Projet Bataille Navale

Antoine Roulin

# Table des matières

[Table des matières 2](#_Toc37260854)

[1 Analyse préliminaire 3](#_Toc37260855)

[1.1 Introduction 3](#_Toc37260856)

[*1.2* Objectifs 3](#_Toc37260857)

[1.3 Planification initiale 3](#_Toc37260858)

[2 Analyse / Conception 5](#_Toc37260859)

[2.1 Concept 5](#_Toc37260860)

[2.1.1 Menu : 5](#_Toc37260861)

[2.1.2 Login : 5](#_Toc37260862)

[2.1.3 Aide : 6](#_Toc37260863)

[2.1.4 Score : 6](#_Toc37260864)

[2.1.5 Jeu : 7](#_Toc37260865)

[2.1.6 Logs : 8](#_Toc37260866)

[2.1.7 Fichiers des cartes : 8](#_Toc37260867)

[*2.2* Stratégie de test 8](#_Toc37260868)

[*2.3* Planification 9](#_Toc37260869)

[2.3.1 Version 1.0 9](#_Toc37260870)

[2.4 Dossier de conception 10](#_Toc37260871)

[2.4.1 Outils HardWare utilisé : 10](#_Toc37260872)

[2.4.2 Système d’exploitation utilisé : 10](#_Toc37260873)

[2.4.3 Logiciels utilisé : 10](#_Toc37260874)

[3 Réalisation 11](#_Toc37260875)

[*3.1* Dossier de réalisation 11](#_Toc37260876)

[*3.2* Description des tests effectués 11](#_Toc37260877)

[*3.3* Erreurs restantes 11](#_Toc37260878)

[*3.4* Liste des documents fournis 11](#_Toc37260879)

[4 Conclusions 12](#_Toc37260880)

[5 Annexes 12](#_Toc37260881)

[*5.1* Sources – Bibliographie 12](#_Toc37260882)

[*5.2* Journal de travail 12](#_Toc37260883)

# Analyse préliminaire

## Introduction

Le projet effectuer est une bataille navale 1 joueur faite en C, ce projet est réalisé dans le but d’être évaluer pour une note de module. Ce projet m’a permis d’aller plus loin que les notions de base en C effectuer dans le module ICT-403 et d’apprendre comment structurer un projet.

## Objectifs

L’objectif de ce programme est de pouvoir jouer à la Bataille Navale seul sur des cartes créer, se login avec un nom pour l’enregistrement du score, enregistrer un score, afficher tous les scores précédant et afficher l’aide du jeu. Les objectifs cités précédemment ont tous été atteint.

## Planification initiale

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

**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 a menu principal

**Sprint 4 :**

S’authentifier (pas réussi)

