



POLYTECH[®]
SORBONNE



POLYTECH SORBONNE - MAIN4

ALGORITHMIQUE RÉPARTIE

Projet Algorithmie Répartie :
On a marché sur la lune Mars !



Félix CHOI
Antoine DAFLON
Thomas KNIEBIHLER

Encadrant : François PÊCHEUX
(Lip6)

Table des matières

1	Planétarium	2
1.1	Création des Objets	2
1.2	Recréer les planètes	3
1.3	Gestion de la lumière	3
1.4	Décors (Skybox)	3
2	Réseaux	4
3	Voyage Interstellaire	5
3.1	Conception de la Fusée	5
3.2	Transfert orbital de Hohmann	6
3.2.1	Quelle trajectoire suivre?	6
3.2.2	Quand faut-il partir?	6
3.2.3	L'implémentations	7
3.3	Se déplacer dans l'espace	7

Introduction

Afin de mettre en application les notions d'*Algorithmie Répartie* avons vu en cours ce semestre, nous avons choisi de nous investir dans le projet Planétarium. Nous avons donc, en partant du code fourni, proposé une application en réseau permettant de se déplacer dans l'espace. La partie réseau de ce projet est la continuité de la matière *Système d'exploitation* étudié au dernier semestre de MAIN3. Nous avons donc différentes étapes dans notre travail, la partie Réseau gérée par UDP, la partie graphique obtenue grâce à Urho3D et enfin notre voyage interstellaire permis grâce à notre fusée obtenue sous Blender.

1 Planétarium

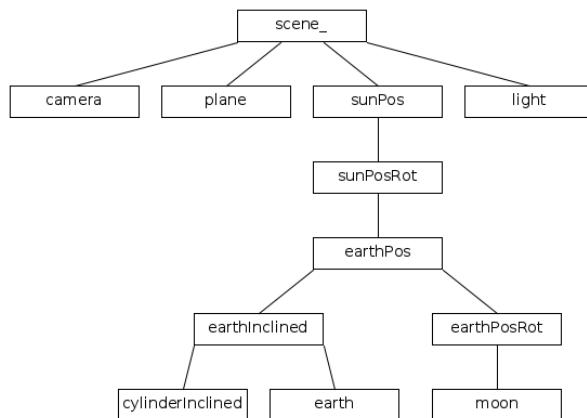
Urho3D est un moteur de jeu C++ permettant la création d'application 2D et 3D. C'est à l'aide de cet outil que nous allons donc créer notre planétarium.



1.1 Création des Objets

Afin de créer un objet sous Urho3D, il faut créer une cascade de Noeud dépendant les uns des autres. Dans notre cas, nous avons donc une scène dans laquelle sont positionnés des sources de lumières, une caméra, un ensemble de planète représentant le système solaire (plus la fusée!).

Le mouvement de chaque planète sera dépendant de la position de ce soleil, on peut considérer le Soleil comme fixe. On peut considérer l'arbre des relations suivant pour un objet, ici pour la Terre, le premier objet fourni.



Les planètes sont créées à partir de 3 noeuds représentant l'inclinaison, la position, et la modélisation. Ces 3 noeuds sont respectivement imbriqués les uns dans les autres. Les planètes sont toutes filles du Soleil. Le même principe est appliqué pour la Lune qui dépend de la Terre.

Les objets sont perçus par leurs modèles et textures. Il est important de ne pas oublier le "Materials" permettant la pose correcte des textures.

Pour la création du planétarium, les proportions des vitesses de révolution des astres sont respectées par rapport à la période de la Terre, le planétarium va beaucoup plus vite que la réalité (un an par révolution terrestre c'est long!). Nous avons essayé de reproduire au mieux les distances et tailles des planètes mais elles ne sont pas tout à fait correctes pour garder un minimum de visibilité. Néanmoins les 4 premières planètes sont à l'échelle par rapport à la Terre. En effet nous obtenions assez vite des distances et des tailles beaucoup trop grandes pour les planètes les plus éloignées, ce qui altérait la compréhension du planétarium. Nous avons donc réalisé un compromis entre réalisme et lisibilité des échelles.

1.2 Recréer les planètes

Afin de créer le visuel d'un objet, on récupère une texture sur Internet (<https://www.solarsystemscope.com/>). On crée ensuite le fichier XML dans le dossier Materials qui fera appel à la technique diff.xml et notre texture (juste un fichier image). Les planètes sont créées à partir de fichiers .mdl qui sont les modèles 3D des objets, ici on utilise des sphères.

