

Bataille Navale



DA SILVA FERNANDES DIOGO
Diogo.DA-SILVA-FERNANDES@cpnv.ch

TABLE DES MATIERES

1	Introduction	3
1.1	Cadre, description et motivation	3
1.2	Organisation	3
1.3	Objectifs	3
1.4	Planification initiale	3
2	Analyse	4
2.1	Use cases et scénarios	4
2.1.1	Use Case 1: S'authentifier	4
2.1.2	Use Case 2: Afficher les scores précédents	4
2.1.3	Use Case 3: Jouer	5
2.1.4	Use Case 4: Apprendre à jouer	5
2.2	Stratégie de test	6
3	Implémentation	6
3.1	Vue d'ensemble	6
3.2	Choix techniques	6
3.3	Points techniques spécifiques	7
3.3.1	Système de musique de fond/effet	7
3.3.2	Mise en pleine écran automatique	8
3.4	Livraisons	8
4	Tests	8
4.1	Tests effectués	8
4.2	Erreurs restantes	8
5	Conclusions	9
6	Annexes	9
6.1	Sources – Bibliographie	9
6.2	Journal de bord du projet	10
6.3	Journal de travail du projet	10

1 INTRODUCTION

1.1 Cadre, description et motivation

Dans ce projet, une Bataille Navale a été développée pendant 6 semaines avec plusieurs outils. Le langage de programmation utilisé est le C. Ce projet m'a permis de découvrir plus en profondeur Git/Git Hub ainsi qu'approfondir mes connaissances en C.

1.2 Organisation

L'organisation du développement de cette Bataille Navale est comme suit :

Développeur : DA SILVA FERNANDES Diogo, Diogo.DA-SILVA-FERNANDES@cpnv.ch, 078/791.62.32

Chef de Projet : FAVRE Raphaël, raphael.favre@cpnv.ch, 024/557.60.82

1.3 Objectifs

Pour cette Bataille Navale, plusieurs objectifs avaient été mis en place comme celui de pouvoir jouer une partie et afficher l'aide de jeu pour la version 0.1, finalisée le 11 mars 2021, et pour objectifs suivant, le projet avait pour but de contenir les logs, des scores ainsi qu'avoir plusieurs grilles, une d'elle choisie aléatoirement dès le début d'une partie. Tout ces objectifs ont été atteints et d'autres se sont ajouter. Le dernier objectif ajouter était celui d'ajouter une musique de fond ainsi que des effets, celui-ci aussi atteint.

1.4 Planification initiale

Chaque version (0.1 et 1.0) avait 3 semaines de développement prévues pour chacune. Le début du projet est daté au 15 février 2021 avec le début des maquettes, Use Cases, etc, la V0.1 a été rendue fonctionnelle le 11 mars et la V1 a été rendue fonctionnelle (uniquement avec les objectifs principaux) le 25 mars.

2 ANALYSE

2.1 Use cases et scénarios

2.1.1 Use Case 1: S'authentifier

Identifiant + Titre	BN001 - Authentification
En tant que	Joueur
Je veux	Authentifier
Pour	pouvoir jouer
Priorité	S

Action	Conditions	Réaction
J'ouvre l'application		Un menu de connexion s'affiche
Je suis déjà enregistré ducoup je sélectionne le menu (2)		Demande le pseudo
Je rentre mon pseudo	Pseudo existant	Le menu du jeu s'affiche
		Un message d'erreur s'affiche et redemande le pseudo

2.1.2 Use Case 2: Afficher les scores précédents

Identifiant + Titre	BN002 - Afficher les scores précédents
En tant que	Joueur
Je veux	Afficher mes scores précédents
Pour	Connaitre mes scores
Priorité	C

Action	Conditions	Réaction
Reprise de BN001		
Je souhaite afficher mes scores en sélectionnant sur le menu (3)	Le Joueur possède des scores	Affiche les scores par partie
		Retourne un message disant qu'il n'y a aucun score enregistré

