Projet Bataille Navale

Antoine Roulin

# Table des matières

[Table des matières 2](#_Toc37329867)

[1 Analyse préliminaire 3](#_Toc37329868)

[1.1 Introduction 3](#_Toc37329869)

[*1.2* Objectifs 3](#_Toc37329870)

[1.3 Planification initiale 3](#_Toc37329871)

[2 Analyse / Conception 5](#_Toc37329872)

[2.1 Concept 5](#_Toc37329873)

[2.1.1 Menu : 5](#_Toc37329874)

[2.1.2 Login : 5](#_Toc37329875)

[2.1.3 Aide : 6](#_Toc37329876)

[2.1.4 Score : 6](#_Toc37329877)

[2.1.5 Jeu : 7](#_Toc37329878)

[2.1.6 Logs : 8](#_Toc37329879)

[2.1.7 Fichiers des cartes : 8](#_Toc37329880)

[*2.2* Stratégie de test 8](#_Toc37329881)

[*2.3* Planification 9](#_Toc37329882)

[2.3.1 Version 1.0 9](#_Toc37329883)

[2.4 Dossier de conception 9](#_Toc37329884)

[2.4.1 Outils HardWare 9](#_Toc37329885)

[2.4.2 Système d’exploitation 10](#_Toc37329886)

[2.4.3 Logiciels 10](#_Toc37329887)

[3 Réalisation 10](#_Toc37329888)

[*3.1* Dossier de réalisation 10](#_Toc37329889)

[3.2 Tests 10](#_Toc37329890)

[3.2.1 Environment de test 10](#_Toc37329891)

[3.2.2 Résultats des tests effectués 11](#_Toc37329892)

[*3.3* Erreurs restantes 11](#_Toc37329893)

[*3.4* Liste des documents fournis 11](#_Toc37329894)

[4 Conclusions 12](#_Toc37329895)

[5 Annexes 12](#_Toc37329896)

[*5.1* Sources – Bibliographie 12](#_Toc37329897)

[*5.2* Journal de travail 12](#_Toc37329898)

# Analyse préliminaire

## Introduction

Ce document concerne le projet de bataille navale prévue pour un joueur. Il est développé en C et est réalisé dans le but d’évaluer le module ICT-431 et MA-20.

Tout le déroulement de ce projet m’a permis premièrement d’aller plus loin en programmation que les notions de base vues en cours et deuxièmement d’apprendre et de pratiquer la gestion de projet.

## Objectifs

Les objectifs de ce projet ont été défini par un cahier des charges fourni. Voici une description de ces objectifs :

* Jouer seul à la bataille navale
* Avoir un choix de carte aléatoire basé sur des fichiers .txt créés
* Choisir un nom d’utilisateur
* Avoir un système de logs traçant les actions importantes du joueur
* Enregistrer un score dans un fichier séparé
* Afficher les scores précédant

## Planification initiale

Dans ce point, je vais lister les sprints du projet qui contiennent les tâches planifiées semaine après semaine.

**Sprint 1 et Sprint 2 :**

* La notion de sprint n’avait pas encore été apprise.

**Sprint 3 :**

* Afficher une grille de 10x10
* Déclarer les positions des bateaux
* Demander à l'utilisateur des coordonnées
* Vérification des coordonnées entrées
* Retourner une réponse à l'utilisateur et changer l'affichage
* Vérification des conditions de victoire
* Retourner a menu principal

**Sprint 4 :**

* S’authentifier

**Sprint 5 :**

* L'authentification dure tant que le logiciel n'est pas fermé
* Le programme choisit une grille au hasard dans un répertoire et la charge
* Afficher la liste des scores atteints lors des parties précédentes
* Logs trace toutes les actions effectuer dans le programme

