

Universidade Federal de Santa Catarina
Centro Tecnológico
Departamento de Informática e Estatística
Curso de Ciência da Computação
INE5417 - Engenharia de Software I

Projeto Card Chess: Conquer the Board

Maria Eduarda de Melo Hang
Thiago Sant' Helena da Silva

Florianópolis
16 de Abril de 2019

Conteúdo

1	Tabela de Versões	2
2	Introdução	3
2.1	Objetivos	3
2.2	Referências	3
2.3	Definições	3
2.3.1	Tipo de peças	3
2.3.2	Regras	3
2.3.3	Fases de um turno	3
2.3.4	Cartas	4
3	Visão geral do Sistema	5
3.1	Arquitetura da aplicação	5
3.2	Premissas de desenvolvimento	5
4	Requisitos funcionais da aplicação	6
4.1	Conexão com o servidor	6
4.2	InterfaceJogador	6
4.3	Utilização de uma carta pelo jogador ativo	7
4.4	Interface Gráfica	7
5	Requisitos não-funcionais da aplicação	8

Capítulo 1

Tabela de Versões

Versão	Autor(es)	Data	Ação
1.0	Maria Eduarda e Thiago	06/04/19	Levantamento de Requisitos
1.1	Maria Eduarda e Thiago	28/05/19	Refinamento de casos de uso
1.2	Maria Eduarda e Thiago	01/07/19	Produção de Algoritmos e Implementação

Capítulo 2

Introdução

2.1 Objetivos

Desenvolver um programa, "Card Chess: Conquer the Board", baseado nas regras de xadrez com a presença de cartas que proporcionam novas estratégias.

2.2 Referências

O tabuleiro do programa segue as definições encontradas no xadrez que podem ser vistas neste link: https://pt.wikipedia.org/wiki/Leis_do_xadrez. O estilo de movimentação das peças rei, torre e bispo são mantidos, mas o alcance foi alterado. A definição de vitória é igual ao jogo original.

2.3 Definições

2.3.1 Tipo de peças

Nome	Movimento	Alcance do movimento	Quantidade por jogador
Rei	Cruz e Diagonal	1	1
Torre	Cruz	4	2
Bispo	Diagonal	2	2

2.3.2 Regras

- Vitória: Retirada do rei adversário do tabuleiro;
- Empate: Restam dois reis no tabuleiro;
- Movimentação inválida: Sobrepor as próprias peças ou pular peças adversárias;
- Estado de compra de cartas: Quando o jogador jogar suas 5 cartas iniciais, ele entra no estado "Compra de cartas" e compra uma carta no início de cada turno, antes da primeira fase de utilização de cartas;
- Derrota por deck vazio: Se o jogador for comprar uma carta e não houverem mais cartas no seu deck, ele perde o jogo;
- Quantidade inicial de cartas na mão por jogador: 5;
- Quantidade inicial de cartas no deck por jogador: 10;

2.3.3 Fases de um turno

O turno de um jogador (jogador ativo) é definido da seguinte forma:

- Início: Caso o jogador esteja no estado "Compra de cartas", ele compra uma carta. Em seguida, poderá utilizar uma carta de sua mão ou passar essa fase;

- Dano: Em caso de peças sobrepostas, a peça com a movimentação mais antiga será retirada. Se o rei for removido, o jogo acaba;
- Movimento: Escolher uma peça e movimentá-la ou passar essa fase;
- Fim: Se o jogador não tiver usado cartas na primeira fase do turno, poderá usar uma carta ou terminar o turno;

2.3.4 Cartas

- "Sair pela tangente": Movimentar uma peça própria 1 casa na diagonal. Não surte efeito quando a peça foi afetada pela carta "Mãos ao alto!";
- "Movimento brusco": Movimentar uma peça própria 1 casa em cruz. Não surte efeito quando a peça foi afetada pela carta "Mãos ao alto!";
- "Salve o rei!": Troque as posições de uma torre sua e do seu rei. Não surte efeito quando a uma das peças foi afetada pela carta "Mãos ao alto!"; Esse efeito não conta como uma movimentação. Se sobrar apenas o rei, a carta será usada e descartada;
- "Silêncio, por favor": Durante o próximo turno, o jogador adversário não pode usar cartas;
- "Mãos ao alto!": Durante o próximo turno, uma peça não pode se mover mesmo com o uso de cartas;
- "Corredores experientes": Uma peça própria poderá se deslocar 1 casa a mais na fase de movimentação no turno atual;
- "Escudos!": Não ocorre dano no turno atual;
- "Sacrifício necessário": Sacrifique uma peça para destruir uma peça que não seja um rei;

Capítulo 3

Visão geral do Sistema

3.1 Arquitetura da aplicação

Aplicação orientada a objetos com o suporte de um sistema distribuído do tipo cliente - servidor.

