



CO023 - Programação Orientada a Objetos

Prof. Elder Cirilo

Trabalho 04

INFORMAÇÕES SOBRE ESTA ATIVIDADE:

- Deve ser entregue via Moodle um arquivo compactado chamado Trabalho-4-<nome_dos_alunos_no_grupo>.zip
- O arquivo deve conter apenas os arquivos .java, alguns exemplos de arquivos de entrada, um arquivo texto explicando como executar o seu programa, e um relatório explicando seu projeto (por exemplo comentando sobre os padrões de projeto utilizados) e um diagrama de classes UML contendo as classes e interfaces definidas, os relacionamentos entre elas, métodos, atributos, etc.
- Prazo final para entrega é **10/12/2014**. Serão marcadas apresentações dos grupos nos dias **11/12/2014** e **12/12/2014**.
- Além da apresentação final, cada grupo deve apresentar evoluções parciais do trabalho nas seguintes datas:
 - **06/11/2014**
 - **20/11/2014**
 - **04/12/2014**

*** A não apresentação de evoluções parciais do trabalho nas datas mencionadas acarretará em uma perda de 10 pontos na nota total do trabalho, por não apresentação.

- As notas dos membros do grupo podem ser diferenciadas, em função do conhecimento demonstrado sobre o trabalho durante as apresentações, final e parciais. Por isso, é fundamental que todos os membros do grupo participem ativamente de todas as apresentações. Será cobrada a presença de **todos** os membros do grupo nos dias das apresentações de acompanhamento e na apresentação final. Faltas nos dias de apresentações serão penalizadas com perda de 10 pontos na nota do trabalho.
 - O trabalho não será avaliado apenas se o código funciona. Também será considerada a qualidade do projeto, principalmente sob o ponto de vista do emprego de técnicas de programação orientada a objetos.
 - Cópias (total ou parcial) serão penalizadas com **nota zero**.
 - O trabalho pode ser realizado em grupo de **ATÉ 4 alunos**. Não será aceito trabalhos de grupos com mais de 4 alunos.
-

A ideia do Trabalho 04 é o desenvolvimento de um jogo onde ocorre uma disputa entre robôs submarinos para prospecção de petróleo. Os robôs submarinos são lançados ao mar, a partir de um helicóptero, sobre uma região onde se acredita haver petróleo. Esses robôs atingem o solo submarino em posições aleatórias. Duas ou mais companhias competidoras lançam robôs semelhantes para competir por poços na mesma região. Vence o jogo a companhia cujos robôs conseguirem produzir mais petróleo durante um período determinado.

Os robôs são equipados com diversos sensores que fornecem constantemente informações, tais como a posição do robô, a pressão subterrânea no local, o gradiente da pressão no local, etc. Além disso, cada robô pode perfurar o solo e iniciar a prospecção de petróleo em sua posição corrente. Sabe-se que a produção de petróleo (quantidade e qualidade) em um local é diretamente proporcional à pressão subterrânea no local. Por isso, os robôs devem encontrar os locais com maior potencial de produção e iniciar perfurações nesses locais. O time de robôs que obtiver medidas que determinem petróleo em maior quantidade ganha a disputa.

Terreno

O jogo se passa em um terreno que consiste de uma região retangular, sob a qual há uma distribuição de pressão. Os limites externos do terreno funcionam como paredes, isto é, os robôs não conseguem sair dos limites do terreno. O terreno é dividido em pequenas unidades quadradas de espaço, denominadas *células*. As células são identificadas por suas coordenadas cartesianas (x,y) , onde x e y são números naturais. A origem do terreno (a célula com coordenadas $(0,0)$) é a célula no canto inferior esquerdo do terreno. Cada célula possui um conjunto de propriedades associadas:

- pressão subterrânea: um valor dentro do intervalo $[0,1]$ e seu valor default é 0.

- faixa do coeficiente aleatório de erro de leitura do valor da pressão: dois valores representando os limites do intervalo para o coeficiente de erro, podendo representar qualquer intervalo contido em $[0, 0.1]$, se seu valor default é $[0.02, 0.02]$.
- profundidade: valor normalizado dentro do intervalo $[0, 1]$ e seu valor default é 1.

Em uma região submarina pode haver mais de um campo petrolífero. A quantidade de petróleo a ser explorado é diretamente proporcional à pressão de uma célula do terreno. Sabe-se que a distribuição de pressão em um campo petrolífero obedece a uma função suave, isto é, sem rupturas ou descontinuidades. Entretanto, devido à geografia do subsolo (existência de maciços rochosos, transições tectônicas e etc) a distribuição de pressão pode corresponder a somente uma porção limitada da função. Se uma célula do terreno for perfurada numa área próxima ao robô, com a posição perfurada no centro, não pode haver outra perfuração.

Deve ser definida uma fórmula que relacione a pressão de uma célula com o volume de petróleo que um robô consegue extrair nessa célula. Essa fórmula irá determinar quantos barris de petróleo um robô irá contabilizar para sua equipe quando fizer uma prospecção nessa célula.

