



Rendu le:

8/12/19

Rapport du projet “BATAILLE NAVALE” en langage **C**



Auteurs:

SAID SALIMO Mlamali - CISSÉ Mathieu - KABA Laurène
[CPI-2-01]
[8/12/2019]

EISTI - 2019-2020

TABLE DES MATIÈRES



p. 2 I - BATAILLE NAVALE.....

- A) Le jeu.....
- B) Les règles du jeu.....

p. 2-4..... II - DÉVELOPPEMENT DU PROJET.....

- A) Le problème.....
- B) Description de notre
programme.....

p. 4..... III - DIFFICULTÉS RENCONTRÉES.....

- A) Fonctions/Procédure.....
- B) Compilation.....

p. 4 IV - CONTRIBUTION ÉLÈVES.....

- A) Répartition du
travail.....

p.5 V - BILAN.....

- A) Bilan général.....
- B) Bilan personnel de chaque élèves.....

I - BATAILLE NAVALE:

A) Le jeu:

La Bataille navale (« Touché, Coulé ! » ou encore « Battleship » en anglais) est un jeu de société dans lequel deux joueurs doivent placer des navires sur une grille tenue secrète et tenter de couler les navires adverses.

B) Les règles du jeu:

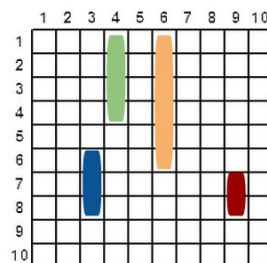
Le gagnant est celui qui arrive à torpiller complètement les navires de l'adversaire avant que tous les siens ne le soient et pour faire couler un bateau, il faut avoir torpillé les coordonnées de toutes les cases que composent le bateau.

Chaque joueur possède les mêmes navires, dont le nombre et le type dépendent des règles du jeu choisies. La taille de la grille peut varier selon le jeu aussi.

La configuration la plus utilisée (standard) est la suivante:

- Grille taille 10x10, numérotée de 1 à 10 horizontalement et de A à J verticalement.
- 2 joueurs J1 et J2 jouant tour à tour,
- Navires de chaque joueurs :
 - 1 **porte-avion** de 6 cases,
 - 2 **croiseurs** de 4 cases,
 - 3 **sous-marins** de 3 cases,
 - 4 **torpilleurs** de 2 cases.

exemple de grille:



II - DÉVELOPPEMENT DU PROJET:

A) Le problème:

Le but de ce projet était de développer en langage C un programme permettant de jouer au jeu « Bataille navale » de 2 manières différentes: une opposant deux humains et l'autre opposant un humain à une IA.

Le principe de la bataille navale étant de couler la flotte ennemie en premier, nous avons utilisé plusieurs fonctions et procédures afin d'alléger le programme principal.

B) Description de notre programme:

- Pour éviter d'avoir trop de tableau, on a créé une structure **struct BATEAU**, donc au lieu d'avoir des tableaux de tableaux on a des tableaux de bateau.
- **void Menuparam(int n, int *a1, int *b1, int *c1, int *d1, int *a2, int *b2, int *c2, int *d2, int *taille):** permet de paramétrer le nombre de bateau en fonction de la taille de grille choisie.
- **int chenille(int i, int j, int n, int m, BATEAU t[n][m]):** permet de renvoyer un le nombre de case coulée, si celui-ci correspond au nombre de case d'un bateau entier alors il nous permet de changer d'état les cases du tableau pour afficher "\$" ce qui signifie que le bateau entier a été coulé.
- **void saveGrille(FILE *fic, int n, int m, BATEAU t[n][m], int a1, int b1, int c1, int d1, int joo):** permet la sauvegarde de la partie en cours.
- **void affichage(int g, int n, int m, BATEAU tab0[n][m])// void affichage2(int n, int m, BATEAU tab0[n][m]):** les deux procédures permettent l'affichage des grille de jeu de J1 et J2.
- **void dead(int i, int j, int n, int m, BATEAU t[n][m]):** permet de changer l'état d'une case si le tir en celle-ci est réussi: s'il est réussi la case prend la valeur 1 sinon -1.
- **int TirIA(int level, int *i, int *j, int *coup, int *i0, int *j0, int *cas, int *touche0, int n, int m, BATEAU t[n][m]):** permet de générer des tirs de manière aléatoire pour l'IA en fonction du niveau choisi pour l'IA. Plus le niveau est élevé plus l'IA tirera de manière intelligente.
- **int SaisieTir(int aa, int int_choix_levelia, int joo, int *booldie, int n, int m, BATEAU t1[n][m], BATEAU t2[n][m], int *a1, int *b1, int *c1, int *d1, int *a2, int *b2, int *c2, int *d2, int *coup, int *i0, int *j0, int *cas, int *touche0):** permet de vérifier si les coordonnées de la case choisie pour le tir sont valables ou de recommencer un tir si on a déjà tiré sur la case saisie.
- **int tir(int i, int j, int n, int m, BATEAU t[n][m]):** parcourt la grille du joueur pour savoir si on a un tir sur la case en question, renvoie 1 si la case n'est pas vide 0 sinon.
- **void ini0(int n, int m, BATEAU tab0[n][m]):** sert à initialiser la grille de jeu du joueur.
- L'option "(1: QUIT)" lors des parties après chaque tir permet de sortir du jeu et de sauvegarder la partie en cours afin de charger la partie après si les joueurs veulent reprendre la partie précédente.
- **Raisonnement utilisé pour la conception de l'IA:**
 - Mode facile (LEVEL 1): choix des coordonnées des cases aléatoire.
 - Mode avancé (LEVEL 2):
 - 1er choix au hasard dans la grille:
 - si touché : jouer au hasard entre Nord, Sud, Est, Ouest de la case (si cases libres)
 - si pas touché : prochain tir au hasard entre les trois autres possibilités,...
 - Coup suivant:
 - si touché au:
 - *Nord → jouer au hasard entre Nord de la nouvelle case (si libre) et Sud de la première case (si libre)
 - *Sud → jouer au hasard entre Sud de la nouvelle case (si libre) et Nord de la première case (si libre)
 - *Est → jouer au hasard entre Est de la nouvelle case (si libre) et Ouest de la première case (si libre)

