

Universidade Federal de Santa Catarina
Centro Tecnológico CTC

Departamento de Informática e Estatística INE
Curso de Sistemas de Informação
Disciplina: Análise e Projeto de Sistemas (INE5608)

Prof.: Ricardo Pereira e Silva
Maurício F. Galimberti

Alunos: Francisco Sacco Flores Almeida Teixeira (13200997)
Irenio Lima Jesus de Aragão (13101283)
Julia Baldissera (13201009)

JOGO DE CARTAS SABONETE

Versão	Autor	Data	Ação
1.0	Francisco Teixeira, Irenio Aragao, Julia Baldissera.	13/04/2015	Estabelecimento dos requisitos
1.0.1	Francisco Teixeira, Irenio Aragao, Julia Baldissera.	27/05/2015	Ajustes de requisitos
1.0.2	Francisco Teixeira, Irenio Aragao, Julia Baldissera.	12/07/2015	Ajustes de requisitos e melhoria na clareza das regras do jogo

Conteúdo

1. Introdução
2. Visão Geral
3. Requisitos de Software

4. Esboço Interface Gráfica
5. Referências

1. Introdução

O objetivo do projeto é desenvolver um jogo virtual Java em rede para interação de usuários, através da necessidade de criar uma sequência de cartas, previamente desordenadas, contando com apenas um baralho de cartas.

2. Visão Geral

2.1. Arquitetura do Software

- **Paradigma de Programação:** Programa Orientado a Objetos;
- **Tipo aplicação:** distribuída, suportando arquitetura cliente-servidor;

2.2. Premissas de desenvolvimento

- **Desenvolvimento:** Implementação em Java;
- **Modelagem:** Especificação de projeto baseada em UML 2 com Visual Paradigm;
- **Conectividade:** A aplicação deve utilizar a ferramenta NetGamesNRT;
- **Interface Gráfica:** deve obrigatoriamente ter uma interface;

2.3. Definições

- **Monte:** Pilha de cartas embaralhadas voltadas para baixo e alinhadas.
- **Lixo:** Pilha de cartas descartadas voltadas para cima.
- **Mesa:** 13 cartas inicialmente voltadas para baixo. Cada jogador tem sua mesa.
- **Carta Revelada:** É a carta da mesa voltada para cima e posicionada em ordem crescente, de acordo com seu número.
- **Carta Oculta:** É a carta da mesa voltada para baixo.
- **Turno:** Intervalo em que cada jogador faz sua(s) jogada(s) de acordo com as regras do jogo.

2.4. Regras do Jogo

- Número de jogadores: 2.
- São distribuídas 13 cartas retiradas do monte, formando a mesa de cada jogador.
- O primeiro jogador que solicitar o início da partida deve iniciar jogando.

- Em cada turno o jogador busca uma carta do monte ou do lixo, fazendo quantas jogadas forem possíveis.
- Em cada jogada, o jogador observa a carta de sua mão e verifica:
 - 1) Se a posição da mesa, respectiva ao número da carta, não estiver revelada, o jogador troca a carta oculta pela carta de sua mão. A carta oculta passa a ser a carta de sua mão, e o jogador faz uma nova jogada;
 - 2) Se a posição da mesa, respectiva ao número da carta, estiver revelada, o jogador descarta a carta no lixo, finalizando seu turno.
- É vencedor aquele que conseguir revelar todas as cartas de sua mesa primeiro, formando a sequência de cartas A, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K.

3. Requisitos de software

3.1. Requisitos Funcionais

- **Conectar:** O software deve apresentar em seu menu a opção “Conectar”, para assim estabelecer conexão com o servidor.
- **Iniciar Partida:** O programa deve apresentar a opção “Iniciar Partida”, que começa uma nova partida somente se a conexão já estiver estabelecida.
- **Preparar partida:** O programa deve, após iniciar partida, embaralhar e distribuir as cartas, retirando 13 cartas do monte embaralhado, para formar a mesa de cada um dos jogadores (cartas inicialmente voltadas para baixo);
- **Proceder jogada:** O Software deve apresentar de quem é a vez do turno, e permitir que apenas este usuário clique no monte ou no lixo para pegar uma carta. Em seguida, deve permitir que o mesmo faça substituições válidas da carta da mesa pela da mão, clicando na carta oculta respectiva. Também deve permitir que o jogador descarte a carta da mão, clicando no lixo;
- **Receber jogada:** O Software deve identificar o término do turno do jogador adversário, atualizando, então, na interface gráfica do jogador atual, as cartas do adversário que estão voltadas para cima, a existência de cartas no monte, a última carta adicionada ao lixo e o status do jogo;
- **Verificar jogada:** O software deve verificar se a carta da mesa selecionada pelo Jogador esta oculta. Também deve verificar se o numero da carta na mão do jogador é igual a posição da

carta da mesa. Caso todas as verificações sejam válidas, o programa deve trocar a carta oculta pela carta de sua mão. A carta oculta passa a ser a carta da mão do jogador;

- **Verificar estado da partida:** A aplicação deve, a cada final de turno, verificar se as mesas dos jogadores estão com as cartas reveladas na sequência crescente. Anuncia o vencedor ao identificar uma mesa totalmente revelada e ordenada;
- **Desconectar:** O programa deve apresentar em seu menu a opção “Desconectar”, para assim finalizar a conexão com o servidor, podendo finalizar a partida.

