Titre (nom) du projet



(Une image originale représentant le projet)

Nom de l’auteur – Classe

Lieu

Durée

Nom du chef de projet

(Nom et adresse du mandant)

Table des matières

[1 Spécifications 3](#_Toc308526316)

[1.1 Titre 3](#_Toc308526317)

[1.2 Description 3](#_Toc308526318)

[1.3 Matériel et logiciels à disposition 3](#_Toc308526319)

[1.4 Prérequis 3](#_Toc308526320)

[1.5 Cahier des charges 3](#_Toc308526321)

[1.5.1 Objectifs et portée du projet (objectifs SMART) 3](#_Toc308526322)

[1.5.2 Caractéristiques des utilisateurs et impacts 3](#_Toc308526323)

[1.5.3 Fonctionnalités requises (du point de vue de l’utilisateur) 3](#_Toc308526324)

[1.5.4 Contraintes 3](#_Toc308526325)

[1.5.5 Travail à réaliser par l'apprenti 4](#_Toc308526326)

[1.5.6 Si le temps le permet … 4](#_Toc308526327)

[1.5.7 Méthodes de validation des solutions 4](#_Toc308526328)

[1.6 Les points suivants seront évalués 4](#_Toc308526329)

[1.7 Validation et conditions de réussite 4](#_Toc308526330)

[2 Planification Initiale 4](#_Toc308526331)

[3 Analyse 4](#_Toc308526332)

[3.1 Opportunités 4](#_Toc308526333)

[3.2 Document d’analyse et conception 4](#_Toc308526334)

[3.3 Conception des tests 5](#_Toc308526335)

[3.4 Planification détaillée 5](#_Toc308526336)

[4 Réalisation 5](#_Toc308526337)

[4.1 Dossier de Réalisation 5](#_Toc308526338)

[4.2 Modifications 5](#_Toc308526339)

[5 Tests 5](#_Toc308526340)

[5.1 Dossier des tests 5](#_Toc308526341)

[6 Conclusion 5](#_Toc308526342)

[6.1 Bilan des fonctionnalités demandées 5](#_Toc308526343)

[6.2 Bilan de la planification 5](#_Toc308526344)

[6.3 Bilan personnel 5](#_Toc308526345)

[7 Divers 6](#_Toc308526346)

[7.1 Journal de travail 6](#_Toc308526347)

[7.2 Bibliographie 6](#_Toc308526348)

[7.3 Webographie 6](#_Toc308526349)

[8 Annexes 6](#_Toc308526350)

# Spécifications

## Description

Le but de ce projet est d’appliquer les apprentissages concernant l’orienté objet en c#.

Le cœur du projet est de créer un jeu vidéo ou un joueur sera dans un environnement ou lui et les ennemis pourrons se tirer dessus. Le but pour le joueur est de progresser dans le jeu pour atteindre le boss final, le tuer et remporter la partie.

## Matériel et logiciels à disposition

Le jeu tournera sur Windows Form et se basera sur la version fournie par le chef de projet, nommé "scrolling".

## Cahier des charges

### Fonctionnalités requises (du point de vue de l’utilisateur)

La grande majorité des fonctionnalités demander dans le cahier des charges sont présentent dans le programme. La seul qui n’est pas respecter, est le point a du point 7.3 du cahier des charges, le fait que « minimum 5 obstacles […] être détruit » à été remplacer, avec l’accord du chef de projet, par un sol qui se déplace de droite à gauche.

### Contraintes

La contrainte principale est que le projet est réalisé sur Windows Forms, cela fait que le moteur est peu optimisé pour du jeu, très rigide et ne contiens pas d’aide au développement direct.

Le jeu devant être codé en c# orienté objet, cela rajoute une difficulté d’apprentissage de cet orienté objet.

# Planification Initiale

Semaine 3 :

Création de l’environnement de jeu.

Semaine 4 :

Création du joueur et des déplacements.

Semaine 5 :

Déplacement

Semaine 6 :

Tirs et intégration des points de vie.

Semaine 7 - 8

Création des ennemis et leur comportement.

# Réalisation

## Dossier de Réalisation

Le jeu se base sur quelques grands principes : Le joueur et ses déplacements, les tirs, les ennemis et les points de vie.

### Gameplay

Le gameplay du jeu est relativement simple, des ennemis apparaissent et le joueur doit tirer dessus pour les éliminés et gagner des points (100 par ennemi). Une fois arriver à 2000 points le joueur gagne la partie. Les ennemis

tirs également et si la vie du joueur atteint 0 il a perdu la partie.

