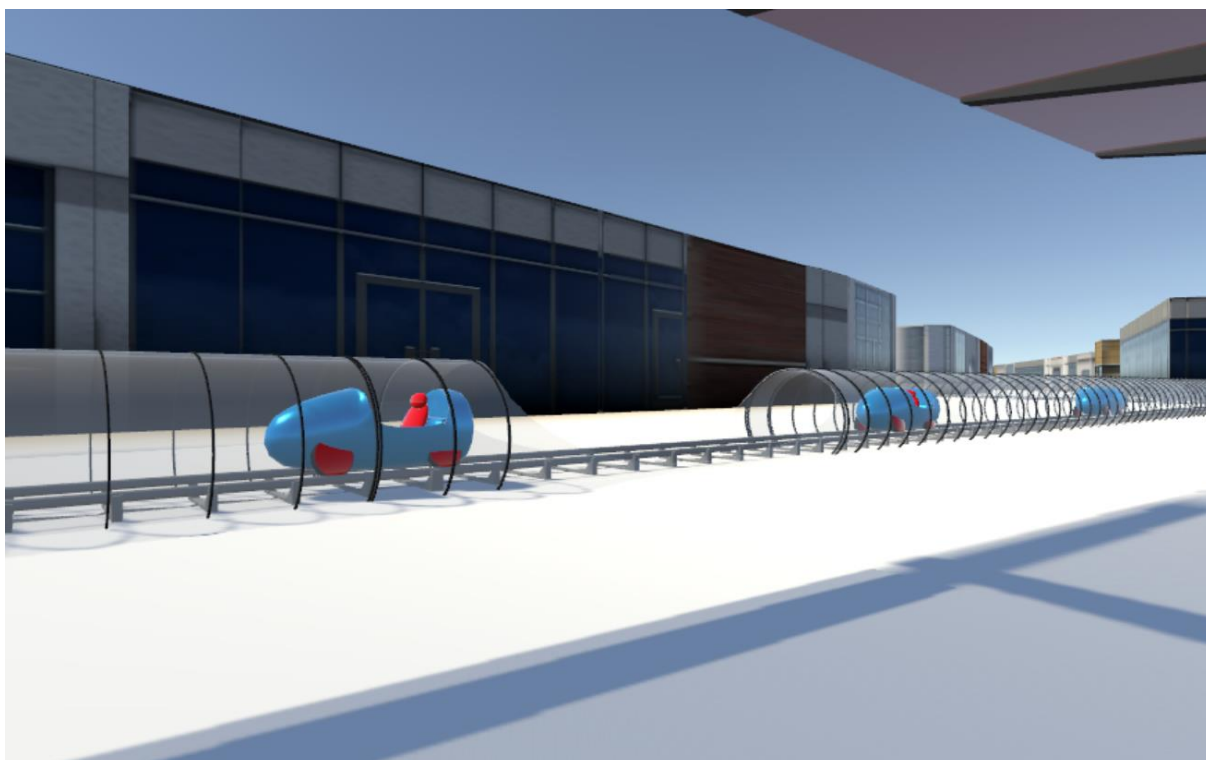


Projet de 2^e année,

Modélisation 3D de l'Urbanloop



Auteur : Léo Hardy

Encadrant : Pascal Vaxivière

SOMMAIRE

Présentation p.3

Modélisation 3D p.4

Choix des outils : Blender et Unity

Semaine 1^o années / 2^o année : capsule et abri

Rails et tuyaux

Textures et materials

Choix des données : Grand Nancy ou Mapbox ?

Assemblage dans Unity p.9

Intégration des objets

Scripts présents dans le dépôt

Animations

Prise de vidéos MP4

Remerciements p.12

Annexes p.13

PRESENTATION

L'Urbanloop est un projet innovant de réseau de transport à travers la métropole nancéenne. Il est constitué d'un réseau de capsules individuelles circulant sur différentes boucles reliées entre elles. Cet ambitieux projet prévoit un coût d'installation relativement faible, un impact sur l'environnement réduit et des temps de transport faibles : 1 minute 30 pour parcourir 2km. Un premier prototype est en construction depuis mars 2019 sur le plateau de Brabois et l'objectif est d'implémenter le réseau de transport dans Nancy d'ici 2024.

Depuis la naissance du projet les écoles d'ingénieurs sont impliquées dans le projet et 100 élèves ingénieurs travaillent chaque année sur des problèmes divers allant de la conception des capsules et du réseau à l'analyse des risques liées à ce mode de transport.