3.2. Requisitos Não Funcionais

- **Especificações do projeto:** Codificação deve ser feita em Java e especificação de projeto baseada em UML 2;
- **Padronização código:** Utilizar lowerCamelCase (iniciados por letras minúsculas) na definição de variáveis e métodos, e utilizar UpperCamelCase (iniciados por letras maiúsculas) na definição de Classes.
- **Interface Gráfica:** A interface gráfica deve ser intuitiva, única e compartilhada para ambos os jogadores, deixando expostas as cartas reveladas das mesas;
- **Portabilidade:** O programa deve funcionar em qualquer plataforma que disponha da máquina virtual Java, versão 8 ou superior;
- **Internacionalização:** O jogo deve estar disponível em Português.

4. Esboço Interface Gráfica (Fig. 4.1)

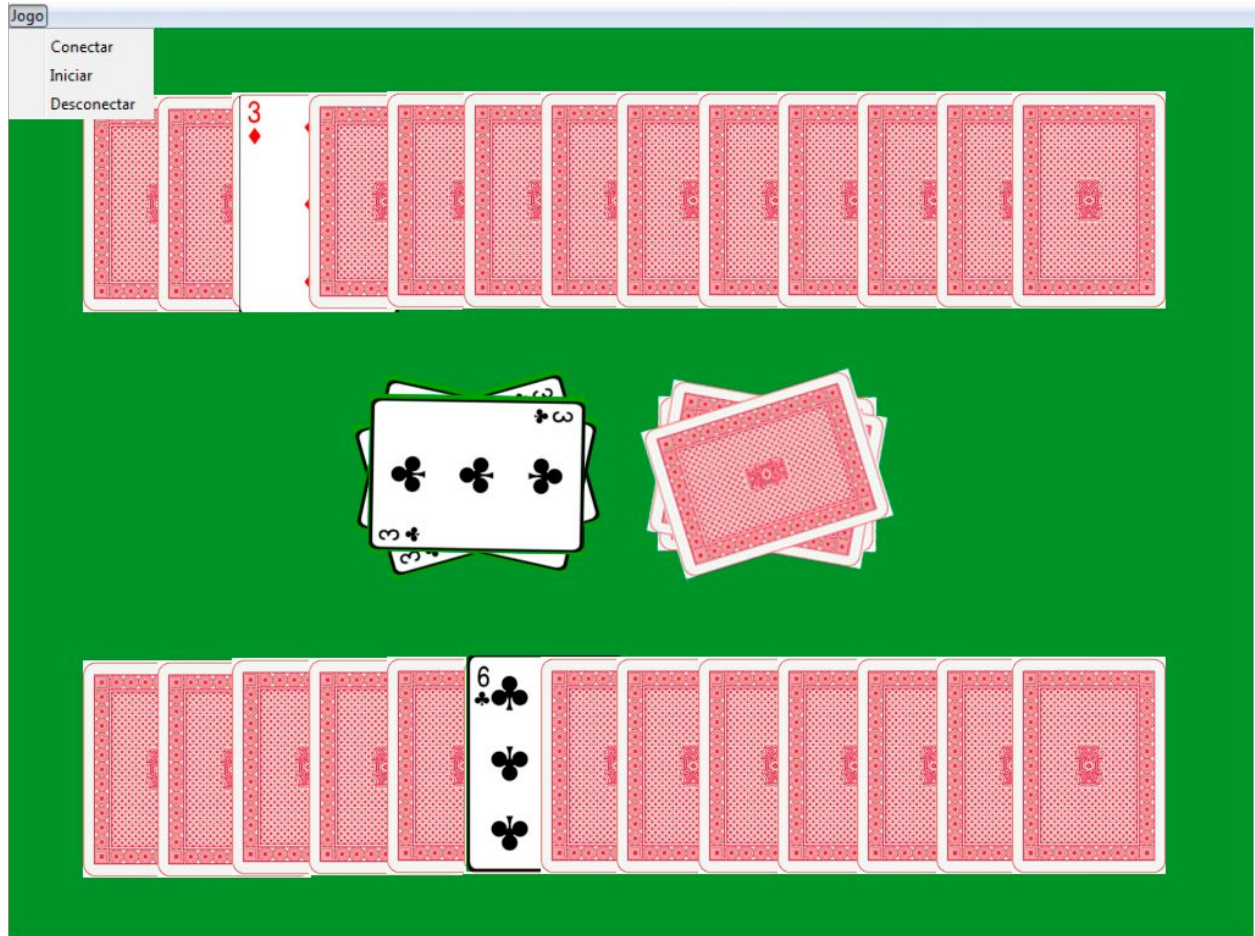


Fig. 4.1

5. Referências

Jogo Sabonete - <https://verinhaalfabetizacao.wordpress.com/2010/04/30/jogo-sabonete/>
Baralho - <http://pt.wikipedia.org/wiki/Baralho>
NetGamesNRT - <http://www.inf.ufsc.br/~netgames/NetGamesNRT/>
UML 2 em Modelagem Orientada a Objetos - <http://www.inf.ufsc.br/~ricardo/uml2moo/>
Como Modelar com UML 2 - <http://www.inf.ufsc.br/~ricardo/cmuml2/>