Algoritmo A* no ambiente Agent0

Beatriz Silva (2019108992@uac.pt), Pedro Sousa (2019101451@uac.pt), Salif Faustino (20172005@uac.pt)

Resumo

Este trabalho foi realizado no âmbito da disciplina de Inteligência Artificial e teve como objetivo a interpretação e implementação do algoritmo A* no Agente0. O software consiste num ambiente onde o agente se consegue movimentar. Com base nas experiências efetuadas com o algoritmo escolhido, foi possível interpretar e perceber os diversos cenários a que o agente foi submetido.

Introdução

Na cadeira de Inteligência Artificial, do 3º Ano da Licenciatura em Informática, foi-nos proposto a interpretação e implementação do algoritmo A* no Agente0.
A versão utilizada neste projeto foi o Agent0_minotauro, que permite explorar a interação entre um agente e um ambiente. Este software consiste num ambiente com um tabuleiro onde o agente se pode movimentar. De modo a movimentar-se, o agente explora o tabuleiro e desloca-se em frente ou muda de direção nas respetivas casas. O agente pode utilizar diversos algoritmos para chegar ao seu objetivo e é possível observar as casas com diversas cores. A versão final do programa pode ser encontrada no github, através do link seguinte: https://github.com/SalifNTC/Projeto_G10.git .

Descrição do Algoritmo

O algoritmo A* é um algoritmo de pesquisa do tipo **best-first**, isto é, um tipo de algoritmo de pesquisa informada em árvore ou grafo em que os nós a expandir são os com menor custo com base numa função de avaliação denominada de **f(n)**. A pesquisa diz-se informada pois esta função de avaliação recorre a conhecimento específico ao problema, ou heurísticas. No caso do algoritmo **A***, **f(n)** é a soma do custo para chegar ao nó em questão (função denominada por **g(n)**) e uma estimativa do custo para chegar desse nó até ao objetivo, função conhecida como heurística ou **h(n)**.
O algoritmo A* pode ser visto como uma expansão mais eficiente do algoritmo *Uniform Cost Search*, uma vez que neste último, a função de avaliação é apenas **g(n)**. Ambos são ótimos e completos. Pode ser também contrastado com o algoritmo *Greedy Best-First* Search, um algoritmo informado que usa apenas a função h(n) como função de avaliação, mas não é nem ótimo, nem completo.
O algoritmo escolhido tem como objetivo uma etapa base, a qual consiste em implementar o algoritmo com as seguintes características:

1) Mostrar o algoritmo a executar as ações após ter estabelecido o percurso com e sem obstáculos;
2) Enquanto o agente procura/pensa, é apresentada, em sequência, as diferentes hipóteses encontradas até atingir o objetivo. Esta funcionalidade pode ser ligada ou desligada.

Implementação do Algoritmo A*

Tendo em conta que o algoritmo A* considera sempre o caminho que tiver menor custo, foi optado por criar uma função para ordenar os nós fronteiras de modo a garantir que o próximo nó a ser expandido seja o que corresponder a esse menor custo.

```
def pop(self):
    self.queue_data.sort(key=self.funcao_ordenacao) #ordenar antes de remover o Nó
    return self.queue_data.pop(0)
```

Figura 1- Função "pop" da Queue

```
def distancia(self,x1,y1):
    """metodo calcular a distancia"""
    x2, y2=self.getGoalPosition()
    return  abs(x1-x2)+ abs(y1-y2)
```

Figura 2 – Função que calcula a distância de um ponto ao objetivo

```
def getNode(self,parent_node,action):
    state = self.step(parent_node.getState(),action)
    g = parent_node.getPathCost() + self.getPatchCost(state)
    h=self.distancia(*state) #Calcular a heuristica
    return Node(state, parent_node, action, g,g+h)
```

Figura 3 – Função que devolve o nó com o f(n) devidamente calculado

Na função pop (figura 1) é eliminado o primeiro elemento da queue depois de ordenada por ordem crescente tendo em conta **f(n).**
Na figura 2 temos a fórmula principal da distância utilizada, a fórmula de Manhattan, que recebe as coordenadas de dois pontos e retorna a distância de Manhattan somando o valor absoluto da diferença dos valores x dos pontos com a diferença do valor absoluto dos valores y dos pontos.
Na figura 3 temos a função que retorna o nó criado depois de feitos os cálculos para **f(n),** os quais incluem calcular o **g(n)** e somá-lo à heurística utilizada, **h(n)**. Assim o nó resultante vem com **f(n)** já calculado.

