

# Survival And Monsters

Ernest Niederman      Francisco Ernesto Suarez Roca  
Nicolas-Thomas Marie-Sainte      Lohan Morvan

March 2024

## Table des matières

1	Introduction	3
2	Description générale du projet	3
3	Exigences	4
4	Documentation	5
5	Contraintes	5
6	Organisation du développement	6
7	Diagramme de Gantt	6
8	Validation et tests	6
9	Ressources	7
10	Sitographie	7

# 1 Introduction

Inspiré du célèbre jeu Vampire Survivor, Survival And Monsters est un jeu de survie de type Rogue-Lite dans lequel vous êtes plongés dans un univers hostile. Dans le jeu le joueur ne peut que se déplacer et attaquer, certaines attaques sont automatiques. Pour garantir votre survie vous devez de subir le moins de dégâts possible face à des vagues de monstres qui vous attaquent en se ruant vers vous. **En êtes-vous capable ?**

Le principal objectif de la conception de ce jeu vidéo est de nous permettre de nous familiariser avec la dynamique des travaux de groupe afin de réussir efficacement un projet. C'est en étudiant la faisabilité du développement du jeu, en s'organisant afin distribuer les tâches, mais également en rédigeant un rapport et des comptes rendus, que nous pourrons ainsi mener à bien ce projet et développer de nouvelles compétences. De plus, un autre objectif est, à travers l'utilisation du langage de programmation python, d'appliquer toutes nos connaissances acquises des modules du semestre précédent et de nous introduire à la programmation orientée objet.

Ce projet est effectué dans le cadre du module du second semestre de première année de Cursus Master en Ingénierie Informatique. Dans ce module, les élèves sont répartis en groupes de 3 ou de 4 et se concertent afin de se mettre d'accord sur un projet, un nom et diverses fonctionnalités. C'est ainsi que nous nous sommes mis en groupe et grâce à nos points communs nous avons rapidement trouvé un terrain d'entente sur le fonctionnement, le thème, et le type du jeu.

# 2 Description générale du projet

Afin de faciliter la répartition des tâches lors du développement nous avons décidé de développer le jeu sous forme de fichiers python accueillant les différentes classes qui seront toutes importées dans un fichier main.py dans lequel la boucle du jeu tournera.

Divers menus seront proposés, avant le jeu, en pause et pour signaler la fin du jeu. Les joueurs pourront déplacer le personnage avec les flèches directionnelles et celui-ci infligera des dégâts de manière automatique. La vie du joueur sera visible grâce à une barre de vie et il y aura également des ennemis capables de se mouvoir et d'infliger des dégâts au joueurs. Tous ces éléments vont bien sûr évoluer dans un décor fini, une carte délimitée.

Le jeu prend place dans un univers fantastique avec une architecture qui rappelle celle du Moyen Âge. Le joueur sera donc immergé dans un univers inconnu dans lequel il doit survivre. Les amateurs de jeux de survie et d'action friand de nouveaux jeux pourront donc sauter sur l'occasion afin de découvrir un nouveau Rogue-Lite à jouer en solo.

### 3 Exigences

Afin d'étudier la faisabilité de notre projet ainsi que de dresser nos exigences avant la remise du code. Nous avons décidé de dresser une liste exhaustive de toutes les fonctionnalités et toutes les tâches à effectuer lors du développement du jeu.

I différentes taches :

(A) Menu d'accueil

- 1) faire un bouton pour lancer le jeu
- 2) faire un bouton pour quitter
- 3) faire des paramètres :
- 4) Régler le son
- 5) Gérer les contrôles :
  - 1 Régler le son
  - 2 Gérer les contrôles

(B) Menu pause

- 1) mettre le jeu en pause
- 2) reprendre
- 3) retour au menu
- 4) gérer le son et les commandes

(C) Interface

- 1) joueur au centre de l'écran
- 2) barre de vie
- 3) barre d'expérience
- 4) Timer
- 5) objet en la possession du joueur

