

# Rapport de projet

## *Stickman History*

MOGADE Fabien P1713244

EL BESRI Adam P1810928

AKSAKAL Hikmet P1709541

Sous la soutenance de **Alexandre Meyer**

Pour cette UE nous avons choisi le sujet AM3 : Développement d'un jeu vidéo 2D sous Unity avec Mister RED.

Ce sujet nous a particulièrement intéressé car nous avons déjà plusieurs idées en tête à la suite de la lecture du sujet. De plus, c'était pour nous l'occasion de découvrir un nouvel outil de programmation : Unity

Unity est un moteur de jeu multiplateforme (smartphone, ordinateur, consoles de jeux vidéo et Web) développé par Unity Technologies. Il est l'un des plus répandus dans l'industrie du jeu vidéo, aussi bien pour les grands studios que pour les indépendants du fait de sa rapidité aux prototypages et qu'il permet de sortir les jeux sur tous les supports.

Le logiciel a la particularité d'utiliser du code (C#). Il est basé sur Visual Studio Community.

### **Objectif**

L'objectif de notre projet est de réaliser un Street Fighter Like programmé en 3D mais avec une vue 2D et mettant en scène des stickmans.

## Organisation et séparation du travail

Le logiciel Unity intègre un outil de travail en équipe : Unity Collaborate. Il permet d'avoir un code commun à tous les participants du projet. A chaque modification du code, nous pouvions mettre à jour le code commun du projet et également restaurer une ancienne sauvegarde du projet afin de débayer.

En ce qui concerne la communication, nous avons opté pour un salon Discord dédié au projet où nous nous retrouvions en vocal pendant les horaires de l'UE. L'utilisation du partage d'écran fut très utile. Parfois nous travaillions tous les trois sur un même point pour avancer plus rapidement. D'autres fois nous avons chacun une tâche à programmer individuellement.

## Vue d'ensemble du travail réalisé

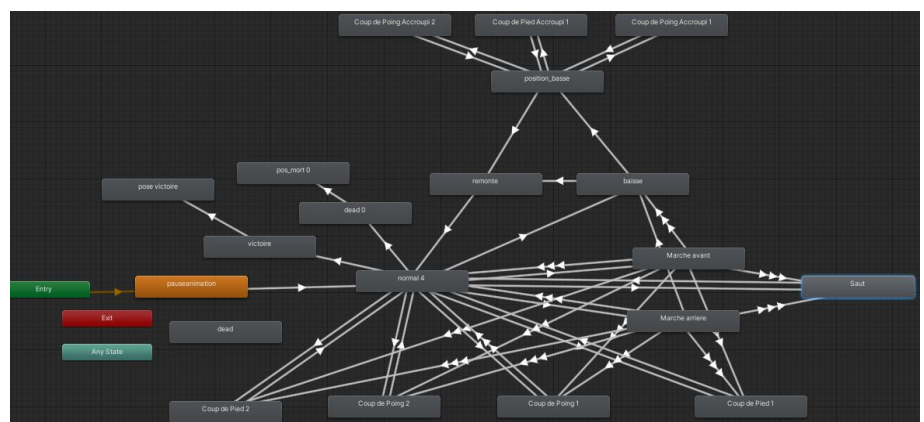
## Déplacement du joueur (récupération des input clavier)

Les déplacements du jeu sont gérés dans le script nommé “testerb.cs”.

Chaque joueur possède un “tag” (joueur1 ou joueur2) qui lui est propre. Le script est séparé en deux parties: une pour les déplacements du joueur 1 et une autre pour ceux du joueur 2. La fonction de saut est également présente dans le script des déplacements.

## Animation

Les animations ont toutes été créées grâce à l'Animator de Unity, sur une courte période (moins d'une seconde). On déplace le personnage ou un membre du personnage et l'Animator s'occupe de le déplacer sur l'intervalle du point de départ jusqu'au point



d'arrivé. Cela crée un mouvement frame par frame.

Toutes les animations sont alors rassemblées au sein de l'Animator et des transitions sont créées entre elles. Il existe plusieurs façon de faire et celle que nous avons décidé d'utiliser est la fonction "anim.SetBool("avance", true);". Ici la fonction met le booléen de transition a vrai qui permet d'entrer dans l'animation où le joueur avance.

