S1.02 Bataille Navale



Dumai Etienne TD2/TP3

Marques Da Silva Thomas TD2/TP3

Table des matières

[Principe du jeu : 3](#_Toc152954250)

[Spécification du besoin : 3](#_Toc152954251)

[Exemple d’Interface du jeu : 3](#_Toc152954252)

[Dictionnaire des données  des types abstraits: 4](#_Toc152954253)

[Dictionnaire des données des types non abstraits 5](#_Toc152954254)

[Images Algorithme 6](#_Toc152954255)

[Initialiser la partie 6](#_Toc152954256)

[Premiere Partie : 6](#_Toc152954257)

[Deuxième partie : 6](#_Toc152954258)

[Jouer La Partie : 7](#_Toc152954259)

[Jouer la Partie Part 1 : 7](#_Toc152954260)

[Jouer la Partie Part 2 : 7](#_Toc152954261)

[genererPositionBateaux : 8](#_Toc152954262)

[Switch nbDirBateau : 8](#_Toc152954263)

[Finaliser La Partie : 9](#_Toc152954264)

[Déclaration des sous Programmes : 10](#_Toc152954265)

# Principe du jeu :

- Il s’agit d’un jeu de Bataille Navale à 2 joueurs.

 - Le joueur 1 doit couler le bateau 1, le joueur 2 doit couler le bateau 2.

 - A chaque tir le jeu informe par : « dans l’eau », « bateau 1 touché » ou bien « bateau 2 touché ».

 - La partie se termine dès qu’un des deux bateaux est « coulé » ; à savoir, lorsque les quatre cases consécutives qu’il occupe horizontalement, verticalement ou bien diagonalement, sont touchées.

# Spécification du besoin :

- Les règles du jeu sont toujours visibles.

- Les coordonnées du bateau 1 et du bateau 2 sont définies au hasard par le jeu et ne changent pas durant la partie.

- Bateau 1 et bateau 2 n’ont pas de cases en commun.

- Le numéro du tir est communiqué à chaque coup.

- A chaque coup, le joueur saisit les coordonnées voulues pour son tir (x,y) ϵ { A..I } x { 1..9 }

- Les valeurs des coordonnées saisies sont vérifiées avec les messages d’erreur correspondants.

- Chaque tir est tracé sur la mer : lorsqu’il tombe dans l’eau (ex. ‘.’), touche le bateau 1 (ex. ‘o’) ou bien le bateau 2 (ex. ’x’)

- A tout instant, un joueur peut abandonner

- On attend que les 2 joueurs aient joué pour décider s’il y a vainqueur, abandon ou bien exæquo. Attention 2 abandons dans le même tour signifie qu’ils sont exæquos

- Pour faciliter la correction, les coordonnées du bateau 1 et du bateau 2 sont affichées.

# Exemple d’Interface du jeu :

B A T A I L L E   N A V A L L E

Chaque joueur doit couler un bateau de 4 cases ( vertical ou horizontal ou diagonal)

Bateau1 = (D,2) (E,3) (F,4) (G,5)

Bateau2 = (A,7) (B,7) (C,7) (D,7)

Joueur 1 = Christophe

     A       B       C      D       E       F      G       I

1     1

2                              O        .        .                       2

3     3

4     4

5     5

6     6

7     7

8     8

9     9

     A       B       C      D       E       F      G       I

Votre 1er tir (ex. A,3) ou abandonner (@@) ?  D,2

# Dictionnaire des données  des types abstraits:

|  |  |  |  |
| --- | --- | --- | --- |
| Nom | Type | Composition | Signification |
| Direction | enuméré | DIR\_HORIZONTALE\_GAUCHE = 0,      DIR\_HORIZONTALE\_DROITE = 1,      DIR\_VERTICALE\_HAUT = 2,      DIR\_VERTICALE\_BAS = 3,      DIR\_DIAGONALE\_NO = 4,      DIR\_DIAGONALE\_NE = 5,      DIR\_DIAGONALE\_SO = 6,      DIR\_DIAGONALE\_SE = 7, | Chaque direction dans laquelle un bateau peut se générer est représentée par un entier qui vont de 0 à 7. |
| Coordonnees | enregistrement | int x = 0   int y = 0; | Type de variable ou il y a les coordonnées de l’abscisse et de l’ordonnée. |
| Bateau | enregistrement | Coordonnees pos[Nb\_CASES\_BATEAU] | Tableau de coordonné où l’indice 0 de ce tableau représente le bateau 1 et l’indice 1 représente le bateau 2 |

