

4

Les éléments préfabriqués

4.1. Comprendre les prefabs

Un jeu vidéo est un monde virtuel à part entière. Ce monde est composé d'éléments individuels comme par exemple des personnages, des arbres ou des maisons. Tous ces éléments sont créés par des personnes qui les assemblent pour constituer cet univers. Il faut bien comprendre que la modélisation d'un personnage, même secondaire, demande du temps et de l'argent. De plus, les modèles 3D doivent être stockés sur l'ordinateur du joueur. Il est impossible (humainement parlant et technologiquement parlant) de modéliser et de stocker chaque arbre d'une forêt par exemple. Vous avez certainement déjà joué à beaucoup de jeux dans lesquels les personnages étaient les mêmes (ou au mieux avec une texture différente). Si les concepteurs devaient modéliser chaque modèle 3D afin qu'il soit unique, la réalisation d'un jeu prendrait des années et il occuperait trop de place sur le disque. Les développeurs pratiquent donc la réutilisation de ressources. Comme vous pouvez le constater, la forêt de palmiers que nous avons créée dans le chapitre précédent n'est composée que d'un seul arbre. Nous pouvons tricher et changer le rendu d'un objet 3D en modifiant sa taille ou sa couleur afin de faire croire qu'il est différent. Cela nous fait gagner du temps et de la place en mémoire. C'est ce que nous allons apprendre à faire dans ce chapitre.

Qu'est-ce qu'un prefab ?

Un prefab est un objet constitué de plusieurs éléments. Par exemple, un personnage 3D constitué d'une texture, d'un objet de collision et d'un script de mouvement est un prefab car il s'agit d'un personnage préconfiguré prêt à être réutilisé ultérieurement. Prenons un autre exemple : une cabane en bois construite avec plusieurs cubes et plusieurs textures peut être stockée dans un prefab afin que nous puissions la réutiliser par la suite.

Comment créer un prefab ?

Avec Unity, la procédure est très simple. Vous devez dans un premier temps préparer un objet ; dans notre cas, nous allons créer une cabane en bois à partir de cubes. Puis, une fois l'objet créé, vous le glissez/déposez de la hiérarchie vers la fenêtre de projet. Nous allons faire la manipulation à la section suivante.

4.2. Créons un prefab de cabane

Pour commencer, nous devons créer une cabane en bois. Pour cela, nous avons besoin de planches en bois (cube + texture).

Du cube au pilier

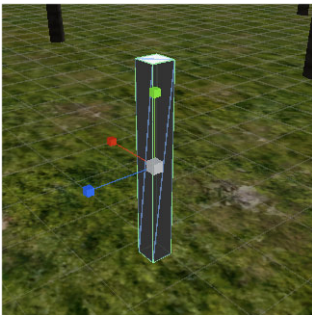
Construisons les quatre pieds qui vont porter la cabane. Nous allons créer le premier à partir d'un cube, puis nous le dupliquerons trois fois. Pour créer le cube, allez dans le menu GAME OBJECT/CREATE OTHER/CUBE. Le cube apparaît alors dans la scène. Dans la barre d'outils, sélectionnez l'outil SCALE permettant la déformation d'un objet :

Figure 4.1 : L'outil Scale



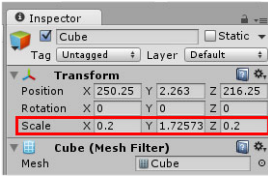
Maintenant vous pouvez déformer le cube afin de l'allonger et de diminuer son diamètre. Utilisez les trois axes pour effectuer la déformation :

Figure 4.2 : Déformation d'un cube



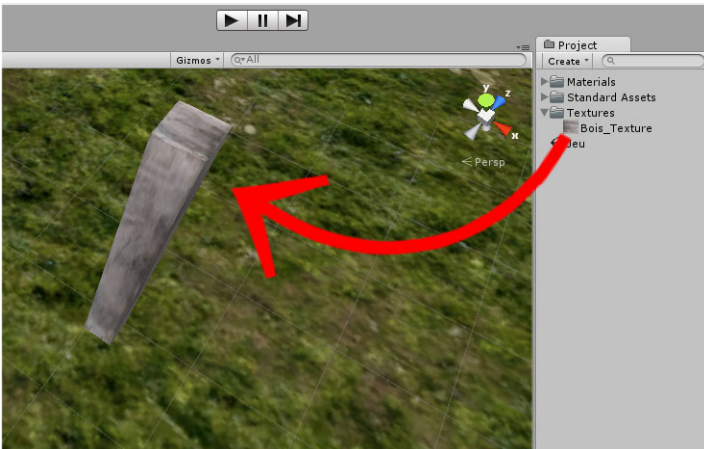
Le pilier prend forme. Sachez que vous pouvez aussi passer par l'inspecteur pour déformer votre cube avec plus de précision. Vous pouvez modifier manuellement les valeurs de la propriété SCALE du cube dans la section TRANSFORM :

