# High5ive

The goal of the project was to construct an application to explore data relating to commercial flights, including methods to load and read data, to select specific queries, and to present only the chosen queries on the screen. Over the course of this module, we achieved all these goals, while using a much larger database of flights sourced by our team. On top of that, we choose to, apart from using more common forms of presentation such as charts and graphs, to present the data using a 3D model of the earth.

Alex created a UI toolkit similar to Windows Forms. This includes an event system where handlers are added to a Widgets event *[button.getClickEvent().addHandler(e -> ClickEvent());]*. This event system simplifies the usage of Widgets, as you simply need handlers, which are called on the event. All Widgets are derived from the Widget class, which allows every screen to hold a polymorphic list. Events are raised by the screen automatically, by iterating through the list of Widgets and checking if they are an instance of a relevant interface. Widgets used included Button, Textbox, Listbox, Scrollable dropdown, Checkboxes, and Radio buttons.

We decided to preprocess the data into raw binary for performance. Cosmo converted all decimal values to binary equivalents, then used a small processing script that runs separately from the main body of the project to replace text values with a numerical key. These keys were stored in a series of lookup tables - one for each of airports, aircraft and airlines - which stored details like the name or country. This allowed the database files to be very small and load quickly into processing using Tom's code, while still retaining human readability when converted. Cosmo also wrote functions to access these lookup tables, using native Processing spreadsheet functions. She was able to find lots of additional data online for larger world-sized datasets using web scraping techniques, allowing the scope of the program to be expanded.

The compact dataset allowed Tom to use advanced features for reading like Map Buffer which can read a file of 2GB (equivalent to apx. 67 million flights) in 2 ms. Tom serialised this data into a FlightType Object. With this FlightType Object, Tom created query functions with dynamic enum query typing for: querying by operation, query between a range and by frequency. He also implemented a function for sorting the

data by a type. Tom was also responsible for the actual design of UI, such as the home screen and color palette.

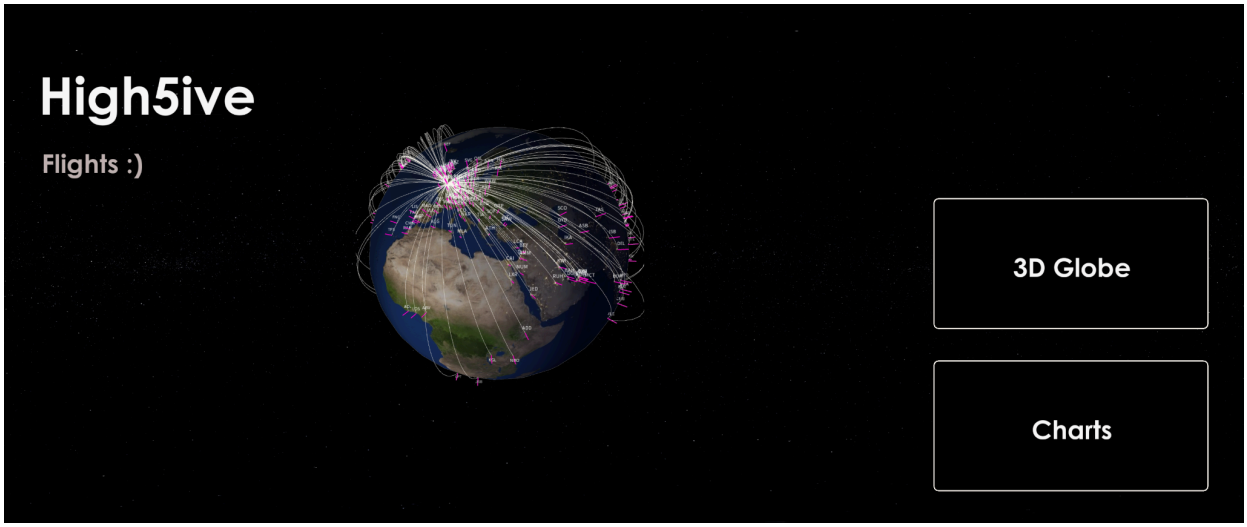
Finn created a 3D flight map to represent connections in a visually engaging way. Multiple vertex and fragment shaders were written to deal with lighting calculations, texture blending, specular highlights, normal mapping, texture translation and post-processing. Normal mapping was done using downsized online earth surface maps and normal matrices to find the aligned normal vectors. Geometric Spherical Linear Interpolation was used to create the line segments for the flight connection of great circle arcs. A standard skysphere was created by forcing the depth buffer value of the mesh to be infinite. The pie chart was made using the built-in Processing arc() function to draw sectors of a circle. The scatter plot initially seemed daunting due to poor performance but he was able to optimise them by using a mesh with vertices to represent the points and skipping duplicates.

