps et comment nous avons ramené ce problème en deux dimensions au moyen des projections en perspective. Il nous restera ensuite à décrire la manière dont nous avons élaboré le programme.

2. Modélisation du problème en trois dimensions

Dans ce paragraphe, nous allons préciser comment les corps sont définis et comment on peut caractériser leur mouvement.

Pour modéliser le problème, on va faire l'hypothèse que le mouvement de chacun des corps est un mouvement circulaire uniforme autour d'un point qui sera le centre de l'orbite. Ce centre pourra soit être fixe par rapport à l'observateur (voir le paragraphe sur les projections en perspective), soit dépendant d'un autre corps. On peut alors modéliser le problème de la manière suivante.

Pour simplifier le problème du calcul du mouvement, on associe à l'orbite de chacun des trois corps un repère orthonormé. Chaque repère est caractérisé par son origine, qui est le centre de l'orbite du corps associé, et par le plan que forme les axes X et Y , qui est le plan de l'orbite du corps associé. Chaque corps tourne donc dans le plan XY autour de l'axe Z .

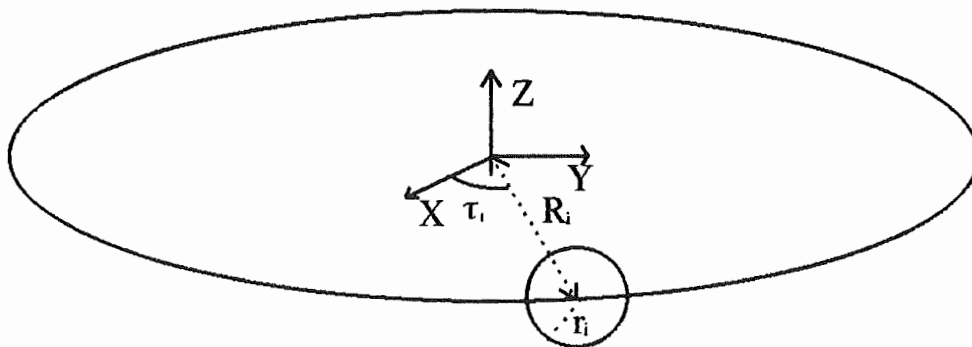
Comme le centre de l'orbite d'un corps peut être soit fixe, soit dépendant,

chacun des repères sera soit fixe par rapport à l'observateur, soit associé à un autre corps. Le repère du corps A est dit associé au corps B, si l'origine du repère du corps A est le centre du corps B. Donc, si le corps B se déplace, le repère du corps A et donc son orbite, se déplacent de la même façon. On peut, par exemple, caractériser le problème de la Terre et de la Lune de la manière suivante : la Lune serait le corps A et la Terre le corps B.

Il est à noter qu'il existe une multitude de repères pour chacun des corps. En effet, aucune précision n'a été faite jusqu'à présent sur la position des axes X, Y et Z, si ce n'est que XY est le plan de l'orbite, et que le vecteur de rotation est dans le prolongement de l'axe Z. Dès lors, il est nécessaire d'imposer d'autres contraintes à ce repère :

- Il doit être direct
- Le mouvement du corps se fait dans le sens des aiguilles d'une montre en regardant dans le sens de l'axe Z.
- L'angle τ_i qui détermine la position d'un corps c_i sur son orbite sera compté positivement à partir de l'axe X vers l'axe Y.

L'orbite de chaque corps c_i est alors définie par le repère qui lui est associé et par son rayon R_i qui définit la distance entre l'origine du repère et le centre du corps. Le volume de chaque corps peut être calculé grâce à son rayon r_i .



Modélisation pour le corps c_i

Figure 9

Pour que la modélisation du mouvement soit complète, il ne reste plus maintenant qu'à préciser la vitesse de déplacement du corps c_i . Cette vitesse sera connue grâce au temps de révolution (*1) An_i .

*1 Il s'agit de la révolution sidérale

Comme nous voulons permettre de visualiser le phénomène à l'écran selon différents angles de vue, il est nécessaire que l'utilisateur précise sous quel angle de vue il désire observer le phénomène. Il faudra donc notamment savoir d'où on regarde le phénomène (position de l'observateur). Cette position doit être rattachée à la modélisation existante, et doit donc être donnée dans au moins un des repères associés aux corps. Par convention, nous choisirons le repère du corps c1, on l'appellera le repère de référence. La position de l'observateur sera fixe par rapport à ce repère. Les autres repères seront donnés par la matrice de changement de repère qui permet d'exprimer les coordonnées de ces repères dans le repère de référence.

exemple : Modélisation du système Terre, Lune, Soleil.

Si on considère que le Soleil est le corps de référence, nous avons que :

$$R1 = 0 \text{ Km}$$

$$R2 = 149600000 \text{ Km}$$

$$R3 = 384400 \text{ Km}$$

$$r1 = 700000 \text{ Km}$$

$$r2 = 6372 \text{ Km}$$

$$r3 = 1738 \text{ Km}$$

et la matrice de changement de repère associée à l'orbite de la Terre est :

$$\begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{vmatrix}$$

et l'origine du repère associé à l'orbite de la Terre est donné par le vecteur (0, 0, 0) (coordonnées cartésiennes dans le repère de référence).

Les deux repères peuvent être identiques puisque le Soleil est au centre de l'orbite de la Terre et qu'il est fixe.

La matrice associée à l'orbite de la Lune est quant à elle:

$$\begin{vmatrix} 1 & 0 & 0 \\ 0 & 0.9961 & -0.0871 \\ 0 & 0.0871 & 0.9961 \end{vmatrix}$$

et l'origine du repère associé à l'orbite de la Lune est donné par le vecteur (Tx, Ty, Tz) puisque la Lune peut être associée à la Terre

(origine dépend de la Terre (T_x, T_y, T_z)) et que son orbite est inclinée de 5 degrés par rapport au plan de l'écliptique (plan de l'orbite terrestre).

3. Représentation du phénomène à l'écran : projections en perspective

Lorsque l'on veut représenter sur un écran ou sur une feuille, c'est à dire sur une surface plane, des objets qui se déplacent dans l'espace, il est nécessaire de transformer la représentation tridimensionnelle des objets en une représentation bidimensionnelle. Ce passage de l'espace au plan peut se faire grâce aux projections. Comme le phénomène des éclipses se produit dans l'espace, et que nous voulons le visualiser sur un écran, nous utiliserons des projections.

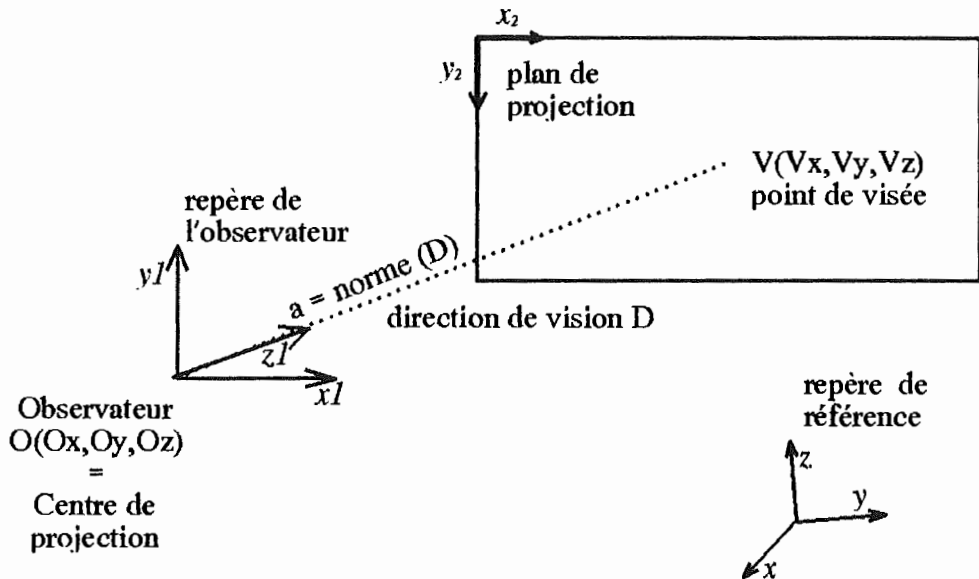
Voici un bref rappel des principes des projections que nous utiliserons dans le programme : les projections en perspective.

Une projection en perspective est définie par un plan appelé plan de projection et par un point appelé centre de projection. La projection d'un objet est définie par la projection de chacun de ses points. La projection d'un point est définie par l'intersection du plan de projection et de la droite qui relie le point à projeter au centre de projection (appelée rayon de projection).

La projection en perspective, définie par le plan de projection et le centre de projection, que nous utiliserons pour représenter à l'écran le mouvement des trois corps sera caractérisée par les paramètres suivants :

- (O_x, O_y, O_z) : coordonnées de l'observateur dans le repère de référence. Elles définissent le centre de projection. Il s'agit du point d'où on regarde le phénomène.
- (V_x, V_y, V_z) : coordonnées du point de visée dans le repère de référence. Le point de visée est défini de manière à regrouper deux notions de la projection en perspective. D'une part, il permet de définir la direction de vision D (regard de l'observateur), et d'autre part il permet de situer le plan de projection par rapport à l'observateur. En effet, le point de visée est défini comme l'intersection de la direction de vision et du plan de projection. Ce plan de projection est perpendiculaire à la direction de vision pour conserver la vue que l'observateur a de l'écran.

Le vecteur D de la direction de vision qui nous servira à calculer la matrice de changement de repère, est donc donné par $O-V$, tandis que le plan de projection est à la distance $a = \text{norme}(D)$ de l'observateur.



Projection en perspective

Figure 10

Après avoir défini, la projection en perspective que nous utiliserons, nous allons maintenant expliquer comment, à partir des coordonnées d'un point de l'espace dans le repère de référence, nous obtenons les coordonnées de ce point à l'écran.

Comme nous l'avons vu dans la modélisation du problème, le repère du corps $c1$ est le repère de référence. Pour visualiser le phénomène depuis la position de l'observateur donnée par les coordonnées (O_x, O_y, O_z) dans le repère de référence, nous procéderons à un changement de repère avant de faire la projection en perspective proprement dite. Il est à noter que le repère que nous recherchons devra avoir ses axes x_1 (transformation de l'axe y du repère de référence) et y_1 (transformation de l'axe Z du repère de référence) respectivement dans la direction de l'axe x_2 et y_2 de l'écran.

Ce changement de repère peut être réalisé grâce à la matrice homogène M , qui est la composition d'une translation (du centre du repère de référence vers la position de l'observateur) et de deux rotations (une autour de l'axe Z (φ) et une autour de l'axe Y (θ)) (voir annexe pour la détermination de cette matrice) :

$$M = \begin{vmatrix} -\sin\varphi & \cos\varphi & 0 & \sin\varphi.Ox - \cos\varphi.Oy \\ -\sin\theta.\cos\varphi & -\sin\theta.\sin\varphi & \cos\theta & \sin\theta.\cos\varphi.Ox + \sin\theta.\sin\varphi.Oy - \cos\theta.Oz \\ \cos\theta.\cos\varphi & \cos\theta.\sin\varphi & \sin\theta & -\cos\theta.\cos\varphi.Ox - \cos\theta.\sin\varphi.Oy - \sin\theta.Oz \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

Les sinus et cosinus des angles φ et θ peuvent être obtenus directement à partir du vecteur qui indique la direction de vision D , sa projection D_{proj} sur le plan $Z=0$ et leurs normes :

$$\begin{aligned} \cos\varphi &= D_x / |D_{proj}| \\ \sin\varphi &= D_y / |D_{proj}| \\ \cos\theta &= -|D_{proj}| / |D| \\ \sin\theta &= -D_z / |D| \end{aligned}$$

En multipliant par la matrice M les coordonnées homogènes du point à projeter données dans le repère de référence, on obtient les coordonnées de ce même point, mais dans le repère observateur.

Une fois cette multiplication effectuée, il reste encore à trouver les coordonnées homogènes de ce point après la projection en perspective, c'est à dire les coordonnées du point projeté sur le plan de projection. On obtient ce résultat en multipliant les coordonnées par la matrice :

$$\begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/a & 0 \end{vmatrix}$$

où a est la distance entre l'observateur et le plan de projection.

