



Norwegian University of
Science and Technology

Lines in a 3D-model

Making a Line Tool for an Infrastructure
Design Software

Sindre Punsvik

Industrial Design Engineering

Submission date: June 2016

Supervisor: Trond Are Øritsland, IPD

Co-supervisor: Idar Kirkholm, Vianova Systems

Norwegian University of Science and Technology
Department of Product Design

Sindre Punsvik

Lines in a 3D-model

- making a line tool for an infrastructure design software

Trondheim, juni 2016

II

Abstract

Background

It is important to have a good software tool to plan and design infrastructure and transport projects. When various disciplines are collaborating on a 3D-model the software they use must be reliable and flexible. A fundamental tool in these softwares are the one for constructing and editing lines.

Objectives

The goal of this project is to identify what is needed for a line tool, and develop one or more concepts of how this can be implemented for an existing software.

Methods

The thesis uses a user centered design approach. User research, interviews, and validation are the basis for the final concept. By understanding the software, and the issues that come with implementing a new line tool in it, possible solutions can be explored and evaluated.

Result

Based on the insight gathered throughout the project a concept for an idealized line tool are suggested. The thesis presents a suggested implementation sequence for how this line tool could be developed. The sequence would start with a simplified, but still usable line tool. Later the tool could be developed with increased functionalities and be refined through testing and user feedbacks.

Sammendrag

This is a norwegian version of the abstract

Bakgrunn

Det er viktig å ha et godt verktøy for å planlegge og utvikle infrastruktur og samferdselsprosjekter. Når ulike disipliner samarbeider om en 3D-modell må programvaren de bruker være pålitelig og fleksibel. Et grunnleggende verktøy i en slik programvare er det som brukes til å lage og endre linjer.

Mål

Målet med denne oppgaven er å kartlegge hva som er nødvendig for et linjeverktøy og utvikle et eller flere konsepter for hvordan dette kan gjøres i en allerede eksisterende programvare.

Metode

Avhandlingen benytter en brukersentrert designmetodikk. Brukerundersøkelser, intervjuer, og validering danner grunnlaget for det endelige konseptet. Ved å forstå programvaren og problemer som kan oppstå ved implementeringen av et nytt linjeverktøy i den, kan mulige løsninger bli utforsket og evaluert.

Resultat

Basert på innsikt innhentet gjennom masteroppgaven, foreslås et konsept for et idealisert linjeverktøy. Det blir presentert en mulig implementerings sekvens for denne løsningen. Sekvensen vil starte med et enkel, men brukbar linjeverktøy. Senere kan verktøyet utvikles videre gjennom nye funksjoner og vil bli raffinert gjennom testing og tilbakemeldinger fra brukere.

Forward

This master thesis was written at the Department for Product Design at The Norwegian University of Science and Technology during the spring of 2016. The Thesis has been made in collaboration with and for Vianova Systems - A Trimble Company.

I want to give my thanks to my supervisors Trond Are Øritsland (NTNU) and Idar Kirkholm (Vianova Systems) for their help in forming this thesis.

I want to express my gratitude to each, and everyone that I've been in contact with, who have helped me get a deeper understanding of the technical issues concerning the thesis, and given me feedback on how to work with a line in a 3D-model.

I also want to thanks friends and families for their support. In particular those that have used their time to look over this thesis and given me suggestions for how to improve it.

VI

Table of Contents

Abstract	II
Sammendrag	III
Forword	V
Table of Contents	VI

1. INTRODUCTION	1
The Background of the Thesis	2
The Problem Description	4
Overview of the Report	6
The Solution	7
The Partner Company	8

2. METHODOLOGY	11
What is a Design Process	12
Methods	14
The Process	19
Meeting the Users	22

3. BACKGROUND	25
What is a Line	26
Novapoint	34
Other Programs	48
User Groups	52

4. ANALYSIS	59
The Analysis	60
Navigating the Line Tool	62
Making Lines	78
Editing Lines	92
Communicating the Lines	96
Making it Smart	105
Other Aspects	112
5. CONCEPTS	119
The Concepts	120
How to Develop it	121
The Main Vision	136
6. EVALUATION	139
Evaluation of the Concepts	140
Evaluation of the Process	142
The Future	144
REFERENCES	147
The List	148

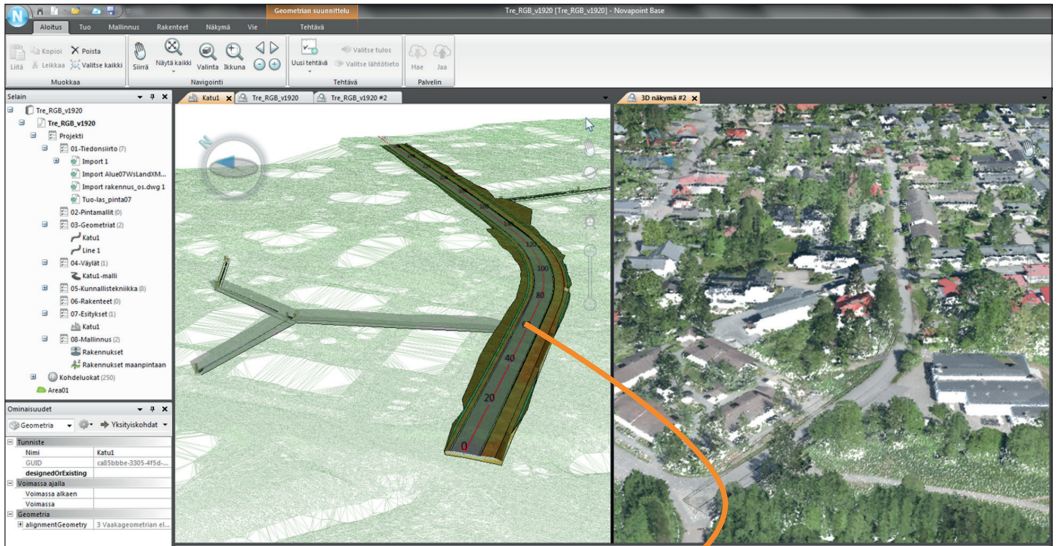
1. INTRODUCTION

An overall introduction to the thesis, its background, goal, structure and result.

The Background of the Thesis

The last few years I've been working as a summer intern at Vianova Systems, a company which main product is Novapoint, a software for planning and designing infrastructure projects. Earlier this has been a plug-in for AutoCAD, but in 2012 they launched NovapointDCM 19 and QuadriDCM, a new environment (Novapoint Base) for cooperative cross-disciplinary work on a shared 3D-model. Most of the construction and work on the shared model is still done in AutoCAD before it is exported and imported into Novapoint Base, but Vianova Systems is working on phasing more and more of these tasks into its new environment.

When the company asked if I could be interested in doing a master for them, I suggested that I could look at how we could implement a line tool in Novapoint Base. Lines are the basic element for infrastructure projects using Novapoint. When I proposed my idea for a master's thesis to Vianova Systems, the company didn't see any reason to speculate about other suggestions, as this was an important next step in their development of their software.



How to make the lines?



The Problem Description

Making og a line in a 3D-model

Vianova Systems AS is a Norwegian IT-company which main product is the software Novapoint^{DCM}. Novapoint^{DCM} is a software for planning and designing infrastructure and transport projects. Novapoint^{DCM} has previously been running in AutoCAD, but in 2012 Novapoint^{DCM} 19 was launched as a separate environment for creating 3D models of infrastructure and transport situations. However, most 3D modeling still take place in AutoCAD before being imported into Novapoint^{DCM} 19, including the drawing and constructions of lines.

The goal of this project is to identify what are needed for a line tool in Novapoint^{DCM} 19, and develop one or more concepts of how this can be implemented. The project must establish an understanding of what a line in a 3D model entails, and explore different ways of how a three-dimensional line can be constructed on a two-dimensional screen. The requirements for a line tool varies depending on the user, a line that serves as a center line of a road meets stricter requirements, compared with a line defining an area for a landscape architect.

During the project it will be of importance to find relevant design methodology and theory when focusing on various issues, applications and the mapping of different needs of various user groups. An professional reflection will also be of use when developing concepts, which will have to be evaluated by users to get valid feedback.

The goal of the project includes:

- Involvement of users, programmers and other participants
- Gathering and analyze information
- Develop ideas and concepts for a new line tool
- Evaluation of the concept(s)



Masteroppgave for student Sindre Punsvik

Lage en linje i en 3D-modell

Making of a line in a 3D-model.

Vianova Systems AS er et norsk IT selskap som står bak programvaren Novapoint^{DCM}. Novapoint^{DCM} er en programvare for planlegging og prosjektering av infrastruktur- og samferdselsprosjekter. Novapoint^{DCM} har tidligere fungert i AutoCad, men i 2012 lanserte de Novapoint^{DCM} 19 som er et eget miljø for å skape 3D-modeller av infrastruktur- og samferdselssituasjoner. Fremdeles foregår mye av 3D-modelleringen i AutoCad for så å bli importert inn i Novapoint^{DCM} 19, deriblant tegning og konstruksjon av linjer.

Opgaven går ut på å kartlegge hva som er nødvendig for et linjeverktøy i Novapoint^{DCM} 19, og utvikle et eller flere konsepter for hvordan dette kan implementeres. Opgaven må etablere en forståelse hva en linje i en 3D-modell innebærer, og utforske ulike måter en tredimensjonal linje kan konstrueres på en todimensjonal skjerm. Kravene til et linjeverktøy vil være forskjellig vil være avhengig av bruken, blant annet vil det stilles strengere krav til en linje som fungerer som senterlinjer for en vei, i forhold til en linje som avgrenser et område.

Det vil i gjennomføringen av oppgaven være viktig å finne relevant designmetodikk og teori når på ulike problemstillinger, bruksområder og ulike behov hos ulike brukergrupper skal kartlegges. En faglig diskusjon vil også være av betydning i det videre arbeidet med utvikling konsepter som skal kunne evalueres i møte med brukere for å få gode tilbakemeldinger.

Opgaven vil blant annet omfatte:

- Involvering av brukere, programmerere og andre aktører
- Informasjonsinnhenting og -analyse
- Idé- og konseptutvikling
- Konseptevaluering

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

Ansvarlig faglærer: Trond Are Øritsland
Bedriftskontakt: Idar Kirkholm

Utleveringsdato: 14. januar 2016
Innleveringsfrist: 9. juni 2016

Trondheim, NTNU, 14. januar 2016

Trond Are Øritsland
ansvarlig faglærer

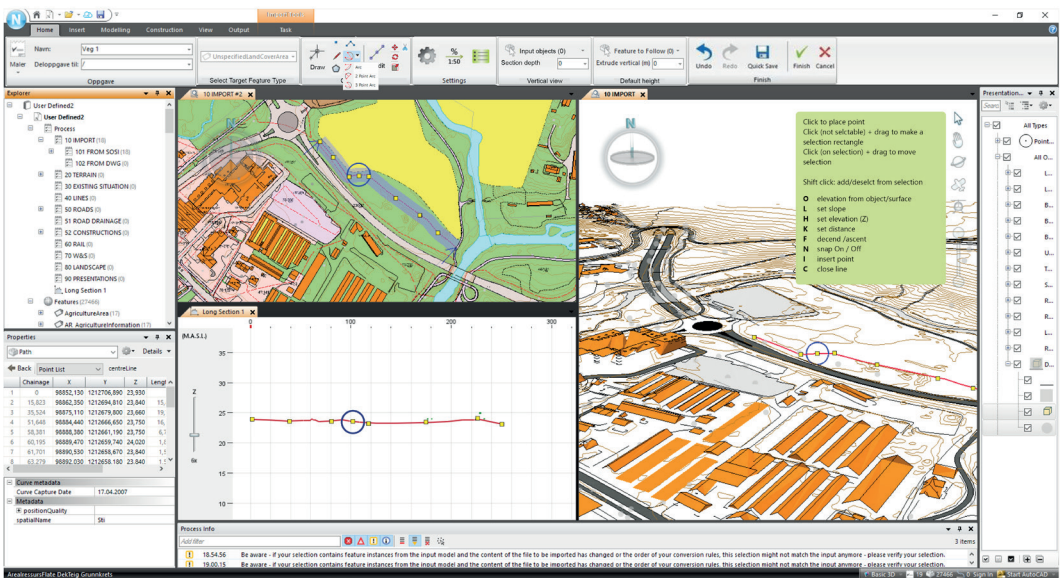
Casper Boks
instituttleder

Overview of the Report

The master's thesis report is divided into 6 parts to give a presentation of the project. The six parts are: introduction, methodology, background, analysis, concepts, and evaluation.

- 1. Introduction:** an overall introduction to the thesis, its background, goal, structure and result.
- 2. Methodology:** a presentation of the design principles and methodology used for this project, and an overview over the design process.
- 3. Background:** Presentation of the context for this project: what is a 3D-line, what software will the line tool be made for, and what are some of the alternatives on the market.
- 4. Analysis:** A presentation and analysis of the findings from the research phase of the project.
- 5. Concepts:** The main concepts and ideas for the new line tool and how it can be developed.
- 6. Evaluation:** An evaluation of the project and the resulting concept

The Solution



Example screenshot of the suggested solution where the user works dynamical with the line in a 3D-view, plan-view, and a vertical view.

The overall proposed solution of a line tool is a tool where the user easily can start drawing a line in both 2D and 3D. Using various drawing and editing tools the user can create the wanted line and always look at how it dynamically updates any tasks using the line as an input. Various disciplines will use various settings to get a tailor fit information about their work and be told when the line does not fulfill specific given criteria.

It would be impossible to make and ship a fully evolved line tool with all of the needed specifications immediately. Therefore, the project will present a suggested development plan of the line tool, where a basic polyline tool is implemented first, it will then be developed to handle more complex geometry, before it gets included in a smarter systems that affects the whole background model of the program.

