

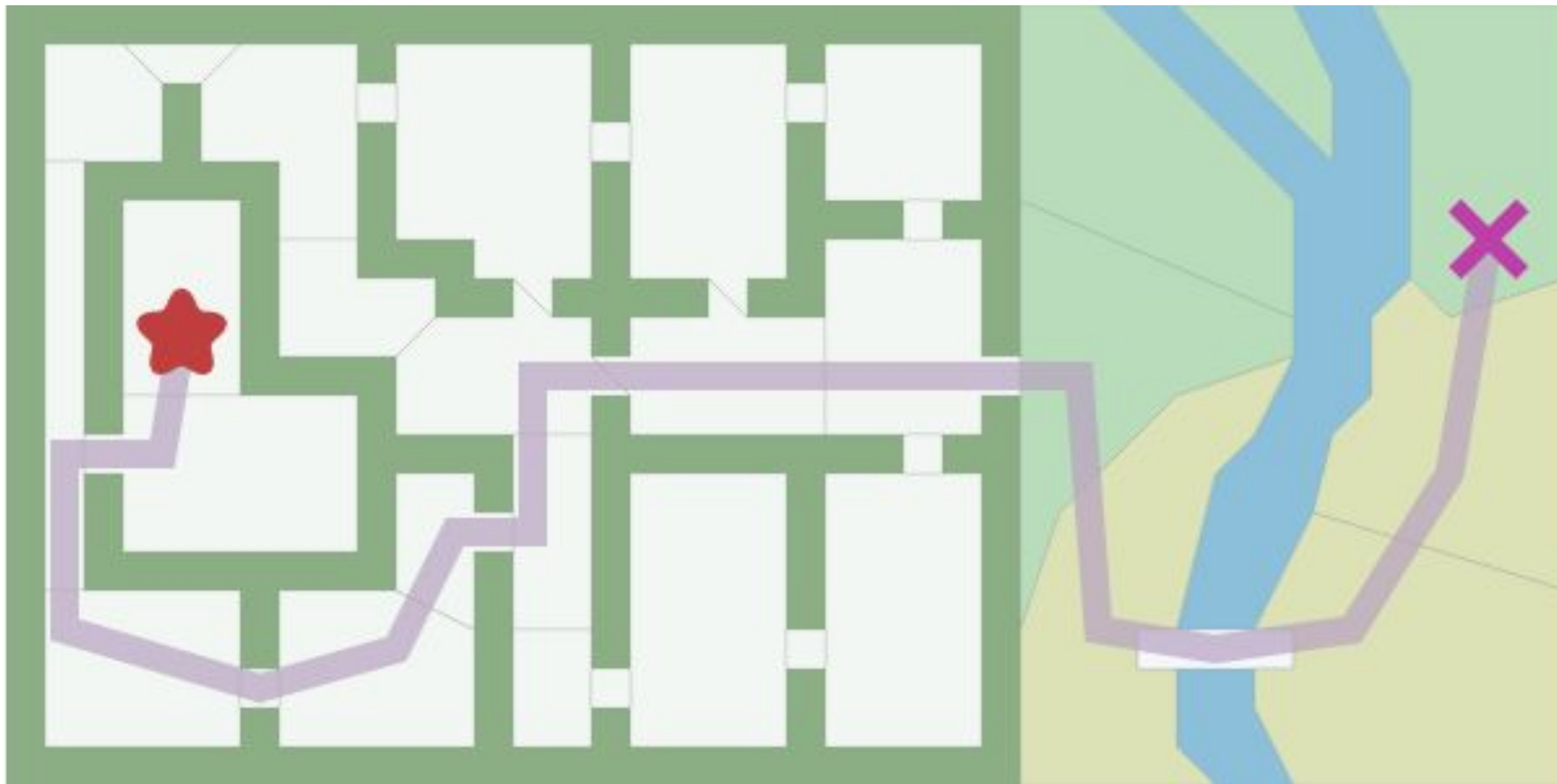
A*

**Roger de Almeida Matos Junior
Guilherme Melo de Medeiros**

Link do github do projeto

Motivação

- Encontrar o caminho de menor custo entre a origem e o destino.