Figure 4.3 : Déformation d'un cube via l'inspecteur



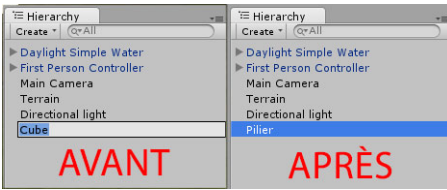
Nous allons maintenant ajouter une texture à notre pilier en bois. Une texture n'est ni plus ni moins qu'une photo représentant la surface d'un matériau qui, une fois appliquée à l'objet, lui en donne la consistance. Vous pouvez télécharger une texture de bois en tapant "wood texture" sur Google et en sélectionnant le visuel de votre choix. Pour un jeu commercial, vous devez faire très attention à n'utiliser que des textures libres de droits. Vous pouvez prendre n'importe quelle image ou récupérer celle que j'ai utilisée dans les sources téléchargeables de ce livre. Pour importer la texture dans Unity, vous devez la glisser/déposer dans la fenêtre de projet. Une fois la texture importée, vous pouvez l'appliquer à votre pilier en la faisant cette fois glisser de la fenêtre Projet sur l'objet dans la fenêtre Scène (voir Figure 4.4).

Figure 4.4 : Ajout d'une texture au pilier



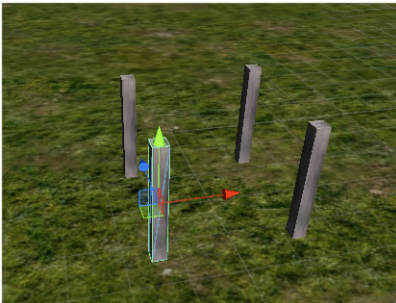
Vous allez maintenant organiser votre scène en nommant vos objets. Pour cela, cliquez sur l'élément Cube se trouvant dans la liste des objets de la hiérarchie et appuyez sur la touche F2 pour le renommer en *Pilier*.

Figure 4.5 : Modification du nom d'un objet



Nous avons besoin de quatre piliers pour construire notre cabane. Nous allons en créer un que nous allons ensuite dupliquer afin d'en obtenir trois copies supplémentaires. Pour dupliquer un objet, sélectionnez-le dans la scène en cliquant dessus, puis pressez les touches Ctrl+D. Vous pouvez aussi cliquer droit sur son nom dans la hiérarchie et sélectionner l'entrée DUPLICATE dans le menu contextuel. Répétez l'opération trois fois pour obtenir quatre piliers en bois. Utilisez l'outil TRANSFORM afin de déplacer les piliers pour les espacer.

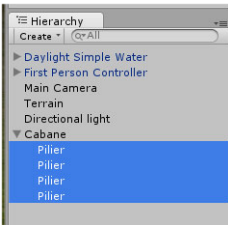
Figure 4.6 : Duplication du pilier



Créer un objet Cabane

Nous allons une nouvelle fois devoir organiser notre projet. Comme vous pouvez le constater, vous avez quatre objets Piliers dans la hiérarchie. Si vous créez trois cabanes, vous aurez douze objets avec le même nom. Afin de ne pas surcharger votre espace de travail, vous allez les regrouper au sein d'un objet unique Cabane. Cliquez sur le menu GAMEOBJECT/CREATE EMPTY : un objet vide nommé GameObject va alors apparaître dans la hiérarchie. Renommez-le *Cabane*, puis sélectionnez les quatre piliers et glissez/déposez-les dedans.

Figure 4.7 : Création de l'objet Cabane



Cela va nous permettre d'une part de mieux nous organiser et d'autre part de nous faciliter le travail lorsque nous créerons le prefab de notre cabane une fois celle-ci terminée.