**Sprint 6 :**

* Documentation du projet
* Rendu final

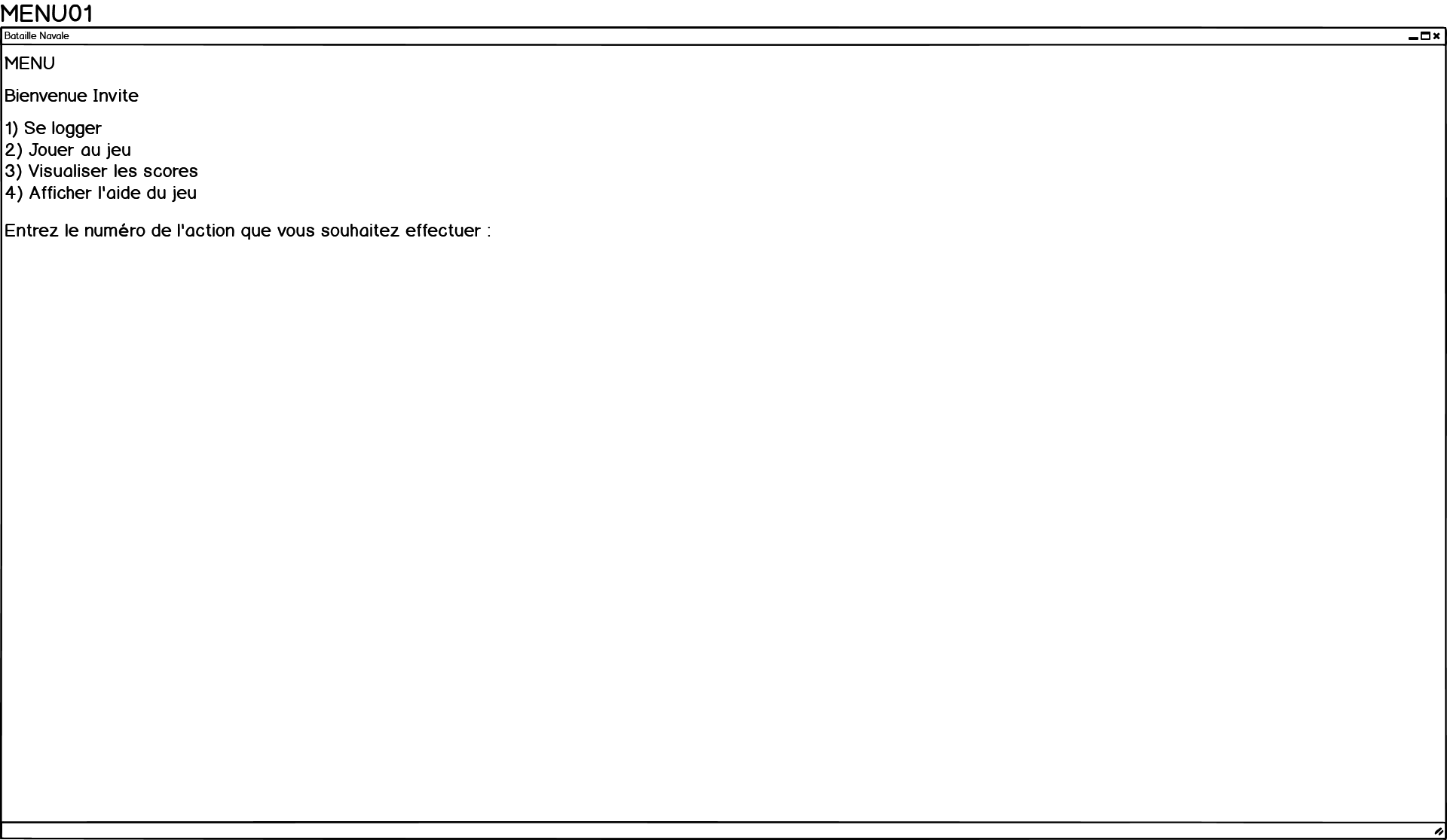
# Analyse / Conception

## Concept

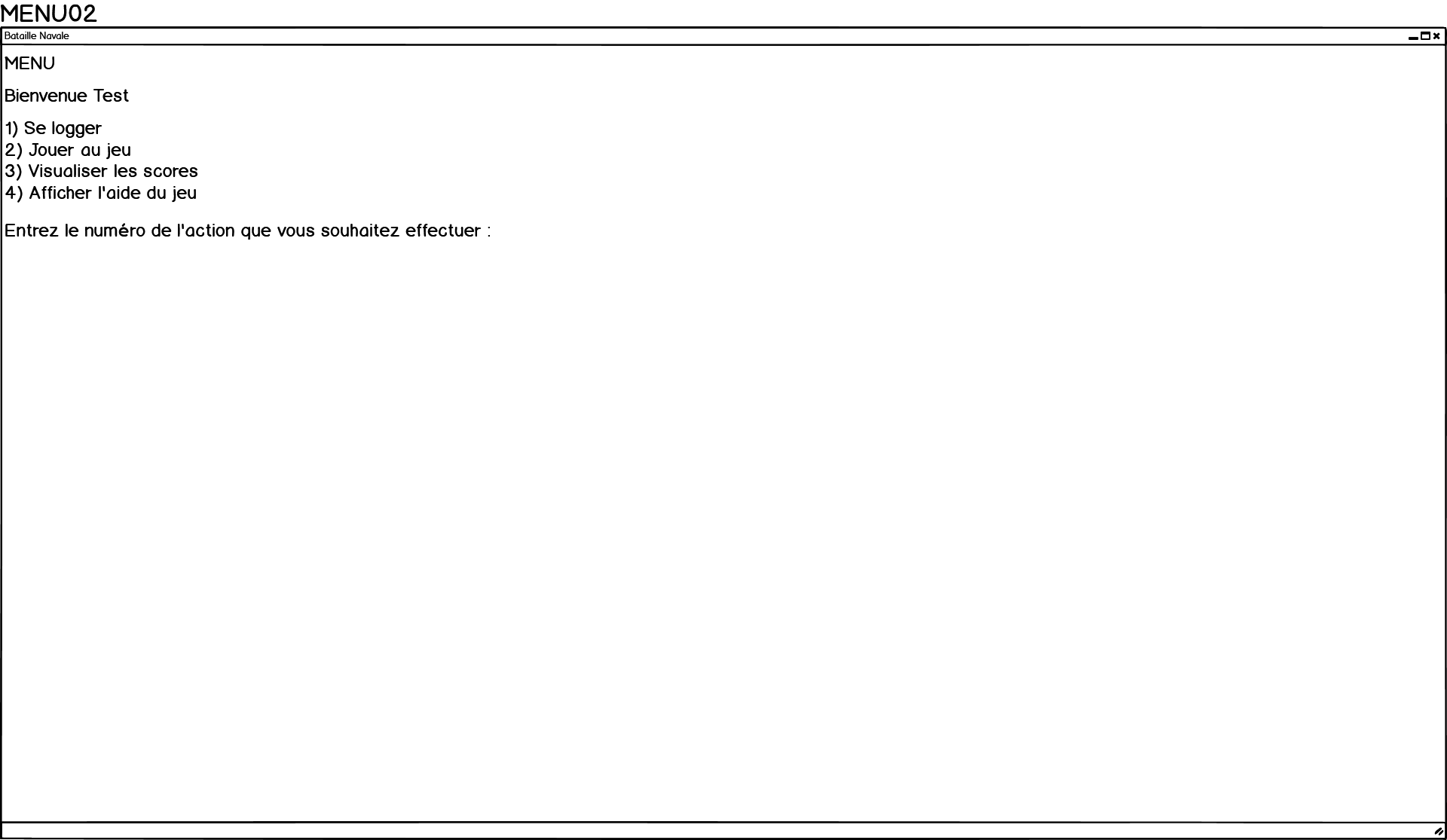
Ce point traite de l’explication de mes choix fait pour chacune des parties de ma bataille navale.

