



Présentation de Unity 3D :

Unity est un moteur 3D de création de jeux/d'applications, qui intègre le PhysX de Nvidia. Il n'est pas un modéliseur, mais fonctionne avec une scène 3D gérant lumières, shaders, caméras. Il permet l'insertion de code dans différents langages, allant du javascript au C-sharp, séparément ou même simultanément. Des plug-ins d'aide à la création sont intégrés au soft, tel un générateur de terrain, de végétation, ou encore des packs préfaits facilitant le contrôle du joueur, des caméras etc... Unity utilise un système de « drag and drop » dans son interface, ainsi qu'un système d'Assets, sortes de « clips » pour ceux qui ont déjà utilisés Flash. Intuitif et puissant donc, il présente l'énorme avantage d'autoriser une publication multi-plateformes (Playstation3, Xbox, Wii, PC/MAC, WebPlayer, iOS, Android).

La version de base du logiciel, gratuite, permet de se « faire la main » et de pouvoir malgré tout, mener des projets plutôt complexes ! Par rapport à la version dite « pro » seuls vont manquer des options de rendu, d'optimisation en règle général. Nous utiliserons pour ce cours cette version gratuite.

Le meilleur moyen de cerner les capacités d'un moteur 3D/2D restant l'exemple :
iOS : ShadowGun, Battleheart (2D), Guerilla Bob, Samurai II
PC : MonkeyQuest (mmo), Tron Legacy LightCycle, Max and the Magic Marker
Wii : Max and the Magic Marker

La communauté, notamment US, est hyperactive et recense sur le site officiel un très grand nombre de tutoriaux, conseils, partage de ressources et dispose même d'un store en ligne proposant des assets pouvant être très utiles (interaction avec facebook par exemple) ça se passe ici :

www.unity3d.com

La communauté française est beaucoup moins active mais grâce à la contribution énorme de Krys, propose néanmoins énormément de tutoriaux et est mis à jour très régulièrement.

www.unity3d-france.com

Des sites communautaires recensent bon nombre de jeux réalisés par des amateurs, parfois des pros :

www.woogle.com

www.blurst.com

www.musegames.com

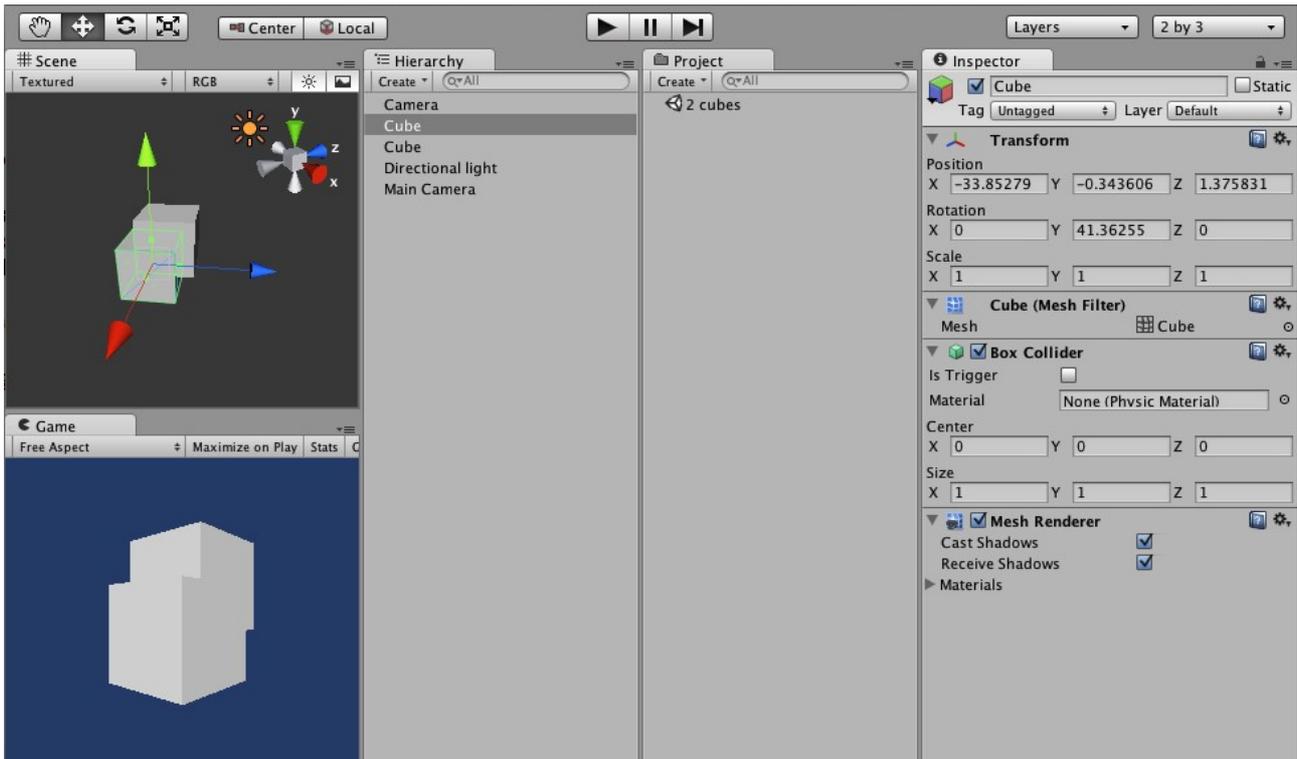


Nous allons faire un tour de l'interface de Unity, comprendre comment s'organisent les fenêtres, cerner les principes fondamentaux de fonctionnement, puis nous lancer dans un petit exercice pratique.

