MAURY 2022

Thomas

Projet de modélisation 3D et carte interactive du campus du Moulin de la Housse





Mr Rabat Cyril

Mr Delisle Pierre

# Table des matières

[I. Table des matières 2](#_Toc18882)

[II. Table des figures 2](#_Toc22517)

[III. Introduction 4](#_Toc8285)

[IV. Environnement et ressources 5](#_Toc13518)

[1) Environnement 5](#_Toc6765)

[Introduction 5](#_Toc31979)

[L’interface et fonctionnement de UNITY 5](#_Toc26318)

[2) Ressources 8](#_Toc21185)

[V. Conception modélisation 11](#_Toc14594)

[1) La modélisation des bâtiments 11](#_Toc21284)

[2) L’environnement 11](#_Toc31871)

[3) La partie fonctionnel 13](#_Toc15835)

[VI. Bibliographie 15](#_Toc4979)

# Table des figures

[Figure 1 : Interface de UNITY 6](#_Toc15257)

[Figure 2 :Menu hiérarchie des objet d'une scène 6](#_Toc29731)

[Figure 3 : Menu Projet 7](#_Toc19080)

[Figure 4 : Menu inspecteur d'un objet 7](#_Toc9366)

[Figure 6 : Menu de selection d'objet,de point 8](#_Toc8750)

[Figure 5 : Menu des outils de chaque type de sélection 8](#_Toc27390)

[Figure 7 : Screen vu du ciel du campus sur Google Maps comme calque 9](#_Toc21255)

[Figure 8 : Exemple de vue 3D avec Google Maps du campus 10](#_Toc21595)

[Figure 9 : UML de la classe listeBat et Batiment 13](#_Toc4017)

# Introduction

Le Campus 3D est un projet qui a pour objectif de modéliser le campus du Moulin de la Housse et permettre au nouveaux élèves de se repérer plus facilement ainsi que de trouver leur salle de cours. Le projet se présente de la manière suivante. Une modélisation 3D de tous les bâtiments ainsi que du terrain et des routes de la végétation pour une immersion et reproduction au plus proche de la réalité. Ensuite la mise en place d’un moyen d’afficher les informations concernant chaque bâtiment pour que visiteurs virtuels puissent trouver les informations qu’ils sont venus chercher. Nous pouvons donc nous demander comment et qu’a demandé le projet pour être réalisé. Nous verrons dans un premier temps l’environnement de travail et les ressources nécessaires, puis dans un second temps quelles ont été les étapes la manière de modéliser et concevoir le projet.

# Environnement et ressources

## Environnement

### Introduction

Le projet a été modélisé sur UNITY qui est un moteur de jeu multiplate-forme. Il est l’un des logiciels de conception et modélisation de jeu vidéo le plus répandu de l’industrie. En effet il permet de créer des projets simples qu’aussi complexes et des maquettes rapide ou des projets à long terme. C’est cette forte adaptation qui lui vaut sa popularité, de plus UNITY permet de générer n’importe quel projet pour n’importe quel système exploitation ou même en intégration web. Le C# et C++ sont les langages qu’il utilise et notamment le C# qui est le plus présent qui permet grâce aux bibliothèques très complètes de Microsoft et aux classes développer spécialement pour le logiciel. Nous le verrons plus tard comment ces bibliothèques sont une vraie force du logiciel et facilitent grandement la programmation. Visual Studio est d’ailleurs l’éditeur de code conseillé pour un projet UNITY puisqu’il reconnaît tous les mot clef et classe spécifique à UNITY.

