

# **Synthèse d'Images**

## **Rapport de Projet**

### ***THE LIGHT CORRIDOR***

**Christian Ky & Edouardo Rodrigues**

## TABLE DES MATIÈRES

---

<i>Installation</i> . . . . .	3
<i>Fonctionnement</i> . . . . .	4
<i>Résultats obtenus</i> . . . . .	4
<i>Détails d'implémentation</i> . . . . .	7
<i>Méthode de travail</i> . . . . .	8
<i>Difficultés rencontrées</i> . . . . .	9

## Introduction

The Light Corridor est un jeu de raquette avec un ballon, le but étant de faire rebondir la balle le plus longtemps possible sans qu'elle ne vous échappe en dehors de l'écran.

Nous commencerons par indiquer les détails d'installation du jeu ainsi que décrire les différentes étapes nécessaires exécuter le jeu.

La partie fonctionnement décrit comment le jeu fonctionne, quels sont les buts et les règles de celui-ci.

Nous partagerons ensuite dans les résultats obtenus, l'état final du jeu, à travers du texte et différentes illustrations.

Nous détaillerons ensuite quelques détails d'implémentations, tels que le langage utilisé, les choix de conception qui ont été faits pour réaliser les différentes mécaniques du jeu.

Pour finir, nous parlerons de notre organisation afin de mené a terme et a bien ce projet ainsi que les différentes difficultés rencontrés tout au long du processus de développement.

## Installation

L'installation du projet est plutôt simple, il suffit de télécharger le dossier, tout le nécessaire se trouve à l'intérieur.

Concernant les dépendances, nous utilisons « glut » ainsi que « glfw3 » ainsi que « glew », ce sont les différentes librairies utilisés en TD.

Pour lancer le jeu, il suffira de lancer la commande « make » à la racine du projet, et lancer l'exécutable à l'aide de la commande « ./bin/TheLightCorridor ».

## Fonctionnement

Le jeu compile et est fonctionnel. Il a été développé dans son intégralité et répond à toutes les exigences demandés par le cahier des charges.

Le fonctionnement du jeu est simple, au lancement de celui-ci, un menu très explicite attendra de la part du joueur une interaction afin de jouer ou bien quitter le jeu.

Une fois le jeu lancé, le joueur contrôle la raquette à l'aide des mouvements de la souris. Lorsque la balle touche la raquette, elle rebondit dans la direction opposée. Si la balle s'échappe en dehors de l'écran, le joueur perd une vie et le jeu continue jusqu'à ce que toutes les vies soient épuisées.

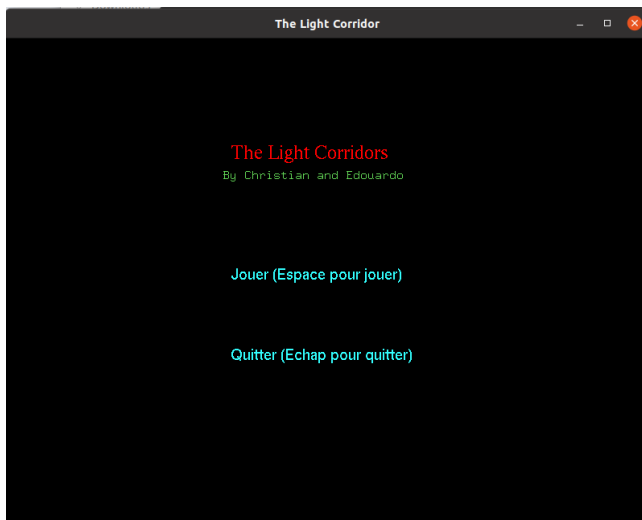
Le joueur peut bénéficier de différents types de bonus simplement en les traversant avec la raquette.

Le jeu possède plusieurs niveaux qui sont passés en fonction du score minimum à atteindre.

