

Projeto Jaipur

Especificação de Requisitos de Software

Versão 3.0

02/07/2017

Versão	Autor	Data	Ação
1.0	Douglas Soares Molina	30/04/2017	Estabelecimento dos requisitos
2.0	Douglas Soares Molina	22/05/2017	Mudanças requisitos não-funcionais
3.0	Douglas Soares Molina	02/07/2017	Mudanças regras fim de turno

Conteúdo:

1. Introdução;
2. Visão Geral;
3. Requisitos de software;
4. Esboço da Interface Gráfica;

1. Introdução

Objetivo: Desenvolvimento de um jogo (programa distribuído) que possibilite dois usuários jogarem entre si o jogo de cartas Jaipur (inventado pelo autor do projeto).

O Jogo: Jaipur é um jogo sobre trocas e vendas. O objetivo é se tornar o vendedor particular de Mahajara's ficando mais rico que seu oponente no final de cada semana (round). Para isso, deve-se coletar e negociar bens (cartas) através do mercado, e vendê-las por rupias (fichas de pontuação). Se o jogador conseguir fazer um bom negócio (três cartas ou mais), então ele receberá fichas extras de pontuação. Porém, o relógio não está a favor dos jogadores. Quanto mais os negócios movimentam, menor o retorno do mercado de trocas.

Regras Gerais:

- O jogo se baseia em 55(cinquenta e cinco) cartas de bens com tipos diferentes, dentre eles: Diamante, Ouro, Prata, Pano, Comida e Couro. Além desses tipos, existem as cartas do tipo Camelo.
- Cada tipo de bem possui um valor específico.
- O mercado tem um limite de 5(cinco) cartas e começa obrigatoriamente com 3(três) cartas do tipo Camelo e 2(duas) cartas tiradas aleatoriamente (podem também ser cartas Camelo).
- O mercado possui um número limitado de rupias por tipo de bem, sendo que cada carta vendida por jogadores precisa necessariamente ser trocada por uma rupia do seu tipo específico.
- Cada jogador começa com 5(cinco) cartas.
- No começo do jogo, todas as cartas do tipo Camelo de cada jogador é colocada na sua pilha de Rebanho específica (essa pilha é onde ficam todas as cartas de Camelo).
- As demais cartas são colocadas em uma pilha de compra geral do jogo.
- O jogador poderá executar apenas uma ação por turno.

Opções de Ação (existem apenas dois tipos):

- Pegar Cartas
 - Se o jogador escolher pegar cartas, ele deve escolher **apenas uma** das seguintes opções:
 - Pegar **todas as cartas de bens** do mercado (as cartas podem ser de tipos diferentes), em troca do **mesmo número** de cartas de bens do jogador. As cartas retornadas podem ser bens da mão do jogador, cartas do Rebanho, ou uma combinação das duas.
 - Pegar apenas **uma carta**, repondo o mercado pela primeira carta do topo da pilha de compra geral do jogo.
 - Pegar **todas as cartas do tipo Camelo**, repondo o mercado por compras sucessivas do topo da pilha de compra geral do jogo até completar o limite de cartas do mercado.

Atenção: Os jogadores nunca devem ter mais do que 7(sete) cartas em suas mãos até o fim do turno. As cartas em excesso podem ser descartadas em uma pilha separada de descarte.

- Vender Cartas
 - Para vender cartas, o jogador deve escolher um tipo de bem e descartar(vender) quantas cartas achar necessário na pilha de descartes. Cada carta vendida implica em uma ficha(rupia) de tipo específica, e se a compra for boa o suficiente, um bônus.
 - Como realizar uma venda:
 1. Selecione quantas cartas desejar, colocando-as viradas para cima na pilha de descarte.
 2. Retire o mesmo número de cartas descartadas em fichas(rúpias), começando pelo topo da pilha de fichas daquele tipo (começando pelas fichas de maior valor).
 3. Se você vender 3(três) cartas ou mais, retire o bônus correspondente.
 - **BÔNUS:** O valor do bônus não é conhecido até ser retirado pelo jogador (demonstrado na ficha). Os valores de bônus variam:
 - 3 cartas vendidas: Retorno de 1(uma), 2(duas) ou 3(três) rupias.
 - 4 cartas vendidas: Retorno de 4(quatro), 5(cinco) ou 6(seis) rupias.
 - 5 cartas vendidas: Retorno de 8(oito), 9(nove) ou 10(dez) rupias.

