

Especificações dos Requisitos

Projeto CIVILIZATION

Evandro Chagas Ribeiro da Rosa
Lucas Cavalcante de Sousa

November 30, 2016

Versão 1.1

Versão	Autores	Data	Ação
1.0	Evandro Chagas e Lucas Sousa	07/09/2016	Estabelecimento de Requisitos
1.1	Evandro Chagas e Lucas Sousa	17/09/2016	Regras e correção dos requisitos
1.2	Evandro Chagas e Lucas Sousa	22/10/2016	Correção das Premissas
1.3	Evandro Chagas e Lucas Sousa	30/11/2016	correções finais

Índice

1	Introdução	1
1.1	Objetivo	1
1.2	Nomenclatura e siglas usada	2
1.3	Regras do Jogo	2
1.3.1	Construções e Unidades	2
1.3.2	Movimentos	3
2	Visão Geral	4
2.1	Arquitetura do programa	4
2.2	Premissas de desenvolvimento	4
3	Requisitos	5
3.1	Requisitos Funcionais	5
3.2	Requisitos não funcionais	5
4	Interface com o jogador	5

1 Introdução

1.1 Objetivo

Desenvolvimento de um software distribuído que simule uma guerra entre duas nações em um jogo de tabuleiro baseado em turno.

1.2 Nomenclatura e siglas usada

1. Bacoïn: Unidade monetária de jogo.
2. Tropa: unidade de batalha.
3. Campo: posição no tabuleiro.
4. Capturável: passível de ser dominada pelo inimigo.
5. Intransportável: Uma unidade inimiga não pode passar por esse campo.
6. HP: Pontos de vida.
7. ATK: Pontos de ataque.
8. R: Distancia, em campos, de ataque da unidade.
9. MV: Distancia, em campos, de deslocamento da unidade.
10. P: Custo de produção.
11. M: Custo de manutenção.

1.3 Regras do Jogo

Para o jogo é necessário 2 jogadores, cada um deles com representados por uma nação, Coquinho e Stark. O objetivo do jogo é dominar a Cidade Estado do inimigo.

1.3.1 Construções e Unidades

1. Construções
 - (a) Cidade Estado: Construção primordial da nação.
 - i. HP: 100
 - ii. M: +10
 - (b) Fazenda: Produtora de Baicons da nação.
 - i. HP: 20
 - ii. M: +10
 - iii. P: 10
 - iv. Capturável
 - (c) Vila: Provem unidades para nação
 - i. HP: 20
 - ii. M: -10
 - iii. P: 15
 - (d) Muralha: Protege a nação

- i. HP: 80
- ii. M: -10
- iii. P: 20
- iv. Intransportável

2. Unidades

Todas as unidades possuem preço 10 e manutenção 5.