## Résultats obtenus

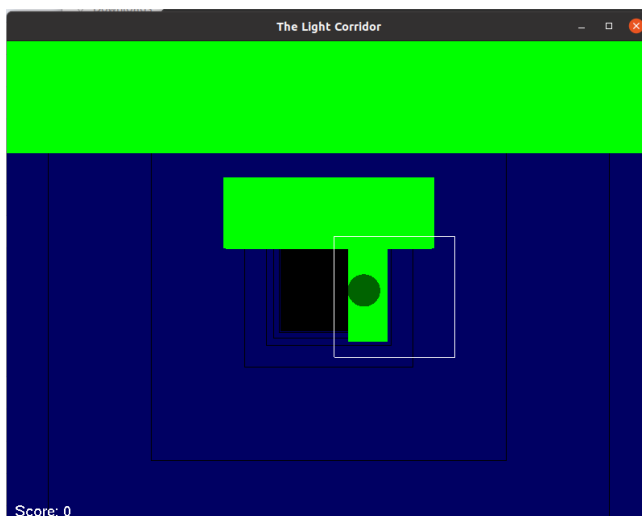
Les résultats obtenus sont très satisfaisants et répondent aux exigences du cahier des charges.

Voici le menu de début de jeu :



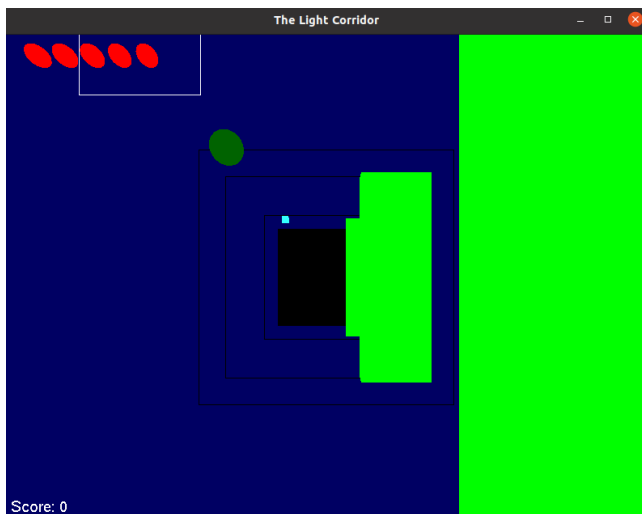
Comme indiqué, le joueur peut commencer à jouer d'un simple appuie sur la touche espace, ou bien quitter le jeu en appuyant sur echap.

Voici l'interface de jeu en pleine partie, on y retrouve le score du niveau en bas à gauche. A chaque passage de niveau on y retrouve l'inscription en haut de l'écran.



On retrouve donc dans cet illustration la raquette transparente, les différents obstacles ainsi que le ballon de jeu.

Voici maintenant une illustration de l'interface de jeu avec cette fois-ci un bonus à récupérer qui est en haut à gauche.



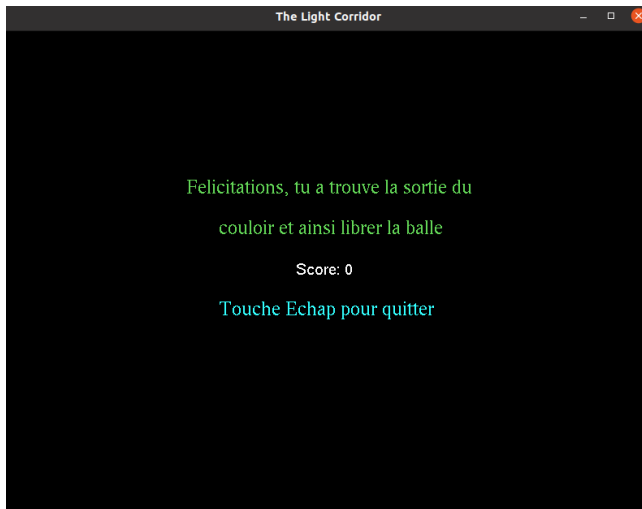
On retrouve donc dans cet illustration le bonus en bleu clair, pour le récupérer, il suffit de faire passer la raquette à travers celui-ci.

A présent, voici l'interface de fin de jeu quand le joueur perd la partie.



On y voit le score ainsi que la touche à presser pour quitter le jeu.

Et pour conclure, voici l'interface de jeu gagnante.



Après de nombreux niveaux passés, il arrive un moment où le joueur gagne la partie, on y verra alors son score de partie, ainsi que les félicitations pour sa partie. Idem, on presse la touche indiquée pour quitter la partie.

## Détails d'implémentation

Le jeu a été entièrement développé à l'aide du langage C.

Les collisions de la balle contre la raquette est un élément clé du jeu, c'est la partie délicate du développement car c'est sur cette physique du ballon que repose la jouabilité de celui-ci.

