# P_OO “SHOOT ME UP”

Borodkin Oleksandr – GRP2D
ETML - Vennes
32p
Xavier Carrel

# Table des matières

# 1 SPÉCIFICATIONS

## 1.1 Titre

Shoot me up !

## 1.2 Matériel et logiciels à disposition

- Un PC ETML
- Accès à Internet

## 1.3 Prérequis

Modules de programmation de base
- ICT-320 en cours

# 2 PLANIFICATION INITIALE

### *2.1.1 Semaine 1*

- Création de concept de jeu.
- Création des maquettes

### *2.1.2 Semaine 2*

- Création des User stories

### *2.1.3 Semaine 3*

- Création de la grille de jeu
- Codage du joueur et ses mouvements
- Codage des ennemies et ses mouvements
- Codage de la « zone de spawn » pour ennemies
- Codage de la zone de tirs pour les ennemies

### *2.1.4 Semaine 4*

- Codage des missiles de joueur et des ennemies
- Codage des obstacles et sa position aléatoire
- Codage de la base de joueur
- Création des textures pour le jeu

### *2.1.5 Semaine 5*

- Codage des fonctionnalités des vies, missiles et score
- Codage du boss

### *2.1.6 Semaine 6*

- Codage des écrans de la victoire et de la défaite

- Livraison 80% du projet

#### *2.1.7 Semaine 7*

- Améliorations du projet

#### *2.1.8 Semaine 8*

- Améliorations du projet
- Livraison finale du projet

# 3 INTRODUCTION

## 3.1 Concept de jeu

Le jeu « Arstotzka Invaders » est un jeu qui ressemble à Space Invaders et s'inspire de Papers, Please, avec son style pixel art et son univers basé sur Arstotzka (le nom du pays dans le jeu). Cependant, il se distingue par sa jouabilité.
Dans ce jeu, le joueur incarne un drone chargé de défendre sa base contre des vagues de drones et d'avions ennemis pendant une minute, avant d'affronter le boss final.

## 3.2 Objectifs pédagogiques

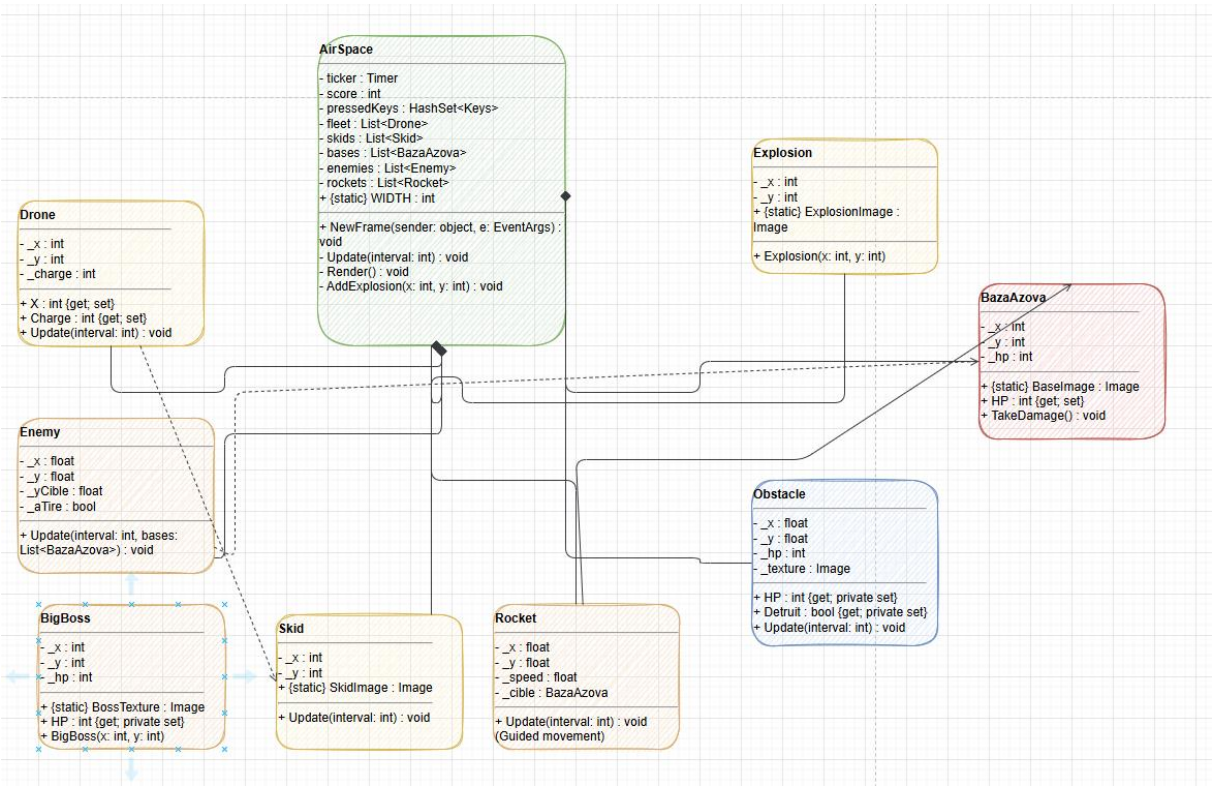
Ce projet est utile sur le plan **pédagogique**, car il permet d'apprendre :