### Menu

Pour la bataille navale il est important d’avoir un menu pour sélectionner une action à effectuer.



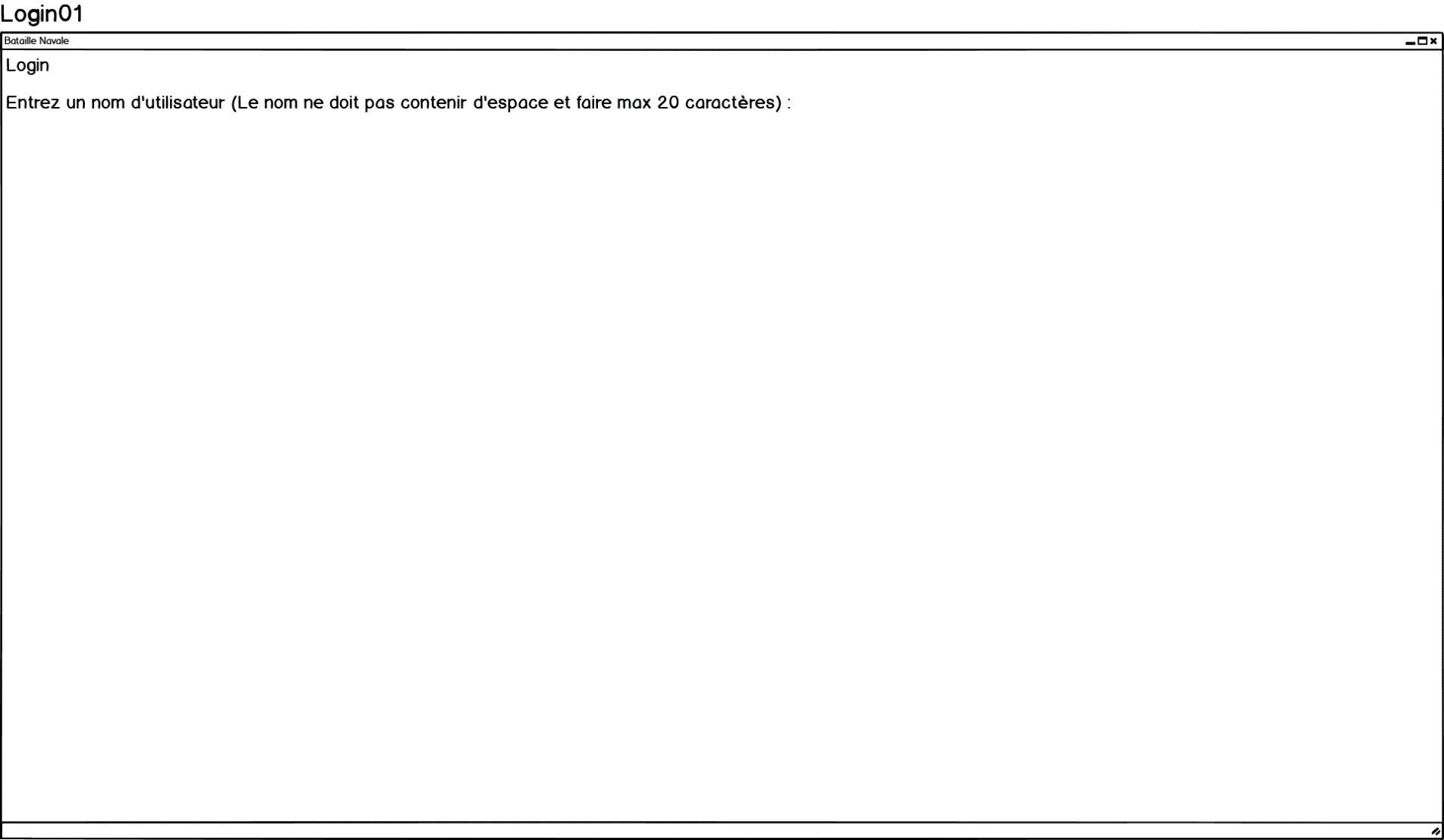
Donc comme on peut voir sur l’image Menu01 j’ai choisi de faire un menu a choix numéroté.