Nous avons donc une animation centrale qui nous sert de position d'attente et ensuite plusieurs chemin possibles avec chacun leur transition d'entrée(input du clavier).

### Détection des collisions

```
if(anim.SetBool("pied") || anim.SetBool("pied2") || anim.SetBool("poing") || anim.SetBool("poing2")) {  
    if( (this.name== "MainD" || this.name== "MainG" || this.name== "pied G" ||  
        this.name== "pied D") && (other.name == "Coup" || other.name == "Bas" ))  
    {  
        if (this.name == "MainD" || this.name == "MainG") audioPoing.PlayOneShot(poing, 0.2F);  
        if (this.name == "pied G" || this.name == "pied D") audioPied.PlayOneShot(pied, 0.2F);  
        if(Joueur.tag == "player1")  
        {  
            if(other.name == "Coup") vital.currentHealthJ2=vital.currentHealthJ2-5;  
            else if(other.name == "Bas") vital.currentHealthJ2=vital.currentHealthJ2-2;  
        }  
        if(Joueur.tag == "player2")  
        {  
            if(other.name == "Coup") vital.currentHealthJ1=vital.currentHealthJ1-5;  
            else if(other.name == "Bas") vital.currentHealthJ1=vital.currentHealthJ1-2;  
        }  
    }  
}
```

Le stickman est en fait un ensemble de "GameObject" lié les un au autre. Chaque partie du corps a son propre collider et son propre rigidbody .

Les mains et les pieds de chaque personnage sont équipés du script "collision.cs" ce script appelle une unique fonction nommée " void OnTriggerEnter(Collider other)". Cette fonction est appelée automatiquement lorsque le "GameObjet" auquel elle est liée touche un autre collider. On sait alors facilement quelle partie du corps les mains ou les pieds du personnage ont touchés.

C'est également dans ce script que l'on applique les dégâts en fonction de la zone touchée.

### Création de L'IA

```

if (Mathf.Abs(this.transform.position.x - position_adv.position.x) > 12)
{
    sens = Mathf.Abs(this.transform.position.x - position_adv.position.x) / (this.transform.position.x - position_adv.position.x);
    if (sens == -1)
    {
        anim.SetBool("recule", true);
        HorizontalForce = 15;
        rb.velocity = new Vector3(HorizontalForce, VerticalForce, 0);
    }
    if (sens == 1)
    {
        anim.SetBool("avance", true);
        HorizontalForce = - 15 ;
        rb.velocity = new Vector3(HorizontalForce, VerticalForce, 0);
    }
}
else
{
    HorizontalForce = 0;
    rb.velocity = new Vector3(HorizontalForce, VerticalForce, 0);
    anim.SetBool("recule", false);
    anim.SetBool("avance", false);
    coup = 5;
}

```

La création de l'IA est un des points sur lequel le version 2 de notre projet se porterait en priorité. Pour l'instant l'IA se réside uniquement à avancer le personnage qui possède le script "IA" et à le faire attaquer une fois qu'il est proche de l'adversaire

```

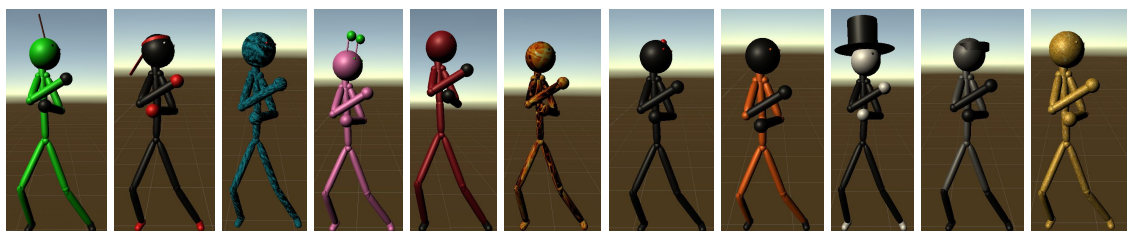
private void frappe()
{
    anim.SetBool("poing", false);
    anim.SetBool("pied", false);
    anim.SetBool("poing2", false);
    anim.SetBool("pied2", false);
    if (coup == 5)
    {
        coup = Random.Range(0, 100);
        if (coup == 1)
        {
            anim.SetBool("poing", true);
        }
        if (coup == 2)
        {
            anim.SetBool("pied", true);
        }
        if (coup == 3)
        {
            anim.SetBool("poing2", true);
        }
        if (coup == 4)
        {
            anim.SetBool("pied2", true);
        }
    }
}

