

Projeto Jogo FDP

Especificação de Requisitos de Software

Versão: 1.0

09/04/2024

Versão	Autor(es)	Data	Ação
1.0	Erik Orsolin de Paula, Pedro Augusto da Fontoura, Rafael Correa Bitencourt	09/04/2024	Estabelecimento dos requisitos

Conteúdo:

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

Apêndice: Regras do FDP

1. Introdução

1.1 Objetivo

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

1.2 Definições, abreviaturas

Regras do jogo: ver apêndice.

1.3 Referências

Regras do jogo disponíveis no site:

<https://todosjoga.wordpress.com/2012/03/25/filho-da-puta-35/>

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 de código, deve ser produzida especificação de projeto baseada em UML, segunda versão.

3. Requisitos de Software

3.1 Requisitos Funcionais

Requisito funcional 1 - Iniciar o programa: ao ser executado, o programa deve apresentar na interface a mesa do jogo, um círculo com um número representando os pontos do jogador (inicialmente zero), e um campo de texto ao lado que será contabilizado quantas “mesas” cada jogador disse que faria e quantas de fato fez (métrica responsável por definir as pontuações de cada jogador). Após isso solicitar conexão com o DOG server (utilizando os recursos de DOG). O resultado da tentativa de conexão deve ser informado ao usuário. Apenas em caso de conexão

bem-sucedida as demais funcionalidades estarão habilitadas. No caso de conexão mal sucedida, a única alternativa deve ser encerrar o programa;

Requisito funcional 2 - Iniciar jogo: o procedimento de início do jogo 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 da partida caso o contrário. A interface do programa deve ser atualizada com as informações recebidas (círculo que contabiliza pontos fica verde para o jogador que vai iniciar a partida). Caso o jogador local seja quem inicia a partida, a interface deve estar habilitada para seu procedimento de escolha de "quantas diz que faz" (botões de opções de quantas faz devem estar habilitados).

Requisito funcional 3 - Selecionar "quantas diz que faz": o programa deve permitir a um jogador habilitado selecionar um botão que determina "quantas diz que faz", se o botão selecionado não estiver habilitado não acontece nada, caso contrário, o campo de texto "Disse: ?" do jogador da vez, é atualizado com a quantidade que está gravada no botão selecionado, e a vez passa para o jogador seguinte. Quando todos os jogadores tiverem feito a ação, as cartas da mão do primeiro jogador se habilitam para jogar. O botão se encontra desabilitado quando, na vez do último jogador, seu valor somado aos "quantas diz que faz" dos outros jogadores totaliza o número de cartas com cada jogador.

Requisito funcional 4 - Jogar carta: o programa deve permitir ao primeiro jogador habilitado (todos já selecionaram "quantas diz que faz") clicar em uma carta para jogá-la na mesa, após isso a vez é passada para o próximo jogador (círculo de pontos fica verde) e as cartas do mesmo se habilitam para a ação.

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 2 - iniciar jogo*', isto é, a interface do programa deve ser atualizada com as informações recebidas e caso o jogador local seja quem inicia a partida, a interface deve estar habilitada para seu procedimento.

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 ser regular e conter as informações especificadas como carta jogada e qual jogador jogou. O programa deve mostrar a carta do jogador adversário sendo jogada na mesa. Após isso, avaliar se todas as cartas foram jogadas para determinar o vencedor da "mesa" (cada rodada, terá n "mesas", sendo n a quantidade de cartas inicialmente na mão de cada jogador). No caso de todas as cartas serem jogadas, deve ser adicionado 1 no campo "fez" do

jogador vencedor. No caso de não terem sido jogadas todas as cartas, deve ser habilitado o próximo jogador da vez para que possa proceder a jogada. Além disso, se for a última “mesa” da rodada, deve-se adicionar a pontuação específica de cada jogador naquela rodada (diferença absoluta entre “disse” e “fez”) e constatar o fim do jogo caso seja a 7ª rodada, anunciando o vencedor baseado na menor quantidade de pontos (anunciando os vencedores empatados, se for o caso).

