

```
import pickle
import random
import time
insperdex={
    "Pikaxu":{
        "Ataque":50, "Defesa":30, "Vida":200, "Exp":15, "Chance":10, "Evo":20
    },
    "Kapuznakara":{
        "Ataque":70, "Defesa":30, "Vida":200, "Exp":20, "Chance":7, "Evo":25
    },
    "Xanaina":{
        "Ataque":105, "Defesa":10, "Vida":230, "Exp":30, "Chance":2, "Evo":30
    },
    "Kingnaldo":{
        "Ataque":120, "Defesa":15, "Vida":180, "Exp":35, "Chance":2, "Evo":35
    },
    "Douglas":{
        "Ataque":90, "Defesa":25, "Vida":210, "Exp":30, "Chance":2, "Evo":30
    },
    "Cetaxu":{
        "Ataque":70, "Defesa":50, "Vida":220, "Exp":30, "Chance":0, "Evo":1000
    },
    "Bonenakara":{
        "Ataque":70, "Defesa":50, "Vida":220, "Exp":30, "Chance":0, "Evo":1000
    },
    "Xanalna":{
        "Ataque":120, "Defesa":30, "Vida":240, "Exp":30, "Chance":0, "Evo":1000
    },
    "Godnaldo":{
        "Ataque":140, "Defesa":35, "Vida":200, "Exp":30, "Chance":0, "Evo":1000
    },
    "Showglas":{
        "Ataque":110, "Defesa":45, "Vida":230, "Exp":30, "Chance":0, "Evo":1000
    }
}
#Dicionários dos Inspermons

dexjog={} #InsperDex
seus_inspermons=[] #Lista com seus inspermons

def save(arquivo):
    dados= open(arquivo,'wb')
    pickle.dump({"dados" : [dexjog, exp, jogador,seus_inspermons,niveljog,expj]}, dados)
    dados.close()

def load(arquivo):
    dado=pickle.load(open(arquivo,"rb"))
    return dado
```

```
def critico(): #Função Critico
    if random.randint(1,100)<= 20:
        return True
    else:
        return False

def nivelopo(niveljog):
    nvlopo=random.randint(niveljog-1,niveljog+1)
    return nvlopo

def luta(): #Função Fuga
    if random.randint(1,10) <=3:
        return True
    else:
        return False

def captura(oponente,vidaopo): #Função captura
    if random.randint(1,100) >=int(round((vidaopo/insperdex[oponente]["Vida"])*85+10)):
        return True
    else:
        return False

lista_evo=[]
lista_chance=[] #Lista de chance de encontro no passeio
for i in insperdex:
    if insperdex[i]["Chance"]>0:
        x=0
        while x<insperdex[i]["Chance"]:
            lista_chance.append(i)
            x=x+1
    elif insperdex[i]["Chance"]<=0:
        lista_evo.append(i)

def roll(): #Randomizador
    poke= random.randint(0, len(lista_chance)-1)
    aleatorio=lista_chance[poke]
    return aleatorio

def experiencia(jogador, oponente, exp):
    expganha=insperdex[oponente]["Exp"]
    posicao=seus_inspermons.index(jogador)
```

```
xpi=exp[posicao]
xptotal=xpi+expganha
exp.remove(exp[posicao])
exp.insert(posicao,xptotal)
return xptotal

def nivel(niveljog, expj):
    expj = expj + insperdex[oponente]["Exp"]
    return expj

def evolucao(jogador):
    if jogador == "Pikaxu":
        posicaopok=seus_inspermons.index("Pikaxu")
        seus_inspermons.remove("Pikaxu")
        seus_inspermons.insert(posicaopok,"Cetaxu")
        return "Cetaxu"
    if jogador == "Kapuznakara":
        posicaopok=seus_inspermons.index("Kapuznakara")
        seus_inspermons.remove("Kapuznakara")
        seus_inspermons.insert(posicaopok,"Bonenakara")
        return "Bonenakara"
    if jogador == "Xanaina":
        posicaopok=seus_inspermons.index("Xanaina")
        seus_inspermons.remove("Xanaina")
        seus_inspermons.insert(posicaopok,"Xanalna")
        return "Xanalna"
    if jogador == "Kingnaldo":
        posicaopok=seus_inspermons.index("Kingnaldo")
        seus_inspermons.remove("Kingnaldo")
        seus_inspermons.insert(posicaopok,"Godnaldo")
