

Samuel HENTRICS LOISTINE

Cédric ETCHEPARE



S1.01

Implémentation d'un besoin client

Exercice 2 : Master Mind

PACK 15

Sommaire

- 1. Équipe - p.2**
- 2. Exercice traité : Pack 15 - p.2**
- 3. Rappel des spécifications du programme - p.2-5**
 - 3.1 Spécifications initiales - p.2-5**
 - 3.2 Spécifications complémentaires = extensions traitées - p.5**
- 4. Algorithmes du programme (action principale et ses sous-actions) - p.5-19**
 - 4.1 Initialiser la partie - p.5-10**
 - 4.1.1 But de l'action - p.5**
 - 4.1.2 Stratégie de l'algorithme mise en oeuvre - p.5**
 - 4.1.3 Algorithme - p.5-9**
 - 4.1.4 Dictionnaire des éléments associés à cet algorithme - p.9-10**
 - 4.2 Jouer la partie - p.10-18**
 - 4.2.1 But de l'action - p.10**
 - 4.2.2 Stratégie de l'algorithme mise en oeuvre - p.10**
 - 4.2.3 Algorithme - p.10-17**
 - 4.2.4 Dictionnaire des éléments associés à cet algorithme - p.18**
 - 4.3 Finaliser la partie - p.19**
 - 4.3.1 But de l'action - p.19**
 - 4.3.2 Stratégie de l'algorithme mise en oeuvre - p.19**
 - 4.3.3 Algorithme - p.19**
 - 4.3.4 Dictionnaire des éléments associés à cet algorithme - p.19**
- 5. Traces d'exécution - p.20-21**
- 6. Remarques - p.21**
- 7. Code C++ - p.21**

1. Équipe

HENTRICS LOISTINE Samuel TD II - TP3

ETCHEPARE Cédric TD II – TP3

2. Exercice traité : Pack 15

Exercice 2 : Master Mind

3. Rappel des spécifications du programme

3.1 Spécifications initiales

Description du jeu : Une combinaison de 4 lettres est initialisée au début de la partie mais n'est pas dévoilée au joueur. Le joueur a par la suite 12 tentatives pour essayer de retrouver cette combinaison sachant que les lettres sont : A, B, C, D et qu'elles peuvent se répéter.

S'il y arrive en 12 coups ou moins, il gagne la partie, sinon il perd.

A chaque combinaison proposée par le joueur, le jeu informe sur :

- le nombre de lettres proposées qui sont dans la combinaison et à la bonne place
- le nombre de lettres proposées qui sont dans la combinaison, mais pas à la bonne place

- Spécifications externes

- Scénario nominal : Le joueur gagne la partie

MASTER MIND

Le joueur doit trouver en 12 coups maximum, une combinaison de 4 lettres parmi les lettres : A B C D combinaison générée aléatoirement par le jeu

A chaque proposition du joueur, le jeu informe sur :

- le nombre de lettres proposées qui sont dans la combinaison, et à la bonne place,
- le nombre de lettres proposées qui sont dans la combinaison, mais pas à la bonne place.

On recherche : BCCC

Coup (1/12). Saisir 4 caractères parmi A B C D ? ABCD
1 bien placés 1 mal placés

Coup (2/12). Saisir 4 caractères parmi A B C D ? BCCC
4 bien placés 0 mal placés

B R A V O : 2 tentatives pour trouver.

- Scénario alternatif : Le joueur perd la partie

MASTER MIND

Le joueur doit trouver en 12 coups maximum, une combinaison de 4 lettres parmi les lettres : A B C D combinaison générée aléatoirement par le jeu

A chaque proposition du joueur, le jeu informe sur :

- le nombre de lettres proposées qui sont dans la combinaison, et à la bonne place,
- le nombre de lettres proposées qui sont dans la combinaison, mais pas à la bonne place.

On recherche : CDBA

Coup (1/12). Saisir 4 caractères parmi A B C D ? ABCD

0 bien placés 4 mal placés

Coup (2/12). Saisir 4 caractères parmi A B C D ? AABD

1 bien placés 2 mal placés

Coup (3/12). Saisir 4 caractères parmi A B C D ? ADDC

1 bien placés 2 mal placés

Coup (4/12). Saisir 4 caractères parmi A B C D ? DABC

1 bien placés 3 mal placés

Coup (5/12). Saisir 4 caractères parmi A B C D ? DBCA

1 bien placés 3 mal placés

Coup (6/12). Saisir 4 caractères parmi A B C D ? ADCB

1 bien placés 3 mal placés

Coup (7/12). Saisir 4 caractères parmi A B C D ? ABCD

0 bien placés 4 mal placés

Coup (8/12). Saisir 4 caractères parmi A B C D ? CCDB

1 bien placés 2 mal placés

Coup (9/12). Saisir 4 caractères parmi A B C D ? ACCD

0 bien placés 3 mal placés

Coup (10/12). Saisir 4 caractères parmi A B C D ? ADCB

1 bien placés 3 mal placés

Coup (11/12). Saisir 4 caractères parmi A B C D ? DCAB

0 bien placés 4 mal placés

Coup (12/12). Saisir 4 caractères parmi A B C D ? CDAB

2 bien placés 2 mal placés

P E R D U : Vous avez perdu

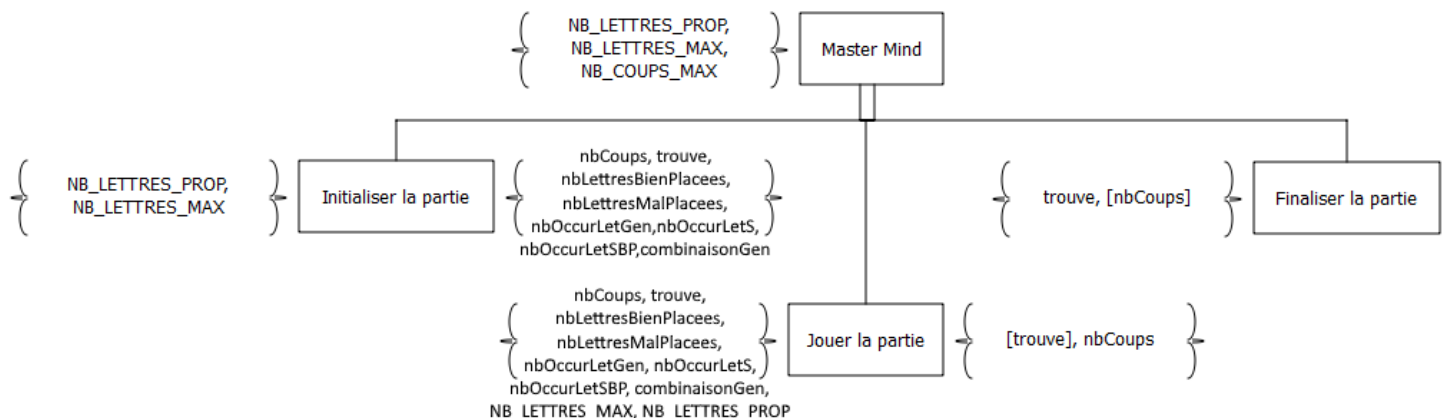
Propriété(s)

- On considère que le joueur écrit une combinaison de 4 lettres, ni plus ni moins toutes en majuscules et qu'il n'utilise que les lettres A, B, C, D
- Le joueur a 12 tentatives pour gagner sinon il perd.
- Si le joueur met plus que 4 lettres, on ne prendra que les 4 premières lettres

3.2 Spécifications complémentaires = extensions traitées

Aucune extension n'a été traitée.

