

Alexandre KINGUÉ

Synthèse d'images I

Projet : The IMAC Light Corridor

Le but de ce projet est de mettre en pratique les nouvelles notions obtenues pendant les cours de Synthèse d'Images. Pour se faire, vous devrez concevoir un jeu vidéo relativement simple, inspiré plus que largement sur un (très) ancien jeu vidéo appelé "The Light Corridor".

Compilation et exécution :

Pour compiler le projet lancer la commande « make » après avoir fait un 1^{er} « make clean ».

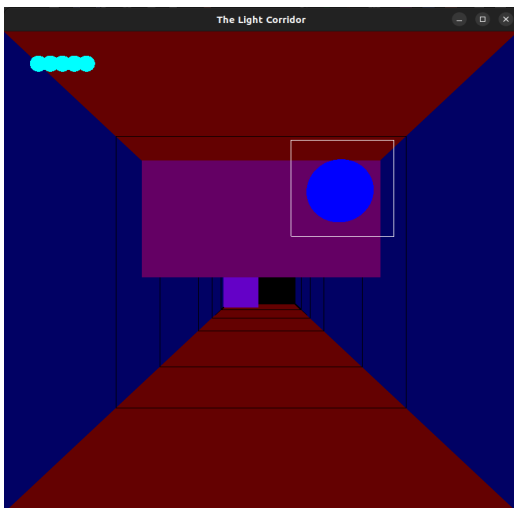
Une fois cela fait vous pouvez exécuter le programme avec la commande « ./bin/main »

Commandes utilisateurs :



Une fois l'exécutable lancée vous vous retrouver devant un menu servant à lancer 3 niveau différents ou à quitter le jeu directement.

Les niveaux plus élevés ont des corridor plus long et un plus grand nombre d'osbactcles.



Une fois un niveau sélectionné, vous voilà dans le jeu :

Vous avez pour commencer la balle qui est accroché à la raquette du Joueur et qui attends son clic pour envoyer la balle.

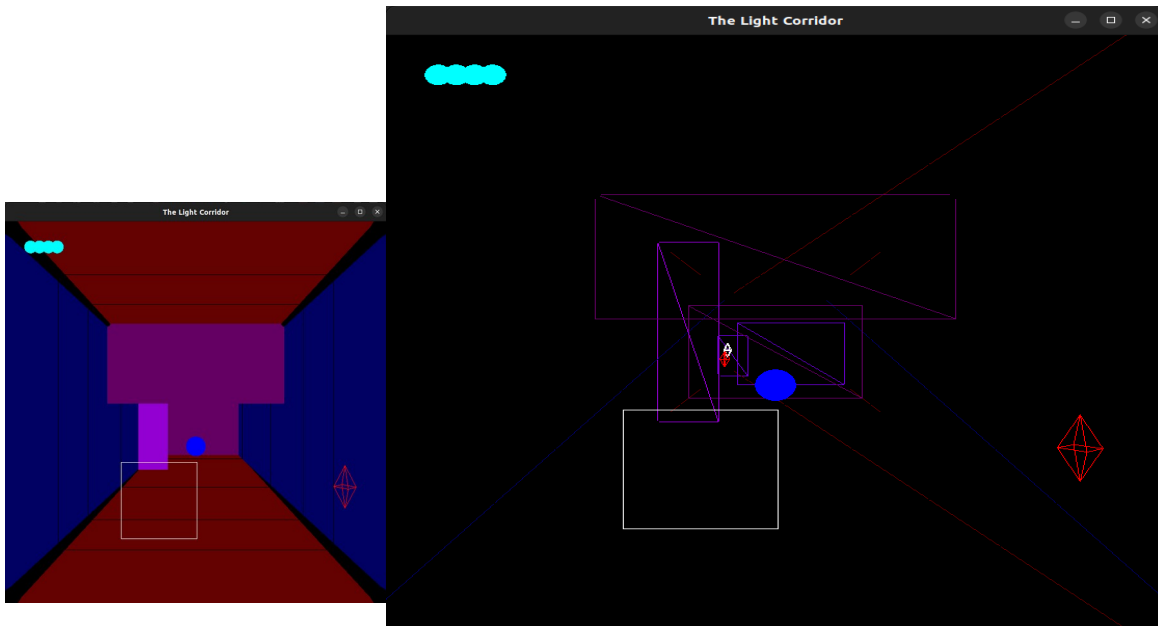
La raquette est symbolisé par un cadre blanc et la balle par une sphère bleu.