### Le joueur

Le joueur est représenté sur le AirSpace par un vaisseau que le joueur peut déplacer à l’aide des touche W, A, S et D. Le joueur est bloqué lors qu’il essaie de sortir de l’écran ou de descendre en dessous du sol.

### Tirs

Le jeu contient plusieurs types de tir, le joueur peut tirer des tirs normaux à l’infinie et des tirs de missiles, qui eux sont en nombre limité. Au début du jeu le joueur possède 3 missiles et lors qu’il en tire un, il est retiré de sa réserve. En tuant des ennemis, il a une chance sur quatre que l’ennemi laisse tomber un objet missile ramassable par le joueur pour incrémenter sa réserve. Les missiles, s’ils touchent l’ennemi, lui retire trois points de vie contre un pour les tirs normaux. Les tirs basiques ont un temps de recharge de 0.4 seconde.

Les ennemis de types "basic" et "front" ont des tirs qui parte en ligne droite de droite à gauche et disparaissent s’ils quittent l’écran ou touche le joueur.

Les ennemis de type "sniper" tirent un rayon laser qui prends toute la largeur de AirSpace instantanément après leur tir.

Les ennemis de type "tank" ne tirent pas.

### Ennemis

Il existe 4 types d’ennemis,

Les ennemis de type "basic" bouge aléatoirement de haut en bas et reste sur une position latérale qui leur est propre et défini aléatoirement. Ils ne tirs que des tirs basiques.

Les ennemis de type "front" avance lentement vers la gauche de l’écran, en tirant des tirs basiques, puis retourne rapidement à droite de l’écran. Il se déplace légèrement de haut en bas puis remonte après avoir touché le sol.

Les ennemis de type "sniper" choisissent aléatoirement une position puis s’y déplace et s’y arrête avant de tirer, puis recommence. Ils tirent des lasers qui font instantanément toute la largeur de l’écran.

Les ennemis de type "tank" ne tire pas et possède uniquement un point de vie. En revanche, il n’est pas sensible aux tirs basiques et ne peux être vaincu uniquement à l’aide de missile.

Les déplacements des différents ennemis ne leur permettent pas, ni de sortir du AirSpace, ni de passer sous le sol du niveau.

Au lancement du jeu, il y a 10 ennemis à l’écran, et il en apparait toutes les 4 à 7 secondes.

### Points de vie

Chaque entité, que se soit joueur ou ennemi, possède des points de vie. Ces points de vie diminuent, sauf exception, au contacte des projectiles. Alors que les tirs basiques font un point de dégât, les missiles en font trois. Le joueur à la possibilité de se soigner en récupérant des objets de soin qui peuvent être lâché par les ennemis à leur mort.

Lors que les points de vie arrivent à zéro, un ennemi meurt et est donc retirer du jeu. Si les points de vie du joueur arrivent à zéro, la condition de défaite est remplie, donc le joueur explose et un message de défaite apparait.

## Réalisation technique

# Conclusion

## Bilan des fonctionnalités demandées

Il s’agit de reprendre point par point les fonctionnalités décrites dans les spécifications de départ et de définir si elles sont atteintes ou pas, et pourquoi.

Si ce n’est pas le cas, estimer en « % » ou en « temps supplémentaire » le travail qu’il reste à accomplir pour terminer le tout.

## Bilan de la planification

Distinguer et expliquer les tâches qui ont généré des retards ou de l'avance dans la gestion du projet. Indiquer les différence entre les planifications initiales et détaillées avec le journal de travail.

## Bilan personnel

Si c’était à refaire:

Qu’est-ce qu’il faudrait garder ? Les plus et les moins ?

Qu’est-ce qu’il faudrait gérer, réaliser ou traiter différemment ?

Qu’est que ce projet m’a appris ?

Suite à donner, améliorations souhaitables, …

Remerciements, signature, etc.

# Divers

## Journal de travail

Voir fichier nommé "T-P\_POO-gillhorset-JournalTravail.pdf"

## Webographie

Les différents assets utilisés pour le jeu, viennent du jeu "Gradius" sortie sur la console de jeu NES en 1988. Le gameplay s’en inspire également.

<https://learn.microsoft.com/en-us/dotnet/csharp/>

<https://www.w3schools.com/cs>

# Annexes

Listing du code source (partiel ou, plus rarement complet)

Guide(s) d’utilisation et/ou guide de l’administrateur

Etat ou « dump » de la configuration des équipements (routeur, switch, robot, etc.).

Extraits de catalogue, documentation de fabricant, etc.