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



4.1 Initialiser la partie

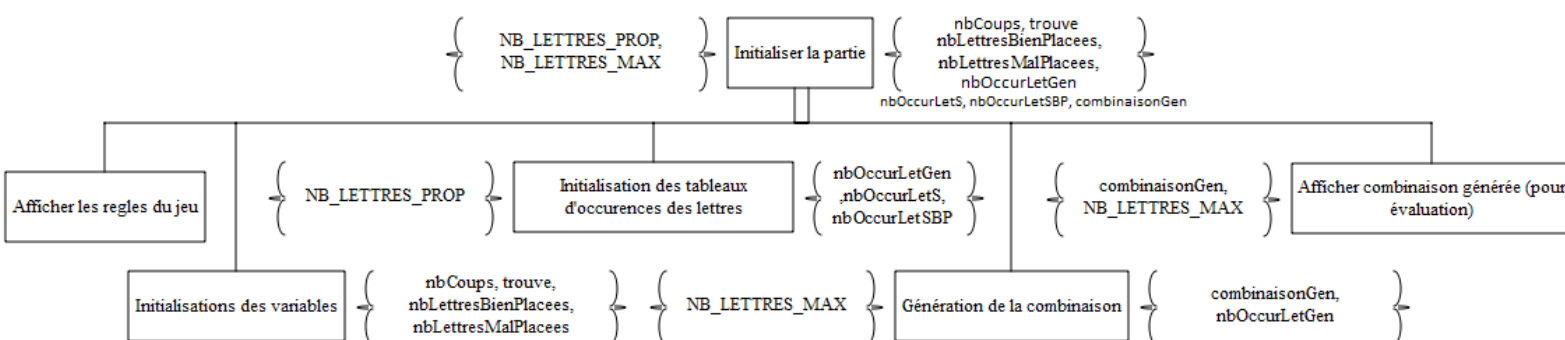
4.1.1 But de l'action

On affiche les règles du jeu au joueur puis on initialise les variables ainsi que les tableaux d'occurrences des lettres. La combinaison est ensuite générée puis affichée (uniquement pour l'évaluation de la SAE).

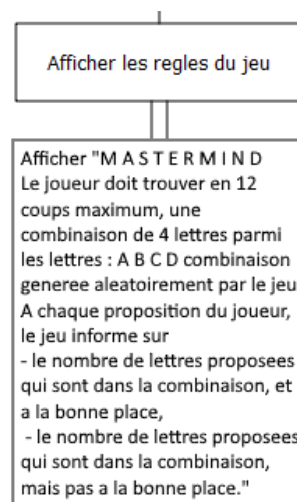
4.1.2 Stratégie de l'algorithme mise en oeuvre

L'algorithme va afficher les règles du jeu puis initialiser les variables pour compter le nombre de coups ainsi que le nombre de lettres bien/mal placés, le booléen trouve est également initialisé en cas de défaite. Une boucle itérative permet d'initialiser les différents tableaux d'occurrences du nombre de lettres saisies (combien de A, combien de B, combien de C et combien de D) à 0. A partir d'une boucle itérative, puis d'un switch, on utilise la fonction random du module game-tools de Patrick Etcheverry afin de générer un entier aléatoire entre 1 et 4 qui va être transformé en lettre (A, B, C ou D) et on augmente le nombre d'occurrence de lettres dans la combinaison générée selon la lettre choisie aléatoirement. Finalement, on affiche la combinaison grâce à une structure itérative qui parcourt le tableau de caractères contenant la combinaison générée afin de pouvoir évaluer le programme.

