

TER 2009 Groupe J : Mouvement 3D des Sims

KOVALEVA Svetlana

KOUIDER Sarra

LY Paul

RYSER Guillaume

6 février 2009

Résumé

De nos jours, le jeu vidéo s'est rapidement hissé à un rang d'activité majeure. L'économie créée autour de cette pratique a eu des conséquences notables sur notre société. Les statistiques faites récemment montrent la montée flagrante de chiffres de ventes des jeux vidéos cette dernière décennie, comme le montre le graphique 3.

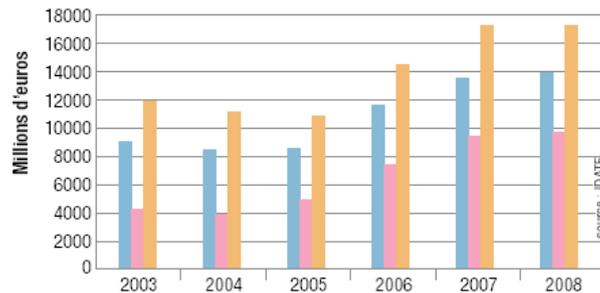


FIG. 1 – Marché mondial du jeu vidéo : Europe en bleu, Japon en rose et USA en orange

Un des problèmes les plus récurrents, surtout depuis l'apparition des jeux multi-joueurs en ligne comme les Sims ou les RPG, est de déplacer un ou plusieurs personnages d'un point à un autre d'une scène 3D sans entrer en collision avec les objets ou les autres acteurs de la scène. Plusieurs opérations entrent en jeu dans le mouvement 3D de personnages :

- la recherche d'un chemin du point de départ au point d'arrivée,
- la coordination individuelle qui permet de déplacer les différentes parties du corps en harmonie,
- la coordination collective qui intervient dès lors que l'on cherche à déplacer un groupe,
- la détection de collisions qui est fondamentale dans un univers solide.

Table des matières

1	Descriptif du sujet	3
1.1	Attentes	3
1.2	Besoins	3
1.3	Contraintes	3
2	Solution envisagée	4
2.1	Solution minimale proposée	4
2.2	Ajouts envisagés	4
3	Organisation du travail	5
3.1	Temps forts	5
3.2	Partage des tâches	5
A	Diagramme de Gantt	6
B	Diagramme des Ressources	6

Table des figures

1	Marché mondial du jeu vidéo : Europe en bleu, Japon en rose et USA en orange	1
2	Diagramme de Gantt	6
3	Diagramme des ressources (bleu = 100%)	6

1 Descriptif du sujet

1.1 Attentes

Ce TER a pour but de nous familiariser avec quelques bases de la programmation 3D, en particulier à travers les points suivants :

- La familiarisation avec un logiciel de modélisation 3D (notre choix s'orientant vers 3D STUDIO MAX)
- Le déplacement d'un ou plusieurs personnages d'un point à l'autre en évitant les obstacles et les collisions (l'un des points cruciaux du développement dans l'industrie du jeu vidéo de nos jours).
- Initiation aux méthodes d'intelligence artificielle liées à la recherche d'un chemin (algorithmes de *Pathfinding*).

Par conséquent, en prenant en considération les contraintes du temps, il faudra que notre production finale comporte au minimum deux aspects importants :

- La prise en main de la modélisation 3D, à travers la génération des cartes 3D via OPENGL et la création des personnages avec 3D STUDIO MAX ;
- L'utilisation non triviale d'algorithmes de *Pathfinding* et de tout ce qui peut être lié au problème de déplacement de personnages, au départ en envisageant le déplacement d'un seul personnage à la fois avec l'algorithme A*, puis par la suite, en fonction de notre avancement, l'utilisation de SMA* pour un mouvement collectif et la résolution du problème de flocking.

1.2 Besoins

Les cours de l'UE Algorithmique de l'IA nous fournira les algorithmes de déplacement que nous devrions utiliser, et donc une grosse partie du moteur de déplacement, cependant, l'aspect tri-dimensionnel n'étant traité qu'en Master 2, nous devons donc apprendre à utiliser 3D STUDIO MAX et OPENGL, ce qui nous prendra beaucoup de temps en début de projet, lors de la phase d'analyse et de recherche autour du sujet.