Pour ma part, j'ai travaillé lors de mon projet de Seconde Année à l'Ecole des Mines Nancy sur une modélisation 3D de l'Urbanloop. Cette modélisation pensée avec mes professeurs, avait tout d'abord pour objectif de pouvoir servir comme une vitrine à l'Urbanloop. Après un rendez-vous avec Philippe Mangin, Responsable de projet chez Urbanloop, j'ai changé d'angle d'attaque : comme Urbanloop recevait à partir de février 2019 un stagiaire architecte-urbaniste pour six mois qui allait réaliser des vues d'artiste, mon projet devenait redondant avec son travail. Ainsi il devenait plus intéressant que je m'oriente vers une modélisation du réseau intégrant des contraintes réelles afin qu'elle devienne plus un outil de prévision ou de décision plutôt qu'un objet à visée commerciale.

L'objectif de ce rapport est de faire un état de l'art de l'avancement de ce projet à la date où mon travail s'arrête, à la mi-juin 2019, et de faciliter la prise en main du programme créé par un utilisateur intéressé, ou un futur collaborateur d'Urbanloop souhaitant améliorer mon travail.

Tous les fichiers du travail final sont disponibles sur le dépôt git suivant :

https://github.com/leo-hardy/Projet2A_Urbanloop

Une vidéo de présentation est disponible à cette adresse :

https://drive.google.com/open?id=1RxJYMoGL8w_QDMgl3MsgHX-kJ9yUzIMM

La version d'Unity utilisée est la 2018.1.9.

MODELISATION 3D

Choix des outils : Blender & Unity

Pour réaliser la modélisation du réseau, j'ai été conseillé dès le début d'année d'utiliser le logiciel Unity : initialement prévu pour réaliser des jeux-vidéos, il permet de réaliser des animations d'objets 3D et on peut lui déléguer la gestion des événements physiques (gravité, contact entre objets, ...). De plus, grâce à la grande communauté derrière ce logiciel, de nombreux packages permettent d'obtenir des fonctionnalités avancées, des objets déjà modélisés ou des décors et les forums de discussion sont très complets : à part pour l'intégration du modèle 3D de Nancy j'y ai systématiquement trouvé la réponse à mes questions.



Figure 1 : logo Unity



Figure 2 : logo Blender

Les moyens le plus simple d'ajouter de nouveaux objets 3D dans Unity est de les modéliser à partir de Blender, et c'est ce pour quoi j'ai utilisé ces 2 logiciels.

Je pense que les différents tutoriels (par exemple sur Open Classroom) disponibles sur internet permettent de bien comprendre les outils nécessaires à une maîtrise correcte de Blender.

Semaine 1^{re} années / 2^e année : capsule et abri

Au mois de mai est a été organisée aux Mines une semaine durant laquelle les 2^e années font travailler les élèves de 1^{re} année sur leur projet : j'ai ainsi pu faire travailler 2 groupes de 2 élèves pendant 2 jours. J'avais choisi de les initier à Blender et de leur faire réaliser, au choix : un modèle de capsule, de gare, ou tout autre élément qu'il jugeaient pertinent pour l'Urbanloop.

Ces élèves, Alizée Evrard, Julie Heurtaux, Hippolyte Karakostanoglou et Stanislas Rambaud ont ainsi modélisé les objets suivants : une capsule et tout son mobilier intérieur, et un abri pour les stations.

