



UNEQUAL LENGTH MAZES

TIA_G7
VINÍCIUS ABRUNHOSA
BÁRBARA NETO
MARCO DINIS SOUSA

FORMULAÇÃO DO PROBLEMA

Estados

São dados por uma matriz $n*m$ que representa a situação atual do jogo - o labirinto.

Estado inicial

Matriz $n*m$ com todas as posições livres exceto $(n-1,0)=\text{X}$ e $(0,m-1)=\text{O}$ e os obstáculos.

Estado objetivo

Matriz $n*m$ com todas as posições ocupadas onde as setas formam um trajeto de $(n-1,0)$ a $(0,m-1)$, desviando -se dos obstáculos.

Cada par (x,y) pode tomar os seguintes valores:

none: Posição livre

#: Posição com obstáculo

O: Posição objetivo

X: Posição atual a desenvolver

seta: Posição onde ocorreu o movimento dado pela seta

Exemplo de um labirinto:

	#	↓	←	←	←	0
→	→	→	→	→	→	↑
→	→	→	→	→	→	↑

$n=6$
 $m=6$

• *Operadores*

1) Subir

2) Descer

3) Esquerda

4) Direita

CONDIÇÕES PRÉVIAS	<ul style="list-style-type: none"> • $x > 0$ • $(x-1, y) = \text{None}$ OU • $(x-1, y) = 0 \wedge \text{Count}(\text{None}) = 0$ 	<ul style="list-style-type: none"> • $x < n-1$ • $(x+1, y) = \text{None}$ 	<ul style="list-style-type: none"> • $y > 0$ • $(x, y-1) = \text{None}$ 	<ul style="list-style-type: none"> • $y < m-1$ • $(x, y+1) = \text{None}$ OU • $(x, y+1) = 0 \wedge \text{Count}(\text{None}) = 0$
	CONDIÇÃO PRÉVIA GERAL			Length (current move) ≠ Length (last move)
EFEITOS	$x = x - 1$ $(x, y) = \text{X}$	$x = x + 1$ $(x, y) = \text{X}$	$y = y - 1$ $(x, y) = \text{X}$	$y = y + 1$ $(x, y) = \text{X}$
CUSTO	1	1	1	1

NOTAS: A função *Count(None)* conta o número de entradas da matriz não visitadas.
A função *Length(move)* conta o número de casas que se moveu na direção move.



Estratégia para desenvolver o trabalho

Classe maze

Algoritmos desinformados

Heurísticas

Algoritmos informados

Interface gráfica para ver a resolução da AI

Interface gráfica para o jogador

Pedido de dicas à AI pelo jogador

Testes para apurar resultados experimentais

Classe “Maze”

ATRIBUTOS

Array (numpy) - o labirinto
Move history - labirintos anteriores
Célula a desenvolver
Lista de tuplos dos obstáculos
Shape(numpy) - n° linhas e colunas
Target cell
Last e Current move

FUNÇÕES

innit, eq, hash e str
Gerar obstáculos através do input
Copiar uma instância da classe
Movimentos segundo os operadores
children - posições válidas seguintes
is_solved - verifica se o jogo está terminado
print_sequence - imprime o move_history

Algoritmos

BFS
DFS
Limited DFS
Iterative Deepening
Greedy Search
A* Search
Weighted A* Search

Heurísticas

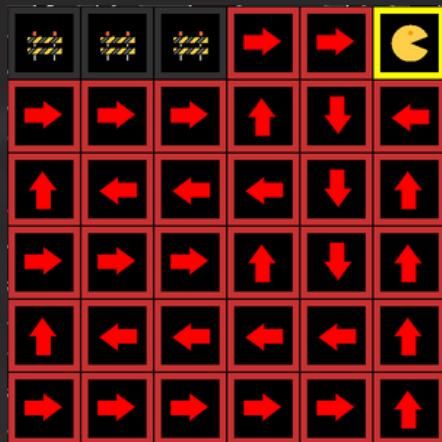
Distância Euclideana
Distância de Manhattan

Interface gráfica

AI & JOGADOR



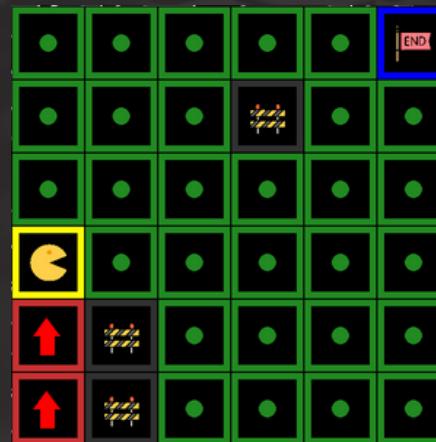
Escolher se se quer jogar ou se quer ver um jogo a ser resolvido pelos algoritmos.



Ver um labirinto a ser resolvido passo a passo.



Escolher a dificuldade do labirinto que se quer resolver.



Resolver um labirinto.



