

Date: February 16, 2021

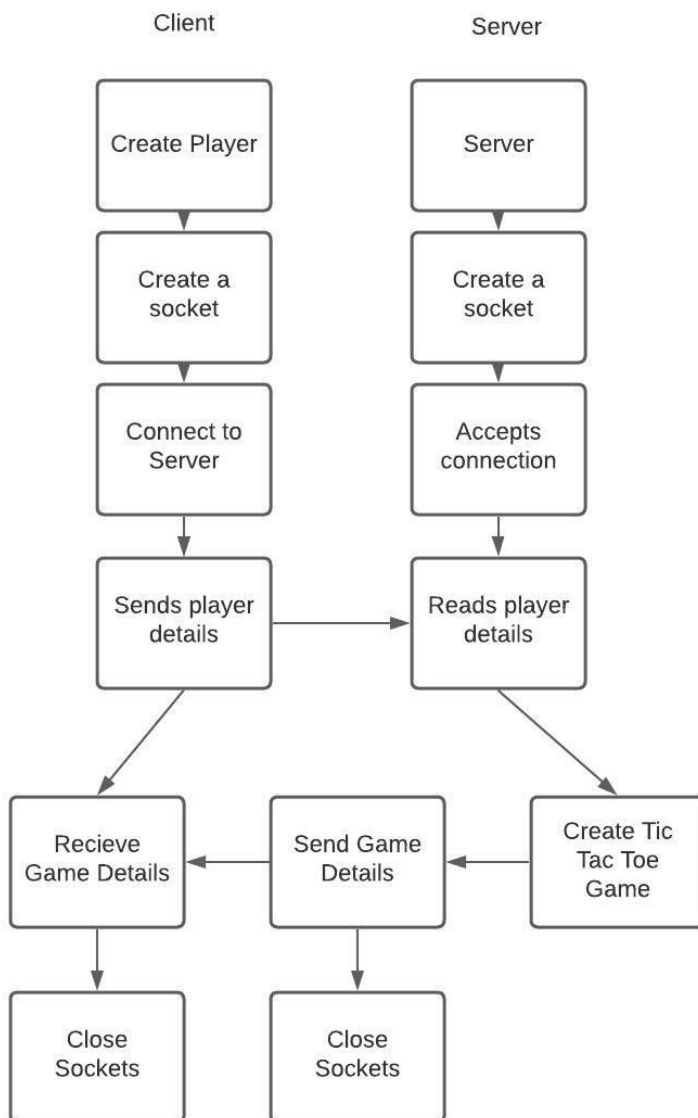
To: CSS 432, Winter 2021

From: Team Real Coders (Andrea Cseri, Jose Palomera)

Subject: *Progress Report 1*

## Work Completed:

### High-Level Architecture



## Features Implemented

### Client.java & Server.java

We have made a basic client and a multi-threaded server. They are able to connect and multiple clients can connect to the server. As of right now the client connects to the server and is then asked to login.

### TicTacToe.java

A significant feature that we have included is the base of the Tic Tac Toe game. This is meant to be its own class where it contains all the logic of the game, and can display the current view of the graph. The game is set up to contain two players where each player will have their own turn, and once a player has conducted a move, the game will be checked to see if there has been a winner determined.

### Player.java

Another feature is the Player class that is supposed to represent a single player along with their details. This includes, name, amount of games won, and this class is meant to be created once we are having a client connect to a server.

### Driver.java

One final feature is the Driver class, which at the moment holds the beginning of starting a tic-tac-toe game. Our idea with this class, would be to merge it along with the Client and Server classes since here we have the interaction between both players before the game has started.

## Files Included On GitHub at the moment

AndreaC1 Add files via upload		b10a2f9 1 hour ago History
..		
Client.java	Add files via upload	1 hour ago
Driver.java	Add files via upload	2 hours ago
Player.java	Add files via upload	2 hours ago
Server.java	Add files via upload	1 hour ago
TicTacToe.java	Add files via upload	2 hours ago

## Problems Encountered:

Our major problem is that our third group member is not communicating with us at this point and has not contributed to the project. Some actions to consider would be to drop

the group member and work with the team of two, and we are also thinking about emailing Professor Peng to let him know of the situation.

