

## Tarefa 04: Busca de Custo Uniforme x A\*

### Objetivos de aprendizagem

- compreender o que é solução ótima
- compreender a diferença entre busca cega (custo uniforme) e busca informada (A\*)
- compreender o que é heurística admissível e consistente

### Método

#### Equipe

Até 2 pessoas

#### Objetivo da tarefa

A partir do ambiente Labirinto e do agente construídos nas tarefas anteriores, implemente dois algoritmos: (i) **busca de custo uniforme** e (ii) **A\***. O agente tem por objetivo encontrar um caminho ótimo (de menor custo) que defina uma sequência de ações de uma posição inicial ( $S_0$ ) até uma posição destino = objetivo (posição  $S_g$ ) desviando das paredes. Este plano deve ser construído automaticamente pelos dois algoritmos. As paredes serão colocadas conforme desenho abaixo. Seguem os parâmetros:

- LINHAS=COLUNAS=9
- $S_0$  = posição do agente (8, 0) // (linha, coluna)
- $S_g$  = posição objetivo (2, 8)
- ações(S) é um subconjunto de {N, NE, L, SE, S, SO, O, NO}
- $\text{custo}(S, a, S') = 1$  se  $a \in \{N, L, S, O\}$   
 $= 1,5$  se  $a \in \{NE, SE, SO, NO\}$
- heurísticas para A\*: crie duas heurísticas h1 e h2 admissíveis e consistentes

	0	1	2	3	4	5	6	7	8
0	XXX XXX				XXX XXX XXX XXX				
1	XXX						XXX		
2				XXX XXX XXX				G	
3				XXX XXX XXX		XXX			
4									
5		XXX XXX			XXX		XXX		
6		XXX		XXX XXX		XXX			
7		XXX		XXX		XXX			
8	A XXX XXX								

#### Fazer e entregar

- **Código fonte:** implementação que permita escolher entre uma das seguintes estratégias:
  - custo-uniforme;
  - A\* com h1;

- A\* com  $h_2$ ;
- É **OBRIGATÓRIO** implementar a árvore de busca com alocação dinâmica de nós (ver classe `TreeNode`)
- É **OBRIGATÓRIO** implementar a fronteira ordenada de acordo com a estratégia de busca em uso (A\* pelo  $f(n)$ ; custo-uniforme pelo  $g(n)$ )
- Uma **tabela comparativa** dos três algoritmos contendo os seguintes elementos:
  - **Complexidade temporal**  
**Número de nós gerados:** insira todos os nós na árvore mesmo que não sejam incluídos na fronteira por já terem sido explorados ou porque já estão na fronteira com custo melhor. Ao substituir um nó que está na fronteira por um melhor, não o retire da árvore (apenas retire-o da fronteira).
    - Conte todos os nós não inseridos na fronteira por já terem sido explorados em `ct_ja_explorados`;
    - Conte todos os nós não inseridos na fronteira porque já há um melhor na fronteira em `ct_descartados_front`;
    - Conte todos os nós substituídos na fronteira porque um melhor foi encontrado em `ct_descartados_front`.
  - **Complexidade espacial**  
Considere cada uma das iterações, ao final mostre o valor máximo obtido dentre todas as iterações para as variáveis abaixo:
    - Número de nós armazenados na árvore de busca na iteração
    - `ct_ja_explorados`
    - `ct_descartados_front`
  - A **solução** encontrada
  - O **custo da solução** encontrada

**A cada ciclo, o agente deve:**

- se primeiro ciclo então
  - executar algoritmo de busca com base na formulação do problema
  - armazenar solução
  - imprimir solução
- imprimir o estado do mundo (e não as crenças do agente acerca do mundo!)
- retirar a próxima ação do plano
- imprimi-la
- executar a ação

## Referências

- slides 011a-Busca-Cega.pdf
- slides 012a-Busca-Informada-ou-Heuristica.pdf
- cap. 3 Russel & Norvig 3ed.