**** 

**République du Sénégal**

Un peuple-Un but-Une foi

**MINISTERE DE L’ENSEIGNEMENT SUPERIEUR DE LA RECHERCHE ET DE L’INNOVATION**

**UFR SCIENCES ECONOMIQUES ET SOCIALES**

**DEPARTEMENT DE MANAGEMENT DES ORGANISATIONS**

**Master 1 en ingénierie des données et intelligence artificielle**

Algorithme et Complexité

**Projet sur les structures de contrôle**

p

**PRESENTES PAR :**

**PROFESSEURS:**

**Pr Mamadou BOUSSO**

**M. Mansour DIOUF**

**Abdoulahi MBENGUE**

**Alioune CISSE**

**Souhoude OUEDRAOGO**

**Année académique 2019-2020**

**Découpages en bloc**

Notre projet est découpé en 10 fonctions que sont:

* Fonction Saisie
* Fonction choix du niveau
* Fonction chargement de la liste et choix hasardeux d’un mot
* Fonction saisir lettre
* Fonction vérifier si la lettre avait été déjà saisie
* Fonction vérifier si c’est une consonne qui n’est pas dans le mot
* Fonction vérifier si c’est une voyelle qui n’est pas dans le mot
* Fonction vérifier si c’est un caractère différent des lettres de l’alphabet
* Fonction vérifier si la lettre est acceptée et récupérée
* Fonction vérifier si le mot est correct ou pas et affichage
* Fonction principale

**Commentaires de spécification**

* **Fonction saisie :**

1. Objectif : Saisir le niveau de jeu
2. Méthodes : Lecture à partir du clavier
3. Besoins :
4. Connus :
5. Entrées :
6. Sorties : niveau
7. Hypothèses :
8. Résultat :

* **Fonction choix du niveau :**

1. Objectif : Vérifier le niveau
2. Méthode : Boucle « while »
3. Besoins : niveau
4. Connus :
5. Entrées : niveau
6. Sorties : niveau
7. Hypothèses : 0<niveau<4
8. Résultat : Afficher niveau choisi

* **Fonction chargement et choix d’un mot :**

1. Objectif : Afficher le nombre de mots de la liste et choix au hasard d’un mot
2. Méthode : Condition « if », « elif », « else », utilisation de « open » et « random »
3. Besoins : niveau
4. Connus :
5. Entrées : niveau
6. Sorties : niveau, mot
7. Hypothèses : mot<=5 lettres pour niveau 1, mot <=8 lettres pour niveau 2 et mot >=10 lettres pour niveau 3
8. Résultats : Affichage du nombre de mots chargés

* **Fonction saisir lettre :**

1. Objectif : Saisie d’une lettre par l’utilisateur
2. Méthode : Lecture à partir du clavier
3. Besoins :
4. Connus :
5. Entrées :
6. Sorties : lettre
7. Hypothèses :
8. Résultat :

* **Fonction vérifier si la saisie est correcte mais déjà saisie avant**

1. Objectif : Vérifier si la lettre saisie par l’utilisateur est valide et n’a pas été déjà saisie
2. Méthode : Condition « if »
3. Besoins :
4. Connus :
5. Entrées : lettre, mot\_en\_construction, mot
6. Sorties : lettre, mot\_en\_construction, mot
7. Hypothèses :
8. Résultats : Dire à l’utilisateur qu’il a déjà saisi cette lettre

* **Fonction vérifier si la saisie est une consonne qui n’est pas dans le mot**

1. Objectif : Vérifier si l’utilisateur a saisi une consonne qui n’est pas dans le mot
2. Méthode : Condition « if »
3. Besoins :
4. Connus :
5. Entrées : lettre, consonne, mot
6. Sorties : lettre, consonne mot
7. Hypothèses :
8. Résultat : Dire à l’utilisateur que cette consonne n’est pas dans le mot

