An Enhanced Visualizer for Future Space Population

Student Name: Luyang Han

The Institution Name: University College London (UCL)

The Year of Submission: 2021

Supervisor Name: Santosh Bhattarai

Word Count: 11298

Submitted in partial fulfilment of the requirements of the degree of MSc in Geospatial Sciences (Geographic Information Science and Computing), University College London

Abstract

The population of man-made objects on orbit will drastically increase over the next several decades. This unprecedented rate of space population growth is primarily driven by the great intensity of space race for next-generation, more reliable and high-speed Internet services with coverage of being as wide as possible including hard-to-reach areas. Thus, a large number of satellite mega-constellations, comprising thousands of satellites will be launched by high-tech companies such as SpaceX and Amazon. A UCL research team has focused on the future space population study, developing a simulator for future space population catalogue and a baseline visualizer to display the future space objects.

An enhanced visualizer for future space population based on the UCL baseline visualizer has been developed in this study in attempt to improve the visual performance of the baseline visualizer through personal enhancements. The enhanced process is also supported by CesiumJS library file and future space catalogue data.

The results have shown that compared to the baseline visualizer, the enhanced visualizer successfully enhance the visual performance with each personal enhancement. In addition, it has been distinctly recommended that limitations be brought into consideration for further research.

Acknowledgements

Firstly, I'd like to express my sincerely gratitude to my supervisor Dr Santosh. During the pandemic, I faced many problems, but Dr Santosh's patience and encouragement make me more confident to this research project. With his guidance and suggestions, I am inspired to finish this difficult project and dissertation.

Secondly, thanks to UCL future space study team, they provide the future space catalogue and UCL baseline visualizer which give me the motivation to develop an enhanced visualizer for future space population.

Also, thanks to my family and my friends in GIS and Computing, they all encourage me a lot to overcome the tough time during the pandemic.

Table of Contents

Abstract....2

Acknowledgements....3

List of Figures....7

List of Tables....8

List of Abbreviations....9

1. Introduction....10

    1.1 Space Population....10

        1.1.1 Definition of Space Population....10

        1.1.2 Current Space Population....10

    1.2 Research Context....13

    1.3 Motivation....14

    1.4 Research Aims and Objectives....14

    1.5 Report Structure....16

2. Literature Review....17

    2.1 Introduction....17

    2.2 4D and Browser-based 4D Visualization of Space Objects....17

        2.2.1 The CelesTrak Visualizer....19

        2.2.2 UCL Baseline Visualizer....20

    2.3 Principles of Visualization....21

3. Data....23

    3.1 Satellite Catalogue (SATCAT) File....23

    3.2 Future Space Population Catalogue (FSPCAT) File....23

4. Enhanced Future Space Population Visualization System....25

4.1 Overview of The Enhanced Visualization System 25
4.2 Baseline Methods 27
4.2.1 CesiumJS library.. 27
4.2.2 Methodology in Coordinate Transformation 27
4.2.3 Baseline Functions in UCL Visualizer 30
4.3 Enhanced Module 1 – Original User Input Module for Radar (Telescope) Position 33
4.3.1 Enhanced Process.. 34
4.3.2 Visualization Principle 36
4.4 Enhanced Module 2 – Enhanced Clock and Timeline Controller 36
4.4.1 Enhanced Process.. 37
4.5 Enhanced Module 3 – Enhanced Radar Viewer 38
4.5.1 Enhanced Process.. 39
4.6 Enhanced Module 4 – Statistical Analysis Module 41
4.6.1 Enhanced Process.. 41
4.6.2 Legend and Counter for Satellites by Operational Status 41
4.6.3 Visualization for Counted Number of Satellites by Countries Using Pie Chart 43
4.6.4 Visualization for Counted Number of Satellites by Categories Using Bar Chart 45
5. Results 46
5.1 The Enhanced Visualizer 46
5.1.1 Catalogue Controller Module 49
5.1.2 Enhanced Clock and Timeline Widget 49
5.1.3 Enhanced User Input Module and Radar Viewer.................... 50
5.1.4 Enhanced Analysis Module.. 51

5.2 Test for the Visualizer..53
6. Discussion 53
6.1 Discussion of Enhancements and Result..53
6.2 Discussion of Enhanced Process 55
6.2.1 Three Identifiers..56
6.2.2 Visualizing Strategy 57
6.3 Discussion of the Visualizing Principles 59
6.4 Limitations...60
6.5 Further Development 61
7. Conclusions 62
References 64
Appendices 70
Appendix 1: Guidance for Using the GitHub File 70

List of Figures

Figure 1: Increasing trend of active satellites (Jonathan, 2021).................... 11

Figure 2: The CelesTrak visualizer 20

Figure 3: UCL baseline visualizer 20

Figure 4: Process of enhancing the visualization of future space population26

Figure 5: Keplerian parameters (Montenbruck, 2000) 29

Figure 6: Camera Scene (Learn WebGL, 2016) 32

Figure 7: The enhancement from the user input module to the visualizer 35

Figure 8: The original design of the user input module.................................36

Figure 9: Process of enhanced radar viewer 39

Figure 10: Analysis module in the enhanced visualizer 41

Figure 11: Visualization of predicted 2023 space population 47

Figure 12: Visualization of predicted 2028 space population........................47

Figure 13: Visualization of predicted 2043 space population........................47

Figure 14: Guide steps for users to use the enhanced visualizer 48

Figure 15: Visualization of catalogue controller GUI 49

Figure 16: Visualization of enhanced clock and timeline widget 49

Figure 17: Visualization of enhanced user input module and radar viewer...51

Figure 18: Overview of the enhanced analysis module 52

Figure 19: Pie chart for statistical data 52

Figure 20: Bar chart for statistical data 52

Figure 21: Info box for visualizing the information of objects 62

List of Tables

Table 1: Categories of satellites by main operations (Mohanta, 2021) 12

Table 2: FSPCAT fields 24

Table 3: Matched time information for the catalogue and JulianDate input .. 37

Table 4: Fields of operational status and its representation 42

Table 5: Pie chart versus bar chart (Chartio, 2016) 44

List of Abbreviations

SATCAT = Satellite Catalogue

FSPCAT = Future Space Population Catalogue

RCS = Radar Cross Section

1. Introduction

1.1 Space Population

1.1.1 Definition of Space Population

Space population is defined as the number of space objects in space environment (Garcia,2017). In this study, space population regards to population of man-made space objects, which is defined as tangible objects launched into orbit including active objects and inactive objects such as space debris.

Active objects are artificial satellites launched into orbit and still in orbit to provide functional services. Based on the applications and provided services, satellites can be categorized into different types, for example, communication satellites, astronomical satellites, earth observation satellites, weather satellites, space stations, etc., (Omicsonline, 2016). According to the location of circular (geocentric) orbit, satellites can also be classified into low earth orbit (LEO), middle earth orbit (MEO) and geostationary orbit (GEO) (Montenbruck, 2000; Esa.int, 2019;).

Inactive objects in space are those artificial objects which no longer provide service. For example, these objects can be non-operational or abandoned launch vehicle compartments and spacecrafts. They can also be the debris from rocket body or satellites. Debris larger than 10 cm can be measured by current systems but objects smaller than 10 cm will pose a potential threat to operational objects (Adushkin, 2020).

1.1.2 Current Space Population

Many satellites have been continuously launched into space orbit in recent years by governments, organizations and commercial companies, especially over the last decade, in order to capture useful and accurate data to serve a large number of missions such as scientific research, meteorological monitoring and forecasting, military surveillance, positioning and navigating (Mohanta, 2021).

According to Union of Concerned Scientists (UCS), by July 2021, the total number of satellites in space is about 7500, among which 4500 are active and about 3000 in-active. However, the report from UCS by 1st January 2021 shows 3372 satellites are active and 3170 satellites are inactive, which indicates that there are almost 1100 more active satellites since last 7 months. With Figure 1 from Jonathan's Space Page and the statistical data from UCS, it is significant that the number of active satellites dramatically increases during the recent five years, especially in 2020 and 2021, which is in coordination with the ballooning number of satellites launched in recent years.

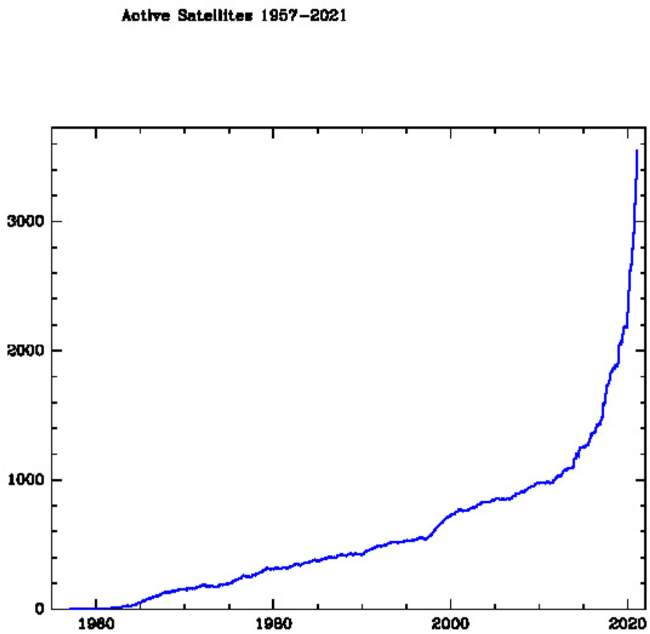

Figure 1: Increasing trend of active satellites (Jonathan, 2021)

The first reason for the increasing number of satellites launched in recent years is that the intensifying trend of dependence of various industries all over the world on different satellite data and services (Mohanta, 2021). Different types of satellites serve different operations while some satellites focus on their main operations of gathering series of data while others serve multiple operations. Recorded by UCS, Table 1 below shows the satellite number and their main operations by 1st January, 2021.

Table 1: Categories of satellites by main operations (Mohanta, 2021)

Number of satellites (total 3372)	Main operation
1832	Communication
906	Earth Observation
350	Technology development
150	Navigation and positioning
104	Space science and monitoring
20	Earth science
10	Other operation

Secondly, the emergence of the small sized CubeSat which means that a great number of such small sized satellites can be launched into space at the same time which leads to the exponential increase of satellites. It is generally agreed that in previous years one rocket once could only launch at most 2 satellites. According to the record from Nanosatellite and CubeSat Database, more than 1400 smaller sized satellites have been launched into space orbit (Millan et al., 2019).

1.2 Research Context

As stated in chapter 1.1, the increasing space population in recent years results in more congested space environment (McDowell, 2020). With the advancement of science and technology, the space population on orbit will drastically increase over the coming decades as companies such as SpaceX and Amazon race to launch satellite mega-constellations, comprising thousands of satellites (Beck, 2021; Ziebart, Bhattarai et al., 2019). This unprecedented rate of space population growth is primarily driven by a demand for next-generation, high-speed Internet services. On the other hand, the base station and signal tower is still formidable to build in hard-to-reach areas (villages, airborne aircrafts, marine vessels at sea, mobile homes, etc.), let alone good quality signal coverage in these areas. As a result, mega-constellations have been proposed as a solution to the problem of providing quality internet coverage in hard-to-reach areas. However, the thousands of planned satellite mega-constellations are still considered a key factor which helps to aggravate the congestion of space environment (McDowell, 2020).

Moreover, the inactive objects including debris in the space can cause potential danger to space environment in future (Tegler, 2021). Dating back to 1996, an article published by the UK Parliament shows that a continuous deterioration in near-earth space has been caused by the increasing number of debris and inactive satellites (impacts on earth from space, 1996). In recent researches, the risks are again indicated and emphasized, e.g., debris can cause various types of on-orbit collisions with operational satellites, or even catastrophic break-ups to space-crafts (Sotskiy et al., 2017; The world must cooperate to avoid a catastrophic space collision, 2021), inactive satellites can also interfere the designed functions of other active satellites (Tarran, 2021).

In conclusion, future space population will continue to increase and inactive objects will cause potential risks in the next few decades.

1.3 Motivation

The information of increasing space population should be displayed by some methods. The first motivation of this research project is to depict the future space population by visualization on browsers to users in order to gradually raise the public awareness of the increasing future space population. Secondly, the study concept of combining the future space data and baseline visualizer in order to develop an enhanced visualizer to provide enhanced visualization of future space population is motivated by the UCL future space population research team, who focuses on the future space population simulator and UCL baseline visualizer. Thirdly, developing an enhanced visualizer for future space population with the enhanced function as a decision-support tool, and as a foundation for further development constitutes another motivation.

