



Jogo de Damas em Python

Alunas:

Nikolle Gambaro Coutinho

Stephany Correa Godoy



■ Por que um Jogo de Damas?

- Elaboramos um jogo de damas por ser um jogo simples e divertido.

Objetivo do trabalho

- Obter um Jogo de Damas programado na linguagem pythom onde os usuários possam competir entre si para que no final haja um campeão (rosas ou girasol)

O código

```
import pygame, sys, math

VAZIO, NORMAL_ROSA, NORMAL_AMARELO, Q_ROSA, Q_AMARELO, SELECIONADA = 0,1, 2, 3, 4, 5


#os numeros definem as cores, e ja estao pre definidos
PRETO=(0,0,0)
BRANCO=(255,255,255)
AMARELO=(255,255,0)
CINZA=(205, 193, 197)
ROSA=(255,62,150)

# LARGURA e ALTURA das celulas do tabuleiro
LARGURA=65

ALTURA=65
RAIO=30

margem=0 # margem entre celulas do tabuleiro
tamanhoJanela=[520, 520]

#Inicializando pygame e recursos importantes
pygame.init()
screen = pygame.display.set_mode(tamanhoJanela)
pygame.display.set_caption("Jogo de Damas") #mudar o nome na janela do jogo
fim = False
pygame.font.init()
myfont = pygame.font.SysFont('Comic Sans MS', 30)#so define a fonte mesmo
```

```

# funcoes auxiliares e facilitadoras:
def cor_quadrado(linha, col): # determinar cor dos quadrados.
    return BRANCO if (linha + col) % 2 == 0 else PRETO #Quadrado par eh branco

def getColunaClick(mouse_pos):
    x = mouse_pos[0]
    for i in range(1, 8):
        if x < (i * ALTURA):
            return i - 1
    return 7

def getLinhaClick(mouse_pos):
    y = mouse_pos[1]
    for i in range(1, 8):
        if y < (i * LARGURA):
            return i - 1
    return 7

def reDesenharTabuleiro(tabuleiro):
    desenharTabuleiro()
    desenharPecas(tabuleiro)
    pygame.display.flip() #atualiza o conteudo da exibicao

def mudarPecaLugar(linha, coluna, pecaSelecionada):
    tabuleiro[pecaSelecionada[0]][pecaSelecionada[1]] = VAZIO
    tabuleiro[linha][coluna] = pecaSelecionada[2]

def trocaTurno(turno):
    if turno==NORMAL_ROSA:
        return NORMAL_AMARELO
    else:
        return NORMAL_ROSA

```

```
#desenhar quadrados do tabuleiro
def desenharTabuleiro():
    for tabLinha in range(8):
        for tabColuna in range(8):
            xCoordenada=(margem+LARGURA) * tabColuna + margem
            yCoordenada=(margem+ALTURA) * tabLinha + margem
            currentColour = cor_quadrado(tabLinha, tabColuna)
            pygame.draw.rect(screen,currentColour,[xCoordenada,yCoordenada, LARGURA, ALTURA])
            #desenha os retangulos que estao do tabuleiro

# Reseta o jogo
def novoTabuleiro():
    tabuleiro = [[0,1,0,1,0,1,0,1],
                  [1,0,1,0,1,0,1,0],
                  [0,1,0,1,0,1,0,1],
                  [0,0,0,0,0,0,0,0],
                  [0,0,0,0,0,0,0,0],
                  [2,0,2,0,2,0,2,0],
                  [0,2,0,2,0,2,0,2],
                  [2,0,2,0,2,0,2,0]]

    return tabuleiro
```

```

#desenha pecas no tabuleiro
def desenharPecas(tabuleiro):
    for y in range(8):
        for x in range(8):
            # se o lugar no tabuleiro nao for vazio, desenha um circulo para indicar a peca
            if(tabuleiro[x][y] != VAZIO):
                #desenhar nas coordenadas com um pequeno ajuste grafico
                xCoordenada=(margem+LARGURA) * x +33
                yCoordenada=(margem+ALTURA) * y +33
                #desenhar os circulos nas cores certas
                if tabuleiro[x][y]==NORMAL_ROSA:
                    pygame.draw.circle(screen,ROSA,(yCoordenada,xCoordenada),RAIO) #desenha pecas
                elif tabuleiro[x][y]==NORMAL_AMARELO:
                    pygame.draw.circle(screen,AMARELO,(yCoordenada,xCoordenada),RAIO)
                elif tabuleiro[x][y]==Q_AMARELO:
                    pygame.draw.circle(screen,AMARELO,(yCoordenada,xCoordenada),RAIO)
                    # letra Q para indicar a rainha
                    letraQ = myfont.render('Q', False, PRETO)
                    screen.blit(letraQ, (yCoordenada-25,xCoordenada-33))
                elif tabuleiro[x][y]==Q_ROSA:
                    pygame.draw.circle(screen,ROSA,(yCoordenada,xCoordenada),RAIO)
                    # letra Q para indicar a rainha
                    letraQ = myfont.render('Q', False, PRETO)
                    screen.blit(letraQ, (yCoordenada-25,xCoordenada-33))
                elif tabuleiro[x][y]==SELECIONADA:
                    pygame.draw.circle(screen,CINZA,(yCoordenada,xCoordenada),RAIO)

