Kelly Brennan and Celine Ta
Software Design – Spring 2015 Section 1
Professors Paul Ruvolo and Ben Hill
March 12th, 2015

Interactive Visualization Overview

Project Overview:

Our project presents the ratings of individual happiness, security, enjoyment of music, time out doors and frequency of smiling of each state, in addition to the state GDP, in the mainland states of the United States as an interactive visualization. The visualization takes the form of a map, displaying information when the user hovers over a given state. A home page houses links to all of the maps to allow users to be directed to the map containing the data that they are interested in.

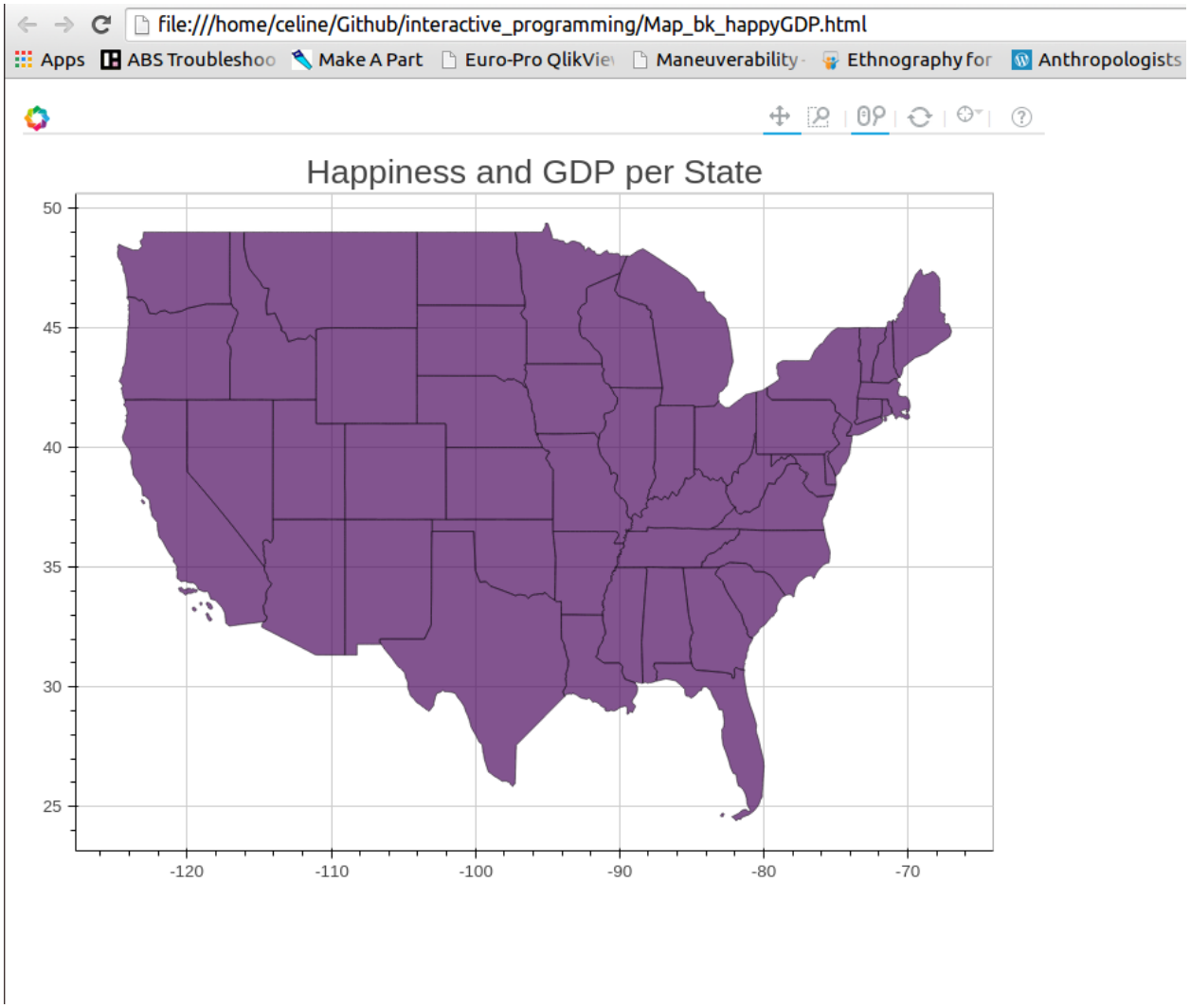
Results:

We have an HTML website that displays the different map options and allows the user to select between maps displaying data of interest.