We decided to attempt to create a 2D flight map with similar capabilities for data presentation to the 3D map. Mati created the 2D FlightUI and the 2DFM classes, which included the base image for the map, the UI to interact with the map, and functions to convert latitude and longitude for our data to a location on the screen. However the 2D map was cut from the final product due to a lack of time.

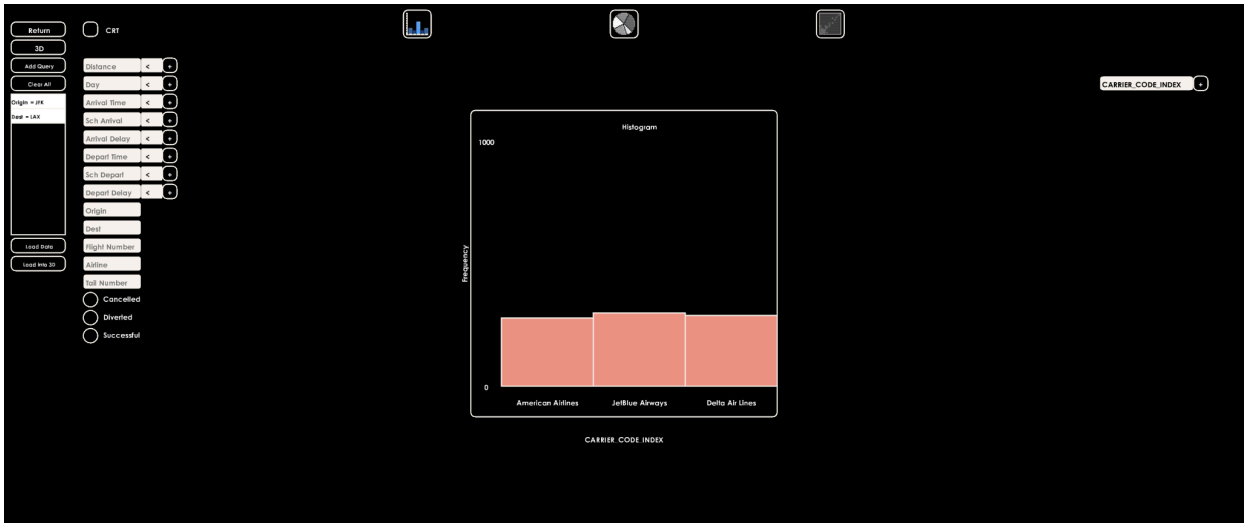
Matthew's responsibility was to create the User Querying interface for the 3D map and charts system.
He achieved this by creating input fields that took in a user input, then communicating with the querying system that was made by Tom. This then created an event which transferred an arrayList of queried flights returned by our flight querying system into the 3D flight map to be rendered. This interface used textboxes, radio buttons and combined textboxes with dropdowns. This allows users to seamlessly search data from both our world and US dataset, with the result being displayed accordingly onto our 3D flight map or Chart screen. A struggle that Matthew came upon when coding was that not all Querying types were compatible with the world data. To solve this issue he created a system that would lock off incompatible queries when the world data set was in focus.

