

5 juin 2023

Cappelié Lila
Teyssier Salomé

Synthèse d'images 1

Projet: The IMAC Light Corridor



1. Mode d'installation

Le dossier contient un fichier **main.cpp**, un header **draw_scene.h** qui initialise les différentes structures et fonctions, ainsi que cinq autres fichiers .cpp qui définissent chacun les fonctions d'un élément (setup initial du couloir, les différents obstacles, les différents bonus, les collisions de la balle, et les menus et textures). Il se compile avec Cmake. Le dossier est constitué dans un template compilable sur Linux et Windows.

2. Fonctionnement

Lorsqu'on lance le jeu, un premier menu s'affiche. L'utilisateur peut alors cliquer sur E pour commencer à jouer, ou sur R pour quitter. Lorsqu'il lance la partie, le départ de la balle se fait au clic gauche. Les mêmes commandes E et R sont disponibles sur les écrans de fin (gagné ou perdu).

3. Résultats obtenus

L'objectif du jeu est de garder la balle dans le couloir grâce à une raquette qui suit la souris du joueur : si la balle sort du couloir, le joueur perd une vie. Lorsque le joueur lance une partie, il possède cinq vies, qui sont affichées en bas de l'écran.

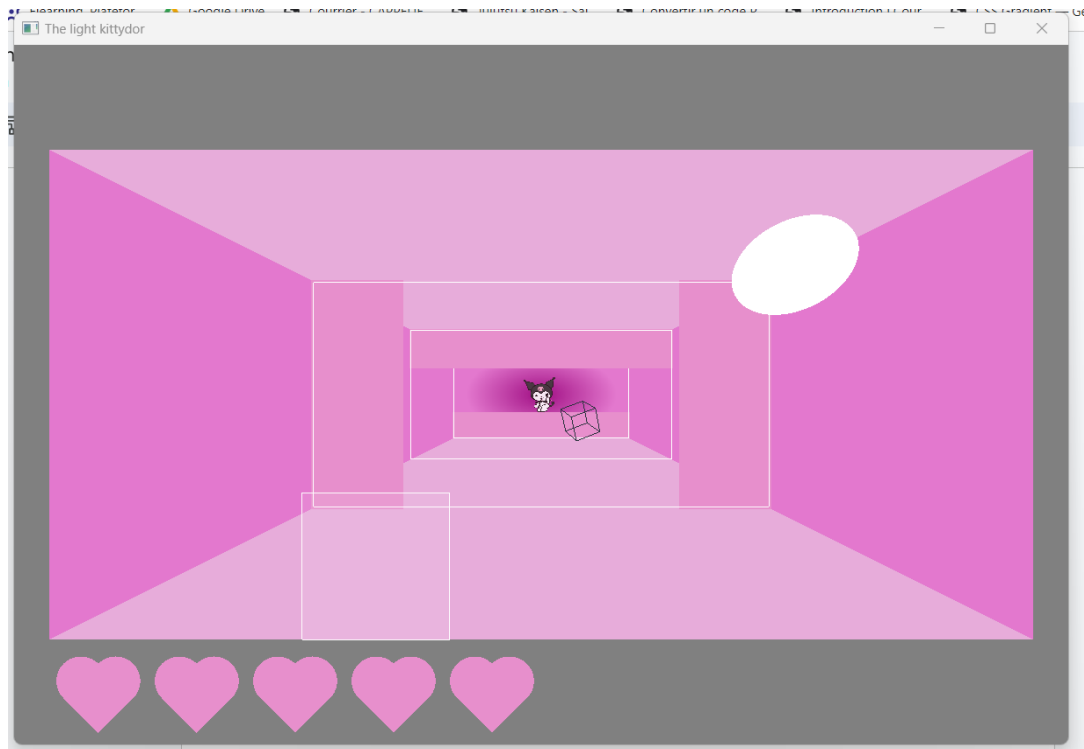
On commence à avancer dans le couloir à partir du premier tir de la balle. L'utilisateur se trouve alors face à des obstacles qui avancent vers lui, il doit faire en sorte que la balle les évite. Il y a huit obstacles différents, dont six fixes et deux qui bougent horizontalement et verticalement. Trois obstacles sont présents à la fois dans le couloir, ils sont choisis aléatoirement.

