



# NTNU

Kunnskap for en bedre verden

# Bacheloroppgave

**IE303612 Bacheloroppgave i automatisering/data  
AI & Interactive Art**

803, 805, 820

Totalt antall sider inkludert forsiden: 135

Antall vedlegg: 10

Innlevert Ålesund, 27.05.2016

## Obligatorisk egenerklæring/gruppeerklæring

Den enkelte student er selv ansvarlig for å sette seg inn i hva som er lovlige hjelpemidler, retningslinjer for bruk av disse og regler om kildebruk. Erklæringen skal bevisstgjøre studentene på deres ansvar og hvilke konsekvenser fusk kan medføre. **Manglende erklæring fritar ikke studentene fra sitt ansvar.**

Du/dere fyller ut erklæringen ved å klikke i ruten til høyre for den enkelte del 1-6:		
1.	Jeg/vi erklærer herved at min/vår besvarelse er mitt/vårt eget arbeid, og at jeg/vi ikke har brukt andre kilder eller har mottatt annen hjelp enn det som er nevnt i besvarelsen.	<input checked="" type="checkbox"/>
2.	Jeg/vi erklærer videre at denne besvarelsen: <ul style="list-style-type: none"><li>• ikke har vært brukt til annen eksamen ved annen avdeling/universitet/høgskole innenlands eller utenlands.</li><li>• ikke refererer til andres arbeid uten at det er oppgitt.</li><li>• ikke refererer til eget tidligere arbeid uten at det er oppgitt.</li><li>• har alle referansene oppgitt i litteraturlisten.</li><li>• ikke er en kopi, duplikat eller avskrift av andres arbeid eller besvarelse.</li></ul>	<input checked="" type="checkbox"/>
3.	Jeg/vi er kjent med at brudd på ovennevnte er å <u>betrakte som fusk</u> og kan medføre annullering av eksamen og utestengelse fra universiteter og høyskoler i Norge, jf. <a href="#">Universitets- og høyskoleloven</a> §§4-7 og 4-8 og Forskrift om eksamen.	<input checked="" type="checkbox"/>
4.	Jeg/vi er kjent med at alle innleverte oppgaver kan bli plagiatkontrollert i Ephorus, se Retningslinjer for elektronisk innlevering og publisering av studiepoenggivende studentoppgaver	<input checked="" type="checkbox"/>
5.	Jeg/vi er kjent med at høyskolen vil behandle alle saker hvor det forligger mistanke om fusk etter NTNUs studieforskrift.	<input checked="" type="checkbox"/>
6.	Jeg/vi har satt oss inn i regler og retningslinjer i bruk av kilder og referanser på biblioteket sine nettsider	<input checked="" type="checkbox"/>



# Publiseringsavtale

Studiepoeng: 20

Veileder: Anniken Karlsen, Arne Styve, Robin Bye

## Fullmakt til elektronisk publisering av oppgaven

Forfatter(ne) har opphavsrett til oppgaven. Det betyr blant annet enerett til å gjøre verket tilgjengelig for allmennheten ([Åndsverkloven §2](#)).

Alle oppgaver som fyller kriteriene vil bli registrert og publisert i Brage med forfatter(ne)s godkjenning.

Oppgaver som er unntatt offentlighet eller båndlagt vil ikke bli publisert.

Jeg/vi gir herved NTNU i Ålesund en vederlagsfri rett til å gjøre oppgaven tilgjengelig for elektronisk publisering:

ja  nei

Er oppgaven båndlagt (konfidensiell)?

ja  nei

(Båndleggingsavtale må fylles ut)

- Hvis ja:

Kan oppgaven publiseres når båndleggingsperioden er over?

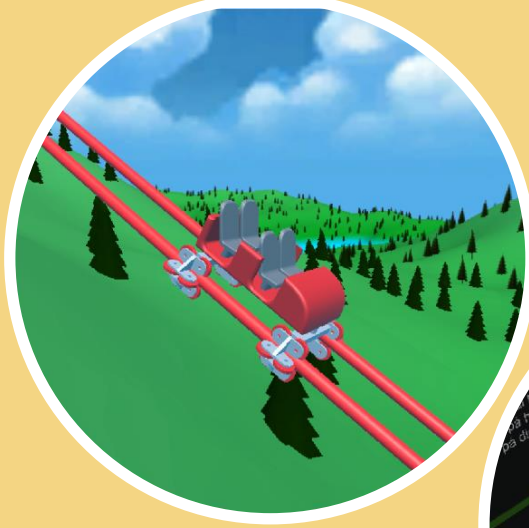
ja  nei

Er oppgaven unntatt offentlighet?

ja  nei

(inneholder taushetsbelagt informasjon. [Jfr. Offl. §13/Fvl. §13](#))

Dato: 27.05.2016



# AI & Interactive Art

Erik Espenakk - Stein Sæter - Terje Eik

The artistic idea behind the project is about the production of social subjectivity and human collectivities within contemporary society. Normally we humans think that we are individuals with free will that can move, think and talk as we want to, but then we notice that there are social strains, and that our movements and whereabouts are strongly affected by others like friends, family and governmental bodies. We are continuously being measured, registered and controlled. Social media, television and electronic networks are all influential to how we interact in different arenas, and may also exert pressure or force upon us to take part in things we might not want.

The thesis investigates this problem using Unity 3D for Oculus Rift and Microsoft Kinect by using Facebook data to generate a terrain, a rollercoaster track and minigames.

Special thanks to our supervisors Anniken Karlsen, Robin T. Bye, Arne Styve and our external assigners Terese Longva and Harald Dyrkorn.

# Abstract

Interactive art is an art form that emerges between the installation and the audience. With the advent of computer-based interactivity in the 1990s, interactive art has become increasingly popular, with programmable machines enabling the creation of unique artwork for each audience. In academic communities within NTNU, interactive art has been incorporated as a problem domain in software development and programming classes to encourage and motivate students and act as a driver for learning.

The artistic idea behind the project is about the production of social subjectivity and human collectivities within contemporary society. Normally we humans think that we are individuals with free will that can move, think and talk as we want to, but then we notice that there are social strains, and that our movements and whereabouts are strongly affected by others like friends, family and governmental bodies. We are continuously being measured, registered and controlled. Social media, television and electronic networks are all influential to how we interact in different arenas, and may also exert pressure or force upon us to take part in things we might not want.

To investigate this an environment has been created in Unity 3D for Oculus Rift. The user is sent through an interactive roller coaster with stations where certain tasks has to be fulfilled before the user can move on, focusing on social subjectivity. Microsoft Kinect is used to implement interaction, detecting the hand gestures and movements to lift up object and move them.

This thesis focuses on the planning and development of the installation, and goes through relevant theory in addition to the methodologies and technologies used. The result of the project is also discussed, comparing the installation to other similar installations found on the Internet.

# Abbreviations

- CSS** Cascading Style Sheets. A language to define the design of HTML pages. Colors and other style information is defined in the CSS.
- CSV** Comma Separated Values. A file type with values split up by a special character, often a comma. Used to store data and often used to import the data in to tables.
- cURL** cURL is a computer software project providing a library and command line tool for transferring data using various protocols, for example HTTP.
- HTML** HyperText Markup Language. The standard markup language to format web pages. To some degree, this could also change the look of the website.
- HMD** Head Mounted Display. A display attached to your head, for example Oculus Rift.
- HTTP** Hypertext Transfer Protocol. An application protocol for the transfer of hypertext content. The foundation of data communication via the World Wide Web.
- IDE** Integrated Development Environment. A software application that facilitates for developers to create new software.
- IxD** Interaction Design.
- PHP** A server-side script language for web development that can be embedded in HTML-code.
- SDK** Software Development Kit. A set of software development tools that allows the creation of software for a certain software or hardware platform.
- TCP** Transmission Control Protocol. Network protocol to transfer data.
- API** Application Programming Interface. A set of routines, protocols and tools for building software and applications.

# Terms

**C#** C#(pronounced as see sharp) is a multi-paradigm programming language encompassing strong typing, imperative, declarative, functional, generic, object-oriented, and component-oriented programming disciplines.

**Git** Source code management system with distributed revision control with an emphasis for speed, data integrity and support for non-linear work flows.

**JavaScript** A script language to create dynamic elements on a website. It can for example be used to modify text, styles, data validation.

**Microsoft Kinect** Kinect is a line of motion-sensing input equipment made by Microsoft. It was originally made for Xbox, but Microsoft has opened up the technology for developers to use it in other ways as well.

**Oculus Rift** The Rift is a head-mounted virtual reality display developed by Oculus. Oculus Rift has one OLED-display for each eye, each display having with a resolution of 1080x1200 resolution on 90 Hz.

**Prefab** Prefabrication. An set of components that are assembled as one to be used several times, for example a set of squares rendering a room in Unity 3D.

**Python** Python is a general-purpose high-level programming language. Emphasises readability and to express concepts in as few lines as possible.

**Unity 3D** Unity is a cross-platform game engine often used to create interactive 2D and 3D content.

**WalkMouse** A brand of omnidirectional treadmills, a mechanical device similar to a regular treadmill but allows movement in any direction. Used in virtual environments to allow movement within the virtual space.

## Table of Contents

<b>Abstract</b>	<b>v</b>
<b>Abbreviations</b>	<b>vi</b>
<b>Terms</b>	<b>vii</b>
<b>List of Figures</b>	<b>xi</b>
<b>List of Tables</b>	<b>xiii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Background . . . . .	1
1.2 Project Description . . . . .	1
1.3 Scope . . . . .	2
1.4 Thesis Structure . . . . .	2
<b>2 Theory</b>	<b>3</b>
2.1 Art . . . . .	3
2.1.1 Traditional Art . . . . .	3
2.1.2 Digital Art . . . . .	4
2.1.3 Comparing Traditional Art & Digital Art . . . . .	4
2.2 Software Development . . . . .	5
2.2.1 Agile Software Development . . . . .	5
2.2.2 Scrum . . . . .	6
2.2.3 Version Control . . . . .	7
2.3 Path Generation . . . . .	8
2.3.1 Cubic Bézier Curves . . . . .	8
2.3.2 Catmull-Rom Splines . . . . .	9
2.4 Procedural Terrain Generation . . . . .	9
2.4.1 Terrain Representation . . . . .	10
2.4.2 Diamond Square Algorithm . . . . .	10
2.4.3 Perlin Noise . . . . .	11
2.4.4 Fractional Brownian Motion . . . . .	11
2.5 Interaction Design . . . . .	13
2.5.1 The Five Dimensions of Interaction Design . . . . .	13
2.5.2 Best Practices . . . . .	13
2.6 Interaction Technologies . . . . .	15
2.7 Computer Graphics . . . . .	17
2.8 Swarm Behavior . . . . .	18
2.9 Laws and Regulations . . . . .	18
2.9.1 Personal Data Act ( <i>Personopplysningsloven</i> ) . . . . .	18
<b>3 Method</b>	<b>20</b>
3.1 Project Organization . . . . .	20
3.2 Development . . . . .	20

3.2.1	Methodology . . . . .	20
3.2.2	Tools . . . . .	21
3.3	Resources . . . . .	24
3.3.1	Software . . . . .	24
3.3.2	Libraries . . . . .	25
3.3.3	Hardware . . . . .	28
<b>4</b>	<b>Results</b>	<b>29</b>
4.1	System Architecture . . . . .	29
4.2	Solution . . . . .	29
4.2.1	Web Login . . . . .	29
4.2.2	Information Retrieval From Facebook . . . . .	31
4.2.3	Main Menu . . . . .	31
4.2.4	Terrain Generation . . . . .	32
4.2.5	Track Generation . . . . .	34
4.2.6	Interactivity via Microsoft Kinect . . . . .	38
4.2.7	Stations . . . . .	38
4.3	Development Process . . . . .	41
4.3.1	Statistics . . . . .	41
4.3.2	Sprint Reports . . . . .	41
4.4	Information Security . . . . .	42
4.4.1	Personal Data . . . . .	42
4.4.2	Authentication . . . . .	42
<b>5</b>	<b>Discussion</b>	<b>44</b>
5.1	Engineering Approach vs. Artistic Approach . . . . .	44
5.1.1	Exploration . . . . .	44
5.1.2	Customer Interaction . . . . .	46
5.2	Development . . . . .	47
5.2.1	Guidance and Meetings . . . . .	48
5.2.2	Applications and Tools . . . . .	48
5.3	Platform . . . . .	49
5.3.1	Game Engine . . . . .	49
5.3.2	Peripherals . . . . .	49
5.4	Terrain Generation . . . . .	51
5.4.1	Plane . . . . .	51
5.4.2	Noise Functions . . . . .	51
5.4.3	Brownian Plane . . . . .	52
5.4.4	Optimization . . . . .	52
5.4.5	Infinite Terrain . . . . .	54
5.4.6	Coloring . . . . .	57
5.4.7	Processing Facebook data . . . . .	57
5.5	Track Generation . . . . .	58
5.6	Software Related Challenges . . . . .	58
5.6.1	Programming Languages . . . . .	58
5.6.2	Choosing Type of Splines . . . . .	59

5.6.3	3D Models vs. 2D Sprites . . . . .	59
5.7	Further Work . . . . .	60
5.7.1	Find new uses of the Facebook data . . . . .	60
5.7.2	Generate More Stations . . . . .	61
5.7.3	Interaction Technology . . . . .	61
5.7.4	User Testing . . . . .	63
5.7.5	Optimize Code For Better Graphics . . . . .	63
5.7.6	Ambient Sounds . . . . .	63
5.7.7	Port For Use In The Visualization Lab . . . . .	64
5.7.8	Input From Microphone . . . . .	64
5.7.9	Visual Facebook Timeline . . . . .	65
5.7.10	Appropriate Objects Populating Terrain . . . . .	65
<b>6</b>	<b>Conclusion</b>	<b>66</b>
	<b>Appendices</b>	<b>69</b>
<b>A</b>	<b>- Hardware Requirements</b>	<b>70</b>
<b>B</b>	<b>- Available Facebook data</b>	<b>72</b>
<b>C</b>	<b>- System Documentation</b>	<b>73</b>
<b>D</b>	<b>- User Manual</b>	<b>75</b>
<b>E</b>	<b>- Installation Guide</b>	<b>80</b>
<b>F</b>	<b>- Sketches</b>	<b>84</b>
<b>G</b>	<b>- Preliminary Report</b>	<b>87</b>
<b>H</b>	<b>- Meeting Reports</b>	<b>106</b>
<b>I</b>	<b>- Sprint Reports</b>	<b>112</b>
<b>J</b>	<b>- Project Poster</b>	<b>121</b>



## List of Figures

1	The Moschophoros (Calf-Bearer), in the Acropolis Museum. [6]	3
2	A 2D Bézier curve.	9
3	Example of a Catmull-Rom spline. [15]	10
4	Example of the diamond square algorithm (source: Wikipedia).	11
5	Example of Perlin noise. [17]	12
6	Fractional Brownian Motion based heightmap using Perlin noise	12
7	The five dimensions of interaction design. Copyright holder: Samir Dash.	14
9	A swarm of birds (source: Wikipedia).	19
10	The Scrum Process (figure from zenexmachina.com).	20
11	Platforms supported by Unity.	21
12	Example of a query to Facebook Graph Explorer.	22
13	Example of the retrieval of an access token within Facebook Graph.	23
14	An example menu in the No Touch GUI asset pack.	27
15	System Overview.	29
16	Screenshot from the web login page.	30
17	The main menu.	32
18	Generated environments	34
19	Gizmos.	35
20	Triangle count.	36
21	Mesh generation example	37
22	Screenshot from Unity showing Kinect interactivity.	39
23	Screenshot from the word cloud scene.	40
24	Git: Activity Graph from the Facebook Graph API repository.	41
25	Git: Activity Graph from the Unity repository.	41
26	Error message if the Facebook login is not as declared.	43
27	Example of the IntelliSense in Visual Studio.	48
28	WalkMouse in use, image from WalkMouse.com.	50
29	Unity's built in plane prefab.	51
30	Fractional Brownian Motion based terrain generation	53
31	Diagram showing the generation procedure	55
32	UML diagram showing components and relations to TerrainGenerator.	56
33	Mesh "y-coordinate" based coloring.	57
34	Gradient for terrain coloring	57
35	Distribution of programming languages within the Git repository.	59
36	3D model of a tree and LOD-groups in Unity.	60
37	The HTC Vive in action, image courtesy of PCWorld.	63
38	The visualization lab at NTNU Ålesund.	64
39	System Overview	73
40	System Architecture	74
41	Popup showing what the app wants access to.	75
42	Success - Credentials downloaded.	76
43	The ArtApp icon.	76
44	The main menu.	77

45	Kinect hand gestures – Closed (left), lasso (middle) and open (right).	78
46	Editing the hosts-file. . . . .	80
47	Finding the startup folder via the Run-function in Windows. . . . .	81
48	Sketch of a social forest. . . . .	84
49	Sketch of a graph showing when during the day your friends likes your posts. . . . .	85
50	Sketch of a graph showing the most active friends on your posts. . . . .	86

## List of Tables

1	Computer specifications . . . . .	28
2	List of other hardware equipment . . . . .	28
3	Parameters used from Facebook. . . . .	31
4	Hand gestures for the claw. . . . .	38
5	Hand gestures for the flashlight. . . . .	38
6	Hand gestures for the hand. . . . .	38
7	Requirements for Oculus Rift - Consumer Edition. . . . .	70
8	Requirements for Oculus Rift - DK2. . . . .	70
9	Requirements for Microsoft Kinect 2. . . . .	71
10	PC from Komplet.no . . . . .	71
11	Collection of parameters available from Facebook. . . . .	72
12	Hand gestures for the claw. . . . .	79
13	Hand gestures for the flashlight. . . . .	79
14	Hand gestures for the hand. . . . .	79

# 1 Introduction

## 1.1 Background

Throughout history art has been a medium for people to express their ideas, meanings or messages through a creative expression. Some of the oldest pieces of art discovered were drawings on a cave wall in Indonesia. According to uranium-thorium dating methods it was made at least 40,000 years ago [1]. As the available materials and methods evolved, so did the art. With the rise of the computers and availability of them, art has moved onto the digital platform. This dates all the way back to the 1960s and has been an interesting intersection between the two different fields [2]. Within the academic community, especially within NTNU, there is a great interest for using digital art to encourage students to learn programming. Anniken Karlsen has used this in her Software Engineering classes and has studied the results of the approach with Robin T. Bye.

Local artists Harald Dyrkorn and Terese Longva are the external assigners in this project. Terese Longva holds a Master of Fine Art from Kunst- og designhøgskolen i Bergen (Bergen Academy of Art and Design). She is an interdisciplinary artist that looks at different aspects of the human condition such as: how we construct different selves in reaction to varied expectations of society and how we constantly struggle to learn our social roles and at the same time break out of them to know ourselves. She also investigates how the new appears through emergent and self-organizing processes. Longva is based in Bergen and partly at Longva. Harald Dyrkorn is a social scientist. He is a bachelor in social anthropology from The University of Bergen. His interests are how we relate to our surroundings and our social management of objects (things). Since 2012 he has been collaborating with Terese Longva. Longva and Dyrkorn share a common background from neighbouring island communities a few miles north of the town of Ålesund. They also share research interests in how social subjectivity is changing through processes similar to dynamic complex systems with emergence and self-organization.

## 1.2 Project Description

The aim of the project has been to create a work of art that gives the audience a feeling of how our behaviour is influenced by others, and often in surprising ways. The initial idea was to create a computer game or interactive hypermedia of any kind, suitable for use with *Den Kulturelle Skolesekken* in 10th grade of secondary school (*ungdomsskole 10. trinn*) and also in highschool (*videregående skole*). *Den Kulturelle Skolesekken* is a national initiative to help Norwegian students meet professional art and culture of all kinds. The project is a collaboration

between the Ministry of Education (*Kunnskapsdepartementet*) and the Ministry of Culture (*Kulturdepartementet*) [3].

The intention was that the artwork should take form as a collaboration between the artists and the project group and should emphasize on a social experience. The thought behind the project was that there is a bigger social subjectivity within society that expresses itself through individuals, much like an echo. The project group were given a lot of freedom to choose technologies, they were also tasked with navigating most of the creative process.

### 1.3 Scope

The scope of this thesis is limited to the exploration and creation of the interactive art installation. Because this is a creative process, the project is not clearly defined. The creative process can and will change the specifications and goals underway. Also, there almost always ways to develop such an installation further, in other words: There is no final form of the installation.

If the project is a success it may be used by the artists at schools to give students an impression of what art *could look like*, and show that it is not just paintings and sculptures. The possibility of featuring the project in *Den Kulturelle Skolesekken* has not set any requirements when it comes to the solution, but could be an alternative depending on the result when the project is finalized and what equipment is used. For this to be considered the art installation would have to be portable to some degree and should be possible to set up by a person alone.

### 1.4 Thesis Structure

This section will give an overview of the project, including background, description of the project, goals and the scope of the thesis.

Section 2 will follow up with relevant theory concerning art in general, software development, path and terrain generation, interaction design and swarm behavior before the ending with relevant laws and regulations.

In section 3 the method of work is described. Some of the topics are the organization within the project, how choices were made, principles applied and what resources were used (hardware, software and licenses).

The results of the project will be presented in section 4. We will discuss our solution, the design, system architecture and the development process.

Whether we reached our goals or not will be discussed in section 5.

Section 6 contains final words and we will try to tie together the loose ends.

## 2 Theory

### 2.1 Art

#### 2.1.1 Traditional Art

Because of its subjective nature, art is a term that is difficult to define. According to Oxford Dictionaries, art is defined as *"the expression or application of human creative skill and imagination, typically in a visual form such as painting or sculpture, producing works to be appreciated primarily for their beauty or emotional power"*[4]. Humanity has always had a need to create, and art has been a part of our history for thousands of years. Many of the traditions in art has roots back to the great ancient civilizations; from Ancient Greece to Ancient Egypt. Typically, the bigger the civilization was the more art could be recovered.

Each of the civilizations developed unique styles of art. In Ancient Greece artists developed their skills to show musculature and correct human anatomy. An example of this is the sculpture pictured below, called the Moschophoros (Calf-Bearer). This sculpture dates back to ca. 560 BC and was originally around 1,65 meters in height. According to the inscription on the base of the sculpture the statue shows the artist himself bringing an offering to the Greek goddess Athena [5].



Figure 1: The Moschophoros (Calf-Bearer), in the Acropolis Museum. [6]

Traditional arts are often categorized according to the technique or medium used, unlike fields of science where the meaning or subject is the category, for example biology or chemistry. The **art form** is the specific shape that the artistic expression takes. This could for example a sculpture. The sculpture as a form must have three-dimensional shape and respond to gravity, giving it limitations as a medium. These limitations are called the **formal qualities** of the artwork. A **genre** is a set of styles and conventions that has the same properties. An example of this is painting, which has since the 17th century referred to painting scenes of every day life. The **style** of an artwork or artist is the distinct method or form followed by the art piece. An example of this is loose brushy, dripped or poured abstract paintings which are called *expressionistic* [7].

### 2.1.2 Digital Art

Digital art encountered a lot of resistance in its earlier days. Established artists and the art community were hesitant to accept digital art as a true form of art [8]. But after its initial struggle, digital art is now one of the most exciting art forms out there [9]. Digital art uses technology as an essential part of the creative process or presentation. There are many subgenres of digital art – including digital photography, fractal art, immersion (virtual reality), interactive art and motion graphics.

Digital art has changed traditional art forms like sculpturing, painting and drawing in addition to creating new art forms like digital art installations, virtual reality and Internet art. In later years with the birth of products like the HTC Vive, Oculus Rift and Google Cardboard virtual reality technology has become publicly available. Artistic tools are also being released as the technology progresses. Google has released a software for the HTC Vive called Tilt Brush – a virtual reality drawing application that melts genres of sculpturing and drawing with digital tools ([www.tiltbrush.com](http://www.tiltbrush.com)).

Because of the attention virtual reality is getting from big companies it is reason to believe that the field of virtual reality art will continue to grow even more in the future as the technology gains momentum and popularity.

### 2.1.3 Comparing Traditional Art & Digital Art

Traditional works of art can be thought of as static pieces of art. The expression is made by the artist and the visual representation is not changed after the initial birth of the artwork (except for wear and tear). In some digital arts, for example within immersive art, the content can be more dynamic. The artwork can change through time as defined in the artwork itself by the artist, or by input from the audience in the form of audio, motion or using controllers. This

can generate a new dimension of artistic impression giving different audiences different experiences.

Formal qualities (limitations) of a digital artwork are different than the formal qualities of a traditional piece of art. A traditional painter's canvas will have limitations, these limitations are inherent from the physical laws we live by. These limitations govern our expectations of what can be done on a canvas. For digital art these limitations are more abstract and less obvious. Recognizing the limitations of programming, what it can or can not do for an artwork, requires experience and knowledge. This is not just a limitation for art projects, but for all software projects in general.

Seeing the possibilities of digital art can be difficult in the same way as seeing the limitations. Better hardware for lower prices in addition to new technologies becoming available gives artists/developers new ways to express their message. Knowing what technology is available and how it can be used is a big part of developing digital art and requires both artistic and technological skills.

## 2.2 Software Development

The development of software is a process that consists of programming, documenting the work, testing and bug fixing. This is a continuous work process in the software's life cycle. According to Software Engineering by Ian Sommerville, good software should deliver the required functionality and performance to the user and should be maintainable, dependable, and usable [10].

In the following sections some methodologies of software engineering will be discussed, focusing on agile software development and Scrum in particular.

### 2.2.1 Agile Software Development

Traditionally software development projects has had an incredible rate of failure. According to a study presented by McKinsey & Company, 17% of large IT projects go so badly that they can threaten the very existence of the company [11]. The same study says that on average, large IT projects run 45% over budget and 7% over time, while delivering 56% less value than predicted. There is no easy answer to explain why this is happening, but a combination of issues makes software development a challenge. A lack of standard with all the different programming languages (Java, C++, C#, PHP, Python, Scala, Ruby, Groovy, Objective-C, Swift, JavaScript to name some) is one of the reasons. Incompatible platforms is another. If applications are developed for Microsoft Windows there is no easy way to export the code and get it to run perfectly on Apple OSX. Another reason is the customer not knowing what he/she wants or needs, or understanding the limitations of developing software. If you want to build a bridge over a river, most people understand there must be pillars



that support the construction of the bridge – if not the bridge would collapse. To understand the same limitations within software development is not as easy for people without a deeper knowledge of programming, and this is one of the pitfalls with software projects. Understanding the requirements of the projects and defining these unambiguously on paper is therefore very important, as well as documenting who will take the cost if the project has to be expanded or changed.

Agile software development is a response to the traditional ways of handling software development and helps cope with the challenges of software development. It helps teams respond to unpredictability through incremental, iterative work cadences, known as sprints. Agile methodologies involves the customer underway in the project to overcome communication issues by having acceptance tests and demonstration of the different iterations of the software.