1.4 Research Aims and Objectives

As the motivations stated in chapter 1.3, this research project focuses on the technology research and development. Therefore, the core aim is to combine the future space catalogue file, UCL baseline visualizer and CesiumJS with personal designed enhancements to facilitate the UCL space research team and other users understand that the enhancements are significant for the development of future space population research. The following questions need to be answered:

1) What enhancements can be developed for UCL baseline visualizers?
2) Why enhanced visualizers can be significant to users?

The core aim is supported by two main objectives.

1) To develop enhancements through combining baseline resources for

presenting a better visualization of space objects. This objective can be separated into four detailed secondary objectives.

a) To combine some developed functions and methods from UCL baseline visualizer to enhance the visualization of future space population.

b) To apply useful visualization principles which are critical analyzed by some literature reviews to support the enhanced visualization e.g., 1) the enhanced visualizer use different styles (including color and size) to visualize different space objects categorized by radar cross section, operational status and other options on radar viewer and main viewer via CesiumJS; 2) some visualizing principles provide a guide to users or present a better visualizing performance to users.

c) To provide some user customized options to visualize the space population. For example, the app should assist the user to set the latitude and longitude of the telescope position on earth surface in order to visualize different radar views by users' own needs.

d) To focus on some details and settings for widget ignored by the UCL baseline visualizer and to improve or solve them in the enhanced visualizer.

2) To achieve the enhancements on using statistical analysis tools to analyze the future space population and visualize the statistical results on screen. This objective is supported by 3 detailed secondary objectives.

a) To display the statistical result for LEO, MEO and GEO categorized by the altitude above the earth surface via statistical analysis tools.

b) To display the statistical result of objects launched by different countries on screen via statistical analysis tools.

c) To display the statistical result of active space objects and inactive space objects.

1.5 Report Structure

Chapter 1: This section describes the context of this research project, the background of the space population, the research aims and objectives.

Chapter 2: This section presents some critical analysis of Literature review on 4D and browser-based 4D visualization of space objects and visualizing principles.

Chapter 3: This section describes the data – future space catalogue used in the visualization and analysis of the future space population in 2023, 2028, and 2043.

Chapter 4: This section introduces work flow, methodology and research process of the enhanced visualizer. It includes the key methods and functions used in the UCL baseline visualizer useful to this enhancement process and each enhancement based on the UCL baseline visualizer.

Chapter 5: This section focuses on the results of enhanced visualizer and the visualizations it provides. Each enhancement in this research project will be indicated.

Chapter 6: This section discusses the results in section 5 and reviews the research aims comparing to the provided results. The research process is reviewed as well. Then, the limitations of this research project are figured out. Finally, the further development based on the enhanced visualizer is stated.

Chapter 7: The final main body chapter draws the conclusions derived from the previous chapters.

2. Literature Review

2.1 Introduction

The related literature and researches have been critically reviewed which consists of two sections, literature review of 4D visualizing technology in simulating the space environment and browser-based 4D visualization applications for space objects in chapter 2.2 and principles of visualization in chapter 2.3.

2.2 4D and Browser-based 4D Visualization of Space Objects

In recent years, 3D visualization technology has been widely used in space research which require 3D scenes (Han, 2010). With the fourth dimension – time added to the 3D visualization, the 4D visualization technology has also been used in some visualizing applications (Kaur, 2017). In space research, as the time is an important dimension to visualizing the moving space objects, 4D visualization technology is widely employed to display the space objects with time dimension. In addition, browser-based 4D visualization applications are developed on the basis of 4D visualization technology to visualize the space objects and space environment to public.

By modelling the spatio-temporal data of space objects, Qing (2011) visualized the moving space objects based on 4D visualization technology and simulated the near-earth space environment by modelling but the key research was conducted via modelling to simulate the current space environment rather than developing a browser-based visualizer. Möckel et al. (2011) used modelling algorithms to simulate the space debris environment and displayed the space environment based on 4D visualization technology. Similar to Qing (2011), this research focused on modelling current space debris rather than developing a browser-based visualizer. Yang (2012) used a webGL-based method via data changes to simulate the dynamic environment, this research also focused on the visual simulation of the space environment. In order to simulate the space junk existed in the space environment from the 1957 to 2016, Grey (2015) used the 4D visualization technology to indicate space environment full of space junk, and to claim that more than 90 percent of all tracked objects are inactive or debris. Like previous research, the key focus was not on developing browser-based visualizer.

With the existence and fast development of some open-sourced 4D visualization libraries like CesiumJS, some browser-based 4D visualizers for space objects have been widespread to the public. Maiti et al. (2018) developed an open-sourced browser-based 4D satellite tracking and visualization application using current satellite catalogue data. The browser-based 4D Astriagraph's visualization software program was started in 2017 to track the orbital position and speed of the man-made objects in Earth's orbit in order to keep those on-orbit satellites away from the threaten of debris or space junk (Moriba, 2017). In 2018, the Astriagraph space object visualizer displayed the space congestion in Earth orbit especially the low-Earth orbit, while the real-time information between on-orbit satellites and space debris is also provided to alert those near-collisions (Astriagraph' software program combats space junk, 2018). CelesTrak orbit Visualizer was developed by TS Kelso team, which propagates the public satellite catalogue, and provides the visualization of all space objects launched since 1957 with update data (CelesTrak Orbit Visualization, 2021).

However, the visualizers referenced above are the presentation of the current or previous space population. For the future space population, the relevant research is yet to comprehensively carry out. A UCL team started the future space population program in 2019 with plans to visualize the space objects by browsers and now the simulator of the space objects (exclude debris) has been built to visualize and analyze the active satellites and inactive satellites in the future space environment (in 2023, 2028 and 2043) but the visualizer – UCL baseline visualizer is not completed and need to be enhanced and further developed (Ziebart, Bhattarai et al., 2019).

In the following sections, two browser-based 4D visualizers will be stated as two examples.

2.2.1 The CelesTrak Visualizer

The 4D CelesTrak Visualizer shown in Figure 2 was developed by T.S. Kelso team using the satellite catalog database maintained by T.S. Kelso to visualize the space objects from 1957 until now (celestrak.com, 2021). The satellite catalog was maintained by satcat.txt on celestrak.org and the browser-based visualizer presents some core information on the page (celestrak.com, 2021). For current version of the visualizer, it uses the TLE format to obtain the provided general perturbations orbital data (GP) to ingest data into Simplified General Perturbations 4 orbit propagator (SGP4).

The visualizer was developed using Cesium JS and online database (the space object catalogue data source is stored in an online database) and it is a professional visualization software of the space objects, using different colors to show the active objects, inactive objects and debris. In the visualizer, it provides detailed information for catalogued objects when clicking the satellite catalogue option (celestrak.com, 2021).

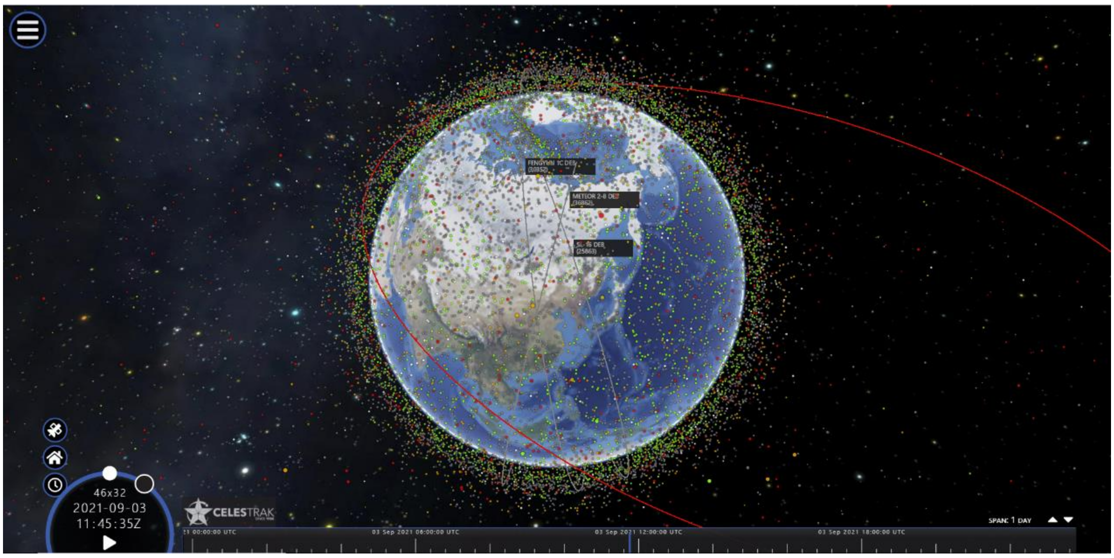

Figure 2: The CelesTrak visualizer

2.2.2 UCL Baseline Visualizer

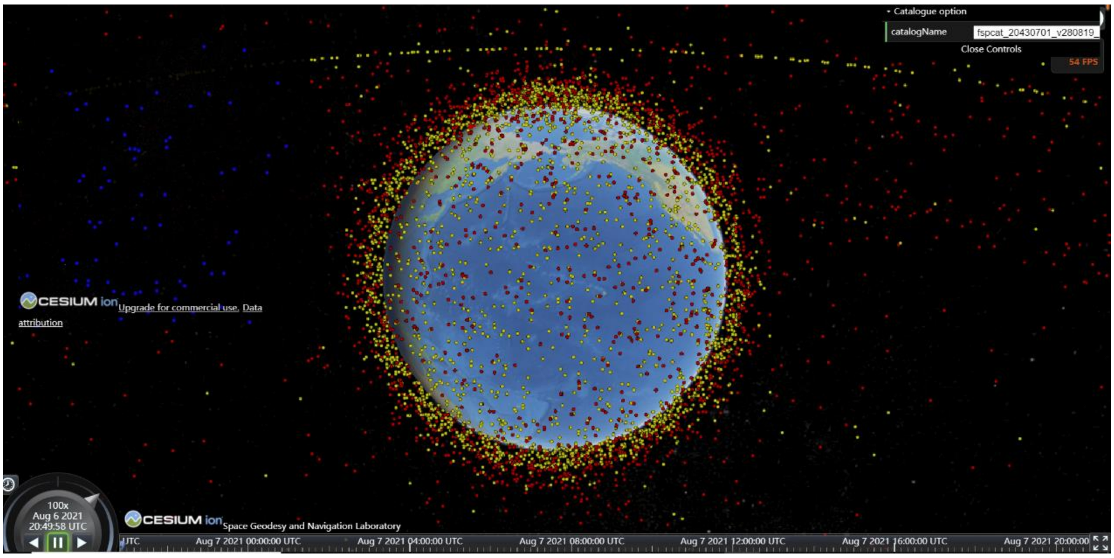

Figure 3: UCL baseline visualizer

The UCL baseline visualizer shown in Figure 3 is the foundation of the improvement and focus of this study, which was initiated by the author Zhen Li from UCL future space population research team in 2019 (Ziebart, Bhattarai et al., 2019). The baseline visualizer, a browser-based 4D visualization application, provides some basic functions and inspiration of the enhancements for the enhanced visualizer (Ziebart, Bhattarai et al., 2019). The data file for the visualizer is generated from the FSPCAT simulator developed by UCL future space population research team. Baseline visualizer merely provides limited visualizations with a radar viewer and a main viewer. The enhancements for the baseline visualizer will be stated in chapter 4.

2.3 Principles of Visualization

For this research project, one task is to enhance the UCL baseline visualizer by applying some visualizing principles. The main goal of visualization is to help the visualizer manage the data and display the data with better performance, which can indicate the relationships between data and visualized key information. The visualizing principles will be critically stated in the separated sections.

- Balance

Balance means that elements of visualizing such as shape and color, can be similar but not necessarily equally in each visualizing section. Three different types of balance are commonly used for visualization balance: symmetrical – both side for the visualizer is set as the same with each other; asymmetrical – both sides for the visualizer are different but with similar visualizing weight; radial – elements are placed around a central visualizing part (Tufte, 2002; Chen, 2010). These three different methods can be used for different purposes. In this study, this principle of balance will be considered as to which type of balance is used in order to balance the visualization of each part.

