

Projeto Cassino Solitário

EXA 836 – Padrões e Frameworks – UEFS, turma 2018.1

Gustavo Boanerges
Universidade Estadual de Feira de Santana
Graduando em Engenharia de Computação
Feira de Santana, BA, Brasil
g.boanerges2@gmail.com

Jadson Firmo
Universidade Estadual de Feira de Santana
Graduando em Engenharia de Computação
Feira de Santana, BA, Brasil
jadson.uefs@gmail.com

Resumo—Este documento descreve a construção de um jogo paciência em linguagem Java requisitado pelo professor José Amâncio, referente à disciplina de Padrões e Frameworks do curso de Engenharia de Computação da Universidade Estadual de Feira de Santana.

Palavras-chave—java, games, jogos, cartas, paciência

I. INTRODUÇÃO

O jogo de paciência é um dos jogos mais populares de todos os tempos, sobretudo devido a sua disponibilidade gratuita e *offline* em sistemas operacionais da Microsoft, apresentando inclusive algumas variações de estilo de jogo (como o Paciência *Spider*, por exemplo).

Trata-se de um jogo de cartas para um único jogador, que compete contra seus próprios números, tentando resolver o jogo em menos tempo ou em menor quantidade de jogadas.

O presente relatório descreve a construção de um jogo de paciência (estilo de jogo tradicional) desenvolvido em linguagem Java, com interface totalmente caracter, acessível através de console de IDE's ou linha de comandos.

O objetivo do jogo é construir 4 pilhas de naipes em ordem ascendente, começando com os ases nas “fundações”, a fim de serem completados os 4 naipes – cada fundação deve receber um único naipe de cartas.

O baralho possui 52 cartas e 4 naipes (espadas, copas, paus e ouros). Na “mesa de jogo” deve haver um estoque de cartas “viradas para baixo” (inicialmente 26 cartas) e uma pilha de descartes onde a jogada ocorre (são exibidas as cartas “viradas para cima” para que o jogador possa montar suas combinações). Deve existir também quatro “fundações” (inicialmente vazias) para receber cada conjunto de carta de acordo com seu naipe, ou seja, cada fundação deve receber um naipe completo.

Além disso, na “mesa de jogo”, deve haver sete pilhas de fileiras de cartas com cartas “viradas para baixo” e a primeira carta “virada para cima”. Nestas pilhas 28 cartas são distribuídas da esquerda para a direita, de modo a respeitar a seguinte configuração: 1 carta na primeira pilha, 2 cartas na segunda pilha, 3 cartas na terceira pilha e assim por diante até 7 cartas na última pilha da direita. A Figura 01 ilustra a “mesa de jogo”.

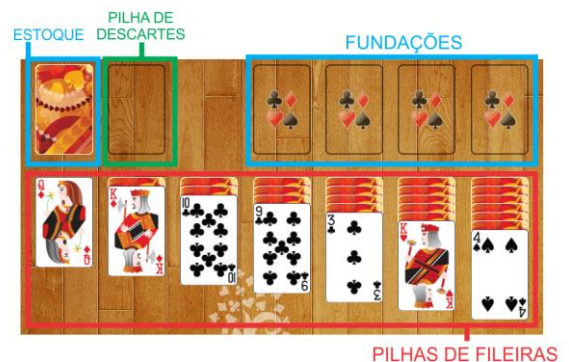


Figura 01: Ilustração das partes que compõe a “mesa de jogo”.

Uma jogada será criada virando as cartas do estoque e colocando-as com a face para cima na pilha de descarte. As fundações devem receber as cartas de cada naipe, começando com um ás e seguindo por ordem ascendente (por exemplo: A, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K de copas, de baixo para cima).

As pilhas de fileiras devem alternar entre cartas de naipes pretos (espadas e paus) e vermelhos (copas e ouros), e devem ser construídas sob pontuação descendente. Portanto, o jogador pode colocar tanto um Q de copas ou um Q de ouros em um K paus, mas não um K de paus ou K de espadas, já que são de naipes de mesma cor. A figura 02 exemplifica como deve ocorrer a alternância entre as cartas.



Figura 02: Ilustração da alternância correta entre naipes nas pilhas de fileiras.

A Pilha de descarte é construída virando a carta do topo do estoque e colocando-a com a face para cima na pilha de descarte. A pilha de descarte é reduzida jogando-se a carta maior na pilha de fileira ou numa fundação.

