Group #: 15

Name:

Lawford Liu - hliu204@uic.edu
Jason Zhang - gzhang37@uic.edu
Hilda Mendoza-Avila - hmendo3@uic.edu
Aksa Chacko - achack8@uic.edu

Changes made in the Original code:

- GameInfo data member
    - playerId, OpponentId, playerPlays,OpponentPlay
    - Int TypeOfMessage for instruction when receiving new gameinfo what should be done with it
    - An ArrayList of clients on the server for update Gui clients list
- Gui update for both client and server;
    - Add a few Consumer<Serializable> object to make an easier call to update information according to instruction for gameinfo(Using Gui)
    - Add new Scene for list of clients waiting for challenges
    - Proceed to the game when successfully challenge or challenged
    - Clear listView whenever update ArrayList and adding a new LIstItems
- Server
    - Receive messages from client's challenges
        - Check status for challenged client then send them both the "setgame" instruction
    - Receive plays
        - Wait for another player's plays
            - Had both plays, evaluate
    - Receive client's quit or join information
        - Update accordingly
- Client
    - Receive instruction of successfully game challenge
        - Set game scene
            - Sit and wait for plays
    - Receive message need to be posted to the game status(evaluation of the games)
        - Update LIstView accordingly
    - Receive message opponent quit
        - Clean all gameinfo for this round
        - Back to clients choosing scene then update List of clients online accordingly
    - Quit
        - Set info.Onlineplayer to 0(offline) 3

The original specification of PJ# 3

- clientThread and Client passing GameInfo each other to communicate
- clientThread update ServerGui by calling callback.accept()
- Client update ClientGui by calling callback.accept()
- GameInfo contain Player_Plays and String that the message is going to display
- TheServer accept(make connection to) Clients, update ServerGui
    - Adding all the clients (clientThread) to an ArrayList

Member Contribution

Lawford Liu - Pdf writeup, ClientGui, ServerGui, helped with Server Class
Jason Zhang - ClientGui, Client, helped with Server Class
Aksa Chako - Server Class
Hilda Mendoza-Avila - Server Class

Server - 1）client1,2,3,4 connect to server. Update join information

//clientThread &clientsGui&ServerGui - gameinfo.instruction == “join” (which update message) - （or maybe not needed ）

Client -1) Scene choosing opponent

//clientGui

Client -2） 1 challenge 3, send gameinfo.opid gameinfo.insturction"challenge" to Serverthread

//clientGui -> client -> ClientThread -> Server

Server - 2) receive challenge message. Check both clientThread.gameinfo.status for availability. -----if (yes) {set status to false}

//Serverthread -> clientThread

Server -3) Update client1,3 clientThread.gameinfo.opid.
// ServerThread

Server -4) send tmp.gameinfo.instruction(set client Scene) to both clients
//ServerThread -> client ->clientGui

Client -3) send plays to server
//Client -> clientThread -> ServerThread

Server -5) receive 1 plays, check Id , put its play to clientthread.gameinfo.play and clientthread.gameinfo.opplays to its op id.
//clientthread

Server -6) check if they have both plays, if yes call whowon();
//clientthread

Client -4) send plays to server
//client -> clientthread

Server -7) receive 1 plays, check Id , put its play to clientthread.gameinfo.play and clientthread.gameinfo.opplays to its op id.
//clientthread

Server -8) check if they have both plays, if yes call whowon();
//clientthread

Server -9) set both clients.gameinfo.instruction/opId/plays/opplays to default. Send "whowon" to clients with gaminfo.instruction "endGame" (should also display message of who won)
//clientthread

Client -5) back to choosing opponents
//client -> clientgui

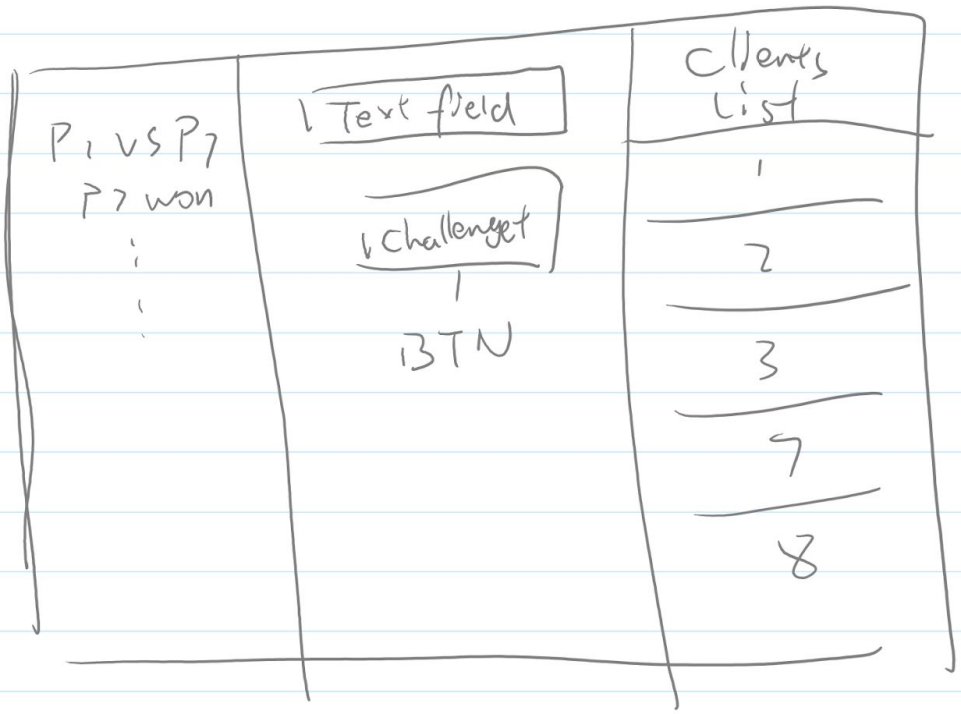
Client -6) quit. Send message to Server
// clientgui -> client -> clientthread -> serverthread

