

PCS 3848

PRIMEIRO EXERCÍCIO PROGRAMA

Dupla

Nome	Nº USP	E-mail
Fernanda Monteiro Lopes	9793158	fernanda.monteiro.lopes@usp.br
Tiago DAgostino	3759732	tiago.dagostino@usp.br

Link para o Código

<https://github.com/djhamo/pacman>

Descrição do Código

A função *aStarSearch* vai receber como argumentos:

- *problem* : descreve o problema (espaço de estados, estado inicial, função de custo, função sucessor e o teste de objetivo)
- *heuristic* : heurística que será utilizada na busca

E retornará uma lista contendo as ações realizadas para chegar ao estado objetivo, sendo as ações possíveis: North, South, East, West e Stop

A lista *fronteira* abrigará os nós de fronteira ordenados por prioridade, de forma que o primeiro a ser explorado seja sempre o de menor custo. Os nós nesta lista conterão suas respectivas informações de estado, ações que levaram até aquele estado e custo.

A lista *nohExplorado* conterá todos os nós que já foram explorados, cada nó contendo as suas respectivas informações de estado e custo (nesta lista não é necessário conter as ações realizadas pois ela só será usada para verificar se o nó já foi explorado ou não, e se o custo é diferente no caso de nós já explorados)

É colocado na *fronteira* o *nohInicial*, contendo o estado inicial, a lista de ações vazia e o custo igual a zero.

Enquanto houver algum nó na *fronteira*, serão realizadas as seguintes operações:

- O nó de menor custo da fronteira é obtido e separado em: estado atual, ações feitas até o estado atual, e custo atual. Respectivamente *curEstado*, *todasAcoes* e *curCusto*;
- Este nó é adicionado na lista de Nós Explorados (*nohExplorado*);
- Se o estado atual for o estado objetivo será retornada a lista *todasAcoes*.
- Se o estado atual não for o estado objetivo, o nó atual é expandido, obtendo seus sucessores na lista *sucessores*, onde um sucessor é composto pelo estado sucessor, pela ação que leva a este estado vindo do estado atual e pelo custo desse passo;
 - Para cada sucessor da lista:
 - A ação desse passo é anexada a lista de todos os passos que levaram a este estado
 - O custo total para chegar até este estado é calculado,
 - Com essas informações, um novo nó é criado.
 - Cada novo nó será comparado com os nós da lista de explorados;
 - Se o novo nó já foi explorado anteriormente, ele só será substituído na lista *fronteira* se o custo deste novo nó for menor que o custo que já constava para ele na fronteira
 - Se o novo nó não foi explorado ainda, ele será adicionado na lista *fronteira*
- Caso a fronteira ainda contenha nós, o processo se repete novamente para o próximo nó de menor custo, até que a lista da *fronteira* esteja vazia, e então será retornada a lista de ações realizadas para se chegar ao objetivo com o menor custo, dada a heurística utilizada

Descrição das Rotinas de Heurística Testadas

Os testes foram executados em um computador com processador I7 7500U, com 16Gb de memória RAM rodando Ubuntu 18 dentro de uma máquina virtual, sobre o Windows 10.

● Mapas

Mapa	Custo (Expansão ótima de nós)
bigMaze	210
mediumMaze	68
tinyMaze	8
testMaze	7

*A expansão ótima de nós é a quantidade de nós se refere ao número mínimo de expansões para se chegar ao destino

- myHeuristic (Manhattan)

