

California State University Monterey Bay

Mike Menendez

Edgaras Slezas

Felix Ramon Romero-Flores

Frank Piva

Wesley Modes

CST 205: Multimedia Design and Programming

OOP Problem-Solving Approach to Project Design

04/29/2020

1. Two things we learned from this article is to understand the problem and how to pseudocode the high level approach. The first concept helped us with the layout of our application because we didn't know where certain UI elements like the graphs were going to live. Once we started describing the problem in detail, it helped us define the specifications required to make graphs possible. We then arrived at a solution of placing them on the country page. Describing the high level approach in pseudocode helped us iterate quickly while collaborating. This is a small project with a short timeline, so we don't have cycles for creating detailed designed documents. As such, we need to be able to verbally describe the solution so that someone else could start working on the solution.
2. The main class we have is the Data class. It is responsible for taking a payload of data and creating a pandas DataFrame. It has four visualization methods that take in a dataframe and perform some matplotlib operations to render graphs or plots. If we have time, another class we might introduce is a Location class for our heat map. Each location is responsible for its coordinates and size. It would likely need setter methods to update the size of the location point. There may be other map related classes that we import from the Google Maps API.
3. The Approach
 - a. Part 1: Understanding the problem
 - i. Our problem is separating code to make it cleaner and easier to work with.
 - ii. Nothing in our project is really object oriented as it is a simple ReST API with a web wrapper.
 - iii. However to keep our clean, it would be a good idea to make an auxiliary grouping of functions so that they can be imported and bundled together. We can make this work as an object for the sake of this assignment
 - b. Part 2: High level pseudocode
 - i. Create dataframe
 - ii. Transform data
 - iii. Generate visualizations and export them to SVG files
 - c. Part 3:
 - i. Fetch data from remote API
 - ii. Convert and filter data as needed
 - iii. Pass data to object constructor
 - iv. Transform data to dataframe
 - v. Visualize data to external files for export
 - d. Part 4: Pythonic Pseudocode
 - i. To bundle our code as a grouping, we can initialize the data in the constructor and have a bunch of included methods that are responsible for transforming the data
 - ii. Class Data:
Init (data):
init the data to a data frame

```
Async def hist(keys):  
    # build histogram using the keys to access the data frame  
Async def viz2(keys):  
    # build a second visualization based on provided keys  
Async def viz3(keys):  
    # build a third visualization based on provided keys  
Async def viz4(keys):  
    # build a fourth visualization based on the provided keys  
Async def reg(keys):  
    # perform logistic regression to demonstrate growth rates  
Async def arma(keys):  
    # perform time series analysis of the the data in hopes of  
    # building a prediction with moderate accuracy (doubtful  
    # as this is predicting the future) (sigmoid)
```