|  |
| --- |
| Martin Rule, Lane Cotgrove, James Bayliss |
| Motion Project |
| Feature 1.1 Connect Kinect client to device |

|  |
| --- |
| Martin Rule  7/30/2012 |

Table of Contents

[1. Overview 2](#_Toc337421407)

[2. Feature team 2](#_Toc337421408)

[3. Sequence diagram 3](#_Toc337421409)

[4. Overallobject model 4](#_Toc337421411)

[5. Method prologues 5](#_Toc337421412)

[6. Testing 6](#_Toc337421413)

[7. Design inspection 7](#_Toc337421414)

[8. References 7](#_Toc337421415)

## 1. Overview

This feature can be accessed us to use all Kinect device features needed for this project. This feature can be accessed both upon start up and when selected via the GUI. The feature will initialise the Kinectdevice providing the necessary support to gather information needed from the device.

## 2. Feature team

For the design of this feature we are using the following team members.

Martin Rule – Project Manager, Developer  
Lane Cotgrove – Lead developer  
James Bayliss – Developer, Tester

## 3. Sequence diagram

1. This function calls upon the Kinect API to setup a connection to an attached Kinect device

2. This function is the inbuilt system from the Kinect API used to setup a new device

3.NuiInit() returns an HRESULT indicating the success or failure of Kinectset up process

4. The user is then informed about the result

## 4. Overallobject model



## 5. Method prologues

//---------------------------------------------------  
// Name: NuiInit()  
// Author: Microsoft Corporation  
// Inputs: NULL   
// Outputs: HRESULT  
//   
// Desc: This function searches for an attached Kinect device // then initialises it for use within a program.  
//---------------------------------------------------

## 6. Testing

For this feature we will use unit tests to determine the accuracy of data being produced. Without an end client we are not able to define if the information is correct at this point. Instead we will check to make sure the information is within valid ranges and fully populated.

For this feature we had to use the assert function in two of the .cpp files, Skeletalviewer.cpp and NuiImpl.cpp. To test these files we had to implement the assert.h header file which would allow us to place assert function calls throughout the application.

TEST 1

|  |  |  |  |
| --- | --- | --- | --- |
|  | Pass | Fail | Comments |
| NuiImpl.cpp – Creation of sensory index  (Line 222) |  | 🗶 | This test was designed to fail to ensure the application would not continue to process if the assert fails |
| NuiImpl.cpp – NuiInit() return is successful  (Line 323) |  | 🗶 |
| Skeletal Viewr.cpp – Confirms the Sensor has been initialized correctly  (Line 242) | ✓ |  |

TEST 2

|  |  |  |  |
| --- | --- | --- | --- |
|  | Pass | Fail | Comments |
| NuiImpl.cpp – Creation of sensory index  (Line 222) | ✓ |  | All tests passed which ensures that in the event that a Kinect device is present a connection between the device and application will be successfully created. |
| NuiImpl.cpp – NuiInit() return is successful  (Line 323) | ✓ |  |
| Skeletal Viewr.cpp – Confirms the Sensor has been initialized correctly  (Line 224) | ✓ |  |

## 7. Design inspection

Design inspection was performed by Martin Rule, Lane Cotgrove and James Bayliss on the 30th July 2012.  
  
Advisor inspection was performed by Andrew Eales.

## 8. References

Palmer, S. (2009). *An introduction to Feature Driven Development*. Retrieved from  
 <http://agile.dzone.com/articles/introduction-feature-driven>

Ambler, S. W. (2009).Feature Driven Development. Retrieved from  
 <http://www.agilemodeling.com/essays/fdd.htm>

Dawson, C. W. (2009). *Projects in Computing and Information Systems, A Student’s Guide*. Harlow, England: Pearson Education Limited.

Palmer, S. R. &Felsing, J. M. (2002).*A practical Guide to Feature Driven Development*.Upple Saddle River, NJ: Prentice-Hall.