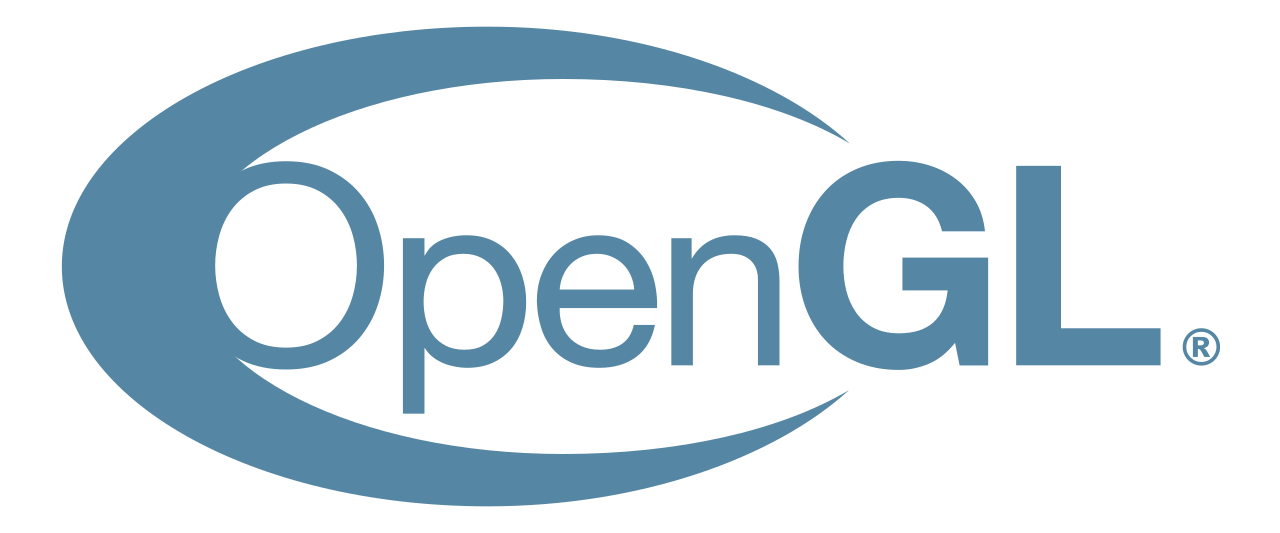
DURET Guillaume

LECONTE Thibault

3ETI / Groupe D

**Compte – Rendu TSI**

**TP Synthèse d’image**

Introduction

Dans ce projet, il nous était demander de créer un jeu en 3D minimaliste de notre choix, que nous devions coder à l’aide de l’outil OpenGL.

L’objectif était de nous faire prendre en main des scènes virtuelles 3D en temps réel. Nous devions donc développer des compétences techniques dans l’envoie de données sur la carte graphique, la manipulation de shaders (pour le contrôle de l’affichage), etc.

Nous avons donc décidé de créer comme jeu un « stand de tir » où l’utilisateur doit toucher des cibles qui sont placées aléatoirement dans un décor en 3D.

Le développement de notre jeu s’est donc tourné autour de plusieurs problématiques, que nous expliciterons ci-après, et auxquelles nous avons apporté des solutions pour parvenir à nos fins.

Il est important de souligner que pour l’intégralité du code nous avons réutilisé le code du main du programme Programme\_10 donné parmi les sources du support de TP. Notre travail s’est découpé en deux majeures parties.

