Explications concernant le développement du morpion

S.A.É. 1.02 : Comparaison d'approches algorithmiques

B.U.T. Informatique : semestre 1 (2021-2022)
I.U.T de Bayonne et du Pays Basque

Pour le lundi 10/01/2021

DARGAZANLI Nicolas, TD 1 TP 2
MAURICE Alexandre, TD I TP 2

1 Préambule

Pour pouvoir utiliser les programmes que nous avons développés, vous devrez vous munir au préalable de la bibliothèque game-tools car celle-ci n'est pas fournie dans l'archive (à déposer dans le même dossier), qui est utilisée uniquement dans le *main.cpp*.

Nous avons développé notre programme et notre module de manière que le module soit complètement fonctionnel pour quelqu'un qui souhaiterait développer son propre morpion, ou encore s'il ne souhaite pas effacer les règles et différentes étapes de la phase de jeu. Pour ce faire, nous avons dû utiliser la primitive *effacer( )* dans le programme principal, ce qui justifie le fait que celui-ci ne soit pas uniquement constitué de sous programmes.

Le programme disponible traite toutes les spécifications présentées dans le 2, y compris les deux extensions.

Pour des soucis de lisibilité, il se peut que certains algorithmes ne respectent pas les niveaux algorithmiques du formalisme et soient donc légèrement enchevêtrés (comme c'est le cas pour l'algorithme principal par exemple). Nous préférons procéder ainsi pour ne pas complexifier la lecture de ce PDF, et qu'on puisse ainsi le lire sans trop s'arrêter. Pour que ce document soit aussi plus pertinent nous avons décidé de mettre dans le même ordre les algorithmes et la déclaration/définition des sous-programmes dans le module.

2 Rappel des spécifications du programme

2.1 Rappel des spécifications externes du programme

On demande d'afficher :

- Les règles du jeu avant de commencer le jeu.
- L'état de la grille à chaque étape du jeu, ainsi que le numéro de l'étape en cours.
- Un message de fin de jeu indiquant que la partie est finie, le type de fin de partie, éventuellement le nom du gagnant et le numéro de l'étape de la victoire.

2.2 Rappel des spécifications internes du programme

Ce programme devra :

- Demander la dimension du morpion.
- Procéder à une phase de personnalisation des joueurs (noms, symboles). Le nom et le symbole du joueur 2 sera vérifié.
- Jouer la partie (système de tour par tour, vérification de victoire).
- Permettre à chaque joueur de saisir son symbole à une coordonnée précise vérifiée (numéro de ligne commençant par 1, de même pour le numéro de colonne).

2.3 Spécifications complémentaires (extensions traitées)

Nous avons traité les deux extensions, à savoir :

- Le joueur n°2 est obligé de fournir un prénom et symbole différents de ceux du joueur n°1.
- Les dimensions de la grille peuvent être paramétrées en début de jeu : entre 9 et 27 cases (une grille 3x3 à une grille de 9x9). La grille devant toujours être carrée.

3 Algorithmes du programme (action principale et ses sous-actions)

3.1 Algorithme principal

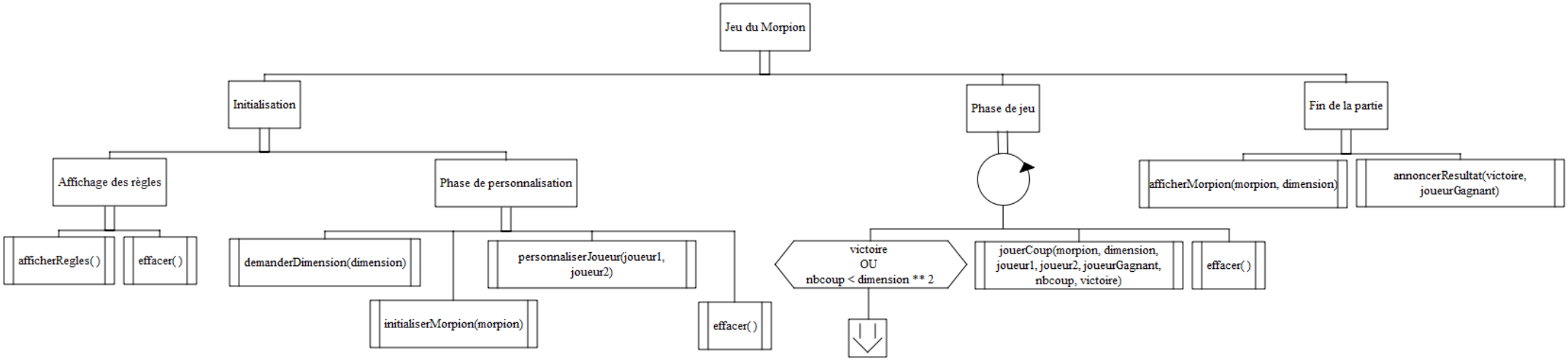
3.1.1 But de l'action

Ce programme permet de dérouler la partie convenablement en effaçant l'invite à chaque action. Il a en fonction du module morpion que nous avons développé, et comprend donc des primitives issues de *game-tools* pour des questions de modularité.

3.1.2 Stratégie de l'algorithme mise en œuvre

Il s'agit ici d'une décomposition séquentielle constituée des trois grandes phases classiques d'un jeu. Pour la phase de jeu, nous justifions la décomposition itérative pour les mêmes questions de modularité énoncées en 3.1.1.

3.1.3 Algorithme



Nous détaillons dans le reste du 3. chacun des sous-problèmes de l'algorithme général. L'algorithme de la procédure *afficherRegles( )* ne sera pas présent sur ce document car il consiste juste en une série d'affichages sommaires. Pour ce qui est de *effacer( )*, nous admettons simplement qu'il efface ce qui est affiché actuellement à l'écran, il dépend entièrement de la bibliothèque *game-tools*.

3.1.4 Dictionnaire de données associé

Nom	Type	Signification
morpion	Matrice de caractères	Support de jeu, permet d'enregistrer les coordonnées saisies par les joueurs
dimension	Entier non-signé	Dimension jouable du morpion
nbCoup	Entier non-signé	Nombre de coups joués pendant la partie
victoire	Booléen	Indique si une victoire est déclarée dans le jeu
joueur1	Joueur	Le premier joueur de la partie
joueur2	Joueur	Le deuxième joueur de la partie
TAILLE_MAX	Entier non-signé	Taille maximale possible pour une grille de morpion

3.2 Sous-programme initialiserMorpion( )

3.2.1 But de l'action

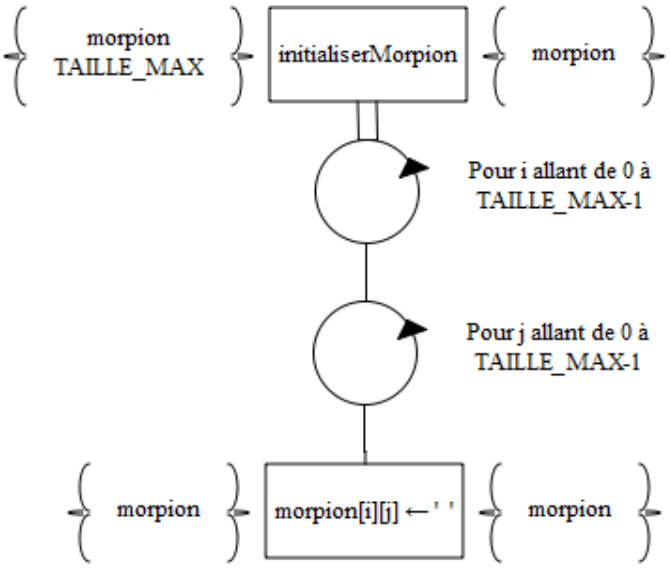
Cette procédure permet d'initialiser la grille du morpion. Afin d'éviter les « variables magiques » dans notre code, et pour des soucis de passages de paramètres de la procédure, nous avons décidé de créer un tableau de caractères de taille fixe TAILLE_MAX = 9. Le jeu se jouera donc avec une autre variable nommée dimension (cf. *demanderDimension( )*), qui délimitera la zone jouable.