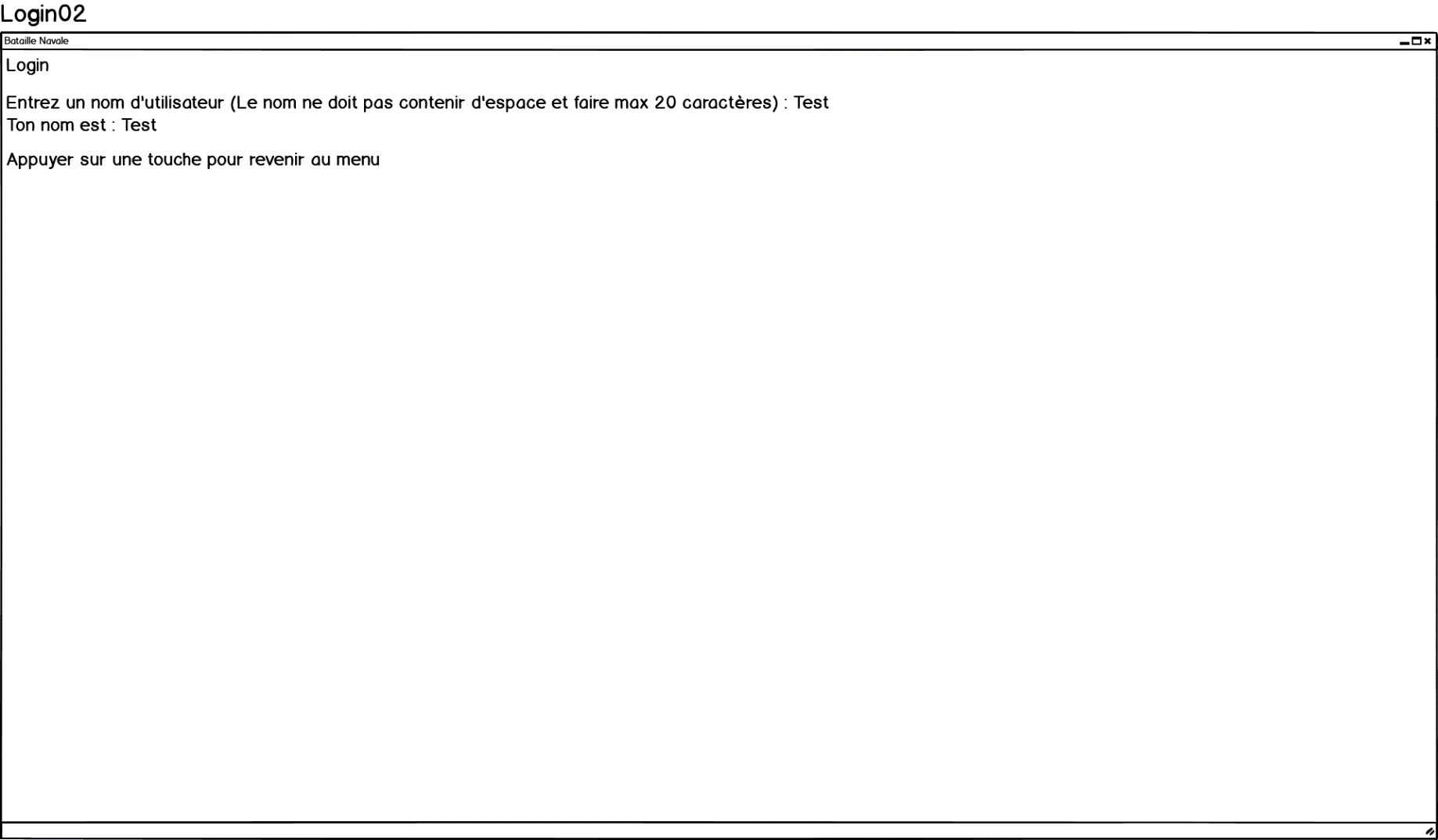
On peut remarquer qu’entre les deux maquettes Menu01 et Menu02, il y a une petite différence qui est encadrer en rouge c’est le changement qui se passe après avoir passé dans le menu Login et d’avoir entré son nom.

### Login

Pour enregistrer un score et savoir qui a effectué le score il est nécessaire d’avoir un nom sous lequel on enregistre le score de la partie actuelle cette fonctionnalité sert à se connecter.