The philosophy of agile software development is reflected in the agile manifesto. The agile manifesto has many principles, but can be summarized in four core values:

1. **Individuals and interactions** over processes and tools
2. **Working software** over comprehensive documentation
3. **Customer collaboration** over contract negotiation
4. **Responding to change** over following a plan

### 2.2.2 Scrum

Scrum is an agile framework for developing and sustaining complex products. It is a lightweight framework that is very simple to understand, but very difficult to master. According to the Scrum Guide, Scrum has been used to manage complex product development since the 1990s [12]. The framework is based on empirical process control theory, or empiricism. Empiricism asserts that knowledge comes from experience and making decisions based on what is known. Every implementation of empirical process control has three pillars that uphold them: transparency, inspection and adaptation.

The definition of the Scrum framework is having a team over developers, ideally 3-9 people in size – the same size as a family. This is not a coincidence since having more people than this could lead to the group becoming too big and some voices would not be heard in the crowd. The developers should work towards a common goal without having a leader pressing down on them and trying to push them to work faster than they ideally work. They should have mixed backgrounds and relevant knowledge. The work is iterated through sprints where different goals and tasks defined, taken from a product backlog.

## Scrum Glossary

The following glossary contains key definitions within Scrum gathered from [scrum.org](http://scrum.org). The full glossary can be viewed at: [www.scrum.org/Resources/Scrum-Glossary](http://www.scrum.org/Resources/Scrum-Glossary)

**Burn-down Chart** A chart showing the evolution of remaining effort against time. Burn-down charts are an optional implementation within Scrum to make progress transparent.

**Daily Scrum** Daily time-boxed event of 15 minutes, or less, for the Development Team to re-plan the next day of development work during a Sprint. Updates are reflected in the Sprint Backlog.

**Development Team** The role within a Scrum Team accountable for managing, organizing and doing all development work required to create a releasable Increment of product every Sprint.

**Increment** A piece of working software that adds to previously created Increments, where the sum of all Increments -as a whole - form a product.

**Product Backlog** An ordered list of the work to be done in order to create, maintain and sustain a product. Managed by the Product Owner.

**Product Owner** The role in Scrum accountable for maximizing the value of a product, primarily by incrementally managing and expressing business and functional expectations for a product to the Development Team(s).

**Scrum Master** The role within a Scrum Team accountable for guiding, coaching, teaching and assisting a Scrum Team and its environments in a proper understanding and use of Scrum.

**Sprint** Time-boxed event of 30 days, or less, that serves as a container for the other Scrum events and activities. Sprints are done consecutively, without intermediate gaps.

**Sprint Backlog** An overview of the development work to realize a Sprint's goal, typically a forecast of functionality and the work needed to deliver that functionality. Managed by the Development Team.

**Stakeholder** A person external to the Scrum Team with a specific interest in and knowledge of a product that is required for incremental discovery. Represented by the Product Owner and actively engaged with the Scrum Team at Sprint Review.

### 2.2.3 Version Control

Version control is a system that records changes made to files or a set of files. The system can hold different versions of the files to ensure that changes made

by different developers does not interfere with each other. The version control system (typically Git, Subversion, Sourceforge etc.) holds a master copy of the files and the developer holds a local copy on his/hers machine. If the developer wants to send this information to the version control system it can be committed (often called *pushing*) to the version control system. If a developer wants to retrieve information from the system, this can be downloaded (often called *pulling*). A set of files that belong to the same project are often called a *repository* – and these could both be public (everyone can see the source code and copy it) or private (only the creator and invitees have access to the source code).

## 2.3 Path Generation

In the following section we will introduce different theories regarding the generation of curves from coordinate points. Later in the report we will explore them as alternatives for generating a path.

### 2.3.1 Cubic Bézier Curves

A cubic Bézier curve is a parametric curve. The curve is interpolated from four control points. The technique used to generate the curve was discovered by Pierre Bézier. He was working for Renault as head of the design department during the 1960s, and realized that he required visual representations of mechanical parts on a computer. His initial idea was to represent a "basic curve" as the intersection of two elliptic cylinders. The mathematical basis for the Bézier curves is the Bernstein polynomial and has been known since 1912, but that was not discovered until A. R. Forrest studied Bézier's work [13].

The following equation is used to calculate a point on the cubic Bézier curve:

$$B(t) = (1 - t)^3 P_0 + 3(1 - t)^2 t P_1 + 3(1 - t) t^2 P_2 + t^3 P_3 \quad (1)$$

$$\begin{aligned} J_{3,0}(t) &= (1 - t)^3 &= 1(1 - t)^3 t^0 \\ J_{3,1}(t) &= 3(1 - t)^2 t &= 3(1 - t)^2 t^1 \\ J_{3,2}(t) &= 3(1 - t) t^2 &= 3(1 - t)^1 t^2 \\ J_{3,3}(t) &= t^3 &= 1(1 - t)^0 t^3 \end{aligned} \quad (2)$$

Figure 2 shows a visual representation of the curve, showing how a curve looks with four control points.

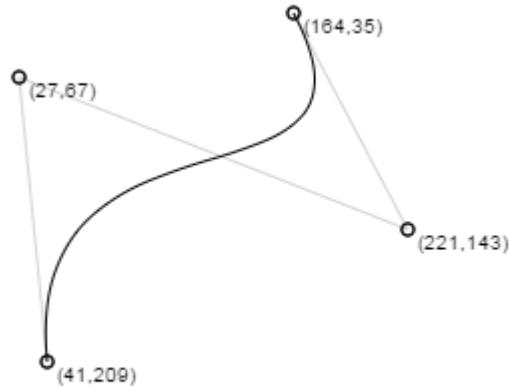


Figure 2: A 2D Bézier curve.

### 2.3.2 Catmull-Rom Splines

A spline is a numeric function which possesses a high degree of smoothness where the polynomial pieces connect. Catmull-Rom splines were formulated by Edwin Catmull and Raphael Rom according to the work of Barry and Goldman. It is an interpolating spline going through four defined control points  $P_0$ ,  $P_1$ ,  $P_2$  and  $P_3$ , only drawing the curve from  $P_1$  and  $P_2$  [14].

Catmull-Rom is mathematically defined as:

$$\begin{aligned}
 q(t) = & 0.5 * ((2 * P_1) + \\
 & (-P_0 + P_2) * t + \\
 & (2 * P_0 - 5 * P_1 + 4 * P_2 - P_3) * t^2 + \\
 & (-P_0 + 3 * P_1 - 3 * P_2 + P_3) * t^3)
 \end{aligned} \tag{3}$$

Figure 3 shows an example of a Catmull-Rom spline. This shows how  $P_0$  and  $P_4$  affects the curve rendered from  $P_1$  to  $P_2$ .

## 2.4 Procedural Terrain Generation

Procedural methods of generating terrain are associated with requiring little to no input from the designers, this also leads to the designers having less control

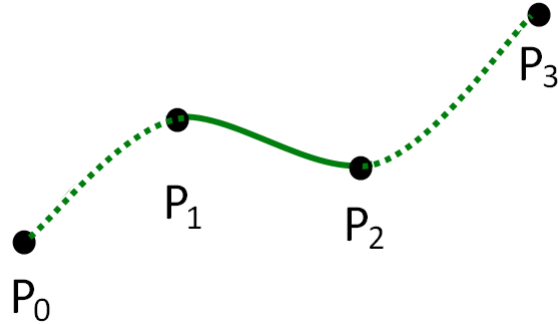


Figure 3: Example of a Catmull-Rom spline. [15]

over the outcome of the generation. Procedural generation methods are well suited for generating realistic terrains automatically, they are often used in games where exploration of interesting terrain is a focus.

#### 2.4.1 Terrain Representation

The most common way of representing a terrain when doing procedural generation is to use a heightmap/heightfield. A heightmap can be mathematically represented as a function that maps the  $x$ - and  $y$ -coordinates to a height  $z$ .

$$z = f(x, y) \quad (4)$$

One of the drawbacks using this form of representation is that one can only have one height value per point on the map. Because of this, one heightmap will not be able to model caves in the landscape since it would require several height values per coordinate point.

#### 2.4.2 Diamond Square Algorithm

The diamond square algorithm is an algorithm for generating heightmaps in computer graphics, introduced by Fournier, Fussell and Carpenter at SIGGRAPH 1982 [16]. The algorithm starts with a 2D grid and generates random terrain heights by filling up the entire grid with squares. The terrain heights are generated from four seed values. Terrains generated using the diamond square algorithm are often realistic looking.

The diamond square algorithm is initialized with a 2D array of size  $2n + 1$ . The four corner points are then set to initial values. The diamond and square steps are then performed alternately until all array values have been set. These

steps are illustrated in Figure 4 where the algorithm is performed on a 5x5 grid until all points has a new value.

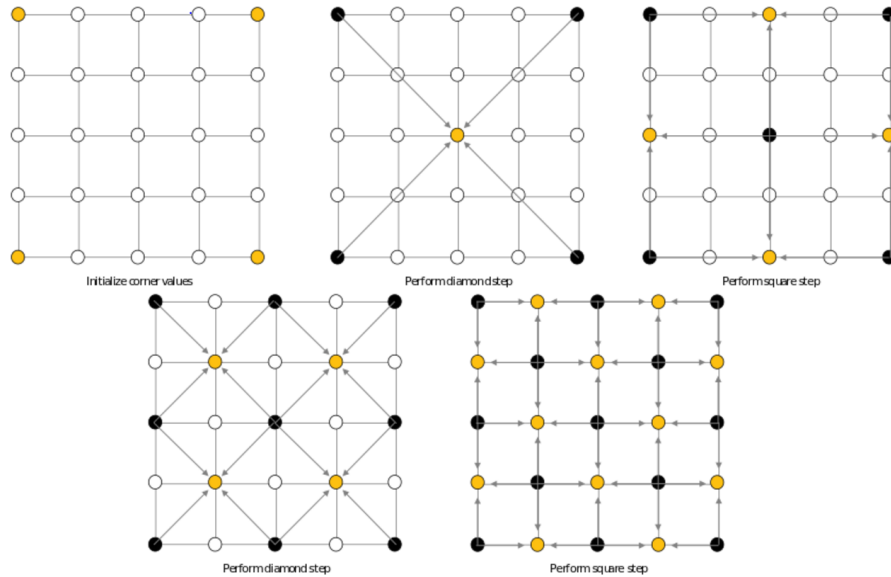


Figure 4: Example of the diamond square algorithm (source: Wikipedia).

### 2.4.3 Perlin Noise

Perlin noise was first created by Ken Perlin in 1983. He later published his article "An Image Synthesizer" at the SIGGRAPH 1985 conference [17], in which he describes his algorithm. Perlin noise is a form of gradient noise. Gradient noise is comprised of the gradients created by linearly interpolating between points in an arrangement of random or pseudo-random values. The figure 5 is an example of Perlin noise mapped as a gradient onto a 2d surface.

### 2.4.4 Fractional Brownian Motion

Fractional Brownian Motion, often called Brown noise, is created when combining several layers of noise – commonly referred to as *octaves*. Each octave will have higher frequency and lower amplitude than the preceding octave [18]. The result of combining several octaves of Perlin noise is shown in figure 6.

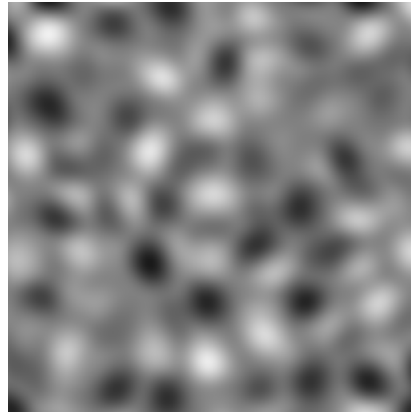
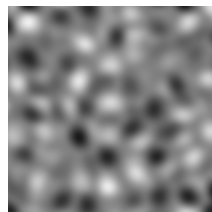
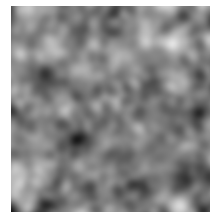


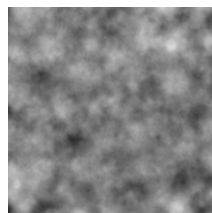
Figure 5: Example of Perlin noise. [17]



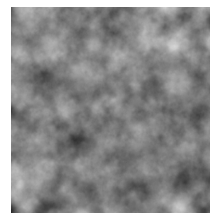
(a) One octave of Perlin noise



(b) Two octaves of Perlin noise



(c) Four octaves of Perlin noise



(d) Eight octaves of Perlin noise

Figure 6: Fractional Brownian Motion based heightmap using Perlin noise

## 2.5 Interaction Design

In human-computer interaction, software development and design, interaction design (**IxD**) is about shaping digital things for people's use and focuses on creating engaging interfaces with well thought out behaviors [19]. Like other sub-genres of design IxD focuses on form, but mostly on behavior. What clearly marks interaction design as a design field opposed to a science is the synthesis and imagination of how things are, instead of focusing on how things are. [20]

### 2.5.1 The Five Dimensions of Interaction Design

The dimensions of interaction design were first introduced by Gillian Crampton Smith in the book *Designing Interactions*. Gillian Crampton Smith stated that there were four dimensions of interaction design. Kevin Silver added an additional fifth dimension. [21]

**1D Words** The dimension that defines interactions, because words are what the user use to interact.

**2D Visual representations** The things the user interacts with on the interface. May include typography, diagrams, icons, and other graphics but is not limited to these forms.

**3D Physical objects or space** Physical objects or space is the third dimension that the user interacts with. It defines space or objects with which or within which the user interacts with.

**4D Time** The time with which the user interacts with the user interface. Some examples of this includes content that changes over time like sound, video, or animation.

**5D Behavior** Behavior defines the user actions reaction to the user interface and how they respond to these actions.

Figure 7 shows a visual representation of the five dimensions of interaction design.

### 2.5.2 Best Practices

Usability.gov has a list of question to consider when it comes to designing user interaction. The following list is freely rewritten from their webpage: [www.usability.gov/what-and-why/interaction-design.html](http://www.usability.gov/what-and-why/interaction-design.html)

#### Define How Users Can Interact with the Interface

What can a user do with their mouse, finger, or stylus to directly interact with the interface? This includes drag and drop, pushing buttons and changing functions of tools.



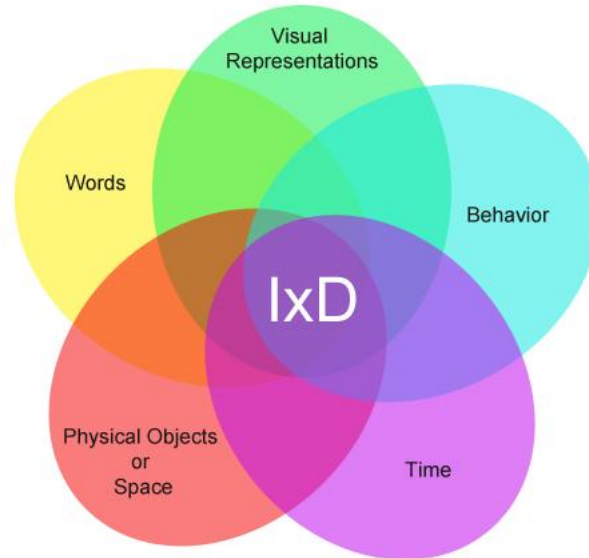


Figure 7: The five dimensions of interaction design. Copyright holder: Samir Dash.

What commands can a user give, that are not directly a part of the product, to interact with it? For example, by using Ctrl+C the user expects to be able to copy the selected item.

**Give Users Clues about Behavior before Actions are Taken**

What about the appearance (color, shape, size, etc) gives the user a clue about how it may function? Help the user by providing additional info about the objects familiar to what is already known to the user.

What information do you provide to let a user know what will happen before they perform an action? This could be giving warnings or notifications, a label on a button or instructions before submitting.

**Anticipate and Mitigate Errors**

Are there constraints put in place to help prevent errors? The Poka-Yoke Principle says that placing these constraints forces the user to adjust behavior in order to move forward with their intended action. This could be rendering an object immovable, skyboxes to prevent the user from "falling out of the map" etc.

Do error messages provide a way for the user to correct the problem or explain why the error occurred? Show the user how to fix the problem if a problem occurs.

**Consider System Feedback and Response Time**

What feedback does a user get once an action is performed? When the user performs an action, for example saving a file, the system should re-

spond to the action performed by the user.

How long between an action and a product's response time? Responsiveness (latency) can be characterized at four levels: immediate (less than 0.1 second), stammer (0.1-1 second), interruption (1-10 seconds), and disruption (more than 10 seconds).

### **Strategically Think about Each Elements**

Are the interface elements a reasonable size to interact with? Buttons need to be big enough for the user to click it.

Are edges and corners strategically being used to locate interactive elements like menus? Fitts' Law also states that since the edge provides a boundary that the mouse or finger cannot go beyond, it tends to be a good location for menus and buttons.

Are you following standards? Users have an understanding of how interface elements are supposed to function. You should only depart from the standards if a new way improves upon the old.

### **Simplify for Learnability**

Is information chunked into seven (plus or minus two) items at a time? George Miller found in his studies that users are only able to hold five to nine items in their short term memory at once before forgetting or making errors.

Is the user's end simplified as much as possible? Tesler's Law of Conservation says that as a developer one should try to remove complexity from the users side and rather have the system handle it. With that said, he also notes to keep in mind that things can only be simplified to a certain point before they no longer function.

Are familiar formats used? Hick's Law states that decision time is affected by how familiar a format is for a user to follow, how familiar they are with the choices, and the number of choice they need to decide between.

## **2.6 Interaction Technologies**

### **Oculus Rift**

Oculus Rift is a virtual reality headset developed and manufactured by Oculus VR. It was released for the public in late March 2016, becoming the first to kick-start consumer-targeted virtual reality headsets. The headset has two screens, one for each eye, running at 1080x1200 @ 90 Hz. Before the consumer edition was released, Oculus VR released two development versions of the headset, DK1 and DK2.

The positional tracking system, called "Constellation", a USB stationary IR LED sensor, which normally is positioned at the user's desk or screen. This sensor pinpoints the entire room creating a 3D space, allowing the user to use it while sitting, standing or walking around in the room [22].

### **Oculus Touch**

Oculus Touch is a set of two controllers developed by Oculus VR. The controllers are mirrors of each other and features a joystick and two buttons. Tracking of the controllers is done by the same tracking system as the Rift, the Constellation. Oculus Touch is not released with the Oculus Rift and is not yet for sale because they are not authorized as required by the rules of Federal Communications Commission **oculustouch**

### **HTC Vive**

HTC Vive is a virtual reality headset developed by HTC and Valve, released early in April 2016. It is optimized to utilize "room scale" technologies to turn whole rooms into a 3D space using sensors, allowing the user to move within the 3D space and manipulating objects using precise controllers. The HTC Vive has the same resolution as the Oculus Rift, having a resolution of 1080x1200 per eye @ 90 Hz.

The device utilizes more than 70 sensors including an accelerometer, laser position sensors and a MEMS gyroscope. The sensors can create a 3D space of 4.6 by 4.6 meters, tracking the user with sub-millimeter precision [23]. HTC Vive also has a front-facing camera that allows the software to identify objects, static or moving, as part of the "Chaperone" safety system. This will display a feed from the camera to the user if the software detects obstacles in front of the user.

### **Microsoft Kinect**

Kinect is a line of motion sensors developed and manufactured by Microsoft. The Kinect was originally meant as a controller for the Xbox, but Microsoft has opened up the technology for other uses. The device is a horizontal bar placed in front of the user, using a RGB camera, a depth sensor and a multi-array microphone to detect placement of sound.

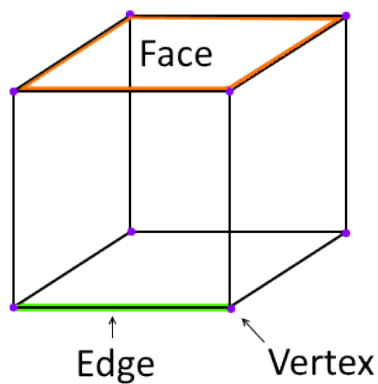
### **Leap Motion**

Leap Motion is a sensor device created by Leap Motion Inc. that supports hand and finger inputs, much like a mouse, without hand contact or touch. In February 2016 the software was optimized for virtual reality and adding a VR sensor to mount on the VR headset.

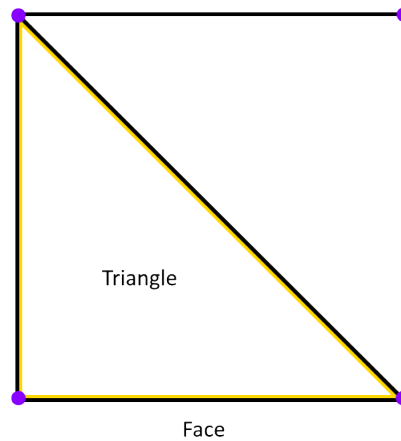
## 2.7 Computer Graphics

### 3D Models

3D models are representations of real world object in the computer world. They consist of data that the computer requires to be able to render the object. The more detailed the object is, the bigger the file size is and the more demanding it is to render. The polygon count is a normal measurement of the detail in the model. Typically 3D models are made with modeling tools like Blender by a 3D artist. 3D models is an essential component in modern computer graphics. 3D models can contain many elements, armatures, animations, textures and meshes. Normally you will at least find a texture and a mesh. Meshes consists of vertices, edges and faces. The mesh for a cube is 8 vertices, 12 edges and 6 faces.



(a) Vertices, Edges and Faces.



(b) A face consists of two triangles

### Sprites

Sprites are two-dimensional pictures. They can still be used in a 3D environments and has been for a long time. Rendering sprites is much less demanding than rendering 3D models. For modern graphics they are typically used when the user can't tell the difference and for the performance gain.

## 2.8 Swarm Behavior

Complexity science has shown that groups of animals (swarms) generate a common behavior based on local interaction between the members of the group. All these systems can be characterized by a set of familiar features.

1. Lack of a central leader overseeing the dynamics of the group, instead the latter emerges through self-organization
2. Local perception of the environment leading to a certain level of knowledge in the group because of effective information sharing
3. High degree of adaptation to changing circumstances. This includes problem solving capabilities known as swarm intelligence, making the group able to collectively perform tasks that exceeds the individuals ability

The reasons why this happens are many. Foraging, predator avoidance, and nesting are some of the key reasons [24].

Emergent behaviors is an interesting feature of swarms. Emergence is the appearance of new properties in the course of development or evolution, and behaviors of people seem to change when being a part of a group – consciously or unconsciously. Peter Miller describes emergent behavior as "whenever you have a multitude of individuals interacting with one another, there often comes a moment when disorder gives way to order and something new emerges: a pattern, a decision, a structure, or a change in direction" [25]. Miller also says that finding patterns in complex systems can be difficult, but this does not mean that the actions that are performed are done at random. Complexity is not the same as chaos. This can be seen in Figure 9 where a swarm of birds travel around, with no particular leader, but still moving as a unit.

Social subjectivity is a term used to describe an emergent behavior produced within a social system. The subjects that are produced can be of different scales. It is opinions, norms, taboos, and ideas that rise through social medias, groups of friends, culture, traditions, religion, laws and regulations, politics etc., and can be seen as an echo. This echo is not a repetition of the same, but a repetition where something new has been added along the way, *creating something more* as a result of emergent behavior.

## 2.9 Laws and Regulations

### 2.9.1 Personal Data Act (*Personopplysningsloven*)

Personal Data Act § 1 - "The Purpose of the Act" [26] states that the purpose of the Act is to protect the individual privacy, and prevent violating the individual through processing of personal data. The Personal Data Act should ensure that



Figure 9: A swarm of birds (source: Wikipedia).

personal data is processed according to basic privacy policies, for example the need for personal integrity, privacy and adequate quality of the data.

According to § 2 – “Definitions” [26], personal data is any facts or considerations that could be linked to an individual. This could be name, address, phone number, email, license plate on the car, pictures, fingerprints etc. Data about a persons behavior, such as where the person does their shopping, where they move and what they browse on the Internet is also considered personal data.

§ 13 – “Information security” [26] states that the data controller and the data processor shall through planned and systematic measures ensure that satisfactory data security with regard to confidentiality, integrity and availability when personal data is processed. To achieve this the controller and the data processor should document information and security measures. The documentation shall be made available for the employees of the data controller and the data processor, and for The Norwegian Data Protection Authority (*Datatilsynet*) and *Personvernemnda*.

This implies that the data processor has the responsibility to ensure that this is followed. There is no specifications on what is considered satisfactory security, and it is up to the controller and processor to make sure their systems are up to date.

## 3 Method

### 3.1 Project Organization

**Assigners** The assigners of the projects are local artists Harald Dyrkorn and Terese Longva.

**Scrum Team** The Scrum Team consists of three computer engineering students at NTNU Ålesund. Student numbers: 110440, 120874 and 120848.

**Supervisors** The supervisors for the project are Anniken Karlsen, Robin T. Bye and Arne Styve.

**Environment** The system is developed to be a stand-alone art installation that Terese Longva can take with her while visiting schools. If the installation has access to the Internet it has everything that it needs.

### 3.2 Development

The following section describes the methodology chosen for the project, the tools used to document and share information, tools used in the development process and resources used (both hardware and software).

#### 3.2.1 Methodology

Many of the aspects used in the projects are based on Scrum. Because of the dynamic process and exploration of ideas with the artist this was a natural choice. However, following all the aspects of Scrum slavishly would be unnatural. With such a small development team would force several people to hold many roles within the Scrum framework. Therefore only some aspects of the Scrum framework is followed.

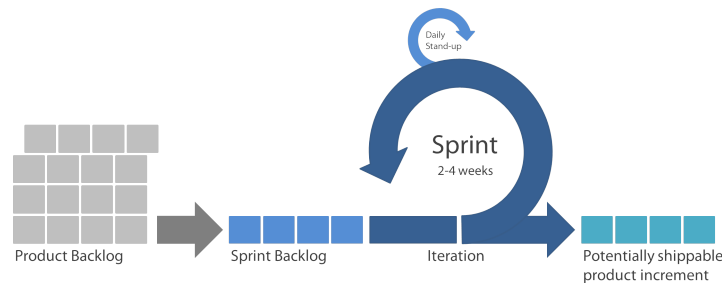


Figure 10: The Scrum Process (figure from zenexmachina.com).

We have decided set the length sprints to 2 weeks. The main reason for doing so is to make sure the dialogue with the artists are continuous in order to get important feedback. With such an open project, redefining what the goal is must be done while exploring the ideas that we get underway. Also, filling up a sprint planning two weeks ahead will not be easy, but we will start each sprint by filling in all tasks that we have agreed with the assigners and fill in tasks as we go along.

### 3.2.2 Tools

#### Unity



Unity is a powerful development platform and game engine for 2D and 3D environments developed by Unity Technologies. The platform enables developers to create games and interactive experiences. When it was released in 2005, the applications developed in Unity was only for Apples OSX . Now, the applications can be ported to multiple platforms [27]. Figure 11 shows the platforms that are supported by Unity.

iOS					
			WebGL	PS3	PS4
PSVITA	XBOX ONE	XBOX 360	Wii U		
	Gear VR				

Figure 11: Platforms supported by Unity.

