|  |
| --- |
| Projet BATNAV |

Table des matières

[1 Analyse préliminaire 3](#_Toc99715512)

[1.1 Introduction 3](#_Toc99715513)

[1.2 Objectifs 3](#_Toc99715514)

[1.3 Planification initiale 4](#_Toc99715515)

[2 Analyse / Conception 4](#_Toc99715516)

[2.1 Concept 4](#_Toc99715517)

[2.2 Stratégie de test 8](#_Toc99715518)

[2.3 Planification 9](#_Toc99715519)

[2.4 Dossier de conception 11](#_Toc99715520)

[3 Réalisation 13](#_Toc99715521)

[3.1 Dossier de réalisation 13](#_Toc99715522)

[3.2 Description des tests effectués 14](#_Toc99715523)

[3.3 Erreurs restantes 16](#_Toc99715524)

[3.4 Liste des documents fournis 17](#_Toc99715525)

[4 Conclusions 18](#_Toc99715526)

[5 Annexes 19](#_Toc99715527)

[5.1 Résumé du rapport de la bataille navale / version succincte de la documentation 19](#_Toc99715528)

[5.2 Sources – Bibliographie 19](#_Toc99715529)

[5.3 Journal de travail 20](#_Toc99715530)

[5.4 Archives du projet 20](#_Toc99715531)

# Analyse préliminaire

## Introduction

*Ce projet qui, écrit en langage de programmation C, est une représentation graphique et ludique du jeu courant encore aujourd’hui, la bataille navale.*

La bataille navale est un jeu consistant et résultant à faire couler des bateaux, est un jeu stratégique où le joueur devra faire preuve de réflexion.

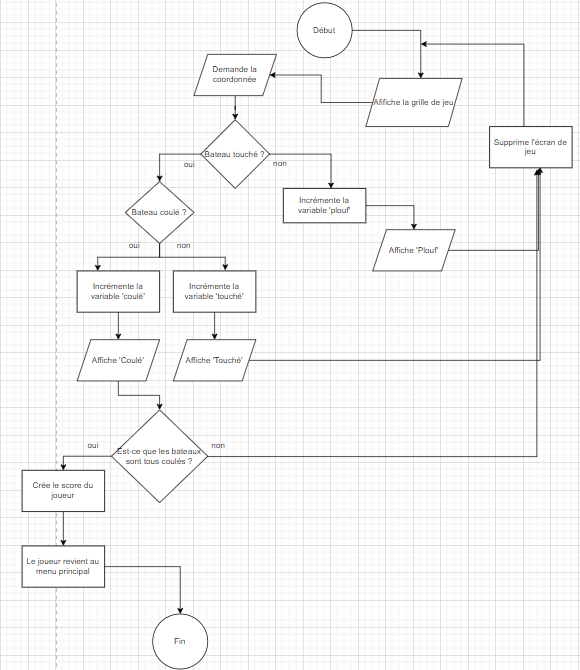
Le cadre du projet est administré par l’école, j’avais également des bases en programmation C auparavant, qui m’ont bien aidés à faire ceci.

## Objectifs

*Les objectifs de mon projet sont les suivants :*

* *Ecrire et administrer indépendamment une bataille navale écrite en langage de programmation C.*
* *La conception d’un menu de jeu où le joueur pourra choisir diverses options.*
* *La réalisation d’une map graphique dans le programme.*
* *Le placement de grilles dans le jeu.*
* *L’interaction entre le joueur et ses bateaux ennemis.*
* *La finition d’une session de jeu et la création de scores émis lors des sessions.*
* *L’affichage d’une aide de jeu qui aura comme principe :*
  + *Les règles du jeu.*
  + *Les diverses options du joueur au cours d’une session.*
* *L’affichage d’une liste de scores émis par chaque joueurs au cours des sessions de jeu.*

## Planification initiale

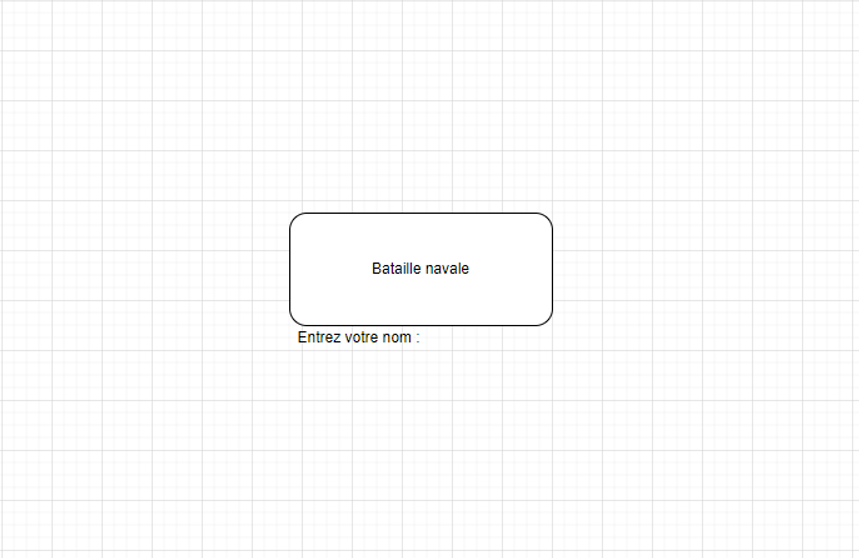


# Analyse / Conception

## Concept

* Conception du menu général ( en texte ) :

*Le menu général est le menu dans lequel on inscrit notre username ( nom du joueur ).*

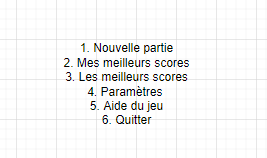
**

* *Conception du menu général ( final ):*

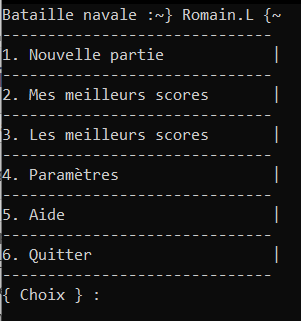
**

* *Conception du menu principal ( en texte ) :*

*Le menu principal est le menu dans lequel le joueur choisit une option parmi ceux représentées et lance la partie.*