**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

**Sprint 6 :**

Faire la documentation du projet

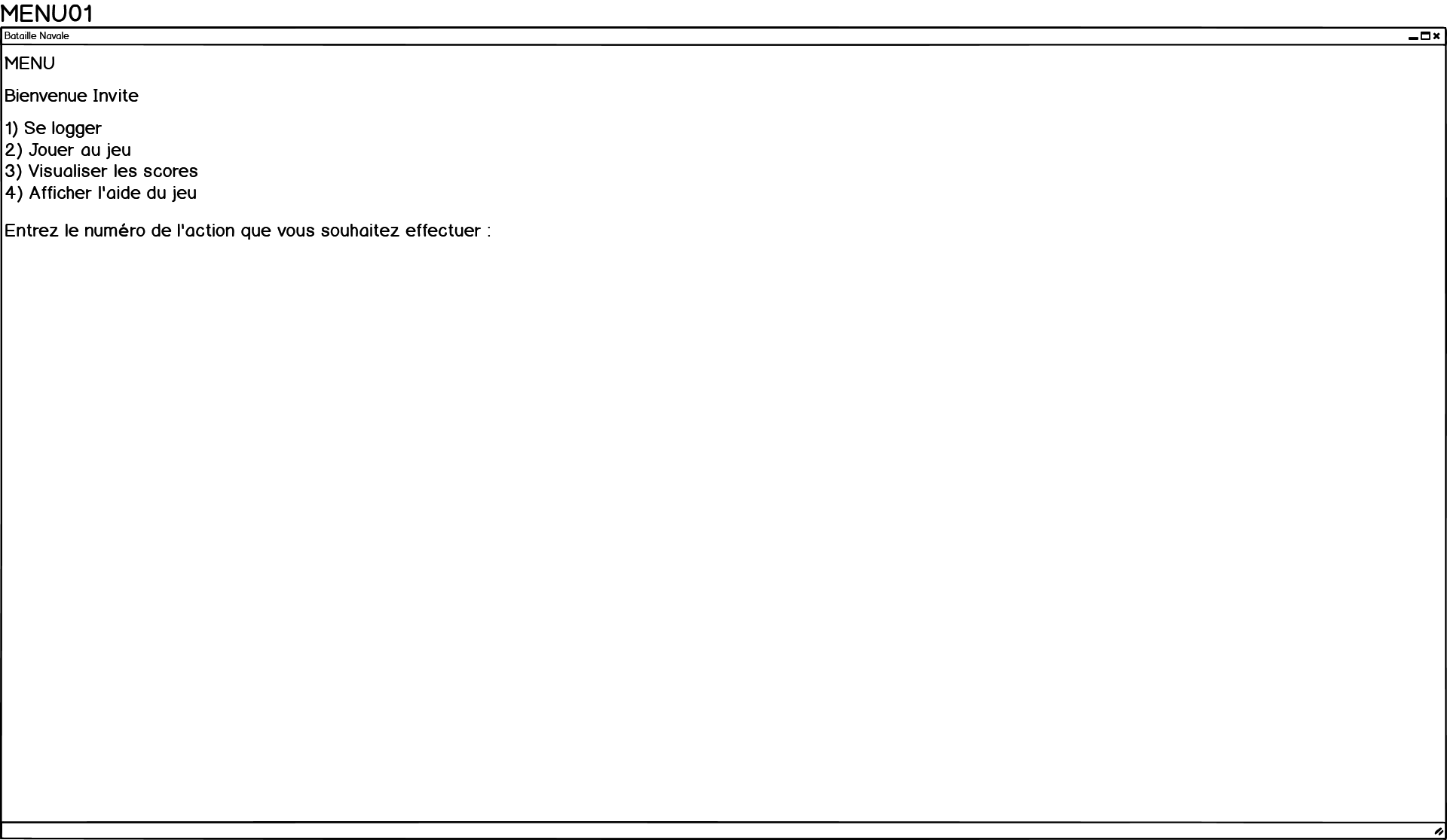
Faire le rendu final

