

🎮 Projet Unity AR — Angry Bird AR (Partie 4)

🎯 Objectif

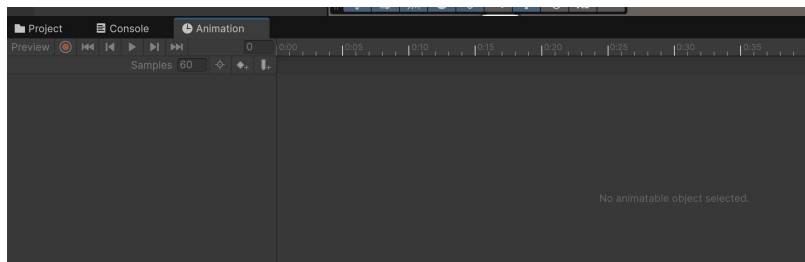
Faire une application en réalité augmentée, on place une pile d'objets et on lance des balles pour la faire tomber. Un score est calculé.

Dans cette quatrième partie, vous allez faire une cible qui bouge, difficile à toucher.

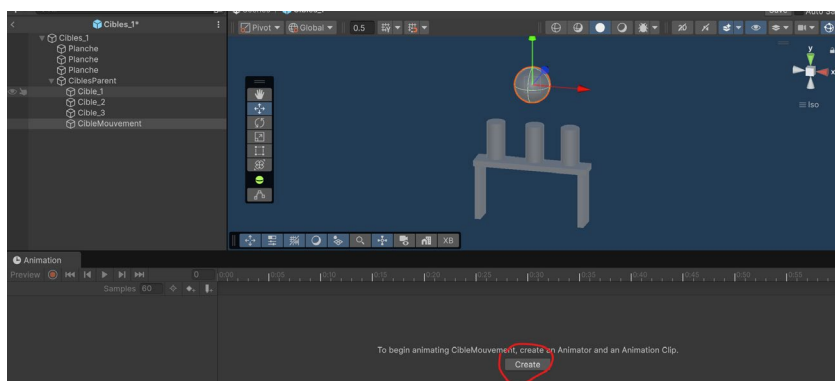
On va voir la notion d'animation d'objets avec deux variantes. Sois vous décidez d'utiliser la fenêtre d'animation d'Unity et d'enregistrer une animation fixe, soit vous pouvez coder votre propre animation seulement par code. A vous de voir.

✅ Animation traditionnelle (ceux qui veulent tester d'animer avec du code passez à l'étape d'après pour avoir quelques indices).

1. Pour animer dans unity, commencez par ouvrir la fenêtre d'animation dans Window > Animation > Animation
2. Vous pouvez déplacer la fenetre et l'ancrer en bas, à côté de la console et de la vue Project

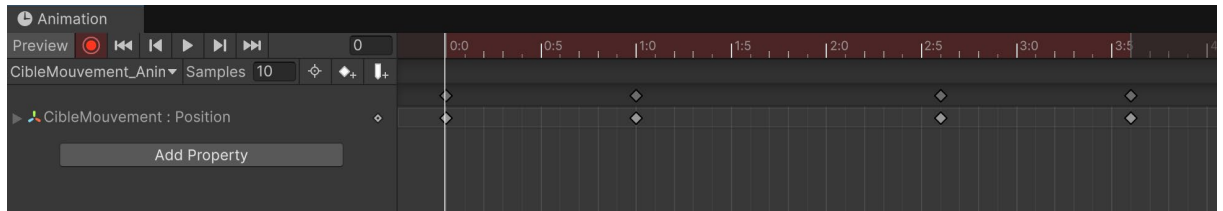


- 3.
4. L'idée c'est de créer une autre cible dans votre prefab Table, qui ne réagira pas a la gravité et sera animé. Créez un objet dans le prefab et sélectionnez le.
5. La cible n'aura pas besoin de Rigidbody à priori
6. Ici j'ai fait une sphere, et en la sélectionnant dans la hiérarchie, la fenêtre Animation me propose de créer une anim

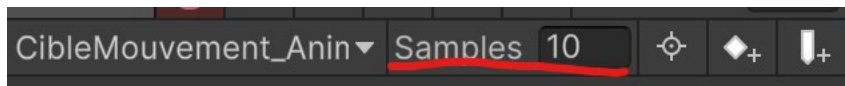


7.