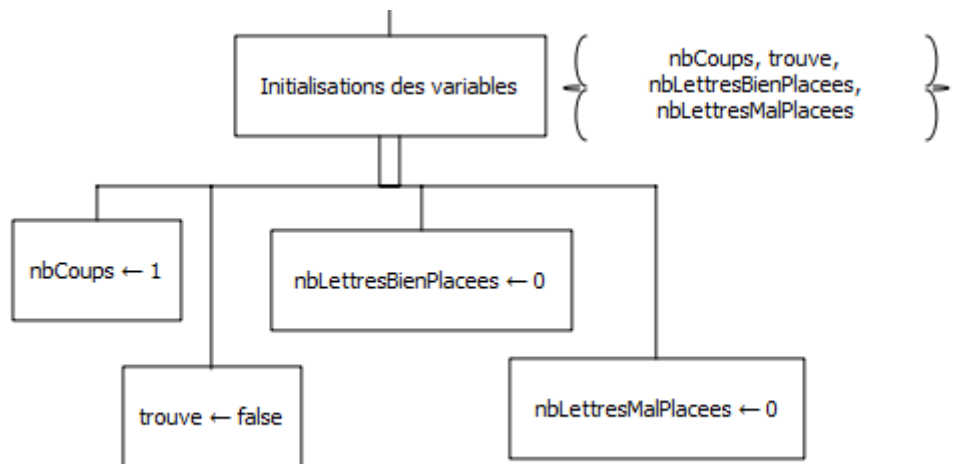
4.1.3 Algorithme



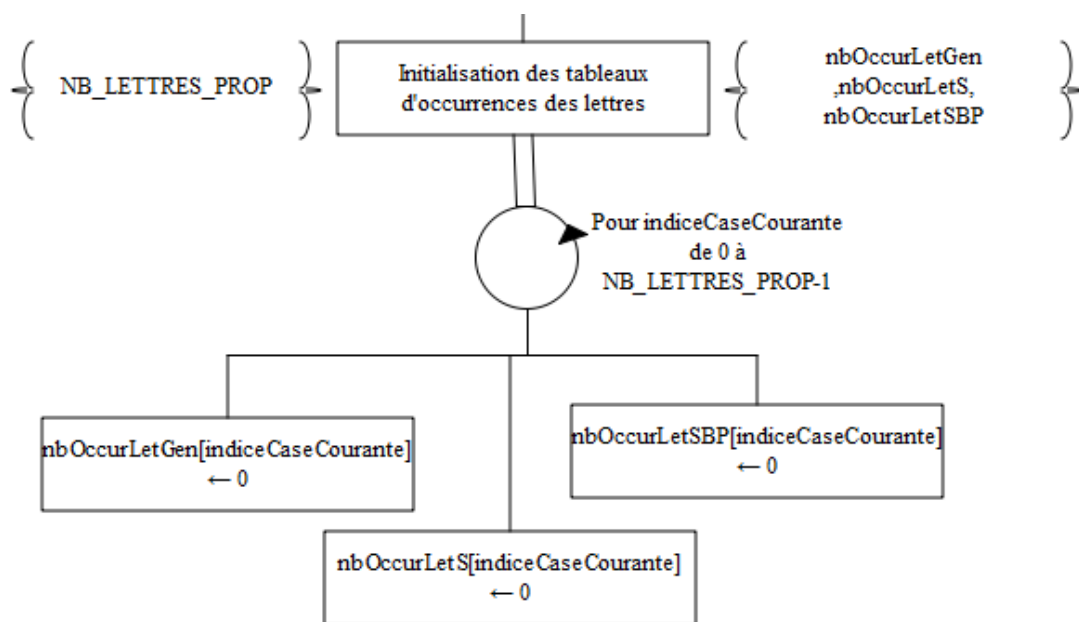
4.1.3 a) Afficher les règles du jeu



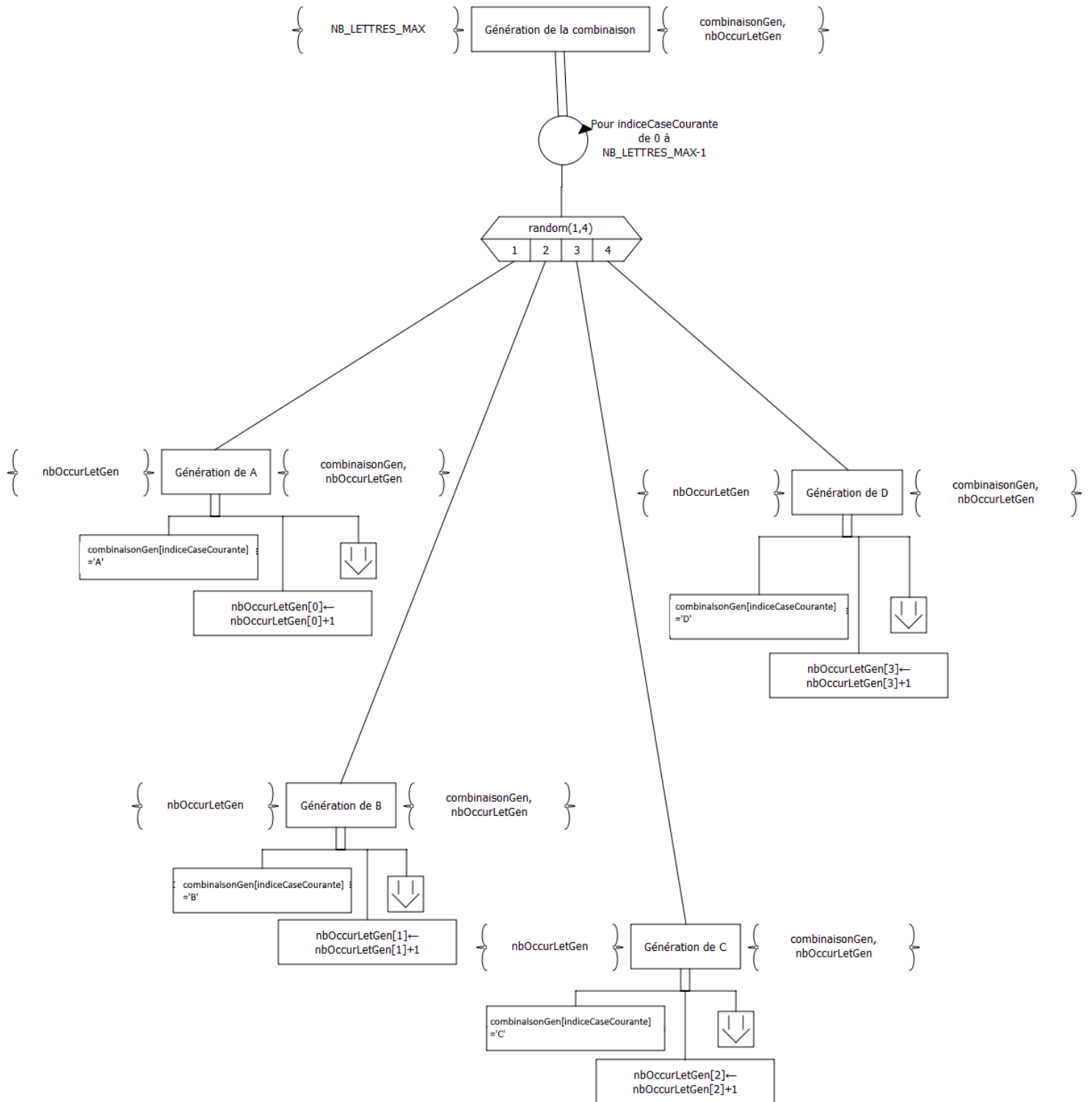
4.1.3 b) Initialisation des variables



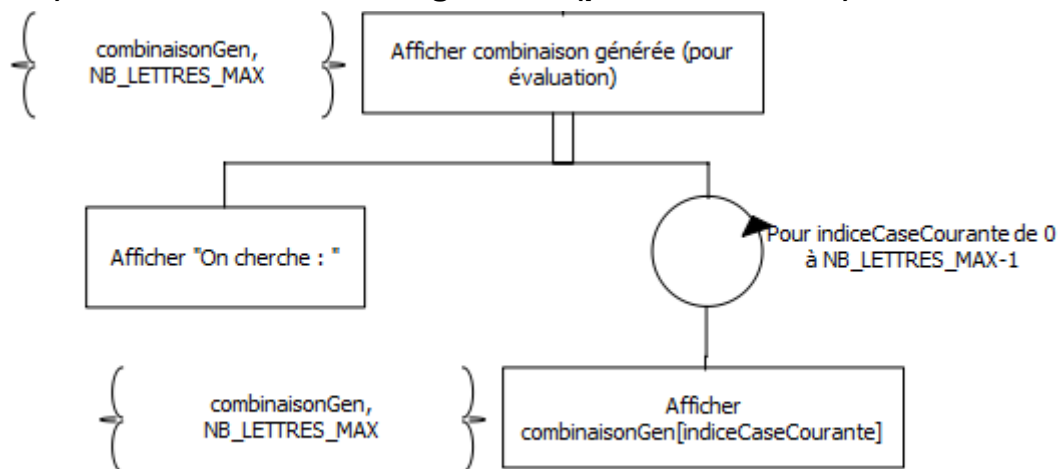
4.1.3 c) Initialisation des tableaux d'occurrences des lettres



4.1.3 d) Génération de la combinaison



4.1.3 e)Afficher combinaison générée (pour évaluation)



4.1.4 Dictionnaire des éléments associés à cet algorithme

Nom	Type	Signification
NB_LETTRES_PROP	Entier positif	Nombre de lettres proposées (A, B, C, D)
NB_LETTRES_MAX	Entier positif	Nombre maximum de lettres proposées dans une combinaison
nbCoups	Entier positif	Nombre de coups du joueur
trouve	Booléen	Détermine si le joueur a trouvé la combinaison sans perdre
nbLettresBienPlacees	Entier positif	Nombre de lettres bien placées dans la combinaison proposée par le joueur
nbLettresMalPlacees	Entier positif	Nombre de lettres mal placées dans la combinaison proposée par le joueur
nbOccurLetGen	Tableau d'entiers positifs	Nombre d'occurrences de lettres dans la combinaison générée
nbOccurLetS	Tableau d'entiers positifs	Nombre d'occurrences de lettres bien saisies dans la combinaison saisie

nbOccurLetSBP	Tableau d'entiers positifs	Nombre d'occurrences de lettres bien saisies dans la combinaison saisie
combinaisonGen	Tableau de caractères	Combinaison générée qui doit être devinée par le joueur

4.2 Jouer la partie

4.2.1 But de l'action

Cette partie permet de jouer au jeu, tout d'abord on vient saisir une combinaison qui va nous permettre de calculer le nombre d'occurrences bien ou mal placées ainsi que leur nombre. On va ensuite afficher le nombre de lettres bien et mal placées, par la suite on va gérer 2 cas qui nous feront sortir de cette partie si la combinaison est trouvée ou si le nombre de coups maximum est atteint. Pour finir, on réinitialise les variables utilisées ainsi que les tableaux.