2.1.3 Use Case 3: Jouer

Identifiant + Titre	BN003 - Jouer
En tant que	Joueur
Je veux	Jouer
Pour	Faire une partie
Priorité	C

Action	Conditions	Réaction
Reprise de BN001		
Je sélectionne le menu (1) correspondant à la création d'une nouvelle partie		La partie se lance

2.1.4 Use Case 4: Apprendre à jouer

Identifiant + Titre	BN004 - Apprendre à jouer
En tant que	Joueur
Je veux	Apprendre à jouer
Pour	Connaitre les règles
Priorité	C

Action	Conditions	Réaction
Reprise de BN001		
Je sélectionne le menu (02)		Le menu de choix s'affiche
Je choisis d'apprendre les règles dans le menu (02-01)		Le text s'affiche

2.2 Stratégie de test

La stratégie de test prévue pour cette Bataille Navale est la suivante :

- Un minimum de 2 personnes teste les fonctionnalités du jeu sur 2 ordinateurs différents, les 2 opérant sous Windows 10.
- Chacun des 2 doivent faire au minimum 2 parties et entrant de temps en temps de mauvaises valeurs.
- Pour donner suite aux parties, ils doivent vérifier leur score enregistré.
- Ceux-ci doivent faire un retour des possibles bugs trouver.

3 IMPLEMENTATION

3.1 Vue d'ensemble

Cette application interagit avec l'utilisateur à l'aide de plusieurs systèmes de choix, ainsi l'utilisateur peut faire ce qu'il souhaite dans ce qui lui est proposé.

3.2 Choix techniques

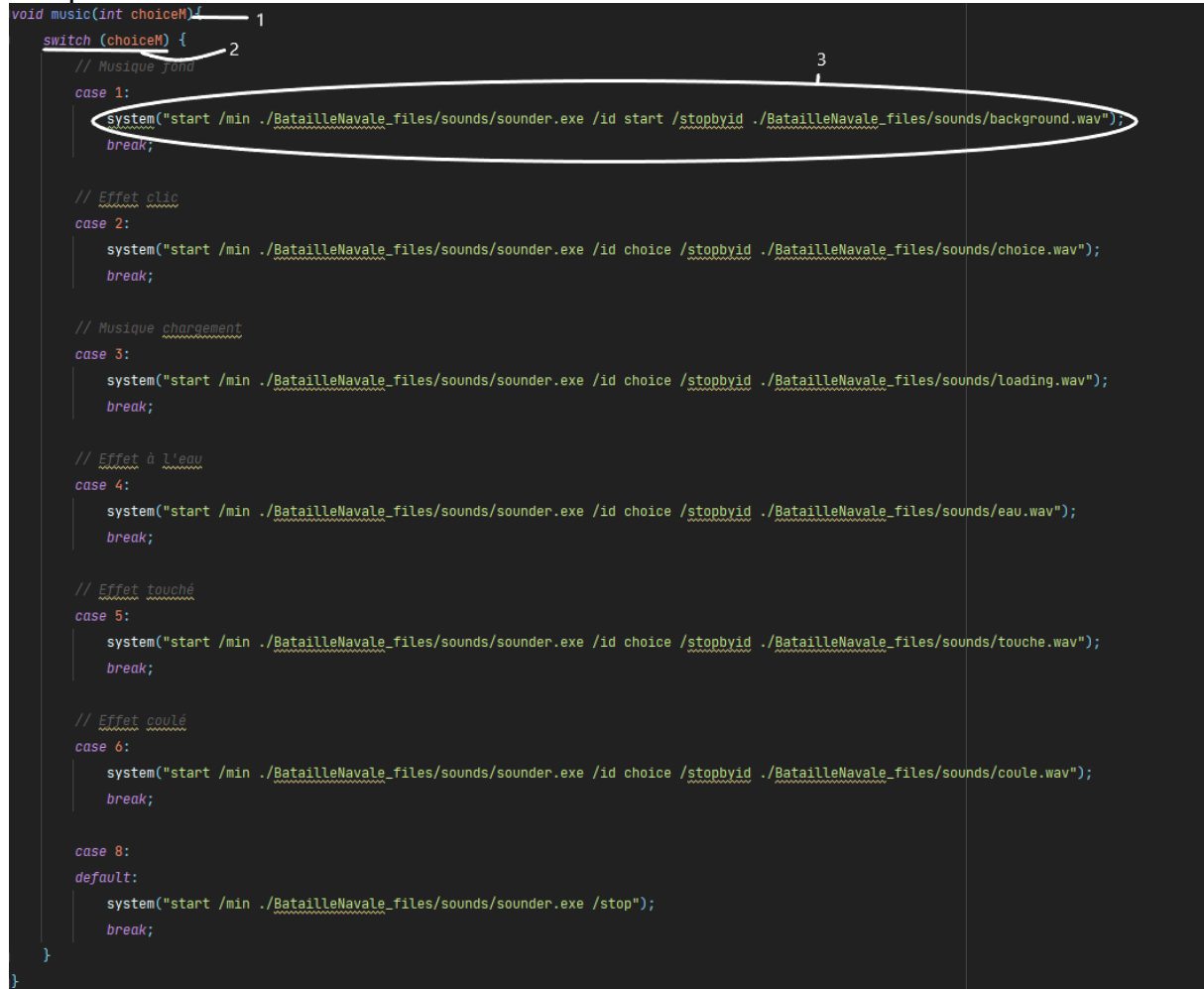
Pour la réalisation du mandat, j'ai utilisé :