```

## Design du jeu, des niveaux et des événements de fin



l'ensemble de nos menus et nos designs sont des "matériel" intégrant des images et les boutons de chaque scène renvoient vers un bout de script permettant de naviguer entre les scènes



## Caméra qui suit les joueurs:

La caméra se trouve à distance égale entre les deux joueurs et en cas de déplacement d'un des deux joueurs, la caméra se déplace de façon à toujours garder les deux joueurs visible sur l'écran.

## Barre de vies et Timer



Les barres de vie sont représentées par des canvas auxquels on rattache un sprite de barre de vie avec à l'intérieur de lui un élément "Slider".

Depuis le script BarreDeVie nous pouvons mettre à jour la valeur du "Slider" et ainsi baisser la barre de vie visuellement.

Le Time possède un script très simple : On l'initialise à 90 secondes puis dans Update() on soustrait le temps qui s'écoule depuis le lancement du combat à notre timer.

## Ajout d'un Menu principal et Menu pause pendant le jeu

Le menu principal est la première scène du jeu. C'est un panel auquel on a ajouté des éléments qui forment nos boutons cliquables. Avec la fonction "onClick()" et le script "Menu" nous pouvons coder ce qui se passe lorsque l'on clique sur le bouton à l'aide de la souris. Dans notre cas, cliquer sur le bouton "Jouer" charge la scène permettant de sélectionner un mode de jeu.

Au cours du jeu, nous pouvons avec la touche "échap" mettre le jeu en pause en figeant le temps et afficher un menu similaire au menu principal. Nous pouvons alors depuis ce menu pause faire les actions suivantes : Reprendre le jeu, se rendre au menu principal ou quitter le jeu.

## Difficultés

La première difficulté fut de créer un Stickman complexe qui permet de réaliser des animations précises. Après plusieurs stickman différents nous avons fini par construire celui qui nous satisfaisait le plus.

La seconde difficulté était la détection des collisions entre les deux personnages. En effet, au départ nous gérons les déplacements du stickman

grâce à son character controller. Par la suite, nous avons compris qu'il fallait absolument avoir un rigidbody pour les déplacements et des box/sphere Collider pour détecter les collisions.

Naturellement, nous avons fait face à de nombreux bugs freinant notre avancement. Cependant, grâce à la documentation Unity très fournie ainsi que la multitude de tuto présent sur le net nous avons toujours pu continuer d'avancer malgré tout.

### **Bilan**

#### **Bon points :**

- Les animations très précises grâce à l'Animator
- Pouvoir gérer l'affichage en temps réel et construire des objets 3D en un clic de souris.
- Découverte d'un nouvel outil de programmation
- Le très grand nombre de tutoriels Youtube pour apprendre à mieux utiliser Unity
- Unity Collaborate est très pratique pour le travail en groupe et nous a permis travailler ensemble avec facilité

#### **Mauvais Points :**

- Unity est très gourmand
- Difficile à prendre en main au départ
- Nous avons perdu beaucoup de temps sur des points non essentiels ou des mauvaises pistes (CharacterController au lieu de Rigidbody)

### **Piste d'améliorations :**

- **IA** : Développer l'IA pour qu'elle prenne en compte le temps restant et le niveau de sa santé et de celle du joueur. Ajouter plusieurs niveaux de difficultés.

- **Animation** : Faire des animations encore plus réalistes et ajouter des éléments visuels à chaque impact de coup.

- **Inertie** : Améliorer les déplacements au niveau du RigidBody pour faire quelque chose de plus maniable et fun.

- **Attaque** : Rajouter des attaques en plus ( attaques à distances, attaques spéciales, attaques sautées)

- **Défense** : Développer un système de parade pour bloquer les coups et contre attaquer.

- **Barre d'énergie** : Rajouter une barre de Stamina permettant de faire des coups spéciaux plus destructeurs.