

# Big River Poker Simulator

Erik S. Peterssen - [esp2114@columbia.edu](mailto:esp2114@columbia.edu)

## **Abstract**

Historically, poker has been viewed by many as a game for the undesirable and destitute. Its players have been forced into the crowded back-quarters of bars and gasoline-soaked garages. Some viewed the game as one of many get-rich-quick schemes provided by casinos. The mere name conjured up an image of slovenly men hunched over a table smoking cigars and drinking beer.

Recently, due to the popularity of Texas Hold'em, poker is on its way to wide-scale acceptance. Its players, now esteemed white collar professionals, are looking for a battle of wits. Evidence of this can be seen in television programming. Popular shows on ESPN, Bravo, and the Travel Channel chronicle the games of celebrities and professional poker players alike.

This interest has spilled over into the digital world. Gaming websites and instruction pages are proliferating at substantial rates. Gaming stocks are heavily traded in the securities markets. Even the government is taking an interest by reviewing the legality of these sites. Gaming entities are looking for fast development times and stable solutions to meet this increasing interest.

## **Solution**

Poker has turned into a hot commodity and fans are looking to 'Buy-In.' Development of poker applications is becoming competitive as different entities are struggling to absorb the influx of new fans who are quick to spend their money for self-improvement. Time-to-market is a critical component of development in this type of market. Reliable applications need to be written economically. Big River delivers a stable yet versatile solution to designing poker simulators. The game is designed around Texas Hold'em but is easily configured for most other poker games. Games such as five card draw, seven card stud are easily simulated by Big River.

## **Goals**

Big River design poker simulators containing configurable and intuitive players with different playing styles and emotional compositions in attempt to recreate real-life games. Players can be "designed" to bluff or to be careless with their money as such players are often the most difficult to play against. Different games can be created with a higher-level knowledge of programming. If the creators know the game of poker and can follow instructions, they game design a useful poker simulation application.

## Data Types and Attributes with Implementation Details

### Player

- Name
- Style
  - Honest-Bluffing
  - Passive-Aggressive
  - Tight-Loose
    - Defaults?
- Active: In the game or sitting out
- Money
  - Active
  - Reserve
    - Persistence?

### Table

- Number of seats
  - Vector of chairs
    - Pointers to Player(s)
- Community Cards
  - Vector of pointers to Deck Card(s)
- Pot
  - It is possible to have several pots. This might be a vector or a class.

### Hand

- Aces high, low, both.
- # hole cards
  - Vector or pointers to Deck Card(s)

Muck: The graveyard. Card(s) retire here.

- Vector of pointers to Deck Card(s)
- Asterisk to denote all (\*).

Deck: Contains Standard 52 deck Card(s) with the following functionality:

- Draw() returns a pointer to a card to be consumed by a Hand, Table, or Muck.
  - Random card generator % 52
- Shuffle()
  - Mostly likely some random selection routine.
- Asterisk to denote all (\*).

Card: Created on combinations of the following:

- Suit (enum)
  - Spades
  - Clubs

- Hearts
  - Diamonds
- Number (enum)
  - Ace (A)
  - King (K)
  - Queen (Q)
  - Jack (J)
  - Ten (T)
  - Nine (9)
  - Eight (8)
  - Seven (7)
  - Six (6)
  - Five (5)
  - Four (4)
  - Three (3)
  - Two (2)
- Color (enum)
  - Red (R)
  - Black (B)
- Visible (enum)
  - Open (O)
  - Closed (C)

#### Rules

- Split Pot allowed
- Best hand out of 5, 7, ...
- Standard Poker
  - Straight Flush
  - Full House
  - Flush
  - Straight
  - Three of a Kind
  - Two Pair
  - Pair
  - High Card
- Best Hand, Worst Hand, Both

#### Language Restrictions

- All Card(s) in the Deck must be in a Hand or on the Table, or Muck(ed).
- A Deck cannot be shuffle(d) while there are outstanding references to its Card(s).
- A Table has Players.
- A player has a hand.
- A hand consists of (pointers to) Card(s)\*.
- A Table has Card(s)\*.
- A Muck has Card(s)\*.

The above three\* are mutually exclusive at the individual Card level.

## **Sample Code**

```
table { // Establish the table and table components

    // table attributes
    seats<4>
    ccards<2>
    pot<-1>    // specifies no limit

    // game attributes/sequence
    game {
        deal+1    // dealer increments by one each deal
        bet
        hole<2r> // deal 2 hole cards to each player (round-robin)

        bet
        burn<1>  // always burn one card before flipping cards
        community<3>

        bet
        burn<1>
        hole<1>

        bet
        burn<1>
        hole<1>
        bet
    }

    rules {
        split<0> // no high/low
        hand<5>  // best 5 cards
        aces<-+> // aces both low and high
    }

    // players at table
    player {
        name<Sally>

        style.honest<5>
        style.passive<5>
        style.tight<5>

        money.active<50>
        money.reserve<50>
    }

    // players at table
    player {
        name<Eduardo>

        style.honest<9>
        style.passive<8>
        style.tight<8>

        money.active<50>
        money.reserve<50>
    }
}
```

```
}  
  
// players at table  
player {  
  name<John>  
  
  style.honest<3> // bluffer  
  style.passive<7> // plays a decent amount of hands  
  style.tight<9> // bets heavily  
  
  money.active<50>  
  money.reserve<50>  
}  
  
player {} // This is the user  
}
```