Petite astuce : Les anneaux de Saturne ont été créés en aplatisant le modèle de tore disponible. :)

1.3 Gestion de la lumière

L'important pour le planétarium était de créer une carte visible des différentes planètes. Pour cela nous avons placé la source de la lumière au centre de notre planétarium, soit à la place du Soleil. Seulement en faisant cela, le soleil n'était plus du tout visible car il ne recevait plus de lumière. C'est pourquoi nous avons créé six nouvelles sources de lumière permettant d'illuminer cette étoile, et les autres. Nous plaçons ces sources de lumière très proche du soleil afin de simuler une source de lumière unique.

Les planètes éloignées (à partir de Jupiter) ont droit à une source de lumière en face d'elles.

Les sources lumineuses sont toutes des Spotlights.

1.4 Décors (Skybox)

Dans le but de rendre cette visualisation plus réaliste, nous avons mis en place une skybox (procédé graphique permettant de donner l'illusion qu'un espace est plus étendu qu'il ne l'est réellement) représentant l'espace environnant. Cela permet également de mieux visualiser les rotations de la caméra.

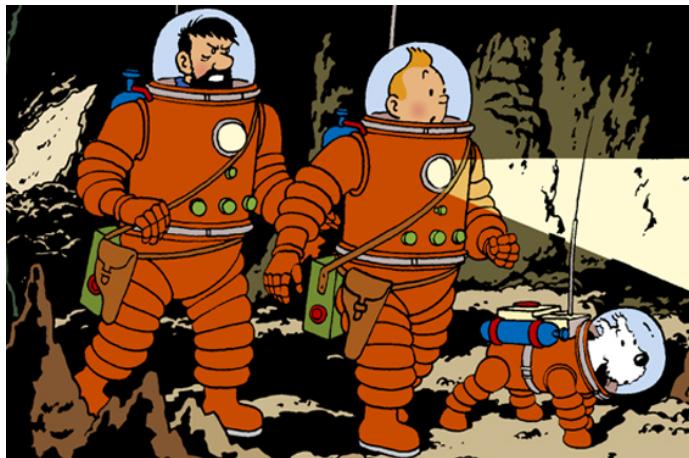
Elles consistent tout simplement à enfermer le planétarium dans une boîte sur laquelle sont intelligemment posés les décors (des images pour faire simple). Une Skybox est une classe existante de Urho3D, il ne faut pas oublier d'inclure le .h nécessaire (il est dans *Urho3D/Graphics*).

La Skybox est une bonne idée si la rotation de la caméra est fixe dans l'espace ; à l'inverse, avec une forte rotation de la caméra, on ressent très vite le mal de mer (ou d'espace!). Nous avons donc ajouter une commande permettant de masquer la Skybox et de garder un fond totalement noir et vide.



Démonstration sur 5 serveurs

2 Réseaux



Afin d'assurer une immersion totale dans notre planétarium, nous avons pour objectif de réaliser cette application en réseau. Nous faisons donc appel au protocole UDP qui assure la transmission de données de manière très simple entre deux entités. On le préfère dans ce projet à TCP que nous avions utilisé l'année dernière car UDP est plus rapide grâce à l'absence des communications supplémentaires permettant de s'assurer l'absence d'erreurs et de pertes. En effet il n'est ici pas conséquent d'avoir des pertes de paquets minimes, et on préfère obtenir une application plus réactive et sans latence.

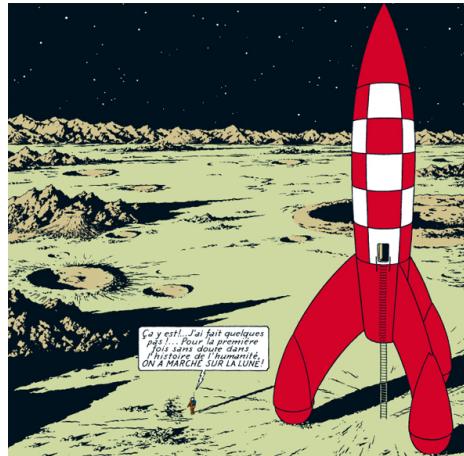
Nous sommes donc dans un modèle client-serveur où le client est l'ordinateur qui dirige la caméra et se déplace dans l'espace. Toutes les informations de ce client sont transmises sur chaque serveur (ici chaque ordinateur) qui sont indépendants et possède le même code, à l'exception de l'orientation de la caméra. En effet afin de se déplacer dans un univers 3D et être immergé dans le planétarium, chaque écran est décalé l'un à l'autre dans l'espace afin d'obtenir un angle de vue très large.

Les déplacements et changements d'orientation de la caméra se font ici par envoi de messages ponctuels, et non pas par pression de touche instantanée. Ainsi, le client doit envoyer des messages à un caractère pour effectuer une action. Le détail des touches disponibles se trouve dans la partie 3.2.