Nous devons aussi trouver une solution pour l'interfaçage entre notre moteur de déplacement, que nous développerons en C++, et les formats de sortie de OPENGL et 3D STUDIO MAX. Nous envisageons pour le moment 2 solutions : la première serait de passer par le XML pour stocker les données de façon standardisées, et la seconde serait d'utiliser la SDL fournie par C++. De ces deux solutions nous prendrons la plus pratique, ce choix se fera dans peu de temps, en fonction des documentations trouvées et étudiées.

1.3 Contraintes

Contraintes de temps

Nous disposons de 3 mois pour implémenter et produire les différentes fonctionnalités, en plus du travail attendant aux autres UE. Nous avons donc compté 2 mois pour obtenir une version minimale valable, répondant au minimum des attentes.

Contraintes pour le projet

Nous devons avoir recours à un logiciel de modélisation 3D pour la création des personnages, à choisir entre 3D STUDIO MAX, Maya ou Blender. Il nous faut de plus une librairie graphique pour la création des scènes (OPENGL, DirectX, ...).

Pour la gestion du *Pathfinding*, il nous faut utiliser les algorithmes de recherche d'un chemin pour aller d'un point à un autre. Il nous faudra envisager la coordination individuelle permettant le déplacement des différentes parties du corps du personnage en harmonie, une coordination collective lors du déplacement d'un groupe de personnage, la détection de collisions, fondamentale dans un univers solide, et enfin, envisager la création d'une interface utilisateur, permettant à toute personne extérieure à notre groupe de pouvoir manipuler le plus simplement possible les différents composants de la production (chargement de scènes, placement initial des personnages à des emplacements valides, ou déplacement automatique d'un personnage ou d'un groupe de personnages en fixant un point cible.

2 Solution envisagée

2.1 Solution minimale proposée

En solution minimale, nous proposons de nous axer sur le *Pathfinding* et la prise en main de la modélisation 3D, avec les points suivants :

- 2 cartes 3D préchargées sans texture ni relief du sol générées via OPENGL;
- 4 personnages sans texture sur la carte générés via 3D STUDIO MAX;
- 1 seul personnage bouge à la fois, évitant les obstacles, et les autres personnages;
- déplacement traité par A*, avec comme fonction d'approximation : la distance euclidienne.

Il est évident que cette solution n'est pas optimale, cependant, nous pensons qu'il est possible dans le temps imparti de rendre cette production. Dans le doute, nous présentons une solution minimale très restreinte, en imaginant d'hors et déjà des ajouts, intégrés en fonction du temps et du bon fonctionnement de ceux-ci.

Notons que la distance utilisée ne sera appréciable que s'il n'apparaît pas de dénivelés sur la carte, sans quoi il nous faudra utiliser la distance topologique.

2.2 Ajouts envisagés

Voici une liste non exhaustive et non triée des ajouts que nous avons envisagés et que nous pensons réalisables.

Mouvement simple de plusieurs personnages

On négligera dans un premier temps les risques de collision ou traversement entre les personnages en mouvement. Cette solution nous fera implémenter SMA* et sera orientée *Pathfinding*.

Flocking lors du déplacement d'un seul personnage

En utilisant A*, nous traiterons avec cette extension les problèmes de contournement d'un groupe de personnage statique. Le fait d'utiliser A* bloquera l'utilisation de cette extension au déplacement d'un seul personnage, sans quoi les ressources demandées seraient trop importantes.

Flocking général

Dans la même optique que la proposition précédente, mais cette fois en utilisant SMA*, nous proposons ici une version plus optimisée de déplacement, qui prendra en compte les groupes de personnages pour les déplacements, et les éclatera temporairement pour permettre le passage d'un personnage en mouvement, quel qu'il soit.

Graphisme généreux pour les personnages et/ou la carte

Cette extension propose un rendu visuel plus important, et donc plus ludique, orientant notre travail davantage sur la conception 3D et le traitement de données. En fonction du temps et de l'avancement de notre projet, nous ajouterons des textures aux personnages, aux cartes, ou aux deux.

