



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA

CIVILIZATION - LOTR

ESPECIFICAÇÃO DE REQUISITOS DE SOFTWARE

VERSÃO 2.0

DANIEL HENRIQUE KOCK
FILIPE VOGES
MATHEUS LAUREANO

FLORIANÓPOLIS
2017

Sumário

1. Introdução

1.1. Objetivo do desenvolvimento

1.2. Definições e abreviaturas

1.3. Referências

1.3.1. Regras do jogo:

2. Premissas de desenvolvimento

2.1. Interface gráfica

2.2. Implementação

2.3. NetGamesServer

2.4. Tipos de Tropas e suas habilidades

2.4.1 Espadachim

2.4.2 Arqueiro

2.4.3 Cavaleiro

2.4.4 Mago

2.4.5 Herói

2.5. Tipos de Tropas e suas habilidades

2.6. Saga do Anel

2.7. Dinâmica das batalhas

3. Requisitos da aplicação

3.1. Requisitos funcionais

3.1.1. Iniciar uma Partida

3.1.2. Escolher a raça do Jogador

3.1.3. Reforma de Construção

3.1.4. Atacar uma tropa

3.2. Requisitos não funcionais

3.2.1. Linguagem de Programação

3.2.2. Servidor de Jogos e Framework de rede

3.2.3. Especificação de projeto

3.2.4. Interface gráfica para usuário

4. Esboço da Interface

1. Introdução

1.1. Objetivo do desenvolvimento

O objetivo deste desenvolvimento é a criação de um jogo baseado em turnos para dois jogadores, que através do acesso a um servidor, pode ser jogado de qualquer lugar, com o intuito promover entretenimento e interação entre os usuários.

1.2. Definições e abreviaturas

Personagem: Personagens que são controlados pelo jogador para cumprir um ou mais objetivos durante a partida.

Vida : Pontos de vida dos personagens.

Mapa : É o tabuleiro.

1.3. Referências

1.3.1. Regras do jogo:

O LOTR-Civilization é um jogo baseado em turnos ao qual você gerencia exércitos e construções, ele possui como objetivo destruir todas as construções do(s) jogador(es) inimigo(s);

No início do jogo, os jogadores devem escolher suas respectivas raça e o servidor no qual se conectará, e assim o jogo se inicia. Cada jogador é posicionado em um ponto do mapa, com as suas construções devidamente construídas no início da partida, e uma certa quantidade de recursos disponíveis, dando possibilidade do jogador:

- Criar tropas a partir de uma construção;
- Atacar uma tropa ou cidade inimiga com suas tropas;
- Reformar uma construção caso esteja danificada;

Algumas ações possuem uma quantidade de recursos necessários para ser feita. Os jogadores podem ver a ação dos adversários no momento em que o jogador atual

executa uma ação.

Para vencer o jogo o jogador precisa destruir todas as construções do oponente, restando apenas uma cidade no mapa, outra possibilidade de vitória é a por desistência, onde o jogador desistente é declarado como derrotado e o adversário como vitorioso.

2. Premissas de desenvolvimento

2.1. Interface gráfica

O programa apresenta uma interface gráfica bidimensional.

2.2. Implementação

O programa é implementado em Java, sendo executado em qualquer plataforma que disponha de uma máquina virtual Java (JVM).

2.3. NetGamesServer

O programa é distribuído utilizando NetGamesServer.

2.4 Tipos de tropas e suas habilidades

O jogo possui cinco tipos de tropas. Cada uma com vantagens e desvantagens sobre a outra.

2.4.1 Espadachim

O espadachim, que é um modelo de tropa com resistência maior que outras tropas. Tem vantagem de batalha contra cavalaria e desvantagem contra arqueiro.

2.4.2 Arqueiro

O arqueiro é um modelo de tropa com dano maior que as outras tropas que possui vantagem contra os espadachins, e desvantagem contra cavaleiros.

2.4.3 Cavaleiro

E temos o cavaleiro, que pode se locomover mais rapidamente que outras tropas,

e tem vantagem contra arqueiros, e desvantagem contra espadachins.

2.4.4 Mago

O mago possui vantagem contra todas as outras tropas e não possui nenhuma desvantagem, porém, para invocá-lo é necessário derrotar o Gollum em seu jogo de charadas (mais informações no tópico 2.6).

2.4.5 Herói

E como última tropa, temos o herói, essa tropa possui vantagem contra todas tropas e também não possui desvantagem, ele é invocado pela construção principal e cada jogador possui seu herói específico, como podemos ver a seguir.

2.5. Tipos de heróis e sua habilidades

O herói será definido pela raça escolhida pelo jogador pois, cada raça terá seu herói específico.

Se **Humano**, o herói será o **Aragorn**.

Se **Elfo**, o herói será o **Legolas**.

Se **Uruk Hai**, o herói será **Lurtz**.