According to the Unity website [28], Unity holds approximately 45% of the market within full feature game engines. With 47% of game developers relying on Unity as their main development tool, 4.5 million registered developers and over 600 million players, it is safe to say that Unity is one of the leaders within the industry.



## Facebook Graph API



Facebook Graph API is the primary way to get data in and out from Facebook’s platform [29]. The API itself is a low-level HTTP API, meaning that it bases on regular web requests to the Facebook servers, and the results are returned over HTTP. The API can be used for requesting data, posting new posts, uploading photos, managing ads et cetera.

The Graph API got its name from the idea of a ‘social graph’, a representation of the information available on Facebook. The social graph consists of:

1. **Nodes:** Elements within the Facebook universe. A post, a user, a photo, a comment.
2. **Edges:** Connections between nodes, for example a photo’s comments.
3. **Fields:** Information about nodes, similar to attributes to a entity within a database, for example the name of a page.

Facebook has created a tool for developers to play around with the API called Facebook Graph Explorer. Within the tool, developers can learn how to build up their queries to the Facebook platform or see what information that is available in the database. This could trigger ideas with creative developers. The figure below shows an example of a query in the Graph Explorer. This query could be ported to the format of the desired SDK: Android SDK, iOS SDK, JavaScript SDK, PHP SDK or cURL.



Figure 12: Example of a query to Facebook Graph Explorer.

To access the information Facebook Graph needs an access token to verify that the user has agreed that the application can retrieve the information and what information can be used by the application. When a user logs into an application and the app requests for permission, an access token is generated in the form of a long string. This string is usually added to every query. The string contains information about what information the application can access, the token also has a time-stamp and a expiration date. The access-token system is set up in order to protect the users information, giving the user authority over his/her information. To sum it up, no Facebook developer can see or download user

data without having an access token. Access tokens are only valid for a certain period of time and has to be renewed every now and then. Access tokens can be requested by apps but must be approved by the user.

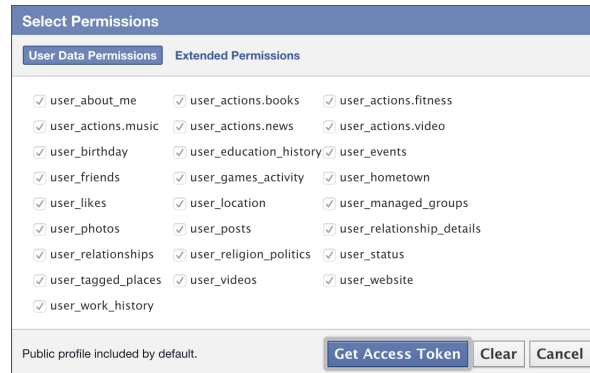


Figure 13: Example of the retrieval of an access token within Facebook Graph.

## JIRA



JIRA is an issue- and project tracking system developed by Atlassian. The software combines many tools into one, providing a versatile way to tie together an agile development team. JIRA tracks and documents all tasks done in the project and provides easily generated reports containing for example burn down charts etc. Other known concepts from Scrum are also found in JIRA, including the creation of stories, issues, sprints and product backlog.

JIRA is available for purchase in different versions, the most likely for bachelor projects being the "Small team" package. This cost a one-time fee of \$10 if you host it on your own server. NTNU Ålesund has purchased a license available for students, in this project this license is used.

## Confluence



Confluence is a collaboration tool developed by Atlassian. The software serves as a storage for all useful information in the project. PDF-files, sketches, meeting logs etc. can easily be uploaded into the software. Users can tag other team members when posting something, giving them a notification that they are mentioned in a post. Confluence has a powerful built-in online editor for writing posts. A lot of other features are also available, for example best practice templates, discussion forums, page and file versioning and of course integration to JIRA.

Like JIRA, Confluence costs \$10 if you host it on your own server. NTNU Ålesund also has a license for this software available for students.

## GitHub



GitHub is a web-based Git repository service. It offers distributed revision control and source code management. The service allows users to create own repositories for their software, adding their team members and allowing them to upload code into the project repository. Many repositories are publicly available, allowing total strangers to read their source code and create their own versions of the software. For this bachelor project a private repository is used, meaning that only invitees can see the code and contribute. There are other alternatives providing the same type of service as GitHub, but because the project group were familiar with GitHub and had acquired licenses to create private repositories, Github was chosen.

## ShareL<sup>A</sup>T<sub>E</sub>X



ShareL<sup>A</sup>T<sub>E</sub>X is an online collaborative L<sup>A</sup>T<sub>E</sub>X editor. L<sup>A</sup>T<sub>E</sub>X is a de facto standard for the communication and publication of scientific documents. The main difference between L<sup>A</sup>T<sub>E</sub>X and other Word processors like Microsoft Word is that it uses plain text, and relies on its markup language to fix the formatting. ShareL<sup>A</sup>T<sub>E</sub>X is very easy to use and requires no installation. One of the main reasons this was chosen was the possibility for collaboration and document history. The tool also provides the possibility to synchronize the code through Dropbox or Git. The software is used by approximately 500,000 students and academics worldwide [30].

## 3.3 Resources

### 3.3.1 Software

#### PyCharm



PyCharm is a Python IDE developed by JetBrains. JetBrains strive to make programming as easy as possible by creating effective developer tools with automated routine checks and corrections to speed up production. IDEs developed by JetBrains also allows for many customizations to create your own environment for generating software. PyCharm has good integration to GitHub for version control.

## WebStorm



WebStorm is a lightweight JavaScript IDE developed by JetBrains. It applies the same automated routine checks and corrections as PyCharm, and follows the JetBrains look and feel. This makes it easy for developers that are used to the interface to get to know WebStorm. In addition to JavaScript WebStorm allows developers to work with HTML and CSS. The IDE also comes with a built-in HTTP-server to host the files, so you can check your work in your browser as you go.

## Microsoft Visual Studio



Microsoft Visual Studio is an IDE to develop software for Windows as well as web-services and applications. Visual Studio supports different programming languages, including C, C++, C#, VB.NET et cetera. If you install additional language packs the IDE can also support Python, M and Ruby. The application has since 2015 been free to use in a community edition, not only for students via the DreamSpark licensing program. Visual Studio is integrated with Unity 5.

### 3.3.2 Libraries

#### Website template

To ease some of the work with the login to the application, a simple template from quackit.com was used to get the website up and running. The templates found on their website are free to use and modify. The selected template is named Business and can be found under their Business Website Templates, see the link below.

[www.quackit.com/html/templates/](http://www.quackit.com/html/templates/)

#### Bootstrap

Bootstrap is a popular framework for HTML, CSS and JavaScript. It has a collection of components (dropdown menus, buttons, alerts) to design websites in an uniform way.

## Facebook-SDK for Python

Facebook-SDK for Python is used to retrieve the users Facebook posts and other data using the Graph API. The SDK contains premade functions to gather posts, pictures, likes, location information et cetera in an easy way. The SDK is a community project, not an official SDK made by Facebook.

## Py2exe

Py2exe is an extension that converts Python scripts to executable files. This makes it possible to run the scripts without a Python installation. To create the executable file you import Py2exe and reference your original Python script. See an example below, converting the GetPosts.py-script to GetPosts.exe.

```

1 # import Py2exe
2 from distutils.core import setup
3 import py2exe
4
5 # run setup and reference the script
6 setup(console=['GetPosts.py'])
    
```

## Requests for Python

Requests is a library to send HTTP requests for Python. Requests is used by facebook-SDK to connect and send requests to the Facebook servers. Below is an example of code that connects to the Facebook API using the SDK and gathers posts with the wanted fields.

```

1 # import the SDKs used
2 import facebook
3 import requests
4
5 ...
6 # connect to the Facebook API with an access token
7 graph = facebook.GraphAPI(access_token)
8 # get the ID of the user logging in
9 profile = graph.get_object(user)
10 # set up what fields you want from the posts
11 wantedfields = "full_picture,comments,message,likes"
12 # send a get-requests for the wanted posts
13 posts = graph.get_connections(profile['id'], 'posts', fields="
    wantedfields")
    
```

### No Touch GUI for VR

No Touch GUI for VR is an asset available on the Unity Asset Store. It includes a set of menus, doors, sprites et cetera to give users a touch-free virtual reality experience. It can be used to create menus and keyboards, setup door controllers, interaction dialogues, buttons, switches, movement amongst other things. An example can be seen in Figure 14 where a menu is shown and the user can toggle the settings using the VR headset and looking at the options. A progress bar is then shown and when the progress reaches 100% the option is toggled.



Figure 14: An example menu in the No Touch GUI asset pack.

The asset pack is not free, and we would like to thank Sean P. Smith from ProDigital VR for giving us a free voucher code for the asset pack to use it in our project.

### Microsoft Kinect SDK

The Microsoft Kinect SDK is an abstraction made to simplify the process of developing for the Microsoft Kinect. The SDK contains APIs, providing support for “.NET” languages like C# and the native language C++. The “October 2014” release of the SDK included support for Unity. The Unity integration comes in the form of a Unity package and can be downloaded separately. A gesture building system also comes bundled with the SDK, this will allow you to specify certain gestures and train the system to recognise these gestures.

## Oculus Rift SDK

The Oculus Rift SDK provides native integrations for Unity engine, Cry engine and Unreal engine. It is a free and complete SDK which handles many aspects of making content for virtual reality, for example optical distortion and advanced rendering techniques. Since the package is free, developers do not need any approval or verification to develop, distribute, or sell content for it, and do not have to pay any licensing fees.

### 3.3.3 Hardware

All hardware needed in this project was provided by NTNU Ålesund. This is not the equipment that the assigners will use when the project is finished, but it will be very similar in terms of specifications. The assigners have received a component list of what is needed to set up the installation, this can be found in the appendix.

#### Computer

Computer name	Operating System	Memory	Storage
interactive_art	Windows 10	32 GB	512 GB SSD

Table 1: Computer specifications

#### Other Hardware Resources

Manufacturer	Product Name	Description
Microsoft	Kinect 2	Motion sensor
Oculus	Rift	VR-headset

Table 2: List of other hardware equipment

## 4 Results

### 4.1 System Architecture

An overview of the structure of equipment used in the solution.

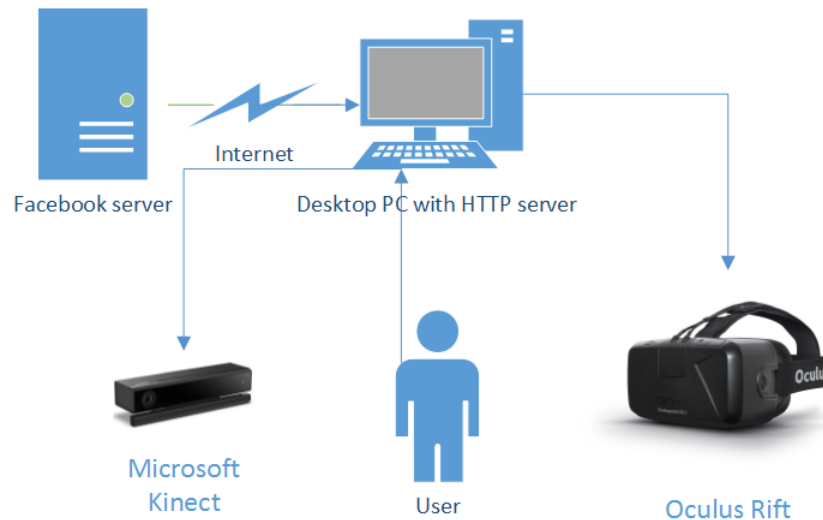


Figure 15: System Overview.

### 4.2 Solution

The solution is divided into seven key parts – web login, information retrieval, menus, terrain generation, track generation, interactivity via Kinect and stations. These seven parts combined generates a personalized virtual reality experience. The intention of the solution is to create a fun experience that makes the player (the person who is using the solution) rethink their behavior on social medias and generally in their social life. One of the main ideas behind this is swarm behaviour – what emergent behaviors are there when people are influenced by others. The solution will try, in a game-based way, make the player aware of this emergence.

#### 4.2.1 Web Login

A user ID and an access token (key) is required to set up the connection to Facebook's Graph API. This tells Facebook that the user has authorized the use



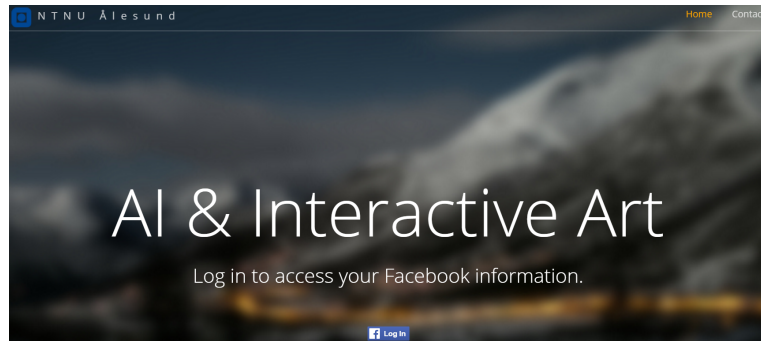


Figure 16: Screenshot from the web login page.

of Facebook data in the application. To generate these credentials a simple web page was created with the ability to log on via Facebook. There are other ways of obtaining a token such as doing it via facebook-SDK for Python, but using a web page was chosen because the users are familiar with the user interface of a web page. Also, having a nice graphical interface where the user logs in with the click of a button is easier than to make the user fill in their credentials in a Python script. When logging in to the application the user gets an alert of what information is being retrieved by the application and is asked to authorize or decline the information retrieval. If access is allowed a small CSV-file is downloaded to the computer. This file contains the ID of the user and an access token (a long string created by Facebook when logging in).

SimpleHTTPServer, a module within the http.server in Python, is used to host the web site on the local computer. The module sets up a TCP server to handle the requests sent to the web site. In the code example below you can see how the HTTP server is set up.

```
1 import SimpleHTTPServer
2 import SocketServer
3 import os
4
5 # Set the port
6 port = 80
7 # Create a request handler
8 handler = SimpleHTTPServer.SimpleHTTPRequestHandler
9 # Set up a server with the assigned port and request handler
10 httpd = SocketServer.TCPServer(("", port), handler)
11
12 # Get the current working directory
13 currentdir = os.getcwd()
14
15 # Change directory to the folder containing the website
16 os.chdir(currentdir + '/../..../Web-TokenGeneration')
17
18 print "Serving at port", port
19 # Handle requests until the application is closed
20 httpd.serve_forever()
```

### 4.2.2 Information Retrieval From Facebook

Personalizing an interactive experience can be done in many ways. Sound sensors, cameras etc. are some alternatives, using social media profiles is another possibility. We decided to go for collecting data from Facebook to get parameters to use in our application. The main reason for this is the big collection of data that Facebook has about their users and that this is accessible by an API developed by Facebook. All of Facebooks data is stored much of it can be queried. This can give a very detailed view of a person, because many people use Facebook actively and therefore Facebook has a lot of data about them. According to a Facebook post by Mark Zuckerberg, the founder of Facebook, from the 27th of April 2016 Facebook has 1.65 billion – 1 650 000 000 – monthly users.

When the application is launched a set of Python scripts are run. These scripts use the credentials downloaded from the web page to build a catalog of information. This mainly focuses on the users' own posts and the interaction on these posts. Posts on Facebook has a lot of data attached to them, including the type of post (link, picture, text etc.), pictures, geographical information, timestamps, likes and comments. Information is then organized, giving a total score of how much interaction a specific Facebook friend has had with the player's posts. The scripts also download the profile pictures of the friends that has interacted to the posts, giving the possibility to use these to create a personalized look and feel of the environment.

The parameters used from Facebook for terrain and path generation are:

Parameter name	Description
Comments	Total number of likes on the users Facebook posts
Likes	Total number of likes on the users Facebook posts
Years	The number of years since the first post on Facebook

Table 3: Parameters used from Facebook.

In addition to this there is a lot of information available that is not used. To get a full overview of available data, check Facebooks own documentation of the API: <https://developers.facebook.com/docs/graph-api>

We have set up a collection of some of the parameters we found interesting and believe can be used in further development of the application. These can be found in the appendix.

### 4.2.3 Main Menu

When the application is started a menu is shown. The main menu is optimized for virtual reality, meaning that the position of the Oculus Rift is used to maneuver the menu instead of using a mouse or a keyboard. These assets are

downloaded from the Unity Asset Store in the No Touch GUI for VR-package. A crosshair is shown in the direction that the Oculus is looking and to choose a button from the menu the crosshair must hover over the element. While the crosshair is hovering the element a progress bar will appear on the button to show the user how long he/she must hover the element for it to be chosen. This is shown in Figure 44 where the "Play"-button is chosen. After hovering the "Play"-button until the progress bar reaches 100% the application will load a new scene.



Figure 17: The main menu.

#### 4.2.4 Terrain Generation

We created components in Unity for generating terrain. The components require a set of inputs and from these values they will generate a terrain using different amounts of Perlin noise. The idea behind was to extrapolate the input numbers from Facebook data, this way, every player would see something slightly different.

##### Input Parameters

We wanted to expose parameters that could provide simple manipulation of the terrain. With these parameters one can change the overall detail of the terrain. The terrain is mostly generated using several layers of Perlin noise, this is evident in the input parameters, where one can specify how many layers/octaves of noise to use.

**Height scale** This is a multiplier that basically scales the terrain data from values between [0-1] to the desired range.

**Seed** This is how far we want to offset the Perlin noise. The offset is diagonal and negative numbers are accepted.

**Sea level** This is the level of the sea [0-1]. It defines a global water level.

**Octaves** Layers of perlin noise, each layer will add more fine detail than the preceding.

**Frequency** Controls the frequency of the noise, higher frequency gives noise with more tightly packed peaks.

**Amplification** The amplitude of the noise, the difference between the apex and bottom.

**Size** The radius of terrain generation around the player.

**Color Gradient** This is a color gradient that is used by the shader to color the terrain based on height.

### PerlinNoisePlain

A component that consists of a small piece of land, it is used as a building block in order to create a large landscape. The component consists of a script with the same name, a transform, a collider and a mesh. The script uses input parameters and the coordinates from the transform in order to modify the mesh into the correct shape. The script does this by using the transform's coordinates as a offset for the Perlin noise. Because of this, we are able to make a bigger seamless terrain using several "PerlinNoisePlain" components.

### Brownian Motion Plane

The details of the landscape is achieved using a fairly regular method of combining several layers of noise in order to create Brownian noise. We call the Perlin noise function with different frequencies, adjust the amplitude and sum them together. This way the terrain gets both macro and micro features.

```

1  public float getNoise(float x, float z)
2  {
3      float range = 1f;
4      float noise = 0f;
5      float gain = 1f;
6      for (int i = 1; i < octaves+1; i++)
7      {
8          noise += Mathf.PerlinNoise(seed * i + (x * gain / frq),
9                                     seed * i + (z * gain / frq)
10                                    ) * amp / gain;
11          range += amp / gain;
12          gain *= 2f;
13      }
14      return noise/range;
15  }
    
```

### Height Based Coloring

The plane component features a height based coloring. A simple shader applies a gradient to the terrain based on the y axis. This makes it possible to further enhance the generation by providing preset gradients or generate these as well.

### Infinite Terrain

We also implemented a script that we call Terrain Generation. This script takes the players coordinates as an input. It uses these coordinates to continually generate terrain in a set radius around the player. The terrain that goes out of bounds has its gameObject set inactive and therefore saves some memory and CPU time.

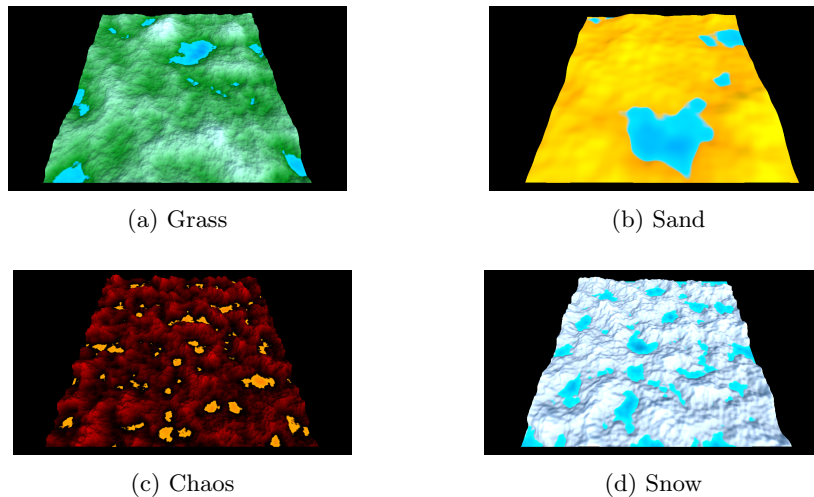


Figure 18: Different environments made by the terrain generator seen from above

#### 4.2.5 Track Generation

##### Drawing Path

We started with displaying the Catmull-Rom path using gizmos. Gizmos lets us draw lines in the editor. Being able to draw in the editor means we do not have to run the game to test. Combined with the ability to move the points we use for the path we could quickly test it and see when it performed best. A problem that occurred when we started making really long paths was that the gizmos quickly became quite demanding. In response we added a slider to

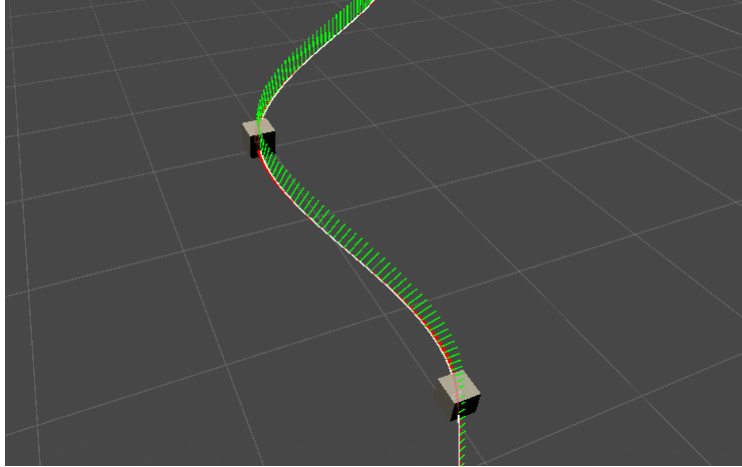


Figure 19: Gizmos.

adjust the spline step size, meaning less details/lines. We also added the ability to toggle it, so we could turn it off completely.

## Generating the Mesh

A mesh consists of several components. Vertices, triangles, normals and UV's. Vertices are points in space, triangles are connected vertices that determine how it is drawn, normals are the direction the vertices are facing and determines lighting and uv's are used for aligning the texture. Not all of them are necessary for making a mesh, and since we were not going to use textures for our track we do not use UV's.

### Positioning Vertices

To begin generating the mesh we start with positioning the vertices. As our goal is for it to be able to use any shape we give it, the vertice positions will be determined by the Catmull-Rom spline and the shape. To correctly place it we need to rotate the shape so that it points in the direction the Catmull-Rom is pointing and then take the position we get from the Catmull-Rom and add the rotated shape position.

$$VertexPos = Catmull - RomPos + Catmull - RomRotation * ShapePos \quad (5)$$

### Generating Triangles

The next step is assigning triangles to the vertices. The main thing we need to consider is the shape size (amount of points) and the amount shapes. So if we would want to make a triangle that gets made on 5 different points, we would need a minimum of 15 vertices. When making the triangles the order is important, it is therefore also important that the points inn the shape is also in the right order. The reason is that the triangle only render on one side, while the other side is see through. If you take a look at Figure 20 you can see the triangles in action.

### Splitting

Since we do not want it to be one giant mesh, we instead make smaller pieces. The process is still the same, we just need to control where they are corresponding to the Catmull-Rom. Since we are setting an amount of vertices per mesh, we can determine how long each piece will be by taking the step size and the shape size.

### Adjusting Performance

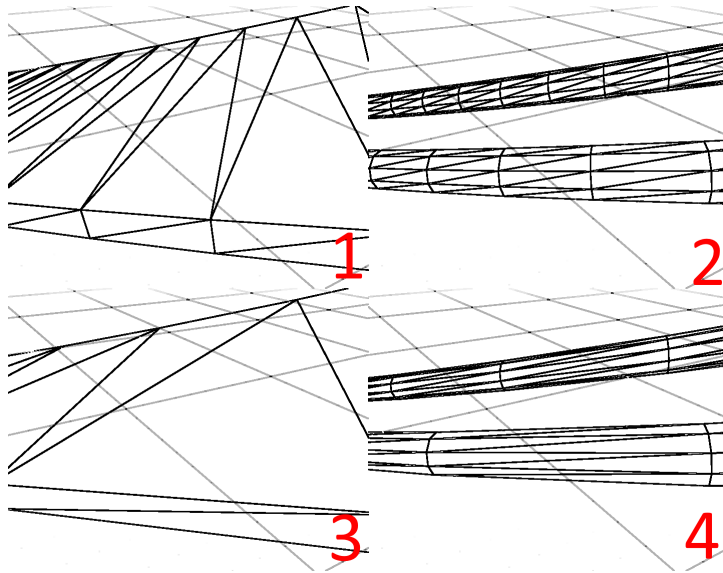


Figure 20: Triangle count.

There are two main ways of controlling the complexity of the mesh – the step size and the shape. A square for example is a much simpler shape than a

circle because the squares does not require far less points to look right. This means less triangles to render it, which in turn means that it is less demanding. By adjusting the step size we essentially determine the amount of times we place the points of the shape(smaller distance). So by lowering the step size we can also reduce the complexity. This will result in a less smooth surface, as a compromise. Figure 20 displays examples of the different parameters.

