

1 Equipe

CHABANAT Matis TD 1 – TP1

LE MENN Arthur TD 1 – TP1

2 Exercice traité : TD-1-Sujet3_BatailleNavale - 1

3 Rappel des spécifications du programme

3.1 Spécifications initiales

Description textuelle du comportement des différents scénarios du programme.

Nous avons dû réaliser le jeu de la bataille navale en C++. Nous devons créer une grille de 9x9 cases avec un bateau positionné au hasard sur la grille. Ce bateau prenait une place de 4 cases de long et pouvait avoir 4 sens de direction différents (vertical, horizontal, et les 2 diagonales).

Le but du joueur est de saisir des coordonnées et le système vérifie si ces coordonnées correspondent à un point du bateau. Si le bateau est touché, nous plaçons un ‘o’ sur la grille, sinon nous plaçons un ‘.’.

Si le joueur souhaite abandonner, il n’a qu’à saisir ‘@@’ lors de la saisie des coordonnées.

Si le joueur abandonne, nous obtenons le nombre de touches du bateau que le joueur a fait après un certain nombre de tirs.

3.2 Spécifications complémentaires = extensions traitées

Aucune extension a été traitée.

4 Jeux d’essais

Scénarios qui seront testés.

Pour chaque scénario, liste des valeurs entrées et comportement attendu (et/ liste de valeurs résultats)

Jeux pour la saisie-verif :

Valeurs saisies	Résultats attendus	Résultats obtenus
A3	Pas d’erreur, les prochaines étapes sont réalisées	Pas d’erreur, les prochaines étapes sont réalisées
V4	Message envoyé : « Erreur en X ! »	Message envoyé : « Erreur en X ! »
A0	Message envoyé : « Erreur en Y ! »	Message envoyé : « Erreur en Y ! »
M0	Message envoyé : « Erreur en X ! » « Erreur en Y ! »	Message envoyé : « Erreur en X ! » « Erreur en Y ! »
@@	Pas d’erreur, le programme s’arrête (abandon)	Pas d’erreur, le programme s’arrête (abandon)

5 Algorithmes du programme (action principale et ses sous actions)

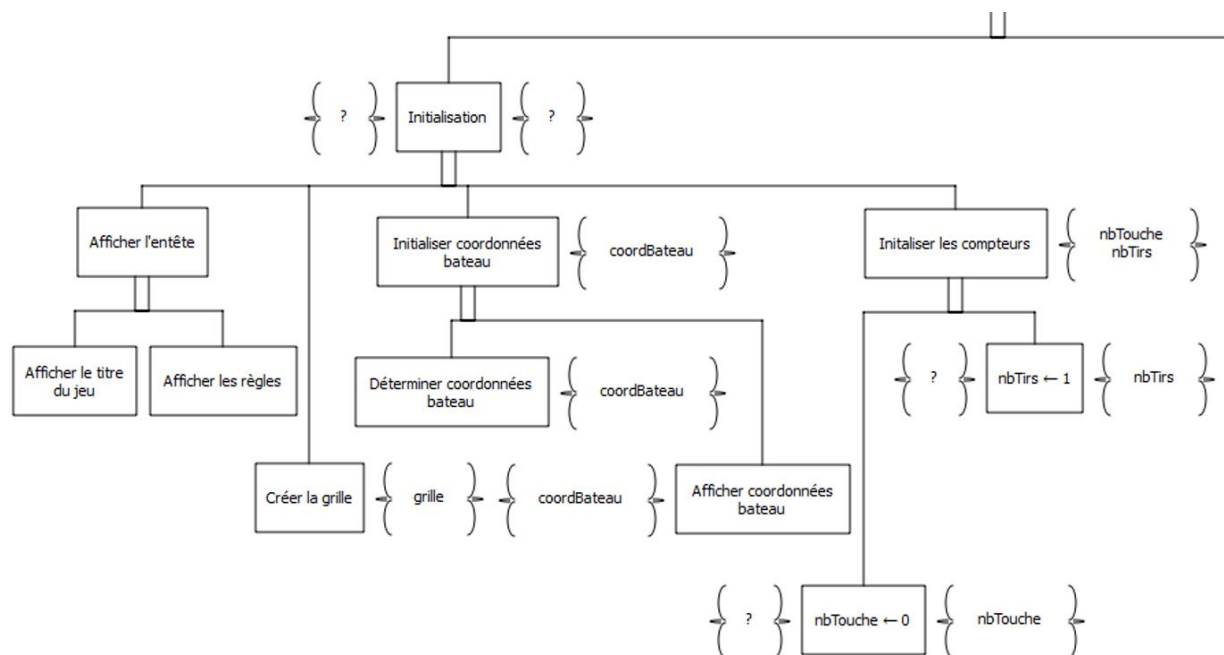
5.1 Initialisation de la partie

5.1.1 But de l'action

Initialiser la partie donc créer le plateau de jeu, mettre les compteurs à leurs valeurs par défaut et afficher les différentes règles (et pour le cas de cette version afficher la coordonnée du bateau pour faciliter la correction des professeur).

5.1.2 Stratégie de l'algorithme mise en œuvre et justification

5.1.3 Algorithme



5.1.4 Dictionnaire des éléments associés à cet algorithme

Nom	Type	Signification
grille (plateau dans certains algo / code)	Tableau de char à deux dimension	Représente la grille de jeu de la bataille navale
nbTouche	Entier court non signé	Le nombre de fois que le bateau est touché
nbTirs	Entier court non signé	Le nombre de fois que le joueur tir
coordBateau	Tableau d'indiceCoordonnée	Représente les 4 coordonnées sur lesquelles se trouve le bateau

5.2 Jouer partie 1

5.2.1 But de l'action

Vérifier la condition d'arrêt de la boucle général de « Jouer », initialiser la manche, faire une saisie vérif des coordonnées du tir

5.2.2 Stratégie de l'algorithme mise en œuvre et justification

Tout d'abord on va vérifier la condition d'arrêt (si le nombre que de tir qui ont touché le bateau est égal à sa longueur alors la partie est finit puisque le bateau est entièrement détruit), ensuite on va initialiser la

manche puis on va faire une saisie vérif assez complexe pour que les coordonnées du tir soient correctement rentrées (dans cette forme « A1 ») sinon le code renvoie une erreur qui précise la position de l'erreur (« X » ou « Y »)

5.2.3 Algorithme

Page 15

5.2.4 Dictionnaire des éléments associés à cet algorithme

Nom	Type	Signification
grille	Tableau de caractères à deux dimension	Représente le plateau de jeu de la bataille navale
nbTouches	Entier court non signé	Le nombre de fois que le bateau est touché
nbTirs	Entier court non signé	Nombre de tirs effectués depuis le début de la partie
coordBateau	Tableau d'indiceCordonnee	Les coordonnées du bateau
erreurSaisie	booléen	L'utilisateur à fait une erreur de saisie ou pas
abandon	booléen	Vrai si le joueur souhaite abandonner, faux sinon
coordTir	Coordonnee	Représente les coordonnées du tir saisies par le joueur
estTouche	booléen	Indique si le tir touche le navire

5.3 Jouer partie 2

5.3.1 But de l'action

Déterminer si le tire rentrer par le joueur touche le bateau ou non, marqué la grille en conséquence, incrémenter les variables.

