Online Poker

Software Specification

Authors:
Anna Ahn
Ishika Narain
Filiberto Alvarez
Bilal Malik
Rishi Tirumala
Tuaha Khan

Producer: Full House of Fools
Affiliation: FHF Software Inc.
Version: 0.0.3 (Alpha)

Table of contents:

Table of contents:.......... 2
Glossary:.......... 3
1 Poker Client Software Architecture Overview.......... 6
1.1 Main data types and structures.......... 6
1.2 Major software components.......... 6
1.3 Module interfaces.......... 7
1.4 Overall program control flow.......... 10
1.5 Automated Client: Poker Bot.......... 11
2 Poker Server Software Architecture Overview.......... 12
2.1 Main data types and structures.......... 12
2.2 Major Software components.......... 12
2.3 Module interfaces.......... 13
2.4 Overall program control flow.......... 14
3 Installation.......... 15
3.1 System requirements.......... 15
3.2 Unpacking and configuration.......... 15
3.3 Building, Compilation, installation.......... 15
4 Documentation of packages, modules, interfaces.......... 16
4.1 Detailed description of data structures.......... 16
4.2 Detailed description of functions and parameters.......... 19
4.3 Detailed description of the communication protocol.......... 20
5 Development plan and timeline.......... 21
5.1 Partitioning of tasks.......... 21
5.2 Team Member Responsibilities.......... 21
References.......... 22
Index.......... 22
Copyright.......... 23

Glossary:

Poker Terms

1. *All-in* - A type of bet that the player can make, where they put all of their available money into the pot. If the previous player has made an all-in bet which is greater in value than the current player's, then a call will put all of the current player's money into the pot. However, if the current player makes an all-in bet and the next player also wants to go all-in, they can only match the total value of the previous player's all-in.
2. *Bet* - The amount of money placed by a player on their cards. A bet contributes to the pot.
3. *Blind (small and big)* - blinds are mandatory bets that are placed into the pot before any cards are dealt. The blinds are paid each hand by the players who are occupying the "small blind" and "big blind" seats at the table. Poker blinds help drive the action forward and prevent players from simply folding until they are dealt premium cards.
4. *Call* - One of the options given to the player when betting on their cards. This move is only valid when the player before has placed a bet. If a player wants to check, they match the raise made by the previous player. Check is no longer an option.
5. *Card hands (in order of strength)*
   a. *Royal flush* - A hand that consists of a straight with Ace, King, Queen, Jack, and a 10, all in the same suit.
   b. *Straight flush* - Any straight with all five cards of the same suit.
   c. *Four of a kind* - Any four cards of the same rank.
   d. *Full house* - Three cards of the same rank along with two cards of the same rank that are different from the first three.
   e. *Flush* - Any five cards of the same suit.
   f. *Straight* - Any five cards in a sequence.
   g. *Three of a kind* - Any three cards of the same rank.
   h. *Two pairs* - Two cards of the same rank along with another two cards of the same rank that are different from the first two.
   i. *(One) Pair* - Two cares of the same rank
   j. *High Card* - The highest card in the hand.
6. *Check* - One of the options given to the player when betting on their cards. This move skips the player's bet to the next player.
7. *Client* - The computer host receiving information from the server
8. *Deck* - The stack of cards being used in the game.

9. *Flop* - The first three cards shown in Texas Hold'em. There is one card discarded before the flop.
10. *Fold* - One of the options given to the player when betting on their cards. This move is always available to the player, and allows the player to back out of the game. The bets made by the player during the game are still lost to the player, and they sit out for the rest of the game.
11. *Formal card hand names* (Other names card hands are known by)
    a. *Quads* - Another name for a four of a kind.
    b. *Boat* - Another name for a full house.
    c. *Pocket queens* - Another name for a pair of Queens.
    d. *Pocket rockets* - Another name for a pair of Aces.
12. *Hand* - The cards given to the player that only they can see.
13. *House* - Also known as the Dealer, they are the ones who deal all of the cards.
14. *Player(s)* - The person(s) playing the game.
15. *Pot* - The cash prize that is made up of bets contributed by all of the players in the game and is given to the player who wins the game.
16. *Raise* - One of the options given to the player when betting on their cards. This option is always available to the player, and allows them to place a bet. The bet placed has to be more than the current bet amount. Only call allows the player to match a bet made by a previous player.
17. *Pre-Flop* - The round of betting before the first 3 cards are revealed. As the name suggests, it is before the flop.
18. *Position* - The order in which a player acts for their turn.
    a. *Small Blind* - First to act
    b. *Big Blind* - Second to act
    c. *Under The Gun* - Third to act
    d. *Under The Gun + 1* - Fourth to act
    e. *Middle Position / Lojack* - Fifth to act
    f. *Middle Position + 1* - Sixth to act
    g. *Hijack* - Seventh to act
    h. *Cutoff* - Eighth to act
    i. *Button* - Ninth to act
19. *River* - The last card shown in Texas Hold'em, after a card is discarded.
20. *Rank* - A category used in a deck of cards. There are 13 ranks (in order of ascending value): Ace, 2, 3, 4, 5, 6, 7, 8, 9, 10, Jack, Queen, and King. The Ace can be placed either under a 2 or above a King.

21. *Server -* Remote computer that provides information
22. *Suit -* A category used in a deck of cards. There are four suits: Club, Spade, Diamond, and Heart.
23. *Straddle bet -* Usually the player Under the Gun, optionally, will place a bet before any cards are dealt. This bet is higher than the big and small blind, and makes them the last to act on the pre-flop if no one raises.
24. *Texas Hold'em -* A type of poker game. At the start of the game, each player is dealt two cards. The small blind places their bet first, … after all of the bets have been placed and no one else raises the bet, a card is discarded from the deck and the Flop is dealt. Betting is started with the small blind again, and is finished when no one else raises the bet. Then, a card is discarded from the deck and the Turn is dealt. Again, betting is started with the small blind again, and is finished when no one else raises the bet. And finally, a card is discarded from the deck and the River is dealt. Betting is started with the small blind again, and is finished when no one else raises the bet. Once all of the bets have been placed, the players reveal their cards in turn, with the player with the best hand winning the pot.
25. *Turn -* The second card shown in Texas Hold'em, after a card is discarded.
26. *Int -* a type that represents and integer
27. *Struct*/class- a type of custom data with a defined structure
28. *Enumerator -* a way to number a set of items with a list
29. *Pointer -* A type that links itself to a location and memory and calls on it whenever the pointer is called
30. *Doubly linked list -* a set of struct linked to each other in memory through pointers in their struct
31. *Void -* a key word that denotes a function return no value
32. *NULL -* a keyword that denotes that a type has no value
33. *Git -* A repository containing the code and files used
34. *Push -* Push code and files to the git repository
35. *Pull -* Pull code and files from the git repository
36. *Main -* The main function where the code is executed from
37. *GUI -* The graphical user interface for the program
38. *Widget -* Components that make up the GUI application