The Partner Company



Vianova Systems, is the market leader in Scandinavia within civil engineering software for designing transport infrastructure. Vianova systems has since 1988 developed infrastructure design tool with a model-based, multi-discipline software vision. Today their main products are Novapoint^{DCM} and Quadri^{iDCM}. Novapoint is a design tool, which has various design applications made for different disciplines like road, landscape, water and sewerage, and railway. Novapoint is integrated with Quadri, a cloud-based building information modeling (BIM) collaboration solution. (Vianova Systems, URL)

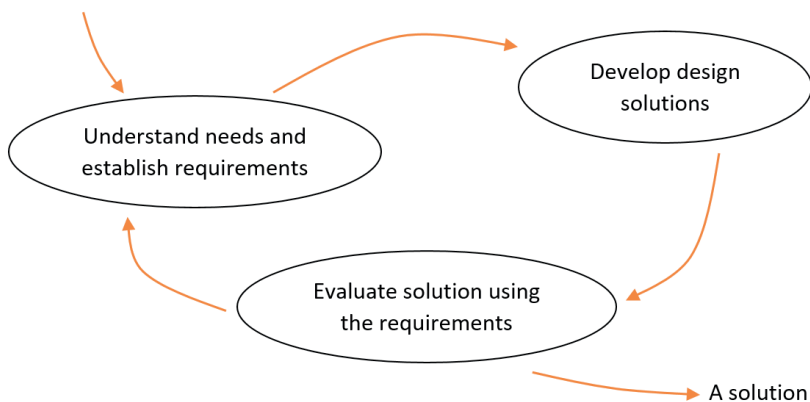
In September 2015 Trimble, an international company focusing on position-centric solutions, acquired the Norwegian company. In 2014 Trimble had 2.4 billion dollar in revenue, and the past several years it has grown by acquisitions to establish beachheads in new market spaces, fill in product line gaps, or add new technologies to their solutions portfolio. Because of this the name of Vianova Systems - will change during the fall of 2016. (Trimble, URL)

2. METHODOLOGY

the key design principles and methodology used for this project, and an overview of the design process

What is a Design Process

The design process is based on an iterative user centered design process. I've made a simple model for this based on various from others (NS-EN ISO 9241-210:2010, 2011. Sommerville, 2011. Rogers, Sharp, and Preece 2011)



Model of an iterative user centered design process

Understanding needs and establish requirements

This is to identify what the users need, how they want to work and what problems they might encounter. Furthermore, the process goes on to take a closer look at how these needs are met and what still lacks in design solutions. It is important to involve users in this stage since it is they who know best where the shoe pinches and can provide valuable insight in how they want the system to work.

Develop design solutions

Here, you try to find various possible solutions to have the system meet the needs and requirements that have been set. We can divide this section into two things: conceptual design and physical design. Conceptual design involves what the product should do, how it should look and how it should behave. The physical design goes more in depth and looking at details such as color, sound and the menus will look. To get tested models it is important to make prototypes. These can be anything from paper models to more sophisticated software-based models.

Evaluation of designs against requirements

We must continually evaluate our solutions both direct tests against specific requirements, but not least through user involvement. This can happen through specific user testing of prototypes, or talk about what they think about design solutions. After the evaluation is beginning on a new iteration, and see if you are satisfied with the requirements or if there are new needs that require to go through another round of design solutions.

Methods

This section will present some of the key design methods used throughout the project, to gather and analyze information, and develop the concepts.

Interview

Interview can also be called a focused conversation that aims to collect information. These range from unstructured conversations that acquire information through open/spontaneous questions about specific areas, to predetermined questions (Equivalent to a questionnaire), or something in between, where you use specific questions, but goes beyond them when needed to ask follow up questions. (Rogers, Sharp, and Preece 2011)

Contextual Inquiry

Is an ethnographic approach and aims to see how the user works with existing systems. The user and the interviewer has a master/apprentice relationship, where the user demonstrates and explains how he works, while interviewer asks if there are things that need further elaboration. The advantage of this is that it is easy to implement, the user gets to show how he/she works, and specific details of the process can be uncovered that would otherwise be forgotten. (Martin and Hanington, 2012)



A 3D plastic whiteboard I constructed to facilitate talking about lines in a 3D model.

Literature/Net Research

Literature studies are important to collect and synthesize information on a given topic. The intention of a literature research is to distill information from various sources, be it manuals, articles or webpages. (Martin and Hanington, 2012)

Competitive Mapping

Researching and evaluating competitors' products to evaluate how they and learn from them. By going through and looking at other products that tries to solve the same or similar problems we can draw valuable lessons about what kind of approaches are possible, and try to understand what works and what doesn't work with their solution and if this is something that could be implemented in a better way in our product. It is also about understanding the existing product and how it stands in the market (Martin and Hanington, 2012)

Prototyping

Prototype are sketches of the product being developed. In the beginning, these can be some drawings on paper, and eventually as the product progress, it can become more and more technical. The most important reasons to make prototypes is that it will help testing out ideas and make it possible for people to give a feedback on the proposed design.(Schneiderman and Plaisant, 2010)

User Testing

User testing means to have potential users test a prototype of a product, to get feedback of its usability. The tests are usually conducted by having a user try to perform various tasks with the prototype, and then various challenges with the products can be found through observations and feedback from the user. In a user testing setting it is important to make the user relax and it should always be voluntarily. It is important to take good notes of what type of challenges the user encounters during the test, sometimes a recording device can make this easier to catch, and double check the notes, in particular if there are few testers. The user should be allowed to give their own feedback on their experience, this can be given through thinking aloud during the test, by filling out a questionnaire or just speak freely about their interaction. (Schneiderman and Plaisant, 2010)

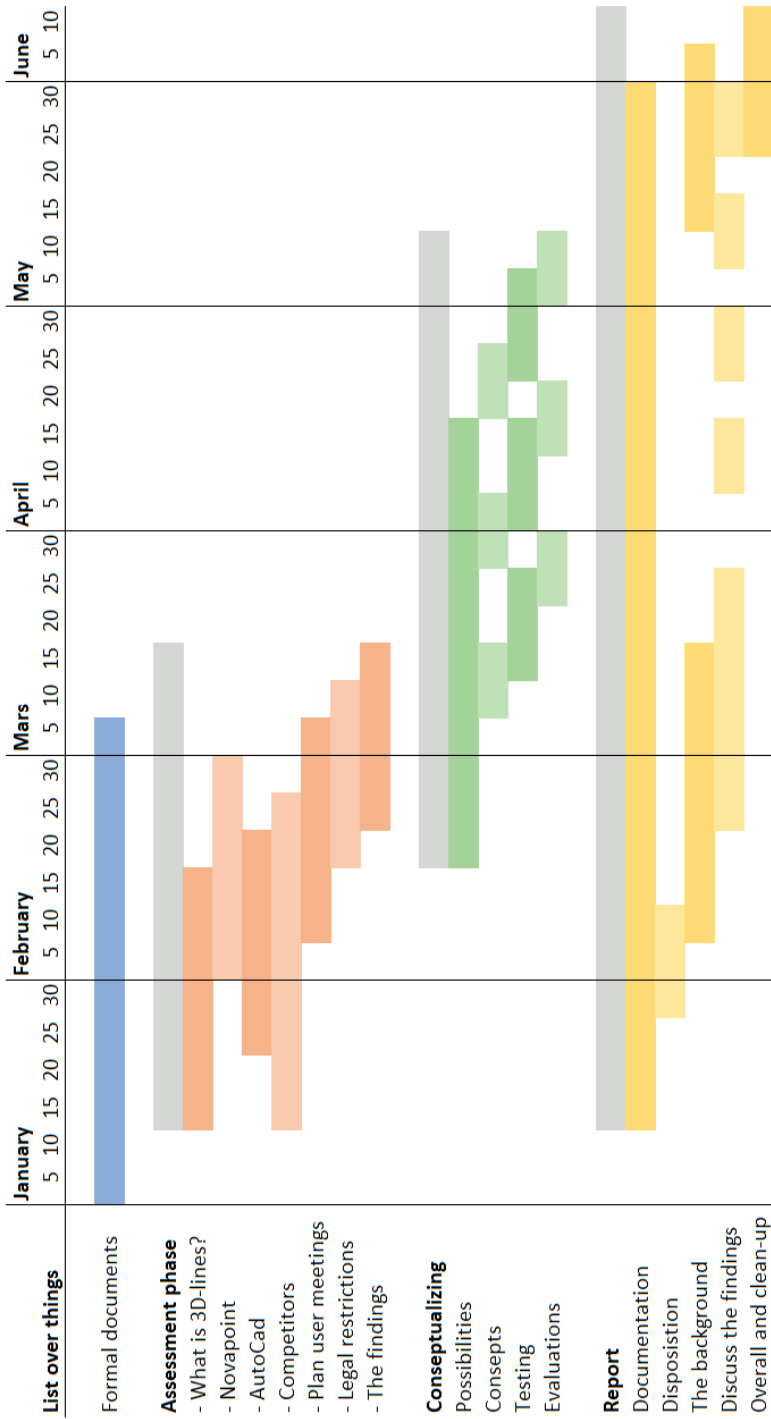
The Process

Early on in the project period I made a plan for how I wanted to work with the master's thesis. The plan was based on the principles for a user centered design process that was presented earlier. A visualized in the form of a gantt-diagram, can be seen in the illustration. The plan was divided into three phases:

1. The assessment phase
2. The conceptualization phase
3. The report phase

In the assessment phase I would do research and information gathering, through literature research, market analysis and conversations with users and programmers. The conceptualization phase would involve developing concepts and prototypes, and develop them through iterations. In parallel to these phases, I would have to document my findings and write my report.

This thesis was based on an iterative user centered design processes like presented earlier. The model of how it such a process is conducted is an idealized and simplified version. The truth is that a design process seldom goes as streamlined and straightforward as those models. The various phases merge into each other and spill over, and unexpected factors reveal themselves and influences the project. These changes means the process gets chaotic and explorative



A gantt-diagram showing the intended plan for the master

Throughout the work with the thesis it became apparent that I could try to make a concept for an idealized line tool, however this concept would be vague and would have to be thoroughly implemented and tested to make sure it would be valid. Therefore I choose to present ideas about important aspects of this idealized state, some more concrete than others. In addition, I suggested a plan for how a line tool could be develop and implemented. This implementation sequence would start with a simple, but still usable line tool, and later be developed with increased functionalities, and refined through testing and user feedback. A simple tool is better than no tool, like some of the users I've met said, they are missing a way to just draw some simple lines in the Novapoint Base environment.

When it comes to the prototype, I also met a challenge. From the very beginning of this project, I agreed to work on this with the premises that we would find someone to implement a prototype of the line tool. This to be able to conduct user-test to gain helpful insight about the direction of the line tool. At some point, I started working with one of the programmers to implement the prototype. Not being able to meet face to face, communication about this prototype development happened through mails. This gave me a basic understanding for how the Novapoint base functions are based on finite state machines (FSM). Sadly, there was a miscommunication along the line and both the programmer and I thought the other person was responsible for coding the prototype and implementing it. When this miscommunication was discovered, it was too late to fix this and make a workable prototype. Therefore, I have not had the chance to conduct user testing with on computers as I had intended from the beginning of the project.

Meeting the Users

For this project, it has been essential to get user insight to understand the needs and requirements for a line tool. I had some basic experience of how Novapoint worked from my earlier summer internships in Vianova, however I lacked a deeper understanding of the underlying principles of the software, and insight into how the target users would want to work with lines in a 3D model. I needed help understanding how a desired line tool could work.

The user groups I chose to focus on in this master's thesis are road engineers, areal planners, landscape architects, and water and sewerages engineers. These are some of the biggest user groups of Novapoint, and require a good line tool in their work.

I have talked to several people some has been in-house working for Vianova Systems, while others have been external resources which I've gotten into contact with through Vianova Systems. In-house people have been both programmers and people Vianova Systems employs that have had relevant work experiences. These people are professionals that help develop Novapoint, and give users support through courses or by finding solutions for specific problems.

When I had meetings with external people, not working for Vianova Systems, I made sure to make it clear for them that they are voluntarily helping me. To give them a framework for the meeting I made them sign a statement of consent. This statement stated that this is happening because of the master, that the participation is voluntarily, and that any records from the meeting would be kept confidential unless a new agreement with the user is made. Statement based on one from Toftøy-Andersen's book (2011).

The way I have met these people have varied, some I have just chatted unformal with about the project, while others have been more formal interview settings. Sometimes the meeting has been one-on-one, while other times I have conducted group interviews. During these meetings, we talked about how the lines are made today and their wishes for the future. I have observed how they work with current software. In addition, I have introduced them to my concept ideas to get feedback from them.

3. BACKGROUND

The context for this project: what is a 3D-line, what software will the line tool be made for, and what are some of the existing alternatives on the market

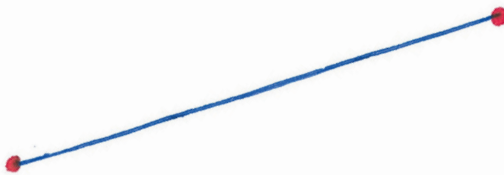
What is a Line

From the oxford dictionaries we can read that in mathematics a line is defined as: "A straight or curved continuous extent of length without breadth." For this thesis we'll be working with lines as they are presented when working with 3D-models. This usually means the data describing the line that the computer has stored, and how it is presented on the screen when the data visualizes the 3D-model.

This section will present some common ways lines can be presented and derived in 3D-models. Throughout the thesis the terms of straight line, line string, arc, transition curve and horizontal/vertical alignment will play a central role, due to already being implemented as valid geometries in Novapoint, and being the working method that most of its users are used to.

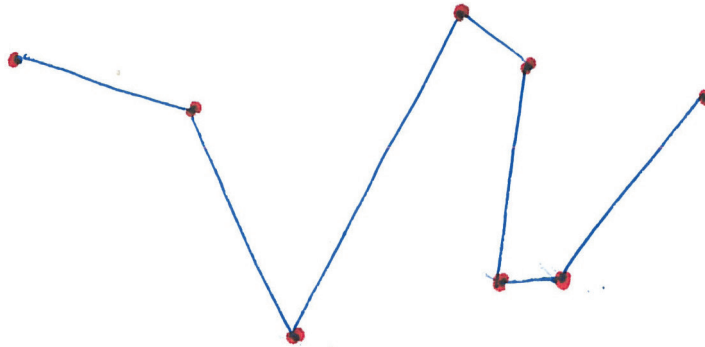
Straight line

In general a straight line can be said to be the shortest route between two points.



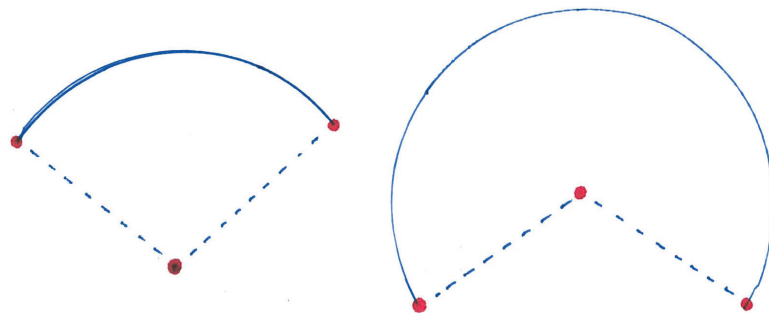
Line String

A line string is a series of straight lines. This can be defined as a list of points where the stopping point of a line is the same as the starting point of the next line in the string. Line strings are often called a polyline.



