

Détection des touches du clavier

Dans Unity, vous pouvez détecter les touches du clavier en utilisant la classe `Input`. Cette classe fournit plusieurs méthodes pour vérifier si une touche spécifique est enfoncée, relâchée ou maintenue enfoncée. Voici quelques exemples pour vous aider à comprendre comment détecter les touches du clavier.

1.1 Détecter si une touche est enfoncée

Pour détecter si une touche spécifique est enfoncée pendant la frame actuelle, vous pouvez utiliser la méthode `Input.GetKeyDown()`. Cette méthode retourne `true` uniquement lors de la frame où la touche est enfoncée.

La méthode prend en paramètre un `KeyCode`, qui est une énumération représentant toutes les touches du clavier.

Des exemples courants de `KeyCode` incluent :

- `KeyCode.Space` pour la touche Espace
- `KeyCode.Return` pour la touche
- `KeyCode.A` pour la touche A
- `KeyCode.LeftArrow` pour la flèche gauche
- `KeyCode.RightArrow` pour la flèche droite
- `KeyCode.UpArrow` pour la flèche haut
- `KeyCode.DownArrow` pour la flèche bas
- `KeyCode.Escape` pour la touche Échappe

Voici un exemple de code qui détecte si la touche Espace est enfoncée :

```
void Update()
{
    if (Input.GetKeyDown(KeyCode.Space)==true)
    {
        Debug.Log("La touche Espace a été enfoncée.");
    }
}
```

1.2 Détecter si une touche est relâchée

Comme pour la détection de l'enfoncement d'une touche, vous pouvez utiliser la méthode `Input.GetKeyUp()` pour détecter si une touche spécifique a été relâchée pendant la frame actuelle. Cette méthode retourne `true` uniquement lors de la frame où la touche est relâchée.

```
void Update()
{
    if (Input.GetKeyUp(KeyCode.Space)==true)
    {
        Debug.Log("La touche Espace a été relâchée.");
    }
}
```

1.3 Détecter si une touche est maintenue enfoncée

Pour détecter si une touche spécifique est maintenue enfoncée, vous pouvez utiliser la méthode `Input.GetKey()`. Cette méthode retourne `true` tant que la touche est enfoncée, ce qui permet de vérifier l'état de la touche à chaque frame.

```
void Update()
{
    if (Input.GetKey(KeyCode.Space)==true)
    {
        Debug.Log("La touche Espace est maintenue enfoncée.");
    }
}
```

1.4 Où placer la détection des touches

La détection des touches doit être placée dans la méthode `Update()` de votre script. La méthode `Update()` est appelée une fois par frame, ce qui permet de vérifier l'état des touches à chaque frame.

1.5 Actions prédéfinies pour les touches

1.5.1.GetAxis

Unity propose également des actions prédéfinies pour certaines touches courantes, telles que les touches fléchées pour le déplacement. Vous pouvez utiliser ces actions en utilisant les méthodes `Input.GetAxis()`. Par exemple, pour détecter le mouvement horizontal avec les touches fléchées gauche et droite ou un joystick, vous pouvez utiliser l'axe "Horizontal". Pour détecter un mouvement vertical avec les touches fléchées haut et bas ou un joystick, vous pouvez utiliser l'axe "Vertical".

Attention : Assurez-vous que les axes "Horizontal" et "Vertical" s'écrivent bien avec une majuscule au début, car Unity est sensible à la casse.

```
void Update()
{
    float deplacementHorizontal = Input.GetAxis("Horizontal");// Valeur entre -1 (gauche) et 1 (droite)
    personnage.transform.Translate(new Vector3(deplacementHorizontal, 0, 0) * vitesse * Time.deltaTime);

    float deplacementVertical = Input.GetAxis("Vertical");// Valeur entre -1 (bas) et 1 (haut)
    personnage.transform.Translate(new Vector3(0, deplacementVertical, 0) * vitesse * Time.deltaTime);
}
```

```
}
```

1.6 GetButton

Unity permet également de configurer des boutons d'action personnalisés via le gestionnaire d'Input (Edit > Project Settings > Input Manager). Vous pouvez définir des actions comme "Jump" (sauter) ou "Fire" (tirer) et les associer à différentes touches du clavier ou boutons de la manette. Pour détecter si un bouton d'action est enfoncé, vous pouvez utiliser la méthode `Input.GetButton()` et fournir le nom de l'action définie.

```
void Update()
{
    if (Input.GetButton("Jump")) // Vérifie si le bouton "Jump" est enfoncé qui correspond souvent à la touche Espace
    {
        Debug.Log("Le bouton Jump est enfoncé.");
    }
}
```

1.7 Configurer les touches dans l'Input Manager

Il existe un nouveau système de détection des touches appelé "Input System" qui est plus flexible et puissant que l'ancien système "Input Manager". Il permet, entre autres, de gérer plus facilement les entrées provenant de différents périphériques (clavier, souris, manette, écran tactile, etc.).

Cependant, cela est plus complexe à configurer. Pour débuter nous resterons sur l'ancien système "Input Manager" qui est suffisant pour des besoins basiques de détection des touches du clavier.

Cependant, si vous souhaitez explorer le nouveau système "Input System", vous pouvez consulter la documentation officielle de Unity pour plus d'informations sur son installation et sa configuration : [Unity Input System Documentation](#).