# Dictionnaire des données des types non abstraits

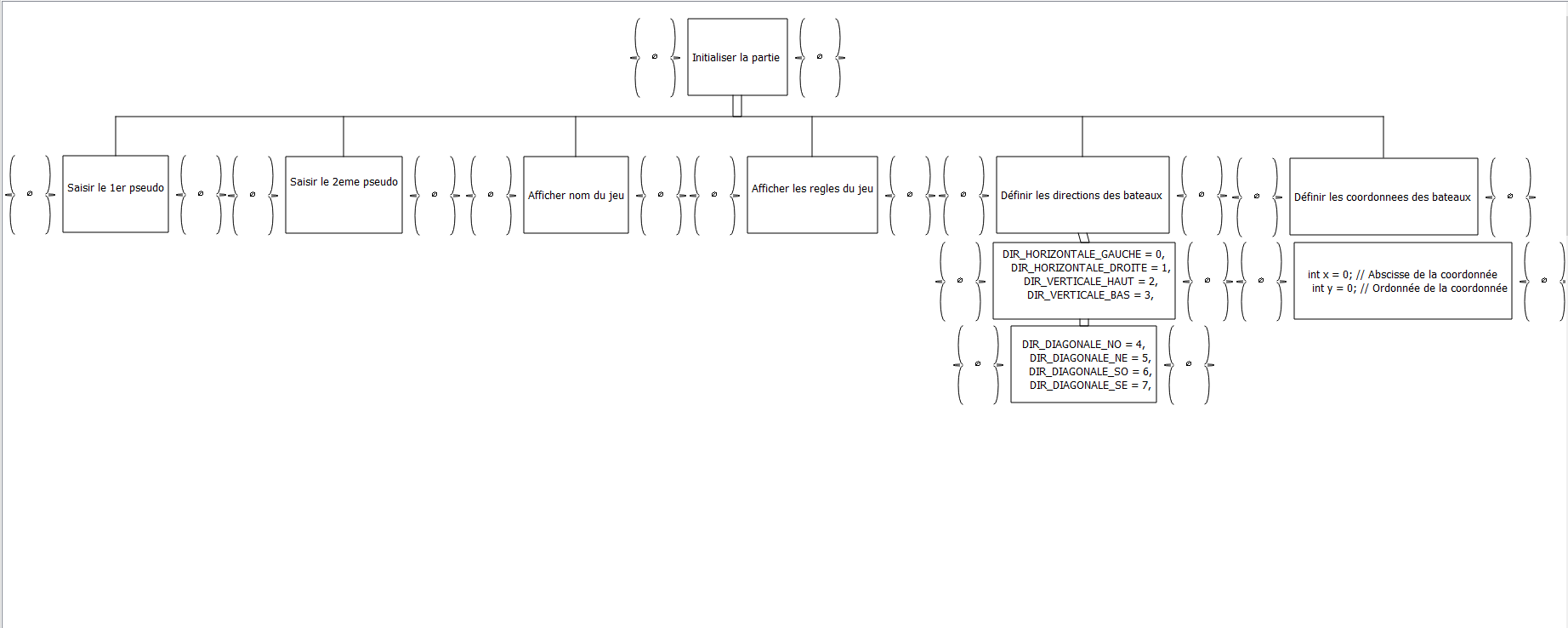
|  |  |  |  |
| --- | --- | --- | --- |
| Nom | Types | Signification | Mode d’initialisation |
| NB\_CASES\_BATEAU | Entier constant non-signé et court | Nombre de cases d’un bateau dans le jeu | Prédéfinie à 4 |
| NB\_CASES | Entier constant non-signé et court | Nombre de cases du tableau de jeu | Prédéfinie a 9 |
| tableauDeJeu[NB\_CASES][Nb\_CASES] | caractère | Tableau du jeu pouvant contenir que des caractères  En 2 dimensions de coté 9 | Prédéfinie a ‘ ‘ mais est remplie au fur et à mesure que la partie avance |
| pseudoJ1 | Chaine de caractères | Pseudo du joueur 1 qui peut être choisi en début de partie | Prédéfini a ‘ ‘ mais est modifié au début de la partie selon le choix du joueur1 |
| pseudoJ2 | Chaine de caractères | Pseudo du joueur 2 qui peut être choisi en début de partie | Prédéfini a ‘ ‘ mais est modifié au début de la partie selon le choix du joueur2 |
| NB\_BATEAUX | Entier constant non-signé et court | Nombre de bateaux présent dans le jeux | Prédéfinie à 2 |
| Bateaux | Bateaux (enregistrement) | Tableau de bateau, qui dans chaque case contient un tableau de position de longueur 4, qui contient dans chaque case un entier x et un entier y | Prédéfinie à vide mais se remplie dans l’initialisation de la partie |

