



NTNU – Trondheim
Norwegian University of
Science and Technology

3D modelling using Leap Motion

Focusing on homogeneous transforms

John Edvard Reiten

Master of Science in Computer Science

Submission date: July 2014

Supervisor: Rune Sætre, IDI

Co-supervisor: Igarashi Takeo, University of Tokyo

Norwegian University of Science and Technology
Department of Computer and Information Science

Abstract

Leap Motion is a novel approach to human-machine interaction, which allows the computer to be controlled by hand and finger movement without the need of other equipment. It can track more than ten fingers with an accuracy of 0.01 mm, and introduces a new market for motion tracking applications and games available for personal computers. Creating a three-dimensional (3D) modeling application using Leap Motion as the input device may make it easy for novice users to move and rotate objects in 3D virtual space. During the project, a prototype was developed to try out different strategies for moving and rotating 3D objects. The objective was not to create a fully functional 3D modeling application, but to create strategies for manipulating 3D objects using Leap Motion as the input interface. Four different strategies were implemented. A literature survey was done in advance of implementing the application in order to investigate what have been done previously, and take inspiration from similar applications to create the strategies in this project. A user test and a survey with 12 users was conducted to figure out if it is easy to use Leap Motion to accurately move and rotate 3D objects with the proposed strategies. The research concludes that it is difficult to find a strategy well suited for accurately rotating 3D objects using Leap Motion. Moving objects however is easy, but combining the actions (moving and rotating) make it more difficult to use for beginners. Users often experience fast fatigue, making the proposed strategies unsuitable for working for longer periods of time. Even though the controls were difficult to use, users found the controls easy to understand.

Summary

In this project, an implementation of an interactive application that focus on how to do homogeneous transformations using Leap Motion as the interaction interface have been made. A literature study was conducted in the start of the project to get an idea of what had already been done, and to get ideas on how to implement the software in this project. An evaluation process guided by the DECIDE framework and the GQM model was used in order to evaluate the project. After implementing a prototype, a pilot test was followed by an extensive user test after improving the prototype. The results from the user test have been quantified, and it was discovered that none of the techniques to rotate 3D models are suitable for working with for longer periods of time. It also takes too long time to complete a rotation with the techniques implemented in this project compared to a how fast a user is able to rotate objects using a computer mouse. Even though the suggested techniques for rotating and translating objects did not prove to be easy, it is still worth noting that the techniques might be more useful for making freeform 3D models with a more creative touch where precision is not important. Leap Motion is still being developed, and motion tracking is not yet applicable for replacing mouse and keyboard for accurate 3D transforms. This research might give insight to others who want to explore tracking devices and 3D modeling, and understand how tracking devices may be used more easily for users.

Sammendrag

En implementasjon av en interaktiv applikasjon som fokuserer på homogene transformasjoner ved bruk av Leap Motion som interaksjonsmedie har blitt laget i dette prosjektet. Et litteraturstudium ble gjennomført i starten av prosjektet for å få en oversikt over hva som allerede har blitt funnet ut og lagd, samt skaffe idéer til hvordan man kan implementere programvaren i dette prosjektet. En evalueringsprosess ved hjelp av DECIDE rammeverket og GQM modellen ble brukt for å evaluere prosjektet. En pilottest ble gjennomført for å teste en prototype av applikasjonen. Etter å ha forbedret prototypen ble en omfattende brukbarhetstest gjennomført. Resultatene fra brukbarhetstesten ble kvantifisert, og basert på resultatene ble det oppdaget at ingen av teknikkene for å rotere 3D modeller kan taes i bruk for å jobbe over lengre tid. Det tar for lang tid å gjennomføre en rotasjon ved hjelp av de teknikkene som ble implementert i dette prosjektet i forhold til hvor hurtig brukere klarer å rotere objekter ved hjelp av en datamus. Selv om teknikkene for å rotere og flytte objekter implementert i dette prosjektet ikke var enkle å bruke, så er det fortsatt viktig å merke seg at teknikkene kan bli brukt for å gjøre andre oppgaver som ikke krever god presisjon, for eksempel å lage kreative friform 3D modeller hvor presisjon ikke er en viktig del av modelleringen. Leap Motion er fortsatt under utvikling, og bevegelsessporings kan ikke erstatte mus og tastatur for å presist kunne utføre homogene transformasjoner. Forskningen i dette prosjektet kan gi innsikt til andre som ønsker å utforske bevegelsessporingsenheter og 3D modellering, og kan gjøre det mulig for brukere å enkelt bruke bevegelsessporingsenheter.

Preface

This is the master thesis for the graduation at NTNU by John Edvard Reiten. A Master thesis at NTNU is part of the study program: computer science. The project was carried out in the spring 2014, at NTNU with the cooperation of the University of Tokyo.

The idea of the project arose after having an autumn project in 2013 concerning first person controller navigation using Leap Motion. As the Japanese professor who supervised the student has a passion for 3D modeling, it was a good opportunity to see how well Leap Motion could be used to create 3D models, considering it is a new technology.

The main aspects of this report is how to decide upon a problem to solve, how to solve it, and how the problem actually was solved following a discussion about what went well and what could have been done better. The reader will get an understanding about what 3D modeling is, what motion tracking devices are, and the process in which software engineers take an idea from the sketch board and create an application that users can interact with on their computer.

I learned that many people have different ways of using a system with a motion controller. Some are able to quickly grasp how to use Leap Motion and understand what limitations it have, while some gets frustrated because Leap Motion does not respond according to their intentions.

I would like to thank the following persons for their great help during the project. I want to thank Professor Takeo Igarashi from the University of Tokyo(UTokyo), and Associate Professor Rune Sætre from the Norwegian University of Science and Technology (NTNU) for their help and guidance through the development of the application, as well as helping me to answer all questions that I had concerning my research. I would also like to thank the members of Professor Takeo Igarashi's research lab for helping me having an enjoyable life in Tokyo, and giving constructive feedback on the application I made. I would like to thank Julia Zazhingina from the international office at NTNU, and Kaori SATO from the international office at UTokyo for helping me being an exchange student at UTokyo.

Trondheim, 2012-12-16

(sign)

John Edvard Reiten

Problem description

In this project, an interactive application for doing homogeneous transforms on digital 3D models using the optic hand and finger tracking device Leap Motion will be developed. An investigation about what has been done previously in the field of 3D modeling, and find relevant experiences with using motion tracking technology for creating 3D models should be conducted. The project focuses on experimenting with different techniques and approaches to how a user would interact with models in 3D space with a special focus on homogeneous transforms.

Assignment given : 2014 January 14

Supervisors: Rune Sætre and Takeo Igarashi

Abbreviations

GQM - Goal Question Metric

DECIDE - Determine the goals, explore the questions, choose the evaluation paradigm and techniques, identify the practical issues, decide how to deal with the ethical issues, evaluate, interpret, and present the data

SWOT - Strengths, Weaknesses, Opportunities, Threats

3D - 3Dimensional

2D - 2Dimensional

2.5D - 2.5Dimensional

E3 - Electronic Entertainment Expo

USB - Universal Serial Bus

USD - United States Dollar(s)

SDK - Software Development Kit

PC - Personal Computer

A/C - Alternate Current

HMD - Head Mounted Display

VR - Virtual Reality

URL - Uniformed Resource Locator

CAD - Computer-Aided Design

Contents

Abstract	i
Summary	ii
Sammendrag	iii
Preface	iv
Problem description	v
Abbreviations	vi
1 Introduction	2
1.1 Background and goals	3
1.2 Limitations	5
1.3 Approach	6
1.4 Structure of the report	6
2 Methodology	8
2.1 Research questions and goals	8
2.1.1 The goal question metric	8
2.1.2 Main objectives	11
2.2 Evaluation framework	12
2.2.1 Determine goals	13
2.2.2 Explore questions	13
2.2.3 Evaluation paradigm	13
2.2.4 Practical issues	13
2.2.5 Ethical issues	14
2.2.6 Evaluate, interpret, and present data	14
2.3 Evaluation process	15

2.3.1	Literature study	15
2.3.2	Software implementation	15
2.3.3	Evaluation paradigm	16
2.3.4	Practical issues	17
2.3.5	Ethical issues	17
2.3.6	Questionnaires and interviews	17
2.3.7	Interpreting data	18
3	State of the art	20
3.1	Literature study	21
3.1.1	Sketch-based	21
3.1.2	Gesture-based	23
3.1.3	Motion controlled	31
3.2	Similar applications	36
3.2.1	Freeform	37
3.2.2	Leopoly	37
3.2.3	Hot Pot Factory	38
3.2.4	Autodesk Maya	39
3.2.5	Leapouts	40
3.3	Technologies	40
3.3.1	Leap Motion	40
3.3.2	Microsoft	45
3.3.3	Nintendo	45
3.3.4	Sony	46
3.3.5	Sixense	46
4	Development process	49
4.1	Planning and work process	49
4.1.1	development models	50
4.1.2	Trello	52
4.1.3	Tom's planner	53

4.1.4	Risk Management	54
4.2	Architecture	56
4.2.1	Development view	57
4.2.2	Physical view	58
4.2.3	Logical view	59
4.2.4	Process view	62
4.2.5	Scenario view	66
4.2.6	Rotation strategies	69
5	User study	75
5.1	Pilot test	76
5.1.1	Tests	76
5.1.2	Suggestions from pilots	78
5.1.3	Next step	79
5.2	User test	80
5.2.1	Tests	80
5.2.2	Comments and observations	84
6	Result from user tests	87
6.1	Test results	87
6.1.1	Logged data	88
6.1.2	Questionnaire	91
7	Discussion and conclusion	94
7.1	discussing the results	95
7.1.1	User experience	95
7.1.2	Research experience	97
7.2	Answering the research questions	97
7.3	Challenges	100
7.4	Future work	101
7.5	Summary	102

CONTENTS

x

A Pilot test questionnaire	104
B Project schedule	109
Bibliography	113
Curriculum Vitae	118

List of Figures

1.1	Leap Motion is an optic hand tracking device. The device is able to track both hands and all 10 fingers	3
2.1	The GQM model. Each Goal has its own questions, while many questions can share the same metric	9
2.2	The GQM model. " A goal is defined with one or more research questions associated to the goal. The research questions are then measured using defined matrices. The picture is taken from (Basili et al., 1994)	10
3.1	Teddy in use on a display-integrated tablet. The picture is taken from (Takeo Igarashi, 1999)	22
3.2	ILoveSketch used by a professional product designer. The picture is taken from (Seok-Hyung Bae, 2008)	23
3.3	A concept of 3D image externalization. The picture is taken from (Hiroaki Nishino, 1998)	24
3.4	Different ways of deforming, moving, and rotating objects in 3D space. The picture is taken from (Hiroaki Nishino, 1998)	24
3.5	Hand motions create shapes which float above the Responsive Workbench. The picture is taken from (Steven Schkolne, 2001)	25
3.6	The user interface consists of a PHANToM haptic device, a standard 2D mouse, and on-screen GUI controls. The picture is taken from (Kevin T. McDonnell and Wlodarczyk, 2001)	26
3.7	System setup with cameras, finger props, and a physical sponge proxy with markers. The picture is taken from (Jia Sheng, 2006)	27

3.8	KidCAD system, with deformable gel and co-located project below, and secondary screen with 3D view above. Here a Zebra toy is imprinted, note that the zebra pattern is captured as well. The picture is taken from (Sean Follmer, 2012)	28
3.9	Setup of the mock-up builder. The picture is taken from (?)	29
3.10	Creating a 3D butterfly using BodyAvatar. (a) Scan the initial shape. (b) Drag antennas. (c) Sweep wings. (d) Sculpt a big belly. (e) Grow legs. (f) Paint color. The picture is taken from (Yupeng Zhang, 2013)	30
3.11	Operation effects on the avatar model in first-person mode. The picture is taken from (Yupeng Zhang, 2013)	31
3.12	The various operations designed for the manipulation of (a) the virtual handle bar, (b) a single object, (c) multiple objects, and (d) the virtual camera and their associated bimanual hand gestures. The picture is taken from (Peng Song, 2012)	32
3.13	Two cameras are used to track the position and gesture of the hands in 6 degrees of freedom. The picture is taken from (Robert Y. Wang, 2011)	33
3.14	They support a few simple gestures for 3D manipulation and camera adjustment	34
3.15	Magnetic orientation sensor used to rotate objects in 3D space	35
3.16	Four samples of different commands from (Zigelbaum et al., 2010). These four commands may be used with Leap Motion	36
3.17	Freeform can be used to sculpt a piece of clay, glass or plastic.	37
3.18	Children and adults could enjoy making 3D models at the Toronto Mini Maker Faire and get a physical version printed out using the 3D printer to the left.	38
3.19	model created by a bypasser at the Toronto mini maker fair using Hot Pot Factory's system.	39
3.20	Rough schematic of the Leap Motion device.	41
3.21	Leap Motion is already being integrated into keyboards and laptops.	41
3.22	The diagnostic visualizer displays some impressive results when it comes to detecting hands and fingers.	44
3.23	Kinect is an motion controlled optic device, designed primarily for gaming.	45
3.24	Wii Remote used for gaming.	46
3.25	PlayStation Move for playing games with motion controllers.	46

3.26	The STEM System consists of two motion controllers, wireless motion trackers (STEMs), and a stationary reference for each STEM.	47
4.1	Development tool for creating tasks, and assign them to members.	53
4.2	Strengths, weaknesses, opportunities, and threats (SWOT) matrix.	54
4.3	4+1 View model.	56
4.4	The main parts of the software can be hierarchically arranged, where the top is the highest level of abstraction, and the bottom is the lowest level of abstraction.	58
4.5	Leap Motion will scan for hands, and send all the data to the computer using USB. The computer has Unity installed, the shared libraries provided by the developers of Leap Motion is installed, and the software written in this project is also installed on the computer.	58
4.6	The logical view for the software made in this project. Logic is split into different packages, and the relation between the classes are represented with arrows.	61
4.7	Different strategies for rotating objects in the application can be used. The strategies can be changed at run time, or using Unity's inspector.	63
4.8	The software may run in different working modes, the user can choose which mode to use when working with the application. The different modes are: rotate, translate, and scale.	64
4.9	The sequence in which a hand is registered and updated in the system.	65
4.10	The sequence in which objects are rotated and translated in the game loop.	66
4.11	The different actions the user and the software are able to perform.	67
4.12	The hand is used to grab objects by folding it. This can be reversed if wanted.	68
4.13	A key tap gesture is recognized when the tip of a finger rotates down toward the palm and then springs back to approximately the original position, as if tapping. The tapping finger must pause briefly before beginning the tap.	68
4.14	The metaphor for the "Copy Hand Rotation" strategy. a) pick up an object, b) apply a rotation to the target object by rotating the hand, c) the finishing result of the target object after applying a rotation	70
4.15	The metaphor for the "Joystick Rotation" strategy. a) roll, b) pitch, c) yaw	71

4.16	An AnimationCurve may be used for non-linear equation evaluation without specifying a formula. The curve is editable in Unity's inspector.	72
4.17	How a circular gesture is made according to the Leap Motion's API. A circle movement is recognized when the tip of a finger draws a circle within the Leap Motion Controller field of view.	73
5.1	Setup for the pilot test.	76
5.2	First assignment for the test users. The purpose was to make the users familiar with moving objects around in 3D space.	77
5.3	Second assignment for the test users. Users were asked to create a snow man out of two spheres and two rectangular parallelepipeds.	77
5.4	A proposed method by one of the pilot users to do rotation. One hand will be used for grabbing and moving objects, while the other will be used for rotation within the specific area. The rotation area is the area the user needs to have its secondary hand in order to do a rotation.	79
5.5	Users were sitting in front of the computer with Leap Motion on their right hand side.	80
5.6	Users were asked to match the transparent 3D object and the opaque 3D object by rotating and translating the opaque object.	81
5.7	A strategy would be selected, and then described to the user. The user would then start the assignment, and complete as many levels as possible within the time limit (a level is simply to match the opaque cube to the transparent cube). After the time limit, a new strategy would be selected. If there are no more strategies to test, the user test would end.	82
5.8	The SwipeGesture class represents a swiping motion of a hand and a finger or tool.	86
6.1	The three charts display the error distance for the different strategies. The positions are stored as Unity units. Each circle/dot in the graph represents the result for one completed task by a user. The slightly bigger circle/dot represents the average score for each of the different axes.	88

6.2 Two objects are positioned exactly one Unity unit away from each other (on the x-axis). The sphere in the middle of each object represents the center of the objects. The charts are aligned next to each other to make it easier to compare the strategies 89

6.3 The three charts display the difference in rotation accuracy for the different strategies. The data are measured in angles. Each circle/dot represents one completed task for a user. The slightly bigger circle/dot represents the average score for each of the different axes. The charts are aligned next to each other to make it easier to compare the strategies 90

6.4 The chart displays the time spent for each task for each strategy. 91

6.5 Results from what users think of the “Copy Hand Rotation” strategy. Each bar represents the score for each of the questions in the questionnaire. The left most bar is the score for the top most question, and the right most bar is the score of the bottom question. 92

6.6 Results from what users think of the “Joystick Rotation” strategy. Each bar represents the score for each of the questions in the questionnaire. The left most bar is the score for the top most question, and the right most bar is the score of the bottom question. 93

6.7 Results from what users think of the “Two Handed Joystick Rotation” strategy. Each bar represents the score for each of the questions in the questionnaire. The left most bar is the score for the top most question, and the right most bar is the score of the bottom question. 93

7.1 Having the hand angled like the picture is not something that Leap Motion is able to read. The fingers are occluding each other, and Leap Motion thinks that it is just one finger pointing on the screen. 96

List of Equations

4.1 Copy Hand Rotation strategy 70

4.2 Gesture Rotation strategy 73

Chapter 1

Introduction

As traditional software for creating three-dimensional (3D) models are sophisticated and adapted to professionals, much research has been done to ease the use of creating 3D models for novice users. Leap Motion (see Figure 1.1) is an optic tracking system, consisting of a system of infrared cameras that lets the users control different applications by moving their hands above the device (lea, 2014). Leap Motion has sold “hundreds of thousands” of devices, although no official sales figures have been released (Metz, 2013), and the device is already being integrated into laptops and keyboards. Users will have an extra input interface available to them without the need of extra equipment. Creating 3D models utilizing the six axes of freedom from the Leap Motion) as well as an open space to move the hands around in, may present beginners with an easy and user-friendly way to create 3D models. Spending money on expensive and intrusive equipment, or spending a significant amount of time learning how to use traditional 3D modeling software such as Blender (ble, 2014) or Maya (may, 2014) is not required to start modeling. There are however a few 3D applications available in Leap Motion’s own store called Airspace which focus on molding virtual clay. These applications do not fully utilize Leap Motion’s potential, but a review of them can be found in Chapter 3.

The main objective of this project is to create a software for translating and rotating 3D models using Leap Motion as the input device, and to understand if it is possible to use the device to precisely position objects. The goals, and the metric used to evaluate the goals will be presented in Chapter 2. Different techniques for rotating objects will be implemented and tested by users. A user study will be conducted to figure out if novice user are able to use the system to rotate ob-

jects and position them in 3D space without using only their hands as controls. A pilot test will be held early in the project to quickly understand if the first proposed techniques are suitable for rotating objects. The last user test seeks to understand if users are able to precisely rotate and position objects by matching one object's orientation to another. The results from the user study will be outlined and discussed in Chapter 6. Hopefully, this research can help cover some of the limitations and opportunities with Leap Motion in the process of making the application, and study the results from the user tests.

Figure 1.1: Leap Motion is an optic hand tracking device. The device is able to track both hands and all 10 fingers



1.1 Background and goals

Using a motion tracking device to perform actions, commands, and gestures to manipulate virtual 3D objects without a mouse or keyboard is the problem to investigate in this project. The problem is of interest as motion tracking devices such as Leap Motion, and Microsoft's Kinect are becoming affordable for consumers, and may help novice users to easily create models in 3D space. Research about what has been done to ease the use of creating 3D models for beginners, which problems needed to be solved, and the main goals for this project will be given in the following sections.

Literature survey

An introduction to similar research in the field of 3D modeling will be given in this section.

3D objects (or 3D models) are typically made using complex software based on a conventional graphical user interfaces(GUI). 3D models and animations are common in engineering, design, commercials, games and movies. Much research has been conducted to make the process of making 3D models easier for users unfamiliar with the complex 3D-modeling softwares such as Blender ([ble, 2014](#)) and Maya ([may, 2014](#)).

Interpreting 2D sketches into 3D shapes (sketch-based modeling), or constructing and manipulate 3D shapes using hand gestures performed in 3D space (gesture-based modeling) are common approaches to help novice users create 3D models. Some technologies to create 3D digital models using a physical medium (or a physical proxy) are also available and fall in the gesture-based category. It is worth mentioning that technology and research that scan real world objects to create 3D models exist. These methods will not be discussed in this project as they are not concerned with how to manipulate objects in 3D virtual space. Teddy ([Takeo Igarashi, 1999](#)) is an example of a sketch-based modeling system. You can draw a 2D contour, and a 3D model will be created based on the 2D sketch. 6D-hands ([Robert Y. Wang, 2011](#)) is an example of a gesture-based modeling system. They use cameras to track your hands in order for you to assemble 3D-models in a 3D space.

Rotating objects are primarily done by using two hands, and a handlebar metaphor ([Peng Song, 2012](#)) seems to be a good way to rotate and move objects in 3D space. Selecting and moving objects are usually done by pointing at them, or selecting them with a given interaction device. The user can then move the objects in virtual space according to the position of the device they are holding in their hand(s). These are the main aspects from the literature survey that reflects the goals in this project. An extensive literature study has been conducted, and the findings from the study will be given in Chapter 3.

What remains to be done?

Issues such as fast fatigue is a perennial problem when using mid-air interaction for precise control is a problem that is difficult to eliminate, and will also be difficult to solve with Leap

Motion.

Most applications require users to use extra (intrusive) equipments, such as markers, or controls. Eliminating those problems require technology such as Leap Motion and Kinect.

Seeking increasingly facile ways to bridge the gap between physical and digital media in order to effectively turn humans' mental concepts into reality, and to remove the confines of the mouse from the hand is a grand and vision. This research might bring us a little bit closer to achieve this goal by continue to explore alternative input and interaction modalities that exploit inherent human capabilities.

Goals

The goals for this project are to find out how Leap Motion can be used as an interface for creating 3D digital models. Motion controllers might be a new way of interacting for most people, so it is interesting to see how they grasp the technology, and how they are able to use it. Developing a motion controlled application on a computer is a fairly new concept, and figuring out what options that already exists may be of great help when developing the application in this project. This study do not seek to create a fully functional application, but to focus on a few areas within 3D modeling using the hand tracking device, Leap Motion, as the interface to interact with. The goals are:

- G1: The study seeks to understand and evaluate how users may assemble 3D models using Leap Motion.
- G2: The study seeks to understand how other applications create 3D models.

Each of these goals have two to three research questions associated with them, and these can be found in [Chapter 2](#).

1.2 Limitations

There are some limitations to this project. A thorough analysis of the strengths, weaknesses, opportunities, and threats to the project can be found in [Section 4.1.4](#). The main limitation

is that Leap Motion is not a device that all people have. It is not as common as mouse and keyboard, and it is therefore difficult to make a test that can be sent to users, and let them try the application on their own.

1.3 Approach

An introduction to the scientific approach used to meet the main objectives mentioned in Section 1.1 will be explained here. The scientific approach used in this project is an engineering approach (Atila Ertas, 1996). "G1" will mainly be answered by implementing a software, test it, and evaluate the results obtained from the user tests. "G2" will mainly be answered by having a literature study, focusing on the information concerning interaction with the 3D models. The information gained from the literature study will be evaluated and discussed to help achieve the goal. A detailed explanation of the scientific approach will be given in Chapter 2.

1.4 Structure of the report

The structure of the report starts with an introduction, following the methodology used. The next three chapters about literature survey, development process, and user study are the method used to answer the goals, while the result is the chapter quantifying the data from the two previous ones. The reports final chapter is a discussion about the project itself.

Chapter 1, Introduction provides the reader with some information about what the project is about, and a presentation of the main goals.

Chapter 2, Methodology gives an introduction to the methodology used in this project. Goals, research questions, and methods used to evaluate and answer the questions will be described.

Chapter 3, State of the art outlines similar research in the field of 3D modeling. Technologies similar to Leap Motion will be discussed and compared to Leap Motion. Information about applications that enable the user to create or manipulate objects in 3D virtual space with other means than a computer mouse will also be evaluated.

Chapter 4, Development process contains information about how the application in this

project was made on. The development process will be given, tools used will be discussed, and how the project was conducted in terms of planning and workflow will be outlined.

Chapter 5, User study gives an overview of all the user study that was done in this project. How user tests were planned and scheduled, as well as discussing some of the results and comments made by users.

Chapter 6, Result discusses and outlines the results from the literature study made in Chapter 3. The results from the user tests from Chapter 5 will be quantified, and answering the research questions from Chapter 2 will also be given.

Chapter 7, Discussion is the final chapter, and will briefly discuss challenges that arose during the project as well as what went well. Discussion about the results and future work following a conclusion will wrap up the project.

Note about language

The language used in this report is American-English. When the word “we” appears in the text, “we” is meant to be the reader(s) and the writer. Some words are used interchangeably in this report. The words “he” and “she” are being used interchangeably, and should be read as he/she. When writing about the software made in this project, words such as “application”, “software”, “system”, and “program” are being used interchangeably, and they all refer to the same software implemented in this project. “thesis” and “report” are used interchangeably, and both mean this written document. “3D model” and “3D object” are used interchangeably, and both mean the same. Text or words taken from the computer code written in this project will be displayed with a different font, such as `this`, in this report. It is easier to distinguish real world objects and software objects in the report using different fonts. Words such as `LG.Hand` and `hand` are more easily distinguished, where `LG.Hand` is a software term, and `hand` is a real hand.

Chapter 2

Methodology

In the introduction, we found that it is interesting to investigate good commands, gestures, and actions to use when manipulating virtual objects using Leap Motion, in order to ease the use of creating 3D models for beginners. In this chapter, the methods used to solve the established problems will be outlined. The Goal Question Metric (GQM) approach was used to analyze and evaluate the results of the study and will be described in Section 2.1. In the next section, a description of which framework to use and how to use it when evaluating the goals for the project will be given. The DECIDE framework was used together with the results from the GQM approach as the evaluation process, and the framework will be presented in Section 2.2. In the final section, the framework/guidelines from the previous section are being used to evaluate the project. The choices made in the evaluation process will be given in Section 2.3.

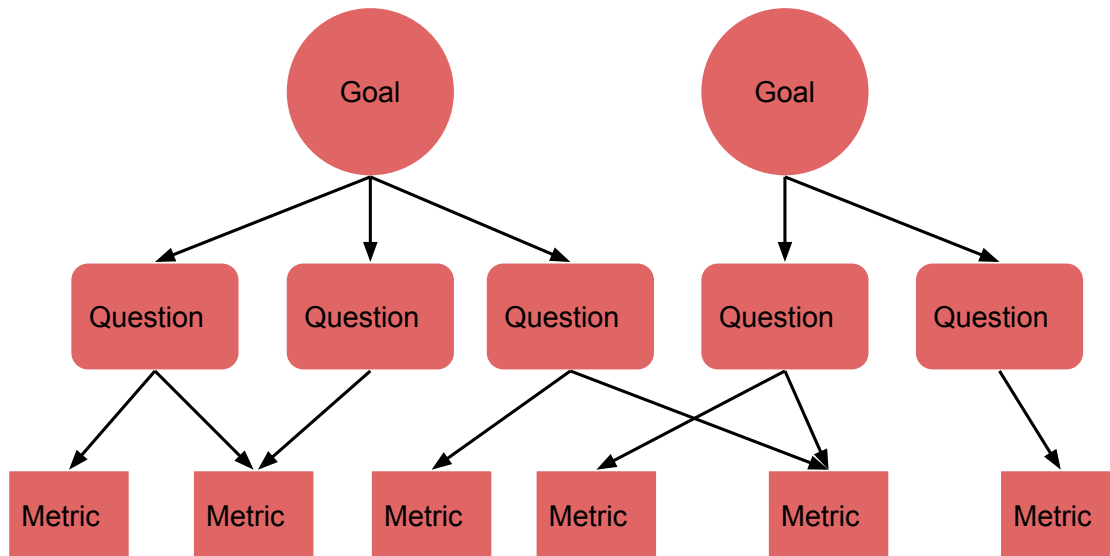
2.1 Research questions and goals

In this section, the GQM approach will be described, as well as how the GQM approach have been used to define the main goals for this project.