1) Les fondamentaux

On crée un nouveau projet :

Nous allons organiser l'interface par Windows>Layouts> 2by3



En haut au centre de notre 2by3, trois boutons, play, pause, et image par image permettent de tester en temps réel le projet. Quand ce dernier est en mode play, les modifications, éditions, ne seront pas enregistrées.

Détaillons d'abord succinctement chaque partie :

De gauche à droite :

-La scène : C'est l'éditeur du projet, représenté par une scène 3D dans laquelle on peut naviguer, disposer ses objets.

-Le game Player : Lié à la scène, c'est en fait la vue de la main Camera, les interactions sont possibles, une vraie prévisualisation du projet final sans passer par un « build ».

-Hierarchy : Une liste des objets présents sur la scène actuelle, permet aussi de créer des éléments via un menu contextuel.

-Project : Le dossier qui contient toutes les ressources du projet, celles qui sont importées apparaissent ici.

-Inspector : Le cœur de Unity : tout est géré ici, du scripting aux paramètres au sens large de tous les éléments du projet.

Nous allons voir maintenant plus en détail les parties qui composent la zone de travail, en commençant par la Scene.

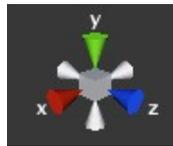
2) La Scene

De base, une main camera est présente sur la scène.

Créons un cube pour mieux visualiser les fonctionnalités du player : menu gameObject > create other > cube

a) Comment naviguer dans la scène 3D ?

- clic gauche : sélectionne un objet, fait apparaître par défaut les axes de position, le clic de base de manière générale pour l'édition.
- clic droit : en le maintenant enfoncé et en bougeant la souris, on bouge une caméra centrée sur le monde
- alt+clic gauche : en le maintenant enfoncé et en bougeant la souris, on bouge une caméra centrée sur l'objet sélectionné
- clic milieu : « glisse » la vue, raccourci en fait de l'outil « main »
- appui sur barre espace : passe la zone sélectionnée (player ou scene par exemple) en mode plein écran
- la touche « f » permet de faire un focus sur l'objet préalablement sélectionné (nous le verrons plus tard, équivaut également au double clic dans la hierarchy sur objet)
- molette centrale : zoom, dézoom
- les axes x,y,z permettent de passer sur la vue normée correspondante en cliquant dessus, un clic sur le cube passe la vue en perspective.



- de gauche à droite : déplacement vue, déplacement objet, rotation et scale > tous les paramètres de rotation/position/scale sont réglables dans l'inspecteur



b) Les éléments d'onglet de la Scene

De gauche à droite :

- le type d'affichage : mode filaire, texturé, filaire+textures etc...
- modes : RGB (classique), les autres modes permettent de cibler une fonction précise (mipmaps, alpha ou overlays)
- active/désactive : les lumières présentes sur la scène, la grille et éléments GUI (interface), les sons.
- Iconification de la scène : active/désactive les icônes présentes sur la scène, permet également d'en choisir la taille. Très pratique pour visualiser rapidement des éléments précis dans les scènes complexes.

En cliquant sur l'objet main camera sur la scène (ou dans la hierarchy), une petite fenêtre apparaît, affichant ce que visualise la caméra. C'est la vue que nous avons dans la prochaine zone que nous allons découvrir, le player.

3) Le player

Véritable test en temps réel du projet, sans passer par un « build » (qui peut prendre un certain temps suivant la l'ampleur du projet). Nous pouvons tester ici l'interactivité qui sera celle de la version finale du projet. Le player se lance à l'appui de la touche Lecture, en haut au milieu de l'interface de Unity.

a) L'aspect du player : contient les formats les plus répandus ainsi que les formats spécifiques à la plateforme sur laquelle nous développons (pour un projet iOS par exemple, on aura les formats de l'iphone 3gs, 4, de l'ipad etc...)

b) Les autres éléments d'onglet

De gauche à droite :

- active/désactive le mode plein écran du player
- la pop-up « stats » est un compte-rendu détaillé technique du projet, elle affiche par exemple le nombre d'images par secondes, le nombre de faces calculées dans la scène, ou encore l'utilisation en mémoire vidéo. Cette pop-up est très importante dans la compréhension de l'optimisation d'un projet (notamment pour iOS).
- Iconification du player, fonctionnant exactement comme dans la Scene que nous avons vue juste avant

