CSU11013 - 2023: PROGRAMMING PROJECT
JUNIOR FRESHMAN - HILARY TERM
GROUP 10 PROJECT REPORT

Ellen Brennan - Conor Daly - James Treanor - Dhruv Ranajit Choudhury

PROJECT OVERVIEW:

The overarching objective of this project is to analyse and visualise a dataset consisting of more than half a million flights in a single month from the USA. The first step we took was to convert the dataset, originally provided to us as a CSV file, to a SQLite Database for easier querying and management. We were successful in building an interactive dashboard which consists of major sections: Airport statistics, Airline Statistics, Heatmaps (Data for states) and a Flight path visualisation. Each section has different forms of data visualisations with interactive user interfaces.

FEATURES OF THE PROGRAM:

The first thing the users sees upon entering our programme “Flight Finder” is our loading screen.A non looping gif of a plane flying runs while our SQL queries are going on in a separate thread from the “Animation Thread” of Processing. Once our queries are finished, the UI appears and the user is able to navigate to our data visualisation. With the still last frame of the gif remaining as the homepage background for the duration of the programme.

The next page in our application is the airport stats screen, it allows the user to filter the amount of flights arriving/departing from a particular airport And break it down by days of a particular week, or weeks in a month via a bar chart. The user is able to hover over each bar to see individual points of data.

The airline stats page functions similarly but makes use of a histogram with a randomly generated colour scheme. It presents to the user airlines by popularity, number of cancellations and diversions.Similar to barcharts, able to hover over each bar to see individual points of data.

The heatmaps page takes in data from our SQL queries by state, and colours out each state based on a given colour palette and ranges outlined by us. It displays three different kinds of heatmaps, total flight activity by state, arrivals and departures.

Lastly we have our flight pathing page, The flight pathing page makes use of drop-down menus to allow a user to map out the general direction their flight is going to take. It does this by animating a line(using interpolation) between the origin and destination airports.Where the origin airport is outlined in green, and the destination in red.

Workload split:
We first divided our effort into three sections: database, graphing, and screen/widget approaches.

Our first tasks were as follows:
James had successfully learned SQL for our project and had had many SQL queries running to allow us to graph our data.

Conor looked into constructing other graphing methods such as histograms, bar charts, and pie charts (which we didn't get to utilise).

Dhruv had created our basic Screen class, examined the ControlP5 library, and successfully integrated drop down menus, radio buttons, and even multithreading!

I got our loading screen's gifs to work, as well as our widget approach and a heat mapping method that worked by manipulating the XML code of a VSG file.

As we worked together on bug fixes and new features, there was a lot more overlap in everyone's duties and responsibilities, so I've produced a more thorough chart below to offer additional specifics on each of our team members' achievements in this assignment.

MEMBER NAME	FEATURES IMPLEMENTED
James Treanor	- Physically drew out the plan for each of our Screens - Learned SQL for ease of data manipulating - Formatted data from SQL queries to be usable by graphing methods
Conor Daly	- Researched manipulating our database in Java incase we couldn’t work with SQL - Developed scalable histogram method - Developed scalable bar charts method - Developed pie charts
Dhruv Ranajit Choudhury	- Developed skeleton screen class - Created dropdown menus with ControlP5

	- Created radio buttons with ControlP5 - Sped up our SQL queries with the largest database by creating Indexes (Can be seen in load-data.txt file in data) - Put our SQL queries into a separate thread that ran in parallel with loading screen - Formatted data from SQL queries to be usable by graphing methods - Developed flight pathing method - Mapped out coordinates for flight pathing method - General bug fixing
Ellen Brennan	- Developed skeleton widget class - Made loading screen gif - have main menu UI appear when our queries were finished - Developed heat mapping method - Developed flight pathing method - Mapped out coordinates for flight pathing method - Updated screen/widget class in order to allow for - ability to go between screens - General bug fixing

TEAM ORGANISATION:

Our team for this assignment was not really formed with a hierarchy in mind. We would instead make use of online tools such as when2Meet and discord to find out a time that worked with everyone’s schedule and brainstorm until we came up with an idea we all agreed upon, and tasks everyone felt comfortable pursuing.