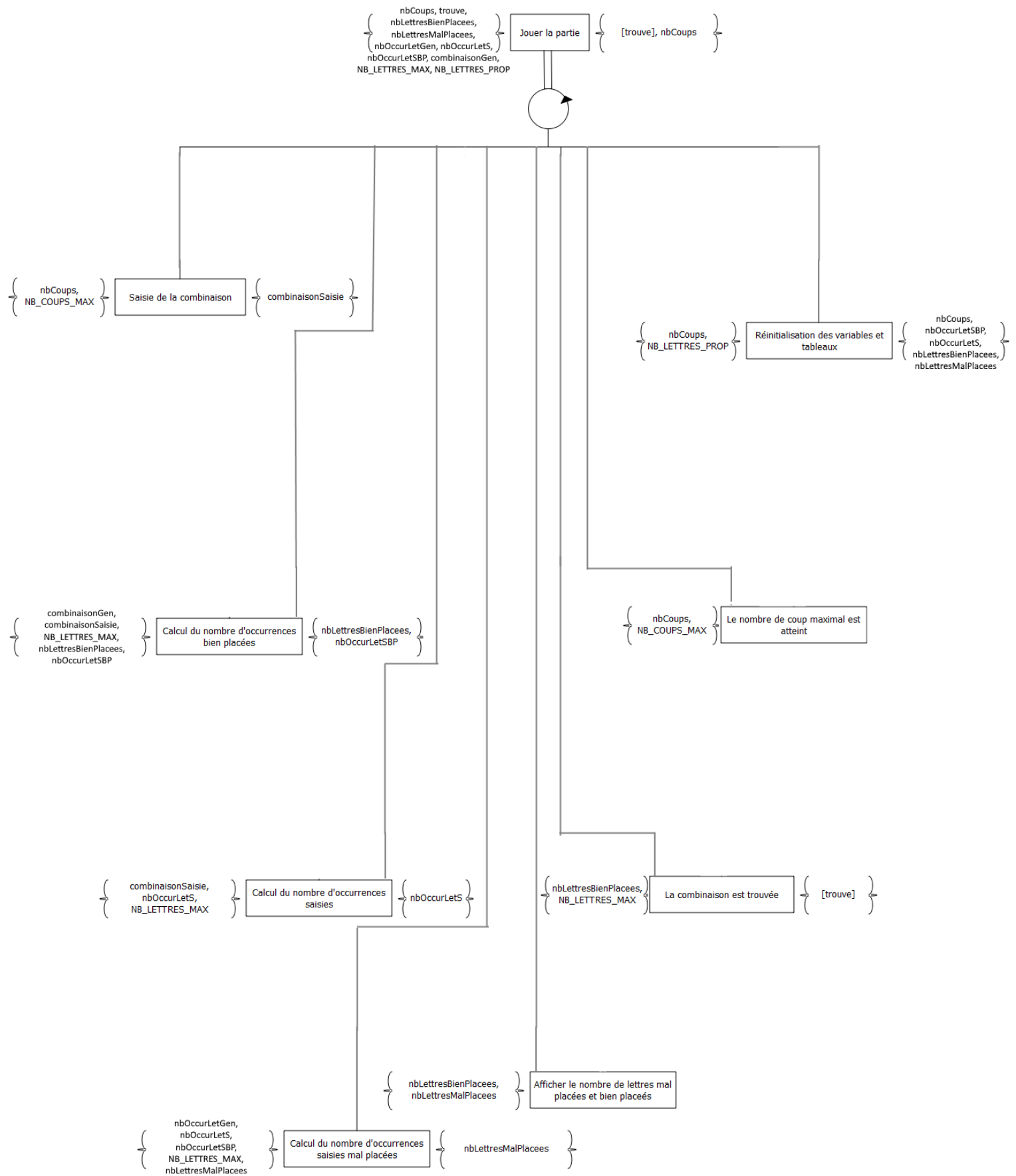
4.2.2 Stratégie de l'algorithme mise en oeuvre

Dans un premier temps, on entre une combinaison. Ensuite, on calcule le nombre d'occurrences bien placées à l'aide d'une structure itérative parcourant la combinaison puis une condition déterminant les bonnes lettres dans la combinaison saisie. Si la condition est vérifiée, on incrémente le nombre de lettres bien placées et le tableau d'occurrences des lettres à l'aide d'un switch (pour savoir s'il s'agit d'un A, d'un B, d'un C ou d'un D). Ensuite, on calcule le nombre d'occurrences saisies à l'aide d'une structure itérative et d'un switch vérifiant si les lettres sont bien A, B, C et D. Pour finir, on détermine le nombre de lettres mal placées : encore à l'aide d'une structure itérative et d'une condition, on calcule le nombre de lettres mal placées grâce à un calcul entre deux tables d'occurrences.

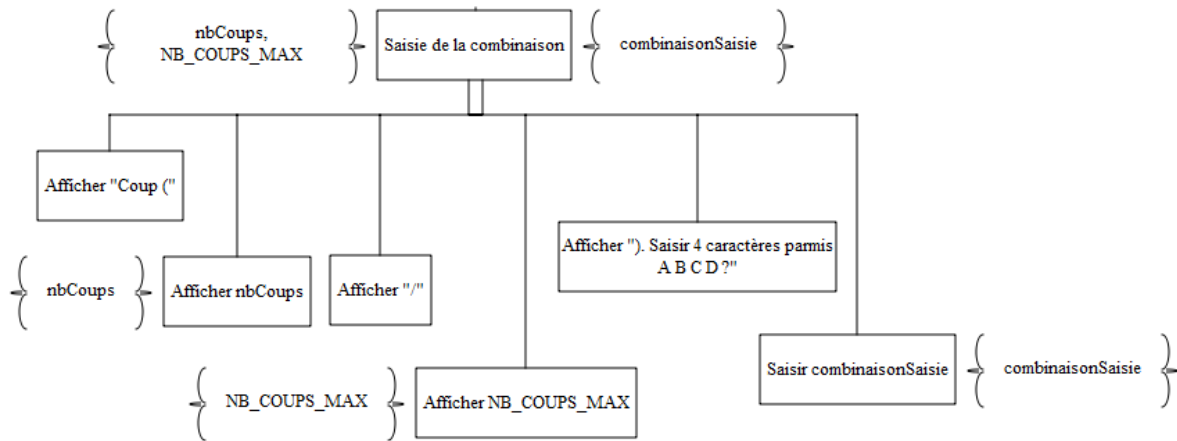
Dans un deuxième temps on va afficher le nombre de lettres mal placées et bien placées que l'on vient de calculer. Par la suite, on traite les 2 cas qui nous feront sortir de la partie: soit le nombre de coups maximum est atteint, soit la combinaison est trouvée (le booléen trouve prend donc "vrai" et l'on sort de la partie).

Dans un dernier temps, on réinitialise les tableaux d'occurrences des lettres saisies et saisies bien placées (grâce à une structure itérative) ainsi que le nombre de lettres bien et mal placées afin de recalculer le nombre de lettres bien et mal placées lors de la prochaine combinaison saisie par le joueur. Le nombre de coups est incrémenté de 1 puisque l'on passe à une autre tentative.

