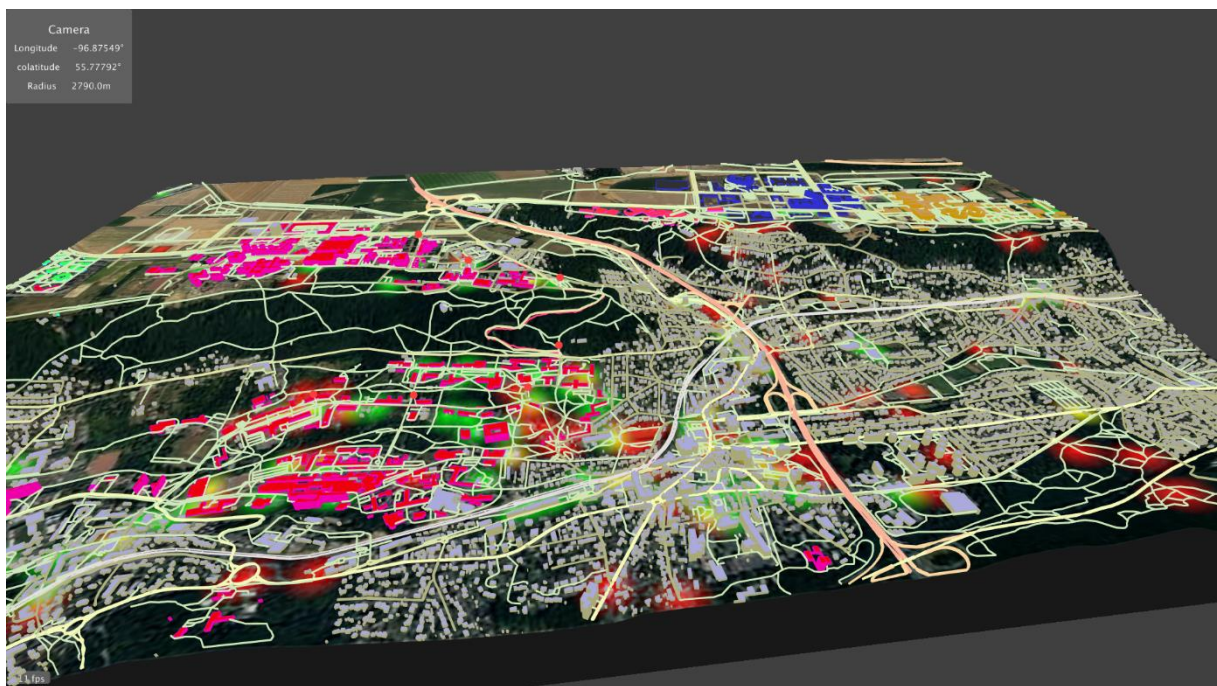


RAPPORT DE PROJET :

Représentation 3D de l'université

Paris-Saclay



PLAN :

- I) Mise en place de l'espace de travail
- II) Modélisation du terrain
- III) Affichage des voies
- IV) Tracé des bâtiments
- V) Carte de chaleurs
- VI) Conclusion

- I) Mise en place de l'espace de travail

Pour commencer le projet sur de bonnes bases, il a été nécessaire de créer une grille ainsi qu'un « gizmo » pour que nous puissions nous repérer lors de l'ajout de objets.

Dans mon projet, le repère est un repère orthonormé dans lequel chaque axe est de longueur 100.

Suite à cela, j'ai créé une grille qui est une PShape de type QUADS avec chaque maille de longueur 100. Vous trouverez le code chargé de la création de la grille dans le fichier Workspace du projet.

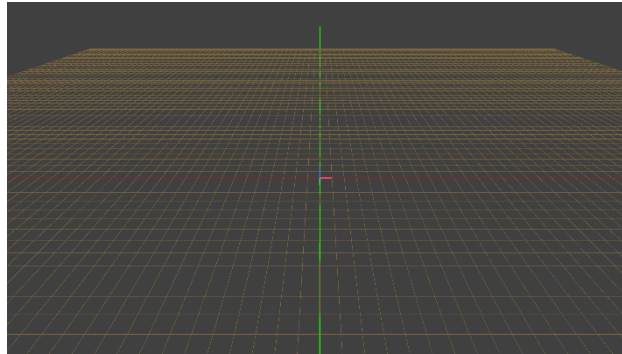


Image 1 : grille de repérage+ gizmo

Enfin, il m'a fallu rendre la caméra mobile pour pouvoir naviguer dans la future carte du projet. Pour cela, j'ai implémenté une nouvelle classe « Camera » chargée de pouvoir gérer les changements de points de vue. Notre caméra navigue dans un repère sphérique centré sur l'origine. Notre caméra possède un attribut « radius » qui traduit la distance a laquelle elle se trouve de l'origine, un attribut colatitude dans lequel est stocker la colatitude de notre caméra et enfin, un attribut longitude.

J'ai de plus ajouter une procédure « determineXYZ » chargé de convertir les coordonnées sphériques en coordonnées cartésiennes.

