

Universidade Federal de Santa Catarina
Departamento de Informática e Estatística – INE
Ine 5417: Engenharia de Software I -Turma 04208A(2015-2)
Professor Ricardo Pereira e Silva

Especificação de requisitos

Projeto: Jogo Worms Armaggedon

Grupo:

Luca Fachini Campelli – 13200659

Luiz Gustavo Coelho Dutra – 13204978

Vilmar José Ramos Júnior – 13203826

Histórico

| Data: | Versão: | Autores: | Progresso: |
|--------------|----------------|--------------------|-------------------|
| 05/09/2015 | 0.1a | Luca, Luiz, Vilmar | Início do projeto |
| 25/11/2015 | 1.0 | Luca, Luiz, Vilmar | Fim do projeto |

1. Introdução

- 1.1. Objetivo do Desenvolvimento.....3
- 1.2. Referências.....3

2. Visão Geral do Sistema

- 2.1. Arquitetura da Aplicação.....3
- 2.2. Descrição do Jogo.....3
- 2.3. Premissas de Desenvolvimento.....4

3. Requisitos de Software

- 3.1. Requisitos Funcionais.....5
- 3.2. Requisitos Não Funcionais.....6

Anexo 1 - Esboço da Interface.....7

1.Introdução:

1.1.Objetivo do desenvolvimento:

Este trabalho é a avaliação final da disciplina Ine 5417 – Engenharia de software I do Curso de Ciências de Computação, ministrada pelo professor Ricardo Pereira e Silva.

O objetivo deste trabalho é desenvolver um programa que implemente o jogo Worms Armaggedon em um sistema NRT de jogo em rede, permitindo a disputa de dois jogadores em rede ou a disputa offline.

1.2.Referências:

<http://wormsarmageddon.team17.com/main.html?page=home&area=welc>

https://pt.wikipedia.org/wiki/Worms_Armageddon

2.Visão geral do Sistema:

2.1.Arquitetura da Aplicação:

Programa Orientado a objetos desenvolvido em linguagem Java de programação.

2.2.Descrição do jogo:

Worms é um jogo onde o objetivo é derrotar todas as minhocas adversárias, se utilizando do terreno, das armas variadas disponíveis. Os cenários são pisos 2D que podem ser destruídos e danificados para criar buracos e caminhos. As minhocas se movem para esquerda e para direita, pulam e utilizam armas para derrotar as outras minhocas. Cada jogador em seu turno possui pontos de ação para jogar, ao fim dos pontos de ação o jogador passa a vez ao outro. O vencedor é declarado quando todas as minhocas do outro jogador chegarem a 0 ponto de vida. Se todas as minhocas forem derrotadas ao mesmo tempo, haverá um empate.

Em seu turno um jogador pode:

- 1.Mover a minhoca selecionada e posicioná-la no mapa
- 2.Utilizar uma arma do arsenal apenas uma vez, tentando atingir as minhocas adversárias.

Se ao final de um turno todas as minhocas de um jogador estiverem sem pontos vida, o outro jogador é declarado vitorioso.

Todas as armas e ferramentas utilizadas são limitadas exceto pela bazuca normal, que é ilimitada.

2.3.Premissas de Desenvolvimento:

O programa deve ser distribuído e desenvolvido na linguagem de programação Java, sendo utilizado o paradigma de programação orientado a objetos.

O programa deve possuir uma interface 2D, sendo o programa controlado via teclado e mouse.

O jogo deverá ser compatível para ser jogado em rede por dois jogadores pelo servidor NetGamesNRT.

3.Requisitos de Software

3.1. Requisitos Funcionais:

Requisito 1: Iniciar Partida: o programa deve oferecer a função de iniciar uma partida, tendo a escolha da quantidade de jogadores, cada um com seu nome.

Requisito 2: Conectar com servidor: o programa deverá oferecer pela sua interface, uma forma de se conectar com o servidor NetGames NRT para que as partidas em rede sejam efetuadas.

