Epitech Rennes - Semestre 4 2023
Proposition de Projet Hub
Imagerie numérique, 3D
Guillaume Papineau

Moteur de rendu 3D Graphique : Rastérisation

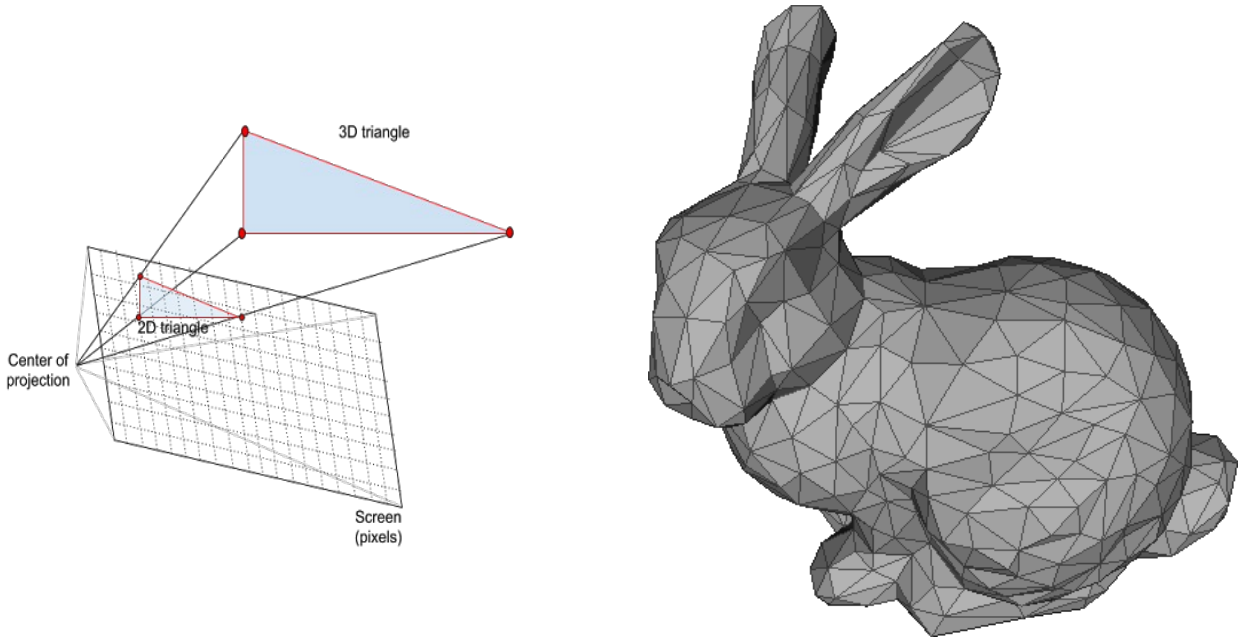

Figure 1: Exemple de Rastérisation. À gauche: Visuelle technique. À droite: Rendue possible.

Contexte et but du projet

Le but de ce projet est d'**implémenter la méthode de rastérisation** utilisé par de nombreux moteurs 3D pour sa rapidité et ses possibilités comparé au Ray Tracing.

Mon objectif final est de pouvoir proposer un rendu 3d graphique de n'importe quel objet importé ainsi que de pouvoir me déplacer pour l'observer sous d'autres angles en temps réel.

Porteurs du projet

Actuellement **en deuxième année**, je serais **le seul étudiant sur ce projet**. Je suis intéressé le domaine de l'infographie et des jeux vidéo en général. Je Souhaite donc en apprendre plus sur les techniques utilisés dans ces domaines.

Environnement technique / technologique

Mon projet utilisera **CSML,** une librairie graphique indépendante de langage et cross-plateforme, permettant une implémentation rapide des fenêtres de rendu et des input clavier/souris. Je compte utiliser cette librairie en **C**.

Description du livrable

Mon projet sera disponible sur Linux uniquement. La **CSFML est un prérequis** pour la compilation (via Makefile) ainsi que le lancement du programme. De plus, le projet sera **open-source** et sera disponible sur la plateforme Github.

Un fichier .obj d'exemple sera mis à disposition dans le dossier assets/obj_examples avec les textures associés dans le dossier assets/textures.

Organisation et temporalité

J'ai divisé mon projet en 7 parties distinctes:

Nom de la tâche	**Commentaire**	**Estimation** **Chaque couleur = 25% du projet**
Recherche et Documentation		2 j/h
Parsing de fichier .obj	Lecture d'un fichier .obj pour récupérer la liste des triangles constituant l'objet	3 j/h
Projection matricielle	C'est la projection des différents modèles de l'image vers le plan 2D : *- Model to World(M2W)* *- World to View* (*W2V*) - View to Projection(V2P)	4 j/h
Clipping de Cohen-Sutherland	Troncage des triangles en cas de sortie du champ de l'écran	3 j/h
Implémentation en 3D avec caméra se déplaçant dans l'espace	La caméra pourra être déplacée pendant la simulation grâce aux événements claviers (et souris dans un moindre cas)	2 j/h
Texture mapping	Parsing des textures, et gestion des coordonnées de textures	4 j/h
Z-buffer/Deph-buffer	Déterminer quels éléments de la scène doivent être rendus, lesquels sont cachés par d'autres et dans quel ordre l'affichage des primitives doit se faire.	3 j/h
		Total: 21 J/H