Il est à noter que dans le programme, l'ensemble de ces calculs se fera en coordonnées cartésiennes, et donc que les matrices données ici seront décomposées en une matrice 3x3 et un vecteur à trois dimensions (dans la suite du programme, l'ensemble matrice-vecteur sera appelé matrice). Elles ont été données ici en coordonnées homogènes, pour simplifier l'écriture.

Jusqu'à présent, nous avons décrit la projection sur un plan (le plan de l'écran par exemple). Cependant, comme nous voulons réaliser une simulation sur un ordinateur, la projection devra se faire sur un écran qui est limité. Comme le rapport d'aspect de cet écran (longueur de l'écran sur largeur de l'écran) est connu, et que l'on veut une vue rectangulaire, il est nécessaire pour décrire la projection de

manière appropriée à notre projet que nous introduisons un paramètre supplémentaire : γ . Cet angle permet de définir la hauteur de l'écran (partie de l'espace considérée), et par conséquent sa largeur connaissant le rapport d'aspect.

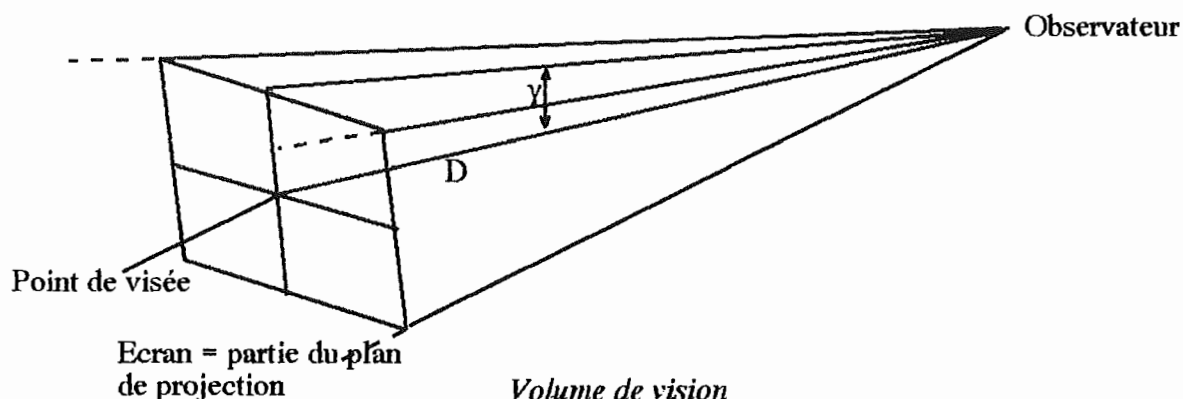


Figure 11

4. Réalisation du programme

Après avoir décrit la modélisation du mouvement des trois corps, il nous reste à commenter la réalisation du programme. Nous avons décomposé l'analyse du programme en quatre parties :

- Les structures de données
- La structure du programme
- La réalisation du module interface utilisateur
- La réalisation du module application

4.1 Les structures de données

Avant d'analyser le programme proprement dit, décrivons les structures de données nécessaires étant donné notre modélisation du problème du mouvement des trois corps.

Comme nous avons programmé en Turbo Pascal, toutes ces structures seront définies en Pascal.

Avant d'aborder les structures du problème, il est utile de définir deux structures plus générales qui nous serviront par la suite : la structure de matrice et la structure de vecteurs. Elles sont représentées en Pascal par des tableaux.

Type *matrice* = array [1..3,1..3] of real;
 vecteur = array [1..3] of longint;

Les structures suivantes ont été jugées nécessaires dans ce programme :

- les trois corps
- la projection
- le temps
- la perspective
- le point

La structure de données des trois corps découle de la modélisation décrite au point 2. Elle contient le rayon orbital (**R**), le rayon du corps (**rayon**), l'angle de sa position par rapport à l'axe X (**tau**), sa dépendance vis à vis d'un autre corps (**dependant**), le numéro du corps dont il dépend (**corpdep**) (on ne tient compte de cette variable que si **dependant** est vrai), la matrice de changement de repère (repère du corps, vers repère du corps c1) (**reper** et **origine**), le temps (*1) en une unité arbitraire qui sera appelée "jours" pour effectuer une rotation complète (**an**). Nous avons complété la structure des trois corps par un paramètre ne dépendant pas de la modélisation : la couleur du corps (**couleur**). Les données des trois corps seront stockées dans un tableau à trois dimensions qui sera du type *array[1..3] of corp*.

```
corp = record  R : integer;  
               rayon : integer;  
               tau : integer;  
               dependant : boolean;  
               corpdep : byte;  
               reper : matrice;  
               origine : vecteur;  
               an:integer;  
               couleur:byte;  
  
               end;
```

*1 La variable temps sert à calculer les vitesses relatives de chaque corps sur leur orbite. La vitesse réelle de chaque corps à l'écran dépend du type de l'ordinateur et du paramètre temps. Ce paramètre peut être diminué ou augmenté, il changera de cette manière l'écart entre deux positions successives des corps sur leur orbite. Il est à noter qu'une trop grande ou trop petite valeur pour une modélisation donnée ferait qu'on ne distingue plus le mouvement à l'écran.

La structure de données définissant la projection en perspective peut être décrite par trois éléments seulement : la position de l'observateur (*o*), la position du point de visée (*v*) et enfin l'angle qui définit la portion visible du plan de projection (*gamma*).

```
projection = record o:vecteur;  
                 v:vecteur;  
                 gamma:integer;  
            end;
```

La structure *temps* est composée de deux paramètres : *t* qui a pour rôle de déterminer le temps écoulé pour la simulation entre deux affichages successifs, et *inc* qui permet d'augmenter ou de diminuer *t*.

```
temps = record t:real;  
              inc:real;  
            end;
```

Pour calculer le changement de repère associé à la perspective (ou à l'observateur), il faut multiplier chaque point de l'espace par la matrice définie au point trois de ce chapitre. D'où l'utilité de la structure de données qui contient les paramètres suivants : $\cos\theta$ (*costheta*), $\sin\theta$ (*sintheta*), $\cos\varphi$ (*cosphi*), $\sin\varphi$ (*sinphi*), le vecteur de translation (*o*), ainsi que le paramètre définissant la portion de l'espace (*gamma*).

```
perspective = record costheta:real;  
                   sintheta:real;  
                   cosphi:real;  
                   sinphi:real;  
                   gamma:integer;  
                   o:vecteur;  
            end;
```

La structure de données qui définit un point de l'espace en coordonnées cartésiennes est utilisée pour décrire l'orbite des trois corps. Cette structure étant donné l'usage qu'on en fait sera donc un tableau du type *array[1..3,0..35] of point*, puisqu'une orbite est définie par l'ensemble de ses points.

```
point = record x:real;  
              y:real;  
              z:real;  
            end;
```

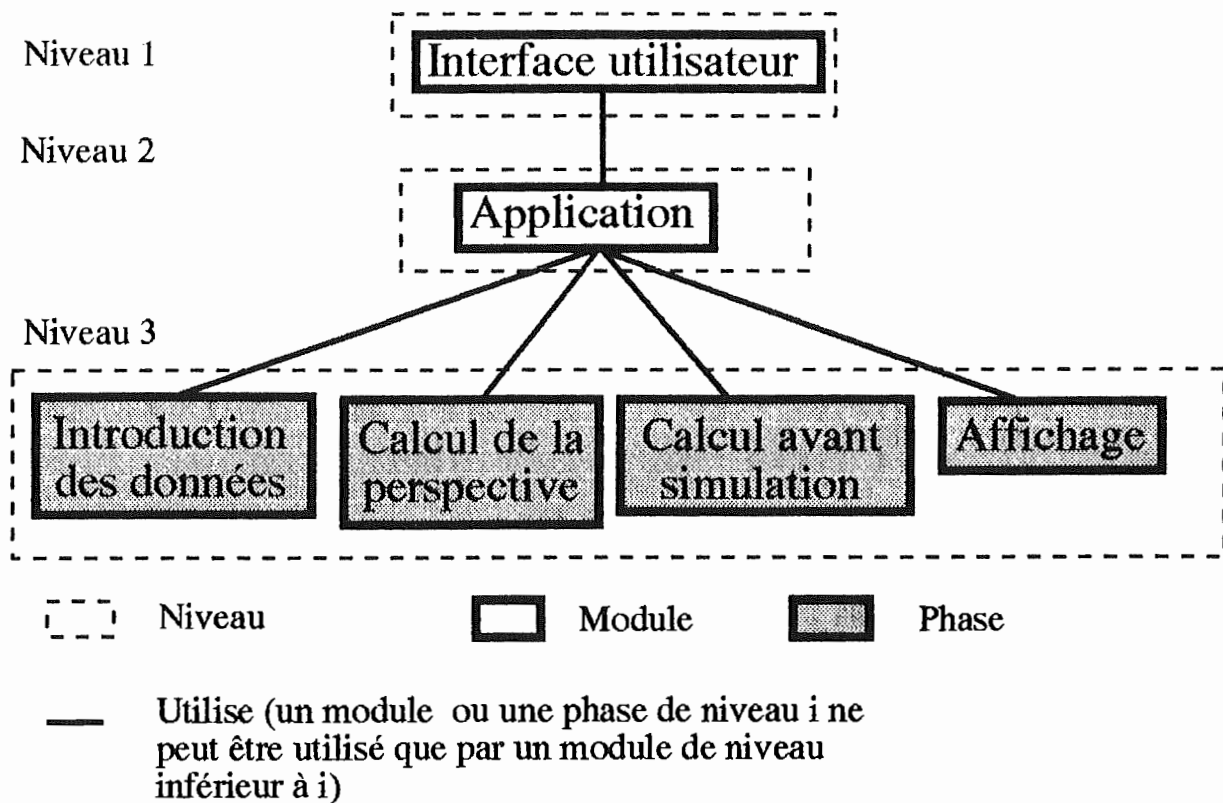
4.2 La structure du programme

Afin de faciliter la compréhension et la maintenance du logiciel, le programme a été découpé en modules répartis sur différents niveaux.

Les services offerts par les différents modules sont les suivants :

- Le module interface utilisateur doit permettre à un utilisateur d'introduire les paramètres de la simulation. Ce module n'a été réalisé que pour permettre l'utilisation du module application actuel.
- Le module application est le coordinateur de la simulation. Il est chargé d'utiliser successivement l'ensemble des phases de niveau 3 qui réalisent la simulation. Ces appels ont été regroupés dans ce module car on veut que l'application soit indépendante du module d'interface. Ce qui nous permet d'atteindre le troisième objectif de notre projet : à savoir permettre l'intégration facile de la simulation des éclipses dans un logiciel plus général sur l'astronomie. En effet, seul le module interface devra être supprimé et réécrit pour réaliser ce programme général.
- Les quatre phases qui composent le niveau 3 gèrent la simulation. La première, la phase d'introduction des paramètres, permet au module application de convertir les données fournies par le module interface utilisateur dans ses propres structures. La deuxième, la phase du calcul de la perspective, calcule les paramètres de la matrice de changement de repère du corps c1 vers le repère de l'observateur. La troisième, la phase du calcul avant simulation, devra calculer l'ensemble des variables qui ne changent pas en cours de simulation (pour le moment ce module calcule certaines positions des orbites). La dernière, la phase d'affichage, doit calculer la position des corps, des cônes d'ombres, ... et les afficher.

Voici une représentation schématique de la structure du programme :



Structure du programme

Figure 12

4.3 Réalisation de l'interface utilisateur

L'interface a été développée en respectant les critères suivants :

- permettre un usage aisé de la simulation
- rester indépendante de l'application : elle ne doit pas faire corps avec elle, afin d'être facilement supprimée et remplacée.

Nous avons donc conçu cette interface comme un module qui ne fait que fournir au module application une liste de données que ce module devra par la suite interpréter et sauver dans ses propres structures de données.

L'interface utilisateur que nous avons développée se décompose en deux parties. La première permet l'introduction de données de manière classique : une donnée est entrée par l'intermédiaire du clavier. La deuxième partie est visuelle, elle est activée à la demande de l'utilisateur et lui permet de modifier sur un graphique les coordonnées tridimensionnelles de la position de l'observateur et du point de visée.

