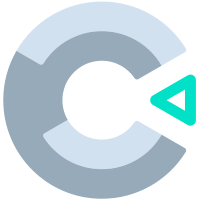
VIRUS 306

Document de développement d’un jeu en 2D pour le module M306. Le but de ce document est de présenter le jeu et d’expliquer l’histoire, les mécanique et le fonctionnement du jeu

Développé par : Callau Vera Ivan, Loretz Gaetan et Zeqiri Erblin Module : M306 Groupe : I.IE-D3A Professeur : Dominique Aigroz



Project Scope

Table des matières

[Introduction 2](#_Toc98856119)

[Objectif du projet 2](#_Toc98856120)

[Spécification du projet 2](#_Toc98856121)

[Catégorie de jeu 2](#_Toc98856122)

[Histoire du jeu 2](#_Toc98856123)

[But 2](#_Toc98856124)

[Vue générale du jeu 2](#_Toc98856125)

[Analyse 3](#_Toc98856126)

[Les entités 3](#_Toc98856127)

[L’avatar du joueur 3](#_Toc98856128)

[Les ennemis 3](#_Toc98856129)

[Le HUD 4](#_Toc98856130)

[Les projectiles 5](#_Toc98856131)

[La map 5](#_Toc98856132)

[Propriétés 6](#_Toc98856133)

[Tâches 7](#_Toc98856134)

[Création d’un niveau 7](#_Toc98856135)

[Création du gameplay 7](#_Toc98856136)

[Conception des graphiques 7](#_Toc98856137)

[Organisation du projet 9](#_Toc98856138)

[Convention de nommage 10](#_Toc98856139)

[Nom des entités 10](#_Toc98856140)

[Variables 10](#_Toc98856141)

[Conclusion 11](#_Toc98856142)

[Références 12](#_Toc98856143)

# Introduction

Pour notre projet nous devons créer un jeu de survie du type « shoot ‘em up » en vue de dessus nommé « Virus 306 », en référence au module du cours. Le joueur se servira d’une arme à distance pour éliminer les ennemis qui lui fonceront dessus pour le tuer. Une partie se termine quand le joueur n’a plus de points de vie et meurt. Une fois la partie terminé le nombre d’ennemis tués est comptabilisé et le temps de survie pour donner un score au joueur.

# Objectif du projet

Création d’un jeu en 2D sur l’outil "Consctruct 3". Ce jeu consistera à tuer des vagues de zombis qui viennent attaquer le protagoniste.

## Spécification du projet

### Catégorie de jeu

Un jeu "Shoot ’em up" en 2D et une vue au-dessus du personnage. Le concept de ce type de jeu consiste à, muni d’une arme à distance, éliminer les ennemies tout en survivants aux différentes vagues.

### Histoire du jeu

Vous êtes le dernier survivant de l’épidémie du Virus 306 qui a transformé l’humanité en créatures monstrueuse et moche. Plutôt que d’en finir tout de suite, P.Tron (protagoniste) décide d’en emporter le plus possible avec lui avant de mourir!

### But

Il faut survivre à tout prix aux vagues des créatures. À l’aide de vos armes, il faudra entrainer votre "Aim" pour tous les exterminer. Le score contiendra le temps survécu et le nombre d’ennemis tués.

Vue générale du jeu

# Analyse

Dans un premier temps nous avons discuté des éléments nécessaires pour le jeu. Nous retrouvons parmi eux :

## Les entités

Une entité est un objet dans le jeu avec lequel il y aura une interaction. Parmi les entités nous trouverons :

L’avatar du joueur

Sa position sera au centre de l’écran à moins que la bordure de la fenêtre touche la limite de la map, auquel cas l’avatar se décollera du centre pour toucher les bords.

Il sera muni d’une arme à feu à distance pour se défendre et tuer les ennemis.

L’arme fera apparaitre des balles lorsque la souris sera pressée, les balles partiront du bout du canon dans la direction de la souris par rapport à la position de l’avatar.

Le joueur déplacera son avatar avec les touches classiques du jeux vidéo sur PC : W, A, S, D.

Sa vitesse de déplacement sera la référence de base des autres entités ou des projectiles.

