INF1771 - Inteligência Artificial
TRABALHO 2 – LÓGICA

Atenção: O segundo trabalho é opcional, cada aluno pode escolher se deseja fazer ou não este trabalho. Se o aluno entregar o trabalho até a data limite, sua nota obtida neste trabalho será utilizada no cálculo final da média dos trabalhos (N2). Para quem não entregar o trabalho, o cálculo da média dos trabalhos (N2) não incluirá essa nota.
Lembrando que, a média final do semestre (NF) é composta pela média dos trabalhos (N2) que possui peso 8 e pela média das listas de exercício (N1) que possui peso 2: NF = (N1 * 0,2) + (N2 *0,8).

Descrição do Trabalho:

O segundo trabalho é baseado no "Mundo do Wumpus", uma versão bem parecida com a que foi vista em sala de aula. O Mundo do Wumpus considerado neste trabalho 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 perceber a luz que vem do ouro. Também é possível sentir o mal cheiro exalado pelo terrível Wumpus.

Cada grupo de no máximo 2 pessoas está responsável por criar o programa seguindo as características propostas a seguir.

O trabalho 2 consiste em implementar uma interface principal em C, C++, C#, Java ou qualquer linguagem com interface para o swi-prolog para o Mundo do Wumpus. A representação de conhecimento e as decisões deverão ser feitas através do Prolog e o programa apenas será utilizado para a interface com o 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 2 tipos de inimigos: 1 tipo com dano de 20, 1 tipo com dano de 50.
- O Agente terá 100 pontos de energia inicial.
- A munição do agente dá 20-50 de dano nos inimigos de forma aleatória.
- A munição não possui recarga e permite 5 disparos.
- O agente pode executar as seguintes ações:
  - Mover_para_frente;
  - Virar_a_direita;
  - Virar_a_esquerda;
  - Pegar_objeto – Para pegar o ouro (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;
    - 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;
    - Receber dano pelos inimigos = - valor do dano.
    - Atirar = -10;
- O agente possui os seguintes sensores:
    - Em salas adjacentes aos inimigos, exceto diagonal, o agente sente um mal cheiro;
    - Em salas adjacentes a um poço, exceto diagonal, o agente sente uma brisa;
    - 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;
    - 3 Pedras de ouro;

- 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 após pegar os ouros ou quando ele morrer por dano.
- Nos slides das aulas 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). Será sua responsabilidade escolher uma linguagem que tenha essa interface com o SWI-Prolog.
- O Prolog deve ser utilizado somente para representar o conhecimento do agente e a tomada de decisão. 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 também deve exibir a pontuação do agente enquanto ele executa as ações. Assim como a pontuação final.

Observações:

- Faça um relatório com o que foi feito no trabalho, o relatório deve conter obrigatoriamente: Introdução, Metodologia (contendo os comandos prolog utilizados, dificuldades, descrição do que cada membro do grupo fez), Explicação de como rodar o programa, Conclusão.
- Deverá ser feita uma apresentação de no máximo 15 minutos explicando o que foi feito no trabalho. Todos os membros do trabalho devem falar. Prepare um arquivo para ser apresentado (estilo power point).
- O relatório e a apresentação do trabalho deverão ser enviados (juntamente com o código fonte) até a data limite de entrega, ambos devem ser arquivos pdf. Outros formatos não serão aceitos. E a não entrega implicará em perda de ponto.
- O trabalho pode ser feito em grupo de no máximo 2 pessoas.
- Data de Entrega: 20/11 23:59h. **Não serão aceitos trabalhos entregues fora do prazo.**
- Forma de Entrega: O código fonte, o relatório e a apresentação devem ser enviados para o e-mail rcapua@gmail.com até a data limite de entrega. O programa deve ser apresentado **impreterivelmente** na data que será marcada pelo professor (será no dia 21/11).
- Não enviem trabalhos após a data limite. Se a apresentação não puder ser feita nos dias marcados, o trabalho receberá nota zero que contará na média dos trabalhos (N2).
- Não copie nenhum material disponível na internet, qualquer programa que tenha cópia parcial ou total de outra fonte, independente do motivo, receberá nota zero. E a nota contará na média dos trabalhos (N2).
- Todos os membros do grupo devem participar da codificação e entender sobre a parte feita em prolog.