

1 Equipe

MAURICE Alexandre TD I – TP2

DARGAZANLI Nicolas TD I – TP1

2 Exercice traité : Pack 09

Jeu 2 : Jeu du pendu

3 Rappel des spécifications du programme

On demande d'afficher :

Avant la phase de jeu :

- Le nom du jeu
- Les thèmes à choisir par le joueur qui sont associés à un nombre

Pendant la phase de jeu :

- Le nom du jeu
- Un mot constitué de tirets qui devra se révéler au fur et à mesure de la partie
- Le nombre de tentatives erronées restantes
- Un message indiquant au joueur si la lettre qu'il a proposée est ou non dans le mot
- Si le joueur propose une lettre déjà proposée, un message le lui en informe. Si la lettre était bien dans le mot, il ne perd pas de points, cependant, il en perd si elle n'était pas dans le mot.

Après la phase de jeu :

- Le résultat de la partie (victoire ou défaite)
- Le mot qui était à deviner

Les extensions sont soulignées.

3.1 Spécifications initiales

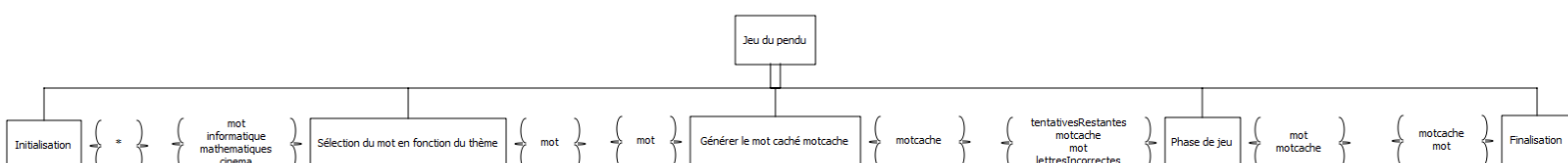
Ce programme devra, après avoir affiché le nom du jeu et invité l'utilisateur à saisir un thème correct, lui demander une lettre à saisir. Tout au long du jeu, un mot caché (constitué de tirets) sera affiché pour aider le joueur à deviner le mot. Au début, il pourra connaître le nombre de lettres du mot en comptant le nombre de tirets, et par la suite, le mot caché servira aussi à deviner le mot complet à l'aide des lettres déjà dévoilées. La fin de la partie se déclenche selon deux événements : ou le joueur n'a plus de tentatives, il a donc perdu, ou le mot qu'on lui affiche pour l'aider est strictement égal au mot qu'il doit chercher, auquel cas il emporte la partie.

3.2 Spécifications complémentaires = extensions traitées

Nous avons traité les deux extensions proposées. (Messages d'information si la saisie a déjà été faite, et si cette dernière était correcte ou erronée)

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

Algorithme général :



Nous détaillons ci-dessous chacun des sous-problèmes de l'algorithme général

L'astérisque (*) correspond aux résultats du sous-problème initialisation qui sont : nbMotTheme, mot, tentativesRestantes, lettresIncorrectes, ainsi que les trois tableaux de thèmes informatique, mathématiques et cinema.

4.1 Initialisation

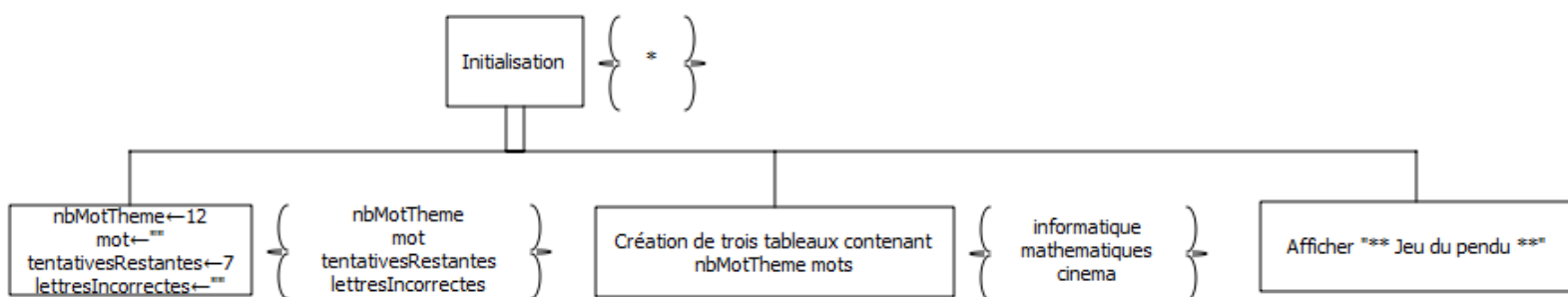
4.1.1 But de l'action

Initialise le jeu en prenant en compte les règles : le nombre de mots par thème, le nombre de tentatives ainsi que la création des trois tableaux de thèmes. Nous avons choisi ici les thèmes « Informatique », « Mathématiques » et « Cinéma ». On affiche aussi le titre du jeu pour le prochain sous-problème algorithmique : la sélection et vérification du thème.