- Emphasize the key section and visualizing information

Emphasize means the visualizing design should not only emphasize the most significant section but also in each section to focus on the key information to visualize (Tufte, 2002). It is suggested that a certain level of caution should be taken when comes to choose meaningful color, associated text and visualized position in order to effectively attract the user's attention and make them focus on the key visualizing information (Upson, 1989; Williams, 2000). For example, chart is often used to emphasize the statistical data, the graphic is the main visual part, however, associated text is crucial to describe the graphics, without it, the chart could not make sense (Streit, 2014); as human eyesight has only a narrow range of focus, graphics should be placed very near their associated text in order to make them as a whole to emphasize a meaningful chart (Malamed, 2009; Ware, 2012). If the associated text could not combine with the graphic well, the emphasis on the chart is not a success which means it would distract the users instead of emphasizing the key information (Ware, 2013). The emphasis also refers to the position on the screen. In general, when the user enters the visualizer, the top-left corner is the first position to be focused on (Evergreen & Metzner, 2013). Therefore, it makes sense to add more important information or visualizing module there to quickly catch user's eyes. If the key information cannot be effectively conveyed to the users by the visualizer, thus some improvement is definitely needed in terms of the design of the visualizer.

- Illustrating movement

Movement guides the user's attention in a certain direction which means the visualization modules or elements should lead the users' eyes in a certain direction e.g., clockwise from top left or bottom left and eventually back there (Tufte, 2002). Alternatively, the design can be made by a main key element, and circled with several accessorial elements with a certain visual direction (Midway, 2020). In a sum, this design principle is working as a guidance for users to familiarize them with the visualizer, therefore, the design should be clear enough to make users know the visual direction.

- Proportion

The proportion is reflected by the size of each element, in visualization, the proportion or size of the element indicates the relationship between each visualizing section, and also represents the significance in the whole visualizing page (Chen, 2010).In this research project, the size of each visualizing module and element to decide each module's relationship and contents of display have been considered.

3. Data

3.1 Satellite Catalogue (SATCAT) File

The satellite catalogue is defined as a file or database which contains the detailed information of observed objects in space environment especially the near-Earth space environment (Ziebart, Bhattarai et al., 2019).

3.2 Future Space Population Catalogue (FSPCAT) File

This file, generated from the future space population simulator developed by Dr Santosh, Dr Ziebart et al. from UCL future space population research team, origins from the latest satellite catalogue (SATCAT). The future space population research project uses the Library of UCL Orbit Dynamics Library (UCL-ODL), which develops a new class – Space Object Catalogue (SOCAT), so the satellite catalogue can be alternatively defined as space object catalogue. The future space catalogue file (FSPCAT) used in this research project is generated from the future population simulator, which process the combination of the latest CelesTrak SATCAT data field information and the SOCAT future space population information (Ziebart, Bhattarai et al., 2019). For the future space catalogue, some fields from the latest SATCAT are deleted for designed product; the key designed fields are shown in Table 2.

The FSPCAT was first generated from the future space population simulator in csv format and modified as json format with the fields shown in Table 2. From the FSPCAT provided by the UCL research team, it focuses on year 2023, 2028 and 2043 (Ziebart, Bhattarai et al., 2019). Therefore, the research project focuses on these three time points.

Table 2: FSPCAT fields

Field Name	Field Description
COSPAR_ID	A unique Identified number contains launched information for each resident space object (RSO)
RSO_Name	A unique name for each RSO in FSPCAT
RSO_Type	The types of RSO are classified as satellites, space stations, debris and rocket bodies(Liu et al.2016).
Payload_operational_status	For this research project is defined as operational objects and non-operational but on-orbit objects.
Owner	This field represents the country who launch the object
Launch_Date and Decay_Date	The launch date and decay date referenced from TS Kelso's SATCAT
Apogee_hgt	The farthest point from earth according to the path depicted directly above the earth's surface to the satellites.

Perigee_hgt	The closest point from earth according to the path depicted directly above the earth's surface to the satellites.
Radar Cross-section (m^2)	A measure of how detectable an object is by radar, is used to adjust different size points to visualize in the research project.
Keplerian Elements (6 fields)	Six Keplerian elements to identify a specific orbit include: Semimajor axis, eccentricity, inclination, longitude of the ascending node (RAAN), argument of perigee, true anomaly.

4. Enhanced Future Space Population Visualization System

4.1 Overview of The Enhanced Visualization System

The enhancements of the visualizer for future space population are based on the UCL baseline visualizer created by Zhen Li from University College London (Li, 2019). The Figure 4 shows the process and working flow in this research project which indicates the comparison between the baseline visualizer and enhanced visualizer.

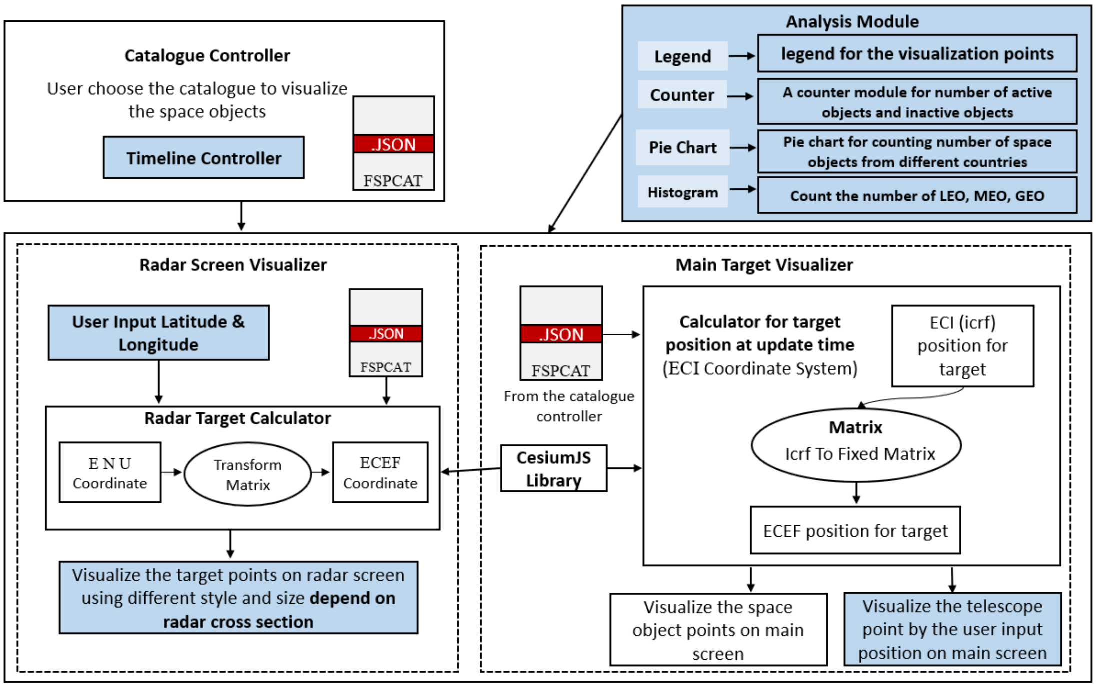

Figure 4: Process of enhancing the visualization of future space population

The parts with light blue background in Figure 4 are designed enhancements in this research project based on the UCL baseline visualizer. As the baseline of the enhanced visualizer, the UCL version originally creates three main modules or sections – Catalogue controller for loading the future space catalogue (FSPCAT); basic radar screen viewer providing the observations of the space objects in a range set by the camera parameters in CesiumJS; main screen viewer shows the globe 3D scene of the earth and the orbital space object points.

The process of enhancement is shown as follows：

- Modification of the visualization of the clock widget and timeline based on different catalogue in order to realize time control.
- Customized user input model for the telescope position (latitude and longitude) on earth surface.
- Visualization of the radar point by the user input position; enhanced

visualization of the space objects on radar viewer module by different styles depends on objects' radar cross section field.

- The original analysis module (based on the future space catalogue) includes the legend part for representing the object points, a counter for active objects and inactive objects that still in space environment, the pie chart for statistic number of space objects launched by countries and a histogram for calculating the number of LEO, MEO and GEO.

4.2 Baseline Methods

4.2.1 CesiumJS library

In this research project, choosing a JavaScript library with good performance is very crucial for developing an enhanced visualizer. The CesiumJS, an open-sourced 4D library, is suitable for creating the 3D global earth and space objects with time dimension as it supports the best performance of hardware acceleration and displaying dynamic data, and it provides the high precision for space objects (CesiumJS, 2021). The enhanced visualizer in this research project focuses on the space objects and view on the global earth, as the CesiumJS library is used in most current space object visualizers such as CelesTrak and Astria discussed in the previous chapters, so the foundation JavaScript coding library is set as CesiumJS in this research project.

4.2.2 Methodology in Coordinate Transformation

To achieve the objectives of the enhanced visualizer described in section 4.1, the process of the coordinate transformation is essential.

The Cesium.Cartesian3 class in CesiumJS is covered in the process of transformation, as Cartesian3 is a three-dimensional coordinate in the Earth-Centered Earth-Fixed (ECEF) coordinate frame, which is Cesium's default internal frame for rendering in 3D, the coordinates of the space object and the telescope position should be all in ECEF system. As the original coordinates of the space objects calculated by the formula are not explicit in ECEF system, the transformation method should be used for the process.

The transformation methods used in the research process include Earth-centered inertial (ECI) coordinate frame to Earth-centered and Earth-fixed coordinate (ECEF) frame; local East, North, Up (ENU) coordinate to ECEF.

- **ENU to ECEF**

This process is shown in the Radar Target Calculator in Figure 4. As the radar viewer is controlled by the Cesium.camera to show the observation, it is used to transform the camera position from ENU coordinates to ECEF coordinates. For the radar viewer, the position of the radar camera was originally set in the ENU coordinate system calculated by distance, azimuth and elevation, while the target coordinate is in ECEF system so transformation from the ENU to ECEF is essential. To achieve the ECEF coordinate, the method is using the provided Cesium function - Cesium.Transforms.eastNorthUpToFixedFrame to calculate the transformation matrix.

- **ECI to ECEF**

This transformation is used in the process of updating position of the space object where the Keplerian orbital elements should be used. The catalogue file provides the information of 6 Keplerian orbital elements with fields of "semi_major_axis", "eccentricity", "inclination", "RAAN", "argument_of_perigee" and "true_anomaly". The semimajor axis is the half of the sum of apoapsis distance and periapsis distance. The orbital eccentricity of an orbital object is the determination of the amount by which its orbit around earth deviates from a perfect circle. The inclination element represents the inclination of the planet’s orbital plane and the ecliptic plane. RAAN represent the longitude of the ascending node -the short name for the right ascension of the ascending node. True anomaly at an epoch defines the position of the orbiting body along the ellipse at a specific time - "epoch" (Jo, et al. 2011). Some of these elements are shown in Figure 5.

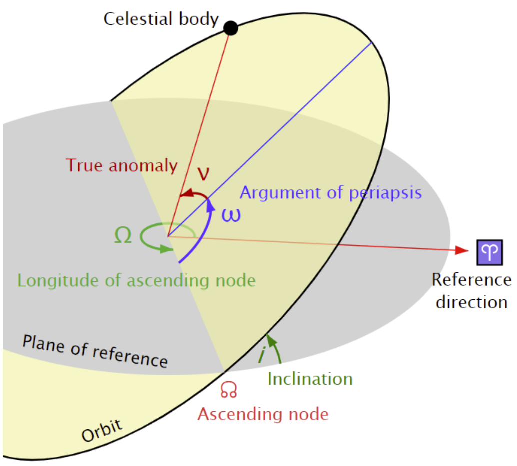

Figure 5: Keplerian parameters (Montenbruck, 2000)

These Keplerian orbital elements are used as 6 parameters to identify the orbits of celestial objects and the whole 6 elements compose the unique identifier of the position and velocity of the space object in a given reference frame, which can be used to calculate the cartesian coordinates (ECI). As the target coordinate system is ECEF, the process of the transformation should be separated into 2 stages (Jo, et al., 2011).

- Using the 6 Keplerian orbit element to calculate the Cartesian state vectors.
- Obtain the ECI coordinate of the space objects from state vectors.

These two stages have been realized by the functions coding in UCL visualizer which is described in section 4.2.3. The ECI coordinates are then transformed to ECEF by the transformation matrix calculated by the CesiumJS.