2.1.1 The goal question metric

The GQM model is used to define and evaluate goals for a particular project in a particular environment (Basili et al., 1994). A GQM model is a hierarchical structure as seen in Figure 2.1.

Figure 2.1: The GQM model. Each Goal has its own questions, while many questions can share the same metric



One or more goals are specified, and these are the main objectives for an organization. The goals are refined into several questions, which is questions that can be answered to fulfill the goals. Each question is then refined into metrics, a way of quantifying the results from the questions in order to make conclusions and achieve the goals. The metrics are used to get an accurate answer to the research questions. A metric can be a combination of measurements (such as meters, or seconds), questionnaires, and/or literature studies. An example of a complete GQM model can be seen in [Figure 2.2](#).

Figure 2.2: The GQM model. " A goal is defined with one or more research questions associated to the goal. The research questions are then measured using defined matrices. The picture is taken from (Basili et al., 1994)

Goal	Purpose Issue Object (process) Viewpoint	Improve the timeliness of change request processing from the project manager's viewpoint
Question		What is the current change request processing speed?
Metrics		Average cycle time Standard deviation % cases outside of the upper limit
Question		Is the performance of the process improving?
Metrics		$\frac{\text{Current average cycle time}}{\text{Baseline average cycle time}} * 100$ Subjective rating of manager's satisfaction

In this project, it is helpful to understand what the research questions are, and how they should be evaluated. It is somewhat different from an ordinary GQM model, as the metrics are not easy to quantify in this project. However, the model may also be used for qualitative studies, such as a literature study and interviews.

This study seeks to investigate how the new technology Leap Motion can be used to assemble 3D models. A focus on human-machine interaction and technology are the main aspects of the study. Two goals have been suggested. One regarding human-machine interaction, and the other regarding motion-controlled technology. The specific goals can be found in Section 2.1.2. Since the scope of the project is limited to one semester, only a limited range of research questions have been suggested. It was important to have questions that did not involve a total change in the prototype after a user test, as it may take a significant amount of time to develop different versions of the prototype. The goals, questions, and metrics arrived at based on the GQM model can be seen in Section 2.1.2

2.1.2 Main objectives

The goals, questions, and metrics in this section will have the following structure:

Goal

- Research Question
 - Metric

Answering these questions are the main purpose of this research, and is also an important part of the project. Development of the software is of the necessary for the evaluation.

G1: The study seeks to understand and evaluate how users may assemble 3D models using Leap Motion

- RQ1.1: What are good commands, actions and gestures for navigating, translating, and rotating objects in 3D space using Leap Motion?
 - Results from user tests with different control schemes. Users will give a score from one to five when rating control schemes.
 - Questionnaires and interviews.
- RQ1.2: Do people find the controls intuitive and easy to use?
 - Results from user tests with different control schemes. Users will give a score from one to five when rating control schemes.
 - Questionnaires and interviews.
- RQ1.3: Are users able to rotate and position objects accurately and fast?
 - Results from user tests with different control schemes. Log the time spent(seconds) and accuracy(rotation and position) by completing tasks in the user test.
 - Questionnaires and interviews.

Each of these bullet points will be explained in this section, and acts as guidelines for how to evaluate the project. How they were used in this project will be explained in Section [2.3](#).

G2: The study seeks to understand how other applications create 3D models

- RQ2.1: What are common ways to navigate, translate, rotate, and scale objects in other applications where a traditional mouse is not the main interface?
 - Literature study of the relevant technologies.
- RQ2.2: Can Leap Motion introduce new/improved ways of navigating, translating, and rotating objects in 3D space compared to similar technologies?
 - Literature study of the relevant technologies.
 - Results from user tests with questionnaires and interviews.

To explain what what are meant by commands, actions, and gestures mentioned in RQ1.1, some examples will be given:

- *Gestures*: A gesture is a recognized movement by the user. Leap Motion looks for certain movement patterns to distinguish it as a gesture. For example, a movement from side to side with the hand can indicate a swipe gesture, while a finger poking forward can indicate a screen tap gesture.
- *Commands*: A command is similar to a gesture, but a command is a discrete movement instead of an analog one. An example of a command is to fold the hand, or to open the hand.
- *Actions*: actions are movements the user do with her hand, that is not a common movement pattern. An example of an action is to move the hand in and out of Leap Motion's field of view.

2.2 Evaluation framework

The framework chosen for evaluation is the DECIDE framework ([Sharp et al., 2007](#)). The framework provides a checklist which is helpful to evaluate the project.

- Determine the overall goals that the evaluation addresses (see Section [2.2.1](#)).

- Explore the specific questions to be answered (see Section [2.2.2](#)).
- Choose the evaluation paradigm and techniques to answer the questions ([2.2.3](#)).
- Identify the practical issues that must be addressed, such as selecting participants (see Section [2.2.4](#)).
- Decide how to deal with the ethical issues (see Section [2.2.5](#)).
- Evaluate, interpret, and present the data (see Section [2.2.6](#)).

2.2.1 Determine goals

The high-level goals of the project should be clarified as the first step in planning an evaluation. The goals for this study have already been outlined in Section [2.1.2](#). The goals influence the evaluation approach and guides the study.

2.2.2 Explore questions

Specific questions must be answered in order to satisfy the high-level goals. These questions can be broken down into more specific sub-questions to make the evaluation even more specific. The questions for this study can be found in Section [2.1.2](#).

2.2.3 Evaluation paradigm

When the goals and questions are have been identified, an evaluation paradigm and technique must be chosen. The most reasonable set of methods to evaluate the questions must be provided. In this study, the evaluation consists of user tests, interviews, and a literature study.

2.2.4 Practical issues

There are many practical issues to consider when doing an evaluation, and it is important to identify the issues before doing an evaluation. These issues may be equipment, users, facilities, schedule and budget, and evaluator's expertise. Users are a key aspect of this project. Finding

appropriate users are necessary in order to get an accurate evaluation. Different users from different professions, gender, cultural diversity, education experience, and personality differences need to be taken into account when evaluating the product. Furthermore, user tests need to be held in suitable environments and according to the guidelines given in (Svanæs, 2012). A detailed description of how the user tests were conducted can be found in Chapter 5.

2.2.5 Ethical issues

Ethical issues concern the privacy of users. Names and personal information are not taken into account when conducting user tests. Age, gender, and profession are usually enough of personal information. In cases where more personal information is required, it should be kept confidential. Users should be informed about the evaluation they are taking part of, and know that it is not the users who are being tested, but the system. The users are also free to leave the test when they feel like it. When people spend their free time to help with an evaluation, they should be treated as good as possible, and not feel uneasy. Guidelines given in (Svanæs, 2012) should be followed. A detailed description of how the user tests were conducted can be found in Chapter 5.

2.2.6 Evaluate, interpret, and present data

Choosing the evaluation paradigm to answer the research questions is an important step. Decisions on how to collect data, which data to collect, and how to analyze it must be made.

It is also important to ask questions such as:

- is the evaluation technique reliable?
- is the evaluation technique measuring what it was supposed to measure?
- do biases occur?
- what is the scope of the evaluation study?
- is the environment suitable?

2.3 Evaluation process

We have explained the steps in the DECIDE framework, and will now describe how this project will be evaluated. An explanation of how the evaluation process took place will be given in this section. Research question RQ2.1 will be answered through the literature study which can be read in Chapter 3.1. RQ2.2 will be partial answered with the literature study. Experimenting with the Leap Motion technology using the API provided by the developers of Leap Motion together with user tests will answer RQ2.2.

2.3.1 Literature study

The literature study consists of a literature review of articles about creating 3D-models. The literature study seeks to answer RQ2.1 and RQ2.2 as well as to understand how the techniques used in other hand tracking systems work. RQ2.1 will be answered by looking at commands, gestures, and actions that have been working well in similar projects. RQ2.2 will be answered by adapting the techniques that have worked well in similar research, as well as introducing actions, commands, and gestures well suited for Leap Motion by utilizing the API provided by the Leap Motion software developers. The study is useful when implementing the software in this project since techniques used in similar research was an inspiration in this project as well.

2.3.2 Software implementation

A prototype is needed in order to answer all the research questions (except for RQ2.1). The prototype will undergo many changes during the project. It is important that the prototype is modifiable and flexible. Different patterns such as the strategy pattern and the decorator pattern will shape the software. A detailed description of the design and architecture of the software will be given in Chapter 4.

Since the scope of the project should fit within one semester, the software will not be a finished product, but a proof of concept. The purpose of the project is meant to understand how users cope with the technology, thus giving an evaluation of the specific parts of the study.

2.3.3 Evaluation paradigm

The evaluation of the software will be an engineering approach ([Atila Ertas, 1996](#)). For this project, an engineering approach is a well suited evaluation paradigm since it requires user tests to judge the quality of the system in order to answer the goals. The engineering design process contains a set of steps involved in creating or inventing something new. The steps involved can be defined as the following:

- Define the problem
- Do background research
- Specify Requirements
- Brainstorm solutions
- Choose the best solution
- Do development work
- Build a prototype
- Test and redesign

Each of these steps will not necessarily be visited only once, but many times during the project through iterations. The system will undergo subsequent user tests throughout the project, thus applying changes to the software after a user test. Interviews and questionnaires will be made in order to get both quantitative data and qualitative data. It is interesting to understand why users find a method better than another one. A formative study with more than five users would help to get a basic understanding of how users approach the system, and also to underline the difficult parts of the system, as well as the parts that work well. It is helpful in order to know what parts of the system need improvements. The user tests will be held in a user's typical environment, which is in front of their desk. The user tests will also be held in a controlled environment, as the users will be observed and given different tasks to solve ([Basili et al., 1994](#)). The software will undergo changes after user tests in order to improve faults with the software.

2.3.4 Practical issues

The most practical issue with this study is that the software needs a Leap Motion in order to work properly. Leap Motion is not distributed to a lot of people, and it is difficult to find a significant amount of users to test the application. More about the issues regarding this project can be found in Section 4.1.4 Leap Motion is a fairly new device, and we might see more of them in the future. There are online communities that like to discuss applications using Leap Motion¹. These communities may be of help in order to have a broader scope of applicants to test the software implemented in this study. It is difficult to control a user test when you are not observing the user using the application. It may be possible to let online communities test the application, but the user tests will not be equally comparable to the user tests under controlled circumstances.

2.3.5 Ethical issues

As mentioned in Section 2.2.5 there are some ethical issues to consider when dealing with users. This study needs age and gender, but name and other personal information are not necessary. There might be a need to ask the users for additional information after the user test, thus requiring an email-address for each user. It is optional for the users to provide their email address. In the case of receiving users' email addresses, these need to be kept confidential and should not be misused.

2.3.6 Questionnaires and interviews

The questionnaires will be used to ask questions about how it was to use the application. A similar form as the System Usability Scale (SUS) suggested by (Brook, 1996) will be used to evaluate the software. Inspiration from that form and some of the questions will be applied to the questionnaire in this project. The questionnaire should be

The interviews will consist of questions regarding the problems observed during the user test. These questions may be different from user to user, as different users might experience

¹<http://www.reddit.com/r/leapmotion/> is a site for leap motion enthusiasts, both developers and end-users. Accessed 28-03-2014

different problems or issues during the user test. It is interesting to understand why there were issues, and how they can be solved. The interview will also consist of a few general questions about what they think could be done differently. It is understandable that users' time is precious, making the interview optional. The users are encouraged to speak freely, but it is important to have an agenda to stay on topic. A detailed description of how the user tests were conducted can be found in Chapter 5.

2.3.7 Interpreting data

After the user tests, a great amount of data will be gathered. The data need to be evaluated. Evaluation methods have different properties that must be considered, as mentioned in Section 2.2.5. An outline of the different properties will be answered in this section.

Reliability

Because of the fact that the user tests are in controlled environments with specific tasks, it should be fairly simple to replicate the experiment and get similar results ([Sharp et al., 2007](#)).

Scope

It is important to notice that making any generalizations about how users relate to a broad category such as 3D-modelling software using tracking devices is difficult after studying only one type of application with a limited number of users. This means that the scope is not broad. The application is not something that users use daily, and it will only cover a specific field of 3D-modeling, which means that it is difficult to generalize the answer from the users in this study. People have different personalities, and creating 3D models might be something that users are not familiar with, or even know exists. Some might be skeptical to the technology and the concept, while others might think it is enjoyable. This may distort the data if only one group of people are to test the application. It is probably more preferable to test the application on users who would want to use 3D-modeling applications since they are the target group for such a system. Even though the scope is limited, all information gathered are interesting data to evaluate.

Validity

Considering that Leap Motion is starting to get integrated into laptops and keyboards, Leap Motion might be positioned differently from machine to machine. The device used in this project is an external device, and can be positioned where it feels the most comfortable or practical. The position of where the leap is placed may affect the validity of the project. For the user tests in this project, the Leap Motion will be placed at the same spot for each user test, making it possible to accurately replicate the experiment made in this project.

Bias

Interviews can easily be interpreted differently from person to person. It can be the tone in the way the interviewer asks questions, or it can depend on the mood of both the user and the interviewer. Some questions might be leading questions, which should be avoided at all costs. The observation of the user tests might be interpreted differently from person to person doing the observation. It is easy to see problems from a subjective point of view instead of an objective point of view. This is unfortunate in the case that the observer has specific desires or intentions from the user tests and the project. Take recordings of the user tests are possible, but only with the users' permission. In this way, it is possible to have observers from outside the project, and get a different point of view from them. These persons may be fellow students or professors in the same field of study.

Chapter 3

State of the art

In the previous chapters, we have found that Leap Motion may be used for controlling 3D objects in virtual space, and we have followed the DECIDE framework and GQM model to be the evaluation process in this project. In this chapter, relevant theory related to the project will be presented. The reason for having a literature study is to figure out what has previously been done in the field of creating digital 3D models. Not only will 3D-modeling applications using hand-tracking devices be evaluated, but research concerning the creation of 3D models in general, as well as research that serve to navigate in 3D space by other means than a computer mouse. Inspiration from the findings will be taken into account when developing the system in this project. Research question RQ2.1 and RQ2.2 (See Section 2.1.2 for the list of research questions) are relevant in the literature study, as they will cover convenient methods to navigate, translate, and rotate objects in 3D space without using a computer mouse. Applications where you can create 3D models using Leap Motion will be studied as well, and an explanation of the different applications will be covered in Section 3.2. An introduction to motion tracking technologies available to consumers will be given in Section 3.3 to give an overview of which alternatives are available for consumers for a reasonable price, and could be an alternative to Leap Motion.

Even though the research in this project is concerned about how Leap Motion can be used to rotate and translate objects, understanding how other applications create 3D models is important for further studies. A future goal on how to use Leap Motion to create 3D models inspired by research made of others, including how to manipulate the virtual objects or models may

be interesting to investigate. It is also worth mentioning that this project is concerned about contributing to a creative modeling tool rather than using the device for digitizing real-world objects.

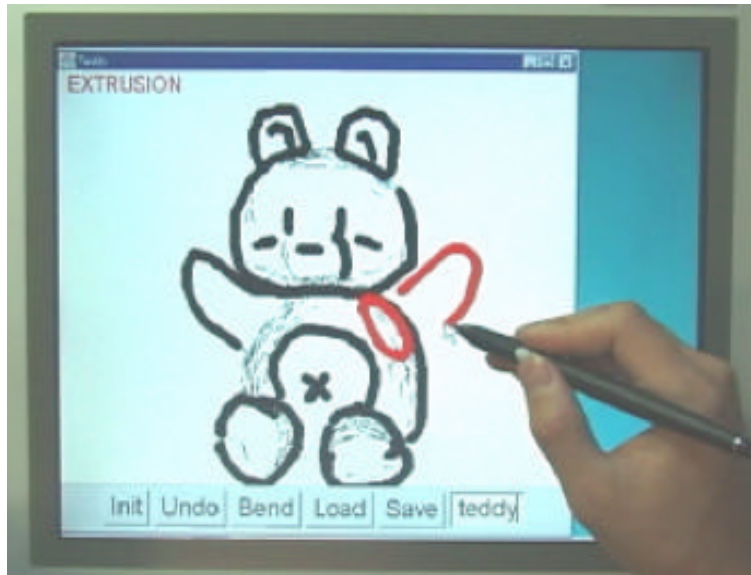
3.1 Literature study

Much research have been done in the field of 3D modeling. This section will cover relevant research around 3D modeling using unconventional user interfaces without a mouse as the main interface. We can make a separation between sketch-based modeling and gesture-based modeling. Creating 3D models out of 2D sketches is part of a sketch-based modeling approach which will be described in Section 3.1.1. Constructing and manipulating 3D models using hand gestures performed in 3D space are part of the gesture-based modeling approach which will be described in Section 3.1.2. These methods are designed to help novice users create 3D models without the need to delve in the sophisticated 3D modeling applications such as Blender(ble, 2014) and Maya(may, 2014). A section about how other applications that not necessarily focus on making 3D models, but focus on navigating, translating, and rotating objects in 3D space will be given in Section 3.1.3.

3.1.1 Sketch-based

The common task of sketch-based methods are to create 3D models out of 2D sketches. The user can draw a circle, and a sphere would then be the output 3D model from the 2D sketch. The application Teddy (Takeo Igarashi, 1999) interprets a user's 2D silhouette sketch, and creates a 3D shape of the sketch. The essential idea is the use of freeform strokes as an expressive design tool. Teddy supports operations like extrusion, bending, and cutting. Teddy's physical user interface is based upon traditional 2D input devices such as a standard mouse or tablet. They use a two button mouse with no modifier keys. Their software supports direct camera manipulation using the secondary mouse button based on a virtual trackball model (Hultquist, 1990). A picture of the software can be seen in Figure 3.1.

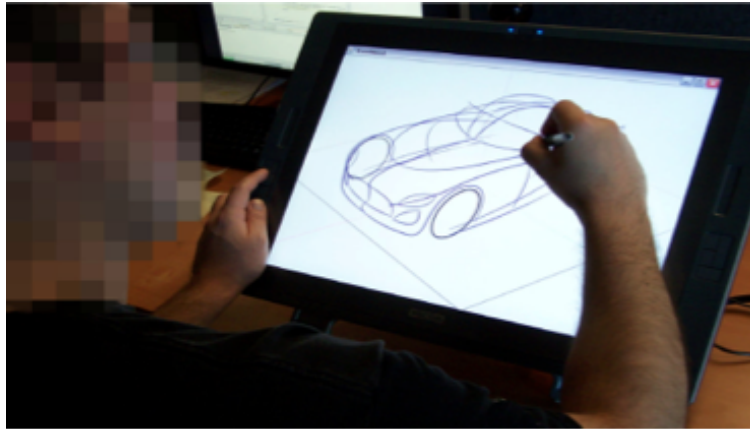
Figure 3.1: Teddy in use on a display-integrated tablet. The picture is taken from (Takeo Igarashi, 1999)



A more sophisticated application named ShapeShop (R. Schmidt, 2005), inspired by Teddy, support creations of more complex and detailed 3D models. Their sketch-based modeling interface has been designed primarily to support use on large interactive displays. The idea to create models in ShapeShop is the same as with Teddy. A user can draw a 2D sketch, and the software will create a 3D model out of the sketch.

I Love Sketch (Seok-Hyung Bae, 2008) is a 3D curve sketching system that captures some of the affordances of pen and paper for professional designers, allowing them to iterate directly on concept 3D curve models. The sketch based models require pen input and are well suited for people's natural drawing abilities.

Figure 3.2: ILoveSketch used by a professional product designer. The picture is taken from (Seok-Hyung Bae, 2008)



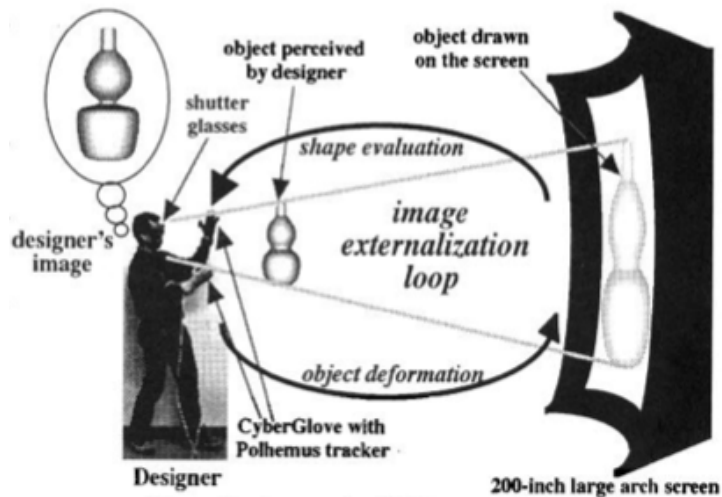
These applications could be extended with Leap Motion as the input device. Even though Leap Motion works in 3 dimensions, it is possible to treat it like a 2D input device, ignoring the depth information. Users would then be able to draw 2D shapes in the air, and have the same functionality as in the applications mentioned. Using Leap Motion may also introduce extra functionality to the existing softwares as it can be used for more than just a pen tool (e.g. using special hand gestures).

3.1.2 Gesture-based

Construct and manipulate 3D shapes using hand gestures performed in 3D, often based on virtual sculpting metaphors are the main purposes of gesture-based methods. Most applications using the gesture-based approach requires expensive equipment, are intrusive, not mobile, and it takes a significant amount of time to set it up.

(Hiroaki Nishino, 1998) lets you create a 3D model using bimanual gesture gloves, shutter glasses and a 200 inch stereoscopic screen, enabling the user to perceive depth information of the model (see Figure 3.3).

Figure 3.3: A concept of 3D image externalization. The picture is taken from (Hiroaki Nishino, 1998)

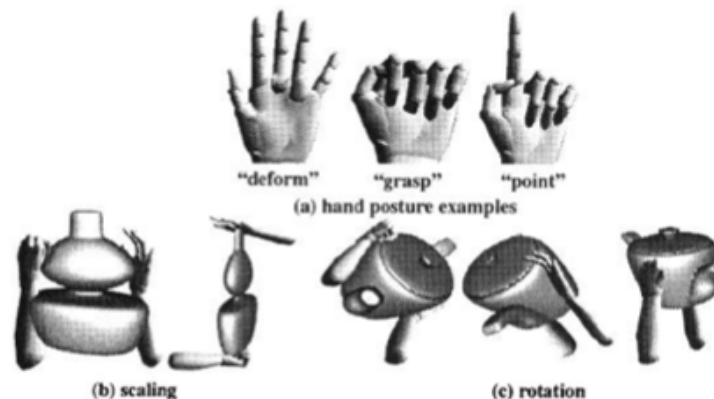


The application lets a user create a model by following three steps:

1. prepare some primitive shapes.
2. combine these primitives to form a rough shape.
3. deform the rough shape to make it finer one.

They implemented a high-performance gesture learning and recognition function, allowing the users to register their preferred gestures right before using the system (Barr, 1981). The idea is to create primitive shapes, and then blending them together to make a single shape. From their system, it is possible rotate objects by grabbing them with two hands as seen in Figure 3.4.

Figure 3.4: Different ways of deforming, moving, and rotating objects in 3D space. The picture is taken from (Hiroaki Nishino, 1998)



Drawing organic surfaces in the air using your hands is proposed by (Steven Schkolne, 2001). They demonstrate a magnet tool which is attached to the hands, and the user can sketch surfaces in the air above a responsive workbench. The users can see their model over the bench using stereoscopic shutter glasses as seen in Figure 3.5. Their interaction philosophy is to make a path of the hand in space to form marks. Marks are characterized by (W, 1995) as continuous streams of uninterpreted coordinates in space representing the user's movement.

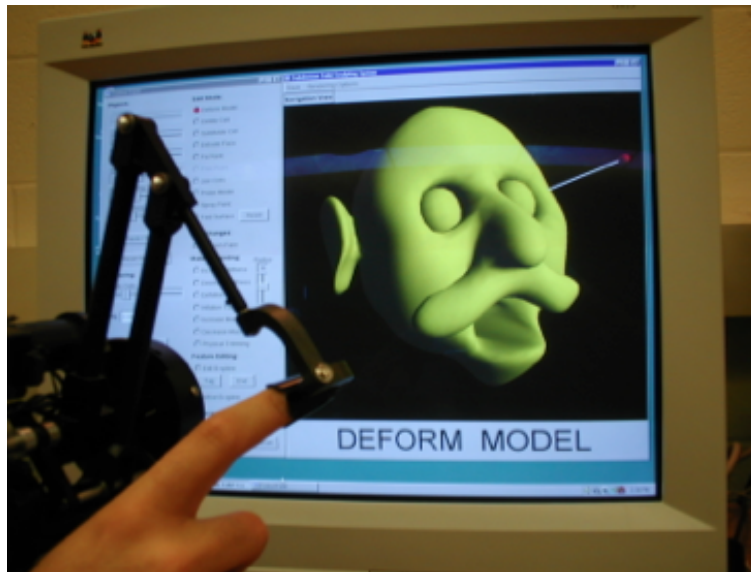
Figure 3.5: Hand motions create shapes which float above the Responsive Workbench. The picture is taken from (Steven Schkolne, 2001)



Users are able to move objects around by using a kitchen tong as a tool outfitted with magnetic trackers and sensors to detect when they are closed. When a tong is closed, the user can grab an object and move it in 3D virtual space. A second tong can be used together with the first to scale objects by moving the tongs apart from each other to increase the size of the object. Moving the tongs together will shrink the scale of the object.

(Kevin T. McDonnell and Wlodarczyk, 2001) have developed a sculpting system that allows the deformation of elastic virtual clay using a haptic-based interaction tool, providing a realistic sculpting experience. The setup of their system can be seen in Figure 3.6.

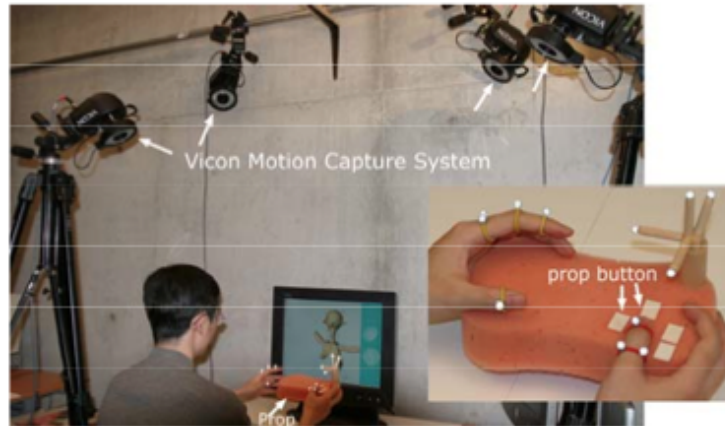
Figure 3.6: The user interface consists of a PHANToM haptic device, a standard 2D mouse, and on-screen GUI controls. The picture is taken from (Kevin T. McDonnell and Wlodarczyk, 2001)



They use different tools to apply forces on the user's hand and to cause geometric changes to the 3D model (which behaves like a semi-elastic virtual clay). One way of using a haptic tool is to deform a shape much in the same way that an artist mold a piece of clay with her fingers. Geometrical tools are used to select cells in an object using a 3D cursor to either extrude, subdivide, or extrude them. Physical tools are introduced to shape an object by editing physical properties such as mass points and stiffness distribution. Objects can be shrunk or inflated. The article does not mention how objects are rotated nor translated.