3.2 Premissas de desenvolvimento

- O programa deve possuir uma interface gráfica;
- A linguagem de programação adotada para o desenvolvimento deve ser Java;
- O programa deve ser executado distribuído com o suporte de NetGamesNRT;
- A modelagem deve ser baseada em UML;

Capítulo 4

Requisitos funcionais da aplicação

4.1 Conexão com o servidor

- Conectar-se ao servidor: Ao selecionar a opção "Conectar" no menu da interface gráfica, o programa deverá estabelecer uma conexão com o servidor da NetGamesNRT para dois jogadores disputarem uma partida, sendo em computadores diferentes ou não;
- Iniciar partida: A interface gráfica deverá possuir uma opção de "Iniciar partida" no menu, cuja funcionalidade será enviar uma solicitação ao servidor da NetGamesNRT se estiver previamente conectado. O programa deverá esperar a resposta do servidor para começar a partida;
- Receber solicitação de início: Ao receber a informação do servidor de que a partida deve começar, o programa deverá notificar o usuário sobre o início da partida. Posteriormente, o jogo é instanciado e o jogador é associado a ele. As peças e o controlador de cartas serão instanciados e associados. O controlador de cartas deverá sortear cinco cartas, que serão retiradas do deck, e associar ao jogador;
- Desconectar: O programa deverá ter a opção de desconectar do servidor no menu se o jogador estiver conectado. Caso existir uma partida em andamento, o jogador que foi desconectado do servidor perderá a partida e o oponente ganhará.
- Receber jogada: O programa deve receber as informações do servidor sobre a última jogada do adversário e atualizar a interface para mostrar os acontecimentos do turno, atualizando as posições das peças e se ela foi retirada do tabuleiro, o número de cartas tanto na mão como no deck de cada jogador e se uma carta foi utilizada. Além disso, deve-se verificar o término do jogo, caso aconteça, deve ser notificado aos jogadores o vencedor e a partida deve ser encerrada;

4.2 InterfaceJogador

- Usar uma carta: Se o jogador não estiver sob o efeito da carta "Silêncio, por favor", poderá utilizar uma carta, caso contrário, será notificado sobre essa impossibilidade e a próxima fase será "Movimento". Ao usar uma carta que afeta o turno atual, como "Sair pela tangente", o programa deverá aplicar o efeito dessa carta antes da fase de movimentação. Para as cartas que influenciam o turno adversário, o efeito deverá ser armazenado e aplicado na primeira fase do oponente. Nas duas situações anteriores, o controlador de cartas deverá remover a carta usada da mão e colocá-la no descarte. As cartas que afetam peças deverão ser escolhidas primeiro, uma mensagem será enviada pedindo a escolha de uma peça. A próxima fase será "Movimento";
- Mover uma peça: Ao clicar com o botão direito sobre uma peça no tabuleiro, a interface deverá mostrar as posições possíveis [4.4] e o jogador poderá clicar com o mesmo botão sobre uma dessas posições para movê-la. Se o jogador tentar sobrepor uma peça aliada ou pular uma adversária, a ação será negada e ele será notificado sobre a impossibilidade. Depois disso, o jogador poderá escolher outra peça. Por fim, o tabuleiro será atualizado e a próxima fase será o "Usar uma carta" se nenhuma carta foi utilizada anteriormente, caso contrário, o turno terminará;
- Passar etapa: Deverá existir um botão com o texto "Passar etapa" para que o usuário passe uma etapa do jogo quando quiser, sendo que a próxima dependerá das regras do jogo.

4.3 Utilização de uma carta pelo jogador ativo

- "Sair pela tangente": O jogador deverá escolher uma das posições fornecidas pela interface [4.4] em diagonal referente à posição atual da peça escolhida e depois o programa a movimentará, atualizando o tabuleiro;
- "Movimento brusco": O jogador deverá escolher um das posições fornecidas pela interface [4.4] em cruz referente à posição atual da peça escolhida e depois o programa a movimentará, atualizando o tabuleiro;
- "Salve o rei!": Ao escolher uma peça aliada, exceto o Rei, o sistema deverá trocar as duas peças, a escolhida e o Rei, de lugar no tabuleiro e atualizá-lo. Caso o jogador tente escolher uma peça adversária, uma notificação será enviada avisando sobre essa impossibilidade. E se restar apenas o Rei, a carta será usada mas sem efeito sobre o tabuleiro;
- "Silêncio, por favor": O programa deverá impedir o oponente de usar cartas no próximo turno dele;
- "Mãos ao alto!": O jogador deverá escolher uma peça adversária e o programa impedirá essa peça de se mover no turno do oponente;
- "Corredores experientes": Deve-se escolher uma peça aliada para adicionar uma unidade no alcance de movimento no turno, e quando ele acabar, o alcance deverá voltar ao normal;
- "Escudos!": A fase "Dano" [2.3.3] será ignorado no turno;
- "Sacrifício necessário": Primeiramente, o jogador deverá escolher uma das peças aliadas e depois uma adversária seguindo as regras da carta [2.3.4]. Por último, o tabuleiro deve ser atualizado.

4.4 Interface Gráfica

- Mostrar movimentação da peça: Ao clicar em uma peça, a interface deverá preencher os quadrados do tabuleiro de verde para as posições que ocorreria uma movimentação válida [2.3.2] e de vermelho para peças aliadas e adversárias que estejam no alcance do movimento;
- Mostrar descrição de uma carta: Quando o jogador passar o mouse sobre uma carta da mão, uma descrição deverá aparecer;

Capítulo 5

Requisitos não-funcionais da aplicação

- Paradigma: Orientação a objetos;
- Linguagem de programação: Java;
- Ferramenta de modelagem: Visual Paradigm
- Interface Gráfica:
 - Utilização da biblioteca Swing do Java;
 - 2D;
 - Visualização da quantidade de cartas na mão e no deck;
 - Mostrar a fase atual do turno e nome dos jogadores;
 - Exibição das cartas da mão apenas para o respectivo jogador;
- Plataforma de conexão entre jogadores: NetGamesNRT;
- IDE: Eclipse;