4.1.2 Stratégie de l'algorithme mise en oeuvre

L'algorithme est une simple décomposition séquentielle qui initialise les variables nécessaires au fonctionnement du jeu, ainsi que la création des trois tableaux de thèmes à choisir par le joueur.

4.1.3 Algorithme



L'astérisque (*) correspond aux résultats du sous-problème initialisation qui sont : nbMotTheme, mot, tentativesRestantes, lettresIncorrectes, ainsi que les trois tableaux de thèmes informatique, mathématiques et cinema.

4.1.4 Dictionnaire des éléments associés à cet algorithme

Nom	Type	Signification
nbMotTheme	entier	Nombre de mots dans chaque thème.
mot	chaîne de caractères	Mot à deviner.
tentativesRestantes	entier	Nombre de tentatives incorrectes restantes.
lettresincorrectes	chaîne de caractères	Toutes les lettres incorrectes déjà proposées par le joueur.
informatique	tableau de chaînes de caractères de taille nbMotTheme	Tableau contenant des mots à deviner potentiels pour le thème « Informatique ».
mathematiques	tableau de chaînes de caractères de taille nbMotTheme	Tableau contenant des mots à deviner potentiels pour le thème « Mathematiques ».
cinema	tableau de chaînes de caractères de taille nbMotTheme	Tableau contenant des mots à deviner potentiels pour le thème « Cinema ».

4.2 Sélection du mot en fonction du thème et vérification

4.2.1 But de l'action

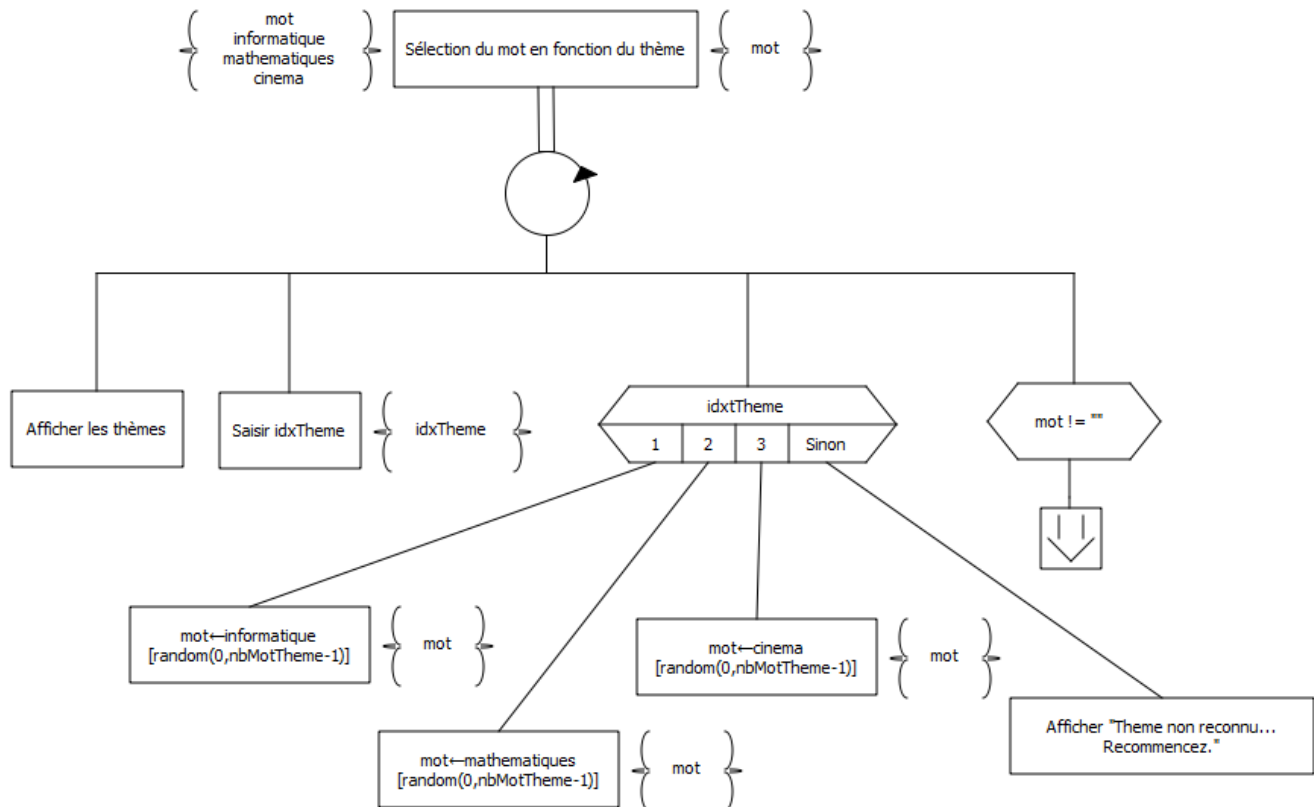
L'utilisateur doit entrer un chiffre correspondant au thème. À partir de ce thème, on génère alors le mot qu'il devra deviner.

