

INF1771 - INTELIGÊNCIA ARTIFICIAL

TRABALHO 1 – BUSCA HEURÍSTICA

Descrição:

“Um grupo de elite dos aliados foi designado para atacar estrategicamente bases inimigas que foram construídas entre os desfiladeiros de Tikrit no oriente médio. Para esta missão serão utilizados 5 caça-aviões que deverão percorrer um percurso e destruir as 11 bases inimigas o mais rápido possível para que se possa evitar um contra-ataque. Os caças devem seguir juntos, iniciando no começo do desfiladeiro, atacar as 11 bases e alcançar a base aliada que foi instalada ao final do percurso.

Grandes perigos aguardam o grupo de elite dos aliados. Além das bases inimigas, existem alguns trechos onde os ventos são mais fortes e devem ser evitados pois podem gerar turbulência. Ainda existem trechos onde os inimigos instalaram baterias antiaérea e trechos de planalto que devem ser evitados pois não foram totalmente mapeados e são considerados extremamente perigosos. ”



Figura 1. O desfiladeiro de Tikrit.

O Trabalho 1 consiste em implementar um agente capaz de guiar autonomamente os aviões através do desfiladeiro para destruir as forças inimigas, planejando a melhor forma de atacar as suas bases. Para isso, você deve utilizar o **algoritmo de busca heurística A***.

O agente deve ser capaz de calcular automaticamente a **melhor rota para percorrer as 11 bases inimigas no menor tempo possível**. O Mapa da região de Tikrit é mostrado na figura 2.

No trecho do desfiladeiro onde encontram-se as 11 bases 4 **tipos de terrenos**: planatos desconhecidos (região cinza escuro), plano (região cinza), turbulento (região cinza claro) e com bateria anti-aérea. Para passar por cada tipo de terreno, os caça-aviões gastam uma determinada quantidade de tempo devido o perigo eminente:

- **Montanhoso (em cinza escuro):** +200 minutos
- **Plano (em branco):** +1 minuto
- **Rochoso (em cinza claro):** +5 minutos