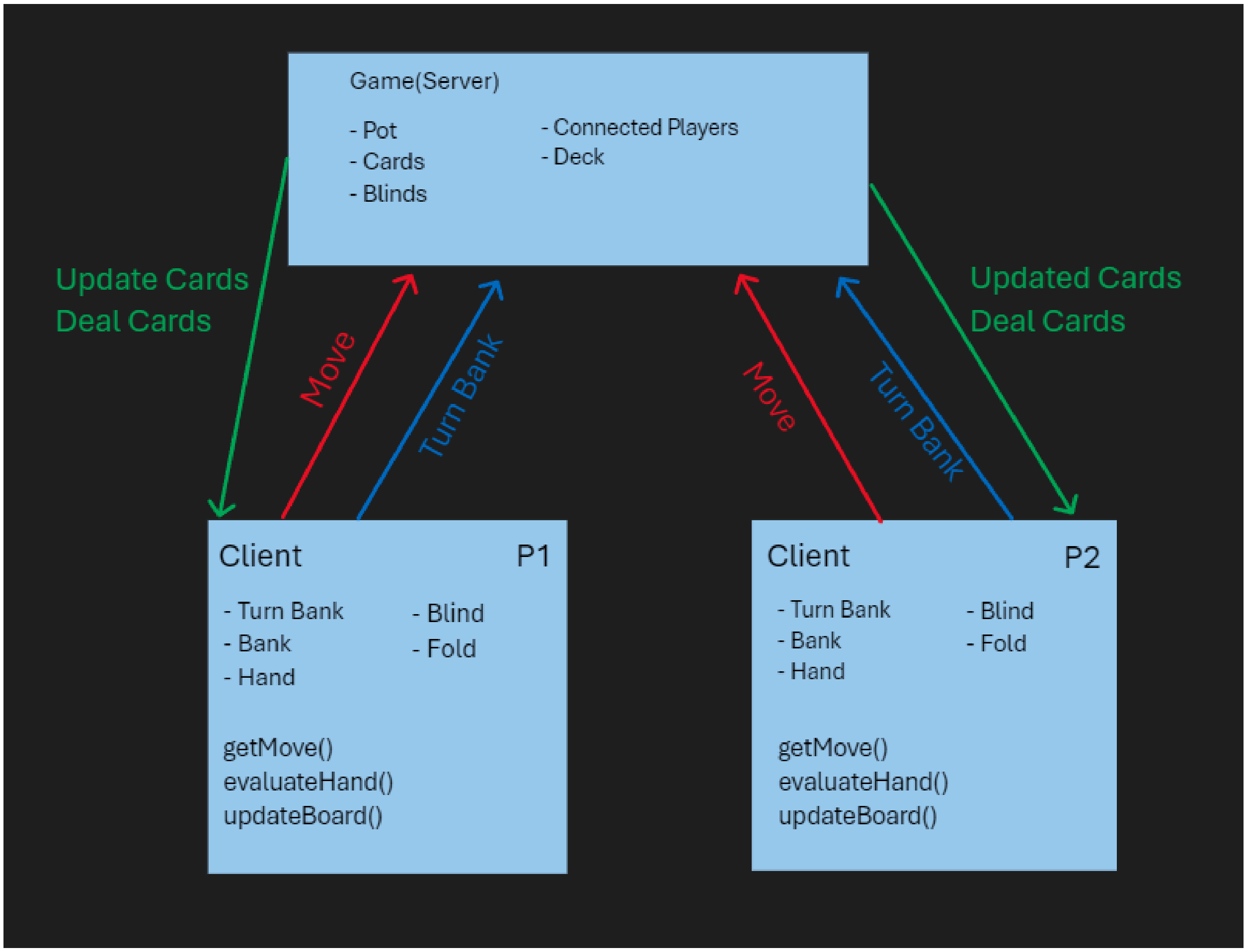
1 Poker Client Software Architecture Overview

1.1 Main data types and structures

- Doubly linked list: Structure that saves/keeps the current/past moves made by the user/bot.
- Class: defined data types
  - Game
    - String[52] deck
    - String[5] cardsOut
    - Player[MAX_PLAYER] connectedPlayers
    - int Bank
    - int smallBlind, bigBlind
  - Player
    - String[2] hand
    - bool fold
    - Int bank
    - bool smallBlind
    - bool bigBlind
    - int turnBank
- Int: used to represent turns/player number/pot/etc.
- Pointer: used to point to other players in the game.

1.2 Major software components

- Diagram of Module Hierarchy



- Software Components
  - Poker
    - The main file which controls the moves and screens of the poker program
  - GUI
    - Custom file build on top of graphics library to display poker for the client

1.3 Module interfaces

Major Functions:

Connection:

- int initializeConnection

  Description: Initialize connection to server

  Parameters: server address, port

  Return: Integer (for determining connection status)

- void closeConnection

  Description: Close connection to server

  Parameters: None

  Return: None

Graphical User Interface:

- void initializeGUI

  Description: initialize GUI

  Parameters: none

  Return: None
- void displayHand

  Description: Shows individual player their current hand

  Parameters: 2 cards

  Return: None
- void displayTableCards

  Description: Show cards on the table

  Parameters: 5 cards

  Return: None
- int displayPlayerActions

  Description: Shows action (call, raise, etc.)

  Parameters: None

  Return: None
- Void closeGUI

  Description: Closes GUI

  Parameters: None

  Return: None

Game:

- void initializeGameState

  Description: Initilize game state (player size, chips, etc.)

  Parameters: None

  void updateGameState

  Description: Update game state (player size, chips, etc.)

