Decision Making-Bluff Bot

This document explains how the bot makes its decisions during the game.
In the game flow, the bot has to make the following **three main decisions**, and each decision has its own purpose, logic, and influencing variables.

The overall decision-making process is the same across all three modes (Easy, Medium, Hard), but the variable values and equations change depending on the difficulty level.

The bot's decisions are based on three key factors:
Personality (how aggressive or cautious it is)
Memory (what it remembers about opponents' behavior so far)
Game state (current cards, stage of game, etc.)

1. Whether to Raise (Challenge) the Placed Cards or Not

Description

This decision checks if the opponent is bluffing and whether the bot should challenge their claim.
It calculates a **challenge score (CS)** using the opponent's bluff history, the cards claimed, how many of those cards are still likely to exist, trust in the opponent, and the bot's own risk tolerance.
If the calculated **score is high enough**, the bot challenges the opponent's move. Otherwise, it lets the play continue.

Variables Used

- ClaimedValue: The value of the cards claimed by the opponent (e.g., "3" or "K").
- CardsPlaced: Cards currently played by the opponent.
- RemainingCards: How many cards of the claimed value are still left in the game.
- BluffRatio: How often the opponent has bluffed in the past.
- TrustScore: How much the bot trusts the opponent, based on their history.
- RiskTolerance: How much risk the bot is willing to take.
- GameStage: How far along the game is (0 = start, 1 = near end).
- Randomness: Adds unpredictability to avoid being too obvious.

How Variables Affect Decision

- Higher BluffRatio and lower TrustScore → more likely to challenge.
- If RemainingCards < 0 → forced challenge (since claim is impossible).
- More aggressive personality → more likely to challenge even in borderline situations.

Equations

Remaining Cards:

Calculates how many cards of the claimed value are left

```
R = 4 − cardsInDiscardPile − botCardsOfClaimedValue
```

Challenge Score (CS):

The main score used to decide whether to challenge:

```
CS =
```

```
+2 if R < 0

+1 if R = 0 and claimed ≥ 2

+(bluffRatio − 0.3) × 1.5

+(1 − trust) × 1.0

+0.2 × riskTolerance (if gameStage > 0.7)
```

Breakdown:

- `+2 if R < 0`: The claimed cards are impossible — highly suspicious.
- `+1 if R = 0 and claimed ≥ 2`: No cards left but still claiming ≥2 — suspicious.
- `+(bluffRatio − 0.3) × 1.5`: If opponent has history of bluffing → score goes up.
- `+(1 − trust) × 1.0`: If bot distrusts opponent → score goes up.
- `+0.2 × riskTolerance (if late game)`: If the bot is willing to take risks and game is near the end → score goes up slightly.

Final Decision:
If `CS ≥ 0.8`, the bot challenges. Otherwise, it passes.

2. Whether to Pass or Place Cards

Description

When it's the bot's turn, it decides whether to **pass** or **place cards on the pile**.
This decision takes into account the bot's personality, how many cards it holds, the declared value (bluff text), how much has already been played of that value, and how many consecutive passes have already occurred.
The idea is to avoid making suspicious or impossible moves while also avoiding too many passes.

Variables Used

- PersonalityType: Determines if bot is cautious or aggressive.
- DeclaredValue: The card value currently in play.
- CardsInHand: Cards the bot currently has.
- AlreadyPlayed: How many cards of the declared value are already on the table.
- BasePassProbability: Base chance of passing (depends on difficulty and personality).
- ConsecutivePasses: How many passes have happened in a row.
- Randomness: To make it less predictable.

How Variables Affect Decision

- Cautious personality or no matching cards → more likely to pass.
- If all cards of the declared value are already played → forced to pass.
- If many players have already passed → bot less likely to also pass (to keep play moving).

Equations

Declared Value Gone:
If the declared value is already exhausted and the bot has no matching cards:

```
alreadyPlayed ≥ 3 AND no cards in hand → pass
```

Adjusted Pass Probability:
Calculates likelihood to pass

```
AdjustedPassProbability =
  BasePassProbability
  + cautiousModifier
  − aggressiveModifier
  − consecutivePassesEffect
```

Final Decision:
If a random number between 0–1 is less than AdjustedPassProbability, bot passes.
Otherwise, it plays cards.

3. If Placing: Which Cards & Bluff Text to Declare

Description

If the bot decides to place cards, it must choose **which cards to play** and **what bluff text to claim.**
This decision is influenced by the bot's strategy level (beginner, intermediate, advanced), bluff probability (how often it bluffs), and what would look plausible to other players.
At higher difficulties, the bot bluffs in smarter and more believable ways.

Variables Used

- StrategyLevel: Determines how smart and realistic the bluff is.
- BluffProbability: How likely the bot is to bluff on this turn.
- BluffValueChoice: The declared value to bluff as.
- CardsByValue: Cards grouped by value in the bot's hand.
- RemainingCards: Cards of claimed value remaining in game.
- GameStage: Early or late stage of the game.
- Randomness: To make behavior unpredictable.

How Variables Affect Decision

- Higher BluffProbability → bot bluffs more.
- Advanced StrategyLevel → bot picks more believable bluffs.
- If bot has many cards of the declared value → more likely to play honestly.

Equations

Bluff Decision:

```
random() < bluffProbability → bluff
```

Bluff Value:
If bluffing, picks a bluff value depending on strategy:

- **Beginner** → chooses random value.
- **Intermediate** → chooses a value close to actual cards.
- **Advanced** → chooses a value that seems most plausible based on remaining cards.