4.3.1 Réalisation de l'interface traditionnelle

Cette interface a été réalisée grâce à une librairie existante dans le Turbo Pascal 6.0 : Turbo Vision. Cette librairie gère automatiquement l'utilisation de la souris et du clavier. Elle impose que toutes les données d'un même écran soient introduites dans un même record. Elle est de conception orientée objet, ce qui permet une intégration facile de nos propres méthodes. En effet, il nous a suffi de préciser les dimensions de la fenêtre, les zones d'introduction et leurs libellés, et les méthodes de validation de l'entrée des données pour décrire l'interface.

exemple : Comment programmer l'introduction du rayon de l'orbite des corps.

- On fournit au programme le libellé de la zone "rayon de l'orbite", et la position de ce libellé dans la fenêtre.
- On donne la dimension et la position de la zone d'introduction correspondante.
- On signale au programme qu'il s'agit d'une zone d'introduction d'entier. Or lui ne gère que des entrées de chaînes de caractères. Nous avons donc du réaliser une méthode qui restreint ces caractères au domaine des chiffres, une qui valide ces entiers, une qui convertit un entier en une chaîne de caractères, et une dernière qui permet de convertir une chaîne de caractères en un entier.

4.3.2 Réalisation de l'interface visuelle

Le but de l'interface visuelle est de représenter l'observateur o_1 , le point de visée v_1 et le repère de référence de la simulation simultanément à l'écran. Les coordonnées des points o_1 et v_1 sont données dans l'espace par l'interface traditionnelle. On en calcule la projection sur l'écran à l'aide d'une projection standard, dont on a initialisé les paramètres o_2 , v_2 , et γ_2 . On vérifie que la projection des points o_1 , v_1 se trouve sur l'écran (On ne modifie pas v_2 , car on veut garder le centre du repère de référence au milieu de l'écran). Tant que ce n'est pas le cas, on modifie les paramètres o_2 , γ_2 de la projection standard par raffinement successif jusqu'à ce que les points o_1 et v_1 de la simulation se trouvent sur l'écran. Dès que cette condition est remplie, on affiche le résultat de la projection à l'écran et on réagit en fonction des demandes de l'utilisateur. A chaque demande de l'utilisateur pour modifier la position du point o_1 ou du point v_1 , on vérifie s'il n'est pas nécessaire de réajuster la projection en perspective comme expliqué ci-dessus.

Cette interface utilise deux phases du module application qui permettent d'initialiser les structures du programme et de calculer les paramètres du changement de repère associé à une projection en perspective.

4.4 Réalisation de l'application

Dans la structure du programme définie actuellement, le module application reçoit de l'interface les valeurs des paramètres de la simulation avec lesquelles il va initialiser ses propres structures de données. Il appelle ensuite les différentes phases qui permettent d'effectuer la simulation :

1. calcul des paramètres du changement de repère associé à la projection en perspective
2. calcul de l'orbite de chaque corps dans le repère de l'observateur (repère associé à la projection en perspective)
3. calcul et affichage du mouvement des trois corps

L'introduction des paramètres dans les structures de données du programme ne sera pas détaillée plus. Il ne s'agit en effet que d'une suite d'assignation.

4.4.1 Calcul des paramètres de la matrice de changement de repère

Pour pouvoir calculer la position des différents corps à l'écran, il est nécessaire, comme nous l'avons vu précédemment, de réaliser une projection en perspective, définie par l'utilisateur au moyen des paramètres α , v , γ . Pour ce faire, il faut d'abord effectuer un changement de repère (repère de référence vers repère de l'observateur), avant de réaliser la projection proprement dite.

Les paramètres qui conditionnent ce changement de repère ne sont pas donnés de manière directement utilisable. En effet, nous connaissons α , v et γ , et nous devons utiliser $\cos\theta$, $\sin\theta$, $\cos\varphi$, et $\sin\varphi$. Nous avons donc jugé qu'il était utile de regrouper ces calculs de paramètres dans une phase.

Les paramètres à calculer pour le changement de repère sont liés aux paramètres fournis par l'interface de la manière suivante (voir paragraphe consacré aux projections en perspective) :

$$\begin{aligned}\cos\varphi &= D_x / |D_{proj}| \\ \sin\varphi &= D_y / |D_{proj}| \\ \cos\theta &= -|D_{proj}| / |D| \\ \sin\theta &= -D_z / |D|\end{aligned}$$

avec $D = o1-v1$.

4.4.2 Calcul de l'orbite des corps dans le repère observateur

Afin d'augmenter la qualité du mouvement de la simulation même sur de petites machines, on va effectuer tous les calculs des variables qui ne se modifient pas pendant la simulation avant de commencer la simulation. C'est ainsi qu'une partie des calculs des positions successives sur l'orbite de chacun des corps dans le repère de l'observateur pourra être effectuée avant la simulation.

Ces calculs s'effectuent pour chaque corps en trois temps :

- Calcul de la matrice de changement de repère du corps c_i vers le repère de l'observateur à partir de la matrice de changement de repère du corps c_i vers le corps $c1$ et de la matrice de changement de repère du corps $c1$ vers le repère observateur.
- Calcul des positions de l'orbite dans le repère associé au corps c_i sur lequel on travaille. On calcule une position sur l'orbite tous les dix degrés, car nous avons jugé inutile, vu la faible précision de l'écran, de connaître plus de valeurs pour tracer une ellipse. Une simple interpolation linéaire entre deux points consécutifs suffit pour afficher de manière appropriée l'ellipse représentant la projection de l'orbite.

Pour chacun des corps dont le repère est fixe, on calculera les coordonnées de la position de l'orbite dans le repère observateur pour les angles 0, 10, 20, ..., 350 degrés (les angles sont comptés depuis l'axe X vers l'axe Y) soit 36 positions ou encore 108 valeurs. Pour les corps dont le repère est dépendant d'un autre corps, ces 36 positions seront calculées comme si le repère du corps était fixe. L'origine du repère étant mise à la position donnée par un angle de 0 degré pour le corps dont il dépend.

- Calcul du changement de repère : repère associé au corps c_i vers repère associé à l'observateur.

Le résultat de ces calculs sera stocké dans une table. Cette table est définie dans le programme comme un tableau (**mouv**) à deux dimensions : l'indice des

lignes correspond au numéro des corps et l'indice des colonnes correspond à l'angle déterminant la position sur l'orbite divisé par dix. Le type de ce tableau est le type *array[1..3,0..35] of point*. Le type *point* est défini dans les structures de données au point 4.1.2.

Exemple :

- repère fixe : *mouv[1 , 2].x* correspond à la coordonnée x de l'orbite du corps c1 pour un angle de 20 degrés dans le repère de l'observateur.
- repère dépendant : *mouv[2 , 20].y* correspond à la coordonnée y de l'orbite du corps c2 pour un angle de 200 degrés dans le repère de l'observateur comme si le corps dont il dépend était fixe à la position donnée par un angle de 0 degré.

4.4.3 Calcul et affichage de la position des corps au temps t

Cette phase est découpée en deux parties. Dans la première partie (procédure générale d'affichage), on calcule la position des corps à l'écran et on affiche le corps lumineux. Dans la seconde partie (procédure particulière d'affichage), on affiche les corps non lumineux, on calcule et on affiche les cônes d'ombres produits par le corps lumineux et le corps non lumineux.

4.4.3.1 Réalisation de la procédure générale d'affichage

Avant d'expliquer l'enchaînement des étapes de ce module, il faut souligner que ce programme ne peut fournir un affichage convenable qu'avec une carte de type EGA ou VGA, et que de plus le mode utilisé ne peut être décidé par l'utilisateur. Cela est dû à deux phénomènes. Tout d'abord, sans les couleurs ou une carte les interprétant il est difficile de distinguer les zones d'ombre, ainsi que les trois corps. Ensuite, et c'est là une raison encore plus importante, seule ces deux cartes permettent de travailler sur plusieurs pages à la fois en mode graphique : on peut travailler sur une page, pendant que l'autre est affichée. Cela permet de faire des animations sans scintillement de l'écran.

La réalisation de la procédure générale d'affichage comporte les étapes suivantes :

1. initialisation du système (page activée, ratio de l'écran, ...)

2. incrémentation du temps dans la simulation (temps := temps + ut.t avec temps du type *real* et ut du type *temps*, ut.t contient la valeur introduite par l'utilisateur pour déterminer le temps écoulé dans la simulation entre deux affichages successifs)
3. calcul pour chaque corps de l'angle qui permet de connaître sa position sur l'orbite (angle := (temps*360/corps[i].an) mod 360)
4. calcul des coordonnées de la position de chaque corps sur l'orbite dans le repère de l'observateur au temps **temps** grâce à une interpolation linéaire entre les deux points les plus proches de sa position se trouvant dans la table **mouv** à partir de l'angle calculé au point 3 (calcul en trois dimensions).

exemple : si le corps **c2** se trouve sur son orbite à un angle de 15 degrés, il faudra effectuer une interpolation entre les deux points calculés précédemment c'est à dire **mouv[2,1]** et **mouv[2,2]**.

5. calcul de la projection du centre de chacun des trois corps, ainsi que de leur rayon (i.e. multiplier les coordonnées x et y du corps par une fonction de z (voir le paragraphe sur les projections en perspective)) (calcul en deux dimensions).
6. tri des corps suivant leur éloignement de l'observateur.
7. effacement de l'écran précédent.
8. affichage des trois corps selon leur éloignement par rapport à l'observateur (D'abord le plus éloigné). L'affichage dans l'ordre d'éloignement permet d'avoir une vue réelle des corps sans devoir faire de nombreux calculs, ni devoir sauver de nombreuses valeurs. En effet un corps qui se trouverait derrière un autre et qui serait donc invisible pour l'observateur, ne sera pas visible à l'écran puisque le corps se trouvant devant sera affiché au-dessus de lui.

Si le corps est lumineux, l'affichage peut se faire tout de suite, sans calcul supplémentaire. On affiche donc le corps lumineux dans la couleur désirée par l'utilisateur, à la position calculée. Si par contre il s'agit d'un corps non lumineux, l'affichage ne peut pas se faire directement. En effet, à ce moment du programme, les cônes d'ombres

de chaque corps n'ont pas encore été calculés. C'est pourquoi, on va passer la main à la procédure qui se charge de calculer et d'afficher les corps non lumineux et les cônes d'ombres. La description de l'algorithme de calcul et d'affichage sera vue plus loin (point 4.4.3.2).

9. si aucune touche n'a été frappée par l'utilisateur, retour en 2, sinon poursuivre.

10. gestion des demandes de l'utilisateur.

Cette étape reprend l'ensemble des instructions qui peuvent être utilisées au cours de la simulation. La réalisation de ce genre de problème est assez classique, il s'agit d'un enchaînement de conditions qui, si elles sont réalisées, produisent l'effet désiré. Voici la structure que cela donne :

Si l'utilisateur a frappé sur la touche :

w (zoom avant) : alors changer l'angle de vision si c'est encore possible ($\text{gamma} := \text{gamma} \div 2$) > 0

x (zoom arrière) : alors changer l'angle de vision si c'est encore possible ($\text{gamma} := \text{gamma} * 2$) < 90

c : alors changer l'état de la variable qui permet de savoir si on dessine le cône d'ombre ou non

p (pause) : arrêt temporaire de la simulation et affichage des différentes orbites. L'affichage des orbites se fait grâce à la table de mouvements. En effet, il suffit de reprendre toutes les positions de l'orbite d'un corps, de faire la projection et de tracer une ligne entre les différentes positions ainsi calculées.

Flèches vers le haut : accélération de la vitesse de la simulation. On augmente la variable d'incrémentations de la valeur choisie par l'utilisateur dans l'interface.

Flèches vers le bas : diminution de la vitesse de la simulation. On diminue la variable d'incrémentations de la valeur choisie par l'utilisateur dans l'interface.

Shift Flèches vers le haut : accélération importante de la vitesse de la simulation. On augmente la variable d'incrémentation de dix fois la valeur choisie par l'utilisateur dans l'interface.

