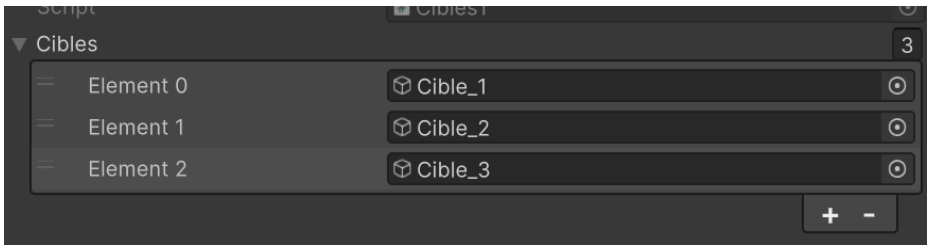
🎮 Projet Unity AR — Angry Bird AR **(Partie 2)**

🎯 Objectif

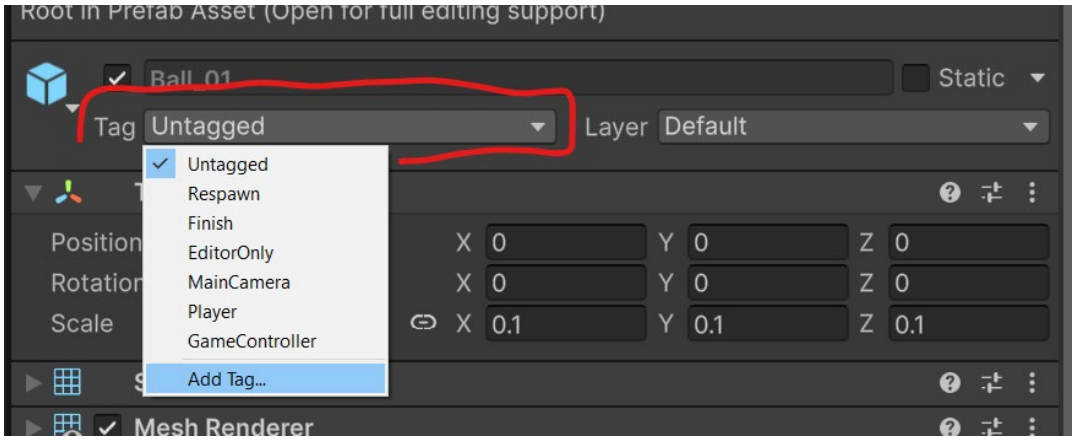
Faire une application en réalité augmentée, on place une pile d'objets et on lance des balles pour la faire tomber. Un score est calculé.

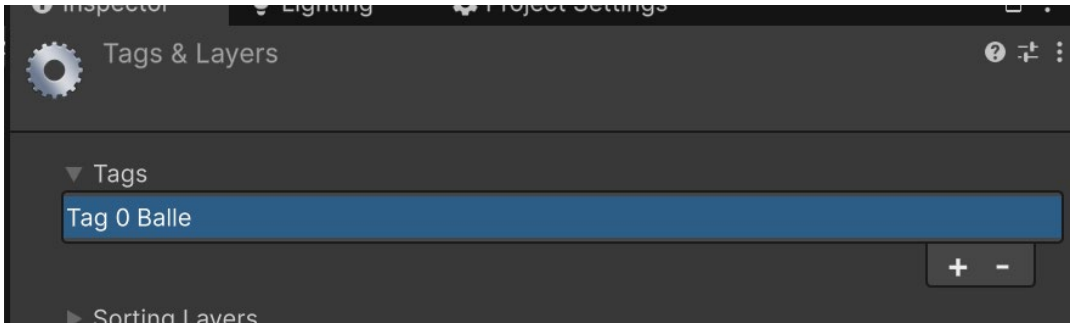
Pour cette partie 2, on se concentre sur la détection des cibles qui sont tombés, pour en faire apparaître de nouvelles.

