

**Samuel HENTRICS LOISTINE**

**Ahmed FAKHFAKH**



# **S1.02 - Comparaison d'approches algorithmiques Sujet n°1 : Morpion PACK 3**

# Sommaire

<b>1. Équipe</b>	<b>3</b>
<b>2. Exercice traité : PACK 3</b>	<b>3</b>
<b>3. Rappel des spécifications du programme</b>	<b>3</b>
3.1. Spécifications initiales	3
3.2. Spécifications complémentaires = extensions traitées	10
<b>4. Algorithmes du programme (action principale et ses sous-actions)</b>	<b>11</b>
4.1 Initialiser la partie	12
4.1.1 But de l'action	12
4.1.2 Stratégie de l'algorithme mise en oeuvre	12
4.1.3 Algorithme	12
4.1.4 Dictionnaire des éléments associés à cet algorithme	14
4.2 Jouer la partie	15
4.2.1 But de l'action	15
4.2.2 Stratégie de l'algorithme mise en oeuvre	15
4.2.3 Algorithme	16
4.2.4 Dictionnaire des éléments associés à cet algorithme	22
4.3 Finaliser la partie	23
4.3.1 But de l'action	23
4.3.2 Stratégie de l'algorithme mise en oeuvre	23
4.3.3 Algorithme	23
4.3.4 Dictionnaire des éléments associés à cet algorithme	25
4.4 Tous les sous-programmes	26
4.4.A Pour la grille	26
4.4.A.1. initialiserGrille	26
4.4.A.2. afficherGrille	27
4.4.A.3. insererCoordonnee	29
4.4.B Pour le jeu	30
4.4.B.1. saisirJoueur	30
4.4.B.2. saisirJoueur (variante)	31
4.4.B.3. coordonneeCorrecte	33
4.4.B.4. caseVide	34
4.4.B.5. verifierLigne	35
4.4.B.6. verifierColonne	38
4.4.B.7. verifierDiagonaleGaucheDroite	41
4.4.B.8. verifierDiagonaleDroiteGauche	44
4.4.B.9. afficherRegles	47
4.4.B.10. personnalisation	48

4.4.B.11. determinerJoueur	51
4.4.B.12. saisirCoordonnee	53
4.4.B.13. resultat	56
4.4.B.14. verifierGagnant	59
<b>5. Traces d'exécution</b>	<b>62</b>
<b>6. Remarques</b>	<b>71</b>
<b>7. Code C++</b>	<b>72</b>

# 1. Équipe

HENTRICS LOISTINE Samuel TD II - TP3

FAKHFAKH Ahmed TD II – TP3

## 2. Exercice traité : PACK 3

Sujet n°1 : Le jeu du morpion

## 3. Rappel des spécifications du programme

### 3.1. Spécifications initiales

Description du jeu (reprise du sujet) :

Jeu qui se joue à deux joueurs nécessitant une grille de 9 cases (3 lignes et 3 colonnes).

Chaque joueur joue à tour de rôle.

Le but du jeu est d'aligner, avant son adversaire, 3 symboles identiques, horizontalement, verticalement ou en diagonale.

La partie se termine :

- Quand l'un des joueurs a aligné ses 3 symboles, et il gagne,
- Quand la grille est complète, et il y a égalité (pas de gagnant),
- Quand l'un des joueurs abandonne, et c'est alors l'autre joueur qui gagne.

Spécifications externes

- Phase initiale de personnalisation (commune à tous les scénarios)

```

----- M O R P I O N -----

Jeu avec 2 joueurs humains et une grille de 9 cases : 3 lignes et 3 colonnes.
Chaque joueur joue a tour de role.
Le but du jeu est d'aligner, avant son adversaire, 3 symboles identiques, horizontalement, verticalement ou en diagonale.

La partie se termine :
- quand l'un des joueurs a aligne ses 3 symboles, et il gagne,
- quand la grille est complete, et il y a une egalite (pas de gagnant)
- quand l'un des joueurs abandonne, et c'est alors l'autre joueur qui gagne.
Chaque joueur a donc son propre symbole.

Dans cette version du Morpion, une phase prealable de personnalisation permet a chaque joueur de saisir son prenom et le symbole choisi pour jouer
Lorsqu'arrive son tour de joueur, un joueur peut :
- saisir la position de la case ou placer son symbole, sous la forme LigneColonne (exemple : 12 pour Ligne 1, Colonne 2)
- saisir A (ou a) pour abandonner le jeu

-----

--- Phase de personnalisation ---

JOUEUR 1, entrez votre prenom : Maia
Maia, entrez votre symbole (1 lettre ou 1 chiffre) : m

JOUEUR 2, entrez votre prenom, autre que 'Maia' : Ana
Ana, entrez votre symbole (1 lettre ou 1 chiffre), autre que 'm' : a

```

- Scénario d'exception : Phase initiale de personnalisation, le joueur 2 saisie un prénom et/ou symbole identique au joueur 1

```

----- MORPION -----

Jeu avec 2 joueurs humains et une grille de 9 cases : 3 lignes et 3 colonnes.
Chaque joueur joue a tour de role.
Le but du jeu est d'aligner, avant son adversaire, 3 symboles identiques, horizontalement, verticalement
ou en diagonale.

La partie se termine :
- quand l'un des joueurs a aligne ses 3 symboles, et il gagne,
- quand la grille est complete, et il y a une egalite (pas de gagnant)
- quand l'un des joueurs abandonne, et c'est alors l'autre joueur qui gagne.
Chaque joueur a donc son propre symbole.

Dans cette version du Morpion, une phase prealable de personnalisation permet a chaque joueur de saisir
son prenom et le symbole
choisi pour jouer
Lorsqu'arrive son tour de joueur, un joueur peut :
- saisir la position de la case ou placer son symbole, sous la forme LigneColonne (exemple : 12 pour
Ligne 1, Colonne 2)
- saisir A (ou a) pour abandonner le jeu

-----

--- Phase de personnalisation ---

JOUEUR 1, entrez votre prenom : Maia
Maia, entrez votre symbole (1 lettre ou 1 chiffre) : m

JOUEUR 2, entrez votre prenom, autre que 'Maia' : Ana
JOUEUR 2, entrez votre prenom, autre que 'Maia' : Ana
Ana, entrez votre symbole (1 lettre ou 1 chiffre), autre que 'm' : m
Ana, entrez votre symbole (1 lettre ou 1 chiffre), autre que 'm' : a

```

- Scénario nominal : Un joueur gagne

<p>Tour numero 1</p> <p>[     ] [     ] [     ]</p> <p>---&gt; Maia, entrez :</p> <p>Une coordonnee (LigneColonne, ex. : 12 pour Ligne 1, Colonne 2), ou bien A(bandonner).</p> <p>Votre choix ? 12</p>	<p>Tour numero 2</p> <p>[  m  ] [     ] [     ]</p> <p>---&gt; Ana, entrez :</p> <p>Une coordonnee (LigneColonne, ex. : 12 pour Ligne 1, Colonne 2), ou bien A(bandonner).</p> <p>Votre choix ? 31</p>
<p>Tour numero 3</p> <p>[  m  ] [     ] [a    ]</p> <p>---&gt; Maia, entrez :</p> <p>Une coordonnee (LigneColonne, ex. : 12 pour Ligne 1, Colonne 2), ou bien A(bandonner).</p> <p>Votre choix ? 11</p>	<p>Tour numero 4</p> <p>[m m  ] [     ] [     ]</p> <p>---&gt; Ana, entrez :</p> <p>Une coordonnee (LigneColonne, ex. : 12 pour Ligne 1, Colonne 2), ou bien A(bandonner).</p> <p>Votre choix ? 22</p>
<p>Tour numero 5</p> <p>[m m  ] [  a  ] [ a    ]</p> <p>---&gt; Maia, entrez :</p> <p>Une coordonnee (LigneColonne, ex. : 12 pour Ligne 1, Colonne 2), ou bien A(bandonner).</p> <p>Votre choix ? 13</p>	<p>Tour numero 5</p> <p>[m m m  ] [  a  ] [ a    ]</p> <p>--- Fin de la partie ---</p> <p>Partie gagnée par Maia au Tour numero 5</p>

- Scénario alternatif : Un joueur abandonne

<p>Tour numero 1</p> <pre> [     ] [     ] [     ] </pre> <p>---&gt; Maia, entrez :</p> <p>Une coordonnee (LigneColonne, ex. : 12 pour Ligne 1, Colonne 2), ou bien A(bandonner).</p> <p>Votre choix ? 12</p>	<p>Tour numero 2</p> <pre> [  m  ] [     ] [     ] </pre> <p>---&gt; Ana, entrez :</p> <p>Une coordonnee (LigneColonne, ex. : 12 pour Ligne 1, Colonne 2), ou bien A(bandonner).</p> <p>Votre choix ? 31</p>
<p>Tour numero 3</p> <pre> [  m  ] [     ] [a    ] </pre> <p>---&gt; Maia, entrez :</p> <p>Une coordonnee (LigneColonne, ex. : 12 pour Ligne 1, Colonne 2), ou bien A(bandonner).</p> <p>Votre choix ? a</p>	<p>Tour numero 3</p> <pre> [  m  ] [     ] [a    ] </pre> <p>--- Fin de la partie ---</p> <p>Partie gantee par Ana suite a l'abandon de Maia au Tour numero 3</p>

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



<p>Tour numero 1</p> <p>[     ] [     ] [     ]</p> <p>---&gt; Maia, entrez :</p> <p>Une coordonnee (LigneColonne, ex. : 12 pour Ligne 1, Colonne 2), ou bien A(bandonner).</p> <p>Votre choix ? 12</p>	<p>Tour numero 2</p> <p>[  m  ] [     ] [     ]</p> <p>---&gt; Ana, entrez :</p> <p>Une coordonnee (LigneColonne, ex. : 12 pour Ligne 1, Colonne 2), ou bien A(bandonner).</p> <p>Votre choix ? 31</p>
<p>Tour numero 3</p> <p>[  m  ] [     ] [a    ]</p> <p>---&gt; Maia, entrez :</p> <p>Une coordonnee (LigneColonne, ex. : 12 pour Ligne 1, Colonne 2), ou bien A(bandonner).</p> <p>Votre choix ? 22</p>	<p>Tour numero 4</p> <p>[  m  ] [  m  ] [a    ]</p> <p>---&gt; Ana, entrez :</p> <p>Une coordonnee (LigneColonne, ex. : 12 pour Ligne 1, Colonne 2), ou bien A(bandonner).</p> <p>Votre choix ? 32</p>
<p>Tour numero 5</p> <p>[  m  ] [  m  ] [a a  ]</p> <p>---&gt; Maia, entrez :</p> <p>Une coordonnee (LigneColonne, ex. : 12 pour Ligne 1, Colonne 2), ou bien A(bandonner).</p> <p>Votre choix ? 33</p>	<p>Tour numero 6</p> <p>[  m  ] [  m  ] [a  a m]</p> <p>---&gt; Ana, entrez :</p> <p>Une coordonnee (LigneColonne, ex. : 12 pour Ligne 1, Colonne 2), ou bien A(bandonner).</p> <p>Votre choix ? 11</p>
<p>Tour numero 7</p> <p>[a m  ] [  m  ] [a  a  m]</p> <p>---&gt; Maia, entrez :</p> <p>Une coordonnee (LigneColonne, ex. : 12 pour Ligne 1, Colonne 2), ou bien A(bandonner).</p> <p>Votre choix ? 21</p>	<p>Tour numero 8</p> <p>[a m  ] [m m  ] [a a  m]</p> <p>---&gt; Ana, entrez :</p> <p>Une coordonnee (LigneColonne, ex. : 12 pour Ligne 1, Colonne 2), ou bien A(bandonner).</p> <p>Votre choix ? 23</p>

<p>Tour numero 9</p> <pre>[a m  ] [m m  ] [a  a  m]</pre> <p>---&gt; Maia, entrez :</p> <p>Une coordonnee (LigneColonne, ex. : 12 pour Ligne 1, Colonne 2), ou bien A(bandonner).</p> <p>Votre choix ? 13</p>	<p>Tour numero 9</p> <pre>[a m  ] [m m a] [a a  m]</pre> <p>--- Fin de la partie ---</p> <p>Egalite (pas de gagnant) car grille complete</p>
---	--

- Scénario d'exception : Le joueur saisit une coordonnée déjà utilisée ou une coordonnée qui n'existe pas

<p>Tour numero 1</p> <pre>[     ] [     ] [     ]</pre> <p>---&gt; Maia, entrez :</p> <p>Une coordonnee (LigneColonne, ex. : 12 pour Ligne 1, Colonne 2), ou bien A(bandonner).</p> <p>Votre choix ? 12</p>	<p>Tour numero 2</p> <pre>[  m  ] [     ] [     ]</pre> <p>---&gt; Ana, entrez :</p> <p>Une coordonnee (LigneColonne, ex. : 12 pour Ligne 1, Colonne 2), ou bien A(bandonner).</p> <p>Votre choix ? 31</p>
<p>Tour numero 3</p> <pre>[  m  ] [     ] [a    ]</pre> <p>---&gt; Maia, entrez :</p> <p>Une coordonnee (LigneColonne, ex. : 12 pour Ligne 1, Colonne 2), ou bien A(bandonner).</p> <p>Votre choix ? 31 Votre choix ? 13</p>	<p>Tour numero 3</p> <pre>[  m  ] [     ] [a    ]</pre> <p>---&gt; Maia, entrez :</p> <p>Une coordonnee (LigneColonne, ex. : 12 pour Ligne 1, Colonne 2), ou bien A(bandonner).</p> <p>Votre choix ? 45 Votre choix ? 21</p>

- Propriété(s)
  - Si le joueur saisit une coordonnée commençant par A puis des caractères qui se suivent, on considère que le joueur souhaite abandonner.
  - Si le joueur insère une coordonnée avec plus de 2 informations (ligne/colonne), seul les deux premiers caractères seront pris en compte.
  - Si le joueur insère une coordonnée avec qu'une information (ligne), le joueur devra ressaisir une coordonnée.

