**MVC**

**Summary:** MVC, or Model View Controller, is a pattern used to create modularity and structure in an application. In the MVC pattern, the view is the interface which formats data, the model is the actual data being presented to the user, and the controller is a liason between the view and the model. Every action by the user is carried out by the controller, who fetches whatever data the user requested from the model and hands it off to the view, who then displays it in a readable and organized way. The view and the model should never even know that either exists. They only know about the controller, and they only interface directly with the controller in order to get tasks done.

**Code from Internet:** When studying about MVC, I came across this tutorial, along with example code: <http://www.newthinktank.com/2013/02/mvc-java-tutorial/>. It explains using a simple calculator program how the MVC pattern is used.

**My Sample Code:** Android, by its very nature, is designed to utilize the Model-View-Controller pattern. That being said, I created an Android project in order to demonstrate how MVC works. I have a Java Server which collects Pokemon data and then waits for client connections. The data is stored in a HashMap, which – in this case – is our **model**. In android, the **view** is created using an XML document to tell the device how and where the data is supposed to be presented. The Java backend of an Android app is the **controller**. It communicates with the model in order to retrieve the data the user wants, then simply passes that data along to the view. It doesn’t do any formatting or placement of the data; that’s the view’s job. In this case the controller is a demonstration of the Application controller Pattern.

Throughout this app, QCJSON, Collections, HTTP, Socket I/O, the Application Controller Pattern, and Parallel Processing are all used in conjunction with MVC in order to make the application function properly.

The following code samples demonstrate the MVC Pattern:

* Controller: <https://github.com/Lundberg-Jonathan/MVC/tree/master/app/src/main/java/com/jdlundberg/pokemoncollection>
  + This code sample shows the Application Controller Pattern, which allows the user to add Pokemon to his/her pokemon collection (AddPokemonHandler.java) or remove pokemon from his/her pokemon collection (RemovePokemonHandler.java). This is essentially the exact same as the Application Controller Pattern sample code.
  + The controller also reaches out to the server (model) on startup in order to obtain a list of Pokemon (GetPokemonInfo.java class).
* Model: <https://github.com/Lundberg-Jonathan/Multithread-Server/tree/master/MultithreadServer/src/multithreadserver>
  + The model reaches out to the Internet using HTTP communication in order to retrieve all the data needed for the clients. It stores the data in an ArrayList and sends that ArrayList through the socket to the controller when the client connects.
* View: <https://github.com/Lundberg-Jonathan/MVC/blob/master/app/src/main/res/layout/activity_main.xml>
  + This XML document tells the Android device how the data is supposed to be presented to the user.

**Sharing Video:** <https://www.youtube.com/watch?v=xMA-drai0M8>

**Group Meeting Teaching:** <https://youtu.be/P21CvG6x74Q?t=18m> (ends at 21:05)