1. Simple Shape + High step count
2. Complex Shape + High step count
3. Simple Shape + Low step count
4. Complex Shape + High step count

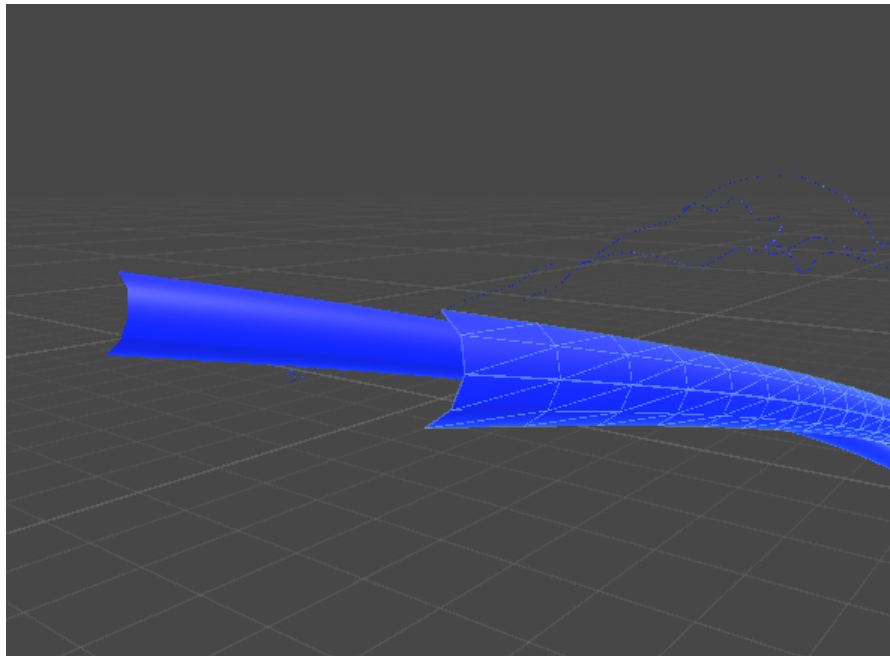


Figure 21: Mesh generated for track using a circle shape. Wireframe on the right.(Notice the triangles)



#### 4.2.6 Interactivity via Microsoft Kinect

The Kinect was used to create interactivity between the user and the system. Using motion detection gives a better feeling of being involved in the environment than using a traditional controller, for example a PlayStation-controller. By using the Kinect we can replicate the movement of the player in-game, making it more immersive. The Microsoft Kinect 2 has the ability to check where the joints of the users body is, and can also recognize hand gestures. This is used as triggers in the application. The three states that are used are "open hand", "closed hand" and "lasso" (two fingers pointing up, rest of hand closed). Below the different states of some of the prefabs we created in the project.

	State
<b>Open</b>	Claw open
<b>Closed</b>	Claw closed
<b>Lasso</b>	-

Table 4: Hand gestures for the claw.

	State
<b>Open</b>	Flashlight on
<b>Closed</b>	Flashlight off
<b>Lasso</b>	Change color of the light

Table 5: Hand gestures for the flashlight.

	State
<b>Open</b>	-
<b>Closed</b>	Throw object
<b>Lasso</b>	-
<b>Point at object</b>	Pick up object

Table 6: Hand gestures for the hand.

Figure 22 shows a screenshot from Unity showing the flashlight in action. The green line is the hand seen by the Kinect sensor. In the script there is a flashlight asset connected to right hands recognized by the Kinect, and that flashlight has a light source attached to it (as can be seen from the photo).

#### 4.2.7 Stations

Stations are where the artistic thoughts are meant arise. The purpose of this is not to make the users understand something, there is nothing specific about the experience that the application is meant to make the user understand. The purpose of a station is to make the user think – and the direction the thought takes

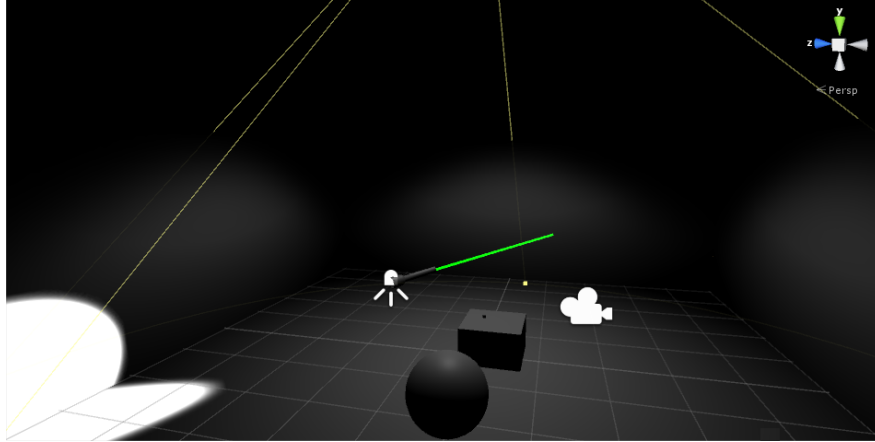


Figure 22: Screenshot from Unity showing Kinect interactivity.

is up to each user. As of now the application has two stations, but this can be expanded easily using the framework that is provided within the project.

### Station 1 - Order Friends By Activity

Something that we found interesting from our Facebook data was to see which friends a user interacted with the most. This could give an impression of who were close friends of the user and who was merely an acquaintance. A theory that the artists had was that the top friends on social media (according to likes and comments) was not necessarily the people that the user considered to be their best friends. Displaying this in the application will make people think about who they interact with on Facebook, and will subconsciously compare this to the people they consider their best friends "in real life".

To create this station we have collected all Facebook posts from the users timeline. All likes and comments are counted towards a score, giving each friend that has interacted with the user an integer to represent *how much of a friend* that person is. This integer is not shown to the user, but used in the background for calculation. A list of five Facebook friends are then presented to the user and the user is asked to list the top three friends (in order) according to how much they have interacted on Facebook. To make the experience more interesting the user has to throw trophies marked with "1", "2" and "3" onto the pictures of their friends to guess their activity level. When this is completed the roller coaster continues to the next level.

## Station 2 - Word Cloud

The language used when posting to Facebook is a very interesting topic of discussion. According to Jon Reed (a lecturer on social media) in a on OxfordWords blog (website owned by Oxford Dictionaries), social media is changing our language [31]. New words like *"unfriend"* and *"selfie"* together with new meanings for common words like *"friend"*, *"post"*, *"wall"* and *"status"* are some of the things that social media changes with our language. To cite Joe Reed: *"Because the language we use to communicate with each other tends to be more malleable than formal writing, the combination of informal, personal communication and the mass audience afforded by social media is a recipe for rapid change."*

To observe this via our application was something that we and the artists found very interesting. Therefore, one of our stations creates a world cloud of the most common words used in the users Facebook posts and comments. This could show trends in the language that the user is not aware about (for example mixing dialect language with formal language in social media according to who their audience is) or maybe even see new words. The frequency of how much a word is used is represented by the scale of the word in the word cloud. The words are appearing all around the user and the user can freely look around using the Oculus Rift.



Figure 23: Screenshot from the word cloud scene.

## 4.3 Development Process

### 4.3.1 Statistics

GitHub provides insightful graphs to give an overview of the projects progress. One of these graphs shows how many lines of code has been added to the repository per day. Since our project has had two repositories – one for the information retrieval and one for the project in Unity – it is also interesting to note the differences between the two repositories. The information retrieval repository is represented by the graph in Figure 24. Figure 25 shows the progress of the Unity repository, where the scripting for in-game behavior and assets are.



Figure 24: Git: Activity Graph from Facebook Graph API repository.

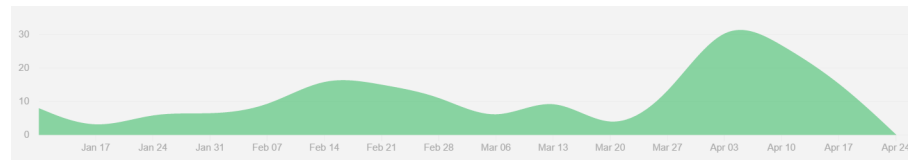


Figure 25: Git: Activity Graph from the Unity repository.

As can be see from the graphs, the activity for the Facebook Graph API-repository were mostly in the start of the project. The reason for this was mainly to get the framework of information retrieval up and running. At that point in time the general direction of the project was not decided either, so gathering information from Facebook to see if it was usable was the main activity. Logically, after having the information retrieval in place we could go on investigating how to use that information in Unity. Therefore the activity in the Unity repository was mainly after the information retrieval was done.

### 4.3.2 Sprint Reports

After each segment of the project was done, before having a meeting with the artists and the supervisors, we made a sprint report to document the work done in that specific period of time. Each sprint report contained sections explaining the planned activities, main activities (activities actually done), reasons for deviation, requests/needs for guidance and planned activities for the next sprint. Since JIRA and Confluence were down for some time during the project and

that we did not push through to follow the actual sprint lengths as set before the project, the sprint reports vary in terms of how long of a period they cover. All sprint reports can be found in the appendix.

## 4.4 Information Security

### 4.4.1 Personal Data

After logging into an application that downloads personal data it is very important to clean up the data after the user terminates the application. Pictures, Facebook posts, user ID and session tokens are removed when the user terminates the application so that the user can feel secure about logging into such an application on a public computer. It is our responsibility as the data processor to make sure that all data are handled in a secure manner according to The Personal Data Act (*Personopplysningsloven*).

### 4.4.2 Authentication

Authentication is handled safely by Facebook's own API. After someone pressing the Facebook login button, an "OnClick"-trigger goes off. This runs a function to check if the user is logged in to Facebook and/or the app. That function returns a response message from Facebook containing the status of the login. The following JavaScript function (`statusChangeCallback`) then runs to check the response and act correspondingly.

```

1 function statusChangeCallback(response) {
2     // The response object is returned with a status field that
3     // lets the
4     // app know the current login status of the person.
5
6     if (response.status === "connected") {
7         // Logged into the app and Facebook.
8         runAPI();
9
10    } else if (response.status === 'not_authorized') {
11        // The person is logged into Facebook, but not the app.
12        document.getElementById('status').innerHTML = 'Please log '
13        +
14        'into this app.';
15    } else {
16        // The person is not logged into Facebook, so we're not
17        // sure if
18        // they are logged into this app or not.
19        document.getElementById('status').innerHTML = 'Please log '
20        +
21        'into Facebook.';
22    }
23 }
```

One security feature that is implemented by Facebook is the verification of source (the web page domain/IP address or app ID) from which the user logs in to applications. The developers of an application has to declare a white list of sources that are allowed to log in to the application. This prevents people with bad intentions to abuse our code to download personal data. If the code is used at an unauthorized domain, Facebook displays an error message. This error message is shown in Figure 26.



Figure 26: Error message if the Facebook login is not as declared.

## 5 Discussion

### 5.1 Engineering Approach vs. Artistic Approach

An interesting topic of discussion has been the interactions between artists and engineers, when engaged in a cooperative venture troubles are likely to ensue. Because of radically different mindsets and personality types, the joint venture would require patience and understanding from both parts in order to succeed. Engineers are known to have a problem-oriented approach. G.F.C Rogers describes engineering as *"the practice of organising the design and construction of any artifice which transforms the physical world around us to meet some recognised need"* [32]. In practice this means that engineers look at the problem and start thinking about possible solutions to the problem. It is therefore very convenient for engineers to have a clearly defined problem, for which they can define a working solution. Artists on the other hand, they may not be as concerned with the solution right away. Artists often want to investigate the problem, philosophically. They often look at the different ways of approaching the problem, both on a philosophic level and an artistic level. This is something that we have experienced in our project, especially in the early phases of the project. In the exploration phase of the project, the project group was very keen to define the problem/task and was already thinking of ways to do it both in terms of technologies and code. The artists were not as fixated on the solution, but rather the thought behind the project.

These differences between engineers and artists did in our case lead to some strained interactions in the start-phase of the project. The project group was left with a feeling of having to push the project onto the next phase. The artists would have liked to spend more time on each step of the way. The time constraints of the project and the importance of delivering a good project and a good report has been hanging over the group all the way. This pressure may have led to the exploration phase not converging onto the best possible solution. Given more time, especially in the earlier phases of the project *could* have yielded a different and possibly more suitable solution. We used about 20% of the total time allocated to the project to explore alternatives, though a lot of this time was lost because of issues with compatibility. We felt at this point some pressure to push things forward and start working towards a solution/product.

#### 5.1.1 Exploration

The exploration phase was allotted a large part of the project's time budget. We estimated to use around 20% of the total time allocated to explore different options. This was due to the creative nature of the project and lack of technological constraints. The idea to "make an interactive experience that provokes

certain thoughts and or feelings about how people are interconnected socially” could be solved in a variety of ways, choosing a good path straight from the start was important. Early on in the exploration phase the project group would research the different computer interaction technologies available – Oculus Rift, Microsoft Kinect and the WalkMouse.

The first technology we attempted to research was the Microsoft Kinect. The Kinect we borrowed from the university was the newer 2.0 version – this version had a short list of supported USB controllers. It turned out that neither of our three laptops were supported by the device. This in turn led to us putting the Kinect exploration on ice until we had a desktop computer with a supported USB controller. We decided that we would move onto testing the Oculus Rift instead.

Researching the Oculus Rift also turned out to be a bit difficult, because of issues regarding compatibility. The project group’s laptops were not compatible with the latest Oculus SDK. This was due to Oculus dropping support for NVIDIA Optimus enabled devices, which covers most laptops. The NVIDIA Optimus is a technology that is used to conserve energy when dealing with graphics on mobile devices, sending the graphic signal through both the integrated graphics card and the external graphics card. Only one of the graphics cards is used, and if the system can conserve energy by using the integrated card it does exactly that automatically. The project group therefore had to acquire a sufficiently powerful desktop computer in order to test the Oculus Rift. We contacted our supervisors to get a desktop PC in order to continue developing for the Oculus Rift.

Because there were no available computers for us to use we had to reconsider our options about how we could construct a interaction. This led us to the idea about using past interactions to create a new interaction, using Facebook’s vast collection of data to produce some sort of individual experience. We started researching ways of easily obtaining the Facebook data. At first we had to use a manual way of retrieving the Facebook data. This meant getting the access token from the Facebook developer site and pasting it into a small Python 2.7 script. The Python script would then retrieve all the user’s posts and save it to a text file. This however was a quite cumbersome procedure that would not be much of a user-friendly experience.

In order to make the Facebook information retrieval more user friendly we registered a Facebook application. Having a Facebook application registered opened up for the possibility of retrieving the access token directly through the JavaScript in the browser. We created a small web page with a title and a Facebook login button, this enabled us to easily extract the user’s token so that it could be used by the Python script.

After about 6-7 weeks we were moved to the visualization lab and assigned a desktop computer. We quickly started testing the Oculus Rift, trying to size up its potential in the project. Having explored the Oculus Rift for about a



week we decided to give the artists a preview, in order for them to experience the possibilities of the technology. We also got the Kinect 2.0 to work with the desktop computer's front USB controller, this enabled the exploration of the Kinect. Having tested both the Kinect and the Oculus Rift, the artists were positive towards using these technologies in the project. The Oculus Rift had made an especially strong impression because of its ability to bring a person into a virtual reality.

### 5.1.2 Customer Interaction

The interactions between the project group and the artists was mostly satisfactory. Though the project group is left with a concern. Because of the limited time, more suitable solutions might have been lost due to rushing the artists' exploration phase. What the artists are left with might not be the best solution and they might not be able to return to that fresh "mindset/point of view" that they had in the exploration phase. The artists have decided to take the project with our solution to a convention for artists, there they can cooperate with other artists and try to further develop the idea behind.

The time issue is something that has been a topic of discussion in other, similar projects as well, for example in the development of Sonic Onyx in Trondheim [33]. The Sonic Onyx project was a collaboration between an artist and a group of bachelor students - much like our project. They had similar experiences when dealing with the issue of time in their bachelor's project. In at least one specific part of the project the students had to continue with the development of the project even though the artist felt like they had to wait for on-site testing of the artwork, but the students had to continue developing due to time-limitations. In the Sonic Onyx project the students developing the installation had a clear view of what was to be created before they started. They knew that it was going to be an interactive sound installation that used the latest technologies and the location where the artwork was going to be installed (outside a secondary school in Trondheim). Having a clearer view of what was going to be created and what technologies to use seems to have eased the interaction with the customer. One of the reasons why they were going to develop was because the artist had knowledge of the technologies and therefore had a better overview. This led to the development team having a consultant in the project that was familiar with the technologies and could assist them with the technical issues.

From our own experience in dealing with a very open and abstract goal, we found the customer interaction to be challenging. To explore the different technologies and choose what technology is the most appropriate to express the message of the artwork has been a time-consuming, and at times frustrating, part of the project. We have used *many* hours to explore what has been done by others, reading up on different technologies. To determine if the complexity of developing such an installation is within the scope of a bachelor's project. Finding ways of adapting the technologies to match the underlying idea of a

”collective intelligence” has been difficult. Our experience is that deciding what to create (what technologies to use and the form of the artwork) is very time-consuming and would be better if this was set in advance.

Interaction between the project group and the external assigners hit a bump from the end of March until the middle of April. One of the assigners was sick and therefore unreachable and the other one was busy with exams. Because of the time frame of the project (final delivery 27th of May) we had to make some choices on our own without much involvement from the assigners – this was mostly terrain generation and track generation. After meeting with them and showing how had chosen to solve the challenges we had the artist expressed that they were satisfied by the work we had done and they apologized that they had not been able invest much time in the project in that particular time period.

To summarize how the interaction with the external assigners went we feel that the dialogue was good between us. There was always room for making suggestions and all suggestions were discussed in an equal manner. Getting a continuous discussion is something that could have improved. Some unfortunate events with sickness and exams in addition to the artists living in Oslo led to the communication issues, and these things could not be predicted at the beginning of the project.

## 5.2 Development

Scrum was chosen as the software development framework in this project. Working hours were set to 7,5 hours per day (08.00-15.30) from Monday to Friday, with some exceptions due to other activities the group members were obligated to attend to (work, exams, lectures). The tools provided by the university to support agile software development (JIRA and Confluence) fit well to the chosen framework and the members of the group got the needed training to use the software in the lectures of *”Ingeniørfaglig systemteknikk og systemutvikling”* from Arne Styve.

Due to the fact that the accounts to log in to JIRA and Confluence were distributed quite late in the semester (late January) we started using the tools quite late. Ideally these would have in place at the start of the project early in January. Also, around Easter there was a setback because of a hard drive crash on the server running JIRA and Confluence. Some of the data was retrieved, some of it was gone. This led to a quite big gap in the use of the tools and this ruined the use of breakdown graphs etc. in this period. The project group should maybe have taken backups of the spaces and project that were saved in JIRA and Confluence, but there was no easy way to export all these files and they had to be manually downloaded.

Scrum as a framework functioned quite well with an open and experimental project like this. The ability to adjust the path along the way was something

that was much needed, and using for instance the Waterfall method in this setting could have ruined the progress of the project.

### 5.2.1 Guidance and Meetings

Since the assigners of the project are mainly based in Oslo and meetings with them required some effort to arrange, the meetings throughout the project were not planned in the start. Ideally all meetings should be set in the start of the project so that everyone would know the dates early on. Instead meetings were planned along the way, giving every participant an email with the location, time and agenda the week before so that they could prepare for the meeting and put it in their calendars.

### 5.2.2 Applications and Tools

The tools that were used throughout the project proved very helpful, some more than others. The IDEs that were used (Microsoft Visual Studio, PyCharm and WebStorm) had very intelligent tools to help with the coding process, both correcting mistakes and making sure obsolete methods not were used. Visual Studio was used on a daily basis. PyCharm and WebStorm were used mostly in the start of the project to provide the information from Facebook. All three IDEs had good integration with GitHub, to ensure that no code was lost while sharing code between the developers.

Visual Studio's IntelliSense in particular provided very clear and insightful input while coding. Below is an example of how the IntelliSense can come in handy, stating that the method used is obsolete and giving a tip of what should be used instead.

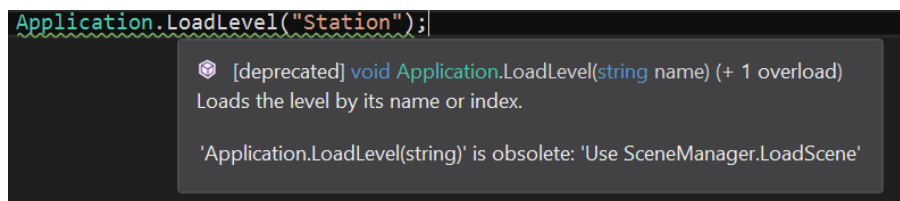


Figure 27: Example of the IntelliSense in Visual Studio.

ShareLaTeX and Unity was also used a lot, respectively for documentation and for the development of the game environment. Both were easy to use, had good and informational guides and overall were a good experience to use.

The use of Confluence and JIRA became a topic of discussion within the group. We had used Confluence as a Wiki for the project, adding reports, brainstorming and documentation and used JIRA as a support tool for the Scrum framework.

After Easter the server hosting the services had a hard drive crash, resulting in the service becoming unavailable for some time and some other virtual servers hosted on the same hardware were corrupted. The backup was not configured properly, so some servers became permanently unavailable. One of the reasons of using these services hosted by the university was the safety of a centralized backup solution, securing our data from events like hard drive crashes. We dwelled for some weeks before starting to use the solutions again because of this, resulting in holes in our formal sprints, stories and tasks in JIRA. After considering our possibilities we decided to clean up the project in JIRA and finish strong, using both Confluence and JIRA. When the services were up and running we found that they were very handy to formalize the work flow and help structure our work.

### 5.3 Platform

The process of choosing a platform was experimental. Since the project owners did not set any restrictions when it came to the platform, it was all up to the team to decide based on earlier experiences and what seemed to be the best platform to present the message in the artwork. To do this the project group used a lot of time to study earlier bachelor's projects, other pieces of interactive art and technologies to see what was available on the market.

#### 5.3.1 Game Engine

Unity has been used as a game engine for many bachelor's project at NTNU Ålesund in earlier years and there are resourceful people at the university with knowledge of Unity. NTNU Ålesund just acquired a Unity Pro license for the new visualization lab. This includes standards assets, like for example a camera rig to render Unity projects at the 180 degree canvas found in the visualization lab. The Unity community also has a lot of forums and knowledge databases with information available aswell. Therefore Unity became a natural choice as a game engine.

#### 5.3.2 Peripherals

Using Oculus Rift and Microsoft Kinect also became a natural choice since the university already had the equipment available and the project group could borrow the equipment free of charge. Having the equipment on campus also meant that there were people to ask if there were specific challenges related to the equipment. Microsoft Kinect has been used in similar interactive art projects, and also has a lot of knowledge within the community of Internet forums.

In addition to the equipment mentioned above we were interested in trying to use the WalkMouse to move around in the virtual environment. The WalkMouse is a circular omnidirectional treadmill used to translate the users movement into input in the program. Using an omnidirectional treadmill while using a VR-headset like Oculus Rift could be dangerous without the proper safety equipment. With the headset on, the user would not know his or hers position on the treadmill, and could therefore fall off. To make that safe one would have to have a railing or straps to ensure that the user is in the "right" position. The wire from the headset to the computer could also get stuck in the rotating spirals on the treadmill, destroying the equipment. In addition to the safety challenges of using the WalkMouse, the WalkMouse itself was not quite as robust as it should have been if it was to be used in such a project. The students who are responsible for testing and maintaining informed us of this at an early phase, and said that the WalkMouse was a prototype not ready for these kinds of projects. If the WalkMouse had been fully functional it could be used with the 180 degree projector view in the visualization lab instead of the Oculus Rift. This would remove the safety issues that occurs when the user does not know where he/she is positioned on the WalkMouse. See Figure 28 to see the WalkMouse in use.



Figure 28: WalkMouse in use, image from WalkMouse.com.

## 5.4 Terrain Generation

The idea was to generate a landscape based on someones Facebook data. Our approach to this was making a terrain generation algorithm capable of generating different landscapes using simple inputs. Further work would revolve around post processing Facebook data into a viable input for the terrain generation algorithm. The end goal would be a terrain that was generated from Facebook data.

### 5.4.1 Plane

We used the standard plane prefab from Unity as the smallest building block for the terrain. The standard Unity plane prefab has a transform, mesh and a collider. A collider is a piece of code that lets the game engine know when something is colliding with something else. What we had to do in order to make this into a more interesting terrain was to modify the mesh, more specifically, modify the vertices' y-coordinates (height). To make it more realistic we opted to use some noise algorithms in order to create more detailed meshes, that more closely resembled natural terrain.

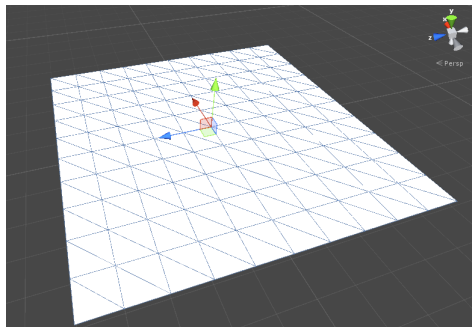


Figure 29: Unity's built in plane prefab.

### 5.4.2 Noise Functions

Noise functions are very widely used within procedural generation of terrain, textures and other forms of visuals. Noise functions enables us to create levels of detail without expending a lot of time or resources.

We explored several different noise functions like the diamond square algorithm, Value noise and Perlin noise. The diamond square algorithm is very simple and usually produces a very detailed and interesting terrain. The downside is that it makes it difficult to implement infinite terrain. This is due to the algorithm working its way from the outside of the map and inwards. In the

end we decided to go with Unity Mathf's implementation of Perlin Noise. This made implementing an infinite terrain much easier.

### 5.4.3 Brownian Plane

In order to make the terrain more interesting one can use the noise function several times at different frequencies and sum it together. The result will be a terrain that contains both macro features and small micro details. This way of combining several layers of noise results in something often referred to as fractal noise or "Fractional Brownian Motion" [18].

```

1  public float getNoise(float x, float z)
2  {
3      float range = 1f;
4      float noise = 0f;
5      float gain = 1f;
6      for (int i = 1; i < octaves+1; i++)
7      {
8          noise += Mathf.PerlinNoise(seed * i + (x * gain / frq),
9                                     seed * i + (z * gain / frq)
10                                ) * amp / gain;
11
12         range += amp / gain;
13         gain *= 2f;
14     }
15     return noise/range;
    
```

This code snippet describes the function where we sum up several layers of noise and normalize it to a [0.0,1.0] range. This example uses `Mathf.PerlinNoise()` which is a part of Unity's standard Math library. For simplicity we chose to use `Mathf.PerlinNoise()` instead of implementing our own noise function.

### 5.4.4 Optimization

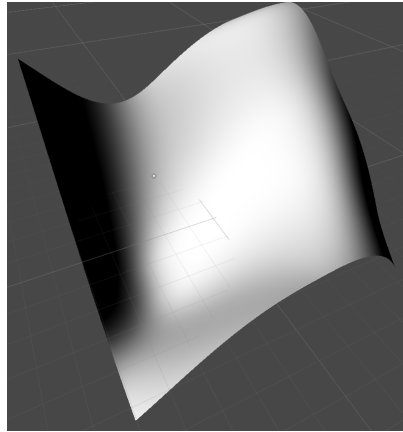
In some of our earlier iterations we had some performance issues regarding the terrain generation. We quickly came to the conclusion that it had to be related to the mesh generation. After some debugging, trial and error we discovered that Unity's function for retrieving the array of vertices from a mesh was slowing things down.

Calling "mesh.vertices" did not return a reference to the array but rather a copy of it. This turned out to be slowing down the program because every time "mesh.vertices" was modified a new copy of the array would have to be made.