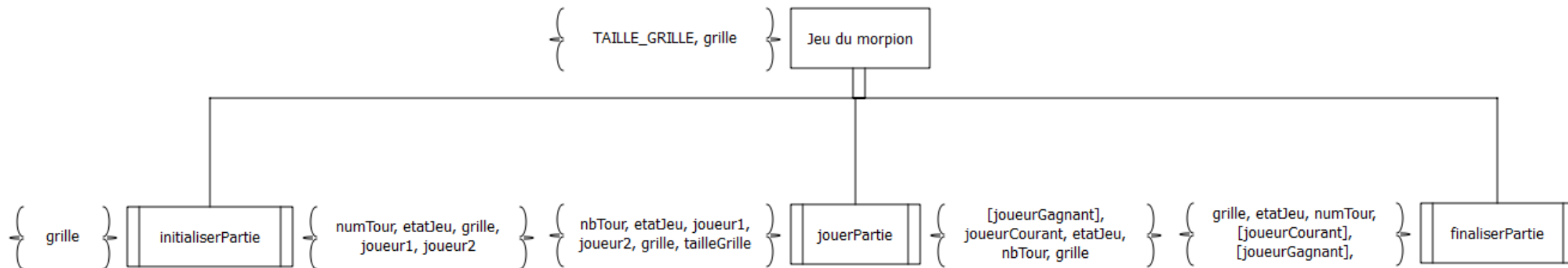
- Un joueur ne pourra que gagner au bout du 5ème tour (en effet,  $3 \times 2 - 1 = 5$  tours).

### **3.2. Spécifications complémentaires = extensions traitées**

L'extension "*Le joueur n°2 est obligé de fournir un prénom et symbole différents de ceux du joueur n°1.*" a été traitée.

## 4. Algorithmes du programme (action principale et ses sous-actions)

Tous les sous-programmes utilisés dans Initialiser la partie, Jouer la partie ou même Finaliser la partie, se trouvent dans la partie 4.4



## **4.1 Initialiser la partie**

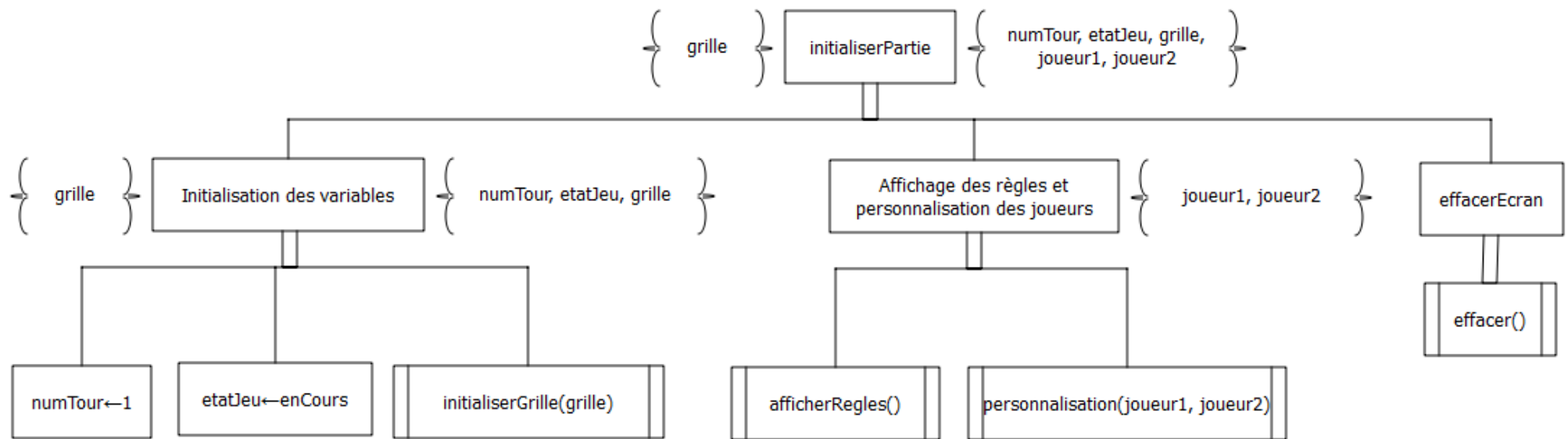
### **4.1.1 But de l'action**

Prépare la partie : le numéro de tour, l'état du jeu, la grille ainsi que les joueurs

### **4.1.2 Stratégie de l'algorithme mise en oeuvre**

On initialise le numéro du tour à 1, l'état du jeu à "enCours" puis on utilise le sous-programme inisialiserGrille() pour initialiser la grille, ensuite, on affiche les règles du jeu avec afficherRegles(), enfin, on demande à remplir les informations des deux joueurs grâce au sous-programme personnalisation() et enfin on efface l'écran avec effacer().

### **4.1.3 Algorithme**



#### 4.1.4 Dictionnaire des éléments associés à cet algorithme

Nom	Type	Signification
numTour	Entier positif	Numéro du tour actuel
etatJeu	UnEtat	Etat du jeu
grille	Tableau de caractère	Grille du jeu
joueur1	UnJoueur	Joueur 1 qui va jouer avec joueur 2
joueur2	UnJoueur	Joueur 2 qui va jouer avec joueur 1

## **4.2 Jouer la partie**

### **4.2.1 But de l'action**

Permet de jouer la partie entre les deux joueurs tant que :

- un des joueurs décide d'abandonner
- un des joueurs gagne la partie
- il y a égalité car le nombre de tours maximum est atteint

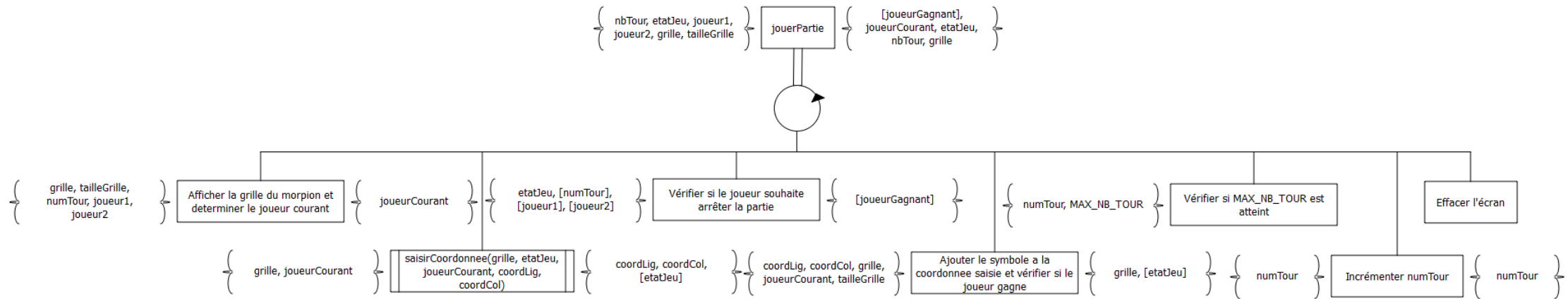
### **4.2.2 Stratégie de l'algorithme mise en oeuvre**

Dans une boucle itérative non fixe, on a mis en place cette stratégie :

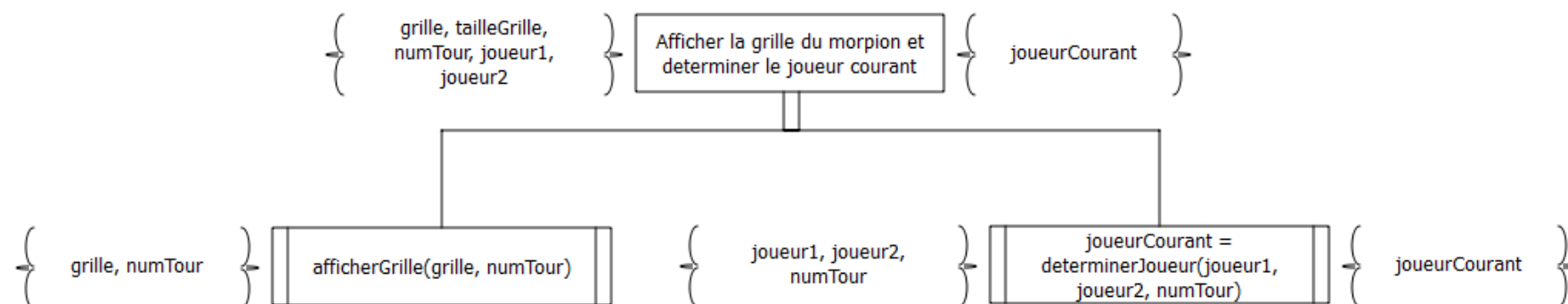
Tout d'abord, on affiche la grille avec `afficherGrille()`, ensuite, on va déterminer le joueur courant avec `determinerJoueur()`, ensuite, le joueur courant qui est déterminé va devoir saisir une coordonnée avec le sous-programme `saisirCoordonnee()`, si le joueur souhaite arrêter la partie, l'état du jeu passe à "abandon" et on détermine le joueur gagnant (sachant que c'est le prochain de ce tour), si le joueur n'arrête pas alors on insère sa coordonnée si elle est correcte (`saisirCoordonnee` vérifie que la coordonnée est correcte) avec `insererCoordonnee()`. Ensuite, le sous-programme `verifierGagnant()` va vérifier si le joueur courant gagne (sur une ligne/colonne ou une des deux diagonales), si c'est le cas, l'état du jeu passe donc à "gagnant" et on peut quitter la boucle sinon, on continue de jouer en incrémentant le nombre de tours tant que le numéro de tour est différent du nombre de tours maximum et on efface l'écran pour le prochain tour avec `effacer()`.



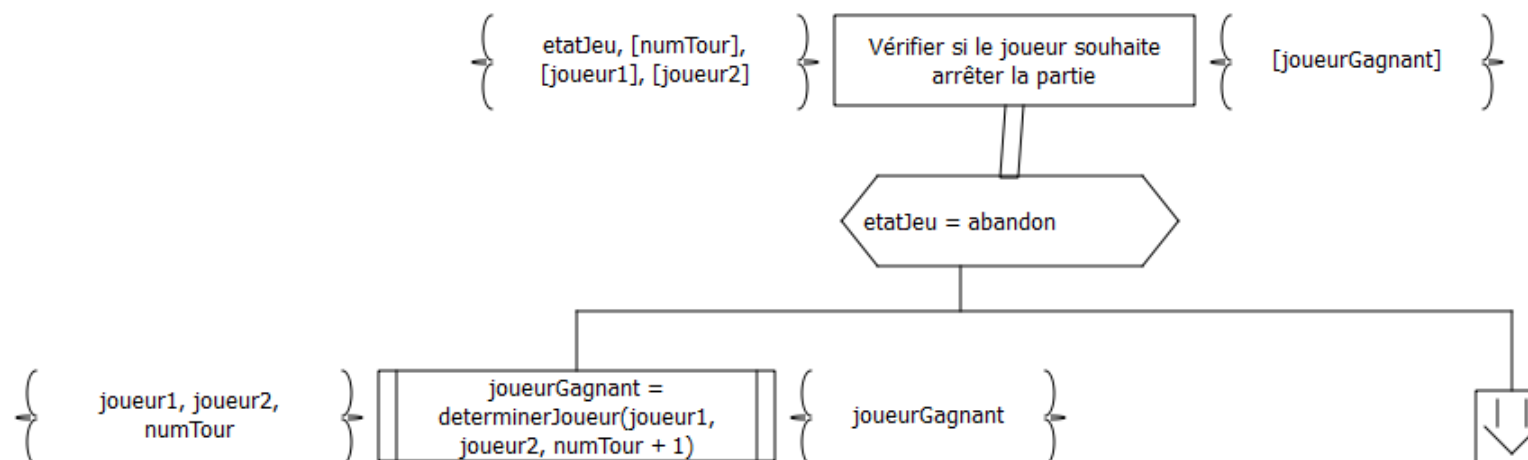
### 4.2.3 Algorithme



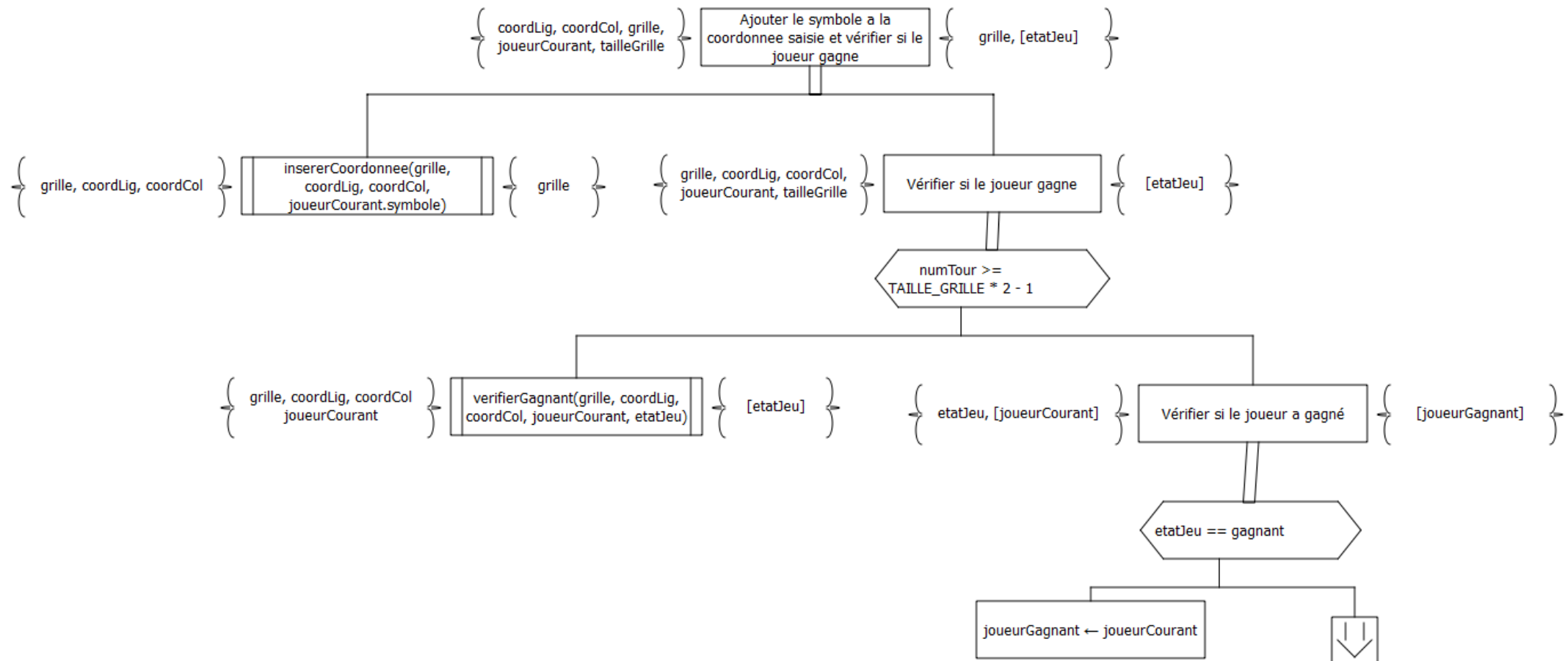
#### 4.2.3.1 - Afficher la grille du morpion et determiner le joueur courant



