Pontifícia Universidade Católica de Minas Gerais
Instituto de Ciências Exatas e Informática (ICEI)
Engenharia de Computação
Disciplina: Algoritmos e Estruturas de Dados I
Professor: Sandro Jerônimo de Almeida

Trabalho Prático Final – Agentes Inteligentes (20 Pontos)

Poderá ser feito individualmente ou em dupla

O trabalho prático final da disciplina consistirá na implementação de um sistema de simulação de agentes inteligentes utilizando os conceitos de Programação Orientada a Objetos. A ideia é criar um mundo virtual e agentes capazes de interagir nesse mundo.

A título de exemplificação, considere o agente WALLs. Esse agente deve se deslocar do ponto que foi criado até uma das paredes e começar um movimento de contorno das paredes, conforme ilustrado na figura 1.

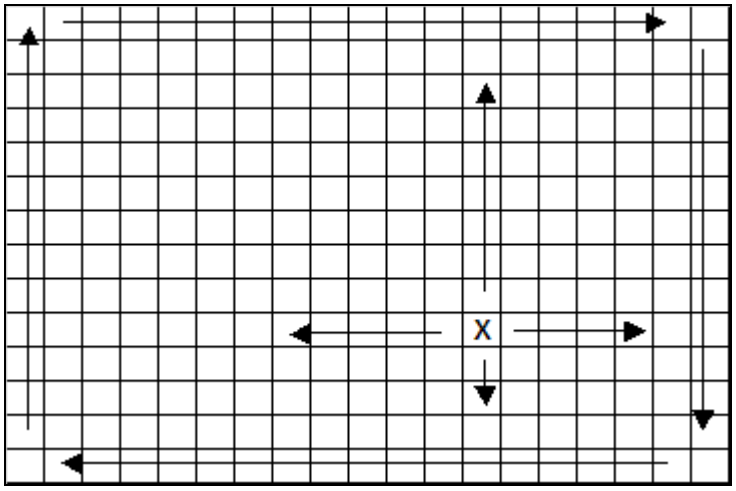

Figura 1: Deslocamento de uma agente um ambiente

Etapas

O trabalho é dividido em etapas, com entregas parciais, a saber:

1° Etapa – Classes e Objetos (16/06) – 5 pontos

Você deverá criar as classes do seu projeto, conforme sugerido a seguir:

Classe mundo: O mundo será uma matriz de agentes (MxN), onde cada posição deverá ser preenchida com NULL ou uma referência para um agente. Crie um método imprimir na classe Mundo, que será capaz de imprimir o conteúdo da matriz. Imprima 0 ou 1 para representar espaços (0) e agente (1). Será importante inicializar a matriz assim que o mundo for instanciado.

Classe Agente: classe abstrata que irá representar os agentes. Os agentes devem ter uma posição no mundo (x,y), uma variável de estado inteira (ex. 1 = andar, 2=correr) e uma referência para o mundo (ponteiro).

Classe Walls: essa classe representa o agente Walls (ilustrado anteriormente). Deverá herdar as propriedades e métodos da classe Agentes.

Crie uma função *main* para exemplificar a instanciação de objetos das classes acima criadas (Mundo, Agente e Walls).

Após a entrega desta 1° Parte uma implementação desta solução estará disponível para os alunos poderem fazer download, comparar com a própria implementação ou mesmo dar seguimento nas próximas etapas usando o código disponibilizado.

Além do código: tente diversificar seu ambiente de desenvolvimento aprendendo uma nova IDE. Sugere-se o uso do VS Code. O uso da ferramenta GitHub para gerenciar código também pode ser um excelente recurso para você trabalhar em equipe.

2° Etapa – Agentes e Comportamentos (23/06) – 10 pontos

Nesta etapa você deverá criar e dar vida aos seus agentes!

Classes Presa e Predados: crie um agente *Presa* e um agente *Predador*. O predador persegue a presa quando a percebe em um raio de visão de até três quadros de distância. O agente *Presa* foge sempre que percebe a presa em um raio de até dois quadros de distância. Os agentes deslocam-se aleatoriamente quando estão fora do raio de visão do adversário.

Métodos - os seus agentes deverão ter os seguintes métodos:
- Inicializar (recebe uma referência do mundo e uma posição inicial x,y).
- mover (nova posição x,y)
- decidir: decide o que irá fazer (escolha de estado baseado nas suas percepções)
- executar: executa um comportamento com base no estado definido anteriormente)

Main e Menu
- Crie um escalonador no *main* que, em loop, capaz de chamar os métodos decidir e executar dos agentes. Os seus agentes deverão se comportar conforme comportamento estabelecidos.

Além do código: apresente as regras de transição de estados dos seus agentes. Sugere-se a apresentação das regras por meio de uma máquina de estados finito (*Finite State Machine*) .

3° Etapa – Cenários e Simulação (28/06) – 5 pontos

O sistema deverá ter um menu para o usuário escolher qual sistema deseja simular (*Agente Walls* ou *Presa vs Predador*).

- O usuário deverá apertar alguma tecla para passar de uma iteração para próxima. A cada interação a tela deverá ser limpa e o cenário (matriz) exibido.

- O sistema deverá ter duas opções de inicialização: (i) de forma aleatória (ii) carregando cenário inicialmente configurado em um arquivo texto. Nesse caso, a primeira linha do arquivo deverá conter o número de linhas da matriz e o número de coluna. As demais linhas deverão ter um dos seguintes valores:

0 – Vazio (NULL)
1 – Agente Walls
2 – Agente Presa
3 – Agente Predador

Os dados no arquivo deverão ser separados por espaço. Veja o exemplo a seguir:

```
3 5
0 0 0 0 1
0 0 0 0 0
0 1 0 0 0
```

Cenario1-Walls.txt

```
4 5
0 0 2 0 0
0 0 0 0 1
0 0 0 3 0
0 0 0 0 0
```

Cenario2-PresaPredador.txt

- A qualquer instante o usuário poderá pedir para interromper a simulação e salvá-la (matriz em exibição) em um arquivo a ser informado pelo usuário.

Além do código: apresente uma documentação do seu código. Acrescente uma seção de “testes” onde você descreve experimentos realizados.