|  |
| --- |
| Projet Space Invaders |

Pour projet Shoot Me Up avec XCL

Table des matières

[1 Analyse préliminaire 3](#_Toc499021832)

[1.1 Introduction 3](#_Toc499021833)

[1.2 Objectifs 3](#_Toc499021834)

[1.3 Planification initiale 4](#_Toc499021835)

[2 Analyse / Conception 4](#_Toc499021836)

[2.1 Concept 4](#_Toc499021837)

[2.2 Stratégie de test 4](#_Toc499021838)

[2.3 Risques techniques 4](#_Toc499021839)

[2.4 Planification 4](#_Toc499021840)

[2.5 Dossier de conception 5](#_Toc499021841)

[3 Réalisation 5](#_Toc499021842)

[3.1 Dossier de réalisation 5](#_Toc499021843)

[3.2 Description des tests effectués 6](#_Toc499021844)

[3.3 Erreurs restantes 6](#_Toc499021845)

[3.4 Liste des documents fournis 6](#_Toc499021846)

[4 Conclusions 6](#_Toc499021847)

[5 Annexes 7](#_Toc499021848)

[5.1 Résumé du rapport du TPI / version succincte de la documentation 7](#_Toc499021849)

[5.2 Sources – Bibliographie 7](#_Toc499021850)

[5.3 Journal de travail 7](#_Toc499021851)

[5.4 Manuel d'Installation 7](#_Toc499021852)

[5.5 Manuel d'Utilisation 7](#_Toc499021853)

[5.6 Archives du projet 7](#_Toc499021854)

*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 projet demande une application C#. Le contenu s’agit d’un jeu du style Space Invaders qui permet de se déplacer, attaquer, se faire attaquer, pouvoir gagner, etc. Le thème de mon projet personnel c’est un jeu où l’utilisateur joue en tant qu’une poule qui doit abattre des vaisseaux spatiaux afin de remporter une partie. Le but de ce projet c’est de développer nos connaissances acquises lors du module 320, avec les classes, les tests unitaires, l’encapsulation, etc.

## Objectifs

Mes objectifs généraux pour ce projet c’est d’améliorer mes compétences en C# et de mieux comprendre certains concepts. Plus concrètement, j’aimerais pouvoir développer un jeu dont on peut attaquer, pouvoir se déplacer à l’intérieur des limites de l’écran, avoir des ennemis qui attaquent l’utilisateur et finalement avoir plusieurs niveaux afin d’augmenter la complexité du jeu.

## Gestion de projet

La manière dont la gestion de projet a été mise en place a été avec IceScrum et un journal de travail qui se situe dans le repo Github. Sur IceScrum, de nombreux backlogs ont été formés afin de s’assurer que tous les objectifs ont été effectués. Un daily scrum a également été fait à chaque début de séance de projet.

# Analyse / Conception

## Gameplay

* Je joueur : Le modèle du joueur est l’image d’une poule. Le joueur peut se déplacer à l’intérieur des bordures de l’écran en utilisant les touches W, A, S et D, sur l’axe X, mais également sur l’axe Y. En termes du système d’attaque, le joueur peut envoyer des projectiles en utilisant les flèches de direction (Haut, Bas, Gauche, Droite).
* Les ennemis : Il existe plusieurs types d’ennemis dans le jeu, ceux qui faut éliminer se sont les soucoupes volantes. Certaines d’entre elles se font éliminés à l’impact avec le projectile du joueur tandis que d’autres doivent encaisser quelques tirs avant de disparaitre. Tous les ennemis se déplacent sur l’axe X de l’écran. Lorsqu’ils arrivent à la limite des bordures de l’écran, certains d’entre eux se téléportent au début de l’écran tandis que d’autres commencent à se déplacer en sens inverse. Il y a aussi des scarabées en tant que ennemis, ceux-ci ne peuvent pas être éliminés, le joueur dois donc les éviter car s’il entre en contact avec ceux-ci le joueur est éliminé. Cependant les scarabées se déplacent de manière lente comparés aux autres objets.
* Les déplacements : Le système de déplacement est différent pour chaque objet. Le joueur c’est le seul objet dont on peut contrôler manuellement le déplacement, les ennemis se déplacent tous soit vers la gauche ou la droite.
* Les niveaux : Il y a seulement deux niveaux dans le jeu, le premier commence dès l’exécution du jeu et il se termine lorsque toutes les soucoupes ont été éliminées. Durant le premier niveau, lorsque toutes les soucoupes d’un type de soucoupe ont été éliminés, une soucoupe mère viendra déposer de nouveaux ennemis. Lorsque le premier niveau est terminé, un écran disant « Round 2 » va s’afficher, le niveau 2 va donc débuter lorsque l’utilisateur entra « Enter ». Le niveau 2 contient de nouveaux ennemis qui sont plus dur à battre. Une fois que ce niveau est terminé, le joueur remporte la partie. Contrairement au premier round, lorsque le joueur meurs dans le deuxième, il a la possibilité de recommencer celui-ci en faisant « Enter ».
* Le tir : Il y a 3 types de tir dans ce jeu, le tir du joueur, qui s’agit d’un œuf, et qui peut instantanément éliminer certains ennemis. Les ennemis peuvent tirer deux types de projectiles. Les ennemis du premier round tirent un seul projectile par tir de la même taille que l’œuf, alors que ceux du deuxième round envoient trois projectiles par tir. Chaque projectile enlève un point de vie au joueur.
* Gestion des vies : Le joueur commence chaque round avec 3 cœurs. Lorsque celui-ci est frappé par un projectile 1 cœur lui est enlevé. Cela arrive quelquefois que des cœurs apparaissent aléatoirement dans la carte, pour les récupérer, le joueur doit simplement entrer en contact avec le cœur. Cependant, le joueur ne peut uniquement ramasser le cœur s’il en possède moins de 4.

