

# Structure d'un moteur de jeu : scène, objets, assets, scripts

Dans Unity, le moteur de jeu repose sur une structure bien définie qui permet aux développeurs de créer des jeux 2D et 3D de manière modulaire et flexible. Ce cours couvre les éléments fondamentaux qui constituent la structure d'un moteur de jeu Unity : la **scène**, les **objets**, les **assets**, et les **scripts**. Ces éléments sont les briques de base avec lesquelles vous construisez vos jeux.

---

## I. La Scène dans Unity

### 1. Qu'est-ce qu'une scène ?

Une scène dans Unity est un environnement ou un niveau de jeu où se déroulent les événements. Elle contient tous les objets du jeu, les éléments de l'interface utilisateur, les sources lumineuses, les caméras, les scripts, et tout ce qui constitue le jeu dans un environnement donné.

- **Exemple** : Un jeu de plateforme pourrait avoir plusieurs scènes : un menu principal, un niveau de jeu, un écran de victoire, etc.

### 2. Structure de la scène

Chaque scène dans Unity est construite autour de **GameObjects** et peut être vue et manipulée dans l'éditeur de scène.

- **GameObject** : Les objets dans Unity sont des instances d'un **GameObject**. Chaque GameObject peut avoir un ou plusieurs composants (comme des rendus, des scripts, des colliders, etc.).
- **Hiérarchie** : Les objets d'une scène sont organisés dans une vue **Hiérarchie**, qui est une liste des GameObjects présents dans la scène, et qui représente une structure hiérarchique (parent-enfant). Par exemple, un personnage peut être un objet parent contenant plusieurs objets enfants comme des accessoires (pistolet, épée, etc.).

### 3. Les scènes multiples

Vous pouvez créer plusieurs scènes dans un même projet Unity. Par exemple :

- **Menu Principal** : Scène affichant les options du jeu.
- **Niveau 1** : Scène représentant le premier niveau du jeu.
- **Game Over** : Scène qui affiche un message de fin de jeu.

Il est possible de charger et de basculer entre ces scènes à l'aide des scripts.

---

## II. Les Objets dans Unity

### 1. Qu'est-ce qu'un GameObject ?

Le **GameObject** est l'élément fondamental dans Unity. Il représente une entité dans la scène, que ce soit un personnage, un objet interactif, une caméra, une lumière, ou tout autre élément du jeu.

- Un GameObject peut être vide, mais il doit avoir au moins un **composant** pour lui donner des caractéristiques ou comportements spécifiques (par exemple, un modèle 3D, un script, un collider, etc.).

### 2. Les composants d'un GameObject

Chaque GameObject peut avoir plusieurs composants qui définissent ses propriétés et son comportement. Par exemple :

- **Transform** : Ce composant est automatiquement attaché à chaque GameObject et définit sa position, sa rotation et son échelle dans l'espace.
- **Render** : Ce composant permet d'afficher un modèle 3D ou une image (sprite) dans la scène.
- **Collider** : Permet à un GameObject d'interagir avec d'autres objets en détectant des collisions (utile pour la physique).
- **Rigidbody** : Ajoute des comportements physiques à un GameObject (comme la gravité ou les forces).
- **Scripts** : Un GameObject peut également avoir des scripts attachés qui contrôlent son comportement via du code.

### 3. Manipulation des GameObjects

Les GameObjects peuvent être manipulés de différentes façons dans l'éditeur Unity :

- **Déplacement, Rotation, Échelle** : Dans la **Scene View**, vous pouvez manipuler directement les GameObjects en utilisant les outils **Move**, **Rotate**, et **Scale**.
  - **Hierarchie** : Vous pouvez faire glisser des GameObjects dans la hiérarchie pour les organiser en parent/enfant.
- 

## III. Les Assets dans Unity

### 1. Qu'est-ce qu'un Asset ?

Les **Assets** sont tous les éléments que vous utilisez dans votre jeu, tels que les modèles 3D, textures, sons, animations, scripts, matériaux, etc. Ils sont stockés dans le dossier **Assets** du projet Unity, et Unity les convertit en formats optimisés pour le moteur.

- **Exemples d'Assets** : Textures, modèles 3D (FBX, OBJ), sons (WAV, MP3), animations (animators), scripts (C#).