(Jia Sheng, 2006) propose an interface for virtual 3D sculpting using a physical proxy. Their application is able to manipulate 3D models using a sponge as a proxy. They use camera-based motion tracking technology to track passive markers on the fingers and the physical proxy as seen in Figure 3.7.

Figure 3.7: System setup with cameras, finger props, and a physical sponge proxy with markers. The picture is taken from (Jia Sheng, 2006)



They use polygon meshes as the underlying representation of the virtual models their system. They created what they call a PropWidget that moves in accordance with the movement of the physical prop (the sponge), similar to the mouse pointer being a PropWidget for the physical mouse. Translating and rotating models are achieved by movements of the prop in the physical 3D space while pressing the prop's middle marker. Scaling is done by having the non-dominant thumb pressing on the prop's two right markers. They also use an arcball technique(Shoemake, 1992) to perform rotations when the PropWidget is not directly connected to the virtual mesh.

A study by (Sean Follmer, 2012) introduce a digitally clay interface for children to remix toys. The system enables users to imprint 2.5D shapes from physical objects into their digital models by deforming a malleable gel input device as seen in Figure 3.8.

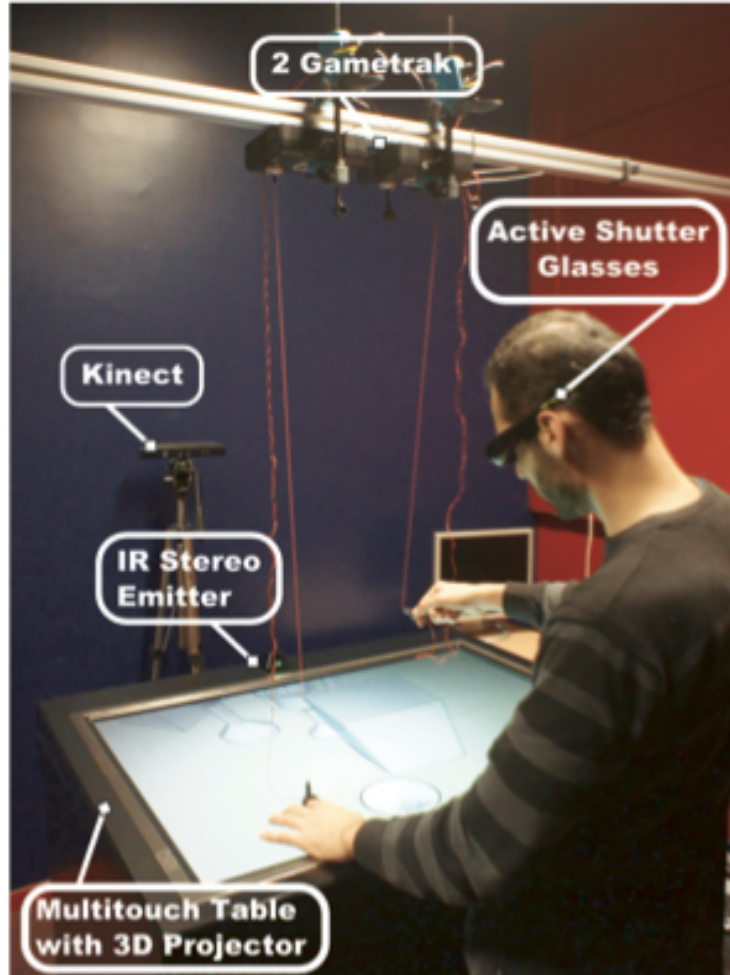
Figure 3.8: KidCAD system, with deformable gel and co-located project below, and secondary screen with 3D view above. Here a Zebra toy is imprinted, note that the zebra pattern is captured as well. The picture is taken from (Sean Follmer, 2012)



(Bruno R. De Araújo and Jorge, 2012) present on-and-above-the-surface interaction techniques to take advantage of the interaction space for creating and editing 3D models in a stereoscopic environment. It uses a 42 inch stereoscopic multi-touchable surface together with Microsoft's Kinect camera (kin, 2014), and two Gametraks¹ to track the hands and fingers above the surface as seen in Figure 3.9.

¹See <http://en.wikipedia.org/wiki/Gametrak> for details

Figure 3.9: Setup of the mock-up builder. The picture is taken from (?)

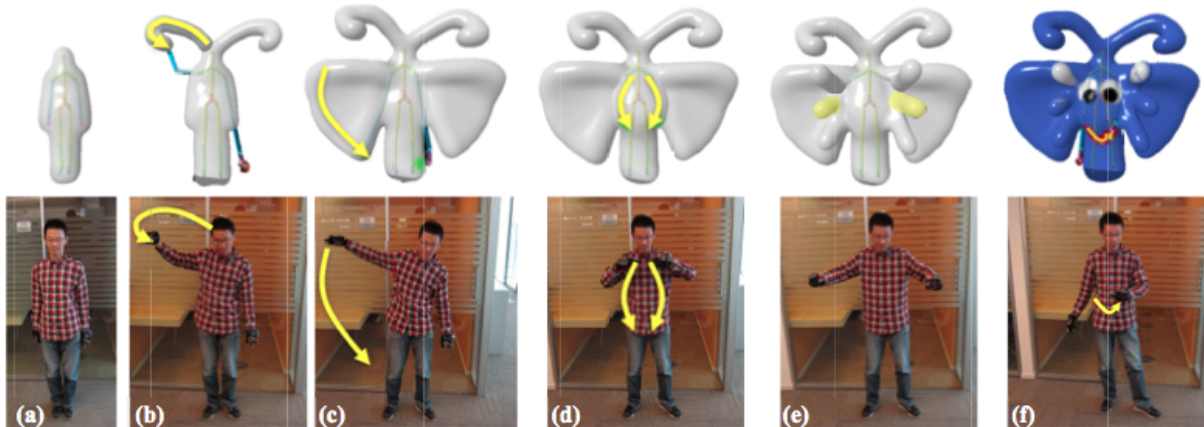


They have different input methods for their system. One is called on-surface, and the other is above-surface. The on-surface input method lets the user use the screen as a multi-touch device, using the fingers to draw sketches and moving objects around. The above-surface input method requires trackers attached to the index finger of each hand. A user are able to move objects using the non-dominant hand, while the dominant hand is used for sketching. Using both hands, the users are able to scale and rotate objects. Scaling is done by moving the hands closer or further apart from each other. Moving the hands together will shrink the model, and moving the hands apart from each other will increase the scale of the model.

BodyAvatar ([Yupeng Zhang, 2013](#)) has done recent research on how to create 3D models in a centered first-person “you are the avatar” metaphor using Microsoft’s Kinect. This approach helps eliminate some of the difficulties with the mental understanding of 3D models on a 2D

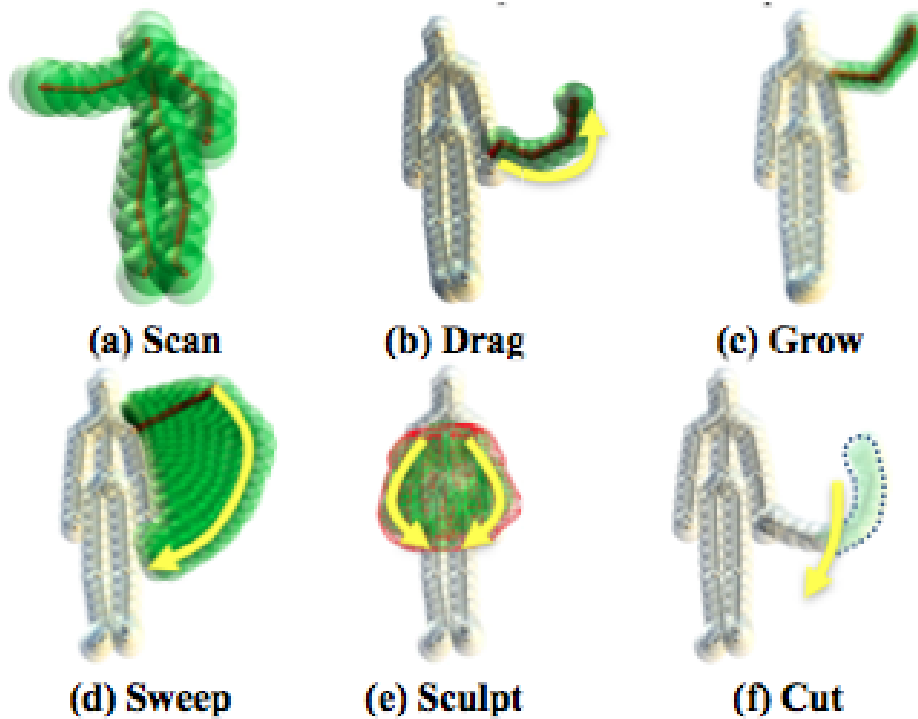
screen. Figure 3.10 displays a series of pictures of how the system can be used.

Figure 3.10: Creating a 3D butterfly using BodyAvatar. (a) Scan the initial shape. (b) Drag antennas. (c) Sweep wings. (d) Sculpt a big belly. (e) Grow legs. (f) Paint color. The picture is taken from (Yupeng Zhang, 2013)



A user can work in two different modes. One is inside the avatar (first-person mode), and one is outside of the avatar (third-person mode). From inside the avatar, the user is able to create a base model based on the user's initial position, and then further build the model by using the hands (with a fist pose) to extrude/grow parts of the model. It is also possible to sweep using a flat hand pose, which adds a surface with a given thickness to the avatar (e.g. creating a wing). The user can sculpt the model using two hands (with a flat hand pose) by tracing around an imaginary rotund volume rooted on a user's body. A user can only add more "clay" to the model by sculpting, making it possible to make the model bigger than how far a user can reach with her hands. It is also possible to cut pieces of the model using a pistol pose. All of the commands mentioned are illustrated in Figure 3.11. From outside the avatar, the user is able to scale, translate, and rotate the objects using two hands as if the objects are skewered on a bimanual handle bar, similar to what presented in (Peng Song, 2012). The models in the application are modeled by an implicit surface constructed from a number of meta-balls in 3D space (H. Nishimura, 1985).

Figure 3.11: Operation effects on the avatar model in first-person mode. The picture is taken from (Yupeng Zhang, 2013)



They found that users were able to create avatars under 12 minutes, where less than seven minutes was used for modelling the shape, and less than five minutes for painting colors.

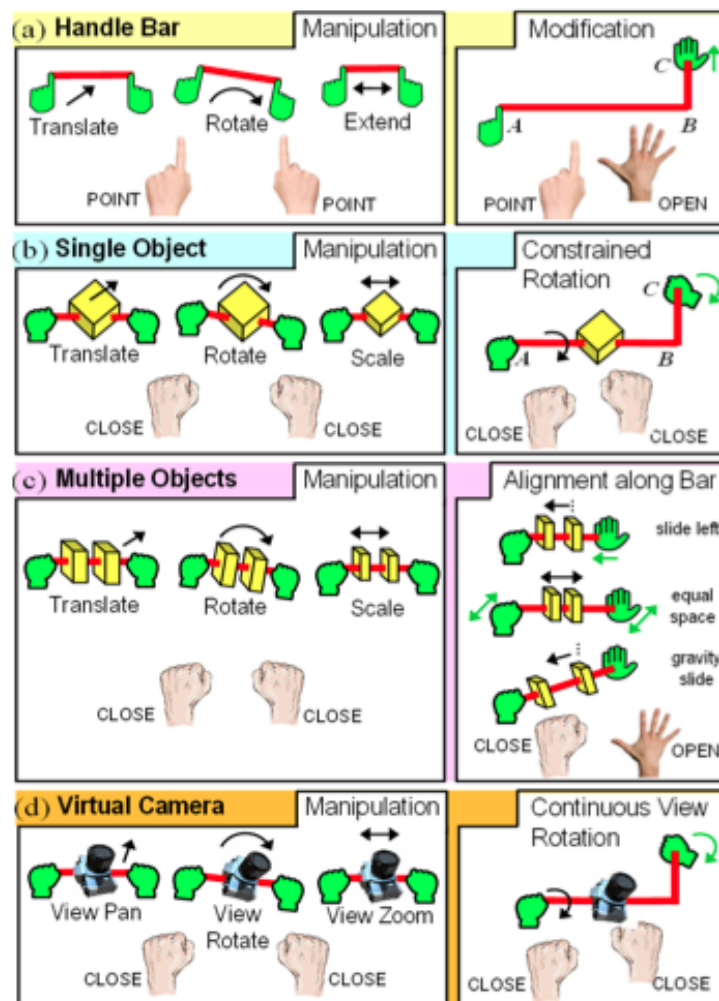
3.1.3 Motion controlled

As the two previous sections had a focus on different ways to create 3D models, this section cover different motion controlled applications. The definition of gestures, actions, and commands are described in 2.1.2. Commands, actions and gestures used to do affine transformation from those applications will be outlined in this section.

(Peng Song, 2012) use the Kinect to track hand movements and hand position. They show how to use a handle bar metaphor to grab objects, rotate them and translate them. Kinect is not able to distinguish many different hand poses, but it can easily distinguish a closed hand and an open hand, and a hand pointing. These commands Kinect is able to easily recognize. A handle bar metaphor is therefore an effective way to perform mid-air interactions that manipulate the

pose and scale of 3D virtual objects². Figure 3.12 lists different operations for manipulating objects.

Figure 3.12: The various operations designed for the manipulation of (a) the virtual handle bar, (b) a single object, (c) multiple objects, and (d) the virtual camera and their associated bimanual hand gestures. The picture is taken from (Peng Song, 2012)



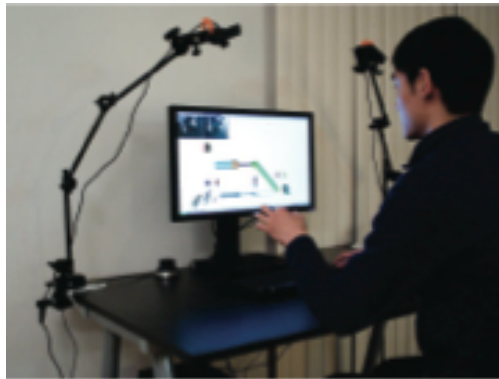
Translation is done by first selecting the object you want to translate. An object is selected by closing both hands when the target object is in the middle of both hands. Translating is then done by moving the hands (without moving the hands together or apart from each other) to where you want to move the object. Rotation is done by selecting an object, and then rotate as if you were holding a bar between your hands. The object will rotate according to the way the

²A movie explaining the system can be found on this URL: <http://www.youtube.com/watch?v=pOEM9Ejv0r0>, accessed February-27-2014.

virtual bar between your hands would rotate. Scaling is done by selecting an object, and then move the hands apart from each other to increase the size of the object, or decrease the size of the object by moving the hands apart from each other. The handle bar metaphor is an effective way to perform mid-air interactions that manipulate the pose and scale of 3D virtual objects. The handle bar metaphor can be implemented using Leap Motion as the input device, but there might be problems with occlusion in cases when the hands overlap each other on the y-axis.

(Robert Y. Wang, 2011) use two cameras to track hand gestures and positions. The cameras are located above the hands as seen on Figure 3.13.

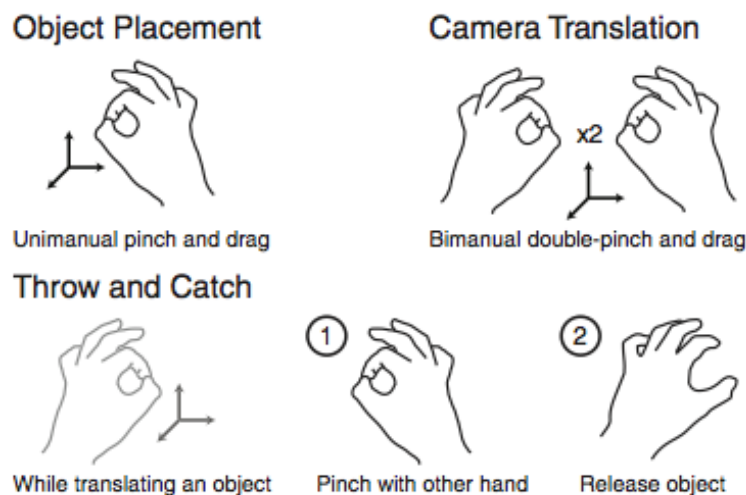
Figure 3.13: Two cameras are used to track the position and gesture of the hands in 6 degrees of freedom. The picture is taken from (Robert Y. Wang, 2011)



They use efficient queries of a precomputed database that relates hand silhouettes to their 3D configuration in order to capture the pose of the hands. Bimanual hand tracking is being used to move objects around in 3D space (See Figure 3.14) to assemble 3D parts together. Their research combines hand postures and gestures to perform actions in 3D virtual space. They use different gestures for rotation and translation because users tend to perform rotation and translation separately according to (Masliah and Milgram, 2000). Some of the gestures they use can be seen in Figure 3.14. A user can pinch with one hand, and the closest object will be selected, and can be dragged. If the object gets close enough to a second object, they will snap together. They also use a technique called “throw and catch” (Kenrick Kin and Agrawala, 2011). A user can select an object by pinching. She can then pinch with the other hand to tell the system that the object should move to that place. The object will be translated from the first hand to the second hand when the use opens the first hand, i.e. stops pinching. This is useful to do long-distance

translations in order to reduce the tiring of the arms. The user can rotate the camera, or objects by pinching with both hands, and then imagine holding a piece of paper. The user can then rotate objects, or the camera the same way as rotating the paper.

Figure 3.14: They support a few simple gestures for 3D manipulation and camera adjustment



The pinching gesture is not possible to perform using the existing Leap Motion API (May 2014), but it may be possible to implement pinching by the application developer.

(Ken Hinckley, 1997) report results from multidimensional input techniques based on magnetic orientation sensors. They let users try different rotation techniques with different input devices. They compare their own “3D Ball” sensor and “Tracker” sensor for rotating virtual objects. They compare these two input methods to the existing Virtul Sphere (M. Chen, 1988), and the Arcball (Shoemake, 1992) techniques.

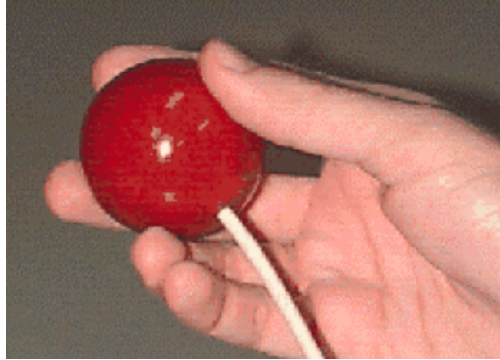
The “Virtual Sphere” is controlled with a 2D mouse. The user can click and drag on a virtual trackball. The virtual object will then rotating according to the drags. Dragging up will rotate about the x-axis. Dragging to the sides will rotate about the y-axis. It is also possible to simulate a drag about the z-axis by dragging outside of the virtual trackball.

The “Arcball” is similar to the “Virtual Sphere”, but it uses quaternion rotations, and does therefore not suffer from gimbal lock.

The “3D Ball” can be seen in Figure 3.15. The ball acts as an absolute rotation controller and consists of a plastic sphere instrumented with a magnetic tracker. The “Tracker” is also a

magnet orientation sensor, but it is not packed inside a plastic ball.

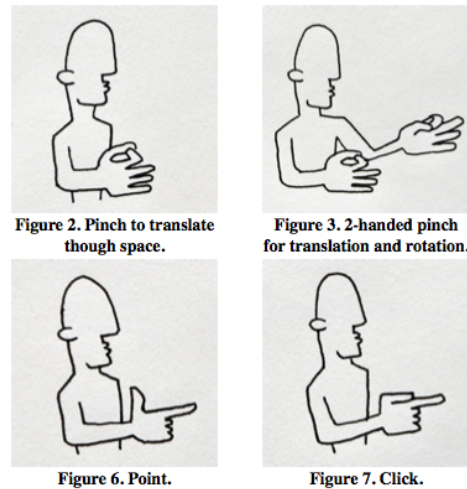
Figure 3.15: Magnetic orientation sensor used to rotate objects in 3D space



Their research found that users were able to do rotations more quickly (without sacrificing accuracy) with the “3D ball” and the “Tracker” than the 2D mouse techniques, despite the fact that users have had years of practice using a mouse. Users did rotations more than six seconds faster in average using the physical props compared to the 2D mouse techniques.

(Zigelbaum et al., 2010) present a gestural interface for interacting with videos. Videos is not the focus in this study, but the research by (Zigelbaum et al., 2010) is interesting because it may enlighten useful information of commands, actions, and gestures that can be used in the software implemented in this study. A pair of gloves is used to register positions, movements, and gestures to execute certain tasks in their interface. They propose many gestures and commands. A few of them can be seen in Figure 3.16.

Figure 3.16: Four samples of different commands from (Zigelbaum et al., 2010). These four commands may be used with Leap Motion



They have 21 commands in total, but after having user tests, they discovered that 21 commands are too many to remember, and many of the commands were difficult to execute. The system was not easy to use because of all the commands to remember.

3.2 Similar applications

There are much research about creating 3D-models as described in the previous section. However, it is also important to cover applications that have already been made. These applications may not be research, but products released for free or commercial use. Leap Motion has its own store (or marketplace), named Airspace, for applications designed to be used with Leap Motion. There were two different applications that focused on creating 3D models found in Airspace. One is named Freeform³ and the other is named Leoply⁴. These applications as well as some other applications not found on the Airspace store will be covered in this section.

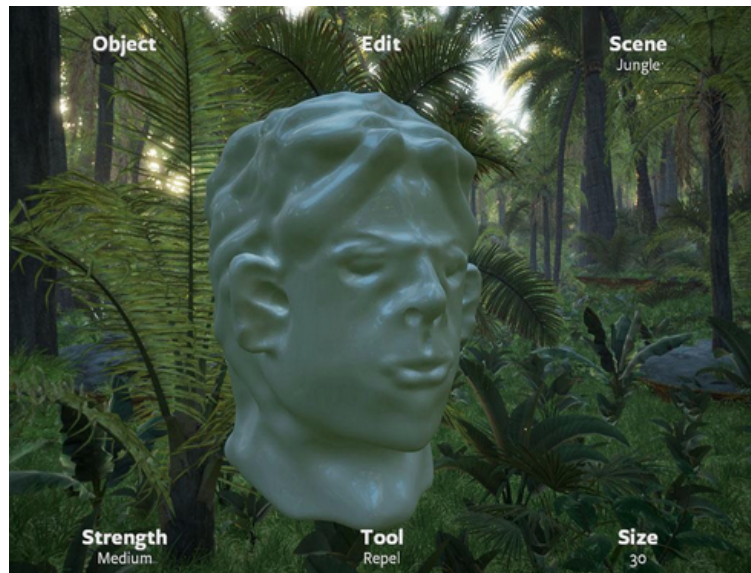
³Freeform can be downloaded from the Airspace on this URL: <https://airspace.leapmotion.com/apps/freeform/>

⁴Leoply can be found on this URL: <http://leoply.com>

3.2.1 Freeform

Freeform focuses on a sculpturing metaphor similar to (Kevin T. McDonnell and Wlodarczyk, 2001). Freeform is an easy to use application where a spinning a mold of clay, glass, or plastic are placed on an invisible turntable. Users can use their finger as a tool to shape the mold. Freeform provides a set of different sculpting tools in different sizes to shape the mold. The user interface is traditional in the way that you need to use your finger as a mouse pointer, and select the desired tool or material to use. A sample screenshot from the application can be found in Figure 3.17.

Figure 3.17: Freeform can be used to sculpt a piece of clay, glass or plastic.



3.2.2 Leopoly

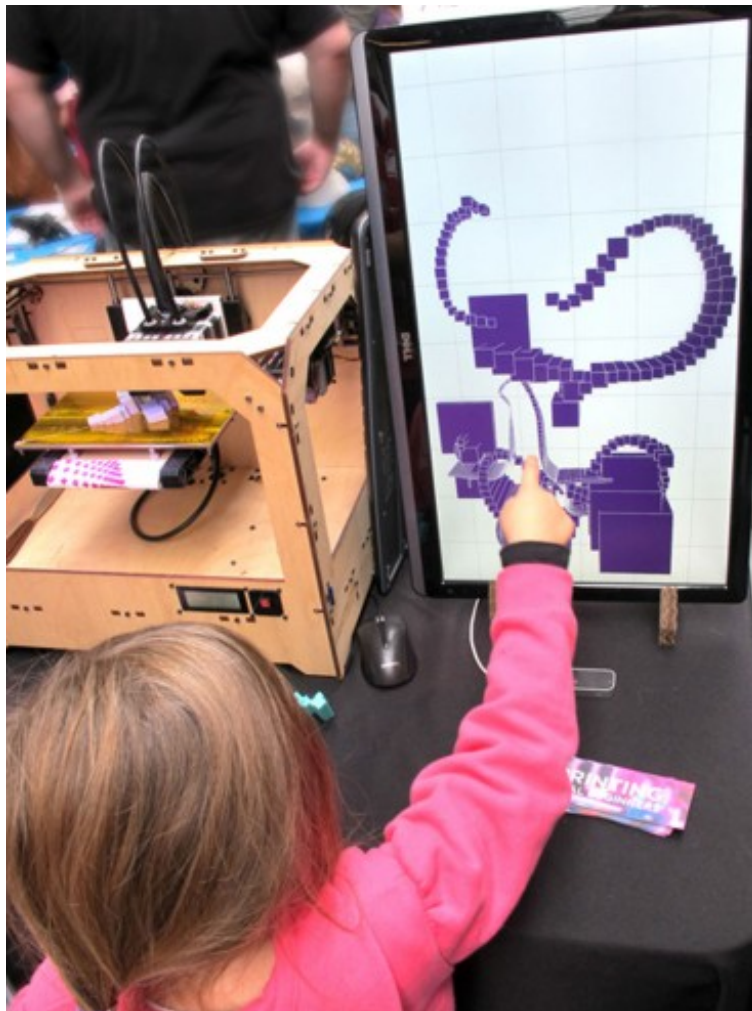
Leopoly is running in the web browser, and does not need Leap Motion to work. The Leap Motion is used as a 3D mouse pointer. The metaphor Leopoly uses is similar to playing with deformable clay. The interface is a conventional graphical user interface, i.e. it is composed of buttons and menus. You are presented with a small piece of clay when you start the application. You can grab portions of that clay by selecting it (either with the mouse or with Leap Motion) and you have to press space to grab the clay. When grabbing a piece of clay, you can deform it and create small canyons or valleys in the clay depending on the direction of your mouse pointer

movement.

3.2.3 Hot Pot Factory

Hot Pot Factory is a company interested in 3D printing. They made an application to create freeform 3D models using Leap Motion, and had their own stand at the Toronto Mini Maker Fair in September 2013, where children and adults could create their own 3D model, and get it printed using a 3D printer. A picture of a child trying the application can be seen in Figure 3.18.

Figure 3.18: Children and adults could enjoy making 3D models at the Toronto Mini Maker Faire and get a physical version printed out using the 3D printer to the left.



