Game-based Motion Capture Training

Author
Robert Gustavsson
*School of Innovation, Design and Engineering*
*Mälardalens University*

Supervisors
*Imagination Studios:*
Christian Sjöström
*Mälardalens University:*
Oguzhan Özcan

Examiner
*Mälardalens University:*
Rikard Lindell
August 20, 2012
Abstract
This report describes a thesis work in computer science that was carried out at Imagination Studios. Imagination Studios is a motion capture studio that also does animations and has its main customers in the game sector. Motion capture is a way to make games and animated videos look realistic by letting real actors perform and capture scenes that are then superimposed on the animated characters in the game or video. Imagination Studios research and development department needed a tool for their future research that has the possibility to compare captured motions in an easy way. So the task for this thesis was to create a tool that makes it possible for a user to see the differences in captured motions and also come up with a creative graphical user interface. During this thesis some game scenarios were made, and then video sketches to describe how an actor should act during these scenarios. Some shoots doing these scenarios to get some test data for the final program. The final product contains of two different programs that make it possible to compare captured motions. The main program shows positions of different markers in three different graphs representing X, Y and Z coordinates over time and has the possibility to save the data down to a csv-file. The second program shows a marker cloud of the actual motions in 3D where the user has the possibility to show one or multiple takes at the same time for easier comparison.
List of Abbreviations

IMS : Imagination Studios
MdH : Mälardalens University
IDE : Integrated Development Environment
DOF : Degrees Of Freedom
ID : Identity
IR : Infrared
LED : Light-Emitting Diode
CCD : Charge-Coupled Device
PSD : Position Sensitive Detector
MoCap : Motion Capture
LMA : Laban Motion Analysis
API : Application Programming Interface
SDK : Software Development Kit
UML : Unified Modeling Language
GUI : Graphical User Interface
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Passive markers</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>Active markers</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>Active marker hand</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>Xbox kinect</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>Color glove</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>Inertial suit</td>
<td>4</td>
</tr>
<tr>
<td>7</td>
<td>Magnetic suit</td>
<td>5</td>
</tr>
<tr>
<td>8</td>
<td>Intro screen to video sketch</td>
<td>9</td>
</tr>
<tr>
<td>9</td>
<td>Environment differences</td>
<td>9</td>
</tr>
<tr>
<td>10</td>
<td>Motion description</td>
<td>9</td>
</tr>
<tr>
<td>11</td>
<td>End of scene screen</td>
<td>10</td>
</tr>
<tr>
<td>12</td>
<td>Last screen of video sketch</td>
<td>10</td>
</tr>
<tr>
<td>13</td>
<td>First GUI drawing</td>
<td>10</td>
</tr>
<tr>
<td>14</td>
<td>First GUI drawing, Animation</td>
<td>10</td>
</tr>
<tr>
<td>15</td>
<td>Second GUI drawing</td>
<td>11</td>
</tr>
<tr>
<td>16</td>
<td>Second GUI drawing, Animation</td>
<td>11</td>
</tr>
<tr>
<td>17</td>
<td>New GUI drawing</td>
<td>11</td>
</tr>
<tr>
<td>18</td>
<td>Final GUI</td>
<td>11</td>
</tr>
<tr>
<td>19</td>
<td>Laban effort graph</td>
<td>6</td>
</tr>
<tr>
<td>20</td>
<td>Graph for comparing emotions in motions</td>
<td>7</td>
</tr>
<tr>
<td>21</td>
<td>High level design</td>
<td>12</td>
</tr>
<tr>
<td>22</td>
<td>Use case diagram</td>
<td>12</td>
</tr>
<tr>
<td>23</td>
<td>L- and T-frame</td>
<td>15</td>
</tr>
<tr>
<td>24</td>
<td>T-pose</td>
<td>15</td>
</tr>
<tr>
<td>25</td>
<td>Hand markers</td>
<td>15</td>
</tr>
<tr>
<td>26</td>
<td>Markers on hand reflection</td>
<td>16</td>
</tr>
<tr>
<td>27</td>
<td>Markers on arm</td>
<td>16</td>
</tr>
<tr>
<td>28</td>
<td>Cameras in front of the actor</td>
<td>17</td>
</tr>
<tr>
<td>29</td>
<td>Cameras behind the actor</td>
<td>17</td>
</tr>
<tr>
<td>30</td>
<td>Props used in shoots</td>
<td>17</td>
</tr>
<tr>
<td>31</td>
<td>Camera used in shoots</td>
<td>17</td>
</tr>
<tr>
<td>32</td>
<td>Start up screen of main program</td>
<td>18</td>
</tr>
<tr>
<td>33</td>
<td>Loaded Takes</td>
<td>19</td>
</tr>
<tr>
<td>34</td>
<td>Two takes selected</td>
<td>19</td>
</tr>
<tr>
<td>35</td>
<td>Three takes and graphs zoomed in</td>
<td>20</td>
</tr>
<tr>
<td>36</td>
<td>Csv file containing marker data</td>
<td>20</td>
</tr>
<tr>
<td>37</td>
<td>Start up screen</td>
<td>21</td>
</tr>
<tr>
<td>38</td>
<td>Producer top selected</td>
<td>21</td>
</tr>
<tr>
<td>39</td>
<td>New take selected</td>
<td>22</td>
</tr>
<tr>
<td>40</td>
<td>Show skin selected</td>
<td>22</td>
</tr>
<tr>
<td>41</td>
<td>Multiple takes at the same time</td>
<td>23</td>
</tr>
<tr>
<td>42</td>
<td>Class diagram</td>
<td>24</td>
</tr>
<tr>
<td>43</td>
<td>Ini file where the external program file path is saved</td>
<td>25</td>
</tr>
<tr>
<td>44</td>
<td>Zoomed in on graph in main program</td>
<td>25</td>
</tr>
</tbody>
</table>
# Table of Contest