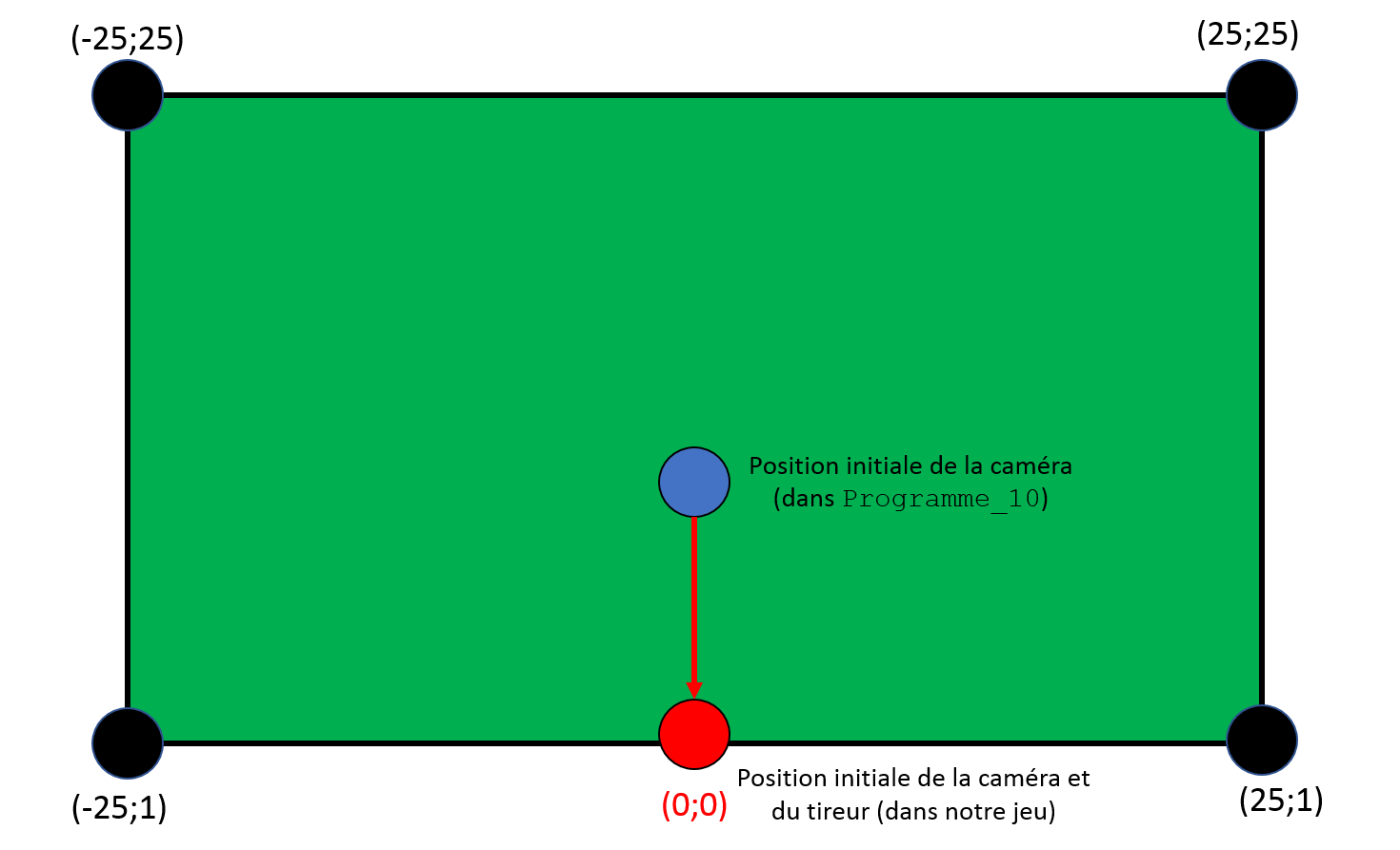
Dans un premier temps nous avons codé toute la partie « calcul » nécessaire pour le jeu, c’est-à-dire la position de la caméra, les positions des objets, les collisions entre objets, les distances entre les objets, les interactions, etc en utilisant uniquement pour objet le stégosaure.

Puis dans un second temps nous avons traité la partie graphique et affichage, en utilisant de nouveaux objets et de nouvelles textures.

1° Première problématique : position du tireur, de la caméra.

La première étape en réutilisant le code du Programme\_10 était de modifier les coordonnées du sol afin que la caméra se trouve sur le centre d’un côté du rectangle (qui correspond à la carte).