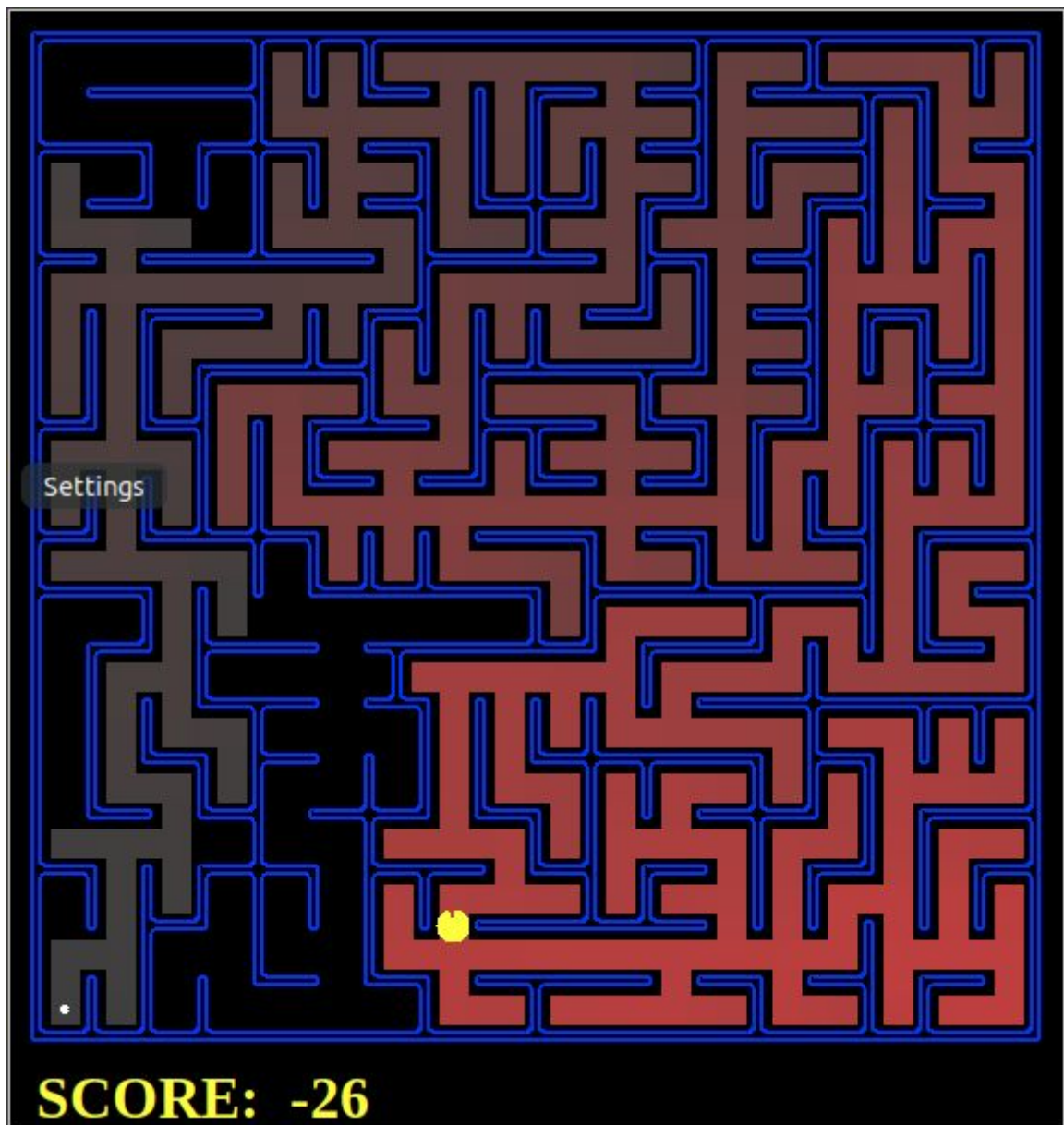
Usa como heurística a distância Manhattan $|Destino - Origem|$

```
$ python pacman.py -l **MAZE** -z .5 -p SearchAgent -a fn=astar,heuristic=myHeuristic
```

****MAZE**** - substituir pelo nome do labirinto desejado (bigMaze, mediumMaze, tinyMaze e teseMaze)

Resultados:

BigMaze



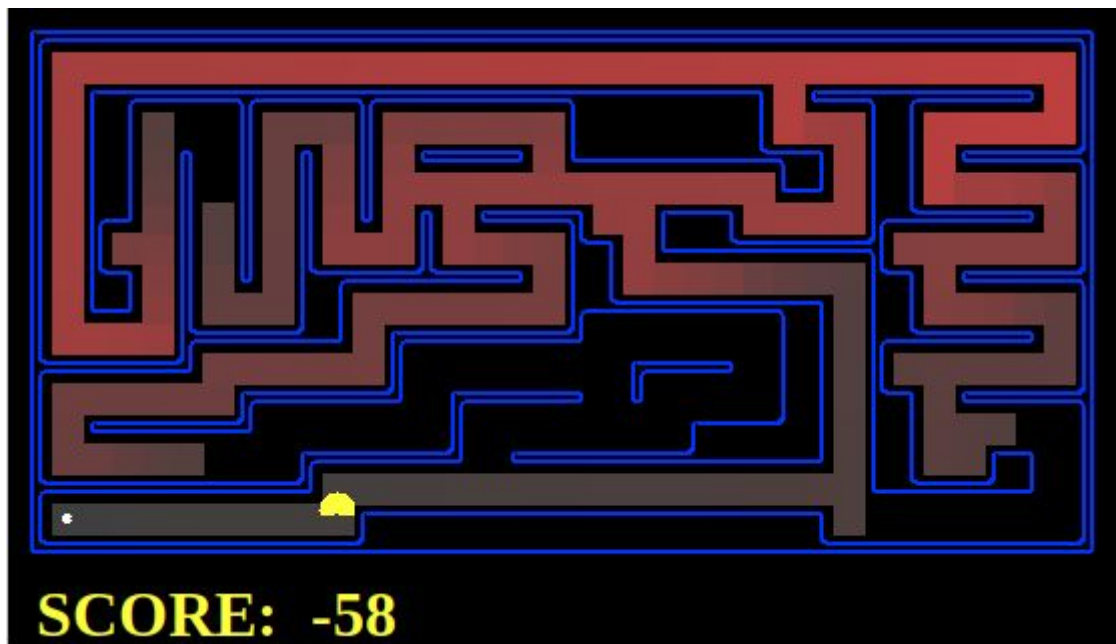
```

aluno@pmr3309:~/Documents/pacman/search$ python pacman.py -l bigMaze -z .5 -p SearchAgent -a fn=astar,heuristic=myHeuristic
[SearchAgent] using function astar and heuristic myHeuristic
[SearchAgent] using problem type PositionSearchProblem
Path found with total cost of 210 in 0.6 seconds
Search nodes expanded: 549
Pacman emerges victorious! Score: 300

```

Tempo	0,6 segundos
Nós expandidos	549
Score	300

mediumMaze



```

aluno@pmr3309:~/Documents/pacman/search$ python pacman.py -l mediumMaze -z .5 -p SearchAgent -a fn=astar,heuristic=myHeuristic
[SearchAgent] using function astar and heuristic myHeuristic
[SearchAgent] using problem type PositionSearchProblem
Path found with total cost of 68 in 0.1 seconds
Search nodes expanded: 222
Pacman emerges victorious! Score: 442

```

Tempo	0,1 segundos
Nós expandidos	222
Score	442

tinyMaze

```
aLuno@pmr3309:~/Documents/pacman/search$ python pacman.py -l tinyMaze -z .5 -p SearchAgent -a fn=astar,heuristic=myHeuristic
[SearchAgent] using function astar and heuristic myHeuristic
[SearchAgent] using problem type PositionSearchProblem
Path found with total cost of 8 in 0.0 seconds
Search nodes expanded: 14
Pacman emerges victorious! Score: 502
```

Tempo	0,0 segundos
Nós expandidos	14
Score	502

testMaze

```
aLuno@pmr3309:~/Documents/pacman/search$ python pacman.py -l testMaze -z .5 -p SearchAgent -a fn=astar,heuristic=myHeuristic
[SearchAgent] using function astar and heuristic myHeuristic
[SearchAgent] using problem type PositionSearchProblem
Path found with total cost of 7 in 0.0 seconds
Search nodes expanded: 7
Pacman emerges victorious! Score: 503
```

Tempo	0,0 segundos
Nós expandidos	7
Score	503

- myHeuristic2 (Euclidiana)

