

Projeto Briscola

Especificação de Requisitos de Software

Versão 3.0

20/06/2024

Versão	Autor(es)	Data	Ação
1.0	Hugo Matos de Carvalho Eric Alex Cardoso Friess Francisco Bortolanza	31/03/2024	Estabelecimento dos requisitos
2.0	Hugo Matos de Carvalho Eric Alex Cardoso Friess Francisco Bortolanza	11/05/2024	Criação do requisito funcional 2 “conectar”, criação do requisito não funcional 3 e alteração do requisito funcional 3 “iniciar jogo”
3.0	Hugo Matos de Carvalho Eric Alex Cardoso Friess Francisco Bortolanza	20/06/2024	Alteração do requisito não funcional 5

Conteúdo:

- 1. Introdução
- 2. Visão geral
- 3. Requisitos de software

1.Introdução

1.1 Objetivo

Desenvolvimento de um programa distribuído que suporte a disputa de partidas de Briscola na modalidade usuário contra usuário.

1.2 Definições, abreviaturas

Regras do jogo: ver referências

1.3 Referências:

Regras do Jogo:

<https://pt.wikipedia.org/wiki/Briscola>

2.Visão Geral

2.1 Arquitetura do programa

Cliente-servidor distribuído.

2.2 Premissas de desenvolvimento

- O programa deve ser implementado em Python;
- O programa deve usar DOG como suporte para execução distribuída;
- Além do código, deve ser produzida especificação de projeto baseada em UML, segunda versão.

3.Requisitos de Software

3.1 Requisitos Funcionais:

Requisito funcional 1 – Iniciar programa: ao ser executado, o programa deve apresentar na interface o menu inicial do jogo (com o nome do jogo “Briscola”, botão “CONECTAR” e espaço de digitação habilitado para receber o nome do jogador);

Requisito funcional 2 – Conectar: o programa deve apresentar a opção de menu “CONECTAR” para se conectar com DOG Server . Em primeiro lugar, o programa deve verificar se há algo escrito no espaço de digitação do menu. Caso não haja nada escrito, o usuário deve ser avisado da necessidade de escolher um nome. Após a escolha do nome, o programa deve solicitar conexão (utilizando os recursos de DOG), registrando o que está escrito no espaço de digitação como nome do jogador. O resultado da tentativa de conexão deve ser informado ao usuário. No caso de conexão mal sucedida, o jogador permanecerá no menu inicial, podendo tentar novamente ao apertar o botão “CONECTAR”. No caso de sucesso, o programa deve desabilitar a mudança de nome na interface e o botão “CONECTAR” deve ter seu nome alterado para “JOGAR” e sua funcionalidade alterada para iniciar uma partida;

Requisito funcional 3 – Iniciar jogo: após o usuário estar conectado a DOG Server, o programa deve apresentar a opção de menu “JOGAR” para o início de uma nova partida. O procedimento de início de partida consiste em enviar uma solicitação de início a Dog Server, que retornará o resultado, que será a identificação e a ordem dos jogadores, em caso de êxito, ou a razão da impossibilidade de início de partida, caso contrário. Em caso de sucesso, a interface do menu inicial deve ser atualizada para a interface de jogo do programa com as informações recebidas, cada jogador deve receber três cartas aleatórias do baralho em sua mão e a carta do topo do baralho deve ser baixada como a briscola. Caso o jogador local seja quem inicia a partida, a interface deve estar habilitada para seu procedimento de lance;

Requisito funcional 4 – Jogar carta: O programa deve permitir a um jogador habilitado selecionar uma carta presente em sua área de cartas. A carta selecionada deve ser movida para o centro da tela. Se a ação for executada após o jogador adversário já ter jogado uma carta naquela rodada, devem ser verificados os critérios do lance (ver regras do jogo) para saber qual dos jogadores vai vencer o lance e pegar as cartas da mesa. Caso o outro jogador ainda não tenha tido sua vez, ele estará habilitado a fazer seu lance, o jogador local deve ser desabilitado e o programa deve aguardar a jogada do adversário (ver Requisito funcional 6) ou de notificação de abandono (ver Requisito funcional 7). Após o jogador local realizar seu lance, o programa deve enviar a jogada ao adversário (utilizando os recursos de DOG). A jogada enviada deve conter o naipe da carta e seu valor. Após ambos os jogadores realizarem seus descartes, o jogador que pegar as cartas da mesa terá a pontuação destas cartas adicionadas à sua pontuação total da rodada. Em seguida, o programa deve verificar se a rodada chegou ao fim. No caso de não encerramento, o programa deve adicionar uma carta à mão de cada jogador se ainda houverem cartas no baralho e prosseguir para o próximo lance. No caso de encerramento da rodada, deve ser verificado se o vencedor chegou a três rodadas vencidas na partida, encerrando a partida se essa condição for verdadeira ou mostrando o vencedor da rodada na tela e iniciando uma nova rodada caso seja

falsa. No caso de encerramento da partida, deve ser notificado o nome do jogador vencedor e placar final;

