

# Écriture d'un script simple pour déplacer un objet (ex. contrôle du joueur)

Dans ce cours, nous allons apprendre à écrire un script simple pour déplacer un objet dans **Unity** en utilisant le langage **C#**. Ce type de script est essentiel dans le développement de jeux, notamment pour les jeux de type plateforme, de simulation ou d'aventure, où le joueur doit contrôler un personnage ou un objet.

---

## 1. Introduction aux Scripts de Mouvement

Dans Unity, les objets d'un jeu sont manipulés par des scripts. Un script de mouvement permet de déplacer un objet en fonction des entrées de l'utilisateur (comme les touches du clavier ou un joystick). Ce script prend en compte les commandes de mouvement et applique une translation ou une rotation à l'objet.

---

## 2. Composants nécessaires

Pour déplacer un objet, nous avons besoin de quelques composants essentiels dans Unity :

- **Rigidbody** : C'est un composant qui permet à un objet d'être affecté par la physique (gravité, collisions). Lorsqu'on applique des mouvements à un objet via un **Rigidbody**, Unity gère la simulation de ces mouvements de manière réaliste.
- **Collider** : Un composant permettant de détecter les collisions entre les objets.

Cependant, pour un simple déplacement d'objet sans gestion de physique complexe, nous allons principalement utiliser la méthode `Transform.Translate` dans ce cours.

---

### 3. Création du Script

Nous allons créer un script simple qui permet de déplacer un objet selon les entrées du clavier. Pour cela, nous allons utiliser les touches **fléchées** ou **WASD** pour déplacer l'objet dans un environnement 3D.

#### Étape 1 : Créer un nouveau script C#

1. Dans Unity, ouvrez le **Project Window** et faites un clic droit dans le dossier **Assets**.
2. Sélectionnez **Create > C# Script**.
3. Nommez le script **PlayerMovement**.
4. Double-cliquez sur le script pour l'ouvrir dans votre éditeur de code (Visual Studio, par exemple).

#### Étape 2 : Écrire le Code

Voici le code simple pour déplacer un objet :

```
using UnityEngine;

public class PlayerMovement : MonoBehaviour
{
    // Vitesse de déplacement du joueur
    public float moveSpeed = 5f;

    // Update est appelé une fois par frame
    void Update()
    {
        // Récupérer les entrées du joueur sur l'axe horizontal et vertical
        float horizontal = Input.GetAxis("Horizontal"); // A/D ou flèches gauche/droite
        float vertical = Input.GetAxis("Vertical");      // W/S ou flèches haut/bas

        // Créer un vecteur de mouvement basé sur les entrées du joueur
        Vector3 movement = new Vector3(horizontal, 0, verti
```

```

    cal) * moveSpeed * Time.deltaTime;

    // Appliquer le mouvement à la position de l'objet
    transform.Translate(movement);
}
}

```

## Explication du code :

- `public float moveSpeed = 5f;` : Déclare une variable publique `moveSpeed` qui détermine la vitesse de déplacement de l'objet. Cette valeur peut être modifiée directement depuis l'éditeur Unity.
- `float horizontal = Input.GetAxis("Horizontal");` : Récupère l'entrée de l'utilisateur sur l'axe horizontal (touches `A` / `D` ou flèches gauche/droite). `Input.GetAxis` renvoie une valeur flottante entre -1 et 1 en fonction de la direction.
- `float vertical = Input.GetAxis("Vertical");` : Récupère l'entrée de l'utilisateur sur l'axe vertical (touches `W` / `S` ou flèches haut/bas).
- `Vector3 movement = new Vector3(horizontal, 0, vertical) * moveSpeed * Time.deltaTime;` : Crée un vecteur de déplacement basé sur les entrées horizontales et verticales. Le paramètre `Time.deltaTime` assure que le mouvement soit indépendant du taux de frames par seconde (FPS), garantissant ainsi un déplacement fluide.
- `transform.Translate(movement);` : Déplace l'objet en fonction du vecteur de mouvement. Cette méthode applique une translation (déplacement) sur l'objet dans la scène.

## 4. Attacher le Script à un Objet

Une fois le script écrit, il faut l'attacher à un objet dans la scène pour qu'il soit exécuté.

1. Sélectionnez l'objet que vous souhaitez déplacer dans la scène (par exemple, un cube ou un personnage).
2. Faites glisser le script `PlayerMovement` depuis l'**Inspector** sur cet objet, ou cliquez sur **Add Component** et recherchez `PlayerMovement`.
3. Ajustez la **vitesse de déplacement** dans l'inspecteur si nécessaire.

## 5. Tester le Mouvement

Pour tester le déplacement de l'objet, procédez comme suit :

1. Cliquez sur le bouton **Play** en haut de l'interface Unity.
2. Utilisez les touches **fléchées** ou **WASD** pour déplacer l'objet dans la scène.
3. L'objet se déplacera dans les directions définies par les touches, en fonction de la vitesse que vous avez définie.

---

## 6. Améliorations possibles

Ce script peut être amélioré de plusieurs façons pour rendre le mouvement plus réaliste ou plus adapté à vos besoins :

- **Ajouter une rotation** : Faire tourner l'objet dans la direction de son mouvement.

Exemple de code pour ajouter une rotation :

```
transform.Rotate(Vector3.up * horizontal * rotationSpeed * Time.deltaTime);
```

- **Déplacement avec Rigidbody** : Utiliser un **Rigidbody** pour appliquer un mouvement basé sur la physique du jeu (par exemple, pour avoir des réactions réalistes aux collisions).

Exemple de code avec Rigidbody :

```
Rigidbody rb = GetComponent<Rigidbody>();  
Vector3 move = new Vector3(horizontal, 0, vertical) * moveSpeed * Time.deltaTime;  
rb.MovePosition(transform.position + move);
```

---

## 7. Conclusion

Dans ce cours, vous avez appris à créer un script simple pour déplacer un objet dans Unity en utilisant le langage C#. Vous avez vu comment récupérer les entrées du joueur pour déplacer l'objet dans l'espace 3D et comment appliquer ce mouvement en utilisant `transform.Translate()`. Ce type de script est fondamental pour créer des interactions de base dans un jeu et peut être étendu pour gérer des contrôles plus complexes ou des physiques avancées.

Les scripts de mouvement constituent la base du développement de nombreux types de jeux, qu'il s'agisse de jeux de plateforme, d'aventure ou d'action.