Compte Rendu des TP 3 & 4 de Réalité Virtuelle / Réalité Augmentée

M2 IMAGINA

Jérémy DAFFIX

https://github.com/jeremydaffix/hmin320-tp3

TP3

Scène extérieure

Capture d'écran du timer et de son label :

Pour l'amélioration du jeu, j'ai implémenté les 3 "mécanismes" listés dans le sujet du TP :

- Le nombre de points (qui peut être négatif !) associé à une pièce est configurable à partir d'une simple variable int exposée, et est passé comme paramètre au message AddCoin.

- La téléportation vérifie que le score est d'au minimum 3 pour s'effectuer.

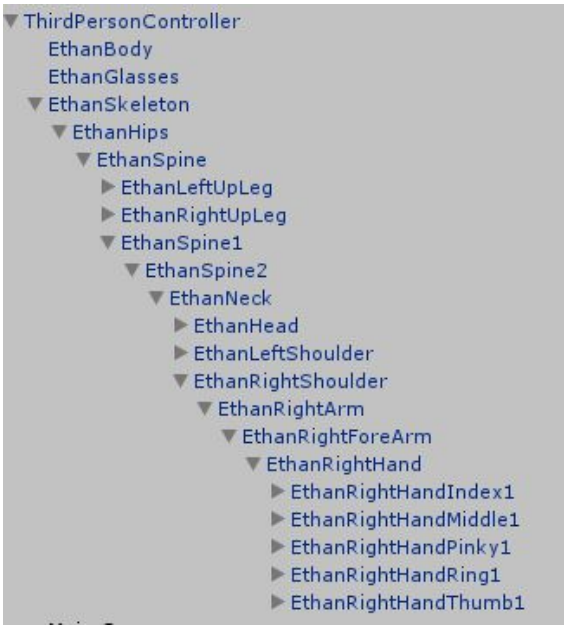
```
void OnTriggerEnter(Collider other)
{
    if(GameVariables.coins > 2) // score minimum avant de voyager !
        SceneManager.LoadScene(levelToLoad) ;
}
```

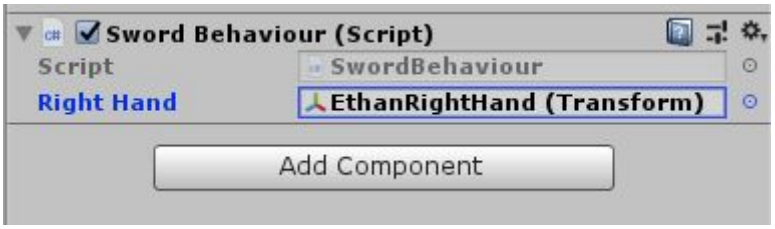
- Lorsque le joueur est au score maximum (il a trouvé les 3 pièces, et a évité celle qui fait baisser le nombre de points), le message affiché à la place du nombre de points est "YOU WIN! :o".

```
public void AddCoin(int nbrPoints)
{
    GameVariables.coins += nbrPoints;

    if (GameVariables.coins < 0) GameVariables.coins = 0;

    if(GameVariables.coins >= 4)
        CoinsText.text = "YOU WIN! :o";
    else
        CoinsText.text = "$" + GameVariables.coins.ToString("00");
}
```

