|  |
| --- |
| Oculus RC Buggy |
| Submitted By: Jack Sanchez |
| Submitted To: Calvin Caldwell |
| E-mail: jack.sanchez@oit.edu |
| Date: Tuesday, Oct 21st 2014 |
| Version: 1.0 |
| Responsible Analyst: Jack Sanchez |
|  |
|  |

# 1.0 Signatory Page\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Submitted By: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_\_

Company: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Job Title: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Approved By: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_\_

Company: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Job Title: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# 2.0 Revision History\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Author** | **Company** | **Version** | **Date** | **File Name** | **Comments** |
| J. Sanchez | OculusRC | <0.1> | Nov. 16, 2014 | ObjectModel.docx | Initial Object Model |
| J. Sanchez | OculusRC | <0.5> | Nov. 30, 2014 | ObjectModel.docx | Object model revision |
|  |  |  |  |  |  |

# 3.0 Table of Contents\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Contents

[1.0 Signatory Page\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 1](#_Toc405087049)

[2.0 Revision History\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2](#_Toc405087050)

[3.0 Table of Contents\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 3](#_Toc405087051)

[4.0 Executive Class Diagram\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 5](#_Toc405087052)

[5.0 Detailed Class Diagrams\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 6](#_Toc405087053)

[5.1 Business Layer\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 6](#_Toc405087054)

[5.2 Data Layer\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 7](#_Toc405087055)

[6.0 Class Specifications\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 8](#_Toc405087056)

[6.1 ClientManager\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 8](#_Toc405087057)

[6.2 ControllerManager\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 10](#_Toc405087058)

[6.3 OculusManager\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 12](#_Toc405087059)

[6.4 NetworkManager\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 14](#_Toc405087060)

[6.5 VideoManager\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 16](#_Toc405087061)

[6.6 VideoStreamManager\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 18](#_Toc405087062)

[6.7 ControllerStreamManager\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20](#_Toc405087063)

[6.8 ServerControlManager\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 22](#_Toc405087064)

[6.9 RTSPClient\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 24](#_Toc405087065)

[6.10 StreamClientState\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 26](#_Toc405087066)

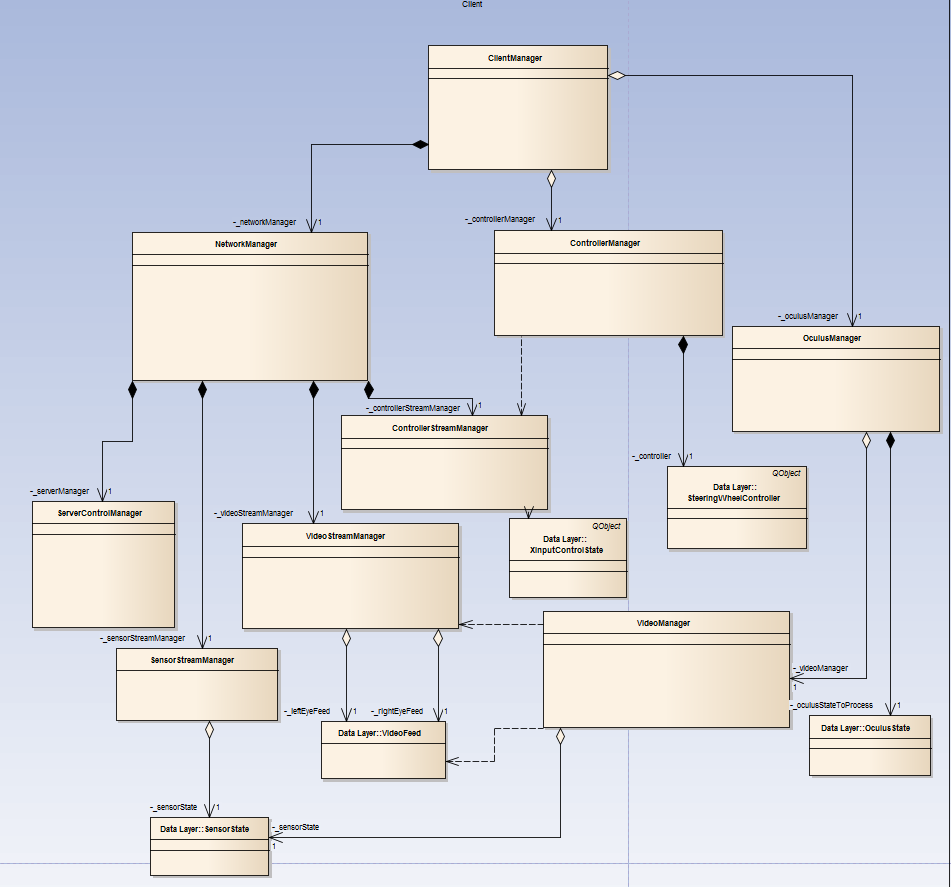
[6.11 SteeringWheelController\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 28](#_Toc405087067)

[6.12 XInputControlState\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 31](#_Toc405087068)

[6.13 OculusState\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 33](#_Toc405087069)

[6.14 RCManager\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 35](#_Toc405087070)

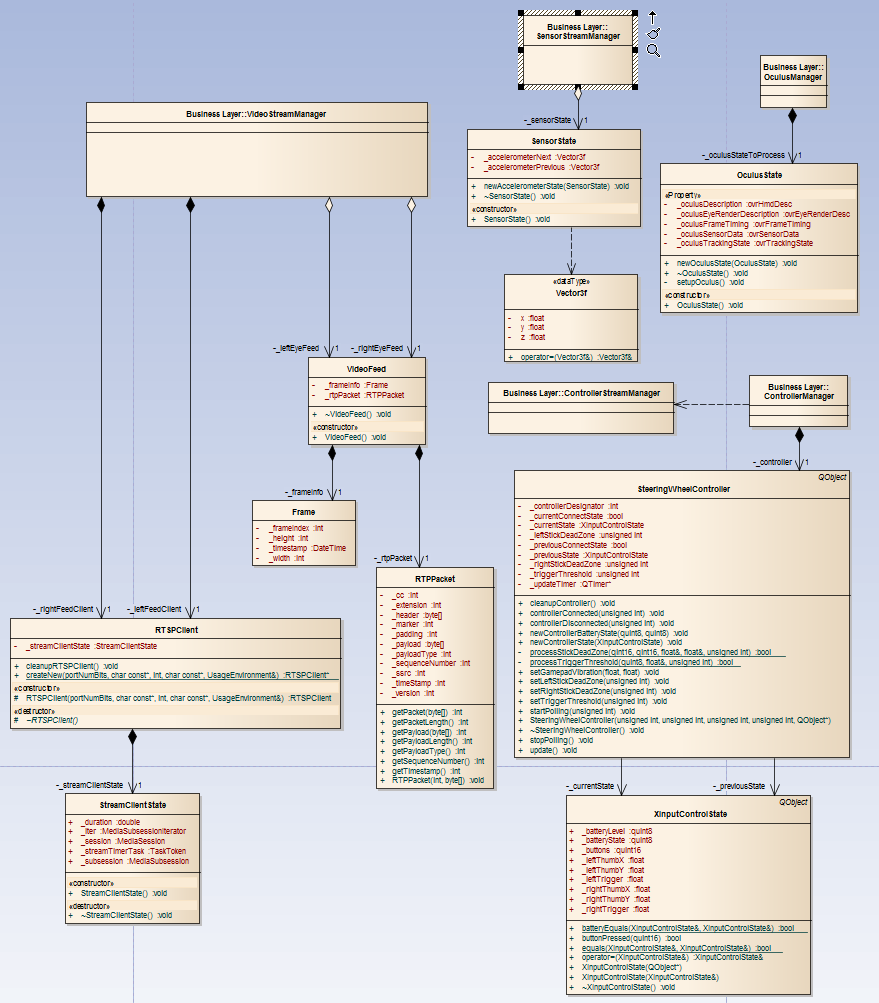
# 4.0 Executive Class Diagram\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