5.3.2 Stratégie de l'algorithme mise en œuvre et justification

Tout d'abord on vérifie si le joueur veut abandonner (s'il a rentré @@ la saisie vérif aura mis la variable booléenne « abandon » à true, ensuite on va déterminer si le tire touche ou non en comparant la coordonnée du tir du joueur au coordonnée du bateau si la coordonnée du tir correspond à l'une des coordonnée du bateau alors le bateau est touché et après si le tir n'a pas déjà été effectuer auparavant (si la grille n'est pas déjà marqué à cette endroit-là) le bateau est marqué comme touché et la grille est marqué du symbole « o » et les variable sont incrémenter en conséquence.

5.3.3 Algorithme

Page 16

5.3.4 Dictionnaire des éléments associés à cet algorithme

Nom	Type	Signification
grille	Tableau de caractères à deux dimension	Représente le plateau de jeu de la bataille navale

nbTouches	Entier court non signé	Le nombre de fois que le bateau est touché
nbTirs	Entier court non signé	Nombre de tirs effectués depuis le début de la partie
coordBateau	Tableau d'indiceCordonnee	Les coordonnées du bateau
erreurSaisie	booléen	L'utilisateur à fait une erreur de saisie ou pas
abandon	booléen	Vrai si le joueur souhaite abandonner, faux sinon
coordTir	Coordonnee	Représente les coordonnées du tir saisies par le joueur
estTouche	booléen	Indique si le tir touche le navire

5.4 Fin de la partie

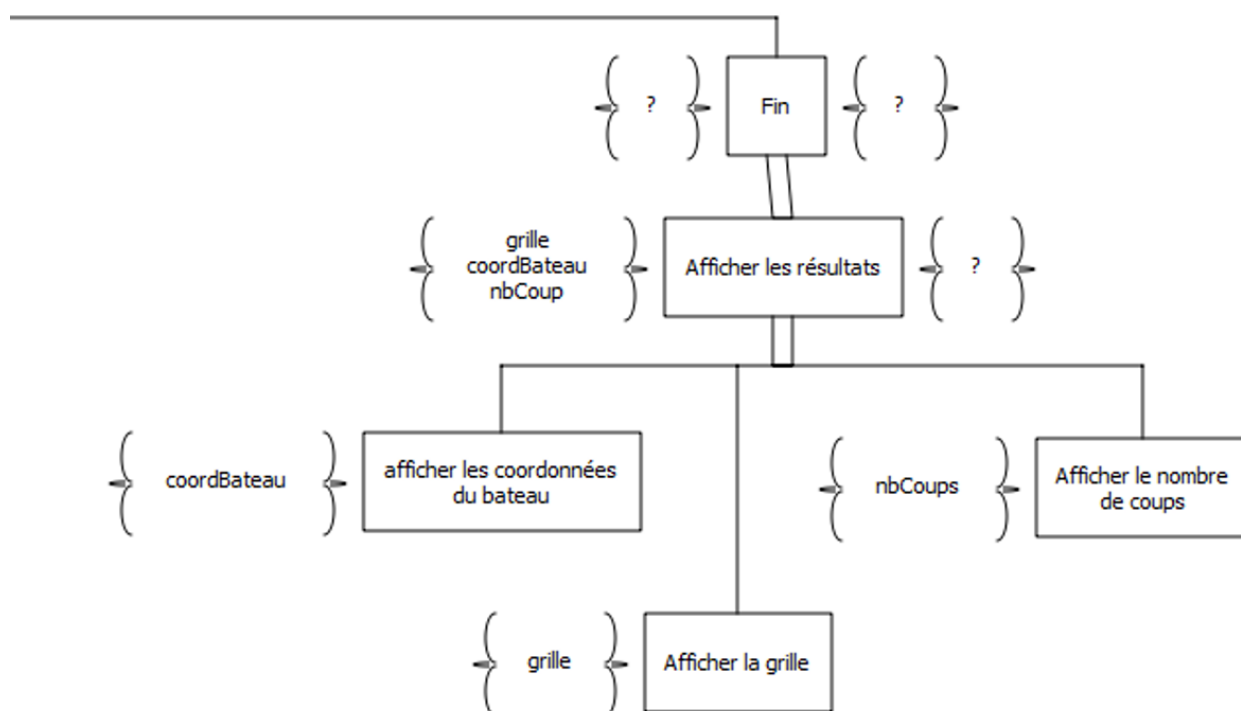
5.4.1 But de l'action

Finir la partie et afficher les résultats finals.

5.4.2 Stratégie de l'algorithme mise en œuvre et justification

On affiche simplement la grille et les données demandé par les spécifications (coordonnées du bateau, les nombre de tirs total).

5.4.3 Algorithme



5.4.4 Dictionnaire des éléments associés à cet algorithme

Nom	Type	Signification
grille (plateau dans certains algo / code)	Tableau de char à deux dimension	Représente la grille de jeu de la bataille navale
nbTirs	Entier court non signé	Le nombre de fois que le joueur tir

coordBateau	Tableau d'indiceCoordonnée	Représente les 4 coordonnées sur lesquelles se trouve le bateau
-------------	----------------------------	-----------------------------------------------------------------

5.5 Placer bateau partie 1

5.5.1 But de l'action

Initialiser le placement du bateau : Déterminer la coordonné de base du bateau, Déterminer dans quel sens ira le bateau

5.5.2 Stratégie de l'algorithme mise en œuvre et justification

La coordonnée de base du bateau est choisie au hasard grâce à la fonction random de la bibliothèque « gameTools », la coordonnée est représentée sous la forme d'un type que l'on a créé : « IndiceCoordonnee » qui est composé de deux entiers « coordX » et « coordY » qui représente le point sur l'axe X et l'axe Y.