- Matériel
 - Ordinateur de classe
 - Ordinateur personnel
- Système d'exploitation
 - Windows 10
- Logiciels tiers
 - CLion
 - Balsamiq
 - Draw.io
 - Git/Git Hub
 - Microsoft Office
 - Notepad++

3.3 Points techniques spécifiques

3.3.1 Système de musique de fond/effet

Dans ce projet, un système de musique et d'effet a été ajouté. Pour mieux comprendre voici un screen :



```
void music(int choiceM) {
    switch (choiceM) {
        // Musique fond
        case 1:
            system("start /min ./BatailleNavale_files/sounds/sounder.exe /id start /stopbyid ./BatailleNavale_files/sounds/background.wav");
            break;

            // Effet clic
        case 2:
            system("start /min ./BatailleNavale_files/sounds/sounder.exe /id choice /stopbyid ./BatailleNavale_files/sounds/choice.wav");
            break;

            // Musique chargement
        case 3:
            system("start /min ./BatailleNavale_files/sounds/sounder.exe /id choice /stopbyid ./BatailleNavale_files/sounds/loading.wav");
            break;

            // Effet à l'eau
        case 4:
            system("start /min ./BatailleNavale_files/sounds/sounder.exe /id choice /stopbyid ./BatailleNavale_files/sounds/eau.wav");
            break;

            // Effet touché
        case 5:
            system("start /min ./BatailleNavale_files/sounds/sounder.exe /id choice /stopbyid ./BatailleNavale_files/sounds/touche.wav");
            break;

            // Effet coulé
        case 6:
            system("start /min ./BatailleNavale_files/sounds/sounder.exe /id choice /stopbyid ./BatailleNavale_files/sounds/coule.wav");
            break;

        case 8:
        default:
            system("start /min ./BatailleNavale_files/sounds/sounder.exe /stop");
            break;
    }
}
```

The screenshot shows a C code snippet for a sound system. It features a function `music(int choiceM)` that uses a `switch` statement to play different sounds based on the `choiceM` parameter. Annotations 1, 2, and 3 point to specific parts of the code: 1 points to the `choiceM` parameter, 2 points to the `switch` statement, and 3 points to the `system` call in the first case.