Le nombre de vie est affiché sous forme forme de sphère bleu clair en haut à gauche de l'écran.

Vous pouvez avancé d'un clic droit et reculer d'un clic gauche (améliration).

Alexandre KINGUÉ

Options (amélioration) :

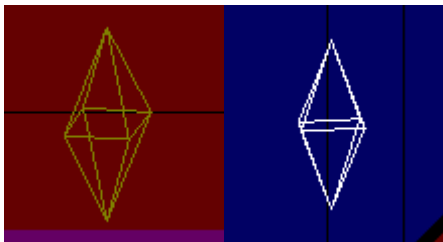


Vous avez 3 options, la pause qui peut être mise et enlever par la touche B, un mode transparent qui peut être mis avec la touche L (utile pour mieux voir ce qu'il se passe devant) et un mode plein qui sert à revenir sur le mode normal après le mode transparent avec la touche P.

Obstacles :

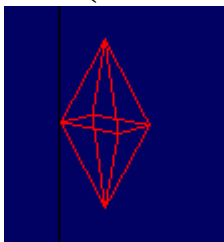
Il y a 3 types d'obstacles, la balle peut entrer en collision (sur la face et les bords), pour les dépasser avec votre raquette vous devez avancés (clic gauche) avec votre raquette mais dans les espaces qui ne sont pas des obstacles en face de vous.

Bonus :



Il y a la présence de bonus qui accorde 2 type de bonus différents (différenciable par leur couleur), l'un rajoute une vie et l'autre rend la raquette collante ce qui permettra de saisir la balle avec la raquette la prochaine fois qu'elle la touchera.

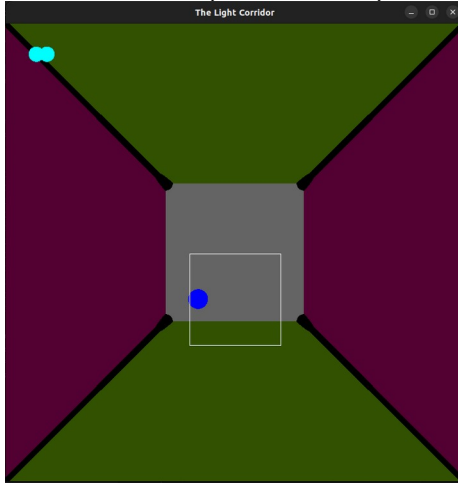
Malus (amélioration) :



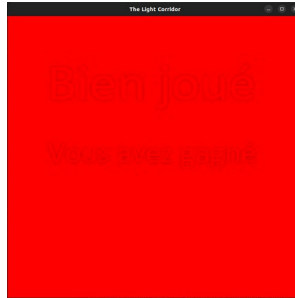
J'ai ajouté la présence de malus qui ajoute de la vitesse à la balle la rendant très dure à suivre. Ils sont donc à éviter.

Alexandre KINGUÉ

Fin de niveaux (amélioration) :



A la fin de chaque niveau il y aura un « Mur Boss » à détruire pour sortir du Corridor et finir le niveau. Le mur renverra la balle avec plus de force et dans direction modifiés pour piéger le joueur. Le joueur devra tirer sur le mur 10 fois de suite pour le détruire et finir le niveau.



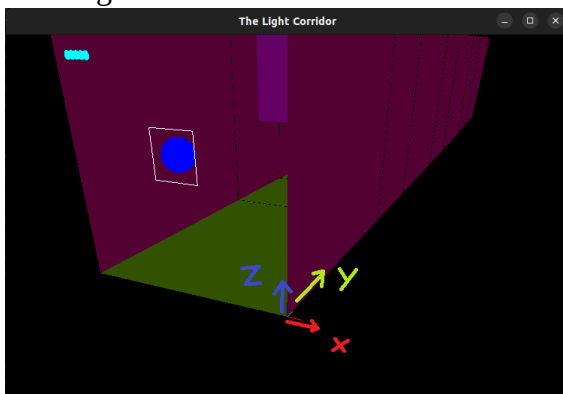
Méthode de travail :

N'ayant pas réussi à me mettre avec quelqu'un j'ai donc du faire le projet en solitaire, j'ai donc laissé tomber l'Illumination par manque de temps. J'ai commencé par imaginer les structures des Murs du Corridor ainsi que la balle pour la faire rebondir dessus et avoir une base de travail. Après chaque fonctionnalité correctement implantés je faisais un rendu git pour m'assurer de progression. Pour certains calcul (la collisions entre les bords des obstacles et la balle), j'ai du faire des recherches pour savoir comment la balle devait réagir.