3.2.2 Stratégie de l'algorithme mise en œuvre

Cet algorithme est constitué d'une double décomposition itérative afin de créer la totalité du morpion lignes par colonnes. Décomposition à la fin de laquelle on assigne à la position courante (i, j) le caractère « espace ».

Note : un des joueurs ne peut pas saisir un espace à la place de son symbole (ce qui le ferait en théorie gagner instantanément avec cette manière d'initialisation) car l'invite de commande ne permet pas de saisir un espace.

3.2.3 Algorithme



3.2.4 Dictionnaire de données associé

Nom	Type	Signification
morpion	Matrice de caractères	Support de jeu, permet d'enregistrer les coordonnées saisies par les joueurs
TAILLE_MAX	Entier positif non-nul constant	Taille maximum du morpion
i	Entier positif non-nul	Indice de boucle
j	Entier positif non-nul	Indice de boucle

3.3 Sous-programme personnaliserJoueur( )

3.3.1 But de l'action

Cette procédure permet d'assurer la personnalisation des joueurs. Le joueur1 peut choisir librement son nom et son symbole (à condition qu'il ne contienne qu'un seul caractère et qu'il soit accepté par l'invite de commande), tandis que le joueur2 est obligé de saisir un nom et un symbole différent de celui du joueur1.

3.3.2 Stratégie de l'algorithme mise en œuvre

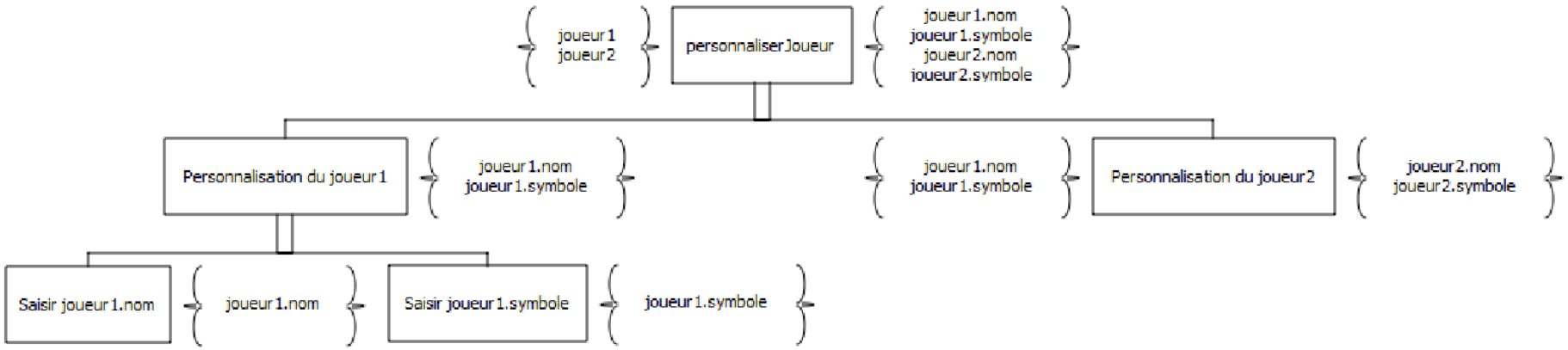
Simple saisie du nom et symbole du joueur1, puis une *saisie-vérif* du nom et du symbole du joueur2.

3.3.4 Dictionnaire de données associé

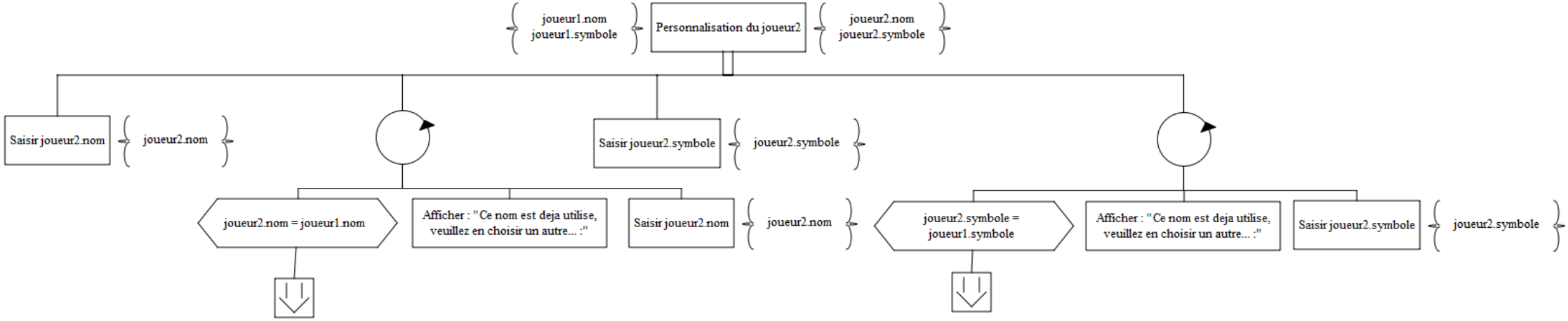
joueur1	Joueur	Le premier joueur de la partie
Joueur2	Joueur	Le deuxième joueur de la partie

L'algorithme est placé à la suite du dictionnaire pour des raisons de gain d'espace.

3.3.3 Algorithme



Le sous-problème « Personnalisation du joueur2 » est illustré ci-dessous.



