**DIAGRAMME FONCTIONNEL**

**Étape 1 : Préparer le fichier mots.txt**

1. Créez un fichier texte nommé mots.txt.
2. Insérez au moins 15 mots, chacun sur une ligne. Par exemple :

CopierModifier

python

pendu

programme

ordinateur

developpement

graphe

ecran

clavier

souris

bibliotheque

fonction

boucle

variable

affichage

liste

**Étape 2 : Installer Pygame**

1. Installez Pygame si ce n'est pas déjà fait :

CopierModifier

pip install pygame

**Étape 3 : Planifier le code**

Voici les fonctionnalités que nous allons programmer, étape par étape :

1. Charger les mots depuis le fichier mots.txt.
2. Ajouter un menu principal avec deux options :
   * Jouer.
   * Ajouter un mot dans mots.txt.
3. Implémenter le mécanisme de sélection aléatoire d'un mot et afficher les underscores (\_) correspondant à ses lettres.
4. Permettre au joueur de proposer une lettre et révéler les lettres correctes.
5. Dessiner le pendu en cas d'erreurs.
6. Terminer le jeu lorsque le joueur gagne ou perd.

**Étape 4 : Code et Explications**

Voici le programme avec les explications détaillées dans chaque étape :

**1. Charger les mots depuis le fichier**

Commencez par lire les mots dans un fichier texte et les stocker dans une liste :

python

CopierModifier

import random

import pygame

# Charger les mots du fichier mots.txt

def charger\_mots(fichier):

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

mots = file.read().splitlines() # Lit chaque ligne comme un mot

return mots

**2. Ajouter un menu principal**

Le joueur peut choisir de jouer ou d’ajouter un mot. Nous ajoutons une fonction pour insérer un nouveau mot :

python

CopierModifier

# Ajouter un mot au fichier mots.txt

def ajouter\_mot(fichier):

nouveau\_mot = input("Entrez un mot à ajouter : ").lower()

if nouveau\_mot.isalpha(): # Vérifie que le mot ne contient que des lettres

with open(fichier, "a") as file:

file.write(nouveau\_mot + "\n")

print(f"Le mot '{nouveau\_mot}' a été ajouté avec succès.")

else:

print("Veuillez entrer un mot valide.")

**3. Créer le menu principal**

Affichez un menu pour choisir une option (jouer ou ajouter un mot) :

python

CopierModifier

def menu\_principal(fichier):

while True:

print("\n=== Menu Principal ===")

print("1. Jouer")

print("2. Ajouter un mot")

print("3. Quitter")

choix = input("Choisissez une option (1-3) : ")

if choix == "1":

jouer(fichier)

elif choix == "2":

ajouter\_mot(fichier)

elif choix == "3":

print("Au revoir !")

break

else:

print("Choix invalide. Veuillez réessayer.")

**4. Implémenter la fonction pour jouer**

1. Choisissez un mot aléatoire et initialisez les variables nécessaires :
   * mot\_mystere : le mot à deviner.
   * mot\_affiche : les underscores pour les lettres non découvertes.
   * erreurs : le nombre d’erreurs commises.
2. Configurez une boucle de jeu où le joueur propose une lettre à chaque tour.
3. Utilisez Pygame pour afficher le mot, le pendu et le clavier.

Voici le code complet pour la fonction jouer :

python

CopierModifier

def jouer(fichier):

# Charger les mots

mots = charger\_mots(fichier)

mot\_mystere = random.choice(mots) # Mot choisi aléatoirement

mot\_affiche = "\_" \* len(mot\_mystere)

lettres\_proposees = []

erreurs = 0

max\_erreurs = 6 # Le pendu se dessine en 6 étapes

# Initialiser Pygame

pygame.init()

largeur, hauteur = 600, 400

ecran = pygame.display.set\_mode((largeur, hauteur))

pygame.display.set\_caption("Jeu du Pendu")

# Couleurs

blanc = (255, 255, 255)

noir = (0, 0, 0)

rouge = (255, 0, 0)

# Fontes

fonte = pygame.font.Font(None, 36)

jeu\_en\_cours = True

while jeu\_en\_cours:

for event in pygame.event.get():

if event.type == pygame.QUIT:

jeu\_en\_cours = False

if event.type == pygame.KEYDOWN:

lettre = event.unicode.lower()

if lettre.isalpha() and lettre not in lettres\_proposees:

lettres\_proposees.append(lettre)

if lettre in mot\_mystere:

# Met à jour le mot affiché

mot\_affiche = "".join([

lettre if mot\_mystere[i] == lettre else mot\_affiche[i]

for i in range(len(mot\_mystere))

])

else:

erreurs += 1

# Dessiner l'écran

ecran.fill(blanc)

texte\_mot = fonte.render(f"Mot : {mot\_affiche}", True, noir)

ecran.blit(texte\_mot, (20, 20))

# Dessiner le pendu selon le nombre d'erreurs

if erreurs > 0:

pygame.draw.circle(ecran, noir, (300, 100), 30, 2) # Tête

if erreurs > 1:

pygame.draw.line(ecran, noir, (300, 130), (300, 200), 2) # Corps

if erreurs > 2:

pygame.draw.line(ecran, noir, (300, 150), (270, 180), 2) # Bras gauche

if erreurs > 3:

pygame.draw.line(ecran, noir, (300, 150), (330, 180), 2) # Bras droit

if erreurs > 4:

pygame.draw.line(ecran, noir, (300, 200), (270, 250), 2) # Jambe gauche

if erreurs > 5:

pygame.draw.line(ecran, noir, (300, 200), (330, 250), 2) # Jambe droite

# Vérification de la fin du jeu

if mot\_affiche == mot\_mystere:

texte\_gagne = fonte.render("Vous avez gagné !", True, rouge)

ecran.blit(texte\_gagne, (20, 300))

jeu\_en\_cours = False

elif erreurs >= max\_erreurs:

texte\_perdu = fonte.render(f"Perdu ! Le mot était : {mot\_mystere}", True, rouge)

ecran.blit(texte\_perdu, (20, 300))

jeu\_en\_cours = False

pygame.display.flip()

pygame.quit()

**5. Exécuter le programme**

Ajoutez un point d’entrée pour lancer le menu principal :

python

CopierModifier

if \_\_name\_\_ == "\_\_main\_\_":

menu\_principal("mots.txt")

Avec ce code, vous avez un jeu du pendu fonctionnel utilisant un fichier mots.txt et Pygame pour l'interface graphique. 🎮

4o