Com o início do jogo, o baralho será criado, com suas 52 cartas, e separado nas pilhas de fileiras e o restante das cartas formará a pilha de estoque. Em seguida as últimas cartas (cartas do topo) das pilhas de fileiras serão exibidas, e o usuário poderá começar a jogar. A pilha de estoque exibirá uma ou três cartas, ajustável ao início de cada partida: se for pedida uma carta para estoque e a atual não for utilizada, esta será enviada para a pilha de descarte. Em cada movimentação, as pilhas de fileiras e fundações irão verificar os valores das cartas, se a sequência é permitida, tanto da numeração quanto do naipe.

É importante destacar que o projeto deve sempre esperar por implementações futuras, então precisa ser de fácil manutenção e o menos específico possível, facilitando a implementação de novos jogos dentro do mesmo projeto (Ex: Inclusão do jogo paciência Big Bertha).

As sessões a seguir detalham a metodologia de construção do jogo e apresentam os resultados e discussões e as considerações finais acerca do projeto desenvolvido.

II. METODOLOGIA E RESULTADOS

O jogo foi desenvolvido através da IDE Netbeans, uma das IDE's mais utilizadas para desenvolvimento de projetos em Java, devido a sua interface amigável e sua alta compatibilidade com a referida linguagem de programação. Como o trabalho foi realizado em dupla, foi necessária a utilização do Git para permitir o versionamento do código.

A construção do código partiu dos requisitos estipulados para o projeto, contemplando basicamente as regras do jogo paciência.

Com a perspectiva de tornar o código modularizado e de fácil compreensão, foi tomada a decisão de projeto de trabalhar com a metodologia MVC (*Model-View-Controller*), de modo que a classe principal (*main*) é a *view*, responsável por fazer a mediação entre usuário e sistema, proporcionando a execução do programa e gerenciando as ações escolhidas pelo jogador (Ex: Iniciar jogo, mover carta etc). O *controller* (denominado Mesa) é responsável pela inicialização do jogo, contendo a lógica de distribuição das cartas. E as *models* são as classes que definem cada objeto do jogo (Ex: Baralho, Estoque, Fileira etc).

Na classe *main*, foram criados menus manuseados a partir da captação de respostas do usuário no console (ou linha de comando) através da utilização da biblioteca scanner do Java. Esses menus são exibidos na tela para usuário a partir das escolhas realizadas. Cada escolha efetuada pelo jogador, de acordo com o que o menu exhibe, foi gerenciada com a utilização de switch-case, que permite tratamentos diferenciados para cada caso de opção de menu escolhida.

No menu principal, o usuário tem a opção de jogar ou de sair do sistema. Este menu foi pensado desta forma para possibilitar a inserção de outros jogos futuramente, o que daria ao jogador a possibilidade de escolher qual jogo desejaria jogar. A Figura 1 mostra o exemplo do funcionamento do menu com a inserção de mais um jogo, o Big Bertha.

```
run:
MENU PRINCIPAL

Escolha uma das opcoes abaixo:
1 - Jogar Paciencia
2 - Jogar Big Bertha
3 - Sair

Digite a opção: 1

=====
BEM VINDO AO PACIENCIA!
=====
>> Criando cartas...
>> Embaralhando cartas...

Escolha a quantidade de cartas a serem exibidas no descarte: 1 ou 3 carta(s)?

Digite a opcao: 1
>> Iniciando Fileiras...
>> Criando Estoque...
>> Criando Fundacoes...

=====
INICIANDO...
=====
```