# 

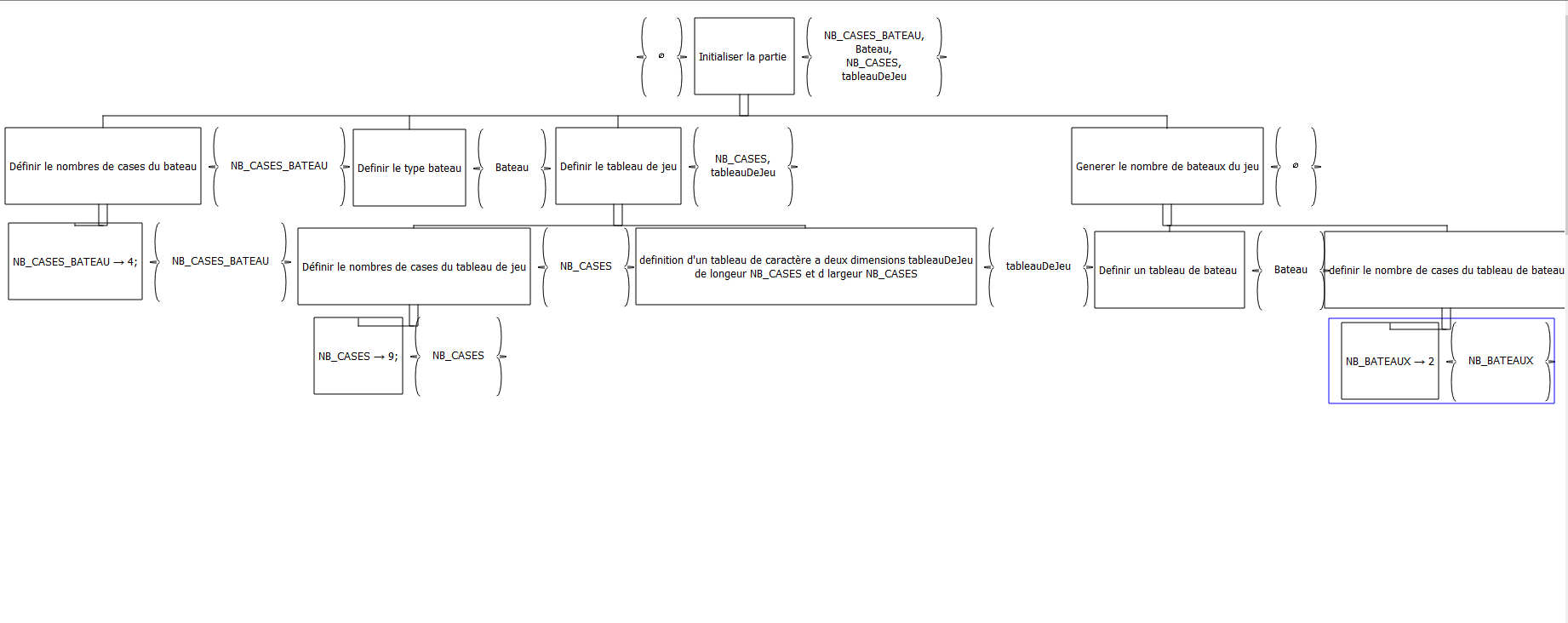
# Images Algorithme

## Initialiser la partie

