**1.Chargement des scores (charger\_scores)** :

* + Cette fonction charge les scores depuis le fichier scores.json. Si le fichier n'existe pas, il en crée un vide.
  + Code clé :

python

CopierModifier

def charger\_scores(fichier\_scores):

if not os.path.exists(fichier\_scores):

with open(fichier\_scores, "w") as file:

json.dump({}, file)

with open(fichier\_scores, "r") as file:

scores = json.load(file)

return scores

1. **Sauvegarde des scores (sauvegarder\_scores)** :
   * Cette fonction enregistre les données des scores mises à jour dans scores.json.
   * Code clé :

python

CopierModifier

def sauvegarder\_scores(fichier\_scores, scores):

with open(fichier\_scores, "w") as file:

json.dump(scores, file, indent=4)

print(f"Scores sauvegardés dans {fichier\_scores}.")

1. **Mise à jour des scores en cas de défaite (mettre\_a\_jour\_score\_defaite)** :
   * Elle gère les défaites, en incrémentant le compteur et en réduisant les points selon des règles spécifiques.
   * Code clé :

python

CopierModifier

def mettre\_a\_jour\_score\_defaite(fichier\_scores, joueur):

scores = charger\_scores(fichier\_scores)

joueur = joueur.lower()

if joueur not in scores:

scores[joueur] = {"points": 0, "victoires": 0, "defaites": 1, "defaites\_consecutives": 1}

else:

scores[joueur]["defaites"] += 1

scores[joueur]["defaites\_consecutives"] += 1

if scores[joueur]["defaites\_consecutives"] == 2:

scores[joueur]["points"] -= 3

else:

scores[joueur]["points"] -= 1

sauvegarder\_scores(fichier\_scores, scores)

1. **Afficher l'historique des scores (afficher\_scores)** :
   * Cette fonction lit et affiche les scores en un format lisible.
   * Code clé :

python

CopierModifier

def afficher\_scores(fichier\_scores):

scores = charger\_scores(fichier\_scores)

if scores:

for joueur, data in scores.items():

print(f"{joueur.capitalize()} - Points : {data['points']}, Victoires : {data['victoires']}, Défaites : {data['defaites']}")

else:

print("Aucun score enregistré.")

**Étape 3 : Gestion des mots**

1. **Ajouter un mot (add\_word)** :
   * Permet d’ajouter un mot dans words.txt tout en vérifiant qu’il n’existe pas déjà.
   * Code clé :

python

CopierModifier

def add\_word():

new\_word = input("Veuillez entrer un nouveau mot : ").strip().lower()

with open("words.txt", "r", encoding="UTF-8") as secret\_words:

secret\_words\_list = [word.strip().lower() for word in secret\_words.readlines()]

if new\_word in secret\_words\_list:

print("Ce mot existe déjà")

else:

with open("words.txt", "a", encoding="UTF-8") as secret\_words:

secret\_words.write("\n" + new\_word)

print("Mot ajouté avec succès.")

1. **Récupérer un mot aléatoire (get\_guess\_word)** :
   * Sélectionne un mot au hasard dans le fichier words.txt.
   * Code clé :

python

CopierModifier

def get\_guess\_word():

with open("words.txt", 'r', encoding="UTF-8") as secret\_words:

secret\_words\_list = secret\_words.readlines()

guess\_word = list(unidecode(secret\_words\_list[secrets.randbelow(len(secret\_words\_list))].strip().lower()))

return guess\_word

**Étape 4 : Jeu principal**

1. **Affichage du menu (display\_menu)** :
   * Permet à l'utilisateur de choisir une action (nouvelle partie, ajouter un mot, afficher les scores, etc.).
   * Code clé :

python

CopierModifier

def display\_menu():

menu\_choice = input(" === MENU === \n"

"1. Nouvelle partie \n"

"2. Partie en cours \n"

"3. Historique des scores \n"

"4. Ajouter un mot \n"

"5. Quitter le jeu \n").strip()

return menu\_choice

1. **Déroulement d'une partie (game\_loop)** :
   * Gère le jeu du pendu : vies, lettres proposées, et victoire/défaite.
   * Code clé :

python

CopierModifier

def game\_loop(fichier\_scores, joueur, life\_count, guess\_word, user\_word\_format):

while life\_count >= 0:

guess\_letter = unidecode(input("Veuillez entrer une lettre : ").lower()[0:1])

if not guess\_letter.isalpha() or len(guess\_letter) != 1:

continue

if guess\_letter in guess\_word:

for i in range(len(guess\_word)):

if guess\_word[i] == guess\_letter:

user\_word\_format[i] = guess\_letter

else:

life\_count -= 1

print(f"Il vous reste {life\_count} vies.")

if "\_" not in user\_word\_format:

print("Vous avez gagné !")

break

**Étape 5 : Intégration dans la fonction principale**

1. La fonction main() :
   * Coordonne les actions en fonction des choix du joueur.
   * Code clé :

python

CopierModifier

def main():

scores\_file = "scores.json"

while True:

menu = display\_menu()

if menu == "1":

joueur = player\_id(scores\_file)

guess\_word = get\_guess\_word()

user\_word\_format = display\_user\_word(guess\_word)

life\_count = 7

game\_loop(scores\_file, joueur, life\_count, guess\_word, user\_word\_format)

elif menu == "3":

afficher\_scores(scores\_file)

elif menu == "4":

add\_word()

elif menu == "5":

break

En suivant ces étapes, ton amie pourra comprendre chaque partie du code et son rôle dans le fonctionnement global du jeu du pendu.