Le sens dans lequel se dirigera le bateau est aussi déterminer au hasard grâce à la fonction random avec un entier qui va de 1 à 4, chaque entier dans cette range à un sens qui lui ai définit par la suite (après réflexion nous aurions pu faire un type Enum pour le sens du bateau)

5.5.3 Algorithme

Page 17

5.5.4 Dictionnaire des éléments associés à cet algorithme

Nom	Type	Signification
tableau	Tableau d'IndiceCoordonnee	Le tableau contenant toutes les coordonnées du bateau (c'est ce tableau qui est retourné à la fin de l'algorithme)
longueurBateau	Entier court non signé	La longueur du bateau
largeurPlateau	Entier court non signé	La largeur du plateau de jeu
longueurPlateau	Entier court non signé	La longueur du plateau de jeu
sensBateau	Entier court non signé	Le sens du bateau (chaque entier
continuer	Booléen	Condition d'arrêt de la boucle principal de l'algorithme, si continuer est sur vrai la boucle va tourner sans s'arrêter jusqu'à que les coordonnée du bateau soit correct (ne sorte pas du plateau) quand c'est le cas alors continuer passe à faux.

5.6 Placer bateau partie 2

5.6.1 But de l'action

Déterminer le reste des coordonnées des points du bateau par rapport à la coordonnée de base définit juste avant et par rapport à la direction.

5.6.2 Stratégie de l'algorithme mise en œuvre et justification

Ici ça sera juste une boucle qui va aller de 1(pas de 0 car la coordonnée de base est déjà définit) à la longueur du bateau dans lequel il y a un switch qui prends toute les possibilité de direction (1, 2, 3, 4) et par

rapport à cette direction va d'abord vérifier si les coordonnées ne vont pas sortir du plateau si c'est le cas alors on va sortir de la boucle et on va redéfinir la coordonnée de base et ainsi de suite jusqu'à trouver une coordonnée de base correcte. Si la coordonnée de base est correcte alors on va commencer à entrer dans le tableau les coordonnées.

5.6.3 Algorithme

Page 18

5.6.4 Dictionnaire des éléments associés à cet algorithme

Nom	Type	Signification
tableau	Tableau d'IndiceCoordonnee	Le tableau contenant toutes les coordonnées du bateau (c'est ce tableau qui est retourné à la fin de l'algorithme)
longueurBateau	Entier court non signé	La longueur du bateau
largeurPlateau	Entier court non signé	La largeur du plateau de jeu
longueurPlateau	Entier court non signé	La longueur du plateau de jeu
sensBateau	Entier court non signé	Le sens du bateau (chaque entier
continuer	Booléen	Condition d'arrêt de la boucle principal de l'algorithme, si continuer est sur vrai la boucle va tourner sans s'arrêter jusqu'à que les coordonnées du bateau soit correct (ne sorte pas du plateau) quand c'est le cas alors continuer passe à faux.

5.7 Afficher la grille

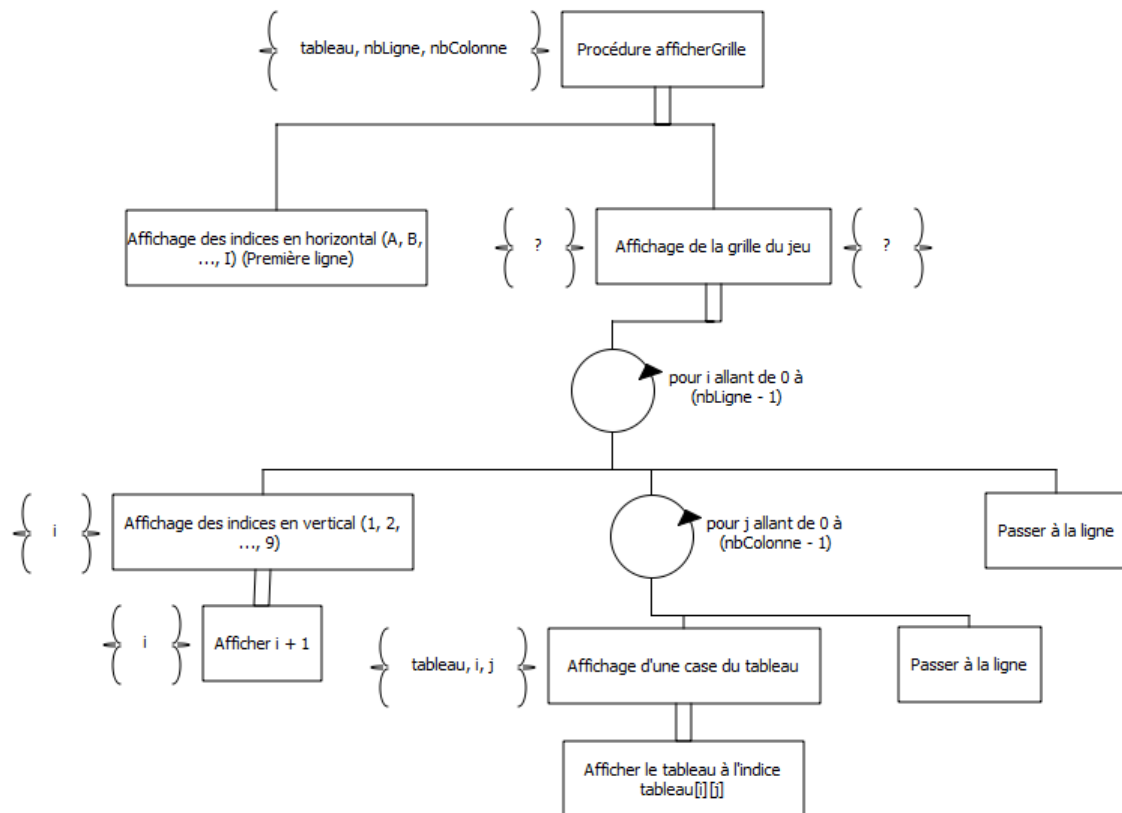
5.7.1 But de l'action

Cette procédure permet d'afficher un plateau de jeu de Bataille Navale. Cela affiche à l'écran une barre en haut avec les indices A, B, C ..., une barre à gauche avec les indices 1, 2, 3 ... et le tableau bidimensionnel au centre.

5.7.2 Stratégie de l'algorithme mise en œuvre et justification

Utilisation d'une double boucle. Une boucle permettant de gérer l'affichage en abscisses et l'autre boucle en ordonnées (autrement dit les lignes et les colonnes).

