

Stellarium

Github: <https://github.com/Stellarium/stellarium>

Table of Content

<u>1</u>	<u>TECHNOLOGY AND PLATFORM</u>	<u>2</u>
1.1	LANGUAGE	2
1.2	BUILD SYSTEM	2
1.3	LIBRARY	2
<u>2</u>	<u>TESTING.....</u>	<u>2</u>
2.1	CODE COVERAGE METRICS	2
2.2	CI PLATFORM.....	2
2.3	COMPUTING PLATFORM COMBINATION TESTED.....	3
<u>3</u>	<u>SOFTWARE ARCHITECTURE.....</u>	<u>3</u>
3.1	FUNCTIONALITY	3
3.2	ASYNCHRONOUS	3
3.3	DIAGRAM.....	4
3.4	SEPARATION OF CONCERN	4
3.5	ARCHITECTURAL PATTERNS	4
3.6	OBJECT ORIENTED	4
<u>4</u>	<u>ANALYZE ISSUES.....</u>	<u>5</u>
4.1	SHOW MORE DETAILS	5
4.2	REAL TIME SUNSPOT	5
<u>5</u>	<u>DEMONSTRATION APPLICATION</u>	<u>5</u>
<u>6</u>	<u>REFERENCE:</u>	<u>6</u>

1 Technology and Platform

1.1 Language

Stellarium is a software using C and C-plus-plus. If the project starts today, C and C++ will still be used because it is supported by almost all operating systems. Python or JAVA might be some other options for this project. I prefer Python since it is a easier language to write and read.

1.2 Build System

The build system used is CMake. It uses Visual Studio as build tool.

1.3 Library

QtWebApp Library. It is a HTTP server library in C++ inspired by Java Servlets). For Linux, Windows, Mac OS and many other operating systems that the Qt Framework supports.

QtWebApp contains the following components

- HTTP Server
- Template Engine
- File Logger
- Windows Service Installer

2 Testing

2.1 Code Coverage metrics

To ensure the test is meaningful, they use Coveralls to test code coverage metrics. Coverity, consisting primarily of static code analysis tools, is also used to find defects and security vulnerabilities in custom source code.

2.2 CI platform

They use Travis-CI to test the units.

2.3 Computing Platform combination tested

Linux, Mac, GCC, Clang

3 Software architecture

3.1 Functionality

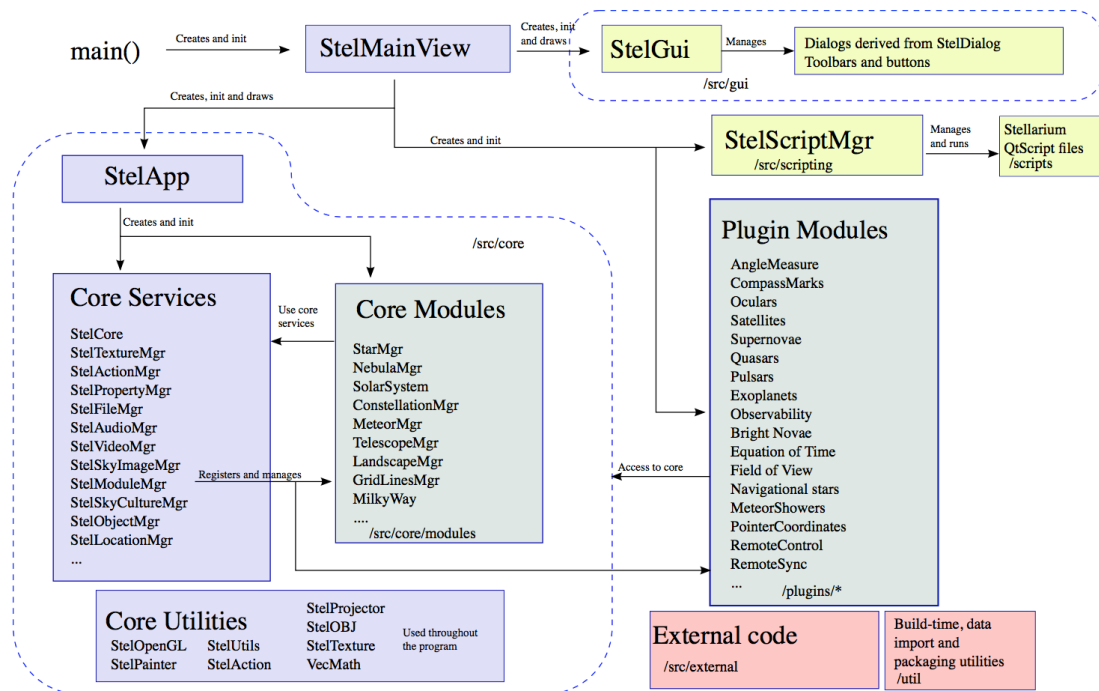
First we have to know the main function is StelMainView. This single-instance class is created at startup by the main() function. It hosts the —main OpenGL viewport (based upon QGraphicsView) and manages StelApp and StelGui as subobjects.

If we want to add/edit functionality, we can edit code in the corresponding module. And then edit code in main function if necessary. For external projects wants to use this project, they are free to call the corresponding module, such as StelModuleMgr and StelCore. The first one manages the collection of StelModule instances which are registered in the program. The second one provides performance critical features for rendering various elements using openGL, or for computing coordinate transformation and other mathematical services.

