Software Specification

Family Friendly Poker

Version 1.0 - June 2022

Authors: Daniel McClure, Angelina Negrete, Christiano Gravina, Martin Qinyun Tang, and Zhenyu Tang

DMCA, Inc.

Table of Contents

Glossary ……pg 2

1. Player Software Architecture Overview……..……...pg 4

1.1 Main data types and structures ……..…pg 4

1.2 Major software components ……....pg 4

1.3 Module interfaces ……..…pg 5

1.4 Overall program control flow…….……..pg 5

2. Poker Server Software Architecture Overview ……pg 6

2.1 Main data types and structures ……pg 6

2.2 Major software components ……..……..pg 6

2.3 Module interfaces ……..…..pg 6

2.4 Overall program control flow ……..……pg 7

3. Installation ……pg 7

3.1 System Requirements …….……pg 7

3.2 Setup and Configuration …….…..…pg 7

3.3 Building, compilation, installation …….……pg 7

3.4 Uninstalling …….pg 8

4. Documentation of packages, modules, interfaces……...…….…….pg 8

4.1 Data Structures…….…..……pg 8

4.2 Functions and Parameters ……..……pg 10

4.3 Communication Protocol …… ……..pg 11

5. Development plan and timeline…….……pg 11

5.1 Partitioning of tasks ……..…….pg 11

5.2 Team member responsibilities ……/….…….pg 12

Glossary

Suit: One of the categories the cards of a deck are divided into. The cards typically display symbols that denote which suit they belong to. There are four suits, which are Clubs, Diamonds, Hearts, and Spades.

Hand: The name for the set of cards each player plays with in Poker. Each hand consists of five cards.

Rank: Another category by which the cards of a deck are divided into. There are thirteen ranks, with 2 as the lowest and A as the highest.

Call: Calling is the act of matching a bet or raise during a betting round to stay in the game.

Check: An action during a betting round where the player chooses to not bet or raise when given the opportunity.

Raise: An action during a betting round where the player increases the stakes by wagering more points than are currently needed to match.

Fold: An action during a betting round where the player chooses to forfeit their chances at winning the game in order to not have to further bet.

Board: The community cards dealt by the dealer that the players must play off of.

Flop: The first three cards that the dealer reveals after the initial round of betting.

Turn: The fourth community card revealed after the flop.

River: The fifth community card revealed after the river.

Royal Flush: The highest-ranking combination of straight flush is given a special name, the Royal Flush. It consists of A, K, Q, J, and 10 all in the same suit.

Straight Flush: This hand consists of five cards of the same suit in sequence, excluding the highest-ranking combination, which is given a different name. Example: 5♣, 6♣, 7♣, 8♣, 9♣

Four of a Kind: This hand contains four cards of the same rank and a fifth card of a different rank. Example: 3, 3, 3, 3, 10

Full House: This hand consists of three cards of one rank and two cards of another rank. Example: 2, 2, 2, J, J

Flush: Five cards all of the same suit, but not in sequence. Example: 3♠, 7♠, 10♠, Q♠, A♠

Straight: A hand where the five cards are in sequence, but they don't all have the same suit. Example: 4♦, 5♣, 6♠, 7♥, 8♣

Three of a Kind: A combination where three cards are of the same rank, and the last two cards are each a different rank from any other. Example: 5, 5, 5, Q, 9

Two Pair: This hand contains a pair of one rank and a second pair of a different rank, with the fifth card being a different rank from the two pairs. Example: A, A, 2, 2, 8

Pair: A combination that consists of two cards with the same rank with the other three cards being different and unique ranks. Example: 1, 1, K, 4, 7

No Pair: A hand that has no pairs or otherwise matching cards, do not have the same suit, and are not in sequential order.

High Card: A hand where the only valuable thing is the card with the highest rank.

Small Blind: This is the amount (as determined by the group) the person to the immediate left of the dealer has to put in the pot before any cards have been revealed.

Big Blind: This is the amount the person two to the left of the dealer has to put in the pot before any cards have been revealed; it is two times the small blind.