### L’interface et fonctionnement de UNITY

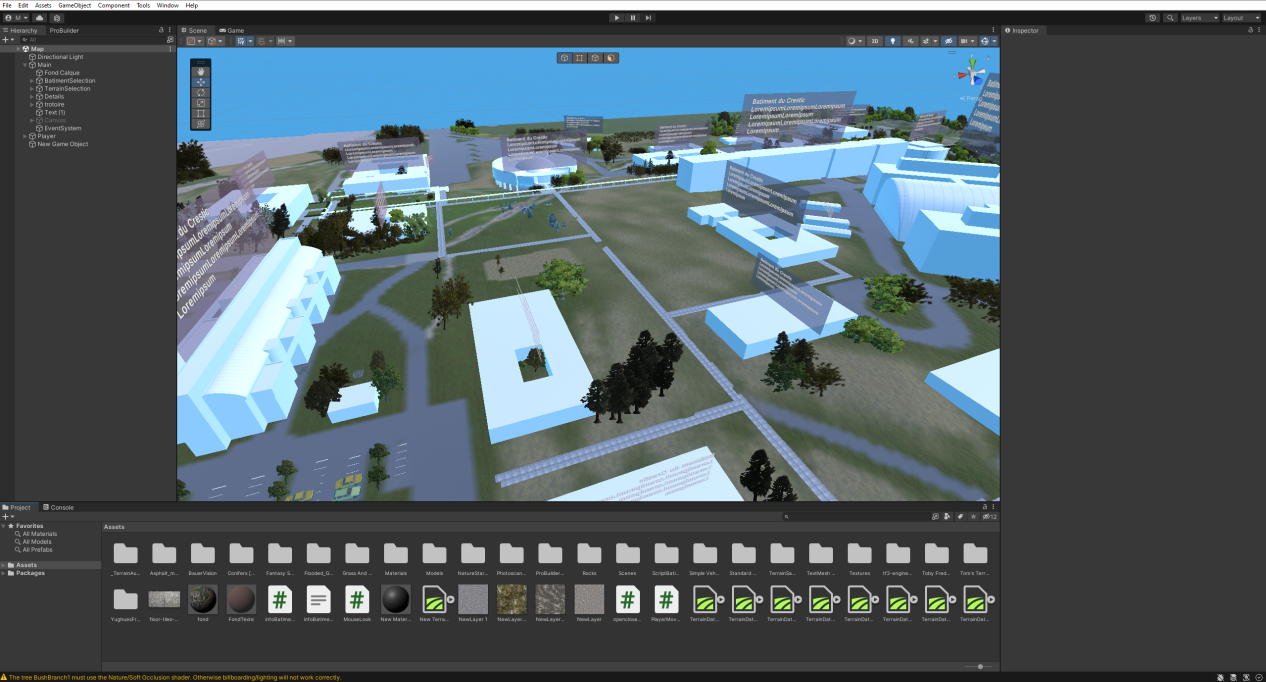


Figure 1: Interface de UNITY

UNITY permet de créer simplement des objets qui sont des listes de point dans un espace et forment des arêtes formant des faces et donc créent des objet simple (comme des cubes, les arrondis utilisent donc de multiples faces pour donner cette impression d’arrondi ). Au lieu de créer chaque objet dans le code à la main UNITY est un éditeur simple pour ces objets ( Les éditeurs comme Blender permettent la création d’objets complexes qui pourront ensuite être importés dans UNITY). Ces objets se situent dans une scène. Il peut y en avoir plusieurs pour de plus complexe projet. Chaque scène possède ses objets qui lui sont propres que l’on retrouve dans le menu à gauche sur le screen. Les objets peuvent former une arborescence, ainsi un objet peut être le parent d’un autre. Cela implique alors que les coordonnés de l’objet fils seront influencés par celui de l’objet père. Plus simplement si le fils se trouve en (0,0,0) par rapport au père il sera donc au centre du père et non au centre de la scène. Enfin si le père est déplacé dans la scène le fils gardera donc les mêmes coordonnées par rapport au père et se déplacera donc avec lui. Les autres actions qui sont la rotation et le changement de taille auront le même effet que la translation précédemment décrite. Toutes ces mécaniques peuvent être effectuées avec les outils du menu en haut à gauche sur le screen, enfin nous verrons plus tard qu’il est possible d’y accéder via des scripts.

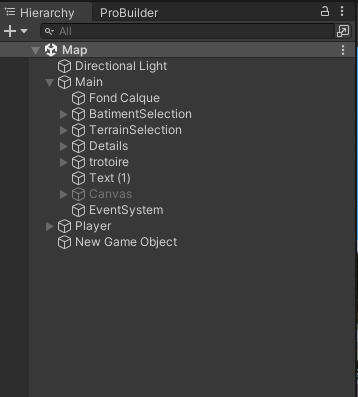


Figure 2:Menu hiérarchie des objet d'une scène

La seconde partie importante est le menu projet. Il est en réalité un affichage des fichiers propres au projets. On retrouve dedans tous les dossiers contenant les textures ou des modèles prédéfinis, les scripts, les fichiers de configurations et tout autre fichier qui est utile au fonctionnement du projet. Cette partie est commune à toutes les scènes.

