



**UNIVERSIDADE FEDERAL DO PARANÁ**  
**CURSO DE MATEMÁTICA**

Nikolle Gambaro Coutinho

Stephany Correa Godoy

**JOGO DE DAMAS EM PYTHON**

**CURITIBA**

**2018**

**NIKOLLE GAMBARO COUTINHO (20185665)**

**STEPHANY CORREA GODOY (20185669)**

## **JOGO DE DAMAS EM PYTHON**

Relatório apresentado à disciplina  
Fundamentos da Programação de  
Computadores do Curso de Gradu-  
ação em Matemática da Universida-  
de Federal Paraná.

Orientador: Prof. Jackson Antônio  
do Prado Lima

**Curitiba, novembro de 2018**

## SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	<b>3</b>
<b>2 OBJETIVOS.....</b>	<b>4</b>
2.1 OBJETIVO GERAL.....	4
<b>3 DESENVOLVIMENTO.....</b>	<b>5</b>
3.1 O TRABALHO.....	5
<b>4 CONCLUSÃO.....</b>	<b>11</b>
<b>REFERÊNCIA .....</b>	<b>12</b>

## **1 INTRODUÇÃO**

Este relatório apresenta o desenvolvimento e a explicação do código do jogo de damas elaborado em linguagem python. Utilizando o que aprendemos ao longo do semestre somados à pesquisa sobre bibliotecas dentro do sistema, sempre buscando a maneira mais fácil e coesa de montar nosso código.

## **2 OBJETIVOS**

### **2.1 OBJETIVO GERAL**

O objetivo é obter um jogo de damas, programado na linguagem python, onde os usuários possam competir entre si para que no final haja um campeão (rosa ou girassol).

### 3 DESENVOLVIMENTO

#### 3.1 O TRABALHO

Começamos a montagem do jogo importando as bibliotecas pygame, sys e math. Optamos por utilizar a pygame por sua fácil manipulação e eficiência tanto na parte lógica quanto gráfica do código. Também, já definimos algumas variáveis globais (que serão utilizadas sempre), as cores que vamos utilizar no jogo, largura, altura e raio das peças e o tamanho da janela.

```
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]
```

Em seguida inicializamos o pygame e abrimos uma janela com o nome “Jogo de Damas”. A variável *fim* foi criada para fechar o programa posteriormente.

Dentro da biblioteca pygame, utilizando seus recursos, definimos a fonte que aparece na peça quando esta vira rainha.

```
#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
```

Definimos algumas funções que serão úteis no decorrer do programa. A função *cor\_quadrado* define as cores dos quadrados que serão desenhados posteriormente a partir de comandos. A função *getColunaClick* e a *getLinhaClick* pega as coordenadas (x,y) da tela que o usuário clicou e transforma no lugar correspondente da matriz do jogo.

```
# 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
```

A função *reDesenharTabuleiro* é responsável por interpretar as mudanças feitas pelo usuário e imprimi-las no tabuleiro, utilizando outras funções que logo serão definidas, e recursos do pygame. A função *mudarPecaLugar* serve para mudar as posições das peças, alterando a matriz inicial que representa o tabuleiro do jogo. Já a função *trocaTurno* troca o turno do jogo, ou seja, varia as jogadas entre os jogadores.

```
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
```

A seguinte função *desenharTabuleiro* é responsável por organizar e definir a posição dos quadrados dentro do tabuleiro do jogo. As coordenadas criadas vão colocar as linhas e colunas da matriz relacionadas aos pixels da janela.

```
#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])
```

A função *novoTabuleiro* apenas cria o tabuleiro inicial que será modificado diversas vezes. Foi criada no formato de matriz (uma lista de listas), onde 0 são os campos vagos, 1 são as peças rosa e 2 as peças amarelas.

```
# 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
```

Em *desenharPecas* utilizamos comandos que primeiramente diferenciam espaços vagos dos que tem peças. Depois organiza essas peças de maneira central dentro do tabuleiro já definido. Para desenhar as peças foi usada a função *draw.circle* do pygame e as cores delas são as que foram definidas como variáveis globais no início do programa. Também criamos o princípio para separar as peças convencionais das rainhas (Q), que dentro do jogo apresentariam comportamentos diferentes.

```
#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)
```

A função *isFimDeJogo* basicamente imprime na tela o vencedor, ou seja, quando o numero de peças amarelas for zero (0), rosa será campeão e vice-versa.



```

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

```

Para abrir a parte inicial do jogo chamamos as funções que serão usadas e abrimos comandos para execução.

```

#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()

```

Abrimos um *while* que vai introduzir a parte logica do jogo. Onde iniciamos comandos para clicar com o mouse nos botões.

```

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)

```

Começamos finalmente a parte logica, que somente irá funcionar se o quadrado selecionado não for nulo. Dentro do *if* chamamos funções auxiliaadoras e criamos diversas restrições que fazem parte de algumas das regras do jogo.

```
#----- 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
```

Aqui nesse *elif* restringimos os movimentos somente para diagonais, sempre avançando. E fizemos os comandos para “comer” uma peça do adversário.

```
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)
```

Por fim temos a parte final que encerra o jogo.

```

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()

```

## **4 CONCLUSÃO**

Este trabalho foi um pouco estressante, mas com ele tivemos a oportunidade de aplicar tudo aquilo que aprendemos neste semestre sobre programação em python, além de outros recursos que tivemos que aprender a utilizar para poder incrementar no nosso projeto.

## REFERÊNCIAS

Sites acessados:

<<https://www.youtube.com/watch?v=Z-Q5DHPXfdg&t=3s>>

<<https://www.youtube.com/watch?v=FEtw4p00Rnw&t=346s>>

<<https://www.youtube.com/watch?v=abDnqoDjDSg>>

<<https://www.youtube.com/watch?v=6H8UEXMzS10&t=418s>>

<[https://github.com/LukeHxH/mp\\_damas](https://github.com/LukeHxH/mp_damas)>

<<https://www.pygame.org/docs/>>