

Université De Montpellier

L2 INFORMATIQUE

HLIN302
Projet noté « Casse-Briques »

Rapport de projet – HLIN302

Étudiants :

- BOYAN BECHEV
- KEZZOUL MASSILI
- YASHVEER TEELUCK
- ROMAIN FOURNIER

Année :2018-2019

introduction

Dans le cadre du module HLIN302 de L2 au premier semestre, Nous avons développé le jeu « Casse-Briques ». Notre groupe était composé de BOYAN BECHEV, KEZZOUL MASSILI, YASHVEER TEELUCK et ROMAIN FOURNIER.

Le travail fut repartit de sorte que chacun avait une tache spécifique à accomplir afin de compléter à temps le projet. La réalisation du jeu a nécessité environ 8 semaines

Analyses

L'approche la plus efficace afin de programmer convenablement le jeu était de créer plusieurs classes. Chacune des classes définissait des différents paramètres qui interagissaient les un avec les autres afin de définir les différentes composantes du jeu.

Notamment sept grandes classes :

- Terrain
- Ball
- Brick
- raquette
- player
- score
- sauvegarde

Les différentes classes

Ci dessous nous allons décrire les différentes classes ,leur fonctions, et les algorithmes impliqués afin de faire fonctionner convenablement le jeu.

La classe « terrain »

Cette classe définit le terrain ou seront confinés la balle, la raquette et les briques. La couleur du cadre et du fond sont aussi choisis avant de débuter le jeu. La taille du terrain limite le mouvement de la raquette et de la balle seulement à l'intérieur d'elle.

La classe « ball »

Elle définit la position et la vitesse initiale de la balle dans le terrain(fait par le constructeur). Cette classe gère aussi les collisions de la balle avec les briques, la raquette et les bordures du terrain. Nous avons défini notamment 4 types de collisions :

1. haut(1)
2. bas(3)
3. droite(2)
4. gauche(4)

Le sinus et le cosinus des angles de collisions permettent de calculer la vitesse selon l'axe des x et l'axe des y. Ainsi l'algorithme retourne 5 directions possibles dépendant du type de collision(0=pas de collision)

La classe « brick »

Cette classe a de nombreuses fonctions essentielles :

- Elle choisi les briques en fonction du niveau atteint par le joueur
- Elle définit la résistance des briques dépendant du niveau
- Elle attribue un nombre de point plus ou moins important par rapport à la résistance des briques.
- Elle donne la position des briques, la dimension des briques , et aussi le nombre de briques présentes sur le terrain en temps réel.

La classe « raquette »

Elle a 2 fonctions principales qui sont :

- Gérer la taille de la raquette
- Définir Le mouvement selon l'axe des x qui fait bouger la raquette à droite (les x positifs) et à gauche(les x négatifs)

La classe « player »

Son rôle est de gérer et d'afficher les détails du joueur au cour de son évolution dans le jeu, notamment :

- Le nom du joueur
- Le nombre de balles restant(afficher comme « vie »)
- Le score(afficher comme « point »)
- Le niveau du joueur(afficher comme « niveau »)

La classe « score »

Elle est utilisée pour la lecture des fichiers qui stocke les meilleurs scores et trie les scores précédents en ordre décroissant.

La classe « sauvegarde »

Cette dernière classe définit le chemin et le nom des fichiers qui stockent les parties précédentes afin de les charger

Le menu du jeu

Cette fonction affiche le menu principal avant le début de la partie et propose les options suivantes :

- Jouer
- Charger (permet de charger une sauvegarde ou de la supprimer)
- Options (Permet de paramétrer le jeu)
- Score (Affiche les scores sauvegardés précédemment)
- Quitter

Les différentes étapes pour la réalisation du projet

- Tout d'abord nous nous sommes concertés afin de choisir le nombre de classes nécessaires pour faire fonctionner le jeu convenablement.
- Nous avons ensuite défini spécifiquement le rôle de chaque classe au sein du programme et comment elles allaient interagir les unes avec les autres

- Nous avons d 'abord commencé à coder les classes les plus compliquées , notamment celles qui gèrent les propriétés physiques du jeu(La classe Ball,Raquette...)
- nous compilions et exécutions le programme a chaque fois qu'on avait finit de rédiger une classe afin de détecter d'éventuel « bug »
- Une fois toutes les classes complétées , et tout les bugs corrigés, Nous avons testé le jeu au complet pour vérifier sa fonctionnalité.
- Dernièrement la dernière partie du projet consistait à rédiger un rapport synthétisant toutes les étapes .

Sommaire des Annexes :

annexe 1 : introduction.....	2
annexe 2 : Analyses.....	2
annexe 3 : La classe « terrain ».....	3
annexe 4 : La classe « ball ».....	3
annexe 5 :La classe « brick».....	4
annexe 6 :La classe « raquette ».....	4
annexe 7 :La classe « player ».....	4
annexe 8 : Le menu	5
annexe 9 :La classe « score ».....	5
annexe 10 :La classe « sauvegarde».....	5
annexe 11 : Les étapes.....	5-6