CS6630 Visualization Fall 2017 Project
A Mirror of History

Yanqing Peng, Yuwei Wang

December 2, 2017

The title **A mirror of history** comes from the epitaph of Wei Zhen, a famous Chinese historian:

"Using copper as a mirror allows one to keep his clothes neat. Using history as a mirror allows one to see the future trends. Using a person as a mirror allows one to see what is right and what is wrong."

1 Basic Information

Title   A Mirror of History

Group members

- Yanqing Peng, u1076076, yq.peng@utah.edu

- Yuwei Wang, u1140944, yuwei.utah@gmail.com

Github link   https://github.com/uvril/VisProject

2 Background and Motivation

History is always fascinating. It's important to learn lessons from the past, by understanding the causality behind historical events. To learn about history is to learn about humanity, which basically means developing an understanding of ourselves. Therefore, knowledge of history helps us prepare better for future.

However, it can be boring to learn history by texts and numbers. We can know from textbook that the British Empire used to rule 35,500,000 km² area of land. But how large is it? Showing a map of British Empire is much more interesting than just telling the number. Similarly, we know that the economy of both Japan and Singapore grow dramatically after World War II, but both slow down in 1990s. How can we present a comparison between the economy development of Japan and Singapore? Showing the GDP table will be boring and hard to understand. Instead, showing a line chart for the GDP of Japan and Singapore is much clearer. Therefore, visualization is essential for learning history.

Our textbooks already contain plenty of figures to visualize history. However, it's not enough. While learning history, people have different focuses. Alice is interested in the border of British Empire in 1913, while Bob wants to see a map of French colonies in 1908. Charlie would like to know the comparison between Japan and Singapore, while Eve is curious about the comparison between India and Pakistan. Only with interactive visualizations can one develop the knowledge of history with her own interests.

In this project, our goal was to make a deeper insight into history that will help students and others who are interested in history to explore and learn about how countries have changed over time) as well as comparison among countries in different dimensions, by constructing an interactive visualization on history.

3   Related Work

When searching for project ideas, we found that there are some good vitualizations for history already, such as Delicious Visual Map of History(https://waitbutwhy.com/2015/02/delicious-visual-map-history.html), World Population History(http://worldpopulationhistory.org/map/1700/mercato 1339.8571428571427/-55.971428571428646/), Histography(https://histography.io) and so on. However, plenty of them focuses on different aspects (e.g history events, population, empires) of history so it is difficult for users to get an overview. What's more, some of them are even static, which means users can not interact with the graph. figure?? figure?? figure?? Among the visualizations, Chronas(http://chronas.org/history) did the best job. It includes many aspects and also animations on the process of history over years. figure?? figure?? However it only focuses on displaying the overviews, without showing any analysis on each single country. Besides, some visualizations are too complicated to be understood as figure ?? shows. Therefore, we think making a interactive and general visualization on history is crucial.

4   Visualization Objectives

The main objective of this project is to interactively visualize the current situation of the world, as well as the historical rise and fall. Our visualization is able to show the following results:

- Show the historical world map at a specified year.

- Show the basic information (e.g. capital) of a specified country.

- Show the comparision on different aspects of a specified country.

- Show the religious composition of a specified country.

- Show the rankings of some statistics of a specified country.

- Show the population/GDP/religion distribution of the world.

- Show the historical statistics of a specified country.

- Compare the historical statistics of a set of countries.

- Show how historical events related to the historical statistics of a set of countries.

5   Data and Resources

In our project, we used a very wide range of data and resources. We spent a lot of time on data collection and processing. All data were processed into json format and different catagories of data are stored in different files. For example, "pop.json" stores all population data. If the data file is a historical dataset, then the year attribute is the root attribute, and the country ID is the secondary attribute. Otherwise, the country ID is the root attribute. In following sections, we will describe the sources of the data, and how we process the data.

5.1   Country Indexing

A major challenge we encountered when processing the data was the issue of country indexing. Our dataset contained formal countries that no longer exist, so we couldn't use any modern country code to index the countries. Moreover, there are countries that have alternative names. For example, "South Korea" and "Republic of Korea" are the same country, but they are used interchangably in different datasets. Therefore we have to insure that they point to the same identifier. We tried several methods to solve this problem and finally we decided to use the Wikidata id as the country index. It can uniquely represent any item that appers in wikipedia. Moreover, it helped us to retrieve data from wikidata. In our project, the Capital of a country and Head of State of a country are retrieved by wikidata.