(D) Joueur

- 1) joueur au centre de l'écran
- 2) barre de vie
- 3) barre d'expérience
- 4) Timer
- 5) objet en la possession du joueur

(E) Ennemis

- 1) Gérer les déplacements
- 2) Gérer les attaques
- 3) Gérer les points de vie
- 4) Gérer l'expérience
- 5) Gérer la montée de niveau (choix d'améliorations)

- 6) Gérer les items
- 7) États(feux,poison,...) pareil pour les ennemis
- (F) Carte
  - 1) Gérer les déplacements
  - 2) Gérer les attaques
  - 3) Gérer la vie
  - 4) Faires différents ennemis(avec possiblement différent comportement)
- (G) Design
  - 1) trouver et appliquer une direction artistique cohérente
  - 2) Avoir de beaux sprites
  - 3) Avoir de belles animations

## 4 Documentation

Au lancement du jeu, le joueur est directement face à une interface lui demandant s'il souhaite lancer une session. Une petite icône lui permettra également de lire les règles et lui expliquera de manière succincte le fonctionnement du jeu. Afin de rendre l'expérience la plus immersive, nous avons opté pour qu'après avoir cliquer sur le bouton, le joueur se retrouve directement dans une partie. Ainsi, à sa mort, le joueur verra son score affiché et son temps de survie.

En tant que développeurs du jeu, notre documentation se résume à celle des divers modules, ou éditeurs utilisés. Principalement Pygame, Tiled et Pyscroll ainsi que celle de python.

Afin d'assurer le fonctionnement du jeu sur n'importe quelle machine nous avons décidé que le jeu serait disponible sous forme de fichier exécutable sur nos githubs en public mais également il serait disponible de télécharger le code du jeu. Ainsi n'importe qui pourra contribuer (après la soutenance du projet) au développement du jeu et diverses mises à jour du jeu pourront être effectuées.

## 5 Contraintes

Pour ce projet il y a donc plusieurs contraintes à prendre en compte. Premièrement, bien que la bibliothèque Pygame possède une communauté assez active et une documentation, on peut rapidement manquer d'informations cruciales ou d'éléments qui pourraient faciliter le développement ou permettre l'optimisation du jeu et trouver de l'aide pourrait s'avérer être assez difficile. De plus, d'un point de vue performance, Pygame peut s'avérer être moins performant que d'autres bibliothèques de jeux ou plus lent ce qui pourrait porter atteinte au résultat final en rendant l'expérience du joueur peu appréciable.

Une seconde contrainte est l'utilisation de Tiled pour effectuer les maps. Etant un logiciel qui n'est pas installé sur les machines de la faculté nous serons

alors forcés d'effectuer le développement de la map sur nos ordinateurs personnels.

Une troisième contrainte sera l'aspect graphique du jeu. Nous devons dessiner nous-même les divers sprites et diverses interfaces ou différents décors ce qui peut être très chronophage.

La dernière contrainte, et la plus évidente, est l'organisation et la répartition des tâches. En effet, étant un groupe de 4, il peut être difficile de se séparer les différentes tâches de manière équitable.

