

I320 Shoot them up



Damien Chabal
ETML
Xavier Carrel

Table des matières

1 INTRODUCTION 3

1.1 OBJECTIFS PÉDAGOGIQUES 3

1.2 PRODUIT 3

2 PLANIFICATION 3

3 ANALYSE FONCTIONNELLE 3

4 CONCEPTION 5

5 USAGE DE L'IA 6

1 Introduction

Le projet Shoot Me Up, consiste à développer un jeu vidéo 2D. Il vise à appliquer les compétences en programmation et gestion de projet tout en respectant un cahier des charges. Pour pouvoir livrer un jeu fonctionnel.

1.1 Objectifs pédagogiques

Ce projet va aider à apprendre :

- Faire de la programmation en OO
- Faire du code avec Windows Forms pour faire un programme plus poussé
- La gestion projet avec GitHub

1.2 Outils utiliser

Visual studio code
GitHub desktop
GitHub Project
Windows Forms

1.3 Produit

Un jeu en 2D dans lequel l'utilisateur contrôle le personnage grâce au touches directionnelles et a la barre d'espace pour tirer des projectiles sur les ennemis et les obstacles. Le but du jeu est de vaincre les plus d'ennemis possibles pour avoir le plus haut score tout en esquivant les tirs des ennemis. (Il est possible d'encaisser deux tris avant d'être éliminé.) Quand le premier ennemi est éliminé il y en a trois qui apparaissent puis quand un ennemi est vaincu un nouveau apparaît de sorte à ce qu'il y ait constamment trois ennemis contre le joueur a part au début.

2 PLANIFICATION

Semaine 1	Introduction et analyse
Semaine 2	Création des maquettes et des User Story
Semaine 3	Commencer et finir à coder un personnage et les déplacements du joueur
Semaine 4	Commencer et finir de coder les ennemis et coder les déplacements des ennemis
Semaine 5	Commencer / finir de coder les projectiles
Semaine 6	Commencer / finir de coder les obstacles
Semaine 7	Coder la vie des objets/personnages
Semaine 8	Voir et faire ce qu'il reste à faire

3 ANALYSE FONCTIONNELLE

3.1 Objectif du jeu

Pouvoir contrôler un personnage en se déplaçant avec les touches directionnelles et pouvoir tirer grâce à la touche espace sur les ennemis et les obstacles.

3.2 Contrôle du joueur

Pour contrôler le joueur il faut utiliser les touches directionnelles et pour qu'il tire un projectile il faut cliquer sur la barre d'espace.



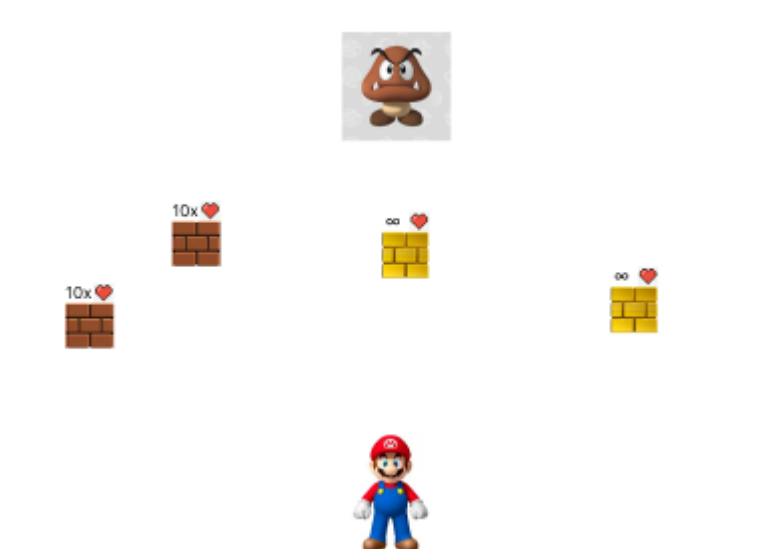
3.3 Fonctionnement des ennemis

Au début du jeu un ennemi apparaît en haut de l'écran puis cet ennemi se déplace de droit à gauche et il tire des projectiles une fois par secondes. L'ennemi a trois points de vie donc quand il se fait toucher par trois projectiles il disparaît et trois autres ennemis apparaissent puis trois ennemis sont constamment en vie dans le jeu.



3.4 Fonctionnement des obstacles

Au début du jeu il y a cinq obstacles qui apparaissent aléatoirement vers le milieu de la fenêtre. Chaque obstacle a 10 points de vie et ils perdent un point de vie à chaque fois qu'ils reçoivent un projectile d'un joueur ou d'un ennemi. Quand un obstacle a 0 points de vie il disparaît et un nouveau apparaît aléatoirement dans la fenêtre.



Finalement il n'y a pas d'obstacles incassables.

3.5 Fonctionnement des projectiles

Pour le joueur : Quand l'utilisateur appuie sur la barre d'espace sa fait apparaitre devant le joueur un projectile qui va droit devant.

Pour les ennemis : Toute les seconde un projectile apparait devant l'ennemi et le projectile avance tout droit verticalement.

3.6 Recharge des projectiles

Toutes les seconde le joueur récupère un projectile automatiquement et le maximum de projectile est de 7. La recharge s'affiche en haut à gauche.

