

Atividade: "Robô Explorador Inteligente"

Objetivo Geral:

Esta atividade tem como objetivo solidificar a compreensão sobre as diferentes tipologias de agentes inteligentes. Em grupos, vocês irão projetar, implementar e avaliar o comportamento de um robô simulado que navega em um ambiente de grid $n \times n$. O desafio aumentará progressivamente, exigindo que o agente se torne mais "inteligente" a cada etapa, evoluindo de reações simples para a tomada de decisões baseada em objetivos e utilidade.

Cenário:

Um robô autônomo foi colocado em um ambiente desconhecido, em uma posição inicial aleatória, representado por um grid de $n \times n$ células (iniciaremos com $n=10$). O robô pode se mover para as células adjacentes (Norte, Sul, Leste, Oeste), uma célula por rodada. A tarefa é programar o robô, passando por quatro grandes "atualizações", cada uma correspondendo a um tipo de agente mais avançado.

Progressão da Complexidade:

O projeto será dividido em quatro etapas:

Etapa 1: Agente Reativo Simples

- **Descrição:** nesta fase, o agente não tem memória (estado interno) das posições visitadas. Sua decisão de movimento é baseada apenas na sua percepção atual (sua posição e se há uma parede nos limites do grid). O ambiente é um grid vazio, sem obstáculos.
- **Objetivo do Agente:** o objetivo do agente é explorar o ambiente até ter colidido com as quatro paredes limites (norte, sul, leste e oeste).
- **Métricas de Avaliação:**
 - Deteção Completa do Perímetro:** o agente conseguiu determinar corretamente os limites do grid (sim/não).

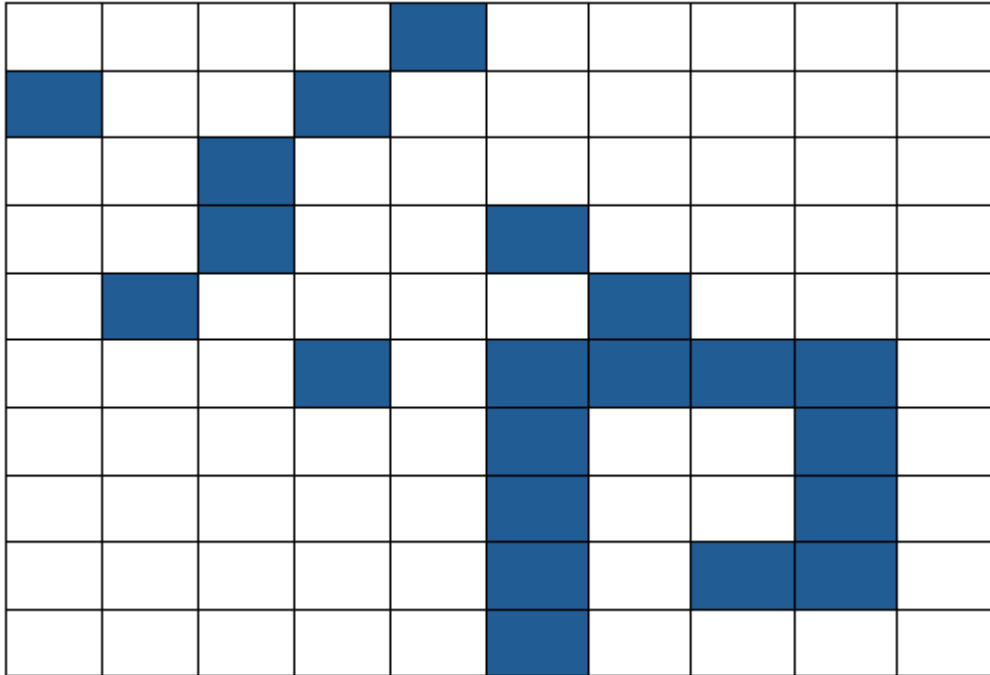
Etapa 2: Agente Reativo Baseado em Modelo

- **Descrição:** o agente agora é atualizado com uma memória (um estado interno), que inicia *zerada*. Ele deve manter um "mapa" de quais células já visitou.

Inicialmente, o ambiente é livre. Depois, introduzimos obstáculos estáticos em posições pré-definidas.

- **Objetivo do Agente:** visitar o maior número possível de células do grid, evitando repetir células já visitadas e contornando obstáculos (quando presentes).

Obstáculos: introduzir obstáculos (exemplo abaixo):



- **Métricas de Avaliação:**

Compleitude da Exploração: qual a porcentagem de células acessíveis que o agente visitou? (Meta: 100%).

Eficiência de Exploração: qual o número de passos redundantes (visitas a uma mesma célula mais de uma vez)? (Meta: o mais próximo de zero possível).

Sucesso no Desvio: o agente consegue explorar todas as áreas acessíveis mesmo na presença de obstáculos complexos? (Sim/Não).