Integration de cartes complexes

Cette extension peut jumeller l'aspect 3D avec une orientation *Pathfinding* du projet, en ajoutant du relief aux cartes, et donc en adaptant la fonction d'approximation des distances aux distances topologiques, et non plus euclidiennes.

Autres extensions

En dernier points, si nous avons le temps, nous envisagerons l'implémentation de constructeurs de scènes et/ou de personnages, orientant le projet vers la 3D. Ceux-ci devraient nous permettre de rendre un logiciel complet, incluant des constructeurs graphiques, et un moteur de déplacement correct.

3 Organisation du travail

3.1 Temps forts

On peut voir dans le diagramme de Gantt A que plusieurs tâches ne sont pas encore attribuées. Cela vient du fait que ce sont les tâches les plus importantes et que leur répartition ne peut et ne doit pas être faite à la légère, ainsi nous attendons d'avoir terminé les recherches sur le sujet pour les répartir par binômes. Voici une synthèse des tâches non attribuées :

Implémentation scène + Tests

Il va de soit que le binôme qui a travaillé sur l'implémentation des scènes se devra de faire les tests dessus. Cette tâche consiste à maîtriser `OPENGL` et à implémenter les deux cartes de la version minimale (ou plus, selon les besoins) et à trouver un format d'export des données satisfaisant pour une utilisation ultérieure.

Implémentation moteur de déplacement + Tests

De même que précédemment, il faudra tester les algorithmes afin d'y trouver la moindre faille, tout en faisant en sorte de définir au plus vite une structure de donnée pour le graphe concordant avec les données exportées par `OPENGL`. L'implémentation en `C++` ne devrait pas prendre beaucoup de temps, mais les tests et l'optimisation seront théoriquement plus longs.

Interfaçage Moteur / Scene 3D

Cette tâche doit consister à mettre en commun les deux tâches précédentes, et à optimiser le transfert de la carte dans l'analyseur de déplacement, et éventuellement à retoucher le code de l'analyseur en cas de problème.

Interfaçage pour l'utilisateur

Surement en utilisant `SDL`, il faudra, dans cette tâche, récupérer la sortie de l'analyseur, et trouver un moyen de faire un rendu visuel, afin de voir les déplacements des personnages dans l'environnement. Ce sera aussi au cours de cette tâche que nous établirons les libertés laissées à l'utilisateur quand à la sélection des personnages à déplacer, leur placement initial,...

3.2 Partage des tâches

Nous avons fait en sorte que tous les membres du groupe aient un rôle dans le développement du projet équirépartis dans le temps. Si nous nous tenons au diagramme de ressources B, personne ne devrait avoir de surcharge de travail par rapport aux autres, sauf en fin de projet, où nous devons tous travailler d'arrache pied pour le mener à bien.

De plus, il a été défini que, en cas de problème lors d'une tâche (retard, difficultés, ...), des réunions extraordinaires du groupe seraient organisées, et si le problème n'est pas résolu, nous demanderons de l'aide extérieure. Le but étant de travailler le plus possible en autonomie, tout en développant nos acquis et en partageant nos connaissances.

Une réunion hebdomadaire est prévue, et d'autres seront ajoutées en fonction des besoins et des disponibilités de tous. Entre temps, une communication permanente sur l'avancement des tâches par email nous permettra de savoir où en sont l'intégralité des tâches, et ainsi, nous aurons tous un aperçu de l'avancement du projet en général.