Experiências Realizadas ou Exemplos de Aplicação

No conjunto de experiências apresentadas consideramos três estados fundamentais:

- Agente no ambiente;
- Agente após expandir o ambiente;
- Agente com o objetivo atingido.

Experiência 1, percurso sem obstáculos:

Nesta experiência (figura 4) o agente encontra-se inserido num mundo sem obstáculos, o que faz com que a expansão do mundo seja feita sem os ter em conta. Com base nessa experiência é possível observar que o agente expande o mundo utilizando o algoritmo A*, isto é, os nós são expandidos com prioridade. Essa expansão só termina quando o objetivo é atingido e não há um novo nó fronteira com menor custo o qual ainda não tenha sido expandido.
Na imagem mais à esquerda, temos a localização do objetivo e do agente, na do meio o resultado da pesquisa, e, finalmente, na imagem mais à direita temos, além do resultado da pesquisa, o caminho escolhido pelo agente para chegar ao objetivo. As cores apresentadas correspondem ao seguinte: vermelho escuro para os nós pesquisados pela função de pesquisa, vermelho claro os nós fronteira no final da pesquisa, e azul claro os nós usados pelo agente para atingir o objetivo.

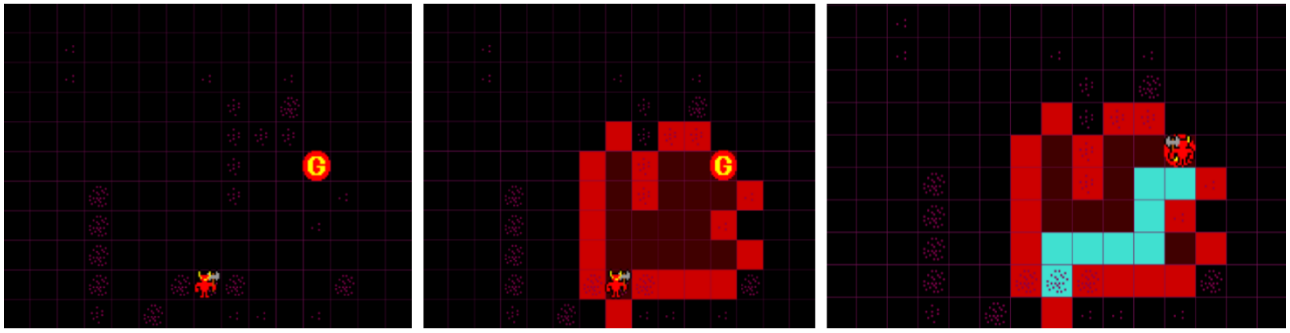

Figura 4 – Percurso sem obstáculos

Tempo utilizando a fórmula de Manhattan	**28.12 segundos**
Tempo utilizando a fórmula da Hipotenusa	**38.57 segundos**

Tabela 1- Tempos obtidos na Experiência 1

Experiência 2, percurso com obstáculos (exemplo 1):

Nesta experiência (figura 5) o agente encontra-se inserido num mundo com obstáculos, o que faz com que a expansão do mundo seja feita considerando os mesmos, não expandindo as casas com obstáculos. Essa expansão só termina quando o objetivo for atingido e quando não houver um novo nó fronteira com menor custo que ainda não tenha sido explorado.
Na imagem mais à esquerda, temos a localização do objetivo e do agente, na do meio o resultado da pesquisa, e, finalmente, na imagem mais à direita temos, além do resultado da pesquisa, o caminho escolhido pelo agente para chegar ao objetivo. As cores apresentadas correspondem ao seguinte: vermelho escuro para os nós pesquisados pela função de pesquisa, vermelho claro os nós fronteira no final da pesquisa, e azul claro os nós usados pelo agente para atingir o objetivo.

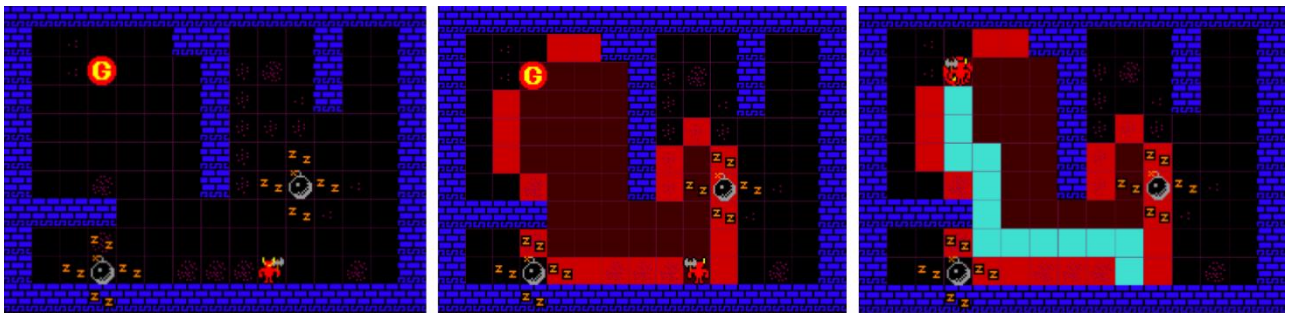

Figura 5 – Percurso com obstáculos 1

Tempo utilizando a fórmula de Manhattan	**25.06 segundos**
Tempo utilizando a fórmula da Hipotenusa	**35.83 segundos**