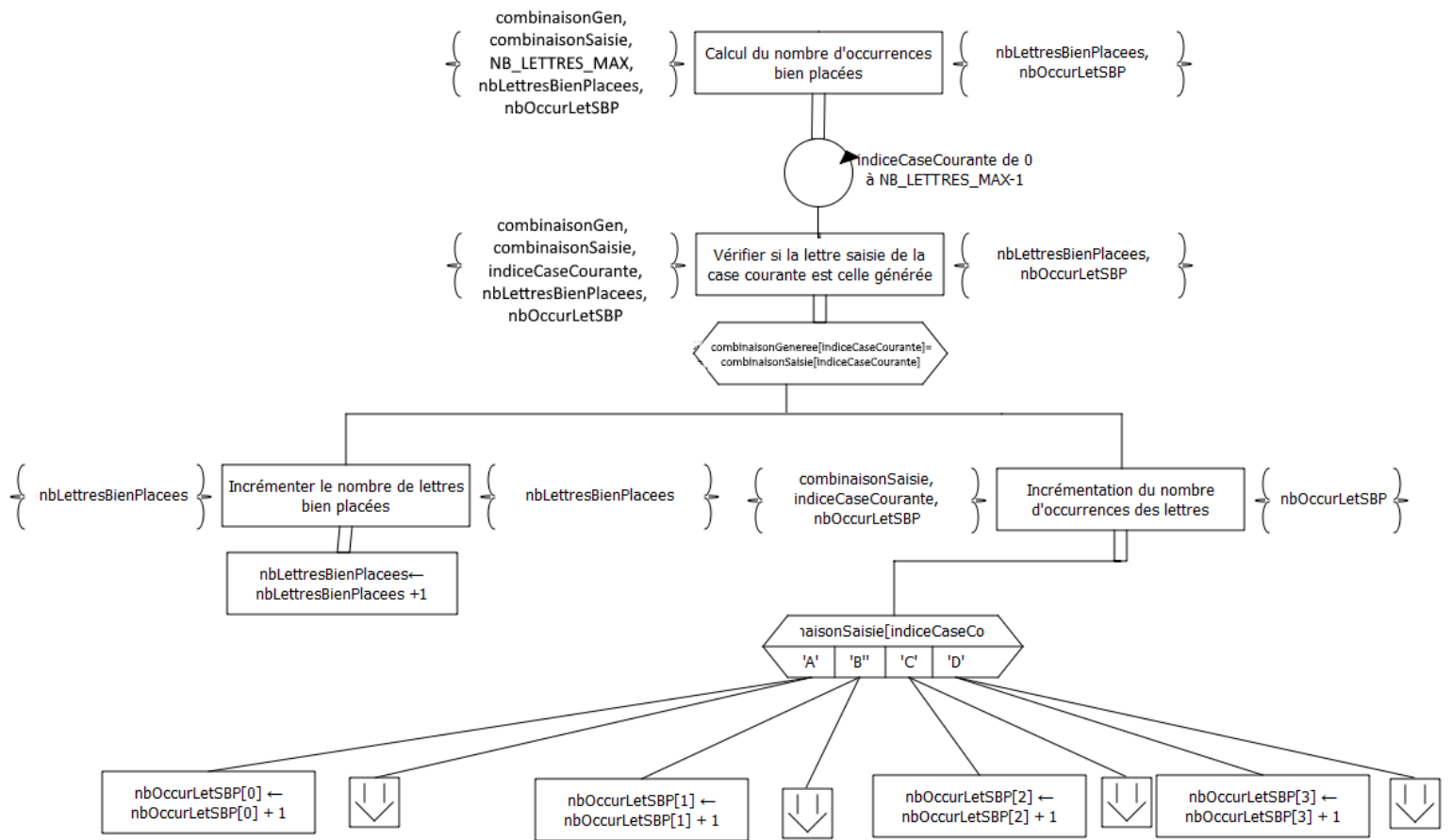
4.2.3 Algorithme



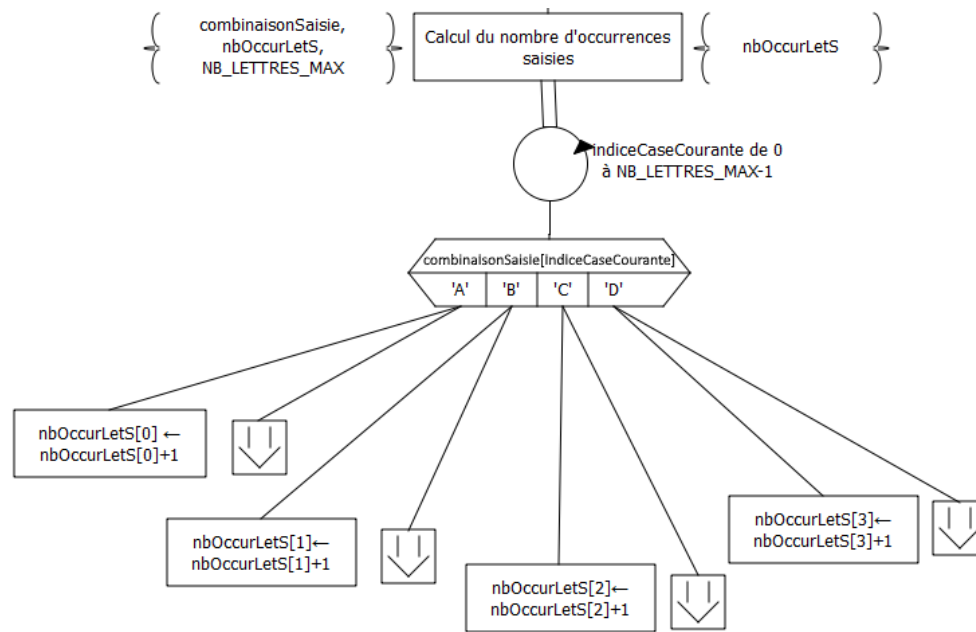
4.2.3 a) Saisir la combinaison



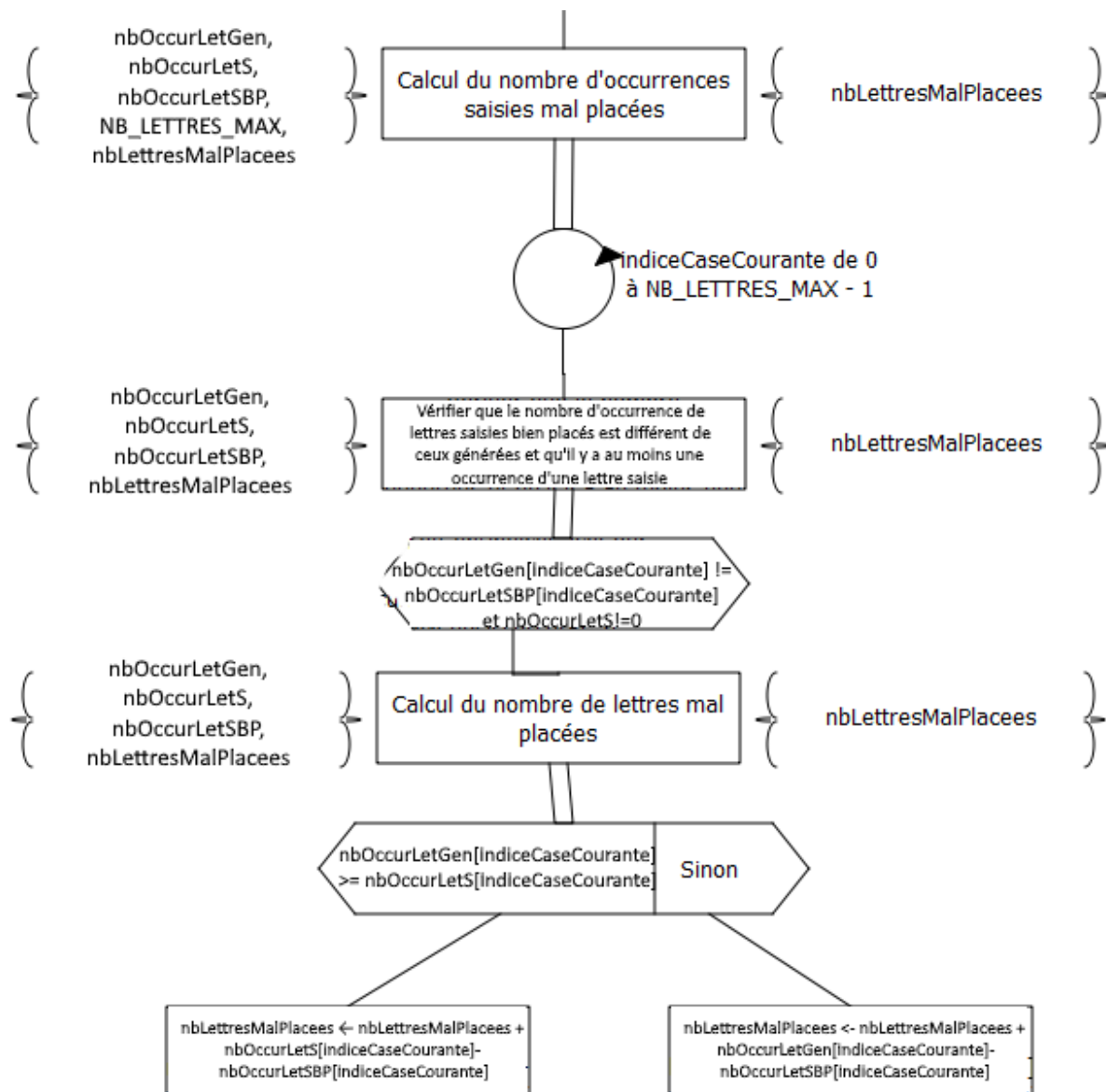
4.2.3 b) Calcul du nombre d'occurrences bien placées



