POLITÉCNICO DE LEIRIA

DEPARTAMENTO DE ENGENHARIA INFORMÁTICA

LICENCIATURA EM ENGENHARIA INFORMÁTICA

INTELIGÊNCIA ARTIFICIAL

2021/2022 – 2º semestre

PROJETO – MUMMY MAZE SOLVER

---

1. Mummy Maze

O *Mummy* Maze é um jogo onde o herói é um caçador de tesouros. Enquanto busca por tesouros perdidos, a personagem principal do jogo vai atravessando vários níveis onde tem de evitar ser apanhado por inimigos e pisar armadilhas.

Em cada nível ou câmara está uma passagem para o nível ou câmara seguinte. Desta forma, o objetivo do herói é deslocar-se para a zona de acesso ao próximo nível evitando quaisquer perigos que possam ocorrer: ser morto por um inimigo ou cair numa armadilha. A zona de acesso ao nível seguinte é a célula que tem uma escada adjacente (ver figura seguinte).

Este jogo é jogado por turnos, onde primeiro se move o Herói e em seguida se deslocam os inimigos. Em cada turno, o Herói é sempre o primeiro a deslocar-se e os inimigos são sempre os últimos, mesmo que aquele já tenha chegado à célula objetivo.

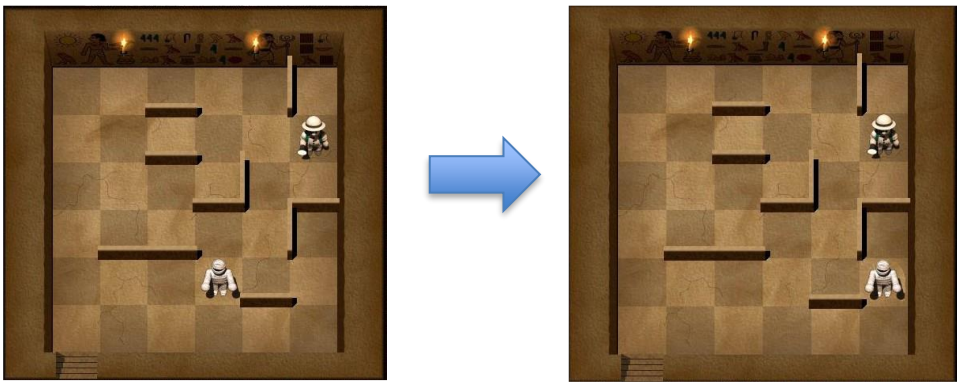
2. Elementos constituintes do jogo

Herói

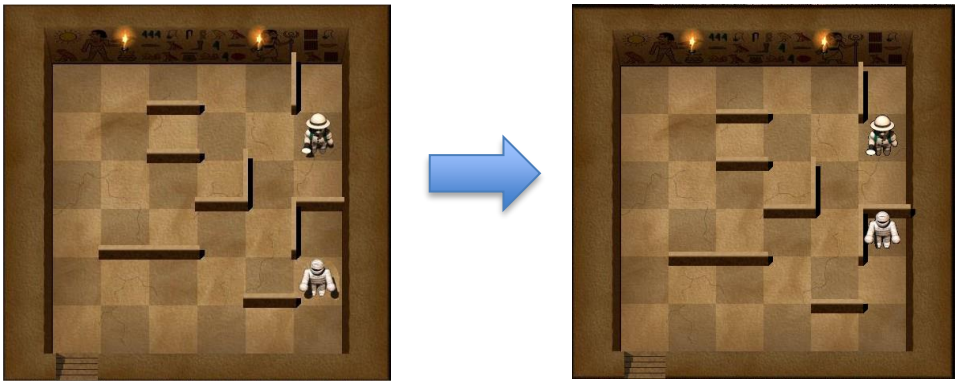
O Herói consegue deslocar-se apenas uma casa de cada vez em cada turno. Isto é: em cada turno o herói pode passar para a quadrícula imediatamente acima, abaixo, à esquerda ou à direita da quadrícula onde se encontra, a não ser que haja uma parede a bloquear esse caminho. O Herói pode ainda optar por não se movimentar durante o seu turno.

Múmia Branca

A múmia branca desloca-se até duas casas em cada turno. O objetivo da múmia é matar o Herói e para isso vai tentando deslocar-se para a posição onde este está. A múmia branca primeiro procura estar na mesma coluna onde se encontra o Herói e só depois é que tenta deslocar-se para a mesma linha que este. Usando o exemplo anterior, caso o Herói não efetuasse nenhum movimento, a movimentação da múmia seria duas células para a direita, de forma a ficar na mesma coluna que o Herói (ver figura seguinte).



