

INF1771 - INTELIGÊNCIA ARTIFICIAL

TRABALHO 2 – LÓGICA

Descrição:

“A sua cidade foi dominada por **zumbis** e você é o único sobrevivente! Mas os seus problemas estão apenas começando. Todas as saídas da cidade estão bloqueadas, a única forma de conseguir escapar é chegando até o único prédio da cidade que possui um heliporto com um **helicóptero** que você pode utilizar para sair em segurança da cidade. Porém, as ruas da cidade estão repletas de zumbis famintos. E para piorar ainda mais, um **denso nevoeiro** está sobre a cidade e você consegue enxergar poucos metros a sua frente. Por sorte os zumbis fazem bastante **barulho** e é possível reconhecer quando eles estão se aproximando. Você também tem uma **arma** que pode utilizar para matar os zumbis, mas cuidado para a sua **munição** não acabar. Se ficar sem munição, você pode tentar chegar a alguma das **delegacias abandonadas** que existem na cidade. Se você for ferido por um zumbi, você deve correr o mais rápido possível até um dos **hospitais** da cidade para pegar um **antídoto** antes que você também seja transformado em um zumbi. Lembre-se de levar com você a maior quantidade possível de antídotos, talvez o resto do mundo também tenha sido contaminado. Boa sorte!”

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1																				
2																				
3			😊																	
4																				
5																				
6										H										
7																				
8																				
9																				
10																				
11																				
12																				
13										M										
14																				
15																				
16																				
17																				
18																				
19																				
20																				

Figura 1. Mapa da Cidade.

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

Informações Adicionais:

- A cidade deve ser representada por uma matriz 20 x 20 (igual à mostrada na Figura 1).
- O agente sempre inicia a jornada na casa dele (posição [3, 3]).
- A posição [18,17] sempre representa o heliporto que permite que o agente escape em segurança da cidade.
- O agente pode executar as seguintes **ações**:
 - Mover_para_frente;
 - Virar_a_direita (rotação de 90°);
 - Pegar_objeto – Para pegar munição ou antidoto, se eles existirem no local onde o agente se encontra;
 - Atirar – Para dar um tiro em linha reta na direção que o agente está olhando;
 - Usar_Antídoto – Para usar um antidoto e ser curado.
 - Ligar_Helicóptero – Para ligar o helicóptero e sair da cidade (a ação somente pode ser executada na sala [18,17]);
- Cada ação executada pelo agente possui um **custo**:
 - Andar = -1;
 - Virar em uma direção = -1;
 - Pegar objeto = -1;
 - Atirar = -10;
 - Ligar o helicóptero = -1;
 - Ser mordido por um zumbi = -100;
 - Matar um zumbi com um tiro = +20;
- Se o agente for **mordido por um zumbi** ele ficará contaminado e o custo de todas as ações que ele executar enquanto estiver contaminado será multiplicado por 2. Se ele for mordido por dois zumbis, o custo das ações será multiplicado por 3. Toda vez que ele for mordido por mais zumbis o fator de multiplicação dos custos será aumentado em +1.
- Por sorte o agente sabe lutar. Toda vez que é atacado por um zumbi, ele consegue **matar o zumbi** que o atacou automaticamente. Mas ele sempre é mordido antes de conseguir deter o zumbi.
- Devido ao nevoeiro o agente não consegue ver o que existe a sua frente, mas ele possui alguns sensores para perceber o ambiente. O agente possui os seguintes sensores:
 - Em locais adjacentes a zumbis, exceto diagonal, o agente ouve os gemidos dos zumbis;
 - Em locais adjacentes a hospitais, exceto diagonal, o agente ouve as sirenes de ambulâncias que foram deixadas no local;
 - Em locais adjacentes a delegacias, exceto diagonal, o agente ouve as sirenes dos carros de policia que foram deixados no local;
 - Ao caminhar contra uma parede o agente sente um impacto;
 - Quando um zumbi morre o agente ouve um grito;

- O sensor que capta os gemidos dos zumbis também é capaz de detectar a intensidade de gemidos. Podendo identificar a quantidade de zumbis que estão próximos. A intensidade de gemidos é incrementada em +1 para cada zumbi que estiver nos locais adjacentes à posição do agente.
- A cidade tem a estrutura de ruas ilustrada na Figura 1. Mas é desconhecida a localização dos hospitais, delegacias e agrupamentos de zumbis. Sabe-se apenas que a cidade tem:
 - 4 Hospitais;
 - 4 Delegacias;
 - 60 Agrupamentos de Zumbis;
- A posição dos hospitais, delegacias e agrupamentos de zumbis deve ser sorteada aleatoriamente no início do programa. Mas o agente NÃO PODE ter acesso direto a essas informações.
- A localização dos hospitais e delegacias deve estar sempre dentro de um prédio e acessível da rua. Assim como ilustrado na Figura 1 (letra H e M).
- Os agrupamentos de zumbis devem sempre estar localizados nas ruas.
- Cada agrupamento de zumbis pode possuir uma quantidade de N de zumbis (entre 1 e 10). O número de zumbis em cada agrupamento também deve ser sorteado aleatoriamente na inicialização do programa.
- Cada delegacia possui uma quantidade de N munições (entre 30 e 100). O número de munições em cada delegacia deve ser sorteado aleatoriamente na inicialização do programa.
- Cada hospital possui uma quantidade de N antídotos (entre 0 e 10). O número de antídotos em cada hospital deve ser sorteado aleatoriamente na inicialização do programa.
- Inicialmente o agente possui 30 munições e 1 antídoto.
- Se o agente entrar em um local e encontrar um agrupamento de zumbis, ele é atacado por todos os zumbis que estiverem naquele local.
- O jogo acaba quando o agente conseguir sair da cidade ou quando ele for mordido consecutivamente por mais de 20 zumbis sem ter tomado nenhum antídoto.
- Ao conseguir sair da cidade, o agente ganha um bônus de +200 para cada antídoto que ele ainda estiver carregando.

Requisitos:

- O programa deve ser implementado em C ou C++ 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:
 - Java (<http://www.swi-prolog.org/packages/jpl/>)
 - 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++.
- Não é permitido realizar nenhum processo de tomada de decisão em C/C++, 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 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.

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;
 - `hospital(10,6).` - identifica que existe um hospital em (10, 6);
 - `delegacia(10,13).` - identifica que existe uma delegacia em (10, 13);
- A maneira mais simples de codificar a comunicação entre o Prolog e o C/C++ 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(ligar_helicoptero) :- em(18,17).`
 - `melhorAcao(pegar_objeto) :- em(X,Y), (hospital(X,Y) ; delegacia(X,Y)).`
- 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 zumbis e aplicar os custos de movimentação.

Bônus:

- O trabalho que apresentar o agente com a melhor pontuação final em uma determinada cidade 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 as posições dos grupos de zumbis, hospitais e delegacias.

Programa Base (Projeto do Visual Studio 2010):

O programa base fornecido é um exemplo de como realizar consultas ao Prolog a partir de um programa feito em C++. O exemplo pode ser acessado no seguinte link:

http://edirlei.3dgb.com.br/aulas/ia_2012_1/Trabalho2ProgramaBase.zip

Forma de Avaliação:

Será avaliado se o trabalho atendeu a todos os requisitos especificados anteriormente. O trabalho que atender a todos os requisitos receberá nota 10.

Data de Entrega:

29/10

Forma de Entrega:

O programa deve ser apresentado na aula do dia 29/10 (segunda) e enviando até o dia 29/10 para o email edirlei.slima@gmail.com.