Homepage:

Clicking on the map options redirects the user to the website that displays the map.

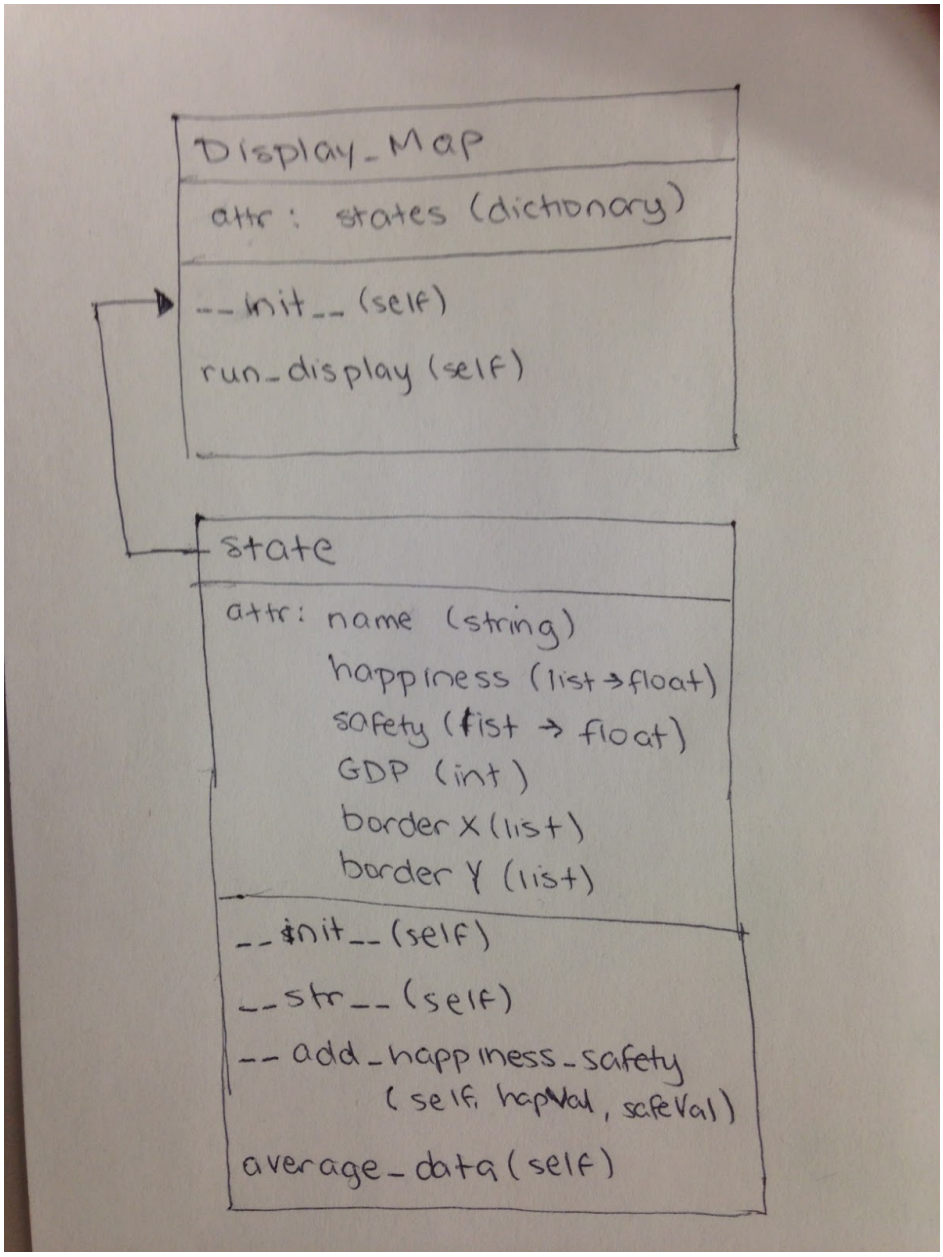
Example of a Map:



We use Bokeh's hover and tooltips functionalities to display specific data about states when the user hovers over a state. In the above graph, we display the state name, the happiness level reported by residents there (on a 1-5 scale), and contribution to GDP. Additional attributes are easily added to the information architecture by incorporating them into the State constructor and instantiation.

We successfully implemented object oriented programming for processing the data and displaying the graphs to address our learning goals of practicing object-oriented programming, even though our graphs may be more efficiently graphed without it.