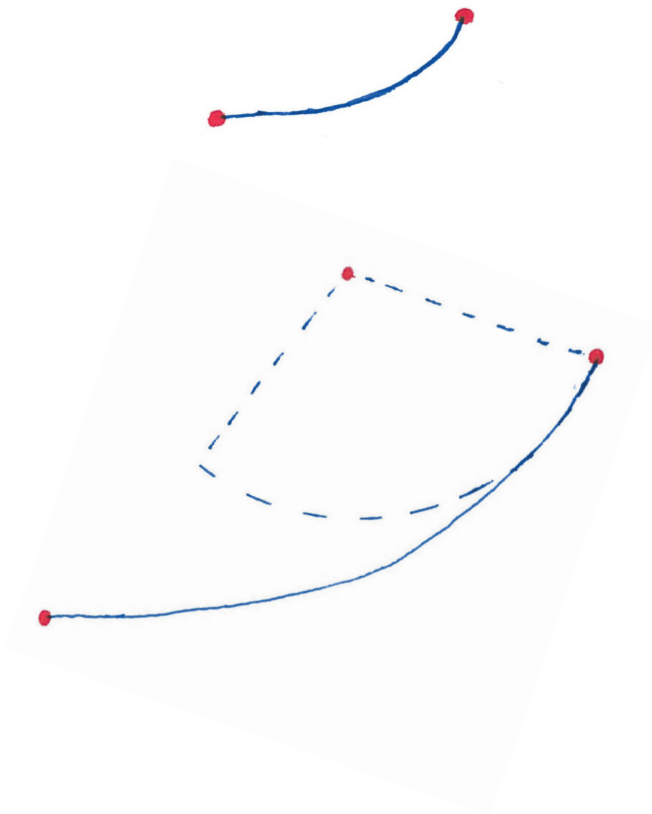
Arc

An arc is a line that has a constant curvature, it can be seen as a circle element, it can be seen as a partial circle. If the arc is defined by its start and end point, it can be of two variations when the curvature is given, either a short or a long arc. This is illustrated in the figure.



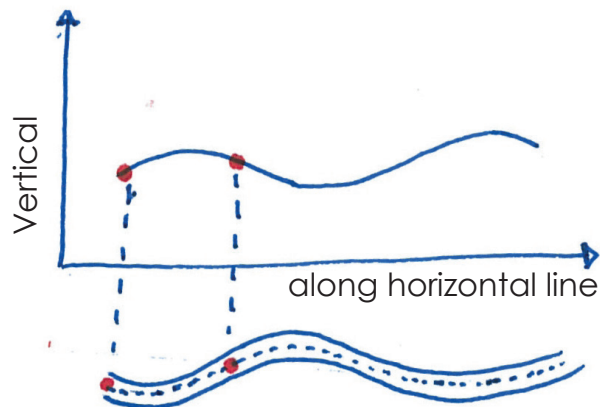
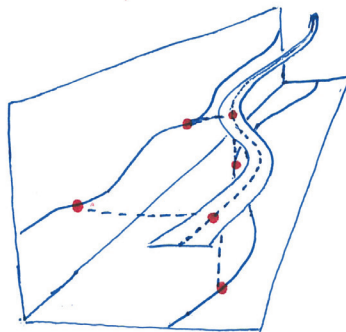
Transition Curves

Transition curves are a line which has a variable curvature going from its beginning to its end. How the curvature is changing between its beginning and end defines what kind of transition curve it is. Today, normal transition curves used for road alignments are the clothoid which are shown in the illustration. Other types that could be used are a sinus or a biquadratic transition curve.



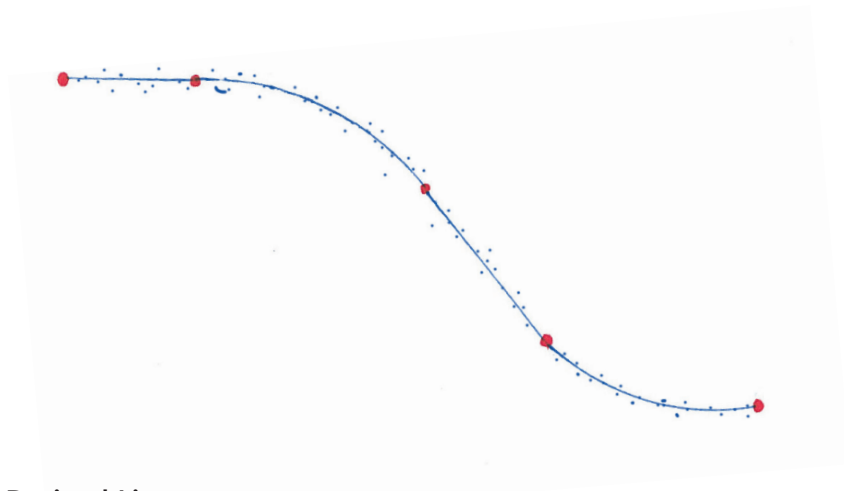
Horizontal/Vertical Alignment

To make a line in 3D, a common method is to operate with an horizontal and vertical description of the line. This technical way to construct a line is well established in fields like road constructions and architecture. By combining the data of a how a line is drawn in the horizontal plane together with how the height varies along the line the result will be a 3D line. Both the vertical and horizontal description of the line can be described as a series of the three type of line-elements described earlier: straight line, arcs, and transition curves.



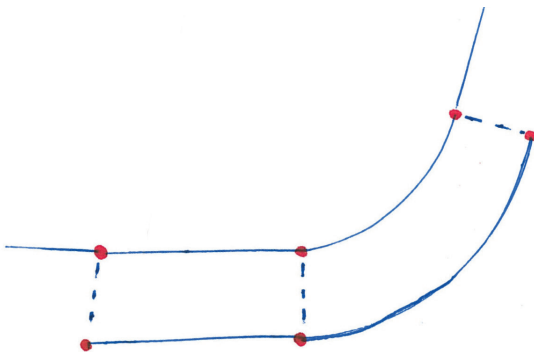
Lines from Points

Lines could also be created given a dataset of points, in which the computer calculates the line that would best represent this line.



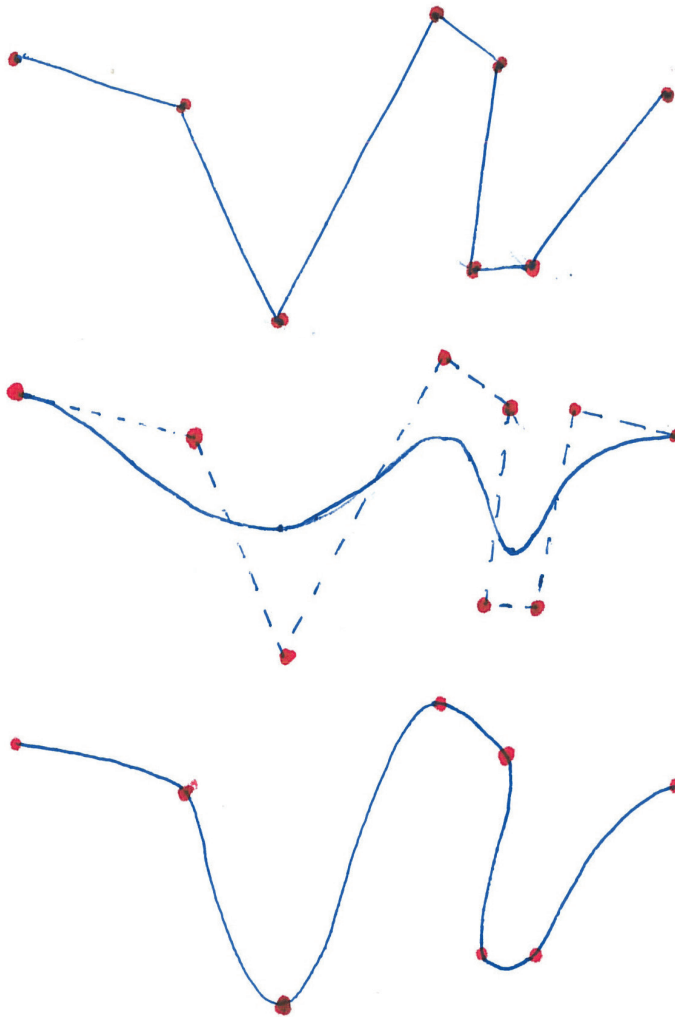
Derived Line

A derived line is a line that is made based on an already existing geometry in the 3D-model. This could be a line that is parallel or following another line with a given distance, also called an offset. Lines could also be derived from edges on existing 3D-volumes, or the intersections of 3D-surfaces.



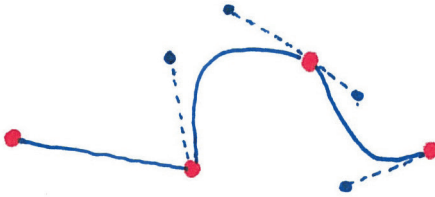
Curves Defined by Points

A normal way to make curves in other 3D modelling programs is to use points to define the curvature. Either that the curves goes through the points a so called fit-curve, or by using the points as a control frame and the points acts as control vertices.



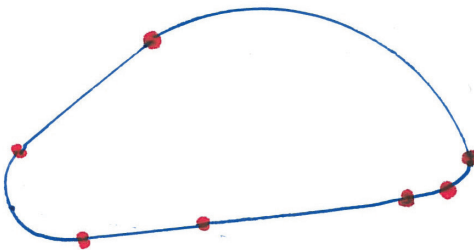
Curve made Using Handles

Another way to make curves is using handles that works as defining tangents for a curved element between the nodes.



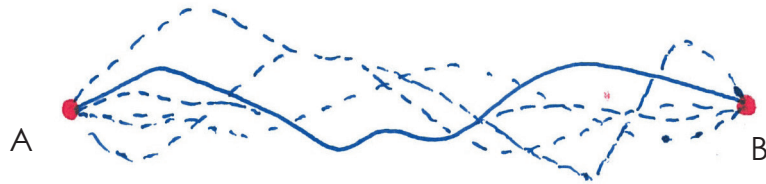
Closed curve

A closed line is a line that starts and ends in the same point. This figure is called a polygon if it only consist of straight lines.



Lines made by Optimizations

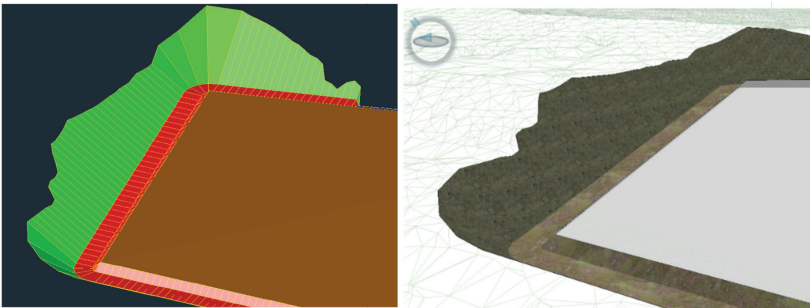
If the user knows the starting and ending points of a line, there are programs that are able to find optimized routes for the line if given the right parameters, like terrain data and which standards.



Novapoint

This sections will present some of the central terms and features of the current Novapoint software, and act as a framework for the discussion later in the report.

Novapoint^{DCM} is a software for planning and designing infrastructure and transport projects. NovapointDCM has previously been running in AutoCAD, but in 2012 Novapoint^{DCM} 19 was launched, and Novapoint Base were introduced as a separate environment for collaborating in to making and review 3D models of infrastructure and transport situations. However, most 3D modeling still take place in AutoCAD before being imported into Novapoint Base. The object of this master is to help the transition of having more functions workable directly in Novapoint Base, and will look at how lines can be made in this new environment.



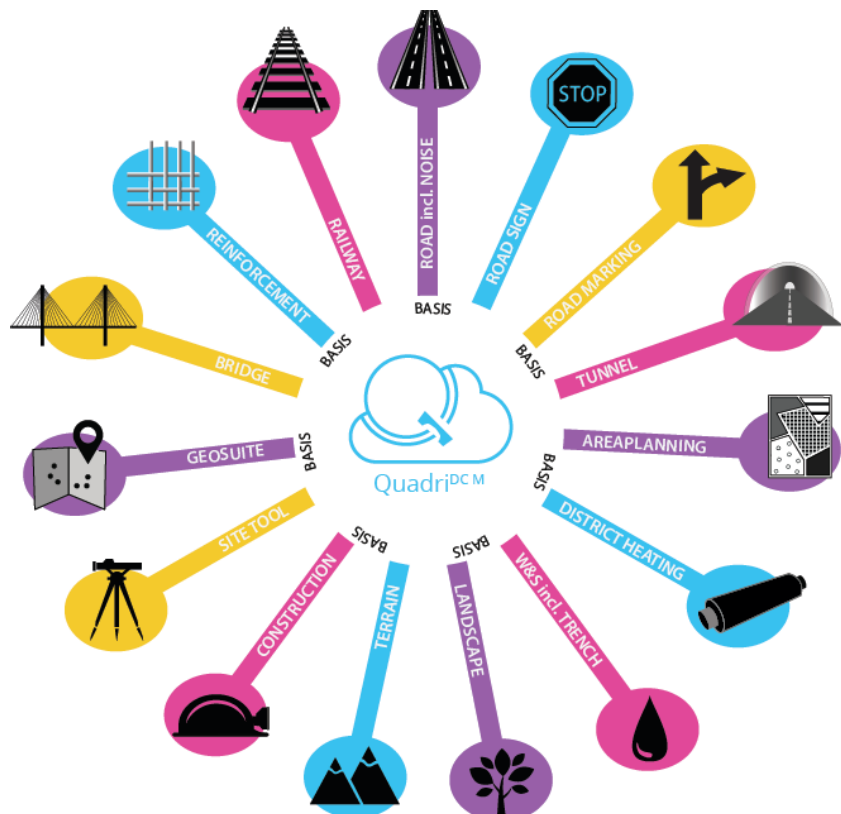
Example of a function in AutoCAD (left) and in the new Novapoint Base (right)

The main reasons to make the transition out from AutoCAD is because modern projects demands “smarter” models, and to make it easier for people to collaborate cross-disciplinary when working on the same project. This will also help Novapoint to be

more self-reliant. Two important terms regarding cross-disciplinary and “smarter” models are BIM and QUADRI

BIM: Building information modeling, the models contain more information than simply the geometry.

