

Ajout de Composants Physiques aux Objets (Rigidbody, Collider)

Les composants **Rigidbody** et **Collider** sont essentiels pour donner des comportements physiques réalistes aux objets dans Unity. Dans ce cours, vous apprendrez comment ces composants interagissent, comment les ajouter à vos objets et comment ils influencent le mouvement, la collision et la simulation physique dans votre jeu.

1. Le Composant Rigidbody

Le **Rigidbody** permet à un objet de réagir aux forces physiques dans Unity, comme la gravité, les forces appliquées par le joueur ou des objets interactifs. Il transforme un objet en un corps physique qui peut être influencé par des événements physiques, comme les collisions, la gravité, ou des forces externes.

Rôle du Rigidbody :

- Appliquer la **gravité** : L'objet sera attiré vers le bas à moins que la gravité ne soit désactivée.
- Gérer les **forces physiques** : Comme les impulsions, les forces appliquées par des collisions ou des événements.
- Permettre l'utilisation de **mouvements physiques réalistes** comme le rebond, la friction, et les rotations.
- Manipuler des **objets dynamiques** dans la scène en fonction des interactions physiques.

Propriétés principales du Rigidbody :

- **Mass** : La masse de l'objet, influençant la force nécessaire pour le déplacer.

- **Drag** : La résistance à l'air (lenteur du mouvement de l'objet).
- **Angular Drag** : La résistance à la rotation.
- **Use Gravity** : Active ou désactive l'effet de la gravité sur l'objet.
- **Is Kinematic** : Permet de contrôler l'objet via le script sans qu'il soit affecté par les forces physiques.
- **Collision Detection** : Détermine la méthode utilisée pour détecter les collisions avec d'autres objets (par exemple, **Discrete**, **Continuous**).

Ajouter un Rigidbody à un Objet :

1. Sélectionnez l'objet dans la hiérarchie.
 2. Dans le panneau **Inspector**, cliquez sur **Add Component**.
 3. Tapez "Rigidbody" et cliquez pour l'ajouter.
-

2. Le Composant Collider

Le **Collider** est un autre composant clé pour la gestion des interactions physiques dans Unity. Un Collider définit la forme géométrique d'un objet et permet à Unity de détecter les collisions avec d'autres objets.

Rôle du Collider :

- Définir la forme de collision de l'objet pour détecter les impacts avec d'autres objets.
- Déterminer si un objet peut être traversé ou non par un autre objet.
- Appliquer des forces physiques en cas de collision.

Types de Colliders :

- **BoxCollider** : Un cube ou un rectangle utilisé pour des objets de forme simple.
- **SphereCollider** : Un Collider sphérique, idéal pour des objets ronds.
- **CapsuleCollider** : Utilisé pour des objets ayant la forme d'un cylindre (par exemple, les personnages).
- **MeshCollider** : Utilisé pour des objets ayant une forme complexe. Cependant, ce type est plus coûteux en termes de performance.

- **TerrainCollider** : Utilisé pour les terrains, permettant de gérer les collisions avec la surface du terrain.
- **WheelCollider** : Utilisé pour les véhicules, permettant de gérer les collisions spécifiques aux roues.

Propriétés principales des Colliders :

- **Is Trigger** : Si activé, le Collider ne générera pas de réactions physiques (pas de rebond), mais il déclenchera des événements de type **Trigger** (entrée, sortie, ou maintien dans la zone).
- **Center** : Le centre du Collider dans l'espace de l'objet.
- **Size / Radius** : Détermine la taille du Collider (utile pour les BoxCollider et SphereCollider).

Ajouter un Collider à un Objet :

1. Sélectionnez l'objet dans la hiérarchie.
2. Dans le panneau **Inspector**, cliquez sur **Add Component**.
3. Tapez "Collider" et sélectionnez le type de Collider approprié (BoxCollider, SphereCollider, etc.).

3. Interactions entre Rigidbody et Collider

Lorsque vous ajoutez un **Rigidbody** à un objet, vous devez également ajouter un **Collider** pour que Unity puisse détecter et répondre aux collisions.

