“1-2-3 Pass”

Group name: FULL HOUSE
Roxelle Cortes
Jose Ingan
Natalie Soluren
Caroline Tolentino

Description of the game:

1-2-3 pass is a game of luck, and strategy. Players get handed 4 cards out of a pool of cards depending on the player size. For example, 2 players would play from a pool of cards comprising of all the suits of A and all the suits of 2, whereas three players would also have all the suits of 3. Players will declare “1-2-3 pass” and pass their card to the right. The game ends if a player both completes all the suits of a specific card and lay their cards first before anyone does.

Programming language used:

Java

External libraries used:

None

Integrated Development Environments:

InteliJ Idea Ultimate Edition

Link of Github Repository:

https://github.com/nbsoluren/1-2-3-Pass-Card-Game-

Game layout:

- **Server:**

Running the Server Side:

1. Go to the main directory of the project using the terminal and then type: javac server/*.java && java server.ServerMain <no of players from 2 to 13>

   e.g. javac server/*.java && java server.ServerMain 3

Game Load:

1. Once the terminal compiles the Java files, the terminal will output: "Shuffled Cards in play are:[AC, 3D, 3S, 3C, 2S, AH, 2H, 3H, 2D, AD, AS, 2C]"
   - The specified cards are the cards available for playing
   - The naming convention of the cards is first letter signifies the number (Ace, 2, 3, 4, 5, 6, 7, 8, 9, Ten, Jack, Queen, King) and the second letter signifies the suit (Spades, Hearts, Diamonds, Clubs).

     e.g. AC would signify Ace of Hearts
   - The shuffled cards released depend on the number of players joining the game.

2. Once the server side is loaded, the terminal will output that the server side is waiting for players to join the game.

   e.g. "Waiting for 3 more player/s on port 8818..."

3. Each time a player joins the server, the terminal will output the user's IP address and the player's position.

   e.g. "User from/192.168.1.10:64783 is now player#1"

Gameplay:

1. Once all the players have joined the game, server side will print "Game about to start" and will direct the players to pass their cards.
2. Server side will print the initial cards of all the players and will wait until the players have passed their cards.
3. Each time a player has passed a card, the server side is notified of the card that the player passed. It will output the player's position and the card that the player passed.
4. Once all the players have passed, the server side will show the current pool of cards that each player has.
5. Server side will instruct all the players to pass their cards until a player has matched all his/her cards.

6. Once a player/players matched all their cards, server side will instruct players to press Enter. The first player to press Enter wins the game.

- **Client**

Running the Client Side:

1. Go to the main directory of the project using the terminal and then type: javac gameUi/*.java
2. Then type: javac client/*.java && java client.EchoClient <insert IP address of the server>
   e.g. javac client/*.java && java client.EchoClient 192.168.1.10
3. The terminal will then print the initial cards that the player received from server
   - The naming convention of the cards is first letter signifies the number (Ace, 2, 3, 4, 5, 6, 7, 8, 9, 10, Jack, Queen, King) and the second letter signifies the suit (Spades, Hearts, Diamonds, Clubs).
     e.g. 3D2D2C3H
   - The player has the cards 3 of Diamonds, 2 of Diamonds, 2 of Clubs and 3 of hearts as the player's initial deck.

Gameplay:

