

6

Le script de la caméra

Dans ce chapitre, nous allons nous pencher sur notre premier gros script. Nous le placerons sur la caméra et il gèrera le comportement de celle-ci. Ce script permettra à l'utilisateur d'orienter la caméra afin de tourner autour de la balle. Et il lui donnera la possibilité de viser dans la direction où il souhaite propulser la balle.

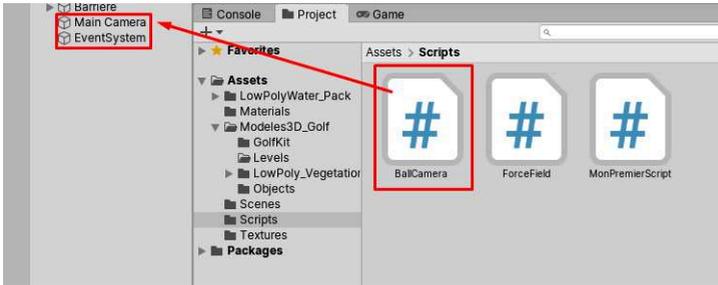
Notre jeu est un jeu mobile. Nous allons donc respecter toutes les contraintes liées au mobile, gérer l'écran tactile et adapter le gameplay à notre smartphone. Cependant, avec à peine d'effort supplémentaire, il nous est possible de le faire fonctionner avec la souris d'un ordinateur. Nous verrons donc comment faire le portage du jeu vers plusieurs plateformes. Cela nous permettra de le tester directement sur notre ordinateur ce qui nous fera gagner énormément de temps.

6.1. Mise en place

Avant de nous lancer dans l'écriture du script, nous devons prendre en compte une petite subtilité. Le jeu devra fonctionner sur un écran tactile. En touchant l'écran, il sera possible d'orienter la caméra. Certaines parties de l'écran seront occupées (bouton de pause, interface utilisateur, barre de puissance). Si l'utilisateur touche l'écran sur une de ces zones, le script de la caméra ne doit pas s'activer. Pour travailler avec les éléments d'interface (boutons, images), nous utilisons un `EventSystem`. C'est ce dernier qui gère notamment le clic. Dans le cas du script de la caméra, l'`EventSystem` nous permettra de savoir si le joueur touche un élément d'interface ou non. Lorsque nous créons une interface utilisateur, l'`EventSystem` est créé automatiquement. Actuellement nous n'avons pas d'interface donc pas d'`EventSystem`. Vous devez donc en créer un manuellement via le menu `GAMEOBJECT > UI > EVENT SYSTEM`. Cela ajoutera un `EventSystem` à la scène. Sans ce `GameObject`, notre script ne pourra pas fonctionner.

Vous pouvez maintenant commencer à coder ! Créez un nouveau script dans le dossier des scripts et appelez-le `BallCamera`. Vous devriez donc avoir votre `EventSystem` et votre script. Glissez ce dernier sur la caméra.

Figure 6.1 : Ajout du script à la caméra



Pour l'instant, ce script est vide, il ne change rien au comportement de la caméra. Nous allons donc l'ouvrir afin d'écrire notre code.

6.2. Les variables

Nous allons commencer par créer nos variables. Nous allons avoir besoin de six variables au total pour le bon fonctionnement de notre script.

Le GameObject de la balle

Il nous faudra stocker le GameObject de la balle car la caméra doit savoir quel objet elle doit regarder :

```
public GameObject balle; // La balle
```

La distance caméra-balle

Nous allons définir la distance de la caméra par rapport à la balle :

```
public float camDistance = 3.5f; // Distance de la cam à la balle
```

Ces deux variables sont publiques car nous souhaitons pouvoir les ajuster dans l'inspecteur. Les variables suivantes seront privées car nous n'aurons pas besoin d'y avoir accès par la suite.

La rotation

Il nous faudra stocker les angles X et Y de la caméra pour gérer sa rotation :

```
float x = 0.0f;
float y = 0.0f;
```

Et nous aurons besoin d'une référence vers cette rotation pour pouvoir agir dessus :

```
Quaternion rotation;
```

Le type `Quaternion` permet de stocker les informations spécifiques à la rotation d'un objet.

La position du doigt

Enfin, comme notre jeu est avant tout un jeu à destination des mobiles (ou tablettes), nous allons avoir besoin de stocker la position du doigt de l'utilisateur sur l'écran afin qu'il puisse agir sur la rotation :

```
// Référence doigt si on touche l'écran tactile
Touch touch1;
```

Nous avons tout ce qu'il nous faut pour la suite de ce script. La dernière étape sera d'initialiser certaines variables dans la fonction `Start`. Au lancement de notre jeu, la caméra a une rotation par défaut (selon la manière dont vous l'avez placée). Notre script agira sur sa rotation. Nous avons donc besoin de récupérer la rotation par défaut afin de l'utiliser comme référence. Cela nous permettra de conserver la vue par défaut et de proposer le meilleur angle possible au lancement du niveau. Pour accéder aux angles de la caméra, nous utilisons `transform.eulerAngles`.

```
void Start()
{
    x = transform.eulerAngles.y;
    y = transform.eulerAngles.x;
}
```

Ce sont les seules variables que nous avons besoin de renseigner dans notre fonction `Start`. Vous pouvez passer au code principal.