Process book - Data Visualization
Trade.io

Mert Soydinç 321131
Maximilian Gangloff 322220
Ziwei Liu 336553

Introduction

World trade data reveals trading relations between countries and the dynamics of global economics. However, such data is not easy to interpret for the general public with any previous trade knowledge. We designed and implemented a website to show complex trade networks between countries based on their import/export relations using easy to understand colored flow of products on a 3D interactable globe. This enables the broader public who are interested in world trade and economics to quickly understand the trade relations and their evolution between any two countries.

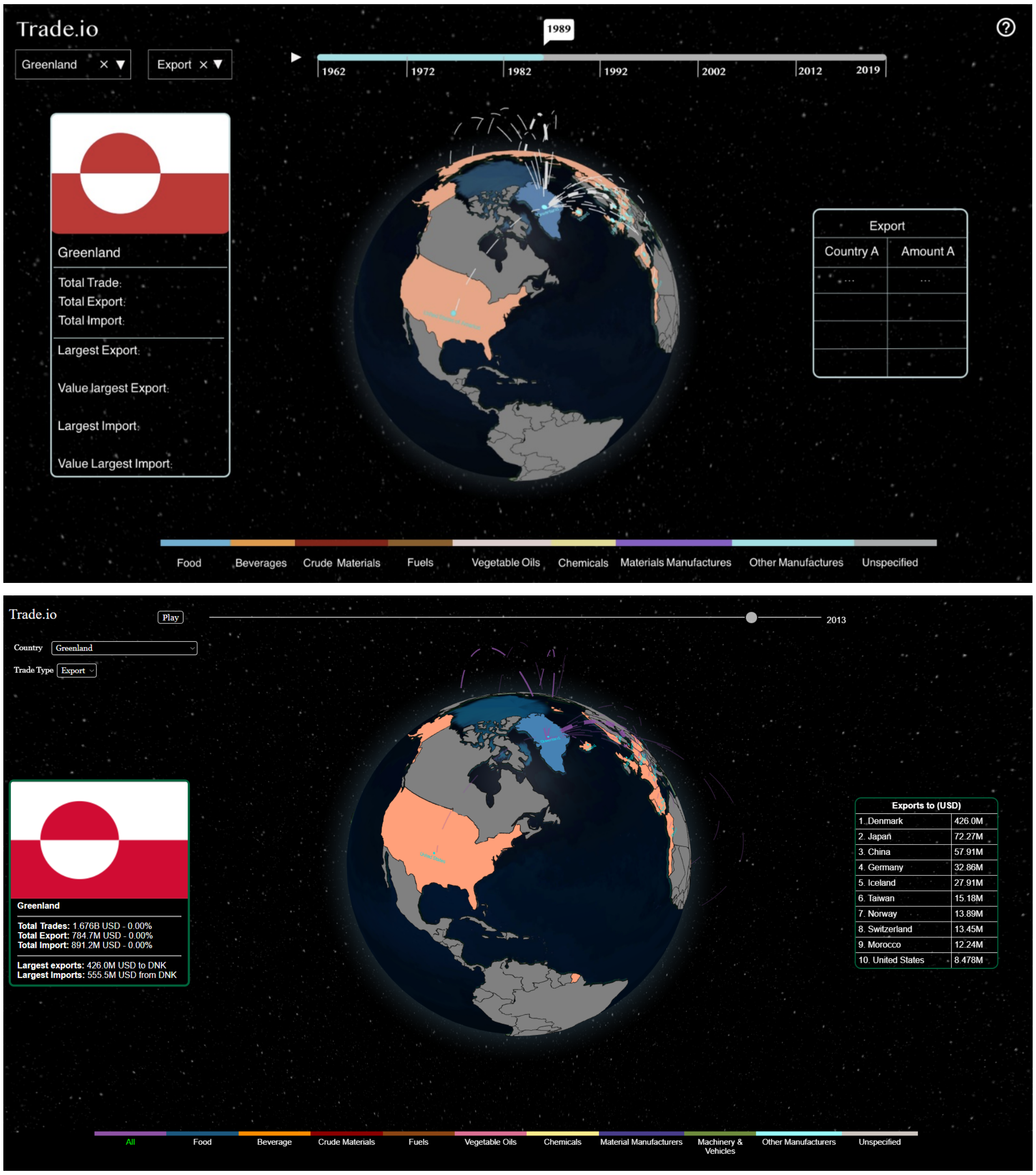
With such visualization, users without any trade background would be able to easily find answers to questions such as which countries are the biggest importers/exporters of a given product, which countries are strongly linked or dependent, and what product each country imports and exports the most.

