



UFMT – Universidade Federal do Estado de Mato Grosso
IC – Instituto de Computação

Relatório

Exercício Prático

Disciplina – Inteligência Artificial

Professora – Vanessa de Oliveira Campos

Alunos: Bruna Keiko Hatakeyama, Leony Tamio Hatakeyama e Lucimarck Junior da Silva Dias



UFMT – Universidade Federal do Estado de Mato Grosso
IC – Instituto de Computação

Introdução

A Inteligência Artificial (Artificial Intelligence - AI) figura com um avanço tecnológico que permite aos sistemas simularem uma inteligência similar à humana, mais especificamente a capacidade de aprender, perceber e decidir quais caminhos seguir, de forma racional, diante de determinadas situações, indo além da programação com ordens programadas para tomar decisões.

Justificativa

Aplicação dos conhecimentos adquiridos na disciplina de Inteligência Artificial, visando a resolução de um problema específico, utilizando: técnicas de buscas, funções heurísticas, lógica proposicional e lógica, árvores de decisão, algoritmos genéricos, outros.

Apresentação do problema

O Trabalho consiste em implementar um agente capaz de guiar autonomamente Seiya, Shiryu, Hyoga, Shun e Ikki pelas 12 Casas do Zodíaco, planejando a melhor forma de derrotar os 12 Cavaleiros de Ouro e salvar Atena.

Para isso, você deve utilizar um dos algoritmos de solução de problemas estudados na disciplina de inteligência artificial.

O agente deve ser capaz de calcular automaticamente a rota para percorrer as 12 Casas do Zodíaco e derrotar os 12 Cavaleiros de Ouro.

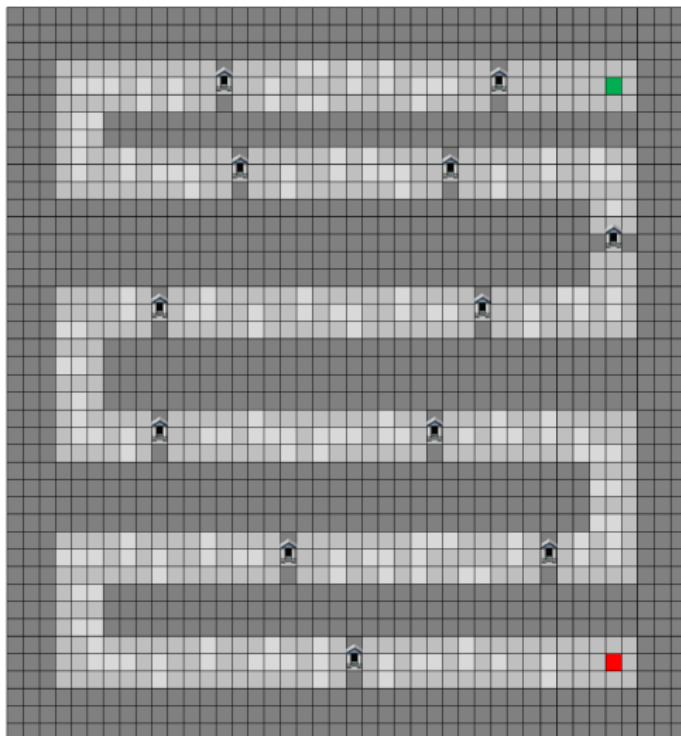


Figura 3. Mapa das 12 Casas do Zodíaco.

No caminho das 12 Casas do Zodíaco existem 3 tipos de terrenos: montanhoso (região cinza escuro), plano (região cinza) e rochoso (região cinza claro).

Para passar por cada tipo de terreno, os Cavaleiros gastam uma determinada quantidade de tempo:

Montanhoso: +200 minutos

Plano: +1 minuto

Rochoso: +5 minutos

Os Cavaleiros de Bronze iniciam a sua jornada na entrada do santuário (região em vermelho no mapa) e terminam ao



UFMT – Universidade Federal do Estado de Mato Grosso
IC – Instituto de Computação

chegar à casa do Grande Mestre (região verde no mapa).

Ao chegar a uma Casa do Zodíaco, o agente deve decidir quais Cavaleiros vão lutar contra o Cavaleiro de Ouro que protege a casa. Cada Cavaleiro de Ouro apresenta um nível de dificuldade diferente. Este nível determina o tempo gasto pelos Cavaleiros de Bronze para poder vencê-lo e avançar para a próxima Casa.