We would generally discuss plans over the weekend via discord and meet on Wednesday afternoons to discuss our progress and assist others where necessary, the next few tasks were then elected at said in person meeting, and adjusted accordingly based on how the Thursday lab session went.

As the deadline for this project came a lot closer we would meet up more frequently via online calls over the long weekend.

PROBlEMS ENCOUNTERED:

While we stand by the work we did on this project, there were some teething problems.

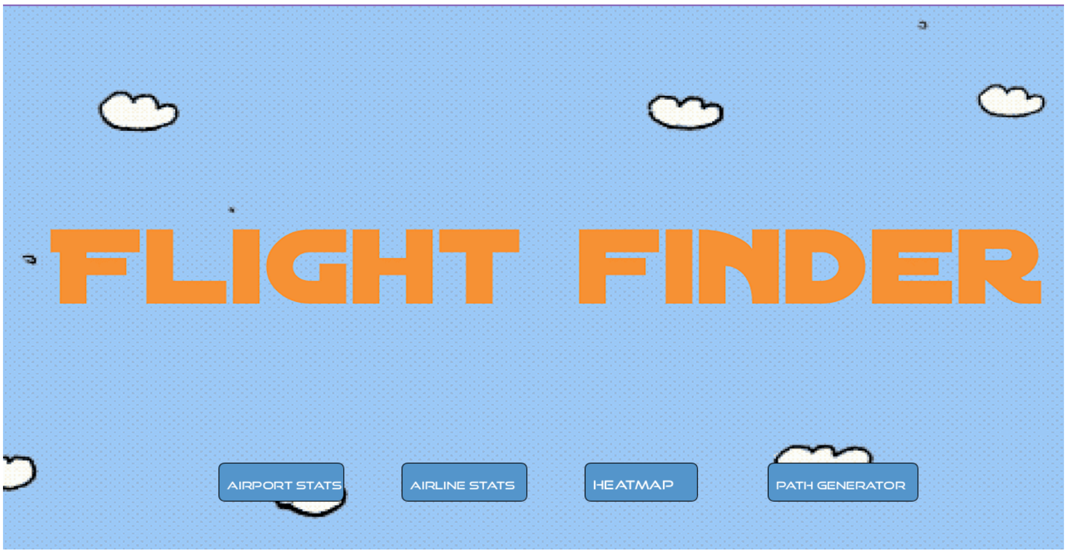
As we began to grow used to programming together, our program suffered greatly from attempting to integrate our work, because we did not initially construct our individual components with efficient merging in mind. This resulted in us having three distinct sorts of queries but on different PCs when we needed them all on the same screen, resulting in us losing marks for work we had finished but not merged.

Another major issue we had was within our screen class, we had a hard time identifying why only one widget per screen would be functional at a time, despite our widget syntax being in order.
It turned out that we had actually written a for loop in such a way, that it would only reach the first index of our widget arrayList, before terminating.

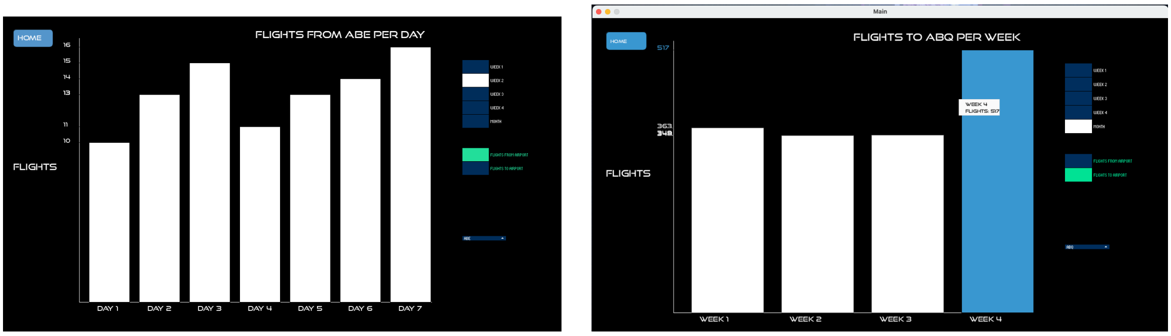
Lastly, we had the issue of our SQL queries taking up too much time, there would be around 25-30 seconds of delay upon initially running our programme as it would tackle our queries first, then begin to draw the graphics. We resolved this problem by implementing two fixes: speeding up our SQL queries by creating Indices in the database and having them run in parallel with our loading screen GIF via multithreading.