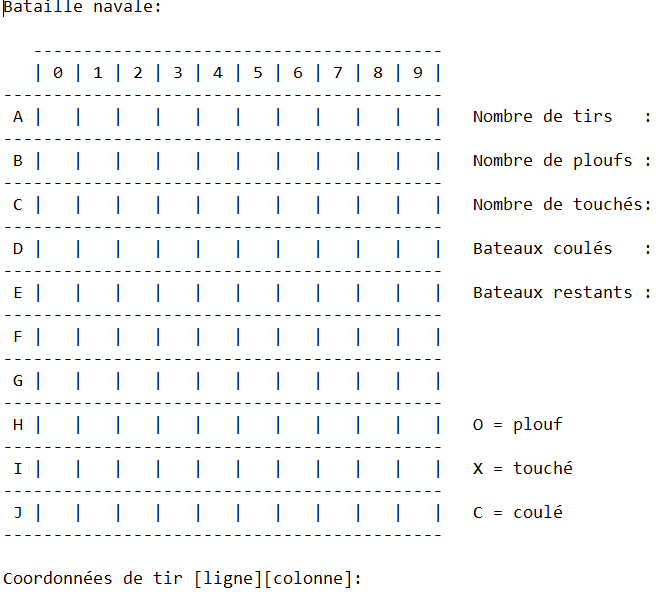
**

* *Conception du menu principal ( final ) :*

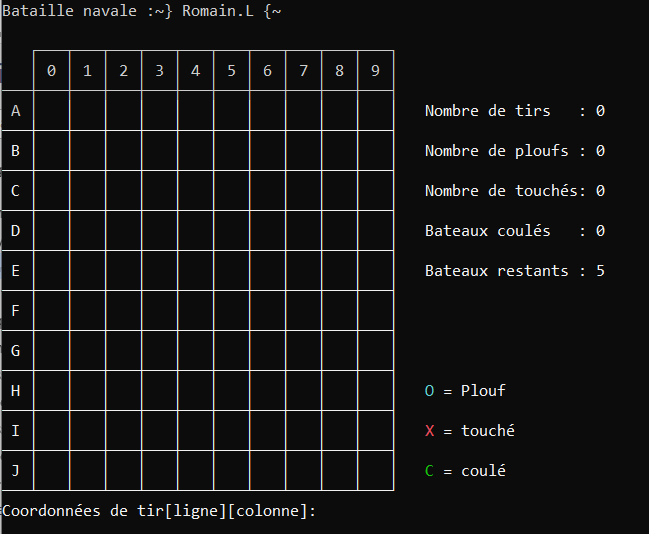
**

* *Conception de la map graphique ( en texte ) :*

*Cette conception de map, créé en prototype avant la création du code, m’a servi de modèle pour la création du plan.*

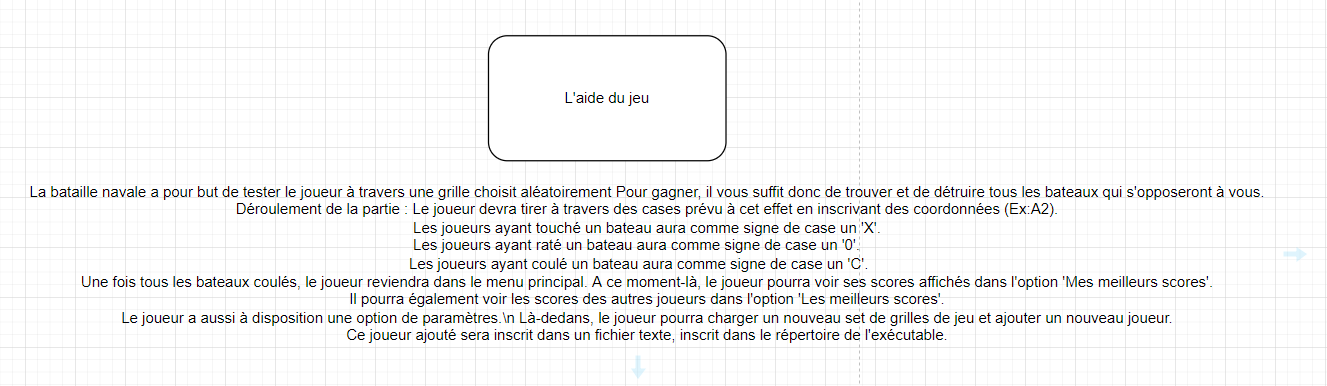
**

* *Conception de la map graphique ( final ) :*

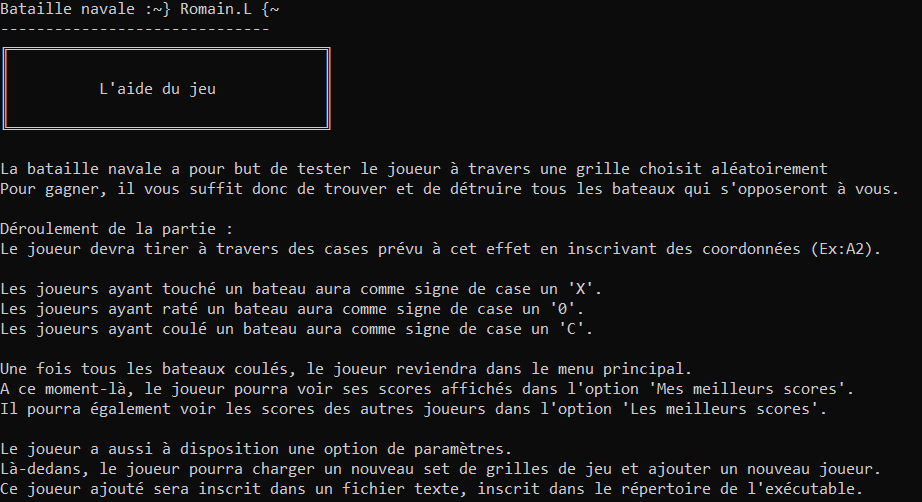
**

* *Conception de l’aide du jeu ( en texte ) :*

*Cette conception se déroule dans l’option de l’aide du jeu. Elle référencie toutes les règles et options du joueur.*