O número de Cavaleiros de Bronze participando das batalhas contra os Cavaleiros de Ouro

	Casa	Dificuldade
1°	Casa de Áries	50
2°	Casa de Touro	55
3°	Casa de Gêmeos	60
4°	Casa de Câncer	70
5°	Casa de Leão	75
6°	Casa de Virgem	80
7°	Casa de Libra	85
8°	Casa de Escorpião	90
9°	Casa de Sagitário	95
10°	Casa de Capricórnio	100
11°	Casa de Aquário	110
12°	Casa de Peixes	120

Tabela 1. Níveis de dificuldade das 12 Casas do Zodíaco.

influência o tempo gasto na batalha. Além disso, cada Cavaleiro possui um determinado nível de poder cósmico que também influencia no tempo gasto nas batalhas.

Quanto mais cavaleiros lutando, mais rápido o Cavaleiro de Ouro será derrotado.

O tempo gasto nas batalhas contra os Cavaleiros de Ouro é dado por:

Cavaleiro	Poder Cósmico
Seiya	1.5
Shiryu	1.4
Hyoga	1.3
Shun	1.2
Ikki	1.1

Tabela 2. Poder cósmico dos Cavaleiros de Bronze.



UFMT – Universidade Federal do Estado de Mato Grosso
IC – Instituto de Computação

$$\text{Tempo} = \frac{\text{Dificuldade da Casa}}{\sum \text{Poder Cosmico dos Cavaleiros Participando da Batalha}}$$

Além do poder cósmico, cada Cavaleiro de Bronze também possui 5 pontos de energia.

Ao participar de uma batalha, o Cavaleiro perde 1 ponto de energia. Se o Cavaleiro perder todos os pontos de energia, ele morre.

Informações Adicionais

- O mapa principal deve ser representado por uma matriz 42 x 42.
- O agente sempre inicia a jornada na entrada do santuário (região em vermelho no mapa).
- O agente sempre termina a sua jornada ao chegar à casa do Grande Mestre (região verde no mapa).
- O agente não pode andar na diagonal, somente na vertical e na horizontal.
- O agente obrigatoriamente deve utilizar um algoritmo de busca para encontrar o melhor caminho e planejar as batalhas.
- Deve existir uma maneira de visualizar os movimentos do agente, mesmo que a interface seja bem simples. Podendo até mesmo ser uma matriz desenhada e atualizada no console.
- Os mapas devem ser configuráveis, ou seja, deve ser possível modificar o tipo de terreno em cada local. O mapa pode ser lido de um arquivo de texto ou deve ser facilmente editável no código.
- A dificuldade das casas e o poder cósmico dos Cavaleiros de Bronze devem ser configuráveis e facilmente editáveis.
- O programa deve apresentar o caminho percorrido.
- O programa deve exibir o custo do caminho percorrido pelo agente ao terminar a execução.
- O programa deve apresentar as equipes para lutar contra os Cavaleiros de Bronze em cada uma das 12 Casas do Zodíaco, bem como o tempo gasto nas batalhas.
- O programa pode ser implementado em qualquer linguagem.
- O programa deverá ser executado em Windows (se bibliotecas auxiliares forem usadas, todos os arquivos necessários deverão ser incluídos no projeto para que ele possa ser executado).

Dicas

Neste trabalho existem dois problemas distintos:

- Encontrar o caminho para passar pelas 12 Casa do Zodíaco e chegar até a Casa do Grande Mestre;
- Encontrar a ordem de equipes para lutar contra os Cavaleiros de Bronze.



UFMT – Universidade Federal do Estado de Mato Grosso
IC – Instituto de Computação

Os dois problemas podem ser resolvidos individualmente ou tratando ambos em um único problema. O grupo deve definir a melhor maneira de estruturar a sua solução.