8. Je clic sur Create et j'enregistre mon animation quelque part dans les assets
9. Maintenant, si j'appui sur le bouton rouge d'enregistrement et que je bouge mon objet, puis déplace le curseur de temps, je peux enregistrer plusieurs positions



- 10.
11. Voir le gif animé dans Whaller
12. Le paramètre Samples permet de régler la vitesse de l'animation, plus il est bas plus elle est lente



- 13.
14. Utilisez le bouton lecture pour prévisualiser votre animation



- 15.
16. Essayez de faire boucler l'animation en dupliquant les keyframe pour que l'objet revienne à la position initiale
17. Vous pouvez aussi modifier sa taille et ça s'enregistrera, mais également pleins de paramètres sur les composants de l'objet
18. Une fois l'animation finie, vous pouvez quitter le mode enregistrement, lancer le jeu et voir le résultat
19. Adaptez votre code pour que celle nouvelle cible soit comptabilisée dans le calcul des scores (vous pouvez la désactiver quand elle est touché ou quelque chose du genre)

✅ Animation par code (pour les vrais.es programmeurs. euses)

1. Créer un nouveau composant qui servira à faire bouger un objet
2. Si vous voulez faire bouger l'objet sur différentes positions, vous pouvez avoir une liste de transform (objets empty) qui représenteront les différentes destinations de votre objet
3. Vous pouvez définir une nouvelle variable qu'on va appeler Cible par exemple

4. En utilisant cette fonction, qui va progressivement vers une cible, on peut faire bouger notre objet vers un autre

```
public float vitesse; * Unchanged
public Transform cible; * Unchanged

Event function
private void Update()
{
    transform.position = Vector3.Lerp(transform.position, cible.position, t:Time.deltaTime * vitesse);
}
```

- 5.
6. Vector3.Lerp possède 3 arguments (position actuelle, position ciblé, vitesse)
7. Faites un calcul de distance pour savoir quand vous êtes suffisamment proche de la cible

```
float distance = Vector3.Distance(transform.position, cible.position);|
```

- 8.
9. Si c'est le cas, passer a la prochaine cible dans la liste
10. Si il n'y a plus de cible disponible, vous pouvez tenter de lire la liste à l'envers pour revenir au point initial...
11. Adaptez votre code pour que celle nouvelle cible soit comptabilisée dans le calcul des scores (vous pouvez la désactiver quand elle est touchée ou quelque chose du genre)

✓ Sauvegarde du meilleur score et affichage dans le menu

1. Pour sauvegarder des valeurs, Unity propose Players Prefs
2. Vous pouvez enregistrer une valeur, en lui donnant un nom, puis la retrouver même si vous avez quitté le jeu
3. Pour enregistrer une valeur, comme un score, on peut faire ça :

```
PlayerPrefs.SetInt("BestScore", 100);|
```

- 4.
5. Le nom de la variable sauvegardée est « BestScore » et j'ai donné une valeur en dur pour l'exemple
6. Pour charger cette valeur, il faut faire comme ça :

```
PlayerPrefs.GetInt(key:"BestScore");
```

- 7.
8. Quand le temps restant est écoulé et que la partie se termine, on va vérifier si le joueur a battu le meilleur score
9. On peut charger la valeur de score et la comparer avec l'actuelle
10. Si l'actuelle est plus grande, on sauvegarde cette nouvelle valeur
11. Affichez cette valeur dans le menu principal pour connaître le record à battre

12. Bonus si vous êtes chaud : afficher un petit texte qui dit « nouveau record » dans le menu de fin de partie