Requisito funcional 5 – Receber determinação de início: o programa deve poder receber uma notificação de início de partida, originada em Dog Server, em função de solicitação de início de partida por parte de outro jogador conectado ao servidor. O procedimento a partir do recebimento da notificação de início é o mesmo descrito no ‘Requisito funcional 3 – Iniciar jogo’, isto é, a interface do menu inicial deve ser atualizada para a interface de jogo do programa com as informações recebidas, cada jogador deve receber três cartas aleatórias do baralho em sua mão e a carta do topo do baralho deve ser baixada como a briscola. Caso o jogador local seja quem inicia a partida, a interface deve estar habilitada para seu procedimento de lance;

Requisito funcional 6 – Receber jogada: o programa deve poder receber uma jogada do adversário, enviada por Dog Server, quando for a vez do adversário do jogador local. A jogada recebida deve conter as informações especificadas para o envio de jogada no ‘Requisito funcional 4 – Jogar Carta’. O programa deve remover a carta da área de cartas do adversário (parte superior da tela) e colocá-la no centro da tela. Após isso, deve seguir a lógica apresentada no ‘Requisito funcional 4 – Jogar Carta’: verificar se o jogador local ainda não fez seu lance, e em caso positivo, habilitar este jogador; em caso negativo, deve se verificar qual jogador vai pegar as cartas da mesa e avaliar se a rodada chegou ao fim, se a partida chegou ao fim e proceder de acordo;

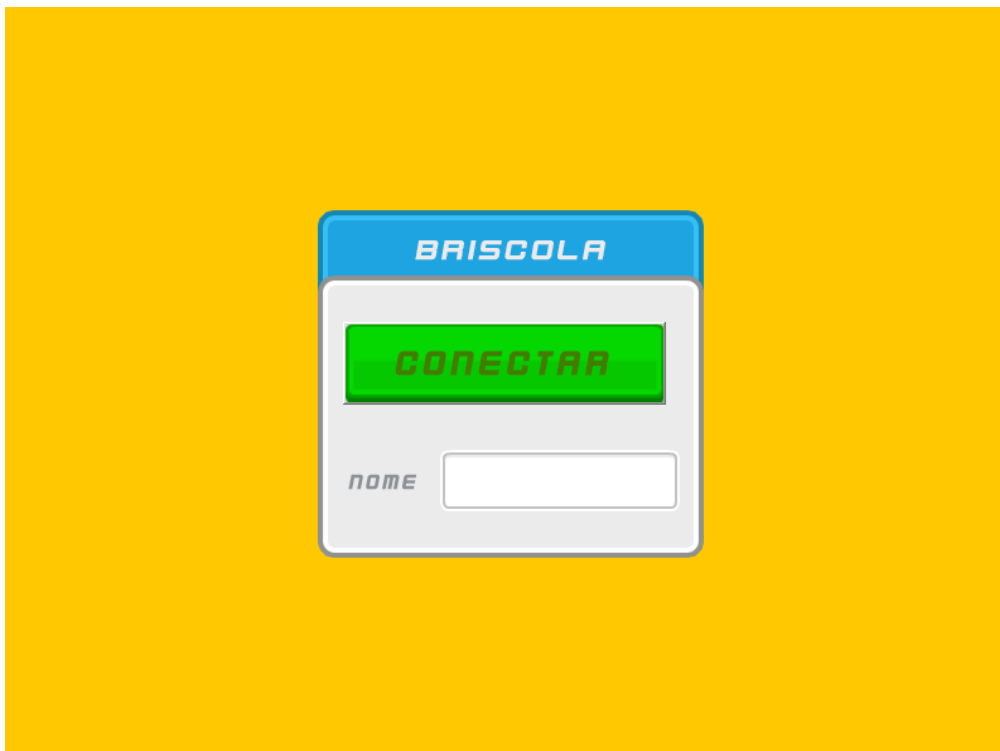
Requisito funcional 7 – Receber notificação de abandono: o programa deve poder receber uma notificação de abandono de partida por parte do adversário remoto, enviada por Dog Server. Neste caso, a partida deve ser considerada encerrada e o abandono notificado na interface.

3.2 Requisitos Não Funcionais

Requisito não funcional 1 – Tecnologia de interface gráfica para usuário: A interface gráfica deve ser baseada em *TKinter*;

Requisito não funcional 2 – Suporte para a especificação de projeto: a especificação de projeto deve ser produzida com a ferramenta Visual Paradigm;

Requisito não funcional 3 – Interface do menu do programa antes do usuário se conectar: A interface do menu do programa será produzida conforme o esboço da imagem abaixo.



Requisito não funcional 4 – Interface do menu do programa após conexão do usuário: A interface do menu do programa será produzida conforme o esboço da imagem abaixo.



Requisito não funcional 5 – Interface de jogo do programa: A interface de jogo do programa será produzida conforme o esboço da imagem abaixo.