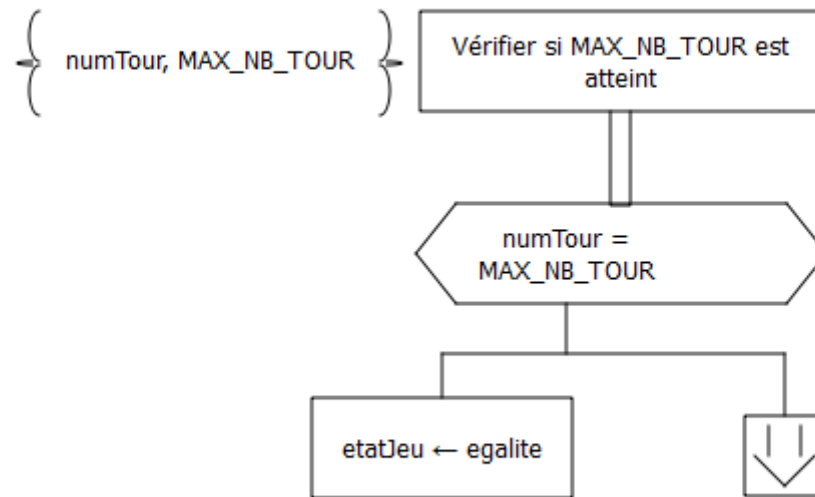
#### 4.2.3.2 - Vérifier si le joueur souhaite arrêter la partie



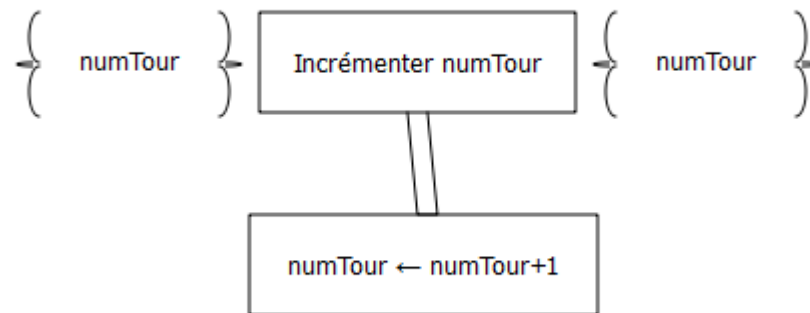
#### 4.2.3.3 - Ajouter le symbole a la coordonnee saisie et vérifier si le joueur gagne



#### 4.2.3.4 - Vérifier si MAX\_NB\_TOUR est atteint



#### 4.2.3.5 - Incréments numTour



#### 4.2.4 Dictionnaire des éléments associés à cet algorithme

Nom	Type	Signification
grille	tableau de caractère	Grille du jeu
numTour	Entier positif	Numéro du tour courant
MAX_NB_TOUR	Entier positif	Constante désignant le nombre maximum du tour qu'on peut jouer
coordLig	Entier positif	Coordonnée de ligne choisie par le joueur
coordCol	Entier positif	Coordonnée du colonne choisie par le joueur
joueur1	UnJoueur	Joueur 1 qui va jouer avec joueur 2
joueur2	UnJoueur	Joueur 2 qui va jouer avec joueur 1
joueurCourant	UnJoueur	Joueur qui joue actuellement
joueurGagnant	UnJoueur	Le joueur qui a gagné
etatJeu	UnEtat	Contient l'état actuel du jeu

## **4.3 Finaliser la partie**

### **4.3.1 But de l'action**

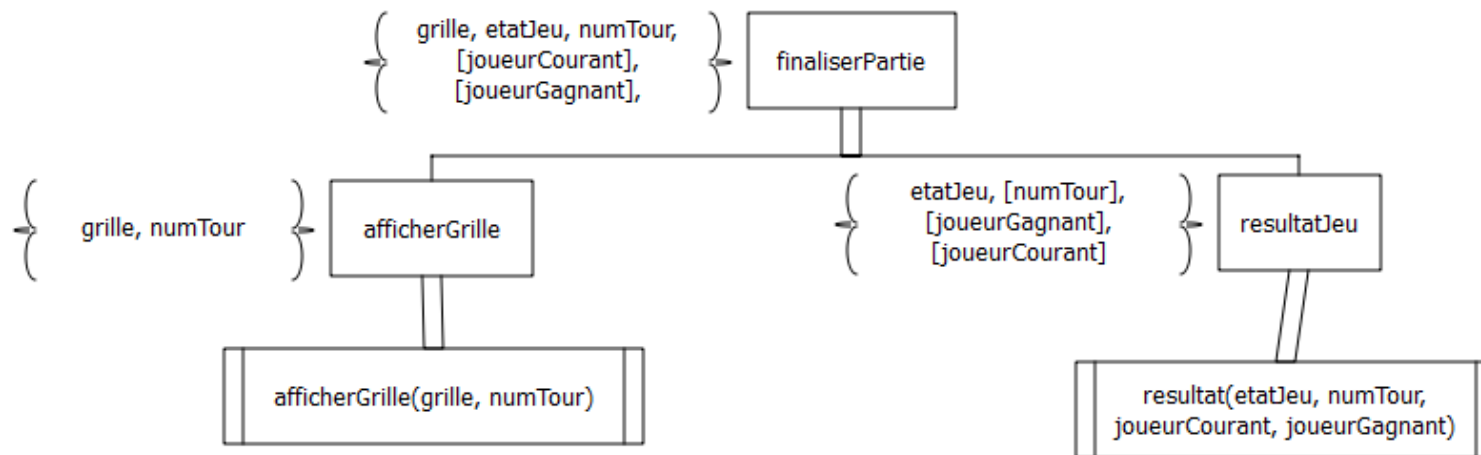
Permet de finaliser la partie en affichant la grille finale ainsi que le résultat de la partie.

### **4.3.2 Stratégie de l'algorithme mise en oeuvre**

On a utilisé les sous programmes `afficherGrille()` pour afficher la grille et `resultat()` pour afficher le résultat

### **4.3.3 Algorithme**





#### 4.3.4 Dictionnaire des éléments associés à cet algorithme

Nom	Type	Signification
grille	Tableau de caractère	Grille du jeu
numTour	Entier positif	Numéro du tour actuel
etatJeu	Entier positif	Contient l'état du jeu actuel
joueurCourant	UnJoueur	Joueur courant du tour actuel
joueurGagnant	UnJoueur	Le joueur qui a gagné

## 4.4 Tous les sous-programmes

### 4.4.A Pour la grille

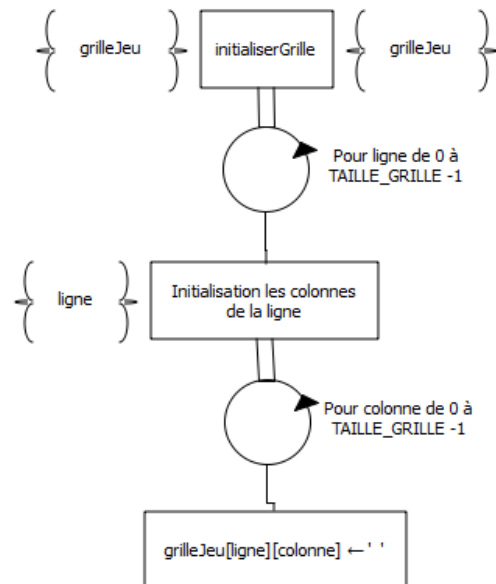
#### 4.4.A.1. initialiserGrille

But de l'action: À partir d'une grille donnée (grilleJeu), initialiser la grille en la remplissant d'espace.

Stratégie de l'algorithme mise en oeuvre: Parcours complet avec traitement systématique

Traitement : Pour la case courante, mettre un espace.

Algorithme :



#### Dictionnaire des données

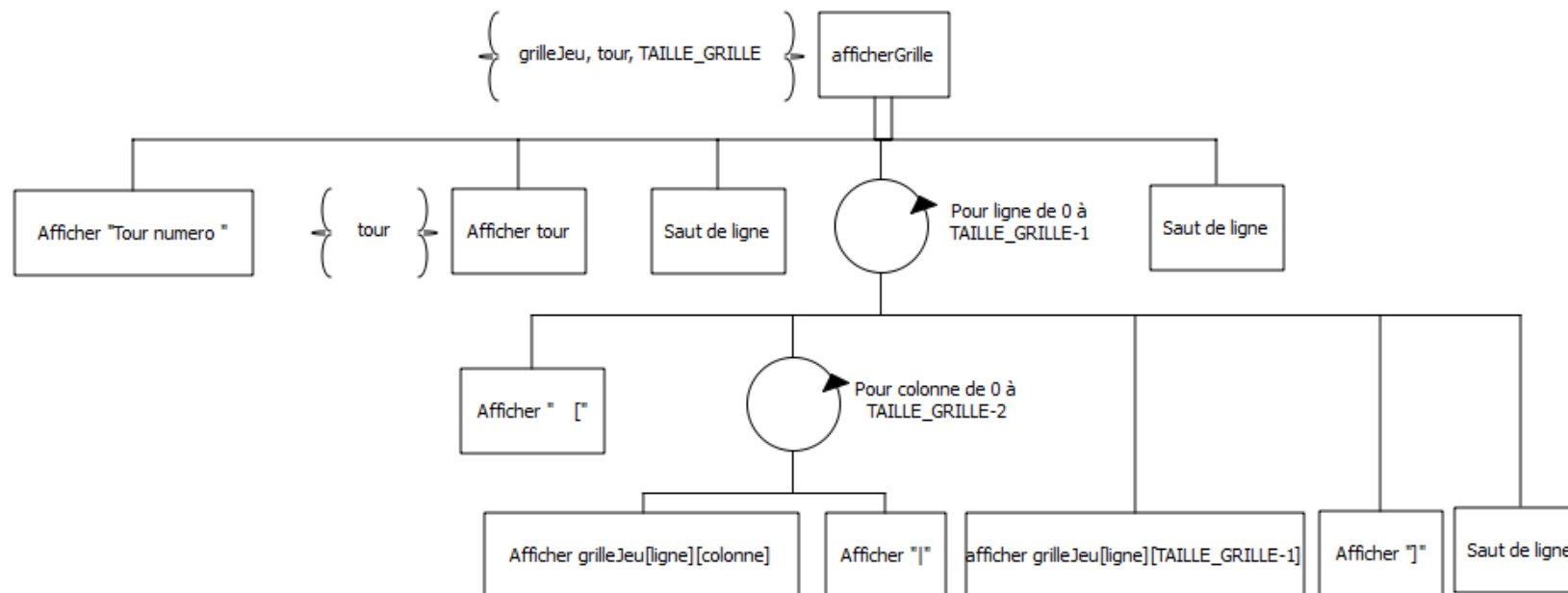
Nom	Type	Signification
grilleJeu	Tableau de caractère	Grille qui doit être initialisée
TAILLE_GRILLE	Entier positif	Constante représentant la taille de la grille
ligne	Entier positif	Ligne courante qui sera traitée
colonne	Entier positif	Colonne courante qui sera traitée

#### 4.4.A.2. afficherGrille

But de l'action: Afficher la grille du jeu ainsi que le tour actuel.

Stratégie de l'algorithme mise en oeuvre: Parcours complet avec traitement systématique.

Traitement : Afficher la case courante du tableau (ligne et colonne).



### Dictionnaire des données

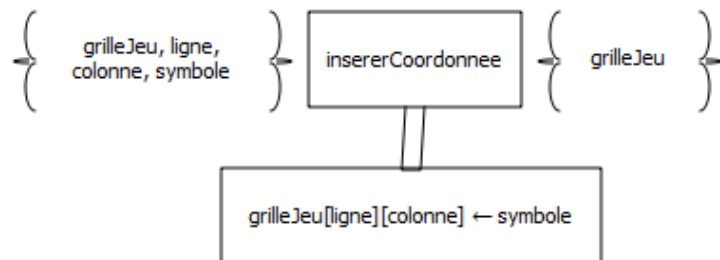
Nom	Type	Signification
grilleJeu	Tableau de caractères	Grille qui doit être affichée
TAILLE_GRILLE	Entier positif	Constante représentant la taille de la grille
tour	Entier positif	Numéro du tour actuel
ligne	Entier positif	Ligne courante qui sera traitée
colonne	Entier positif	Colonne courante qui sera traitée

#### 4.4.A.3. insererCoordonnee

But de l'action: Insérer un symbole dans la grille de jeu à partir d'une ligne et d'une colonne donnée.

Stratégie de l'algorithme mise en œuvre: On affecte dans la grille à la colonne et la ligne données, le symbole donné en paramètre.

