|  |
| --- |
| Bataille Navale C |

Dossier de projet

Table des matières

[1 Analyse préliminaire 3](#_Toc37226557)

[1.1 Introduction 3](#_Toc37226558)

[*1.2* Objectifs 3](#_Toc37226559)

[1.3 Planification initiale 3](#_Toc37226560)

[2 Analyse / Conception 4](#_Toc37226561)

[2.1 Concept 4](#_Toc37226562)

[2.1.1 Modèles conceptuels de données 4](#_Toc37226563)

[2.1.2 Maquettes 4](#_Toc37226564)

[2.1.3 Use Cases 6](#_Toc37226565)

[*2.2* Stratégie de test 6](#_Toc37226566)

[*2.3* Planification définitive 7](#_Toc37226567)

[*2.4* Dossier de conception 7](#_Toc37226568)

[3 Réalisation 8](#_Toc37226569)

[*3.1* Dossier de réalisation 8](#_Toc37226570)

[3.2 Description des tests effectués 8](#_Toc37226571)

[3.3 Erreurs restantes 9](#_Toc37226572)

[3.3.1 Bug des espaces 9](#_Toc37226573)

[*3.4* Liste des documents fournis 9](#_Toc37226574)

[4 Conclusions 9](#_Toc37226575)

[5 Annexes 10](#_Toc37226576)

[*5.1* Sources – Bibliographie 10](#_Toc37226577)

[5.1.1 URLs 10](#_Toc37226578)

[5.2 Journal de travail (Sans les « remarques/problèmes ») 11](#_Toc37226579)

# Analyse préliminaire

## Introduction

Ce projet est une bataille navale développée en C dans le cadre du module MA-20 (Partie programmation et structure du code) et ICT-431 (Partie documentation et structure d’un projet) du CPNV.

Inventaire des travaux précédents :

* Module MA-03 (Introduction à la programmation en Visual basic)
* Module ICT-403 (Développement en C de manière procédurale)
* Cours OpenClassroom ([Apprenez à programmer en C](https://openclassrooms.com/fr/courses/19980-apprenez-a-programmer-en-c/6760501-entrainez-vous-a-ameliorer-le-jeu-du-plus-ou-moins))

## Objectifs

Le programme doit pouvoir avoir les fonctionnalités suivantes :

* Jouer à la bataille navale à l’aide une grille au hasard dans un répertoire et la charger
* Un menu pour accéder aux fonctionnalités
* Afficher l’aide du jeu
* S’authentifier en tant que joueur
* Logger les faits importants durant toute la durée de fonctionnement de l’application
* Afficher la liste des scores atteints lors des parties précédentes

