



Projet de synthèse d'images

Thomas is not alone

Imac première année

Abstract

L'objectif de ce projet est de réaliser un petit jeu temps réel permettant d'exploiter vos nouvelles connaissances en OpenGL. Vous travaillerez par binômes.

1 Introduction

Il s'agit de vous approprier le jeu *Thomas was alone*, un jeu très simple et pourtant ludique. Le but du jeu est de faire traverser un ensemble de rectangles mobiles un parcours qu'ils seraient incapables de franchir seuls.



1.1 Déroulement

Le début du parcours est défini par les emplacements des rectangles “personnages” que nous allons déplacer. Chacun d’eux a une taille différente. A chaque instant, un des rectangles est sélectionné et la vue du jeu est (relativement) centrée sur ce rectangle. On peut le déplacer ou changer de rectangle à l’aide des touches du clavier. Le but du jeu est de pouvoir déplacer l’ensemble des rectangles “personnages” pour les amener à leur position finale, matérialisée par un ensemble de rectangles blancs de même taille que les rectangles mobiles.



2 Les entités du jeu

2.1 Les rectangles

Les rectangles sont définis par leur attributs :

- taille
- couleur
- position de départ
- position d’arrivée
- puissance de saut
- ...

Tous les rectangles se manipulent de la même manière :

- droite
- gauche
- saut vers le haut, orientable à droite et gauche pendant le saut

2.2 La carte

La carte définie par un ensemble de rectangles statiques et éventuellement par une identité visuelle.



2.3 Les niveaux

Un niveau est défini par un fichier contenant :

- l'ensemble des rectangles statiques formant la carte
- le nombre et les attributs des rectangles personnages
- d'autres paramètres optionnels que vous souhaiterez peut-être ajouter

2.4 Les collisions

Une des principales difficulté de ce projet est de gérer les collisions entre rectangles. Vous proposerez la solution que vous trouverez la plus adaptée.

3 Options

Ce projet est très libre dans la façon que vous aurez de le concevoir, mais aussi dans les options que vous voudrez y apporter. Vous pouvez par exemple :

- faire une carte dynamique (background ou rectangles supports)
- rajouter du son
- donner une identité visuelle (textures ou autres)
- ...

Par contre, veuillez à ne pas commencer à rajouter d'options tant qu'une version basique et jouable du jeu n'est pas réalisée.

4 Le programme

Votre programme sera codé en langage C et devra compiler dans les salles machines de l'université. Il sera composé de plusieurs fichiers .c et .h compilés à l'aide d'un Makefile. Il devra compiler sans warning (mais avec l'option `-Wall`). Vous rendrez votre projet sous forme d'archive nommée `nom1_nom2_radial.tgz`. Cette archive doit générer un répertoire nommé `nom1_nom2_radial/` contenant votre projet. Votre programme ne comportera aucun `scanf`, les paramètres nécessaires au fonctionnement du programme (s'il y en a) seront transmis en ligne de commande (cf. `argc` et `argv`).

5 Le rapport

Vous fournirez en version papier et en version électronique un rapport de 10 pages maximum. **Consacrez suffisamment de temps au rapport car il représente une bonne partie de la note finale.** Essayez de respecter au mieux les directives suivantes :

- Ne perdez pas de temps à réexpliquer le sujet du projet, l'enseignant le connaît déjà, faites seulement un bref résumé de quelques lignes. De manière plus générale, ne détaillez pas des méthodes déjà expliquées dans l'énoncé à moins que vous les ayez modifiées.

- Un rapport sert surtout à montrer comment vous avez fait face aux problèmes (d'ordre algorithmique). Certains problèmes sont connus (on en parle dans l'énoncé), d'autres sont imprévus. Montrez que vous les avez remarqués et compris. Donnez la liste des solutions à ce problème et indiquez votre choix. Justifiez votre choix (vous avez le droit de dire que c'est la méthode la plus facile à coder). A savoir : "on a eu un super problème par ce que ça compilait pas ..." ne nous intéresse pas.
- Il ne doit figurer aucune ligne de code dans votre rapport. Un rapport n'est pas un listing de votre programme où vous détaillez chaque fonction. Vous devez par contre détailler vos structures de données et mettre du pseudocode pour expliquer vos choix algorithmiques. Il est autorisé d'utiliser des "raccourcis" tels que "**initialiser le tableau `tab` à 0**" plutôt que de détailler la boucle faisant la même chose.
- n'hésitez pas à mettre des images dans votre rapport pour illustrer vos propos et vos résultats.
- Enfin, il est **très important** de faire la liste de ce que vous avez fait, de ce qui fonctionne correctement et la liste des dysfonctionnements. Précisez quand il s'agit d'options que vous avez rajoutées en plus de ce qui était demandé.

6 Pour finir

Vous pouvez laisser libre cours à votre imagination, toute amélioration sera la bienvenue.

Bon courage!