

Pierre-Luc Handfield et Mathieu Narbonne

Graphisme par ordinateur

SIF1032

Projet de session (phase 1)

Travail présenté à

François Meunier

Département de mathématiques et informatique

9 mars 2015

DESCRIPTION DE LA SCÈNE

La scène du projet représente un jeu d'Arkanoid. Il s'agit d'une version 3D d'un jeu populaire développé par la compagnie Taito en 1986. Ce jeu consiste en un tableau de jeu contenant des briques de différentes couleurs, une balle qui brise ces briques et d'un « vaisseau », représenté par un parallélépipède rectangle, qui peut être bougé de manière à faire rebondir la balle et la garder en jeu. Le but du jeu est de briser toutes les briques pour passer au prochain niveau. Le joueur a trois vies au départ. Pour chaque balle qui sort de la zone de jeu, le joueur perd une vie. Quand le joueur est à court de vies, le jeu recommence au début.

Contrôles

Touche	Fonction
Échap	Ferme le jeu
A	Bouge le vaisseau à gauche
D	Bouge le vaisseau à droite
W	Zoom in sur la scène
S	Zoom out sur la scène
Flèche gauche	Translation de la caméra vers la gauche
Flèche droite	Translation de la caméra vers la droite
Flèche haut	Rotation de la caméra sur l'axe des x
Flèche bas	Rotation de la caméra sur l'axe des x
Pavé numérique 1*	Changer la vitesse de la balle (lente)
Pavé numérique 2*	Changer la vitesse de la balle (moyenne)
Pavé numérique 3*	Changer la vitesse de la balle (rapide)

* Les touches qui influencent la vitesse de la balle sont utilisées uniquement à des fins de tests. Dans la réalité, le joueur n'a pas de contrôle direct sur la vitesse de la balle.

CLASSES PRINCIPALES

Game1.cs

Classe principale du jeu. Elle contient les méthodes de XNA `Initialize()`, `LoadContent()`, `Update()` et `Draw()`. Elle permet de gérer les différentes facettes du jeu (afficher les modèles, vérifier les collisions, gérer les vies du joueur, vérifier si la balle est hors limite, etc).

Ball.cs

Cette classe représente la balle. Elle contient les coordonnées de la balle, sa vitesse et des méthodes qui permettent de déterminer la trajectoire de la balle dans le jeu.

Brick.cs

Classe abstraite qui permet de définir des éléments communs à toutes les briques.

BlueBrick.cs et RedBrick.cs

Représente une brique bleue et rouge. Elle contient le modèle, la position et la texture de la brique.

BrickController.cs

Contient toutes les briques du jeu. Son rôle est de contrôler le "layout" du niveau. Il crée l'animation des briques en début de jeu.

LevelLayout.cs et level.cs

Permet de définir le placement des briques dans le niveau. La position des briques est définie dans le fichier texte "Levels.txt".

Paddle.cs

Représente le "vaisseau". Il bouge de gauche à droite pour empêcher la balle de sortir des limites. Cette classe contient son modèle et sa position.

Wall.cs

Représente un des trois murs de la scène (gauche, droit, fond). Contient le modèle et la texture.

CRÉATEUR DE NIVEAU (EN DÉVELOPPEMENT)

Il existe un outil pour créer d'autres niveaux dans le jeu. Cet outil est programmé dans Excel avec Visual Basic. Le fichier se nomme "Level Generator.xlsm". Il faut créer une copie d'une feuille de niveau (Level1, Level2, etc.). Il faut ensuite dessiner le niveau et appuyer sur CTRL + Q. Cette dernière action créera une chaîne de caractères en B33 qui pourra être collée dans le fichier "Levels.txt" ce qui aura pour effet de créer un nouveau niveau.

AMÉLIORATIONS FUTURES

Les éléments suivants pourront être développés dans le futur :

- Ajouter des "powerups" qui aiderais le joueur (ex. ralentir la balle, allonger le vaisseau, pouvoir tirer des projectiles sur les briques, etc.).
- Ajouter d'autres niveaux.
- Ajouter d'autres couleurs de briques.