1. Research Question(s):

As a person who is very passionate and interested in travelling, I wanted to visualize how the world is connected and the routes available to fellow travelers from their home airports. I narrowed down my mode of transportation to air travel, as that is the most common and connected source of transport across the globe.

Here are my questions that motivated me:

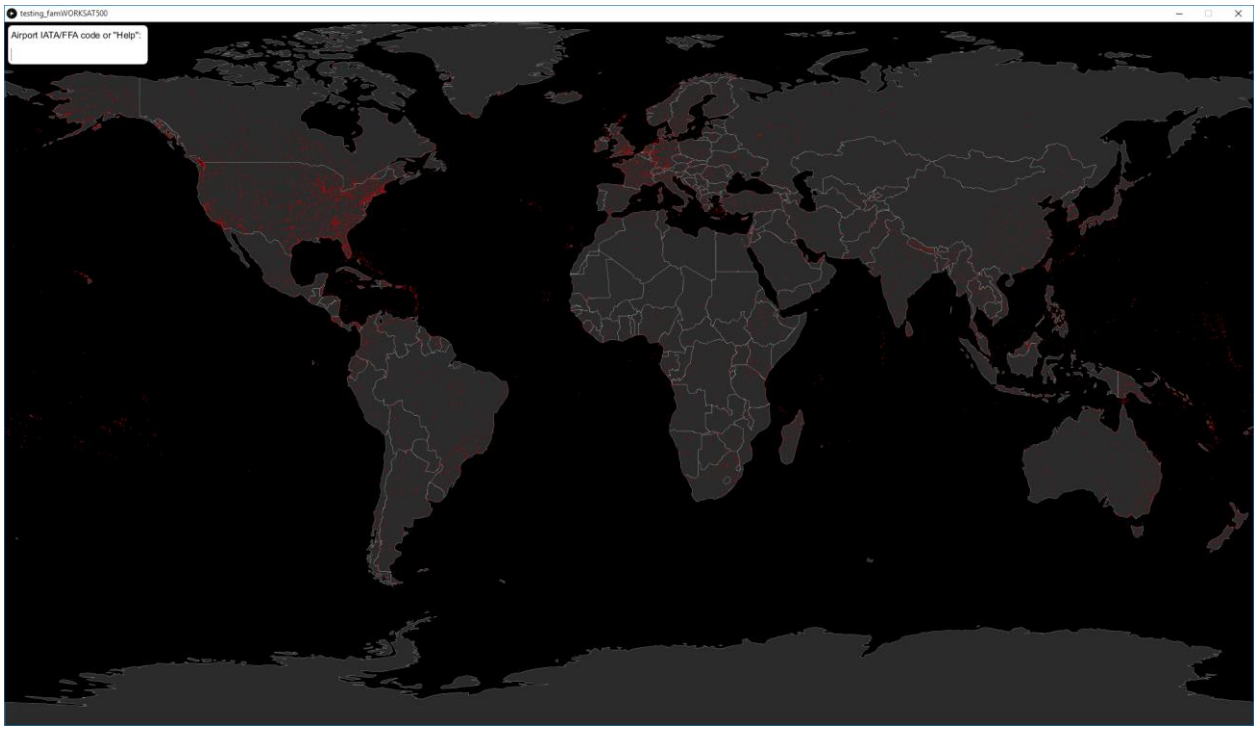
- **What are the possible connections from each airport?**
- **How is the world connected through air travel?**

2. Data Source(s):

OpenFlights – Airport, airline and route data
URL - http://openflights.org/data.html
GitHub - https://github.com/jpatokal/openflights

