

CAIXA PRETA

Função rodarAnimação(screen)

Entrada	Resultado
Botão esquerdo do mouse selecionado	Pular animação de abertura
Sair do jogo	Pula
Qualquer outro botão selecionado	Ver toda animação de abertura

Enquanto nenhum botão for selecionado, exceto quando o botão esquerdo do mouse ou se o jogador tiver fechado o jogo, a animação de abertura será exibida na tela

Função gerarTabuleiro(matriz, largura, altura, dimensoes)

Entrada	Resultado
matriz gerada largura = qualquer valor altura = qualquer valor dimensoes = 8 ou 16	Matriz com largura e altura ajustadas por causa da dimensão selecionada
matriz gerada largura = qualquer valor altura = qualquer valor dimensoes = qualquer outro valor que não seja 8 ou 16	Não existe
matriz não gerada largura = qualquer valor altura = qualquer valor dimensoes = 8 ou 16	Não existe

Sempre irá receber essas dimensões, pois as mesmas são preestabelecidas na escolha da dificuldade do jogo fácil (dimensão: 8) ou difícil (dimensão: 16)

Função tamanhoFonte(dimensoes)

classes:

dimensoes	Resultado
x = 8	40
x = 16	20
Qualquer outro valor	Não existe

dados:

dimensoes	Resultado
8	40
16	20
15	Não existe
-4	Não existe

Teste da função tamanhoFonte, a qual possui como parâmetros um número nomeado como dimensoes. Essa função deve retornar o tamanho da fonte do texto indicando o número de rastreamento e se há bomba ou não na tela do jogo.

Função gerarMatriz(num)

classes:

num	Resultado
$x < 0$	Não existe
$x = 0$	Lista vazia
$x > 0$	Matriz quadrada x

dados:

num	Resultado
-5	Não existe
-1	Não existe
0	Lista vazia
1	Matriz quadrado 1
5	Matriz quadrada 5

Teste da função gerarMatriz, a qual possui como parâmetro um número nomeado como num. Essa função deve retornar uma matriz quadrada com a dimensão do num dado como input.

Função rodarMenu(screen)

classes:

Entrada	Resultado
Botão jogar e fácil selecionados	8
Botão jogar e difícil selecionados	16
Botão sair selecionado	0

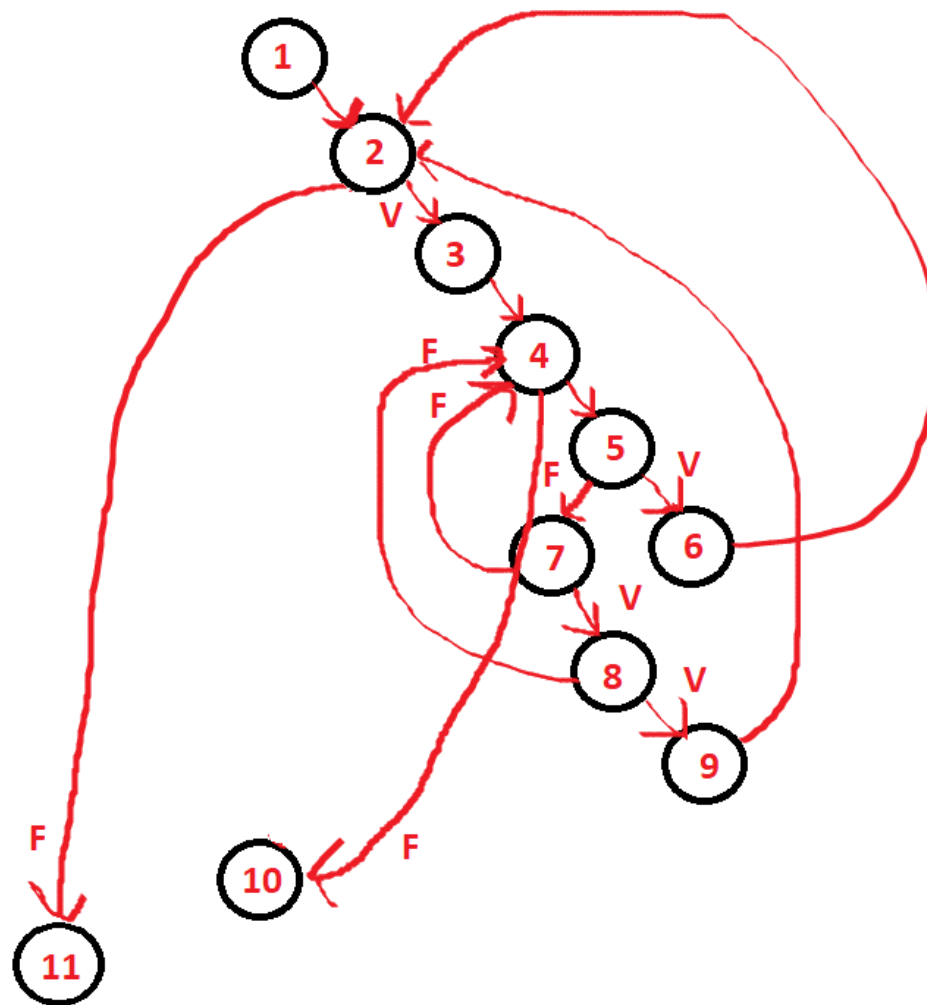
dados:

Entrada	Resultado
Botão jogar e fácil selecionados	8
Botão jogar e difícil selecionados	16
Botão sair selecionado	0

Teste da função rodarMenu(screen), a qual possui como parâmetro a tela onde deve imprimir suas operações, chamado de screen. O retorno da função não depende do parâmetro. A função deve retornar o número que indicará as dimensões das matrizes que será utilizada no jogo ou 0, indicando que é para fechar o jogo.