Restrição de vendas: Ao vender um dos três bens mais valiosos (diamantes, ouro e prata), a venda deve incluir um mínimo de 2(duas) cartas. *(A regra se aplica mesmo se houver apenas uma ficha do tipo específico.)*

Atenção: Os jogadores só podem vender 1(um) tipo de bem por turno, nunca mais.

Fim do turno:

- Um turno acaba imediatamente se:
 - 3(três) tipos de fichas de bens são esgotados no mercado.
 - Não existem cartas suficientes na pilha de compra ao repor o mercado.

Pontuação:

- O jogador com mais cartas do tipo Camelo no seu Rebanho recebe 5(cinco) rupias bônus.
- Os jogadores somam o valor total de rupias que possuem para decidir quem é o mais rico.
- O jogador mais rico receberá 1(um) ponto.
- Em caso de empate, o jogador com o maior número de **fichas de bens** vence.

Novo turno:

- Se nenhum dos jogadores possuírem ainda 2(dois) pontos, o jogo é restabelecido do começo e um novo turno começa. O jogador que perdeu o turno passado começa jogando.

Restrições finais:

- Quando retirando cartas do mercado, o jogador recebe cartas de bens **OU** cartas Camelo, mas **nunca** os dois.
- Se o jogador decidir retirar cartas do tipo Camelo, este deve retirar **todas** as cartas Camelo do mercado.
- Quando realizando uma troca:
 - As cartas descartadas podem ser qualquer tipo de bem, Camelos ou uma mistura dos dois.
 - Os mesmos tipos de bens não podem ao mesmo tempo ser descartadas e retiradas no mercado.
 - O jogador **nunca** pode trocar apenas 1(uma) carta de sua mão contra 1(uma) carta do mercado. Uma troca deve sempre envolver **pelo menos** 2(duas) cartas por 2(duas) cartas.
- Se os dois jogadores possuírem o mesmo número de cartas Camelo no final de um turno, nenhum deles recebe o bônus de 5(cinco) rupias.
- Em algumas ocasiões, há um número menor de fichas disponíveis do que cartas em uma venda. Nesse caso, o jogador ainda recebe a ficha bônus pelo número de cartas vendidas.
- As cartas Camelo não contabilizam para o limite de 7(sete) cartas da mão.

2. Visão Geral

Arquitetura do programa: programa orientado a objetos, sistema distribuído do tipo cliente-servidor.

Premissas de desenvolvimento:

- O programa deve ser implementado em Java, devendo executar em qualquer plataforma que disponha de máquina virtual Java.
- O programa deve apresentar interface gráfica bidimensional.
- O programa deve executar distribuído ao NetGamesNRT.

3. Requisitos de Software

3.1 Requisitos Funcionais:

- *Requisito funcional 1 – Conectar:* O software deve apresentar a opção de estabelecer uma conexão com o servidor NetGames.
- *Requisito funcional 2 – Desconectar:* O software deve permitir o usuário se desconectar do servidor, encontrando uma possível partida em fase de início.
- *Requisito funcional 3 – Iniciar partida:* O software deve apresentar a opção de iniciar uma nova partida, realizando a identificação dos jogadores.
- *Requisito funcional 4 – Configurar jogo:* O software deve organizar e distribuir as cartas onde for necessário seguindo as regras de aplicação do jogo.

- *Requisito funcional 5* – Realizar jogada: O software deve demonstrar o jogador da vez e o permitir que utilize o mouse (com cliques do botão esquerdo) para interação com a interface em meios de realizar as ações definidas pelas regras do jogo.
- *Requisito funcional 6* – Atualização do estado: Para cada ação de um jogador, a aplicação deve orientar e demonstrar claramente o novo estado da partida. Deve ser indicado de alguma forma se a vez passou para outro jogador ou se o turno foi finalizado.

3.2 Requisitos Não Funcionais:

- *Requisito não funcional 1* – Especificação de projeto: Código realizado em linguagem Java e especificação de projeto baseado em UML.
- *Requisito não funcional 2* – Interface gráfica para o usuário: A aplicação deverá ter uma interface gráfica única, compartilhada entre os jogadores.
- *Requisito não funcional 3* – Interface gráfica baseada em Java, biblioteca gráfica Swing.

4. Esboço da Interface Gráfica