Abstract ................................................................................................................................. ii

List of Abbreviations ........................................................................................................... iii

List of Figures ....................................................................................................................... iv

Table of Contest ................................................................................................................... v

1 Introduction ...................................................................................................................... 1

1.1 Background .................................................................................................................. 1

1.2 Problem Definition ..................................................................................................... 1

1.3 Requirements ............................................................................................................. 1

1.4 Target Audience ........................................................................................................ 1

2 Background ..................................................................................................................... 2

2.1 Motion Capture ........................................................................................................... 2

2.1.1 Close-Range Motion Capture .............................................................................. 2

2.2 Environmental Scanning ............................................................................................ 2

2.2.1 Optical Systems .................................................................................................... 2

2.2.2 Non-optical Systems ........................................................................................... 4

2.2.3 Environmental Conclusion ................................................................................ 5

2.3 How to Analyze Motions .......................................................................................... 6

2.3.1 Laban Movement Analysis ................................................................................. 6

2.3.2 Coordinates and Rotations ................................................................................ 7

2.3.3 What to Use .......................................................................................................... 7

3 Design Methodology ...................................................................................................... 8

3.1 Scenarios ................................................................................................................... 8

3.2 Video Sketching ........................................................................................................ 8

3.2.1 Video Sketches ..................................................................................................... 9

3.3 Graphical User Interface .......................................................................................... 10

4 System Design ............................................................................................................... 12

4.1 Program Design ........................................................................................................ 12

4.2 Use Case ................................................................................................................... 12

5 Implementation .............................................................................................................. 14

5.1 What Tools Were Used ........................................................................................... 14

5.1.1 How to Put All Together .................................................................................. 14

5.2 Collecting Test Data ................................................................................................ 15

5.2.2 Shoots ................................................................................................................ 16
1 Introduction

This report describes a thesis work in computer science that was carried out at Imagination Studios. Imagination Studios is a motion capture studio that also does animations and has its main customers in the game sector. Motion capture is a way to make games and animated videos look realistic by letting real actors do what the animated characters should do in the game or video. Imagination Studios research and development department needed a tool for their future research that has the possibility to compare motions in an easy way.

1.1 Background

This thesis is made at the company Imagination Studios (IMS) that is located in Uppsala, Sweden. IMS is a motion capture studio with a shoot space measuring 7x12x5 meters and has the capability to capture 10 actors at the same time. They also have fully equipped sound studio with facial and voice capture for performance capturing, and they have done the cut scenes and in game animations for Battlefield 3, Alan Wake, Bulletstorm and a lot of other big games.

1.2 Problem Definition

This thesis was created because the research and development department at Imagination Studios needed a tool that has the possibility to analyze motions for their future research in motion capturing.

1.3 Requirements

The company had some requirements but most of the requirements were discussed during the project. The basic requirements were:

- The program should be written in C++\(^1\)
- The IDE should be Visual Studio 2010\(^2\)
- The program should be able to handle a common file format for motion capture
- It should provide the capability to easier compare motions
- Simple user interface
- Investigation into creative user interfaces

1.4 Target Audience

This program is addressed to the researches and technicians at IMS who have knowledge about the used files, formats and MoCap technologies.

---
\(^1\) [http://wwwcplusplus.com/info/description/](http://wwwcplusplus.com/info/description/)
2 Background

2.1 Motion Capture
Motion Capture is a way to record motions. The data from the capture can be used to animate characters in films, games, virtual environments etc. to get more realistic motions. MoCap can also be used in sport and medical programs to analyze human gait and motions. Motion capture can be seen in Titanic, Lord of the Ring, Avatar and many more films and it is also used in many games such as Battlefield 3, Grand Theft Auto, Tony Hawk Pro Skater and a lot more (Motion Capture - Who Uses Motion Capture, 1998).

2.1.1 Close-Range Motion Capture
Close-range motion capture also known as performance capture is generally the same as regular motion capture, the difference is that close range is zoomed in on a specific body part such as the face or an arm (Human MoCap, 2000). The face and hands/fingers have more special capturing equipment and for the other body parts you can in general use the same as in regular motion capture. The special equipment can be a camera attached to the actors head and capturing only the motions of the face and for the hand it could be a special glove. But in general the close-range MoCap needs the cameras in closer range because of the markers are attached tighter together.

2.2 Environmental Scanning

2.2.1 Optical Systems
Optical systems use calibrated cameras to triangulate positions of different markers attached to an actor’s body. The number of cameras and markers adjust how realistic the data should be and how much data that has to be cleaned up. The optical systems produce data with 3 DOF for each marker and to get rotation it needs information from at least three relative markers, such as shoulder, wrist and elbow to get the angle of the elbow.

The cameras are often placed around the actor in a circle to get full coverage and this system needs at least two cameras, but you can have as many as desired. The average number of cameras for full body capture is around eight and preferably is positioned in a circle around the actor at different height (Motion Capture Lab Setup, 2002). The cameras can be cameras with 16Mpixel and high resolution (Vicon T-series and T-Series S Edition, 2012) to low-resolution webcams (Budiman R, 2004) and because optical system use calibrated cameras it means that the cameras will have to be calibrated before use.