1. `choiceM`, le paramètre de fonction, est là pour faciliter le choix du son que l'on souhaite utiliser dans le programme. Lorsque vous appelez la fonction dans le code, faites-le comme ceci -> `music(NUM_SON)`;
2. Le `switch` est utilisé pour faciliter l'exécution du code. Dépendant du numéro que vous mettez lorsque vous appelez la fonction, le `switch` va vérifier le numéro et va exécuter le son choisi.
3. Le `system()` sert pour le lancement du mini-programme permettant de faciliter l'exécution d'un son en arrière-plan. Le mini-programme est `sounder` by Eli Fulkerson. Pour le lancement de l'application, plusieurs paramètres sont utilisés, veuillez-vous aider de son site web pour comprendre les différents paramètres.

3.3.2 Mise en pleine écran automatique

```
void keyboardSimulate(){
    // Fonction qui permet de mettre le cmd en plein écran (proviens de https://batchloaf.wordpress.com/2012/04/17/simulating-a-keystroke-in-win32-c-or-c-using-sendinput/)
    // Input event
    INPUT ip;

    // Met en place un clavier virtuel
    ip.type = INPUT_KEYBOARD;
    ip.ki.wScan = 0;
    ip.ki.time = 0;
    ip.ki.dwExtraInfo = 0;

    // Simule la pression de la touche
    ip.ki.wVk = 0x7A;
    ip.ki.dwFlags = 0;
    SendInput(1, &ip, sizeof(INPUT));

    // Simule le relâchement de la touche
    ip.ki.dwFlags = KEYEVENTF_KEYUP;
    SendInput(1, &ip, sizeof(INPUT));
}
```

Cette fonction permet l'appuie virtuel de la touche F11 pour que le terminal se mette automatiquement en plein écran. L'exécution de cette fonction se fait en 3 étapes :

1. En premier, il met en place le clavier virtuel
2. Ensuite, il simule la pression de la touche
3. En dernier, il simule le relâchement de la touche.

3.4 Livraisons

Durant ce projet, 2 livraisons ont eu lieu. La première a été la livraison de la V0.1 prévue pour le 12 mars 2021. La deuxième livraison a été celle de la V1 prévue pour le 2 avril 2021.

4 TESTS

4.1 Tests effectués

	Ethann SCHNEIDER	Adam SIFATE
Partie 1 : Vérification Entrée Grille	OK	OK
Partie 2 : Vérification Erreurs d'entrée	OK	OK
Scores : Vérification Lecture/Écriture Score	OK	OK
Aide de jeu : Vérification orthographe et fonctionnalité	OK	OK

4.2 Erreurs restantes

À la suite des différents tests, aucun bug n'a été trouvé.

5 CONCLUSIONS

Pour conclusion, cette Bataille Navale est correctement terminée, aucun bug détecté à la suite des tests et tous les objectifs prévus ont été atteints ainsi qu'une fonctionnalité tierce non-prévue dès le début a été ajoutée avec l'accord du chef de projet. Cette Bataille Navale a apporté beaucoup de connaissances supplémentaires en C et à la recherche d'informations par soit même. Dans ce projet, les principales difficultés ont été la lecture et écriture de fichier. Une amélioration serait à prévoir pour rendre l'affichage du jeu plus accueillant ainsi que l'ajout d'une fonctionnalité supplémentaire qui est la création de grilles personnalisées.

6 ANNEXES

6.1 Sources – Bibliographie

CodeVsColor – Affichage heure/date pour les logs (<https://www.codevscolor.com/c-print-current-time-day-month-year>)

Programiz – Lecture/Ecriture dans un fichier – (<https://www.programiz.com/c-programming/c-file-input-output>)

Batchloaf – Appui virtuel F11 – (<https://batchloaf.wordpress.com/2012/04/17/simulating-a-keystroke-in-win32-c-or-c-using-sendinput/>)

