INF1771 - INTELIGÊNCIA ARTIFICIAL

TRABALHO 2 – LÓGICA

Descrição:

*"Para tornar-se um **Mestre Pokémon** é necessário aventurar-se por **locais desconhecidos** e capturar todos os **150 pokémons** que habitam o **Continente de Kanto**.*

*Para capturar os pokémons você precisa utilizar **Pokébolas**. Cada pokébola permite capturar apenas um pokémon. Se você ficar sem pokébolas, lembre-se de ir até uma **Loja Pokémon** para conseguir mais pokébolas.*

*Encontrar e capturar os 150 pokémons não é uma tarefa simples. Alguns pokémons são extremamente raros e **difíceis de serem encontrados**. Além disso, a qualquer momento você pode ser **desafiado por outro treinador** para uma batalha pokémon.*

*Após uma batalha pokémon, lembre-se sempre de ir a um **Centro Pokémon** para recuperar a energia dos seus pokémons."*

Figura 1. Lista dos 150 pokémons.

O Trabalho 2 consiste em implementar um agente baseado em conhecimento capaz de raciocinar nesse ambiente desconhecido. Você deve implementar uma interface em C/C++ ou Java para representar visualmente esse ambiente e utilizar a linguagem Prolog para representar o conhecimento do agente.

O mapa do continente é mostrado na Figura 2.