Shift Flèches vers le bas : diminution importante de la vitesse de la simulation. On diminue la variable d'incrémentation de dix fois la valeur choisie par l'utilisateur dans l'interface.

q : permet à l'utilisateur de quitter la simulation

11. si l'utilisateur ne désire pas quitter la simulation retour en 2.

4.4.3.2 Réalisation de la procédure particulière d'affichage

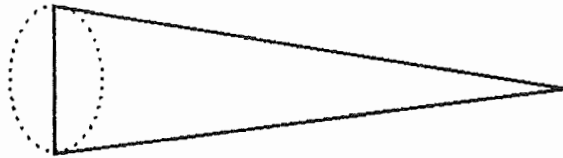
Rappelons que cette procédure sert à afficher les corps non lumineux et les cônes d'ombres. Elle est décomposée en cinq grandes étapes :

- Calcul du cône d'ombre
- Affichage du cône d'ombre
- Affichage du corps
- Calcul de la zone d'ombre sur le corps
- Affichage de la zone d'ombre sur le corps

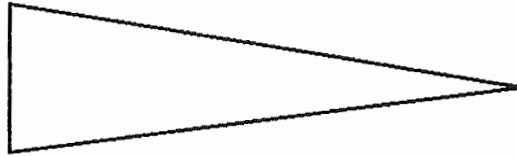
Nous ne commenterons que le calcul du cône d'ombre et le calcul de la zone d'ombre sur le corps.

Pour calculer le cône d'ombre, on a besoin de la projection du centre du corps lumineux, de la projection du centre du corps non lumineux dont on veut calculer le cône d'ombre et du rayon des deux corps.

La projection du cône d'ombre à l'aide d'une projection en perspective est composée d'une ellipse (projection de l'intersection du cône d'ombre du corps qui le produit et du corps) et d'un triangle, l'un ou l'autre pouvant être dégénéré et ne donner qu'une droite. Voici trois cas possibles de projection du cône :



Vue du cône de côté : composé d'une ellipse et d'un triangle



Vue du cône de profil : un triangle



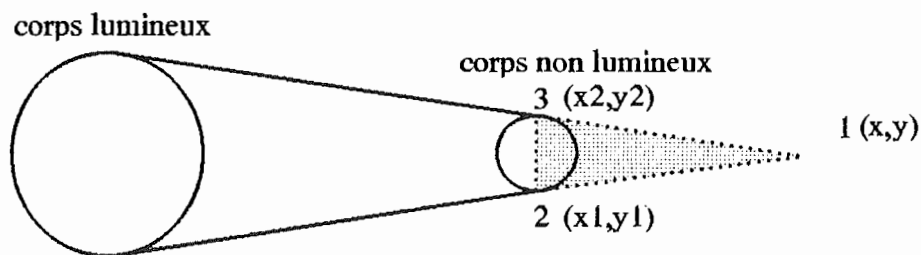
Vue du cône de face : un cercle


Projection du cône d'ombre

Figure 13

a) Calcul du triangle du cône d'ombre

Le triangle qui définit la projection peut être déterminé de la manière suivante:



 Cône d'ombre dont on recherche les coordonnées des points (1, 2, 3)

Rappel : Le triangle du cône d'ombre

Figure 14

1. recherche de l'équation de la droite passant par le centre des deux corps

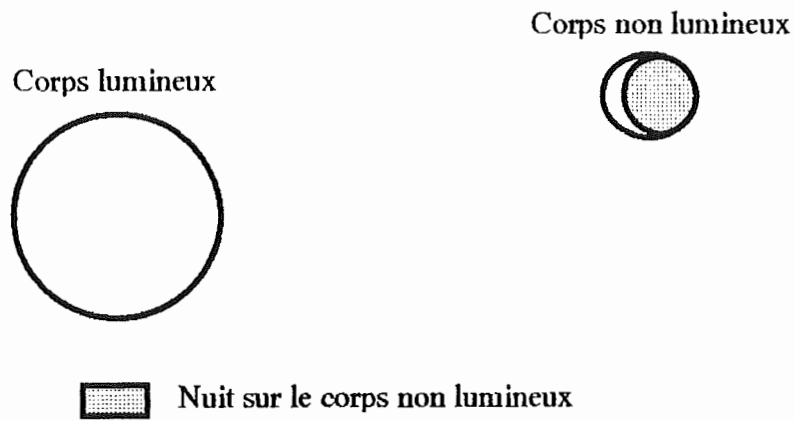
2. calcul de la distance entre le centre du corps non lumineux et l'extrémité du cône d'ombre (pointe du cône $1(x,y)$). Cette distance est obtenue en multipliant le rayon du corps non lumineux par la distance entre les deux centres et en divisant par la différence des rayons.
3. recherche des coordonnées de l'extrémité du cône d'ombre grâce à la distance calculée au point précédent (méthode : (x,y) appartient à la droite et se trouve à une distance connue du centre du corps non lumineux dans le sens opposé au corps lumineux).
4. recherche de l'équation de la droite passant par les deux autres extrémités du cône d'ombre. On va l'approximer par la droite perpendiculaire à celle calculée au point 1, et passant par le centre du corps non lumineux (méthode : coefficient angulaire de la droite = $-1/\text{coefficient angulaire de la droite du point 1}$, la droite passe par le centre du corps non lumineux).
5. recherche des coordonnées des deux extrémités restantes du cône d'ombre (méthode : (x_1,y_1) et (x_2,y_2) appartiennent à la droite recherchée au point 4 et ils se trouvent à une distance égale au rayon du centre du corps non lumineux). Dans ce calcul, il faut rappeler qu'on ne cherche que les extrémités du triangle.

b) Calcul de la zone d'ombre sur le corps

Le calcul de la zone d'ombre sur le corps (la partie du corps qui se trouve dans la nuit ou encore la partie elliptique de la projection du cône) est plus difficile car il faut considérer le problème de manière tridimensionnelle, alors que pour le cône d'ombre on pouvait travailler directement sur la projection, autrement dit en deux dimensions.

Pour calculer cette zone d'ombre, on a besoin des données suivantes : la projection du centre du corps lumineux, la projection du centre du corps non lumineux dont on veut calculer le cône d'ombre, du rayon des deux corps, et de la position des deux corps, mais dans l'espace.

Voici d'abord ce que l'on désire avoir à l'écran :



Vue à l'écran
Figure 15

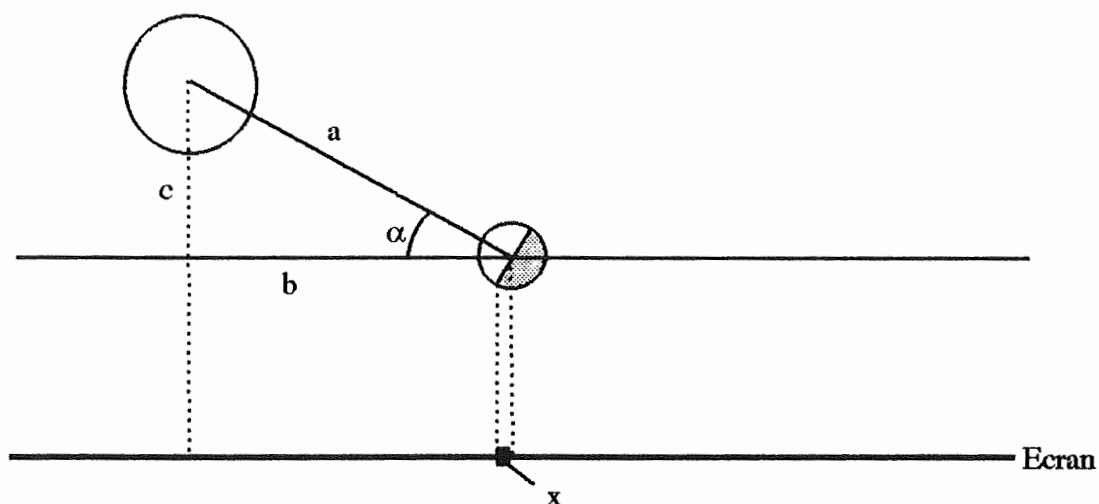
Pour pouvoir afficher ce genre de choses il est nécessaire de connaître les coordonnées des points qui reliés entre eux forment la zone d'ombre. Nous avons choisi, pour ce faire le paramètre x , qui est la distance entre la projection de l'extrémité de la zone d'ombre sur la droite qui relie la projection du centre du corps lumineux et la projection du centre du corps non lumineux, et la projection du centre du corps non lumineux.



Distance entre l'extrémité de la nuit et le centre du corps
Figure 16

Cette distance peut être connue grâce à un calcul trigonométrique sur les corps dans le plan perpendiculaire à l'écran passant par les deux centres :

Plan perpendiculaire à l'écran passant par les deux centres



Construction dans l'espace

Figure 17

où si le centre du corps lumineux est donné par les coordonnées (x_1, y_1, z_1) et le centre du corps non lumineux par (x_2, y_2, z_2) , et le rayon du corps non lumineux égal à r on a:

a = distance entre les centres

$$c = z_1 - z_2 = a \sin(\alpha)$$

$$\sin(\alpha) = c / a$$

$$x = r \sin(\alpha) = r \cdot c / a$$

Les étapes qui constituent l'algorithme de calcul de la zone d'ombre sont alors :

1. Calcul de la distance (x sur le schéma) entre la projection du centre du corps non lumineux et la projection du point extrême de la zone d'ombre.
2. Recherche du point p se trouvant à distance x du centre du corps sur la droite reliant les deux centres des corps.
3. Recherche du centre de l'arc passant par ce point extrême p et par les points 2 et 3 du cône d'ombre.
4. Recherche des coordonnées des points qui reliés entre eux forment l'arc et le demi cercle qui composent la zone d'ombre. Le stockage de ces coordonnées se fait dans une structure de polygone qui est défini dans le langage Turbo Pascal.

5. Affichage du polygone qui représente la zone d'ombre grâce à la structure de polygone.

Manuel d'utilisation du programme

Le niveau de détail de ce chapitre est sans doute trop important pour un informaticien ou une personne qui utilise souvent un ordinateur, mais il faut se rappeler que le programme écrit dans le cadre de ce mémoire est destiné avant tout à un usage didactique. Par conséquent, les personnes qui l'utiliseront ne connaissent peut être rien à la manipulation d'un PC. Ce chapitre a donc été écrit afin que ces personnes puissent utiliser facilement ce programme.

1. But du programme

Le but de ce programme est de simuler le mouvement de trois corps sphériques dans l'espace afin de comprendre le phénomène des éclipses étant donné qu'un des trois corps représenté est lumineux. On affiche donc les cônes d'ombres et les zones éclairées des corps non lumineux en fonction de la position du corps lumineux. L'ombre provoquée par un corps non lumineux sur un autre corps non lumineux ne sera pas représentée.

2. Matériel nécessaire

Pour faire fonctionner ce programme de manière efficace, il faut disposer d'au minimum un processeur du type Intel 80286. Ceci afin que la simulation ne paraisse pas trop saccadée. De plus, il est nécessaire de posséder un écran couleur géré par une carte EGA ou VGA afin de pouvoir distinguer correctement les corps et les ombres.

3. Modélisation du système

Avant de commenter l'utilisation du programme proprement dit, il est nécessaire d'expliquer la modélisation que nous avons choisie pour le mouvement des trois corps. Cela permettra en effet à l'utilisateur de comprendre l'utilité des paramètres initialisés dans le fichier et des paramètres qu'il peut modifier via l'interface (ceux-ci sont repris dans le point 4 sur la structure des fichiers de paramètres).

Nous avons modélisé le problème de la manière suivante. Le système est composé de trois corps : c_1 , c_2 , c_3 , qui ont chacun pour orbite un cercle. Chaque orbite est définie par le repère qui lui est associé. Ce repère a son origine au centre de l'orbite et ses axes XY définissent le plan de l'orbite, tandis que l'axe Z donne la direction de rotation. Ces repères sont exprimés dans le repère du corps c_1 , qui est le repère de référence, grâce à la matrice de changement de repère et le vecteur

origine. Les repères peuvent être soit fixes par rapport à l'observateur, soit dépendants d'un autre corps (i.e. le centre du repère de l'orbite du corps se déplace avec un autre corps). Ceci constitue l'ensemble des paramètres non modifiables à partir de l'interface, il faudra si on veut les changer, éditer un fichier du type décrit au point suivant.

