Goldfish 1.0

By Kunal Paode, Jeffrey Yu, Tommy Kung, Zachary Qin, Chenhao Yang

Table of Contents

Glossary

1. GUI (Graphical User Interface)

   -an interface that uses icons/menus, and a mouse to manage interaction with the system

2. Doubly linked list

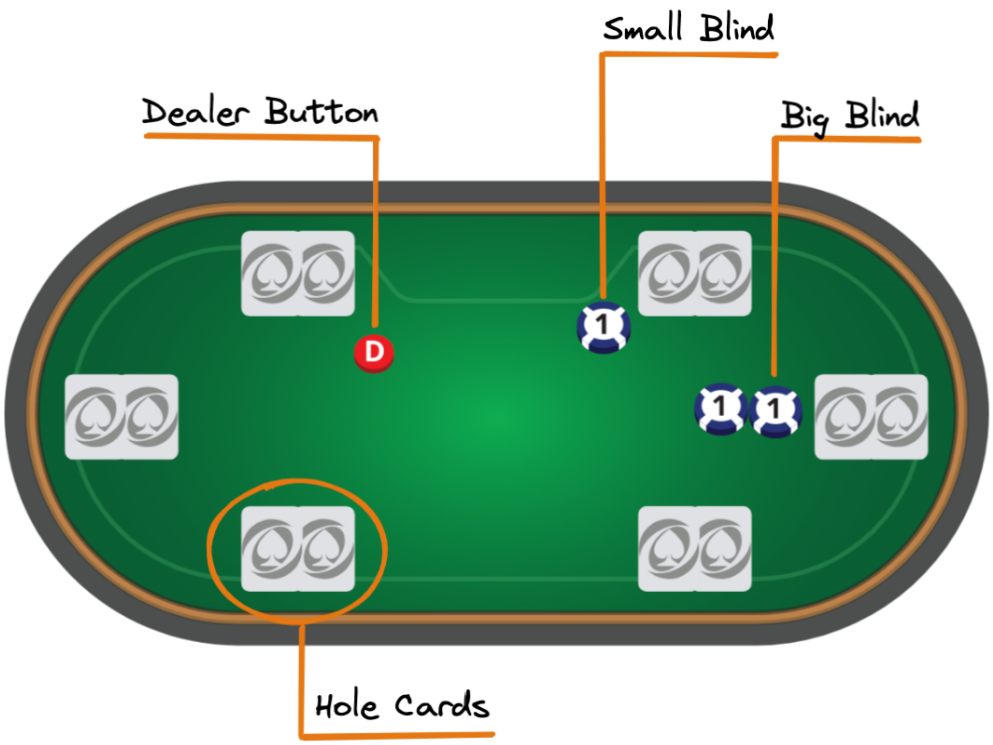
   -a linked data structure that consists of nodes that have pointers to the previous and next node. In which navigation is possible in both directions.

3. Enum

   -a data type that allows assignment of naming conventions to constants

4. Struct

   -structure is another user defined data type available in C that allows to combine data items of different kinds.



5. Dealer Button
   A round disc that sits in front of a player and is rotated one seat to the left every match. Helps to determine which player at the table is the acting dealer.

6. Small Blind
   A forced bet that begins the wagering. The player directly to the left of the button posts the small blind. Generally half the amount of the big blind.

7. Big Blind
   A forced bet that begins the wagering. the player directly to the left of the small blind posts the big blind.

8. Hole/Pocket Cards
   Two face down cards dealt to each player at the start of the game. Hole cards are used in combination along with the community cards to build a player's best possible five-card poker hand.

1. Community Cards
   Five face-up cards free for each player to use in combination with their hole cards to build the best possible five-card poker hand.

