

ESPECIFICAÇÃO DE REQUISITOS

Projeto – Battle of Gods

Especificação de requisitos de software

Versão 4.0

10/07/2016

Versão	Autores	Data	Ação
1.0	Guilherme Noronha, Mateus Rissi e Rafael Rocker	17/04/2016	Estabelecimento dos requisitos
2.0	Guilherme Noronha, Mateus Rissi e Rafael Rocker	01/05/2016	Correção de incoerências
3.0	Guilherme Noronha, Mateus Rissi e Rafael Rocker	01/06/2016	Correção de incoerências
4.0	Guilherme Noronha, Mateus Rissi e Rafael Rocker	11/07/2016	Correção de incoerências

Conteúdo:

1. Introdução
2. Visão Geral
3. Requisito de Software
4. Esboço da Interface Gráfica

1. Introdução:

1.1. Objetivo:

- 1.1.1. Desenvolvimento de um jogo por turno, que suporte a modalidade usuário vs. Usuário.

1.2. Referências:

- 1.2.1. <https://pt.wikipedia.org/wiki/Pok%C3%A9mon_Red_e_Blue>
- 1.2.2. <[https://en.wikipedia.org/wiki/Medabots_\(video_game\)](https://en.wikipedia.org/wiki/Medabots_(video_game))>
- 1.2.3. http://bulu-monster.wikia.com/wiki/Bulu_Monster

2. Visão Geral

2.1. Arquitetura do Programa:

- 2.1.1. Programa orientado a objetos.
- 2.1.2. Desenvolvido com linguagem Java.
- 2.1.3. Modelo cliente-servidor.

2.2. Premissas de Desenvolvimento:

- 2.2.1. Utilizar o framework “NetGamesNRT” para realizar a interação usuário versus usuário.
- 2.2.2. Implementação em Java. O programa deve ser executado em qualquer plataforma que disponha da máquina virtual Java.
- 2.2.3. Interface bidimensional.
- 2.2.4. Sistema de processamento distribuído.
- 2.2.5. O programa deve permitir que sejam disputadas partidas entre dois usuários, garantindo que os jogadores não possam executar movimentos inválidos de acordo com as regras do jogo e que no final exista apenas um vencedor.

2.3. Regras do Jogo:

Battle of Gods tem como personagens deuses de várias crenças. O objetivo do jogo é derrotar a dupla de “deuses” inimigos. A primeira etapa é definir seu time, escolhendo 2 (dois) deuses de um catálogo com todos os personagens disponíveis. Após cada jogador definir seu time, a batalha pode ser iniciada.

As batalhas são baseadas em turno. O jogador irá realizar uma ação com cada um de seus dois personagens a cada turno, esta ação será uma das habilidades do personagem ou um bloqueio (faz com que o personagem em questão fique na defensiva e receba metade do dano que iria levar caso fosse atacado pelo inimigo). Cada personagem possui 4 habilidades (únicas do personagem) que, podendo variar entre ataques físicos e mágicos, podem realizar diversos efeitos como congelar, paralisar, queimar, envenenar; e efeitos que podem aumentar os atributos de aliados ou diminuir os atributos dos inimigos.

Os personagens possuem atributos que lhes dão vantagens e/ou desvantagens numa batalha. Tais atributos são: Vida; Ataque; Defesa; Velocidade; Ataque Mágico; Defesa Mágica;

O jogador que atacará primeiro é definido de maneira aleatória. A partida possui um único “round”. Um round chegará ao fim quando ambos os personagens de um dos jogadores fiquem com 0 (zero) pontos de vida, levando este jogador a derrota.

3. Requisito de Software

3.1. Requisitos Funcionais:

- 3.1.1. Iniciar Partida: o programa deve apresentar a opção de menu “Iniciar” para o início de uma partida.
- 3.1.2. Conectar: o programa deve apresentar a opção de menu “Conectar” para conectar ao servidor.
- 3.1.3. Desconectar: o programa deve apresentar a opção de menu “Desconectar” para desconectar do servidor.
- 3.1.4. Slots: o programa deve apresentar dois botões para que o usuário escolha seus personagens.
- 3.1.5. Procedimento de lance: o programa deve permitir que o jogador da rodada realize as ações que deseja, ações que não violem as regras do jogo, após a execução de cada ação, o programa deve verificar se um dos jogadores é o vencedor daquela partida.
- 3.1.6. Sair: o programa deve apresentar a opção “Sair”, permitindo o encerramento da aplicação.

3.2. Requisitos Não-Funcionais:

- 3.2.1. Especificação de projeto: devem ser produzidas especificações de projeto baseadas em UML.
- 3.2.2. Interface gráfica: o programa deve apresentar uma única interface gráfica, compartilhada pelos usuários.
- 3.2.3. Símbolos dos personagens: cada personagem possuirá uma “Sprite/2Dart”.
- 3.2.4. Tecnologia da interface gráfica para usuário: a interface gráfica deve ser baseada em Java-Swing.