# 5.0 Detailed Class Diagrams\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## 5.1 Business Layer\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## 5.2 Data Layer\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



# 6.0 Class Specifications\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## 6.1 ClientManager\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| **Class Information** |
| **Class Name:** ClientManager  Abstract Type: No Persistence: No |

|  |  |  |
| --- | --- | --- |
| **Trace-ability Information** | | |
| **Use Case ID** | **Use Case Name** | **Steps** |
|  |  |  |
|  |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Public Methods** | | | | | | | |
| **Signature** | **Description** | | **Collaborating Class::Methods** | **Preconditions** | | | **Post Conditions** |
| systemInit() | Initialize all system dependencies | NetworkManager::initServerConnection  ControllerManager::controllerInit  OculusManager::oculusInit  VideoManager::initVideo | | The system has not yet been initialized. The server system is ready for operation | | | The system is now initialized |
| systemCleanup() | Releases all system resources | NetworkManager::cleanupNetwork  ControllerManager::cleanupController  OculusManager::cleanupOculus  VideoManager::cleanupVideo | | The system has been running and has not encountered an exception condition | | | The system will have released all resources and be ready to gracefully exit |
| conectServer() | Manually toggles the connection to the Jetson TK1 server module | NetworkManager::toggleServerConnection | | This method has no preconditions, it is intended to handle all cases of connection | | | If the server was connected, it is disconnected.  If the server was disconnected, it is connected.  If the server is unavailable, the method returns false. |
| connectOculus() | Manually toggles the connection to the Oculus Rift HMD | OculusManager::toggleOculusConnection | | This method has no preconditions, it is intended to handle all cases of connection | | | If the Oculus was connected, it is disconnected.  If the Oculus was disconnected, it is connected.  If the Oculus is unavailable, the method returns false. |
| connectController() | Manuallly toggles the connection to any XInput compatible controller | ControllerManager::toggleControllerConnection | | | This method has no preconditions, it is intended to handle all cases of connection | If a controller is connected, it is disconnected.  If a controller is disconnected, it is connected.  If a controller is unavailable, the method returns false. | |
|  |  | |  |  | | |  |
|  |  | |  |  | | |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Protected Methods** | | | |
| **Signature** | **Description** | **Preconditions** | **Post Conditions** |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Private Methods** | | | |
| **Signature** | **Description** | **Preconditions** | **Post Conditions** |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Attributes** | | | | | |
| **Name** | **Type** | **Object (Y/N)** | **Instance/Static** | **Visibility** | **Description** |
| \_controllerManager | ControllerManager | Y | Instance | Private | Manages the lifecycle of an XInput compatible controller |
| \_networkManager | NetworkManager | Y | Instance | Private | Manages the lifecycle of all network interactions |
| \_oculusManager | OculusManager | Y | Instance | Private | Manages the lifecycle of an Oculus Rift peripheral device |

|  |  |
| --- | --- |
| **Concurrency** | |
| **Threading Issue** | **Description** |
| N/A | The Qt Framework, by nature, employs the observer pattern on all C++ objects which are exposed to the QML engine. This allows all asynchronous operations to be handled asynchronously and in a highly decoupled manner. |
|  |  |

|  |  |  |
| --- | --- | --- |
| **Major Exceptions** | | |
| **Name** | **Trigger** | **Action** |
| SystemResourceError | Erroneous object handles during cleanup method | Safely retry the cleanup, if second failure, let OS cleanup |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Instance Information** | | | |
| **Minimum** | **Maximum** | **Mean** | **Fixed** |
| 1 | 1 | 1 | 1 |
|  |  |  |  |

|  |  |  |
| --- | --- | --- |
| **General Comments** | | |
| **Author** | **Date** | **Comment** |
| Sanchez | 11/17/2014 | Initial draft of ClientManager class specification |
|  |  |  |

## 6.2 ControllerManager\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| **Class Information** |
| **Class Name:** ControllerManager  Abstract Type: No Persistence: No |

|  |  |  |
| --- | --- | --- |
| **Trace-ability Information** | | |
| **Use Case ID** | **Use Case Name** | **Steps** |
|  |  |  |
|  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Public Methods** | | | | | | | | | | |
| **Signature** | **Description** | | | | | | **Collaborating Class::Methods** | **Preconditions** | | **Post Conditions** |
| cleanupController() | Closes and releases all system resources related to the controller | | | | ControllerStreamManager::sendControllerState | | | The controller had previously been connected | | The controller is no longer connected |
| controllerInit() | Create all XInput resources and prepare for controller input | | | | | | N/A | Manager object did not already exist | | The manager object has initialized required resources and is ready to be updated with a connected controller |
| processControllerState(XInputControlState) | | Handle any controller events found by a timer driven monitor | | | | SteeringWheelController::newControllerState  SteeringWheelController::newControllerBatteryState | | | The startPolling method has already been invoked and the system has emitted the signal that the controller state has changed | The system has encountered and processed the controller input and has invoked the appropriate behavior |
| readySendControllerState(XInputControlState) | | | Notifies the ControllerStreamManager that the new control input is ready to be shipped over the network | | | | ControllerStreamManager::sendControllerState | A controller event has occurred and been processed | | The controller event has been sent to the ControllerStreamManager and is sent over the network to the RC server system |
| toggleControllerConnection() | | | | Manually connects or disconnects an XInput controller | | | N/A | There are no preconditions, this method is intended to handle any connection configuration | | If the controller was connected, it is disconnected.  If the controller was disconnected, it is connected.  If no controller is available, the method returns false. |
|  |  | | | | | |  |  | |  |
|  |  | | | | | |  |  | |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Protected Methods** | | | |
| **Signature** | **Description** | **Preconditions** | **Post Conditions** |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Private Methods** | | | |
| **Signature** | **Description** | **Preconditions** | **Post Conditions** |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Attributes** | | | | | |
| **Name** | **Type** | **Object (Y/N)** | **Instance/Static** | **Visibility** | **Description** |
| \_controller | QObject \* | Y | Instance | Private | Any XInput controller class will have derived from a QObject, the ControllerManager can hold many different XInput compatible controllers |
|  |  |  |  |  |  |