3.4 Sous-programme *demanderDimension( )*

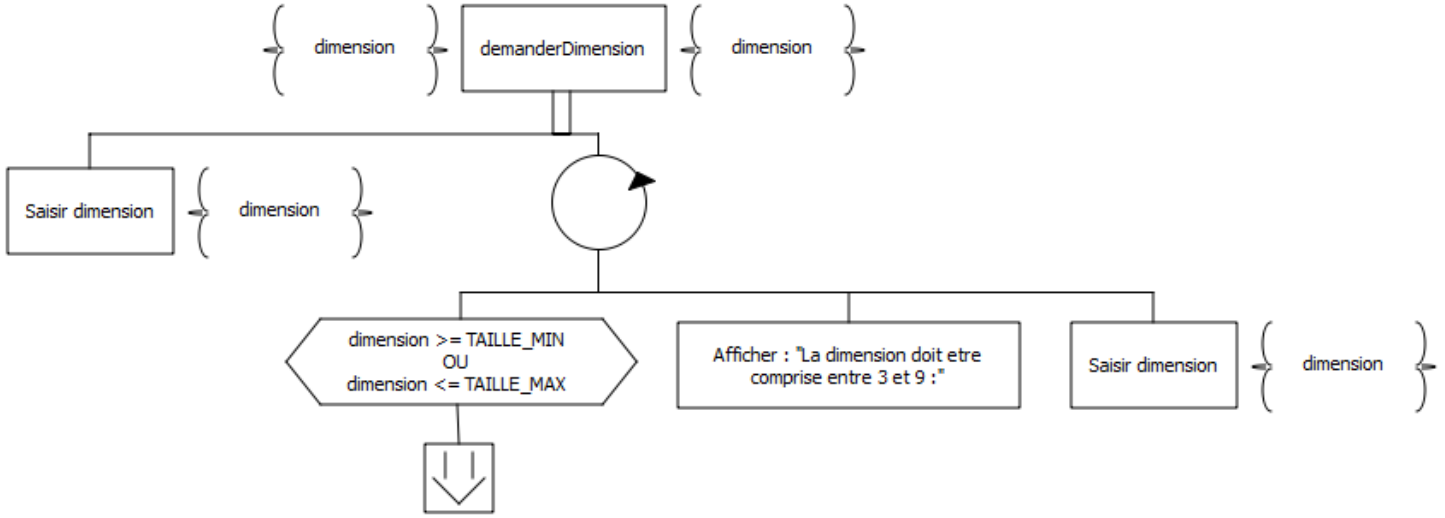
3.4.1 But de l'action

Cette procédure demande à l'utilisateur de saisir la dimension du morpion. Un *saisie-vérif* est effectué pour s'assurer que la dimension du morpion est comprise entre 3 et 9. Nous avons choisi une procédure afin d'éviter d'écrire ***unsigned short int dimension = demanderDimension( )*** qui peut porter à confusion et crée une répétition.

3.4.2 Stratégie de l'algorithme mise en œuvre

Saisie-vérif de dimension.
Post-condition : 3 ≤ dimension ≤ 9

3.4.3 Algorithme



3.4.4 Dictionnaire de données associé

Nom	Type	Signification
dimension	Entier positif non-nul	Dimension du morpion

3.5 Sous-programme *afficherMorpion( )*

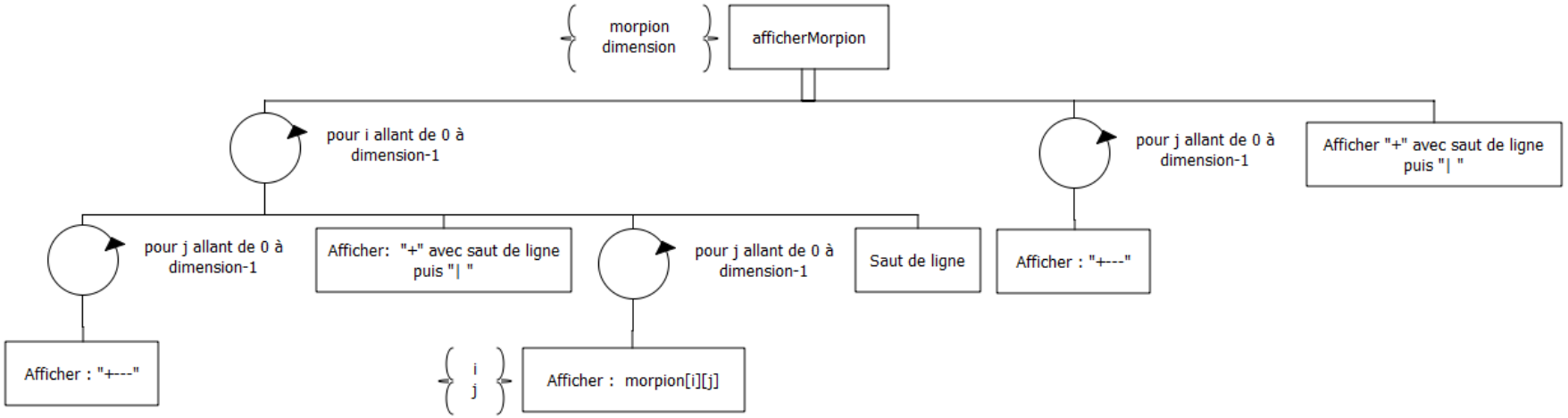
3.5.1 But de l'action

Procédure permet d'afficher un morpion graphique complété par les saisies des joueurs.

3.5.2 Stratégie de l'algorithme mise en œuvre

Pour un affichage que nous jugeons le plus correct possible, nous effectuons un encadrement avec d'abord une double décomposition itérative permettant d'assurer la séparation des lignes et des colonnes, en affichant aussi les symboles saisis par les joueurs. Pour fermer ce quadrillage, nous ajoutons une boucle à la fin et terminons par la dernière séparation verticale située en bas à droite.

3.5.3 Algorithme



3.5.4 Dictionnaire de données associé

Nom	Type	Signification
Morpion	Matrice de caractère	Support de jeu, permet d'enregistrer les coordonnées saisies par les joueurs
dimension	Entier non signé	Dimension jouable du morpion
i	Entier non signé	Indice de ligne du morpion
j	Entier non signé	Indice de colonne du morpion

3.6 Sous-programme *annoncerResutltat( )*

3.6.1 But de l'action

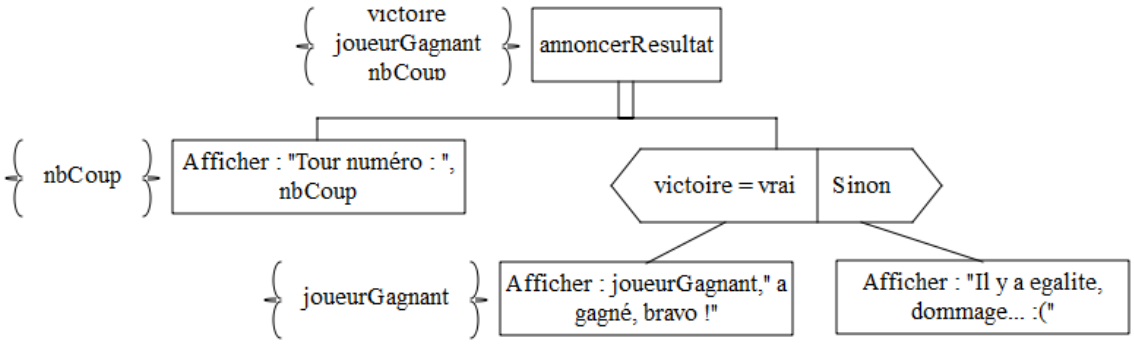
Permet d'afficher le résultat en fonction de s'il y a une victoire, et si c'est le cas on affiche le joueurGagnant. Sinon, il y a égalité. On affiche aussi le numéro de la dernière tentative : celle qui a permis de finir la partie.

3.6.2 Stratégie de l'algorithme mise en œuvre

Décomposition séquentielle permettant d'afficher le nombre de tours, puis une condition de type *si-sinon*.
Décomposition alternative suivant deux conditions :

- La partie a été remportée, et on affiche le joueurGagnant.
- La partie mène a une égalité.