Algorithme :



#### Dictionnaire des données

Nom	Type	Signification
grilleJeu	Tableau de caractère	Grille qui va recevoir selon une ligne et une colonne donnée un symbole
ligne	Entier positif	Numéro de la ligne qui va recevoir le symbole
colonne	Entier positif	Numéro de la colonne qui va recevoir le symbole
symbole	caractère	Symbole qui sera ajouté dans la grille

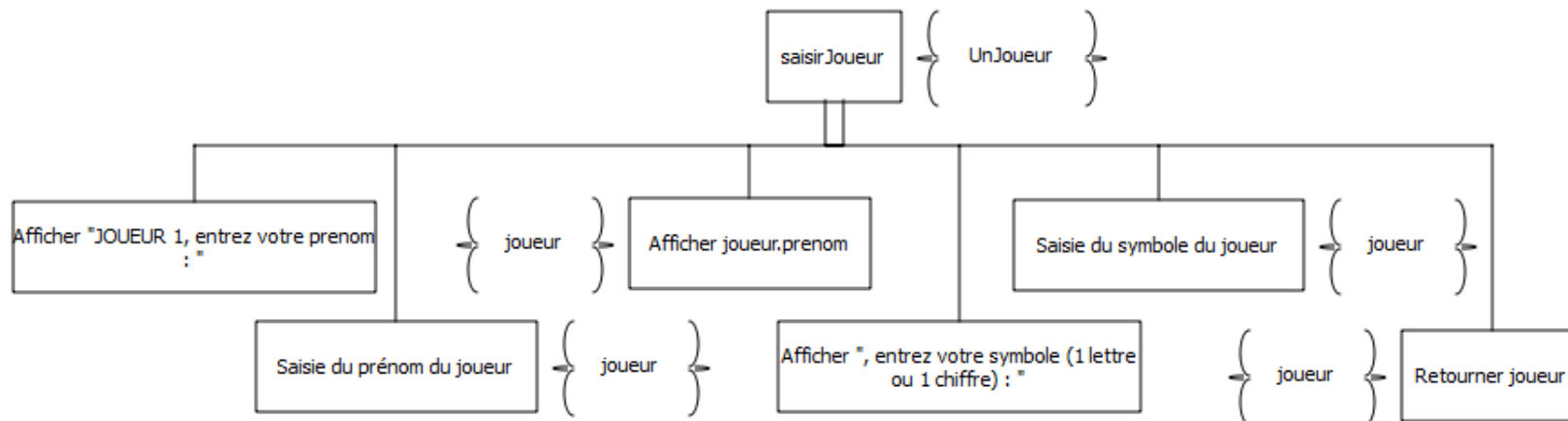
## 4.4.B Pour le jeu

### 4.4.B.1. saisirJoueur

But de l'action: Retourne un joueur avec un prénom et un symbole

Stratégie de l'algorithme mise en oeuvre: On demande au joueur de saisir son prénom puis son symbole, ensuite, la fonction retourne une variable de type UnJoueur.

Algorithme :



#### Dictionnaire des données

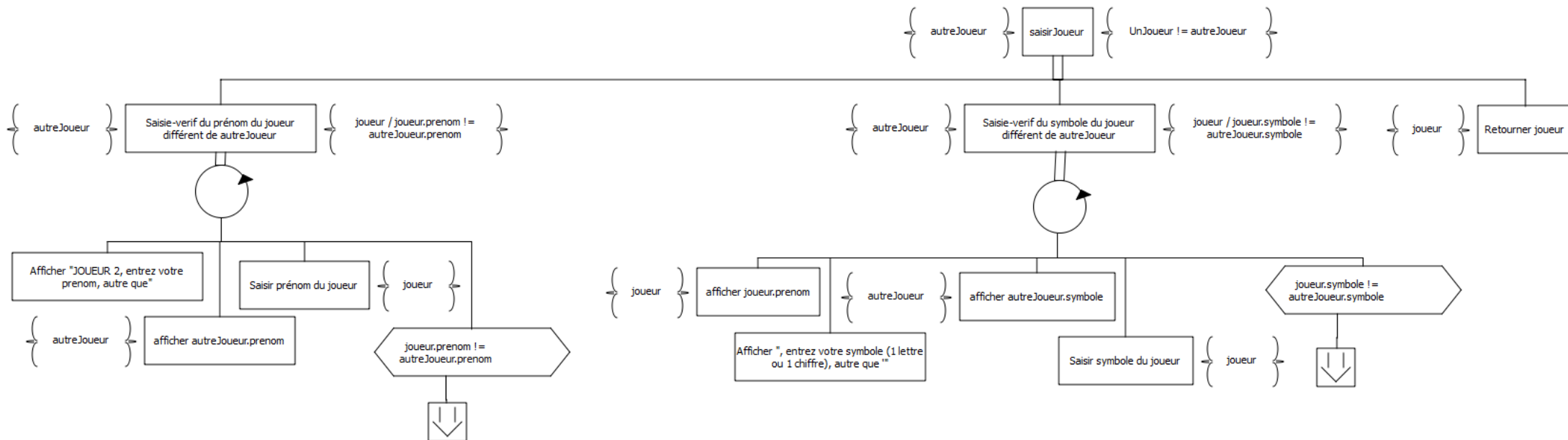
Nom	Type	Signification
joueur	UnJoueur	Contient les informations sur un joueur

#### 4.4.B.2. saisirJoueur (variante)

But de l'action: Retourne un joueur avec un prénom et un symbole différent de autreJoueur

Stratégie de l'algorithme mise en oeuvre: Grâce à une méthode de saisie-verif, on va vérifier que le prénom, comme le symbole soient différents de ceux déjà saisis dans la variable autreJoueur. Ensuite, on retourne un joueur de type UnJoueur.

Algorithme :





## Dictionnaire des données

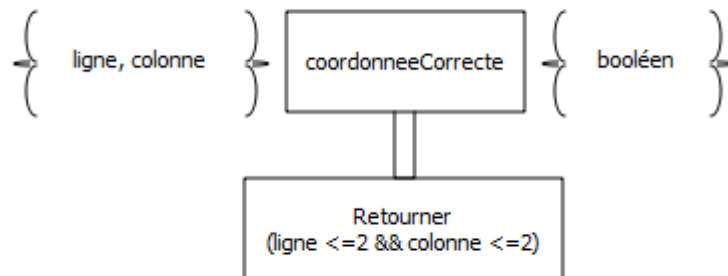
Nom	Type	Signification
joueur	UnJoueur	Le joueur qui sera retourné par cette fonction ayant un prénom et un symbole différent de autreJoueur
autreJoueur	UnJoueur	Joueur qui va permettre à "joueur" d'avoir des informations différentes de les siennes

#### 4.4.B.3. coordonneeCorrecte

But de l'action: Vérifier si la coordonnée est dans la limite de la taille de la grille, retourne vrai, si elle se trouve dans les limites, sinon, retourne faux.

Stratégie de l'algorithme mise en oeuvre: Sachant que ligne, comme colonne sont des entiers positifs, on va ainsi vérifier que celles-ci soient inférieures ou égales à deux. (Pas besoin de vérifier que celles-ci soient supérieures ou égales à zéro étant donné que ce sont des entiers positifs).

Algorithme :



#### Dictionnaire des données

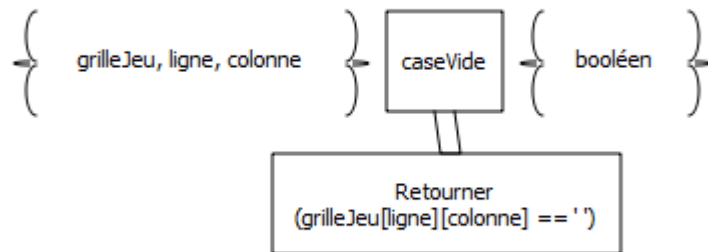
Nom	Type	Signification
ligne	Entier positif	Coordonnée de la ligne à vérifier
colonne	Entier Positif	Coordonnée de la colonne à vérifier

#### 4.4.B.4. caseVide

But de l'action: Vérifier si la case de la coordonnée est vide, retourne vrai, si la case est vide, sinon, retourne faux.

Stratégie de l'algorithme mise en œuvre: Cette fonction va vérifier si la coordonnée saisie par le joueur est vide, c'est-à-dire, si elle a un espace. On retourne "vrai" si la case a un espace, sinon on retourne "faux".

Algorithme:



#### Dictionnaire des données

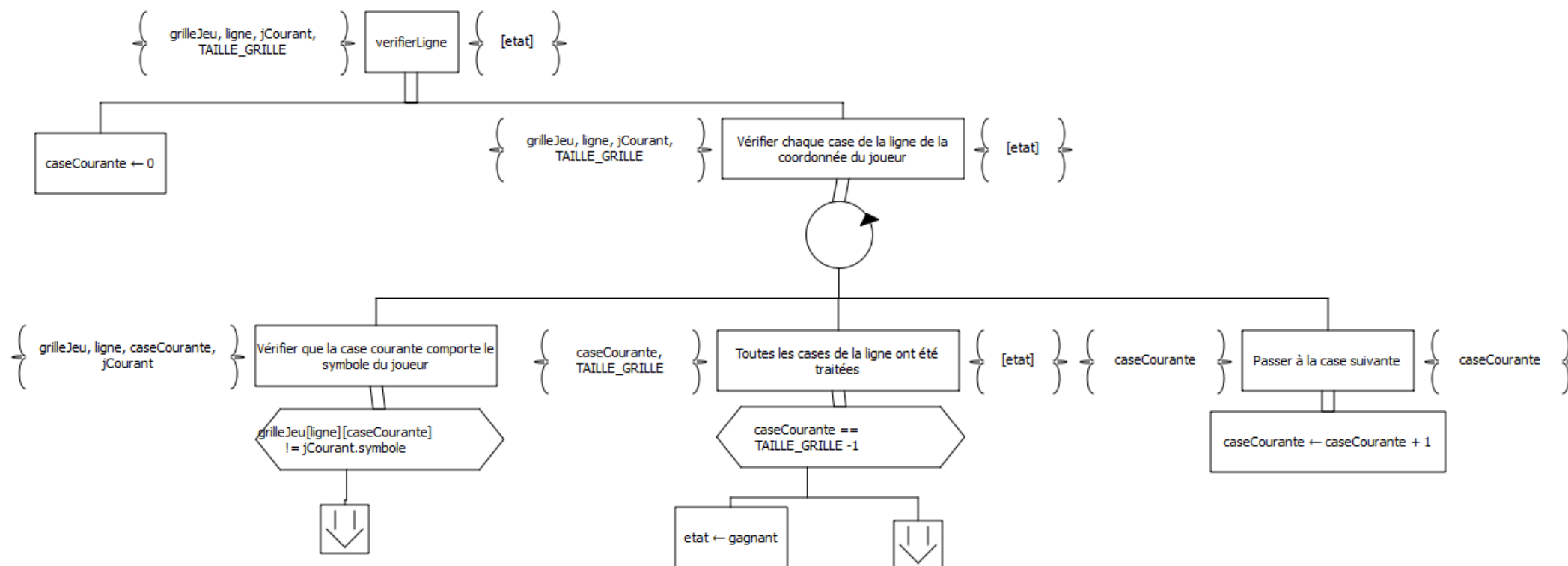
Nom	Type	Signification
grilleJeu	Tableau de caractère	Grille du jeu
ligne	Entier positif	Ligne de la coordonnée qui va être vérifiée
colonne	Entier positif	Colonne de la coordonnée qui va être vérifiée

#### **4.4.B.5. verifierLigne**

But de l'action: Vérifier si le joueur courant (jCourant) gagne sur la ligne de sa coordonnée dans la grille. Si le joueur gagne sur la ligne, l'état du jeu (etat) passe à gagnant.

Stratégie de l'algorithme mise en oeuvre: On va traiter toutes les cases de la ligne (en incrémentant caseCourante à chaque fin d'itération), si une des cases n'a pas le symbole du joueur courant, on sort de la boucle. Sinon si toutes les cases ont été traitées, cela veut dire que le joueur courant a gagné, l'état du jeu passe ainsi à "gagnant" et on sort de la boucle.

## Algorithme :



## Dictionnaire des données

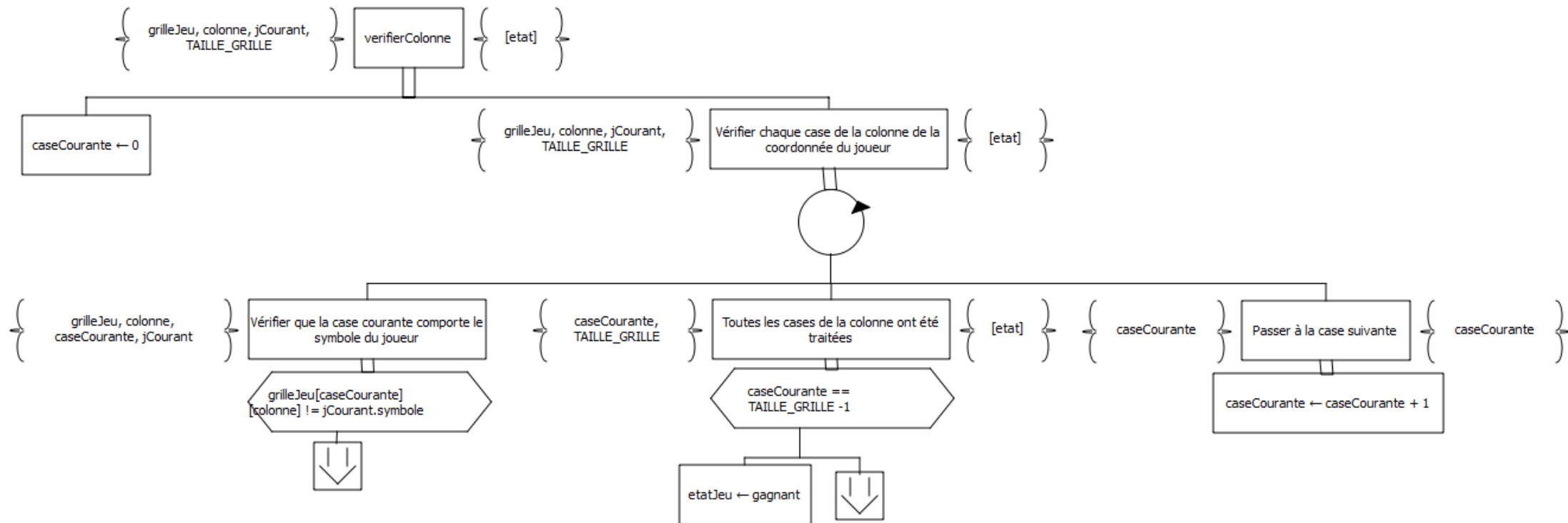
Nom	Type	Signification
grilleJeu	Tableau de caractère	Grille du jeu
TAILLE_GRILLE	Entier positif	Constante représentant la taille de la grille
ligne	Entier positif	Ligne courante à vérifier
caseCourante	Entier positif	Case courante qui vérifiera les conditions d'arrêt (soit, la case n'a pas le symbole du joueur, soit, toutes les cases ont été traitées et le joueur a gagné)
etat	UnEtat	Etat du jeu
jCourant	UnJoueur	Joueur courant où on va vérifier s'il gagne sur la ligne de sa coordonnée ou non.