        return "Godnaldo"
    if jogador == "Douglas":
        posicaopok=seus_inspermons.index("Douglas")
        seus_inspermons.remove("Douglas")
        seus_inspermons.insert(posicaopok,"Showglas")
        return "Showglas"

def batalha(jogador,oponente,niveljog): #Função Batalha
    if niveljog>1:
        nvlopo=nivelopo(niveljog)
    elif niveljog==1:
        nvlopo=1
    listavida=[]
    for i in seus_inspermons:
        listavida.append(insperdex[i]["Vida"]+10*niveljog)
    vidajog=listavida[seus_inspermons.index(jogador)]
    vidaopo=insperdex[oponente]["Vida"]+10*nvlopo
    print("Você encontrou um {} selvagem nã-vel {}".format(oponente,nvlopo))
```

```

time.sleep(1.0)
print("O seu oponente tem: ")
print("Ataque: {}".format(insperdex[oponente]["Ataque"]))
print("Defesa: {}".format(insperdex[oponente]["Defesa"]))
print("Vida: {}".format(insperdex[oponente]["Vida"]))
print("E da {} de experiÃancia".format(insperdex[oponente]["Exp"]))

if oponente not in dexjog:
    dexjog[oponente]=insperdex[oponente]
time.sleep(1.5)

while vidajog>0 and vidaopo>0:

    if jogador == oponente or oponente in seus_inspermons:
        acao=str(input("Qual o seu comando? (Lutar(L) ou Fugir(F) ou Trocar Inspermon(T))")).lower() #Escolha
de comando

    else:
        acao=str(input("Qual o seu comando? (Lutar(L), Fugir(F), Capturar(C) ou Trocar Inspermon(T))")).lower(
) #Escolha de comando

    if acao == "lutar" or acao == "l":
        time.sleep(1.0)

        if ((insperdex[jogador]["Ataque"]+10*niveljog)-(insperdex[oponente]["Defesa"]+10*nvlopo))>0:

            if vidajog>0: #Ataque Jogador

                if critico()==True:
                    vidaopo=vidaopo-((insperdex[jogador]["Ataque"]+10*niveljog)-(insperdex[oponent
e]["Defesa"]+10*nvlopo))*1.5

                if vidaopo>0:
                    print("O {} leva um ataque CRÃ\215TICO e fica com {} de vida".format(o
ponente,vidaopo))
                    time.sleep(1.0)

                elif vidaopo<=0:
                    print("O {} leva um ataque CRÃ\215TICO e desmaia!".format(oponente))
                    time.sleep(1.0)

            else:
                vidaopo=vidaopo-((insperdex[jogador]["Ataque"]+10*niveljog)-(insperdex[oponent
e]["Defesa"]+10*nvlopo))

                if vidaopo>0:
                    print("O {} leva o ataque e fica com {} de vida".format(oponente,vidao
po))
                    time.sleep(1.0)

```

```

        elif vidaopo<=0:
            print("O {} leva o ataque e desmaia!".format(oponente))
            time.sleep(1.0)

    elif ((insperdex[jogador]["Ataque"]+10*niveljog)-(insperdex[oponente]["Defesa"]+10*nvlopo))<=0:
        print("O seu ataque não deu dano!")

    if ((insperdex[oponente]["Ataque"]+10*nvlopo)-(insperdex[jogador]["Defesa"]+10*niveljog))>0:

        if vidaopo>0: #Ataque Oponente
            print("O {} selvagem se prepara para atacar!".format(oponente))
            time.sleep(1)

            if critico()==True:
                vidajog=vidajog-((insperdex[oponente]["Ataque"]+10*nvlopo)-(insperdex[jogador]
["Defesa"]+10*niveljog))*1.5

                if vidajog>0:
                    print("O seu {} Ã© atacado CRITICAMENTE e fica com {} de vida".format(
jogador,vidajog))
                    time.sleep(1.0)

                elif vidajog<=0:
                    print("O seu {} Ã© atacado CRITICAMENTE e desmaia!".format(jogador))
                    time.sleep(0.5)
                    print("O seu Inspermon foi levado ao InsperCenter!")