- Le **concept de la programmation orientée objet** et son fonctionnement ;
- L'**utilisation et la gestion de projet** avec **GitHub Projects** ;
- La **gestion de projet en méthodologie Agile**.

## 3.3 Objectifs produit

L'objectif du produit est de **créer un jeu en C#**, basé sur la **programmation orientée objet**, qui s'inspire de **Space Invaders** et, dans une certaine mesure, de **Papers, Please**, afin d'en **améliorer le concept et l'expérience de jeu**.

## 3.4 Diagramme UML

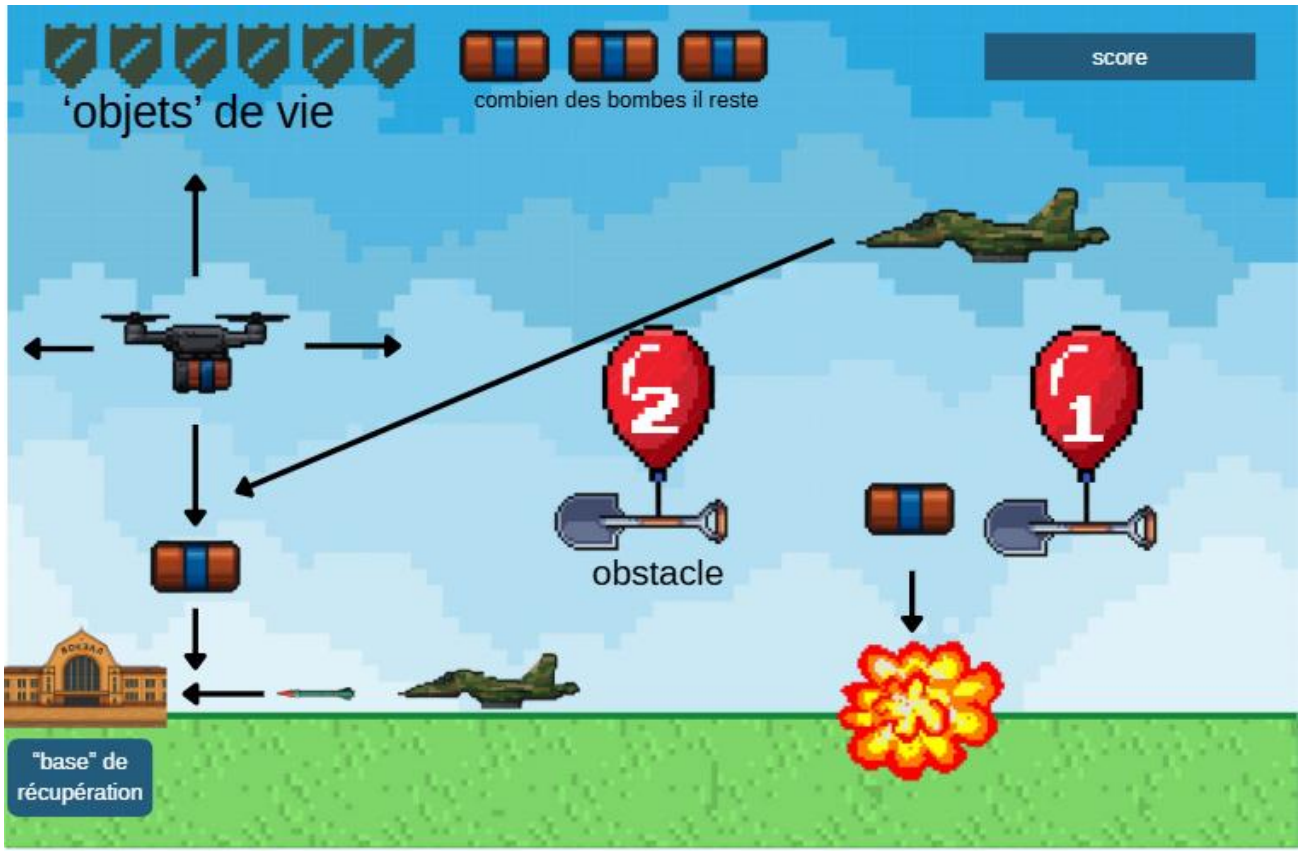



Charge – combien des projectiles sont possédés par joueur
Cible – la base cible des ennemies
BigBoss – constructeur du boss
SkidInage . Image du projectile
BaseImage – image de la base de récupération
Detruit – si l'obstacle est detruit

