Projet C++

Rapport d'activité

« Jetpack Joyride at the bakery »

Encadrant: Pablo Rauzy.
Année Universitaire: 2014/2015.

BOULANGER Matthieu EI-SE 4

Sommaire:

1- Introduction

Dans le cadre de notre formation d'ingénieur, nous devons pour le semestre 7, coder un jeu vidéo en se servant de toutes les notions de C++ apprises en cours.
Pour cela, nous disposons de tous les outils nécessaires, c'est-à-dire le cours de C++, les TPs et l'Internet.

Pour ce projet, j'ai choisi de programmer un mini Jetpack Joyride qui a fait grand succès sur les tablettes et smartphones. Ce jeu est simplement une version plus moderne du jeu « Copter ».

Le principe de ce jeu est de contrôler un personnage qui doit parcourir la plus grande distance sans se faire percuter par des objets. Plus on appuie sur l'écran et plus le personnage monte. Si on ne touche pas l'écran, le personnage redescend par gravité.

J'ai donc programmé une version proche de ce jeu, qui s'appellera « Jetpack Joyride at the bakery ». Ce jeu tourne autour de la thématique de la boulangerie (qui est en réalité un simple 'private joke').

Le principe étant quasiment le même, les lignes électriques seront remplacées par des baguettes et les missiles par des miches. Il n'y aura qu'un seul gadget: le gadget 'anti-gravity'.

L'histoire du jeu résulte d'un conflit entre notre personnage, qui s'appellera 'Sam', et un boulanger. Ce dernier très en colère, a décidé de blesser notre héro en lançant des baguettes et des miches. Si ces derniers sont, dans notre vie limpide, inoffensifs, ils sont dans ce jeu, très dangereux.
Le scénario n'est pas encore définitif. Il peut à tout moment changer au cas où le créateur aurait une meilleure idée.

2- Structures et organisation

a- Interface

L'interface est assez simple. On dispose seulement de 3 touches.

- 2 touches directionnelles.

  La première correspond à la touche « UP » qui permet de surélever le personnage. Il faut appuyer assez longuement pour le faire monter.
  La deuxième touche est la touche « DOWN » qui permet, lorsque le personnage mangera un cupcake magique, de faire descendre le personnage car la gravité sera inversée.

- La touche « ESCAPE ».
  Cette touche permet simplement de quitter le jeu.

Le contrôle du jeu via la souris est seulement possible sur l'écran du menu. D'un simple clique, on peut choisir entre observer les derniers scores ou jouer.

b- Audio

C'est une pseudo-classe. J'ai préféré choisir de contrôler la musique par le biais d'une classe car il y avait trop de fichier à gérer (au moins 3 chansons + effet audio).
La première chanson concernera l'écran de menu, la deuxième, le jeu et enfin la troisième, l'affichage des scores.
Les effets audio ne sont, à ce jour, pas encore codés.

c- Fond d'écran – class Background

Il y aura bien un fond d'écran au niveau de la Game play. Ce fond d'écran défilera au fur et à mesure que le personnage avancera. Le fond d'écran représentera les rues de notre chère capitale: Paris.

d- Personnage – class Character

Un unique personnage qui court ou vole sans cesse. Il n'a qu'un seul profil. Ce personnage a par défaut une seule vie. Cependant, il peut gagner des vies qui lui sera proposé pendant le jeu. Il devra toucher des cœurs.

/* Projet*/
Le personnage devra récolter des pièces pendant sa course. Plus il gagnera de pièce et plus le score grimpera!

e- Cupcake – class Cupcake

/* Projet en cours */

Un cupcake apparaîtra à l'écran, et si le personnage le touche, la gravité changera de direction. Ce cupcake n'est donc considéré ni comme une arme, ni comme un soutien.

f- Baguette et miche - class French_Stick and Round_loaf

La baguette et la miche sont considérées comme des armes. Elles seront placées aléatoirement et défileront de droite à gauche (c'est-à-dire, face au personnage). Si le personnage touche une de ces armes alors le jeu s'arrêtera définitivement.
Les baguettes apparaissent au premier niveau et les miches à partir du second niveau. La différence entre les baguettes et les miches est que les baguettes sont statiques alors que les miches défilent plus vite.
Au niveau difficile, les baguettes tournent sur elle-mêmes.

