

Especificação de Requisitos

Jogo de Tabuleiro Hnefatafl

Projeto Hnefatafl

Especificação de Requisito de Software

Versão 1.3
04/07/2017

Versão	Autor(es)	Data	Ação
1.0	Eduardo Schmitt Kreuch	09/04/2017	Estabelecimento dos Requisitos
1,1	Eduardo Schmitt Kreuch	18/04/2017	Correção em requisitos funcionais
1.2	Eduardo Schmitt Kreuch	26/05/2017	Retirada função de "Salas"
1.3	Eduardo Schmitt Kreuch	04/07/2017	Correção em requisitos funcionais, da regra do jog e as imagens de referência

Conteúdo:

1. Introdução;
2. Visão geral;
3. Requisitos do Software;
4. Esboço Interface do Software;
5. Figuras de Apoio às Especificações;

1 - Introdução

Objetivo: desenvolvimento de um programa para disputas de partidas de Tafil em modalidade Hnefatafl com disputa usuário contra usuário.

Regras do jogo:

- O jogo é jogado em um tabuleiro 11x11. A configuração inicial é tal como mostrada no diagrama da figura 1.

- O rei começa no quadrado central ou castelo, o *konakis*, que nenhuma outra peça pode ocupar.
- Os doze defensores, chamado *suecos* (em nossa versão, Reino de Wessex), começam nos doze quadrados adjacentes ao *konakis*.
- Os vinte e quatro atacantes, chamado *moscovitas* (em nossa versão, Vikings), começam em grupos de seis no centro de cada borda do tabuleiro.
- Os quadrados de canto [1-1], [1-11], [11-1] e [11]-[11] são os refúgios para o rei e somente ele pode ocupá-los.
- Todos os quadrados restantes (*zona neutra*) podem ser ocupados por qualquer peça durante o jogo.
- Qualquer peça pode mover-se qualquer número de espaços vagos em qualquer linha reta [$\leftarrow \uparrow \rightarrow \downarrow$], mas não na diagonal.
- Nenhuma peça pode transpor outra peça em seu caminho.
- Qualquer peça, pode ser capturada e removida do tabuleiro quando cercada em dois lados opostos pelos inimigos ou por um lado o *Konaki* ou o refúgio e do outro o inimigo. (*isto é conhecido como a captura de prisão*).
- A vitória dos *suecos* acontece quando o **rei** consegue chegar sem ser capturado em um dos refúgios nos cantos do tabuleiro.
- A vitória dos *moscovitas* acontece quando o rei é capturado.
- Se qualquer um dos exércitos forem aniquilados, o oponente vence.

Referências:

<<https://pt.wikipedia.org/wiki/Tafl>> (Reconstrução - Usado como base, mas alterado para outra modalidade do jogo Tafl)

2 - Visão Geral

Arquitetura do programa: Programa orientado a objetos, partida distribuída em modelo cliente-servidor.

Premissas de Desenvolvimento:

- O programa deve ser implementado em Java na versão 8;
- O programa deve utilizar a ferramenta NetGamesNRT desenvolvida por Leonardo Brasil para suporte a partidas distribuídas em arquitetura cliente-servidor.

3 - Requisitos de Software

3.1 Requisitos Funcionais:

Requisito Funcional 1 - Conecta: O programa deve apresentar a opção de “Conectar” para que o jogador possa se conectar ao servidor onde poderá ter outros jogadores dispostos a jogar Hnefatafl.

Requisito Funcional 2 - Desconectar: O programa deve apresentar a opção de “Desconectar” para que o jogador possa se desconectar do servidor que está conectado.

Requisito Funcional 3 - Iniciar Partida: O programa deve apresentar a opção de menu “Iniciar Partida” para o início de uma nova partida, considerando as seguintes restrições:

- Jogador já deve estar conectado ao servidor;
- Jogador não escolhe o time que irá jogar.

Requisito Funcional 4 - Procedimento de Lance: O programa deve oferecer suporte ao procedimento de lance do usuário em sua respectiva vez, possibilitando o mesmo a selecionar a peça e onde ele deseja colocá-la com *clicks* do botão esquerdo do mouse. A operação deve ter as seguintes restrições:

- O programa só deve aceitar lances válidos de acordo com as regras de jogo (ver introdução);
- A cada lance válido o jogo deve verificar se houve captura de alguma peça, caso sim, o programa deve remove-la do tabuleiro;
- A cada lance válido o jogo deve verificar se houve ganhador da partida;

Requisito Funcional 5 - Comunicação do Estado da Partida: A cada procedimento de lance do jogador, está jogada deve ser repassada ao outro jogador conectado, assim atualizando o tabuleiro de ambos e passando a vez entre os jogadores.

Requisito Funcional 6 - Ajuda: O programa deve ter um botão “Ajuda”, onde este abre um descritivo completo das regras do jogo para auxílio do jogador.

Requisito Funcional 7 - Desistir da Partida: O jogador pode ter a opção de desistir de uma partida mas se mantém conectado ao servidor, o programa deve verificar as seguintes restrições:

- O jogador já deve estar conectado a uma partida para ter a opção de desistir;
- O jogador adversário deve ser avisado de sua vitória e disponibilizado a opção de buscar novas partidas.

Requisito Funcional 8 - Sair: O programa deve ter a opção “Sair” para encerrar o jogo, o programa deve verificar as seguintes condições:

- Caso jogador que saiu estiver conectado ao servidor, encerrar a conexão;
- Caso o jogador estiver durante uma partida, deve avisar o outro jogador que poderá buscar outro oponente para jogar e encerrar a conexão do jogador que saiu.