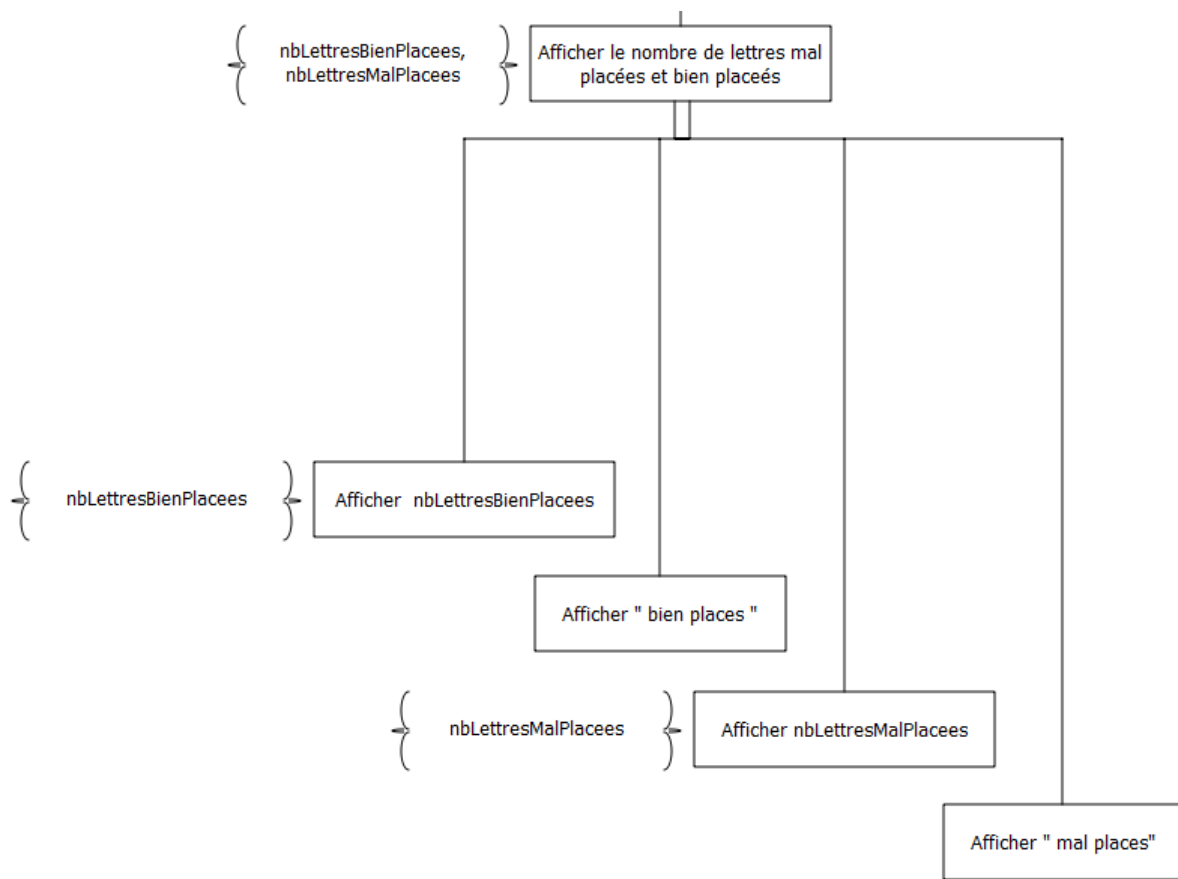
4.2.3 c) Calcul du nombre d'occurrences saisies



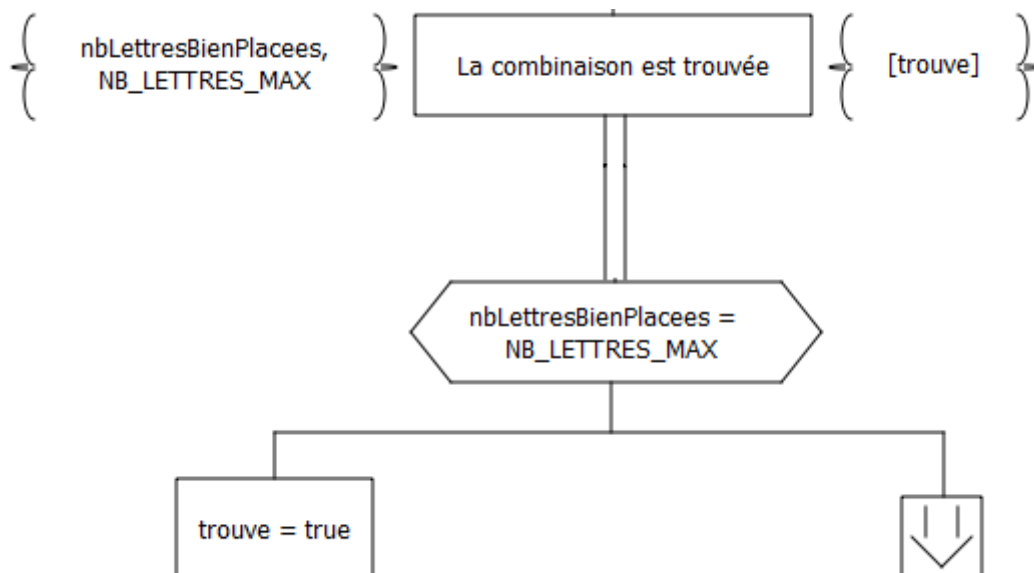
4.2.3 d) Calcul du nombre d'occurrences saisies mal placées



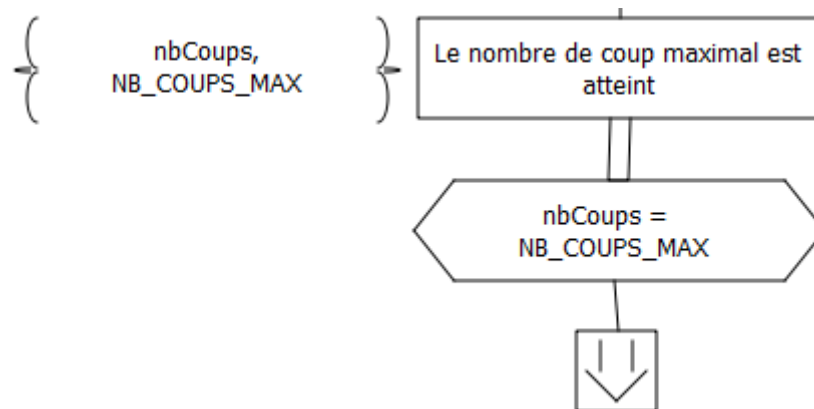
4.2.3 e) Afficher le nombre de lettres mal placées et bien placées