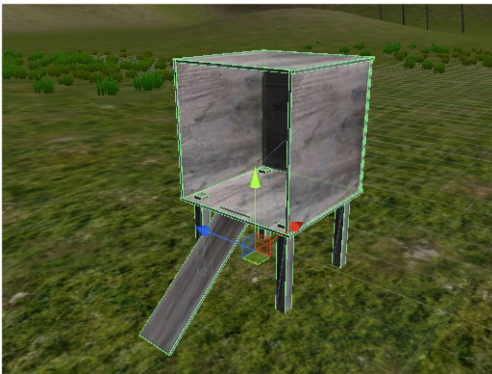
Nous allons maintenant créer le plancher, les murs et le toit de la cabane. Vous allez partir d'un nouveau cube que vous agrandirez et aplatirez pour former une sorte de planche en bois. N'oubliez pas d'ajouter la texture. Une fois terminé, dupliquez-le cinq fois (trois murs, un toit, un plancher) et glissez les cubes dans l'objet Cabane créé précédemment. Pensez à utiliser l'outil de rotation  pour faire tourner les cubes si besoin. Voici le résultat que j'ai obtenu :

Figure 4.8 : La cabane complète

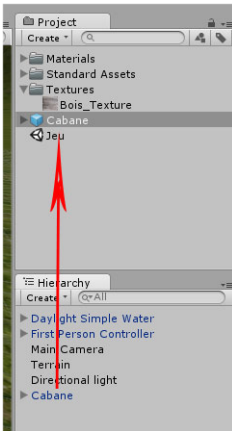


J'ai ajouté une planche permettant au joueur de pouvoir rentrer dans la cabane. Vous devriez obtenir une cabane similaire à la mienne mais vous êtes totalement libre de créer la cabane de vos rêves (plus grande, plusieurs pièces, plusieurs étages, etc.). Vous avez réussi à créer votre premier objet 3D en partant de plusieurs cubes !

De l'objet au prefab

Maintenant que vous vous êtes donné du mal pour obtenir ce bel objet, il faudrait pouvoir le réutiliser plusieurs fois dans votre jeu. Vous allez donc créer un prefab de votre cabane. Rien de plus simple : glissez/déposez-le de la hiérarchie vers la fenêtre Projet. Un cube bleu devrait alors apparaître dans votre projet. Ce cube est votre cabane. Pour créer une nouvelle cabane, vous n'aurez plus qu'à glisser/déposer le prefab de la fenêtre de projet vers la scène !

Figure 4.9 : Création du prefab



Comme vous le voyez, vous pouvez facilement créer des prefabs afin de réutiliser vos créations. Testez celui-ci en créant plusieurs cabanes dans votre scène. Pour donner l'illusion qu'elles sont différentes, vous pouvez changer la taille, l'orientation et même la texture si vous le souhaitez. Vous pouvez également supprimer ou ajouter des planches pour en modifier la structure sans pour autant tout recommencer depuis le début.

Note > Pensez à créer systématiquement des prefabs pour tous les objets que vous allez réutiliser dans votre jeu. Cela vous fera gagner du temps de développement et vous permettra de mieux vous organiser.

Organiser ses prefabs en package

Vous pouvez rassembler vos prefabs sous forme de packages. Ces packages seront exportés sur votre ordinateur et vous pourrez à tout moment les importer dans Unity, par exemple dans un futur projet de jeu. Pour exporter un package, sélectionnez tous les éléments que vous souhaitez exporter dans la fenêtre de projet, cliquez droit et sélection-

nez EXPORT PACKAGE dans le menu contextuel. Unity va lister tous les éléments à exporter ; cliquez sur EXPORT... en bas de la fenêtre pour spécifier un nom et choisir l'emplacement où vous souhaitez enregistrer le package. Si dans un futur projet vous souhaitez importer ce package, ouvrez le menu ASSETS/IMPORT PACKAGE/CUSTOM PACKAGE et allez chercher votre package fraîchement créé.

Vous savez désormais créer et réutiliser des objets. Vous pouvez donc encore enrichir votre monde virtuel. Dans le prochain chapitre, les choses vont commencer à se corser car nous allons apprendre à programmer nos premiers scripts C# pour pouvoir interagir avec l'environnement et permettre de ramasser des objets. Avant de coder ce script, nous aborderons quelques concepts de base du langage.

Au cours de ce chapitre, vous avez vu comment créer des objets 3D et les réutiliser en passant par le système de prefabs de Unity. Vous savez donc :

- modéliser un objet à partir de primitives ;
- créer un prefab ;
- utiliser des prefabs ;
- exporter des packages.

Note > On appelle primitive une forme simple comme un cube, une sphère, etc.