1 Software Architecture Overview

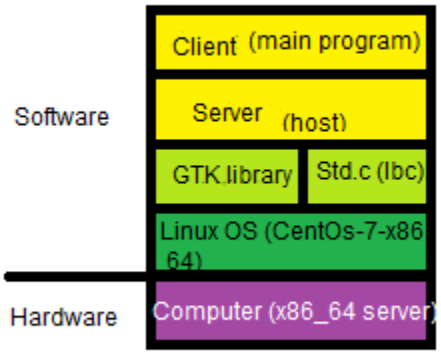
1.1 Main data types and structures

PLAYER *createPlayer: This holds all player data such as name, bet amount, and if they have made an “action” (call, fold, raise) so that it can be compared with data coming from the server.

CARD *CreateCard: This creates the cards and contains the suit and ranks

BOARD *CreateBoard: This creates the board or table where the bet is held

1.2 Major software components



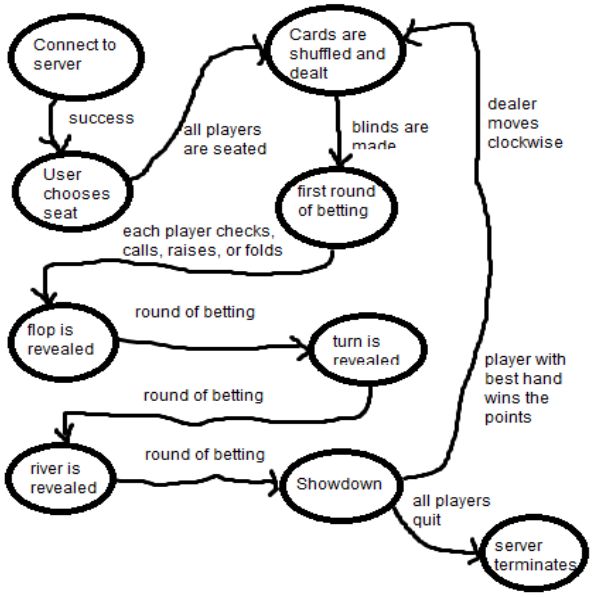
1.3 Module interfaces

Poker_Client.c

Provides: main program with GUI

Requires: Deck.h, Constants.h

1.4 Overall program control flow



2 Poker Server Software Architecture Overview

2.1 Main data types and structures

The server will include:

PLAYER struct

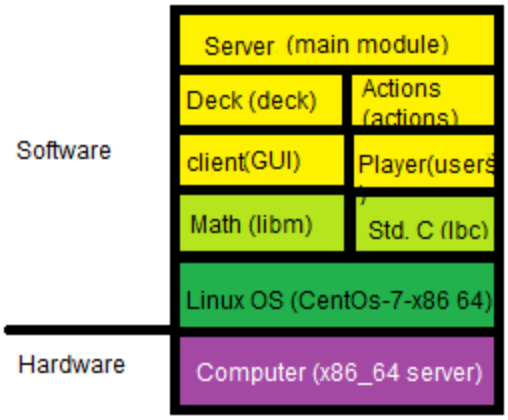
CARD struct

An array that serves as the deck and holds 52 CARD structs

SEAT *CreateSeat: This creates the “seat” where a player will play at

TABLE *CreateTable: This creates and returns a double linked list that contains all the players that will be participating in the game

