

## **Especificação de requisitos**

### **Projeto do jogo Shogi**

#### **Especificações de requisitos de software**

Versão 1.0

16/09/2014

<b>Versão</b>	<b>Autores</b>	<b>Data</b>	<b>Ação</b>
1.0	Yuri Kayser da Rosa, Thiago Deon	16/09/2014	Estabelecimento dos requisitos
1.1	Yuri Kayser da Rosa, Thiago Deon	01/12	Correção de erros, alteração do requisito Sair da Partida, alteração do Esboço de Interface

#### **Conteúdo:**

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

#### **1. Introdução**

##### **Objetivo:**

Desenvolvimento de um programa que possibilite dois usuários jogarem em rede o jogo “Shogi”.

## **Referência:**

As regras do jogo estão especificadas no item 2.1.

## **2. Visão geral**

### **2.1 Regras do Jogo Shogi**

#### **a) Estrutura e Posicionamento Inicial:**

O jogo começa com um tabuleiro 9x9. Cada jogador possui vinte peças, sendo elas:

- 1 Rei
- 2 Generais de Ouro
- 2 Generais de Prata
- 2 Cavalos
- 2 Lanceiros
- 1 Bispo
- 1 Torre
- 9 Peões

Os peões se colocam na terceira linha ; o bispo na segunda linha e segunda coluna; a torre na segunda linha e oitava coluna; os lanceiros na primeira linha, uma na primeira coluna e outro na nona coluna; os cavaleiros na primeira linha, um na segunda coluna e outro na oitava coluna; os generais de prata na primeira linha, uma da terceira coluna e outro na sétima; os generais de ouro na primeira linha, um na quarta coluna outro na sexta; e o rei na primeira linha e quinta coluna.

#### **b) Movimentação das peças:**

Cada peça possui regras de movimentação específicas. são elas:

- Rei: Uma casa para qualquer direção;
- Torre: Qualquer número de casas na vertical ou horizontal desde que não existam peças interrompendo a trajetória;
- Bispo: Qualquer número de casas na diagonal desde que não existam peças interrompendo a trajetória;
- General de Ouro: Uma casa em qualquer direção, menos nas diagonais para trás

- General de Prata: Uma casa para frente na diagonal ou na vertical, ou uma casa para trás na diagonal;
- Cavalo: Num mesmo movimento, uma casa na vertical para frente e outra na diagonal para frente; não pode recuar;
- Lanceiro: Qualquer número de casas na vertical para frente desde que não existam peças interrompendo a trajetória; não pode recuar;
- Peão: Uma casa para frente;

#### **c) Eliminando peças:**

Se após a realização de um movimento houver uma peça do seu adversário na posição que sua peça irá ocupar, a peça do adversário é eliminada.

#### **d) Final de jogo:**

O jogo termina quando o Rei de um dos jogadores for eliminado.

### **2.2 Arquitetura do programa:**

- Programa orientado a objetos, na linguagem java.
- Aplicação distribuída, cliente-servidor;

### **2.3 Premissas de desenvolvimento:**

O software deve seguir as seguintes premissas:

- ser apresentado em uma interface gráfica de duas dimensões;
- ser implementado em Java, devendo executar em qualquer plataforma que disponha da máquina virtual Java;
- ferramenta NetGamesNRT para fazer interação entre os usuários.

## **3. Requisitos de software**

### **3.1 Requisitos Funcionais:**

**Conectar:** O programa deve apresentar uma opção conectar, para estabelecer conexão com o servidor. O jogador deverá definir o nome que o identificará durante a partida nesta etapa.

**Desconectar:** O jogo deve apresentar a opção “Desconectar”, permitindo desconectar-se do servidor. Caso haja uma partida em andamento, a mesma será encerrada.

**Iniciar Partida:**

O programa deve apresentar uma opção iniciar para iniciar uma partida. É necessário estar conectado ao servidor para iniciar uma nova partida.

**Mover Peça:**

O movimento se dá com o jogador da vez selecionando uma de suas peças e clicando em uma das casas do tabuleiro. Se a casa estiver de acordo com as restrições de movimento da peça, descritas no item B da seção 2.1, ocorre o movimento.

**Estado da partida:**

A cada tentativa de procedimento de movimento dos jogadores(bem sucedida ou não) o novo estado da partida deve ser exibido na interface, considerando as seguintes restrições:

- O tabuleiro deve ser atualizado com a movimentação correta da peça de acordo com a jogada correspondente, no caso de jogada bem sucedida;
- No caso de jogada irregular, deve ser comunicado o tipo de irregularidade. Exemplos: “Posição já ocupada”; “Esta peça é do oponente”, “Movimento inválido”.

**Sair da Partida:**

O programa deve apresentar uma opção de “sair” que desconecta o jogador caso esteja conectado, e fecha o programa.

**3.2 Requisitos Não Funcionais:****Especificação de projeto:**

Software escrito na linguagem de programação Java, e especificação feita em UML segunda versão.

**Interface gráfica para usuário:**

O programa deverá ter interface gráfica única, partilhada pelos usuários.

**Símbolos dos jogadores:**

As peças serão diferenciadas pelas orientação das peças. As peças do jogador local estarão apontando para cima e as do jogador remoto apontando para baixo;.

**Tecnologia de interface gráfica para usuário:**

A interface gráfica deve ser baseada em *Java-Swing*.

#### 4. Esboço da interface gráfica