Caso o Herói não se volte a mover durante o seu turno, o movimento da múmia branca será o seguinte:



Neste caso, a múmia branca desloca-se apenas uma casa: como já está na mesma coluna que o Herói, tenta posicionar-se na mesma linha indo para cima; após o primeiro passo para cima, a múmia branca encontra uma parede, o que a impede de se deslocar mais para cima.

Na situação ilustrada na figura abaixo, a múmia tenta deslocar-se primeiro para a esquerda mas não consegue. Como ainda não realizou nenhum movimento, tenta de seguida aproximar-se pelas linhas e desce uma casa. Como ainda lhe resta um movimento, tenta agora aproximar-se novamente pelas colunas e move-se para a esquerda, matando o herói. Ou seja, a múmia tenta sempre gastar os dois movimentos, dando sempre prioridade à aproximação pelas colunas.

Múmia Vermelha

Tal como a múmia branca, a múmia vermelha pode realizar até dois movimentos em cada turno. A múmia vermelha procura primeiro estar na mesma linha que o herói e só depois é que se desloca para a mesma coluna.

Escorpião

O escorpião desloca-se da mesma forma que a múmia branca, embora apenas se desloque uma quadrícula por turno.

Chave

Sempre que o Herói passa pela quadrícula onde se encontra a chave, abre ou fecha uma porta. Se a porta estiver fechada passa a estar aberta e vice-versa.

Porta

Quando está fechada, a porta funciona como uma parede: bloqueia os movimentos do Herói e seus inimigos.

Armadilha

O Herói morre sempre que pisar uma armadilha. As armadilhas não têm efeito nos inimigos.

3. Como matar múmias

Em alguns níveis onde existam duas múmias pode ser necessário eliminar primeiro uma múmia para que o nível possa ter solução. Caso o Herói consiga fazer com que duas múmias ocupem a mesma casa, elas lutam entre si restando apenas uma múmia viva. A múmia morta desaparece do nível.

4. Trabalho a Realizar

Pretende-se que seja desenvolvida uma aplicação que, recorrendo aos algoritmos de procura implementados nas aulas práticas, seja capaz de jogar o jogo Mummy Maze. Para os algoritmos de procura informados deverão ser formuladas pelo menos duas heurísticas distintas adequadas ao problema. O programa deverá permitir ao utilizador escolher um determinado nível a resolver, o algoritmo de procura e a heurística a utilizar (se aplicável).

Pretende-se que seja também realizado um estudo comparativo do desempenho dos vários algoritmos de procura bem como das heurísticas que forem utilizadas na resolução dos problemas. Nomeadamente, pretende-se estudar os seguintes aspetos:

- O desempenho dos algoritmos de procura não informados;
- O desempenho dos algoritmos de procura informados;
- O desempenho dos algoritmos de procura não informados versus o desempenho dos algoritmos de procura informados;
- A qualidade das heurísticas utilizadas.

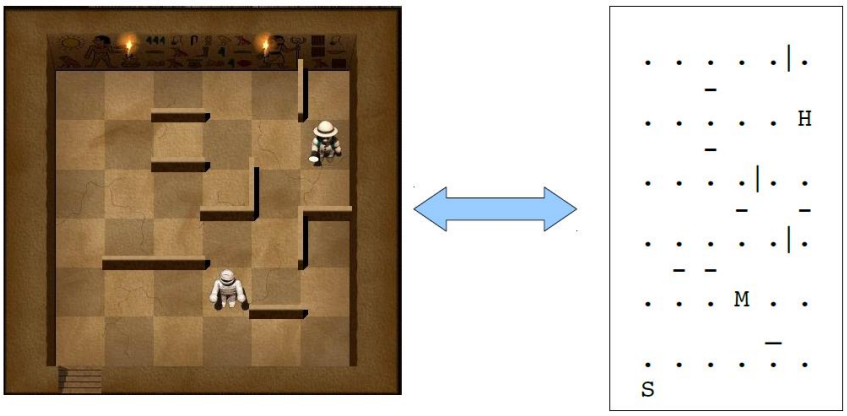
5. Representação dos estados

São muitos os níveis que constituem o jogo Mummy Maze. Neste projeto, o programa a desenvolver deve estar preparado para jogar apenas com níveis com as dimensões iguais aos do nível apresentado nas figuras anteriores: uma matriz de 6 x 6 células. No entanto, deverá ser possível definir diferentes configurações para os níveis. Para isso, deverão ser utilizados ficheiros de texto que permitam definir diferentes níveis: a posição inicial do Herói e dos inimigos, bem como as paredes, chaves, portas, armadilhas e a posição do objetivo.