                    time.sleep(1.0)

            else:
                vidajog=vidajog-((insperdex[oponente]["Ataque"]+10*nvlopo)-(insperdex[jogador]
["Defesa"]+10*niveljog))

                if vidajog>0:
                    print("O seu {} Ã© atacado e fica com {} de vida".format(jogador,vidaj
og))
                    time.sleep(1.0)

                elif vidajog<=0:
                    print("O seu {} Ã© atacado e desmaia!".format(jogador))
                    time.sleep(1.0)

    elif ((insperdex[oponente]["Ataque"]+10*nvlopo)-(insperdex[jogador]["Defesa"]+10*niveljog))<=0:
        print("O ataque do {} não deu dano!".format(oponente))

    if vidaopo<=0: #Resultado Batalha
        print(("Você venceu! Seu {} ganhou {} de exp!".format(jogador,insperdex[oponente]["Exp"])))
        time.sleep(1.0)

```

```

        return True

    elif vida_jog<=0:
        print("Você perdeu e seu Inspemon foi levado ao InsperCenter!")
        time.sleep(1.0)
        return False

    if acao == "fugir" or acao == "f": #Tentativa de Fuga
        time.sleep(1.0)

    if luta()==True:
        print("Fuga Falhou!")
        time.sleep(1.0)

        if ((insperdex[oponente]["Ataque"]+10*nvlopo)-(insperdex[jogador]["Defesa"]+10*niveljog))>0:

            if vidaopo>0: #Ataque Oponente
                print("O {} selvagem se prepara para atacar!".format(oponente))
                time.sleep(1)

                if critico()==True:
                    vida_jog=vida_jog-((insperdex[oponente]["Ataque"]+10*nvlopo)-(insperdex[
jogador]["Defesa"]+10*niveljog))*1.5

                    if vida_jog>0:
                        print("O seu {} Ã© atacado CRITICAMENTE e fica com {} de vida"
.format(jogador,vida_jog))
                        time.sleep(1.0)

                    elif vida_jog<=0:
                        print("O seu {} Ã© atacado CRITICAMENTE e desmaia!".format(jog
ador))
                        time.sleep(1.0)

                else:
                    vida_jog=vida_jog-((insperdex[oponente]["Ataque"]+10*nvlopo)-(insperdex[
jogador]["Defesa"]+10*niveljog))

                    if vida_jog>0:
                        print("O seu {} Ã© atacado e fica com {} de vida".format(jogad
or,vida_jog))
                        time.sleep(1.0)

                    elif vida_jog<=0:
                        print("O seu {} Ã© atacado e desmaia!".format(jogador))
                        time.sleep(1.0)

            elif ((insperdex[oponente]["Ataque"]+10*nvlopo)-(insperdex[jogador]["Defesa"]+10*niveljog))<=0
:

```

```

        print("O ataque do {} não deu dano!".format(oponente))
    else:
        print("Fuga Sucedida")
        time.sleep(1.0)
        break

if acao == "capturar" or acao == "c": #capturar inspermon
    time.sleep(1.0)

    if jogador != oponente and oponente not in seus_inspermons:
        cc= captura(oponente,vidaopo)

        if cc == True:
            seus_inspermons.append(oponente)
            exp.append(0)
            print("Parabéns! Você agora possui {} como um de seus inspermons!".format(oponente))
            break

        if cc == False:
            print("Foi por pouco! {} se recusa a ter sua liberdade retirada!".format(oponente))

            if ((insperdex[oponente]["Ataque"]+10*nvlopo)-(insperdex[jogador]["Defesa"]+10*niveljog))>0:

                if vidaopo>0: #Ataque Oponente
                    print("O {} selvagem se prepara para atacar!".format(oponente))
                    time.sleep(1)

                    if critico()==True:
                        vidajog=vidajog-((insperdex[oponente]["Ataque"]+10*nvlopo)-(insperdex[jogador]["Defesa"]+10*niveljog))*1.5

                        if vidajog>0:
                            print("O seu {} foi atacado CRITICAMENTE e fica com {} de vida".format(jogador,vidajog))
                            time.sleep(1.0)

                        elif vidajog<=0:
                            print("O seu {} foi atacado CRITICAMENTE e desmaia!".format(jogador))
                            time.sleep(1.0)

                    else:
                        vidajog=vidajog-((insperdex[oponente]["Ataque"]+10*nvlopo)-(insperdex[jogador]["Defesa"]+10*niveljog))

                        if vidajog>0:
                            print("O seu {} foi atacado e fica com {} de vida".format(jogador,vidajog))

```

```

time.sleep(1.0)

elif vidajog<=0:
    print("O seu {} Ã© atacado e desmaia!".format(jogador))

)

time.sleep(1.0)

elif ((insperdex[oponente]["Ataque"]+10*nvlopo)-(insperdex[jogador]["Defesa"]+10*nivel
jog))<=0:

    print("O ataque do {} nÃ£o deu dano!".format(oponente))

else:
    print("VocÃª jÃ¡ possui esse inspermon!")