2. Flop: Round in which the first three community cards are dealt

3. Call: Match the betting amount of the big blind

4. Raise: Increase the bet within specific limits of the game

5. Fold: Throw hand away. Cards go into the muck

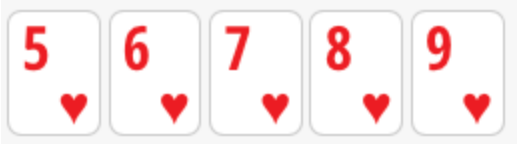
6. Check: Pass the action to the next player in hand (do nothing)

7. The Muck: Pile for cards no longer in play

8. Showdown: More than one player remains at the last round, so each remaining player shows their hand to determine who has the best

9. Five-Card Poker Hands (Best to Worst)
   Note: There is no Suit-Ranking in Texas hold'em
   1. Royal Flush: The best possible hand: 10, J, Q, K, A…all same suit

   2. Straight Flush: Five cards of the same suit in sequential order



Note: Straight Flush can be ranked best to worst based on highest card

3. Four-of-a-Kind: Any four numerically matching cards

   Note: in this example, the King of Diamonds can be replaced with any other card and this hand still applies

   Note: Four-of-a-kinds can be ranked from best to worst based on highest four-of-a-kind

4. Full House: Combination of three-of-a-kind and a pair in the same hand

   Note: Full Houses can be ranked from best to worst based on highest three-of-a-kind, then highest pair

5. Flush: Five cards of the same suit in any order

   Note: a Flush can be ranked from best to worst based on first highest-ranking card, then second, and third, etc…

6. Straight: Five cards of any suit, in sequential order

   Note: a Straight can be ranked from best to worst based on highest-ranking card

7. Three-of-a-Kind: Any three numerical matching cards

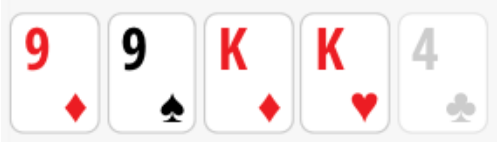
   Note: in this example, the Four of Clubs and Five of Hearts can be replaced with any other card and this hand still applies

   Note: a Three-of-a-kind can be ranked from best to worst based on highest three-of-a-kind, then (out of the 2 other cards remaining,) highest-ranking card

8. Two Pair: Two different pairs in the same hand

Note: in this example, the Four of Clubs can be replaced with any other card and this hand still applies

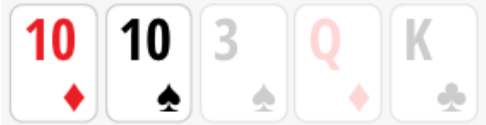
Note: a Two Pair can be ranked from best to worst based on highest-ranking pair, then second-highest pair, then highest-ranking final card



9. One Pair: Any two numerically matching cards

Note: in this example, the Three of Spades, Queen of Diamonds, and King of Clubs can be replaced with any other card and this hand still applies

Note: a One Pair can be ranked from best to worst based on highest-ranking pair, then (out of the 3 remaining cards,) highest-ranking cards



10. High Card : The highest ranked card in your hand with an ace being the highest and two being the lowest

Note: in this example, the Two of Hearts, Four of Diamonds, Eight of Diamonds, and Queen of Spades can be replaced with any other card and this hand still applies

Note: a High Card can be ranked from best to worst based on highest-ranking card, then second-highest card, etc…