5.7.3 Algorithme



5.7.4 Dictionnaire des éléments associés à cet algorithme

Nom	Type	Signification
i	Entier non signé	Compteur de la boucle 1
j	Entier non signé	Compteur de la boucle 2
tableau	Tableau de caractères	Tableau qui représente le plateau de jeu de la bataille navale
nbLigne	Entier non signé	Il s'agit du nombre de lignes du tableau
nbColonne	Entier non signé	Il s'agit du nombre de colonnes du tableau

5.8 Afficher coordBateau

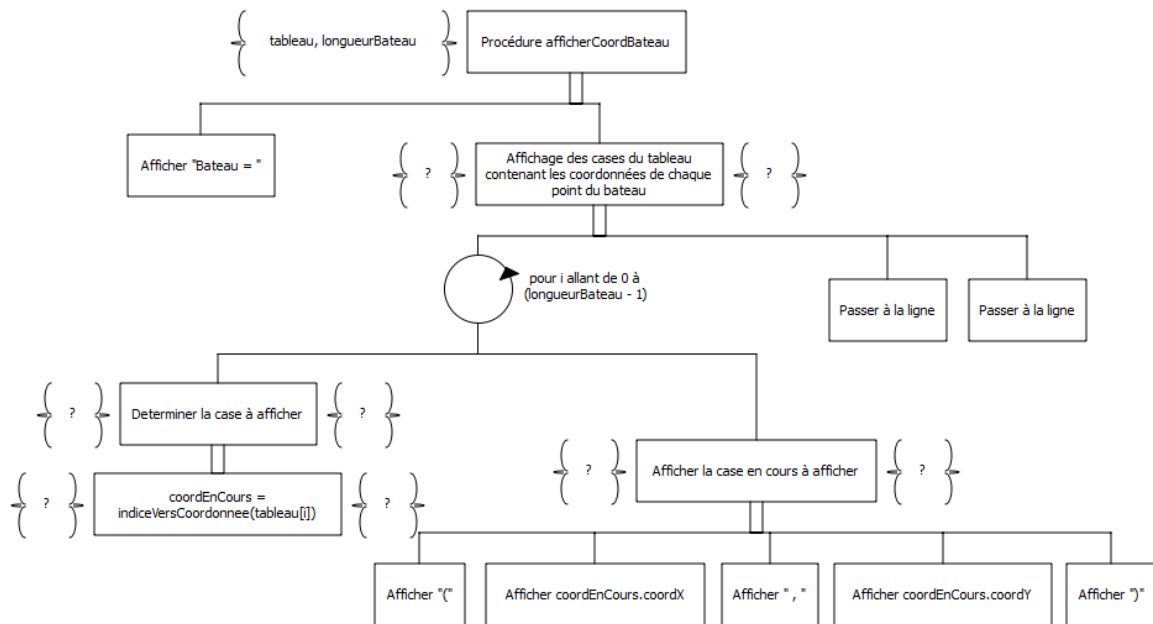
5.8.1 But de l'action

Afficher les coordonnées du bateau précédemment définies grâce à la procédure placerBateau

5.8.2 Stratégie de l'algorithme mise en œuvre et justification

Afficher chaque case d'un tableau grâce à une boucle

5.8.3 Algorithme



5.8.4 Dictionnaire des éléments associés à cet algorithme

Nom	Type	Signification
tableau	Tableau de Coordonnee	Tableau qui contient les coordonnées du bateau (chaque case contient un x et un y)
i	Entier non signé	Compteur de la boucle 1
longueurBateau	Entier court non signé	La longueur du bateau
coordEnCours	Coordonnee	Représente un x et y à afficher

5.9 Coordonne vers Indice

5.9.1 But de l'action

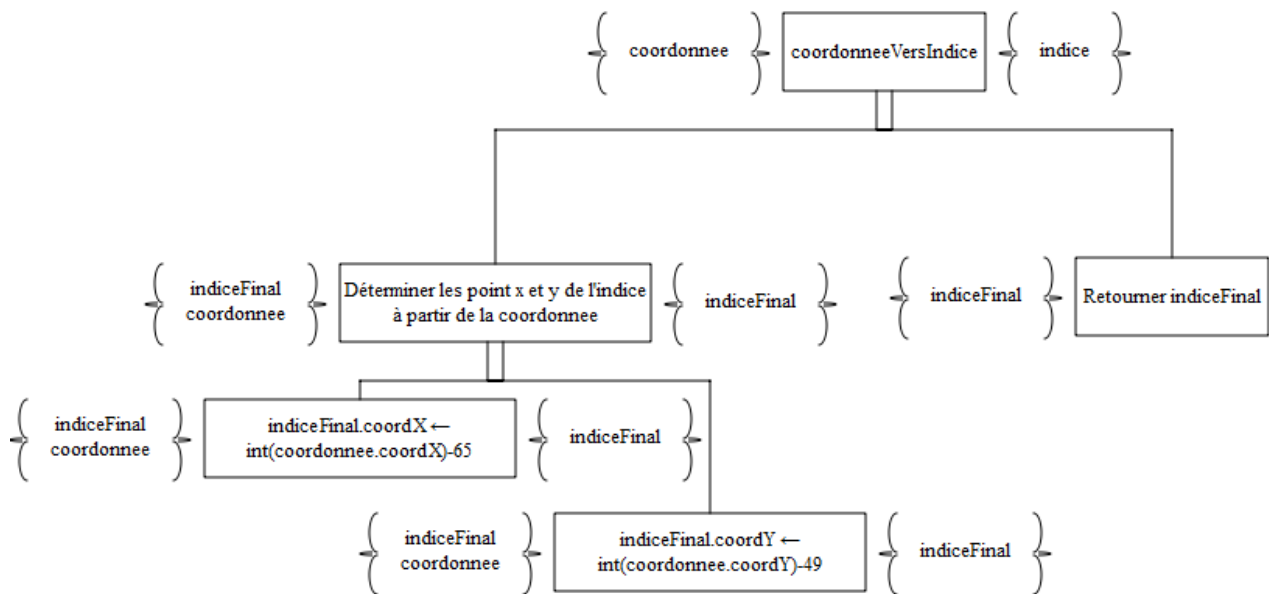
Convertir un de nos types créé (Coordonnee) en un autre type que nous avons créé (indiceCoordonnee).

