```
import tkinter as tk
from tkinter import messagebox
import random
import pygame

window = tk.Tk()
window.title("Word search")
window.geometry("1910x1000")

pygame.init()
pygame.mixer.init()
pygame.mixer.music.load("WhatsApp Audio 2023-08-08 at 01.13.50.mpeg")
pygame.mixer.music.play(-1)

im=tk.PhotoImage(file=".png")
label=tk.Label(window,image=im)
label.place(x=0,y=0)

colours = ["blue",'yellow','pink','green','orange']




f1= open("Text_file1","w")
f1.write("B T R Y M \nD C L F I \nE N G E X \nN H U M D \nB U N J A \nBUN,DEN,TRY,MIX,GUN" )
f1.close()

f2= open("Text_file2","w")
f2.write("A K N E T \nE I U G C \nG E T B F \nG Y O U A \nH F J D R \nNET,EGG,NUT,FAR,YOU")
f2.close()
```

```
f3= open("Text_file3","w")
f3.write("F A T O B \nI F J H L\nX I D M P \nG C O E I \nM Y D K N \nFAT,FIX,PIN,IF,ME")
f3.close()

f4= open("Text_file4","w")
f4.write("F O N E H \nI N K F T \nN T E B O \nB W C G P \nT O X D S \nFIN,TOP,TWO,ON,INK")
f4.close()

f5= open("Text_file5","w")
f5.write("G P H A K \nN A R V I \nD H I P T \nJ S N K I\nK L T A P \nKIT,HIP,IN,TIP,TAP")
f5.close()

f6= open("Text_file6","w")
f6.write("D H P K C \nO T S C O \nT N E E W \nS P E A O \nH I T R K \nSEE,COW,EAR,HIT,DOT")
f6.close()

f7= open("Text_file7","w")
f7.write("G H O P K \nA A T O R \nP H I T N \nI T E A V \nS J L A P \nTEA,POT,AT,HOP,LAP")
f7.close()

f8= open("Text_file8","w")
f8.write("G B A G P \nN R N S A \nI V H Z T \nB A P P J \nK I A D R \nBAG,APP,AN,NIB,PAT")
f8.close()

f9= open("Text_file9","w")
f9.write("A N E C O \nZ G A O K \nR N A L L \nV H V U B \nB I N M Y \nALL,BIN,ECO,BY,HI")
f9.close()

f10= open("Text_file10","w")
f10.write("U R G V M \nS N O P Y \nT L J E M \nK Z V T U \nC O T W X \nUS,MY,PET,NO,COT")
f10.close()
```

```
files =
["Text_file1","Text_file2","Text_file3","Text_file4","Text_file5","Text_file6","Text_file7","Text_file8",
"Text_file9","Text_file10"]

x = random.choice(files)

print("WORD SEARCH GAME")

file1 = open(x,"r")

k = file1.readlines()

k1 = []

for i in range(len(k)-1) :

   print(k[i])

   j = k[i].split()

   k1.append(j)


words = (k[len(k)-1].replace(',',' ')).split()

GRID_SIZE = 5


class WordSearchGame(tk.Tk):

   def _init_(self):

      super()._init_()


      self.title("Word Search Game")

      self.geometry("1910x1000")

      self.config(bg="lavender")

      self.check_word_but()

      self.create_game()

      self.selected_cells=[]

      self.selected_word = ""

      self.found_words=[]
```

```
        button=tk.Button(self,text="New Game",bg = "blue",fg = "white",height = '2',width =
'10',font=("Times New Roman",15,'bold'),command=lambda:[self.withdraw(),window.deiconify()])
        button.place(x=580,y=580)
    def check_word_but(self):
         self.b=tk.Button(self,text="Check Word",bg="white",fg="red",font=("Times New
Roman","22","bold"),command=lambda:[self.check_selected_word()],height = '1',width = '10')
         self.b.place(x=880,y=580)
    def create_game(self):
        self.word_list = words
        self.grid = self.generate_grid()

        self.words = tk.Frame(self)
        self.words.pack(pady=10)

        for word in self.word_list:
            label = tk.Label(self.words, text=word, font=("Times New Roman", 12,'bold'))
            label.pack(side=tk.LEFT, padx=10)

        self.canvas = tk.Canvas(self, width=400, height=400)
        self.canvas.pack()
        self.draw_grid()
        self.canvas.bind("<Button-1>", self.on_click)

    def generate_grid(self):
        grid=k1
        return grid

    def draw_grid(self):
        cell_width = 400 / GRID_SIZE
        cell_height = 400 / GRID_SIZE

        for i in range(GRID_SIZE):
```

```
        for j in range(GRID_SIZE):
            x0 = i * cell_width
            y0 = j * cell_height
            x1 = x0 + cell_width
            y1 = y0
            x2 = x0
            y2 = y0 + cell_height
            x3 = x1
            y3 = y2
            self.canvas.create_polygon(x0, y0, x1, y1, x3, y3, x2, y2, fill="white", outline="black")

            letter = self.grid[j][i]
            self.canvas.create_text(x0 + cell_width / 2, y0 + cell_height / 2, text=letter, font=("Arial",