|  |  |
| --- | --- |
| **Concurrency** | |
| **Threading Issue** | **Description** |
| Minor | The controller system listens for the controller implementation class to notify that a controller event has been fired |
|  |  |

|  |  |  |
| --- | --- | --- |
| **Major Exceptions** | | |
| **Name** | **Trigger** | **Action** |
| SystemResourceError | cleanupController | Retry the disposal of resources, if second failure, let OS handle the cleanup |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Instance Information** | | | |
| **Minimum** | **Maximum** | **Mean** | **Fixed** |
| 1 | 1 | 1 | 1 |
|  |  |  |  |

|  |  |  |
| --- | --- | --- |
| **General Comments** | | |
| **Author** | **Date** | **Comment** |
| Sanchez | 11/17/2014 | Initial draft of the ControllerManager specification |
|  |  |  |

## 6.3 OculusManager\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| **Class Information** |
| **Class Name:** OculusManager  Abstract Type: No Persistence: No |

|  |  |  |
| --- | --- | --- |
| **Trace-ability Information** | | |
| **Use Case ID** | **Use Case Name** | **Steps** |
|  |  |  |
|  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Public Methods** | | | | |
| **Signature** | **Description** | **Collaborating Class::Methods** | **Preconditions** | **Post Conditions** |
| cleanupOculus() | Releases all Oculus resources | VideoManager::cleanupVideo | The Oculus has already been connected | The Oculus is now disconnected and no resources remain |
| displayToOculus(VideoFeed, VideoFeed) | Slot method invoked by the VideoManager::readyForOculus(VideoFeed, VideoFeed) signal | VideoManager::readyForOculus | The Oculus has already been connected, and the video stream over the network has begun | The current set of frames from the VideoFeed objects are now displayed on the Oculus Rift |
| oculusInit() | Initializes all members and resources for the Oculus HMD | VideoManager::initVideo | The Oculus has not yet been connected nor initialized | The Oculus Rift is connected and initialized |
| toggleOculusConnection() | Manually toggles the connection state of the Oculus peripheral | N/A | No preconditions, this method is designed to handle all scenarios | If the Oculus is connected, it is disconnected.  If the Oculus is disconnected, it is connected.  If the Oculus is unavailable, the method returns false. |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Protected Methods** | | | |
| **Signature** | **Description** | **Preconditions** | **Post Conditions** |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Private Methods** | | | |
| **Signature** | **Description** | **Preconditions** | **Post Conditions** |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Attributes** | | | | | |
| **Name** | **Type** | **Object (Y/N)** | **Instance/Static** | **Visibility** | **Description** |
| \_oculusStateToProcess | OculusState | Y | Instance | Private | Custom OculusState object to keep track of the lifecycle and status of the HMD |
| \_videoManager | VideoManager | Y | Instance | Private | Video Manager class to handle the flow of the video stream from the network to the Oculus HMD |

|  |  |
| --- | --- |
| **Concurrency** | |
| **Threading Issue** | **Description** |
| Minor | This object flow shall live on its own thread to allow asynchronous HMD tracking |
|  |  |

|  |  |  |
| --- | --- | --- |
| **Major Exceptions** | | |
| **Name** | **Trigger** | **Action** |
| OculusNotAvailable | displayToOculus(VideoFeed, VideoFeed) | Inform the user that the Rift has become disconnected |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Instance Information** | | | |
| **Minimum** | **Maximum** | **Mean** | **Fixed** |
| 1 | 1 | 1 | 1 |
|  |  |  |  |

|  |  |  |
| --- | --- | --- |
| **General Comments** | | |
| **Author** | **Date** | **Comment** |
| Sanchez | 11/17/2014 | Initial draft of the OculusManager class specification |
|  |  |  |

## 6.4 NetworkManager\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| **Class Information** |
| **Class Name:** NetworkManager  Abstract Type: No Persistence: No |

|  |  |  |
| --- | --- | --- |
| **Trace-ability Information** | | |
| **Use Case ID** | **Use Case Name** | **Steps** |
|  |  |  |
|  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Public Methods** | | | | |
| **Signature** | **Description** | **Collaborating Class::Methods** | **Preconditions** | **Post Conditions** |
| cleanupNetwork() | Closes and disposes of all network resources and pipelines | ServerControlManager::cleanupServer  ControllerStreamManager::cleanupControllerStream  VideoStreamManager::cleanupVideoStream  SensorStreamManager::cleanupSensorStream | The system networking components have all been connected and running | The system networking components have all had their resources released and cleaned up |
| initServerConnection() | Creates and initializes all networking resources | ServerControlManager::initializeServer | The server has not already been initialized | The server has been initialized and is ready to begin streaming |
| toggleServerConnection() | Manually connects or disconnects the server system | ServerControlManager::toggleConnection | No preconditions exist for this method, it is intended to handle all connection scenarios | If the server is connected, it is disconnected.  If the sever is disconnected, it is connected.  If the server is unavailable, method returns false. |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Private Methods** | | | |
| **Signature** | **Description** | **Preconditions** | **Post Conditions** |
| initControllerStream() | Creates and initializes the controller stream from the client to the server | The controller stream has not yet been initialized | The controller stream is ready to begin sending and receiving data |
| initSensorStream() | Creates and initializes the sensor stream from the server to the client | The sensor stream has not yet been initialized | The sensor stream is ready to begin sending and receiving data |
| initVideoStream() | Creates and initializes the video stream from the server to the client | The video stream has not yet been initialized | The video stream is ready to begin sending and receiving data |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Attributes** | | | | | |
| **Name** | **Type** | **Object (Y/N)** | **Instance/Static** | **Visibility** | **Description** |
| \_controllerStreamManager | ControllerStreamManager | Y | Instance | Private | Object to manage the lifecycle of the controller stream |
| \_sensorStreamManager | SensorStreamManager | Y | Instance | Private | Object to manage the lifecycle of the sensor stream |
| \_serverManager | ServerControlManager | Y | Instance | Private | Object to maintain the server connection |
| \_videoStreamManager | VideoStreamManager | Y | Instance | Private | Object to manage the lifecycle of the video streams |