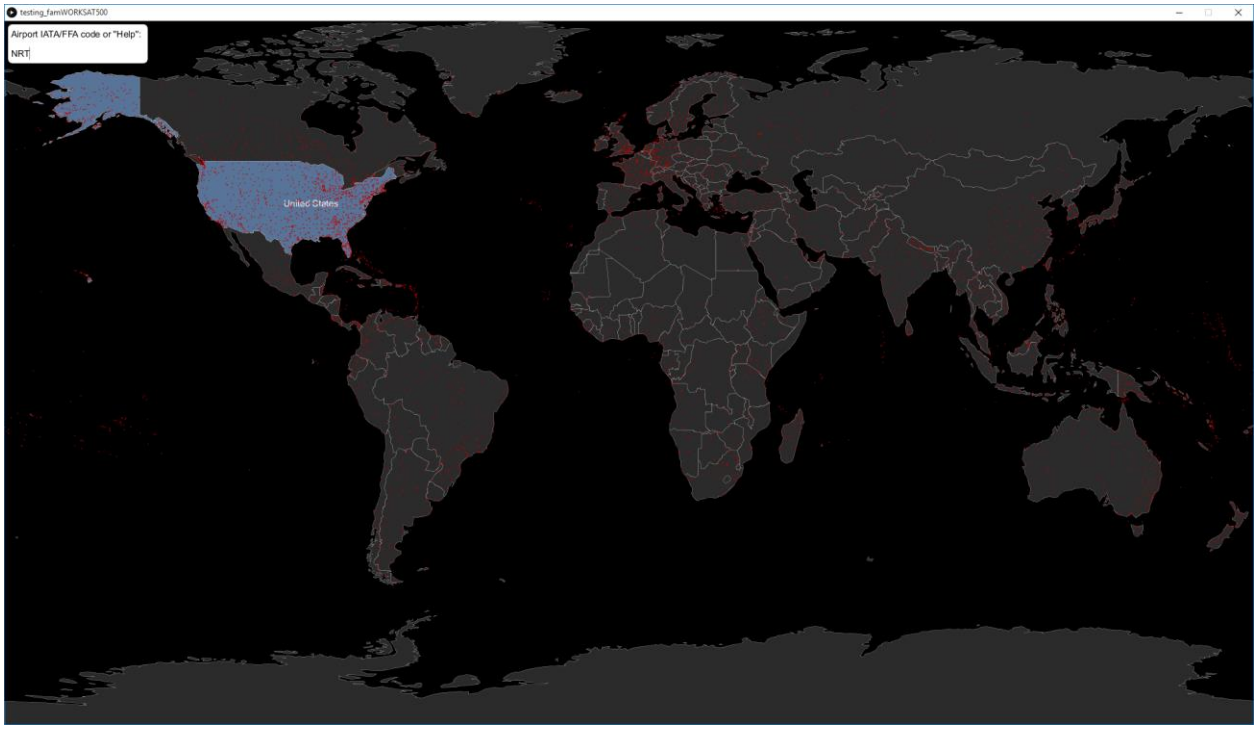
3. User Instructions:

Step 1: Initial Screen; the small red circles represent airport locations across the world.