A Diagramme de Gantt

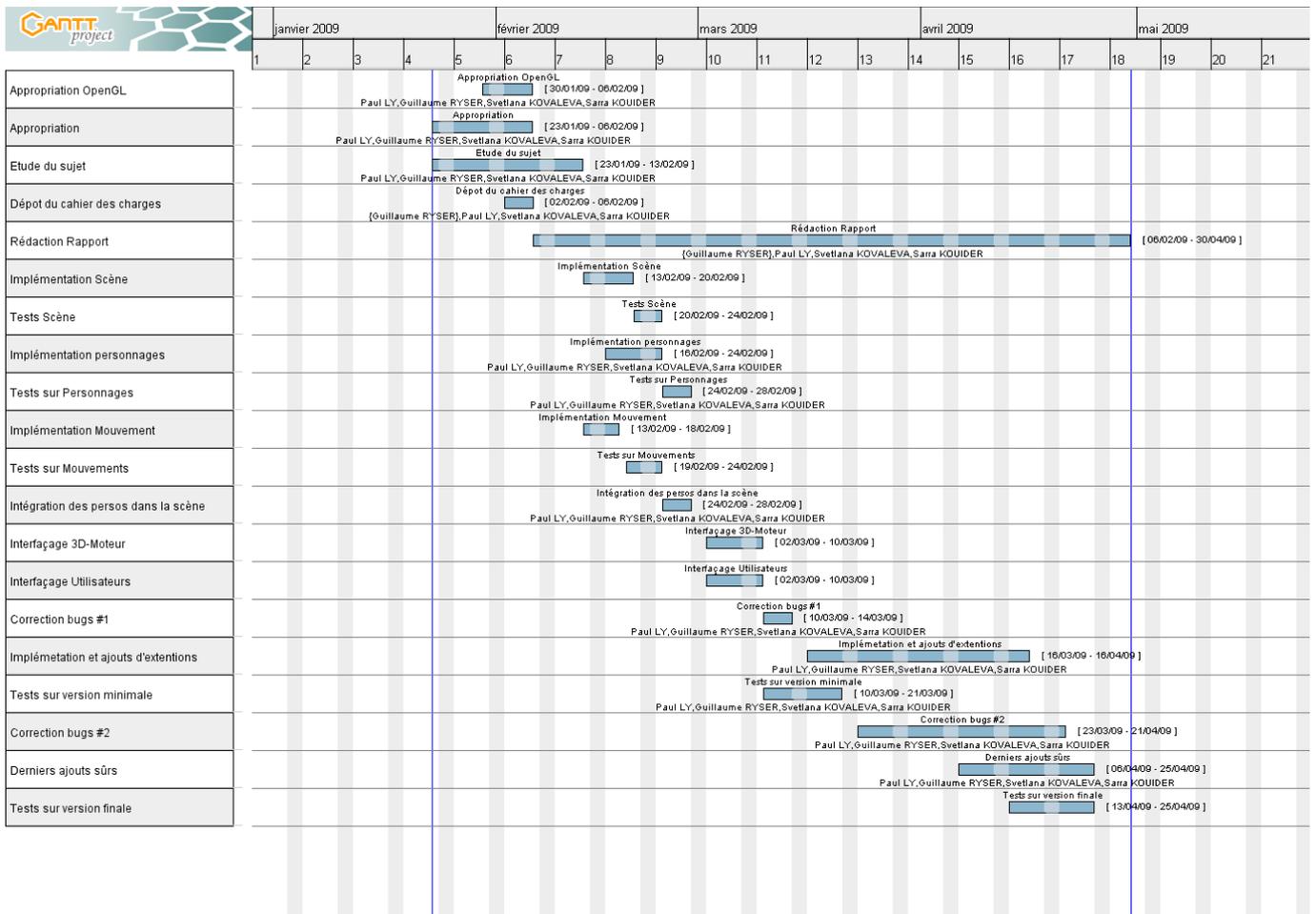


FIG. 2 – Diagramme de Gantt

B Diagramme des Ressources

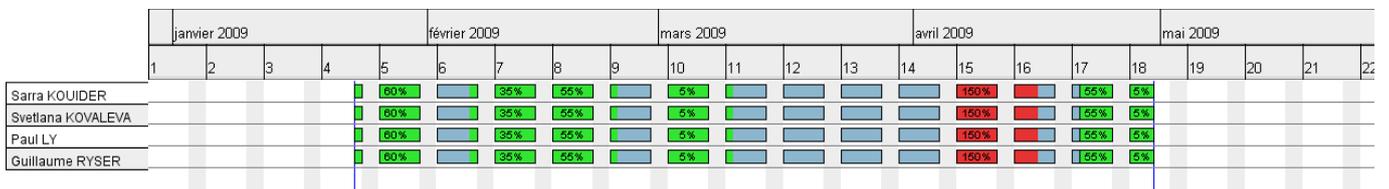


FIG. 3 – Diagramme des ressources (bleu = 100%)