5.9.2 Stratégie de l'algorithme mise en œuvre et justification

Coordonnee étant un type composé de deux variable de type char sous cette forme : « A5 » et indiceCoordonnee étant un type composé de deux variable entier qui corresponde au indice de la coordonnée dans le tableau de la grille donc « A5 » en type Coordonnee va donner « 04 » en variable indiceCoordonnee et inversement.

Pour cela on a simplement utilisé la fonction « int » qui utilise le code ASCII et on a fait une soustraction différente pour chacun des axes (-65 pour la lettre et -49 pour le chiffre).

5.9.3 Algorithme



5.9.4 Dictionnaire des éléments associés à cet algorithme

Nom	Type	Signification
coordonnee	Coordonnee (struct composé de deux caractère : coordY et coordX)	La coordonnee que l'on souhaite convertir en type IndiceCoordonnee (struct composé de deux entiers : coordY et coordX)
indiceFinal	IndiceCoordonnee	C'est la variable coordonnee convertit en IndiceCoordonnee

5.10 Indice vers Coordonne

5.10.1 But de l'action

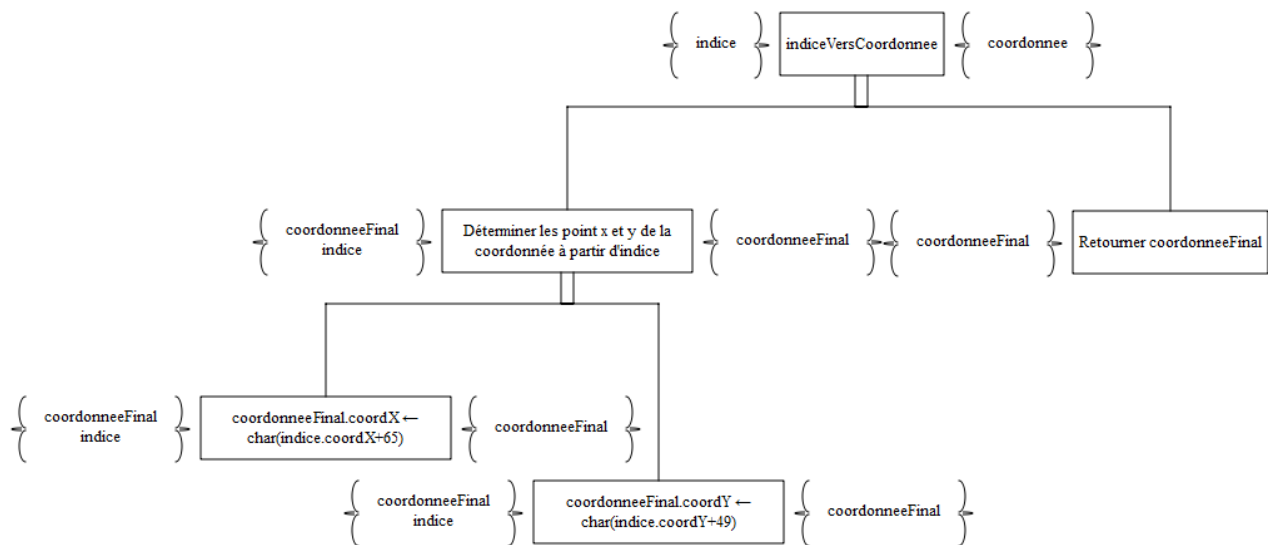
Convertir un de nos types créé (indiceCoordonnee) en un autre type que nous avons créé (Coordonnee).

5.10.2 Stratégie de l'algorithme mise en œuvre et justification

Coordonnee étant un type composé de deux variable de type char sous cette forme : « A5 » et indiceCoordonnee étant un type composé de deux variable entier qui corresponde au indice de la coordonnée dans le tableau de la grille donc « A5 » en type Coordonnee va donner « 04 » en variable indiceCoordonnee et inversement.

Pour cela on a simplement utilisé la fonction « char » qui utilise le code ASCII et on a fait une addition différente pour chacun des axes (+65 pour la lettre et +49 pour le chiffre).

5.10.3 Algorithme



5.10.4 Dictionnaire des éléments associés à cet algorithme

Nom	Type	Signification
indice	IndiceCoordonnee (struct composé de deux entiers : coordY et coordX)	Une variable que l'on souhaite convertir en type Coordonnee (struct composé de deux caractère : coordY et coordX)
coordonneeFinal	Coordonnee	C'est la variable indice convertit en Coordonnee

6 Traces d'exécution

Copies d'écran correspondant au comportement décrit au point 3 et jeux d'essais prévus au point 4.

A noter qu'en inversant les couleurs, les couleurs utilisées pour le titre du jeu et pour les messages d'erreurs ont été modifiées.

Le titre, ici en jaune, est en réalité affiché en bleu.

Les messages d'erreurs, ici en bleu ciel, sont en réalité affichés en rouge.

```

BATAILLE NAVALE

Vous devez couler un bateau de 4 cases
(vertical ou horizontal ou diagonal).

Bateau = (B,4) (C,4) (D,4) (E,4)

  A B C D E F G H I
1
2
3
4
5
6
7
8
9

Votre 1e tir (ex. A3) ou abandonner (@@) :

```

Figure 1 : Début de la partie

```

BATAILLE NAVALE

Vous devez couler un bateau de 4 cases
(vertical ou horizontal ou diagonal).

Bateau = (B,4) (C,4) (D,4) (E,4)

  A B C D E F G H I
1
2 .
3
4
5
6
7
8
9

Votre 2e tir (ex. A3) ou abandonner (@@) :

```

Figure 2 : Saisie correcte, qui ne touche pas le bateau (A2)

```

BATAILLE NAVALE

Vous devez couler un bateau de 4 cases
(vertical ou horizontal ou diagonal).

Bateau = (B,4) (C,4) (D,4) (E,4)

  A B C D E F G H I
1
2 .
3
4 o
5
6
7
8
9

Votre 3e tir (ex. A3) ou abandonner (@@) :

```

Figure 3 : Saisie correcte, qui touche le bateau (B4)

```

BATAILLE NAVALE

Vous devez couler un bateau de 4 cases
(vertical ou horizontal ou diagonal).

Bateau = (B,4) (C,4) (D,4) (E,4)

  A B C D E F G H I
1
2 .
3
4 o
5
6
7
8
9

Votre 3e tir (ex. A3) ou abandonner (@@) : A2
Vous avez déjà tiré ici !

