

TRABALHO FINAL

JOGO DE TRUCO

Relatório apresentado como requisito parcial da
Disciplina CSI032- Programação de computadores || do
curso de Engenharia da Computação, Ministrada pelo
professor Rafael Alexandre.

Discente: Adriana Mirian Mendes Cardoso

Matrícula: 16.2.5915

João Monlevade
Agosto/2017

SUMÁRIO

1	INTRODUÇÃO	3
2	INTRODUÇÃO AO JOGO	4
2.1	Regras gerais	5
3	DIAGRAMA DE CLASSES.....	5
3.1	Procedimentos.....	5
4	CONCLUSÃO.....	6
	REFERÊNCIAS	7

1. INTRODUÇÃO:

O presente trabalho tem por objetivo implementar um jogo de cartas em Java, o Truco, utilizando o Eclipse. Buscou-se implementar o jogo com o paradigma atual mais utilizado na construção de softwares, a orientação a objetos .

Dentre suas vantagens, podemos citar o reaproveitamento de código e aumento da manutenibilidade dos sistemas assim desenvolvidos.

2. INTRODUÇÃO AO JOGO:

2.1 Regras gerais

O truco tem como particularidade o fato de que não é jogado a dinheiro, mas por pontos. Ele é disputado em mãos que são divididas em três rodadas. Quando um jogador ou mais alcançam ou ultrapassam os 12 pontos, vencem a rodada. Em cada rodada os jogadores colocam uma de suas cartas na mesa e vence quem tiver a de valor mais alto. Cada mão tem um valor que pode variar de 1 a 9 pontos, variando de acordo com as apostas feitas.

O truco mineiro utiliza um baralho sem as cartas 8, 9, 10 e curinga, totalizando 40 cartas. As 4 cartas mais altas do jogo são chamadas de manilhas, conforme a ordem abaixo:

- 4 de paus: Maior carta do jogo, chamada de zap.
- 7 de copas: Segunda maior carta do jogo.
- A de espada: Terceira maior carta do jogo, chamada de espadilha.
- 7 de ouros: Quarta manilha.

Depois das manilhas vem as outras cartas. O valor delas é dado pelo número apenas, o naipe não importa. Conforme a ordem abaixo:

- 3 todos os naipes
- 2 todos os naipes
- A todos os naipes, exceto espada que é manilha
- K todos os naipes
- J todos os naipes
- Q todos os naipes
- 7 somente paus e espada
- 6 todos os naipes
- 5 todos os naipes
- 4 todos os naipes, exceto paus que é o zap

No início da partida as cartas são embaralhadas. Após embaralhar, cada jogador recebe 3 cartas e em seguida uma pessoa começa jogando uma carta na mesa. A vez de iniciar é alternada a cada partida.

Depois que uma carta é lançada na mesa o adversário precisa mandar a carta dele. Depois que todos jogaram finaliza a rodada. Ganha quem mandou a carta de maior valor. Quem conseguir vencer 2 rodadas ganha a partida e recebe 2 pontos.