The optical process:

1. Camera calibration
2. Subject calibration
3. Capture
4. 2D marker identification
5. 3D position reconstruction
6. Skeletal motion reconstruction
Companies that sell optical systems: Vicon\(^3\), MotionAnalysis\(^4\) (passive) and PhaseSpace\(^5\) (active)

2.2.1.1 Passive Markers
Passive markers use retro-reflective materials on the markers to reflect a light made by the camera. These markers are often small rubber balls and this system can handle a lot of markers and high resolution but still have high frames per second (120-160fps). But these systems couldn’t be used outdoors and the actors could not wear any glossy materials.

The cameras used with passive markers are often red light, IR or IR strobes. Because they rely on the reflection to the cameras occlusion can occur (Blake A, 2008). Occlusion is when not enough cameras can see a passive marker to triangulate its position. One type of camera is the CCD.

Because the passive system can handle a lot of markers in good frame rate it is good to use and it is pretty good to use in close-range as well because of the small markers and no cables or heavy batteries has to be attached.

2.2.1.2 Active Markers
Active markers light themselves (instead of reflecting light) to mark their relative position and because of that, active markers could be used in larger spaces. The markers could be LED or IR and require cables or batteries to get electricity. The batteries make’s the markers heavier and bigger and the cables could be irritating for the actor. But if cables are used the markers could be less than two millimeters and that’s smaller then passive.

With active markers couldn’t occlusion occur because each active marker as an ID and the computer know them, because of this the data is cleaner and it is easier to know which actor each marker belong to. Active marker systems often use high-speed cameras and one type is a PSD.

Because active markers can be as small as two millimeters it’s easy to attached them on hands and they are robust against dirt and there will be no merging. One negative thing is that it’s captured on

\(^3\) [http://www.vicon.com/](http://www.vicon.com/)
\(^4\) [http://motionanalysisinc.com/](http://motionanalysisinc.com/)
\(^5\) [http://www.phasespace.com/](http://www.phasespace.com/)
lower frequency than passive markers and each marker needs to have a cable attached or containing a battery.

There is one work that uses active markers for hand capturing but they only use fingertips and wrist as points then they use inverse kinematics to calculate the other positions (A., 2010). This is one thing that should be possible to do. But it is better to have markers on all spots that are going to be captures for more accuracy.

2.2.1.3 Marker-less Systems
There are many ways to track an actor’s motion using a marker-less system, such as using calculation (Sundaressan, 2005) to calculate the positions of the actor, use silhouettes (Rosenhahn B, 2006) for the actor and use color glove (Wang R, 2009). They often base on poses saved in a database to match the movements or silhouettes. A common Markerless system is the Xbox Kinect (Pheatt C, 2012) which uses a RGB-camera and an IR-light as a depth sensor to sense the motions.

None of the marker-less systems is as accurate as system with markers. But the actors don’t have to wear anything and it’s good to use when the accuracy is not the top priority, and this is why marker-less system will not be used in this thesis.

2.2.2 Non-optical Systems
Non-optical systems doesn’t use cameras, they use different “sensors” (these sensors are described below) on the actor to get information about the motions.

2.2.2.1 Inertial System
Inertial system has inertial sensors on the actor that gathers data, then the data is sent to a computer wireless and there the motion is captured. These systems use gyroscopes to record rotation of the actor and support six DOF. The more gyroscopes the system has the more realistic motions, just as optical systems markers.

Inertial system is easy to use, portable, no occlusion and has big capturing areas. But it needs calibration, doesn’t support high frame rate, not as accurate as optical systems
and has the “sliding feet” syndrome (no global position).

The “sliding feet” syndrome happens because inertial system is mostly for capturing rotation and thereby it’s hard for it to give accurate positions for body parts. This is why sometimes an actor could act as a marionette puppy in the software.

This system can also be used in a data glove (Sayeed S, 2010) to capture motions of an actor’s hand. In one work the inertial full-body suit has been used in gait analysis (Cloete T, 2008). In this work the goal is to show that the inertial system can be used for analyzing gait.

Companies that sells inertial systems: Animazoo\(^6\) and Xsens\(^7\)

### 2.2.2.2 Magnetic System

This system uses a magnetic field to measure the motions. This is done by a local transmitter and a set of sensors on the actor. These sensors reports position and rotation to the transmitter source and is then shown in a software. The sensor are very expensive and the transmitting from sensors to transmitter could be done wireless or with wires.

Magnetic systems require less markers than optical systems if inverse kinematics is used, but it has smaller capture volumes. Magnetic systems can get disturbed by other materials and the actor has a lot of cables on him which could infect the performance of the actor. This system was more popular in the past and this system could also be used in a data glove.

There has been some work made with electromagnetic system that tries to capture the motions on the fingers when writing (Mitobe K, 2010) and also to capture a pianist that plays on the piano (Mitobe K K. T., 2006). These works uses a data glove to capture the hand motions and get great output.

Company that sells magnetic system: Ascension\(^8\)

### 2.2.3 Environmental Conclusion

If only the hand would be captured a data glove would be the best choice. But because it will be a whole arm and a hand capture, there are better systems to use such as optical systems with passive or active markers. The first suggestion for IMS was active markers, because there are no marker swapping, high accuracy and easy to use. But IMS had passive markers, so after a discussion with them the usage was passive markers. Not only because IMS have it, also because it is the most accurate system out there.

---

\(^6\) [http://www.animazoo.com/](http://www.animazoo.com/)

\(^7\) [http://www.xsens.com/](http://www.xsens.com/)

\(^8\) [http://www.ascension-tech.com/](http://www.ascension-tech.com/)
2.3 How to Analyze Motions
This part is about how you should describe the motions so you easily could compare them and also show the difference in the program.