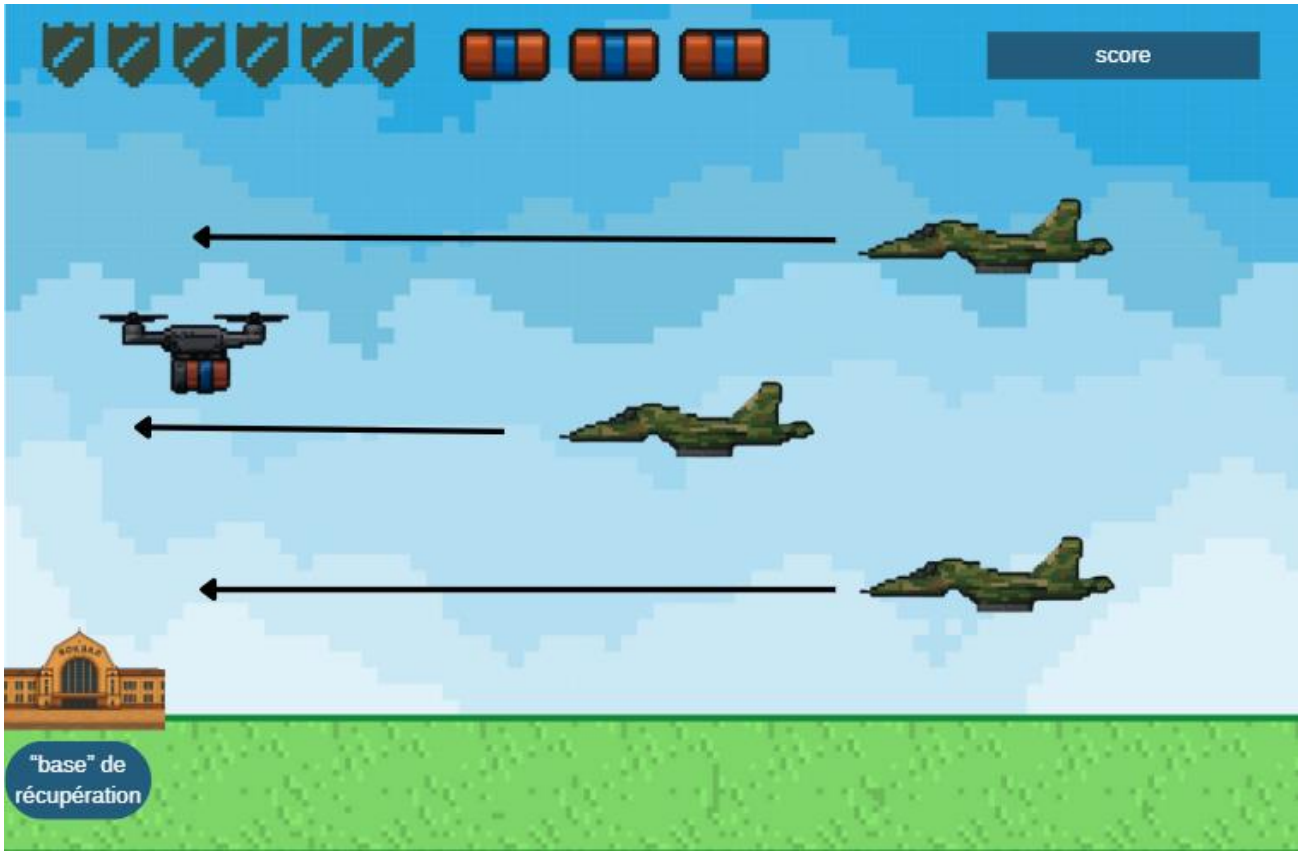
# 4 ANALYSE FONCTIONNELE

## 4.1 Maquettes

Le jeu principal




Mouvement des ennemies




Boss battle

Ecran de victoire

Ecran de défaite


## 4.2 Fonctionnalités basiques de jeu

### *4.2.1 Joueur*

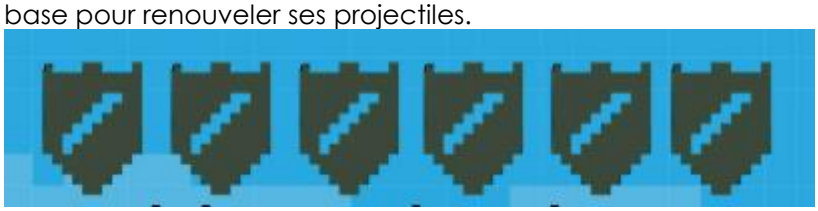
Pour se déplacer, le joueur doit appuyer sur les touches W, A, S, D ou sur les touches directionnelles (flèches).

Pour lancer un projectile, l'utilisateur doit appuyer sur la touche Espace.
L'utilisateur ne dispose que de trois projectiles à la fois.


Pour récupérer des projectiles, le joueur doit revenir à la base.


### 4.2.2 Base de récupération

La base dispose de 6 points de vie (PV) et le joueur doit la protéger tout en revenant à la base pour renouveler ses projectiles.