As seen by the efforts of our team, we strived to go above and beyond the project brief and we hope that was reflected in the final product. The team came together very effectively, developing workflows that took advantage of everyone's strengths in programming, being able to work collaboratively where it gave the most benefit, while also being able to develop functions individually, especially at early stages, allowing rapid progress to be made with the base framework for the program

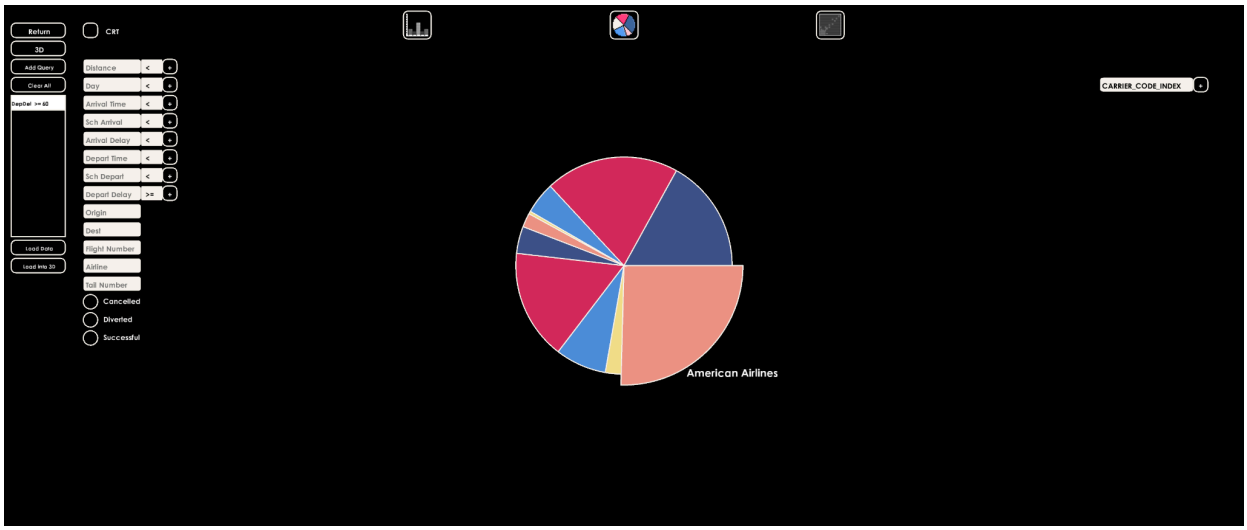
Screenshots:
Home screen, showing flights from London Heathrow in the world dataset (March 2024)



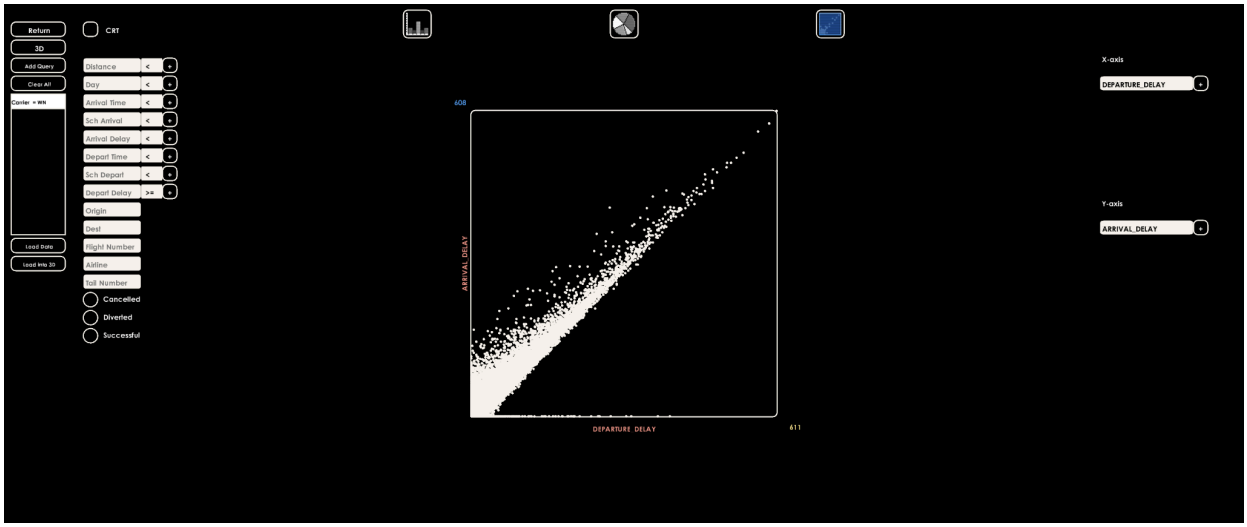

Histogram showing which airlines operated flights from New York JFK to Los Angeles LAX in the US dataset (July 2023)