**

*Conception de l’aide du jeu ( final ) :*

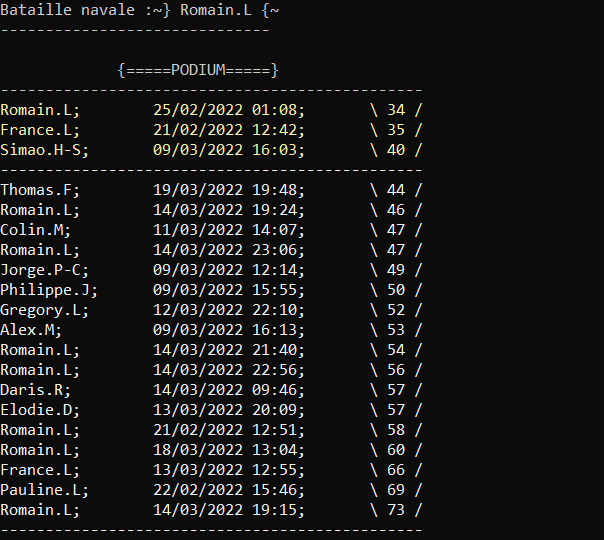
**

* *Conception des scores ( en texte ) :*

*Les scores se feront afficher par le code sous deux options : ‘Mes meilleurs scores’ et ‘Les meilleurs scores’ et servira à connaître le score des joueurs au cours des sessions de parties.*

**

* *Conception des scores ( final ) :*

**

## Stratégie de test

1. Test de robustesse : test qui a pour effet de décrire si le programme peut rester actif pendant de longues périodes lors du lancement du programme.
2. Test de capacité : test qui a pour effet de vérifier si oui ou non les logges du joueur marchent.
3. Tests aux limites : test qui a pour effet de vérifier que le programme lors du logge ne bugge pas, quelque qu’en soit le taux d’itérations écrites.
4. Test de capacité : test qui a pour effet de vérifier si oui ou non les choix du menu principal du joueur marchent.
5. Test aux limites : test qui a pour effet de vérifier que le programme lors des choix du menu principal ne buggent pas, quelque qu’en soit le taux d’itérations écrites.
6. Test de capacité : test qui a pour effet de vérifier si oui ou non les coordonnées du joueur marchent.
7. Tests aux limites : test qui a pour effet de vérifier que le programme lors des choix de coordonnées ne bugge pas, quelque qu’en soit le taux d’itérations écrites.
8. Test de capacité : test qui a pour effet de vérifier si oui ou non les Meilleurs scores du joueur connecté marchent.
9. Test de capacité : test qui a pour effet de vérifier si oui ou non les Meilleurs scores des joueurs marchent.
10. Test aux limites : test qui a pour effet de vérifier que le programme lors des choix des paramètres ne buggent pas, quelque qu’en soit le taux d’itérations écrites.
11. Test de capacité : test qui a pour effet de vérifier si oui ou non le choix du paramètre que le joueur a choisi marchent.
12. Test de capacité : test qui a pour effet de vérifier si oui ou non l’aide du jeu marche.
13. Test de capacité : test qui a pour effet de vérifier si oui ou non l’option de quitter le programme marche.

