

Primeiro Trabalho de Programação Orientada a Objetos 2005.1

Prof. Renato Cerqueira

Objetivo

Neste semestre, serão desenvolvidos dois trabalhos de implementação complementares. A idéia é que no final do semestre tenha sido desenvolvido um jogo inspirado no sistema utilizado na disciplina Introdução à Engenharia do ciclo básico do CTC/PUC-Rio.¹ Esse jogo é 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 predeterminado.

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 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 desenvolvimento do jogo será dividido em duas partes:

- Trabalho 1
Desenvolver um editor gráfico para a construção do cenário (terreno submarino) do jogo.
- Trabalho 2
Desenvolver o jogo propriamente dito.

Este enunciado trata apenas da primeira parte do projeto.

Descrição

Nesta primeira etapa do projeto, deve-se desenvolver um editor de terrenos submarinos. Um terreno consiste de uma região retangular, sob a qual há uma distribuição de pressão. Os limites extremos 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 $[0, 0.1]$, e seu valor *default* é $[0.02, 0.02]$.

¹Mais detalhes em <http://www-di.inf.puc-rio.br/~hermann/introeng/introeng.htm> e <http://www.inf.puc-rio.br/~mascarenhas/robots/>.

- profundidade: valor normalizado dentro do intervalo $[0, 1]$ e seu valor *default* é 1.

Além dessas propriedades, o usuário do editor pode definir novas propriedades, informando seu nome e valor *default*. Propriedades definidas pelo usuário vão ser sempre do tipo *string* e, uma vez definida, todas as células passam a possuir essa propriedade.

Para começar a editar o terreno, o usuário deve primeiro definir as dimensões horizontal e vertical (número de células) do terreno e a dimensão n em *pixels* de cada célula (lembre-se que cada célula é quadrada: $n \times n$ pixels).

Após a definição das dimensões do terreno, o editor deve exibir um *mapa* do terreno, através do qual o usuário poderá definir os valores das propriedades das células. Ao definir o valor de uma propriedade de uma célula, o usuário poderá alternativamente fornecer um nome de uma classe Java que será usada pelo jogo para dinamicamente fornecer o valor da propriedade para uma dada célula (x, y) .

O editor deve permitir que o usuário forneça o valor de uma propriedade para várias células de uma única vez. Para isso, o editor deve oferecer um mecanismo para seleção múltipla de células.

É fortemente recomendável que o editor ofereça algum mecanismo de *feedback* para o usuário informando as células que já tiveram o valor *default* de suas propriedades alterado. A idéia é orientar o usuário sobre quais células já foram configuradas e quais talvez ainda precisem ser editadas.

A descrição do terreno deve ser salva em formato XML ou Lua. O editor deve ser capaz de ler um arquivo com a descrição de terreno para poder continuar sua edição. Os grupos que optarem por usar Lua podem utilizar a ferramenta LuaJava (<http://www.keplerproject.org/luajava/>).

Observações

- **Data de entrega:** 2 de maio.
- Trabalhos atrasados perdem 1 ponto por dia de atraso.
- O trabalho pode ser feito em grupo de até 3 participantes.
- Até o dia **8/abril** deve ser enviada uma mensagem para rcerq@inf.puc-rio.br informando os membros do grupo. Os alunos que não tiverem seu grupo informado até essa data perderão meio ponto na nota do trabalho.
- Na correção do trabalho, não será avaliado apenas se o programa funciona. Também serão considerados a qualidade do projeto, sob o ponto de vista de orientação a objetos, e aspectos de usabilidade do editor.
- Os grupos podem ser convocados para apresentar seus trabalhos.
- Cola, total ou parcial, implica em grau zero para todos os envolvidos.
- Os trabalhos devem ser enviados por email para rcerq@inf.puc-rio.br.
- O email deve conter apenas os arquivos `.java`, um arquivo texto explicando como compilar e executar o seu programa, e um relatório explicando seu projeto (classes e interfaces definidas, os relacionamentos entre os objetos, padrões de projeto utilizados, etc.). Opcionalmente, os arquivos podem ser enviados em um único arquivo `zip` ou `tgz`.
- Só considere o seu trabalho entregue após receber uma mensagem de confirmação minha.