

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO**

**HENRIQUE PEREIRA RAMOS
JOÃO PEDRO THEODORO
LUCAS DAVI CASCAES BRENA**

**Especificação de Requisitos
do jogo Santorini**

FLORIANÓPOLIS, 2023.

Índice

Índice	2
1. Introdução	3
1.1. Objetivo do desenvolvimento	3
1.2. Definições, abreviaturas	3
1.3. Referências	3 2.
Visão geral do sistema	3
2.1. Arquitetura de software.....	3
2.2. Premissas.....	3 3.
Requisitos de aplicação	3
3.1. Requisitos funcionais	3
3.2. Requisitos não funcionais	4
4. Apêndice	4
4.1. Objetivo do jogo	4
4.2. Regras	5
4.2.1 Posicionando suas peças	5
4.2.2 Movimentação	5
4.2.3 Construção	5
4.2.4 Vencer o jogo	5

Versão	Autores	Data Ação
2.0	Identificados na capa	18/09/2023 Reformulação do documento

1. Introdução

1.1. Objetivo do desenvolvimento

Desenvolver o jogo de tabuleiro "Santorini" de forma distribuída para dois jogadores

1.2. Definições, abreviaturas

- RF: requisito funcional
- RNF: requisito não funcional

1.3. Referências

- Site do jogo (é possível acessar o manual do jogo através do link):
<https://www.mundogalapagos.com.br/jogo-de-tabuleiro-santorini/produto/SAN001>
- Vídeo apresentando o jogo:
<https://www.youtube.com/watch?v=98hO2eTpvDw&t=1s>

2. Visão Geral do sistema

2.1. Arquitetura de software

Cliente-servidor distribuído

2.2. Premissas

- O programa deve ser implementado em Python.
- O programa usará como suporte à execução distribuída o framework PyNetGames.
- A modelagem será realizada utilizando UML2.

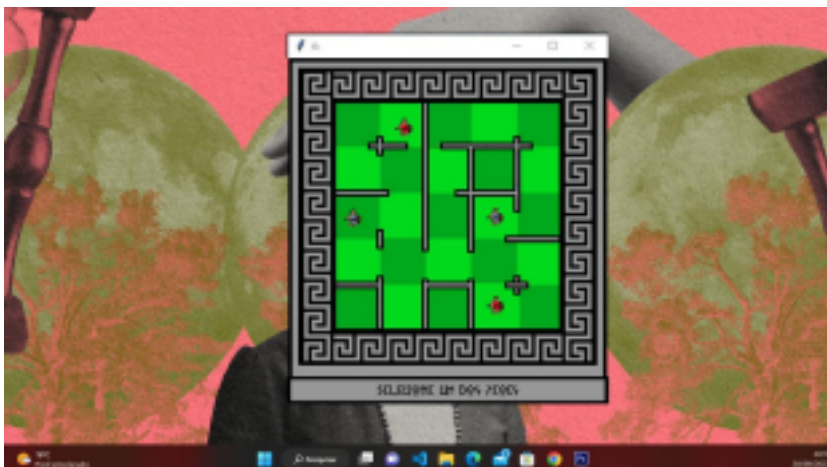
3. Requisitos de aplicação

3.1. Requisitos funcionais

- **RF 1 - Iniciar o programa:** Assim que iniciado, o programa deve mostrar ao usuário o menu inicial, bem como todas as opções disponíveis.
- **RF 2 - Encontrar partida:** Quando iniciado o jogo, o programa deve conectar 2 jogadores utilizando o servidor do PynetGames.
- **RF 3 - Procedimento de lance:** o programa deve suportar o procedimento de lance dos usuários, por meio do click do botão esquerdo do mouse a cada turno.
- **RF 4 - Analisar as células adjacentes ao construtor:** Assim que o jogador selecionar/mover um construtor, o programa deve analisar as células adjacentes à posição atual do construtor, constatando se estão livres ou ocupadas por uma construção ou por outro construtor.
- **RF 5 - Processar Jogada:** O programa deve ser capaz de processar as jogadas realizadas através da ação de click do jogador.
- **RF 6 - Marcação de célula:** O programa deve, após cada click, marcar a última célula com a qual o jogador interage. Cada evento de click é marcado na tela, mesmo sendo uma jogada inválida.
- **RF 7 - Encerrar a rodada:** Após o jogador mover o construtor e construir um andar de uma edificação, o programa deve encerrar a rodada.
- **RF 8 - Analisar a situação do jogo:** Durante a rodada, o programa deve avaliar se algum dos construtores chegou ao terceiro andar de uma construção e/ou se algum dos construtores está impossibilitado de se mover. Neste caso, a partida deve ser encerrada.
- **RF 9 - Passar a rodada:** Caso nenhum dos jogadores tenha sido vitorioso, o programa deve passar a rodada ao próximo jogador.
- **RF 10 - Definir o vencedor:** A partir do requisito 8, o programa deve definir qual jogador foi o vencedor da partida (Caso o construtor tenha chegado ao terceiro andar de alguma construção, ele será o vencedor. Se o jogador estiver impossibilitado de se movimentar e / ou construir, então ele perde, e o outro jogador será o vencedor).