  Parameter: Data from server
- void getPlayerHand: Get individual players hand
- void getTableCards: Gets the cards on the table, takes in the 5 cards as parameters

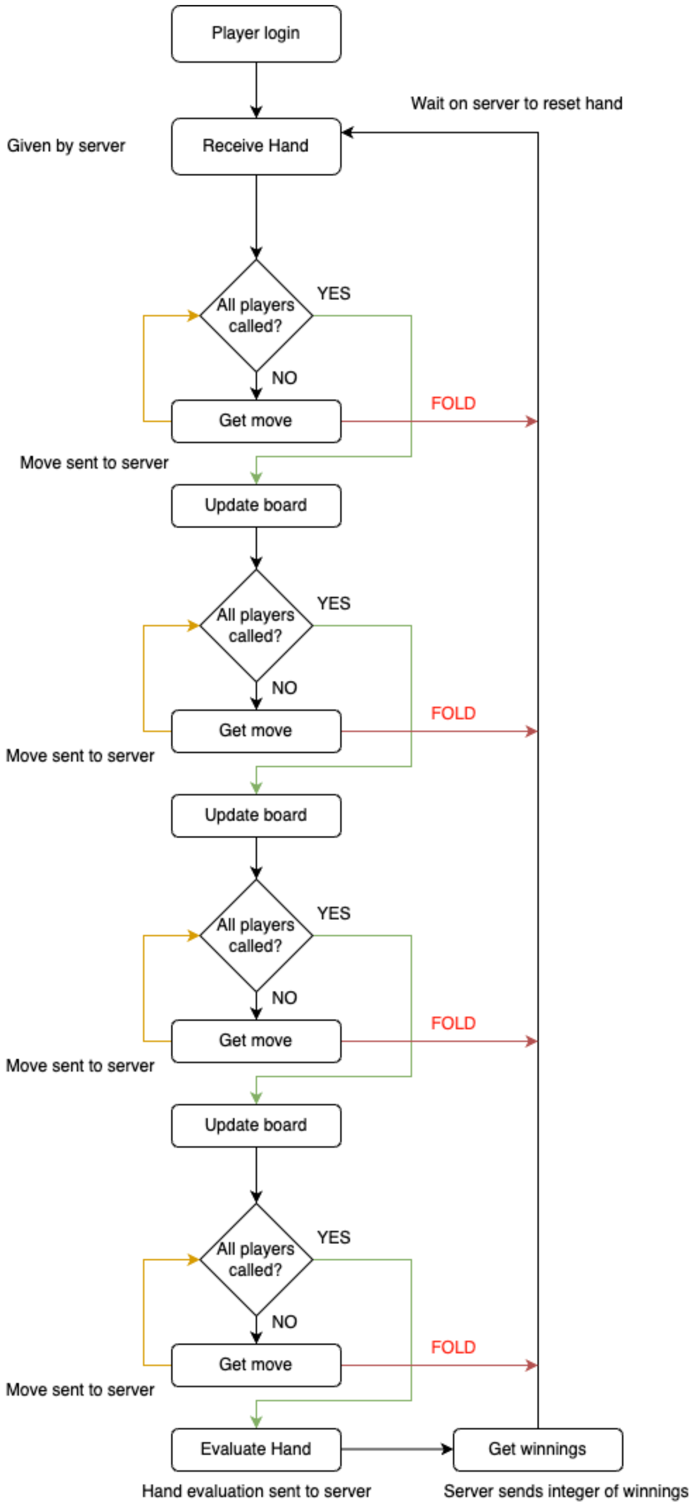
Player:

- void initializePlayer

Description: initializes a player
Parameter: Player pointer

- int getPlayerChips:
  Description: Get the player's chips
  Parameter: Player pointer
  Return: Players chips
- void updatePlayerChips:
  Description: Update player chips
  Parameter: Player pointer
- void performPlayerAction
  Description: Perform action chosen by player
  Parameter: Player pointer, game pointer

1.4 Overall program control flow



1.5 Automated Client: Poker Bot

- Players will have the ability to add bot players to the game.
- The bot will act like a player, with its own hand, bank, turn bank, and the same options for moves (Raise, Fold, Call, etc.).
- Bot will calculate poker equity, or probability of winning with the current hand it is dealt
    - If probability is too low, the bot will fold and the turn will continue to the next person.
    - If probability is high, the bot will call or raise.
    - If probability is higher than some percentage, i.e 60%, the bot will go all in.

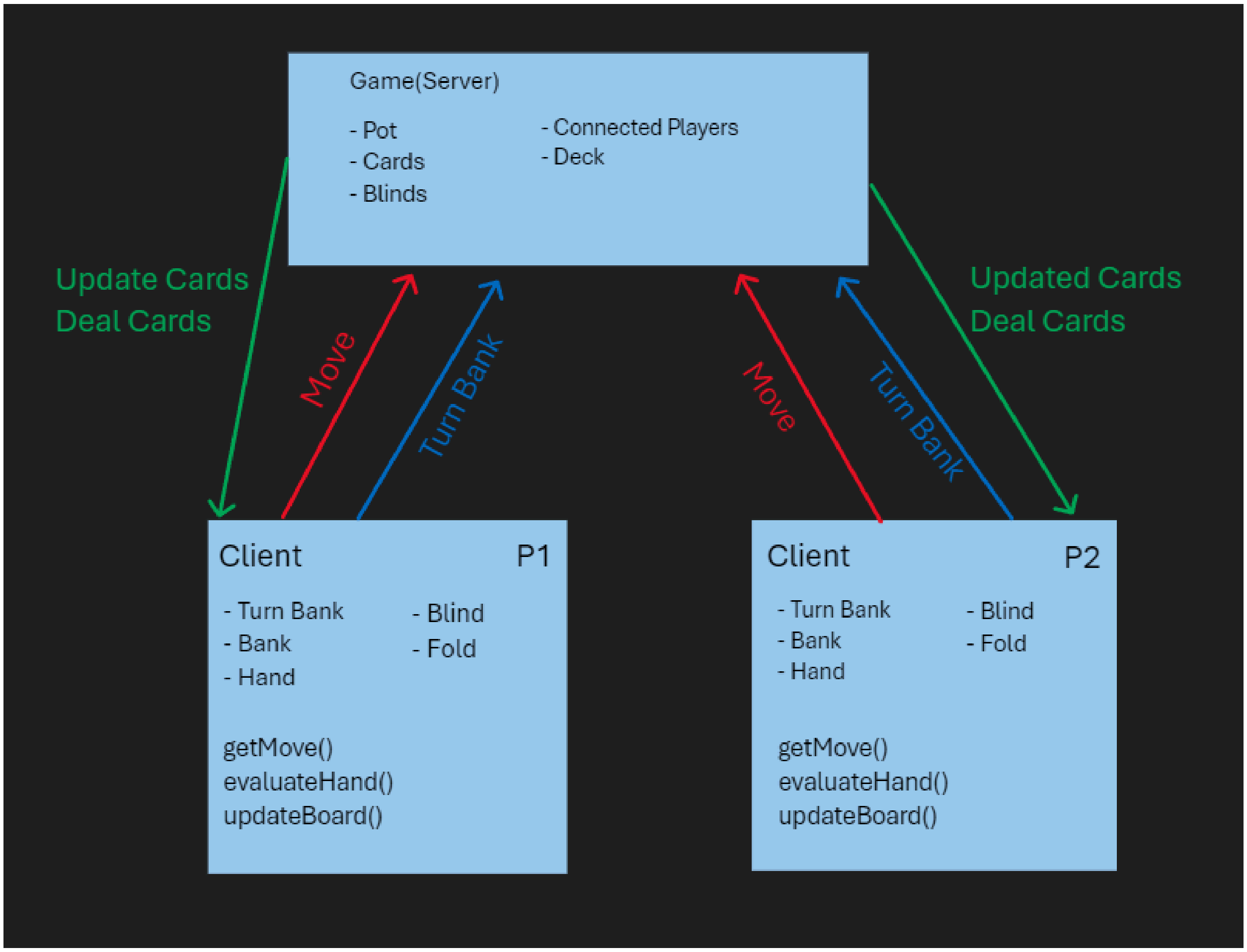
2 Poker Server Software Architecture Overview