#### **4.4.B.6. verifierColonne**

But de l'action: Vérifier si le joueur courant (jCourant) gagne sur la colonne de sa coordonnée dans la grille. Si le joueur gagne sur la colonne, l'état du jeu (etat) passe à gagnant.

Stratégie de l'algorithme mise en oeuvre: On va traiter toutes les cases de la colonne (en incrémentant caseCourante à chaque fin d'itération), si une des cases n'a pas le symbole du joueur courant, on sort de la boucle. Sinon si toutes les cases ont été traitées, cela veut dire que le joueur courant a gagné, l'état du jeu passe ainsi à "gagnant" et on sort de la boucle.

Algorithme :





## Dictionnaire des données

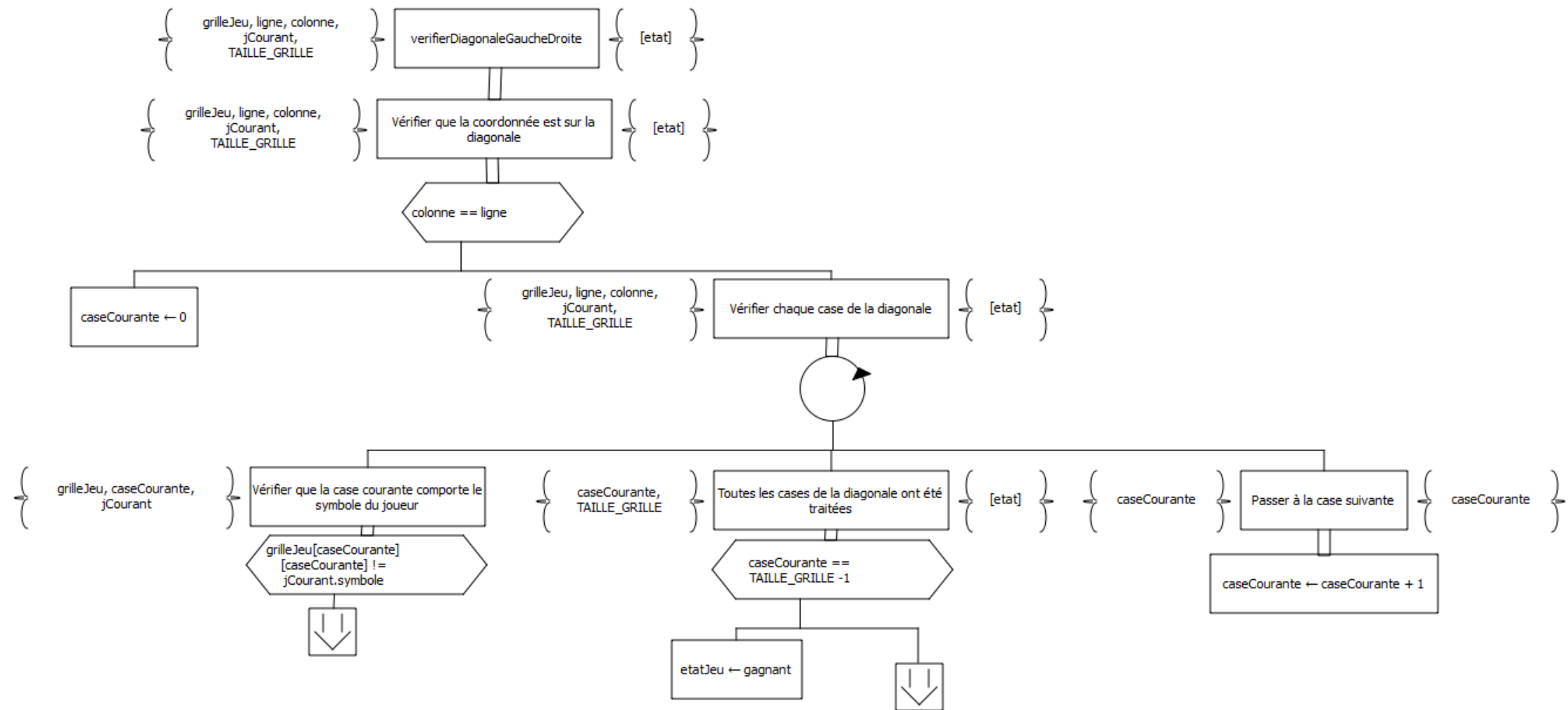
Nom	Type	Signification
grilleJeu	Tableau de caractère	Grille du jeu
TAILLE_GRILLE	Entier positif	Constante représentant la taille de la grille
colonne	Entier positif	Ligne courante à vérifier
caseCourante	Entier positif	Case courante qui vérifiera les conditions d'arrêt (soit, la case n'a pas le symbole du joueur, soit, toutes les cases ont été traitées et le joueur a gagné)
etat	UnEtat	Etat du jeu
jCourant	UnJoueur	Joueur courant où on va vérifier s'il gagne sur la colonne de sa coordonnée ou non.

#### **4.4.B.7. verifierDiagonaleGaucheDroite**

But de l'action: Vérifie si le joueur courant (jCourant) gagne sur la diagonale de gauche à droite de la coordonnée dans la grille. Si le joueur gagne sur la diagonale (gauche à droite), l'état du jeu (etat) passe à gagnant.

Stratégie de l'algorithme mise en oeuvre: Afin d'optimiser le code, tout d'abord, on vérifie si la coordonnée du joueur est sur cette diagonale (numéro de la colonne qui est égale à celle de la ligne), si c'est le cas, on vérifie chaque case de la diagonale (en incrémentant caseCourante à chaque fin d'itération), si une des cases n'a pas le symbole du joueur courant, on sort de la boucle. Sinon si toutes les cases ont été traitées, cela veut dire que le joueur courant a gagné, l'état du jeu passe ainsi à "gagnant" et on sort de la boucle.

Algorithme :



## Dictionnaire des données

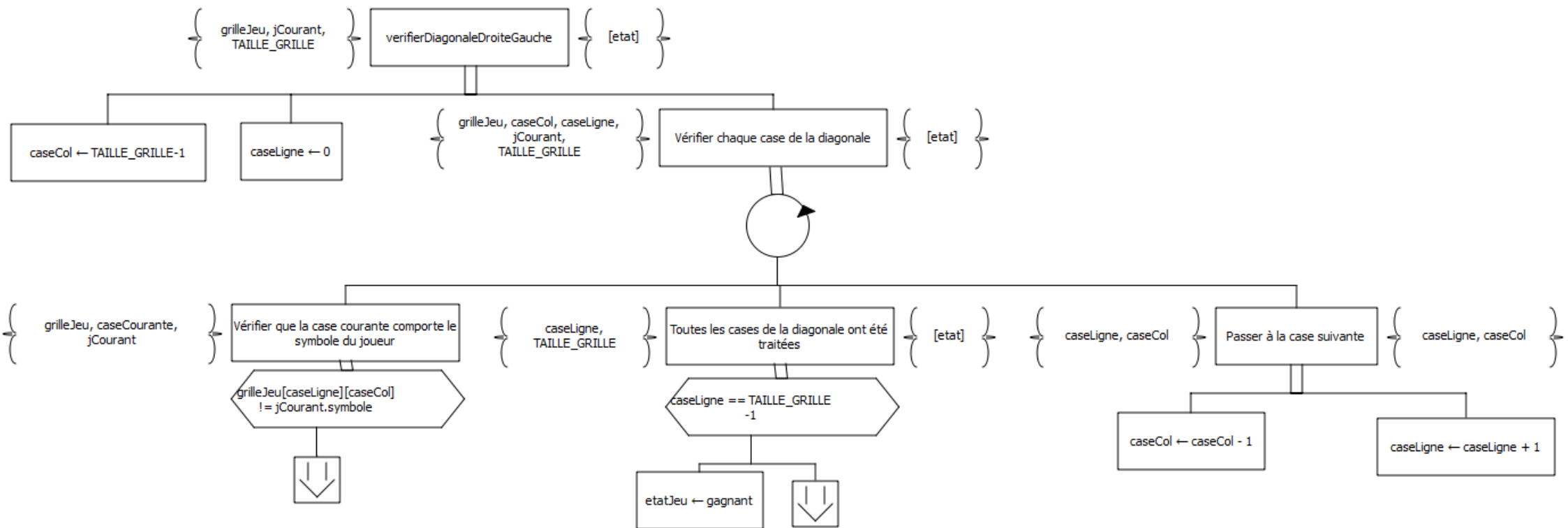
Nom	Type	Signification
grilleJeu	Tableau de caractère	Grille du jeu
TAILLE_GRILLE	Entier positif	Constante représentant la taille de la grille
ligne	Entier positif	Ligne courante qui sera vérifiée
colonne	Entier positif	Colonne courante qui sera vérifiée
caseCourante	Entier positif	Case courante qui vérifiera les conditions d'arrêt (soit, la case n'a pas le symbole du joueur, soit, toutes les cases ont été traitées et le joueur a gagné)
etat	UnEtat	Etat du jeu
jCourant	UnJoueur	Joueur courant où on va vérifier s'il gagne sur la diagonale (de gauche à droite) de sa coordonnée ou non

#### **4.4.B.8. verifierDiagonaleDroiteGauche**

But de l'action: Vérifie si le joueur courant (jCourant) gagne sur la diagonale de droite à gauche de la coordonnée dans la grille. Si le joueur gagne sur la diagonale (droite à gauche), l'état du jeu (etat) passe à gagnant.

Stratégie de l'algorithme mise en oeuvre: On vérifie chaque case de la diagonale (en incrémentant caseLigne à chaque fin d'itération et décrémentant caseCol), si une des cases n'a pas le symbole du joueur courant, on sort de la boucle. Sinon si toutes les cases ont été traitées (on prendra caseLigne qui est égale à deux pour sortir) , cela veut dire que le joueur courant a gagné, l'état du jeu passe ainsi à "gagnant" et on sort de la boucle. (PS : Nous n'avons pas trouvé de solution pour éviter de traiter cette étape si la coordonnée ne se trouve pas sur cette diagonale).

Algorithme :



## Dictionnaire des données

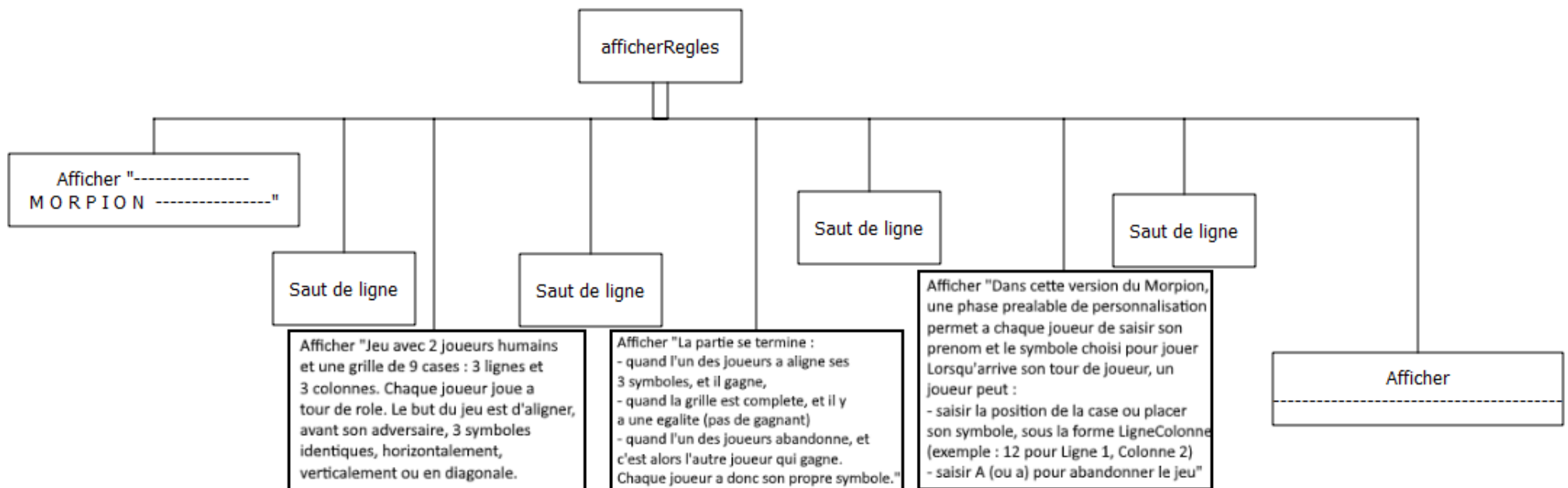
Nom	Type	Signification
grilleJeu	Tableau de caractère	Grille du jeu
TAILLE_GRILLE	Entier positif	Constante représentant la taille de la grille
caseCol	Entier positif	Case de la colonne courante qui vérifiera la conditions d'arrêt : la case n'a pas le symbole du joueur grâce à aussi la variable caseLigne
caseLigne	Entier positif	Case de la ligne courante qui vérifiera les conditions d'arrêt (soit, la case n'a pas le symbole du joueur grâce à aussi la variable caseCol, soit, toutes les cases ont été traités et le joueur a gagné)
jCourant	UnJoueur	Joueur courant où on va vérifier s'il gagne sur la diagonale ( de droite à gauche) de sa coordonnée ou non
etat	UnEtat	Etat du jeu