## Planification initiale

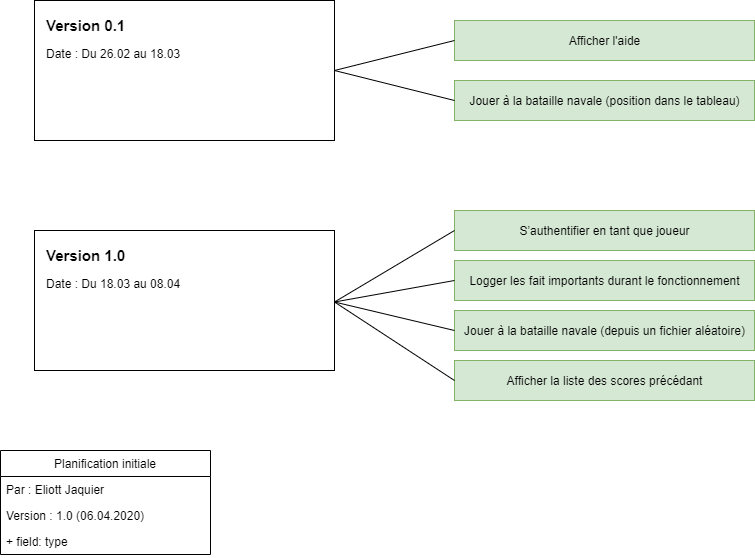


Diagramme 1 - Planification initiale

# Analyse / Conception

## Concept

### Modèles conceptuels de données

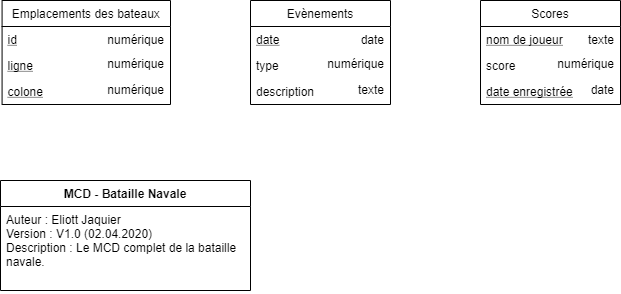
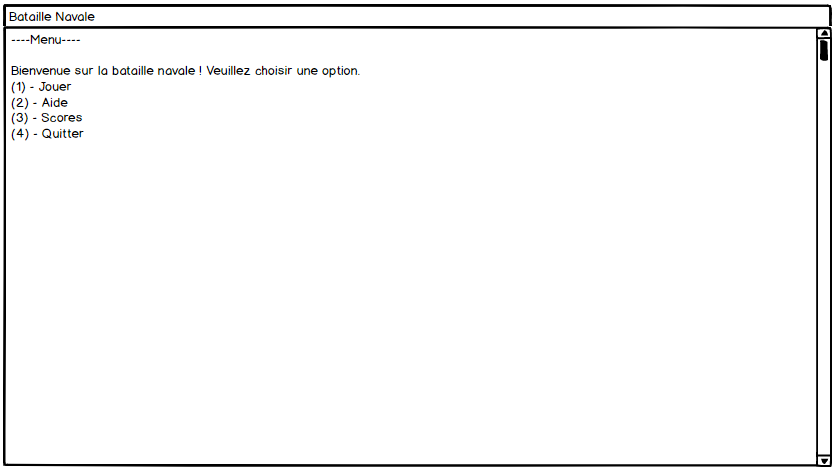
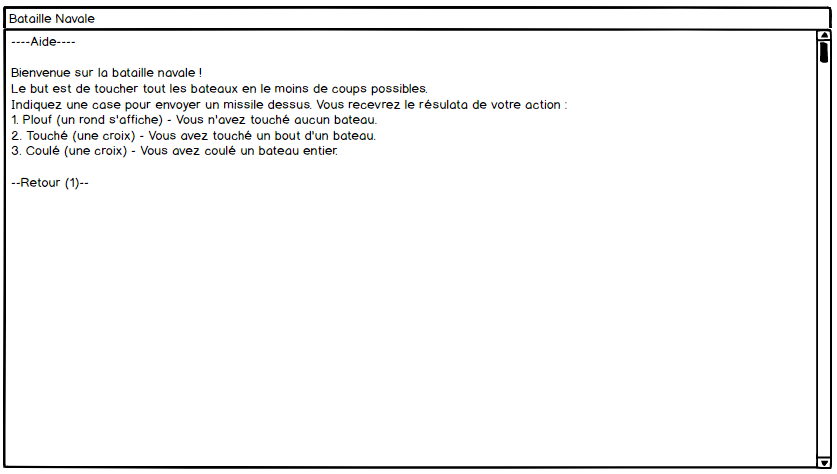


