|  |
| --- |
| Bataille Navale |



West Gwenael

Gwenael.West@cpnv.ch

SI-C1b



06 mars 2019 – 2ème semestre,   
1ère année, 2019

Table des matières

[1 Introduction 3](#_Toc5571991)

[1.1 Cadre, description et motivation 3](#_Toc5571992)

[1.2 Organisation 3](#_Toc5571993)

[1.3 Objectifs 3](#_Toc5571994)

[1.4 Planification initiale 4](#_Toc5571995)

[2 Analyse 4](#_Toc5571996)

[2.1 Use cases et scénarios 5](#_Toc5571997)

[2.1.1 Use case 1: Jouer 5](#_Toc5571998)

[2.1.2 Use case 2: Lancer Programme 5](#_Toc5571999)

[2.1.3 Use case 3: Tirer 5](#_Toc5572000)

[2.1.4 Use case 4: Menu Aide 6](#_Toc5572001)

[2.1.5 Use case 5: Menu Quitter 6](#_Toc5572002)

[2.2 Stratégie de test 6](#_Toc5572003)

[3 Tests 6](#_Toc5572004)

[3.1 Tests effectués 6](#_Toc5572005)

[3.2 Erreurs restantes 7](#_Toc5572006)

[4 Conclusions 7](#_Toc5572007)

[5 Annexes 7](#_Toc5572008)

[5.1 Sources – Bibliographie 7](#_Toc5572009)

[5.2 Journal de bord du projet 8](#_Toc5572010)

[5.3 Journal de travail 8](#_Toc5572011)

# Introduction

## Cadre, description et motivation

Pour les Modules ICT 341 et MA-20 nous avons pour but de réaliser un code du jeu « Bataille Navale ». Nous devons fournir un programme opérationnel et toute la documentation accompagnant le projet. La motivation vient pour ma part du fait que nous pouvons réaliser le projet individuellement et en étant autonome. Nous avons 8 semaine pour le réaliser.

## Organisation

Eleve: West, Gwenael, Gwenael.West@cpnv.ch, 078.637.81.36

Responsable de projet et Expert 1: Benzonana, Pascal, [pascal.benzonana@cpnv.ch](mailto:pascal.benzonana@cpnv.ch), 076.230.23.13

Expert 2 : Carrel, Xavier, xavier,carrel@cpnv.ch, 024.557.60.80

## Objectifs

**Créer un programme « Bataille Navale » fonctionnel en 8 semaine pour ensuite le livrer au chef de projet.**

L’utilisateur peut afficher un menu aide qui lui explique les base d’une bataille navale.

L’utilisateur peut quitter le programme depuis le menu principal avec une touche « Quitter ».

Le programme affiche en combien de coup l’utilisateur a vaincu tous les navires.

L’utilisateur peut lancer une partie via le menu.

L’utilisateur peut sélectionner une case à attaquer grâce à un système de ligne et colonne.

L’utilisateur peut couler les bateaux et finir la partie.

## Planification initiale

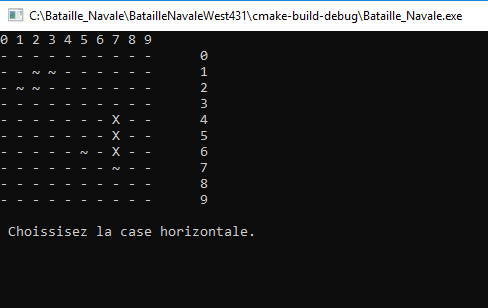
Date de début : 6 Mars 2019 Date de fin : 7 Avril 2019

Une image contenant capture d’écran

Description générée automatiquement

# Analyse

**Maquette :**



## Use cases et scénarios

### Use case 1: Jouer

|  |  |  |
| --- | --- | --- |
| *Action* | *Condition* | *Réaction* |
| Jouer | Couler tous les bateaux | Finir partie (gagner) |

