

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS

Algoritmos e Estruturas de Dados – 2/2023

TRABALHO PRÁTICO

Prof. Edwaldo Soares Rodrigues

Instruções:

- I. O trabalho deverá ser feito em grupos de no máximo 3 estudantes.
- II. O trabalho deverá ser realizado usando a linguagem de programação C#.
- III. Deverão usar os conceitos aprendidos na disciplina Algoritmos e Estruturas de Dados, levando em consideração as melhores estruturas para representar os itens do jogo (baralho, mesa, e as mãos dos jogadores).
- IV. O trabalho deverá ser entregue até a data 04/12/2023, via Canvas.
- V. Somente poderão ser utilizadas estruturas e conceitos vistos em sala de aula.
- VI. A avaliação do trabalho será por meio de apresentação do código, sendo a avaliação individual e em grupo. Terão questionamentos individualizados, e em grupo. Se ficar claro que algum(ns) integrantes do grupo não saibam explicar o código o grupo todo será penalizado.
- VII. Comece a fazer este trabalho logo, enquanto o problema está fresco na memória e o prazo para terminá-lo está tão longe quanto jamais poderá estar;

Problema – Rouba Montes:

Um dos jogos de cartas mais divertidos para crianças, pela simplicidade, é o Rouba- Monte. O jogo utiliza um ou mais baralhos normais e tem regras muito simples. Cartas são distinguidas apenas pelo valor (ás, dois, três, . . .), ou seja, os naipes das cartas não são considerados (por exemplo, ás de paus e ás de ouro têm o mesmo valor).

Inicialmente as cartas são embaralhadas e colocadas em monte na mesa de jogo, chamado de monte de compra, com face voltada para baixo. Durante o jogo, cada jogador mantém um monte de cartas, com face voltada para cima; em um dado momento o monte de um jogador pode conter zero ou mais cartas. No início do jogo, todos os montes dos jogadores têm zero cartas. Ao lado do monte de compras encontra-se uma área denominada de área de descarte, inicialmente vazia, e todas as cartas colocadas na área de descarte são colocadas lado a lado com a face para cima (ou seja, não ficam uma sobre a outra).

Os jogadores, dispostos em um círculo ao redor da mesa de jogo, jogam em sequência, em sentido horário. As jogadas prosseguem da seguinte forma:

- O jogador que tem a vez de jogar retira a carta de cima do monte de compras e a mostra aos outros jogadores; vamos chamar essa carta de carta da vez.
- Se a carta da vez for igual a alguma carta presente na área de descarte, o jogador retira essa carta da área de descarte colocando-a, juntamente com a carta da vez, no topo de seu monte, com as faces voltada para cima, e continua a jogada (ou seja, retira outra carta do monte de compras e repete o processo).
- Se a carta da vez for igual à carta de cima de um monte de um outro jogador, o jogador "rouba" esse monte, colocando-o em seu próprio monte, coloca a carta da vez no topo do seu monte, face para cima, e continua a jogada.
- Se a carta da vez for igual à carta no topo de seu próprio monte, o jogador coloca a carta da vez no topo de seu próprio monte, com a face para cima, e continua a jogada.
- Se a carta da vez for diferente das cartas da área de descarte e das cartas nos topos dos montes, o jogador a coloca na área de descarte, face para cima, e a jogada se encerra (ou seja, o próximo jogador efetua a sua jogada). Note que esse é o único caso em que o jogador não continua a jogada.

O jogo termina quando não há mais cartas no monte de compras. O jogador que tiver o maior monte (o monte contendo o maior número de cartas) ganha o jogo. Se houver empate, todos os jogadores com o monte contendo o maior número de cartas ganham o jogo.

Metodologia:

Deve ser incluída uma classe para representar uma Carta, esta deve ter dois atributos, **número** e **naipe**. Observe que o membro naipe não será utilizado, mas deve estar presente na classe. O baralho, a mesa e as mãos dos jogadores devem ser compostos por Cartas.

Da mesma forma, deve ser incluída uma classe para representar os jogadores, contendo os atributos: **nome**, **posição** (posição do jogador na última partida), **número de cartas em mãos** (última partida), uma fila contendo o ranking do jogador nas últimas 5 partidas.

Seu código deve permitir que se escolha a quantidade de jogadores que vão jogar, bem como a quantidade de cartas que poderão ser usadas no jogo. Em seguida, após a execução do jogo, o código deve apresentar o(s) ganhador(es) (nome do jogador, posição e número de cartas em mãos), além de apresentar

de forma ordenada as cartas presentes na mão do ganhador(es). Deve apresentar ainda, o ranking dos jogadores, ordenando de acordo com a quantidade de cartas em mãos.

Forneça ainda um método que permita pesquisar o histórico de posições de um jogador em específico. (apresentar os dados da fila que contém o histórico).

Finalmente, todas as ações da partida devem estar presentes em um histórico de logs, registrando cada uma das ações da partida. Ex: O baralho foi criado com X cartas. A partida atual será pelos jogadores (apresente o nome dos jogadores), na sequência, mostre qual o jogador iniciará a partida, posteriormente mostre a carta que o usuário retirou do monte de cartas, e assim sucessivamente, mostrando tudo o que acontece na partida.

Critérios de avaliação:

- Escolha das estruturas de dados para representação dos itens do jogo;
- Código legível, identado e bem comentado;
- Variáveis, classes, métodos e funções com nomes representativos;
- Uso adequado dos recursos de linguagem;
- Modularização e organização do código;
- Uso de arquivos para registrar os logs da partida;