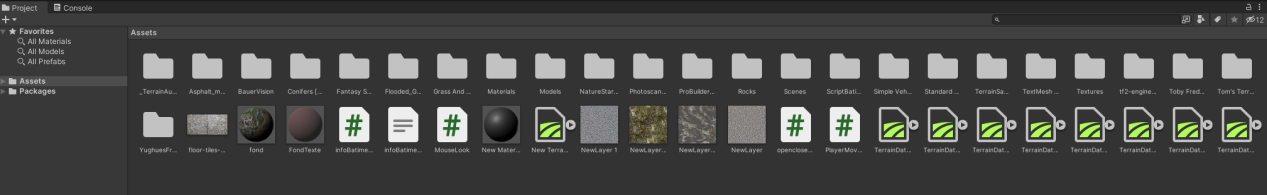


Figure 3: Menu Projet

La dernière est celle à droite qui permet elle aussi d’interagir avec l’objet sélectionné en lui donnant des valeurs précises, mais aussi agir sur des paramètres qui sont propres au type d’objet sélectionné ou de modifier les textures etc ...

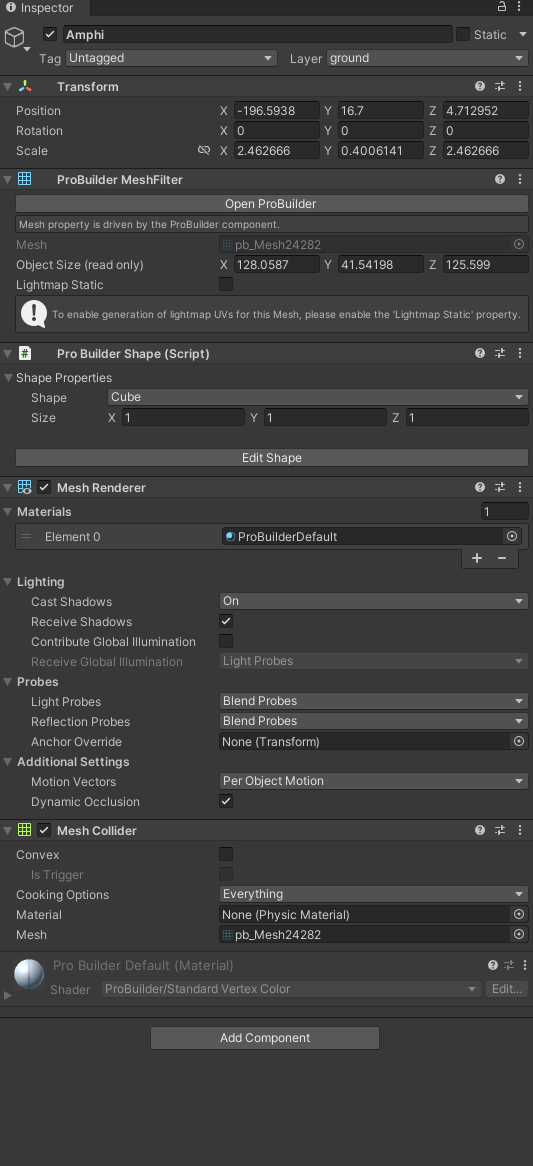


Figure 4: Menu inspecteur d'un objet

Il faut ajouter qu’une scène UNITY comprend une caméra et plusieurs et des sources de lumière. De multiples paramètres sont disponibles pour permettre de créer la scène qui correspond à nos attentes et grâce à toutes les ressources disponibles en ligne tous les styles et univers sont possibles à créer.

## Ressources

Les ressources pour UNITY et la modélisation 3D en général sont abondantes sur internet. En effet il y a de grandes chances de trouver des choses déjà existantes dans le style et la forme qui nous intéresse, et recréer de nous-même chaque pièce pour un projet quand elles existent déjà serait une perte de temps. Mais les textures ou modèle 3D ne sont pas les seules ressources utiles. Des scripts pour des fonctionnalités simples sont trouvables et des plugins vont aussi permettre d’améliorer UNITY en lui-même et permettre de rendre plus complet certaine fonctionnalité toujours dans l’optique de modéliser et créer de manière plus précise est plus réaliste ou proche de l’idée qu’on se fait du résultat.

