

2

Création des niveaux

Dans ce chapitre, nous attaquons la création de notre jeu avec la conception d'un premier niveau. Nous ne nous occuperons pour l'instant que du visuel ; nous n'écrirons pas de code. Ce sera également l'occasion d'aborder les notions de solidité, de gravité et de collision.

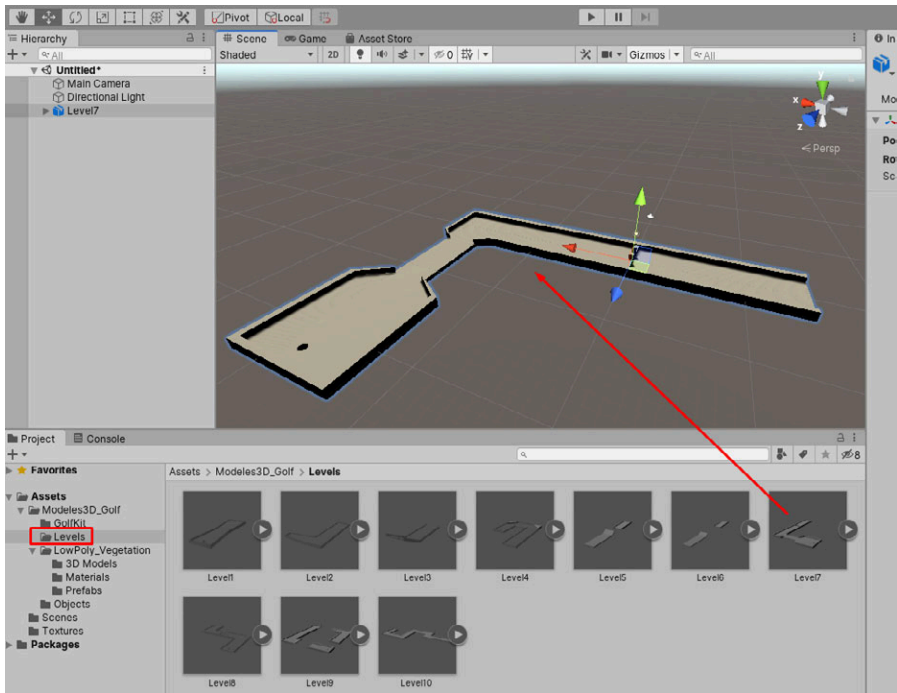
Figure 2.1 : Exemple de niveau que nous allons réaliser



2.1. Utilisation des modèles 3D

Pour que vous puissiez créer un jeu avec plusieurs niveaux fonctionnels, je vous ai fourni dix modélisations de niveaux que vous trouverez dans le dossier `Assets/Models3D_Golf/Levels`. Pour utiliser l'un de ces modèles, il vous suffit de le glisser/déposer de la fenêtre de projet vers la scène comme sur la [Figure 2.2](#).

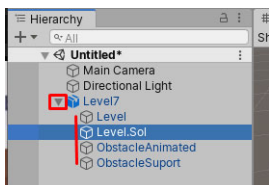
Figure 2.2 : Ajout d'un modèle 3D à la scène



Sur cet exemple, il s'agit du niveau 7 (le premier niveau sera une simple ligne droite). Je vous présente celui-là car il est composé d'un obstacle ; il sera plus intéressant à analyser.

Chaque niveau est composé de plusieurs sous-objets. Pour afficher ces objets enfants, passez par la hiérarchie et déployez le niveau via la petite flèche.

Figure 2.3 : Affichage des objets enfants

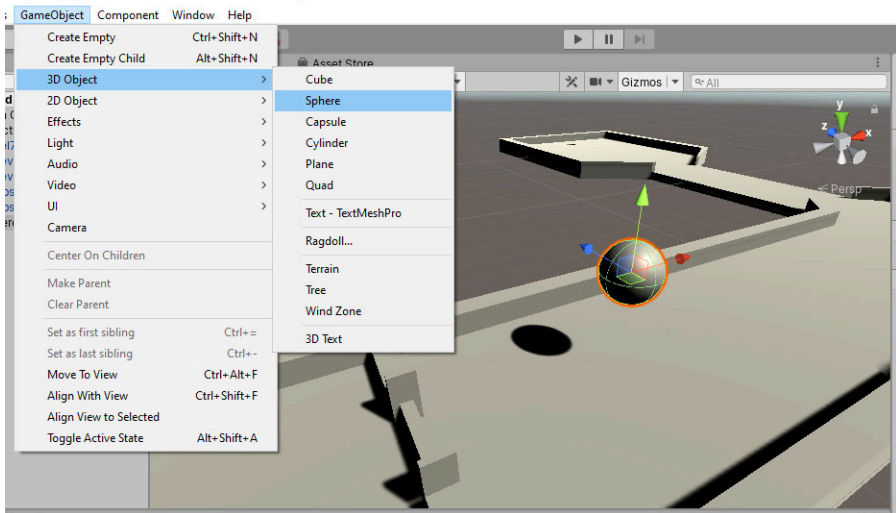


Vous constatez qu'ici le niveau est composé de quatre sous-objets. L'objet `Level` est le contour du niveau : la bordure empêchant la balle de tomber dans le vide. Parfois il n'y aura pas de contours. L'objet `Level.Sol` est le sol du niveau : la surface sur laquelle la balle peut rouler. Pour vous faciliter le travail, j'ai volontairement séparé le contour du sol. Cela permettra de colorer très facilement les niveaux avec des couleurs différentes pour chaque objet enfant.