Player Software Architecture Overview

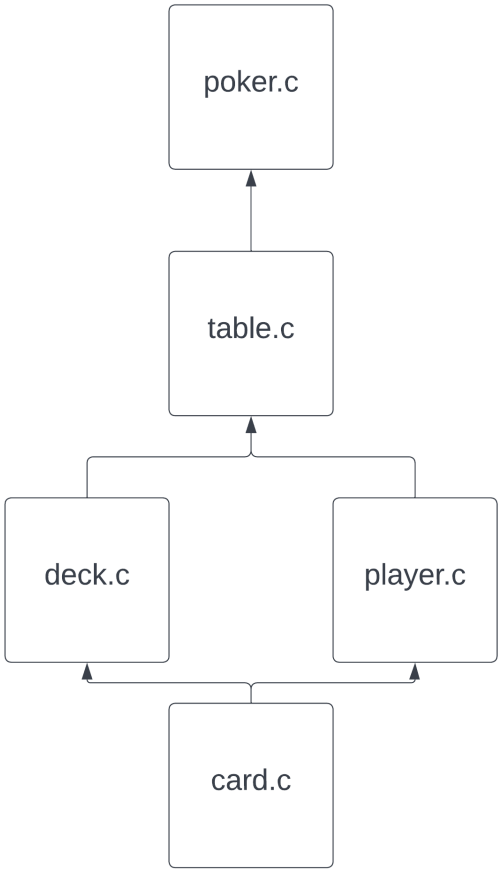
1.1 Main data types and structures

Main enums:
enum Suit
enum Num
enum Token
enum HasFolded

Main structs:
struct Card