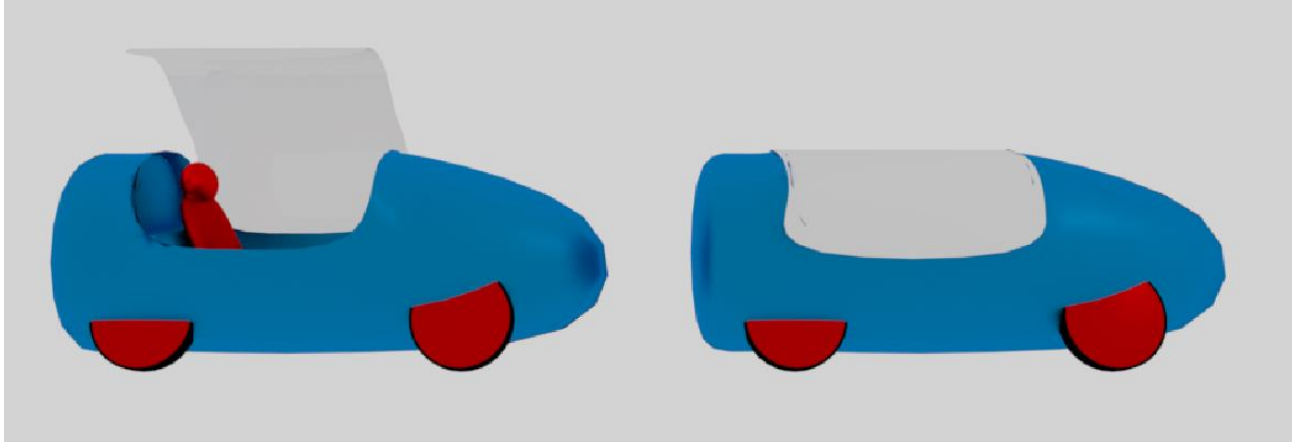


Figure 3 : vue d'une capsule avec le toit ouvert et fermé

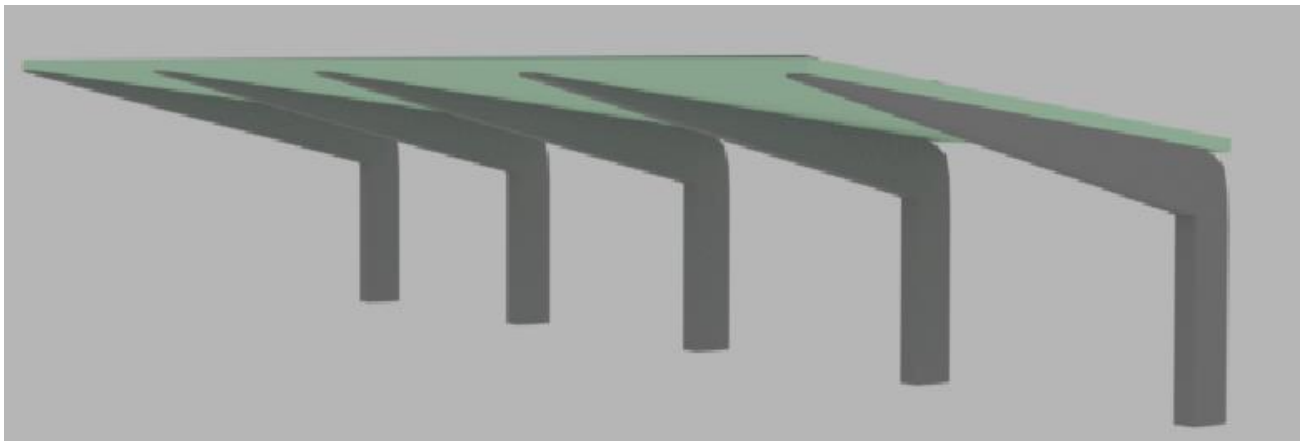


Figure 4 : vue d'une gare

Rails et tuyaux

Concernant les rails, j'ai utilisé une librairie gratuite de Blender et repris le modèle trouvé ici : <https://free3d.com/fr/3d-model/coaster-rail-15788.html>

J'ai ensuite créé le tube autour de ces rails, les arceaux, puis créé des circuits droits, des virages et des rails d'arrivée en station où le tube s'efface pour permettre aux utilisateurs de sortir des capsules.

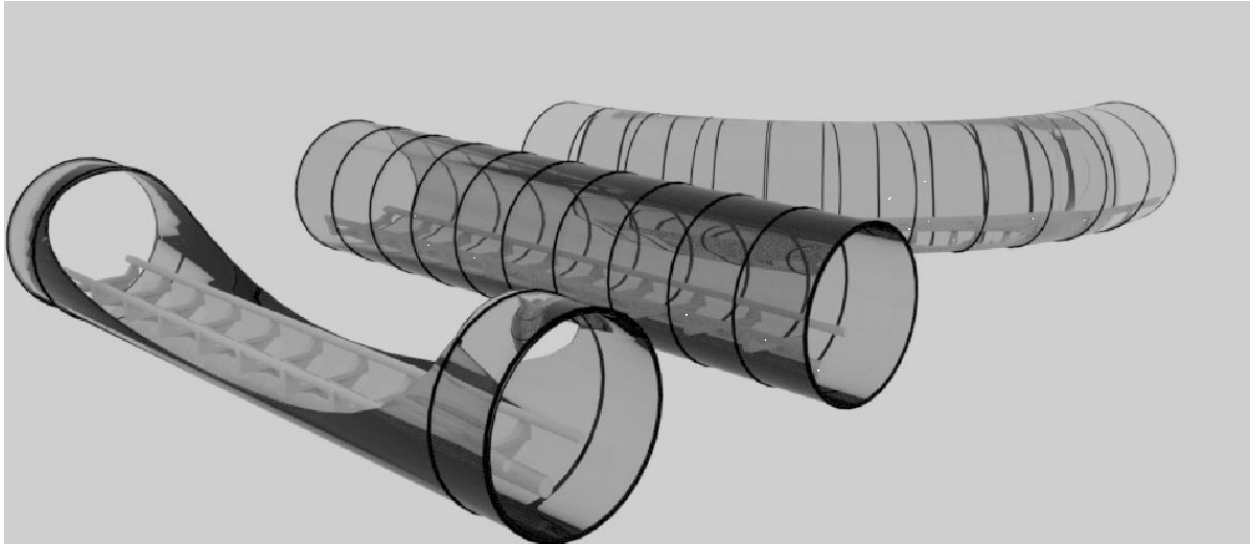


Figure 5 : vue de différents types de rails

Textures et materials

Blender et Unity permettent d'ajouter des *materials* et des *textures* aux objets afin de gérer les couleurs et les interactions de la lumière sur ces objets.