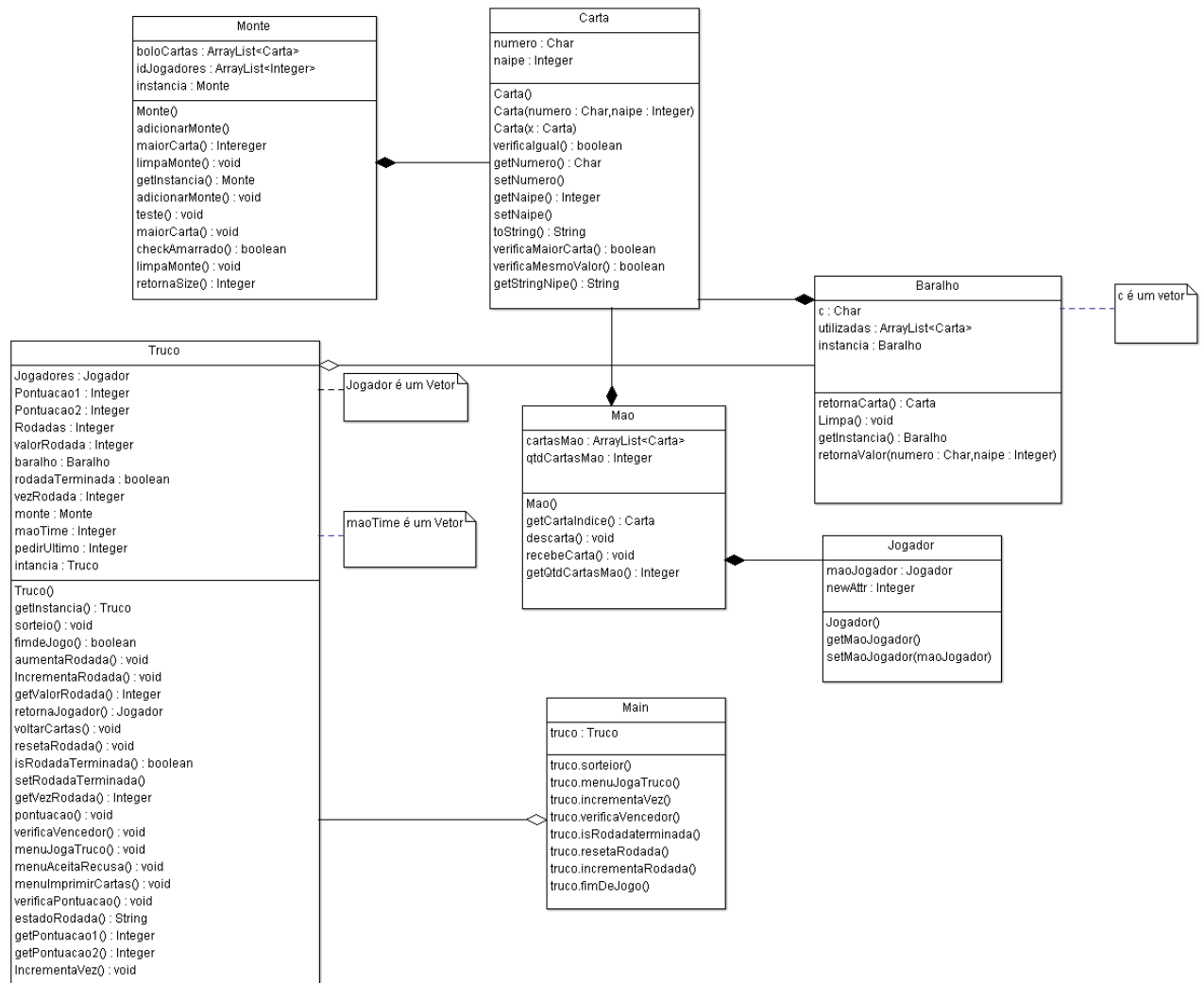
Terminada a partida uma nova é iniciada, embaralhando e distribuindo as cartas. Isso ocorre até que um dos jogadores complete 12 pontos. Quando um jogador completa 12 pontos ele vence um jogo e todos tem os pontos zerados.

A cada 2 jogos ganhos, marca-se uma queda e zera os jogos e pontos de todos.

Quando as cartas jogadas tem o mesmo valor ocorre o empate. Se ocorrer na primeira rodada, quem empatou deve mostrar a maior carta, em seguida o adversário. O maior valor ganha. Caso ocorra empate novamente os jogadores devem mostrar a terceira carta. Quando o empate ocorre na segunda ou na terceira jogada, ganha quem fez a primeira.

Quando um dos jogadores pede truco ele está pedindo para aumentar o valor da partida para 4 pontos. O adversário pode aceitar ou correr. Se aceitar, a partida fica valendo 4, ao invés de 2. Se correr, a partida acaba sem ninguém mostrar mais as cartas e quem pediu truco ganha 2 pontos. Se uma pessoa truca, somente o adversário pode ainda aumentar o valor, pedindo seis a qualquer momento. A partida fica, dessa forma, valendo 8 pontos, mas somente se o outro aceitar. Quando um jogador não aceita o seis e corre ele perde a partida, que está valendo 4 pontos. Da mesma forma acontece com o nove que vale 10 pontos e o doze vale 12 pontos.

