

Enseignant(s)

VIDAL Nicolas

Email(s)

nvidal@myges.fr

Projet Annuel 3A - Extreme Survivor Game Intensity

1 Matières, formations et groupes

Matière liée au projet :

Formations : -

Nombre d'étudiant
par groupe :

3 à 4

Règles de constitution des groupes: **Libre**

Charge de travail
estimée par étudiant : **50,00 h**

2 Sujet(s) du projet

Type de sujet : **Imposé**

3 Détails du projet

Objectif du projet (à la fin du projet les étudiants sauront réaliser un...)

Réaliser un premier Jeu Vidéo de type Survivor, en utilisant au maximum l'ensemble des outils mis à disposition par l'environnement de développement de jeu vidéo Unity avec l'objectif de pouvoir réutiliser la majeure partie des assets développés pour une éventuelle production plus sérieuse (pipeline d'intégration générique, code réutilisable, prefabs génériques, système ECS, ...).

Descriptif détaillé

ESGI : Extreme Survivor Game Intensity

Réaliser un jeu vidéo de type Survivor, avec certaines contraintes technologiques et de Gameplay imposées.

Contraintes

Type de Jeu

Le thème de cette année est orienté Survivor, mais en laissant une liberté d'approche pour permettre aux étudiants de valoriser leur création dans le cadre de leur scolarité, tout en ayant un socle commun permettant une évaluation sensée des différents projets les uns par rapport aux autres.

Innovation

L'originalité sera appréciée, dans le but de ne pas avoir un copycat complet d'un jeu existant.

Il sera attendu de la part du groupe d'étudiant de pouvoir justifier la totalité des choix d'implémentations et d'architecture. Il sera important de réfléchir longuement avant de se lancer dans l'écriture du code.

Gameplay 'Obligatoire'

Le gameplay doit s'adapter aux contraintes suivantes :

- Interactions avec de (très) nombreuses entités
- Interactions "physiques" entre les entités
- Environnement 3D
- Un minimum de 10 types d'ennemis standard devront être proposés
- Un minimum de 5 boss différents devront être proposés
- Les 'attaques' du joueur doivent être au minimum au nombre de 10
- Il doit être possible de faire évoluer les attaques du joueur en partie
- Un minimum de 2 environnements devront être jouables
- Le ramassage de bonus doit être possible dans l'environnement
- On doit pouvoir jouer depuis deux points de vue possible (ex, vue FP et vue de dessus)
- Une minimap doit être visible

Challenges Techniques (1 au choix)

- Multijoueur online (serveur de matchmaking, serveur de jeu, ...)
- Mobile Friendly : le jeu devra pouvoir fonctionner sur un téléphone Android de type Pixel 8 à 60fps constant.
- Replay : le jeu devra proposer au joueur un replay de la partie, sans que cela soit trop gourmand en stockage ou performance (limite de 100Mos par replay).
- Progression hors jeu, déblocage de pouvoir et achievements

Challenges Technologiques (1 au choix)

- Réseaux Sociaux : Le joueur sera incité à des moments opportuns à partager ses exploits sur différents réseaux sociaux (Facebook, X et Instagram), seule l'utilisation des frameworks et plugins officiels sera autorisée ici.
- Stockage en ligne du profil de joueur et ranking : Le joueur devra pouvoir créer un compte utilisateur et stocker ses informations sur un serveur distant, un site web associé devra être développé pour afficher le score ou les exploits des utilisateurs (en plus de pouvoir directement visualiser ceux-ci dans le jeu).
- Stockage en ligne du profil de joueur, et boutique : Le joueur devra pouvoir obtenir une monnaie virtuelle tout en faisant des parties, et cette monnaie lui permettre de débloquent des objets/compétences/... entre les parties. Une interface d'administration (web) permettant d'ajouter des objets devra également être développée pour pouvoir mettre à jour cette boutique sans avoir à mettre en ligne une nouvelle version du jeu.
- Publicités : Le jeu devra intégrer un framework publicitaire, et l'utiliser pour proposer des bannières, des interstitiels et des rewarded videos. Par exemple ces dernières peuvent permettre de débloquent des items ou de commencer le jeu avec de meilleures statistiques.