Dataset Choice

Initially, we did not have a solid idea of what we wanted to work on. We browsed through some dataset platforms such as Kaggle and Google Datasets, looking for interesting data. Our idea was that if we found a good dataset that was already processed and cleaned and also contained accurate information about an interesting subject then we could spend more time on the features rather than struggling with the data source. After some searching, we came across some promising candidates that included datasets about World Trade, Climate Change, Bestseller Books and Hotel Booking.

After some discussion among ourselves, we decided on the world trade dataset partly because it was already very cleanly processed and it was readily available from many sources. We ended up using a dataset provided by Harvard due to the Harvard dataverse being better structured and organized than its alternatives. Compared to other datasets, it provides trade data over a longer time span, from 1962 to 2019.

After looking at some related visualizations, we decided to show complex trade networks between countries based on their import/export relations for different product categories using easy to understand colors and visuals on a 3D interactable globe. We decided that the main target audience for the visualization should be the broader public interested in world trade and economics without any past trade knowledge who wants to quickly understand the trade relations and their evolution between any two countries.



Greenland

Total Trades: 1.676B USD - 0.00%
Total Export: 784.7M USD - 0.00%
Total Import: 891.2M USD - 0.00%

Largest exports: 426.0M USD to DNK
Largest Imports: 555.5M USD from DNK

Exports to (USD)	
1. Denmark	426.0M
2. Japan	72.27M
3. China	57.91M
4. Germany	32.86M
5. Iceland	27.91M
6. Taiwan	15.18M
7. Norway	13.89M
8. Switzerland	13.45M
9. Morocco	12.24M
10. United States	8.478M

Figure 1. Initial sketch (top) vs Final Design (bottom)

Website design

Because of the fact that we were visualizing trading data between countries, we chose to use a world map globe to allow the user to have more interaction with the webpage.

Our main screen is an interactive world map showing the country's borders and can be spun around. This map is colored according to the percentage of either their import or export of the selected category by the user. We assigned colors to different product groups so that they will be visually distinct.

There are 2 modes: export and import; information on each category will be visualized independently. The user can select a single product from the bottom selection bar to see only the colors (imports & exports) corresponding to that product. Selecting a country would clear the map of all colors unrelated to that country and the map will now show arrows representing the flow of trade going in and out of the selected country.

Since our data has depth in time, we provide a way for users to see the trading information in different years and also its evolution. We, therefore, chose to place a slider on the top of the page, where users can select a specific year. A play function also enables users to see the continuous evolution of trading.

Data Analysis and Wrangling

We knew from the dataset description that we would have the export and import trade values between many countries in the dataset. After the analysis of the data, we realized that we had 2 variables that would be great to use during visualization. These variables were the year the trade took place and the product type of the trade. The dataset listed every individual trade between countries which is not necessary for our implementation. What we needed was a dataset that when given a country, a product type and a year would give us the top 20 other countries it traded the most and the corresponding trade values. Because of this, we decided to create a .json file to hold the required data which would be a lot smaller and the dictionary nature of .json fits very well with our cause. For this, using python notebooks, pandas and NumPy, we created 11 different JSON files each for a different product group including the all products category. Each of these JSON files has the year as its uppermost key. When accessed with a key-value ranging from 1962 to 2019, it returns another dictionary, this time having keys which are the ISO_2 code representation of countries. This in turn returns the value of total imports and exports made, the percentage that these values represent compared to the imports and exports made of the entire world during this year and 40 data rows, top 20 exports and 20 imports, where each row contains the partner countries ISO_2 code and the trade value.