Server -10) update clients list , send back to client for updates
//???(update list) -> clientthread -> client -> clientGui

Client -6) updates client list
//client -> clientGUi

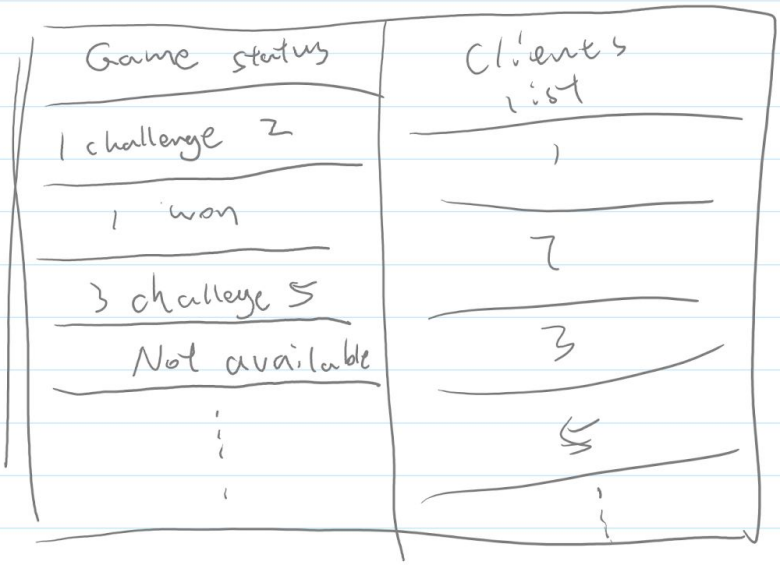
client GUI

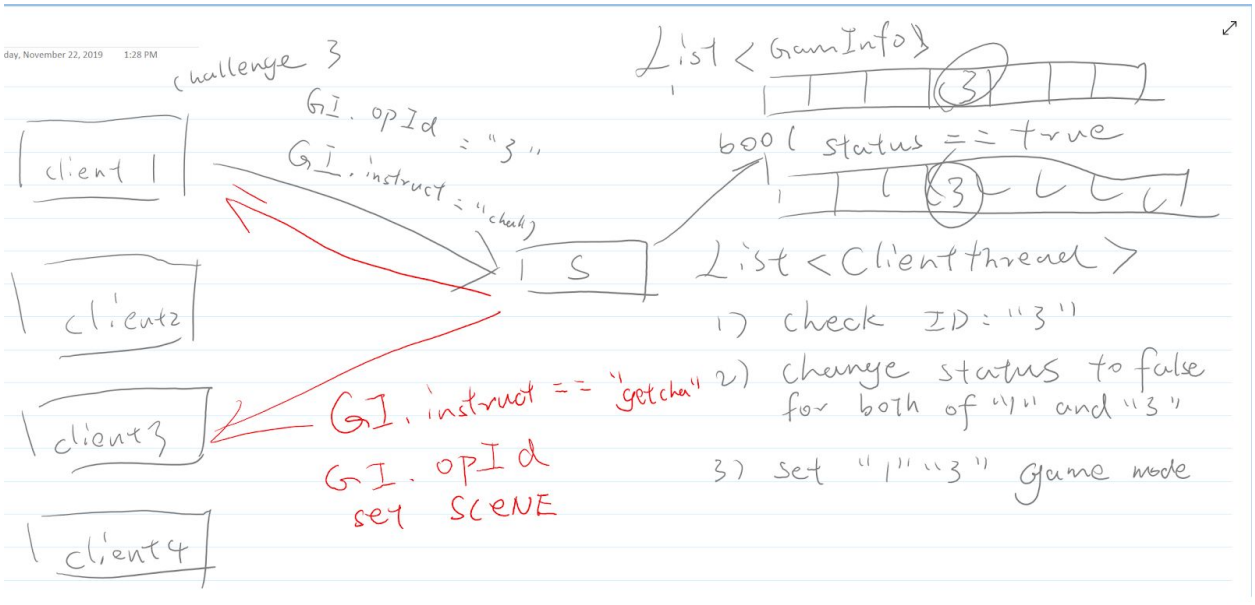
Friday, November 22, 2019 1:30 PM



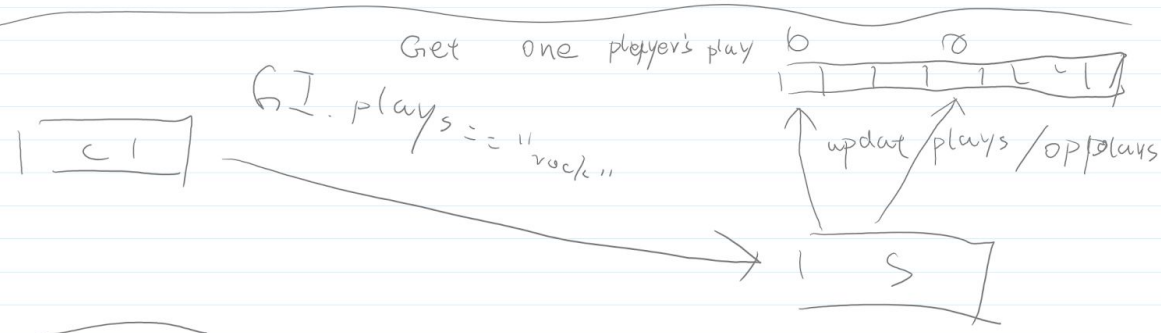
Server GUI