struct Deck
struct Player
struct Table

1.2 Major software components

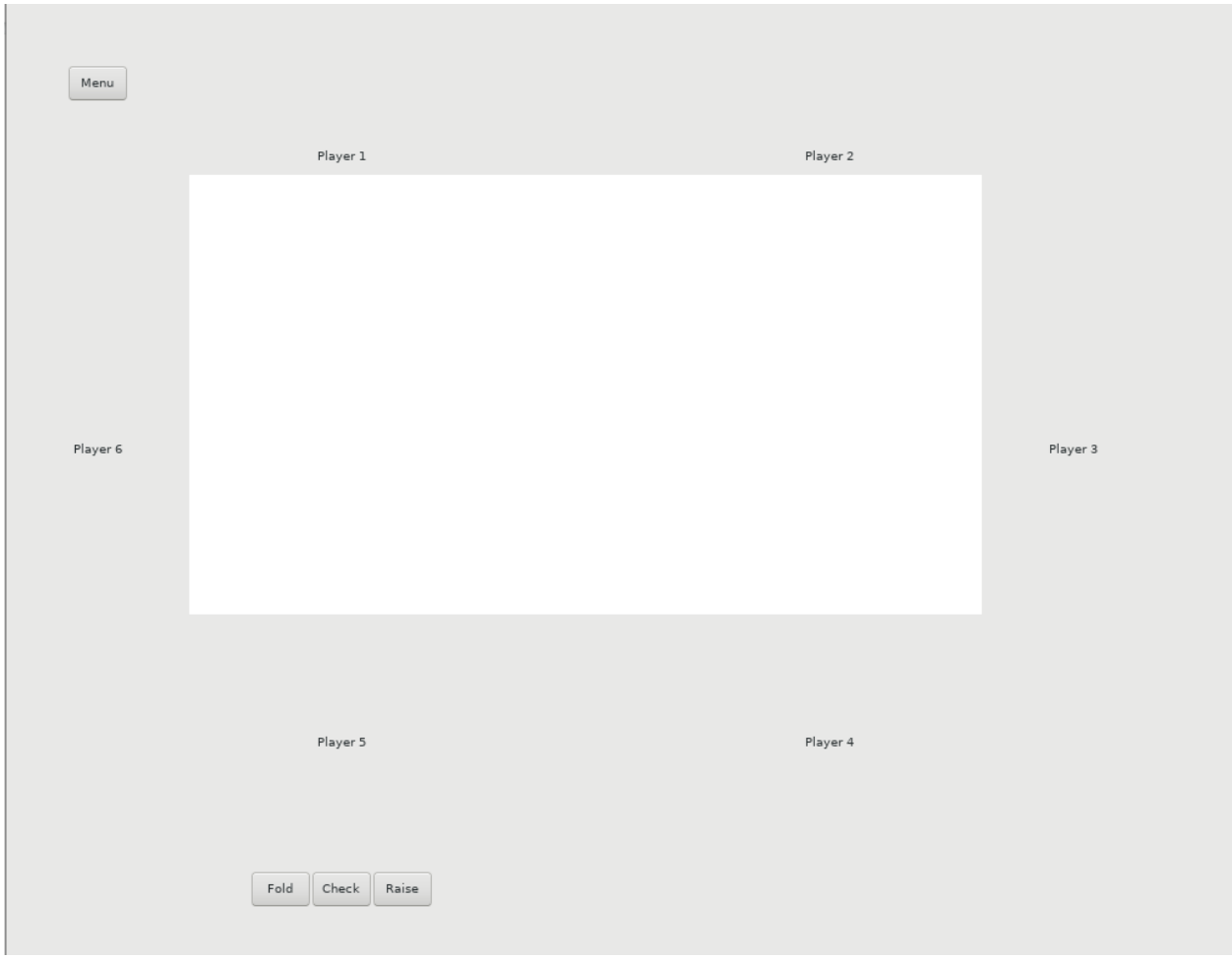
- Diagram of module hierarchy



-

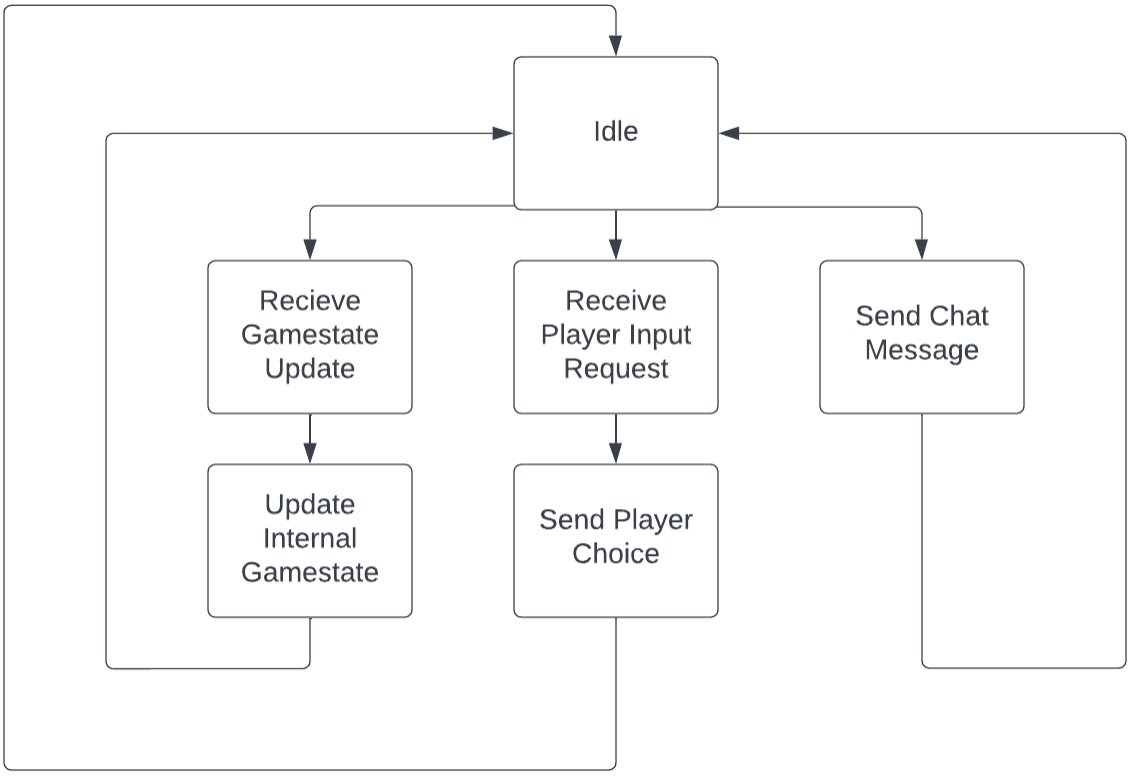