## Concept

Le diagramme de classe a été réalisé avec Draw.io, il se trouve dans le repo Github.

# Réalisation

Toute la description de la réalisation a pu être mise en commentaire dans le programme.

## Déroulement

Le déroulement s’est passé de manière agréable concernant les taches à effectuer, il n’y a pas eu d’imprévu et toues les tâches ont été finalisées.

## Mise en place de l’environnement de travail

## Description des tests effectués

Sans avoir accès au IceScrum, il est impossible de revoire les tests d’acceptance.

# Analyse de l’UX

## Conception centrée utilisateur

A close-up of a person's face

Description automatically generated

## Choix de la palette graphique

Deux maquettes ont été produites, par conséquent deux palettes graphiques ont été formées. Cependant seulement la première va être présentée car la deuxième est très semblable à la première. Voici la palette de la première maquette :

A screenshot of a computer screen

Description automatically generated

En tant que couleur primaire le choix a été d’utiliser des couleurs proches à du bleu foncé car cela représente bien le thème qui est un jeu dans l’espace, de plus, il est plus agréable de posséder des couleurs foncées en arrière-plan. Contrairement aux couleurs primaires, le choix pour les couleurs secondaires a été d’utiliser un ensemble de couleurs vives qui seront utilisées majoritairement utilisées pour les boutons d’action. Finalement, le blanc et le noir sont les deux couleurs de contraste car en cas contraire il y aurait beaucoup trop de couleurs ce qui ne serait moins agréable pour l’utilisateur.

## Eco-conception

Plusieurs aspects ont été pris en compte afin de disposer d’un site ergonomique :

* + - Les couleurs : Afin d’émettre le moins de luminosité possible, les couleurs principales s’agissent de couleurs plutôt sombres.
    - Les animations : Le site représenté par les maquettes n’aura aucune animation.
    - La résolution : La grande majorité des images sont pixélisées, ce qui rend leur résolution très basse.

## Accessibilité

Afin de mettre en place un site accessible à tout le monde, plusieurs aspects ont été pris en compte :

* + - Le contraste : La couleur des boutons d’action et du texte est très vive comparé à l’arrière-plan, ce qui facilite la lecture à ceux possédant des difficultés visuelles.
    - La navigation : Chaque bouton de navigation est assez grand pour pouvoir être accédé facilement, de plus, chacun d’entre eux représente bien son action.
    - Le gameplay : Le gameplay du jeu a été conçu pour être passablement simple afin que chaque individu puisse profiter du jeu sans beaucoup de difficulté.

# Conception

Avant chaque écran, un wireframe a été conçu. Le premier wireframe s’agit de la page d’accueil :

A screenshot of a game

Description automatically generated

Ensuite, c’est l’écran du sélectionneur de niveau :

A screenshot of a computer screen

Description automatically generated

Puis, on a un écran de gameplay :

A screenshot of a computer screen

Description automatically generated

Ensuite on a l’écran des Highscores :

A screenshot of a computer screen

Description automatically generated

Ensuite on a l’écran de personnalisation du personnage.

A screenshot of a computer

Description automatically generated

Finalement il y a l’écran de création de la carte/map :

A screenshot of a computer

Description automatically generated

## Ecran « Editeur de niveau »

Voici l’écran de l’éditeur de niveau :

A screenshot of a video game

Description automatically generated

Cet écran possède un icone pour accéder à la page d’accueil, un icone d’information qui offre un tutoriel à l’utilisateur si celui-ci a de la difficulté avec la manipulation de la page, un titre de la page et en bas, des onglets permettant la sélection des objets que l’on souhaite glisser sur la carte, avec un icone pour déplacer ces onglets.

# 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

Le journal de travail se trouve dans le repo github « P\_OO-Chickinvaders ».