2.3.1 Laban Movement Analysis
Laban is one of the most widely used systems of human movement analysis and its describing, visualizing, interpreting and notating the movement. LMA was developed to describe the kinematic and dynamic changes in the structure of a movement, such as directions, changes in relationship to the body or environmental pathways. But non-kinematic changes also, such as power or rhythm. This system is used by many different occupations and has four main categories body, effort, shape and space which all describes different things about the motion.

Body This category describes which body part that’s moving, which body parts that are connected and which parts that are affected by each other.

Effort describes the strength, the time and the control of the movement. Effort could differ a lot depending on the emotions.

Shape This category describes how the shape of the body changes during a movement. The shape could change in three different ways.

- *Shape Flow* responds to internal and external disorders. Internal could be breath or thoughts and external could be environment or sound.
- *Shaping* is the qualitative changes in the shape of the body and there are six qualities defined in LMA, *Rising*, *Sinking*, *Spreading*, *Enclosing*, *Advancing*, and *Retreating*.
- *Directional* is the relationship where the body is pointing towards a part of the environment.

Space describes where in space the movement is done, the path of the motion and the direction of the motion the space has a width, height and a depth.

One work that has been done with LMA is to analyze a reach motion of a patient after a stroke (Foroud A, 2006) and then describe the motion with a universal language, LMA. Another work is about analyzing dance motions using LMA to pick out characteristic motions from the dance (Hachimura K, 2005). After reading these proceedings the feelings that were arisen was that it would be hard to implement a way to describe captured motions with LMA in smooth way.
2.3.2 Coordinates and Rotations
Another way to compare motions is to compare XYZ coordinates of different spots of a motion and then show how the coordinates change in a graph. For example if you are going to lift a cup, you could have a spot at the hand and then capture the positions of the hand through the movement to get a curve in the graph that are comparable.

But for better results and higher accuracy you could add more spots to check locations but also add different angles, such as elbow, fingers, shoulder. These angles could also be compared in a graph.

There is a work made that analyze a motion with different emotions (Amaya K, 1998) that uses graphs to display the differences in the motions (Picture 20). In this work they show how motions are dissimilar with different emotions. Another is to analyze the motions of a patient after a stroke (Kim K, 2011). Here they gave a number of healthy patients and a number of stroke patients some tasks to complete by only moving there right arm and then they compared a healthy patient with a stroke patient using graphs.

2.3.3 What to Use
In some of the scientific articles read (Amaya K, 1998), (Kim K, 2011), (Cloete T, 2008) there are graphs that is used for analyzing captured motions and thereby graphs was also used to describe and make it possible to compare the motions in this program. But the program also uses 3D-models that have the possibility to do the actual motions as an extra thing for the user. IMS thinks this makes it easier for the user to understand the motion because the user can then see it. The user can also see several 3D-models in each other, and in that way see the difference in the motions.
3 Design Methodology

The project was divided into four different parts, and the first part was state-of-the-art where some background research was done and also an environmental scanning of different MoCap systems. The main purpose of this part was to get enough knowledge about different MoCap systems to be confident with the technology at IMS and also to know which MoCap system will be best to use for this program. The second part was “interactive design”, this part was to get a good knowledge how the scenes that should be captured would look like and what kind of motions to compare. This was done by making different scenarios for discussion, and later on does video sketching on the scenarios. This part also contained the initial GUI drawings of the program. Then there was a part for the system design where the main system design was made. The last part was implementation there all ideas and requirements were implemented.

3.1 Scenarios

The scenarios explain an environment that the actor should imagine during the captures, and each scenario has four to five different scenes that explain what the actor should do in the environment. All the scenarios are taken from the imagination and have some connection with the game sector. These were also made to get some ideas what you could capture in a close range environment. The final scenario that was used during this thesis is a mixture between three different scenarios and their scenes.

There were five scenarios made, the first was about a police that was questioning a suspect person. During this questioning the police had multiple motions to (scenes), such as show a picture, move a cup, start a recording machine and write on a paper. The second scenario was about a zombie attack where all grandma’s had turned in to zombies, and a grandson had to do multiple actions to survive. Like pick up a key, open a padlock, pick up a dagger and throw the dagger. The third scenario was about a group of soldiers trapped on a spaceship on an alien planet and they had to do some things like, pull a claw out of ones chest, squeeze a metal cube, type on an alien computer and pick up a fist-weapon. The final scenario was about a crazy scientist that was sitting in a wheelchair in is lab and did some experiments.

The first and second scenarios was almost dropped, one scene of each scenario was saved and the last one was dropped because all motions was based on sitting in a wheelchair. So the final scenario was about a group of soldiers that was trapped in an abandoned spaceship on an alien planet. The scenes were about different happenings that happened on this spaceship, pulling a claw out of a soldier stomach, typing on an alien keyboard, drinking wine from a wine glass and throwing a knife towards an alien. All of these scenes have one motion attach to them for the actor to do.

3.2 Video Sketching

Video sketching is used for different things, it could be used to explain your ideas, make prototypes or communicate with user, clients etc. The video sketching is used by some design schools, design companies and the creative industry, and it has become a very good and easy-to-use tool for everybody. Video sketches could be done in most of the presentation programs out there like Microsoft Power Point, Microsoft Movie Maker, Apple iMovie etc.
Why use video sketching?

- It’s an easy way to get people to understand your idea.
- It’s a communication tool
- It’s very fast to achieve a lot (low production cost)
- Fast material collection
- Let you focus on design, context and usability early
- It discovers issues and holes in the concept

3.2.1 Video Sketches

In this thesis the sketches was done using Microsoft Power Point and are based on the final scenario described in section 3.1. Each scene has its own video sketch and starts with a main screen telling what this is and who made it. Next slide (Picture 10) shows how to distinguish between what the actor need to keep in mind (imagination) and what the actor really need to do (reality). It also shows the scenario environment the actor should imagine and how it would look like in reality.

