

Especificação de Requisitos

Projeto Uno en Medio

Especificação de Requisitos de Software

Versão: 3.0

Data: 04/07/2022

Versão	Autor(es)	Data	Ação
1.0	Bruno da Silva Castilho, Lucas Coelho Pini de Sousa e Rafael Begnini de Castilhos	01/05/2022	Definição inicial dos requisitos do projeto
2.0	Bruno da Silva Castilho, Lucas Coelho Pini de Sousa e Rafael Begnini de Castilhos	06/06/2022	Adicionado imagens para ilustrar as regras do jogo e o esboço da interface.
3.0	Bruno da Silva Castilho, Lucas Coelho Pini de Sousa e Rafael Begnini de Castilhos	04/07/2022	Adição de regra para sortear jogador inicial e sugestões de posições adjacentes

Conteúdo:

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

1- Introdução

Objetivo do projeto: Desenvolvimento de um programa que suporte a disputa de partidas do jogo Uno em Medio entre dois jogadores em uma mesma máquina.

Abreviações: Nesse documento, as seguintes abreviações estarão sendo utilizadas:

- ❖ RF: Requisito Funcional
- ❖ RNF: Requisito Não Funcional

Regras: As quatro peças iniciam nas quinas do tabuleiro, sendo que as duas peças da mesma cor devem ser postas do mesmo lado do tabuleiro (figura 1).

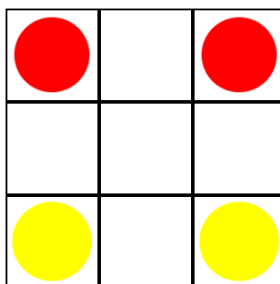
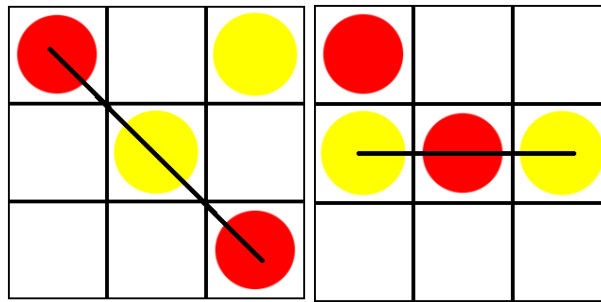


Figura 1 (Disposição inicial das peças no tabuleiro)

Os lances são alternados e, na sua vez, o jogador escolhe uma de suas peças e a movimenta por uma única casa adjacente em alguma orientação: horizontal, vertical ou diagonal. Saltos não são permitidos.

A condição de vitória é alinhar 3 peças em qualquer uma das direções (horizontal, vertical ou diagonal), de modo a colocar duas peças de uma cor nas pontas e uma peça da outra cor entre elas. O jogador responsável por fazer a movimentação que leva a esse alinhamento será considerado o ganhador, não sendo necessário que as duas peças que compõem o alinhamento sejam da cor do jogador que o efetuou. Exemplos de alinhamentos são demonstrados abaixo (figuras 2 e 3):



Figuras 2 e 3 (Alinhamento de peças para satisfazer a condição de vitória)

2 - Visão Geral

Arquitetura do programa: programa orientado a objetos, stand-alone.

Premissas de desenvolvimento:

- ❖ O programa deve apresentar uma interface gráfica bidimensional;
- ❖ O programa deve ser implementado em Python, devendo executar em qualquer plataforma que disponha da linguagem interpretada Python;
- ❖ O programa deve dar suporte à interação local entre os usuários;

3.1 - Requisitos Funcionais

[RF1] Iniciar partida - O programa deve apresentar uma tela de início que precede a tela do jogo, onde os jogadores devem escolher sua cor e inserir seus nomes. Após isso sorteado um número pseudo aleatório entre 0 e 1, se for 0 é definido como jogador que possui a vez de jogar o jogador 1, caso contrário é definido como jogador que possui a vez de jogar o jogador 2, assim dando início a partida, podendo escolher a peça que deseja movimentar.

[RF2] Estratégia - O programa deve permitir que sejam disputadas partidas entre dois jogadores em modo standalone, sendo que o software deve garantir o cumprimento das regras do jogo (referência: **introdução - regras**).