Pour que la modélisation soit complète, il faut cependant définir d'autres paramètres. Pour chaque corps, nous devons également connaître le rayon orbital, le rayon du corps, le temps de révolution, la position de départ sur l'orbite (qui est donnée par un angle compté à partir l'axe X vers l'axe Y), et le numéro du corps lumineux.

4. Structure des fichiers de paramètres

Les fichiers de paramètres contiennent l'ensemble des paramètres nécessaires à la simulation, ce qui signifie que, outre les paramètres de la modélisation, ils contiennent les paramètres suivants : la position de l'observateur, la position du point de visée, l'angle de vision, le temps entre chaque affichage, l'incréméntation de ce temps, ainsi que la couleur de chaque corps (l'explication de ces paramètres est donnée au point 6.3.1).

Ces paramètres sont stockés comme suit dans le fichier :

- taille du commentaire sur les paramètres (maximum 255 caractères)
- commentaire sur l'ensemble des paramètres
- pour $i=1$ à 3
 - matrice (3x3 de changement de repère de C_i vers C_1)
 - vecteur origine de C_i
 - dépendance du repère de C_i (0 : Fixe, 1 : Dépendant)
 - corps dont il dépend (valeur prise en compte uniquement si le repère est dépendant)
- position de l'observateur
- position du point de visée
- temps entre deux affichages
- incréméntation du temps

- le numéro du corps lumineux

- pour $i=1$ à 3

- couleur de C_i
- rayon de C_i
- rayon de l'orbite de C_i
- angle de départ de C_i
- temps de révolution

Chacune de ces valeurs est suivie par un carriage return (retour à la ligne ou frappe d'un ENTER dans l'édition du fichier). Ce fichier est de type texte (ASCII). Il suffit donc pour le modifier d'utiliser un traitement de texte quelconque.

Voici un exemple d'un tel fichier. Les paramètres qu'il contient déterminent un système où les corps C_1 et C_2 ont des repères fixes. Ces deux repères sont équivalents (i.e. Les deux corps tournent dans le même plan et ont leur centre d'orbite identique). Le troisième repère est dépendant du corps C_2 et le plan de son orbite est incliné de cinq degrés par rapport au plan C_1 . Un commentaire des paramètres du fichier est écrit entre accolades. Il est bien entendu que ces commentaires ne figurent pas dans le fichier.

exemple : fichier ECL1.ECL

```
43                               { Taille du commentaire qui suit }
vue d'une éclipse de Soleil depuis la Terre
1.0000000000E+00                { Matrice de changement de repère de C1 }
0.0000000000E+00                { vers C1 : matrice identité }
0.0000000000E+00
0.0000000000E+00
1.0000000000E+00
0.0000000000E+00
0.0000000000E+00
0.0000000000E+00
1.0000000000E+00
0                               { Vecteur origine de C1 }
0
0
0
0                               { 0 signifie que le repère est fixe par rapport }
0                               { l'observateur. }
1.0000000000E+00                { Matrice de changement de repère de C2 }
0.0000000000E+00                { vers C1 }
0.0000000000E+00
```

0.0000000000E+00
 1.0000000000E+00
 0.0000000000E+00
 0.0000000000E+00
 0.0000000000E+00
 1.0000000000E+00
 0 { Vecteur origine de C2 }
 0
 0
 0 { 0 signifie que le repère est fixe par rapport }
 0 { à l'observateur }
 1.0000000000E+00 { Matrice de changement de repère de C3 }
 0.0000000000E+00 { vers C1 }
 0.0000000000E+00
 0.0000000000E+00
 9.9610000000E-01
 -8.7100000000E-02
 0.0000000000E+00
 8.7100000000E-02
 9.9610000000E-01
 1496 { Vecteur origine de C3 }
 0
 0
 1 { 1 Signifie que le repère est dépendant }
 2 { C3 dépend du corps 2 }

{ Seules les paramètres qui suivent sont affichés à l'écran lors de l'utilisation du programme et peuvent donc être modifiés dans celui-ci }

1400 { Position de l'observateur }
 0
 0
 0 { Point de visée }
 0
 0
 20 { Angle de vision }
 1.0000000000E-02 { Temps entre deux affichages successifs }
 1.0000000000E+00 { Valeur d'incrément de ce temps }
 1 { Numéro du corps lumineux : ici C1 }
 15 { Couleur de C1 : Blanc }
 170 { Rayon de C1 }
 0 { Rayon de l'orbite de C1 }
 0 { Angle de départ de C1 }

0	{ Temps de révolution de C1 : C1 est Fixe }
1	{ Couleur de C2 : Bleu }
63	{ Rayon de C2 }
1496	{ Rayon de l'orbite de C2 }
0	{ Angle de départ de C2 }
365	{ Temps de révolution de C2 }
2	{ Couleur de C3 : Vert }
35	{ Rayon de C3 }
384	{ Rayon de l'orbite de C3 }
180	{ Angle de départ de C3 }
27	{ Temps de révolution de C3 }

5. Chargement du programme

Nous supposons que l'utilisateur travaille avec le programme stocké sur disquette. Dans ce cas, le chargement du programme doit s'effectuer de la manière suivante, une fois l'ordinateur allumé :

- Insérez la disquette contenant le programme dans un des lecteurs correspondant au format de la disquette utilisée, le a par exemple.
- Mettez-vous sur le lecteur choisi (Tapez a: puis tapez sur la touche **ENTER**).
- Regardez le directory (Tapez dir puis tapez sur la touche **ENTER**).
- Vérifiez que dans la liste qui apparaît à l'écran se trouve bien un fichier nommé : **ECLIPSE.EXE** et au moins un autre fichier terminé par **ECL**. Si **ECLIPSE.EXE** n'existe pas, il ne doit pas s'agir de la bonne disquette. Si, par contre, il n'existe aucun fichier se terminant par **ECL**, vous pouvez en créer un grâce à un éditeur de texte. La structure du fichier a été détaillée auparavant.
- Si la liste contient bien les fichiers mentionnés ci-dessus, tapez alors **eclipse** puis tapez sur la touche **ENTER**.

Notons que l'utilisation de ce programme sur un disque dur est possible, mais nous n'en détaillerons pas l'usage car la connaissance de plusieurs instructions DOS est nécessaire.

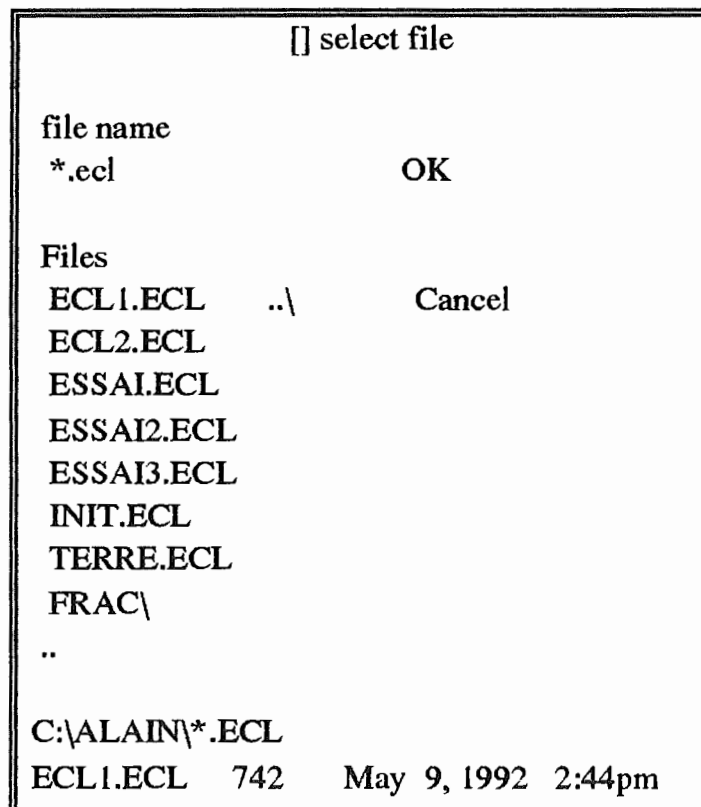
6. Utilisation du programme

Dans ce paragraphe, nous allons décrire l'utilisation proprement dite du programme en détaillant les points suivants :

- Initialisation des paramètres
- Activation d'un cadre
- Interface traditionnelle
- Interface visuelle
- Chargement et sauvegarde des paramètres
- Quitter le programme
- La simulation

6.1 Initialisation des paramètres

Après avoir exécuté la liste des opérations décrites dans le paragraphe ci-dessus, vous vous trouverez dans le programme. La première chose qui apparaît à l'écran est une fenêtre dont voici un contenu possible :



Fenêtre 1

Cette fenêtre permet de sélectionner le fichier de paramètres que l'on désire charger. On peut le choisir de trois façons :

- en cliquant deux fois sur le nom désiré grâce à la souris
- en se déplaçant dans les menus grâce à la touche **TAB** et en sélectionnant le nom grâce aux touches de direction, puis en appuyant deux fois sur la touche **ENTER** (une fois pour sélectionner le nom du fichier et une fois pour confirmer)
- en tapant le nom du fichier dans le cadre prévu à cet effet

Une fois le nom choisi, le programme lit le commentaire correspondant à ce fichier et l'affiche dans une fenêtre. Si il s'agit bien du fichier que vous désirez charger, taper sur **ENTER**, ou sur la lettre **o**, ou encore cliquer avec la souris sur le cadre *Ok*. Les paramètres se chargeront et la fenêtre suivante apparaîtra à l'écran. Si il ne s'agit pas du fichier que vous désirez charger, mettez vous sur le cadre *Cancel* grâce à la touche **TAB**, puis tapez sur la touche **ENTER**, ou encore cliquez sur le cadre *Cancel*, grâce à la souris, vous retournerez alors dans la fenêtre de choix du fichier à charger.

6.2 Activation d'un cadre

Sur base de cet exemple constituant la première étape de l'exécution, nous pouvons dire que d'une manière générale, l'activation d'un cadre peut se faire de trois manières :

- Cliquer avec la souris sur le cadre que l'on désire activer.
- Se positionner sur le cadre que l'on désire activer grâce à la touche **TAB**, puis appuyer sur **ENTER**.
- Taper sur une touche du clavier qui correspond à une lettre de couleur jaune à l'écran, cela déclenchera immédiatement l'action dont cette lettre fait partie.

6.3 Interface traditionnelle

Revenons à l'exécution du programme. Une fois le fichier de paramètres chargé, la fenêtre ci-dessous apparaîtra à l'écran :

[] Simulateur d'éclipses					
Coordonnées de l'observateur < 200 , 0 , 0 >					
Coordonnées du pt de visée < 10000 , 0 , 0 >					
Angle de vision 20					
Temps entre deux affichages 0.10 Incrémentation du temps 1.00					
Le corps lumineux est le corps 1					
	Couleur	Rayon du corps	Rayon de l'orbite	Angle de départ	Année
Corps 1	15	70	0	0	0
Corps 2	1	63	1496	0	365
Corps 3	2	20	384	180	27
Start	Vue	Charger	Sauver	Quitter	

Fenêtre 2

Tous les paramètres qui figurent dans cet écran peuvent être modifiés, il suffit que le cadre dans lequel est affichée la valeur du paramètre à modifier soit activé. Il faut ensuite introduire la nouvelle valeur.

6.3.1 Explication des paramètres que l'utilisateur peut modifier interactivement

Certaines explications pourront paraître redondantes par rapport à ce qui a été dit dans les chapitres précédents, mais il faut se rappeler que ce chapitre sert surtout de manuel d'utilisation.

Les trois premières valeurs, qui sont regroupées sous le titre *coordonnées de l'observateur*, représentent comme leur nom l'indique la position de l'observateur (i.e. où on se trouve pour observer le phénomène). Cette position est exprimée en coordonnées cartésiennes dans le repère de référence (voir modélisation).

Les trois valeurs suivantes, regroupées sous le titre *coordonnées du pt de visée*, permettent de définir la direction de vision de l'observateur (i.e. vers où il regarde). Cette position est exprimée en coordonnées cartésiennes dans le repère de référence (voir modélisation).