2.1 Main data types and structures

- Doubly linked list: Structure that saves/keeps the current/past moves made by the user/bot.
- Class: defined data types
    - Game
    - Player
    - Bot
- Int: used to represent turns/player number/pot/etc.
- Pointer: used to point to other players in the game.

2.2 Major Software components

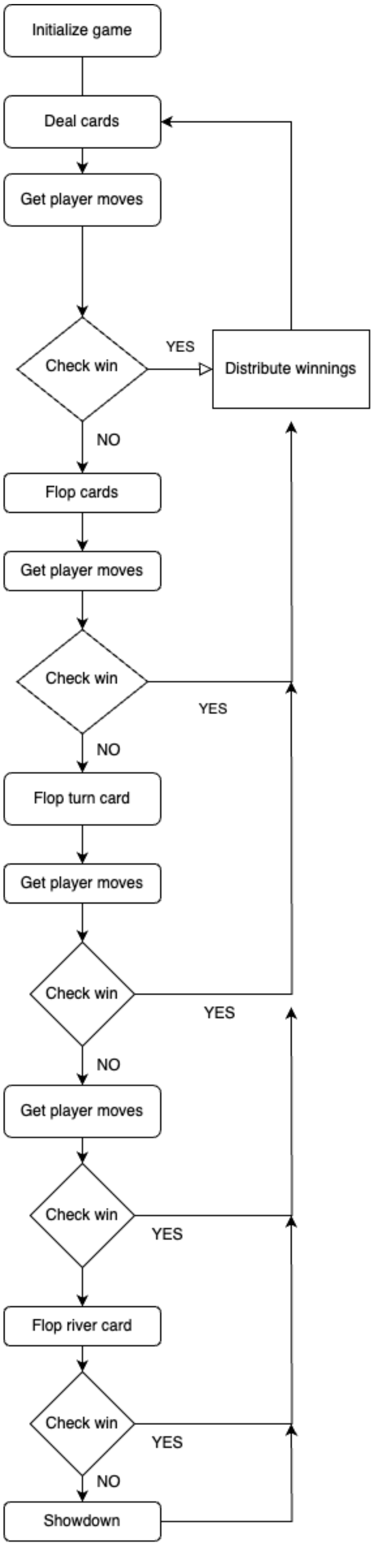
- Bot
    - Provides: Output for Raise, Call, or fold
    - Requires: Player Hand and Table Hand
    - Exported Functions: None
- Poker
    - The main file which controls the moves and screens of the poker program
- GUI
    - Custom file build on top of graphics library to display poker for the server
- Diagram of module hierarchy



2.3 Module interfaces

- Bot
  - Description: Custom file containing the code for the poker bot
  - Parameters: 5 Cards , 2 Card Hand
  - Return: Raise, Call Fold
  -
- GUI
  - Description: File containing the code for the graphics
  - Parameters: Pot size, Card in hand
  - Return: N/A
- Server
  - Description: Module containing Control Flow code for Server
  - Parameters: N/A
  - Return: N/A

2.4 Overall program control flow



3 Installation

3.1 System requirements

- Windows/Mac with built in terminal
- Able to use/SSH into a linux server
- Download Xming(windows) to be able to display the poker GUI in separate window

3.2 Unpacking and configuration

- mkdir poker
- cd poker
- Download files poker directory in linux server
- tar -xvzf Poker_V1.0_src.tar.gz
- make all
- cd bin/
- ./Poker
- To clone repo use the command “git clone team2bondi.eecs.uci.edu:repos/poker.git”