So the user would be immediately able to open the programme,but the UI on the homepage would not appear until our queries had finished.

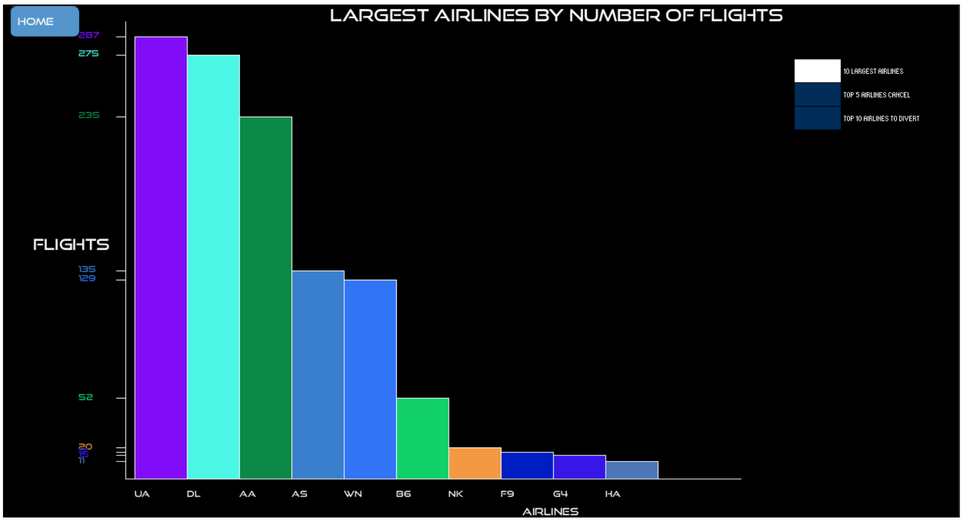
SCREENSHOTS



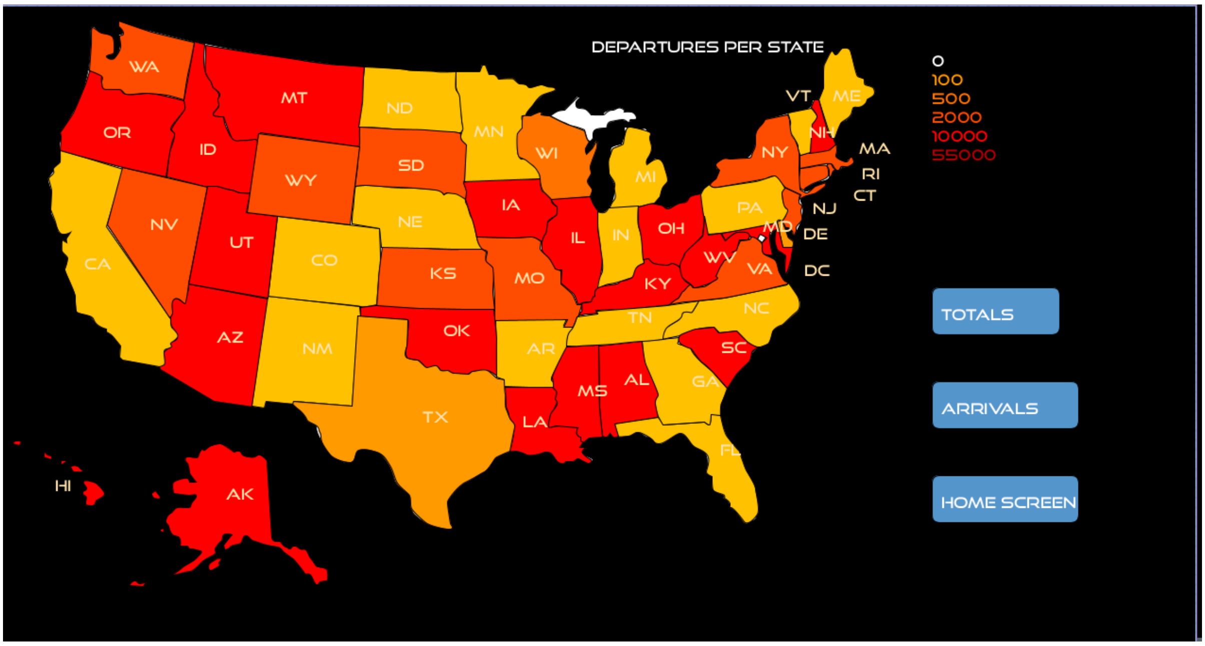
Home Page with UI to navigate to the rest of the program



