"""sumary\_line

Jogo da forca:

- Guardar a palavra secreta

- Pedir a letra

- Verificar se a letra está na palavra

- Mostrar a palavra com as letras descobertas

- Mostrar a quantidade de letras que faltam

- Mostrar as letras que já foram chutadas

- Desenhar o boneco da forca

- Verificar se o jogo terminou

- Mostrar o resultado

- Perguntar se quer jogar de novo

"""

import tkinter as tk # Importa o módulo tkinter renomeando-o para tk

from tkinter import messagebox # Importa o módulo messagebox do tkinter

from tkinter.font import BOLD

# Crie uma classe chamada Forca que herda de tk.Frame que é um container para organização de widgets (componentes gráficos) # noqa: E501

class Forca:

# Crie o método construtor da classe, que recebe como parâmetro o objeto fdo tkinter e configura a janela principal # noqa: E501

def \_\_init\_\_(self):

self.janela = tk.Tk() # Cria a janela principal

self.janela.geometry("380x500") # Define o tamanho da janela principal

self.janela.title("JOGO DA FORCA") # Define o título da janela principal

self.janela.resizable(False, False) # Impede que a janela seja redimensionada

self.janela.iconbitmap("forca.ico") # Define o ícone da janela principal

self.janela.iconbitmap.\_\_reduce\_\_ # Define o ícone da janela principal

self.janela.bg\_image = tk.PhotoImage(file="forca.png") # Define a imagem de fundo da janela principal # noqa: E501

# Crie os widgets da janela principal

self.janela.bg\_label = tk.Label(self.janela, image=self.janela.bg\_image) # Cria o label para a imagem de fundo # noqa: E501

self.janela.bg\_label.place(x=250, y=1) # Posiciona o label para a imagem de fundo

self.palavra\_label = tk.Label(self.janela, text="Digite a palavra secreta:") # Cria o label para a palavra secreta # noqa: E501

self.palavra\_entry = tk.Entry(self.janela, show='\*', width=30) # Cria o Entry para a palavra secreta # noqa: E501

self.palavra\_entry.focus\_set() # Coloca o foco no Entry para adivinhar

self.adivinhar\_label = tk.Label(self.janela, text="Digite uma letra:") # Cria o label para a letra a ser adivinhada # noqa: E501

self.adivinhar\_entry = tk.Entry(self.janela, width=5) # Cria o Entry para a letra a ser adivinhada # noqa: E501

self.adivinhar\_entry.focus\_set() # Coloca o foco no Entry para adivinhar

self.adivinhar\_button = tk.Button(self.janela, text="PLAY", command=self.check\_adivinhar, bg="green", font=("symbol",12,BOLD), width=8) # Cria o botão para adivinhar a letra # noqa: E501

self.adivinhar\_button.bind("<Return>", lambda event: self.check\_adivinhar()) # Liga a tecla Enter ao método check\_adivinhar # noqa: E501

self.restart\_button = tk.Button(self.janela, text="RESTART", command=self.restart\_game, bg= "red",font=("symbol",12,BOLD), width=8) # Cria o botão para reiniciar o jogo # noqa: E501

self.restart\_button.bind("<Return>", lambda event: self.restart\_game()) # Liga a tecla Enter ao método restart\_game # noqa: E501

self.resultado\_label = tk.Label(self.janela, text="") # Cria o label para o resultado # noqa: E501

self.erros\_label = tk.Label(self.janela, text="") # Cria o label para as letras digitadas # noqa: E501

self.erros2\_label = tk.Label(self.janela, text="") # Cria o label para as letras digitadas # noqa: E501

self.palavra\_progress\_label = tk.Label(self.janela, text="") # Cria o label para o progresso da palavra # noqa: E501

self.enforcado\_canvas = tk.Canvas(self.janela, width=250, height=200,bg='black') # Cria o Canvas para desenhar o boneco da forca # noqa: E501

# Posicione os widgets na janela principal usando o gerenciador de layout place do tkinter # noqa: E501

self.palavra\_label.place(x=10, y=10) # Posiciona o label para a palavra secreta

self.palavra\_entry.place(x=10, y=35) # Posiciona o Entry para a palavra secreta

self.adivinhar\_label.place(x=10, y=68) # Posiciona o label para a letra a ser adivinhada # noqa: E501

self.adivinhar\_entry.place(x=140, y=68) # Posiciona o Entry para a letra a ser adivinhada # noqa: E501

self.adivinhar\_button.place(x=90, y=108) # Posiciona o botão para adivinhar a letra

self.restart\_button.place(x=200, y=108) # Posiciona o botão para reiniciar o jogo # noqa: E501

self.resultado\_label.place(x=90, y=145) # Posiciona o label para o resultado

self.erros\_label.place(x=10, y=180) # Posiciona o label para a frase letras digitadas # noqa: E501

self.erros2\_label.place(x=50, y=200) # Posiciona o label para as letras digitadas # noqa: E501

self.palavra\_progress\_label.place(x=50, y=230) # Posiciona o label para o progresso da palavra # noqa: E501

self.enforcado\_canvas.place(x=60, y=270) # Posiciona o Canvas para desenhar o boneco da forca # noqa: E501