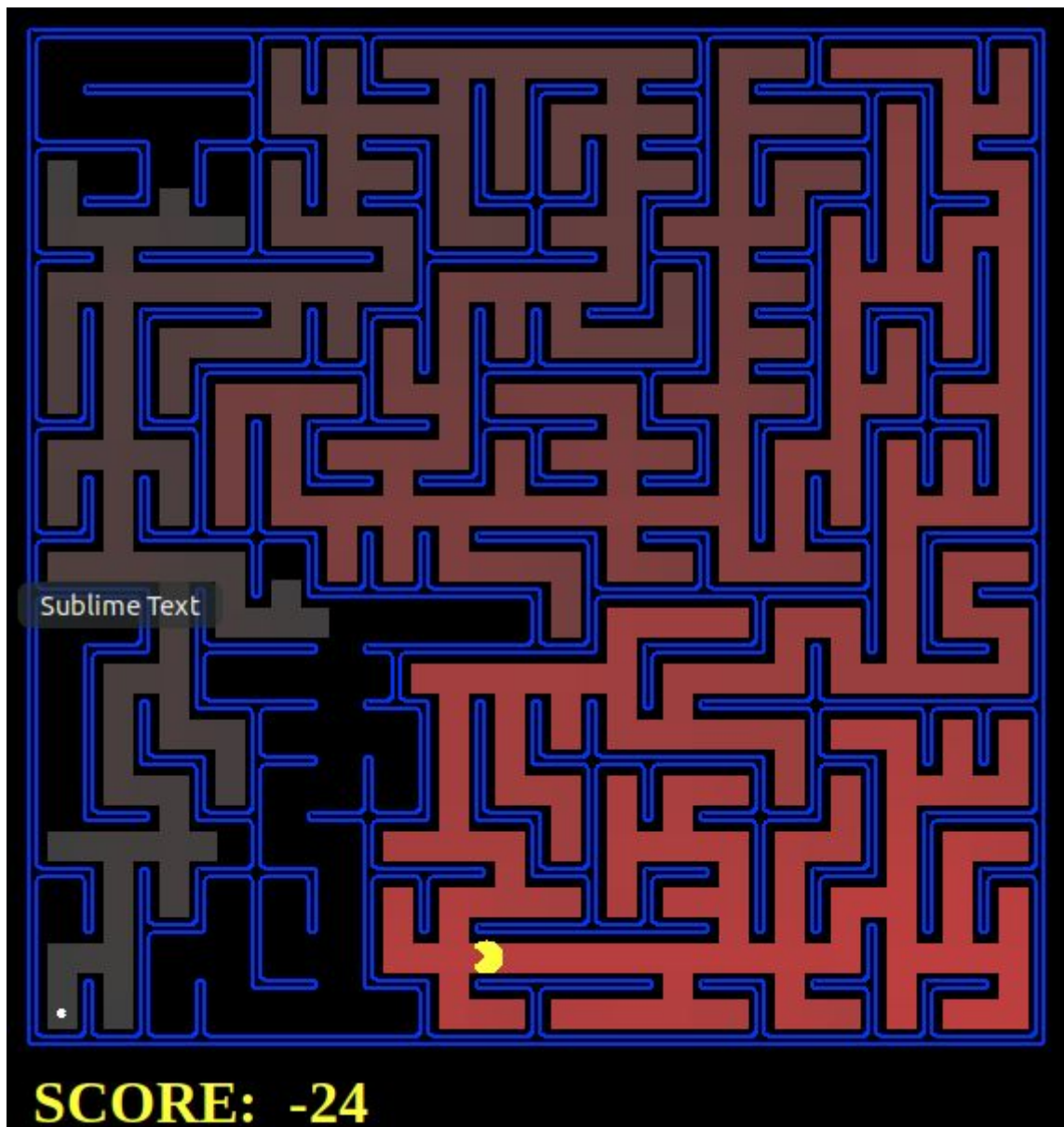
Usa como heurística a distância Euclidiana $\sqrt{(Destino - Origem)^2}$

```
$ python pacman.py -l **MAZE** -z .5 -p SearchAgent -a fn=astar,heuristic=myHeuristic2
```

****MAZE**** - substituir pelo nome do labirinto desejado (bigMaze, mediumMaze, tinyMaze e testeMaze)

Resultados:

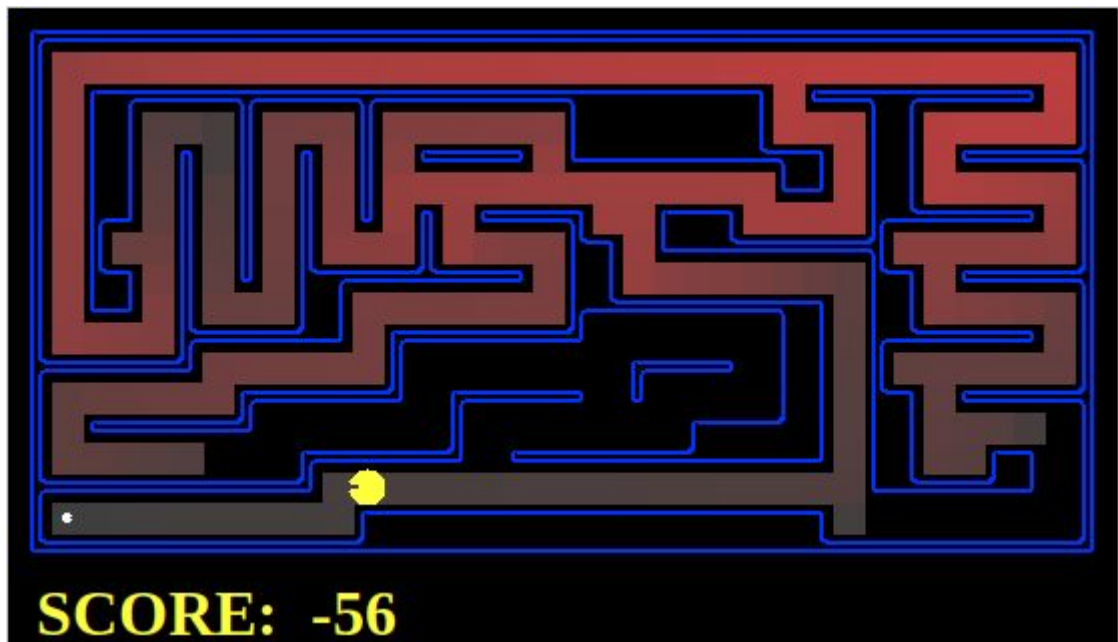
BigMaze



```
aluno@pmr3309:~/Documents/pacman/search$ python pacman.py -l bigMaze -z .5 -p SearchAgent -a fn=astar,heuristic=myHeuristic2
[SearchAgent] using function astar and heuristic myHeuristic2
[SearchAgent] using problem type PositionSearchProblem
Path found with total cost of 210 in 0.5 seconds
Search nodes expanded: 557
Pacman emerges victorious! Score: 300
```

Tempo	0,5 segundos
Nós expandidos	557
Score	300

MediumMaze



```
aluno@pmr3309:~/Documents/pacman/search$ python pacman.py -l mediumMaze -z .5 -p SearchAgent -a fn=astar,heuristic=myHeuristic2
[SearchAgent] using function astar and heuristic myHeuristic2
[SearchAgent] using problem type PositionSearchProblem
Path found with total cost of 68 in 0.2 seconds
Search nodes expanded: 226
Pacman emerges victorious! Score: 442
```

Tempo	0,2 segundos
Nós expandidos	226
Score	442

tinyMaze

```
aluno@pmr3309:~/Documents/pacman/search$ python pacman.py -l tinyMaze -z .5 -p SearchAgent -a fn=astar,heuristic=myHeuristic2
[SearchAgent] using function astar and heuristic myHeuristic2
[SearchAgent] using problem type PositionSearchProblem
Path found with total cost of 8 in 0.0 seconds
Search nodes expanded: 13
Pacman emerges victorious! Score: 502
```

Tempo	0,0 segundos
Nós expandidos	13
Score	502

testMaze

```
aluno@pmr3309:~/Documents/pacman/search$ python pacman.py -l testMaze -z .5 -p SearchAgent -a fn=astar,heuristic=myHeuristic2
[SearchAgent] using function astar and heuristic myHeuristic2
[SearchAgent] using problem type PositionSearchProblem
Path found with total cost of 7 in 0.0 seconds
Search nodes expanded: 7
Pacman emerges victorious! Score: 503
```