La plupart du temps l'export des *materials/textures* se fait sans soucis de Blender à Unity donc les *materials* finaux dans Unity sont ceux créés dans les fichiers Blender du dépôt Git. J'ai dû refaire un seul *material* dont l'export n'est pas bien passé : celui pour le verre des tuyaux autour des rails, appelés « Glass » dans mes fichiers Unity.

Choix des données : Grand Nancy ou Mapbox ?

Une part importante de mon travail et qui m'a demandé du temps a été la recherche des modèles 3D de la ville de Nancy, et trouver comment les utiliser.

J'ai notamment trouvé 2 jeux de données, présentant chacun des spécificités.

Données de la métropole du Grand Nancy

Tout d'abord, la métropole du Grand Nancy met à disposition librement un grand nombre de données sur son site, et notamment les bâtiments en 3D de toute l'agglomération, en fichiers .kml (fichiers lisibles par *Google Earth*) et .3ds. Les kml me permettaient de localiser les bâtiments, les 3ds de les ouvrir dans Blender. Cependant alors que l'altitude des lieux et leur position relative sont disponibles dans les kml, ils sont absents des 3ds.

Ils sont disponibles ici :

<http://opendata.grandnancy.eu/jeux-de-donnees/rechercher-un-jeu-de-donnees/>

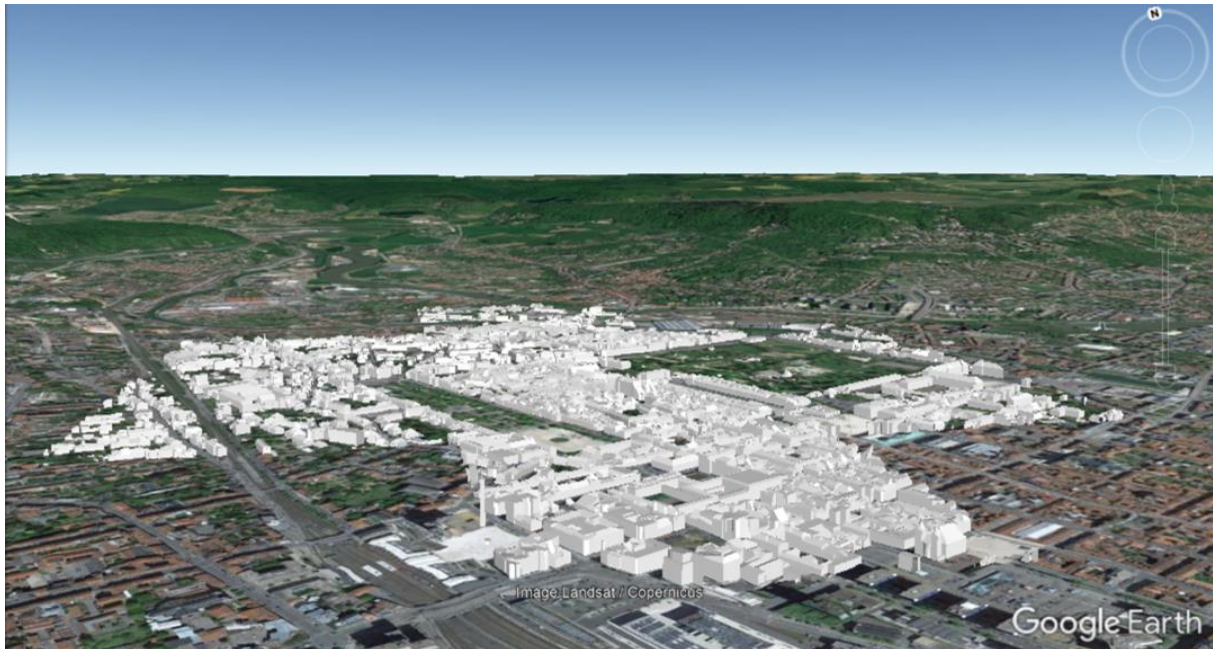


Figure 6 : vue dans Google Earth des bâtiments modélisés par la métropole du Grand Nancy

J'ai écrit en annexe quelques conseils d'utilisation, au cas où quelqu'un me lisant souhaite utiliser ces données un jour.