if acao == "trocar inspermon" or acao == "t":
    listavida[seus_inspermons.index(jogador)]=vidajog
    print("Seus inspermons atuais sÃ£o:")
    for h in range(0,len(seus_inspermons)):
        print("{}({})".format(seus_inspermons[h],h+1))
    time.sleep(0.5)
    hh=int(input("Insira o numero do inspermon desejado :"))
    jogador=seus_inspermons[hh-1]
    vidajog=listavida[seus_inspermons.index(jogador)]
    time.sleep(1.5)
    print("Seu inspermon agora Ã© {}".format(jogador))

if ((insperdex[oponente]["Ataque"]+10*nvlopo)-(insperdex[jogador]["Defesa"]+10*niveljog))>0:

    if vidaopo>0: #Ataque Oponente
        print("O {} selvagem se prepara para atacar!".format(oponente))
        time.sleep(1)

        if critico()==True:
            vidajog=vidajog-((insperdex[oponente]["Ataque"]+10*nvlopo)-(insperdex[jogador]
["Defesa"]+10*niveljog))*1.5

            if vidajog>0:
                print("O seu {} Ã© atacado CRITICAMENTE e fica com {} de vida".format(
jogador,vidajog))

                time.sleep(1.0)

            elif vidajog<=0:
                print("O seu {} Ã© atacado CRITICAMENTE e desmaia!".format(jogador))
                time.sleep(1.0)

        else:
            vidajog=vidajog-((insperdex[oponente]["Ataque"]+10*nvlopo)-(insperdex[jogador]
["Defesa"]+10*niveljog))

```



```
og))

        if vidajog>0:
            print("O seu {} Ã© atacado e fica com {} de vida".format(jogador,vidajog))
            time.sleep(1.0)

        elif vidajog<=0:
            print("O seu {} Ã© atacado e desmaia!".format(jogador))
            time.sleep(1.0)

    elif ((insperdex[oponente]["Ataque"]+10*nvlopo)-(insperdex[jogador]["Defesa"]+10*niveljog))<=0:
        print("O ataque do {} nÃ£o deu dano!".format(oponente))

expj=0 #experiencia jogador
niveljog=1
exp=[] #experiencia pokemons
lista_saves=[]
print("Bem Vindo ao Mundo de Inspermon!")
time.sleep(0.5)
while True:
    inicio=str(input("New Game(N) ou Load Game(L)? ")).lower()

    if inicio=="new game" or inicio=="n":
        try:
            jogador=str(input("Qual seu Inspermon inicial? (Xanaina, Kingnaldo ou Douglas) ")).title()
            dexjog[jogador]=insperdex[jogador]
            seus_inspermons.append(jogador)
            exp.append(0)
            time.sleep(0.5)
            break
        except: #se o usuario digitar o nome do inspermon errado, o codigo nao para
            continue

    if inicio=="load game" or inicio=="l":
        try:
            lista_saves=pickle.load(open("lista","rb"))
            if len(lista_saves)>0:
                print("Os saves disponÃveis sÃ£o: {}".format(lista_saves))
                time.sleep(1)
                file=str(input("Qual save deseja carregar? "))
                dado=load(file)
                dexjog=dado["dados"][0]
                exp=dado["dados"][1]
                jogador=dado["dados"][2]
                seus_inspermons=dado["dados"][3]
                niveljog=dado["dados"][4]
                expj=dado["dados"][5]
                print("Carregando....")
                time.sleep(2)
```

```
        print("Sucesso!")
        time.sleep(0.5)
        print("O seu inspermon Ã© o: {}".format(jogador))
        break
    except:
        print("NÃ£o hÃ¡ saves ainda!")

