
Projeto Mu Torere

Especificação de Requisitos de Software

Versão 0.4

Versão	Autor(es)	Data	Ação
0.1	Guilherme Adenilson de Jesus Gian Carlo Figueiredo Ferrari Vicente Cardoso dos Santos	17/08/2023	Especificação de requisitos
0.2	Guilherme Adenilson de Jesus Gian Carlo Figueiredo Ferrari Vicente Cardoso dos Santos	24/08/2023	Revisão das especificações de requisitos
0.3	Guilherme Adenilson de Jesus Gian Carlo Figueiredo Ferrari Vicente Cardoso dos Santos	06/09/2023	Revisão das especificações de requisitos
0.4	Guilherme Adenilson de Jesus Gian Carlo Figueiredo Ferrari Vicente Cardoso dos Santos	06/10/2023	Revisão das especificações de requisitos

TABELA DE CONTEÚDOS

1. INTRODUÇÃO

- 1.1. Objetivo do desenvolvimento
- 1.2. Definições e abreviaturas
- 1.3. Referências

2. VISÃO GERAL

- 2.1. Arquitetura do software
- 2.2. Premissas de desenvolvimento

3. REQUISITOS DA APLICAÇÃO

- 3.1. Regras de negócio
- 3.2. Requisitos funcionais
- 3.3. Requisitos não-funcionais

4. APÊNDICE

- 4.1. Regras do Jogo

1. INTRODUÇÃO

1.1. Objetivo do desenvolvimento

O presente projeto visa o desenvolvimento de um jogo de tabuleiro, Mu Torere, para computador no formato usuário contra usuário e com arquitetura distribuída.

1.2. Definições e abreviaturas

RN: Regra de negócio

RF: Requisito funcional

RNF: Requisito não-funcional

1.3. Referências

Regras do jogo: Ver apêndice.

Funcionamento em vídeo do jogo:  Beating my wife in Mu Torere

2. VISÃO GERAL DO SISTEMA

2.1. Arquitetura do software

Cliente-servidor distribuído.

2.2. Premissas de desenvolvimento

- O programa deve ser implementado em Python;
- A especificação de projeto a ser desenvolvida deve ser baseada em UML, segunda versão;
- O *software* deve utilizar DOG como suporte para execução distribuída.
- A especificação do projeto deve ser produzida com a ferramenta Visual Paradigm.

3. REQUISITOS DA APLICAÇÃO

3.1. Regras de negócio

RN 01: O jogo precisa de dois (2) jogadores;

RN 02: Ganha quem bloquear o movimento do outro jogador;

RN 03: O movimento das peças deve obedecer às regras expressas no apêndice.

3.2. Requisitos funcionais

RF 01 – Inicialização do programa: ao executar o *software*, a interface principal e seus elementos (indicada na **RNF 01**) devem ser carregados, junto a uma caixa de entrada que requisita o apelido do jogador. Quando o usuário confirma seu nome, tenta-se uma conexão com o servidor. Caso ocorra uma falha de comunicação, o jogador será notificado e deverá reiniciar o programa.

RF 02 – Início da partida: quando um dos jogadores clicar no botão “Iniciar Partida” na tela principal, o servidor fará um pareamento com um oponente conectado ao servidor DOG (segundo **RN 01**). Caso nenhuma partida seja encontrada, o servidor enviará uma resposta, que será mostrada na interface, e o usuário terá a opção de iniciar a partida novamente.

RF 03 – Identificação de movimentos possíveis: ao início do turno, o *software* deve verificar e destacar, para o jogador do turno, as peças que podem ser movidas, conforme **RN 03**.

RF 04 – Escolha da peça a ser movida: após a identificação das peças válidas, o jogador deve selecionar, clicando, qual delas irá mover. Após a seleção, a interface atualizará a posição da peça, o turno será encerrado e a jogada enviada ao servidor.

RF 05 – Determinação de vitória: após a movimentação da peça, o *software*, através da análise do resultado obtido em **RF 03**, determina a configuração do tabuleiro, conforme a **RN 02**. No caso de haver uma vitória, ambos os jogadores serão notificados e a partida é encerrada.

RF 06 – Desistir da partida: a qualquer momento durante a partida, ambos os jogadores têm a opção de desistir clicando no botão relevante da interface. No caso de desistência por um jogador, a desistência será enviada para o servidor, seu oponente receberá a vitória e a partida é encerrada.

RF 07 – Receber jogadas: o *software* deve ser capaz de receber e transmitir jogadas pelo servidor após o fim do turno. O jogador da vez enviará pelo servidor a jogada, e o seu oponente receberá e terá seu estado de jogo atualizado, de forma a manter a partida sincronizada.

RF 08 – Receber notificação de abandono: o *software* deve ser capaz de receber notificação de abandono de partida de seu oponente, enviada pelo servidor. Dessa forma, o jogador restante será notificado pela interface, receberá a vitória e a partida será encerrada.

RF 09 – Receber determinação de início: após a inicialização, o *software* deve ser capaz de receber uma mensagem de início de partida pelo servidor. O jogador relevante será transferido para a partida que foi iniciada.

3.3. Requisitos não-funcionais

RNF 01 – Interface gráfica do programa: a interface gráfica deve ser implementada usando TKinter conforme o esboço na Figura 1 abaixo.