```

```
def isFimDeJogo(tabuleiro):
    rosa=0
    amarelo=0
    for y in range(8):
        for x in range(8):
            if (tabuleiro[y][x]==NORMAL_ROSA):
                rosa+=1
            elif (tabuleiro[y][x]==NORMAL_AMARELO):
                amarelo+=1
    if (rosa == 0):
        print("Acabou - As Girassol Venceram!!!")
        return True
    elif (amarelo==0):
        print("Acabou - As Rosas Venceram!!!")
        return True
    return False
```



```
# ----- Inicio -----
```

```
#rosa eh o primeiro a jogar
```

```
turno = NORMAL_ROSA
```

```
pecaSelecionada = None
```

```
screen.fill(CINZA)#essa eh a cor de fundo
```

```
tabuleiro = novoTabuleiro()
```

```
desenharTabuleiro()
```

```
desenharPecas(tabuleiro)
```

```
pygame.display.flip()
```

```
while not fim:
```

```
    #aguarda o evento do mouse ser pressionado, inicia verificacoes
```

```
    for event in pygame.event.get():
```

```
        if event.type == pygame.QUIT:
```

```
            fim = True
```

```
        elif event.type == pygame.MOUSEBUTTONDOWN:
```

```
            #pega posicao do click do mouse
```

```
            posClick = pygame.mouse.get_pos()
```

```
            # relacionar linha e coluna do tabuleiro com posicao do ponteiro do mouse
```

```
            linha, coluna = getLinhaClick(posClick), getColunaClick(posClick)
```

```
#----- CORPO LOGICO DO JOGO -----
```

```
#Ja existe uma peca selecionada? ENTAO EH UM MOVIMENTO!
```

```
if(pecaSelecionada is not None):
```

```
    if(pecaSelecionada[0]!=linha and pecaSelecionada[1]!=coluna):
```

```
        #se o espaco clicado for vazio entao da pra mover a peca
```

```
        if(tabuleiro[linha][coluna]==VAZIO):
```

```
            #mover a peca apenas para a vizinhanca correta dependendo da cor!
```

```
            if((pecaSelecionada[2]==NORMAL_ROSA and linha==(pecaSelecionada[0]+1)) or  
               (pecaSelecionada[2]==NORMAL_AMARELO and linha==(pecaSelecionada[0]-1))):
```

```
                if((coluna==pecaSelecionada[1]-1) or (coluna==pecaSelecionada[1]+1)):  
                    #so movimenta aqui
```

```
                    mudarPecaLugar(linha,coluna,pecaSelecionada)
```

```
                    reDesenharTabuleiro(tabuleiro)
```

```
                    #rodar os turnos, so pode ser feito no fim da jogada
```

```
                    turno = trocaTurno(turno)
```

```
                    pecaSelecionada=None
```

```
            #verificar se EH UM MOVIMENTO DE CAPTURA
```

```

elif((pecaSelecionada[2]==NORMAL_ROSA and linha==(pecaSelecionada[0]+2)) or
(pecaSelecionada[2]==NORMAL_AMARELO and linha==(pecaSelecionada[0]-2))):
    if(coluna==(pecaSelecionada[1]-2)):
        if(pecaSelecionada[2]==NORMAL_AMARELO and (tabuleiro[linha+1][coluna+1] not in [VAZIO, pecaSelecionada[2]])):
            tabuleiro[linha+1][coluna+1] = 0
            #coisas do movimento (inicio
            mudarPecaLugar(linha,coluna,pecaSelecionada)
            reDesenharTabuleiro(tabuleiro)
            #rodar os turnos, so pode ser feito no fim da jogada
            turno = trocaTurno(turno)
            pecaSelecionada=None
            #fim das coisas do movimento
        elif(pecaSelecionada[2]==NORMAL_ROSA and (tabuleiro[linha-1][coluna+1] not in [VAZIO, pecaSelecionada[2]])):
            tabuleiro[linha-1][coluna+1]= 0
            #coisas do movimento (inicio
            mudarPecaLugar(linha,coluna,pecaSelecionada)
            reDesenharTabuleiro(tabuleiro)
            #rodar os turnos, so pode ser feito no fim da jogada
            turno = trocaTurno(turno)
            pecaSelecionada=None
            #fim das coisas do movimento)

