

Projeto Red7

Especificação de Requisitos de Software

Versão 1.0

26/03/2023

Versão	Autor(es)	Data	Ação
1.0	Brenda Silva Machado	26/03/2023	Estabelecimento dos requisitos

Conteúdo:

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

1.Introdução

1.1 Objetivo

Desenvolvimento de um jogo distribuído para a disciplina Engenharia de Software I (INE5417) que suporte a disputa de partidas de Red7 na modalidade usuário contra usuário, estando os usuários em máquinas diferentes.

1.2 Definições, abreviaturas

- Paleta: termo que se refere à cor que rege aquela rodada;
- Rodada: designa a vez de um determinado jogador, na qual ele pode fazer um conjunto de ações que pode levar a sua vitória na rodada;
- Regra: designa uma sentença que cada cor possui, a qual só tem efeito quando essa cor é colocada na paleta e rege o jogo até que a mesma perca o efeito;
- Tela: diz respeito à região da mesa em que o jogador põe suas cartas;
- Partida: outro termo para jogatina, ou seja, o conjunto de rodadas do jogo.

1.3 Explicação do jogo

O programa simula uma jogatina de Red7, um jogo de cartas da Asmadi Games e distribuído no Brasil pela PaperGames. O original permite de 2 a 4 jogadores, porém para reduzir a complexidade de desenvolvimento, esse projeto foi adaptado para apenas 2 jogadores.

O jogo possui um baralho com 49 cartas, sendo que cada carta possui um número de 1 a 7 e uma de sete cores (vermelho, laranja, amarelo, verde, azul, índigo e roxo). Existe apenas uma carta para cada combinação. A ordem de grandeza das cartas, qual carta é mais forte que outra, vai crescendo dos números 1 ao 7 e do roxo para o vermelho. A maior carta do jogo é o 7 vermelho e a menor o 1 roxo. Cada cor possui uma regra associada a ela, descritas na tabela a seguir:

Cor da carta	Regra da Paleta
Vermelho	Maior carta
Laranja	Cartas de um número
Amarelo	Cartas de uma cor
Verde	Cartas pares
Azul	Cartas de cores diferentes
Índigo	Cartas em sequência
Roxo	Cartas abaixo de 4

As regras das cartas só fazem efeito quando aquela cor é a cor da paleta, ou seja, enquanto estiver na mão do jogador, a regra não vale para o jogo. À exceção da vermelha, as demais regras dizem respeito à quantidade de cartas, ou seja, o jogador com maior número de cartas na tela que atendam aquele requisito é o vencedor da rodada.

O objetivo do jogo é sempre ganhar a rodada que joga, ou seja, sempre que for a vez daquele jogador, ele deve garantir que a rodada fique à favor dele, seja mudando a cor da paleta e/ou baixando cartas na sua tela. Caso não seja possível, não haja ações que possam levar à vitória daquele jogador em sua rodada, ele perde a partida.

Caso as cartas de um jogador acabem, ele automaticamente vence o jogo.

Ao iniciar a partida, cada jogador recebe 7 cartas do baralho. A cor da paleta é automaticamente vermelha, antes das rodadas começarem. Um jogador é sorteado para iniciar sua rodada. As ações possíveis para cada jogador em sua rodada são:

- Baixar uma carta de sua mão na tela;
- Mudar a cor da paleta.

Ambas as ações só podem ser feitas uma vez durante a rodada, em qualquer ordem.

Depois que as ações terminam e o jogador vence a rodada, o próximo jogador inicia sua rodada. Caso o primeiro jogador perca, o outro jogador ganha a partida.

Os critérios de desempate, quando ambas as telas dos jogadores vençam pela regra da paleta, é a grandeza das cartas, sendo o primeiro aspecto considerado os

números maiores e o segundo as cores mais altas.

1.4 Referências:

<https://papergames.com.br/red7/>

http://www.tabuleirocriativo.com.br/post_red7.html

<https://www.ultraboardgames.com/red7/game-rules.php>

Apresentação das regras do jogo (video do canal Vem Ka Jogar):

<https://www.youtube.com/watch?v=qcX-XYdsKAY>

2. Visão Geral

2.1 Arquitetura do programa

Programa distribuído orientado a objetos.

2.2 Premissas de desenvolvimento

- O programa deve ser implementado na linguagem Python;
- O programa deve usar DOG como meio de aplicação distribuída;
- O programa deve ser modelado com base no paradigma de programação orientada a objetos (OO);
- A modelagem do programa deve seguir as especificações da linguagem UML 2.

3. Requisitos de Software

3.1 Requisitos Funcionais:

Requisito funcional 1 – Iniciar programa: ao ser executado, o programa deve mostrar o logo e o botão de iniciar. Ao clicar em “iniciar”, o jogador é levado ao início do jogo.

Requisito funcional 2 – Iniciar jogo: uma solicitação de início de uma partida é enviada

ao DOG, que verificará se há outro jogador querendo jogar também. No caso afirmativo, antes de iniciar a partida propriamente dita, o programa entrega 7 cartas do baralho para cada jogador, estas cartas são aleatoriamente sorteadas e dispostas na mão do jogador. As outras informações da interface da partida são carregadas na tela. Após isso, um jogador é sorteado para começar.

Requisito funcional 3 – Baixar carta na tela: o jogador pode realizar essa ação em sua rodada apenas. O jogador seleciona o botão e escolhe uma das cartas de sua mão para clicar e baixar em sua tela. A carta sai do seu baralho e é movida para a tela.

Requisito funcional 4 – Mudar a paleta: o jogador pode realizar essa ação em sua rodada apenas. O jogador seleciona o botão e escolhe uma carta de sua mão para mudar a paleta. A cor da carta escolhida não pode ser igual a cor vigente na paleta. Caso as cores sejam diferentes, a carta sai da mão do jogador e a paleta muda de cor.

Requisito funcional 5 - Passar turno: o jogador pode realizar essa ação em sua rodada apenas. O jogador seleciona o botão e sua rodada termina. O jogo verifica se aquele jogador ganhou ou não a rodada, analisando a regra e a mesa, se sim, passa para o próximo, senão o jogo encerra com a derrota desse jogador.

Requisito funcional 6 - Aviso de início: caso haja outro jogador que queira iniciar a partida, o servidor DOG irá mandar uma mensagem correspondente. A partir daí segue os procedimentos já descritos no 'Requisito funcional 2 – Iniciar jogo'.

Requisito funcional 7 - Receber jogada: o DOG server deve atualizar as informações toda vez que um jogador realiza uma ação, isto é, na rodada do jogador, quando ele realizar uma das ações possíveis, as mesmas devem ter seus efeitos mostrados logo em seguida. O mesmo ocorre para as ações do outro jogador na rodada seguinte.

Requisito funcional 8 - Aviso de desconexão: O DOG server avisará quando um dos jogadores desconectar do jogo. O jogo é encerrado e a mensagem aparecerá na interface.

Requisito funcional 9 - Avaliar vencedor: sempre que o turno passa, o jogo avalia se o jogador que passou venceu ou não a rodada. Caso afirmativo, o jogo continua, senão o outro jogador ganha o jogo automaticamente.

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 – Interface gráfica compartilhada: A interface gráfica deve ser atualizada conforme o compartilhamento entre os jogadores. A mesma deve exibir as informações do jogador que tem a rodada atual e trocar para as informações do

próximo jogador quando a rodada acabar;

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