De plus, nous devions faire en sorte que les serveurs démarrent dans un état de pause, afin de pouvoir lancer tous les serveurs de manière synchrone depuis le client.

3 Voyage Interstellaire

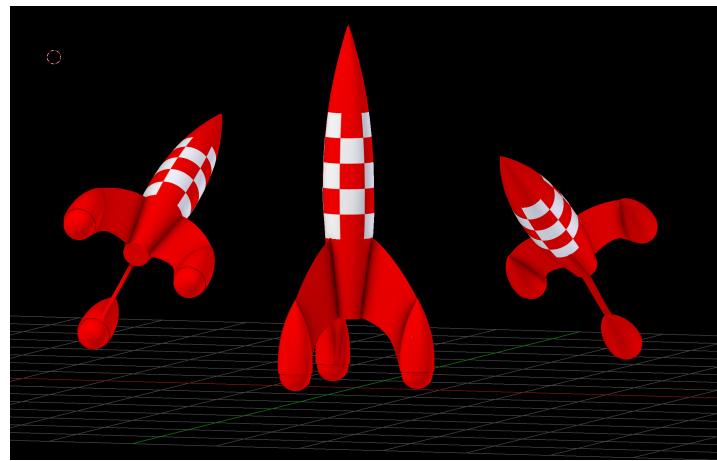
Afin d'optimiser du temps et de l'énergie, Tintin et de nombreux astronautes ont fait appel à l'Orbite de transfert de Hohmann. Ces dernières nous indiquent que la condition requise pour le lancement de notre fusée de la Terre vers Mars est lorsque ces deux planètes sont séparés d'un angle de 44 degrés. La fusée suit ensuite une trajectoire en ellipse pour rejoindre sa destination.



3.1 Conception de la Fusée

Pour voyager dans l'espace, nous devons fournir à nos astronautes un vaisseau. Nous avons voulu donc reprendre l'esthétique de la première fusée à avoir mis les réacteurs sur la lune : celle que Tintin avait emprunté il y a maintenant 65 ans.

Afin de recréer cette fusée nous avons fait appel au logiciel Blender et avons été aidé par de nombreux tutoriels sur *YouTube*. Ce qui n'a finalement pas servi à grand chose car ce dernier était disponible sur le Web Internet.



Fusée expérimentale XFLR6

Le plus dur dans la conception de cet objet a été l'exportation au format *.mdl* de la fusée. En effet Blender ne nous fournissait que des *.obj* nous avons donc dû trouver une extension Blender afin de l'obtenir au bon format. Il existe un add-on trouvé sur GitHub qui permet cela, et nous a sauvé.

3.2 Transfert orbital de Hohmann

Attention ici, c'est du sérieux car le transfert de Hohmann n'est pas une mince affaire. Il est aujourd'hui le plus économique car il n'utilise que 2 accélérations : une au départ pour changer d'orbite et une à l'arrivée pour s'arrêter sur l'orbite de l'objet cible.

On peut séparer le problème en 2 : quelle est la trajectoire à suivre ? Et quand faut-il partir ?

3.2.1 Quelle trajectoire suivre ?

Le but est d'envoyer la fusée sur une orbite reliant l'orbite de la Terre et celle de Mars par les points extrêmes de l'ellipse, ce sont aussi le point de départ et le point d'arrivée.