Implementation:



Our implementation uses two classes, Display_Map and State. Display_Map reads data into the program, oversees State objects, and displays the figure. The State class stores information specific to each State. The State class's main function is to store and process data by state. Considering that the data is imported or read into the program in neatly organized dictionaries, our implementation did not need more than one class. In fact, it may have been more efficient to eliminate the State class for the scope of this visualization. That said, using a second State class had two benefits. First, it allowed us to practice working with custom objects in object-oriented programming. Second, it allows for more flexibility when restructuring. For example, we considered and attempted to make this program into app form. By using State objects, it was easy to get state data from various classes and made the transition easier. While Bokeh's widget and app foundations are currently unstable and undocumented, and we were unable to complete the transition, having a State class makes future work more feasible.

We gathered state boundary, individual ratings and GDP data to incorporate into the maps from a variety of sources. We adapted the state boundary data from the Bokeh sample data in a CSV file called US_Regions_State_Boundaries.csv. The US centric data from <insert source> is based on how individuals living in specific states answer questions on a scale from 1 to 5, where 5 is very high/frequently and 1 is very low/infrequent. The questions answered were: “How often do you try to make other people happy?”, “Are you surroundings physically safe?”, “How often do you listen to music you enjoy?”, “How often are you outdoors?”, “How often do you smile every day?”. We incorporated participants answers into the code for each rating by averaging their answers for each state. Unfortunately, this questionnaire did not reach participants from all 48 states included in our map, which means that not all of the states include this data. Finally, we obtained the GDP data from the US Bureau of Economic Analysis (BEA). Since the data we incorporated into the code came from three different sources, we had to integrate data together in the code by using a dictionary.

We decided to organize our visualizations using an HTML landing page due to limits with Bokeh’s app function. In our development process, we generated our display maps to test the code using Bokeh’s HTML display. However when we converted the data and display to an app using bokeh, we had code without any running errors and a functional app domain, but our actual visualization would not display. After reaching out to our SoftDes NINJAs and fellow classmates with the same issue, as well as researching the problem, we were ultimately unsuccessful in implementing the app option in developing a more ‘beautiful’ interactive display. In the early morning, this led us (and other teams) to settle on alternative options. We decided to use HTML to organize and display our graphs to maintain the user-interactions feature of our work. Despite our failure in using bokeh apps, we learned a lot by practicing data processing, object oriented-programming, and data visualization python skills.

Reflection:

Our project highlight was working with each other because we work well together. Even when we are really tired, we still make each other laugh :). Due to our process, the work was divided equally between us. To divide the work initially, we met to work on separate parts of the project together. As we began to integrate the program, we started working together but using one computer to prevent merge conflicts at a such a critical stage. Despite our attempt to reduce merge conflicts, we still faced a couple github issues (including a few merge conflicts). Even though this project helped us get more github experience, we could definitely improve on using it more efficiently and effectively for collaboratory use.

Although the project idea was the appropriately scoped, we were a little late to finalize our exact direction following our proposal feedback and later hit a significant limitation in the implementing bokeh apps that prevented us from achieving our objective. Unit testing was not the best way for us to test our code because we only use three functions, one of which includes the main elements of our visualization and two basic functions that converts the data from a list into an object and calculates the mean of individual ratings. Because of our unit testing limitations, we continually debugged and tested our code as we iterated our project code.

In the future, we wish we knew Bokeh’s interactive visualization limitation before beginning our project journey. Specifically, it’s structure does not lend itself well to object-oriented programming, which creating some tension between our learning goals and Bokeh’s functionality. Next time we would not use bokeh for this project to attain our learning goals of practicing object-oriented programming. Also, we would create a UML diagram earlier in our process to better define our classes and functions. This would have helped us develop

a better understanding of the program structure and would have saved us some time. This would help us better understand the structure of the program. Nonetheless, Although we wish to have developed a better understanding of object-oriented programming structure and functionality, we now know how to use Bokeh to develop visually appeal graphs in the future.

Data:

"Regional Data." U.S. Bureau of Economic Analysis (BEA). US Department of Commerce, 12 Dec. 2014. Web. http://www.bea.gov/iTable/iTable.cfmreqid=70&step=1&isuri=1&acrdn=1#reqid=70&step=1&isuri=1

Tuva Labs. (2015). Global Happiness Project – Survey Data [Data file]. Retrieved from https://tuvalabs.com/datasets/csv/global_happiness_project__survey_data/