## Planification

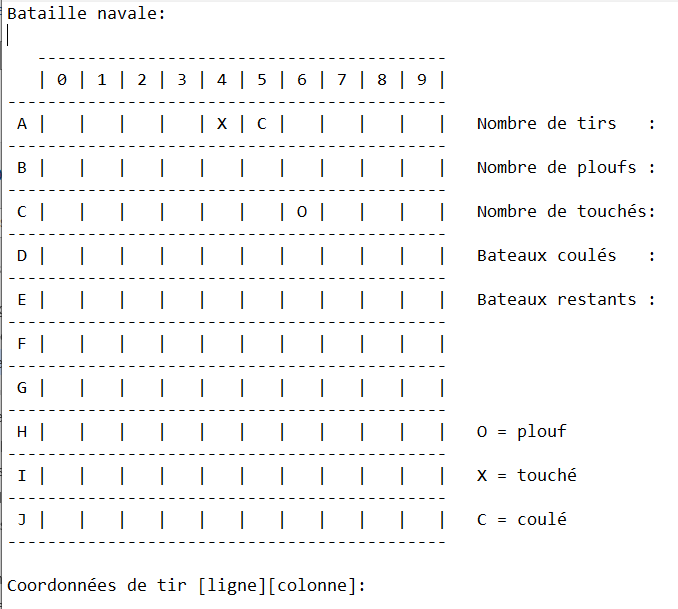
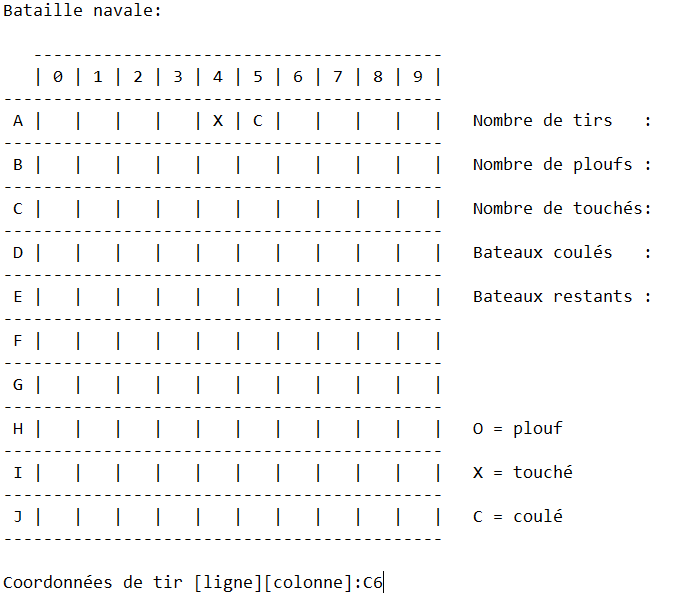
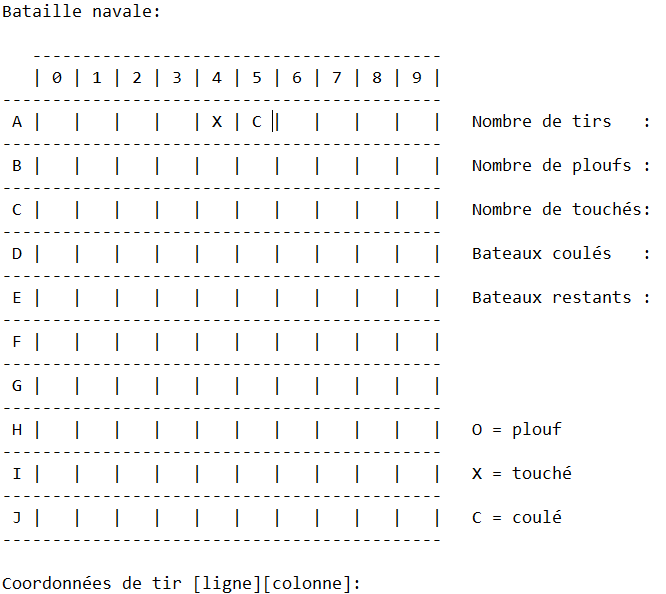
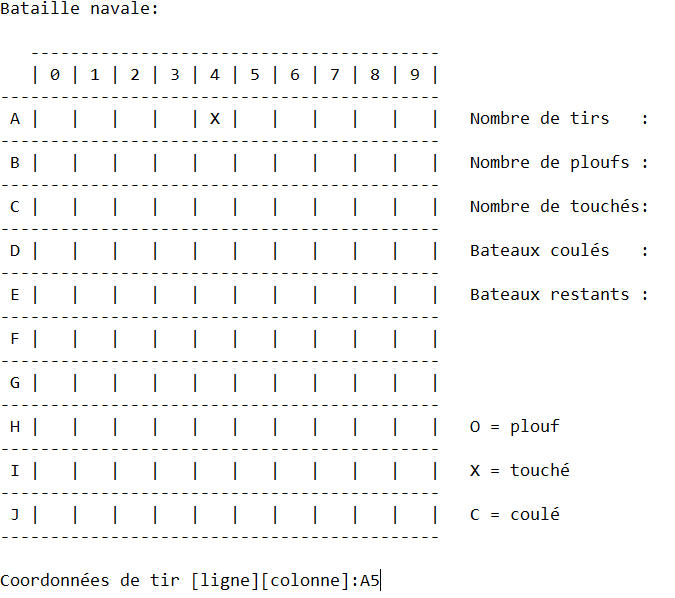
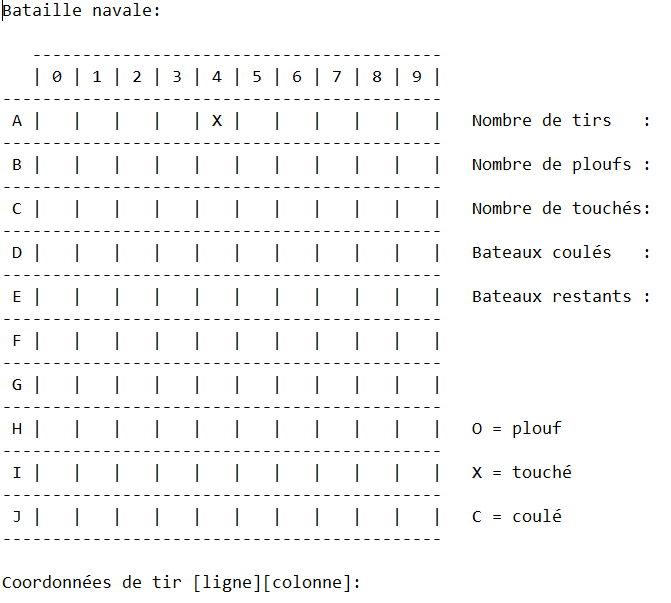
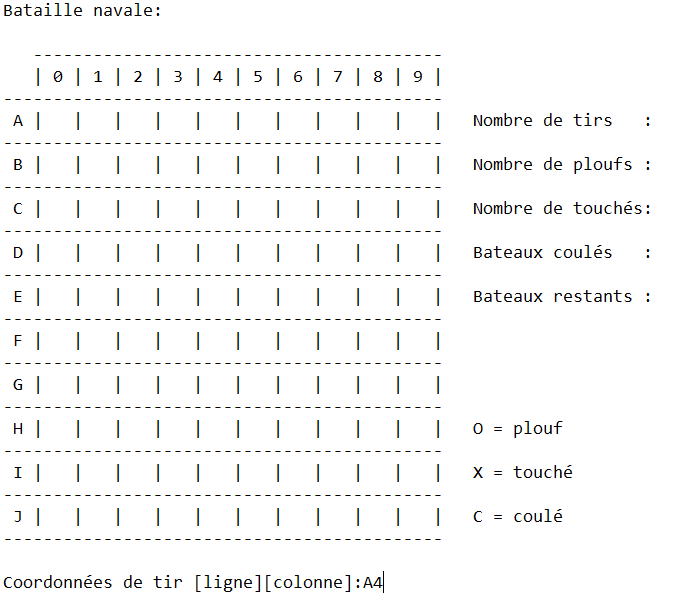
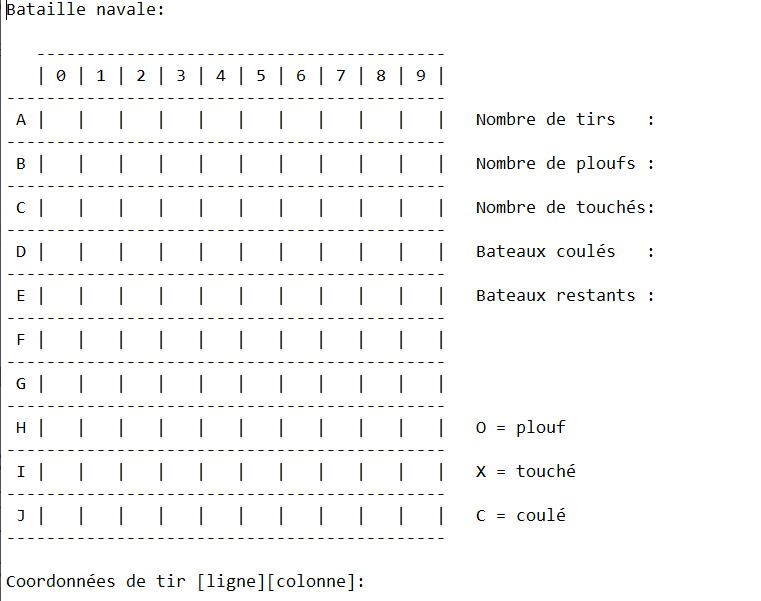
* *Version 0.1 – Rendu initial : [11.03.2022]*
  + *Sprint 1 – [16.02.2022-22.02.2022]*
    - *Faire le menu d’affichage*
      * *Afficher les divers options du menu d’affichage*
      * *Mettre au point un système de lancement du jeu*
      * *Mettre au point un système de règles dans le menu d’affichage*
      * *Mettre au point un système d’à propos dans le menu d’affichage*
      * *Mettre au point un système pour quitter le code*
      * *Faire des options de scores*
    - *Créer le système de tables*
      * *Créer et administrer la fonction de création et de génération de tables*
      * *Inscrire la fonction de générateur de tables dans le code*
  + *Sprint 2 – [23.03.2022-01.03.2022]*
    - *Créer l’algorithme des bateaux*
      * *Créer les variables qui représenteront les bateaux*
      * *Mettre en place les taille des différents bateaux dans le code*
      * *Insérer les bateaux dans le code du fait qu’ils soient statiques*
  + *Sprint 3 – [02.03.2022-08.03.2022]*
    - *Mettre en place le contrôle du joueur*
      * *Afficher « touché » ou « coulé » en fonction de la condition*
      * *Afficher un écran de fin lorsqu’il n’y a plus de bateau, que le joueur ait gagné*
      * *Mettre des conditions et repérer si un bateau est touché ou non*
      * *Mettre les différents variables appropriées*
      * *Définir l’algorithme du touché, coulé*
  + *Sprint 4 – [09.03.2022-11.03.2022]*
    - *Finaliser la version 0.1* 
      * *Commenter les fonctions*
* *Version 1.0 – Rendu initial : [01.04.2022]*
  + *Sprint 1*
    - *Placer aléatoirement les bateaux*
      * *Placer les bateaux aléatoirement du fait qu’il ne se touchent pas entre eux*
      * *Mettre en place l’algorithme du placement dans le programme et créer une fonction liées*
    - *Mettre en marche le système de points* 
      * *Pointer le score sur les variables*
      * *Créer les variables en rapport avec les points obtenus*
      * *Afficher le score obtenus lors des parties précedentes*
    - *Commenter et administrer le code*
      * *Faire en sorte qu’on ne puisse pas faire buguer le programme avec les ‘scanf’*
      * *Faire une fonction pour la vérification des coordonnées*
      * *Mettre des couleurs dans la bataille navale*
      * *Commenter le code afin de le finaliser*

