Développement de jeux vidéo

Projet final



Développer son propre jeu à partir d’un thème

# Objectifs

Le but est de créer une expérience (jeu ou simulation) complète à votre goût en intégrant les notions apprises dans le cours.

Une thématique devra être intégrée et sera automatiquement générée. La thématique sera affichée sur le site web pour référence. La thématique **DOIT** être intégrée dans votre jeu, mais elle peut l’être légèrement au besoin si vous aviez une idée de jeu très précise. Venez m’en parler si ça vous inquiète.

# Modalités

La remise est au plus tard le **mardi 21 mai avant 23:59**.

Vous devrez remette sur Léa un lien vers votre répertoire GIT.

Envoyez une demande d’accès sur BitBucket ou Github.

* Nom usager : Akilea
* Courriel : [phoenix.lavoie@gmail.com](mailto:phoenix.lavoie@gmail.com)

Votre répertoire doit inclure :

* Projet Unity en entier
  + Ne pas oublier le *.gitignore* sinon le projet sera trop gros.
* Version publiée sur Windows dans un dossier nommé Exec
* À la racine du projet, un fichier *Readme.md* contenant vos noms

# Limites

### Thématique

La vulgarité, la violence extrême ou autres thèmes jugés inapproprié sont à proscrire. Ne craignez tout de même pas de faire des jeux lugubres/effrayants/politiques. Venez m’en parler cas de doute.

### Ressources

Vous pouvez réutiliser quelques ressources utilisées dans le cours tels que images, animations, effets de particules, arrière-plan/script mais pas plus de 4.

# Équipe

Chaque projet se fait en équipe de 2 à 3 personnes. Dans la section suivante (*Exigence*), chaque item doit être multiplié par le nombre de personne. Par exemple :

* [x2] couches d’affichage toutes utilisées.

Signifie

* 4 couches d’affichage pour les équipes de 2.
* 6 couches d’affichage pour les équipes de 3.

# Exigences

Voici le minimum que vous devez effectuer. Bien entendu, il n’y a aucun problème à en faire plus!

### Propriétés générales

* ~~[x2] couches d’affichage toutes utilisées~~
* ~~[x2] masques tous utilisés~~
* ~~[x4]~~ *~~prefabs~~*
* ~~On doit pouvoir terminer le jeu en [x2] minutes~~.

### Scène menu

Il doit avoir…

* ~~Titre~~
* ~~Sous-titre~~ 
  + ~~Style différent du titre~~
* ~~Bouton démarrer~~
  + Doit avoir un fondu au noir avant de changer de scène
  + On doit avoir un fondu lorsqu’on entre dans le jeu
* ~~Bouton quitter~~
* [x1] effet de particule
* ~~1 arrière-plan~~
* ~~[x1] images (animées ou non)~~
* [x1] séquence animée avec une coroutine
  + Ex : joueur qui se fait courir après par un ennemi

### Scène fin de jeu

* ~~Nom du/des créateur(s)~~

~~[x1] images (animées ou non) différentes du menu~~

* ~~[x1] séquence animée avec une coroutine différente du menu~~
* ~~1 arrière-plan~~

### Scènes principales

**~~Au minimum~~** ~~[x1]~~ **~~niveaux avec minimum 2 défis chacun.~~**

Chacun doit avoir…

* ~~1 tuilage unique~~

**Collectivement, ils doivent inclurent :**

#### Visuel

* ~~[x3] objets de jeu issue d’images/~~*~~spritesheet~~* ~~différents~~
  + ~~Maximum 4 images statiques~~
* ~~[x1] effets de particules (explosions)~~
  + ~~Ils doivent être déclenché par code~~
* [x2] effets de particules (continu)
* ~~[x1] indicateur dans une interface~~
  + ~~Ex : indiquer le nombre de vie~~
  + ~~Ex2 : indiquer les munitions~~
  + ~~Ex3 : indiquer le temps~~
* ~~[x1] queue (~~*~~motion trail~~*~~) avec une fin triangulaire et un gradient de couleur~~

#### Mécanique

* ~~[x2] objets de jeu qui sont bougé par code~~
  + ~~Ajuster la position, la vitesse ou utiliser~~ *~~AddForce~~*
* ~~[x1] objet qui peut être poussé par un joueur/ennemi/rayon~~
  + ~~C’est un objet physique dynamique~~
* [x1] matériel physique
* ~~[x2] réponses à une collision (~~*~~OnCollisionEnter2D~~*~~)~~
  + ~~1 doit filtrer avec les masques (~~*~~layers~~*~~)~~
* ~~[x1] réponse à une collision (~~*~~OnTriggerEnter2D~~*~~)~~
* ~~[x1] rayon avec une réponse au touché~~
  + ~~Ex : pour faire un AI~~
  + ~~Ex2 : pour faire une arme~~
* [x2] évènements
  + Ils doivent être invoqués et écoutés

## Évaluation technique

Les points suivants seront évalués

* Cohérence visuelle
* Cohérence design de l’expérience
* Quantité/impact des bugs
* Qualité du code
* Organisation dans Unity