Names: Jake Tusa, Michael Cheung, Billy Ko

Date: 5/6/17

Purpose: CSE 310 Final Project Server Documentation

SYSTEM DOCUMENTATION

1 Reverse Tic-Tac-Toe Protocol

1.1 METHODS:

Method =

"210" : LOGIN

| "211" : PLACE

| "212" : EXIT

1.1.1 210 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.

1.1.2 211 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.

1.1.3 212 EXIT

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

1.2 STATUS CODES:

Status-Code =

"200" : OK

| "213" : WAIT

| "214" : START

1.2.1 200 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

1.2.2 213 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.

1.2.3 214 START

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

2 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.

2.1 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\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

2.2 Player Class

3 Reverse Tic-Tac-Toe Client