Exemple : la vitesse du joueur est 10px/sec (pixels par secondes).

### Les ennemis

Les ennemis apparaitront au hasard sur la map et seront attiré par la position actuelle du joueur.

Les ennemis auront une vitesse de déplacement plus grande que celle du joueur, afin de forcer le joueur à s’en débarrasser au lieu de les esquiver. Cette différence sera de l’ordre de 10%. Exemple : si la vitesse de l’avatar est de 10px/sec alors les ennemis auront une vitesse de 11px/sec.

## Le HUD

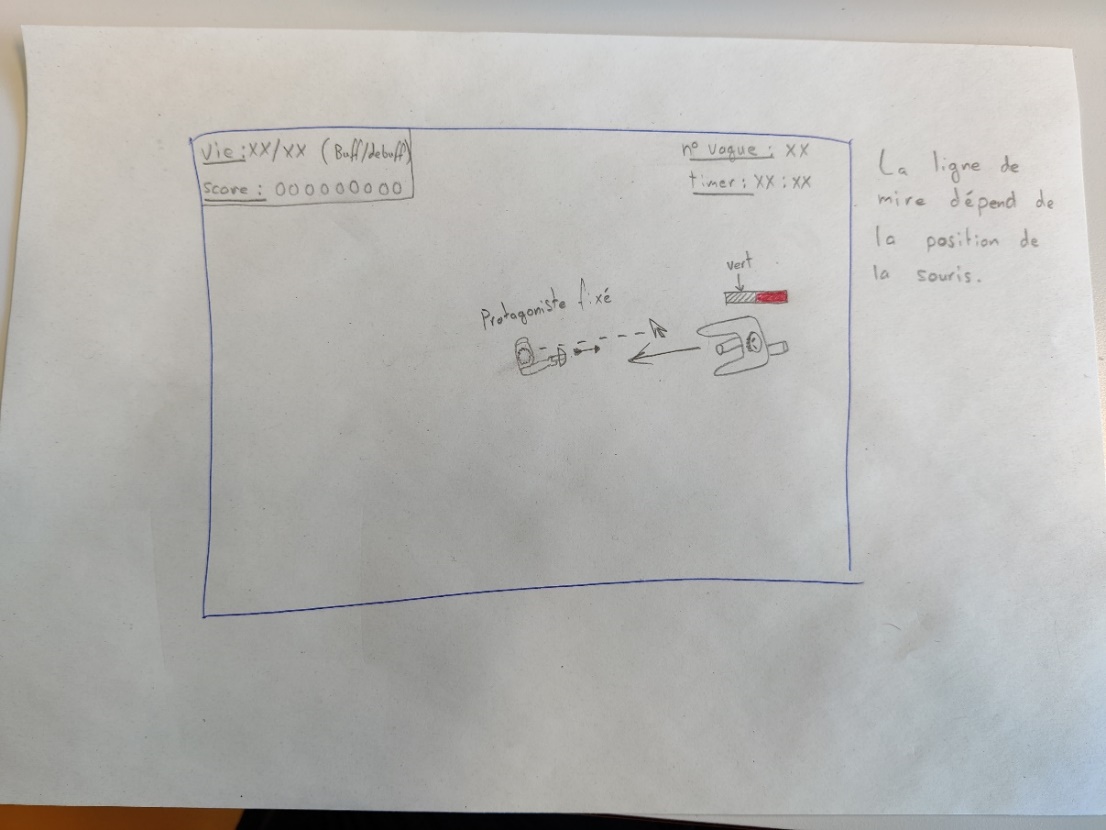
Le HUD (heads-up display), ou « l’affichage tête haute » en français, est un layer qui affichera les informations du jeu en cours au joueur pendant une partie.

Dans le schéma qui suit nous avons fait une ébauche de ce que le joueur verra et quelques idées de l’apparence finale du HUD.

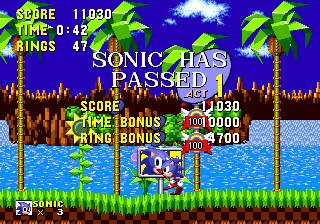
Nous y retrouvons à gauche : la vie du joueur, une zone ou les buffs et debuffs s’affichent et en dessous le score qui s’affiche au fil du jeu.

