CSA Final Project: Texas hold 'em Implementation

Arnav Dagar, Nathan Qiu, Siddharth Nallagatla, Graham Winter

1. Core Classes:

- **Card:**
  - Represents a single playing card.
  - Attributes:
    - suit (e.g., Hearts, Diamonds, Clubs, Spades)
    - rank (e.g., 2, 3, 4, ..., 10, Jack, Queen, King, Ace)
  - Methods:
    - Constructor to initialize suit and rank.
    - getSuit(): Returns the suit of the card.
    - getRank(): Returns the rank of the card.
    - toString(): Returns a string representation of the card (e.g., "Ace of Spades").
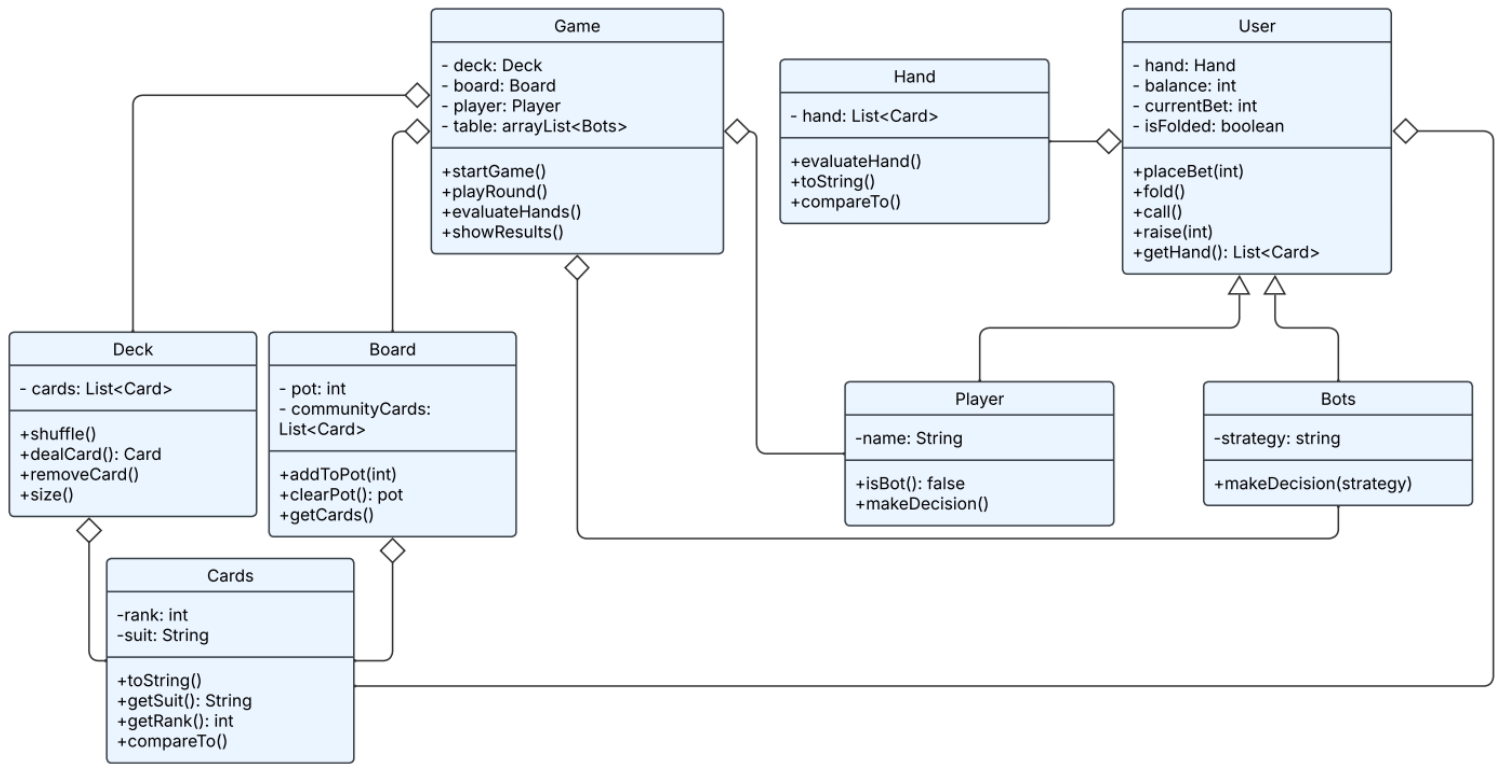   - compareTo(): Compares two cards, useful for sorting hands.
- **Deck:**
  - Represents a deck of 52 playing cards.
  - Attributes:
    - cards: A collection (e.g., ArrayList) of Card objects.
  - Methods:
    - Constructor to create a standard deck of 52 cards.
    - shuffle(): Shuffles the deck using Collections.shuffle().
    - dealCard(): Removes and returns the top card from the deck.
    - isEmpty(): Checks if the deck is empty.
    - size(): Returns the number of cards remaining in the deck.
- **Hand:**
  - Represents a player's hand of cards. For simplicity, let's start with a 5-card hand.
  - Attributes:
    - cards: A collection (e.g., ArrayList) of Card objects.
  - Methods:
    - Constructor to initialize an empty hand.
    - addCard(Card card): Adds a card to the hand.
    - getCards(): Returns the list of cards in the hand.
    - evaluateHand(): Determines the poker hand ranking (e.g., Straight Flush, Four of a Kind, Full House). This is the most complex method.
    - toString(): Returns a string representation of the hand.
    - compareTo(): Compares this hand to another hand.

- Player:
  - Represents a player in the game
  - Attributes:
    - name: The player's name.
    - hand: A Hand object representing the player's current hand.
  - Methods:
    - Constructor to set the player's name.
       getName(): Returns the player's name.
    - getHand(): Returns the player's hand.
    - setHand(Hand hand): Sets the player's hand.
- **Game:**
  - Manages the overall game flow.
  - Attributes:
    - deck: A Deck object.
    - players: A collection (e.g., ArrayList) of Player objects.
    - communityCards: A collection of Card objects (for games like Texas Hold'em).
  - Methods:
    - Constructor to initialize the game with a deck and players.
    - dealHands(): Deals an initial hand to each player.
    - dealCommunityCards(): Deals community cards (if applicable).
    - getPlayerHands(): Returns a list of all player hands.
    - determineWinner(): Determines the winner(s) based on hand rankings.
    - playRound(): Manages a single round of the game (dealing, betting, evaluating).
    - startGame(): Sets up the game and starts the rounds.

UML Diagram:

Game
- deck: Deck
- board: Board
- player: Player
- table: arrayList<Bots>

+startGame()
+playRound()
+evaluateHands()
+showResults()

Hand
- hand: List<Card>

+evaluateHand()
+toString()
+compareTo()

User
- hand: Hand
- balance: int
- currentBet: int
- isFolded: boolean

+placeBet(int)
+fold()
+call()
+raise(int)
+getHand(): List<Card>

Deck
- cards: List<Card>

+shuffle()
+dealCard(): Card
+removeCard()
+size()

Board
- pot: int
- communityCards: List<Card>

+addToPot(int)
+clearPot(): pot
+getCards()

Player
-name: String

+isBot(): false
+makeDecision()

Bots
-strategy: string

+makeDecision(strategy)

Cards
-rank: int
-suit: String

+toString()
+getSuit(): String
+getRank(): int
+compareTo()

Wireframe:

User Flow

Start Screen

Empty Poker Table

During a Hand

During a Hand (spectating)

Showdown

All screenshots taken from Poker Now

Play against bots or other players

Whale   Fish   Normal   Nit   Crusher

5 difficulties of bots