

Navigation dans un environnement 3D

Marc BEYSECKER, Tom GIMENEZ, Valentin HIRSON, Léo RIZZON

Université Montpellier II

Master 1 Informatique

Table des matières

Chapitre 1

Remerciements

Ma mère, ta mère. Merci mon cul bonsoir messieurs dames.

Chapitre 2

Introduction

Contexte

Dans les jeux vidéos, les personnages non joués par des humains doivent pouvoir se déplacer de manière autonome et cohérente. Un environnement 3D est constitué d'un graphe énorme avec des milliers de sommets. Même en ne prenant que le sol, le graphe est encore très gros et, surtout, n'est pas seulement constitué des points naviguables. C'est à dire que le personnage ne doit pas pouvoir se déplacer dessus.

Comme nous l'avons étudié, la recherche de chemins dans un graphe est un problème classique mais qui peut s'avérer très lourd sur de gros graphes. En prenant en compte que les jeux mettent en scène un grand nombre d'agents il s'agit de minimiser les temps de calculs.

Dans le cadre de notre unité d'enseignement intitulée Algorithmes de l'Intelligence Artificielle, nous avons réalisé un projet qui consiste en .

Objectifs

L'objectif de ce projet était, d'une part, la compréhension poussée d'un algorithme d'exclusion mutuelle et l'étude d'un système tolérant aux pannes. Toutefois, il s'agissait d'assimiler les problèmes et défauts qu'il comportait pour bien appréhender l'amélioration proposée.

D'autre part, l'implémentation de ces modèles pour appliquer toute la théorie à un exemple concret.

Notons également que l'étude s'est faite à partir d'un article en anglais, comme la majorité des articles scientifiques, ce qui constitue une expérience intéressante.

L'implémentation consiste donc en l'application de l'algorithme de à un certain nombre de sites, en réseaux, qui cherchent à accéder à une même ressource alors qu'il n'y qu'un accès possible à la fois.

Chapitre 3

Recherches préliminaires

3.1 Une scène en 3D avec Blender

3.2 Récupérer notre scène en 3D

3.2.1 OpenSceneGraph

3.2.2 OpenGL avec nos propres structures de données

Chapitre 4

Implémentation finale

4.1 Un parser pour récupérer notre scène

4.2 Nos structures de données

4.2.1 La scène

4.2.2 Le graphe généré

4.3 Travaux effectués sur le graphe

4.3.1 Création du graphe des way-points

Parcours

Heuristique

4.3.2 Simplification du graphe obtenu : merging

Chapitre 5

Conclusion

Bilan technique

Nous avons réussi à implémenter tous les algorithmes demandés. Tous les mécanismes de l'amélioration de l'algorithme de sont opérationnels. Nous avons effectué une solide batterie de tests en essayant de rencontrer un maximum de configurations possibles.

Toutefois on pourra noter que notre code n'est pas parfait, nos choix se portant parfois d'avantage sur la simplicité que sur l'élégance du code.

Bilan personnel

Ce projet a été bien plus important que nous ne l'imaginions au premier abord, tant en quantité de travail que sur le nombre et la diversité des domaines sur lesquels nous avons travaillé.

Premièrement ces algorithmes, forts connus, nous ont plongés au coeur de l'algorithmie distribuée nous obligeant à en bien comprendre les principes, en nous confrontant à bon nombre de problèmes classiques. L'amélioration de l'algorithme de est un modèle assez complexe qui n'a pas été facile à bien cerné. Pourtant, le travail terminé, nous avons réellement la sensation d'avoir progressé dans cette discipline.

Il est clair que notre expérience de développement, bien que n'étant pas la première, est toujours aussi importante et nous a familiarisé de nouveau avec le langage C, le réseau, le multi-threading.

Enfin ce projet a été l'occasion de travailler en un groupe qui ne l'avait jamais été. Une excellente surprise que nous tenterons de renouveler à l'avenir. Une équipe complémentaire et équilibrée qui a permis d'arriver au bout de nos objectifs.