À Droite : Le numéro de la vague en cours et le temps qui s’est écoulé depuis le début de partie.

Nous avons aussi pensé à afficher la vie des ennemis par une barre verte au-dessus de leur tête.



Il aura une barre de vie qui indiquera au joueur s’il est proche du « Game over ». Une fois tombé à zéro, la partie s’arrête et le score est affiché en grand au joueur ainsi que son temps de survie.



Exemple de score : Sonic the Hedgehog

## Les projectiles

Les projectiles seront essentiellement le moyen avec lequel le joueur se débarrasseras des ennemis. Mais certains ennemis seront aussi capables d’en lancer au joueur pour le tuer.

Il y a les balles de l’arme à feu de base du joueur, mais nous avons aussi pensé à avoir différentes armes au fil du jeu comme par exemple :

* Lance grenade -> projectile : bombes
* Arbalète -> projectile : flèches
* Lance-flammes -> projectile : feu

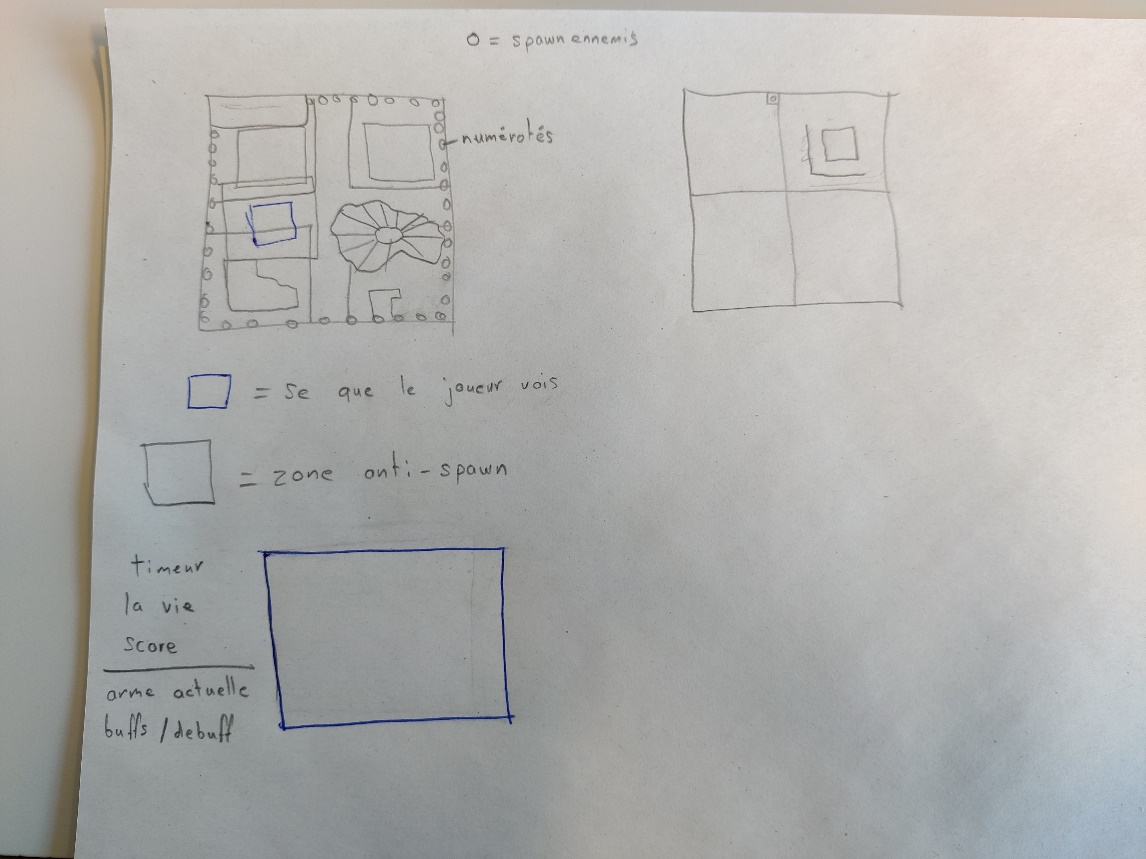
Les projectiles lancés par le joueur auront une vitesse supérieure au joueur de l’ordre de 100%. Exemple : si le joueur est à 10px/sec les balles se déplaceront à 20px/sec.

