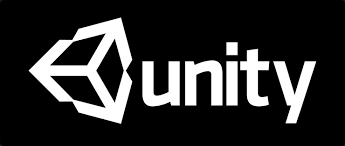
Tutoriel : les bases d’Unity 3D



# Lancer Unity 3D et créer un nouveau projet

Lancer le logiciel Unity 3D. Une fenêtre de dialogue apparaît, vous demandant de vous connecter avec un compte utilisateur. Vous avez donc deux options : soit vous vous créez un compte maintenant, soit vous choisissez l’option « Work offline ». Vous pourrez réaliser ce tutoriel sans problème en choisissant l’option « Work offline ». Cependant, si plus tard vous souhaitez récupérer des données de l’*asset store* Unity 3D (modèles 3D, scripts, etc.), vous aurez besoin d’un compte (il sera possible de créer et de vous connecter avec un compte plus tard même si vous choisissez l’option « Work offline » maintenant).

* Créer un nouveau projet Unity 3D en cliquant sur « NEW ». Choisir 3D, ainsi que le nom et la destination de votre projet. Vous n’avez pas besoin de charger de packages prédéfinis pour le moment (vous pourrez le faire par la suite ci-besoin).
* Enregistrer la scène courante (extension *.unity*) en lui donnant un nom (File > Save Scenes).

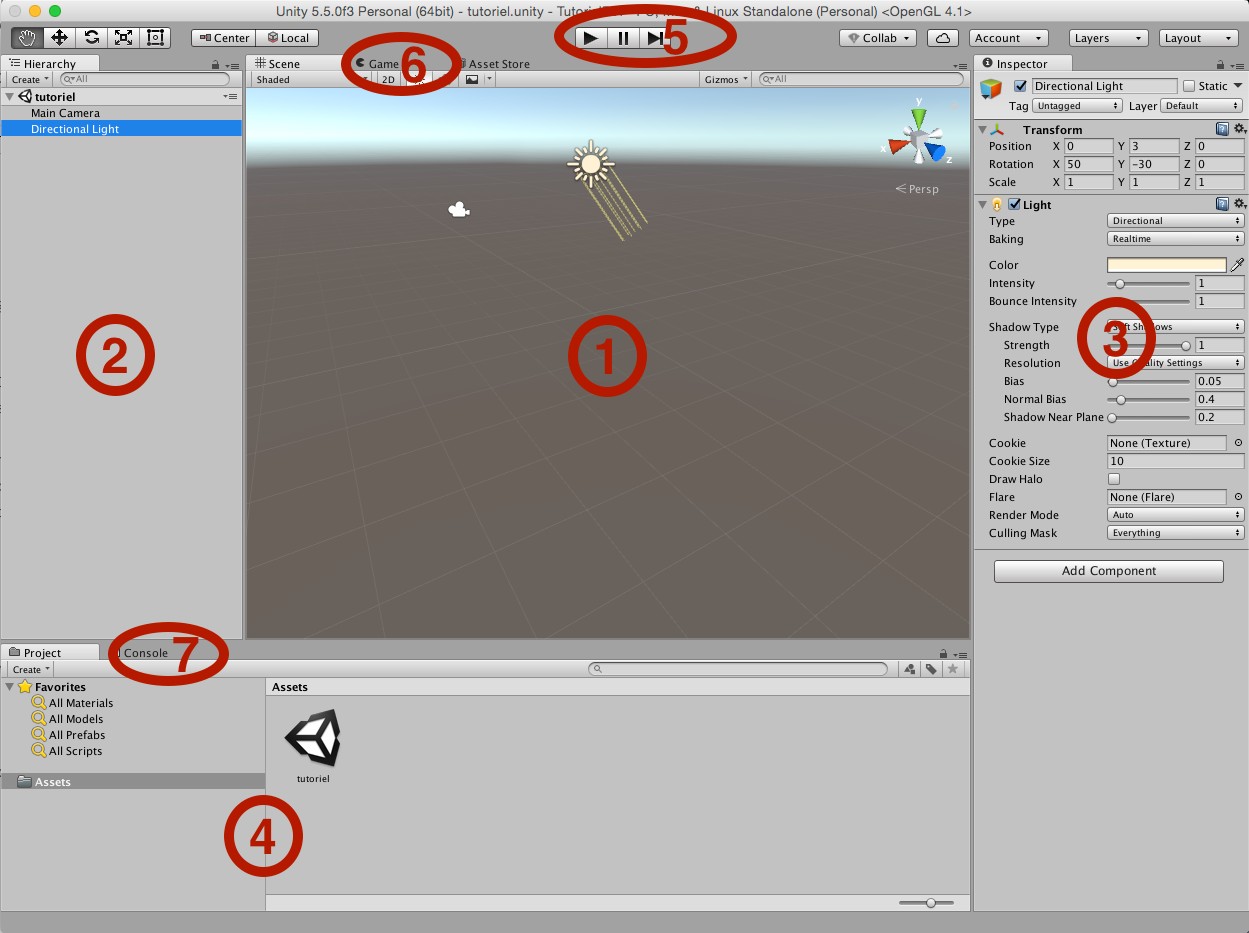
**Sauvegarder votre travail : projet et scenes.**

