

# INF1771 - INTELIGÊNCIA ARTIFICIAL

## TRABALHO 2 – AGENTE LÓGICO – PITFALL!

*Nosso agente deve planejar o caminho do personagem “Pitfall Harry”, através de uma floresta tipo labirinto em uma tentativa de recuperar os tesouros o mais rápido possível. Nesse tempo ele deve derrotar a densa floresta, com inimigos (como cobras e insetos). Também deve ter muito cuidado com morcegos e urubus que podem carrega-lo para outro ponto na floresta. No caminho, Pitfall deverá superar inúmeros riscos, incluindo poços que podem se abrir num piscar de olhos. Harry deve evitar esse obstáculo pois morrerá instantaneamente quase o encontre.*

### 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.

O trabalho 2 consiste em implementar as regras para o planejador para o swi-prolog. A representação de conhecimento e as decisões deverão ser feitas através do Prolog que deverá exibir o plano e possivelmente uma representação da base de conhecimento na tela (através de uma representação do mapa).

### 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 (monstro pequeno), 1 com dano de 50 (monstro grande) e 1 que teletransporta (morcego, o agente aparece em outro quadrado aleatório, podendo cair no buraco/obstáculo).
- O Agente terá 100 pontos de energia inicial.
- Três powerups que carregam 20 de energia.
- O agente pode executar as seguintes ações:
  - Andar;
  - Virar\_a\_direita (rotação de 270°);
  - Virar\_a\_esquerda (rotação de 90°);
  - Pegar – Para pegar o outro (se ele existir) na sala em que o agente se encontra;
- 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.
- 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 teletransporter. 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 SWI-Prolog + Python (utilizar o código exemplo).
- O Prolog deve ser utilizado somente para representar o conhecimento do agente e a tomada de decisão. A interface visual, A\* e demais controles devem ser implementados em python (ou aproveitados do código exemplo).
- O programa também deve exibir a pontuação do agente enquanto ele executa as ações. Assim como a pontuação final. (já implementado na interface exemplo)
- Deve existir algum método para ler o arquivo com o mapa e adicionar manualmente a matriz do labirinto.

### **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:**

Muito do trabalho consiste em entender como representar os fatos, como definir o loop e como implementá-lo em Prolog. Tragam suas dúvidas e façam as perguntas para que eu possa ajuda-los durante o desenvolvimento