



Especificação de Requisitos de Software

Sequência

Cristiano Azevedo dos Santos Antonio (20100520)

João Mai (20200407)

Victor do valle Cunha (20104135)

Especificação de Requisitos de Software

Versão 2.0

17/10/2022

Versão	Autores	Data	Ação
1.0	Cristiano Azevedo dos Santos Antonio João Mai Victor do Valle Cunha	07/09/2022	Estabelecimento de requisitos
2.0	Cristiano Azevedo dos Santos Antonio João Mai Victor do Valle Cunha	17/10/2022	Atualização dos requisitos

Conteúdo

Introdução	3
Objetivo	3
Definições e abreviaturas	3
Referências	3
Visão Geral	3
Arquitetura do programa	3
Premissas de desenvolvimento	3
Requisitos de software	3
Requisitos funcionais	3
Requisitos não funcionais	6
Apêndice: Regras do Sequência	7
Objetivos	7
Preparação	7
Lances dos jogadores	7
Valetes e casas curingas	8
Encerramento	8
Notas	8

1. Introdução

1.1. Objetivo

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

1.2. Definições e abreviaturas

Regras do jogo: apêndice

1.3. Referências

<https://copag.com.br/blog/detalhes/sequence>

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: Quando o programa for executado deve mostrar uma interface com o tabuleiro em seu estado inicial. Nesta tela deve conter um tabuleiro com 100 cartas, um espaço reservado para as cartas do jogador, um espaço para o descarte. Em seguida, solicita-se uma conexão com o DOG Server (utilizando os recursos do DOG). Caso a conexão seja bem sucedida, o usuário é liberado para iniciar uma partida. Caso o contrário, o programa é encerrado.

Requisito funcional 2 – Iniciar jogo: O programa deve conter em sua aba de configurações uma opção chamada “iniciar jogo” para começar uma nova partida. Esse procedimento envia uma requisição de início para o Dog Server. Caso a solicitação seja feita corretamente, o DOG Server retorna a identificação dos jogadores juntamente com a informação de quem irá iniciar a partida. Caso o contrário, o servidor informa qual foi o motivo da falha. Em seguida, o jogador que irá iniciar a partida é habilitado a realizar as ações possíveis do jogo enquanto que outro jogador só será atualizado com informações sobre o início da partida e que não é o seu turno.

Requisito funcional 3 – Restaurar estado inicial: O programa deve conter em sua aba de configurações uma opção chamada “restaurar estado inicial” para retirar todas as peças do tabuleiro, esvaziar a pilha de descartes e redistribuir as cartas para os jogadores. Esse procedimento só deve ser liberado quando uma partida estiver finalizada.

Requisito funcional 4 – 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’, ou seja, 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 de lance.

Requisito funcional 5 – 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 um lance regular e conter as informações especificadas para o envio de jogada no ‘Requisito funcional 9 –Colocar ficha’. O programa deve remover a peça de origem definida e colocá-la no destino. Após isso, deve-se avaliar o encerramento de partida. No caso de encerramento de partida, deve ser notificado o nome do jogador vencedor; no caso de não encerramento, deve ser habilitado o jogador local, para que possa proceder a seu lance;

Requisito funcional 6 – 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.

Requisito funcional 7 – Selecionar carta: O programa deve habilitar o jogador que estiver em sua vez para selecionar uma carta em sua área “mão”. A carta selecionada deve ser visualmente destacada na interface do programa. Caso a ação seja regular, ou seja, há posições livres no tabuleiro, o programa deve retornar uma nova carta do monte se o jogador estiver descartando uma “carta morta” e deve notificá-lo para selecionar uma nova carta para o descarte. No entanto, se o jogador estiver descartando uma carta qualquer, sua peça deve ser colocada na posição desejada (neste caso, ver Requisito funcional 4) e a interface deve mostrar a carta descartada na pilha de descartes. Caso a ação seja irregular, o jogador deve ser notificado que a sua ação não é permitida e, conseqüentemente, permiti-lo selecionar uma outra carta.