g- Score

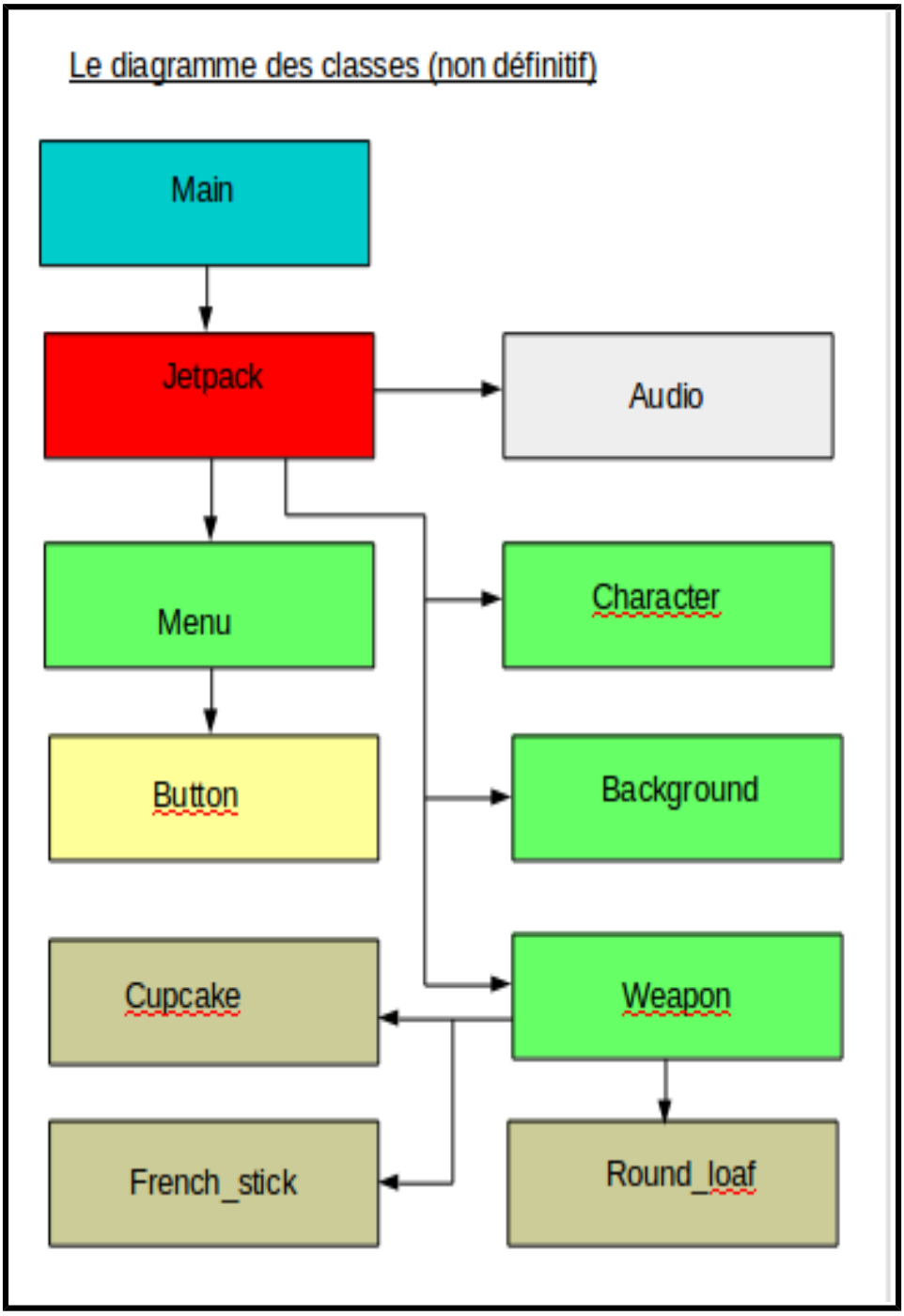
Le score est affiché durant la GamePlay. C'est tout simplement un kilométrage de la distance parcouru par le personnage.

Lorsque l'on perd, le score est inscrit dans un fichier texte, puis retransmis à l'écran « High Score ».

/*Projet*/
Il y aura également un affichage du nombre de pièce récoltée pendant le jeu. Un algorithme s'exécutera à la fin du jeu pour faire un score qui globalise la distance parcouru et le nombre de pièce récoltée.