Método

Como um dos problemas do exercício seria encontrar o melhor caminho, considerando todas os aspectos e obstáculos do mapa e com base nos exemplos visto em aula, conclui-se que aquele cenário se tratava de uma busca heurística. Considerando que, utilizando o método de Busca Gulosa, cujo sempre busca encontrar o melhor caminho decidindo apenas em expandir a menor quantidade de nodos possíveis, concluímos que, para este cenário, a Busca A* seria a mais adequada pois ela, além de decidir quais nodos expandir, ela busca em encontrar o caminho com o menor custo resultando em soluções mais otimizadas. No código fonte da aplicação a implementação dessa busca heurística pode ser vista no arquivo **PathIA.cs** cujo o qual, essencialmente, parte da premissa que uma matriz de duas dimensões será fornecida pelo usuário que, no nosso caso, será a partir de um arquivo **.txt** fornecido pela docente com a configuração do mapa que é constituído de 42 linhas onde cada qual possui 42 valores separados por uma tabulação representando a legenda do mapa:

- 0 – Início
- 1:12 – Casas dos Cavaleiros de Ouro
- 13 – Final
- 14 – Montanhoso (+200 min)
- 15 – Plano (+1 min)
- 16 – Rochoso (+5 min)

Basicamente, a lógica utilizada para encontrar o melhor caminho seria em encontrar todos os destinos vizinhos da localização atual e agenda-los em uma estrutura de dados FIFO (*First In First Out*) popularmente conhecida como Fila. Logo, para cada vizinho é verificado se ele já está no percurso do caminho atual, caso sim, verifica-se também se o custo do percurso do vizinho é maior que o custo do percurso atual somados com o tempo necessário para passar naquele tipo terreno, se sim, o caminho do vizinho é atualizado a partir do caminho atual incluindo a localização do mesmo e ao final ele é agendado na Fila. Contudo, caso o vizinho não esteja dentro do percurso atual, um caminho novo é gerado dentro das possíveis soluções e também o mesmo é agendado na Fila.

Por outro, a lógica utilizada para encontrar a melhor combinação de batalha, resumidamente se baseia nos conceitos dos algoritmos genéticos e cujo o seu gene é formado por 13 combinações possíveis (12 Casas de Ouro + Final) alocando os cavaleiros de modo que sejam utilizados apenas 5 vezes dentro de batalhas durante a sessão (limitação imposta nas instruções). Logo, o algoritmo genético consiste, inicialmente, em criar 5 soluções possíveis e dentro delas são criadas mais 300 instâncias que, ao final, são ordenadas de forma crescente utilizando a somatória do tempo necessário para passar pelas 12 casas usando as combinações geradas pela mutação, que se resume em mover os cavaleiros de uma casa a outra, pegando dessas 300 instâncias as 5 melhores



UFMT – Universidade Federal do Estado de Mato Grosso
IC – Instituto de Computação

combinações. Além disso, uma simples rede neural é formada que, o resultado obtido pelo PathIA é passado para o BattleIA. O processo repete até ele atender algumas condições:

- Das 5 instâncias, os valores do primeiro e do último devem ser diferentes **OU**
- A melhor combinação dessas 5 instâncias, somado com o resultado de PathIA, não deve ultrapassar de 12 Horas (Tempo máximo descrito no anime para salvar a Deusa Athena) **E**
- O processo não pode ultrapassar de 60 repetições (Para evitar um possível Loop Infinito).

Aplicação

A aplicação foi desenvolvida utilizando a linguagem C# com o framework .NET Core utilizando o modelo de aplicativo web Blazor. Para utiliza-lo há três possíveis maneiras:

- Ir até a página web em qualquer navegador no endereço: <https://ic-ufmt-ia.github.io/>
 - Obs.: Como o modelo de aplicativo Web Assembly ainda se encontra em desenvolvimento, o desempenho dele será drasticamente afetado nesse cenário, recomendamos este modo como ultimo recurso disponível.
- Execute o executável **CDZ.Web.Server.exe** dentro da pasta **_exe** que ele abrirá uma janela console exibindo que um servidor web foi inicializado, mas atenção nas portas utilizadas pois qualquer aplicação web rodando em algumas dessas portas, a aplicação terá problemas em iniciar podendo até mesmo ter a sua execução cancelada. Logo após, a aplicação poderá ser acessada em qualquer navegador web (Recomendamos os baseados em Chromium. Ex.: Google Chrome, Microsoft Edge) com os seguintes endereços de acesso:
 - <http://localhost:80>
 - <https://localhost:443>
- Com .NET Core SDK 3.1 instalado em sua máquina, dentro da pasta CDZ.Web.Server contido no código fonte da aplicação, execute o seguinte comando:
 - **dotnet run --configuration Release**

Ao acessar a página web, a seguinte interface de usuário será mostrada:



UFMT – Universidade Federal do Estado de Mato Grosso
IC – Instituto de Computação