|  |  |
| --- | --- |
| **Concurrency** | |
| **Threading Issue** | **Description** |
| Major | This module of the system requires a high precision concurrency solution to manage the four simultaneous network streams and properly disperse the data to the appropriate manager classes |
|  |  |

|  |  |  |
| --- | --- | --- |
| **Major Exceptions** | | |
| **Name** | **Trigger** | **Action** |
| UnexpectedClosedPipeline | If any network methods fail based on an erroneous network endpoint | Attempt to reopen the pipeline, if second failure, inform user that a connection problem exists |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Instance Information** | | | |
| **Minimum** | **Maximum** | **Mean** | **Fixed** |
| 1 | 1 | 1 | 1 |
|  |  |  |  |

|  |  |  |
| --- | --- | --- |
| **General Comments** | | |
| **Author** | **Date** | **Comment** |
| Sanchez | 11/17/2014 | Initial NetworkManager class specification |
|  |  |  |

## 6.5 VideoManager\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| **Class Information** |
| **Class Name:** VideoManager  Abstract Type: No Persistence: No |

|  |  |  |
| --- | --- | --- |
| **Trace-ability Information** | | |
| **Use Case ID** | **Use Case Name** | **Steps** |
|  |  |  |
|  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Public Methods** | | | | | | |
| **Signature** | **Description** | | **Collaborating Class::Methods** | | **Preconditions** | **Post Conditions** |
| cleanupVideo() | Frees all resources related to the video manipulation on the client system | | N/A | | The VideoManager has previously been initialized and used | The VideoManager object has no more resources allocated |
| initVideo() | Creates and initializes all video management objects on the client system | | VideoFeed::VideoFeed | | The VideoManager has not yet been created | The VideoManager and both Oculus video feeds have been initialized and are ready to be sent to the Oculus |
| processNextFrameSet(VideoFeed, VideoFeed) | | Slot method invoked when the newFeedSet signal is emitted by the VideoStreamManager | VideoStreamManager::newFeedSet | The VideoStreamManager has properly been created and connected to the Server system and the video stream has been started | | The current frame set has been processed, any heads-up display overlay has been drawn and processed, and the frame set is ready to be drawn on the Oculus Rift HMD |
| readyForOculus(VideoFeed, VideoFeed) | Signal emitted by the processNextFrame method once the frame set is ready to ship to the Oculus | | OculusManager::displayToOculus | | The Oculus Rift has been properly connected and initialized and the frame set has been processed by the VideoManager | The Oculus Rift will be displaying the latest processed frame set sent by the VideoManager |
|  |  | |  | |  |  |
|  |  | |  | |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Protected Methods** | | | |
| **Signature** | **Description** | **Preconditions** | **Post Conditions** |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Private Methods** | | | |
| **Signature** | **Description** | **Preconditions** | **Post Conditions** |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Attributes** | | | | | |
| **Name** | **Type** | **Object (Y/N)** | **Instance/Static** | **Visibility** | **Description** |
| \_leftEyeFeed | VideoFeed | Y | Instance | Private | The data container for the OpenGL video frame data and descriptors about the state of the frame data as well as video timestamps and frame indexes |
| \_rightEyeFeed | VideoFeed | Y | Instance | Private | The data container for the OpenGL video frame data and descriptors about the state of the frame data as well as video timestamps and frame indexes |
| \_sensorState | SensorState | Y | Instance | Private | The data container, which is extensible, for all current sensor data to be drawn on the Oculus Rift display |

|  |  |
| --- | --- |
| **Concurrency** | |
| **Threading Issue** | **Description** |
| Major | This object will have to manage the flow of data of 3 separate streams, all of which need as close to real-time processing as possible for the ideal situation. This object will need to be highly optimized. |
|  |  |

|  |  |  |
| --- | --- | --- |
| **Major Exceptions** | | |
| **Name** | **Trigger** | **Action** |
| See general comment section. | N/A | N/A |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Instance Information** | | | |
| **Minimum** | **Maximum** | **Mean** | **Fixed** |
| 1 | 1 | 1 | 1 |
|  |  |  |  |

|  |  |  |
| --- | --- | --- |
| **General Comments** | | |
| **Author** | **Date** | **Comment** |
| Sanchez | 11/17/2014 | This object will not generate any internal exceptions. It only receives data if the stream manager sends the signal and then if the data is processed erroneously, it will either cause a problem in the OculusManager object, or will merely display a “broken” image on the HMD. The plan is to run a watchdog of sorts to monitor the stream for error and, if detected, display a black screen with any error information to the user. |
|  |  |  |

## 6.6 VideoStreamManager\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| **Class Information** |
| **Class Name:** VideoStreamManager  Abstract Type: No Persistence: No |

|  |  |  |
| --- | --- | --- |
| **Trace-ability Information** | | |
| **Use Case ID** | **Use Case Name** | **Steps** |
|  |  |  |
|  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Public Methods** | | | | | |
| **Signature** | **Description** | | **Collaborating Class::Methods** | **Preconditions** | **Post Conditions** |
| cleanupVideoStream() | | Frees the resources used in the VideoStreamManager | VideoFeed::~VideoFeed | The video stream has been running and received the shutdown signal | The video stream is offline and the object has been destroyed and resources freed |
| newFeedSet(VideoFeed, VideoFeed) | Signal which will be emitted to invoke the processNextFrameSet method | | VideoManager::processNextFrameSet | The video stream has been initialized and is receiving video feeds from the server system | The signal has been emitted and the VideoManager has been given the video feed data for processing |
| startVideoStream() | Method invoked when the user begins driving to send the start signal to the server | | ServerControlManager::startCameras  VideoStreamManager::startCameras | The video stream has not yet been running and the server is connected | The video stream has been started and the client system is receiving video data |
| startCameras() | Signal which will be emitted to invoke the slot method startCameras | | ServerControlManager::startCameras | The cameras have not yet been started and the server is connected | The cameras are running and a video stream is being received by the client video streaming system |
|  |  | |  |  |  |
|  |  | |  |  |  |
|  |  | |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Protected Methods** | | | |
| **Signature** | **Description** | **Preconditions** | **Post Conditions** |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Private Methods** | | | |
| **Signature** | **Description** | **Preconditions** | **Post Conditions** |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Attributes** | | | | | | |
| **Name** | **Type** | | **Object (Y/N)** | **Instance/Static** | **Visibility** | **Description** |
| \_leftEyeFeed | VideoFeed | | Y | Instance | Private | The container receiver for the left video stream |
| \_rightEyeFeed | | VideoFeed | Y | Instance | Private | The container receiver for the right video stream |

|  |  |
| --- | --- |
| **Concurrency** | |
| **Threading Issue** | **Description** |
| Moderate | This object will be blocked unless the video stream is live, at that point it will be receiving video data as quickly as possible and signaling the VideoManager for processing |
|  |  |

|  |  |  |
| --- | --- | --- |
| **Major Exceptions** | | |
| **Name** | **Trigger** | **Action** |
| ServerNotConnected | startCameras | Inform user that the server must be connected in order to start the camera stream |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Instance Information** | | | |
| **Minimum** | **Maximum** | **Mean** | **Fixed** |
| 1 | 1 | 1 | 1 |
|  |  |  |  |

|  |  |  |
| --- | --- | --- |
| **General Comments** | | |
| **Author** | **Date** | **Comment** |
| Sanchez | 11/17/2014 | Initial draft of VideoStreamManager class |
|  |  |  |

