Analyse and Design of Astronomy and Astrophysics C++ Library

Teacher : Dr. Saman Shishe Chi
Student : Ramtin Kosari

QSSL

QSSL

```
# ifndef QSSL
    /**
     * @file QSSL.hpp
     * @author Ramtin Kosari (ramtinkosari@gmail.com)
     * @brief QSSL Open Source Astronomy and Astrophysics Library
     * @note Enjoy Using this Library
     * @date 2024-01-06
     */
    # define QSSL
# endif // QSSL
```

Table of Contents

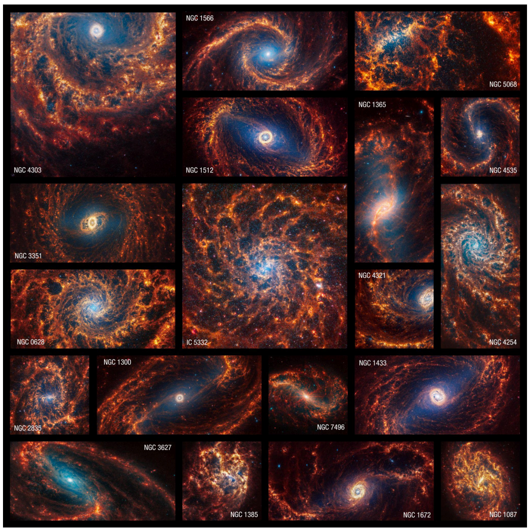
Overview

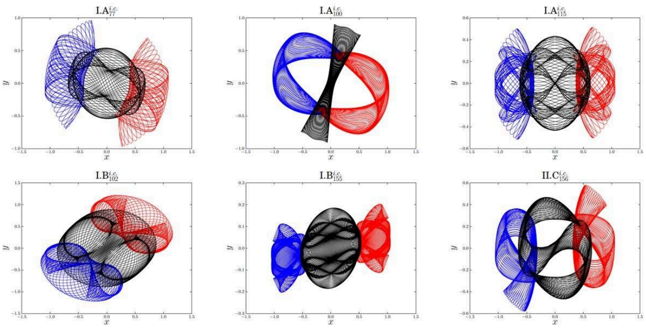
QSSL is a comprehensive open source C++ project that serves as an all-in-one toolset for astrophysicists and concerned developers. it covers a wide range of astronomical subjects including calculations of orbits, celestial trigonometry, celestial mechanics, astrophysics and cosmology also powerful tools for astronomical image processing based on artificial intelligence

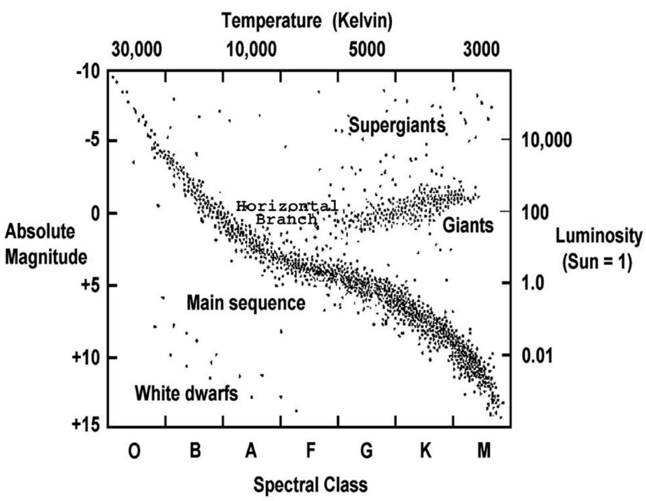
Our goal is to create a community-driven project that meets the needs of all those interested in astronomy and astrophysics. Whether you're an amateur astronomer, a researcher, a gaming group who develops a Sci-fi astronomical game or also a space company that wants to calculate the fastest and cheapest way to reach mars !