4.2.2 Stratégie de l'algorithme mise en oeuvre

Cet algorithme est une décomposition itérative parmi laquelle se trouve un switch où l'expression est la variable « idxTheme », l'indice (au sens linguistique ici, c'est-à-dire un indice donnant une indication

au joueur pour réduire l'éventail des possibilités de mots à deviner) pour le thème de mots. La vérification est alors assurée par la chaîne de caractères mot, en effet, le mot doit se générer si la saisie est correcte. On ne sort de la boucle que quand le mot n'est plus vide. Un message d'erreur doit être affiché si la saisie est incorrecte.

4.2.3 Algorithme



4.2.4 Dictionnaire des éléments associés à cet algorithme

Nom	Type	Signification
idxTheme	entier	Chiffre correspondant à un thème saisi par le joueur.
mot	chaîne de caractères	Mot qui prend alors une chaîne de caractères aléatoire parmi le tableau sélectionné.
informatique	tableau de chaînes de caractères de taille nbMotTheme	Tableau contenant des mots à deviner potentiels pour le thème « Informatique ».
mathematiques	tableau de chaînes de caractères de taille nbMotTheme	Tableau contenant des mots à deviner potentiels pour le thème « Mathématiques ».
cinema	tableau de chaînes de caractères de taille nbMotTheme	Tableau contenant des mots à deviner potentiels pour le thème « Cinema ».

4.3 Générer le mot caché

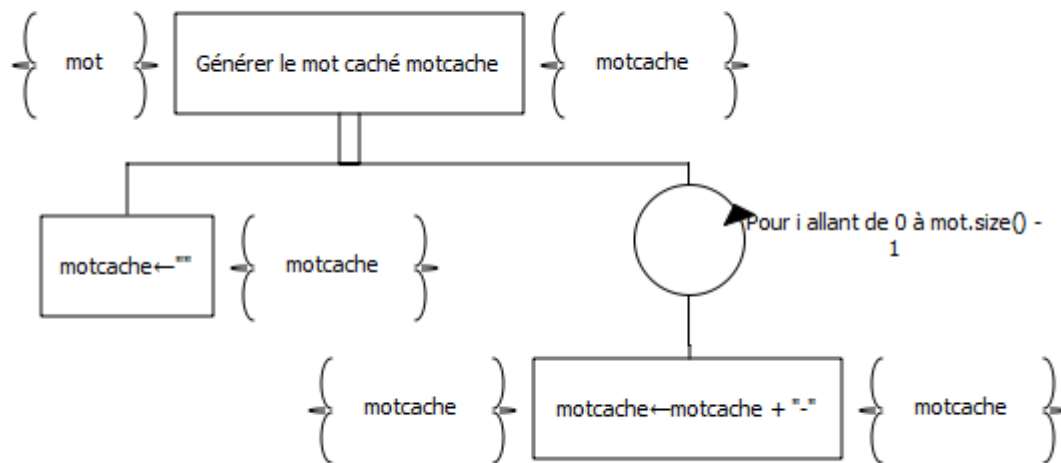
4.3.1 But de l'action

Crée le mot à afficher. On crée un mot constitué d'autant de tirets que la taille du mot à chercher

4.3.2 Stratégie de l'algorithme mise en oeuvre

Décomposition séquentielle suivie d'une décomposition itérative afin de créer le mot qui va être affiché.