2.2 Major software components



2.3 Module interfaces

Actions.c

Provides: All available actions

Requires: Actions.h and Deck.h

Deck.c

Provides: creation of the card deck

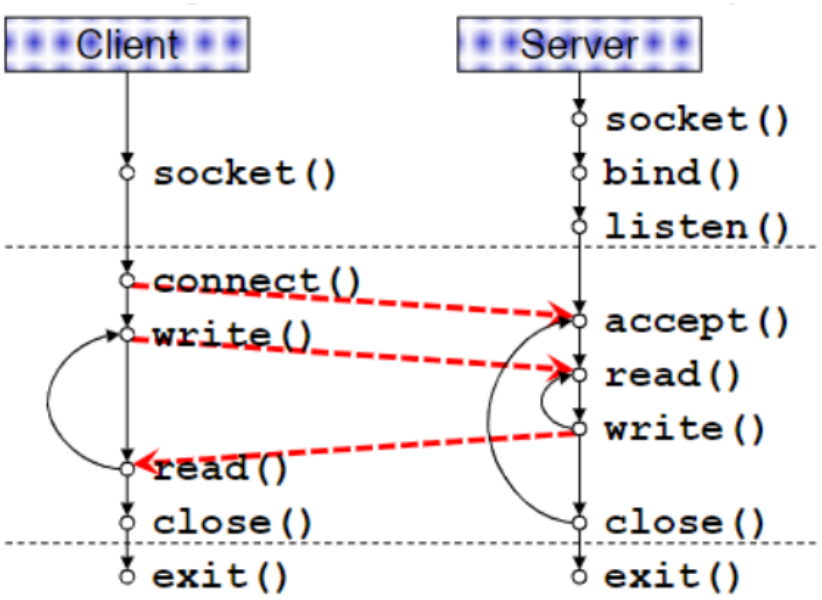
Requires: Deck.h and Constants.h

Poker_Server.c

Provides: information to client

Requirese: Deck.h, Constants.h, Actions.h, Players.h

2.4 Overall program control flow



3 Installation

3.1 System requirements, compatibility

Computer with PC hardware

Linux operating system

3.2 Setup and configuration

The program is already configured and just needs to be installed by the user. See below for installation instructions.

3.3 Building, compilation, installation

1. Program comes in a tar.gz package named Poker_V1.0.tar.gz
2. Install the package and extract it
3. Unpack the archive and type make to create the game
4. Then type cd bin and ./Server 10090 to start the host server
5. ./Client to connect to host (server name: bondi.eecs.uci.edu ; port number: 10090)

3.4 Uninstalling

To uninstall the program, type in the terminal : make clean

4 Documentation of packages, modules, interfaces

4.1 Data structures

This is the data structure which represents the cards in the deck. Its suit is an int which corresponds to the specific card suit the card belongs to (i.e. 0 would be Clubs, 1 would be Diamonds, 2 would be Hearts, etc.). Its rank is also an int and corresponds to the rank of the card(i.e. 0 would be 2, 1 be 3, 12 would be A).

```
typedef struct Card{
        int suit;
        int rank;
} CARD;
```

This is the data structure for the individual player.

```
typedef struct player {

   PLAYER *player = malloc(sizeof(PLAYER));
   player->prev = NULL;
   player->next = NULL;
   player -> name = name; /* setting to "test" for testing */
   player->address = address;
   player->length = length;
   player->socketFD = socketFD;
   player -> bet = 0;
   player->fold = 0;
   player->raiseConst = 0;
   player->callConst = 0;
   player->dealer = 0;
} PLAYER;
```

The following data structure is for the communal deck and bet for each round:

```
BOARD *CreateBoard(void){
   BOARD *board = malloc(sizeof(BOARD));
```

```
    board->bet = 0;
    return board;
}
```

Holds all the seats available in the game.

```
TABLE *CreateTable(){
    TABLE *table = malloc(sizeof(TABLE));
    table->length = 0;
    table->first = NULL;
    table->last = NULL;

    return table;
}
```

This returns a seat to a player that is ready to play. It also updates the table “length” or game size

```
SEAT *CreateSeat(PLAYER *player, TABLE *table)
{
    SEAT *seat = malloc(sizeof(SEAT));
    seat->table = table;
    seat->player = player;
    if(table->length > 0)
    {
        seat->prev = table->last;
        table->last->next = seat;
        seat->next = table->first;
        table->length++;
    }
    else if(table->length == 0)
    {
        table->first = seat;
        table->last = seat;
        table->length++;
    }
    return seat;
}
```

4.2 Functions and parameters

void PrintMenu(void)

- this function prints the menu

PLAYER *CreatePlayer (char name[16], char ID[16])

- This function creates the structure of a player. It takes the name and ID character strings input by the user as parameters and stores the parameters into the player structure.

void DeletePlayer (PLAYER *player)

- This function is used to delete the player structure. It sets the pointers within the player structure to NULL and frees up the memory space.