Diagramme 2 - Modèle conceptuel de donnée complet

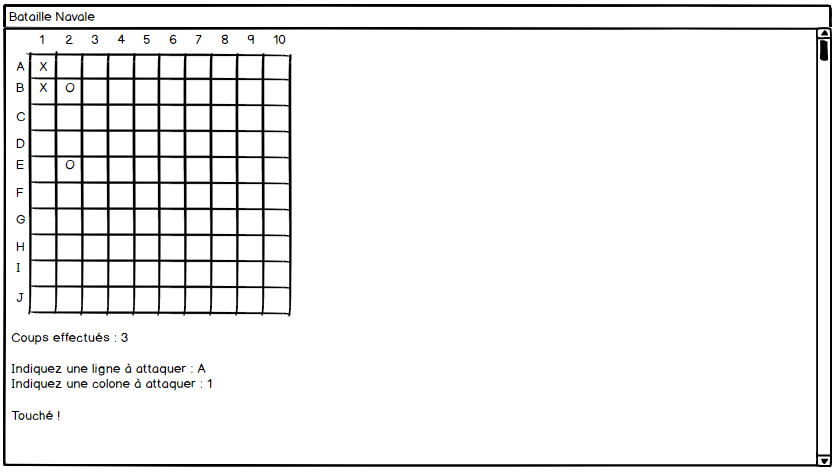
### Maquettes



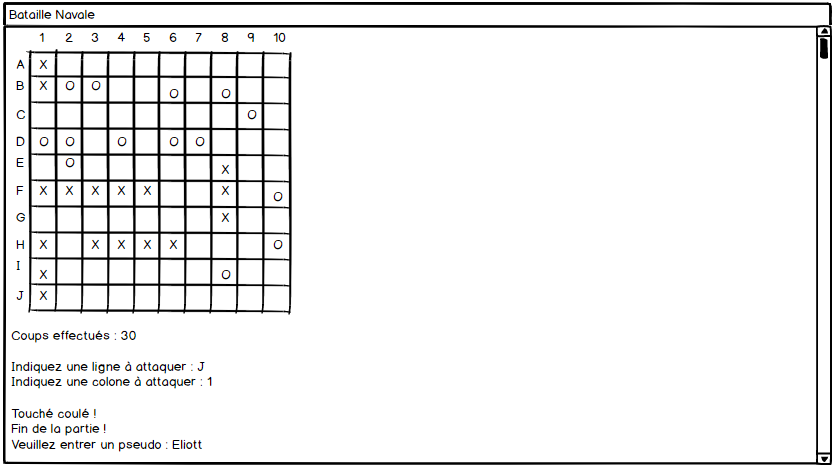
Maquette 1 - Afficher le menu



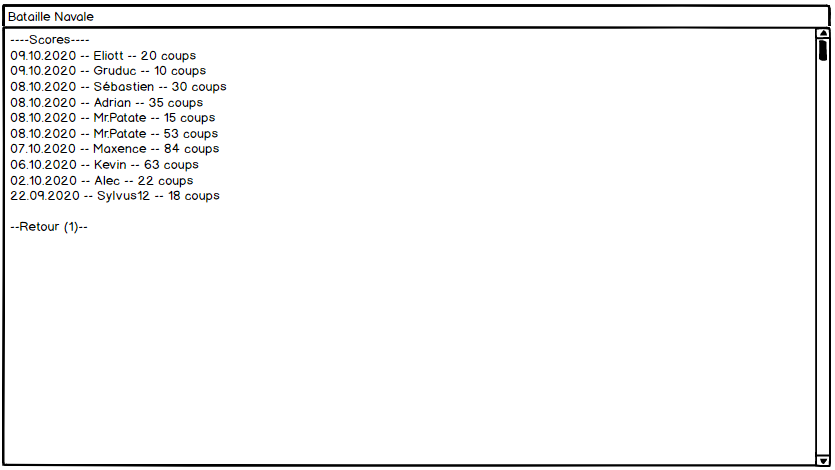
Maquette 2 - Afficher l'aide



Maquette 3 - Jouer une partie



Maquette 4 - Enregistrer le score



Maquette 5 - Afficher les scores

### Use Cases

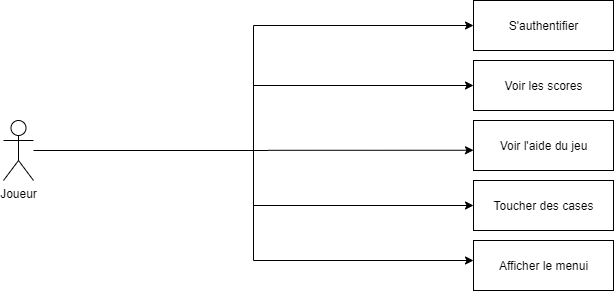


Diagramme 3 - Use Cases

## Stratégie de test

Les tests vont tous se faire sur des ordinateurs dotés d’un système d’exploitation Windows x64 (Axé sur Windows 10). L’ordre des tests va être pris selon le document Excel de UseCasesScénarios. Pour les tests internes (Console intégrée CLion), le projet sera téléchargé depuis GitHub pour être compilé localement avec CLion. Pour les tests externes (version définitive en .exe), les testeurs pourront télécharger le fichier zip qui sera à disposition sur GitHub. Les tests s’axeront principalement sur les tests système (obligatoire pour les possesseurs de la version .exe).