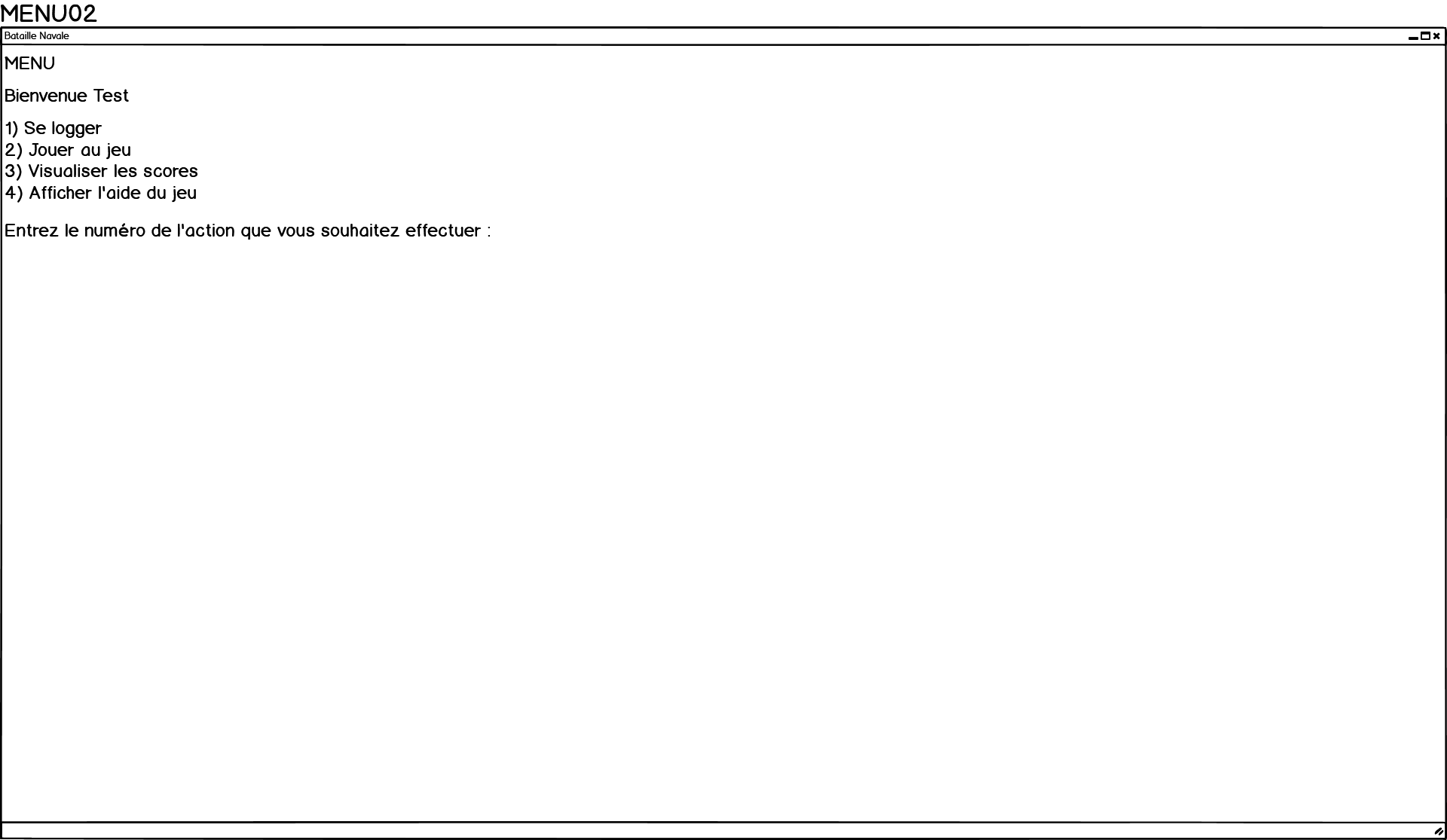
# Analyse / Conception

## Concept

Dans ce point je vais expliquer mes choix fait pour chacune des parties de ma bataille navale.