L'*angle de vision*, qui est la valeur suivante, définit la portion de l'espace qui est visible par l'utilisateur. Soulignons que cette valeur de départ pourra être modifiée plus tard. En effet, comme on le verra plus loin, le simulateur permet en cours de simulation de faire des zooms. Cela revient à modifier la portion de l'espace visible par l'utilisateur, c'est à dire modifier l'angle de vision.

Les deux paramètres suivants, c'est à dire *temps entre deux affichages* et *incrément du temps*, permettent, quant à eux, de modifier la vitesse du simulateur. Le premier est une valeur de départ qui exprimera la vitesse initiale, tandis que le second est le paramètre qui permet de connaître la valeur que l'on désire ajouter ou retirer, chaque fois que l'utilisateur en fera la demande en cours de simulation.

Les paramètres qui suivent sont d'une autre nature que les précédents. En effet, l'ensemble des paramètres que nous avons définis jusqu'à présent permettaient essentiellement de définir la simulation du point de vue de l'observateur. Ces paramètres sont regroupés dans la première partie de l'écran. Les paramètres que nous allons décrire ci-dessous permettent, eux, de caractériser la simulation proprement dite. Ils sont regroupés dans la deuxième partie de la fenêtre.

La première valeur de ce deuxième groupe de paramètres, le *numéro du corps lumineux*, permet de définir, comme son nom l'indique, quel est le corps lumineux. Le choix de ce corps se fait grâce à son numéro. Il est à noter que pour pouvoir introduire la valeur que l'utilisateur désire, il est nécessaire qu'il connaisse la modélisation qu'il a chargé au départ (voir point 3 modélisation), c'est à dire qu'il connaisse l'ordre des corps.

Les paramètres suivants sont relatifs à la modélisation du problème. Une valeur doit être introduite pour chacun des corps, c'est pour cela que nous avons placé les valeurs dans un tableau (voir deuxième partie de la fenêtre)

- *couleur* : permet de définir la couleur du corps correspondant en introduisant un entier compris entre 0 et 15. Ces valeurs permettent de définir les couleurs suivantes :

0 : Noir

1 : Bleu

- 2 : Vert
- 3 : Cyan
- 4 : Rouge
- 5 : Magenta
- 6 : Brun
- 7 : Gris clair
- 8 : Gris foncé
- 9 : Bleu clair
- 10 : Vert clair
- 11 : Cyan clair
- 12 : Rouge clair
- 13 : Magenta clair
- 14 : Jaune
- 15 : Blanc

- *rayon du corps* : définit le rayon du corps dans l'espace, c'est à dire avant la projection
- *rayon de l'orbite* : définit le rayon orbital du corps correspondant dans l'espace
- *angle de départ* : cet angle définit la position initiale du corps sur son orbite, il est à noter qu'elle dépend également des positions des axes X et Y du repère associé au corps
- *année* : donne le nombre de jours que met le corps pour effectuer une révolution. Un nombre de jours égal à zéro signifie que le corps ne se déplace pas.

6.4 Interface visuelle

Il existe dans le programme une autre manière d'introduire les deux premiers paramètres décrits au point précédent : la position de l'observateur et le point de visée. En sélectionnant l'action *Vue* de l'interface, on obtient une vision dans l'espace du repère de référence, de l'orbite des corps, de la position de l'observateur, du point de visée et de sa direction de vision qui est la droite reliant l'observateur au point de visée.

Cette vision est intéressante à plus d'un titre, elle permet de situer dans l'espace des points difficiles à se représenter, mais en plus elle permet de modifier la position de ces points.

Avant de voir comment on peut modifier ces points, voyons un exemple. Les axes du repère sont affichés en bleu, la direction de vision est en vert, les orbites sont en rouge et le point que l'on peut déplacer est affiché en vert avec une lettre qui l'identifie (O: position de l'observateur, V : point de visée). La projection de la direction de vision sur les axes, est, elle, affichée en bleu clair.

Le dessin qui suit est la *Figure 18* qui contient un exemple de ce qui est affiché lors d'un appel à VUE. Il s'agit de la modélisation vue jusqu' à présent, mais avec le position de l'observateur = (10000,0,2000) et le point de visée = (0,0,0).

x
x=x+10

y
y=y+10

z
z=z+10

X
x=x+100

Y
y=y+100

Z
z=z+100

-
moins

v
Visée

q
Quitter



La modification des coordonnées se réalise à l'aide des touches suivantes :

- **x** : Si le mode addition est actif, pousser sur cette touche ajoutera 10 à la coordonnée x du point identifié à l'écran, sinon elle soustraira 10 à cette même coordonnée
- **y** : Si le mode addition est actif, pousser sur cette touche ajoutera 10 à la coordonnée y du point identifié à l'écran, sinon elle soustraira 10 à cette même coordonnée
- **z** : Si le mode addition est actif, pousser sur cette touche ajoutera 10 à la coordonnée z du point identifié à l'écran, sinon elle soustraira 10 à cette même coordonnée
- **X** : Si le mode addition est actif, pousser sur cette touche ajoutera 100 à la coordonnée x du point identifié à l'écran, sinon elle soustraira 100 à cette même coordonnée
- **Y** : Si le mode addition est actif, pousser sur cette touche ajoutera 100 à la coordonnée y du point identifié à l'écran, sinon elle soustraira 100 à cette même coordonnée
- **Z** : Si le mode addition est actif, pousser sur cette touche ajoutera 100 à la coordonnée z du point identifié à l'écran, sinon elle soustraira 100 à cette même coordonnée
- **-** : permet de passer en mode soustraction
- **+** : permet de passer en mode addition
- **v** : identifie à l'écran le point de visée
- **o** : identifie à l'écran la position de l'observateur
- **q** : permet de quitter la vue et de retourner à l'écran précédent (fenêtre 2).

Un rappel des touches disponibles et de leur effet est comme on a pu le voir sur l'exemple, affiché en permanence sur la droite de l'écran (en jaune la touche, en noir son effet).

6.5 Chargement et sauvegarde des paramètres

Pour charger un fichier de paramètres existants à partir de l'interface, il suffit que l'utilisateur choisisse l'action *Charger* dans la fenêtre 2. La fenêtre 1 s'affiche alors avec la liste des fichiers disponibles (rappelons que cette fenêtre est la première qui apparaît lors du lancement du programme). L'utilisateur doit alors en sélectionner un comme indiqué au paragraphe 4.1.

Si l'utilisateur modifie l'une ou l'autre valeur des paramètres du fichier chargé, il a la possibilité de les sauvegarder dans le même fichier ou dans un autre. Il doit pour ce faire choisir l'action *Sauver*. La fenêtre 1 apparaît pour choisir le nom du fichier de sauvegarde. Ensuite, l'utilisateur pourra ajouter son propre commentaire sur le fichier.

6.6 Quitter le programme

Pour quitter le programme en cours de simulation, il suffit d'activer le cadre *Quitter*, puis de pousser simultanément sur les touches ALT et X.

6.7 La simulation

Dès que les paramètres ont été initialisés à la valeur désirée, l'utilisateur peut lancer la simulation. Pour ce faire, il lui suffit d'activer la fenêtre *Start*.

L'écran qui apparaît est composé de la manière suivante :

- A gauche, l'ensemble des touches de commandes (un détail de chacune de ces touches sera donné plus tard)
- En bas à gauche le temps écoulés pour la simulation en jours
- En bas à droite, le temps de révolution de chaque corps indiqué dans la couleur du corps correspondant. Ceci permet de se faire une idée de l'évolution du temps sur chacun des corps.
- Au centre, la partie visible de l'espace pour l'observateur.

Voici, avant de détaillé les commandes le genre de choses (*figure 19*) qu'on peut voir à l'écran si on considère la modélisation décrite au début de ce chapitre, mais avec le point de visée et la position de l'observateur définie comme dans la *figure 18*.

W
Zoom +
X
Zoom -
P
Pause
C
Ombre
↑
Temps +
↓
Temps -
Shift ↑
Temps ++
Shift ↓
Temps --
Q
Quitter



Nombre de jours pour 1 révolution

Temps : 6.000 jours

Fixe
365
27

fo.

Les commandes que l'utilisateur peut utiliser en cours de simulation sont les suivantes :

- **w** : Procède à un zoom avant si l'angle de vision est supérieur à 2 (i.e. angle de vision = angle de vision / 2).
- **x** : Procède à un zoom arrière si l'angle de vision est inférieur à 90 (i.e. angle de vision = angle de vision * 2).
- **p** : Pause de la simulation : arrêt du mouvement des corps et affichage de chacune des orbites.
- **c** : Changement de l'affichage : si on affichait le cône d'ombre, on n'affiche plus que la nuit, tandis que si on n'affichait que la nuit sur un corps, on affiche le cône d'ombre.
- **Flèche vers le haut** : diminue la vitesse de la simulation. Le temps entre deux affichages successifs sera incrémenté de la valeur introduite dans l'interface.
- **Flèche vers le bas** : augmente la vitesse de la simulation. Le temps entre deux affichages successifs sera diminué de la valeur introduite dans l'interface.
- **Shift Flèche vers le haut** : augmente la vitesse de la simulation. Le temps entre deux affichages successifs sera incrémenté de dix fois la valeur introduite dans l'interface.
- **Shift Flèche vers le bas** : diminue la vitesse de la simulation. Le temps entre deux affichages successifs sera diminué de dix fois la valeur introduite dans l'interface.
- **q** : permet de quitter la simulation et de retourner à l'interface

7. L'usage didactique

Nous avons voulu consacrer les quelques lignes qui suivent à montrer comment l'application développée dans ce mémoire peut être utilisée de manière didactique.

Pour documenter l'approche théorique du phénomène des éclipses (voir chapitre 2), le professeur pourrait préparer différents fichiers de paramètres correspondants chacun à un cas particulier ou à une étape importante dans le déroulement des éclipses. Au cours de la leçon, il suffirait alors de charger au fur et à mesure les différents fichiers de paramètres et de lancer chaque fois la simulation.

Le professeur pourrait par exemple montrer :

- Le mouvement des trois corps dans l'espace suivant différents angles de vues.
- Une éclipse vue d'un des trois corps.
- La difficulté d'avoir une éclipse de Soleil en un endroit précis de la Terre.
- L'inclinaison non négligeable de la Lune par rapport à l'orbite de la Terre.
- Le nombre plus important d'éclipse de Soleil sur an que d'éclipses de Lunes.
-

Un fichier de paramètres existe pour chacun des points cités ci-dessus (voir le directory et le commentaire associé à chacun des fichiers).

Les améliorations possibles du programme de simulation

Le programme de simulation des éclipses que nous avons développé est limité par rapport à ce qu'il serait possible de réaliser. Ceci résulte de certains choix délibérés (modélisation que nous avons choisie), mais également du fait que ce programme est l'objet d'un mémoire dont la durée est limitée à une année. De plus, le langage Pascal et les machines grâce auxquelles ce mémoire a été développé ne permettent pas le nombre considérable d'opérations qu'il faudrait exécuter entre deux affichages successifs pour intégrer dans la simulation toutes les idées que nous n'avons pas programmées, tout en gardant le mouvement fluide.

Il nous a paru intéressant de commenter les idées d'amélioration auxquelles nous avons réfléchi. Avant de commenter ces idées, en voici d'abord la liste :

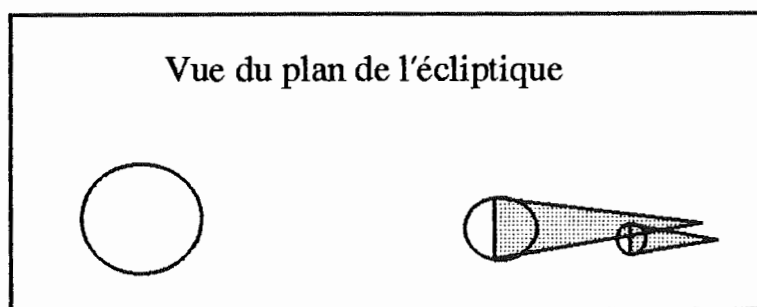
1. Plusieurs vues simultanément affichées à l'écran.
2. Modification d'autres paramètres lors de la simulation
3. Pause automatique lors de phénomènes intéressants (Eclipse, passage aux noeuds, ...).
4. Affichage des zones de pénombre.
5. Affichage d'ombres qui ne sont plus des cônes.
6. Affichage de la projection de l'ombre d'un corps sur un autre.
7. Affichage des différentes orbites à tout moment
8. Observateur sur un corps qui tourne sur lui-même
9. Observateur sur un corps qui ne tourne pas sur lui-même
10. Modification plus aisée des positions du centre de l'orbite de chaque corps.

