

Projeto Detalhado de Software

Universidade Federal de Goiás (UFG)

Kleudson Rodrigues
Wallace Luis do Carmo

Relatório - Roda a Roda

Este documento é responsável por registrar as atividades realizadas pelos responsáveis pelo projeto a fim de mensurar e detalhar o trabalho realizado pela equipe na construção do software Roda a Roda. O documento seguirá os requisitos do sistema estabelecidos a seguir pelo diretor do projeto.

Requisitos do sistema - Liberações

L1: Ter uma versão do jogo capaz de:

1. Escolher quantidade de jogadores: 1, 2 ou 3;
2. Escolher a quantidade de palavras por etapa: 1, 2 ou 3;
3. Escolher a quantidade de etapas: 1, 2, 3, 4, 5, 6 ou 7;
4. Funcionar para um jogador, com uma palavra por etapa, com uma etapa somente.
Para este caso o tema é animais e a palavra é cachorro.
5. Rodar a roleta. Não é necessário interface gráfica. A roleta tem 20 divisões, sendo:
 - a. duas divisões de perde tudo,
 - b. duas de passa a vez,
 - c. quatro divisões de 100 pontos,
 - d. quatro divisões de 200 pontos,
 - e. quatro divisões de 400 pontos,
 - f. duas divisões de 500 pontos,
 - g. duas divisões de 1000 pontos.
6. Informar se jogador é vencedor ou perdedor.

L2: Ter uma versão do jogo capaz de:

1. Funcionar para dois jogadores, com a quantidade de etapas selecionada e com a quantidade de palavras por etapa selecionada.
2. O tema disponível continua sendo animais, mas há pelos menos vinte e uma palavras (a sua escolha), que são lidas de um arquivo de texto. O jogo deve realizar um sorteio das palavras selecionadas por jogo. No futuro, deverá existir um arquivo para cada tema disponível.

L3: Ter uma versão do jogo capaz de:

1. Funcionar para três jogadores, com a quantidade de etapas selecionada e com a quantidade de palavras por etapa selecionada.
2. Funcionar com os seguintes temas: profissão, cidade e filme. Sendo um arquivo de texto para cada tema.

3. Usar diversas estratégias de sorteio da roleta. O compromisso inicial é fazer o sorteio da roleta usando Math.random, mas no futuro outras estratégias poderão surgir.

L4: A liberação 4 não é uma liberação de fato, mas os resultados produzidos até aqui são testados com o suporte do JUnit.

L5: O sorteio da roleta deve utilizar o padrão de projeto strategy. Duas estratégias são conhecidas: uma aleatória e outra viciada. Na viciada, sempre na segunda vez que o jogador girar, o resultado deve ser passe a vez. O padrão de projeto observer deve ser utilizado para tratar ações do jogo. Por exemplo, após escolher uma letra, observadores podem ser utilizados para avaliar se a letra é faz parte da palavra e se o jogo terminou.

	Data	Descrição	Info. Adicional
Kleudson Rodrigues Wallace Luis do Carmo	17 de agosto de 2015 - 23 de agosto de 2015	<ul style="list-style-type: none">• inicialização do projeto;• reunião referente aos requisitos;	Foi estabelecido o primeiro contato com o problema a ser resolvido pela equipe
Kleudson Rodrigues Wallace Luis do Carmo	09 de novembro de 2015 - 15 de novembro de 2015	<ul style="list-style-type: none">• início do desenvolvimento em java, relacionado a primeira liberação;	Liberação L1 foi apresentada a equipe
Kleudson Rodrigues Wallace Luis do Carmo	25 de Janeiro de 2016	<ul style="list-style-type: none">• Mudança na forma de construir o projeto (projeto construído por etapas);	Inicialmente o projeto estava sendo desenvolvido com uma visão fechada a solução do problema como um todo. A solução é construir o software por etapas, assim como as liberações são distribuídas
Kleudson Rodrigues	25 de Janeiro de 2016	<ul style="list-style-type: none">• Criação do repositório no GitHub;	Gerenciamento de projeto
Kleudson Rodrigues Wallace Luis do	05 de Fevereiro de 2016	<ul style="list-style-type: none">• Comit de implementação;• Comit do primeiro	

Carmo		documento de diagrama de classe	
Kleudson Rodrigues Wallace Luis do Carmo	07 de Fevereiro de 2016	<ul style="list-style-type: none"> • L1 • L2 	Conclusão das liberações 1 e 2.
Kleudson Rodrigues Wallace Luis do Carmo	22 de Fevereiro de 2016	<ul style="list-style-type: none"> • L3 	Jogo jogável com até 3 jogadores.
Kleudson Rodrigues Wallace Luis do Carmo	25 de Fevereiro de 2016	<ul style="list-style-type: none"> • L5 	Implementação preparada para entrega.

Relatório de Construção

Aqui estarão descritas todas as estratégias utilizadas para a construção do software Roda a Roda, tendo como base o documento UML e o próprio código fonte. Um documento de apoio foi anexado logo abaixo deste relatório.

Segundo os requisitos do sistema do projeto Roda a Roda, restrições relacionadas ao jogo devem ser seguidas para que o próprio possa realizar aquilo que foi proposto corretamente. Tendo tais regras como base, pôde ser observado pela equipe que a classe 'Roleta' seria uma das preocupações em relação a modelagem do sistema.

