Criterion E - Evaluation

Success Criteria

1. **The cartogram should accurately reflect population data with the size of each country.**

I made sure the size of each country in the cartogram directly correlates to its population data. When the region is smaller than it should be, there is a pressure force moving outwards, and vice versa. This pressure is always there, so the relative sizes of the countries throughout time do reflect the population data well.

2. **The animation between years should be smooth and stable, with no visual glitches or abrupt changes.**

I implemented interpolation techniques to ensure that the transition between years is fluid. Where data was limited, the program makes a smooth estimation of the population in that given year. While most of the animation runs smoothly, there's a significant visual glitch as far as self-intersection and overlapping. There are no abrupt changes and the program changes gradually with an easing function inside the region class to slow changes down.

3. **Countries should remain recognizable in shape and position despite distortion.**

Despite the distortions due to population changes, I ensured that each country's general shape and position are preserved. I used a spring model to try and tether the masspoints to their original positions, and added constraints to the kinematics to prevent significant shape distortions. Despite my best efforts though, some countries are unrecognizable at times because of overlapping. This is clear in Appendix A, where my client is disappointed by this.

4. **The visualization should clearly show how populations have changed over time.**

I made sure that the animation clearly shows how populations change over time by using color coding and size scaling. Each region slowly grows and shrinks, so the continuation of each region throughout the simulation is clear. Also, while gradual, the size changes are very clear. To support that, the simulation supports moving backwards and forwards so each change is more obvious.

5. **The controls for interacting with the animation should be easy to understand and use.**

I designed the controls to be simple and intuitive. There are some debug options with the switches on the side, but for the practical purpose of being used in a lesson, this program is sufficient. There is a simple draggable slider at the top of the screen which clearly correlates with the progression of time.

6. **It should perform well in a web browser, running smoothly without crashing or slowing down.**

I optimized the cartogram for smooth performance, ensuring that it runs efficiently on virtually any browser. The program consistently runs above 70 frames per second and has no visible frame dropping.

Improvements

Beyond fixing the self-intersections, I would like to add zoom and pan controls to the cartogram to provide users with more flexibility when exploring the visualization. This would allow users to zoom in on specific countries or regions to see finer details of population changes or zoom out for a broader, global view. I could implement mouse wheel controls for zooming and dragging functionality for panning, making the interaction intuitive. This would be especially

useful for users who are interested in a more granular look at specific regions or who want to compare countries at different scales. Additionally, I could add keyboard shortcuts or touch gestures for mobile users to make the experience even more fluid across devices.

I would like to dive deeper into performance optimizations, especially for large datasets or when running the simulation on lower-performance devices. For example, I could simplify the country borders or reduce the resolution of the cartogram for less important regions, loading higher-resolution data only when zoomed in.

In hindsight, I could simplify much of the code significantly by using a softbody system where each region has a lattice of many points inside connected to one another by springs. This system would move points around solely through spring mechanics, by tuning the desired spring length between each point in the lattice. This would remove all the other force calculations and be significantly easier to expand on for people trying to adapt my code.