1.3 Module interfaces

- API of major module functions



1.4 Overall program control flow

Client Control Flow



Poker Server Software Architecture Overview

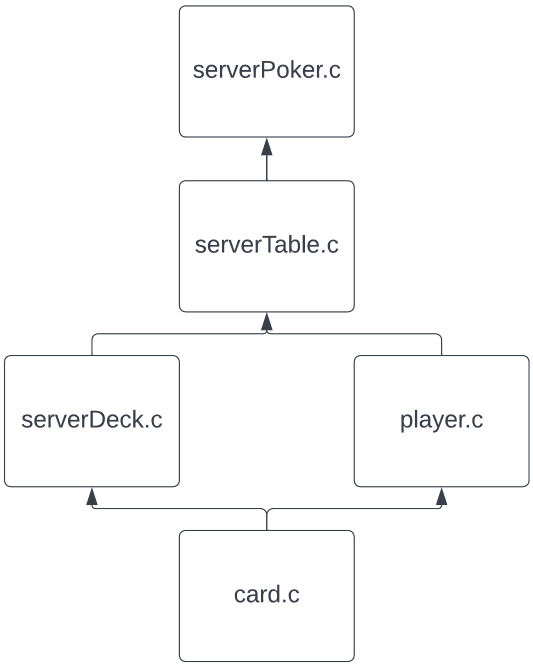