1. Plusieurs vues simultanément affichées à l'écran

Le multifenêtrage permettrait à l'utilisateur d'avoir une meilleure vue du phénomène tridimensionnel. En effet, le fait de pouvoir observer plusieurs vues simultanément à l'écran permettrait à l'utilisateur de reconstituer plus facilement la position relative des objets dans l'espace. Cela éviterait de mauvaises interprétations des phénomènes.

En effet, si le programme ne propose qu'une seule vue, on peut avoir, par exemple, l'impression qu'il y a une éclipse alors qu'il ne s'agit que d'une illusion d'optique provoquée par la projection. La possibilité de regarder simultanément d'autres vues permettrait de distinguer les cas (voir les figures ci-dessous) :

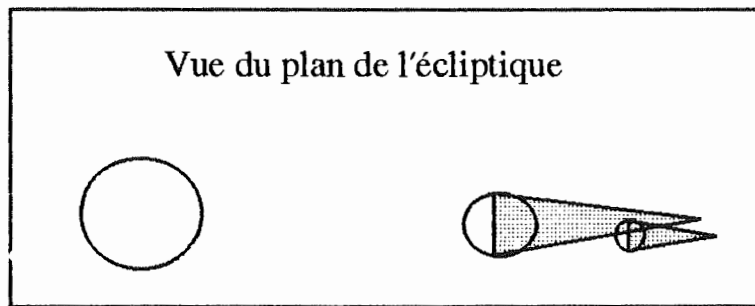
1. Cas de non éclipse :



Cas de non éclipse alors que la vue unique depuis le plan de l'écliptique pourrait laisser supposer le contraire

Figure 20

2. Cas d'éclipse :



Cas d'éclipse : la même projection sur l'écliptique peut signifier qu'il y a une éclipse

Figure 21

D'un point de vue conceptuelle la programmation de l'affichage de plusieurs vues simultanément est assez simple. Il suffit de réutiliser pour chacune des vues les phases décrites pour une vue dans ce programme (exemple : calcul de paramètres du changement de repère associé à la projection en perspective pour chacune des vues). Ainsi l'apport d'une deuxième fenêtre multiplierait par deux le temps de calculs entre deux affichages successifs.

2. Modification d'autres paramètres lors de la simulation

Le programme permet lors de la simulation de faire une série de zooms, d'accélérer ou de freiner la simulation, ou encore de faire une pause qui permet de voir les orbites des corps.

Il peut être intéressant de proposer à l'utilisateur du programme d'agir directement sur la simulation par d'autres moyens que ceux déjà évoqués. L'utilisateur devrait par exemple pouvoir modifier la position ou la grandeur d'un corps grâce à la souris ou au clavier.

Si cette proposition semble attrayante, elle pose néanmoins un problème de programmation et plus particulièrement un problème de temps de réponse. En effet, le simulateur a été programmé en faisant un calcul des orbites des corps à priori, or un changement de la position de ces corps pourrait changer leurs orbites, ce qui nécessiterait de recalculer ces orbites au cours de la simulation, d'où modification de la structure du programme et ralentissement du temps de réponse.

De plus, la position que l'on fournirait au programme serait une position sur le plan de projection, d'où ambiguïté quant à l'interprétation de la position du corps dans l'espace.

3. Pause automatique lors de phénomènes intéressants

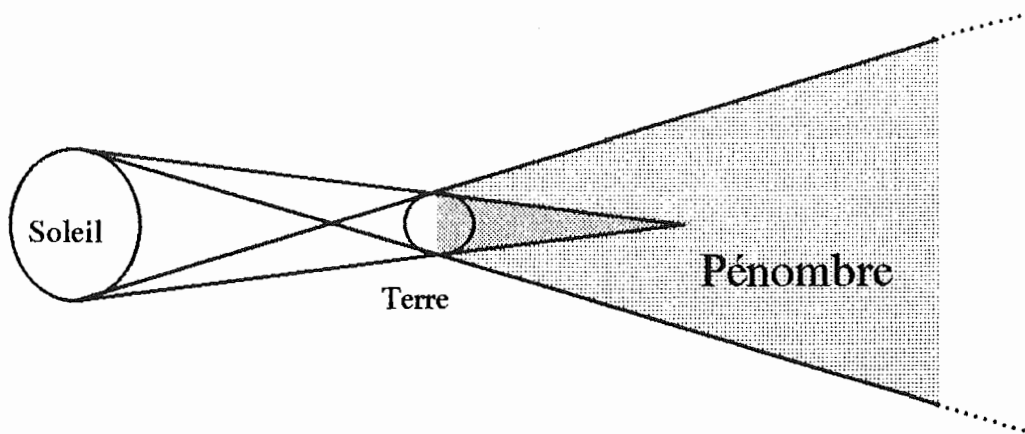
Le simulateur ayant pour but essentiel la compréhension des éclipses, il serait intéressant que celui-ci réagisse lorsqu'une éclipse se produit. Dans le même ordre d'idée, il peut sembler opportun de signaler à l'utilisateur d'autres phénomènes importants tels que le passage d'un corps aux noeuds, la conjonction des corps,

Ces propositions pourraient être ajoutées dans le programme en calculant et conservant avant la simulation le moment et la période où se produisent ces phénomènes (i.e. dans la phase "calcul avant simulation"). Ensuite, lors de l'affichage, il faudrait stopper la simulation chaque fois que le système est le plus proche d'un des moments calculés.

4. Affichage des zones de pénombre

Ce point est très important pour la compréhension du phénomène des éclipses (voir chapitre 2). Dans le programme tel qu'il existe actuellement, seul le cône d'ombre est affiché. On perd donc une partie du phénomène en ne l'affichant pas. Pourquoi dès lors ne pas l'avoir programmé ?

Si on affichait les zones de pénombre des deux corps non lumineux, on ne verrait, dans certains cas, plus rien à l'écran. En effet, contrairement, aux cônes d'ombres, la zone de pénombre n'est pas limitée dans l'espace

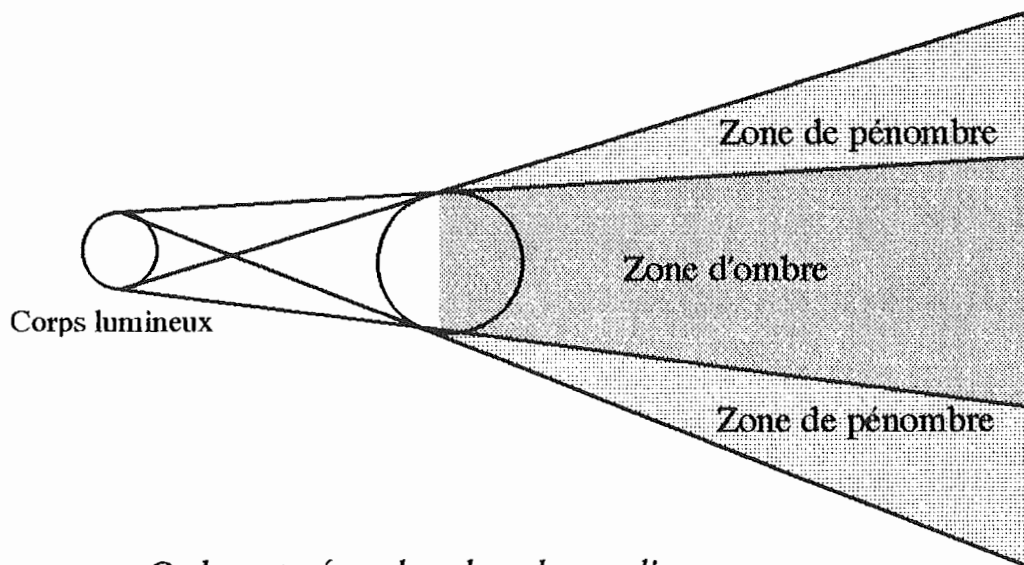


*Grandeur de la zone de Pénombre
Figure 22*

Une trop grande partie de l'écran serait alors consacrée à cette zone, ce qui diminuerait considérablement la vision du phénomène. Bien entendu, on pourrait tout de même envisager d'autres solutions que l'affichage complet de la zone de pénombre. On pourrait par exemple, ne voir que sa projection sur les corps.

5. Affichage d'ombres qui ne sont plus des cônes

Dans tout le mémoire, nous nous sommes basés sur le fait que l'ombre créée par un corps était toujours un cône. Ceci est dû au fait que les corps lumineux sont généralement plus gros que les corps non lumineux. Cependant, on pourrait imaginer qu'un corps lumineux soit plus petit qu'un corps non lumineux. On aurait alors le phénomène représenté sur le schéma qui suit :



*Ombre et pénombre dans le cas d'un corps
lumineux plus petit que le corps non lumineux
Figure 23*

Ce phénomène pourrait s'avérer intéressant à représenter dans une simulation. Cependant il poserait les mêmes problèmes que pour l'affichage de la zone de pénombre.

Il est à noter que dans un cas comme celui-ci le programme actuel n'affichera pas le cône d'ombre. Il affichera le corps soit entièrement dans la nuit (si le corps lumineux est plus éloigné de l'observateur que le corps non lumineux) soit entièrement dans le jour (dans le cas inverse).

6. Affichage des différentes orbites à tout moment

Un autre point que nous avons envisagé et qu'il pourrait être intéressant de programmer est l'affichage constant des orbites des différents corps. Rappelons que pour le moment, l'affichage des orbites des différents corps n'est réalisé que lors d'une pause. L'affichage constant des orbites pourrait être intéressant pour mieux suivre l'évolution des corps sur leurs orbites.

Cependant, il ne faut pas oublier que dans la modélisation que nous avons proposée, l'orbite pouvait, comme c'est le cas dans la réalité, être dépendante d'un autre corps. Si nous voulons afficher de manière permanente les différentes orbites il faudrait recalculer à chaque fois les nouvelles orbites entre deux affichages ce qui ralentirait la simulation.

7. Affichage de la projection de l'ombre d'un corps sur un autre

S'il est un point que nous aurions voulu ajouter au programme, c'est celui-ci. En effet, dans l'état actuel des choses, il est impossible de voir à l'écran ce qui se produit réellement lors d'une éclipse, c'est à dire une partie du corps est non éclairée par suite de l'éclipse.

Or ce point est très important pour la compréhension des éclipses et surtout pour l'observation des moments où il y a une éclipse. Par exemple, si on considère le cas d'une éclipse de Soleil vue d'un point situé au dessus de la Lune, on ne verra aucune ombre sur la Terre à l'écran même s'il y a une éclipse.

Cette idée n'a pas été implémentée. En effet, le phénomène est assez lourd car il faut déterminer l'intersection de deux corps dans l'espace, ce qui n'est pas facile à résoudre étant donné que les calculs ne peuvent être trop longs pour ne pas ralentir l'affichage de la simulation. Il faudrait pouvoir calculer cette intersection de manière plus rapide et plus efficace. Or, à l'heure actuelle nous n'avons pas trouvé de solution intéressante.

8. Observateur sur un corps qui tourne sur lui-même

La rotation des corps sur eux-mêmes avec la possibilité d'attacher l'observateur à un des corps permettrait d'avoir une vue bien plus réaliste du phénomène des éclipses lors de l'observation de celui-ci depuis un des corps. On pourrait ainsi observer des levers de Soleil, un horizon,

Il faudrait pour ajouter ce point au programme actuel, revoir en partie la modélisation du problème : il faudrait associer à chaque corps un repère tangent à la surface.

9. Observateur sur un corps qui ne tourne pas sur lui-même

Une solution au problème posé dans le point précédent sans changer la modélisation, est de se mettre sur la surface du corps mais comme si on se déplaçait à la même vitesse que lui dans le sens opposé. Ce qui dans le programme consiste à réaliser une modélisation où le corps sur lequel on veut se mettre est fixe, et à positionner l'observateur sur ce corps.

