TEAM: FairyMander

Overview: The main purpose of the "Technical Demos" is to very clearly explore the extent to which the team has implemented the key functional and performance requirements for their project. Grading is based on two factors:

- **Completeness:** Have all of the key requirements been implemented. To what extent does the product have all of the functionalities and performance that was promised.
- **Quality:** Just having basic functionality is the bare minimum. What is the quality of the implementation? Is the resulting product aesthetically pleasing, easy to use, and a pleasure to work with; to what extent is it "ready for market"?

This template is fleshed out by the team, reviewed and approved by CS mentor beforehand, and then brought to demo in hardcopy for the mentor to use as a grading sheet.

Requirements Review
Based on our requirements acquisition work and evolution during implementation, the following are the key technical requirements driving of our product:

R1: District Generation Algorithm. An algorithm that, given a shapefile for a U.S. state, can generate a district plan. It must be able to generate state district plans that can be reasonably evaluated for fairness.

- `generator` module in python package contains a `DistrictGenerator` class that can be used to generate voting districts for a given state
- Generator returns a given number of maps that are generated for a given state, optimizing for a specific fairness metric (polsby-popper score or efficiency gap)

R2: Fairness Metrics. Utilities should be developed that can evaluate current districts as well as districts generated from R1 so that developers can generate and compare maps. These utilities will be used to determine the maps that will be presented on the FairyMander Website

- `fairness` module in python package contains utilities for evaluating a generated map, including individual metrics as well as a comparison option for two maps

R3: Interactive District Maps. Maps generated by R1, as well as maps of the current Districts, must be converted into interactive maps that can then be displayed on the FairyMander website.

- `folium_converter` module in the python package contains utilities for converting the district maps into interactable maps suitable for the website.

R4: Results Website. A website must be developed for presenting FairyMander's results. This website must have a Map of the U.S. where users can select each state and compare our results to the current state district plan.

- Described website is running, hosted on https://fairymander.info/.
- Website has a U.S. map detailing our results, comparing our generated district plans to the existing state district plans.

R5: Redistricting Education. The results website must be educational, informing users about the redistribution process. The presented information should be easy to understand and follow.

- State-specific redistricting laws are included for each state analysis page.
- A Glossary page contains key definitions to help users understand the content of each district analysis.

Demonstration Sequences:
This section outlines the demonstration sequences prepared to prove the above functionalities. Each sequence is a coherent walk-through of some piece/area of the product, designed to highlight implementation of specific requirements/functionalities outlined in the last section.

Demonstration Sequence 1: User generates a map for <state>

Requirements demonstrated: R1, R3

Flight Plan for this demo sequence: Step by step plan of things to do/demo

1. User selects a state for district plan generation (In our demo, this means the reviewer will select a state).
2. User initializes DistrictGenerator with initial params:
    a. Compact, deviation = 0.003, steps = 50
3. User calls `run` on DistrictGenerator, producing a district plan
4. User visualizes result map in Folium

Evaluation (filled in real-time by mentor):

ü Convincingly demo'd each of targeted requirements?

ü Quality, aesthetics and other evaluative comments:

Demonstration Sequence 2: User compares generated map to current map

Requirements demonstrated: R2

Flight Plan for this demo sequence: Step by step plan of things to do/demo

1. User loads a generated map from a shapefile
2. User performs fairness analysis on a single map
3. User loads the current map
4. User runs a comparison fairness analysis on the two maps, with the resulting analysis breaking down the comparison

Evaluation (filled in real-time by mentor):

ü Convincingly demo'd each of targeted requirements?

ü Quality, aesthetics and other evaluative comments:

Demonstration Sequence 3: Website Navigation and Demonstration

Requirements demonstrated: R2, R3, R4, R5

Flight Plan for this demo sequence: Step by step plan of things to do/demo

1. Landing page consisting of the navigation tabs, interactive United States map, and a hamburger dropdown menu.
2. Educational information on the redistricting process will be displayed upon scrolling down and a glossary page from the more navigation tab.
3. Upon selecting a state from the interactive map or dropdown menu, a side-by-side comparison is displayed of current congressional districts and fairymandered districts using folium.
4. Demographic distributions are displayed from a dropdown menu for all districts in a pie chart form.
5. Fairness metrics of both maps will be displayed in a side-by-side comparison.

Evaluation (filled in real-time by mentor):

ü Convincingly demo'd each of targeted requirements?

ü Quality, aesthetics and other evaluative comments:

Other challenges recognized by not addressed by demo:

Hawaii is not completed on the map. Since it is composed of islands, Hawaii is exceedingly hard to redistrict algorithmically when trying to form continuous districts. As a consequence, Hawaii has been left out of our results.