**Socket Programming**

Jacob Bourgoine

Partner: Jacob Richardson

GitHub:

April 20, 2019

**Background**

For this project, a client needed to be made that would ask for a string from the user, accept the string, and pass it on to a server program using. This server would then check the string provided by the client and determine whether the string was a palindrome (string is read the same backward as foreword). Then the server would send the result back to the client, and the client would display the result back to the user. All of this needed to be done using sockets.

**Introduction to Socket Programming**

Sockets are used for communicating between two different machines or two separate programs on one machine using Transmission Control Protocol, or TCP. In Java there are Socket and SocketServer classes that are used for creating sockets. In order to create a connection between server and client, the steps are as follows:

1. The server program instantiates a SocketServer object with the port number that it will be communicating on.
2. The SocketServer object invokes the accept() method and waits for a client to connect on the port that was given.
3. Once the server has been set up and waiting, the client will instantiate a Socket object with the name or IP address of the server and the port number.
4. The client socket will attempt to connect to the server and port number and if successful, then the server will return a new reference to a socket on the server that is connected to the client.

At this point, the server and the client will be able to communicate with each other using input and output streams.

**How Project Was Performed**

The first thing programmed in this project was the networking component. Once both the server and client were able to communicate with one another, we moved on to programming the client-side user interface, the processing of the string by the server, and finally the output to the user by the client. Some preliminary tests were performed to test basic functionality before the code was turned over for more extensive testing.

**How Program Works**

The user must open and start the server first, then the client. Both can have arguments that dictate what ports and addresses to use, or the user may use the default address “localhost” and port 1200. The server and client will connect using the steps mentioned above. The client requests a string that the user will enter to check for a palindrome, which is then sent to the server using the sendString() method. The server will accept the string and use the trimStr() method. This method changes the string to all lower-case characters and eliminates whitespace and other characters that are not numbers or letters. This string is then returned and used as an argument for the checkPalindrome() method, which uses a for loop to test if the characters are the same backward and forward. It will return true if yes, and false if no. The result will be sent back to the client, where the client will tell the user the response. The client will keep running until an empty line is entered and will close. The server will close when the process is terminated.

**Contributions**

For this project, most of the coding of both the server and client programs was done by me. Also, some preliminary testing to make sure that the program had basic functionality was also done by me.

**Conclusion**

We decided to leave out the optional requirement that the server be able to accept multiple clients. It would have required creating multiple threads and implementing runnable, which is not that difficult. We ran out of time to add that functionality and to properly test it and as of right now, our code works well without it.

The programming for this project was relatively straightforward. We have done projects like this before, just without the networking component. Therefore, once the networking aspect was programmed, the rest was simple.