



Projeto Rei nos Cantos

Especificação de Requisitos de Software

Versão 2.0

30/09/2024

Versão	Autore(s)	Data	Ação
1.0	Jessica Regina dos Santos Marina Benvenuti Cardeal Yano Henrique de Melo Ribeiro Cavalcante	30/09/2024	Estabelecimento dos requisitos
2.0	Jessica Regina dos Santos Marina Benvenuti Cardeal Yano Henrique de Melo Ribeiro Cavalcante	27/10/2024	Reavaliação dos requisitos

Conteúdo:

1. Introdução
 2. Visão Geral
 3. Requisitos de Software
- Apêndice: Regras Reis nos Cantos

1. Introdução

1.1 Objetivo

Desenvolvimento de um programa distribuído que suporte a disputa de partidas de Reis nos Cantos (*Kings in the Corner*) na modalidade usuário contra usuário.

1.2 Referências

[Apresentação das regras do jogo \(vídeo do canal VemKaJogar\).](#)

1.3 Descrição e Definições

Descrição:

- Dois jogadores.
- Baralho comum de 52 cartas, sem coringas.
- Inicialmente, cada jogador recebe sete cartas.
- O objetivo do jogo é ser o primeiro a se livrar de todas as cartas na mão, organizando-as em sequências decrescentes na mesa, similar ao jogo Paciência. As cartas são organizadas alternando as cores vermelho e preto, independente dos naipes.

Definições:

- Rei no Canto: se um jogador tiver um Rei (carta K, preta ou vermelha), ele pode colocá-lo em qualquer um dos quatro cantos, se disponíveis. A ação é opcional, mas estratégica, pois abre novas aberturas para mover cartas.
- Ordem das cartas (da menor para a maior): Ás (A), 2, 3, 4, 5, 6, 7, 8, 9, 10, Valete (J), Dama (Q), Rei (K). Essas cartas estão presentes nos quatro naipes: Copas (♥), Ouros (♦), Espadas (♠), e Paus (♣). O Ás é sempre considerado a menor carta.

2. Visão Geral

2.1 Arquitetura do Programa

Programa orientado a objetos, cliente-servidor distribuído.

1.2 Premissas de Desenvolvimento

- O programa deve ser implementado na linguagem Python.
- O programa deve usar DOG como suporte para execução distribuída.
- Além do código, devem ser produzidas especificações de projeto baseadas em UML, segunda versão.

3. Requisitos de Software

3.1 Requisitos Funcionais

Requisito Funcional 1 – Iniciar programa

Ao ser executado, o programa deve apresentar a tela inicial com o logo do jogo e um campo de texto para o jogador digitar seu nome. Em seguida, será solicitada a conexão com o DOG Server utilizando os recursos disponíveis. O resultado da tentativa de conexão deve ser informado ao jogador. Caso a conexão seja bem-sucedida, a interface inicial da mesa será exibida, e as demais funcionalidades do jogo serão habilitadas. Caso a conexão não seja bem-sucedida, o usuário será notificado, e a única opção será encerrar o programa, podendo tentar novamente, conforme sua escolha.

Requisito Funcional 2 – Começar partida

O programa deve apresentar o botão funcional “Iniciar Jogo” para o início de uma nova partida. Para iniciar uma partida, ocorre uma solicitação de início ao Dog Server, que retornará um resultado: êxito ou fracasso. Em caso de êxito, o retorno será a identificação e ordem dos jogadores. Em caso de fracasso, a impossibilidade de início de partida. A interface do programa deve ser atualizada com as informações recebidas. Caso o jogador local seja o primeiro a jogar, a interface deve estar habilitada para o procedimento de lance, enquanto os jogadores restantes aguardam. Esta funcionalidade só estará habilitada se o programa estiver em estado inicial, ou seja, sem partida em andamento e com a mesa em seu estado inicial (quatro cartas abertas formando um padrão semelhante aos pontos cardeais Norte, Sul, Leste e Oeste, o restante das cartas fica no centro da mesa, servindo como baralho de compra para os jogadores. Além disso, são exibidos os botões funcionais do jogo, a mão do jogador apresentada exclusivamente para sua visualização e a mão do seu oponente, com visualização exclusiva do mesmo).

Requisito Funcional 3 – Receber determinação de início

O programa recebe uma notificação de início de partida, enviada pelo DOG Server, quando outro jogador também conectado ao servidor solicita o início de uma partida. A seguir, a interface deve ser atualizada e habilitada para que o jogador local possa iniciar o jogo e realizar as jogadas subsequentes (ver Requisito 2).

Requisito Funcional 4 – Iniciar nova mão

Ao fim de uma partida, seja por vitória de um jogador ou pelo abandono de partida por um jogador, o programa deve recolher todas as cartas restantes nas mãos dos jogadores perdedores e na mesa, retornando-as ao baralho. Em seguida, o baralho deve ser embaralhado e as cartas redistribuídas aos jogadores para o início de uma nova partida, caso desejado.