➤ Données cartographiques de Mapbox

Mapbox est un outil pour développeurs qui permet d'obtenir des données de localisation pour réaliser des applications, sites internet, ou des visualisations de lieux réels. Après s'être créé un compte, faire appel à leurs services demeure gratuit jusqu'à 50.000 utilisateurs par mois, et ils proposent un SDK (kit de développement) pour Unity.

L'installation fastidieuse (voir les étapes en annexe) m'a permis entre autres de superposer sur une carte de Nancy les bâtiments de la ville.

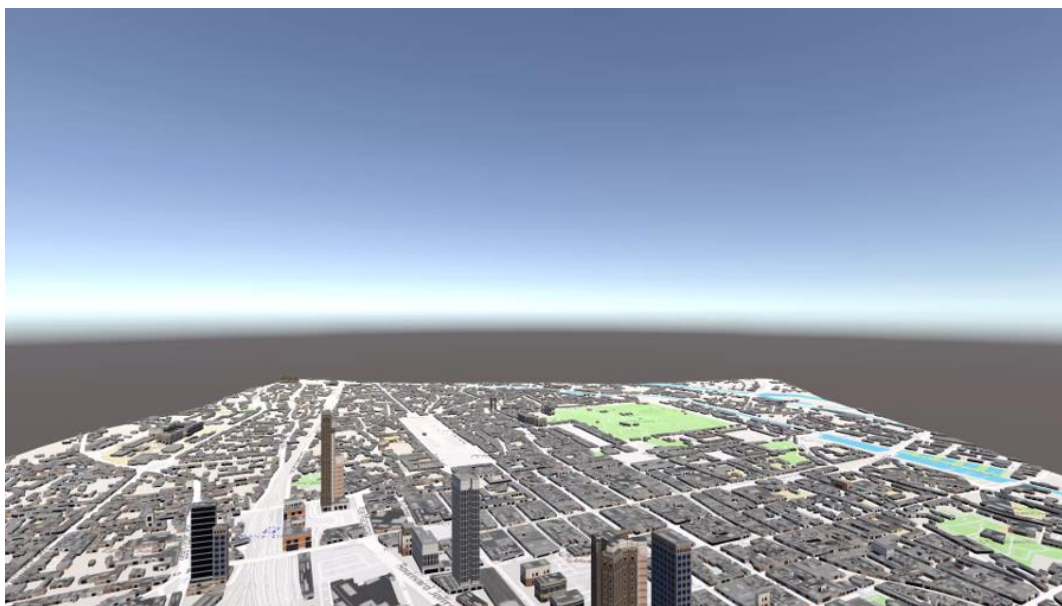


Figure 7 : vue de Nancy dans Unity avec le SDK Mapbox

La carte affichée est personnalisable sur le site de Mapbox. De plus, il est possible d'ajouter une texture qui se veut « réaliste » sur les bâtiments, comme ci-dessus.

L'avantage de Mapbox par rapport aux données du Grand Nancy est que l'implémentation dans Unity est directe, qu'on peut modifier la plupart des paramètres depuis Unity (voir annexe), et que l'altitude des lieux est directement accessible.

Néanmoins, les bâtiments sont modélisés plus finement dans le jeu de données du Grand Nancy



Figure 8 : Vue de la basilique Saint-Epvre, à gauche avec les données de la métropole, à droite avec celles de Mapbox

En fait, l'altitude des bâtiments de Mapbox est discrétisée avec relativement peu de paliers. Mais comme je m'intéressais plus à leur implémentation qu'à leur hauteur, cela m'a largement suffi.

ASSEMBLAGE DANS UNITY

Intégration des objets

► Collider et Rigidbody

Les *collider* permettent d'ajouter une enveloppe physique aux objets, qui va permettre les collisions notamment.

Le moyen le plus simple pour faire se mouvoir un objet est de lui assigner un *rigidbody* et de lui assigner dans un script un changement de position à effectuer sur ce *rigidbody*.

Le choix des bonnes géométries pour les *collider* a été assez compliqué. Après avoir essayé d'ajouter un *collider* épousant la forme des rails, et pour la capsule des sortes de pinces qui devraient glisser sur les rails, et bien d'autres possibilités, j'en suis arrivé à un modèle très simple : le rail est modélisé par un parallélépipède, et la capsule glisse par-dessus celui-ci grâce à 6 *collider* (3 à l'avant, 3 à l'arrière) qui l'entourent.