4.2.3 Baseline Functions in UCL Visualizer

- **compute_debri_position_eci(isat, time) function**

This function is used to first calculate the ECI coordinates for the space objects via the information of the Cartesian State Vectors. While the state vectors (position and velocity) are calculated by function getStateVector() in Baseline visualizer code (Li, 2019) from the 6 Keplerian orbital parameters, the Cartesian X, Y and Z coordinates in ECI system can then be assessed from the position state vectors. The code example below displays the process of getting space object's position at an updating time.

```
compute_debri_position_eci ( isat, time ){

    if(isat < this.debris_kep.length) {

    var idebri = this.debris_kep[isat];

    var positionAndVelocity={position:{x:0,y:0,z:0},velocity:{x:0,y:0,z:0}};

    var kep = new KeplerianElement();

    kep.setElements(idebri['semi_major_axis'], idebri["eccentricity"],

                    idebri["inclination"], idebri["RAAN"],
                      idebri["argument_of_perigee"], idebri["true_anomaly"]);

    var tt0 = idebri["epoch_of_orbit"];

    var time_diff = (time - tt0)/1000.0; /// in seconds

    kep.updateElements(time_diff);

     var pv = kep.getStateVector();

     positionAndVelocity.position.x = pv[0];//cartesian X
```

```
        positionAndVelocity.position.y = pv[1];//cartesian Y

        positionAndVelocity.position.z = pv[2];//Cartesian Z

        return positionAndVelocity;

            }

}
```

Code Example 1 – Example code used to calculate the ECI coordinate of the space objects at an updating time.

- **update_debris_position() function**

This function is essential for the enhancement of the radar viewer. It is created to calculate the ECEF position of the space object points at an updating time by using the returned ECI coordinate from the compute_debri_position_eci() function shown in code example 2 below. This function is also supported by some extra coding in baseline visualizer (Li, 2019).

```
var icrfToFixed = Cesium.Transforms.computeIcrfToFixedMatrix(time_utc);

for (var i = 0; i < length; ++i) {

      var point = points[i];

      if (Cesium.defined(icrfToFixed)) {

          var positionAndVelocity = mycatlog.compute_debri_position_eci(i,
time_date_js);

          var position_eci = new
Cesium.Cartesian3(positionAndVelocity.position.x*1000,positionAndVelocity.positi
on.y*1000,positionAndVelocity.position.z*1000);
```

```
        position_ecef = Cesium.Matrix3.multiplyByVector(icrfToFixed, position_eci,
position_ecef);
        point.position = position_ecef;
    }
}
```

Code Example 2 – Example code used to calculate the ECEF coordinate of the space objects at an updating time from the ECI coordinates.

- **radar_screen(radar_position_ecef) function**

This function is the baseline of user input module and visualization of the space objects on the radar viewer. This function is based on using the radar position in ECEF coordinate as the center point of the ENU reference frame to transform the ENU camera's position to ECEF position, and then set the camera parameters for observation of the space objects. The parameters of the camera scene are shown in Figure 6 below.

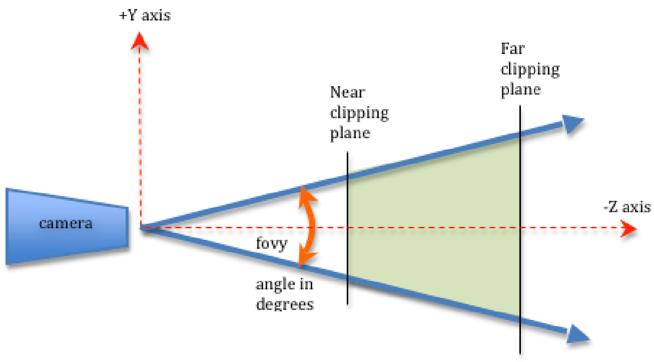

Figure 6: Camera Scene (Learn WebGL, 2016)

The camera scene setting process is based on the Cesium.camera while the camera's position in ECEF is used as "camera_position_ecef" in coding. The example code for setting the camera scene is shown below:

```
  radar_viewer.camera.direction = camera_position_ecef;

  var normal_tmp = new Cesium.Cartesian3();

  var camera_up = new Cesium.Cartesian3();

  camera_up   =
Cesium.Cartesian3.cross(camera_position_ecef,normal_tmp,camera_up);

  radar_viewer.camera.up = camera_up;

  radar_viewer.camera.frustum.fov = 60/180* Cesium.Math.PI;

  radar_viewer.camera.frustum.near = 0.1;

  radar_viewer.camera.frustum.far = 20000000.0;

  radar_viewer.camera.frustum.aspectRatio =
radar_viewer.scene.canvas.clientWidth/radar_viewer.scene.canvas.clientHeight;
```

Code Example 3 – Example code for setting parameters for the camera scene.

4.3 Enhanced Module 1 – Original User Input Module for Radar (Telescope) Position

"What are the reasons for setting the user input module for the radar position in this research project?"

Firstly, the user input modules are commonly provided by most current visualizers for the space objects as it provides customized options for the users to meet their need. However, for the UCL baseline visualizer, the user input system is not realized because the position of the radar is set at a stationary position, and radar viewer only demonstrates the observation of space objects from this singular position. Secondly, from the perspective of users who work in space research organizations and want to get the information of the space objects observed by different radar stations in the world for space exploration or space monitoring, in this way they can input their target positions of the radar or telescope position on the screen and access different radar viewer of the space objects, which meets their expectations. All in all, one of the target enhancement objectives is to set the user input system.

4.3.1 Enhanced Process

The user input system is designed for the need of users to set a radar point by their choices of longitude and latitude, which can be keyed into the JavaScript input box and then showed on the main screen view. A default position is needed in case that the user ignores the input or instance of any fault. As this research project was started in London, the default position of the radar position will be set as central London with latitude 51.0 and longitude 0.0. Also, this enhanced system is set to update the user input position once they change the entered longitude and latitude, which means that the previous radar point will be removed before the new point added to globe scene on the main viewer. The code example 4 below shows the function of setting user input longitude and latitude.

```
function set_value(){
    var radar_position_ecef =   new Cesium.Cartesian3(0,0,0);
    var longitude = document.getElementById("user_longitude").value;
    var latitude   = document.getElementById("user_latitude").value;
    var height = 10.0;
    radar_position_ecef = Cesium.Cartesian3.fromDegrees(longitude, latitude,
height);
    var point = viewer_main.entities.getById("London");
```

```
    viewer_main.entities.remove(point);

    // // /// show the position of the telescope

    var radar_point = viewer_main.entities.add({

      id: "London",

      name : 'Telescope Point',

      position: Cesium.Cartesian3.fromDegrees(longitude, latitude, height),

      point: {pixelSize : 15,color :   Cesium.Color.DEEPPINK}

      });

    radar_screen(radar_position_ecef);

}
```

Code Example 4 – Code used to create the radar point on the globe scene of main viewer with position updating functions.

The process of the user input module supporting the enhanced visualizer is shown in Figure 7 below.

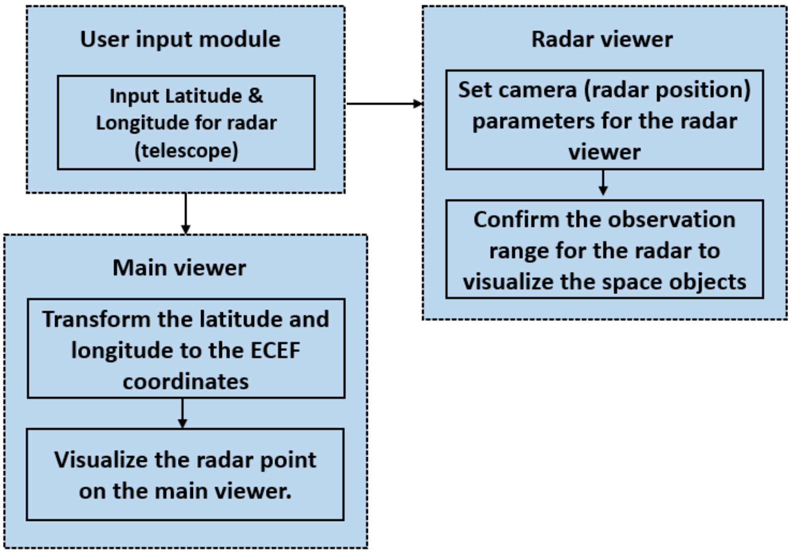

Figure 7: The enhancement from the user input module to the visualizer

The enhanced process of the user input module is to first convey the longitude and latitude information to the function for setting radar viewer to set the radar position and camera parameters in order to show observations of the space objects via functions for visualizing the object points, which will be described in the later sections. Then for the main viewer, it applies the position to visualize the radar point on the globe view. Therefore, the user input module is closely linked to the main viewer and radar viewer.

4.3.2 Visualization Principle

For the enhancement of realizing customized user input longitude and latitude for the radar position, the user input module is designed intuitively to position at a correct spot on the screen with the visualization principle shown in Figure 8. To realize the principle of balance (Tufte, 2002), the radar viewer module is shown below the linking connection and the ability of users to focus on the module.

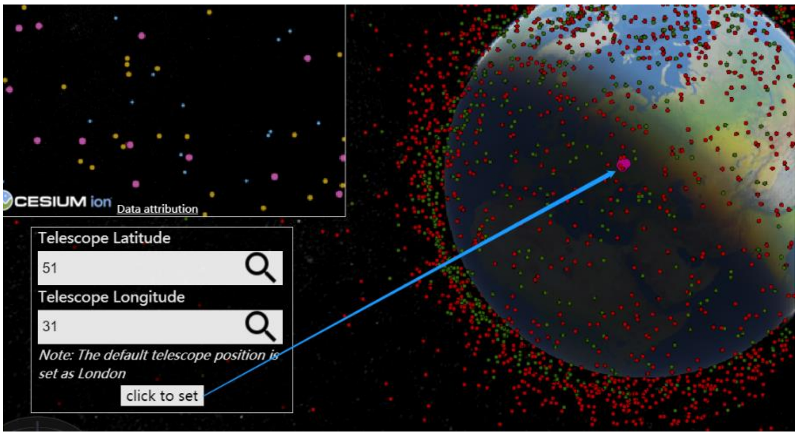

Figure 8: The original design of the user input module

4.4 Enhanced Module 2 – Enhanced Clock and Timeline Controller

For the UCL baseline visualizer, the author ignored that the clock widget and timeline widget should match the information of the future space catalogue.

According to the UCL version, current UTC time when the user loads the catalogue is used rather than the future time from future space catalogue, which leads to a wrong match of information between the clock & timeline widgets and future space catalogue.

Automatically updating the clock widget and timeline widget is strengthened by this enhancement shown in Figure 4 “Timeline Controller” when the user loads the future space catalogue file of a future time point. This means the time widget can automatically show the time the same as the future time point of the catalogue.

4.4.1 Enhanced Process

Depend on the catalogue file year provided by UCL research team (Ziebart, Bhattarai et al., 2019), the research project is focusing the provided time point of future 2023, 2028 and 2043 and a baseline time point in 2019, and the time information of the future space catalogue is captured first. To control the clock widget and timeline widget, the classes of Cesium.JulianDate and Cesium.Viewer have been used. The matched time information of the catalogue and the JulianDate input are shown in Table 3 below.

Table 3: Matched time information for the catalogue and JulianDate input

Catalogue time information	Matched JulianDate input
fspcat_baseline_20190522_v280819_nodeb	2019-05-22T00:00:00Z
fspcat_20230101_v280819_nodeb	2023-01-01T00:00:00Z
fspcat_20280101_v280819_nodeb	2028-01-01T00:00:00Z
fspcat_20430701_v280819_nodeb	2043-07-01T00:00:00Z

The method is to set the new current time and start time for these two widgets and zoom to the new time line with the process of creating an “if” statement to identify the matched time. The example code of the matched identifier is shown below:

```
if (value.substring(6, 15) == "_20280101") {

        start_jd = Cesium.JulianDate.fromIso8601("2028-01-01T00:00:00Z");

        viewer_main.clock.currentTime = Cesium.JulianDate.fromIso8601("2028-
01-01T00:00:00Z");

        viewer_main.clock.clockRange = Cesium.ClockRange.UNBOUNDED;

        viewer_main.timeline.updateFromClock();

        viewer_main.timeline.zoomTo(start_jd,
Cesium.JulianDate.addSeconds(start_jd, 86400, new Cesium.JulianDate()));

}
```

Code Example 5 – Code used to create an identifier to automated change the clock and timeline widget when loading the future space population at the predicted time.

