CS 342
Project 4
Team 5
Cole Cunningham - ccunni3@uic.edu
Yihua Pu - ypu5@uic.edu
Haoran Yu - hyu63@uic.edu

Instructions to run:

1. Run ServerFx and select a port for the server
2. Start the server
3. Run ClientFx and select an IP and a port for each client
4. Connect clients to the server
5. Choose an opponent and challenge them
6. Play a round of RPSLS
7. Choose a new opponent and repeat

Changes to specifications:

- Support more than 2 clients
- Display all connected clients
- Matches consist of a single round

Changes to code:

ClientFX.java:

- Added a player select dropdown to allow players to challenge other players
- Changed the command handler function to implement the protocol described below
- Added a label to display the player's assigned name
- Removed the wait screen since challenges are allowed

Client.java:

- Changed constructor to accept one callback for GUI updates

Server.java:

- Replace two callback functions with a more general callback that can parse all the commands in our protocol

ServerFX.java

- Add a TextField to show the name of all the connected clients
- Add a label to display the number of connected clients

Game Protocol:

Messages: (command and parameter delimited by '#')

Sent By	**Command**	**Parameter**	**Reason**

Server	NAME	String: clientName	Inform client of its assigned name
Server	CONNECTED	String: clientName	Inform client that an opponent is connected
Server	DISCONNECTED	String: clientName	Inform client that an opponent is not connected
Client	QUIT	none	Inform server that client is quitting
Client	CHALLENGE	String: clientName	Inform server that the client challenges an opponent
Server	REJECT	none	Inform client that the challenged opponent cannot be played
Server	ACCEPT	String: clientName	Inform client that the challenged opponent can be played and the name of the opponent
Client	PLAY	Enum<PLAY> := {ROCK, PAPER, SCISSORS, LIZARD, SPOCK}	Inform server of client's play
Server	DRAW	Enum: play	Inform client that it tied with opponent
Server	WIN	Enum: play	Inform client that it won
Server	LOSE	Enum: play	Inform client that it lost

Logic:

1. Client connects to server
    a. Server generates client name and records it on client
    b. Server adds client to vector of currently connected clients
    c. Server logs new client connected and updates count
    d. Server sends NAME#{clientName} to client, client shows its name on GUI

    e. Server sends CONNECTED#{clientName} to the newly connected client for each existing connected client
    f. Server sends CONNECTED#{clientName} to all other connected clients for the newly connected client
2. Client receives NAME#{clientName}
    a. Set canPlay = false
    b. Set name = clientName
    c. Prompt user to select an opponent
3. Client receives CONNECTED#{clientName}
    a. Add clientName to list of opponents
4. Client sends QUIT
    a. Server logs that client quit
    b. Server sends DISCONNECTED#{clientName} to all other connected clients
5. Client receives DISCONNECTED#{clientName}
    a. Remove clientName from list of opponents
6. Client sends CHALLENGE#{clientName} to server
    a. Server checks if opponent (client specified by clientName) is currently playing
        i. If true, Server sends REJECT# to client
        ii. If false,
            1. Server sets opponent field on both client and opponent to each other
            2. Server sends ACCEPT#{clientName} to both client and opponent where clientName is the name of the opponent
7. Client receives ACCEPT#{clientName}
    a. Client logs that the user can make a play
    b. Client displays opponent's name (specified by clientName)
    c. Set canPlay = true
8. Client receives REJECT#
    a. Client logs that challenged opponent is currently playing
    b. Set canPlay = false
9. Client sends PLAY#{ROCK, PAPER, SCISSORS, LIZARD, SPOCK}
    a. Server checks if client has an opponent
        i. If true,
            1. Server records play on client object
            2. Server checks if client has opponent
                a. If not, Server sends REJECT#
            3. If opponent has made a play, server calculates winner
                a. If draw, Server sends DRAW#{opponentPlay} to both client and opponent
                b. Else,
                    i. Server sends WIN#{opponentPlay} to winner
                    ii. Server sends LOSE#{opponentPlay} to loser
                c. Always, Server sets client's opponent and play fields to null

        ii. If false, Server sends REJECT# to client

10. Client receives WIN#{play}
    a. Client logs what the opponent played ({play}) and that it won
    b. Set canPlay = false
11. Client receives LOSE#{play}
    a. Client logs what the opponent played ({play}) and that it lost
    b. Set canPlay = false
12. Client receives DRAW#{play}
    a. Client logs what the opponent played ({play}) and that it tied
    b. Set canPlay = false