|  |
| --- |
| Martin Rule, Lane Cotgrove and James Bayliss |
| Motion Project |
| Overall domain model v1.2 |

|  |
| --- |
| Martin Rule  7/18/2012 |

Contents

[Overall Model Development 2](#_Toc331155948)

[1. Form modelling team 3](#_Toc331155949)

[2. Domain walk-through 4](#_Toc331155950)

[2.1 High level system description 4](#_Toc331155951)

[2.2 Domain #1 Kinect Client 4](#_Toc331155952)

[2.4 Domain #3 Servers 4](#_Toc331155953)

[3. Study documents 5](#_Toc331155954)

[4. Develop the model 6](#_Toc331155955)

[4.1 Domain #1 Kinect Client 7](#_Toc331155956)

[4.2 Domain #2 3D Clients 8](#_Toc331155957)

[4.3 Domain #3 Server 9](#_Toc331155958)

[4.4 Data Flow Diagram 10](#_Toc331155959)

[5. Refine overall object model 11](#_Toc331155960)

[6. Model notes 12](#_Toc331155961)

[6.1 Skeletal client 12](#_Toc331155962)

[6.2 Avateering client 12](#_Toc331155963)

[7. Internal and external assessment 13](#_Toc331155964)

[8. References 14](#_Toc331155965)

## Overall Model Development

These activities on the following pages involve all development team members and are guided by experienced advisors.  
  
This document is a walkthrough of all high level planning for the project. It discusses the scope of the system and the context it is performed in. The project is split into multiple domain areas. These areas are worked on by team members to develop domain models. These domain models are then approved for each area. These are merged into an overall model. The object model included will be subject to change from decisions made at the Design by Feature stage.

## 1. Form modelling team

The team in charge of modelling for this project is  
  
Martin Rule - Project manager  
Lane Cotgrove - Lead developer  
James Bayliss - Developer

## 2. Domain walk-through

This is a high level description of the system and its domains.

## 2.1 High level system description

This projects purpose is to design and build a 3D human movement data capture system. This system will utilise the Microsoft Kinect Hardware and SDK. It will be able to capture and transfer data from a local computer to a remote web server. The data will be available to any client application. We will be using the SDK named Avateering for our 3D display client. This will be split into three main domains.

## 2.2 Subject area #1 Kinect Client

This will be the client that interacts with the Kinect movement capture device. It will pull information from the Kinect device, transform it into a suitable format then push that information to a socket server. It must register itself with the server and attempt to ensure the information it is sending is valid.  
  
2.3 Subject area #2 3D Client  
This will be a client capable of replicating and displaying queue information pulled from a server. It will use the Avateering SDK for its 3D environment. It must register itself with the server and attempt to ensure the information it is sending is valid. We will endeavour to use XNA and the XNA Game Studio to develop this platform.

2.4 Subject area #3 Servers  
This will be a server used locally and remotely that is capable of handling data developed in the Kinect client. A server already exists but may need modification. This sever should be capable of handling both the push and pull of movement information at a reasonable level of performance using a database. This server should also allow both push and pull clients to register themselves.

## 3. Study documents

To develop the overall model described in this document we researched into the existing problem context and environment. We also performed research into other solutions available and software we planned to use in our final working product.

We have received existing code from our client. This code has been worked on previously by students at a work shop comprised of people from industry and tertiary institutes. We will continue to add functionality to this code to interpret and send the data captured between the local client and the remote web server.

## 4. Develop the model

This is where we partition each domain into a number of layers to develop individual models. These domains are split into four main layers.  
- User Interface  
- Problem Domain  
- System Interface  
- Data Management

## 

## 4.1 Subject area #1 Kinect Client

**User Interface requirements**

Indication of connection within server  
Indication of successful interfacing with Kinect Device  
Display of current Kinect output  
Display of Information being pushed to server  
Error display

**Problem Domain requirements**

Transformation of Kinect data to the format that will be suitable for the server.

**System Interface requirements**

Connecting successfully with Kinect device  
Ability to pull relevant information from the Kinect Device  
Connecting successfully with local server  
Connecting successfully to remote server  
Ability to push information to servers  
Ability to register with servers

**Data Management requirements**  
  
Successful storage of local information

**Draft Model**



## 4.2 Subject area #2 3D Clients

**User Interface requirements**

Display of 3D environment and avatar  
Display of movement queue data  
Indication of successful connection to server  
Indication of successful retrieval of captured data  
Error Display

**Problem Domain requirements**

Reconstruction of movement queue data  
Applying movement data to 3D environment and avatar

**System Interface requirements**

Connecting successfully to local server  
Connecting successfully to remote server  
Ability to pull information from sever  
Ability to register with server

**Data Management requirements**

Successful storage of local information

**Draft Model**



## 4.3 Subject area #3 Server

**User Interface requirements**

Indication that server is performing  
Status display

**Problem Domain requirements**

Run successfully as server  
Acceptable levels of performance  
Ability to store and retrieve information in a meaningful way

**System Interface requirements**

Ability to listen for new clients  
Ability to process registration information from clients  
Ability to accept pushed movement data from clients  
Ability to allow connected clients to pull movement data

**Data Management requirements**

Database for queued movement data  
Movement data stored in meaningful way

## 4.4 Problem control Flow Diagram

This is a diagram defining how information will move around the completed system



## 5. Overall object model



## 6. Model notes 6.1 Skeletal client

**Kinect SDK**

The Kinect SDK interacts with the hardware device extracting and processing information. We will be pulling relevant information generated by this SDK.

**SkeletalGUI**

The skeletal GUI is an existing example from the Kinect SDK that displays the information being collected by both the hardware device and the Kinect SDK. We will be using this to display movement information. This object may be expanded to display information about other classes within this domain.

**SkeletalHandler**

This object will handle the data pulled from the Kinect SDK. It will check the data is correct (to a certain standard) then transform the data into a form suitable for sending over a network connection.

**NetworkModel**

This object will handle connecting to both local and remote servers. It will handle both server registration submission and data submission.

### 6.2 Avateering client

**Avateering**

This application takes bone orientation information generated by a KinectSDK and displays the information through an avatar on the screen.

**AvateerHandler**

This object reconstructs movement information received across the network. It also attempts to makes sure the data it receives is consistent.

**3D Environment**

This is the display environment generated by the Avateering application

**NetworkModel**

This object will handle connecting to both local and remote servers. It will handle both server registration submission and data retrieval.

## 7. Internal and external assessment

This document was assessed by Martin Rule, Lane Cotgrove and James Bayliss on the 26th July 2012

This document was assessed by Andrew Eales on 16th August 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.

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