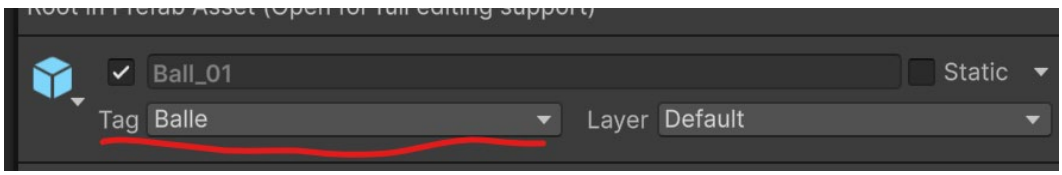
✅ À faire (ordre conseillé)

1. Détection des cibles touchées (Partie 1)
    1. Pour détecter cet évènement, on va devoir vérifier si la balle rentre en collision avec une cible, et si c'est le cas, la valider.
    2. On peut créer un script sur notre prefab de table (sur l'objet vide) qu'on peut appeler « Table » si on veut
    3. On va initialiser une liste de GameObject, pour lister toutes nos cibles
    4.
```
using System.Collections.Generic;
```
    5.
```
public List<GameObject> cibles = new List<GameObject>();
```
    6. On peut retourner dans l'inspector pour assigner nos cibles
    7. 

2. Détection des cibles touchées (Partie 2)
    1. L'étape suivante va nécessiter un autre script, qu'on va rattacher a chacune de nos cibles, et qui va détecter les collisions avec OnCollisionEnter
    2. On crée un script qui peut s'appeler « CibleIndividuelle » et qu'on va mettre sur chaque cible de notre prefab (qui contiennent chacun un RigidBody)
    3. Dans ce script, on va créer une fonction OnCollisionEnter

4.
```
private void OnCollisionEnter(Collision other)
{
    print(message: "Touché");
}
```
5. Le problème, c'est que actuellement, peut importe l'objet qui rentre en collision avec la cible, lance la fonction
6. C'est-à-dire que si l'objet touche la table (ce qui est le cas) cela va lancer et c'est problématique si on veut compter des points
7. Du coup, il faudrait créer un Tag dans Unity, qu'on va appeler par exemple « Balle » et on va l'assigner a notre prefab balle
8. 

9. 

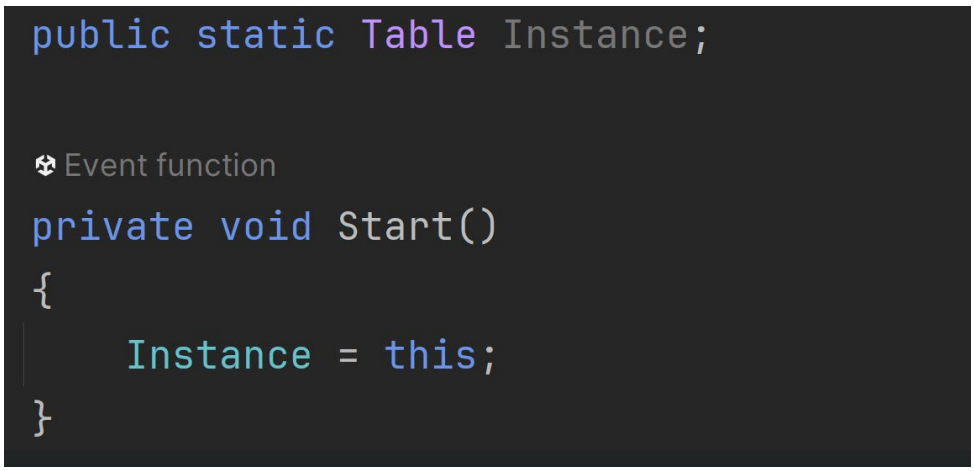
10. Une fois le tag créée, on retourne dans notre prefab et on l'assigne dans le dropdown Tag
11. 