## La map

La map sera un carré de 1700 pixels de côté.

La bordure sera la limite de la zone de jeu.

Les ennemis apparaitront sur toute la map en dehors de la vue du joueur. Nous avons initialement pensé à diviser la map en quarts, pour gérer les spawns des ennemies. Mais l’idée a été jugée irréalisable car trop exploitable. Nous nous sommes donc penché sur une zone un peu en dehors de ce que le joueur peut voir pour empêcher les ennemis d’apparaitre trop prêt du joueur.



## Propriétés

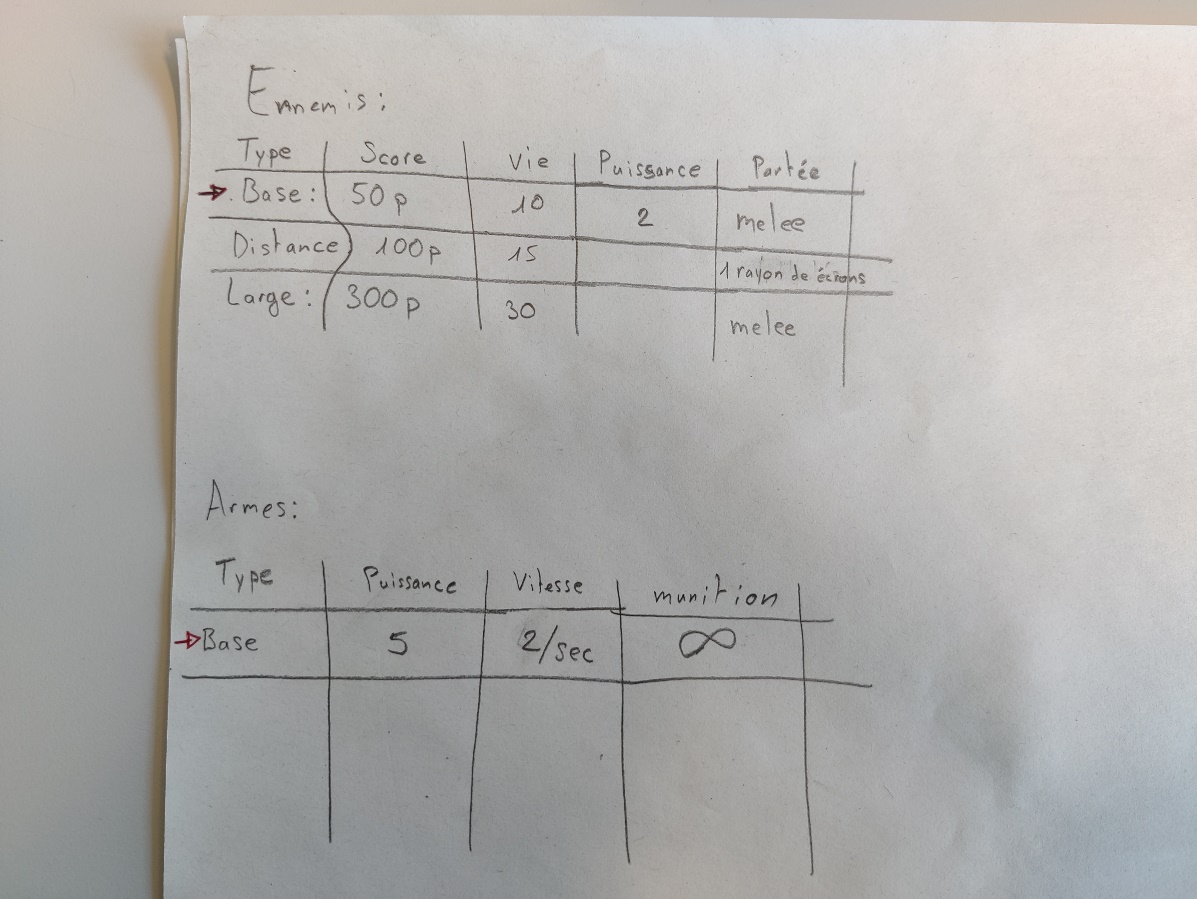
Nous avons discuté des propriétés que nos entités auraient besoin.

