

Bataille navale



Giorgis, Esteban
Ch. De Fiez-Pittet 14
1422, Grandson
esteban.giorgis@cpnv.ch

cpnv
Centre professionnel du Nord vaudois
Filière d'Informatique

SI-C1b

06.02.2019

Table des matières

1	Analyse préliminaire	2
1.1	Introduction	2
1.2	Organisation	3
1.3	Objectifs	3
1.4	Planification initiale	3
2	Analyse	4
2.1	Cahier des charges détaillé	4
2.2	Définition de l'audience	4
2.3	Cas d'utilisation	4
2.4	Etude de faisabilité	4
2.5	Stratégie de test	4
2.6	Planification détaillée	4
3	Conception	5
3.1	Analyse de l'environnement	5
3.2	Détermination de l'arborescence du site et des rubriques	5
3.3	Maquette graphique	5
3.4	Conception du Code	7
4	Réalisation	8
4.1	Dossier de réalisation	8
4.2	Description des tests effectués	10
4.3	Erreurs restantes	11
5	Mise en service	11
5.1	Liste des documents fournis	11
6	Conclusions	11
7	Annexes	12
7.1	Sources – Bibliographie	12
7.2	Journal de bord de chaque participant	12
7.3	Archives du projet	12

1 Analyse préliminaire

1.1 Introduction

Ce projet est dans le cadre d'un module. Ce projet a pour but de nous familiariser non seulement avec du code C et l'organisation d'un projet. Le projet est un jeu de bataille navale.

Personnellement ce projet me motive beaucoup car il m'apprend à faire un long programme et en plus à l'organiser et pour sa mettre en place des stratégies.

1.2 Organisation

Elève : Giorgis, Esteban, esteban.giorgis@cpnv.ch et 079/501/19/20
 Responsable de projet (enseignant : BENZONANA, Pascal,
 pascal.benzonana@cpnv.ch et 024 / 55 + 76079)
 Expert 1 : BENZONANA, Pascal, pascal.benzonana@cpnv.ch 024 / 55 +
 76079
 Expert 2 : CARREL, Xavier, xavier.carrel@cpnv.ch 024 / 55 + 76080

Intervenants externes : (Mon Papa) : Nicolas, Giorgis, 079/467/39/93

1.3 Objectifs

Objectif général : De créer un programme qui a pour but de permettre à l'utilisateur de pouvoir jouer au jeu de la bataille navale et a accès à différant option dans le menu. Date d'échéance : 07.04.2019

Objectif spécifique : Au lancement du programme, l'utilisateur pourra taper tapera son nom prénom, qui seront enregistré dans un fichier text nomer score.txt, puis le programme affichera le menu.

L'utilisateur pourra jouer depuis le menu.

Il pourra lire les règles du jeu depuis le menu.

Il pourra quitter le programme depuis le menu.

Après une partie gagnée il annoncera en combien de coup il a gagné.

Après une partie gagnée ou perdue le menu s'affichera.

Après avoir sélectionné dans le menu l'option pour lire les règles du jeu les règles du jeu s'afficheront ainsi que le menu.

Après avoir sélectionné l'option Quitter le programme s'arrête.

Lien : https://github.com/EGiorgisCPNV/Bataille_navale/issues

1.4 Planification initiale

6 Open ✓ 3 Closed		Sort ▼
Sprint 5 Updated just now	Le but est de finaliser se programme pour qu'il soit livrable et utilisable	...
Sprint 1 Updated 2 days ago	Analyse et conception du programme Définir les objectifs Définir les uses cases Échéance: 29.03.2019	...
Sprint 4 Updated a day ago	Créer et insérer les actions du jeu Échéance : 29.03.2019	...
Sprint 3 Updated a day ago	Créer et afficher la grille et placer les bateaux Échéance : 22.03.2019	...
Sprint 2 Updated 4 days ago	Créer et afficher le menu et les interfaces Créer les différentes actions possible a faire Échéance : 22.03.2019	...
Planning général Updated 14 days ago	Sprint 1 : Analyse et conception Sprint 2 : Créer et afficher le menu et les interfaces Sprint 3 : créer et afficher la grille et placer les bateaux sprint 4 : Créer et insérer les actions du jeux sprint 5 : Tous mettre en relation	...

