

# INF1771 - INTELIGÊNCIA ARTIFICIAL

## TRABALHO 1 – BUSCA HEURÍSTICA

### Descrição:

*“Um promissor treinador de pokemons da cidade de Viridian foi convidado para continuar seu treinamento na cidade Neon (uma cidade longe de sua terra natal). Para isto, deverá cruzar parte da região de Kanto para conseguir seu tão sonhado treinamento. Nesta missão, nosso herói está levando seus 5 pokemons e devera cruzar diversas cidades – 12 delas com ginásios Pokemon.*

*Grandes perigos o aguardam nesta aventura! Além dos ginásios, existem alguns trechos com florestas densas e florestas mais fáceis de trafegar. Lembre-se de poupar seus pokemons e de que ao menos um deverá chegar com o mínimo de energia para que se possa iniciar seu treinamento. Avante nesta aventura! Será que nosso herói conseguirá pegar todos? ”*



Figura 1. Pokemons da região de Kanto (todos os 151)

O Trabalho 1 consiste em implementar um agente capaz de guiar nosso competidor através dos ginásios que estão no caminho entre a sua cidade natal e a cidade destino. Para isso, você deve utilizar o **algoritmo de busca heurística A\***.

O agente deve ser capaz de calcular automaticamente a **melhor rota** para **percorrer por até 12 ginásios no menor tempo possível**. O Mapa da região é mostrado na figura 2.

No trecho do desfiladeiro onde encontram-se os 12 ginásios há 3 **tipos de terrenos**: florestas densas (região cinza escuro), pastagens livres (região em branco) e florestas (região cinza claro). Para passar por cada tipo de terreno, nosso herói gasta uma determinada quantidade de tempo devido à dificuldade do terreno:

- Floresta densa (em cinza escuro): +200 minutos
- Floresta (em cinza claro): +5 minutos
- Livre (em branco): +1 minuto