CARD *CreateCard (int suit, int rank)

- This function creates a card struct from a given int value representing the suit and an int value representing the rank

void DeleteCard (CARD *card)

- This function frees a card struct from memory.

void playerMove (PLAYER *player)

- This function allows the player to perform actions such as call, check, fold or raise. It will change the values stored in the members of the PLAYER structure.

DECK *CreateDeck(void)

- This function creates the deck.

Void DeleteDeck(DECK *deck)

- This function deletes the deck and frees up the memory space occupied by it

int HandCompare(PLAYER *first, PLAYER *second, PLAYER *third, PLAYER *fourth, DECK *deck)

- This function compares the hand of the four players. It returns an integer that signals which player wins the round.
- This function creates the player's hand by adding the deck of each round to the “CARD *hand[7]” array.

Void SetDeck(DECK *deck)

- This function sets 5 random cards onto the deck as well as initializes the bet to zero

Void ServerRequest (int ServerSocket)

- This function processes the client’s requests. It allows the server to take different actions depending on the client’s request. It takes a parameter that represents the server’s socket

Int CreateSocket(void)

- This function creates the server socket which is used in the server function.

Void ServerMainLoop(int ServerSocket, int Terminate)

- This is the main function for the server. The server should not terminate if the parameter “Terminate” is not zero.

4.3 Communication Protocol

Communication will include a keyword before each message to ensure that no cross communication between other servers and clients occurs. Information will be differentiated with dashes and colons.

HAND- player's own cards

FLOP-12:0-0:1-:3:2 <--the flop

RIVER-12:0-0:1-3:2-8:0 <--the flop + the fourth card

TURNER-12:0-0:1-3:2-8:0-9:0 <--the river + the fifth card

CONNECTION <--somebody has connected

GETBET-2000 <--to get the bet from client, minimum bet is 2000

SEATSAVAILABLE-0-1-2-3 <--to send which seats are available

5 Development plan and timeline

5.1 Partitioning of tasks

1. Create a card structure
    a. Each card has a suit and rank
2. Create a deck structure to hold the cards
    a. Shuffle the deck after each game
    b. Distribute two cards to each player
    c. Reveal flop, turn, and river in correct order
3. Create a player structure
    a. Holds player name and seat
    b. Dealer is initially random, then rotates clockwise
    c. Small blind and big blind
4. Create a hand structure
    a. Each player has their own hand
    b. Assign values to each hand depending on their strength
    c. Compare hands at the end of the game, strongest hand wins
5. Create poker rules for determining winning hands
6. Communicate between client and host server
7. Implement a GUI

5.2 Team member responsibilities

Poker main program - Martin

User interface (GUI) - Daniel

Player main program - Angelina

Poker rules - Zhenyu

Connections (data structures for sockets) - Christiano

Documentation - GROUP

Testing - GROUP

References

EECS 22L: Project 2: Online Poker; Prepared by Vivek G. and Yutong Wang; Prof. Rainer Dömer; 3 May 2022.

EECS 22L: Poker Software Specification Grading Criteria; Prepared by Vivek G. and Yutong Wang; Prof. Rainer Dömer; 9 May 2022.

GTK Documentation. https://www.gtk.org/.

Index

Array pg. 6

Client pg. 6, 9, 10

Communication pg. 9, 10

Data Structure pg. 4, 6, 8

GUI pg. 4, 5, 10, 11

Hand pg. 2, 3, 8, 9, 10

Install pg. 7

Player pg. 2, 4, 5, 6, 8, 9, 10, 11

Poker pg. 4, 5, 6, 7, 11

Server pg. 4, 5, 6, 7, 9, 10

Uninstall pg. 8