Unity 3D sauvegarde de façons différentes les informations à propos de votre projet. Cela veut donc dire que vous devez sauvegarde différemment votre travail selon les informations que vous avez modifiées :

* La scene correspond à la scène 3D contenant une hiérarchie d’objets (GameObject) ainsi que leur paramètres (position, etc.). Lors que vous effectuez une sauvegarde de la scene, le projet est également automatiquement sauvegardé. Plusieurs scenes peuvent être créées dans un même projet.
* Le projet permet de stocker les informations qui ne sont pas spécifiques aux scenes, comme par exemple les paramètres du projet, les scripts, les packages importés, les paramètres de « Build », etc.

# L’éditeur d’Unity 3D

Lorsque vous créez un projet, vous obtenez par défaut dans la scène une caméra (Main Camera) et une lumière pour éclairer la scène 3D (Directional Light). La fenêtre de l’éditeur se présente alors de la façon suivante :



Cette fenêtre de l’éditeur est composée des éléments suivants :

* 1. L’éditeur graphique de la scène 3D permettant de manipuler directement des objets de la scène.
  2. L’arborescence de la scène permettant de visualiser la hiérarchie des objets présents dans la scène.
  3. L’inspecteur permettant de voir les différents scripts rattachés à l’objet sélectionné (par 1 ou 2) et de modifier leurs paramètres.
  4. L’explorateur de projet permettant de voir les différents composants du projet et de les ajouter à la scène par un simple glisser-déposer. Pour l’instant, il y a seulement la scène dans le projet.
  5. Les boutons permettant de lancer l’exécution de la scène 3D dans l’éditeur ce qui est très utile pour tester et debugger la scène (nous verrons plus tard comment créer un exécutable pour la scène).
  6. La visualisation de la scène 3D à partir du point de vue de la caméra virtuelle. L’éditeur bascule automatiquement sur cette visualisation lorsque l’exécution de la scène est lancée.
  7. La console permettant de voir les messages d’erreurs ou les logs lors de l’exécution de la scène.

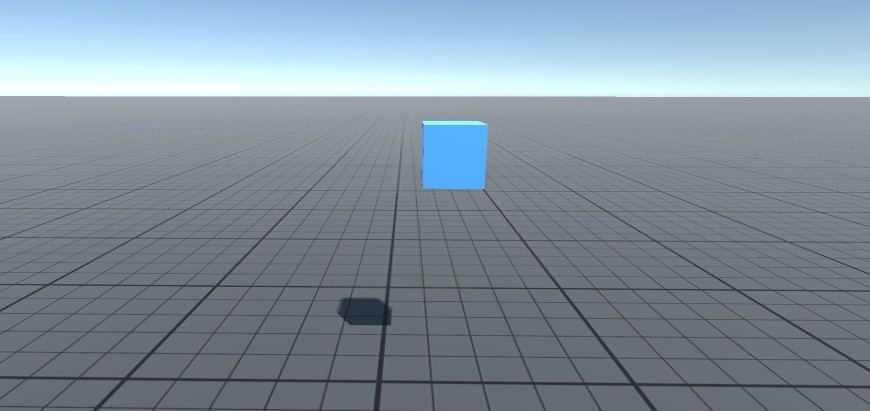
Les interactions dans l’éditeur graphique peuvent se faire à partir des boutons suivants :