Munitions : 4/7

Munitions : 7/7

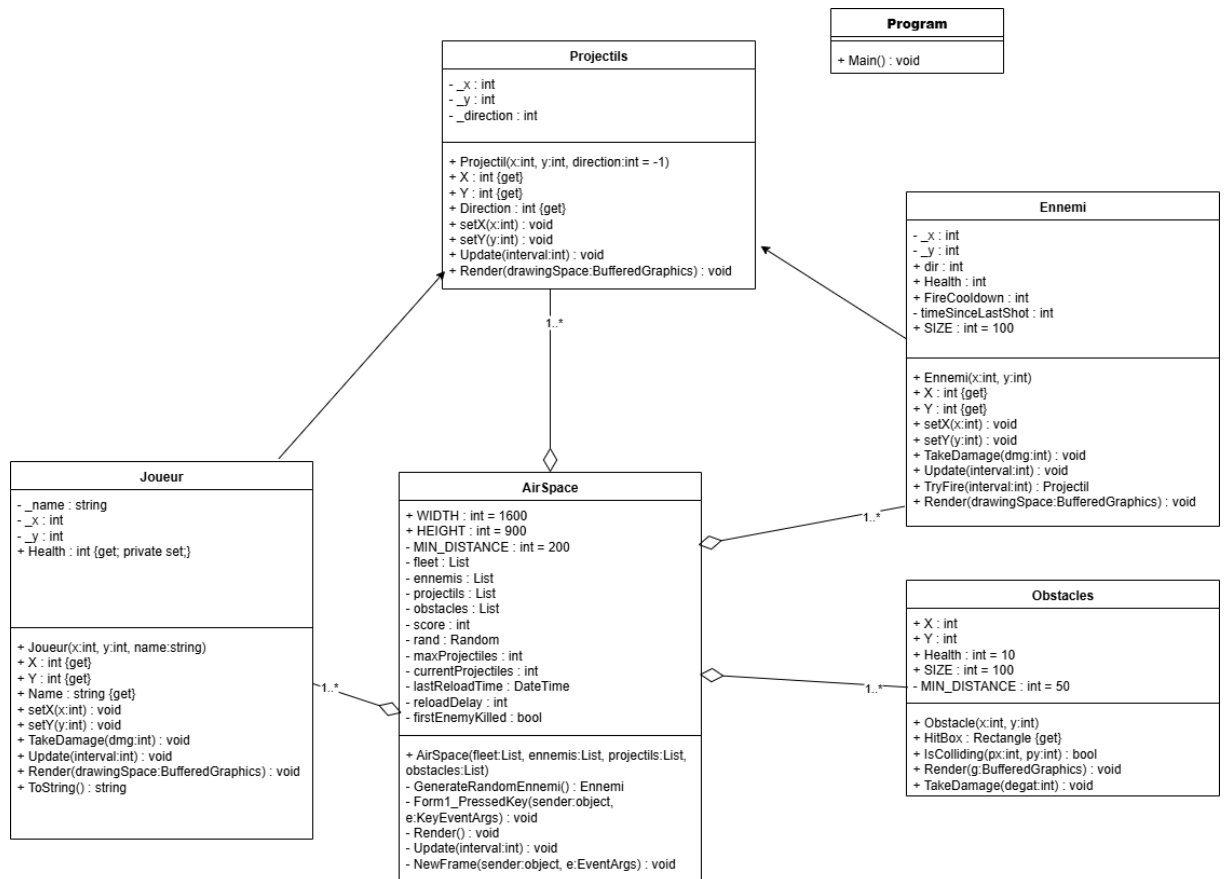
3.7 Score

Quand le joueur a éliminé un ennemi le score augment de 100 pour chaque ennemi éliminer. Le score s'affiche en haut à gauche.

Score : 100

4 CONCEPTION

4.1 UML



4.1.1 Explication utile du diagramme

Le Health pour le joueur et les ennemis sert à avoir la quantité de projectiles qu'ils peuvent encaisser.

Le _direction dans projectiles sert à faire aller le projectile de bas en haut ou de haut en bas

Le maxProjectiles dans AirSpace sert à montrer le maximum de projectiles.

Le currentProjectiles dans AirSpace sert à montrer la quantité de projectiles du joueur.

Le lastReloadTime sert à avoir la date du dernier tir tiré par chaque ennemi.

Le reloadDelay sert à dire le temps que l'ennemi met pour faire un tir.

Le firstEnemyKilled sert à savoir si le premier ennemi est mort pour faire apparaître les autres ennemis.

5 USAGE DE L'IA

Pour avoir la distance entre les obstacles

```

int attempts = 0;
for (int i = 0; i < 5; i++)
{
    bool validPosition = false;

```

```
int x = 0, y = 0;

while (!validPosition && attempts < 100)
{
    attempts++;
    x = rand.Next(50, WIDTH - 100);
    y = rand.Next(200, HEIGHT - 200);

    validPosition = true;
    foreach (var obs in this.obstacles)
    {
        double distance = Math.Sqrt(Math.Pow(x - obs.X, 2) + Math.Pow(y - obs.Y, 2));
        if (distance < Obstacle.SIZE + MIN_DISTANCE)
        {
            validPosition = false;
            break;
        }
    }
}

if (validPosition)
    this.obstacles.Add(new Obstacle(x, y));
}
```