Notamment pour les ennemis :

* Le type (Type)
* Le score quand éliminé (Score)
* Ses points de vie (vie)
* La quantité de dégât engendré (Puissance)
* La portée des dégâts (Portée) : Au contact (melee) et à distance

C’est aussi sur cette feuille que nous avons survolé les propriétés de l’arme de base :

* Ses dégâts (Puissance)
* Sa cadence de tir (Vitesse)
* Sa limite de munition (munition)



# Tâches

Nous avons décidé de manière générale que : Ivan s’occupera des maps et de la doc, Gaëtan de la mécanique des ennemis et du score, et finalement, Erblin de la mécanique de l’avatar du joueur et la barre de vie.

Cela a conduit à définir les tâches principales :

## Création d’un niveau

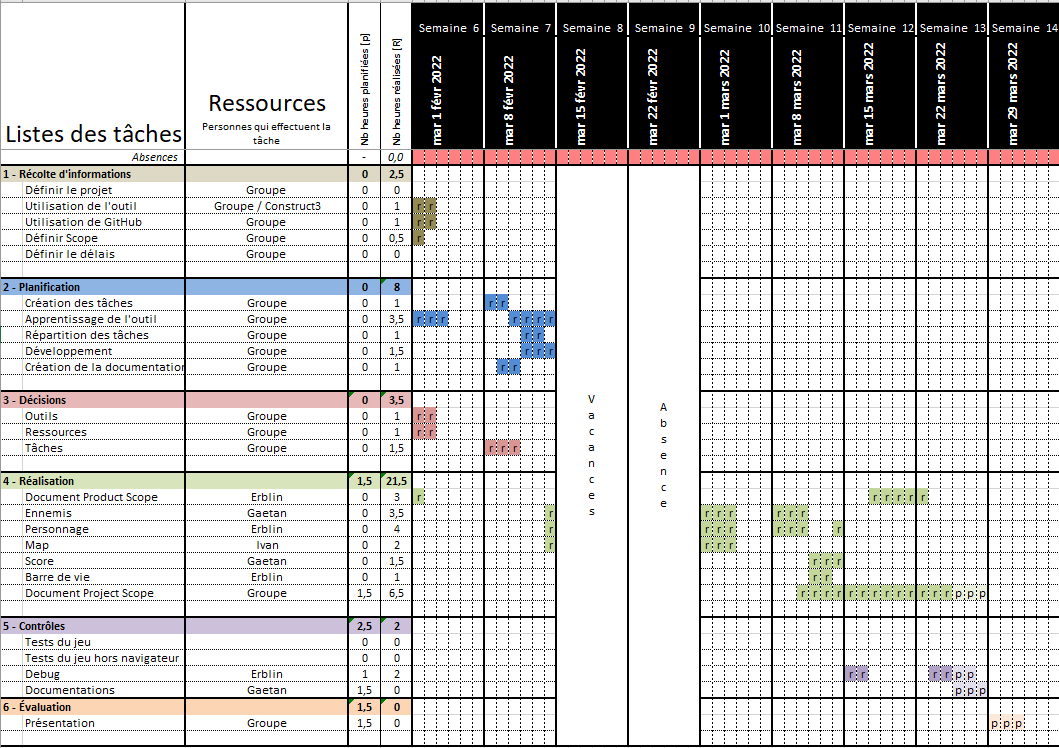
* Arrière-plan :
  + Obstacles.

## Création du gameplay

* Ennemis :
  + Vie.
  + Mouvements.
  + Variété.
  + Comportement.
* Protagoniste :
  + Vie.
  + Mouvements.
* Armes :
  + Variété.
  + Mode de tir (press).
  + Type de projectiles.
* Objets de buff.
* Score.
* Chronomètre.