They have not released their product on the Airspace store, but they made their own blog

post about the fair⁵. They wanted to create an application making it possible for anyone to make 3D models, without any training at all. The application itself creates voxels based on finger position in 3D space. Figure 3.19 is a model created by a bypasser at the fair. Their application is not concerned about how to rotate and move objects, but it is interesting to see how they are able to let users create creative shapes without any previous experience.

They have also made an application they call the kissing booth. A Kinect is being used to scan a 3D model of people (standing still), and make a 3D figure out of the model using a 3D printer. It is interesting to know if Leap Motion can be used as a 3D printer the same way, but it is not part of this research.

Figure 3.19: model created by a bypasser at the Toronto mini maker fair using Hot Pot Factory's system.



3.2.4 Autodesk Maya

Autodesk is the company that develops the popular 3D modeling application called Maya. They have created a Leap Motion plug-in for their modeling software where modelers can use Leap Motion to edit models in Maya. They can grab, push, pull, and mold their creations into new forms and shapes. It is similar to Freeform as a sculpturing tool, but you are still dependent on keyboard and mouse interactions. The software is suitable for professionals, or semi professionals who are familiar with Maya. Novice users are not able to start using the software without being familiar with sophisticated 3D modeling application. There are video tutorials available

⁵<http://www.hotpopfactory.com/blog/2013/10/15/3d-printer-finger-painting/>, accessed 28-03-2014.

on how to get started using Leap Motion with Maya⁶. Autodesk currently has six videos, each video lasting from one minute to over ten minutes with instructions on how to install the plugin, an overview of the functions available, and how to do sculpting.

3.2.5 Leapouts

Leapouts is a collaborative 3D modeling system to be used together in Google Hangouts controlled with Leap Motion interaction⁷. Leapouts was made as part of the PennApps hackathon February 2014 in Canada. The users are presented with a plane, which can be moved around using the Leap Motion controller. It is possible to use the spacebar to cycle through available 3D shapes, and the key “n” adds them to the scene. They use a 2D cursor to point at objects, and the objects can be selected by extending the thumb (showing two fingers). When an object is selected, it can be scaled and translated in the scene. The object will automatically be updated on the views of all the collaborators when the shape is deselected.

3.3 Technologies

A coverage of the Leap Motion technology and technologies similar to Leap Motion will be given in this section. There are a range of motion controlled devices in the gaming industry. These devices are designed to be used for playing computer games, however the devices may be used in other applications as well. Microsoft’s Kinect is an example of a popular motion controlled device that have been used for other purposes than gaming. As mentioned in Section 3.1, Kinect may be used to create 3D models and to do affine transformations in 3D space.

3.3.1 Leap Motion

Leap Motion was released for commercial use in 2013. The device itself consists of two Infrared (IR) cameras, three red light-emitting diodes (LEDs). The device is an optical tracking system based on stereo vision. Leap Motion is roughly 8cm in width, 3cm in depth and 1cm in height,

⁶Tutorials on how to integrate Leap Motion with Maya: <http://area.autodesk.com/mayaleapplugin>, accessed 27-05-2014

⁷Leap Motion’s blog post of Leapouts: <https://www.leapmotion.com/blog/leapouts-collaborative-3d-modelling-in-google-hangouts/>, accessed 27-05-2014

See Figure 3.20. The device can be placed in any desired location (as long as the USB-cable is long enough). It can be bought for 79.99 USD from the Leap Motion web pages (<https://leapmotion.com>, February 2014). Leap Motion is starting to get integrated into keyboards and laptops⁸. An example image of where Leap Motion will be integrated into keyboards can be seen in Figure 3.21.

Figure 3.20: Rough schematic of the Leap Motion device.

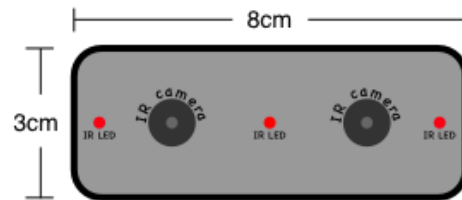


Figure 3.21: Leap Motion is already being integrated into keyboards and laptops.



Leap Motion focus on tracking palms, fingers, and tools (such as a pencil). Microsoft's Kinect is similar to Leap Motion, but Kinect focus on tracking people and positions. The Leap Motion features a much higher precision than the Kinect and other motion tracing devices (Weichert et al., 2013) in the same price range.

The red LEDs on the Leap Motion creates a 3D field of dots of infrared light⁹. The two IR cameras generate 290 frames per second of reflected data, which is sent through the USB cable from Leap Motion to the computer, where the Leap Motion software analyzes the data. However lea (2014) reports a frame rate of 290, other sources indicate a lower frame rate of around 150 frames per second (Vikram et al., 2013).

⁸a blog post written on Leap Motion's web pages explains how Leap Motion will be integrated into laptops and keyboards: <http://blog.leapmotion.com/post/69095530774/hp-expands-leap-motion-integration-into-broad-range-of>, accessed 28-03-2014

⁹a video on YouTube illustrates this: http://www.youtube.com/watch?v=UI5EBzU_QqM, accessed 2014-02-28

Little is known about the algorithm used by Leap Motion. Non-official sources on the Leap Motion forums claim that they use “complex math” in a way that has not yet been disclosed by the company, in some way synthesizing 3D position data by comparing the 2D frames generated by the two cameras¹⁰

The Leap device detects and tracks palms, fingers and tools within its field of view. It classifies finger-like objects according to shape. A tool is longer, thinner, and straighter than a finger.

For pointables (tools and fingers), the following characteristics can be obtained from¹¹ through the Leap Motion API.

Research about Leap Motion

Although Leap Motion is still a very new device, some research about it has been published.

Weichert et al. (2013) analyzed the accuracy of the Leap Motion, which by Leap Motion is claimed to be less than 0.01 mm. This is an article published in *Sensors*, a journal which is one of the leading international, peer-reviewed, open access journals on the science and technology of sensors and biosensors (mdp, 2014).

Bökesoy (2013) has explored how Leap Motion can be used as an instrument (controlling a synthesizer). He is a researcher in the field of computer music. He has published the article in a journal in Paris (Journées Informatique Musicale), and has a Ph.D. in Computer Music & Composition at University of Paris.

Vikram et al. (2013) used Leap Motion to recognize handwriting. The focus of the article was on how to recognize handwriting in a continuous stream of 3D coordinates without being able to perform pen-up and pen-down events. The article gives interesting observations about how to practically use Leap Motion as an input device. The article was published by ACM (<http://dl.acm.org/>) in a book named “CHI’13 Extended Abstracts on Human Factors in Computing Systems”.

¹⁰Sources can be found at Leap Motion’s forum <https://forums.leapmotion.com/forum/general-discussion/general-discussion-forum/1058-technical-specifications?1005-Technical-specifications=&p=8906&viewfull=1#post8906> and <https://forums.leapmotion.com/forum/general-discussion/general-discussion-forum/434-the-unofficial-leap-faq?420-The-unofficial-Leap-FAQ>, accessed 27-05-2014.

¹¹ Leap API overview can be found at https://developer.leapmotion.com/documentation/Languages/CSharpandUnity/Guides/Leap_Overview.html, accessed 27-05-2014.

Rivera et al. (2013) has done preliminary work on obtaining the phalangeal joint angle to be able to create realistic finger animations using Leap Motion. This started as a research proposal for University of Pennsylvania, and got accepted February 13. “Project PAALM” is a framework for estimating finger joint angles and animating hand models in Autodesk Model using the Leap Motion Controller. They held a presentation at the University of Maryland, Baltimore County, 21st Annual McNair Research Conference, where they received astounding feedback. More information about the project can be found at their blogpost (projectpaalm.blogspot.com).

Latency

The latency of Leap Motion is low compared to other similar devices. It usually takes less than 12 ms from the user moves her hand, until the data arrive in the application (if the device is connected with USB 3.0).¹² The application and the display will be the bottleneck in terms of responsiveness. By comparison, the Wii Remote usually has latencies in the range of 100 ms to 143 ms (Pavlovych and Stuerzlinger, 2009). The mouse has latencies between 8 ms to 20 ms, and the Kinect has latencies between 106 ms to 243 ms (if it is only one player playing a game) (Livingston et al., 2012). The PlayStation Move has an average latency of 115 ms (Tanaka et al., 2012).

Having low latency is important for users if they want to feel inside a virtual world (immersion). From Swink (2009):

The computer’s half of the process must take less than correction cycle for the player. At 50 ms, response feels instantaneous. Above 100 ms, the lag is noticeable, but ignorable. At 200 ms, the response feels sluggish.

Bad readings

Leap Motion has significant issues with noise in some conditions:

From¹³:

¹²Source can be found on Leap Motion’s website at <http://labs.leapmotion.com/post/55354873140/understanding-latency-part-2>, accessed 2014-04-09

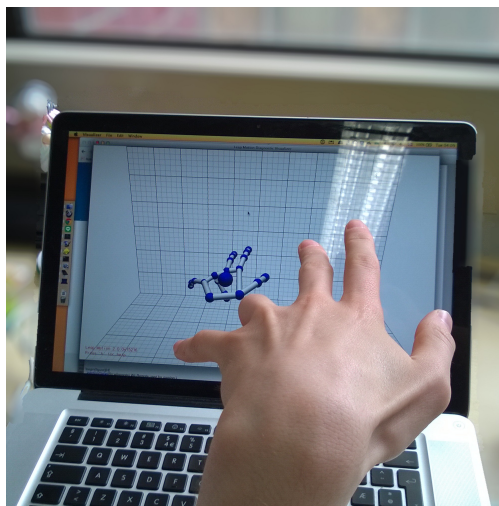
¹³The quote can be found on Leap Motion’s website <https://support.leapmotion.com/entries/39187857-Using-the-Leap-Motion-Diagnostics-Test>, accessed 2014-04-09

The Leap Motion Controller achieves its best performance in an environment without any external infrared light sources. [...] Note that the Leap Motion device will still work adequately under most poor lighting conditions. However, the tracking smoothness, range, and accuracy may suffer.

Burton Jr (2013) also found “occasional bad readings” and deemed it necessary to smooth the data in order to use it. Such smoothing introduces latency and might be a problem for some applications.

If fingers or hands move out of the Leap Motion’s vision, they will be assigned a new ID. The application programmer then has to decide whether it is still the same finger or hand. This also happens when one finger is hidden behind another finger. The developers of Leap Motion continuously publish new updates for their device. Version 2.0 (the newest version, as of May 2013) of the Leap Motion API introduces a new skeletal tracking model. Featuring additional information about hands and fingers. The model also includes more robust hand tracking that continues to provide data even when a hand is partially occluded. This update seem to have fixed the issues with distinguishing fingers. Putting two fingers together will also be recognized as two fingers instead of one. A skeletal model of the hand from Leap Motion’s diagnostic visualizer can be seen in Figure 3.22.

Figure 3.22: The diagnostic visualizer displays some impressive results when it comes to detecting hands and fingers.



3.3.2 Microsoft

Microsoft has made a success with their motion controlled device, Kinect ([kin, 2014](#)), and can be bought for approximately 100 USD (February 2014). It was for the first time announced in 2009 under the Electronic Entertainment Expo (E3) conference. Kinect uses an infrared projector, a camera, and a special microchip to track the movement of objects and individuals in three dimensions. The infrared project creates depth images by emitting a 3D pattern of infrared dots, almost the same way as Leap Motion. The Kinect creates a skeletal model of the human body which it stores in a database. The Kinect converts the depth image into possible recognized body parts, and from those body parts it creates joints for the skeleton ([Madden, 2012](#)).

There are no buttons on the Kinect, and you only use your body, hands, and feet to communicate with the console. The technology is most similar to Leap Motion compared to the other motion tracking devices mentioned later in this chapter.

Figure 3.23: Kinect is an motion controlled optic device, designed primarily for gaming.



3.3.3 Nintendo

Nintendo has marked itself in the gaming industry for many years. They released their newest gaming console: Wii U ([nin, 2014](#)) in January 2012. Nintendo has a special motion controller called Wii Remote (see [Figure 3.24](#)) with a built in accelerometer. The newest model of Wii Remote can be bought for 40 USD (February 2014). The controller can be connected to Nintendo's Wii or Wii U using Bluetooth technology. The Wii Remote requires an infrared sensor (placed below or above the television) in order to transfer position and motion data to the console. The Wii Remote has been further upgraded to support even more accurate motion detection since their first version.

Figure 3.24: Wii Remote used for gaming.



3.3.4 Sony

Sony was the last game console company that incorporated motion controllers to their console, PlayStation (pla, 2014). PlayStation Move is Sony's motion controller and can be connected to PlayStation 3 or PlayStation 4 by means of Bluetooth technology. A glowing ball at the end of the controller is used to register movements and position. The controller requires a camera to register where the glowing ball is located. The controller uses accelerometers and gyroscopes combined with the camera and a filtering algorithm that lets the PlayStation determine the angle, position, and any motions done with the controller.

Figure 3.25: PlayStation Move for playing games with motion controllers.



It is possible to buy a bundle with a motion controller, camera, and a game for 40USD (February 2014), which is cheaper than buying the controller and camera separated.

3.3.5 Sixense

Sixense is a company that have made an application called MakeVR, using their wireless motion tracking system called STEM System as seen in Figure 3.26. The STEM System uses STEMs for

tracking the full orientation and position of a user's hands, head, and body. The STEMs can be positioned to the parts you want to track, and it does not have to be on the user's body. Their technology uses an alternate current (A/C) electromagnetic field to determine the position and orientation of each STEM, which means that you do not need to have line of sight between the STEM trackers to determine the position or orientation of them.

Figure 3.26: The STEM System consists of two motion controllers, wireless motion trackers (STEMs), and a stationary reference for each STEM.



According to Sixense webpage¹⁴: “MakeVR is a 3D modeling application built around two key components: an innovative 3D multi-touch interface that replaces the traditional mouse and keyboard inputs for all functions and a professional-grade CAD engine”. Unlike Leap Motion, it requires the user to hold a set of motion controllers in their hands. In addition, MakeVR is intended to be used with a binocular head mounted display (HMD) so the user is able to perceive the scene depth in a natural way. In MakeVR, it is possible to translate, scale, and rotate the environment similar to how to use a 2D multi-touch device, such as a smartphone, or a tablet. The following lists describes some of the ways to manipulate the 3D environment in MakeVR:

- Translate - It is possible to translate the environment by reaching out and grab a point in space and use it to drag (translate) the world.
- Scale - If a user grab two points, she can move the hands closer to to scale the environment up, or move the hands apart to scale the world down.

¹⁴official site to Sixense: <http://sixense.com/sixensestudios/makevr>, accessed 19 May 2014.

- Rotate - A user can rotate the world by grabbing two points in space, and rotating the hands. It is possible to yaw, pitch and/or roll the world.

Consumers may buy a STEM System bundle, including two motion controllers, and a STEM base, but no extra STEMs for 299.99 USD. Each extra STEM costs 99.99 USD (May 2014).

Chapter 4

Development process

In the previous chapters we have established that it is interesting to see if Leap Motion can be used as an interaction device for creating 3D models, with a focus on homogeneous transforms to make it easier for beginners to create 3D models. We have found relevant theory and practices for 3D modeling applications, and explained the difference in technologies similar to Leap Motion. In this chapter, an outline of how the project was conducted, tools and software used, and architecture of the application will be given. The 4+1 architectural view model was used to provide a clear overview of the software (Krutchen, 1995). The model is divided into five views, where each view will look at the system from different perspectives. The different views provide a description of the system from the viewpoint of different stakeholders. The five views consist of a logical view (see Section 4.2.3), a process view (see Section 4.2.4), development view (see Section 4.2.1), a physical view (see Section 4.2.2, and a scenario view (see Section 4.2.5).

a SWOT analysis to evaluate the **S**trengths, **W**eaknesses, **O**pportunities, and **T**hreats in the project will be introduced in Section 4.1.4, which also is a part of identifying the practical issues according to the DECIDE framework described in Chapter 2.

4.1 Planning and work process

An outline of how the project was planned will be given in this section. Inspiration from both Kanban and Scrum was implemented as the development model in this project. The basic ideas will be outlined in this section.

4.1.1 development models

A brief introduction of the waterfall model, the scrum model, and the kanban model will be presented in this section following an implementation of the development model used in this project.

Waterfall

The waterfall model is a model which assumes that every step in the model is done before moving on to the next step in the development process (Sommerville, 2010). All the requirements have to be specified before going on to the next step, which is design and software architecture. In order to proceed to the implementation step, everything in the software architecture step need to be done. The same goes for the rest of the steps in the model too, which are integration, testing, installation, and finally maintenance. The waterfall model is mostly used in pedagogic education, as it is proved not to work well in practice, but the theory in each step can easily be understood step by step.

Kanban

Kanban is an agile development model focusing on not overloading teams, which may cause thrashing, creating waste, lower quality, and harm capacity (Shalloway, 2011)(Sommerville, 2010). The work tasks in Kanban is visible to all team members at all times. The word kanban is Japanese, and means sign, or sign post. The term Kanban is most famous from the car manufacturer Toyota, where they focused on a lean development of their cars, reducing waste.

Scrum

Scrum is an incremental, iterative, and agile development model. It focuses on having sprints which usually lasts from one week to two weeks (Sims, 2013)(Sommerville, 2010). The team will focus on working for short periods at a time. Spending time for reviewing and reflecting on the tasks done within the sprints are important to ensure that the product do not deviate from initial plan, as well as having continuous improvements on the software. The software will undergo changes after each sprint, and a working prototype will always be available for testing.

The model suits teams of different number of members, but usually suits team sizes from five to nine people. The Scrum model usually have three distinct roles: product owner, scrum master, and team members.

The product manager ensures that the team members understand the requirements, implementing the correct functionality, not creating waste. She also controls the product backlog, which is the tasks and requirements for the product. Requirements are often made in the form of user stories (e.g. “As a <role>, I want <a feature>, so that I can <accomplish something>”). The product owner represents the customers, and she is the only one that can prioritize the the items in the product backlog, and ask the team to do work.

The scrum master acts as the coach, guiding the team to perform better, and to deliver a good product. She helps the team to learn and apply scrum, and related agile practices to the team’s best advantage. Even though this role sounds like the boss of the team, it is not the case. The scrum master is not ranked higher than the other members, but acts as the facilitator for the team.

The team members are self-organized, they decide themselves how the tasks should be solved, and they decide amongst themselves which person should be responsible for a given task from the backlog. Tools and techniques are chosen based on the team members preferences.

Implemented development model

Since it is only one person developing the software for this project with the help of two supervisors, tasks were created each week as seen necessary. Each Friday, the work done would be presented to the professor in Japan together with the other master and doctor students. The supervisor and the other students would come with suggestions on what to improve, and how to proceed further with the project. A meeting over Skype ([skype.com](https://www.skype.com)) with the professor in Norway would take place afterwards. A video meeting was not done every Friday, but messages were exchanged about what have been done, and what is going to be done the next week.

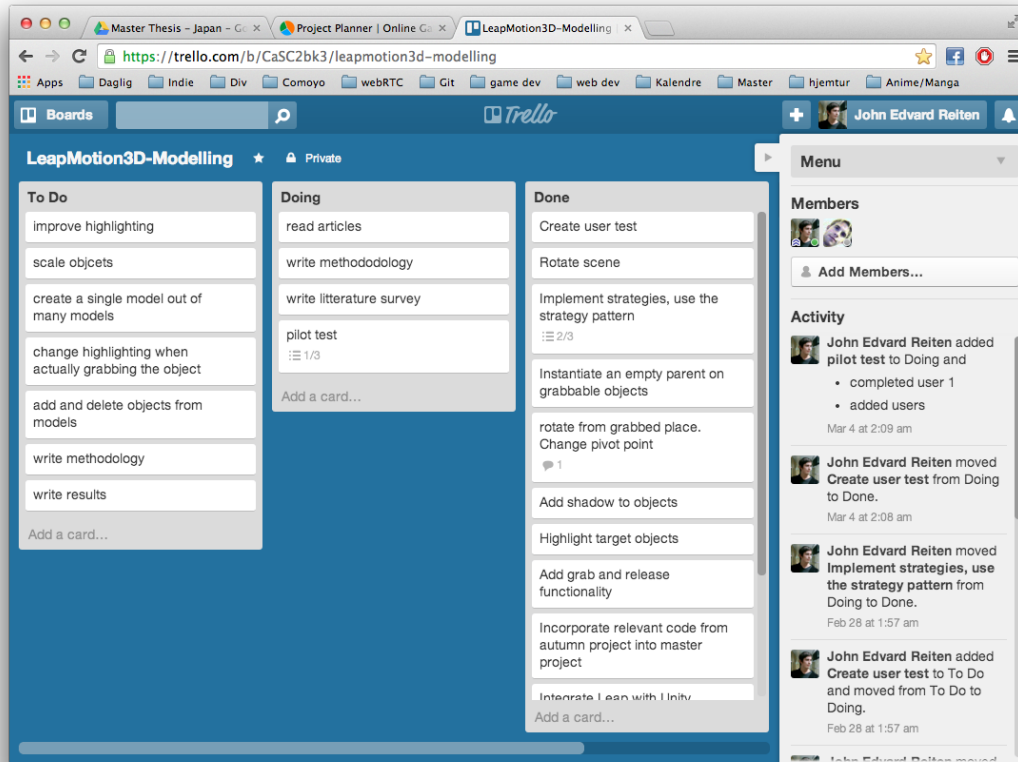
Trello (trello.com) was the software tool used to manage the working tasks. A description of how Trello was used will be given in Section 4.1.2.

4.1.2 Trello

Trello is an application well suited for development models such as Scrum and Kanban. The application uses a post-it metaphor. You can create post-it notes with a description of a task, and move the post-it note in different columns. The columns created can be categorized into desired categories. For this project, the category “To do”, “Doing”, and “Done” were made. A picture of the application can be seen in Figure 4.1. Tasks were created as seen necessary. The “To do” category contains tasks that are suggested by the supervisors or the student. Tasks that are in the “To do” column are free to be assigned to anyone. When the task is assigned to someone, the post-it note should be transferred to the “Doing” category. When the task is completed, the task should be transferred from “Doing” to “Done”. A task is done when the developer is able to test the functionality, and it is working properly. There might be cases when there are bugs in the completed task. A new task would then be made, which is to fix the bug. This task might have a higher priority than other tasks in the “Doing” category. In case there are more important tasks to do, it is possible to move a task from “Doing” to “To do”, thus replacing them with more important tasks.

Even though this project was carried out by one student, it is useful for the supervisors to understand which tasks that are being worked on at all times.

Figure 4.1: Development tool for creating tasks, and assign them to members.



4.1.3 Tom's planner

Tom's planner is an online tool to create Gantt diagrams. Other options such as using a spreadsheet was considered, but spreadsheets are not so easy to modify as the online tools available. There were other online tools available as well, but most of them cost money. Tom's planner was a good alternative to the planner applications available, and was used in in this project. A Gantt diagram is useful to understand how well the project is going. The diagram is used to get an overview of the project's status and can be found in Appendix B. The working schedule together with the post-it board (using Trello) are good tools to understand how the process of the project is going.

4.1.4 Risk Management

This section contains the risk analysis for the project, and a SWOT analysis was chosen as the preferred model. The main purpose of a SWOT analysis is to get an overview of the strengths, weaknesses, opportunities, and threats within an organization. The strengths and weaknesses are internal factors, while opportunities and threats are external factors. See Figure 4.2 for an overview of the SWOT matrix. A SWOT analysis is a good framework to identify the areas that can be improved, as well as making decisions in an organization (Hill and Westbrook, 1997). The internal factors are meant to strengthen the positive sides in the organization as well as improve, or at least be aware of the weaknesses. The external factors are meant to discover bonds that need to be maintained as well as discover potential challenges that may occur in the future.

Figure 4.2: Strengths, weaknesses, opportunities, and threats (SWOT) matrix.

Internal	Strengths	Weaknesses
External	Opportunities	Threats

Strengths

The strengths in this project are several. There are two supervisors to help with the study. The professor in Japan is very familiar with computer graphics, and may be of great help in complicated implementation details. The associate professor in Norway is an expert in natural language, which is not a focus in this study, but he will be most helpful with the report. The profes-

sor in Japan has published several articles about human-computer interactions, which makes him very experienced in the field, and can help with the report as well.

The project will be implemented using Unity, which the student is already familiar with from previous projects. More focus on the actual implementation can take place instead of learning a new language and a new development tool.

Weekly meetings with other students as well as the professors open up for discussion and good advice from both professors and students. Other students at the computer lab have experience with Unity as well, which means that it is possible to get help for Unity related problems.

Weaknesses

The greatest weakness may be language barriers. Most of the habitants of Japan do not speak English very well. Fortunately, the students at computer science are able to speak English very well. Cultural differences may also be an issue. It might be easy to misunderstand each other, and information might get lost in translation. A good rule of thumb is to never assume anything, but ask and discuss until both parts are able to understand each other. Since most people in Japan do not speak English, it might be a problem to find a sufficient amount of test users. The students in the computer lab are good candidates, as well as people who are taking the same Japanese classes as the student.

Opportunities

As the study will be done in both Norway and Japan, test users from different disciplines may have a positive contribution to the user tests. The student is living with a host family in Japan, which have many guests visiting from all over the world. These guests might be suitable candidates if they want to and have time to help with the study in this project.

Threats

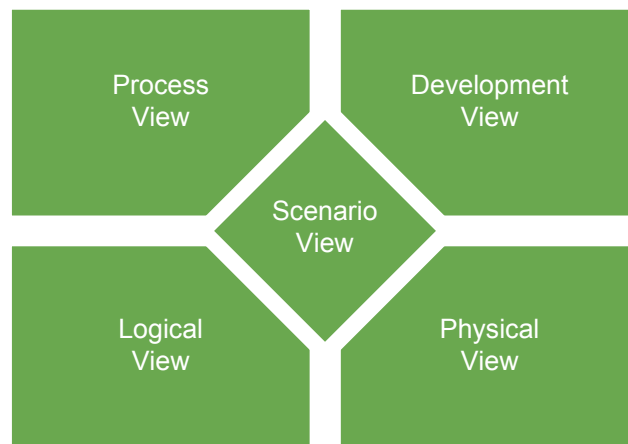
Communicating with the supervisor in Norway might be a problem due to the eight hour time difference. A scheduled meeting every Friday will take place, but other than that, communication might get delayed. The school schedule is different in Japan and Norway. The school holiday in Japan is from end of February until the start of April, and the easter holiday in Norway

is in the middle of April. Japanese classes are held at the university of Tokyo, and these classes might take extra time, leaving less time for the main project. The student will have to contribute some extra working hours in the weekends to compensate for the time spent learning Japanese.

4.2 Architecture

The architecture described in this section will be described in a similar way to the 4+1 Architectural View Model, made by Philippe Kruchten ([Kruchten, 1995](#)). The 4+1 View Model is meant to describe the system from the viewpoint of different stakeholders, such as end-users, developers and project managers. The four views of the model are: logical view, development view, process view and physical view. The +1 View is the scenario view. The different views can be seen in [Figure 4.3](#).

Figure 4.3: 4+1 View model.



All the views will play a role in this project, and each subsection of this section will describe how the different views act as blueprints for the implemented software in this project. Section [4.2.1](#) gives an overview of the system components in this project. Section [4.2.2](#) contains an overview of the physical parts involved in the project. Section [4.2.3](#) contains a logical view of the software made in this project. Section [4.2.4](#) contains a process view of how the user can change between different modes and strategies in the application. Section [4.2.5](#) will contain information of what users are able to do with the system. The scenario view will also have an outline the different actions, commands, and gestures a user is able to do in the application. The final

section 4.2.6 contains information about the different rotation strategies implemented in this project.