- **Bateria Anti-aérea (em vermelho): + 50 minutos**

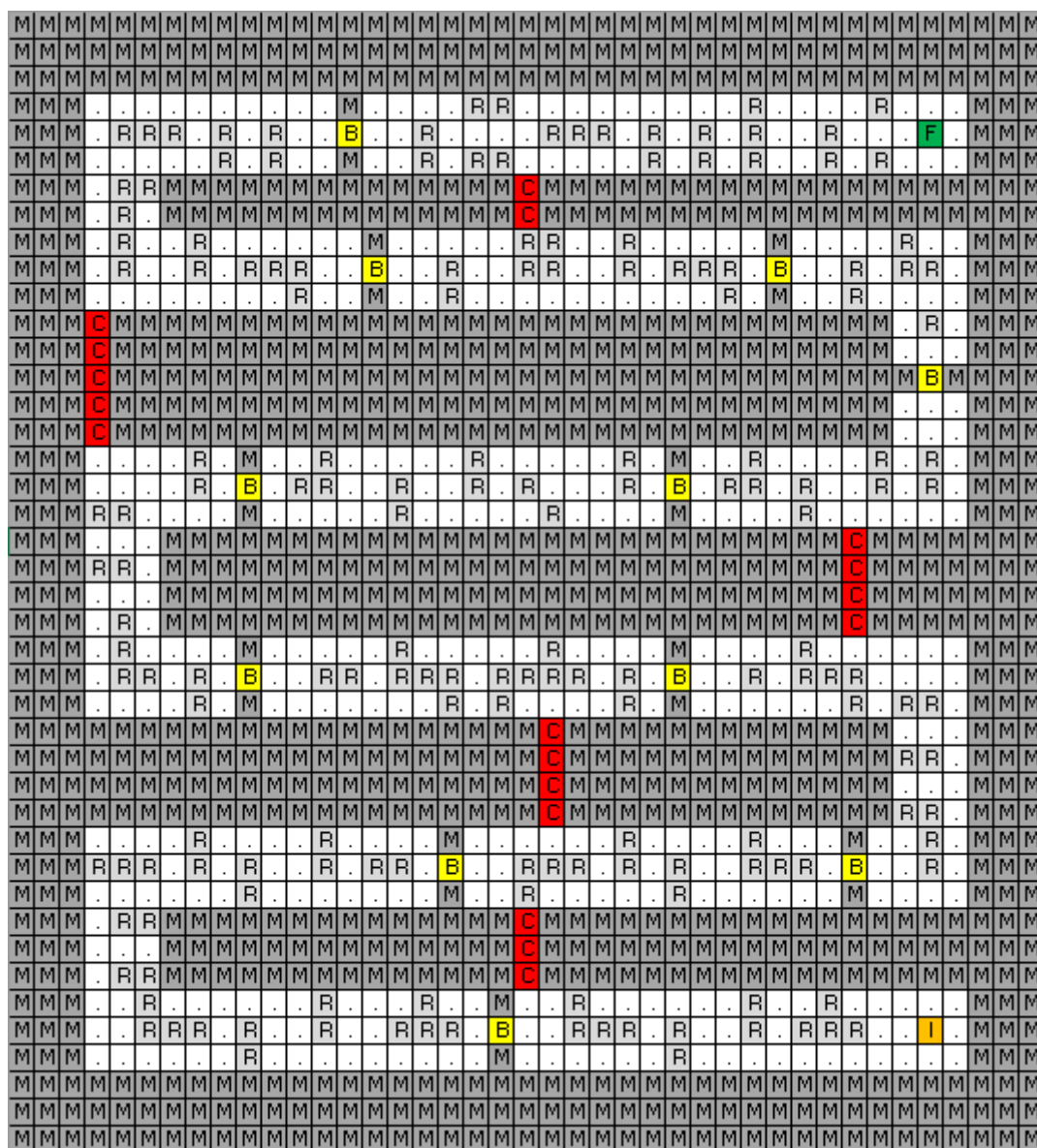


Figura 2. Mapa estratégico do desfiladeiro de Tikrit.

Os Caça-aviões iniciam a sua jornada na **entrada do desfiladeiro** (região em laranja no mapa) e terminam ao chegar à **base aliada** (região verde no mapa).

Ao chegar a uma base, o agente deve **decidir quais Aviões serão usados na batalha**. Cada **base inimiga** apresenta um **nível de dificuldade** diferente de acordo com seus armamentos e preparo. Este nível determina o tempo gasto pelos Aviões para poder vencer a batalha e avançar. A tabela abaixo mostra os níveis de dificuldade das 11 bases inimigas.

Ordem da Base	Dificuldade
1ª Base	60
2ª Base	65
3ª Base	70
4ª Base	75
5ª Base	80
6ª Base	85
7ª Base	90
8ª Base	95
9ª Base	100
10ª Base	110
11ª Base	120

Tabela 1. Níveis de dificuldade das 11 Bases Inimigas.

O **número de aviões** participando das batalhas contra as bases influencia no tempo gasto na batalha. Além disso, cada avião possui um determinado nível de **poder de fogo** que também influencia no tempo gasto nas batalhas. Quanto mais aviões atacando, mais rápido a base é destruída. A tabela abaixo mostra o poder de fogo dos aviões.

Caça-Avião	Poder de Fogo
F-22 Raptor	1.5
F-35 Lightning II	1.4
T-50 PAK FA	1.3
Su-46	1.2
MiG-35	1.1

Tabela 2. Poder de fogo dos aviões.

O tempo gasto nas batalhas contra as bases inimigas é dado por:

$$Tempo = \frac{Dificuldade\ da\ base\ inimiga}{\sum Poder\ de\ fogo\ dos\ aviões\ que\ participam\ da\ batalha}$$

Além do poder cósmico, cada avião também possui **5 pontos de energia**.

Ao participar de uma batalha, o cada avião perde -1 ponto de energia. Se o avião perder todos os pontos de energia, ele cai/explode.

Informações Adicionais:

- O mapa principal deve ser representado por uma matriz 42 x 42 (igual à mostrada na Figura 3).
- O agente sempre **inicia** a jornada na entrada dos desfiladeiros (região em vermelho no mapa).

- O agente sempre **termina** a sua jornada ao chegar à base aliada (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 bases e o poder de ataque** dos aviões devem ser configuráveis e facilmente editáveis.
- O programa deve **exibir o custo do caminho** percorrido pelo agente enquanto ele se movimenta pelo mapa e também o **custo final** ao terminar a execução.
- O programa pode ser implementado em **qualquer linguagem**.
- O trabalho pode ser feito **individualmente** ou em **grupos** de no máximo 3 pessoas.
- O programa deve ser apresentado durante a aula por **todos os membros do grupo**:
 - 1) O membro do grupo que **não comparecer** receberá nota **zero**;
 - 2) O membro do grupo que **não souber explicar** algo relacionado ao trabalho perderá 5.0 pontos.

Dicas:

- Neste trabalho existem dois problemas distintos:
 - 1) Encontrar o melhor caminho para passar pelas bases;
 - 2) Encontrar a melhor combinação de equipes para atacar as bases.
- Os dois problemas podem ser resolvidos individualmente ou tratando ambos em um único problema. Você deve definir a melhor maneira de estruturar a sua solução.

Forma de Avaliação:

Será avaliado se:

- (1) O trabalho atendeu a todos os requisitos especificados anteriormente;
- (2) Os algoritmos foram implementados e aplicados de forma correta;
- (3) O código foi devidamente organizado;
- (4) O trabalho foi apresentado corretamente em sala de aula;

Obs: O código deverá ser disponibilizado através de um repositório GitHub.

Bônus:

- (1) A interface gráfica não é o objetivo desse trabalho, mas quem implementar uma **“boa” interface gráfica** (2D ou 3D) para representar o ambiente e o agente receberá até 2 pontos extras na nota.
- (2) O trabalho que conseguir encontrar a **solução ótima no menor tempo de execução do algoritmo de busca**, receberá 1.0 ponto extra na nota. Para poder participar da competição, o trabalho deverá implementar um mecanismo para calcular o tempo de execução do algoritmo e 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).