NTNU – Trondheim
Norwegian University of
Science and Technology

Interaction Design in a 4D sea map

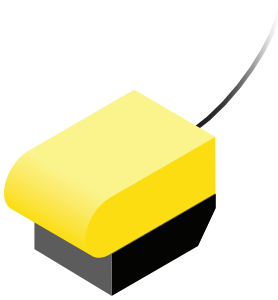
Stian Surén

Industrial Design Engineering
Submission date: June 2015
Supervisor: Thomas Porathe, IPD

Norwegian University of Science and Technology
Department of Product Design

Master thesis written by Stian Surén, Department for Product Design,
Norwegian University of Science and Technology, Spring 2015

INTERACTION DESIGN IN A 4D SEA MAP



SAMMENDRAG

BAKGRUNN
ROV er er i ferd med å bli en stor del av den maritime industrien idag. Når bedrifter skal prøve å utnytte mer av potensialet i subsea-industrien, er ROV ene tilstede for å ta på seg oppgaver flere hundre meter under havoverflaten.

ROV operasjoner blir stadig mer komplekse, med mange ROV er og fartøy som samarbeider på store dyp. Nedetiden til en oljerigg og leie av subsea-utstyr er dyrt. Med en sømløst planlagt operasjon, er det mye aktørene i markedet har å spare.

MÅL
Målet har vært å utvikle et konsept for å navigere virtuelle ROV er gjennom en 3D-representasjon av en reell undervannslokasjon. Ved å lagre banen til ROV en, kan man planlegge operasjonen i detalj. Hvis banene er realistiske, kan operasjonen bli kortet ned i tid. Tidsaspektet er interaktivt, og tilsvarer den fjerde dimensjonen i sjøkartet.

En viktig del var å undersøke om en HMD slik som Oculus Rift kan brukes for et slikt formål. Forskjellige typer kontrollere ble også testet.

METODE
Oppgaven har hatt en brukersentrert designtilnærming. Brukerinnsikt, testing og dybdeintervjuer er grunnlaget for et endelig konsept. Brukere har blitt analysert, og personas har blitt laget som et resultat. Brukerreisen har blitt visualisert gjennom scenarier.

Funksjonelle prototyper laget i spillutviklingsplattformen Unity ble mye brukt til brukertesting.

RESULTAT
Innsikt har vist at det ikke finnes en industristandard for hvordan en ROV operasjon blir planlagt. Sikkerhet kommer først, og det er mange regulasjoner som sikter på å standardisere sikkerhetstiltak. ROV operasjoner blir utsatt eller kansellert på grunn av skiftende værforhold. ROV piloter må ha god stedsans når de navigerer i mørket på store dyp. Det hender at piloter kjører seg vill, og det er dyrt.

Oculus Rift har blitt testet. Med effektivitet som målestokk, indikerer testen små forbedringer med Rift sammenlignet med en vanlig dataskjerm ved navigering i en 3D-verden. Bakdelene ved å bruke Rift blir diskutert, og det blir konkludert med at fordelene ikke overvinner bakdelene.

Flere kontrollere ble testet. En PS3-kontroller og en joystick blir anbefalt som de beste alternativene for å navigere i en 3D-verden. Keyboardkontroll blir også foreslått, på bakgrunn av tilgjengelighet.

Resultatet er et konsept for en skrivebordsapplikasjon som lar ROV supervisors planlegge en ROV operasjon ved å lage baner for de involverte fartøyene og ROV ene. Havstrømmer og navlestrengen til ROV en blir simulert for å være realistiske.

ABSTRACT

BACKGROUND
ROVs are becoming a huge part of the maritime industry today. As the industry is looking to exploit more of the subsea potential, ROVs are there to take on difficult tasks several hundred meters below sea level.

ROV missions become increasingly more complex, with several ROVs and vessels cooperating at great depths. The down time cost of an oil rig and the renting rates of subsea equipment makes ROV missions expensive. With a tightly planned mission, actors in the subsea market have a lot to save.

OBJECTIVE
The goal has been to develop a concept for navigating virtual ROVs through a 3D representation of an actual subsea site. By saving the trajectories, the mission can be planned in detail. If the trajectories are realistic, then operations can be compressed to shorter periods of time. The time aspect is interactive, and is the fourth dimension of the sea map.

An important part was to research if an HMD such as the Oculus Rift could be used for such a purpose. An array of controllers were also tested.

METHOD
The thesis used a user centered design approach. User research, testing and depth interviews are the basis for the final concept. Users have been analyzed, and personas created as a result. The user journey has been visualized through scenarios. Functional prototypes created in the game development platform Unity have been widely used for user testing.

RESULT
Research showed that there is no industry wide standard to plan an ROV mission. Safety comes first, and there are many regulations aimed at standardizing safety procedures. ROV missions get delayed or cancelled by shifting weather conditions. ROV pilots need to have a well developed sense of direction, as they navigate in the darkness at great depths. It occurs that pilots lose their way in the deep, which is expensive.

The Oculus Rift has been tested. When effectiveness is measured, the test indicates a small improvement with the Rift compared to a regular computer screen as the user navigates a 3D-world. The drawbacks of using the Rift are discussed, and it is concluded that, at present time, the advantages do not overcome the discussed drawbacks.

Several controllers were tested. The PS3 controller and a regular gaming joystick are recommended as the best alternatives for navigating in the 3D world. For the sake of availability, keyboard controls are also proposed.

The result is a concept for a desktop application that enables ROV supervisors to plan an ROV mission by creating trajectories for the involved vessels and vehicles. Ocean current and the tether from the ROV is simulated to add realism.

PREFACE

This master thesis has been written at the Department for Product Design at the Norwegian University of Science and Technology during the spring of 2015.

I would like to thank Thomas Porathe for the initial thesis idea and for guidance throughout the period. Additionally, I thank Trond Are Øritsland for the introduction to Unity and help along the way. Thank you also to the Department of Product Design for ordering whatever gadget I wanted.

Bekk Consulting AS deserves a thank you for lending me office space, providing unlimited access to coffee, and also giving me a few tips and tricks.

Thank you to David Knutsen at Oceaneering for giving me valuable insight.

A big thanks is sent to those who were patient enough to endure my user tests, and follow up with great feedback.

Thank you to my friends and family for support and input along the way.

Thank you to my girlfriend. Your encouragement has been greatly appreciated.

The last tribute is given to my dear classmates for making the five years at NTNU a great experience.

CONTENT

1 BRIEF

>	Introduction	13
1	Background	15
2	General	19

2 METHODOLOGY

>	Introduction	23
1	Approach	25
2	Research	29
3	Analysis	33
4	Concept Development	35

5 THE USERS

>	Introduction	83
1	The Users	85
2	User Scenario	93

6 CONCEPT DEVELOPMENT

>	Introduction	105
1	Within-Group Design	107
2	Prototyping & Testing	109
3	Controllers	111
4	HMD vs Screen	119
5	On Immersion	139

3 PRIMARY RESEARCH

>	Introduction	41
1	The ROV	43
2	ROV Missions	47
3	Controlling an ROV	49
4	Navigation & GUI	55

4 SECONDARY RESEARCH

>	Introduction	65
1	Head Mounted Displays	67
2	Reality vs Virtuality	69
3	Cognitive Load	71
4	Situation Awareness	75
5	Useful Field of View	79

7 FINAL CONCEPT

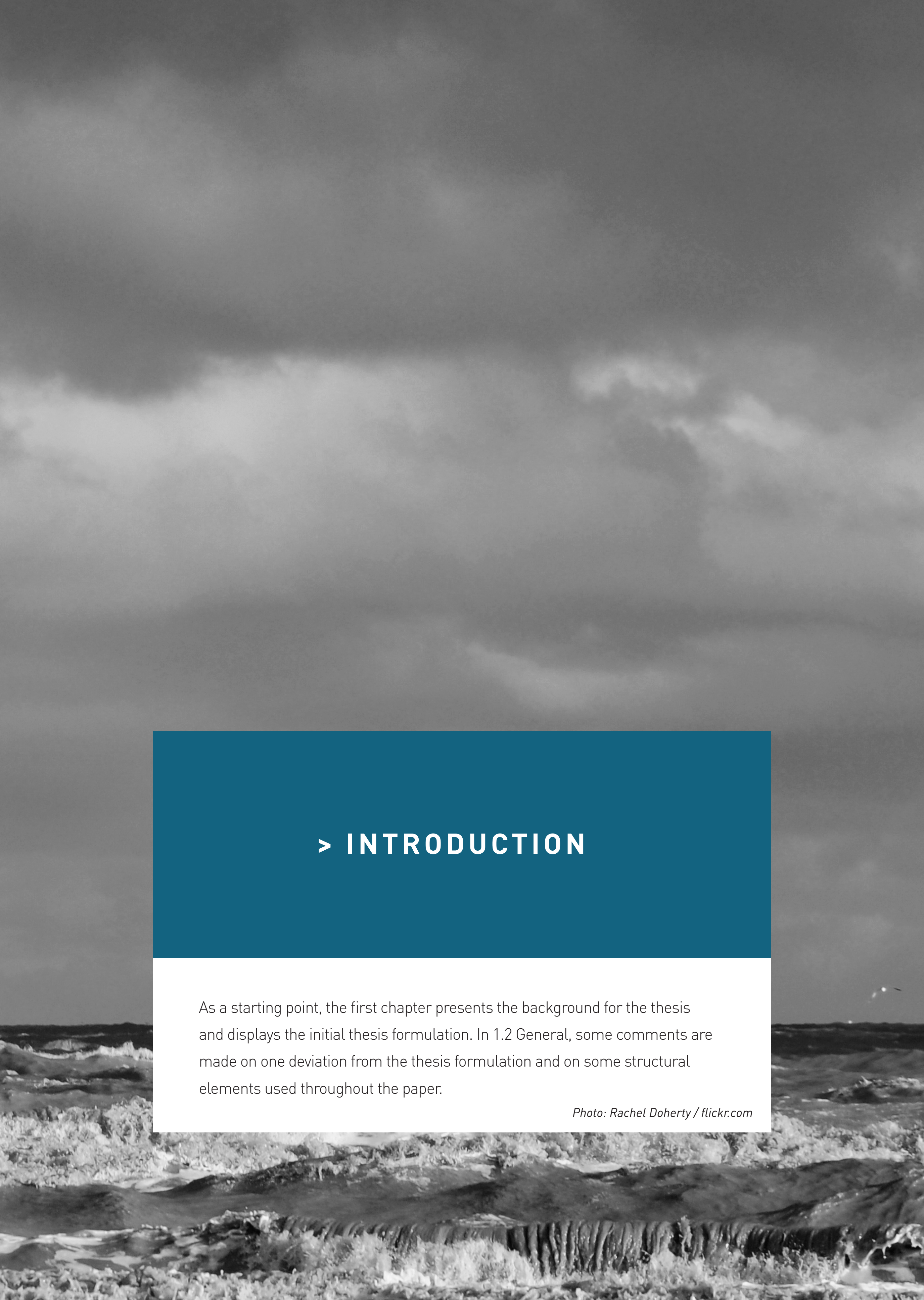
>	Introduction	145
1	Concept in Context	147
2	Concept in Detail	165
3	Navigation: Point of View	177
4	Navigation: Controllers	181
5	Navigation in Detail	183

8 EVALUATION

>	Introduction	193
1	Reflection	195

1 BRIEF

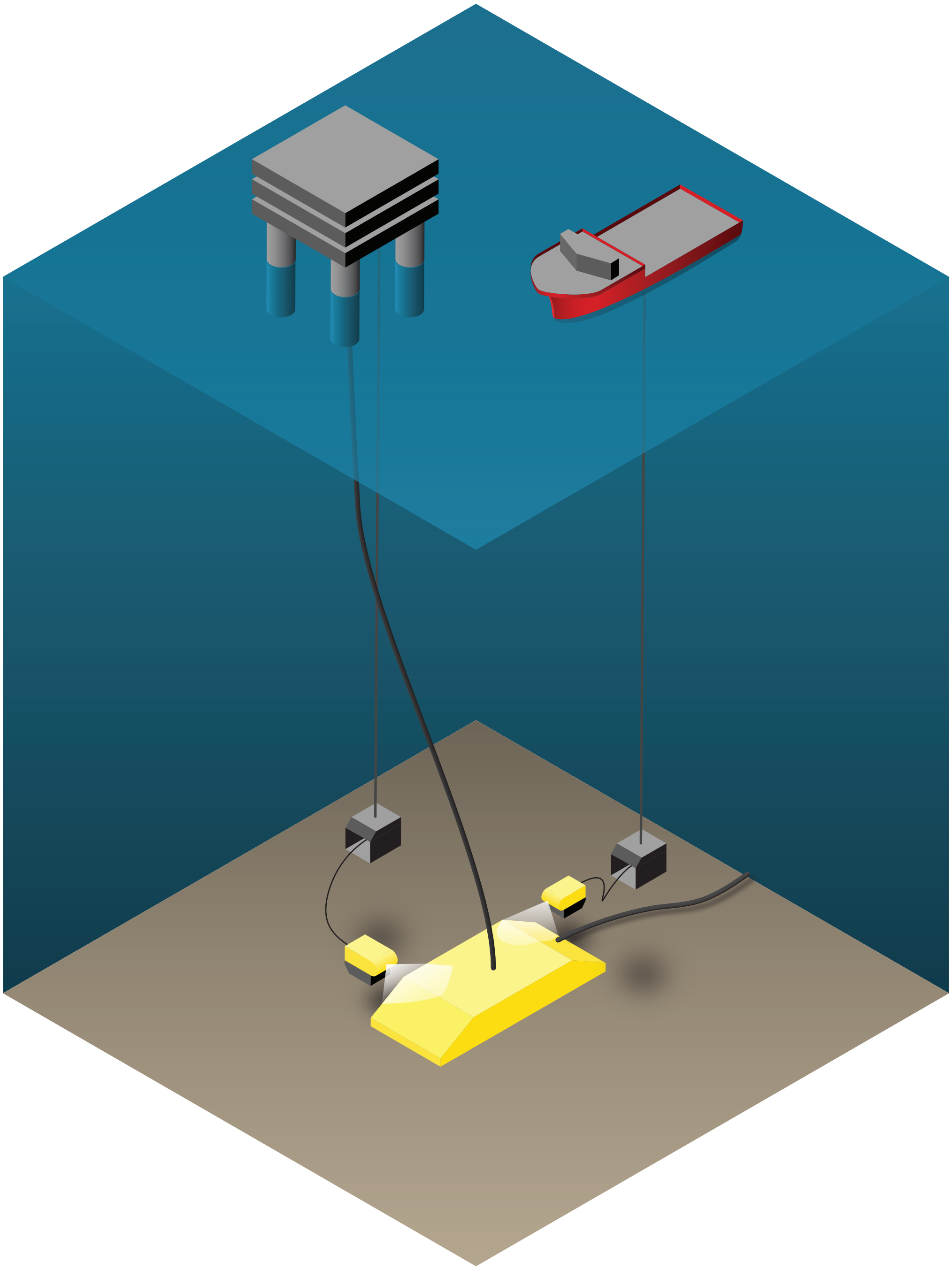
>	Introduction	13
1	Background	15
2	General	19



> INTRODUCTION

As a starting point, the first chapter presents the background for the thesis and displays the initial thesis formulation. In 1.2 General, some comments are made on one deviation from the thesis formulation and on some structural elements used throughout the paper.

Photo: Rachel Doherty / flickr.com



1 BACKGROUND

ROVs are becoming a huge part of the maritime industry today. As the industry is looking to exploit more of the subsea potential, ROVs are there to take on difficult tasks several hundred meters below sea level.

ROV
Remotely Operated underwater Vehicle.

THE PROBLEM
The sea is a treacherous place. Security measures restrict ROV missions to operate within weather windows, often leading to cancelled or delayed missions. The reason is that ROV missions take a lot of time, especially if more than one ROV is involved. The risk of several tethers tangling up or ROVs colliding, causes ROV operators to take extra care when maneuvering their ROV, sometimes going moving one at a time.

This is a consequence of perspective. ROV operators have live camera feeds directly from the ROV, normally from several angles. However, the visibility is sometimes as low as a meter, and the risk is that the operator has no time to stop the vehicle if something unexpected shows up. Other than that, operators navigate with 2D maps, depth measurements and sonar.

weather window
a period of time where external conditions such as wave height, wind speed etc., are within their maximum security levels.

IDEA
What if the ROV operators where in the water themselves, several hundred meters below sea level? What if the visibility was endless, and if something happened, they could just rewind and start over? What if they had all the time in the world to execute missions?

This is all possible with the technology we have today. By replicating the underwater world around the subsea installations in a virtual 4D environment (the fourth dimension is time), and inputting the world into an HMD (head mounted display) like the Oculus Rift, the ROV operators will feel immersed into the deep. In such a virtual world, there is endless visibility, no weather windows and the possibility to push the reset button when something goes wrong.

After executing a mission in the virtual world, the final trajectories of the different ROVs and different types of ships and vessels on the surface are saved as a 4D map. These trajectories are then followed by the different actors in real life, for a more streamlined, safe and effective mission.

THESIS SCOPE

My master thesis will focus the planning phase: is it possible to use Oculus Rift and a virtual environment to create better ROV missions? Research will revolve around the physical interface of controls, the mapping of these controls in the 3D environment and the GUI (graphical user interface) the operator will receive in the virtual world.

GUIDANCE

Professor at the Institute for Product Design at NTNU, Thomas Porathe, has guided me during the process of writing my thesis. Associate professor Trond Are Øritsland has helped with the development of the Unity prototypes and given interaction design feedback.

NTNU
Norges teknisk-naturvitenskapelige universitet

Fakultet for ingeniørvitenskap og teknologi
Institutt for produktdesign

Masteroppgave for student Stian Surén

Interaksjonsdesign i 4D sjøkart

Interaction Design in a 4D Sea map

Kongsberg Oil and Gas Technologies spesialiserer seg på å levere undervannsteknologi i oljeindustrien. De ønsker å kunne planlegge ROV-operasjoner ved hjelp av 4D-visualisering av forhold og komponenter under vann. Førere av ROV-ene vil bli eksponert for denne 4D-visualiseringen gjennom Oculus Rift og vil kunne styre ROV-ene ut ifra informasjonen de får gjennom brillene. Etter å ha planlagt en rute for ROV-ene, kan man etterpå endre ruten ved å manipulere waypoints i x, y og z-retning, og i tid.

Masteroppgaven vil fokusere på å lage en applikasjon til Oculus Rift som tar i bruk eksisterende visualiserings-software fra Kongsberg slik at ROV-førere kan styre virtuelle ROV-er i en virtuell representasjon av forholdene under vann ved en aktuell marin installasjon. Analyse av brukere og brukssituasjoner vil legge et grunnlag for hvordan GUI i applikasjonen vil se ut, samtidig som det må undersøkes hvilken input systemet trenger for å styre ROV-ene.

Oppgaven inneholder blant annet:

- Kartlegging og analyse av brukere og behov
- Utforsking av mulig input for styring av ROV
- Idégenerering og konseptutvikling av GUI for Oculus Rift
- Prototyping og brukertesting
- Utvikling av designkonsept for Oculus Rift

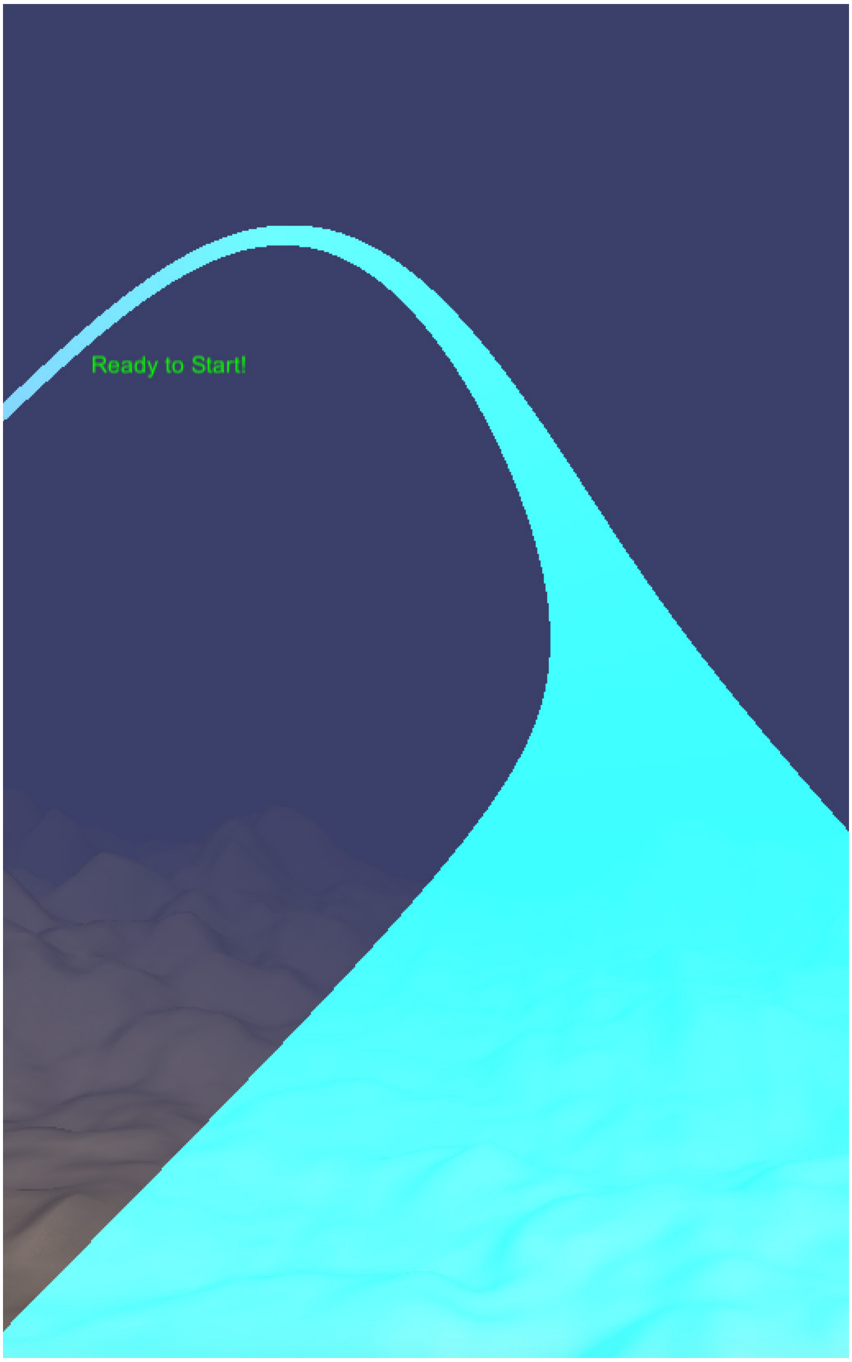
Oppgaven utføres etter "Retningslinjer for masteroppgaver i Industriell design".

Ansvarlig faglærer:	Thomas Porathe
Bedriftskontakt:	Thomas Hammer
Utleveringsdato:	27.februar 2015
Innleveringsfrist:	24.juli 2015

Trondheim, NTNU, 27.februar 2015

Thomas Porathe
Ansvarlig faglærer

Casper Boks
Instituttleder


Ready to Start!

2 GENERAL

This small section is used to give general comments on the basis for the thesis and the structure of the paper.

KONGSBERG

As seen in the thesis formulation in the previous section, Kongsberg Oil and Gas Technologies were originally planned to play a role in this thesis. This was only materialized through two meetings at the beginning of the period. The thesis is therefore conducted as a solo project.

STRUCTURE

Each section has its own photo count. Photos and figures are counted seperately.

Extra columns on the sides are used to give extra information or definitions. A word with an underline tells you to look to the outer margin of that page.

All pictures, illustrations and figures are taken or created by me, unless otherwise noted.

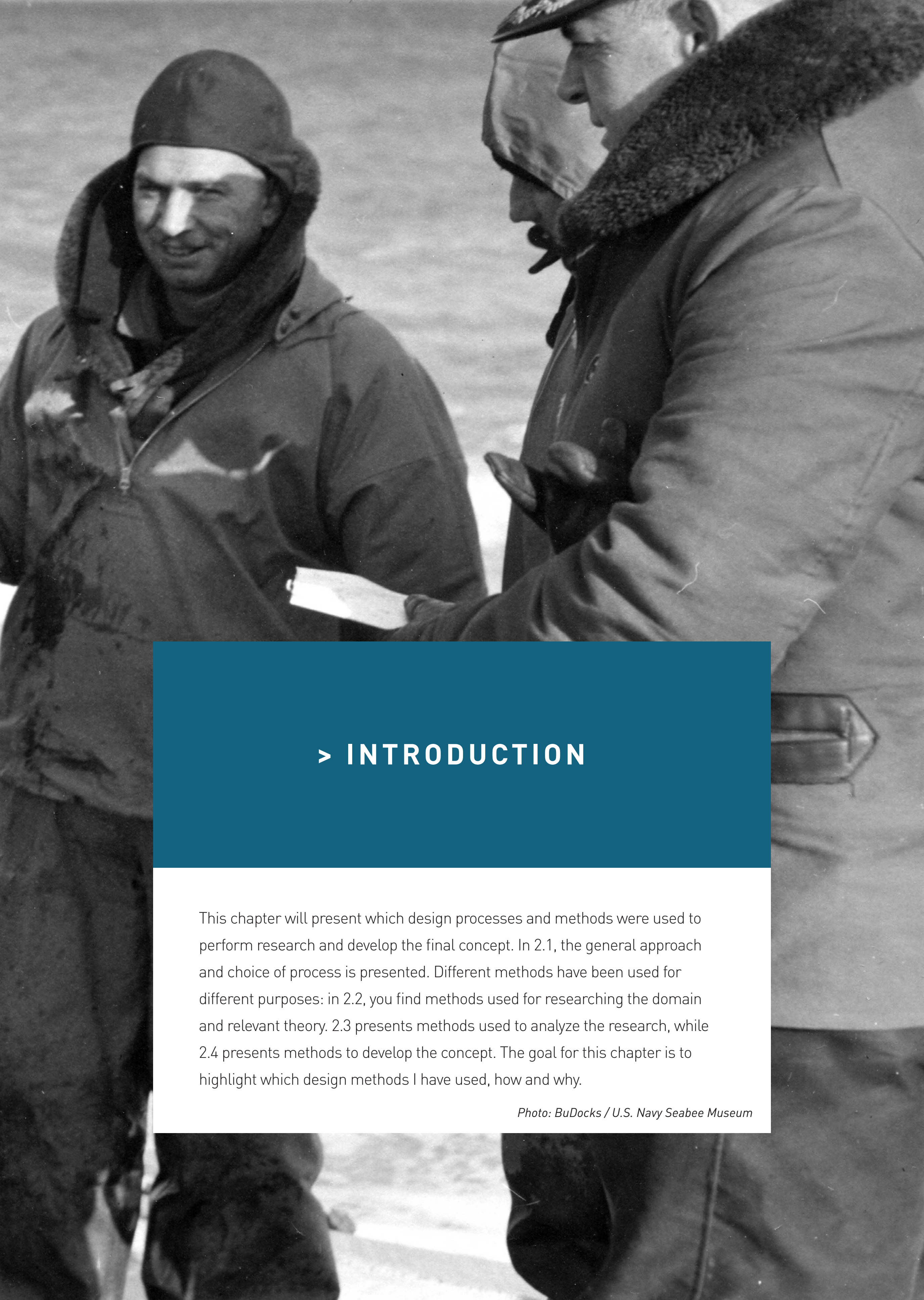
underline

Used to link a word in the text to extra information in the outer margin.

2 METHODOLOGY

>	Introduction	23
1	Approach	25
2	Research	29
3	Analysis	33
4	Concept Development	35

Photo: BuDocks / U.S. Navy Seabee Museum



> INTRODUCTION

This chapter will present which design processes and methods were used to perform research and develop the final concept. In 2.1, the general approach and choice of process is presented. Different methods have been used for different purposes: in 2.2, you find methods used for researching the domain and relevant theory. 2.3 presents methods used to analyze the research, while 2.4 presents methods to develop the concept. The goal for this chapter is to highlight which design methods I have used, how and why.

Photo: BuDocks / U.S. Navy Seabee Museum

1 APPROACH

USER CENTERED DESIGN

Abras et al. (2004) say this about user-centered design: "... (it) is a broad term to describe design processes in which the end-users influence how a design takes shape." At the beginning of this process, it was obvious to me that the user had to be in the center of attention. It is a process I have experience with from previous projects and employments. It helped me choose methods that would emphasize the user's need and remove assumptions.

Service design is a term that is in the wind, though I did not set out with the intent to use any specific service design methods. A service is a series of interactions (Stickdorn and Schneider, 2012), and it was only after a while that I realized I had to use a service design mind set when looking at how an ROV operation was planned and executed: from a client who wants a job done, to the ROV operator who makes it happen.

This thesis is focused on the human interaction with machines, and how to take advantage of the possibilities a digital tool can have on the planning and execution of an ROV operation in the subsea industry.

PROCESS

A user-centered design process starts with gathering domain knowledge and user knowledge, analysis, an iterative concept development period, then detailing and then end up with a final prototype. You ideate and then decide, and you have relatively distinct stages in the process.

When I started, I was really excited about the thesis: I wanted to try all the gadgets and see if the ideas I had were possible. Simultaneously, I started gathering domain specific information, user knowledge and relevant theory. It all happened sort of mixed up together: one day I would work on creating prototypes, another on learning about ROVs. This resulted in a fuzzy front-end process (Koen et al., 2002): discoveries influenced the prototypes, and the prototypes influenced the questions I had. As I made discoveries, I could pinpoint the direction I wanted to go in. On the next page is a visualization of how my fuzzy front-end process was conducted.

Two focal points were constant throughout the process: the wish to learn about the user's needs, behavior and habits, as well as a wish to test and apply cognitive psychology theories to my prototypes. This is apparent in the theoretical section of this thesis, where most sections are based on theory from academic and applied psychology, rather than from design, although they are undoubtedly linked.

METHODS

I have intentionally tried to explore the seemingly infinite world of design methods. Books like 200 Ways to Apply Design Thinking (Curedale, 2012) and the follow-up 200 More Ways to Apply Design Thinking (Curedale, 2013) live to prove the vast collection of methods out there. I have chosen methods that fit with the user-centered design approach. The methods are presented in the following sections: 2.2 presents methods used to learn about the users, 2.3 are methods to analyze and reflect upon the gained knowledge, while different methods to conduct user testing and concept development are presented in 2.4.

RESOURCES

For domain and user knowledge, I have consulted industry professionals in Oslo and Stavanger, as well as the Department of Marine Technology at NTNU, Trondheim. For theoretic input on interaction design and psychology, I have benefited from Bekk Consulting AS, and classmates and professors at the Institute of Product Design, NTNU. For inspiration and motivation, I made a visit to the CERN Media Lab to take a behind-the-scenes look at their interactive, multimedia installations.

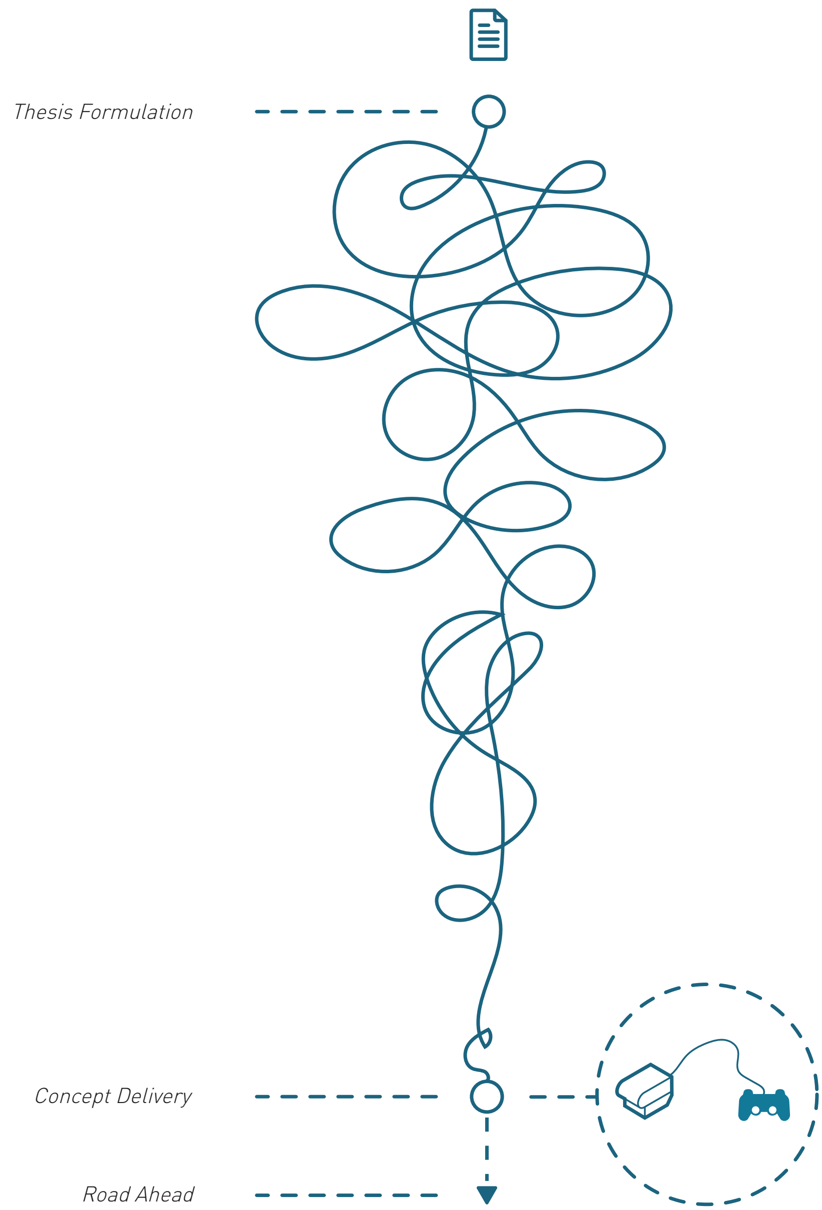


Figure 1: Visualization of the fuzzy front-end design method. Prototyping has helped raise questions, while answers have influenced prototyping.

Photo 1: Surface Control Unit for the Minerva NTNU ROV.

2 RESEARCH

The domain of ROV operations was completely new to me, and it was therefore important that I got an understanding, not only of the subject itself specifically, but also how the industry works and how ROV operations fit in with the whole.

This section presents which methods were used to gain knowledge about the domain: from how an ROV is built, to how an operation is planned and executed, and, importantly, who the user is. The methods are described, followed by how they were used and why.

LITERATURE REVIEW

The purpose of a literature review is to review and understand the body of knowledge that exists on a subject (Curedale, 2012). A quick google search gave me thorough information about ROVs, while published books and articles were helpful for understanding relevant theory. This material was a key in getting an early understanding of the domain, and helped when I later communicated with experts in the field.

PRODUCT SAFARI

This is my own revised version of the service safari design method. A service safari is used to explore what gives good and bad user experiences by services in use (Stickdorn and Schneider, 2012), and involves putting yourself in the shoes of the user through an entire service.

Obviously, I found that I could not plan and execute a real ROV mission myself. What I could do was to drive an actual ROV at the bottom of Trondheimsfjorden. Additionally, I tried the same ROV simulator Oceaneering in Stavanger uses to train ROV pilots. By using these products, I could identify with the problems and challenges associated with controlling an ROV. I could also compare the differences between the NTNU Minerva ROV and the Oceaneering Simulator, to see which elements work in which settings.

DEPTH INTERVIEWS

Interviews are a fundamental research method to gather information about the user, their opinions, attitudes, perceptions and experiences (Martin et al., 2012).

The interviews conducted in this study were semi-structured interviews with different experts within the ROV industry; an industry ROV pilot and instructor, an industry head of ROV education, an industry ROV supervisor and a professor in Marine Sciences at NTNU. Because the subjects have a varied professional, educational and personal history, a standardized questionnaire was discarded (Louise Barriball and While, 1994).

Semi-structured in this context means that I went into the interview with an agenda of subjects and questions I was interested in, but drifted from the agenda whenever something unexpected and interesting came up, which was often.

The results from the interviews helped improve my understanding of the domain and the users, and gave important input to the development of personas in chapter 5 The Users. It gave an important basis for further analysis. A template agenda for my semi-structured with the industry ROV pilot can be seen in the Appendix.

OBSERVATON & FIELD STUDY

Observation involves observing people in their natural activities and usual context (Curedale, 2012). To observe the user in action is an easy way to understand how the users work and interact in context, and may reveal activity that are not mentioned in interviews. A field study is similar, but also involves an obtrusive part where the designer asks questions and asks for certain behavior (Curedale, 2012).

During the process, I observed an actual ROV operation with the NTNU Minerva ROV and an industry ROV pilot/instructor use a standard ROV simulator. I have also watched videos of real ROV missions in the oil and gas industry, in the field of archaeology and in the field of wildlife studies. While with the user in context, I purposely switched between only observing and periods where I would ask questions, in order to benefit from both field study and observation advantages. This way, I could see how the users behaved in context, and they could give me answers while in context themselves.

The results from this study made it easier for me to understand the elements in an ROV operation, and to first-hand observe user behavior and hiccups in the current system.

Photo 2: Pilot of the NTNU Minerva ROV.

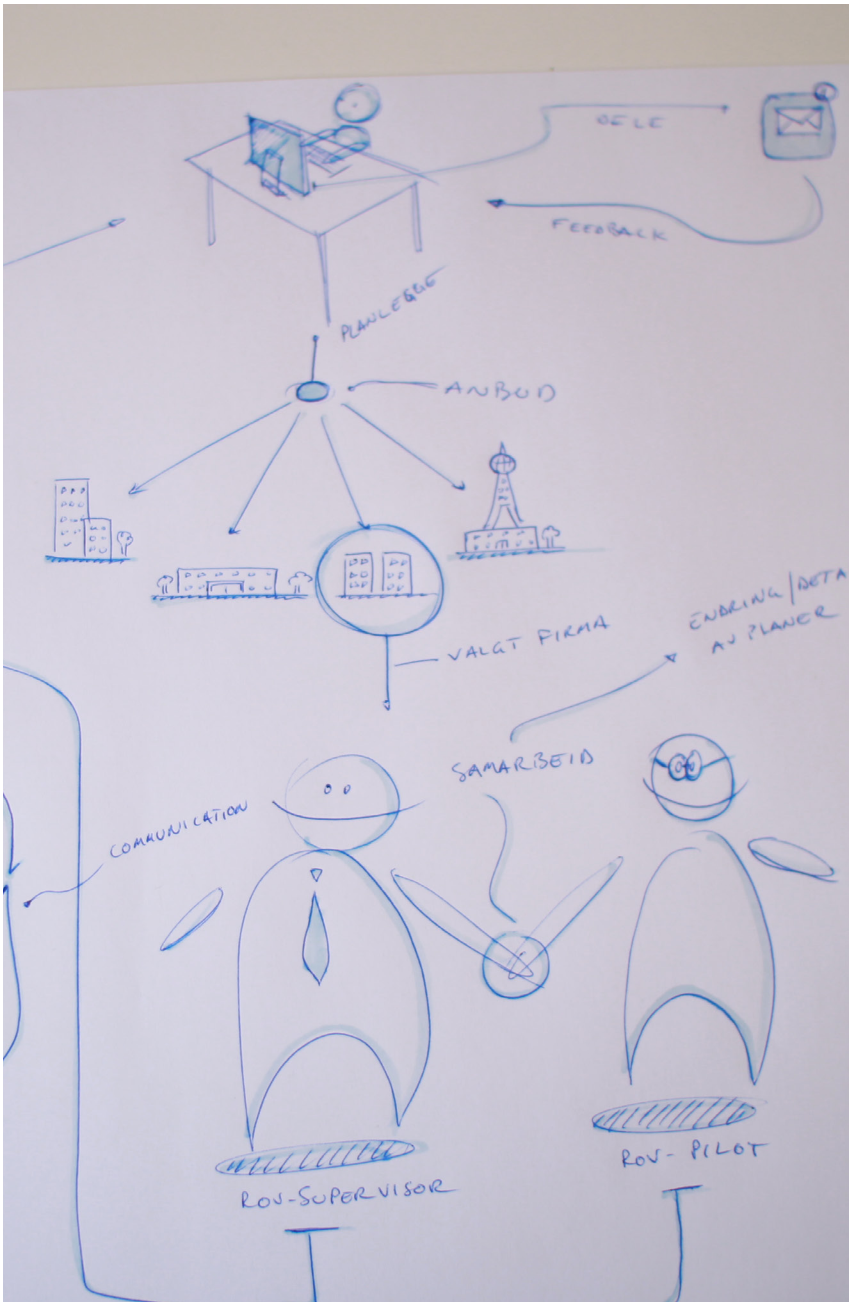


Photo 1: Scenario sketching was a good help to see the whole picture.

3 ANALYSIS

The presented methods were used to analyze, group and help make the research tangible.

PERSONAS

Personas are a description of fictional persons who represent the shared characteristics of the most important users of a solution (Cooper et al., 2007). They give an understanding of who the users are and make sure that the developed solution cover the needs of the user (Cooper et al., 2007).

The personas are developed as a result of the depth interviews and the observation and field study methods presented in the previous section. The personas are given a sufficient amount of personality to be realistic, and are divided into a primary user and a secondary user, and if in conflict, the needs of the primary user will be prioritized.

During the further analysis, the personas have been used to create realistic and useful scenarios. They have also been used in the concept development phase to help tailor the design for the specific user. It has also been easier to use the name of the relevant persona, instead of using, in this case, ROV supervisor or ROV pilot.

DESIGN SCENARIOS

Design scenarios are hypothetical stories, created with sufficient detail to meaningfully explore a system or a service (Stickdorn and Schneider, 2012). They shift the focus from functional specifications, to describing how users will use the system to accomplish tasks (Rosson and Carroll, 2009).

I used storyboards in a comic-strip style to visualize how the needs of the users would transform throughout the process of planning and executing an ROV operation. The data from the research methods were used to construct plausible situations, and the personas were used to give a face to the characters involved in the story.

The scenarios helped pinpoint where the problems lie in the existing solutions, and gave important input to the further concept development.

Photo 1: The paper box was a "magic" prototype to kickstart the project on a highly conceptual level.

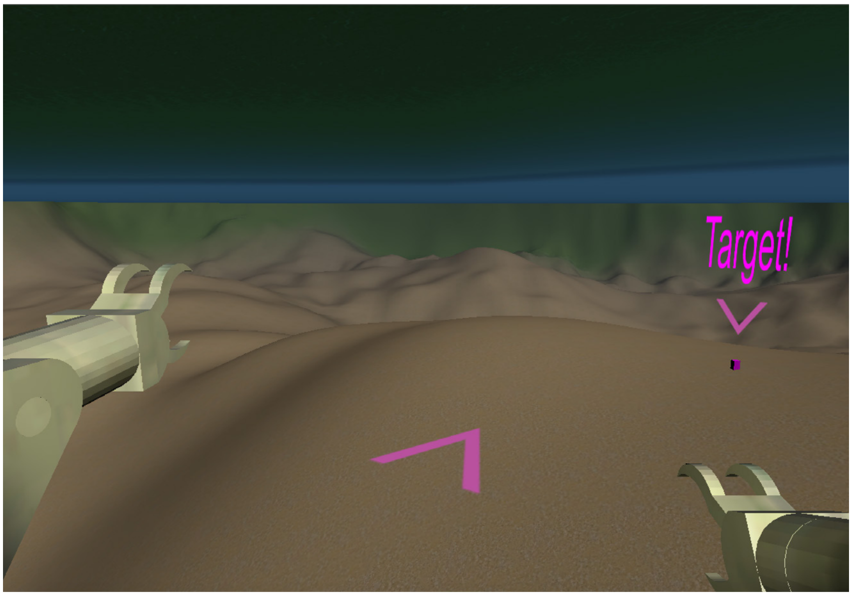


Photo 2: A realistic, functional prototype made it easier to get good feedback from users.

4 CONCEPT DEVELOPMENT

As mentioned, I had a fuzzy front-end approach to the project. This means that, despite the order of the sections in this chapter, the methods were not used in a linear fashion. Testing of my hypotheses began early, while the scenario testing was conducted later, when the users and scenarios were established.

Presented in this section are the overall methods used to develop the concept. In chapter 6 Concept Development, I will go deeper into how I have conducted some aspects of my user testing, as well as the underlying theory.