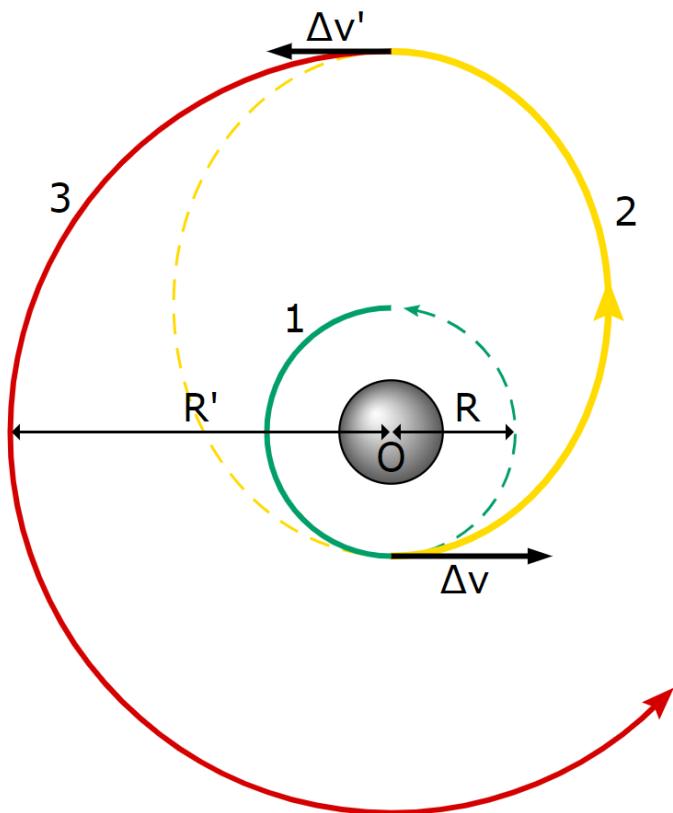


FIGURE 1 – 1 - orbite de la Terre, 2 - orbite de transfert, 3 - orbite de Mars

Par souci d'implémentation, l'orbite de transfert (la 2, en jaune, sur l'image) sera tout simplement un cercle, dont le centre est le centre du segment entre les 2 extréums de l'ellipse, soit de longueur $Terre_Soleil + Soleil_Mars = 2.5$ UA

3.2.2 Quand faut-il partir ?

On considère que la Terre est toujours prête à partir et qu'il faut déterminer la position relative de Mars afin de déterminer le bon moment. La Terre est donc toujours située sur l'extréum de l'ellipse de Hohmann. On utilise ici la 3ème loi de Kepler afin de déterminer la période de révolution de l'orbite de Hohmann.

3ème loi de Kepler :

$$T^2 = a^3$$

ou T est la période de révolution de l'orbite de Hohmann et a le demi-grand axe de l'ellipse de Hohmann.

On sait que :

$$a = \frac{\text{Terre_Soleil} + \text{Soleil_Mars}}{2} = 1.25\text{UA}$$

Ainsi on obtient que :

$$T = 1.4 \times P_t$$

ou P_t est la période la Terre.

On en deduit que le voyage dure environs $0.7 \times P_t$.

On peut alors retracer la position de Mars à $0.7P_t$ du moment où Mars est au 2ème extremum de l'ellipse. Mars a une période d'environ $1.8P_t$. L'angle parcouru pendant le voyage de la fusée est :

$$360 \times \frac{0.7}{1.8} = 136$$

Les valeurs ne sont pas très précises mais il faut parcourir 136 degrés, donc 44 degrés entre Mars et la Terre, avec Mars "devant".

On a ainsi les conditions pour envoyer la fusée sur Mars.

3.2.3 L'implémentations

Pour implémenter l'envoi de la fusée le noeud correspondant au centre de l'orbite de la fusée tourne autour du Soleil de façon à ce qu'il soit toujours à l'opposé de la Terre. Une fois que les conditions de lancements sont vérifiées, le noeud est immobilisé dans l'espace et la fusée peut suivre sa trajectoire qui est fixé. A l'arrivée, la fusée est rattacher au noeud de mars pour la sortir de l'orbite de transfert et rester sur Mars.

3.3 Se déplacer dans l'espace

Afin de pouvoir être totalement immergé dans le planétarium, nous pouvons nous déplacer dans l'espace.

Nous disposons de 8 messages pour déplacer la caméra et l'orienter :

- "z" : Avancer
- "q" : Aller à gauche
- "s" : Reculer
- "d" : Aller à droite

- "o" : Aller en haut
- "k" : Regarder à gauche
- "l" : Aller en bas
- "m" : Regarder à droite

De plus, nous disposons de plusieurs touches afin de fixer la caméra autour de certains objets :

- "S" : Fixation au Soleil
- "t" : Fixation à la Terre
- "f" : Fixation à la fusée
- "j" : Fixation à Jupiter

- "u" : Fixation à Uranus
- "r" : Fixation à un point tournant autour du Soleil
- "y" : c'est la surprise...

Le message "p" permet de mettre en pause le mouvement de tous les objets ou de les reprendre. Lorsque les mouvements sont en pause, on peut néanmoins utiliser tous les mouvements de caméra. Enfin, le message "b" permet de désactiver ou de réactiver l'affichage de la Skybox, par exemple quand la caméra est fixée sur un objet en mouvement, car cela pourrait engendrer des nausées chez certaines personnes.

git : <https://gitlab.com/felixchoi/planetarium>

Conclusion

Le planétarium n'était que le début, nous avons maintenant réalisé un voyage interstellaire et permis à Tintin de rallier Mars sans incident. Nous avons également amélioré le jeu de caméra, et réparti le planétarium sur différentes machines afin de rendre l'expérience la plus immersive possible.

À bientôt pour de nouvelles expériences spatiales !

Longue vie et prospérité