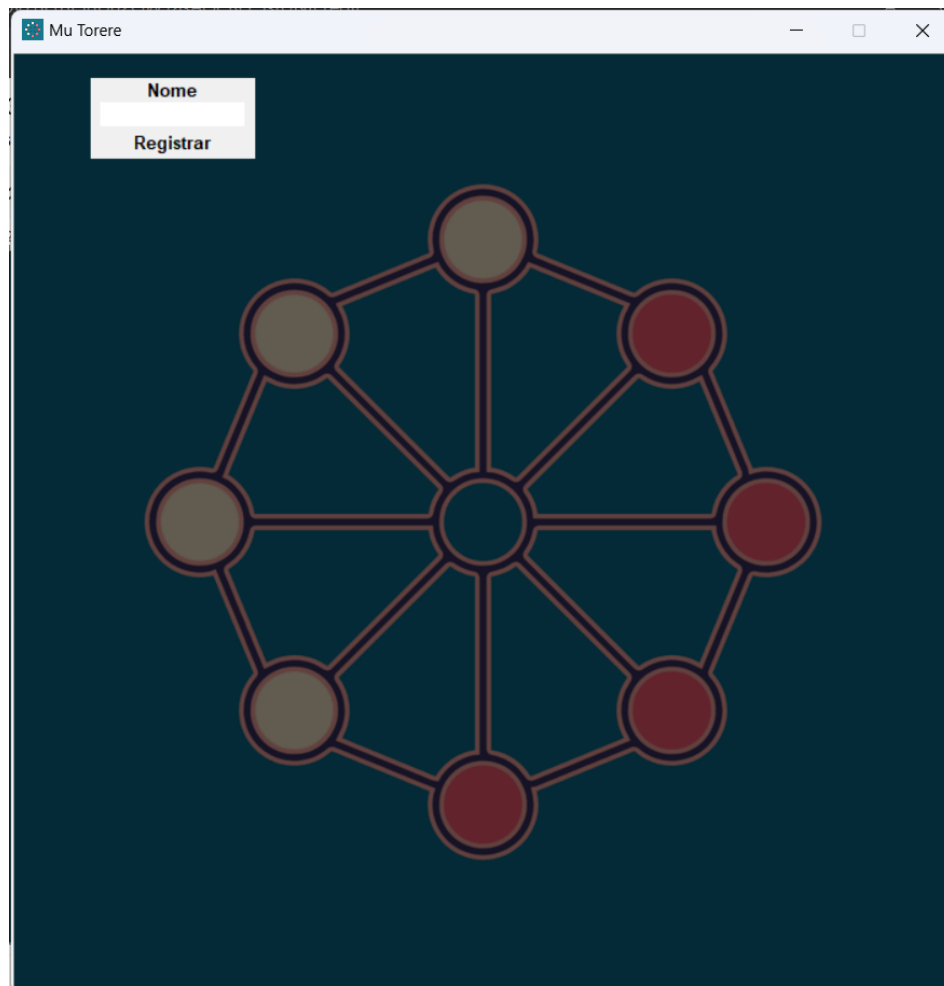


Figura 1 - Esboço de Interface Gráfica
Fonte: Autoria Própria

4. APÊNDICE

4.1. Regras de jogo

O tabuleiro do jogo é um octógono, e cada jogador começa com quatro peças distribuídas adjacentes às suas outras peças, de modo que o tabuleiro inicial é composto por 4 peças brancas e 4 peças pretas em 'arcos' opostos. Há um ponto central ligando os oito vértices do octógono, e estas 9 casas compõem o espaço de

movimento das peças (Figura 2 abaixo). O jogador com primeiro turno é decidido arbitrariamente.

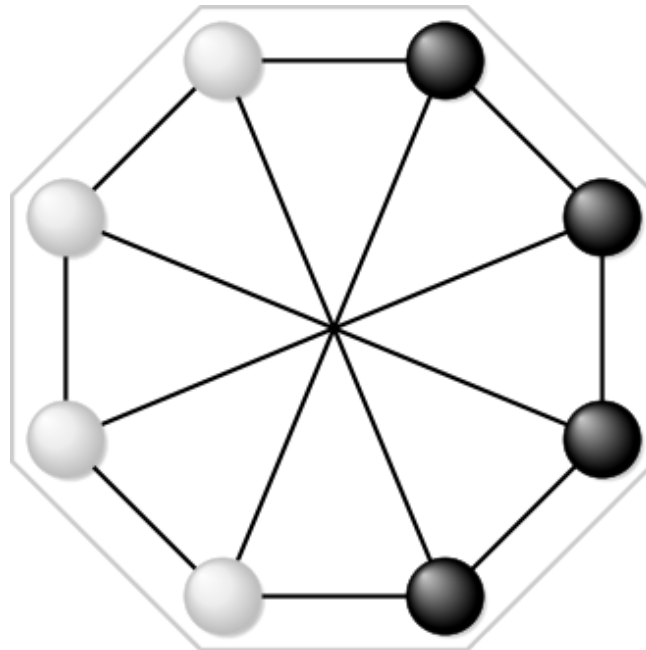


Figura 2: Tabuleiro do Mu Torere

Fonte: https://en.wikipedia.org/wiki/M%C5%AB_t%C5%8Drere

As peças podem se mover de três formas:

- De um vértice para outro vértice, que é sempre permitido (assumindo que o vértice destino não esteja ocupado por outra peça);
- De um vértice para o centro, que só é permitido se a peça em questão estiver adjacente à uma ou duas peças inimigas;
- Do centro para um vértice vazio, que sempre é permitido.

O objetivo do jogo é configurar o tabuleiro de tal forma que as peças adversárias não tenham movimentos válidos, e neste cenário o jogador cujas peças bloqueiam o movimento adversário é tido como vencedor da partida.

(Disponível em: https://en.wikipedia.org/wiki/M%C5%AB_t%C5%8Drere.)