

The Light Corridor

[GitLab du projet](#)



Présentation de “The Light Corridor”	3
Installation de “The Light Corridor”	3
Commandes utilisateurs.....	3
Le menu :.....	3
En jeu :.....	3
Attribution des touches :.....	3
Touches en Commentaires :.....	3
Fin de jeu :.....	4
Fonctionnalités implémentées.....	4
Fonctionnalités demandées :.....	4
Prise de Liberté :.....	4
Ajout Bonus :.....	4
Difficultés Rencontrés.....	4
Méthode de travail.....	5

Présentation de “The Light Corridor”

The Light Corridor est un jeu sorti en France en 1990 et développé et édité par Inforgrames, ce jeu fait partie de la Collection Crystal du Développeur,

Dans ce jeu on a une raquette que l'on peut déplacer horizontalement mais aussi verticalement sur un plan situé devant la caméra. On dispose aussi d'une touche permettant de nous déplacer vers l'avant dans le couloir (corridor). Nous disposons aussi de plusieurs vies qui se perdent lorsque l'on n'arrive pas à faire rebondir la balle avec la raquette (donc que la balle passe derrière nous).

Le but du jeu est d'atteindre le couloir en poussant la balle vers l'avant sans perdre toutes ses vies.

Installation de “The Light Corridor”

Décompressez le fichier, placez-vous dans le dossier “the_light_corridor”, là où se trouve le Makefile, puis dans un terminal la commande “make”.

Il n'y aura pas besoin d'exécuter un fichier de sortie.

Le jeu se compilera et se lancera automatiquement et lors de la fermeture tous les éléments temporaires seront supprimé automatiquement.

[Video1](#) [Video2](#) (cas où le projet ne marche pas)

Commandes utilisateurs

Le menu :

Une fois lancé nous retrouvons le menu

Il faudra cliquer sur jouer ou bien appuyer sur la touche Entrer pour pouvoir jouer

Pour quitter on peut cliquer sur quitter ou bien appuyer sur la touche ESC ou encore cliquer sur la croix.

En jeu :

Attribution des touches :

Fleche du haut : Déplace la raquette vers le haut

Fleche du bas : Déplace la raquette vers le bas

Fleche de gauche : Déplace la raquette vers la gauche

Fleche de droite : Déplace la raquette vers la droite

Entré : Lorsque la balle est collé à la raquette, envoi la balle vers l'avant

Tabulation : Avance la camera

Shift Gauche : Recule la caméra

Touches en Commentaires :

Ces touches servent au developpeur pour le débuggage, ils sont pour l'instant en commentaire

PavNum 9 : Zoomer en avant

PavNum 3: Zommer en arrière

PavNum 8 : Avancer avec la camera sans bouger le jeu

PavNum 4 : Aller à gauche avec la camera sans bouger le jeu

PavNum 5 : Reculer avec la camera sans bouger le jeu

PavNum 6 : Aller à droite avec la camera sans bouger le jeu

P : Augmentation de l'angle de luminosité de la source de lumière

M : Diminution de l'angle de luminosité de la source de lumière

Fin de jeu :

Selon si nous avons réussis à traverser le couloir ou non avec toutes nos vies, il sera affiché en cas de victoire "You Win" sinon "Game Over".

Fonctionnalités implémentées

Fonctionnalités demandées :

Le jeu est fonctionnel, et les propositions du sujet sont présentes :

- Des menus;
- Un couloir, avec une texture;
- Une balle fonctionnelle (avec les rebonds), et la raquette associée;
- Des shaders, dont une lumière au niveau de la caméra, une seconde sur la balle;
- Un système de vie, et un Game over en conséquence ou de victoire.

Prise de Liberté :

Nous avons décidé que la raquette serait contrôlée via des touches du clavier (Nous étions partis dessus au début et n'avons remarqué que à la fin qu'il fallait utiliser la souris.)

Ajout Bonus :

Nous avons ajouté en plus de ce qui est demandé deux musiques de fond qui peuvent être changé dans le code, à l'aide de la librairie SDL2 (voir annexe).

Autre :

Des fonctionnalités supplémentaires peuvent néanmoins être ajoutées, telle que les mots de passes pour différents niveaux, des niveaux aléatoires et procéduraux, des effets visuels ...

Difficultés Rencontrés

Nous avons rencontré beaucoup de difficultés :

- La gestion de la 3d : Lorsqu'on a commencé le projet nous ne voyons pas comment faire pour afficher l'espace en 3d, finalement c'est grâce à un TD que nous avons résolu le problème.
- Les rebonds de la balle : Quand nous avons voulu coder les rebonds de la balle, nous pensions faire des rebonds réels qui sont calculés par rapport à l'angle et la normal sur un mur, mais cette idée était bien trop compliquée et nous sommes partis sur un rebond assez simple qui lorsqu'on arrive sur un mur on repart en arrière.
- La texture : Nous avons fait un TD que vous nous avez demandé de sauter pour pouvoir faire les textures, cependant le TD faisait des Textures en 2D et nous voulions appliquer sur des Formes en 3D, après mainte recherche nous avons réussi à moitié à appliquer la texture sur les murs de notre couloir, après énormément d'essais, le rendu n'est pas vraiment ce qui est escompté mais on s'en rapproche.
- La balle Noclip parfois : Dans certains cas la balle noclip à travers les murs.
- Temps de travail : On trouve que le projet a une bonne durée de travail, cependant avec nos projets de réseau qui comptent plus nous avons pris moins de temps pour coder le jeu ce qui ne nous a pas permis de faire tout ce qu'on voulait.

Méthode de travail

Le travail a été effectué à deux via des sessions de code sur l'extension CodeTogether sur Visual Studio Code.

Nous avons aussi fourni un [Gitlab](#) pour mieux travailler.

En revanche, la plupart des commit et push sont originaires du compte de Steven.

Annexe

Comme nous avons ajouté des musiques sur le jeu, il faudra pour pouvoir compiler avec SDL2 et SDL2-mixer.

Pour installer la librairie SDL, sur ubuntu:

```
$ sudo apt-get install libsdl2-2.0-0
```

```
$ sudo apt-get install libsdl2-mixer-2.0-0
```

```
$ sudo apt-get install libsdl2-dev
```

```
$ sudo apt-get install libsdl2-mixer-dev
```