```

Figure 4 : Coordonnées déjà saisies

```

BATAILLE NAVALE

Vous devez couler un bateau de 4 cases
(vertical ou horizontal ou diagonal).

Bateau = (B,4) (C,4) (D,4) (E,4)

  A B C D E F G H I
1
2 .
3
4
5
6
7
8
9

Votre 2e tir (ex. A3) ou abandonner (@@) : V2
Erreur en 'X' !
Votre 2e tir (ex. A3) ou abandonner (@@) :

```

Figure 5 : Coordonnées saisies contenant une erreur en X

```

BATAILLE NAVALE

Vous devez couler un bateau de 4 cases
(vertical ou horizontal ou diagonal).

Bateau = (B,4) (C,4) (D,4) (E,4)

  A B C D E F G H I
1
2 .
3
4
5
6
7
8
9

Votre 2e tir (ex. A3) ou abandonner (@@) : A0
Erreur en 'Y' !
Votre 2e tir (ex. A3) ou abandonner (@@) :

```

Figure 6 : Coordonnées saisies contenant une erreur en Y

```

BATAILLE NAVALE

Vous devez couler un bateau de 4 cases
(vertical ou horizontal ou diagonal).

Bateau = (B,4) (C,4) (D,4) (E,4)

  A B C D E F G H I
1
2 .
3
4   o
5
6
7
8
9

Votre 3e tir (ex. A3) ou abandonner (@@) : V0
Erreur en 'X' !
Erreur en 'Y' !
Votre 3e tir (ex. A3) ou abandonner (@@) :

```

Figure 7 : Coordonnées saisies contenant une erreur en X et Y

```

BATAILLE NAVALE

Vous devez couler un bateau de 4 cases
(vertical ou horizontal ou diagonal).

Bateau = (B,4) (C,4) (D,4) (E,4)

  A B C D E F G H I
1
2 .
3
4   o
5
6
7
8
9

A B A N D O N - Bateau touche 1 fois sur 2.

```

Figure 8 : Cas où le joueur souhaite abandonner (saisie de @@)

```

BATAILLE NAVALE

Vous devez couler un bateau de 4 cases
(vertical ou horizontal ou diagonal).

Bateau = (A,4) (A,5) (A,6) (A,7)

  A B C D E F G H I
1
2
3
4 o
5 o
6 o
7 o
8
9

B A T E A U   C O U L E en 4 tirs !

```

Figure 9 : Cas où le bateau est coulé (victoire)

7 Remarques

Informations que les étudiants souhaitent communiquer aux enseignants au sujet de cette SAé :

- Nous avons décidé de nous partager les fichiers grâce à GitHub pour faciliter la gestion du projet.

8 Code C++

8.1.1 Choix d'organisation des fichiers composant le code source

Quelques mots pour expliquer les choix d'organisation du code dans le/les fichiers rendus.

Liste des fichiers et contenu (en intention = quels types de sous-programmes il contient ; et/ou extension = liste des sous-programmes qu'il contient).

Nous avons organisé le code avec un **main**, et le fichier **d'interface** et le **cpp** du sous-programme.

Le main contient toutes les instructions à réaliser l'exécution du jeu, l'interface contient toutes les déclarations des sous-programmes que nous avons utilisé ainsi que les types que nous avons créé spécialement pour la bataille navale, et enfin dans le cpp, on retrouve le corps de tous les sous programmes.

Voici une liste de tous les sous programmes que nous avons utilisés :

- afficherRegles : procédure qui affiche le titre du jeu et les règles du jeu
- afficherGrille : procédure qui affiche le tableau rentré en paramètres
- placerBateau : procédure qui définit le sens du bateau et qui place les 4 points du bateau
- afficherCoordBateau : procédure qui affiche les coordonnées du bateau
- indiceVersCoordonnee : fonction qui permet de changer le type d'une coordonnée (ici de l'indice (pour étudier dans le tableau) à une coordonnée réelle (telle que A3))
- coordonneeVersIndice : fonction qui permet de changer le type d'une coordonnée (ici de la coordonnée réelle à l'indice)

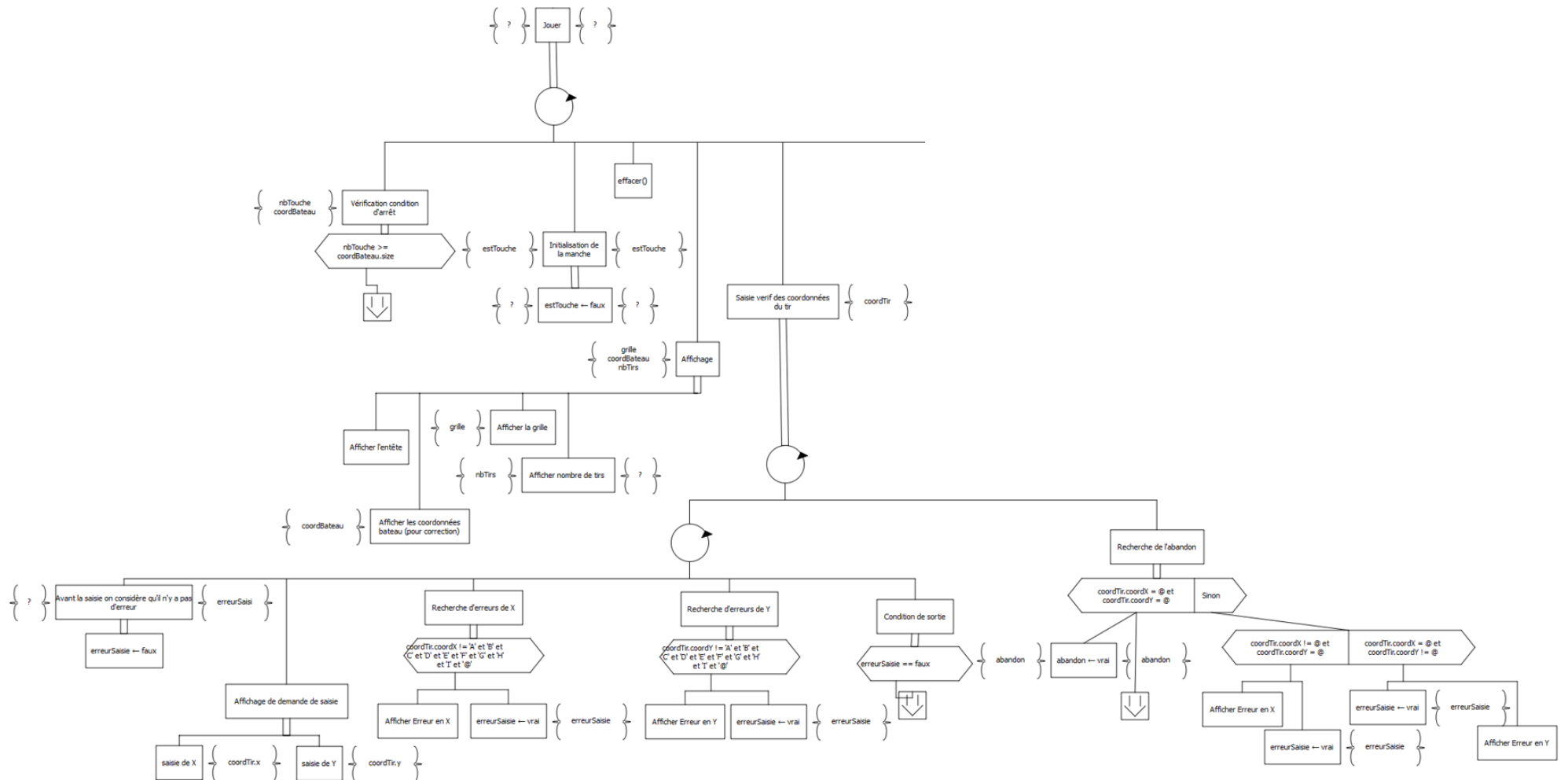
De plus, nous avons utilisé Game-Tools de Patrick Etcheverry pour ajouter de la couleur ou effacer l'écran par exemple.