2 Analyse

2.1 Cahier des charges détaillé

Pouvoir jouer au jeu de la bataille navale avec une grille, et pouvoir faire certaine action comme tirer, entrer son nom prénom, choisir les options ou bien évidemment quitter.

2.2 Définition de l'audience

Des personnes de tout âge

2.3 Cas d'utilisation

Sur Excel

Chemin d'accès :

C:\Users\Esteban.GIORGIS\CLionProjects\BatailleNavalExa\Repository\Bataille_navale\Doc

2.4 Etude de faisabilité

Librairies : stdio.h, Tableau.h, windows.h

2.5 Stratégie de test

Sur Excel

Chemin d'accès :

C:\Users\Esteban.GIORGIS\CLionProjects\BatailleNavalExa\Repository\Bataille_navale\Doc\stratégie de test toute les versions

2.6 Planification détaillée

GitHub : lien

https://github.com/EGiorgisCPNV/Bataille_navale/projects

3 Conception

3.1 Analyse de l'environnement

Matériel HW : Système d'exploitation : Windows 10 (X64) 1709

- CPU : Intel Core i7 6700 Cadencé à 3.40 GHz
- Carte mère : Dell Inc. 0X2MKR A00
- Mémoire : 16 Go DDR4
- GPU : Intel(R) HD Graphics 530

Logiciels / IDE : CLion

3.2 Détermination de l'arborescence du site et des rubriques

Tout d'abord au lancement du jeu il vous sera demandé de rentrer votre nom puis prénom et il sera stocké. Ensuite le menu s'affichera et vous proposera de choisir entre jouer, lire les règles du jeu ou tout simplement quitter. Si vous choisissez jouer la grille s'affichera et le jeu commencera. Si vous choisissez d'afficher les règles du jeu les règles s'afficheront avec le menu du jeu in peu plus bas. Sinon si vous choisissez de quitter le programme se fermera.

3.3 Maquette graphique

```

-----
      Bataille navale
-----

Voici la grille
  1   2   3   4   5   6   7   8   9   10
1 | ~ | ~ | ~ | ~ | ~ | ~ | ~ | ~ | ~ |
2 | ~ | ~ | ~ | ~ | ~ | ~ | ~ | ~ | ~ |
3 | ~ | ~ | ~ | ~ | ~ | ~ | ~ | ~ | ~ |
4 | ~ | ~ | ~ | ~ | ~ | ~ | ~ | ~ | ~ |
5 | ~ | ~ | ~ | ~ | ~ | ~ | ~ | ~ | ~ |
6 | ~ | ~ | ~ | ~ | ~ | ~ | ~ | ~ | ~ |
7 | ~ | ~ | ~ | ~ | ~ | ~ | ~ | ~ | ~ |
8 | ~ | ~ | ~ | ~ | ~ | ~ | ~ | ~ | ~ |
9 | ~ | ~ | ~ | ~ | ~ | ~ | ~ | ~ | ~ |
10 | ~ | ~ | ~ | ~ | ~ | ~ | ~ | ~ | ~ |

La lettre X s'affichera a l'endroit ou vous avez touché et la lettre O s'affichera ou vous avez loupé

```

Programme au lancement

Modèle logique avec les valeurs indiquées :

touché

	1	2	3	4	5	6	7	8	9	10
1	X	~	~	~	~	~	~	~	~	~
2	~	~	~	~	~	~	~	~	~	~
3	~	~	~	~	~	~	~	~	~	~
4	~	~	~	~	~	~	~	~	~	~
5	~	~	~	~	~	~	~	~	~	~
6	~	~	~	~	~	~	~	~	~	~
7	~	~	~	~	~	~	~	~	~	~
8	~	~	~	~	~	~	~	~	~	~
9	~	~	~	~	~	~	~	~	~	~
10	~	~	~	~	~	~	~	~	~	~

Affichage d'X quand c'est touché

loupé

	1	2	3	4	5	6	7	8	9	10
1	X	~	~	~	~	~	~	~	~	~
2	~	~	~	~	~	O	~	~	~	~
3	~	~	~	~	~	~	~	~	~	~
4	~	~	~	~	~	~	~	~	~	~
5	~	~	~	~	~	~	~	~	~	~
6	~	~	~	~	~	~	~	~	~	~
7	~	~	~	~	~	~	~	~	~	~
8	~	~	~	~	~	~	~	~	~	~
9	~	~	~	~	~	~	~	~	~	~
10	~	~	~	~	~	~	~	~	~	~