|  |
| --- |
| *Voici en parallèle*   * *mon IceScrum : https://icescrum.cpnv.ch/p/BATNAVROM/#/project* |

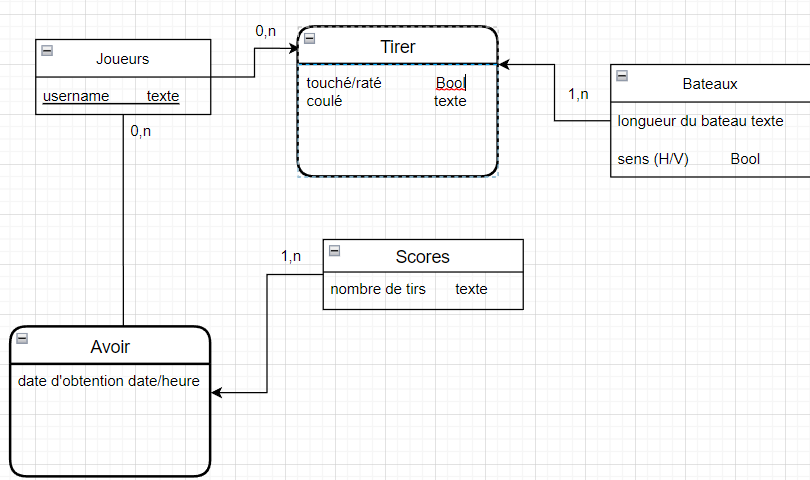
## Dossier de conception

*Fournir tous les document de conception:*

* *Mon matériel HardWare utilisé pour ce projet est mon poste de travail au CPNV ainsi que mon ordinateur virtuel*
* *Mon système d’exploitation utilisé pour ce projet est uniquement sous windows*
* *Pour ce projet, j’ai utilisé GitHub pour la conservation et pour archiver mes fichiers, IceScrum pour la conception de story, de tâches et de sprints, CLion et Visual Studio Code pour écrire le programme et cmd pour l’exécution*
* *Maquettes :*

