LANGUILLE Antoine

MICHELI Sébastien

Rapport de projet : TSI

Introduction :

Ce projet a pour but de nous faire manipuler les fonctionnalités d’OpenGL à travers la réalisation d’un jeu. Nous avons décidé de nous inspirer du jeu Geometry Dash : le but du jeu est d’aller au bout d’un niveau en sautant pour éviter les obstacles. Votre personnage (un cube) se déplace tout seul et il faut le faire sauter pour passer au-dessus des différents obstacles sur votre route.

Pour lancer le jeu, il suffit d’exécuter le programme *main*. Si vous êtes sous Linux, entré « y » pour ouvrir la fenêtre du jeu, sinon sous Windows entré « n ».

Commandes :

Touches dans l’écran titre :

- s : lance la sélection du niveau

- echap : ferme le jeu

Touches dans la sélection du niveau :

- flèche > : sélectionne le niveau suivant

- flèche < : sélectionne le niveau précédent

- entrée : lance le niveau

- echap : ferme le jeu

Touches durant la partie :

- barre espace : fait sauter le cube

- echap : ferme le jeu

Structure du code :

Le code est composé de différents programme permettant chacun d’effectuer une partie précise du jeu. Nous allons donc vous expliquer les parties importantes de chacun d’entre eux.

Programme *main* :

Il permet de lancer le jeu en appelant tous les autres programmes et en définissant les paramètres dont ils ont besoin.

Programme *background* :

On définit le fond d’écran de la zone jouable. Pour cela on définit la taille qu’elle doit occuper ainsi que l’image à mettre sur cette zone.

(Image fond d’écran en jeu)

Programme *cpe3d* :

Programme *glutils* :

Ce programme comporte des fonctions utilisées dans d’autres programme. Tout d’abord il permet la compilation des shaders.

(Image fonction compile\_shader)

Ensuite on crée un programme GPU.

Programme *level* :

Ce programme comporte 2 classes : une pour gérer une potentielle erreur lors de l’exécution, et l’autre pour définir les fonctions utiles à l’exécution du jeu. On va décrire ici ces fonctions.

Dans un premier temps nous avons la fonction *\_init\_* permettant l’initialisation du niveau en définissant la caméra et une liste d’obstacles (pour l’instant vide).

La fonction suivante *add\_element* permet de rendre visible un obstacle et l’ajoute à la liste des obstacles.

Programme *mesh* :

Programme *obstacles* :

On définit une classe pour chaque type d’obstacles. La première classe correspond aux pics qui vont être les principaux obstacles. La fonction contient toutes les données nécessaires pour les placés durant le jeu. Elles sont toutes identiques hormis sur leur orientation qui pourra varier.

Programme *player* :

Cette partie est dédiée aux fonctionnalités rattachées au joueur. La première fonction sert à placer le joueur au début du niveau.

La fonction suivante *set\_terrain* va lancer la création du terrain du niveau.

La fonction *tick\_clock*

La fonction death est la fonction appelée lorsque le joueur a perdu. On affiche « You died » et on replace le joueur au début du niveau.

La fonction *step* définit la vitesse de déplacement du joueur.

La fonction *jump* fait sauter le personnage lorsque l’on appuie sur …

La fonction longjump permet

La fonction test\_collisions

Programme *reflist* :

Programme *viewerGL* :

Conclusion :

Récap de ce qu’on a fait.