

Problématique

La création de rendus itératifs 3D (*3D forward rendering*) nécessite une compréhension approfondie des appels OpenGL de bas niveau et de la communication avec la carte graphique.

Ce n'est généralement pas à la portée des petits projets d'inclure ce genre de travail approfondi.

De plus, les engins existants sont très lourds, nécessitent des machines très puissantes et sont difficiles à prendre en main.

Il y a donc un important besoin dans l'industrie pour un moteur de rendu léger.



Méthodologie

Abstraction OpenGL : Création de classes pour abstraire les appels OpenGL de bas niveau.

Chargement de textures : Implémentation du chargement de textures, y compris cubemaps et skyboxes.

Chargement de modèles : Utilisation d'Assimp pour le chargement de modèles 3D.

Gestion des shaders : Compilation et utilisation de multiples shaders pour divers effets.

Système d'éclairage : Support de multiples sources lumineuses avec un éclairage de Phong

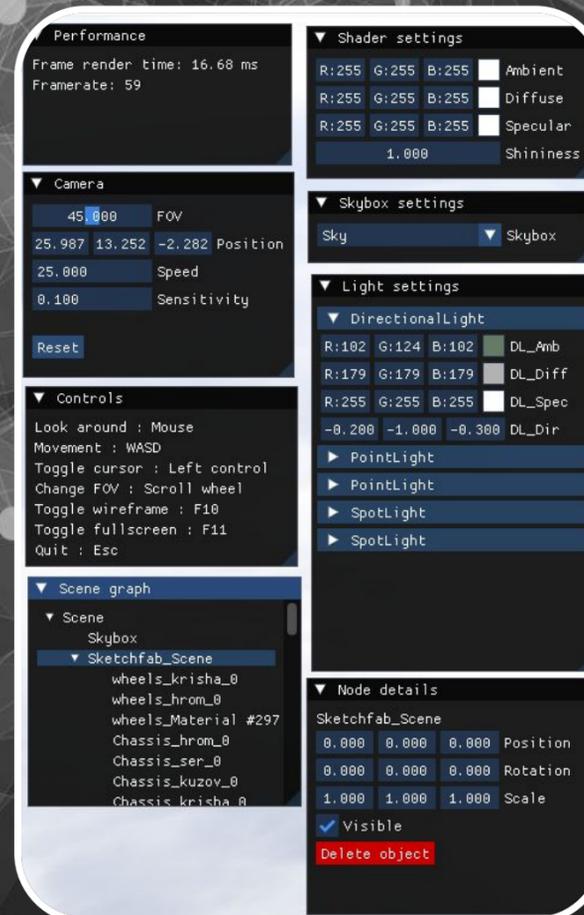
Objectifs

Construire un moteur de rendu 3D itératif (*3D forward renderer*) en utilisant C++ et OpenGL.

Architecturer une librairie avec un bon niveau d'abstraction des appels OpenGL.

Interface utilisateur

- Implémenter une interface pour l'interaction dynamique avec les éléments du rendu : scène, lumières, skybox, et shader.
- Implémenter un système de caméra et d'éclairage dans un environnement 3D



Résultats

Le moteur de rendu développé a permis de réaliser un rendu 3D basique avec des fonctionnalités de chargement de textures et de modèles, de gestion des lumières, et d'interactions via une interface utilisateur.

Bien qu'il ait été conçu comme un outil pédagogique plutôt que pour une application professionnelle, le projet a atteint plusieurs objectifs clés dans :

- l'abstraction des appels OpenGL,
- le chargement des modèles,
- et la manipulation des shaders.

SCANNEZ ce QR code pour essayer le moteur VectorGL



<https://github.com/razor7877/VectorGL>