**ESPECIFICAÇÃO DE REQUISITOS**

**Jogo Tribal Wars**

**Especificação de requisitos de software**

Versão 1.0

Data: 08/04/2015

|  |  |  |  |
| --- | --- | --- | --- |
| **Versão** | **Autores** | **Data** | **Ação** |
| 1.0 | Lucas F. da Costa  Henrique Prandi | 08/04/15 | Estabelecimento dos requisitos |

**Conteúdo**:

1. Introdução;
2. Visão geral;
3. Requisitos de software;
4. Esboço da interface gráfica;

**1. Introdução**

**1.1. Objetivo**: Desenvolvimento de um programa que suporte partidas do jogo Tribal Wars (regras abaixo) em rede na modalidade usuário vs. usuário.

**1.2. Referências:**

< http://www.tribalwars.com.br/>

< http://pt.wikipedia.org/wiki/Clash\_of\_Clans>

**2. Visão geral**

**2.1. Arquitetura do programa:**

· Programa orientado a objetos, escrito na linguagem Java;

· Aplicação distribuída, cliente-servidor.

**2.2. Premissas de desenvolvimento:**

· O programa deve apresentar uma interface bidimensional;

· O programa deve utilizar o framework “NetGamesNRT” para realizar a interação usuário *versus* usuário;

· O programa deve ser implementado em Java, executando em qualquer

plataforma que disponha da máquina virtual Java.

**2.3 Regras do jogo**:

O objetivo do jogo é reduzir os pontos de vida da tribo adversária para 0 ou fazer o adversário render-se.

A cada turno o jogador pode escolher realizar uma dentre as seguintes ações:

* Coletar Recursos
* Fortalecer Muralha
* Gerar Tropas
* Atacar (caso possua tropas)
* Render-se

Existem dois tipos de recursos: comida e madeira, que podem ser obtidos através da ação coleta de recursos. A coleta de recursos irá gerar uma distribuição aleatória de quantidade de comida e madeira, assim, cabe ao jogador tomar as melhores decisões baseado em sua atual situação de recursos.

A muralha serve como uma espécie de escudo, ela também possui pontos de vida. Enquanto os pontos de vida da muralha forem superiores à 0, ela irá tomar todo o dano proveniente de ataques. A única maneira de incrementar a quantidade de pontos de vida da muralha é realizando a ação de fortalecê-la. Após a muralha ser totalmente destruída, o jogador precisa esperar alguns turnos para poder reconstruí-la.

Para gerar tropas serão consumidas unidades do recurso comida, já para fortalecer a muralha serão consumidas unidades do recurso madeira.

Para poder atacar o jogador atacante precisa possuir tropas, essas tropas serão enviadas ao combate para lutar até a morte e causar uma quantidade de dano na muralha ou na vila (caso a muralha esteja destruída) que varia de acordo com a força e a quantidade dos soldados, além, é claro, de um pouco do fator sorte.

**3. Requisitos de Software**

**3.1. Requisitos Funcionais:**

**3.1.1. Conectar:** O programa deve ter a opção de menu *“Conectar”*

para estabelecer uma nova conexão com o servidor, nesta operação é definido a

identificação dos jogadores. O sistema confirmará se a conexão foi estabelecida ou não;

**3.1.2. Iniciar:** O programa deve apresentar a opção de menu

*“Iniciar”* para início de uma nova partida. A opção só estará habilitada quando o jogador estiver conectado.

**3.1.3. Fechar**: O programa deve apresentar a opção de ser fechado a qualquer momento, apresentando uma mensagem de confirmação, permitindo confirmar ou cancelar.

**3.1.4. Escolha de quem inicia**: O programa deve escolher aleatoriamente o jogador que dará início a partida.

**3.1.5. Procedimento de escolha da ação**: Em seu turno o jogador deve poder escolher qual ação irá realizar. As ações poderão ser: construir tropas, coletar recursos, fortalecer muralha, atacar, render-se. Após essa escolha ocorre a troca de turno.

**3.1.6. Procedimento de construir tropas**: Ao escolher a ação de construir tropas o programa irá construir o máximo de tropas possíveis. Cada unidade de tropa, tem um custo de algumas unidades de comida. As tropas geradas serão acrescentadas ao número total de tropas do jogador que realizou a ação.

**3.1.7. Procedimento de coletar recursos**: Ao escolher a ação de coletar recursos o programa irá gerar uma distribuição aleatória de 10 pontos de recurso entre carne ou madeira. Os pontos de recurso gerados serão acrescentados aos recursos atuais do jogador que realizou a ação.

**3.1.8. Procedimento de fortalecer muralha**: Ao escolher a opção de fortalecer muralha o programa irá adicionar alguns pontos de pontos de vida à muralha do jogador que realizou a ação, consumindo unidades de madeira.

**3.1.9. Procedimento de atacar**: Ao escolher a opção de atacar, o programa irá consumir todas as unidades de tropa do jogador que realizou a ação, e irá remover pontos de vida da muralha adversária, caso ela exista. Caso contrário serão removidos pontos de vida diretamente da vila adversária. Caso a muralha exista, porém os pontos de vida a serem removidos, sejam maiores que a vida da muralha, os pontos sobressalentes serão removidos da aldeia adversária. Os pontos de vida a serem removidos, serão calculados de acordo com a quantidade de unidades de tropa, e um multiplicador aleatório.

**3.1.10. Procedimento de render-se**: Em seu turno o jogador pode escolher a opção de render-se, terminando então a partida e dando a vitória ao jogador adversário.

**3.1.11. Procedimento de troca de turno**: Após realizada a ação, o programa irá verificar se o jogador que terá a próxima vez está com 0 pontos de vida. Caso esteja, o sistema declara vitória ao jogador cuja vila está com mais que 0 pontos de vida. Caso contrário, o programa habilita a ação do próximo jogador.

**3.2. Requisitos Não-Funcionais:**

**3.2.1. Especificações do projeto:** Desenvolvido em linguagem Java a partir de uma especificação modelada em UML2.

**3.2.2. Conectividade:** O programa atuará de maneira distribuída (cliente/servidor) utilizando a ferramenta NetGamesNRT.

**3.2.3 Interface gráfica**: A aplicação possui uma única interface gráfica compartilhada pelos dois usuários.

**3.2.4 Tecnologia da interface gráfica**: Será utilizado o pacote SWING do Java.

**3.2.5 Escolha de posição na tela**: Ao ser iniciada a partida, o jogador que clicou em “Iniciar” recebe a aldeia superior, e o outro a inferior.

**4. Esboço da interface gráfica**:

