|  |
| --- |
| ETML |
| Space Invaders |
| P\_OO |

|  |
| --- |
| Abiram Muthulingam  04/09/2024 |

Table des matières

[1 Analyse préliminaire 2](#_Toc181006527)

[1.1 Introduction 2](#_Toc181006528)

[1.2 Objectifs 2](#_Toc181006529)

[1.3 Gestion de projet 3](#_Toc181006530)

[2 Analyse / Conception 3](#_Toc181006531)

[2.1 Gameplay 3](#_Toc181006532)

[2.2 UX 4](#_Toc181006533)

[2.3 Concept 4](#_Toc181006534)

[2.4 Analyse fonctionnelle 4](#_Toc181006535)

[2.4.1 Création du Menu principal 4](#_Toc181006536)

[2.4.2 Déplacement gauche ou droite 4](#_Toc181006537)

[2.4.3 Capacité de tir 5](#_Toc181006538)

[2.4.4 Nombre de vies 5](#_Toc181006539)

[2.4.5 Tir 5](#_Toc181006540)

[2.4.6 déplacement des ennemis 5](#_Toc181006541)

[2.4.7 Nombre de vies des ennemies 6](#_Toc181006542)

[2.4.8 Lancement du fichier exécutable 6](#_Toc181006543)

[2.4.9 Déplacement de l'obstacle 6](#_Toc181006544)

[2.4.10 Durée de vie de l'obstacle 6](#_Toc181006545)

[2.4.11 Ajouts d'un 2ème obstacle 6](#_Toc181006546)

[2.5 Stratégie de test 7](#_Toc181006547)

[3 Réalisation 7](#_Toc181006548)

[3.1 Points de design spécifiques 7](#_Toc181006549)

[*3.1.1* *…* 8](#_Toc181006550)

[*3.1.2* *…* 8](#_Toc181006551)

[*3.1.3* *…* 8](#_Toc181006552)

[3.2 Déroulement 8](#_Toc181006553)

[3.3 Mise en place de l’environnement de travail 8](#_Toc181006554)

[3.4 Description des tests effectués 8](#_Toc181006555)

[3.5 Erreurs restantes 8](#_Toc181006556)

[4 Conclusions 9](#_Toc181006557)

[5 Annexes 10](#_Toc181006558)

[5.1 Manuel de référence 10](#_Toc181006559)

[5.2 Journal de travail 10](#_Toc181006560)

*NOTE L’INTENTION DES UTILISATEURS DE CE CANEVAS :*  
*Toutes les parties en italiques sont là pour aider à comprendre ce qu’il faut mettre dans cette partie du document. Elles n’ont donc aucune raison d’être dans le document final.*

*De plus, en fonction du type de projet, il est tout à fait possible que certains chapitres ou paragraphes n’aient aucun sens. Dans ce cas il est recommandé de les retirer du document pour éviter de l’alourdir inutilement.*

# Analyse préliminaire

## Introduction

Le but de ce projet, est de réaliser un jeu de tirs en programmation orienté objet en C#. Le jeu est fait en 3 projet différent (UX, OO, DB). Dans ce projet, nous implémentons seulement la partie du gameplay et du menu, qui permettra à l’utilisateur de jouer. Durant le projet, l’outil IceScrum est utiliser pour la gestion du projet, ainsi qu’un journal de travail et d’un rapport expliquant les détails du jeu.

*Ce chapitre décrit brièvement le projet, le cadre dans lequel il est réalisé, les raisons de ce choix et ce qu'il peut apporter à l'élève ou à l'école. Il n'est pas nécessaire de rentrer dans les détails (ceux-ci seront abordés plus loin) mais cela doit être aussi clair et complet que possible (idées de solutions). Ce chapitre contient également l'inventaire et la description des travaux qui auraient déjà été effectués pour ce projet.*

*Ces éléments peuvent être repris des spécifications de départ.*

## Objectifs

L’objectif de ce projet, est de créer un jeu de tirs en programmation orienté objet en C#, avec comme conception des ennemis, le joueur (de l’utilisateur), ainsi que des obstacles qui serviront de protection. A la fin de ce projet, il doit y a avoir un jeu accessible sans bug.

*Ce chapitre énumère les objectifs du projet. L'atteinte ou non de ceux-ci devra pouvoir être contrôlée à la fin du projet. Les objectifs pourront éventuellement être revus après l'analyse.*

*Ces éléments peuvent être repris des spécifications de départ.*

## Gestion de projet

L’outil IceScrum, est utilisé durant le projet, afin de pouvoir gérer la gestion de projet.

IceScrum permet de planifier les tâches à effectuer, et aussi d’avoir des tests pour vérifier que le jeu est fonctionnel. Un journal de travail est mis à jour à chaque semaine, expliquant le travail effectuer.

***Ce chapitre décrit la méthode de gestion de projet utilisée, ainsi que les éventuelles particularités requises par le contexte et/ou le chef de projet***

# Analyse / Conception

## Gameplay

Le joueur :

Le joueur contrôlera un vaisseau qui se trouve en bas de l’écran et peut se déplacer à gauche et à droite (flèche gauche, droite ou lettre A, D), mais ne peut pas sortir du cadre du jeu. Le vaisseau n’a pas de munition limitée et peut donc tirer à volonté verticalement. Si le vaisseau se fait toucher par un ennemi, la partie est terminée.

Les ennemis :

Les ennemis apparaissent aléatoirement depuis le haut de l’écran et s’approche du vaisseau pour le toucher afin de le détruire. Il y a 2 types d’ennemis (petit et grand), les petits ont seulement 1 point de vie et les grands ont 3 points de vie.

Les obstacles :

Les obstacles servent de protection est empêchent les tirs du vaisseau (quand le vaisseau tirs) et de l’ennemi, mais quand le vaisseau ne tire pas l’obstacle se transforme pour pouvoir tirer en continu (pour aider le joueur) afin de toucher des ennemis. Les obstacles sont destructibles seulement sous transformation de tirs, en recevant 5 tirs (phase de test). Les obstacles se déplacent horizontalement à gauche et à droite.

Niveau 2 :

Le niveau 2 sera plus difficile que le niveau 1 (niveau de base). Les ennemis sont plus rapides et peuvent aussi tirer. Le vaisseau du joueur aura un nombre limité de munition et son projectile est un missile qui fait de dégâts aux alentours, mais qui a un délai plus long que la normal.

* *Le joueur*
* *Les ennemis*
* *Les déplacements*
* *Les niveaux*
* *Le tir*
* *La gestion des vies*
* *…*

## UX

Conception centrée utilisateur :

Création de 2 persona, âgés de 18 et 12 ans pour diverses raisons de jouer au jeu.

Persona 1 : Lisa, âgée de 18 ans, voulant réussir à jouer pour le fun dans un jeu pas très compliqué.

Persona 2 : Marius est un jeune homme de 12 ans, qui a pour but de persévéré dans les jeux, et de devenir le meilleur.

Choix de la palette graphique :

FF1E1E A01A1A

Ces couleurs rouges sont utilisées pour les titres, les grands textes et pour la sélection, pour que ce soit bien lisible.

E6EED6 DEC222

Ces couleurs sont utilisées pour le texte se trouvant dans la partie high score.

0D0C0C FFFFFF

Le noir est utilisé pour le background, le blanc est utilisé sur le texte pour les choix de sélection.

Accessibilité :

Le jeu peut être utilisé seulement avec les touches de clavier et à une seule main. Pour les personnes qui sont malvoyant, il faudrait qu’on ajoute un bouton hautparleur pour qu’il puisse lire à haute voix le texte.

Choix effectués :

## Concept

* *Diagramme de classe*
* *Diagramme(s) d’état*

## Analyse fonctionnelle

### Création du Menu principal

|  |
| --- |
| En tant qu'utilisateur, je souhaiterais avoir un menu, afin de pouvoir accéder au jeu |
| Tests d'acceptance:   |  |  | | --- | --- | | Naviguer entre les boutons verticalement | Dans le menu Quand je presse flèche du haut ou flèche du bas je peux choisir de jouer, changer de niveau ou de quitter le programme | | En appuyant enter | Dans le menu Quand j'appuie sur la touche ENTER je valide la proposition dont j'ai choisi et cela me lance le jeu, choix du niveau ou l'arrêt du programme | |