Une fois la caméra implémentée et capable de bouger, il a fallu qu'on puisse interagir avec. J'ai donc implémenté dans le programme principal une procédure « keyPressed ». Cette procédure permet à l'utilisateur de pouvoir grâce à une pression sur les touches haut/bas du clavier de pouvoir ajuster la colatitude de la caméra, sur les touches gauche/droite de pouvoir ajuster la longitude de la caméra et sur « + » / « - » d'ajouter le rayon de la caméra.

Enfin, pour informer l'utilisateur de la position de la caméra, il a fallu implémenter une classe « Hud » qui prends en paramètre une Camera et informe l'utilisateur en temps réel de sa position.

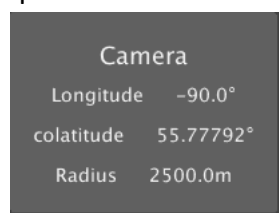


Image2 : hud

Nous disposons maintenant d'un environnement de travail prêt à l'emploi.

II) Modélisation du terrain

Maintenant que nous avons un environnement de travail fonctionnel, il a fallu modéliser le terrain de l'université. Pour cela, je disposais de données topologiques sous la forme d'un fichier .data et d'une classe Map3D qui me permettait de passer des coordonnées sphériques aux coordonnées processing.

Grâce à ces outils, j'ai pu modéliser le terrain en fil de fer. Pour cela, j'ai créé une PShape de type QUADS à laquelle je donnais 4 points espacés de « tileSize ».

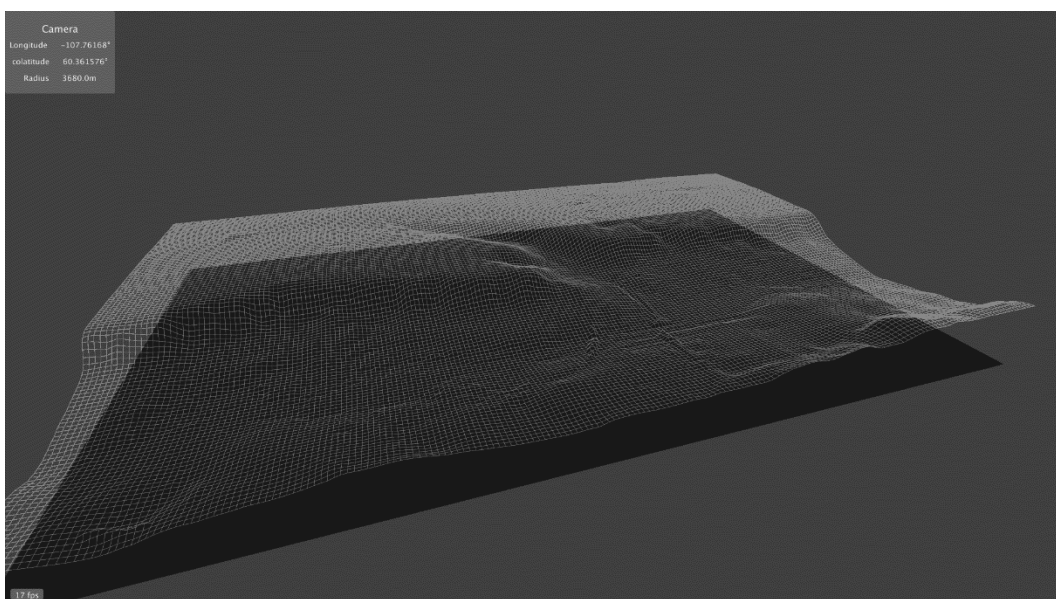


Image 3 : maillage du terrain en fil de fer

Maintenant que nous disposons d'un maillage en fil de fer, je lui ai appliqué la texture fournie pour avoir une vue satellite de l'université.

J'ai enfin modifié la fonction keyPressed pour pouvoir switcher entre le maillage en fil de fer et la vue satellite.

III) Affichage des voies

Une fois que j'avais la vue satellite, il fallait maintenant implémenter un tracé GPS. Pour cela, j'ai utilisé un fichier .GeoJson qui contenait les points clés ainsi que tous les points du tracé GPS. Après les avoir convertis en ObjectPoint (points en coordonnées processing), je les ai donnés à une PShape de type LINE_STRIP ce qui m'a permis d'avoir un tracé GPS. J'ai ensuite placé aux points importants 2 PShape : une de type LINES pour représenter le corps de l'épingle et une de type POINT pour représenter la tête.

Pour ce qui est de l'interaction avec l'utilisateur, j'ai rajouté une variable booléenne dans le programme principal pour dire si un utilisateur avait cliqué sur une épingle. La fonction

« `moussePressed` » renvoyait la position de la souris que je donnais à la fonction `clic`, chargée d'afficher la description au bon endroit.