Temps estimé : 45 minutes – 1 heure



2° Deuxième problématique : rotation du tire sur sur un angle de 180°.

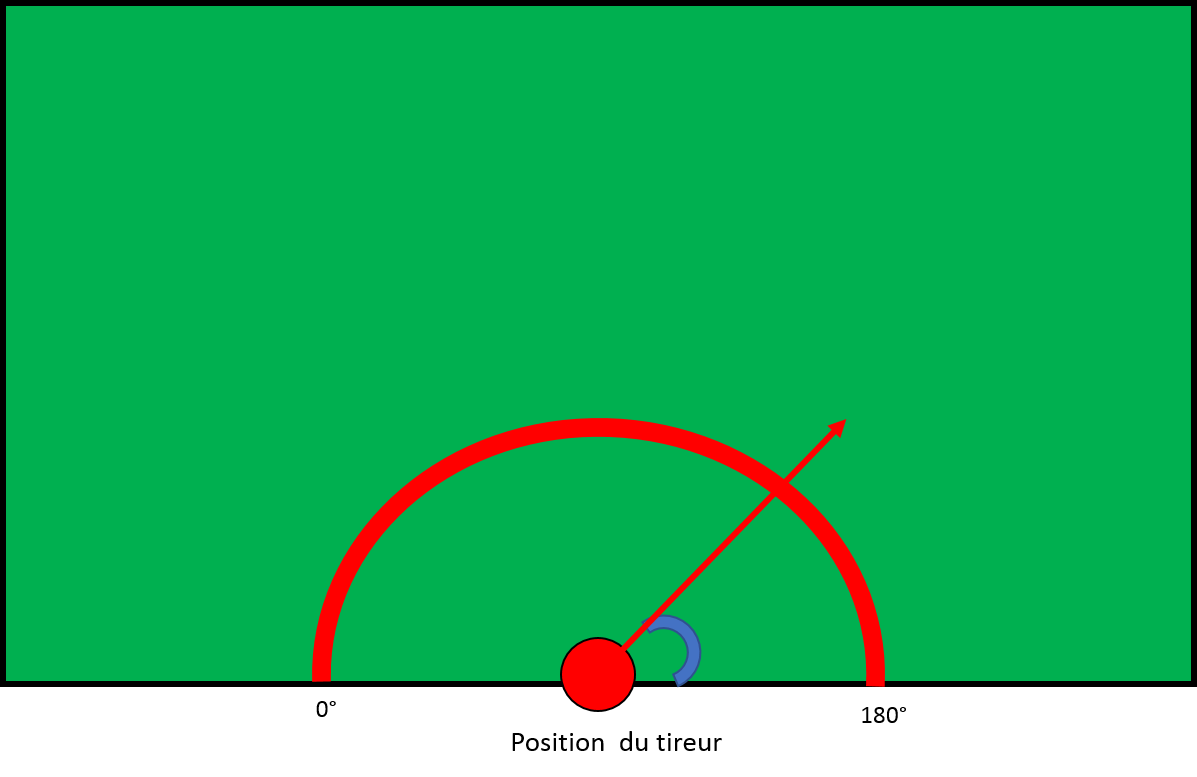
L’objet « tireur » doit en effet pouvoir visualiser sur 180° l’environnement qui l’entoure pour pouvoir atteindre les cibles positionnées en dehors du champ de vision de la caméra

Afin de pouvoir assurer la rotation sur son axe du tireur uniquement sur 180° depuis la position initiale on a utilisé des conditions if.

Il nous a donc fallu utilisé un recalibrage les valeurs d’angle pour pouvoir les utiliser avec les coordonnées polaires des objets.

De plus l’objet représentant le tireur ne devait « faire qu’un » avec la caméra. C’est-à-dire que provoquer une rotation de l’objet « tireur » devait provoquer simultanément la rotation de la caméra (du point de vue)

Temps estimé : 3 heures



3° Troisième problématique : positionnement de cibles aléatoirement.