2.1 Main data types and structures

Main enums:
enum Suit
enum Num
enum Token
enum HasFolded

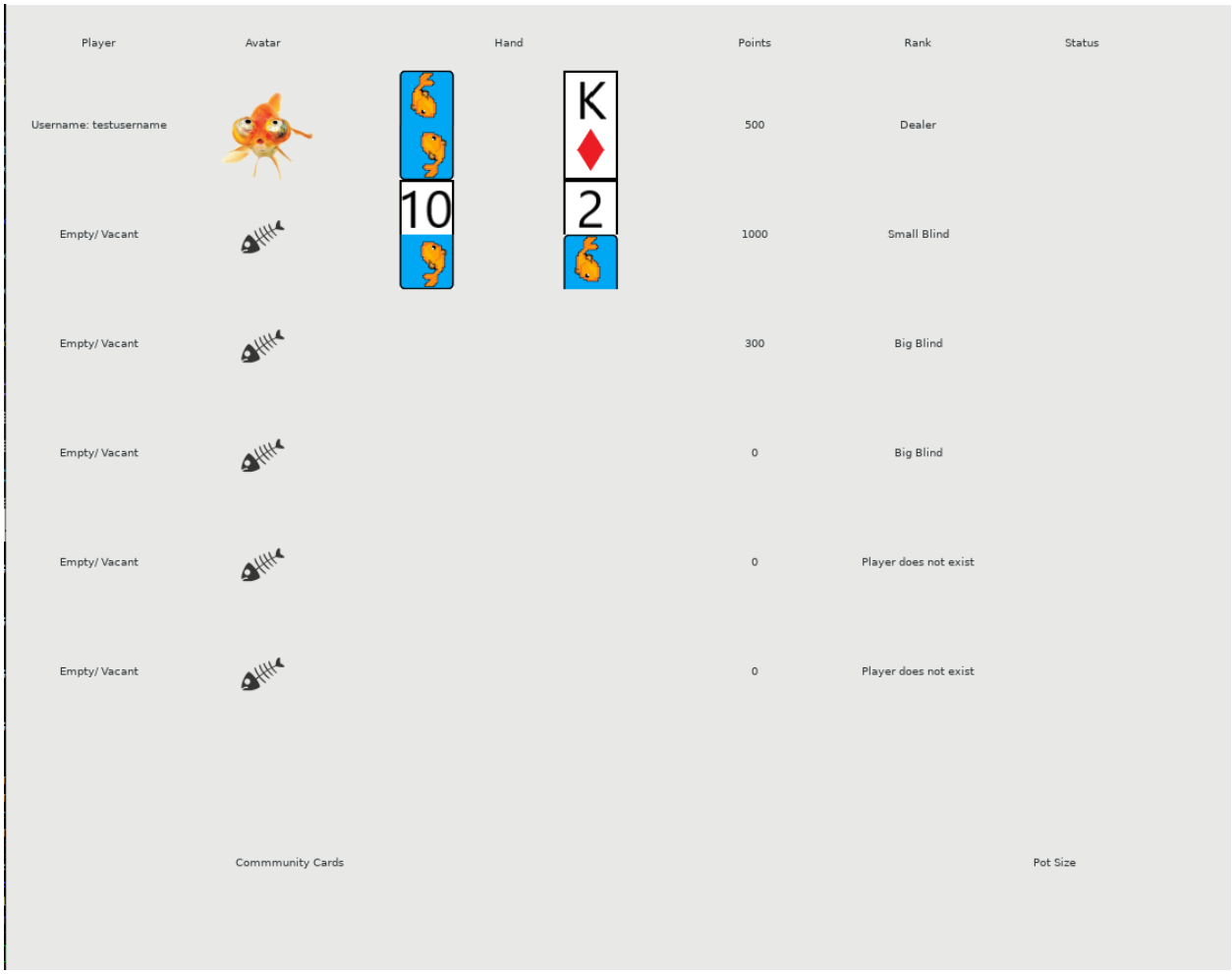
Main structs:
struct Card
struct Deck
struct Player
struct Table