## Planification définitive

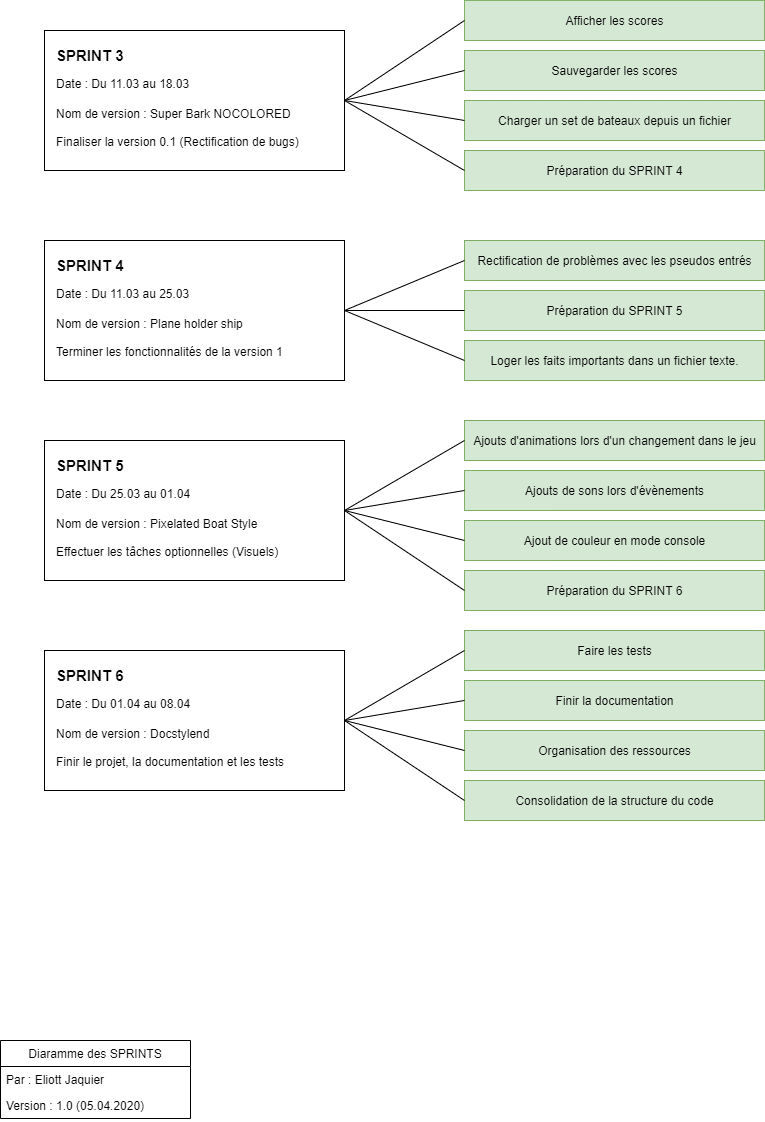


Diagramme 4 - Planification Finale des SPRINT

## Dossier de conception

Le projet a été réalisé avec un ordinateur du CPNV tournant sur Windows 10 ainsi qu’un ordinateur externe étant également sur Windows 10. L’utilisation est prévue et a été testée pour des ordinateurs tournant avec Windows x64.

L’éditeur employé est [Clion de JetBrain](https://www.jetbrains.com/fr-fr/clion/) et le compilateur pris est [MinGW](http://www.mingw.org/).

GitHub est utilisé pour gérer les SPRINTs ainsi que pour servir d’hébergement au repository (Projet ici : [github.com/EliottJaquierCPNV/BatailleNavale](https://github.com/EliottJaquierCPNV/BatailleNavale))

Pour créer les maquettes, [Balsamiq](https://balsamiq.com/) a été utilisé. [DrawIO](https://app.diagrams.net/) a servi pour tous les diagrammes ainsi que pour la conception des MCD.

