



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA

Especificação de Requisitos de Software Bridge

Marcelo Pietro Grützmacher Contin
Wesly Carmesini Ataíde

Florianópolis
2020

| Versão | Autor(es) | Data | Ação |
|--------|--------------------------------|------------|--|
| 1.0 | Marcelo Contin Wesly Ataide | 18/09/2020 | Estabelecimento de Requisitos |
| 1.1 | Marcelo Contin Wesly Ataide | 25/10/2020 | Ajustes após refinamento de casos de uso |

Sumário

1. Introdução
2. Visão Geral do Sistema
3. Requisitos da Aplicação

Introdução

Objetivo do Desenvolvimento

Esta aplicação visa fornecer uma plataforma digital e distribuída para o jogo Bridge, a princípio funcional quando houver 4 jogadores conectados.

Para a realização deste software serão utilizados conhecimentos adquiridos em disciplinas anteriores a fim de desenvolver toda a aplicação de código e, também, aprendizados relacionados à aplicação da UML (*Unified Modeling Language*) obtidos na disciplina de Engenharia de Software I, resultando na aplicação final do jogo e todo o material de documentação relacionado.

Referências (regras do jogo):

<https://www.megajogos.com.br/bridge-online/regras>

Visão Geral do Sistema

Arquitetura de Software

A aplicação terá a arquitetura cliente-servidor distribuído.

Premissas do Desenvolvimento

- O programa será desenvolvido na linguagem de programação orientada a objetos Java.
- Deverá ser executado em qualquer plataforma que possua a máquina virtual Java versão.
- O suporte a execução distribuída deve ser fornecido pelo framework NetGames.
- A documentação UML será desenvolvida na plataforma Visual Paradigm.

Requisitos da Aplicação

Requisitos Funcionais

- Conectar
Os jogadores devem possuir a opção de se conectarem ao jogo através do servidor, caso não estejam conectados. Essa ação é necessária para iniciar uma partida e deve ser executada antes de qualquer ação no jogo.
- Desconectar
Os jogadores devem poder se desconectarem do jogo, caso estejam conectados. Isso deve estar disponível durante o jogo. No caso de desconexão, a partida deve ser abortada.
- Iniciar partida
Os jogadores devem possuir a capacidade de iniciar uma partida quando 4 jogadores estiverem conectados. Através da interface gráfica, um jogador deve determinar o início da partida, o que irá ser refletido nas interfaces com os outros jogadores.
- Receber a determinação de início
O programa precisa estar pronto para receber a determinação de início do servidor, fazendo as instâncias apropriadas quando recebê-la e preparando o início do jogo.
- Escolha de dupla
Os jogadores terão a opção de escolher sua dupla ao iniciar uma partida, antes de prosseguir para o leilão. Essa escolha importa para os estágios mais avançados. O jogador que iniciar a partida poderá escolher sua dupla, definindo assim as duplas da partida.
- Receber escolha de dupla
A escolha do primeiro jogador deve aparecer para o primeiro jogador e ser replicada para os outros jogadores, através do servidor, logo após a escolha.
- Fazer aposta
Os jogadores deverão realizar as apostas para definir o contrato da partida, sempre antes de uma sessão de descarte. Cada jogador, na sua vez, deve poder escolher entre as opções definidas nas regras de uma aposta válida, até que três jogadores passem consecutivamente.
- Receber aposta
A escolha de cada jogador de aposta deve aparecer para ele e para os outros jogadores, através do servidor, logo após a escolha.
- Descarte
Os jogadores poderão escolher uma carta disponível da sua mão para jogar na Vaza, com o objetivo de ganhar uma vaza para sua dupla. Essa sessão de descarte vem logo após o leilão e deve se repetir até todas as cartas dos jogadores serem usadas. O declarante também deve poder decidir a jogada do Morto.
- Receber descarte
O descarte de um jogador precisa aparecer para ele mesmo e para todos os jogadores, através do servidor, após a escolha de carta.
- Verificar ganhador da vaza
Quando uma vaza estiver terminada, é necessário saber qual o vencedor com uma checagem no programa. De acordo com esse vencedor, guardamos a informação de número de vazas, o que definirá pontuações e vitórias.

- Definir ordem dos jogadores
O programa precisa preservar a ordem dos jogadores na mesa, logo, quando definimos quem começa em um descarte, precisamos fazer uma checagem mantendo essa ordem.
- Definir apostas possíveis
Antes de qualquer aposta, é necessário saber quais as apostas que podem ser feitas. O programa deve implementar esse comportamento habilitando ou desabilitando as apostas de acordo com as regras.
- Definir cartas jogáveis
Antes de um descarte, é necessário saber quais as cartas que podem ser jogadas na vaza. O programa deve implementar esse comportamento habilitando ou desabilitando as cartas de acordo com as regras.
- Cálculo de pontuação
O programa precisa calcular e disponibilizar automaticamente o placar dos jogadores, calculado de acordo com as regras. O placar deve ser atualizado a cada término do turno das vazas. O cálculo pode ser feito a cada receber descarte. Foi optado por uma versão simplificada de pontuação do jogo.
- Receber ação
O programa deverá receber 3 tipos de ações: descarte, aposta e escolher dupla. Ações as quais são necessárias serem recebidas para atualizar as interfaces de cada jogador e dar continuidade ao jogo.

Requisitos Não-funcionais

- Versão do Java
O programa será desenvolvido na linguagem Java versão 11.
- Biblioteca gráfica escolhida
A biblioteca gráfica escolhida é a Swing, devido a facilidade da extensão Window Builder do Eclipse.
- Suporte ao software distribuído
O suporte a execução distribuída deve ser fornecido pelo framework NetGames.
- Modelagem UML
A documentação em UML 2 será desenvolvida na plataforma Visual Paradigm.