### Déplacement gauche ou droite

|  |
| --- |
| En tant qu'utilisateur, je souhaiterais pouvoir bouger mon vaisseau de gauche à droite. |
| Tests d'acceptance:   |  |  | | --- | --- | | déplacement à l'horizontal | En cours de partie Quand je presse flèche gauche ou flèche droite, le vaisseau se déplace dans la direction de la flèche | | ne sort pas du cadre du jeu | En cours de partie quand je déplace le vaisseau jusqu'au bord, il est bloqué | | déplacement à l'horizontal | En cours de partie Quand je presse A ou D, le vaisseau se déplace à gauche ou à droite | | Maintenir la touche pour bouger à gauche ou droite | En cours de partie Quand je maintiens la touche pour bouger à gauche ou à droite le vaisseau avance en continu dans la direction choisi, une fois la touche relâchée le vaisseau s'arrête. | |

### Capacité de tir

|  |
| --- |
| En tant qu'utilisateur, je souhaiterais savoir les capacité de tirs(direction des tirs, munitions) du vaisseau afin de pouvoir l'utiliser contre les ennemies. |
| Tests d'acceptance:   |  |  | | --- | --- | | Tirer est vérifier le cooldown | En cours de partie quand un missile à été tiré Je peux pas en tirer un deuxieme avant qu'il à été disparu | |

### Nombre de vies

|  |
| --- |
| En tant qu'utilisateur je voudrais savoir le nombre de vie du vaisseau, afin de savoir quand sera la fin de la partie |
| Tests d'acceptance:   |  |  | | --- | --- | | Mourir 1 fois | En cours de partie Quand je me fais toucher par un ennemi le vaisseau est détruit et la partie est terminée. | |

### Tir

|  |
| --- |
| En tant qu'utilisateur, je souhaiterais savoir la direction des tirs, afin de pouvoir tirer sur les ennemis. |
| Tests d'acceptance:   |  |  | | --- | --- | | Le vaisseau tirs | En cours de partie Quand je presse sur espace, le vaisseau tire verticalement | | déplacement du missile | En cours de partie une fois que j'ai tiré, le missile monte tous seul | | disparition du missile | En cours de partie quand le missile arrive en haut de l'écran le missile disparait. | |

### déplacement des ennemis

|  |
| --- |
| En tant qu'utilisateur, je souhaiterais savoir par où viennent les ennemis, afin de pouvoir se protéger. |
| Tests d'acceptance:   |  |  | | --- | --- | | Déplacement vertical | En cours de partie Quand je joue les ennemis apparaissent depuis le haut de l'écran et va jusqu'en bas | | Les ennemis ne s'entremêle pas | En cours de partie Quand les ennemis apparaissent Les ennemis ne se touche pas entre eux (il n'y a pas de collision) | |

### Nombre de vies des ennemies

|  |
| --- |
| En tant qu'utilisateur, je souhaiterais savoir le nombre de point de vie de chaques ennemies, afin de pouvoir les contré. |
| Tests d'acceptance:   |  |  | | --- | --- | | Tirs sur les ennemis commun | En cours de partie Quand je touche un ennemi avec mon projectile, Il disparait de l'écran. | | tirer sur les ennemis rare | En cours de partie Quand je touche 3 fois un ennemi de type rare, il disparait de l'écran | |

### Lancement du fichier exécutable

|  |
| --- |
| En tant qu'utilisateur, je souhaiterais lancer un fichier exécutable, afin de pouvoir accéder au menu du jeu. |
| Tests d'acceptance:   |  |  | | --- | --- | | Lancer le fichier exécutable | Lancer le fichier et voir un menu qui s'affiche | |

### Déplacement de l'obstacle

|  |
| --- |
| En tant qu'utilisateur je souhaiterais savoir le déplacement de l'obstacle, afin de pouvoir me protéger. |
| Tests d'acceptance:   |  |  | | --- | --- | | Déplacement à l'horizontal | En cours de partie automatiquement, l'obstacle se déplace horizontalement (droite, gauche) | | ne sort pas du cadre du jeu | En cours de partie Quand l'obstacle s'approche du bord de l'écran, il repars dans le sens opposé | |

### Durée de vie de l'obstacle

|  |
| --- |
| En tant qu'utilisateur je souhaiterais savoir à quel moment l'obstacle disparait, afin de pouvoir l'utilisais correctement. |
| Tests d'acceptance:   |  |  | | --- | --- | | Disparition de l'obstacle | En cours de partie, Quand l'obstacle reçois 5 projectiles, il disparait de l'écran (il est détruit). | | Résistance de l'obstacle | En cours de partie, Quand l'obstacle reçois moins de 5 projectiles, Il est toujours en fonctionnement | |