PROTOTYPING

Prototypes come in many shapes and sizes, and are used to make ideas and concepts tangible and testable. Prototypes were used to ideate, to test very specific parts of the system, or to test the system as a whole. Prototypes were also used in the scenario testing.

Different levels of prototyping were tested:

> *"Magic"*

This was the first type of prototyping I did. Magic prototypes rely on the imagination of the test user. They were given small, "magic" objects they could interact with and that would spark ideas. I used magic prototypes to get ideas and inspiration for the project, not to test ideas I already had. It became a sparring session with the user, and proved useful as a kickstart to the project.

> *Paper*

Paper prototypes are fast and simple prototypes often used to test certain types of a solution. Traditional, low fidelity paper prototypes were used to test the GUI of the concept I will present, both for the navigating the overall system and for the 3D world within it. The results were qualitative and were used to filter which concepts I would integrate into the functional prototypes.

> *Functional*

Functional prototypes were made to create a high-fidelity, realistic experience for the user, so that the feedback could be more on point. I mostly used the game engine Unity, in addition to the development environment Processing, to create my prototypes, and was able to not only get qualitative responses, but also quantitative results for further analysis. Different prototypes were created, to explore different aspects and directions for the concept. These prototypes lay the foundation for my work, and you can discover more in chapter 6 Concept Development.

USER TESTING

User testing was conducted at different levels, from very informal testing of the magic prototypes, to formal testing of certain aspects of the functional prototypes. This was dependent on the types of responses I wanted. Informal was used where I wanted ideas and an open conversation, formal was use where I wanted quantitative results and less open feedback.

informal
"based on rules of thumb and the general skill and experience of the evaluators".

formal
"using exact models and formulas to calculate usability measure".

(Nielsen, 1994)

Indsustry professionals in the ROV industry proved difficult to come by as Trondheim have no subsea industry. Test subjects were mostly students in their twenties from different disciplines in Trondheim, both male and female.

> Cognitive psychology

I was interested at the beginning to delve into the world of cognitive psychology, and wanted to use methods from this realm of academics as an addition to my regular user tests. Specifically, I wanted a way to test a user's cognitive load while using my system, or certain parts of my system. I practiced a method called the tapping test, which is designed to measure the real-time cognitive load of the user (Tracy and Albers, 2006).

> Scenario Testing

The primary reason I developed personas and scenarios as explained in the previous sections, was to utilize the insight by testing the scenarios.

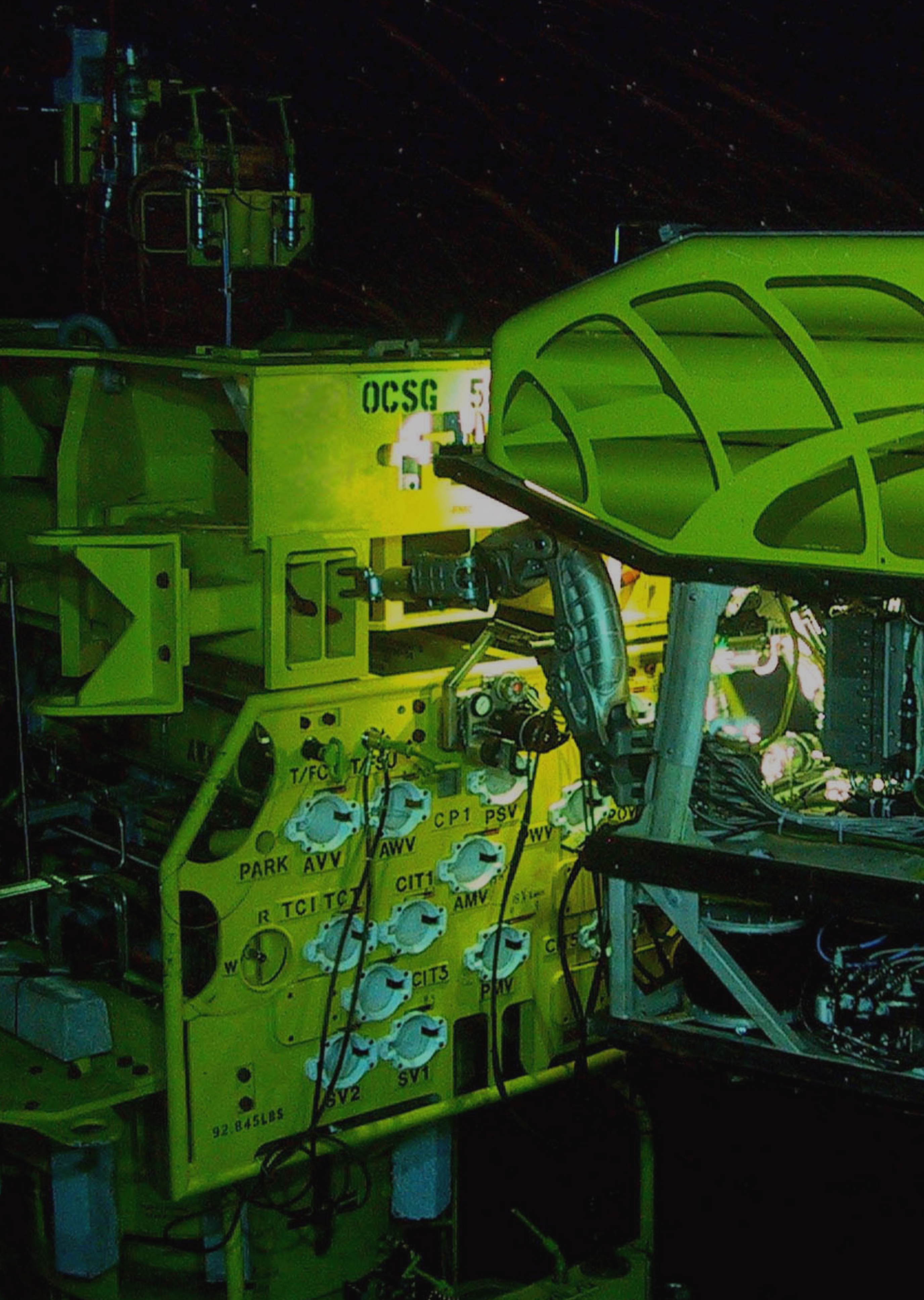
The testers were put into context, and tried to solve the problem or fulfill the need they were given. If stuck, the users were urged to solve the problems on their own, without any involvement on my part. This helped me uncover flaws in the design, and improve both the overall flow of the user experience and the small details and functionalities.

The user testing is explained more thoroughly in 6 Concept Development.

Photo 3: The Oculus Rift was tested along with the PS3 controller.

3 PRIMARY RESEARCH

>	Introduction	41
1	The ROV	43
2	ROV Missions	47
3	Controlling an ROV	49
4	Navigation & GUI	55

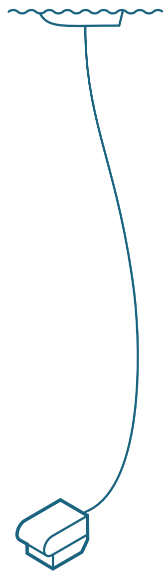

OCSG
T/FC
CP1
PSV
PARK
AVV
AWV
CIT1
AMV
R
TCI
W
CIT3
PMV
SV2
SV1
92.845LBS

> INTRODUCTION

Domain knowledge was very important to understand what I was designing for. The primary research chapter will unveil discoveries made about the ROV industry. First, the ROV is taken apart and analyzed. Then, in 3.2, the way ROV missions are conducted is brought forward. 3.3 focused on how the ROV is controlled, both in terms of the physical interface to the user, but also how the ROV itself behaves. The ROV pilot's way of navigating and the GUI in their control unit is displayed in 3.4, while challenges are discussed in 3.5.

Photo: Frank van Mierlo

Photo: oceaneering.com



1 THE ROV

WHAT IS AN ROV?
ROV, in the context if this thesis, is an acronym for remotely operated underwater vehicle. It is commonly a tethered underwater robot controlled from a boat or from land. (Wikipedia, 2015)

ROVs are used for a wide variety of operations, most predominantly in the subsea oil and gas technology field. However, they are also used by other industries in the maritime sector, by researchers and educators, military and recently an open-source ROV (OpenROV, 2015) was released, opening up the underwater world to more people worldwide.

CONSTRUCTION
The ROV is usually made with a floating pack on top, followed by an aluminum frame beneath. The floating pack is made of syntactic foam, which traditionally is an epoxy filled with hollow micro glass spheres. Ceramics have replaced epoxy for ROVs travelling to extreme depths. The floating counteracts the negative buoyancy of the remainder of the ROV, making the whole thing slightly positively buoyant. The result is that in case of failure, the ROV will very slowly ascent to the surface.

Within the frame, thrusters are mounted in several directions. The tether is normally mounted on top or at the back. Additionally, cameras and lights are mounted on the frame or on top of the floating pack. One camera is usually mounted to monitor the tether.

Manipulators, in other words arms or tools, are also an important part of the ROV. Depending on the size, an ROV can have up to several manipulators, designed to perform specific tasks. There are other sensors in addition to the cameras, such as sonars and pressure sensors.

AREAS OF OPERATION
As mentioned, the ROV is used for many things. However, in the context of this thesis, I will present the three operations relevant to subsea installations.

> *Work*
The ROV has up to several manipulators and is used to perform 'hands-on' tasks.

> *Monitor*
These are typically small ROVs used to monitor operations and to perform visual inspections of structures.

> *Survey*
The ROV is fitted with multiple sonars to get topological information of the sea bed and structures on the bed itself.

tether
is a flexible cord containing electrical wires and fiber optics, linking the ROV to the surface

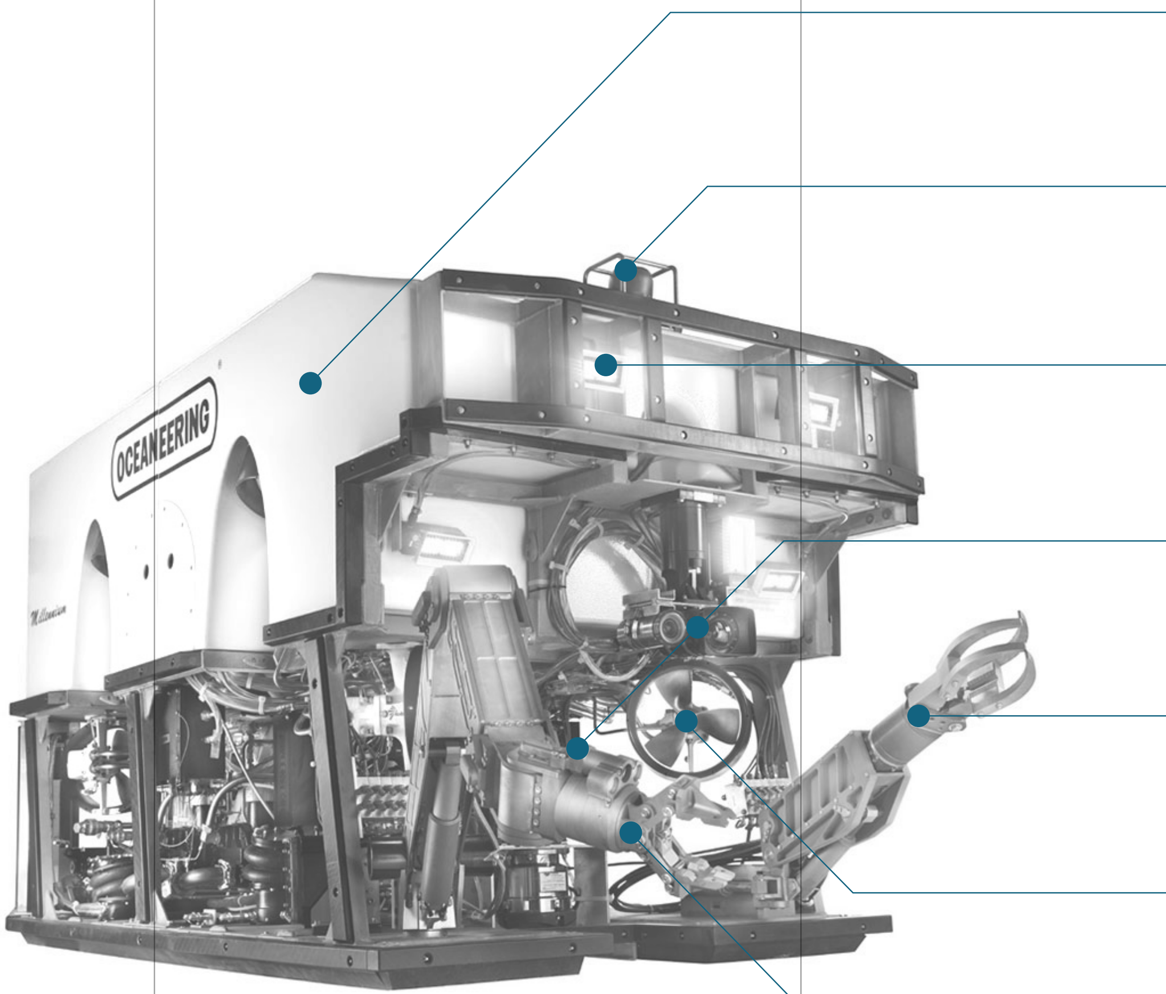


Photo: oceaneering.com

THE ROV DECONSTRUCTED

The most common and important ROV elements. This particular ROV is a work-class ROV from Oceaneering designed to perform heavy operations on depths up to several thousand metres (Oceaneering, 2015).

FLOATING PACK
Counters the heavy gear. Gives the ROV a slight positive buoyancy, so it floats up in case of failure.

TRANSPONDER
Used to determine the position of the ROV using acoustic signals. A common type is the Doppler Velocity Log.

LIGHTS
Work lights are essential in the dark. These are 250 watt high intensity LEDs.

HD-CAMERAS
Cameras are mounted on the front, on the manipulators, and on top of the ROV, looking on the tether.

SIMPLE MANIPULATOR
Controlled by the ROV pilot, the claw can rotate in all three rotational axes, and grab or cut objects.

THRUSTER
There are eight thrusters on the big ROVs. Four for vertical, and four to go forward, sideways or rotate, by pairing up differently.

ADVANCED MANIPULATOR
Controlled by the co-pilot with a slave arm. The pilot moves the arm, and the manipulator follows.

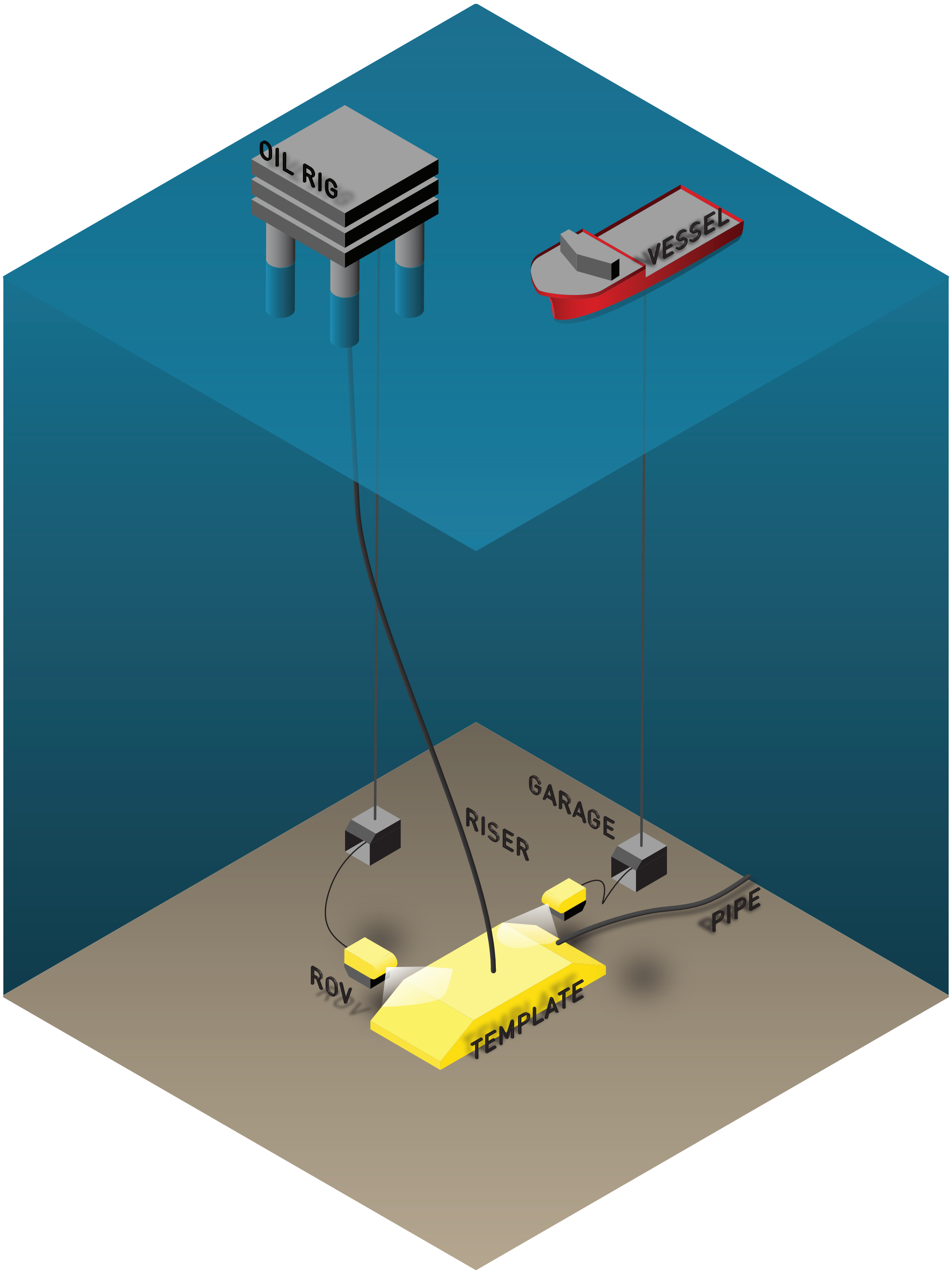


Figure 1: Common components in an ROV mission. A template is the access point to an oil well.

2 ROV MISSIONS

In the subsea oil and gas industry, an estimated 65% of all ROV missions are launched from oil rigs, while the remaining 35% are based on vessels. Vessel based operations are mostly for service and mounting operations on the sea bed.

Rigs have one work class ROV and one observation ROV, while vessels can have up to two work class ROVs and one observation ROV. There are three persons per work class ROV: one pilot, one co-pilot and one supervisor. The pilot controls the ROV and one simple arm, the co-pilot controls an advanced, slave arm and will help the pilot on request. The supervisor will, as implied by the name, keep control of the situation, help the pilots if needed, and keep in touch with the client. For the observation ROV, only one person is needed.

A typical mission will conducted like this: a rig operating company responsible for the oil rig will order a task from an ROV company to perform a task on equipment under water. In most cases, the ROV missions are standard missions in the eyes of the ROV operator, and they spend little time planning. They get a task and they do it.

The rig operator, also known as the client, will follow the operation either in the control room, another office on the rig/vessel or from an office on land. Upon request, the client will ask for specific video feeds from the ROV, and it is the job of the co-pilot or the supervisor to provide it to them. The pilots will rarely make decisions during a task, and will wait for an order from the client if something unexpected comes up.

On some occasions, the ROV companies are asked by the rig operator to make an animation of how an operation can be executed. This is used by the rig operator to plan the operation, and when they tender the job to the open market. Based on quotas they receive by ROV companies, they decide who gets the job.

For less common, unique operations, e.g. where there is unforeseen damage and special tooling is needed, a more thorough planning process is needed. The ROV company can then be included in the planning at an early stage, to aid with their expertise.

Before a mission, no matter what type, the pilot has a responsibility to look at maps and structures to get familiar with the surroundings on the sea bed. It can happen that the ROV pilot gets lost in the darkness a thousand metres below. Maps are often only used prior to operations, and not during them.

“You go in and out. It's like driving a car.
ROV pilot

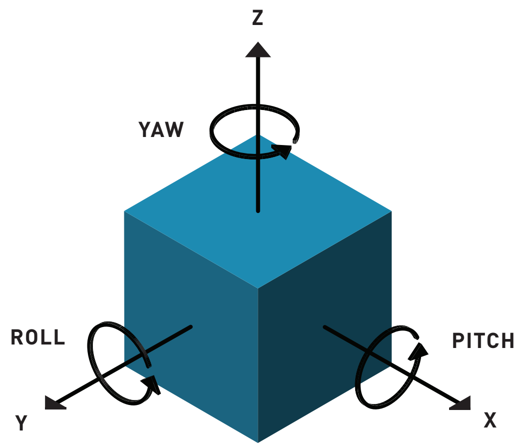


Figure 1: Six degrees of freedom

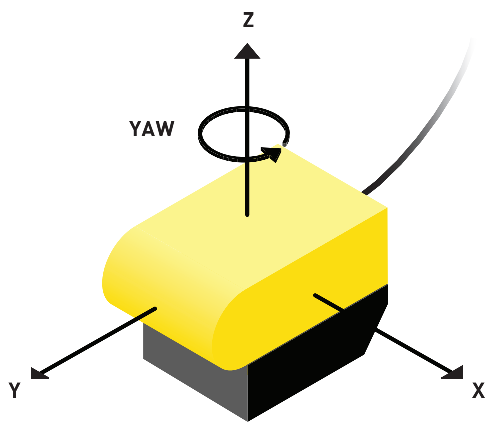


Figure 2: The ROV translates in all three axis, but only revolves around the z-axis

3 CONTROLLING AN ROV

THE ROV IN 3D SPACE

ROVs, as all rigid bodies moving in 3D space, have six degrees of freedom (DOF), three translational and three rotational. The translational components define the trajectory, and the rotational components define the behavior of the ROV along the trajectory (Hale, 1997). This makes for a pretty complicated system to control (García-Valdovinos et al., 2014) (figure 1).

However, ROVs are designed to only make use of 4 DOF (x, y, z, yaw), while the other two, roll and pitch, are considered intrinsically stable (García-Valdovinos et al., 2014). This implies that the ROV is always horizontally aligned. Thus, the ROV operator has a simpler job of keeping track of the orientation of the vehicle.

Given that an ROV is operating underwater, the vehicle will experience much more friction than vehicles operating on land or in the air. This means the thrusters must be powered continuously to move in any direction. Also, as ROVs are almost neutrally buoyant, gravity will not affect the ROV.

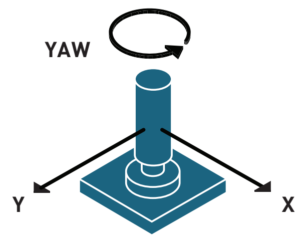


Figure 3: Single joystick setup. Y-axis (forward) is forward, and the x-axis (slide sideways) is to the sides. Yaw maps to the rotation of the joystick.The z-axis (up and down) is controlled by two buttons on the joystick or an external lever on controlled by the other hand.

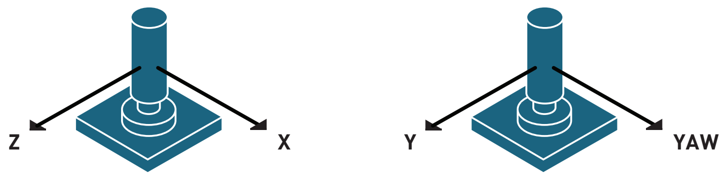


Figure 4: Double joystick setup. The joystick on the right maps the y-axis (forward) to forward on the joystick, and yaw (rotation) to the sides. The left joystick on the left maps the z-axis (up and down) to forward on the joystick, and the x-axis (slide sideways) to the sides.

HOW REAL ROVS ARE CONTROLLED TODAY

To control an ROV today, a pilot sits on the surface and communicates with the vehicle using a Surface Control Unit (SCU). Some specific tasks are autonomous, such as position tracking, dynamic positioning, auto-heading and auto-depth control (García-Valdovinos et al., 2014). The Surface Control Unit consists usually of several screens showing a live camera feed from cameras mounted on the ROV itself. The screens are usually not arranged in any particular order according to the orientation on the ROV itself. All maps are also 2D and from a birds eye view, like any other normal map.

Modern ROVs are usually controlled by a single joystick. A rotating joystick offers a good one-to-one relationship between how the ROV moves and how the joystick is used (figure 3).

Some ROVs have a double joystick setup. The joystick on the right normally controls forward motion and yaw, while the left one controls up and down and sideways sliding (figure 4).

The simple manipulator is controlled by a simplified joystick (photo 1 on the next page) or by a set of buttons. It is controlled by the main pilot.

Advanced manipulators are controlled by the co-pilot with a master arm in the SCU. A slave arm on the ROV itself will then mimic the motion of the master arm. Force feedback is given to the pilot so that he or she is sensitive to the force applied to objects (photo 2 on the next page).

dynamic positioning
A vessel will automatically keep a position by working against winds and currents.

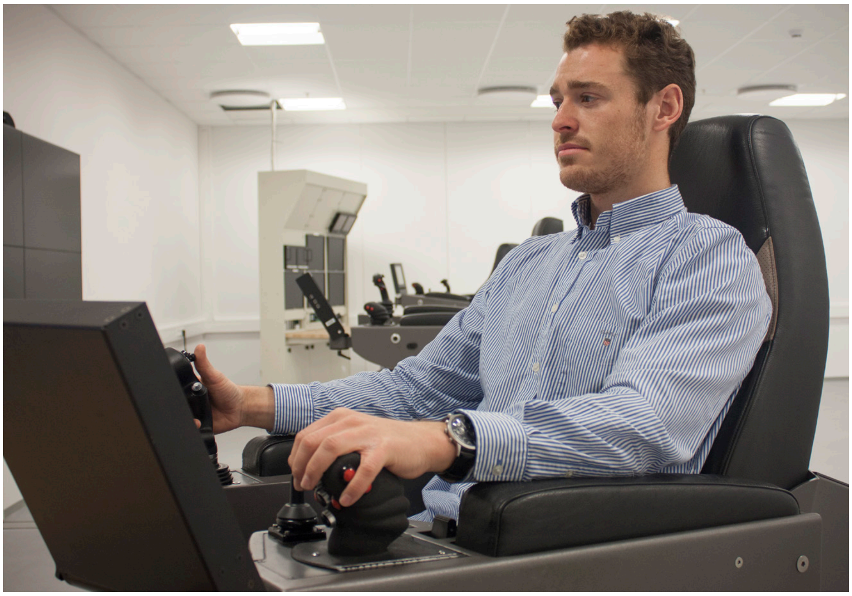


Photo 1: ROV Simulator used to train pilots at Oceaneering in Stavanger, Norway. The right joystick controls the ROV, while the left controls the simple manipulator.

Photo 2: Screen layout for the Oceaneering ROV Simulator. Six screens display information to the pilot.

Photo 3: The most modern simulator gear at Oceaneering. To the right is the master arm that controls the advanced manipulator on the ROV.

Photo 4: Portable controls for the Minerva ROV used by NTNU, using a double joystick setup.

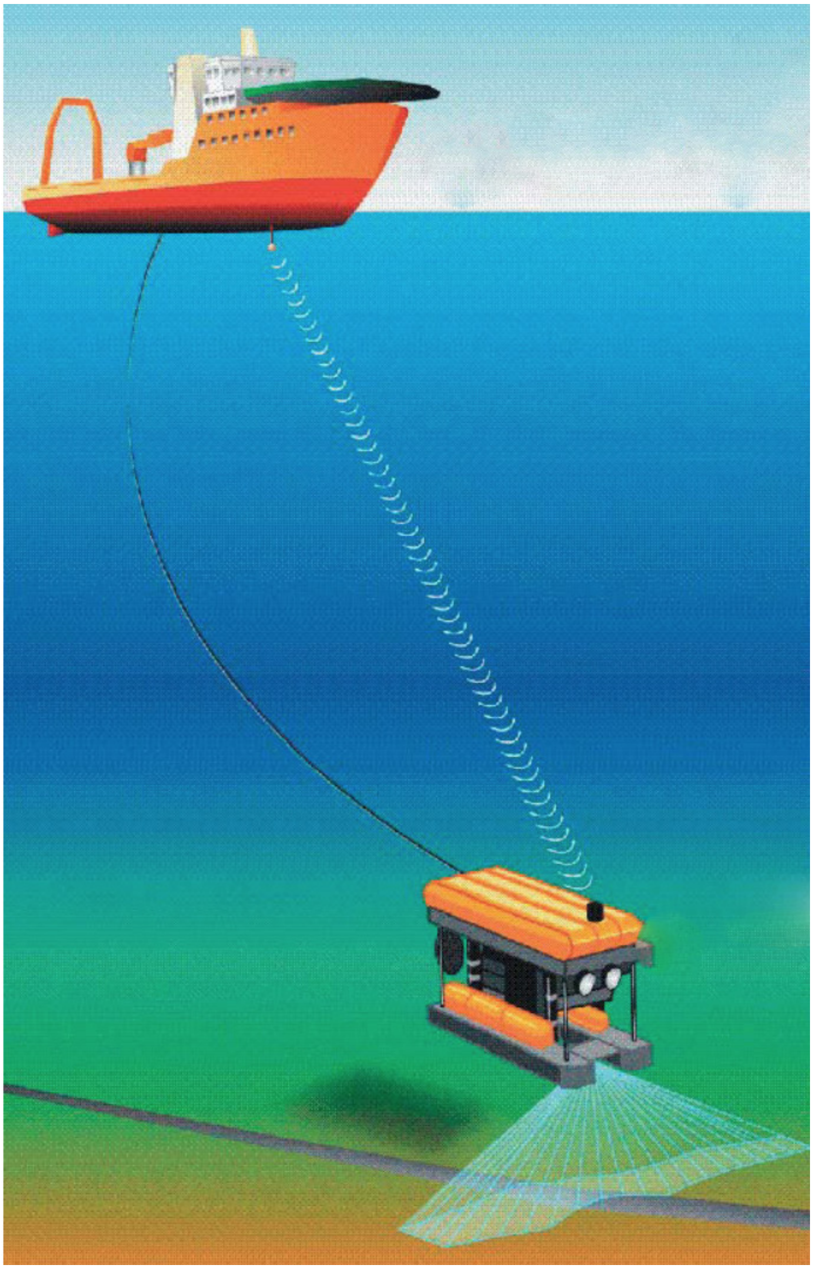


Figure 1: An ROV tracked by the HAIN system.

Photo: Kongsberg Maritime / km.kongsberg.com

4 NAVIGATION & GUI

The ability to navigate the ROV flawlessly in the darkness of the deep is undoubtedly important. To successfully do this, ROV pilots are aided both by technology and good, old maps. Acoustic positioning, pressure sensors, sonars: they are all used to help the pilot know where he is. There are several different technologies to track an ROV, such as the Kongsberg HAIN system, long baseline grid and the ultra short baseline grid (Kongsberg Maritime, 2015). To describe these technologies in detail is outside the scope of this thesis, but it is safe to say that positional information is available.

The type of information the pilot receives on his screens are dependent on the type of mission and the pilot's own personal preferences. However, presented on the next page is a range of "outputs" that is common to display.

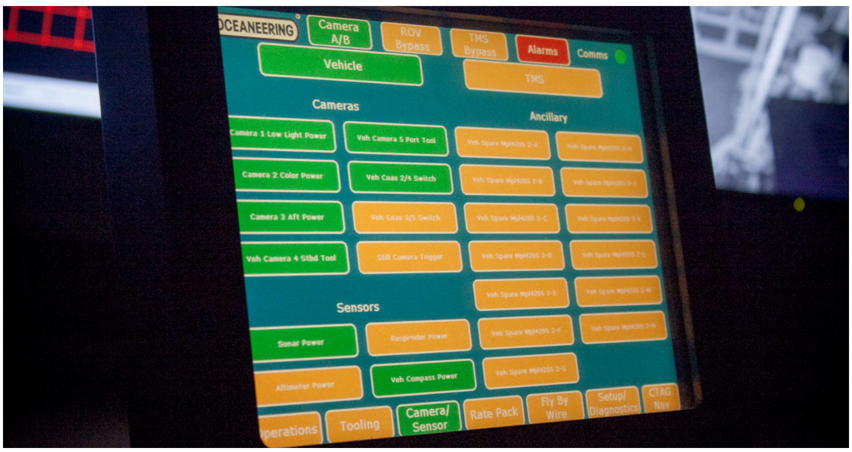


Photo 1: A touch screen mounted on the pilot's chair. The pilot controls the cameras, can toggle auto functions and turn on or off sensors among other things.

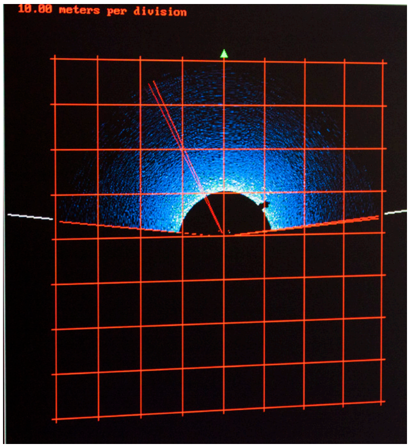


Photo 2

Sonar from the Oceaneering ROV showing only a sandy bottom ahead. Each grid cell is 10 by 10 metres in reality.

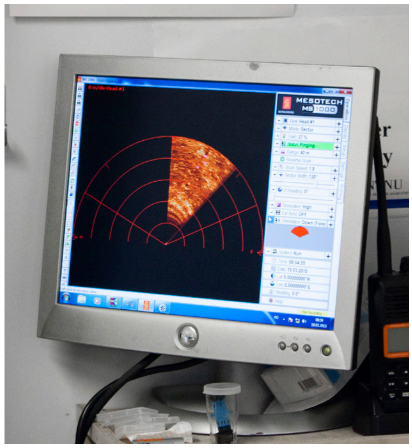


Photo 3 ▸

Sonar on the NTNU Minerva ROV. Has a smaller measuring angle than the Oceaneering sonar.

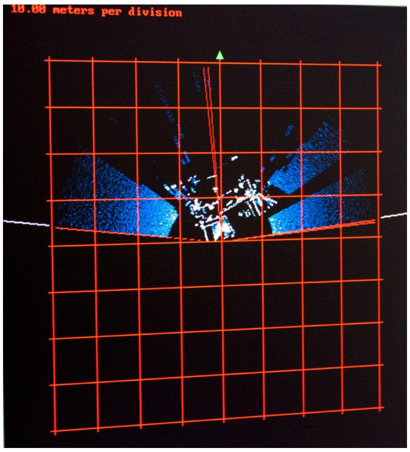


Photo 4

Sonar from the Oceaneering ROV showing a template ahead.

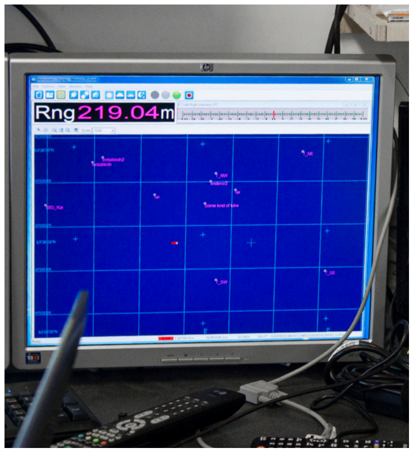


Photo 5 ▸

The live map from the SCU of the Minerva NTNU ROV. Each installation is only a dot in a grid. The number on the top is the distance from the launch site of the ROV and the ROV itself.

SONAR

A sonar sends out acoustic waves in front of the ROV at an angle of almost 180 degrees. The signals will reflect on surfaces, and the results help the pilot identify objects. Photos 2 and 3 are from ROVs with only a sandy bottom ahead. In photo 4, there is an oil well template coming up. Each grid cell is ten meters wide for photos 2 and 4.

MAP

As mentioned in section 2.2 , maps are not used very much during an operation. If there is a map, however, it is usually solely based on acoustic information, and will only show the position of the transponders on the seafloor as dots in a grid system. In other words, it is a live map with no other information than positions in 2D, as displayed in photo number five.

Photo 6
Display from the Oceaneering ROV Simulator showing direction, directional history, depth of ROV, depth of garage, altitude, netto rotations etc.

ALTITUDE
An altimeter is used to retrieve the distance to the sea floor from the ROV. The altitude is in many cases more important than depth itself.

DEPTH
A pressure sensor is used to measure the ambient pressure, giving an exact reading of the depth the ROV is positioned at. If there is a garage, the depth of the garage is also given.

DIRECTION
The direction is primarily measured with a compass, though may ROVs also have gyros to measure rotation. Direction is a primary source of navigation, and many GUIs also have a visualization of directional history

TURNS
The net number of turns the ROV makes around its own axis is important to avoid twisting the tether too much. At the end of an operation, that number should be 0. This measurement is also given for a garage, if present.

Photo 1: Inside the Surface Control Unit for the Minerva NTNU ROV.

5 CHALLENGES

SAFETY

Safety comes first when planning an ROV mission. There are many rules to follow and considerations to take.

> Safe distance

It is stated by law that a vessel stay a given safety distance away from a rig. The rig might experience traffic or perform dangerous operations. In addition, the vessel should not be able to crash into the rig due to human error. In case of failure, the vessel must not be upwind from a rig, in order to prevent blow-on situations. The same counts for currents, and they are necessarily the same direction as the wind.

> ROV position

The position of the ROV according to the vessel and other structures is at all times important to monitor. In case of failure, the ROV will float upwards, and it must therefore not float into the vessel and propellers, other installations or have their tether catch the tether of another ROV. It can not float up on the other side of the vessel from where it was dispatched.

If attached to a garage, the ROV can be pulled into the garage by the tether. Either way, crossing tethers is still a danger.

> The tether

The tether truly adds a dimension of difficulty for the ROV operator. This was clear during the ROV mission I witnessed with the NTNU Minerva ROV in March 2015: during a similar mission four days previously, the tether had been hooked around a transponder on the seafloor and the ROV had pulled it over. Depending on depths and if the mission is land based or from a vessel, the tether can be up to 1000 metres.

> Logging

In the NORSOK Standard U-102 (NORSOK, 2012) on ROV Services, there are several directions on how ROV personnel has to log events, list equipment and create timetables. It also says that each day, the crew has to hand in a plan for the next 24 hours. These procedures take time, and have a potential for improvement.

“ Driving that Mini-ROV is really difficult.
Student

“ 100 hours of practice helps.
ROV pilot

VISIBILITY
One issue for ROVs today is speed versus visibility. When navigating in water using cameras, high speeds will mean less visibility due to the debris in the water. This means that the pilot must pay extra attention when travelling at high speeds. With water that is heavy with debris, the light on the ROV will give backscatter into cameras that are close to the light. Often, these cameras are the ones pointing forward, which makes it even trickier to travel at high speeds.

VISUAL REFERENCES
During observation and interviews with ROV operators, I found that they like to use a visual references. This means that they will cruise close to the bottom due to the poor visibility. A variation in the bathymetry can lead to the ROV crashing into a slope. Often, crashing into the sea floor is harmless to the ROV itself, but the worst part is that sand is blown up from the bottom, and the visibility gets worse.

Even though the camera feed only has a delay of a couple milliseconds, the position of the ROV in the water is not updated as frequently because it uses acoustic signals. Depending on the technology used, the pilot will get position updates every 1 to 5 seconds. Also, if the ROV falls in the shadow of a transponder, the exact position of the ROV will become uncertain.

MENTAL ROTATION
The screens are not ordered in any way according to where the cameras are on the ROV. On the NTNU ROV Minerva, the feed from the camera looking slightly downwards is next to the feed from the camera looking straight forward, while in reality the latter is above the former. The map is also in 2D from above and is also set to one zoom level that usually covers the entire field relevant to the mission. Consequently, it is difficult to use the map as a guide to perform accurate maneuvering.

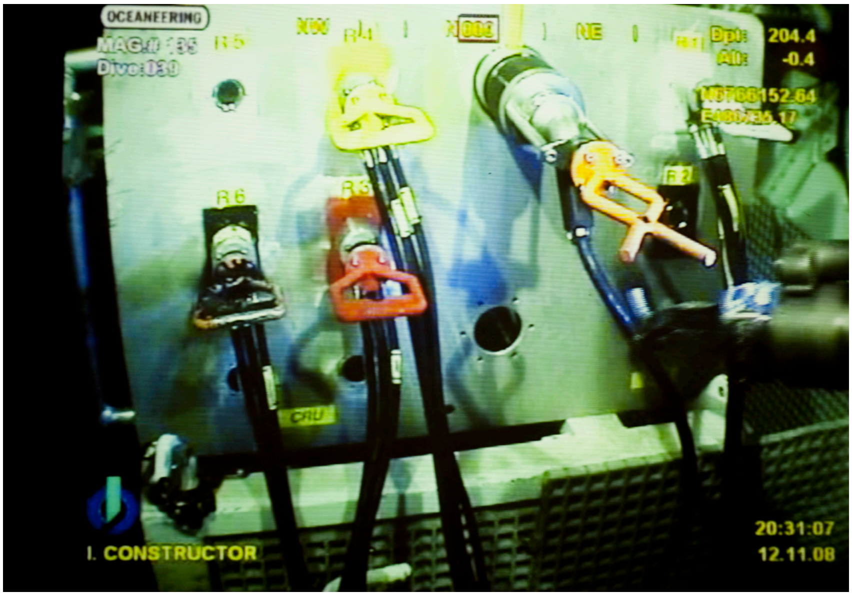


Photo 2: Video feed from a real ROV mission performed by Oceaneering. The ROV is working on a template at the bottom of the sea.

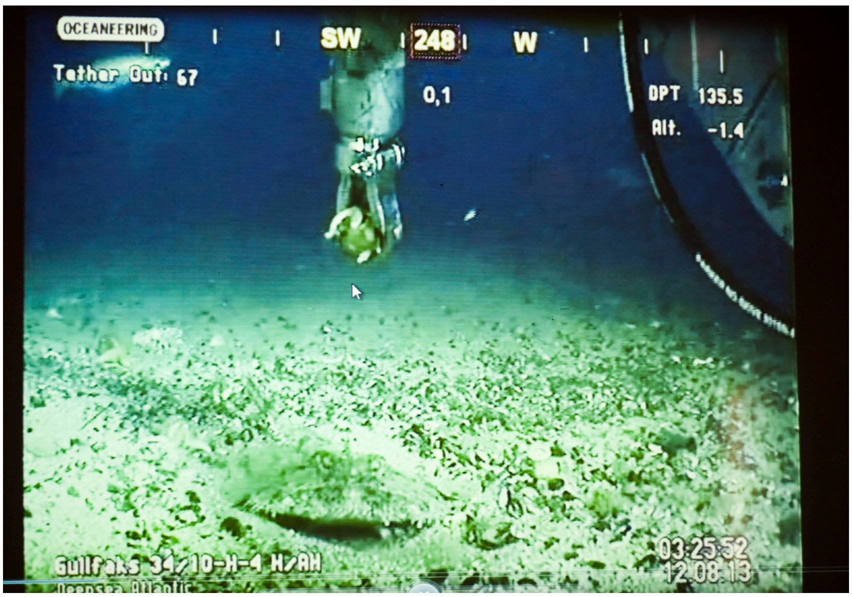
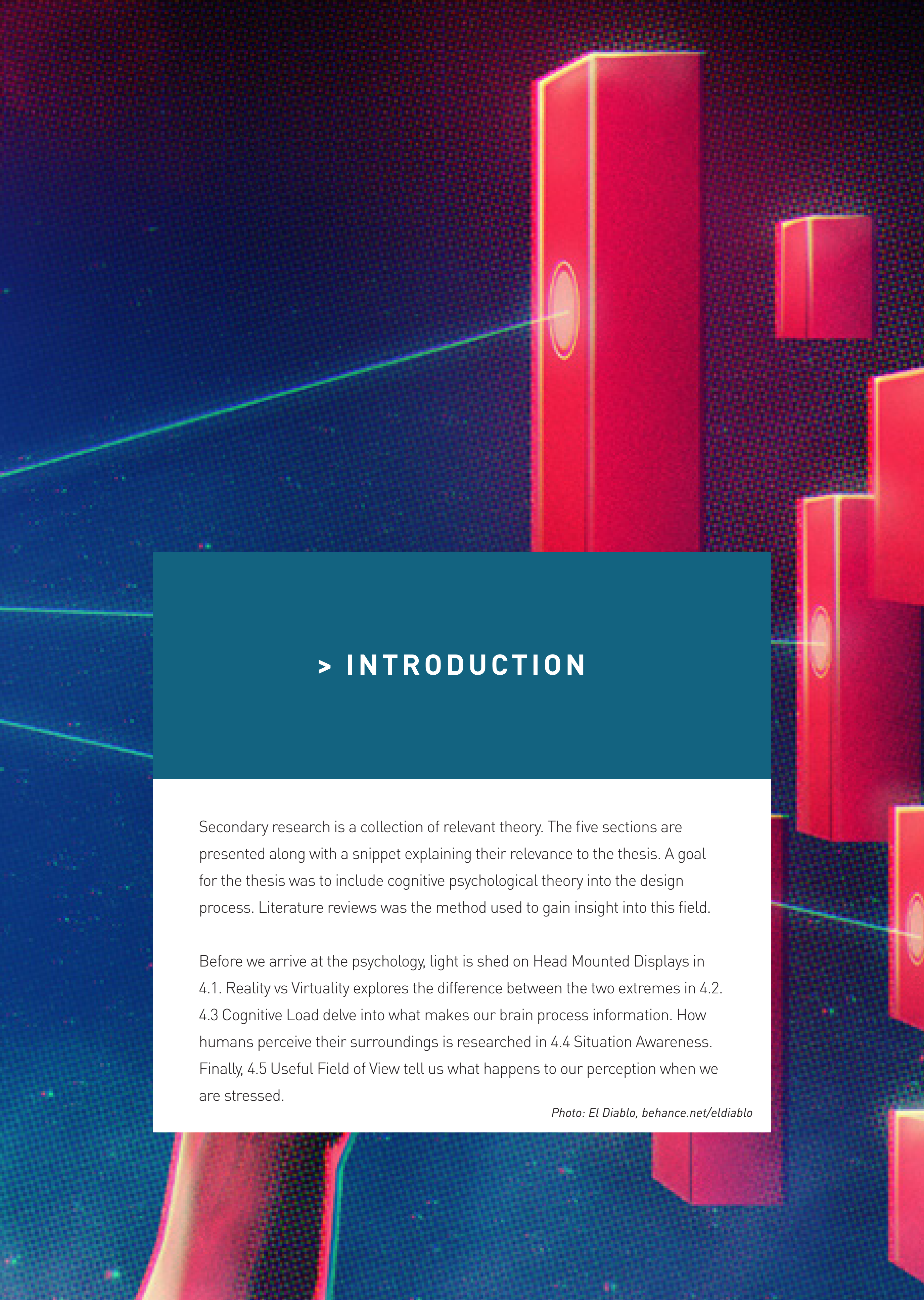


Photo 3: Video feed from an Oceaneering ROV. The ROV is trying to feed a flounder.