12,'bold'))

    def on_click(self, event):

        cell_width = 400 / GRID_SIZE
        cell_height = 400 / GRID_SIZE
        self.x = int(event.x // cell_width)
        self.y = int(event.y // cell_height)

        if (self.x, self.y) not in self.selected_cells :
            self.selected_cells.append((self.x, self.y))
            self.highlight_selected_cells()

    def highlight_selected_cells(self):
        cell_width = 400 / GRID_SIZE
        cell_height = 400 / GRID_SIZE

        for self.x, self.y in self.selected_cells:
```

```
            x0 = self.x * cell_width
            y0 = self.y * cell_height
            x1 = (self.x + 1) * cell_width
            y1 = (self.y + 1) * cell_height
            self.canvas.create_rectangle(x0,y0,x1,y1, outline="blue")

    def check_selected_word(self):

        self.on_click(self)

        if self.selected_cells[-1]==(0,0):
            self.selected_cells.remove(self.selected_cells[-1])

        for self.x,self.y in self.selected_cells:
                self.selected_word += self.grid[self.y][self.x]

        if self.selected_word in words:
            self.found_words.append(self.selected_word)
            self.highlight_found_words()
            self.selected_word=""
            self.selected_cells=[]
            self.score = len(self.found_words)
            self.score_label=tk.Label(self,text=f"Score:
{self.score}",bg="lavender",fg="black",font=("Times New Roman","20"))
            self.score_label.place(x=750,y=580)
            messagebox.showinfo("found","Congratulations!! , You found a word.\n More {} words to
find".format(5-len(self.found_words)))

        else:
            print("false")
            cell_width = 400 / GRID_SIZE
            cell_height = 400 / GRID_SIZE
```

```
            for self.x, self.y in self.selected_cells:
                x0 = self.x * cell_width
                y0 = self.y * cell_height
                x1 = (self.x + 1) * cell_width
                y1 = (self.y + 1) * cell_height
                self.canvas.create_rectangle(x0,y0,x1,y1, fill="white")
            for self.x,self.y in self.selected_cells:
                x0 = self.x * cell_width
                y0 = self.y * cell_height
                letter = self.grid[self.y][self.x]
                self.canvas.create_text(x0 + cell_width / 2, y0 + cell_height / 2, text=letter, font=("Arial",
12,'bold'))
            self.selected_word=""
            self.selected_cells=[]
            messagebox.showinfo("not found","Oops!! WRONG WORD You lost a chance.")
        print(self.found_words)
        if len(self.found_words) == len(self.word_list):
            self.game_over()



    def update_score(self):
        score_label=tk.Label(self)
        score = len(self.found_words)
        score_label.configure(text=f"Score: {score}")
    def highlight_found_words(self):
        self.clr=random.choice(colours)
        cell_width = 400 / GRID_SIZE
        cell_height = 400 / GRID_SIZE

        for self.x, self.y in self.selected_cells:
            x0 = self.x * cell_width
```

```
            y0 = self.y * cell_height
            x1 = (self.x + 1) * cell_width
            y1 = (self.y + 1) * cell_height
            self.canvas.create_rectangle(x0,y0,x1,y1, outline="blue",fill=self.clr)


        for self.x,self.y in self.selected_cells:
            x0 = self.x * cell_width
            y0 = self.y * cell_height
            letter = self.grid[self.y][self.x]
            self.canvas.create_text(x0 + cell_width / 2, y0 + cell_height / 2, text=letter, font=("Arial",
12,'bold'))



    def game_over(self):
        self.canvas.unbind("<Button-1>")
        self.canvas.create_text(200, 200, text="Game Over!", font=("Arial", 24))


button=tk.Button(window,text="Play",bg = "blue",fg = "white",height = '2',width = '12',font=("Times
New Roman",15,'bold'),command=lambda:[window.withdraw(),WordSearchGame()])
button.place(x=580,y=620)
button=tk.Button(window,text="Exit",bg = "lavender",fg = "black",height = '2',width =
'12',font=("Times New Roman",15,'bold'),command=lambda:window.destroy())
button.place(x=880,y=620)


window.mainloop()
```