4.5 Enhanced Module 3 – Enhanced Radar Viewer

It is suggested that the radar cross section value cannot be identified by radar viewer of the UCL baseline visualizer. However, the radar cross section is an important value to identify how the space object is detectable and a larger radar cross section reflects that the object is easier to be detected by radar (Knott, 1993). Radar viewer which can identify a space object by the radar cross section value and indicate the information on the screen is rated as very effective to deliver the object detectable information. As a result, the enhanced radar viewer is designed to show the observation of the space objects by the value of radar cross section.

4.5.1 Enhanced Process

The three stages of enhancing the radar viewer by radar cross section are shown in Figure 9.

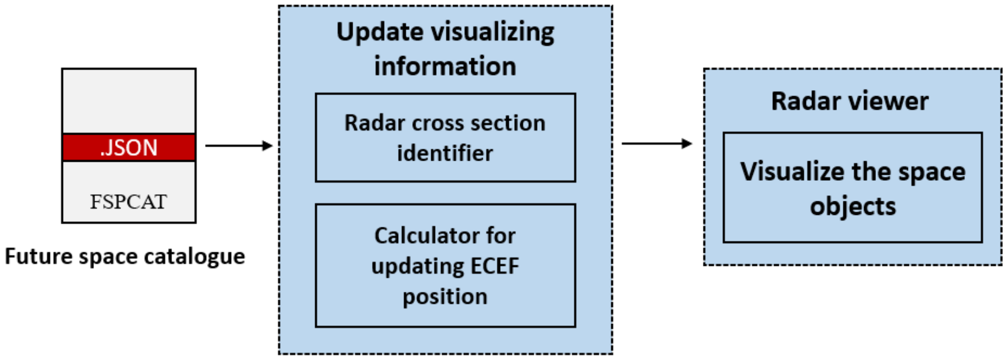

Figure 9: Process of enhanced radar viewer

The future space catalogue is loaded at the first stage in order to enhance the radar viewer. At the second stage, the radar cross section information of each space object is captured, and different visualizing styles based on radar cross section by their ids are set. With the updating ECEF position information calculated by the update_debris_position function described in chapter 4.2.3, the final stage is to show the space objects in different styles depends on radar cross section on radar viewer.

While the main enhancement in the process to the enhanced radar viewer is the radar cross section identifier showed in Figure 9, which is designed to first capture the radar cross section information of each object by loop and then set a value as an identifier to decide the visualizing style. The function getDebriCross_Section(i) is developed for the loop of each space object by id and set an identifier value for variable “m” shown in code example 6. Three cases are created – radar cross section (RCS) under or equals 1 m^2; RCS between 1 m^2 and 10 m^2; RCS larger than 10 m^2. The consideration of choosing dividing line of 1 and 10 is based on two reasons: one is the RCS reference value for point-like targets – man as a target with about 1 m^2 RCS (Radartutorial.eu, 2021), the other is that 10 m^2 is used as a filter value from the catalogue file. If RCS value is less than a man's reference RCS, it is considered as rather small space objects as a man's reference value is rather small in all reference RCSs. The 10 m^2 is a filter value from the analysis of catalogue file which ensures that the proportion of those space objects with RCS larger than 10 m^2 is less than 10 percent (8.6%, 6.4%, 9.8% in 2023, 2028 and 2043 catalogue), which is to emphasize those more detectable objects with large RCS value.

```
getDebriCross_Section(isat){

       var m = -1;

       if (isat < this.debris_kep.length){

           var cross_section_0 = this.debris_kep[isat]["radar_cross_section"].trim();

           var cross_section = Number(cross_section_0);

       if (cross_section <= 1) {m = 1}

       else if (cross_section <= 10 && cross_section > 1) {m = 2}

       else if (cross_section > 10) {m = 3}

           }

       return m;

}
```

Code Example 6 – Example code for identify the radar cross section field.

4.6 Enhanced Module 4 – Statistical Analysis Module

4.6.1 Enhanced Process

While the UCL baseline visualizer does not provide the analysis module to visualize the useful statistical data on the screen, the statistical information captured from the future space catalogue is available by this enhanced module. It is designed to use four separate parts for the completeness of enhanced analysis module: legend for the space objects shown on the globe scene of main viewer; a counter for the number of active satellites and inactive satellites (by operational status); a pie chart for the statistical data of the number of satellites launched by different countries; a histogram for the statistical data of the number of satellites in different orbital categories (LEO, MEO and GEO). The whole enhanced analysis module of the visualizer is shown in Figure 10.

The detailed designed methodology including identifier information and code example in each separate analysis module is indicated in section 4.6.2 – 4.6.4.

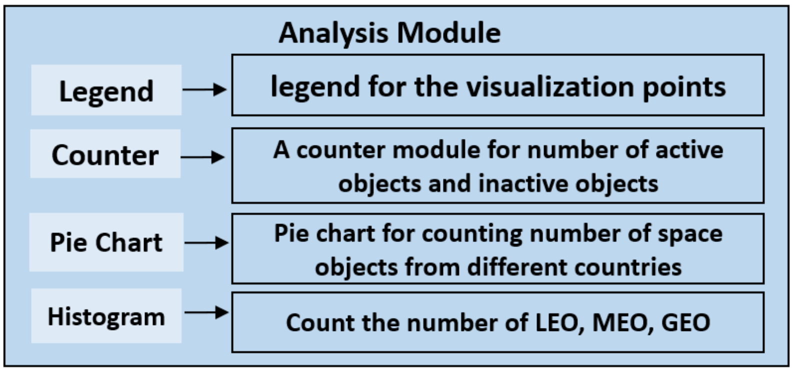

Figure 10: Analysis module in the enhanced visualizer

4.6.2 Legend and Counter for Satellites by Operational Status

This counter module is designed with the convergence of the legend module by using an identifier for the operational status information captured from the future space catalogue in "payload_operational_status" field and different values in this field representing different operational status shown in Table 4 below. While this counter is performed by active space objects and in-active space objects, the active space objects include operational and partially operational objects, backup objects and objects with extended missions, the in-active objects indicate the no-operational objects.

Table 4: Fields of operational status and its representation

Operational status field	Representation
'+ '	100% Operational objects
'P '	Partially operational objects
'B '	Backup objects
'X '	Objects with extended missions
'- '	Non-operational objects

Code example 7 shows the identifier for operational status. In the code of the identifier, the format of the operational status field is unified at first, and then set a unique number for space objects with different field value. The variable “s” in the code is the indication for active objects and inactive objects and is returned as the result of the identifier. If “s” variable is larger than 0, the objects are set as active, otherwise inactive.

```
getDebriOperation_status(isat){
    var s = -1;
    if(isat < this.debris_kep.length){
        var aa = this.debris_kep[isat]["payload_operational_status"];
    if(aa == '+      ') {s = 1;}
```

```
        else if(aa == '-        ')  {s = -1;}

        else if(aa == 'P        ')  {s = 0.5;}

        else if(aa == 'B        ')  {s = 0.2;}

        else if(aa == 'X        ')  {s = 0.3;}

        else  {s = -1;} }

                return s;

}
```

Code Example 7 – Example code for the identifier of operational status.

4.6.3 Visualization for Counted Number of Satellites by Countries Using Pie Chart

As the launched satellites number by countries is increasing rapidly in recent years, especially for United states, Russia and China (Mohanta, 2021), will the on-orbit space objects be dominated by these countries in the future? It is important and useful to assess the statistical data of satellites launched by different countries which are still on orbit in future time points to get the answer by the results.

This enhanced analysis module is designed to show the statistical information of satellites launched by different countries in future time points. For intuitive visualization, it is suggested to use statistical tools to show the number. When choosing the tool, two alternatives have been considered – pie chart and bar chart. The characteristic and advantages between these two alternatives are shown in Table 5. According to the advantages and the need to show the proportion dominated by different countries, it is preferable to choose the pie chart.

Table 5: Pie chart versus bar chart (Chartio, 2016)

Pie chart	Bar chart
Focus on the proportion	Focus on the numeric value
Intuitively to show the change of proportion in each year	Intuitively to show the change of numeric value in each year

The statistical data is generated based on the field of "owner" which contains the launched country information in future space catalogue. Six different countries and districts are chosen as targets – United states, China, Russia, UK, European Union and rest countries in the world.

An identifier is used to achieve the statistical data in this enhanced module shown in code example 8 below. While the "owner" field is related to the SATCAT launch site code assessed on CelesTrak website (CelesTrak: SATCAT launch site, 2020), variable "c" is used in the identifier to set unique value for different countries and districts and returned as the result of the identifier. Pie chart eventually displays the statistical data via the identifier.

```
getDebriCountry(isat){
    var c = -1;
    if (isat < this.debris_kep.length){
    var bb = this.debris_kep[isat]["owner"];
    if (bb == 'US     ') {c = 1} //The United states
    else if (bb == 'PRC   ') {c = 2} //China
    else if (bb == 'CIS   ') {c = 3 } //Russia
    else if (bb == 'UK     ') {c = 4} //UK
```

```
        else if (bb == 'ESA   ') {c = 5} //European Union
        else {c = 6} //other countries }
            return c;
}
```

Code Example 8 – Example code for identifying countries.

4.6.4 Visualization for Counted Number of Satellites by Categories Using Bar Chart

Based on the location of circular (geocentric) orbit, due to the different altitude above the earth's surface, the satellites are categorized into LEO, MEO and GEO. The orbit altitudes of these three categories are defined as LEO lower than 2000 km, MEO between 2000 km and 35768 km, GEO at 35768 km. This enhanced analysis module is set to produce the statistical data of LEO, MEO and GEO.

An identifier of the satellite category shown in code example 9 is created to achieve the statistical result. The information captured by the identifier from 2 fields in future space population catalogue – "apogee_hgt" represents the highest altitude of an on-orbit object from the earth surface; "perigee_hgt" indicates the lowest altitude of an on-orbit object from the earth surface. The "cat_0" is a variable for the value of the apogee height while the variable "cat_1" is the value of perigee height. The two values are used together in this research project to identify the category of the satellites. The variable "cat_num" is set as the identifier to return unique value for the final statistic.

```
getDebriCategory(isat){
        var cat_num = -1;
```

```
        if (isat < this.debris_kep.length){

              var cat_00 = this.debris_kep[isat]["apogee_hgt"].trim();

              var cat_11 = this.debris_kep[isat]["perigee_hgt"].trim();

              var cat_0 = Number(cat_00);

              var cat_1 = Number(cat_11);

        if ( cat_0 <= 2000 && cat_1 <= 2000 ) {cat_num = 1} //LEO

        else if ( (cat_0 > 2000 && cat_0 < 35786) && (cat_1 > 2000 && cat_1 <
35786) ) {cat_num = 2} //MEO

        else if ( cat_0 == 35786 || cat_1 == 35786) {cat_num = 3} //GEO

        else { cat_num = -1 }}

        return cat_num;

}
```

Code Example 9 – Example code for identifying LEO, MEO and GEO.

5. Results

5.1 The Enhanced Visualizer

The methodology of the enhanced process has been introduced in chapter 4, thus the result of the enhanced future space population visualizer is shown in this chapter. The enhanced visualizer displays the future space objects listed by the future space catalogue (FSPCAT). The visualization results of the future space population in three future time points – 2023, 2028 and 2043 supported by the FSPCAT are shown in Figure 11 – Figure 13.

