CS1013 Programming Project 2022

1. Goal

The goal of the project is to construct an application to explore data relating to artificial space objects, such as satellites and debris. We will use a public dataset – the General Catalog of Artificial Space objects [1]. The application will read in data from a file "gcat.tsv", render it, and allow the user to interact with it. The data will take the format of *tab separated values*, which will contain a number of rows, each of which will describe one space object. Each object will be described by data in a number of columns (separated by tab characters - '\t' in Java). You may ignore most of the columns to begin with, and focus on the name, launch date, status, state, mass, diameter, perigee and apogee.

Each entry appears on a new line. Varying size datasets will be available from links on blackboard, and the ability to deal with these will depend on the efficiency of your program. You may make use of other datasets to add additional data such as launch catalogs. A full description of the fields is available at https://planet4589.org/space/gcat/web/cat/cols.html

Name – The name of the object – e.g. "Apollo BP-13"
Launch date - The launch date is given in UTC, with 1 day precision. Format is YYYY Mon DD e.g. 1959 Feb 7
Status – A text code representing the current phase of the object e.g. "O" for In orbit. "R" for re-entered.
State - The OrgCode for the country/state associated with the owner/operator. E.g. "US".
Mass – Mass of the object at launch in kg.
Diameter – Longest dimension of the main body of the object in metres.
Perigee – perigee height (closest to the earth) of the orbit in km.
Apogee – apogee height (farthest from the earth) of the orbit in km.

Fields may be empty, so it is best to program defensively.

2. Structure

The program will contain the following components:

- code to read in the data from a file and place it in classes.
    - Processing provides both `loadBytes` and `loadStrings` commands. The `split` method should also be helpful.
    - A simple (although not particularly efficient) solution would be to define a DataPoint class which represents a single space object. There would be one instance of the class for each entry in the input file.
    - You may wish to store the data in a format which is more efficient to access.
- code to select a subset of this data.
    - Not all the data will be shown on the screen at one time, and so a set of queries must be defined in your code. At a minimum, the following queries should be implemented:
        - Objects associated with a particular country (state).
        - Objects within a certain altitude (perigee/apogee) range
        - Objects sorted by launch date.
- code to draw the data to the screen.
    - The results of each query will need to drawn on the screen.
    - You are encouraged to use graphical representations where appropriate (eg. the data could be on a barchart, a 2D or 3D representation of orbits, or more complex visualisations such as heatmaps).
- code to handle user commands.
    - Selecting what data is to be displayed (the query), the country name, mass, launch date range, etc.
- code to put everything together
    - You are advised to have an outline of this as early as possible (first week of project).

3. Assessment

The project marks will be allocated according to both your individual effort and the effort you put into the group. This is a group project, and part of the project is to manage the group effort. Each individual will receive a mark based on their own contribution to both the individual and group components**. If you make no contribution to individual or group tasks, you will receive no marks**.

You will be required to submit and demonstrate the current status of your project **every week** until the end of the semester. **Half** of the project marks will be allocated for the **weekly submissions**. DO NOT MISS THE LABS.

Hence it is not possible to leave the project until the end of the semester.

Code is to be submitted via the subversion revision control system. All code must be commented, and the authors indicated. You **must** use comments to indicate revisions to the code (eg. " M. Jones, Added Graph class for displaying results, 8pm, 10/3/2020". "J. Smith, Updated to show the dates on the chart, 7pm 14/3/2020", "L. White, Fixed bug which stopped user from going back to homepage, 9pm, 15/3/2020". etc.). It is in your own best interest to get credit for the code you have written!

Attendance at the labs is **mandatory** and you will be expected to work with your team during the labs. The project will take more time than is available at the labs. Remember that CS1013 is marked **only on coursework**. You will need to schedule time in your groups outside of lab time to work on your assignment / divide up workload.

The assessment will focus on the demonstration of working code; a check on code authorship; a check on quality of code and documentation, progress towards the overall goal of the project, and the features implemented. Your code is to be accompanied by a short report of **maximum 4 pages** (plus screenshots) outlining your design and any ways in which your solution goes beyond the original project brief.

You will present your project at the lecture in the last week of term. The best team, as judged by the panel, will win a prize for Best JF Programming Project. If you wish to be considered for best project you must record a version of your demo as a 5-minute video and upload the video to subversion.

4. Assessment Schedule

The following is the schedule for the project, including milestones to demonstrate:

Week 6	Project Handout	Plan project. Start thinking about it. Outline main program.
Week 8	File reading and code outline	Demonstrate reading of data from file and give outline of main program, including major classes, sketch of screens.
Week 9	Initial rendering of data	Graph and textual output of data.
Week 10	Data selection and rendering	Apply user queries to data set and present results on screen.
Week 11	User Interaction	Select different options and change screens
Week 12	(14th April) Demo of project (15th April) Final Submission	Full demo to panel and submission of report – **5 pages MAXIMUM**.

The project demos will be during the normal lab slots on 14th of April. Report due on Friday 15th.

5. Frequently Asked Questions.

- You **may** use multiple tables. You may use additional datasets. You may pre-process your data into files.
- You may **not** use an alternative development environment (IDE) **without permission**. Such permission will **only** be granted if **all** team members indicate they are happy to do so, and I am convinced that all team members are fully engaged with the project.
- You **must** use Subversion rather than any other revision control system. It is used to track individual contributions.
- If you are using SQL, you must implement a version which uses a local database (with a smaller dataset) so that your program can be checked independently. Your program **must** work when downloaded from SVN to a lab machine.

[1] McDowell, Jonathan C., 2020. General Catalog of Artificial Space Objects, Release 1.2.7 , https://planet4589.org/space/gcat