## Conception des graphiques

* Effets :
  + Sang.
  + Coup de feu / Flash.
  + Animation de mort.



# Organisation du projet

Il a été décidé que le projet serai mis à jour et partagé à travers GitHub sous le nom de « CFPT\_M306 ». Le repository a été créé sous le compte de Ivan Callau Vera (IvanCallau) et son accès a été donné aux autres membre du groupe : Erblin Zeqiri (erblinzqr) et Gaëtan Loretz (VodkGaLaXy), ainsi qu’au maitre de classe : Dominique Aigroz (DominiqueAigrozEDU).

Nous avons ensuite décidé de séparer le projet en différents dossiers :

Assets : (En français « Ressources ») Ce dossier contient les images utilisées pour le projet : Fonds de la map, avatar joueur, ennemis, etc…

cr3 : (Abréviation de « Construct 3 ») Ce dossier contient le code que chacun aura écris dans un dossier avec le prénom du développeur comme nom ainsi qu’un dossier « Main » qui sera utiliser pour fusionner les travaux de tous en un.

docs : (Abréviation de « Documents ») Ce dossier contient tous les documents relatifs au projet :

* « Product Scope » est un fichier Word qui contient la demande client pour ce projet.
* « Project Scope » est le fichier Word que vous lisez en ce moment et qui détail ce qui a été fait lors de ce projet.
* « journal debord\_v2022 » est un tableau Excel qui contient le déroulement des jours passé sur ce projet.
* « M306\_Planning\_2022 » est un tableau Excel qui contient le planning du projet sur sa durée.

Une fois ceci mis en place nous avons ensuite délibéré sur quels étaient les éléments les plus critique du projet et nous en somme arrivé à la conclusion que 3 éléments étaient indispensables au fonctionnement du jeu : l’avatar du joueur, les ennemis et la map. Nous avons donc réparti la charge entre nous trois pour par la suite mettre en commun nos travaux.

# Convention de nommage

Pour le développement du projet, nous avons décidé que tout ce qui serai dans le code, c’est-à-dire : les entités et les variables, seraient toutes nommées en anglais et en camelCase.

## Nom des entités

* Avatar du joueur -> player
* Ennemis -> enemy
* Obstacles :
  + Trous -> hole
  + Murs -> wall
* Projectiles :
  + Balles -> bullet
  + Attaques à distance ennemi -> rangedShot
* Zone de d’apparition d’ennemis -> spawn
* Chronomètre -> timer
* Points obtenus -> score
* Barres de vie -> healthBar
* Souris -> mouse
* Clavier -> keyboard

## Variables

* Santé -> health
* Cadence de tir -> fireRate
* Vitesse de déplacement -> groundSpeed
* Vitesse de projectiles -> shotSpeed
* Dégâts provoqué -> damage
* Numéro de vague -> wave
* Nombres d’ennemis par vagues -> enemyNumber

# Conclusion

Malheureusement le temps impartis pour ce projet, même avec les prérequis minimums pour rendre un produit fonctionnel, n’était pas suffisant.

L’outil Construct3 était relativement facile à prendre en main, mais malgré cela des problèmes bloquants sont arrivé lors du développement. Notamment quand est arrivé le temps de tester la fusion de nos codes en un, ce qui a provoqué un mal fonctionnement du code. S’en sont suivis des tentatives de récupération de fichiers et de debugging qui nous ont couté plusieurs heures de travail, des heures que nous n’avions pas…

D’autres soucis ont émergés lorsque nous nous sommes rendu compte que nos documents n’étaient, ni à jour, ni complets. Ce qui nous a forcé à stopper le développement de la map afin de combler les blancs. L’expérience obtenus des problèmes rencontrés, comme notre manque d’expérience en gestion de projets et du fonctionnement de l’outil, nous permettrons à l’avenir de mieux gérer un projet sur Construct3.

D’autres part, nous pensons que c’était intéressant de faire un projet comme ça en groupe. Cela nous a permis de voir sous un autre œil le temps que des développeurs passent sur des jeux et de l’expérimenter par nous-même. La leçon que nous en avons tiré est que :

**Développer un jeux ce n’est pas aussi facile que l’on pourrait y croire.**

# Références