## 6.7 ControllerStreamManager\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| **Class Information** |
| **Class Name:** ControllerStreamManager  Abstract Type: No Persistence: No |

|  |  |  |
| --- | --- | --- |
| **Trace-ability Information** | | |
| **Use Case ID** | **Use Case Name** | **Steps** |
|  |  |  |
|  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Public Methods** | | | | | |
| **Signature** | **Description** | | **Collaborating Class::Methods** | **Preconditions** | **Post Conditions** |
| cleanupControllerStream() | | The method invoked by the NetworkManager which will release and close all resources for the controller stream | NetworkManager::cleanupNetwork | The controller stream has been running and the user sent the signal to shut down the network system | The controller stream has successfully been closed and freed |
| sendControllerState(XInputControlState) | The method invoked by the readySendControllerState signal | | ControllerManager::readySendControllerState(XInputControlState) | The controller stream has been started and the controller has been connected and the system is processing the control input data | The server has been sent the next controller state |
| startControllerStream() | The method invoked when the NetworkManager is initialized | | NetworkManager::initServerConnection() | The stream has not yet been connected and the user has chosen to connect the controller to the server | The controller stream is live and the server is being sent controller state data |
|  |  | |  |  |  |
|  |  | |  |  |  |
|  |  | |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Protected Methods** | | | |
| **Signature** | **Description** | **Preconditions** | **Post Conditions** |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Private Methods** | | | |
| **Signature** | **Description** | **Preconditions** | **Post Conditions** |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Attributes** | | | | | |
| **Name** | **Type** | **Object (Y/N)** | **Instance/Static** | **Visibility** | **Description** |
| \_controllerState | XInputControlState | Y | Instance | Private | The controller state container which holds the current value of every button on an XInput compatible controller |
| \_controllerStream | QTcpSocket | Y | Instance | Private | The api class from the Qt Framework which connects to the server to send the controller data |

|  |  |
| --- | --- |
| **Concurrency** | |
| **Threading Issue** | **Description** |
| Minor | This object will only receive data asynchronously from the ControlManager and the QTcpSocket object handles a majority of the network management |
|  |  |

|  |  |  |
| --- | --- | --- |
| **Major Exceptions** | | |
| **Name** | **Trigger** | **Action** |
| ServerNotAvailable | startControllerStream | Inform the user that the controller stream has failed to initialize and check that the server is available and ready to connect |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Instance Information** | | | |
| **Minimum** | **Maximum** | **Mean** | **Fixed** |
| 1 | 1 | 1 | 1 |
|  |  |  |  |

|  |  |  |
| --- | --- | --- |
| **General Comments** | | |
| **Author** | **Date** | **Comment** |
| Sanchez | 11/17/2014 | Initial draft of ControllerStreamManager class specification |
|  |  |  |

## 6.8 ServerControlManager\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| **Class Information** |
| **Class Name:** ServerControlManager  Abstract Type: No Persistence: No |

|  |  |  |
| --- | --- | --- |
| **Trace-ability Information** | | |
| **Use Case ID** | **Use Case Name** | **Steps** |
|  |  |  |
|  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Public Methods** | | | | |
| **Signature** | **Description** | **Collaborating Class::Methods** | **Preconditions** | **Post Conditions** |
| cleanupServer() | The method invoked when the user signals the system for shutdown | NetworkManager::cleanupNetwork | The server has been connected and running and the user has sent the shutdown signal | The server connection stream has been closed and the resources freed |
| initializeServer() | This method is invoked when the NetworkManager is initializing the networking module, this will create the base connection to the server system | NetworkManager::initServerConnection | The server has not already been connected and is available for access | The server has been connected and sending and receiving data from the client system |
| isConnected() | Safety check which will be available for error detection in the user interface controls | N/A | No preconditions exist | A system module has checked for functionality based on server status |
| startCameras() | Slot method to send the start camera signal to the server, helper to the VideoStreamManager so that it does not need to contain another QTcpSocket | VideoStreamManager::startCameras | The VideoStreamManager has been running and sent the startCameras signal | The camera feed has been started and is being received by the VideostreamManager |
| toggleConnection() | Switches the server connection on and off, it is exposed to the ItemConnectionIndicator in the QML user interface | ItemConnectionIndicator::onPressed | This method has no pre conditions | If the server is connected, it is disconnected.  If the server is disconnected, it is connected.  If the server is available, the method returns false. |
|  |  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Protected Methods** | | | |
| **Signature** | **Description** | **Preconditions** | **Post Conditions** |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Private Methods** | | | |
| **Signature** | **Description** | **Preconditions** | **Post Conditions** |
| disconnectServer() | Helper method to safely cleanup the server connection, ensuring that it is the last connection to the server module first | cleanupServer method has been invoked | The server has been safely disconnected and all network connections are severed |
|  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Attributes** | | | | | |
| **Name** | **Type** | **Object (Y/N)** | **Instance/Static** | **Visibility** | **Description** |
| \_tcpClientSocket | QTcpSocket | Y | Instance | Private | The main server connection which is tied to the lifecycle of the server connection |
| \_connected | Bool | Y | Instance | Private | True/False flag based on the status of the server connection |

|  |  |
| --- | --- |
| **Concurrency** | |
| **Threading Issue** | **Description** |
| Minor | This object merely sends and receives status and status update messages between the client and server system and can be asynchronously invoked by the Qt Framework to allow extensible message options as new features become available. |
|  |  |

|  |  |  |
| --- | --- | --- |
| **Major Exceptions** | | |
| **Name** | **Trigger** | **Action** |
| ServerEndpointUnexpectedlyDisconnected | Any method which attempts to communicate with the server system under the assumption it is connected | Inform the user the server is disconnected, update the connected status, and under some circumstances – attempt to reconnect to the server |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Instance Information** | | | |
| **Minimum** | **Maximum** | **Mean** | **Fixed** |
| 1 | 1 | 1 | 1 |

|  |  |  |
| --- | --- | --- |
| **General Comments** | | |
| **Author** | **Date** | **Comment** |
| Sanchez | 11/17/2014 | Initial draft of the ServerControlManager |

