

# Rapport de projet

Raphael COUTINHO, Florian BRIAND et Erwan BOUVART

8 Decembre 2019

# Table des matières

I	Instruction de compilation . . . . .	3
II	Introduction . . . . .	3
III	Développement . . . . .	3
	III- 1 Les règles . . . . .	3
	III- 2 Le placement des bateaux . . . . .	4
	III- 3 Le fonctionnement du jeux . . . . .	4
	III- 4 L'intelligence artificielle . . . . .	5
IV	Conclusion . . . . .	6

# I Instruction de compilation

Pour la réalisation de ce projet, nous avons utilisé la console de Linux pour compiler notre projet. Pour une bonne compilation, notre fichier principale est intitulé "Projet-S1" et celui-ci nécessite d'autres fichiers pour compiler, leur liste est détailler dans le "README.txt".

## II Introduction

Lors de ce premier semestre de seconde année à l'EISTI, nous avons eu pour projet de créer en deux semaines, un programme en langage C qui serait capable de faire fonctionner une bataille naval. La problématique nous a conduit à rechercher par quels moyens pourrions-nous arriver à créer ce programme ? Nous allons vous expliciter les différentes étapes, qui nous ont permis de créer de A à Z notre bataille naval. Tout d'abord, nous vous expliquerons les règles du jeu. Ensuite, nous verrons comment nous avons réussi à placer les bateaux sur la grille. Suite à cela, nous vous parlerons de comment nous faisons fonctionner le jeux. Et pour finir, nous vous dirons comment marche l'intelligence artificielle.

## III Développement

### III- 1 Les règles

Notre projet permet de choisir :

- De jouer contre soit une intelligence artificiel ou soit un autre joueur.
- La taille de la grille sur laquelle on joue, qui selon sa taille définit le nombre de bateaux maximal que l'on pouvait placer.
- Le nombre de bateaux que vous vouliez placer et qui ne pouvait pas dépasser le nombre maximal fixé plus tôt.
- Le placement de vos bateaux sur la grille sachant qu'ils ne pouvaient ni se chevaucher ni être côte à côte.
- Sauvegarder sa partie en cours pour pouvoir la relancer plus tard ou bien en recommencer une autre.

Une fois le placement des bateaux fini, le jeu peut commencer. Le premier joueur choisi une case de la grille du joueur adverse où il veut tirer. Suite à cela, trois scénario peuvent se dérouler.

Dans le premier cas, le joueur à tiré dans l'eau, c'est à dire qu'il n'a touché aucun bateaux adverses, alors son tour se termine et c'est à l'autre joueur de jouer.

Dans le deuxième cas, le joueur à touché un bateaux adverse, dans ce cas là c'est encore à lui de jouer.

Dans le troisième et dernier cas, le joueur a coulé un bateau adverse c'est à dire que le bateau adverse a été touché sur toute les cases de la grille où il était placé, le tour se termine alors pour le joueur et c'est autour du joueur adverse. Le jeu se termine une fois que l'un des deux joueurs a coulé tous les bateaux de l'autre joueur.

### III- 2 Le placement des bateaux

Le placement des bateaux est géré à l'aide de tableaux composés de 0 et de 1 (0 pour l'eau et 1 pour les cases contenant les bateaux). Ceux-ci nous permettent d'effectuer différentes vérifications pour placer ou non les bateaux.

Lorsque l'on veut placer un bateau, il faut avant tout vérifier si il peut être placé à cette endroit.

Pour commencer, on demande l'orientation (vertical ou horizontal) du bateau puis les coordonnées de la première case du bateau. Lorsque l'on veut placer un bateau, il faut avant tout vérifier s'il peut être placé à cette endroit. Une fois cette étape faite, il faut vérifier si le bateau peut-être placé à cet endroit, il faut procéder à trois vérifications.

Tant que les trois vérifications ne renvoient pas 1 le programme redemandera les coordonnées et l'orientation du bateau.

La première vérification est celle qui permet de savoir si le bateau déborde ou non de la grille. Cette vérification une fois faite renvoie 1 si le bateau ne déborde pas de la grille.

La seconde vérification regarde s'il n'y a pas de bateaux qui se chevauchent, pour cela nous avons créé un autre tableau où nous plaçons un bateau qui est représenté par des 1 est l'eau par des zéros. Si vous voulez placer un autre de bateau de façon à ce qu'il chevauche le précédent alors les sommes des cases du tableau avant qu'on place le tableau et celle après avoir placé le tableau s'additionne, s'il n'y a que des 1 et des zéros c'est que les bateaux sont bien placés, s'il y a un 2 c'est que deux bateaux se chevauchent et donc la vérification ne renvoie pas 1.

La dernière vérification s'assure qu'il y a bien un couloir d'eau (un couloir de zéro) minimum entre chaque bateau. Pour cela, le programme regarde dans quelle orientation le bateau est tourné, prenons l'exemple d'un bateau verticale, s'il y a un 1 à droite ou à gauche du bateau alors que le bateau est vertical alors le programme comprends que deux bateaux sont collés, ou si le bateau dépasse sa taille, le programme comprend une fois de plus que de bateaux sont collés. Si deux bateaux se collent alors la vérification ne renvoie pas 1. Si ces trois vérifications renvoient 1 alors le bateau se place.

Une fois ces vérifications passées, les tableaux des bateaux sont convertis en fichiers txt qui nous permettent de sauvegarder la partie en cours.