Pie chart showing which airlines had the most flights delayed by more than 60 minutes on departure in the US dataset (July 2023)



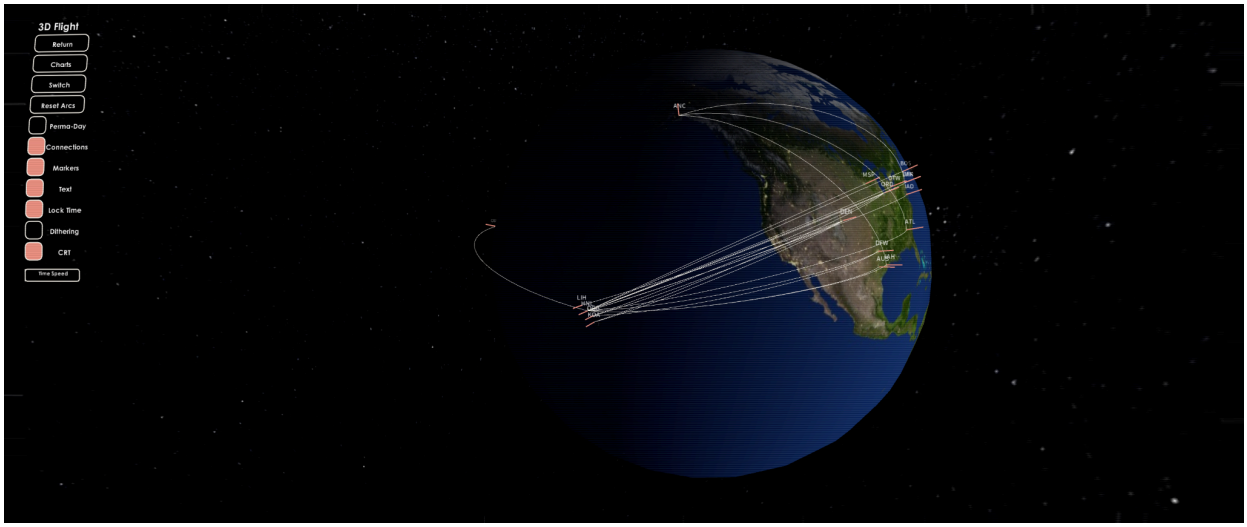

Scatter plot showing departure delay vs arrival delay for Southwest flights in the US dataset (July 2023)