Third Person Character

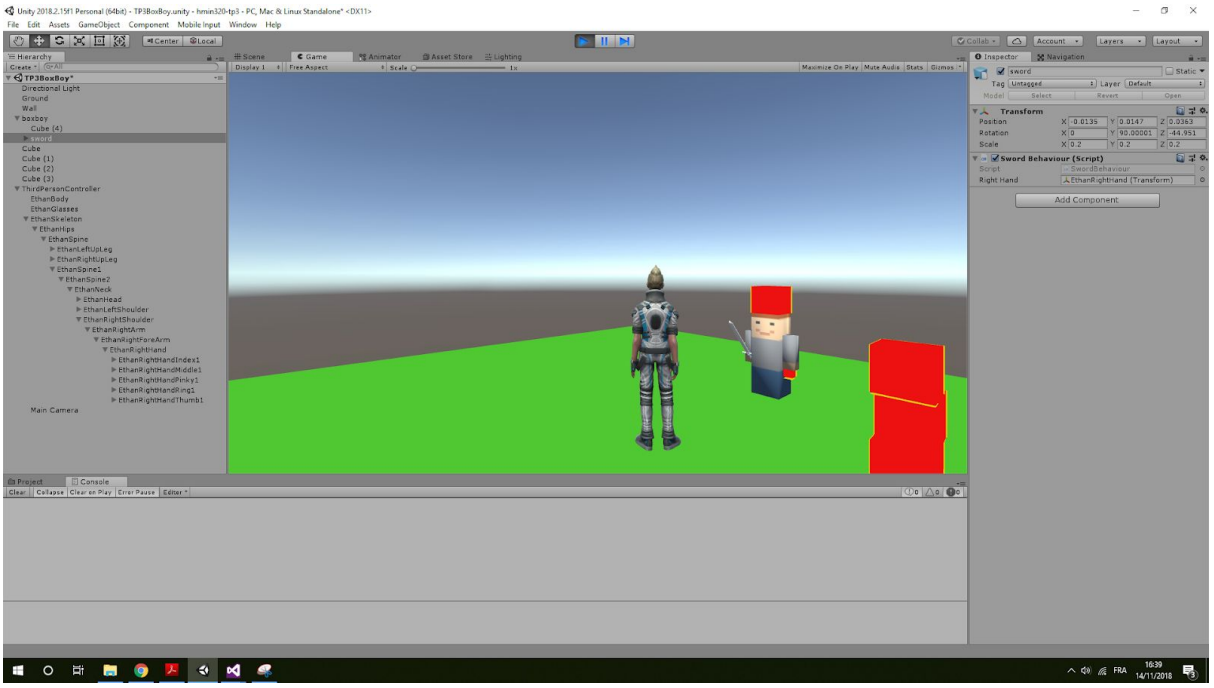
Pour que l'épée suive les mouvements et les animations de la main d'Ethan (quand il court, saute, etc), on va lui mettre comme parent non pas le TPC, mais l'objet *EthanRightHand* :

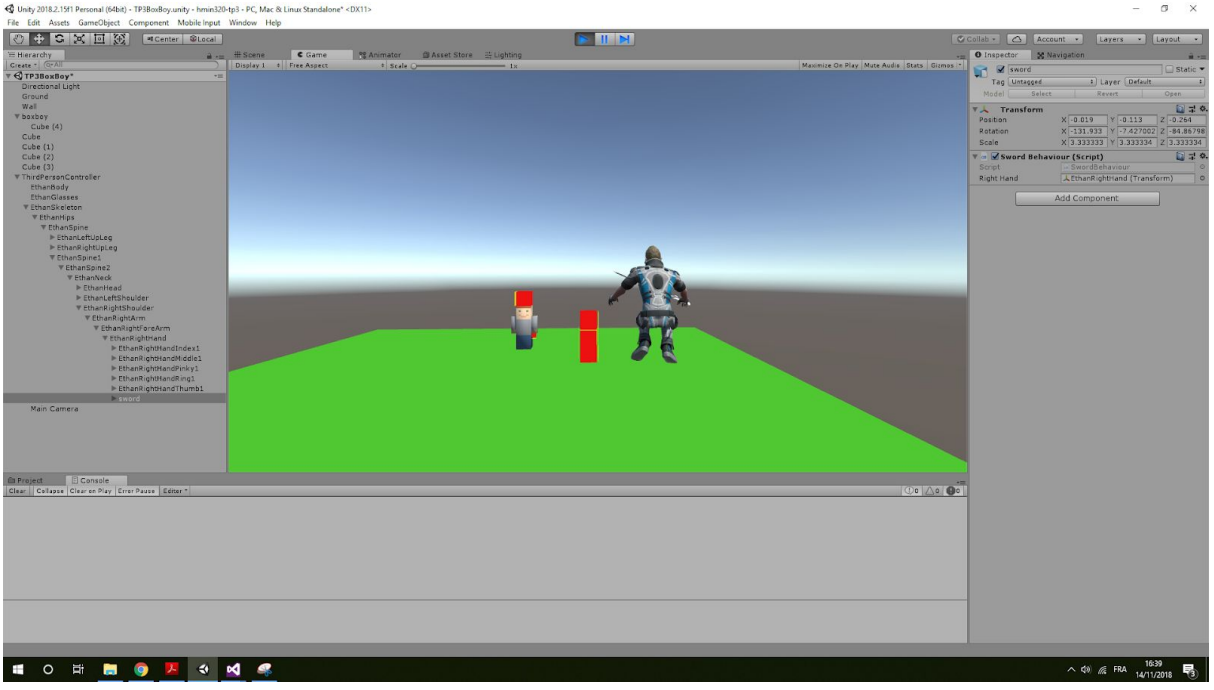





```
public class SwordBehaviour : MonoBehaviour {

    float distance = 10f;
    GameObject player;

    public Transform rightHand;

    // Use this for initialization
    void Start()
    {
        player = GameObject.FindWithTag("Player"); // pour trouver le personnage
    }

    // Update is called once per frame
    void Update()
    {
        distance = Vector3.Distance(this.transform.position, player.transform.position);

        if (distance < 3.5f)
        {
            this.transform.parent = rightHand;
            this.transform.localPosition = new Vector3(-0.019f, -0.113f, -0.264f);
            this.transform.localRotation = Quaternion.Euler(-131.933f, -7.427f, -84.868f);
        }
    }
}
```