We can use Wikidata queries to convert a country name to its Wikidata ID, using the following query:

```
SELECT ?child ?childLabel
WHERE {
        ?child wdt:P31 wd:Q6256.
        ?child  rdfs:label ?childLabel.
        filter (lang(?childLabel) = "en").
        filter (contains(?childLabel, "COUNTRYNAME")).
}
```

Figure **??** illustrates an example for looking up the wikidata id for United States. From the result we can see that the wikidata id for United States is 30. However, for some countries that have alternative names, we had to manually build the mapping for them.

The converted mapping is stored in **data/wd.csv**.

For non-historical dataset, most of them contains an ISO 3166 country code field to help identify the countries. With the country code, we could convert them to wikidata id using the following query:

```
SELECT ?name WHERE {
?name wdt:P298 "COUNTRYCODE".
}
```

Figure **??** illustrates an example for looking up the wikidata id for country with country code "DZA". With the country code, we didn't need to worry about the alternative name things. However, as explained above, it works only for non-historical dataset.

The converted mapping is stored in **data/cc.csv**.

5.2   GeoJSON Map Data

We collected the GeoJson map data from www.thenmap.net. It provides world maps after World War II. Although this website doesn't provide the off-the-shelf dataset, one of its APIs supports querying for the world map at a specific year. The API we used looks like following:

```
wget http://api.thenmap.net/v1/world-2/geo|data/1946?data_props=name|wikidata
```

This query retrieves the world map at 1946 with the metadata of country name as well as the wikidata id for the country. The wikidata ID is essential for future data collection such as population.

Therefore, we wrote a script to automatically download all map data from 1946 to 2017 from the website. To save space, we further did a comparison between data of consecutive years, and removed the latter data if the world map doesn't change.

The download script is **data_collection/download.py**.

The comparison script is **data_collection/remove_dup.py**.

5.3   National Flag Icon Resource

We used the national flag icons from https://vathanx.deviantart.com/. The origional icons were named "flag_of_COUNTRYNAME" respectively. We used data/wd.csv to convert them into names "WIKIDATAID.ico". These icons are stored in folder **icons/**.

5.4   Statistics Data

The statistics data were retrieved from wikidata. Since we used the Wikidata ID as the unique ID of countries, retrieving information from wikidata was very easy.

5.5   Population, GDP, CPI Historical Data

We collected the original data from the World Bank and the source links are https://data.worldbank.org/ind https://data.worldbank.org/indicator/NY.GDP.MKTP.CD, https://data.worldbank.org/indicator/FP.CP We extracted the country indexing, year and the statistics and stored them in **data/pop.json**, **data/gdp.json**, **data/cpi.json** respectively.

5.6　National Power Radar Plot Data

The National Power Radar used five datasets: GDP for economy, HDI for development, EIU for democracy, EPI for environment, and GFP for military. Since these datasets have different domains, we scaled all data into range $[0, 100]$ before plotting them on the screen.

　　The radar plot dataset is stored in **data/radar.json**.

5.7　Religion Data

The original dataset from http://www.thearda.com/Archive/Files/Descriptions/WRDNATL.asp provides very detailed catagory. To simplify visualization, we only used a coarse granularity of catagory (say, Christianity), instead of detailed catagories (say, Catholics).

6　Design Evolution

6.1　Dataset Change

In our proposal and Milestone, we use a dataset contains ancient data (as early as 2000 BC.). In our final design, however, we only use dataset since 1960. This is mainly because the statistics for ancient countries are largely missing. Especially for modern indexes like Human Development Index (HDI), obviously we are not able to get the corresponding data for ancient countries. In order to make our visualization consist, we made the hard decision to use only recent data.

6.2　Visualization Change

Our original design is shown in Figure **??**. However, when implementing it, we realized that there were a lot of issues in this design.

　　The major issue is that we underestimed the spaced needed for world map. Since we are showing a world map and wish that each country is selectable, the map requires a very large space. Figure **??** shows the original design. We can see that it's very hard to select a desired country, even with zooming. Therefore, we deside to use the whole screen for world map, and reorganize other panels either aggregated in map or create a new tab for them.