3.6.3 Algorithme



3.6.4 Dictionnaire de données associé

Nom	Type	Signification
victoire	Booléen	Indique si une victoire est déclaré dans le jeu
joueurGagnant	Joueur	Le joueur qui est considéré gagnant
nbcoup	Entier non signé	Nombre de coups joués pendant la partie

3.7 Sous-programme *prochainJoueur( )*

3.7.1 But de l'action

Dans cette fonction, on retourne le joueur qui devra jouer le prochain coup. Pour ce faire, on utilise nbCoup pour le calcul :

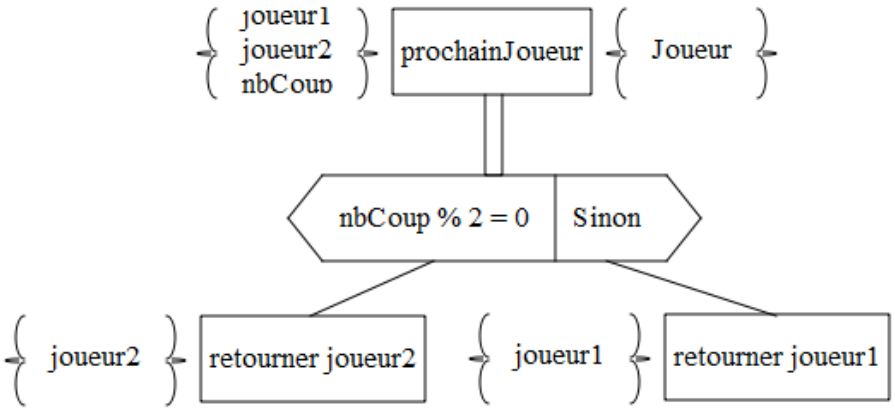
- Si nbCoup est pair, c'est au joueur2 de jouer.
- Sinon, c'est au joueur1 de jouer.

3.7.2 Stratégie de l'algorithme mise en place

Décomposition alternative de type *si-sinon* où la condition est que le nombre de coups nbCoup est pair. On associe alors dans ce cas le joueurActuel (retourné en tant que type Joueur, mais observé dans le la procédure *jouerCoup()*).

Note importante : le joueur1 correspond aux nombres de coups impairs car on considère que le premier coup est compté avant que celui-ci joue. Nous avons décidé de procéder ainsi pour faciliter la lisibilité du programme et avoir un affichage du nombre de coups corrects.

3.7.3 Algorithme



3.8 Sous-programme *jouerCoup( )*

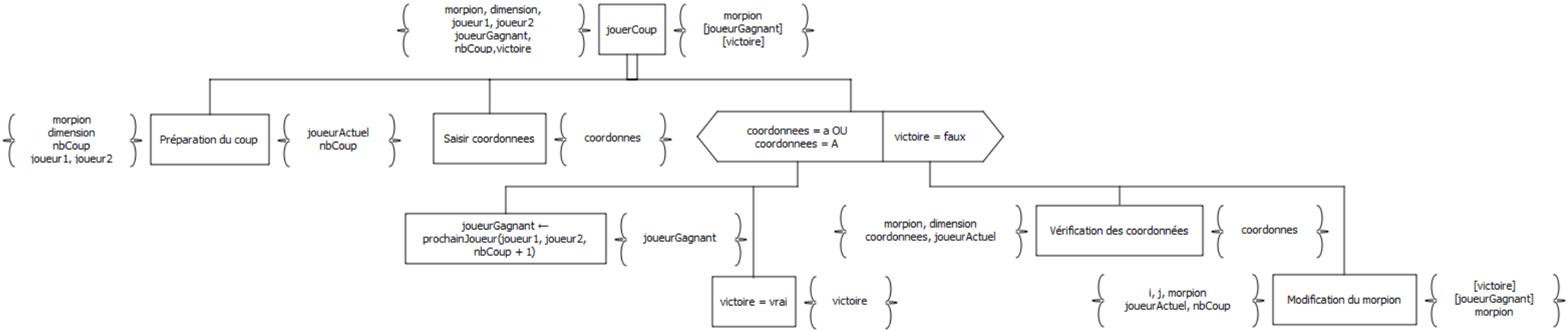
3.8.1 But de l'action

Cette procédure permet simplement de jouer un coup, en prenant en compte le système de tour par tour. Pour optimiser un maximum les opérations, on ne vérifie la victoire qu'aux lignes concernées par la dernière coordonnée saisie par le joueur.

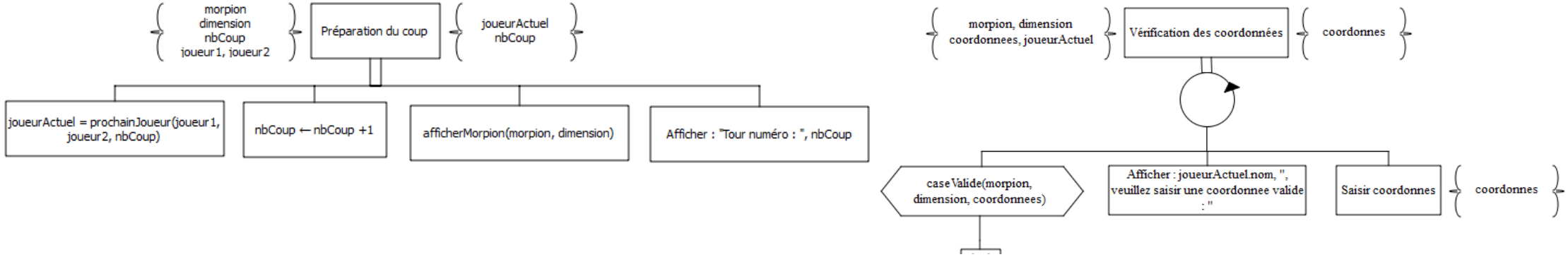
3.8.2 Stratégie de l'algorithme mise en place

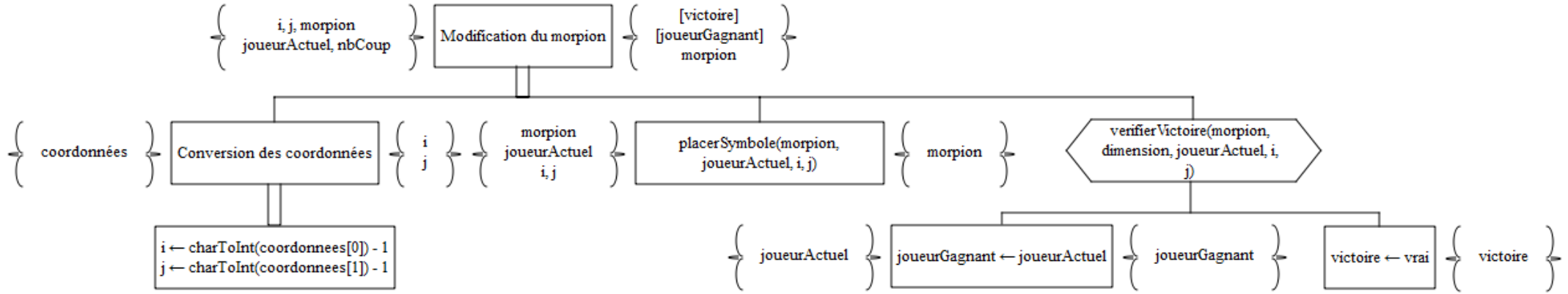
Après avoir préparé correctement le morpion, nous incrémentons de 1 la variable nbCoup (afin d'avoir un nombre de tours affiché à l'écran correct). On demande ensuite au joueur de saisir les coordonnées, en vérifiant alors s'il souhaite abandonner ('a' ou 'A', sans utiliser la méthode *toupper*). S'il souhaite effectivement abandonner, victoire sera passé à vrai, et donc on ne vérifie pas la condition suivante. Sinon, on effectue alors un *saisie-vérif* des coordonnées, et on procède alors à la modification du morpion (*placerSymbole( ) + verifierVictoire( )*).