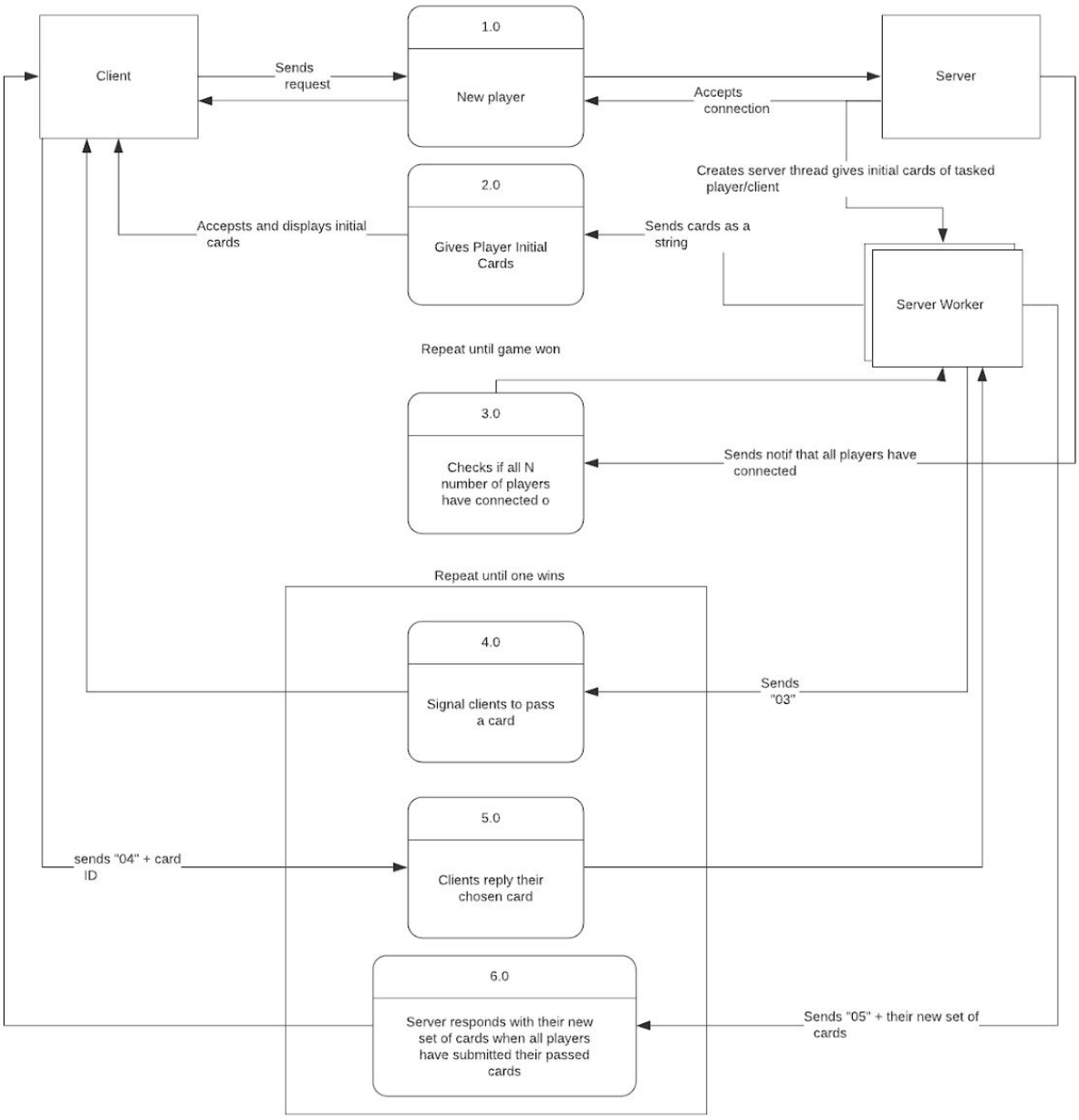
1. Once all the players have joined the game, the server side will direct the client side players to pass their cards.
2. The players will be asked to enter a card. The players should only choose from their list of cards.
3. To choose a card the player should input the card's number followed by its suit.
   e.g. 2H3SASAH
   - the player can only input 2H or 3S or AS or AH
4. After choosing a card, the player is notified of the card that it passed
   e.g. "You, player #2 sent me 3C"
5. After all the players have passed, the player is notified of its current pool of cards. The player will have its card passed to the player on the right.
6. After all the players have passed, the player is notified of its current pool of cards. The player will have its card passed to the player on the right.
7. Once a player has matched all his/her cards, the player will be prompted to press "Enter".
8. The first player who completes all suits of a specific card and press "Enter" wins.

Protocol:

The protocol that we chose to employ is **TCP**. **UPD** is used by most real-time games as it would not be beneficial for the game to keep retrying to send the same data if it's already outdated. 1-2-3 pass however is not a real-time game. For the game to work, it relies that all

players must receive their cards and at the same time the server must also receive all the cards passed by the players. This will prevent a player from passing a card again and again without the cue of the other players. Thus, our choice to use TCP to ensure that the data will always be received when asked.

- **Data Flow Diagram**



- **Structure of the data to be exchanged.**
  - Server tells the server worker (server thread) to send “03” to indicate that the clients should pass;
  - When the client gets this they will respond “04” + the ID of the card they wish to pass, for example:

    - “04TH” (10 of hearts) the client does not need to know what player number she is as the server worker thread already knows what client its handling.
- The server worker upon getting notified that all players have passed, sends new cards strings to all clients with their passed card replaced with a card passed by another person on their left. The structure would be like this “05” + 4 ID cards of their updated hand, example:
    - “052H2D2C2S” the player received 2 hearts, diamonds, clover and spades.