## 6.9 RTSPClient\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| **Class Information** |
| **Class Name:** RTSPClient  Abstract Type: No Persistence: No |

|  |  |  |
| --- | --- | --- |
| **Trace-ability Information** | | |
| **Use Case ID** | **Use Case Name** | **Steps** |
|  |  |  |
|  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Public Methods** | | | | |
| **Signature** | **Description** | **Collaborating Class::Methods** | **Preconditions** | **Post Conditions** |
| createNew(portNumBits, char const\*, int, char const\*, UsageEnvironment&) | This method returns a new instance of an RTSPClient object | VideoStreamManager::startVideoStream | This method has no preconditions, it simply assumes that it has been invoked and the invoker has a plan of usage for the pointer memory which is returned | The invoking object has a new RTSPClient object pointer which the invoker is responsible for cleaning |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Protected Methods** | | | |
| **Signature** | **Description** | **Preconditions** | **Post Conditions** |
| RTSPClient(portNumBits, char const\*, int, char const\*, UsageEnvironment&) | The protected constructor for the RTSPClient class which inherits the RTSPClient object from the Live555 library | No preconditions, this is a constructor | The object has been constructed and initialized according to the implementation of the constructor logic |
| ~RTSPClient() | Custom implemented destructor for the RTSPClient object | The object already existed | The object has been destroyed and the memory appropriately freed |

|  |  |  |  |
| --- | --- | --- | --- |
| **Private Methods** | | | |
| **Signature** | **Description** | **Preconditions** | **Post Conditions** |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Attributes** | | | | | |
| **Name** | **Type** | **Object (Y/N)** | **Instance/Static** | **Visibility** | **Description** |
| \_streamClientState | StreamClientState | Y | Instance | Private | Custom container class to manage the RTSP stream state |
|  |  |  |  |  |  |

|  |  |
| --- | --- |
| **Concurrency** | |
| **Threading Issue** | **Description** |
| Moderate | Not so much of a threading issue as a synchronizing issue, this object will be managing the receipt of video stream data on a UDP extension which will require it to gather and organize as many packets as it can as fast as it can |
|  |  |

|  |  |  |
| --- | --- | --- |
| **Major Exceptions** | | |
| **Name** | **Trigger** | **Action** |
| BrokenPipeException | startVideoStream, startDriving | Inform the user of stream connection error, attempt to reconnect, then inform the user for further investigation if still receiving connection error |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Instance Information** | | | |
| **Minimum** | **Maximum** | **Mean** | **Fixed** |
| 1 | 2 | 2 | 2 |
|  |  |  |  |

|  |  |  |
| --- | --- | --- |
| **General Comments** | | |
| **Author** | **Date** | **Comment** |
| Sanchez | 11/30/2014 | Initial draft of RTSPClient class specification |
|  |  |  |

## 6.10 StreamClientState\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| **Class Information** |
| **Class Name:** StreamClientState  Abstract Type: No Persistence: No |

|  |  |  |
| --- | --- | --- |
| **Trace-ability Information** | | |
| **Use Case ID** | **Use Case Name** | **Steps** |
|  |  |  |
|  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Public Methods** | | | | |
| **Signature** | **Description** | **Collaborating Class::Methods** | **Preconditions** | **Post Conditions** |
| StreamClientState() | Constructor to initialize the StreamClientState object | RTSPClient::RTSPClient | RTSPClient has been created and prepared for use and needs to initialize its internal helper object | StreamClientState has been created and its internal member have been prepared for use by the RTSPClient owner |
| ~StreamClientState() | Destructor | RTSPClient::~RTSPClient | The StreamClientState and RTSPClient have already been initialized and running | RTSPClient and StreamClientState have been destroyed and all memory has been freed |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Protected Methods** | | | |
| **Signature** | **Description** | **Preconditions** | **Post Conditions** |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Private Methods** | | | |
| **Signature** | **Description** | **Preconditions** | **Post Conditions** |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Attributes** | | | | | |
| **Name** | **Type** | **Object (Y/N)** | **Instance/Static** | **Visibility** | **Description** |
| \_duration | double | N | N/A | Public | Current duration of the RTSP stream |
| \_iter | MediaSubsessionIterator | Y | Instance | Public | Library object to manage collections of MediaSubsessions |
| \_session | MediaSession | Y | Instance | Public | Library object to manage a MediaSession |
| \_streamTimerTask | TaskToken | Y | Instance | Public | Library object to help manage the scheduling of the task |
| \_subsession | MediaSubsession | Y | Instance | Public | Library object to manage any MediaSubsessions which exist |

|  |  |
| --- | --- |
| **Concurrency** | |
| **Threading Issue** | **Description** |
| Minor | This object is managed by an object which is already maintaining concurrency solutions |
|  |  |

|  |  |  |
| --- | --- | --- |
| **Major Exceptions** | | |
| **Name** | **Trigger** | **Action** |
| N/A |  |  |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Instance Information** | | | |
| **Minimum** | **Maximum** | **Mean** | **Fixed** |
| 1 | 4 | 2 | 2 |
|  |  |  |  |

|  |  |  |
| --- | --- | --- |
| **General Comments** | | |
| **Author** | **Date** | **Comment** |
| Sanchez | 11/30/2014 | Initial draft of StreamClientState class specification. No major exceptions exist as this object is managed by library code |
|  |  |  |

## 6.11 SteeringWheelController\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| **Class Information** |
| **Class Name:** SteeringWheelController  Abstract Type: No Persistence: No |

