

# *Bataille Navale*



*Eaaswaralingam, Kaarththigan*  
*Kaarththigan.EAASWARALINGAM@cpnv.ch*

## Table des matières

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
2.1.1	(Use case 1) .....	5
2.1.2	(Use case 2) .....	5
2.1.3	(Use case 3) .....	5
2.2	Stratégie de test.....	5
2.3	Budget .....	6
3	Implémentation .....	6
3.1	Vue d'ensemble .....	6
3.2	Choix techniques .....	6
3.3	Modèle Logique de données .....	6
3.4	Points techniques spécifiques .....	8
3.4.1	Point 1 : Choisir un menu .....	8
3.4.2	Point 1 : Saisir un numéro valable sinon le programme va demander de choisir une case en boucle. ....	8
3.5	Livraisons .....	8
4	Tests.....	9
4.1	Tests effectués .....	9
4.2	Erreurs restantes .....	9
5	Conclusions .....	9
6	Annexes.....	10
6.1	Sources – Bibliographie.....	10
6.2	Journal de bord du projet.....	10

## **1 Introduction**

### **1.1 Cadre, description et motivation**

**Motivation** : Cela va me préparer à créer des jeux beaucoup plus gros et beaucoup plus ambitieux par la suite et aussi faire des recherches par moi-même sur internet sur diverses fonctionnalités du « C ».

**Description** : Reproduire le jeu « Bataille Navale » en programmation « C » et insérer toutes les fonctionnalités du jeu.

### **1.2 Organisation**

Élève : Eaaswaralingam Kaarththigan

Responsable du projet : Pascal Benzonana

### **1.3 Objectifs**

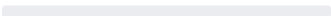




- Le but de ce projet est de réaliser un jeu ressemblant à la bataille navale qui devra avoir TOUTES les fonctionnalités puis devoir le finir le 05.04.2019.
- Commencer par les uses cases scénarios, ensuite github, et enfin la maquette. Et quand tout ceux-là sera finit on commence par programmer le menu du jeu, ensuite la grille sur laquelle vont devoir jouer les joueurs. Placer des bateaux sur une griller fixe. Créer une fonction qui va pouvoir, d'abord, regarder la grille (voir à quoi elle ressemble) ensuite créer la fonction pour pouvoir jouer. Et ensuite, pour finir, on créer les fonctions : règle, options, quitter, score. Et finir ceux-là le 05.04.2019.
- Pouvoir finir la documentation avec les uses case, les scénarios et github le lundi 01.04.2019.

### **1.4 Planification initiale**

Date de début : 11.03.2019

Date de fin : 07.04.2019

But : Finir la documentation ainsi que le programme

<b>Sprint 5</b> 	Créer tous les autres fonctions du jeu (Aide, option, quitter ...)	...
Updated 11 days ago Échéance : 04.04.2019		
<b>Sprint 4</b> 	Créer et insérer les fonctions du mode de jeu "jouer".	...
Updated 11 days ago Échéance : 29.03.2019		
<b>Sprint 3</b> 	Créer et afficher la grille et placer les bateaux	...
Updated 11 days ago Échéance : 22.03.2019		
<b>Sprint 2</b> 	Créer et afficher le menu et les interfaces	...
Updated 11 days ago Échéance : 15.03.2019		
<b>Sprint 1</b> 	Analyse et conception	...
Updated 11 days ago Échéance : 08.03.2019		
<b>Planning général</b> Updated 11 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	...
	Échéance : 07.03.2019	

## 2 Analyse

Maquette :

Choisissez un numero de grille ci-dessous :

```

- 1
- 2
- 3
1
1 2 3 4 5 6 7 8
| - | - | - | - | - | - |
| - | - | - | - | - | - |
| - | - | - | - | - | - |
| - | - | - | - | - | - |
| - | - | - | - | - | - |
| - | - | - | - | - | - |
| - | - | - | - | - | - |
| - | - | - | - | - | - |

```

Veillez choisir une ligne :

1

Veillez choisir une colonne :

## 2.1.1 (Use case 1)

Identifiant	Jouer au bataille navale				
En tant que	Joueur				
Je veux	Jouer				
Pour	M'amuser				
Priorité	M				
Action	Condition	Réaction			
lancer le programme		Le programme se charge			
Choisir l'option jouer		Une partie se charge			
Placer bateau	Les case sont vides	Des cases de la grille sont réservées			
Choisir case puis tirer	La case contient un bout de bateau	Supprime la case			
	La case contient le dernier bout d'un bateau	La case se supprime et il restera moins de case entier à éliminer			
	La case est la dernière case contenant un bout de bateau qui reste	La partie se finit et l'utilisateur gagne			
	La case choisit ne contient rien du tout	Le programme ne fait rien			
Attendre le tour de l'adversaire					
L'adversaire choisir case puis tirer	La case choisit contient un bout de bateau	La case est supprimé et l'utilisateur perd un bout de bateau			
	La case contient le dernier bout d'un bateau	La case est supprimé et l'utilisateur perd un bateau			
	La case est la dernière case contenant un bout de bateau qui reste	La partie se finit et l'utilisateur perd			
	La case choisit ne contient rien du tout	Le programme ne fait rien			