Picture 10: Intro screen to video sketch

Next slide shows how the scene environment would look like and how the actor is going to imagine it (Picture 9). This is so the acting should be more realistic and make it easier for the actor to act as a soldier. After that slide a number of slides comes that describes in general how the actor should do the motion described in the scenes (Picture 10).

Picture 11: Environment differences

Picture 12: Motion description
When all those slides have passed there are three slides left. Two of them describes that the actor will redo this motion, one time with the claw and then the actor should train the movement ten times then redo it again but without a claw (Picture 13). The final slide describes one difference that was thought to occur (Picture 12). So for example the scene when the actor was going to pull out a claw of a soldier, the thought difference was that there will be a difference where in space the actor grabs the claw.

**3.3 Graphical User Interface**

A part of this thesis was to investigate a creative interface, so because of that the thinking was outside of the box in the beginning. The first idea was just taken from the imagination without trying to copy anyone else GUI. That idea was to have some kind of bubbles in colors representing the buttons (Picture 13, Picture 15). At this state the knowledge about fbx files was low so this GUI had the possibility to load multiple fbx files then the user could choose to either show a selected number of markers in different graphs or as an animation. Depending on which check box the user selected a window would pop showing the information (Picture 14, Picture 16). But after some more fact about the fbx files knowing that it could contain multiple takes and after some discussions with both the supervisors at IMS and MdH this GUI was dropped. Because the users would be skilled in computer programs and are used with professional tools, they don’t need a fancy GUI. They need a GUI that they could easily understand and familiarize themselves with.
That’s why a new GUI had to be made. With more knowledge about the fbx file and how it was structured, it was easier to know what to have in the GUI. At this point it was known that there had to be some kind of list box to show all the takes that a fbx file contains and also a list box to show all the markers that a take contains. There also had to be a load button to load a file and one button to get the markers from the take, and because an external program shows the animations, a button to start that application has to be there. The initial version of the new GUI looked like this (Picture 19).

After some discussions with the IMS supervisors the GUI had some remakes such as adding text that is showing the name of the last selected take and marker, moving the contents of the GUI and adding a save button to save down the data in the graphs to a csv file. At this point the GUI was good enough so an implementation of it could begin, and the final result of the GUI looks like this (Picture 18):
4 System Design

4.1 Program Design
After knowing how to show the motion in a way that makes it possible to compare it, it was time for system design. The first thing that was done was a use-case with a user, so it would be easy to know what the program should contain and also think about what the user should be able to do. After that a high level system design was made to get an idea how to solve the problem and to know what everything should do. Because the fbx sdk has a sample that has the possibility to load fbx files and show it as an animation, it was easiest to rewrite it and make an external program that shows the animations. Thereby there had to be a main program that starts a new process when the user wants to see animations and kill the process when the user doesn’t want to see it anymore. So thereby the main application has the possibility to load an fbx-file, and then show the marker data taken from the file in three different graphs and also save the data in the files into a csv file (Picture 21).

Main Application
Show Graphs
Save Data

External Application
Show Animation

Fbx File

Picture 21: High level design

4.2 Use Case

4.2.3.1 Use case UC1: Save Data

Initiator: User

Goal: Save data from graphs into a file

Main Scenario:
1. The user presses save data button
2. The system checks so there is data in the graphs
3. The system provides a save dialog
4. The user selects where and what name the file should have
3. The system saves the data into the chosen .csv file

Extensions:
2a. there was no data in graphs.
   5a1. the system shows a message box telling the user to have data in the graphs.

4.2.3.2 Use case UC2: Show Graphs

Initiator: User
Goal: Show data of markers in the graphs
Main Scenario:
1. The user has loaded a file
2. The user selected one or more takes
3. The user presses get marker button
4. The system adds all the markers in the take to marker list box
5. The user selects a marker
6. The user presses show in graph button
7. The system shows the marker data in the graphs

Extensions:
1a. a file is not loaded
   1a1. the system shows a message box telling the user to load a file.

4.2.3.3 Use case UC3: Show Animation

Initiator: User
Goal: Show an animation of the take for the user
Main Scenario:
1. The user has loaded a file
2. The user presses show animation button
3. The system starts a new process which can show animations

Extensions:
1a. a file is not loaded
   1a1. the system shows a message box telling the user to load a file.

4.2.3.4 Use case UC4: Load File

Initiator: User
Goal: Load a file into the application
Main Scenario:
1. The user presses the load file button
2. The system provides an open file dialog to choose file
3. The user selects a file
4. The system load the file into the application
5. The system shows all available takes in the file in a list box
5 Implementation

5.1 What Tools Were Used

One requirement is to use the IDE visual studio 2010 and another to use the programming language C++. But there were still a lot of libraries and API’s that was needed to complete this thesis. For managing .fbx files there were not many tools to choose between. The mostly used one that had good documentation and good samples were the Autodesk Fbx SDK. The SDK samples made it easy to understand how it worked and thereby it was pretty easy to work with.

But for the GUI there were a lot of different API’s, and the ones that was checked out during this thesis were Microsoft winAPI, Microsoft Windows Forms, Nokia Qt Visual Studio add-in and OpenFrameworks. WinAPI was dropped pretty fast because the lack of doing good looking GUI’s and was windows only. OpenFrameworks was mostly for touch screens and didn’t have any good default GUI library. There were a lot of them and none of them seemed to fit this project good enough. The two that made the best impression were Windows Forms and Qt. But because Qt are compatible with all operative systems and had style sheets build in their Qt Designer, it seemed to be the best choice for this thesis. For the graphs there were some different plugins for Qt but there was one that was used by almost all users and was the one latest updated Qwt plot.