3.8.3 Algorithme



Les sous-problèmes « Préparation du coup », « Vérification des coordonnées » et « Modification du morpion » sont illustrés ci-dessous





3.9 Sous-programme *placerSymbole( )*

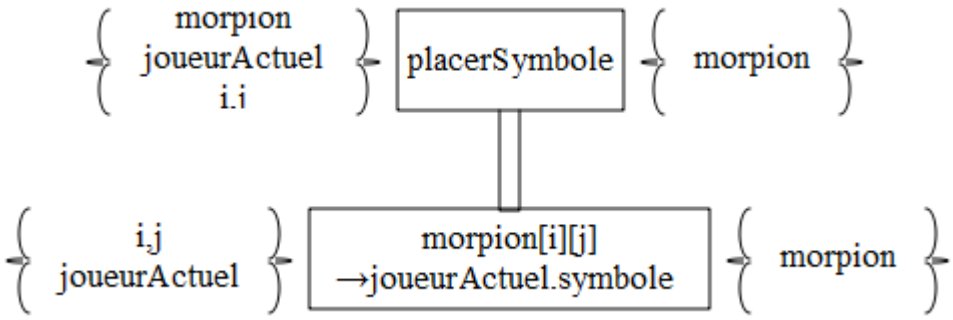
3.9.1 But de l'action

Place le symbole aux coordonnées (i, j) saisies précédemment par le joueurActuel. Il n'y a pas de saisie vérif car elle est déjà effectuée dans le programme *jouerCoup( )*.

3.9.2 Stratégie de l'algorithme mise en place

Simple modification du caractère aux coordonnées *morpion[i][j]*.

3.9.3 Algorithme



3.9.4 Dictionnaire des données associé

Nom	Type	Signification
Morpion	Matrice de caractère	Support de jeu, permet d'enregistrer les coordonnées saisies par les joueurs
i	Entier non signé	Indice de ligne du morpion
j	Entier non signé	Indice de colonne du morpion
joueurActuel	Joueur	Le joueur qui joue le tour actuel

3.10 Sous-programme *victoireLigne( )*

Ce sous-programme programme est très similaire aux sous-programmes déterminant une victoire à savoir *victoireColonne( ), victoireDiagonalePrincipale( )* et *victoireDiagonaleSecondaire( )*. Ils sont tous regroupés dans le sous-programme *verifierVictoire( )*.

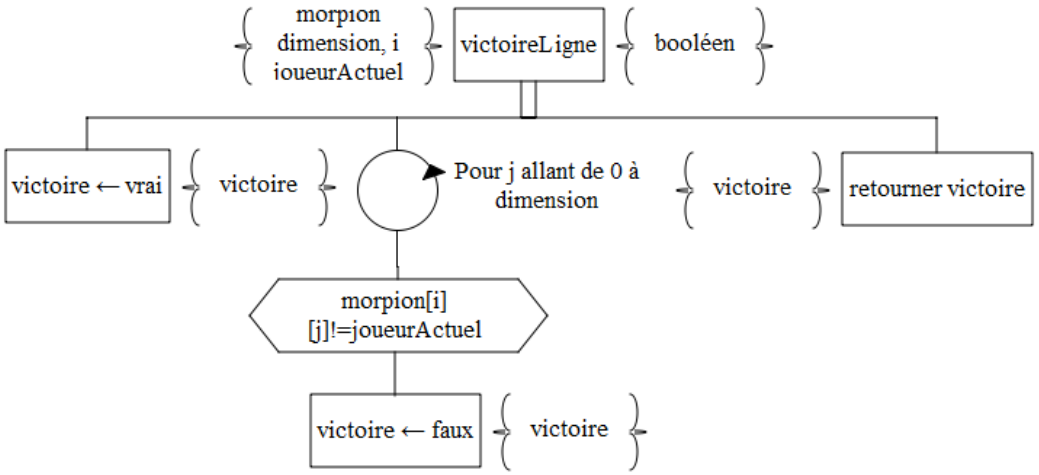
3.10.1 But de l'action

Fonction de type booléen qui détermine à partir d'une ligne i saisie par le joueurActuel si cette ligne est complétée par le symbole de ce même joueur. Si c'est le cas, il a gagné, sinon, la partie continue.

3.10.2 Stratégie de l'algorithme mise en œuvre

Parcours complet avec traitement conditionnel sur un tableau de caractères morpion à accès direct. Le traitement qui est effectué est : établir la victoire à faux, car la condition qui le précède peut se traduire comme « si le symbole présent dans la position courante ne correspond pas au symbole qui vient d'être joué ».

3.10.3 Algorithme



3.10.4 Dictionnaire de données associé

Nom	Type	Signification
Morpion	Matrice de caractère	Support de jeu, permet d'enregistrer les coordonnées saisies par les joueurs
dimension	Entier non signé	Dimension jouable du morpion
i	Entier non signé	Indice de ligne du morpion
j	Entier non signé	Indice de colonne du morpion
joueurActuel	Joueur	Le joueur qui joue le tour actuel
victoire	Booléen	Indique si une victoire est détectée dans les conditions

3.11 Sous-programme *victoireColonne( )*

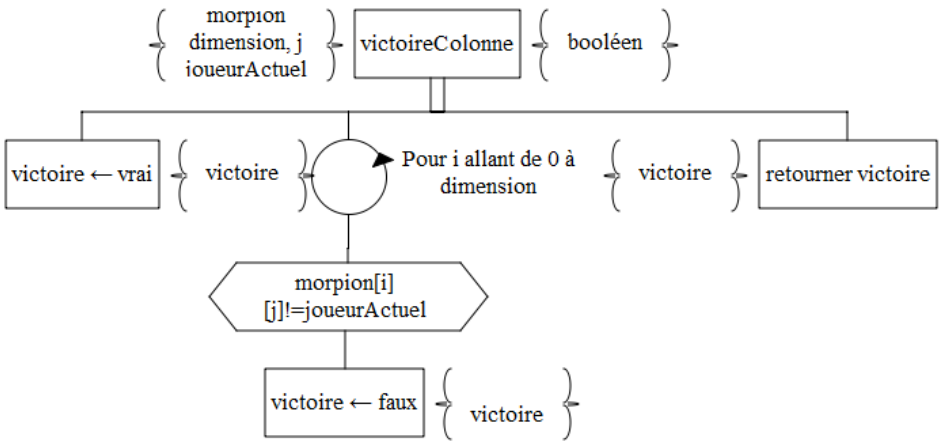
3.11.1 But de l'action

Fonction de type booléen qui détermine à partir d'une colonne j saisie par le joueurActuel si cette colonne est complétée par le symbole de ce même joueur. Si c'est le cas, il a gagné, sinon, la partie continue.

3.11.2 Stratégie de l'algorithme mise en œuvre

Parcours complet avec traitement conditionnel sur un tableau de caractères morpion à accès direct. Le traitement qui est effectué est : établir la victoire à faux, car la condition qui le précède peut se traduire comme « si le symbole présent dans la position courante ne correspond pas au symbole qui vient d'être joué ».

