

Algoritmos e Programação I

Trabalho Final – Parte 2

Pif-Paf Vil

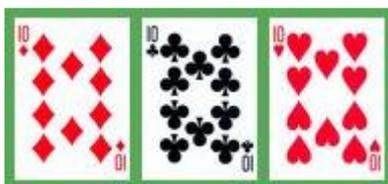
O objetivo desta parte 2 do Trabalho Final é implementar o jogo de cartas Pif-Paf Vil, utilizando a classe Carta (que é fornecida) e as classes GrupoDeCartas, Jogador, PifPafVil e Cassino, que cada aluno deve, individualmente, desenvolver para permitir diversas rodadas de Pif-Paf Vil, entre dois jogadores, simulando um cassino.

As regras do jogo

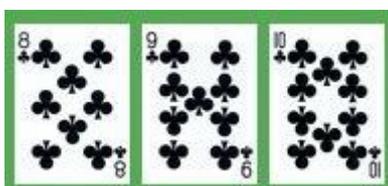
Pif-Paf Vil é um jogo de cartas para dois jogadores, usando-se dois baralhos completos com 104 cartas no total. Um dos jogadores é virtual, chamado “Mestre Vil” (daqui em diante “Mestre”). O outro jogador é humano (daqui em diante “Você”). O primeiro jogador a jogar na primeira rodada é sorteado ao tirar (aleatoriamente) a maior carta do baralho. Nas outras rodadas esta ordem se alterna. No início do jogo, cada jogador tem sete “vidas” para apostar.

No início de cada rodada, as cartas devem ser embaralhadas e 9 cartas são, alternadamente, distribuídas a cada jogador, sempre a partir do topo (primeira carta) do baralho, sendo que o primeiro jogador a jogar recebe a primeira carta. O grupo das cartas restantes é colocado no “monte” para compras. As cartas descartadas, durante uma rodada, vão para o “descarte”. Quando o “monte” fica vazio, o “descarte” o substitui na ordem em que está, sendo que a última carta descartada será a última carta do novo “monte”.

Cada jogador só pode ver suas próprias cartas e a última descartada. O objetivo é formar trincas ou sequências. Uma trinca é constituída por 3 cartas de mesmo valor, mas de naipes diferentes. Por exemplo:



Uma sequência é constituída por 3 cartas do mesmo naipe, mas com valores em sequência (é permitida também a sequência Q K A). Por exemplo:



Após a distribuição das cartas, cada jogador analisa e organiza as suas e aposta uma “vida”. Ao final de cada rodada, o vencedor recolhe as duas “vidas” apostadas.

Cada jogador, na sua vez, deve:

- comprar uma carta do topo (primeira) do “monte” ou, se houver, a última carta descartada no “descarte”,
- analisar e organizar, se desejar, suas cartas e
- descartar uma delas no “descarte”.

Uma rodada termina quando um dos jogadores “bate”, ou seja, vence ao descartar uma carta ficando na “mão” com 3 jogos completos, por exemplo, duas trincas e uma sequência ou qualquer outra combinação de trincas e sequências.

O jogo termina quando um dos jogadores desiste do jogo ou quando não tem mais “vidas”. “Mestre” nunca desiste de uma rodada nem do jogo.

Sugestões para as classes Carta, GrupoDeCartas e Jogador

Carta
numero: int naipe: int
Carta(int,int) getNaipe(): int getNumero(): int getNaipeChar(): char getNaipeString(): String getNome(): String getNomeExtenso(): String

Jogador
nome: String mao: GrupoDeCartas vidas: int
Jogador(String) getNome(): String setMao(GrupoDeCartas): void getMao(): GrupoDeCartas compra(Carta): boolean descarta(Carta): boolean getVidas(): int finalizaRodada(int): void

GrupoDeCartas
grupo: ArrayList<Carta>
GrupoDeCartas() getCartas(): ArrayList<Carta> setCartas(ArrayList<Carta>): void incluirCarta(Carta): void excluirCarta(Carta): boolean retiraCartaTopo(): Carta getCarta(int): Carta embaralha(): void ordenaPorNumero(): void ordenaPorNaipe(): void estaVazio(): boolean trocaPosicao(int,int): boolean mostraCartas(): void

Na classe Jogador, o método setMao recebe um grupo de cartas a ser incluído na “mão”. O método compra recebe uma carta (retirada do “monte” ou do “descarte”) e a inclui na “mão”. O método descarta recebe uma carta que deve ser retirada da “mão”, ou seja, descartada. O método finalizaRodada recebe o valor que deve ser somado ou diminuído ao atributo “vidas”. Este método também acerta todos os outros atributos para a próxima rodada.

Na classe GrupoDeCartas, o método construtor cria uma lista vazia. O método retiraCartaTopo retira a carta da primeira posição do grupo. O método getCarta recebe uma posição e retorna a carta desta posição do grupo. O método embaralha re-organiza o atributo “grupo” colocando suas cartas em posições aleatórias na lista. O método trocaPosição recebe as posições originais de duas cartas e as troca.

As classes PifPafVil e Cassino

A classe PifPafVil deve ter atributos e métodos relacionados às regras do jogo, permitindo dar apoio às decisões a cada etapa das rodadas e do jogo. Os seguintes métodos são obrigatórios:

- bate(GrupoDeCartas): boolean
Recebe um grupo de cartas e retorna verdadeiro se houver três jogos que permitam a “batida”, mesmo que as cartas estejam desorganizadas.
- descartaPior(GrupoDeCartas): Carta
Recebe um grupo de cartas e retorna a carta que tem menor probabilidade de compor um jogo (trinca ou sequência).

“Você” pode solicitar o método descartaPior ou decidir manualmente qual carta será descartada. “Mestre” sempre solicita o método descartaPior para fazer seu descarte.

A classe Cassino deve ter o método main e o método executa. Neste deve ser simulado um cassino para as rodadas de Pif-Paf Vil entre “Mestre” e “Você”, com as seguintes tarefas:

- instanciar um grupo de cartas com dois baralhos (com 52 cartas cada um, de Ás a Rei em 4 naipes);
- definir os dois jogadores com seus nomes e suas sete “vidas”;
- iniciar o jogo;
- iniciar as rodadas, sempre re-embaralhando as cartas;
- permitir opções aos jogadores em suas vezes, orientando-os com as informações necessárias de acordo com as regras do jogo, e, ao final,
- informar o total de “vidas” de cada jogador e quem é o vencedor.

Critérios de avaliação

- Jogadores (2 pontos)
 - atributos, construtor e consultas simples
 - gerenciamento da mão
 - gerenciamento de vidas e apostas
- Grupos de cartas (2 pontos)
 - atributo, construtores e consulta
 - atualização do grupo
 - embaralhamento
- Pif-Paf Vil (3 pontos)
 - atributos e construtor
 - controle do jogo
 - definição da batida
 - descarte da pior carta
- Cassino (3 pontos)
 - geração de baralhos
 - definição de jogadores
 - controle das rodadas
 - opções e informações aos jogadores
 - informações finais