Project Scenario

QSSL is an open-source C++ project designed to be a comprehensive toolset for astronomers, astrophysicists, developers, and space enthusiasts. The library includes functionalities for calculating orbits, performing celestial trigonometry, modelling celestial mechanics, and exploring astrophysics and cosmology. Additionally, QSSL incorporates advanced tools for astronomical image processing powered by artificial intelligence (AI). The project's goal is to foster a community-driven environment that caters to a diverse range of users, from amateur astronomers to space industry professionals.

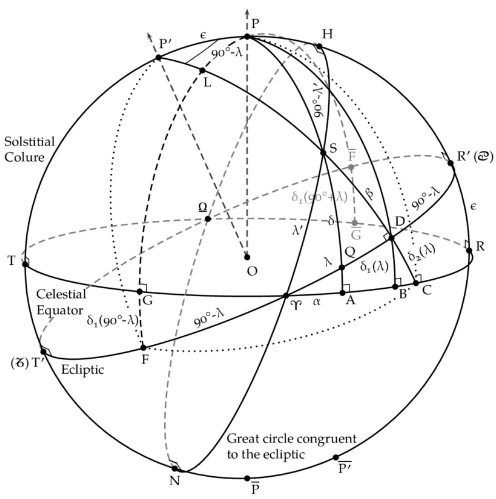


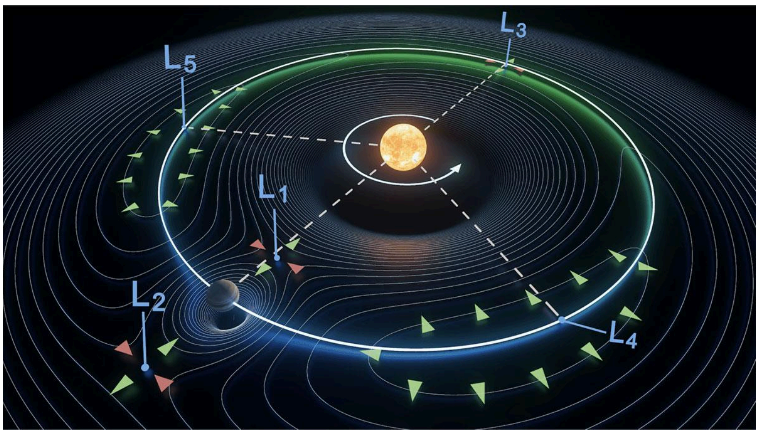




$$\left(H^2 - \frac{8}{3}\pi G\rho\right)R^2 = -kc^2$$

$$\left[\left(\frac{1}{R}\frac{dR}{dt}\right)^2 - \frac{8}{3}\pi G\rho\right]R^2 = -kc^2$$





Objectives

1. **Develop a Robust Library**
   Create a reliable and efficient C++ library that covers essential astronomical calculations and processes.
2. **Community Engagement**
   Encourage contributions from a broad user base to continuously enhance the library's capabilities.
3. **User-Friendly Documentation**
   Provide comprehensive documentation and tutorials to make the library accessible to users of all skill levels.
4. **AI Integration**
   Implement AI-based tools for enhanced astronomical image processing and data analysis.
5. **Cross-Platform Compatibility**
   Ensure the library is compatible with major operating systems and development environments.
6. **High Performance and Efficiency**
   Optimise the library for high performance, ensuring it can handle large datasets and complex calculations efficiently. Implement parallel processing and GPU acceleration where applicable.
7. **Educational Resources**
   Create educational materials to support learning and teaching in astronomy and astrophysics. Collaborate with educational institutions to integrate QSSL into their curricula and research programs.
8. **Universe Sandbox**
   Implement features that allow users to create their own virtual universes, complete with stars, planets, and other celestial bodies. Provide tools for simulating space environments and dynamics, facilitating space exploration scenarios and the design of space-themed games.

