Spicy Invaders

ETML – Lausanne

LG – JS – AB – TR

Tables des matières

[1. Introduction 3](#_Toc41643426)

[2. Planification et journal de travail 3](#_Toc41643427)

[3. Schéma UML 4](#_Toc41643428)

[4. Document technique 5](#_Toc41643429)

[5. Mode d’emploi 6](#_Toc41643430)

[6. Conclusion 7](#_Toc41643431)

# Introduction

Ce document a pour but d’explique notre projet Spicy invader inspiré d’un célèbre shoot 'em up sur borne d’arcade.

Notre objectif était de faire une nouvelle version du jeu en C# sur console. Pour ce faire, nous avons travaillé en groupe, notamment avec l’aide de gitKraken pour gérer les avancées de chacun.  
Lien vers le git : <https://github.com/BetterBlood/SpicyInvaders>

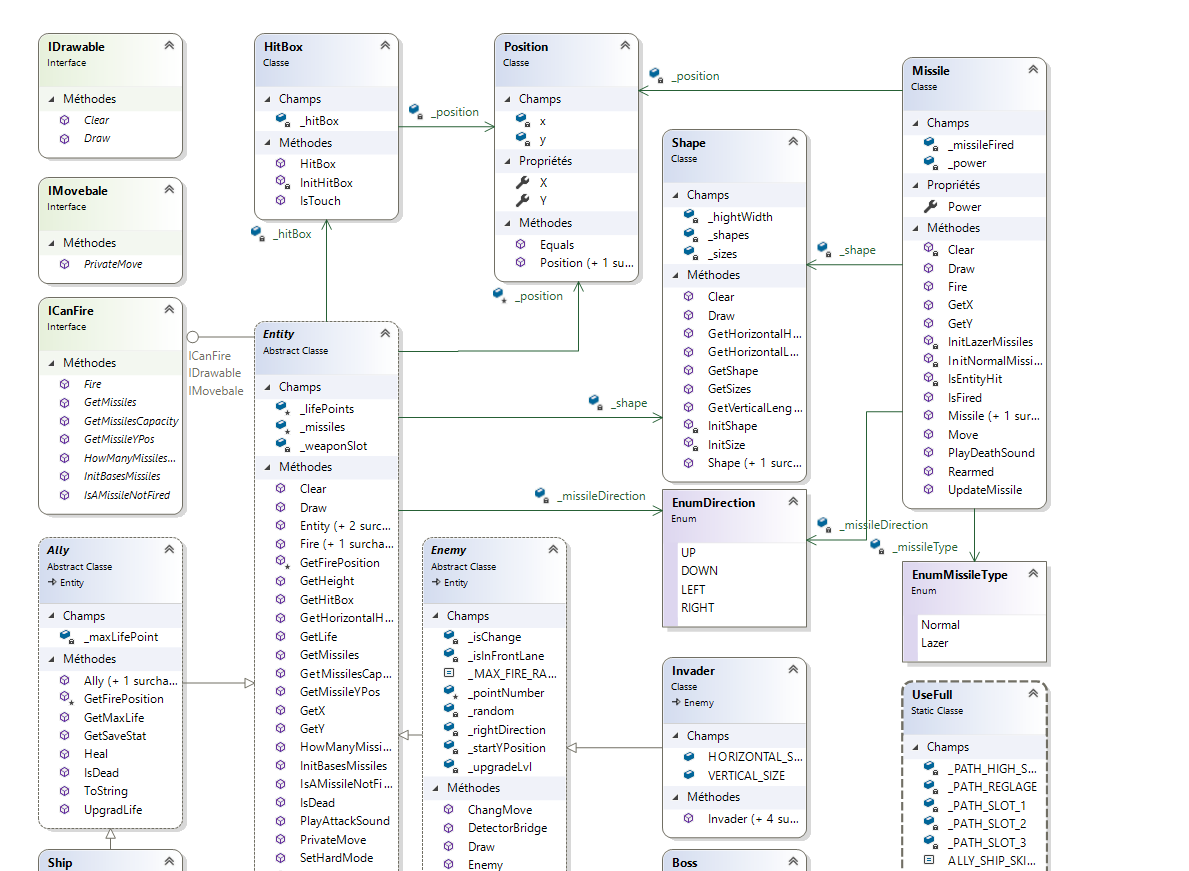
# Planification et journal de travail

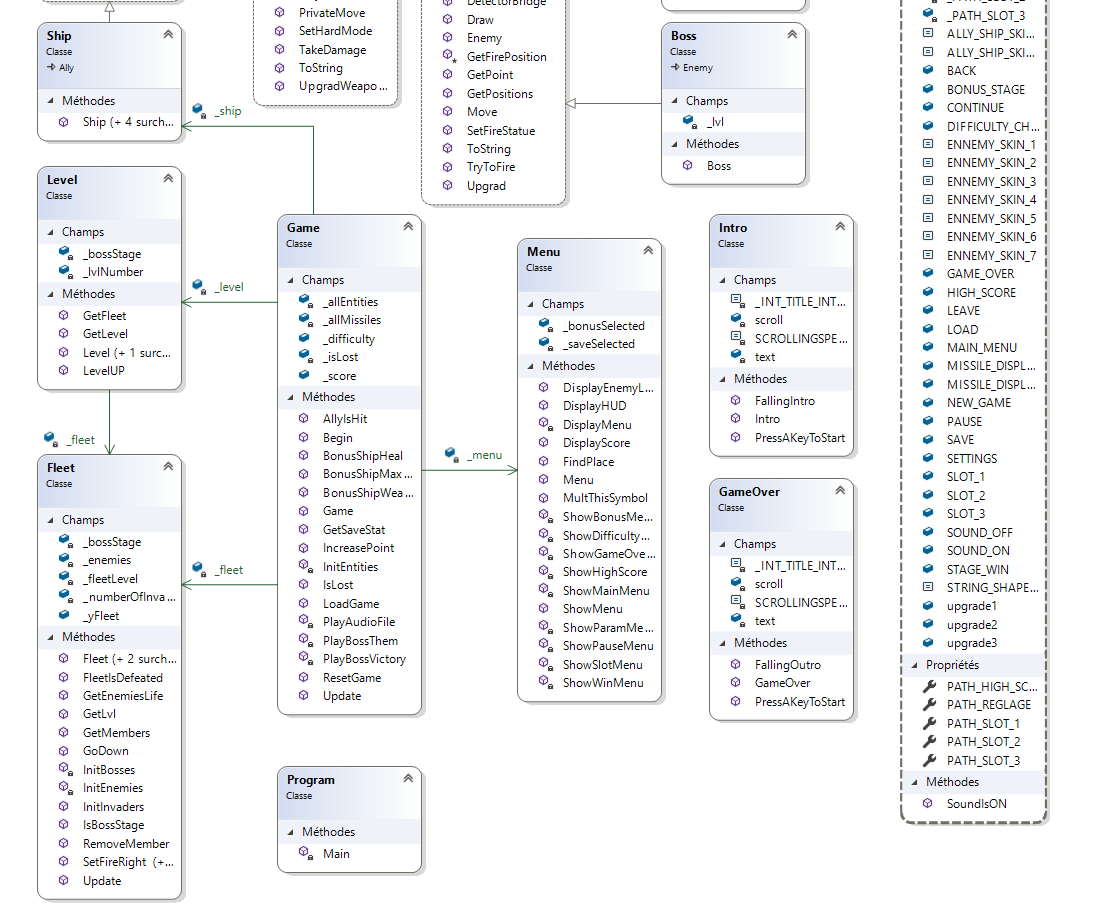
Nous avons décidé de ne pas montrer la planification car en raison du corna-virus et du projet pluridisciplinaire, il est très éloigné de la réalité. Par conséquent nous avons choisi de présenter la répartition des tâches durant ce projet.

| Titre | Description | Auteurs |
| --- | --- | --- |
| Recherche d’information | Recherche d’information global sur le projet et lecture du CDC | Adrian, Laetitia et Jeremiah |
| Planification | Planification globale du projet | Toine, Adrian, Laetitia et Jeremiah |
| Menu | Création du menu du jeu | Adrian |
| Action du joueur | Le joueur tire et se déplace | Adrian, Laetitia et Jeremiah |
| Squelette du code |  | Jeremiah |
| Hiérarchie des classes |  | Jeremiah |
| Création des ennemis |  | Jeremiah |
| Déplacement des ennemis | Déplacement en groupe des ennemis | Laetitia |
| Animation | Animation d’introduction et de Game Over | Adrian et Jeremiah |
| Gestion des niveaux et des difficultés | Création de divers niveaux et choix des difficultés | Jeremiah |