Requisito 3: Procedimento de Turno:

O programa deve permitir aos jogadores que efetuem certas ações durante seu turno:

3.3.1: Mover sua minhoca dentro do espaço de jogo.

3.3.2: Selecionar e utilizar uma das armas ou ferramentas disponíveis no arsenal das minhocas.

3.3.3:Coletar uma caixa de suprimentos.

3.3.4: o jogo deve gerenciar as minhocas e cada arma que cada minhoca possui e quanto dano receberam.

Requisito 4: Customização: O jogo deve possuir um menu de opções para que regras únicas sejam selecionadas.

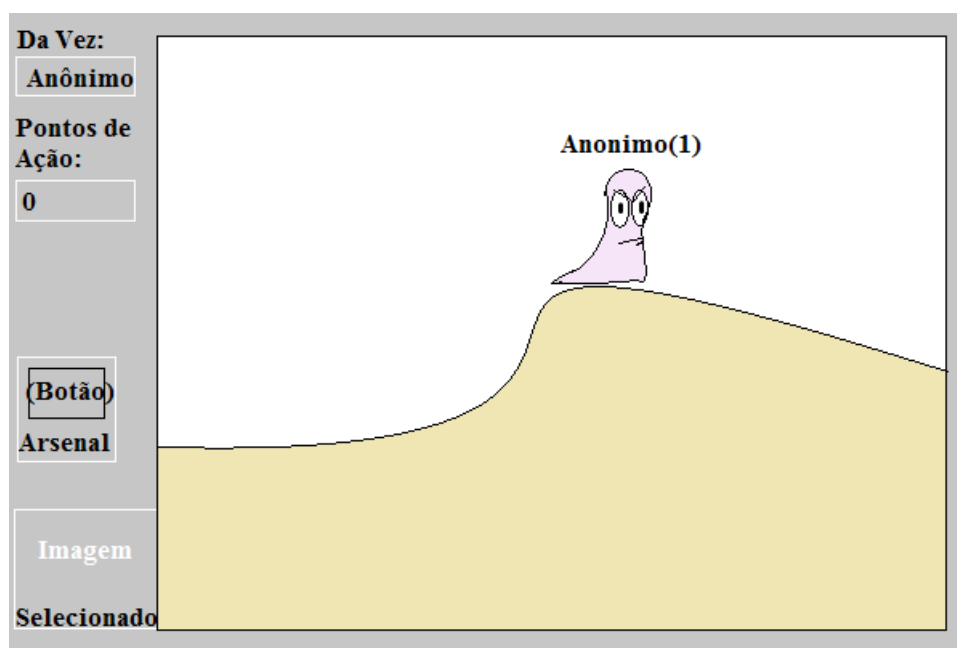
Requisito 5: Atualização de Turno: a cada jogada, o jogo deve atualizar o estado das minhocas, informar ao jogador da vez que é sua vez, e mostrar aos jogadores os efeitos do que foi efetuado na jogada anterior.

3.2 Requisitos não funcionais:

Requisito 1: O jogo deve ser projetado, e planejado, produzindo uma especificação de projeto baseada em UML2, utilizando a ferramenta visual paradigm.

Requisito 2: O jogo deverá possuir uma interface gráfica, a qual os dois usuários possuem acesso, desenvolvido com o pacote java swing e java awt.

Anexo 1: Esboço da Interface:



Botão arsenal: Abre a aba de armas e ferramentas para serem selecionadas.

Pontos de ação: Mostra o quanto se ainda pode fazer em sua jogada, utilizar uma arma ou ferramenta consome todos os pontos de ação restantes.

Da Vez: mostra o jogador da vez.

Selecionado: mostra a arma ou ferramenta selecionada.

Na tela: Marrom: chão do mapa

Minhoca: mostra a minhoca selecionada.