3.2 Asynchronous

There's no asynchronous part.

3.3 Diagram



3.4 Separation of Concern

The modules are separated into several modules. Including stellar's texture, location, action, property, sky culture, audio, video, other plug in modules and so on.

3.5 architectural patterns

I think the architectural patterns is Model-view-controller pattern. Because the project has an app for users to take a view the sky. Then switch to any stellar user select and display more detailed information of that stellar.

3.6 Object Oriented

Stellarium is object oriented.

4 Analyze issues

4.1 Show More details

There's one issue suggesting showing more details when an object has been selected.

- Lower/Upper Transit (Meridian) : Time, Az, Alt, Disk Duration
- Lower/Upper Culmination (Max/Min Alt) : Time, Az, Alt, Disk Duration
- Mid Day/Night (Half Set-Rise) : Time, Az, Alt
- Rise/Set (at alt=0 and at visible alt) : Time, Az, Alt, Disk Duration This issue needs developer to add some new function.

4.2 Real Time Sunspot

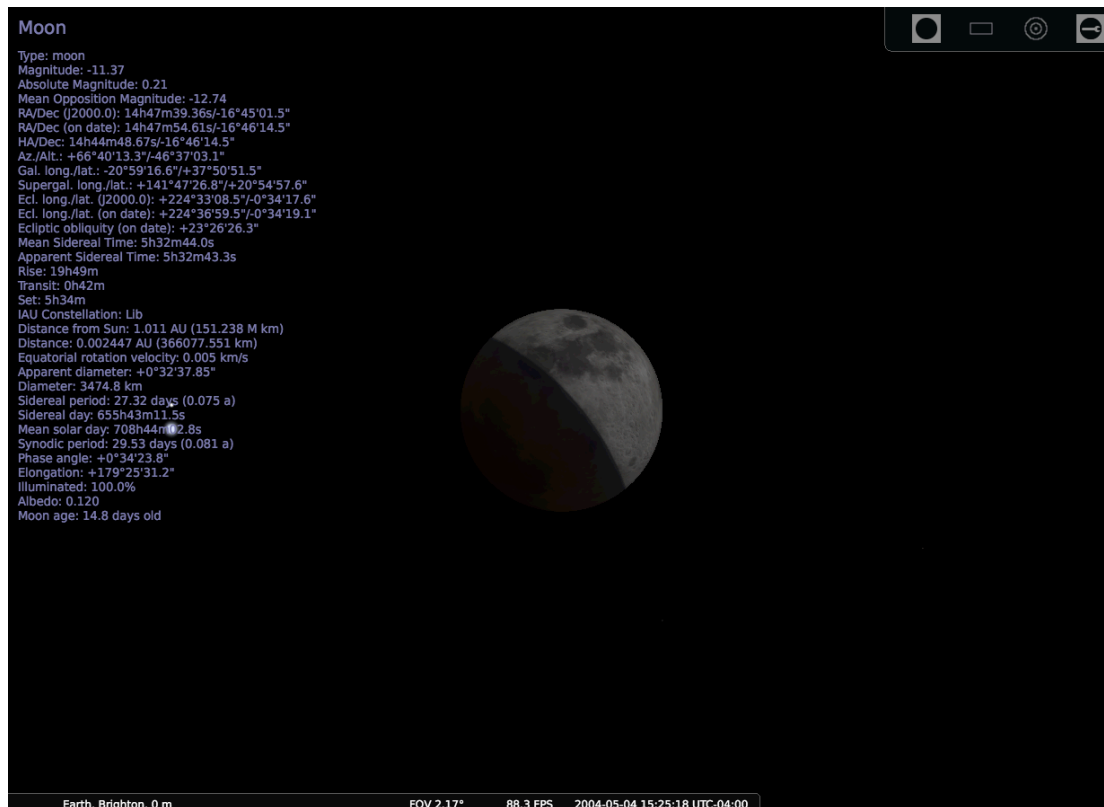
This issue is asking for if it is possible to have real time sunspot location. It needs to add some new function.

Spaceweather.com have real time sunspot location on the sun and the sunspot ID. It can be looked as a reference and used as plug in module.

4 Demonstration Application

The script below is used to display the process of lunar eclipse. The screenshot is the picture of lunar eclipse at a certain moment.

```
1. core.wait(1);
2. core.clear("natural");
3. LandscapeMgr.setFlagLandscape(false);
4. LandscapeMgr.setFlagAtmosphere(false);
5. LandscapeMgr.setFlagFog(false);
6. core.wait(1);
7. core.setDate("2004:05:04T17:51:00");
8. core.selectObjectByName("Moon", false);
9. StelMovementMgr.setFlagTracking(true);
10. core.wait(2);
11. StelMovementMgr.autoZoomIn(2);
12. core.wait(2);
13. core.setTimeRate(250);
```



6 Reference:

Github location

<https://github.com/Stellarium/stellarium>

QtWebApp HTTP Server in C++

<http://stefanfrings.de/qtwebapp/index-en.html>

main page of Stellarium

<http://stellarium.org/doc/0.18/index.html>