**Engins 1, TP2:**

***Cinemachine*, audio,**

***shaders* et effets visuels (*VFX*)**

**Énoncé**

Dans ce second devoir, nous ajouterons des fonctionnalités à notre jeu fait lors du premier TP. Nous ajouterons plusieurs effets visuels et audio, ainsi que deux courtes cinématiques à l’aide de ***cinemachine*** et de ***timeline***.

Certains éléments viendront s’insérer dans notre machine à états qu’il nous faudra maintenir.

Le travail est à effectuer individuellement, mais l’entraide est permise et encouragée.

Ce travail compte pour **25%** de la note finale.

**Éléments du devoir**

**Mise en place d’une machine à états pour le *GameManager***

Nous devrons créer un *GameManager* qui gérera l’état de notre jeu en entier, de la jouabilité aux cinématiques aux transitions de scène.

Celui-ci prendra la forme d’une machine à états, fonctionnant de manière similaire à celle programmée dans le TP1.

Ses états possibles seront:

1. GameplayState
2. CinematicState

Finalement, je demande que le code qui est commun à notre machine à états *main character* et celle *game manager* soit mis dans une classe parent **ou** interface qui gère le code similaire.

**Cinématique, *timeline* et intégration de cinemachine**

Nous allons intégrer l’outil ***cinemachine*** dans notre jeu. Nous allons l’utiliser pour contrôler notre caméra principale, mais également pour créer **une courte cinématique**.

La cinématique servira à introduire votre niveau et devra inclure plusieurs éléments.

* Démonstration des parties de votre niveau, [similaire à cette introduction](https://youtu.be/R6PKzNppulI?t=46)
* Suivre le personnage se déplaçant à sa position initiale dans le niveau
* Montrer le personnage faisant une animation de type “*emote*”

Dans cette cinématique, c’est **une ou plusieurs autres caméras qui représenteront la scène**. Nous verrons comment utiliser plusieurs caméras dans une même scène et comment alterner entre elles.

Il y aura également l’ajout d’un état cinématique dans notre machine à états principale dirigeant le personnage du joueur. Dans cet état, les **inputs** sont simplement inexistants, et c’est une ***timeline*** qui dirigera notre cinématique (autant le personnage que les autres éléments de la scène).

**Système audio**

Un système audio sera intégré à notre jeu. Nous verrons comment faire pour que le son soit joué localement (comme si on entendait ce que la caméra ou ce que le joueur entend) ou globalement (généralement préférable pour la musique, par exemple).

Divers éléments audio sont exigés. Entre autres:

* Son lors des coups infligés ou reçus par le joueur
* Son des bruits de pas, dont l’atténuation dépend de la vitesse du joueur. (Bonus)
* Son du saut et de l’atterrissage du joueur
* Musique

**Effets visuels, *camera shake* et dilatation du temps**

En plus des effets visuels et audio demandés pour la cinématique, quelques éléments doivent être ajoutés dans le mode de jeu principal.

**Réception de dégâts:** Lors d’une réception de dégâts, un ***camera shake*** doit être lancé. L’intensité des dégâts doit affecter l’intensité du ***camera shake***. Un **effet audio** doit être également lancé, ainsi qu’un lancement de ***vfx*** (nous verrons les ***shaders*** pour cette partie)

**Lors d’une attaque du joueur:** Lorsqu’une attaque du joueur connecte avec un agent ennemi, 4 éléments doivent être lancés: un ***camera shake,*** un **effet audio**, un ***vfx*** et finalement une **dilatation du temps.** Nous verrons comment faire chacun de ces éléments en classe.

**Changement de scène**

Finalement, un changement de scène doit être effectué. Celui-ci se fera simplement avec une commande de **debug** (une simple touche). La scène devant être lancée débute par la cinématique d’introduction, puis enchaîne vers un niveau (pouvant être simplement le niveau *sandbox*).

Nous verrons comment conserver des éléments lors d’une transition de scène, et comment effectuer un changement de scène adéquatement.

**Pondération**

Ce travail vaut pour 25% de votre note pour ce cours. La pondération est comme suit:

* **Cinématique:**
  + **Création d’une machine à états *GameManager* fonctionnelle, et refactorisation du code commun à de multiples machines à états en une classe parent. - 20%**
  + **Intégration et utilisation de *Timeline* et de *Cinemachine*. Changement de caméras et des états lors des cinématiques - 10%**
  + **Contrôle via *Timeline* des différents éléments de la scène (caméras, personnages) - 10%**
* **Introduction de sons**
  + **Musique (son global) - 5%**
  + **Bruit lors d’un coup (son local) - 5%**
  + **Bruit du saut et de l’atterrissage du joueur (son local) - 5%**
  + **Qualité de la structure de code régissant ces éléments - 10%**
* **Effets lors d’une réception de dégâts**
  + ***camera shake* - 10%**
  + ***VFX* et *SFX* - 5%**
* **Effets lors d’un coup infligé par le joueur**
  + **Dilatation du temps - 5%**
  + **VFX et SFX - 5%**
* **Changement de scène via une commande de debug - 10%**

* **Je me permets de pouvoir réduire la note en fonction de la qualité du code, noms de variables et de méthodes appropriées, jusqu’à 20%**

**Format de l’évaluation**

Date de remise de l’évaluation mardi 31 octobre. Remettre en ligne sur LÉA le lien vers votre **github.** Nommer la version à évaluer (via un commentaire de ***commit***): **Candidat TP2**.

Assurez-vous que cette version est stable et fonctionnelle, et que le projet soit fait avec **Unity 2022.3.7f1**

**Entraide**

L’entraide est permise et encouragée. Vous pouvez vous entraider sur la logique des scripts et sur l’explication des technologies vues en classe. Par contre, il est interdit de partager son code, de copier d’aucune manière le code de quelqu’un d’autre ou de remettre un travail effectué par quelqu’un d’autre (incluant une IA comme chatgpt).  Il y a une différence très marquée entre l’entraide et le plagiat.

**Ressources disponibles**

Il existe d’excellentes ressources en ligne pour de multiples éléments du TP. ***Cinemachine, Timeline,*** les **effets visuels (*vfx*) et audio (*sfx*)** sont extrêmement biens documentés. N’hésitez pas à approfondir vos connaissances avec des recherches!

De plus, la documentation officielle de Unity est claire et droite au but. Je vous encourage fortement à la consulter.

Pour toutes questions, il me fera plaisir de vous répondre en Mio ou sur la page de votre cohorte sur discord.

**Références**

[Exemple de dilatation du temps](https://youtu.be/CJkH1I2RrT4?t=100)

[Exemple d’introduction de niveau (animation d’introduction)](https://youtu.be/R6PKzNppulI?t=46)

[Exemple de *camera shake* (effectué avec *Cinemachine!*)](https://youtu.be/ACf1I27I6Tk)