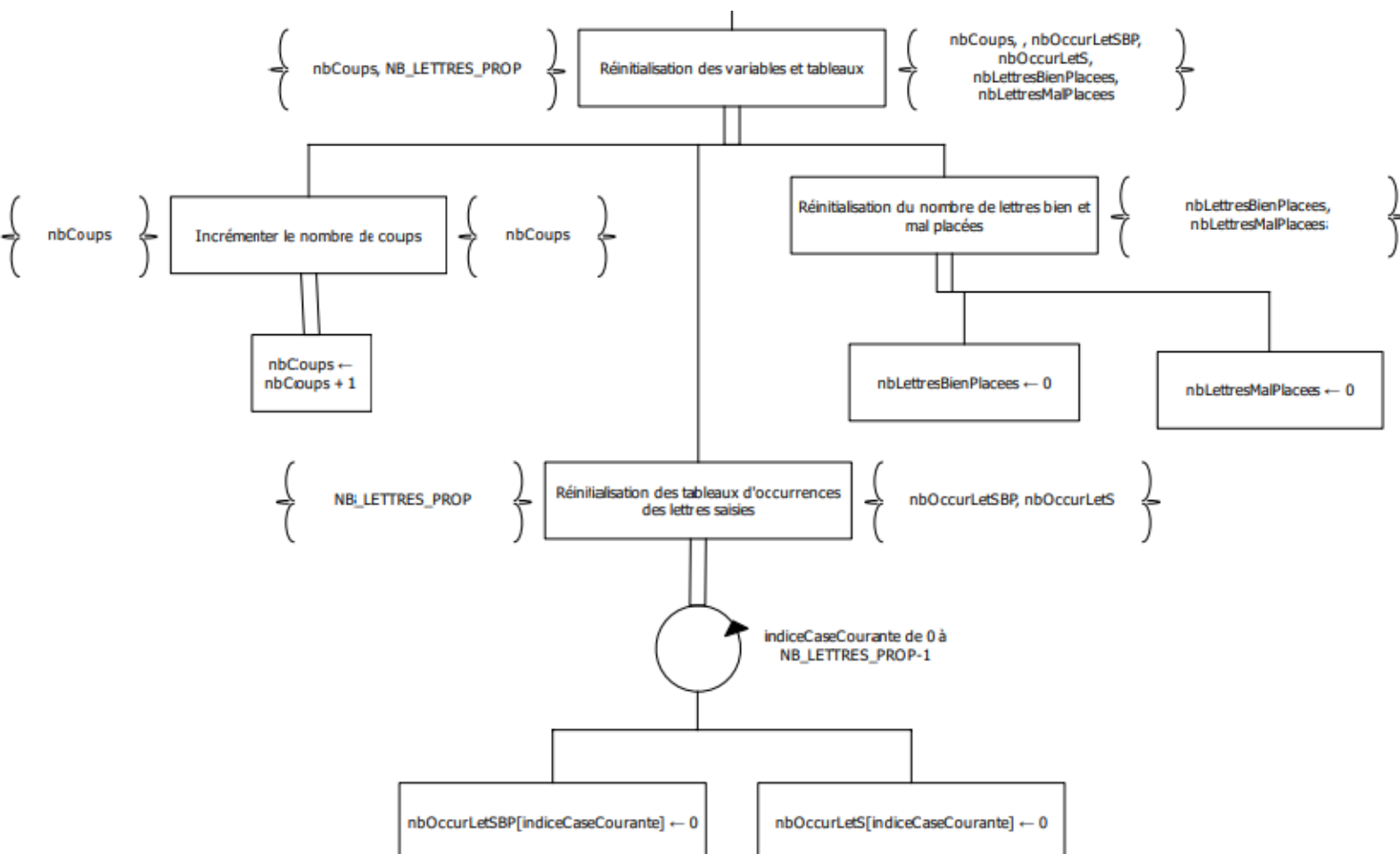
4.2.3 f) La combinaison est trouvée



4.2.3 g) Le nombre de coup maximal est atteint



4.2.3 h) Réinitialisation des variables et tableaux



4.2.4 Dictionnaire des éléments associés à cet algorithme

Nom	Type	Signification
nbCoups	Entier positif	Nombre de coups du joueur
trouve	Booléen	Détermine si le joueur a trouvé la combinaison sans perdre
nbLettresBienPlacees	Entier positif	Nombre de lettres bien placées dans la combinaison proposée par le joueur
nbLettresMalPlacees	Entier positif	Nombre de lettres mal placées dans la combinaison proposée par le joueur
nbOccurLetGen	Tableau d'entiers positifs	Nombre d'occurrences de lettres dans la combinaison générée
nbOccurLetS	Tableau d'entiers positifs	Nombre d'occurrences de lettres dans la combinaison saisie
nbOccurLetSBP	Tableau d'entiers positifs	Nombre d'occurrences de lettres bien saisies dans la combinaison saisie
combinaisonGen	Tableau de caractères	Combinaison générée qui doit être deviné par le joueur
combinaisonSaisie	Tableau de caractères	Combinaison saisie par le joueur qui doit deviner la combinaison générée
NB_LETTRES_MAX	Entier positif	Nombre maximum de lettres proposées dans une combinaison
NB_LETTRES_PROP	Entier positif	Nombre de lettres proposées (A, B, C, D)

4.3 Finaliser la partie

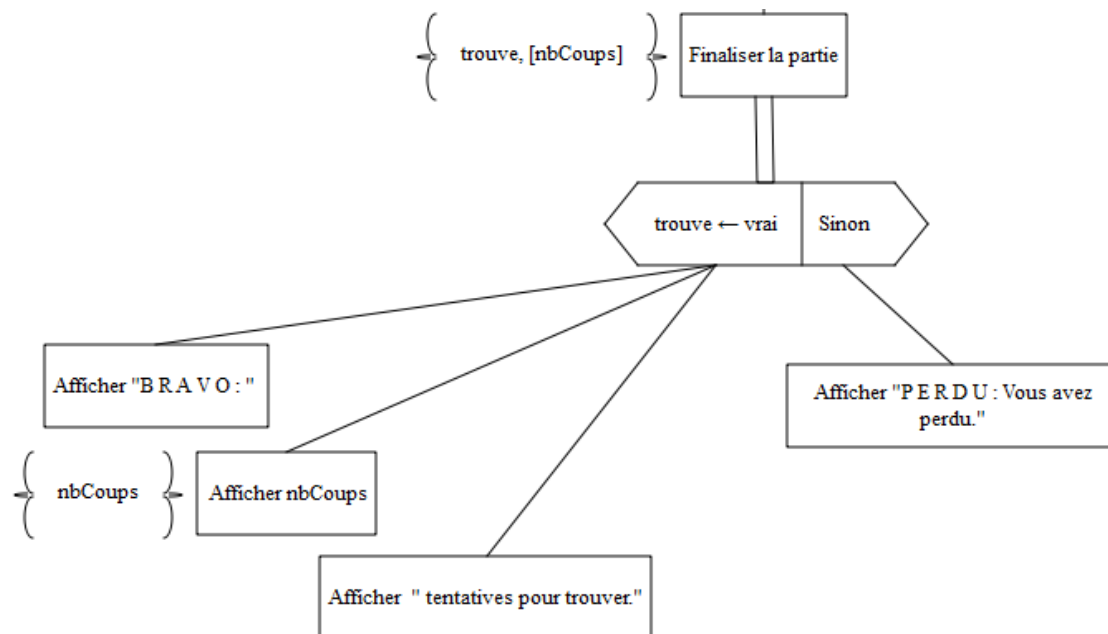
4.3.1 But de l'action

Finaliser la partie en affichant le nombre de coups ainsi qu'un message de félicitation en cas de victoire ou un message indiquant que le joueur a perdu.

4.3.2 Stratégie de l'algorithme mise en oeuvre

Utilisation d'une structure alternative "si/sinon" utilisant un booléen (trouve) qui permet de vérifier si le joueur a trouvé la bonne combinaison. On affiche ensuite, "B R A V O : nbCoups tentatives pour trouver" si le joueur a gagné, sinon on lui affiche "P E R D U : Vous avez perdu;"