Tabela 2 – Tempos obtidos na Experiência 2

Experiência 3, percurso com obstáculos (exemplo 2):

Nesta experiência (figura 6), inserimos três obstáculos à volta do objetivo. Assim, conseguimos perceber que o agente ignora os caminhos com obstáculos.
Na imagem mais à esquerda, temos a localização do objetivo e do agente, na do meio o resultado da pesquisa, e, finalmente, na imagem mais à direita temos, além do resultado da pesquisa, o caminho escolhido pelo agente para chegar ao objetivo. As cores apresentadas correspondem ao seguinte: vermelho escuro para os nós pesquisados pela função de pesquisa, vermelho claro os nós fronteira no final da pesquisa, e azul claro os nós usados pelo agente para atingir o objetivo.

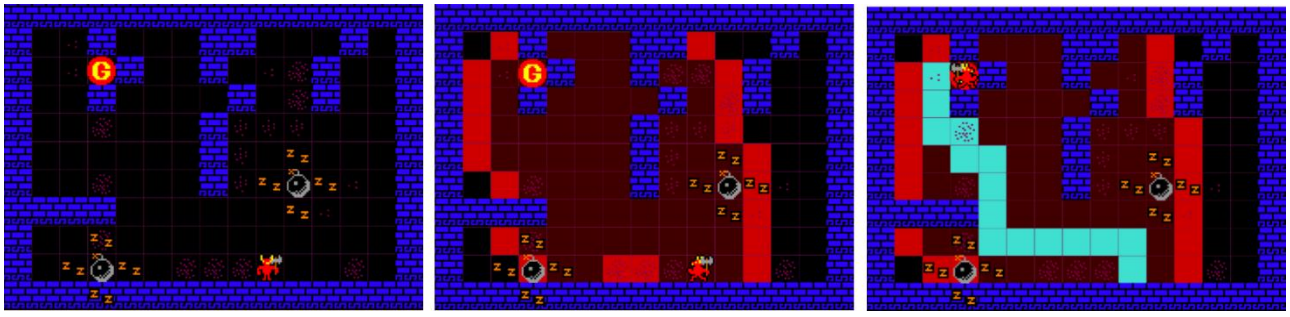

Figura 6 – Percurso com obstáculos 2

Tempo utilizando a fórmula de Manhattan	**44.70 segundos**
Tempo utilizando a fórmula da Hipotenusa	**55.58 segundos**

Tabela 3 - Tempos obtidos na Experiência 3

Experiência 4, percurso com obstáculos (exemplo 3):

Nesta experiência (figura 7), inserimos um quarto obstáculo à volta do objetivo, sendo assim, é observado que não existe nenhum caminho possível para atingir o objetivo.
A expansão é feita inserindo os nós explorados na fila dos nós visitados e os nós ainda por expandir são inseridos na fila dos nós fronteira.
A uma dada altura o agente fica sem nenhum nó por expandir visto que a fila dos nós fronteira encontra-se vazia.
Na imagem mais à esquerda, temos a localização do objetivo e do agente e na imagem mais à direita temos a pesquisa feita pelo agente pelo mapa sem conseguir atingir o objetivo por este ser impossível de alcançar. Neste caso existe uma única cor, vermelho escuro, representa todos os nós pesquisados até notar que não consegue atingir o objetivo.