De plus de nombreux tutoriel existent et m’ont permis de me former aux bases de UNITY. Enfin la dernière ressource qui m’a été indispensable est Google Maps et son option 3D que nous verrons plus tard.

Premièrement les plugins qui sont un type de ressources particulier. En effet ils vont apporter des modifications au logiciel en lui-même. Dans ce projet il y en a deux importants. Le premier est ProBuilder développé par UNITY même. Comme on peut le voir sur le screen, deux nouveau menu apparaissent. Le premier petit va permettre de selectionner les parties d’un objet. C’est à dire soit l’objet entier, soit un point, une arrête ou une face. Une fois une sélection faite le deuxième menu va permettre d’interagir avec la sélection. Les outils disponibles sont nombreux. Création de formes prédéfinis avec préréglages (ex : escalier réglage du nombre de marche, sa courbe) dessin pour créer une forme, extrusion, et bien d’autres qui s’adaptent à la sélection qu’elle soit faite de faces ou d’arêtes.





Figure 6: Menu de selection d'objet,de point

d'arête ou de face

Figure 5: Menu des outils de chaque type de sélection

Le deuxième plugin permet d’améliorer la fonction de terraforming des objets «terrain» de UNITY. Nous verrons la notion de terraforming et donc le fonctionnement de la fonctionnalité.

Ensuite viens les textures et modèles 3D qu’on appelle assets. UNITY met à disposition une bibliothèque réunissant tout les assets de la communauté. La force de ce magasin est qu’il est directement lié à son compte UNITY. Ainsi chaque asset du magasin ajouté à sa bibliothèque personnelle pourra être téléchargé et importer dans son projet pour pouvoir l’utiliser sans manipuler les fichiers. Grâce à ça toute devient plus simple et rapide, ainsi j’ai pu faire d’ample test de texture et modèle jusqu’à obtenir le résultat souhaité. Voici une liste non exhaustive des assets utilisés.

Enfin voici la dernière ressource que j’ai utilisée du début à la fin du projet qui est Google Maps et sa fonction 3D. Sur l’image ci-dessous on voit le campus vu de dessus. Cela m’a permis de créer un référentiel pour que le projet garde des proportions qui soient celle le plus proche de la réalité. Il a aussi agi comme un calque afin de dessiner les bâtiments à leur emplacement et donc ainsi avoir le bon espacement entre chaque bâtiment mais aussi pouvoir dessiner les routes, placer la végétation toujours au plus proche de la réalité.



Figure 7: Screen vu du ciel du campus sur Google Maps comme calque

Ensuite la fonction 3D de google maps avec un exemple comme ci dessous m’a permis d’observer chaque détail des bâtiments, mais aussi évaluer leur hauteur. J’ai aussi pu évaluer le relief du terrain et les détails présent aux alentours.



Figure 8: Exemple de vue 3D avec Google Maps du campus

# Conception modélisation

## La modélisation des bâtiments

Comme expliqué dans la précédente partie, j’ai utilisé le calque que j’ai créé avec Google Maps pour en extraire leur forme générale. Et grâce à l’outil de probuilder qui permet de dessiner des faces et extruder la face en un objet 3D. Pour certains bâtiments comme le 17, sur lequel j’ai ajouté plus de détails, j’ai pu utiliser d’autre outil plus spécifique. En revanche c’est ici que se fait ressentir la limite de UNITY pour la modélisation de chose précise. En effet comme je l’avais introduit précédemment des logiciels comme Blender sont plus à même de créer à la fois des bâtiments beaucoup plus détaillés, mais aussi de recréer leur intérieur et ceux de manière plus rapide. C’est quelque chose qui aurait été intéressant à faire dans le cadre du projet mais qui n’aura pas été possible avec UNITY. De plus apprendre un second logiciel aurait été trop long et n’était pas l’objectif du projet, en revanche cela reste une idée pour le futur et pour une amélioration potentiel de la modélisation du campus. Il est parfaitement possible d’imaginer que chaque bâtiments seraient modélisés sur un logiciel dédié puis importés dans le projet. Avec ce moyen de modélisation il serait alors possible de recréer en grande partie les salles et rendre de plus en plus immersif la visite ou peut être même créer des jeux se passant dans le campus virtuel.