3.11.3 Algorithme



3.11.4 Dictionnaire de données associé

Nom	Type	Signification
Morpion	Matrice de caractère	Support de jeu, permet d'enregistrer les coordonnées saisies par les joueurs
dimension	Entier non signé	Dimension jouable du morpion
i	Entier non signé	Indice de ligne du morpion
j	Entier non signé	Indice de colonne du morpion
joueurActuel	Joueur	Le joueur qui joue le tour actuel
victoire	Booléen	Indique si une victoire est détectée dans les conditions

3.12 Sous-programme *victoireDiagonalePrincipale( )*

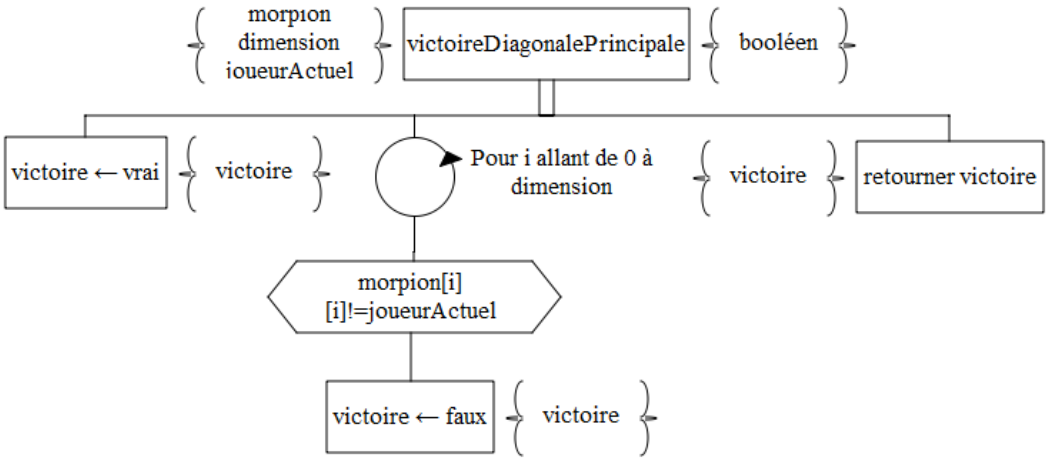
3.12.1 But de l'action

Fonction de type booléen qui détermine si la diagonale principale (haut gauche → bas droit) est complétée par le symbole du joueurActuel (qui vient de jouer). Si c'est le cas, il a gagné, sinon, la partie continue.

3.12.2 Stratégie de l'algorithme mise en œuvre

Parcours complet avec traitement conditionnel sur un tableau de caractères morpion à accès direct. Le traitement qui est effectué est : établir la victoire à faux, car la condition qui le précède peut se traduire comme « si le symbole présent dans la position courante ne correspond pas au symbole qui vient d'être joué ».

3.12.3 Algorithme



3.12.4 Dictionnaire de données associé

Nom	Type	Signification
Morpion	Matrice de caractère	Support de jeu, permet d'enregistrer les coordonnées saisies par les joueurs
dimension	Entier non signé	Dimension jouable du morpion
i	Entier non signé	Indice de ligne & colonne du morpion
joueurActuel	Joueur	Le joueur qui joue le tour actuel
victoire	Booléen	Indique si une victoire est détectée dans les conditions

3.13 Sous-programme *victoireDiagonaleSecondaire( )*

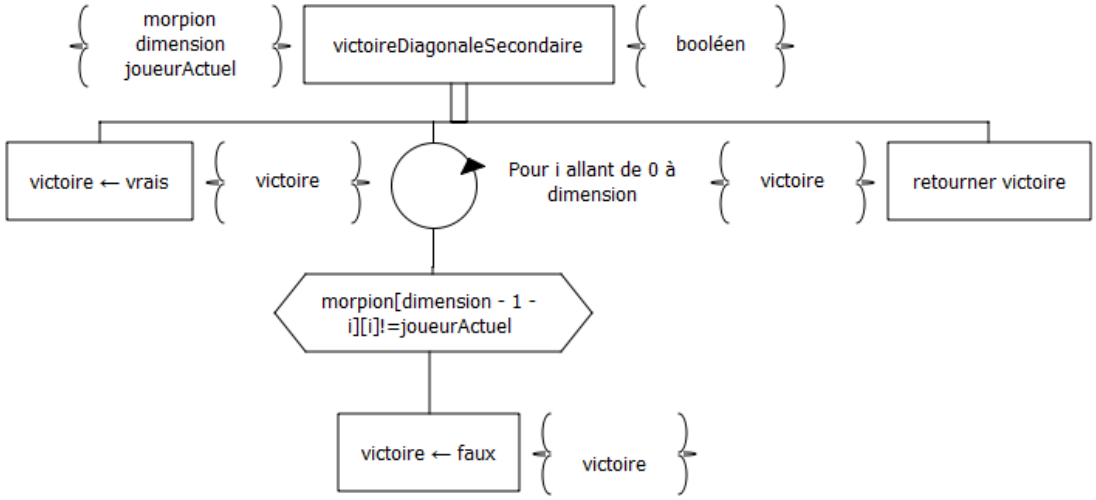
3.13.1 But de l'action

Fonction de type booléen qui détermine si la diagonale secondaire (haut droit → basgauche) est complétée par le symbole du joueurActuel (qui vient de jouer). Si c'est le cas, il a gagné, sinon, la partie continue.

3.13.2 Stratégie de l'algorithme mise en œuvre

Parcours complet avec traitement conditionnel sur un tableau de caractères morpion à accès direct. Le traitement qui est effectué est : établir la victoire à faux, car la condition qui le précède peut se traduire comme « si le symbole présent dans la position courante ne correspond pas au symbole qui vient d'être joué ».

3.13.3 Algorithme



3.13.4 Dictionnaire de données associé

Nom	Type	Signification
Morpion	Matrice de caractère	Support de jeu, permet d'enregistrer les coordonnées saisies par les joueurs
dimension	Entier non signé	Dimension jouable du morpion
i	Entier non signé	Indice de ligne & colonne du morpion
joueurActuel	Joueur	Le joueur qui joue le tour actuel
victoire	Booléen	Indique si une victoire est détectée dans les conditions