4 SECONDARY RESEARCH

>	Introduction	65
1	Head Mounted Displays	67
2	Reality vs Virtuality	69
3	Cognitive Load	71
4	Situation Awareness	75
5	Useful Field of View	79



> INTRODUCTION

Secondary research is a collection of relevant theory. The five sections are presented along with a snippet explaining their relevance to the thesis. A goal for the thesis was to include cognitive psychological theory into the design process. Literature reviews was the method used to gain insight into this field.

Before we arrive at the psychology, light is shed on Head Mounted Displays in 4.1. Reality vs Virtuality explores the difference between the two extremes in 4.2. 4.3 Cognitive Load delve into what makes our brain process information. How humans perceive their surroundings is researched in 4.4 Situation Awareness. Finally, 4.5 Useful Field of View tell us what happens to our perception when we are stressed.

Photo: El Diablo, behance.net/eldiablo

1 HEAD MOUNTED DISPLAYS

VIRTUAL REALITY

Virtual reality is a concept that was born with the emergence of computer graphics. In the 90's, the development of HMDs seemed to make virtual reality a commodity. The HMD enables the user to be immersed into a completely virtual world. Turning your head will turn the view in virtual reality in the same way you would expect in real life.

However, the state of technology was not advanced enough at the time, and the user would feel trapped, looking through two holes. Nausea was also very common (Aftenposten, 2014), and the development stopped in the mid 90's as focus shifted towards the development of the world wide web.

Today, technology has come a long way. Computers are thousands of times faster, the images almost fill the visual span of the user, and computer graphics has become an advanced field, creating content that can trick the user into thinking that what they see is actually real.

On the horizon, there are several companies developing VR-hardware and software. Oculus with Rift and Sony with Morpheus are leading the race (The Verge, 2015), but Microsoft, Google and HTC are all developing their own systems. It seems quite clear that in the rising world of ubiquitous computing, virtual reality definitely will earn its place in the spotlight the coming years.

HMDs
Head mounted displays

OCULUS RIFT

The HMD developed by Oculus, a company recently bought by Facebook for 2 billion dollars, is the most renowned in the VR community at the moment (Oculus VR, 2015). The developer kit 2 improves graphics, the updating frequency is higher, the responsiveness is vastly improved and they have introduced head tracking so the user can peek around corners and put their heads through car windows.

RELEVANCE FOR THESIS

Being able to immerse the user into the world he is navigating in have a promise to relieve the user of a high cognitive load due to difficult mental rotations associated with the classic 2D projection map. Although technology today may not be perfect, many experts claim that the VR and HMD industry will explode in the near future.

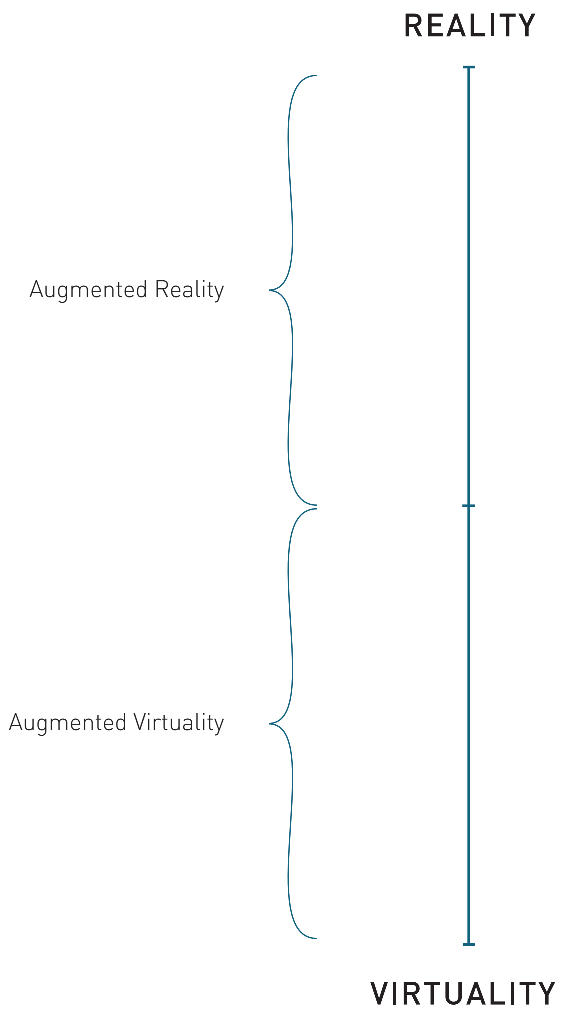


Figure 1: The scale from reality to virtuality. Even though the development revolves around a virtual environment, real aspects can be simulated to make the system more useful and effective.

2 REALITY VS VIRTUALITY

While developing the virtual underwater environment, I was struck with two opposing factors: when will it be beneficial to accurately simulate the real world and when should more precise computer simulated experiences be emphasized? For example, should the positive buoyancy of the ROV be simulated in the virtual world? Should the system simulate the fluid mechanics of the water?

SIMULATION

Simulation is defined as the imitation of real-world situations or actions. Computer simulators are made to encompass a wide range of uses, from games made purely for entertainment to complete space station simulators made to prepare astronauts for space travel. Simulations can also be strictly numerical, though this is outside the scope of this thesis. The different uses have different demands for how close to reality they need to be. The motivation of the user in the context of the simulation is therefore relevant.

MOTIVATION

Players of video games who have different motivations, also choose different strategies to complete the game. Pasch et al (2009) identify two different motivations for players of exertion games: Relax and Achieve.

exertion games
physical games including the whole body, most commonly known through the Wii and Kinect, such as Wii Golf

Players with the motivation to achieve will often use a "game" strategy and not a realistic strategy to perform better in-game, while players who play to relax will often choose a more realistic strategy because they get pleasure from moving.

The former was apparent when testing which controller could steer the ROV through a track the fastest and the motivation for the users were obviously to achieve. Let us say that the maximum speed in any of the translational axis of freedom is 1. After several trials, it became apparent that the user could combine the forward speed of 1 and the slide movement of 1 to create a combined speed vector with a magnitude of $\sqrt{2}$. By rotating the ROV 45 degrees to the forward direction, the player could improve the final time by several seconds. This will not work with a real ROV.

RELEVANCE FOR THESIS

In the context of this thesis, the motivation of the user will be to achieve. However, the way they achieve is by planning the most precise and realistic scenario for real-world execution. The goal should be to create a simulation where there is little possibility for a "game" strategy to give an unrealistic outcome, and the pitfalls of such a strategy is apparent to the user.

Figure 1: An example of what a dog schema might look like.

3 COGNITIVE LOAD

"Cognitive load is the total amount of mental activity in working memory at an instance in time (Cooper, 1998)". Cognitive ability is limited and if this limit is exceeded when performing one or more tasks, performance on these tasks will drop (Cohen et al., 2004).

To be as effective as possible, a system of any kind should therefore be designed to keep cognitive load beneath the crucial limit of the user. The primary task should therefore allocate most of the mental resources of the user, while secondary tasks should only demand attention when necessary for the primary task.

SCHEMAS
The critical limit for how much information working memory can process varies, not only from person to person, but also due to fatigue and stress etc. However, it is clear that domain expert users perform better than beginners. This is due to schemas. People use schemas to organize information of the world and provide a framework for future understanding (Anderson, 1990).

For example, imagine you see a new breed of dog for the first time. How do you know it is a dog? The reason you probably recognize it as a dog, is that you have your previous knowledge of dogs organized as a schema. The schema contains information saying that a dog is furry, four legged, it barks and so on (figure 1). Since the new breed of dog fits the schema, you recognize what it is. At the same time, your schema is expanded to include the new breed.

domain expert users
experts in a specific domain, not necessarily in computer science, who use computer environments to perform their daily tasks (Costabile et al., 2003).

Domain expert users have more developed schemas than beginners, and is faster to link new information up with relevant actions or behavior. When it is easier to link new information up with existing information, it is more likely to stick in long-term memory and easier to retrieve later. Therefore, an expert can process more information without experiencing cognitive overload.

An ROV operator is a domain expert user. They have well-developed schemas on what it is to operate an ROV and what they need to do as new information is given to them.

OVERLOAD

Cognitive overload is separated into two categories: saturation and pollution. The first occurs when the current task simply requires too much processing for the mind to deal with (figure 3). The second happens when too much unnecessary and flawed information is given to the user, polluting his attention (figure 2).

Poorly designed interfaces can cause both events to occur simultaneously (Tracy and Albers, 2006).

RELEVANCE FOR THESIS

As I design the virtual representation of controlling an ROV, I must balance between adjusting to existing schemas or creating new ones. It is also important to avoid that the user experiences cognitive overload.

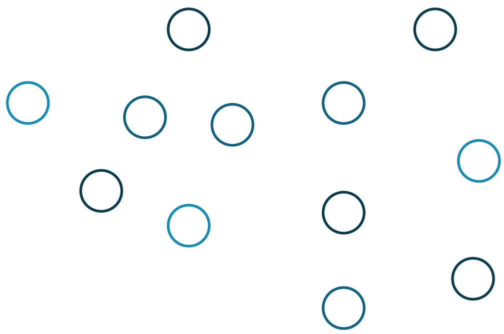


Figure 2: Cognitive overload by pollution occurs when too much unnecessary information given to the user.

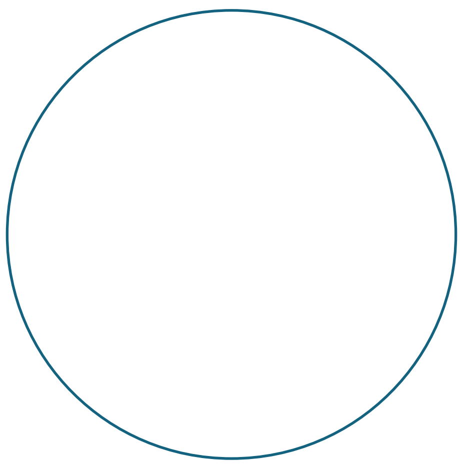


Figure 3: Cognitive overload by saturation happens when the current task requires more prossesing power than the mind can deal with.

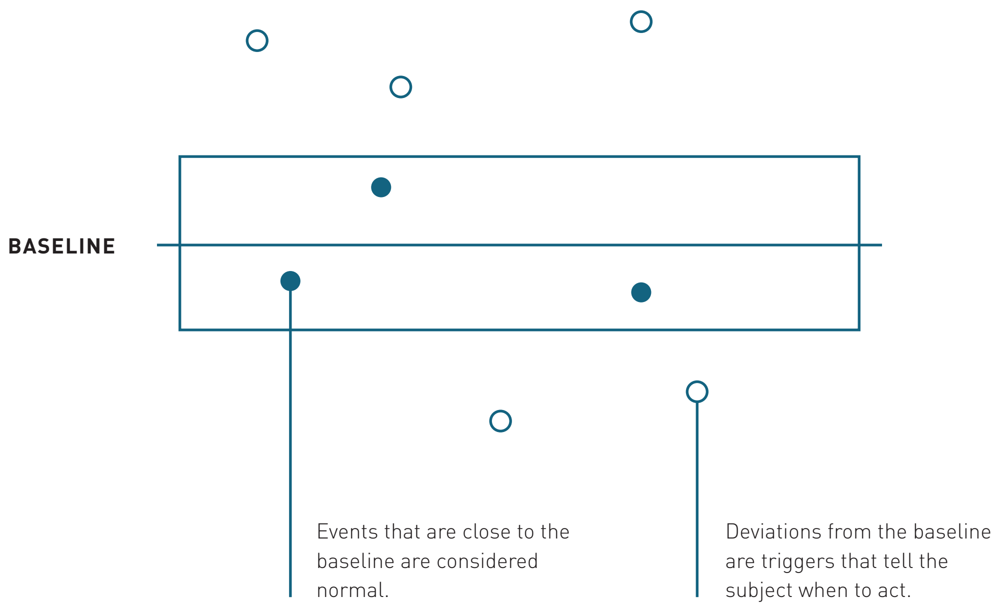


Figure 1: Establishing a baseline is important to filter out "noise", so that deviations from average behavior is recognized and acted upon.

4 SITUATION AWARENESS

How do you know where you are in space? Or in time? Or why you are in a specific place at that specific time?

To answer this question, psychologists have developed the term "situation awareness". A broadly acknowledged definition amongst most fields is the one from Endsley (1995): "the perception of elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future". In other words, situation awareness is defined as being aware of what is happening around you, and to understand and act upon those events.

Many people might confuse situation awareness with spatial understanding. Spatial understanding is a primitive version of spatial awareness, and only relates to the understanding of the surrounding 3D environment. A good measure for situation awareness is the ability to recognize important events and deciding how to respond to these, or, alternatively, to filter "noise".

BASELINE

To master situation awareness, it is important to establish a baseline. The baseline is how the surroundings behave under normal conditions. It requires experience in any surrounding to determine what the normal conditions are. The baseline includes the behavior of other people, objects, lights, sounds, weather and so on.

Deviations from the baseline are the triggers for the mind to know when to act. If something changes, then the subject should be able to recognize the change, understand why it is changing and then decide on an action according to that change. The brain subconsciously makes any number of these decisions every day, most of which is to do nothing. When someone opens a window in the classroom, you might recognize the event, but without thinking about it, you would normally just continue working.

THREATS

There are numerous "threats" to reduce a subjects situation awareness, and they are all related to a high cognitive load reducing the useful field of view (see next section). In everyday life, mobile phones is one of the greatest threats to situation awareness, as they require a lot of mental activity from the user, and block out input from the surroundings.

Change blindness is the term used when a user fails to see a change in the surroundings. It often occurs that, even though the user has seen the change, as monitored with eye-tracking, the user still fails to pay attention to it (Weinschenk, 2011).

An example situation that is prone to change blindness is when you are driving a car: you are within the speed limit of 80 km/h and you know it. Over time, there is an increase in speed and you are slowly passing the speed limit. However, it is really difficult to see the small, incremental increase in angle for the needle in the speedometer, and that you are now driving illegally is not noticed (figure 2).

RELEVANCE FOR THESIS

As mentioned, situation awareness is related to cognitive load, and the same considerations applies. However, in the virtual environment there is also an opportunity to help the user establish a baseline faster than normal, and to continuously draw the user's attention to relevant changes. Another takeaway is that by practicing an operation in an open, virtual environment, spatial awareness in the following real life operation will be improved.

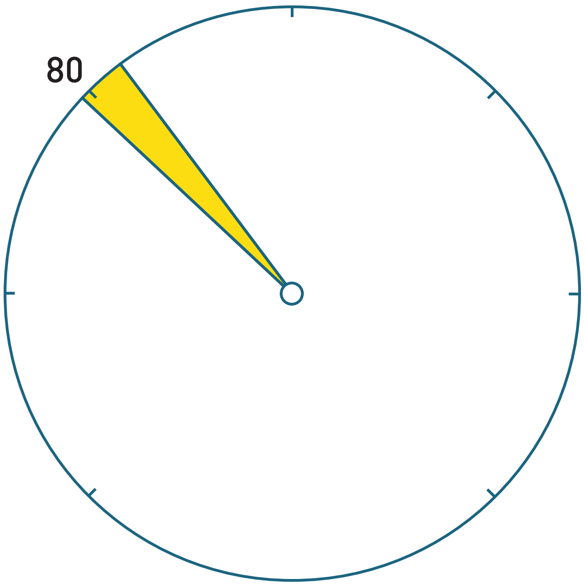


Figure 2: The change between legal and illegal driving is only indicated by a small change of angle in the speedometer.

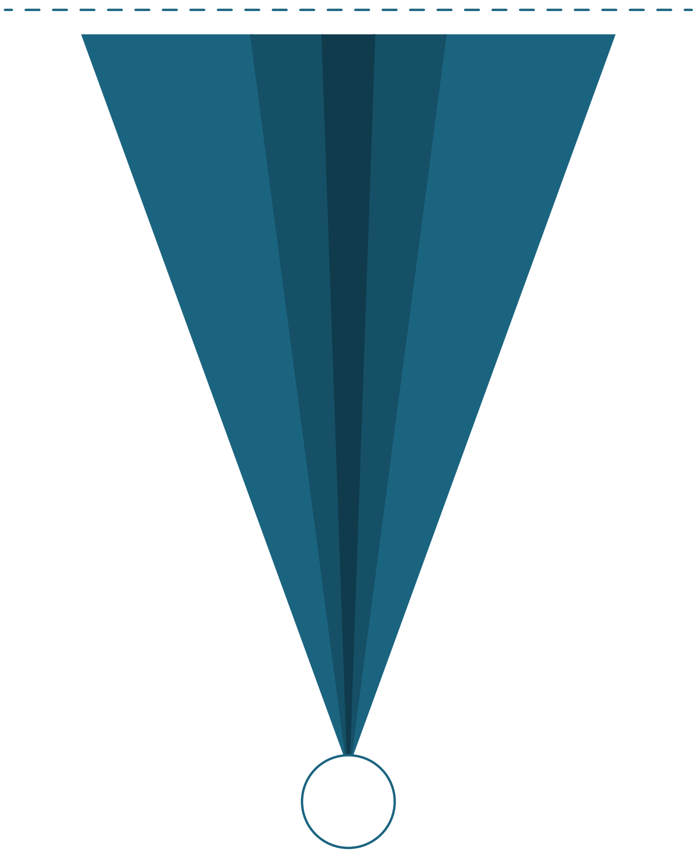


Figure 1: Difference between a UFOV of 40°, 15° and 4°.

5 USEFUL FIELD OF VIEW

The attention system is concentrated around the fovea, where vision is most detailed (Ware, 2012). However, the visual field we pay attention to can vary based on the task, stress level and information density. Useful Field of View (UFOV) defines the size of the field of vision where information can be retained.

fovea centralis
the part of the eye which gives sharp central vision. In humans, this vision is only 2.5 degrees wide

In information-dense interfaces, the UFOV can become very small, even smaller than the sharp central vision (Wickens, 1992). On the other hand, less information-dense interfaces have a larger UFOV, around 15 degrees (Drury and Clement, 1978). In general, the UFOV varies to account for a constant number of targets.

For moving targets, Peterson and Dugas (1972) showed that we can expect an even larger UFOV. Up to 40 degrees of the visual field was tested and users could responds in less than a second to moving objects, compared to 4 degrees for static objects. Possibly, the UFOV covers the entire visual field for moving targets (Bartram et al., 2003). This implies that additional visual cues (such as blinking) can be added to objects to increase probability of users noticing them.

COGNITIVE LOAD

As mentioned in 3.4 Cognitive Load, cognitive load is a measure on the total amount of mental activity in working memory at any instance. The UFOV is closely related to cognitive load. Tunnel vision is a familiar concept associated with people under stress, where the UFOV is narrowed to only accommodate for the most important information at the center of the field of view (Ware, 2012).

Williams (1985) found that as tasks became more complex and hence initiated a high cognitive load on the subjects, objects in the peripheral vision were increasingly more difficult to detect. In other words, as cognitive load increases, the UFOV decreases. Mack and Rock (1998) adds to this notion by showing that unless we have some expectations to see an object, humans will miss it most of the time.

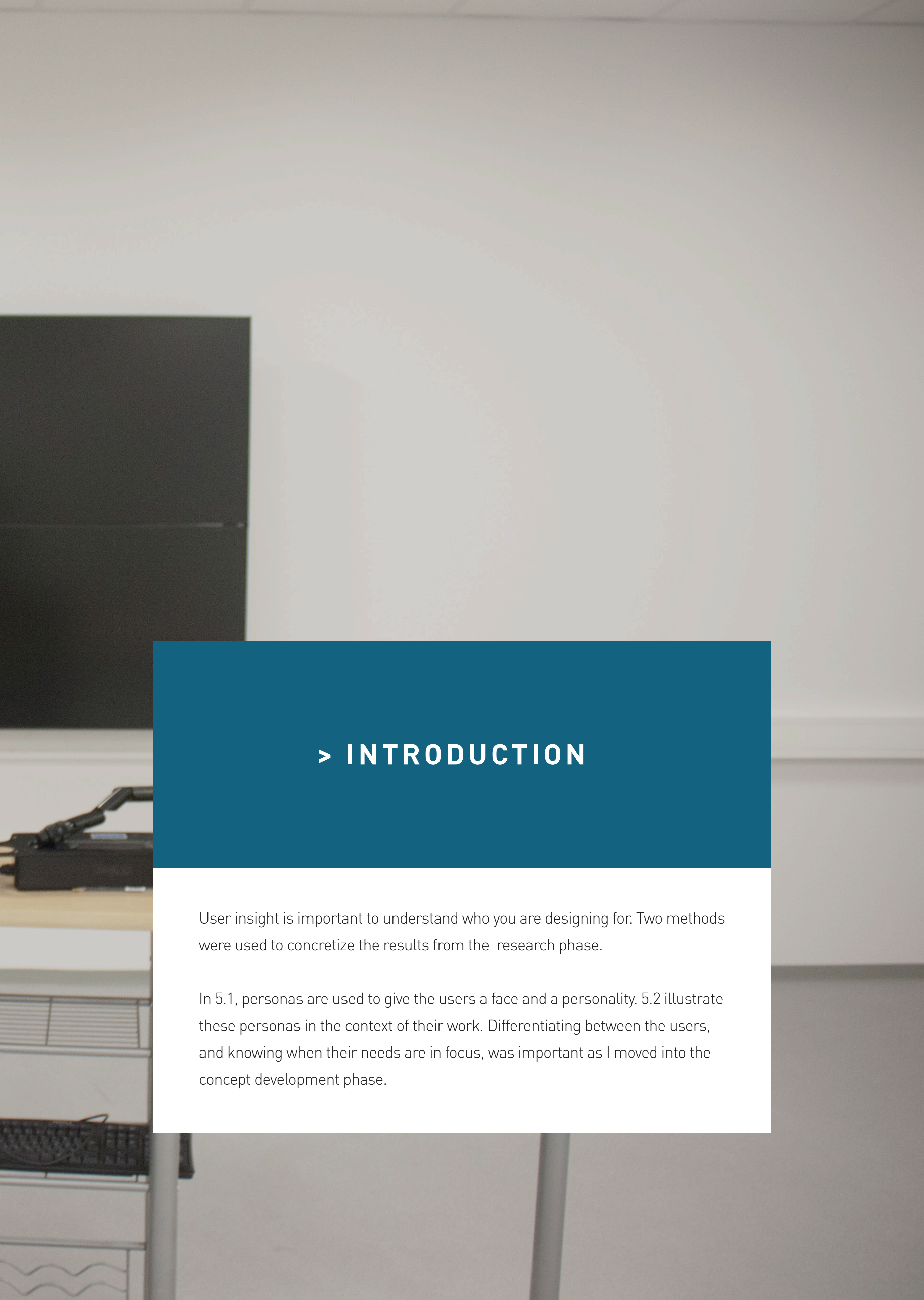
RELEVANCE FOR THESIS

Planning an ROV mission by moving several different objects around in a 3D environment demands a high mental activity of the user. It is important to retain a large UFOV, and to give crucial information at a central point in the UFOV.

5 THE USERS

>	Introduction	83
1	The Users	85
2	User Scenario	93

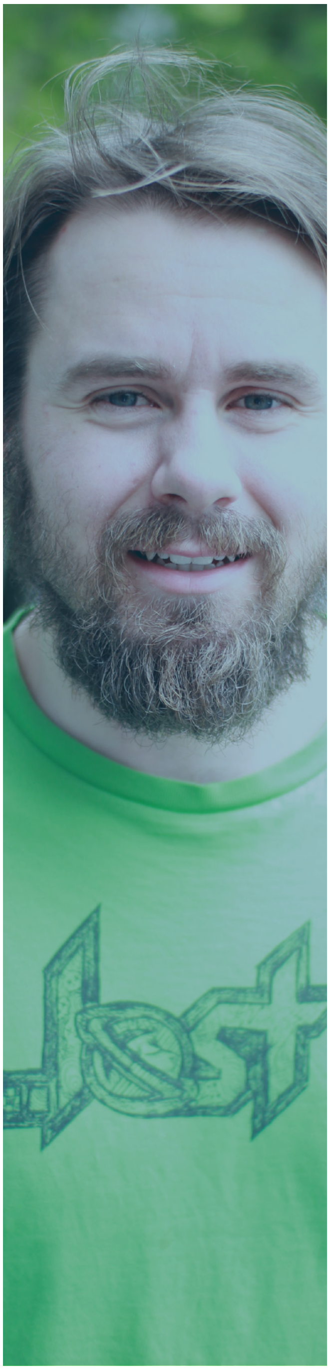

Your perfect team player
OCEANEERING

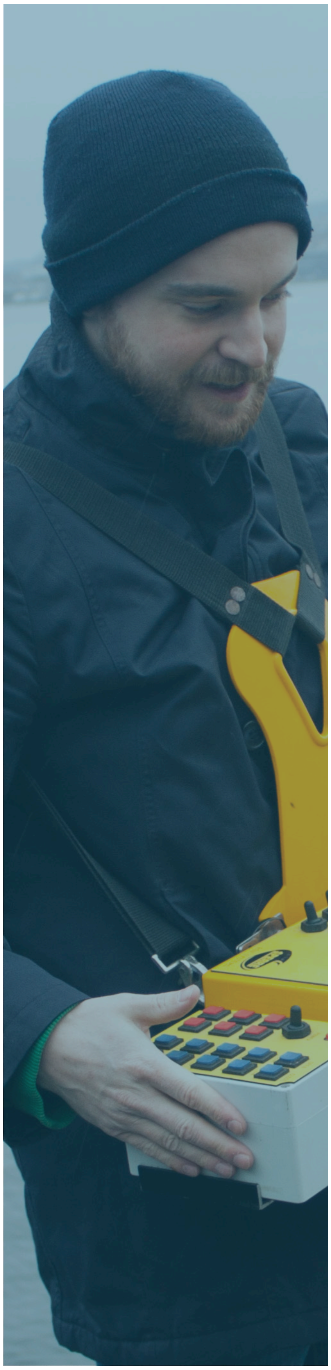


> INTRODUCTION

User insight is important to understand who you are designing for. Two methods were used to concretize the results from the research phase.

In 5.1, personas are used to give the users a face and a personality. 5.2 illustrate these personas in the context of their work. Differentiating between the users, and knowing when their needs are in focus, was important as I moved into the concept development phase.





1 THE USERS

Understanding the user is a keystone in creating an experience users want to use, and keep on using, to fulfill their goals. Loranger (2014) states: "the first requirement in creating an exemplary User Experience is to meet the exact needs of the customer".

In the user-centered design process, personas are used to create a deeper understanding of who we are designing for and is a description of who the users actually are. Personas are divided into primary users, secondary users and sometimes tertiary users or users with special needs. The personas are given believable traits to help the designer visualize the user. In the design process, the needs of the primary user have a "right of way", while the needs of the secondary and tertiary user come are considered additions (Mulder and Yaar, 2006).

In this thesis, the definition of the user is related to their line of work; the primary user is an ROV supervisor at Oceaneering, who delivers ROV services. The secondary user is a project manager at Statoil, who hire Oceaneering to do work for them. Included is also an ROV pilot at Oceaneering, although he does not have a leading role in planning the ROV mission. He may, however, benefit from an improved situation.

PRIMARY USER

> “ The pilot loses the intuitive feel with the water with automatic functions

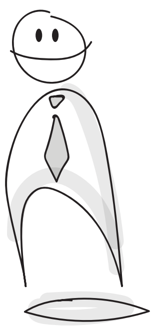
HENNING GRAN (36)

FROM BERGEN
WORKS AS AN ROV SUPERVISOR AT OCEANEERING

JOB
> *Plan ROV missions for clients*
> *Find smart solutions to complex problems*
> *Study safety regulations*
> *Use his experience as a former ROV pilot to lead ROV missions*

GOALS
> *Plan effective missions to reduce costs*
> *Maintain a good reputation with the client*
> *Keep missions within safety regulations*

CONCERNS
> *Damage the company's relationship with the client*
> *Work is always far away, and requires many miles travelling*



SECONDARY USER

Photo: Johannes Jansson / norden.org

> Every minute wasted costs thousands

CARL DITLEFSEN (44)

FROM ÅLESUND
WORKS AS A PROJECT MANAGER AT STATOIL

JOB
> Plan ROV missions
> Tender the ROV mission to the market
> Hire an ROV company, at a good price
> Supervise the ROV mission

GOALS
> Plan quick and accurate missions to reduce costs
> Get promotion
> Buy new car

CONCERNS
> Accidents that damage the company's reputation
> Accidents that damage the environment
> Wasted time/money during ROV missions
> ROV pilots damaging company property

TERTIARY USER

> You go in and out, it's like driving a car

BREDE OLAUSSEN (29)

FROM STAVANGER
WORKS AS AN ROV PILOT AT OCEANEERING

JOB

> *Occasionally help plan ROV missions*

> *Learn about: the site, the components, their locations, the bathymetry, the route of the ROV and which other ROVs, machines and boats are present.*

> *Control the real ROV in a mission*

GOALS

> *Be effective*

> *No errors*

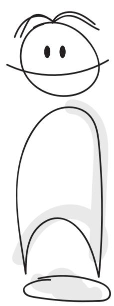
> *Communicate well with others on the team and the client*

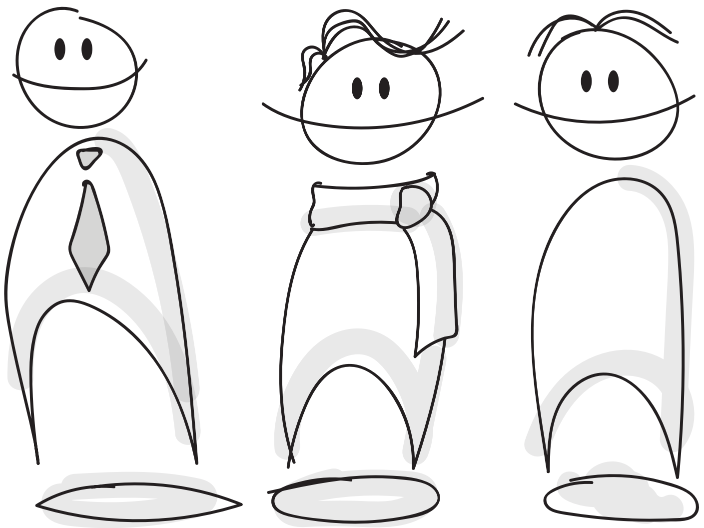
CONCERNS

> *Losing track of the ROVs position*

> *Wasting time during a mission*

> *Damaging the ROV or the rig-components*



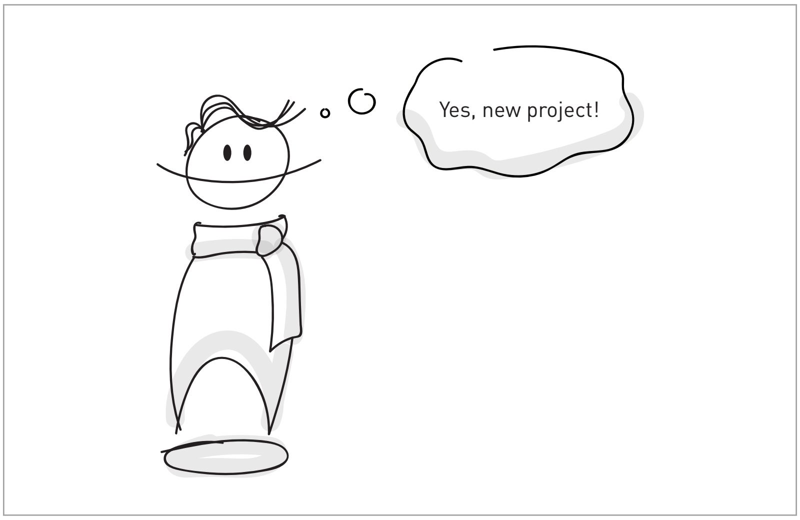


2 USER SCENARIO

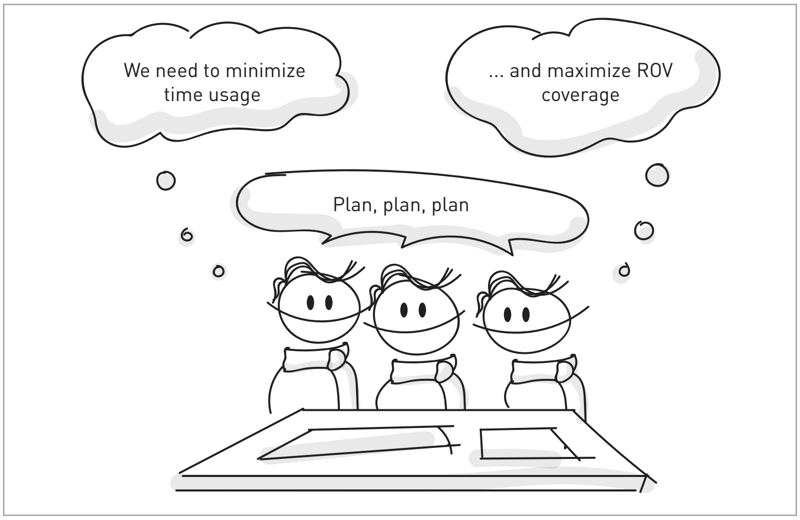
User scenarios are used to visualize how the users perform the action or service that is researched (Rosson and Carroll, 2009). This section is used to highlight how an ROV mission is planned and executed, through the eyes of the personas. The scenarios are a result of the depth interviews and observations of industry professionals, and represent a realistic take on ROV missions. I have to add that this view is not true to all situations. Therefore, the scenario is just as important to show the role of the personas in the big picture, as it is to show the procedure itself.

The emphasis of the scenario is on the planning of an ROV mission, but the execution is added at the end to show the roles of the personas in this context as well.

PLANNING

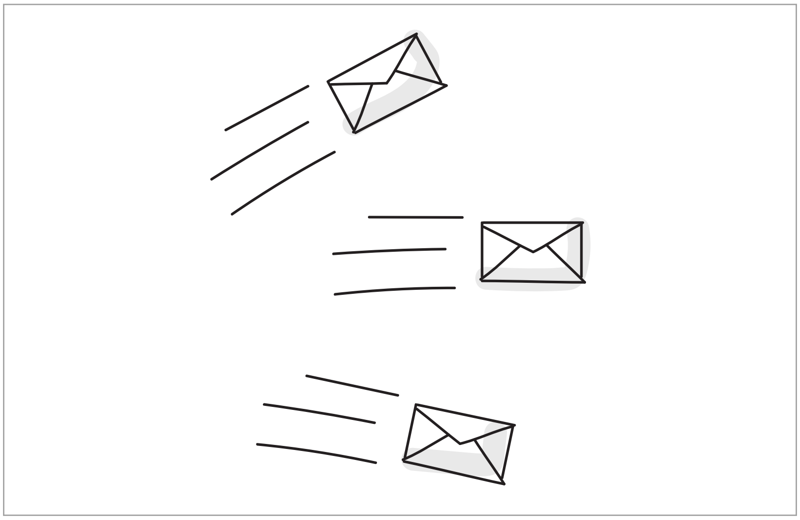


Carl, project manager at Statoil, is excited to start a new project, an ROV mission on the seafloor below the Troll platform.



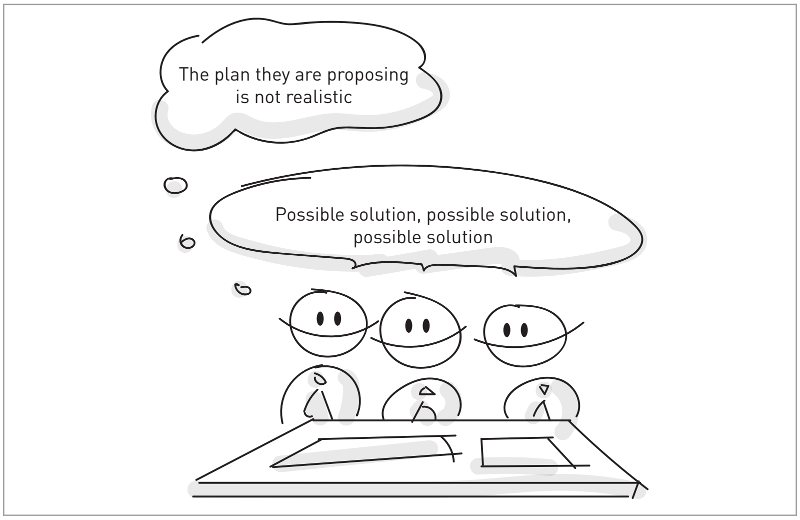
He starts planning the mission with help from colleagues at Statoil, a rig operator in the oil and gas industry. Their focus is on minimizing costs.

Carl creates a draft of the mission, creating a "problem" statement for the mission. It is important to be accurate, as Statoil will have to outsource the mission.

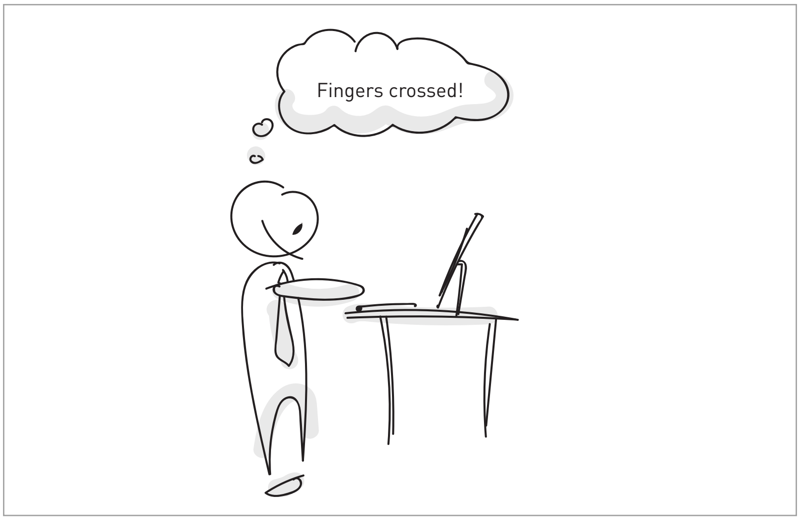


When the job proposal is done, the job is tendered to the open market. All interested ROV firms receive the job proposal.

Henning, an ROV supervisor at Oceaneering, an ROV firm, scans through the job proposal sent out by Statoil.

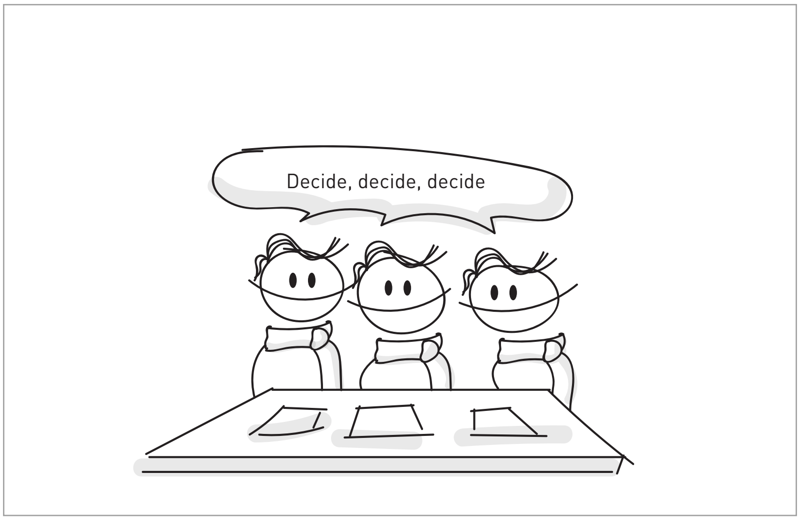


Henning gathers some colleagues to create an offer on the job proposal. The specify how they think they can solve the job, and what type of vessels and ROVs they can provide.

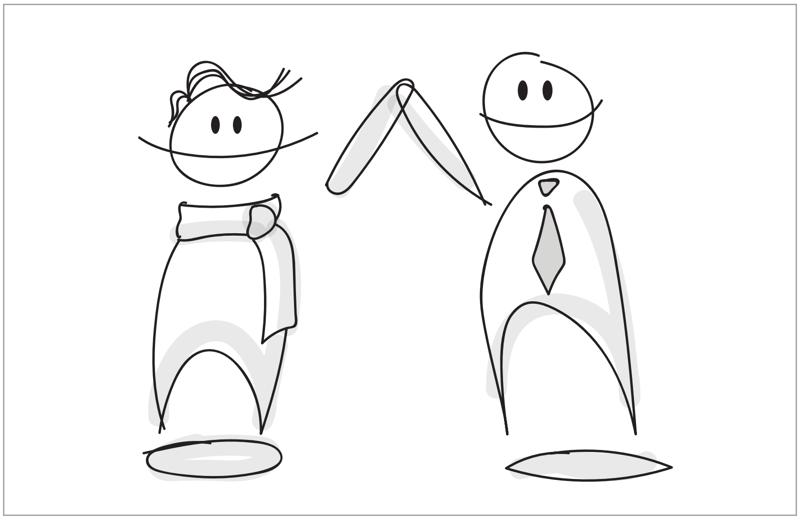


Henning sends the offer to Statoil.

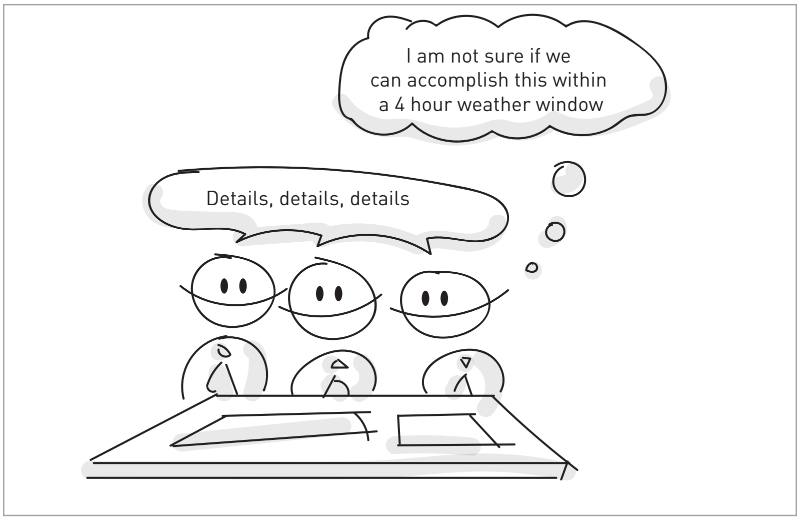
Back at Statoil, Carl has received offers from several ROV firms. He decides to make the decision with his colleagues.