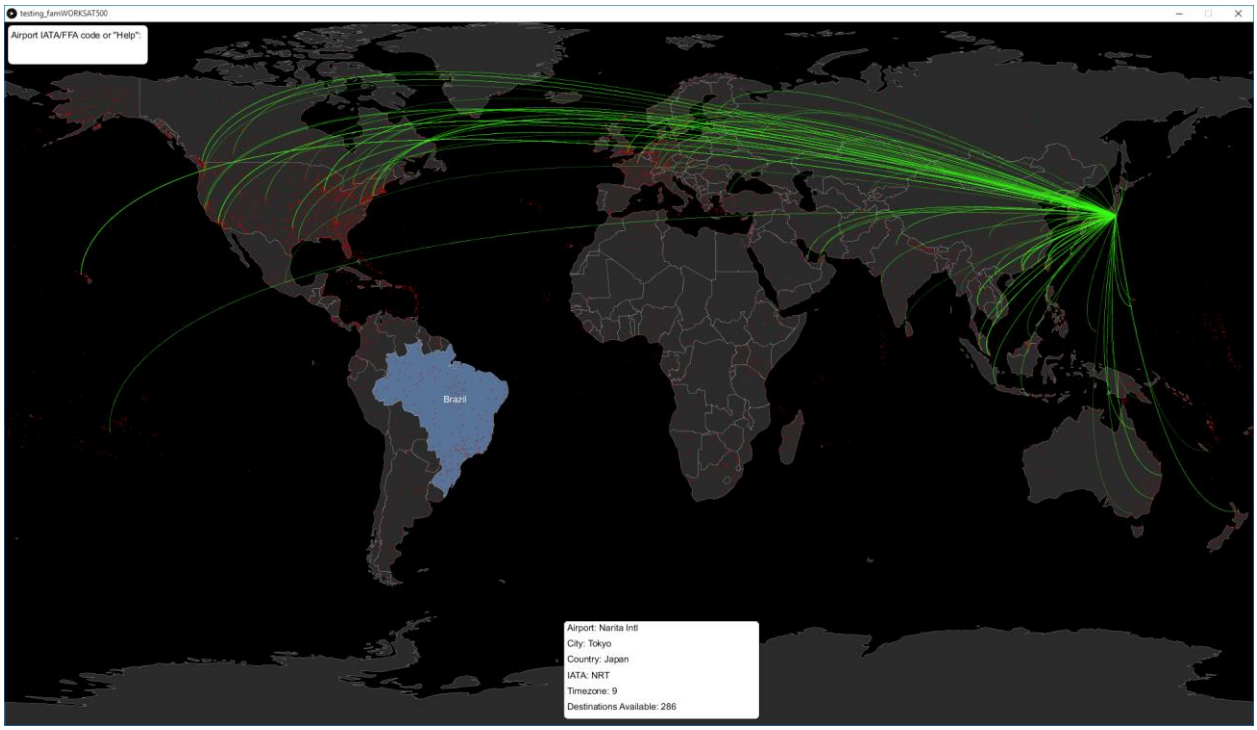


Step 2: Searching for an Airport and its possible destinations; please use this link to find your desired airports IATA/FAA Code:
https://en.wikipedia.org/wiki/List_of_airports_by_IATA_code:_A
After you've found your desired airports code, type it in capital letters, in the top left corner of the screen and hit ENTER. You can also hover over a country and it will display the name of the country. You can also display all the possible routes between all the airports by

typing in “everything” (be aware, it takes up to 30seconds to load). Type “help” for directions.



Step 3: Results; once you’ve pressed ENTER, you will be shown all the possible direct routes from your airport, in order to search another airport, just press ENTER again.



You will also be presented with a short info box for your selected airport, such as Airport Name, City, Country, IATA, Timezone and the amount of direct destinations available. As we can see Narita Airport, Japan has 286 possible destinations which are mainly located in Europe and North America.

4. Design Justification:

4.1 Visual Encoding:

For the layout of the sketch, I used the GeoMaps library from giCentre as it was very handy to display an interactive world map due to the library features. Initially I wanted to display the world without including Antarctica as I thought it would be redundant as there are no commercial airports there, however I decided it would be a very good idea to keep the continent in my sketch as allowed me to visualize the fact that there are no possible connections by commercial air.

In order to represent the route paths I used Bezier curves which would have a span from airport to airport per route, I felt like this was the best way to represent paths as for one, it's a good representation of how airplanes fly in the physical world, which I felt would correlate well onto my 2d world map. In order to represent airports on the map I plotted each airport as a small red circle on the map.

When it came to giving color to my sketch, I wanted to go for a high contrast radar look. So I made my background black, with the world map being a darker grey with light border edges to show clear separation of countries. I felt like this would allow the whole sketch to have a darker theme while still allowing the world map to stand out. I chose to plot the airport locations with red, as this would allow for high contrast between the dark maps as well as the light green that I chose for the curved edges that represent the routes.

I made sure that the hover over feature that distinguished countries would allow the user make no mistake between countries so I chose a shade of blue as the fill for the hovered country. I felt like this would provide high contrast against the existing red and green colors.

Lastly, I created the onscreen textboxes as white, as these text boxes are there to provide the users with information, so I made sure they were readable, and white was the perfect color to provide readability on a black background.

4.2 Interaction:

The interaction in my sketch is kept to a minimal, I wanted users to have a very simple experience when using my sketch. Users are presented with a text input box where they type in their airport search query by FAA/IATA code input or displaying all the possible destinations by typing in "Everything" and simply pressing enter.

Users are also able to type "help" which will display a popup box with directions and an explanation of the sketch. Also press "s" to save an image of the sketch.

Users are also able to interact with the sketch by using the mouse cursor to hover over the world map, each country they hover over will display the countries name for the user to see where destinations are headed if unfamiliar with particular parts of the world.

5. Further Work:

There is very much I envisioned when starting this project, however, I was not able to implement my vision in its entirety, due to time constraints as well as resource constraints. The main thing I would like to further implement into my visualization is use of live data, which could be gained from the likes of SkyScanner API. This would enable me to provide the user with more information such as live prices and live routes from home airports, which would inform the user of possible trips and what it would cost them, which I could visualize by providing a price cap on the routes, so for example users would be able to select a price cap and type in their airport, and it would then display all the possible routes within the selected price range. Another thing that I would be able to implement with a live flights API would be live flights and their location/duration, by animating planes along their axis in proportion to their live location.

In terms of visual encoding, I strongly feel the use of a 3D map world such as possibly the Google Earth API, which would perhaps reduce the visual clutter of the route edges that is seen on the 2d world map and create a more feasible visual model, which would be easier to navigate through in 3d space for the user.

Lastly, optimize code to reduce lag. Right now EVERYTHING takes 30 seconds to load, on my machine, which is quite bad. So I would like to either produce a loading feature which informs the user of loading time, or optimize the lag through code.