|  |
| --- |
| Projet Bataille Navale |

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)

# Analyse préliminaire

## Introduction

Ce projet concerne un jeu de bataille navale que j'ai créé. Toute la programmation derrière ce jeu a été écrite en Clion. Les raisons du choix de ce projet étaient d'avoir un jeu simple à programmer puis d'apprendre à le documenter. J'ai grandement bénéficié de l'apprentissage de la réalisation d'un projet comme celui-ci, et j'ai l'impression que cela m'a aidé à comprendre comment documenter un projet

## Objectifs

Vous trouverez ci-dessous les objectifs que nous nous sommes fixés avant de démarrer le projet.

Mon premier objectif était de programmer les bases du jeu (grille, bateaux, menu de démarrage). Je voulais comprendre les bases et ensuite faire des choses plus avancées.

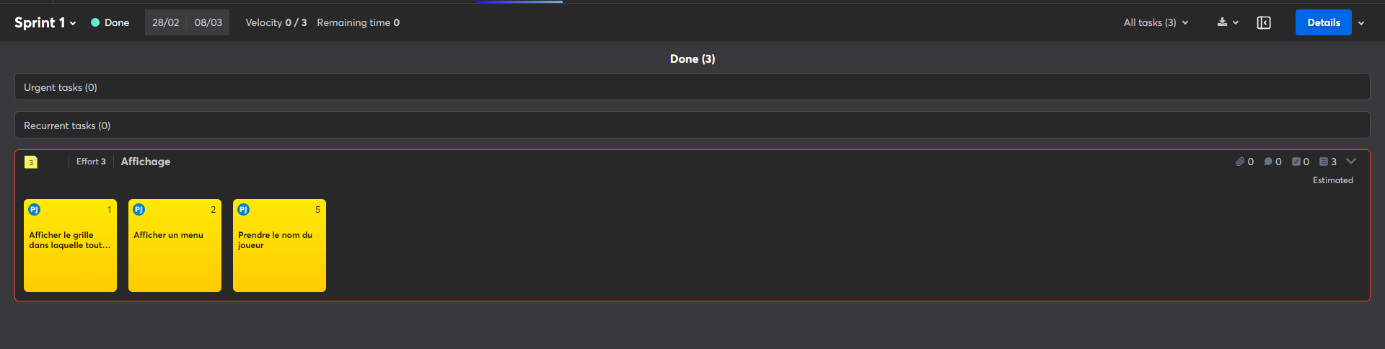
Mon deuxième objectif était de tirer parti de ce programme. Je voulais ajouter de meilleures choses comme un système de score, une page d'aide et une sélection au départ.

Mon objectif final était de documenter tout ce que je faisais. J'ai un journal de travail que je documenterais chaque fois que je travaillerais sur le projet. J'avais aussi un journal pour chaque fois que quelque chose m'aidait sur le projet, et les sources de cette aide.

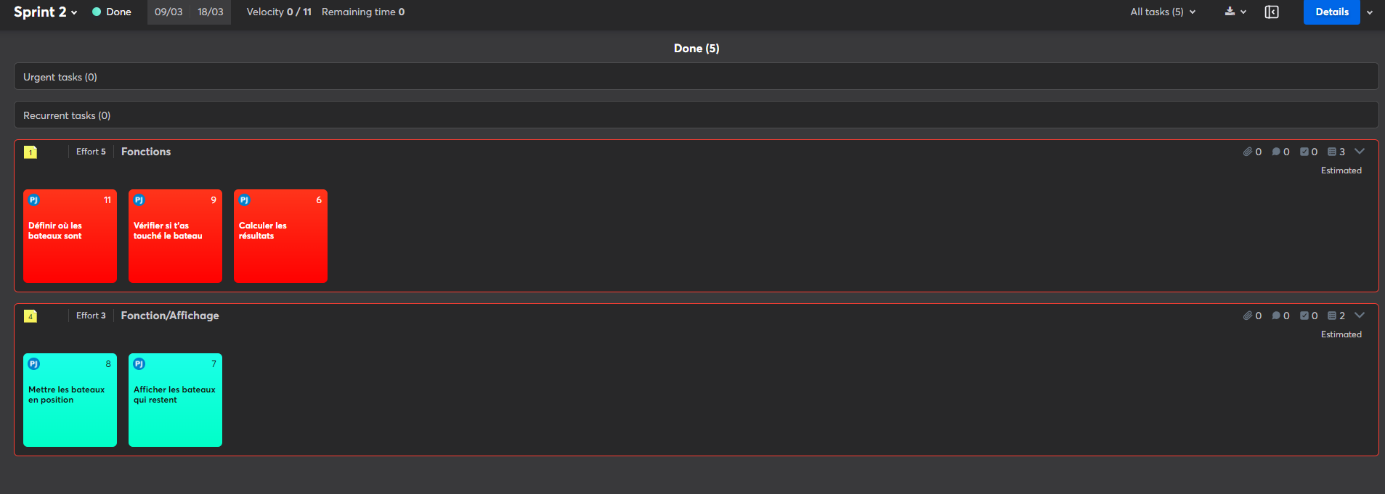
## Planification initiale

Vous trouverez ci-dessous les travails que nous nous sommes fixés avant de démarrer le projet. Ils sont organisés dans les sprints (périodes de travail)