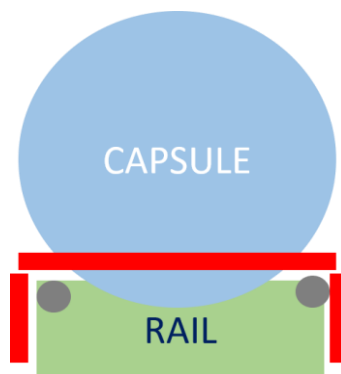


Figure 9 : schéma des collider en vue de face : en vert celui des rails, en rouge ceux de la capsule

Je ne suis pas parvenu à une solution pour faire tourner la capsule dans les virages par cette solution. Un ruban circulaire à face rectangulaire n'est pas implémentable pour les virages, car Unity n'accepte que les formes convexes pour les *collider* 'exotiques'. Une solution, en trichant un peu, serait d'écrire un script qui déplace la capsule dans les virages, en s'affranchissant des collisions.

► Assemblage des rails

Concernant l'assemblage des rails, un outil intéressant est la méthode *Object.Instantiate* de Unity. On peut multiplier les instances, qui ne seront affichées

qu'à partir du moment où on 'lance' le programme, d'un objet à partir d'un seul à partir de cette fonction. (Voir le paragraphe concernant les Scripts pour plus de détails).

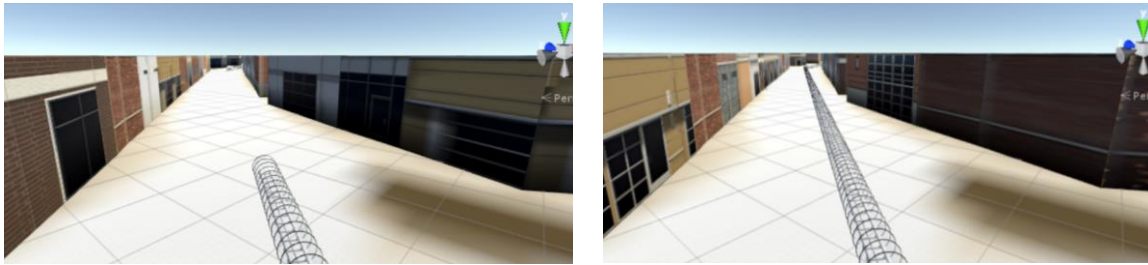


Figure 10 : à gauche, le rail simple, à droite, une fois la méthode *Instantiate* appliquée

J'ai retracé le parcours de ce projet de boucle de l'Urbanloop.

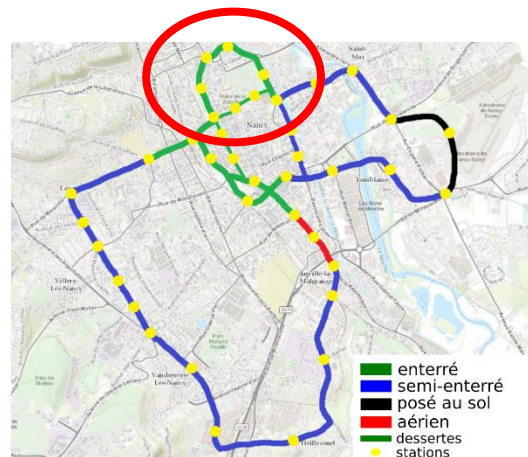


Figure 11 : projets de boucles de l'Urbanloop

Scripts présents dans le dépôt

Les scripts sont le moyen d'effectuer ou de faire effectuer une action à un objet. J'ai écrit, en langage C#, les scripts qui permettent à la capsule d'avancer, et de créer des nouveaux rails.

Lorsqu'on lie le script *CapsuleController* à une capsule, l'*Inspector* permet de fixer sa vitesse. Il faut que la capsule ait un *rigidbody* pour pouvoir se mouvoir !

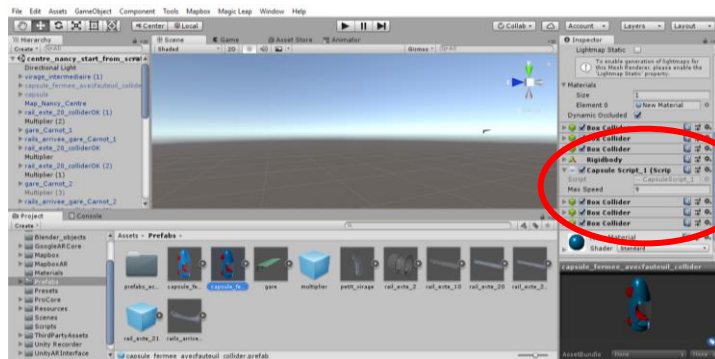


Figure 12 : inspector avec paramètres du script CapsuleController

De même, si on lie le script MultiplierController à un objet vide, l'inspector permet de choisir le rail à multiplier, la longueur de ce rail, et la direction dans laquelle il faut faire apparaître les nouveaux rails.