Friday, November 22, 2019 1:32 PM



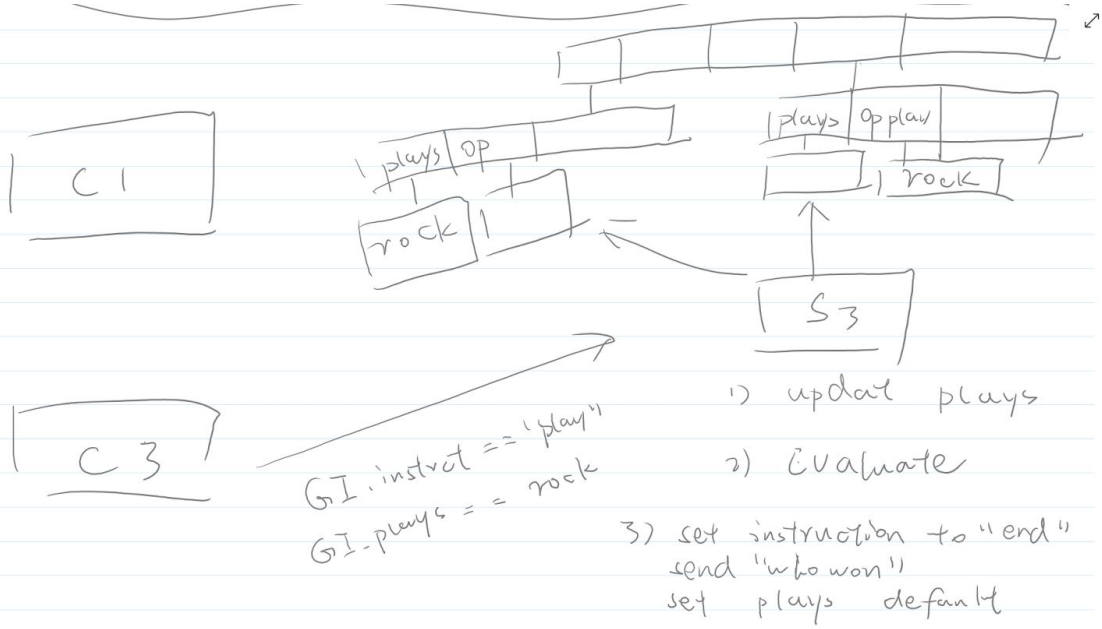


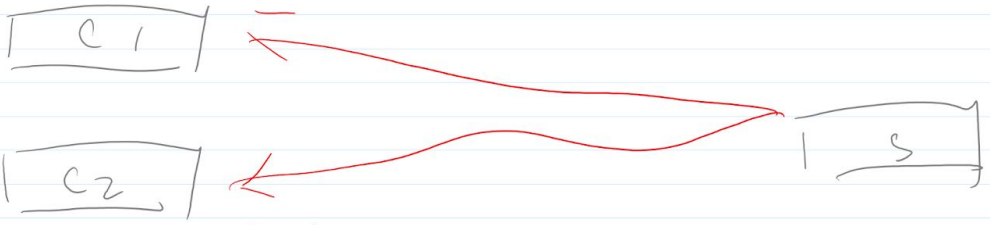
Get one player's play



1) updat plays and opplay
2) Evaluat,

```
if (has 2 plays) {
    logic(p1, p2)
}
updat client()
```





1) who won

2) instruction (back to choosing Scene)

3) Set OpID, plays, opplays default
for both in clientthread