Seth Hanusik

Visual Interfaces Final Exam

1. Data example and sketch
   a. Data description
      i. X: Spatial
      ii. Y: Spatial
      iii. Shift: Categorical
      iv. Date: Temporal
      v. Age: Categorical
      vi. Primary Fur Color: Categorical
      vii. Running: Categorical
      viii. Chasing: Categorical
      ix. Climbing: Categorical
      x. Eating: Categorical
      xi. Foraging: Categorical
      xii. Approaches: Categorical
      xiii. Indifferent: Categorical
   xiv. Runs From: Categorical
   b. Sketch
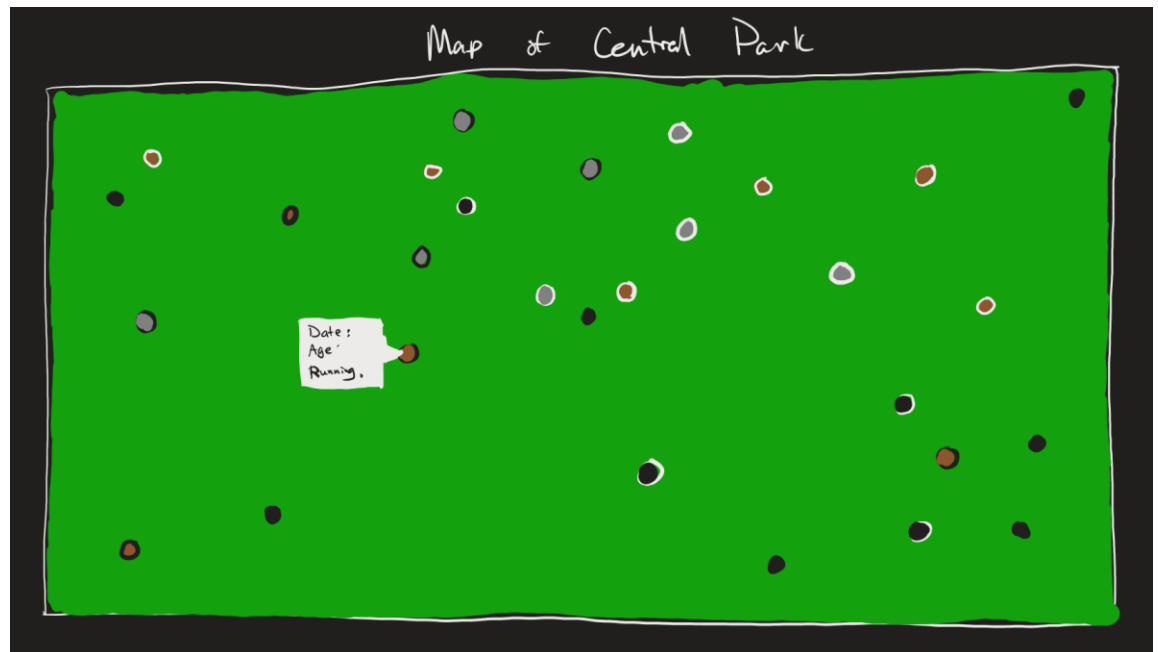
Each squirrel sighting is encoded as a circle on a map of Central Park using the Latitude and Longitude values to place them. The circle is colored based on the fur color of the squirrel: cinnamon, grey, and black. The outline of the circle represents the shift of when it was spotted: AM is white, PM is black. Then hovering over each circle displays a tooltip showing any number of other attributes desired, or interaction could be built in so a click could link out to a page displaying all information about the squirrel.

