BowMasters



(Une image originale représentant le projet)

Maël Naudet – FID1

Lausanne

40 périodes

Patrick Chenaux

Table des matières

[1 Spécifications 3](#_Toc189218480)

[1.1 Titre 3](#_Toc189218481)

[1.2 Description 3](#_Toc189218482)

[1.3 Matériel et logiciels à disposition 3](#_Toc189218483)

[1.4 Prérequis 3](#_Toc189218484)

[1.5 Cahier des charges 3](#_Toc189218485)

[1.5.1 Objectifs et portée du projet (objectifs SMART) 3](#_Toc189218486)

[1.5.2 Caractéristiques des utilisateurs et impacts 3](#_Toc189218487)

[1.5.3 Fonctionnalités requises (du point de vue de l’utilisateur) 3](#_Toc189218488)

[1.5.4 Contraintes 4](#_Toc189218489)

[1.5.5 Travail à réaliser par l'apprenti 4](#_Toc189218490)

[1.5.6 Si le temps le permet … 4](#_Toc189218491)

[1.5.7 Méthodes de validation des solutions 4](#_Toc189218492)

[1.6 Les points suivants seront évalués 4](#_Toc189218493)

[1.7 Validation et conditions de réussite 4](#_Toc189218494)

[2 Planification Initiale 4](#_Toc189218495)

[3 Analyse 4](#_Toc189218496)

[3.1 Opportunités 4](#_Toc189218497)

[3.2 Document d’analyse et conception 4](#_Toc189218498)

[3.3 Conception des tests 5](#_Toc189218499)

[3.4 Planification détaillée 5](#_Toc189218500)

[4 Réalisation 5](#_Toc189218501)

[4.1 Dossier de Réalisation 5](#_Toc189218502)

[4.1.1 Fenêtre de jeu 5](#_Toc189218503)

[4.1.2 Création des joueurs 5](#_Toc189218504)

[4.1.3 Création des tours 5](#_Toc189218505)

[4.1.4 Création de la balle 5](#_Toc189218506)

[4.2 Modifications 6](#_Toc189218507)

[5 Tests 6](#_Toc189218508)

[5.1 Dossier des tests 6](#_Toc189218509)

[6 Conclusion 6](#_Toc189218510)

[6.1 Bilan des fonctionnalités demandées 6](#_Toc189218511)

[6.2 Bilan de la planification 6](#_Toc189218512)

[6.3 Bilan personnel 6](#_Toc189218513)

[7 Divers 6](#_Toc189218514)

[7.1 Journal de travail 6](#_Toc189218515)

[7.2 Bibliographie 6](#_Toc189218516)

[7.3 Webographie 6](#_Toc189218517)

[8 Annexes 6](#_Toc189218518)

# Spécifications

## Titre

BowMasters – Projet C# pour le module 320

## Description

Dans une application console, 2 joueurs, séparés par 2 grandes tours s’affrontent en se lançant des balles chacun à leur tour. Pour gagner la partie, il suffit de toucher 3 fois le joueur adverse.

## Matériel et logiciels à disposition

* Accès à Internet
* Visual Studio

## Prérequis

* Avoir suivi le module I319
* Suivre le module I320

## Cahier des charges

### Objectifs et portée du projet (objectifs SMART)

A compléter. Il s’agit d’ébaucher des réponses aux questions de l’acronyme CQQCOQP (Combien, Quoi, Qui, Comment, Où, Quand, Pourquoi)

### Caractéristiques des utilisateurs et impacts

A compléter… Il s’agit d’identifier le(s) profil(s) de(s) utilisateur-trice(s) type, et les conséquences que cela va avoir sur la conception (couleurs, ergonomie, utilisation, etc.)

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

A compléter par une espèce de mode d’emploi du produit. S’il s’agissait d’une montre, décrire qu’à part l’heure, il y aura la possibilité d’utiliser un chronomètre, un réveil, …

### Contraintes

Sécurité, backups, disponibilité, système utilisé, interfaces avec autres logiciels, etc.

### Travail à réaliser par l'apprenti

Décrire à quoi doit ressembler le travail produit, ce qu’il faudra rendre …

### Si le temps le permet …

Objectifs complémentairesau cas où le projet n’est pas assez ambitieux dans le temps imparti…

### Méthodes de validation des solutions

Comment les tests vont être entrepris, quels tests doivent être entrepris, etc.…

## Les points suivants seront évalués

* Le rapport
* Le code et les commentaires
* Les documentations de mise en œuvre et d’utilisation

## Validation et conditions de réussite

* Compréhension du travail
* Possibilité de transmettre le travail à une personne extérieure pour le terminer, le corriger ou le compléter
* Etat de fonctionnement du produit livré

# Planification Initiale

Ce paragraphe présente le planning d'origine (date de début, date de fin, vacances et congés, liste hiérarchique des tâches ou GANTT, jalons, durée totale)

Toutes les mises à jour subies par le planning sont à reporter (avec date de mise à jour) et peuvent déboucher sur plusieurs versions de plannings.