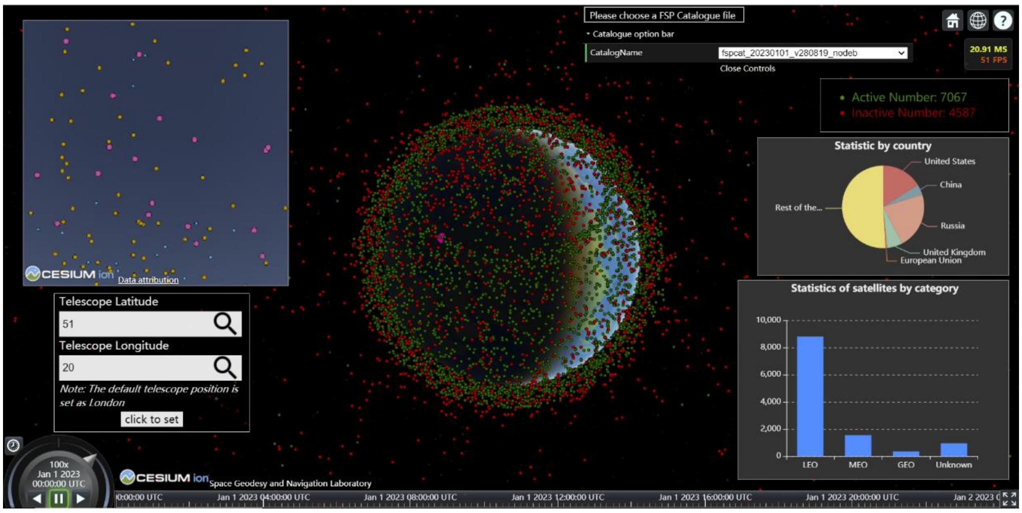

Figure 11: Visualization of predicted 2023 space population

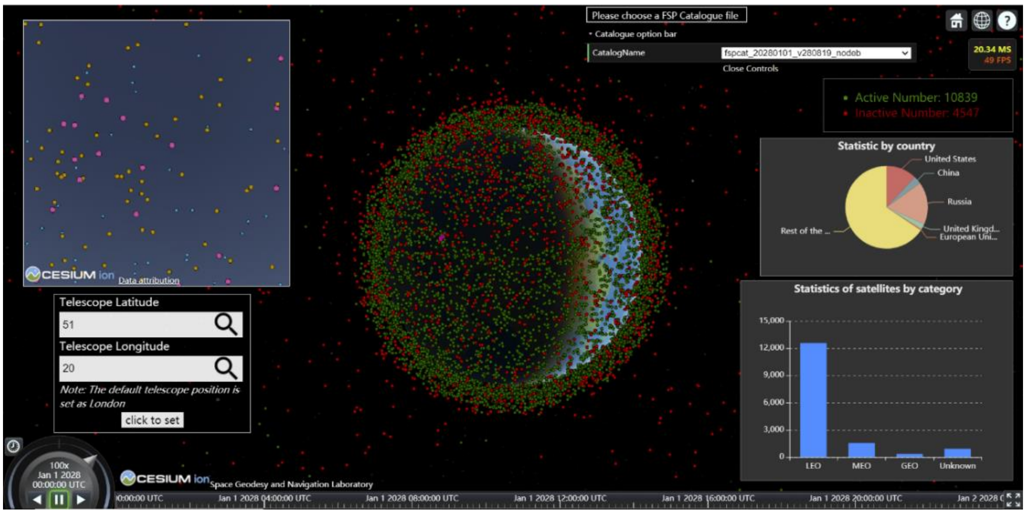

Figure 12: Visualization of predicted 2028 space population

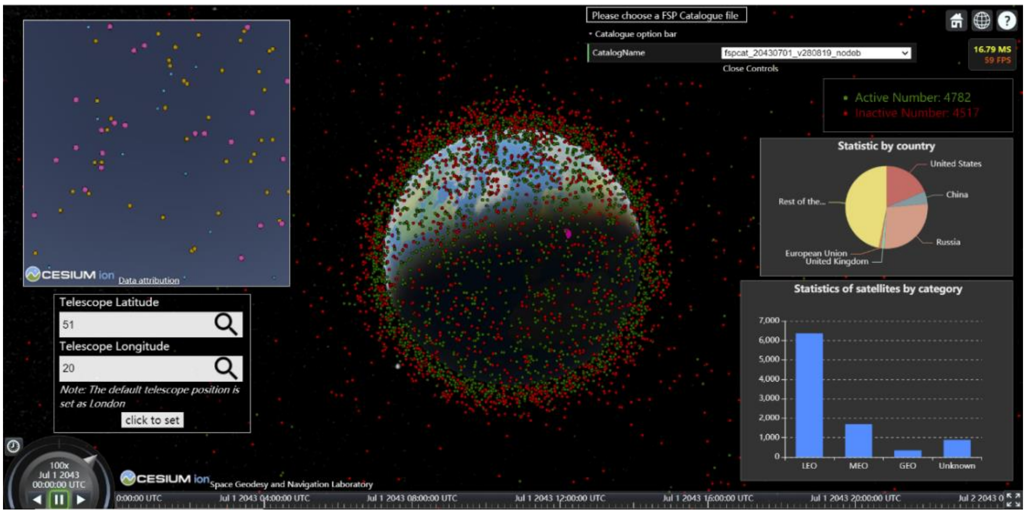

Figure 13: Visualization of predicted 2043 space population

Four enhanced modules presented in chapter 4 for enhancing the visualization of the future space population are contained in the enhanced visualizer – the enhanced user input module, the enhanced timeline controller, the enhanced radar viewer and statistical analysis module for the space objects.

The guidance for users to use the visualizer is shown in Figure 14:

Step 1: user input the telescope position from the user input module at bottom left corner and click to set the point on the main viewer.

Step2: automatically initialize the telescope position for the radar viewer.

Step 3: user click the catalogue option bar to load a catalogue.

Step 4: automatically visualize and analyze the space objects.

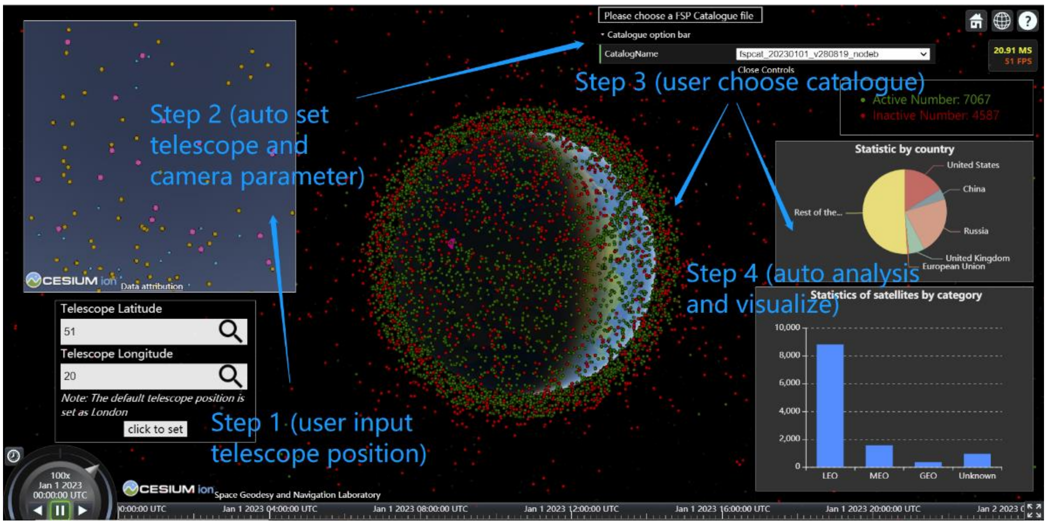

Figure 14: Guide steps for users to use the enhanced visualizer

From the user guidance, it is apparent that the enhanced visualizer as a whole is designed based on the “Illustrating movement” principle which is described in section 2.3 with a clockwise movement for users to set the inputs and visualize them on screen.

5.1.1 Catalogue Controller Module

The catalogue controller is created for loading the future space catalogue and is set on the right top corner with the visualization principle of “clockwise visualizing” shown in Figure 15. It is designed in a GUI option bar for intuitive visualization useful for guiding the users to click the catalogue control menu and choose the future space catalogue to load the space objects on the main viewer and radar viewer of the screen.

Figure 15: Visualization of catalogue controller GUI

5.1.2 Enhanced Clock and Timeline Widget

The enhanced clock and timeline widgets are designed to match the current time to future space catalogue which has been detailed described in chapter 4.4. The result in Figure 16 shows that the clock and timeline widget can be automatically loaded as the same as the loaded future space catalogue shown in Figure 15. It is indicated that the match from the clock and time line to the catalogue has been successfully provided in the enhanced visualizer.

Figure 16: Visualization of enhanced clock and timeline widget

5.1.3 Enhanced User Input Module and Radar Viewer

The enhanced user input module and the radar viewer are closely linked to each other, and are positioned on the left side of the screen. From Figure 17 below, the user input module is set under the radar viewer and based on the visualizing principle of balance to facilitate users to choose their target telescope position. Both modules are developed to attract the user's attention when they enter the visualizing system.

The user input module has been tested for its accuracy and efficiency. During the test with the aim to find the rate of veracity of the radar viewer set by the user input position, about 100 orbital objects are randomly chosen using the radar viewer observation to check if the radar viewer linked to the user input position can work properly. It is suggested by the results that the user input module is correctly linked to the radar viewer.

The space objects on the screen can be visualized with the enhanced radar viewer by using different point styles dependent on the radar cross section value. Three different styles of points are used in the radar viewer based on the identifier described in chapter 4.5. Larger radar cross section is presented as bigger dots.

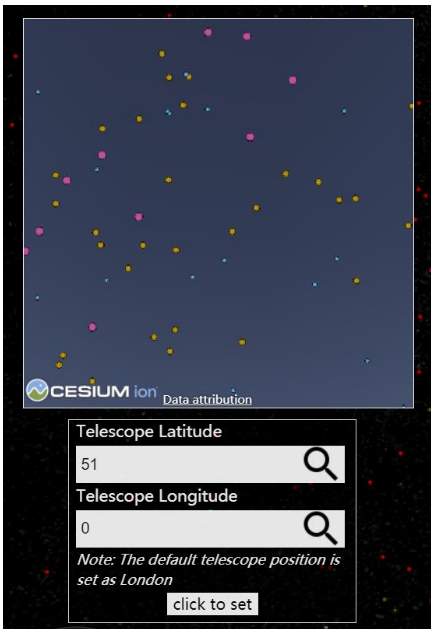

Figure 17: Visualization of enhanced user input module and radar viewer

5.1.4 Enhanced Analysis Module

Three separate modules shown in Figure 18 are consisted in the enhanced analysis module: legend with counter, pie chart statistical tool and bar chart statistical tool. The analysis module is placed on the right side of the screen and placed vertically. For the legend counter, the color is set the same as the objects visualizing on the main viewer: green for active satellites and red for inactive satellites. In the pie chart, the statistical data of launched satellites by country, the proportion information and counted value when hovering the mouse on the module which are displayed in Figure 19, while the bar chart visualizes the statistical data of satellites by category, and the counted number will be shown when hovering the mouse on the module shown in Figure 20.

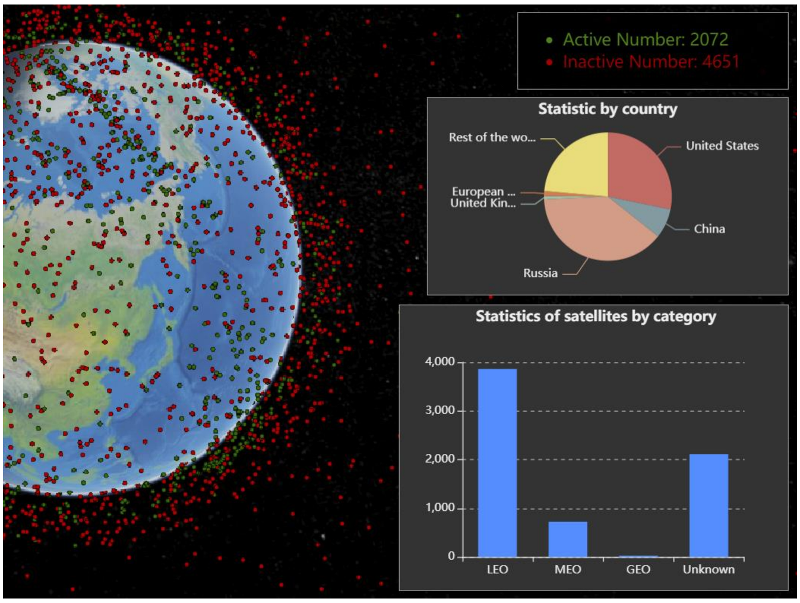

Figure 18: Overview of the enhanced analysis module

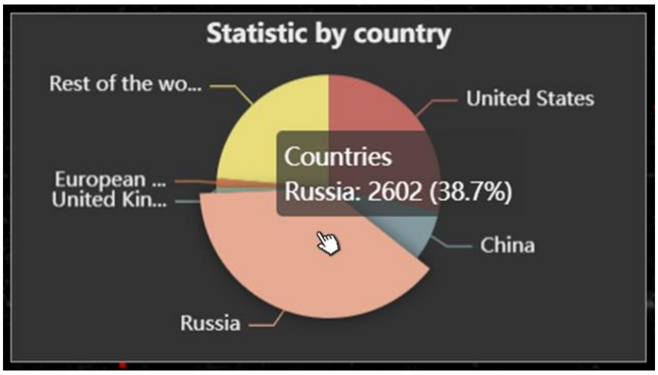

Figure 19: Pie chart for statistical data

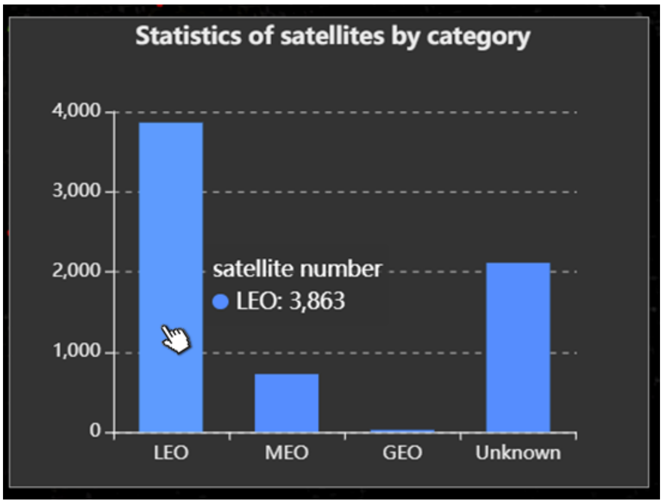

Figure 20: Bar chart for statistical data

5.2 Test for the Visualizer

As the visualizer is designed to visualize the space objects with top quality, it has been tested the compatibility to different browsers which are commonly used today. Google Chrome, Microsoft Edge and Firefox are all included. As it is on the stage of research, all the test are on author's local server of personal computer, the test result shows this enhanced visualizer is compatible to common browsers with expected visualizing results.