**

* *MCD*

**

*Joueurs : A un username bien à lui.*

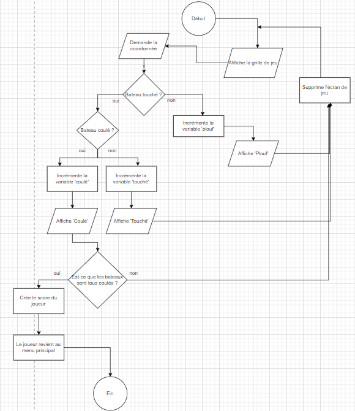
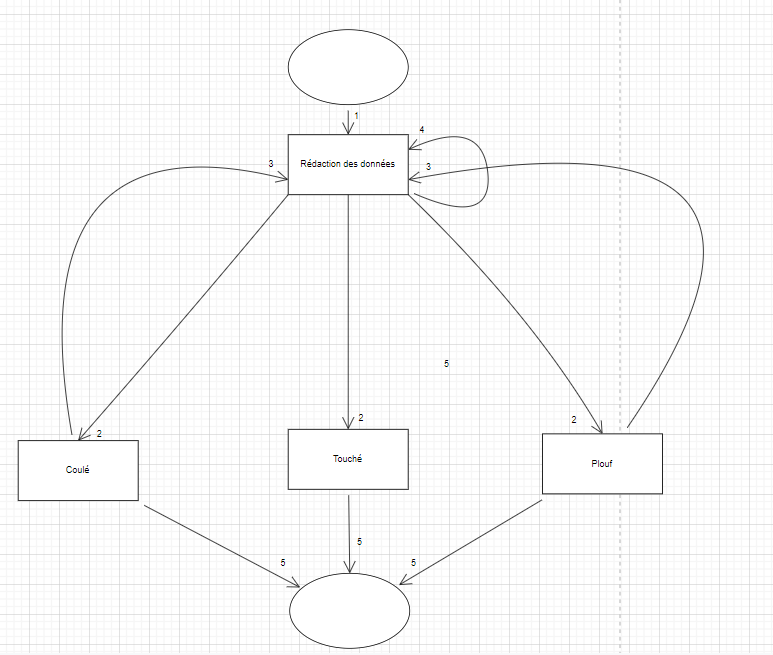
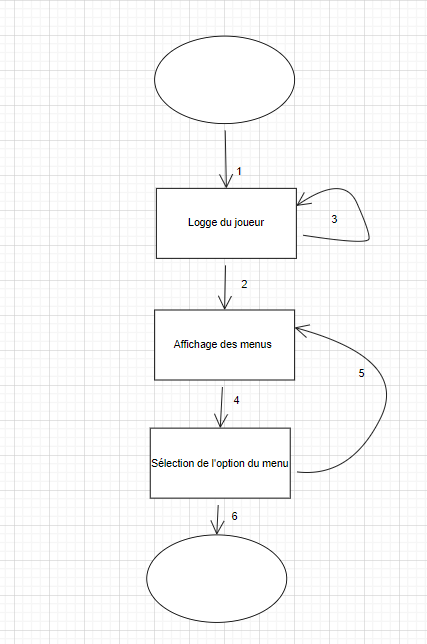
*Tirer : Action de tirer aux bateaux adversaires, Touché ou raté, ou coulé.*

*Bateaux : Incarne les bateaux adverses, longueur du bateau et sens ( s’il est posé à la vertical ou à l’horizontal.*

*Avoir : Possession, date d’obtention des scores pour le joueur.*

*Scores : Tableau des scores, est déterminé par le nombre de tir.*

* *Organigramme et états :*

**

# Réalisation

## Dossier de réalisation

* *Le logiciel, mon code et mon exécutable se trouve sur mon serveur, sous un répertoire GitHub*
* *Historique des fichiers*
* *Les fichiers cmake-build : fichiers administratifs*
* *Les fichiers Model : fichiers de diagramme et de maquettes*
* *Les fichiers de versions : fichiers de la version 0.1*
* *Fichier CMakeLists : fichier administratifs*
* *les versions des systèmes d'exploitation et des outils logiciels*
  + *Microsoft Windows 10 Famille 64 bits 10.0.19043, version G5113QM.316*
  + *CLion : version 3.21*
  + *Visual Studio 2022 : version 17.1.2*
* *la description exacte du matériel*
  + *Windows*
    - [**Windows 10**](https://fr.wikipedia.org/wiki/Windows_10) est un [système d'exploitation](https://fr.wikipedia.org/wiki/Syst%C3%A8me_d%27exploitation) développé par Microsoft, qui le décrit comme un « [système d'exploitation en tant que service](https://fr.wikipedia.org/wiki/Logiciel_en_tant_que_service) » qui reçoit des mises à jour permanentes de ses caractéristiques et de ses fonctionnalités, ainsi que la possibilité pour les versions destinées aux entreprises de voir ce rythme ralenti pour les mises à jour non-critiques ou, lors du support à long terme, recevoir uniquement les mises à jour critiques, telles que des correctifs de sécurité. [Terry Myerson](https://fr.wikipedia.org/wiki/Terry_Myerson), vice-président de Microsoft chargé des systèmes d'exploitation, explique que ce système a pour but de réduire la fragmentation de Windows sur les différentes plateformes[1](https://fr.wikipedia.org/wiki/Historique_des_versions_de_Windows_10#cite_note-zdnet-onebillion-1).
      * Source : <https://fr.wikipedia.org/wiki/Historique_des_versions_de_Windows_10>
  + CLion
    - *CLion est conçu pour le développement multiplateforme natif en C et C++, ainsi qu'en Rust et Swift. Il fournit également une prise en charge essentielle du langage CMake, Python, des technologies Web populaires (comme JavaScript, XML, HTML, Markdown et autres), et cela inclut également la prise en charge de toutes les autres langues disponibles via les plugins groupés et de référentiel.*
      * *Source :* [*https://www.jetbrains.com/clion/features/*](https://www.jetbrains.com/clion/features/)
  + *Visual Studio Code*
    - *Visual Studio Code est un éditeur de code extensible développé par Microsoft pour Windows, Linux et macOS2. Les fonctionnalités incluent la prise en charge du débogage, la mise en évidence de la syntaxe, la complétion intelligente du code, les snippets, la refactorisation du code et Git intégré. Les utilisateurs peuvent modifier le thème, les raccourcis clavier, les préférences et installer des extensions qui ajoutent des fonctionnalités supplémentaires.*
      * *Source :* [*https://fr.wikipedia.org/wiki/Visual\_Studio\_Code*](https://fr.wikipedia.org/wiki/Visual_Studio_Code)