Features hors gameplay obligatoires

- Menu, options, etc. Le jeu devra contenir au minimum :
 - o Un Ecran titre
 - o Un écran de lobby : Espace de configuration de la partie (choix de la carte, ...)
 - o Un écran d'options permettant :
 - De désactiver les effets sonores et/ou la musique (ou régler différents volumes sonores)
 - De changer la langue de manière dynamique

- De régler la qualité visuelle du jeu
 - o Menu InGame permettant de mettre le jeu en pause, et de quitter le jeu en revenant au menu principal
 - o Un écran de fin de partie résumant les statistiques et résultats de la partie

- Localisation

- o Le jeu devra être disponible en Anglais et Français, changer de langue devra être possible sans avoir à redémarrer le jeu depuis le menu d'options.

- Robustesse

- o Mise en pause in-game
- o Indépendance à la résolution d'écran
- o Indépendance au format d'écran

- Performances

- o 60fps constant sur la plateforme cible

Pour permettre au jury de pouvoir évaluer correctement le jeu produit il sera nécessaire de montrer en peu de temps le déroulé d'une partie. Trois options s'offrent à vous :

- Vous baser sur un format court de partie (5-8 minutes)
- Proposer un mécanisme pour avancer dans la partie plus rapidement lors de la soutenance (sauvegarde/chargement in-game)
- Cheat codes

Plateformes Cibles

La plateforme cible : Windows (ou Android si challenge Technique choisi)

Programmation d'outils

En dehors du rendu du jeu proprement dit, il sera demandé le développement d'un minimum de 2 outils (pouvant prendre la forme de scripts/plugins pour Unity ou de programmes complètement indépendants) ayant pour but de faciliter la conception du jeu.

Ex : génération automatique de niveau, génération de scènes de test, visualisation de l'équilibre des unités ou factions si plusieurs, etc.

Outils

Moteur de Jeu

L'utilisation d'Unity comme moteur/environnement de développement est requise.

Scripting

IDE

L'utilisation de JetBrains Rider ou Microsoft Visual Studio ou Monodevelop est vivement conseillée, de même que l'utilisation du langage C# pour le scripting, il sera également possible d'avoir recours à l'utilisation de plugins natifs dans le but d'obtenir de meilleures performances.

Coding Style

Tous les scripts développés devront respecter un coding style identique. Il est envisageable de reprendre des coding styles classiques et existants tels que :

- http://www.codeproject.com/KB/cs/c__coding_standards.aspx
- <http://csharpguidelines.codeplex.com/releases/view/46280>

Dans tous les cas, le code produit devra être propre (notamment exempt de traces de debug), et commenté.

Versionning

L'utilisation de Git SCM est obligatoire. De même, il est vivement conseillé de produire régulièrement des archives de l'intégralité du projet en cours et de dupliquer les supports de sauvegardes dans le but d'éviter les traditionnelles pannes de disques durs au moment des soutenances.

Sources

L'intégralité des sources devra être accessibles au moment du rendu final et à chaque jalon dans le but de pouvoir contrôler la bonne application des contraintes présentes dans ce document.

Valorisation de l'enseignement

L'intégralité des enseignements ayant trait à la production d'un jeu vidéo devra être mis en avant dans ce projet selon leur applicabilité lors des différents jalons (GameDesign, Modélisation3D, Gestion de projet, ...).

Par exemple, les étudiants seront tenus de faire figurer leurs productions 3D issues de leur cours de modélisation/animation dans le produit multimédia résultant.

Authoring

Dans le but de pouvoir clairement identifier le travail de chacun au sein du groupe de projet, tout fichier produit devra porter soit dans son nom, soit dans son entête, le nom de son auteur ainsi que son numéro de version. Si cela se révèle être problématique pour le nommage de certains assets, il est aussi envisageable de créer un fichier texte récapitulatif, décrivant clairement pour chaque fichier produit la contribution de chacun.

Ouvrages de référence (livres, articles, revues, sites web...)