4) La Hierarchy

C'est la liste des éléments présents sur la scène. Les éléments peuvent y-être renommés, dupliqués, copiés etc...Un double clic sur un élément de la liste, par exemple sur notre cube, fait un focus sur l'objet en question, très utile dans les scènes complexes. En haut à gauche de la zone Hierarchy on trouve un menu déroulant « Create », qui est en fait un raccourci du menu GameObject > Create Other. Détaillons le pour voir les possibilités de création qu'il nous offre :

- Particle systems : générateur d'effets, allant de la pluie à l'explosion, grâce aux particules, modifiables par la physique
- Camera : crée une caméra, de base en projection perspective, qui peut-être aussi orthographique, pour les projets 2D par exemple
- >> Les éléments GUI sont indépendants de la scène 3D et non-affectés par la caméra.
- GuiTEXT : couche 2D de texte, systématiquement « au dessus » des éléments 3D de la scène, utilisé généralement pour les interfaces
- GuiTexture : couche image 2D, qui comme le GuiTEXT, sera au dessus de la couche 3D
- 3DText : le texte est sous la forme d'une mesh 3D, il est donc affecté par la caméra
- >>Les éléments lumières peuvent afficher leur rayonnement via l'inspecteur
- DirectionnalLight : lumière diffuse type soleil, réglable notamment par la rotation de celle-ci
- PointLight : lumière linéaire
- SpotLight : lumière type spot classique, diffuse sur une zone restreinte
- Primitives : création de primitives de bases incluant, cube, sphere, capsule, cylindre et plane
- Clothes : Module de création de corps « mous », utilisé notamment pour les vêtements, interagit avec la physique
- AudioReverbZone : Zone de réverbération du son
- Ragdoll : Module de création de pantin articulés intégrant un système d'ossature complet, interagit avec la physique
- Tree : Module de création d'arbres réalistes, interagit avec la physique
- WindZone : Zone physique de vent

5) Le Project

Il liste toutes les ressources du projet, des sons, textures aux meshes 3D, tout y est. Les ressources qui sont importées, de Unity ou de l'extérieur apparaissent ici. Cette zone se gère comme un véritable bureau, autorisant la création de dossiers, le classement des ressources, etc... C'est également là qu'on retrouve les différentes scènes du projet, reconnaissables par le logo Unity de leurs icônes.

Un clic droit dans cette zone permet d'afficher un menu déroulant, raccourci du menu Assets.

Le projet est une véritable fenêtre ouverte sur votre ordinateur : il est tout simplement le dossier physique de votre projet. Par conséquent, effacer un fichier depuis Project l'effacera également physiquement de votre ordinateur. Unity met ainsi à jour le projet en temps réel, au fur et mesure qu'on glisse dans le dossier projet des nouvelles ressources.

Tout comme la Hierarchy, le Project dispose d'un menu create : Détaillons le également.

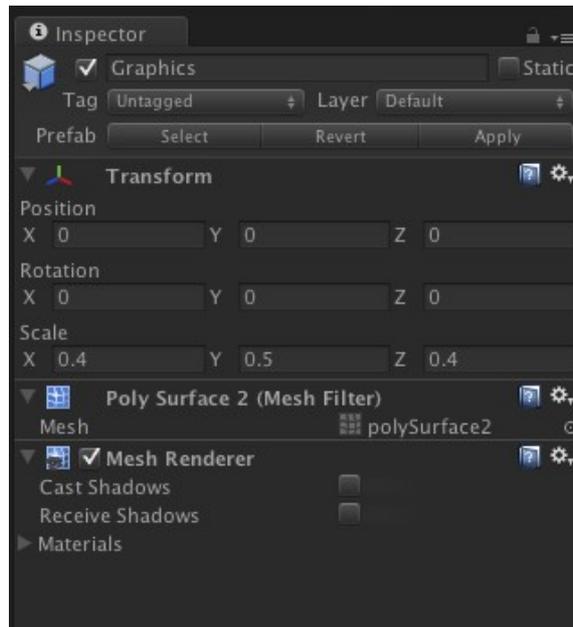
- Folder : création d'un dossier vide
- Code : crée un fichier script qui peut-être du javascript , Boo, C# ou encore un script permettant la création de shaders
- Prefab : un objet qui stocke différents GameObjects avec sa propre hierarchie, pour les utilisateurs de flash, purement un MovieClip
- Material : paramètres concernant la texture appliquée sur un élément 3D
- Les autres possibilités ne rentrent pas dans une formation initiation

A la création d'un nouveau projet, Unity demande si l'on veut importer ou non des packages pré-faits qui vont faciliter le développement du projet. Par exemple un controller FPS (que nous utiliserons dans l'exercice de cette formation), ou encore un script de caméra. Là encore et au même titre qu'une ressource importée, ces packages apparaissent dans la zone Project, ce sont les assets. Ne pas hésiter dans Project à agencer à sa guise les prefabs, textures etc...pour se retrouver facilement.

