

# 21

## Collisions et gestion de la vie

---

Dans ce chapitre, nous allons voir comment détecter les collisions par le code. Cela nous permettra de savoir si une balle touche le joueur, s'il tombe dans un piège, s'il ramasse un objet ou s'il tire sur un ennemi. Nous allons programmer tout cela ici.

### 21.1. Perdre des points de vie

La première chose à faire c'est créer un script de collision qui sera placé sur le joueur. Créez donc un script `PlayerCollision` et placez-le sur le `FPSController`. Ouvrez ensuite ce script. Vous pouvez supprimer les fonctions `Start` et `Update`. Ce script détectera si un ennemi a touché le joueur et lui fera perdre de la vie. Perdre de la vie signifie également mettre à jour l'interface utilisateur. Pour pouvoir utiliser des fonctionnalités du système UI de Unity, nous devons le déclarer en préliminaire au début du programme à l'aide du mot clé `using` : `using UnityEngine.UI;`. Au niveau des variables, il nous faudra une variable de type `Text` qui stockera le texte affichant la vie : `public Text healthTxt;`. Notez que sans le `using` précédent, il nous serait impossible d'utiliser le type `Text`. Nous avons également besoin d'une variable de type `int` qui stockera la valeur de la vie : `public int health = 100;`.

Nous pouvons maintenant coder la détection de collision. Au chapitre [précédent](#), nous avons créé un [projectile pour tourelles](#). Ce projectile a un `Collider` et un `Tag`. Pour ma part, ce tag est `"Tourelle"`. Nous voulons donc détecter la collision avec un objet tagué `Tourelle`. Pour détecter une collision, Unity propose la fonction `OnCollisionEnter` qui prend en paramètre une collision :

```
void OnCollisionEnter(Collision collision)
{
}
```

Dans cette fonction, nous pouvons écrire une condition afin de tester si le tag de la collision correspond à celui que nous souhaitons tester :

```
// Détecter les collisions
void OnCollisionEnter(Collision collision)
```

```

{
  // Si l'objet qui touche le joueur a le Tag Tourelle
  if(collision.gameObject.tag == "Tourelle")
  {
  }
}

```

À l'intérieur de cette condition nous pouvons décrire ce qu'il se passe lorsque le personnage joueur est touché par un projectile. Dans notre exemple, il perdra de la vie, disons 20% : `health = health - 20;`. Puis nous devons mettre à jour le texte affiché à l'écran : `healthTxt.text = health + "%";`. Pensez également à détruire le projectile une fois qu'il a touché le joueur avec `Destroy(collision.gameObject);` car il n'est pas nécessaire de le conserver actif. Voilà le code complet :

```

using UnityEngine;
using UnityEngine.UI;

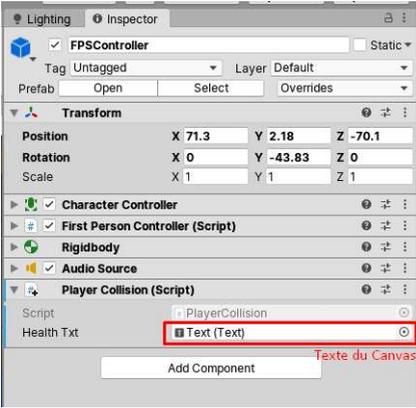
public class PlayerCollision : MonoBehaviour
{
  public Text healthTxt;
  public int health = 100;

  // Détecter les collisions
  void OnCollisionEnter(Collision collision)
  {
    // Si l'objet qui touche le joueur a le Tag Tourelle
    if(collision.gameObject.tag == "Tourelle")
    {
      health = health - 20; // Le joueur perd 20% de vie
      healthTxt.text = health + "%"; // Mise à jour du texte
      Destroy(collision.gameObject); // Détruire le projectile
    }
  }
}

```

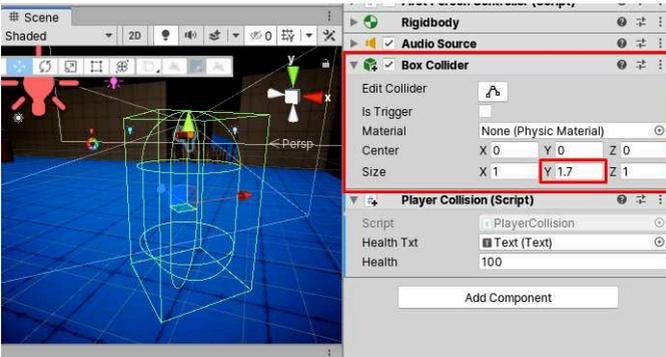
Avant de tester votre programme, renseignez la variable du script :

Figure 21.1 : Configuration du script de collision



Puis pensez à ajouter un BoxCollider à votre personnage afin qu'il dispose d'une surface de collision (le CHARACTERCONTROLLER ne fonctionnerait pas ici).

Figure 21.2 : Ajout du Collider au personnage



Lancez ensuite votre jeu et faites vous tirer dessus par une tourelle.