6. Discussion

6.1 Discussion of Enhancements and Result

With the research aims and objectives stated in chapter 1.4, the study mainly focuses on developing an enhanced visualizer based on the UCL baseline visualizer and available resources. In this research project, several new enhancements are proposed to better visualizing the space objects either in main globe viewer or in radar viewer, and to analyze the statistical data based on the visualizer. Firstly, the enhancements will be discussed as follows.

- Enhancement 1 – User input module

The first enhancement is an enhanced user input module for users to customize their radar position for research. It is claimed to be significant to some users, e.g., researchers who work on exploring the visualizations of space population from different space monitoring stations, and need the visualizing data from these stations at the same time to compare. By using the user input module, they can just change the telescope position and compare the two visualization results to find useful information. Besides, users working in space agencies who need frequently change telescope positions in order to monitor the status of space objects, it is beneficial for them to use the user input module. In addition, it provides a foundation for users' further developments on the field of customized visualization. Therefore, this enhancement provides a significant contribution to the development of visualizing space population compared to UCL baseline visualizer in terms of customized visualization and users' benefits.

- Enhancement 2 – Automated clock and timeline widget

Shown as the result in chapter 5, this enhancement is considered as a useful contribution to the UCL baseline visualizer as it provides the connection between the current time system and loaded catalogue. By accessing the current time information and the loaded catalogue as well, it is easier for users to catch the visualization of space objects at a specific time, if they work for simulating space environment in space agencies, who need to analyze the statistical data from different time points and a large number of catalogue files providing space object information with different time points are needed. In this case, building a connection between time system and catalogue is crucial for users to quickly tackle these data files. In addition, it provides a solid foundation for users to further develop the clock and time system. Therefore, it is a significant enhancement based on the UCL baseline visualizer.

- Enhancement 3 – Enhanced radar viewer

The result in chapter 5 demonstrates that the space objects can be visualized with quality images by using the enhanced radar viewer due to their radar cross section (RCS). Furthermore, the RCS information can be directly and clearly passed on to the users compared to the UCL baseline visualizer, which is considered to be significant to some users: e.g., researchers who want to explore detectability of the space objects. As the enhanced radar viewer is connected to the user input coordinates and the view of space objects changes with telescope position accordingly, they can get the detectable information from different telescope positions to analyze the space environment from different radar views. Furthermore, space operators from space agencies working on monitoring the space objects need the radar viewer indicating the RCS information as they need the space objects to be detectable under surveillance in order to constantly alert the objects with any deviation from normality. Finally, it also provides the foundation for the further development.

- Enhancement 4 – Statistical analysis Module

This enhancement is also considered as a significant enhancement to the UCL baseline visualizer with the pie chart and bar chart for statistical analysis based on satellite operational status, launched countries and satellite category. From the result shown in chapter 5.1.4, useful statistical data for future space population in 2023, 2028 and 2043 is provided by this enhanced module, which is a significant contribution to the users especially data analysts in space agencies who need to predict the developing trend of the future space population by each year's statistical data.

In summary, with the result of the enhanced visualizer stated in chapter 5 and discussion in chapter 6.1, the study fulfils the aims and objectives with developed enhancements in terms of its functions and different users' benefits.

6.2 Discussion of Enhanced Process

In sum, the research process in the research project is based on the UCL visualizer with some available codes and frames to develop an enhanced visualizer for the future space population.

It is agreed that an ideal enhanced visualizer provides a better visualization outcome compared to the previous one. Thus, the evaluation of whether the enhanced process in visualization for this research is reasonable and whether the identifier (stated in chapter 4.3, 4.4 and 4.6) used for visualizing could perform better need to be discussed.

6.2.1 Three Identifiers

The first discussion is about the three identifiers stated in chapter 4.3, 4.4 and 4.6 for satellite operational status, radar cross section and satellite category.

- *Identifier for operational status*

The first identifier is used for the statistical analysis of the active and inactive objects derived from the information in future space catalogue. The identifier is designed to set active and inactive numbers of the satellite while the field of the operational status includes several cases like partially operational, backup, operational, which are all considered as active. The consideration of only two cases – active and active - is to intuitively visualize the objects on the screen using green and red for users avoiding creating a messy view of several colors for more than two operational statuses. However, although using active and inactive as a means to be intuitive and neat for statistical, it is not regarded as an optimal choice for some users who need a more detailed category for satellites. Therefore, the identifiers, which effectively simplify the cases for the satellites and successfully provide a neat category, are unfortunately considered not useful for some users.

- *Identifier for radar cross sections*

This identifier is used to decide the size of the space objects visualizing on the radar screen. The radar cross section (RCS) in the units of m^2 reflects how detectable a space object is by radar while a larger RCS means more easily to be detected. The identifier stated in section 4.5.1 use 1 m^2 and 10 m^2 as dividing lines with the consideration based on the mankind's reference RCS value and analysis of the range of cross section value in space catalogue data: 1 m^2 represent the most common man's reference RCS. It is suggested that larger than 10 m^2 , with only less than 10 percent of space objects in this RCS range are more detectable objects as they are rare. However, 1 m^2 is just a critical evaluation for a rather small space object based on reference value, and 10 m^2 is just a data analysis from future space catalogue at three time points, it is not regarded as general case if more detailed reference values and more catalogue files are provided. Therefore, the value 1 and 10 are reasonable to be set as dividing lines on the condition that they are generated by critical analysis instead of general choices by recorded knowledge.

- *Identifier for LEO, MEO and GEO*

This identifier is used to decide the category of the space objects by altitude above the earth surface as stated in section 4.6.4, and the dividing line of the identifier is set as 2000 km and 35786 km to distinguish the LEO, MEO and GEO. However, as the criterion of altitude for the dividing line is not universal in all the authorized space research websites, aerospace security website is chosen as the criterion for this research project with the consideration that a better identifier should be set by different criteria based on different users and different research context. It is concluded that this identifier for the research project provides useful statistical result to achieve the research objectives even though improvements are needed in the future.

6.2.2 Visualizing Strategy

The second discussion is about the visualizing strategy on enhanced radar viewer and main viewer.

- *Object style in enhanced radar viewer*

As the objects visualized on the enhanced radar viewer is designed on the basis of radar cross section (RCS), and object size reflected in the radar viewer depends on the value of RCS, the visualization strategy should be discussed.

The style for visualizing the object is set as - point with pixel size 7 colored purple pink for radar cross section > 10 m^2; point with pixel size 5 colored orange for radar cross section between 1 m^2 and 10 m^2; point with pixel size 3 colored sky blue for cross section < 1 m^2. The radar viewer can successfully distinguish space objects with different RCS values and visualize the results. However, a potential problem is that the RCS value captured from the space catalogue simply reflects how detectable the object is at a time point as same as the loaded future space catalogue, which means the radar viewer can merely reflect the visualization based on RCS at the given time with stationary status. If the clock starts working, the RCS will also change, it is admitted that the visualization will not so accurate. Therefore, the consideration of setting a formula for the calculation of RCS at an updating time based on the baseline RCS value in future space catalogue is regarded as an improvement strategy, which is discussed in 6.4 and 6.5 section.

- *Visualization on main viewer*

In general, the visualization on main viewer is designed to show the future space population object points with two styles – green for active satellites and red for inactive satellites. It is evident that its advantage is intuitive and clear view, but the issue of providing more choices to cater for different users' expectations need to be addressed. The alternative option with setting more styles for the satellites based on operational status could also help the visualization, as users could choose the option for visualizing satellites with 2 operational statuses or more than 2 operational statuses based on their need.

In summary, the research strategy as a whole is correct and reasonable but some details and criteria could be improved.

6.3 Discussion of the Visualizing Principles

For the overview of the result shown in chapter 5, the enhanced visualizer is designed with the balance principle: radial type which means that each visualization module is placed around the central 3D earth scene and the points of space objects around the 3D earth. Because the 3D earth scene is the direct visualization by CesiumJS, while each enhanced module makes its own contribution to the whole enhanced visualizer and they are placed around the central 3D earth.

Emphasis on the key information is selected as another visualizing principle, especially for one of the significant sections – enhanced radar viewer. Tufte (2002) indicated that the most significant section should be placed at top left of a visualizer because it is the prime location to capture the user's attention. Therefore, the emphasize principle is first reflected by the position - the radar viewer is placed at the top left corner and has also been used in the interior of each section, e.g., use reasonable color and size for points shown in radar viewer in order to emphasize the information of radar cross section value. In addition, the associated texts around the bar chart and pie chart are designed with consideration of relative position to make them as a whole to attract the user's attention.

The illustrating movement principle is also used in the enhanced visualizer is designed with clockwise direction shown in Figure 14. As the user is presumably considered to use the user input module for the first time, it is first placed on the bottom left corner and based on clockwise movement principle. The radar viewer and catalogue controller are paced on the top left corner and top right corner, while the last enhanced module with three statistical analysis sections is placed from middle right to bottom right in order. This clockwise design is illustrated as the movement guide for users to use the enhanced visualizer.

Proportion principle has also been applied in the design for the size of each visualization module, especially the enhanced statistical analysis module with the reasonable size of each separate analysis section – legend, bar chart and pie chart. The size of module is designed depending on the relative position between each module in case that a module is displayed with inappropriate size.

6.4 Limitations

The discussions in 6.1, it is suggested that the enhancements are significant to some users and the result of the enhanced visualizer is proved satisfactory to fulfil the research aims. In chapter 6.2 the enhanced strategy is discussed, and some feasible improvements advised. However, there are also some limitations generated from this research project.

The first limitation is related to the evaluating strategy in discussion. The discussion in this research project is based on internal evaluation without external evaluation which means no valid users' feedback and suggestions collected, e.g., the contributions of each enhancement to users are evaluated by author with critical analysis which could be considered as hypothesis. Because of the specificity of the project, the study is practically not accessible to any users at time being due to its lack of the external evaluations from users' feedbacks. Therefore, it is considered that lack of external evaluation becomes the first limitation in this research.

The second limitation is related to the choice of API in CesiumJS for visualization. For this research project, all the objectives visualized on the screen are via the "Primitive" API in CesiumJS which can be used directly to control the visualization of a large number of point objects and graphic display. However, although the point objects visualized on the screen display the space population, it is simply the visualization without any functions for click events. For the current visualizer online such as Astria visualizer, it provides the click event for users to get the information of each object on screen by info box. This means a better choice of API for visualization could be used – “Entity” which is used in Astria visualizer. This API is a higher-level of visualization API than “Primitive” as it contains info box item, label items and other items for enhanced visualization. Therefore, it is considered as an improvement for further development discussed in section 6.5.