## 2.1.2 (Use case 2)

Identifiant	Choisir le menu				
En tant que	Joueur				
Je veux	Choisir un option du jeu				
Pour	Utiliser des fonctionnalité du programme				
Priorité	M				
Action	Condition		Réaction		
Lancer le programme			Le programme se charge		
Choisir l'option Jouer			Les règles s'affichent		
Choisir l'option Règles	Les case sont vides		Des cases de la grille sont réservées		
Choisir l'option Quitter	La case contient un bout de bateau		Le programme se ferme		
Choisir l'option Aide	La case contient le dernier bout d'un bateau		Une page contenant de l'aide s'affiche		

## 2.1.3 (Use case 3)

Identifiant	Jouer au bataille navale				
En tant que	Joueur				
Je veux	Jouer				
Pour	M'amuser				
Priorité	M				
Action	Condition	Réaction			
Lancer le programme		Le programme se charge			
Choisir l'option jouer		Une partie se charge			
Placer bateau	Les case sont vides	Des cases de la grille sont réservées			
Choisir case puis tirer	La case contient un bout de bateau	Supprime la case			
	La case contient le dernier bout d'un bateau	La case se supprime et il restera moins de case entier à éliminer			
	La case est la dernière case contenant un bout de bateau qui reste	La partie se finit et l'utilisateur gagne			
	La case choisit ne contient rien du tout	Le programme ne fait rien			
Attendre le tour de l'adversaire					
L'adversaire choisir case puis tirer	La case choisit contient un bout de bateau	La case est supprimé et l'utilisateur perd un bout de bateau			
	La case contient le dernier bout d'un bateau	La case est supprimé et l'utilisateur perd un bateau			
	La case est la dernière case contenant un bout de bateau qui reste	La partie se finit et l'utilisateur perd			
	La case choisit ne contient rien du tout	Le programme ne fait rien			

## 2.2 Stratégie de test

- Tester le lancement du programme.
- Tester les différents caractères sur notre clavier quand on peut insérer quelque chose.
- Vérifier les différentes fonctions.

## 2.3 Budget

- On a un temps de huit semaines pour pouvoir finir le projet.
- Le programme doit au moins avoir les fonctions jouer ainsi que le score fonctionnel.
- On dispose de Clion ou CodeBlocks pour pouvoir effectuer ce projet.

## 3 Implémentation

### 3.1 Vue d'ensemble

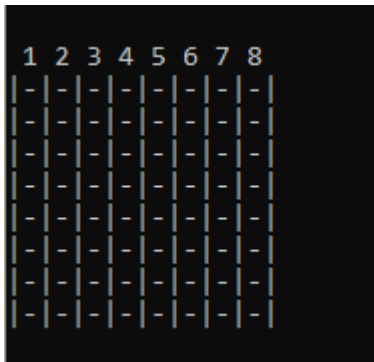
### 3.2 Choix techniques

Ce projet est réalisé sur :

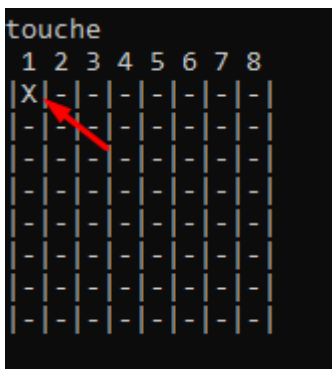
- Les programmes : Clion
- Système d'exploitation : Windows 10
- Programmation en C de base

### 3.3 Modèle Logique de données

Le programme fonctionne par rapport au niveau. C'est-à-dire que le joueur choisit la difficulté de jeu et cette difficulté correspond à la taille de la map ainsi que le nombre de bateaux.



Voici à quoi ressemble la grille de base.



Voici à quoi elle ressemble quand elle est touchée.

```
loupe
 1 2 3 4 5 6 7 8
|X|O|-|-|-|-|-|
|-|-|-|-|-|-|-|
|-|-|-|-|-|-|-|
|-|-|-|-|-|-|-|
|-|-|-|-|-|-|-|
|-|-|-|-|-|-|-|
|-|-|-|-|-|-|-|
|-|-|-|-|-|-|-|

Veuillez choisir une ligne :
```

Voici à quoi elle ressemble quand elle est loupée.

```
Choisissez un numero valable !!!

loupe
 1 2 3 4 5 6 7 8
|X|O|-|-|-|-|-|
|-|-|-|-|-|-|-|
|-|-|-|-|-|-|-|
|-|-|-|-|-|-|-|
|-|-|-|-|-|-|-|
|-|-|-|-|-|-|-|
|-|-|-|-|-|-|-|
|-|-|-|-|-|-|-|

Veuillez choisir une ligne :
```

Quand elle n'est pas valable

```
touche coule
 1 2 3 4 5 6 7 8
|X|O|-|-|-|-|-|
|X|-|-|-|-|-|-|
|X|-|-|-|-|-|-|
|-|-|-|-|-|-|-|
|-|-|-|-|-|-|-|
|-|-|-|-|-|-|-|
|-|-|-|-|-|-|-|
|-|-|-|-|-|-|-|

Veuillez choisir une ligne :
```

Quand le bateau est coulé.