4.3.3 Algorithme



4.3.4 Dictionnaire des éléments associés à cet algorithme

Nom	Type	Signification
mot	chaîne de caractères	Mot qui prend alors une chaîne de caractères aléatoire parmi le tableau sélectionné.
motcache	chaîne de caractères	Mot qui sera affiché en tant qu'aide pour le joueur. Il est pour l'instant constitué de tirets.

4.4 Phase de jeu

4.4.1 But de l'action

Constitue le corps du programme. Cette action doit dérouler la partie en affichant correctement le nom du jeu, les tentatives restantes, le mot caché qui servira d'indice (toujours au sens linguistique) pour le joueur. Le joueur est invité à chaque fois à saisir une lettre, et obtient à chaque fois un retour sur ce qu'il a saisi. La phase de jeu se termine quand le joueur n'a plus de tentatives erronées restantes ou qu'il a deviné le mot (c'est-à-dire que le mot qui est affiché en tant qu'aide est exactement égal au mot qu'il doit trouver).

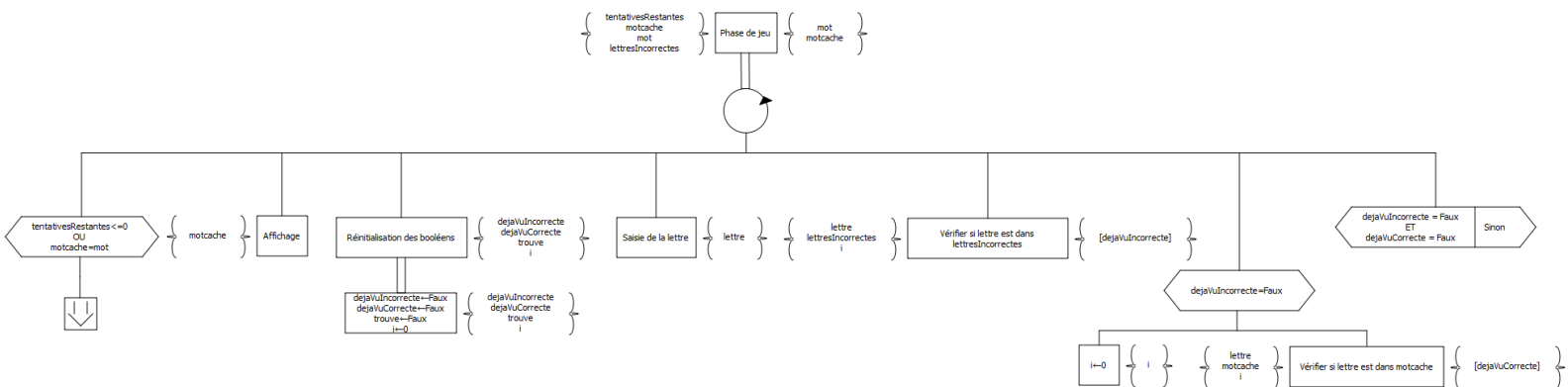
4.4.2 Stratégie de l'algorithme mise en oeuvre

Cet algorithme est une décomposition itérative. Elle est d'abord constituée de la condition d'arrêt : le joueur n'a plus de tentatives ou il a trouvé le mot. Si ce n'est pas le cas, le problème « Affichage » prend la suite. On réinitialise ensuite les booléens qui permettent au programme d'assurer le fonctionnement des extensions. Le joueur saisit ensuite la lettre, qui sera suivi d'un algorithme vérifiant si la lettre proposée était déjà proposée par le joueur et était incorrecte. Si cette condition n'est pas vérifiée, le sous problème qui vérifie si la lettre est correcte prend la suite. Finalement, une condition si la lettre n'a jamais été proposée suivie d'un sinon assure l'affichage du retour et éventuellement si la lettre est correcte, change le mot caché affiché.

Les sous problèmes sont soulignés.

4.4.3 Algorithme général de la phase de jeu

En raison de la taille conséquente de cette partie de l'algorithme, nous sommes obligés de le rediviser en sous-problèmes. Nous ne représentons alors ici que les grands sous-problèmes.