### 4.2.3 Ennemies et Boss

Il y a deux types d’ennemis qui apparaissent toutes les 5 secondes (2 à 3 ennemis à la fois) : les drones.

Et avions

Il n’y a aucune différence entre les ennemis, sauf au niveau de la texture.
Les ennemis se déplacent vers la position X de la base avec une coordonnée Y aléatoire. Lorsqu’ils se rapprochent à 170 px de la base, ils tirent puis repartent. Chaque ennemi ne

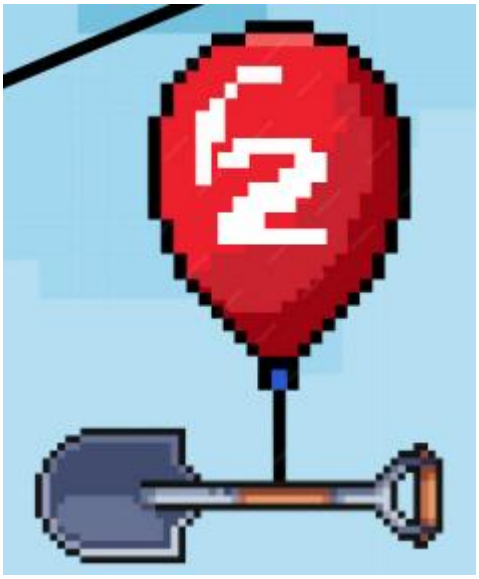
peut tirer qu'une seule fois.Pour les détruire le joueur doit de lancer un projectile dessus, après la destruction le jouer va recevoir 10 points.

Après une minute de jeu, tous les ennemis disparaissent et le boss apparaît.