Avant de nous lancer dans la pratique, regardons en détail la dernière zone de l'interface de Unity, l'Inspector.

6) L'Inspector

Le panneau Inspector est le point névralgique de Unity, le principal outil pour ajuster chaque élément des objets de jeu et des ressources de votre projet. Tout comme l'Inspecteur de propriétés dans Flash, le contenu de ce panneau est sensible au contexte. Autrement dit, il affiche les propriétés correspondant à l'objet sélectionné. Dans le cas de notre Cube, nous voyons s'afficher toutes les informations relatives à sa position, sa taille, sa rotation etc...



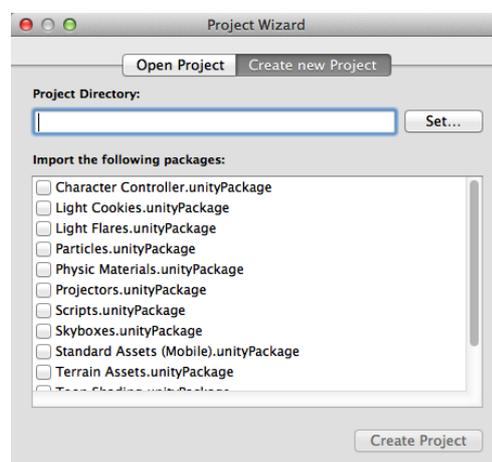
Le panneau Inspector affiche tous les composants de l'élément sélectionné ce qui permet d'en ajuster les variables à l'aide d'éléments simples comme des zones de saisie de texte, des curseurs, des boutons et des menus déroulants. Beaucoup de ces variables utilisent également le système de glisser-déposer de Unity, ce qui signifie qu'avec cette méthode vous pouvez définir des paramètres plutôt que de sélectionner une valeur dans un menu déroulant, si cela est plus pratique.

Ce panneau n'a pas uniquement pour rôle d'afficher les éléments qui composent les objets. Il peut également afficher les préférences de votre projet selon l'option que vous sélectionnez dans le menu Edit (Édition). Les propriétés des composants s'affichent dès que vous sélectionnez de nouveau un objet ou une ressource.

Après ce tour global des zones (panneaux) de l'interface de Unity, nous allons mettre la main à la pâte en créant un mini-projet qui consistera à pouvoir diriger un personnage en vue 1ère personne sur une île entourée d'eau modelisée par vos soins.

7) L'exercice

Créons un nouveau projet par File > New Project... : une pop-up apparaît proposant de cocher ou non des packages qui seront importés ensuite. Nous allons en cocher quatre pour notre exercice.



Dans les éléments qu'il nous faut : - le module **terrain** pour modeler l'île

- le module **water (basic)** pour gérer l'eau autour
- **skyboxes** qui va créer l'environnement ciel
- et enfin **character controller** qui embarque (entre autres) une caméra 3ème personne et une maniabilité simple

