

Projet Spicy Invader



Scerri Yann – FIN1
ETML(Vennes)
2 trimestres
P.Chenau

Table des matières

| | |
|--|----------|
| 1 INTRODUCTION | 3 |
| 2 PLANIFICATION INITIALE | 4 |
| 3 DIAGRAMME DE CLASSE..... | 5 |
| 4 LISTE DES TESTS UNITAIRES | 6 |
| 4.1 TESTS UNITAIRES POUR LA MÉTHODE ISALIVE() DE LA CLASSE SPACESHIP | 6 |
| 4.2 TEST UNITAIRE POUR LA MÉTHODE TAKEDAMAGE() DE LA CLASSE BUNKER | 6 |
| 4.3 TEST UNITAIRE POUR LA MÉTHODE UPDATEMISSILE() DE LA CLASSE MISSILE..... | 7 |
| 5 DOCUMENTATION DOXYGEN | 7 |
| 6 MODE D'EMPLOI DU JEU | 8 |
| 6.1 MENU DU JEU | 8 |
| 6.2 ECRAN DE JEU | 9 |
| 6.3 ECRAN DE VICTOIRE | 10 |
| 6.4 ECRAN DE DÉFAITE | 10 |

1 INTRODUCTION

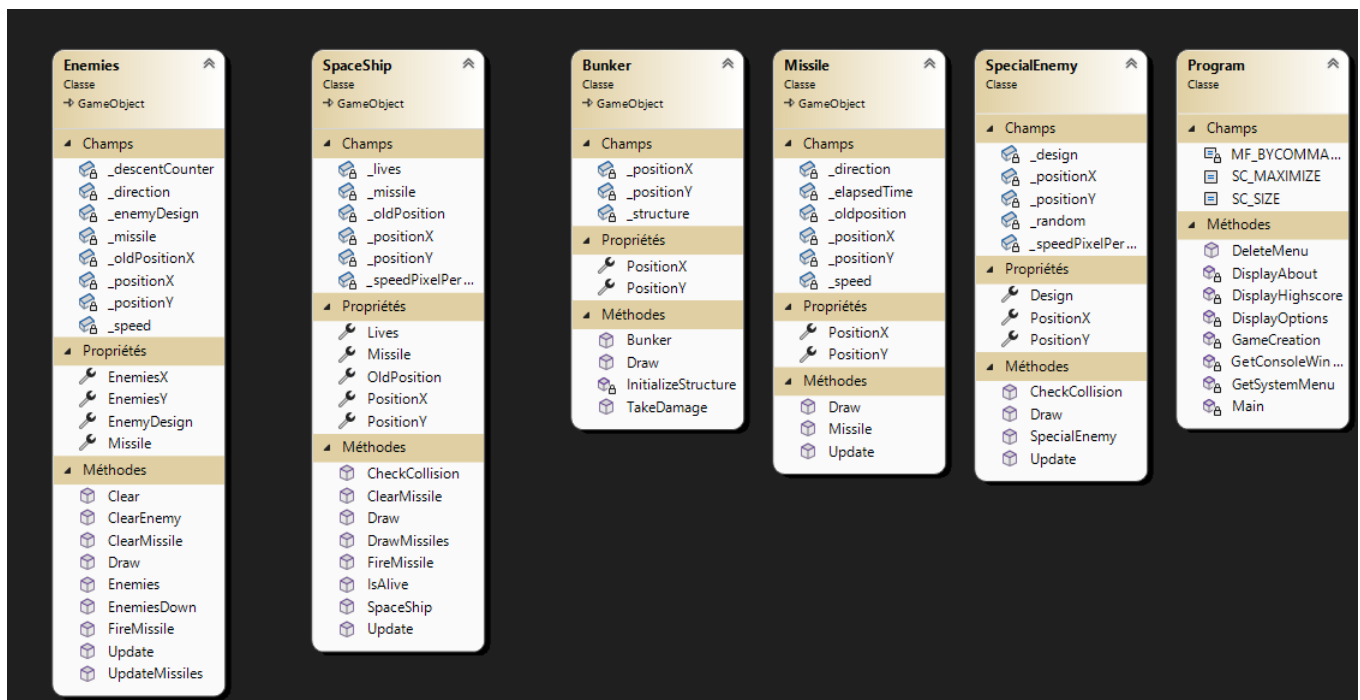
Dans le cadre de notre parcours dans le module ICT226 nous avons entrepris la réalisation d'un projet emblématique : un jeu Space Invader. Ce jeu emblématique, initialement conçu en 1978, est devenu un symbole de l'industrie du jeu vidéo et un pilier du genre des jeux d'arcade.

Notre objectif était de mettre en pratique les principes fondamentaux de la programmation orientée objet que nous avons appris tout au long de notre formation. En utilisant le langage de programmation C#, nous avons cherché à créer une version du jeu fidèle à l'original tout en exploitant les avantages offerts par les concepts de la programmation orientée objet.