#### 4.4.B.9. afficherRegles

But de l'action: Affiche à l'écran les règles du jeu

Stratégie de l'algorithme mise en oeuvre: On affiche tout le texte des règles

Algorithme :



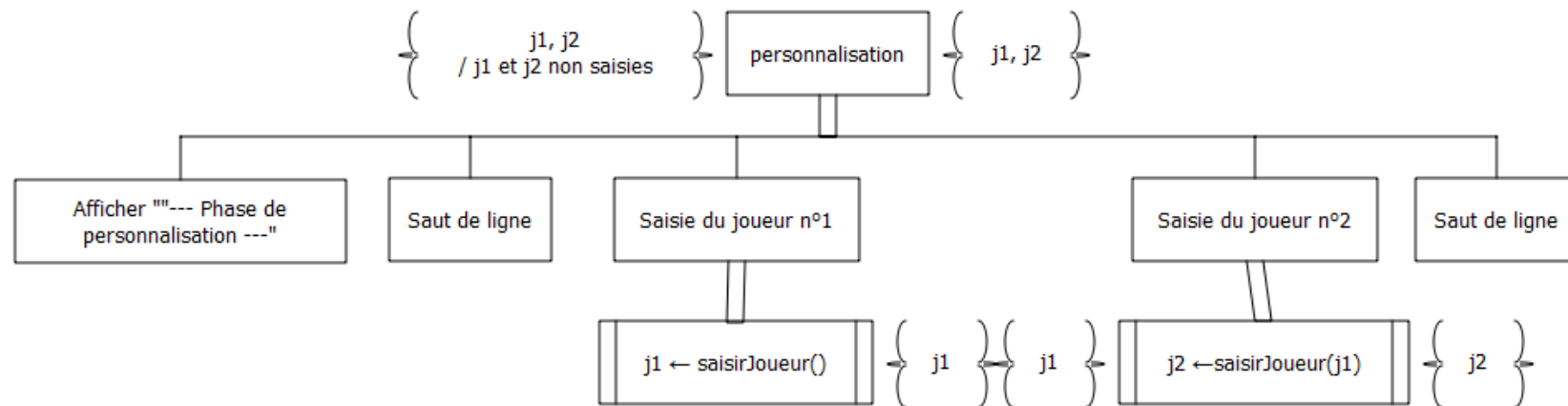


#### **4.4.B.10. personnalisation**

But de l'action: Permet de personnaliser le prénom et le symbole des joueurs j1 et j2

Stratégie de l'algorithme mise en oeuvre: On va d'abord appeler la fonction saisirJoueur sans paramètre afin de saisir le premier joueur qui sera dans la variable j1 puis on appelle la fonction saisirJoueur avec le paramètre j1 qui va permettre de saisir le second joueur qui sera différent du premier (le second joueur sera stocké dans la variable j2).

Algorithme:



## Dictionnaire des données

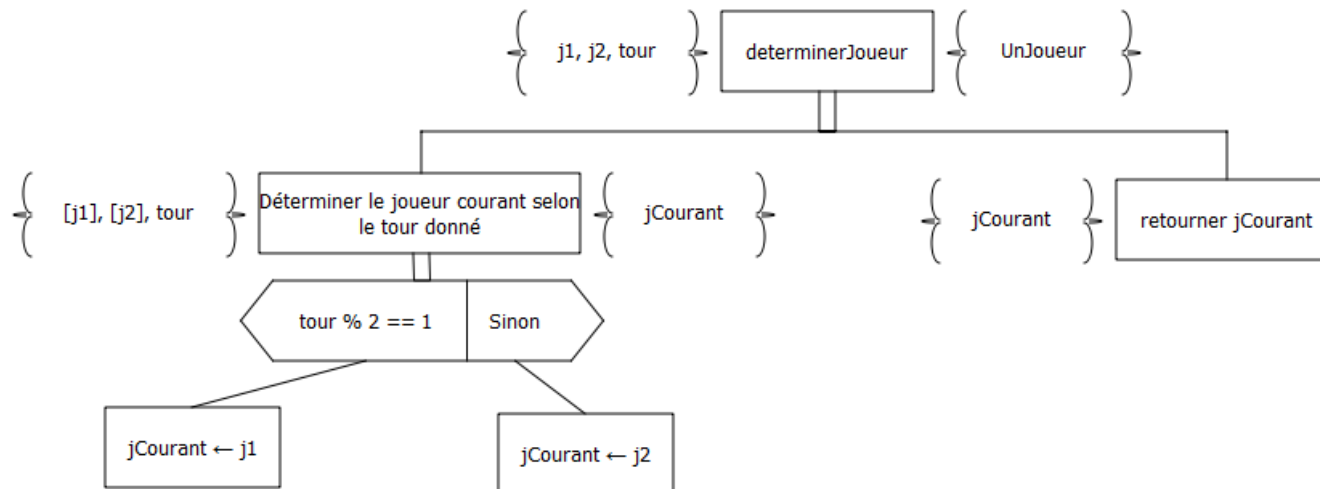
Nom	Type	Signification
j1	UnJoueur	Premier joueur qui sera saisi
j2	UnJoueur	Second joueur qui sera saisi

#### 4.4.B.11. determinerJoueur

But de l'action: Retourne un joueur entre le joueur j1 et j2 selon le numéro de tour.

Stratégie de l'algorithme mise en oeuvre: Grâce au modulo du tour par le nombre de joueurs (2), si ce modulo est égal à 1, le joueur courant sera le premier joueur, sinon, ce sera le second joueur. Puis on retourne ainsi le joueur courant

Algorithme:



## Dictionnaire des données

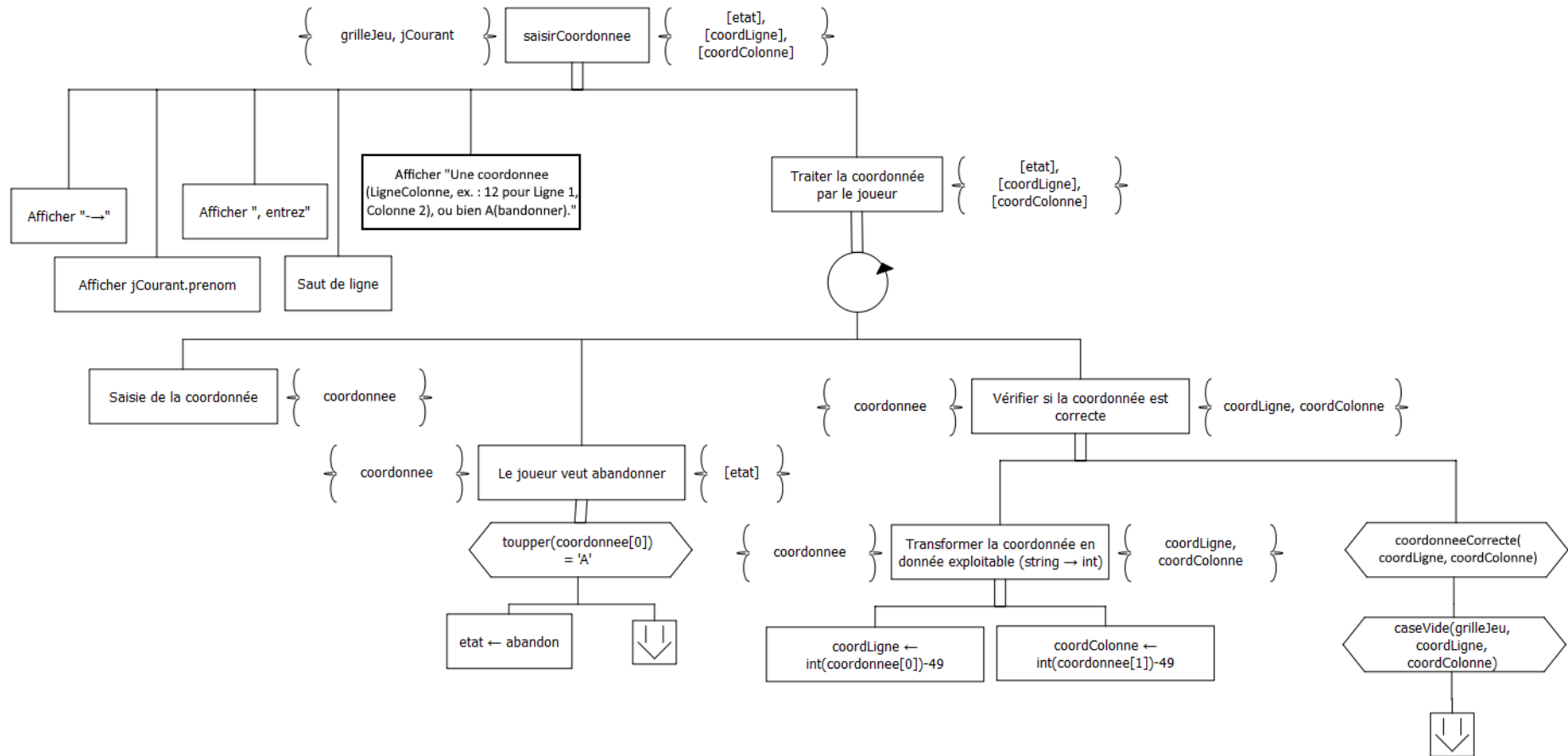
Nom	Type	Signification
j1	UnJoueur	Joueur 1 qui peut être le joueur courant
j2	UnJoueur	Joueur 2 qui peut être le joueur courant
jCourant	UnJoueur	Joueur courant déterminé grâce au numéro du tour
tour	Entier positif	Numéro du tour qui va permettre de déterminer le joueur courant

#### **4.4.B.12. saisirCoordonnee**

But de l'action: Saisie de la coordonnée par le joueur courant (jCourant) qui sera mise dans la grille de jeu (grilleJeu). Si le joueur saisie A (ou a), l'état du jeu (etat) passe à 'abandon'.

Stratégie de l'algorithme mise en oeuvre: Tout d'abord, on invite le joueur a saisir une coordonnée. Grâce à un schéma "saisie-verif", le joueur saisie puis on traite sa donnée : on va d'abord vérifier s'il souhaite quitter le jeu. S'il souhaite quitter le jeu, etat prend "abandon" et on sort de la boucle. Sinon, on transforme notre coordonnée (qui est une chaîne de caractère) en des données exploitables (en entier positif) : on a ainsi, la coordonnée en numéro de ligne et en numéro de colonne. Ensuite, on vérifie si cette coordonnée est correcte (c'est à dire dans la taille de la grille) puis que la case de la coordonnée est vide. Si toutes les conditions sont vérifiées, on quitte la boucle.

## Algorithme :



## Dictionnaire des données

Nom	Type	Signification
grilleJeu	Tableau de caractère	Grille du jeu
TAILLE_GRILLE	Entier positif	Constante représentant la taille de la grille
coordLigne	Entier positif	Coordonné de ligne choisie par le joueur
coordColonne	Entier positif	Coordonné du colonne choisie par le joueur
etat	UnEtat	Etat du jeu
jCourant	UnJoueur	Joueur courant qui va saisir la coordonnée
coordonnee	chaîne de caractères	Coordonnée saisie par le joueur



#### **4.4.B.13. resultat**

But de l'action: Retourne le résultat de fin de jeu,

Si etat = abandon, un joueur (jGagnant) gagne car l'autre joueur (jCourant) a quitté le jeu puis on affiche à quel tour le joueur a gagné (tour)

Si etat = gagnant, un joueur (jGagnant) a gagné puis on affiche à quel tour le joueur a gagné (tour)

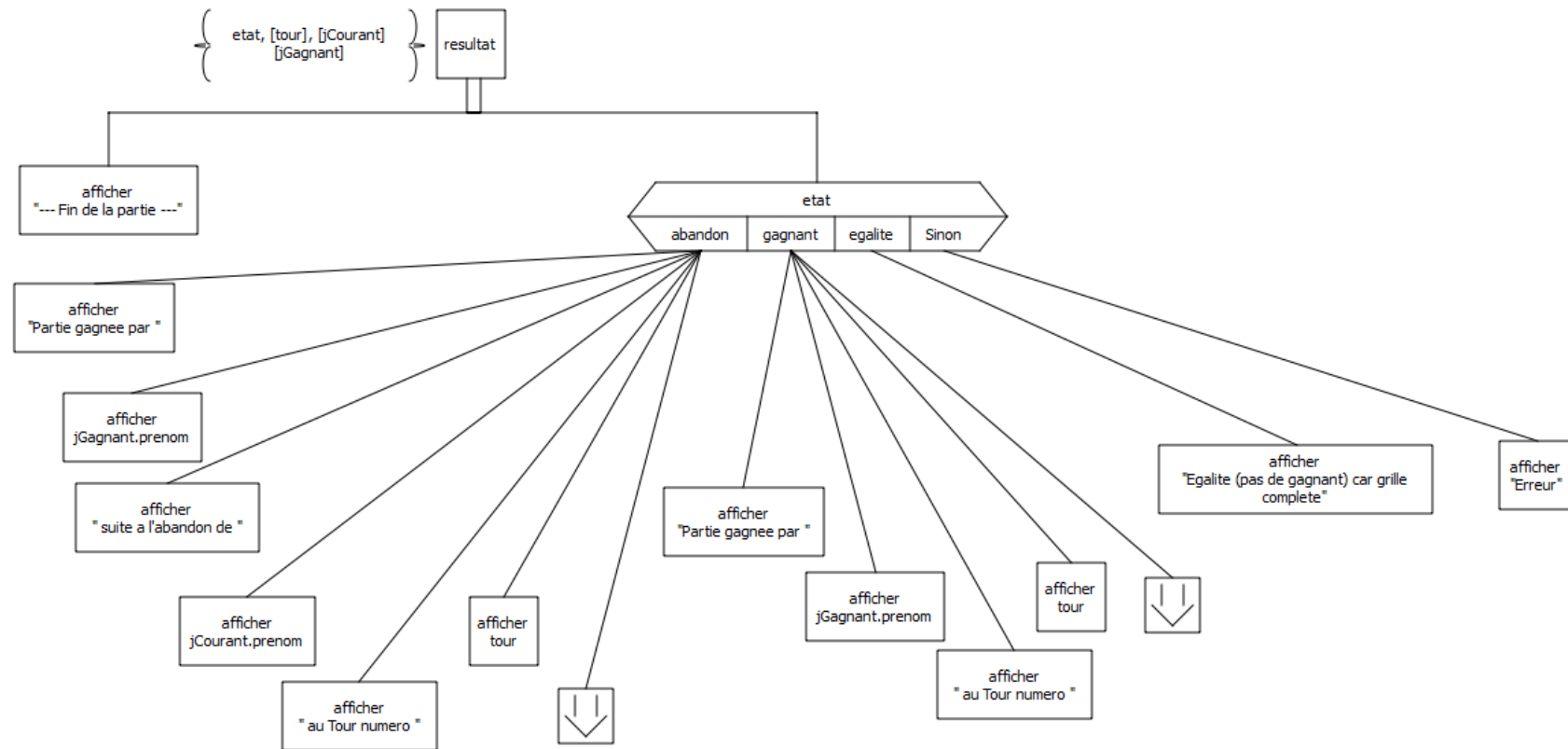
Si etat = egalite, aucun joueur n'a gagné : il y a égalité

Pré-condition : etat ne peut avoir que pour valeur : abandon, gagnant ou egalite

Stratégie de l'algorithme mise en oeuvre: On utilise un switch afin de vérifier l'état de la variable état. Si l'état est incorrect, on affiche un message d'erreur.

