

Compte-rendu du projet de traitement et synthèse d'image

PRIGENT Hugo - VINCENT Louis

Introduction:

Dans le cadre du projet de traitement et synthèse d'image, nous avons choisi de reproduire à notre façon le jeu SUPERHOT développé par Superhot team. Le principe du jeu est d'éliminer tous les ennemis d'un niveau à l'aide d'armes trouver dans ce dernier. La spécificité du jeu est que la vitesse d'écoulement du temps dépend des déplacements du joueur.



figure (1) - Image du jeu SUPERHOT

Présentation du jeu

Notre jeu reprend le principe de SUPERHOT, il s'agit d'un jeu de tir dans lequel l'écoulement du temps dépend des mouvements du joueur.

Il consiste à essayer d'éliminer les ennemis dans une carte cubique au style hangar industriel. Certains ennemis sont armés et tous se déplacent en direction du joueur cherchant à l'éliminer.

La vitesse de déplacement des ennemis et des balles tirés par pistolet dépendent de la vitesse du jeu déterminer par :

- Le joueur tourne la caméra
- Le joueur se déplace
- Le joueur sprint

Si ces 3 conditions sont respectées, la vitesse du jeu est donc au maximum.



figure (2) - Image de notre jeu

Partie 1: Gestion de la carte et remarques générales

Le jeu se déroule dans une carte cubique de 50 x 50 par qui a un aspect gris grâce à une texture style béton figure(3).



figure (3) - Texture de la carte

La forme cubique est définie à l'aide de 8 points, on construit les faces en mettant dans un tableau chacun de leurs points, normale, couleur et texture. La carte contient aussi trois containers former à l'aide de 2 objets cube.obj collé et sur lesquels on a une texture représentant un container (figure(4)).



figure (4) - Container

Côté pratique, étant donné que plusieurs objets identiques sont importés, nous passons par un seul VAO.

Partie 2 : Gestion de la caméra et du personnage joueur

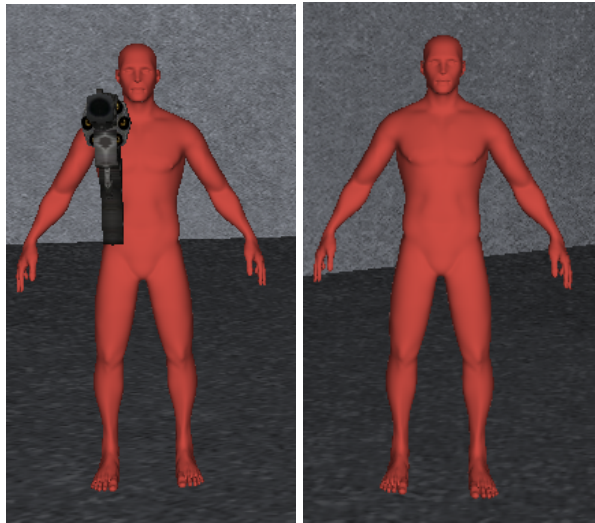
La gestion de la caméra se fait de la même manière que les FPS (First Person Shooter) classiques. Elle suit donc les déplacements du curseur de la souris symbolisé ici par un viseur. Le personnage joueur est équipé d'un pistolet et se déplace avec les touches ZQSD. La touche Shift permet de courir. Bouger la caméra fait aussi bouger le pistolet autour du joueur. Lorsqu'on tire avec le pistolet, l'angle de rotation du joueur est récupéré et la balle part en ligne droite dans cette direction. Le pistolet est équipé de 6 balles. Si ces 6 balles sont tirées, le joueur doit attendre que l'une des ces balles soit détruite par collision pour pouvoir tirer à nouveau.



figure(5) - Pistolet, viseur (en violet) et balle (en vert).

Partie 3 : Gestion des personnages non joueurs

Il y a 6 personnages non joueurs dans notre jeu dont trois armés du même pistolet que le joueur. Ils apparaissent aléatoirement dans la carte à une distance de minimum 25 unité de distance par rapport au centre. Ces personnages sont modélisés par un objet 3D ayant l'allure figure (6).



figure(6) - Personnage armé et personnage non armé

Dès leur apparition, ils se déplacent à la vitesse du jeu en direction du joueur. Pour cela, on récupère leur position et celle du joueur pour obtenir la direction dans laquelle ils devront avancer. Une fois la direction trouvée on modifie la position de chaque personnage pour qu'elle converge vers celle du joueur en prenant en compte la vitesse du jeu.

Partie 4: Gestion de la vitesse du jeu

La vitesse du jeu est donc déterminée par trois critères : le déplacement de la caméra, les mouvements du joueur, le sprint. A chaque itération de la boucle principale, on teste ces trois critères. Tous les déplacements des ennemis et des balles seront multipliés par un coefficient initialisé 0. Le mouvement de la

caméra ajoutera 10% à ce coefficient, les déplacements du joueur ajouteront 40%, et enfin le sprint ajoutera 50%.
Si ces trois conditions sont respectées, le coefficient sera égal à 1 et la vitesse sera maximale.

Partie 5 : Gestion des collisions

Pour ce qui est des interactions et des collisions, nous avons décidé d'attribuer des hitboxes pour chaque objet d'intérêt.
Le joueur, les personnages non joueurs et les balles possèdent une hitbox de forme parallélépipédique calculée à partir des points d'extrémité haute P2 et d'extrémité basse P1 de l'objet à la façon de la figure(7).

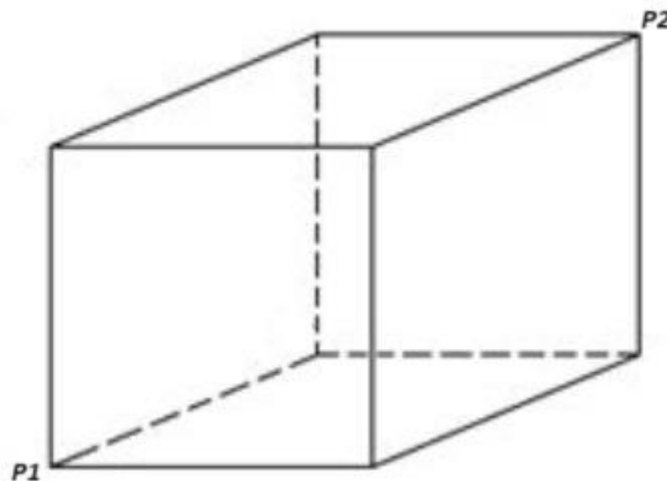


figure (7) - Modèle d'une hitbox

Ces points sont déterminés dans le mesh. Ainsi les collisions et interactions entre les objets se font à partir des coordonnées de ces points.
Les balles étant petites, on les considère comme des points pour la détection de collision.

Conclusion:

Travailler sur ce projet nous a permis d'approfondir nos connaissances de l'API OpenGL. Notre jeu est fonctionnel mais certains aspects restent à améliorer, si le temps accordé était plus conséquent, nous aurions amélioré le mécanisme de déplacement des personnages non joueurs, de sorte à ce qu'ils ne restent pas bloqués par les obstacles.