2 PLANIFICATION INITIALE

Lors du démarrage du projet Spicy Invader, il nous a été fourni comme moyen d'aide un mode d'emploi de réalisation d'un jeu similaire en application Windows Forms. J'ai donc initialement planifié mon projet en suivant la marche à suivre suivante : [Space Invaders - Piste Verte — Documentation Programmation Orientée Objet 1 \(esiee.fr\)](#). Cependant, les étapes de la marche à suivre s'éloignait assez rapidement d'une version console et se complexifiaient aussi. J'ai néanmoins pu garder le nommage des classes de la piste verte pour mon projet et je les ai conservées jusqu'à la fin du projet tout en ajoutant quelques classes supplémentaires lorsque cela était nécessaire.

3 DIAGRAMME DE CLASSE



4 LISTE DES TESTS UNITAIRES

4.1 Tests unitaires pour la méthode IsAlive() de la classe SpaceShip

◆ IsAlive1()

```
void SpicyInvaderTests.UnitTest1.IsAlive1 ( )
```

méthode de test 1 de la méthode IsAlive de la classe SpaceShip

Definition at line 115 of file [UnitTest1.cs](#).

◆ IsAliveTest2()

```
void SpicyInvaderTests.UnitTest1.IsAliveTest2 ( )
```

méthode de test 2 de la méthode IsAlive de la classe SpaceShip

Definition at line 130 of file [UnitTest1.cs](#).

◆ IsAliveTest3()

```
void SpicyInvaderTests.UnitTest1.IsAliveTest3 ( )
```

Definition at line 143 of file [UnitTest1.cs](#).

Ces 3 méthodes permettent de tester si la méthode IsAlive() analyse correctement les vies du vaisseau. 3 tests ont été effectués, chacun avec une valeur du nombre de vies différentes

4.2 Test unitaire pour la méthode TakeDamage() de la classe Bunker

◆ BunkerDamageTest()

```
void SpicyInvaderTests.UnitTest1.BunkerDamageTest ( )
```

Méthode de test pour la méthode TakeDamage de la classe Bunker.

Definition at line 15 of file [UnitTest1.cs](#).

Cette méthode permet de tester la méthode TakeDamage afin de savoir si les missiles détruisent correctement les bunkers (1 pixel à la fois).

4.3 Test unitaire pour la méthode UpdateMissile() de la classe Missile

◆ UpdateMissileTest()

```
void SpicyInvaderTests.UnitTest1.UpdateMissileTest ( )
```

méthode de test de la méthode Update Missile de la classe Missile

Definition at line 71 of file [UnitTest1.cs](#).

Cette méthode de test permet de vérifier que la méthode UpdateMissile met correctement à jour les missiles.

5 DOCUMENTATION DOXYGEN

La documentation Doxygen a été créée via le Wizard.exe et est disponible dans l'archive .zip du rendu final