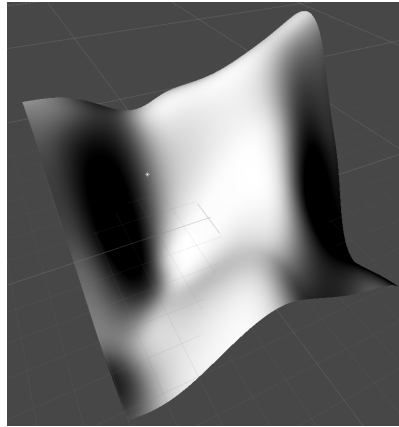
```

1  Mesh mesh = GetComponent<MeshFilter>().mesh;
2  Vector3[] verts = mesh.vertices;
    
```

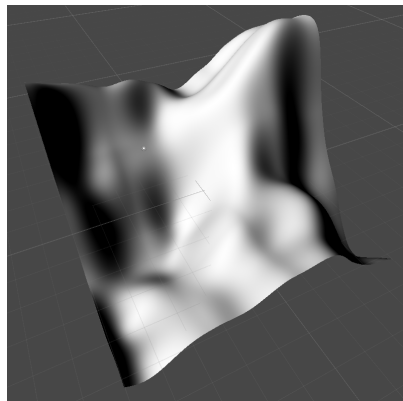
Our solution was to create our own array of vertices. Make all the necessary modifications to it and then assign it to the mesh.



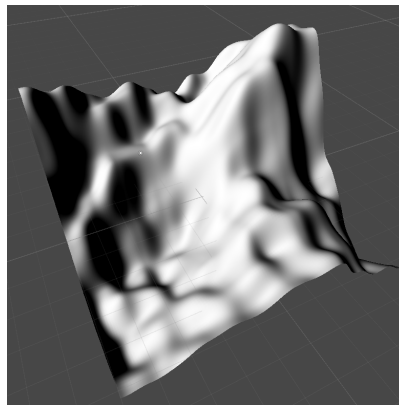
(a) One octave of Perlin noise



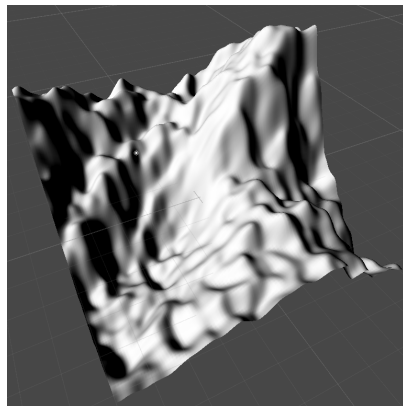
(b) Two octaves of Perlin noise



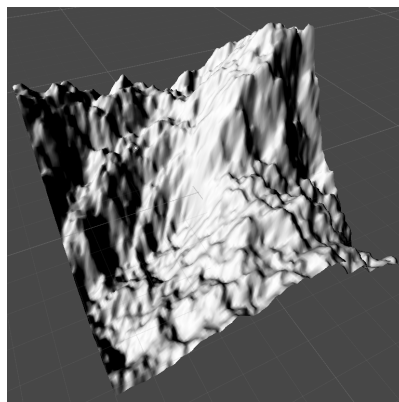
(c) Three octave of Perlin noise



(d) Four octaves of Perlin noise



(e) Five octave of Perlin noise



(f) Eight octaves of Perlin noise

Figure 30: Fractional Brownian Motion based terrain generation



```
1 Vector3[] verts = new Vector3[121];
```

In our final code we used a double for loop to traverse the vertices, assigning height values to them based on the summed Perlin noise. The function `getNoise(x,z)` will always return a value between `[0.0,1.0]`. The `getHeight()` returns a value that is the sample value multiplied by a `heightScale`. This translates the `[0.0,1.0]` range into something that is more similar to a terrain. The `getHeight()` function also checks if the point is above or below the water level. The "coloring.Evaluate()" function takes a number between `[0.0,1.0]` and uses this number to select a color on a gradient.

```
1 Mesh mesh = GetComponent<MeshFilter>().mesh;
2 Vector3[] verts = new Vector3[121];
3
4 float posX = transform.position.x;
5 float posZ = transform.position.z;
6 // transform.position represents the position of the
7 // terrain plane itself.
8
9 for (int i = 0, z = 0; z <= 100; z+=10)
10 {
11     for (int x = 0; x <= 100; x+=10, i++)
12     {
13         float xx = (posX + x);
14         float zz = (posZ + z);
15
16         float sample = getNoise(xx, zz);
17         verts[i] = new Vector3(
18             x,
19             getHeight(xx, zz, sample),
20             z
21         );
22         colors[i] = coloring.Evaluate(sample);
23     }
24 }
25 mesh.vertices = verts;
26 mesh.colors = colors;
```

#### 5.4.5 Infinite Terrain

We decided to go with the naive approach and generate terrain around the player's position with a predefined radius. When the player would move the length of a terrain plate in either direction we would update the terrain. To achieve this we made a new component called `TerrainGenerator`, this component takes the player position and a radius as inputs. It always ensures that the player has terrain generated in all directions.

In order to optimize the generation process we used a `HashMap` to store the generated "TerrainPlates", for the key we used a string consisting of the `x` and `z` coordinate of the plate. This made the process of checking if a certain `TerrainPlate` was generated, much faster and trivial. The "TerrainGenerator"



Figure 31: Diagram showing the generation procedure

component first retrieves the player position. It then starts iterating through the coordinate space around the player within a box determined by the radius. In each iteration it looks up the coordinate in a hashmap to see if the "tile" exists.

```

1 HashMap tiles = new HashMap();
2
3 for (int i = -radius; i < radius; i++){
4     for (int j = -radius; j < radius; j++){
5         string name = "x=" + (i * planeSize + player.pos.x) +
6             " z=" + (j * planeSize + player.pos.z);
7         if (tiles.ContainsKey(name)){
8             // Omitted
9         }
10        // Omitted
  
```

The "tile" construct is a inner class in "TerrainGenerator". It is used to store a reference to a TerrainPlate and a time stamp. The time stamp is the time that the tile was last modified. We use this time stamp in order to deactivate plates that are beyond the radius of the player. This was done as a optimization in order to minimize resource usage.

```

1 public class Tile
2 {
3     public float timeStamp;
4     public GameObject plate;
5     public Tile(float timeStamp,GameObject plate)
6     {
7         this.timeStamp = ts;
8         this.plate = p;
9     }
10 }
  
```

The deactivated plates are only removed from the rendering queue. They still have their mesh and colliders. Enabling a deactivated tile is therefore much less taxing on the system compared to generating a new one. This is mostly due to not having to rerun the Perlin noise procedure to generate the mesh.

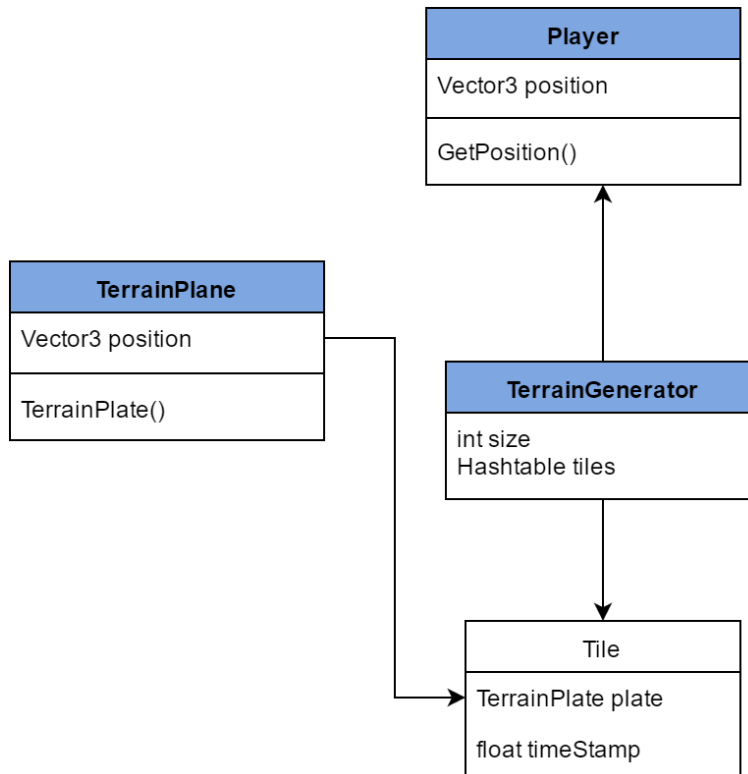


Figure 32: UML diagram showing components and relations to TerrainGenerator.

### 5.4.6 Coloring

To color the terrain we used Unity’s gradient implementation. We added colors to the gradient and used the gradient’s ”Evaluate()” function to color the terrain based on height. The Evaluate function takes a float input with range [0.0,1.0] and translates this into the corresponding color on the gradient. It was instantly compatible since our noise function returns a value in the same range.

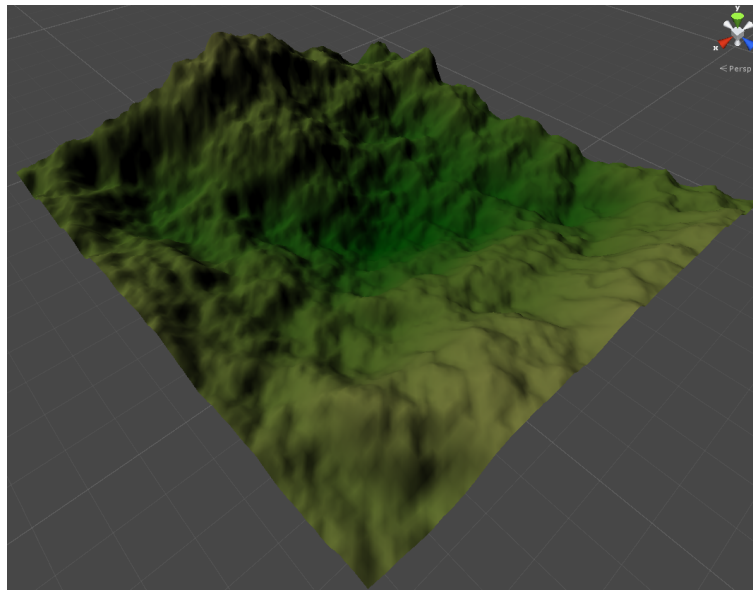


Figure 33: Mesh ”y-coordinate” based coloring.

We made gradients both by hand and by using an algorithm that looked at the colors of pictures.



Figure 34: Gradient for terrain coloring

### 5.4.7 Processing Facebook data

Most of the input data can be generated using numbers and their median values for scaling. An example would be a user’s amount of friends. The scaling would take into account the median amount of friends on Facebook accounts. The median is used in order to still have representative numbers after scaling. The input is clamped as to not allow unsupported values. Following this approach, one can use most Facebook data as input as long as non local medians are

available. The coloring of the terrain uses the accounts profile picture. A gradient is generated based on the average colors of different sections of the picture.

## 5.5 Track Generation

Since the user's hands are busy controlling the Kinect the user needs to be transported through the terrain. To solve this we have generated tracks that the user follows through the environment. The generation of the tracks are separated into two parts. The mathematical smoothing between points and the mesh generation. For the mathematical smoothing Catmull-Rom was used, it lets us create splines that smooth the ride and gives it some curves. We chose Catmull-Rom instead of Bézier because cubic Bézier would make the track a set of four-point tracks as sections. These sections would then have to be put in order. We decided to use Catmull-Rom because adding control points to the curve and customizing it was easier with Catmull-Rom than Bézier.

The mesh generator needs points and rotations calculated by the Catmull-Rom function. It can then extrude the shape (circle, square..) given to it along the splines. This means that it creates vertices based on the shape, positions and rotations. It then makes triangles between the vertices which are used for the rendering. For optimization purposes it makes smaller pieces of mesh rather than one big one. Since we can give it any shape, it is highly customizable though it is restricted to smooth edges since its main purpose is rounded shapes such as circles.

## 5.6 Software Related Challenges

### 5.6.1 Programming Languages

Scripts in Unity and the Kinect SDK required some knowledge of C#. No one in the group had worked with C# before, but due to the similarities with Java and the big knowledge databases available on the Internet this was not seen as a big problem, but required some reading. Python was not as unknown as C# since all members of the project group had used Python to learn key concepts within the Information Security- and Operating Systems course at NTNU Ålesund. One interesting thing to mention is the distribution of programming languages within the Git repositories used in the project. GitHub provides a feature where you can see a distribution of all the different languages used in the repository, and in this particular project C# and JavaScript were the most popular languages. A full distribution can be seen below in Figure 35.

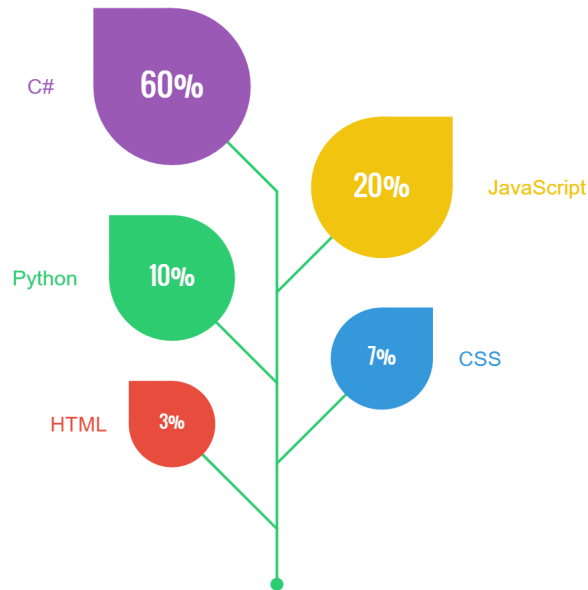


Figure 35: Distribution of programming languages within the Git repository.

### 5.6.2 Choosing Type of Splines

When making the roller-coaster we faced the challenge of choosing which type of spline to use. We had to consider if we should use Bézier or Catmull-Rom. Both had advantages and disadvantages. Bézier was much more flexible and could create more interesting roads. Drawback being that two of the four points was used only for manipulating the spline. This meant it was difficult to create a smooth path. Catmull-Rom however passes through every point (except the endpoints when not looping), this meant we had more control over how it would move. Of course there are workarounds with Bézier but then its advantages would not be utilized. The main problem was that we were going to be using points generated by a script, rather than placing them manually. So we had to either choose between the simplicity of Catmull-Rom or the flexibility of Bézier.

### 5.6.3 3D Models vs. 2D Sprites

Game development is a very deep and complex field, and there are many tricks to reduce the system load of an application. One trick is to render 3D models of static objects as 2D sprites at certain distances. Creating a 3D model is much more complex than creating a 2D drawing, and therefore is more expensive. However, if the model is going to be animated 2D is much more complex than

3D. We used this knowledge when creating models of a tree.

The initial idea was to fill up the terrain with trees to improve the feeling of velocity when taking the roller-coaster. We found that having static objects in the scene gave the player a more pronounced sensation of speed. First we created a 3D model of a tree and populated the terrain with the model. This required a big part of the system’s resources. Most of the load was caused by the system having to render all the 3D models, no matter how far they were from the player’s camera. After reading on the Unity forums we decided to try to minimize the load by using Unity’s built-in level of detail (LOD) groups. This enables developers to render one model if the player is within a certain range, and another less advanced model when the player is further away. It is possible to implement several LOD-entries, but we decided to go for two – one for close distances and one for longer distances. See Figure 36 to see a snapshot from Unity’s LOD groups.

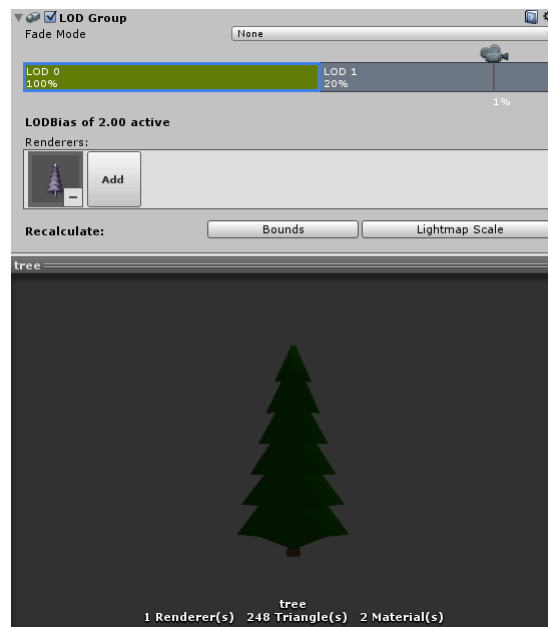


Figure 36: 3D model of a tree and LOD-groups in Unity.

## 5.7 Further Work

### 5.7.1 Find new uses of the Facebook data

There are lots of data available in the Facebook Graph API, the biggest limitation is knowing how to use it. There are plenty of other ways to use the data

than what is done in this thesis, and the implementation should not be a problem. By doing this the user would get an even more personalized experience and this could make the application more interesting. Attached to this thesis there is an appendix showing some of the data that potentially could be extracted from Facebook.

### 5.7.2 Generate More Stations

The application now contains two stations. Stations are where the user will get to experience their Facebook data – this could for example be in the form of puzzles/mini games or a word cloud like the ones already implemented. It is via these experiences that the user will truly understand how much data Facebook has about them and that maybe their behavior on social media is not as they expected themselves. Expanding the number of stations can be one of the ways of expanding the application. As mentioned in the previous paragraph, check the appendix to see examples of what data can be found on Facebook.

One important thing to note is that when expanding the application by adding stations, the time it takes to run through the application expands. Since the user is wearing Oculus Rift and experiencing motion in the form of a roller-coaster this could become tiresome and/or nauseating if it takes too long to complete or is not done in a proper way. While testing this in the lab we found that around 10 minutes is enough when wearing VR gear.

### 5.7.3 Interaction Technology

When the project description was formalized in early January not all the interaction technologies mentioned in the theory-section had been released. The consumer edition of the Oculus Rift had not been released and neither had the HTC Vive. This meant that if we were going to use virtual reality in our project we would have to use the DK2 edition of the Oculus Rift as our virtual reality headset, and we would not get to use the precision controllers included in the HTC Vive-kit. Fortunately, Oculus Rift was available at the university.

Both Leap Motion and Oculus Touch had not been released either. This led to having to look for alternatives, and Microsoft Kinect became an obvious alternative. The device was available at the university and the sensor's API gave us many of the possibilities we looked for in a controller.

#### Microsoft Kinect

Our implementation of the Kinect technology was based around a seated position with focus on recording the players arms and hands. Therefore we had to



adapt the Kinect through software, in order to get a reasonable result. We disabled the full body tracking and selected the necessary limbs to track. The limbs we selected to track were (Arm, elbow to wrist), (Head, neck to head).

During our testing we found the Kinect skeleton data to be a bit irregular, especially when the arms were oriented towards the Kinect. The tracking of the (Wrist ,wrist to fingertip) limb was particularly affected, that is why we decided to only use the (Arm, elbow to wrist) limb.

The Kinect's tracking of the arm limb was more robust but suffered from a slight jitter. This jitter turned out to be the Kinect struggling with estimating the exact position of the elbow joint. We solved the jitter issue using linear interpolation to smooth the problem data.

While the interpolation solved our initial problem it also created a slight delay between input and action on screen. After testing different values of interpolation we landed upon 100ms for the elbow and 30ms for the wrist. Having a higher interpolation on the elbow did not negatively affect the experience, this could be due to the joint's more stationary nature "whilst in a seated position".

### Other Technologies

With the application running as it is per today, the interaction via the Microsoft Kinect is not optimal. This because the sensor works best watching larger joints on your body, and fine-tuning with your hands could at times prove to be annoying because of the inaccuracy in the sensor. One of the features that put the Microsoft Kinect in front of other interaction technologies was its ability to read if a mouth was open, eyes closed etc. Making a system that can read the mood of the user could be very interesting, but when using the Oculus Rift this proves quite difficult. We would recommend that if interaction should be the main focus of the further development of the application that HTC Vive will be used. The Vive has a controller device for each hand, letting the application know the exact positions in space where the hands are. This would be a much less frustrating experience if interaction should be the main focus of the development. Below, in Figure 37, you can see the HTC Vive in action, image courtesy of PCWorld.

Interacting with the cart could be one an interesting way to further develop the interaction part of the application. Since the player is sitting in a fixed position within the cart one could add controls to the cart via buttons on a dashboard. This could for example give the user ability to accelerate/decelerate the cart. For the implementation of this interaction we would recommend using position-tracking hand controllers (HTC Vive, Oculus Touch), this due to the higher precision and robustness .The Microsoft Kinect out of the box is not fine-tuned for small hand gestures, but rather full body tracking. This makes it less viable for hand tracking in VR.



Figure 37: The HTC Vive in action, image courtesy of PCWorld.

#### 5.7.4 User Testing

In the course of the project the main testers of the application were the developers. When a new function became available, the developer that created the function used the other two as test personnel. This testing removes most of the minor and major faults in the code, letting someone unbiased evaluate the work that has been done. To ensure that the application is interesting enough to its intended audience and as bug-free as an application can be, further testing is required by people that can evaluate the application with new and honest eyes.

#### 5.7.5 Optimize Code For Better Graphics

Experiencing lags, spikes or glitching in the graphics can break a user's immersion into the virtual reality. The general experience with the application is good, but there is definitely room for improvement, some of our implementations are naive and unoptimized. Our project group had no prior knowledge or experience with unity or game development.

#### 5.7.6 Ambient Sounds

Sounds are a big part of any virtual experience. Ambient sounds can create immersion, giving the user a feeling of being in the environment that has been generated. The sounds that has been used in this project are royalty-free sounds found in online databases and are not custom made for the application. Creating better sounds that are more fitting to the environment could improve the experience. The framework for playing sounds on triggers has been created, sounds have not been prioritized outside of that. Development was focused more on the visuals rather than sound, due to using HMD, this would have the most immersive effect.

### 5.7.7 Port For Use In The Visualization Lab

NTNU Ålesund will in the second quarter of 2016 open the new visualization lab. The visualization lab is a new lab for students to visualize their projects on a 180 degree canvas. No equipment is permanently put in the lab as it is meant to be a multi-purpose lab. The image is made from 12 projectors and warped to compensate for the angle between the projector and the canvas. The lab also contains a speaker setup with 10 speakers and 2 sub woofers to create a full stereo experience. This would open up some doors when it comes to perceiving the application as a group, and one could develop special stations or terrains based on a group of people rather than an individual. See Figure 38 below from the visualization lab showing the projectors and a part of the canvas.



Figure 38: The visualization lab at NTNU Ålesund.

During the project NTNU Ålesund acquired a Unity Pro-license to use in the visualization lab. This includes assets such as a camera rig that can be fitted to use the projectors at the visualization lab instead of using the Oculus Rift. If the further development of the application is done by students at NTNU Ålesund porting the application to use in the visualization lab would be a natural next step.

### 5.7.8 Input From Microphone

Using sound as an input is something that we did not prioritize during the project, this was due to time limitations. The Kinect has four microphones.

They are configured as an array. The microphones are strategically placed, this enables the Kinect to estimate the distance and direction of recorded sounds. This enables the use of sounds and directions as interactive elements in an application. This feature could be an interesting addition if the application is ported for use in the visualization lab with larger groups of people.

### **5.7.9 Visual Facebook Timeline**

Facebook operates with user profiles being represented as a timeline. Posts are arranged by the time they were created and a user can scroll through their posts. The posts can be arranged in the same way in a virtual reality environment, showing pictures and text posts as the user "travels through time". To make this even more interesting their own posts could be mixed with things that were news at that certain date. This could for example be the launch of the first iPhone, the election of a new president or the start of a war. It could be interesting to see if there is a correlation between the content that is posted by the user and the news.

### **5.7.10 Appropriate Objects Populating Terrain**

As of now the application only has one type of object to populate the terrain – pine trees. This is good for forest-like terrains, but not a particularly appropriate object to put in a desert or other type of terrain. For each terrain there should be appropriate objects to populate the terrains. An example of this could be deserts, where tumbleweeds or cacti could be a fitting prop in the terrain.

## 6 Conclusion

The projects intention was to explore the possibilities of using interactive technologies to create an artwork that focused on making the user aware of emergent behaviors and their general behavior on social media. In addition to this our goal was to create the first edition of such an installation. We feel that we have managed to create an application that meets the requirements that were set for the project and that the application is interesting enough to make young people (our target audience) want to spend time using it. As far as we have seen when searching the Internet for similar installations, no one has made an interactive version of a roller coaster. This will hopefully mean that users will find it interesting since they have not seen anything like it before.

Being part of a project where the goal is not clearly defined in the start but rather found along the way has been a challenge and an interesting experience. It is been a challenge to acquire enough knowledge about the available technologies to make a decision whether this would be a useful supplement to the project or if it should be put to side. It is very probable that this will resemble projects in jobs in the future – because no matter how detailed the requirements will be, trying out different technologies to satisfy the demands will be a key part of the projects.

Many aspects of the project have been unfamiliar to us and we have had to learn the fundamentals within many disciplines. Game development (including both behavior of objects and generating visual content), virtual reality and interactive technologies (Microsoft Kinect) are some examples of disciplines where we had to start by learning the basics. This has been an interesting challenge and developing an application that takes advantage of these technologies has been very motivating since 2016 will most probably be the birth year of consumer virtual reality experiences with the release of Oculus Rift and HTC Vive. Studying the information stored on Facebook has also been an interesting experience and given us an insight in just how much Facebook knows about their users and how this can be used to create revenue.

Our assigners have expressed that they are satisfied with the delivery that the project has provided and they look to develop it further with other artists, possibly also other students. We are very satisfied with the result and look forward to seeing what the application will be in the future.