　　Besides, we find that if we just list the statistics like GDP as Figure **??**, these information overlap with the information in the comparison panel. Therefore, we adopted a different visualization: we now use a radar chart to represent an overview of national power in different aspects, and a donut chart to show the religion in the country. In other words, visualizations in information panels are supposed to be fancy and highly intuitive, and those in comparison panel should be formal, accurate and detailed.

7   Implementation

Get started!   (Nov 4)

After spending a lot of time on data collecting, we successfully collected the raw data of the historical country borders. The data we collected come from two different data sets:

- For maps after World War II (1946-2017), we collect from www.thenmap.net. Although this website doesn't provide the off-the-shelf dataset, one of its APIs supports querying for the world map at a specific year after WW2. The API we used looks like following:

  ```
  wget http://api.thenmap.net/v1/world-2/geo|data/1946?data_props=name|wikida
  ```

  This query retrieves the world map at 1946 with the metadata of country name as well as the wikidata id for the country. The wikidata ID is essential for future data collection such as population.

- For maps from 2000BC to 1945, we find the dataset in a cartography forum (http://www.cartotalk.com The author is anonymous. The dataset only contains a few years. That's reasonable enouth – we never expect to find a dataset contains all data from 2000BC to 2017AD!

For other data (e.g. population), it seems promising to retrieve them from Wikidata. We plan to explore the possibility for that. The first dataset already includes the Wikidata ID for all countries. We may have to do some data processing on the second dataset to add the Wikidata ID information to it.
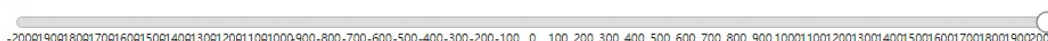
Implementing the map   (Nov 5)

We implemented a prototype of the map chart and the year chart. It looks like the follwing (Figure 1):

Figure 1: Current state of our project, Nov 5.

For the year chart, there is a slide bar and a circle which we can drag to choose the desired year. When the choice of year is updated, the map chart read the dataset with the nearest year before the selected year, and draw it on the screen. The projection we currently use is d3.geoPatterson().

Adding label is quite challenging. If we simply put all labels on the map, then there will be a mess in Europe. Currently we only show the labels for countries with areas larger than a threshold. That doesn't work for countries with large area but extremely long names (e.g. Democratic Republic of the Congo). Whether or not a label is shown should depend on both country area and the length of country name. We should think carefully about it.

The location of the labels is another issue. Currently we put the label at the centroid of the corresponding path. But it doesn't work for some countries like the U.S. The label of U.S. is actually located inside Canada because of Alaska. We have to find a smarter way to locate the labels.

Layout design, and more on label placement   (Nov 6)

We finished a prototype of the info panel. We also improved the label placement. As shown in Figure 2, now it looks better, isn't it?
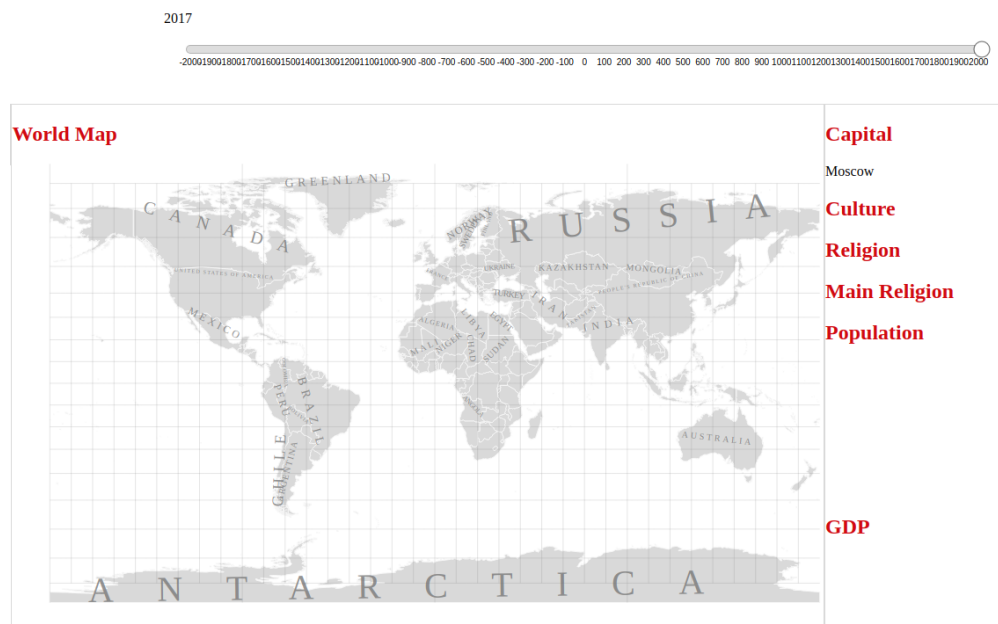