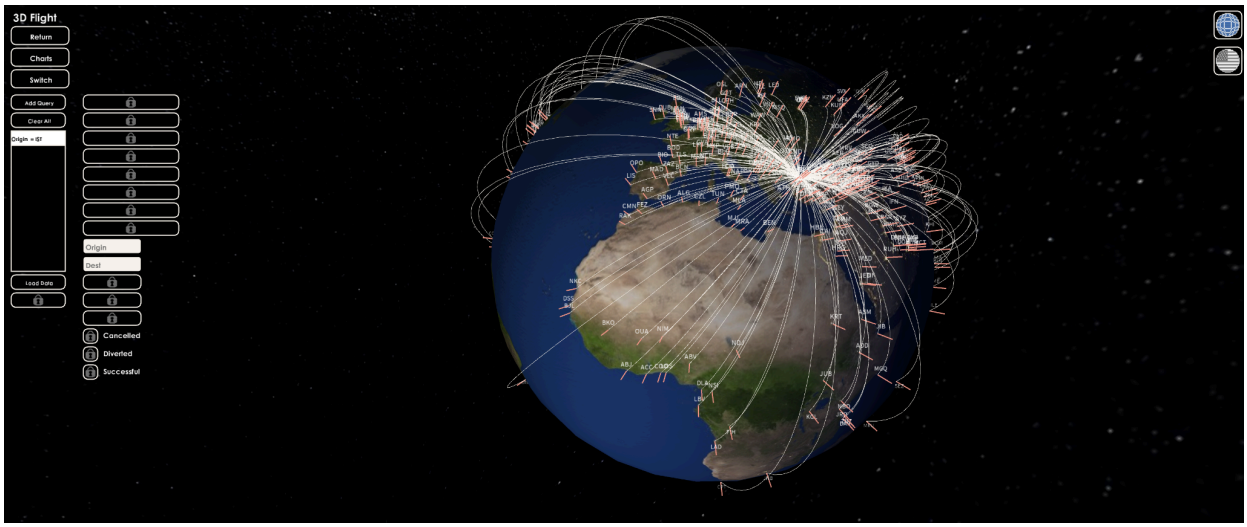

3D globe showing all flights operated by N10156 (a United Airlines Embraer 145 regional jet) in the US dataset (July 2023)


3D globe showing all flights under 250 km in the US dataset (July 2023)


3D globe showing all flights over 5000 km in the US dataset (July 2023), with CRT filter



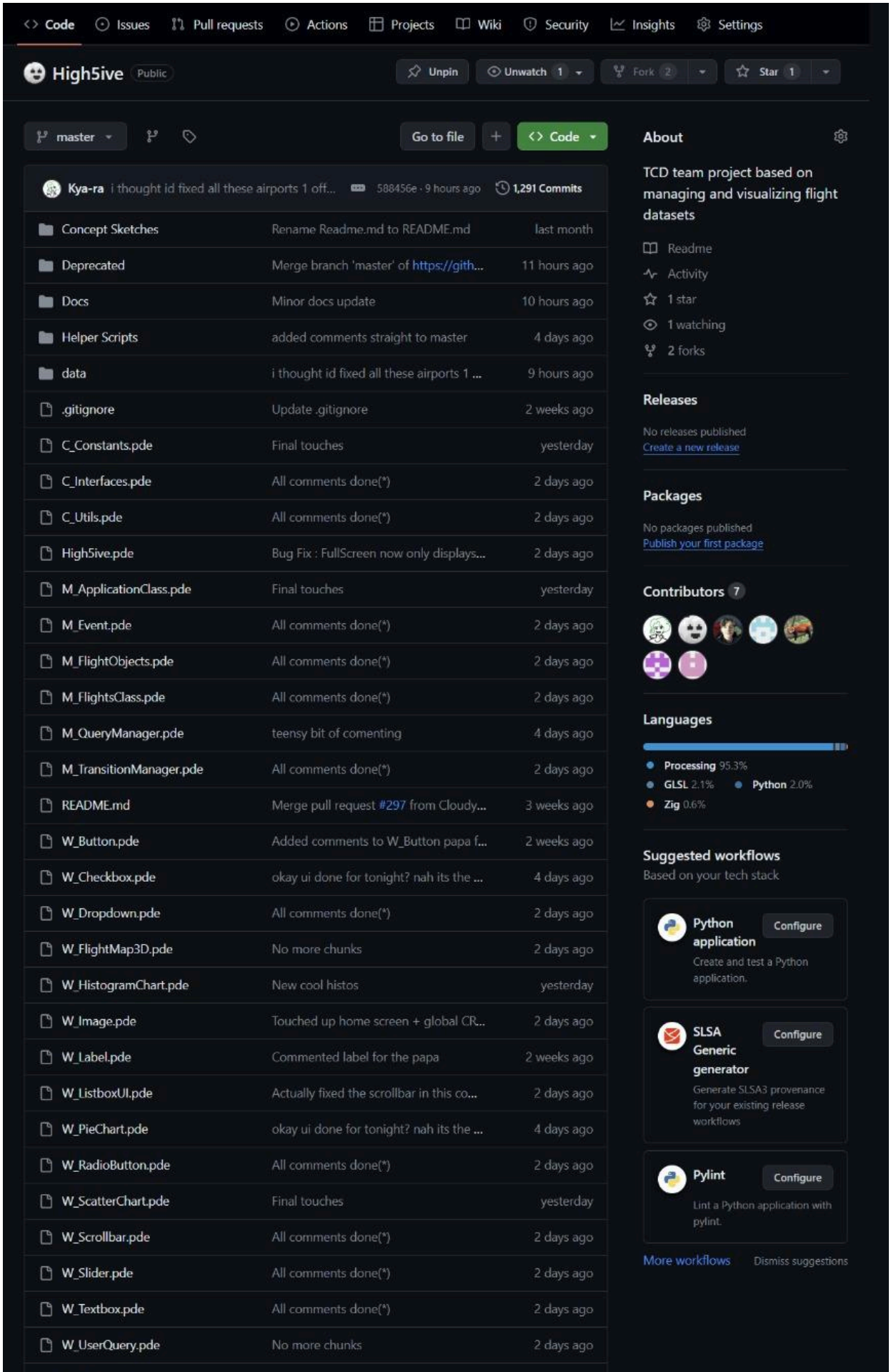
3D globe showing all flights from İstanbul Havalimanı IST in the world dataset (March 2024)