## References

- [1] D. Cyranoski, *World's oldest art found in indonesian cave*, [www.nature.com/news/world-s-oldest-art-found-in-indonesian-cave-1.16100](http://www.nature.com/news/world-s-oldest-art-found-in-indonesian-cave-1.16100), [Online; accessed 10.02.2016], 2014.
- [2] J. Braman, *Handbook of Research on Computational Arts and Creative Informatics*. IGI Global, 2009.
- [3] D. K. Skolesekken, *About "den kulturelle skolesekken"*, <http://www.denkulturelleskolesekken.no/om-skolesekken/>, [Online; accessed 04.05.2016], 2016.
- [4] O. Dictionaries, *Art - definition*, [www.oxforddictionaries.com/definition/english/art](http://www.oxforddictionaries.com/definition/english/art), [Online; accessed 28.03.2016], 2016.
- [5] N. Spivey, *Greek Sculpture*. Cambridge University Press, 2013, pp. 114–115.
- [6] W. user: Marsyas, *Acma moschophoros*, [https://commons.wikimedia.org/wiki/File:ACMA\\_Moschophoros.jpg](https://commons.wikimedia.org/wiki/File:ACMA_Moschophoros.jpg), [Online; accessed 28.03.2016], 2016.
- [7] Saylor.org, *Module 1: introduction and definitions*, <http://www.saylor.org/site/wp-content/uploads/2011/12/Module-1.pdf>, [Online; accessed 04.05.2016], 2016.
- [8] G. D. Taylor, *The soulless usurper: Reception and criticism of early computer art*. University of California Press, 2012.
- [9] D. Kuspits, *The matrix of sensations*, [www.artnet.com/magazineus/features/kuspit/kuspit8-5-05.asp](http://www.artnet.com/magazineus/features/kuspit/kuspit8-5-05.asp), [Online; accessed 28.03.2016], 2016.
- [10] I. Sommerville, *Software Engineering, 9th Edition*. Pearson Education Inc, 2010, p. 6.
- [11] S. B. Michael Bloch and J. Laartz, *Delivering large-scale it projects on time, on budget, and on value*, [www.mckinsey.com/business-functions/business-technology/our-insights/delivering-large-scale-it-projects-on-time-on-budget-and-on-value](http://www.mckinsey.com/business-functions/business-technology/our-insights/delivering-large-scale-it-projects-on-time-on-budget-and-on-value), [Online; accessed 17.02.2016], 2012.
- [12] K. Schwaber and J. Sutherland, *The scrum guide*, [www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-US.pdf](http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-US.pdf), [Online; accessed 17.02.2016], 2013.
- [13] J. H. Gerald E. Farin and M.-S. Kim, *Handbook of Computer Aided Geometric Design*. Elsevier, 2002, pp. 5–6.
- [14] E. Catmull and R. Rom, "Computer aided geometric design", *RE Barn*, 1974.
- [15] W. user: Hadunsford, *Catmull-rom spline*, [https://commons.wikimedia.org/wiki/File:Catmull-Rom\\_Spline.png](https://commons.wikimedia.org/wiki/File:Catmull-Rom_Spline.png), [Online; accessed 30.03.2016], 2016.

- [16] A. Fournier, D. Fussell, and L. Carpenter, “Computer rendering of stochastic models”, *Communications of the ACM*, vol. 25, no. 6, pp. 371–384, 1982.
- [17] K. Perlin, “An image synthesizer”, *SIGGRAPH*, 1985.
- [18] B. B. Mandelbrot and J. W. V. Ness, “Fractional brownian motions, fractional noises and applications”, *SIAM: Society for Industrial and Applied Mathematics*, 1968.
- [19] Usability.gov, *Interaction design basics*, [www.usability.gov/what-and-why/interaction-design.html](http://www.usability.gov/what-and-why/interaction-design.html), [Online; accessed 30.03.2016], 2016.
- [20] A. Cooper, R. Reimann, and D. Cronin, “About face 3: The essentials of interaction design. 3rd”, *Indianapolis, Indiana, published simultaneously in Canada: Wiley publishing, inc*, 2007.
- [21] S. Tayal, “Engineering design process”, *International Journal of Computer Science and Communication Engineering*, pp. 1–5, 2013.
- [22] O. VR, *The oculus rift, oculus touch, and vr games at e3*, <https://www.oculus.com/en-us/blog/the-oculus-rift-oculus-touch-and-vr-games-at-e3/>, [Online; accessed 26.05.2016], 2016.
- [23] L. Kelion, *Htc reveals virtual reality headset with valve at mwc*, <http://www.bbc.com/news/technology-31664948>, [Online; accessed 26.05.2016], 2016.
- [24] R. Bouffanais, *Design and Control of Swarm Dynamics*. Springer, 2016, p. 1.
- [25] P. Miller, *Smart swarm*. Collins London, UK, 2010.
- [26] Lovdata, *Personal data act (personopplysningsloven)*, <https://lovdata.no/dokument/NL/lov/2000-04-14-31/>, [Online; accessed 29.04.2016], 2016.
- [27] Unity, *About unity*, <https://unity3d.com/unity>, [Online; accessed 08.02.2016], 2016.
- [28] —, *Fast facts*, <https://unity3d.com/public-relations>, [Online; accessed 09.02.2016], 2016.
- [29] F. Inc, *About graph api*, <https://developers.facebook.com/docs/graph-api/overview/>, [Online; accessed 08.02.2016], 2016.
- [30] ShareL<sup>A</sup>T<sub>E</sub>X, *About*, <http://www.sharelatex.com>, [Online; accessed 10.02.2016], 2016.
- [31] J. Reed, *How social media is changing language*, <http://blog.oxforddictionaries.com/2014/06/social-media-changing-language/>, [Online; accessed 26.04.2016], 2016.
- [32] G. F. C. Rogers, *The nature of engineering: A philosophy of technology*. Macmillan Press, 1983.
- [33] S. M. Salah Uddin Ahmed Letizia Jaccheri, “Sonic onyx: Case study of an interactive artwork”, *NTNU*, 2009.

# Appendices

**A - Hardware Requirements**

**B - Available Facebook data**

**C - System Documentation**

**D - User Manual**

**E - Installation Guide**

**F - Sketches**

**G - Preliminary Report**

**H - Meeting Reports**

**I - Sprint Reports**

**J - Project Poster**



# A - Hardware Requirements

Below the hardware requirements for the different components are listed. Note that there is one table showing the requirements for the consumer edition of Oculus Rift and one for the older development kit (DK2). The development kit is not for sale anymore, but can be bought used for example from Ebay. This would be a cheaper alternative, but the developer version is not as good as the consumer edition. Therefore the consumer edition is a bit more expensive. Buying the consumer edition also sets higher requirements for the specifications for the computer. Therefore the whole package will become even more expensive. We would recommend investing in the consumer edition since this is the edition released for the general public and will be supported for a longer time than the developer version. Also, having better hardware specifications will make it easier to create a better experience, getting higher frame rates and less lag. Including more features in the future will also be easier with better hardware.

## Oculus Rift - Consumer Edition

Component	Requirement
Graphics card	NVIDIA GTX 970 / AMD 290 equivalent or greater, with 1 HDMI-port.
Processor	Intel i5-4590 equivalent or greater.
RAM	8GB+ RAM.
Connectors	2 USB 3.0 ports.

Table 7: Requirements for Oculus Rift - Consumer Edition.

## Oculus Rift - DK2

Component	Requirement
Graphics card	Nvidia GTX 600 series or AMD Radeon HD 7000 series (or better) with DVI-D or HDMI graphics output.
RAM	8GB+ RAM.
Connectors	2 USB 2.0 ports (at least one powered).

Table 8: Requirements for Oculus Rift - DK2.

## Microsoft Kinect 2

Component	Requirement
Graphics card	DX11 capable graphics card.
Processor	I7 3.1 GHz (or better).
RAM	4GB+ RAM.
Connectors	Built-in USB 3.0 host controller (Intel or Renesas chipset).

Table 9: Requirements for Microsoft Kinect 2.

## Example PC Configuration From Komplet.no

We have set up an example of a PC that fulfills all the requirements for the newest Oculus Rift and Microsoft Kinect 2. This is just an example to show the price range of the components needed to run applications with the recommended hardware. We chose to get all prices and components from a single supplier – Komplet.no – since they are the most used web shop for computer parts in Norway.

Component	Product	Price	Article no.
PSU	CM G750M 750W	949	795139
SSD	Kingston 120GB 2.5"	479	773515
RAM	HyperX DDR3 8GB	499	812852
CPU	Intel Core i7-4790K	3490	817261
Graphics Card	EVGA GTX970 4GB	3690	835338
Motherboard	MSI H97 GAMING 3	1299	815065
Cabinet	CM MasterCase 5	1195	851058

Table 10: PC from Komplet.no

Total price for all parts: 11 601 kr.

Note that this requires the buyer to build the computer. If this is not an option one could contact Komplet.no customer support to get a price for the built computer. This would probably be an extra fee of around 500-1000 kr.

## B - Available Facebook data

This is information that is attached to each Facebook post and can be used for creating personalized experiences.

Parameter name	Description
id	The post ID.
application	Information about the app this post was published by.
caption	The caption of a link in the post (appears beneath the name).
created_time	The time the post was initially published. For a life event, this will be the date and time of the life event.
description	A description of a link in the post (appears beneath the caption).
from	Information about the profile that posted the message.
full_picture	Link to the picture in the best resolution available.
link	The link attached to this post.
message	The status message in the post.
place	Any location information attached to the post.
privacy	The privacy settings of the post.
shares	The shares count of this post.
status_type	Description of the type of a status update.
story	Text not intentionally generated, such as when two people become friends.
to	Profiles mentioned or targeted in this post.
type	A string indicating the object type of this post.
with_tags	Profiles tagged as being 'with' the publisher of the post.

Table 11: Collection of parameters available from Facebook.

# C - System Documentation

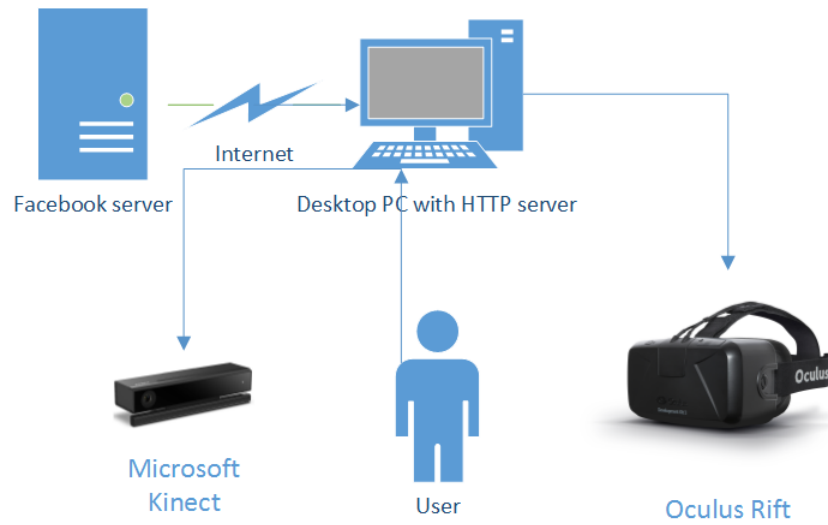


Figure 39: System Overview

System Architecture

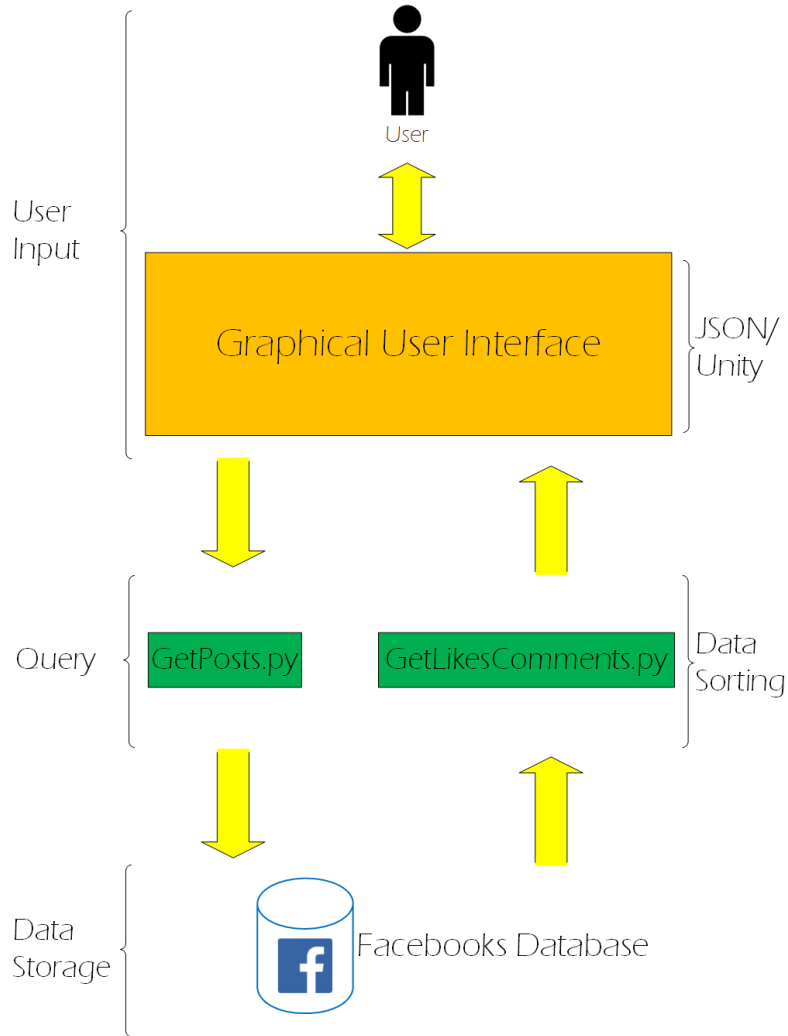


Figure 40: System Architecture

# D - User Manual

## Downloading Facebook credentials

To download your Facebook data, you have to verify that the application is authorized to contact Facebook servers and get data. To do this, open your Internet browser (*Google Chrome, Mozilla Firefox, Internet Explorer, Safari etc.*) and go to `http://art.app:8080`. This will lead you to the login page.

Press the blue Facebook login button as to log in via Facebook. This opens up the popup as seen in Figure 41. Ignore the error message in red - this is because the application is not available to the public at the time of the screen capture.

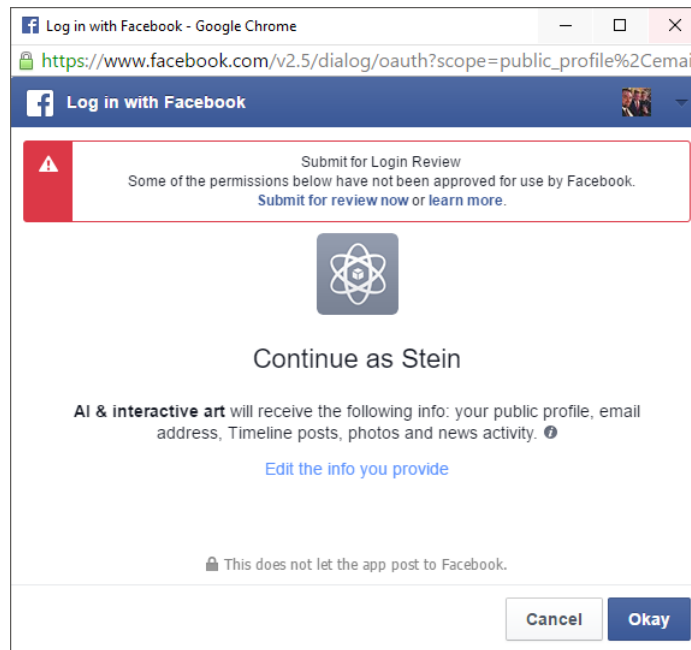


Figure 41: Popup showing what the app wants access to.

After pressing "Okay" you will be transferred to the web page as seen in Figure 42, and a download of a CSV-file with your credentials is also started.

If there are multiple files called credentials in your download-folder the application will automatically select the newest edition of the file. This is done because the access token has an expiration time and has to be renewed every now and then.

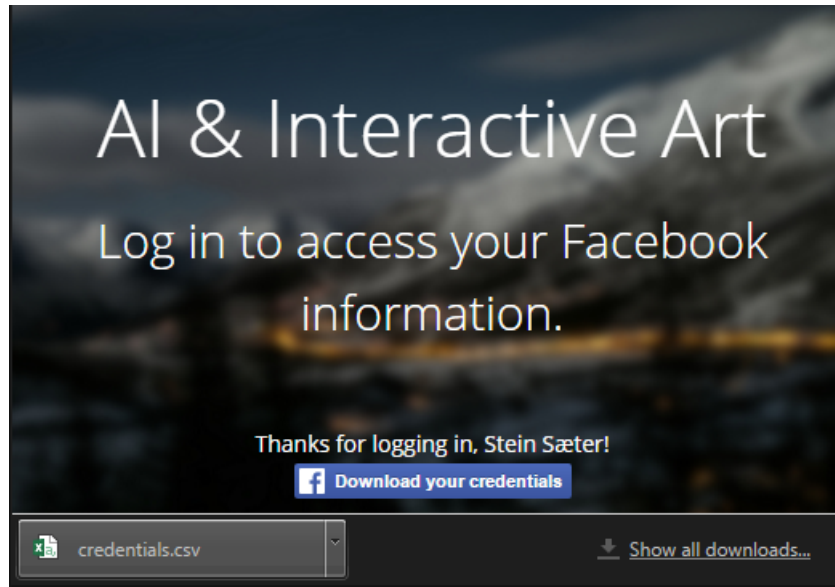


Figure 42: Success - Credentials downloaded.

## Launch Application

To launch the application go to the installation folder and open ArtApp.exe. The icon of the application will look like Figure 43 with the Unity symbol as an icon.

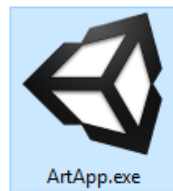


Figure 43: The ArtApp icon.

## Main Menu

When the application is started a menu is shown. The main menu is optimized for virtual reality, meaning that the position of the Oculus Rift is used to maneuver the menu instead of using a mouse or a keyboard. A crosshair is shown in the direction that the Oculus is looking. To select a button on the menu simply let the crosshair hover over it. While the crosshair is hovering the element a

progress bar will appear on the button to show the user how long he/she must hover the element for it to be chosen. This is shown in Figure 44 where the "Play"-button is chosen.



Figure 44: The main menu.



## Hand Gestures

Hand gestures are used to give input to the application via Microsoft Kinect. The Kinect is a motion sensor that detects bodies, finding hinges, hand gestures, face gestures etc. The left hand is set to control a claw with a gripping function and the right hand is used to control a hand with the possibility to lift and throw objects. The flashlight is not yet implemented, but the state table is presented nonetheless.

To change the state of the flashlight and the claw, three different hand gestures are used – open, closed and lasso. These are shown below in Figure 45.



Figure 45: Kinect hand gestures – Closed (left), lasso (middle) and open (right).

Note that since the Kinect only sees gestures from one angle the hand gestures must be done pointing towards the sensor. If your hands are behind your body it is impossible for the Kinect to recognize the gesture. Some times, if the hand is hard to distinguish from the clothing it could also read the wrong hand gesture even while pointing at the sensor. Move the hand out from the center of the body if this happens, this will make it easier to distinguish the gesture from the background.

### Claw

The table below shows the possible states for the claw with the corresponding hand gesture.

	State
<b>Open</b>	Claw open
<b>Closed</b>	Claw closed
<b>Lasso</b>	-

Table 12: Hand gestures for the claw.

### Flashlight

The table below shows the possible states for the flashlight with the corresponding hand gesture.

	State
<b>Open</b>	Flashlight on
<b>Closed</b>	Flashlight off
<b>Lasso</b>	Change color of the light

Table 13: Hand gestures for the flashlight.

### Hand

The table below shows the possible states for the hand with the corresponding hand gesture.

	State
<b>Open</b>	-
<b>Closed</b>	Throw object
<b>Lasso</b>	-
<b>Point at object</b>	Pick up object

Table 14: Hand gestures for the hand.

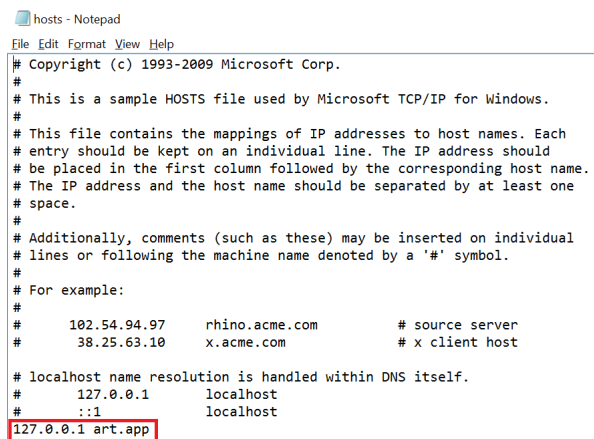
# E - Installation Guide

## SETUP FOR USE

### Adding A Hostname To The Hosts-file

Since Facebook requires the application to have listed what servers can be authorize to talk to Facebook servers on behalf of the application, we have found that the best way of going around this without purchasing a domain and putting the application on the internet is to add a line to the hosts-file in the Windows operating system. This file is checked by the computer before asking the DNS what the IP address for that domain. Here is a step-by-step guide on how to add a hostname to the hosts-file:

1. Move to the location of the hosts-file: `C:\Windows\System32\drivers\etc`
2. Copy the hosts-file to another directory, for example your desktop.
3. Open the copy with a text editor, for example Notepad.
4. At the end of the file, add a new line with "127.0.0.1 art.app" (see picture below).
5. Save and close the file.
6. Move the edited file back to `C:\Windows\System32\drivers\etc`



```

hosts - Notepad
File Edit Format View Help
# Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#       102.54.94.97       rhino.acme.com          # source server
#       38.25.63.10      x.acme.com              # x client host

# localhost name resolution is handled within DNS itself.
#       127.0.0.1        localhost
#       ::1             localhost
127.0.0.1 art.app
    
```

Figure 46: Editing the hosts-file.

Ping art.app from the command line tool to check that the configuration is working as intended. If you get replies from 127.0.0.1 everything is as it should.

If you get replies from 127.0.53.53, redo the steps above.

### Local HTTP Server

A local HTTP server is included in the source files. Run the "httpserver.exe" found in FacebookGraphAPI%Python-InformationRetrieval%web to start the server. If this computer is primarily used for the art installation, then the "httpserver.exe" file should be added to the "startup" folder. This makes it run on startup. To access the "startup" folder, first open Run by pressing the Windows-button and R at the same time (Win + R). When "Run" has appeared, enter "shell:startup" and press OK. This will open up a explorer window in the startup folder. Add a shortcut to the HTTP server to this folder. The server will now automatically run the next time the user logs in.

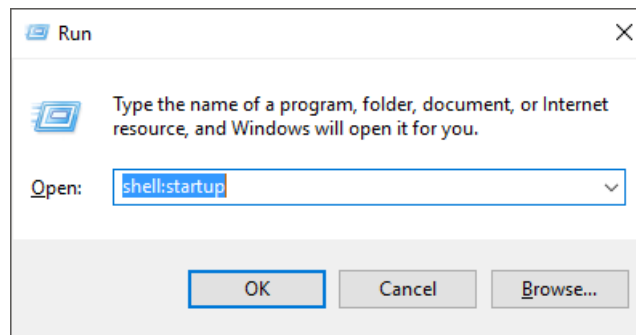


Figure 47: Finding the startup folder via the Run-function in Windows.

## SETUP FOR DEVELOPMENT

### Oculus SDK

Download the Oculus SDK to get the native support in Unity. The SDK can be found in the link below.

<https://developer.oculus.com/downloads/>

### Kinect SDK

Download the Kinect SDK to get the native support in Unity. The SDK can be found in the link below.

<https://developer.microsoft.com/en-us/windows/kinect>

### Python 2.7

Python 2.7 is used to gather the Facebook data and setup the HTTP server. Note that there are two different versions available, 2.7.x and 3.5.x. *These are two completely different versions of Python and they do not contain the same types of functions!* Download the latest version of Python 2.7 from the link below. Make sure that the option for installing Python as an environmental variable is checked (this is not checked by default). This will let you open the command line tool and type "python filename.py" to open Python files.

<https://www.python.org/downloads/>

We would strongly advise to use a Python IDE to edit the Python scripts. This will make the development much easier as the IDE understands what functions are available in Python. There are many good IDEs available and it all comes down to individual preferences. We recommend using the IDE we used, PyCharm from JetBrains. PyCharm is free and can be downloaded from the link below.

<https://www.jetbrains.com/pycharm/download/>

The Python scripts require some libraries. These can be installed directly from the command line tool in Windows by typing "python -m pip install NameOfPackage". The packages used are:

- facebook-sdk (for Facebook integration)
- requests (to send HTTP requests to Facebook)
- py2exe (to convert Python files to executables)

### **Web IDE**

An web IDE is also highly recommended to use to edit HTML/CSS/JavaScript. Like for Python, there are many good IDEs available for free. We recommend using WebStorm from JetBrains. WebStorm is free and can be downloaded from the link below.

<https://www.jetbrains.com/webstorm/download/>

### **Unity**

Unity is the development platform for tying the building blocks within the application (scenes, assets, scripts, models etc.) together. The Personal Edition of Unity is free and can be downloaded from the link below.

<https://unity3d.com/get-unity/download>

### **Microsoft Visual Studio**

Unity has a built-in IDE for editing C#-files (scripts). In our experience Microsoft's IDE was a better alternative, mostly because of the IntelliSense that helps the developer by providing auto-finishing of code and good explanations of the functions directly from the development environment. The Community Edition of Visual Studio is free and can be found at the link below.

<https://www.visualstudio.com/>

## F - Sketches

Throughout the exploration phase of the project we often used sketching as a tool to see how things would look on paper before trying to implement it. Drawing it on the computer or on paper takes a lot less time than to actually implement it. Also, making changes on paper is very "cost-efficient" opposed to making changes in the code.

### Social Forest

Below you can see a sketch where we tried to look at the possibilities of creating a social forest by using the Microsoft Kinect. The intention of this was that a screen would show a tree where leaves/branches grew out if people stood together in groups in front of the Kinect.

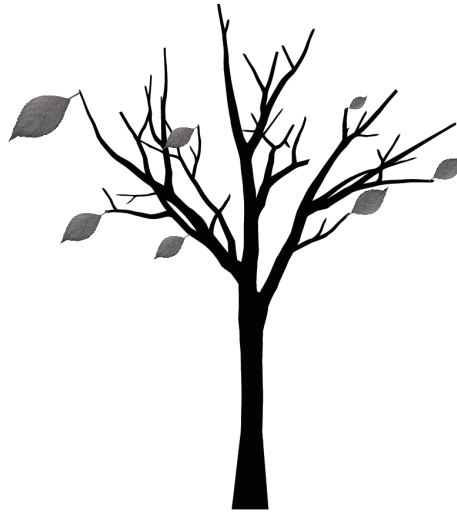


Figure 48: Sketch of a social forest.

### Graphs

We looked into using graphs and visual representations in order to uncover patterns in peoples behaviour on social media. Behavior that they might not be aware of themselves. An example of this could be a graph that shows other peoples activity levels on your wall, from this you might be able to find common interests. The first sketch shows activity-over-time during the scope of one day. This could possibly be used to find out when you should post to Facebook to

get more feedback. The second graph is a representation of which friends were the most active on a users posts (likes and comments).

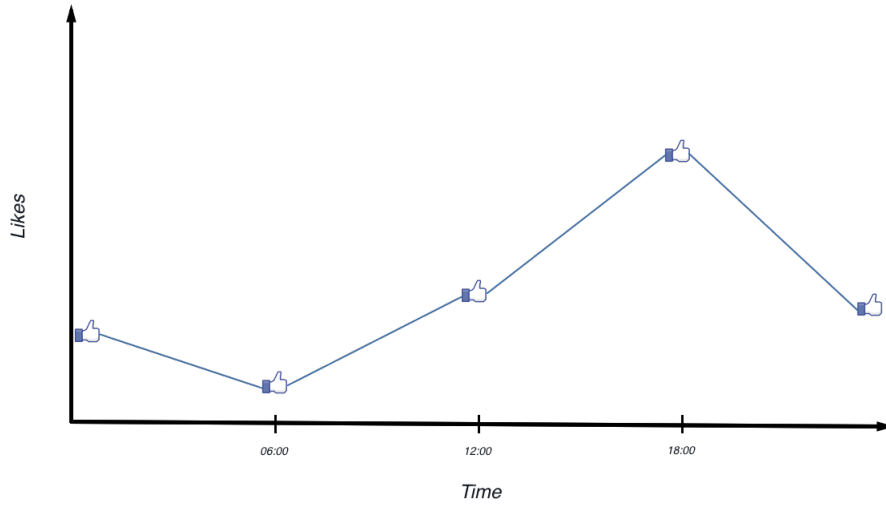


Figure 49: Sketch of a graph showing when during the day your friends likes your posts.