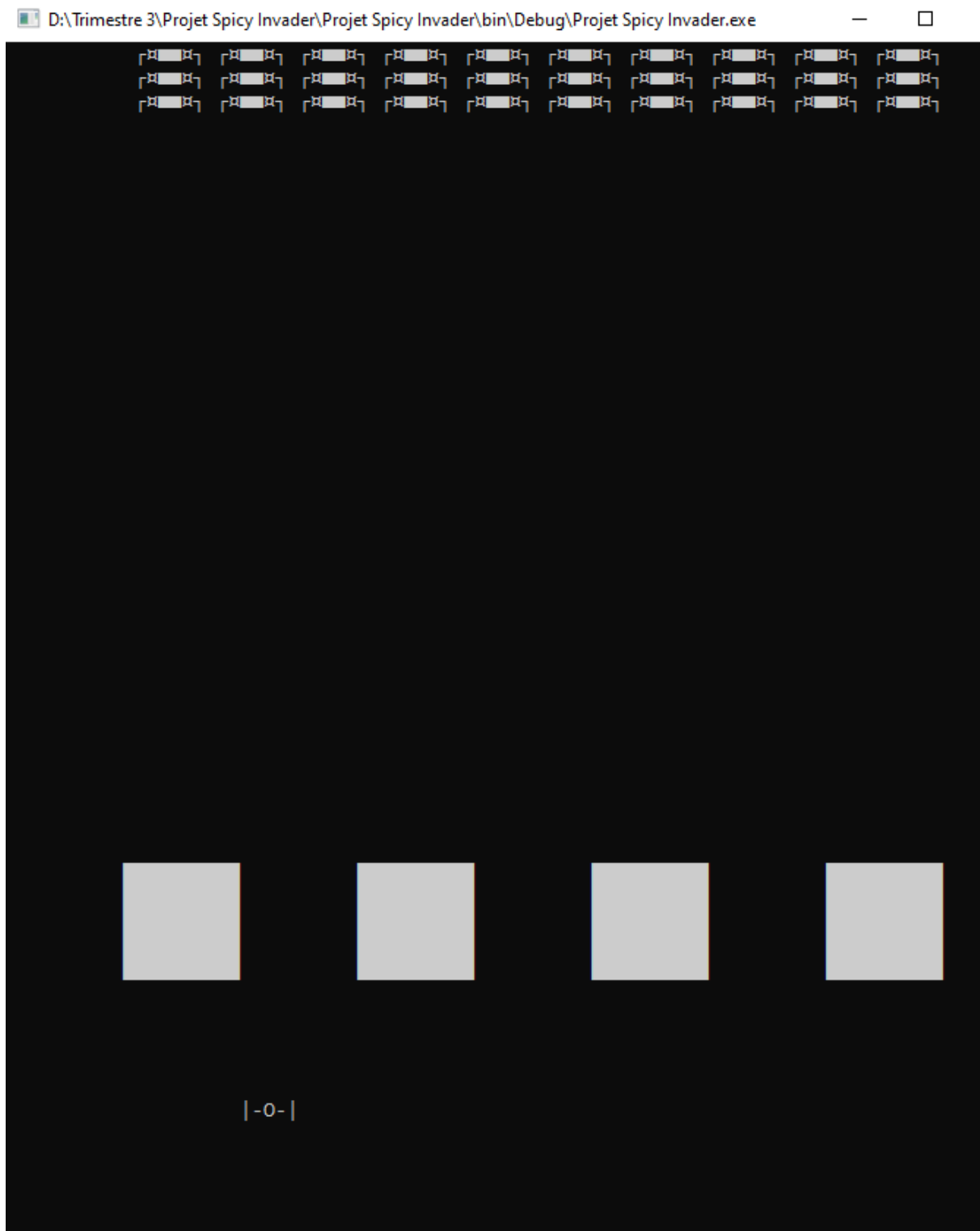
6 MODE D'EMPLOI DU JEU

6.1 Menu du jeu



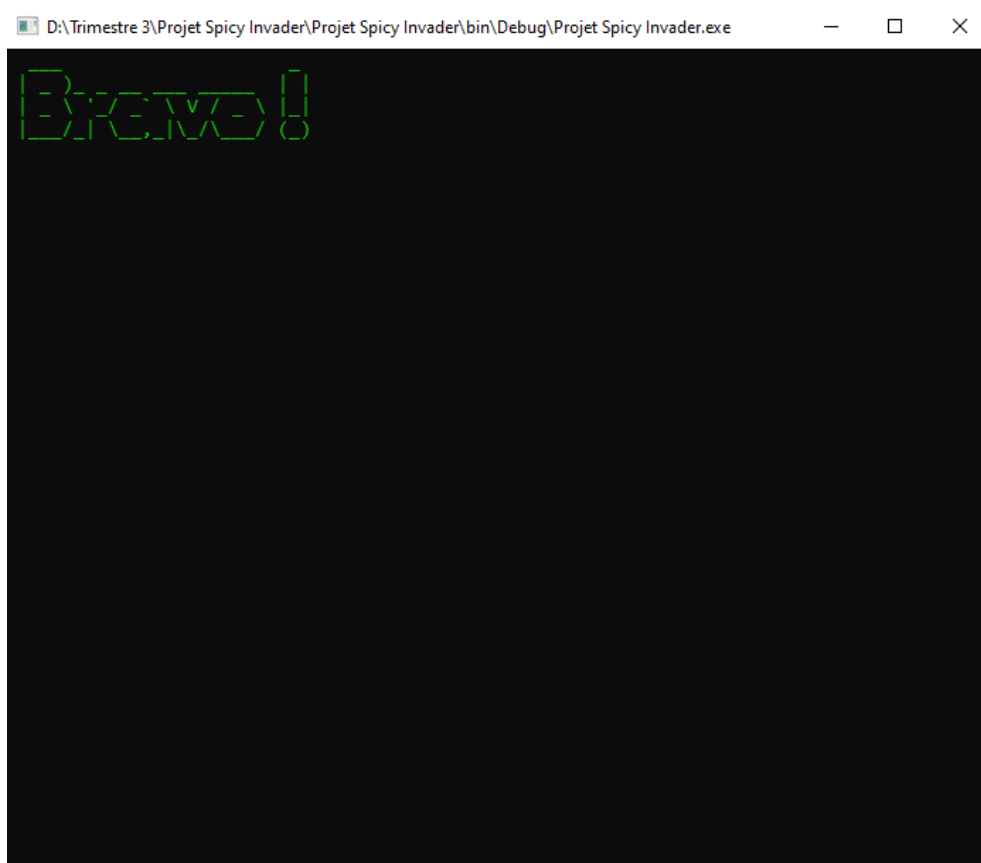
Le menu propose les options de naviguer entre les différentes options du programme.

6.2 Ecran de jeu



L'écran de jeu se présente ainsi. Le joueur peut tirer sur les ennemis et gagne la partie s'il parvient à éliminer tous les ennemis avant de se faire atteindre. Si le joueur se fait toucher 3 fois par les missiles ennemis, la partie s'arrête.

6.3 Ecran de victoire



6.4 Ecran de défaite