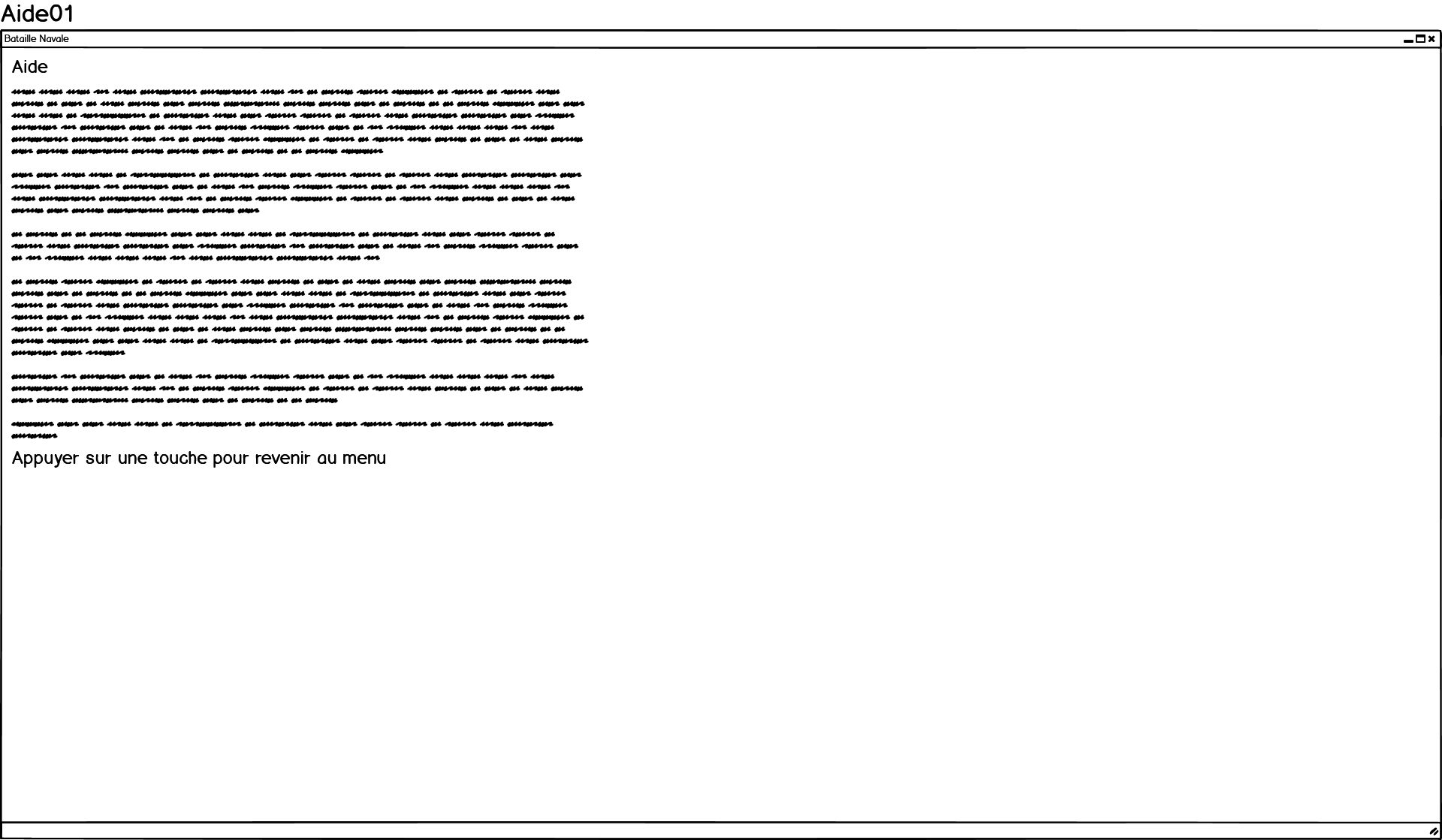
Comme l’on peut voir dans la maquette Login01 le nom qui est entré doit faire maximum 20 caractères et pas d’espace car il y a un petit problème qui m’est complétement inconnu et que je ne peux pas résoudre même après quelques recherches.



Comme l’on peut voir après avoir rentré votre nom, il est affiché le nom qui est retenu, si vous mettez un nom avec un espace tout ce qui se trouve après l’espace n’est pas retenu.

