David Wagner

CS341

Design Decisions and Difficulties

My project went through several changes throughout the course of my completing the final version. My first goal was to quickly build a menu console driven program to ensure that I have something if I was unable to complete the GUI using JavaFX. Thankfully I was able to reach my goal of developing a fully functional GUI application to search and filter a list of nearly 5000 movies and television shows that are currently available on the streaming service Netflix. I originally wanted to include more than just one streaming service in my application, however, acquiring the full libraries of other services like Amazon Prime Video and Hulu proved to be near impossible. In fact, it was only yesterday that my wget scrape abruptly failed, most likely due to being booted from the server due to excessive traffic. I managed to grab 200gb of information that I will be able to later parse and potentially add to my library of media. In the beginning I feared I may not even be able to grab the Netflix library, due to them removing their public API so I had to find a different way to the honeypot of content.

I found a site called “whats-on-netflix.com” and used a wget –r in my command terminal to download the content of the page. While browsing the index.htm I discovered a warning message saying it couldn’t connect to the data due to an API being used underneath the page source called Datatables. After a little research and poking around I discovered that when I did a search on the webpages source for the “Datatable” data type and a URL appeared underneath the word I was searching for that appeared to be a link to directly download a JSON file containing the entire library of movies on Netflix. Coincidently when I repeated the process on the page source for the full library of TV shows I found the same type of URL for a JSON containing all the TV shows on Netflix. This was by far one of the more difficult tasks during the completion of my project. Now that I have the JSON files, I was able to apply methods I learned from doing the Dictionary homework assignment to read the JSON file in as a text file into my IDE as a String. From there it was a difficult road to parse each element into its’ respective variable, however, luckily JSON files have a very specific format. The data for each movie or tv show lived between two curly braces { }, so parsing the data was a lot like peeling an onion. I chose to use a List because the size of the list was not large enough to need to use a map.

I chose to utilize two interfaces one called MediaType and one call Person. The MediaType interface would be implemented by the classes TelevisionShow and Movie while the Person object would be implemented by the classes Actor and Director. For the GUI I landed on a simple idea to use the TableView element of the JavaFX protocol to create a column for each variable in my MediaType object. I was able to write a single method using nested “if” statements which would allow the user to filter the results of the list in real time using 5 TextField boxes. Each TextField represented a listener for a different column, Title, Actor, Director, Genre, and Release Date. The method I wrote allows the user to filter using multiple tables together, for example if I type “rock” in the title TextField and “Stall” in the Actor TextField the table will only show the movies Rocky with Stallion as the Actor.