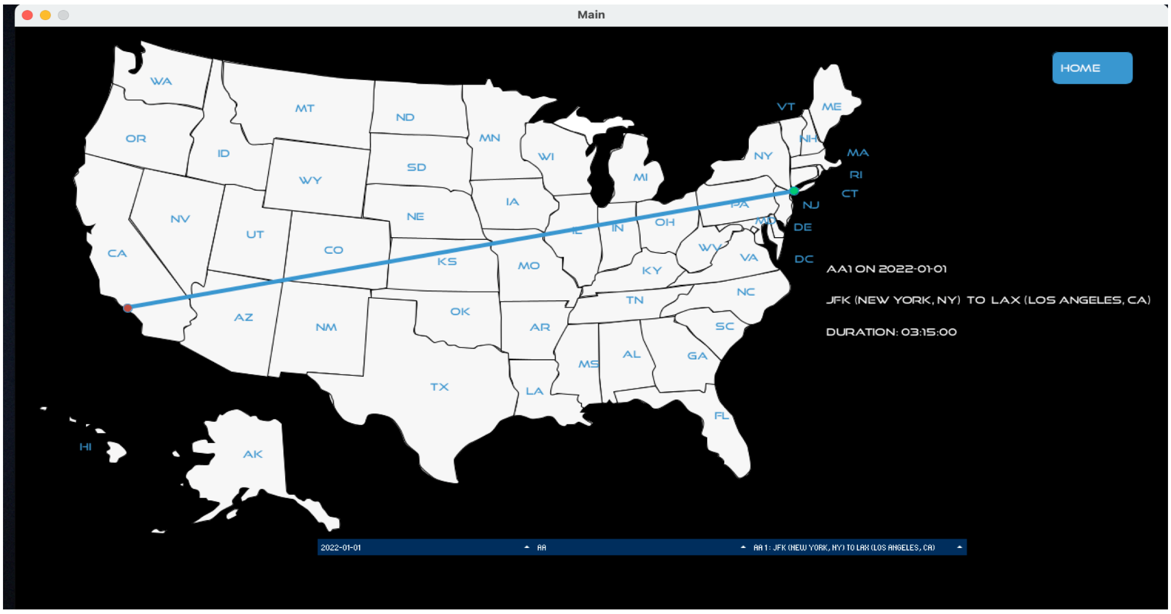
Dynamic, calable Barcharts for Airport Statistic



HISTOGRAMS FOR AIRLINE STATISTICS



HEATMAPS FOR VISUALISING FLIGHTS TO, FROM STATES AND TOTAL FLIGHT ACTIVITY IN STATES



AA1 ON 2022-01-01

JFK (NEW YORK, NY) TO LAX (LOS ANGELES, CA)

DURATION: 03:15:00

2022-01-01 ▲ AA ▲ AA 1: JFK (NEW YORK, NY) TO LAX (LOS ANGELES, CA) ▲

FLIGHT PATH GENERATOR