Ce qui nous donne le résultat suivant :

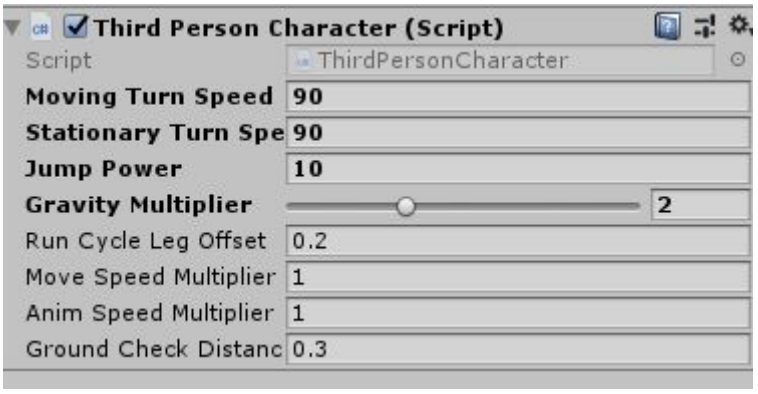




Quand le personnage saute, la main change de position par rapport au corps, mais l'épée reste bien dedans.

Plateformes

Pour que le personnage soit un peu plus maniable, j'ai modifié les paramètres du TPS :