# Crie as variáveis de instância palavra\_secreta, adivinhar\_letras, erros2 e game\_over # noqa: E501

self.palavra\_secreta = "" # Palavra secreta

self.adivinhar\_letras = set() # Conjunto de letras adivinhadas

self.erros2 = 0 # Número de erros

self.game\_over = False # Flag de fim de jogo

# Crie o método para validar a entrada da palavra secreta

def check\_palavra(self):

self.palavra\_secreta = self.palavra\_entry.get().upper() # Pega a palavra digitada e converte para maiúscula # noqa: E501

self.palavra\_entry.config(state=tk.DISABLED) # Desabilita o Entry para adivinhar

if len(self.palavra\_secreta) < 4: # Verifica se a palavra secreta tem pelo menos 4 letras # noqa: E501

messagebox.showwarning("Erro", "A palavra secreta deve ter pelo menos 4 letras!") # Mostra uma mensagem de erro # noqa: E501

self.palavra\_entry.config(state=tk.NORMAL) # Habilita o Entry para adivinhar

self.palavra\_secreta = "" # Apaga a palavra secreta

return

if not self.palavra\_secreta.isalpha(): # Verifica se a palavra secreta tem apenas letras # noqa: E501

messagebox.showwarning("Erro", "A palavra secreta deve ser composta apenas por letras.") # Mostra uma mensagem de erro # noqa: E501

self.palavra\_entry.config(state=tk.NORMAL)

self.palavra\_secreta = ""

return

if len(self.palavra\_secreta.split()) > 1: # Verifica se a palavra secreta tem mais de uma palavra # noqa: E501

messagebox.showwarning("Erro", "Só vale uma palavra!")

self.palavra\_entry.config(state=tk.NORMAL)

self.palavra\_secreta = "" # Apaga a palavra secreta

return

# Crie o método check\_adivinhar para verificar se a letra está na palavra secreta

def check\_adivinhar(self):

adivinhar = self.adivinhar\_entry.get().upper() # Pega a letra digitada e converte para maiúscula # noqa: E501

self.adivinhar\_entry.delete(0, tk.END) # Apaga o texto do Entry para adivinhar

if self.game\_over: # Verifica se o jogo acabou

return

if len(adivinhar) != 1: # Verifica se foi digitada apenas uma letra

messagebox.showwarning("Erro", "Digite pelo menos uma letra!")

return

if not adivinhar.isalpha(): # Verifica se foi digitada uma letra

messagebox.showwarning("Erro", "Digite só uma letra por vez!")

return

if adivinhar in self.adivinhar\_letras: # Verifica se a letra já foi digitada

messagebox.showwarning("Erro", "Esta letra já foi!")

return

if not self.palavra\_secreta: # Verifica se a palavra secreta já foi digitada

self.palavra\_secreta = self.palavra\_entry.get().upper()

self.palavra\_entry.config(state=tk.DISABLED)

if not self.palavra\_secreta.isalpha(): # Verifica se a palavra secreta tem apenas letras # noqa: E501

messagebox.showwarning("Erro", "Só vale letras!")

self.palavra\_entry.config(state=tk.NORMAL)

self.palavra\_secreta = "" # Apaga a palavra secreta

return

if len(self.palavra\_secreta.split()) > 1: # Verifica se a palavra secreta tem mais de uma palavra # noqa: E501

messagebox.showwarning("Erro", "Só vale uma palavra!")

self.palavra\_entry.config(state=tk.NORMAL) # Habilita o Entry para adivinhar # noqa: E501

self.palavra\_secreta = "" # Apaga a palavra secreta

return

if len(self.palavra\_secreta) < 4: # Verifica se a palavra secreta tem pelo menos 4 letras # noqa: E501

messagebox.showwarning("Erro", "A palavra secreta deve ter pelo menos 4 letras!") # Mostra uma mensagem de erro # noqa: E501

self.palavra\_entry.config(state=tk.NORMAL) # Habilita o Entry para adivinhar # noqa: E501

self.palavra\_secreta = "" # Apaga a palavra secreta

return

if adivinhar in self.palavra\_secreta: # Verifica se a letra está na palavra secreta

self.adivinhar\_letras.add(adivinhar) # Se estiver, adiciona a letra ao conjunto de letras adivinhadas # noqa: E501

progress = '' # Cria uma string com as letras adivinhadas e \_ para as letras não adivinhadas # noqa: E501

for letter in self.palavra\_secreta: # Percorre a palavra secreta

if letter in self.adivinhar\_letras: # Verifica se a letra está no conjunto de letras adivinhadas # noqa: E501

progress += letter + ' ' # Se estiver, adiciona a letra e um espaço

else:

progress += '\_ ' # Se não estiver, adiciona um \_ e um espaço

# Atualiza o label com o progresso

self.palavra\_progress\_label["text"] = progress.strip() # strip() remove os espaços no início e no fim da string # noqa: E501