## **3.4 Points techniques spécifiques**

**3.4.1** Point 1 : Choisir un menu  
Point 2 : Choisir une grille

**3.4.2** Point 1 : Saisir un numéro valable sinon le programme va demander de choisir une case en boucle.

Le programme va vérifier si la case a déjà été touché ou pas.

Point 2 : Saisir un numéro entre 1 et 3 (sinon la question sera reposée) qui définira une grille sur laquelle le joueur devra jouer.



## **3.5 Livraisons**

03.04.2019 : Première envoie du dossier « Bataille Navale » complet.

03.04.2019 : Mise à jour du dossier avec toutes les modifications faites.

04.04.2019 : Mise à jour du dossier avec le score.

<https://github.com/EaaswaralingamKarththigan/Bataille-Navale>



## 4 Tests

### 4.1 Tests effectués

Scénario	Test	Remarques
1.1) Lancer le programme	OK	
1.2) Choisir l'option "jouer"	OK	
1.3) Choisir l'option "option"	KO	Il n'est pas encore implémenté dans le programme
1.4) Choisir l'option "règles"	OK	
1.5) Entrer une lettre	KO	Il n'est pas encore implémenté dans le programme
1.2.1) Choisir l'option "Commencer"	OK	
1.2.2) Choisir l'option "Grille"	OK	
1.2.3) Choisir l'option "retour"	OK	
1.2.1.1) Choisir une case	OK	
1.2.1.2) Empêcher de choisir une même case	OK	
1.2.1.3) Le programme peut lire un fichier txt	OK	
1.2.1.4) Faire afficher la grille	OK	
1.2.2.1) Faire afficher la grille	OK	
1.3.1) Afficher les options	KO	Il n'est pas encore implémenté dans le programme
1.4.1) Quitter le programme	OK	

### 4.2 Erreurs restantes

Problème au niveau de l'insertion de la case de la grille.

- Si l'on met une lettre de l'alphabet, le programme crash.

## 5 Conclusions

- Les objectifs atteints :
  - Pouvoir jouer à la bataille navale.
  - Avoir nos propres règles.
  - Pouvoir avoir un score à la fin.
- Les objectifs non-atteints :
  - IA non programmée.
  - Il n'y a pas d'interface personnalisée.
- Points positifs :
  - Le programme est fonctionnel et nous pouvons aisément y jouer dessus.
- Points Négatifs :
  - L'interface est simplette, il n'y a pas beaucoup de personnalisation.
- Difficultés :
  - Programmer le score
  - Créer l'option « coulé »

## 6 Annexes

### 6.1 Sources – Bibliographie

Aides externes :

- Jessy Borcard
- Adam Gruber
- Esteban Giorgis
- Pascal Benzonana
- Dylan Berney

Sites Webs :

- OpenClassroom

### 6.2 Journal de bord du projet

Journal de travail

Jour	Semaine	Temps [h]	Type	Description
15.mars	1911	0.1 h	Analyse	Placement des issues dans les sprints
15.mars	1911	0.016h	Analyse	Création des issues
15.mars	1911	0.016h	Analyse	Création des sprints
08.mars	1910	0.1 h	Analyse	Création du projet sur github
20.mars	1912	1h	Implémentation	Fonction Jouer
20.mars	1912	0.1 h	Implémentation	Afficher la grille
20.mars	1912	0.016h	Implémentation	Création de la grille
20.mars	1912	0.1 h	Implémentation	Afficher le menu
20.mars	1912	0.016h	Implémentation	Création du menu
20.mars	1912	0.1 h	Documentation	Création des objectifs
20.mars	1912	0.1 h	Implémentation	Création des uses cases

Temps [h]	Type	+	Description	Remarques/problèmes
0,16	Documentation		Finalisation du document word	
1,00	Implémentation		Création de 3 grilles	3 grilles qui se retrouvent dans un autres fichiers qui vont être appelés par des fonctions
1,00	Implémentation		Test du score	Le score qui sera affiché dans un autre fichier txt nommé "Score.txt"
0,33	Implémentation		Création du menu + grille	
0,16	Documentation		Création des sprints	
0,16	Documentation		Création des issues	
0,16	Documentation		Création du repository	
0,16	Documentation		Création des uses cases scénarios	