### Premiere Partie :

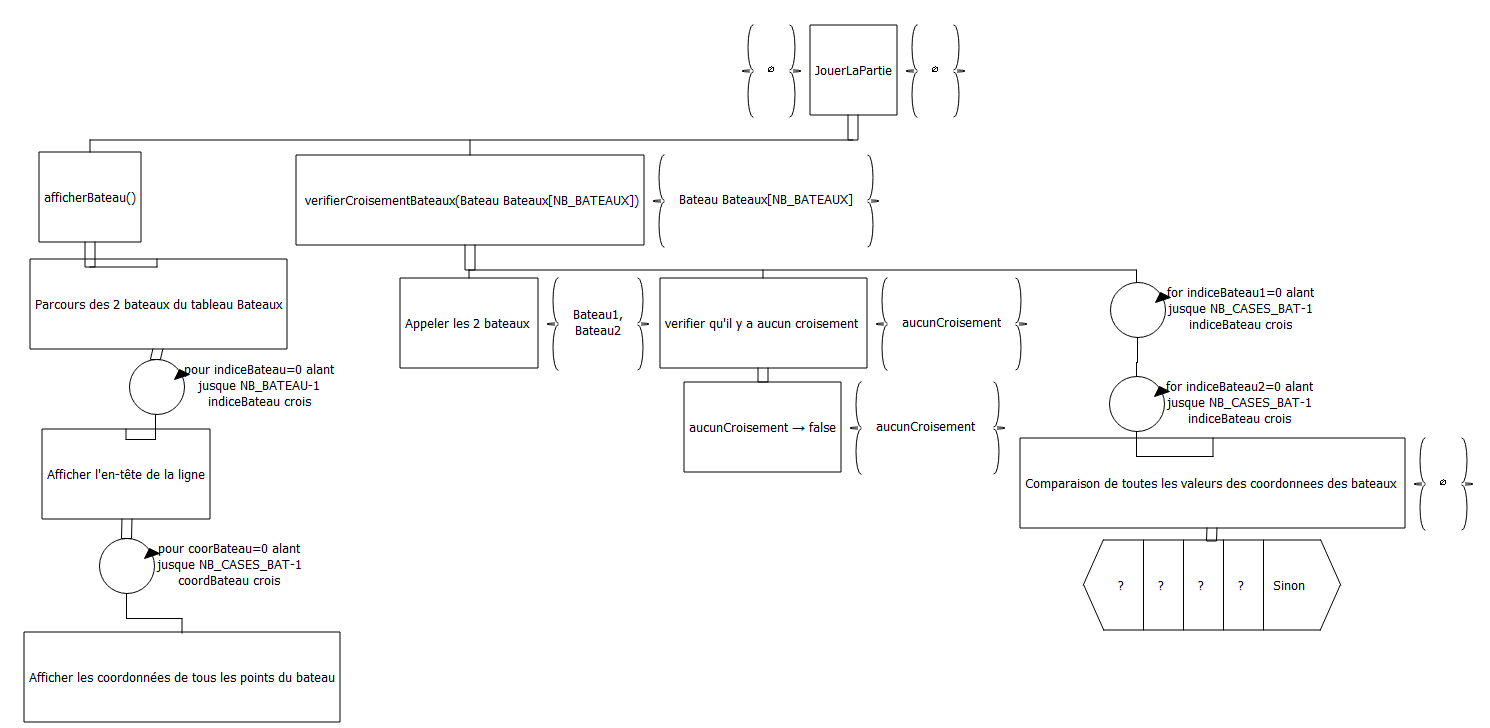


### Deuxième partie :