For the external application the best choice was Autodesk fbx sdk for managing fbx files and OpenGL for drawing 3D-models, and that because the fbx sdk sample had code for showing fbx file content in an OpenGL window and it was easiest to rewrite that sample to fit this thesis. Final Setup is Microsoft Visual Studio 2010 Ultimate, Qt Visual Studio Add-in v.4.8.1, FBX sdk v.2013.1 and Qwt Plot v.6.0.1.

5.1.1 How to Put All Together

The first thing needed is a version of visual studio 2010 that is higher than the express version, and this is so the Qt add-in should work. When visual studio is in place the fbx sdk could be downloaded and installed. Download the Qt Visual Studio Add-in from Qt homepage and compile it with visual studio. When Qt is working with Visual Studio download and compile Qwt Plot and create a Qt project in Visual Studio. Add all the linker dependencies and additional libraries for the Fbx and Qwt plot in the project properties.

9 http://usa.autodesk.com/adsk/servlet/index?siteID=123112&nid=7478532
10 http://msdn.microsoft.com/en-us/library/s2zy4k8w(v=vs.71)
12 http://qt.nokia.com/products
13 http://www.openframeworks.cc/
15 http://qwt.sourceforge.net/
16 http://www.opengl.org/
17 http://download.autodesk.com/us/fbx/20112/FBX_SDK_HELP/index.html?url=WS8e4c2438b09b7f9c50e6e6531197ccd93c5-7ffa.htm,topicNumber=d0e2241
18 http://www.holoborodko.com/pavel/2011/02/01/how-to-compile-qt-4-7-with-visual-studio-2010/
19 http://qwt.sourceforge.net/qwtinstall.html
20 http://download.autodesk.com/us/fbx/20112/FBX_SDK_HELP/index.html?url=WS1a9193826455f5ff-150b16da11960d83164-6bf0.htm,topicNumber=d0e1518
5.2 Collecting Test Data

To get files to work with during implementation some shoots had to be made at IMS. A shoot is when you capture an actor motions in the capture volume, and during this thesis the student was the actor and the supervisors at IMS were the motion capture team.

To set up a motion capture volume there were several steps. First all the sunlight had to be eliminated. Then the camera was calibrated, and this was done in multiple steps. First start with laying an L-frame with four markers in the middle of the volume, this is for the XYZ-coordinates of the volume and also the positions of the cameras. After the L-frame a person walks around in the volume with a T-frame with three markers to capture data all over the volume, this is for better camera position adjustment (Picture 23). The cameras have been roughly calibrated after this step. So now the floor has to be adjusted, this is because the floor in the studio is not completely horizontal. This adjustment is done in the software. The camera calibration is same for both close range motion capture and regular motion capture, the only difference is that the volume is often smaller in the close range so there has to be a smaller T-frame when collecting data.

The actor now puts on a motion capture suit which doesn’t reflect any light at all and get markers attached where it is needed. If there will be a regular motion capture shoot the actor walks around in the volume for some final cameras calibration, and after that does some specified motions for a template representing the actor in the software. The motions are:

- T-pose
- Clap
- Crunch
- Stretch
- Lift legs forward
- Rotate legs backwards
- Move fingers and thumb
- Shake legs
- Reach ground with fingertips
- Stands on right/left knee

But if there will be a close range shoot a person has to walk around the volume again with the T-frame and after that, the actor does some close range specified motions for the template. For the arm and hand the motions are:

- Pan hand right/left
- Rotate wrist
- Move fingers and thumb
- Move arm
When the actor has done his template-moves the template in the software can be made. This takes some time because sometimes markers couldn’t be captured properly or markers can swap with each other. This has to be fixed manually in the software, but when this is done the real shooting can start.

During the shoot you can watch a marker cloud moving around in the software doing the same motions as the actor doing in the capture volume. When the shooting is all done, the marker has to be fixed again from marker swapping and disappearance. When this is fixed the file is “clean” and can be saved as a trc file. Everything is now finished in the first software and the saved file is loaded in to motion builder where all after work is done. Here the marker cloud is transformed in to a real body and a real character is made. When this is finished the character can be saved as fbx-file and that’s the file format used in this thesis. That format is used because that is the format that IMS work with and it’s a common format.

5.2.2 Shoots

In this thesis there was one regular motion capture shoot and one close range capture shoot. The regular was for getting an early test file to work with and the close range was the final which would test the application.

5.2.2.1 Regular Shoot

For the regular motion there was just one scenario and one scene, which was the claw pullout. Here the actor had a full body suit with markers all over his body and stood in the middle of the motion capture volume. In the first shoot the actor imagine that there was a claw on the table in front of him, then the actor had a claw-like prop in front of him and in the final shoot the actor had trained with the claw a number of times and then imagine the claw in front of him again.

5.2.2.2 Close Range Shoot

For the close range capture there was one scenario and four scenes, claw pullout, typing an alien keyboard, throw a knife and moving a wine glass. All of these are focused on the right arm and right hand motions. So the actor only had a jacket with markers attach to his right arm and markers glued on his hand and fingers (Picture 27). There were three markers on the shoulder, two on the elbow, three on the wrist and four markers on each finger (Picture 26).