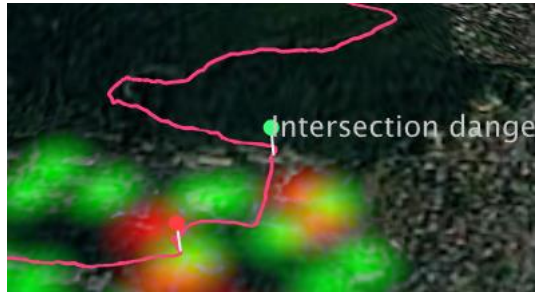


Image 4 : visuel du tracé GPS.

Une fois que j'avais fait ce travail, je l'ai dupliqué pour tracer les axes importants tels que les voies de R.E.R ou les grands axes routiers. Ces derniers sont représentés par des PShapes de type GROUP contenant plusieurs PShapes de type QUAD_STRIP. En faisant une disjonction de cas sur le type de voie, il fut possible de tracer des routes de différentes tailles et différentes couleurs.



Image 5 : cartes des voies

IV) Tracé des bâtiments

L'avant dernière étape du projet est le tracé des bâtiments. Pour ce faire, grâce aux données d'openStreetMap, j'ai pu tracer les bâtiments sous la forme de PShape QUADS en leur donnant une taille de 3 mètres par étages.

N'ayant pas réussi à régler l'émissivité des bâtiments, je me suis contenté d'en modifier la couleur pour que l'on voit les différences entre le toit et les murs.

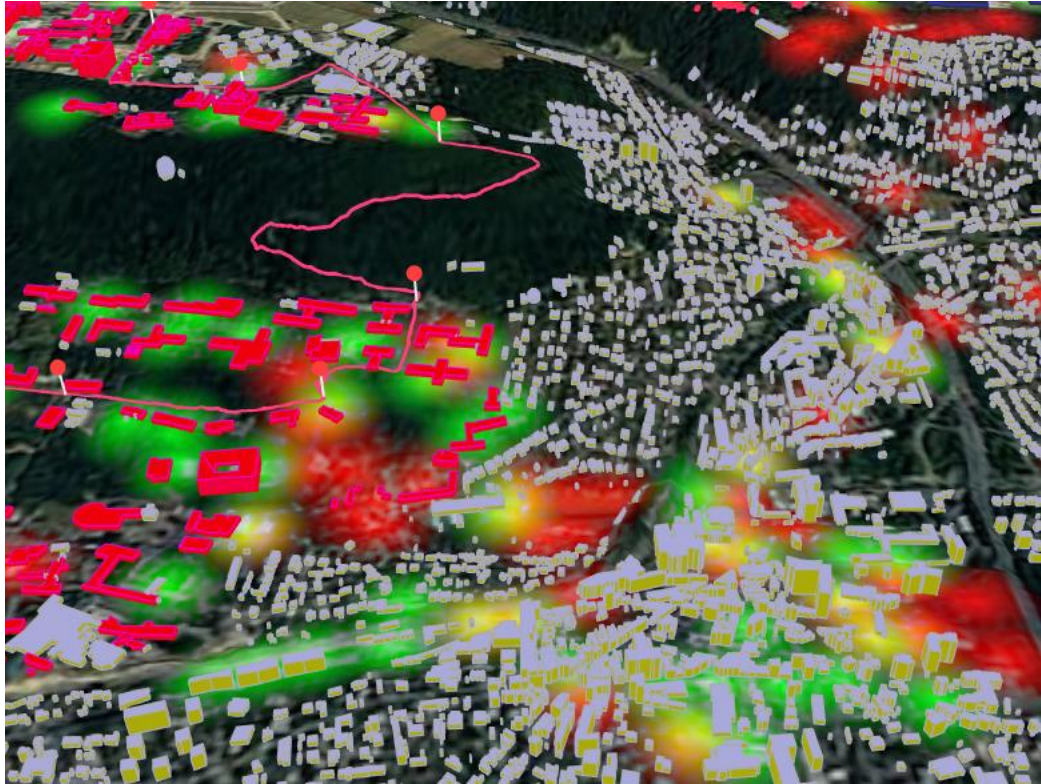


Image 6 : vue des bâtiments

V)

Cartes de chaleur

Pour réaliser les cartes de chaleur, j'ai implémenté une classe Points contenant un tableau de couples qui sont les coordonnées des points d'intérêts données par les fichiers GEOJson convertis en ObjectPoint.

J'ai ensuite rajouté 2 attributs de type Points « bench » et « PicNic » ainsi qu'une méthode pour calculer la distance minimale a un point d'intérêt pour un vertex donné. Enfin lors de la création d'un point de satellite, j'ai ajouté un attribut a ce point contenant la distance au point d'intérêt.

Enfin, dans mon Fragment Shader, j'ai réalisé une interpolation linéaire des couleurs :

$$gl_FragColor.r = (-1.0/100)*vertHeat[0]+1;$$

VI)

Conclusion

Pour conclure, ce projet m'a permis de mettre en pratique tout ce que j'avais appris lors des séances de tp. Il m'a aussi permis de comprendre l'intérêt de l'infographie dans le quotidien et dans de nombreux domaines tels que la représentation de données sur une carte ou la modélisation 3D de modèles pour la robotique par exemple.