Target Audience

1. **Amateur Astronomers**
   Individuals with an interest in astronomy who need tools for observing and analyzing celestial events.
2. **Researchers**
   Scientists conducting research in astrophysics and related fields who require advanced computational tools.
3. **Game Developers**
   Developers creating sci-fi games that require accurate astronomical simulations and visualizations.
4. **Space Industries Professionals**
   Companies and organizations involved in space exploration and technology looking for reliable tools to plan missions and analyze space data.

Features and Services

1. **Orbital Calculations**
   Predicting the paths and orbit duration of celestial bodies based on gravitational forces.
2. **Spherical Trigonometry**
   Solving geometric problems related to positions and movements of celestial objects.
3. **Celestial Mechanics**
   Studying the dynamics of celestial bodies and their interactions.
4. **Astrophysics**
   Exploring the physical properties and phenomena of celestial objects and the universe.
5. **Cosmology**
   Investigating the origin, evolution, and structure of the universe on large scales.
6. **Data Analyzing**
   Analyzing astronomical data to extract meaningful insights and patterns.
7. **Image Processing**
   Enhancing and analyzing astronomical images using computational techniques.
8. **Electromagnetic Theory and Quantom Physics**
   Understanding the behavior of electromagnetic waves and particles at the quantum level.

9. **Thermodynamics**
   Studying the transfer of heat and energy in astronomical systems.
10. **Spectroscopy and Atomic Physics**
    Spectroscopy and Atomic Physics: Analyzing the spectral lines of celestial objects to study their composition and properties.
11. **Nuclear Physics**
    Understanding nuclear processes and reactions occurring in stars and other astronomical objects.

Beneficiaries Desires

Several questions about features and services have been asked from users, including astronomers, professors of astrophysics and cosmology and …

Here are their answers :

1. **Ramtin Kosari** - Computer Engineer, Amateur Astrophysicist
   "I need a reliable tool to predict celestial events accurately for my research on planetary transits. The current tools I use are either outdated or too complicated."
2. **Ramtin Kosari** - Computer Engineer, Amateur Astrophysicist
   "As a student preparing for the Astrophysics Olympiad, I need access to a comprehensive library of solved problems and detailed solutions. Tools that simulate real Olympiad scenarios and offer step-by-step problem-solving strategies would be incredibly helpful for my preparation."
3. **Ehsan Ebrahimian** - Sharif PhD Student in Cosmology, Silver Medal in IOAA
   "In my research on celestial mechanics, I need robust tools to solve the equations governing the stable orbits at Lagrange points L4 and L5. A library that provides precise computational methods and visualizations of these stable orbits would be crucial for advancing my studies."
4. **Ehsan Ebrahimian** - Sharif PhD Student in Cosmology, Silver Medal in IOAA
   "I work on developing algorithms for improving astronomical images. Access to a library with advanced noise reduction techniques such as variance stabilizers, wavelet noise filters, and Anscombe transformations would greatly aid in refining the quality of images captured by low-sensitivity cameras."

5. **Lida Molla Mohyeddin** - Amateur Astronomer

   “As an amateur astrophotographer using a weak camera, I need powerful image processing algorithms to enhance the quality of my star images. Tools that include variance stabilizers, wavelet noise filters, and Anscombe transformations to remove noise would help make my astrophotography much clearer and more detailed.”

6. **Ehsan Rostami Darestani** - Aerospace Engineer, Game Developer, Musician

   “I am developer of a space-themed game that requires accurate calculations of celestial mechanics to simulate realistic space environments and spacecraft navigation. A library that provides detailed algorithms for orbital dynamics, gravitational interactions, and real-time simulation of celestial movements would be essential for creating an engaging and scientifically accurate game experience.”

7. **Ata Moradi** - Sharif Master Student in Mechanics, Silver Medal in IOAA

   “In my work with satellite data, I need advanced tools for handling and visualizing satellite orbits. Features that allow me to accurately model, simulate, and visualize the trajectories of satellites in 3D would greatly enhance my ability to analyze orbital mechanics and predict satellite positions”

8. **Cosmologist**

   “I need creative tools for visualizing and analyzing large-scale structures in the universe. Features that allow customization of visualization parameters and interactive exploration of data would enhance my research presentations.”