During this process, we encountered several bugs from dataset issues. The most glaring one was that our dataset did not have the correct ISO_2 code for France, Norway and Kosovo which we manually fixed. Another problem we faced was that the country code of Namibia is NA which is interpreted as Not a Number(NaN) by pandas which disturbs the entire data pipeline without giving an explicit error. We again manually fixed it. While not a bug, we encountered an issue born from the fact that countries are not constant and not every country is recognized by others. In our globe visualization, it was possible to click the countries of Northern Cyprus and Somaliland while our dataset did not contain information about them. We fixed this by merging them with their bigger counterparts and making them unclickable. This problem is similar to another problem where we have data for some years for a certain country but not for every year. This is however not a problem with the dataset but has more of a historical reason since there are quite some countries that did not exist in 1962 that exist today for example all the countries that were under the control of the Soviet Union. These countries are colored in a gray color such that they are easily identifiable by the user.

Another thing to note is that our 3d-globe data has implicit political statements in it such as showing Crimea as a part of Russia. We are simply using the geographic data provided by the Natural Earth Data as this is not really relevant to the course. We also realized during this time that we needed point location in terms of latitude and longitude for each country in order to visualize trade's originating and terminating positions. We found another dataset that contained a central location for each country and processed it into a .json file that is again accessed using the ISO_2 code of a country.

The geographic data is contained in a .geojson. The file is similar to a .json file but contains a geometry feature where the structure of the country is represented as a MultiPolygon. These MultiPolygons are then used by the globe.gl library to draw a layer of polygons that represents the countries over the globe. The .geojson file was adapted to our needs, the data that was not needed was removed and some base statistics that are displayed on the country cards were added. These changes were made in the "generate_geojson" notebook. It was also in this notebook that the countries of Northern Cyprus and Somaliland were merged with their bigger counterparts.

Comparison with the original design

Compared to our original sketch, we modified the positions of the country menu and trade selection menu to make more space for the slider. We also changed the appearance of the website a little to fit better with the globe background.

When the user's cursor hovers over a country, a card appears under the hover, showing the country's largest import, export and total amount. It was first planned to be always displayed on the left side of the webpage instead of appearing and moving just under the mouse. However, we liked this function of the globe.gl and left the label appear just below the mouse and now only displays the country card on the left side when a country is clicked.

Also, for the category panel on the button, compared to the initial design, a new category "All" was added since with our initial design, the user needed to reload the webpage to get again the default data of all the exports and imports after selecting a product category. Also, the "Machinery & Vehicles" category was missing in the initial design, since the initial design only showed the placement and design of the categories bar but not the final content.

We also added a loading page to avoid the broken look when the globe hasn't been loaded.

Features and Implementation Challenges

The Globe: This component is our main visualization tool to show the trade between countries. It is located at the center of the screen and it is fully interactable. It has two main modes, the first one is the default one where no country is selected. In this mode, all countries are visible and colored according to the relative percentage of their trade value. If the user hovers over a country then the country gets lifted and an info panel on the left appears. If the user then left-clicks inside the borders of the country, the globe enters the second mode. In this mode, only the selected country and its top trade partners are colored with everything else being gray. Several arcs connect the selected country with its partners which depending on the trade amount take different sizes and colors. Also, the Top Partners Panel appears on the right.

We implemented the globe with its direct features using the globe.gl library. globe.gl is a very comprehensive library that can be used to achieve many visualization tasks on a globe. It has built-in features for creating arcs between two points, displaying text and colors on a given point, spinning the globe and creating many other visual effects. This meant that we only had to supply it with our data and call the correct functions when they were needed. It took a while for us to understand the documentation of the library since some of the instructions and descriptions regarding the fields and functions were subpar. Thankfully, the library has a number of built-in examples that they showcase on their Github page. By reading their code and playing around with the resulting visualization, we were able to get everything working as we intended.