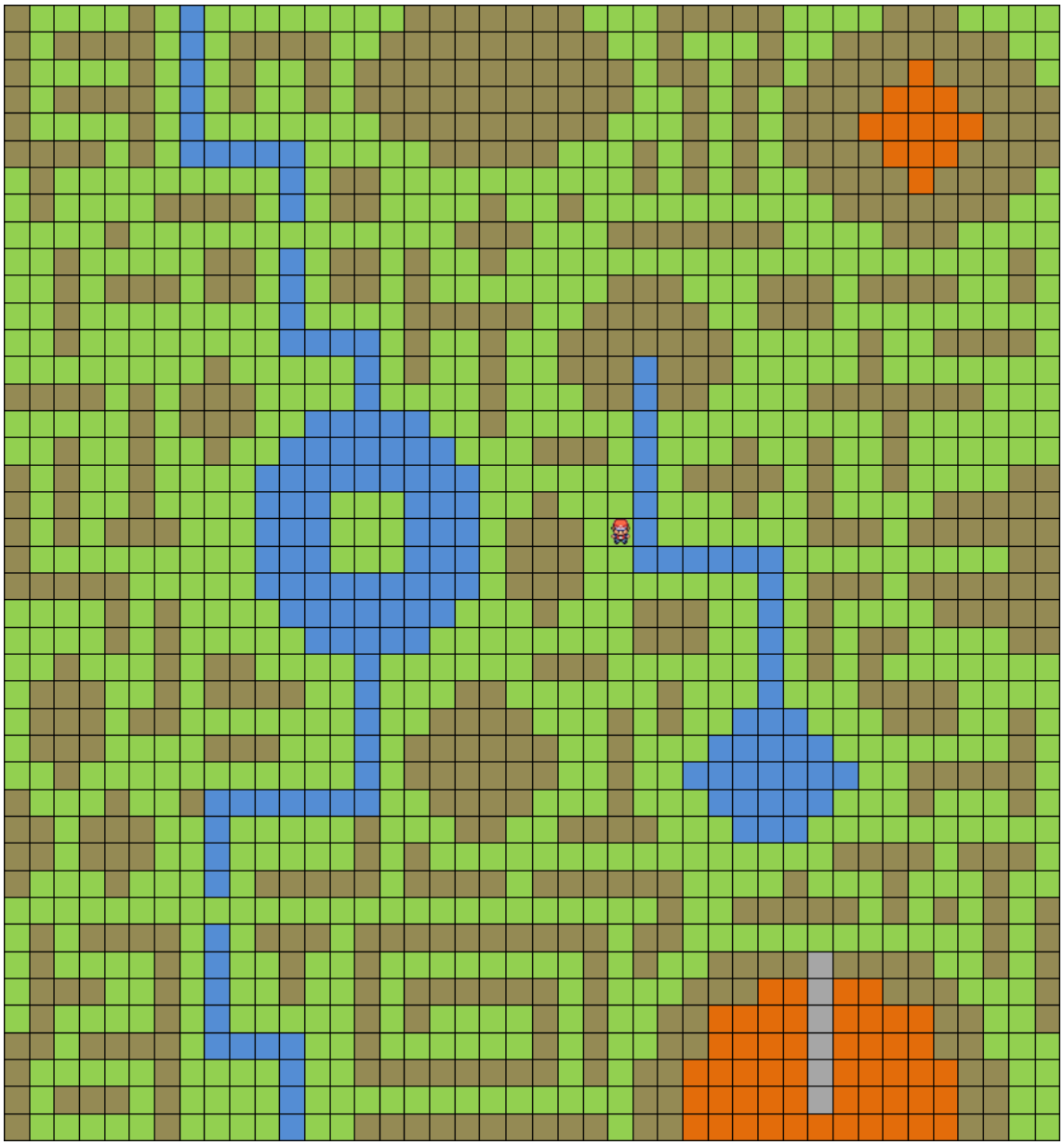

Figura 2. Mapa do continente de Kanto.

O continente é formado por 5 **tipos de terrenos**: grama (região verde), água (região azul), montanha (região marrom), caverna (região cinza) e vulcão (região laranja).

Inicialmente o agente pode passar livremente por terrenos de grama, mas somente pode passar pelos outros tipos de terrenos ao capturar determinados **tipos de pokémons**:

- **Pokémon de água** – pode passar livremente pela água;
- **Pokémon elétrico** – pode passar livremente pela caverna;
- **Pokémon voador** – pode passar livremente pela montanha;
- **Pokémon de fogo** – pode passar livremente pelo vulcão;

Para saber o tipo de cada um dos 150 pokémons você pode consultar a tabela disponível no seguinte endereço: http://goo.gl/6ifWYL

O agente conhece o mapa e o tipo do terreno de cada local, mas ele não conhece inicialmente a localização dos pokémons, centros, lojas e treinadores. O agente deve utilizar seus **sensores** para explorar o mapa.

Informações Adicionais:

- O mapa deve ser representado por uma matriz 42 x 42 (igual à mostrada na Figura 2).

- O agente sempre **inicia** a jornada no laboratório do Professor Carvalho (ponto onde está o personagem no mapa). A aventura **termina** quando o agente conseguir capturar os 150 pokémons.

- O agente pode executar as seguintes **ações**:
  - Mover para Frente;
  - Virar a Direita (rotação de 90°);
  - Virar a Esquerda (rotação de 90°);
  - Usar Pokébola – Para usar uma pokébola e capturar o pokémon que estiver no local onde o agente se encontra;
  - Pegar Pokébolas – Para pegar 25 novas pokébolas. A ação somente pode ser executada uma vez dentro de cada loja pokémon;
  - Recuperar Pokémons – Para recuperar a energia dos pokémons. A ação somente poder ser executada dentro de um centro pokémon.

- Cada ação executada pelo agente possui um **custo**:
  - Mover para Frente = -1;
  - Virar a Direita = -1;
  - Virar a Esquerda = -1;
  - Usar Pokébola = -5;
  - Pegar Pokébolas = -10;
  - Recuperar Pokémons = -100;
  - Derrotar um treinador pokémon = +150
  - Perder a batalha contra um treinador pokémon = -1000

- O agente sempre consegue **derrotar os outros treinadores** se os seus pokémons estiverem totalmente recuperados. Caso o agente entre em uma batalha com pokémons feridos, ele sempre **perderá a batalha**.

- Após uma batalha, os pokémons **sempre estarão feridos** e devem ser levados o mais rápido possível para um centro pokémon.

- O agente não tem acesso a nenhuma informação do mapa, mas ele possui alguns sensores para perceber o ambiente. O agente possui os seguintes sensores:
    - Em locais adjacentes a treinadores pokémon, exceto diagonal, o agente ouve os gritos do treinador desafiando o agente;
    - Em locais adjacentes a centros pokémon, exceto diagonal, o agente sente o perfume da enfermeira Joy.
    - Em locais adjacentes a lojas lokémon, exceto diagonal, o agente ouve o vendedor oferecendo pokébolas.
    - Em locais onde existe um pokémon a pokédex indica a presença e o número/nome do pokémon que está naquele local;

- O mapa tem a estrutura ilustrada na Figura 2. Mas é desconhecida a localização dos pokémons, centros, lojas e treinadores pokémon. Sabe-se apenas que existem:
    - 150 pokémons;
    - 20 centros pokémon;
    - 15 lojas pokémon;
    - 50 treinadores pokémon;

- As posições dos pokémons, centros, lojas e treinadores devem ser sorteadas aleatoriamente no inicio do programa. Mas o agente NÃO PODE ter acesso direto a essas informações.