Podendo pedir que a AI diga qual o próximo passo!

RESULTADOS EXPERIMENTAIS

ALGORITMOS DESINFORMADOS

Dificuldade do labirinto: Média
Profundidade máxima: 50

Algoritmo	Nós visitados	Tempo	Passos até solução
DFS	168	0.184776	35
BFS	8925	7.549649	35
Iterative Deepning	1712	1.284116	35
Limited DFS	168	0.145601	35

Nota: Corremos todos os programas mais do que uma vez e concluímos que estes resultados são uma boa média dos anteriores.

ALGORITMOS INFORMADOS & HEURÍSTICAS

Dificuldade do labirinto: Média
Weighted do A*: W = 2

Algoritmo	Nós visitados	Tempo	Passos até solução
Greedy euclideana	8308	7.281116	35
Greedy Manhattan	7300	6.226764	35
A* euclideana	1539	1.308535	35
A* Manhattan	1539	1.308812	35
Weigthed A* euclideana	1539	1.316826	35
Weigthed A* Manhattan	1539	1.308102	35

RESULTADOS EXPERIMENTAIS

LABIRINTOS FÁCEIS

Algoritmo	Nós visitados	Tempo	Passos até solução
Greedy	1954	1.402854	32
A*	2811	2.036752	32
Weighted A*	2811	2.056238	32

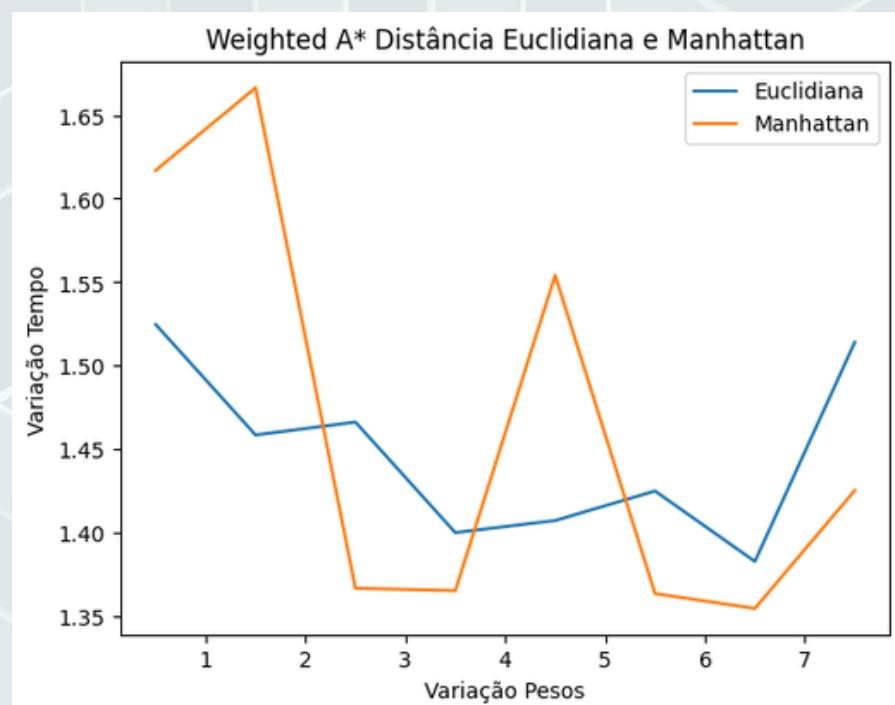
LABIRINTOS DIFÍCEIS

Algoritmo	Nós visitados	Tempo	Passos até solução
Greedy	102092	151.4111261	44
A*	89976	141.126828	44
Weighted A*	89976	141.202241	44

A heurística (**distância de Manhattan**) e o seu peso (**1.5**) foi escolhida através deste teste:

VARIAÇÃO DE PESO DO A*

Dificuldade do labirinto: Média



CONCLUSÕES

- Os **algoritmos os informados** são melhores que os desinformados independentemente da heurística usada.
- Para este problema o **melhor algoritmo desinformado** a ser usado é o **DFS** e o **limited DFS** que seria ideal caso houvesse labirintos impossíveis.
- O **melhor algoritmo informado** a ser usado é o **weighted A***.
- O **melhor peso para o weighted A*** é 1,5 quando combinado com a distância de Manhattan e 7,5 com a distância euclideana.
- A **melhor heurística** entre as duas testadas é a distância de Manhattan. O que seria de esperar pois a esta distância coincide com os movimentos usados no jogo, já a euclideana que “funciona em todas as direções” não.
- O **tempo e memoria** usada aumenta rapidamente quando aumentamos o **grau de dificuldade** do puzzle. Para labirintos maiores torna - se inviável usar algoritmos não informados.



Bibliografia

Notebooks disponibilizados pelos professores

Chat-GPT

Documentação do pygame

<https://erich-friedman.github.io/puzzle/unequal/>

T1A_G7

VINÍCIUS ABRUNHOSA

BÁRBARA NETO

MARCO DINIS SOUSA