Affichage d'O quand c'est loupé

3.4 Conception du Code

Point technique 1 : Détection touché

La technique que j'ai utilisé pour savoir quoi après qu'une partie d'un bateau est touché est un switch.

```
//condition si toute les parties du bateau sont touchées indique coulé
switch(tableau_analyser_par_ordi[horizontal-1][vertical-1]){
    case TORPILLEUR:
        //Condition qui repete la demande de la coordonné si il a déjà tiré dessus
        if(tableau_afficher[horizontal - 1][vertical - 1] == "X" || tableau_afficher[horizontal - 1][vertical - 1] == "O"){
            tire_restant++;
            printf("Vous avez déjà touché cette case choisissez en une autre\n\n");
            //boucle pour redemander a chaque fois les coordonnées tant qu'il ne sont pas dans les règles
            while(horizontal<1 || horizontal>10) {
                printf("\n\nChoisissez une position horizontal: ");
                scanf("%d", &horizontal);
            }
            while(vertical<1 || vertical>10) {
                printf("\n\nChoisissez une position vertical: ");
                scanf("%d", &vertical);
            }
        }
        else{
            point_vie_torpilleur++;
            compteur_victoire++;
            compteur++;
        }

        printf("\n\nTouché\n");
        printf("Coup restant : %d\n", tire_restant);
        //remplace la vagueslette par X
        tableau_afficher[horizontal - 1][vertical - 1] = "X";
        if(point_vie_torpilleur==POINT_VIE_TORPILLEUR){
            printf("vous avez coulé le torpilleur\n");
        }
        break;
}
```

Quand l'utilisateur rentre la coordonnée horizontale puis verticale qui sont égale à une partie d'un bateau comme l'image du switch en dessus il va commencer par analyser voir si il n'a pas déjà touché cette case si ce n'est pas bon il va lui redemander la position et va ajouter 1 a la variable des tirs restants, si c'est bon il va ajouter 1 aux variables : point de vie du bateau, compteur de victoire, compteur et afficher touché avec la lettre X sur la grille.

Pseudocode :

Si les coordonnées horizontale verticale rentrée sont égale a une partie d'un bateau alors

- Si la case est déjà touchée alors
 - Ajoute 1 a la variable des tirs restants
 - Affiche qu'il a déjà touché cette case
 - Redemande Les coordonnées
- Si la case n'est pas déjà touchée alors
 - Ajoute la lettre X dans la grille
 - Ajouter 1 a la variable des points de vie du bateau
 - Ajoute 1 au compteur de victoire
 - Ajoute 1 au compteur

Ensuite affiche touché
Affiche les tirs restantes

Point technique 2 : Détection coulé

1) Tout d'abord j'ai créé des constantes de points de vie

```
#define POINT_VI_TORPIEUR 2  
#define POINT_VI_SOUS_MARIN 3  
#define POINT_DE_VI_CONTRE_TORPILLEUR 3  
#define POINT_VI_CROISEUR 4  
#define POINT_VIE_PORTE_AVIONS 5
```

2) Puis des variables de points de vie

```
int point_vi_torpieur=0;  
int point_vi_sous_marin=0;  
int point_vi_contre_torpilleur=0;  
int point_vi_croiseur=0;  
int point_vi_porte_avion=0;
```

3) Voici comment il est calculé

```
if(point_vi_torpieur==POINT_VI_TORPIEUR){  
    printf("vous avez couler le torpilleur\n");  
}
```

Pseudocode :

Si les points de vie du bateau sont égaux à la constante des points de vies du bateau
Alors il affiche Qu'il a coulé le bateau en question

4 Réalisation

4.1 Dossier de réalisation

- Logiciel utilisés : CLion, la console windows

Matériel HW : Système d'exploitation : Windows 10 (X64) 1709

- CPU : Intel Core i7 6700 Cadencé à 3.40 GHz
- Carte mère : Dell Inc. 0X2MKR A00

- Mémoire : 16 Go DDR4
- GPU : Intel(R) HD Graphics 530

Arborésance :

-Bataille_navale

- .idea
- cmake-build-debug
- CMakeLists.txt
- main.c
- Menu.h
- README.md
- Tableau.h