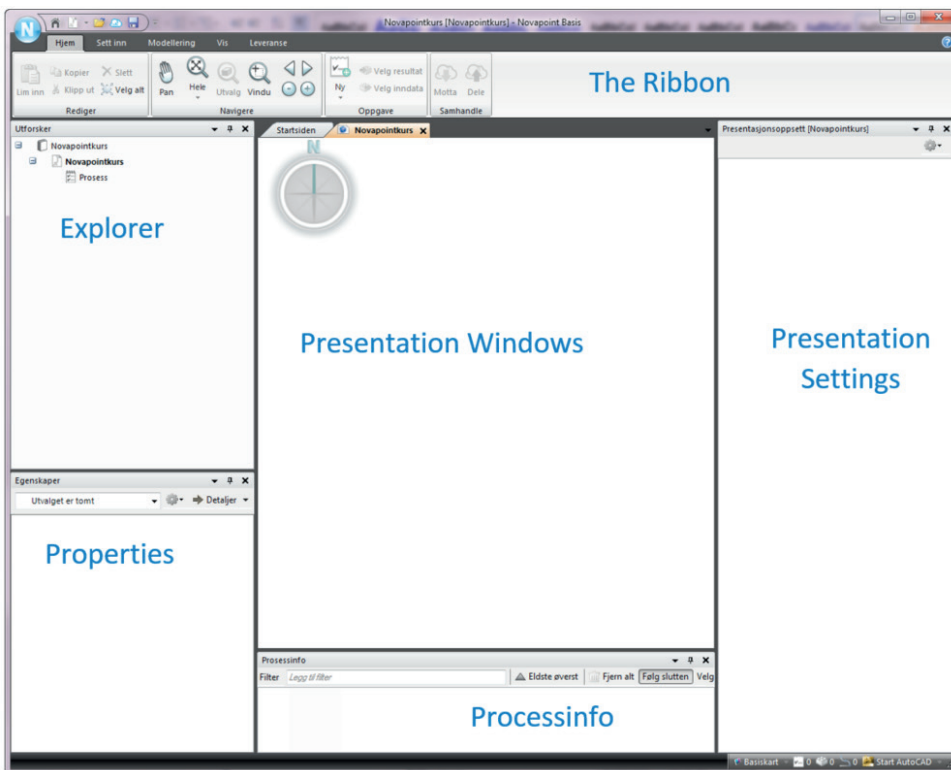
QUADRI: The system that is used to make it possible for users of Novapoint to work and collaborate on the same model.



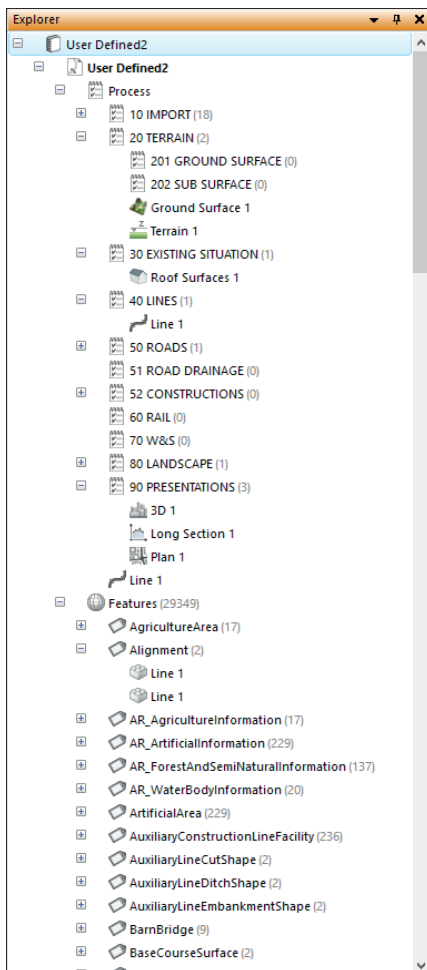
Novapoint has many modules to cater various professions that can be involved in an infrastructure project. This illustration shows some of them.

Novapoint Base

This section will give a basic presentation of Novapoint base, and how it works. The picture shows the main components of Novapoint base's user interface. What kind of windows that are shown, and how they're placed can vary since this can easily be adjusted by the user.



The main component of the user interface in Novapoint Base



The Explorer

The explorer displays a hierarchy of the Quadri model(s). The models can be divided into two main parts:

- A process tree that displays a hierarchy of tasks that belong to the project.
- Objects, listed according to its object type. These objects are made using tasks in the process tree.

Task

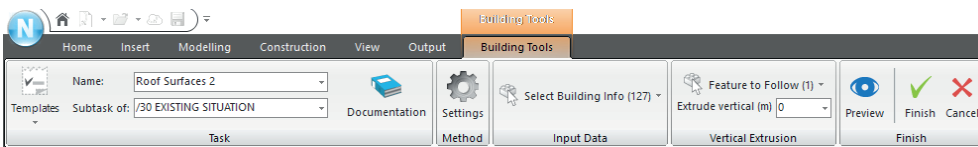
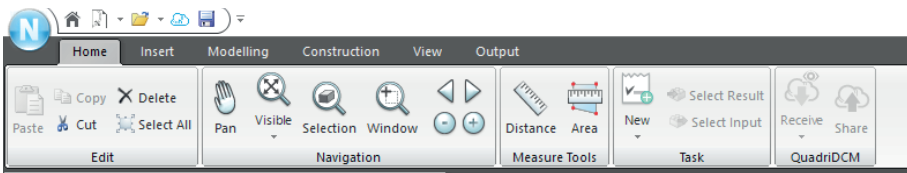
A task is something that has to be done to make an alteration in the project. Tasks can be of various categories, and for each category, there will be a tool associated to it. A task can be edited and repeated until you are satisfied with the result.

Object

An object in a Quadri model usually represents a physical object in a model with its characteristics (geometry is one of the attributes). Objects can also be without geometry, for example an event (accident, animal collisions, etc.). Each object has a feature type, and always a name, a lifecycle status (existing or planned), unique ID, and the period that the object is valid. The rest of the properties are set by its feature type.

The Ribbon

The ribbon is the main menu of Novapoint base and where the user can choose which functions to use. New specific folders will sometimes open when the user initiates various task which makes the ribbon interactive where various parameters get defined.



*The home ribbon, the modelling ribbon
and an example of a task ribbon*

Properties

This window shows the properties of the item selected. You can select Quadri model, tasks or objects. If more than one thing is selected a summary will appear. In the property window the properties can be found, and edited if it is possible. If the detail is opened, a table describing the geometry will be presented. Currently a line string can be edited here, while the horizontal/vertical alignment cannot be edited directly in this window.

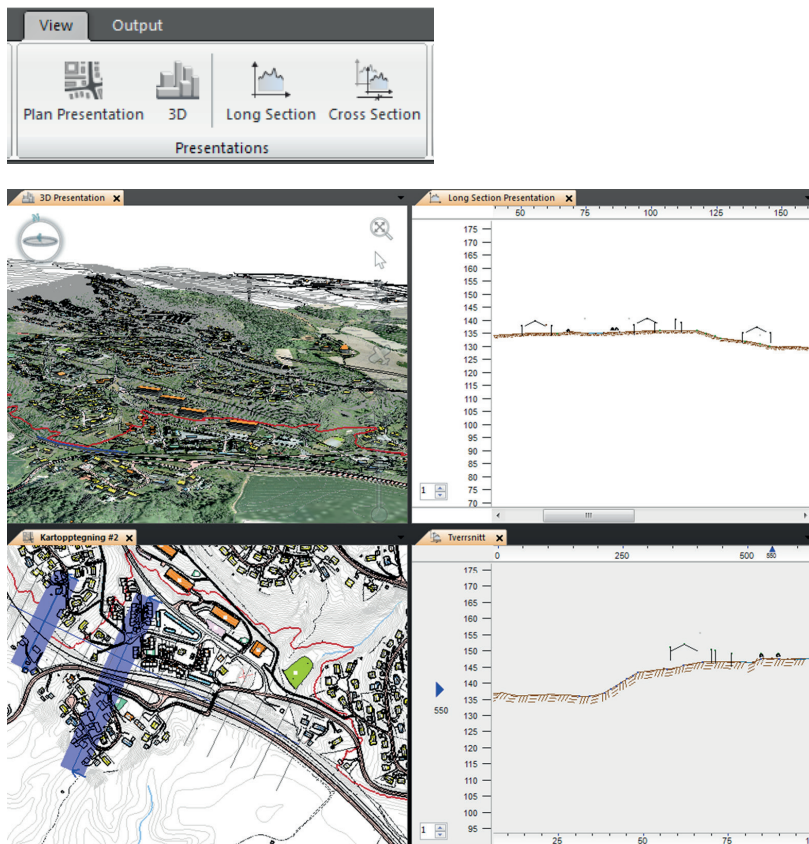
The left screenshot shows the 'Properties' window for a 'Tree' object. The 'Identification' section includes 'Name' (designedOrExisting) and 'Innmåttre' (Existing [2]). The 'Tasks' section lists 'Task (1)' (Import 1) and 'Task (2)' (Import Naturinfo u.sos). The 'Valid time period' section shows 'Valid from' (01.01.2000) and 'Valid until'. The 'Specific' section includes 'treeType' (Deciduous [3]), 'genericPropertyList' (2 elements), 'sosiobjekttype' (Innmåttre), 'TRE_TYP' (3), 'metadata' (08.05.1996 00.00, 22.03.2011 00.00), 'captureTime' (08.05.1996 00.00), 'externalIdentifier' (79), 'localId' (79), 'featuresCopy', 'copyTime' (22.03.2011 00.00), 'origin' (Bærum kommune), and 'registrationVersionSOSI' (3.4 eller eldre, FKB). The 'Geometry' section shows a 'point' with coordinates (98472,380; 1212969,400; 26,150) and 'X', 'Y', 'Z' values (98 472,380; 1 212 969,400; 26,15). It also includes 'heightReference' (Unknown [99]) and 'positionQuality'.

The right screenshot shows the 'Properties' window for an 'Alignment' object. The 'Horizontal elements' section displays a table of alignmentGeometry elements:

	Element	From	To	Radius	Length	Parameter	Start X
1	Straight Line	0,000	93,343	0	93,343		99000,910
2	Clothoid	93,343	109,343	0,000 > ...	16,000	21,909	98952,918
3	Arc	109,343	143,159	30,000	33,816		98945,964
4	Clothoid	143,159	159,159	-30,000 ...	16,000	21,909	98955,138
5	Straight Line	159,159	273,793	0	114,633		98968,825
6	Clothoid	273,793	289,793	0,000 > ...	16,000	21,909	99072,033
7	Arc	289,793	312,647	30,000	22,854		99085,720
8	Clothoid	312,647	328,647	30,000 ...	16,000	21,909	99095,881
9	Straight Line	328,647	561,396	0	232,749		99094,515

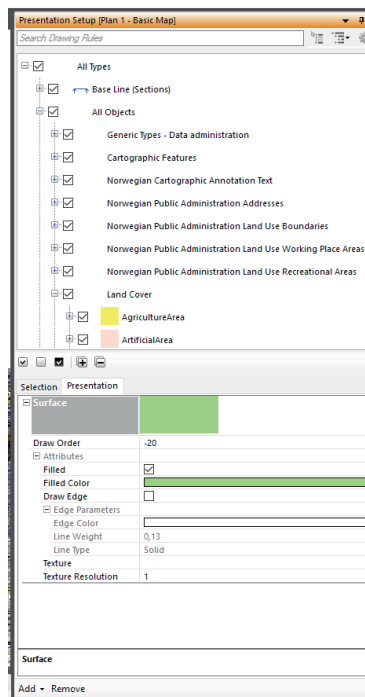
Presentation window

These windows shows a graphical presentation of the model, or parts of the model. There are 2 main ways to view the model, either as a 3D presentation, or as a plan presentation. In plan the view is from above looking down on the model. It is also possible to create a long or cross section view of the model.



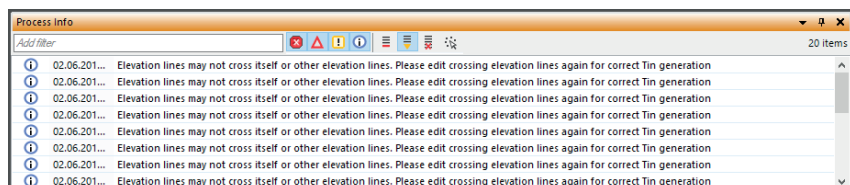
Presentation settings

Here are rules of how objects appear visually in the model, in the presentation windows. These rules can be adjusted, for example altering color, texture, line tickness, etc.



Process info

Process info displays system messages when tasks are performed. This may be errors, but also information about what is being done.

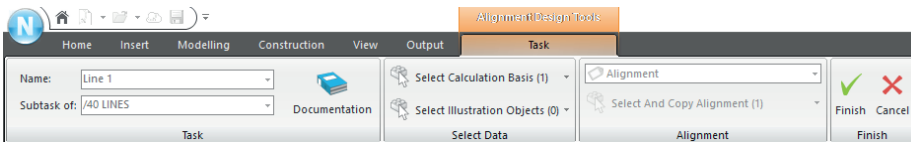


Relevant Tasks and Features

A short presentation of some existing tasks that have relevance when designing a new way to make lines.

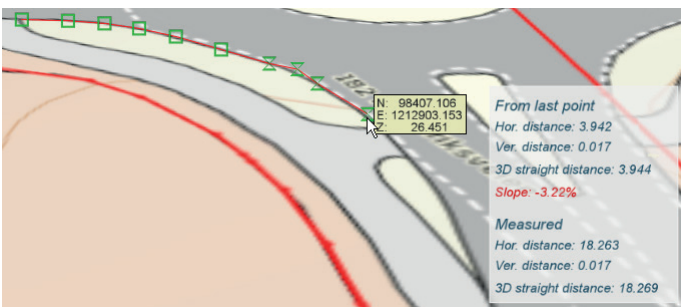
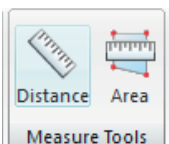
Alignment

This is the closest thing to an existing line task. Currently it is used to import or choose a line that can be used as a reference line. The task itself doesn't make a line, but can become a copy of an existing line in the model, or associated with a line that is created in AutoCAD.