4.2.1 Development view

The development view gives an overview of the system components. It is also concerned with the software packages involved in the system. Usually, separate teams of software engineers can work on different packages on the system, and later glue the packages together. As an example, one team can work on the user interface, while another team can work on the networking.

From Figure 4.4, we can notice that there are three main aspects to the software made in this project. the Leap Application, which is the code written in this project, and that code is running on the Unity game engine's software. Leap Motion can be integrated into Unity, and the Leap Shared Libraries can be used at the highest level of abstraction through Unity. A thorough explanation on how Leap Motion can be integrated in Unity can be found at the web pages of Leap Motion (<https://leapmotion.com>). Data from the Leap Motion device are being sent to Unity, and it is up to the software developers to interpret the data from Leap Motion using the Leap Shared Libraries. Since the coordinate system in Unity is left handed, and the coordinate system in the data received from Leap Motion is right handed, we have to invert the z-axis when passing the data to the Leap Application.

The different software components in this project are slightly cohesive, making it difficult to create separate packages for the different parts of the application. It is however possible to separate the the packages according to Figure 4.6, which will be described in Section 4.2.3. Since this project was done by one student, the need for separating the project into different packages was not strictly necessary, but is a good idea to do so.

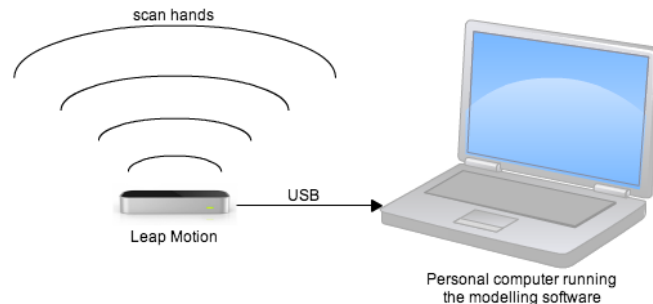
Figure 4.4: The main parts of the software can be hierarchically arranged, where the top is the highest level of abstraction, and the bottom is the lowest level of abstraction.



4.2.2 Physical view

This section contains a physical view of the software. The figure describes where the different packages from Figure 4.4 are physically installed. The Leap Shared Libraries are provided by the developers of Leap Motion, and are installed on the computer. Unity is the game engine used in this project, and it is also installed on the computer. The code written in this project are integrated with Unity's game engine, and also run on the computer. The Leap Motion sensor is reading raw data from what it is scanning, sending it to the computer, and the computations/interpretation of the data are done on the computer. See Figure 4.5 for a reference of which components that are needed in order to use the application.

Figure 4.5: Leap Motion will scan for hands, and send all the data to the computer using USB. The computer has Unity installed, the shared libraries provided by the developers of Leap Motion is installed, and the software written in this project is also installed on the computer.



4.2.3 Logical view

A class diagram of the system will be presented in this section, and can be seen in Figure 4.6. The class diagram is composed of four different (blue) packages. They are separated into different packages to distinguish what has been made in this project, and what has been made by others. The packages also act as an overview of how the components in the software can be distinguished from each other, making it possible to work on the different packages at the same time. The blue Leap C# Interface package contains boiler plate code from example code provided by the developers of Leap Motion. The code will dispatch lost-, found-, and updated events. The data are high level raw data that the Leap Motion device is reading. As an example, if a hand is detected in Leap Motion's field of view, the Leap C# Interface will dispatch a HandFound event, which is picked up by the LeapInputAdapter (which can be seen inside the LeapGame package).

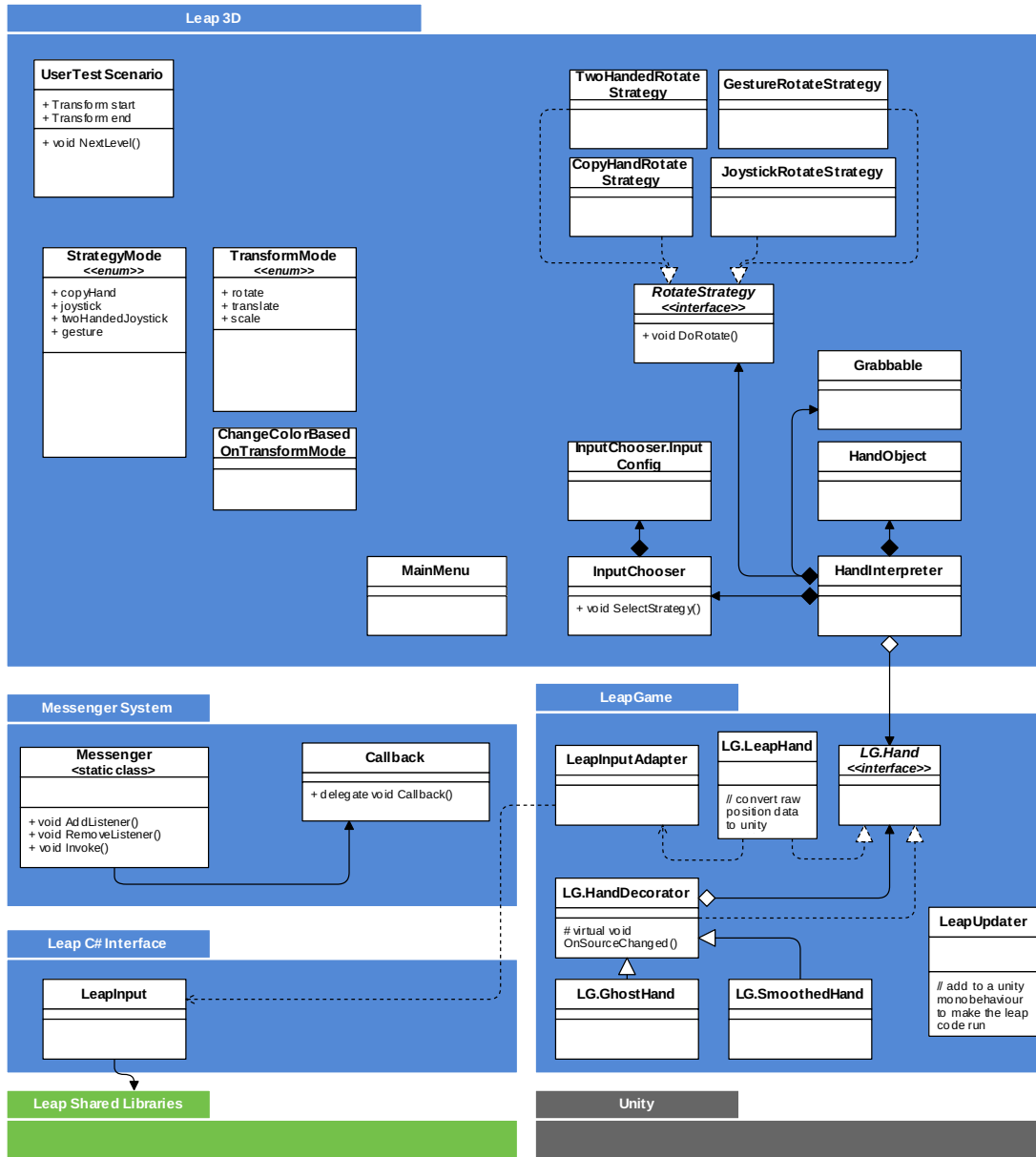
The LeapGame package is code written in a previous project ([Johan Klokhammer Helsing, 2013](#)), where Leap Motion was used as a device for controlling an avatar in a first person perspective game. LeapInputAdapter contains code to determine which hand is which. That is, the code is responsible to determine if it is the left hand or the right hand that is being detected.

Leap Motion has some problems with noise and bad readings. Creating own abstractions for hands is therefore necessary to filter out the bad readings. LeapHand is the interface that is implemented by the abstract class LG.HandDecorator. LG.HandDecorator is a class used to mix different types of LG.Hands, instead of making separate classes for specific LG.Hands. The decorator pattern was used for this purpose ([Bass, 2013](#)). LG.SmoothedHand and LG.GhostHand inherit LG.HandDecorator. It is therefore possible to create a mix between those two LG.Hand classes, creating a "smoothed ghost hand". The intention was to be able to create different types of LG.Hands, and the decorator pattern is therefore very useful.

LG.SmoothedHand can take an LG.LeapHand as an argument when created, keeping the properties of a LG.Hand as well as modifying the specific properties of an LG.SmoothedHand (which is to take the moving average ([Oppenheim et al., 1999](#)) of the hand's position). The LG.GhostHand was made to prevent the hand from disappearing from Leap Motion in case some bad readings from Leap Motion occurred, saying that the hand was lost. The LG.GhostHand will assume that a hand cannot be lost before some amount of time has passed. The amount of

time needed to pass before we can say for sure that the hand is lost is editable through Unity's inspector.

Figure 4.6: The logical view for the software made in this project. Logic is split into different packages, and the relation between the classes are represented with arrows.



4.2.4 Process view

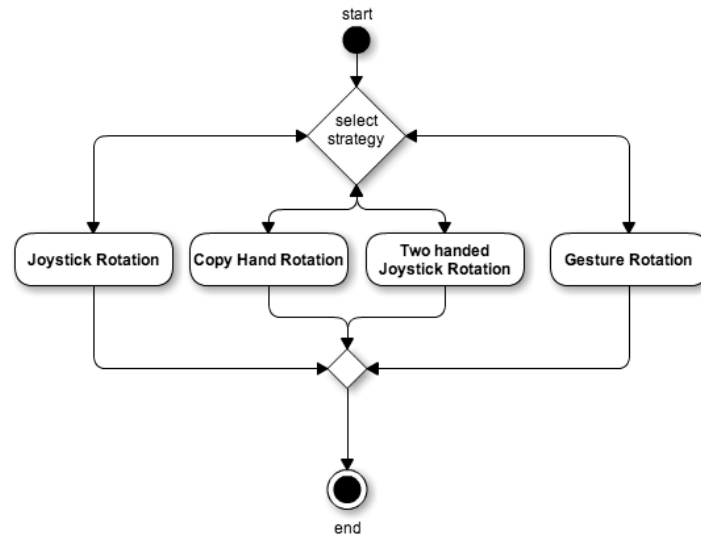
This section gives an overview of the most important processes in the application. There are four different figures explaining different processes. Figure 4.7 and Figure 4.8 focus on user interactions, while Figure 4.9 focuses on the sequence in which the data are received from Leap Motion, and processed in the software made in this project. Figure 4.10 illustrates what happens in the main loop of the software.

Two different types of diagrams have been chosen for the process view. One is a flow chart diagram, and the other is a sequence diagram. The sequence diagram is used primarily to show the interactions between objects in the sequential order that those interactions occur. The flow diagram is used to show how a certain part of the software can be in different states, and how and what is needed for the software to change states. A flow diagram is also used to visualize what is going on in the software to help users to understand the process.

Changing strategies

As seen on Figure 4.7, there are four different rotation strategies implemented in this system. They are called “Copy Hand Rotation” strategy, “Joystick Rotation” strategy, “Gesture Rotation” strategy, and “Two Handed Rotation” strategy. These are the four strategies a user can use to rotate models. The strategies can be changed using Unity’s inspector, or at runtime by the user. These strategies are the proposed strategies for evaluation. A detailed description of how the different strategies work can be found in Section 4.2.6.

Figure 4.7: Different strategies for rotating objects in the application can be used. The strategies can be changed at run time, or using Unity’s inspector.



Changing modes

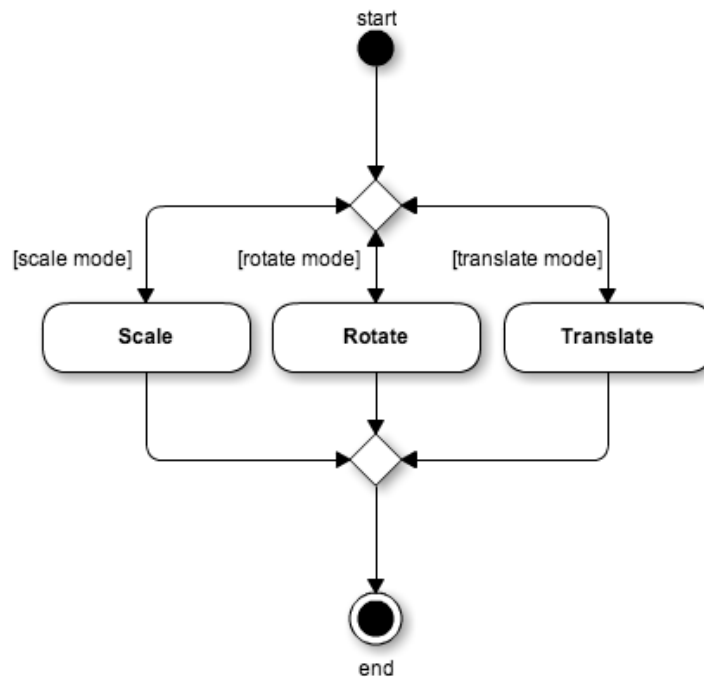
The application offers to operate in different modes. Modes in this project means that the software is able to perform the different homogeneous transforms separately, offering a more precise workflow (see Figure 4.8). It will be possible to move objects independently, or translate objects independently, or scale objects independently when operating in the different modes. The different modes are rotate, translate, and scale. These modes can be changed by the user, and two different approaches to change a mode have been suggested:

1. A user can tap with a finger to switch between modes (see Figure 4.13 for the definition of a finger tap gesture)
2. A user can display a certain number of fingers with the secondary hand, where the number of fingers specify the mode to use.

The tapping gesture proved to be too unreliable when it was prototyped. Tapping gestures were not recognized good enough, and sometimes they were recognized even though no tapping gesture had been performed. The same behavior occurred even though the secondary

hand was used to change modes. Displaying a specific amount of fingers with the secondary hand worked better.

Figure 4.8: The software may run in different working modes, the user can choose which mode to use when working with the application. The different modes are: rotate, translate, and scale.

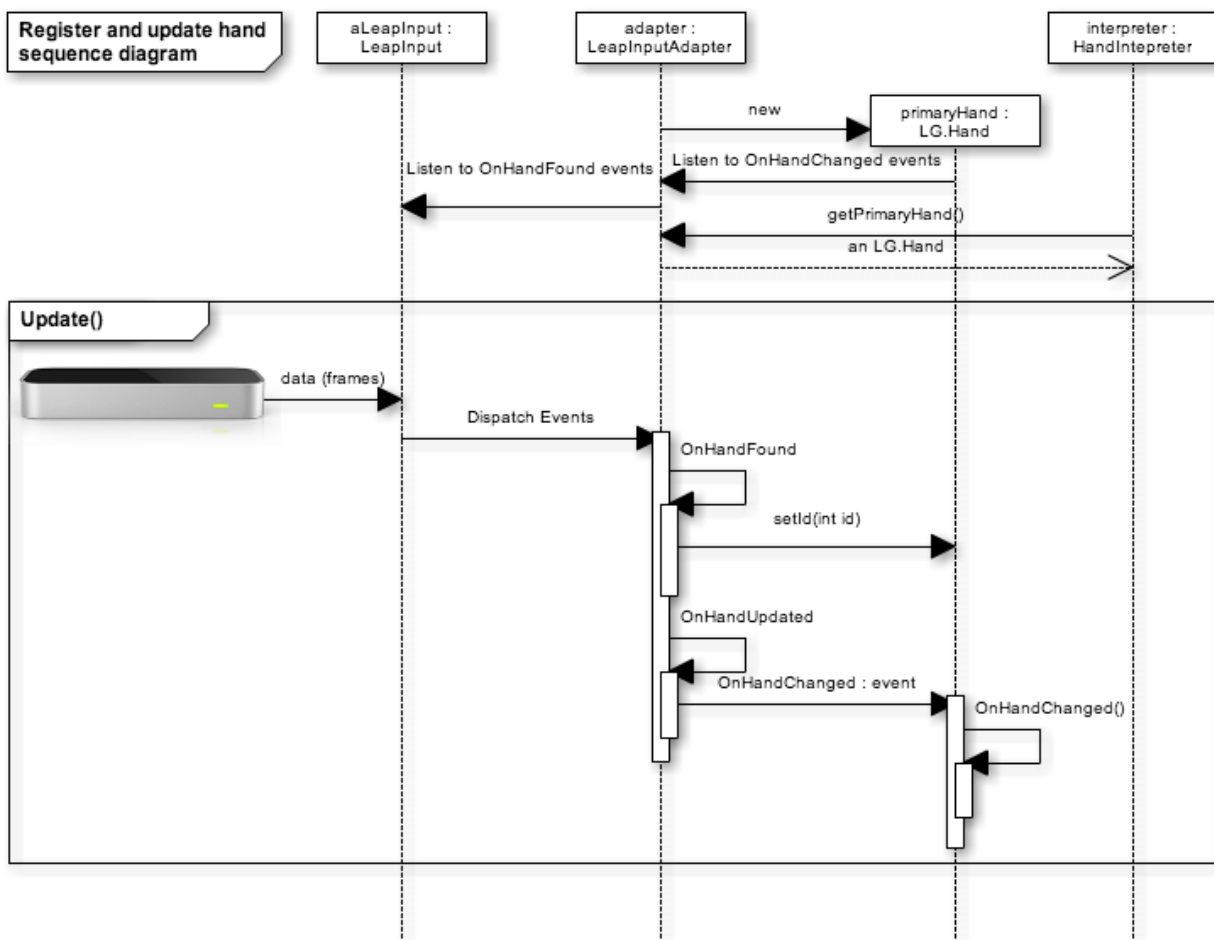


Interpreting data from Leap Motion

For the application to work, it must interpret the data Leap Motion sends to the computer. Figure 4.9 illustrate the sequence on how data from Leap Motion are interpreted. The data are passed from Leap Motion as Frames to Unity. These Frames are then interpreted by the class LeapInput, which dispatch events to all other classes registered as listeners to those events. The LeapInputAdapter is the class that listens to events from LeapInput, and is also responsible for updating the position and orientation of the primaryHand and the secondaryHand (the secondaryHand is omitted from the sequence diagram to make the diagram more comprehensible). LeapInputAdapter is the class that creates a new LG.Hand object, and it is up to the HandInterpreter to get the object. The HandInterpreter will then use the parent LG.Hand object as a reference when creating a GhostHand, SmoothedHand, or a mix of both of them (not

shown in the diagram). In the update cycle within Unity, Leap Motion will keep sending frames to LeapInput as often as it can. It is up to the LeapInput to interpret the frames, and dispatch events. LeapInputAdapter receives events, and will manage the different LG.Hand objects (giving them the correct id, and/or tell them that they need to update themselves with new properties such as position and orientation). The HandInterpreter is only interested in the primaryHand and the secondaryHand, but it does not care about how they are found.

Figure 4.9: The sequence in which a hand is registered and updated in the system.

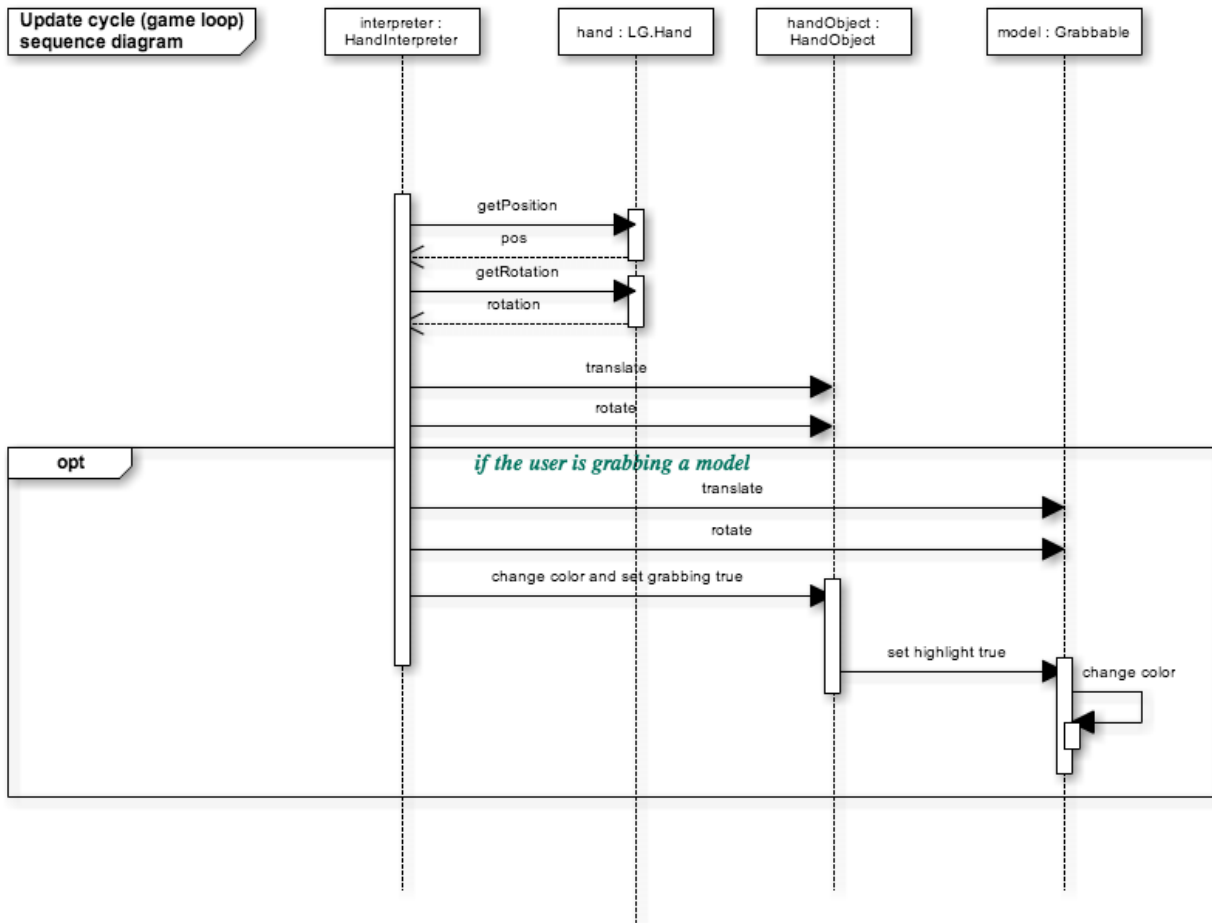


How to transform objects

The most important sequence in this project, might be how the models are actually translated, rotated and scaled (see Figure 4.10). The HandInterpreter is the class that manages the main

update cycle for the application. It will get the position from where the hands are registered by Leap Motion and transform the positions to fit within the scene in Unity. It will then use the orientation and position of the hands to display two spheres in the user interface, that represents the hands of the user. The orientation and the position of the real hands are also used to apply a rotation and translation to objects in Unity when they are being grabbed by the user.

Figure 4.10: The sequence in which objects are rotated and translated in the game loop.

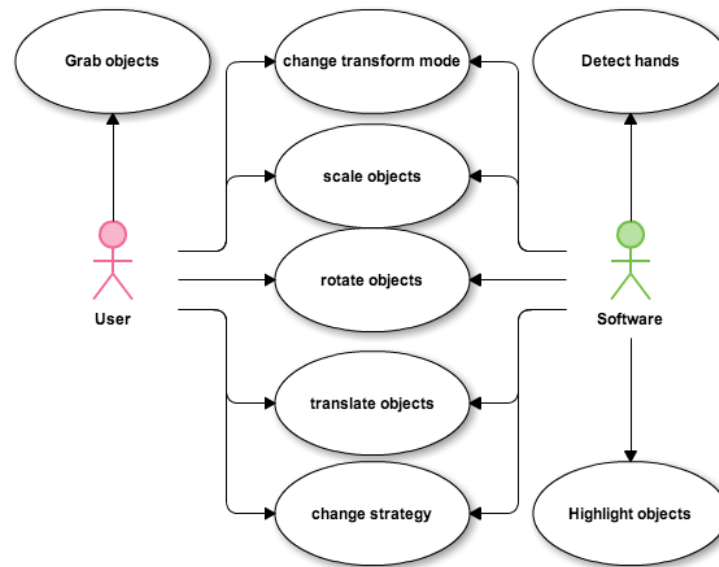


4.2.5 Scenario view

The scenario view illustrates all the functionality that should be possible to do in the application. The view can also help stakeholders (such as the supervisors, and users) understanding the requirements of the system. Figure 4.11 displays a user, and the different functionality the user can do, as well as what the system is able to do. Some functionality is the same for both the

system and the user. The user should be able to rotate and translate objects, and the system has to respond accordingly to the user's needs. However, only the system is concerned about how hands are detected, and only the system is responsible for highlighting objects.

Figure 4.11: The different actions the user and the software are able to perform.



Gestures and commands

An outline and explanation of the different actions, gestures, and commands available for the user will be presented in this section. The definition of what the difference between a gesture, action, and a command was described in Section 2.1.2.

1. Commands

- 1.1 **Grabbing objects** - spread the fingers with your primary hand to grab an object. This is the grabbing pose (see Figure 4.12).
- 1.2 **Release objects** - put your fingers together. This is the normal pose (see Figure 4.12).
- 1.3 **Change mode** - it is possible to change mode (translate, rotate, or scale) by displaying n number of fingers. The number of fingers being displayed determines the mode to use.

2. Actions

2.1 **Translate objects** - while grabbing an object, move the hand around to translate objects.

2.2 **Rotate objects** - while grabbing an object, rotate the hand to apply a rotation to the object. There are different ways of using the hand(s) to rotate grabbed objects. A detailed description of the different ways to rotate can be found in Section 4.2.6.

3. Gestures

3.1 **Rotate objects** - it is possible to rotate grabbed objects by using the secondary hand to create circular gestures with any finger. See Section 4.2.6.

3.2 **Change mode** - it is possible to cycle between modes using tapping gestures on either hand. See Figure 4.13.

Figure 4.12: The hand is used to grab objects by folding it. This can be reversed if wanted.

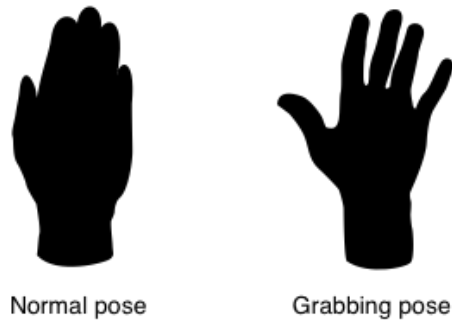


Figure 4.13: A key tap gesture is recognized when the tip of a finger rotates down toward the palm and then springs back to approximately the original position, as if tapping. The tapping finger must pause briefly before beginning the tap.



As mentioned in the list with the commands, actions, and gestures, the command for grabbing is not so intuitive. To grab an object, you need to spread your fingers. Initially, a grab command was executed by folding your hand as seen in Figure 4.12. It was very difficult to rotate the objects according to the rotation of the hand when the it was folded. Leap Motion has some difficulties with reading the normal vector and direction/forward vector when the hand is folded. It is much easier for Leap Motion to read the normal vector and the direction/forward vector when the fingers are spread. However, it still has some difficulties with reading the orientation of the hand when the fingers start to occlude each other.

4.2.6 Rotation strategies

A detailed explanation of how the different rotation strategies were implemented will be given in this section. Four different rotation strategies were implemented. “Joystick rotation” and “Copy Hand Rotation” was implemented as the first strategies. After having a pilot test, the results suggested that some extra strategies would be necessary for a more intuitive and easier way to rotate objects. The “Two Handed Joystick Rotation” strategy and the “Gesture Rotation” strategy was therefore implemented as extra strategies.

