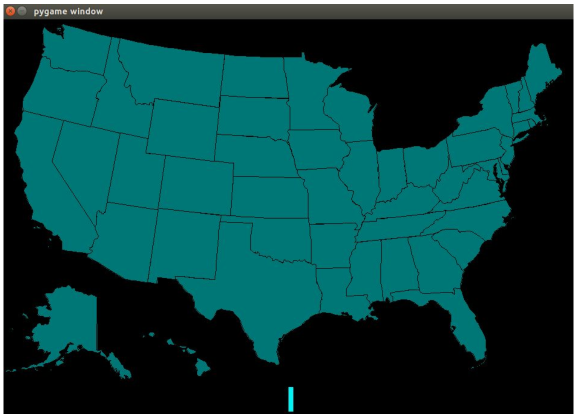
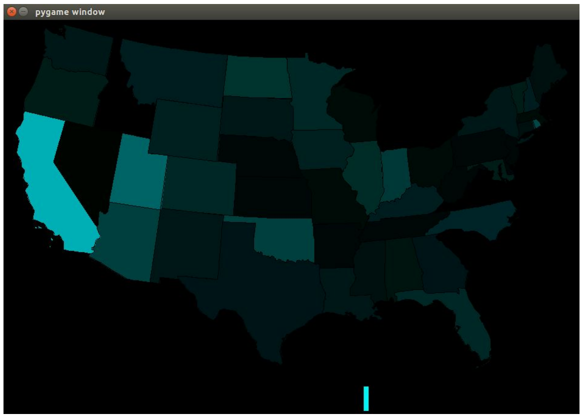
Software Design Section 2 Celina Bekins, Cecilia Diehl
MP - 4

Project Overview:

This interactive map of the United States was built to help draw attention to our growing air pollution issue. Taking data from from the Environmental Protection Agency for the year 2015, this map visualizes ozone concentrations by varying the color of each state depending upon the concentration of ozone in the air on a specific day. The user slides a bar at the bottom of the map to determine which point in 2015 they want to view. By sliding the bar around the user witnesses the changes in ozone throughout the United States that can occur in as little as year.

Results:



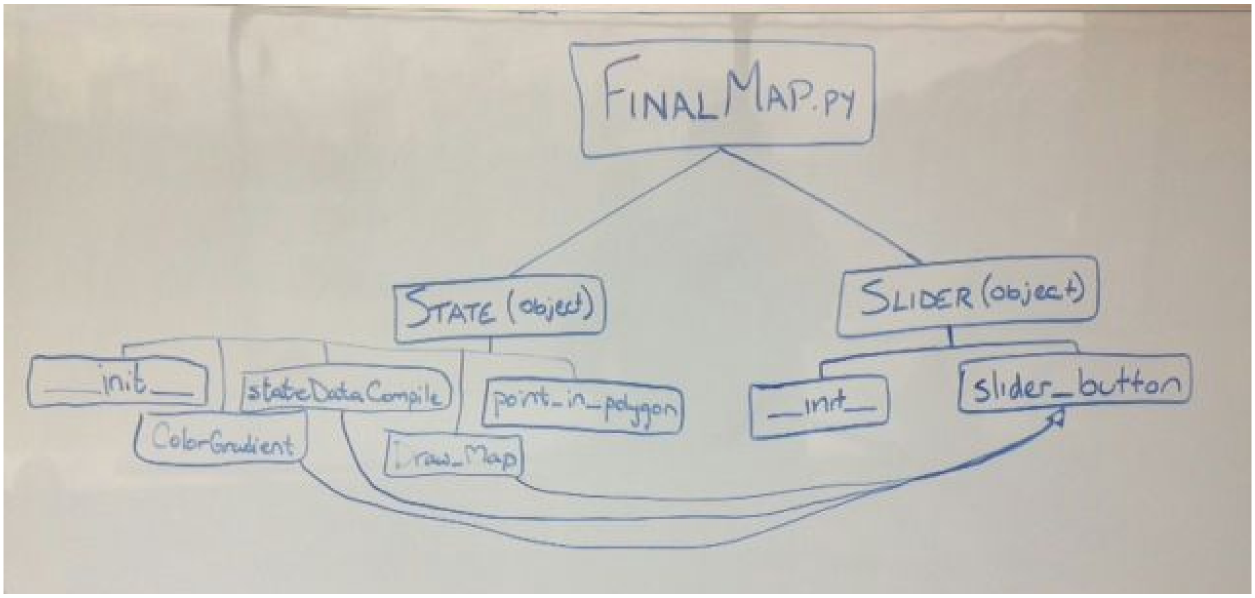


We set out with the goal of visualizing air pollution data over time across the country, and we accomplished that goal. Our code creates a map of the United States and a slider on the bottom of the screen. The slider moves through time; sliding to the left brings the date closer to January 1, 2015, and sliding to the right brings the date closer to December 31, 2015. Then, our code uses the date to find the ozone concentration on that date for each state, and relates the ozone concentration to a color gradient. Finally, that color gradient appears on the map of the US. The ozone concentration and associated color are specific to each of the 50 states.

The above images show the initial map when the window first opens as well as the map after the slider has been moved to the left. As shown, the colors of the states vary, the darker the color the higher the concentration of ozone, the lighter the color the lower the concentration of ozone.

Implementation:

Our final script includes two classes: a Slider class and a State class. These classes are what we use to generate every aspect of the images we have on the screen. Within the Slider class, we have a slider_button function which allows the user to move the slider by clicking on it and dragging it across the screen. The Slider.slider_button() function also uses our other class, State, to alter the map based on slider position (redrawing each state with the correct color each time the slider moves). This function was inspired by code that we researched on StackOverflow while trying to understand how we could most effectively make a slider work. Within the State class, we have functions which establish the shape and ozone concentration for each state. Ozone concentration was gathered from data that we found on the US EPA website, and the State.stateDataCompile() function we have figures out what day it is (based on the slider's position) and associates an ozone concentration with that day. This concentration is then related to color, giving each state an appropriate color within a blue-black gradient. Finally, the State.DrawMap() function generates an image of each state, using the borders and color predetermined by the other class functions.



During this project we knew that we would eventually end up with a complete map of the US, however we didn't start out with one. Using the US map generating code from Oliver we decided to first focus on getting one state to work. We chose Texas because it was large enough that we could easily see if things like the color gradient were working. By starting out with only one state we didn't have to worry about all of the lists and loops that we later incorporated to map the rest of the states.

One main decision we made was to scale back a little on the data we used in our map. Originally we were planning on taking data for around ten years and incorporating that. For sanity's sake we limited it to just 2015.

Another decision we made in the end was to not stress about the speed at which our code runs. However, we are conscious of the fact that our code moves a little slowly so we spend time think of ways we could change our code to make it run faster and smoother. If we were to expand upon this project we would spend a bit more time and speed up the running time.

Reflection:

Originally when we went into this project we knew that scheduling would be a bit challenging, so we decided to divide and conquer. This included first splitting the project up into map generation and data input. Unfortunately, we both came up against some difficulties that we needed help with. This resulted in a a lot more pair programming than we had originally intended, but it worked pretty well. While it did mean that we moved a bit slower on the overall project, it was much less stressful and we spent less time trying to fix one problem. If we came up against a problem one of us couldn't figure out the other might already know that answer and visa-versa.

As far as project scope I think we were right on track with our abilities and the amount of time we originally had. It was a project that, with some help, we could figure out. However, knowing that we had enough time to complete the project and using our time efficiently are two different things. During the time that we were still trying to work separately we did not get nearly enough done on the project in the amount of time we used. It was not until we started pair programming that we made some real headway. This will be good to remember for future projects where dividing and conquering is not always the best use of time.