Epitech Rennes - Semestre 4
2023 Proposition de Projet Hub
Algorithmique, 3D

Guillaume Papineau

Moteur de rendu 3D Graphique : Collision et Optimisation

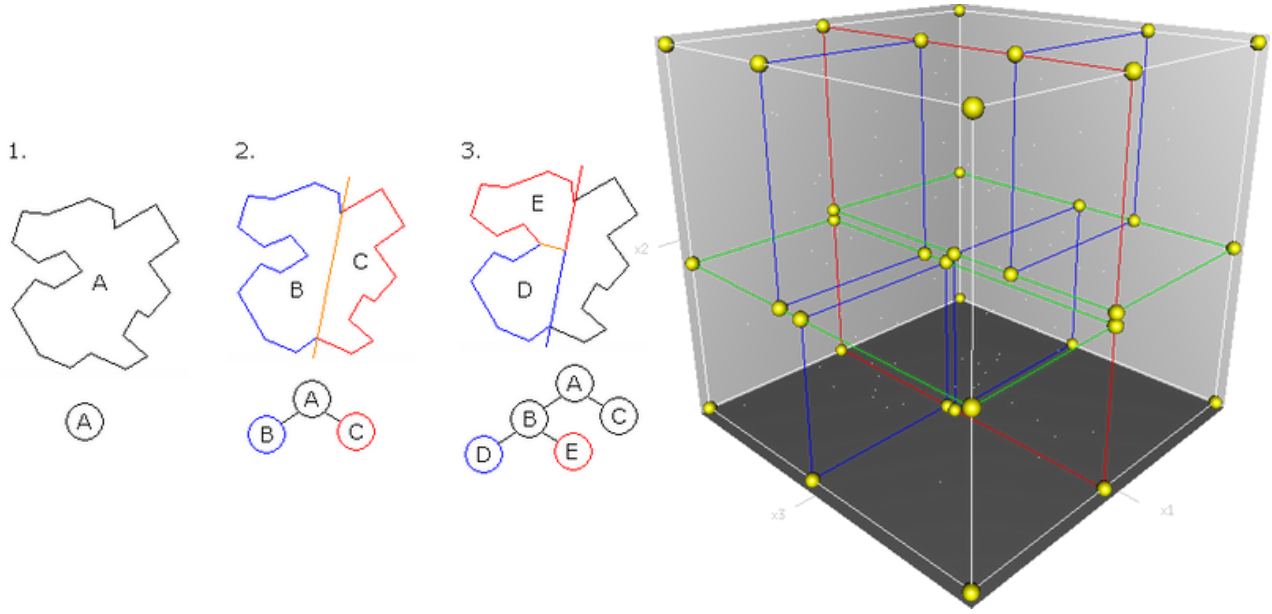

Figure 1: Exemple de gestion de partitionement d'une zone par *BSP.
À gauche: Schéma technique. À droite: Rendue de partitionnement 3d.

Contexte et but du projet

Le but de ce projet est d'implémenter un système de collision premièrement basé sur la méthode de partitionnement de l'espace (*BSP) utilisé par de nombreux moteurs 3D de l'époque. Et ensuite je souhaite ajouté d'autres algorithmes pour venir renforcer les performances et augmenter l'optimisation de la méthode première et du programme.
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 troisième année, je serais le seul étudiant sur ce projet. Je suis intéressé par le domaine de l'infographie et l'algorithmie en général. Je souhaite donc en apprendre plus sur les techniques utilisées dans ces domaines.

Environnement technique / technologique

Mon projet ce basera sur le travaille déjà effectué lors du projet "Moteur de rendu 3D Graphique : Rastérisation", permettant un test rapide du développement du systhème de collision par la fenêtre de rendu ainsi que les inputs clavier/souris. Ainsi l'implémentation restera en C.

*BSP: Binary space partitioning

Description du livrable

Le projet sera disponible dans sa nouvelle version sur Linux uniquement.
La CSFML est un prérequis pour la compilation (via Makefile) ainsi que le lancement du programme. De plus, le projet sera toujours open-source et est toujours disponible sur la plateforme Github à ce lien: https://github.com/MasterLaplace/Engine-3D.
Des fichier .obj d'exemple seront 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, documention et ajout dans le pipeline		2 j/h
Implémentation de la méthode de Partition binaire de l'espace	pour créer un arbre pour chaque mesh et y partitioner les triangles	4 j/h
Recherche et Documentation		1 j/h
Convertion du binary tree en QuadTree	pour améliorer la vitesse de traitement des collisions, soit le temps de recherche dans l'arbre	3 j/h
Recherche, documentation et mise à jour du pipeline		2 j/h
Mise en place d'un World Partitionning au niveau des meshs	pour charger les meshs uniquement dans le chunck où se situe la camera	5 j/h
Mise en place d'un QuadTree par chunck	pour organiser les meshs au sein du chunck et permettre une amélioration dans la sélection des meshs à surveiller	3 j/h
		Total: 20 J/H