BluffOrBuff Poker game in JAVA

Name: Ali Omran
Email: ali.omran.24@lehre.mosbach.dhbw.de
Git Repository: https://github.com/kretekarfolyam https://github.com/NewStudy2024/Programming1_kretek

---

1. Vision

I aim to create a **text-based console poker game** where a human player competes against **2 or 3 AI opponents**. The game will feature **three difficulty levels** (Beginner, Normal, Expert), adjusting both AI behavior and the level of assistance given to the player.

That said, I'm not entirely sure how complex this will become. Implementing AI that can bluff and adapt to the player's behavior sounds quite challenging. Will I be able to pull it off in time? I'll have to figure out a way to simplify things if needed.

---

2. Components

- **Card & Deck Classes**: Handles card creation, shuffling, and dealing.
- **Player Class: Storing hand data, chips, and making decisions should be manageable.**
- **Poker AI Class**: This is where things get tricky—making the AI behave differently based on difficulty might take more effort than I expect.
- **Game Manager**: Handling rounds, betting, and winner determination is essential, but will I be able to structure it properly without making it too complex?

3. Tech Stack

- **Java**: Core programming language.
- **Object-Oriented Programming (OOP)**: Used for game structure.
- **Algorithms & Logic**: AI decision-making, probability calculations.
- **Data Structures**: Arrays, Lists, HashMaps for managing cards and players.

4. Expected Result

- Ideally, by the end, I'll have a **working console-based poker game** that:
  - Lets a human player compete against AI opponents.
  - Adjusts AI strategy and player assistance based on difficulty.
  - Features basic AI bluffing and adaptation (if I can figure it out).
  - Has a simple but effective text-based interface.

  But… will everything work together as planned? I might have to cut down on certain features if they become too difficult or time-consuming.

5. Concepts from Class and Beyond

- **Core Java:** Loops, functions, conditionals, data types—I'm comfortable with these.
- **Object-Oriented Programming:** Structuring the game properly using OOP principles will be key.
- **Algorithms:** Probability calculations for AI decision-making—this part worries me a bit.
- **Game Theory**: AI opponent behavior (bluffing, adapting to player strategies)—seems advanced, but I'll try my best!