Le boss est inspiré de Dr. Eggman (Dr. Robotnik) de l'univers de Sonic.
Il dispose de 10 points de vie (PV) et fait apparaître 5 ennemis toutes les 5 secondes.
Après sa destruction, le joueur reçoit 1000 points.

### *4.2.4 Obstacles*

L'obstacle a la texture d'un ballon avec une pelle et un chiffre.



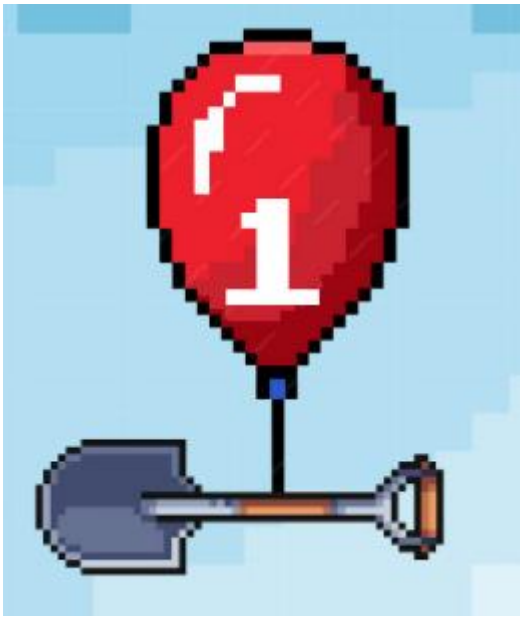

Le chiffre affiché sur l'obstacle correspond à ses points de vie (PV).

Chaque obstacle a 2 PV et ni le joueur ni les ennemis ne peuvent le traverser.

Si un obstacle se trouve à proximité d'un ennemi, celui-ci modifie son chemin.

Le joueur peut détruire un obstacle avec 2 projectiles.

La texture de l'obstacle change lorsqu'il perd des PV.




Quand il a 0 PV, il disparait.

### 4.2.5 *Victoire et défaite*

Quand le PV de la base est 0 le joueur perd et l'écran de la défaite s'affiche


Par contre, si boss a 0 PV, le joueur gagne et l'écran de victoire s'affiche


## 4.3 Fonctions Principales

**Structure et gestion du jeu**

• **Gestionnaire principal du jeu (Airspace.cs)**
Initialise, met à jour et dessine tous les éléments du jeu à chaque frame (NewFrame) via un ticker. Il gère également le score, la fin de partie et le chargement des ressources.

• **Gestion des attaques du joueur (Airspace.cs, Drone.cs, Skid.cs)**
Ces classes gèrent le mouvement du drone à l'aide des touches pressées (pressedKeys) ainsi que le tir des projectiles pour détruire les ennemis.

• **Génération et gestion des ennemis (Airspace.cs, Enemy.cs, Rocket.cs, BigBoss.cs)**
Les ennemis se déplacent, tirent des roquettes sur la base, et le boss a la capacité de faire apparaître d'autres ennemis.

• **Gestion des projectiles du joueur (Airspace.cs, Drone.cs, BazaAzova.cs)**
Le nombre de projectiles actifs est limité à trois lorsque le drone se trouve sur la base.

• **Gestion des obstacles (Airspace.cs, Obstacle.cs)**
Ces classes créent des objets infranchissables qui bloquent le passage du drone et des ennemis.

- **Effets visuels (Explosion.cs)**

Cette classe gère les effets d'explosion lors de la destruction d'ennemis ou d'objets.

# 5 UTILISATION D'IA

Dans ce projet, l'intelligence artificielle a été utilisée à plusieurs reprises pour faciliter le développement et améliorer la qualité du rendu final.
Elle a notamment servi à générer les textures du jeu, permettant ainsi de gagner du temps dans la création graphique tout en maintenant un style visuel cohérent avec l'univers du projet.

L'IA a également été utilisée comme outil d'assistance technique, par exemple pour comprendre comment intégrer des images embarquées dans le code et optimiser certaines parties du développement.

Enfin, elle a été mise à contribution pour la correction orthographique et grammaticale du rapport, garantissant un texte clair, fluide et professionnel.

Grâce à ces apports, l'IA a contribué à accélérer le développement, à améliorer la présentation du projet et à optimiser la qualité globale du travail rendu.