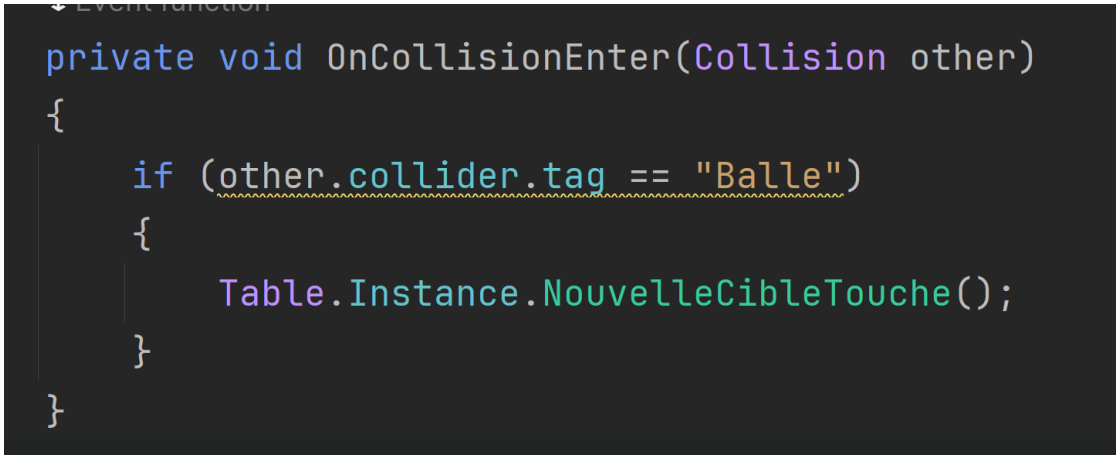
12. Donc maintenant, notre balle à un Tag, on peut mettre une condition dans OnCollisionEnter pour exécuter le code seulement si l'objet « other » possède ce tag

13.
```
private void OnCollisionEnter(Collision other)
{
    if (other.collider.tag == "Balle")
    {
        print( message: "Touché");
    }
}
```

3. A partir de là, on sait quand nos cibles sont touchées par la balle, mais on est censé vérifier si toutes les cibles ont été touchées afin de faire apparaitre une nouvelle table.
    1. On va revenir sur notre script « Table » (qui est censé être ajouté au prefab table, objet parent)
    2. On va créer une variable de type « int » qui va nous servir de compteur pour savoir si toutes les cibles ont été touchés
    3.
```
public List<GameObject> cibles = new List<GameObject>();  Serializable
public int ciblesTouches = 0;  Unchanged
```
    4. A chaque fois qu'une cible sera touchée par la balle, on va rajouter 1 à ce compteur
    5. Quand le compteur est égal à la taille de la liste, on peut dire que toutes les cibles ont été touchées
    6. On peut donc créer une fonction qu'on peut appeler « NouvelleCibleTouche » dans notre script
    7.
```
public void NouvelleCibleTouche()
{
    ciblesTouches += 1;

    if (ciblesTouches >= cibles.Count)
    {
        //Ici ça veut dire que notre compteur
        // a atteint la taille de la liste
    }
}
```
4. Relier le OnCollisionEnter des cibles individuelle a cette fonction

1. Pour appeler cette fonction depuis OnCollisionEnter, on peut utiliser l'architecture Singleton
2. Elle permet de rendre un script global, pour pouvoir appeler ses fonctions de n'importe où
3. Pour ce faire, on peut écrire ça dans notre script de la table
4. 

5. Ma variable static est de type Table, elle doit correspondre au nom de ma classe ! Donc si vous avez appelé votre script autrement changez le
6. La void Start est nécessaire pour assigner l'instance au démarrage du jeu
7. Maintenant, pour appeler notre fonction depuis n'importe quel script, on peut faire :
8. 

9. Je vous laisse faire une condition pour éviter de lancer plusieurs fois cette fonction une fois que la balle a été touchée une fois
10. Indice : vous pouvez utiliser une variable « bool » et une condition dans le OnCollisionEnter…

5. A partir de la, vous devez vérifier si dans le code de la table, on rentre bien dans la condition de victoire une fois qu'on a touché toutes les cibles

1. Vous pouvez faire un print pour vérifier que une fois toutes les cibles touchées, ça affiche bien
2. Il va falloir maintenant trouver comment supprimer la table et en mettre une nouvelle, a la même position…
3. Comme ça dès que le joueur a tout fait tomber, on lui replace une table jusqu’à ce qu’on décide de quand terminer la partie