Pseudocode

Project Server

Easy Group

Define static final integer (MAX\_MATCHES)

Define static in variable

Create an array of private socket for player1

Create an array of private socket for player2

Create an array of private object output stream for both players

Create an array of private object input stream for both players

Create an array of private data input stream for both players

Create an array of private data output stream for both players

Create a text area for logging in

Declare an array of private Boolean (test if server is connected)

**Start**

Create a scene and place it in the stage

Create a server socket

Create a while loop to check the amount of games that is been played (leaves loop when amount of games is more than the max)

In while loop

Define an integer for number of players (set =0)

Create an array of socket for player (for 2 players)

Set each index in the array to a socket

Create a while loop (loops if number of players is less than 2)

In while loop

If number of player is 1, listen for a connection request for that player

Inside if statement(Create try/catch)

In try

Write to data output stream for player 1

Read from data input stream for player 1

Find player 1’s host name/ip address and join

Create output stream to receive object from server (for player2)

Create output stream to receive data from server

Write integer to data stream (for player 2)

Flush object output stream (for player2)

Create input stream to receive object from server (p2)

Create input stream to receive data from server(p2)

Write object to output stream(p2)

Flush object output stream(p2)

Write object (play tiles) to object output stream(p2)

Flush object output stream(p2)

Set number of players to 2

In catch

Set first index in array socket to second index

Set second index to a socket

Find servers host and ip address

Create output stream to receive object from server (p1)

Create output stream to receive data from server

Write integer to data stream (p1)

Flush object output stream (p1)

Create input stream to receive object from server (p1)

Create input stream to receive data from server(p1)

Write object to output stream(p1)

Flush object output stream(p1)

Read object (play tiles)

Set number of players to 1

Else Statement

In Else statement

Set first index in array socket to second index

Set second index to a socket

Find servers host and ip address

Create output stream to receive object from server (p1)

Create output stream to receive data from server

Write integer to data stream (p1)

Flush object output stream (p1)

Create input stream to receive object from server (p1)

Create input stream to receive data from server(p1)

Write object to output stream(p1)

Flush object output stream(p1)

Read object (play tiles)

Set number of players to 1

Read Object (play tiles)

Write Object (play tiles) to object output stream (p1)

Flush object output stream (p1)

Create String class for user2’s username

Write string class to object output stream(p1)

Flush object output stream (p1)

Call run game function.

Increment the amount of matches

Define run game function

Set Boolean class = true

Write String object to object output stream for both players (Game Started)

Construct a thread, insert runnable classes as arguments

Create runnable class (Processing game)

Inside runnable class

Declare private object input/output stream variable

Declare private data input/ output stream variable

Declare private string variable

Implement the run method for the thread

While loop (loops if the Boolean class is true)

Define a try

If code is CHAT

Read object and convert to string

Write integer to output stream data

Write string object to output stream object

Flush output stream object

Else if code is CHANGE TILES

Write the integer to output stream data

Declare an array for play tile/read in the array to the input stream

Write object play tiles to output stream object

Else if code is play or game over

Create an array for board tiles and read it as an object

If code is play

Write integer PLAY to output stream data

Else if code is game over

Write integer GAME OVER to output stream data

Write object board tiles to output stream

Flush output stream

Create an array play tiles

Write play tiles to output stream

Write integer score to output stream

Flush output stream data

Define catch

Set Boolean class is connected to false

Define try

Close input stream

Close output stream

Close input stream data

Close output stream data

Define catch

Print out text displaying “Sockets are disconnected”

Return;

**End**