9. **Chemistry Teacher**

   “In my chemical physics research, I often use spherical trigonometry to calculate angles between atoms in molecules like CH_4. Having a library with precise trigonometric functions tailored for such calculations would be a game-changer.”

Tech Stack

- **Core Components**
  - **Library Language**
    - **C++ :** This library is being developed with C++ Programming Language
    - **STL (Standard Template Library) :** Used to utilize containers, algorithms and iterators for efficient data handling
    - **Versions :** Supporting C++ versions from C++11 to C++20
  - **Mathematics and Algorithms**
    - **Boost :** Offers extensive libraries for various functionalities like numerical operations, date and time, and more.
    - **Eigen :** A C++ template library for linear algebra.
  - **Time and Date Handling**
    - **Howard Hinnant's Date Library :** A great C++ library for handling dates and times, including time zones.
- **Data Handling**
  - **File Formats**
    - **CFITSIO :** A library for reading and writing FITS (Flexible Image Transport System) files.
    - **CSV and JSON Libraries :** For handling common data formats.
  - **Database Integration**
    - **PostgreSQL :** For handling database needs
- **Visualization**
  - **GUI Libraries**
    - **QT :** For creating cross-platform graphical user interfaces.
    - **ImGui :** For immediate mode GUIs which is used for simple visualization tools.
  - **Plotting**
    - **matplotlib-cpp :** A C++ wrapper for the popular Python plotting library.
- **Performance Optimization**
  - **Parallel Computing**
    - **OpenMP :** For parallel programming.
    - **Cuda :** For GPU acceleration and Inferencing
  - **Optimization Libraries**
    - **Intel MKL :** For optimized math routines.
    - **SIMD :** Utilize SIMD instructions for performance-critical sections.

- **Development**
  - **Build Systems**
    - **CMake :** A cross-platform build system generator.
    - **Make :** Traditional build automation tool.
  - **Documentation**
    - **Doxygen :** For generating documentation from annotated C++ source code.
  - **Version Control**
    - **Git :** For source code management and version control.

Methodology

- **Methodology for Open-Source C++ Library**
  - **Selecting Methodology**
    In developing an open-source C++ library tailored for the needs of astronomers and astrophysicists, various methodologies could be considered, including Agile, Waterfall, Scrum, Spiral and more. After evaluating these options, Scrum, a subset of Agile, emerged as the most suitable approach due to its structured, iterative process that encourages continuous integration and improvement, which is ideal for the collaborative nature of open-source projects.

- **Scrum Methodology**
  - **Why Scrum ?**
    So why Scrum ? The decision to select Scrum as the development methodology for the Astronomy and Astrophysics C++ Open Source Library is based on several key reasons that align with the specific needs and workflows of astronomers, astrophysicists, and developers. These reasons ensure the project remains flexible, collaborative, and focused on delivering valuable and high-quality features.