On a décidé de pouvoir générer au début de notre programme un nombre que l’on choisit arbitrairement. Notre programme devait donc générer aléatoirement n objets cibles sur la carte.

Pour cela nous avons donc initialisé toute les variables nécessaires dans des tableaux et on ajouter un paramètre int aux fonctions init et draw pour générer et numéroté les objet cible(l'entier int correspond au numéro de case des tableaux associé à l’objet)

La position des cibles sur le rectangle (la carte) ne nous semblait pas important, c’est pourquoi nous voulions rendre leur position aléatoire.

Pour l’aspect graphique des cibles, nous avons importé un modèle d’éléphant, triangularisé et réduit à l’aide du logiciel Blender , auquel nous avons ajouté une texture.

Temps estimé : 6 heures



4° Quatrième problématique : projectiles lancés par le « tireur »

Il était nécessaire de créer un projectile qui serait lancé lorsque l’utilisateur utilise la commande de « tir ». Nous avons donc imaginé qu’un objet soit lancé dans la direction dans laquelle regarde le « tireur ». Il nous fallait donc simuler une trajectoire rectiligne pour le projectile.

Pour cela nous avons réalisé une boucle while en utilisant les coordonnées polaires et en augmentant progressivement le rayon.

Ensuite on avait pour ambition d’avoir 3 balles différentes pour avoir la possibilité que même si une balle est encore en chemin cela nous empêcherait pas tirer encore 2 fois.

On a cependant pas réussi car il nous restait pas assez de temps par rapport aux modifications à apporter au programme…

En effet le problème vient de la trajectoire de nos balles par des boucles while car celles-ci monopolisent le programme qui est « bloqué » tant que la boucle n’est pas fini.

On aurait du utiliser une incrémentation des balles dans une fonction qui est appelé en boucle dans le temps par les fonctions de la librairie glut pour espérer avoir 3 balles indépendantes différentes….

Temps estimé : 3 heures + 4heures non concluantes

5° Cinquième problématique : collisions entre projectiles et cibles.

Pour les collisions nous avons juste comparé la trajectoire du projectile, et les coordonnées de la cible. Si les cibles se trouvent sur le chemin du projectile, alors la cible est touchée. Il nous fallait donc les supprimer, pour montrer à l’utilisateur que celui-ci a touché une cible.

Puisque nous n’avons pas réussi à les enlever de la fenêtre, nous avons décidé de faire disparaitre les cibles en les mettant sous la carte, ainsi ils sont invisibles par l’utilisateur.

En effet la collision prenant en compte les 3 dimensions, une fois la cible descendue elle n’est plus un obstacle pour la balle.

Temps estimé : 2h30



Améliorations

Nous avons conscience que notre jeu n’est pas optimal, et qu’il est possible d’y ajouter de nombreuses améliorations.

Une amélioration est donc de gérer comme il le faut les balles pour les utilisé simultanément sans que le jeu « bloque »

De plus dans notre programme les cibles sont toutes situées dans le même plan, il serait bien qu’elles puissent de trouver dans d’autres plans. Il faudrait alors ajouter d’autres rotations du tireur pour qu’il puisse déplacer son viseur dans d’autres directions.

Ensuite il serait bien d’ajouter un compteur pour que l’utilisateur sache combien de cibles il doit encore éliminer, ainsi qu’un système de niveaux de difficultés ou un mode survie où les éléphants se rapproche et ne doivent pas attendre l’utilisateur...

Le décor que nous avons proposé est très minimaliste, il aurait été bien de créer un véritable univers, avec des décors qui donneraient l’impression au tireur d’être dans un stand de tir.

Conclusion

Dans ce projet, nous avons donc réaliser un jeu très simple de « stand de tir », qui nous a permis de développer de nouvelles compétences dans les outils graphiques, le développement en C++, ainsi que la synthèse d’images.

Nous avons trouvé un véritable intérêt, et un vrai plaisir dans ce projet qui nous permis de découvrir un domaine de l’informatique : la synthèse d’image qui nous était relativement étranger.