

Projeto: Isolation

Especificação de Requisitos de Software

Versão 2.0

Data de criação: 06/05/2018

Versão	Autor	Data	Ação
1.0	Hugo Oredes Agapito	26/03/2018	Estabelecimento dos Requisitos
2.0	Hugo Oredes Agapito	06/05/2018	Retirado as regras do jogo da sessão sobre regras de negócio e colocadas na introdução.

Conteúdo:

1. Introdução
2. Visão Geral
3. Requisitos da Aplicação
4. Esboço da Interface Gráfica

1 Introdução

1.1 Objetivo do Desenvolvimento

Desenvolvimento de um programa distribuído que possibilite de dois usuários jogarem entre si o jogo Isolation.

1.2 Definições Abreviaturas

Casa Livre: Uma posição no tabuleiro que não possui peça do adversário ou marcador.

Casa Ocupada: Uma posição no tabuleiro que possui peça do adversário ou marcador.

Marcador: Peça neutra que é colocada pelos jogadores em sua rodada em uma casa livre.

1.3 Referencias

[https://en.wikipedia.org/wiki/Isolation_\(board_game\)](https://en.wikipedia.org/wiki/Isolation_(board_game))

1.4 Regras do Jogo

R1: O jogo é realizado entre dois jogadores em um tabuleiro 7x7 (49 posições)

R2: O jogador 1 controla a peça Branca e o jogador 2 controla a peça Preta.

R3: A peça Branca inicia a partida na posição [1, 4] e a peça Preta na posição [7, 4].

R4: O jogador 1 inicia a partida.

R5: Em sua vez de jogar, primeiro o jogador deve movimentar sua peça para uma casa livre adjacente (horizontal, vertical, diagonal)

R6: Após movimentar sua peça o jogador precisa selecionar uma casa livre qualquer no tabuleiro para colocar um marcador na mesma deixando ela ocupada e assim impedindo que uma peça se movimente para ela.

R7: Após marcar uma casa livre é a vez do outro jogador assim repetindo o processo de

jogada, assim a cada turno o tabuleiro vai ficando mais ocupado dificultando a movimentação do adversário.

R8: Perde o primeiro jogador que no início de seu turno não conseguiu movimentar sua peça por ter todas as casas adjacentes ocupadas.

2 Visão Geral

2.1 Arquitetura da Aplicação

Programa orientado a objetos, e um sistema distribuído do tipo cliente-servidor.

2.2 Premissas de Desenvolvimento

- O programa deve apresentar uma interface gráfica bidimensional
- O programa deve ser implementado em Java, devendo executar em qualquer plataforma que disponha da máquina virtual Java
- O programa deve obrigatoriamente executar distribuído, com o suporte para aplicações distribuídas, NetGamesNRT

3 Requisitos da Aplicação

3.1 Requisitos Funcionais.

Requisito Funcional 1 – Conectar: A aplicação deve apresentar a opção de menu “Conectar”. Ao ser selecionada o programa primeiro verifica se já está conectado, em caso negativo, a aplicação solicita o nome do Jogador e tenta estabelecer conexão com servidor informando falha para o usuário.

Requisito Funcional 2 – Desconectar: O programa de ter a opção de menu “Desconectar” que primeiro verifica se já não está desconectado. Em caso negativo envia solicitação de desconexão ao NetGames.

Requisito Funcional 3 – Iniciar Partida: O programa deve apresentar a opção de menu “Iniciar Partida”, que realiza as seguintes ações em ordem.

- Primeiro verifica se já está conectado informando ao usuário em caso negativo.
- Verifica se há partida em andamento, em caso positivo questiona ao usuário se quer interromper a partida em andamento.
- Caso aceite ou não haja partida em andamento, envia solicitação de Iniciar Partida e fica pronto para receber solicitação de outro jogador.

Requisito Funcional 4 – Receber Iniciar Partida: O programa primeiro verifica se está pronto para receber iniciar partida, em caso positivo cria o tabuleiro, solicita o id do adversário, determina a ordem dos Jogadores, coloca as peças dos Jogadores nas posições iniciais e informa ao primeiro Jogador que é sua vez.

Requisito Funcional 5 – Procedimento de Lance: O programa deve suportar o procedimento de lance através de um click com o botão esquerdo do mouse sobre uma posição do tabuleiro sendo necessário dois clicks em posições diferentes do tabuleiro o primeiro para movimentar a peça conforme R5 e segundo para bloquear uma posição livre conforme R6, passando assim a vez conforme R7 e verificando se venceu a partida conforme R8. A cada click valido o tabuleiro deve atualizar e enviar o lance para o

adversário.

Requisito Funcional 6 – Receber Jogada: O programa deve receber cada lance do adversário e atualizar o tabuleiro de acordo com o tipo de lance efetuado. Se for movimentação mostrar nova posição da peça do adversário e for bloqueio e passe a vez mostrar nova posição bloqueada e informar se perdeu a partida.

3.2 Requisitos Não Funcionais

Requisito não funcional 1 – Especificação de Projeto: Código em linguagem Java e especificação de projeto baseada em UML 2;

Requisito não funcional 2 – Interface Gráfica para Usuário: A aplicação deverá ter interface gráfica única, compartilhada pelos usuários;

Requisito não funcional 3 – Símbolos dos Jogadores: As peças dos jogadores serão, identificadas com a cor preta e do outro jogador com a cor branca;

Requisito não funcional 4 – Tecnologia de Interface Gráfica para Usuário: Interface Gráfica baseada em Java, biblioteca gráfica Swing.

4. Esboço da Interface Gráfica