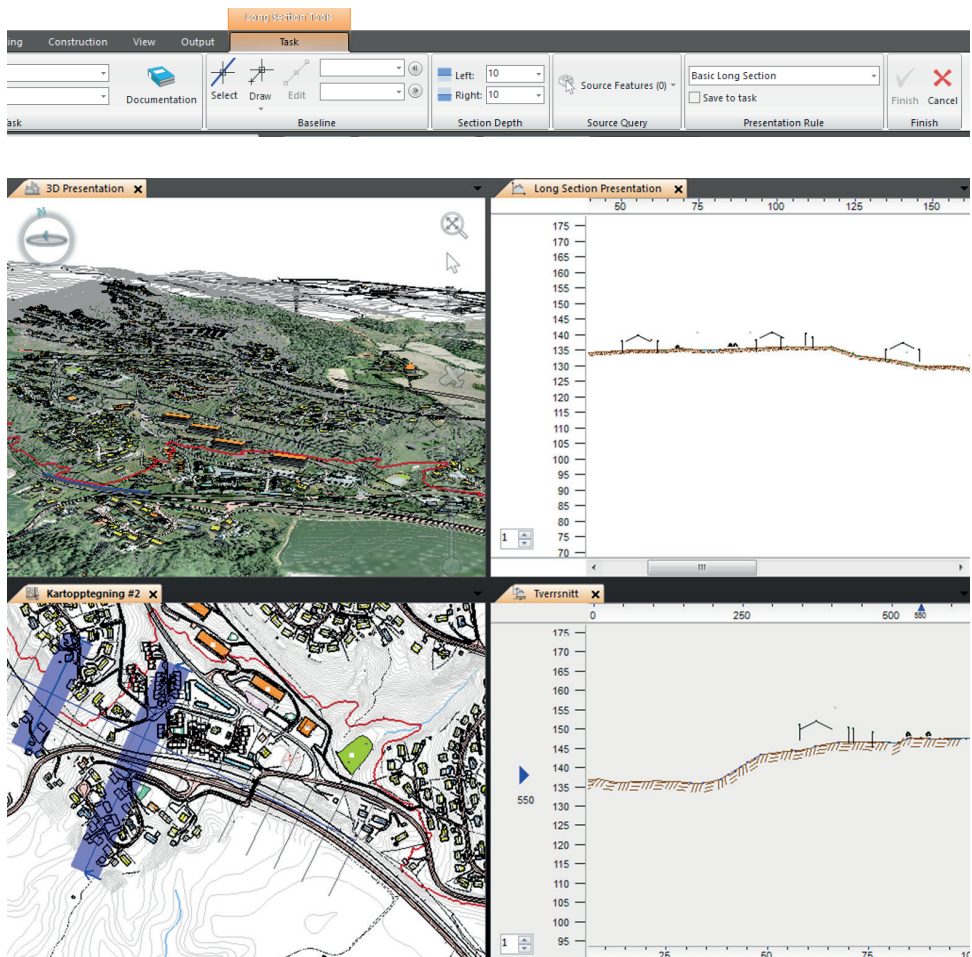
Measurements

This is one of several places in Novapoint base where the user makes simple polylines directly in Novapoint Base, but currently there are no tasks that allow for these lines to be saved as objects. This tool has a snap function that makes the point snap to existing geometry and lets the user draw the line in both the plane and 3D view window.



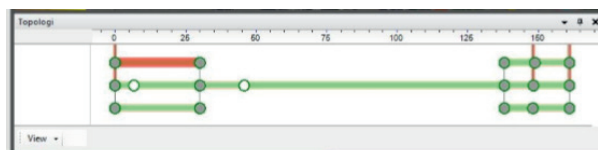
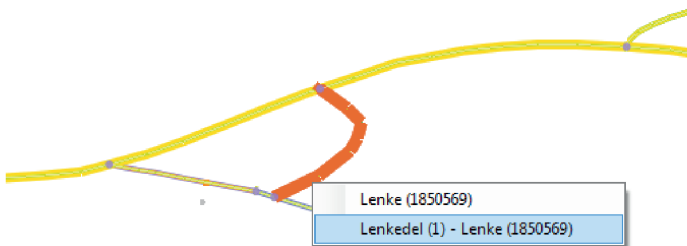
Long-/Cross Section

Gives the option for the user to select or draw a linestring to generate and see a long-/cross section view of it.



Road net

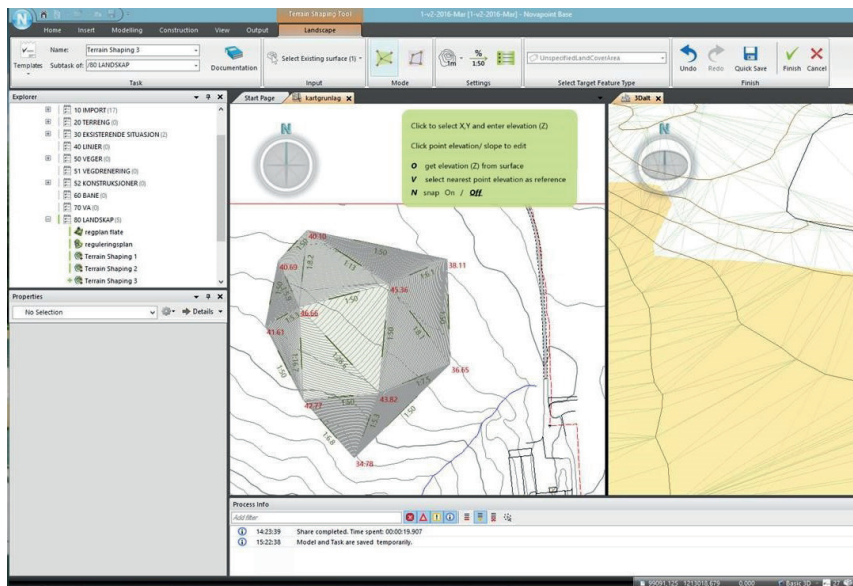
Road net is a task to work with the road network. This is one of the currently biggest and most complex tasks made for Novapoint base. It has various features that is about making lines and editing them (but only in 2D). It also includes a new window that shows a schematic overview over the road, and makes it easier to get an overview over the network.



A description of the pictures. The top shows parts of the ribbon tool for Road net, the lowermost illustrates how a menu with various tools opens up. There is an illustration of how the 2D network can look like, with the choice menu that pops up. An example of the topology window is also present.

Terrain Shaping

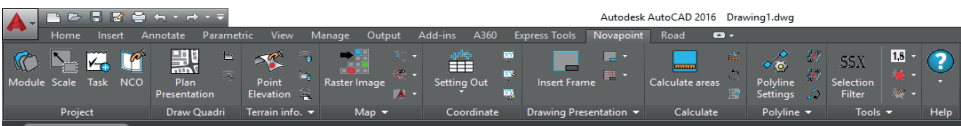
Terrain shaping is the other big editing tool that is made for Novapoint Base. The terrain is shaped using points of various elevations in the plane view. An important thing this introduces for the workflow is the widget, a way of working using mainly the keyboard. This is the only place in Novapoint where the widget is used at this current stage



An important aspect with the Terrain shaping tool is the introduction of the "Widget" menu, a list over choices that can be activated by using hot keys, and varies depending on its state.

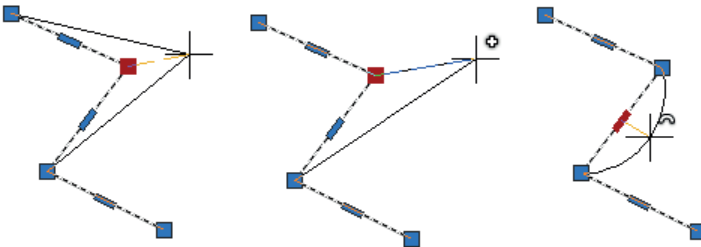
AutoCAD

AutoCAD is still the environment where most of the making and editing of elements to the Quadri-model is done. Here are some important aspects from it when working on lines



The Line Tools

The line tool in AutoCAD includes tools for drawing lines and arcs. Polylines are one important tool to create lines. When working on a polyline the user can use grips to move, add, or delete individual vertices, and convert between arc and straight segments.



*Illustration demonstrating polyline and conversion to arc
(Autodesk AutoCAD)*

Line Construction

Today Novapoint has a horizontal/vertical alignment tool that works in AutoCAD. The process is to first make a horizontal line consisting of the line-elements: straight line, arcs, and transition curves, and then create a vertical description of it.



To make the curve a separate window with a table describing the horizontal/vertical alignment and various editing tools are used. The lines can either be made and edited in the normal window or changed by adjusting the numeric values in the table.

Alignment Design Professional - 40 LINES Line 1 - Calc. basis ok

Object Edit View Insert Modify Tools Data Window Help

No.	Element	Type	Radius	Length	Param.	Bearing	Arc type	E1	M1	E2	N2
1	Line	X↔X		93.343		2.111		99000.910	1212548.850		
2	Transition	↔K	0.000	16.000	21.909						
3	Arc	↔K	30.000	33.816			Long	98931.741	1212664.240		
4	Transition	↔K	30.000	16.000	21.909						
5	Line	X↔X		114.633		0.450					
6	Transition	↔K	0.000	16.000	21.909						
7	Arc	↔K	-30.000	22.854			Long	99099.890	1212745.520		
8	Transition	↔K	-30.000	16.000	21.909						
9	Line	X↔X		232.749		1.745				99054.080	1213005.200

Input H Result H Errors H Nado In H Nado Res H Nado Ten H Ref Points History Input V Result V Errors V Nado in V Nado Res V NYLP PRF

Pick Object < Recalculate All OK - Draw Cancel

Calc OK HORIZONTAL TERRAIN Fill=0% Cut=0%

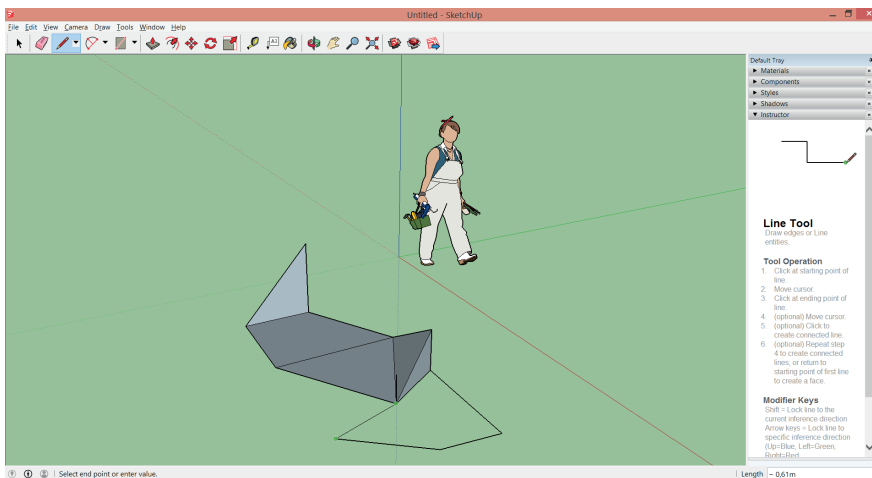
Other Programs

In addition to Novapoint I have looked at some other programs to learn from them. How do they work with lines in a 3D setting? Looking at other companies' solution provides an insight into what competing software can offer, and can give insight about possible dos and don'ts when it comes to implementing a line tool.

This section will present very shortly some of the programs that have been used as references for this thesis. Examples of how these programs works will be used throughout the analysis section.

SketchUp

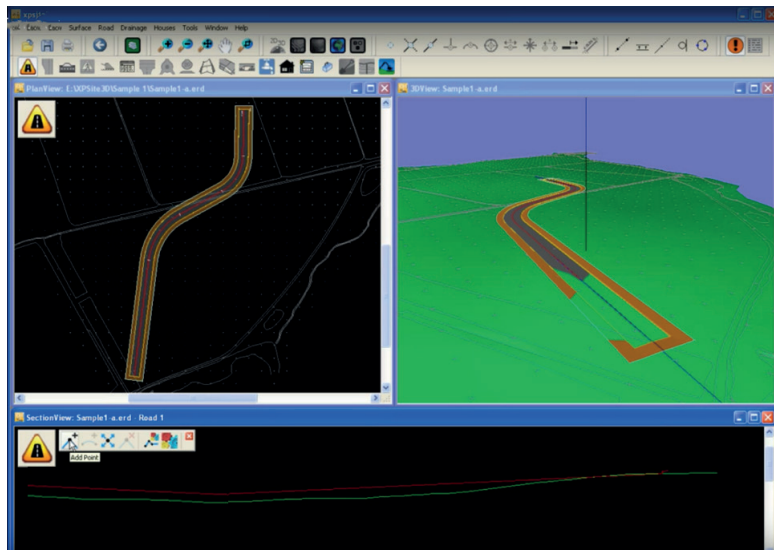
SketchUp is one of the best-known and popular 3D-modelling tools as of today. It is known for its ease of use and intuitive workflow. SketchUp was also bought up by Trimble, so it is now a sister company of Vianova Systems. Being part of the same company it would be natural for Vianova Systems to draw inspiration from the part of SketchUp that makes it successful.



Screenshot (SketchUp)

Site 3D

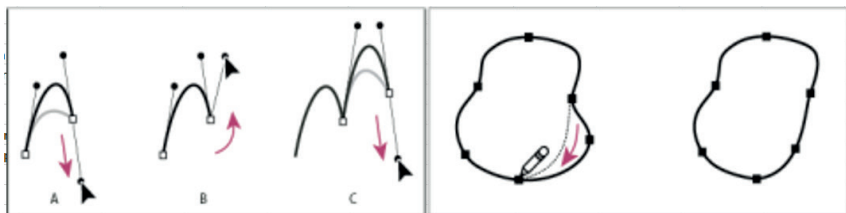
A road software that works with horizontal/vertical alignment with a 3D view that shows the resulting road.



Screenshot (Site3D)

Adobe Illustrator

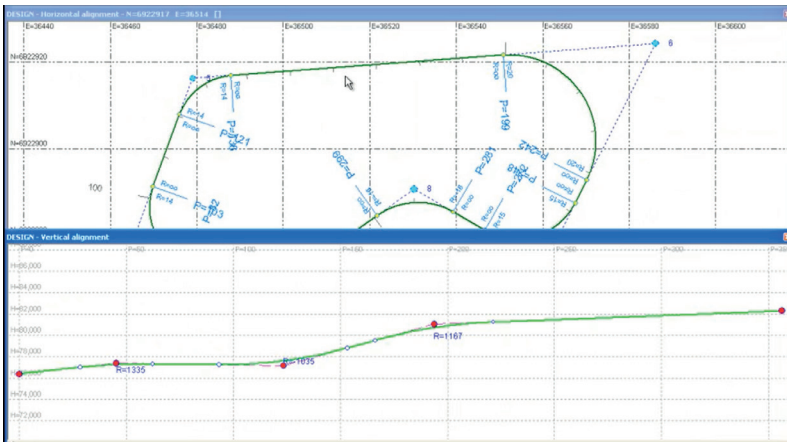
This popular drawing program was an interesting study in various ways of drawing and editing a line in 2D.