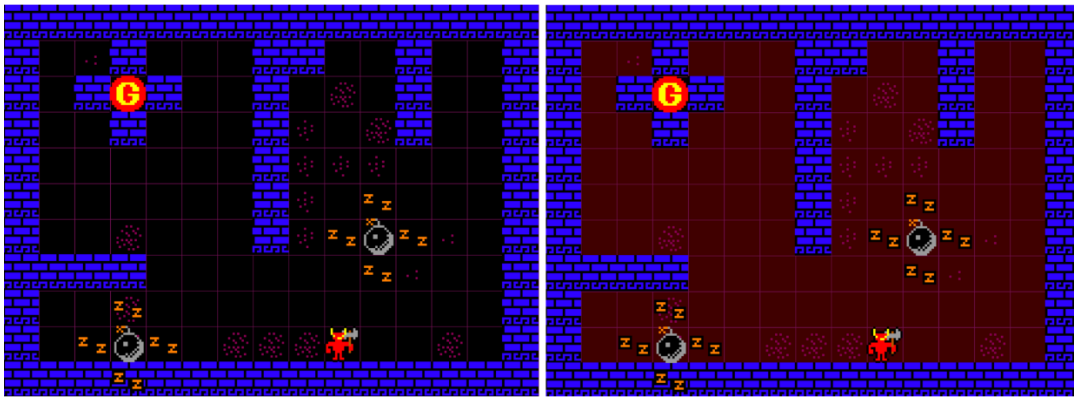

Figura 7 - Percurso com obstáculos 3

Tempo utilizando a fórmula de Manhattan	**N/A**
Tempo utilizando a fórmula da Hipotenusa	**N/A**

Tabela 4 – Tempos obtidos na Experiência 4

```
    node_to_expand = self.frontier_nodes.pop()
  File "C:\Users\K\Documents\GitHub\Projeto_G10\client\a_star.py", line 24, in pop
    return self.queue_data.pop(0)
IndexError: pop from empty list
```

Figura 8 - Erro dado quando o agente não encontra o objetivo

(tenta remover elemento de uma lista vazia)

Discussão e Conclusão

Através deste projeto, é possível concluir que a implementação do algoritmo A* proporcionou-nos a capacidade de interpretar e implementar um algoritmo num ambiente, fazer experiências com este e perceber o comportamento do mesmo nas várias situações a que foi sujeito.
Foi possível observar na figura 9,10 que as fórmulas matemáticas utlizadas para o cálculo da heurística influenciam no desempenho do algoritmo. Utilizando a fórmula do cálculo da hipotenusa, cuja fórmula é a raiz quadrada da soma da diferença entre a abcissa dos pontos ao quadrado e a diferença entre a ordenada dos pontos ao quadrado (figura 10) o algoritmo leva muito mais tempo em relação ao cálculo utilizado na fórmula de Manhattan (figura 9) porque a mancha criada pelo algoritmo que é maior. Este facto tem a ver com o valor calculado ser mais aproximado ao valor real.
Concluímos que que a fórmula de Manhattan seria a mais eficiente devido ao agente chegar ao objetivo com menor tempo que a fórmula da hipotenusa e devido à forma como o mundo é expandido.

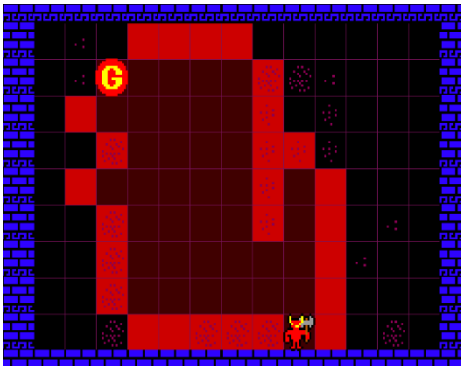

Figura 9 – Cálculo da heurística com base na distância de Manhattan

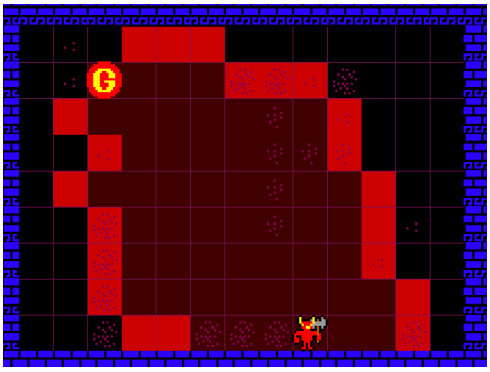

Figura 10 – Cálculo da heurística com base na hipotenusa

A utilização de uma heurística permite orientar a procura pelos ramos mais promissores, reduzindo o espaço de procura e, dessa forma, o tempo e o espaço utilizados nessa procura.
Na experiência dois, o tempo de pesquisa da experiência 2 levou menos tempo a atingir o objetivo foi menor em relação ao da experiência 1, isto experiência anterior. Isto porque o agente encontra-se inserido num mundo com obstáculos, o que faz com que o agente tenha menos nós por expandir.

Bibliografia

https://web.archive.org/web/20061222182121/http://policyalmanac.org/games/aStarTutorial_port.htm
Russel, Stuart J.; Norvig, Peter (2010) Artifical Intelligence, A Modern Aproach, Third Edition