2.2 Major software components

- Diagram of module hierarchy

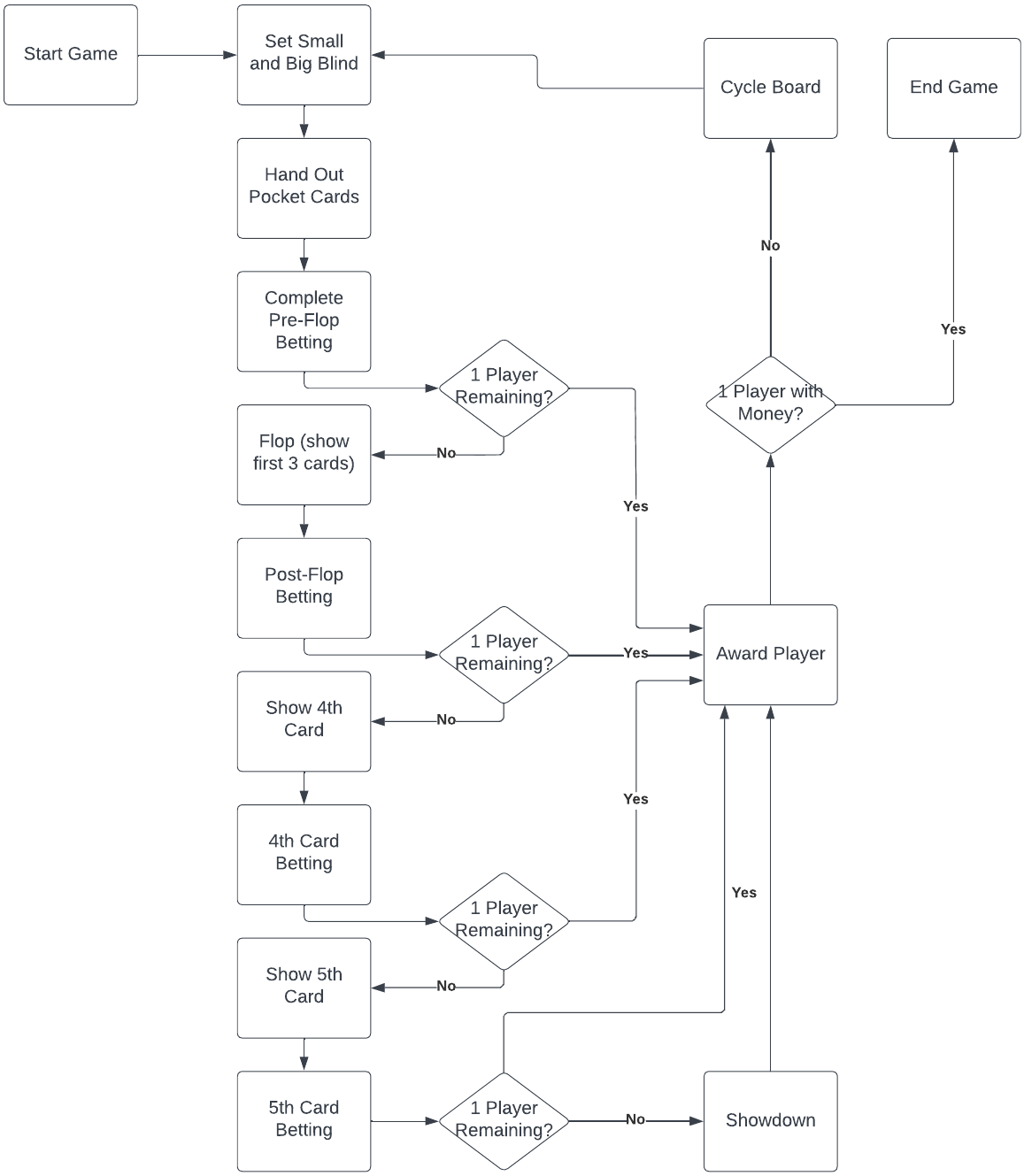


2.3 Module interfaces



2.4 Overall program control flow

Server Game Control Flow:



Installation

2.1. System Requirements

Linux with display function
std=c11 enabled
gtk library installed

2.2 Setup and configuration

Download tar.gz file and Makefile
Extract tar.gz file with command “gtar xvzf”
[Linux] Type “make all” to build program
[Linux] One Player host the server Type “./GoldfishServer”
[Linux] Type “./GoldfishClient 10111”

2.3 Uninstalling

[Linux] Type “make clean”
[Linux] Type “rm Makefile”

Documentation of Packages

3.1 Data structures

- Snippets of source code

```
typedef struct {
    Suit suit;
    Num num;
} Card;

typedef struct {
    int totCard;
    Card cards[52];
} Deck;

typedef struct {
    Button token;
    int money;
    Card pocket[2];
    HasFolded hasFolded;
    int currentBetAmount;
    char username[];
    int position;
} Player;

typedef struct {
    Player* players;
    Deck deck;
    int totPlayers;
    int totActivePlayers;
    int pot;
    int dealerNum;
    int currentMinBet;
} Table;
```

3.2 Functions and parameters

- Function prototypes and brief explanation

```
//handles entire betting round by iterating through table until all
players have either bet the same amount or folded; Both sends and requests
player info
Table BettingRound(Table table, int startingPlayer)
//sets big and small blind bets
Table SetBigAndSmallBlind(Table table, int smallblind, int bigblind)
//Places bet and updates information on table and for players
Table PlaceBet(Table table, int playerNum, int amount)
//returns a position of a player who is on the table who has money
int GetNextPlayerWithMoney(Table table, int playersAhead,int startingPos);
//sends game info to designated player
void SendGameState(Table table, int PlayerNum);
//requests player's choice
void RequestPlayerInput(Table table, int PlayerNum);
//returns position of player with best hand
int BestHand(Table table);
```