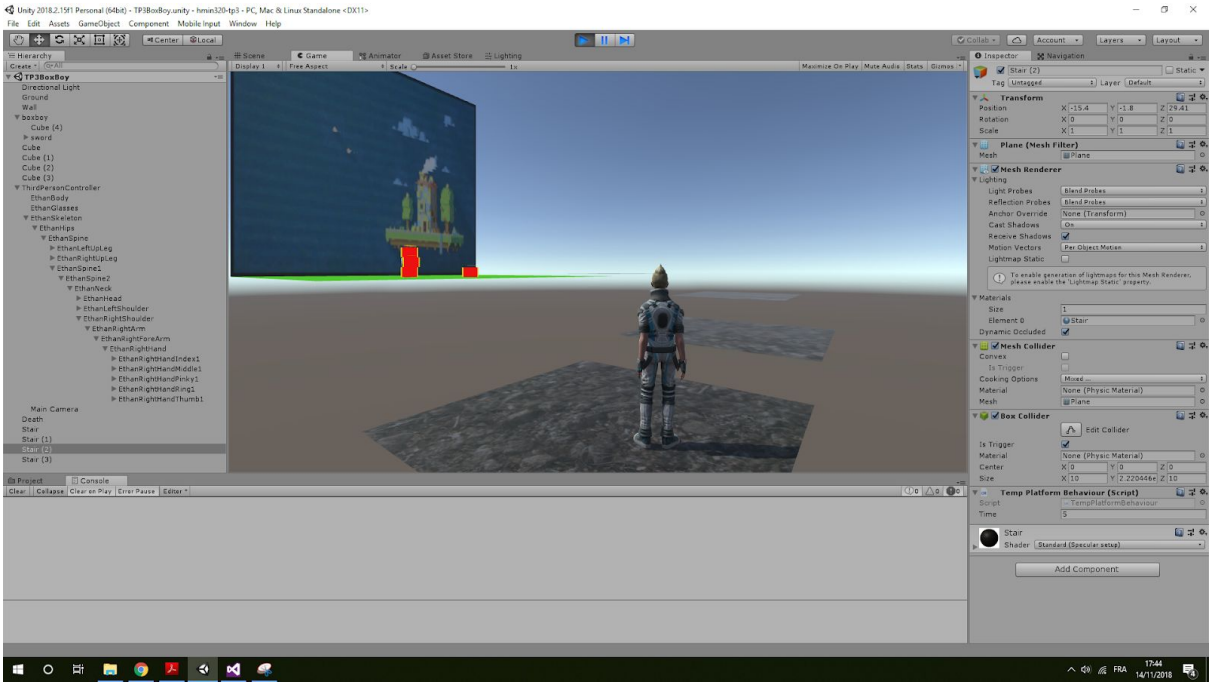


Comme difficulté supplémentaire, j'ai fait en sorte que les 2 escaliers du milieu soient des "plateformes temporaires", c'est à dire qu'elles se détruisent X secondes après que le personnage les ait touchées. Cela oblige le joueur à aller vite, et potentiellement à faire des erreurs mortelles.

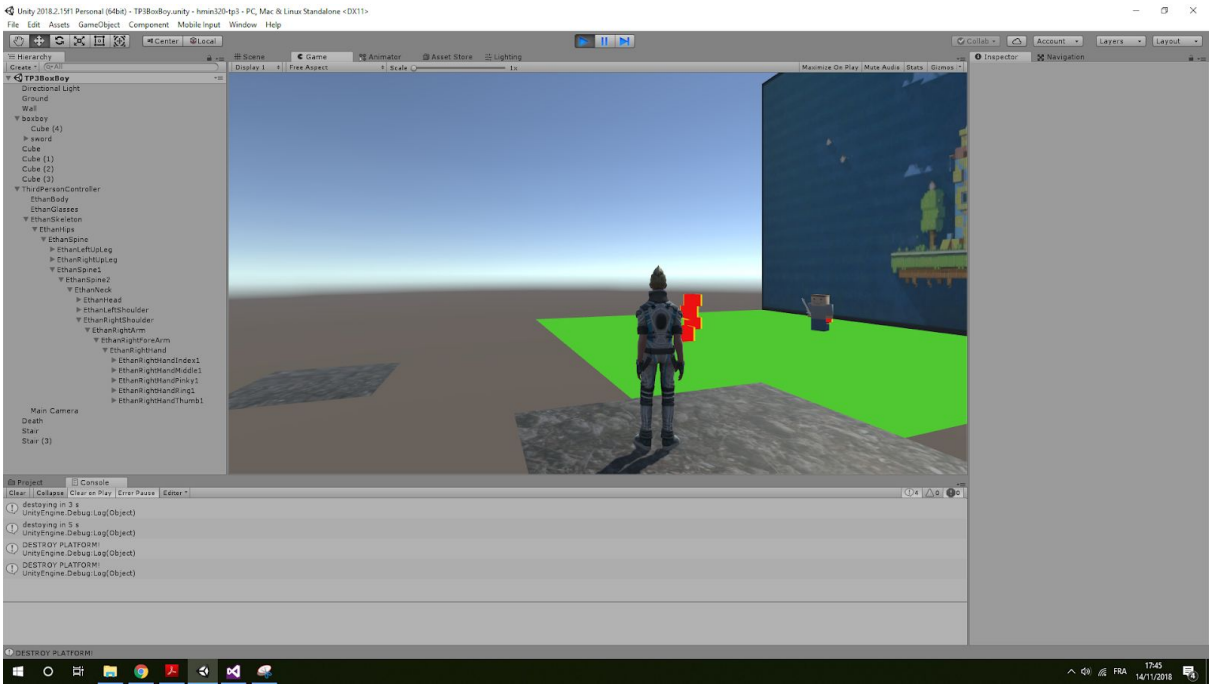
Un script *TempPlatformBehaviour* permet d'implémenter cela. La durée de vie de la plateforme est configurée à partir d'une variable int exposée. Lorsque le trigger est déclenché, le timer se lance.

```
public class TempPlatformBehaviour : MonoBehaviour {

    public int time = 3;

    bool launched = false;


    void OnTriggerEnter(Collider other)
    {
        // le joueur est là : lancement du timer avant autodestruction ! :o
        if (!launched)
        {
            launched = true; // pour éviter que le timer se lance plusieurs fois
            StartCoroutine(TimerTick());
        }
    }


    IEnumerator TimerTick()
    {
        Debug.Log("destoying in " + time + " s");

        while (time > 0)
        {
            //attendre 1 seconde
            yield return new WaitForSeconds(1.0f);
            time--;
        }


        // fin du timer : on s'autodétruit
        Debug.Log("DESTROY PLATFORM!");
        Destroy(gameObject);
    }
}
```