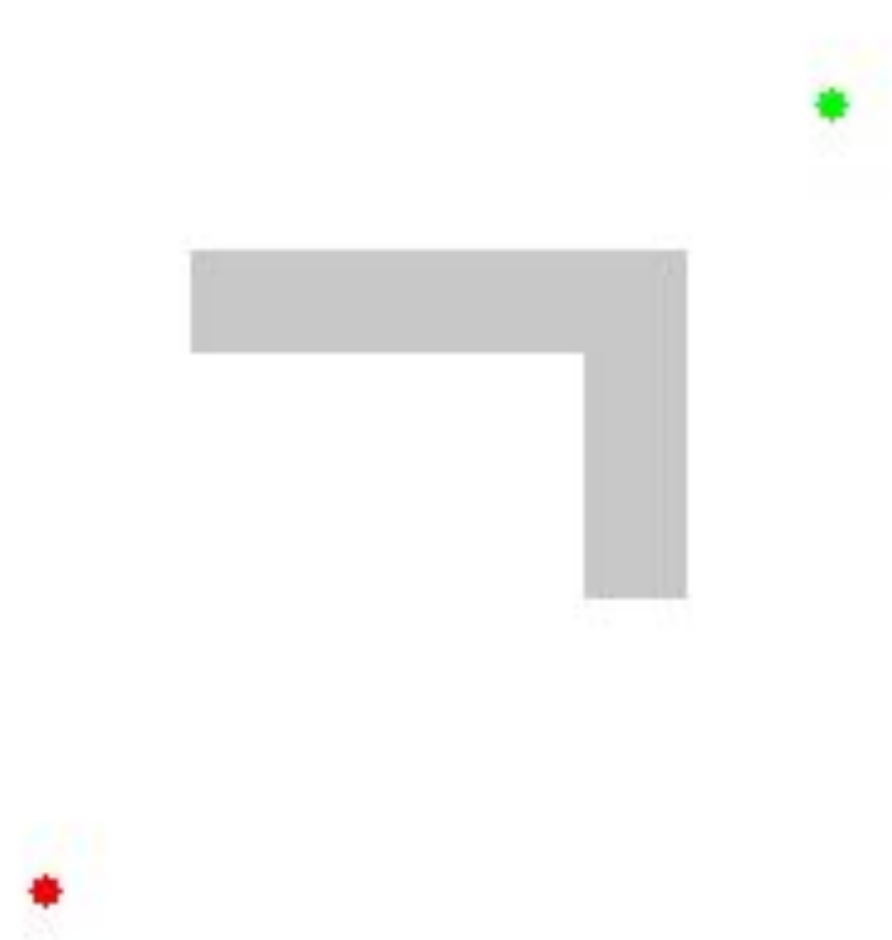
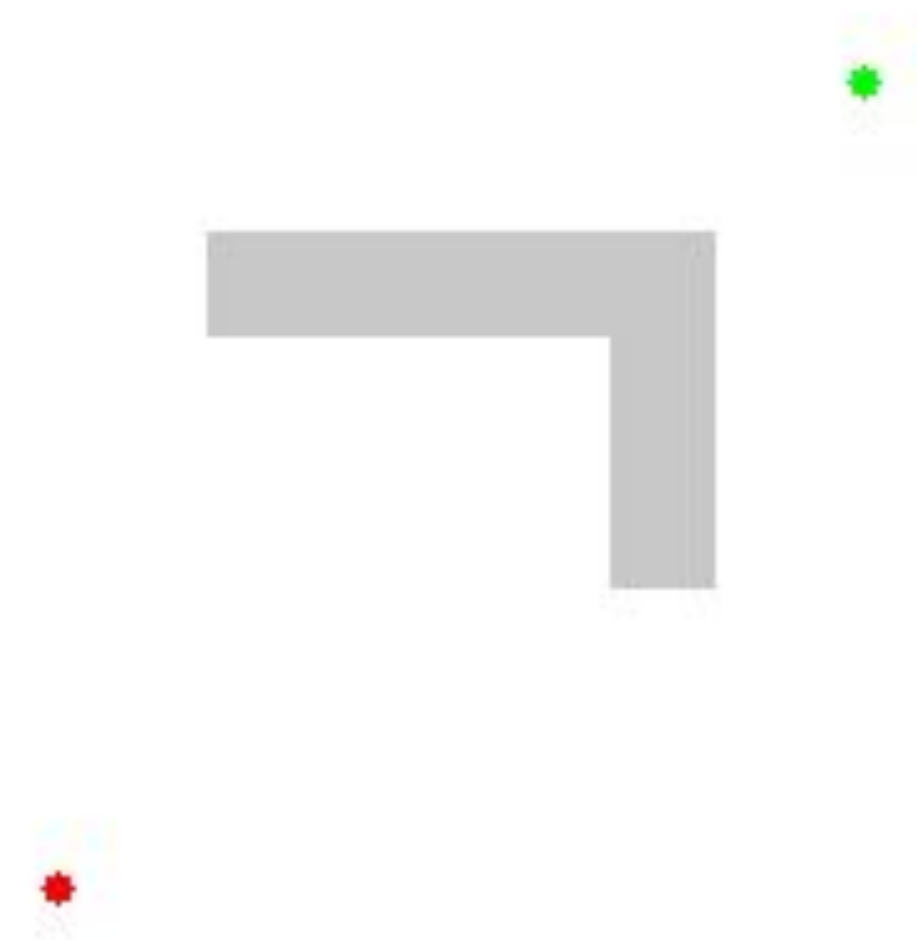
A*