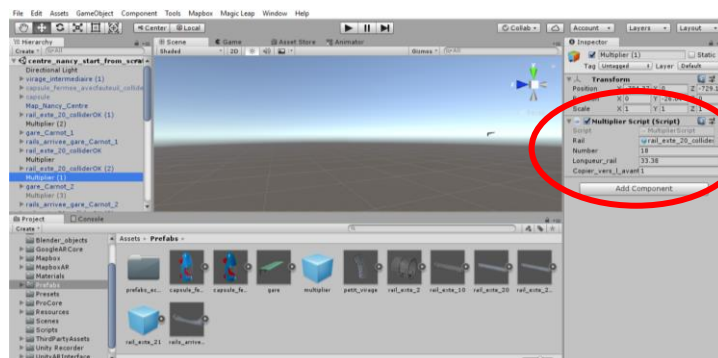


Figure 13 : inspector avec paramètres du script MultiplierController

Animations

Les animations sont réalisées grâce à des mouvements de caméra, gérés par l'Animator lié à la Main Camera de la scène.

Prise de vidéos mp4

Les captures de vidéo sont réalisées grâce au package Unity Recorder et quelques lignes présentes en annexes détaillent comment capturer une vidéo.

REMERCIEMENTS

Un grand merci à mon encadrant de projet Pascal Vaxivière pour ses conseils et sa confiance durant toute l'année,

Merci beaucoup à Alizée Evrard, Julie Heurtaux, Hippolyte Karakostanoglou et Stanislas Rambaud pour leur motivation et leur aide durant la semaine 1^o année / 2^o année,

Un immense merci à Aurélian Stab pour l'initiation à Blender, à Unity, pour ses conseils et son aide tout au long de l'année,

Merci à Philippe Mangin pour le temps qu'il m'a accordé afin de mener au mieux ce projet.

ANNEXES

Liste des annexes :

- Utiliser les données de la métropole du Grand Nancy
- Exporter en mp4 depuis Unity
- Créer un projet Unity utilisant des données Mapbox



Annexe 1 : Utiliser les données de la métropole du Grand Nancy

<http://opendata.grandnancy.eu/jeux-de-donnees/rechercher-un-jeu-de-donnees/>

J'ai trouvé le référencement de ces fichiers assez mal fait, et quelqu'un qui voudrait les utiliser doit savoir qu'ils sont découpés en dalles carrées, et présentées sous formes de bande à travers toute la zone couverte. Le premier numéro est le numéro de la bande, puis le second celui de la dalle dans la bande. Par exemple, le numéro 1765_230 et en dessous du 1764_230 et entre 1765_229 et 1765_231. Pour le centre-ville de Nancy, les dalles à utiliser seraient les :

- 1764_235, 36
- 1765_234 à 237
- 1766_235, 36



Annexe 2 : Exporter en mp4 depuis Unity

Dans l'asset store télécharger Unity Recorder

Choisir dans la barre d'outils : Window -> Recorder

Fixer paramètres de la vidéo.

Lorsqu'on lance « start recording », le programme se lance et la capture de l'écran de jeu débute.

Annexe 3 : Créer un projet Unity utilisant des données Mapbox

(Ecrit en février 2019, avec la version 2018.1.9 de Unity)

Les diverses installations sont longues, je conseille à celui qui suivrait ce pas à pas de ne pas sauter d'étapes.

Lorsqu'il faut télécharger depuis un dépôt Git, le Readme.md est à suivre scrupuleusement.

Installation des packages MAGICLEAP et MAPBOX

- <https://github.com/mapbox/mapbox-unity-sdk/tree/magic-leap-updates/sdkproject/Assets/MapboxMagicLeap>
- <https://blogs.unity3d.com/2018/03/19/take-the-next-leap-unity-and-magic-leap-partner-to-deliver-technical-preview/>

Quelques tuto et doc officielle du package Mapbox :

- <https://docs.mapbox.com/unity/maps/help/>
- <https://docs.mapbox.com/help/tutorials/create-a-map-in-unity/>

Une fois qu'on ouvre un projet avec le sdk, aller dans la barre d'outils :

Mapbox -> setup -> coller la clé Mapbox :