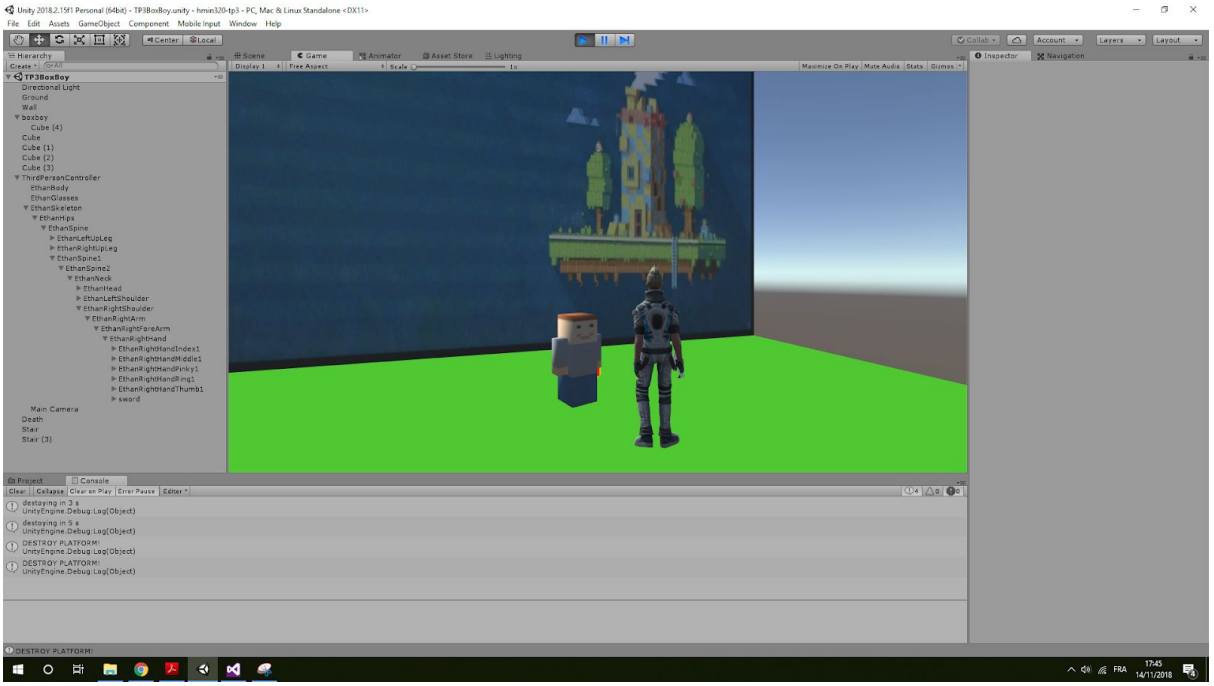
Ce qui nous donne ceci :



Plateforme de départ



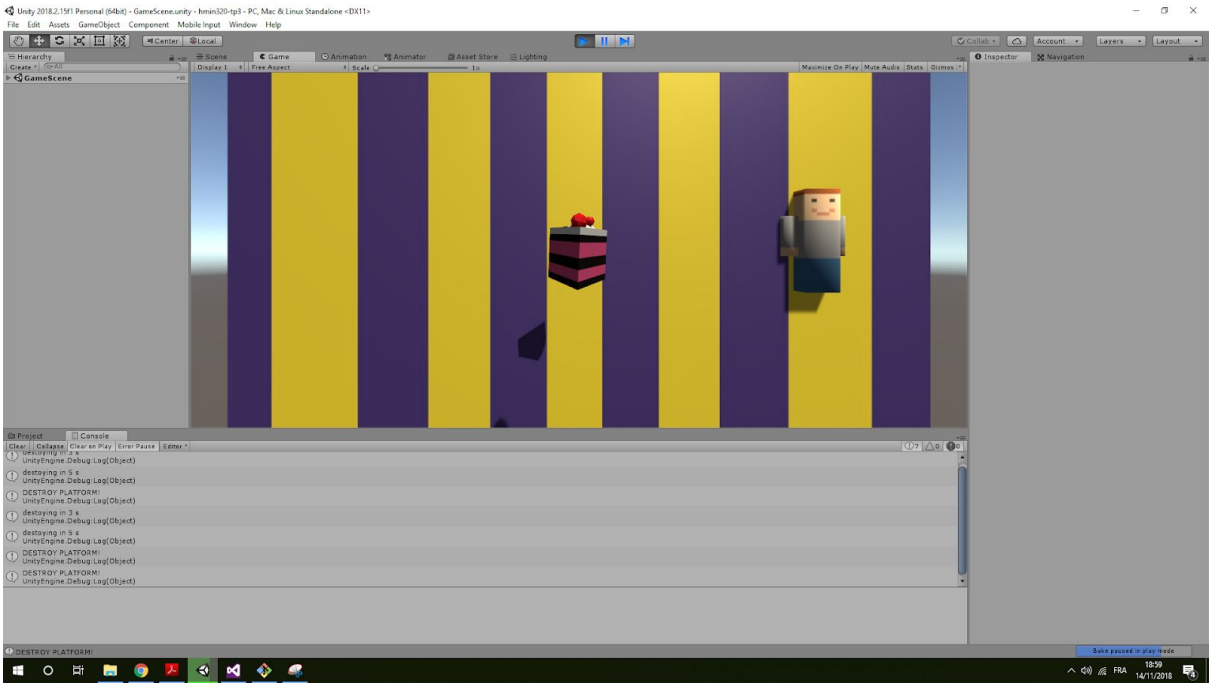
Plateforme d'arrivée, les 2 plateformes précédentes ont été détruites