* **Fonction vérifier si la saisie est une voyelle qui n’est pas dans le mot**

1. Objectif : Vérifier si l’utilisateur a saisi une voyelle qui n’est pas dans le mot
2. Méthode : Condition « if »
3. Besoins :
4. Connus :
5. Entrées : lettre, voyelle, mot
6. Sorties : lettre, voyelle, mot
7. Hypothèses :
8. Résultat : Dire à l’utilisateur que cette voyelle n’est pas dans le mot

* **Fonction vérifier si la saisie n’est pas une lettre de l’alphabet**

1. Objectif : Vérifier si l’utilisateur a saisi une lettre qui n’est pas de l’alphabet français
2. Méthode : Condition « if »
3. Besoins :
4. Connus :
5. Entrées : lettre, consonne, voyelle
6. Sorties : lettre, consonne, voyelle
7. Hypothèses :
8. Résultats : Dire à l’utilisateur qu’il ne peut entrer que des caractères de l’alphabet

* **Fonction vérifier si la saisie est correcte et n’a pas été saisie avant**

1. Objectif : Vérifier si la saisie est correcte et n’a pas été déjà saisie
2. Méthode : Condition « if »
3. Besoins :
4. Connus :
5. Entrées : lettre, mot\_en\_construction, mot
6. Sorties : lettre, mot\_en\_construction, mot
7. Hypothèses :
8. Résultat : Dire à l’utilisateur que la saisie est correcte

* **Fonction vérifier si le joueur a gagné ou perdu et affichage**

1. Objectif : Vérifier si le joueur a gagné ou perdu et affichage résultat
2. Méthode : Boucle « while », condition « if »
3. Besoins :
4. Connus :
5. Entrées : lettre, mot\_en\_construction, mot, nbre\_tentatives
6. Sorties :
7. Hypothèses :
8. Résultats : Afficher « victoire » ou « échec »

**Diagramme de flux**

Lettre

Niveau, mot

Niveau

Niveau

Niveau

Niveau

9

8

4

3

1

10

2

Lettre, tentatives, mot\_en\_construction, mot

Lettre, mot, mot\_en\_construction

Lettre, mot, mot\_en\_construction

Lettre, voyelle, mot, consonne

Lettre, voyelle, mot, consonne

Lettre, mot\_en\_contruction, mot

Lettre, mot, voyelle

Lettre, mot\_en\_construction, mot

Lettre, mot, voyelle

Lettre, consonne, mot

Lettre, mot, consonne

5

7

6

**Légende :**

1

Fonction Saisie

2

Fonction choix du niveau

3

Fonction chargement de la liste et choix hasardeux d’un mot

4

Fonction saisir lettre

Fonction vérifier si la lettre avait été déjà saisie

5

6

Fonction vérifier si c’est une consonne qui n’est pas dans le mot

7

Fonction vérifier si c’est une voyelle qui n’est pas dans le mot

8

Fonction vérifier si c’est un caractère différent des lettres de l’alphabet

9

Fonction vérifier si la lettre est acceptée et récupérée

10

Fonction vérifier si le mot est correct ou pas et affichage

Tableaux de flux

|  |  |  |
| --- | --- | --- |
|  | Reçoit | Fournit |
| Fonction 1 |  | niveau |
| Fonction 2 | niveau | niveau |
| Fonction 3 | niveau | niveau, mot |
| Fonction 4 |  | lettre |
| Fonction 5 | lettre, mot\_en\_construction, mot | lettre, mot\_en\_construction, mot |
| Fonction 6 | lettre, consonne, mot | lettre, consonne, mot |
| Fonction 7 | lettre, voyelle, mot | lettre, voyelle, mot |
| Fonction 8 | lettre, voyelle, consonne, mot | lettre, voyelle, consonne, mot |
| Fonction 9 | lettre, mot\_en\_construction, mot | lettre, mot\_en\_construction, mot |
| Fonction 10 | lettre, mot, mot\_en\_construction, tentatives |  |