### 2. Importation des Assets

Vous pouvez importer des assets dans Unity de différentes manières :

- **Drag-and-drop** : Faites glisser un fichier directement dans le dossier **Assets** dans l'éditeur.
- **Menu d'importation** : Allez dans **Assets > Import New Asset** pour sélectionner manuellement un fichier à importer.
- **Unity Asset Store** : Unity propose un magasin d'assets où vous pouvez acheter ou télécharger des modèles, scripts, textures, etc.

Une fois les assets importés, Unity les convertit automatiquement en formats compatibles et les affiche dans le dossier **Project** de l'éditeur.

### 3. Types d'Assets

- **Textures** : Images utilisées pour appliquer des surfaces (par exemple, des murs ou des personnages).
- **Modèles 3D** : Objets tridimensionnels qui peuvent être importés dans Unity au format FBX, OBJ, ou 3DS.

- **Audio** : Sons utilisés dans le jeu, comme la musique de fond ou les effets sonores.
- **Animations** : Séquences d'animations qui définissent les mouvements d'un objet, d'un personnage, ou d'un environnement.
- **Scripts** : Fichiers C# contenant du code qui définit le comportement du jeu.

## 4. Manipulation des Assets

- **Inspector** : L'outil **Inspector** dans Unity vous permet de modifier les propriétés des assets après les avoir sélectionnés dans le dossier **Project**. Par exemple, vous pouvez ajuster la texture d'un matériau, modifier les paramètres d'un son, ou configurer un script.
- **Références aux Assets** : Un GameObject peut référencer un asset via un composant, par exemple, un **Material** pour appliquer une texture à un objet.

## IV. Les Scripts dans Unity

### 1. Qu'est-ce qu'un script ?

Un **script** dans Unity est un fichier de code qui contrôle le comportement des GameObjects et des éléments de votre jeu. Unity utilise principalement le langage **C#** pour les scripts.

### 2. Création et gestion des scripts

- **Créer un script** : Dans le **Project** window, faites un clic droit dans le dossier **Assets** et sélectionnez **Create > C# Script**. Donnez-lui un nom, puis ouvrez-le pour commencer à coder.
- **Éditeur de scripts** : Unity s'intègre avec des éditeurs de code comme **Visual Studio**. Lorsque vous double-cliquez sur un script, il s'ouvre automatiquement dans l'éditeur externe.

### 3. Structure d'un script Unity

Un script Unity hérite de la classe **MonoBehaviour** et comprend des méthodes principales :

- **Start()** : Appelée au début de l'exécution, après que tous les objets soient initialisés.

- **Update()** : Appelée chaque image du jeu (frame). C'est ici que vous gérez les actions qui doivent être vérifiées constamment, comme les entrées de l'utilisateur.
- **Awake()** : Appelée avant **Start()**. Utile pour les initialisations qui doivent être effectuées avant le démarrage du jeu.

Exemple de script simple :

```
using UnityEngine;

public class PlayerController : MonoBehaviour
{
    public float speed = 10f;

    // Update is called once per frame
    void Update()
    {
        float move = Input.GetAxis("Horizontal") * speed *
        Time.deltaTime;
        transform.Translate(move, 0, 0);
    }
}
```

Ce script permet de déplacer un objet horizontalement en fonction des entrées du joueur (généralement les touches fléchées ou "A" et "D").

## 4. Attacher des scripts à des GameObjects

Une fois qu'un script est créé, vous pouvez l'attacher à un GameObject :

1. Sélectionnez un GameObject dans la scène.
2. Glissez le script depuis le dossier **Assets** dans l'**Inspector** du GameObject, ou utilisez **Add Component** pour l'ajouter.

## V. Résumé de la Structure d'un Moteur de Jeu Unity

- **Scène** : Environnement de jeu, contient les objets et les éléments interactifs.

- **GameObjects** : Entités fondamentales qui représentent des objets dans la scène. Ils possèdent des composants pour définir leur comportement.
  - **Assets** : Ressources externes utilisées dans le jeu, comme des textures, des modèles 3D, des sons et des scripts.
  - **Scripts** : Fichiers C# qui contrôlent le comportement des GameObjects et les interactions dans le jeu.
- 

## Conclusion

La structure d'un moteur de jeu Unity repose sur une organisation claire des scènes, objets, assets, et scripts. Chaque élément joue un rôle crucial dans le développement du jeu. Maîtriser cette structure est essentiel pour pouvoir créer des jeux interactifs et dynamiques. En suivant ce cours, vous avez appris à organiser votre projet Unity et à comprendre comment chaque composant interagit pour donner vie à votre jeu.