

```
import random
import time
import pickle
from tkinter import *
from PIL import Image, ImageTk
import tkinter as tk

respostasesperadas = ["Fuja", "Ataque"]

#definindo a classe inspermon
class Inspermon():
    def __init__(self, ataque, defesa, vida, experiencia):
        self.a = ataque
        self.d = defesa
        self.pv = vida
        self.exp = experiencia

#definindo a função que padronizar a resposta
def padroniza(resposta):
    resposta = resposta.strip(" ")
    resposta = resposta.title()
    return resposta

#definindo a função de batalha
def batalha(inspermon1, inspermon2, bd, experiencia_atual):
    resultado1 = [bd[inspermon2].pv]
    resultado2 = [bd[inspermon1].pv]
    for i in range (1, 1000):
        print("\n11 o seu turno, preste atenção!")
        time.sleep(1)
        batalhando = input("Seu inspermon tem duas escolhas, mande ele ao (ataque), ou fale: (fuja)! ")
        batalhando = padroniza(batalhando)
        dano1 = bd[inspermon1].a - bd[inspermon2].d
        dano2 = bd[inspermon2].a - bd[inspermon1].d
        if batalhando == "Ataque":
            resultado1.append(resultado1[i-1] - dano1)
            resultado2.append(resultado2[i-1] - dano2)
            print("Seu Inspermon está atacando")
            time.sleep(2)
            if resultado1[i] > 0:
                print("Os pontos de vida de seu inimigo foram de {0} para {1}".format(resultado1[i-1], resultado1[i]))
            if resultado1[i] <= 0:
                print("Os pontos de vida de seu inimigo foram de {0} para 0".format(resultado1[i-1]))
                print("{0} desmaiou, {1} é o vencedor da batalha!".format(inspermon2, inspermon1))
                time.sleep(2)
                sortepv = random.randint(0, 6)
```

```

        if sortepv < 2:
            print ("Oh não, seu inimigo tinha uma carta na manga! Ele tomou uma poção e seus po
ntos de vidas foram regenerados em 50!")
            resultado1[i] = 50
        if sortepv >= 2:
            print("Continuando a sua aventura, os pontos de vida de seu inspermon foram regenerado
s")

            time.sleep(2)
            experiencia_nova = experiencia_atual + bd[inspermon2].exp
            return experiencia_nova

time.sleep(2)
print("Agora é o turno de seu inimigo, se prepare para um ataque!")
time.sleep(2)
print("...")
time.sleep(1)
if resultado2[i] > 0:
    print("Seus pontos de vida foram de {0} para {1}".format(resultado2[i-1], resultado2[i]))
if resultado2[i] <= 0:
    print("Seus pontos de vida foram de {0} para 0".format(resultado2[i-1]))
    print ("Seu inspermon desmaiou, que pena... Espere até que ele se regenere")
    time.sleep(2)
    print("...")
    time.sleep(3)
    return experiencia_atual
#criando uma probabilidade de fuga da batalha
if batalhando == "Fuja":
    resultado1 = resultado1 + [resultado1[i-1]]
    resultado2 = resultado2 + [resultado2[i-1]]
    print ("Seu inspermon tentará fugir!")
    time.sleep (2)
    probD = random.randint(0, 5)
    if probD in range(0, 3):
        print("Ele conseguiu!")
        return experiencia_atual
    else:
        resultado2[i] = resultado2[i-1] - dano2
        print ("Seu inspermon falhou, e por isso perdeu um turno... vamos continuar a batalha!")
        time.sleep(2)
        if resultado2[i] > 0:
            print("Seu inimigo te atacou e voce foi de {0} para {1} pontos de vida!".format(result
ado2[i-1], resultado2[i]))
        else:
            print("Seu inimigo te atacou e você foi de {0} para 0 pontos de vida".format(resultad
o2[i-1]))

            time.sleep(2)
            print("Infelizmente seu inspermon desmaiou. Espere ele se regenerar!")
            time.sleep(1)
            print("...")
            time.sleep(4)

```

```
        return experiencia_atual
    #caso o jogador escreva algo diferente do esperado
    if batalhando not in respostasesperadas:
        resultado1 = resultado1 + [resultado1[i-1]]
        resultado2 = resultado2 + [resultado2[i-1]]
        print ("Desculpa, não entendemos o que você quer fazer")

def SaveGame(insperdex, experiencia, inspermon_inicial, nome_do_save):
    arq = open(nome_do_save + "1.txt", 'wb')
    pickle.dump(insperdex, arq)
    arq.close()
    brt = open(nome_do_save + "2.txt", 'w')
    brt.write(inspermon_inicial)
    crt = open(nome_do_save + "3.txt", 'w')
    crt.write(str(experiencia))

def LoadGame(nome_do_save):
    arq = open(nome_do_save + "1.txt", 'rb')
    dic = pickle.load(arq)
    arq.close()
    brt = open(nome_do_save + "2.txt", "r")
    inspermon_inicial = brt.readlines(1)
    crt = open(nome_do_save + "3.txt", "r")
    experiencia = crt.readlines(1)
    return dic, inspermon_inicial, experiencia

#adicionando a interface gráfica;fica
def Play():
    window.destroy()

window = tk.Tk()
window.title ("Inspermon")
window.geometry("700x600+500+500")
pokebola = 'pokebola.gif'
img = ImageTk.PhotoImage(Image.open(pokebola))
foto = tk.Label(window, image = img)
foto.pack(side = "top", fill = "both", expand = "yes")
texto = tk.Label(window, text = "Bem vindo ao jogo Inspermon!")
texto.pack()
botão = tk.Button(window)
botão.configure (text = "Vamos Jogar!")
botão.configure (command = Play)
botão.pack(side = tk.BOTTOM)
window.mainloop()
```

