



**UNIVERSIDADE FEDERAL DE SANTA CATARINA**

MARCELO EMILIO VENDRAMIN

## **Especificação de Requisitos**

FLORIANOPOLIS

2018

## 1. Introdução

O jogo será disputado em um tabuleiro 4x4 que possui algumas posições bloqueadas no seu início. Estas posições bloqueadas são definidas de forma aleatória através do tabuleiro e variam de seis a oito, e uma posição bloqueada corresponde a uma posição em que nenhum jogador pode colocar suas cartas.

Os jogadores recebem cinco cartas no início da rodada, cada uma delas possuem dois valores de ataque, dois valores de defesa, um elemento e um número entre 3 e 6 de direções de ataque. Os dois valores de ataque e os dois valores de defesa correspondem aos valores mínimo e máximo que esta carta pode atingir, e na hora de realizar um ataque, estes valores serão utilizados para decidir o menor e maior resultado possível.

No momento de realizar um ataque a carta que esta atacando gera um valor de ataque e a carta que esta na defesa gera um valor de defesa.

Para realizar um ataque com uma carta é gerado, na hora do ataque, um valor de ataque para a carta atacante e um valor de defesa para a carta atacada. O valor de ataque consiste na escolha aleatória de um número inteiro que está entre os valores de máximo e mínimo ataque disponíveis, e este valor é dobrado se o elemento da carta atacante possui vantagem contra o elemento da carta atacada. O valor de defesa consiste na escolha aleatória de um número inteiro que está entre os valores de máximo e mínimo defesa disponíveis na carta defensora. O ataque ocorreu com sucesso se o valor de ataque é maior que o valor de defesa.

Os elementos são Terra, Agua, Fogo, Ar e Magia. Alguns elementos possuem a vantagem de dobrar o resultado do ataque ao atacar outros elementos, conforme segue:

Fogo tem vantagem sobre Ar;

Ar tem vantagem sobre Terra;

Terra tem vantagem sobre Água;

Água tem vantagem sobre Fogo.

Uma direção de ataque indica para que posições adjacentes do tabuleiro uma carta consegue realizar um ataque. As direções de ataque de uma carta são definidas na distribuição de cartas aos jogadores, e permanecem imutáveis até o fim do jogo. Existem oito direções de ataque, sendo elas: Cima, Baixo, Direita, Esquerda, Cima-Direita, Cima-Esquerda, Baixo-Direita e Baixo-Esquerda.

Ao colocar uma carta no tabuleiro é atacado todas as cartas inimigas que estão nas direções possíveis de ataque. Se o atacante ganhar, a carta derrotada passa a ser controlada pelo atacante, e a carta que foi controlada verifica se há outras cartas inimigas que podem ser atacadas por ela (estão em uma de suas direções de ataque).

A partida acaba quando todas as posições do tabuleiro estão em uso (contendo uma carta ou bloqueada) ou quando algum dos jogadores da partida desiste. Se o fim da partida acontece por desistência, o vencedor é o jogador contrario ao que desistiu, mas se a partida acaba pela utilização de todas as posições do tabuleiro, o vencedor será o jogador que controlar mais posições no tabuleiro. Se a quantidade de posições controladas pelos jogadores for igual será declarado empate.

## **1.1. Objetivo do desenvolvimento**

O objetivo do projeto é desenvolver o jogo Ethereum Masters, que será jogado por dois jogadores de forma online através da forma de conexão cliente servidor adotada pela NetGamesNRT. Também será desenvolvido toda a documentação necessária para a compreensão do projeto.

## **2. Visão Geral do sistema**

### **2.1. Arquitetura**

Será utilizada uma arquitetura de cliente-servidor distribuído, que possibilitará o jogo de forma online pelos jogadores através da NetGamesNRT.

### **2.2. Premissas de desenvolvimento**

#### **2.2.1. Linguagem de Programação**

O jogo será desenvolvido na linguagem de programação Java.

### **2.2.2. Jogabilidade online**

O desenvolvimento da jogabilidade online será através do servidor e framework da NetGamesNRT

### **2.2.3. Desenvolvimento da Documentação**

A especificação de requisitos será desenvolvida utilizando a língua portuguesa, enquanto os diagramas serão desenvolvidos utilizando-se a linguagem UML e o programa de edição Visual Paradigm.

## **3. Requisitos da aplicação**

### **3.1. Requisitos funcionais**

#### **3.1.1. Atacar Posição**

Descrição: Recebe do programa a posição que realizará o ataque e a posição que realizará a defesa; calcula os valores para ataque e defesa, e se o valor obtido como ataque for maior que o de defesa, captura a posição da carta defensora para o jogador adversário e realiza todos os ataques possíveis para aquela posição.

Condição: As duas posições estão ocupadas com cartas de jogadores diferentes.

Muda: Se o ataque ocorre com sucesso muda o jogador que controla a carta presente na posição de defesa e o placar.

Envia: Se o ataque ocorreu com sucesso.

Resulta: Se o ataque ocorreu com sucesso ocorre mudança sobre que jogador controla a carta presente na posição atacada, e acontece todos os ataques disponíveis para aquela posição. Senão partida no estado anterior à operação.

#### **3.1.2. Desistir**

Descrição: Cede a vitória ao oponente do jogador que desistir.