Requisito Funcional 5 – Receber notificação de desistência

O programa deve ser capaz de receber uma notificação de abandono de partida por parte do adversário remoto via DOG Server. Nesse caso, o jogo deve ser encerrado, e o abandono deve ser notificado na interface.

Requisito Funcional 6 – Avaliar vencedor

Para ganhar o jogo, o jogador deve ter colocado e manipulado corretamente todas as suas cartas na mesa. O programa deve verificar se não há mais cartas na mão de um jogador. Se o jogador estiver com a mão vazia e tiver manipulado suas cartas de acordo com as regras do jogo, ele será declarado o vencedor da partida.

Requisito Funcional 7 – Indicar vencedor

Ao final de cada partida deve ser apresentado tanto ao jogador local quanto ao remoto uma tela indicando o nome do jogador que ganhou aquela partida.

Requisito Funcional 8 – Receber jogada

O programa deve poder receber uma jogada do adversário, enviada pelo DOG Server, quando for a vez do adversário do jogador local. A jogada recebida deve corresponder a uma das seguintes ações: "Manipular cartas" ou "Colocar Rei no canto". Cada jogada mencionada deve respeitar os requisitos funcionais de sua respectiva ação. Após isso, o programa deve avaliar o encerramento da partida. No caso de encerramento, o jogador vencedor será notificado conforme o Requisito Funcional 7 – Indicar vencedor. Se não houver encerramento, o jogador local será habilitado para realizar sua jogada.

Requisito Funcional 9 – Passar a vez

Se for o turno do jogador, ele pode pressionar o botão "Passar a Vez". Ao realizar essa ação, o programa deve encerrar o turno do jogador atual e transferir o turno para o outro jogador. O botão "Passar a Vez" estará disponível apenas durante o turno do jogador.

Requisito Funcional 10 - Enviar popup

Se o jogador tentar realizar uma jogada inválida ao manipular suas cartas, o sistema deve exibir um popup informando o erro. Isso ocorre nas seguintes situações: quando o jogador tenta colocar uma carta que não seja o Rei em um dos quatro cantos da mesa, ou tenta abrir mais de quatro cantos, o que não é permitido; se o jogador intercalar cartas da mesma cor consecutivamente, violando a regra de alternância entre cartas pretas e vermelhas, ou se desrespeitar a sequência lógica das cartas (Ás, 2, 3, 4, 5, 6, 7, 8, 9, 10, Valete, Dama, Rei). Em qualquer uma dessas condições, o popup deve notificar o erro, impedindo o jogador de continuar até que ele corrija a jogada de acordo com as regras.

Requisito Funcional 11 - Comprar carta

O programa deve "obrigar" o jogador habilitado a comprar uma carta do monte central no início de uma jogada. A compra deve ser registrada e a nova carta adicionada à mão do jogador. Essa ação é obrigatória para que o jogador possa continuar a jogada. Após a compra da carta, o jogador pode então prosseguir.

Requisito Funcional 12 - Jogar carta

O programa deve permitir que o jogador habilitado selecione uma coluna ou um dos cantos, onde será colocada a carta de sua escolha (da sua mão ou da mesa). O programa deve avaliar se a posição de destino pode receber a peça selecionada. Se o destino for válido conforme as regras do Reis nos Cantos, a jogada deve ser efetuada (isto é, a carta sair do seu posicionamento inicial e ir para o destino). Caso contrário, o programa deve notificar a jogada irregular e aguardar uma nova ação (tentar novamente). A partir da jogada realizada, o programa deve enviar a jogada ao adversário via DOG Server (a jogada enviada deve conter as movimentações/ações do jogador previamente habilitado). Se o jogo continuar (não houve vitória ainda), o jogador local é desabilitado e o programa fica em aguardo. Em caso de vitória, a partida é encerrada e deve ser notificado o jogador vencedor.

Requisito Funcional 13 - Manipular cartas

O jogador deve ser capaz de manipular as cartas sobre a mesa, formando sequências válidas de acordo com as regras do jogo. Isso inclui o ato de mover cartas de uma posição para outra e intercalar cores (pretas e vermelhas) de forma alternada. A sequência lógica das cartas (Ás, 2, 3, 4, 5, 6, 7, 8, 9, 10, Valete, Dama, Rei) deve ser respeitada ao formar essas sequências.

Requisito Funcional 14 - Colocar Rei no canto

O jogador habilitado deve poder colocar um Rei em um dos quatro cantos disponíveis da mesa, abrindo uma nova possibilidade de sequência de cartas. Apenas o Rei pode ser colocado nos cantos, e essa ação permite que o jogador continue organizando as cartas de acordo com as regras do jogo.

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 do projeto deve ser produzida com a ferramenta Visual Paradigm.

Requisito Não Funcional 3 – Interface do programa

Os principais elementos da interface gráfica do jogo estão presentes nas figuras abaixo.