### Use case 2: Lancer Programme

|  |  |  |
| --- | --- | --- |
| *Action* | *Condition* | *Réaction* |
| Cliquer sur le programme |  | Ouvrir le programme |
| Cliquer sur Jouer |  | Ouvrir la carte de jeu |

### Use case 3: Tirer

|  |  |  |
| --- | --- | --- |
| *Action* | *Condition* | *Réaction* |
| Choisir une case |  | Tire sur la case choisi |
|  | Si touché | Le programme affiche :Touché |
|  | Si raté | Le programme affiche : Raté |
|  | Si touche la dernière partie du bateau | Le programme affiche : Coulé |
|  | Si dernier bateau touché et coulé | Le programme affiche : Gagné |

### Use case 4: Menu Aide

|  |  |  |
| --- | --- | --- |
| *Action* | *Condition* | *Réaction* |
| Cliquer sur « Aide » |  | Affiche Aide |
| Appuyer sur une touche |  | Retourne sur le menu principal |

### Use case 5: Menu Quitter

|  |  |  |
| --- | --- | --- |
| *Action* | *Condition* | *Réaction* |
| Cliquer sur « Quitter » |  | Quitte le programme |

## Stratégie de test

* Test unitaire pour vérifier que chaque bloc de code fonctionne, et ensuite le Test d’intégration afin de constater que tout fonctionne.
* Pour les deux niveaux de test nous allons faire des test fonctionnel, test de performance et le test de robustesse.
* Un ami non familier a la programmation a testé le code afin d’y déceler d’éventuel bug.

# Tests

## Tests effectués

|  |  |
| --- | --- |
| L’utilisateur choisit le choix jouer | OK |
| L’utilisateur choisit le choix aide | OK |
| L’utilisateur choisit le choix quitter | OK |
| L’utilisateur rentre les coordonnées | OK |
| Bateau est touché | OK |
| Bateaux peuvent couler | OK |
| Gagne la partie quand tous les bateaux sont ko | OK |
| Si l’utilisateur met une lettre au lieu d’un chiffre, ne plante pas | OK |
| L’utilisateur voit son score a la fin | OK |

## Erreurs restantes

Lorsque on lance le jeu depuis Clion, l’affichage de la grille disparait par moment.

# Conclusions

Le programme est fonctionnel et on peut jouer sans gros problèmes. J’ai globalement apprécié le projet bien que j’aurais aimé faire un projet de groupe.

**Objectifs atteints :**

**-** On peut jouer une partie complète de Bataille Navale.

**Objectifs non atteints :**

* On ne peut pas choisir une carte aléatoire.
* Le score ne se sauvegarde pas dans un fichier externe.

**Points positifs :**

Le fait d’être assez libre et autonome.

**Points négatifs :**

On peut se sentir un peu perdu parfois.

**Difficultés particulières :**

Implémenter certaine fonction à partir de rien.

Certains bugs

Amélioration possible :

* Implémenter plusieurs grilles
* Implémenter une IA

# Annexes

## Sources – Bibliographie

* Sites internet : W3school, open Classroom.
* Aide externe : Jan

## Journal de bord du projet

|  |  |
| --- | --- |
| **Date** | **Evénement** |
| 22.03 | Théorie Journal de Bord |
| 27.03 | Création d’un tableau char |
| 27.03 | Théorie sur la méthode agile |
| 28.03 | Création d’un tableau int « tampons » |
| 28.03 | Corrigé problème du menu « Aide » qui quittait le programme |
| 01.04 | Théorie sur GitHub Desktop |
| 04.04 | Finalisation du Code |
| 07.04 | Finalisation du projet |

## Journal de travail

|  |
| --- |
| Théorie des uses-cases |
| Théorie sur les objectifs SMART |
| Théorie sur les fonctions |
| Théorie sur la méthode Agile |
| Théorie journal de bord et de travail |
| Théorie Méthode Agile 2 |
| Théorie de Github et Github Desktop |