3.3 Building, Compilation, installation

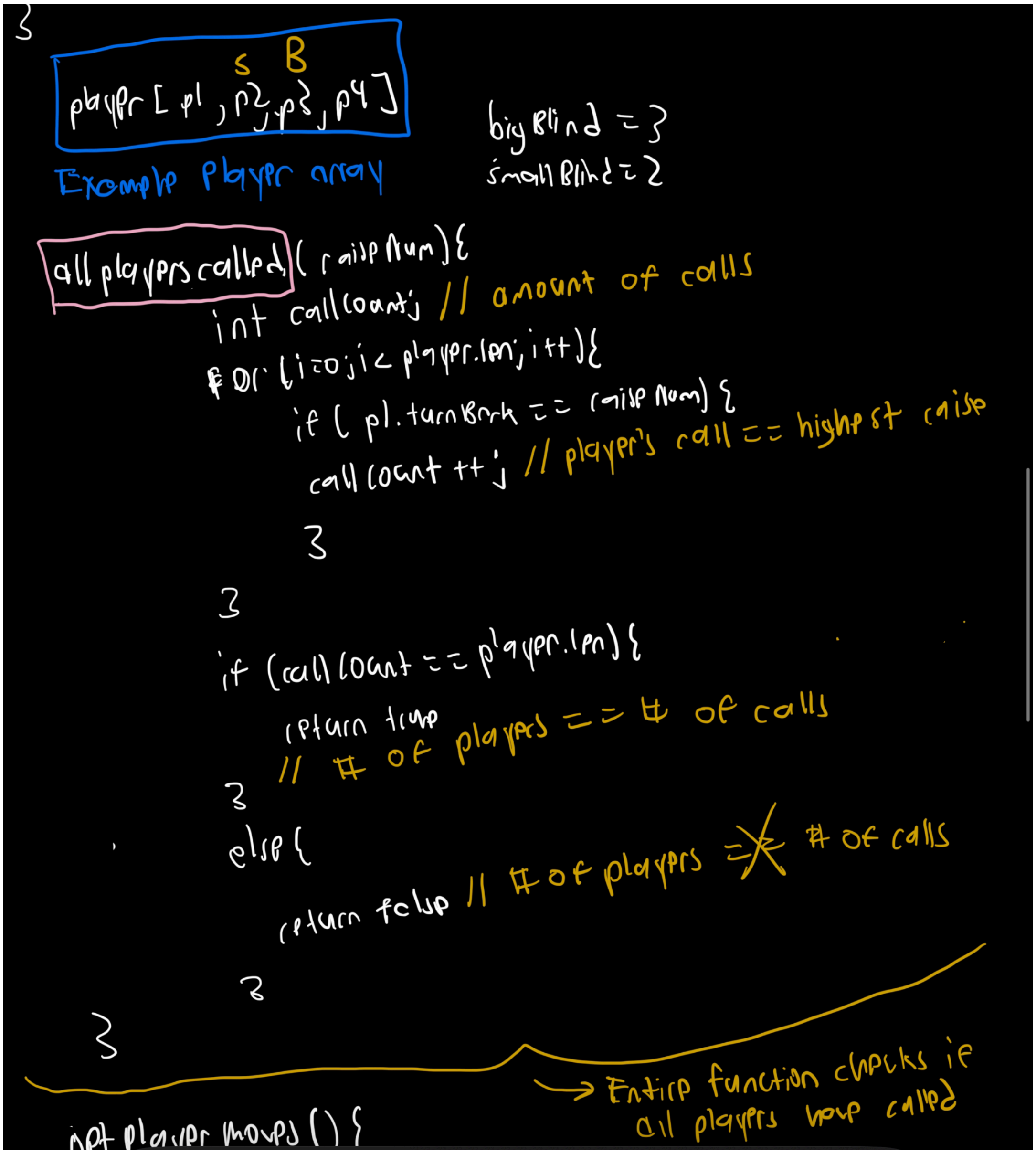
- Follow the unpacking and configuration instructions
- Navigate to home file directory
- make all
- cd bin/
- ./Poker

4 Documentation of packages, modules, interfaces

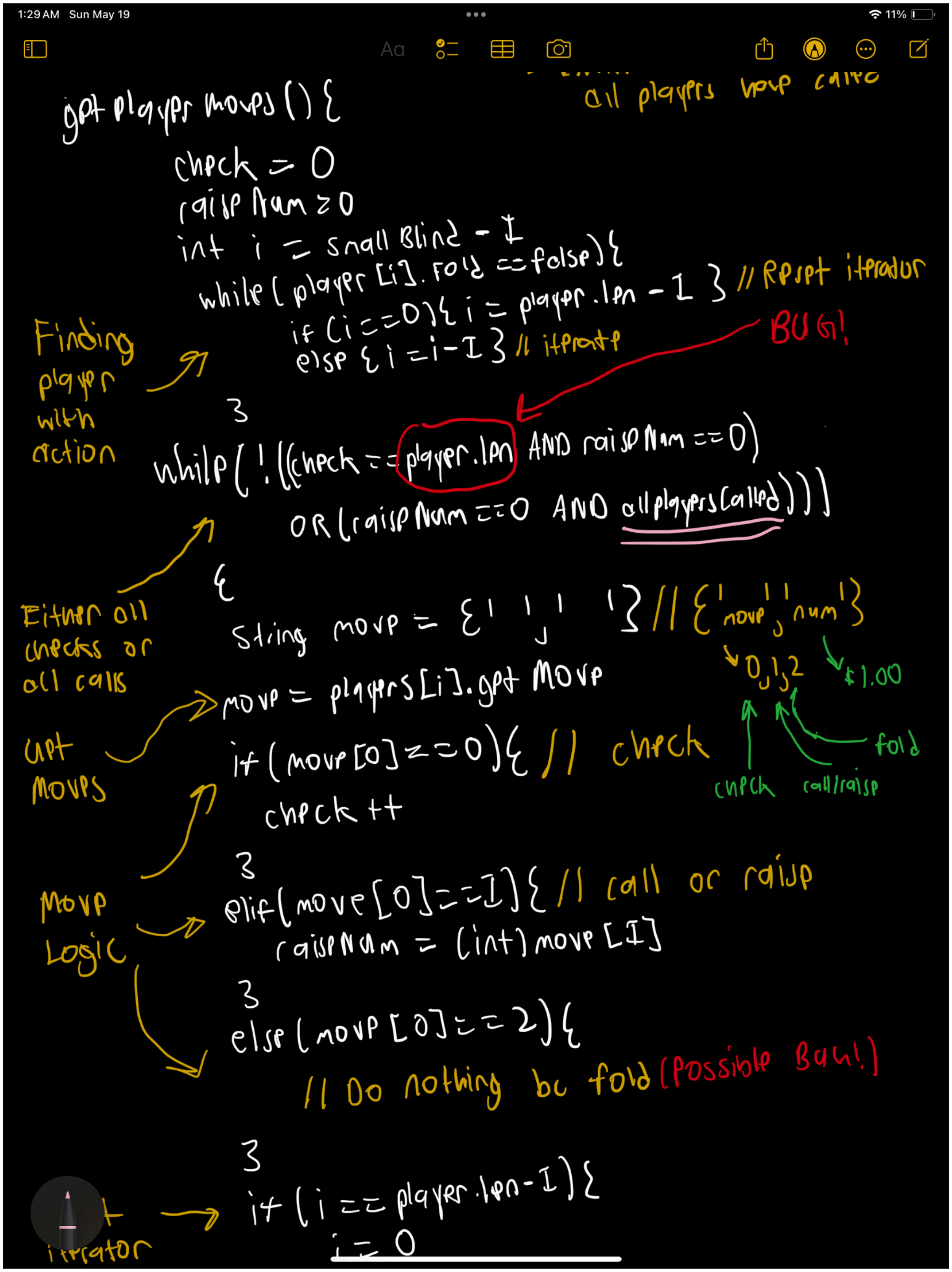
4.1 Detailed description of data structures