4.4.4 Dictionnaire des éléments associés à cet algorithme

Nom	Type	Signification
tentativesRestantes	entier	Nombre de tentatives erronées restantes au joueur.
motcache	chaîne de caractères	Mot qui sera affiché en tant qu'aide pour le joueur. Il est pour l'instant constitué de tirets.
mot	chaîne de caractères	Mot à deviner par le joueur.
dejaVuIncorrecte	booléen	Vrai si la lettre a déjà été saisie et est incorrecte.
dejaVuCorrecte	booléen	Vrai si la lettre a déjà été saisie et est correcte.
trouve	booléen	Vrai si la lettre est présente dans le mot.
lettresIncorrectes	chaîne de caractères	Chaîne contenant toutes les lettres incorrectes déjà saisies par le joueur.

4.5 Affichage

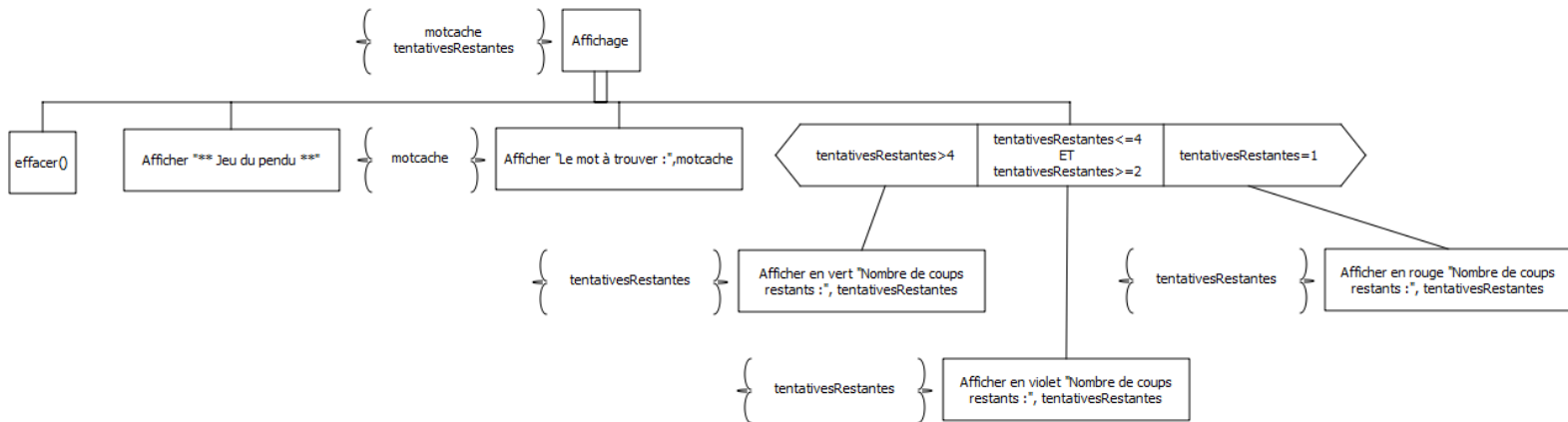
4.5.1 But de l'action

Efface ce qui avait été écrit précédemment et réaffiche le nom du jeu ainsi que le mot caché et le nombre de tentatives restantes. La couleur de ce dernier varie en fonction du nombre.

4.5.2 Stratégie de l'algorithme mise en oeuvre

Cet algorithme est une décomposition séquentielle dans laquelle se trouve une décomposition alternative pour l'affichage en couleur du nombre de tentatives.

4.5.3 Algorithme



4.5.4 Dictionnaire des éléments associés à cet algorithme

Nom	Type	Signification
tentativesRestantes	entier	Nombre de tentatives erronées restantes au joueur.
motcache	chaîne de caractères	Mot qui sera affiché en tant qu'aide pour le joueur. Il est pour l'instant constitué de tirets.

4.6 Vérifier si lettre est dans lettresIncorrectes

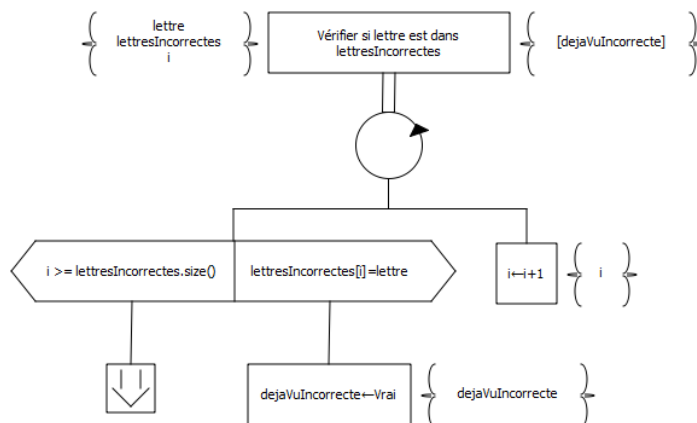
4.6.1 But de l'action

Vérifie si la lettre saisie par le joueur a déjà été saisie par le joueur et est incorrecte.

