Names: Jake Tusa, Michael Cheung, Billy Ko

Date: 5/6/17

CSE 310 Final Project Server Documentation

========================**USER DOCUMENTATION**=======================

**OBJECTIVE OF REVERSE TIC-TAC-TOE**

The game has two players and involves a 3x3 board. One player will use ‘X’ pieces, while the other uses ‘O’ pieces. The two players will alternate and place their pieces onto the board one at a time. The first player to place three pieces in a straight line loses the match. If there are no more available spots on the board but neither player has three pieces in a straight line, then the match is a draw.

**SETTING UP A SERVER**

**INSTRUCTIONS**

1. Run server.py in a command line

**RUNNING THE CLIENT**

**COMMANDS**

help this command takes no argument. It prints a list of supported commands, which are ones in this list. For each command, it prints a brief description of the command function and the syntax of usage.

login this command takes one argument, your name. A player name is a userid that uniquely identifies a player. Your name is entered with this command and is sent to the server.

place this command issues a move. It takes one argument n, which is between 1 and 9 inclusive. It identifies a cell that the player chooses to occupy at this move.

exit the player exits the server. It takes no argument.

**INSTRUCTIONS**

1. Run client.py in a command line with two arguments:

1. The name of the machine running the server

2. The port number that the server is listening at

If the machine you are trying to connect to is not running the server or does not exist, or if the port number is incorrect, then you will receive an error message with the reason for the error.

2. After connecting to the client, use the “login” command followed by a username. If the username is not already taken, then it will become your username. Otherwise, the server will return an error message stating that the username is unavailable and the user must use the login command again. Using the “place” command at this stage will return an error message and nothing else will happen.

3. Once you have logged in, the client will then wait until the server has two players to begin a new match of reversed tic-tac-toe. Once the server finds two players, a new match of reverse tic-tac-toe will automatically begin, and your client will display the tic- tac-toe board as well as the opponent’s username.

At any time, you may use the command ‘exit’ to exit the server and log out, ending any active matches.

4. You will be notified via command line when it is your turn. During your turn, use the “place” command followed by the position on the board that you wish to place your piece on. If the position is already occupied, or the number you entered is invalid, you will receive an error message and must use the “place” command again to make your move. Using the “place” command when it is not your turn will return an error and nothing else will happen. After successfully making a move, the client will display the updated tic-tac-toe board and you must wait for the other player to make his/her move.

5. Once one player has 3 pieces in a row or if the board is fully occupied, the client will display whether you won, lost, or drew. You will then automatically wait for another match to begin.

6. To stop playing, use the “exit” command.

=======================**SYSTEM DOCUMENTATION**======================

**REVERSE TIC-TAC-TOE PROTOCOL**

**METHODS:**

**200 LOGIN**

The LOGIN method means an attempt to for a client to log into the server. The contents of the request includes an ID set by the user.

**211 PLACE**

The PLACE method means an attempt for a client to place a piece on the tick-tac-toe board. The contents of the request includes the position on the board in which the player wishes to place a piece on.

**212 EXIT**

The EXIT method means that a client has exited from the game.

**STATUS CODES:**

**200 OK**

The request has succeeded. The information returned with the response is dependent on the method used in the request, for example:

PLACE the status of the newly updated tick-tac-toe board

**213 WAIT**

The WAIT status code is used for two scenarios:

1. The login request is successful, but there are not enough players to begin a match. The client must wait for a match to begin.

2. The client is participating in a match, but it is the other player's turn to make a move. The client must wait for the other client to make a move before it can make one.

**214 START**

This indicates that there are enough players to begin a match, and a match has begun.

**REVERSE TIC-TAC-TOE SERVER**

The server is initiated by running server.py, which has 3 classes: a server class, a player class, and a game class.

**SERVER CLASS**

The server class contains global variables for each of the protocol methods and status codes, which can be found in section 1.1 and 1.2 in this documentation, as well as local variables:

numPlayers the number of players waiting for or in an active game

nameList a list of IDs of current players; used to check if an ID is available for login

player1 a Player object to represent the first player in a match

player2 a Player object to represent the second player in a match

game a Game object to represent status of the match

The server class runs an indefinite loop, accepting all connections and returning a “200 OK” protocol message to connected clients upon connecting. In each loop, the server waits for two clients to log into to the server.

Clients log into the server by sending a “210 LOGIN” protocol message, after which the server checks to see if the ID associated with the login message is available. If it is available, the server increments the number of players and instantiates a Player object with an ‘available’ status to represent the player. Otherwise, the server sends a “400 ERROR” protocol message back to the client. All other protocol messages during this time are rejected and the server sends a “400 ERROR” protocol message back to the client.

Once there are two concurrent clients connected, the server instantiates a Game object with an ‘active’ status, sets the status of both players to ‘busy’ and sends the opposite player’s ID to each of the two players.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Placeholder\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**PLAYER CLASS**

**REVERSE TIC-TAC-TOE CLIENT**

The client contains a global variable for each of the protocol methods and status codes, as well as two variables for the port and hostname:

PORT the port number that the server is listening at

HOST the hostname of the server

The client checks the argument count to determine if there is a valid number of arguments and returns an error if there isn’t.

The client then attempts to establish a connection with the given hostname and port number, returning an error if it failed to establish a connection.

If the client successfully establishes a connection, it runs an indefinite loop, taking in user input.

Each user input is checked to see if it matches one of the defined commands, executes if it is valid, and then waits for a response from the server

**======================TESTING DOCUMENTATION**======================