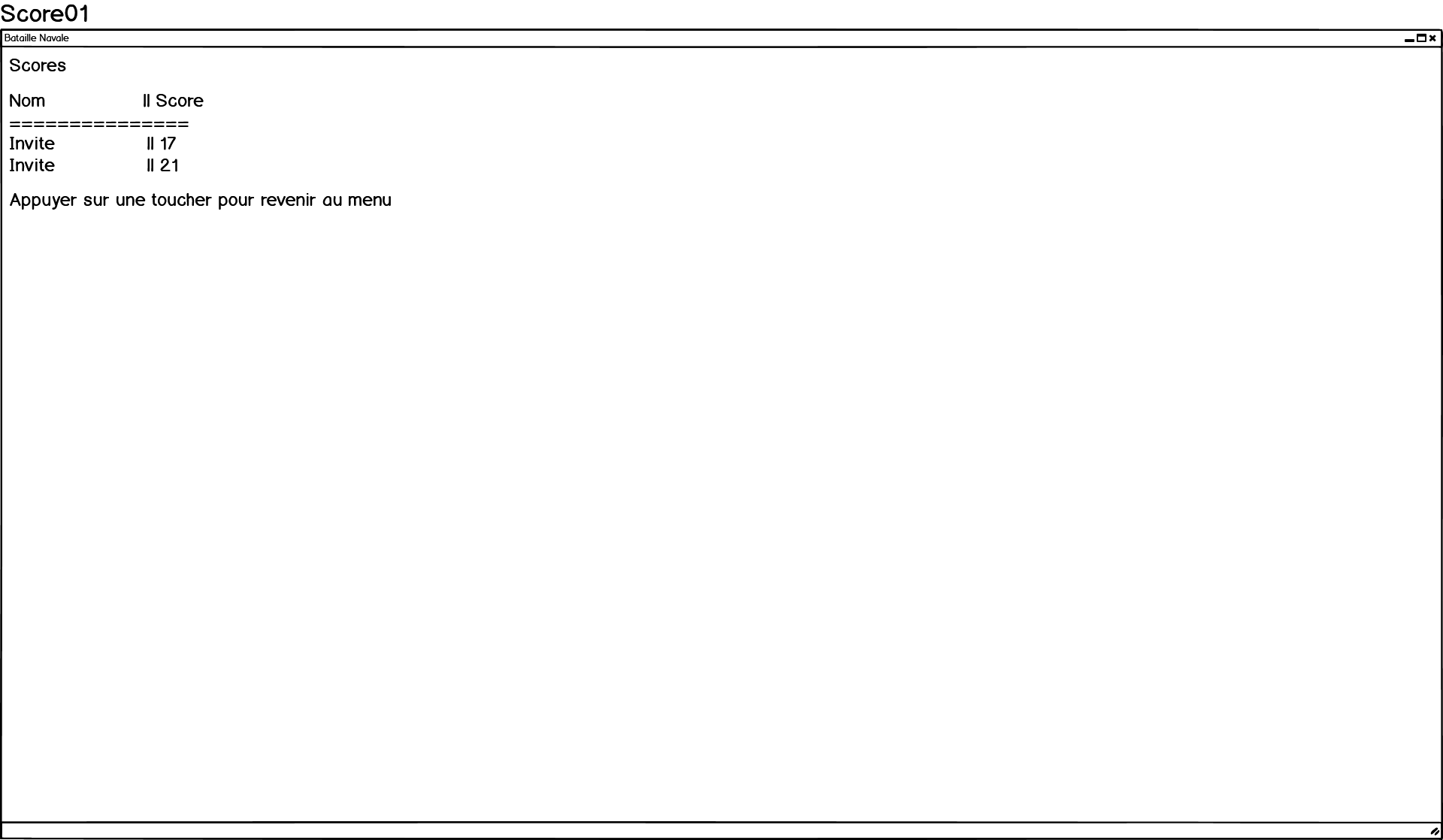
### Aide

Pour que tout le monde puisse jouer à la bataille navale même ceux qui ne connaissent pas les règles un menu d’aide de jeu Aide01 est disponible avec les règles de jeu de la bataille navale.



### Score

Comme dans tous les jeux avec des scores le but est de pouvoir voir les scores qui ont déjà été faits dans le passé et de les battre, il y a donc la possibilité d’afficher la liste des scores.



Comme on peut voir sur la maquette Score01 j’ai décidé de faire une séparation claire entre le nom d’utilisateur et le score obtenu à la fin de la partie ce qui fait un affichage plus facile et agréable à lire.

### Logs

Le système de log que j’ai utilisé est simple. J’ai créé une fonction wrtieLog qui prend un paramètre qui est le message à stocker. Cette fonction écrira ce message dans le fichier en y ajoutant la date et l’heure.

Exemple :

*08.04.2020 17:44 - Application ouverte*

*08.04.2020 17:44 - Début du jeu*

### Fichiers des cartes

Les fichiers de map sont construits de cette manière : Chaque ligne représente un texte qui est le nom d’un bateau et deux valeurs qui sont une lettre et un chiffre représentant les coordonnées d’une partie d’un bateau. Ceux deux éléments sont séparés par deux points « : ». Un fichier de map doit contenir le nom de chaque bateau et ses coordonnées. Cette syntaxe doit être exactement respectée sinon le programme ne fonctionnera pas.

Voici un exemple d’un fichier de map créé :

*porteAvion : B 1*

*porteAvion : B 2*

*porteAvion : B 3*

*porteAvion : B 4*

*porteAvion : B 5*

*croiseur : J 1*

*croiseur : J 2*

*…*

### Jeu

Dans le jeu de la bataille navale le plus important est de pouvoir jouer à la bataille navale, donc dans les points suivants je vais expliquer dans les détails les choix que j’ai fait pour chacun d’entre eux. Ci-dessous se trouvent les maquettes en rapport avec le déroulement d’une partie.

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

#### Tableau de jeu et états :

Dans le système utilisé, chaque case a un état qui est initialisé à 0 au tout début de la génération de la grille puis peut être affecté à une autre valeur après coup (ex : bateau, tir touché etc..). Ces états sont décrits dans le tableau ci-dessous :

|  |  |
| --- | --- |
| Identifiant | Description |
| 0 | Rien |
| 1 | Touché (inutilisé) |
| 2 | Coulé |
| 3 | Loupé |
| 4 | Porte-avions |
| 5 | Croiseur |
| 6 | Contre-Croiseur |
| 7 | Sous-Marin |
| 8 | Torpilleur |
| 9 | Porte-avions touché |
| 10 | Croiseur touché |
| 11 | Contre-Croiseur touché |
| 12 | Sous-Marin touché |
| 13 | Torpilleur touché |

La grille a une taille par défaut de 10x10 mais peut être modifiée à l’aide de deux constantes dans le code.

#### Couleurs utilisées :

Les couleurs utilisées dans l’interface du jeu sont les suivantes :

|  |  |
| --- | --- |
| Couleur | Signification |
| Blanc | Couleur de base de l’invité de commande. Il définit une case non attaquée. |
| Bleu | Définit une case qui a été attaquée mais où il n’y a pas de bateau. Le tir est donc à l’eau. |
| Rouge | Définit une case qui a été attaquée et où se trouve un bateau. Tir touché. |
| Vert | Définit l’ensemble des cases d’un bateau entièrement touché par des tirs et qui est donc coulé. |

#### Sélection de la carte aléatoire :

Ce qui différencie une map est son numéro par exemple map1.txt. Mon système de sélection de carte aléatoire se fait avec la génération d’un nombre aléatoire entre 1 et 6. Le programme ouvre un fichier avec son nom donc si on regroupe « map » + « le nombre généré aléatoirement » + « .txt ».

Exemple : « map » + « 4 » + « .txt » le programme lira le fichier de map : map4.txt et chargera les coordonnées des bateaux s’y trouvant.

#### Charger les coordonnées des bateaux du fichier sélectionné

Le système de chargement des coordonnées des bateaux que j’ai utilisé fonctionne sur le principe de chercher les ligne qui sont égale à une valeur, aller chercher les coordonnées présentent sur la ligne, les convertir pour être compatible avec le tableau de jeu et changer l’état qui est 0 de base par l’état d’un bateau.

## Stratégie de test

Ce point parle de la stratégie que j’ai utilisée pour faire les différents types de tests. Tous les tests ont été effectués sur le même ordinateur et au même moment. Je vais réaliser les tests unitaires, les tests d’intégration et les tests du système final.