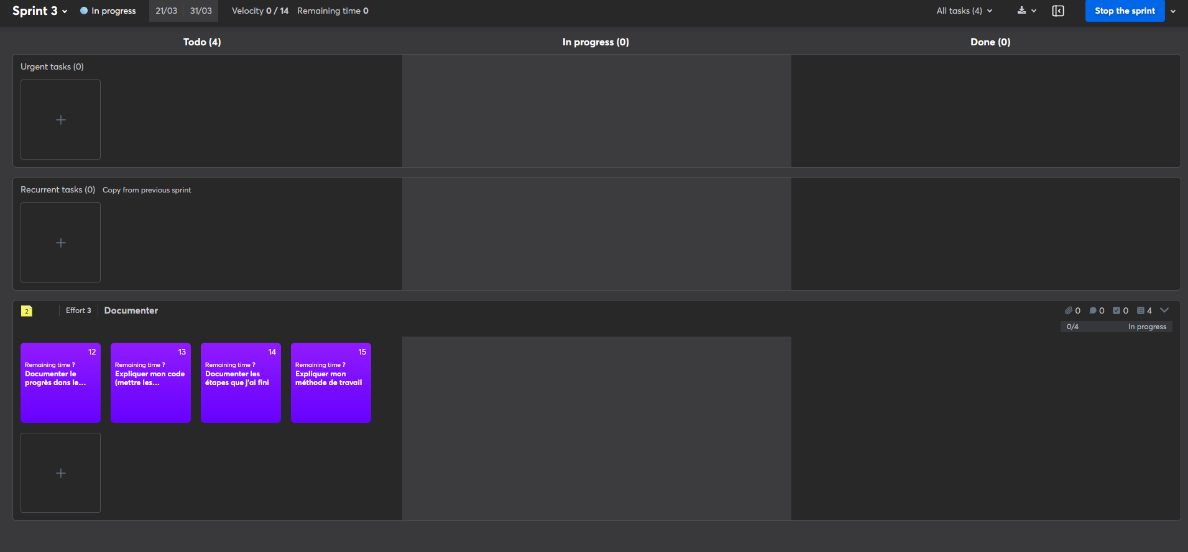
Voici le premier sprint :



Voici le deuxième



Voici le troisième



# Analyse / Conception

## Concept

Ces concepts ont été réalisés dans draw.io, une plateforme de création de

diagrammes en ligne gratuite. C'était l'analyse fonctionnelle de la programme

Diagram

Description automatically generated

A piece of paper with writing on it

Description automatically generated with low confidence

Diagram

Description automatically generated

## Stratégie de test

Les tests étaient simples : tant que le code ne casse pas, la programmation a réussi le test. Chaque fois que j'ajoutais quelque chose au code, je le testais toujours immédiatement après. Chaque partie du code doit fonctionner, donc tout ce qui ne va pas l'arrêtera.

## Risques techniques

Parfois, utiliser C pour programmer est compliqué, car il faut des fichiers externes pour construire le code. Il est assez efficace dans le bâtiment réel, mais il a besoin de beaucoup de fichiers externes pour le faire.

## Planification

La planification de ce projet a été découpée en sprints sur les 4 semaines que nous avons dû travailler sur ce projet. Cela expliquera les 4 semaines de planification à travers ces sprints.

Le premier sprint porterait sur les bases. Ce n'était qu'un sprint d'une semaine, car c'était juste un travail simple. Je voulais pouvoir d'abord construire une grille où seraient tous mes bateaux. Je voulais aussi faire un menu de sélection au démarrage pour le joueur, quand il charge le programme. Enfin, je voulais pouvoir avoir le nom d'utilisateur afin de pouvoir l'utiliser pour des fonctionnalités ultérieures, si je le voulais.