3. DIAGRAMA DE CLASSES:



3.1 Procedimentos

O jogo apresenta suas principais classes que são a “Truco”, “Jogador”, “Main”, “Monte”, “Mao”, “opacaoInvalidaException”, “Baralho”, “Carta”.

Utilizou-se o **Math.random** para embaralhar as cartas, onde cada carta tem um numero e naipe.

Para criar os quatro jogadores do Time 1 e do Time 2, utilizou-se um vetor de Jogadores. Que foi instanciado em uma das classes do Jogo chamada Truco.

Na classe “Baralho” utilizou-se “**ArrayList**” apenas das cartas utilizadas no baralho. Contudo para gerar uma carta aleatória, utilizou-se o método retornaCarta(), que sorteia uma carta aleatória do baralho.

Ainda na classe Baralho, verifica-se uma método chamado retornaValor(),

que associa o peso de cada carta para verificar quem vence a rodada.

Na classe Carta, temos construtores da classe que recebe numero e naipe da carta.

Inicialmente, temos uma classe chamada “Main” que possui a chamada de todos os métodos principais para iniciar o jogo. Temos dois Menus inicialmente, que são chamados de “truco”.

O Jogo inicialmente, foi implementado utilizando “*ArrayList*”, classes bem distribuídas entre Jogadores, procedimento de jogo e baralho. Buscou-se implementar o projeto utilizando herança, padrão de projeto, tratamento de exceção, polimorfismo, sobrecarga e sobrescrita de métodos.

Uma das principais preocupações inicialmente era verificar se o jogo estava amarrado, ou seja, se existia empate. Diante disso criou-se um método para verificar se o jogo estava amarrado e caso sim, existe outro método na classe truco responsável por verificar o vencedor de acordo com as cartas na mão de cada time.

A relação de herança do projeto do Truco foi na classe “*opacaoInvalidaException*” que “*extends*” de *Exception*. Essa classe tem a função de levantar um erro de exceção durante a execução do programa.

Verificou-se que algumas classes do jogo só precisavam ser instanciadas apenas uma vez, sendo assim, foi utilizado o padrão de projeto *singleton* para instanciar apenas um objeto de “baralho” e “monte”.

Outra parte essencial e crucial do jogo foi verificar o valor da rodada e a atribuir a pontuação correta para cada time. Foram criados alguns atributos e métodos na classe Truco para verificar o valor de cada rodada e atribuição do valor correto da pontuação e da vez da rodada para cada Jogador. Onde o primeiro time que chegasse a 12 pontos vencia o Jogo.

Durante a criação do projeto, compreendi que a dificuldade de se fazer jogos em Java é trabalhar com 3D. A depender do tipo de Jogo, creio que Java pode se sair muito melhor que C++.

Apesar de Java ter um desempenho similar e algumas vezes até superior ao C++ em certos aspectos, tem um deles que tem importância vital nos jogos e é onde Java perde: operações com números flutuantes.

4. CONCLUSÃO

Conclui-se que com a programação em Java orientada a objetos, pode-se verificar maior facilidade para reutilização de código e por consequência do projeto.

Verificou-se que programas grandes são difíceis de manter e por isso, é um bom hábito separá-los em unidades mais ou menos isoladas. Em Java, isso é feito utilizando objetos, que são compostos por atributos e métodos definidos a partir de classes, que por sua vez são organizadas em pacotes. Pude verificar inicialmente uma certa dificuldade, mas posteriormente tornou-se mais fácil desenvolver o jogo.

5. REFERÊNCIAS:

O TRUCO MINEIRO. *Principais regras do Truco*. Disponível em <http://www.trucomineiro.com.br/regras.php> acesso em 02 de ago de 2017.