Demonstrational illustrations (Adobe Illustrator)

Prototype by Hallgeir Hjelle

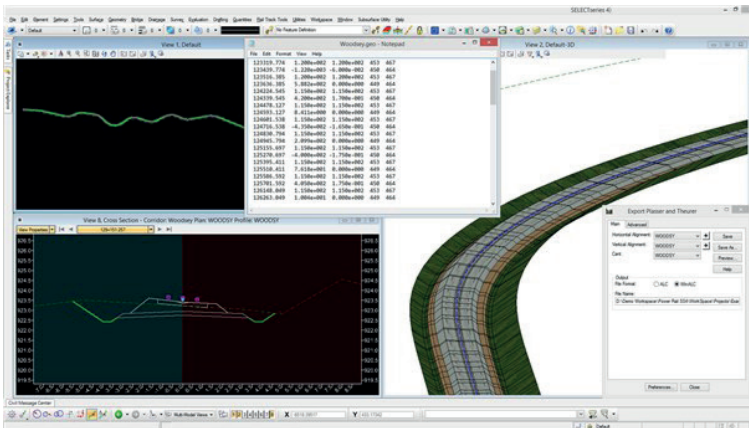
This was a prototype made by Hallgeir Hjelle to demonstrate how the lines could be dynamically updated as the line is edited.



Screenshot of Hallgeir Hjelle's prototype

Bentley 3D Rail Track

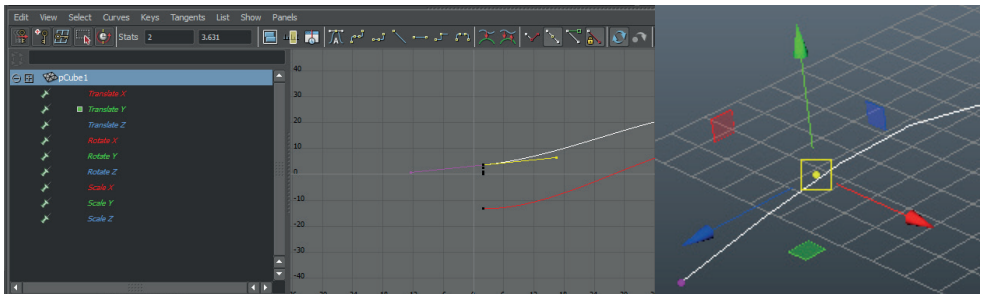
A computer program that was preferred for its possibility to make lines from a selection of points with weighted importance.



Screenshot (Bentley Rail Track)

Maya

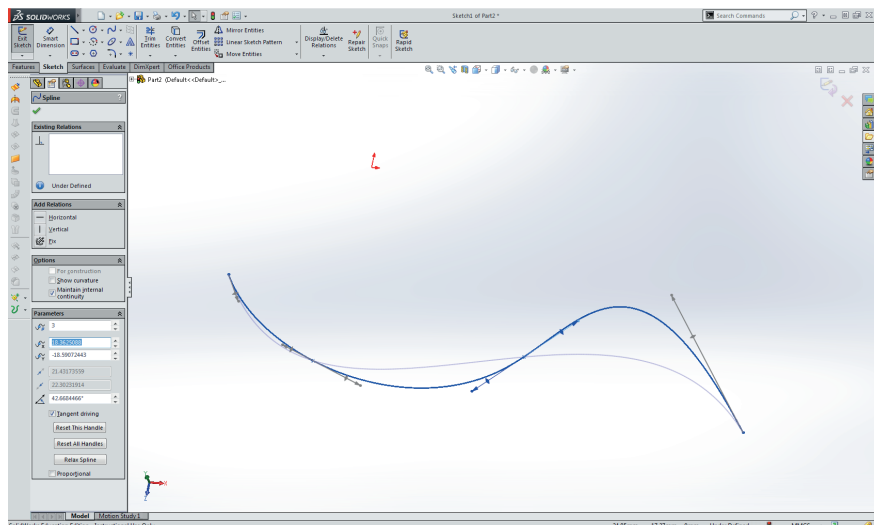
A popular 3D-modelling program in particular for animation and gaming. Some of the aspects that stood out from this was the way it worked with graphs and its manipulator tool.



Screenshots of Maya (Autodesk Maya)

Solidworks

An engineering 3D software. Its use of parametric relations and constrains of the geometry were of particular interests for this thesis.



Screenshot of Solidwork (Dassult Systems)

User Groups

This section will look at some of the main users of Novapoint, and some scenarios where they would need a line tool. These scenarios will not be mentioned much more during this report, but have been in the back of my head as I have worked with the master's thesis developing concepts for a new line tool. A new line tool has to be able to meet these needs that the users have provided, so these scenarios can be used by Vianova Systems to validate their line tool as it is developed.



Road

The road engineers are one of the biggest user groups of Novapoint. Infrastructure projects often revolve around roads, so getting a good tool for placing the road is of outmost essential.

Scenario: A new road

What: Planning of a new road, and placing the center line.

Setting: A new road is to be placed in an urban or a rural area.

Challenges: When working with a centerline road engineers have to stay true to handbooks describing the regulations for how the road has to be made. They need to take into consideration how the curvature of the horizontal alignment compares to the vertical one. While a road engineer is freer to make a line when working in rural areas, they will have to consider the surroundings when the road is to be placed in an urban area. The scale they are working in will also differ, it should be easy to work with the line, both in a scale of km and mm. They need to have good control and flexibility over the horizontal and vertical line.

Scenario: Changing an existing road

What: An old road is going to be rebuilt and made wider.

Setting: There is an existing road with ditches, which will have to be adjusted

Challenges: How to draw the lines and make sure that the old structures can be kept. They would need to construct helping lines to know where it starts and how it can be adjusted.



Landscape

The landscape architects are responsible for the landscaping in the project. This can include sculpting the landscape, or deciding what kind of trees and plants are to be used.

Scenario: Noise Barrier

What: Making a structure to reduce traffic noise from a road.

Challenges: Have to make a line that follows the road and have to be able to have control over how the barrier will meet the existing terrain. Have to be able to draw up both the bottom line and top line of the barrier.

Scenario: Redirecting a Stream

What: Changing the riverbed of a stream.

Setting: There is an existing riverbed, but the stream has to be moved to make place for the new project.

Challenges: Need to be able to place the start of the redirection and where it will have to join the river, and get to know the riverbed's height at each point. Have control over the vertical line and its slopes.

Scenario: Making a Ramp

What: Making of a simple ramp.

Setting: For a project, there is a need to make ramps for wheelchair accessibility in a bumpy terrain.

Challenges: Want an easy way to draw up and construct ramps. It has to fulfill the universal design regulations set by the state (min. 0.9 m wide, max. slope 1:20, and for every 0.6 m elevation a landing of min 1.5 m).



Area Planning

The area planners are regulating the use of a geographical area. They work with developing regions and try to figure out what various areas can be used for.

Scenario: Regulate a region

What: Establishing various areas for a region.

Challenges: Need to be able to mark up the various areas and connect them to the area type. There are also strict technical regulations about how the lines should be defined geometrically, this is something the software should be able to help the user achieve.

Scenario: Adding Additional Regulation

What: Adding another topology to a regulated area.

Setting: There is a region with already regulated areas.

Challenges: The new zones need to follow the old ones where it is applicable and will in some cases be dependent on them, and should change if they change.



Water and Sewer

The engineers planning and controlling how the water and sewer system is going to work are important users of Novapoint.

Scenario: Planning a new Sewer

What: A new road is planned and there will have to be made a new sewer in connection with it.

Challenges: Have to follow the road. Does not turn as the road, but consist of straight elements and manholes connecting them. For the vertical it is important to have control the slope and it should be easy to adjust the line and have the elements connected to it follow it.

4. ANALYSIS

A presentation and analysis of the findings of the research phase

The Analysis

This chapter will discuss some important aspects that have to be taken into consideration when designing a new line tool for Novapoint Base. The reflections considering these aspects are synthesized from the interview, literature research and by studying existing solutions.

The analysis of each aspect will be presented in a manner that will look like this:

Dilemma: A short presentation of the aspect and why it needs to be addressed.

Discussion: Various existing or possible manifestations of each aspect will be presented and discussed.

User insight: This part will be included when needed, and present insights from the users regarding the dilemma, if they haven't already been addressed in the discussion.

Design suggestions: A short summary and some suggestions for what this will indicate for the new line tool.

The various aspects that will be discussed in this master's thesis were sorted into 6 different groups and will in this chapter be presented as shown here. Even if they are sorted into smaller areas, even those areas might cover several different problems, so not everything has been given the consideration it deserves, but are still included to acknowledge that this is something that would have to be solved eventually.

Navigating the Line Tool

- The task structure
- The various views
- Working in 3D space

Making Lines

- Creating line elements
- Using patterns
- Fitting through points
- A to B optimizing
- Supporting lines and points
- Lines from the model
- Getting data from the model

Editing Line

- Editing tools
- Line in a table-form

Communication

- Right-click-menu/Widget
- The line elements
- The labels
- Mouse cursor

Making it smart

- Constrain Control
- Topology
- Different modes
- Automatization

Other aspects

- Types of lines
- Multiple elements
- Undo

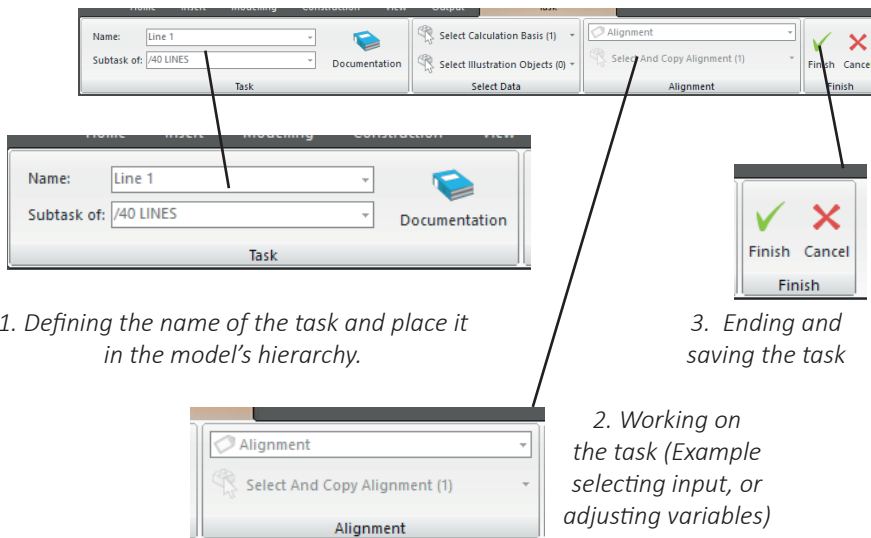
Navigating the Line Tool

How can the user navigate the line tool when constructing lines for a 3D model? This section will look at the challenges that will arise when the user tries to orientate him-/herself working with a line tool in Novapoint Base.

In this section we will look at the task structure, the various view of the model and how the user can be working in a 3D-space. Having a tool which the users can navigating themselves around in confidence while working is of outmost importance for a good user experience.

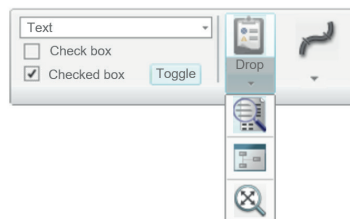
The Task Structure

Dilemma: In Novapoint Base the user works through tasks. A task can be things like importing objects to the model, shaping the terrain, to making a presentation. The line tool will work either as a separate task, or as a sub-operation of other tasks. What does the format of task mean for how the line tool is made, and how can such a task be structured?



Discussion: When a user works with a task, a new ribbon opens up. This ribbon act as the main interaction for the user. A task-ribbon is divided into various segments. All tasks start with a “Task”-segment were the task is given a name and placed into the model’s hierarchy, They end with a “Finish”- section were the user can confirm and save the task, or cancel it. All of the sections in between these two vary depending on the task, and it is through working with these sections the task is performed.

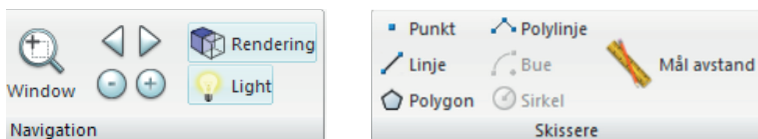
The input of the user is done mainly through textboxes, toggle buttons, checkboxes, lists, buttons that activate various functions, and other form-elements that most people are used to from normal web-browsers and writing programs.



An example of some of the form elements that the user might encounter working with a task. A list, checkboxes, a toggle button and a button with a dropdown menu giving access to more buttons.

Knowing what kind of elements that are available for use and how they work is important when constructing a new task. Having a toolbox can be of great use. Currently in Novapoint Base there is very little use of smaller buttons, they are only used rarely, however looking at other programs we can see that they are quite efficient when it comes to making room for more functions, and they also help differentiate the buttons, bigger being more important for the user, than the smaller ones. Looking at the Road Net Ribbon in the “Road Net”-mode we can see that there is a row of equally sized buttons, this means that it is hard for the user to sort their meaning and importance

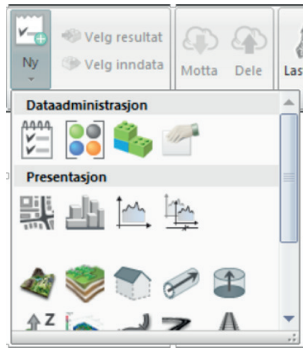
without looking at everyone and then evaluate them in his/her head. Since all of the buttons are big, the whole ribbon is filled up, which means that it has to have a big screen with high resolution to be able to show them all. If they had known how to use smaller buttons when making the task, they could have solved several of these problems.