L'interaction entre ces deux composants permet à Unity de calculer les impacts physiques et de réagir en conséquence.

Exemples d'interaction :

- **Sans Rigidbody mais avec Collider** : L'objet peut détecter des collisions (si **Is Trigger** est désactivé), mais il ne réagira pas physiquement (l'objet ne bougera pas en fonction des collisions).
- **Avec Rigidbody et Collider** : L'objet réagira physiquement aux collisions et aux forces. Si les objets se déplacent l'un vers l'autre, Unity appliquera une réaction physique (réflexion, force d'impact, etc.).
- **Avec Rigidbody, Collider, et Is Trigger activé** : L'objet ne réagira pas physiquement, mais des événements de **Trigger** seront déclenchés.

4. Exemple Pratique : Déplacement et Collision avec un Rigidbody

Prenons un exemple d'application dans Unity pour déplacer un objet avec un **Rigidbody** et détecter des collisions avec un autre objet.

Étapes :

1. Créez un cube dans la scène pour représenter l'objet mobile.
2. Ajoutez un **Rigidbody** à ce cube.
3. Ajoutez un **BoxCollider** au cube et activez l'option **Is Trigger** si vous souhaitez simplement détecter la présence sans réaction physique.
4. Créez un autre objet (par exemple, une sphère) avec un **Collider** (et éventuellement un **Rigidbody**) pour provoquer la collision.
5. Utilisez le script suivant pour déplacer le cube en utilisant le **Rigidbody** :

```
using UnityEngine;

public class MoveObject : MonoBehaviour
{
    public float moveSpeed = 10f;

    private Rigidbody rb;

    void Start()
    {
        // Récupérer le Rigidbody attaché à cet objet
        rb = GetComponent<Rigidbody>();
    }

    void Update()
    {
        // Déplacer l'objet avec le Rigidbody
        float moveHorizontal = Input.GetAxis("Horizontal");
        float moveVertical = Input.GetAxis("Vertical");

        // Appliquer une force de déplacement
        Vector3 movement = new Vector3(moveHorizontal, 0, m
```

```

oveVertical);
    rb.AddForce(movement * moveSpeed);
}

// Fonction pour détecter la collision avec un autre ob
jet
void OnCollisionEnter(Collision collision)
{
    if (collision.gameObject.CompareTag("Obstacle"))
    {
        Debug.Log("Collision avec un obstacle !");
    }
}
}

```

- Ce script permet de déplacer l'objet en fonction des entrées du joueur (les touches fléchées ou WASD), en appliquant une force avec le **Rigidbody**.
- L'événement **OnCollisionEnter()** détecte si le cube entre en collision avec un objet ayant le tag "Obstacle".

5. Utilisation de Rigidbody en Mode Kinematic

Le **Mode Kinematic** du **Rigidbody** permet de contrôler l'objet par script sans que celui-ci soit affecté par les forces physiques, comme la gravité ou les collisions.

Quand utiliser le Mode Kinematic ?

- **Déplacer un objet sans physique** : Si vous voulez déplacer un objet de manière contrôlée sans qu'il soit influencé par la physique.
- **Utilisation avec des animations** : Lorsque vous voulez animer un objet sans qu'il soit affecté par la physique.

Pour activer le mode Kinematic :

1. Sélectionnez l'objet avec le **Rigidbody**.
2. Dans le panneau **Inspector**, cochez l'option **Is Kinematic** dans le composant **Rigidbody**.

6. Conclusion

Les composants **Rigidbody** et **Collider** sont des éléments essentiels pour intégrer la physique dans vos jeux Unity. Le **Rigidbody** permet de simuler des forces physiques réalistes, tandis que le **Collider** définit la forme de l'objet pour la détection des collisions. Ensemble, ils permettent de créer des interactions dynamiques et des comportements réalistes dans un jeu.

Il est important de comprendre quand et comment utiliser chaque composant pour que vos objets réagissent de manière appropriée aux événements physiques, tout en optimisant les performances du jeu.