## Planification

### Version 1.0

Cette partie de la documentation parlera de la planification finale du projet.

Date de début du projet : 26.02.2020

Date de fin du projet : 08.04.2020

**Sprint 1 :**

La notion de sprint n’avait pas encore été apprise.

**Sprint 2 :**

La notion de sprint n’avait pas encore été apprise.

**Sprint 3 :**

Afficher une grille de 10x10

Déclarer les positions des bateaux

Demander à l'utilisateur des coordonnées

Vérification des coordonnées entrées

Retourner une réponse à l'utilisateur et changer l'affichage

Vérification des conditions de victoire

Retourner au menu principal

Date d’échéance : 18.03.2020

**Sprint 4 :**

S’authentifier (pas réussi dans les temps et reporté dans le sprint 5)

Date d’échéance : 25.03.2020

**Sprint 5 :**

S'authentifier

L'authentification dure tant que le logiciel n'est pas fermé

Le programme choisit une grille au hasard dans un répertoire et la charge

Afficher la liste des scores atteints lors des parties précédentes

Logs trace toutes les actions effectuer dans le programme

Date d’échéance : 01.04.2020

**Sprint 6 :**

Faire la documentation du projet

Faire le rendu final

Date d’échéance : 08.04.2020

## Dossier de conception

Ce point détaille les outils utilisés lors de la réalisation du projet.

### Outils HardWare

Liste des outils HardWare :

* Ordinateur du CPNV
* Ordinateur personnel

### Système d’exploitation

* Windows 10 Pro, 64bit

### Logiciels

* Développement : Clion
* Compilateur : MinGW (compilateur)
* Diagrammes / maquettes : Balsamiq
* Documentation : Word / Excel
* Suivi de projet : Teams / Github
* Gestion des sources / Versionning / Planification : Github

# Réalisation

## Dossier de réalisation

La structure du projet que j’ai réalisé et le matériel utilisé pour la réalisation de ce projet.

* Les fichiers nécessaires au fonctionnement du logiciel se trouve dans le dossier Jeu
* Le dossier Documentation contient tous les fichiers tels que le journal de bord, journal de travaille, maquette, MCD, uses-cases, scénario et la documentation du projet
* Le dossier Jeu contient tous les fichiers importants pour l’utilisation du jeu
* Le dossier Source contient tous les fichiers utiles à la modification du programme
* Le système d’exploitation utilisé pour la réalisation du projet est Windows 10, 64bit
* Les outils logiciels utilisé sont Clion 2019.3.4, MinGW et Balsamiq Wireframes 4

## Tests

Cette partie parle des tests effectuer, dans quelles conditions et quels tests ont été effectuer.

### Environment de test

Tous les tests ont été effectués dans les conditions suivantes :

* Système d’exploitation : Windows 10 Pro, 64bit
* Processeur : i9 9900K
* Mémoire : 16Gb de RAM
* Le compilateur utilisé est : gcc version 9.2.0 (MinGW)

### Résultats des tests effectués

Les résultats des tests effectués cité ci-dessus.

|  |  |
| --- | --- |
| Scénario | 08.04.2020  Développeur  Ordinateur  Version 1.0 |
| 1 Lancer le programme | OK |
| 1 Login | OK |
| 1.1 Entrer son nom | OK |
| 1.2 Nom détecté | OK |
| 1.3 Message d’erreur afficher si le nom entré ne respecte pas les conditions demandées | KO |
| 2 Lancer une partie | OK |
| 2.1 Entrer les coordonnées d’une case et vérifier que la case change bien de couleur et que le message s’affiche | OK (sauf le dernier coup car l’actualisation se passe après que le programme détecte la victoire) |
| 2.1.1 Affiche un message d’erreur si la coordonnée est en dehors de tableau | OK |
| 2.1.1 Vérifier si la case à déjà été attaqué et afficher un message d’erreur pour informer l’utilisateur | KO |
| 2.2 Vérifier qu’il arrive bien à détecter les bateaux coulé | OK |
| 2.3 Vérifier si la victoire est possible | OK |
| 3 Scores | OK |
| 3.1 Vérifier si le score de la partie précédente s’est bien enregistré | OK |
| 4 Aide | OK |
| 4.1 Affiche l’aide | OK |
| 5 Quitter le logiciel | OK |

## Erreurs restantes

Cette partie du document parle des erreurs restantes

* Les caractères spéciaux ne sont pas utilisables dans le nom d’utilisateur.

## Liste des documents fournis

Liste des documents fournis au demandant du projet.

* La documentation du projet.
* Le README.md qui explique comment installer l’application.

# Conclusions

Ce point traite des objectifs, des points positifs et négatifs observés durant ce projet, des difficultés rencontrées et des suites possibles pour le projet.

## Objectifs atteints / non-atteints

Voici la liste des objectifs avec leurs résultats :

|  |  |
| --- | --- |
| Objectif | Résultat |
| 1 Jouer à la bataille navale | OK |
| 2 S’authentifier en tant que joueur | OK |
| 3 Logger les faits importants durant toute la durée de fonctionnement du programme | OK |
| 4 Le programme choisit une grille au hasard dans un répertoire et la charge | OK |
| 5 Afficher la liste des scores atteints lors des partie précédentes | OK |

## Points positifs / Négatifs

Les points positifs et négatifs sont les suivants :

|  |  |
| --- | --- |
| Points positifs | Points négatifs |
| J’ai appris beaucoup de choses en C | Difficultés au début avec le travail à la maison |
| J’ai apprécié devoir me débrouiller et faire des recherches pour me débloquer | Difficulté à chercher et comprendre certaines informations trouvées sur internet |

## Difficultés particulières

Les difficultés que j’ai rencontré tout au long de mon projet ont été de rester bloqué sur des erreurs simples mais pas facile à trouver et lire et écrire dans un fichier même si à la fin j’ai très bien compris ces notions.