4. Esboço da Interface Gráfica

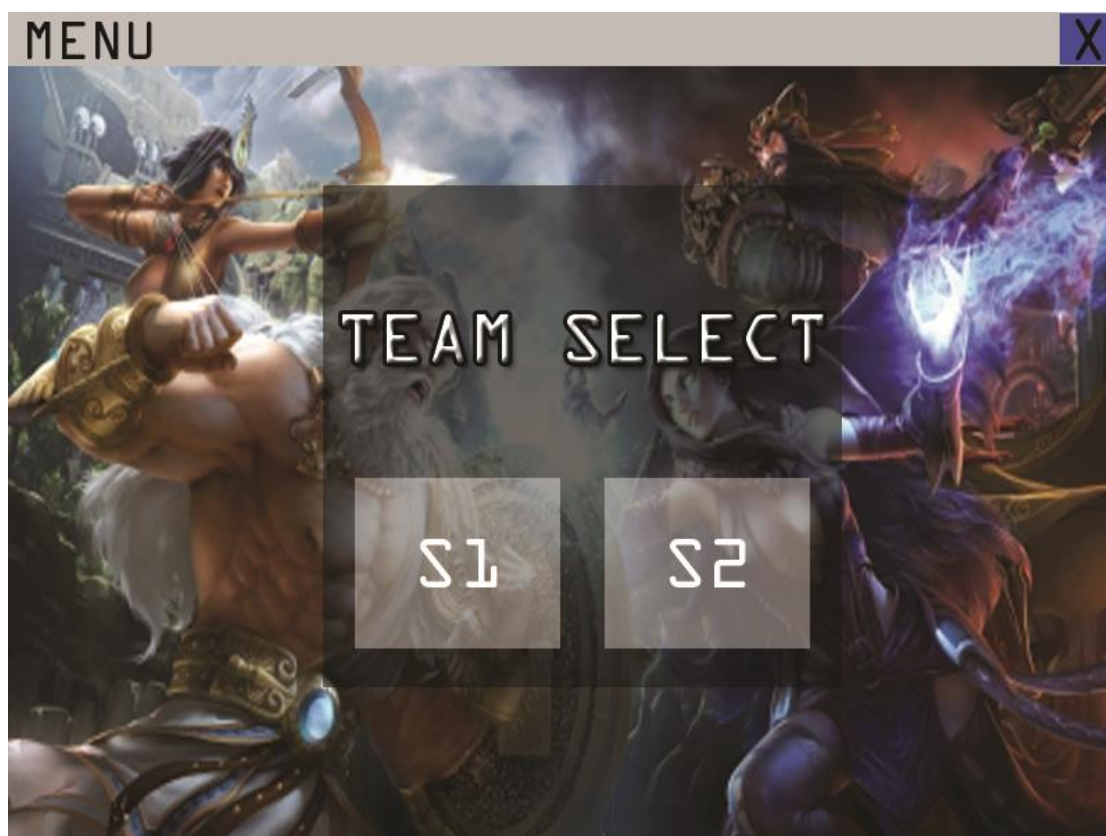


Imagem 1 – Tela Inicial

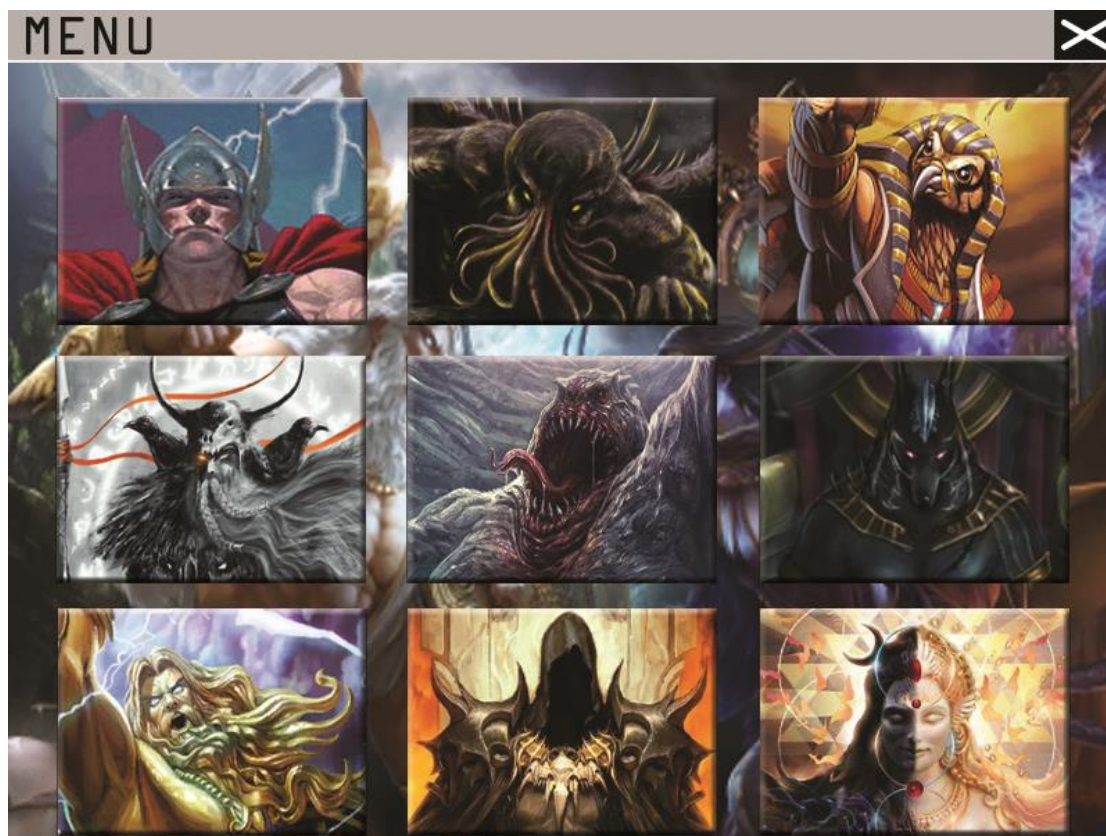


Imagem 2 – Tela Catálogo de Personagens



Imagem 4 – Tela de Batalha