

Group 40 Portfolio 1

Scott Mueller & Sam Eue

Our project is a Tic-Tac-Toe game with a messenger component and online play capability.

Contents

| | |
|------------------------|----------|
| 1 Overview..... | 2 |
|------------------------|----------|

2 New and Complex

| | |
|--------------|---|
| 2.1 New..... | 4 |
|--------------|---|

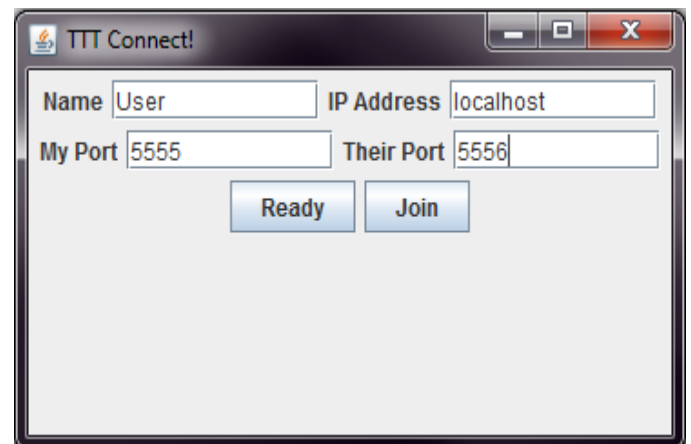
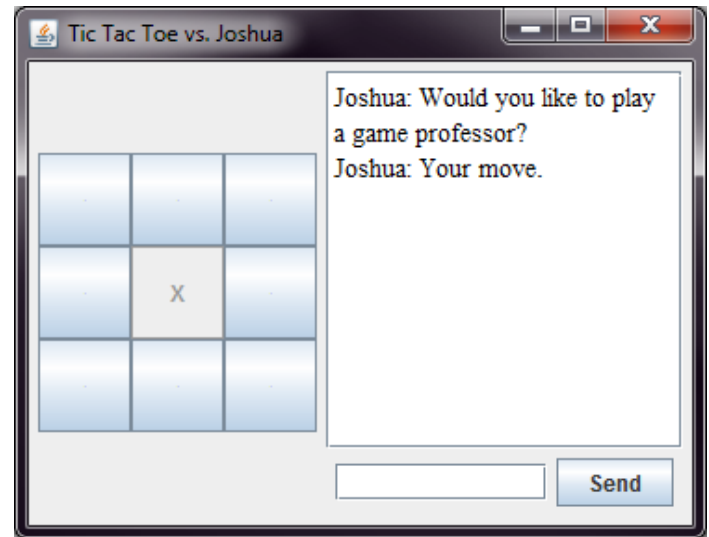
| | |
|------------------|---|
| 2.2 Complex..... | 5 |
|------------------|---|

| | |
|---------------------------------|----------|
| 3 Bloom's Taxonomy | 6 |
|---------------------------------|----------|

1 Overview

In our project we developed a tic-tac-toe game with a message client. The tic-tac-toe game is a JPanel with a grid layout containing nine buttons. The message client is comprised of a JTextPane to display messages and a JTextField to allow for input and a JButton to send text. While doing this we needed to deal with some challenges of disposing of old JFrame objects and creating new ones. When the game is first started, you are presented with a JFrame that takes some information. After we collected that information we needed to communicate some of the data over before starting the batch processing thread, then after handling initial communication we wanted to dispose of this JFrame and create a new one of the game using the gathered information from both clients. We also dealt with issues sending and receiving data between clients and servers. The information we collected from the first JFrame needed to be sent to the other player before the game could begin. During game play, we needed to send information between players about what button was clicked and send any messages the players had for each other through.

We analyzed different parts of Java swing to determine what some of the different components are. There are many ways to do things such as displaying text using Java Swing for



instance. We needed to determine which ones were best for our purposes. We evaluated each options supported methods to determine which to use. Eventually we created a game using what we found.

2 New and Complex

2.1 New

We did a few new things with this project that were not covered in class. We use threads to handle sending data between players and applications.

We used a JButtons to represent a game board. The buttons are all assigned an id based on their location on the board. When a player presses one it is disabled and marked with the player's mark. After each button is pressed, the game performs a check to see if anyone has won.

The game also has a message client. The message client uses a JTextPane to display the messages. The JTextPane uses a JScrollPane so that the players can scroll through the previously sent messages. The text to display in this area is entered through a JTextField with a JButton to read the data from the field and display it in the JTextPane.

For communication we used several threads to listen for data and parse the data out to the various functions it is used in. We ended up using a data structure similar to a queue to store the packets of data yet to be processed which was accessed by the processing thread so that we could be sure that all packets were handled properly.

2.2 Complex

Creating the game window used several different JPanels. The JButton field is a JPanel with a grid layout. The message area is a JPanel with a box layout along the y-axis and the elements in it are a JTextPane and a JTextField. These two JPanels were then added to the main JPanel of the JFrame to create the window.

The logic for figuring out who won is rather complex. The buttons are stored in a Map and have an integer key of 0-8 based on their position on the board. The buttons are arranged 0-2 in the top row, 3-5 in the middle row, and 6-8 in the bottom row. The game needs to check if any of the buttons labeled 0, 1, 2, 3, or 6 are pressed, then if any buttons horizontally, vertically, or diagonally are pressed as well to determine a winner. It then disables all buttons and displays to each player if they won or lost.

As well setting up and communicating information was rather difficult with making sure that the data was formatted correctly to assuring that the client had a valid port to communicate to before it tried to send data. Having the data sent over and read posed an interesting challenge due to the need for multiple threads to share data with one-another, from receiving the data to parsing out the two components of the packet to distributing each component to a different part of the program for use.

3 Bloom's Taxonomy

Each part of the project required careful analysis of what we had learned in the course so far. We looked at what we had learned about Java Swing, threads, and model-view-controller. We analyzed all the stuff we learned about coding in general and coding in Java specifically. Drawing from our past experiences we drew parallels to what we were learning now. Java Swing is a large part of our project as it is used for all the graphics. Threads are used to help with data transfer with the two clients as well as in the background of the swing GUI. We found model-view-controller important to our project and broke the code into three separate classes. Each class works to handle a different function; collecting data, transmitting data, and gathering and displaying the data.

After analyzing what we learned in class, we evaluated everything to determine the best way to implement it. Upon evaluation we got an idea of what is the best course of action for the project. Each decision we made involved a lot of consideration to justify the decision. We decided to make the project three classes to allow it to be more easily read and understood. Dividing the code into three classes allowed each class to be specialized as to what the functions in it do. Java Swing is full of many interesting features. This allows for multiple ways to display different items. For instance, we found that to display message text we could use a JTextPane or a JTextArea. Each offered similar features as well as some that were unique to each object. Upon evaluation of each part of Java Swing we determined what was best to use.

The final part of Bloom's Taxonomy is creating. When we started creating, it gave us a chance to see how the decisions we had made would work when actually implemented. When creating we used the different components we evaluated to be the most useful. What we created is new and demonstrates an understanding of the material learned in the class. It uses Java

Swing to interact with the user, threads to handle data, and server and client threads to pass data between players. The creation was a collaboration between two students and helped to strengthen what we learned in the class.