|  |
| --- |
| Martin Rule, Lane Cotgrove, James Bayliss |
| Motion Project |
| Feature 1.4 push movement data to all connected servers |

|  |
| --- |
| Martin Rule  8/9/2012 |

Table of Contents

[1. Overview 3](#_Toc337424251)

[2. Feature team 3](#_Toc337424252)

[3. Sequence diagram 4](#_Toc337424253)

[4. Overall object model 5](#_Toc337424254)

[5. Method prologues 6](#_Toc337424256)

[6. Testing 7](#_Toc337424257)

[7. Design inspection 8](#_Toc337424258)

[8. References 9](#_Toc337424259)

## 1. Overview

This feature will allows us to send the transformed movement data string to a connected server. It will facilitate the collection of the string from local memory and all network requirements needed.

## 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

## 4. Overall object model

## 

## 5. Method prologues

//---------------------------------------------------  
// Name: getString()  
// Author:   
// Inputs: NULL   
// Outputs: STRING transformedData  
//   
// Desc: This function returns the string transformData from // an MData object. This string contains transformed   
// movement data  
//---------------------------------------------------  
  
//---------------------------------------------------  
// Name: sendString()   
// Author:   
// Inputs: STRING string   
// Outputs: NULL  
//   
// Desc: This function sends a string to a previously   
// connected server  
//---------------------------------------------------  
  
//---------------------------------------------------  
// Name: sendMData()   
// Author:   
// Inputs: MData pendingMData   
// Outputs: NULL  
//   
// Desc: This function sends the relevent string from an   
// MData object to a previously connected server  
//---------------------------------------------------

## 6. Testing

For this feature we will perform testing using a local sever. Checking that not only I the data received but the data is also accurate. We will also perform testing on the loop that runs the main method to make sure no errors come up.

For this feature we had to implement some asserts into the networkModel.cpp file. This is where the Kinect Client calls the methods used to send data over the server. As there are two ways that the client can connect to the server there will be two ways that the client will send data across the server. This means we will have to test both forms of sending data, via the socket server and via the webserver.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Pass | Fail | Comments |
| networkModel.cpp – Testing to confirm str is not null ( line 132) | ✓ |  | These tests were designed to ensure that the data being sent to the server is correct. The tests to the socket server remain unknown as we have decided that a socket server is irrelevant for this project and will be removed. |
| networkModel.cpp – Testing to confirm the data has been sent via socket Server  (Line 139) | ✓ |  |
| networkModel.cpp –  Testing to confirm data has been sent to the web server.  (Line 411) | ✓ |  |

## 7. Design inspection

Design inspection was performed by Martin Rule and Lane Cotgrove and James Bayliss.  
  
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.