EL BACH Achraf Trinity College, Computer Science and Statistics

Programming Project Report

CSU11013 – PROGRAMMING PROJECT

Summary:

Introduction

The goal of this project was to construct an application to explore data relating to artificial space objects. Using the General Catalog of Artificial Space Objects published by Jonathan C. McDowell, we had to parse a large amount of data in an appropriate format and create an environment to visualize it the best we could.

Data Storage and Data Retrieval

A class is responsible for both operations, its name is *FilteringTool*.

Data Parsing

To process data, we need to parse in a format that would facilitate the future operations. We instantiate the FilteringTool using the dataset file. During this operation, every line of data will be used to create a DataPoint which represent a space object. The important data (name, state, status, launch date, mass, diameter, perigee and apogee) will be stored in a complementary class call SpaceItem separate the data retrieval responsibilities from the graphical ones.

Data Filtering

During the parsing operation, we filter information that comes to use when the user interacts with the program: the list of states, minimum perigee, and maximum apogee. Once this operation is done, we can retrieve the list of objects with filter application or not. The dataset can be filtered by multiple attributes: name, state, date or distance. The filters are applied in the order there were entered. Raw data attributes are retrieved using basic accessors from the SpaceItem class.

Individual Data Display

Orbit Trajectory

Using the diameter, the perigee and the apogee, we can calculate the semi-minor and semi-major axis of the satellite orbit following these equations:

- Semi-major axis = (perigee + apogee + diameter) / 2
- Focus Distance (orbit center to earth center) = apogee – *Semi-Major Axis*
- Semi-minor axis = sqrt(Semi-Major Axis ² - Focus Distance ²)

Using the ellipse method of processing with these three variables, we can draw the orbit at the appropriate position with the appropriate size. Then we simulate the movement of the satellite object by calculating its relative position around the orbit through time (an incrementing angle with each frame) using the formulas:

- Horizontal Position = earthX + Semi-Major Axis * cos(angle)
- Vertical Position = earthY + Semi-Minor Axis * sin(angle)

Detailed Information

There is an additional informative window that is only responsible of showing the information of one object at a time. This window is used to display one selected item info in a more detailed format and a greater size. There also is a preview of the object appearance at the top of the window. The object is drawn using a circle of fixed size and is decorated by a crown of regularly spread rectangles. The number of rectangles is random which can give the aspect of a cross, a star or an asterisk. The same shape is used to represent the satellite object on the full canvas.

Collective Data Display

Full orbit system

On the main window, a small blue circle at the center to represent the Earth and the orbit of every object from the list are drawn once in a fictional image to save memory by reducing the number of instructions to one. Then, all the satellite objects are drawn and moved on each frame to give a dynamic aspect. Every object is attributed a random speed and a random direction when generated to give a heterogeneous look.

Object Listing

In another window, we have a listing of every object displayed with every attribute displayed in a small rectangle for preview purposes. Considering the fact that the list might be gigantic, there also is a scrollbar on the side of the window to move along the list. The use of the mouse wheel is possible to scroll along the list.

User Interactions

Orbiting Objects

The object selection was previously mentioned for the information window. To select a satellite object there are multiple ways. The most fun one is to directly click on the object. The user can click on the canvas and the closest object to their cursor will be define as selected. To facilitate this mechanism, once the mouse is pressed all the objects stops and the object is chosen once the left click is released. We also implemented a magnifying glass on top of the cursor along with a crosshair to have a more precise click. The selected object orbit will then be highlighted in pink.

List Items

Clicking on the canvas is good to discover new items but not relevant when we know which item we'd like to visualize. That is why the user can also click on the object list. The selected object will have a red border in the list to easily distinguish it from the rest.

Sort Button

On the object list there also is a button with a funnel design allowing the user to enter the sorting function screen. On the sorting function screen there are all attributes that can possibly be used to sort the list as multiple buttons, a Sort button, and a return button. The user can click the attributes buttons in the order the objects will be sorted (first by name and then by launch date per example). When the choice is made, the user can press the Sort button to be brought back to the display list with the sorted object list.

Filtering Panel

In another window, the user can filter the full dataset according to some attribute constraints. Using a dropdown list, the user can select one or many state from the state list retrieved at the beginning. Using a slider, the user can set the minimum and maximum distance from the earth needed to encapsulate one's orbit. Using a date picker, the user can enter two dates to encapsulate the object's launch date. After having all the attributes constraints set, clicking the filter button will result in a new filtered object list.

