université PARIS-SACLAY | FACULTÉ DES SCIENCES D'ORSAY

Informatique Graphique pour la Science des Données

Des pylônes dans les collines

Rapport

October 20, 2024

HAJJAR Line et RAVENDIRANE Gayathiri
Licence LEU Informatique, L2 S4 G02
Année Universitaire : 2023−2024

1 Description du Projet

Le terrain utilisé dans notre projet est représenté par un modèle nommé *"HYPERSIMPLE"*. Ce paysage virtuel présente des variations d'altitude et des reliefs, définis par des points situés entre −202.09592 et 179.59933 sur l'axe z, et entre −135 et 127 sur l'axe x, ainsi que entre −158 et 159 sur l'axe y. En plus de ce paysage, nous avons ajouté un village intégré dans cet environnement, offrant ainsi une dimension supplémentaire à notre scène graphique. Ces caractéristiques permettent de créer un environnement visuel réaliste pour nos expérimentations graphiques et de simulation.

2 Utilisation

Dans notre projet, nous avons intégré des interactions utilisateur à l'aide des keylisteners et du mouse pour permettre une navigation et un contrôle fluides de la scène. Voici comment nous avons configuré ces interactions :

2.1 Keylisteners

Nous avons utilisé des keylisteners pour détecter les pressions de touches spécifiques et déclencher des actions correspondantes dans notre application graphique. Voici quelques-unes des touches que nous avons utilisées et leurs actions associées :

- **Flèches directionnelles (Haut/Bas/Gauche/Droite)** : Contrôle de la navigation dans la scène en ajustant la vue de la caméra.
- **Touche 'Z'** : Zoom avant pour rapprocher la vue de la caméra.
- **Touche 'S'** : Zoom arrière pour éloigner la vue de la caméra.
- **Touche 'A'** : Zoom arrière pour tourner vers gauche la vue de la caméra.
- **Touche 'D'** : Zoom droite pour tourner vers droite la vue de la caméra.
- **Touche 'Espace'** : Lecture ou arrêt de la musique.
- **Touche '2'** : Affichage ou masquage du shader.
- **Touche '3'** : Affichage ou masquage des pylônes et des lignes.
- **Touche '4'** : Affichage ou masquage des lignes électriques.
- **Touche '5'** : Affichage ou masquage des pylônes.
- **Touche '6'** : Affichage ou masquage de la ligne entre l'éolienne et le centre.

2.2 Mouse

Nous avons également intégré des interactions avec la souris pour permettre un contrôle précis lors de l'exploration de la scène et de l'interaction avec les objets. Voici quelques interactions que nous avons implémentées avec la souris :

- **Molette de la souris** : Zoom avant/arrière pour ajuster le niveau de zoom de la caméra.

3 Résumé du travail effectué

- **Affichage du terrain et repère 3D** : chargement et affichage du terrain avec un repère composé des axes x, y et z. Utilisation du modèle "HYPERSIMPLE" pour représenter le terrain.
- **Lignes de niveau** : en utilisant un shader GLSL, des lignes de niveau sont tracées sur le terrain. Chaque ligne est colorée en fonction de la hauteur z interpolée, ce qui crée un effet visuel distinctif.
- Les **pylônes** sont ajoutés à la scène comme éléments structuraux. Ils sont représentés en gris foncé à l'aide d'un modèle PShape.
- Des **lignes électriques** sont générées pour relier les points d'attache sur les pylônes. Ces lignes sont flexibles et leur courbure dépend de la distance entre les pylônes.
- Des **éoliennes** animées sont positionnées au sommet des collines. Elles utilisent des courbes de Bézier pour leur animation et sont reliées à un village dans la vallée par des lignes électriques.
- D'autres éléments sont intégrés à la scène, tels que des **maisons**, des **arbres**, des **fleurs**, une table, et des **villageois** pour enrichir l'environnement graphique.