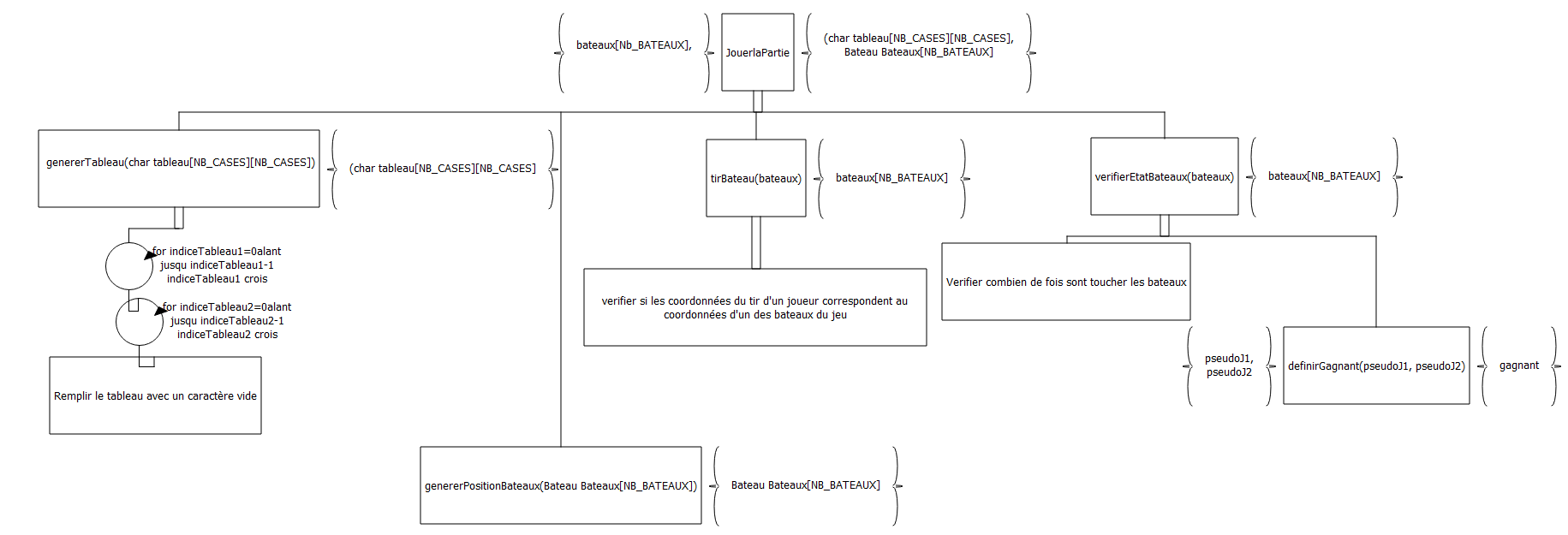


## Jouer La Partie :

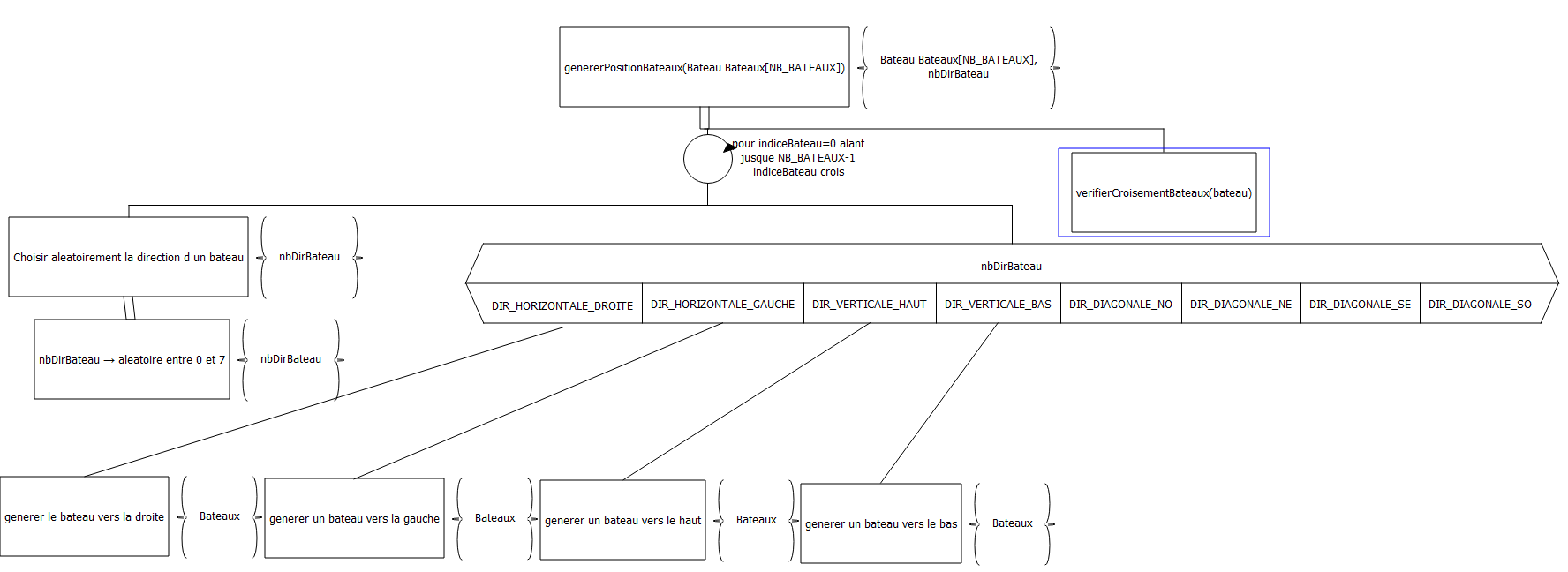