if self.palavra\_secreta == progress.replace(' ', ''): # Verifica se o jogador ganhou comparando a palavra secreta com o progresso sem espaços # noqa: E501

self.resultado\_label["text"] = "VENCEU! PARABÉNS!!!!" # Atualiza o label com o resultado # noqa: E501

self.resultado\_label["fg"] = "green" # Muda a cor do texto para verde

self.palavra\_progress\_label["fg"] = "green" # Muda a cor do texto para verde # noqa: E501

self.game\_over = True # Atualiza a flag de fim de jogo

else:

self.adivinhar\_letras.add(adivinhar) # Se não estiver, adiciona a letra ao conjunto de letras adivinhadas # noqa: E501

self.erros2 += 1 # Incrementa o número de erros

self.desenha\_enforcado() # Desenha o boneco da forca

if self.erros2 == 10: # Verifica se o jogador perdeu

self.resultado\_label["text"] = "A palavra era: " + self.palavra\_secreta

self.resultado\_label["fg"] = "red" # Muda a cor do texto para vermelho

self.game\_over = True # Atualiza a flag de fim de jogo

# Atualiza o label com as letras adivinhadas

self.erros\_label["text"] = "Letras digitadas: "

self.erros2\_label["text"] = "" + ', '.join(self.adivinhar\_letras) # join() junta os elementos do conjunto de letras adivinhadas com uma vírgula

self.adivinhar\_entry.focus\_set() # Coloca o foco no Entry para adivinhar

self.adivinhar\_entry.select\_range(0, tk.END) # Seleciona o texto do Entry para adivinhar

self.adivinhar\_entry.icursor(tk.END) # Coloca o cursor no final do texto do Entry para adivinhar

self.adivinhar\_entry.xview\_moveto(1.0) # Move o texto do Entry para a direita

self.adivinhar\_entry.yview\_moveto(1.0) # Move o texto do Entry para baixo

self.adivinhar\_entry.config(state=tk.NORMAL) # Habilita o Entry para adivinhar

self.palavra\_entry.config(state=tk.NORMAL) # Habilita o Entry para adivinhar

# Desenhando o boneco da forca

def desenha\_enforcado(self):

self.enforcado\_canvas.delete("all")

self.enforcado\_canvas.create\_line(20, 180, 180, 180) # base

self.enforcado\_canvas.create\_line(50, 180, 50, 20) # poste

self.enforcado\_canvas.create\_line(50, 20, 100, 20) # trave

self.enforcado\_canvas.create\_line(100, 20, 100, 40) # corda

if self.erros2 >= 1:

self.enforcado\_canvas.create\_oval(90, 40, 110, 60) # cabeça

if self.erros2 >= 2:

self.enforcado\_canvas.create\_line(100, 60, 100, 100) # corpo

if self.erros2 >= 3:

self.enforcado\_canvas.create\_line(100, 100, 80, 120) # braço esquerdo

if self.erros2 >= 4:

self.enforcado\_canvas.create\_line(100, 100, 120, 120) # braço direito

if self.erros2 >= 5:

self.enforcado\_canvas.create\_line(100, 80, 80, 70) # perna esquerda

if self.erros2 >= 6:

self.enforcado\_canvas.create\_line(100, 80, 120, 70) # perna direita

if self.erros2 == 6:

self.enforcado\_canvas.create\_text(100, 150, text="GAME OVER", fill="red", font=("Arial", 20, "bold")) # Mensagem de GAME OVER em vermelho

if self.erros2 == 6:

self.palavra\_progress\_label["fg"] = "red" # Muda a cor do texto para vermelho

# Reiniciar o jogo

def restart\_game(self):

self.palavra\_entry.config(state=tk.NORMAL) # Habilita o Entry para adivinhar

self.palavra\_entry.delete(0, tk.END) # Apaga o texto do Entry para adivinhar

self.adivinhar\_entry.delete(0, tk.END) # Apaga o texto do Entry para adivinhar

self.resultado\_label["text"] = "" # Apaga o texto do label de resultado

self.erros\_label["text"] = "" # Apaga o texto do label de letras digitadas

self.palavra\_progress\_label["text"] = "" # Apaga o texto do label de progresso da palavra # noqa: E501

self.enforcado\_canvas.delete("all") # Apaga o desenho do boneco da forca

self.palavra\_secreta = "" # Apaga a palavra secreta

self.adivinhar\_letras = set() # Apaga o conjunto de letras adivinhadas

self.erros2\_label["text"] = "" # Apaga o texto do label de letras digitadas

self.erros2 = 0 # Zera o número de erros

self.game\_over = False # Zera a flag de fim de jogo

# Crie o método run para executar o programa

def run(self):

self.janela.mainloop() # Inicia o loop do tkinter

if \_\_name\_\_ == '\_\_main\_\_': # Verifica se este módulo é o programa principal

game = Forca() # Cria o objeto game da classe Forca

game.run() # Executa o método run do objeto game