- Player

○

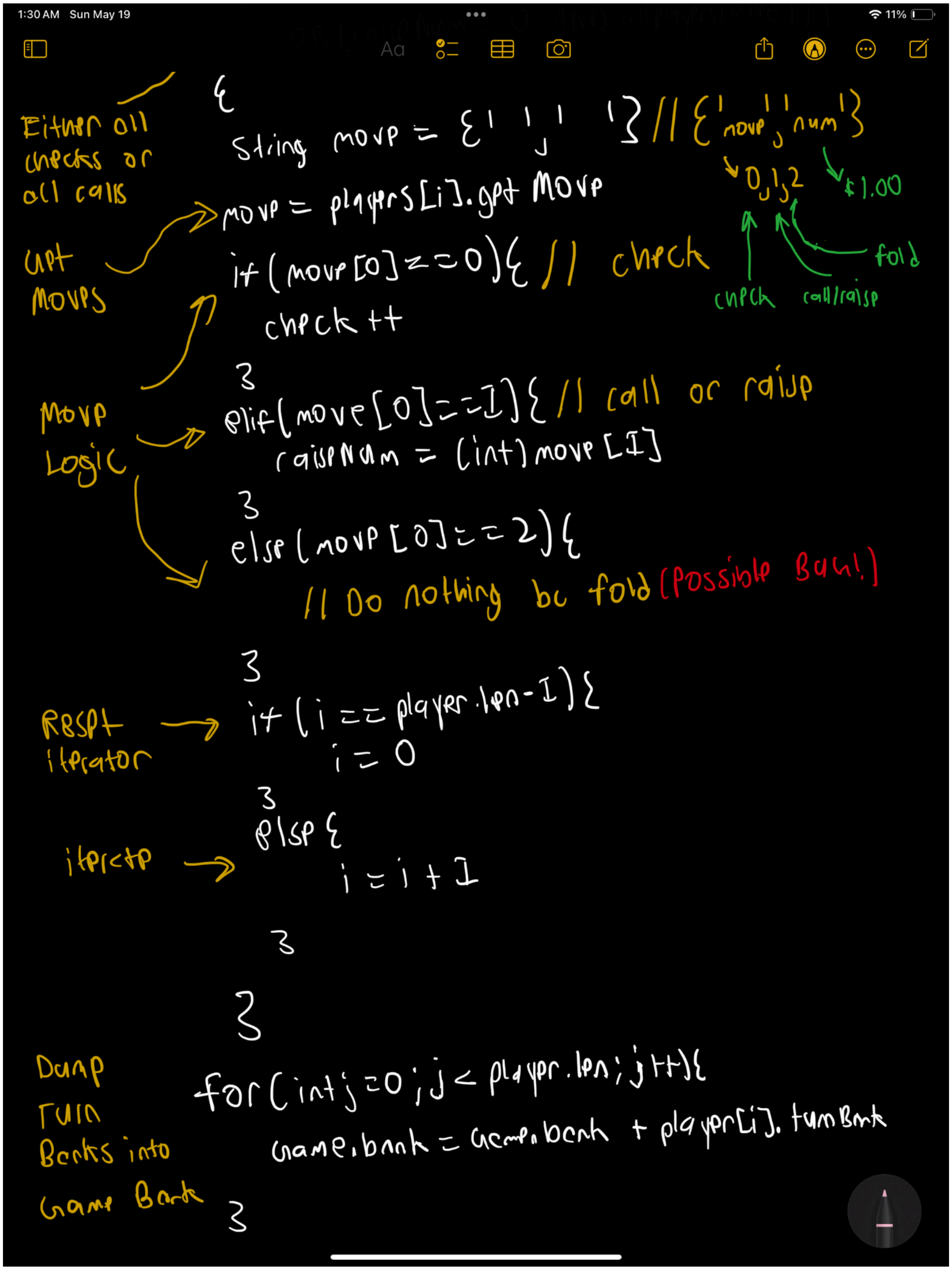


```
}
        s  B
player [ p1 , p2 , p3 , p4 ]
Example player array

bigBlind = 3
smallBlind = 2

allplayerscalled ( raiseNum ){
          int callcount; // amount of calls
          for (i=0;i< player.len; i++){
               if ( p1.turnBack == raiseNum) {
                 call count ++; // player's call == highest raise

                 }

          }

          if (call count == player.len){
               return true
              // # of players == # of calls
          }
          else{
               return false // # of players ≠ # of calls

           }
   }
                         → Entire function checks if
                           all players have called
get player moves () {
```



```
get players moves () {
        check = 0
        raise Num = 0
        int i = small Blind - 1
        while ( player [i]. fold == false){
            if (i==0){ i = player .len - 1 } // Reset iterador
            else { i = i - 1 } // iterate

        }
   while( !((check == player.len AND raise Num == 0)
           OR (raise Num == 0 AND allplayersCalled)))

        {
         String move = {' ', ' '} // {'move', 'num'}
       move = players[i].get Move
       if (move[0]==0){ // check
          check ++

       }
       elif(move[0]==1){ // call or raise
          raiseNum = (int) move [1]
       }
       else (move [0]==2){
          // Do nothing bc fold (Possible Bug!)

       }
       if (i == player.len-1){
          i = 0
```

○



```
{
  String move = {' ',' '}// {'move','num'}
move = players[i].get Move
  if (move[0]==0){ // check
    check ++
  }
  elif (move[0]==1){ // call or raise
    raiseNum = (int) move[1]
  }
  else (move[0]==2){
    // Do nothing bc fold (Possible Bug!)

  }
  if (i == player.len-1){
      i = 0
  }
  else{
      i = i + 1

  }

}
for (int j =0; j < player.len; j++){
    Game.bank = Game.bank + player[i].turnBank
}
```

- Deck
  - Char Deck [52][2]
    - An 52 array of 2 strings to represent the cards in the deck