### Jouer la Partie Part 1 :



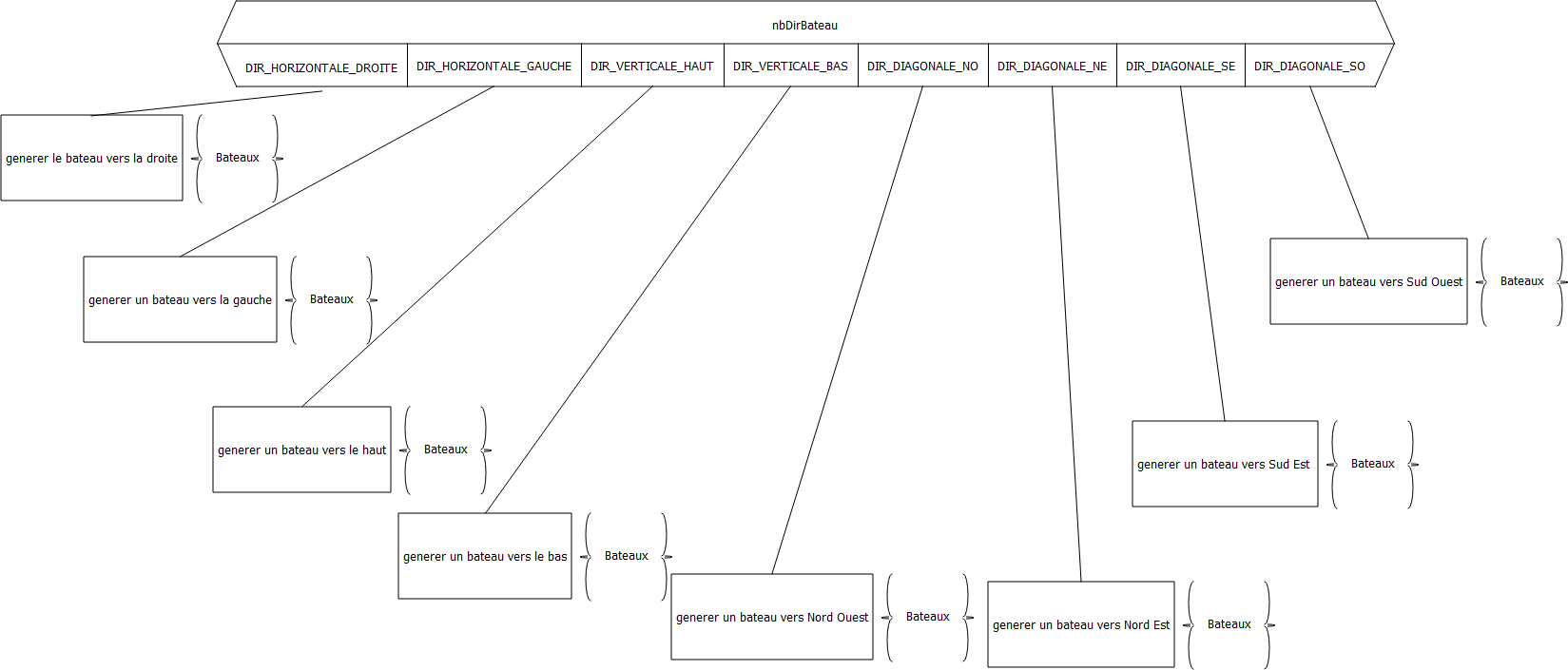
### Jouer la Partie Part 2 :



