Exercício de Programação: Jogo de Truco

Objetivo

Desenvolver um jogo de Truco simplificado na linha de comando utilizando os conceitos de orientação a objetos. O foco é praticar a modelagem de classes, atributos, métodos, e interações entre objetos, aplicando os princípios básicos da programação orientada a objetos.

Criar o Diagrama de Classes do exercício. Entretanto, o diagrama poderá ser feito depois, quem preferir pode focar primeiro no código e funcionamento.

Descrição do Jogo

Truco é um jogo de cartas tradicional. Nesta versão simplificada, o jogo será entre dois jogadores: um humano e o computador. O jogo utiliza um baralho espanhol simplificado de 40 cartas, divididas em 4 naipes (copas, espadas, ouros e paus) e cada naipe tem 10 cartas (1 (Ás) a 7, 10 a 12, considerando 10 como Valete, 11 como Dama e 12 como Rei).

Funcionalidades Básicas

- Distribuição de cartas: No início de cada rodada, cada jogador recebe três cartas.
- Jogada: Os jogadores, começando pelo humano, alternam-se jogando uma carta na mesa.
- Avaliação do vencedor da rodada: A carta mais alta ganha a rodada, a menos que seja chamado "Truco".
- Truco: O jogador pode chamar "Truco", aumentando os pontos da rodada. O oponente pode aceitar, recusar (dando os pontos da rodada ao desafiante) ou aumentar a aposta para "Seis", "Nove" ou "Doze".
- Vencedor do jogo: O primeiro jogador a alcançar 12 pontos vence o jogo.

Classes Propostas

Classe `Carta`

Representa uma carta do jogo.

- **Atributos**: `valor` (int), `naipe` (String).
- **Métodos**: Getters para `valor` e `naipe`.

Classe `Baralho`

Representa o baralho de cartas.

- **Atributos**: `cartas` (Lista de `Carta`).
- **Métodos**:
  - `embaralhar()`: Embaralha as cartas do baralho.
  - `distribuirCartas()`: Retorna uma lista com três cartas para serem distribuídas a um jogador.

Classe `Jogador`

Representa um jogador.

- **Atributos**: `nome` (String), `cartas` (Lista de `Carta`), `pontos` (int).
- **Métodos**:
  - `jogarCarta()`: Seleciona e retorna uma carta para ser jogada.
  - Getters e Setters para `pontos`.

Classe `Partida`

Controla o fluxo do jogo.

- **Atributos**: `jogadorHumano` (Jogador), `jogadorComputador` (Jogador), `baralho` (Baralho), `pontosRodada` (int).
- **Métodos**:
  - `iniciarPartida()`: Inicia o jogo, controlando o fluxo de rodadas.
  - `realizarRodada()`: Controla o fluxo de uma rodada, incluindo distribuição de cartas, jogadas dos jogadores, e avaliação do vencedor da rodada.
  - `avaliarVencedorRodada()`: Determina o vencedor da rodada baseado nas cartas jogadas.
  - `atualizarPontos()`: Atualiza os pontos dos jogadores após cada rodada.
  - `checarVencedorJogo()`: Verifica se algum dos jogadores alcançou os pontos necessários para vencer o jogo.

Requisitos Adicionais

- Implementar uma interface simples na linha de comando para interações com o jogador humano, incluindo mostrar cartas, permitir que o jogador selecione uma carta para jogar, e responder a chamadas de Truco.
- Tratar entradas inválidas de forma graciosa, pedindo ao jogador para tentar novamente.

Desafio Extra

- Adicione funcionalidade multiplayer, permitindo dois jogadores humanos jogarem um contra o outro localmente.
- Implemente estratégias básicas para o jogador computador, melhorando a experiência do jogo.

Este exercício é uma oportunidade para praticar a modelagem de um sistema orientado a objetos, interações entre objetos e lógica de programação. Foque em aplicar os conceitos de modelagem, encapsulamento e interação entre os objetos. Encorajamos você a começar com a funcionalidade básica e, conforme se sentir confortável, expandir o jogo com os desafios extras sugeridos.