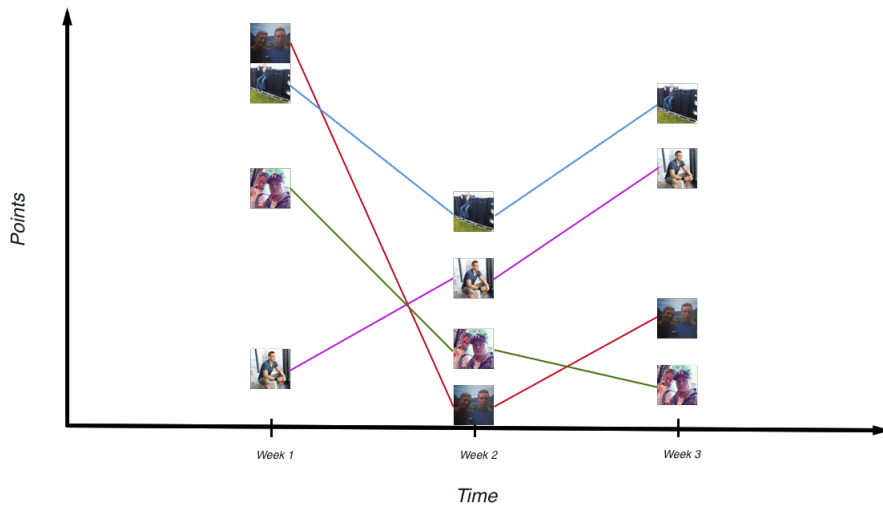


Figure 50: Sketch of a graph showing the most active friends on your posts.

# G - Preliminary Report

## Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Abstract . . . . .	3
1.2	Definitions . . . . .	4
<b>2</b>	<b>Project organization</b>	<b>5</b>
2.1	Project group . . . . .	5
2.1.1	Responsibilities for the project group - organization . . . . .	5
2.1.2	Responsibilities and tasks for the project manager . . . . .	5
2.1.3	Responsibilities and tasks for the secretary . . . . .	6
2.1.4	Responsibilities and tasks for other members . . . . .	6
2.2	Steering group . . . . .	6
<b>3</b>	<b>Appointments and agreements</b>	<b>7</b>
3.1	Agreements with project owner . . . . .	7
3.2	Workplace and resources . . . . .	7
3.3	Norms, rules and cooperation policy . . . . .	7
<b>4</b>	<b>Project description</b>	<b>9</b>
4.1	Problem approach . . . . .	9
4.2	Requirements for solution or project results . . . . .	9
4.3	Development methods . . . . .	10
4.4	Information retrieval . . . . .	10
4.5	Risk analysis . . . . .	11
4.6	Main activities . . . . .	11
4.7	Progress plan . . . . .	12
4.7.1	Main plan . . . . .	12
4.7.2	Management tools . . . . .	12
4.7.3	Development tools . . . . .	12
4.7.4	Internal control - evaluation . . . . .	12
4.8	Decisions . . . . .	13
<b>5</b>	<b>Documentation</b>	<b>14</b>
5.1	Reports and technical documents . . . . .	14
<b>6</b>	<b>Planned meetings and reports</b>	<b>15</b>
6.1	Meetings . . . . .	15
6.1.1	Meetings with the product owner . . . . .	15
6.1.2	Project meetings . . . . .	15
6.2	Periodic reports . . . . .	15

<b>7</b>	<b>Planned handling of deviation</b>	<b>16</b>
<b>8</b>	<b>Requirements for implementation and equipment</b>	<b>17</b>

# 1 Introduction

## 1.1 Abstract

Interactive art is the art that emerges from the interaction between an installation and an audience. In this project the group will make such an installation together with local artists and the supervisors at NTNU Ålesund. Interactive art has evolved with the technology, gaining momentum as the technology is becoming more and more available [1]. Interactive art has become increasingly popular within the academic community at NTNU.

This preliminary report focuses on a project given by local artists Terese Longva and Harald Dyrkorn together with NTNU Ålesund. Terese is a performance artist basing her work on the tension between the individual and society. Harald's work focuses on what the art does and how it can create new insights on society.

The main object of the project is to create an interactive art installation that makes an impression on the individual, rather than focusing on what technologies it takes to create it. We chose this project because of the interesting technologies and the possibilities they bring with them to create something special. Because the project is so open it gives the students a lot of freedom to explore usages of technology during the project.

## 1.2 Definitions

**EEG** Electroencephalography is a monitoring method to record electrical activity within the brain.

**Unity 3D** Unity is a cross-platform game engine often used to create interactive 2D and 3D content.

**Oculus Rift** The Rift is a head-mounted virtual reality display developed by Oculus. Oculus Rift has one OLED-display for each eye, each display having with a resolution of 1080x1200 resolution on 90 Hz.

**Microsoft Kinect** Kinect is a line of motion-sensing input equipment made by Microsoft. It was originally made for Xbox, but Microsoft has opened up the technology for developers to use it in other ways aswell.

**WalkMouse** A brand of omnidirectional treadmills, a mechanical device similar to a regular treadmill but allowing the user to walk in any direction. Used in virtual environments to allow movement within the virtual space.

**C#** C#(pronounced as see sharp) is a multi-paradigm programming language encompassing strong typing, imperative, declarative, functional, generic, object-oriented, and component-oriented programming disciplines.

**SDK** Software development kit. A set of software development tools that allows the creation of software for a certain software or hardware platform.

**API** Application programming interface. A set of routines, protocols and tools for building software and applications.

**NDA** Non-disclosure agreement. Legal contract that outlines confidential material.

**Scrum** Agile software development methodology using an iterative and incremental approach.

**Scrum team** Developers within a Scrum-based project.

**Scrum master** Responsible person within the Scrum team. Organizes meetings and coaches/helps the other team members.

**Daily stand-up** Short morning meeting where the the team goes through today's agenda.

**Product backlog** Prioritized list of task that has to be done in the project.

**Burndown graph** Graphical view of a sprint showing the completed tasks in the sprints.

## 2 Project organization

### 2.1 Project group

Student numbers
120874
110440
120848

#### 2.1.1 Responsibilities for the project group - organization

This project is very open, this entails a lot of work regarding the generation of ideas. This is especially true in the early phase of this project. Therefore the project group will need to do a lot of idea generation using creative techniques like brainstorming. The generated ideas will need to be explored and investigated so that the project group can determine their viability. This investigation will probably revolve around exploring different technologies and ways of implementing an idea. Throughout the project, the main bulk of work will concern programming, testing, documentation. The group will use a scrum-like project management technique. The main tasks will be split into smaller pieces and distributed among incremental releases also called sprints. The project group will be responsible for completing these sprints. The development team will be organizing themselves, this is one of the features of using an agile development method and makes it very suitable for smaller teams.

#### 2.1.2 Responsibilities and tasks for the project manager

In most agile development methods there is no clear role that encompasses the responsibilities of a project manager in a traditional waterfall model. In scrum one has a product owner and a scrum master, both of these roles carry certain elements and responsibilities shared with the traditional project manager role. Since we will not be using pure Scrum but rather a Scrum-like method we will give the project manager role elements from the product owner role. The university is essentially the product owner and because of the limited consulting time, we need to transfer some of the responsibilities from the product owner role to the project manager role.

Our project manager role will have the traditional responsibilities and also most of the responsibility of maintaining the product backlog, optimizing the priorities and adjusting the product backlog as the project evolves. The project manager will also be responsible for booking rooms and calling in to meetings with the product owner and the supervisors.

### 2.1.3 Responsibilities and tasks for the secretary

The secretary role will encompass the responsibilities of the scrum master. One of the main responsibilities will be aiding and making sure that the team is able to self organize. The secretary will also be responsible for improving the working conditions and routines based on feedback from the project team. The secretary will also prevent the team from getting interrupted, this will be accomplished by handling requests and most of the communicating with other invested parties.

The secretary will help the project manager maintain the product backlog, promote self organization, facilitate meetings and team events. Ensure that the team is following the agile method and help educate/inform if not.

### 2.1.4 Responsibilities and tasks for other members

The other members of the project team will be doing development mostly, but since agile development methods are agile they can also contribute and help where this is needed. We are a fairly small team consisting of three members and therefore the roles will be fairly fluid. The managing of the team will not be a enough work to warrant a dedicated person, therefore, everyone will be part of the development team. We will change around who is the project manager and secretary so that all the group members can have their go at the roles.

The roles of the development team members are mainly documenting, programming, testing.

## 2.2 Steering group

### **Institution:**

Norwegian University of Science and Technology (NTNU)

### **Advisors/Supervisors:**

Associate Professor Robin T. Bye

Associate Professor Anniken Th. Karlsen

Assistant Professor Arne Styve

### **External Assigners:**

Terese Longva

Harald Dyrkorn

## 3 Appointments and agreements

### 3.1 Agreements with project owner

We have agreed that that the project should be open to allow for creativity. To ensure that this creative process is going in the intended direction, the project group will have regular meetings and send out status reports. It is also agreed that our communication with the artists are being used as part of a research project.

### 3.2 Workplace and resources

Because the project was provided by NTNU Ålesund, our workspace will be at the university. Most of the time we will work from one of the computer labs. We have been informed by the supervisors that we get access to the visualization lab in order to have a workplace where we can store the equipment we are borrowing in a safe place and to use the WalkMouse placed in the lab.

The project group is allowed to request equipment that fits within the frame of the project. The supervisors will then consider if it is suitable and within the realms of reality to get it.

We will have access to professors, mainly our assigned supervisors. We have a limited amount of supervisor time, it will therefore be important to use this time as efficient as possible. We also have access to earlier work and articles through the library's computer system.

### 3.3 Norms, rules and cooperation policy

The project group will have a strong focus on delegating work in a way that lets members contribute to all topics of interest. This way the members can be credited the same amount of work and gain equal amount of experience across the different topics. We don't want to set any rules that hinders ideas and creativity. Being able to have open discussions is important when working on a creative project. Everyone in the project should be encouraged to take initiative. The open communication is also important for keeping everyone up to date, so all of us have a good overview of how the project is going.

For the programming collaboration we want to follow a coding standard which is a set of rules that the entire development team agree to adhere to throughout the project. So we can keep a consistent style and format of the source code. we also want to have collective code ownership which means everyone is responsible for all the code, this means everyone is allowed to change any part of the code. This goes well with pair programming which we might use. This will also ensure that if the project is used as a base for future projects



it will be easier to get an overview of the code. When using code found on the Internet this code will be properly marked as code that is not our own.

System metaphor is also something we want to use, so that its easy for everyone to understand how the system works give an explanation of how to system works. It also means it should be easy for a team member to guess what a function does. In other words try to make the system function in a logical way and use logical names.

This project has some extra challenges because two different schools of thinking has to find a way to coexist. On the one hand we have the artists, artists are able to generate ideas without being too constrained by outside forces. On the other hand we have engineers, engineers tend to think a bit differently and ask themselves whether or not implementation will be possible. Having these mindsets collaborate can lead to frustration. It is therefore very important to be open and accepting in order to facilitate a good working relationship.

The group has decided that everyone should be at school from 08.00 to 16.00 every day. If, for some reason, one of the members has to leave earlier due to work or other accepted reasons the other group members should be notified at least the day before. In the start of the semester Tuesdays and Wednesdays are not considered as work days due to lectures.

## 4 Project description

### 4.1 Problem approach

The challenge facing the project group is creating an art installation that will allow the individual to think of its place in society on a social level. This installation should be interactive, allowing the audience to input and change the form of the installation. In order to create this, the group will have to closely cooperate with the artists. The artists are very open to what form this installation could take, this allows the project group to explore the technological possibilities out there. One of the preliminary ideas involve using Facebook API to create a personal experience.

**Main goal:** Create an interactive art installation using the available technologies.

**Effect goal:** A fun to use installation that allows people to try the technology and maybe experiencing something deeper on an artistic level while having fun.

**Result goal:** A functioning, interactive installation with the possibilities to develop it further.

### 4.2 Requirements for solution or project results

The project has very loose boundaries, the product owner has not mentioned any specific requirements for the installation at this time. The requirements and specifications will hopefully start to solidify once the artists get a clearer picture of what can be done. The following topics has been discussed during our start up-meetings:

1. Create an art installation that interacts with the audience.
2. Explore the possible technological solutions and find out what technologies should be used in the project.
3. Use social aspects in the installation, for example making a group solve tasks, not the individual.
4. Uphold the financial frame set in the project (not more than approximately 5.000-10.000 NOK).
5. A Facebook-integration for the users would be preferable.

### 4.3 Development methods

The group is going to use a development method with many aspects taken from Scrum, and some of the aspects taken from Extreme Programming (XP).

One of the main features of Extreme Programming is programming in couples, where one more advanced programmer sits down with a less experienced programmer and they do the job together. This advantages both, since the less experienced developer gets continuous help and the experienced developer gets new insights from the less experienced developer.

Scrum is an agile development method that works well with innovative software development, creating iterative versions of the software. This iterative development of the software ensures that the project owner can input what they want underway making the end product what they want by adding features to the product backlog. Adding features or removing features underway becomes much easier and the group thinks that this is essential when programming an art installation where exploring the possibilities in the technology is a big part of the project. Scrum also helps the group set goals for each sprint (a time period of 2-4 weeks where certain goals has to be reached) by taking tasks from the product backlog and adding them to the sprint. After each sprint there is a sprint review meeting. Daily stand-ups is also a key part of Scrum, where the team members discuss their progress. One of the main reasons why we chose to use Scrum as a base for our development is because of the nature of the project. Choosing the Waterfall method would be difficult when we do not have a complete overview of the tasks ahead of us. According to the Waterfall method we would then have to start from scratch if something needs to change.

Read more about Scrum here:

<https://www.scrumalliance.org/why-scrum>

Since the technologies that we are going to use in the project are popular, there is loads of information available on the Internet to overcome roadblocks that may occur during the project.

### 4.4 Information retrieval

Using Rift, Kinect and other interactive technologies is something that has become more popular the last few years. Many fun and interesting installations has been developed, but there are many possibilities that has not yet been explored. For example has the Oculus Rift been used in bachelor projects before and there are people at the university with knowledge of how the technology works, so it is possible to get information and help. Both Oculus Rift and Microsoft Kinect has development forums where developers discuss different approaches to different problems, and these forums will be used. Also, using YouTube to get inspiration from what others have made and getting the artists view on this will make it easier to know that the project group and the artists want the same solution. Unity has a lot of tutorials made by the community, as well as free assets to complement them. This makes it a lot easier to get to know the environment the group will be working in.

## 4.5 Risk analysis

Because the project is as open as it is, one of the main risks is not having clear enough milestones. This could lead to the group not reaching the intended goals in time - and it is important to work closely with the supervisors and the artists in the beginning to get clearly defined milestones. If not, the group could lose morale and feel like the project isn't going anywhere. The project group defines this risk as a low risk.

Since we are dealing with a software development project where most of the work is done in code, there is a risk that a computer crash could set the group back. To prevent this the group will use Dropbox for the backup of files and GitHub for backup of files. In addition to this, the group intends to regularly take backups in the form of .zip-archives and put these on memory sticks that are not to be used other than to create and get backups. The project group evaluates loss of data with current controls as a low risk.

In general, the group will use the following strategy to solve problems that may occur during the project:

1. Identify the problem.
2. Invest resources in solving the problem.
3. Discover what caused the problem.
4. Make changes to prevent the problem from occurring again.
5. Put in extra work to make the effects of the problem as small as possible.

## 4.6 Main activities

No.	Main activity	Cost	Time
A1	Exploring possibilities	No cost	2 weeks
A2	Getting familiar with the technologies	No cost	8 weeks
A21	Exploring Oculus Rift	No cost	
A22	Exploring Microsoft Kinect	No cost	
A23	Testing Unity 3D	No cost	
A3	Setting up the installation components	No cost	1 day
A4	Developing the Kinect part of the software	No cost	4 weeks
A5	Developing the Rift part of the software	No cost	4 weeks
A6	Tie Rift and Kinect software together	No cost	4 weeks
A7	Exploring further possibilities	No cost	Continuous
A8	Finishing the report	No cost	2 weeks
A9	Make a project poster	No cost	1 weeks

## 4.7 Progress plan

### 4.7.1 Main plan

Since the project is yet to be defined in detail, the group does not have a list of activities yet. The main task is therefore the exploration phase and the discussion with the artists to come up with an art installation that satisfies their demands and find out what is possible to do in the time frame that the group has. Challenges and ideas will come as the project moves on the the next phases, and in this moment of time it is difficult to say what they will be. The goal will be to create an installation that is both interesting and artistic. The project group has to come up with ideas that will be presented during a meeting between the group, the artists (project owner) and the supervisors. Exploration of the different possibilities and trying to create a specification that is both interesting and challenging is one of the main activities.

### 4.7.2 Management tools

The group will use Gantt-diagrams during the course of the project to ensure that milestones are met and determine how the project is proceeding. Using these diagrams enables supervisors and project owners to easily get an overview. Having a clear and visible plan also increases the group's efficiency and betters the time management in general. To create these diagrams the group will use an online project management tool called TeamGantt.

<https://www.teamgantt.com>

JIRA will be used as a Scrum management tool to manage the sprints and the product backlog and create burndown-graphs.

### 4.7.3 Development tools

In terms of software, the group will use Visual Studio, Unity 3D and the development kits following the Microsoft Kinect and Oculus Rift to do the project. The group will also use a computer lent from NTNU Ålesund to have sufficient hardware to run the software since laptops are not fully supported with the Oculus Rift and Microsoft Kinect SDKs.

### 4.7.4 Internal control - evaluation

Internal control, according to the book *Praktisk Prosjektledelse* [2], is described as all systematic measures that the project owner or project organization does to ensure that activities are planned, organized, executed and maintained in accordance with the requirements and pursuant to law or regulation which group/supervisors have defined.

The project manager will have the responsibility to ensure that these demands are met and deadlines are held. After reaching a goal or completing a task, the project manager will evaluate the work done in relation to the criteria

that were set. To make it easier to see if a goal was met, clear goals must be established.

## 4.8 Decisions

To make good decisions in the start-up phase of a project is vital. It is cheaper to make changes when there has not been invested hours of work (and/or money in purchases). Changing things later on in the project could prove costly, and therefore the project group must carefully evaluate their options before starting with the programming because these decisions build the foundation for the rest of the project.

To ensure that good decisions are made, the project group will work closely with the project owners and the supervisors, especially in the start. Through planning and frequent meetings the group will do their best to make the right decisions the first time around. The group has not set any specific protocol of what is to be done when a decision has to be made, it all depends of the nature of the decision. The intention is that the group will have a good dialogue when decisions has to be made, and each member will get a chance to express their opinion. Since the different group members, superiors and project owners have different backgrounds, they will have different ways to approach the problems in the project. Opinions expressed by a person with an area of expertise within that specific topic will weigh more heavily than other opinions.

## 5 Documentation

### 5.1 Reports and technical documents

All written code should be written and commented in a way that makes it as easy as possible for an external developer to understand the code, and what each segment of the code does.

There will be a written project report that will go through all aspects of the project. The report will be written in a typical, scientific manner following the norms of technical and scientific reports. Both the preliminary report and the project report will be written in  $\text{\LaTeX}$ , a de facto standard for the communication and publication of scientific documents. We will use  $\text{\Share\LaTeX}$ , an online tool that allows collaboration and version control.

We have created templates for daily- and periodic reports in  $\text{\Share\LaTeX}$  that we will use throughout the project for a consistent and professional design of our documentation.

# 6 Planned meetings and reports

## 6.1 Meetings

### 6.1.1 Meetings with the product owner

The group will have frequent meetings with the supervisors to ensure progress in the project and that the group is going in the intended direction. The project manager will send out invites and book a room every other week, ideally on Tuesdays. The progress report will be an attachment to this invite, so that the supervisors get a chance to read it carefully before attending the meeting.

### 6.1.2 Project meetings

Following the Scrum development method, the project members will have a daily stand-up meeting to give each other an overview of what is on the agenda for the day. If one of the team members need help to complete their assigned tasks, the Scrum master will make sure that this problem is resolved after the meeting.

In addition to this the project members will have a status meeting once per week, ideally on Fridays. In this meeting the group will check how many of the tasks in the sprint has been completed, and to plan what the group will concentrate on in the week to come.

## 6.2 Periodic reports

Every second week the group will produce a progress report to update the supervisors on the project. This will save some time, since the supervisors will get this before the meetings and will be updated on what has been done. This way, more time is left to discuss the contents of the reports. The progress reports will focus on:

- Focus of work since last meeting.
- Planned activities during this period.
- Completed activities during this period.
- Justifications for deviations between planned and completed activities.
- Experiences made from this period.
- Focus for next period.
- Planned activities for next meeting.
- Guidance needs etc.



## 7 Planned handling of deviation

If the project goes in an unwanted direction adjustments will be made to change the direction of the project. The group will then try to investigate why the project was not working in the way that was intended. Since this is a rather open bachelor project which relies on the project group exploring and creating something artistic in collaboration with the artists, the result is very difficult to envision as of right now. Therefore the room for deviation will be bigger than a usual bachelor project where the group tries to fulfill specific demands set by the project owners. The group as a whole has responsibility to help turn the project turn around if deviations occur. The result itself will be flexible, but must be according to the ideas given by the artists. If the final product is not what the product owner wants it will not be used, making the whole project a failure.

## 8 Requirements for implementation and equipment

This project relies on equipment that the group borrows from NTNU Ålesund. Since all of the equipment needed to complete the project is equipment the university owns and we can borrow it without paying for it, it makes it an low cost project. For now - all required equipment is in place, everything is borrowed from the university.

### Software needs

- Visual Studio (*free*)
- Unity 5 (*free*)
- JIRA (*licensed to NTNU Ålesund*)
- ShareL<sup>A</sup>T<sub>E</sub>X (*free*)
- GitHub (*free*)
- TeamGantt (*free*)

### Hardware needs

- Oculus Rift DK2 (*lent from NTNU Ålesund*)
- Microsoft Kinect v.2 (*lent from NTNU Ålesund*)
- Desktop computer (*lent from NTNU Ålesund*)
- WalkMouse (*lent from NTNU Ålesund*)

# References

- [1] Eyal Gever. “The interrelationship and convergence of technology and art”. In: *IEEE Potentials* (November/December 2015), p. 11.
- [2] Johansen Rolstadås Olsson and Langlo. *Praktisk prosjektledelse*. Fagbokforlaget, 2014, p. 104.

# Appendixes

*This preliminary report has no appendixes.*

## H - Meeting Reports

The following pages will contain the reports for the meetings that has been held with the supervisors and/or external assigners.