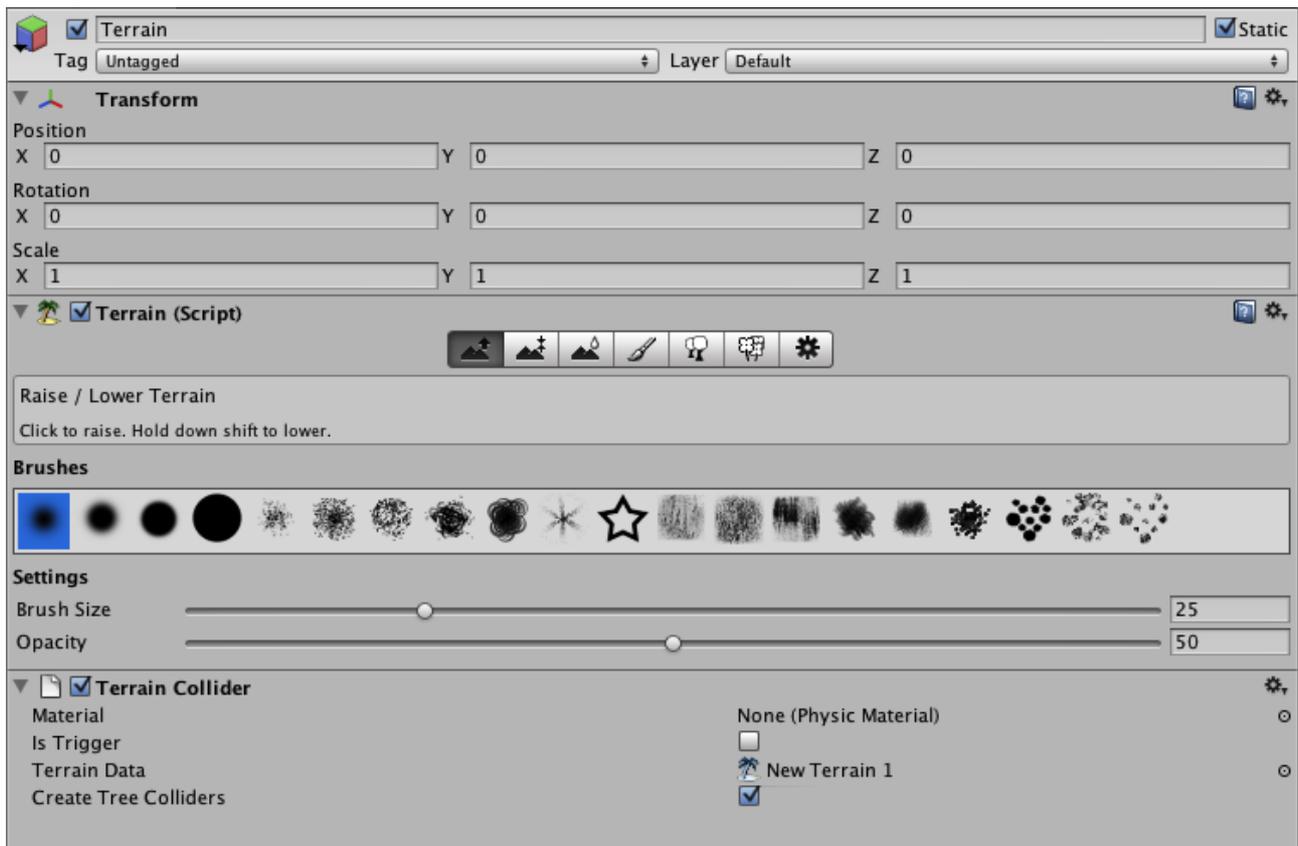
On clique sur Create Project toujours dans la pop-up pour valider. Un chargement indique que les packages sont importés dans le projet. Sachez qu'à tout moment ces packages peuvent être réimportés en interne via Assets > Import Package.

a)Le terrain

Nous venons d'importer le module terrain à la création du projet, nous allons retrouver le menu du même nom dans ceux de Unity. On clique sur Terrain > create terrain. Un objet plane apparaît : il faut réajuster la caméra de la scène pour mieux le visualiser en cliquant sur le cube (vue perspective) des axes de la scène.

Pour éditer ce terrain, on clique sur son nom dans la hierarchy ou directement sur l'objet plane sur la scène. Pour centrer la vue sur lui, double clic sur « terrain » dans la hierarchy.

Nous retrouvons dans l'inspector les paramètres liés au terrain :

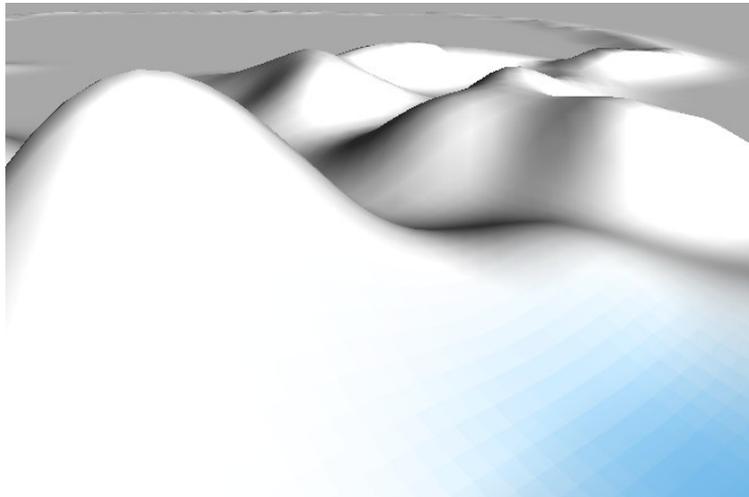


Le composant Terrain comprend sept sections, qui sont facilement accessibles à l'aide des boutons situés en haut du composant :



Voici un rapide aperçu de leurs capacités respectives.

-L'outil *Raise/Lower Terrain* (élever/abaisser le terrain) Il permet d'élever certaines zones sur lesquelles vous peignez à l'aide de l'outil Transform



Vous pouvez également définir le style de brosse, la taille et l'opacité (efficacité) de la déformation que vous effectuez. Pour obtenir l'effet contraire, à savoir un abaissement de la hauteur, appuyez en même temps sur la touche Maj.

-L'outil *Paint Height* (peindre jusqu'à hauteur) Il fonctionne de manière analogue à l'outil Raise/Lower Terrain mais il possède un paramètre supplémentaire : Height.



Cela signifie que vous pouvez définir la hauteur que vous allez peindre. Lorsque la zone de terrain dont vous modifiez l'élévation atteint la hauteur spécifiée, elle s'aplatit, vous permettant de créer des zones planes.