# Réalisation

## Dossier de réalisation

**Les scripts (version 1.5.1) du projet sont installés dans le répertoire « Programme » du repository GitHub :**

* main.c Contient le lancement et les fonction élémentaires du jeu.
* globalvarsinit.h Les variables globales du jeu et les génériques de fonctions.
* visualevents.c Va gérer les actions spéciales du jeu (Touché, coulé…)
* soundsystem.c Va gérer tous les sons du jeu (sons, musique, loops).
* drawer.c Dessine des images en ASCII (Seulement dans le .exe)
* CMakeLists.txt La liste générée par CLion pour compiler.

**Les bibliothèques utilisées sont :**

* Windows.h Gérer les commandes pour la console externe
* String.h Fonctions utiles pour les tableaux de caractères
* Time.h Utilisé pour récupérer la date et l’heure
* Dirent.h Utilisé pour vérifier si un répertoire existe
* Sounder.exe Utilisé pour jouer un son en lui passant des arguments.

**Le ZIP contenant la version 1.5.1 déjà compilée est composé de :**

* BatailleNavale.exe La version 1.5.1 de la bataille navale compilée.
* config.ini Stock les paramètres du joueur (ex : désactiver les sons)
* logs.txt Stock les dernières informations importantes du programme
* Gameassets Stock les informations des joueurs
* Maps Stock les cartes créées
* Sounds Stock le sounder.exe ainsi que les sons / musiques.

## Description des tests effectués

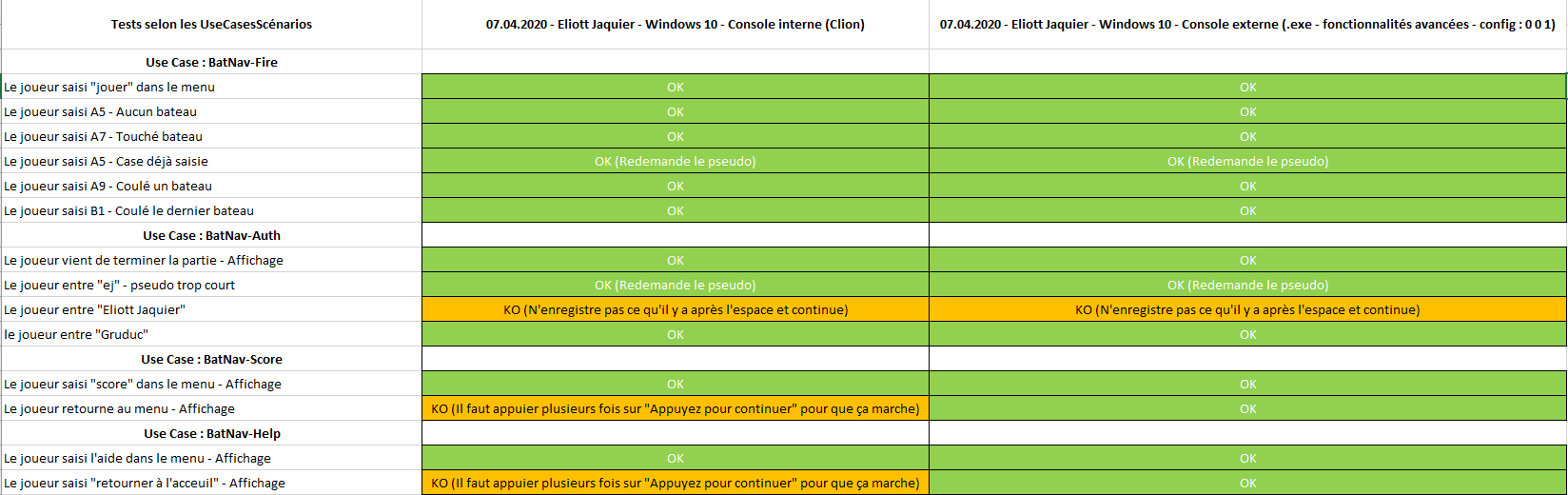


Tableau 1 - Tests selon les UseCasesScénarios

## Erreurs restantes

### Bug des espaces

Les espaces ne sont pas tolérés lors de l’enregistrement du pseudo. Tout ce qui est après sera donc ignoré pour la suite du programme. L’espace est probablement confondu comme étant la fin du tableau lors d’un for. Une solution possible est de trouver une bibliothèque gérant mieux les chaines de caractères que string.h ou d’aller sur un langage qui supporte les strings nativement.

