**Especificação de Requisitos de Software**

Versão 1.3

Data de criação: 28/08/2018

|  |  |  |  |
| --- | --- | --- | --- |
| Versão | Autores | Data | Ação |
| 1.0 | Camila dos Reis  Pablo Duz  Thais Goulart | 28/08/2018 | Estabelecimento de requisitos |
| 1.1 | Camila dos Reis  Pablo Duz  Thais Goulart | 14/09/2018 | Atualização de requisitos |
| 1.2 | Camila dos Reis  Pablo Duz  Thais Goulart | 16/10/2018 | Correção de requisitos |
| 1.3 | Camila dos Reis  Pablo Duz  Thais Goulart | 04/12/2018 | Correção e atualização de requisitos; entrega final |

**Conteúdo:**

1. Introdução

2. Visão Geral

3. Requisitos da Aplicação

4. Esboço da Interface Gráfica

**1 Introdução**

**1.1 Objetivo do Desenvolvimento**

**Objetivo:** Desenvolvimento de um programa distribuído que possibilite dois usuários jogarem Picross de modo cooperativo.

**O jogo:** O picross (também conhecido como nonograma) é um jogo que tem por objetivo formar imagens descobrindo seus pixels através de números e coordenadas. É um jogo cooperativo e por isso deve ter dois jogadores, ambos os jogadores irão construir a imagem. O jogo possuirá diversos níveis de dificuldade.

**1.2 Definições Abreviaturas**

**Quadrado vazio:** Uma posição no tabuleiro que está em branco, ou seja, não foi pintada.

**Quadrado preenchido:** Uma posição no tabuleiro que foi colorida.

**1.3 Referências**

<https://en.wikipedia.org/wiki/Nonogram>

**1.4 Regras do Jogo**

**R1:** O jogo é realizado entre dois jogadores em um tabuleiro 5x5 (25 posições);

**R2:** Os números pista mostram quantos quadrados podem ser preenchidos em cada linha ou coluna.

**R3:** Sempre que houver mais de um número numa mesma linha ou coluna, como por exemplo 3 2 (três e dois), quer dizer que naquela linha ou coluna haverá uma sequência de 3 quadrados preenchidos separados por pelo menos um quadrado vazio e mais dois quadrados preenchidos.

**R4:** Cada jogador tem a sua vez, clicando numa posição da linha ou coluna, selecionando então a posição desejada.

**R5**: Após 3 (três) jogadas o turno do jogador encerra automaticamente e a vez é passada para o outro colaborador em estado de espera.

**R6:** É possível desfazer as jogadas, durante seu próprio turno, ao clicar com o botão esquerdo do mouse em uma posição preenchida.

**R7:** O jogo termina quando os jogadores formarem uma imagem corretamente.

**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 intuitiva com a utilização de elementos gráficos bidimensionais.
* O programa deve ser implementado em Java, portanto poderá ser executado em qualquer plataforma que possua uma máquina virtual Java.
* O programa deve obrigatoriamente executar de forma distribuída, com o suporte para

aplicações também distribuídas, NetGamesNRT.

**3 Requisitos da Aplicação**

**3.1 Requisitos Funcionais.**

**Requisito Funcional 1 – Conectar:** A aplicação possuirá a opção “Conectar” em seu menu. Quando selecionada, o programa vai verificar se já está conectado; caso a resposta seja positiva irá mostrar a mensagem “Conexão já efetuada” e caso a resposta seja negativa, a aplicação solicita o nome do jogador e tenta montar a conexão com servidor, se a conexão com servidor não for possível, mostrará a mensagem “Não foi possível conectar” e se for possível realizar a conexão mostrar mensagem “Conexão efetuada com sucesso”.

**Requisito Funcional 2 – Desconectar:** A aplicação possuirá a opção “Desconectar” em seu menu, na qual primeiro verifica se já não está desconectado; se estiver conectado, a aplicação solicita a desconexão ao NetGames.

**Requisito Funcional 3 – Iniciar Partida:** A aplicação deve possuir a opção de menu “Iniciar Partida”, que realiza as ações subsequentes:

• Faz primeiro a verificação se o jogador está conectado e informa caso não esteja;

• Verifica se a partida se encontra em andamento, e caso o contrário, pergunta ao jogador se deseja parar a partida;

• Se o jogador aceitar ou se não houver uma partida em andamento, a aplicação envia o pedido de Iniciar Partida e fica disponível para receber a solicitação de um outro jogador.

**Requisito Funcional 4 – Receber Iniciar Partida:** A aplicação verifica se está pronta para receber Iniciar Partida, e se sim, mostra o tabuleiro já criado, pede o id do adversário, determina a ordem dos jogadores e informa ao primeiro que é a sua vez.

**Requisito funcional 5 – Realizar Jogada:** A aplicação deve apresentar o jogador da rodada e permitir que este clique em cima da posição que deseja marcar no tabuleiro. Após marcar a posição desejada, verifica se o turno do jogador está completo.

**Requisito funcional 6 – Informação do estado da partida:** Para cada tentativa de procedimento de lance dos usuários o novo estado da partida deve ser exibido na interface.

**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, partilhada pelos usuários;

**Requisito não funcional 3 – Símbolos dos Jogadores:** As posições marcadas pelos jogadores serão identificadas com a cor preta;

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

**4. Esboço da Interface Gráfica**