![Picture 26: Markers on hand reflection](image)
![Picture 27: Markers on arm](image)
But for the close range the cameras has to be closer to the actor because the markers are tighter together. So because of this a discussion about taking down three cameras and put them on tripods in front of the actor was made, but supervisor Christian thought that you could use the cameras on the walls (Picture 28), just move the actor closer to a wall. Christian’s idea worked, so because of that there were five close-range cameras in front of the actor instead of three and the rest of the cameras in the room were for global positioning of the arm markers (Picture 29).

In all the shoots the actor started with the right hand on the table, then did the motion according to the video sketch, and when the shoot was finished the actor put the right hand back on the table. The actor also had to have his feet on the same spots all the time. These things had to be done so the actor always starts on the same position and ends on the same position, so the data would be easier to compare in the program later. The props that were used during the shoots (Picture 30) are

1. The start/end position of the right hand
2. The wine glass to drink from
3. The claw to pull out of a body
4. The knife to throw toward an alien
5.3 Graphical User Interfaces

The graphical user interface for the main application is drawn using GIMP 2.21 for the pictures and icons and the program Qt Designer to combine the graphics with controls. The external application menu is made using the built in menu system in OpenGL.

5.3.1 Main Application (MoCap data comparison)

When the application starts everything is blank (Picture 32), and the only thing the user can do is to load a file. If the user tries to do something else a message box will pop and say “please load a file”. When the user presses the load file icon an open file dialog pop and lets the user navigate through his computer to choose the wanted fbx file.

When the file has been loaded the left list box will be filled with all the takes that are saved in the fbx file (Picture 33). If the user loads a file that contains more than six takes, a message box will pop and say that this application only supports six takes and the rest will be drawn in black. Now the user can select one or multiple takes and press “Get markers” to get the markers used in the takes or the user can press “Show animation” to start the external application that shows the takes as animations. If the user has selected multiple takes and presses “Get markers” the icon above “Show in graphs” will change to multiple curves instead of one. This is to get a better feeling in the application.

21 http://www.gimp.org/
As shown in the picture below (Picture 33) the user has selected two takes, and selected to show the data of the marker TRC:RRHand. The system now automatically adjusts the axes of the graphs to fit the curve as good as possible and now the user can see how the data are different in the takes and compare them. Within the graphs the user has the possibility to either zoom in by holding the left mouse button and drag a rectangle over the wanted area (Picture 44), or press right mouse button to zoom out one step. If the user presses ctrl and right mouse button the graph will zoom out as much as it can.

The user can also pan in the graphs by holding the scroll wheel down and move the mouse. Picture 35 shows when three takes are compared and the user has zoomed in on each graph. Between the reset graphs button and the show animation button the user can see which the last take and marker that was compared. The reset graphs button clears all the graphs.
If the user presses the save graph data button a save file dialog will pop and let the user select a name and where the file should be located. The file format is csv so the data could easily be read (Picture 36) and used in other programs. The file consists of the marker name that have been compared, all the takes (animation layers), the time and all the coordinates at the specific time.

<table>
<thead>
<tr>
<th>Marker name:</th>
<th>time</th>
<th>x-position</th>
<th>y-position</th>
<th>z-position</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grab_Training_Claw_1_Clean-HTR_RG</td>
<td>0</td>
<td>52.895</td>
<td>77.8677</td>
<td>-8.2978</td>
</tr>
<tr>
<td>Grab_Training_Claw_1_Clean-HTR_RG</td>
<td>0.0333333</td>
<td>51.8471</td>
<td>77.8235</td>
<td>-9.0308</td>
</tr>
<tr>
<td>Grab_Training_Claw_1_Clean-HTR_RG</td>
<td>0.0666667</td>
<td>50.9437</td>
<td>78.1859</td>
<td>-9.5697</td>
</tr>
<tr>
<td>Grab_Training_Claw_1_Clean-HTR_RG</td>
<td>0.1</td>
<td>50.3137</td>
<td>78.9331</td>
<td>-10.9646</td>
</tr>
<tr>
<td>Grab_Training_Claw_1_Clean-HTR_RG</td>
<td>0.1333333</td>
<td>50.0075</td>
<td>80.0892</td>
<td>-11.6703</td>
</tr>
<tr>
<td>Grab_Training_Claw_1_Clean-HTR_RG</td>
<td>0.166667</td>
<td>49.9764</td>
<td>81.3905</td>
<td>-12.1931</td>
</tr>
<tr>
<td>Grab_Training_Claw_1_Clean-HTR_RG</td>
<td>0.2</td>
<td>49.9622</td>
<td>82.7764</td>
<td>-12.6087</td>
</tr>
</tbody>
</table>

### 5.3.2 External Program (FBXviewer)

The FBXviewer starts with a 640x480 window which contains a grid and a marker cloud that depicting a human being (Picture 37). The window size could easily change by take the mouse to a corner and press left mouse button and move the mouse. You can then freely change the camera position by holding the left mouse button down and move the mouse, and you can also zoom by holding the scroll wheel down and moving the mouse forward to zoom in and backwards to zoom out. If the user presses the right button a menu pops (Picture 38). Then the user gets multiple options to choose between:
**Picture 37: Start up screen**

**Select Camera:** here the user can select from what angle that suits the best to look at the model or select perspective to move around the camera freely by holding left mouse button (Picture 38).

**Picture 38: Producer top selected**

**Select Take:** here the user chose which take the model should play, depending on the color given to the take in the main application the marker cloud get the same. If the user chose another take, it will start to play automatically (Picture 39).
Select Shading Mode: the user can switch between Wireframe and Shading mode to change the appearance of the model. This is only possible to do if the user has chosen “Show Skin” under the select view mode sub menu.

Select Zoom Mode: here the user can select how the zooming should work.

