|  |
| --- |
| Martin Rule, Lane Cotgrove, James Bayliss |
| Motion Project |
| Feature 2.2 Pull Movement Data from Server |

|  |
| --- |
| Lane Cotgrove & Martin Rule  9/3/2012 |

Table of Contents

[1. Overview 2](#_Toc337411932)

[2. Feature team 2](#_Toc337411933)

[3. Sequence diagram 3](#_Toc337411934)

[4. Refined object model 4](#_Toc337411935)

[5. Class and method prologues 5](#_Toc337411936)

[5.1 Method prologues 5](#_Toc337411937)

[6. Testing 6](#_Toc337411938)

[7. Design inspection 7](#_Toc337411939)

[8. References 8](#_Toc337411940)

## 1. Overview

This feature will be used to pull data from a connected Server. After the application creates a successful connection with a Server (Feature 2.1) it will be able to request the next item from the Server’s queue. This item is then sent back to the client and stored in local memory.



## 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. At a regular interval, the AvateeringXNA object will Update, it will send a notification to the NetworkModel.

2. The NetworkModel object will then contact the registered server and request the next section of movement data from the queue.

3. The next section of movement data from the queue is then returned to the NetworkModel object.

4. The NetworkModel object then stores this new movement data segment into a local string.

## 4. Refined object model



## 5. Class and method prologues

## 5.1 Method prologues

//---------------------------------------------------

// @Name: getDataFromNetwork

// @Author: Lane - PeePeeSpeed

// @Inputs: NULL

// @Outputs: NULL

//

// @Desc: Attempts to pull the latest

// string of data from the server.

//---------------------------------------------------

//---------------------------------------------------

// @Name: getTransformedData

// @Author: Lane - PeePeeSpeed

// @Inputs: NULL

// @Outputs: string transformedData

//

// @Desc: This GET method returns the string

// transformedData.

//---------------------------------------------------

//---------------------------------------------------

// @Name: readFromFile

// @Author: Lane - PeePeeSpeed

// @Inputs: NULL

// @Outputs: NULL

//

// @Desc: Reads a string of data from

// the AvateeringData.txt file.

// Used for testing only.

//---------------------------------------------------

//---------------------------------------------------

// @Name: writeToFile

// @Author: Lane - PeePeeSpeed

// @Inputs: NULL

// @Outputs: NULL

//

// @Desc: Writes a string of data to the

// AvateeringData.txt file. Used

// for testing only.

//---------------------------------------------------

## 6. Testing

For this feature, we will test the connection by attempting to pull the data from the server, and then comparing it to the data captured and sent from the Kinect Client. A technique we will attempt is to store the Kinect Client data into a file before it is sent to the server then, once the data has reached the Avateering Client, we will store the data into another file. We will then write a short application that will compare the 2 files to see if the data matches.

## 7. Design inspection

Design inspection was performed by Martin Rule, Lane Cotgrove and James Bayliss.  
  
Advisor inspection was performed by Andrew Eales on the 27th September 2012.

## 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.