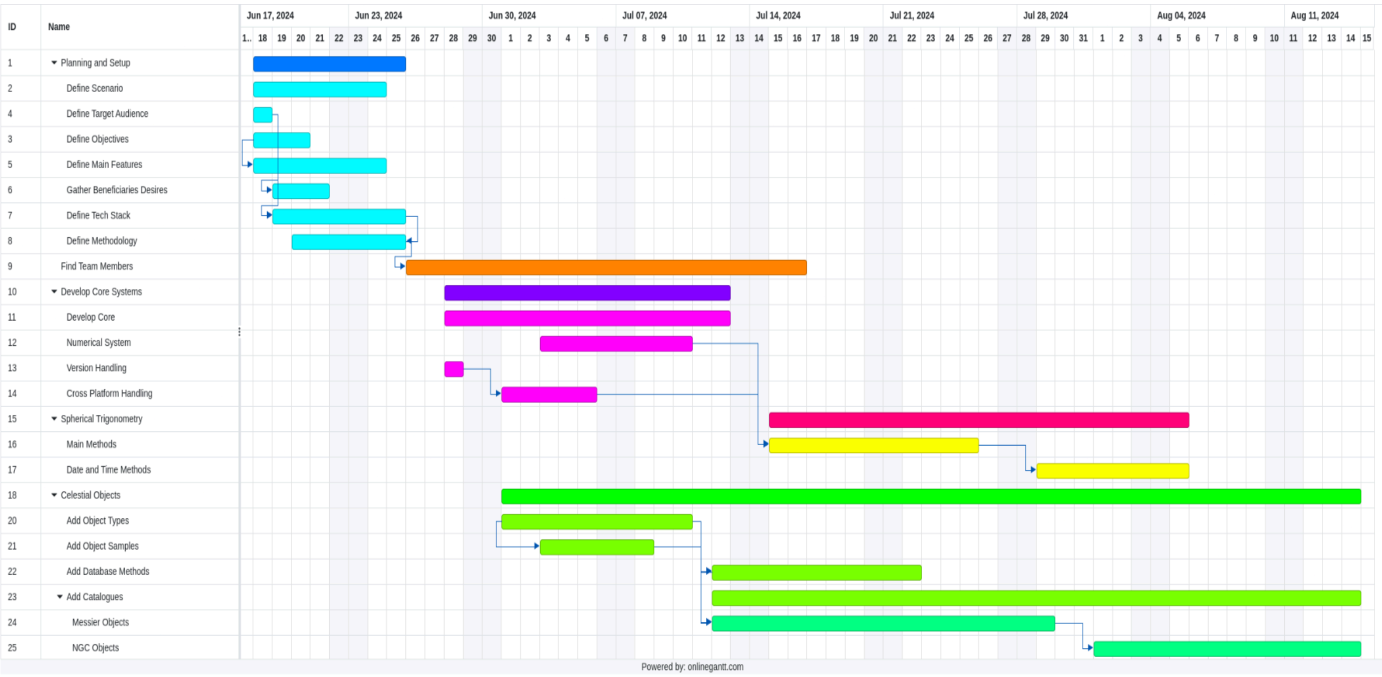
  - **Process**
    - **Iterative Development and Continuous Feedback**
      Astronomers and astrophysicists often work in iterative cycles, analyzing data, deriving conclusions, and refining their methods based on new findings. Scrum's iterative development process mirrors this approach, allowing for regular assessment and adaptation of the library’s features.

- **Collaboration and Communication**
  Astronomy and astrophysics projects often require collaboration between scientists and software developers. Scrum facilitates regular communication and collaboration through daily standups, sprint reviews, and retrospectives, ensuring that both scientific and technical perspectives are integrated effectively. Also Open-source projects thrive on community contributions. Scrum's structured approach to regular feedback loops helps maintain active engagement with the user and contributor community, fostering a collaborative environment.
- **Flexibility and Adaptability**
  The field of astronomy and astrophysics is constantly evolving with new discoveries and technological advancements. Scrum's flexibility allows the development team to quickly adapt to new requirements or changes in project scope, ensuring the library remains relevant and up-to-date.
- **Risk Management**
  The frequent iterations in Scrum help in early detection and resolution of issues, reducing the risk of major problems later in the project. This is particularly important in scientific software, where errors can significantly impact research outcomes.

- **Team Structures**
  - **Core Team**
    - Product Manager
    - Developers
      - Heads
      - Library Developers
      - Testers/QA
  - **Scientific Advisors**
  - **Community Contributors**