Copy hand rotation

The rotation strategy named “Copy Hand Rotation” is intended to copy (or match) the rotation of the grabbed object to the rotation of the hand. The analogy is the same as if you were grabbing a real object, rotating your hand, and put the object back in place. The object would then have its own rotation from before it was being grabbed applied with the extra rotation from the hand (see Figure 4.14).

Figure 4.14: The metaphor for the “Copy Hand Rotation” strategy. a) pick up an object, b) apply a rotation to the target object by rotating the hand, c) the finishing result of the target object after applying a rotation



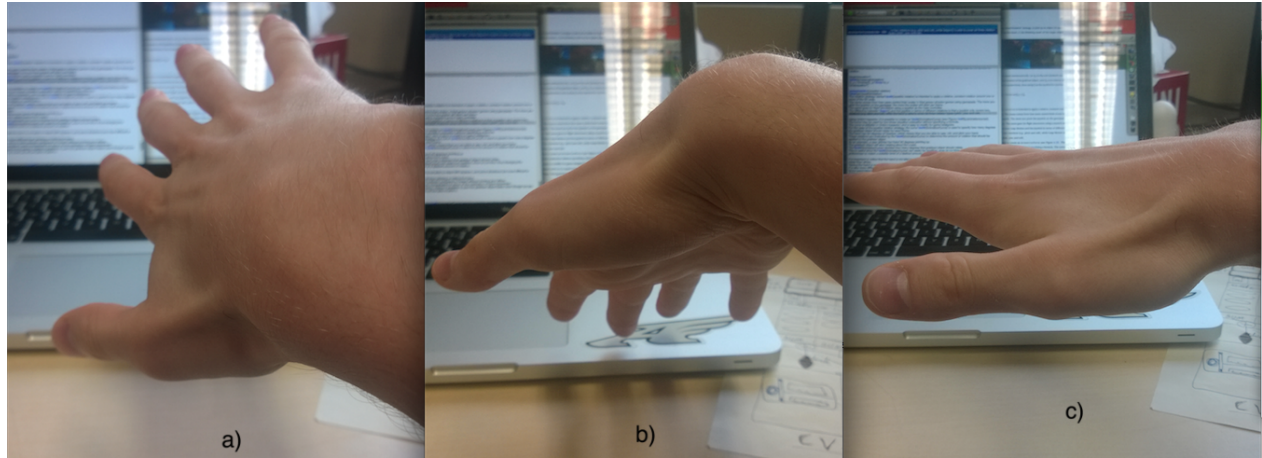
The rotation to the object can be expressed mathematically. Let Q_e be the end (desired) rotation of the grabbed object, Q_s is the initial rotation of the grabbed object, and Q_h is the rotation of the hand. The rotations are represented as quaternions, thus using Q as the symbol for rotation.

$$Q_e = \text{Inverse}(Q_s) \times Q_h \quad (4.1)$$

Joystick rotation

The rotation strategy named Joystick rotation is intended to apply a relative, constant rotation around one or more of the 3D axes. The name comes from how users control their avatar in first person shooter games using gamepads. The more you pitch the joystick on the gamepad, the more the avatar will pitch per frame. The same goes for flight simulators using a joystick as the main controller. The difference between Leap Motion and the joystick in terms of different rotations is that a joystick only covers two rotations (e.g. pitch and roll), while the hand is able to cover all three rotation types (pitch, yaw, and roll). See Figure 4.15 for some examples of how to use this strategy.

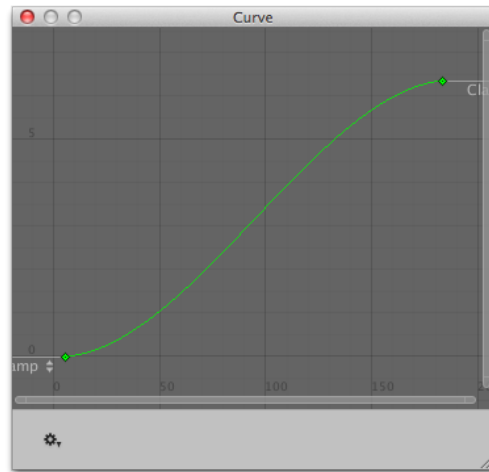
Figure 4.15: The metaphor for the “Joystick Rotation” strategy. a) roll, b) pitch, c) yaw



Using Unity’s inspector, it is possible to create an `AnimationCurve` (see Figure 4.16). The curve may be used for non-linear equation evaluation without specifying a formula. The curve is editable in Unity’s inspector, and is easy to modify for game designers. For the case of the “Joystick Rotation” strategy, the curve `AnimationCurve` is used to specify how many degrees the grabbed object should rotate each frame. The hand has six degrees of freedom, which means that you are able to yaw, roll, and pitch your hand. These rotations are evaluated by the `AnimationCurve` to determine the amount of rotation that should be applied to a grabbed object. As an example, a user pitches her hand 45 degrees pointing up. The grabbed object will start rotating about the x-axis. The `AnimationCurve` decides how many degrees that grabbed object should rotate. This is useful since the `AnimationCurve` can easily be modified, even at run-time, thus changing the preferences to match a suitable level of a desired amount of rotation.

It is important to notice that the hand is not able to rotate 360 degrees, and some directions are more difficult to rotate than others. Yawing the hand is much more difficult than pitching or rolling the hand. It is also more difficult to roll the hand outward compared to inward without moving your elbow. Some parameters in the program code compensate for some of this restriction in hand movement. Adding some extra degrees to the yawing, makes it possible to yaw the grabbed object faster even though we are not able to yaw the hand as much as we are able to pitch it.

Figure 4.16: An AnimationCurve may be used for non-linear equation evaluation without specifying a formula. The curve is editable in Unity's inspector.



Gesture rotation

Users found the controls difficult to use under the pilot test, and other rotation strategies were implemented after the first pilot test. A gesture rotation strategy was one suggestion for rotating objects. This rotation strategy focus on the built in gestures in Leap Motion's APIs, more specifically the `CircleGesture` class. The primary hand is used to grab an object, and the secondary hand is used to make circular gestures (see Figure 4.17). The Circular Gestures are used to rotate the grabbed object around the axis from the center of the circle made by the finger. Leap Motion's API will calculate the axis depending on the circle radius and direction of the finger. If the finger draws a circle in the clockwise direction, the grabbed object should also rotate clockwise. The bigger the circle drawn, the less the object should rotate. It might sound more intuitive to do it the opposite way, but it creates more room for control and precision to draw bigger circles compared to a small one (see Equation 4.2).

Figure 4.17: How a circular gesture is made according to the Leap Motion’s API. A circle movement is recognized when the tip of a finger draws a circle within the Leap Motion Controller field of view.



To describe Equation 4.2, let E_a be the euler angles to rotate the grabbed object. r is the radius of the circle drawn, v is the speed to draw the circle, and k is a constant used to tweak the angles not to be too much or too little. The grabbed object will be rotated the number of angles around the direction axis of the circular gesture.

$$E_a = \begin{cases} r * v * -k & \text{if circle is drawn counterclockwise} \\ r * v * k & \text{if circle is drawn clockwise} \end{cases} \quad (4.2)$$

However, the “Gesture Rotation” strategy turned out not to be suitable as a rotation strategy because the Leap Motion device has trouble reading a 360 degrees circle gesture for all planes in 3D space. Due to occlusion and registering other fingers while doing the circle gesture in certain directions makes the “Gesture Rotation” strategy unsuitable for rotating grabbed objects. This was discovered after implementing the strategy and testing it.

Two handed joystick rotation

The “Two Handed Joystick Rotation” strategy was made in order to use the primary hand for grabbing and translating objects, while the secondary hand should be used to rotate objects. Using this strategy, the user can use the intuitiveness by grabbing objects just like in the material world. The secondary hand will be used to rotate objects in the same way a pilot rotates its airplane. The analogy in this strategy is to virtually grab a joystick (with the same functionality

to the joysticks found in airplanes) with the secondary hand. Users will be able to pitch and roll the grabbed object, but they are not able to yaw the object. To perform a yaw, the user needs to combine a roll and a pitch, which might be a challenge in itself. Rotation is performed when the user virtually grabs a joystick with the secondary hand. The joystick is not visible anywhere, but the user can grab it from anywhere. The moment the user performs a grab command with her secondary hand, the rotation applied to the target object will take place.

The equation to rotate target objects is similar to the “Joystick Rotation” strategy mentioned in Section 4.2.6. The most significant difference is that you are not able to yaw the target objects using the “Two Handed Joystick Rotation” strategy. The strategies also differ in the way the user’s hand is evaluated using Unity’s `AnimationCurve`. In the “Joystick Rotation” strategy, the hand was evaluated by how much it was pitching, yawing and rolling. In the “Two Handed Joystick Rotation” strategy, the rotation is evaluated based on the position of the hand compared to where the initial position was when the user “grabbed” the virtual joystick. The further away (further away along the z-axis) from the initial position the user is, the more the target object should be pitched (downward). The target object would be pitched upward if the user drags the hand toward herself. The target object would roll either left or right depending on the x-position of the hand.

Chapter 5

User study

Now, we have an interest in figuring out if Leap Motion can make it easier for beginners to create 3D models, focusing on moving, and rotating objects in 3D space. The project will be evaluated based on the DECIDE framework together with the GQM model described in Chapter 2. Ideas from similar research have been taken into account, and described in Chapter 3, preventing the reinvention of the wheel. In this chapter, an outline of the different user tests, how they were planned, scheduled, and a discussion of the observations that were obtained will be presented. The user tests consist of usability tests (Stålhane, 2012), which are techniques where users are used to evaluate a system. These tests were used in this project to help evaluate the controls of the system, which was explained in Chapter 4. Usability can also be called an inspection rather than a test. Usability testing focuses on measuring a products capacity to meet its intended purpose. Usability testing measures the usability, or the ease of use, of the system. The main purpose of the usability tests in this project is to figure out if the proposed transform techniques are suitable for 3D modeling applications with Leap Motion as the control interface. A pilot test was held in the early phase of the project with two users, and the pilot test process is described in Section 5.1. After the pilot test, and improving the application prototype, an extensive user test with ten users was conducted (see Section 5.2).

5.1 Pilot test

The first test conducted was a pilot test. The reason for having a pilot test is to find errors about the usability that not necessarily is discovered during the implementation. Pilot users may discover obvious faults that need to be fixed before having a user test with many users. Two pilots were chosen for the early prototype. One of the pilots was a male computer science student studying human machine interactions, making him a good pilot for the test to discuss the technical details of the system as well as the usability of the system. The other pilot was a female worker, not familiar with computer science technology. She was helpful to discover any technical details in the application that was difficult to understand.

The pilot test was performed in a natural working environment. The laptop with Leap Motion was located on a desk, and the users were sitting in front of it (see Figure 5.1).

Figure 5.1: Setup for the pilot test.



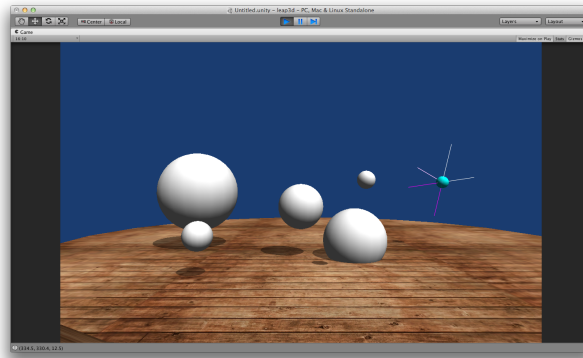
The Leap Motion device was placed right below the arrow keys, giving the users a clear vision of the screen without occluding it with the right hand. The Leap Motion device can be placed anywhere, but putting it at an unsuitable location might make it more difficult for the user to see the screen, or the device might not get good readings. Section 5.1.1 contains information about how the pilot test was planned, and also discuss the results from the test.

5.1.1 Tests

In this section, the pilot test scenario will be described. The users were given three tasks. The first task was to arrange five spheres in increasing order from left to right. This task was merely

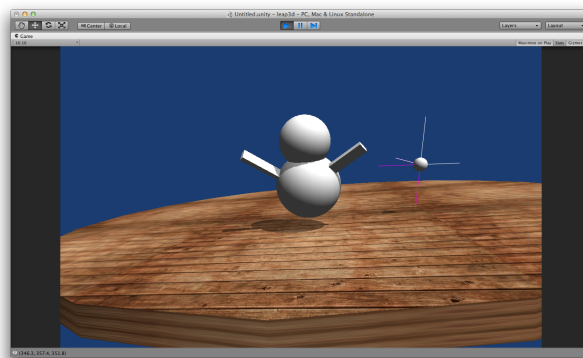
a task for the users to become familiar with moving objects around. A picture of the task can be seen in Figure 5.2.

Figure 5.2: First assignment for the test users. The purpose was to make the users familiar with moving objects around in 3D space.



No rotation took place in this assignment. The second task was to create a snow man using two spheres (body and head) of different sizes, and two rectangular parallelepipeds (arms) as seen in Figure 5.3.

Figure 5.3: Second assignment for the test users. Users were asked to create a snow man out of two spheres and two rectangular parallelepipeds.



These two tasks were completed using the “Copy Hand Rotation” strategy (an explanation of this strategy as well as the other strategies implemented in this project can be found in Section 4.2.6). The third, and final task was the same as the second task, but users were asked to use the “Joystick Rotation” strategy instead of the “Copy Hand Rotation” strategy. That is, the users had to create a snow man (as seen on Figure 5.3) using the “Joystick Rotation” strategy. The “Joy-

stick Rotation” strategy was chosen as the second strategy to use in this user test on purpose, as it was assumed to be the most difficult strategy to use.

The pilot testers struggled to rotate objects using the “Copy Hand Rotation” strategy when it became physically impossible to change the rotation of the hand. It felt unnatural for the users to grab the object, do a rotation, release the object, and then do a new rotation. They were more eager to complete the rotation in one rotation step instead of several steps. Even though the rotation strategies were difficult to comprehend, one of the users said that she felt she became much better using the system on the second attempt to create a snow man. It is reasonable to think that the system can be easy to use after practicing.

The pilot tests were asked to fill out the questionnaire given in Appendix A. One of the pilot testers did not find it intuitive to move the object around, as spreading the fingers is not a natural pose for the hand compared to folding the hand. grabbing objects in the application worked opposite compared to real life. The application suggests you to open your hand as wide as possible, which is the exact opposite of grabbing. The reason for not implementing the folding hand gesture for grabbing was because Leap Motion had problems recognizing the orientation of the hand when there are no fingers present. Spreading the fingers with a wide open hand makes it easier to perform the rotation strategies implemented at this point. Sacrificing the intuitiveness of grabbing to make it easier to rotate objects was a design decision made on purpose. It was assumed that the pilots would find the gesture strange or a bit uncomfortable. The second pilot thought that both controls were very intuitive. It might be because this pilot already had experience with computer games and modeling, but the first pilot did not have much experience at all in these fields.

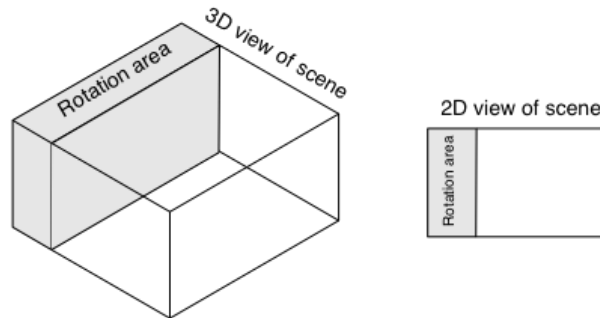
5.1.2 Suggestions from pilots

The users found the “Joystick Rotation” strategy the most complex one, and both strategies were difficult to control efficiently. It may be a good idea to create other rotation strategies in order to find more suitable ways to rotate grabbed objects.

One of the pilots suggested to use one hand for moving objects, and another hand for rotating. The pilot specifically said that maybe it is possible to have a specific area where you can do rotation (see Figure 5.4), and maybe have a virtual joystick that control pitch and roll of the

grabbed object.

Figure 5.4: A proposed method by one of the pilot users to do rotation. One hand will be used for grabbing and moving objects, while the other will be used for rotation within the specific area. The rotation area is the area the user needs to have its secondary hand in order to do a rotation.



The idea was refined to further come up with two new rotation strategies. Instead of having a specific area to control rotation, it seemed better to freely control where you want to perform the rotation independent on area. A detailed description of the suggested strategies can be found in Section 4.2.6. One of the users also suggested to implement an atmospheric effect, which is to change the color of the objects in the scene based on depth information. Objects may take a brighter color if they are far away, and a darker color if they are close to the user. The atmospheric effect applied with shadows makes it easier to distinguish the depth information of the objects.

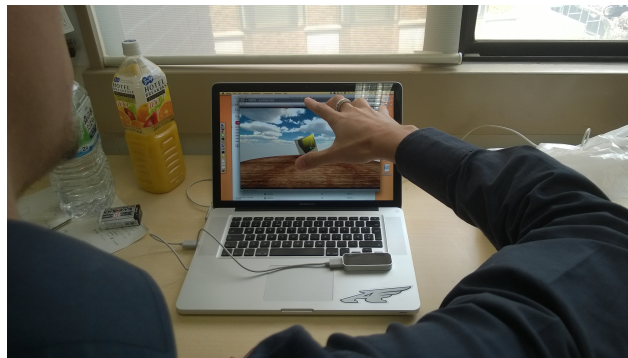
5.1.3 Next step

Creating a test scenario similar to the ones presented in (Ken Hinckley, 1997) may be a suitable scenario for the extensive user tests, with more than the two initial rotation strategies. They present test scenarios where you must match the rotation of an object to be similar to one being pictured on the screen. It will be possible to determine the speed and accuracy of rotating objects to compare the different strategies implemented in this project. The test scenarios in (Ken Hinckley, 1997) did not consider translation and rotation together, only rotation, making the test scenario in this project not directly comparable to their study.

5.2 User test

A description of how the user test was conducted will be given in this chapter. Similar to the pilot test, this user test was performed in a natural working environment as described earlier in Section 5.1, but with a more formal outline following guidelines for user studies as presented in (Svanæs, 2012) and (Stålhane, 2012). A picture of one of the users trying the application can be seen in Figure 5.5.

Figure 5.5: Users were sitting in front of the computer with Leap Motion on their right hand side.



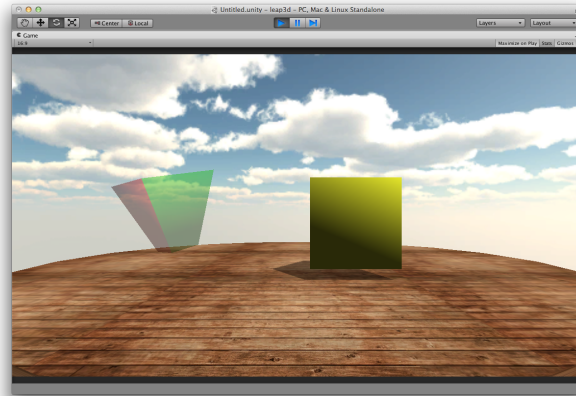
5.2.1 Tests

How the user test was conducted will be given in this section. The subsections in this section will discuss what the different users thought about the different rotation strategies: “Copy Hand Rotation”, “Joystick Rotation”, “Two Handed Joystick Rotation” (a description of how these strategies work can be found in Section 4.2.6).

The users were asked to perform three different test scenarios. One test scenario for each of the rotation strategies: “Copy Hand Rotation”, “Joystick Rotation”, and “Two Handed Joystick Rotation”, was conducted. The “Gesture Rotation” strategy was not part of the user test because it proved to be an inaccurate and imprecise strategy during the development stage. More details about why this strategy was omitted can be found in Section 4.2.6.

The test scenarios conducted in this project was similar to the scenarios presented in (Ken Hinckley, 1997). Each user were asked to rotate and translate a 3D object to match an identical, but transparent 3D object as seen in Figure 5.6.

Figure 5.6: Users were asked to match the transparent 3D object and the opaque 3D object by rotating and translating the opaque object.



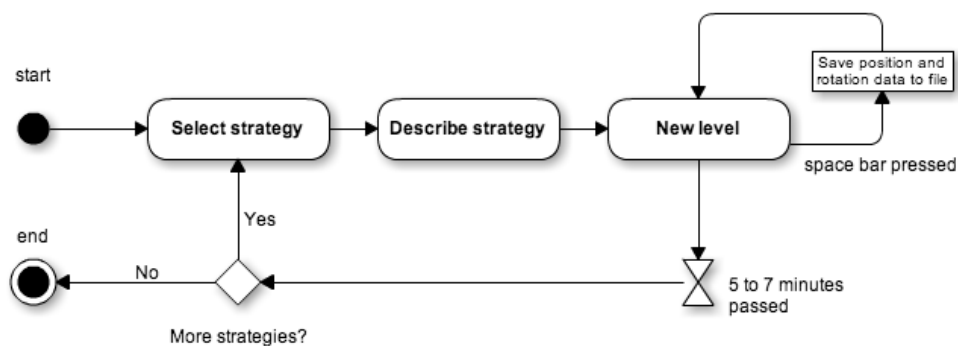
Before starting the user tests, the main points from (Svanæs, 2012) was taken into account when scheduling the user tests. Ten main main points were followed before scheduling the tests with the users.

1. The test conductor introduce himself to the users.
2. The users were told the meaning/reason for having the user test. And that it is the system that is being tested, and not the user.
3. Users were told that they could quit the user test at any time, especially if they felt any pain in their arms while using the system.
4. The users were told what Leap Motion is, and how it works.
5. The test conductor would not be able to physically help the users during the test.
6. The test conductor explained the users how to think loud, and the users were asked to think loud while doing the test scenarios.
7. The assignment was be explained to the user.
8. Users were told that they could ask any questions at any time, both before and during the test.

9. When the user test was done, the users were told to say anything they had on their mind before answering the questionnaire.
10. After the user test, and the questionnaire, the data from the test would then be used to answer the research questions.

The test conductor showed the users how to use the different rotation strategies by example. The test conductor would first tell the users how the “Copy Hand Rotation” strategy worked, and show them how to translate and rotate the objects. If the users had any questions regarding the controls, the test conductor would then assist the users in order to help them understand how to use the controls. Test conductor would not help the users in completing the tasks. Users were asked to perform as many completions as possible within five to seven minutes for each strategy. It was anticipated that users would have difficulties mastering the controls. Each test scenario had a time limit to see how many tasks users were able to solve. This approach would be a better approach than to ask the users to solve n number of tasks. Figure 5.7 illustrates how the test scenarios were conducted using a flow chart diagram.

Figure 5.7: A strategy would be selected, and then described to the user. The user would then start the assignment, and complete as many levels as possible within the time limit (a level is simply to match the opaque cube to the transparent cube). After the time limit, a new strategy would be selected. If there are no more strategies to test, the user test would end.



When the user was satisfied with the position and rotation of the object, the person conducting the test would press the “space bar key”, and a new “level” would start. A “level” is basically

the task in which the user must align the opaque cube into the the randomly positioned and rotated transparent/ghost cube. When five to seven minutes had passed, the next strategy would then be explained to the user, and tested.

After the three assignments, the users were asked to fill out a questionnaire. The questionnaire was the same as the one used in the pilot test with a few additional questions. The questionnaire for both user tests can be seen in Appendix A. This questionnaire is helpful to understand how useful the system is, and also cover which strategies are the best suited ones. The questionnaire is similar to the System Usability Scale (SUS) questionnaire (Brook, 1996). Questions with the keyword “system” was replaced by the keyword “controls” instead. As an example, the question “I thought the system was easy to use” would be replaced by “I thought the controls were easy to use”. The results from the questionnaires will be listed in Chapter 6.

The tasks in this user test would prove to be much more difficult than the tasks in the pilot test. The main reason for this is because the level of precision was required to be much higher in this user test. As the pilot test focused on how easy it was to use the controls, the users did not have any specific goal other than to create a snow man. A snow man do not have to be symmetric, nor does it require precise assembly of the different body parts to look like a snow man. The tasks in this user test however, required the user to be very accurate. Users would try to rotate their hands in positions where Leap Motion did not understand how the orientation of the hands. The system would then interpret the hand as lost, and the user would get frustrated by the fact that the system did not accomplish the desired task the user intended to do. This issue will be discussed in Chapter 7 The test users found it exciting to try out the new technology, even though they found it difficult to use the proposed strategies.

Copy hand rotation

This strategy is the most intuitive one, but some of the users were confused of how the strategy worked (see Section 4.2.6 for a detailed description of how this strategy works). Users missed the haptic feedback, which would make the strategy more easy to understand. When users tried to grab and rotate an object, they would be surprised when the system behaved differently to what their intentions were. Others found the technique very intuitive, but it was a struggle to adjust the hands to rotate the object, especially when yawing.

Joystick rotation

This strategy was the strategy that most users performed the best with (see Section 4.2.6 for a detailed description of how this strategy works). The users did not have to move their hands in a complicated matter to do rotations compared to the “Copy Hand Rotation” strategy. Users thought it was easier to use this strategy because they were able to predict the rotations more easily. It was easier to rotate the object by one of the axes at a time, instead of trying to rotate the object about all the axes as they had to do with the “Copy Hand Rotation Strategy”.

Two handed joystick rotation

This strategy was the most difficult one for the users to comprehend (see Section 4.2.6 for a detailed description of how this strategy works). The least tasks were completed, and users were confused with how the controls worked. There were occasions when the system did not register the users’ hands, and that caused the users to get frustrated. The hands were not registered mostly because the position of a user’s hand became out of reach from Leap Motion’s field of view. Yawing (rotation about the y-axis) was not possible with this strategy because it lacked a third axis to rotate about. Some users did not understand how they would match the opaque cube to the transparent cube when they wanted to “twist”/yaw the opaque cube. They had to combine two rotations (a roll and a pitch) in order to simulate a yaw, but this was difficult to understand for many users. However, the users were able to rest their elbows on the table using this strategy, making it less aching for the hands.

5.2.2 Comments and observations

Users had some interesting comments about the system during the user test. Suggestions from users can be useful for further research, and their comments can help answer RQ2.2. Comments from users, as well as observations made during the user tests will be discussed here.

Doing rotations and translations in the same action was difficult. Some users tried to complete the rotation first, and then place it on top of the transparent object. This tactic seemed to be more difficult because when the rotation seemed correct from a far away position, the user would then have to adjust the position a little bit more when the object came on top of the target

transparent object. The tactic was difficult to execute because of the perspective view, and can be illustrated on Figure 6.2 in Chapter 6. From the figure, it may seem that the rotation of the objects are different, since the red surface on the right object is barely visible, but the rotation of the two objects are equal. The rotations seem different because of the perspective view.

It was anticipated that the “Copy Hand Rotation” strategy (see Section 4.2.6 for the details about the strategy) would be the easiest strategy to use, as it tried to simulate how to grab objects in the real world. However, many users did not find the controls intuitive nor realistic. Many were unsure on how the rotations worked when they grabbed the objects. The rotation felt different if the user grabbed the object from the top (the narrow part of the object) compared to the bottom of the object. Some users also felt that the rotation strategy worked opposite to what they wanted to do. One of the users said:

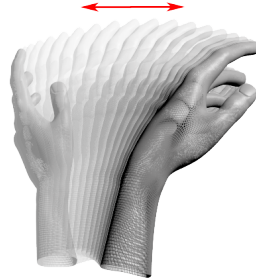
I cannot understand the correspondence between the rotation axis and the hand movement.

Similar comments about the “Copy Hand Rotation” strategy was also mentioned by other users. The main conclusion is that the strategy did not work as intended, and could have been implemented differently (see Chapter 7 for more details).