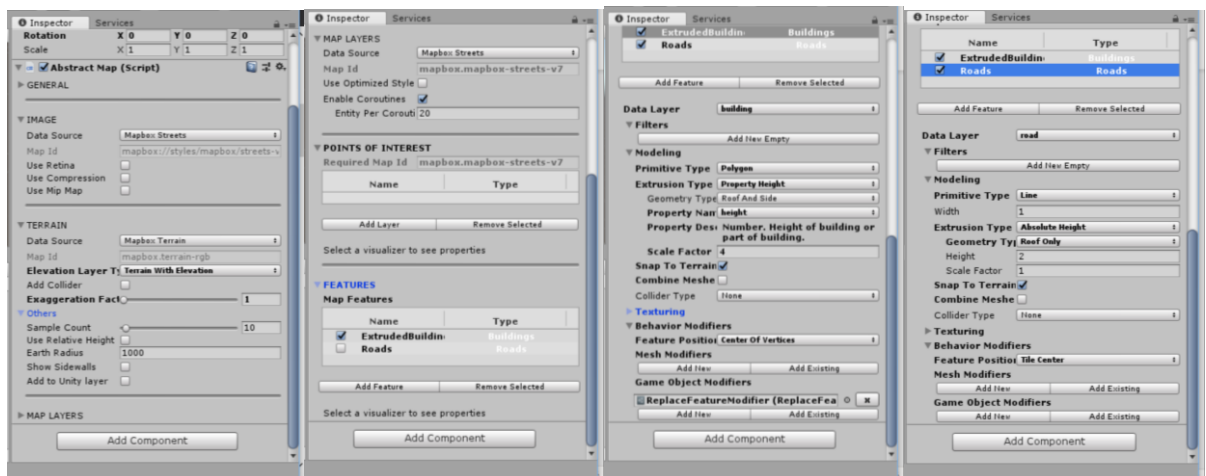
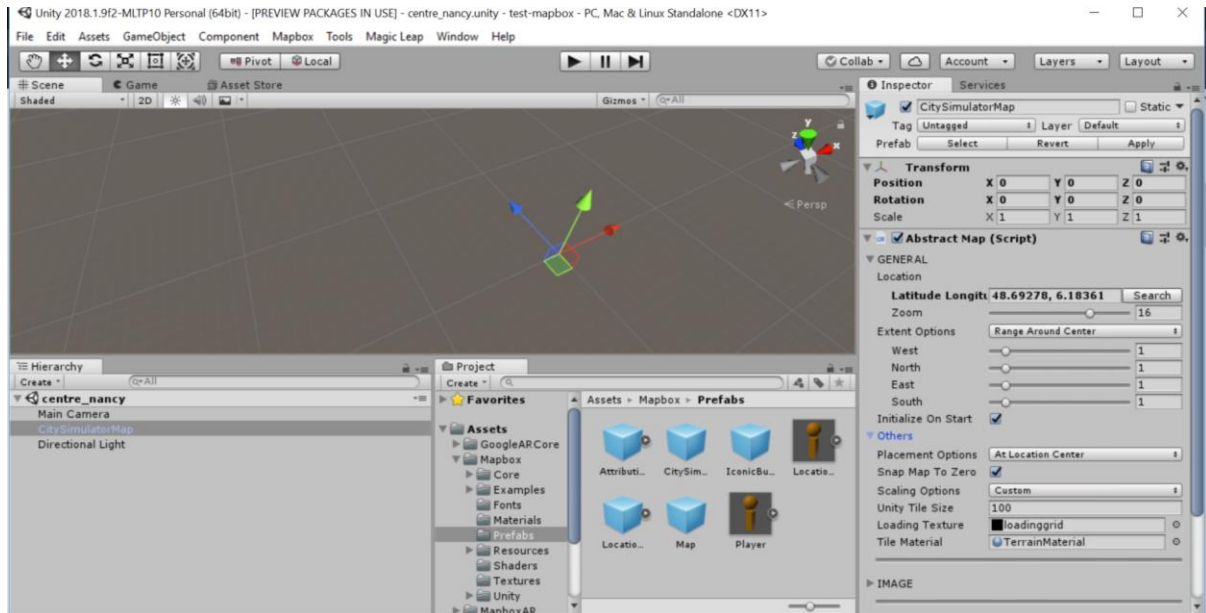
Une clé est disponible lorsqu'on crée un compte Mapbox sur leur site, et celle-ci qui est la mienne reste disponible pour le moment

pk.eyJ1IjoibGVvaGFyZlZkLjhljoiY2pyejJ4aGhyMTByYzN5bnZqa3l2d3E5eSJ9.bpDxG-EgOglyx5YyY9ZcEA

Pour créer une carte de Nancy avec les bâtiments en 3D, je vous propose la marche à suivre suivante :

- Ouvrir un nouveau projet MagicLeap avec Unity
- Activer Mapbox
- Créer une nouvelle scène

- Aller dans : Assets -> Mapbox -> Prefabs -> Map
- Ajouter un objet Map
- Dans l'inspector, lui donner les caractéristiques suivantes :



Pour ajouter des cartes personnalisées, il faut en créer sur le site de Mapbox, puis trouver la clef d'exportation vers Unity, et la coller dans la rubrique Image/MapId ci-dessus.