Select View Mode: Here the user can select to show a marker cloud (default) or show skin mode. The show skin mode only works if the model has a texture on it in the fbx file (Picture 40).
**Select Number of Takes:** Here the user can choose to play all the takes at the same time, this makes it possible for the user to see the differences between the takes using marker clouds instead of the graphs (Picture 41).

![Select Number of Takes](image)

*Picture 41: Multiple takes at the same time*

**Play:** Play/Pause an animation, if the user selects to pause the program the user can use the keys A and S to move frame by frame towards or backwards.

**Exit:** If exit, the program close and takes the user back to the main program.

The things added to this program during this thesis where the different colors of the marker cloud depending on which take selected, the possibility to step forward and backward frame by frame when the program has been paused and the biggest change is the possibility to play multiple takes at the same time which makes it possible for the user to easy see the differences in the captures motions.
5.4 Classes

The classes described in this section are for the main program. The external program will not be written about because the changes done in that samples are just method rewriting and some methods added to classes. The main structure of that sample is still the same.

5.4.1 Class Diagram

![Class Diagram](Picture 42: Class diagram)

5.4.2 MainWindow

The MainWindow class handles all the user input and output, and it doesn’t contain any other logics. The MainWindow has a Ui_GbMCT that contains all the controls of the GUI it also inherits from QMainWindow so it is the startup form. It has an FbxHandler to be able to handle a fbx-file and an Animation to be able to start the external application. The IO is used to save and load all other files and QwtPlotCurve, QwtPlotPanner and Zoomr are used to be able to navigate in and use graphs.

5.4.3 FbxHandler

FbxHandler is the class that handles everything that has to do with the fbx file. This is one of the biggest classes and most of the logic is done here. It has multiple methods for handling the file, but the most important is the load method. It also contains arrays of the selected markers coordinates.
5.4.4 Animation
This class just contains one method, and this method starts the external program. The method takes the file path to the fbx-file and the file path to the external programs exe file and with that it can start a new process that has the possibility to show the animations for the specific fbx file.

5.4.5 IO
This class handles all files that are not an fbx files. It saves the data in the graphs down to a csv file and it loads an ini file which contains the file path to the external program (Picture 43). This path is saved in a file so the program easily can be moved to different computers and the user can easy change it to the new location.

5.4.6 Zoom
This class inherits from a class called QwtPlotZoomer which make it possible to zoom in and out on the graphs. This class is made so the buttons that is used for the zoom function in the graphs are set and some zoom settings are made. To zoom in the user holds the left mouse button to create a rectangle to zoom in on and right mouse button to zoom out.
5.5 Issues to Overcome
The biggest issue to overcome was to get all the tools used to work with visual studio, because it took a lot of hours to get the Qt add-in to work with visual studio and then later on get the fbx sdk and qwt plot to work. The problems were linker errors when compiling and Qt settings, but it was solved after many hours at the official forum for Qt22 and a lot of hours in the visual studio project properties to add the additional includes and directories. Another issue to overcome was to understand how the FBX-file was structured and how to get the marker data out of it, this was also solved with a number of hours in forums and mostly the official forum for fbx sdk23 and was final solved by make an animation curve from the marker data and take the data from the animation curve. The last big issue to overcome was to understand how the Qt worked with connecting a control to an event and vice versa, but this was easily solved after a number of samples read.

22 http://qt-project.org/forums/
23 http://area.autodesk.com/forum/autodesk-fbx/fbx-sdk/
6 Conclusion

6.1 Final product
The final product is two programs which make it possible for Imagination Studios to compare motions recorded using motion capture equipment. The programs handle .fbx files which are a common file format in motion capture. The user of the program can compare the motions by selecting a marker and compare the position of the marker for the different motions in a graph. The user can also see the motions in a 3D-environment where a cloud of markers is moving, there the user also can select to see all the motion at the same time and in that way also see the differences between the motions. If the fbx file contains a skin for the marker cloud, the user also can see the marker cloud with skin on.

After a brief literature review through different article databases such as IEEE, ACM and Google scholar with the search words:

- “MoCap / Motion Capture Comparison”
- “MoCap / Motion Capture Analysis”
- “MoCap / Motion Capture Program”
- “MoCap / Motion Capture Motion Analysis Program”

There seems to be no program that has the possibility to both, show graphs with marker data and show the actual animation. There are articles that involve graphs (mentioned in sec. 4.1.3), but the name of the program is not mentioned and they don’t mention anything about 3D-models for comparison. If there is no program that uses both graphs and 3D-models to compare captured motions, my program is unique.

There is one way found to do this program in another way and it is to write a python script to Motion Builder to see the graphs and another script to save down the data to a csv file. Why this program isn’t done in that way is because then every user that wants to compare fbx files needs to have Motion Builder (which is not cheap) and the knowledge how to work with it. Furthermore, IMS needed an application that can compare motions in data and visual approach at the same time. Moreover, it is possible to extend the application with more features.

6.3 Future work
In the future you could do the external program better by add the names of the takes that are shown in the window and add it to the main application so the whole system contains of just one process.

6.2 My Reflections
I am very satisfied with my work and think I have managed to do a lot more than I thought I was able to do, because I haven’t work so much with C++ and OpenGL and never work with FBX files and QT. So this thesis gave me a lot of new experiences and I think it has been really fun to work with Imagination Studios, not only because the thesis went well, but also because I really think they do cool stuff and the whole environment was new to me.
Bibliography


Acknowledgements

First I would like to thank Imagination Studios for the possibility to do a thesis for them and I would also like to thank my family for their support during my three years at the university.

At last I would like to give a special thanks to my girlfriend and our child Lo for their support and love. I love you.

/ Robert G