Figura 1: Menu principal

Quando o jogador escolhe a opção jogar (Paciência), o sistema começa a montar o jogo, e então pergunta ao usuário qual configuração de jogo deseja com relação a quantidade de cartas que deseja que sejam exibidas no descarte (1 ou 3). Após selecionar uma das opções de descarte, o jogo enfim é inicializado. O jogador visualiza a mesa de jogo, onde estão posicionados o estoque, o descarte, as fundações e as fileiras. Logo abaixo, mais um menu é exibido, desta vez para as ações de jogo, onde o usuário pode exibir a mesa de jogo, mover cartas, atualizar o descarte ou encerrar o jogo, como mostra a Figura 2.

```
=====
INICIANDO...
=====

1 - Estoque: [22 carta(s)]
[<>] | [<>] | [<>] | [<>] | [<>] | [<>] | [<>] | [<>] | [<>] | [<>] | [<>] | [<>] |
2 - Descarte: [1 carta(s)]
5 PAUS

3 - Fundacao:
4 - Fundacao:
5 - Fundacao:
6 - Fundacao:

7 - Fileira: Q OUROS |
8 - Fileira: [<>] | 3 ESPADA |
9 - Fileira: [<>] | [<>] | 9 PAUS |
10 - Fileira: [<>] | [<>] | [<>] | 4 COPAS |
11 - Fileira: [<>] | [<>] | [<>] | [<>] | K ESPADA |
12 - Fileira: [<>] | [<>] | [<>] | [<>] | [<>] | 3 COPAS |
13 - Fileira: [<>] | [<>] | [<>] | [<>] | [<>] | [<>] | A PAUS |

Escolha uma das opcoes abaixo:
1 - Exibir Mesa de Jogo
2 - Mover Carta
3 - Mover Conjunto de Cartas
4 - Atualizar Descarte
5 - Sair do Jogo

Digite a opcao: |
```

Figura 2: Jogo inicializado

A opção exibir mesa de jogo apenas mostra ao usuário novamente a tela inicial do jogo inicializado (Estoque, descarte, fundações e fileiras) para que o jogador possa pensar nas jogadas possíveis, analisando a interface.

A opção mover carta, permite ao usuário movimentar uma única carta respeitando algumas regras:

1. Estoque só pode movimentar carta para descarte;
2. Descarte pode movimentar carta para fundações ou fileiras;
3. Fundações só podem movimentar cartas para fileiras;
4. Fileiras podem movimentar cartas para outras fileiras ou para fundações.

5. Não é permitida a movimentação de uma origem para ela mesma.

As movimentações também seguem os requisitos de sequência entre as cartas:

1. Nas fileiras as cartas devem ser intercaladas entre naipes de cores diferentes, e em valores decrescentes de cartas. Uma fileira vazia só pode receber cartas de valor rei (K), logo depois rainha (Q) de um naipe diferente, e assim sucessivamente, chegando no máximo até o ás (A).
2. Nas fundações as cartas devem ser separadas por naipes: cada fundação com um único naipe. Devem respeitar também a ordem crescente, começando sempre pelos ases (A).

Outra opção desenvolvida, é a de movimentação de conjuntos de cartas. Essa funcionalidade permite que o usuário mova uma sequência de cartas já organizadas para outra fileira (sempre respeitando as regras de sequência das cartas). Essa função só é válida para movimentação de cartas entre fileiras.

Já a opção de atualizar descarte funciona da seguinte forma: quando o usuário precisa chamar cartas do estoque para o descarte ele deve usar essa opção, assim uma carta é retirada do estoque e exibida no descarte, permitindo movimentação para fileiras ou fundações. Na frente da exibição do estoque e de descarte é mostrado ao usuário a quantidade de cartas de cada uma dessas pilhas. Quando o estoque fica vazio, se no descarte houver uma quantidade de cartas maior do que o usuário selecionou como número de cartas a ser exibidas no estoque, o jogador pode selecionar a opção de atualizar descarte novamente, desta forma o sistema restaura o estoque como mostram as Figuras 3 e 4.

```
=====
DESCARTE ATUALIZADO!
=====

1 - Estoque: [VAZIO]

2 - Descarte: [23 carta(s)]
  6 ESPADA

3 - Fundacao:
4 - Fundacao:
5 - Fundacao:
6 - Fundacao:

7 - Fileira: Q OUROS |
8 - Fileira: [<>] | 3 ESPADA |
9 - Fileira: [<>] | [<>] | 9 PAUS |
10 - Fileira: [<>] | [<>] | [<>] | 4 COPAS |
11 - Fileira: [<>] | [<>] | [<>] | K ESPADA |
12 - Fileira: [<>] | [<>] | [<>] | [<>] | 3 COPAS |
13 - Fileira: [<>] | [<>] | [<>] | [<>] | [<>] | A PAUS |

Escolha uma das opcoes abaixo:
1 - Exibir Mesa de Jogo
2 - Mover Carta
3 - Mover Conjunto de Cartas
4 - Atualizar Descarte
5 - Sair do Jogo

Digite a opcao: |
```