4.6.2 Stratégie de l'algorithme mise en oeuvre

Cet algorithme est une décomposition itérative dans laquelle on a deux conditions : une condition de sortie si on a parcouru toute la chaîne de caractères lettresIncorrectes, et une condition qui vérifie si la lettre saisie correspond à une lettre incorrecte déjà saisie. À chaque itération, on incrémente la variable i de 1 afin d'assurer le parcours de la chaîne de caractères lettresIncorrectes.

4.6.3 Algorithme



4.6.4 Dictionnaire des éléments associés à cet algorithme

Nom	Type	Signification
lettresIncorrectes	chaîne de caractères	Chaîne contenant toutes les lettres

		incorrectes déjà saisies par le joueur.
lettre	caractère	Lettre saisie par le joueur.

4.7 Vérifier si lettre est dans le motcache

4.7.1 But de l'action

Vérifie si la lettre saisie par le joueur est dans le mot caché. Si c'est le cas, cela veut dire que cette lettre est correcte car elle est dans le mot à deviner, et que cette lettre a déjà été saisie par le joueur, car elle a été dévoilée.

4.7.2 Stratégie de l'algorithme mise en oeuvre

Correspond exactement à l'algorithme en 4.6.2 en remplaçant toutes les occurrences de "lettresIncorrectes" par "motcache".

4.7.3 Algorithme

Cet algorithme est une répétition de l'algorithme mis en place en 4.6, sauf qu'il faut remplacer toutes les occurrences de "lettresIncorrectes" par "motcache".

4.7.4 Dictionnaire des éléments associés à cet algorithme

Nom	Type	Signification
motcache	chaîne de caractères	Mot qui est affiché en tant qu'aide pour le joueur. Il correspond dans cet algorithme aux lettres déjà saisies et correctes.
lettre	caractère	Lettre saisie par le joueur

4.8 Affichage du retour fait au joueur

4.8.1 But de l'action

Si la lettre saisie par le joueur n'a jamais été vue :

- et qu'elle est correcte, on doit révéler la lettre dans le mot affiché et lui dire qu'elle est correcte
- et qu'elle est incorrecte, on doit lui dire qu'elle est incorrecte et lui retirer une tentative

Si la lettre saisie a déjà été vue :

- et qu'elle est incorrecte, le joueur est informé qu'il a déjà saisi cette lettre et qu'elle est toujours incorrecte. On lui retire au passage un point.
- et qu'elle est correcte, on lui informe qu'il l'a déjà saisie, mais aucun point ne lui est enlevé car il n'a pas commis de faute.

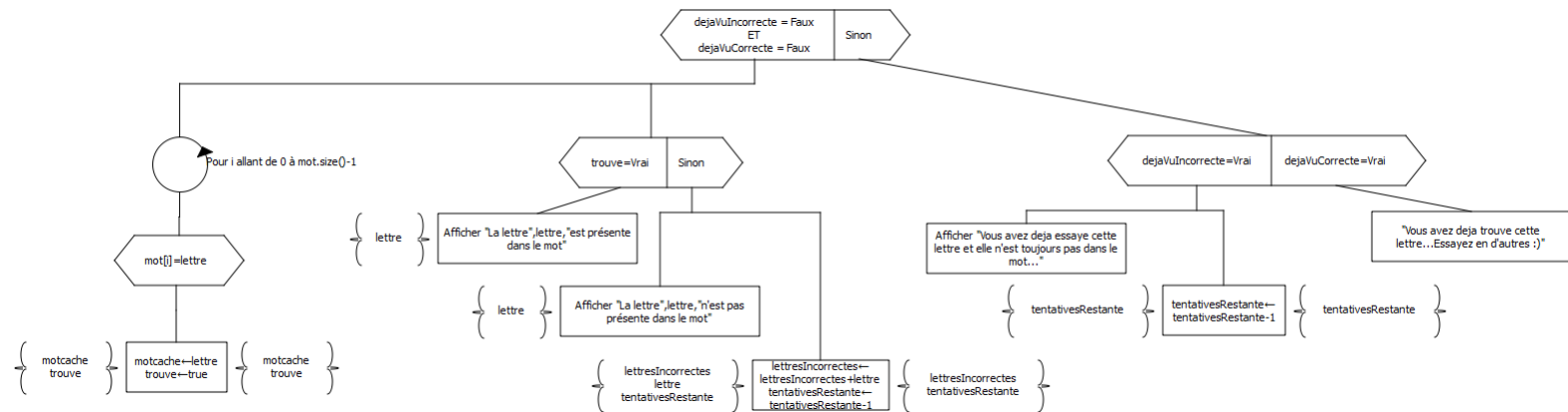
Note : cette sous-partie n'est pas un algorithme en soi. Nous l'avons ajoutée ici en temps que sous-problème mais ne doit pas être considérée comme tel.