Condição: Ter uma partida em andamento.

Muda: Vencedor da partida.

Envia: Qual jogador venceu a partida.

Resulta: Mensagem mostrando o vencedor da partida e com a possibilidade de sair ou reiniciar o jogo.

### **3.1.3. Iniciar Partida**

Descrição: Cria um tabuleiro quadrado que contem 16 posições, e define um numero entre um e seis de posições bloqueadas; distribui as cartas aos jogadores, definindo para cada carta um numero entre 3 e 6 de direções de ataque, escolhidas de forma aleatória. O ataque e defesa mínimo é definido como 1 ou 2 e o ataque e defesa máximo é definido como 3 ou 4, ambos de forma aleatória; inicia o placar da partida.

Condição: Escolha por parte do jogador de inicio ou reinicio de uma partida.

Muda: Valores atribuídos as cartas e distribuição das posições bloqueadas do tabuleiro.

Envia: Estado da partida, posições bloqueadas.

Resulta: Jogadores com os valores de suas cartas definidos, tabuleiro no estado inicial (nenhuma carta ocupa alguma posição), placar zerado.

### **3.1.4. Placar**

Descrição: Mostra o placar da partida na tela, informando quantas posições cada jogador possui em seu controle.

Condição: Ter uma partida em andamento.

Envia: Quantas posições cada jogador possui em seu controle.

### **3.1.5. Utilizar carta**

Descrição: Obtém do jogador a carta a ser utilizada e a posição escolhida; procede com a ocupação da posição pela carta e realização de todos os ataques possíveis.

Condição: Carta esta com o jogador e posição livre.

Muda: Posições livres do tabuleiro e placar.

Resulta: Uma captura de posição por parte de um jogador, novos ataques a serem realizados e muda a vez de que jogador é a vez.

## **3.2. Requisitos não funcionais**

### **3.2.1. Desenvolvimento de documentação**

A documentação do projeto será escrita com a utilização da linguagem UML através do programa visual paradigm.

### **3.2.2. Interface gráfica**

O programa deverá utilizar uma interface gráfica funcional, e a diferença da interface apresentada aos jogadores estará somente nas cartas que eles ainda não utilizaram.

### **3.2.3. Conectividade**

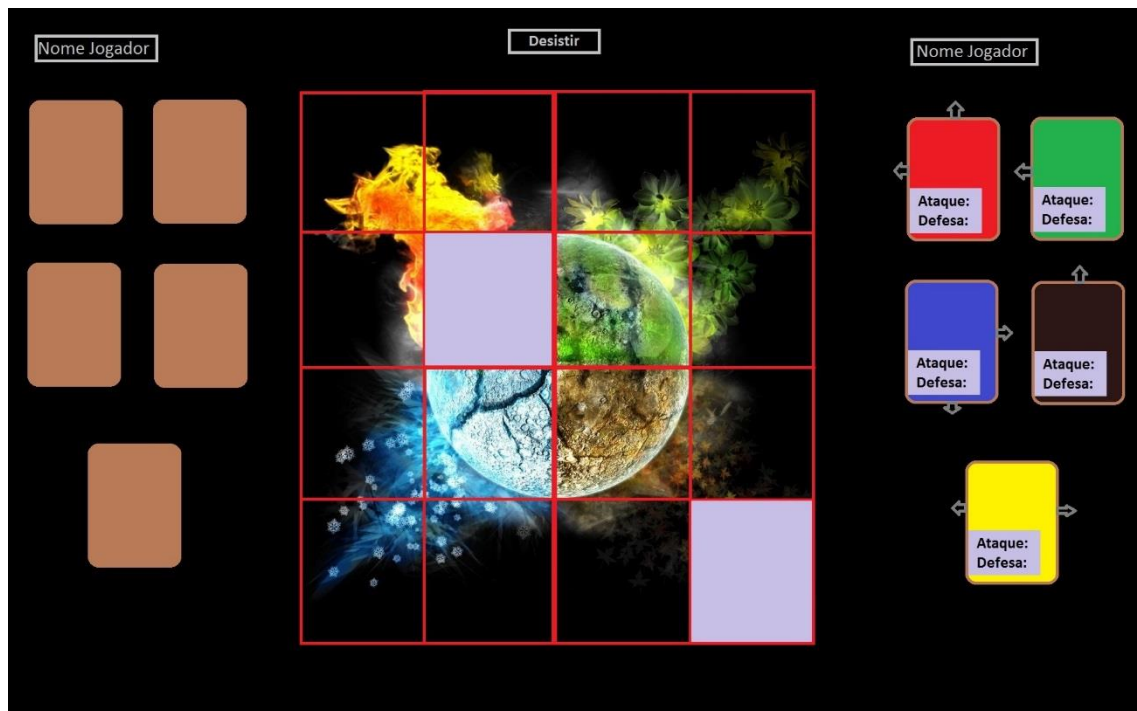
O programa deverá utilizar a conexão cliente/servidor estabelecida pela NetGamesNRT.

### **3.2.4. Execução**

O programa deverá executar sem erros em qualquer maquina que possua a máquina virtual Java instalada.

## **4. Esboço**

### **4.1. Tela do Jogo**



## 4.2. Tela de Inicio