## Liste des documents fournis

* Dossier de projet V.07.04.20
* Manuel d’Installation (Readme) V.01.04.20
* Manuel d’Utilisation (en annexe) V.07.04.20
* Les tests fonctionnels (en annexe) V.07.04.20
* Planification initiale (en annexe) V.06.04.20
* Planification finale des SPRINTS (en annexe) V.06.04.20
* Maquette complète des UseCases (en annexe) V.07.04.20
* Tous les MCD du projet (MCD complet en annexe) V.01.04.20
* Use Cases Scénarios V.01.04.20
* Le cahier des charges V.26.02.20
* Le journal de bord V.06.04.20
* Le journal de travail (en annexe) V.06.04.20
* Références (URLs – en annexe) V.27.03.20

# Conclusions

Tous les objectifs fixés de la 1.0 du cahier des charges ont été accomplis. De plus, des fonctionnalités telle que la prise en charge des couleurs dans la console, des images en ASCII, des animations et des paramètres de configuration ont été ajoutées.

Ce projet a été très intéressant à concevoir sur la partie structuration externe au code (ICT-431) et c’est ici que j’ai appris le plus de nouvelles notions.

Il est tout à fait compréhensible d’avoir choisi le C pour créer ce projet tant que les étudiants ne conçoivent pas de fonctionnalités complexes. Pour créer un jeu-vidéo, j’aurai personnellement choisi un langage plus haut niveau que le C. En effet, avoir les variables « string » au lieu de tableaux de caractères ou de pointeurs ainsi que de « l’orienté objet » auraient aidé pour la réalisation de certaines tâches (comme par exemple : charger des scores ou les enregistrer dans un fichier). Mais il est également compréhensible que l’orienté objet soit une notion qui ne puisse pas encore être abordée au sein des premiers modules de programmation.

La prochaine étape du projet serait probablement de passer en C# ou d’inclure une librairie d’affichage comme la SDL 2 ou encore de rendre le jeu cross-platform.

# Annexes

## Sources – Bibliographie

### URLs

|  |
| --- |
| <https://www.tutorialspoint.com/how-to-clear-console-in-c> |
| <https://stackoverflow.com/questions/8611815/determine-if-char-is-a-num-or-letter> |
| <https://www.quora.com/How-do-I-find-the-length-of-an-array-in-C> |
| <https://stackoverflow.com/questions/10586003/try-catch-statements-in-c> |
| <https://www.geeksforgeeks.org/clearing-the-input-buffer-in-cc/> |
| <https://openclassrooms.com/forum/sujet/vider-le-buffer-de-scanf-90403> |
| <http://sebastienguillon.com/test/jeux-de-caracteres/windows-ascii-fr.html> |
| <https://github.com/gametutorials/tutorials/blob/master/C/Playing%20a%20WAV%20File/playsound.c> |
| <https://magazine.ribambel.com/article/regles-du-jeu-la-bataille-navale_a1999/1> |
| <https://www.geeksforgeeks.org/how-to-append-a-character-to-a-string-in-c/> |
| <https://www.w3resource.com/c-programming-exercises/file-handling/c-file-handling-exercise-4.php> |
| <https://www.holbertonschool.com/coding-resource-strcat-in-c> |
| <https://www.deepl.com/> |
| <http://source-code-share.blogspot.com/2014/07/implementation-of-java-stringsplit.html> |
| <https://stackoverflow.com/questions/50562192/process-finished-with-exit-code-1073740791-0xc0000409-pycharm-error> |
| <https://stackoverflow.com/questions/14386/fopen-deprecated-warning> |
| <https://www.geeksforgeeks.org/taking-string-input-space-c-3-different-methods/> |
| <https://stackoverflow.com/questions/12306591/read-no-more-than-size-of-string-with-scanf> |
| <https://stackoverflow.com/questions/1442116/how-to-get-the-date-and-time-values-in-a-c-program> |
| <https://codes-sources.commentcamarche.net/source/53687-lister-fichiers-et-repertoires-multiplateforme> |
| <https://gist.github.com/mtancoigne/a51fe0686d51c05c6cd6ec5f42c856fc> |
| <https://stackoverflow.com/questions/41421313/make-playsound-non-blocking> |
| <https://elifulkerson.com/projects/commandline-wav-player.php> |
| [Proposé par timothée rapin : https://codes-sources.commentcamarche.net/forum/affich-371867-plein-ecran-c-console donné](https://codes-sources.commentcamarche.net/forum/affich-371867-plein-ecran-c-console%20donné%20par%20timothée%20rapin) |