O conteúdo dos ficheiros de configuração deverá consistir numa matriz de 13 x 13 caracteres em que cada carácter representa uma célula do nível, uma parede ou a saída. Os caracteres a utilizar deverão ser os seguintes:

- '|' ou '–', para uma posição ocupada por uma parede;
- 'H', para o agente;
- '.', para uma posição vazia;
- 'S', para a localização da saída;
- 'M', para a múmia branca;
- 'V', para a múmia vermelha;
- 'A', para a armadilha;
- 'E', para o escorpião;
- 'C', para a chave;
- '=', para porta horizontal fechada;
- '_', para porta horizontal aberta;
- '”', para porta vertical fechada;
- “)”, para porta vertical aberta.

Desta forma, o nível apresentado na figura abaixo à esquerda tem a seguinte representação representada à direita:



No sítio da cadeira, juntamente com o enunciado, está disponível um ficheiro denominado `niveis.zip` onde se encontram alguns níveis a resolver já neste formato. Os estudantes são encorajados a criar novos níveis e até a publicar os mesmos no fórum da unidade curricular para que possam comparar diferentes soluções e heurísticas para o mesmo problema.

6. Visualização das soluções

De modo a permitir uma visualização mais agradável das soluções encontradas é fornecida a biblioteca `MummyMazeShowSolution.jar`. Esta biblioteca deverá ser incluída na aplicação a desenvolver e contém a classe `SolutionPanel`, pertencente à package `showSolution` – onde deverá ser invocado o método `showSolution(List<String> states, double solutionCost)`. Este método é responsável por mostrar graficamente a sequência de estados resultantes da solução encontrada pelos algoritmos de procura e o custo da mesma. Para o funcionamento correto deste componente é necessário que seja passada como argumento deste método uma lista com as *strings* correspondentes à sequência de estados que resultam da aplicação da solução bem como o custo da solução encontrada. As *strings* enviadas têm de obedecer forçosamente ao formato especificado anteriormente.

7. Relatório

Do relatório deverá constar:

- A descrição da representação utilizada para cada estado;
- A descrição das heurísticas utilizadas;
- A apresentação e discussão dos resultados obtidos;
- A descrição da contribuição de cada elemento do grupo no desenvolvimento do projeto e na escrita do relatório;
- Outros aspetos considerados relevantes para uma boa compreensão e avaliação do trabalho realizado.

Alguns dos fatores com mais importância na avaliação do relatório são:

- Clareza na descrição dos componentes da aplicação;
- A forma como os resultados dos testes são compilados e a clareza com que são apresentados (a utilização, mas não só, de tabelas e/ou gráficos pode ajudar);
- A análise e discussão dos resultados.

8. Cotações

10% - Definição do problema
20% - Definição do estado
15% - Definição das ações
20% - Heurísticas
20% - Estudo e Relatório
15% - Extras

Extras sugeridos:

- Implementação de otimizações específicas a alguns algoritmos de procura desenvolvidos nas aulas de modo a torná-los mais eficientes (cabe aos estudantes identificar os algoritmos que podem ser otimizados bem como as otimizações a implementar);
- Implementação de outros algoritmos de procura;
- Realização de um conjunto de experiências de forma automática;
- Implementação de mais heurísticas.

9. Prazos, datas, regras e instruções

1. Data limite de entrega do projeto: **21 de junho de 2022, 23:59**.
2. Data da prova oral: **5 de julho de 2022**.
3. O projeto é realizado em grupos de 2 estudantes. Não são aceites projetos realizados por grupos com mais de 2 elementos. Os estudantes que pretendam realizar o projeto individualmente devem solicitá-lo, por escrito, ao docente responsável pela UC. Apenas em casos bem fundamentados serão autorizados projetos realizados individualmente.
4. O relatório deve ser realizado utilizando o modelo disponibilizado na secção Projeto do sítio da UC no Moodle.
5. O projeto deve ser entregue sob a forma de um arquivo zip, rar ou 7z que contenha todos os elementos do projeto, incluindo o relatório. O nome do arquivo deve ter o formato IA_Projeto_#1_#2.(zip/rar/7z), onde #1 e #2 devem ser substituídos pelos números de estudante dos elementos do grupo. O relatório deve ser entregue em formato pdf e o seu nome deve ter a mesma estrutura do arquivo mas com extensão pdf.