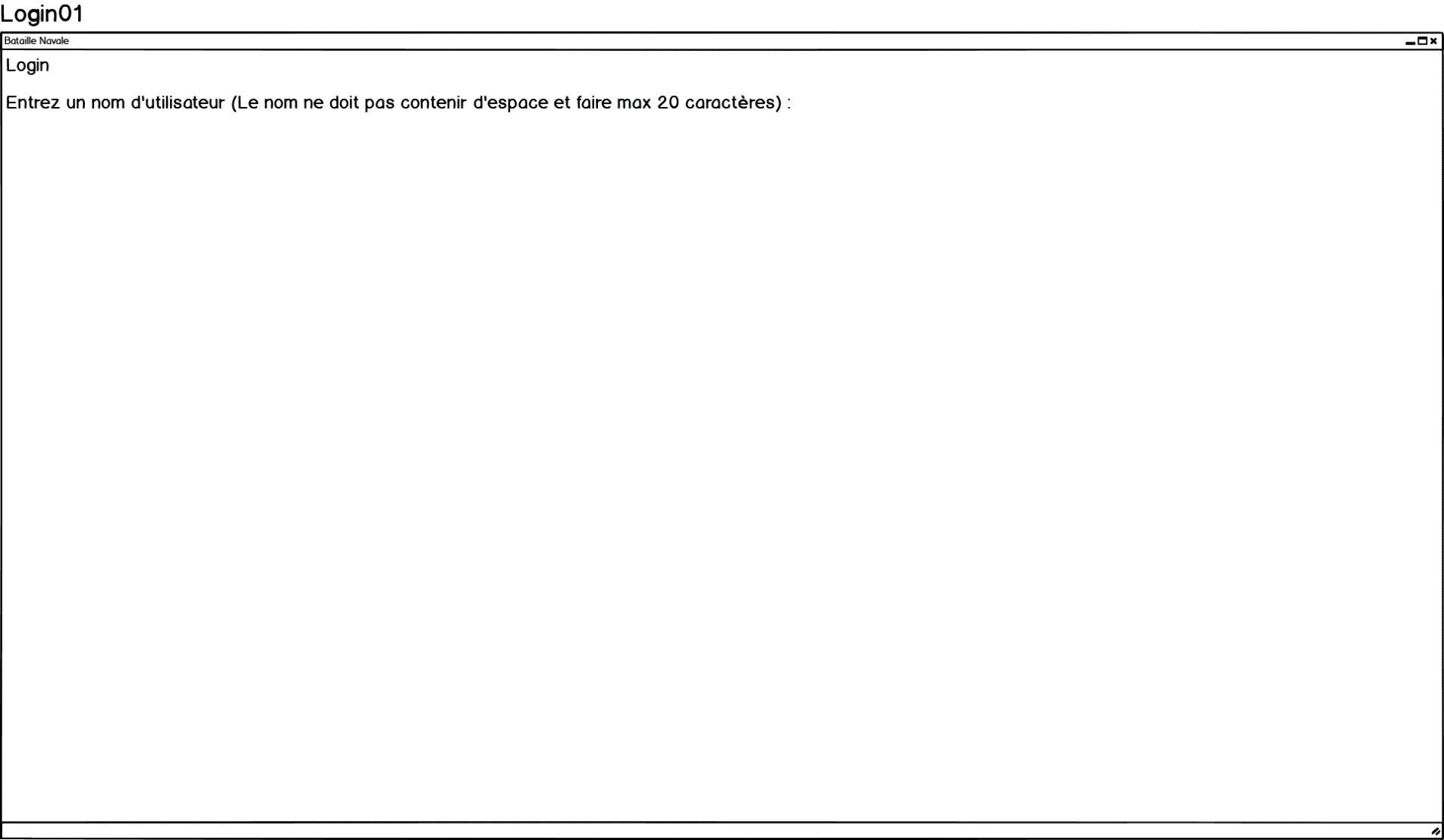
### Menu :

