

Spicy-invader POO

Cyril Napoleone





Spicy-invader POO	1
Cyril Napoleone	1
.....	1
1. Introduction	3
2. Schémas et description des éléments principaux.....	4
2.1. Automatique du code.....	4
2.2. Interactions entre les classes.....	5
2.2.1. Program	5
2.2.2. Game	5
2.2.3. Menu	5
2.2.4. SQL.....	5
2.3. Paramètres	6
2.3.1. Paramètres joueur	6
2.3.1.1. Position joueur	6
2.3.1.2. Pseudo.....	6
2.3.1.3. Score.....	6
2.3.2. Paramètres ennemis.....	6
2.3.2.1. Position ennemie	6
2.3.2.2. Délai d'apparition	6
2.3.2.3. Nombres d'ennemis	6
2.3.3. Paramètres projectiles	6
2.3.3.1. Position projectile.....	6
2.3.3.2. Etat projectile	6
2.3.3.3. Nombre maximum.....	7
3. Cas particuliers	7
4. Conclusion.....	7

1. Introduction

Dans le cadre de ce projet C# nous devons produire un jeu imitation de Space Invaders nommé Spicy Invaders en programmation orienté objet, l'imitation se basait sur un programme exemple présenté au préalable sur lequel nous pouvions nous baser vaguement. J'ai décidé de créer un affichage joli et travaillé pour le menu et le classement, une option jeu, classement, quitter.

Le principe d'une partie sur mon jeu consiste en 10 ennemis tombant verticalement, le joueur doit les tuer et tout se joue sur l'endurance car au fur et à mesure que les ennemis meurent, ils respawn aléatoirement.

Nous devons aussi créer une documentation automatique à l'aide de docfx et l'usage de IceScrub pour avoir un affichage final de notre avancée sur icescrum que l'on peut voir ci-dessous.

story	terminée_le	tâche	temps_estimé	temps_effectif	Totals
Afficher le vaisseau du joueur	11 sept. 2023	Ecrire vaisseau "X" au lancement du jeu	1	0.5	1
		Position vaisseau	1	0.5	1
Afficher un menu	04 sept. 2023	Ecrire les différentes options	0.5	1	1
Déplacer le vaisseau	11 sept. 2023	Bloquer le vaisseau à gauche et droite	0.5	0.2	0
		Ecouter le clavier	0.5	0.5	1
		Mapper le clavier avec le mouvement du vaisseau	0.5	1	1
Ennemis	03 nov. 2023	Système apparition ennemis	1	1	1
		apparition à intervalle différent	0.5	0.5	1
		déplacement verticalement vers le vaisseau	0.5	0.5	1
		position des ennemis aléatoires	0.5	0.5	1
Menus effectifs	04 sept. 2023	Déplacement avec voyant coloré	1	1	1
	05 sept. 2023	Accès nouvelle page	0.5	1	1
Système de points	05 nov. 2023	Disparition ennemi au touché	1	0.5	1
		incréméntation points au touché ennemi	0.5	0.5	1
Système de tir	03 nov. 2023	déplacement projectiles	0.5	1	1
	25 sept. 2023	Génération projectile	2	0.5	1
Victoire/Défaite	03 nov. 2023	implémenter règle défaite	1	0.5	1
	05 nov. 2023	Retenue des points	0.5	0.5	1
Totals					12



2. Schémas et description des éléments principaux

2.1. Automatique du code

D

Class Game

Namespace: [Spicy_Invasion_Core](#)
Assembly: Spicy_Invasion_Core.dll

```
public class Game
```

Inheritance
[object](#) < [Game](#)

Inherited Members
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.GetHashCode\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ReferenceEquals\(object, object\)](#), [object.ToString\(\)](#)

Fields

enemyCount

```
public int enemyCount
```

Field Value
[int](#)

playerX

```
public int playerX
```

Field Value
[int](#)

playerY

```
public int playerY
```

Field Value
[int](#)

Properties

pseudo

```
public string pseudo { get; set; }
```

Property Value
[string](#)

score

```
public int score { get; set; }
```

Property Value
[int](#)

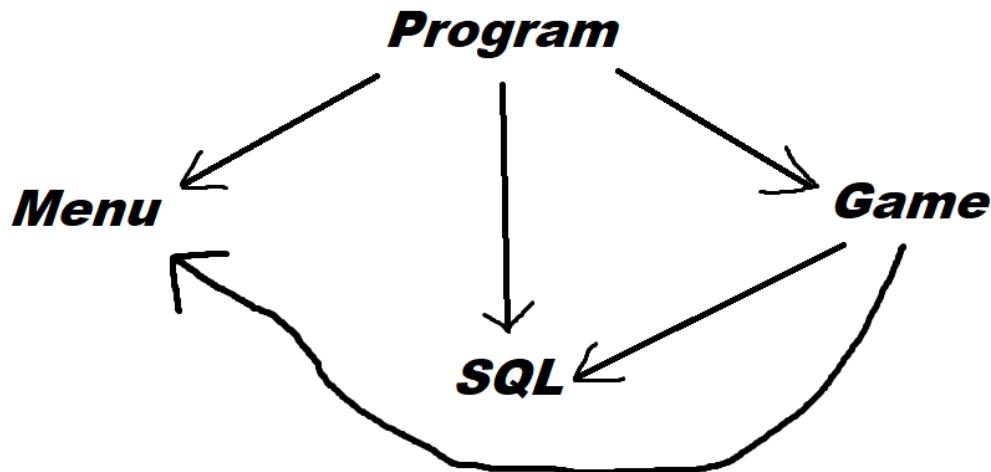
Methods

Play()

```
public void Play()
```

*J'ai fait la procédure Docfx, tout a fonctionné mais malheureusement, seulement la class Game a été documentée. J'ai réessayé multiple fois mais rien ne change.