4.2 Detailed description of functions and parameters

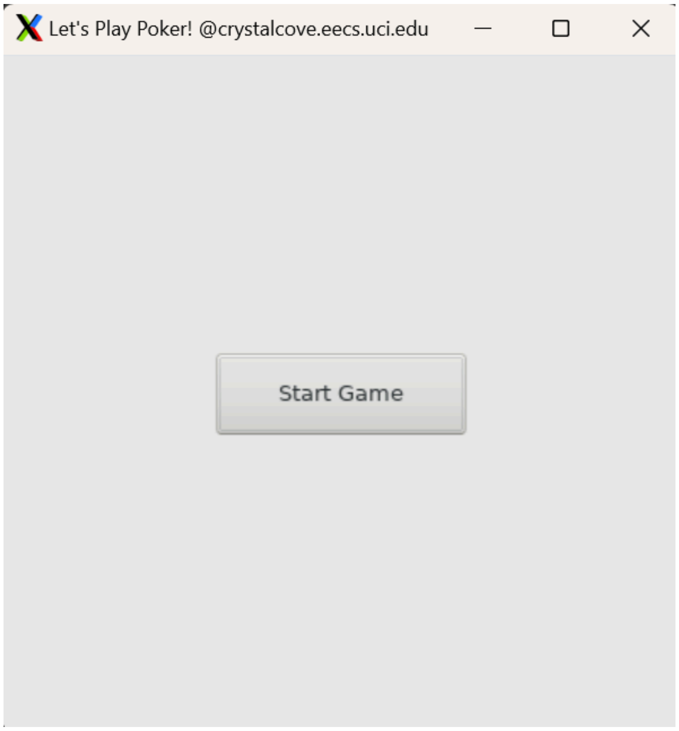
Poker.c

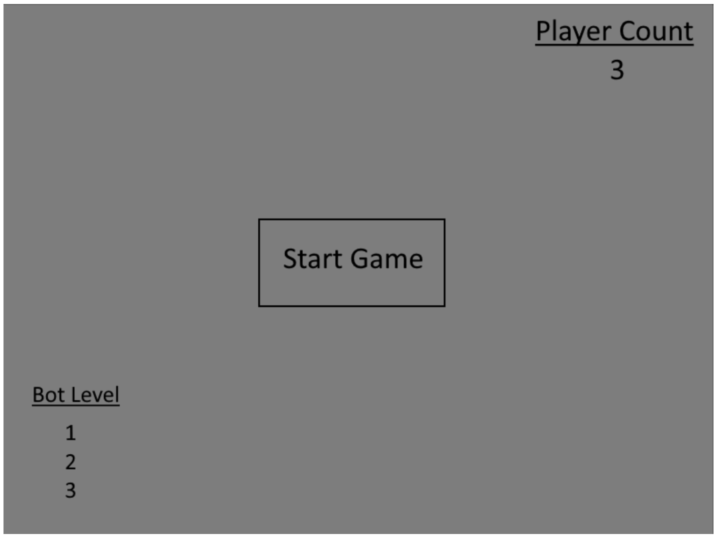
- int main(void);
  - The main function contains the finite state machine which controls the current state of the program

Bot.c

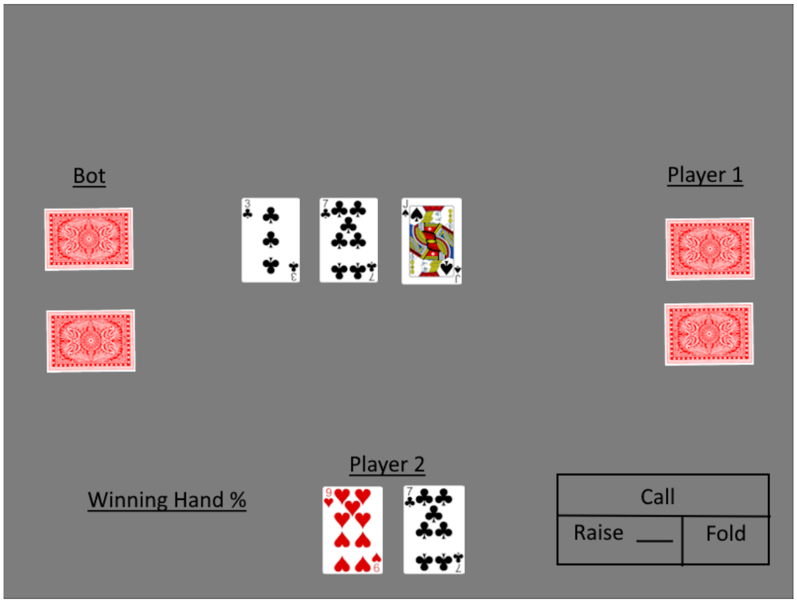
- HandPercentage(Player *Hand, char *Table)
  - A function to read in the cards dealt to the player and the cards dealt to the table and calculate the percentage of success for the current hand
- DecideMove(Player *Hand, char *Table)
  - A function that decides when to raise, check, or fold based on randomization balanced with the players current HandPercentage

Gui.c

- StartGame(Gtk *widget, Gtk *button)
  - Pressing a button to start the button
  - 



- DisplayHand(Player *Hand, char *Table)
    - Function to display the hand of the players



- UpdateHand(Player *Hand, char *Table)
    - Update the hand of each player

4.3 Detailed description of the communication protocol

poker.c

- int main(void);
    - Input: void
    - Output: error code (Expects a 1 for no error)
- int initializeConnection

Description: Initialize connection to server

- void closeConnection
  Description: Close connection to server
- int sendData
  Description: send data to server
  Parameter: Pointer to data that needs to be sent, size of data
- int receiveData
  Description: Receive data from server
  Parameter: Pointer to buffer, size of buffer

1. Connection:
   - Connect: Client request to connect
   - Disconnect: Client ends connection
2. Game State
   - Server deals hand
   - Server shows cards on table
   - Client sends player actions to choose from
   - Server updates the pot
   - Server tells client new round has started
   - Server tells client game ended