CAIXA BRANCA

```
def rodarAnimacaoInicial(screen):  
    tela = screen  
  
    AZUL = (65, 105, 225)  
    1 todas_as_sprites = pygame.sprite.Group()  
    bomberman = Bomberman()  
    todas_as_sprites.add(bomberman)  
    img = pygame.image.load('Sprites/CampoMinado.png')  
  
    relógio = pygame.time.Clock()  
    animacao = True  
    2 while animacao:  
        3 relógio.tick(30)  
        4 tela.fill(AZUL)  
        tela.blit(texto_anim, (110, 350))  
        4 for event in pygame.event.get():  
            5 if event.type == QUIT:  
                6 | pygame.quit()  
                6 | exit()  
            7 if event.type == pygame.MOUSEBUTTONDOWN:  
                8 if pygame.mouse.get_pressed() == (1, 0, 0):  
                    9 | animacao = False  
    10 todas_as_sprites.draw(tela)  
    todas_as_sprites.update(tela)  
    11 pygame.display.flip()
```

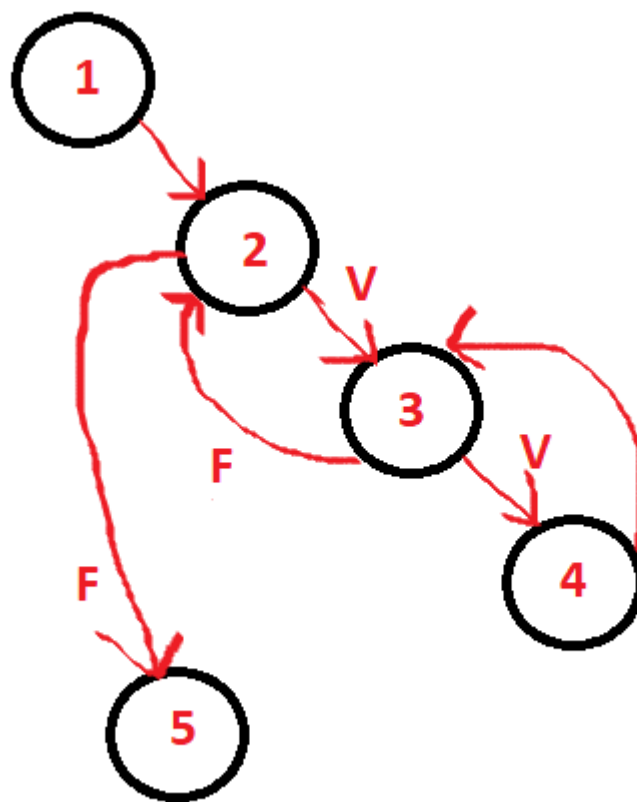


1. {1,2,3,4,5,6,2,11}
2. {1,2,3,4,5,7,8,9,2,11}
3. {1,2,3,4,10}
4. {1,2,3,4,5,7,4,10}
5. {1,2,3,4,5,7,8,4,10}

```

def gerarTabuleiro(matriz, larg, alt, num):
    x_inicial = 0
    y_inicial = 100
    newLarg = (larg) / num
    newAlt = (alt - y_inicial) / num
    for i in range(0, num):
        for j in range(0, num):
            matriz[i][j].rect = [x_inicial + newLarg * j, y_inicial + newAlt * i]