### Ajouts d'un 2ème obstacle

|  |
| --- |
| En tant qu'utilisateur, je souhaiterais avoir plusieurs obstacles afin de pouvoir me protéger |
| Tests d'acceptance:   |  |  | | --- | --- | | Pas de collision avec l'obstacle 1 | En cours de partie, Quand les 2 obstacles se rapproche, il repars dans le sens opposé | |

***Reprendre le contenu des User Stories d’IceScrum : Story + tests d’acceptance (avec IceTools) + maquettes***

## Stratégie de test

Création des tests expliqué clairement, en trouvant des problèmes de fonctionnalités en se mettant à la place d’un utilisateur. Effectuer ces tests une fois qu’une fonctionnalité est terminée, et vérifier qu’aucun problème persiste.

***Décrire quels sont les MOYENS utilisés pour faire les tests, ne pas décrire les tests à effectuer !!!***

***Décrire l’environnement dans lequel se fait la sprint review***

*Décrire la stratégie globale de test :*

* *types de des tests et ordre dans lequel ils seront effectués.*
* *les moyens à mettre en œuvre.*
* *couverture des tests (tests exhaustifs ou non, si non, pourquoi ?).*
* *données de test à prévoir (données réelles ?)* ***et comment elles seront mises en place****.*
* *les testeurs extérieurs éventuels.*

# Réalisation

## Points de design spécifiques

***Ce chapitre est constitué de plusieurs sous-chapitre.***

***Chaque sous-chapitre explique un point de design technique particulier, quelque chose que vous avez dû inventer pour répondre au besoin et qui ne peut pas s’expliquer par de simples commentaires dans le code.***

***Il s’agit d’explications techniques sur le fonctionnement du système. Les explications sont appuyées par des diagrammes, ou de très brefs éléments de code.***

***NE PAS mettre ici des pratiques usuelles que tout professionnel de la branche connaît déjà. Par exemple, n’EXPLIQUEZ PAS ICI CE QU’EST LE PATTERN MVC.***

***Exemple (simplifié à l’extrême) : Protection contre des formulaires mal intentionnés ou modifiés***

* ***Au moment de générer le formulaire, le script php :***
  + ***Concatène les noms de tous les champs contenus dans le formulaire***
  + ***Calcule un hash SHA256 de la chaîne obtenue***
  + ***Ajoute un input nommé « CSRF » de type hidden dans le form***
* ***A la réception du POST du fromulaire***
  + ***Concatène les noms des indices de $\_POST***
  + ***Calcule un hash SHA256 de la chaîne obtenue***
  + ***Vérifie que la valeur du champ CSRF correspond***

### *…*

### *…*

### *…*

## Déroulement

***Résumer comment s’est passé la réalisation de chaque story, ses difficultés, les alternatives envisagées mais rejetées, ses surprises, …***

## Mise en place de l’environnement de travail

* ***Comment accéder au code source***
* *la liste de tous les fichiers et une rapide description de leur contenu (des noms qui parlent !)*
* *les versions des systèmes d'exploitation et des outils logiciels*
* *la description exacte du matériel*

***Ce chapitre décrit précisément comment un employé qualifié peut recréer l’environnement dans lequel vous avez effectué ce travail***

## Description des tests effectués

***Reprendre les tests d’acceptance d’IceScrum au moyen de la feuille ad hoc d’IceTools***

## Erreurs restantes

*S'il reste encore des erreurs :*

* *Description détaillée*
* *Conséquences sur l'utilisation du produit*
* *Actions envisagées ou possibles*

***Reporter la*** [***dette technique***](https://www.premaccess.com/qu-est-ce-que-dette-technique-comment-la-maitriser/#:~:text=La%20dette%20technique%20survient%20quand,de%20plus%20en%20plus%20fr%C3%A9quents.) ***connue. S’appuyer sur la pratique des // TODO***

# Conclusions

*Développez en tous cas les points suivants :*

* *Objectifs atteints / non-atteints*
* *Points positifs / négatifs*
* *Difficultés particulières*
* *Suites possibles pour le projet (évolutions & améliorations)*

# Annexes

## Manuel de référence

***Issu de la génération automatique à partir des commentaires***

## Journal de travail