

Projeto Pife

Versão	Autores	Data	Ação
1.0	Luiz Maurício Luiz Felipe Gabriel Turatti	28/03/2023	Estabelecimento dos requisitos

Conteúdo:

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

1 - Introdução

1.1 - Objetivo

Desenvolvimento de um programa distribuído que suporte partidas na modalidade jogador x jogador do jogo de cartas Pife.

1.2 - Definições, abreviaturas

Trinca tipo trio: Trinca composta por três cartas de número igual, porém com naipes diferentes. Exemplo: A-A-A ou 7-7-7.

Trinca tipo sequência Trinca composta por três cartas de naipes diferentes e numeração diferente, porém que formam uma sequência numérica. Exemplo: A-2-3 ou J-Q-K.

Regras do jogo:

- O jogo utiliza dois baralhos padrão de 52 cartas cada, sem os coringas
- Existe um monte de descarte, onde apenas a carta do topo é visível
- Nove cartas são distribuídas para cada jogador
- O objetivo do jogo é formar trincas com as cartas da mão do jogador, obedecendo às seguintes restrições:
 - Em trincas com cartas de mesmo número, todas devem possuir naipes diferentes. Exemplo: 4-4-4.
 - Em trincas do tipo sequência, todas devem possuir naipe igual. Exemplo: A-2-3.
 - As cartas obedecem a ordem A-2-3-4-5-6-7-8-9-10-J-Q-K. A trinca Q-K-A não é uma trinca válida.
- O turno de um jogador possui duas ações possíveis:
 - O turno sempre começa com a compra de uma carta, podendo escolher entre o baralho e a pilha de descarte.

- O turno sempre termina com o descarte de uma carta da mão do jogador.
- Durante o turno de um jogador, é possível que ele baixe quantas trincas quiser. Cada trinca é imutável e visível para todos os jogadores, uma vez colocada na mesa ela não pode ser alterada.
- O jogo acaba quando algum jogador forma três trincas.

1.3 - Referências

Apresentação do jogo e das regras:

<https://www.megajogos.com.br/pife-pif-paf-online/regras>

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 1 - Iniciar programa: Ao ser executado, o programa deve exibir a interface com a mão do jogador em seu estado inicial e solicitar o nome do jogador. Após isso, deve solicitar conexão

com o DOG server. O resultado da conexão deve ser informado ao usuário. Caso haja uma conexão bem sucedida, iniciar o jogo. Caso contrário, encerrar o programa.

Requisito 2 - Iniciar partida: Apresentar menu “iniciar jogo”. O procedimento de início de partida consiste em mandar uma requisição de início de partida ao servidor, que retorna o resultado, contendo ordem e identificação dos jogadores caso êxito, ou uma mensagem de impossibilidade caso aconteça falha. A interface do programa deve ser atualizada com as informações recebidas e caso o jogador local seja o primeiro, habilitar interface para processo de jogada.

Requisito 3 - Receber início: O servidor envia uma mensagem de início de partida, onde caso o jogador local seja o primeiro a jogar, serão habilitadas as funcionalidades necessárias para realizar sua jogada.

Requisito 4 - Receber jogada: O servidor envia as informações da jogada do oponente remoto para o jogador local, atualizando a interface com os elementos. Checar se a partida foi finalizada. Caso positivo, exibir mensagem de término, caso negativo, habilitar o jogador local a realizar sua jogada.

Requisito 5 - Comprar carta: O jogador, durante seu turno, compra uma carta de uma das localizações disponíveis: baralho e monte descarte(caso não esteja vazio). A escolha do local de onde comprar a carta é clicada pelo jogador.

Requisito 6 - Baixar trinca: O jogador, durante seu turno, pode baixar uma trinca válida de sua mão. Uma trinca válida obedece às restrições estabelecidas nas regras do jogo, sendo elas: Trincas do tipo sequência devem possuir naipes iguais e trincas do tipo trio devem possuir naipes diferentes.

Requisito 7 - Descartar carta: Consiste em descartar uma carta da mão do jogador, colocando-a no topo do monte descarte. Esta ação simboliza o fim do turno do jogador, enviando a jogada realizada para o oponente remoto.

Requisito 8 - Receber desistência: Caso o oponente desista, o programa deve receber uma notificação de abandono do servidor DOG e encerrar a partida atual.

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