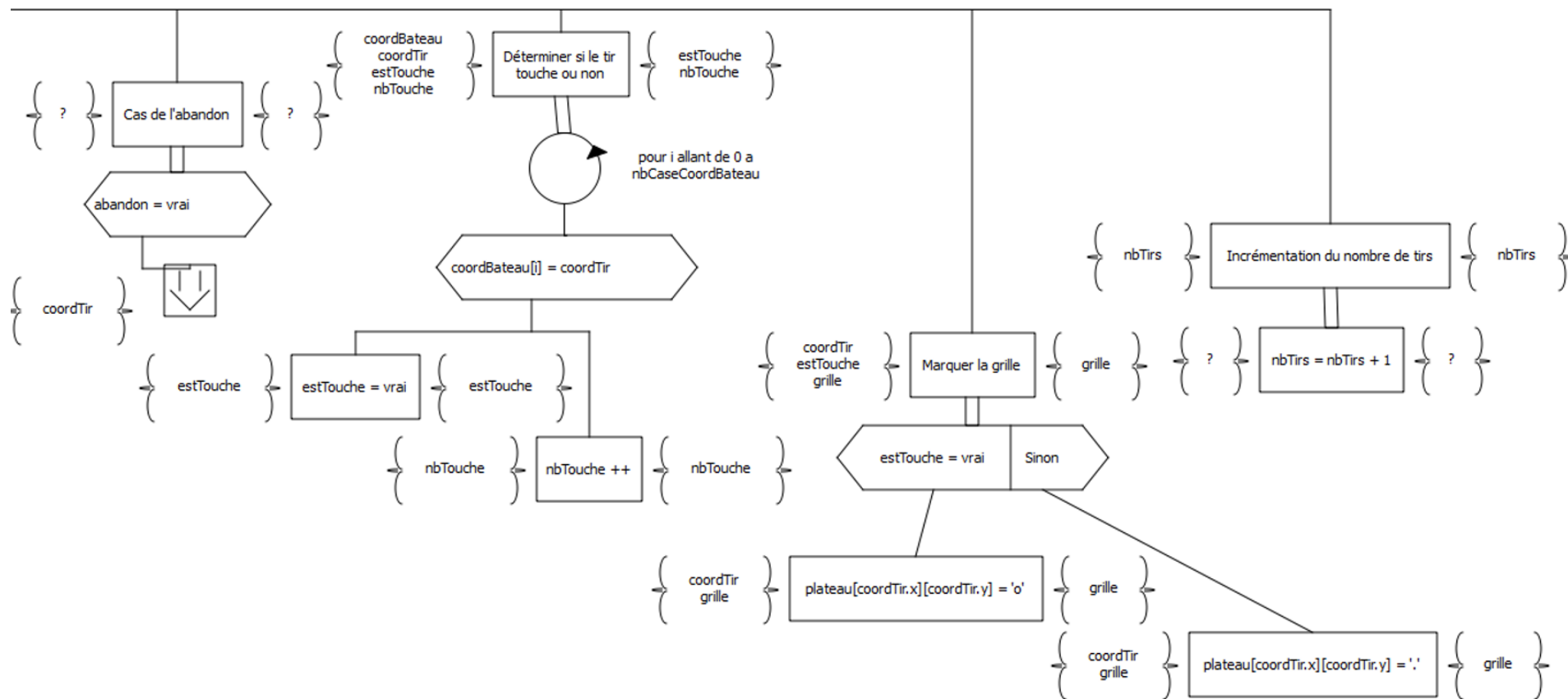
8.1.2 Code source

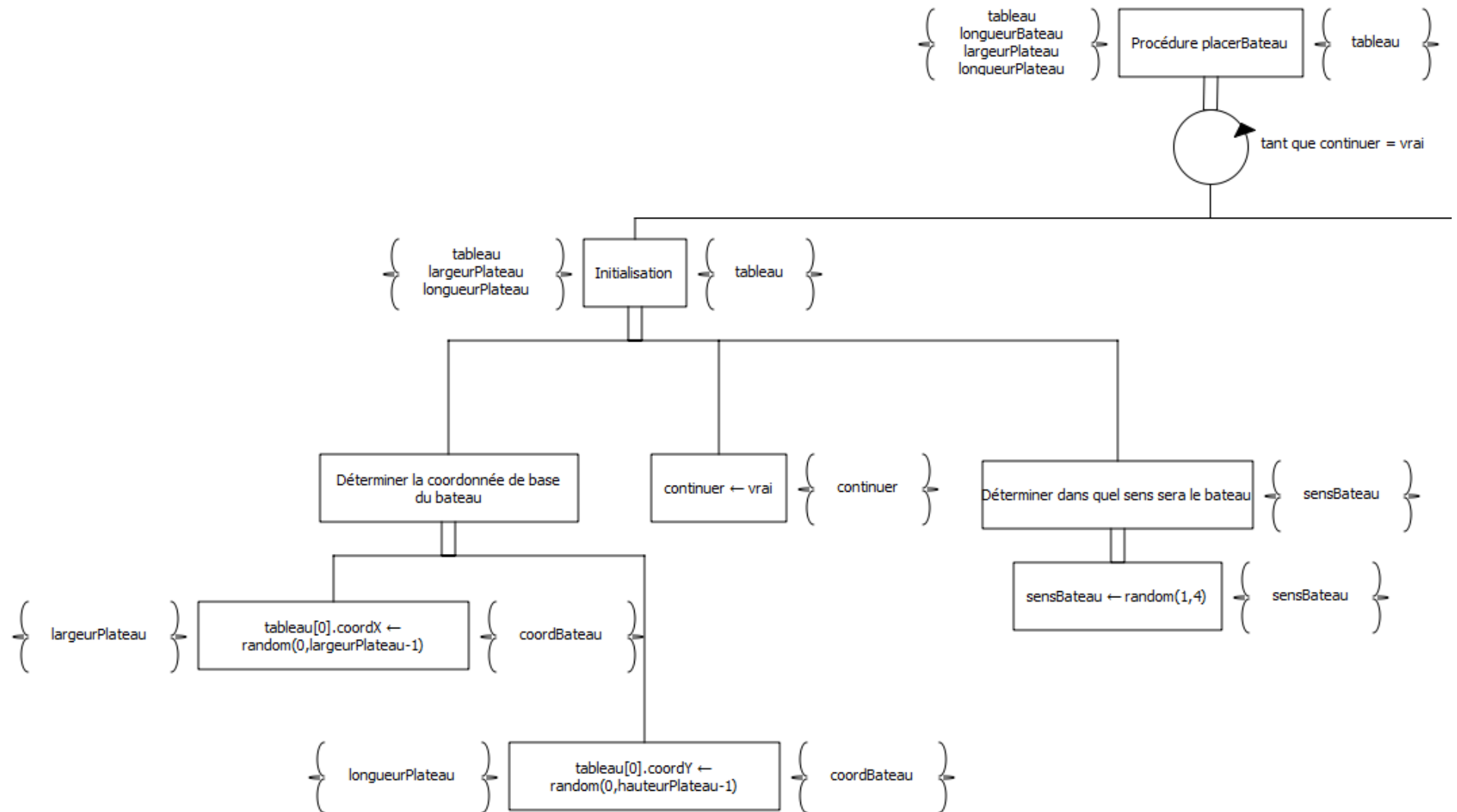
Les fichiers .cpp et .h ne sont pas inclus dans le présent document, mais sont joints au dossier avec un paragraphe en-tête certifiant l'originalité du code produit.