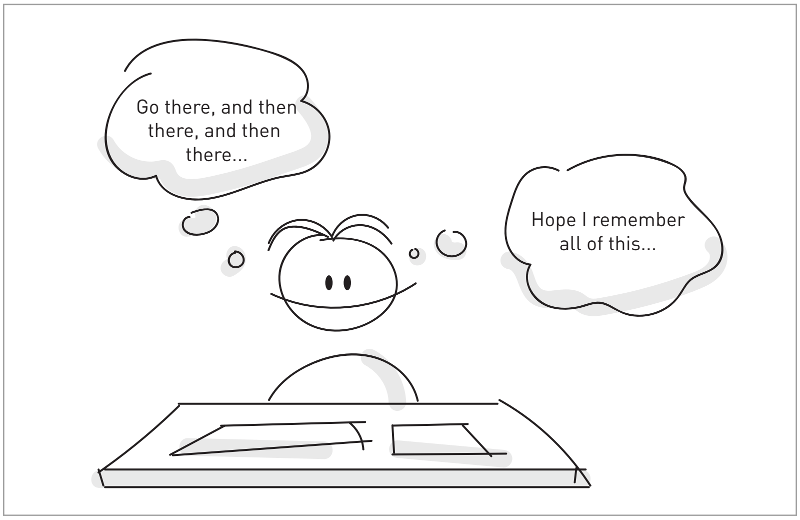
Carl gets together with his colleagues to decide which offer is the best. They end up choosing Henning's firm, based on price, proposed solution and past experiences.



Statoil and Oceaneering are now ready to cooperate, and both Carl and Henning are excited.

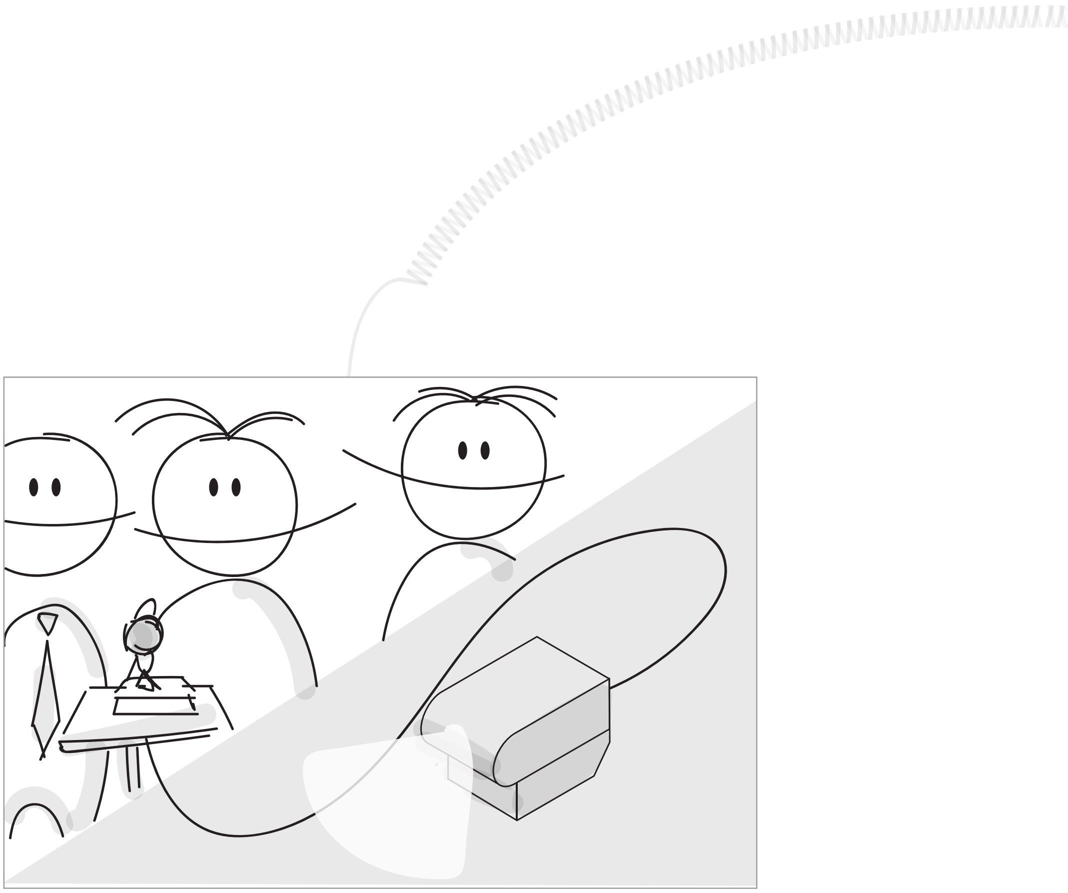


Henning and his team at Oceaneering start planning the mission in detail. They have to take into account many safety measures related to the nearby platforms, ships and the weather.

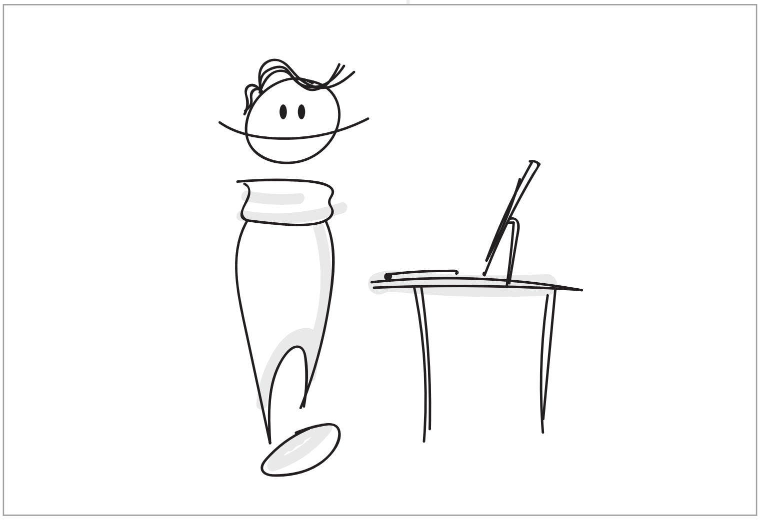


Brede, the ROV pilot, is given the full plan of the mission. He studies the plan, 3D models of the installations he is going to work on, and a map of the site. It is important that he always knows where to go, even in the dark.

EXECUTION



As the mission is executed, Brede is the main-pilot of one of the work class ROVs in the mission. He has a co-pilot and Henning, the ROV supervisor, with him in the control unit. The trio communicate with the client and the other ROV teams through an open communications line.



Carl is in his office on shore, but follows the operation by watching a video feed on his computer. He also communicates with the pilots, and lets them know if they are doing anything wrong, or if he wants them to do something different. This mission is executed normally, and Carl is satisfied he does not have to intervene.

6 CONCEPT DEVELOPMENT

>	Introduction	105
1	Within-Group Design	107
2	Prototyping & Testing	109
3	Controllers	111
4	HMD vs Screen	119
5	On Immersion	139


Oculus

> INTRODUCTION

The goal for the concept development is to use the insight gained in the previous chapters to move towards something tangible. 6.1 Within-Group Design explains the theory behind how users were selected for user testing. The next chapter, 6.2 Prototypes and Testing, introduce the different prototypes and prototyping methods used. 6.3 Controllers display the controllers tested, and show results from the main controller user test. 6.4 HMD vs Screen talks about the main user test with Oculus Rift and navigation in a 3D environment, and discusses the results. The final chapter, 6.5 On Immersion, brings up the thematic from an ealier chapter and important aspects discovered during testing.

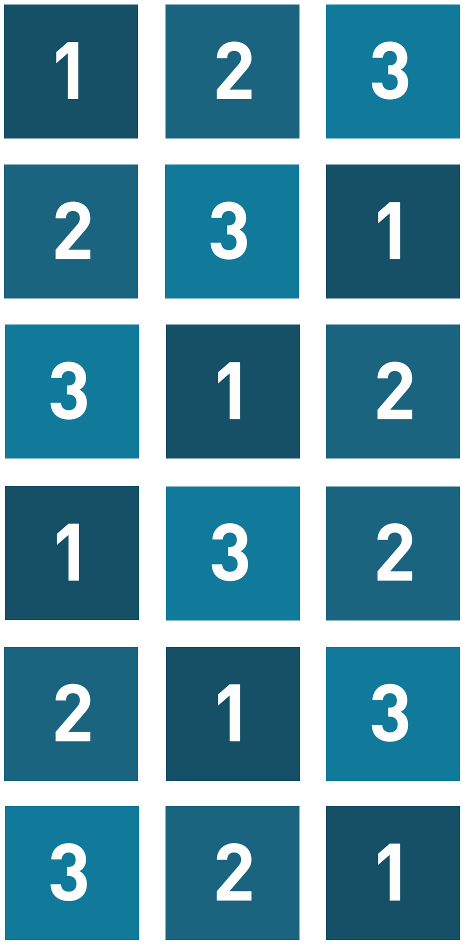


Figure 1: The six rows show the six possible ways a user can be tested with three conditions.

1 WITHIN-GROUP DESIGN

While conducting some early guerilla user testing on the first interactive Unity prototype, where the goal was to compare different type of controls and find the most effective one in terms of speed through a track (section 6.3 Controllers), I realized that no controls are intuitive enough for the user to be proficient from the first try. Before starting the test itself, the user had to learn how to control the vehicle using the different axes in the controls. This means that for every new user, a lot of time goes into just bringing the user up to speed on the fundamentals of the controls.

In order to get the most useful qualitative feedback on user testing and as statistically correct quantitative data as possible, I have conducted research using the within group design method (Greenwald, 1976). This method aims at testing all conditions on all users.

ADVANTAGES

The primary reason for using within group design for this project, is due to the limited number of users to test on. If divided into separate groups, individual differences may affect the results.

Another important reason is due to the value of qualitative feedback in this project. The response will be better if the users themselves have a basis to compare with and can give their pers on the differences.

DRAWBACKS

Learning effects from previous tests might induce better results in the next, because the ability to control the ROV is better and enhances the feeling of achievement. In some cases, the user might get tired cognitively and the performance drops, or the user might get bored and will not perform at the highest level.

To counteract this, the users will perform the tests in different orders. The tests will also be relatively short and concise to get a user in a constant mental state.

Photo 1: The user explains how he would move the paper box with gestures.

Photo 2: The user makes gestures, and I try to translate them by moving the chair. Interesting discoveries were made when intention and translation differed.

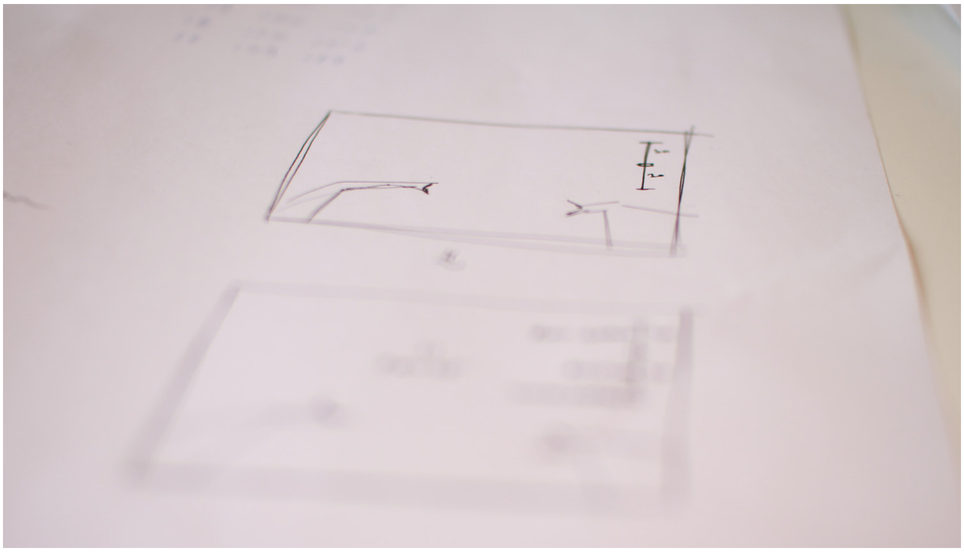


Photo 3: Example of a low-fidelity paper sketch of the GUI presented to the user when navigating the virtual 3D world.

2 PROTOTYPING & TESTING

As mentioned in 2.4 Analysis, the concept development was based on rapid user testing of prototypes. On these pages I will present the main prototypes that were tested, how and what came out of it.

LOW FIDELITY

MAGIC

Two magic prototypes were tested. The first was a paper box held in front of the user, where the user would explain how he wanted to move it around and how he wanted to follow it as it moved (photo 1). For the magic second test, the user would sit in a rolling chair, trying different gestures, and I would push the user according to the gestures (photo 2) .

> Results
The feedback was very good, and got me started working on a highly conceptual level. Even though gesture control was abandoned early in the process, it gave me a better idea on how people relate to moving objects in 3D.

PAPER

Paper prototypes are very common in testing GUI, both on a system scale and a small component scale. Wireframe sketches were given to the user, and their behavior and feedback was noted.

> Results
Paper prototyping helped shape the user flow, the relation between "white" space and "black" space, and language.

HIGH FIDELITY

UNITY

Three main functional tests using the game development platform Unity were made. Two mainly for testing controllers and one mainly for testing navigation in a virtual 3D world using either Oculus Rift or a regular computer screen. Later, the navigational test was expanded to also test GUI.

In addition to the qualitative feedback I received, I wanted quantitative results. To get quantitative results, the testing had to be planned in a more thoughtful way. In the next two sections, I will present the tests conducted to get quantitative and qualitative data.

The tests used within group design and counterbalanced measures design to get comparable results from the participants and to exclude any carry-over effects.

3 CONTROLLERS

This section is dedicated to the physical interface between the user and how the ROV translates in the virtual world, namely controllers.

A range of controllers were researched and tested, based on availability and my own interest. The tested controllers are shown on the next two pages.

Prototypes were used as a means of testing the controllers.

First, I will shortly present test methods and a summary of results for controllers that were subject to simple tests, and that were not considered in the further concept phase due to those results.

Then I will present more a more substantial test of the remaining controllers, how they were designed and conducted, the results from the tests and a discussion of those results. This will eventuall lead to a small summary of each controller.

PS3 CONTROLLER

Wireless game controller for the Playstation 3 game console. Can be connected to a computer using Bluetooth.

KEYBOARD

Use the arrow keys and the WASD keys to control the axes.

SINGLE JOYSTICK

The joystick is used as a control device for many ROV systems. Three axis control. The gaming version connects with USB to the computer.

DOUBLE JOYSTICK

Two joysticks combined are used by some ROV systems. These joystick do not rotate, and control four axes combined.

LEAP MOTION

Gesture sensor. Recognizes both hands, its position and gestures. Uses USB to connect to the computer.

SPACENAVIGATOR

The 3DConnexion SpaceNavigator is a 3D mouse. It offers control in all six possible axes, and connects via USB.

MOUSE

The mouse is already used in many computer games to control direction. Can connect with either USB or Bluetooth.

Photo 1: The Leap Motion test. The user moves his hand to move the colored background on the x-axis. A dot is added for positional reference.

Photo 2: The SpaceNavigator was used to navigate in the Unity development scene, because I was not able to make it work in the actual game.

LEAP MOTION

METHOD
A small prototype was created in processing. The user had to position a background of colors according to a dot on the screen (photo 1), using hand position tracking by the Leap Motion. The background followed the hand; if the hand moved to the right, so would the image. Further to the side, the faster the image would move.

SUMMARY
I, the author, was the only tester. It was very difficult to accurately position the background. The perception of the hand was not constant. Despite the feedback from the image on the screen, I had trouble finding a sweet spot between too fast and too slow. These results showed no promise in the further research, and the Leap Motion was not considered from this point and onwards.

SPACENAVIGATOR

METHOD
It proved difficult to use the Spacenavigator in the Unity prototypes due to technical issues, but I was able to navigate in the development scene itself, and could therefore get a good idea on how the controllers would work (photo 2).

SUMMARY
I, the author, was the only tester. The spacenavigator showed potential. Its axes are perfectly mapped one-to-one to the axes of the ROV. Up is up, rotation is rotation, and so forth. However, there were issues. Side-to-side and rotation were often mixed. When going up, the controller would also lift from the table, which was annoying. This is also due to my inexperience with the controller. It was hard to be accurate. Due to the results, the research on this controller was not continued.

Photo 3: The Mouse. The heading was controlled by the position of the cursor on the screen, thrust was controlled by the keyboard.

Photo 4: The NTNU MInerva ROV had a double joystick control which was tested. The setup was not tested any further during the thesis.

MOUSE

METHOD
The mouse was tested in the Unity prototypes. W on the keyboard was thrust, and the mouse controlled the heading. The ROV had to pitch to go up or down.

SUMMARY
I, the author, was the only tester. This control was easy to learn, but not realistic. Could not go directly upwards and downwards, and it was hard to rotate on the spot. The mouse control was not considered further due the constraints that would not enable the user to control the ROV realistically.

DOUBLE JOYSTICK

METHOD
The double joystick setup was tested in a real setting, controlling the NTNU Minerva ROV.

SUMMARY
The setup was tested by me, the author, a female student at NTNU, and a doctorate student at NTNU with experience in controlling ROVs.

The double joystick is used in the industry, though it has become less common than the single setup in new ROV systems. Both hands must be used to control the ROV. It is accurate in all axes, and it is relatively easy to control the axes separately. Takes a lot of space, and is usually a stationary option. The rotational axis is also normally mapped to a side to side movement, and not a rotation.

It was not tested further because it is a technical challenge to connect two joysticks to the computer simultaneously. Also, the PS3 controller is a portable two joystick controller in a way, and the fact that the single joystick system was also tested, the double joystick did not seem like it would provide any extra insight to the thesis.

Photo 5: The time trial. The user had to navigate through a course of rings using three different controllers.

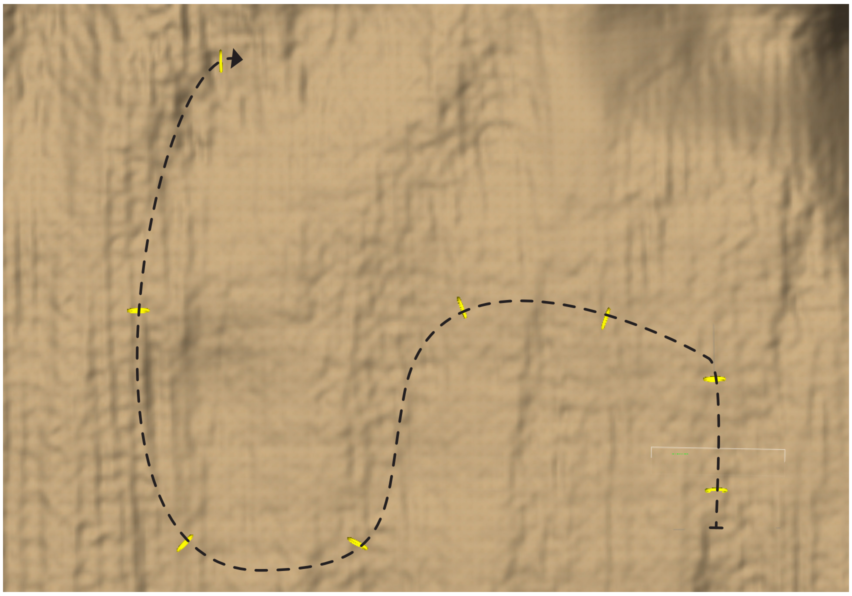


Figure 1: An overview of the ring course used in the time trial.

CONTROLLER TEST

HYPOTHESIS
Using a Playstation 3 controller is more effective to use than other controllers when navigating an ROV through a complex environment.

INTRODUCTION
Today, ROVs use large control units with advanced joysticks. A simple, flexible and wireless game controller might be a better fit for a virtual environment. However, there should not be a loss in performance. Therefore, I wanted to test several types of controllers to see how they perform compared to each other.

METHOD
The test will consist of two parts: a time trial and a precision test. The test environment is made in Unity, so the test is based on virtual performance. Visibility and precision is very high, and inertia is not simulated. The users will get approximately a minute before start to get acquainted to the controls.

> Time trial
Navigate the ROV through a track constrained by rings the ROV needs to go though at the best time possible. The basis for comparison is the final time. See Photo 5 and Figure 1.

> Precision
Control the ROV as close as possible to a line in space. The basis for comparison is the average distance from the line, also called the x-track error. Final time is also measured. If there is a strong correlation between a slow time and a short average distance from the line for the results, they are not valid because it only means that it is easier to be precise when you travel slowly. See Photo 6 and 7.

The tested controls are a keyboard, a PS3 controller and a single joystick setup.

Photo 6: In the precision test, the user had to stay as close to the line as possible.

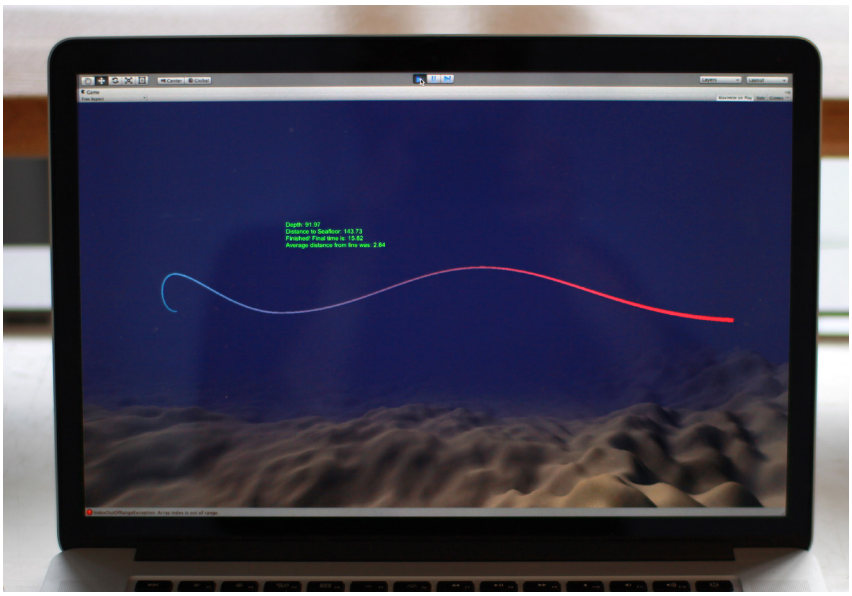


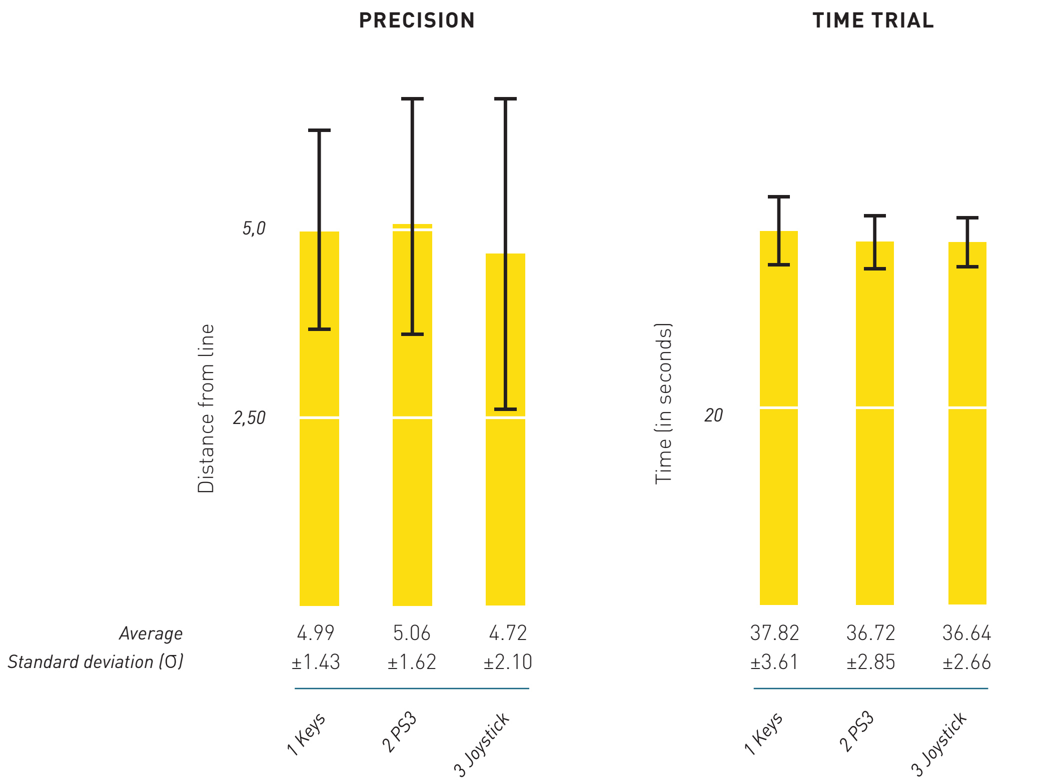
Photo 7: A sideview of the entire line. The design of the line challenged the user to move both up, down, left and right.

RESULTS

PARTICIPANTS
6 men, 1 woman, from 24 to 28 years old. All students in Trondheim, with different experience in gaming. The testers tried several times, from two to three times each.

The joystick was acquired at a later time in the process, and therefore has fewer test results. It was only tested by four of the men from the same group.

	KEYS	PS3	JOYSTICK
PRECISION			
Number of trials	12	12	12
Average time	16.29	16.38	17.52
Standard deviation (σ)	0.93	1.62	3.49
Average distance from line	4.99	5.06	4.72
Standard deviation (σ)	1.43	1.62	2.10
TIME TRIAL			
Number of trials	21	23	15
Average time	37.82	36.72	36.64
Standard deviation (σ)	3.61	2.85	2.66



**A graph for time spent in the Precision test is not made, as the distance from the line is the important variable tested in this condition.*

DISCUSSION

Here I will discuss both the quantitative and qualitative results from the controller test. At the end, I will present a short summary of the three controllers tested.

QUANTITATIVE
Originally, I wanted to perform hypothesis testing to see if there was any statistical significance to the results. As it turned out, the average results in both conditions for the three controllers were very similar, while their respective standard deviations were very large in comparison. It is obvious that there is no statistical basis in the results to say that the slight difference between the controllers are not a coincidence. The results are therefore not analyzed any further statistically.

The lack of differences may imply that the test was not properly designed to induce any differences. Both the time-trial and precision tests were relatively short, and probably not challenging enough, to differentiate between the controllers. The sample size was small, and the testers were mostly men in the same age group. A greater spread in the results may occur if others are tested. Additionally, the testers gave two to three results each, and this variable was not controlled by me.

QUALITATIVE
Users with no background from gaming had a hard time getting used to the keys. Trying to control eight buttons at once did not prove to be simple task. They learned quickly, and were not any worse using the keyboard than the other controls.

In the time-trial, I attached a trail renderer to the ROV, so that I could see the exact path the ROV had through the course. This made me realize that the keys gave a very jagged path. This is due to the fact that the output is either 1 or 0. Joysticks are analogue, and can give incremental thrust in all directions. The ability to make smoother paths is greater.

The single joystick is more accurate than the PS3 controller, but in this setting, the PS3 controller made smoother paths because the lower accuracy levels the curves. Users found the PS3 controller the easiest, but this is also the controller where most of them had any experience.

The joystick's accurate controlling is important when maneuvering close to dangerous structures on the seafloor. The long handle works very much like a lever, and it is easy to provide minimal changes to the input in any direction.

Some users commented the mapping of the axes to the joysticks. Users with experience from gaming, expressed that they would have had inverted the z-axis (up and down). This is common in flight simulators. The counter argument is that the ROV has thrusters that translate it directly upwards. Airplanes pitch backwards to go up, so it makes more sense to pull the joystick backwards as well.

FINAL CONCEPT

For the final concept, I can not recommend one controller over the other as they have very similar results, and the qualitative results show no clear indication in any direction. Some people favor the PS3, but there is a chance this is due to familiarity.

As no one stands out, the final concept should include the possibility to plug and play all three controllers. They are all cheap, and easy to get a hold of. Section 7.5 Navigation: Controllers will explain the details closer.

SUMMARY

On the following pages is a short summary of the pros and cons of each controller. The purpose is to get a better overview how the controllers can be used.

The pros and cons of each controller is presented, along with the skill level the controller is best for. The rating does not say how difficult it is to learn, but who will get the most out of the controller.

The different skill levels are:

> Novice
The novice user needs a simple, recognizable and easy to learn control system.

> Competent
The competent user more experience in the use of technology and controllers in general, and is a fast learner.

> Expert
The expert user is typically an ROV pilot, which means he is skilled in the use of general control technology and in the specifics of ROV maneuvering. They want more advanced and sensitive controls.

KEYBOARD

Available & cheap

ABOUT
The axes are mapped to the WASD and arrow keys. The user must use two hands to control all axes.

BEST FOR: Competent users

+ A keyboard is accessible for everyone, and known by everyone. It will be already used by the system in the first place. There are often several keyboards in a room at the same time, so it offers a good way for online cooperation. There are many other buttons available to control other functionalities.The buttons all have a distinct name or number. A mouse or trackpad is also normally available for other functionalities.

÷ The keys are either pressed or not; there is no incremental way to give thrust. Experienced users will use six fingers for the eight buttons, novice users only use one four buttons at a time, creating a very jagged route. It is almost impossible to use the keyboard with an HMD, because it is very difficult to find the right keys when the user cannot see the keyboard.

PLAYSTATION 3 CONTROLLER

Portable & compact

ABOUT
The PS3 controller uses bluetooth or USB-technology to connect to the computer. The controller has to be acquired separately from the system. Battery time is decent, up to 25 hours.

BEST FOR: Novice and competent users

+ The PS3 controller is wireless, small and light, making it the most portable controller. The controller has 14 buttons besides the two joysticks, and they are all reachable without removing the hands from the controller. This also makes it compatible with an HMD. The joysticks are analogue, so thrust can be adjusted incrementally. The controller is also very well known and needs little introduction. Buttons have recognizable names or icons.

÷ Controller has to be charged through a computer or Playstation. The association with Playstation might create an entry barrier: "I don't know how to play videogames." Joysticks are not accurate compared to a full-size joystick. Rotational axis has to be translated to a side-side control. Must use two hands to control the ROV. Not a realistic controller.

SINGLE JOYSTICK

Accurate & realistic

ABOUT
Joysticks can be bought from stores in most cities. Thrust upwards and downwards will be mapped to two buttons, an external lever or the thumb-sized joystick, depending on the preference of the user.

BEST FOR: Competent and expert users

+ The single joystick is used for real ROVs. It can be rotated, so it maps better with how the ROV behaves. Only one hand is needed to control the ROV, the other is free to do other things. It is very accurate due to the long "lever" design. Some joysticks have a small thumb-sized joystick on top, to pan cameras or to control up and down thrusters. Joysticks are ergonomic.

÷ It is not very portable, and has to stand on a flat surface to be used. Rarely wireless. Because the user controls several axes with the same control, it is hard not to rotate slightly while sliding from side to side, or the other way around. The joystick is not well known by everyone, and the button layout is not persistent amongst different joysticks.

NAVIGATION TEST

HYPOTHESIS

Using Oculus Rift will improve situation awareness compared to an ordinary screen and remove the need for an accompanying 2D map. It will reduce cognitive load.

INTRODUCTION

Today, the Surface Control Unit for ROVs have demands for mental rotations of separate images to correctly interpret where in a 3D world the ROV is. By immersing the user into the 3D world using the Oculus Rift, the hypothesis is that the awareness of where he is in the world will increase and the user can concentrate on the task at hand.

METHOD

The test will consist of three scenarios, as well as a 2 minute introduction to the controls to eliminate learning effects.

REAL	Simulation of real world navigation with ROV, with real conditions such as bad visibility. The user must navigate with a 2D map. The medium is a computer screen and a Playstation 3 controller. This is the "control" condition. See Photo 1 and 2.
SCREEN	The current concept, with very good visibility, marked targets, current target, a dynamic 2D map from above and an arrow pointing to the next target. The medium is one computer screen and a Playstation 3 controller. See Photo 3 and 4.
OCULUS	Same as number 2, only this time the user will wear the Oculus Rift. The 2D map is also removed. See Photo 5 and 6.

Each part will consist of a task where the subject needs to navigate to seven targets in a given order. The constellation of the 3D world is equal in all three scenarios. The test will measure time spent and distance traveled. The results from the tapping are not quantifiable.

The user will also perform a tapping test (Tracy and Albers, 2006), by tapping one foot continuously during the task. The theory is that the user will forget to tap when cognitive load increases, and make it for the observant easier to pinpoint flaws in the design.

4 HMD VS SCREEN

The next pages are dedicated to the research conducted on the Oculus Rift, and whether it provides an improved way of navigating an ROV.

On the left, the test method is introduced. The different test scenarios are shown in pictures on the next pages, followed by the results and a discussion.

A conclusion is made at the end of the discussion, which is brought into the final concept.

Photo 1: The Real condition. Visibility is bad, and the user has to navigate using the 2D map. The ROV is controlled by the PS3 controller.

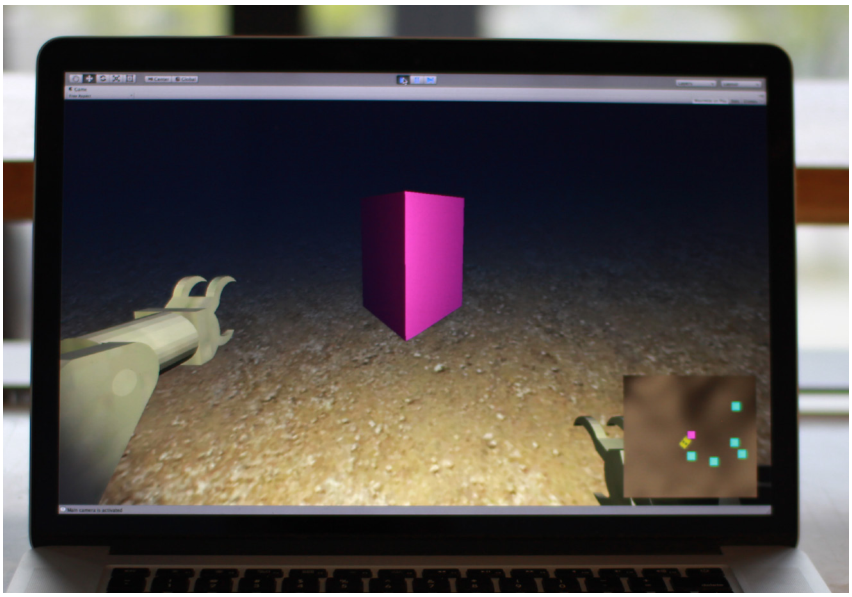


Photo 2: The Real condition. The user is approaching the target.

Photo 3: The Screen condition. Virtuality is exploited, with almost endless visibility and GUI to tell the user where the target is.

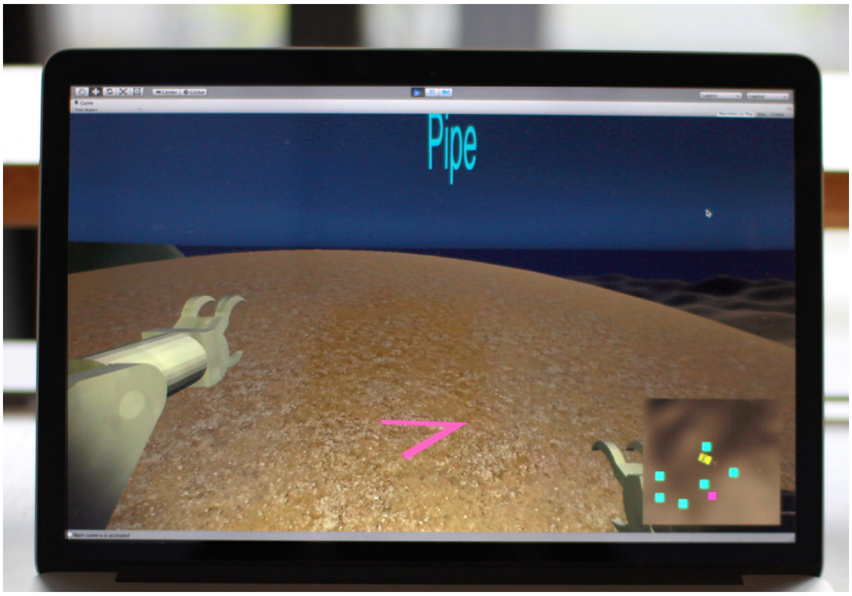


Photo 4: The Screen condition. The user can use the labels, follow the arrow, or look at the 2D map to find the next target

Photo 5: The Oculus condition. The user navigates using the Oculus Rift. The observer can see the progress on a computer screen.

Photo 6: The Oculus condition. The user can turn his head to find labels or he can rely on the arrow to find the next target. The arrow and the arms indicate the forward direction for the user.

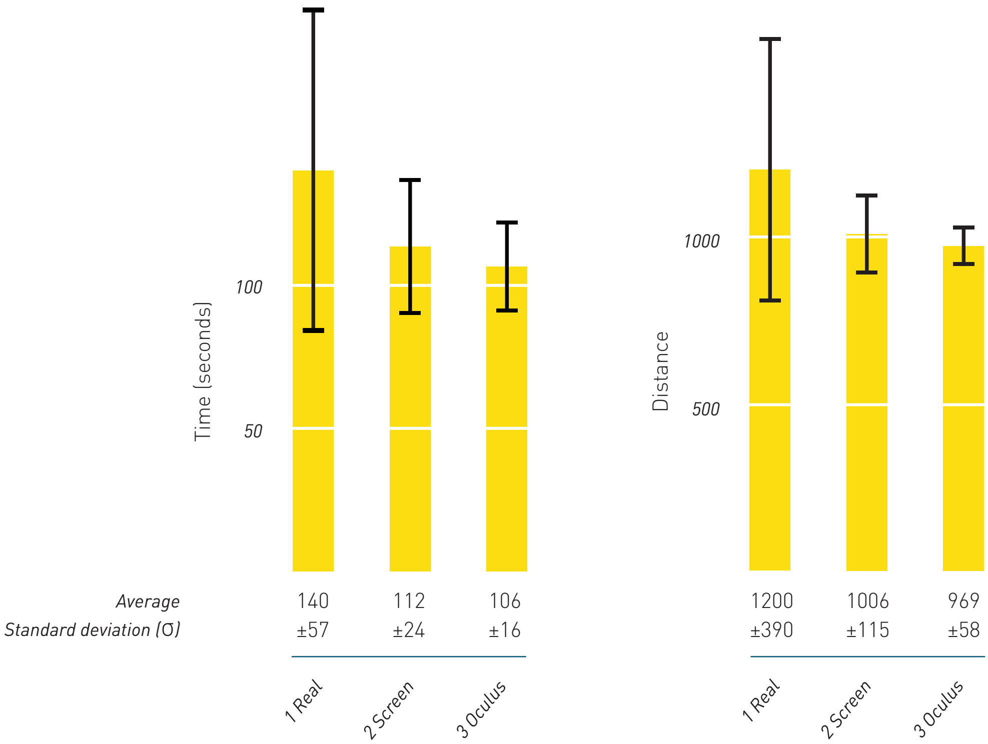
RESULTS

PARTICIPANTS

10 men and 5 women. Average age was 25.1, ranging between 23 and 28. Students in Trondheim or professional computer engineers. Four tests are not included in the quantitative results due to a modification of the test that removed the grounds for comparison. Their results are included in the qualitative analysis.

		VIRTUAL	
	REAL	**SCREEN**	**OCULUS**
Average time (seconds)	140*	112	106
Standard deviation (σ)	57	24	16
Average distance	1200	1006	969
Standard deviation (σ)	390	115	58
RATIO TIME			
Reality	-	0.78	0.74
Screen	1.27	-	0.95
Oculus	1.25	1.06	-
RATIO DISTANCE			
Reality	-	0.82	0.80
Screen	1.21	-	0.97
Oculus	1.26	1.03	-

**Speed for this scenario was 1/2. Average time is divided by half for comparison.*



HYPOTHESIS TEST
The original hypothesis is that the Oculus condition will give better results than both the Real and Screen conditions. An hypothesis test has been conducted to see if the results are just based on pure coincidence, or if they actually are statistical significant.

A null hypothesis is created, stating there is no difference between the Oculus condition and either the Real condition or the Screen condition. The goal is to find the probability that the null hypothesis is still true with my results. Values from the results are put in the formula, and the probability is calculated. If the probability is low, it indicates that the counter hypothesis is true.

It is assumed that the results follow the normal distribution.

HYPOTHESES

Null hypothesis states there is no difference between navigating in the Real condition versus the Screen condition.

$H_0: \mu_1 - \mu_2 = 0$

$H_1: \mu_1 - \mu_2 \neq 0$

Null hypothesis states there is no difference between navigating in the Real condition versus the Oculus condition.

$H_0: \mu_1 - \mu_3 = 0$

$H_1: \mu_1 - \mu_3 \neq 0$

Null hypothesis states there is no difference between navigating in the Screen condition versus the Oculus condition.

$H_0: \mu_2 - \mu_3 = 0$

$H_1: \mu_2 - \mu_3 \neq 0$

FORMULAS

$$z = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}}$$

$$z = \frac{\bar{x}_1 - \bar{x}_3}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_3^2}{n_3}}}$$

$$z = \frac{\bar{x}_2 - \bar{x}_3}{\sqrt{\frac{\sigma_2^2}{n_2} + \frac{\sigma_3^2}{n_3}}}$$

RESULTS, TIME

$z = 1.49$

$P(H_0$ is true$)$ = 13.8%*

$z = 1.88$

$P(H_0$ is true$)$ = 6.2%*

$z = 0.66$

$P(H_0$ is true$)$ = 50.9%*

RESULTS , DISTANCE

$z = 1.59$

$P(H_0$ is true$)$ = 11.2%*

$z = 1.95$

$P(H_0$ is true$)$ = 5.1%*

$z = 0.95$

$P(H_0$ is true$)$ = 34.2%*

** The probabilities are calculated using the "Tabeller og Formler i Statistikk" (Kvaløy and Tjelmeland, 2011)*

DISCUSSION

Here, I will discuss the research and results from user testing the Oculus Rift versus a regular computer screen. I will also present what I have chosen to use go forward with for the final concept.

QUANTITATIVE

Prior to the testing, no significance level was selected. The most common level is P (H_0 is true) ⩽ 5% for the results to be statistically significant. In the context of this thesis, the consequences for drawing a false conclusion are trivial. One can therefore argue that a significance level of P (H_0 is true) ⩽ 10% is sufficient.

With P (H_0 is true) ⩽ 10%, it is probable that the results showing that Oculus is better than Real was not a coincidence. For Real compared to Screen, the results are not within the significance level, but they give an indication.

They also indicate a small improvement with Oculus compared to Screen, but these results are nowhere near statistically significant, and the probability is high that the results were a coincidence.

One can therefore argue if it is reasonable to say that the difference between Real and Oculus is significant, while the difference between Real and Screen is not significant, when the difference between Oculus and Screen is not significant at all.

The participants are selected from a convenience sample, and they do not represent the users presented in this thesis. Participants had a varying level of skills, and carry-over effects were present. The sample size is small, with n=11, and can not be trusted to be large enough to erase those carry-over effects, despite the use of a within-group design.

I will be careful to draw general conclusions from the quantitative results solely.

QUALITATIVE

Qualitative results show a different picture: users found the Oculus Rift to be an improvement. They said it was easier to navigate because you could look around. It was also easier to see depth and sizes.

The users mostly said that they thought the run with the Oculus was the fastest and most accurate, and in many cases they were right. However, the difference was not large in terms of seconds, so I suspect the positive association with the Oculus only made it feel faster. That people thought the run on a normal screen was longer, can mean that it probably required more mental work, which makes perceived time longer (Weinschenk, 2011).