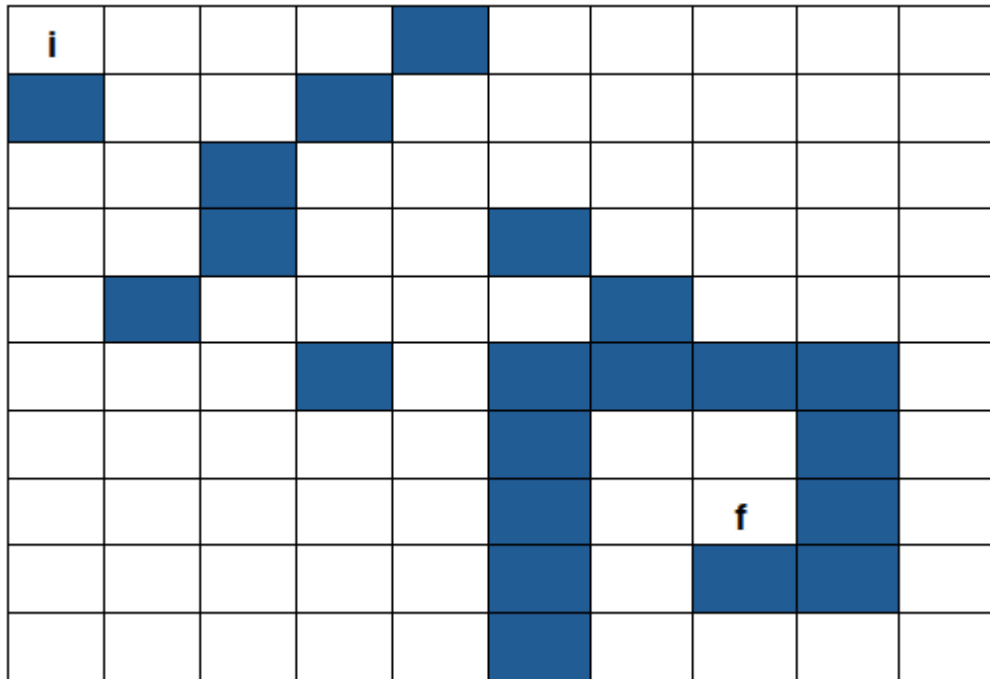
Generalização do agente: como ele se comportaria com obstáculos diferentes?

Etapas 3: Agente Baseado em Objetivos

- **Descrição:** dados uma célula de partida (x_i, y_i) e uma célula de destino (x_f, y_f), o agente deve encontrar um caminho entre elas. O desafio será dividido em duas fases:

Ambiente Livre: inicialmente, o agente deve encontrar o caminho em um grid vazio, com início e fim desconhecidos *a priori*.

Ambiente com Obstáculos: em seguida, o agente deve ser capaz de encontrar o caminho desviando dos obstáculos estáticos definidos na etapa anterior (assumir modelo do ambiente previamente construído, estabelecer início e fim *aleatoriamente*; exemplo abaixo).



- **Objetivo do Agente:** encontrar e percorrer um caminho **válido** do início ao fim.
- **Métricas de Avaliação:**
 - Sucesso na Tarefa:** o agente alcançou a célula de destino? (Sim/Não).
 - Comprimento do Caminho:** quantos passos foram necessários para chegar ao objetivo? (Um caminho mais curto é melhor).

Etapas 4: Agente Baseado em Utilidade

- **Descrição:** agora, não basta apenas chegar ao destino; o agente deve fazer isso da maneira mais eficiente possível, considerando que o ambiente possui "custos" variados. O grid agora representa diferentes tipos de terreno. Cada tipo de célula tem um "custo de movimento" associado para entrar nela.
 - Terrenos e Custos:**
 - Célula 'Normal': Custo 1
 - Célula 'Arenosa': Custo 2
 - Célula 'Rochosa': Custo 3

Varição 1: Ambiente completamente observável

Varição 2: Ambiente parcialmente observável

- **Objetivo do Agente:** dado um ponto de partida e um de destino fixos, encontrar o caminho que **minimiza o custo total** da viagem. Um caminho mais curto (em número de passos) pode não ser o melhor se passar por terrenos de alto custo.
- **Métricas de Avaliação:**
 - **Sucesso na Tarefa:** o agente alcançou a célula de destino? (Sim/Não).
 - **Custo Total do Caminho:** qual foi a soma dos custos de todas as células no caminho percorrido?

1	1	1	1	1	i	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	3	1	1	1	1
1	1	2	2	1	3	3	2	1	1
1	1	2	1	3	3	3	2	1	1
1	1	2	2	3	3	3	2	2	1
1	1	1	2	3	1	2	2	1	1
1	1	1	1	2	3	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	f	1	1	1	1

Entregáveis por Etapa:

Para cada uma das quatro etapas, o grupo deverá entregar:

- **Fluxograma e/ou pseudocódigo:** descrevendo a lógica do agente.
- **Código Python:** um script funcional que simula o agente no ambiente grid.
 - **Visualização (opcional):** um bônus será concedido para grupos que implementarem uma forma simples de visualizar o grid e o comportamento do agente em tempo real ou ao final da execução.
- **Relatório de Análise:** mínimo de 3 parágrafos explicando a lógica implementada para cada agente, o tipo do ambiente e os resultados obtidos com base nas métricas de avaliação, além de comentários da equipe a respeito da implementação.

Regras adicionais:

- 1) Não serão utilizadas técnicas de aprendizado por reforço ou IA conexionista;
- 2) Uso de IA generativa é permitida como recurso complementar. Sugestão: priorizar a construção da lógica de forma autônoma.