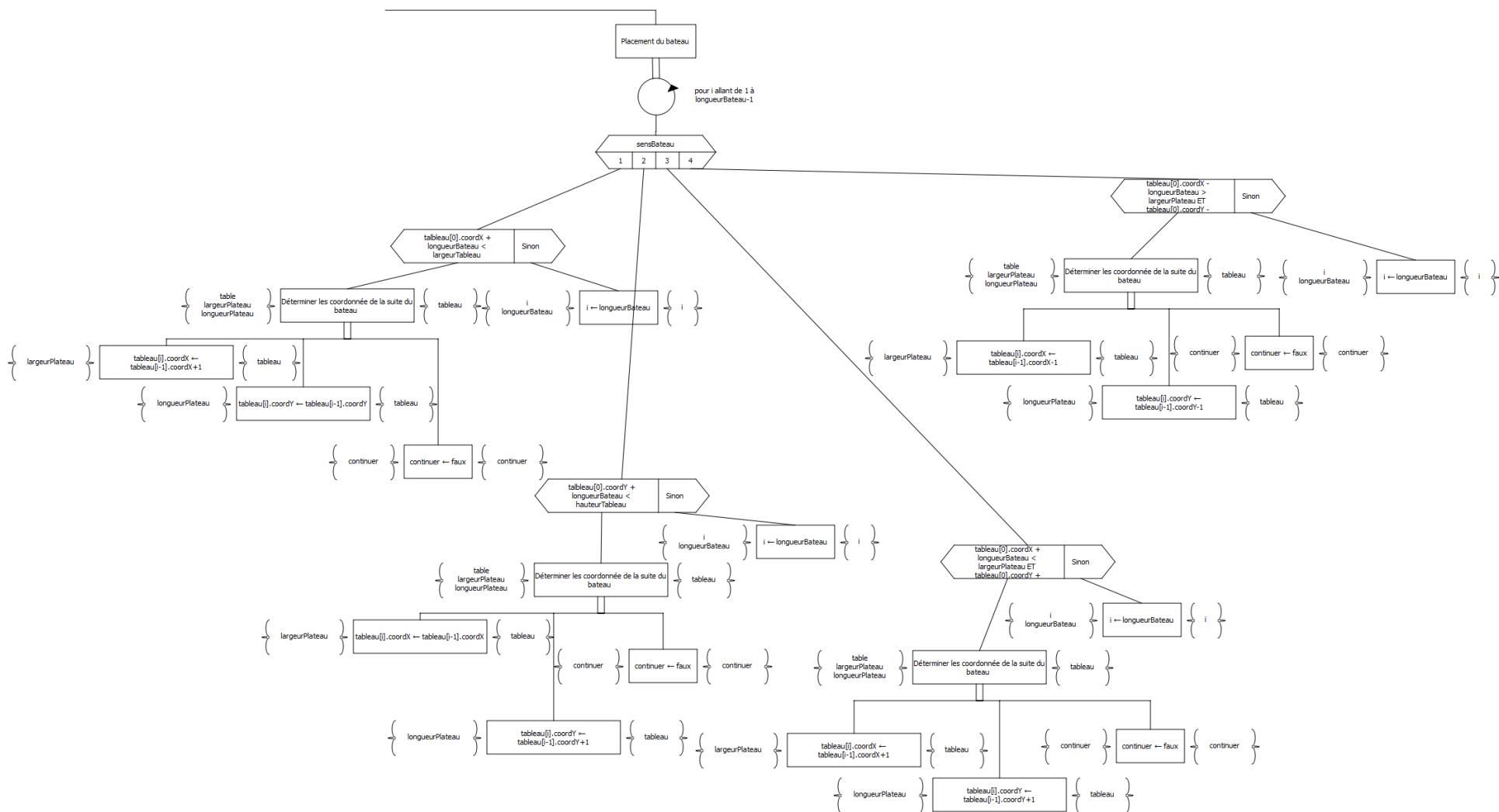
SAÉ 1.02 – Comparaison d’approches algorithmiques

Exemple de mise en page pour insérer un algorithme lisible









Exemple de page située après une / des pages d’algorithmes