|  |  |  |
| --- | --- | --- |
| **Trace-ability Information** | | |
| **Use Case ID** | **Use Case Name** | **Steps** |
|  |  |  |
|  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Public Methods** | | | | |
| **Signature** | **Description** | **Collaborating Class::Methods** | **Preconditions** | **Post Conditions** |
| controllerConnected(unsigned int) | Signal which is emitted when the update loop detects a connected controller | SteeringWheelController::update | SteeringWheelController has been initialized and the polling timer has been started | All slots connected to this signal have been notified of the controller which sent the connected signal |
| controllerDisconnected(unsigned int) | Signal which is emitted when the update loop detects a controller is no longer connected | SteeringWheelController::update | SteeringWheelController has been initialized and the polling timer has been started | All slots connected to this signal have been notified of the controller which sent the disconnect signal |
| newControllerBatteryState(quint8, quint8) | Signal emitted when the controller battery changes state in some way | SteeringWheelController::update | SteeringWheelController has been initialized and the polling timer has been started | All slots connected to this signal have been notified of the controller which sent the battery changed signal |
| newControllerState(XInputControlState) | Signal emitted when the controller state changes in some way | SteeringWheelController::update | SteeringWheelController has been initialized and the polling timer has been started | All slots connected to this signal have been notified of the controller which sent the state change signal |
| setGamepadVibration(float, float) | Method to set the gamepad vibration | N/A | SteeringWheelController has been initialized and the polling timer has been started | The gamepad vibration has been set |
| setLeftStickDeadZone(unsigned int) | Method to set the left stick dead zone | N/A | SteeringWheelController has been initialized and the polling timer has been started | The left stick dead zone has been set |
| setRightStickDeadZone(unsigned int) | Method to set the right stick dead zone | N/A | SteeringWheelController has been initialized and the polling timer has been started | The right stick dead zone has been set |
| setTriggerThreshold(unsigned int) | Method to set the trigger threshold value | N/A | SteeringWheelController has been initialized and the polling timer has been started | The trigger threshold has been set |
| startPolling(unsigned int) | Method used to initiate the controller update cycle which is set to update as the specified time | ControllerManager::controllerInit | Application has been started and the ControllerManager has been created and is initializing the XInput system | The polling timer for the controller has been started and will trigger the update method on every timeout at the interval specified with the parameter |
| SteeringWheelController(unsigned int, unsigned int, unsigned int, unsigned int, QObject \*) | Constructor for the SteeringWheelController XInput object | ControllerManager::controllerInit | ControllerManager has been created | SteeringWheelController has been created |
| stopPolling() | Method to stop polling on the XInput controller | ControllerManager::cleanupController | SteeringWheelController poll timer has been running | SteeringWheelController poll timer has been stopped |
| update() | Main SteeringWheelController method which queries the device for its current state and compares it to the previous state to determine if there is a new signal that needs to be emitted | QTimer::timeout | SteeringWheelController startPolling timer method has been invoked | SteeringWheelController has had the \_currentState member updated with the current device state |

|  |  |  |  |
| --- | --- | --- | --- |
| **Protected Methods** | | | |
| **Signature** | **Description** | **Preconditions** | **Post Conditions** |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Private Methods** | | | |
| **Signature** | **Description** | **Preconditions** | **Post Conditions** |
| processStickDeadZone(qint16, qint16, float&, float&, unsigned int) | XInput Method to set the API dead zone value for the analog sticks | SteeringWheelController has been constructed | SteeringWheelController dead zone values have been set |
| processTriggerThreshold(quint8, float&, unsigned int) | XInput Method to set the API trigger threshold | SteeringWheelController has been constructed | SteeringWheelController threshold values have been set |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Attributes** | | | | | |
| **Name** | **Type** | **Object (Y/N)** | **Instance/Static** | **Visibility** | **Description** |
| \_controllerDesignator | int | N/A | N/A | Private | Integer value of the connected controller descriptor |
| \_currentConnectState | Bool | N/A | N/A | Private | Boolean flag of the current connection status |
| \_currentState | XInputControlState | Y | Instance | Private | Custom container class which holds the state of the connected controller |
| \_leftStickDeadZone | Unsigned int | N/A | N/A | Private | Value assigned to the left stick dead zone |
| \_rightStickDeadZone | Unsigned int | N/A | N/A | Private | Value assigned to the right stick dead zone |
| \_previousConnectState | Bool | N/A | N/A | Private | Boolean flag of the previous connection status |
| \_previousState | XInputControlState | N/A | N/A | Private | Custom container class which holds the state of the connected controller to determine if the newControllerState signal should be emitted |
| \_triggerThreshold | Unsigned int | N/A | N/A | Private | Value assigned to the trigger threshold |
| \_updateTimer | QTimer \* | Y | Instance | Private | Timer which contains the timeout signal to call the update method |

|  |  |
| --- | --- |
| **Concurrency** | |
| **Threading Issue** | **Description** |
| Moderate | This object will need to be run as a separate task or thread to remove the constant polling from the main UI thread, and at that point will be asynchronously signaling any connected slots from a separate thread/task and thus those objects will need to be thread aware |
|  |  |

|  |  |  |
| --- | --- | --- |
| **Major Exceptions** | | |
| **Name** | **Trigger** | **Action** |
| N/A |  |  |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Instance Information** | | | |
| **Minimum** | **Maximum** | **Mean** | **Fixed** |
| 1 | 1 | 1 | 1 |
|  |  |  |  |

|  |  |  |
| --- | --- | --- |
| **General Comments** | | |
| **Author** | **Date** | **Comment** |
| Sanchez | 11/30/2014 | This object generates no major exceptions, it does not affect the behavior if, for instance, the device is suddenly disconnected and is not available for the update method to query. |
|  |  |  |

## 6.12 XInputControlState\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| **Class Information** |
| **Class Name:** XInputControlState  Abstract Type: No Persistence: No |

|  |  |  |
| --- | --- | --- |
| **Trace-ability Information** | | |
| **Use Case ID** | **Use Case Name** | **Steps** |
|  |  |  |
|  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Public Methods** | | | | |
| **Signature** | **Description** | **Collaborating Class::Methods** | **Preconditions** | **Post Conditions** |
| batteryEquals(XInputControlState&, XInputControlState&) | Method which compares and, if applicable, emits the newControllerBatteryState signal | XInputControlState::equals | No assumed preconditions required | Invoking method has a boolean return value for the result of the battery comparison |
| buttonPressed(quint16) | Method which masks a defined button value against the control state object and returns a Boolean value for the pressed condition of the button | ControllerManager::processControllerState | SteeringWheelController object is polling | Invoking method has a boolean return value for the state of a button within the controller state |
| Equals(XInputControlState&, XInputControlState&) | Method which compares two XInputControlStates | SteeringWheelController:update | SteeringWheelController object is polling | Invoking method has a Boolean return value for the comparison of two control states |
| Operator= | Overloaded assignment operator | N/A | N/A | Left hand object is the value of the right hand object |
| XInputControlState(QObject \*) | Default constructor | N/A | N/A | Constructed XInputControlState object |
| XInputControlState(XInputControlState&) | Copy constructor | N/A | N/A | Copy constructed XInputControlState |
|  |  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Protected Methods** | | | |
| **Signature** | **Description** | **Preconditions** | **Post Conditions** |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Private Methods** | | | |
| **Signature** | **Description** | **Preconditions** | **Post Conditions** |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Attributes** | | | | | |
| **Name** | **Type** | **Object (Y/N)** | **Instance/Static** | **Visibility** | **Description** |
| \_batteryLevel | Quint8 | N/A | N/A | Public | Current value of the battery level upon last poll |
| \_batteryState | Quint8 | N/A | N/A | Public | Byte value of current battery state |
| \_buttons | Quint16 | N/A | N/A | Public | Bitfield of the control state buttons |
| \_leftThumbX | Float | N/A | N/A | Public | Current value of the left thumb on the X-Axis |
| \_leftThumbY | Float | N/A | N/A | Public | Current value of the left thumb on the Y-Axis |
| \_leftTrigger | Float | N/A | N/A | Public | Current value of the left trigger |
| \_rightThumbX | Float | N/A | N/A | Public | Current value of the right thumb on the X-Axis |
| \_rightThumbY | Float | N/A | N/A | Public | Current value of the right thumb on the Y-Axis |
| \_rightTrigger | Float | N/A | N/A | Public | Current value of the right trigger |