# 2015-12-16 Bacheloroppgave kunstprosjekt kickoff

## Dato og sted

16 Dec 2015

Rom F423 kl. 10-13

## Deltakere

Harald, Anniken, Arne, Robin, Stein, Terje, Erik

Ikke tilstede: Terese

## Kontaktliste


Navn	Rolle	Epost	Telefon
Harald Dyrkorn	Oppdragsgiver	<a href="mailto:haralddy@online.no">haralddy@online.no</a>	473 47 859
Terese Longva	Oppdragsgiver	<a href="mailto:tereselongva@gmail.com">tereselongva@gmail.com</a>	941 80 100
Robin T. Bye	Veileder	<a href="mailto:roby@hials.no">roby@hials.no</a>	400 82 880
Anniken Karlsen	Veileder	<a href="mailto:ak@hials.no">ak@hials.no</a>	
Arne Styve	Ressursperson visualiseringlab	<a href="mailto:asty@hials.no">asty@hials.no</a>	905 60 299
Stein Sæter	Prosjektgruppe	<a href="mailto:Stein-Are.Saeter@stud.hials.no">Stein-Are.Saeter@stud.hials.no</a>	
Terje Eik	Prosjektgruppe	<a href="mailto:Terje.Eik@stud.hials.no">Terje.Eik@stud.hials.no</a>	
Erik Espenakk	Prosjektgruppe	<a href="mailto:Erik.Espenakk@stud.hials.no">Erik.Espenakk@stud.hials.no</a>	

## Agenda

Time	Item	Who	Notes
15 min	Orientering om bachelorprosjekt beskrivelse i studiehandbok	Anniken	<ul style="list-style-type: none"><li>• Krav</li><li>• Prosess</li><li>• Framdriftsrapporter</li><li>• Deltakere og roller</li><li>• Karakterer</li></ul>
5 min	Gjennomgang av opprinnelig oppgavetekst	Robin	
40 min	Presentasjon av kunstgruppa Bakgrunn og tematikk Definisjoner Hva skal vi lage?	Harald	<ul style="list-style-type: none"><li>• Kulturelle skolesekken -&gt; dataspill</li><li>• Litt om Terese Longva (kunstneren)</li><li>• Litt om Haralds bakgrunn (sosialantropologi, filosofi)</li><li>• Brev til Robin 21.8.2015: Sosial subjektivitet, mulige kunstverk, m.m.</li><li>• Høytlesning fra Thure Erik Lunds bok Compromateria</li><li>• Maskiniske sammenstillinger, maskinen = forbindelser</li><li>• Emergens, emergent behaviour (EB), sosial subjektivitet som EB og som større enn bestanddelene</li><li>• Hva er kunst?</li><li>• Paul Klee: Bli til, nytt, kreativitet</li><li>• Bindinger i ei gruppe av mennesker, diagram på tavla: sosiogram</li><li>• Kunsten skal føre til oppleving av noe (ikke nødvendigvis at du skal "skjønne" noe)</li><li>• Kjønn, status, alder, rolle</li><li>• Sosiale script</li></ul>
30 min	Lunsj		

	Diskusjon	Alle	<ul style="list-style-type: none"> <li>• Tanker og spørsmål så langt?</li> <li>• Møtet mellom kunst og teknologi</li> <li>• Virtuell møteplass i virtuell verden? Second Life?</li> </ul>
20 min	Teknologi	Robin	<ul style="list-style-type: none"> <li>• Presentasjon av ulike teknologier på tavla som kan være relevante: <ul style="list-style-type: none"> <li>• Second Life (virtual world)</li> <li>• Unity 3D (game engine) (Saleh Alaliyat, Siebe van Albada, Lars Ivar Hatledal, Robin, Arne)</li> <li>• Maya, Blender, etc. for 3D modellering (Paul Kleppe m.fl, Avd. for maritime operasjoner (AMO))</li> <li>• Processing + Java, visuell programmering (Anniken)</li> <li>• MS Kinect (Girts Strazdins)</li> <li>• Emotiv EEG (Robin)</li> <li>• HTC VR headset</li> <li>• Oculus Rift VR DK2 (Arne)</li> <li>• Arduino (Arne, Ivar Blindheim, Robin)</li> <li>• Visualiseringslaben (Arne)</li> <li>• Nao robot (Ibrahim Hameed)</li> <li>• UR2 roboter (Ottar Osen)</li> </ul> </li> </ul>
10 min	Diskusjon	Alle	<ul style="list-style-type: none"> <li>• Reaksjonsmønster på noe man skjer, f.eks. Kinect el. EEG oppfatter og tolker publikums reaksjoner</li> </ul>
5 min	Brainstorm om interaksjon	Robin	<ul style="list-style-type: none"> <li>• Publikum i interaksjon med egne skjermer vs. storskjerm, egne aktuatorer vs. felles aktuator</li> <li>• Hvordan lage "hjernen"/grensesnittet for kommunikasjon mellom publikum og kunstverk</li> </ul>
10 min	Diskusjon	Alle	<ul style="list-style-type: none"> <li>• Sosiale tabuer som ikke snakkes om, skambelagte ting</li> <li>• Social priming, psychological priming, James Ashley</li> </ul>
5 min	Prosjektverktøy	Arne	<ul style="list-style-type: none"> <li>• Atlassian software for prosjektsamarbeid <ul style="list-style-type: none"> <li>• JIRA (agile, scrum, etc.)</li> <li>• Confluence</li> <li>• Kobling mot Bitbucket</li> </ul> </li> </ul>

## Action items

- Robin T. Bye Sende "Brev til Robin" til studentene og medveiledere  16 Dec 2015
- Robin T. Bye Opprette en delt Dropbox (se over, legge brev, oppgavetekst osv. der  16 Dec 2015
- Robin T. Bye Lage ei kontaktliste og legge i Dropbox
- Bachelorgruppe:
  - Utforske mulige teknologier, søke inspirasjon
  - Tenke på hva som skal lages, formålet, hva kunstverket skal oppnå, osv.
  - Begynne på utkast til mer konkretisert prosjektbeskrivelse og tenke på forprosjekt
  - Sende ideer, tankespill, notater til kunstnergruppa og veiledere for tilbakemelding
  - Sende liste over telefonnummer slik at kontaktlista blir oppdatert
- Arne Styve Vise fram visualiseringslaben til studentene en gang etter jul
- Arne Styve Oppretter JIRA og Confluence prosjekt
- Anniken: Sende telefonnummer slik at kontaktlista blir oppdatert

# Møtereferat - fredag 12. februar

Sted: NTNU Ålesund, A436

Dato: 12.02.2016 kl 10.00-14.00

Deltakere: Terje Eik, Stein-Are Sæter, Erik Espenakk, Terese Longva, Harald Dyrkorn, Anniken Karlsen, Robin T. Bye

## Kontaktliste

Navn	Rolle	Epost	Telefon
Harald Dyrkorn	Oppdragsgiver	haralddy@online.no	473 47 859
Terese Longva	Oppdragsgiver	tereselongva@gmail.com	941 80 100
Robin T. Bye	Veileder	robin.t.bye@ntnu.no	400 82 880
Anniken Karlsen	Veileder	anniken.t.karlsen@ntnu.no	907 41 704
Arne Styve	Ressursperson VisLab	asty@ntnu.no	905 60 299
Erik Espenakk	Prosjektgruppe	erik.espenakk@stud.hials.no	988 97 550
Terje Eik	Prosjektgruppe	terje.eik@stud.hials.no	992 59 270
Stein Sæter	Prosjektgruppe	stein-are.saeter@stud.hials.no	469 65 436

## Agenda

Tid	Tema	Person	Notater
15 min	Generell informasjon	Stein Sæter	<ul style="list-style-type: none"><li>• Informasjon om hva som skal gåes igjennom</li><li>• Informasjon om hva som har blitt gjort i prosjektet siden sist</li></ul>
45 min	Demo på VisLab	Prosjektgruppe	<ul style="list-style-type: none"><li>• Fremvisning av demoer for å vise spekteret dette kan brukes til<ul style="list-style-type: none"><li>• Back to Dinosaur Island (for å vise Oculus)</li><li>• I Expect You To Die (for å vise interaktivitet)</li></ul></li><li>• Fremvisning av Kinect</li><li>• Vise enkle ting laget i Unity</li><li>• Vise Facebook-integrasjonen vi har laget</li></ul>
120 min	Gjennomgang av felles mål med prosjektet	Harald Dyrkorn	<ul style="list-style-type: none"><li>• Menneskets oppførsel alene og i flokk</li><li>• Mennesket er flere<ul style="list-style-type: none"><li>• "Since both of us were several, there was already quite a crowd"</li><li>• Noe skjer med mennesket, bevisst eller ubevisst når det er en del av en mengde</li></ul></li><li>• Diskusjon av mulige veier å gå på<ul style="list-style-type: none"><li>• Facebook vs ikke-Facebook</li></ul></li><li>• Diskusjon av utstyr mtp. at Terese vil ha med seg dette som en flyttbar installasjon</li></ul>
40 min	Lunsj		
25 min	Oppsummering		

## Oppfølgingsoppgaver

- Stein-Are Sæter: Lag møtereferat og send ut mail med link til Confluence
- Stein-Are Sæter: Lag oversikt over spesifikasjoner som trengs av utstyr og foreslå utstyr
- Erik Espenakk: Lag oversikt over møter fremover



# Møtereferat - mandag 7. mars

Sted: NTNU Ålesund, A436

Dato: 07.03.2016 kl 10.00-14.00

Deltakere: Terje Eik, Stein-Are Sæter, Erik Espenakk, Terese Longva, Harald Dyrkorn, Anniken Karlsen, Robin T. Bye, Arne Styve

## Kontaktliste

Navn	Rolle	Epost	Telefon
Harald Dyrkorn	Oppdragsgiver	haralddy@online.no	473 47 859
Terese Longva	Oppdragsgiver	tereselongva@gmail.com	941 80 100
Robin T. Bye	Veileder	robin.t.bye@ntnu.no	400 82 880
Anniken Karlsen	Veileder	anniken.t.karlsen@ntnu.no	907 41 704
Arne Styve	Ressursperson VisLab	asty@ntnu.no	905 60 299
Erik Espenakk	Prosjektgruppe	erik.espenakk@stud.hials.no	988 97 550
Terje Eik	Prosjektgruppe	terje.eik@stud.hials.no	992 59 270
Stein Sæter	Prosjektgruppe	stein-are.saeter@stud.hials.no	469 65 436

## Agenda

Tid	Tema	Person	Notater
45 min	Status	Stein Sæter	<ul style="list-style-type: none"><li>Gjennomgang av hva som har blitt gjort</li><li>Fremvisning av det som har blitt gjort</li><li>Forklaringer - hvorfor ble ting gjort som det ble</li></ul>
15 min	Demo på VisLab	Prosjektgruppe	<ul style="list-style-type: none"><li>Viste frem interaksjon med lommelykt og klo</li><li>Viste litt bruksområder på Kinect</li></ul>
60 min	Veien videre	Alle	<ul style="list-style-type: none"><li>Diskuterte muligheter, ville spikre prosjektet idag<ul style="list-style-type: none"><li>Togreise</li><li>Bruk av mobil?</li><li>Social priming?</li></ul></li></ul>
30 min	Lunsj		
25 min	Veien videre 2	Alle	<ul style="list-style-type: none"><li>Bestemmelser<ul style="list-style-type: none"><li>Lage en testbar versjon av kunstapplikasjonen</li><li>Bruke togreise og interaksjon med Kinect og Oculus Rift</li><li>Ha "stasjoner" som toget stopper på, her skal brukeren igjennom en øvelse</li><li>Brukeren må fullføre øvelsen før toget går videre</li></ul></li></ul>

## Oppfølgingsoppgaver

- Stein-Are Sæter: Lage en informasjon til kunstnerne med informasjon som kan hentes fra Facebook slik at de kan generere innhold til stasjonene
- Harald Dyrkorn: Bruke informasjonen til å lage ideer til stasjonene

# Møtereferat - tirsdag 19. april

Sted: NTNU Ålesund, F420

Dato: 19.04.2016 kl 13.00-14.00

Deltakere: Terje Eik, Stein-Are Sæter, Erik Espenakk, Anniken Karlsen, Robin T. Bye, Arne Styve

## Kontaktliste

Navn	Rolle	Epost	Telefon
Robin T. Bye	Veileder	<a href="mailto:robin.t.bye@ntnu.no">robin.t.bye@ntnu.no</a>	400 82 880
Anniken Karlsen	Veileder	<a href="mailto:anniken.t.karlsen@ntnu.no">anniken.t.karlsen@ntnu.no</a>	907 41 704
Arne Styve	Ressursperson VisLab	<a href="mailto:asty@ntnu.no">asty@ntnu.no</a>	905 60 299
Erik Espenakk	Prosjektgruppe	<a href="mailto:erik.espenakk@stud.hials.no">erik.espenakk@stud.hials.no</a>	988 97 550
Terje Eik	Prosjektgruppe	<a href="mailto:terje.eik@stud.hials.no">terje.eik@stud.hials.no</a>	992 59 270
Stein Sæter	Prosjektgruppe	<a href="mailto:stein-are.saeter@stud.hials.no">stein-are.saeter@stud.hials.no</a>	469 65 436

## Agenda

15 min	Generelt om fremgangen i prosjektet	Prosjektgruppe	<ul style="list-style-type: none"><li>• Viste frem rollercoaster</li><li>• Viste frem litt hva Facebook-data kan gjøre med opplevelsen</li></ul>
15 min	Gjennomgang av rapport	Alle	<ul style="list-style-type: none"><li>• Forklarte oppbygning</li><li>• Gikk igjennom oppsett</li></ul>
30 min	Generell diskusjon	Alle	<ul style="list-style-type: none"><li>• Gikk igjennom forbedringer på rapport</li><li>• Gikk igjennom hva som står igjen programmeringsmessig</li></ul>

# I - Sprint Reports

The following pages will contain the sprint reports sent out before each meeting with the supervisors and external assigners.

# Sprint Report

Erik Espenakk | Terje Eik | Stein-Are Sæter  
Artifunctional@gmail.com | Terje\_Eik@hotmail.com | SteinSater@gmail.com

21/01/2016

## This sprint's focus

Focus for this period was getting the documentation process up and running, creating good templates and routines. A secondary focus was exploring the possibilities available to us on the technology front, Oculus Rift, Microsoft Kinect. We also did some idea generating using Processing and explored Unity as a possible platform.

## Activities planned

Research relevant technologies	Oculus Rift, Kinect, Simulation lab
Read up on related programming languages	Processing, C#, Java
Develop templates for documentation	L <sup>A</sup> T <sub>E</sub> X
Research and pick technologies for	Version control

## Activities followed through

Research relevant technologies	Oculus rift, Kinect, Simulation lab
Read up on related programming languages	Processing, C#, Java
Develop templates for documentation	Latex
Research and pick technologies for	Version control
Searched and studied literature	AI-Art, AI-Social interaction, Art and technology

## Reason for discrepancies

No discrepancies, we were able to follow through on all our planned activities.

## Changes needed and why

We require no changes in the current plan, method, project content.

## Experiences gained from this period

We have acquired some experience in L<sup>A</sup>T<sub>E</sub>X, and learned how to modify/make your own classes and templates. We have gotten insight into Unity, Oculus Rift, Microsoft Kinect and the programming language C#. We have made our selves familiar with linear interpolation and setting up a multiplayer environment in Unity.

## Focus for the next sprint

Establishing closer communication with the artists/assigners, generating ideas and sketching up prototypes for the artists to review. Working in collaboration with the artists to establish a more clear goal and making the assignment more concrete. Our goal is to now get a clear overview of what we will create during the project and set up sprints etc.

## Next sprint's activities

Establish a common goal with the artists	generating ideas, prototyping
Acquiring/picking essential tools	Desktop computer, JIRA, GitHub
Selecting development methodology	SCRUM, Extreme programming, Agile
Relocate our working environment	visualization lab?
Specify project goal	

## Other information/requests

Still waiting for desktop PC, waiting for Marius Fylling to get in touch with us. We have reminded him earlier this week (19.01.2016) and he promised to get back to us as soon as possible. According to him he was waiting for the computer to be freed from where it was used earlier.

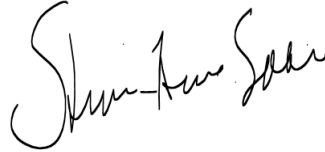
## Signatures



Erik Espenakk  
Technical officer



Terje Eik:  
Creative officer



Stein-Are Sæter:  
Administrative Officer

# Sprint Report

Erik Espenakk | Terje Eik | Stein-Are Sæter  
Artifunctional@gmail.com | Terje\_Eik@hotmail.com | SteinSater@gmail.com

12/02/2016

## This sprint's focus

Focus for this period was exploring ways to collect information from Facebook. We also had a secondary focus, generating ideas as to what kind of interaction one can create using the provided Facebook information.

## Activities planned

Establish a common goal with the artists	generating ideas, prototyping
Acquiring/picking essential tools	Desktop computer, JIRA, GitHub
Selecting development methodology	SCRUM, Extreme programming, Agile
Relocate our working environment	visualization lab?
Specify project goal	

## Activities followed through

Establish a common goal with the artists	generating ideas, prototyping
Acquiring/picking essential tools	Desktop computer, JIRA, GitHub
Selecting development methodology	SCRUM, Extreme programming, Agile
Relocate our working environment	visualization lab
Set up a system for collecting Facebook information using relevant technologies	Facebook-GraphAPI, Python, Python Facebook-SDK
Develop prototype for a mini game/interaction	Sketch App, Unity
Researching and exploring different ways of automatically generate a game/scene environment	Perlin noise
Loading Facebook information into Unity	JavaScript object notation
Getting the Microsoft Kinect to work	

## Reason for discrepancies

Due to getting access to the desktop computer we now have the Microsoft Kinect 2.0 up and running. This opens up for a lot of opportunities regarding interaction. Due to not having fully decided upon a development methodology and this project being a open and creative in nature, we have had some discrepancies. However in this case the discrepancies have been of a positive nature, as we have been able to get more work done than expected.

## Changes needed and why

We require no changes in the current plan, method, project content.

## Experiences gained from this period

We have gained experience in dealing with Facebook Graph API and Json "JavaScript object notation". We have also learned about terrain generation using noise algorithms like Perlin noise. . We have also gained some insight into our group's dynamics as we have started implementing an agile development methodology.

## Focus for the next sprint

Establishing closer communication with the artists/assigners, generating ideas and sketching up prototypes for the artists to review. Working in collaboration with the artists to establish a more clear goal and making the assignment more concrete. Our goal is to now get a clear overview of what we will create during the project and set up sprints etc.

## Next sprint's activities

Further develop mini game concept	Comment quiz..
Guided tour of your online life?	Picture roller-coaster/train

This part is a topic of discussion for the meeting with the artists on Friday the 12th of February. We aim to do some brainstorming involving what we can create with the content we gather from Facebook Graph API to put this more into detail.

## Other information/requests

All of the equipment we have gotten is now in place and is fully functioning.

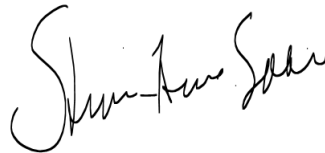
## Signatures



Erik Espenakk  
Creative officer



Terje Eik:  
Administrative officer



Stein-Are Sæter:  
Technical officer

# Sprint Report

Erik Espenakk | Terje Eik | Stein-Are Sæter  
Artifunctional@gmail.com | Terje\_Eik@hotmail.com | SteinSater@gmail.com

26/02/2016

## This sprint's focus

Focus for this period was exploring ways to use information from Facebook. Our secondary focus was implementing a way for the user to create an access token and gather the information without having to put the variables manually into the code.

## Activities planned

Further develop mini game concept	Comment quiz..
Guided tour of your online life?	Picture roller-coaster/train

## Activities followed through

Create a "scoreboard" from Facebook interaction	Ordering information
Explored interaction via Kinect	Grabbing objects in game
Implemented smoothing to counteract unprecise sensor data	Kinect, unity, linear interpolation
Migrated to Jira	Maintaining the product backlog ,Managing sprints
Investigation path generation	Bézier curves
Set up a webpage with Facebook login to gain an access token	Facebook-GraphAPI, Python, Python Facebook-SDK, HTML, CSS, JavaScript
Visualized Facebook posts with graphs and photos	Unity
Wrote the first version of the method section in the report	L <sup>A</sup> T <sub>E</sub> X

## Reason for discrepancies

We decided to prioritize the automation of creating access tokens etc. over developing certain mini games. We also visualized the content rather than creating specific games to see if this could give us some ideas of new content to be created next sprint.

## Changes needed and why

We have considered migrating our version control from github to jira.

## Experiences gained from this period

We have gained experience with creating access tokens via JavaScript (cross-browser, cross-operating system) and downloading the users credentials. We have also researched ways of generating a smooth path using Bézier curves, this could be very useful if we proceed with the roller coaster idea. We have learned more about the different data-streams available to us via the Kinect and the methods used for detecting gestures. There is also potential for training neural networks with the Kinect data.



## Focus for the next sprint

For the next sprint our main focus will be the contents of the virtual world. We have the basis now to let the user log in, and this is a natural next step. Using the Oculus Rift and the Kinect we will develop more content.

## Next sprint's activities

Further develop mini game concept	Investigate more ways show the data
Implement a path using Bèzier curves	Picture roller-coaster/train
Investigate use of Kinect to control a flash-light	Night gallery
Explore ways of integrating AI into the project.	Genetic algorithms, Neural networks, Agent based systems, Ant colony optimization

## Other information/requests

To create better interaction with the users it would be better to use for example Razer Hydra, Leap motion sensor or STEM motion controller. Is this something that is possible to get through the school?

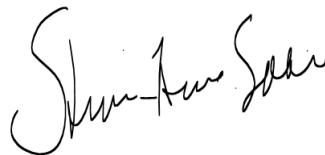
## Signatures



Erik Espenakk  
Administrative officer



Terje Eik:  
Technical officer



Stein-Are Sæter:  
Creative officer

# Sprint Report

Erik Espenakk | Terje Eik | Stein-Are Sæter  
Artifunctional@gmail.com | Terje\_Eik@hotmail.com | SteinSater@gmail.com

15/04/2016

## This sprint's focus

The focus for this (very long) period has been to build up the rollercoaster. We have used a lot of time to figure out path generation, better terrain generation and visual effects. This is the foundation for the rest of the project, only having the "social experience games" on the stations remaining after this.

## Activities planned

Path generation	Automated, individual paths
Fractal Terrain generation	Using the built-in Perlin noise in Unity's Mathf library
Height based coloring of the Terrain	Building a simple shader that will color the map using the heightmap data
Generate a station	Build a station where the rollercoaster will stop
Get 3D models	Download free models or make models for the cart, light fixtures etc
Work on the final report	Write about what we've done so far

## Activities followed through

Path generation	Using Catmull-Rom splines
Mesh generation	Generate meshes along the path (vertices, triangles, normals).
Fractal terrain generation	Using the built-in Perlin noise in Unity's Mathf-library
Height based coloring of the Terrain	Building a simple shader that will color the map using the heightmap data
Visual improvements	Early implementation of water
Station generation	Created assets to use in the scene
Tree generation	Created 3D model of a tree to replicate a forest
Code cleanup and re-factoring	Removed code that was used for testing, cleaned up file names.
Explored different uses for the Facebook data	Using numbers like "amount of friends, likes, comments" to generate terrain or railway using this
Merged the Kinect interaction with the roller-coaster experience	Testing different ways of interaction while seated in a "cart"
Report in the report	Wrote a lot in the theory, method and discussion in the report

## Reason for discrepancies

This period has been way too long. This is mainly our own fault, but we have had some events to set back the reporting and documentation via JIRA by having the server unavailable. The work that has been done in the last few weeks also took a long time, we had to read up on different subjects to have enough knowledge to create code to solve the problems. Also, having the last exam and Easter

kept us from working on the project for a while. We are now back up and running in JIRA and will continue with the reports as the plan was, every second week.

## Changes needed and why

We have considered substituting JIRA/Confluence with another solution due to the hard drive crash that happened, but decided to continue using JIRA/Confluence.

## Experiences gained from this period

We have gained experience with path generation (both using Bèzier curves and Catmull-Rom), mesh. We've also looked into how to increase graphics performance using LOD (2D sprites instead of 3D models at long distances) and generating meshes.

## Focus for the next sprint

For the next sprint our main focus will be the contents of the station. The virtual experience using paths generated to each individual, the terrain and most of the visuals in the environment are now in place.

## Next sprint's activities

Clean up personal data	Remove the users data after terminating the application
Improve graphics performance	Follow optimization guides
Create lakes	Layer of water-like material on top of lakes
Create billboard shader for 2D sprites	To improve graphic performance
Explore ways of integrating AI into the project.	Genetic algorithms, Neural networks, Agent based systems, Ant colony optimization

## Other information/requests

Nothing to note.

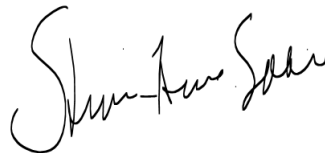
## Signatures



Erik Espenakk  
Administrative officer



Terje Eik  
Technical officer



Stein-Are Sæter  
Creative officer

## J - Project Poster

The next page contains the project poster, created to give a brief insight to the project.

## Introduction

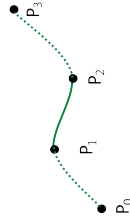
The artistic idea behind the project is about the production of social subjectivity and human col-lectivities within contemporary society. Normally we humans think that we are individuals with free will that can move, think and talk as we want to, but then we notice that there are social strains, and that our movements and whereabouts are strongly affected by others like friends, family and governmental bodies. We are continuously being measured, reg-istered and controlled. Social media, television and electronic networks are all influential to how we in-teract in different arenas, and may also exert pres-sure or force upon us to take part in things we might not want.

The thesis investigates this problem using Unity 3D for Oculus Rift and Microsoft Kinect by using Face-book data to generate a terrain, a rollercoaster track and mini games.

## Theory

### Catmull-Rom Splines

A spline is a numeric function which possesses a high degree of smoothness where the polynomial pieces connect. Catmull-Rom splines were formulat-ed by Edwin Catmull and Raphael Rom according to the work of Barry and Goldman. It is a interpolating spline going through four defined control points  $P_0$ ,  $P_1$ ,  $P_2$  and  $P_3$ , only drawing the curve from  $P_1$  and  $P_2$ .



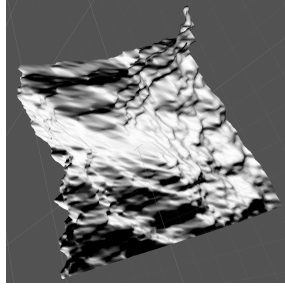
### Perlin Noise

Perlin noise was first created by Ken Perlin in 1983. He later published his article "An Image Synthesizer" at the SIGGRAPH 1985 conference, in which he de-scribes his algorithm. Perlin noise is a form of gradi-ent noise. Gradient noise is comprised of the gradi-

ents created by linearly interpolating between points in an arrangement of random or pseudo-random values.

### Fractional Brownian Motion

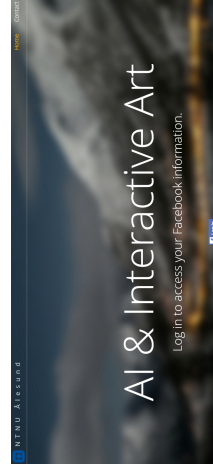
Fractional Brownian Motion (Brown noise) is created when combining several layers of noise, commonly referred to as octaves. Each octave will have a higher frequency and lower amplitude than the preceding octave. This is shown below, using 8 octaves of Per-lin noise to create a terrain.



## Results

### Information Retrieval

Due to the enormous amount of data available and the support of retrieving the information via Face-book Graph API, Facebook was chosen as the source of information.



The user signs in to a web page and when the appli-cation is launched a set of Python scripts are run. These scripts use the credentials downloaded from the web page to build a catalog of information. This mainly focuses on the users' own posts and the in-teraction on these posts. Posts on Facebook has a lot

of data attached to them, including the type of post (link, picture, text etc.), pictures, geographical in-formation, timestamps, likes and comments. Infor-mation is then organized, giving a total score of how much interaction a specific Facebook friend has had with the player's posts.

The scripts also download the profile pictures of the friends that has interacted to the posts, giving the possibility to use these to create a personalized look and feel of the environment. The parameters used from Facebook in the application are comments, likes and numbers of years active on Facebook. In addition to this there is a lot of information available that is not used. To get a full overview, see Face-books docs of the API: <https://developers.facebook.com/docs/graph-api>

### Application

Due to using the users hand for controlling a Kinect, a path is generated using Catmull-Rom splines to move the user through the environment. The path is drawn like a roller coaster and the user is positioned in a cart. The cart is moved along the path, using physics to increase/decrease the velocity. Brown noise is used generate the terrain. This is also implemented to support infinite terrain generation, giving the possibility to move an i finite amount of steps in a direction.



The data from Facebook is also used to create a mini game where the user interacts with the appli-cation via Kinect. The profile pictures of their five top friends (according to activity) on Facebook are shown and the user has to pick up corresponding

trophies (marked with #1, #2 and #3) and thrown them at the targets. When all targets are hit the roll-er coaster moves on to the next stop: a word cloud of the most used words in the users posts and com-ments on the posts.

## Conclusion

Being part of a project where the goal is not clearly defined in the start but rather found along the way has been a challenge and an interesting experience. It's been a challenge to acquire enough knowledge about the available technologies to make a decision whether this would be a useful supplement to the project or if it should be put to side. It's very prob-able that this will resemble projects in jobs in the future – because no matter how detailed the require-ments will be, trying out different technologies to satisfy the demands will be a key part of the projects. Many aspects of the project have been unfamiliar to us and we have had to learn the fundamentals with-in many disciplines. Game development (includ-ing both behavior of objects and generating visual content), virtual reality and interactive technologies (Microsoft Kinect) are some examples of disciplines where we had to start by learning the basics. This has been an interesting challenge and developing an application that takes advantage of these technolo-gies has been very motivating since 2016 will most probably be the birth year of consumer virtual real-ity experiences with the release of Oculus Rift and HTC Vive.

## Further Reading

- E. Catmull and R. Rom, "Computer aided geo-metric design", RE Barn, 1974.
- B. Mandelbrot and J. W. V. Ness, "Fractional brownian motions, fractional noises and appli-cations", SIAM: Society for Industrial and Ap-plied Mathematics, 1968.
- A. Cooper, R. Reimann, and D. Cronin, "About face 3: The essentials of interaction design. 3rd", Indianapolis, Indiana, published simul-taneously in Canada: Wiley publishing, inc,