3.2 - Requisitos Não Funcionais:

Requisito não Funcional 1 - Especificação de Projeto: Além do código Java, deve ser produzida especificações de projeto baseada em UML, segunda versão.

Requisito não Funcional 2 - Interface Gráfica para Usuário: O programa deverá ter interface gráfica única para ambos jogadores implementada com a biblioteca Java Swing.

Requisito não Funcional 3 - Símbolos do Jogo: As peças dos jogadores serão representadas de maneira diferente, as peças dos *suecos* terão o símbolo do reino de Wessex (figura 2) e as peças *moscovitas* serão representados como vikings (figura 3), o rei será representado como uma peça comum mas com uma coroa na representação (figura 4), o konaki, assim como o refúgio serão representados com uma fortaleza na posição do tabuleiro (figura 5), as demais posições serão de cor única.











































4 - Esboço interface do software

Conectar a Uma Sala

Iniciar Partida

Reiniciar Partida

Vez (nomeJogador + simbolo)

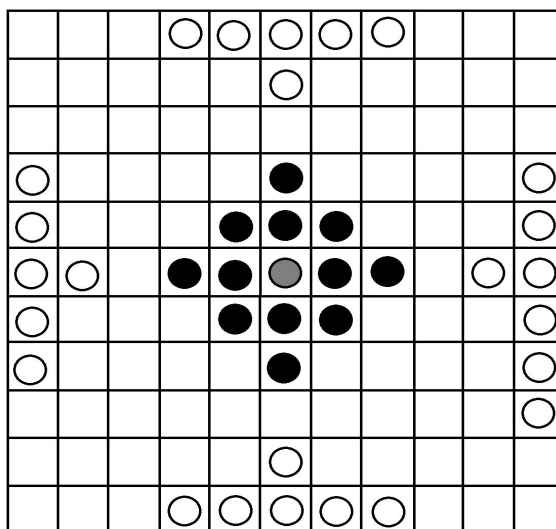
Nome Jogador + simbolo												Nome Jogador + simbolo
												
												
												
												
												
												
												
												
Peças mortas Moscovitas												Peças mortas Suecos

Sair

Desconectar da Partida

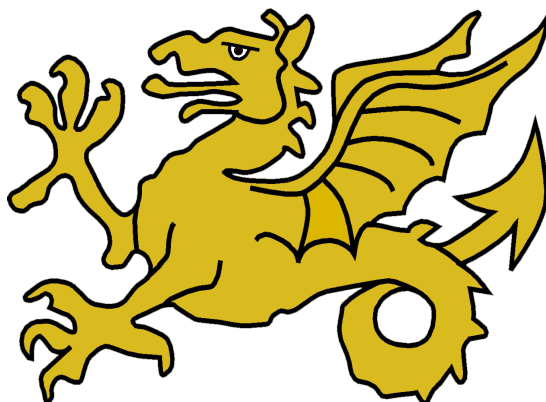
5 - Figuras de Apoio às Especificações

Figura 1 -



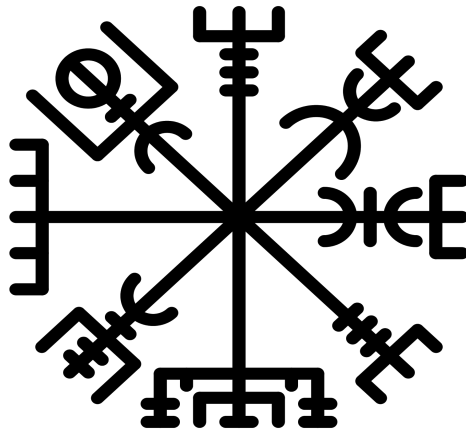
Tabuleiro com peças dispostas, pretas suecos, cinza rei e brancas moscovitas

Figura 2 -



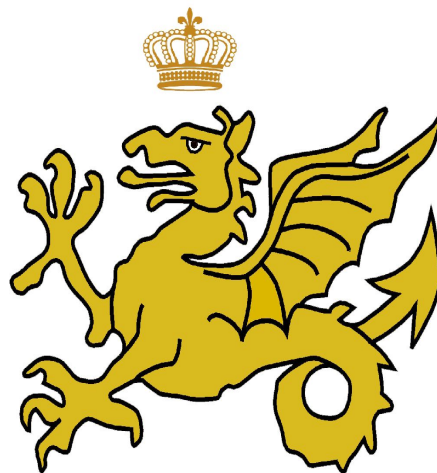
Símbolo de Wessex, representação dos suecos

Figura 3 -



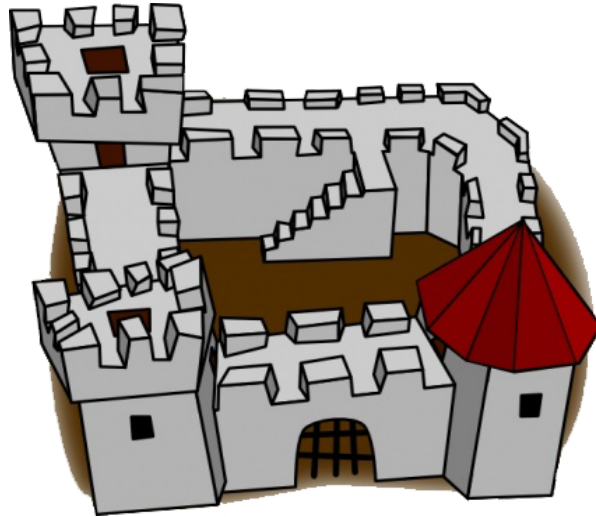
Símbolo Viking, representação dos moscovitas

Figura 4 -



Símbolo de Wessex com Coroa, representação do Rei

Figura 5 -



Símbolo de Fortaleza, representação do konaki e dos refúgios