|  |  |
| --- | --- |
| **Concurrency** | |
| **Threading Issue** | **Description** |
| None | This is simply a data object it is contained |
|  |  |

|  |  |  |
| --- | --- | --- |
| **Major Exceptions** | | |
| **Name** | **Trigger** | **Action** |
| None |  |  |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Instance Information** | | | |
| **Minimum** | **Maximum** | **Mean** | **Fixed** |
| 1 | 10 | 4 | N/A |

|  |  |  |
| --- | --- | --- |
| **General Comments** | | |
| **Author** | **Date** | **Comment** |
| Sanchez | 11/30/2014 | Initial draft of XInputControlState class specification |

## 6.13 OculusState\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| **Class Information** |
| **Class Name:** OculusState  Abstract Type: No Persistence: No |

|  |  |  |
| --- | --- | --- |
| **Trace-ability Information** | | |
| **Use Case ID** | **Use Case Name** | **Steps** |
|  |  |  |
|  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Public Methods** | | | | |
| **Signature** | **Description** | **Collaborating Class::Methods** | **Preconditions** | **Post Conditions** |
| OculusState() | Default Constructor | ClientManager::connectOculus | Oculus was not previously connected | Oculus is now connected and ready for use |
| newOculusState(OculusState) | Signal emitted when the Oculus device has changed state | OculusManager::displayToOculus | Oculus has been connected and running | The new OculusState has been passed to any connected slots and is processed accordingly |
|  |  |  |  |  |
|  |  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Protected Methods** | | | |
| **Signature** | **Description** | **Preconditions** | **Post Conditions** |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Private Methods** | | | |
| **Signature** | **Description** | **Preconditions** | **Post Conditions** |
|  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Attributes** | | | | | |
| **Name** | **Type** | **Object (Y/N)** | **Instance/Static** | **Visibility** | **Description** |
| \_oculusDescription | ovrHmdDesc | Y | Instance | Private | Descriptor object exposed by the Oculus SDK |
| \_oculusEyeRenderDescription | ovrEyeRenderDesc | Y | Instance | Private | Descriptor object exposed by the Oculus SDK, contains the configuration for the Eye Texture requirements |
| \_oculusFrameTiming | ovrFrameTiming | Y | Instance | Private | Frame timing object which contains information on the synchronization state of the Oculus device |
| \_oculusSensorData | ovrSensorData | Y | Instance | Private | Contains the sensor data descriptors of the Oculus device |
| \_oculusTrackingState | ovrTrackingState | Y | Instance | Private | Contains information pertinent to the positional state of the Oculus |

|  |  |
| --- | --- |
| **Concurrency** | |
| **Threading Issue** | **Description** |
| N/A | This is a data object which is used by a manager object |
|  |  |

|  |  |  |
| --- | --- | --- |
| **Major Exceptions** | | |
| **Name** | **Trigger** | **Action** |
| None |  |  |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Instance Information** | | | |
| **Minimum** | **Maximum** | **Mean** | **Fixed** |
| 1 | 3 | 2 | 2 |
|  |  |  |  |

|  |  |  |
| --- | --- | --- |
| **General Comments** | | |
| **Author** | **Date** | **Comment** |
| Sanchez | 11/30/2014 | Initial draft of OculusState class specification |
|  |  |  |

## 6.14 RCManager\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| **Class Information** |
| **Class Name:** RCManager  Abstract Type: No Persistence: No |

|  |  |  |
| --- | --- | --- |
| **Trace-ability Information** | | |
| **Use Case ID** | **Use Case Name** | **Steps** |
|  |  |  |
|  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Public Methods** | | | | |
| **Signature** | **Description** | **Collaborating Class::Methods** | **Preconditions** | **Post Conditions** |
| activateBrake() | Activates the brake function on the ESC device | N/A | The RC system is properly connected | The motor has been reduced to a zero acceleration state |
| deactivateBrake() | Deactivates the brake function on the ESC device | N/A | The RC system is properly connected | The motor is at zero acceleration but ready for a new acceleration vector |
| updateSteeringAngle(int) | Sets the steering angle of the RC servo which controls the steering | N/A | The RC system is properly connected | The servo (steering angle) is set at the intended position |
| updateThrottleValue(bool, double) | Method which controls the throttle value of the motor | N/A | The RC system is properly connected | The motor is at the intended throttle value and set for the direction intended when the method was invoked. |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Protected Methods** | | | |
| **Signature** | **Description** | **Preconditions** | **Post Conditions** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Private Methods** | | | |
| **Signature** | **Description** | **Preconditions** | **Post Conditions** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Attributes** | | | | | |
| **Name** | **Type** | **Object (Y/N)** | **Instance/Static** | **Visibility** | **Description** |
| \_currentSteeringAngle | Int | N/A | N/A | Private | The current steering angle assigned to the steering servo |
| \_isForward | Bool | N/A | N/A | Private | The flag which tracks the desired direction of the motor’s travel |
| \_motorController | MotorController | Y | Instance | Private | The custom object which controls the ESC interface for the main motor driver |
| \_servoController | ServoController | Y | Instance | Private | The custom object which controls the servo interface for the steering system |
| \_throttleLevel | Double | N/A | N/A | Private | The current value of the throttle |

|  |  |
| --- | --- |
| **Concurrency** | |
| **Threading Issue** | **Description** |
| Moderate | This object will not be troubled by the management of its own threading, it will merely need to keep up with the data which it is being sent to process and send to the servo or motor controllers asynchronously |
|  |  |

|  |  |  |
| --- | --- | --- |
| **Major Exceptions** | | |
| **Name** | **Trigger** | **Action** |
| None |  |  |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Instance Information** | | | |
| **Minimum** | **Maximum** | **Mean** | **Fixed** |
| 1 | 1 | 1 | 1 |
|  |  |  |  |

|  |  |  |
| --- | --- | --- |
| **General Comments** | | |
| **Author** | **Date** | **Comment** |
| Sanchez | 11/30/2014 | Initial draft of RCManager class specification |
|  |  |  |