Figure 2: Current state of our project, Nov 6.

As shown in the above figure, the information of the capital of a given country has already been successfully shown. We use Wikidata to retrieve all such information. The query uses SPARQL. As an example, if we want to know the current capital of Russia, the language looks like the following:

```
SELECT ?name WHERE {
        wd:Q159 wdt:P36 ?link.
        ?link rdfs:label ?name.
        filter (lang(?name) = "en").
}
```

Where wd:Q159 is the wikidata id of Russia, wdt:P36 is the wikidata id of the property "capital is", and the variable ?link becomes the capital of Russia. Note that ?link itself is also a wikidata id, which is Q649 in this case. We want to know what Q649 is, so we use the second line in the where clause to retrieve the its label, and use the third line to restrict the answer to be in English.

For the label placement, we devised a smarter way to put the labels. We first decompose each country into continous polygons, and pick the polygon with largest area (e.g. for the U.S., only the continent part will remain). Then we calculated the longest segment within the polygon, and place the label on that line. The font size will be dynamically adjusted according to the ratio between the segment length and the label length.

8

Information retrieval ready   (Nov 7)

We finished most information retrieval part, as well as the css style for selected country. We also did a little improvement on label placing: instead of just placing the text on the diameter of the polygon, now the text is placed on a curve going from one end of the diameter to the centroid, then to the other end of the diameter. As shown in Figure 3, it significantly improve the effect for countries with trapezoidal shapes such as U.S. and China, compared to Figure 2.
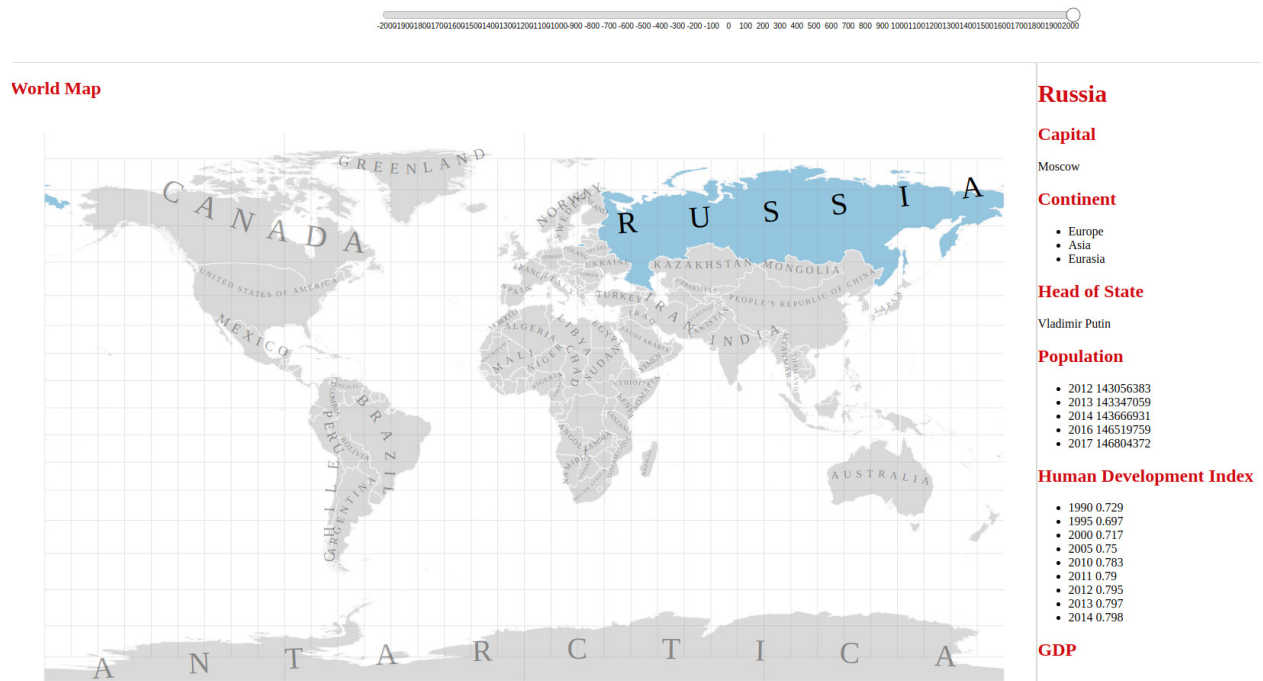