Figura 3: Estoque vazio

```
=====
DESCARTE ATUALIZADO!
=====

1 - Estoque: [22 carta(s)]
 [<>] | [<>] | [<>] | [<>] | [<>] | [<>] | [<>] | [<>] | [<>] | [<>] | [<>] |
2 - Descarte: [1 carta(s)]
  6 ESPADA

3 - Fundacao:
4 - Fundacao:
5 - Fundacao:
6 - Fundacao:

7 - Fileira: Q OUROS |
8 - Fileira: [<>] | 3 ESPADA |
9 - Fileira: [<>] | [<>] | 9 PAUS |
10 - Fileira: [<>] | [<>] | [<>] | 4 COPAS |
11 - Fileira: [<>] | [<>] | [<>] | [<>] | K ESPADA |
12 - Fileira: [<>] | [<>] | [<>] | [<>] | 3 COPAS |
13 - Fileira: [<>] | [<>] | [<>] | [<>] | [<>] | A PAUS |

Escolha uma das opcoes abaixo:
1 - Exibir Mesa de Jogo
2 - Mover Carta
3 - Mover Conjunto de Cartas
4 - Atualizar Descarte
5 - Sair do Jogo

Digite a opcao: |
```

Figura 4: Estoque restaurado

Por fim, a opção sair do jogo, apenas encerra o sistema.

O jogo contém uma regra de vitória que é definida a partir da quantidade de cartas nas fundações: quando as fundações somam juntas 52 cartas significa que o jogo foi concluído com êxito, ou seja, o jogador venceu a partida, desta forma uma mensagem de confirmação da vitória é exibida e o sistema é encerrado.

A Figura 5 ilustra um jogo em andamento, onde é possível notar todo o esquema de movimentação de cartas, respeitando as regras pré-estabelecidas.

```
=====
CARTA MOVIDA COM SUCESSO!
=====

1 - Estoque: [14 carta(s)]
 [<>] | [<>] | [<>] | [<>] | [<>] | [<>] | [<>] | [<>] | [<>] | [<>] | [<>] |
2 - Descarte: [4 carta(s)]
  4 COPAS

3 - Fundacao: A OUROS |
4 - Fundacao:
5 - Fundacao:
6 - Fundacao:

7 - Fileira:
8 - Fileira: [<>] | Q COPAS | J PAUS | 10 COPAS |
9 - Fileira: [<>] | [<>] | J COPAS | 10 PAUS | 9 COPAS |
10 - Fileira: [<>] | 4 OUROS | 8 PAUS | 4 OUROS |
11 - Fileira: [<>] | [<>] | [<>] | 4 COPAS |
12 - Fileira: [<>] | [<>] | 4 COPAS | 3 ESPADA | 2 OUROS | A PAUS |
13 - Fileira: [<>] | [<>] | [<>] | [<>] | 8 PAUS | 7 OUROS | 6 PAUS | 5 OUROS | 4 PAUS |

Escolha uma das opcoes abaixo:
1 - Exibir Mesa de Jogo
2 - Mover Carta
3 - Mover Conjunto de Cartas
4 - Atualizar Descarte
5 - Sair do Jogo

Digite a opcao: |
```

Figura 5: Jogo em andamento

III. CONSIDERAÇÕES FINAIS

O projeto não está totalmente adequado para receber novos jogos, a implantação de um padrão de projetos seria uma forma de resolver este problema, tornando o código mais manutenível e possibilitando a redução da complexidade do código, reduzindo por exemplo a quantidade de laços condicionais.

A etapa de testes de unidade não foi realizada, foram realizados apenas testes de mesa, observando o funcionamento do jogo ao passo que as implementações foram realizadas.

Como melhorias para projetos futuros, além da adaptação para um padrão também poderiam ser incrementadas mais funcionalidades ao jogo, como um contador de movimentações de cartas e um contador de tempo de cada

partida. Seria interessante também a implantação de uma regra para informar ao usuário que as movimentações possíveis acabaram (ou seja, game-over).

REFERÊNCIAS

- [1] Java e Orientação em Objetos. Caelum Cursos. Disponível em: <https://www.caelum.com.br/apostila-java-orientacao-objetos/> - Acesso em: 06/2018.