Time Slider: This component is located at the top center of the webpage and displays a year range from 1962 to 2019. The user is able to pick a year on this slider which will then change all the data displayed to that of the selected year. The selected year is shown on the right side of the slider. When the play button on the left side of the slider is clicked, the year, together with the visualized data, automatically progresses until paused or reaching the maximum year, 2019. If 2019 is picked when the play button is clicked, the year progresses starting from 1962.

This function was implemented by integrating HTML, CSS and javascript. Javascript reads the year value from the slider in HTML, increments it during play mode, and updates the data to be of the year accordingly. As mentioned in the data creation section, all we need to do in order to access a year is to supply the already loaded dictionary with the string representation of the matching year. This component basically assigns the currently selected year value to a global variable which is then used to access the data in every part of the script.

One problem we encountered was that data updating for the selected year is only triggered by a click on the Play/Pause button. As a result, when the year automatically progresses in the slider, the data is not updated. To solve this problem, we imitated a click every time the year value of the slider is changed.

Top Partners Panel: This panel is located on the right side of the webpage. It is basically a list of a maximum of 10 countries that the currently selected country has the strongest trades with. On the top of the Panel, the currently selected Trade type(Export or Import) is displayed. A row of this list consists of a country's name and the associated trade amount in US-Dollar between it and the currently selected country. Rows are displayed in the order of descending trade amount. When clicking on a country in this table, the globe hovers to the country and when clicking on the Trade type at the top of the table the trade type is switched.

We already have the information required to display this component available since the same information is used to display the arcs. The table is nearly completely created and updated in javascript. At the start, the table is empty and does not contain any cells except the title. In the first mode when no country is selected the table is hidden. When then selecting a country, the table gets created if it's empty or updated if another country was already selected before. Updating the table instead of creating a new one caused some problems when countries were selected for which there were not 10 trade partners. This made it that only the first rows were updated and then the following rows would correspond to the previously displayed table. To overcome this problem, we now delete every row when clicking on a country and then add every trade partner in a new row. This makes sure that the start length of the table is always 0.

Selected Country Info Panel: This component is located on the left side of the webpage. When a country is selected (by hovering with the mouse over the country and left-clicking on the Country selection menu), general information corresponding to that country is shown here in order to give users a broader look at the trade data regarding that country. The shown information is as follows: Flag and Name of the country, Total Trade amount, imports and exports, Largest Export and Import conducted by this country together with recipient country code and the corresponding amounts. The flags were obtained from https://corona.lmao.ninja. Also, if no country is currently selected, hovering over a country's borders using the mouse cursor brings this component to view under the cursor in order to show the user what country they are looking at. When it is displayed on the left side, clicking on the country info Panel will center the globe on the selected country.

Country and Trade Type Selection Menus: These two menus exist on the top left of the webpage. The country selection is used to select a country on the map and is identical in functionality to simply left-click inside a country's borders. The Trade type menu is just below the country selection and is used to change the type of trade data displayed. There are two values that this component can have: Export and Import. The trade data shown on the other components changes to reflect the chosen one after it is selected.

Both country selection and trade type selection are implemented in HTML as a form with a list of options. Countries are ordered alphabetically and connected to the javascript part to update data to be of the selected country and find the originating and terminating positions.

We already mentioned that our globe has two states, one where no country is selected and one where a country is selected. We wanted all our components to affect the globe in both states and automatically transform the globe when a component is interacted with. This is implemented by using two functions called reset and click. Country Selection Menu calls click and basically imitates a mouse left-click on the selected country. When this happens, the globe rotates to the center of the selected country and every component is updated. The type selection menu works very similarly to the time slider in the sense that all it does is set a global variable to the selected value. This global variable is used to access the data in a generalized way so when it changes, the shown data automatically updates.