Le deuxième sprint s'est déroulé sur 2 semaines, où je ferais toutes les fonctions difficiles, comme créer des tableaux pour les lignes et les colonnes des bateaux, et déterminer pour mettre en évidence les photos qui ont été prises et manquées. Ce fut certainement la partie la plus difficile du projet, car nous avions un délai à la fin de ces 2 semaines pour rendre la version 0.1 de notre jeu (juste l'essentiel requis et un jeu fonctionnel).

A picture containing graphical user interface

Description automatically generated

Le sprint final consistait à documenter le projet. On nous a donné un fichier de Word de base de [I-CQ VD](https://www.tpivd.ch/index.php/documentation-tpi-cfc-ordo-2014/pour-candidat/39-documentations-cfc-annexe-3-canevas-dossier-de-projet-2k14/48-canevas-dossier-de-projet-2017-18). Cette étape était assez simple, car tout ce que nous avions à faire était de répondre aux questions au mieux de nos capacités. Cette partie du projet est la partie qui compte vraiment, donc la documentation doit être bonne.

A screenshot of a cell phone

Description automatically generated with medium confidence

## Dossier de conception

* Systèmes d'exploitation : Windows
* le choix des outils logiciels pour la réalisation et l'utilisation : Clion
* programmation et scripts: programmation en C, graphique du programme

*Diagram

Description automatically generated*

A piece of paper with writing on it

Description automatically generated with low confidence

Diagram

Description automatically generated

# Réalisation

## Dossier de réalisation

Décrire la réalisation "physique" de votre projet

Les répertoires où le logiciel est installé : C:\BATNAV1.0\NEWBATNAV\

La liste de tous les fichiers et une rapide description de leur contenu (des noms qui parlent !)

* .idea : Bases des données qui sauvegarde l’information sur le projet.
* cmake-build-debug : qui va créer le programme C et va vérifier si c’est propre ou pas.
* CMakeLists : document texte qui va initialiser le programme avec le code dedans et main.c
* main.c : le fichier de script avec tout les instructions pour la programme

Les versions des systèmes d'exploitation et des outils logiciels : Windows et Clion

* La description exacte du matériel : Windows 10
* le numéro de version de votre produit : Clion 2021.3.4
* programmation et scripts: cmake-build-debug, CMakeLists, .idea

## Description des tests effectués

Chaque fois que je teste le code, je le teste toujours une fois dans l'application Clion, puis je teste le fichier exécutable.

## Erreurs restantes

Les seules erreurs dans le code sont si vous tapez une lettre ou un chiffre supérieur à 9 lorsqu'il demande un numéro de ligne ou de colonne.

## Liste des documents fournis

Ce projet est un jeu de bataille navale programmé en C. Le jeu se joue dans une grille 9x9, où vous devez deviner où se trouvent les bateaux. Lorsque vous trouvez les 5, vous gagnez ! Essayez de les obtenir dans le plus petit nombre de tours.

Pour l'installation, vous devez d'abord avoir le lien GitHub pour télécharger le projet. Si vous n'avez pas Clion, vous ne pouvez pas modifier le document et devez jouer au jeu via le fichier exécutable.

Pour jouer au jeu, vous devez d'abord extraire tous les fichiers du dossier zip :

A screenshot of a computer

Description automatically generated with medium confidence

Une fois que vous avez tout extrait, allez dans C:\Users\Nom d’utilisateur\Downloads\BatailleNavale1.0-main\BatailleNavale1.0-main : Graphical user interface, application

Description automatically generated

Ensuite, extrayez les fichiers de ce fichier zip BATNAV1.0 et allez dans \BATNAV1.0\NEWBATNAV\, et la tu va trouver les fichiers de jeu :

Graphical user interface, application

Description automatically generated

Après, vous voudrez aller dans cmake-build-debug et trouver BATNAV 0\_1.exe. C'est le jeu, et pour jouer, vous pouvez simplement double-cliquer sur le fichier, et il commencera :

A screenshot of a computer screen

Description automatically generated with medium confidence

# Conclusions

J'ai pu atteindre tous mes objectifs, il n'y en avait aucun que je n'ai pas pu atteindre dans ce projet. Ces objectifs étaient faciles à suivre pour moi et je ne me suis pas surestimé en faisant quelque chose de trop dur.

Il y avait des points positifs et des points négatifs. Les bons points étaient de pouvoir créer un jeu complet qui avait de multiples fonctionnalités, un gameplay simple mais intéressant, et je l'ai fait tout seul et c'était quelque chose que je pouvais montrer aux gens, les laisser essayer le jeu et obtenir leurs opinions dessus. Je pense que ce projet a pu me donner un peu d'expérience sur la façon de gérer un projet.