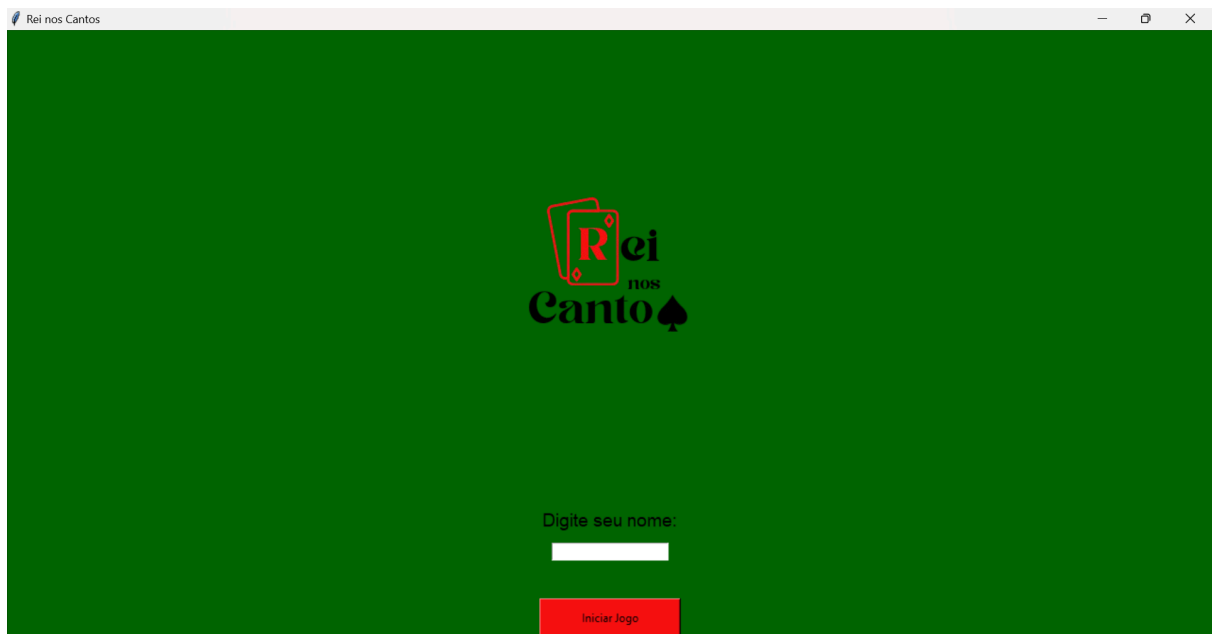


Figura 1. Tela inicial



Figura 2. Ponto de vista do jogador local

Apêndice

Regras Rei nos Cantos

Rei nos Cantos é um jogo de cartas para 2 a 4 jogadores. O objetivo do jogo é ser o primeiro a se livrar de todas as cartas na mão, organizando-as em sequências decrescentes na mesa, similar ao jogo Paciência. As cartas são organizadas alternando as cores vermelho e preto e devem ser manipuladas de acordo com as regras abaixo.

Elementos do Jogo



Figura 3. Créditos [VemKaJogar](https://www.vemkajogar.com)

Cada jogador recebe 7 cartas no início do jogo. Quatro cartas são abertas do baralho central (um baralho comum, sem coringas) e colocadas em torno do mesmo, formando um padrão semelhante aos pontos cardeais: Norte, Sul, Leste e Oeste. O restante das cartas ficam no centro da mesa, de onde os jogadores compram cartas.

Lances dos Jogadores

De forma alternada, os jogadores realizam lances em sentido horário. O jogador sempre começa um turno comprando uma carta do monte central. Em seguida, o jogador poderá:

- **Colocar cartas na mesa**
O jogador pode organizar as cartas de forma decrescente, intercalando as cores. Exemplo: um 8 preto seguido por um 7 vermelho.
- **Mover cartas na mesa**

O jogador pode mover sequências de um lado para o outro livremente, desde que as novas posições respeitem a regra de ordem decrescente e alternância de cores.

- **Abrir um espaço vazio**

Se um espaço na mesa fica vazio (sem cartas), o jogador fica livre para colocar qualquer carta nesse espaço, se desejar.

- **Colocar um Rei nos cantos**

Se um jogador tiver um Rei (carta K, preta ou vermelha), ele pode colocá-lo em qualquer um dos quatro cantos. A ação é opcional, mas estratégica, pois abre novas aberturas para mover cartas.

- **Vencer o jogo**

O jogo termina quando um jogador se livra de todas as cartas de sua mão.

Algumas Estratégias

Rei nos Cantos envolve a manipulação eficaz das cartas na mesa pelos jogadores. Eles devem prestar atenção às oportunidades de reorganizar suas cartas, criando novos espaços e maximizando suas opções de jogadas.

O uso dos Reis é uma jogada decisiva, pois abre novas sequências e pode desbloquear movimentos estratégicos importantes. Mas é fundamental saber o momento certo de usá-los!

Além disso, o jogador deve aproveitar as oportunidades de mover sequências e esvaziar espaços, mantendo o controle da mesa para facilitar o descarte de suas cartas e ganhar o jogo.