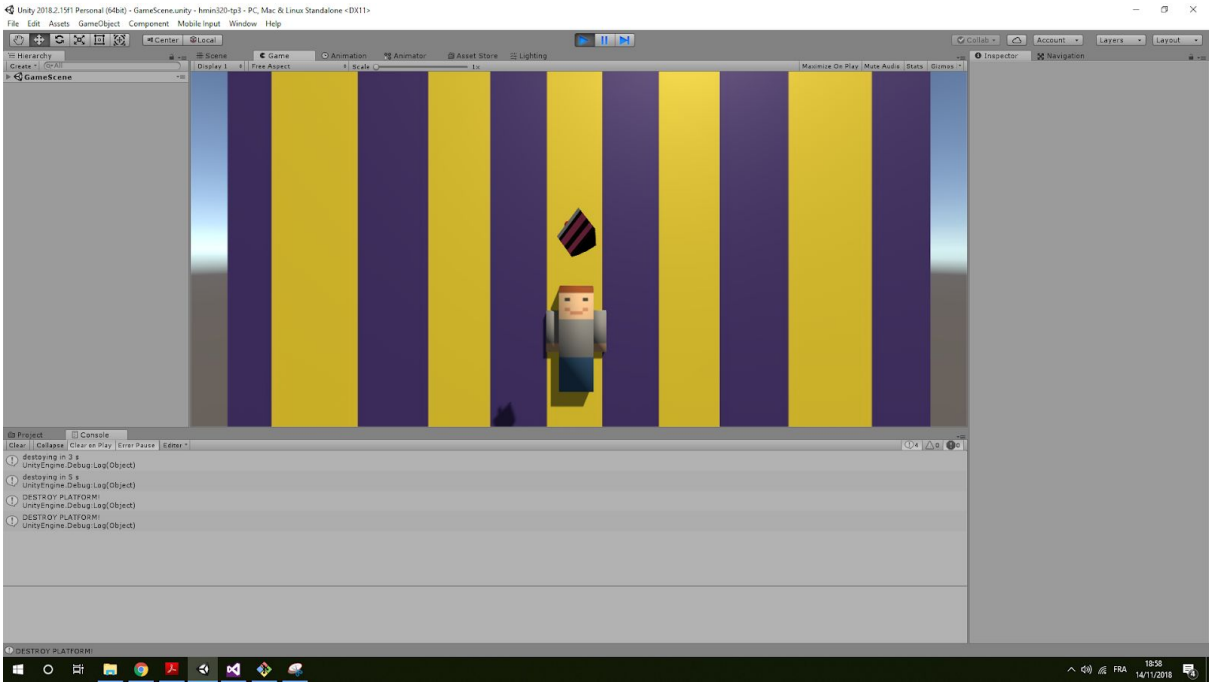


Retour sur la terre ferme !

TP4

Scène Animations





Pour réaliser la transition entre la scène boxboy et la scène des animations, on utilise une coroutine afin de charger la scène des animations 3 secondes après la récupération de l'épée, dans le *SwordBehaviour* :

```
        StartCoroutine(GoShooter());
        
    }
}


IEnumerator GoShooter()
{
    //attendre 3 secondes avant de passer à la scène suivante
    yield return new WaitForSeconds(3.0f);

    SceneManager.LoadScene("GameScene");
}
```