Requisito funcional 8 - Comprar carta: o programa deve comprar uma carta do monte de compra para que o jogador totalize 7 cartas em sua mão. Além disso, após o descarte de uma carta morta, para o monte de descarte, o jogador deve receber uma nova carta em sua mão e a possibilidade de colocar uma ficha (neste caso, ver Requisito funcional 9)

Requisito funcional 9 - Colocar ficha: O programa deve habilitar o jogador para selecionar uma das possíveis posições que sua ficha pode ocupar. No entanto, essa funcionalidade só é liberada quando há mais de uma possibilidade para posicionar uma peça (as duas posições não possuem peças até o momento da jogada). Dessa forma, se houver uma única possibilidade para posicionar uma peça, o programa que seleciona a posição. Após ter colocado a peça na posição, o programa deve enviar a jogada ao adversário (utilizando os recursos de DOG) e avaliar o encerramento de partida. A jogada enviada deve conter a carta na pilha de descartes e a peça colocada no tabuleiro. Se a partida for encerrada, deve ser notificado o nome do jogador vencedor; caso o contrário, o jogador local deve ser desabilitado e o programa fica no

aguardo de jogada do adversário (ver Requisito funcional 5) ou de notificação de abandono (ver Requisito funcional 6);

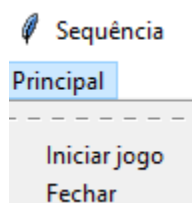
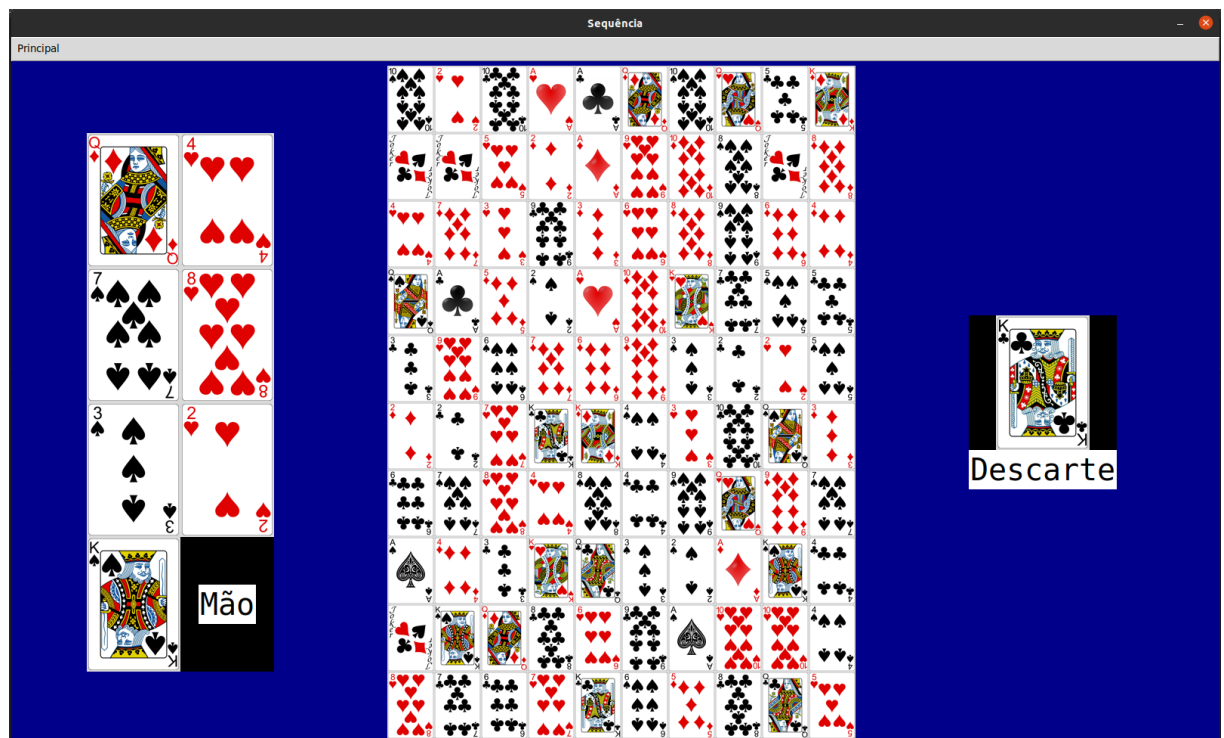
Requisito funcional 10 - Retirar ficha: O programa deve habilitar o jogador a remover uma peça do oponente a partir da jogada de um “**Valete lateral**”, não sendo permitido remover fichas que já formaram uma sequência. Após ter retirada a peça da posição, o programa deve enviar a jogada ao adversário (utilizando os recursos de DOG). A jogada enviada deve conter a carta na pilha de descartes e a peça retirada no tabuleiro. O jogador local deve ser desabilitado e o programa fica no aguardo de jogada do adversário (ver Requisito funcional 5) ou de notificação de abandono (ver Requisito funcional 6);