3.3 Input and output formats

The server will send information to the client each time an event occurs that requires a client to update their own information. The server will convey this information purely through a protocol written in text. Below is an example of such a message. The parenthesis surrounding the “1” in front of the “PLAYER_GAMESTATE” denote that the message is intended to provide the appropriate information for the first player only, hence each other player’s card is not revealed. The match is currently in the pre-flop betting round after the blinds have already been set, and so it has been denoted as round 2. The message first states the table properties, then each player’s properties. The action of the fourth player is set to “Waiting” indicating that the player has not made a move yet.

```
Server:  PLAYER_GAMESTATE(1)
                TABLE
                POT: 734
                CARDS: N/A
                MINIMUM_BET: 407
                DEALER_NUM: 4
                ROUND: 2
                TOTAL_ACTIVE_PLAYERS: 4
                TOTAL_PLAYERS: 4

                PLAYER_NUM: 1
                USERNAME: John
                POCKET: TwoSpades NineDiamonds
                MONEY: 1092
                BET_AMOUNT: 109
                ACTION: AutoBet
                COIN: SmallBlind

                PLAYER_NUM: 2
                USERNAME:  Jane
                POCKET: Unknown1 Unknown2
                MONEY: 503
                BET_AMOUNT: 218
                ACTION: AutoBet
                COIN: BigBlind
```

PLAYER_NUM: 3
USERNAME: Jack
POCKET: Unknown1 Unknown2
MONEY: 708
BET_AMOUNT: 407
ACTION: Raise
COIN: None

PLAYER_NUM: 4
USERNAME: Jason
POCKET: Unknown1 Unknown2
MONEY: 708
BET_AMOUNT: N/A
ACTION: Waiting
COIN: Dealer

- Explanation of communication related function calls

  SendChatMessage();

  - The user sends a chat message to communicate with the rest of the players. The client sends the message to the server. The server then sends to all of the players to print out in the chat log.

  SendActions();

  - The user presses the button that they would like to do such as raise. The client would send the username, amount, and action to the server. The server translates it and updates the games and sends the new information to the other users.

- Explanation of communication protocol flags/settings

  SendActions();

- The user have to check if the actions is legal and have to send back a message to select another action.

Start of game

- The first player presses the start button once all the players are in. The server receives the signal and starts creating the deck of cards and passes out the cards to each of the players. The dealers would be randomly selected. Each player would only be able to see their own hands until the end of the round.

Development Plan and Timeline

4.1 Partitioning of tasks

Week	Goals
1	User Manual
2	Software Specifications Begin initial development of data structures - Specify - Some - Structures
3	Start testing client-server architecture Start the testing of the game Alpha Goldfish Software Release
4	Implement extra functions (?) Continue the testing of the game Modify changes Beta Goldfish Software Release
5	Modify any last-minute changes Final Goldfish Software Release

4.2 Team member responsibilities

Team member	Responsibility
Jeffrey	Server GUI /Client GUI
Zachary	Back-end Poker Game, Server Networking
Kunal	Back-end Poker Game, Server Networking
Tommy	Server GUI /Client GUI
Chenhao	Server Networking, Client Networking

Back Matter

Copyright

References

1. https://developer-old.gnome.org/gtk2/stable/index.html (For GTK library)
2. https://www.gtk.org/ (For GTK library)

Index

1. GUI: Graphical User Interface
   Def. An interface that uses icons/menus, and a mouse to manage interaction with the system
2. Client
   Def. Hardware/Software that request access to a service hosted by a server
3. Server
   Def. Network, computer program, or device the processes requests from a client connecting to the same port
4. Port
   Def. A number designated to a single process on a given IP address
5. Process
   Def. An application running on a computer
6. IP address

Def. A string of characters representing a device over the internet

7. Client Server Architecture

   Def. Software applications communicate via the Internet

8. Protocol

   Def. A set of rules for transmitting data between devices

9. Socket

   Def. An endpoint of a two way communication link between two programs

   a. bind(): associate the socket to an IP address
   b. listen(): the readiness to accept client connection request
   c. connect(): establish a connection between client and server
   d. accept(): accept the connection request from the client
   e. write(): write data to the server
   f. read(): reads the data from the client
   g. close(): shuts down the socket
   h. exit(): terminate connection with the client