When asked if the users used the arrow

or map to navigate, most participants who started with Real, where there only was a map, also used the map at the beginning of Virtual. Many switched over when they noticed what the arrow did. The users who started with the Oculus, with only the arrow, continued using the arrow for Virtual, without any exceptions. Some users actually claimed they used the map in Oculus, when there was no map present.

A few users, including myself, experienced nausea when looking around while the ROV was moving. It passed after a little while, except for one user who still experiences nausea with the Oculus Rift.

TAPPING TEST

The tapping test was conducted to discover where users would find the test challenging, and to see if anyone would experience cognitive overload and a small useful field of view.

The users had most trouble when approaching a target, and trying to find the next one. They were focusing both on hitting the target, while already starting to think about where the next one might be.

OCULUS RIFT

There are benefits to using the Oculus Rift. The question is if they outweigh the disadvantages. It is big and bulky, has long wires and takes time to set up. The resolution is not super, and some users, including myself, experienced nausea. The user is restricted to his seat, and can not see anything else beyond the screen. It does not pair well with a normal computer.

Qualitative analysis showed that users preferred the Oculus over a traditional computer screen. The quantitative data indicate a very small difference, but the data set is not statistically significant.
I want to remind the reader that these users were tested on how well they could navigate a 3D world, and no more than that. If put into the greater context of planning an ROV mission, it is my belief that the Oculus falls short. There is too much hassle for very little gain. The users where perfectly able to navigate the world without the Oculus. Other criteria, such as availability and cooperation, trumps any miniscule benefit of effectiveness.

Both quantitative and qualitative results indicate that the two virtual conditions are more effective than a simulated real one. The screen will therefore be also be an improvement.

It is also my belief that there are other solutions to improve the situation awareness given by the computer screen, while still remaining its inherent benefits over the Oculus.

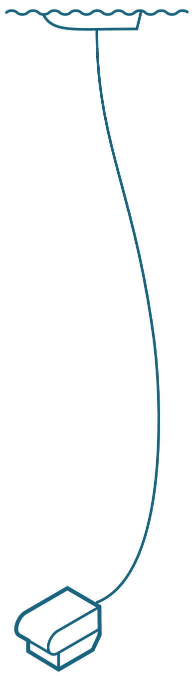


Figure 1: It should be possible to simulate the tether realistically.

5 ON IMMERSION

In section 4.2 Reality vs. Virtuality, I wondered if there was a negative correlation between immersion and effectiveness regarding the task at hand. Through the qualitative user testing, I found this to be true in many cases. In a virtual world, the elements that impose a challenge to the user can be removed, creating a less real, yet more effective interaction. The following section discusses three of these elements, and suggests solutions.

POSITIVE BUOYANCY

The positive buoyancy of the ROV requires attention from the pilot because he constantly has to counter the upwards force by going down. As a novice user, this significantly decreased my attention span when I tried the NTNU Minerva ROV, especially when I had no references to look at and suddenly noticed I was 10 metres higher up than I thought. The primary and tertiary users, on the other hand, are used to this, so I was curious if adding a perfectly neutral buoyancy in the virtual world would actually be more demanding cognitively.

Unfortunately, I was not able to test my theory on any real ROV pilots. One reason was the lack of access to ROV pilots (see Reflection). Another reason is that it is hard to measure cognitive load. However, the Oceaneering simulator used to train ROV pilots does not necessarily include the positive buoyancy, and all pilots are trained using the simulator.

In addition, it is reasonable to assume that different ROVs have different buoyancies and are controlled in different conditions. Therefore, the pilot's controlling is likely more linked to the feedback from the motion on the screen, rather than automatically countering the buoyancy. Feedback is instant, so it should not pose a problem for the expert user to adapt.

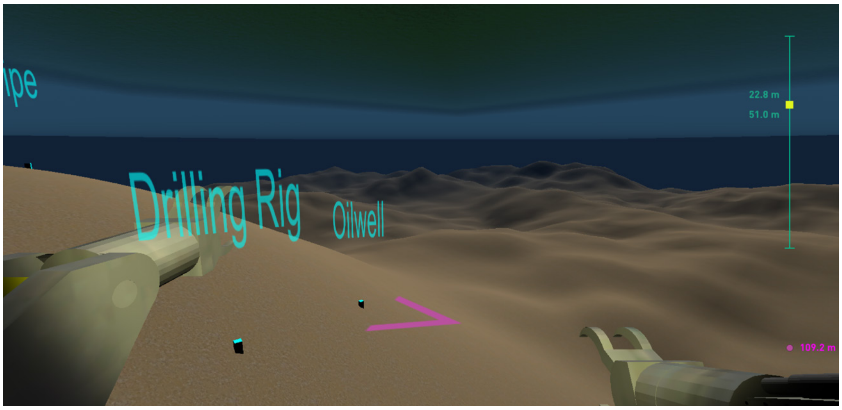


Photo 1: The manipulators cover 1/4 of the sight of the important bottom half of the screen.

> Conclusion
With these points in mind, I have concluded that the buoyancy can be disregarded from the concept. It will simplify the concept, and there is no reason to believe an ROV pilot will find it more challenging to control the simplified version.

MANIPULATORS
The scope of the thesis does not include any simulation of the actual interaction with the installations on the oil and gas field. The interesting thing about the manipulators in conjunction with navigation, is that they are always visible on the screen. They provide a set point for the pilot. If the camera is panned, the pilot can use the manipulators to know which way is forward.

During testing, when the users were in a first person view of the ROV and had to navigate through rings, they felt the rings were much bigger than they were relative to the actual ROV. The manipulators give an indication of the size of the ROV.

In the functional prototype with manipulators, some users would complain that they were in the way. A measurement showed that they actually covered 12.2% of the entire screen, or almost ¼ of the lower half of the screen (photo 1). The last number is significant due to the fact that most visual reference points are below the ROV and as a result appear in the lower half of the screen.

> Conclusion
Provide the user with a reference point to indicate the forward direction and the size of the ROV without blocking view. The user should still feel he is controlling an ROV.

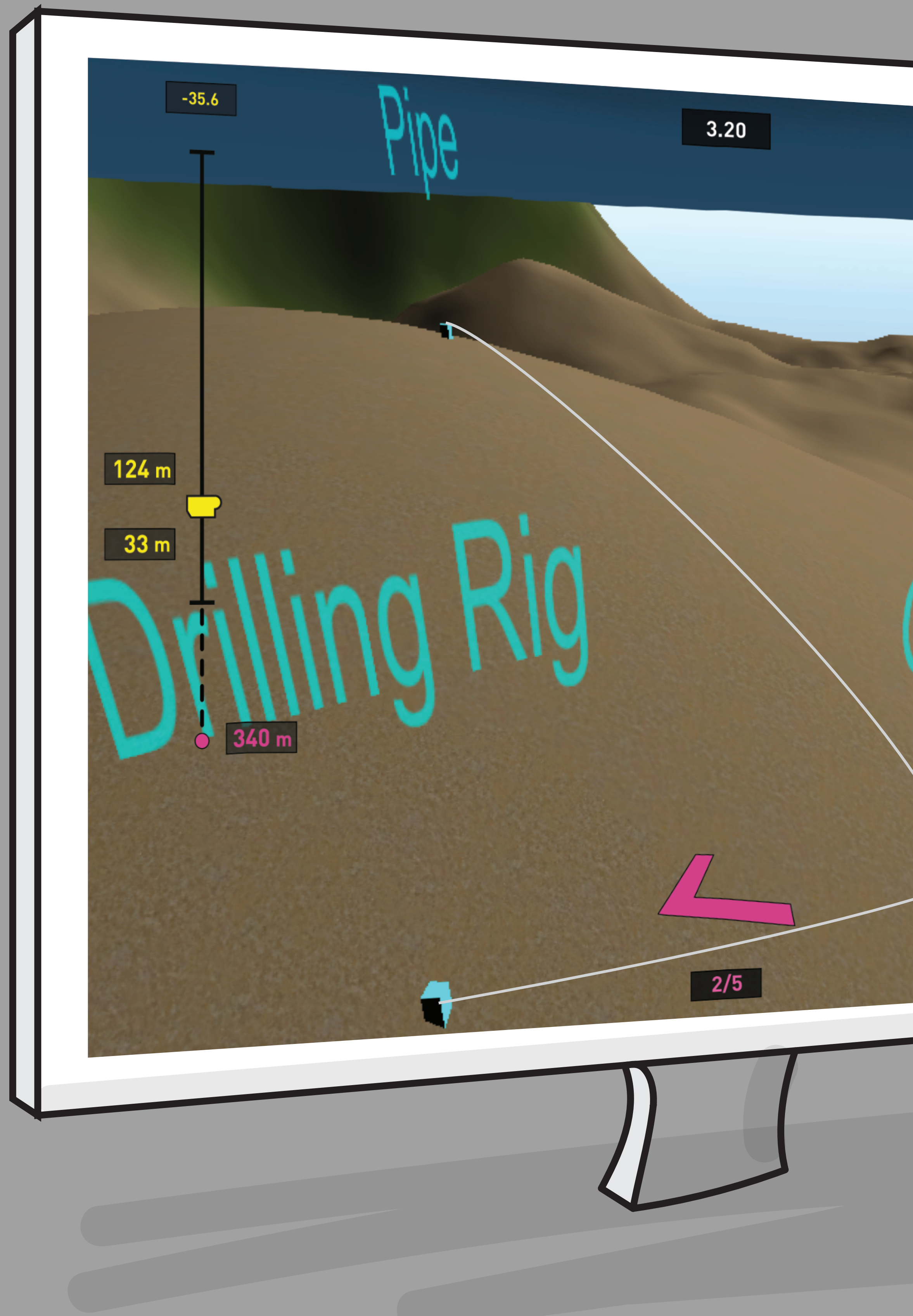
TETHER
According to the research, safety comes first when planning the ROV and the tether is a complicating matter. When several ROVs are in action, the mission needs to be organized and synchronized. The concept of planning the trajectory of an ROV without taking into account the tether does not make much sense. The trajectory would never be realistic, and would in a best case scenario only function as a propositional guide. Worst case, it could cause a misalignment between the different vessels and ROVs, and cause dangerous situations.

ROV simulators today, such as the one visited at Oceaneering, can simulate the tether, and add currents for an even more realistic approach. Fugro Oceanor, a spin-off from Sintef, have measured ocean currents in the Norwegian seas for over 25 years, and they also sell equipment to oil and gas companies to perform such measurements themselves. They have several mathematical models to predict ocean currents. These data are already used as design data for marine constructions. In addition, tidal currents and deep ocean currents are much more predictable compared to wind currents (Minesto AB, 2015).

> Conclusion
This is a scenario where the virtual should be as real as possible, because the final product loses value the further away from reality it is. Luckily, it should be possible, in my opinion, to embed a realistic model of the ocean currents within the 3D model of an oil and gas site. With such a model in place, the behavior of the tether will also be realistic.

7 FINAL CONCEPT

>	Introduction	145
1	Concept in Context	147
2	Concept in Detail	165
3	Navigation: Point of View	177
4	Navigation: Controllers	181
5	Navigation in Detail	183



-35.6
Pipe
3.20
124 m
33 m
Drilling Rig
340 m
2/5

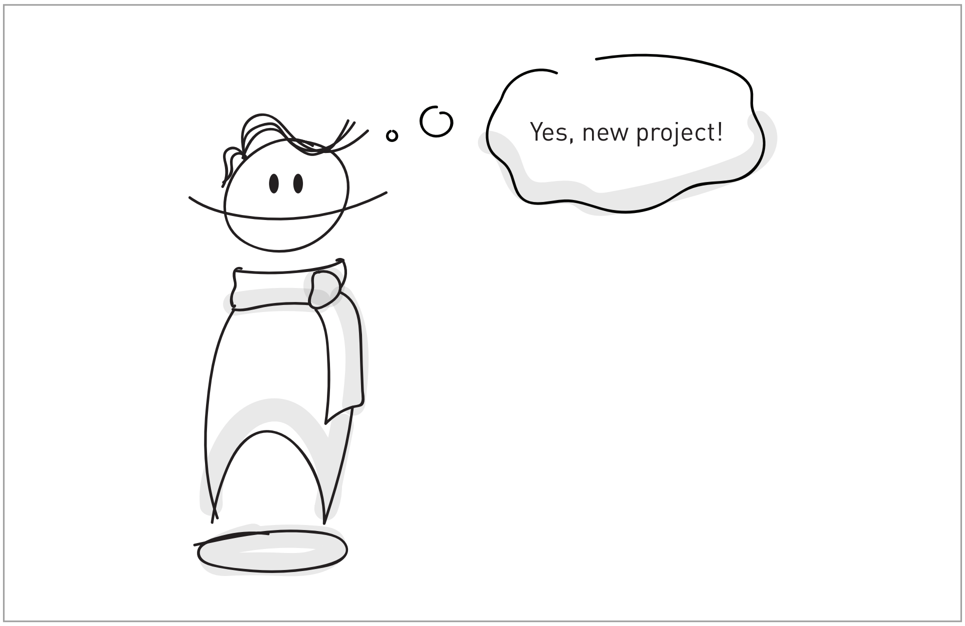
> INTRODUCTION

The title for the thesis is Interaction Design in a 4D Sea Map. Naturally, this has been my main focus. On the other hand, I have also felt the need to create a context for the research. Why do you need an interactive environment to plan a mission? Where does it fit in? Why will the industry want to use this concept?

During research in general, and the user research specifically, I found a moderate excitement in the industry for the prospect of virtually navigating the subsea world as a part of planning an ROV mission. I had to be able to explain where it would all fit in.

During the next chapter, I will present my answer. The concept for the entire system is presented in context first, before the details of the same system are presented in the following section. After that, the final concepts for the viewpoint and the controller while navigating in the 3D world is presented, with a section on the details of the navigation at the end.

1



Carl is excited to start a new project. Step 1 in the concept is the same as in the user scenario.

1 CONCEPT IN CONTEXT

INTRODUCTION

The concept will be presented in three stages, as the system will be in the hands of three users with different needs and tasks. As the system spans the entire planning of an ROV mission, the "primary" user changes. The system will therefore be presented on a fairly chronological timeline, through the eyes of the dominant user at the time.

I have added illustrations from 5.2 User Scenario to help link the concept to a real scenario. These are presented as "Current".

As in 5.2 User Scenario, the secondary user is presented first, then the primary, with the tertiary at as the last one.

Secondary User

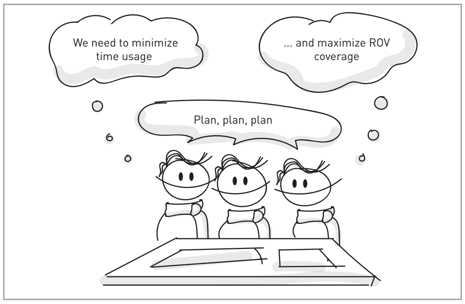
Carl Ditlefsen, Project Manager
Statoil

1. AN ROV MISSION IS NEEDED

Carl starts his day at the office drinking his daily coffee. In a morning meeting with other colleagues, it is decided that Statoil needs to connect cables between different templates on the seafloor at the Ormen Lange gas field outside of Kristiansund. Carl, a Project Manager with experience in ROV operations, is the natural choice for planning and leading the mission on behalf of Statoil.

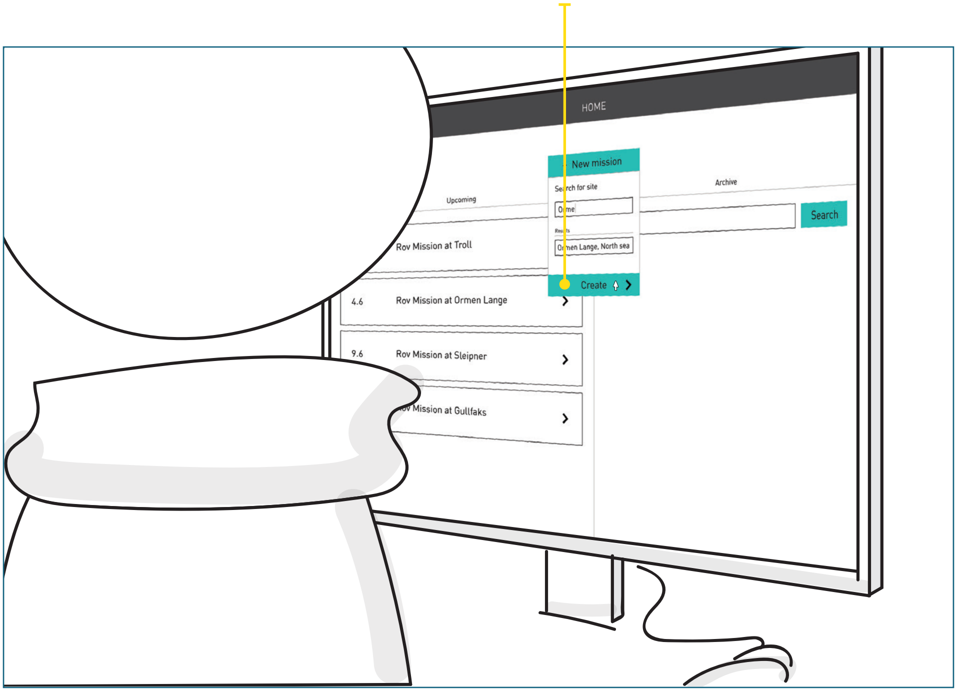
Current

Steps 2, 3, 4 and 5 all happen during the preliminary planning by Statoil.



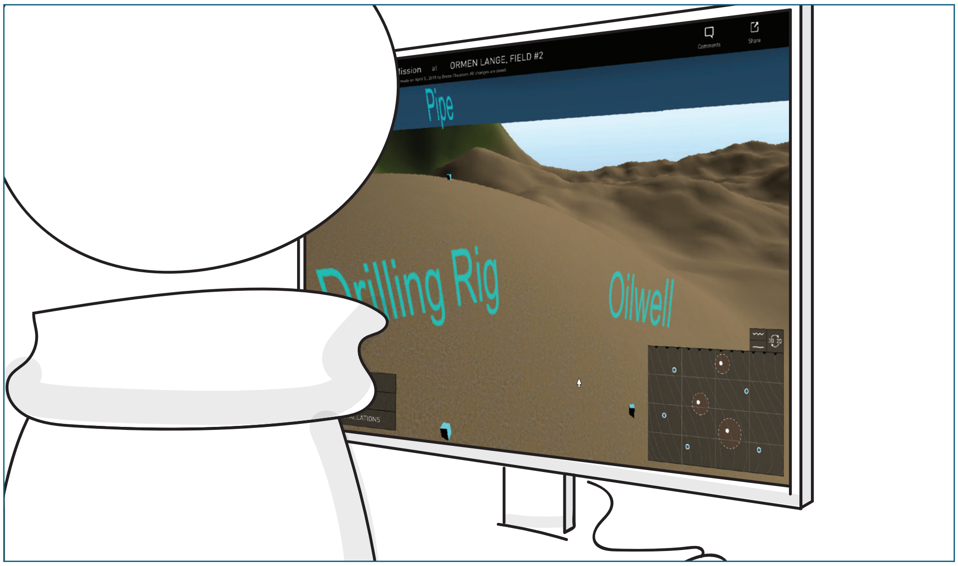
Create

2

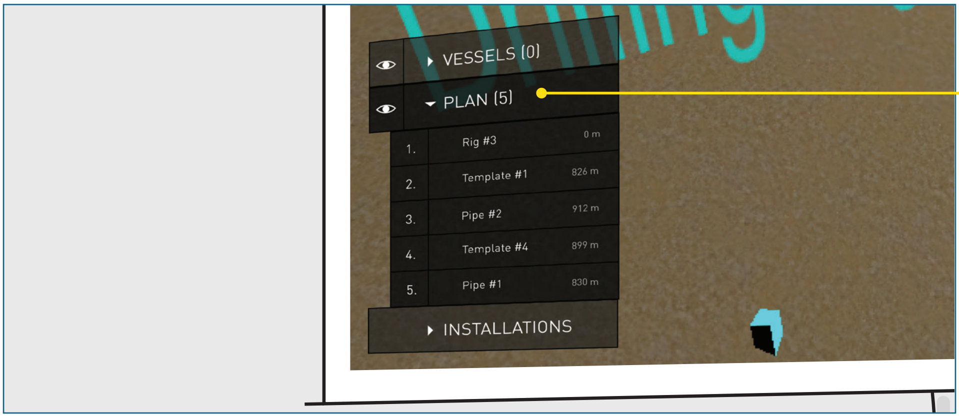


2. CREATE NEW MISSION

After the meeting, Carl sits down at his computer and opens the ROV Mission Planning application on his computer. After opening the application, Carl sees the main menu, displaying other, upcoming missions where he participated in the planning. However, Carl wants to plan a new mission, so he clicks on NEW MISSION, and is prompted to select a site for the mission. He selects the Ormen Lange field and clicks CREATE (2).



3



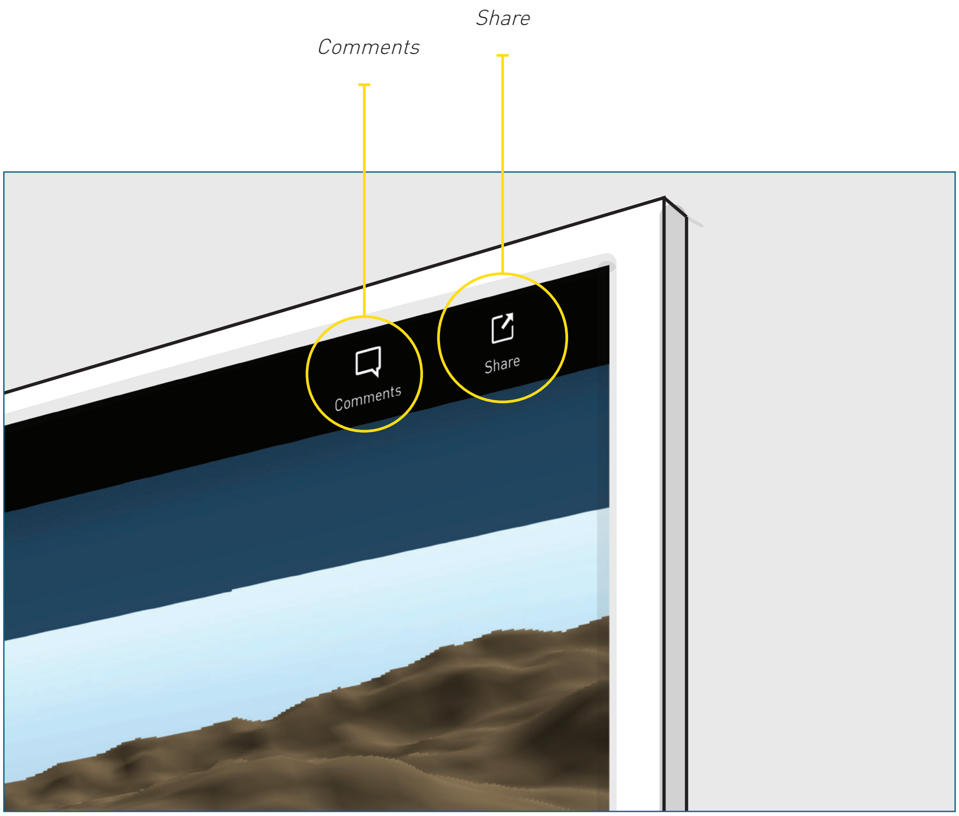
4

Plan

3. CREATE PLAN

Following the creation of the ROV mission at Ormen Lange, Carl now has an overview of the site as both a virtual 3D model and a 2D map (3). All installations are modelled in, as well as safety zones around the rigs. He already discussed some of the details regarding the project with his colleagues at the morning meeting, so he knows which installations on the seafloor where they need to do work. Carl navigates in the 3D world and adds the relevant installations to the plan in the order he thinks the mission could be executed in. They appear on the left side, under PLAN (4), in the order Carl clicked them.

5



Step 4. Carl clicks share, copies the link that appears and sends it to his colleagues.
Step 5. Everyone with access can comment on the project.

4. SHARE WITH COLLEAGUES

Carl is a bit unsure of the technical details on what exactly needs to be done on the installations. He decides to share the mission he created with his colleagues. He clicks SHARE (5), and copies the link that appears. He emails it to relevant colleagues.

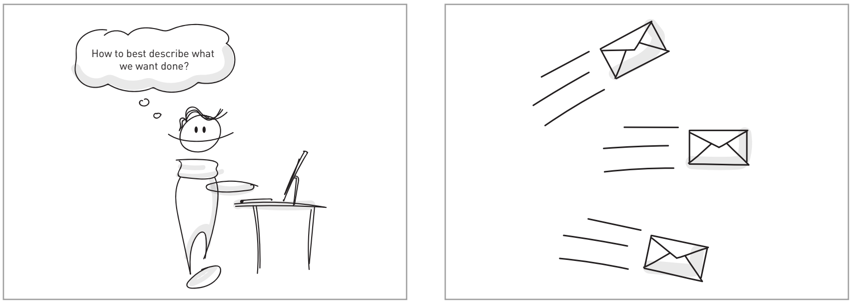
6

5. DISCUSS

As his colleagues get access to the system, they can see everything Carl has done already. Carl posts a comment (5), where he asks for help concerning the technical details of the work Statoil needs help with on the installations. Many replies later, they agree on the exact problems they need help with. Carl adds the instructions to the relevant installations by clicking on them and adding them under notes (6).

Current

Relates to step 6.



7

Copy link

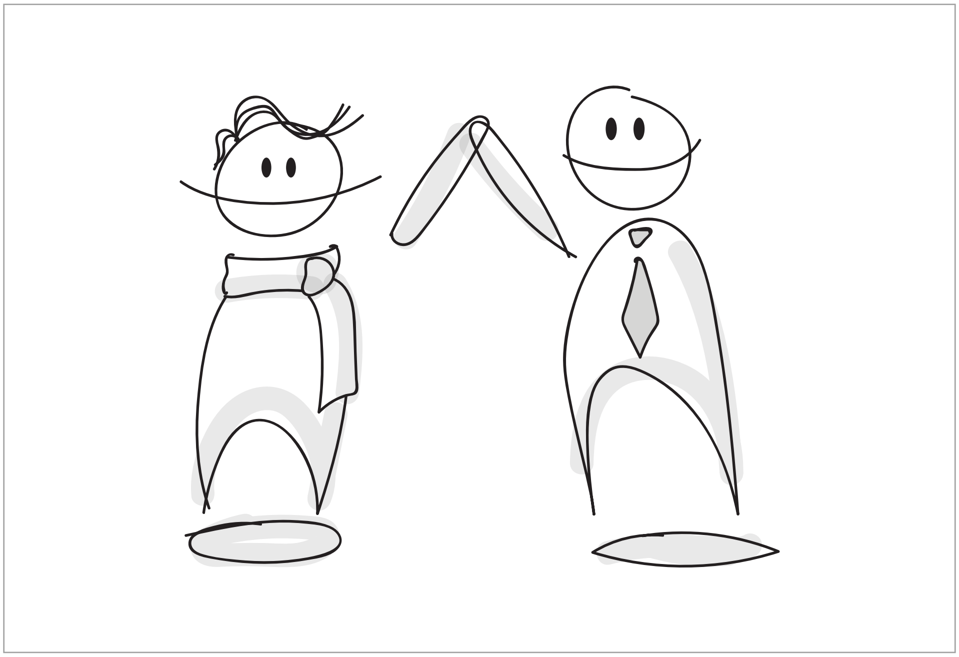
Download PDF

Step 6. Carl clicks "Download PDF" sends it out to relevant ROV firms.
Step 7. After agreeing on terms with Oceaneering, Carl copies the link and sends it to Henning.

6. TENDER THE JOB

As all the problems are defined within the system, it is time for Carl to hire a company specializing in ROV operations. Carl clicks on SHARE, and downloads the plan as a PDF. He then sends it out as an email to relevant firms.

8



Step 7. Carl chose to cooperate with Henning. Yes!

7. ACCEPT OFFER

Carl receives offers from many ROV firms, where they explain what they are able to do and give an estimate of costs. After careful consideration, and discussion with colleagues, Carl chooses to cooperate with Oceaneering. The contract details are briefly discussed and agreed upon between the two firms. When the deal is done, Carl enters the ROV planning system, and clicks on the relevant mission in the HOME menu. He then proceeds to click SHARE, copies the link, and emails it to the Henning, the ROV supervisor at Oceaneering.

8. FOLLOW UP

While Oceaneering, experts in ROV operations, continues to work within the system to improve and develop the plan, Carl pays attention to the work, offers help and clarifies the problems they need solved. When it is time for execution, Carl is on the boat to follow the operation.

9

Step 1. Henning recieves the job proposition from Statoil and decides to make an offer. This step is the same as in the user scenario.

10

Step 2. Henning opens the link he receives from Carl, and enters the project space.

Primary User

Henning Gran, ROV supervisor
Oceaneering

1. SEND OFFER

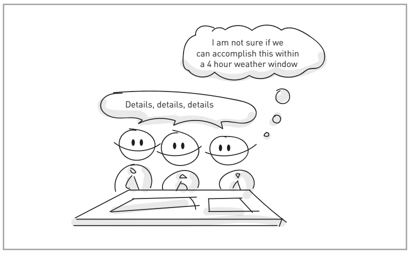
Henning is at his office when he receives the job proposition sent out by Carl at Statoil. "This is a job we are perfectly equipped to handle," he thinks to himself as he scans the pdf (9). During the next few days, he creates an offer with Word, describing what type of vessels and ROVs Oceaneering can provide for the mission, how they are planning to use them, at what price and within an approximate time frame. He emails it back as a PDF.

2. GET ACCESS

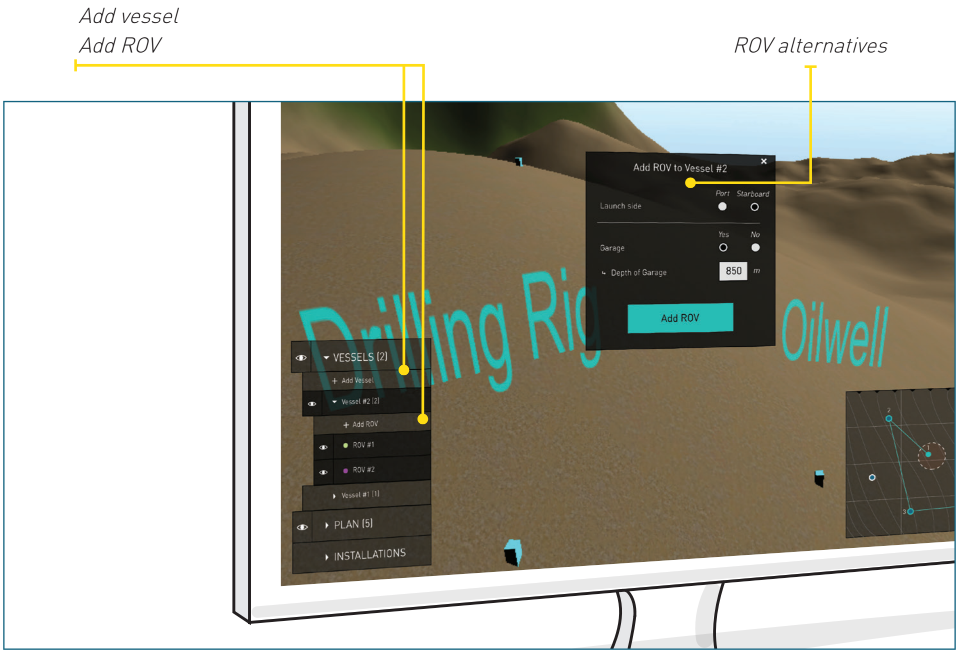
After a brief contract negotiation, Oceaneering gets the job. Henning receives an email with a link to the project space in the ROV planning system. Henning is familiar with the program from previous projects. He clicks the link, the system opens on his computer, and reveals the project Carl has created. "Let's get to work," he thinks (10).

Current

Planning the details is covered by sted 3, 4, 5, 6, 7 and 8.



11



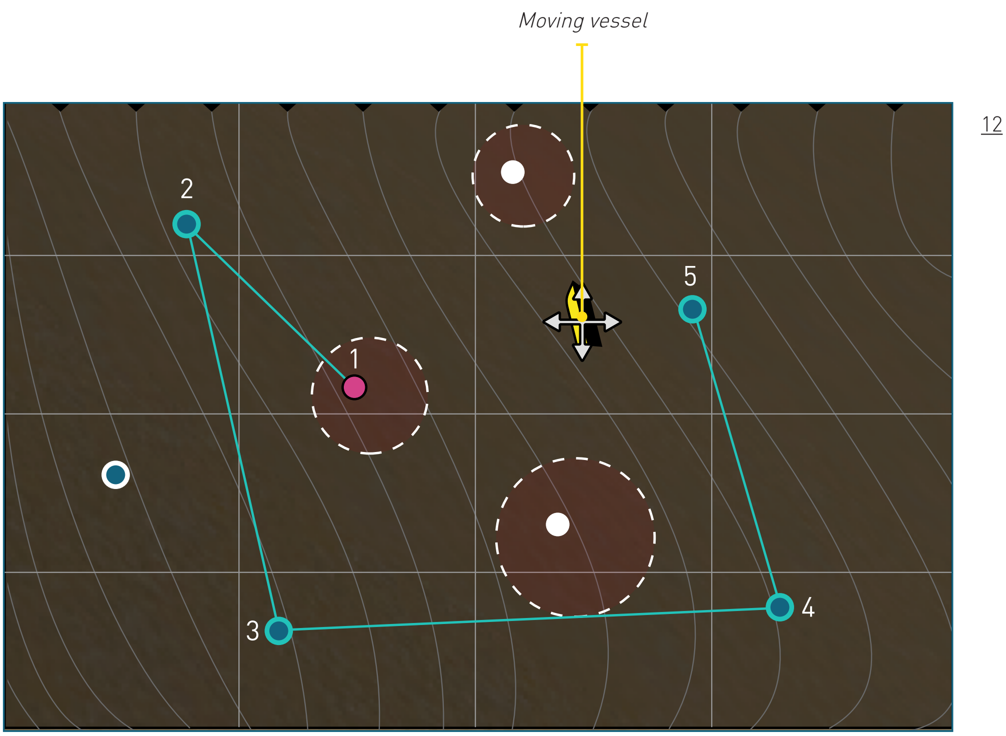
Step 3. Henning adds Vessels to the list, and ROVs under the vessels again. When adding an ROV, Henning has to specify some variables before they are added.

3. ADD VESSELS AND ROVS

Henning studies the safety regulations for the site and the current plan for the mission. He adds two vessels to under the VESSELS layer. They are automatically placed in the 3D world, at a close, yet safe, distance from the rig and from each other. One of the vessels Henning plans to use in the mission have two work class ROVs, while the other has one. He clicks on the vessels and adds ROVs accordingly. As he adds an ROV, he must choose which side of the vessel the ROV is deployed from and if the ROV is in a garage. If the latter is true, he must define the depth where the ROV will leave the garage (11).

4. MOVE VESSEL

Henning selects one of them in the menu on the left. The 3D view centers on the vessel, and reveals an option to move the vessel. As the vessel is on the surface of the ocean, and can only move in 2D, Henning switches the main view from 3D to 2D. He then proceeds to move the vessel freely in 2D to find a position which fits the procedure of events in the plan (12). When finished, he changes back to having the 3D view in the main window again.

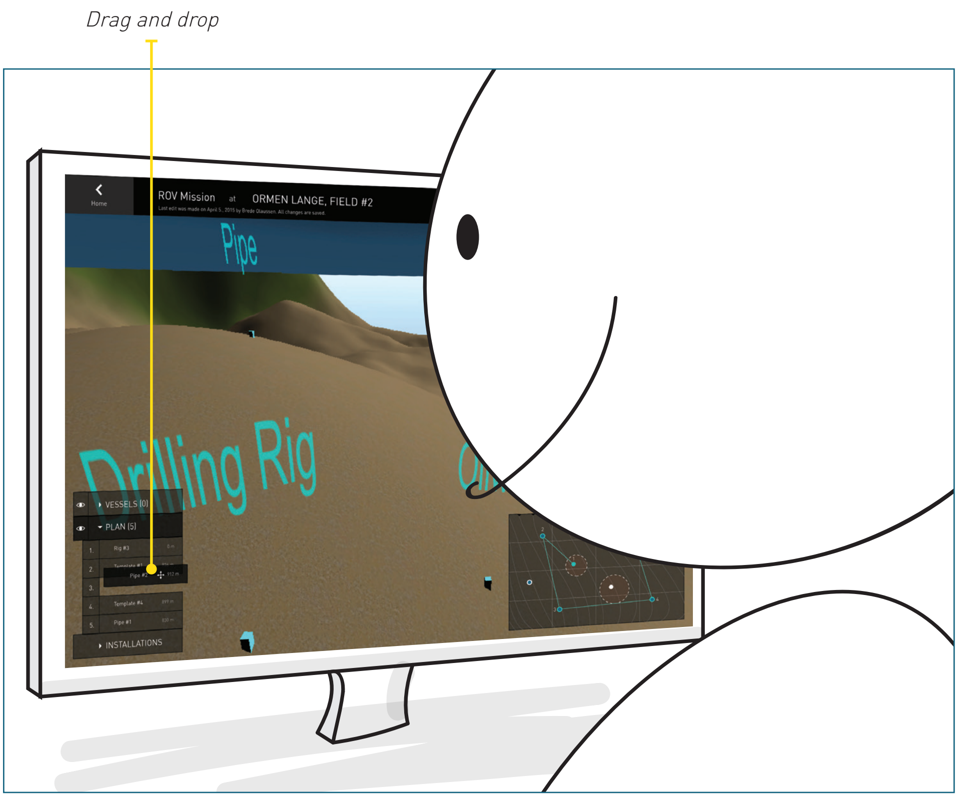


Step 4. Henning can move the position of the vessel on the surface in the 2D map view.

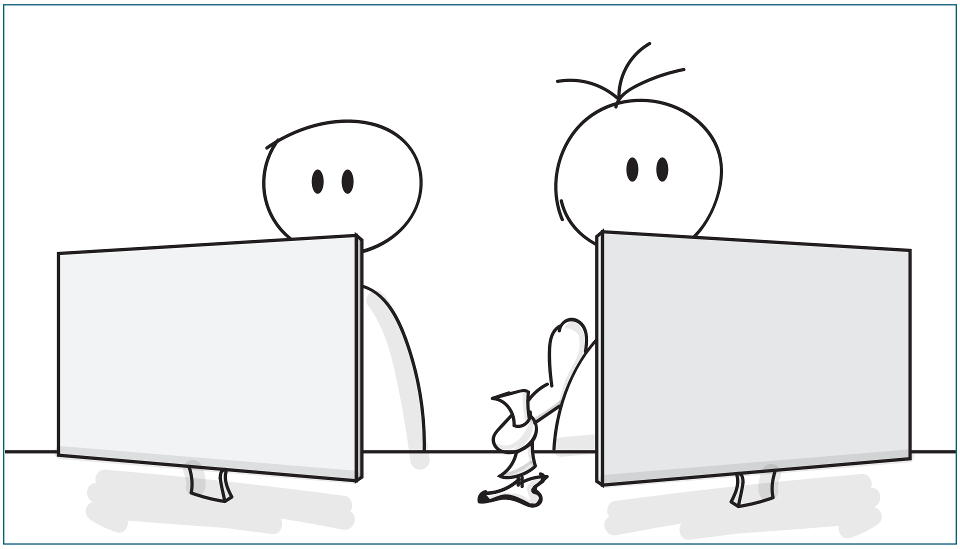
5. CHANGE PLAN

After placing the vessels and their ROVs in the 3D world, Henning realizes that the original chain of events planned by Carl is not the best way to perform the operation. The vessels have strict safety regulations for how they need to be positioned in the water to avoid a "blow-on" situation. This means that the vessel can not drift into any rigs in case of failure. He opens the PLAN layer, and uses drag and drop (13) to change the execution order of the plan. He sees that the lines between the relevant installations in the map change according to the order in the list.

13



Step 5. Henning changes the order of the Plan by using a drag and drop functionality.



14

Step 6. Henning bring in Brede, the ROV pilot, so they can cooperate when creating trajectories for the ROVs in the system. Henning prefers to use the PS3 controller, Brede likes to use a joystick because it is more realistic.

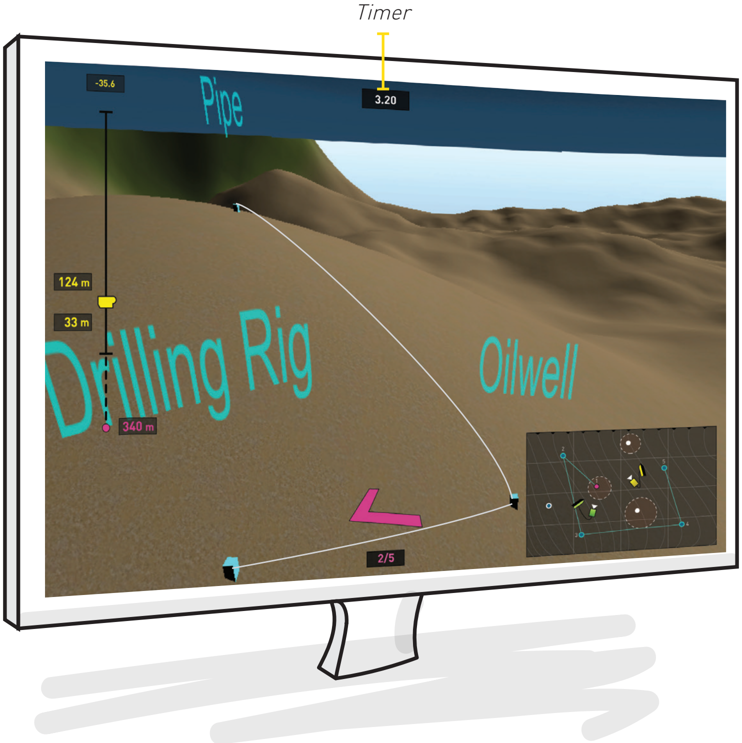
6. CREATE TRAJECTORIES

Henning wants to plan how the ROVs should move according to each other during the course of the mission. He could plan the trajectories of the ROVs one-by-one, but he finds it easier to plan them simultaneously, so he calls in Brede, an ROV pilot at Oceaneering, to a meeting room to help him. Henning shares the project with Brede, who opens it on his own computer. When he sees in the top right corner that Brede is connected, he clicks on an ROV, the 3D view zooms in on it, and he clicks on the appearing option CREATE TRAJECTORY.

Many of the graphical elements slide to the left, new ones slide in from the right, and the view is now from "inside" the chosen ROV. A box appears on top with two choices: either Henning can record a trajectory alone, or he can cooperate with Brede. Henning chooses to cooperate. Brede gets a message on his computer, saying Henning wants to cooperate (14). Brede selects an ROV from the appearing menu and clicks JOIN. His view also changes the same way, and he is now "inside" the ROV he chose. A message appears saying "Move to start".

As Henning makes the first move, the message disappears and his ROV moves according to his the controls. A timer appears on the top, linking the actions to a point in time relative from the start.

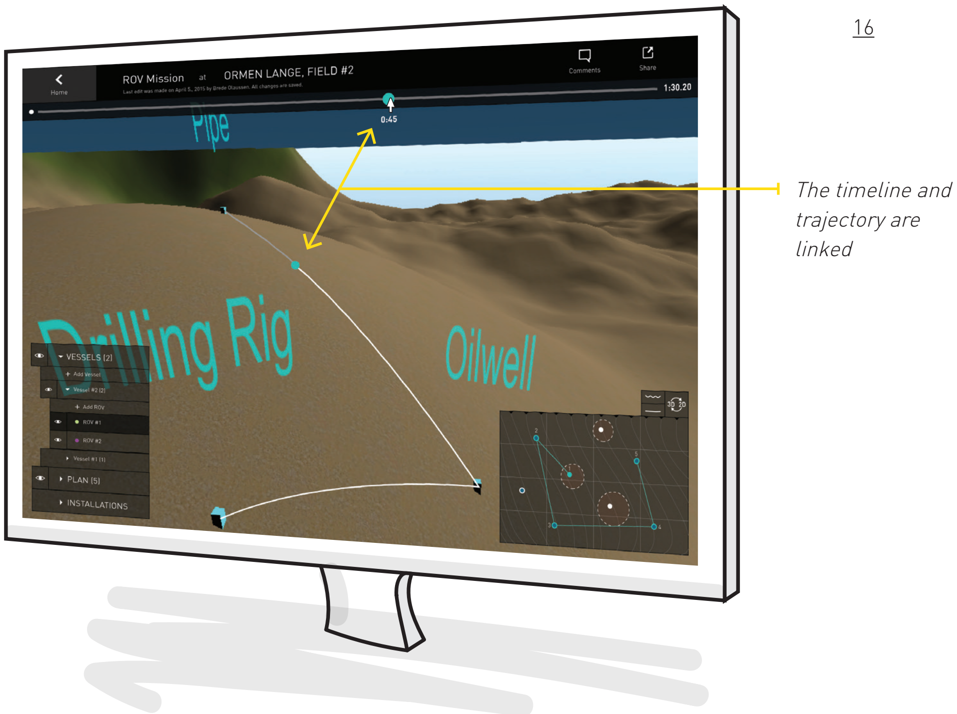
15



Step 7. Navigation mode. Henning and Brede navigate in the 3D world, following the plan. A timer is added at the top, representing the 4th dimension of the sea map.

