INF1771 - INTELIGÊNCIA ARTIFICIAL

TRABALHO 2 – LÓGICA

Descrição:

O segundo trabalho é baseado no exemplo do “Mundo do Wumpus”. O Mundo do Wumpus é caracterizado por um labirinto repleto de abismos e habitado pelo monstro Wumpus. O objetivo do jogo é sair vivo do labirinto com a maior quantidade de ouro possível. No interior da caverna, deve-se ficar muito atento às indicações de perigo, uma vez que o agente não conhece a localização dos obstáculos. Para identificar o perigo, o agente é dotado de percepções que o tornam capaz de sentir a brisa que sai dos abismos espalhados pela caverna ou ouvir o barulho dos morcegos gigantes, os quais são capazes de agarrar e transportar o agente para um ponto qualquer da caverna. Também é possível sentir o mal cheiro exalado pelo terrível Wumpus.

Cada grupo está responsável por definir o contexto, história, e etc seguindo as características abaixo. O grupo deverá utilizar os sprites de acordo com o contexto da história e criar o programa seguindo estritamente as características propostas.

O trabalho 2 consiste em implementar uma interface principal em C, C++, C#, Java ou qualquer linguagem com interface para o swi-prolog. A representação de conhecimento e as decisões deverão ser feitas através do **Prolog** e o programa apenas será utilizado para a interface gráfica do agente.

Características:

- O labirinto deve ser representado por uma matriz 12 x 12.
- O agente sempre inicia na posição [1, 1] do labirinto.
- A posição [1,1] também representa a saída do labirinto.
- Deverão ser implementados 3 tipos de inimigos: 1 com dano de 20, 1 com dano de 50 e 1 que teletransporta (o agente aparece em outro quadrado aleatório, podendo cair no buraco/obstáculo).
- O Agente terá 100 pontos de energia inicial.
- A munição do agente dá 20-50 de dano nos inimigos de forma aleatória.
- Os inimigos têm 100 pontos de energia inicial.
- O agente pode executar as seguintes ações:

    - Mover_para_frente;
    - Virar_a_direita (rotação de 90°);
    - Pegar_objeto – Para pegar o outro (se ele existir) na sala em que o agente se encontra;
    - Atirar – Para atirar a munição em linha reta na direção que o agente está olhando – A munição tem limite e só tem alcance na posição atual;
    - Subir – Para sair da caverna (a ação somente pode ser executada na sala [1,1]);
- Cada ação executada possui o custo de -1. Os demais eventos possuem os seguintes custos/recompensas:
    - Pegar = +1000;
    - Cair em um poço (obstáculo) = -1000;
    - Ser morto pelos inimigos = -1000;
    - Receber dano pelos inimigos = - valor do dano.
    - Atirar = -10;
- O agente possui os seguintes sensores:
    - Em salas adjacentes aos inimigos, exceto diagonal, o agente ouve um som de passos;
    - Em salas adjacentes a um poço/obstáculo, exceto diagonal, o agente sente uma brisa;
    - Em salas adjacentes ao inimigo que teletransporta, exceto diagonal, o agente percebe um flash;
    - Em salas onde existe ouro o agente percebe o brilho do ouro;
    - Ao caminhar contra uma parede o agente sente um impacto. As laterais do labirinto são paredes;
    - Quando o Inimigo morre o agente ouve um grito;
- O labirinto possui os seguintes elementos:
    - 2 Inimigos de dano 20;
    - 2 Inimigos de dano 50;
    - 8 Poços/Obstáculos;
    - 3 Pedras de ouro;
    - 4 Inimigos de teletransporte;

- A posição inicial dos elementos do labirinto deve ser sorteada aleatoriamente no início do programa.
- O agente não pode ter acesso às informações do mapa que foi gerado para o labirinto.
- O jogo acaba quando o agente sair do labirinto ou quando ele morrer por dano ou ao cair em um poço/obstáculo.
- Ao entrar em uma sala onde existe um teletransporter, o agente “acorda” em um lugar aleatório do labirinto, podendo ser um local seguro, um poço, a sala de um inimigo ou a sala onde existe outro teletranporter. Ou seja, o local onde o agente será teleportado deve ser sorteado.
- Nos slides das aulas anteriores existem vários exemplos do mundo de Wumpus, inclusive algumas regras em lógica de primeira ordem que podem ser traduzidas para Prolog.

Requisitos:

- O programa deve ser implementado em qualquer linguagem que possa utilizar a biblioteca do SWI-Prolog que permite acessar diretamente o Prolog.
- O Prolog deve ser utilizado somente para representar o conhecimento do agente, a interface visual e demais controles devem ser implementados na linguagem escolhida.
- 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 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.

Bônus:

- O trabalho que apresentar o agente com a melhor pontuação final em um determinado labirinto receberá 2 pontos extras na nota. Podendo tirar até 12 no trabalho. Em caso de empate ambos receberam a nota extra.

- Para isso é necessário que o trabalho tenha algum método para adicionar manualmente a matriz do labirinto.

Data de Entrega:

09/05 (segunda semana de maio)

Forma de Entrega:

O programa deve ser apresentado na aula do dia 09/05 (segunda) e o código deverá ser disponibilizado em repositório git.

Dúvidas?

Podem enviar as dúvidas para email abaffa@inf.puc-rio.br.