Vous pouvez voir également deux objets `Obstacle`. Il s'agit d'un obstacle animé avec une barrière qui peut se lever et descendre. Cela nous permettra de découvrir le **système d'animation** de Unity.

Lorsque vous ajoutez un modèle 3D à la scène, par défaut il ne possède pas de composant de collision. Cela signifie que les objets pourront traverser votre modèle 3D. Dans le cas présent, ce sera problématique car la balle de golf tombera au travers du niveau. Pour que vous puissiez vous en rendre compte, nous allons créer une sphère 3D qui simulera la balle de golf. Cliquez sur le menu `GAMEOBJECT/3D OBJECT/SPHERE`. Une sphère apparaîtra alors.

Figure 2.4 : Création d'une sphère

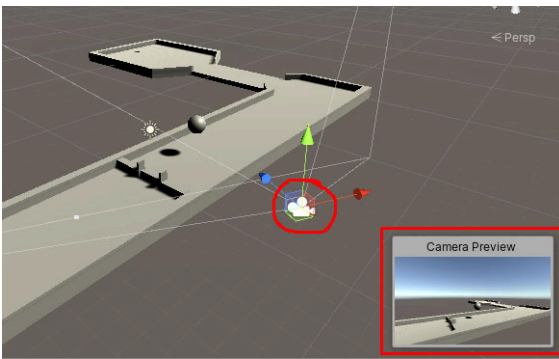


Déplacez la sphère et positionnez-la au-dessus du niveau, de manière un peu surélevée comme sur la [Figure 2.4](#). Pour déplacer la sphère, utilisez l'outil de déplacement et les flèches. Notre objectif va être de faire tomber la sphère sur le niveau pour vérifier s'il est solide ou si la balle passe au travers.

2.2. Playtest

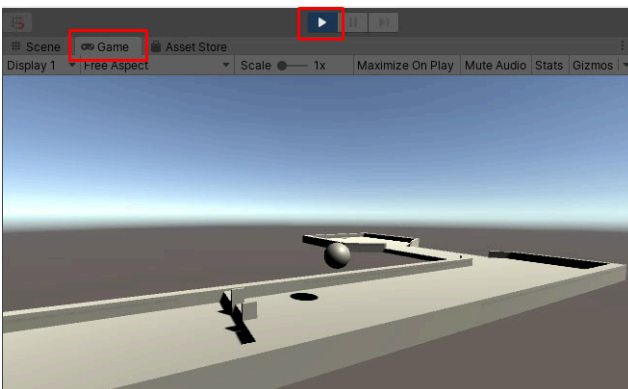
Pour visualiser l'action, nous devons ajouter une caméra à notre scène. Par chance, Unity en met une en place par défaut. Ce que filme la caméra correspond à ce que verra exactement le joueur lorsqu'il jouera à votre jeu. Pour déplacer et orienter la caméra, vous devez d'abord la sélectionner soit via l'inspecteur soit directement dans la scène en cliquant sur son icône. Une fois sélectionnée, utilisez les outils de translation et de rotation. Vous verrez une petite fenêtre de prévisualisation afin de suivre votre manipulation.

Figure 2.5 : Mise en place de la caméra



Lorsque la caméra est placée, vous pouvez tester votre jeu. Pour le lancer, cliquez en haut de l'écran sur l'icône PLAY de la barre d'outils. Vous verrez que Unity se mettra sur la fenêtre GAME au lieu de rester sur la scène. Une fois le jeu lancé, vous pouvez le tester.

Figure 2.6 : Test de notre jeu



Vous devez sans doute vous dire que quelque chose cloche. En effet, la balle reste en lévitation. Or nous voulons qu'elle tombe du fait de la gravité pour vérifier si notre niveau est solide ou non.

2.3. La gravité

Dans Unity, les `GameObjects` (caméra, modèles 3D) fonctionnent avec des composants (Components). Il existe des composants pour gérer le visuel, des composants spécifiques aux collisions ou encore des composants propres à la gravité. C'est ce dernier cas qui va nous intéresser. Pour qu'un objet soit soumis à la gravité, nous devons lui ajouter un `Rigidbody`. Le `Rigidbody` permet de donner une masse à un objet, de définir s'il est affecté ou non par la gravité, de définir des paramètres de frottement qui le freineront au cours du temps, etc.

Spécificité des jeux 2D

Unity dispose d'outils spécifiques pour le développement de jeux 2D. En effet, créer un jeu 3D est différent de la création d'un jeu 2D. Beaucoup de composants sont disponibles pour les deux variantes. Ainsi, vous trouverez le composant `Rigidbody` mais également le composant `Rigidbody2D`. Lorsqu'un composant se termine par 2D, il est spécifique à la 2D.

Quittez le jeu en cliquant de nouveau sur l'icône `PLAY` puis sélectionnez votre balle. Pour ajouter un `Rigidbody` à la balle vous pouvez passer soit par le menu `COMPONENT/PHYSIC/RIGIDBODY` soit par l'inspecteur et le bouton `ADD COMPONENT`. Dans ce dernier cas, saisissez dans le champ les premières lettres de votre composant, par exemple "rigid...", et Unity proposera le `Rigidbody`. Vous n'aurez qu'à cliquer pour l'ajouter à votre sphère.