4 Organisation du travail

Nous avons principalement utilisé deux méthodes de collaboration pour travailler ensemble sur ce projet. Dans un premier temps, on avait fait la première partie, 'le terrain', ensemble. Ensuite, on a partagé le travail en deux sous-parties. Pour bien mener ce projet, nous avons utilisé diverses méthodes de collaboration. Le “pair programming” a été notre approche principale, où chaque membre du groupe a contribué à la rédaction du code. Même lorsque nous avons écrit des classes séparément, nous avons systématiquement passé en revue le code de l'autre pour le déboguer ou l'améliorer. Nous avons utilisé des sessions G-meet et également pratiqué la programmation côte à côte “virtuel”. Concernant la gestion du code source avec Git, nous avons travaillé sur des branches distinctes pour éviter les conflits de versions.

5 Organisation des classes et des méthodes

Nous avons structuré notre code en utilisant des classes pour représenter différentes entités graphiques telles que le terrain, les pylônes, les lignes électriques et les éoliennes. Chaque classe a été conçue pour encapsuler des fonctionnalités spécifiques et faciliter la modularité du code.

- **main.pde** : Point d'entrée principal de l'application. Il charge et initialise l'environnement graphique, crée les instances des classes nécessaires et gère le cycle principal de rendu.
- **Interaction.pde** : Fournit la gestion des interactions utilisateur via les keylisteners et les mouvements de la souris. Il implémente les fonctionnalités pour contrôler la caméra et les divers affichages de la scène.

- **Pylone.pde** : Implémente la classe Pylone pour la représentation et le rendu des pylônes dans la scène. Il gère les propriétés et le comportement des pylônes.
- **ElectricLines.pde** : Contient la classe ElectricLines pour générer et afficher les lignes électriques entre les pylônes.
- **Eolienne.pde** : Définit la classe Eolienne pour représenter les éoliennes animées et leurs interactions avec la scène, y compris les animations avec les courbes de Bézier.
- **Sound.pde** : Ce fichier gère les fonctionnalités audio de l'application telle que des fonctions pour démarrer, arrêter et ajuster le volume des fichiers audio en fonction de la distance entre le joueur et les éléments interactifs (éoliennes, pylônes).
- **Flower.pde** : Il contient les fonctions nécessaires pour dessiner une fleur avec des pétales colorés en rose.
- **TreeAndApples.pde** : Il contient les fonctions nécessaires pour dessiner un arbre réaliste avec des branches et un tronc.

6 Prise de recul

En réfléchissant sur le développement de notre projet, plusieurs points méritent d'être mentionnés :

- Pour la gestion des objets 3D tels que les pylônes et les éoliennes, nous avons simplifié l'accès aux composants en évitant les méthodes standard comme getChildCount(), getChild(), getVertexCount() et getVertex(). À la place, nous avons développé une fonction personnalisée utilisant quadraticVertex.
- Nous avons intégré des éléments sonores pour enrichir l'expérience utilisateur. Plus précisément, nous avons mis en place un système où l'intensité de la musique augmente lorsque l'utilisateur s'approche d'une éolienne ou d'un pylône. Cela crée une immersion supplémentaire et ajoute une dimension sensorielle à notre simulation.
- Nous avons identifié un défi technique lié à la rotation de la caméra lors des commandes de mouvement à gauche ou à droite (les flèches). On a des comportements inattendus.
- Nous avons utilisé des courbes de Bézier pour dessiner les éoliennes. De plus, l'éolienne est animée en utilisant une rotation progressive autour de son axe central. À chaque trame (frame), l'angle de rotation est calculé et appliqué aux pales de l'éolienne, ce qui crée un effet de rotation fluide. Cette rotation est contrôlée par la variable rotationAngle, qui est mise à jour à chaque itération de la boucle draw() en fonction du temps écoulé depuis le début de l'exécution du programme.
- En ce qui concerne les éléments visuels, nous avons également employé 'curveVertex' pour dessiner les fleurs, offrant ainsi des formes organiques et naturelles.

7 Conclusion

En conclusion, ce projet nous a permis d'apprendre à programmer en utilisant Processing et les shaders GLSL. Ces outils nous ont aidés à créer des graphismes et des effets visuels pour notre projet. Grâce à cette expérience, nous avons acquis de nouvelles compétences pratiques en programmation graphique, ce qui sera bénéfique pour notre développement futur dans le domaine de l'informatique.