

Rapport projet de synthèse d'images

Mode d'installation :

Notre projet de synthèse d'image se compile avec la commande "make" suivie de "bin/projet.out", qui affiche la fenêtre du jeu pour Mac ou ".bin/projet.out" pour Linux. Nous avons utilisé la librairie OpenGL pour faire notre jeu.

Fonctionnement :

Le joueur arrive tout d'abord sur le menu où il peut taper sur la touche "J" du clavier pour jouer. Il arrive ensuite dans le jeu en lui-même.

Pour avancer dans le jeu, c'est-à-dire faire avancer la raquette, le joueur peut cliquer sur le bouton gauche de sa souris. Pour lâcher la balle, il peut cliquer sur le bouton droit de sa souris.

A tout moment, il peut quitter le jeu avec la touche "Q" de son clavier ou la croix en haut à droite de sa fenêtre.

La fenêtre ne peut pas être redimensionnée.

Résultats obtenus :

Nous avons pu implémenter les différentes fonctionnalités présentées dans le projet.

En effet, nous avons comme éléments présents : le corridor, la raquette, la balle, les obstacles et les bonus.

Concernant l'interface Homme-Machine et le moteur de jeu, nous avons pu implémenter :

- le mouvement de la raquette
- les collisions. Pour les collisions, nous avons vérifié les conditions sur la profondeur, et les coordonnées des objets que nous souhaitons collisionner. Si les deux objets étaient à la même profondeur, et aux mêmes coordonnées, nous avons inversé la vitesse de la balle sur l'axe x en réalisant l'opération : $ball \rightarrow speedX *= -1$.
- le menu de départ et de fins

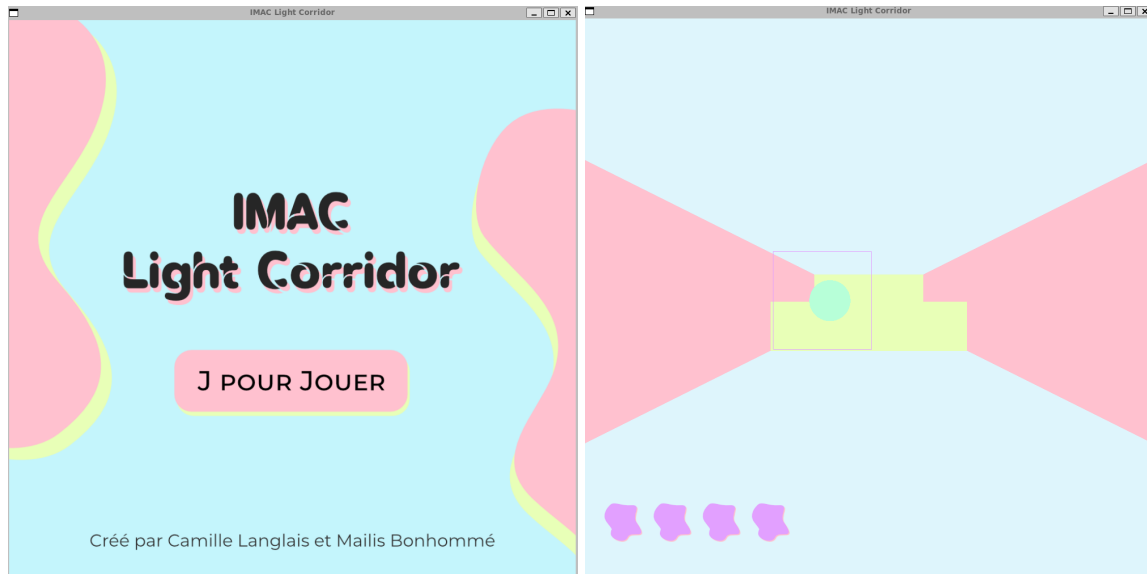
Ainsi, le jeu ne peut avancer que si l'utilisateur lance la balle. Lorsqu'il la lance dans le corridor, celle-ci se dirige alors le long du tunnel, en rebondissant contre le mur et obstacles qu'elle rencontre. La raquette suit la balle tout au long du jeu.

Durant son parcours, elle rencontre également des bonus. Les bonus permettent au joueur d'acquérir différentes fonctionnalités. En effet, selon leur type, si la raquette traverse le bonus, le joueur peut soit recoller la balle au centre de la raquette ou bien gagner une vie. D'ailleurs, le nombre de vie restantes au joueur s'affichent tout au long de la partie, en bas à gauche de l'écran.

Si dans son parcours la raquette entre dans un obstacle, le jeu n'avance plus. Il faut déplacer la raquette pour qu'elle soit hors de l'obstacle et relancer la balle. Il faut donc faire en sorte que la raquette contourne les obstacles durant le jeu.

Nous n'avons cependant pas implémenter l'illumination car nous n'avons pas trouvé comment faire.

Nous avons donc, au final, un jeu fonctionnel avec un système de vie, composé d'un début et d'une fin.



Méthode de travail :

Nous avons décidé de nous répartir les différentes tâches selon nos compétences et nos envies. Nous avons pour cela créé un GitHub pour pouvoir travailler à deux à distance. Cependant nous avons quelques soucis sur GitHub donc nous avons dû recréer un répertoire Git avec donc un nouveau lien.

Camille a donc réalisé :

- Le corridor
- La balle
- Toutes les collisions et les rebonds

Mailis a réalisé :

- La raquette et son mouvement
- Les obstacles
- Les bonus
- Le menu de départ et de fin, ainsi que le design du jeu

Difficultés rencontrées :

Nous avons rencontré de nombreuses difficultés.

En effet, étant toutes les deux très peu familières à l'OpenGL et à ce langage, nous avons passé beaucoup de temps sur certains éléments qui étaient sur le papier plutôt simples.

A cause de cela, nous avons pris énormément de retard au fur et à mesure et nous avons pu seulement avancer réellement à quelques semaines de la date de rendu.

Également, comme dit précédemment, comme nous ne connaissions pas non plus le fonctionnement de GitHub, nous avons dû apprendre sur le tas à nous en servir.

Cela a donc mené à de nombreux problèmes de merge, de conflit et autres.

Nous avons même dû créer un nouveau répertoire pour accueillir notre projet tellement nous avons eu de soucis. A la fin, nous avons su gérer le Git correctement et apprendre de nos erreurs.

Améliorations possibles :

Il y a de nombreuses améliorations possibles dans notre projet notamment dans l'optimisation de notre code.

La plupart de nos draw devrait se situer dans `draw_scene.c`.

Tout ce qui implique les textures des menus ou bien les textures des vies devrait se situer dans des points c à côté et non pas dans le main.

Nous pensons également que les collisions pourraient être mieux optimiser pour en réaliser des plus complexes.

Pour conclure, nous avons trouvé ce projet très intéressant ! Il nous a permis d'apprendre de façon concrète l'OpenGL et de faire un projet en groupe via GitHub.