

```

1 import pygame
2 import os
3
4 # --- LÓGICA DO JOGO (Funções do jogo de terminal) ---
5
6 def criar_tabuleiro():
7     """Cria e retorna o tabuleiro inicial do jogo de damas em uma matriz 8x8."""
8     tabuleiro = []
9     for i in range(8):
10         linha = []
11         for j in range(8):
12             if (i + j) % 2 != 0:
13                 if i < 3:
14                     linha.append("p") # Peças pretas
15                 elif i > 4:
16                     linha.append("b") # Peças brancas
17                 else:
18                     linha.append(" ") # Casa vazia e jogável
19             else:
20                 linha.append("-") # Casa não jogável
21         tabuleiro.append(linha)
22     return tabuleiro
23
24 def eh_movimento_valido(tabuleiro, jogador, l_origem, c_origem, l_destino, c_destino):
25     """Verifica se um movimento é válido (versão simplificada para GUI)."""
26     peca = tabuleiro[l_origem][c_origem]
27     if peca.lower() != jogador or tabuleiro[l_destino][c_destino] != " ":
28         return False
29
30     # Movimento da peça comum
31     if peca in ('b', 'p'):
32         direcao = -1 if jogador == 'b' else 1
33         # Movimento simples
34         if l_destino == l_origem + direcao and abs(c_destino - c_origem) == 1:
35             return True
36         # Movimento de captura
37         if abs(l_destino - l_origem) == 2 and abs(c_destino - c_origem) == 2:
38             l_captura, c_captura = (l_origem + l_destino) // 2, (c_origem + c_destino) //
2
39             peca_capturada = tabuleiro[l_captura][c_captura]
40             if peca_capturada.lower() not in (" ", jogador):
41                 return True
42     return False
43
44 def mover_peca(tabuleiro, l_origem, c_origem, l_destino, c_destino):
45     """Move a peça e promove a Dama se necessário."""
46     peca = tabuleiro[l_origem][c_origem]
47     tabuleiro[l_destino][c_destino] = peca
48     tabuleiro[l_origem][c_origem] = " "
49
50     # Se foi uma captura, remove a peça capturada
51     if abs(l_destino - l_origem) == 2:
52         l_captura, c_captura = (l_origem + l_destino) // 2, (c_origem + c_destino) // 2

```

```
53     tabuleiro[l_captura][c_captura] = " "
54
55     # Promover a Dama
56     if peca == 'b' and l_destino == 0:
57         tabuleiro[l_destino][c_destino] = 'B'
58     elif peca == 'p' and l_destino == 7:
59         tabuleiro[l_destino][c_destino] = 'P'
60
61 # --- PARTE GRÁFICA COM PYGAME ---
62
63 # Inicialização do Pygame
64 pygame.init()
65
66 # Configurações da tela e do tabuleiro
67 LARGURA, ALTURA = 640, 640
68 LINHAS, COLUNAS = 8, 8
69 TAMANHO_CASA = LARGURA // COLUNAS
70 tela = pygame.display.set_mode((LARGURA, ALTURA))
71 pygame.display.set_caption("Jogo de Damas")
72
73 # Cores
74 MARROM_ESCURO = (139, 69, 19)
75 MARROM_CLARO = (222, 184, 135)
76 PRETO = (0, 0, 0)
77 BRANCO = (255, 255, 255)
78 VERDE_DESTAQUE = (0, 255, 0) # Cor para destacar peça selecionada
79
80 def desenhar_tabuleiro(tela):
81     """Desenha as casas do tabuleiro."""
82     tela.fill(MARROM_CLARO)
83     for linha in range(LINHAS):
84         for coluna in range(linha % 2, COLUNAS, 2):
85             pygame.draw.rect(tela, MARROM_ESCURO, (linha*TAMANHO_CASA,
86                 coluna*TAMANHO_CASA, TAMANHO_CASA, TAMANHO_CASA))
87
88 def desenhar_pecas(tela, tabuleiro):
89     """Desenha as peças com base na matriz do tabuleiro."""
90     raio = TAMANHO_CASA // 2 - 10
91     for linha in range(LINHAS):
92         for coluna in range(COLUNAS):
93             peca = tabuleiro[linha][coluna]
94             if peca != ' ' and peca != '-':
95                 centro_x = coluna * TAMANHO_CASA + TAMANHO_CASA // 2
96                 centro_y = linha * TAMANHO_CASA + TAMANHO_CASA // 2
97
98                 cor_peca = BRANCO if peca.lower() == 'b' else PRETO
99                 pygame.draw.circle(tela, cor_peca, (centro_x, centro_y), raio)
100
101                 # Adiciona uma "coroa" para a Dama
102                 if peca in ('B', 'P'):
103                     pygame.draw.circle(tela, VERDE_DESTAQUE, (centro_x, centro_y), raio //
104 2)
```

```
105     rodando = True
106     clock = pygame.time.Clock()
107     tabuleiro = criar_tabuleiro() # Agora esta função existe!
108     jogador_atual = 'b'
109     peca_selecionada = None
110
111     while rodando:
112         clock.tick(60) # Limita o jogo a 60 frames por segundo
113
114         for event in pygame.event.get():
115             if event.type == pygame.QUIT:
116                 rodando = False
117
118             if event.type == pygame.MOUSEBUTTONDOWN:
119                 pos_x, pos_y = pygame.mouse.get_pos()
120                 linha, coluna = pos_y // TAMANHO_CASA, pos_x // TAMANHO_CASA
121
122                 if peca_selecionada:
123                     l_origem, c_origem = peca_selecionada
124                     if eh_movimento_valido(tabuleiro, jogador_atual, l_origem, c_origem,
linha, coluna):
125                         mover_peca(tabuleiro, l_origem, c_origem, linha, coluna)
126                         jogador_atual = 'p' if jogador_atual == 'b' else 'b'
127                         peca_selecionada = None # Desseleciona após a tentativa de movimento
128                     else:
129                         if tabuleiro[linha][coluna].lower() == jogador_atual:
130                             peca_selecionada = (linha, coluna)
131
132                 # Desenho
133                 desenhar_tabuleiro(tela)
134                 desenhar_pecas(tela, tabuleiro)
135
136                 # Destaca a peça selecionada
137                 if peca_selecionada:
138                     l, c = peca_selecionada
139                     pygame.draw.rect(tela, VERDE_DESTAQUE, (c*TAMANHO_CASA, l*TAMANHO_CASA,
TAMANHO_CASA, TAMANHO_CASA), 4)
140
141                 pygame.display.flip()
142
143                 pygame.quit()
144
145 if __name__ == "__main__":
146     main()
```