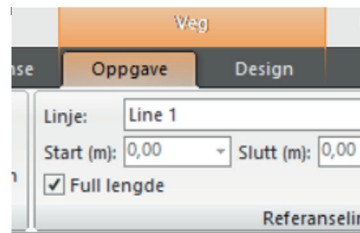
Example of how smaller buttons icons have been used in Novapoint base, and suggestion for how a simple drawing tool can look like.

It can get quite crowded when there are more complex tasks at hand. So far there has been implemented several strategies to solve this problem in existing tasks in Novapoint Base. These strategies include dropdown menus with a button gallery, spreading the task over several ribbons either by temporarily replacing the current ribbon when working with a subtask or by having several folders of ribbons.

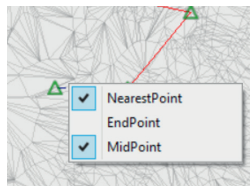
Not all of the interaction with the task happens in the ribbon, sometimes objects will be selected from the explorer or from one of the views of the model, or through a pop-up window where the settings can be adjusted. This is partly due to there being limited space in the ribbon, and because there's sometimes a need to interact and work directly with the model. In the two current tasks of terrain forming and road network were the user is actively working with the model they have in addition to the ribbon tried two different approaches to how to work in the view, either by hot keys, or by right-clicking/pop-up menus. These last two approaches will be considered in more detail later in this chapter.



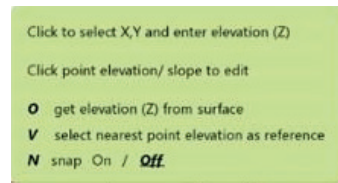
A dropdown button gallery.



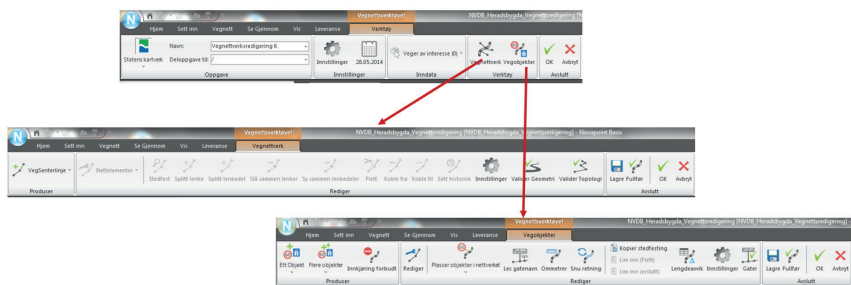
Have more than one ribbon folder for a single task.



Have the option to right-click to open a menu with various alternatives.



Have a "Widget": a dynamic list over hot keys that can be used to initiate various functions.



Have a nested ribbon system, so that by clicking on certain ribbon buttons opens new ribbons with more options.

The fact that lines can be useful as subtasks for other tasks, and there is an uncertain amount of function to be implemented, not all necessary from the very beginning, means that the way the Ribbon is structured should be flexible and not set in stone. It should be able to scale up, or scale down depending on what it will be used for. It is however natural to think that there will be a line tool that works exclusively as a line tool, and this should have a good consistent way of working.

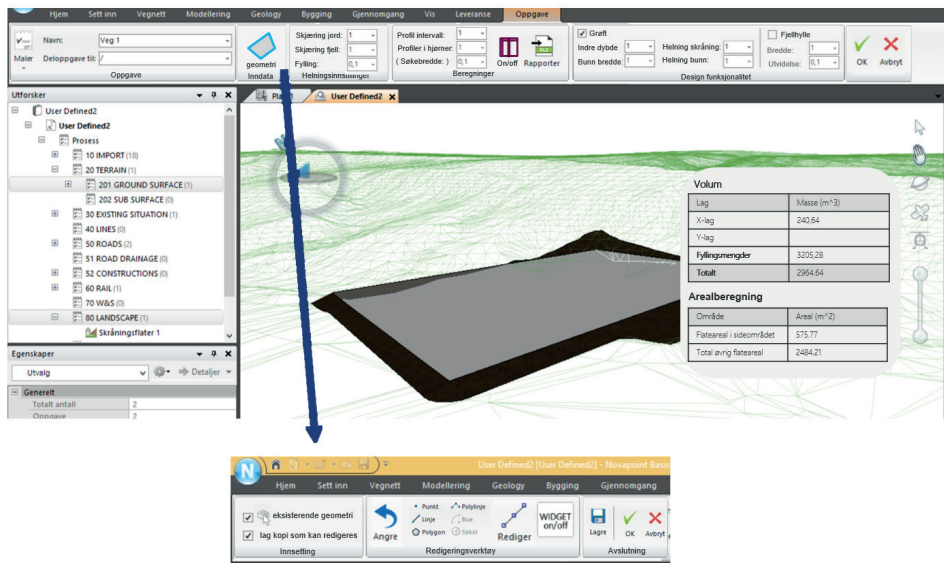
Another thing that the task structure should consider is the difference between expert and novice users, as these have different needs. A novice has to be able to understand and to figure out how it works, while an expert should be able to do his work as efficient as possible, maybe by using things like shortcuts. (Russell-Rose and Tate 2013)

User insight: Some of the feedback I've gotten from the people working in Vianova Systems is that there is a lack of a common design guidelines for how to work and implement new tasks in Novapoint Base. This means that each group that are developing new tasks for Novapoint Base have to figure how the user should interact with the new task. This means that the user interaction of Road Net and Terrain shaping works very different. This will be addressed further later in the thesis.

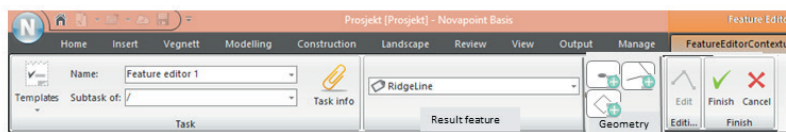
Design suggestions:

- Some part of the structure of the ribbon is given, however there has to be made some plans for how the line tool can evolve, and be able to be a subtask of other tasks.

Some suggestions for how the ribbon can be utilized for the line tool:



1. Let it be a quick subtool for other, where they can start the linetask in whatever tool they are using. This could either be a small simple set of functions in the case of making a linestring.



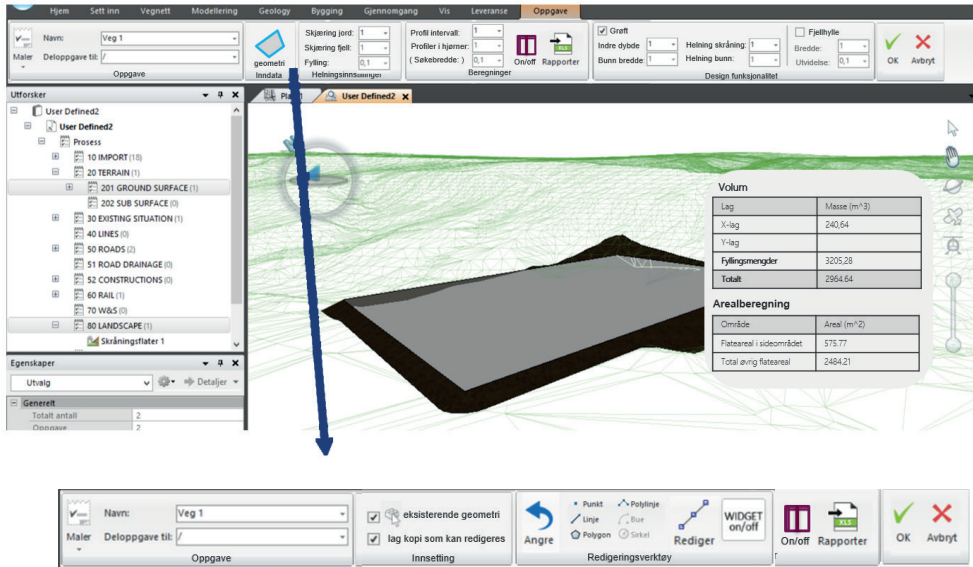
Click to select X,Y and enter elevation (Z)

Click point elevation/ slope to edit

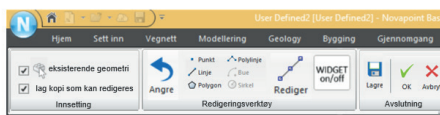
- get elevation (Z) from surface
- select nearest point elevation as reference

N snap On / **Off**

2. Could also just be a drawing tool, working like terrain shaping, whit all the interaction through the hot keys.



3. Collapsing/uncollapsing ribbons.



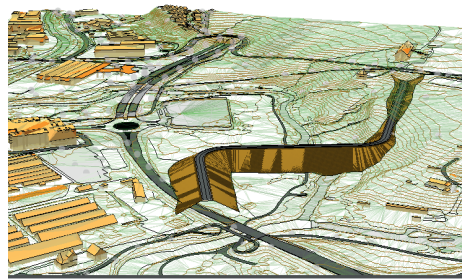
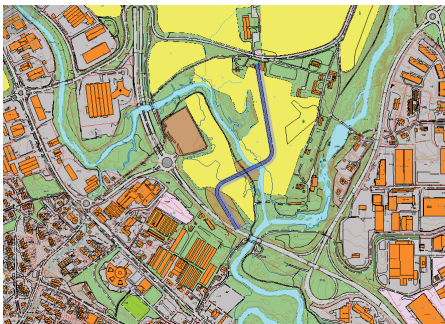
4. Mix of the various methods: use of hotkeys, but with the buttons visible. The widget and mouse text as a way to communicate current input and where in the process they are. Right click and questions to clarify things.

The Various Views

Dilemma: How is the model presented, and what does the user work with as he/she looks at the line.

Discussion: Today there are two main ways to look at the model in Novapoint base, either with a plan or 3D view. In addition it is possible to look at the cross section/long section of the model given a line as a starting point. The user chooses themselves what part of the model that's visible in the various views.

Given that the line tool will have to cater to the vertical/horizontal alignment and this and the line string is the two line types that Novapoint Base currently have built in, it will be natural to mainly focus on these types of lines and not others (more on this later). For this sake, it will be essential to be able to have a vertical view window for the line tool so the user can control and draw the vertical line. The vertical window can however not show anything without there being a line already, as it needs some reference to were in the room it exists.



Show a picture of the plan view and the 3D view, maybe even include the detail of how they can be made right-clicking selected models.



Also show the picture of the cross-section and the way we look at a line in AutoCAD today.

There are several reasons why a side view is less important in Novapoint Software than in the other 3D-software. The first is the type of projects it works with, since these are of such scales that showing everything from the side would just make everything look cluttered. One alternative for having a cluttered view is to let everything but the selected line be opaque, and fade away to nothing, as objects get less relevant with more distance from the line.

The different views needs to work in a way so that they correspond to each other, so that it obvious how the line and objects relates. There has to be good labels and clear graphical indication or a good dynamic system that makes sure that every part that is selected is clearly marked in the other windows as well (included the property window). These graphical aspects will be discussed further later.

When it comes to the vertical height there might be a need to have a section of the ribbon dedicated to it. This would include settings like how close and what kind of objects are included in this view. There is also the issue of how to handle more than one lines if they are selected and how a closed line should be represented.

To help the user get a connection between the various views and able to orientate between them, they could automatically respond to what the user does in the other views, for example if the user zooms in on a particular place on the line in the plane view, the computer can make sure the same part is visible and maybe get zoomed in on in the other views as well.

Other softwares have other variants one in particular to point out is the possibility to have side views. One reason to use them is for the user to know in which plan they are operating and have control over how the various parts of the mode is related to each other. Today Novapoint Base, does not have this option. The Novapoint orientation tools doesn't have much of a "homing" decive to get to certain views (like side-views).

User insight: For the road designer working with the center line of the road it would be essential to have a way to see the vertical version of the line like they are used to from the alignment tool in AutoCAD.

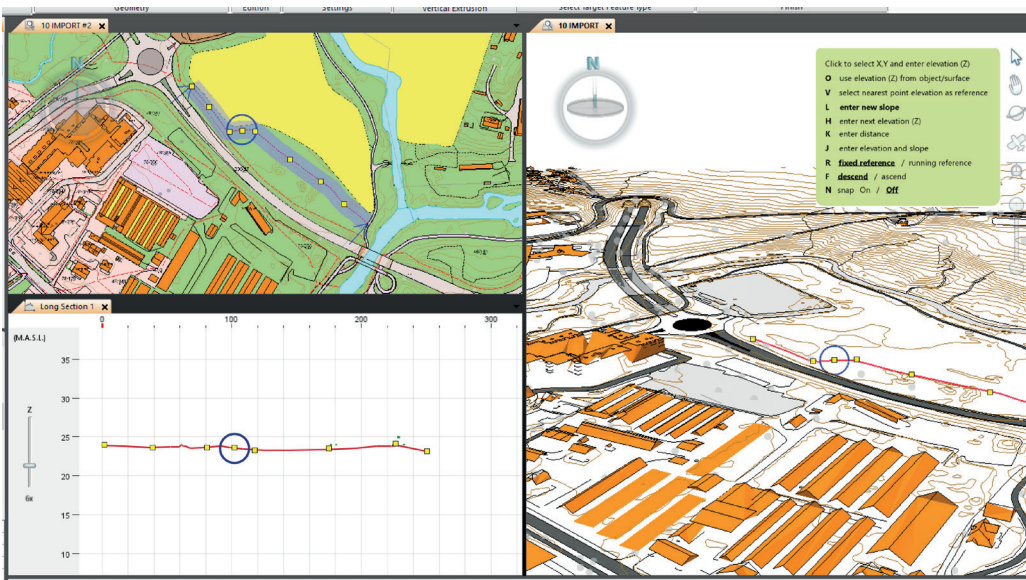
The area planners were mostly working in the plan view so for their sake the other views weren't that important. However, I was told that it would be useful to automatically have the line follow the terrain, and it could be useful to look on the 3D-model as a reference.

A concern several presented was the fear of the screen being too cluttered if there were too many windows that should be open at the same time.

Talking about the side view with one of the programmers I was warned about it the view being a non-Euclidian space, which means that the 1 to 1 presentation isn't necessarily true. For example will this view if showing objects near the line possibly have more instants than one of the same if for example the line does a sharp U-turn, or crosses its own path.