Conclusion

I have experienced a lot of administrative problems with this module since I never got assigned to a group and that I wasn't able to attend labs, but I honestly had an overall great experience doing this project. I had the opportunity to discover Processing and improve my Java skills. I managed to create something that feels like good enough.

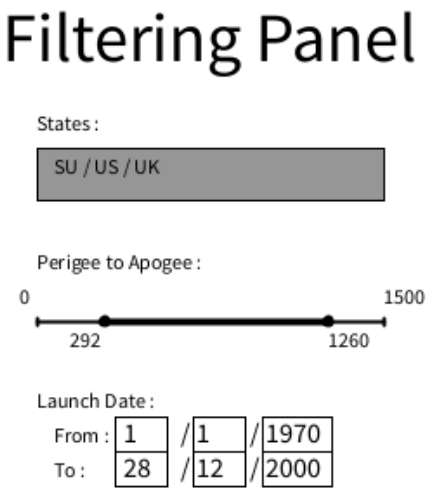
Sources:

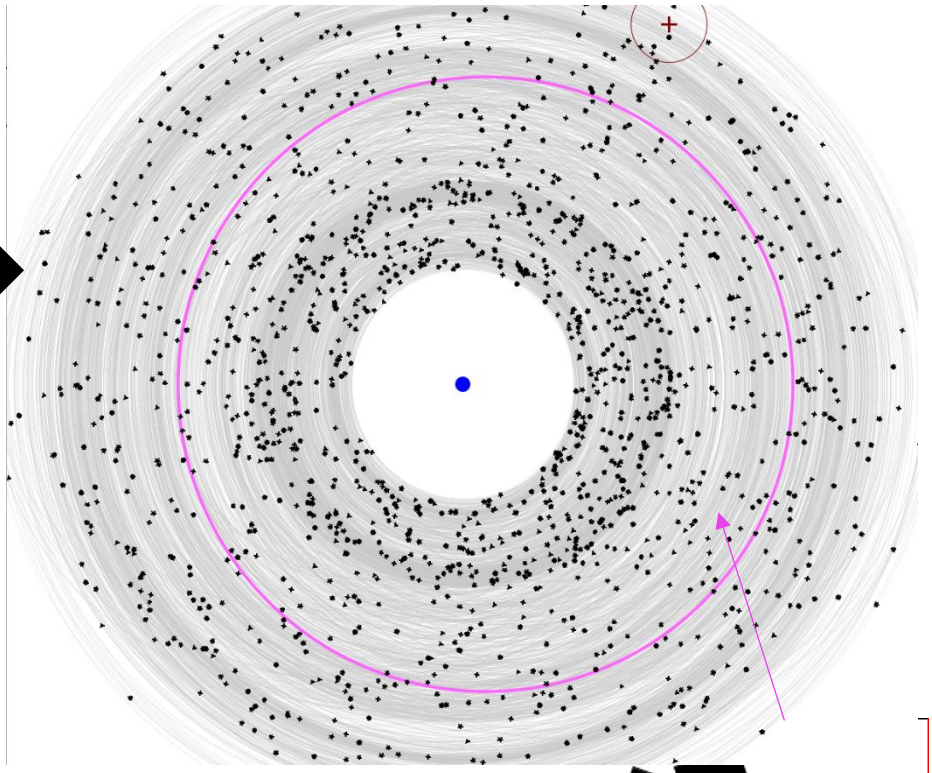
https://github.com/processing/processing/issues/1738	Alternative Mask Method
https://processing.org/reference	Processing API
https://stackoverflow.com/	For error fixing purposes

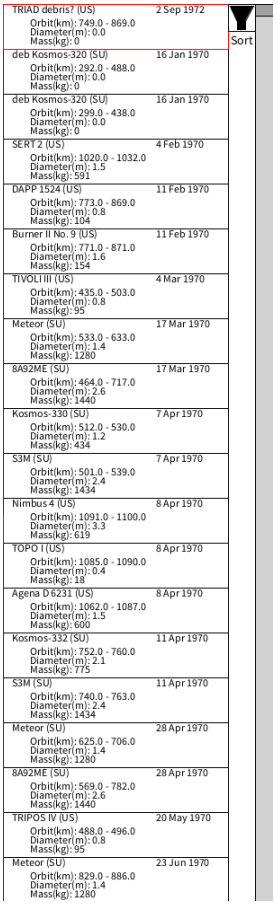
Screenshots :

User InteractionPanel

Main Canvas (full orbit system)







Object Listing

Information Window