while True:
    fazer=str(input("O que vocÃª vai fazer? (Passear(P), Dormir(D), Insperdex(I), Salvar(S), Load(L) ou Trocar Inspermon(T)
)? ").lower() #Escolha de aÃ§Ã£o
    time.sleep(1.0)

    if fazer == "passear" or fazer == "p": #Caso Passeie
        print("Passeando...")
        time.sleep(1.5)
        oponente=roll()
        expneeded = 50*niveljog*(1.1**niveljog)

        if batalha(jogador,oponente,niveljog)==True:
            expe = experiencia(jogador, oponente, exp)
            expj = nivel(niveljog,expj)
            posicao=seus_inspermons.index(jogador)
            print("Seu {} possui {} de experiÃªncia".format(jogador,exp[posicao]))
            time.sleep(0.5)

            if expe >=insperdex[jogador]["Evo"]:
                jogador=evolucao(jogador)
                print(".")
                time.sleep(0.2)
                print(".")
                time.sleep(0.2)
                print(".")
                time.sleep(0.2)
                print("O seu inspermon evolui para...")
                time.sleep(0.5)
                print("{} !!".format(jogador)).upper()
                time.sleep(2.0)
                print("")
                print("Os stats do {} sÃ£o:".format(jogador))
                print("Ataque: {}".format(insperdex[jogador]["Ataque"]))
                print("Defesa: {}".format(insperdex[jogador]["Defesa"]))
                print("Vida: {}".format(insperdex[jogador]["Vida"]))

            dexjog[jogador]=insperdex[jogador]

        if expneeded <= expj and niveljog < 10:
            niveljog = niveljog + 1
            time.sleep(1)
```

```
        print("Parabens! VocÃª passou de nÃ-vel!".format(expneeded))
        time.sleep(1)
        print("Agora seu nÃ-vel Ã© {}".format(niveljog))
        time.sleep(1)
        if niveljog == 10:
            print("VocÃª atingiu o nÃ-vel mÃximo!")
            time.sleep(1)

if fazer == "dormir" or fazer == "d": #Caso durma
    print("Bons Sonhos!")
    break

if fazer == "insperdex" or fazer == "i": #Caso cheque Insperdex
    print("Este sÃo os InspÃrmons que vocÃ jÃ encontrou: {}".format(dexjog))
    time.sleep(0.5)

if fazer == "trocar inspermon" or fazer == "t":
    print("Seus inspermons atuais sÃo:")
    for h in range(0, len(seus_inspermons)):
        print("{}({})".format(seus_inspermons[h], h+1))
        time.sleep(0.5)
    hh=int(input("Insira o numero do inspermon desejado :"))
    jogador=seus_inspermons[hh-1]
    time.sleep(1.5)
    print("Seu inspermon agora Ã© {}".format(jogador))

if fazer == "salvar" or fazer == "s":
    print("Os saves disponÃveis sÃo: {}".format(lista_saves))
    arquivo=str(input("Deseja salvar com que nome? "))
    lista_saves.append(arquivo)
    save(arquivo)
    saves=open("lista", "wb" )
    pickle.dump(lista_saves, saves)
    saves.close()
    print("Salvando....")
    time.sleep(2)
    print("Sucesso!")
    time.sleep(0.5)

if fazer == "load" or fazer == "l":
    try:
        lista_saves=pickle.load(open("lista", "rb"))
        if len(lista_saves)>0:
            print("Os saves disponÃveis sÃo: {}".format(lista_saves))
            time.sleep(1)
            file=str(input("Qual save deseja carregar? "))
            dado=load(file)
            dexjog=dado["dados"][0]
            exp=dado["dados"][1]
```

```
jogador=dado["dados"][2]
seus_inspemons=dado["dados"][3]
niveljog=dado["dados"][4]
expj=dado["dados"][5]
print("Carregando....")
time.sleep(2)
print("Sucesso!")
time.sleep(0.5)
except:
    print("NÃO hã; saves ainda!")
```