

Projeto DogTabuleiro

Especificação de Requisitos de Software

Versão 2.0

11/07/2019

Carolina Alves dos Santos

Leonardo Martins Gonçalves

Pedro Paulo Chiarelli Steil

Release Notes

Versão	Data	Atualização
v0.1	15/04/2019	Definição dos requisitos do software. Primeira versão da interface gráfica com interação.
v0.2	27/05/2019	Criação dos diagramas de atividades, estado e visão geral da interação. Adição do NetGames na modelagem.
v0.5	01/07/2019	Integração com NetGames no projeto. Melhorias na interface gráfica. Correções e adições de diagramas.
v1.0	11/07/2019	Finalizada parte gráfica e lógica do jogo. Ajustes nos diagramas

1 Introdução

1.1 Objetivos

O projeto aqui especificado tem por objetivo o desenvolvimento de um jogo eletrônico baseado no jogo DogTabuleiro, adaptado para propiciar a disputa de partidas apenas entre dois jogadores e com o uso de baralho em vez de dados.

1.2 Referências

As regras do jogo original (chamado de *Dog*) podem ser encontradas em <https://boardgamegeek.com/thread/343159/sorry-meets-ludo-two-packs-cards>.

1.3 Regras do jogo (adaptadas)

Em relação às regras originais, adaptamos o jogo para que seja possível apenas dois jogadores jogarem.

2 Visão Geral

2.1 Arquitetura da aplicação

Programa orientado a objetos e distribuído com base no modelo cliente-servidor.

2.2 Premissas de desenvolvimento

- Programa deve possuir uma interface gráfica;
- Linguagem de programação adotada para desenvolvimento deve ser Java;
- A modelagem da especificação deve ser baseada em UML 2;

- O programa deve obrigatoriamente executar distribuído, com o suporte para aplicações distribuídas *NetGamesNRT*.

3 Requisitos da aplicação

3.1 Requisitos funcionais

- 1) Conectar-se ao servidor: o programa deve ter uma opção de menu para que o jogador possa se conectar ao servidor de uma rede de jogadores e disputar partidas;
- 2) Iniciar nova partida: o programa deve ter uma opção de menu para que o jogador possa iniciar uma nova partida do jogo, se já estiver conectado a um servidor;
- 3) Desconectar-se do servidor: o programa deve ter uma opção de menu para que o jogador se desconecte do servidor, se estiver conectado a um;
- 4) Receber solicitação de início: recebe uma solicitação do servidor, que sorteia o jogador que começará o jogo. Este instanciará a mesa, consequentemente o baralho, e distribuirá um número de cartas para cada jogador. O número de cartas que é entregue aos jogadores depende do turno em que o jogo se encontra, variando sempre de acordo com as regras.

Se houver uma partida em andamento, o programa deve possibilitar ao jogador:

- 5) Escolher Carta: o jogador deve escolher uma carta que possui em sua mão, que determinará quantas posições uma de suas peças poderá se movimentar para frente no tabuleiro;
- 6) Mover Peça: o jogador deve uma de suas peças para movimentar no seu turno;

Se uma peça colocada em cima de outra, o programa deve:

- 7) Trocar a peça: colocar a peça recém-chegada na casa que se encontrava a outra peça, e colocar a peça removida na posição inicial das peças de que pertence ao jogador cuja peça foi removida;

Se acabarem as cartas da mão dos jogadores, o programa deve:

- 8) Distribuir cartas novamente: deve ser distribuída mais cartas para os dois jogadores, sendo que a quantidade dependerá do turno em que o jogo se encontra;

Se acabarem as cartas nos baralhos do tabuleiro quando deveriam ser distribuídas novas cartas aos jogadores, o programa deve:

- 9) Embaralhar novamente o baralho: deve ser embaralhada todas as cartas para ter disponibilidade do uso das cartas novamente.

3.2 Requisitos não-funcionais

- 1) O desenvolvimento deve ser efetuado orientado a objetos, na linguagem Java;

- 2) A modelagem do jogo deve ser efetuada com a linguagem UML 2;
- 3) A ferramenta utilizada para modelagem deve ser o Visual Paradigm;
- 4) A aplicação deve ter uma interface gráfica, com estados compartilhados pelos usuários que estiverem jogando, desenvolvida com a biblioteca gráfica *Swing*;
- 5) A conexão entre os jogadores deve ser realizada com a plataforma *NetGamesNRT*;
- 6) O jogo deve ser disponibilizado aos usuários por meio de um arquivo executável no formato JAR.