Our other problem is figuring out how to implement the game and player classes into the client server classes.

### Work Scheduled:

1. Implement the necessary components to transfer player information between client and server.
2. Have the server create a new Tic Tac Toe game, and return it to the clients.
3. Make sure that we can play the game with at least another player.

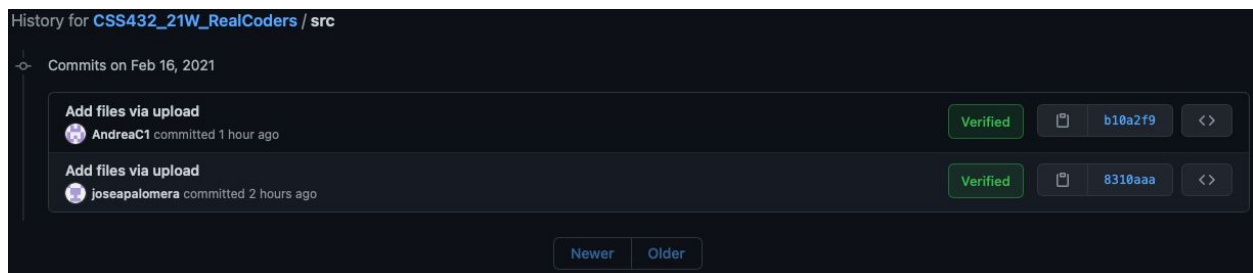
### Evaluation:

Chris: Has not contributed

Andrea: contributed 2 files (Client.java, Server.java)

Jose: contributed 3 files (Player.java, TicTacToe.java, Driver.java)

### Screenshots of Submissions (Andrea, and Jose):



### Screenshot of Andrea's code submission:

Add files via upload

main

Browse files

AndreaC1 committed 6 minutes ago

Verified

1 parent 8310aaa    commit b10a2f9a6de2a56f6069b4360a40e860c3a70c2a

Showing 2 changed files with 101 additions and 0 deletions.

Unified

Split

37 src/Client.java

@@ -0,0 +1,37 @@

1 + import java.net.\*;

2 + import java.io.\*;

3 + import java.util.\*;

4 +

5 +

6 + //used https://www.geeksforgeeks.org/multithreaded-servers-in-java/

7 + public class Client {

8 +    public static void main(String[] args) throws IOException

9 +    {

10 +        Socket cSocket = new Socket("localhost", 8081);

11 +        System.out.println("Please Login");

Screenshot of Jose’s code submission:

Add files via upload

main

Browse files

joseapalomera committed 35 minutes ago

Verified

0 parents    commit 8310aaa6fac88c99fe98d5cf40c74f08c55a3c55

Showing 3 changed files with 587 additions and 0 deletions.

Unified

Split

99 src/Driver.java

@@ -0,0 +1,99 @@

1 + import java.util.InputMismatchException;

2 + import java.util.Scanner;

3 + //import java.io.\*;

4 + /\*\*

5 + \* This program is a driver for the Tic Tac Toe class.

6 + \* This driver is used to play tic tac toe.

7 + \* This program assumes that the user has the tic tac toe class and the player class.

8 + \*

9 + \*/

10 + public class Driver {

11 +

12 +    public static void main(String[] args) {

13 +

14 +        Scanner play = new Scanner(System.in);

15 +        boolean playAgain = false;

16 +

17 +        //Welcomes the players to the game and asks for their names

18 +        System.out.println("Welcome to tic-tac-toe!");

19 +

20 +        System.out.println("Enter player one's name...");

21 +        Player userOne = new Player(play.next());

22 +

23 +        System.out.println("Enter player two's name...");

24 +        Player userTwo = new Player(play.next());

25 +

26 +        //Do while loop to make sure that the players enter a valid