[RF3] Resetar a partida - O programa deve disponibilizar aos usuários um “reset game”, onde o software finaliza a partida em andamento e voltará para a tela de início da partida (referência: **[RF1] Iniciar partida**).

[RF4] Procedimento de lance - O programa deve suportar o procedimento de lance dos usuários por meio de click de botão esquerdo do mouse sobre a posição a onde uma de suas pedras está localizada (origem), e posteriormente a posição para a qual deseja transportar (destino), considerando as seguintes restrições:

- O programa deve realizar a sugestão de posições adjacentes válidas de acordo com o clique na posição de origem desejada.

- O programa deve permitir apenas lances válidos, de acordo com as regras do jogo (referência: **introdução - regras**).
- O programa deverá permitir que o usuário desista de uma jogada em que uma pedra já foi selecionada, mas não poderá voltar uma jogada em que a pedra já foi movimentada.
- A cada lance válido, o programa deve verificar se o jogador que jogou ganhou a partida.

[RF5] Janela de alerta - A interface gráfica deverá ter uma janela de alerta para avisar o fim da partida e decretar o vencedor. Ao clicar em “ok” o usuário deverá ser direcionado para a tela de início (referência: **[RF1] Iniciar partida**).

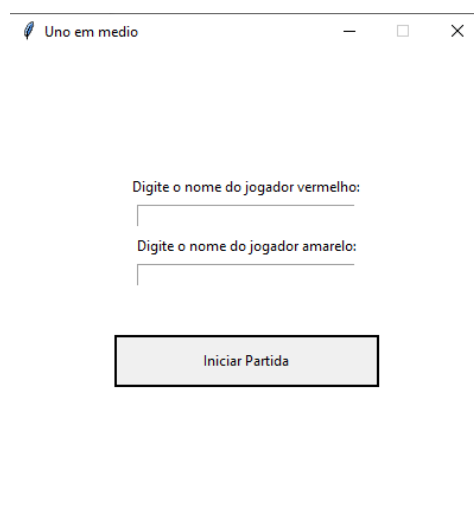
3.2 - Requisitos Não Funcionais

[RNF1] Especificação do projeto - Além do código fonte ser implementado em Python, deve ser produzida especificação de projeto baseada em UML na segunda versão.

[RNF2] Interface gráfica para o usuário - O programa deverá ter interface gráfica única, compartilhada pelos usuários. Com a seguinte cara.

[RNF3] Tecnologia para interface gráfica com o usuário - A interface gráfica deve ser baseada em Tkinter.

[RNF4] Esboço da interface - A interface gráfica deve seguir as imagens seguintes:



The image shows a Tkinter window titled "Uno em médio". Inside the window, there are two text input fields. The first field is preceded by the label "Digite o nome do jogador vermelho:" and the second by "Digite o nome do jogador amarelo:". Below these two fields is a rectangular button with the text "Iniciar Partida".

Imagem 1 (Tela de início)

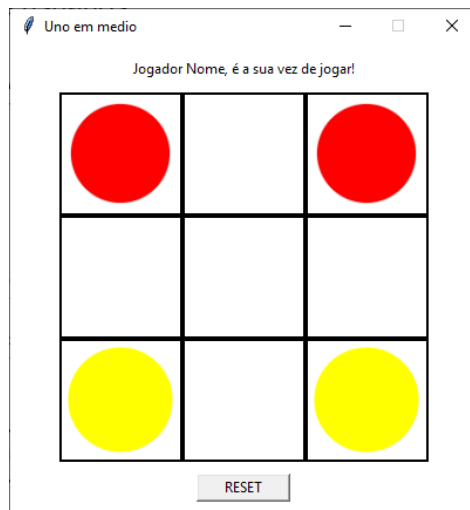


Imagem 2 (Tela do jogo)

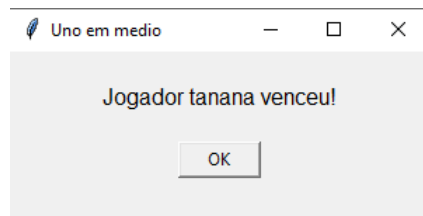


Imagem 3 (Tela de notificação de vitória)