De plus, deux bonus différents apparaissent quelques fois dans le couloir, l'un permettant à l'utilisateur de gagner une vie (le joueur en a sept au maximum), et l'autre permet à la balle de coller à la raquette avant le prochain tir.

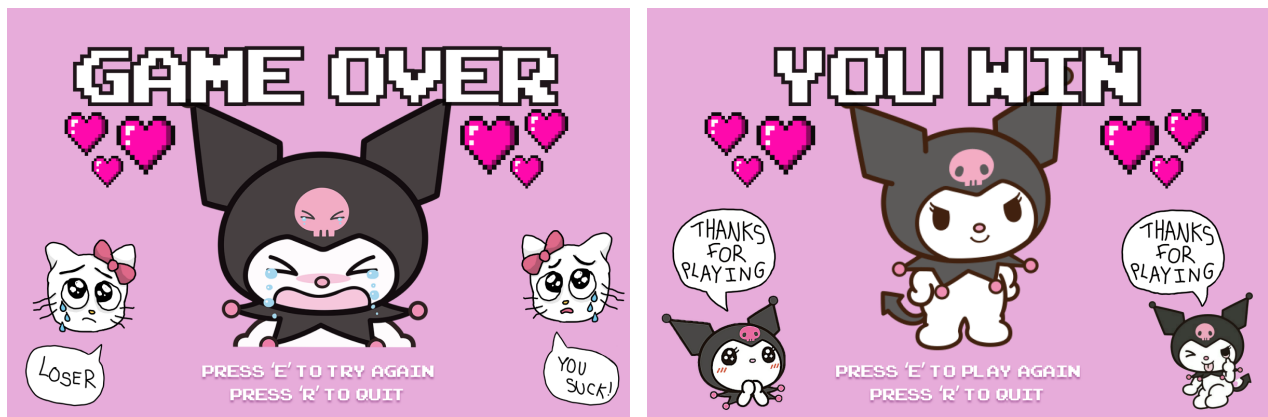
Le joueur gagne s'il parvient à passer 40 obstacles, et perd quand il arrive à zéro vie.



Le menu de départ



Le jeu



Les menus de fin

4. Méthode de travail

L'une fait les fonctions pur opengl, l'autre celles en c++. Nous n'utilisons pas le git : comme l'une faisait les fonctions ne nécessitant pas l'interaction avec les autres, nous nous les envoyons au format texte, pour toutes les mettre en commun sur le pc de l'autre.

5. Difficultés et Amélioration

Nous avons eu des problèmes avec les formats de chargement d'images : certaines, pourtant dans le bon format, ne voulait pas être lu / n'était pas lues correctement. Nous n'avons aussi pas eu le temps de coder les sources de lumière, ainsi que les relations raquette-obstacles. Ici, la raquette avance même si un obstacle est juste devant elle. Pendant longtemps nous avons aussi eu des bugs de collisions, dus au fait que nous avons oublié une condition nécessaire au rebond de la balle. Ceci provoquait des rebonds non voulus.

Pour la clarté du code, il aurait fallu éviter d'initialiser toutes les variables dans le main directement. De plus, pour les fonctions qui chargent les textures, il aurait fallu n'en faire qu'une et passer le chemin de l'image à charger en paramètre de la fonction, au lieu d'avoir 4 fois la même fonction.

Nous aurions pu aussi laisser quelques secondes d'écart entre le moment où la balle sort du couloir et où elle respawn sur la raquette, afin de bien comprendre qu'une vie vient d'être perdue. C'est la même chose pour l'affichage des écrans de fin.

De plus, le fait que la raquette ne soit pas stoppée par les obstacles provoque parfois des problèmes de collisions : si la balle est envoyée par la raquette sur un obstacle trop proche, elle peut parfois le traverser.