5 Development plan and timeline

5.1 Partitioning of tasks

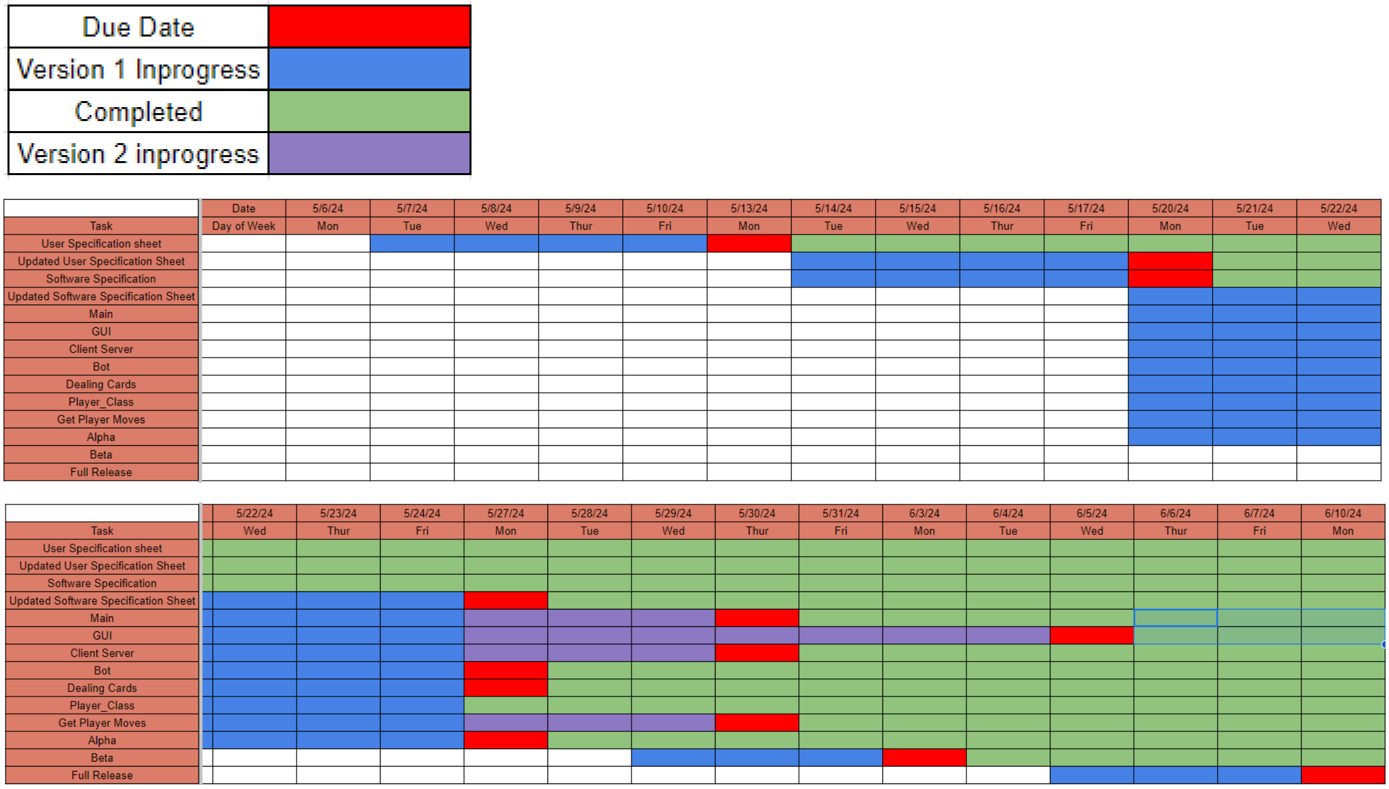
- player_class
- getPlayerMoves
- dealingCards
- clientServer, main
- GUI
- Bot
- Documentation

5.2 Team Member Responsibilities

- Ishika: GUI, clientServer, main
- Anna: GUI

- Fili: Bot, documentation
- Bilal: clientServer, main, dealingCards, player_class
- Tuaha: clientServer, main, dealingCards, player_class
- Rishi: getPlayerMoves (GUI)

Timeline



Due Date	🟥
Version 1 Inprogress	🟦
Completed	🟩
Version 2 inprogress	🟪

	Date	5/6/24	5/7/24	5/8/24	5/9/24	5/10/24	5/13/24	5/14/24	5/15/24	5/16/24	5/17/24	5/20/24	5/21/24	5/22/24
Task	Day of Week	Mon	Tue	Wed	Thur	Fri	Mon	Tue	Wed	Thur	Fri	Mon	Tue	Wed
User Specification sheet			🟦	🟦	🟦	🟦	🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩
Updated User Specification Sheet								🟦	🟦	🟦	🟦	🟥	🟩	🟩
Software Specification								🟦	🟦	🟦	🟦	🟥	🟩	🟩
Updated Software Specification Sheet												🟦	🟦	🟦
Main												🟦	🟦	🟦
GUI												🟦	🟦	🟦
Client Server												🟦	🟦	🟦
Bot												🟦	🟦	🟦
Dealing Cards												🟦	🟦	🟦
Player_Class												🟦	🟦	🟦
Get Player Moves												🟦	🟦	🟦
Alpha												🟦	🟦	🟦
Beta														
Full Release														

	5/22/24	5/23/24	5/24/24	5/27/24	5/28/24	5/29/24	5/30/24	5/31/24	6/3/24	6/4/24	6/5/24	6/6/24	6/7/24	6/10/24
Task	Wed	Thur	Fri	Mon	Tue	Wed	Thur	Fri	Mon	Tue	Wed	Thur	Fri	Mon
User Specification sheet	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
Updated User Specification Sheet	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
Software Specification	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
Updated Software Specification Sheet	🟦	🟦	🟦	🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
Main	🟦	🟦	🟦	🟪	🟪	🟪	🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩
GUI	🟦	🟦	🟦	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟥	🟩	🟩	🟩
Client Server	🟦	🟦	🟦	🟪	🟪	🟪	🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩
Bot	🟦	🟦	🟦	🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
Dealing Cards	🟦	🟦	🟦	🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
Player_Class	🟦	🟦	🟦	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
Get Player Moves	🟦	🟦	🟦	🟪	🟪	🟪	🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩
Alpha	🟦	🟦	🟦	🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
Beta						🟦	🟦	🟦	🟥	🟩	🟩	🟩	🟩	🟩
Full Release											🟦	🟦	🟦	🟥

References

Poker Rules detailed on Canvas by TA

State of California Texas Hold 'Em Rules 2016 Revision

Index

All in 3, 9

Bet 3, 4, 5

Blind 3, 4, 5

Boat 4

Call	3, 4, 9, 10
Card Hands	1, 3, 13
Check	5, 7
Class	5, 6, 12, 21, 22
Client	1, 2, 6, 7, 9, 18, 19
Deck	3
Doubly Linked List	5, 6, 12
Enumerator	5
Flop	4, 5, 6, 7
Flush	3
Fold	3, 5, 7
Four of a kind	3
Full house	3
Formal card hand names	3
Git	5, 15
GUI	5, 8, 9, 12, 13, 14, 19, 21, 22
Hand	2, 3, 4, 7, 8, 11
High card	3
House	3, 4
Int	5, 7, 8, 9, 12, 19, 20, 21
Main	2, 5, 6, 8, 12, 18, 20, 21, 22
Null	5
One pair	3
Player	2, 3, 4, 5, 6, 7, 8, 9, 10
Position	4
Pot	3, 4
Pointer	5, 7, 10, 12
Pre-flop	4
Pull	5
Push	5
Quads	4
Raise	3, 4, 5, 7
River	4
Rank	3, 4
Royal Flush	3
Server	1, 3, 4, 8, 10, 13, 18, 19

Straight	3
Straddle bet	4
Straight Flush	3
Struct	2, 5, 6, 12, 15
Suit	3, 4
Texas Hold ‘Em	4
Three of a kind	3
Turn	4
Two pairs	3
Widget	6, 19
Void	5, 9, 10, 19, 20, 21

Copyright