7. NAVIGATE

While navigating, Henning and Brede are aided by a specialized interface to help them create the best trajectories for the mission (15). This part of the scenario is explained more thoroughly in section 7.5 Navigation in Detail. When finished, the trajectories are saved in the 3D world as a layer underneath its corresponding ROV.



Step 8. Back in the project space, when Henning drags the time slider, the recorded trajectories change accordingly. Trajectories that occur after the time on the timeline are faded.

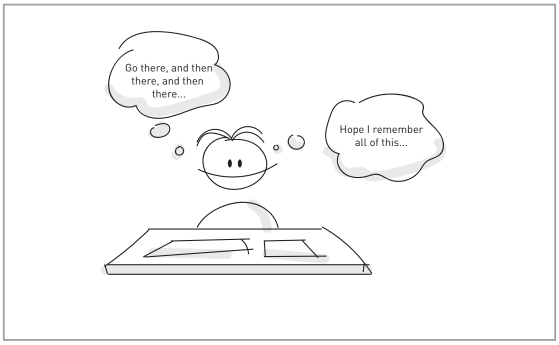
8. CHANGE TRAJECTORY

After the first run with Brede, Henning is not completely satisfied with how one of the ROVs were controlled during the second part. He decided to change its trajectory. He drags the button on the timeline to backwards, and stops when he feels the trajectory starts to go wrong (16). He then clicks on the relevant ROV, and chooses EDIT TRAJECTORY. He is prompted to choose between starting from the point on the timeline or to start from point zero. He chooses the former option, and the view changes to the view of the ROV.

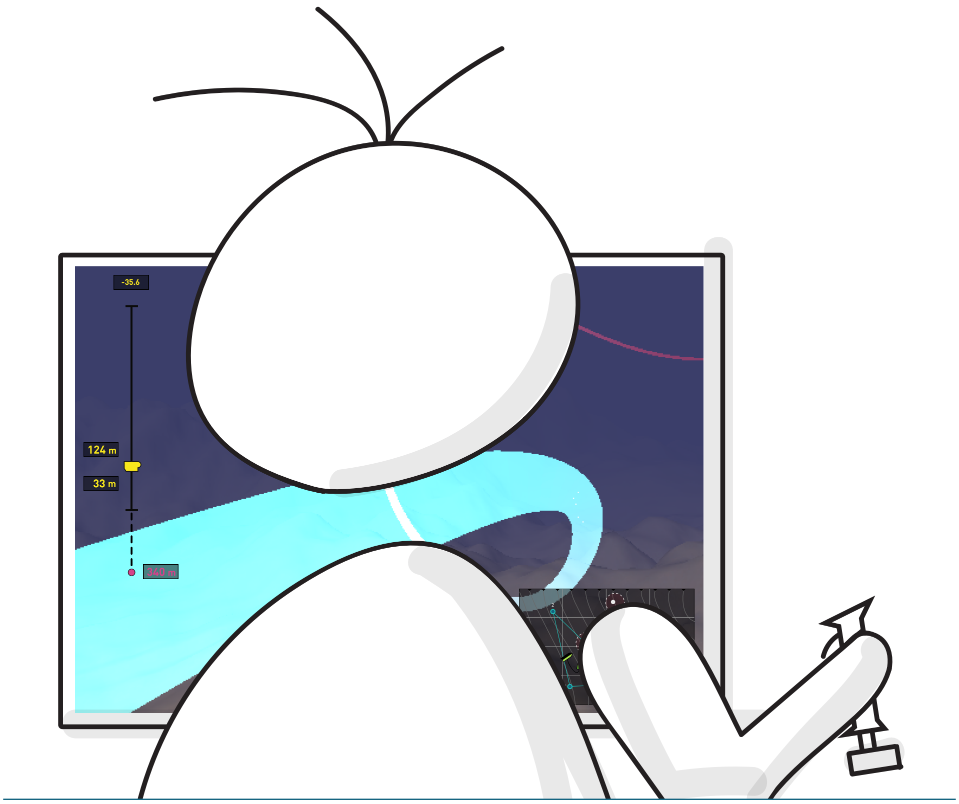
As he now navigates through the 3D world, the other ROV follows its own trajectory as time passes. The trajectory is also visible in the world, so Henning knows where the other ROV will go and can control his current ROV accordingly. "Better watch where my tether is," Henning thinks.

Current

Relates to step 1.



17



Step 1. Brede prepares for the mission by following the trajectories. The virtual world gives the opportunity to learn the site before the mission. Brede prefers to use a joystick.

Tertiary User

Brede Olaussen, ROV operator
Oceaneering

1. PRACTICE

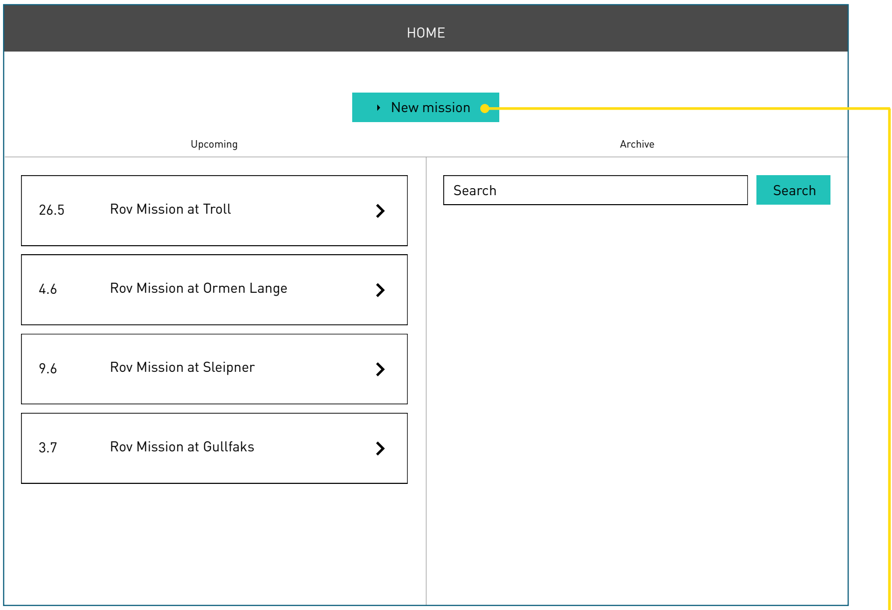
A few days before the ROV mission, Brede is going through the plan to ensure that he knows what to do and where to go when he controls the ROV on the bottom of the sea. Within the ROV planning system, he clicks on the ROV he is assigned to control. The view zooms in on the ROV, and Brede chooses clicks on the FOLLOW TRAJECTORY option that appears. The view changes to the view of the ROV, and in front of the ROV is a line that represents the planned trajectory of the ROV, as created by Henning a few days before. Brede controls the ROV to follow the trajectory (17).

The world around is crystal clear, in stark contrast to the poor visibility 1000 metres below the surface of the ocean. This helps Brede develop a deeper understanding of the surroundings and directions, to help improve his spatial awareness even when the visibility is poor during the real execution of the mission.

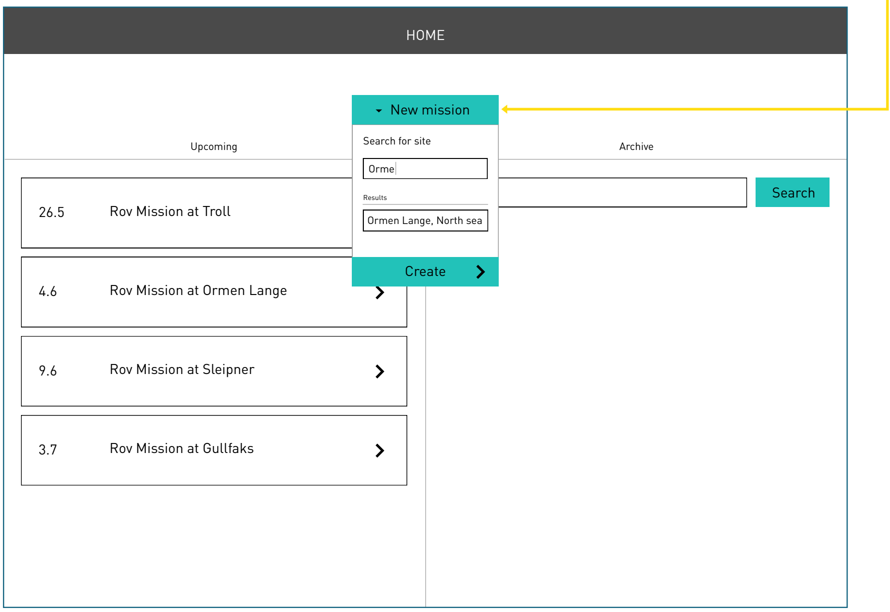
> EXECUTION

As an additional note regarding the system as a planning tool, the information in the system can be used during the real mission execution. The intended trajectories can be used a GUI overlay on top of the video feed from the ROV, and aid Brede in navigating in the dark. The surroundings can also be rendered virtually in real time, together with current security zones and measure. Brede, Henning and Carl will all feel more confident that the mission is as effective and safe as possible. This scenario has not been inside the scope of the thesis.

1
Screenshot: The HOME screen. The user can create a new mission, choose and upcoming mission, or search for dated missions.



2
Screenshot: Before the user can enter the project space of the new mission, he has to specify the site.



2 CONCEPT IN DETAIL

On these pages, I will more precisely present the main functionalities within the system. The navigation of the ROV within the 3D world has its own dedicated section following this one.

Elements that only surround the virtual 4D sea map, have not been the main focus. However, I wanted to explore where the the main work would fit in. This research is based upon knowledge from user research, but it has not been tested by the real end users.

The system is therefore still an early prototype, and can be considered a proposal for further work.

WEB OR NATIVE

This system faces the same dilemma as many mobile applications. Should it be developed as a web application or as a native application for the different operating systems. User research found that the industry are very predominantly Windows users. Nonetheless, a native application requires that the user downloads and installs the app.

A web app, on the other hand, can be accessed from anywhere, and needs to download. Like Google Docs, the web app is perfect for real-time collaborations.

Unlike Google Docs, however, the system I am proposing has to continuously render an interactive 3D environment. Despite javascript frameworks such as Three.js, my belief is that a browser can not process that amount of data fast enough.The system needs to be responsive in all situations, otherwise the user experience will suffer.

As a result, a native Windows application is proposed as the platform for the concept.

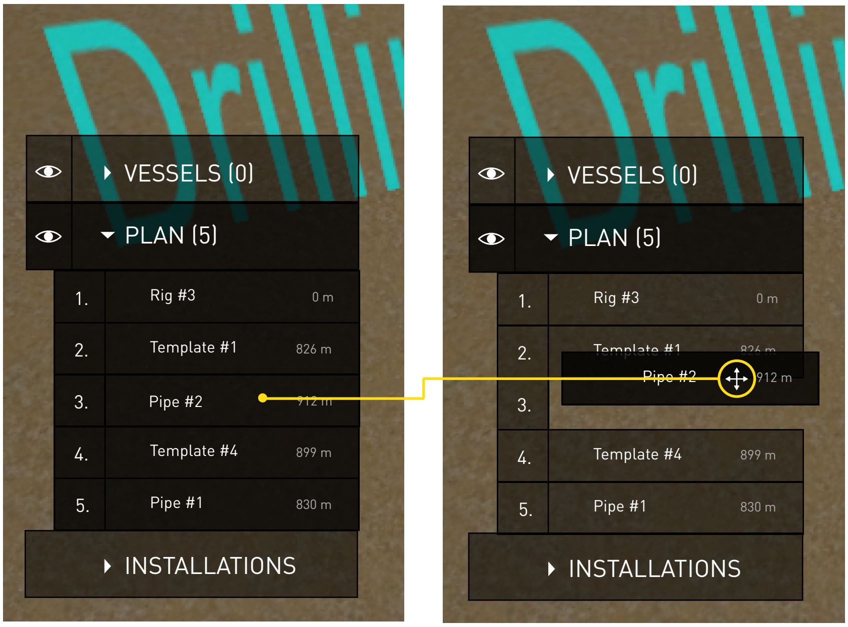
FROM HOME TO PROJECT SPACE

The default Home screen (1) for the application is where the user can choose to start a new mission or to continue working on upcoming missions. The user can also choose to search for already executed missions.

When clicking on NEW MISSION, an action box drops down, and the user has to specify the site of the mission. When CREATE is clicked, the specified site is loaded in the project space (3).

3

Screenshot: The initial project space for the newly created mission.



4
Fragment: The added installations are listed in the order of execution.

5
Fragment: The user can drag and drop the installations in the plan to change order

5.1
The user grabs an item in the list by clicking and holding.

LAYERS
On the left is a layer menu, inspired by the layering found in the Adobe Suite amongst others. The Installations layer contain all installations and rigs on the site. PLAN contains the installations relevant to the mission. Vessels contain the vessels and ROVs used in the mission. The trajectories of the vessels and ROVs will build on the plan, and the plan builds on the installations on site. Hence, the higher the layer, the more mission specific the content is.

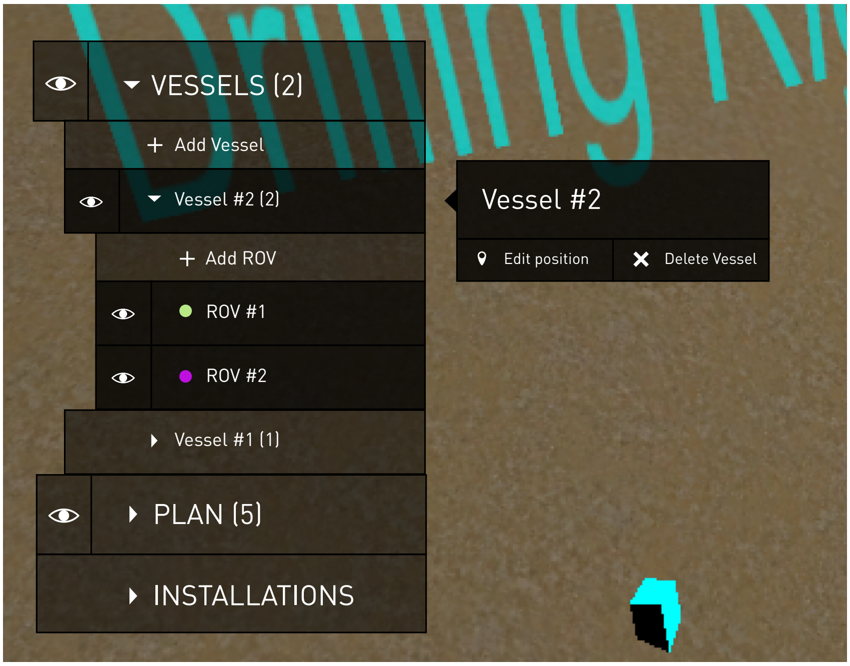
INSTALLATIONS
The bottom layer contains the installations on the site. Placed at the bottom, it represents the foundation, and it can not be hidden. When open, the installations are ordered by type and then alphabetically. When an installation is clicked, it is highlighted in the list, in the 3D world and on the 2D map.

PLAN
The following layer is the PLAN layer. It starts out empty, prompting the user to add the mission relevant installations to it. The added installations will be marked on the map, and a continuous line will traverse between them in both the 3D and 2D view, showing the order of events. This order is decided by the order in the list (4).

Installations are added by selecting them from either the 3D world, the 2D map, or under the Installations layer. When selected, the 3D view will center on it, and an action box will appear next to the installation in the current main view (either 2D or 3D). By clicking ADD TO PLAN, the installation is now added in the list beneath PLAN.

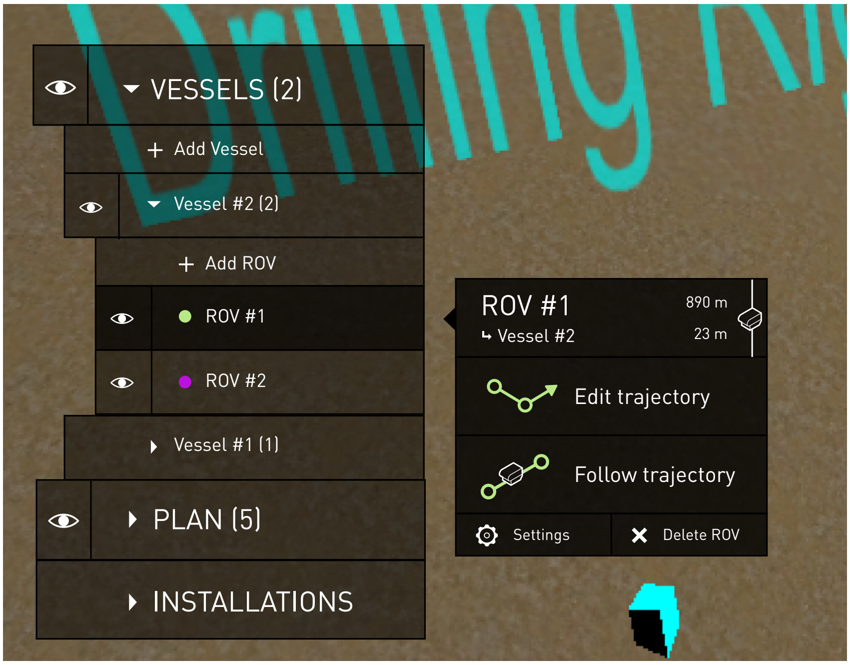
6

Fragment: The list of vessels and their ROVs. An action box appears when the user selects an item in the list.



7

Fragment: The user has selected an ROV and can edit its trajectory, follow it as a practice, or he can reset the settings determined when it was created (8). Deleting the ROV entirely is also an option.



8
Fragment: The user has chosen to add an ROV to a vessel, and must input some options before it can be added to the site.

VESSELS
The next layer is Vessels (6). This layer contains all the vessels included in the mission. ROVs are added from each separate vessel, as they are also physically attached and controlled from a vessel. A vessel can hold several ROVs.

A vessel is automatically added to a default position in the 3D world. The user can then move the vessel to position it according to the relevant installations.

ROVS
An ROV is added under a vessel in the Vessels layer (7). When added, a prompt will appear on the screen (8). The user will have to specify which side of the vessel the ROV is deployed from, if the ROV has a garage and the initial depth of the garage. If a garage is selected, the starting position of the ROV is inside the garage at the specified depth. If not, the ROV will be placed on the surface, 20 "metres" from the vessel on the specified side.

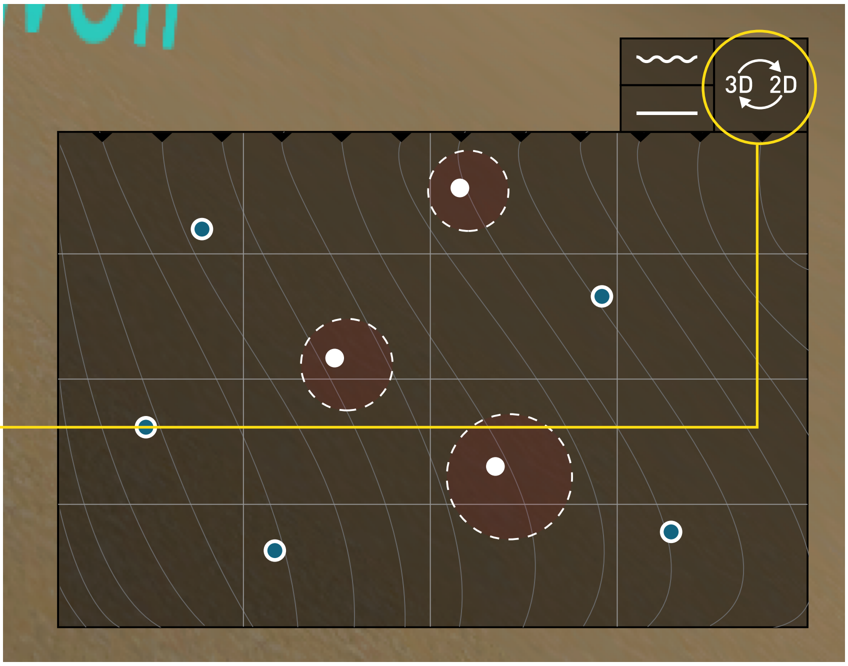
HIGHLIGHTED OBJECTS
The user can highlight an installation, a vessel, or an ROV from three separate places: the layer menu, the 3D view or the 2D map. As they all give the user a unique piece of information about the same object, the object will be highlighted in all three views regardless of where the user clicked.

When highlighted, the 3D view will center on the object, and an action box will appear next to the click. The content of the action box changes according to the type and state of the object, but this box is always the only place user can manipulate an object, whether it is to move, add to plan, add trajectory, delete, etc.

9

Fragment: The 2D map. The circles show the safety zones, and the fluid lines show the current. Triangle at the top tell the user the current is moving downwards.

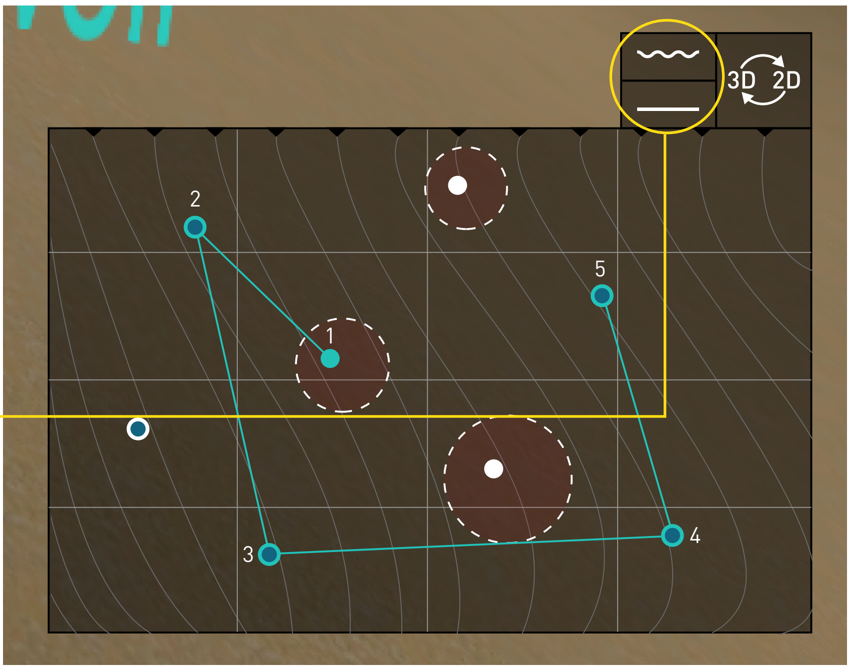
Switch 3D and 2D

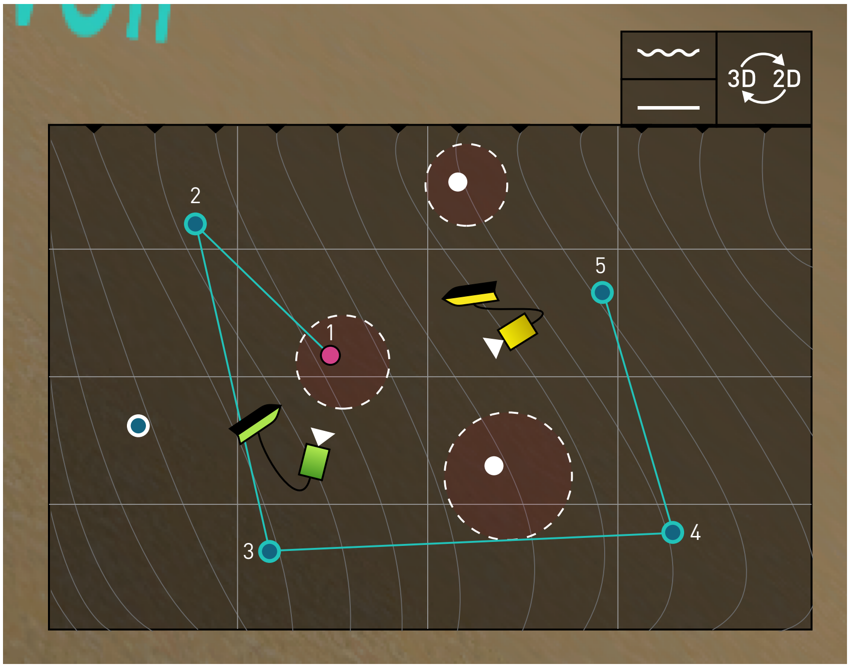


10

Fragment: The lines show the order of the installations in the list. The buttons on top are for either switching between 2D or 3D in the main view or to go to a default position at the surface or the bottom of the site.

Surface/ seafloor





11
Map with added vessels and ROVs. Color on the vessel show which side the ROV is deployed from.

3D AND 2D VIEWS

The project space includes both a 3D and a 2D view. The 3D view gives the user a realistic viewpoint of the site (3), and both the ROV supervisor and pilot benefit from an increased spatial understanding of the site. As 3D trajectories are added, it is also easier to create and see them in 3D.

The 3D view is a relatively stylized render of the site. As there is no interaction with the installations in the concept, the 3D models are simplified. Also rendered in the scene are the safety zones. As mentioned in section 3.5 Challenges, there are several requirements to ensure the safety of the mission.

The 2D map gives an overview of the site. Depth is difficult to see in 3D on a flat screen, and the 2D map helps the user understand the distance between objects in the site.

The 3D view is the default view of the site, while the 2D map is in the bottom right corner. They user can change this by pressing the button in the top right corner of the map (9, 10, 11).

Oil or gas sites are enormous, such as the Ormen Lange reservoir. The good thing is that between the surface and the seafloor, there is nothing. Hence, a large part of the water column does not need to be shown to the user. The user can therefore go to a default position at the surface or the seafloor in the 3D view by clicking the appropriate button above the map (9, 10, 11). The 2D map shows all objects at the surface and seafloor simultaneously.

Ormen Lange
Norways second largest gas reservoir. Covers 350 km2 and the depth is between 800 to 1100 metres. All installations on site are subsea, linked to an on-shore facility.

12

Screenshot: The timeline is placed at the top of the project space if there is a trajectory in the project.

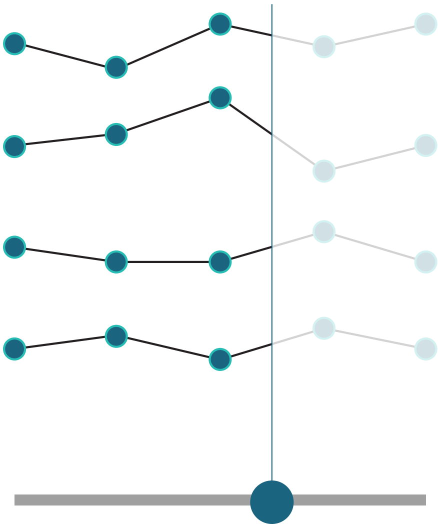
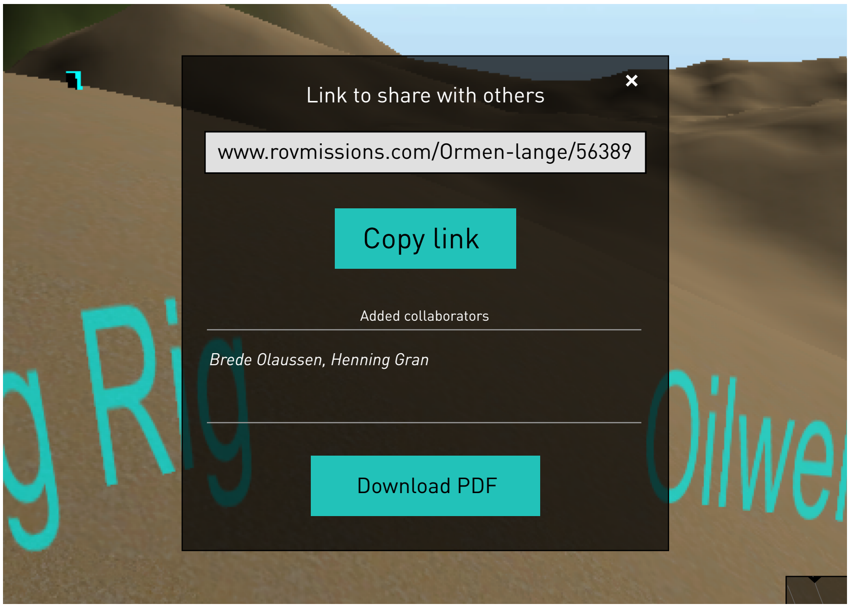


Figure 1: A visual 2D representation of how dragging the knob on the timeline affects the trajectories.



13
Fragment: The action box when SHARE is clicked appears in the middle of the screen. The user can share a link to the project space, or send the information as a pdf.

TIMELINE

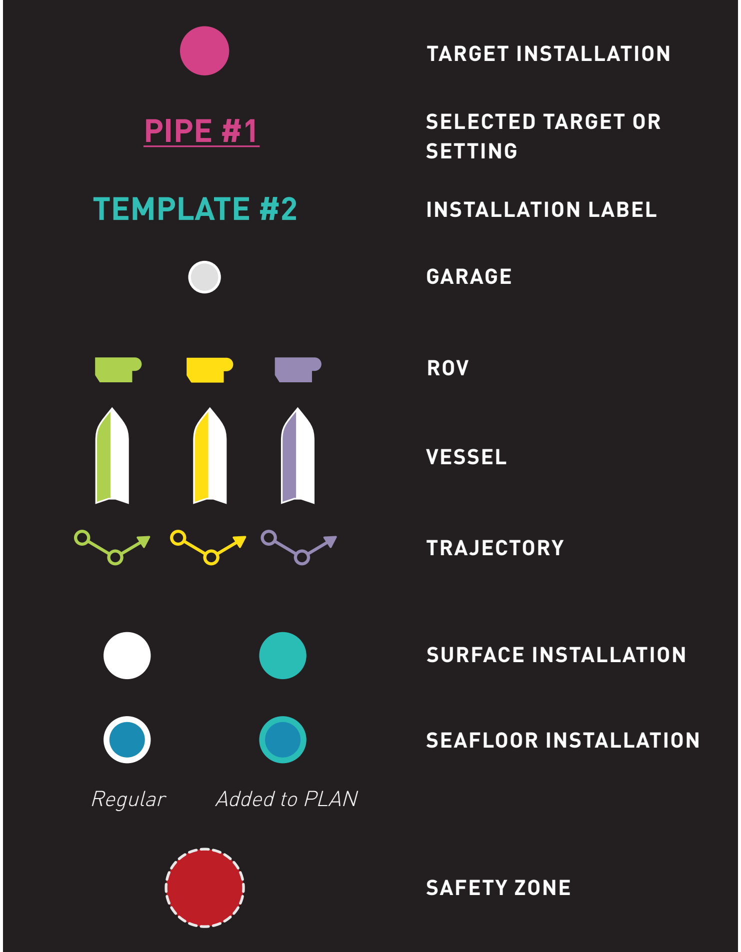
When a trajectory has been added, there is a time value to it. To manipulate this value, a timeline appears just beneath the top bar (12). The length of the timeline is always constant, it is only the resolution of the timebar that changes. The knob on the timeline is automatically to the right, because this is the last position of the ROV.

As the knob is moved backwards, all the ROVs travel backwards along their axis (figure 1). If the user chooses to edit the trajectory of an ROV, he can choose to begin at the point of the knob. On the right side of the timeline, is the total time. If the knob is not at the beginning or at the end, the time value for the position of the knob appears underneath the knob. To the left of the timebar is a small dot, which automatically places the knob at time zero when clicked.

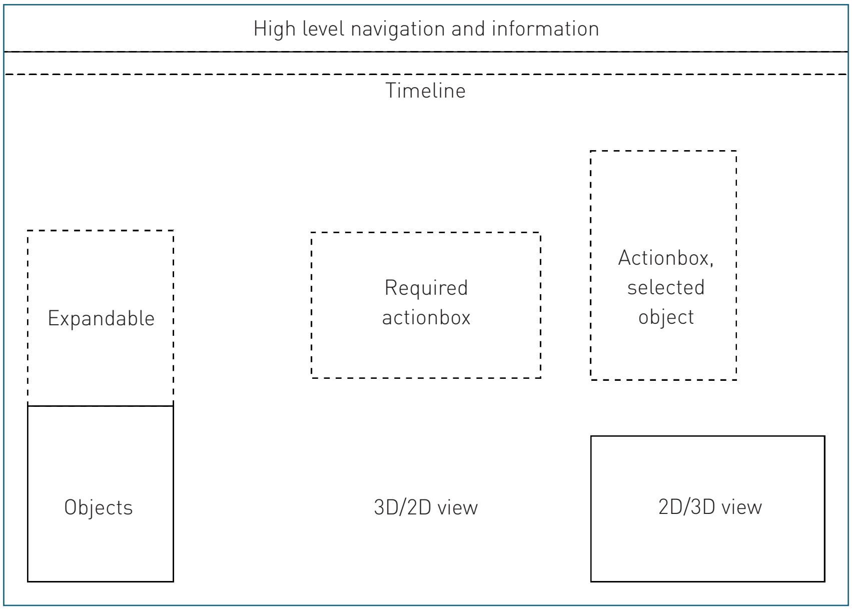
SHARE

The concept includes a collaborative part. A user can give access to the project space to colleagues. The technology used is a big factor in how access is given, and has not been researched. In the current concept, a simple "copy link and send" approach is proposed, similar to that in Google Docs. In the same box, the user can see who has access (13).

When a mission has been specified by the well operator and is ready to be tendered to the market, the user can download a pdf containing the mission information in a compressed manner (13).


TARGET INSTALLATION
PIPE #1
SELECTED TARGET OR SETTING
TEMPLATE #2
INSTALLATION LABEL
GARAGE
ROV
VESSEL
TRAJECTORY
SURFACE INSTALLATION
SEAFLOOR INSTALLATION
Regular
Added to PLAN
SAFETY ZONE

COLORS



14
Figure: The basic framework for the concept. The hard lines are constants, while the dashed lines are dependent on user actions. The actionbox for a selected object will appear next to the position the mouse was clicked.

COMMENT
Collaboration is important. In the project space, users can leave comments to discuss the project. However, I have not researched how commenting should be done, and will therefore not suggest any design for the comments sections. The button to such a comments section is added in the top bar of the project space as a suggestion to where the user could access the functionality.

FRAMEWORK & COLOR
The framework of the layout is shown in (14). It shows how the distribution of actions and navigation is distributed across the screen. To the left is a presentation of the different colors and their meaning in the context of the concept. They are presented on a black backgroud, because the background in the concept itself is also dark.

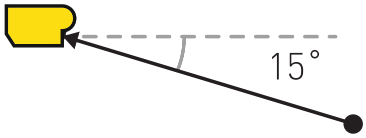


Figure 1: The position of the camera and where it points in the ROV view.

Screenshot 1: The ROV view, with the accompanying GUI overlays and the 2D map

3 NAVIGATION: POINT OF VIEW

For the final concept, I have decided to leave out the Oculus Rift, due to the reasons discussed in section 6.4 HMD vs Screen. This section will present an alternative concept to increase the user's situation awareness in the 3D site, and the platform is a normal computer screen.

During user testing and research, I have identified four key points which the concept should address. It is difficult to...

... know the size of the ROV relative to the site and its installations.

... keep control of the tether.

... get an overview of the site without moving and rotating the ROV an unnecessary amount.

... plan a trajectory keeping in mind all the installations that come next.

The concept will handle these key points by offering the user the freedom to choose his own vantage point. This freedom is expressed through three different viewpoints, in addition to a 2D map.

ROV VIEW
The default choice is a viewpoint from the front of the ROV. Unlike real ROVs, the manipulators are not in the line of sight, in order to clear up the screen as much as possible. The camera is tilted 15 degrees downwards, because the interesting objects during an ROV mission are below the ROV (figure 1). The ROV view offers the most immersive user experience, as this is the most realistic one. The biggest advantage that the user has full control of what comes up close to the ROV.

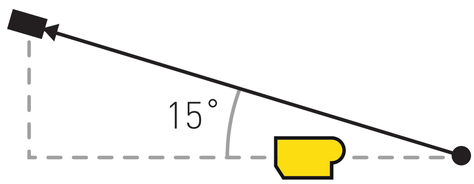
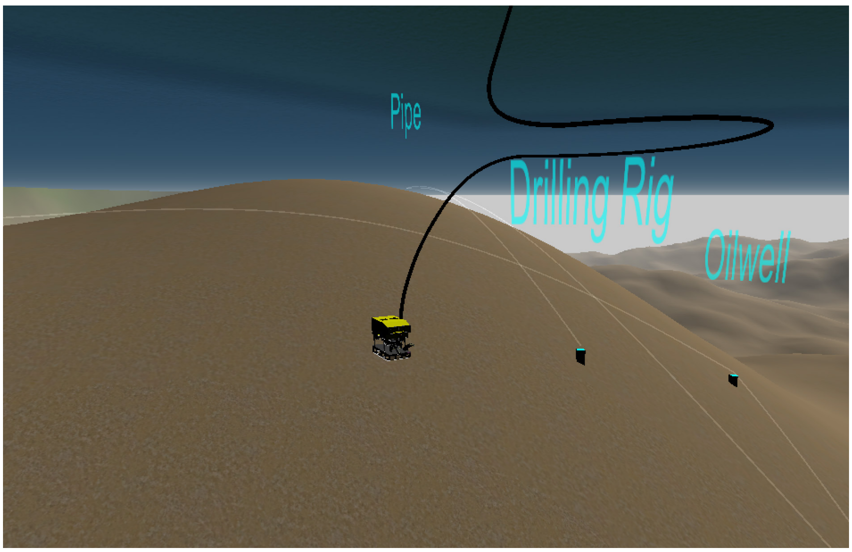


Figure 2: The position of the camera and where it points in the Follow View

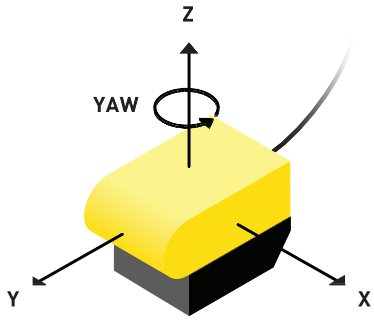
FOLLOW VIEW
This is essentially a 3rd person view, positioned behind the ROV at angle, looking slightly ahead of it (figure 2 and screenshot 2). This is to see more of what is coming up in front of the ROV, and not what it has travelled past already. There are several benefits of the Follow View. The first is that it is easier to see the actual size of the ROV, in case precision control is needed. The second is that the viewpoint is from a higher point, so the user has a better view of the site. The higher viewpoint also allows the user to have a greater awareness of where other ROVs or trajectories are at the same time.

FREE
The free view is all about exploring the site, literally being free. When the user enters the Free View, the current mission is paused and the user steps out of the ROV and the camera is "floating" within the scene (screenshot 3). Using the same controls as for the ROV, the user is now free to inspect the installations in the plan, check out the existing trajectories and determine how to go next. When the user returns to the ROV, the mission continues where it left off. The best thing about the Free View is that it offers an easy way to cruise through the site without actually creating or destroying anything. The Free View will be key in co-planning trajectories for several ROVs.

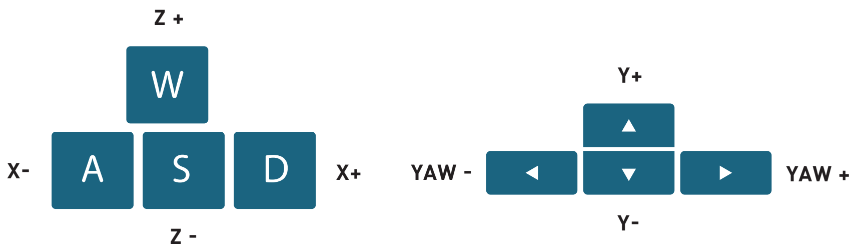
Screenshot 2: The user is in Follow mode, which is a 3rd person view of the ROV. It gives the user a better sense of the size of the ROV in relation to its surroundings.



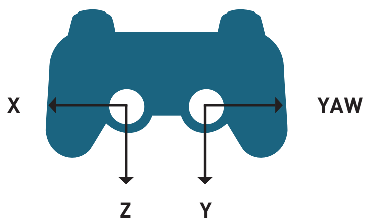
Screenshot 3: The user is in Free mode. The ROV is stationary, while the user can explore the world around without worrying about the time or the trajectory.



Keys

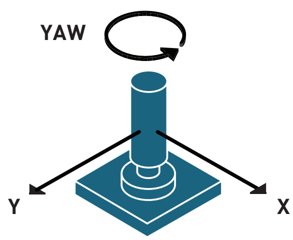


PS3 Controller



Joystick

Note: Depending on the joystick, Z-axis is mapped to two buttons or to a thumbstick on top.



4 NAVIGATION: CONTROLLERS

WHAT AND WHY

The controllers have different qualities, and the three users have different levels of experience. In this section, the concept for the use of controllers will be presented.

The best controllers according to the testing were the keyboard, the PS3 controller and the single joystick. Their strengths are complementary: the keyboard offers overall accessibility, the PS3 controller is wireless and simplified, while the joystick is realistic and accurate. These qualities each fit the system in their own way. The keyboard is perfect for the user who may not have the ROV pilot training, and just wants to navigate quick and easy. The PS3 controller is good for a cooperative setting, where the viewport is on a big screen and the participants of the meeting walk freely. The joystick is best for the ROV pilot who wants to practice navigating the site, because of what they are used to and because it will give a more immersive user experience.

The final concept for the controllers is emphasizing flexibility and an open-source approach, because there is, in my opinion, no controller which is the best of all worlds. Hence, in the final concept I have decided to open the system up for the three controls.

The reason why this is possible is due to the accessibility and simplicity of them all. All of them can be connected via USB, the PS3 controller and some keyboards use bluetooth, and some keyboards are integrated into the computer. They are cheap and easy to get, and easy to learn. In the view of development, the mapping of input axes is quick, easy and cheap. The joystick and the PS3 controller are treated the same way by the machine, controlling the same axes. Essentially, it's just plug and play.

HOW

The mapping of the axis to the controllers are shown on the page to the left. The PAUSE menu is mapped to the SPACE bar on the keyboard and the START button on the PS3 controller. The PAUSE button on the joystick will depend on the model.

The challenge when switching between the controllers is the mapping of the buttons that carry other functions. Therefore, the functions attached to these buttons can also be reached through the PAUSE menu that is accessible at all times. Expert users will still be able to access the functionalities quickly through the buttons on the controller. Due to this, and because I have not tested this mapping, a design suggestion is not presented.

Screenshot 1: The initial project space for the newly created mission.

5 NAVIGATION IN DETAIL

The GUI overlay when the user is navigating the 3D site is presented in this section. As explained in the previous sections, the user will start by clicking CREATE TRAJECTORY for the relevant ROV. The view will change to a first person view of the ROV, with a prompt telling the user that the timer will start as soon as he moves the ROV.

MAP

The 2D map on the bottom right is the same is in the project space, except that it is now dynamic. The ROVs in 2D move according to the ROVs in 3D, and the currents are animated subtly to show the direction.

The map makes it easier for the user to see where he is compared to the other installations, vessels and ROVs. It also enables him to see how the ROV is positioned in relation to the vessel.

In the PAUSE menu, which will be presented later in this section, the user can choose if the vessel should be stationary or follow the ROV. If stationary, there is a chance the ROV can get on the wrong side of the vessel. The user will get a warning and the ROV and vessel will be highlighted in the map.