3.2. requisitos não funcionais

- RNF 1 - O programa deve ser implementado em Python.
- RNF 2 - Especificações do projeto devem ser baseadas em UML2.
- RNF 3 - A tecnologia de interface gráfica para usuário deve ser baseada em Tkinter.
- RNF 4 - O programa deve ter uma interface gráfica única compartilhada pelos usuários em caso de partidas entre usuários.
- RNF 5 - Uso do software Visual Paradigm para as modelagens de diagramas UML.
- RNF 6 - Deve-se tomar como base para a criação da interface gráfica a imagem[1] abaixo



[1]:simulação da interface gráfica do jogo rodando no Windows.

4. Apêndice

4.1. Objetivo do jogo

O objetivo do jogo é construir e chegar ao terceiro andar de uma construção antes do seu adversário.

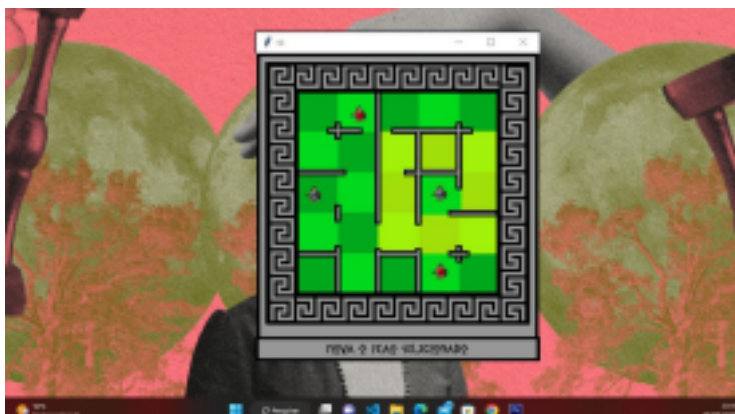
4.2. Regras

4.2.1. Posicionando suas peças

No começo do jogo, cada jogador deve posicionar seus 2 construtores no tabuleiro. Os construtores podem ser posicionados em qualquer casa livre do tabuleiro.

4.2.2. Movimentação

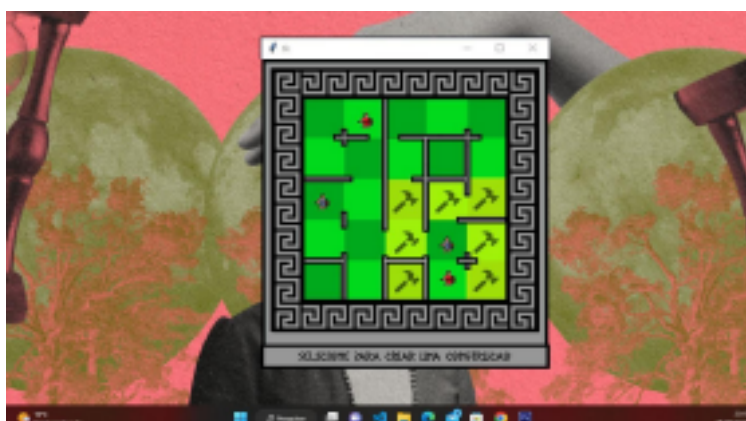
No início de cada turno, o jogador deve mover um dos construtores para qualquer casa adjacente. O construtor pode subir 1 andar por vez, e descer a quantidade de andares que desejar.



[2]: Exemplo de rodada movimentação

4.2.3. Construção

Após movimentar o construtor, o jogador deve construir 1 andar em alguma casa adjacente ao construtor movimentado. O jogador pode começar uma nova construção, ou incrementar um andar em alguma construção já existente, mesmo que esta tenha sido criada por outro jogador. Cada construção pode chegar a até 4 níveis, sendo o quarto nível um "domo", que completa a construção e impede o jogador de acessar o terceiro nível.



[3]: Exemplo rodada de construção

4.2.4. Vencer o jogo

Vence o jogo o jogador que conseguir chegar com qualquer um de seus construtores no terceiro andar de alguma construção, seja esta construída por ele ou não. Outra maneira de vencer é caso o jogador adversário esteja com ambos os construtores presos em algum ponto do tabuleiro impossibilitado de concluir uma completa (se movimentar e/ou construir).