## L’environnement

Comme précédemment expliqué dans les plugins, nous allons voir ce qu’est le terraforming avec les outils qui lui sont dédié ainsi que la manière de travailler qui est importante dans la façon d’aborder cette tâches pour réussir à avoir un résultat convainquant.

Il faut commencer par savoir que UNITY propose un objet qui est le «terrain». Cet objet particulier possède des paramètres qui vont permettre d’utiliser sur ce dernier des outils de terraforming. La fenêtre de paramètre du terrain possède donc cinq menu. Le premier qui est un menu de paramètre globale peu utile dans une utilisation non professionnelle.

À partir de maintenant l’ordre dans lequel nous verrons les menus aura une grande importance. En effet dans le milieu du terraforming quel que soit le jeu vidéo ou logiciel pour réussir à créer le rendu attendu il faut suivre un ordre particulier dans la manière de former et peindre le terrain. Le deuxième onglet comporte plusieurs outils dont les principaux vont permettre de modifier la forme du terrain. Les outils possèdent tous des paramètres pour personnaliser leur effet qui est la forme des brushs ou des pinceaux, leur taille et leur intensité. Pour modifier la forme du terrain, il est donc recommandé d’utiliser les outils grossier avec de gros pinceaux puis petit à petit utiliser des outils qui vont affiner la forme. Le dernier outil de l’onglet est «paint texture» qui comme son nom l’indique va permettre de peindre toujours avec la mécanique de pinceaux les textures voulues. Ainsi grâce aux assets du magasin on peut créer de multiples textures qui sont composées de heigh map (qui permette d’avoir un effet de relief sans que ce sois un objet 3D). De plus en jouant sur l’intensité d’un pinceau, plusieurs textures peuvent être superposé. Ainsi par exemple pour créer un chemin de terre, si on additionne une texture de terre et de roche en réglant sa heigh map, on aura un effet terre avec de la roche dedans.

Viens ensuite le troisième et quatrième onglet qu’on va appeler tout simplement peindre la végétation. Comme son nom l’indique, les outils vont tout simplement nous permettre d’ajouter de la végétation qui est un détail très important pour que cela rende bien à l’œil. La mécanique de pinceaux est la même encore une fois. Cette fois-ci grâce au magasin on va utiliser des modèles 3D d’arbre, de buisson et d’herbe. L’intensité du pinceaux cette fois aura pour effet d’espacer les modèles ou non entre eux

## La partie fonctionnel

Maintenant que toute la modélisation a été faite, il ne reste plus que la partie fonctionnelle qui peut se diviser en deux parties. La première sur la possibilité de se déplacer dans l’espace et la seconde sur l’affichage des informations concernant les bâtiments.

Comme expliqué dans le fonctionnement de UNITY, une scène est composée d’une ou plusieurs caméras et sources de lumière. Ainsi pour créer un personnage qui puisse nous permettre de nous déplacer dans toute la scène, on va vouloir déplacer la caméra lorsqu’on appuie sur les touches. Pour cela on va utiliser les scripts que met à disposition UNITY.

Il y a deux parties importantes. La première faire bouger la caméra en translation dans la scène. C’est le code qu’on retrouve dans le script PlayerMovement.cs présent dans le projet. Pour faire simple à chaque frame d’update de l’application le code est fait dans lequel on déplace en translation selon l’entrée du clavier le personnage et ce en fonction d’un paramètre important appelé «delta time» qui est simplement ce qui va permettre de se déplacer à une vitesse de manière constante. Sans l’utilisation de ce paramètre les déplacements seraient alors liés au nombre de FPS auquel l’application tourne. La deuxième partie est évidemment de pouvoir déplacer la caméra avec les souris pour regarder autour de soi. Le code se trouve dans MouseLook.cs et utilise le même principe que pour le déplacement du joueur, en revanche ce ne sont plus des translation mais rotation autour de la caméra qui sont effectuées.