### III- 3 Le fonctionnement du jeu

Pour faire fonctionner notre programme, une fois les grilles des deux joueurs initialisées et les bateaux placés, on utilise une fonction prénommée "tire" qui prends en compte la grille d'attaque d'un joueur qui est vide initialement et la grille d'affichage de l'autre joueur où sont affichés ses bateaux.

Dans cette fonction, nous demandons au joueur qui joue, les coordonnées auxquelles il veut attaquer.

Si la case où il attaque est remplie avec un zéro, la fonction lui renvoie qu'il a tiré dans l'eau, le zéro est remplacé par un trois sur les deux grilles aux mêmes coordonnées, et c'est la fin de son tour.

Si la case où il attaque est remplie par un 1, cela veut dire que le joueur a touché un bateau, puis le 1 est alors remplacé par un 2 sur les deux grilles aux mêmes coordonnées. Dans ce même cas, si la case est entourée par un autre 1, cela signifie que le bateau n'est pas encore coulé. La fonction renvoie alors "touché" et c'est encore au même joueur de jouer. S'il n'y a plus de 1 autour de la case, le bateau est donc coulé. Ainsi la fonction renvoie alors "touché coulé", c'est donc la fin de son tour.

La fonction se répète mais cette fois-ci la fonction prend en compte l'autre grille d'attaque et l'autre grille d'affichage. Le jeu se termine lorsque l'un des deux joueurs n'a plus de 1 sur son tableau d'affichage.

Nous avons aussi mis au point une sauvegarde, à chaque fin de tour nous demandons si le joueur veut sauvegarder, s'il dit oui, le programme s'arrête et tous les tableaux sont enregistrés dans un fichier texte. Puis quand le programme est relancé, les fichiers textes sont lus et la partie reprend où elle s'était arrêtée.

### III- 4 L'intelligence artificielle

L'intelligence artificielle de notre jeu couvre les deux phases d'une partie de Bataille Naval : le positionnement des bateaux sur la grille défensive du joueur artificiel et la stratégie de tirs sur les bateaux adverses.

Le principe général de positionnement des bateaux est de choisir, pour chaque bateau, une position et orientation aléatoire parmi les emplacements libres disponibles. Bien que peu probable, l'utilisation de choix aléatoires de positions peut entraîner des impossibilités dans le positionnement des bateaux. Pour réduire encore plus ce risque deux mécanismes ont été utilisés pour le codage :

- Le positionnement se fait par ordre décroissant de taille des bateaux
- En cas de blocage un nouveau positionnement des bateaux peut être relancé jusqu'à un nombre de fois prédéfini.

Dans cet algorithme une éventuelle situation de blocage peut être détectée pendant la construction de la liste des positions possibles pour le bateau en cours. Si cette liste est vide un repositionnement de tous les bateaux doit être relancé.

Le comportement de base de l'IA est le suivant :

- Tant qu'il n'a pas touché un bateau adverse, l'IA tire de manière aléatoire sur les emplacements non encore visés
- Lorsqu'un bateau adverse est touché pour la première fois l'IA se focalise sur cette zone jusqu'à couler ce bateau
- Quand un bateau adverse est coulé, l'IA se remet à la recherche du bateau suivant par des frappes aléatoires
- Et ainsi de suite jusqu'au dernier bateau

L'IA fait également des déductions qui servent à remplir la grille d'attaque avec des cases où il est inutile de tirer.

Exemple 1 :

Lorsqu'un bateau adverse a été touché en deux points consécutifs on peut en déduire que les quatres cases « latérales » ne peuvent pas contenir un bateau (règle de la bande d'eau entre deux bateaux). Deux de ces cases peuvent bien sûr être le bord de la grille.

Exemple 2 :

Quand l'IA suit une direction dans son attaque à un bateau et que son tire tombe dans l'eau, on peut en déduire que trois cases « finales » ne peuvent pas contenir un bateau (règle de la bande d'eau entre deux bateaux). Ces cases peuvent bien sûr être le bord de la grille.

L'IA utilise également les bords de la grille pour décider dans quelle direction elle va continuer à attaquer le bateau touché. Si un bateau est touché en bord de grille, l'IA va continuer à attaquer dans le sens opposé au bord touché.

Pour fonctionner l'IA a donc une mémoire. Un peu comme un joueur humain qui se rappelle quel a été son dernier coup joué et dans quelle direction il est en train d'attaquer le bateau en cours. Cette gestion de la « mémoire » reste très simple puisqu'elle se limite à quatres variables :

- Un équivalent de booléen qui indique si un bateau est en cours d'attaque ou pas
- Un entier qui contient la direction de l'attaque en cours
- Le premier point touché sur le bateau en cours
- Le dernier point touché sur le bateau en cours

## IV Conclusion

Au final, nous avons réussi à créer un programme fonctionnel capable de faire jouer l'utilisateur, à la fois contre d'autres utilisateurs, et des robots. Grâce à une bonne répartition et structure du travail, nous avons pu surmonter les différents problèmes et réussir à réaliser le sujet.

Nous avons divisé le travail en trois parties, un premier élève s'est occupé du placement des bateaux, un second s'est chargé de la partie jeux et de la fonction "tire" et le dernier élève s'est occupé de créer l'intelligence artificielle.

Nous avons réussi à développer une solution à notre projet. Elle est perfectible sur certains points. En effet, notre interface du jeu s'est limitée à l'interface Unix, rendant notre jeu moins attractif.

La réalisation de ce projet nous a apporté deux points principaux. D'une part, il nous a permis de développer notre logique algorithmique. D'autre part, il a pu mettre en pratique certains aspects théoriques vus en cours.