4.3.3 Algorithme



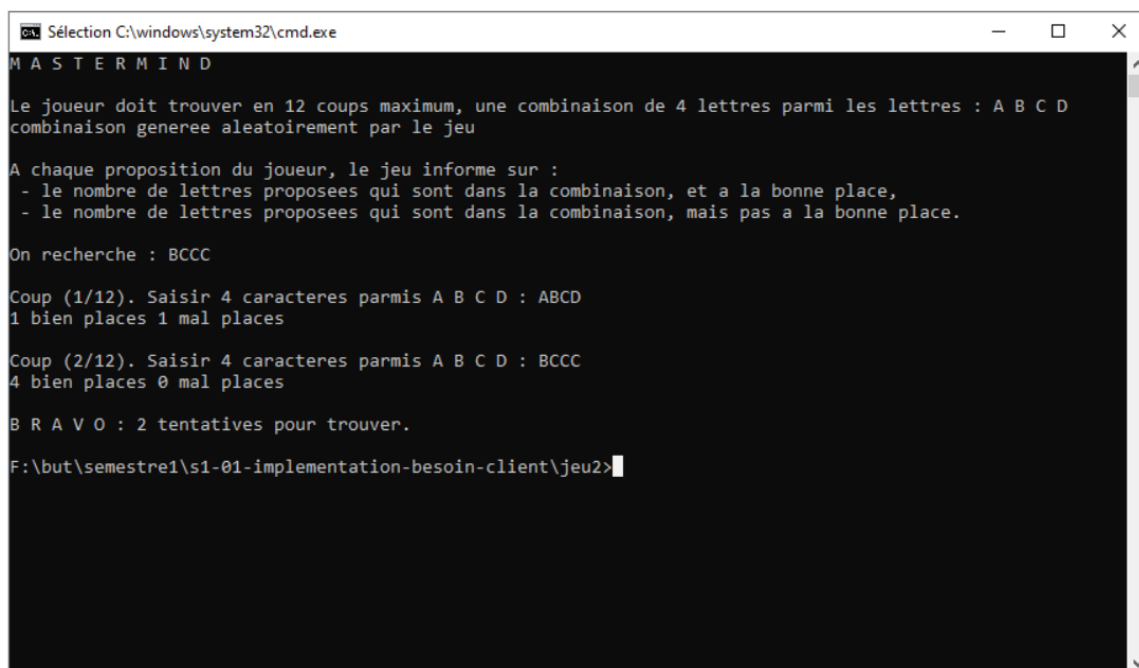
4.3.4 Dictionnaire des éléments associés à cet algorithme

Nom	Type	Signification
trouve	Booléen	Détermine si le joueur a trouvé la combinaison sans perdre
nbCoups	Entier positif	Nombre de coups du joueur

5. Traces d'exécution

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

Figure 1 : Comportement lié au scénario nominatif : Le joueur gagne



```
Sélection C:\windows\system32\cmd.exe
MASTER MIND

Le joueur doit trouver en 12 coups maximum, une combinaison de 4 lettres parmi les lettres : A B C D
combinaison generee aleatoirement par le jeu

A chaque proposition du joueur, le jeu informe sur :
- le nombre de lettres proposees qui sont dans la combinaison, et a la bonne place,
- le nombre de lettres proposees qui sont dans la combinaison, mais pas a la bonne place.

On recherche : BCCC

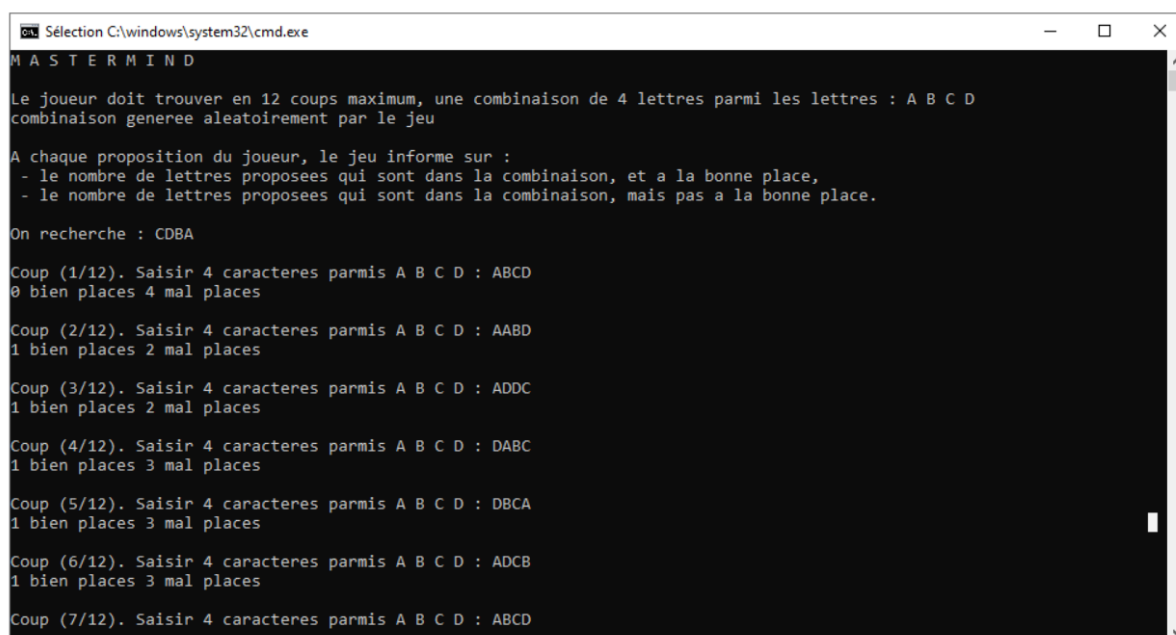
Coup (1/12). Saisir 4 caracteres parmi A B C D : ABCD
1 bien places 1 mal places

Coup (2/12). Saisir 4 caracteres parmi A B C D : BCCC
4 bien places 0 mal places

B R A V O : 2 tentatives pour trouver.

F:\but\semestre1\s1-01-implementation-besoin-client\jeu2>
```

Figure 2 : Comportement lié au scénario alternatif : Le joueur perd



```
Sélection C:\windows\system32\cmd.exe
MASTER MIND

Le joueur doit trouver en 12 coups maximum, une combinaison de 4 lettres parmi les lettres : A B C D
combinaison generee aleatoirement par le jeu

A chaque proposition du joueur, le jeu informe sur :
- le nombre de lettres proposees qui sont dans la combinaison, et a la bonne place,
- le nombre de lettres proposees qui sont dans la combinaison, mais pas a la bonne place.

On recherche : CDBA

Coup (1/12). Saisir 4 caracteres parmi A B C D : ABCD
0 bien places 4 mal places

Coup (2/12). Saisir 4 caracteres parmi A B C D : AABD
1 bien places 2 mal places

Coup (3/12). Saisir 4 caracteres parmi A B C D : ADDC
1 bien places 2 mal places

Coup (4/12). Saisir 4 caracteres parmi A B C D : DABC
1 bien places 3 mal places

Coup (5/12). Saisir 4 caracteres parmi A B C D : DBCA
1 bien places 3 mal places

Coup (6/12). Saisir 4 caracteres parmi A B C D : ADCB
1 bien places 3 mal places

Coup (7/12). Saisir 4 caracteres parmi A B C D : ABCD
1 bien places 3 mal places

F:\but\semestre1\s1-01-implementation-besoin-client\jeu2>
```

```
Sélection C:\windows\system32\cmd.exe
1 bien places 3 mal places
Coup (5/12). Saisir 4 caracteres parmi A B C D : DBCA
1 bien places 3 mal places
Coup (6/12). Saisir 4 caracteres parmi A B C D : ADCB
1 bien places 3 mal places
Coup (7/12). Saisir 4 caracteres parmi A B C D : ABCD
0 bien places 4 mal places
Coup (8/12). Saisir 4 caracteres parmi A B C D : CCDB
1 bien places 2 mal places
Coup (9/12). Saisir 4 caracteres parmi A B C D : ACCD
0 bien places 3 mal places
Coup (10/12). Saisir 4 caracteres parmi A B C D : ADCB
1 bien places 3 mal places
Coup (11/12). Saisir 4 caracteres parmi A B C D : DCAB
0 bien places 4 mal places
Coup (12/12). Saisir 4 caracteres parmi A B C D : CDAB
2 bien places 2 mal places
P E R D U : Vous avez perdu.
F:\but\semestre1\s1-01-implementation-besoin-client\jeu2>_
```

6. Remarques

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

Nous avons aidé des étudiants afin de calculer le nombre de lettres mal placées.

Il faut ajouter le module game-tools de Patrick Etcheverry disponible sur [Github](#) dans le dossier du main.cpp

7. Code C++

Fichier main.cpp joint au dossier avec en-tête certifiant l'originalité du code produit. Il faut cependant ajouter le module game-tools de Patrick Etcheverry disponible sur [Github](#) dans le dossier du main.cpp.