2. Critique and Redesign
   a. Critique
      i. The data is encoded in a 3D axis space with the x-axis representing the country the bananas originated from, the y-axis representing the number of bananas exported, and the z-axis representing the year. The bars are color coded to represent the year as well as the z-axis.
      ii. Keeping the years separate colors, allow for easier differentiation of the bars in the 3D space. Someone would be able to pick out the number of bananas exported from Ecuador in any of the years, and I mention Ecuador specifically because other countries it becomes extremely difficult to match up the height of the bar with the line marker.
      iii. For starters, the 3D space causes some data to get covered up by other bars in front of them, and the 3D space causes difficulty in placing the bars in their specific locations. The background image of the bananas causes reading the lines of the y-axis difficult. The 3D space causes it nearly impossible to determine the height of bars that are not right up against the background lines.
   b. Redesign Sketch
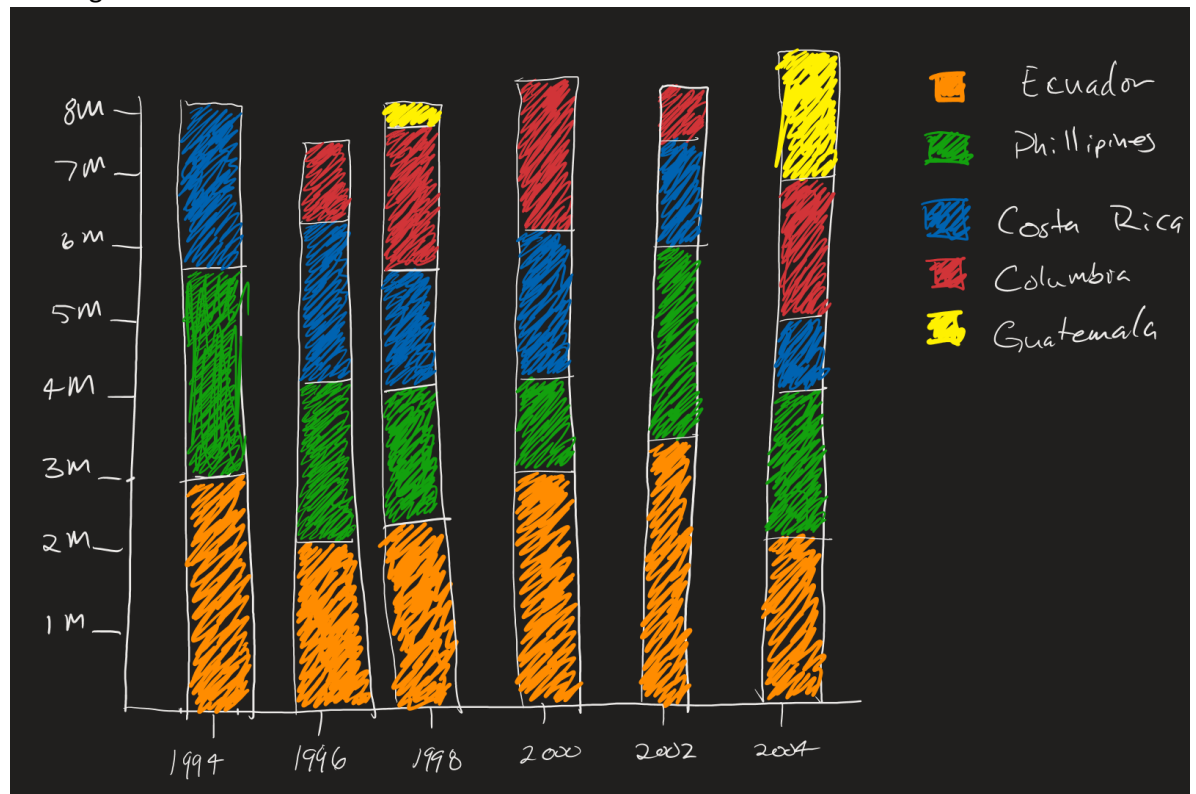
I chose to redesign it as a stacked bar chart. The y-axis still represents the number of bananas exported, and the years are placed on the x-axis, and each bar of the stack represents a country. I chose to use years on the x-axis and stack the countries because Ecuador has such a higher export number than other countries, the chart would have such a large y domain, the bars of other countries would be extremely small; therefore, using countries, the total exports across years will be close enough in height that it will

be valuable. You could also add the ability to switch between the actual number of bananas being exported, and the percentage out of 100% that each country exports.

3. Pseudo-code for creating a visualization
    a. Format the data as an array of objects as

```
{
    countryName,
    region,
    lifeExpency,
    income,
    population
}
```

    b. Initialize scales
        i. Linear scales for the x and y axes
            1. X Domain = [10, 90]
            2. X Range = [svg.height, 0]
            3. Y Domain = [0, 128,000]
            4. Y Range = [0, svg.width]
        ii. Linear scale for the radius of the circle
            1. Domain = [minIncome, maxIncome]
            2. Range = [10, 100]
        iii. Color scale for the region
    c. Select all circles in the SVG.
    d. Set the data array as the data for the all selected circles
    e. Join or append the circles to the SVG
    f. Set the X attribute of the circle by passing the income value to the x scale
    g. Set the y attribute of the circle by passing the life expectancy to the y scale
    h. Set the radius attribute of the circle by passing the population to the radius scale
    i. Set the fill attribute of the circle by passing the region to the color scale

4. I would recommend covering some topic on CSS and covering how to approach positioning numerous visualizations together, and the best approaches. The more existing visualizations with code showing how things work I think is the best way to learn, because if you can see how its done and how changes affect the visualization, it helps to figure out specifics.