Product Categories Bar: This component is the multi-colored bar located at the bottom of the page. By default "All" is selected and all the trade information regardless of the product category is shown. If the user clicks on a product group then only the data corresponding to that group is shown and the colors used in the globe component change in order to convey a visual difference between different product groups.

This is implemented in HTML as a table with colors as the 1st row and text as the 2nd row. When hovered over or selected, the color of the text is emphasized with green color. By clicking on a category, the product category is selected, and this component switches the dataset currently in use. As mentioned earlier, we have a total of 11 different .json files each corresponding to a different category. Since all the logic of the globe and components are written in a general way, we can easily swap one instead of the other and everything is updated automatically.

Play Button: This component is located just left of the time slider. When the play button is pressed the data(and by extension colors) displayed on the globe slowly change to show an impression of evolution through time to the user.

In the implementation, the javascript part reads the text within the play button as an HTML component and is modified when a click is detected. Then during playing, the value of the year is incremented with an interval of 1. When the current year of the slider is before 2019, the progress starts from the current year. When the current year is 2019, it instead starts from the initial year 1962.

Loading screen: The loading of the globe by itself takes quite some time. In addition the 11 .json files are also pretty big. Hence we added a loading screen such that the user does not see the plain website with the product bar, slider, etc. Once the Globe is loaded, the loading screen is removed and the true website gets displayed. However, loading all the data and the Globe at the beginning and only displaying the loading screen would make some users quit the page before the globe even was displayed. Luckily, fetching .json files is by default asynchronous in JavaScript. This caused us however problems as sometimes, the globe would be loaded before the data was loaded which made the globe not display anything until the next reset. To overcome this problem, we wait until the first dataset is loaded and then start the loading of the globe. If the globe finishes loading we display our website with the loaded data. If the user directly selects a category that should not be loaded yet, an alert appears informing the user that the data is being loaded. This makes it so that even if the entire data is not loaded yet, the user can still explore the globe.

Future Work

We believe the core features necessary for the World Trade visualization are complete and any future work would be mainly a stand-alone visualization that would complement our existing globe. One such complementary visualization is the graph view of world trade where each country is represented by a node and is connected to other nodes using colored links which represent different product categories. During the initial stages of our project, we came across similar visualizations but decided to leave them out as it would be too much extra work. Changes in the current visualization of our globe could be to add different colors when selecting a country. Another extension could be made regarding the products. 11 categories are available right now, however for every category, we have also data about their subcategories and their respecting sub-subcategories. When clicking on a product category, the bottom selection could

change into a selection of the subcategories of the selected category and selecting a subcategory could again change the bottom selection into a selection displaying the sub-subcategories.
Another improvement could be made regarding the loading speed and the fluidity of the website. Regarding the loading speed of the data, parts of the data could be cached to be able to display the globe faster and make the loading of the other .json files faster. Regarding the fluidity of the webpage, when quickly changing the years while selecting a country, the webpage gets a bit buggy and the trade arcs stop moving for a short time.

Conclusion

In the end, we are happy with the final result of the website as we managed to implement almost every feature we intended. We also managed to more or less identically replicate our sketch from milestone 2 in the final website. This was also made possible by the fact that we started early with the implementation of our website and had already quite some features implemented for milestone 2. There was surprisingly a lot of data processing involved, more than we expected, but it allowed us to progress our data analysis skills together with our web development knowledge.

Peer-Assessment

<u>Mert Soydinç:</u> Main data Creation regarding the product categories and county locations, part of main globe functionality such as drawing of arcs, resetting the globe, selecting a country, resetting the globe. Linking HTML components (except product selection and Left/Right figures) to globe functionality.

<u>Maximilian Gangloff:</u> Data creation (countries.geojson and some small part of the trade data .json files). Implementation of the following: top partner panel, country info panel, the functionality of the product categories bar, globe navigation (change camera when clicking on an Arc, on the top partner panel and the country info panel), loading screen.

<u>Ziwei Liu:</u> Website user interface design, categories bottom bar, country selection menu, trade type selection menu, top time slider creation and functionality, play button creation and functionality.