TIME

The number at the top (1.1), informs the user of the time passed from the time zero into the overall mission. The number is 3 minutes and 20 seconds, but hours is added if necessary. It ticks along as the user controls the ROV.

ARROW AND LINES

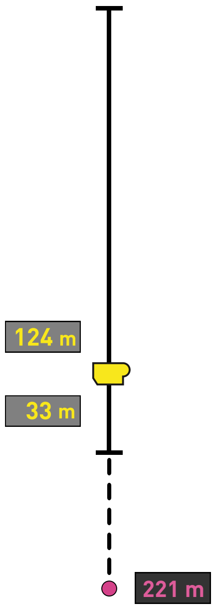
There are two main attributes aimed at helping the user navigate to the next target installation. The first one is the arrow at the bottom of the screen (1.3). It is always pointing at the next target.The arrow will always rotate in the horizontal plane, so it will not tell the user if the target is above or below. During testing, this was a very appreciated form of navigation.

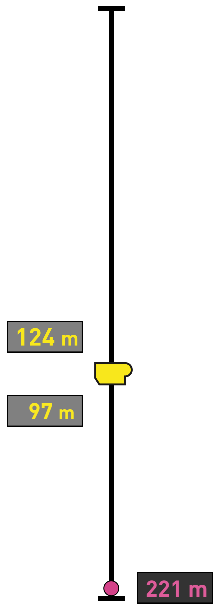
The second one is a line that is extended between the installations in the Plan, in the order they are set. If the user has a good vantage point of the site, this will help him plan the path between them.

To conclude, the arrow tells the user where to go now, while the line tells him where he is going throughout the mission.

2
The current target is at a greater depth than the total depth tat the position of the ROV.

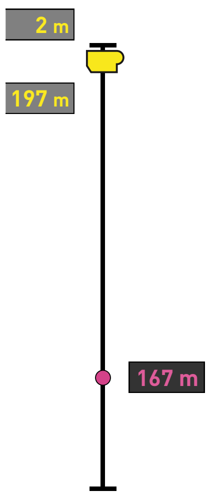
3
The ROV is directly above the target, so the total depth and target depth are equal.

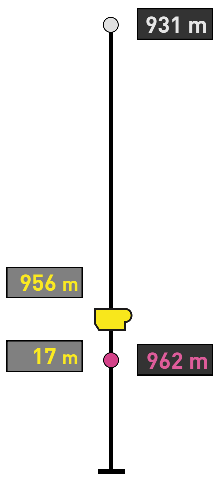


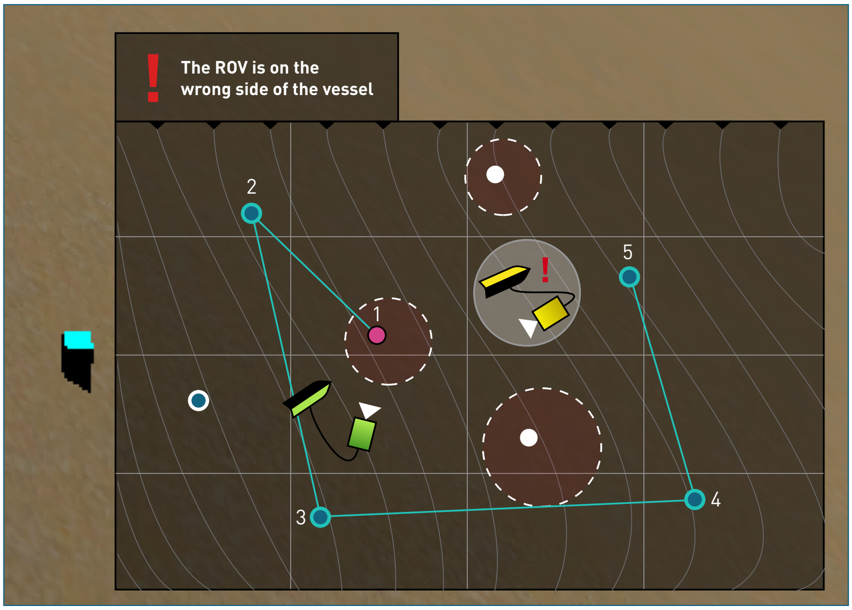


4
The ROV is almost at the surface, and the target is a shallower position than the ROV.

5
The ROV is tether to a garage. The ROV will not go higher than the garage, so it is now the top point in the visualization.







6
Screenshot: The HOME screen. The user can create a new mission, choose and upcoming mission, or search for dated missions.

VERTICAL POSITION
The depth and altitude of the ROV, along with the depth of the next installation on the list, is visualized on the left side of the screen. The yellow icon represents the ROV, while the magenta colored dot is the next target.

The full line is the depth from surface to bottom at the location of the ROV. The dashed line occurs if the next target is at a greater depth than the total depth at the position of the ROV.

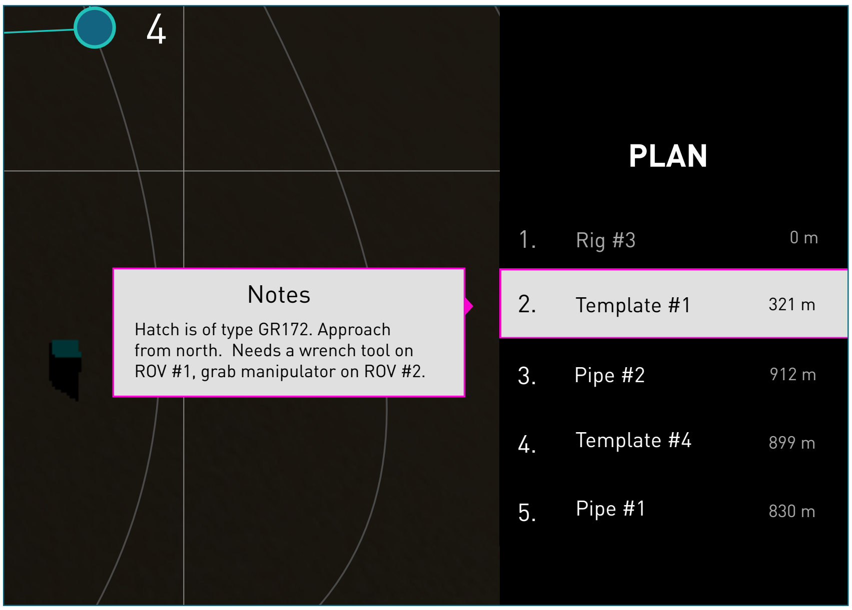
The numbers on the left are the depth and altitude of the ROV. The magenta number is the depth of the target. As the ROV travels, the numbers will change accordingly.

The line will also change. The top of the line is fixed, while the bottom part of the line will extend if the total depth increases. A line 20px from the bottom of the screen equals the maximum depth of the actual site. This means that for each site, the resolution of the line is different. The pay-off, on the other hand, is that if the ROV icon moves 1 cm on the line, it will always equal the same distance regardless of the current depth for the entire site.

The topmost number indicates the number of turns the ROV has made around its own axis. Positive is clockwise, negative is counter-clockwise.

7
Screenshot: The HOME screen. The user can create a new mission, choose and upcoming mission, or search for dated missions.

8
Screenshot: Before the user can enter the project space of the new mission, he has to specify the site.



9
Screenshot: The HOME screen. The user can create a new mission, choose and upcoming mission, or search for dated missions.

INSTALLATIONS
The installations are labelled with their name. The next target is highlighted in the magenta color. The label follows the user, so they are always pointing towards him.

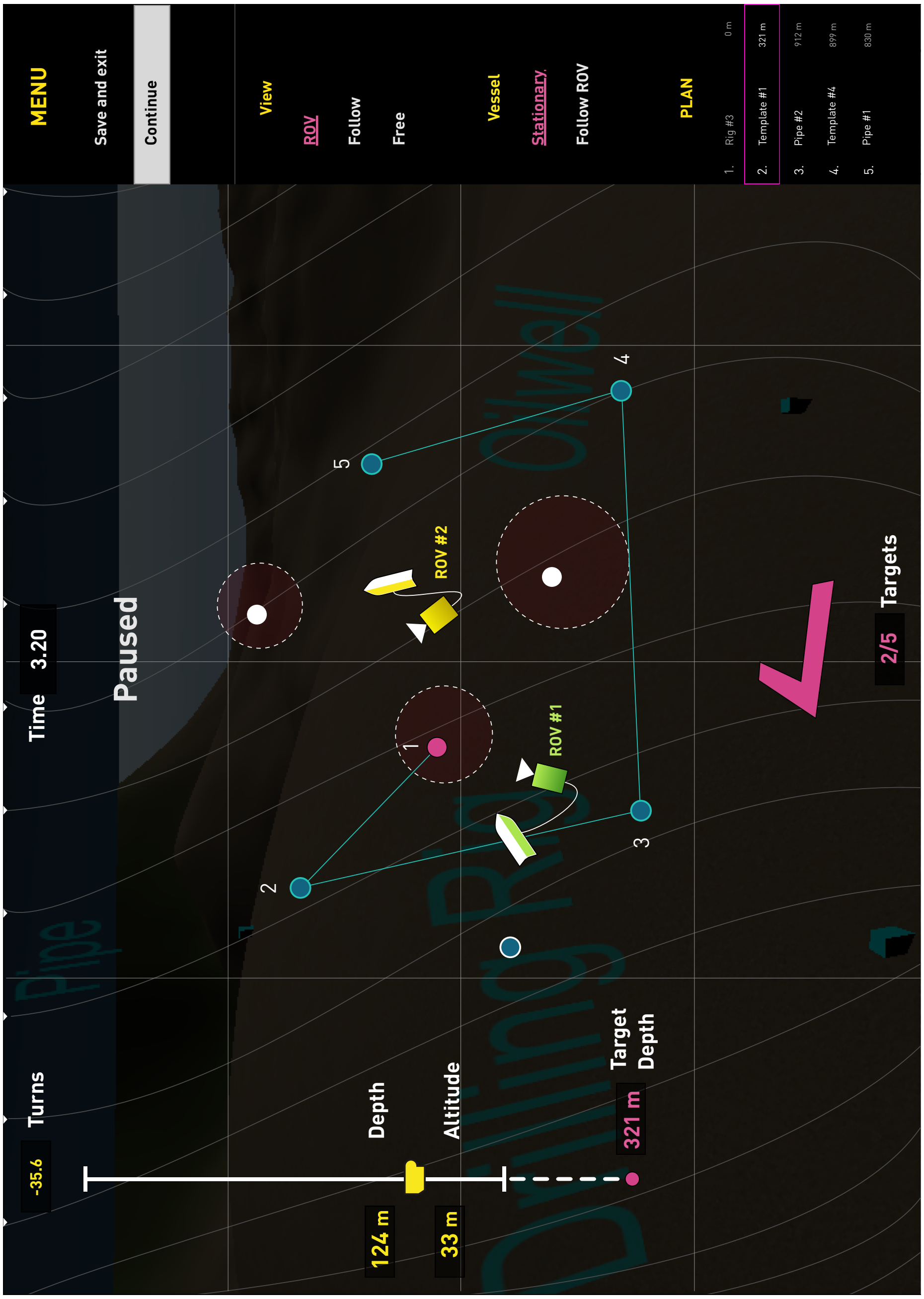
TRAJECTORIES
The trajectories of other ROVs and the current trajectory of the controlled ROV are rendered in the 3D world. Their color matches the color of the ROV.

CURRENT AND TETHER
A current is simulated in the water, and it is based on real data. The tether of the ROV will be affected by the current. The user can choose to have a better view of the tether Follow view or in the Free view. The length of the tether is optimized by the simulation.

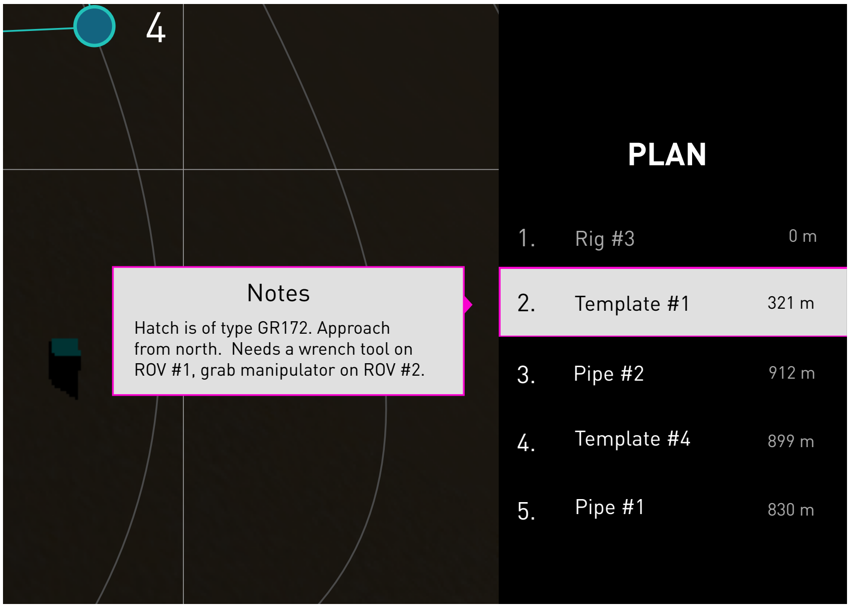
SECONDARY AND TERTIARY USER
The secondary user, the project manager, would only need to navigate in the 3D world to get an overview over the site in order to make a better initial plan for the mission. His needs will be met by the 3D map in the project space.

The tertiary user, the ROV pilot, can use the system during the planning phase along with the ROV supervisor, but the extraordinary scenario for this user is to prepare for the mission by navigating in the 3D world and following the planned trajectories. This will make the user familiar with the site prior to the real mission, and therefore improve the user's situation awareness.

10



Screenshot: The initial project space for the newly created mission.



11
The user can read the notes on the installations in the PLAN.

PAUSE
To user can pause while controlling the ROV. As seen screenshot 8, the 2D map is expanded to an overlay spanning the entire screen. The GUI from navigation mode is placed on top of the overlay. The pause menu places a title or name next to the numbers and icons from the normal navigation view, so there is no doubt what the numbers mean.

On the right of the screen, is the MENU. The menu is designed to fit a controller or keyboard, not a mouse. The default option is to continue controlling the ROV. The user can then choose to go up to Save and Exit, or go down to make other, relevant choices. This feature is deliberate to minimize the risk of pressing Save and Exit by mistake. The risk is low, as the user can choose to continue with the trajectory from the project space in case he was not finished.

Under the VIEW title, the user can choose between the different view modes available, as explained in section 7.3 Navigation: Point of View.

The user can set his preference for wether the vessel should follow the ROV or be stationary in the current position.

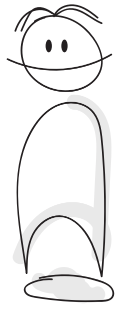
Further down, he can read the notes on the installations in the plan. He can also change the order by selecting an installation and then moving it either up or down.

10 EVALUATION

>	Introduction	193
1	Reflection	195

> INTRODUCTION

The thesis I started out with is not the same thesis I deliver. The path towards the goal has been dynamic and sometimes surprising. In 8.1, I reflect upon the process and important aspects of are discussed. I look in hindsight on things I would do differently, and I look forward by proposing what could be done in the future.



1 REFLECTION

USER-CENTERED DESIGN

The thesis is founded upon a user-centered design approach. The process has been "fuzzy" as the prototyping and research was performed simultaneously, both affecting each others directions. Design methods have been to gather and analyze insight into the subsea ROV domain and the relevant users. Research has also been made into the domain of virtual reality, HMDs, physical controllers of virtual content and cognitive psychology. Prototyping and user testing have brought those two domains together.

The user insight was useful to guide the research in relevant directions, so that the final results were relevant to their needs. Service design methods gave an understanding of what needs the primary, secondary and tertiary users have, and when those needs occur. This was the foundation for the concept as a system, and the GUI design is a proposal based on that foundation.

CONTACT WITH INDUSTRY

I mentioned that I was intent on involving users as much as possible at the start of the project. A lot of time went into trying to find contacts in the industry, but it proved to be very difficult, as there is no ROV industry in the Trondheim area. Originally, Kongsberg was supposed to cooperate on the thesis, but it did not work out that way. In addition, they could not help get users. Kongsberg produce software and hardware for the ROV industry, and thus have no ROV personnel to contact.

Of comparable, existing products, there are none on the market focused on the planning phase. Current products simulate reality, and are designed as a tool for ROV pilots to practice interactions with specific installations. Of such products, only the Oceaneering ROV simulator has been tested. It would have been interesting to test others, but it is not possible to get access to most of them unless you are an industry professional.

PROCESS
At the start, I made a Gantt diagram to visualize the time available for the thesis. I filled the diagram with tasks I wanted to perform, so that I would have a structure for the work at hand and an overview during the process of where I should be and what to do next. The real process did not match the diagram, as low-fidelity prototyping, high-fidelity prototyping and user research all began straight away.

The initial focus was on testing. The motivation for the thesis was the prospect of testing the Oculus Rift and trying to create functional prototypes in the Unity game development platform.

Prior to the thesis, I wanted to work together with a computer science student in order to create a fully functional, final concept, but was unsuccessful in getting a partner. Therefore, the official starting point was to create a design concept, as I have relatively little experience with programming.

Despite the starting point, I wanted to try Unity to see if I was able to create something. I had never programmed with C# before, but the learning curve was steep and I decided early to use Unity prototypes for testing.

Prototyping and testing took a lot of time, and the development of the overall concept and GUI happened late. I still wanted to do it, to make it clear where my research would fit in and how it could be benefitial. Had there been more time and resources, I would travelled to Stavanger, where there is a lot of ROV related industry, for an extended period, doing user research and following real operations.

In hindsight, it would have been beneficial to look at other design methods or to have a strictly theoretical approach from the start to eliminate the need to bring in real users.

ITERATIVE PROTOTYPING
One goal was to have three structured rounds of prototyping and testing. However,

prototyping was done at several levels, so it was not possible to put it on a linear timeline. Formal testing was diversified between navigation and controllers, and it required more time than anticipated.

The formal tests did go through some iterations, but they were to improve the test and its results, rather than design a better concept. For example, after three navigational tests, I made the course more difficult and added distance travelled as a parameter to better differentiate between the conditions.

Informal user testing was conducted with small elements instead of with the entire concept, and they could easily have three iterations during the course of a day.

FORMAL TESTING

It was fun to try formal user testing, with the goal to get comparable, quantitative results. Test users also had fun, and some would continue "playing" after the test was concluded.

Testing conditions have not been up to what one would expect to be a scientific standard. None of the test participants were real end-users. The test users were selected out of convenience, and they represent only a fraction on the entire population. There were more men than women, but ROV pilots are mostly men. There were no considerations taken based on prior gaming skills.

The tested controllers were chosen based on availability. Other types of joysticks or RC controllers might have been interesting to test. There were trouble acquiring a gaming joystick for testing, so I was only able to test it late in the process.

The controller tests were short, which gave less differences than a longer test could have given. The phenomena which made it possible to go faster at an angle is a source of error.

The 3D world in the navigation test was an open field, and results might have been different in a more challenging environment where it could be more useful to look around. Other types of tests could give other results.

CONCEPT

When I presented the thesis to industry professionals I contacted, their reactions were mixed. It was difficult to sell the concept without something real to show for it. Using the Oculus Rift seemed a bit farfetched.

The final concept is influenced by a more reasonable approach than the initial thesis formulation. The mixed responses were due to differing needs and experiences, but they all found some aspects they liked, and wanted me to continue with.

It is my belief that the concept presented in the thesis is a good starting point to tend to those needs.

The GUI for the concept as a whole did not pass through the same formal user testing as the controllers and the Oculus. Small, guerilla style tests were performed on people I knew from before. The GUI has a lot of potential to improve as further testing is conducted. E.g. the layered style menu might not be easy to understand by people who are not familiar with programs such as Photoshop and Illustrator.

NEXT STEP

The final concept is more of a proposal for further work than a final design. If work was to continue, it would be important to get a foothold within an ROV company. They would have an incentive to further the development in order to get a competitive advantage, while the development would benefit from a close contact to end-users. Including the end-user in the development would also anchor the concept with them, making it more likely that they will use the system when it is finished.

As for research goes, I would not exclude HMDs in the future. Augmented reality instead of virtual reality might be a better solution, as virtual content is displayed on top of reality. Microsoft HoloLens will be interesting to look at.

Gesture interfaces are also very interesting, especially for functionality not related to the precision control of the ROV.

Creating trajectories might also be relevant for AUVs, which are autonomous underwater vehicles. Input a set trajectory into the AUV, and it can follow the trajectoy in real life. Interviews uncovered that the industry think AUVs are a big part of the future.

HAVE I LEARNED ANYTHING?

The process of writing a master thesis is a lonely job. The challenge has been to make decisions without sparring with anyone. It has, however, created a greater need to get feedback from testers. Additionally, working alone has given me the chance to focus on what I wanted to learn more about.

As a student, I have done more research and testing than in a real work context. It has provided insight into scientific methods as a tool to get better foundation for design decisions. The intuitive skill to apply the right design method at the right time, has been improved. Nonetheless, there is still much to learn in all these areas.

Time has been of the essence, as they say, and to manage time is a skill in itself. After a project, all students say that they could have worked harder in some phase in order to get a result earlier in the process, so they could have more time to develop the concept or something. I agree that I could always have worked harder in some periods, but I am not sure I would have gotten the same results, and this may not be positive.

I could have shifted focus on to other aspects of the domain, but my main interest lay in developing the functional prototypes. The developing experience I have gained, along with the design methods and scientific methods, will be great assets to have when I now graduate from NTNU.

REFERENCES

Abras, C., Maloney-Krichmar, D., & Preece, J. (2004). User-centered design. Bainbridge, W. Encyclopedia of Human-Computer Interaction. Thousand Oaks: Sage Publications, 37(4), 445-456.

Aftenposten. (2014). "Virtual reality" - denne gangen er det alvor. Retrieved 8.6.2015, from http://www.aftenposten.no/amagasinet/Virtual-reality--denne-gangen-er-det-alvor-7799495.html

Anderson, J. R. (1990). Cognitive psychology and its implications: WH Freeman/Times Books/Henry Holt & Co.

Bartram, L., Ware, C., & Calvert, T. (2003). Moticons: detection, distraction and task. International Journal of Human-Computer Studies, 58(5), 515-545.

Cohen, M. H., Cohen, M. H., Giangola, J. P., & Balogh, J. (2004). Voice user interface design: Addison-Wesley Professional.

Cooper, A., Reimann, R., & Cronin, D. (2007). About face 3: the essentials of interaction design: Wiley Pub.

Cooper, G. (1998). Research into cognitive load theory and instructional design at UNSW. Retrieved August, 8, 2003.

Costabile, M. F., Fogli, D., Letondal, C., Mussio, P., & Piccinno, A. (2003). Domain Expert users and their needs of software development. Paper presented at the UAHCI Conference.

Curedale, R. (2012). Design Methods 1: 200 Ways to Apply Design Thinking: Design Community College Incorporated.

Curedale, R. (2013). Design Methods 2: 200 More Ways to Apply Design Thinking: Design Community College Incorporated.

Drury, C. G., & Clement, M. R. (1978). The effect of area, density, and number of background characters on visual search. Human Factors: The Journal of the Human Factors and Ergonomics Society, 20(5), 597-602.

Endsley, M. R. (1995). Toward a theory of situation awareness in dynamic systems. Human Factors: The Journal of the Human Factors and Ergonomics Society, 37(1), 32-64.

García-Valdovinos, L. G., Salgado-Jiménez, T., Bandala-Sánchez, M., Nava-Balanzar, L., Hernández-Alvarado, R., & Cruz-Ledesma, J. A. (2014). Modelling, Design and Robust Control of a Remotely Operated Underwater Vehicle. International Journal of Advanced Robotic Systems, 11(1). doi: 10.5772/56810

Greenwald, A. G. (1976). Within-subjects designs: To use or not to use? Psychological Bulletin, 83(2), 314.

Hale, L. (1997). Principles and techniques for designing precision machines. (Ph.D.), Massachusetts Institute of Technology.

Koen, P. A., Ajamian, G. M., Boyce, S., Clamen, A., Fisher, E., Fountoulakis, S., . . . Seibert, R. (2002). Fuzzy front end: effective methods, tools, and techniques: Wiley, New York, NY.

Kongsberg Maritime. (2015). Subsea Positioning Systems. Retrieved 8.6.2015, from http://www.km.kongsberg.com/ks/web/nokbg0240.nsf/AllWeb/25018037CECB16D9C125738D004D0BAA?OpenDocument

Kvaløy, J. T., & Tjelmeland, H. (2011). Tabeller og Formler i Statistikk. In NTNU (Ed.): Tapir Akademisk Forlag.

Loranger, H. (2014). UX Without User Research Is Not UX. Retrieved 9.6.2015, from http://www.nngroup.com/articles/ux-without-user-research/

Louise Barriball, K., & While, A. (1994). Collecting data using a semi-structured interview: a discussion paper. Journal of Advanced Nursing, 19(2), 328-335. doi: 10.1111/j.1365-2648.1994.tb01088.x

Mack, A., & Rock, I. (1998). Inattentional blindness: MIT press Cambridge, MA.

Martin, B., Hanington, B., & Hanington, B. M. (2012). Universal Methods of Design: 100 Ways to Research Complex Problems, Develop Innovative Ideas, and Design Effective Solutions: Rockport Publishers.

Minesto AB. (2015). Ocean Energy. Retrieved 12.6.2015, from http://minesto.com/ocean-energy/#tidal-currents

Mulder, S., & Yaar, Z. (2006). The User is Always Right: A Practical Guide to Creating and Using Personas for the Web: Pearson Education.

Nielsen, J. (1994). Usability inspection methods. Paper presented at the Conference Companion on Human Factors in Computing Systems, Boston, Massachusetts, USA.

NORSOK. (2012). U-102 Remotely operated vehicle (ROV) services. Rev. 2, Sept. 2012.

Oceaneering. (2015). ROV Systems. Retrieved 8.6.2015, from http://www.oceaneering.com/rovs/ROV systems/

Oculus VR. (2015). Oculus Rift. Retrieved 8.6.2015, from https://http://www.oculus.com/rift/

OpenROV. (2015). OpenROV. Retrieved 8.6.2015, from http://www.openrov.com/

Pasch, M., Bianchi-Berthouze, N., van Dijk, B., & Nijholt, A. (2009). Movement-based sports video games: Investigating motivation and gaming experience. Entertainment Computing, 1(2), 49-61.

Peterson, H., & Dugas, D. (1972). The relative importance of contrast and motion in visual perception. Human factors, 14, 207-216.

Rosson, M. B., & Carroll, J. M. (2009). Scenario based design. Human-computer interaction. Boca Raton, FL, 145-162.

Stickdorn, M., & Schneider, J. (2012). This is service design thinking: Basics, tools, cases: Bis.

The Verge. (2015). The Rise and Fall of Virtual Reality. Retrieved 8.6.2015, from http://www.theverge.com/a/virtual-reality

Tracy, J. P., & Albers, M. J. (2006). Measuring cognitive load to test the usability of web sites. Paper presented at the Annual Conference-society for technical communication.

Ware, C. (2012). Information visualization: perception for design: Elsevier.

Weinschenk, S. (2011). 100 Things Every Designer Needs to Know About People: New Riders Publishing.

Wickens, C. D. (1992). Engineering psychology and human performance: HarperCollins Publishers.

Wikipedia. (2015). Remotely operated underwater vehicle. Retrieved 8.6.2015, from http://en.wikipedia.org/wiki/Remotely_operated_underwater_vehicle

Williams, L. J. (1985). Tunnel vision induced by a foveal load manipulation. Human Factors: The Journal of the Human Factors and Ergonomics Society, 27(2), 221-227.

APPENDIX

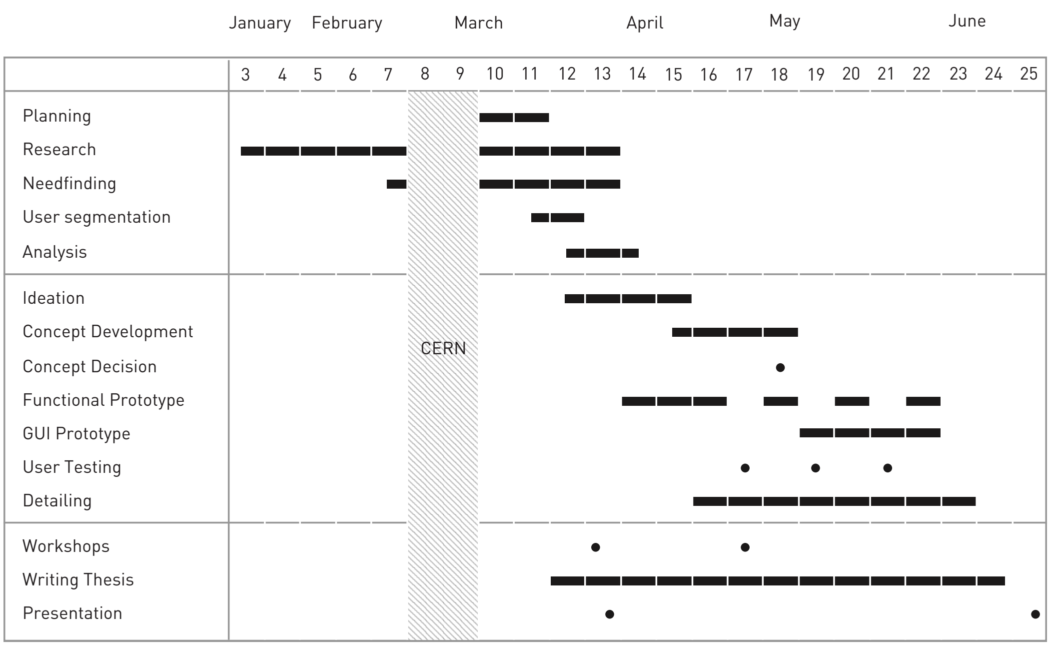
Activity Log	206
Gantt Diagram	207
Navigation Results	208
Controllers Results	212
ROV Observation	214
Interview	216
Unity Prototype	220
Unity Code	227

ACTIVITY LOG

WHEN	WHAT	WHO		WHERE
12.2	Meeting	Thor Hukkelås	Kongsberg Maritime	IPD, NTNU
		Thomas Porathe	Prof. IPD, NTNU	
		Trond Are Øritsland	Prof. IPD, NTNU	
13.2	Meeting	Martin Ludvigsen	Prof. Marine, NTNU	Tyholt, NTNU
27.2	Tour	Joao Pequinao	CERN Media Lab	CERN, Geneve
3.3	Meeting	Thomas Porathe	Prof. IPD, NTNU	KOGT, Asker
		Thomas Hammer	Kongsberg OGT	
		Terje Wiesener	Kongsberg OGT	
		Øystein Hole	Kongsberg OGT	
11.3	Meeting	Casper Boks	Prof. IPD, NTNU	IPD, NTNU
19.3	ROV mission	2 teachers, 4 studs.	Marine Tech., NTNU	Heggedal,
26.3	Mid-presentation	Class	IPD Students	IPD, NTNU
5.5	Guidance	Anna Lunna	Interaction Designer	BEKK Trondheim
13.5	Demo / interview	David Knutsen	Oceaneering	Stavanger
		The manager	Oceaneering	
26.5	Phone interview	Brynjar Wiig	DOF Subsea	Phone/Trondheim
12.6	Delivery			
18.6	Presentation			

** User testing, meetings Thomas, and other research activities are not listed*

GANTT DIAGRAM



NAVIGATION RESULTS

				Times in seconds			Ratio, Time			Delta Time	Distance
Order	**Subject**	**Age**	**Gender**	1 Reality	2 Virtual	3 Oculus	1 on 2	1 on 3	2 on 3	3 - 1	1 Reality
1,2,3	1	24	Male	247	32	32	7.72	7.72	1.00		N/A
1,3,2	2	25	Male	267	42	39	6.36	6.85	1.08		N/A
2,3,1	3	26	Male	240	42	42	5.71	5.71	1.00		N/A
NEW, r = 1/2											
1,2,3	4	28	Male	247	80	80	3.09	3.09	1.00	-2	774
	5	25	Woman	302	115	112	2.63	2.70	1.03	-39	976
1,3,2	6	25	Male	208	84	89	2.48	2.34	0.94	-15	911
	7	25	Woman	505	142	122	3.56	4.14	1.16	-130.5	1843
2,3,1	8	27	Male	179	100	84	1.79	2.13	1.19	-5.5	970
	9	26	Woman	453	162	146	2.80	3.10	1.11	-80.5	1498
2,1,3	10	23	Male	213	139	110	1.53	1.94	1.26	3.5	1071
	11	24	Male	197	103	97	1.91	2.03	1.06	-1.5	983
3,1,2	12	24	Woman	258	96	109	2.69	2.37	0.88	-20	1089
	13	24	Men	186	104	104	1.79	1.79	1.00	11	936
3,2,1	14	25	Woman	393	95	98	4.14	4.01	0.97	-98.5	2054
	15	26	Man	174	90	96	1.93	1.81	0.94	9	873
SUM		25.1		139	112	106	2.48	2.58	1.05	-34	1200
Testere	11										
Avg						119					

		Ratio, Distance			Delta Distance				
2 Virtual	3 Oculus	1 on 2	1 on 3	2 on 3	3 - 1	**Opplevd raskest**	**Opplevd kortest**		
N/A	N/A	N/A	N/A	N/A					
N/A	N/A	N/A	N/A	N/A		2	N/A		
N/A	N/A	N/A	N/A	N/A		2	N/A		
779	789	0.99	0.98	0.99	15	2 and 3	Obs! Other parametres and variables, speed in 1 is		
861	877	1.13	1.11	0.98	-99	3	2		
895	922	1.02	0.99	0.97	11	2 eller 3	2 eller 3		
1083	1014	1.70	1.82	1.07	-829	3	2		
969	956	1.00	1.01	1.01	-14	3	3		
990	959	1.51	1.56	1.03	-539	3	3		
1269	946	0.84	1.13	1.34	-125	3	3		
1001	936	0.98	1.05	1.07	-47				
916	976	1.19	1.12	0.94	-113	3	3 eller 2		
1021	976	0.92	0.96	1.05	40	2	1		
1142	1117	1.80	1.84	1.02	-937	2	2		
913	977	0.96	0.89	0.93	104				
1005	969	1.19	1.23	1.04	-232				
	-195								

ANALYSIS, TIME

	1 Real	Adjusted value	Value - avg	Squared		1 Real	Real Value	Value - avg	Squared		Hypothesis testing		
	302	151	11.5	133			302	23.1	533.2		H0: Real = Screen		
	208	104	-35.5	1257			208	-70.9	5028.1		H1: Real /= Screen		
	505	252.5	113.0	12779			505	226.1	51117.1				
	179	89.5	-50.0	2495			179	-99.9	9981.8			27.6	avg1 - avg1
	453	226.5	87.0	7577			453	174.1	30307.6		z	1.49	
	213	106.5	-33.0	1086			213	-65.9	4344.0		Probability h1 is true	0.9312	
	197	98.5	-41.0	1677			197	-81.9	6709.1				
	258	129	-10.5	109			258	-20.9	437.2		Hypothesis testing		
	186	93	-46.5	2158			186	-92.9	8632.1		H0: Real = Oculus		
	393	196.5	57.0	3254			393	114.1	13016.7		H1: Real /= Oculus		
	174	87	-52.5	2751			174	-104.9	11005.9				
Sum	3068			35278		Sum			141112.9			33.4	avg1 - avg1
Avg		139.5		3207.1		Avg	278.9		12828.4		z	1.88	
Standard deviation				56.6		Standard deviation			113.3		Probability h1 is true	0.9699	
	2 Screen	Value	Value - avg	Squared		3 Oculus	Value	Value - avg	Squared		Hypothesis testing		
		115	3.2	10.1			112	5.9	34.9		H0: Screen = Oculus		
		84	-27.8	773.9			89	-17.1	292.1		H1: Screen /= Oculus		
		142	30.2	910.9			122	15.9	253.1				
		100	-11.8	139.7			84	-22.1	488.0			5.7	avg1 - avg1
		162	50.2	2518.2			146	39.9	1592.7		z	0.66	
		139	27.2	738.9			110	3.9	15.3		Probability h1 is true	0.7454	
		103	-8.8	77.8			97	-9.1	82.6				
		96	-15.8	250.2			109	2.9	8.5				
		104	-7.8	61.1			104	-2.1	4.4				
		95	-16.8	282.9			98	-8.1	65.5				
		90	-21.8	476.0			96	-10.1	101.8				
	Sum			6239.6		Sum			2938.9				
	Avg	111.8		567.2		Avg	106.1		267.2				
	Standard deviation			23.8		Standard deviation			16.3				

ANALYSIS, DISTANCE

1 Real	Real Value	Value - avg	Squared					Hypothesis testing		
	976	-224.4	50339.0					H0: Real = Screen		
	911	-289.4	83731.3					H1: Real /= Screen		
	1843	642.6	412981.5							
	970	-230.4	53067.4						194.9	avg1 - avg1
	1498	297.6	88587.4					z	1.59	
	1071	-129.4	16735.0					Probability h1 is	0.9441	
	983	-217.4	47247.0							
	1089	-111.4	12401.9					Hypothesis testing		
	936	-264.4	69888.1					H0: Real = Oculus		
	2054	853.6	728695.0					H1: Real /= Oculus		
	873	-327.4	107167.0							
Sum			1670840.5						231.6	avg1 - avg1
Avg	1200.4		151894.6					z	1.95	
Standard deviation			389.7					Probability h1 is	0.9744	
2 Screen	Value	Value - avg	Squared	3 Oculus	Value	Value - avg	Squared	Hypothesis testing		
	861	-144.5	20867.1		877	-91.7	8413.9	H0: Screen = Oculus		
	895	-110.5	12200.2		922	-46.7	2183.4	H1: Screen /= Oculus		
	1083	77.5	6013.3		1014	45.3	2049.6			
	969	-36.5	1328.9		956	-12.7	162.0		36.7	avg1 - avg1
	990	-15.5	238.8		959	-9.7	94.6	z	0.95	
	1269	263.5	69456.2		946	-22.7	516.5	Probability h1 is	0.8289	
	1001	-4.5	19.8		936	-32.7	1071.1			
	916	-89.5	8002.1		976	7.3	52.9			
	1021	15.5	241.7		976	7.3	52.9			
	1142	136.5	18644.7		1117	148.3	21984.8			
	913	-92.5	8547.8		977	8.3	68.4			
Sum			145560.7	Sum			36650.2			
Avg	1005.5		13232.8	Avg	968.7		3331.8			
Standard deviation			115.0	Standard deviation			57.7			

CONTROLLER RESULTS

RESULTS AND ANALYSIS, TIME TRIAL

1 Keys	Standard deviatio	3.61			2 PS3	Standard deviatio	2.85			3 Joystick	Standard deviati	2.66	
Count	23				Count	21				Count	15		
SUM	870		300.12		SUM	771		170.19		SUM	550		106.05
Avg	37.82		13.0		Avg	36.72		8.1		Avg	36.64		7.1
	Value	Value - avg	Squared			Value	Value - avg	Squared			Value	Value - avg	Squared
	33.92	-3.90	15.22			35.36	-1.36	1.85			35.58	-1.14	1.30
	34.23	-3.59	12.90			35.46	-1.26	1.58			36.62	-0.10	0.01
	35.69	-2.13	4.54			35.66	-1.06	1.12			39.76	3.04	9.25
	36.01	-1.81	3.28			32.02	-4.70	22.08			35.84	-0.88	0.77
	36.37	-1.45	2.11			35.71	-1.01	1.02			37.69	0.97	0.94
	36.83	-0.99	0.98			36.09	-0.63	0.40			36.06	-0.66	0.43
	36.99	-0.83	0.69			36.17	-0.55	0.30			36.17	-0.55	0.30
	38.68	0.86	0.74			36.33	-0.39	0.15			37.06	0.34	0.12
	42.4	4.5	20.60			36.5	-0.2	0.06			35.3	-1.4	1.98
	37.58	-0.24	0.06			37.08	0.36	0.13			32.94	-3.78	14.28
	35.06	-2.76	7.62			37.63	0.91	0.83			31.27	-5.45	29.69
	48.64	10.82	117.04			40.21	3.49	12.19			37.02	0.30	0.09
	46.24	8.42	70.87			37.02	0.30	0.09			40.86	4.14	17.15
	39.84	2.02	4.08			44.02	7.30	53.31			42.02	5.30	28.11
	38.82	1.00	1.00			43.01	6.29	39.58			35.44	-1.28	1.63
	37.74	-0.08	0.01			39.21	2.49	6.21					
	39.07	1.25	1.56			35.7	-1.02	1.04					
	33.47	-4.35	18.93			33.03	-3.69	13.61					
	36.11	-1.71	2.93			33.55	-3.17	10.04					
	36.81	-1.01	1.02			34.57	-2.15	4.62					
	34.11	-3.71	13.77			36.78	0.06	0.00					
	37.9	0.08	0.01										
	37.42	-0.40	0.16										

RESULTS AND ANALYSIS, PRECISION

1 Keys	Standard deviatio	1.43			2 PS3	Standard deviatio	2.18			3 Joystick	Standard deviatio	2.10	
Count	12				Count	12				Count	12		
SUM	60		24.67		SUM	61		56.86		SUM	57		53.15
Avg	4.99		2.1		Avg	5.06		4.7		Avg	4.72		4.4
	Value	Value - avg	Squared			Value	Value - avg	Squared			Value	Value - avg	Squared
	5.15	0.17	0.03			8.32	3.34	11.12			2.57	-2.42	5.83
	3.17	-1.82	3.29			3.83	-1.16	1.33			2.66	-2.33	5.41
	3.15	-1.84	3.37			2.89	-2.10	4.39			3.86	-1.13	1.27
	4.03	-0.96	0.91			4.66	-0.33	0.11			6.6	1.62	2.61
	4.84	-0.15	0.02			3.02	-1.97	3.86			4.45	-0.54	0.29
	5.31	0.32	0.11			4.85	-0.14	0.02			5.17	0.19	0.03
	4.83	-0.16	0.02			6.58	1.60	2.54			8.24	3.26	10.60
	3.49	-1.50	2.24			2.78	-2.21	4.86			2.16	-2.83	7.98
	6.5	1.5	2.39			7.9	2.9	8.38			4.7	-0.3	0.09
	4.86	-0.13	0.02			3.10	-1.89	3.55			1.91	-3.08	9.46
	6.18	1.20	1.43			8.91	3.93	15.41			6.95	1.97	3.86
	8.28	3.30	10.86			3.85	-1.14	1.29			7.38	2.40	5.74
1 Keys	Standard deviatio	0.93			2 PS3	Standard deviatio	1.62			3 Joystick	Standard deviatio	3.49	
Count	12				Count	12				Count	12		
SUM	196		10.45		SUM	197		31.48		SUM	210		145.77
Avg	16.29		0.9		Avg	16.38		2.6		Avg	17.52		12.1
	Value	Value - avg	Squared			Value	Value - avg	Squared			Value	Value - avg	Squared
	15.82	-0.47	0.22			20.59	4.30	18.45			15.2	-1.09	1.20
	15.4	-0.89	0.80			15.88	-0.41	0.17			17	0.71	0.50
	15.85	-0.44	0.20			15.74	-0.55	0.31			14.53	-1.76	3.11
	15.59	-0.70	0.50			16.01	-0.28	0.08			17.35	1.06	1.11
	18.36	2.07	4.27			15.66	-0.63	0.40			17.49	1.20	1.43
	15.59	-0.70	0.50			16.11	-0.18	0.03			17.74	1.45	2.09
	16.09	-0.20	0.04			15.06	-1.23	1.52			15.21	-1.08	1.18
	15.99	-0.30	0.09			16.01	-0.28	0.08			15.49	-0.80	0.65
	17.3	1.0	1.05			19.0	2.7	7.38			15.4	-0.9	0.84
	17.82	1.53	2.33			15.89	-0.40	0.16			17.79	1.50	2.24
	15.68	-0.61	0.38			14.64	-1.65	2.74			19.89	3.60	12.93
	16.02	-0.27	0.08			15.91	-0.38	0.15			27.18	10.89	118.50

ROV OBSERVATION

19.3.2015

Non-participant observation during an ROV mission in the Trondheimsfjord with the NTNU Minerva ROV

What I want to find out:
Can the ROV go diagonally?
Yes

How to stop?
Release controls and turn on auto-depth

Do you look in other directions than the way the ROV is travelling?
No, no cameras to the side or to the back

How much planning?
Not much, has to account for the umbilical cord not being caught into anything

Which info is most relevant?
Depth
Height over bottom
2D map
Speed
Direction (NWSE)
Sonar to see what is in front

Things to pay attention to:
Space - How is the space around organized?
Dock, container, crane, quite random

Users - Who are the users and the other people involved?
Two teachers, four other students
Frode and Stein

Activities - what do the users do?
Clear cover
Check if everything is fine by testing with controllers
Lower into water with crane

Objects - Which physical objects are there?
The umbilical cord reel
The container with the SCU
In the SCU
Several screens, big black box that receives signals from the ROV

Actions - What are unique individual actions?
Events - Is what is happening a part of a unique event?
Dekksjekk - is everything working?
Time - What is the time sequence?
Start 8.15
Set out 8.30
Finshed approx 10.00

Goal - What is the goal?
Visit all transponders

Mood - What is the mood of the user and the other individuals?
Relaxed

Notes:
Tre kameraer - et HD nede, vanlig oppe, og på arm.
5 or six thrusters, double thruster
LBL - now
USBL

Stereoscopic camera
Lyset oppe gir backscatter i kamera oppe.
Dårlig oppdateringsfrekvens

Kabelen napper opp i pitch
Fra land-slepe kabel
Autodybde og autoheading
Kan trykke inn koordinater
Går rett opp etter å kjørt til en transponder for å ikke knyte kabelen
Corden hang seg fast, fant det ut dagen etterpå.