Ensuite viens peut-être le plus important qui est l’affichage des informations concernant les bâtiments. Pour pouvoir permettre la modification des informations à n’importe quel moment, j’ai choisi d’utiliser un fichier json. Pour exploiter le fichier j’ai donc créé deux classes qui vont permettre de le lire. La classe listeBat et Batiment.

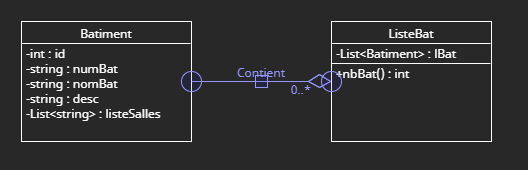


Figure 9: UML de la classe listeBat et Batiment

Ces classes me permettent de récupérer le Json dans un objet qui est ensuite simple à manipuler. Quelque chose dont je n’ai pas encore parlé est la manière dont UNITY permet de lier des objet d’une scène à un script. Il en existe de multiple mais ici dans le fonctionnement de l’affichage des informations il est important de voir que le champ id dans le Json correspond à un tag associé à un objet de la scène. Et grâce à la fonction «FindObjectWithTag» on possède maintenant un lien à l’objet possédant le tag qui correspond à celui donné par l’id. Il est donc important de ne pas le modifier au risque de voir le texte d’un bâtiment inversé avec celui d’un autre.

# Bibliographie

1. Campus du Moulin de la Housse sur Google Maps:

<https://www.google.com/maps/@49.2415161,4.0636297,170a,35y,355.26h,52.65t/data=!3m1!1e3?hl=fr>

1. Le magasin d’asset de UNITY:

<https://assetstore.unity.com/?aid=1011l98LI&on_sale=true&orderBy=1&utm_campaign=unity_affiliate&utm_medium=affiliate&utm_source=partnerize-linkmaker>

-La skybox du projet: <https://assetstore.unity.com/packages/2d/textures-materials/sky/fantasy-skybox-free-18353>

-Modèle de taule pour le toit du chemin : <https://assetstore.unity.com/packages/3d/props/exterior/metal-plates-64240>

-Modèle de voiture:

<https://assetstore.unity.com/packages/3d/vehicles/land/simple-cars-pack-97669>

-Les rocher de la Théorie du Chaos:

<https://assetstore.unity.com/packages/3d/props/exterior/rocks-604>

-Groupe d’asset pour la végétation:

<https://assetstore.unity.com/packages/3d/vegetation/trees/hdrp-pine-tree-214095>

<https://assetstore.unity.com/packages/3d/environments/nature-starter-kit-2-52977>

<https://assetstore.unity.com/packages/3d/vegetation/trees/dream-forest-tree-105297>

<https://assetstore.unity.com/packages/3d/vegetation/trees/dream-forest-tree-105297>

<https://assetstore.unity.com/packages/3d/vegetation/plants/yughues-free-bushes-13168>

<https://assetstore.unity.com/packages/2d/textures-materials/nature/grass-and-flowers-pack-1-17100>

<https://assetstore.unity.com/packages/3d/vegetation/trees/conifers-botd-142076>

-Terrain tools:

<https://assetstore.unity.com/packages/tools/terrain/terrain-tools-64852>

-Flooded Grounds (Multiple modèle et texture utile):

<https://assetstore.unity.com/packages/3d/environments/flooded-grounds-48529>

-Texture de sol:

<https://assetstore.unity.com/packages/2d/textures-materials/floors/outdoor-ground-textures-12555>

-Texture des routes:

<https://assetstore.unity.com/packages/2d/textures-materials/roads/asphalt-materials-141036>

-Complément de pinceau de terraforming :

<https://assetstore.unity.com/packages/3d/environments/landscapes/terrain-sample-asset-pack-145808>

1. Tutoriel sur le terraforming:

<https://www.youtube.com/watch?v=ddy12WHqt-M>

1. Tutoriel sur probuilder:

<https://www.youtube.com/watch?v=Re6wU7zPlXI&list=PLUWxWDlz8PYJGinZ30v7ZEW-FPP85qIRu>

1. Tutoriel sur les basique de C#:

<https://www.youtube.com/channel/UCYbK_tjZ2OrIZFBvU6CCMiA>

1. Le GitHub du projet:

<https://github.com/Dragonheads51/campus3D>