```

```

elif(coluna == (pecaSelecionada[1]+2)):
    if(pecaSelecionada[2]==NORMAL_AMARELO and (tabuleiro[linha+1][coluna-1] not in [VAZIO, pecaSelecionada[2]])):
        tabuleiro[linha+1][coluna-1]=0
        #coisas do movimento (inicio
        mudarPecaLugar(linha,coluna,pecaSelecionada)
        reDesenharTabuleiro(tabuleiro)
        #rodar os turnos, so pode ser feito no fim da jogada
        turno = trocaTurno(turno)
        pecaSelecionada=None
        #fim das coisas do movimento)
    elif(pecaSelecionada[2]==NORMAL_ROSA and (tabuleiro[linha-1][coluna-1] not in [VAZIO, pecaSelecionada[2]])):
        tabuleiro[linha-1][coluna-1]=0
        #coisas do movimento (inicio
        mudarPecaLugar(linha,coluna,pecaSelecionada)
        reDesenharTabuleiro(tabuleiro)
        #rodar os turnos, so pode ser feito no fim da jogada
        turno = trocaTurno(turno)
        pecaSelecionada=None
        #fim das coisas do movimento)

```

```
else:
    tabuleiro[pecaSelecionada[0]][pecaSelecionada[1]] = pecaSelecionada[2]
    pecaSelecionada=None
    reDesenharTabuleiro(tabuleiro)
if(isFimDeJogo(tabuleiro)):
    fim = True
    pygame.quit()
```

```
#SE NAO EXISTE, o jogador clicou num lugar vazio?
elif(tabuleiro[linha][coluna] != VAZIO):
    #se nao clicou, entao ELE SELECIONOU UMA PECA!!
    #ELE SELECIONOU UMA PECA DELE? SE SIM, ENTAO GUARDA A SELECAO!!
    if((turno==NORMAL_ROSA and tabuleiro[linha][coluna]==NORMAL_ROSA)
        or (turno==NORMAL_AMARELO and tabuleiro[linha][coluna]==NORMAL_AMARELO)):
        pecaSelecionada = [linha, coluna, tabuleiro[linha][coluna]]
        #denotar que a peca foi selecionada para o jogador
        tabuleiro[linha][coluna] = SELECIONADA
        reDesenharTabuleiro(tabuleiro)
```

```
pygame.quit()
```