- Os pokémons, centros, lojas e treinadores podem estar localizados em qualquer tipo de terreno, mas não pode existir mais de um elemento na mesma posição.

- Inicialmente o agente possui 25 pokébolas.

- O **jogo acaba** quando o agente conseguir capturar os 150 pokémons.

Requisitos:

- O programa deve ser implementado em C/C++ ou Java utilizando a biblioteca do SWI-Prolog que permite acessar diretamente o Prolog. Também é permitido utilizar outras linguagens, mas antes você deve verificar se ela é compatível com o SWI-Prolog. Exemplos:
    - C# (http://www.swi-prolog.org/contrib/CSharp.html)
    - Python (http://code.google.com/p/pyswip/)
    - PHP (http://www.j-paine.org/dobbs/prolog_from_php.html)

- O Prolog deve ser utilizado somente para **representar o conhecimento do agente,** a interface visual e demais controles devem ser implementados em C/C++ ou Java.

- Não é permitido realizar nenhum processo de tomada de decisão em C/C++ ou Java, a decisão de quais ações o agente vai realizar deve ser feita exclusivamente pelo Prolog.

- 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.

- **O mapa do planeta deve ser configurável**, 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.

- O programa deve exibir um log das consultas e inserções realizadas na base de conhecimento Prolog.

- O programa também deve exibir a pontuação do agente enquanto ele executa as ações. Assim como a pontuação final.

- O trabalho pode ser feito **individualmente** ou em **grupos** de no máximo 3 pessoas.

- **IMPORTANTE:** O programa deve ser apresentado durante a aula por **todos os membros do grupo**:

    - O membro do grupo que **não comparecer** receberá nota **zero**;

    - **Todos os membros** do grupo perderam **5.0 pontos** se alguém do grupo **não souber explicar** algo relacionado ao trabalho.

Dicas:

- Planeje e defina exatamente quais vão ser os predicados necessários no Prolog para codificar o conhecimento que o agente tem do mundo. Exemplos:
    - em(3, 3). - define a posição atual do agente;
    - pokemon(10, 6, 25). - identifica que existe um pokémon na posição (10, 6) e esse pokémon é número 25 (Pikachu);
    - centro_pokemon(10, 13). - identifica que existe um centro pokémon na posição (10, 13);
- Lembre-se de codificar predicados para identificar locais seguros e também locais visitados.
- A maneira mais simples de codificar a comunicação entre o Prolog e o C/C++ ou Java é definindo um predicado "*melhorAção*" no Prolog. Esse predicado deve retornar a melhor ação para ser executada naquele momento. Comece codificando os comportamentos mais simples, como por exemplo:
    - melhorAcao(usar_pokebola(P)) :- em(X,Y), pokemon(X, Y, P).
- A ação "*andar*" não necessariamente precisa ser para um local adjacente a posição do agente. Pode ser um "*andar*" para outro local (X, Y) ainda não visitado. Nesse caso, você pode executar o A* para calcular o melhor caminho para chegar até a posição (X, Y) passando por locais seguros, mas lembre-se de tomar cuidado com os outros treinadores e aplicar os custos de movimentação.

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;

Bônus:

(1) O agente que conseguir capturar os 150 pokémons com o menor custo, dado uma determinada configuração de pokémons, receberá 2 pontos extras na nota. Para participar dessa competição é necessário que o programa inclua uma forma simples de definir manualmente a posição dos pokémons, centros, lojas e treinadores. Em caso de empate, ambos os trabalhos receberão a nota extra.

Data de Entrega:

04/11

Forma de Entrega:

O programa deve ser apresentado na aula do dia 04/11 (segunda) e enviando até o mesmo dia para o email edirlei.slima@gmail.com.

Trabalhos entregues atrasados perderam 0.5 pontos para cada dia de atraso.