# Analyse

## Opportunités

Ce paragraphe énumère la liste des difficultés potentielles de tout ordre :

Liste des compétences à acquérir ou approfondir

Liste du matériel à exploiter

Recherche d’informations particulières

Gestion du travail en équipe & collaboration

Ainsi que les solutions possibles

Si les spécifications de départ ne laissent pas de doutes sur la manière de réaliser un projet, ce chapitre ne fera que renvoyer le lecteur aux spécifications.

## Document d’analyse et conception

Ce paragraphe décrit le fonctionnement de manière détaillée.

Autant que possible de manière graphique, imagée, tableaux, etc.

Tous les cas particuliers devraient y être spécifiés…

Il s’agit d’y présenter les fonctionnalités à développer :

Découpage en étapes, en modules, en fonctionnalités, etc.

Formulaires, interfaces graphiques, pages web, etc.

Schémas de navigation, schémas événementiels, structogramme, pseudocode, etc.

Si le projet inclut une base de données :

Dictionnaire des données

Modèle conceptuel des données, modèles logique des données.

## Conception des tests

Ce paragraphe permet de spécifier la stratégie de test qui sera menée au point 5.1

Qui, quand, avec quelles données, dans quel ordre, etc.

## Planification détaillée

A ce stade, après l’analyse complète du projet, un planning détaillé et complet (avec tâches, sous-tâches, dépendances, durée, …) peut être finalisé.

Le planning détaillé doit s’inscrire dans le planning initial. Il faut que l’on puisse situer cette planification détaillée par rapport à la planification initiale.

# Réalisation

## Dossier de Réalisation

### Fenêtre de jeu

Le projet a démarré avec la création de la fenêtre de jeu et sa modification. Pour cela, j’ai créé une classe *Config* avec les paramètres de grandeurs de la hauteur et de la largeur de la fenêtre et dans la méthode Main j’ai changé la taille de la fenêtre avec ces différents paramètres.

### Création des joueurs

Les joueurs de la classe *Player* ont plusieurs propriétés comme la gestion de leur vie, leur position dans la grille de jeu et leur modèle. Ils peuvent être affichés avec la méthode Display() et peuvent prendre des dégâts avec la méthode TakeDamage(byte damage).

### Création des tours

Pour faire la tour, j’ai d’abord créé les pièces avec la classe *TowerPiece* qui la composent en leur donnant un modèle et une position. J’ai permis de les afficher et de les enlever également. Ensuite, j’ai créé une classe *Tower* qui est en fait une liste de TowerPiece qui peut s’afficher.

### Création de la balle

Pour lancer la balle, il fallait d’abord faire une classe static *Balistic* qui calcule le mouvement d’un objet sur l’axe x et y en fonction du temps, de la vélocité, de la position initiale de l’objet. Ensuite, j’ai implémenté la balle en elle-même. Elle peut uniquement être modifiée avec les méthodes de Balistic et afficher si elle se trouve dans les limites de la console.

### Quatrième semaine

Lors de cette semaine, j’ai surtout amélioré la classe Game avec une meilleure gestion des collisions avec notamment une nouvelle classe HitBox qui est implémentée dans les joueurs et les tours. La puissance est maintenant indépendante du temps maximum donné et sera toujours entre 0 et 50. La logique du jeu est complète avec notamment une fin si un des joueurs n’a plus de vie. J’ai également ajouté des sons pour respectivement le tir, quand une balle touche une tour et quand une balle touche un joueur et quand on maintient la touche espace, on charge la barre au lieu que la barre se charge tout seul avec le fait d’appuyer sur espace pour stocker la valeur.

## Modifications

Historique des modifications demandées (ou nécessaires) aux spécifications détaillées.

Date, raison, description, etc.

# Tests

## Dossier des tests

On dresse le bilan des tests effectués (qui, quand, avec quelles données…) sous forme de procédure. Lorsque cela est possible, fournir un tableau des tests effectués avec les résultats obtenus et les actions à entreprendre en conséquence (et une estimation de leur durée).

Si des tests prévus dans la stratégie n'ont pas pu être effectués :

raison, décisions, etc.

Liste des bugs répertoriés avec la date de découverte et leur état:

Corrigé, date de correction, corrigé par, etc.

# 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

Date, activité (description qui permet de reproduire le cheminement du projet), durée, liens et références sur des documents externes. Lorsqu’une activité de recherches a été entreprise, il convient d’énumérer ce qui a été trouvé, avec les références.

## Bibliographie

Références des livres, revues et publications utilisés durant le projet.

## Webographie

Références des sites Internet consultés durant le projet.

# 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.