

DUMAN Louis-David
CROSNIER Jérémie
ROUSSEAU Louis
FONTAINE Nael

Rapport de PF2

Groupe L12

Table des matières :

Rapport de PF2.....	0
Introduction.....	2
1 - Liste des fichiers rendus.....	2
2 - Choix de Conception.....	2
3 - Regard Critique.....	2
Conclusion.....	2

Introduction

Newtonoid est un projet dont le but est de combiner un jeu de casse-brique (comme Arkanoid), avec des notions physiques issues de la mécanique newtonienne.

Ce projet met l'accent sur la conception logicielle via l'abstraction et la modularité de ses composants, une certaine efficacité algorithmique et l'utilisation de structures de données adaptées (quadtree).

1 - Liste des fichiers rendus

/bin : dossier contenant le programme principal

 newtonoid.ml : fonctions principales du jeu

/lib : dossier contenant les différents éléments implantés en vue du développement du jeu

 ball.ml : définition d'un type ball et de fonctions l'utilisant

 brick.ml : définition d'un type brick et de fonctions l'utilisant

 collision.ml : gestion des collisions

 paddle.ml : définition d'un type paddle et de fonctions l'utilisant

 input.ml : gestion de la récupération du déplacement de la souris

 iterator.ml : définition du type iterator

 level.ml : gestion de la création du niveau

 quadtree.ml : définition d'un type quadtree et de fonctions l'utilisant

 tests.ml : fichier contenant les tests pour la collision et le quadtree

2 - Choix de Conception

2.1 - Objets du Jeu

Le type balle (ball) est caractérisé par ses coordonnées x, y (centre), sa vitesse vx, vy et son rayon.

Le type raquette (paddle) est caractérisé par ses coordonnées x, y (coin haut gauche), sa hauteur et sa largeur, et sa vitesse horizontale.

Le type brique (brick) est caractérisé par ses coordonnées x, y (coin haut gauche), sa hauteur et sa largeur, ainsi que sa valeur.

On représente l'état du système par une balle, une raquette, l'ensemble des briques, un booléen (running) qui représente si la balle bouge, le score actuel, et le nombre de vies.

Le programme associé au level est présent afin de créer des niveaux pour notre newtonoid

2.2 - Utilitaires du jeu

Le fichier config est présent afin de pouvoir centraliser des constantes du problème, par exemple les dimensions de l'écran ou les positions initiales de la balle et du paddle.

Le type Quadtree permet d'implanter la structure quadtree représentant l'ensemble des briques du niveau actuel. Celui-ci permet notamment d'optimiser le jeu en ne vérifiant les collisions qu'avec les briques proches de la balle. On définit en amont le type border qui représente une surface rectangulaire permettant alors de définir la surface associée à un nœud. Ainsi l'arbre représentant le Quadtree est soit un arbre vide associé à une border, soit une feuille associé à une border et les briques présentes dans cette surface, soit un noeud avec des sous arbres sud est, sud ouest, nord est, nord ouest, lorsque l'écran de jeu est découpé en quatre, associé à un borders. On pose d'ailleurs max_in_leaf la valeur maximum que l'on souhaite avoir pour le nombre de briques dans une feuille. Si elle est dépassée, on transforme la feuille en nœud et on répartit les briques dans les fils.

3 - Regard Critique

Le projet est fonctionnel et respecte les contraintes données. Cependant on peut avoir un regard critique sur le travail fait et réfléchir à ce qui peut être amélioré.

- Création et modularité des niveaux : Actuellement, créer des niveaux et les ajouter à notre jeu est fastidieux : il faut définir le niveau et l'ajouter dans create_level(). Idéalement, il faudrait trouver un moyen de créer des niveaux facilement sans avoir à toucher au code.
- Implémentation des extensions : Nous n'avons pas implémenté d'extensions pour ce projet, parmi lesquelles on aurait pu par exemple ajouter des pouvoirs, rendre des briques indestructibles, ajouter du son, des animations ...
- Balle soumise à la gravité : Bien que nous ayons implanté le fait que la balle soit soumise à la gravité, on a l'impression que celle-ci est négligeable, même lorsqu'on augmente la valeur de g ...

Conclusion

Ce projet aboutit à une version fonctionnelle d'un jeu de casse brique, qui respecte au mieux les exigences données.

Cependant, quelques problèmes restent à régler, comme la modularité des niveaux ou le dysfonctionnement de la gravité.