One user suggested that it might have been easier if it was possible to do a rotation on just one axis at the time. This technique could be suitable for the “Joystick Rotation” strategy and the “Two Handed Joystick Rotation” strategy, but applying it to the “Copy Hand Rotation” strategy would probably confuse users even more. For a detailed description of how the strategies work see Section 4.2.6. Locking two of the axes while rotating on only one axis could prove to be a good idea, but it needs to be tested. The user did not say anything about how he would go about doing it using Leap Motion, so the implementation details were left to the developer of the system. This particular issue was thought about, and an idea of how to lock rotations to only one axis at a time was suggested. First, there must be a way to change into a “lock axis”-mode, and that can be done by displaying n number (e.g four fingers) of fingers above Leap Motion. Second, it must be possible to select the desired axis to rotate about. When a user selects “lock axis”-mode, it should be possible to point on objects to select them, and when selected, the user can swipe their finger up, down, or to the side to select the desired axis to rotate about (this can be done by using the `SwipeGesture` as seen in Figure 5.8). Then, the user can grab the object as

usual, and start rotating it, focusing on one axis. If the user wants to change the axis, she must select the object by pointing on the object, and then swipe the finger to indicate which axis she wants to rotate about.

Figure 5.8: The `SwipeGesture` class represents a swiping motion of a hand and a finger or tool.



One of the users suggested to make the “Two Handed Joystick Rotation” strategy more sophisticated, by adding an extra degree of freedom. Since the height position of the hand was not used to do anything, the user suggested that the height-position of your hand could yaw the grabbed object. Implementing this might be a good idea for experienced users, but according to how users operated their hands above the Leap Motion, it is highly likely that it would be even more difficult to rotate the objects by adding the extra degree of freedom.

One of the users suggested to work with the application while standing instead of sitting. Since users were sitting, they had to lift their arms, causing more stress on the upper arms and shoulders. If users could stand and work with the application, they would put less stress on the upper hands and shoulders, thus being able to use the application longer not getting tired. This hypothesis must be tested to figure out whether or not users get less exhausted.

Chapter 6

Result from user tests

From the previous chapters, we have now followed the development process, implemented the software, and gotten data from extensive user tests. In this chapter, the results from the user tests described in Chapter 5 will be evaluated. Accuracy and time was measured during the user tests, and will be discussed and illustrated in Section 6.1.

6.1 Test results

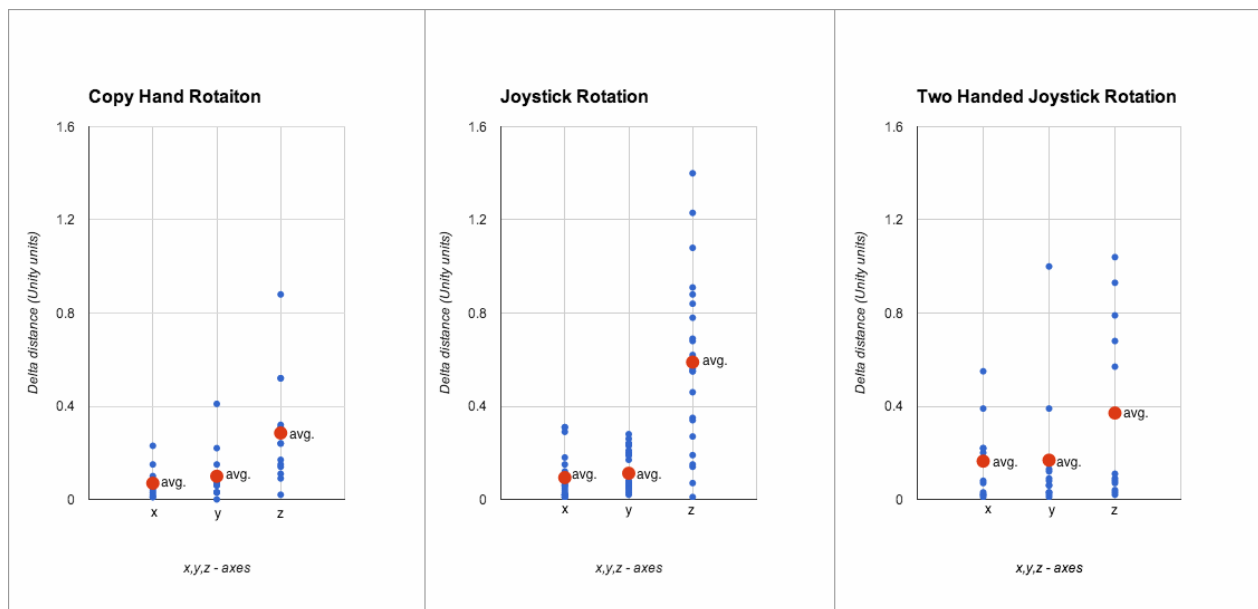
The data from the user test will be displayed and discussed in this section. Ten users helped in the understanding of how mature the control strategies suggested in this project together with Leap Motion are. Four of the research questions outlined in Section 2 can be answered based on the results given in this section. The research questions are:

- RQ1.1: What are good commands, actions and gestures for navigating, translating, rotating and scaling objects in 3D space using Leap Motion?
- RQ1.2: Do people find the controls intuitive and easy to use?
- RQ1.3: Are users able to rotate and position objects accurately and fast in 3D space using Leap Motion?
- RQ2.2: Can Leap Motion introduce new/improved ways of navigating, translating, rotating, and scaling objects in 3D space compared to similar technologies.

6.1.1 Logged data

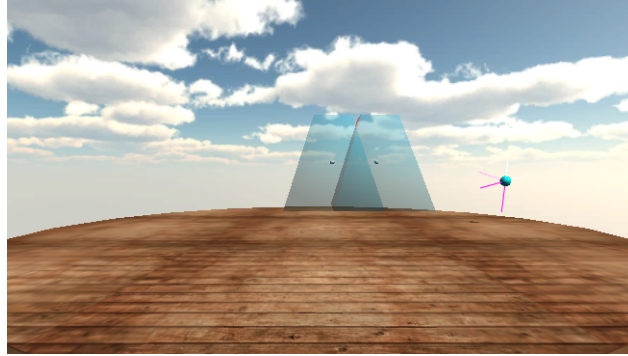
The results for answering RQ1.3 can be measured by storing position and rotation data from the user tests, as well as record the time spent doing the tasks. Part of RQ1.1 can be answered by interpreting the results from the user tests. Figure 6.1, Figure 6.3, and Figure 6.4 are graphs which display all the test results. Figure 6.1 contains position records from the user test.

Figure 6.1: The three charts display the error distance for the different strategies. The positions are stored as Unity units. Each circle/dot in the graph represents the result for one completed task by a user. The slightly bigger circle/dot represents the average score for each of the different axes.



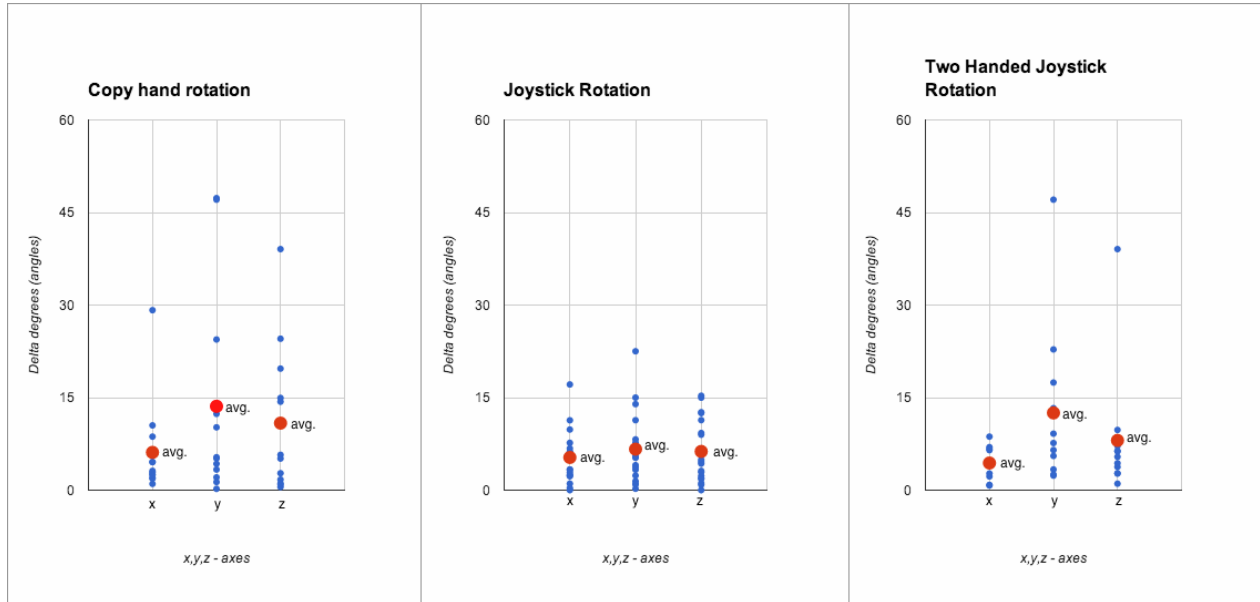
The units in the graph are measured in Unity units. It is difficult to say how far one unit is, but Figure 6.2 shows how much one Unity unit is in the application. The object's center is the point in which distance is measured. Two objects can be equally far away from each other even though they local rotation is different.

Figure 6.2: Two objects are positioned exactly one Unity unit away from each other (on the x-axis). The sphere in the middle of each object represents the center of the objects. The charts are aligned next to each other to make it easier to compare the strategies



According to Figure 6.1, users found it more difficult to position the target object on the z-axis (the depth axis). It is natural to think that the z-axis is the most difficult to position correctly, as a 2D screen does not provide good depth information. A second observation is that the “Joystick Rotation” strategy has the most data, which means that most users were able to complete more tasks using the “Joystick Rotation” strategy compared to the other strategies. According to the results, users are able to position objects accurately with all strategies. More interesting is to see how well users could align rotation. Figure 6.3 contains rotation records from the user test.

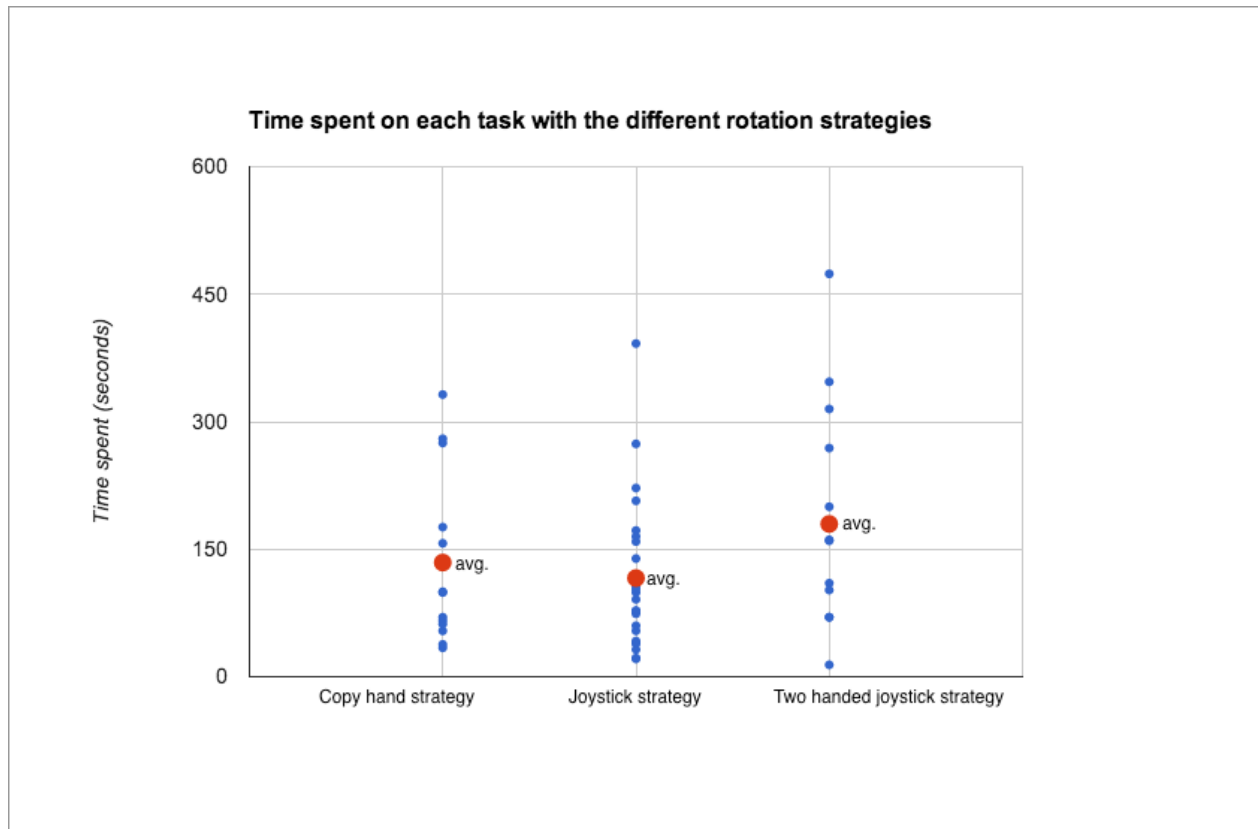
Figure 6.3: The three charts display the difference in rotation accuracy for the different strategies. The data are measured in angles. Each circle/dot represents one completed task for a user. The slightly bigger circle/dot represents the average score for each of the different axes. The charts are aligned next to each other to make it easier to compare the strategies



The units in the graph are angles, and the difference in rotation are plotted in the graph. The y-axis was the most difficult axis to align the objects accurately on, even though the results compared to the two other axes are similar. The “Joystick Rotation” strategy proves to be the strategy with the most completed tasks, and also the lowest error rate in average.

The time spent on each task was also measured. Figure 6.4 displays the time spent on each of the different tasks for each strategy.

Figure 6.4: The chart displays the time spent for each task for each strategy.



The units measured in the graph are seconds. The “Joystick Rotation” strategy is the strategy with the most successful tasks done, as well as the strategy users spent the least amount of time to completing tasks. However, none of the strategies in this project was easy enough to use in order to complete tasks as fast as the results in (Ken Hinckley, 1997). They had average means of 15 to 34 seconds for the different rotation techniques/strategies. The average mean in this research are between 116 to 180 seconds. When it takes two minutes to rotate and place an object, it would take far too much time to accurately assemble many parts together. The rotation techniques may be more useful when precision is not important (this will be discussed in Chapter 7).

6.1.2 Questionnaire

Not only were data logged from the user tests, but the users were also asked to fill out a questionnaire, as well as come with suggestions, tips, and comments to the system. From the ques-

tionnaires, it is possible to answer RQ1.2: “Do people find the controls intuitive and easy to use?”. Users were asked to rate the different rotation strategies, and the results are displayed in Figure 6.5, Figure 6.6, and Figure 6.7. As an overall view of the strategies, the “Joystick Rotation” strategy is the best one, but the results are not too different from each of the strategies.

From the charts, none of the strategies were very comfortable to use. All of the users felt their hands were aching during the tests. They were not able to rest their hands on the table, but had to keep the elbow in mid air in order to do the different rotations.

Most of the users found the strategies easy to understand, but difficult to master. The “Joystick Rotation” strategy is the strategy that proved to be most successful. The reason for that was because users who actually managed to complete quite a few number of tasks were able to do it with the “Joystick Rotation” strategy, giving it high scores in the questionnaires.

Figure 6.5: Results from what users think of the “Copy Hand Rotation” strategy. Each bar represents the score for each of the questions in the questionnaire. The left most bar is the score for the top most question, and the right most bar is the score of the bottom question.

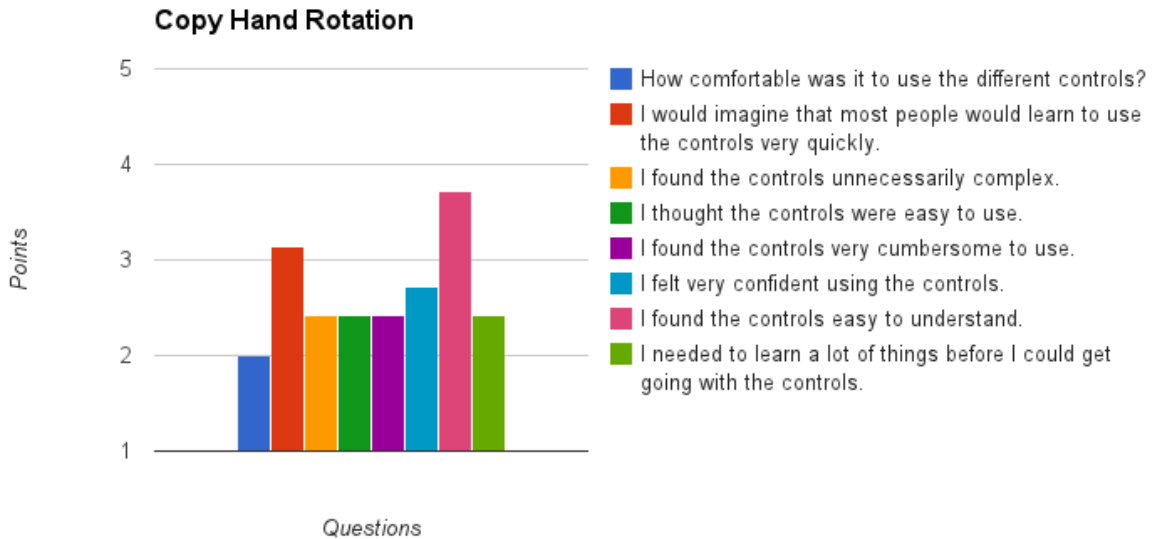


Figure 6.6: Results from what users think of the “Joystick Rotation” strategy. Each bar represents the score for each of the questions in the questionnaire. The left most bar is the score for the top most question, and the right most bar is the score of the bottom question.

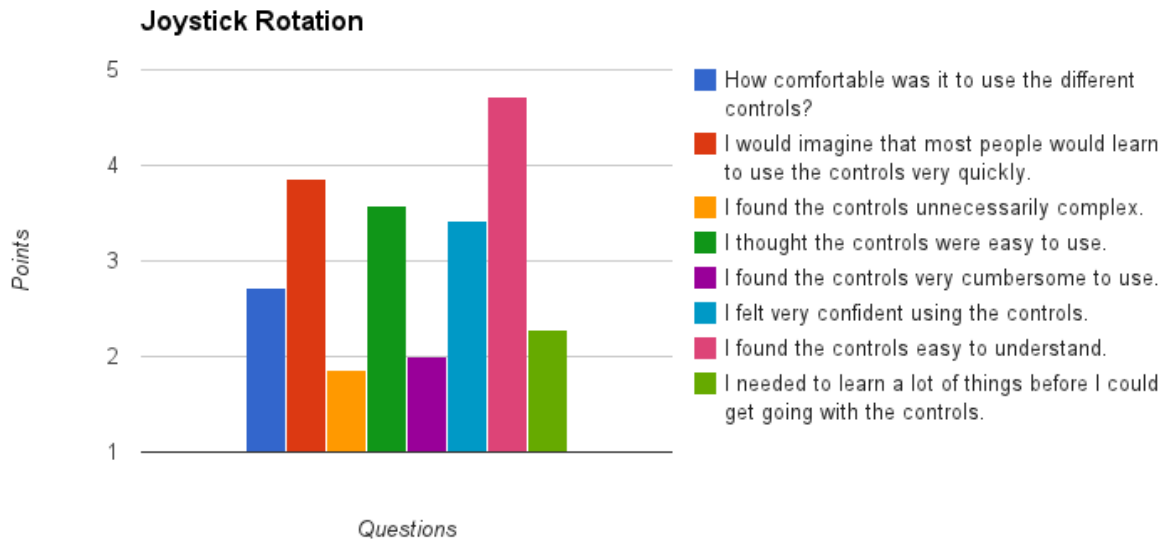
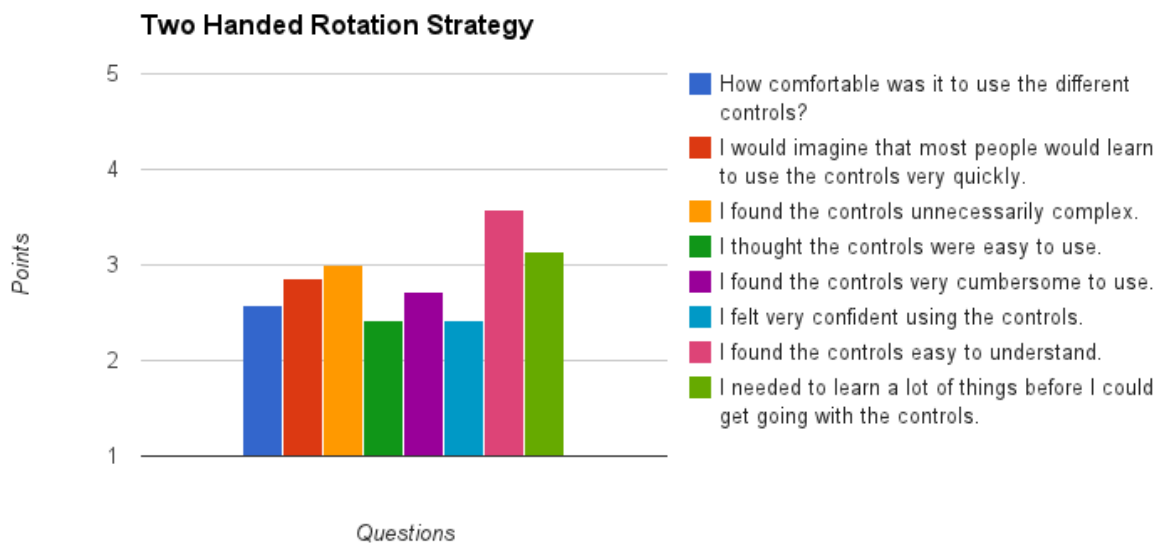


Figure 6.7: Results from what users think of the “Two Handed Joystick Rotation” strategy. Each bar represents the score for each of the questions in the questionnaire. The left most bar is the score for the top most question, and the right most bar is the score of the bottom question.



Chapter 7

Discussion and conclusion

During this project, an application focusing on making homogeneous transforms have been implemented and tested by users. A literature study together with the user tests have made it possible to answer the main goals for this project. We have quantified the results from the user tests, and are ready to discuss the results in more depth, as well as giving a precise answer to the research questions given in Chapter 2. In this chapter the significance of the results will be discussed as well as the challenges during the project and how the project can be developed further. The chapter starts by introducing some comments and interpretation of the results obtained from the user study and the literature survey. The main goals of this project is to understand how well users can use Leap Motion to interact with objects in 3D virtual space. The research questions are listed below, and a discussion of each of them will be done in Section 7.1. Short and precise answers to the research questions can be found in Section 7.2.

- RQ1.1: What are good commands, actions and gestures for navigating, translating, and rotating objects in 3D space using Leap Motion?
- RQ1.2: Do people find the controls intuitive and easy to use?
- RQ1.3: Are users able to rotate and position objects accurately and fast?
- RQ2.1: What are common ways to navigate, translate, rotate, and scale objects in other applications where a traditional mouse is not the main interface?

- RQ2.2: Can Leap Motion introduce new/improved ways of navigating, translating, and rotating objects in 3D space compared to similar technologies?

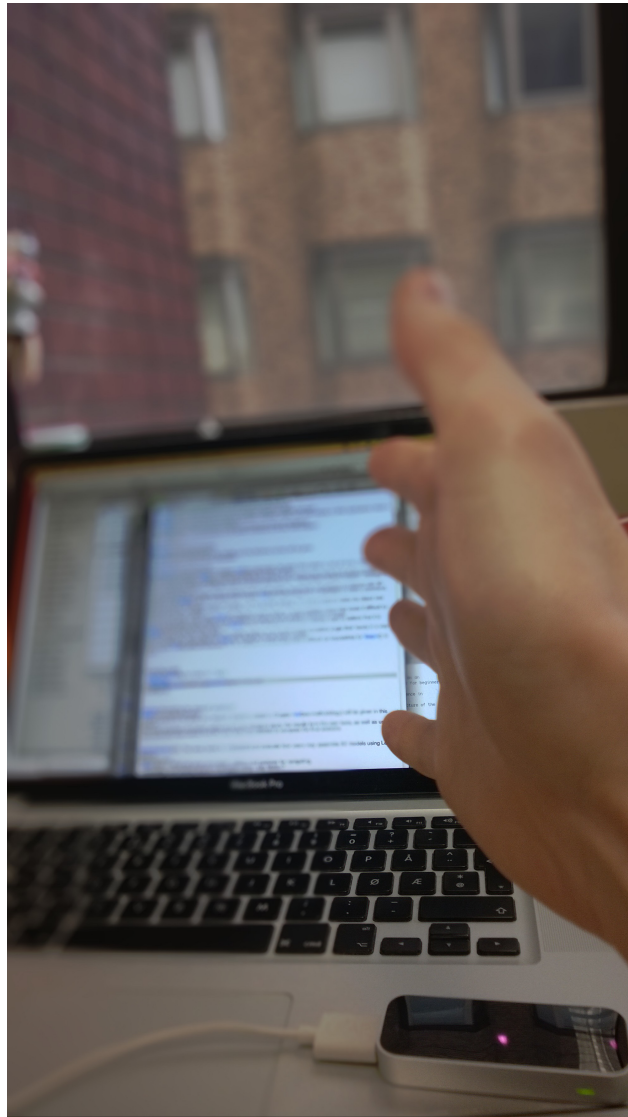
7.1 discussing the results

In this section, a discussion of the results from the user test and the literature survey will be given.

7.1.1 User experience

After the user tests described in Chapter 5, the users were asked to rate all of the rotation strategies (see Section 4.2.6 for a list of all the rotation strategies). The “Joystick Rotation” strategy was rated as the best strategy if we compare the results from all the strategies. The “Copy Hand Rotation” strategy comes on second place, and the “Two Handed Joystick Rotation” strategy was the least favorite one. Implementing these strategies was an attempt to find good actions, commands, and gestures to interact with 3D models using Leap Motion as the controller (see Chapter 2 for a description of what a command, gesture, and an action is). Even though the “Joystick Rotation” strategy was the favorite strategy, the actions used to rotate the object was difficult for users to master. If it was user error, or error with Leap Motion’s detection of the hands, or implementation errors that made it difficult for users to accurately rotate objects is not possible to measure in this project, and is out of scope. [Weichert et al. \(2013\)](#) show that Leap Motion has an accuracy of 0.01mm, making it hard to believe that it is Leap Motion that reads the wrong data. Not only were the users inaccurate with rotating objects, but the users would sometime angle their hands in a way that it is difficult (or impossible) for Leap Motion to read the hands angle correctly. See [Figure 7.1](#) for an example of a hand pose that is difficult (or impossible) for Leap Motion to read.

Figure 7.1: Having the hand angled like the picture is not something that Leap Motion is able to read. The fingers are occluding each other, and Leap Motion thinks that it is just one finger pointing on the screen.



It is however possible to create code that assumes that hand is in the position it is in when Leap Motion has difficulties reading orientation of the hand. The developers of Leap Motion are continuously developing new software to improve the robustness of the device. Since June 2014, they have implemented a better skeletal representation of the hand, as well as improved the reading of data from difficult cases, such as the pose given in Figure 7.1.

Given that it was difficult to rotate objects, users proposed that it might be easier to rotate objects if users were able to rotate objects about one axis at the time. Users also suggested to

separate movement and rotation of objects into two different actions.

