Team 5

Phase 2 Documentation

Jacob Kickbush, Adam Davis, Kyle Emery

Phase 1 Repo/Documentation State:

The documentation we were given was very unorganized and somewhat short but they explained the process they went through when creating their files. Most of what they did was not put into methods, and they did the majority of the work in their main file, which is normally only used for making the original GUI window or only for running the console interface. Working off of their documentation, I understand where the problems were for their repository.

First off, the searchBy function they used was incorrect, they simply printed the single variable for every single earthquake, when it is supposed to a # of arguments and find earthquakes that match those arguments. I had to completely rewrite what they had for this and use my previous code from our phase1 repo in order to try to replace it and make it work.

Second, their earthquake class had every variable as a string even for the numbers and doubles, so it was some extra work in order to parse our data types every single time we called the getters or the earthquake object. They also did not account for any earthquake data that was null, or had nothing inside of it. I had to add some code that checked every variable for a null and filled it with some sort of blank message so that none of the code would break.

Third, their main file was an absolute mess. They made the earthquakes inside of it, then they had a giant while loop that had the possibilty for everything the user could type in the interface and then run the code right there. None of this code was in separate methods and they all existed in the main file, so needless to say this needed to be restructured. The majority of the code itself was fine, but having no methods made it impossible for us to translate this into a GUI and have the controller call the methods. Copy and pasting the code into our controller can be disastrous compared to simply putting it in a method and calling it from our controller.

Lastly, the earthquake collection class is where all of these methods were supposed to be, but they never had an earthquake collection so that is why all of the code was simply in their main class. Since there was nothing in it originally we simply scrapped the earthquakecollection file for phase 2.

Project Breakdown / Architecture

Our first order of business was restructuring the current repo we had in order to use it in a GUI, along with getting started on the GmapsFX so we would hopefully have both parts ready in time for each other. This worked out pretty well, Jake took care of restructuring all of the code into methods and fixing the functions that were not working correctly. Adam and Kyle were working on getting the GmapsFX window working and our GUI set up so that we could combine the two together.

To restructure the code, we took their main class that held all of the code and took our everything except for the code to make our GUI window. This is all that should be in the main window and we moved the rest of it into a new class called Console, which represents the code from the phase 1 console interface. In order to not change the architecture more than was necessary, we left all of our methods in the Console file rather than making a new earthquakeCollection class for all of our methods. We left the console while loop commented out so that it would still be possible to use if needed. We made methods for all of the searchBy functions and a method for creating our ArrayList since these are what we would need for the GUI. For the earthquake class, we had to edit their toString method since it did not show the information effectively for testing purposes. The rest of this class was left untouched.

For the phase 2 breakdown and architecture, we had our main file become what opens our GUI, and then we had the controller for this main file called maincontroller. This is what makes our GUI that shows the Gmaps window and holds all the functions to search with. Basically the start of the program uses Main, and then the MainController talks with the Console class back and forth in order to run the searches. We had to have a popup window in order for the user to save the earthquake data points that they wanted, so we made another popup as well that would be a help box in order to limit the text being shown on our main window and allow the gmaps window to be the centerpoint. This help window was called from maincontroller and had its own helpwindowcontroller that simply displayed the help text for the user, along with a close button to close the window.

To finish, we had a saveWindow along with it’s controller as well. This save window acquired the maincontroller in order to write to a file of the user’s naming choice that held the points or point that they wanted to print out. For a visual representation of the architecture, look at the picture below that shows all of our files.

Unresolved Issues/ Bugs

There were some issues that we simply could not resolve because of the architecture used during phase 1 of the group whose repo we were given. We did our best to fix and use this code in order to make our GUI work the best it possibly could. In our actual GUI we know of only one current bug that we have been unable to fix. When you enter a word such as “apple” for the min or max for a variable like “Latitude” then it will say that there are 0 earthquakes shown, but some markers will still appear on the map. The markers are not the full set of earthquakes, and we have been unable to figure out where they are coming from, but it does not crash the code or the window.