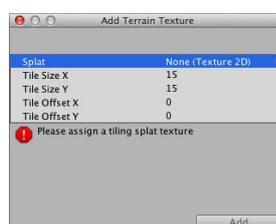
-L'outil *Smooth Height* (adoucir les dénivelés) Il sert principalement en complément d'autres outils, comme Paint Height, pour adoucir les dénivelés abrupts de la topographie.



-L'outil *Paint Texture* (peindre des textures) Il s'utilise pour peindre les textures sur la surface du terrain (nommées *Splats*) dans Unity.



Avant de peindre avec des textures, vous devez d'abord les ajouter à la palette dans la zone Textures de cet outil. Pour cela, cliquez sur le bouton Edit Textures :



Sélectionnez Add Textures puis, dans le menu déroulant Splat, choisissez un des fichiers de texture actuellement dans votre projet et définissez les paramètres de taille de la texture choisie.

Nous pouvons ajouter plusieurs textures dans cette même zone. La première sera par exemple peinte sur tout le terrain par défaut. En combinant ensuite plusieurs textures de différentes opacités et en peignant à la main sur le terrain, vous pouvez donner un aspect réaliste à tous les types de surface.

Pour choisir la texture avec laquelle peindre, cliquez simplement sur son aperçu dans la palette : un contour bleu l'encadre alors.

-L'outil *Place Trees* (disposer des arbres) Comme son nom l'indique, cet outil permet de placer, à la souris (en peignant ou par de

simples clics), des arbres sur le terrain, après que les ressources à utiliser ont été définies.



Comme pour les textures de l'outil Paint Texture, cet outil dispose d'un bouton Edit Trees pour ajouter, modifier et supprimer des ressources de la palette.

-L'outil *Paint Details* (peindre des détails) Il fonctionne d'une manière analogue à l'outil Place Trees, mais il est conçu pour travailler

avec des objets de détails comme les fleurs, les plantes, les rochers et le feuillage de petite taille.



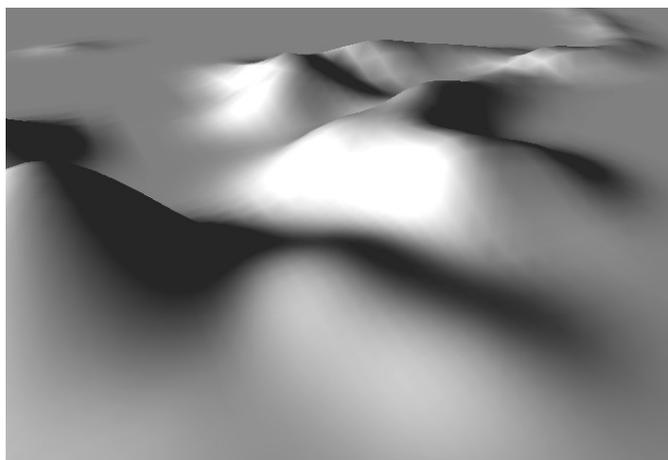
-Terrain Settings (les paramètres de terrain) La section Terrain Settings du composant Terrain (Script) contient différents para-

mètres pour le dessin du terrain par le GPU (processeur graphique) de l'ordinateur.



Ces trois dernières parties sont peu détaillées car nous n'allons pas les utiliser dans notre exercice.

Pour nous aider à modeler le terrain et surtout pouvoir mieux visualiser les reliefs nous pouvons créer une lumière, la plus répandue étant la directionnal light, nous allons dans GameObject > create other > Directionnal Light. On peut ajuster sa direction par la rotation d'un de ses axes, la directionnal étant une lumière globale, sa position dans l'espace n'aura aucun effet sur l'éclairage.

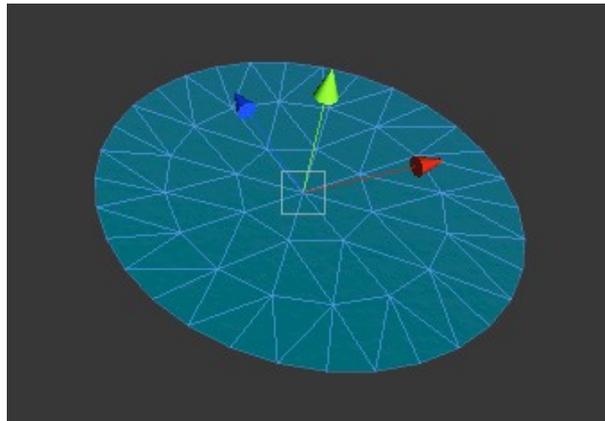


Notre lumière étant maintenant posée et le terrain texturé, nous pouvons créer notre zone d'eau.

b) L'Eau

Là encore donc, l'élément eau est un package préfabriqué, nous allons le retrouver dans la zone Project : dossier Standard Assets > Water (Basic) > Daylight Simple Water (nous avons le choix dans cette version basique de ce package « eau », entre une version jour ou nuit de l'élément).