# Créer une scène simple

* Insérer un plan dans la scène pour créer un sol (GameObject > 3D Object > Plane).

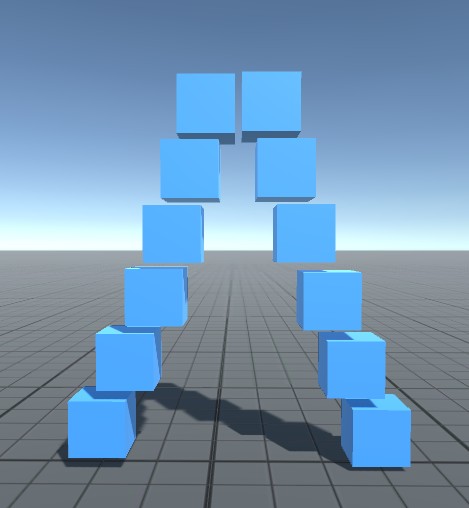
Au moyen de l’inspecteur (3), placer ce plan aux cordonnées (0, 0, 0) et vérifier qu’il est bien visible par la caméra maintenant. Agrandir ce plan en jouant sur l’échelle pour qu'il est une longueur et une largeur de 100 mètres et vérifier qu’il est bien visible à l’infini lorsque l’on prend la vue de la caméra.

* Ajouter un cube dans la scène (GameObject > 3D Object > Cube) et placer le pour qu’il soit visible par la caméra.
* Créer un material pour donner une couleur au cube (Assets > Create > Material). Un nouveau composant est créé dans le projet (4), donner lui un nom. Puis modifier sa couleur dans l’inspecteur pour obtenir le bleu de l’image suivante. Ensuite glisser-déposer le material à partir de l’explorateur de projet (4) sur le cube dans l’arborescence de la scène (2). Le cube devrait prendre la couleur bleue dans l’éditeur graphique (1).
* Ajouter une texture sur le plan : télécharger la texture *grid\_grey.png*, aller dans (Assets > Import New Asset…) et choisir la texture. Glisser-déposer la texture à partir de l’explorateur de projet (4) sur le plan dans l’arborescence de la scène (2). Puis sélectionner le plan dans l’arborescence de la scène (2) et modifier les paramètres du material dans l’inspecteur (3) afin d’obtenir la même texture du sol que dans l’image ci-dessous. Il faudra en particulier jouer sur le nombre de fois que la texture est répétée (« *tiling* »).
* Déplacer le cube et la caméra afin d’obtenir une image similaire lorsque vous exécutez la scène :



* Ajouter un comportement physique au cube : sélectionner le cube et lui ajouter un Rigidbody

(Component > Physics > Rigidbody). Observer ce qui se passe lorsque vous exécutez la scène.

* Glisser-déposer le cube depuis l’arborescence de la scène (2) vers l’explorateur de projet (4) : cela va créer un « Prefab » à partir du cube (cf. icône bleue) qui vous devrez renommer (CubePrefab par exemple). Puis, par l’opération inverse, insérer 11 autres cubes à partir de ce modèle dans la scène et déplacer les instances de ce cube afin d’obtenir la même disposition que dans l’image de droite.
* Exécuter la scène.
* Sélectionner le CubePrefab dans l’explorateur de projet (4) et modifier les paramètres de son Rigidbody afin de changer son comportement physique (en particulier, la masse « Mass » et la friction « Drag »). Exécuter la scène avec plusieurs paramètres

différents. Qu’observez-vous ?