Traitement :

## Algorithme :



## Dictionnaire des données

Nom	Type	Signification
etat	UnEtat	Etat du jeu
tour	Entier positif	Numéro du tour
jCourant	UnJoueur	Joueur courant du numéro de tour qui représente le joueur qui a perdu suite à son abandon
jGagnant	UnJoueur	Joueur gagnant

#### **4.4.B.14. verifierGagnant**

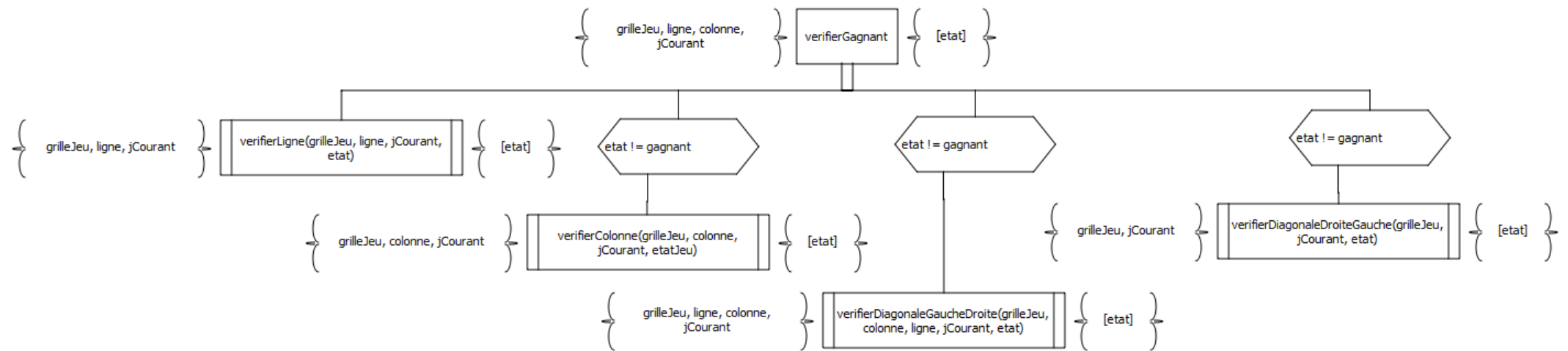
But de l'action: Essaye toutes les combinaisons dont le joueur peut gager à partir d'une coordonnée donnée (ligne et colonne)

Le joueur peut gagner soit en ligne, en colonne ou bien par diagonale (de droite à gauche ou de gauche à droite)

Si le joueur gagne sur une de ses combinaisons, etat prend "gagnant".

Stratégie de l'algorithme mise en oeuvre: On va d'abord vérifier si le joueur gagne sur une ligne, si ce n'est pas le cas, afin d'optimiser le code, on va vérifier bien évidemment que le joueur n'a pas gagné grâce à la condition "etat != gagnant" à chaque vérification, on continue ainsi en vérifiant : la colonne et les 2 diagonales

Algorithme :




## Dictionnaire des données

Nom	Type	Signification
grilleJeu	Tableau de caractère	Grille du jeu
TAILLE_GRILLE	Entier positif	Constante représentant la taille de la grille
ligne	Entier positif	Ligne de la coordonnée du joueur
colonne	Entier positif	Colonne de la coordonnée du joueur
jCourant	UnJoueur	Joueur courant (du tour actuel)
etat	UnEtat	Etat du Jeu

## 5. Traces d'exécution

Copies d'écran correspondant au comportement décrit au point 3.

Figure 1 : Phase initiale de personnalisation (commune à tous les scénarios)



```
C:\Windows\system32\cmd.exe - jeu.exe

----- M O R P I O N -----

Jeu avec 2 joueurs humains et une grille de 9 cases : 3 lignes et 3 colonnes.
Chaque joueur joue a tour de role.
Le but du jeu est d'aligner, avant son adversaire, 3 symboles identiques, horizontalement, verticalement ou en diagonale.

La partie se termine :
- quand l'un des joueurs a aligne ses 3 symboles, et il gagne,
- quand la grille est complete, et il y a une egalite (pas de gagnant)
- quand l'un des joueurs abandonne, et c'est alors l'autre joueur qui gagne.
Chaque joueur a donc son propre symbole.

Dans cette version du Morpion, une phase prealable de personnalisation permet a chaque joueur de saisir son prenom et le symbole
choisi pour jouer
Lorsqu'arrive son tour de joueur, un joueur peut :
- saisir la position de la case ou placer son symbole, sous la forme LigneColonne (exemple : 12 pour Ligne 1, Colonne 2)
- saisir A (ou a) pour abandonner le jeu

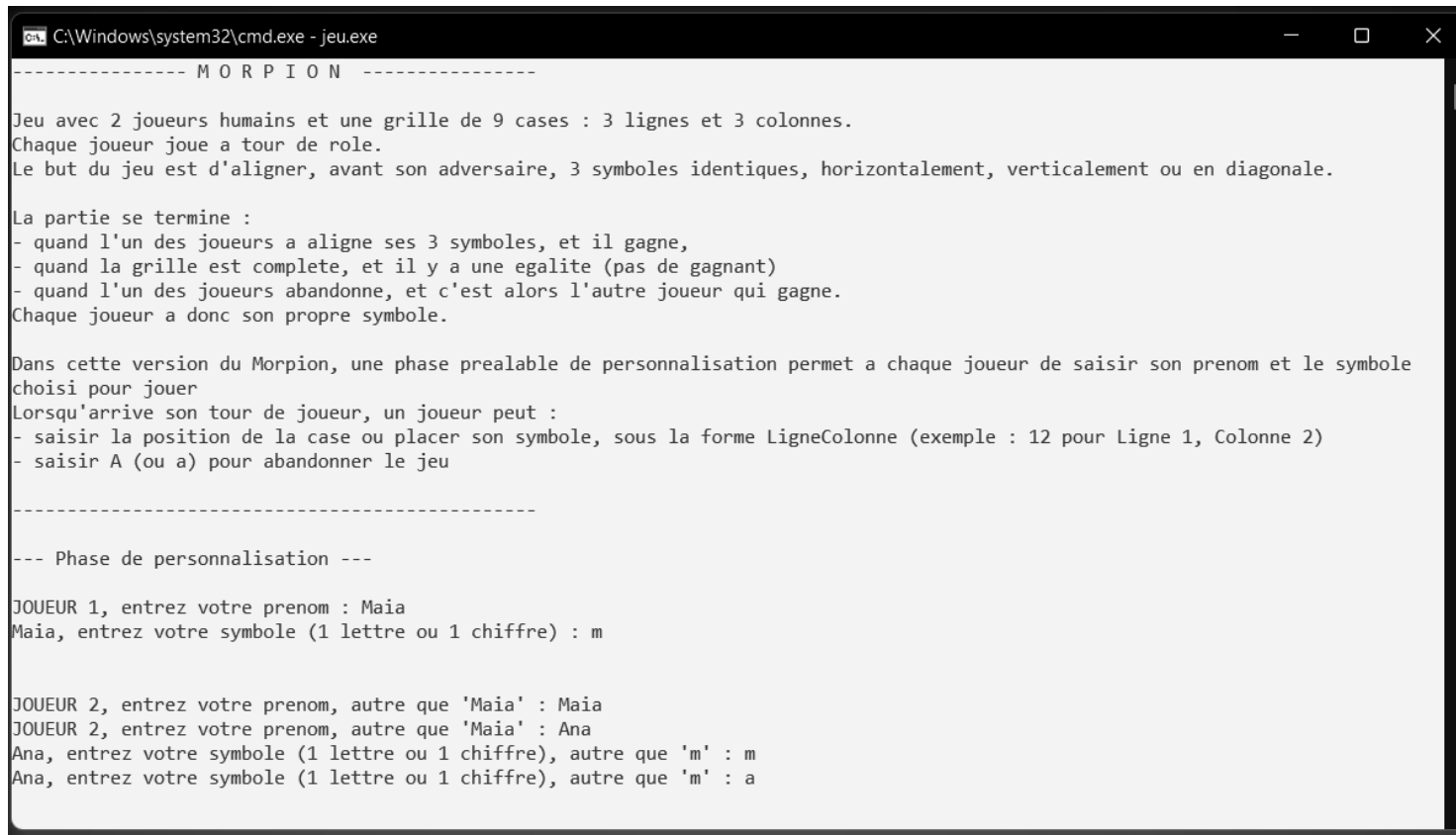
-----

--- Phase de personnalisation ---

JOUEUR 1, entrez votre prenom : Maia
Maia, entrez votre symbole (1 lettre ou 1 chiffre) : m

JOUEUR 2, entrez votre prenom, autre que 'Maia' : Ana
Ana, entrez votre symbole (1 lettre ou 1 chiffre), autre que 'm' : a
```

Figure 2 : Scénario d'exception : Phase initiale de personnalisation, le joueur 2 saisie un prénom et/ou symbole identique au joueur 1



```
C:\Windows\system32\cmd.exe - jeu.exe
----- M O R P I O N -----

Jeu avec 2 joueurs humains et une grille de 9 cases : 3 lignes et 3 colonnes.
Chaque joueur joue a tour de role.
Le but du jeu est d'aligner, avant son adversaire, 3 symboles identiques, horizontalement, verticalement ou en diagonale.

La partie se termine :
- quand l'un des joueurs a aligne ses 3 symboles, et il gagne,
- quand la grille est complete, et il y a une egalite (pas de gagnant)
- quand l'un des joueurs abandonne, et c'est alors l'autre joueur qui gagne.
Chaque joueur a donc son propre symbole.

Dans cette version du Morpion, une phase prealable de personnalisation permet a chaque joueur de saisir son prenom et le symbole
choisi pour jouer
Lorsqu'arrive son tour de joueur, un joueur peut :
- saisir la position de la case ou placer son symbole, sous la forme LigneColonne (exemple : 12 pour Ligne 1, Colonne 2)
- saisir A (ou a) pour abandonner le jeu

-----

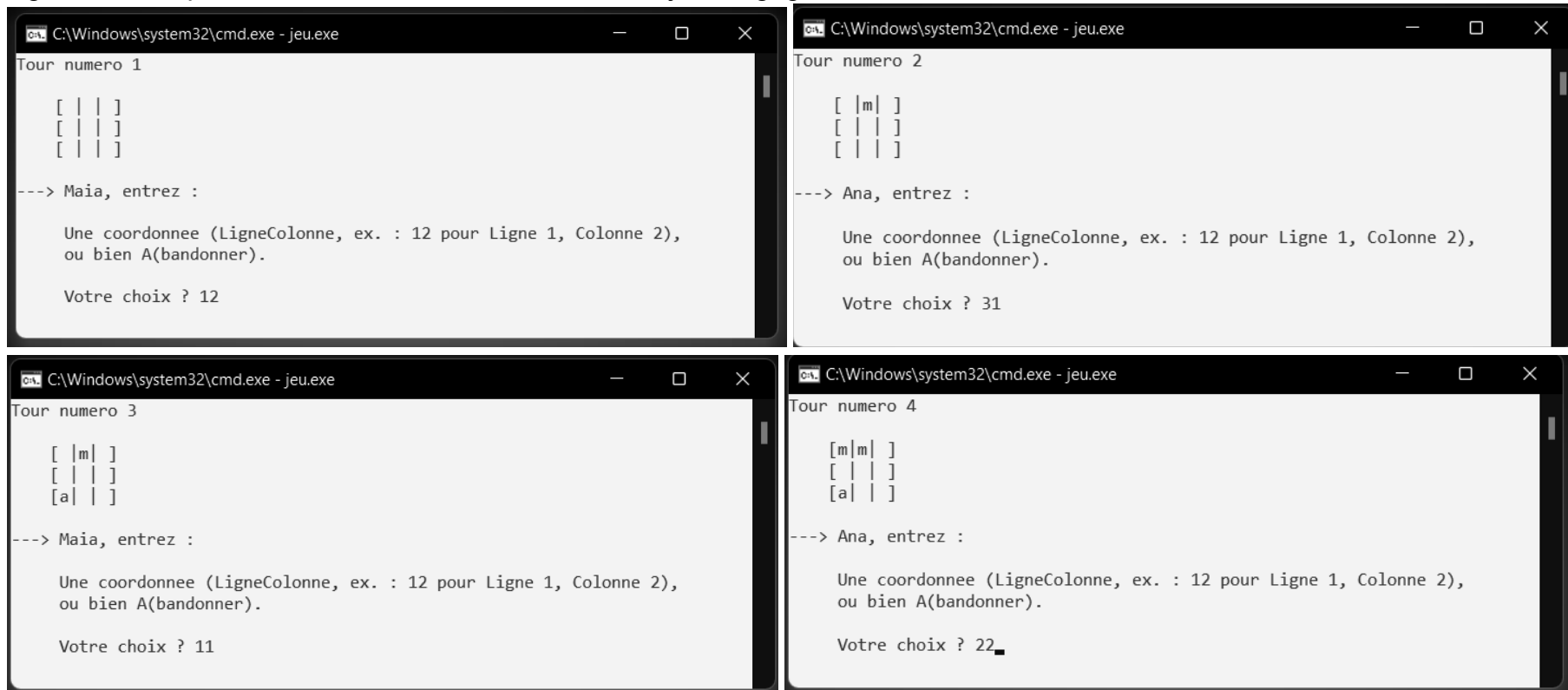
--- Phase de personnalisation ---

JOUEUR 1, entrez votre prenom : Maia
Maia, entrez votre symbole (1 lettre ou 1 chiffre) : m

JOUEUR 2, entrez votre prenom, autre que 'Maia' : Maia
JOUEUR 2, entrez votre prenom, autre que 'Maia' : Ana
Ana, entrez votre symbole (1 lettre ou 1 chiffre), autre que 'm' : m
Ana, entrez votre symbole (1 lettre ou 1 chiffre), autre que 'm' : a
```