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

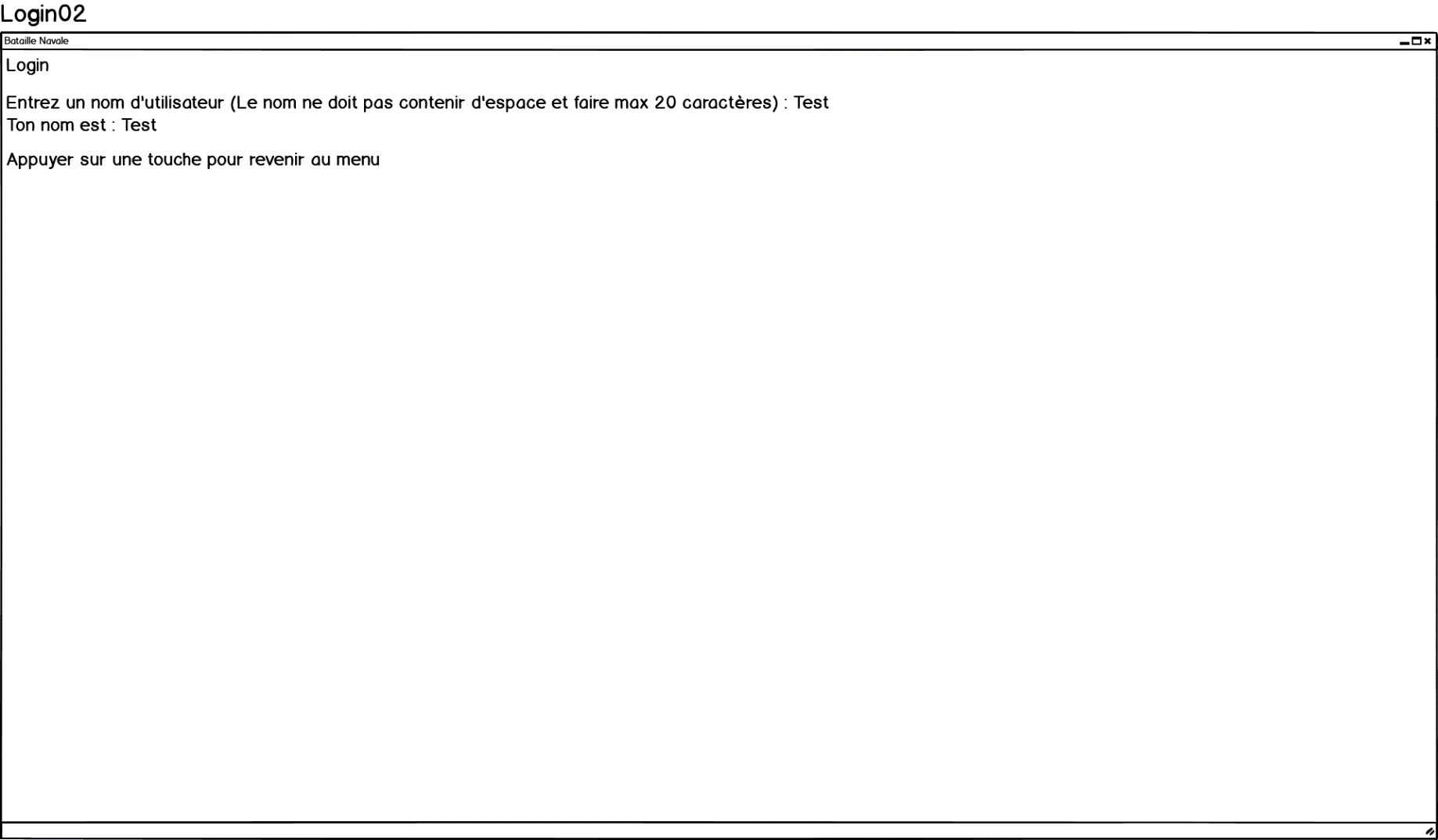
Donc comme on peut voir sur l’image Menu01 j’ai choisi de faire un menu a choix numéroté.

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 rentré son nom.