```

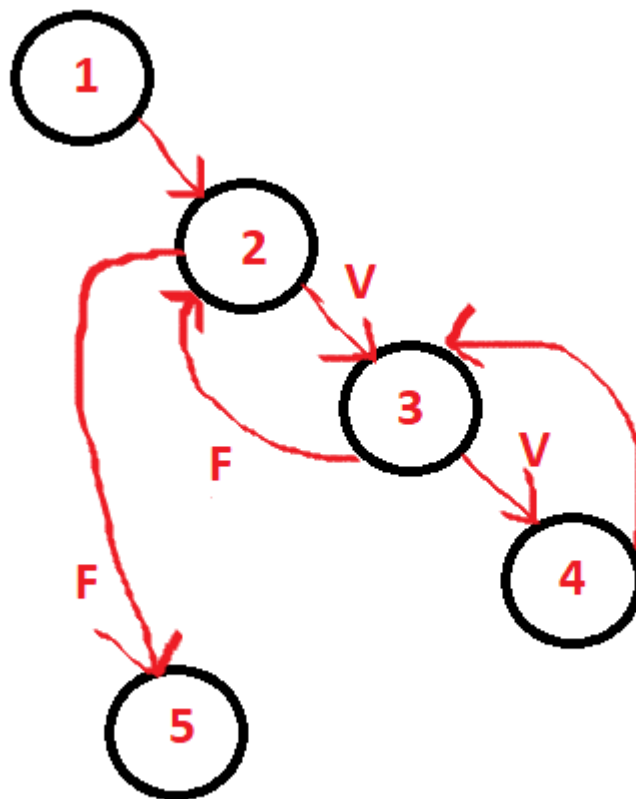


1. {1,2,3,4,3,2,...,5}
2. {1,2,3,2,...,5}
3. {1,2,5}

```

def printMatriz(matriz, screen, larg, alt, num):
    Lista = matriz
    y_inicial = 100
    newLarg = (larg) / num
    newAlt = (alt - y_inicial) / num
    for lista in Lista:
        for cell in lista:
            printarBlock(screen, cell.rect, newLarg, newAlt, cell.cond)

```

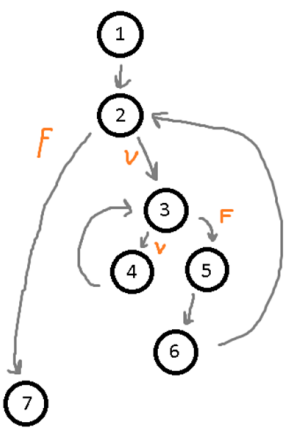


- 1. {1,2,3,4,3,2,...,5}
- 2. {1,2,3,2,...,5}
- 3. {1,2,5}

A função printMatriz exibe na tela o campo minado, a partir da matriz gerada, largura, altura, dimensão e da função printarBlock(imprime cada célula da matriz)

Teste da função gerarObjetos(matriz, tela, newLarg, newAlt)

```
def gerarObjetos(matriz, tela, newLarg, newAlt):
1  objetos = []
2  for i in matriz:
3      for j in i:
4          xy = j.rect
5          obj_aux = pygame.draw.rect(tela, (0, 0, 0), (xy[0], xy[1], newLarg, newAlt))
6          objetos.append(obj_aux)
7  return objetos
```

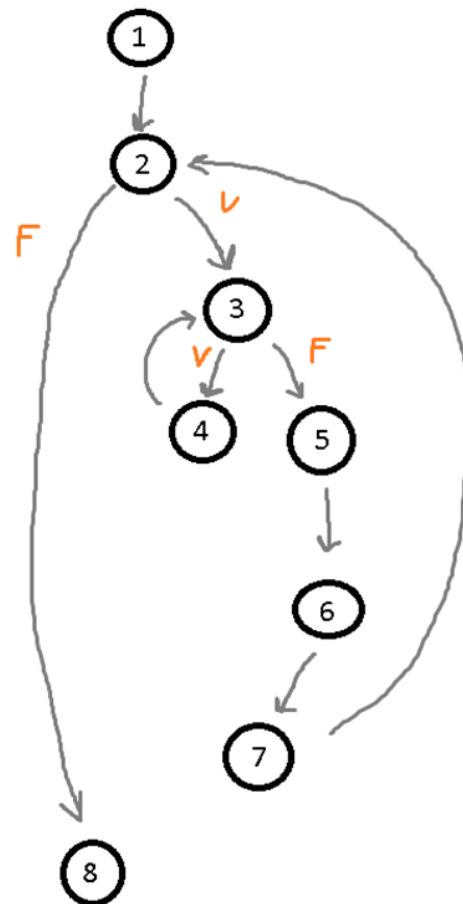


1. {1, 2, 7}	
matriz	Vazia
2. {1, 2, 3, 4, 5, 6, 2, ..., 7}	
matriz	[1, 2, 3, 4]
3.{1, 2, 3, 5, 6, 2, ..., 10}	
matriz	Impossível

Não existe valor possível, pois não é permitido passar uma lista com shape (2, 2) com elementos vazios.

Teste da função gerarMatriz(num)

```
def gerarMatriz(num):
1  matriz = []
  aux = []
2  for i in range(0, num):
    3  for j in range(0, num):
      4  objaux = obj([i, j], 0, 0, 0)
      5  aux.append(objaux)
    6  matriz.append(aux)
    aux = []
8  return matriz
```



1. {1, 2, 10}	
num	0
2. {1, 2, 3, 4, 5, 6, 7, 2, ..., 10}	
num	2
3. {1, 2, 3, 5, 6, 7, 2, ..., 10}	
num	Impossível

Não existe valor possível, pois a função gera uma matriz quadrada. Por consequência, não é possível que entre no loop do nó 2 e não entre no 3.

A fim de tornar o jogo mais eficiente, realizamos algumas mudanças estruturais. O arquivo onde se localiza o main esta muito extenso, logo criamos um arquivo com a função jogar, no qual executa a mesma função que o main, sendo que agora essa função jogar é chamada no main atualizado. Desse modo, o main atualizado ficaria mais enxuto. Além disso, criamos uma função que retorna o tamanho da fonte que deverá ser usada para imprimir os números na tela e por fim uma função que retorna uma lista de objetos que serão utilizados para colisão durante o jogo.