The third limitation has been stated in section 6.2 due to the radar cross section field. As the styles of objects visualized on radar viewer only depend on the RCS value at the particular time point as the time of the space catalogue, it cannot dynamically change the style dependent on the RCS value calculated by a specific formula. Although it can visualize the space objects on a specific time with high accuracy with RCS, the visualization on other time points will not provide a great performance because the RCS changes with time. Therefore, it is a limitation needed improvement in the enhanced visualizer in the future.

6.5 Further Development

With the potential improvements stated in chapter 6.2 and limitations stated in chapter 6.3, there are several recommendations for the further development based on identifiers and space object visualization.

For the identifiers discussed in chapter 6.2, it is considered that improvements can be laid on the aspects of choosing the number of classifications and reasonable divide lines. They need further research and development.

For the visualization of space objects on the main screen with click events is one goal for enhancement, while the “Entity” API can be applied in the future development rather than current “Primitive” API, which shows the information of each space object by an info box shown in Figure 21. The info box will be built on the basis of “Entity” API with more setting properties.

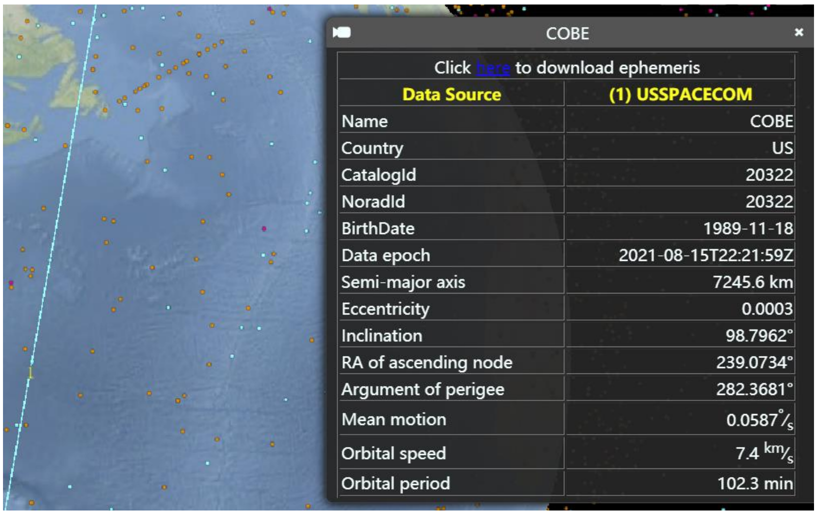

Figure 21: Info box for visualizing the information of objects

For the visualization on the radar viewer, the current object style depends on the radar cross section (RCS) value from the space catalogue at a specific time point, while it is stated that the accuracy of dynamic visualization (at an updating time) is not high in section 6.3. Therefore, the further development will be based on building a formula for calculating the RCS value in updating real time which can visualize the space object depend on the RCS value in real time.

7. Conclusions

In conclusion, the research project focuses on the enhancement on the basis of UCL baseline visualizer to enhance the visualization of future space population in 2023, 2028 and 2043. An enhanced visualizer using CesiumJS library has been developed with the component of an enhanced user input module; an enhanced radar viewer indicating the radar cross section of space objects; an enhanced clock and timeline controller and an enhanced statistical analysis module via pie chart and bar chart – all these enhancements added to the baseline visualizer. The study provides significant contributions to the visualization of future space population. The identifiers and visualizing strategies used in the enhanced process have been discussed as reasonable choices, and the result of the enhanced visualizer shows useful information to the users and build a foundation for their further researches. However, several limitations of the research have been found, therefore, further development is still required to enhance the visualization of the future space population. In sum, the research project has developed several significant enhancements for visualizing the future space population, even though prospective research is necessary and valuable.

References

Garcia, M., 2017. *Space Debris and Human Spacecraft*. [online] NASA. Available at: https://www.nasa.gov/mission_pages/station/news/orbital_debris.html [Accessed 17 Jul. 2021].

Omicsonline.org. (2016). *Types of Satellites and Applications* |. [online] Available at: https://www.omicsonline.org/conferences-list/types-of-satellites-and-applications [Accessed 15 Jul. 2021].

Montenbruck, O. & Eberhard, G., 2000. *Satellite orbits : models, methods and applications / Oliver Montenbruck, Eberhard Gill.*, Berlin: Springer.

Esa.int. (2019). *Types of orbits*. [online] Available at: https://www.esa.int/Enabling_Support/Space_Transportation/Types_of_orbits [Accessed 15 Jul. 2021].

Adushkin, V.V. et al., 2020. The small orbital debris population and its impact on space activities and ecological safety. *Acta astronautica,* 176, pp.591–597.

Mohanta, N., 2021. *How many satellites are orbiting the Earth in 2021?* [online] Geospatial World. Available at: https://www.geospatialworld.net/blogs/how-many-satellites-are-orbiting-the-earth-in-2021/. [Accessed 16 Jul. 2021].

Jonathan, 2021. *Jonathan's Space Report* | *Space Statistics*. [online] Available at: https://planet4589.org/space/stats/stats1.html [Accessed 13 Aug. 2021].

Millan, R.M. et al., 2019. Small satellites for space science; A COSPAR scientific roadmap. *Advances in space research*, 64(8), p.1466.

McDowell, J.C., 2020. The Low Earth Orbit Satellite Population and Impacts of the SpaceX Starlink Constellation. *The Astrophysical Journal Letters,* 892, pp.1-10.

Beck, K., 2021. *SpaceX vs. Amazon: Where we're at in the internet space race*. [online] Mashable. Available at: https://mashable.com/article/amazon-spacex-satellite-internet-race [Accessed 12 Aug. 2021].

M. Ziebart, S. Bhattarai, S. Maskell & L. Devlin., 2019. Future Space Population Study: Catalogue Generation (WP3), produced for Dstl.

Tegler, E., 2021. *When it Comes To The Urgent Issue Of Space Congestion, U.S. Space Command Is Little More Than A Weather Forecaster*. [online] Forbes. Available at: https://www.forbes.com/sites/erictegler/2021/06/13/when-it-comes-to-the-urgent-issue-of-space-congestion-us-space-command-is-little-more-than-a-weather-forecaster/?sh=42049bc5973a [Accessed 14 Aug. 2021].

1996. *IMPACTS ON EARTH FROM SPACE.* [ebook] Post Note. Available at: https://www.parliament.uk/globalassets/documents/post/pn080.pdf [Accessed 17 Jul. 2021].

Sotskiy, M.Y., Veldanov, V.A. & Selivanov, V.V., 2017. Growth in the quantity of debris in Space as AN effect of mutual mechanical collisions of various types. *Acta astronautica*, 135, pp.10–14.

The world must cooperate to avoid a catastrophic space collision, 2021. *Nature*, [online] 596(7871), pp.163–163. Available at:

https://www.nature.com/articles/d41586-021-02167-5 [Accessed 2 Aug. 2021].

Tarran, B., 2021. Prepare for impact: Space debris and statistics. *Significance (Oxford, England)*, 18(3), pp.18–23.

Han Yi, 2010. Web3D and Web's three-dimensional visualization development. *Technology square*, 5, pp.81-86.

Kaur, A. and Sharma, N., 2017. *A Review on 4D Visualization*. [ebook] . Available at: https://www.ijert.org/research/a-review-on-4d-visualization-IJERTCONV5IS03071.pdf [Accessed 24 Aug. 2021].

Qing, X. et al., 2011. Modeling and Visualization of space objects spatio-temporal data. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, 38(5), pp.155–162.

X. Qing et al., 2012. Space moving objects spatio-temporal modeling and visualization. *International archives of the photogrammetry, remote sensing and spatial information sciences.*, XXXVIII-5/W16, pp.155–162.

Möckel, M. et al., 2011. Using parallel computing for the display and simulation of the space debris environment. *Advances in space research*, 48(1), pp.173–183.

Yang Yikang, Liu Xinxing & Liu Lei, 2012. A WebGL-Based Method for Visualization of Space Environment. *2012 Fifth International Symposium on Computational Intelligence and Design*, 2, pp.331–334.

Grey, S. (2015). *Space Debris: 1957 - 2015*. [online] www.youtube.com. Available at: https://www.youtube.com/watch?v=wPXCk85wMSQ [Accessed 31 Aug. 2021].

Maiti, A. et al., 2018. An open source web-gis based precise satellite tracking and visualization tool using two line element data. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 4(5), pp.109–114.

Wang, P. & Cai, Y., 2020. Space environment data field modeling and visualization. *Proceedings of SPIE - The International Society for Optical Engineering*, 11373, pp.113732A–113732A-8.

Moriba, K.J., 2017. *ASTRIAGraph*. [online] sites.utexas.edu. Available at: http://sites.utexas.edu/moriba/astriagraph/ [Accessed 10 Aug. 2021].

Astriagraph' software program combats space junk., 2018. *UWIRE Text*, [online] pp.1–1. Available at: https://link.gale.com/apps/doc/A556545732/AONE?u=ucl_ttda&sid=bookmark-AONE&xid=afe15934 [Accessed 23 Aug. 2021].

celestrak.com., 2021. *CelesTrak Orbit Visualization*. [online] Available at: https://celestrak.com/cesium/orbit-viz.php?tle=/pub/TLE/catalog.txt&satcat=/pub/satcat.txt&referenceFrame=1 [Accessed 23 Aug. 2021].

Tufte, E.R., 2002. *The visual display of quantitative information / Edward R. Tufte*. 2nd ed., Cheshire, Conn.: Graphics Press.

Chen, C., 2010. Information visualization. *Wiley interdisciplinary reviews. Computational statistics*, 2(4), pp.387–403.

Upson, C. et al., 1989. The application visualization system: a computational environment for scientific visualization. *IEEE computer graphics and applications*, 9(4), pp.30–42.

Williams, T.R., 2000. Guidelines for Designing and Evaluating the Display of Information on the Web. Technical communication (Washington), 47(3), pp.383–396.

Streit, M. and Gehlenborg, N, 2014. Bar charts and box plots. *Nature Methods*, 11(2), pp.117–117.

Malamed, C., 2014. Visual language for designers : principles for creating graphics that people understand / Connie Malamed.

Ware, C., 2013. Information visualization 3rd ed., Waltham, Mass: Morgan Kaufmann.

Evergreen, S. & Metzner, C., 2013. Design Principles for Data Visualization in Evaluation. New directions for evaluation, 2013(140), pp.5–20.

Midway, S.R., 2020. Principles of Effective Data Visualization. *Patterns*, 1(9), p.100141.

FSPCAT Visualiser (v2019), Zhen Li for SGNL, UCL, https://spacegeodesy.bitbucket.io/spacedebris_viewer/space_debris.html

Cesium., 2021. *CesiumJS*. [online] Available at: https://cesium.com/platform/cesiumjs/ [Accessed 23 Aug. 2021].

Jo, J.-H. et al., 2011. The Comparison of the Classical Keplerian Orbit Elements, Non-Singular Orbital Elements (Equinoctial Elements), and the Cartesian State

Variables in Lagrange Planetary Equations with J 2 Perturbation: Part I. *Journal of astronomy and space sciences*, 28(1), pp.37–54.

2016. *Learn WebGL — LearnWebGL*. [online] Available at: http://learnwebgl.brown37.net/ [Accessed 23 Aug. 2021].

Knott, E.F., 1993. *Radar cross section measurements / Eugene F. Knott.*, New York: Van Nostrand Reinhold.

Radartutorial.eu., 2021. *Radar Cross-Section - Radartutorial*. [online] Available at: https://www.radartutorial.eu/01.basics/Radar%20Cross%20Section.en.html [Accessed 23 Aug. 2021].

Chartio. (n.d.). *How to Choose Between a Bar Chart and Pie Chart*. [online] Available at: https://chartio.com/learn/charts/how-to-choose-pie-chart-vs-bar-chart/ [Accessed 12 Jul. 2021].

www.celestrak.com. (2020). *CelesTrak: SATCAT Launch Sites*. [online] Available at: http://www.celestrak.com/satcat/launchsites.php [Accessed 25 Aug. 2021].

Appendices

The enhanced visualizer project has been uploaded to personal GitHub account and set as private. Two examiners will be invited as collaborators.

The following link is the .rar format file for downloading:

https://github.com/wadepro233/Final-Research-Project-2021_09

Appendix 1: Guidance for Using the GitHub File

Firstly, unzip the .rar file on the computer.

Secondly, open the command line window, enter **Python -m http.server** to create a sever, then just run the local host in the browser by entering localhost:8000.

Thirdly, open the “fsp21” folder, and then click the “space_debris.html” to open the enhanced visualizer of future space population.