Especificações de Requisitos

Lasca

Versão: 1.1
Data: 17/05/2022

Versão	Autor(es)	Data	Ação
1.0	Anthon Porath Gretter Rian Block Serena Nicolas Elias Santana	09/05/2022	Implementação inicial das especificações de requisitos e da interface gráfica.
1.1	Nicolas Elias Santana	17/05/2022	Imagens da interface grafica.

Índice

1. Introdução

1.1 Objetivo

Desenvolver um programa em Python que funcione como uma versão digital do jogo Lasca.

1.2 Referências

http://www.lasca.org/

2. Visão Geral do Sistema

2.1 Premissa do desenvolvimento

- Implementado em Python 3;
- Interface gráfica com biblioteca nativa Tkinter;
- Controle de versionamento com Git;
- Gerenciamento de projeto com a ferramenta ‘Projects’ do Github;
- Modelagem do desenvolvimento do software em UML, utilizando Visual Paradigm;

2.2 Definições

- **Peça:** Definição abstrata de uma unidade controlada por um jogador;
- **Soldado:** Especialização de **peça**; azul ou vermelha;
- **Oficial:** Especialização de **peça**, azul ou vermelha, marcada com um símbolo;
- **Torre:** Implementação de uma Pilha, composta por empilhamento de **peças**;
- **Posição:** Espaço do **tabuleiro**, podendo conter ou não uma **torre;**
- **Tabuleiro:** Matriz 7x7, composto por 49 **posições**;
- **Jogador:** Representação do Usuário durante o jogo;

2.3 Regras do Jogo

- Cada **jogador** começa com 11 **peças**, definidas pela cor do **jogador**, organizadas de modo idêntico ao jogo de damas;
- O controle e o movimento das **torres** é definido pela **peça** que está no topo;
- Os **soldados** só podem se mover em diagonal para frente, em até 1 **posição**, na direção vertical no sentido lado adversário;
- Os **oficiais**, podem se movimentar nas diagonais, em até 1 **posição**, em ambos os sentidos da direção vertical;
- Sempre haverá possibilidade de captura quando uma **peça**/**torre** aliada se encontra na diagonal com uma **peça**/**torre** inimiga e houver uma posição livre atrás da **peça**/**torre** inimiga, o movimento seria um “salto” sobre ela;
- Sempre que uma **torre**/**peça** for realizar o movimento de captura em uma **torre** adversária, o topo da **torre** inimiga é colocada no fundo da **torre** que a capturou;
- Em caso de uma ou mais possibilidades de captura de uma **peça**/**torre** inimiga, o movimento de captura é obrigatório;
- O jogo é finalizado das seguintes maneiras: quando algum dos jogadores não conseguir movimentar suas peças, ficando assim travado e perdendo a partida; O jogador pode optar por desistir da partida; E quando o jogador captura todas as peças de seu oponente, resulta em vitória, destinada ao jogador que capturou todas as peças do oponente;

3. Requisitos de Aplicação

3.1 Requisitos Funcionais

[RF1] Iniciar Partida

O software deve apresentar, em seu menu, a opção de iniciar uma partida.

[RF2] Montar Tabuleiro

Assim que iniciada a partida, o sistema deverá criar e organizar o tabuleiro de acordo com as regras do jogo previamente estabelecidas.

[RF3] Realizar Jogada

O software deverá permitir ao jogador realizar a sua jogada no turno, mas ele deve se limitar em:

- Selecionar apenas **posições** válidas (quando a **posição** não está vazia, quando está ocupada com **peças**/**torres** do domínio do jogador e no caso de captura, quando apenas as que possuem essa possibilidade de movimento);
- Selecionar **posições** de destino válidas (quando a **posição** está vazia para mover ou quando é a **posição** da **peça**/**torre** a ser capturada)

[RF4] Verificar Jogo

Após o movimento de um dos jogadores a aplicação deve checar se existe a condição para se finalizar o jogo, caso não exista, ele passa a vez de jogar ao outro jogador.

[RF5] Desistir

A qualquer momento durante o jogo, o programa deverá oferecer a escolha ao usuário de abandonar a partida através do botão “DESISTIR”.

[RF6] Finalizar Jogo

Quando o jogador desiste, ou perde/ganha a partida, cabe ao programa encerrar corretamente o jogo e exibir o vencedor.

3.2 Requisitos Não Funcionais

[RN1] Especificações

Projeto feito em python 3 (qualquer versão da 3), organizado através do uso da ferramenta projects do Github e versionado pelo Git.

[RN1] Portabilidade

O programa deve ser funcional em qualquer máquina que suporte e tenha um interpretador python 3

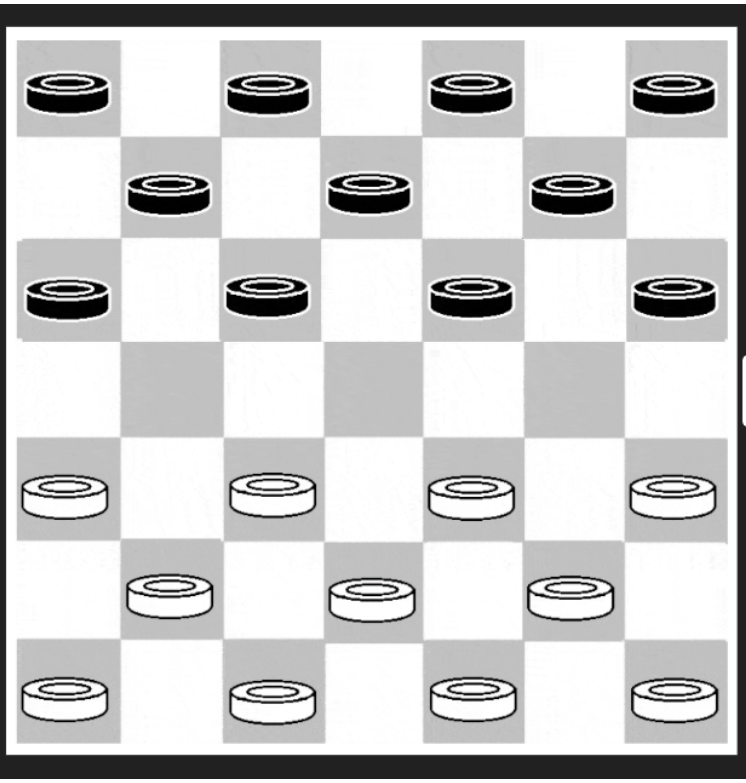
[RN1] Tipo de Arquitetura

Utilizamos a proposta de um software centralizado.

[RN1] Interface Gráfica

Interface interativa, tela com tabuleiro iterativo, através de “Clicks”, Menu principal com os desenvolvedores e telas de vitória de cada jogador.

LASCA



DESISTIR?

FIM DE JOGO!
VENCEDOR VERMELHO

Clique em qualquer lugar para continuar