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.



4. Apêndice: Regras do Sequência

4.1. Objetivos

O jogo Sequência é disputado entre no mínimo 2 jogadores em um tabuleiro de 107 posições, sendo que cada uma das posições é uma carta de um baralho comum. O objetivo dos jogadores é criar 2 (duas) sequências ininterruptas de 5 peças da mesma cor na linha horizontal, vertical ou diagonal.

4.2. Preparação

Os jogadores escolhem uma cor de ficha e recebem 7 cartas cada um. O resto das cartas são utilizadas como “monte”.

4.3. Lances dos jogadores

Os jogadores procedem a seus lances de forma alternada, sendo que na primeira rodada cada um deles retira uma carta do monte de maneira aleatória para decidir

quem começa a partida. Cada jogador, na sua vez, pode efetuar os seguintes lances: descartar uma carta de sua mão e retirar uma carta do monte.

Descartar uma carta da sua mão: Para posicionar uma peça no tabuleiro, o jogador precisa descartar uma carta da sua mão para o “monte de descartes”- com o mesmo valor e "naipe" da posição do tabuleiro o qual ele deseja posicionar sua peça. Cada jogador só pode descartar uma carta por rodada e, conseqüentemente, posicionar uma única peça.

Retirar uma carta do monte: Consiste em retirar uma carta do topo do monte de cartas.

4.4. Valetes e casas curingas

Os Valetes são cartas com poderes específicos que permitem fazer jogadas especiais. Existem 2 tipos de Valetes: Valetes frontais e Valetes laterais.

Os **Valetes frontais** (dois olhos) permitem colocar uma ficha na casa que quiser do tabuleiro, mesmo que não tenha a carta desta casa.

Os **Valetes laterais** (um olho) permitem retirar uma ficha do tabuleiro do jogador oponente.

4.5. Encerramento

O jogo é finalizado quando um dos jogadores consegue montar duas sequências de peças ou não existir mais possibilidade de se gerar sequências com os espaços em branco restantes.

4.6. Notas

- 4.6.1. Uma sequência completa fica protegida. Não se pode eliminar nenhuma ficha.
- 4.6.2. 2 (duas) sequências que se cruzam da mesma cor podem compartilhar uma ficha em comum.
- 4.6.3. Se você estiver segurando uma carta que não serve para nenhum espaço do tabuleiro, pois os espaços que representam esta carta estão ocupados

por uma ficha ou dois valetes foram utilizados para tirar sua peça de uma mesmo, o jogador está com uma carta morta. Nesse caso, o jogador pode descartar essa carta em seu turno e retirar uma nova do monte.