**Mata-Mata Bolas de Gude**

**Especificação de Requisitos de software**

**Versão 1.3**

06/07/2017

|  |  |  |  |
| --- | --- | --- | --- |
| **Versão** | **Autor(es)** | **Data** | **Ação** |
| 1.0 | Clayton R. Veras e Lucas Mauro de Souza | 01/04/2017 | Estabelecimento dos requisitos. |
| 1.1 | Clayton R. Veras e Lucas Mauro de Souza | 22/04/2017 | Correções de objetivo geral  e requisitos. |
| 1.2 | Clayton R. Veras e Lucas Mauro de Souza | 12/05/2017 | Correção de objetivo geral, descrição do jogo e adição de requisitos não funcionais. |
| 1.3 | Clayton R. Veras e Lucas Mauro de Souza | 06/07/2017 | Correção em introdução, requisitos de software, requisitos não funcionais e esboço de interface gráfica. |

**Conteúdo:**

1. Introdução;

2. Visão Geral;

3. Requisitos de software.

4. Esboço de interface gráfica

**1. Introdução**

**Objetivo geral**: Desenvolver um software orientado a objetos distribuído, que implemente um jogo de bolas de gude com interface gráfica em 2D, utilizando os melhores conceitos e práticas de análise, projeto e desenvolvimento de sistemas.

**O Jogo**: O ambiente do jogo será um campo retangular com círculo de tamanho pré-definido ao seu centro. Neste circulo serão posicionadas 13 bolas de gude todas próximas umas às outras. Cada jogador possuirá 1 bola de gude que sempre estará em sua posse. O jogo é composto por turnos. O jogador que solicitar início de partida será o primeiro a jogar e seu adversário será o segundo. Durante seu turno cada jogador irá efetuar apenas um lance com sua bola de gude, com o objetivo de acertar as bolas posicionadas dentro do círculo e removê-las deste. Como as bolas posicionadas inicialmente estão próximas e também porque estas podem ricochetear umas nas outras em um efeito cascata, o jogador pode também acertar duas bolas ao mesmo tempo, dependendo do ângulo da jogada, sendo assim, durante o lance pode ser que o jogador consiga remover mais de uma bola do círculo. Porém, independentemente de acertar ou errar o lance, após sua jogada, seu turno se encerrará. Após removidas todas as bolas de gude do círculo, vence o jogador que tiver removido mais bolas.

**Regras**:

* No mínimo dois jogadores devem estar conectados ao jogo;
* O jogador só pode efetuar um lance em seu turno;
* O jogador que remover o maior número de bolas de gude do círculo após a última bola ser removida será o vencedor;
* O turno do jogador acabará considerando que ele acerte uma bola de gude ou não;
* A bola somente é considerada removida pelo jogador quando ultrapassar a demarcação do círculo;
* O jogador pode remover no mesmo lance uma ou mais bolas de gude;
* O jogador efetuará o lance posicionado fora do círculo;
* Se um jogador desistir da partida, o jogador remanescente será o vencedor;

**2. Visão Geral**

**Arquitetura do programa:**

* Programa orientado a objetos;
* Aplicação distribuída, usando o modelo de comunicação cliente-servidor.

**Premissas de desenvolvimento:**

* O software utilizado para a modelagem UML do programa deve ser o Visual Paradigm;
* O programa deve ser implementado em Java;
* A aplicação deverá apresentar arquitetura cliente/servidor utilizando a ferramenta NetGamesNRT para que rode como uma aplicação distribuída;

**3. Requisitos de Software**

**3.1 Requisitos funcionais:**

**Requisito funcional 1: Conectar -** O software deve apresentar um botão para que o usuário possa se “Conectar”, estabelecer conexão com o servidor NetGames;

**Requisito funcional 2 : Desconectar -** O programa deve apresentar um botão para que o usuário possa se desconectar do servidor, sendo que este sairá de qualquer partida em andamento;

**Requisito funcional 3 : Iniciar** -O programa deve apresentar um botão que permite ao usuário iniciar uma partida do jogo;

**Requisito funcional 4 : Procedimento de lance** - O programa deve possibilitar que o usuário selecione força e ângulo para lançar sua bola de gude contra as bolas posicionadas dentro do círculo. Após lançada a bola deverá ser validada a remoção de bolas do círculo, caso uma ou mais bolas sejam removidas, deve-se verificar se ainda existem bolas em campo para serem removidas e caso não existam, deve-se verificar qual dos jogadores é o vencedor.

**3.2 Requisitos não funcionais:**

**Requisito não funcional 1 :** O programa deve apresentar interface gráfica bidimensional;

**Requisito não funcional 2** : O programa deve executar em qualquer plataforma que disponha máquina virtual Java;

**4. Esboço de interface gráfica**