Går rett ned når det er 30 meter igjen
Sjekket avstand med musen på skjermen
Propellene bak står litt til siden for å få bedre gjennomstrømning,
de brukes også for å rotere ROVen
Gammelt styresystem for arm, er veldig vanskelig å bruke.
Styreboksen er ganske lite kompleks i forhold til andre.

Sitater:
Greit å ha en visuell referanse å kjøre etter - Pilot om bunnen

Den mini ROVen er dritvanskelig å kjøre - student

Det hjelper med 100 timers trening - Pilot

Det verste med å treffe bunnen er at du ødelegger sikten din - Pilot

INTERVIEW

SEMI STRUCTURED INTERVIEW

Interview template made prior to travelling to Oceaneering in Stavanger. The questions were used during a phone interview with an ROV shift supervisor as well.

The players
Who are involved during a mission? Who "orders" the mission? Are the ROV operators hired by Oceaneering? Do Oceaneering deliver equipment? Are there people from different companies in the SCU at the same time? Have they met before?

Planning
Who plans? Is the ROV pilot included in the planning? How many ROVs at the same time and how many vessels? Do you set specific times for each task during a mission? How limited is the weather window? Do you use simulation tools or VR tools to plan? Or only maps?

Navigation/Mission execution
How do use depth information? Do you know the depth of the targets? Do you need exact numbers or is visual representation enough?
Do pilots of different ROVs communicate with each other?
Who are present during a mission? The same ones who plan it?
Do you change plans during a mission? Do you "wing" it?

Controls
Do you only use a two-joystick system? Or one as well? Have you looked at other types of controls? Do pilots suffer from stress in neck/shoulders/any other parts?

Debriefing
Do you rate the successfulness of a mission? What do you learn from one mission to the next? Is there often something you wish you could have done differently?

Users
Who are the users? Ages? Gender? Background? Training? Experience? (In hours) Their routine? Hours on/off? Rest? Mission durations? Food?

NOTES

Notes from the research trip to Oceaneering in Stavanger.

Operations from vessels are mainly for mounting and service, and can have two work class ROVs and one or two observation ROVs. Operations from rigs will have one work class and one observation ROV. There are three persons per work class ROV: one pilot, one co-pilot and one supervisor. An observation ROV is operated by one person. 600 m cable is not uncommon.

A garage is often used to protect the ROV to deep operations. The garage is suspended by rigid steel cables, while the ROV has a flexible cable that supports its own weight. One should not put out too much cable. The cable is armed with kevlar, and has a tensile strength at 7 tons. One can pull the ROV back by the cable in case of malfunction.

One square on the sonar is 10 m. They can change camera and customize the arrangement. They can also change controls according to how they like it. They use black and white images for navigation, and color images for operating on the templates, due to color codes. The color camera has a different lens, giving a more close-up effect. The cameras can be panned. The camera interface is a tablet device mounted on the chair.

A problem is the lack of depth perception. A solution is to see where the shadows are, and how they move.

The control layout should be customizable, so that working position can be varied. David likes to control with one joystick better, as he has the left hand free to do other stuff. For example, he can use a simple manipulator at the same time, and instead of doing precise maneuvering with the arm, he can position the ROV so that the position of the arm coincides with the template. You cannot do everything in one simulator program.

They do not use DP, but an acoustic system called Doppler. They can operate over a large area, and you have to be familiar with the area prior to the mission, by studying a map. "It is really important to create an image of the surroundings in your head" and "you need a sense of direction as an ROV operator". They mostly use maps prior to an operation, not during an operation. They can use the transponder system, but it only provides points showing the rov and the transponders.

They can have 3D models of the template and the installation which they can get up on the screen in order to see how it looks when the visibility is bad.

The price to rent a deep water rig per day is close to 3 MNOK per day. Every minute wasted during an ROV operation where the production is halted, will therefore cost the rig operator approximately 2083 NOK. It happens that ROVs get lost in the dark under there.

The pilots are mostly men between 30 and 40, som older, with a technical background. Mostly within automation and flight mechanics. You have to be able to work in a team, shifts are up to 12 hours (max). Communication is therefore important.

The garage and ROV is left in the deep if there is a short period of time until next operation. The kevlar is strong under tensile strength, but can easily be cut. It is therefore important to know where the cable is.

The client is often following the operation, either in the control room, another place on the rig/vessel or from an office on land. The pilots do nothing on their "own", eg make decisions for the client.

To find the pipe beneath the ocean floor, a metal detector was used with acoustic sound on the speakers. The ship follows the ROV, or it can use DP to be still. There are three zones: green, yellow, red. Green is DP is functioning, red is no DP. The pilot have to be careful not to get wires into the propellers of the vessel. Most of the times, no more than three ROVs are needed.

David sees the benefit of a system like mine. Many times, people lose track of where they are. Something which will come, are AUVs that navigate themselves to the structures and then operators take over and do the operations themselves.

Most of the times, the ROV firms get a basic, standard tasks from the operators, which require little planning in advance. You go in and out, and it's like driving a car.

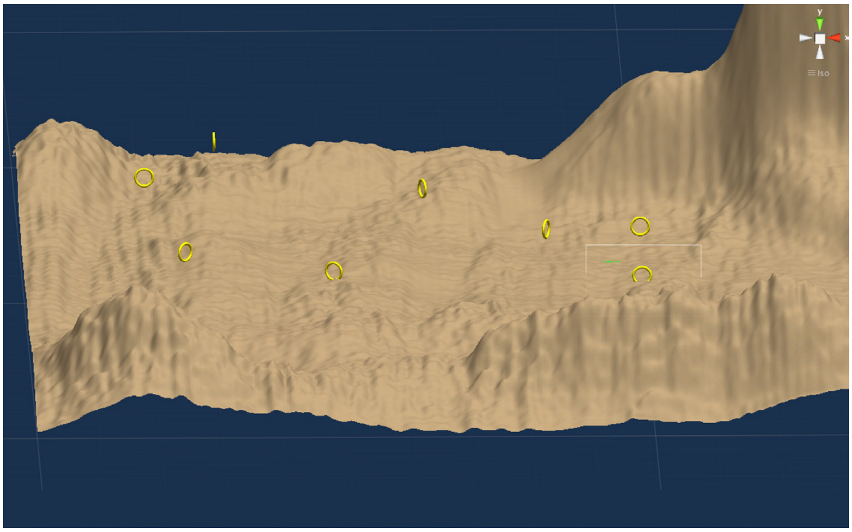
Oceaneering makes animations of some procedures to show how the job can be done, and then they might get the job afterwards. The pilots will then study the animation.

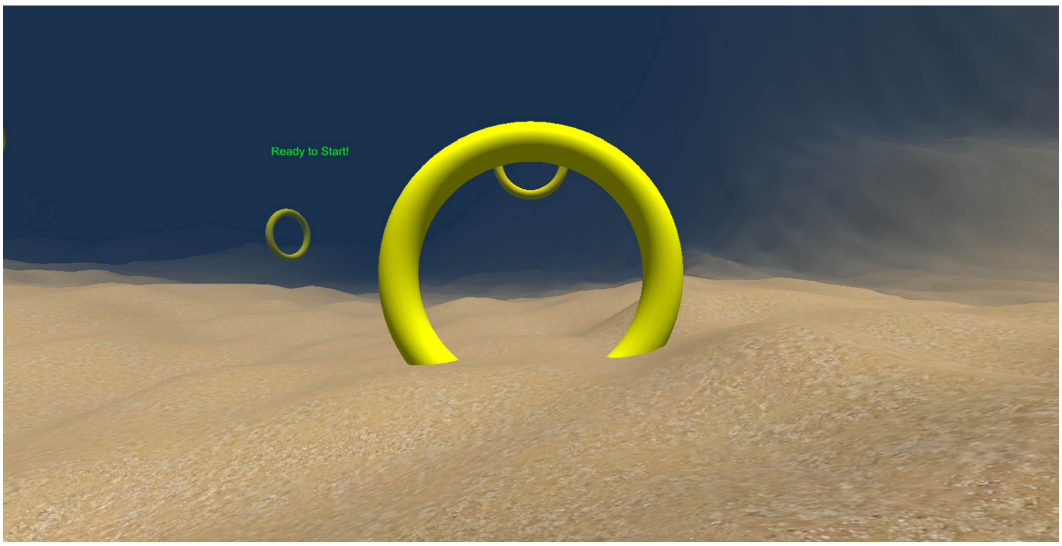
The overlay on the screen may contain a compass, depth, altitude, turns, a logo, time and date, coordinates and analog input from sensors. Not normal with simulations, but it might happen at specific jobs. 65% of the jobs are from rigs, 35% from vessels. There are little to none job related stress injuries, and the sick leave is low. The pilots have five weeks on, and four weeks of. The optimal distribution is 10 days of work each month.

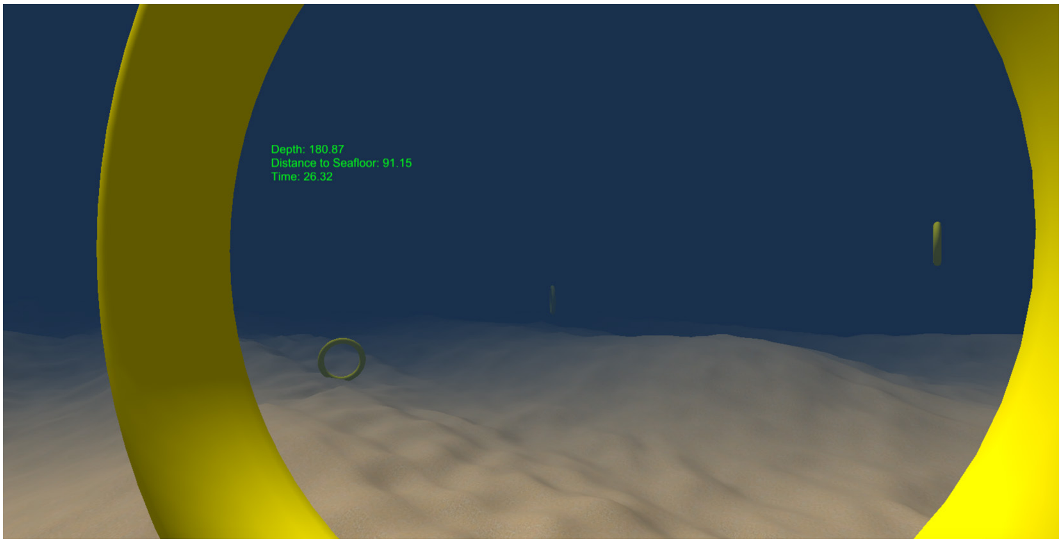
Video feeds are deleted after 24 hours, unless there are specific reasons to save them.

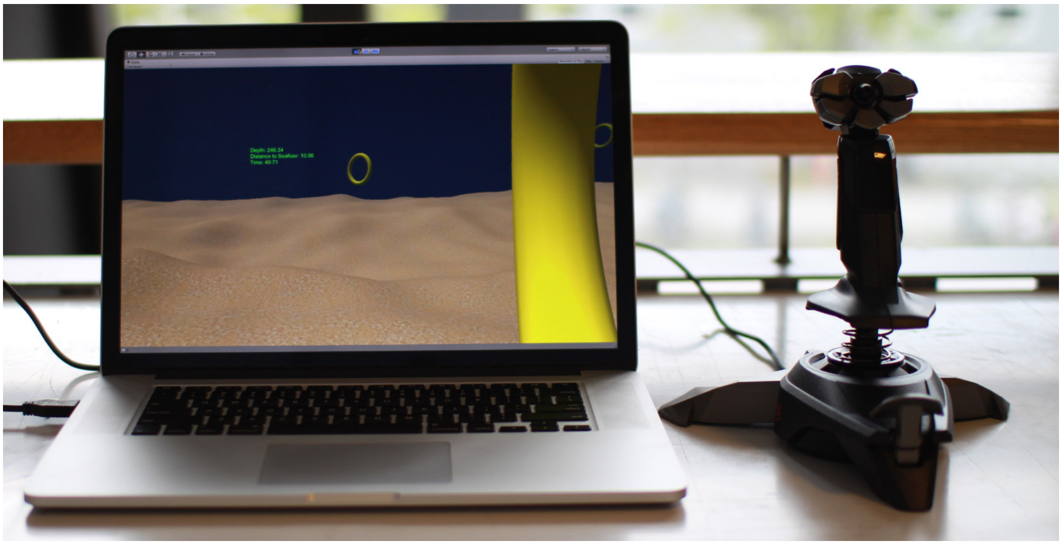
UNITY PROTOTYPES

CONTROLLER TEST, TIME TRIAL

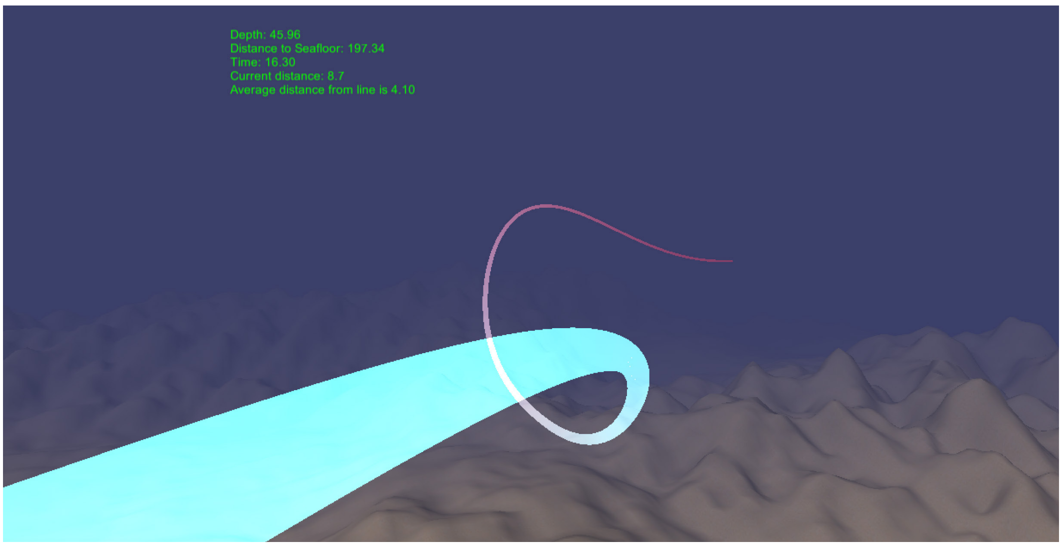


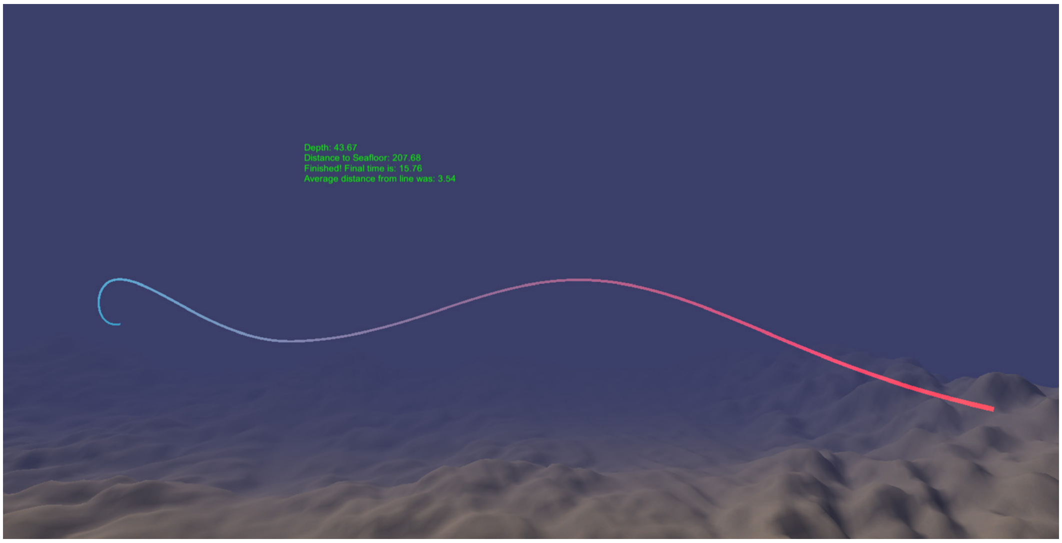


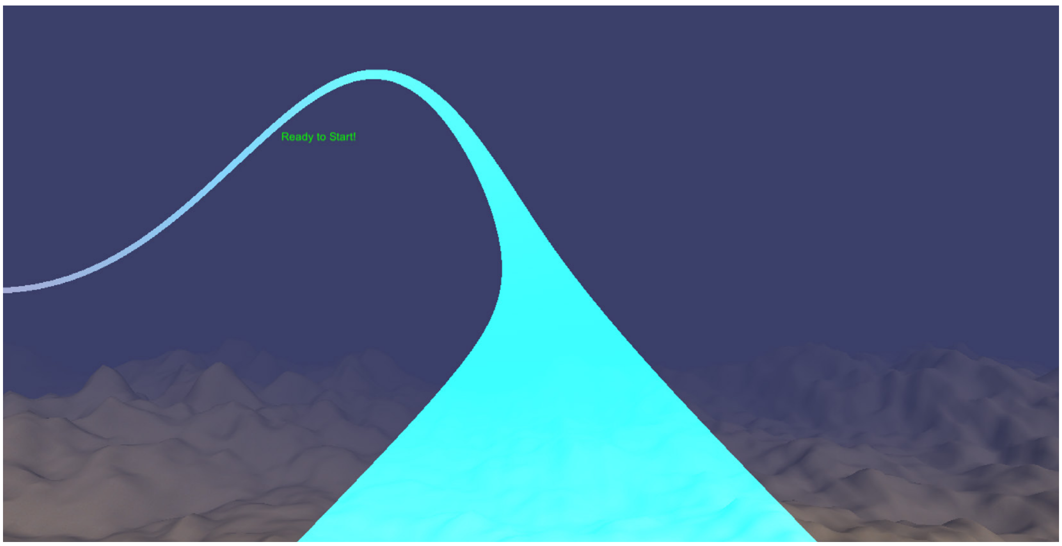

Depth: 180.87
Distance to Seafloor: 91.15
Time: 26.32



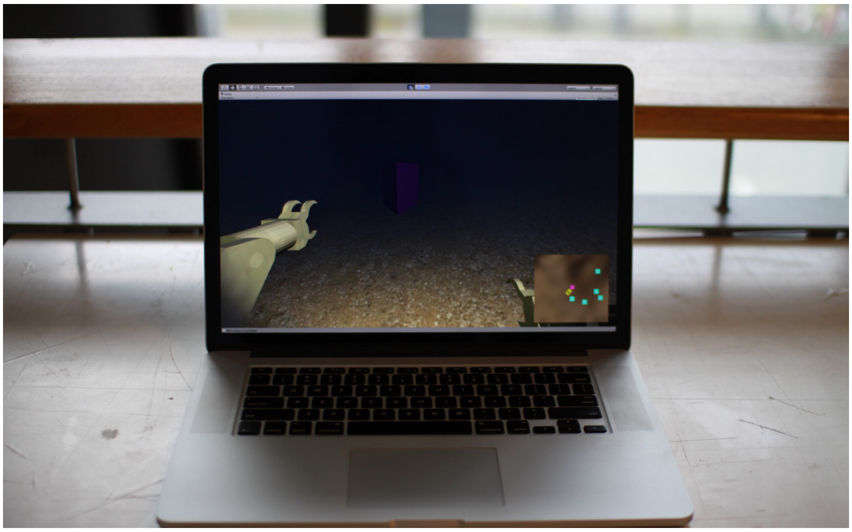
CONTROLLER TEST, PRECISION

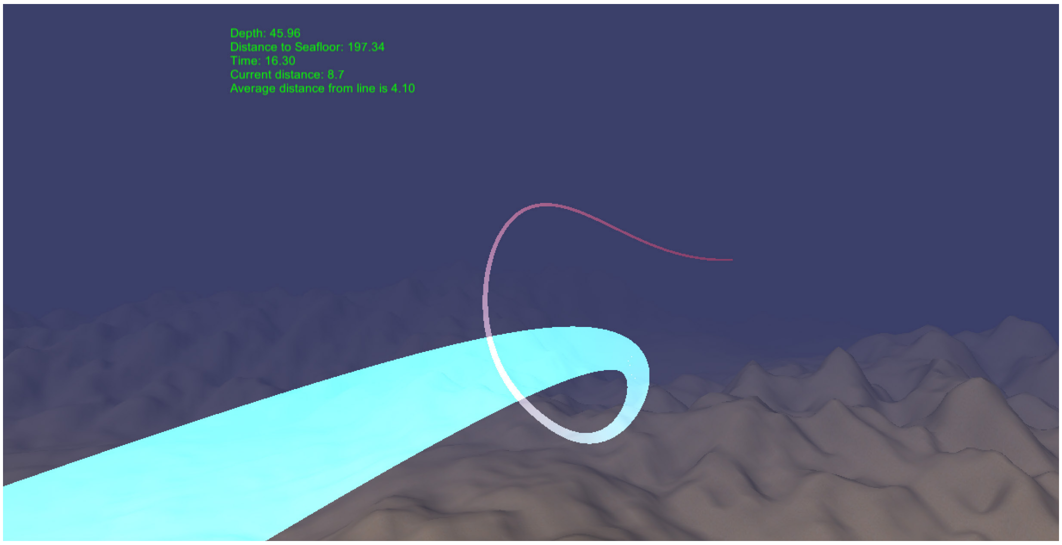



Depth: 43.67
Distance to Seafloor: 207.68
Finished! Final time is: 15.76
Average distance from line was: 3.54

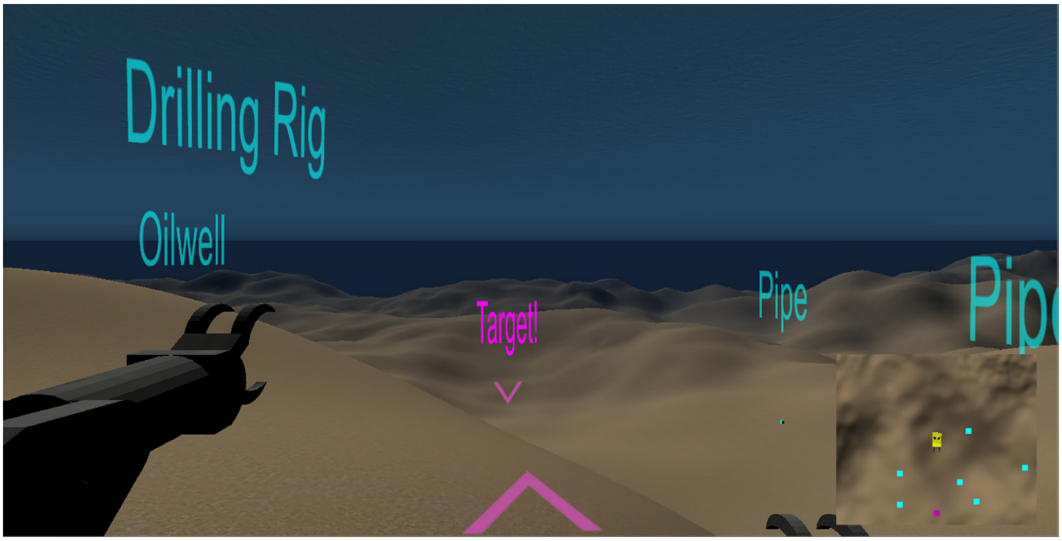

Ready to Start!

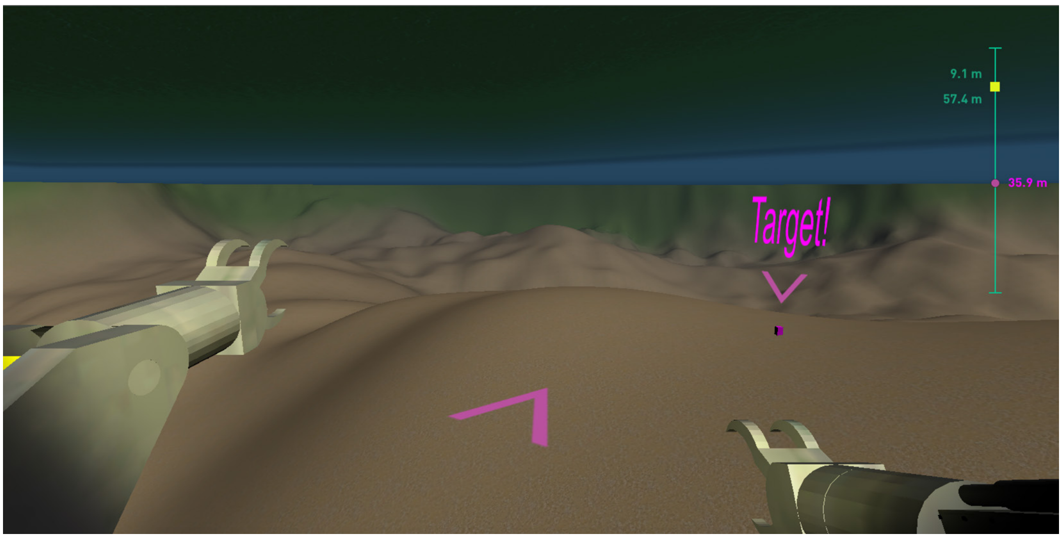
NAVIGATION TEST, REAL





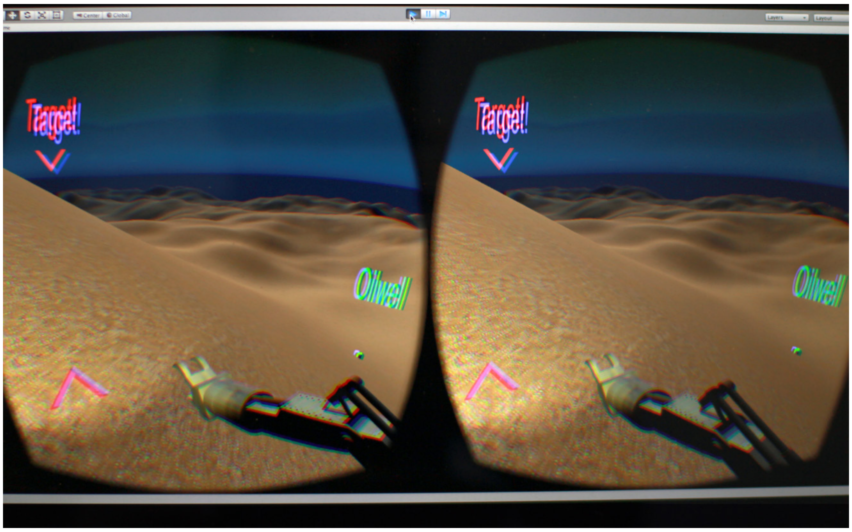
NAVIGATION TEST, SCREEN





Newer prototype that was not included in the quantitative testing

NAVIGATION TEST, OCULUS



UNITY CODE

The most important part of the code for the Navigation test. It gathers info from all the objects in the scene, and manipulates them.

```
using UnityEngine;
using System.Collections;
using UnityEngine.UI;

public class Control : MonoBehaviour {

    //*************** Declarations ************//

    // Connect to Game Objects
    public GameObject ROV;
    public GameObject WaterSurface;
    public Material Sunny;
    public Material DefaultSkyBox;
    public Camera MainCamera;
    public Camera MapCamera;
    public OVRCameraRig OVRRig;
    public Canvas TargetArrow;
    public Text depth;
    public Text altitude;
    public Text targetDepth;
    public GameObject depthViz;
    public Image rovImage;
    public Image line;
    public Image targetLine;
    public Text totalDepth;
    public Text targetName;
    public GameObject LineRenderer;
    public GameObject trail;
    public GameObject Float;

    // Default states
    bool defaultRenderSettings = true;
    bool realisticRendersettings = false;
    bool paused = true;
    bool rotatingIsRunning = false;
    bool trailOn = true;
    public static bool ROVView = false;

    // Public variables
    float seaFloorDepth = 0f;

    // Startposition for ROV
    Vector3 startPosition;
    float sumDistance = 0f;
    float speed = 9f;

    // Import the Target prefab and create a Game Object to hold the current target
    public GameObject Target;
    GameObject currentTarget;
    int currentTargetCount = 0;

    // Placeholders for text and arrow components
    TextMesh text;
    RawImage arrow;

    // Define target positions
    static Vector3 [] targetPositions = new [] {new Vector3 (50f, 62f, 85f), new Vector3 (65f, 61f, -40f), ne

    // Define target attributes
    string[] targetStrings = new string[targetPositions.Length];
    Color selectedColor = new Color (1,0,1,1);
    Color standardColor = new Color (0f, 100f, 50f, 0.6f);

    // Array to hold all targets
    GameObject[] targets = new GameObject[targetPositions.Length];

    // Timer
    float time = 0f;
    float oldTime = 0f;


    //************** Use this for initialization ***************//

    void Start () {

        // Set startposition
        startPosition = new Vector3 (2f, WaterSurface.transform.position.y + 0.3f, 2f);
        ROV.transform.position = startPosition;
        ROV.transform.rotation = Quaternion.identity;
        ROVController.speed = speed;

        // Start trail rendering
        startTrail ();

        // Enable Target arrow
        TargetArrow.gameObject.SetActive (true);

        // Target labels
        targetStrings[0] = "Rig";
        targetStrings[1] = "Oilwell";
        targetStrings[2] = "Pipe";
        targetStrings[3] = "Reef";
        targetStrings[4] = "Oilwell";
        targetStrings[5] = "Pipe";
        targetStrings[6] = "Drilling Rig";

        // Create all targets
        for (int i = 0; i < targetPositions.Length; i ++) {

            // Create target instances
            GameObject targetInstance;
            targetInstance = Instantiate (Target, targetPositions[i], Quaternion.identity) as GameObject;
```

```
            targetInstance = Instantiate (Target, targetPositions[i], Quaternion.identity) as GameObject;
            targets [i] = targetInstance;

            // Set name
            targetInstance.name = "Target_" + (i + 1);

            // Set standard settings
            resetCurrentTarget(targetInstance, i);
        }

        // Set first target as target
        currentTarget = targets [currentTargetCount];
        setCurrentTarget (currentTarget, currentTargetCount);

        // Set Main Camera is Oculus is not connected
        if (Ovr.Hmd.Detect () == 0) {
            OVRRig.gameObject.SetActive(false);
            MainCamera.gameObject.SetActive(true);
            MainCamera.depth = 1;

            Debug.Log ("Main camera is activated");
        }

        // Pausemenu is not enabled
        pauseMenu (0f);

        // Draw Line
//      for (int i = 0; i < targets.Length - 1; i ++) {
//          setLine (targets[i].transform.position, targets[i+1].transform.position);
//      }

        // Set view
        switchView ();
    }

    //**************** Update is called once per frame ********************//

    void Update () {

        // Set arrow to point at next target
        Vector3 currentTargetVector = currentTarget.transform.position;
        currentTargetVector.y = ROV.transform.position.y;
        TargetArrow.transform.LookAt (currentTargetVector);

        // Set position of Map Camera
//      Vector3 positionOfMapCamera = new Vector3 (ROV.transform.position.x, 500f, ROV.transform.position.z);
//      MapCamera.transform.localPosition = positionOfMapCamera;

        // Change current target
        bool changeInstance = Input.GetButtonDown ("Submit");
        if (changeInstance && rotatingIsRunning == false) {
            changeTarget ();
            StartCoroutine( rotateROV() );
        }

        // Change Rendersettings if ROV is above surface;
        if (ROV.transform.position.y >= WaterSurface.transform.position.y && defaultRenderSettings == true) {
            setRenderSettings(Sunny);
        }
        else if (ROV.transform.position.y < WaterSurface.transform.position.y && defaultRenderSettings == false){
            setRenderSettings(DefaultSkyBox);
        }

        // Set or reset realistic rendersettings
        bool realMode = Input.GetButtonUp ("Real");
        if (realMode == true && ROV.transform.position.y < WaterSurface.transform.position.y) {
            setRealisticRenderSettings ();
        }

        // Reset position
        bool resetPosition = Input.GetButtonDown ("Reset");
        if (resetPosition){
            resetROVPosition();
        }

        // Get ROV distance travelled
        sumDistance += ROVController.moveDirection.magnitude;

        // Get time for current run
        if (sumDistance > 0) {
            time = Time.time - oldTime;
        } else {
            oldTime = Time.time;
        }

        // Pause menu
        bool pause = Input.GetButtonUp ("Pause");
        if (pause) {
            pauseMenu (seaFloorDepth);
        }

        // Get distance and time
        bool stopRun = Input.GetButtonDown ("Stop");
        if (stopRun) {
            pauseMenu (seaFloorDepth);
        }

        // Toggle trail
        bool toggleTrail = Input.GetButtonDown ("Trail");
        if (toggleTrail) {
            hideTrail();
```

```
        }

        // Step in and out of ROV
        bool toggleFloat = Input.GetButtonDown ("Float");
        if (toggleFloat) {
            switchView();
        }

//      Debug.Log (Input.GetJoystickNames()[0]);

    }

    //******************** Coroutines *********************//

    // Coroutine to rotate ROV

    IEnumerator rotateROV (){
        rotatingIsRunning = true;

        Vector3 targetPosition = currentTarget.transform.position;
        targetPosition.y = ROV.transform.position.y;

        Quaternion targetRotation = Quaternion.LookRotation (targetPosition - ROV.transform.position);
        while (Quaternion.Angle (ROV.transform.rotation, targetRotation) > 0.3f) {
            ROV.transform.rotation = Quaternion.Slerp (ROV.transform.rotation, targetRotation, Time.deltaTime * 5f);
            yield return null;
        }
        rotatingIsRunning = false;
        yield break;
    }


    // Update GUI Overlay while in navigation mode

    IEnumerator updateGUI () {

        while (paused == false) {
            // Get depth and altitude values
            depth.text = ROVController.depth.ToString ("0.0") + " m";
            altitude.text = ROVController.heightAboveSeaFloor.ToString ("0.0") + " m";

            // Create GUI positions
            float margin = 50f; // Margin from top/bottom of screen
            Vector3 depthVizPosition = new Vector3 (Screen.width / 2f - 100f, margin, 0f);
            Vector3 rovImagePosition = new Vector3 (0f, 0f, 0f);

            // Set visualization position
            depthViz.transform.localPosition = depthVizPosition;

            // Set ROV position
            seaFloorDepth = WaterSurface.transform.position.y + 1f - Terrain.activeTerrain.SampleHeight (ROV.transform.position);
            seaFloorDepth = WaterSurface.transform.position.y + 1f - Terrain.activeTerrain.SampleHeight (ROV.transform.position);
            float heightOfROVViz = map (ROVController.depth, 0f, 140f, 10f, Screen.height - margin - 10f);
            rovImagePosition.y = 300f - heightOfROVViz;
            rovImage.rectTransform.localPosition = rovImagePosition;

            // Set lenght of depth line
            float lengthOfLine = map (seaFloorDepth, 0, 140f, 10f, Screen.height - margin - 10f);
            Vector2 lengthOfLineVector = new Vector2 (2f, lengthOfLine);
            line.rectTransform.sizeDelta = lengthOfLineVector;

            // Update the GUI for the current target
            updateCurrentTargetGUI();

            // Framerate for coroutine is 1/25f
            yield return new WaitForSeconds(1f/25f);
        }
    }

    // Update GUI Overlay while paused

    IEnumerator pauseGUI(){

        while (paused == true) {
            // Update (only) the GUI for the current target
            updateCurrentTargetGUI();

            // Framerate for coroutine is 1/25f
            yield return new WaitForSeconds(1f/25f);
        }
    }

    // Updates the GUI for the current target

    void updateCurrentTargetGUI(){
        // Make sure to keep the same as in updateGUI()
        float margin = 50f;

        // Get current target depth and set color to text
        float currentTargetHeight = Terrain.activeTerrain.SampleHeight (currentTarget.transform.position);
        float heightOfWaterSurface = WaterSurface.transform.position.y + 1f;
        float currentTargetDepth = heightOfWaterSurface - currentTargetHeight;
        targetDepth.text = currentTargetDepth.ToString ("0.0") + " m";
        targetDepth.color = selectedColor;

        // Set lenght of target line
        currentTargetDepth = map (currentTargetDepth, 0f, 140f, 10f, Screen.height - margin - 10);
        Vector2 lengthOfTargetLineVector = new Vector2 (2f, currentTargetDepth);
        targetLine.rectTransform.sizeDelta = lengthOfTargetLineVector;
    }
```

```

//******************** Functions ********************//

// Change view / go in and out of ROV

void switchView(){
    ROVController.enabled = !ROVView;

    Float.gameObject.SetActive (ROVView);
    Float.transform.position = ROV.transform.position - Vector3.forward * 20f;
    Float.transform.LookAt (ROV.transform.position);

    ROVView = !ROVView;
}

// Change target

void changeTarget (){

    // Reset current target
    resetCurrentTarget(currentTarget, currentTargetCount);

    // Update target count
    currentTargetCount += 1;
    if (currentTargetCount >= targets.Length){
        currentTargetCount = 0;
    }

    // Set next target as current target
    currentTarget = targets [currentTargetCount];
    setCurrentTarget (currentTarget, currentTargetCount);
}


// Reset current target by disabling arrow, resetting label and color

void resetCurrentTarget (GameObject target, int count){

    if (!realisticRendersettings) {
        arrow = target.GetComponentInChildren<RawImage> ();
        arrow.enabled = false;
        text = target.GetComponentInChildren<TextMesh> ();
        text.text = targetStrings [count];
        text.color = standardColor;
    }

    target.gameObject.renderer.material.color = standardColor;
    Transform mapCube = target.transform.Find ("Cube");
    mapCube.renderer.material.color = standardColor;
}


void setCurrentTarget (GameObject target, int count){

    if (!realisticRendersettings) {
        arrow = target.GetComponentInChildren<RawImage> ();
        arrow.enabled = true;
        text = target.GetComponentInChildren<TextMesh> ();
        text.text = "Target!";
        text.color = selectedColor;
    }

    target.gameObject.renderer.material.color = selectedColor;
    Transform mapCube = target.transform.Find ("Cube");
    mapCube.renderer.material.color = selectedColor;
}

// Change rendersettings

void setRenderSettings(Material skybox){

    RenderSettings.fog = !RenderSettings.fog;
    RenderSettings.skybox = skybox;
    defaultRenderSettings = !defaultRenderSettings;
}

// Set realistic render settings

void setRealisticRenderSettings (){

    resetCurrentTarget (currentTarget, currentTargetCount);
    realisticRendersettings = !realisticRendersettings;

    // Environment
    RenderSettings.fog = true;
    RenderSettings.skybox = DefaultSkyBox;
    if (RenderSettings.fogDensity != 0.2f) {
        RenderSettings.fogDensity = 0.2f;
    } else {
        RenderSettings.fogDensity = 0.001f;
    }

    // ROV speed
    if (ROVController.speed == speed) {
        ROVController.speed = speed/2f;
    } else {
        ROVController.speed = speed;
    }

    // GUI arrow
    TargetArrow.enabled = !TargetArrow.enabled;
```

```
        for (int i = 0; i < targets.Length; i ++) {
            GameObject targetInstance = targets[i];
            targetInstance.transform.GetChild(1).gameObject.SetActive (TargetArrow.enabled);
        }

        // Activate current target settings for current target if returning from realistic render settings
        setCurrentTarget (currentTarget, currentTargetCount);
    }

    // Enable pause menu

    void pauseMenu (float totDepth){

        // Toggle objects in pause mode. They are all enabled in paused mode.
        paused = !paused;
        totalDepth.enabled = !totalDepth.enabled;
        targetName.enabled = !targetName.enabled;
        MapCamera.enabled = !MapCamera.enabled;

        if (paused) {
            // Run the GUI for paused mode
            StartCoroutine(pauseGUI());

            totalDepth.text = "Total depth: " + totDepth.ToString ("0.0") + " m";
            targetName.text = "Target depth: ";
            targetName.color = selectedColor;
            depth.text = "ROV depth: " + depth.text;
            altitude.text = "Altitude: " + altitude.text;
            ROVController.enabled = false;

        } else {
            // Run the GUI for navigation mode
            StartCoroutine(updateGUI());
            ROVController.enabled = true;
        }
    }

    // Initialize line renderer
    LineRenderer initializeLineRenderer(int vertexCount){
        GameObject LineRendererInstance = Instantiate (LineRenderer) as GameObject;
        Color c1 = new Color (255, 255, 255, 0.1f);
        LineRenderer lineRenderer = LineRendererInstance.AddComponent<LineRenderer>();
        lineRenderer.material = new Material(Shader.Find("Particles/Additive"));
        lineRenderer.SetColors(c1, c1);
        lineRenderer.SetWidth(0.2F, 0.2F);
        lineRenderer.SetVertexCount(vertexCount);
        return lineRenderer;
    }


    void setLine (Vector3 start, Vector3 end){
        int vertexCount = 100;
        LineRenderer lineRenderer = initializeLineRenderer(vertexCount);
        Vector3 incVec = (end - start)/(vertexCount-1);
        Vector3 currentPosition = start;
        for (int i = 0; i < vertexCount; i++) {
            currentPosition.y = Terrain.activeTerrain.SampleHeight ((currentPosition)) + 2f;
            lineRenderer.SetPosition (i, currentPosition);
            currentPosition += incVec;
        }
    }

    // Generic function to map a value from one range to another

    float map (float value, float oldMin, float oldMax, float newMin, float newMax){
        return newMin + (value - oldMin)*(newMin - newMax)/(oldMin - oldMax);
    }

    // Reset ROV position

    void resetROVPosition (){
        ROV.transform.position = startPosition;
        ROV.transform.rotation = Quaternion.identity;
        sumDistance = 0;

        endTrail ();
        startTrail();
    }

    // Instantiate trail rendering
    void startTrail (){
        GameObject trailInstance = Instantiate (trail) as GameObject;
        trailInstance.transform.parent = ROV.transform;
        trailInstance.transform.position = ROV.transform.position;
    }

    void endTrail(){
        GameObject trailInstance = ROV.transform.Find("Trail(Clone)").gameObject;
        Destroy (trailInstance);
    }

    void hideTrail (){
        GameObject trailInstance = ROV.transform.Find("Trail(Clone)").gameObject;
        trailInstance.SetActive (!trailOn);
        trailOn = !trailOn;
    }

    // Get deepest point in terrain

    void getDeepestPointInTerrain (){
        float deepestPoint = 0f;
    for (int a = -1000; a < 1000; a ++){
        for(int b = -1000; b < 1000; b ++){
            Vector3 vectoren = new Vector3 (a, b, 0f);
            float deep = WaterSurface.transform.position.y - Terrain.activeTerrain.SampleHeight (vectoren)
            if(deep > deepestPoint){
                deepestPoint = deep;
            }
        }
    }
    Debug.Log ("Deepest point is " + deepestPoint);
  }
}
```