Bronze

Seiya Pégasus 1.5	Mu Áries 50	Dohko Libra 85
Shiryu Dragão 1.4	Aldebaran Touro 55	Milo Escorpião 90
Hyoga Cisne 1.3	Saga Gêmeos 60	Aiolos Sagitário 95
Shun Andrômeda 1.2	Mascára da Morte Câncer 70	Shura Capricórnio 100
Ikki Fênix 1.1	Aiolia Leão 75	Camus Aquário 110
	Shaka Virgem 80	Afrodite Peixes 120

Ouro

Velocidade
0 ms

0⁺

Nela você poderá customizar o poder de Cosmo de qualquer cavaleiro:

Bronze

Seiya Pégasus 2.8	Mu Áries 50	Dohko Libra 85
Shiryu Dragão 1.4	Aldebaran Touro 55	Milo Escorpião 90
Hyoga Cisne 1.3	Saga Gêmeos 60	Aiolos Sagitário 95
Shun Andrômeda 1.2	Mascára da Morte Câncer 70	Shura Capricórnio 100
Ikki Fênix 1.1	Aiolia Leão 75	Camus Aquário 110
	Shaka Virgem 80	Afrodite Peixes 120

Ouro



UFMT – Universidade Federal do Estado de Mato Grosso
IC – Instituto de Computação

Customizar a velocidade animação em milissegundos com os valores entre 0 até 3000 milissegundos.

Velocidade

240 ms

0⁺⁰

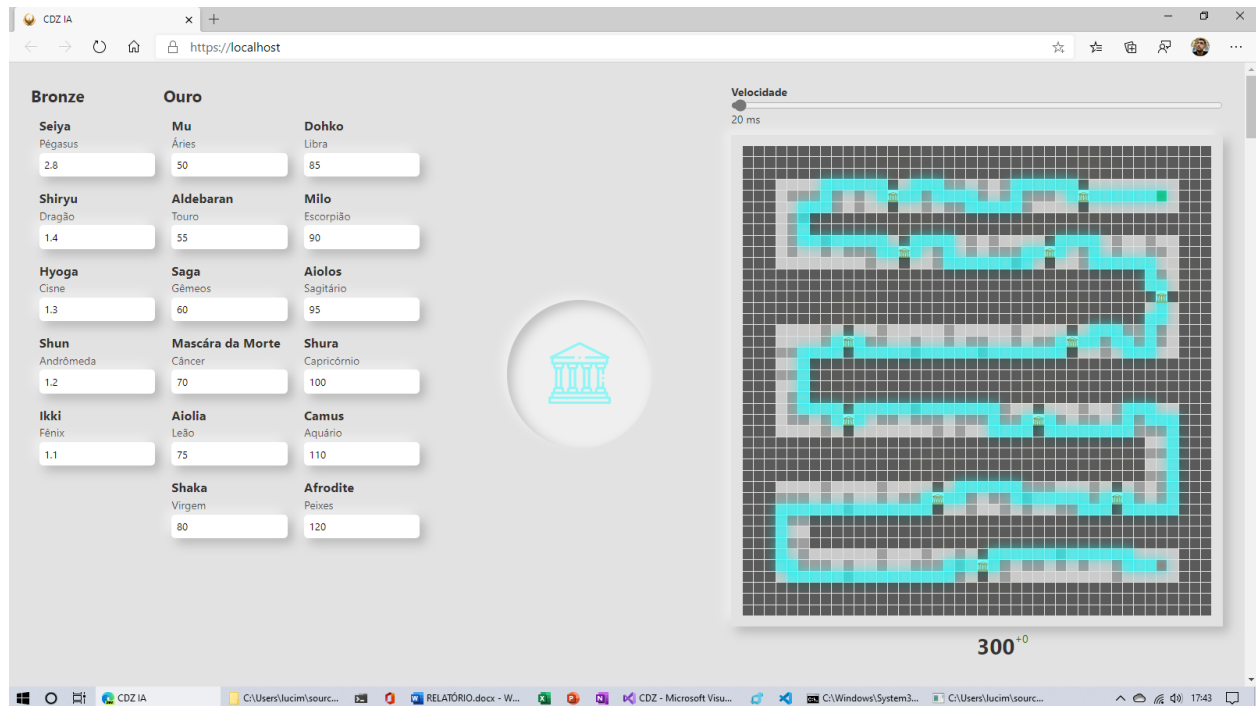
Para iniciar a simulação, clique no botão central com o ícone de um templo grego para calcular e exibir os resultados.