* *Librairies externes* 
  + *<stdio.h>*
  + *<stdlib.h>*
  + *<time.h>*
  + *<windows.h>*
  + *<string.h>*
  + *<conio.h>*
  + *<sys/stat.h>*

## Description des tests effectués

*Pour chaque partie testée de votre projet, il faut décrire:*

1. Test de robustesse : test qui a pour effet de décrire si le programme peut rester actif pendant de longues périodes lors du lancement du programme.

Condition : Le programme marche même au bout d’un certain temps dans la partie du menu général.

Preuve de test : Le programme marche même s’il reste actif pendant longtemps.

1. Test de capacité : test qui a pour effet de vérifier si oui ou non les logges du joueur marchent.

Condition : Le joueur entre un nom d’utilisateur, s’il est inscrit dans le fichier 🡪 Cela marche, si le programme quitte 🡪 le nom du joueur n’est pas représenté dans la liste de joueur.

Preuve de test : Le programme quitte expressément l’exécutable du fait que le joueur n’ait pas encore de nom de joueur.

1. Tests aux limites : test qui a pour effet de vérifier que le programme lors du logge ne bugge pas, quelque qu’en soit le taux d’itérations écrites.

Condition : Le programme ne bugge pas même aux coups du spam du joueur.

Preuve de test : Le joueur peut spammer un petit peu sans risque de bugs, mais peut quand même saturer la mémoire, bug C.

1. Test de capacité : test qui a pour effet de vérifier si oui ou non les choix du menu principal du joueur marchent.

Condition : Le joueur entre une des options écrites, s’il correspond à une des options 🡪 Il marche et rentre dans l’option, si non 🡪 il revient au menu des options.

Preuve de test : le programme marche ou quitte le programme lorsque le joueur y mets son choix.

1. Test aux limites : test qui a pour effet de vérifier que le programme lors des choix du menu principal ne buggent pas, quelque qu’en soit le taux d’itérations écrites.

Condition : le programme marche et ne bugge pas, même au détriment du spam des joueurs.

Preuve de test : Le joueur peut spammer un petit peu sans risque de bugs, mais peut quand même saturer la mémoire, bug C.

1. Test de capacité : test qui a pour effet de vérifier si oui ou non les coordonnées du joueur marchent.

Condition : Le programme prend en compte les coordonnées du joueur et les redirige ou non sur la case prescrite.

Preuve de test : Le programme ne redirige bien les coordonnées dans les cases prescrites.

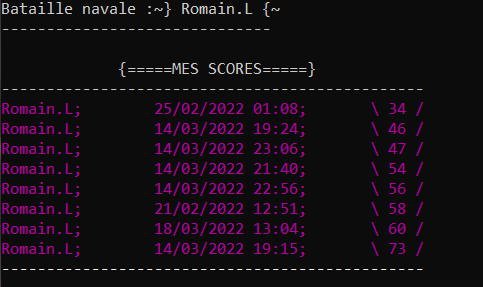
1. Tests aux limites : test qui a pour effet de vérifier que le programme lors des choix de coordonnées ne bugge pas, quelque qu’en soit le taux d’itérations écrites.

Condition : Le programme ne bugge pas lors des éléments émis.

Preuve de test : Le programme ne bugge pas lorsque le joueur écrit autre que des coordonnées.

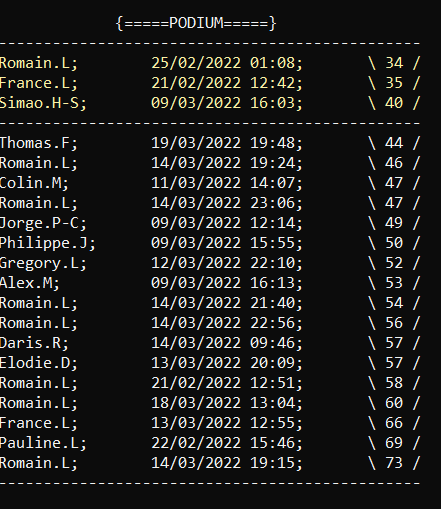
1. Test de capacité : test qui a pour effet de vérifier si oui ou non les Meilleurs scores du joueur connecté marchent.

Condition : Le programme affiche tous les scores du joueur connecté.

Preuve de test : 

1. Test de capacité : test qui a pour effet de vérifier si oui ou non les Meilleurs scores des joueurs marchent.

Condition : Le programme affiche tous les scores des joueurs.

Preuve de test : 

1. Test aux limites : test qui a pour effet de vérifier que le programme lors des choix des paramètres ne buggent pas, quelque qu’en soit le taux d’itérations écrites.

Condition : le programme ne bugge pas, quoi que le joueur écrit.

Preuve de test : Le joueur peut spammer un petit peu sans risque de bugs, mais peut quand même saturer la mémoire, bug C.

1. Test de capacité : test qui a pour effet de vérifier si oui ou non le choix du paramètre que le joueur a choisi marchent.

Condition : Le programme redirige le joueur sur l’option qu’il a émise.

Preuve de test : Le programme redirige bien sur l’option que le joueur à inscrit.

1. Test de capacité : test qui a pour effet de vérifier si oui ou non l’aide du jeu marche.

Condition : le programme affiche l’aide du jeu aux joueurs.

Preuve de test : le programme affiche bien l’aide du jeu aux joueurs concernés.