Tempo	0,0 segundos
Nós expandidos	7
Score	503

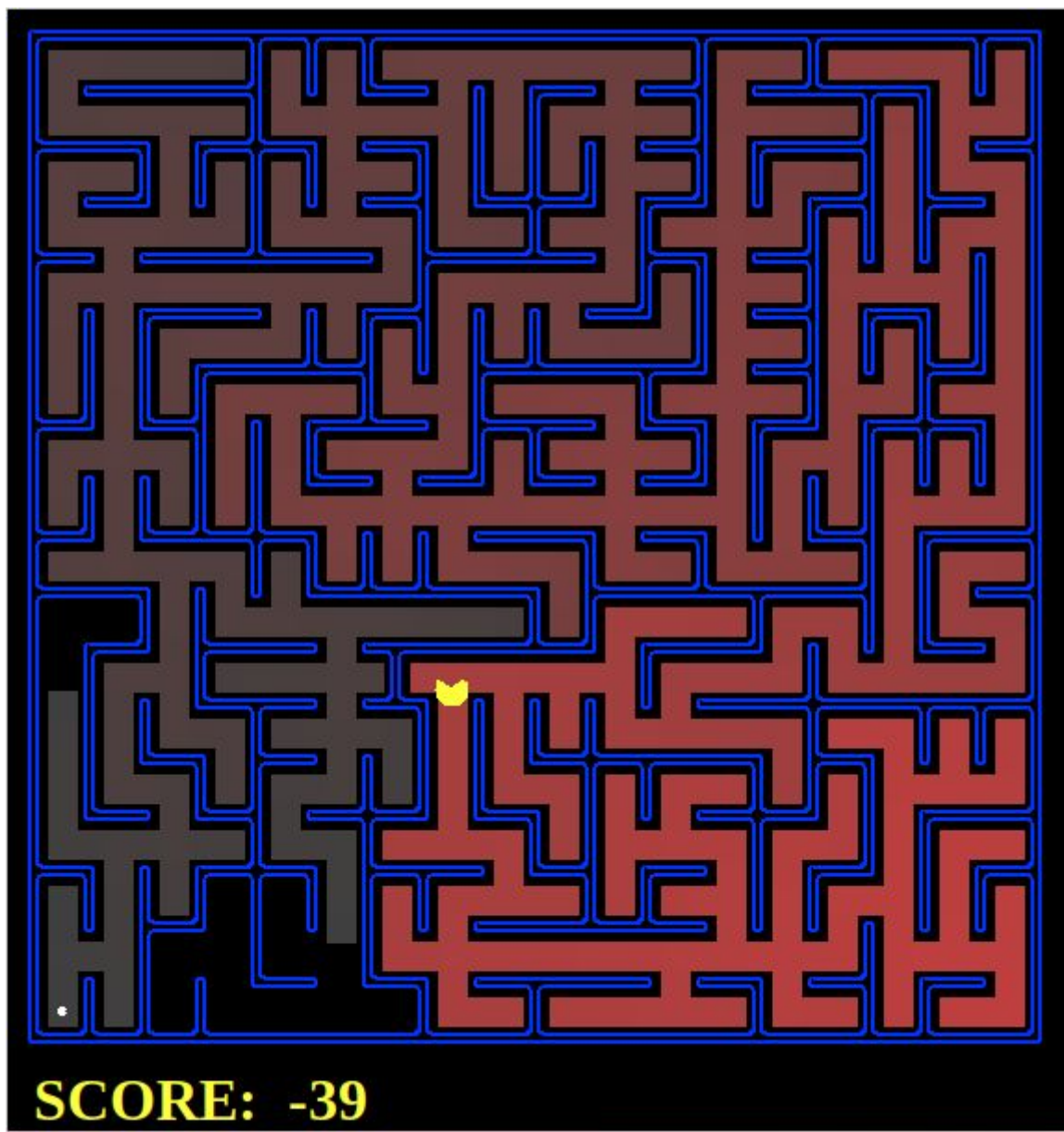
- nullHeuristic (Heurística retorna zero)

```
$ python pacman.py -l **MAZE** -z .5 -p SearchAgent -a fn=astar,heuristic=nullHeuristic
```

****MAZE**** - substituir pelo nome do labirinto desejado (bigMaze, mediumMaze, tinyMaze e teseMaze)

Resultados:

BigMaze



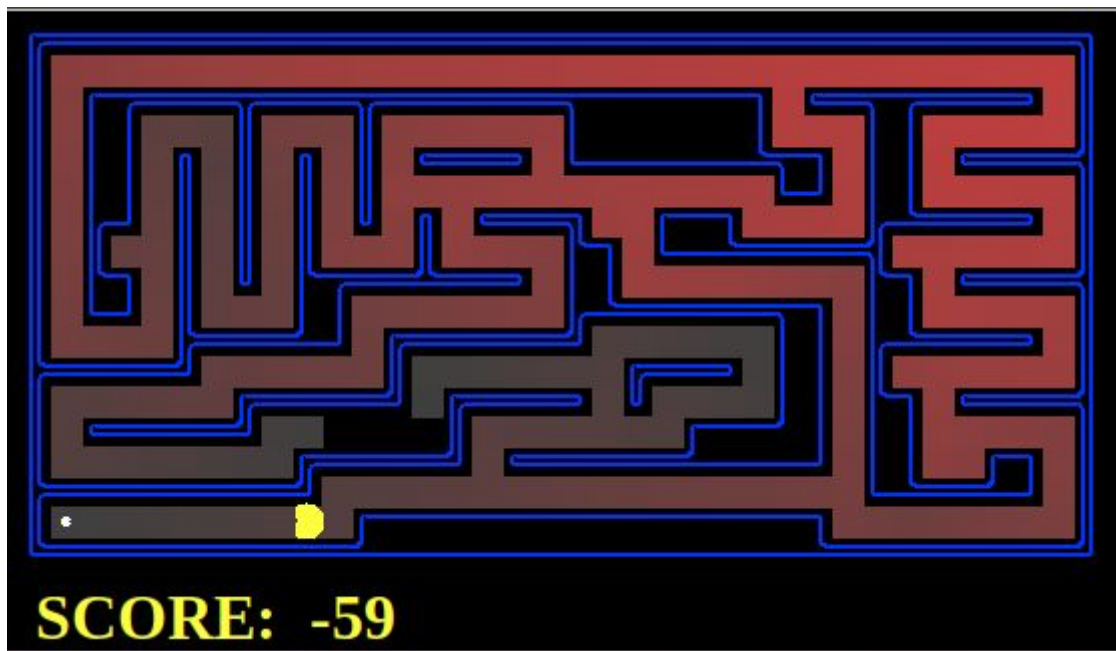
```

aluno@pmr3309:~/Documents/pacman/search$ python pacman.py -l bigMaze -z .5 -p SearchAgent -a fn=astar,heuristic=nullHeuristic
[SearchAgent] using function astar and heuristic nullHeuristic
[SearchAgent] using problem type PositionSearchProblem
Path found with total cost of 210 in 0.6 seconds
Search nodes expanded: 620
Pacman emerges victorious! Score: 300

```

Tempo	0,6 segundos
Nós expandidos	620
Score	300

MediumMaze



```

aluno@pmr3309:~/Documents/pacman/search$ python pacman.py -l mediumMaze -z .5 -p SearchAgent -a fn=astar,heuristic=nullHeuristic
[SearchAgent] using function astar and heuristic nullHeuristic
[SearchAgent] using problem type PositionSearchProblem
Path found with total cost of 68 in 0.1 seconds
Search nodes expanded: 269
Pacman emerges victorious! Score: 442

```

Tempo	0,1 segundos
Nós expandidos	269
Score	442

tinyMaze

```
aluno@pmr3309:~/Documents/pacman/search$ python pacman.py -l tinyMaze -z .5 -p S
SearchAgent -a fn=astar,heuristic=nullHeuristic
[SearchAgent] using function astar and heuristic nullHeuristic
[SearchAgent] using problem type PositionSearchProblem
Path found with total cost of 8 in 0.0 seconds
Search nodes expanded: 15
Pacman emerges victorious! Score: 502
```

Tempo	0,0 segundos
Nós expandidos	15
Score	502

testMaze

```
aluno@pmr3309:~/Documents/pacman/search$ python pacman.py -l testMaze -z .5 -p S
SearchAgent -a fn=astar,heuristic=nullHeuristic
[SearchAgent] using function astar and heuristic nullHeuristic
[SearchAgent] using problem type PositionSearchProblem
Path found with total cost of 7 in 0.0 seconds
Search nodes expanded: 7
Pacman emerges victorious! Score: 503
```

Tempo	0,0 segundos
Nós expandidos	7
Score	503

Conclusão

Observando os resultados do bigMaze e do mediumMaze, nota-se que a busca Manhattan expande significativamente menos nós que as buscas com heurística nula e com heurística Euclidiana. Isso se deve ao custo na heurística Manhattan estar mais próximo do custo real. Como o pacman não anda em diagonal, a heurística Euclidiana subestima mais o custo real do problema do que a Manhattan.

No início da busca, o peso da “heurística” para ditar a direção é maior que o peso do “custo”, porém, mais perto do destino, o “custo” tem mais peso que a “heurística”, o que dita as escolhas de caminho. Essas diferenças ficam aparentes ao comparar as imagens de caminhos percorridos (destacados em tons de vermelho), principalmente próximos do destino.

Otimizando a heurística para o bigMaze, e levando em consideração o que foi dito acima, pode-se criar uma solução específica que consegue expandir menos nós, mas com aumento do custo computacional e sem melhoras significativa em outros labirintos.