- Algoritmo de busca em grafo que utiliza a busca em largura como base do algoritmo.
- Escolhe entre todos os caminhos possíveis de um determinado ponto aquele que aparenta chegar mais rapidamente em seu destino.

Definições

- A cada interação, o A* seleciona o caminho que minimiza a função $f(n) = g(n) + h(n)$, onde $g(n)$ determina o custo do ponto atual ao próximo ponto e $h(n)$ é a heurística do algoritmo, que dita o quão perto esse caminho está do ponto de destino.
- Se $h(n) = 0$, o algoritmo se comportará exatamente como o algoritmo de Dijkstra.

A* X DIJKSTRA



Pseudo-código

1. Inicializa a lista aberta
2. Inicializa a lista fechada
Coloca o nó do começo na lista aberta
(você pode deixar o f como zero)
3. enquanto a lista aberta não estiver vazia
 - a) encontre o nó com o menor f na lista aberta e chame-o de "q"
 - b) retire o q da lista aberta
 - c) gere 8 sucessores de q e coloque o q como pai

d) para cada sucessor

i) se o sucessor for o alvo, pare a busca

sucessor.g = q.g + distância entre q

sucessor.h = distância do alvo ao sucessor

(Isso pode ser feito usando várias formas como
heurística Manhattan, Diagonal e Euclidiana)

sucessor.f = sucessor.g + sucessor.h

ii) se um nó com a mesma posição do sucessor está na
lista aberta e esse tem um f menor que ele, pule o
sucessor

iii) se um nó com a mesma posição que o sucessor está
na lista fechada e tem um f menor que o sucessor,
pule o sucessor, caso contrário, adicione o nó na lista
aberta

final do loop d)

e) coloque o q na lista fechada

final do loop 3

A*

- Utilizado em jogos e em aplicativos de mapeamento.
- Aplicação:

<http://bgrins.github.io/javascript-astar/demo/>