On glisse ce Daylight Simple Water du Project dans la Hierarchy. Un double clic sur ce GameObject dans la hierarchy nous fait le focus sur la scène d'un disque bleu :



De base, il est assez petit (surtout par rapport au terrain), nous allons donc le scaler grâce à l'outil :



Nous entourons l'îlot par la mer, en réajustant la hauteur, la taille si besoin.



Jusqu'à une certaine distance, les effets liés à l'eau sont quasiment invisibles, il nous faut zoomer de manière importante dans la scène pour apercevoir que l'eau est en fait animée. Nous allons pouvoir agir sur la taille et la vitesse des vagues via l'inspecteur. Notez que l'animation est réalisée en textures, ce n'est pas directement une déformation de la mesh (soucis d'optimisation). Tout est donc modifiable, de la teinte de l'eau, jusqu'à sa couleur de réflexion.

Il nous manque une zone de ciel pour être un peu plus immergé dans notre scène.

c) Le ciel

La manipulation est un peu différente pour l'élément ciel. Nous allons aller chercher les paramètres dans le menu Edit > Render Settings . Dans l'inspecteur à droite de notre interface, nous allons chercher un pack de texture en cliquant sur Skybox Material :



On prend par exemple, la Skybox Sunny 3. Le ciel est appelé skybox car c'est en fait une boîte, composée de six faces, parfaitement liées entre elles : nous pouvons observer cela en cliquant sur Standard Assets > Skyboxes > Sunny3 Skybox. L'inspecteur affiche les six textures séparément.

Nous passons volontairement sur la création personnalisée de skybox, trop complexe et peu utile pour notre tutorial !

Voilà ce à quoi pourrait ressembler notre décor au final :

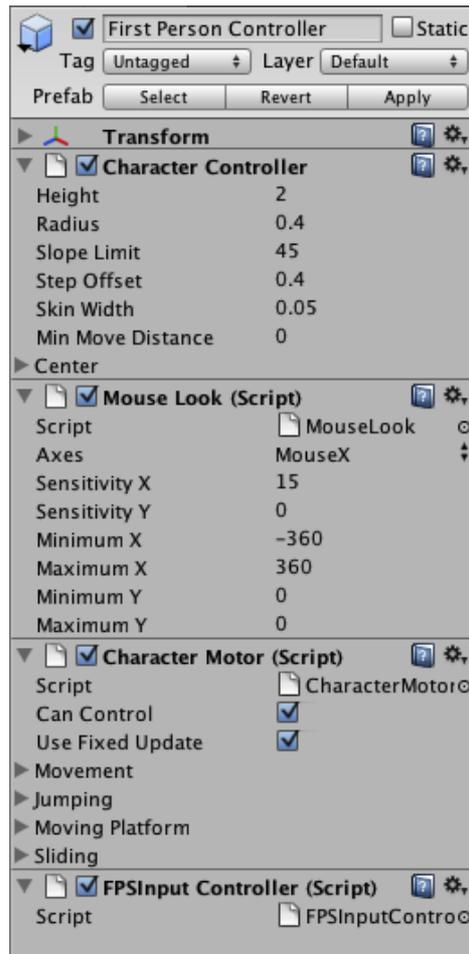


Posons maintenant notre personnage, qui sera sous la forme d'une caméra mobile, une vue à la 1ère personne.

d) Le personnage

Ici nous allons utiliser l'asset Character Controller que nous avons importé à la création du projet. Nous allons le récupérer dans le Project > Standard Assets > Character Controller > First Person Controller. Faisons glisser ce GameObject du Project jusqu'à la Hierarchy.

Il se présente sur la scène, sous la forme d'une caméra. Dans l'inspector nous notons qu'il s'agit d'une caméra scriptée avec quatre paramètres :



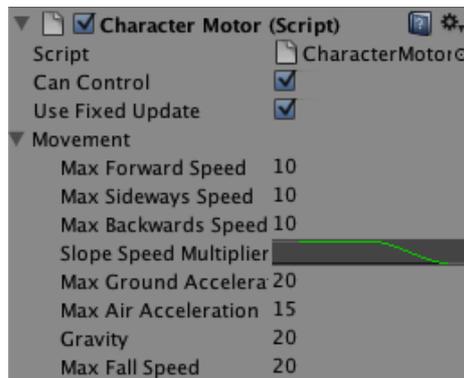
Nous allons nous contenter de ces paramètres de base dans notre exercice, mais succinctement :

- character controller : gère la taille du personnage, la longueur des pas etc...
- mouse look : gère la maniabilité souris, la sensibilité, l'angle de vue maximum/minimum
- character motor : concerne la physique du déplacement de manière détaillée
- fpsinput controller : le script qui permet d'utiliser les flèches du clavier pour se déplacer, la barre espace pour sauter

Il nous faut placer la caméra de Character Controller sur un morceau de notre île pour décider du point de départ de notre personnage. La gravité va se charger de le recaler sur le sol, plaçons le donc simplement au dessus d'une zone terrestre dans notre scène.