- ## Phases

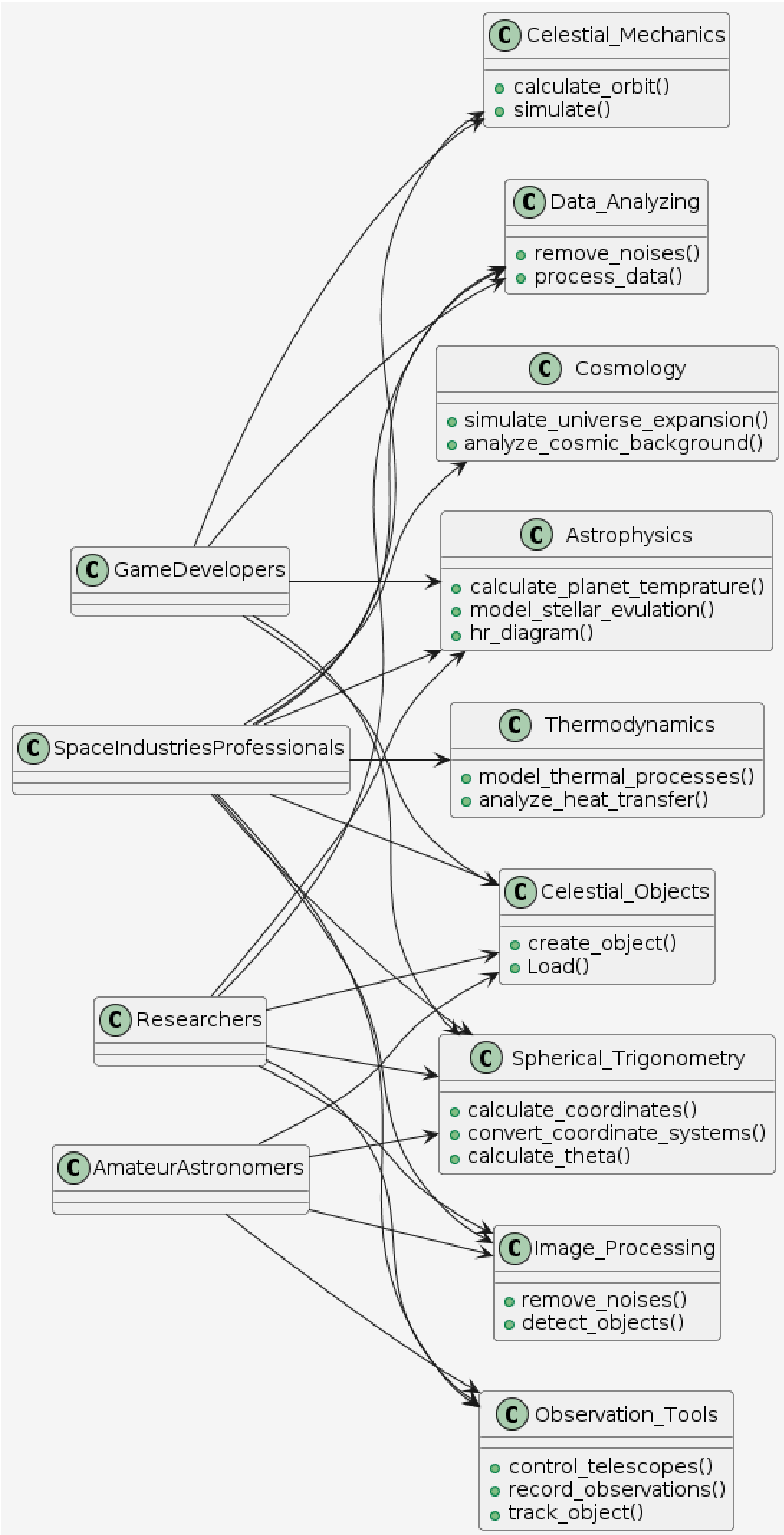
Here is Gantt Diagram of Project for ‘Nova’ version of QSSL



File of this Gantt Chart exists in projects assets directory or attached in email, in this Gantt Chart for Nova version of QSSL it takes about 2 months to finish.

- **UML (Use Case Diagram)**

Here is Use Case Diagram of QSSL Project



Some functions and methods are mentioned in this diagram as you see in classes.

- **Activity Diagram**

Here is Activity Diagram of using QSSL Library