4.8.2 Stratégie mis en oeuvre

Cette sous-partie correspond à une décomposition alternative du type si, sinon. Le si peut être traduit par « Si la lettre n'a jamais été vue ». Dans le si On effectue d'abord dans le un parcours complet avec traitement systématique de la structure à accès direct mot, en vérifiant à chaque fois si la lettre sur laquelle on se focalise correspond bien à la lettre saisie par le joueur. Si c'est le cas, on associe Vrai au booléen trouve, qui servira par la suite à afficher si la lettre est bien présente dans le mot, sinon, on ajoute la lettre dans la liste des lettresIncorrectes déjà saisies, et on retire une tentative.

Sinon, on affiche qu'il a déjà rentré cette lettre. Si elle est erronée, on lui retire une tentative.

4.8.3 Algorithme



4.8.4 Dictionnaire des éléments associés à cet algorithme

Nom	Type	Signification
motcache	chaîne de caractères	Mot qui est affiché en tant qu'aide pour le joueur. Il correspond dans cet algorithme aux lettres déjà saisies et correctes.
tentativesRestantes	entier	Nombre de tentatives erronées restantes au joueur.
lettresIncorrectes	chaîne de caractères	Chaîne contenant toutes les lettres incorrectes déjà saisies par le joueur.
mot	chaîne de caractères	Mot à deviner par le joueur.
dejaVuIncorrecte	booléen	Vrai si la lettre a déjà été saisie et est incorrecte.
dejaVuCorrecte	booléen	Vrai si la lettre a déjà été saisie et est correcte.
trouve	booléen	Vrai si la lettre est présente dans le mot.
lettre	caractère	Lettre saisie par le joueur.

4.9 Affichage du retour fait au joueur

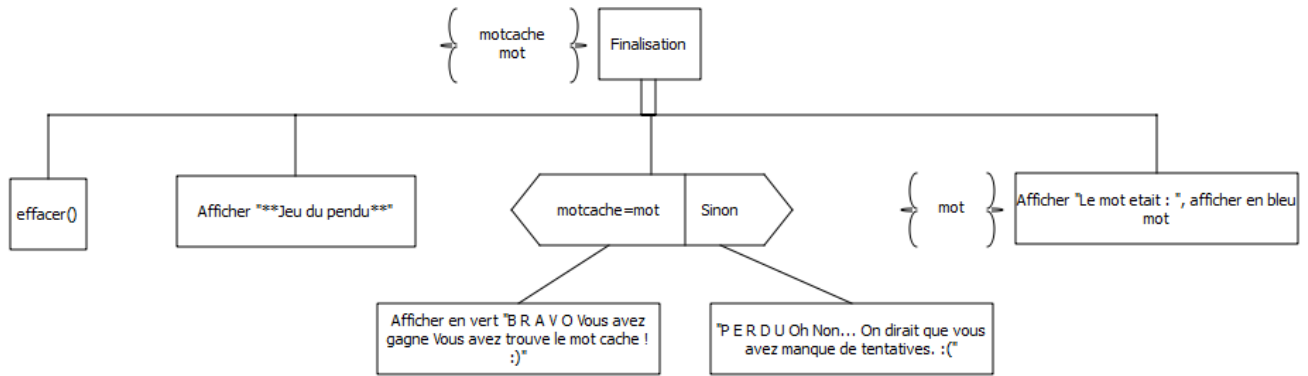
4.9.1 But de l'action

Ici, il s'agit de finir correctement le jeu. On rappelle que l'on arrive à cette partie du programme uniquement si le joueur n'a plus de tentatives ou s'il a trouvé le mot. On doit alors effacer une dernière fois pour l'affichage final et réafficher le nom du jeu. On crée aussi la condition d'affichage de défaite ou de victoire : si le mot qui était caché est égal au mot qui était à trouvé, alors il a gagné, et on le lui annonce. Sinon, on lui affiche la défaite. Le mot à trouver est aussi affiché en bleu pour qu'il puisse le revoir, même s'il a gagné.