Figure 3: Current state of our project, Nov 7.

InfoPanel design   (Nov 8)

For info panel, currently there are eight parts including selected country name, capital, continents the country belongs to, head of state, population, human development index, territory and description from Wiki. We retrieved all the data from Wikidata and just put them as original way without any vitualization. In this way, we found that some data are redundant so deduplicating is necessary. For example, the continents Russia belongs to include Europe, Asia and Eurasia but Eurasia means Europe and Asia. In addtion, we decided to label each tile when the mouse hovoring over it since the population data are quite sparse in some areas.

For the description from Wiki, we first tried to bind the source of link with the direct search result from Wikidata. However, in this way irrelevant infomation takes too much space like the nav bar of Wikipedia. We took into account the limitation of the page size and got rid of these infomation by adding "printable=yes" at the end of the link.

Visualizing the InfoPanel design   (Nov 10)

We were busy these days. The only progress we made in these days is that we tried to visualize the data in the information panel. As shown in Figure 4, now the population is shown as a line chart, and the HDI is shown as a tile chart.
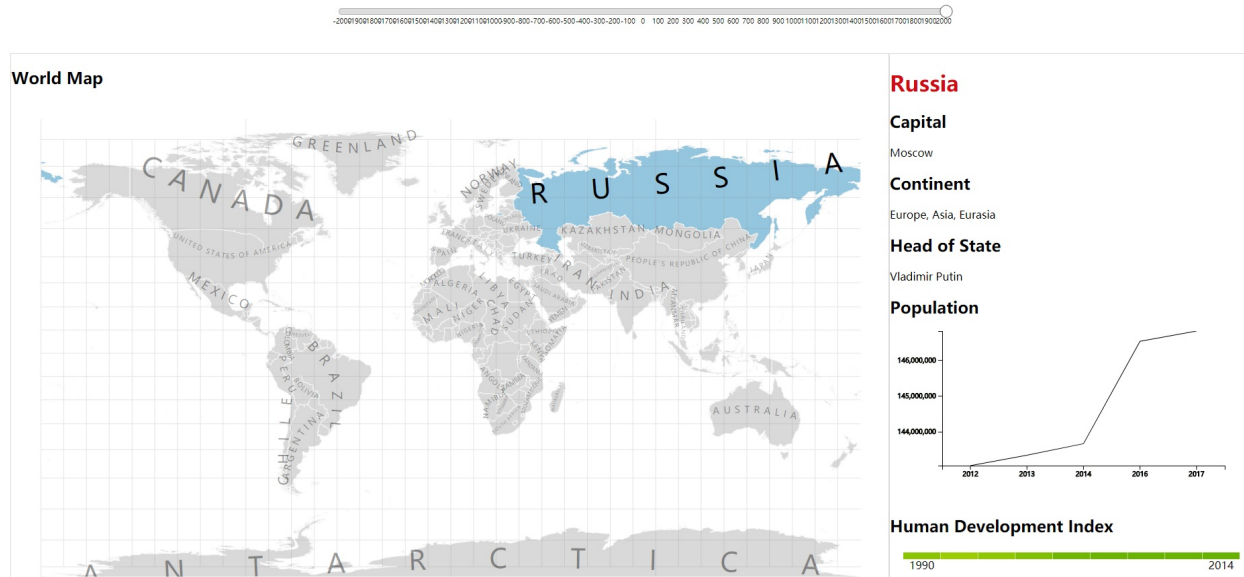

Figure 4: Current state of our project, Nov 10.