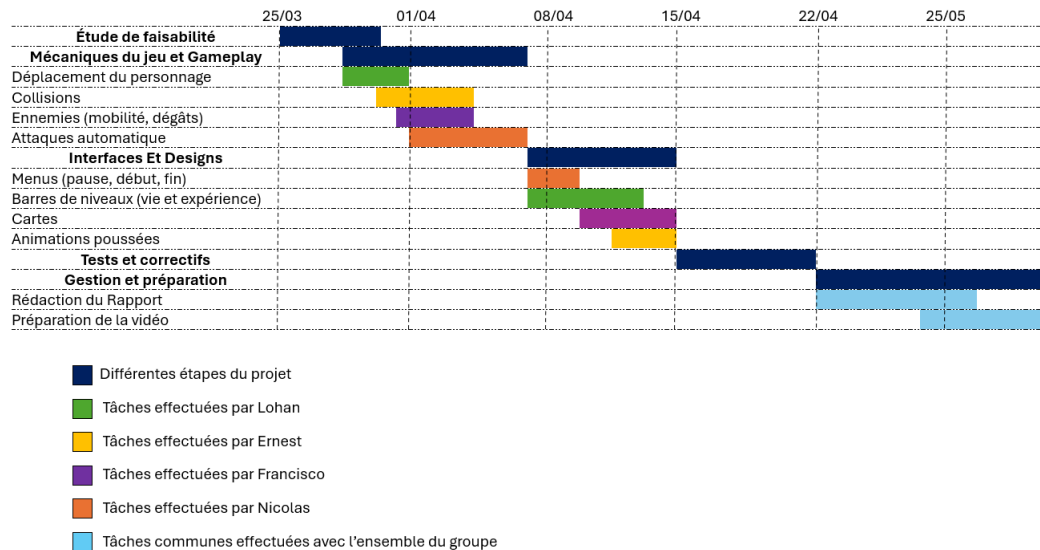
## 6 Organisation du développement

Afin de palier aux diverses contraintes nous avons choisi de s'organiser de la manière suivante:

Le projet sera organisé en plusieurs dossiers. Un premier dossier nommé "source" contenant l'intégralité du code et des différents fichiers .py, un dossier "maps" contenant l'intégralité des différentes cartes du jeu c'est à dire les fichiers .tmx, ainsi qu'un dossier "sprite" contenant l'intégralité des personnages du jeu.

Nous avons également décidé d'organiser les différentes phases de travail en dressant le Diagramme de Gantt suivant:

## 7 Diagramme de Gantt



## 8 Validation et tests

À la fin du développement, afin de valider le bon fonctionnement du jeu, nous passerons par plusieurs phases de test qui nous permettront de corriger les bugs

et améliorer l'expérience du joueur.

Nous estimons qu'à la fin du développement le jeu devra être complètement fonctionnel et remplir tous les objectifs que nous nous sommes fixés. C'est-à-dire, permettre aux joueurs une immersion totale dans le jeu avec un design adapté à celui prévu et des performances de jeu suffisante pour permettre aux joueurs de ne pas sortir de l'immersion d'éviter tous les inconvénients que pourrait rencontrer un utilisateur.

Afin de s'assurer que le jeu respecte tous ces critères nous passerons par plusieurs méthodes de tests. Premièrement nous testerons le jeu nous-même en poussant les performances pour essayer de découvrir les divers bugs et les problèmes de performance possible. Dans un second temps, après avoir veillé à permettre une expérience d'utilisateur convenable (absence de bugs ou de ralentissement) nous essayerons de permettre à des personnes hors du groupe de tester le jeu afin d'obtenir un avis extérieur sur la partie graphique mais également sur les diverses impressions et sensations que l'on peut ressentir que l'on peut ressentir en jouant. Ainsi nous pourrons à l'aide des divers avis, veiller à ce que le jeu soit adapté à la clientèle visée.

## 9 Ressources

Afin de mener à terme ce projet, nous aurons un total de 8 ordinateurs ( un ordinateur à la faculté par personne ainsi que son ordinateur personnel). Disposer de ces ressources nous permettra ainsi de travailler en présentiel avec l'équipe mais également en distanciel depuis chez nous et ainsi pouvoir avancer séparément pour optimiser le développement du jeu. D'un point de vu logiciel nous aurons besoin d'un éditeur de code tel que VScode ou VScodium, ainsi que d'avoir installer en amont python et les extensions nécessaires en utilisant la commande `pip install`. Afin de dessiner l'interface, les divers tilesets et personnages, nous aurons également besoin d'un logiciel de création graphique comme Gimp ou Krita. Enfin afin d'effectuer la création des niveaux. nous aurons besoin d'un éditeur de niveau tel que Tiled.

## 10 Sitographie

- 1 Github: Lien
- 2 Vampire Survivors: Steam