Requisito funcional 9 – Receber notificação de abandono: o programa deve poder receber uma notificação de abandono de partida por parte de algum adversário remoto, enviada por Dog Server. Neste caso, a partida deve ser considerada encerrada e o abandono notificado na interface.

4. 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 programa: A interface do programa será produzida conforme o esboço da imagem abaixo:



Apêndice: Regras do FDP

Este jogo é uma atividade de cartas competitiva, que envolve estratégia, previsão e gestão de risco. Inicia-se com cada participante recebendo uma carta, de forma que todos, exceto o próprio jogador, possam vê-la, essa regra é válida apenas para a primeira rodada, nas próximas, o jogador vê apenas as suas cartas. A quantidade de cartas por mão aumenta progressivamente em cada rodada até atingir o limite de 7.

Os jogadores estimam e apostam no número de “mesas” de cada rodada que acreditam que vão ganhar com base nas cartas que possuem. A aposta ocorre no sentido anti-horário. A soma total das apostas deve ser diferente do número de cartas na mão de cada jogador naquela rodada, cabendo ao último apostador garantir essa desigualdade. Este aspecto da regra força uma estratégia adicional, pois o último jogador a apostar tem a responsabilidade de ajustar sua aposta para evitar um empate no total das previsões.

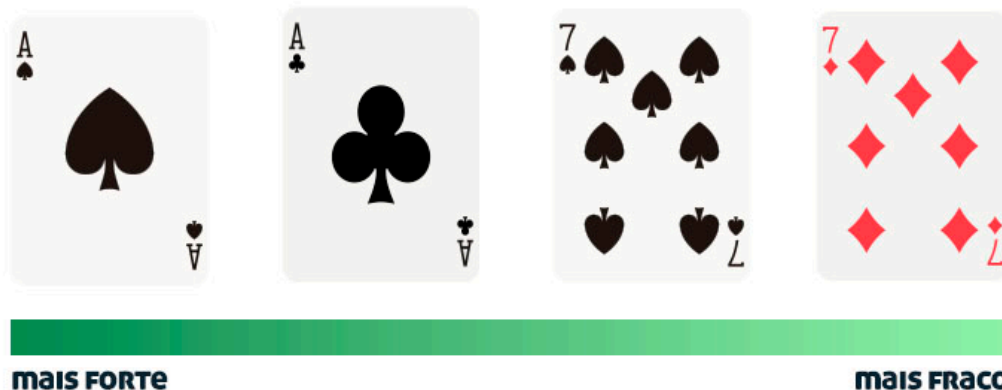
Após a conclusão de uma rodada, os jogadores comparam o número de vitórias reais com suas apostas. A diferença entre a previsão e o resultado real é convertida em pontos. Por exemplo, se um jogador aposta que ganharia três rodadas, mas ganhou apenas uma, ele recebe dois pontos. O objetivo é minimizar a quantidade de pontos recebidos ao longo do jogo.

O jogo termina quando chega-se ao final da 7ª rodada, determinando um ou mais vencedores, que serão aqueles que tiverem a menor quantidade de pontos.

Um detalhe importante é a regra de empate: se duas cartas de mesmo valor são jogadas, ambas são desconsideradas, e a vitória naquela rodada é determinada pela carta de valor mais alto remanescente.

O vencedor de uma “mesa” será o jogador que tiver a carta mais alta na mesa (desconsiderando as que empataram), seguindo a ordem de força do truco espanhol, como abaixo:

MANILHAS



CARTAS

3 2 A K J Q 7 6 5 4

mais forte

mais fraco

Em resumo, ganha quem tiver a melhor capacidade de antecipar o número de mesas vencidas por rodada, com o objetivo de acumular o menor número de pontos possível.