Se **Orc**, o herói será o **Azog**.

Cada herói tem suas habilidades específicas. O Legolas tem habilidades de arqueiro, Lurtz tem habilidades de espadachim, Azog tem habilidades de cavaleiro, e Aragorn tem todas as habilidades citadas.

2.6. A saga do Anel

O Gollum aparece no meio do mapa. Este possui o Anel do Poder, que dá a ele o poder de se tornar praticamente invisível aos olhos dos jogadores.

Caso um jogador encontre o Gollum com o seu herói, ele poderá tentar obter o poder do anel. Para isso, o jogador terá que responder algumas charadas feitas por Gollum, e caso consiga responder todas, o Gollum foge e deixa cair o anel, o jogador

então consegue o mesmo. Caso não consiga responder às charadas, o Gollum come o herói.

Com este anel, o jogador poderá obter mais uma tropa, o Mago. Essa tropa não possui desvantagem e seu dano é superior a todas as outras tropas, fazendo assim uma tropa bem requisitada pelos jogadores.

2.7. Dinâmica das batalhas

A Tropa atacante irá atacar a tropa defensiva com base no seu dano de ataque e sua vantagem, e a tropa defensiva se defenderá com base na sua resistência.

Dano Causado: O dano causado pela tropa atacante será igual o ataque básico da tropa vezes a vantagem que possui sobre a tropa defensiva menos a resistência da tropa defensiva.

Dano de Retaliação: A tropa de ataque receberá de dano metade do valor total de ataque da tropa defensiva vezes a vantagem da tropa defensiva contra a tropa atacante. Se o dano causado derrotar a tropa defensiva, não haverá dano de retaliação.

Cálculo de vantagem: Esse cálculo é feito com base no tipo da tropa atacante e o tipo da tropa inimiga, então é verificado a vantagem ou desvantagem de uma sobre a outra, como apresentado na tabela abaixo, onde as **linhas** representam o **atacante** e as **colunas** representam o **alvo** e o valor retornado é multiplicado ao dano do atacante/defensor

	Espadachim	Arqueiro	Cavaleiro	Herói	Mago
Espadachim	1	0.3	1.8	1	1
Arqueiro	1.8	1	0.3	1	1
Cavaleiro	0.3	1.8	1	1	1
Herói	1.2	1.2	1.2	1.2	1.2
Mago	1.9	1.9	1.9	1.9	

A tropa que ficar com zero de vida morre.

3. Requisitos da aplicação

3.1. Requisitos funcionais

3.1.1. Iniciar uma partida

Operação: Inicia a preparação de uma partida.

Descrição: Verifica se há dois ou mais jogadores conectados para iniciar uma partida.

Lê: Estado atual dos jogadores.

Modifica: Estado do jogador e da partida.

Envia: Alerta aos jogadores que uma partida está para iniciar.

Resulta: Partida iniciada, jogadores alocados.

3.1.2. Escolher raça do jogador

Operação: Abrir tela de escolha de time

Descrição: O Jogador poderá escolher qual raça jogará.

Lê: Raça escolhida pelo jogador.

Modifica: Estado do jogador de acordo com a raça escolhida.

Resulta: Partida em estado inicial e jogadores alocados em suas cidades.

3.1.3. Reforma de construção

Operação: Reformar uma construção para que a mesma retorne a sua vida original após receber dano

Construções disponíveis: Quartel, Estábulo, Campo de Arquearia, Edifício Principal.

Operação: Ao ser feito a reforma na construção a construção voltará a ter a mesma quantidade de vida que tinha no início do jogo

O HP inicial de cada construção é de 100, exceto o edifício principal que tem 300.

O custo de reforma para cada construção será o de 10 vezes o dano que a construção recebeu.

Ex: se a vida do edifício principal estiver em 270, o preço da reforma é de 300 recursos.

3.1.4. Atacar uma tropa.

Operação: Atacar uma tropa inimiga

Descrição: Verifica se a tropa possui a distância necessária para atacar a tropa (ou construção) inimiga, caso sim inicia o modo de batalha.

Modifica: Modo de jogo e estado das tropas.

3.2. Requisitos não funcionais

3.2.1. Linguagem de Programação

A linguagem de programação utilizada no desenvolvimento do software será Java.

3.2.2. Servidor de Jogos e Framework de rede

O servidor de jogos utilizado no projeto é o servidor multijogadores NetGamesNRT e o framework NetGamesNRTFramework, o qual possibilita abstração dos detalhes da rede para o desenvolvedor

3.2.3. Especificação de projeto

O software apresenta especificações de projeto baseada em UML, segunda versão.

3.2.4. Interface gráfica para usuário

O programa possui interface gráfica única, compartilhada pelos usuários durante a partida. A interface gráfica será desenvolvida em Java-Swing.

4. Esboço da Interface