3.14 Sous-programme *verifierVictoire( )*

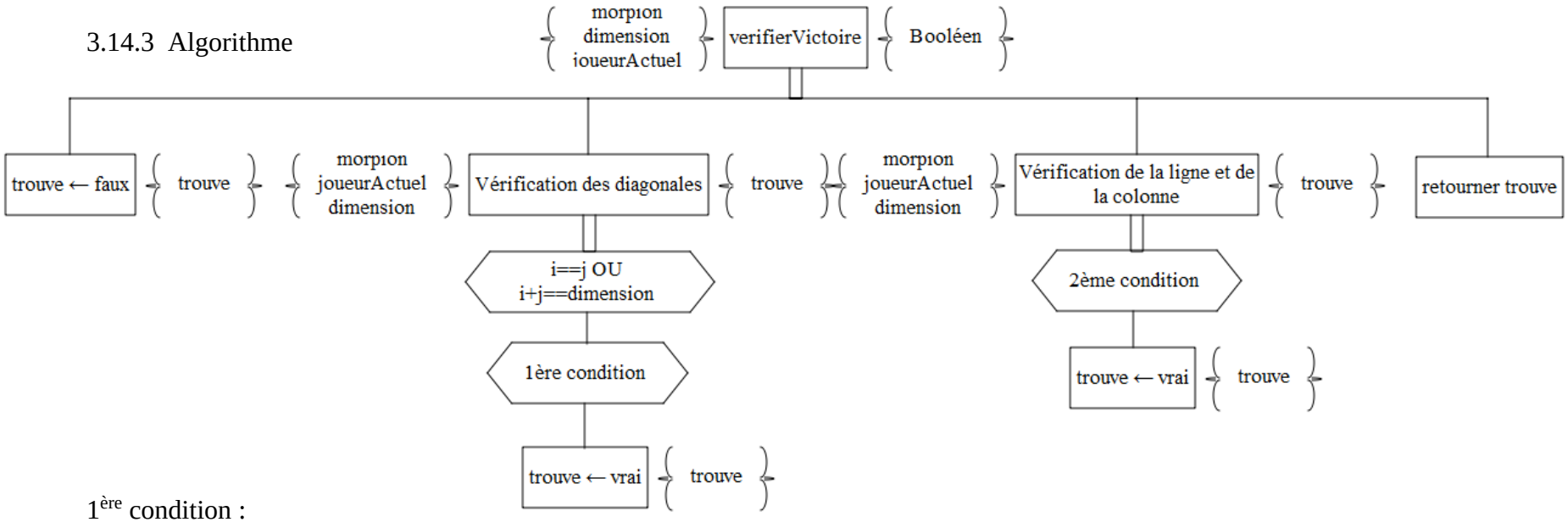
3.14.1 But de l'action

Cette fonction booléenne combine toutes les fonctions du même types énoncés au-dessus (cf. 3.13, 3.12, 3.11, 3.10), pour permettre à l'algorithme *jouerCoup( )* de vérifier s'il y a une victoire à chaque coup.

3.14.2 Stratégie de l'algorithme mise en œuvre

On considère d'abord que la victoire est fausse (la variable booléenne trouve peut paraître légèrement ambigüe, mais elle est totalement similaire à victoire, mais à une portée différente. Ainsi, pour éviter les confusions, nous avons décidé de la nommer autrement.). Ensuite, on vérifie les diagonales à l'aide des sous-programmes de vérification, puis on vérifie les lignes et colonnes. Notons si la victoire est détectée dans une diagonale, l'algorithme ne vérifie pas si la ligne ou colonne vérifie une ligne.

3.14.3 Algorithme



1ère condition :
victoireDiagonalePrincipale(morpion, DIMENSION, joueurActuel) OU victoireDiagonaleSecondaire(morpion, DIMENSION, joueurActuel)

2ème condition : victoireLigne(morpion, DIMENSION, i, joueurActuel) OU victoireColonne(morpion, DIMENSION, j, joueurActuel)) ET !trouve)

3.14.4 Dictionnaire de données associé

Morpion	Matrice de caractère	Support de jeu, permet d'enregistrer les coordonnées saisies par les joueurs
dimension	Entier non signé	Dimension jouable du morpion
i	Entier non signé	Indice de ligne du morpion
j	Entier non signé	Indice de colonne du morpion
nbcoup	Entier non signé	Nombre de coup joué pendant la partie
joueurActuel	Joueur	Le joueur qui joue le tour actuel
trouve	Booléen	Indique si une victoire est détectée dans les conditions

3.15 Sous-programme *caseValide( )*

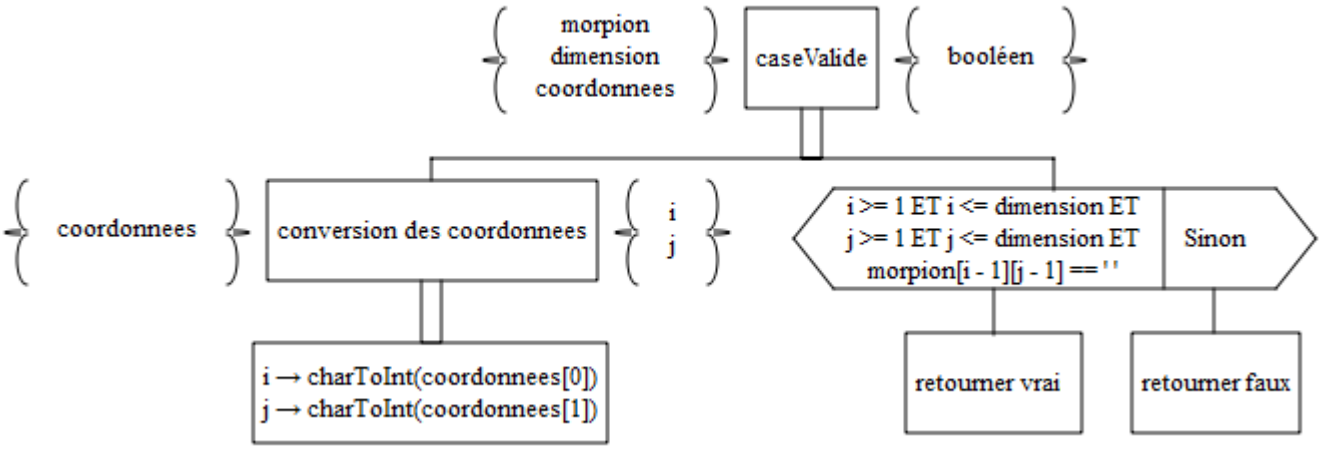
3.15.1 But de l'action

Retourne vrai si la case proposée par le joueur est valide (n'est pas déjà prise par le symbole d'un joueur et se trouve dans la zone jouable de taille dimension).

3.15.2 Stratégie de l'algorithme mise en place

On convertit d'abord grâce à une décomposition séquentielle, puis la condition vérifie si la case se situe dans l'intervalle et si la case n'a jamais été jouée par un autre joueur (caractère « espace »). Si c'est le cas, la fonction retourne vrai, sinon elle retourne faux.

3.15.3 Algorithme



3.15.4 Dictionnaire de données associé

Nom	Type	Signification
Morpion	Matrice de caractère	Support de jeu, permet d'enregistrer les coordonnées saisies par les joueurs
dimension	Entier non signé	Dimension jouable du morpion
i	Entier non signé	Indice de ligne du morpion
j	Entier non signé	Indice de colonne du morpion
coordonnées	Chaine de caractère	Coordonnées saisie par le joueur

3.16 Sous-programme *charToInt( )*

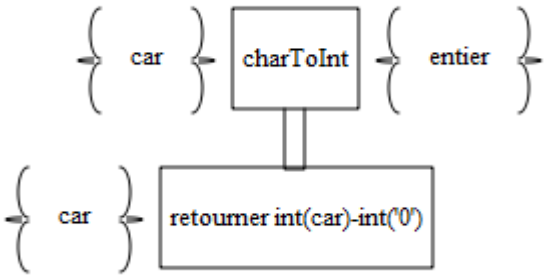
3.16.1 But de l'action

Convertir un chiffre saisi sous forme de caractère type entier.

3.16.2 Stratégie de l'algorithme mise en œuvre

Il s'agit ici d'un simple calcul : soustraction du code ASCII du caractère passé en paramètres par le code ASCII du caractère « 0 ». On se retrouve alors avec un entier.