2.2. Interactions entre les classes



2.2.1. Program

La class Program est la classe principale, la majorité des choses partent d'ici, depuis program, la classe Menu est utilisée pour l'affichage des options du menu, la classe Game est utilisée pour le lancement de la partie au click de l'option "jouer", et SQL pour l'option classement.

2.2.2. Game

Game est la class de jeu, elle permet de faire fonctionner la partie du joueur. Elle est reliée à SQL pour l'enregistrement des données à la fin de la partie (pseudo et score), et après ça la connexion au menu parce que le joueur est redirectionné vers le menu car la partie est finie.

2.2.3. Menu

La class Menu est la classe qui fait fonctionner le style et les options du menu principal du program donc jouer, classement et quitter. Elle est reliée à aucune classe car c'est Program qui s'occupe de faire fonctionner le menu avec les redirections vers les pages

2.2.4. SQL

La class SQL gère les requêtes SQL nécessaires dans l'option classement pour afficher les joueurs et leur score et à la fin d'une partie pour insérer les données du joueur (donc liaison depuis Program et depuis Game)

2.3. Paramètres

Je vais parler des paramètres de jeu important au bon fonctionnement d'un partie.

2.3.1. Paramètres joueur

2.3.1.1. Position joueur

Dans playerX et playerY sont stockés les informations de positions du vaisseau (X, Y). Est utilisé pour le déplacement et le démarrage du jeu (position initial, mouvement clavier, etc...)

2.3.1.2. Pseudo

La variable pseudo est la variable que le joueur entre à la fin de sa partie et qui sera enregistrée dans la base de données pour l'identifier lui et son score

2.3.1.3. Score

Le score est la variable qui représente le résultat de la partie (ennemi touché = +125 points) elle est, comme ci-dessus) enregistré à la fin de la partie puis traité par la classe SQL, notamment pour le classement.

2.3.2. Paramètres ennemis

2.3.2.1. Position ennemie

Comme pour le joueur, chaque ennemi à sa position X et Y propre qui est utilisé pour les mouvements et pour la position en cas de touche.

2.3.2.2. Délai d'apparition

Pour que les apparitions ennemies soient random, un délai d'apparition aléatoire est créé pour chaque ennemi ce qui rend la partie dynamique est moins répétitive.

2.3.2.3. Nombres d'ennemis

Il y a un nombre maximal d'ennemi à la fois que j'ai personnellement défini à 10 mais qui peut être changé dans le code. 10 car c'est le chiffre demandé par le cahier des charges.

2.3.3. Paramètres projectiles

2.3.3.1. Position projectile

Comme tous les éléments précédents, lui aussi à sa position X et Y, qui change pour que le projectile avance de manière constante vers les ennemis.

2.3.3.2. Etat projectile

Sa variable isShooting définit si le projectile est en trajet. Ça permet de gérer ses projectiles comme on veut et choisir quand elles ne doivent plus être là.

2.3.3.3. Nombre maximum

Comme pour les ennemis, il peut qu'y avoir un nombre maximal de projectiles en simultan  sur l' cran. 10 pour  viter tout abus et bug. Bien s r une fois qu'un projectile arrive en haut ou touche, il est r utilisable.

3. Cas particuliers

Le jeu prend en compte la gestion des erreurs et de certaines exceptions, avec notamment des messages d'erreurs ou des limites pr vue   l'avance.

Il y'a des fonctionnalit s sp cifiques tel que la gestion du menu, les mouvements du vaisseau, l'enregistrement du score/pseudo ou encore le syst me d'ennemis.

Durant le projet, j'ai rencontr  certaines probl matiques tel que l'op rationnalit  des requ tes sql dans ce contexte c#, au tout d but aussi le d placement du vaisseau de mani re fluide. Mais le plus gros probl me que j'ai rencontr  est s rement le fait de pouvoir r aliser plusieurs actions en m me temps (bouger vaisseau, bouger ennemis, projectiles en cours, etc.. Et tout  a simultan ment) mais j'ai r ussi gr ce   un syst me de Counter que chaque  l ment utilise diff remment.

4. Conclusion

Ce projet C# a  t  une exp rience enrichissante en programmation orient e objet, bien que complexe.

Malgr  quelques d fis comme dit plus t t. L'interaction entre les classes Program, Menu, Game et SQL a permis de mettre en place un syst me de jeu complet.

En fin de compte, ce projet m'a permis de d velopper mes comp tences en programmation orient e objet, de g rer des fonctionnalit s complexes telles que le menu, les mouvements du vaisseau, l'enregistrement du score et la gestion des ennemis. J'ai  galement appris   g rer plusieurs actions simultan ment gr ce   un syst me de compteur. Dans l'ensemble, Spicy Invaders est un jeu amusant qui repr sente une  tape importante dans mon parcours de d veloppement logiciel.