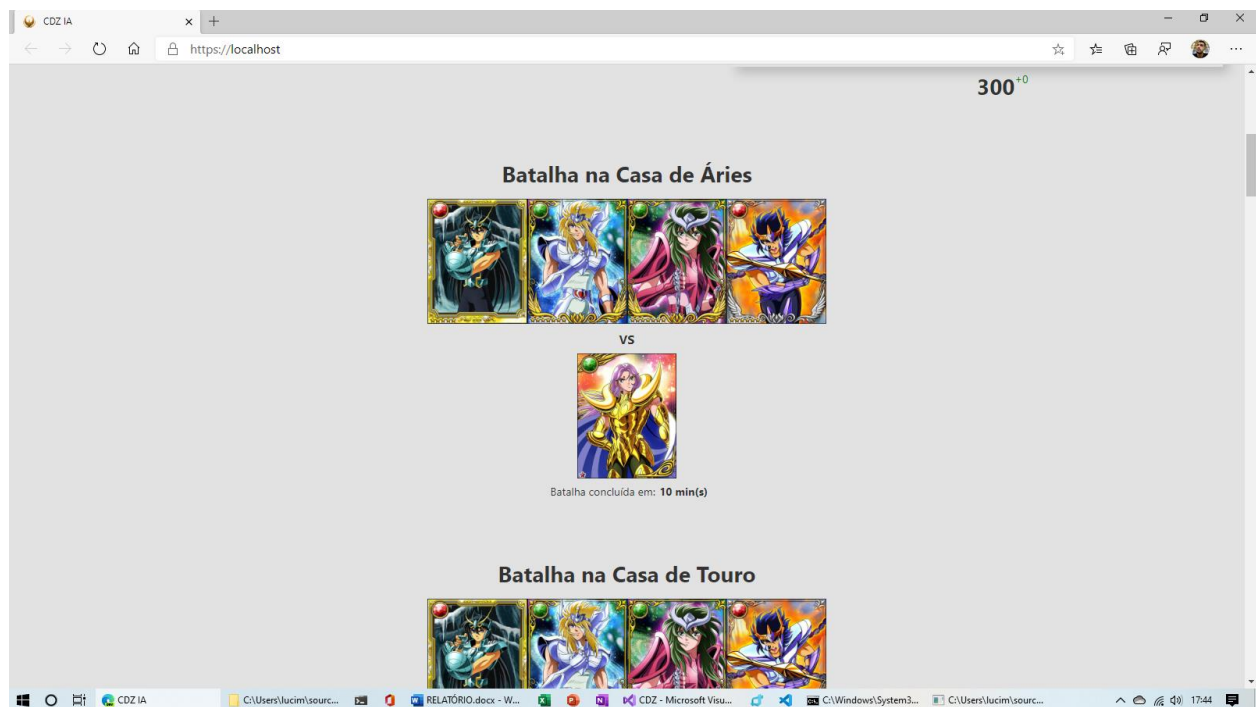
UFMT – Universidade Federal do Estado de Mato Grosso
IC – Instituto de Computação

Resultados

Dentre os resultados, podemos destacar:



O trajeto identificado pelo agente foi avaliado em 300 minutos para concluir o percurso. Abaixo os resultados de cada batalha, bem como as combinações utilizadas, poderão ser vistas na seguinte forma:






UFMT – Universidade Federal do Estado de Mato Grosso
IC – Instituto de Computação



Ao final, o total da sessão é mostrado realizando a somatória do tempo necessário do percurso encontrado pelo agente com a somatória do tempo necessário para passar em casa dos cavaleiros de ouro gerado a partir das combinações criadas pelo agente.

CDZ IA x +

← → ↻ 🏠 🔒 https://localhost ☆ 🗑️ 🔄 🧑


Batalha concluída em: 30,76923076923077 min(s)

Batalha na Casa de Grande Mestre


VS

Batalha concluída em: 0 min(s)

Total
300 + 397,52980506639045 =
697,5298050663905

Windows taskbar: CDZ IA, C:\Users\lucim\sourc..., RELATÓRIO.docx - W..., CDZ - Microsoft Visu..., C:\Windows\System3..., C:\Users\lucim\sourc..., 17:45