*Ouest → jouer au hasard entre Ouest de la nouvelle case (si libre) et Est de la première case

→ Coup suivant:

→ si pas touché: jouer sur l'autre case

→ si touché et bateau encore en vie au:

*Nord → jouer au hasard entre Nord de la nouvelle case (si libre) et Sud de la première case (si libre)

*Sud → jouer au hasard entre Sud de la nouvelle case (si libre) et Nord de la première case (si libre)

*Est → jouer au hasard entre Est de la nouvelle case (si libre) et Ouest de la première case (si libre)

*Ouest → jouer au hasard entre Ouest de la nouvelle case (si libre) et Est de la première case

III - DIFFICULTÉS RENCONTRÉES :

Au cours du projet, nous avons évidemment rencontré plusieurs difficultés. Pour créer la fonction Menuparam qui permet de paramétrer le nombre de bateau nous avons eu quelques problèmes : celle-ci doit permettre de choisir un nombre de bateau adapté pour la taille de la grille choisie, ex: impossible de choisir 1000 avions pour une taille 10x10 donc notre but était de faire une fonction qui permette de proposer une taille de grille minimale à prendre pour jouer. Autre complication: la sauvegarde de la partie pour pouvoir: y rejouer ultérieurement, conception d'une IA intelligente, sauvegarde de fichier. Pour la conception de l'IA nous avons réfléchi à plusieurs raisonnements mais pour certaine partie notamment le niveau 2 nous n'avons pas réussi à tout mettre en oeuvre.

Cependant un de nos problèmes les plus majeur reste la création d'une interface graphique puisque les cours de CPI2 et de CPI1 sur l'interface graphique ne sont vraiment pas clairs, il a donc été compliqué, malgré nos recherches sur internet, de réussir à créer une bonne interface graphique et de comprendre comment en faire. Nous avons donc opté pour des changements de couleurs directement via notre code à l'aide des tableaux suivants:

Fond :		Texte :	
Nombre	Couleur	Nombre	Couleur
40	Noir	30	Noir
41	Rouge	31	Rouge
42	Vert	32	Vert
43	Orange	33	Orange
44	Bleu	34	Bleu
45	Magenta	35	Magenta
46	Cyan	36	Cyan
47	Blanc	37	Blanc

IV - CONTRIBUTION ÉLÈVES:

Nous avons essayé de répartir de manière équitable le travail: chacun a pu contribuer à la réalisation des fonctions et procédures du programme à sa façon mais les contributions majeures de chacun sont les suivantes: Mlamali s'est occupé des fonctions de tirs (humains et IA), de l'affichage des grilles et

les fonctions tour de jeu. Laurène a réalisé l'initialisation des grilles, la sauvegarde des fichiers, la présentation du menu ainsi que la compilation Makefile. Mathieu s'est occupé des en-têtes et des fonctions liées à la pose des bateaux

V - BILAN PERSONNEL DE CHACUN:

A) Bilan personnel de chaque élève:

Laurène: *"Même si le délai pour le concevoir était assez court, le projet était plutôt bien à concevoir, mais je pense que pour un prochain projet de ce type nous devrions au moins avoir un réel cours pratique sur les interfaces graphiques car il est, une fois de plus, esquivé cette année ce qui nous empêche, nous élèves de comprendre ne serait-ce 10% de comment faire une interface graphique que ce soit en c ou en pascal. Même s'il faut chercher sur internet, il faut quand même avoir un minimum utile en cours."*

Mlamali: *"J'ai bien aimé la réalisation de ce projet car il nous a permis de mettre en pratique toutes nos connaissances. Le seul problème que nous avons rencontré est le manque de temps pour bien finaliser notre programme. Le projet nous a permis d'acquérir de nouvelles notions comme les couleurs, les fichiers..."*

Mathieu: *"J'ai trouvé la conception du projet intéressante, même si j'ai trouvé le délai accordé un peu court. Le projet nous a cependant permis de nous enrichir et de nous familiariser au travail en groupe. La bataille navale est un jeu que j'aime beaucoup et j'ai apprécié le reproduire."*

B) Bilan général:

Dans l'ensemble nous avons aimé la conception de ce jeu, ce fût un peu compliqué par moment mais intéressant.

FIN