-Documentation

- Le tout en PDF
- stratégie de test toute les versions
 - StrategieTestV1
 - StrategieTestV2
 - StrategieTestV3
 - StrategieTestV4
 - StrategieTestV5
 - StrategieTestV6
- cahier des charges Bataille navale V0.2
- Journal de travail
- Projet_bataille_navale_Esteban_giorgis CPNV v2017
- UseCasesScénarios

4.2 Description des tests effectués

Stratégie de test version 6

	M	B	V	D	E	F	G	H	I
	Scénario			Esteban Giorgis			Remarque/Problème		
1									
2	Use case 1								
3	Ouvre le programme			ok					
4	entre le nom puis le prénom			ok					
5	affiche mon nom et prénom et le menu			ok					
6									
7									
8									
9	Use case 2								
10	condition de touché			ok					
11	condition loupé			ok					
12	condition coulé			ok					
13	condition gagné			ok					
14	condition perdu			ok					
15	le programme réaffiche le menu après la victoire ou la défaite			ok			Remarque: il redemande en boucle les questions des position vertical et horizontale		
16	réaffiche à chaque fois les tire restant après un tire touché, loupé ou coulé			ok					
17	redemande une valeur horizontal si la valeur rentrée précédemment excède 10 ou inférieur à 1			ok					
18	redemande une valeur vertical si la valeur rentrée précédemment excède 10 ou inférieur à 1			ok					
19	Affiche la grille après avoir presser sur jouer			ok					
20	presse sur le bouton jouer			ok					
21	affiche un X quand c'est touché			ok					
22	affiche un O quand c'est loupé			ok					
23	affiche un H quand c'est coulé			ko					
24	redemande les coordonnées si il tire sur case déjà choisie			ok					
25	enregistre les scores dans un fichier externe			ok					
26									
27									
28									
29	Use case 3								
30	presser sur lire les règles			ok					
31	affichage des règles			ok					
32									
33									
34	Use case 4								
35	presse sur quitter			ok					
36	le programme ferme le fichier des scores			ok					
37	le programme s'arrete			ok					
38									
39									
40									
41									

Le reste des versions est dans les fichiers excel de stratégies de test

Lien Git Hub :

https://github.com/EGiorgisCPNV/Bataille_navale/tree/master/Doc/strat%C3%A9gie%20de%20test%20toute%20les%20versions

4.3 Erreurs restantes

Si l'utilisateur rentre des lettres le programme plante

Si l'utilisateur écrit son nom avec trois espaces le programme s'arrête

Le programme ne met pas la lettre H quand il a coulé un bateau

5 Mise en service

5.1 Liste des documents fournis

Rendu : Doc :

- Stratégie de test
- Journal de travail
- Journal de bord
- Projet
- Cahier des charges

Code

6 Conclusions

- 1) Pour moi c'est un objectif à la fois juste pas vraiment atteint car je sais que j'aurai pu faire mieux et à la fois atteint car je suis quand même très fière d'avoir cossu un programme a 100% avec mes petites mains qui marche et où l'on peut s'amuser.
- 2) + : j'ai découvert un grand amusement à programmer à chaque améliorer peu a peu son programme pour qu'à la fin cela donne quelque chose d'unique à moi-même
- : mon programme aurait pu être avec un peu plus de contenu et plus de fonctionnalités.
- 3) Difficulté à s'organiser et à organiser mon projet à structurer par où commencer dans quelle direction continuer.
- 4) Pourquoi pas l'améliorer et le rendre public dans le futur quand j'aurai bien plus d'expériences.

7 Annexes

7.1 Sources – Bibliographie

- Mon papa
- M.Benzonana
- M.Borcard
- M.Hausmann

7.2 Journal de bord de chaque participant

Date	Durée	Activité	Remarques
06.02.19	15min	Présentation des fonctions	Compliquer à comprendre le return
13.03.19	1h30	Explication GitHub	
20.03.19	20 min	Explication de journaux	
22.03.2019	15 min	Explication des stratégies de tests	
27.03.2019	15 min	Explication plus approfondie de la documentation en générale	
28.03.2019	15min	Explication pour inscrire les scores dans un fichiers externe	
03.04.2019	1h	Explication GitHub	Problème de compréhension
05.04.2019	30 min	Révision du module 431	

7.3 Archives du projet

Lien GitHub : https://github.com/EGiorgisCPNV/Bataille_navale