Par contre, je ne suis pas doué pour me gérer dans le temps. Bien que ce projet m'ait aidé à mieux gérer mon temps et mon travail, je ne me sens toujours pas complètement sûr de mes orientations. J'avais parfois l'impression que j'allais mal répondre aux questions et je me sentais assez sous pression. Je suis aussi très nouveau et pas très doué pour la documentation, donc je sens que je dois travailler là-dessus dans de futurs projets.

La documentation est une grande difficulté pour moi, et je veux y travailler du mieux que je peux. Il est très facile de se perdre lorsque je fais de la documentation, et j'ai toujours l'impression que quelque chose ne va pas chaque fois que je l'écris.

Une autre chose que j'ai trouvée difficile était la gestion de mon temps. J'ai trouvé que je consacrais toujours trop de temps à la programmation, et pas beaucoup de temps à autre chose.

Je pense que le jeu aurait pu se développer beaucoup plus si j'avais eu plus de temps pour le programmer. J'avais quelques idées sympas à ajouter au projet, donc si j'avais eu plus de temps, il aurait pu y avoir plus de fonctionnalités dans mon jeu.

Je pense que j'aurais pu ajouter différents types de bateaux, comme la bataille navale traditionnelle. J'aurais également pu ajouter un meilleur système de notation, avec des noms d'utilisateur et des scores enregistrés. Je pense que s'il avait eu plus de temps, le projet aurait pu beaucoup évoluer.

# Annexes

## Résumé du rapport du TPI / version succincte de la documentation

## Sources – Bibliographie

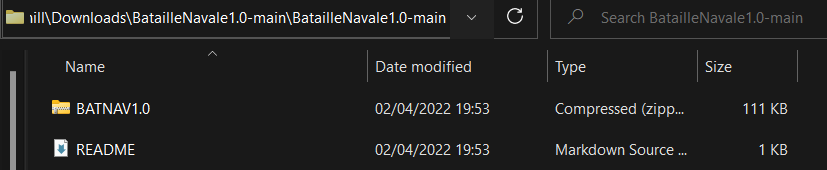
Sources :

* [Comment garder le score en C (video anglais)](https://www.youtube.com/watch?v=2b_UscEciCI)
* [fgetc en C](https://www.javatpoint.com/fputc-fgetc-in-c)

## Journal de travail



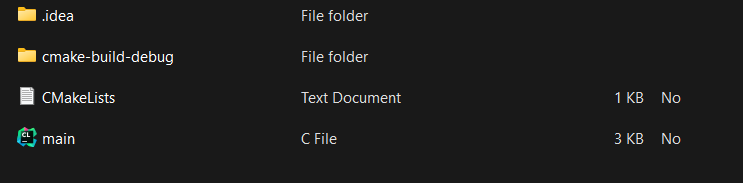
## Manuel d'Installation

Tout d'abord, vous devez télécharger le fichier zip à partir de ce lien GitHub [ici](https://github.com/ChristopherPhillippe/BatailleNavale1.0). Une fois que vous l'avez téléchargé, vous devez extraire les fichiers. Une fois que vous avez fait cela, parcourez les fichiers jusqu'à ce que vous trouviez un dossier avec ceci à l'intérieur :

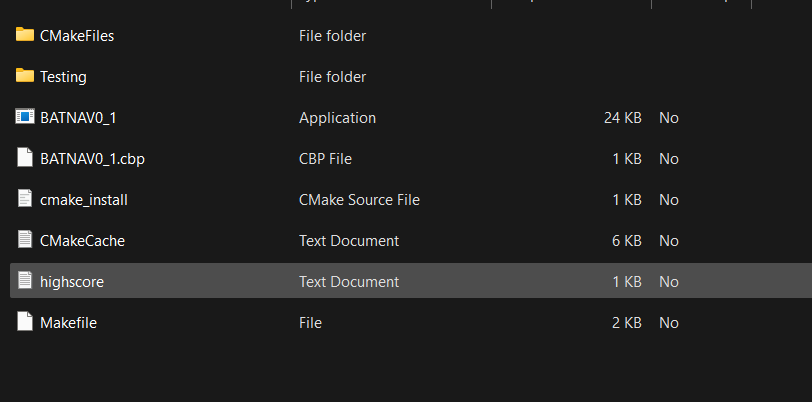
Vous devez à nouveau extraire les fichiers de ce dossier zip dans un autre dossier. Après avoir fait cela, vous avez terminé.

## Manuel d'Utilisation

Jouer est assez simple. Parcourez les fichiers que vous avez extraits et téléchargés et vous trouverez ceci :



Ouvrez le dossier "cmake-build-debug" et trouvez le fichier exécutable à l'intérieur appelé BATNAV 0\_1.



Double-cliquez sur le fichier et le jeu devrait démarrer.