

# Programming Assignment 3

*Due on March 4, 11:59pm.*

This programming assignment assumes that you have fulfilled all course prerequisites and followed along with the tooling videos and readings.

## Template and Submission

We will use **git** repositories for all programming assignments and project milestones. You should already be well acquainted with the process and all necessary git commands from the previous programming assignments.

We created a git repository in your github student account (<https://github.students.cs.ubc.ca/>) that contains a template for the first task that should help you get started. The datasets are included in the /data folder. You will need to modify the .html, .js, and .css files as described below. You will need to add new files for task 2 (vis class, data).

Submit the programming assignment by updating the given repository ( `git push` ). You can commit/push changes as often as you want before the deadline. Work alone.

We will also evaluate code readability and structure (add comments, indent code, use functions, ...).

Cite any external resources and explain exactly what modifications you have made in your README documentation. We created an empty `README.md` file in the root directory of the repo.

## Task 1: Animated Choropleth Map (2-3 hrs)

1. **Familiarize yourself with the given template** ( `task1.html` , `task1.js` , `choroplethMap.js` ) and the datasets in the /data folder.
  - When you start a local webserver and open `task1.html` , you should see a vector map of Canada.
  - `canada_historical_population.csv` contains population counts for all Canadian provinces since 1991.
  - `canada_provinces.topo.json` contains a TopoJson file that you should use to draw a geographic map of Canada.
  - The datasets are loaded in the `task1.js` file and we convert all values of the CSV file to

numbers.

- A rough structure of the D3 visualization is provided in the `choroplethMap.js` file. You can also have a look at the implementation of other D3 choropleth maps such as:
  - [https://www.d3-graph-gallery.com/graph/choropleth\\_basic.html](https://www.d3-graph-gallery.com/graph/choropleth_basic.html)
  - <https://bl.ocks.org/JulienAssouline/1ae3480c5277e2eecd34b71515783d6f>
  - <http://bl.ocks.org/palewire/d2906de347a160f38bc0b7ca57721328>

**2. Select a specific year (e.g., 2019) and color code each province by its population.**

**3. Add labels for each province with the population value.**

- Position the labels in the center of each province path element.
- Use the enter-update-exit pattern (or `.join()` shortcut).
- Format the numbers so that they are easier to read (i.e., 10.5K instead of 10500 and 7.1M instead of 7100000). *Hint: `d3.format()`*

**4. Add an HTML slider to the interface that lets users interactively change the year selection.**

- Listen to slider changes, filter the data, and update the map.
- You will need the minimum and maximum population across all years for the color scale.

**5. Create an animation.**

- Instead of manually selecting a specific year, the visualization should loop through every year and update the colors and labels accordingly, similar to a short video. The population for each year is shown for a certain time.
- Add a control element for the user to play/pause the animation.

See example result: [https://www.students.cs.ubc.ca/~cs-436v/20Jan/images/P3\\_choropleth\\_animation.gif](https://www.students.cs.ubc.ca/~cs-436v/20Jan/images/P3_choropleth_animation.gif)

## Task 2: Unique Visualization (8 hrs)

- In this task you will take full advantage of the D3 skills that you have acquired over the last several weeks to create a unique visualization beyond commonly used chart types.
- The audience of your visualization are readers of an online news paper. Therefore, the goal is to communicate certain aspects of the data effectively and in a creative way instead of supporting the in-depth analysis of domain experts.
- Data: Soviet Space Dogs
  - Download data source: <https://airtable.com/universe/expG3z2CFykG1dZsp/soviet-space-dogs?>

[explore=true](#)

- The database contains two tables:
  - Dogs: 48 dogs who participated in the Soviet space program in the 1950s and 1960s.
  - Flights: 42 missions that the dogs flew on.
- Design space and requirements:
  - Create a unique D3 visualization that is not simply an existing D3 block. You may import existing D3 code, but as always, you must cite your source(s) and explain exactly what modifications you have made in your `README.md` documentation. You must make substantial changes to any existing code that you build upon. If you are unsure of what counts as substantial, discuss it with the TAs in the labs.
  - Visualize three different attributes. At least one quantitative (may be temporal) attribute, in addition to categorical attributes.
  - You don't need to visualize all the given data! Focus on a specific topic or question that you want to address, which may only require looking at a subset of the attributes and/or items.
  - You may want to create custom glyphs to encode the information. These examples could give you some inspiration:
    - <http://giorgialupi.com/laettura>
    - <http://sxywu.com/filmflowers/>
    - <http://giorgialupi.com/visualizing-painters-lives>
    - <http://www.oecdbetterlifeindex.org/#/31111111111>
    - <https://www.scientificamerican.com/article/where-the-wild-bees-are/>
  - You can choose to create a static or interactive visualization.
  - Add a descriptive title and a short description.
  - Don't just use basic D3 blocks (*bar chart, scatter plot, pie/donut chart, scatter plot, line chart, area chart, stacked area/line chart, node-link diagram, bubble chart, simple vector map, simple tree diagram*) or the task will be graded with 0 points.