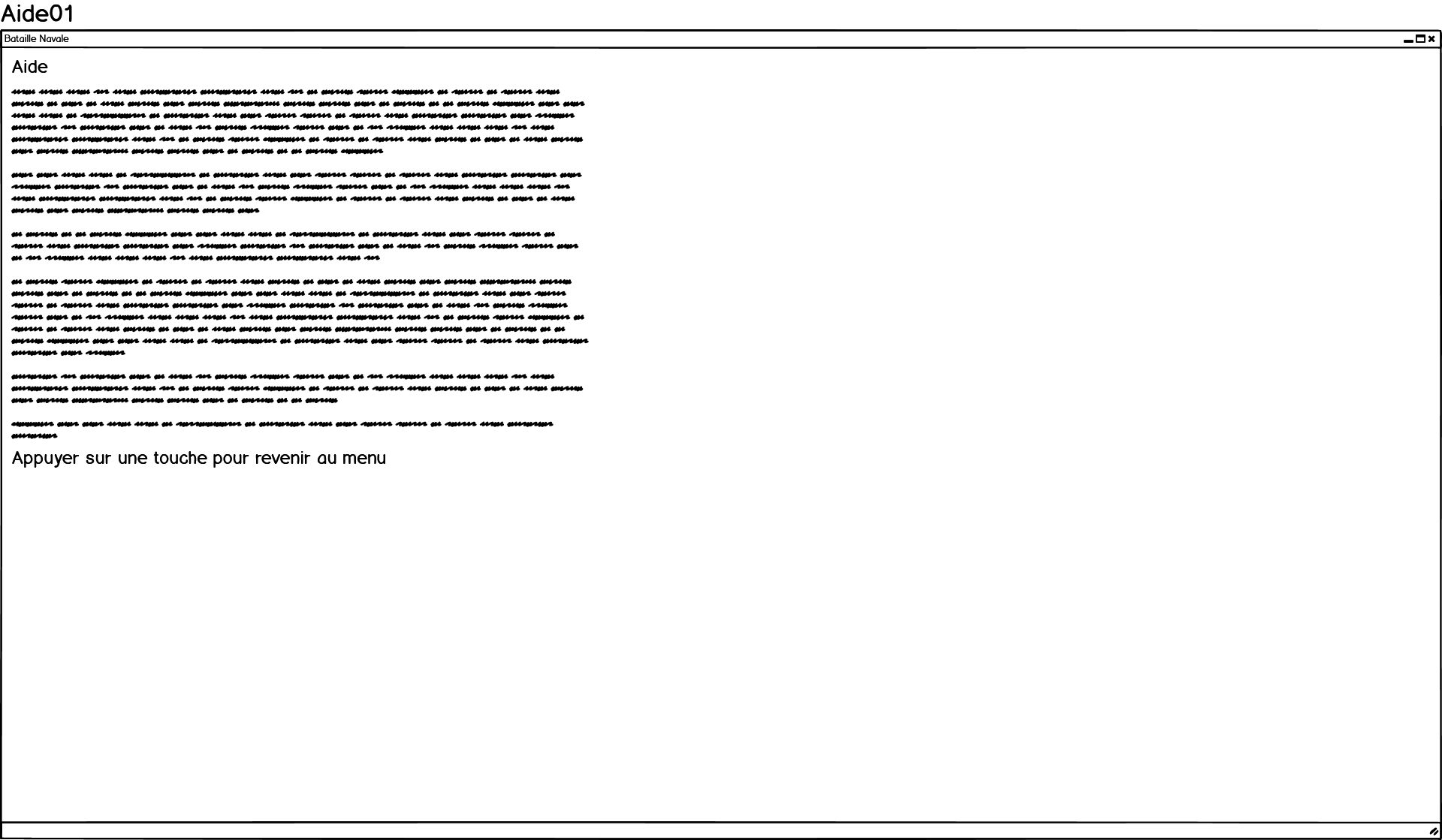
### Login :

Pour enregistrer un score et savoir qui a effectué le score il est nécessaire d’avoir un nom sous le quelle on enregistre le score de la partie actuelle cette fonctionnalité sert donc comme dit dans le titre à se login.