4.9.2 Stratégie mis en oeuvre

Cet algorithme est une décomposition séquentielle qui efface, puis réaffiche le nom du jeu. Un bloc de décomposition alternative fait alors place, servant à annoncer le résultat de la partie, et enfin, on affiche le mot que le joueur devait deviner.

4.9.3 Algorithme



4.9.4 Dictionnaire des éléments associés à cet algorithme

Nom	Type	Signification
motcache	chaîne de caractères	Mot qui est affiché en tant qu'aide pour le joueur. Il correspond dans cet algorithme aux lettres déjà saisies et correctes.
mot	chaîne de caractères	Mot à deviner par le joueur.

5 Traces d'exécution

```

**Jeu du pendu**

[1] Informatique
[2] Mathematiques
[3] Cinema

Choisissez un theme : 5
Theme non reconnu... Recommencez.
[1] Informatique
[2] Mathematiques
[3] Cinema

Choisissez un theme : _
  
```

Figure 1 : Comportement quand le joueur se trompe en saisissant un thème non-reconnu (saisieVerif)

```

**Jeu du pendu**

Le mot a trouver : -----

Nombre de coups restant : 7
Entrez une lettre : e

La lettre n'est PAS presente dans le mot :- (
  
```

Figure 2 : Le joueur saisit une lettre qui n'est pas presente dans le mot

```

**Jeu du pendu**

Le mot a trouver : -----

Nombre de coups restant : 6
Entrez une lettre : r

La lettre r EST presente dans le mot :- )
  
```

Figure 3 : Le joueur saisit une lettre présente dans le mot

```
**Jeu du pendu**  
  
Le mot a trouver : -r-----  
  
Nombre de coups restant : 6  
Entrez une lettre : e  
  
Vous avez deja essaye cette lettre et elle n'est toujours pas dans le mot...
```

Figure 4 : Le joueur saisit une lettre qu'il avait déjà saisi auparavant et qui est incorrecte

```
**Jeu du pendu**  
  
Le mot a trouver : -r-----  
  
Nombre de coups restant : 5  
Entrez une lettre : r  
  
Vous avez deja trouve cette lettre...Essayez en d'autres :)
```

Figure 5 : Le joueur saisit une lettre qu'il avait déjà saisi auparavant et qui est dans le mot

```
**Jeu du pendu**  
  
Le mot a trouver : fra-tion  
  
Nombre de coups restant : 3  
Entrez une lettre : c
```

Figure 6 : À la figure 6, le joueur n'a plus que 3 tentatives restantes, le texte s'affiche donc en violet.

```
**Jeu du pendu**  
  
B R A V O  
Vous avez gagne  
Vous avez trouve le mot cache ! :)  
Le mot etait : fraction
```

Figure 7 : À la figure 6, le joueur a entré la lettre "c", et a donc trouvé le mot "fraction"

```
**Jeu du pendu**  
  
Le mot a trouver : ----n----n-  
  
Nombre de coups restant : 1  
Entrez une lettre : _
```

Figure 8 : Le joueur n'a plus qu'une seule tentative, le texte s'affiche donc en rouge

```
P E R D U  
Oh Non...  
On dirait que vous avez manque de tentatives. :(  
Le mot etait : arrangement
```

Figure 9 : Le joueur est à court de tentatives, il a donc perdu la partie

6 Remarques

- La réalisation des livrables, et en particulier du PDF a été la partie qui a pris de loin le plus de temps
- Bien moins de difficultés à programmer que pour le Jeu 1 car nous connaissions déjà une grande partie de la syntaxe
- Des explications assez fastidieuses dans le PDF, en particulier au niveau des “si”, “sinon” et de l’enchaînement entre décomposition séquentielle/alternative/itérative

7 Code C++

Le fichier main.cpp est livré dans l’archive ainsi que la bibliothèque game-tools nécessaire à son fonctionnement.

Nous avons aussi joint le fichier .tbr (tabula rasa) ainsi qu’une capture d’écran de tout l’algorithme, bien que le .pdf doive se suffire à lui-même.