Git repository activity over the last month


Git repository commit counts since start of project


Overview of Git repository on Github


Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
High5ive
Public
Unpin
Unwatch 1
Fork 2
Star 1
master
Go to file
Code
Kya-ra i thought id fixed all these airports 1 off... 588456e · 9 hours ago 1,291 Commits
Concept Sketches Rename Readme.md to README.md last month
Deprecated Merge branch 'master' of https://gith... 11 hours ago
Docs Minor docs update 10 hours ago
Helper Scripts added comments straight to master 4 days ago
data i thought id fixed all these airports 1 ... 9 hours ago
.gitignore Update .gitignore 2 weeks ago
C_Constants.pde Final touches yesterday
C_Interfaces.pde All comments done(*) 2 days ago
C_Utils.pde All comments done(*) 2 days ago
High5ive.pde Bug Fix : FullScreen now only displays... 2 days ago
M_ApplicationClass.pde Final touches yesterday
M_Event.pde All comments done(*) 2 days ago
M_FlightObjects.pde All comments done(*) 2 days ago
M_FlightsClass.pde All comments done(*) 2 days ago
M_QueryManager.pde teensy bit of comenting 4 days ago
M_TransitionManager.pde All comments done(*) 2 days ago
README.md Merge pull request #297 from Cloudy... 3 weeks ago
W_Button.pde Added comments to W_Button papa f... 2 weeks ago
W_Checkbox.pde okay ui done for tonight? nah its the ... 4 days ago
W_Dropdown.pde All comments done(*) 2 days ago
W_FlightMap3D.pde No more chunks 2 days ago
W_HistogramChart.pde New cool histos yesterday
W_Image.pde Touched up home screen + global CR... 2 days ago
W_Label.pde Commented label for the papa 2 weeks ago
W_ListboxUI.pde Actually fixed the scrollbar in this co... 2 days ago
W_PieChart.pde okay ui done for tonight? nah its the ... 4 days ago
W_RadioButton.pde All comments done(*) 2 days ago
W_ScatterChart.pde Final touches yesterday
W_Scrollbar.pde All comments done(*) 2 days ago
W_Slider.pde All comments done(*) 2 days ago
W_Textbox.pde All comments done(*) 2 days ago
W_UserQuery.pde No more chunks 2 days ago
About
TCD team project based on managing and visualizing flight datasets
Readme
Activity
1 star
1 watching
2 forks
Releases
No releases published
Create a new release
Packages
No packages published
Publish your first package
Contributors 7
Languages
Processing 95.3%
GLSL 2.1%
Python 2.0%
Zig 0.6%
Suggested workflows
Based on your tech stack
Python application
Configure
Create and test a Python application.
SLSA Generic generator
Configure
Generate SLSA3 provenance for your existing release workflows
Pylint
Configure
Lint a Python application with pylint.
More workflows
Dismiss suggestions