Prof. Me. Alexandre Henrick

TI II - 7º período Sistemas de Informação

Atividade Busca - 3 pontos

1 - Resolva o exercício sobre busca A* nos últimos slides da aula sobre **Busca**.

---

2 - O código abaixo é a implementação simples de um jogo do robô aspirador de pó. Nesse jogo o robô se movimenta de maneira autônoma. No entanto, seus movimentos são aleatórios e isso pode fazer com que a limpeza de todos os quadrados seja demorada. Implemente um algoritmo de busca para ajudar o robô a ser mais "inteligente" e realizar a busca de maneira mais rápida. Você pode implementar uma busca em largura ou uma busca A*.

Observações:

- A atividade poderá ser desenvolvida em grupos com 2 ou 3 pessoas. Todos devem enviar a atividade no portal
- O código foi escrito em Python 3 com a biblioteca pygame. Portanto é necessário instalar a mesma no seu ambiente python:
    - ```
      pip install pygame
      ```
- Você pode apenas adicionar uma função extra que realiza a busca no código
- Você poderá desenvolver a atividade em outra linguagem, caso tenha conhecimento em outra ferramenta/biblioteca com o mesmo propósito
- O arquivo .py também está em anexo na atividade
- Envie sua resolução em anexo no portal

```
import pygame
import random

# Configurações iniciais
pygame.init()
largura, altura = 500, 500
tamanho_celula = largura // 5
tela = pygame.display.set_mode((largura, altura))
pygame.display.set_caption("Aspirador de Pó Automático")
fonte = pygame.font.SysFont("Arial", 30)

# Cores
PRETO = (0, 0, 0)
BRANCO = (255, 255, 255)
```
```

```
VERDE = (0, 255, 0)
VERMELHO = (255, 0, 0)
AZUL = (0, 0, 255)

# Inicialização do ambiente
ambiente = [[random.choice([0, 1]) for _ in range(5)] for _ in range(5)]  #
0 = limpo, 1 = sujo
posicao_robo = [random.randint(0, 4), random.randint(0, 4)]  # Posição
inicial aleatória
score = 0

# Função para desenhar o ambiente
def desenhar_ambiente():
    tela.fill(BRANCO)
    for i in range(5):
        for j in range(5):
            cor = VERDE if ambiente[i][j] == 0 else VERMELHO
            pygame.draw.rect(tela, cor, (j * tamanho_celula, i *
tamanho_celula, tamanho_celula, tamanho_celula))
            pygame.draw.rect(tela, PRETO, (j * tamanho_celula, i *
tamanho_celula, tamanho_celula, tamanho_celula), 1)
    # Desenhar o robô
    pygame.draw.circle(tela, AZUL, (posicao_robo[1] * tamanho_celula +
tamanho_celula // 2, posicao_robo[0] * tamanho_celula + tamanho_celula //
2), tamanho_celula // 3)
    # Desenhar o score
    texto = fonte.render(f"Score: {score}", True, PRETO)
    tela.blit(texto, (10, 10))

# Função para mover o robô
def mover_robo():
    global posicao_robo, score
    direcoes = [(0, 1), (1, 0), (0, -1), (-1, 0)]  # Direções possíveis:
direita, baixo, esquerda, cima
    direcao = random.choice(direcoes)
    nova_linha = posicao_robo[0] + direcao[0]
    nova_coluna = posicao_robo[1] + direcao[1]
    # Verificar se a nova posição está dentro do ambiente
    if 0 <= nova_linha < 5 and 0 <= nova_coluna < 5:
        posicao_robo = [nova_linha, nova_coluna]
        # Verificar se há sujeira na nova posição
        if ambiente[nova_linha][nova_coluna] == 1:
            ambiente[nova_linha][nova_coluna] = 0
            score += 1

# Loop principal do jogo
clock = pygame.time.Clock()
rodando = True
while rodando:
    for evento in pygame.event.get():
        if evento.type == pygame.QUIT:
            rodando = False

    mover_robo()
```

```
    desenhar_ambiente()
    pygame.display.flip()
    clock.tick(2)  # Controla a velocidade do robô (2 movimentos por
segundo)

pygame.quit()
```