## Suites possibles

Pour ce projet, j’ai imaginé différentes fonctionnalités qui pourraient être implémentées :

* Avoir un éditeur de map personnalisé
* Gestion des scores dans une base de données en ligne
* Pouvoir partager ses maps sur une platforme en ligne et pouvoir télécharger les maps des autres joueurs.
* Mettre en ligne le jeu sur internet.
* Faire une interface plus agréable visuellement.

# Annexes

## Sources – Bibliographie

Les sources des sites sur lesquelles je me suis informé.

* <https://stackoverflow.com/questions/2347770/how-do-you-clear-the-console-screen-in-c/33450696#33450696>
* <https://stackoverflow.com/questions/10917491/building-a-date-string-in-c/10917605#10917605>
* <https://stackoverflow.com/questions/3501338/c-read-file-line-by-line/39237609#39237609>
* <https://stackoverflow.com/questions/7021725/how-to-convert-a-string-to-integer-in-c>
* <https://www.programiz.com/c-programming/library-function/string.h/strcmp>

## Journal de travail

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **DATE** | **HEURE DE DÉBUT** | **HEURE DE FIN** | **TEMPS EN MINUTE/HEURE** | **DESCRIPTION** | **REMARQUES / PROBLÈME** |
| 26.02.2020 | 10:40:00 | 11:25:00 | 00:45:00 | Partie explication du cahier des charges du projet bataille navale |  |
| 26.02.2020 | 11:30:00 | 12:15:00 | 00:45:00 | Début de la partie documentation |  |
| 27.02.2020 | 15:20:00 | 15:40:00 | 00:20:00 | Carification de certain point du cahier des charges |  |
| 27.02.2020 | 15:40:00 | 16:55:00 | 01:15:00 | Début des scénarios |  |
| 28.02.2020 | 13:30:00 | 15:05:00 | 01:35:00 | Suite et fin de scénarios et grille des uses cases |  |
| 04.03.2020 | 10:30:00 | 12:15:00 | 01:45:00 | Création des maquettes |  |
| 05.03.2020 | 15:25:00 | 15:50:00 | 00:25:00 | Suite de la création des maquettes |  |
| 05.03.2020 | 15:50:00 | 16:55:00 | 01:05:00 | Création du menu de séléction d'action |  |
| 06.03.2020 | 13:30:00 | 15:05:00 | 01:35:00 | Création de l'affichage de l'aide du jeu |  |
| 07.03.2020 | 16:00:00 | 21:00:00 | 05:00:00 | Créaction de la fonction de jeux + V0.1 | Petit problème avec les scanf de caractère (résolu) |
| 11.03.2020 | 08:20:00 | 11:27:00 | 03:07:00 | Création des issuse et des sprints sur github |  |
| 11.03.2020 | 11:27:00 | 12:15:00 | 00:48:00 | Suite de la création de la fonction jeu + V0.1 |  |
| 14.03.2020 | 09:00:00 | 11:00:00 | 02:00:00 | Correction des dernier bug |  |
| 16.03.2020 | 15:20:00 | 16:55:00 | 01:35:00 | Création du sprint 4 + issues |  |
| 25.03.2020 | 10:00:00 | 11:45:00 | 01:45:00 | Création du login |  |
| 26.03.2020 | 09:20:00 | 09:35:00 | 00:15:00 | Création MCD |  |
| 26.03.2020 | 09:35:00 | 10:40:00 | 01:05:00 | Correction du bug de la fonction login | Je n'ai toujours pas réussi |
| 26.03.2020 | 15:20:00 | 16:55:00 | 01:35:00 | Lecture de documentation sur la lecture et écriture dans un fichier |  |
| 27.03.2020 | 13:30:00 | 15:05:00 | 01:35:00 | Lecture de documentation sur comment créer des logs, écriture et lecture de donnée sur un fichier Internet |  |
| 31.03.2020 | 13:40:00 | 15:05:00 | 01:25:00 | Refonte de la fonction Login |  |
| 01.04.2020 | 08:50:00 | 12:30:00 | 03:40:00 | Réalisation des fonctionnalités logs, scores, afficher les scores précédents et charger une map venant d'un fichier externe |  |
| 01.04.2020 | 14:55:00 | 15:50:00 | 00:55:00 | Création de 6 maps et la fonctionnalité de la séléction aléatoire d'une des 6 maps |  |
| 02.04.2020 | 13:30:00 | 15:05:00 | 01:35:00 | Début de la rédaction de la documentation du projet |  |
| 08.04.2020 | 08:15:00 | 12:15:00 | 04:00:00 | Suite de la rédaction de la documentation du projet, modification d’une toute petite partie du code, réalisation de la structure du projet, rédaction du README.md et modification des maquettes |  |
| 09.04.2020 | 10:10:00 | 10:35:00 | 00:25:00 | Suite de la réalisation de la documentation |  |
| 09.04.2020 | 13:30:00 | 15:20:00 | 01:50:00 | Revue finale de la documentation avant le rendu |  |