A roleta do jogo possui 20 possibilidades de jogadas, onde 4 delas são de azar e 16 de sorte, sendo assim os jogadores terão as mesmas oportunidades de jogadas. No entanto, o requisito aponta uma possibilidade de haver uma ou mais possibilidades de rolagem, ou seja, a roleta poderá girar de maneiras diversas. Para que o sistema possa trabalhar com vários tipos de rolagem o Strategy é o padrão de projeto escolhido para lidar com tal situação, devido a facilidade pela qual permiti modificar vários tipos de rolagem sem que uma interfira na outra, através do uso de uma interface. Desse forma foi criado a interface <<Giro>> e duas formas de rolagem, 'roletaViciada' e 'roletaAleatoria' conforme o documento de apoio logo abaixo.

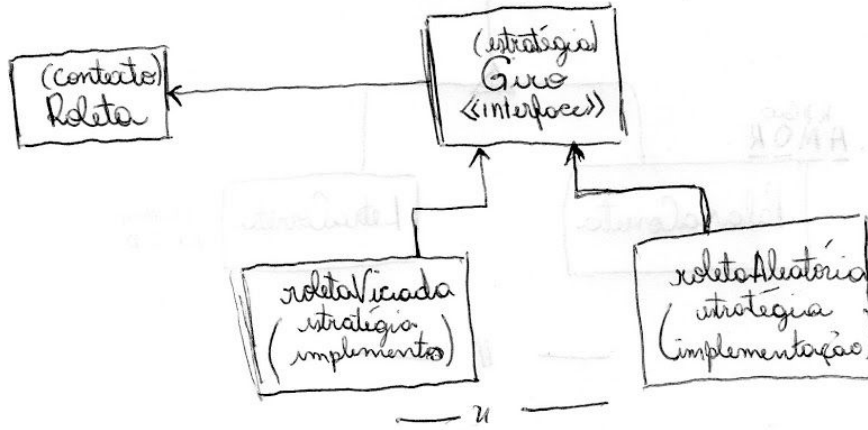
As chamadas "ações do jogo", deverão ser implementadas usando o padrão de projeto Observer, de modo que, a roleta sempre estará responsável por informar (atualizar), as informações do jogo que está sendo jogado, para que possa ser verificado o status do jogo, em fim, definir fim do jogo ou não. Seguindo a padrão Observer, duas classes estarão presentes na modelagem, sendo elas a classe 'Roleta', responsável pela definição dos valores sorteados por cada jogador e atualizar ("Update()"), a classe StatusJogo da situação do jogo. Esta funcionalidade tem o auxílio de duas interfaces, <<roletaInterface>> e <<interfaceAtualizada>>.

O padrão de projeto Iterator também foi utilizado na construção do jogo. Vetores foram encapsulados em coleções através de uma interface integradora.

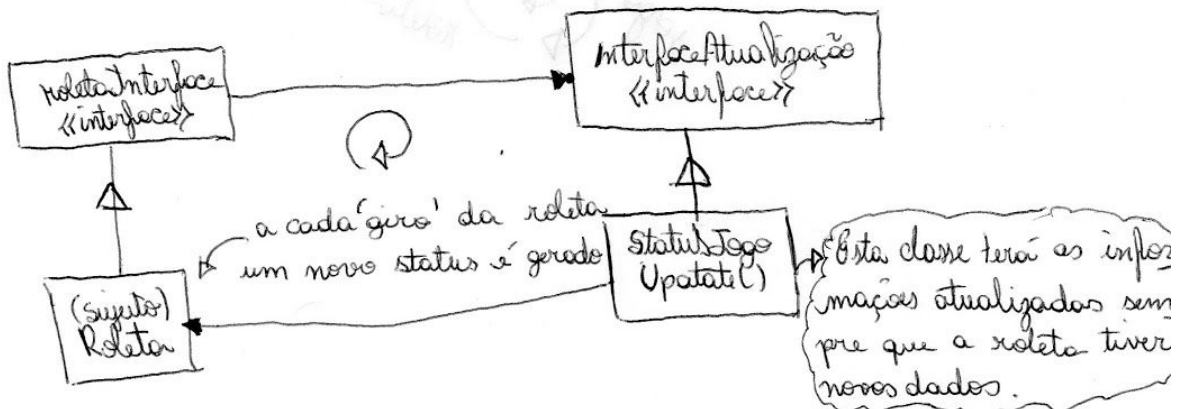
①

Sorteios da Roleta (strategy)

- 100, 200, 400, 500, 1000, Perdeu a Vez, Perdeu Tudo
 - 100, 200, 400, 500, 1000, Perdeu a Vez, Perdeu Tudo
 - 100, 200, 400
 - 100, 200, 400
- } 20 opções na roleta



Ações do Jogo (observer)



Classes:

- **Roleta***
- **Parametros*** (início de cada partida)
- **Jogadores***
- **Main***
- **StatusJogada** (observer)

Interfaces:

- **Gira*** (Comportamento da roleta)
- **StatusRoleta***
- **AtualizarJogo***

2)

Formando Palavras / Obtendo combinações de ^{letras} ~~palavra~~
(Decorator)