3.16.3 Algorithme



3.16.4 Dictionnaire de données associé

nom	type	signification
car	caractère	Caractère de la coordonnée saisie par le joueur

4 Traces d’exécution

4.1 Scénario nominal ; un joueur gagne

```
+---+---+---+
|   |   |   |
+---+---+---+
|   |   |   |
+---+---+---+
|   |   |   |
+---+---+---+

Tour numero : 1
Alexandre, entrez les coordonnees de la case que vous voulez jouer ou 'a' pour abandonner : 11
```

```
+---+---+---+
| O |   |   |
+---+---+---+
|   |   |   |
+---+---+---+
|   |   |   |
+---+---+---+

Tour numero : 2
Nicolas, entrez les coordonnees de la case que vous voulez jouer ou 'a' pour abandonner : 12
```

```
+---+---+---+
| O | X |   |
+---+---+---+
|   |   |   |
+---+---+---+
|   |   |   |
+---+---+---+

Tour numero : 3
Alexandre, entrez les coordonnees de la case que vous voulez jouer ou 'a' pour abandonner : 22
```

```
+---+---+---+
| O | X |   |
+---+---+---+
|   | O |   |
+---+---+---+
|   |   |   |
+---+---+---+

Tour numero : 4
Nicolas, entrez les coordonnees de la case que vous voulez jouer ou 'a' pour abandonner : 32

+---+---+---+
| O | X |   |
+---+---+---+
|   | O |   |
+---+---+---+
|   | X |   |
+---+---+---+

Tour numero : 5
Alexandre, entrez les coordonnees de la case que vous voulez jouer ou 'a' pour abandonner : 33
```

```
+---+---+---+
| O | X |   |
+---+---+---+
|   | O |   |
+---+---+---+
|   | X | O |
+---+---+---+

Tour numero : 5

Alexandre a gagne, bravo !
```

4.2 Scénario alternatif 1 : un joueur abandonne

```
+---+---+---+
|   |   |   |
+---+---+---+
|   |   |   |
+---+---+---+
|   |   |   |
+---+---+---+

Tour numero : 1
Alexandre, entrez les coordonnees de la case que vous voulez jouer ou 'a' pour abandonner : 11
```

```
+---+---+---+
| O |   |   |
+---+---+---+
|   |   |   |
+---+---+---+
|   |   |   |
+---+---+---+

Tour numero : 2
Nicolas, entrez les coordonnees de la case que vous voulez jouer ou 'a' pour abandonner : a
```

```
+---+---+---+
| O |   |   |
+---+---+---+
|   |   |   |
+---+---+---+
|   |   |   |
+---+---+---+

Tour numero : 2

Alexandre a gagne, bravo !
```

4.3 Scénario alternatif 2 : grille complète et pas de gagnant à la fin de la partie

```
+---+---+---+
|   |   |   |
+---+---+---+
|   |   |   |
+---+---+---+
|   |   |   |
+---+---+---+

Tour numero : 1
Alexandre, entrez les coordonnees de la case que vous voulez jouer ou 'a' pour abandonner : 11
```

```
+---+---+---+
| O |   |   |
+---+---+---+
|   |   |   |
+---+---+---+
|   |   |   |
+---+---+---+

Tour numero : 2
Nicolas, entrez les coordonnees de la case que vous voulez jouer ou 'a' pour abandonner : 21
```

```
+---+---+---+
| O |   |   |
+---+---+---+
| X |   |   |
+---+---+---+
|   |   |   |
+---+---+---+

Tour numero : 3
Alexandre, entrez les coordonnees de la case que vous voulez jouer ou 'a' pour abandonner : 31
```

```
+---+---+---+
| O |   |   |
+---+---+---+
| X |   |   |
+---+---+---+
| O |   |   |
+---+---+---+

Tour numero : 4
Nicolas, entrez les coordonnees de la case que vous voulez jouer ou 'a' pour abandonner : 22
```

```
+---+---+---+
| O |   |   |
+---+---+---+
| X | X |   |
+---+---+---+
| O |   |   |
+---+---+---+

Tour numero : 5
Alexandre, entrez les coordonnees de la case que vous voulez jouer ou 'a' pour abandonner : 23
```

```
+---+---+---+
| O |   |   |
+---+---+---+
| X | X | O |
+---+---+---+
| O |   |   |
+---+---+---+

Tour numero : 6
Nicolas, entrez les coordonnees de la case que vous voulez jouer ou 'a' pour abandonner : 12
```

```
+---+---+---+
| O | X |   |
+---+---+---+
| X | X | O |
+---+---+---+
| O |   |   |
+---+---+---+

Tour numero : 7
Alexandre, entrez les coordonnees de la case que vous voulez jouer ou 'a' pour abandonner :
13
```

```
+---+---+---+
| O | X | O |
+---+---+---+
| X | X | O |
+---+---+---+
| O |   |   |
+---+---+---+

Tour numero : 8
Nicolas, entrez les coordonnees de la case que vous voulez jouer ou 'a' pour abandonner : 33
```

```
+---+---+---+
| O | X | O |
+---+---+---+
| X | X | O |
+---+---+---+
| O |   | X |
+---+---+---+

Tour numero : 9
Alexandre, entrez les coordonnees de la case que vous voulez jouer ou 'a' pour abandonner : 32
```

```
+---+---+---+
| O | X | O |
+---+---+---+
| X | X | O |
+---+---+---+
| O | O | X |
+---+---+---+

Tour numero : 9
Il y a egalite, dommage... :(
```

4.4 Scénario alternatif 3 : un joueur se trompe dans la saisie

```
+---+---+---+
|   |   |   |
+---+---+---+
|   |   |   |
+---+---+---+
|   |   |   |
+---+---+---+

Tour numero : 1
Alexandre, entrez les coordonnees de la case que vous voulez jouer ou 'a' pour abandonner : 14
Alexandre, veuillez saisir une coordonnee valide : 94
Alexandre, veuillez saisir une coordonnee valide : 80
Alexandre, veuillez saisir une coordonnee valide : 55
```

4.5 Extension traitée : la dimension du jeu est modifiable

```
----------Phase de Personnalisation----------
Saisissez la dimension du morpion : 15
La dimension doit etre comprise entre 3 et 9 :2
La dimension doit etre comprise entre 3 et 9 :6
Joueur 1, saississez votre nom :
```

4.6 Extension traitée : le joueur2 saisit un nom et un symbole similaire au joueur1

```
----------Phase de Personnalisation----------
Saisissez la dimension du morpion : 3
Joueur 1, saississez votre nom : Alexandre
Alexandre, entrez votre symbole : O

Joueur 2, saississez votre nom : Alexandre
Ce nom est deja utilise, veuillez en choisir un autre... :
```

```
----------Phase de Personnalisation----------
Saisissez la dimension du morpion : 3
Joueur 1, saississez votre nom : Alexandre
Alexandre, entrez votre symbole : O

Joueur 2, saississez votre nom : Alexandre
Ce nom est deja utilise, veuillez en choisir un autre... : Nicolas
Alexandre, entrez votre symbole : O

Ce symbole n'est pas utilisable veuillez en choisir un autre... :
```

5 Remarques concernant la S.A.É.

Nous avons eu beaucoup de difficultés pour terminer le travail par rapport aux pannes des bureaux virtuels qui nous ont beaucoup impactées dans notre travail.

6 Code C++

Le fichier main.cpp est livré dans l’archive sans la bibliothèque game-tools nécessaire à son fonctionnement. Veuillez déposer la bibliothèque dans le même dossier que le *main.cpp*.