Projet GP Engine

I. Choix de l’API graphique

Dans le cadre de ce projet, nous avions le choix entre 3 APIS : OpenGL, DirectX 12 et Vulkan.

Nous avons éliminé DirectX. Bien que plus performant qu’OpenGL, cet API ne fonctionne uniquement sur Windows, ce qui ne représente pas un bon avantage.

Techniquement, Vulkan étant un API adapté pour les moteurs des générations futures, il aurait été un bon choix. En effet, sa faible utilisation du CPU procure un avantage considérable en terme de fluidité et de performances du moteur.

Cependant, Vulkan est une API relativement nouvelle et pas assimilé pour nous. Nous n’avons pas connaissance des éventuels problèmes techniques et/ou de conception durant notre travail. Afin de minimiser ces risques, nous avons préféré recourir à OpenGL, que nous connaissons bien.

II. Choix des librairies à utiliser

En ce qui le dossier toutes les librairies seront enregistrées dans un dossiers libs et dans des sous dossiers à leur nom respectif .

1. GLUT

Cette librairie OpenGL contient toutes les fonctionnalités de création de fenêtre de base pour notre application. *(OpenGL Utility Toolkit)* Sa simplicité permet de ne pas se préoccuper de la partie système, relativement complexe, et de se concentrer sur le fond du programme OpenGL lui-même.

2. GLEW

Librairie qui s’occupe du chargement des extensions d’OpenGL. *(OpenGL Extension Wrangler Library)*
Elle apporte une simplification sur l’utilisation des différentes fonctions d’OpenGL, que ce soit du core ou des extensions.

3. Qt

Pour l’interface graphique de l'éditeur nous utiliserons QT car il s’agit de l’API la plus utilisée en ce qui concerne les interfaces graphique .L’un des avantages de ces API est son arborescence des objets qui s’organise d’eux même sous forme d’arbre, ce qui facilite la

gestion mémoire car avant qu'un objet parent ne soit détruit, Qt appelle récursivement le destructeur de tous les enfants .De plus il sera implémenté dynamiquement .

4. PhysX

En ce qui concerne le moteur physique nous avons décidé d'utiliser PhysX. Tout d'abord, ce moteur est consacré au rendu des jeux, et permet de choisir entre faire fonctionner un jeu via CPU ou via le GPU.
Au vue des résultats de multiple test "PEEL", PhysX est plus performant que Bullet si l'application développée utilise le GPU.

PhysX est une librairie dynamique nécessite de une à trois .dll en fonction de l'utilisation.
Son initialisation simple renvoie un pointeur qu'il est facile à encapsuler.
D'un point de vue moteur, PhysX, inclut un système de scène et d'acteur physique. Dans un loopPhysiX, la scène est passé en revue et les effets physiques sont appliqués aux acteurs de cette dernière.

5. SDL

Le rendu en jeu, média et gestion d'input sera effectuée en utilisant la SDL.
Choix avant tout motivé par des expériences d'utilisation passée ainsi que par ces multiples possibilités :

- l'affichage vidéo ;
- les événements ;
- la gestion des périphériques communs comme le clavier et la souris mais aussi le *joystick* ;
- l'utilisation de *timers* précis.
- l'audio numérique

De plus il nous permet de travailler avec OpenGL .