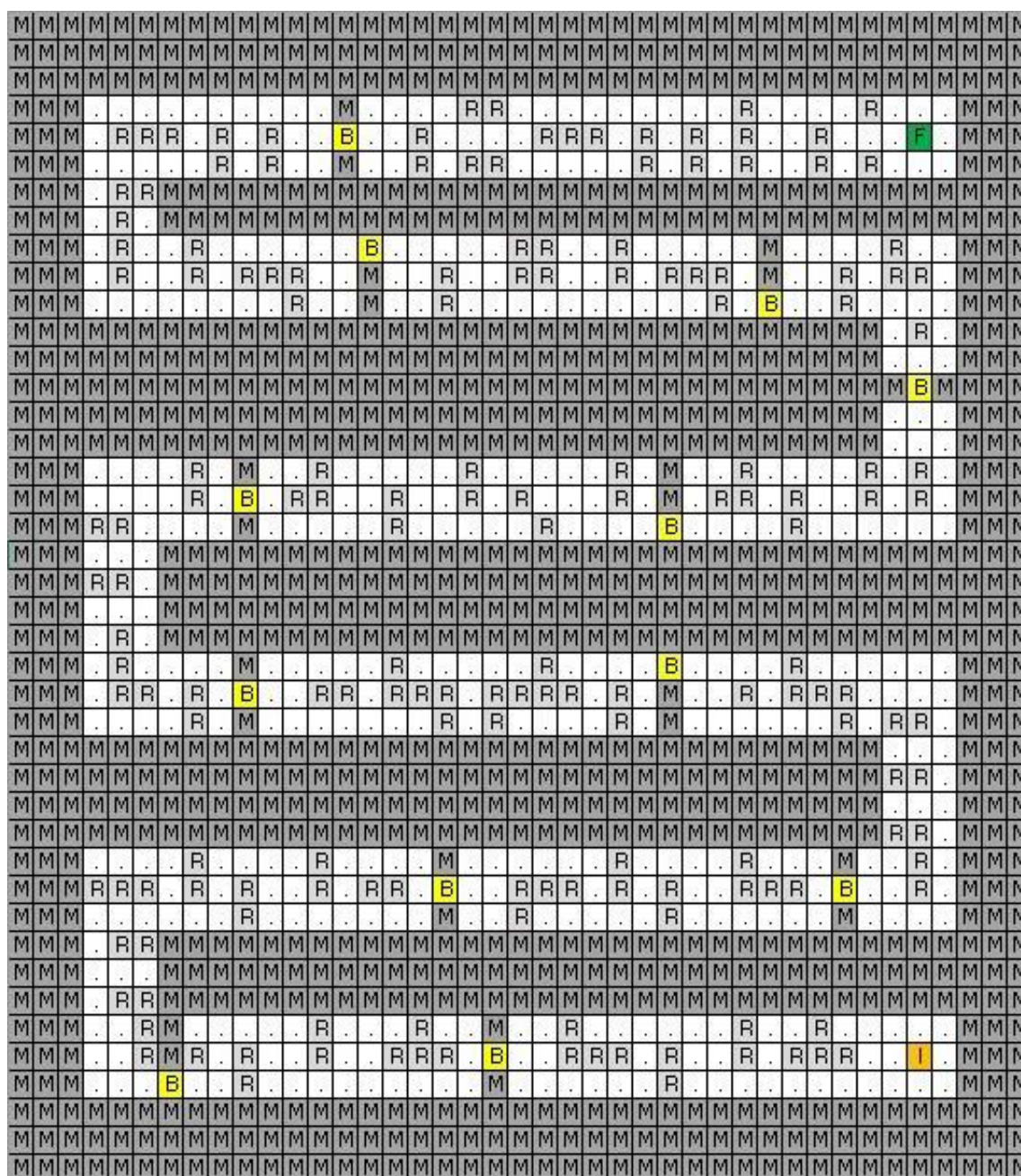


Figura 2. Mapa com as florestas e os ginásios pokemon.

Você inicia a sua jornada na **cidade de origem** (região em laranja no mapa) e termina ao chegar à **cidade destino** (região verde no mapa).

Ao passar por um ginásio, o agente deve **decidir quais Pokemons serão usados na batalha**. Cada **ginásio** apresenta um **nível de dificuldade** diferente de acordo com seus mestres (e devidos pokemons do ginásio). Este nível determina o tempo gasto por você

para poder vencer a batalha e avançar. A tabela abaixo mostra os níveis de dificuldade dos 12 ginásios que podem ser encontrados pelo caminho.

Ordem dos Ginásios	Dificuldade
1º Gym	55
2º Gym	60
3º Gym	65
4º Gym	70
5º Gym	75
6º Gym	80
7º Gym	85
8º Gym	90
9º Gym	95
10º Gym	100
11º Gym	110
12º Gym	120

Tabela 1. Níveis de dificuldade das 12 Bases Inimigas.

O **número de pokemons** participando das batalhas nos ginásios influencia no tempo gasto na batalha. Além disso, cada pokemon possui um determinado nível de **poder** que também influencia no tempo gasto nas batalhas. Quanto mais pokemons atacando, mais rápido o mestre do ginásio é derrotado. A tabela abaixo mostra o fator poder de cada pokemon.

Pokemon	Fator de Poder
Pikachu	1.5
Bulbassauo	1.4
Rattata	1.3
Caterpie	1.2
Weedle	1.1

Tabela 2. Poder dos Pokemons.

O tempo gasto nos ginásios é dado por:

$$Tempo = \frac{Dificuldade\ do\ ginásio}{\sum Poder\ dos\ pokemons\ que\ participam\ da\ batalha}$$

Além do fator de poder, cada pokemon possui **5 pontos de energia**.

Ao participar de uma batalha, cada pokemon perde -1 ponto de energia. Se o pokemon perder todos os pontos de energia, ele cai e fica em hibernação (não podendo participar de novas batalhas).

#### 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 cidade origem (região em vermelho no mapa).
- O agente sempre **termina** a sua jornada ao chegar à cidade destino (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 dos ginásios e o fator de poder** dos pokemons 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 **X** pessoas. (quantidade máxima será definida em aula)
- 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”;

### **Recomendações Finais:**

- (1) A interface gráfica não é o objetivo desse trabalho, mas auxilia na compreensão do resultado. Implementar uma “boa” interface gráfica (2D ou 3D) para representar o ambiente e o agente vale até 1 ponto extra.
- (2) O trabalho valerá 10 pontos divididos da seguinte forma: 8 pontos (para a busca e interface) + 2 pontos para quem conseguir encontrar a solução ótima (menor tempo calculado para alcançar o objetivo).

### **Observações Finais:**

O código fonte deverá ser entregue através de repositório do github. Mantenha o repositório privado e me convite como membro para que eu possa ter acesso.

A apresentação deve ser enviada como vídeo. Pode ser enviada como link do youtube, arquivo no repositório ou google drive. Todos os membros devem apresentar parte do trabalho. Faça a apresentação inicialmente explicando o código e o funcionamento dos algoritmos empregados e em seguida execute o programa para demonstrar o funcionamento do algoritmo. Utilize o zoom para gravar as telas e os membros do grupo – não é obrigatório ligar a câmera.