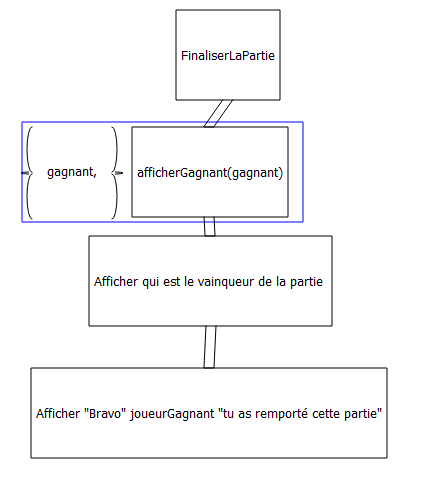
### genererPositionBateaux :



### Switch nbDirBateau :



## Finaliser La Partie :



# Déclaration des sous Programmes :

void genererTableau(char tableau[NB\_CASES][NB\_CASES]);

// BUT : générer un tableau de caractère de NB\_CASES de large et de long

void afficherTableau(const char tableau[NB\_CASES][NB\_CASES]);

// BUT : Afficher le tableau entré en paramètre

void genererPositionBateaux(Bateau Bateaux[NB\_BATEAUX]);

// BUT : générer aléatoirement les coordonnées des bateaux dans le tableau bateaux de type bateau

void afficherBateau();

// BUT : afficher les coordonées des bateaux

void verifierCroisementBateaux(Bateau Bateaux[NB\_BATEAUX]);

/\*BUT : verifier dans le tableau Bateaux de longueur NB\_BATEAU si entre les 2 bateaux

il n'y a pas de croisement (coordonnée exacte dans les 2 bateaux)\*/

void choisirPseudoDesJoueurs(string &joueur1, string &joueur2);

/\* BUT : assigner des pseudo au joueur avec les chaines de caractère

qui seront passées en paramètre\*/

void tirBateau(Bateau Bateaux[NB\_BATEAUX], string joueur1, string joueur2);

/\* BUT : réaliser des tirs entre joueur1 et joueur2 chacun leurs tours en visant les

bateaux qui ont leurs coordonées dans le tableau Bateaux de longeur NB\_BATEAUX\*/

void verifierEtatBateaux(Bateau Bateaux[NB\_BATEAUX]);

/\* BUT : verifier si un bateau du tableau BateauX de longeur NB\_BATEAUX si il est

coulé ou seulement toucher une ou plusieurs fois\*/

void definirGagnant(string joueur1, string joueur2, string gagnantPartie);

/\* BUT : définir le gagnant de la parti entre joueur1 et joueur2 et la chaine de

caractère gagnant prendra son pseudo comme valeur\*/

void afficherGagnant(string gagnantPartie);

/\* BUT : afficher le pseudo du gagnant de la partie qui est contenu dans la variable

gagnantPartie\*/