7.1.2 Research experience

From Chapter 3, other similar applications, research, and technologies were introduced. The handlebar metaphor presented in Peng Song (2012) is a good approach to translate and rotate objects in 3D virtual space. Recent research by Yupeng Zhang (2013) use the same metaphor when editing models from outside of the avatar. The handlebar metaphor is a good approach for Microsoft's Kinect, since it has difficulties with reading many different hand postures, or hand commands. Implementing the handle bar metaphor with Leap Motion could prove to be the best option for this project as well. Leap Motion is able to do more sophisticated commands, actions, and gestures with the hands than what is possible with Microsoft's Kinect. The handle bar metaphor was therefore not implemented, since other techniques might work better. The research from Peng Song (2012) do not say anything about muscle pain in the arms while using the system, but it is highly likely that the arms would get exhausted after a few minutes using the handlebar metaphor with Leap Motion.

7.2 Answering the research questions

A short answer to the goals and research questions stated in Chapter 2 will be given in this section. The most important information gathered from the literature study, the results from the user tests, as well as users' comments and observations from the user tests was selected to compose the final answers.

G1: The study seeks to understand and evaluate how users may assemble 3D models using Leap Motion

- RQ1.1: What are good commands, actions and gestures for navigating, translating, and rotating objects in 3D space using Leap Motion?
 - The "Joystick Rotation" strategy (see Section 4.2.6 for a detailed description of how the strategy works) proposed proved to be the best strategy implemented in this

- project. Users were able to perform the best with the “Joystick Rotation” strategy, and users also rated it as the best strategy out of the strategies proposed.
- The command used to move objects with the “Joystick Rotation” strategy is a good alternative to consider for future hand tracking devices, and applications used to interact with objects in 3D space.
 - Rotating objects using Leap Motion was not an easy task. The best way is to focus on rotations of one of the axes at the time, as beginners are not able to accurately control rotations along all axes in one operation.
 - The command for grabbing objects was not intuitive, but it worked well, as users were able to position objects easily with all strategies implemented in this project
- RQ1.2: Do people find the controls intuitive and easy to use?
 - People find the controls easy to understand, but difficult to master.
 - Half of the users believed that others are able to learn how to use the controls after practicing for a while.
 - The techniques suggested in this research are not ergonomic enough to be used for long periods of time, because users felt muscle aching in their arms just a few minutes after using the controls.
 - RQ1.3: Are users able to rotate and position objects accurately and fast?
 - Users are not able to rotate and position objects fast. It takes almost two minutes in average to translate and rotate objects for beginners. The rotation takes too long time to get correct compared to the time to do a rotation using computer mouse. A computer mouse can do a rotation with the same (or better) accuracy as with the strategies proposed in this project, but within a mean of 22.1 seconds.
 - Positioning objects however, are easy using the strategies in this project.
 - Combining translation and rotation in the same action proved to be difficult as users accidentally failed to keep an accurate rotation of the objects when they tried to position them.

- The mean angular error was between 4.4 degrees and 13.6 degrees for the different implemented strategies.
- The mean time spent to move and rotate an object to a desired location took 116 to 180 seconds for the different implemented strategies.

G2: The study seeks to understand how other applications create 3D models

- RQ2.1: What are common ways to navigate, translate, rotate, and scale objects in other applications where a traditional mouse is not the main interface?
 - The most common way to translate objects are to select them with the interaction device, and then move the interaction device in real world coordinates which translates into virtual coordinate for the object.
 - Scaling objects usually require a two handed approach, where a user can shrink or enlarge objects by moving the hands together or further away from each other respectively.
 - The most common way to rotate objects are also by using two hands. A handlebar metaphor is intuitive and easy to use for novice users to translate and rotate objects.
- RQ2.2: Can Leap Motion introduce new/improved ways of navigating, translating, and rotating objects in 3D space compared to similar technologies?
 - Leap Motion do promote new ways of rotating and translating objects in 3D space, but it prove to be too difficult to use with the strategies proposed for beginners.
 - Leap Motion is not suitable to do sophisticated actions and commands for accurately rotating and positioning objects in 3D virtual space, even though the device is able to recognize many different gestures such as tapping, waving, and making circles.
 - Using the strategies implemented in this project put much stress on the users' hands, making it impossible to use the device for longer periods of time.
 - The device might be more suitable for simple commands and gestures which do not require precise accuracy and precision.
 - Implementing the handlebar metaphor might prove to work well.

7.3 Challenges

Some challenges around the project will be discussed in this section. What went well, and what did not go so well will also be discussed.

During the development of the software, Leap Motion had problems with detecting the hands. The hands would sometimes disappear from Leap Motion's field of view, and not being rediscovered again. This issue would cause users with much irritation, as they had to put their hands on the table, and up over Leap Motion again in order to make the device track their hands again. This issue was also discovered during development, but it occurred less often, and it was difficult to figure out the real problem for this issue. The developers of Leap Motion continuously deliver new versions of their software, improving the underlying code to discover and track hands, fingers and tools. Despite these updates, the system in this project would not improve its hand detection. The problem causing the poor detection of the hands could either be in the implementation of the software, or the bridge between Unity and Leap Motion. The problem arose mostly when users put their hands too close to the device, or put their hands outside Leap Motion's field of view, or rotated the hand in an angle unrecognizable by Leap Motion. The software could have had more visual feedback to the users because of this issue. Even though the software did have some visual feedback about when the users were grabbing objects, and displaying a sphere for each of the detected hands, the software could contain even more visual effects to make it more user-friendly. The spheres (that represented the hands in the software) were too subtle, and they would disappear from the user's field of view if it was occluded by objects in the scene. A solution to this problem could be to make the objects that occluded the "hands" change their transparency.

The validity of the user test could have been improved even more. Instead of having randomly computed locations and rotations for the transparent cube in the user tests, it would be better to generate equal position and rotations for each level, making the test the same for all test users. The first level could let the transparent cube intentionally be positioned and rotated more easily than the other levels, and increase the level of difficulty for each level/task. Having a set with position, and rotation pairs for each level would also increase the validity of the results, and the project. It would make it more accurate for others to create the same test scenarios, and

compare their results to the results in this project. This approach would increase the validity of the user test.

Even though there were some challenges during the project, there were also some tasks that went well, and problems that was anticipated to be difficult turned out not to be difficult at all. Integrating Leap Motion with Unity was anticipated to be difficult, but there were some detailed instructions on the web pages of Leap Motion (<https://leapmotion.com>, accessed 17-06-2014) that made it easy to integrate Leap Motion with Unity. The documentation of Leap Motion's API was also well written, and it was easy to understand how to use the API in the code written in this project.

Integrating some of the code from a the project by [Johan Klokhammer Helsing \(2013\)](#), which focused on how an avatar can be controlled with Leap Motion in first person perspective, went really well. The software made for that project could be easily modified to add extra features, and also making changes to the existing functionality.

7.4 Future work

Future work would be to figure out a way to use Leap Motion to actually create 3D models, and to create a fully functional 3D modeling software exploiting the advantages with Leap Motion. It could be possible to extend the current applications to mold virtual clay. Instead of just removing and forming the clay, it could be interesting to also have the option to add clay to the model. Creating an application where you draw in the air using a finger to create more abstract figures in 3D could be an interesting way for beginners to unleash their creativity. This application can be implemented similar as the work of ([Yupeng Zhang, 2013](#)), focusing on the hand as an avatar instead of the body as an avatar. Using the techniques suggested in this project to rotate the objects could be used for rough positioning, or a handlebar metaphor could be implemented. The main drawback for Leap Motion is that it is exhausting for users to use the device without resting their elbows on a table. Finding ways to rotate, move, and scale objects by resting the elbows a table, or a desk is important if a 3D modeling application using Leap Motion as the control interface should be used for longer periods of time.

Implementing different strategies for rotating objects in 3D virtual space can also be useful,

as the strategies implemented in this project proved to be difficult to master. After the user tests, a different technique for rotating was thought of, which is similar to how 2D applications rotate the scene using a pen. It could be possible to point on objects to select them, and use the finger as a 2D pointer to rotate the object. Sliding the finger to the right would rotate object on the y-axis. Sliding the finger upward would rotate the object by the z-axis. A similar example is how you rotate a marble by putting your palm on top of it, and slide your arm in any direction. Future work would be to implement this strategy, and let users test it too see if it intuitive and easy to rotate objects using that technique.

Even though rotations was the most difficult task for transforming objects, there were also a minor problem with translating them. Positioning objects was done very easily with all the strategies, but it was more difficult to position objects correctly in on the z-axis. implementing an atmospheric effect (tinting the objects lighter and reducing their alpha value the further away from the user the objects appeared) may have improved z-axis positioning. As head mounted tracking display such as Oculus VR(<http://www.oculusvr.com/>, accessed 27-05-2014) are being cheaper and more easily obtainable by end-users, it could be interesting to see if a head mounted tracking device can improve the user's perception of depth information, thus performing actions in 3D better. Maybe the z-axis alignment from the user tests could be improved using a head mounted tracking display.

7.5 Summary

In this project, an implementation of an interactive application that focus on how to do homogeneous transformations using Leap Motion as the interaction interface have been made. A literature study was conducted in the start of the project to get an idea of what had already been done, and to get ideas on how to implement the software in this project. An evaluation process guided by the DECIDE framework and the GQM model was used in order to evaluate the project. After implementing a prototype, a pilot test was followed by an extensive user test after improving the prototype. The results from the user test have been quantified, and it was discovered that none of the rotation techniques are suitable for working with for longer periods of time. It also takes too long time to complete a rotation with the strategies implemented in this

project compared to a how fast a user is able to rotate objects using a computer mouse. Even though the suggested techniques for rotating and translating objects did not prove to be easy, it is still worth noting that the techniques might be more useful for making freeform 3D models with a more creative touch where precision is not important. Leap Motion is still being developed, and motion tracking is not yet applicable for replacing mouse and keyboard for accurate 3D transforms. This research might give insight to others who want to explore tracking devices and 3D modeling, and understand how tracking devices may be used more easily for users.

Appendix A

Pilot test questionnaire

The questions from the first pilot test were made using one of Google's collaborative online applications named Form. This appendix cover all the questions from the pilot test. One question was modified for the second series of user tests, and one question were added. The question: "I found the controls intuitive." was modified to "I found the controls easy to understand.". As an additional question, "I needed to learn a lot of things before I could get going with the controls." was added. The questionnaire for the second series of user tests were the same as for the pilot test with the exception of the two questions just mentioned.

Pilot test

Personal data

Age

- 10-19
- 20-29
- 30-39
- 40-49
- 50-59
- 60-69
- 70+

Gender

- Female
- Male

Left or right handed?

- Left handed
- Right handed
- Both

Pilot test

Multiple choice

How comfortable was it to use the different rotation controls?

	1 - Not comfortable at all	2	3	4	5 - Very comfortable
Copy hand rotation	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Joystick rotation	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

I would imagine that most people would learn to use the controls very quickly.

	1 - Strongly disagree	2	3	4	5 - Strongly agree
Copy hand rotation	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Joystick rotation	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

I found the controls unnecessarily complex.

	1 - Strongly disagree	2	3	4	5 - Strongly agree
Copy hand rotation	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Joystick rotation	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

I thought the controls were easy to use.

	1 - Strongly disagree	2	3	4	5 - Strongly agree
Copy hand rotation	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Joystick rotation	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

I found the controls very cumbersome to use.

	1 - Strongly disagree	2	3	4	5 - Strongly agree
Copy hand rotation	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Joystick rotation	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

I felt very confident using the controls.

	1 - Strongly disagree	2	3	4	5 - Strongly agree
Copy hand rotation	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Joystick rotation	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

I found the controls intuitive.

	1 - Strongly disagree	2	3	4	5 - Strongly agree
Copy hand rotation	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Joystick rotation	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Pilot test

optional

Do you have suggestions for different type of controls?

how would you rotate or move objects?

My e-mail is

(optional)

Appendix B

Project schedule

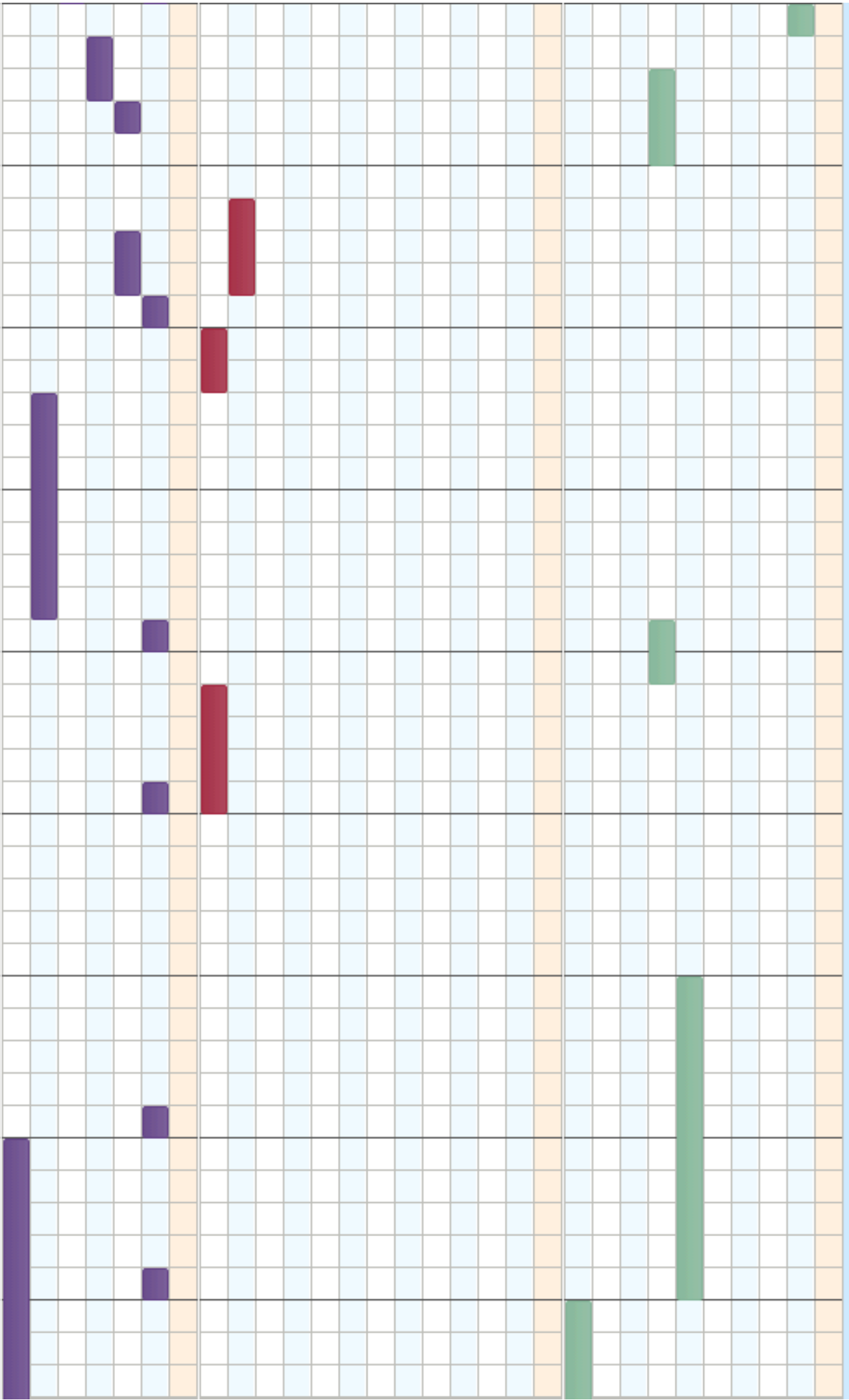
The schedule for this project lasted for 20 weeks, with 40 hours of work each week. An application called Tom's planner was used as a Gantt diagram to create working tasks and time estimates. The application can be found on this url: <http://www.tomsplanner.com/>

March 2014

2014

April 2014

03 04 05 06 07 10 11 12 13 14 17 18 19 20 21 24 25 26 27 28 31 01 02 03 04 07 08 09 10 11 14 15 16 17 18 21 22 23 24 25 28 29 30

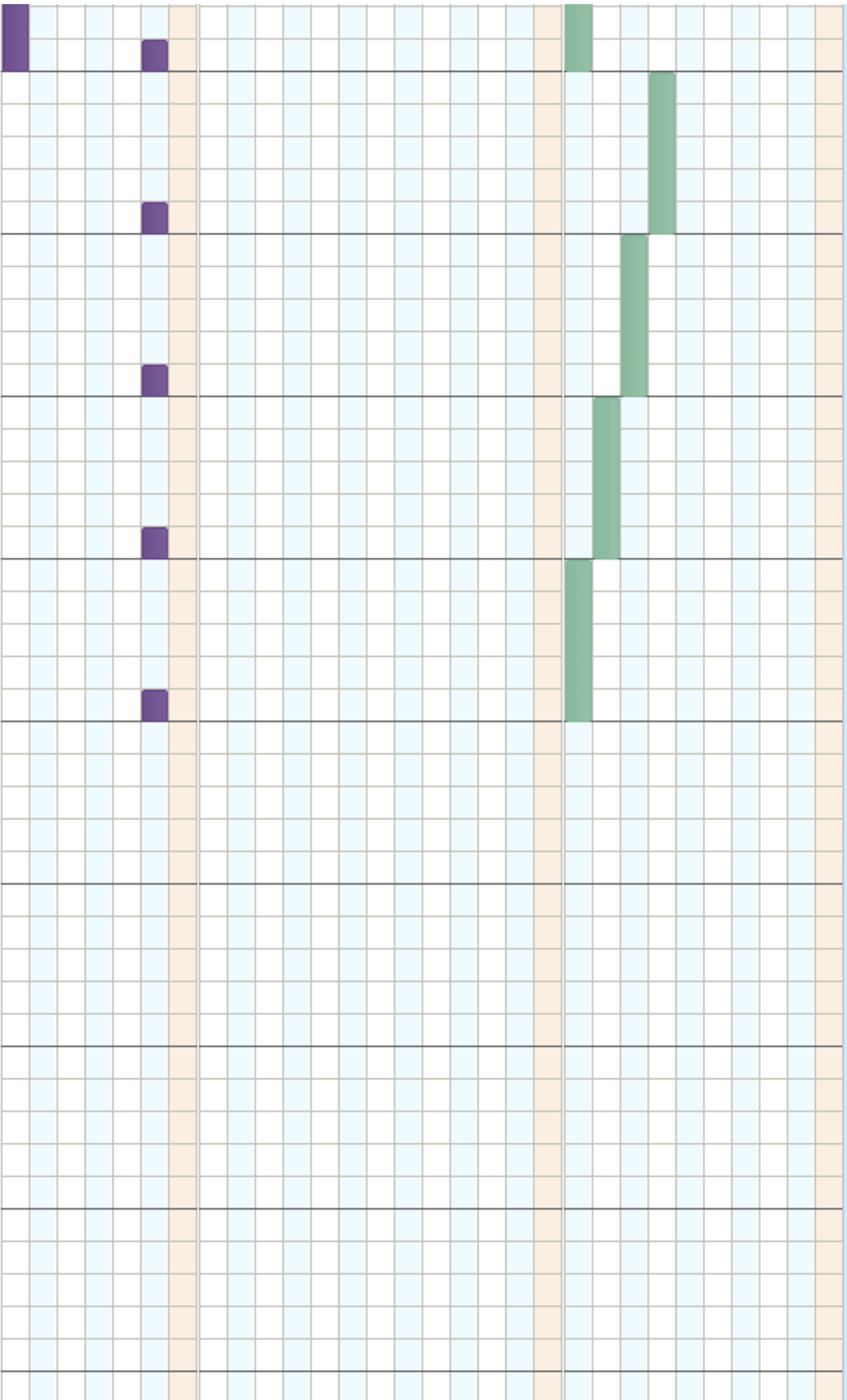


2014

May 2014

June 2014

01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30



Bibliography

(cited, June 2014). Blender. <http://www.blender.org/>.

(cited, June 2014). Kinect. <http://www.xbox.com/en-US/Kinect>.

(cited, June 2014). Leap motion. <https://www.leapmotion.com/>.

(cited, June 2014). Maya. <http://www.autodesk.com>.

(cited, June 2014). Mdpi: Mdpi - open access publishing. <http://www.mdpi.com/>.

(cited, June 2014). Nintendo. <http://www.nintendo.com>.

(cited, June 2014). Playstation. <http://us.playstation.com/>.

Atila Ertas, J. C. J. (1996). *The Engineering Design Process*. John Wiley & Sons, Inc.

Barr, A. (1981). Superquadrics and anglepreserving transformations. *IEEE Computer Graphics and Applications*, 1(1):11–23.

Basili, V. R., Caldiera, G., and Rombach, H. D. (1994). The goal question metric approach. In *Encyclopedia of Software Engineering*. Wiley.

Bass, L. (2013). *Software architecture in practice*. Addison-Wesley, Upper Saddle River, NJ.

Bökesoy, S. (2013). Development of a morphing tool for the cosmosf synthesizer. *JIM13 (Journées Informatique Musicale), Paris 2013*.

Brook, J. (1996). Sus - a quick and dirty usability scale.

Bruno R. De Araújo, G. C. and Jorge, J. A. (2012). Mockup builder: Direct 3d modeling on and above the surface in a continuous interaction space.

- Burton Jr, R. (2013). Motion controlled graphics applications. Senior project, California Polytechnic State University.
- H. Nishimura, M. Hirai, T. K. T. K. I. S. K. O. (1985). Object modeling by distribution function and a method of image generation. *Electronics Communications Conference*, pages 718–725.
- Hill, T. and Westbrook, R. (1997). Swot analysis: it's time for a product recall. *Long range planning*, 30(1):46–52.
- Hiroaki Nishino, Kouichi Utsumiya, K. K. (1998). 3d object modeling using spatial and pictographic gestures. *VRST*.
- Hultquist, J. (1990). A virtual trackball. In Glassner, A., editor, *Graphics Gems*, pages 462–463. Academic Press.
- Jia Sheng, Ravin Balakrishnan, K. S. (2006). An interface for virtual 3d sculpting via physical proxy.
- Johan Klokhammer Helsing, J. E. R. (2013). Leap motion in first-person perspective games. Autumn project at the Norwegian University of Science and Technology.
- Ken Hinckley, Joe Tullio, R. P. D. P. N. K. (1997). Usability analysis of 3d rotation techniques.
- Kenrick Kin, Tom Miller, B. B. T. D. B. H. and Agrawala, M. (2011). Eden: A professional multitouch tool for constructing virtual organic environments. *In Human Factors in Computing Systems (CHI)*.
- Kevin T. McDonnell, H. Q. and Wlodarczyk, R. A. (2001). Virtual clay: A real-time sculpting system with haptic toolkits.
- Krutchen, P. (1995). Architectural blueprints - the “4+1” view model of software architecture. *IEEE Software*.
- Livingston, M. A., Sebastian, J., Ai, Z., and Decker, J. W. (2012). Performance measurements for the microsoft kinect skeleton. *In Virtual Reality Workshops (VR), 2012 IEEE*, pages 119–120. IEEE.

- M. Chen, S. J. Mountford, A. S. (1988). A study in interactive 3-d rotation using 2-d control devices. In *Computer Graphics*, pages 121–129.
- Madden, R. J. (2012). Analysis and comparison of kinect, wii remote, and playstation move.
- Maslah, M. and Milgram, P. (2000). Measuring the allocation of control in a 6 degree-of-freedom docking experiment. In *Human Factors in Computing Systems (CHI)*, pages 25–32.
- Metz, R. (2013). Leap motion's struggles reveal problems with 3-d interfaces. <http://www.technologyreview.com/news/518721/leap-motions-struggles-reveal-problems-with-3-d-interfaces/>.
- Oppenheim, A. V., Schafer, R. W., Buck, J. R., et al. (1999). *Discrete-time signal processing*, volume 5. Prentice Hall Upper Saddle River.
- Pavlovych, A. and Stuerzlinger, W. (2009). The tradeoff between spatial jitter and latency in pointing tasks. In *Proceedings of the 1st ACM SIGCHI symposium on Engineering interactive computing systems*, pages 187–196. ACM.
- Peng Song, Wooi Boon Goh, W. H. C.-W. F. X. L. (2012). A handle bar metaphor for virtual object manipulation with mid-air interaction. *CHI*.
- R. Schmidt, B. Wyvill, M. C. S. J. A. J. T. I. (2005). Shapeshop: Sketch-based solid modeling with blobtrees.
- Rivera, M. L., Badler, N. I., and Normoyle, A. (2013). Project paalm: Phalangeal angle approximation through the leap motion controller. <http://mikeriv.com/>.
- Robert Y. Wang, Sylvain Paris, J. P. (2011). 6d hands: Markerless hand tracking for computer aided design.
- Sean Follmer, H. I. (2012). kidcad: Digitally remixing toys through tangible tools. *CHI*.
- Seok-Hyung Bae, Ravin Balakrishnan, K. S. (2008). Ilovesketch: As-natural-as-possible sketching system for creating 3d curve models.
- Shalloway, A. (2011). *Demystifying Kanban*. Net Objectives, Inc., 1 edition.

- Sharp, H., Rogers, Y., and Preece, J. (2007). *Interaction Design: Beyond Human-Computer Interaction*. Wiley, 2 edition.
- Shoemake, K. (1992). Arcball: A user interface for specifying three-dimensional orientation using a mouse. In *Graphics Interface*, pages 151–156.
- Sims, C. (2013). *Scrum: a Breathtakingly Brief and Agile Introduction*. Dymaxicon.
- Sommerville, I. (2010). *Software Engineering*. Addison-Wesley, Harlow, England, 9 edition.
- Stålhane, T. (2012). Tdt4242 requirements and test. lectures from the courses.
- Steven Schkolne, Michael Pruett, P. S. (2001). Surface drawing: Creating organic 3d shapes with the hand and tangible tools. *CHI*.
- Svanæs, D. (2012). Brukbarhetstesting. Lecture at the Norwegian University of Science and Technology.
- Swink, S. (2009). *Game feel: a game designer's guide to virtual sensation*. Taylor & Francis US.
- Takeo Igarashi, Satoshi Matsuoka, H. T. (1999). Teddy: A sketching interface for 3d freeform design. *ACM SIGGRAPH*.
- Tanaka, K., Parker, J., Baradoy, G., Sheehan, D., Holash, J. R., and Katz, L. (2012). A comparison of exergaming interfaces for use in rehabilitation programs and research. *Loading...*, 6(9).
- Vikram, S., Li, L., and Russell, S. (2013). Writing and sketching in the air, recognizing and controlling on the fly. In *CHI'13 Extended Abstracts on Human Factors in Computing Systems*, pages 1179–1184. ACM, ACM New York, NY, USA.
- W, B. (1995). Touch, gesture & marking. In R. Baecker, J. Grudin, W. B. and Greenberg, S., editors, *In Readings in Human Computer Interaction: Toward the Year 2000*. Morgan Kaufmann Publishers.
- Weichert, F., Bachmann, D., Rudak, B., and Fisseler, D. (2013). Analysis of the accuracy and robustness of the leap motion controller. *Sensors*, 13(5):6380–6393.

Yupeng Zhang, Teng Han, Z. R. N. U. X. T. Y. L. T. S. X. C. (2013). Bodyavatar: Creating freeform 3d avatars using first-person body gestures. *UIST*.

Zigelbaum, J., Browning, A., Leithinger, D., Bau, O., and Ishii, H. (2010). G-stalt: A chirocentric, spatiotemporal, and telekinetic gestural interface. In *Proceedings of the Fourth International Conference on Tangible, Embedded, and Embodied Interaction, TEI '10*, pages 261–264, New York, NY, USA. ACM.

Curriculum Vitae

Name: **John Edvard Reiten**
Gender: Male
Date of birth: 03 April 1989
Nationality: Norway
Email (1): johnedva@stud.ntnu.no
Email (2): johnedvard89@gmail.com
Telephone: +47 94426069