Appuyons sur play dans le controller en haut au centre de l'interface de Unity : Vous pouvez bouger la caméra avec la souris et diriger votre personnage avec les fleches du clavier dans le décor que vous venez de modeliser !

N'hésitez pas à changer les valeurs de vitesse du mouvement si vous trouvez votre personnage trop lent par exemple, en cliquant dans la Hierarchy sur l'objet Character Controller puis en modifiant les paramètres dans l'Inspector. Pour la vitesse de déplacement par exemple, nous avons modifié les valeurs de Max Forward Speed, Max Sideways Speed et Max Backwards Speed à 10 (6 de base) :

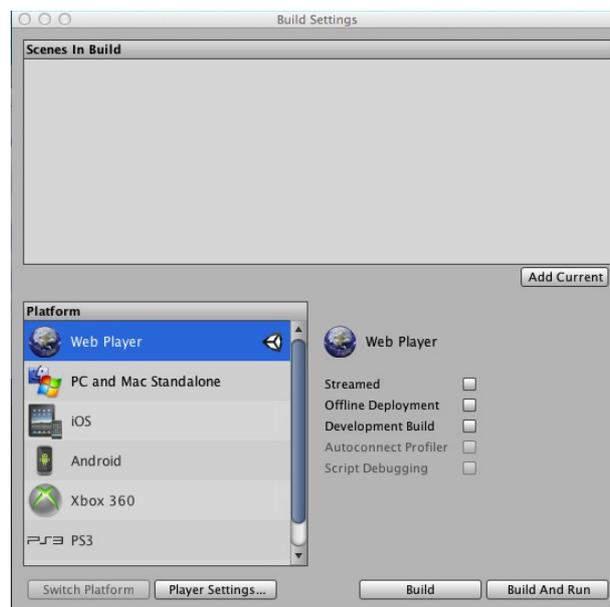


e) La publication

C'est bien de pouvoir tester dans le player intégré à Unity mais en faire une véritable application indépendante serait encore mieux ! Testons maintenant la publication.

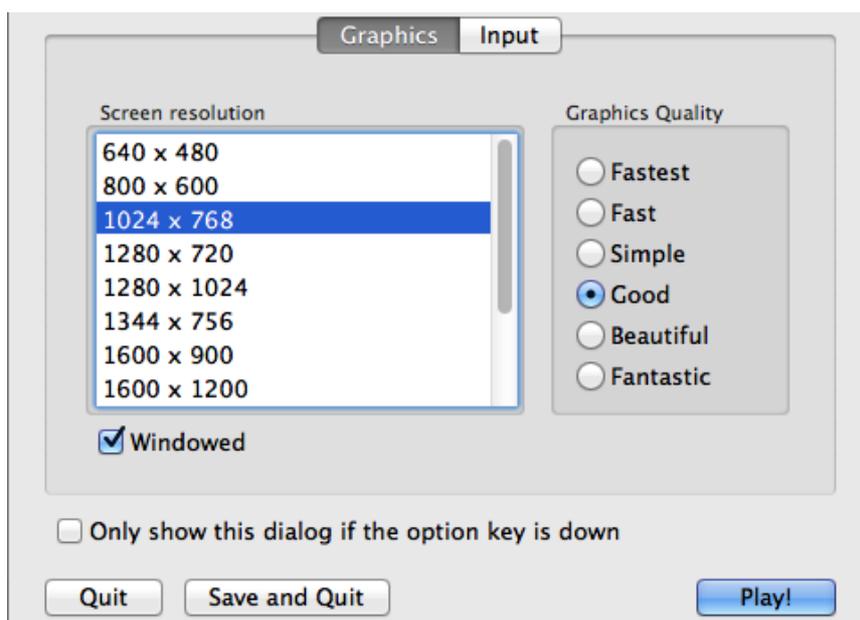
Comme nous l'expliquions en introduction vous avez à votre disposition plusieurs plateformes d'export possibles, dans notre cas nous allons choisir PC and MAC Standalone par ici :

Menu File > Build Settings...



Sélectionnons PC and Mac Standalone puis cliquons sur Build. La publication des projets dans Unity est appelé « build ». Selon la complexité de votre scène, le chargement va prendre plus ou moins de temps. On décide du nom final de notre application et de l'endroit où nous voulons l'enregistrer.

Retrouvons le projet publié sur notre ordinateur et lançons le, cette fenêtre apparaît :



Vous pouvez décider de la résolution et de la qualité du jeu, cette dernière impactant sur la finesse des ombres, lumières, particules, de la physique ou encore des shaders. Pour notre projet nous pouvons sans soucis le lancer en qualité Fantastic et au maximum de la résolution de l'écran que nous utilisons, tant le projet est simple.

Félicitations ! Vous avez crée votre premier projet en Unity 3D!