Eli Fulkerson .com / Elliott JAQUIER (SI-C2b) – Système de sons en arrière-plan – (https://github.com/EliottJaquierCPNV/BatailleNavale/blob/master/Programme/sound_system.c) + (<https://www.elifulkerson.com/projects/commandline-wav-player.php>)

Ethann SCHNEIDER – Aide erreur entrée utilisateur

6.2 Journal de bord du projet

Date	Evenement
25.mars	Fin de la V1 + Release
12.mars	Début V1
11.mars	Fin de la V0.1
01.mars	Début de GitHub
22.févr	Début du pseudo code et rapide début du code
15.févr	Reçu du cahier des charges, prises de connaissance de celui-ci

6.3 Journal de travail du projet

Jour	Semaine	Temps [h]	Type	+	Description	Remarques/problèmes
01.avr	4	1,50	Tests		Tests des fonctionnalités du jeu par 3 personnes (Adam, Ethann, Diogo).	Aucun bug détecté.
30.mars	4	2,00	Implémentation		Debug musique et chargement + nouvelles fonctions commentées	
26.mars	4	1,00	Implémentation		Implémentation effets sonores + chargement	
25.mars	4	1,00	Implémentation		Corrections fautes orthographes et ajout de commentaires au code.	
24.mars	4	0,25	Implémentation		Débugage des parties	
24.mars	4	1,25	Implémentation		Enregistrement/affichage des scores	
23.mars	4	0,75	Implémentation		WIP (scores)	
23.mars	4	0,25	Implémentation		Possibilité de se connecter avec un pseudo	
17.mars	4	0,50	Documentation		Théorie SMART	
17.mars	4	0,25	Implémentation		Bug vérification bateaux coulés résolu	
16.mars	4	2,00	Implémentation		Tentative débogage vérification bateau coulé WIP (bug toujours présent -> non jouable).	
14.mars	4	1,25	Implémentation		Grille aléatoirement choisie	Bug: Le système de vérifications si le bateau est coulé ne fonctionne plus avec ce nouveau système
12.mars	4	0,50	Implémentation		Ajout des logs	
10.mars	4	0,25	Implémentation		Optimisation affichage grille + debug	
10.mars	4	0,50	Implémentation		Optimisation des vérifications	
08.mars	4	0,75	Implémentation		Code commenté au complet + résolution d'un bug: le tableau ne se réinitialisait pas si on relançait une partie	
08.mars	4	0,50	Documentation		Approche plus démonstration méthode agile dans la gestion de projet.	
05.mars	3	1,00	Implémentation		Résolution partielle du Bug des menu	Aide d'Ethann + Bug: lors d'une partie, si on met la ligne 10, ça fais crash le jeu.
04.mars	3	1,50	Implémentation		Mise à jour des plusieurs menus + affichage du cmd en plein écran dès le lancement + passage du code de la partie dans une fonction + ajout aide de jeu	Bug menu toujours présent.
03.mars	3	0,25	Documentation		Git et Github, synthèse premier pas	
03.mars	3	1,50	Implémentation		Débugage de l'action 2 du menu du jeu + ajout des vérifications (touché, raté, coulé, déjà joué) des choix du joueur en jeu + l'affichage de celle-ci sur le tableau de jeu + système de partie terminé	Bug: Dans n'importe quel menu, si on met une lettre, le programme boucle.
01.mars	3	0,25	Implémentation		Codage du menu + accès à l'action 2 et 3	Action 2 du menu du jeu, le message s'affiche en boucle.
22.févr	2	0,25	Implémentation		Codage de l'affichage du tableau	
22.févr	2	0,25	Préparation		Pseudo-Code: Affichage tableau	
19.févr	1	1,00	Préparation		Maquettes	
18.févr	1	0,25	Documentation		Théorie MCD	
18.févr	1	2,00	Préparation		UseCases, Scénarios, Maquettes	
17.févr	1	1,00	Préparation		UseCases, Scénarios	
15.févr	1	0,30	Analyse		Réception et analyse du cahier des charges	
TOTAL		24,05				