| Aspect graphique du jeu | Dessin des personnages et design du menu | Toine, Adrian et Jeremiah |
| Son et paramètre | Musique et bruitage | Jeremiah |
| Sauvegarde et gestion du scores | Sauvegarde des scores | Jeremiah |
| Tests | Vérification du fonctionnement du jeu | Adrian, Laetitia et Jeremiah |
| Documentation technique | Doxygene | Laetitia |
| Commentaire | Tout type de commentaires dans le code | Laetitia et Jeremiah |
| Manuel d’utilisation |  | Adrian, Laetitia et Jeremiah |
| Test Unitaire |  | Jeremiah |
| Schéma UML | UML fait avec Visual Studio Entreprise | Jeremiah |
| Documentation |  | Adrian, Laetitia et Jeremiah |

# Schéma UML

Notre est séparé en deux parties, dans la première partie il y a Entity. il s’agit de la classe principale dans la quel se trouve toutes les méthodes les plus importantes utilisé par les objets du jeu comme le vaisseau et les ennemis

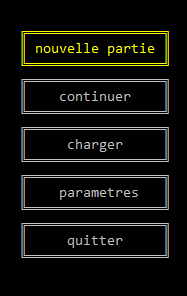




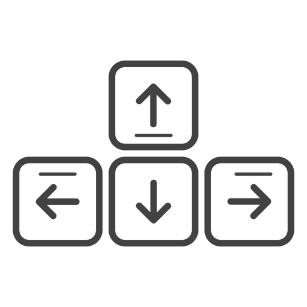
# Document technique

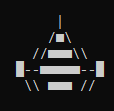
Lien vers le document technique fait avec Doxygene :

# Mode d’emploi

Après la page d’intro, le jeu s’ouvre sur le menu.

Pour se déplacer dans le menu, il faut utiliser les flèches directionnelles haut et bas puis sélectionner avec la touche espace.



Une fois le jeu lancé, pour contrôler votre vaisseau utilisez les flèche directionnelles droite et gauche. La touche espace vous permet de tirer. Attention vous ne pouvez pas tirer à l’infini, une fois votre chargeur vide vous devez attendre un petit temps de rechargement.

Votre vaisseau

Vous ennemis vous tirerons aussi dessus, alors ne tombez pas à cours de vie, si vous voulez éviter le Game Over.

Vos vies

Votre avez perdu !

Votre but est de vaincre un maximum d’ennemi sans obtenir un Game over !

Un boss

Un ennemi

Vos missiles disponibles

# Conclusion