Or, dans ce mémoire, l'entrée des paramètres qui définissent la position de l'observateur et du point de visée peut se faire soit numériquement, soit grâce à une interface graphique. Cependant, même si cette solution permet à l'utilisateur de se positionner n'importe où, elle ne permet pas de se positionner **facilement** sur un des corps. En effet, avant de préciser les coordonnées dans l'interface, l'utilisateur doit faire des calculs pour connaître la position du corps et surtout la position de sa surface.

On pourrait faciliter cette introduction en complétant l'interface, par exemple, avec une action qui calculerait la position de l'observateur sur le corps désiré en fonction d'une latitude et d'une longitude. Ce point n'a pas été pris en compte dans ce mémoire car nous avons axé notre travail sur la réalisation de la simulation.

10. Modification plus aisée des positions du centre de l'orbite de chaque corps

Il s'agit ici d'un problème d'interface. Dans le programme tel qu'il est conçu actuellement (voir chapitre 3), les paramètres qui définissent le repère associé au corps (matrice de changement de repère vers le repère de référence) ne sont pas directement modifiables à partir du programme. En effet, il faut utiliser un éditeur de texte pour corriger les valeurs existantes dans les fichiers de paramétrisation. Cette tâche s'avère assez lourde.

Il serait donc intéressant de pouvoir le faire autrement. Il y a à notre sens deux manières de procéder :

- numériquement : en complétant l'interface avec une fenêtre où seraient affichées les matrices d'orbites. On pourrait ainsi modifier les valeurs des composantes de ces matrices.
- graphiquement : en ayant la possibilité grâce à la souris ou au clavier de bouger le centre de l'orbite.

Le travail n'ayant pas été axé sur l'interface, mais bien sur la simulation, cette amélioration n'a pas été implémentée.

Projet d'avenir

Comme nous en avons parlé dans l'introduction le sujet de ce mémoire est venu de l'idée qu'il pourrait être intéressant de disposer d'une sorte de planétarium sur un ordinateur de type PC, et cela pour un usage didactique. Le programme qui a été réalisé respecte bien cette idée mais ses fonctionnalités sont bien sûr très limitées en comparaison de celles du planétarium. Toutefois, vu les résultats obtenus dans ce mémoire, le projet général présenté dans l'introduction nous semble justifié.

C'est pourquoi, nous avons voulu conclure en consacrant un chapitre de ce mémoire à l'étude du projet général et à l'analyse de quelques pistes de réflexion qui pourraient aider les personnes qui voudraient compléter le programme de ce mémoire.

1. Un programme d'interface général

L'application écrite dans ce mémoire est indépendante de l'interface existante, celle-ci n'a été développée que pour avoir un résultat déjà utilisable (voir chapitre 2 point 4.2.).

L'interface que nous voudrions mettre au point doit être beaucoup plus générale : elle devrait permettre à un professeur de créer ses propres leçons sur ordinateur grâce à un fichier de commandes (choix des graphiques (*1), des commentaires, de leur ordre d'apparition,...) une bibliothèque de graphiques, et un fichier de menus de leçons (composition d'un menu reprenant l'ensemble des leçons créées, leçons déjà vues, ...).

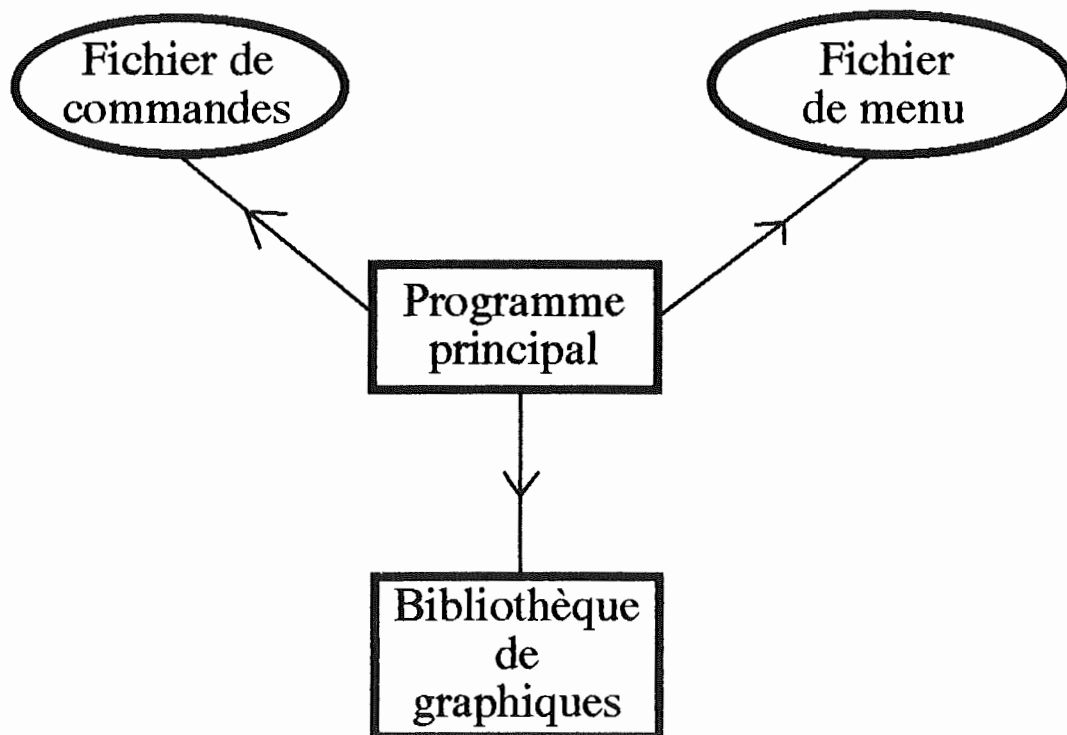
De manière plus détaillée, voici comment l'application pourrait fonctionner lors de l'exécution d'une leçon que le professeur aurait créée :

1. Lecture par le coordinateur d'un fichier menu comprenant la liste des leçons disponibles.
2. Affichage du menu à l'écran

*1 Un graphique serait soit un dessin, soit une simulation. Dans le cas d'une simulation, le coordinateur perdrait la main au profit de la simulation jusqu'à ce que celle-ci soit finie, il ne pourrait donc plus afficher un texte. Tandis que dans un dessin unique, une fois le dessin affiché, le coordinateur reprendrait la main et pourrait ainsi compléter le dessin à l'aide d'un autre dessin, ou d'un texte.

3. Après sélection d'une leçon par l'utilisateur, déroulement de la leçon grâce au fichier de commandes associé à cette leçon. Plus précisément, le coordinateur lit au fur et à mesure dans le fichier de commandes les textes et les graphiques à afficher lors de chaque étape de la leçon (L'ordre d'afficher du texte ou des graphiques se ferait par l'intermédiaire de commandes se trouvant dans le fichier de commandes.). Ensuite, le coordinateur extrait les graphiques désirés de la bibliothèque de graphiques et les affiche à l'écran dans l'ordre imposé.

On aurait ainsi la structure suivante pour l'interface générale que nous voudrions mettre au point :



Structure du programme didactique sur l'astronomie

Figure 24

Avec cette figure, on veut montrer que le programme principal est le coordinateur entre les fichiers et la bibliothèque de graphiques. Il va y chercher les informations qui lui sont nécessaires.

Cette manière de structurer l'interface nous semble plus adaptée à l'usage que nous voulons faire du programme : elle permettra en effet de créer de manière très

souple toutes sortes de leçons sur l'astronomie grâce à la bibliothèque de graphiques et au fichier menu qui pourront s'enrichir à l'infini.

2. Sujets à développer

Voici, pour conclure ce mémoire, une liste de sujets qui pourraient être traités et insérés ultérieurement dans la bibliothèque de graphiques :

- Les saisons : excentricité de l'orbite terrestre et inclinaison sur l'écliptique
- Les lois de Kepler
- Mouvements apparents des corps vus de la Terre
- Les comètes
- Les satellites : une vitesse, une orbite
- Comment se repérer dans l'espace
- Formation d'un cratère sur la Lune
- Apparition du système solaire : Le Big Bang

Annexe

Le passage du repère quelconque au repère associé à l'observateur peut se faire de plusieurs manières grâce à des transformations élémentaires, nous en développerons une qui est composée des transformations suivantes :

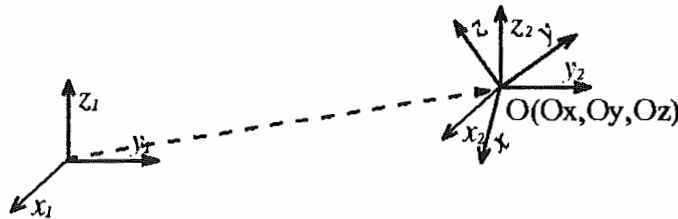
- T : translation vers l'observateur
- R_z : rotation d'axe Z
- R_y : rotation d'axe Y
- E : échange des coordonnées

La première transformation consiste à traduire le repère actuel $(Ox_1y_1z_1)$ à la position de l'observateur pour donner le repère $(Ox_2y_2z_2)$. Cette translation est donnée par la matrice de transformation suivante :

T :

$$\begin{vmatrix} 1 & 0 & 0 & -O_x \\ 0 & 1 & 0 & -O_y \\ 0 & 0 & 1 & -O_z \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

$V(V_x, V_y, V_z)$

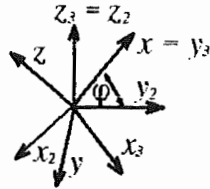


Translation du repère 1 vers le repère 2

La seconde étape consiste à tourner le repère autour de l'axe Z_2 pour aligner X_3 avec la projection sur X_2Y_2 de la direction de vision, car on veut rendre Y_3 colinéaire avec l'axe X du repère observateur. L'angle φ qui intervient dans la rotation est compté positivement de l'axe X vers l'axe Y, ce qui donne la matrice suivante :

R_z

$$\begin{vmatrix} \cos\varphi & \sin\varphi & 0 & 0 \\ -\sin\varphi & \cos\varphi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

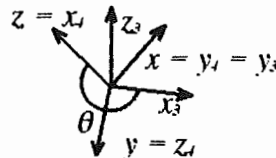


Rotation d'axe Z_2

Le troisième changement de repère est une rotation d'axe Y_3 qui va rendre le futur axe X_4 colinéaire avec la direction de vision. On obtient de la même façon que précédemment la matrice de rotation qui est :

R_y

$$\begin{vmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$



rotation d'axe Y_3

La dernière transformation permute les axes pour qu'ils soient mis comme désiré, c'est à dire que l'axe X du repère observateur soit dans la direction de l'axe X de l'écran et que l'axe Y du repère observateur soit dans la direction de l'axe Y de l'écran. La permutation est:

$$\begin{aligned} X &= Y_5 \\ Y &= Z_5 \\ Z &= X_5 \end{aligned}$$

ce qui donne la matrice suivante E :

$$\begin{vmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

La matrice de changement de repère globale, du repère de départ vers le repère observateur est donc donnée par :

$M = E \cdot R_y \cdot R_z \cdot T$, ce qui donne :

$$\begin{vmatrix} -\sin\varphi & \cos\varphi & 0 & \sin\varphi.Ox - \cos\varphi.Oy \\ -\sin\theta.\cos\varphi & -\sin\theta.\sin\varphi & \cos\theta & \sin\theta.\cos\varphi.Ox + \sin\theta.\sin\varphi.Oy - \cos\theta.Oz \\ \cos\theta.\cos\varphi & \cos\theta.\sin\varphi & \sin\theta & -\cos\theta.\cos\varphi.Ox - \cos\theta.\sin\varphi.Oy - \sin\theta.Oz \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

Bibliographie

1. Astronomie, encyclopédie Atlas du ciel numéros 17, 53, 54 et 55, Editions Atlas, Paris 1984
2. Roger Caratini
L'astronomie, Editions Bordas, Paris 1985
3. Philippe Schweizer
Infographie II, Presses polytechniques romandes
4. Borland
Turbo Pascal 6.0, Reference guide
5. Borland
Turbo Vision