Robôs

Os robôs devem ser capazes de receber comandos, os quais eles reagem apropriadamente. Cada robô deve possuir um Controlador, que é responsável por controlar suas ações, isto é, um Controlador é uma entidade encarregada de enviar comandos para um robô, de acordo com sua estratégia de jogo. Para simular uma situação mais real, os comandos devem possuir um custo de tempo para serem executados. Durante a execução de um comando, o controlador que enviou o comando fica parado esperando sua conclusão. Esse custo de tempo pode variar de acordo com o comando e com o contexto em que o comando deve ser executado. Por exemplo, um comando para movimentar um robô pode levar

mais ou menos tempo em função da diferença de profundidade entre as células de origem e destino. O conjunto de comando que um robô deve ser capaz de processar pode ser dividido em 4 subconjunto:

- comandos de sensoriamento: posição, pressão, gradientes, tempo;
- comandos de movimentação: anda, esquerda, direita;
- comando de prospecção: sonda;

Através dos comandos de sensoriamento, um controlador pode obter a posição (x,y) atual de seu robô (comando posição), assim como obter a pressão em sua posição (comando pressão) e os gradientes de pressão em sua região (comando gradientes). O comando gradientes informa o gradiente de pressão na direção da orientação de movimento do robô (diferença entre as pressões na célula a sua frente e na célula atrás dele) e o gradiente na direção normal ao seu movimento (diferença entre as pressões na célula a sua direita e na célula a sua esquerda). Utilize o intervalo do coeficiente de erro de leitura, definido na configuração do terreno, para introduzir erros aleatórios nas leituras de pressão. O comando tempo permite um controlador obter o tempo decorrido em segundos desde o início da partida.

Os comando de movimentação permitem deslocamentos apenas para células adjacentes à posição corrente do robô. Esses comandos são relativos à posição corrente do robô que, além de sua posição (x,y) no terreno, também indica em que direção o robô está apontando. O comandos anda faz com que o robô ande para frente. Os comandos direita e esquerda fazem com que o robô vire 90° para direita ou esquerda. Durante uma perfuração o robô não pode se movimentar.

O comando sonda perfura um poço na célula onde se encontra o robô. Como foi dito anteriormente, quando um robô começa a extrair petróleo em uma determinada célula, não é mais permitido começar uma perfuração nessa célula e nem nas suas células adjacentes. Isto é, outros robôs posicionados nessas células não poderão mais executar o comando sonda. O tempo do comando sonda deve ser diretamente proporcional a quantidade de petróleo disponível no poço que

está sendo prospectado assim como também da profundidade da célula perfurada.

Controlador

Cada robô possui um controlador, que é responsável por comandar suas ações no jogo. As associações entre controladores e robôs devem ser definidas em um arquivo de entrada que descreve as equipes que irão participar do jogo. O programa deve oferecer pelo menos controladores pré-programados, ou seja, controladores que possuem sua lógica de controle já programada diretamente em Java.

Um outro tipo de controlador interessante, mas que é opcional, e que conta 10 pontos extras, seria um controlador interpretado, que recebe sua lógica de controle em tempo de execução, descrita em alguma notação que será interpretada pelo controlador.

O Jogo e a Interface com o Usuário

Seu programa deve receber como entrada dois arquivos:

- um arquivo com a descrição do terreno submarino;
- um arquivo com a configuração do jogo, que inclui informações como
 - o tempo de duração da partida;
 - a descrição de cada equipe que irá disputar o jogo, como o nome da equipe, a cor de identificação da equipe, e para cada robô, um nome e o nome amplamente qualificado da classe que implementa o código do Controlador.

Os arquivos de entrada podem ser definidos utilizando as linguagens JSON ou XML. O programa deve suportar a leitura de ambos os formatos. O código dos controladores devem ser carregados dinamicamente durante a inicialização do jogo.

Ao iniciar o jogo, os robôs devem ser distribuídos aleatoriamente pelas células do terreno. Dois robôs nunca podem ocupar uma mesma célula em um mesmo instante do jogo. Os limites externos do terreno funcionam como paredes, isto é, os robôs não conseguem sair desses limites. Uma vez que todos os robôs estejam posicionados, estes dão início a suas estratégias de prospecção, que devem ser executadas concorrentemente até o término da partida.

Durante a execução da partida, as equipes competidoras devem poder acompanhar a evolução do volume de petróleo prospectado por elas e por robô. O usuário também deve ser capaz de consultar os valores das propriedades das células de terreno durante o jogo, assim como acompanhar a posição corrente de cada robô. Ao final da partida, deve ser exibido ao usuário o relatório final com as quantidades prospectadas por equipe e por robô.

Os robôs devem ser desenhados com a cor de sua equipe e devem exibir o nome, para que o usuário possa acompanhar a evolução dos robôs individualmente. A representação gráfica do robô deve indicar sua orientação, isto é, se ele está voltado para o norte, sul, leste ou oeste. Essa informação é importante uma vez que as operações de movimentação dos robôs são relativas à direção corrente como comentado anteriormente.