

Game of Life

Pedro Yan Ornelas – 14/0158995

Pietro Bertarini – 14/0159118

Interface Gráfica:

A interface gráfica foi implementada usando Java Swing, uma biblioteca de interfaces gráficas do Java. Para melhor organização, foi criada uma classe GUI que continha todos os elementos relacionado à interface, como JPanels, JButtonse Jframes.

Visando um melhor aproveitamento do que a biblioteca tem a oferecer, o tabuleiro foi portado completamente para a interface, fazendo-se uso de uma abordagem que facilita ao usuário a geração de novas células, onde, ao invés de passar uma posição x e y, basta clicar em algum ponto da matriz que uma célula irá nascer nas coordenadas clicadas.

Para tal implementação, foi criada uma matriz de botões que se comunicam com a engine, que faz o processamento lógico do jogo. Ao clicar num botão no tabuleiro, o status booleano [Alive](#) é invertido, ou seja, se a célula está morta ela fica viva e se a célula estiver viva ela morre. Ao clicar no botão "Próxima geração", o tabuleiro de botões é sincronizado com o tabuleiro lógico, que calcula a situação de todas e células e devolve ao tabuleiro gráfico as posições das células vivas, que são plotadas para o usuário continuar interagindo dinamicamente com o tabuleiro.

Quanto à geração automática, ao invés do loop ser feito de uma maneira recursiva (seguindo o padrão implementado na versão original do projeto) foi necessário fazer uma abordagem mais iterativa, evitando Stack Overflow causado por uma chamada excessiva de funções recursivas. Foi usada uma classe chamada Timer do Java swing para fazer repetição do cálculo das próximas gerações, onde a cada X milissegundos o tabuleiro é atualizado novamente com as novas informações

Injeção de dependência:

A injeção de dependência foi feita em duas partes do programa, a primeira na "main" mostrando como é feita a injeção de dependência direta do XML usando a regra "Conway", a segunda em forma de lista criando um botão que trará todas as opções que contem na lista contida no XML e em seguida ele realiza a injeção de dependência no programa dependendo da opção do cliente.

Primeiramente foi necessário utilizar a interface "EstrategiaDeDerivacao" (encontrada em BR.unb.cic.lp.gol) para que se faça a injeção utilizando essa interface, pois assim podemos criar uma lista de interfaces que será utilizada futuramente. Em seguida foi utilizada a função "XmlBeanFactory" para ler o arquivo XML, onde ele instanciará todas as classes e nele terá o endereço de onde se encontra todas as regras no programa para que possa fazer a injeção de dependência. Assim criamos "beans" para se relacionar com determinado arquivo e serão

esses beans que nos iremos injetar no programa. Então, utilizando o método `getBean` do `facotry` que criamos utilizando `"XmlBeanFactory"` nos podemos escolher qual regra nos queremos injetar no programa. Para finalizar utilizamos a função `"setEstrategia"` utilizando o parâmetro o bean `"Conway"`, ou seja, nos injetaremos o bean `"Conway"` que referencia a regra `"Conway"` encontrada no `.jar` das regras.

Em seguida foi realizada a injeção de dependência utilizando listas, para isso foi criada uma classe chamada `"RegrasList"` (`BR.unb.cic.lp.gol.estrategias`) que fará uma lista de interfaces de `"EstrategiaDeDerivacao"`. Assim conseguimos criar uma lista com as regras do `GameOfLife`, pois todas são implementadas a partir desta interface. Em seguida, dentro do XML criamos um bean chamado `regraslist`, neste bean nos iremos instanciar nele uma lista de beans que se referenciam as regras do programa. Ou seja, as regras que estarão nessa lista serão as regras que irão entrar no programa. Utilizando o `Java Swing`, nos conseguimos mostrar a lista de regras em um botão chamado `Regras`, neste botão conterá uma lista de `string` com o nome das regras que foi obtida utilizando o método `"GetName"` de `"EstrategiaDeDerivacao"`. Após o cliente escolher a regra, ele injetará no programa a função escolhida utilizando a função `"getBean"` e `string` escolhida, assim podemos mudar a regra do programa com em execução quando quisermos.

Compilação

Para compilar o programa, importa as seguintes bibliotecas:

-spring-framework-2.5.5-with-dependencies;

-commons-logging-1.2-bin;

-regrasframework.jar(Encontrado dentro do `GameOfLife`);

(Lembre-se de referenciar o `.jar` ao projeto principal e exportar corretamente o `.jar` ao projeto principal)

-Execute pela `main`;

obs: Se não rodar, procure por `Pietro Bertarini` ou `Pedro Yan` pois o programa foi executado com sucesso em estilo `framework`;