Figure 3 : Comportement lié au scénario nominal : Un joueur gagne



```
C:\Windows\system32\cmd.exe - jeu.exe
Tour numero 5

[m|m| ]
[ |a| ]
[a| | ]

---> Maia, entrez :

Une coordonnee (LigneColonne, ex. : 12 pour Ligne 1, Colonne 2),
ou bien A(bandonner).

Votre choix ? 13
```

```
C:\Windows\system32\cmd.exe
Tour numero 5

[m|m|m]
[ |a| ]
[a| | ]

--- Fin de la partie ---

Partie gantee par Maia au Tour numero 5

C:\Users\samue\Desktop\s1-02-comparaison-d-approches-algorithmiques-main\jeu
1>
```

Figure 4 : Comportement lié au scénario alternatif : Un joueur abandonne

```
C:\Windows\system32\cmd.exe - jeu.exe
Tour numero 1

[ | | ]
[ | | ]
[ | | ]

---> Maia, entrez :

Une coordonnee (LigneColonne, ex. : 12 pour Ligne 1, Colonne 2),
ou bien A(bandonner).

Votre choix ? 12

C:\Windows\system32\cmd.exe - jeu.exe
Tour numero 2

[ |m| ]
[ | | ]
[ | | ]

---> Ana, entrez :

Une coordonnee (LigneColonne, ex. : 12 pour Ligne 1, Colonne 2),
ou bien A(bandonner).

Votre choix ? 31

C:\Windows\system32\cmd.exe - jeu.exe
Tour numero 3

[ |m| ]
[ | | ]
[a| | ]

---> Maia, entrez :

Une coordonnee (LigneColonne, ex. : 12 pour Ligne 1, Colonne 2),
ou bien A(bandonner).

Votre choix ? a

C:\Windows\system32\cmd.exe - jeu.exe
Tour numero 3

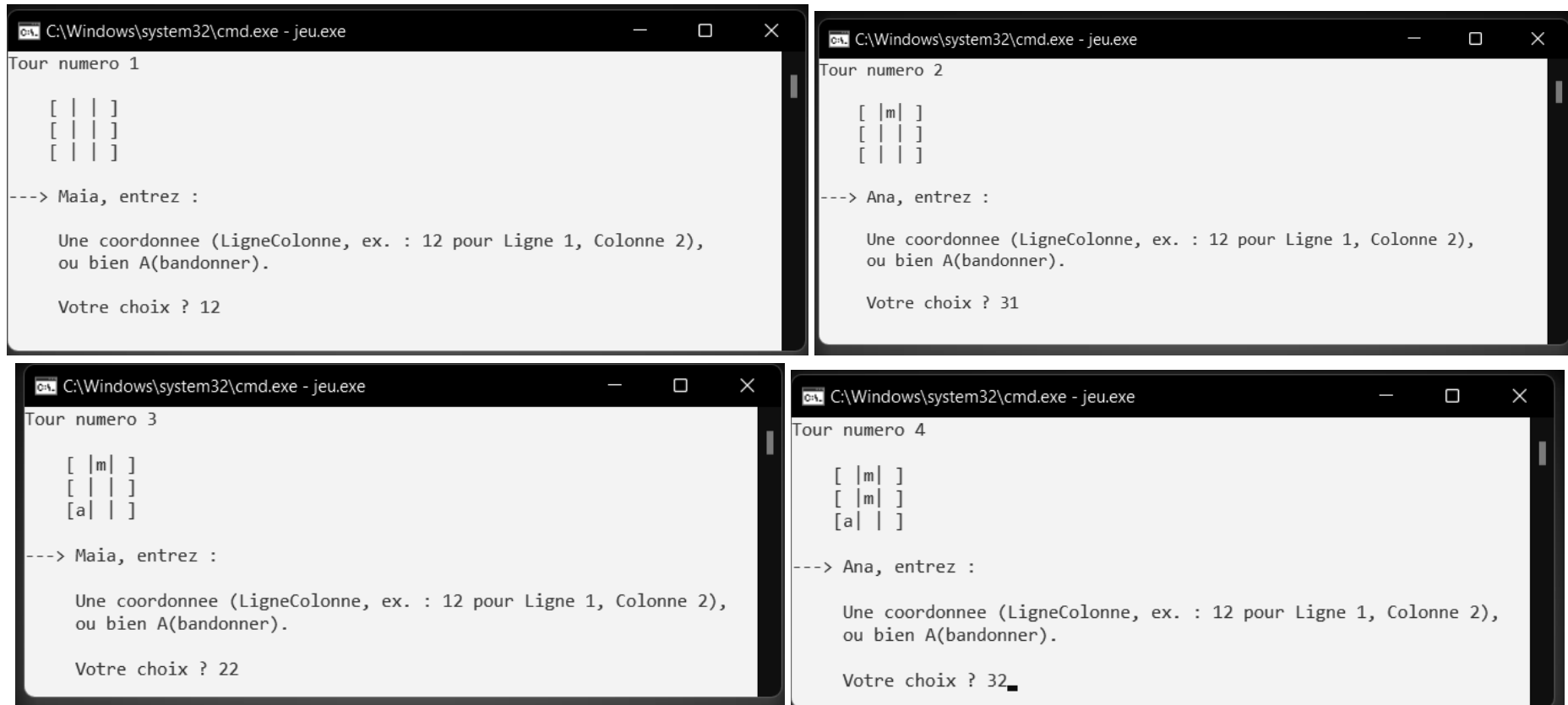
[ |m| ]
[ | | ]
[a| | ]

--- Fin de la partie ---

Partie gagnee par Ana suite a l'abandon de Maia au Tour numero 3

C:\Users\samue\Desktop\s1-02-comparaison-d-approches-algorithmiques-main\
jeu1>
```

Figure 5 : Comportement lié au scénario alternatif: grille complète et pas de gagnant à la fin de la partie



```
C:\Windows\system32\cmd.exe - jeu.exe
Tour numero 5

[ |m| ]
[ |m| ]
[a|a| ]

---> Maia, entrez :

Une coordonnee (LigneColonne, ex. : 12 pour Ligne 1, Colonne 2),
ou bien A(bandonner).

Votre choix ? 33
```

```
C:\Windows\system32\cmd.exe - jeu.exe
Tour numero 6

[ |m| ]
[ |m| ]
[a|a|m]

---> Ana, entrez :

Une coordonnee (LigneColonne, ex. : 12 pour Ligne 1, Colonne 2),
ou bien A(bandonner).

Votre choix ? 11
```

```
C:\Windows\system32\cmd.exe - jeu.exe
Tour numero 7

[a|m| ]
[ |m| ]
[a|a|m]

---> Maia, entrez :

Une coordonnee (LigneColonne, ex. : 12 pour Ligne 1, Colonne 2),
ou bien A(bandonner).

Votre choix ? 21
```

```
C:\Windows\system32\cmd.exe - jeu.exe
Tour numero 8

[a|m| ]
[m|m| ]
[a|a|m]

---> Ana, entrez :

Une coordonnee (LigneColonne, ex. : 12 pour Ligne 1, Colonne 2),
ou bien A(bandonner).

Votre choix ? 23
```

```
C:\Windows\system32\cmd.exe - jeu.exe
Tour numero 9

[a|m| ]
[m|m|a]
[a|a|m]

---> Maia, entrez :

Une coordonnee (LigneColonne, ex. : 12 pour Ligne 1, Colonne 2),
ou bien A(bandonner).

Votre choix ? 13
```

```
Sélection C:\Windows\system32\cmd.exe
Tour numero 9

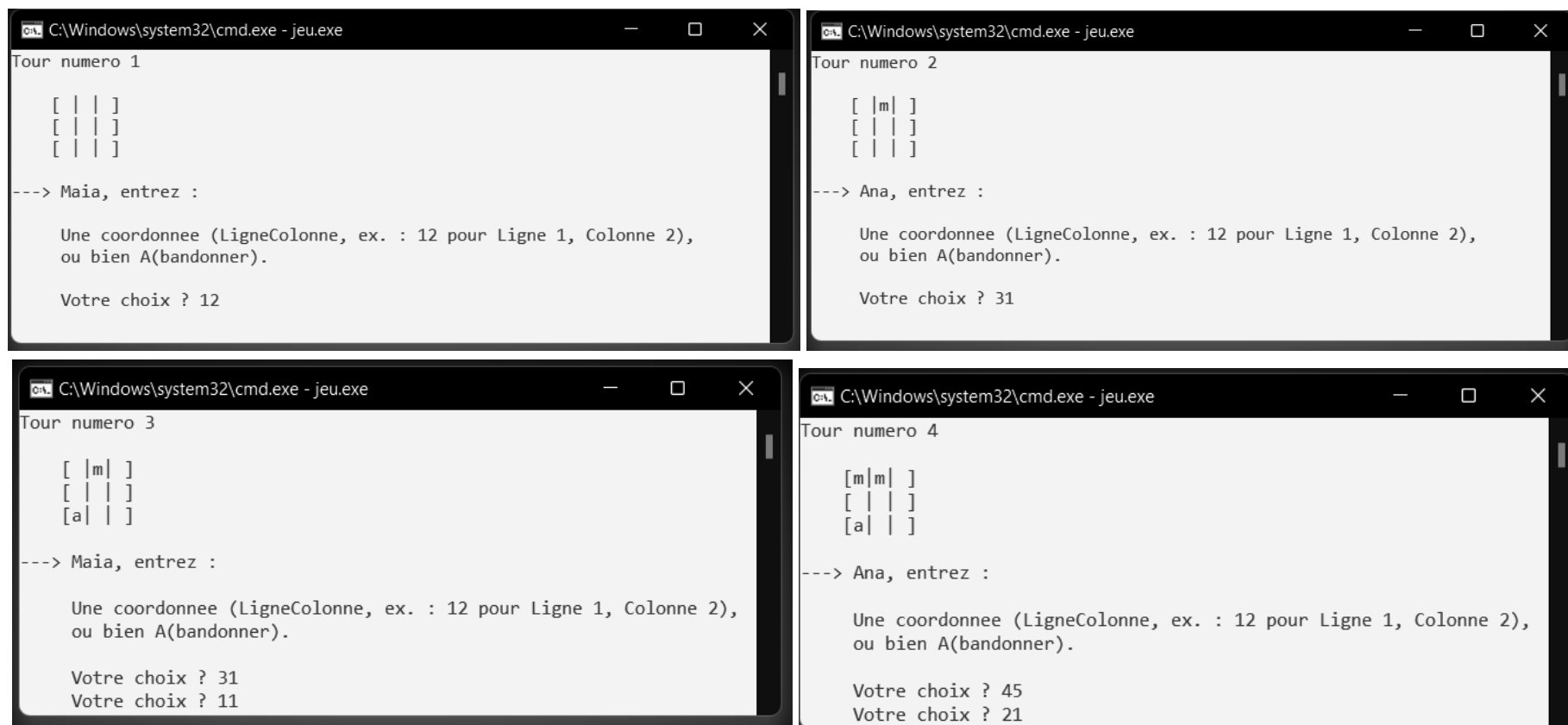
[a|m|m]
[m|m|a]
[a|a|m]

--- Fin de la partie ---

Egalite (pas de gagnant) car grille complete

C:\Users\samue\Desktop\s1-02-comparaison-d-approches-algorithmiques-main\jeu1>
```

**Figure 6** : Comportement lié au scénario d'exception: Le joueur saisit une coordonnée déjà utilisée ou une coordonnée qui n'existe pas.



## 6. Remarques

Nous n'avons pas mis d'accents sur les phrases car nous n'avons pas trouvé de solution pour les accents.

Nous n'avons pas trouvé de solution pour éviter l'étape de vérification de la diagonale de droite à gauche, ce qui fait que l'on traite ce cas tout le temps...



## 7. Code C++

Fichier main.cpp joint au dossier avec en-tête certifiant l'originalité du code produit.

Nom du sous-programmes	Emplacement de la déclaration	Emplacement de la définition
initialiserPartie	main.cpp	main.cpp
joueurPartie		
finaliserPartie		
initialiserGrille	grille.h	grille.cpp
afficherGrille		
insererCoordonnee		
saisirJoueur	jeu.cpp	jeu.cpp
saisirJoueur (avec paramètre)		
coordonneeCorrecte		
caseVide		
verifierLigne		
verifierColonne		
verifierDiagonaleGaucheDroite		
verifierDiagonaleDroiteGauche		
afficherRegles	jeu.h	
personnalisation		
determinerJoueur		
saisirCoordonnee		
resultat		
verifierGagnant		
effacer	game-tools.h	game-tools.cpp