(a) Arqueiro

- i. HP: 20
- ii. ATK: 10
- iii. R: 2
- iv. MV: 1
- v. Tropa

(b) Cavalaria

- i. HP: 25
- ii. ATK: 10
- iii. R: 2
- iv. MV: 1
- v. Tropa

(c) Trabalhador

- i. HP: 5
- ii. ATK: 5
- iii. R: 1
- iv. M: 1

1.3.2 Movimentos

1. Caso o jogador tenha Bacoins e um trabalhador com movimento disponível ele poderá iniciar uma construção, sendo elas:

- (a) Muralha
- (b) Vila
- (c) Fazenda

2. Caso tenha Bacoins suficiente e uma Cidade Estado ou uma Vila disponível poderá requisitar uma unidade, sendo elas:

- (a) Trabalhador
- (b) Cavalaria
- (c) Arqueiros

3. O jogador pode mover uma unidade se ela tiver movimentos disponível. A unidade poderá mover para qualquer posição dentro do seu Range seguindo a seguinte regra:
 - (a) Não poderá haver outra unidade naquela posição.
 - (b) Não poderá se mover sobre Muralha ou Cidade Estado do inimigo.
4. Caso o jogador tenha uma unidade próxima, relativo a unidade, de uma unidade/construção inimiga ele poderá atacar essa unidade/construção inimiga tendo as seguintes consequências:
 - (a) Se zerar a vida (HP) de uma construção:
 - i. Cidade Estado: jogador atacante ganha o jogo.
 - ii. Muralha ou Vila: Ela é destruída.
 - iii. Fazenda: Ela é dominada pelo jogador atacante.
 - (b) Se for uma unidade.
 - i. A unidade atacante sofre metade do ATK da unidade atacada, isso pode significar a morte da unidade atacante.
 - ii. Se o HP da unidade atacada chegar em 0 ela é morta.
5. Caso tenha Bacoins negativo poderá vender algumas de suas unidades a fim de sair do negativo, caso não consiga o jogador perde o jogo.
6. Se render: Declara vitória inimiga.
7. Passar turno: passa o turno para o próximo jogador.

2 Visão Geral

2.1 Arquitetura do programa

programa orientado a objetos, distribuído.

2.2 Premissas de desenvolvimento

1. O programa deve apresentar uma interface gráfica bidimensional, dividida em uma matriz 13x13 onde cada posição é um campo que pode suportar uma construção e uma unidade.
2. O programa fará uso da linguagem de programação Java, com o paradigma de Orientação a Objetos, do software para cliente-servidor NetGames.
3. Fazer a modelagem do projeto utilizando UML, versão 2.
4. Possuir uma interface gráfica.

3 Requisitos

3.1 Requisitos Funcionais

1. Conectar: A aplica deve permitir que os usuários conectem-se ao servidor do NetGames.
2. Desconectar: Devera permitir que o usuário desconecte-se do servidor.
3. Iniciar partida: Apos dois jogadores estarem conectados sera permitido que se inicie uma nova partida.
4. Fazer Movimento: Durante seu turno o jogador poderá fazer movimento conforme as regras.

3.2 Requisitos não funcionais

1. Utilizar a linguagem de programação Java e a biblioteca NetGames.
2. Ter dois jogadores.

4 Interface com o jogador

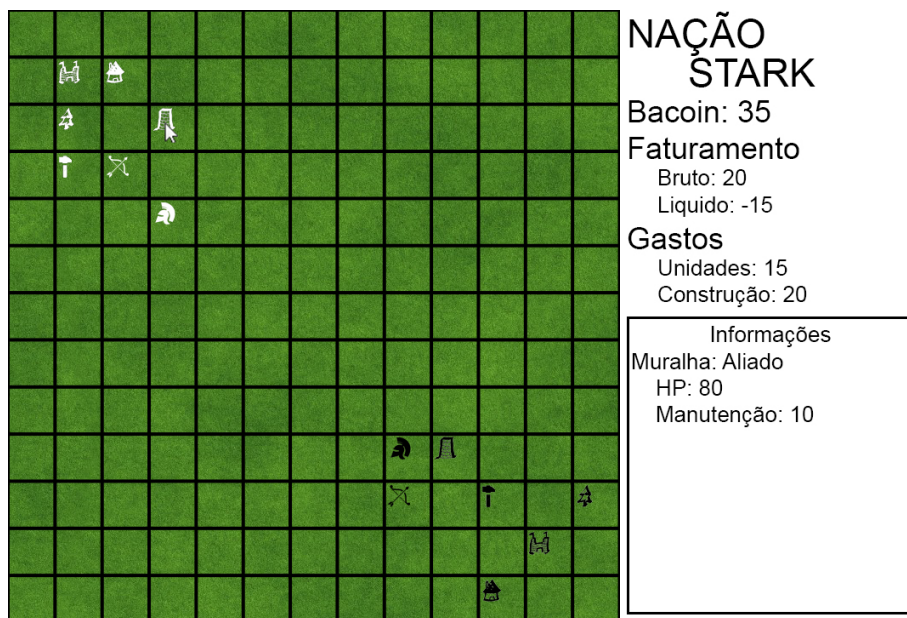


Figure 1: Versão Alpha da interface principal do jogo.