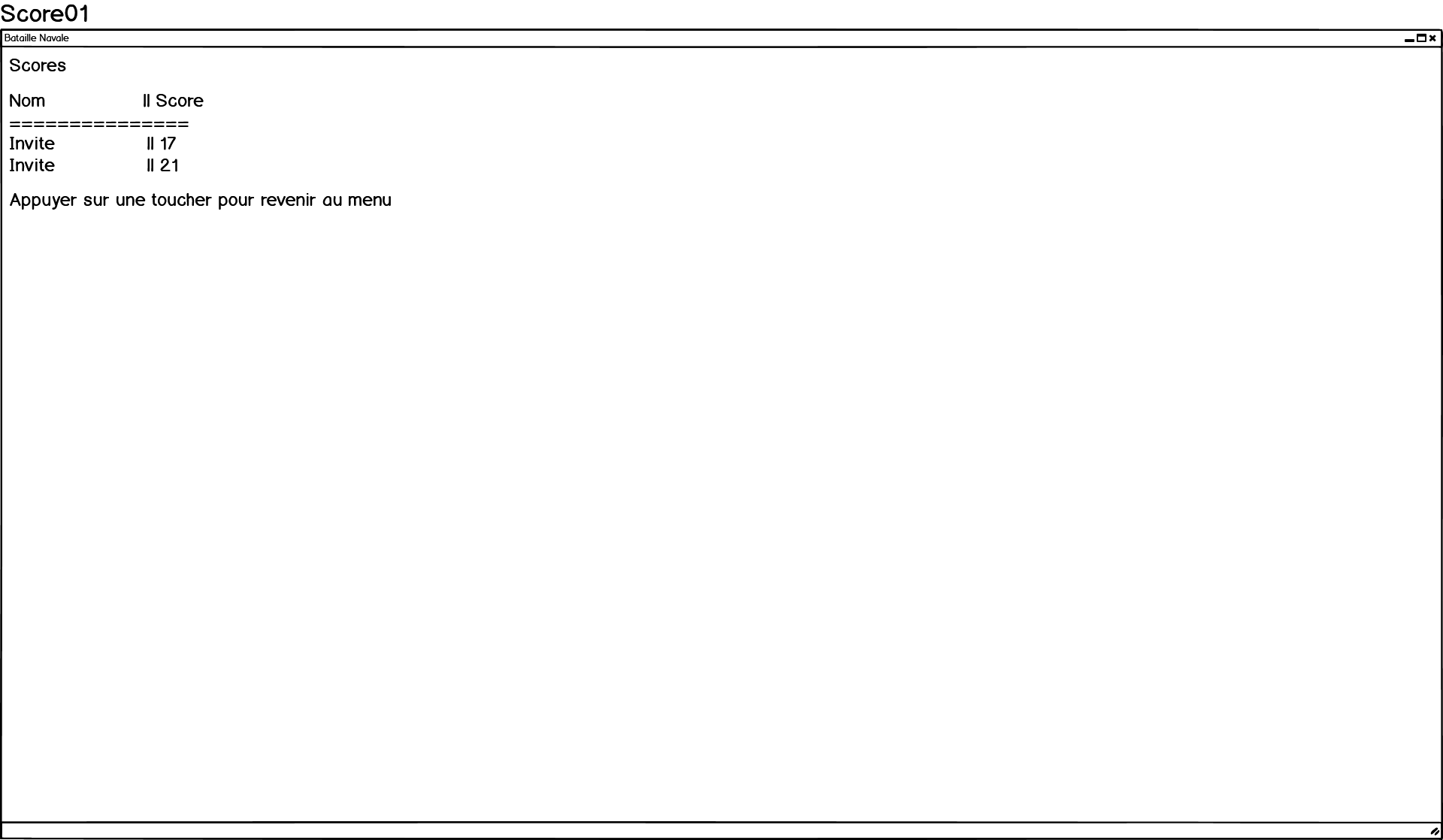
Comme l’on peut voir dans la maquette Login01 le nom qui est entré doit faire maximum 20 caractères et pas d’espace car il y a un petit problème qui m’est complétement inconnu et que je ne peux pas résoudre même après quelques recherches.

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 voire les scores qui ont déjà été fait 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 a la fin de la partie ce qui fait un affichage plus facile et agréable à lire.

### Jeu :

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



#### Tableau de jeu et états :

Le système que j’utilise est simple, il y a un état pour chaque bateau au moment où ils ne sont pas encore découvert (touché), un état qu’ils acquièrent en se faisant toucher, un état pour les tirs loupé et un état pour les bateaux coulé.

#### Couleurs utilisées :

Dans ce point je vais parler du choix que j’ai fait au niveau des couleurs.

J’ai utilisé 4 couleurs assez logique pour 3 d’entre elles mais un peu moins pour la dernière. En premier le blanc qui est la couleur de base de l’invité de commande qui définit une case qui n’as pas été attaqué. En deuxième le bleu qui définit une case qui a été attaqué mais ou il n’y a pas de bateau présent. En troisième le rouge qui définit une case qui a été attaqué et ou il y avait un bateau. En dernier le vert qui définit l’ensemble des cases d’un bateau qui a entièrement été toucher.

#### 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 aléatoire générer (dans ce cas 4) » + « .txt » le programme lira map4.txt et donc chargera les coordonnées des bateaux de la map 4.

#### 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.

### Logs :

Le système de logs que j’ai utilisé est simple à chaque action importante, a chaque action importante une fonction writeLog est appeler et on y rentre le message de l’action importante et la fonction writeLog écrit dans le fichier logs un message composé de la date, l’heure et le message qui a été introduit dans la fonction lors de son appel auparavant.

### Fichiers des cartes :

Les fichiers de maps sont composés du paramètre ; ex. porteAvion et des coordonnées simples avec les lettres et chiffres ex. A 10. Une coordonnée qui peut être lue par mon programme doit être rédiger sous la forme ; porteAvion : A 10. Si la syntaxe n’est pas respectée le programme n’arrivera pas à lire les coordonnées et donc placer des bateaux.

## Stratégie de test

Ce point va parler des tests effectuer sur le programme.

Type de test : Fonctionnelle

|  |  |
| --- | --- |
| **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 |
| 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 |
| 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 |

## Planification

### Version 1.0

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

Date de début du projet :

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)

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 parle des outils utilisé lors de la réalisation du projet.

### Outils HardWare

* Ordinateur du CPNV, Ordinateur personnel

### Système d’exploitation

* Windows 10 Pro, 64bit

### Logiciels

* Clion
* MinGW
* Balsamiq

# 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 est Windows 10, 64bit
* Les outils logiciels utilisé sont Clion 2019.3.4, MinGW et Balsamiq Wireframes 4

## Description des tests effectués

Condition dans les quelles les tests ont été fait.

Windows 10, 64bit

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

* les conditions exactes de chaque test
* les preuves de test (papier ou fichier)
* tests sans preuve: fournir au moins une description

## 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

Lister les documents fournis au client avec votre produit, en indiquant les numéros de versions

* le rapport de projet
* le manuel d'Installation (en annexe)
* le manuel d'Utilisation avec des exemples graphiques (en annexe)
* autres…

# Conclusions

* Les objectifs demandés ont été atteint
* Les points positifs ont été pour moi d’apprendre à me débrouiller par mes propres moyens
* Les points négatifs ont été pour moi de me retrouver bloquer de nombreuses fois sur des erreurs bête et très peu visible
* Les difficultés que j’ai rencontré tout au long de mon projet ont été de rester bloquer sur des erreurs simples mais pas facile a trouvé et lire et écrire dans un fichier même si à la fin j’ai très bien compris ces notions
* Une des fonctionnalités qui pourrait être ajouté serait un créateur de map personnalisé

# Annexes

## Sources – Bibliographie

Liste des livres utilisés (Titre, auteur, date), des sites Internet (URL) consultés, des articles (Revue, date, titre, auteur)… Et de toutes les aides externes (noms)

* C’est ici que vous renseignez les sources notées dans votre journal de travail

## Journal de travail

* Insérez votre journal de travail ici