

StellariumLG

Developer Documentation

INTERACTIVE SPACES

StellariumMasterActivity.java

Location:

StellariumLG/InteractiveSpaces/stellarium.master/src/main/java/stellarium/master/

Public Methods

onActivitySetup ()

Defines the socket for communication with Stellarium instance. Also defines **stellariumListener** thread.

onActivityStartup ()

Launces Stellarium instance based on **space.nativeapplication.executable** and **space.nativeapplication.executable.flags**. Launch the **stellariumListener** thread.

onActivityShutdown ()

Unbind the sockets and shutdown the Stellarium instance.

Private Methods

Thread stellariumListener

Listens to the stellarium instance on the socket and forwards the received message as a JSON message to all clients instances listening on a channel.

StellariumClientActivity.java

Location:

StellariumLG/InteractiveSpaces/stellarium.client/src/main/java/stellarium/client/

Public Methods

onActivitySetup ()

Defines the socket for communication with Stellarium instance. Also defines **stellariumListener** thread.

onActivityStartup ()

Launces Stellarium instance based on **space.nativeapplication.executable** and **space.nativeapplication.executable.flags**. Launch the **stellariumListener** thread.

onNewInputJson (String , Map)

Receives message from master activity on the input channel and forwards the message to the client Stellarium instance.

onActivityShutdown ()

Unbind the sockets and shutdown the Stellarium instance.

STELLARIUM

LGCommunicate

File: StellariumLG/Stellarium/src/core/LGCommunicate.hpp

Public Methods

LGCommunicate (StelCore*, StelMovementMgr*, MODE m, int _offset, string _port)

Initialize the socket communication with Interactive Spaces. Also initialise the SERVER or CLIENT mode with respective offset when using with LG support. Mode is NONE when Stellarium is run as standalone application.

write (Vec3d viewdirection)

Store the ***viewdirection*** to be sent to clients.

write (double fov)

Store the ***fov*** to be sent to clients.

send ()

Send the ***viewdirection*** and ***fov*** to the clients.

listen ()

Receive, parse and process the data received from the master.

read ()

Retrieve the stored ***viewdirection*** and ***fov*** and create the viewing matrix.

Slots

sendTimeRate (double rate)

Send the time rate to the clients. Connected to SIGNAL ***timeRateChanged (rate)***.

sendTimeReset ()

Send time reset signal to the clients. Connected to SIGNAL ***timeReset***.

StelCore::setTimeNow emits ***timeReset*** signal.

Public Enumerator

```
enum MODE {  
    NONE,  
    SERVER,  
    CLIENT  
}
```

LG mode for Stellarium.

Private Attributes

bool viewchanged

True, when new **viewdirection** and **fov** have been received.

False, when viewing matrix is in sync with **viewdirection** and **fov**.

LGListenerThread

File: StellariumLG/Stellarium/src/core/LGCommunicate.hpp

Create a listener thread which executes **LGCommunicate::listen**

StelCore

File: StellariumLG/Stellarium/src/core/StelCore.hpp

lookAtJ2000 (const Vec3d& pos, const Vec3d& aup, int offset)

Compute the view matrix based on **offset**. **pos** is rotated about **aup** vector by (**offset** * **fov**) degrees for each client before computing the view matrix.