1. Test de capacité : test qui a pour effet de vérifier si oui ou non l’option de quitter le programme marche.

Condition : le programme quitte l’exécutable lorsque la permission lui est assignée.

Preuve de test : Le programme, lorsqu’il reçoit la permission de quitter le programme, le quitte.

## Erreurs restantes

* *Il reste encore un gros point à voir, la mémoire en C permet beaucoup de choses mais permet aussi d’être affecté grossièrement par ses variables, on peut spammer un petit peu les zones d’écriture, mais on ne peut pas contourner le fait que le langage C ait une limite de mémoire.*
* *Les conséquences seraient donc que le joueur fasse et réussisse à faire bugger le programme en cassant la limite de mémoire définie par le langage.*
* *Il serait possible d’agrandir la mémoire de certaines variables en utilisant d’autres types par exemple, cependant il n’est pas possible d’y remédier.*

## Liste des documents fournis

* *le rapport de projet*
  + *Le client nous a donné comme appui un cahier des charges*

# Conclusions

* *Mes objectifs atteints / non-atteints*
  + *J’ai réussi a donné à mon code un sens très concret de la bataille navale et j’en suis plutôt fier, j’ai aussi appris plusieurs choses en C, en d’autres termes, j’ai utilisé des structures. Choses qui ne m’a pas vraiment été utile sur d’autres programmes. J’ai réussi à utiliser correctement des pointeurs et ouvert des fichiers, les lire et les écrire. J’ai pu faire un menu principal sobre, mais suffisant. Fait une map graphique que j’aie ensuite mis dans mon répertoire en fichier texte. Fait ensuite la création de plusieurs grilles à l’aide d’une fonction et d’un répertoire qui répertorie 1’000 maps. L’interaction, donc la session de partie a été aussi bien réalisé, avec des signes pour symboliser l’état du tir, que j’ai, suite après cela, modifié pour les rendre en couleurs dans mon programme. La finition de la session se déroule aussi bien, avec le fichier de log des scores qui s’actualise pour donner de vrais stats aux joueurs. L’affichage d’une aide du jeu qui conserve toutes les règles de base d’une bataille navale ainsi que des informations liées aux options que le joueur peut accéder et l’affichage de la liste de scores.*
* *Points positifs / négatifs* 
  + *J’ai réussi à utiliser des structures de stats dans mon programme, le projet en général m’a aidé à comprendre mieux le fait d’administrer un projet, m’a appris en d’autres termes GitHub et IceScrum.*
* *Difficultés particulières*
  + *J’ai eu certaines difficultés avec l’IDE ou environnement de développement CLion, car je le trouve pas optimisé au niveau du code et de la prise en charge du déboggeur. J’ai aussi, au niveau du code, eu certaines difficultés au niveau des pointeurs, car certains appels dans les fonctions ou incrémentations en sont restés difficiles à comprendre, néanmoins j’ai fini par réussir.*
* *Suites possibles pour le projet (évolutions & améliorations)*
  + *Mon projet est finalisé et bien optimisé, mais il pourrait être moins sobre et plus évolutif. aspect graphique, il pourrait avoir plus de ‘dessins informatiques’, que ce soit dans les sessions de jeu ou dans le menu principal. Rajouter des sons, comme je l’avais déjà pensé, des sons qui serait en lien dans les sessions de jeu, pour le plouf, le touché et le coulé, ou encore quand le joueur atteint le menu principal.*

# Annexes

## Résumé du rapport de la bataille navale / version succincte de la documentation

Dans ce rapport, j’ai relevé plusieurs de mes péripéties et idées de code que j’aie eu durant tout le long du projet, j’ai parlé de mes objectifs, de ma planification, du concept, de la stratégie de test, de la réalisation, des descriptions de tests effectués durant le projet, des erreurs potentielles restantes, des documents fournis ainsi qu’une brève conclusion.

## Sources – Bibliographie

|  |
| --- |
| [https://icescrum.cpnv.ch/#/](https://icescrum.cpnv.ch/" \l "/) |
| https://icescrum.cpnv.ch/#/ |
| <https://openclassrooms.com/forum/sujet/placer-les-bateaux-aleatoirement-dans-une-bataille-navale-86985> |
| <https://koor.fr/C/cstdlib/rand.wp> |
| <https://www.google.com/imgres?imgurl=https%3A%2F%2Fwww.maison-gourmande.com%2Fwp-content%2Fuploads%2F2020%2F03%2Fbataille-navale.jpg&imgrefurl=https%3A%2F%2Fwww.maison-gourmande.com%2Fbataille-navale.html&tbnid=jsGjT-cSahChjM&vet=12ahUKEwjy4v2Ikoz2AhUU_hoKHWLMAoMQMygBegUIARDzAQ..i&docid=yZGz0Nue7R5NwM&w=800&h=609&q=bataille%20navale&client=opera-gx&ved=2ahUKEwjy4v2Ikoz2AhUU_hoKHWLMAoMQMygBegUIARDzAQ> |
| https://stackoverflow.com/questions/2347770/how-do-you-clear-the-console-screen-in-c |
| <https://openclassrooms.com/fr/courses/19980-apprenez-a-programmer-en-c/16421-manipulez-des-fichiers-a-laide-de-fonctions> |
| <https://stackoverflow.com/questions/1442116/how-to-get-the-date-and-time-values-in-a-c-program> |
| <https://koor.fr/C/cstdio/fscanf.wp> |
| <https://stackoverflow.com/questions/18801483/press-any-key-to-continue-function-in-c> |
| <https://www.tutorialspoint.com/c_standard_library/c_function_strlen.htm> |
| <http://www.ascii-fr.com/-Bateaux-.html> |
| [https://app.diagrams.net](https://app.diagrams.net/) |
| [https://app.diagrams.net](https://app.diagrams.net/) |
| https://www.tpivd.ch |
| https://www.tpivd.ch |

## Journal de travail

## Archives du projet

[*https://github.com/Romain-len/BATNAV-RL*](https://github.com/Romain-len/BATNAV-RL)