<http://unity3d.com/support/documentation/>
<https://www.redblobgames.com/>
<https://github.com/Unity-Technologies/ml-agents>
<https://docs.unity3d.com/Packages/com.unity.entities@1.0/manual/index.html>
<https://learn.unity.com/course/dots-best-practices>

Références Vidéoludiques

https://store.steampowered.com/app/1794680/Vampire_Survivors/
<https://play.google.com/store/apps/details?id=com.quicksand.pocketnecromancer&hl=fr>
https://dow.fandom.com/wiki/The_Last_Stand (https://www.youtube.com/watch?v=GaJliQbASfU&list=PLNyZLB5zXxQvf-qwfoq_UvHQ6UYMoEBFj&index=1)
<https://store.steampowered.com/app/1942280/Brotato/>
https://store.steampowered.com/app/2067920/Rogue_Genesia/
https://store.steampowered.com/app/2068280/Nordic_Ashes_Survivors_of_Ragnarok/
https://store.steampowered.com/app/2066020/Soulstone_Survivors/

Outils informatiques à installer

Unity(développement) <http://unity3d.com/unity/download> , Suite Office (présentation et document de suivi de projet), Git SCM (source control and versioning) : <http://git-scm.com/>

Rendu

Chaque rendu sera effectué via le pull de la branche master du dépôt Git par l'enseignant. Tout retard et/ou problème sera sanctionné par une perte de points.

Une archive Zip des builds du jeu et des sources correspondant à chaque jalon sera également uploadée sur MyGES.

Réutilisation de code / bibliothèque existant

A priori, aucune utilisation de bibliothèque existante en dehors de l'API de Unity, des packages officiels d'Unity et du SDK de Mono ne sera autorisée à l'exception des frameworks suivants : Zenject, UniRx, DoTween, UnityMLAgents, Cinemachine.

La réutilisation de code existant ne sera tolérée que dans les cas suivants :

- Fonctionnalités d'import/export d'assets aux formats exotiques et non supportés tels quels dans Unity.
- Utilisation de shaders particuliers pour un effet artistique nécessaire au jeu proposé.
- Utilisation de camera effects particuliers pour un effet artistique nécessaire au jeu proposé.
- Plugins officiels des plateformes pour le partage sur les réseaux sociaux

4 Livrables et étapes de suivi

1	Etape intermédiaire	Présentation du concept - Présentation en slides devant la classe du concept de jeu proposé - Gameplay général - Proposition de planning de développement de features - Validation	dimanche 19/01/2025 23h59
---	---------------------	--	---------------------------------

2	Etape intermédiaire	<p>PROTOTYPE</p> <p>Version comprenant la boucle de gameplay principale du projet</p> <p>Livrables :</p> <ul style="list-style-type: none"> Exécutable PC/Windows Sources (répertoire du projet Unity) Bref rapport présentant la TODO list 	jeudi 06/03/2025 23h59
3	Etape intermédiaire	<p>ALPHA</p> <p>Version comprenant la majorité des éléments de Gameplay du produit multimédia.</p> <p>Livrables :</p> <ul style="list-style-type: none"> Exécutable PC/Windows Sources (répertoire du projet Unity) Bref rapport présentant la TODO list 	jeudi 01/05/2025 23h59
4	Etape intermédiaire	<p>BETA</p> <p>Version quasi-complète, incluant l'intégralité des assets, pouvant posséder quelques bugs mineurs.</p> <p>Livrables :</p> <ul style="list-style-type: none"> Exécutable PC/Windows Sources (répertoire du projet Unity) Bref rapport présentant la TODO list 	jeudi 29/05/2025 23h59
5	Etape intermédiaire	<p>Release Candidate</p> <p>Rendu Packagé, Version du jeu Bug Free, avec installer.</p> <p>Bonus : site web vitrine, notice, jaquette, etc.</p> <p>Livrables :</p> <ul style="list-style-type: none"> Exécutable PC/Windows Sources (répertoire du projet Unity) Bref rapport présentant l'évolution du projet initial par rapport au produit réalisé 	mercredi 09/07/2025 23h59
6	Rendu final	<p>Soutenance</p> <p>Présentation 'commerciale' du jeu devant un public ouvert et hétérogène.</p> <p>Livrables :</p> <ul style="list-style-type: none"> Présentation du projet (Powerpoint) Démonstration pertinente et ciblée 	jeudi 17/07/2025 23h59

5	Soutenance	
Durée de présentation par groupe :	30 min	Audience : Publique
Type de présentation :	Présentation / PowerPoint - Démonstration	
Précisions :		