Pour traiter cela, nous avons commencé par calculer l'angle de la vitesse de la balle dans le plan XZ, ceci est fait avec l'instruction « **float angle = atan2f(ballVZ, ballVX);** ». Par la suite on va calculer l'angle entre le centre de la raquette et le centre de la balle dans le plan XY grâce à l'instruction « **float normalAngle = atan2f(racketY - ballY, racketX - ballX);** », il va être utilisé pour déterminer la direction normale de la surface de la raquette au point de collision. On continue avec le facteur d'influence du rebond selon la direction normale de la raquette. Nous avons décidé que celui-ci ne soit pas totalement élastique sans tenir

compte de la direction de la raquette en le mettant à 0, ni totalement dirigé par la direction de celle-ci en le mettant à 1, nous avons donc choisis le juste milieu, ce qui est un bon compromis. Par la suite il nous fallait calculer l'angle de la vitesse de la balle après le rebond, comme ceci « **float reflectionAngle = angle + reflectionFactor \* (normalAngle - angle);** », il est obtenu en ajustant l'angle initial de la balle en fonction de la direction normale de la raquette et du facteur de réflexion. Après avoir calculer tout cela, c'est le moment de calculer la nouvelle vitesse de la balle, en effet, la vitesse est nécessaire pour déterminé les composantes X et Z de la nouvelle vitesse de la balle. « **float speed = sqrtf(ballVX \* ballVX + ballVZ \* ballVZ);** », cette instruction calcule la vitesse de la balle en utilisant la formule de la norme d'un vecteur à deux dimensions. On obtient donc la nouvelle vitesse du ballon en X et en Z avec ces deux instructions : « **ballVX = speed \* cosf(reflectionAngle);** » et « **ballVZ = speed \* sinf(reflectionAngle);** ». La vitesse de la balle en X et en Z est donc mis à jour à l'aide de la nouvelle vitesse du ballon ainsi que de l'angle de réflexion. Enfin, une dernière étape crucial est requise, en effet il manque le changement de direction du ballon avec l'axe Y. C'est pour cela qu'on viens ajouter une dernière instruction à ce mécanisme de collision entre la balle et la raquette, l'inversion de la vitesse du ballon en Y afin de faire rebondir la balle dans la direction opposée sur l'axe Y, on le fait comme ceci « **ballVY = -ballVY;** ».

En combinant donc toutes ces étapes, cela nous permet de calculer le rebond de la balle en fonction de la position de la raquette et de la direction initiale de la balle, donnant ainsi une trajectoire de la balle agréable et réaliste après un rebond.

## Méthode de travail



En ce qui concerne la méthode de travail, nous avons commencé les bases du projet en étant côte à côte afin de se donner mutuellement les premières idées ainsi que le déroulement et comment nous allions procéder au développement de celui-ci. Ensuite nous avons bien évidemment travaillé à l'aide de GitLab, afin de se coordonner sur le travail déjà fait et déduire ce qu'il restait à faire, tout cela accompagné évidemment de quelques appels en visioconférence, ayant pour but la compréhension totale de ce qui a été fait par l'autre et vice-versa.

Pour conclure sur notre méthode de travail, on peut dire que cela a été un succès, nous avons été rigoureux en ayant une bonne organisation, ce qui a permis de construire un projet solide et mené à son terme.

## Difficultés rencontrées

Au cours de ce projet nous avons rencontré plusieurs difficultés, que ce soit au commencement de celui-ci, car établir un projet, avoir une bonne organisation et s'y tenir est très compliqué à tenir et à mettre en place.

Ensuite en ce qui concerne le développement, nous avons eu beaucoup de discussion, que ce soit la forme ou la manière de faire.

Et pour finir, en ce qui concerne l'implémentation, nous avons eu notamment une difficulté majeure, que nous avons surmontée en discutant beaucoup et ayant pris notre temps pour y réfléchir, il s'agit bien-sûr des mouvements et collisions du ballon, qui est la mécanique principale de ce jeu.

## Conclusion

Pour conclure, nous avons donc mis au jour notre jeu issu du célèbre jeu « The Light Corridor », en mettant à l'épreuve nos connaissances en Synthèse d'Images acquise tout au long du semestre ainsi que notre capacité à travailler en équipe.

Cela n'a pas été de tout repos, nous avons bien-sûr rencontrés quelques difficultés, mais grâce à une bonne organisation et à un travail rigoureux, nous avons réussi à mené ce projet à terme.

Nous sommes fier du résultat obtenus et de présenter notre jeu :

**« The Light Corridor by Christian Ky and Edouardo Rodrigues »**