## Journal de travail (Sans les « remarques/problèmes »)

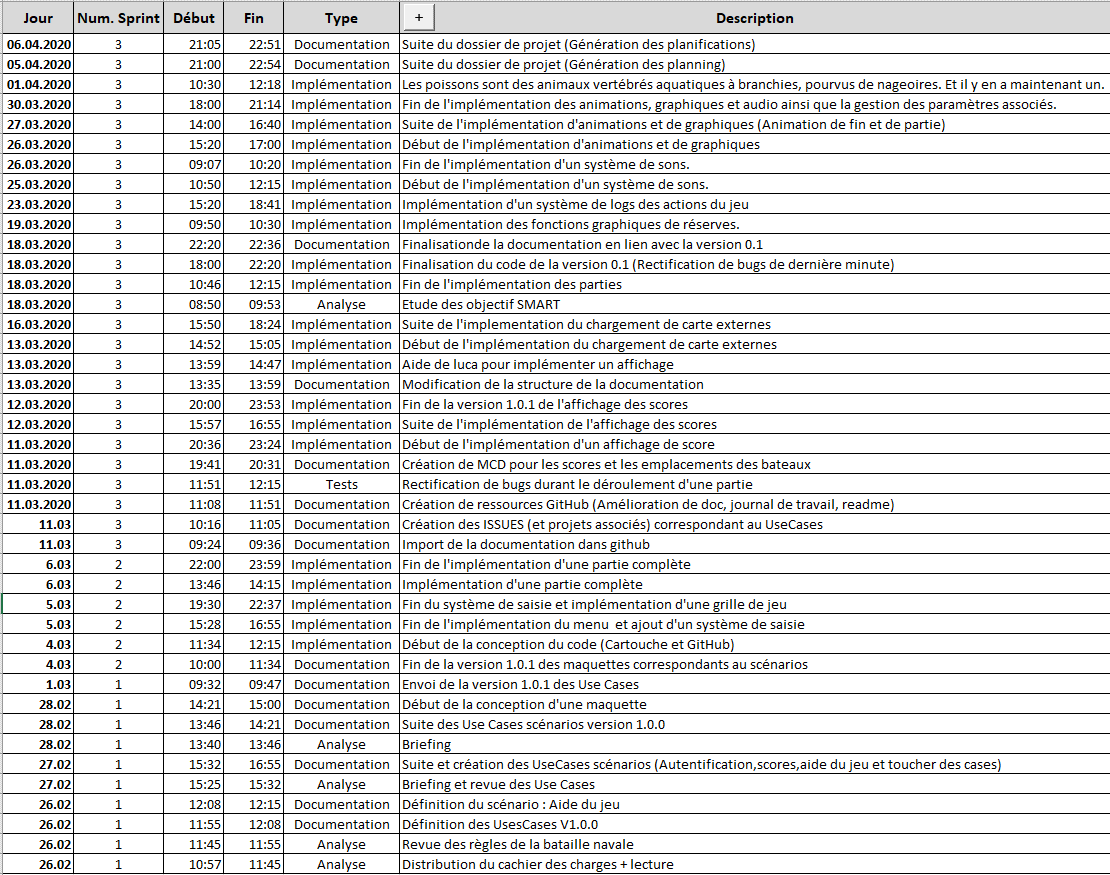


Tableau 2 - Journal de travail jusqu’au 06.04.2020