3- Diagramme des classes

Voici un schéma qui récapitule à ce jour, l'organisation de mes classes. Je me suis rendu compte que, même si mon jeu s'exécute bien, je n'exploite pas assez la puissance du langage C++. Il y aura donc peut être quelques modifications entre ce rapport et le code qui sera rendu.

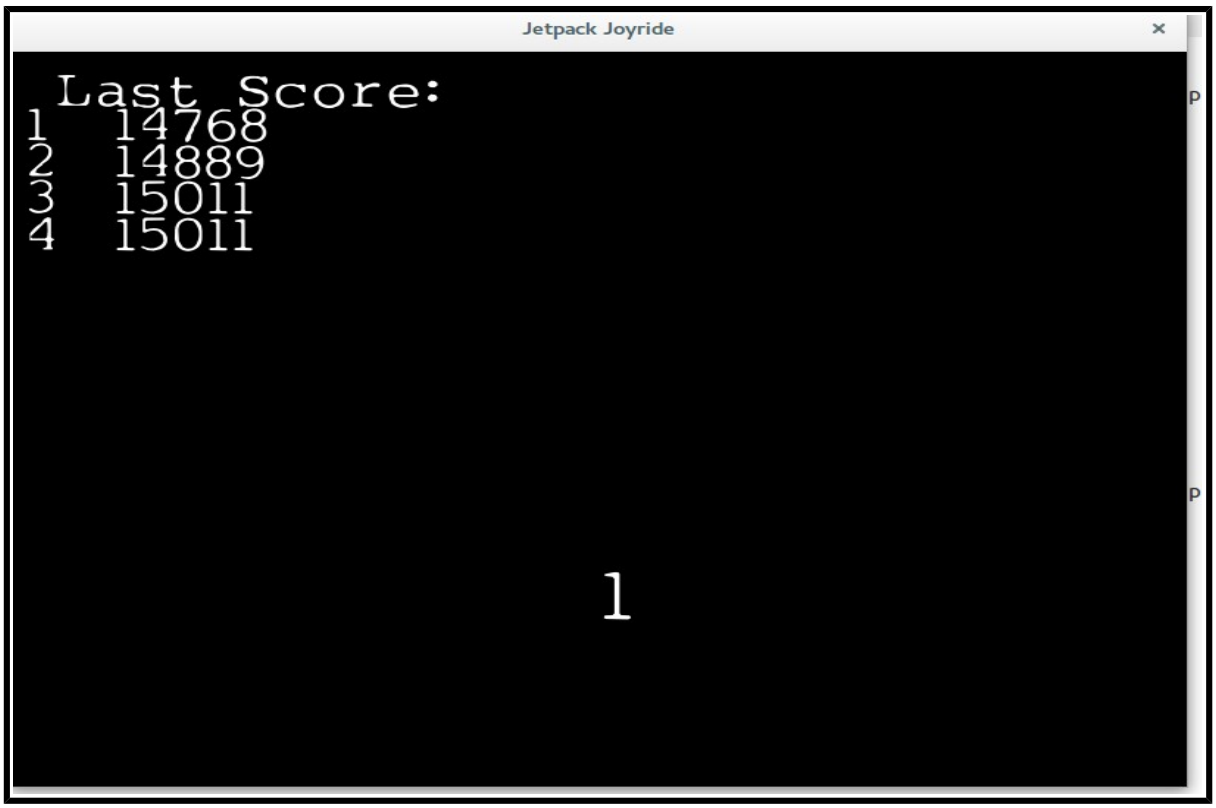


4- Etat d'avancement et projets

Actuellement le jeu fonctionne plutôt bien. Je dispose d'un jeu simple mais efficace. Il me reste encore pas mal de choses à régler, notamment: les histoires de Cupcake et de pièces à récolter. Cela va me prendre encore beaucoup de temps et j'espère qu'elles seront présentent dans mon code final.

Il n'y a pas de réel bug pour l'instant. J'ai utilisé gdb pour déboguer et Valgrind pour les fuites de mémoire. Ce dernier me présente pour l'instant 60 bytes de définitivement perdu. Il faut que je trouve l'origine de ce problème.

Voici quelques captures d'écran du jeu actuel (sous réserve de modification)...


Jetpack Joyride
Last Score:
1 14768
2 14889
3 15011
4 15011
1


Jetpack Joyride

5- Gestion du temps

Étant seul dans ce projet, il n'y a donc aucune répartition de tâches.

Cependant, je code au moins 10 minutes tous les soirs. Le reste du temps, c'est du débogage via gdb et des tests.