Détails techniques :

*C'est une technique connu mais pour l'affichage je n'affichais que ce qui est dans le champ de vision du joueur pour préserver de la puissance de calcul.

*Les dimensions ne sont pas ce qu'elles semblent être, en effet le corridor avance sur l'axe de Y, l'axe Z est l'axe des ordonnées et l'axe X est l'axe des abscisse mais le corridor se situe dans les négatifs de X.



Alexandre KINGUÉ

Détail calcul collision Raquette-Balle :

Résultat recherché : La raquette envoie plus d'effet si la balle frappe plus vers les côtés.

On calcul le centre de la Raquette sur l'axe X/Z.

$\text{centre_z} = (\text{joueur.max_cote_z} + \text{joueur.min_cote_z}) / 2$

$\text{centre_x} = (\text{joueur.max_cote_x} + \text{joueur.min_cote_x}) / 2$

joueur.max_cote_x , joueur.max_cote_z , joueur.min_cote_x et joueur.min_cote_z sont de repères dans l'espace de la raquette.

Ensuite on fait le rapport de la balle sur la haut de la raquette, pour obtenir sa position sur sa raquette sur 1.

$\text{tmp_z} = ((\text{balle} \rightarrow \text{z} - \text{joueur.min_cote_z}) / (\text{joueur.max_cote_z} - \text{joueur.min_cote_z}))$

$\text{tmp_x} = ((\text{balle} \rightarrow \text{z} - \text{joueur.min_cote_x}) / (\text{joueur.max_cote_x} - \text{joueur.min_cote_x}))$

On fait ensuite la soustraction de cette valeur par rapport au centre de la raquette, comme ça si la balle frappe le centre il ne passera rien.

$\text{ajout_vec_z} = \text{tmp_y} - ((\text{centre_z} - \text{joueur.min_cote_z}) / (\text{joueur.max_cote_z} - \text{joueur.min_cote_z}))$

$\text{ajout_vec_x} = \text{tmp_x} - ((\text{centre_x} - \text{joueur.min_cote_x}) / (\text{joueur.max_cote_x} - \text{joueur.min_cote_x}))$

On divise ce résultat par la valeur du côté de la raquette pour être sur d'un faible ajout et inférieur à 1.

$\text{ajout_vec_z} = \text{ajout_vec_z} / \text{joueur.dist_cote}$

$\text{ajout_vec_x} = \text{ajout_vec_x} / \text{joueur.dist_cote}$

ajout_vec_z et ajout_vec_x sont ensuite ajoutés aux vecteurs de la balle.

Difficultés rencontrées :

Le fait d'être seul m'a empêché bien saisir le projet, au fur et à mesure que le projet avançait j'ai compris que certaines choses devaient être revu et que l'Illumination demanderait du temps pour faire une refonte du projet, du temps que je n'avais pas.

Éléments présents :

- Le corridor
- La raquette
- La balle
- Les obstacles
- Les bonus (transparent)

Interface et moteur présent :

- Mouvement de la raquette
- Les rebonds
- Menu de départ et de fin
- Texture

Alexandre KINGUÉ

Améliorations faites:

- Un mode pause (touche B)
- Un mode transparent pour mieux voir le match (touche L)
- On peut revenir en arrière avec la raquette
- Création de malus
- Création des boss de "fin de niveau"
- Création différents niveau de difficultés.

Améliorations possible :

- L'Illumination
- Score(j'ai déjà les éléments dans le dossier input)

Conclusion :

Malgré les problèmes rencontrés, le projet fut intéressant qui lie calcul de trajectoire et modèle 3D tout en demandant des compétences à la programmation. Donc j'ai aimé travailler sur ce projet qui demande les compétences vu pendant ce semestre.

Git : <https://github.com/KingueAlexandre/Projet-The-IMAC-Light-Corridor>