Design suggestions:

- Make a dynamic vertical view of the line



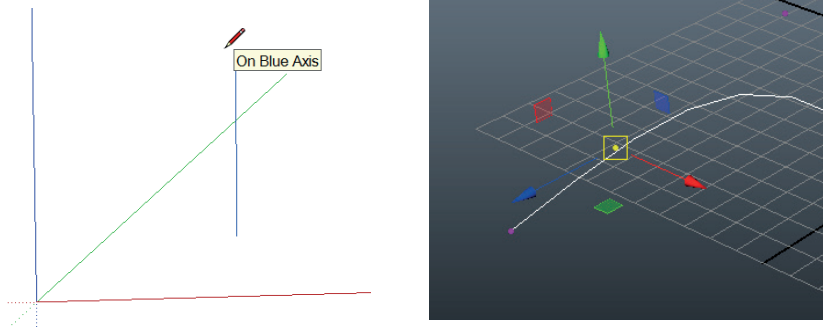
The whole idea of a dynamic vertical view window and the dynamic interactive 3D window is the new innovative part of this part. Make sure they are connected!

Working in 3D space

Dilemma: How can the user have control over the line when working in a 3D space? In particular when the screen they are working on is 2D.

Discussion: One normal solution to let the points that the user is placing or moving snap itself to existing geometry. This is already happening with the measure tool in Novapoint Base. Snapping will be discussed in more later in this chapter.

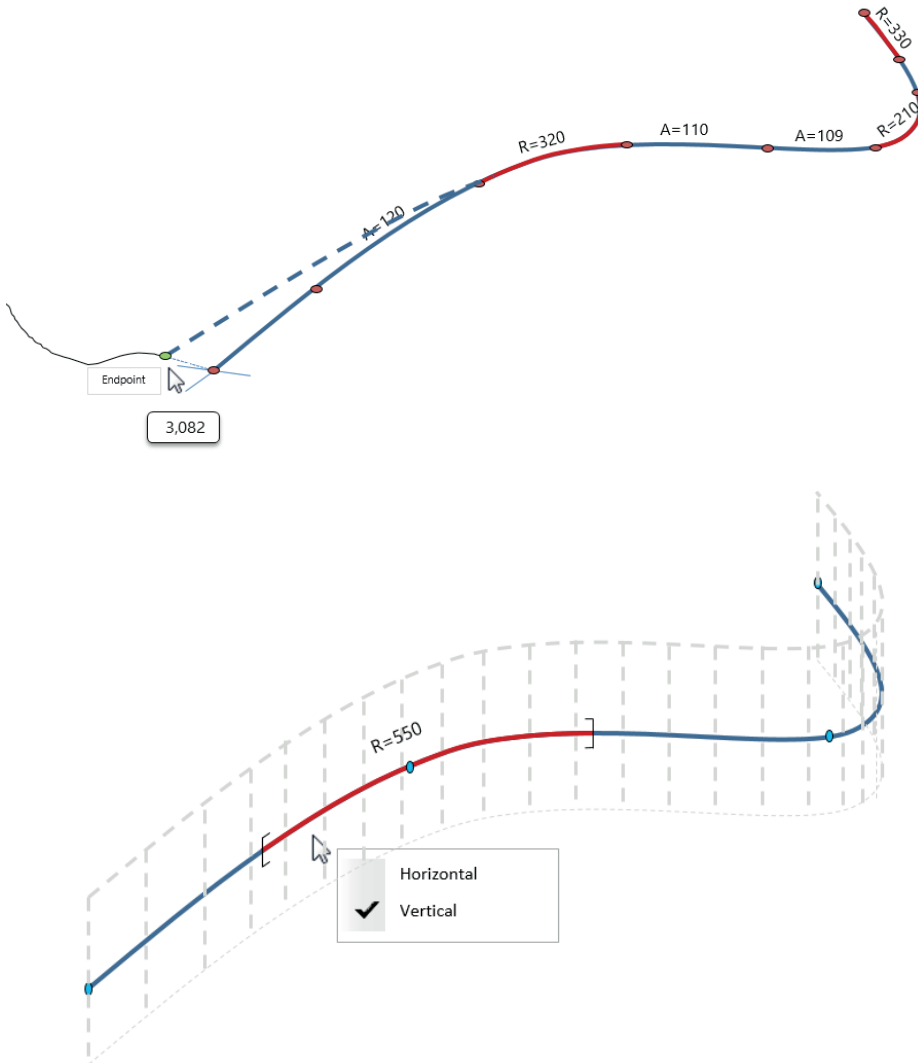
Both SketchUp and Maya uses the three x,y,z axis to help orientate the movement of the tool. SketchUp snaps itself to the three axis as it works, while Maya gets a manipulator tool that lets it drag it in the wanted direction.



*Illustration of SketchUp function and screen shot from Maya.
(SketchUp, Autodesk MAYA)*

You can place straight line without problems in a 3D space, but what about a curved line? There could be drawn an arc in one of them and then the other would just be a line placed. Maybe in either a certain radius or something else. The 3D line might be restricted when it comes to what you can and can't do in the 3D space.

"Easier to edit than to place!" This indicate that we could simply work with placing straight lines into the 3D view and then see what to do with the line afterward.



An illustration of some suggestions for how the line could be edited. This is from an internal document in Vianova System where some ideas about the alignment line and how it could be worked with in 3D.

Snapping and following the terrain is also something that is about working in 3D, however the thesis will come back to these later in this chapter.

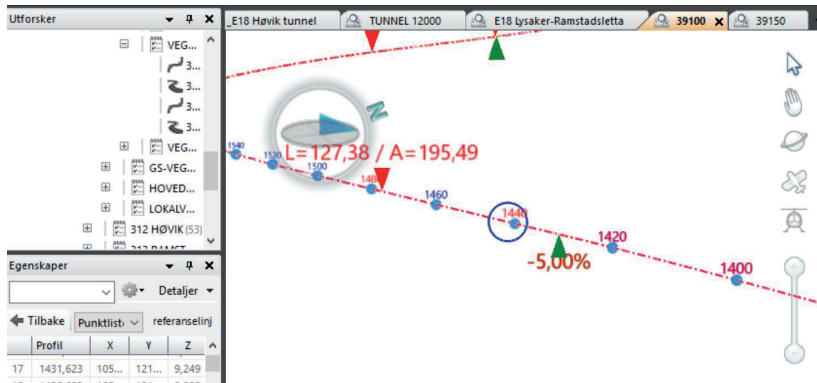
In the pictures notice how a cross comes up when you work with something in plane view so you have control over how you adjust it there. This could mean trouble for the vertical view because it will have to know how that should change, should it follow the line or the "room" if a line get longer? Because this could vary?

Something else that I haven't discussed yet is the scale issue with these projects. The roads they are constructing can be tens of kilometers long, but at the same time they might have to adjust it at times with 0.1 millimeter accuracy. This is challenging! This means that how the line is presented and worked with needs to be able to scale well between these kind of scale differences.

User insight: Having control when working with the line in the 3D view was a concern for several user groups. Today most of their work is primarily done in plane/side views so they weren't necessary working directly in a 3D view.

Concern for how they would know which direction they were working with and have control over how their editing would affect the model.

I also discussed about how bad today's representation of the alignment line worked in Novapoint base, and that it would be important to easily understand the line and what you're seeing and to be able to decode the symbols.



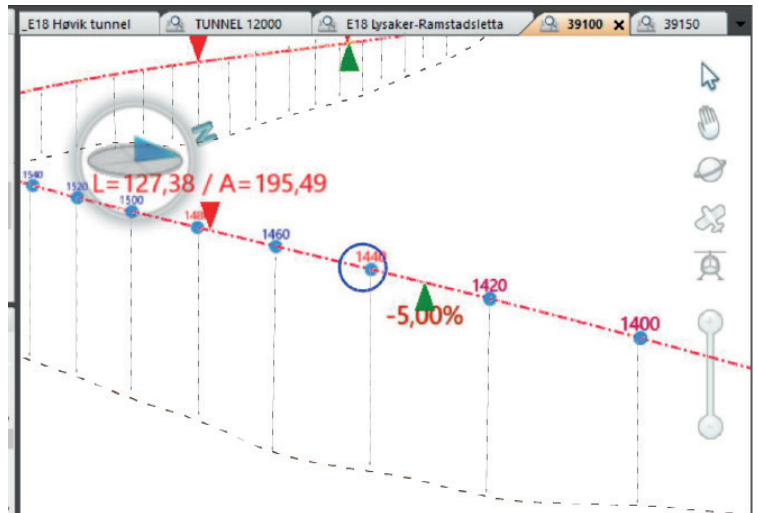
This picture show a line alignment in today's Novapoint Base.

Another thing that this picture reveal is that it is close to impossible to read where in the room this line is, since there are no other geometry to use as a reference. A suggestion from one of the people I talk with was to have reference lines going from the line to a given surface or the zero-height-plane, so that it would be possible to read the line better in 3D.

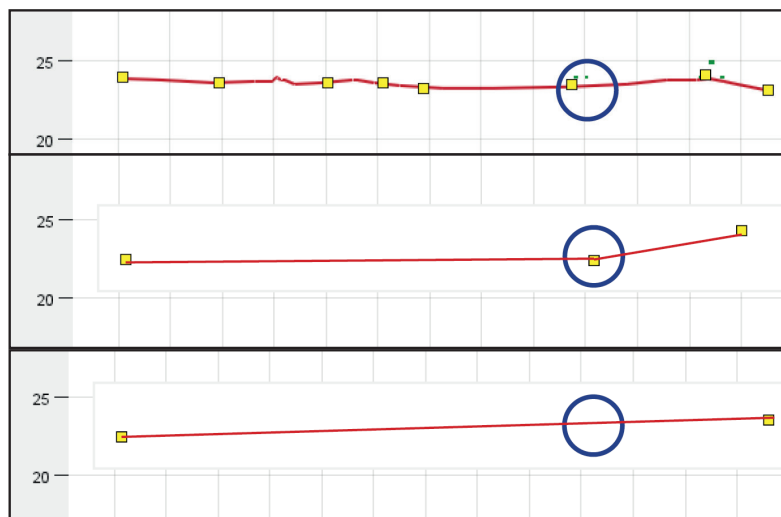
Since the road engineers wants a beautiful road it is important to enable them to see how the vertical curvature is related to the horizontal one in a good way so they can make sure this works well.

Design Suggestions:

- Go for two mode to make and edit the line, one in horizontal plan and one vertical along the horizontal line.
- Need to find a good strategy for how to visually help the user navigate in 3D like SketchUp or Maya.
- On the next page some strategies are presented.



Lines showing height along the line



Some strategies for vertical when drawing horizontal

1. Following the terrain
2. Making height for every point
3. Making a line from start to end, to be edited later

Making Lines

This section is about how to create new line objects in the model. Taking a closer look at various methods to create new line types and various ways this can be done. So this will mainly look at very basic principles and discuss them.

Creating Line Elements

Dilemma: How can the user make the basic line elements that the horizontal and vertical alignment consists of: straight lines, arcs and transition curves?

Discussion: This is the very basic drawing tool in SketchUp, a basic line tool, a free drawing pencil that will generate a composition of lines and arcs, and a couple of variations of how the arc can be made. AutoCAD and partially SketchUp both let the user write in the length of the line, the radius, or other parameters as they draw the line.



Screenshot of SketchUp tools (SketchUp)

Today the transition curve is only made using clothoid, but in the future, this should be expanded to also create sinus, biquadratic transition curves.

A simple line is better than no line. People would have liked to be able to make lines directly in Novapoint, even if it was just simple lines that would have to be sent to AutoCAD to be refined. But it would make it possible to work in the big model where everything is and it would be easier to place it and see how it would work than in AutoCAD. Of course, this would mean that there should be a good way to export it to AutoCAD to work with it there after placing it in the environment.

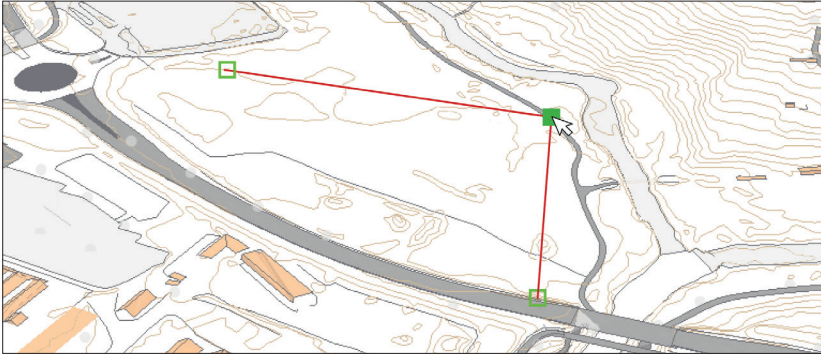
As we have seen there are many ways of making lines it is important to make sure that we have enough functions to make the user feel like they're able to make the line they need, but at the same time not have unused functions that clutters the task.

Design suggestions:

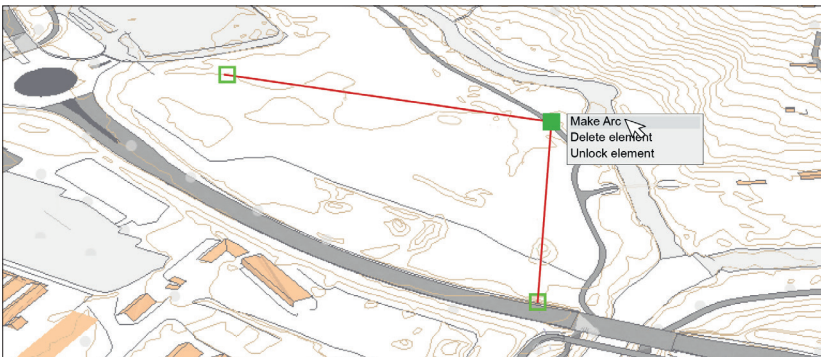
- We have to establish some basic draw functions for the basic line elements. This could include the option to type in values.

- Should allow arcs and transition curves to be made based on existing geometry.

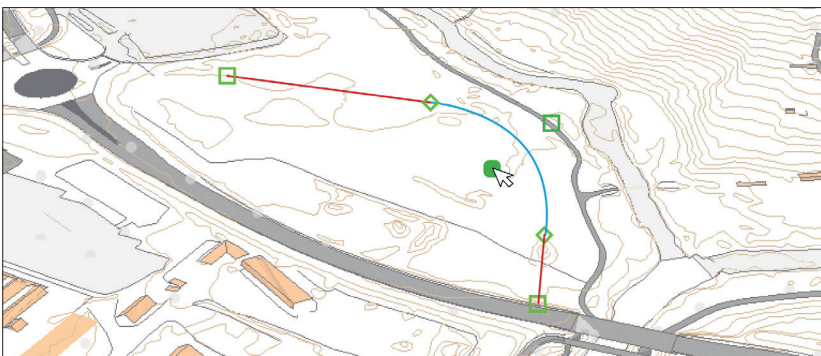
1.



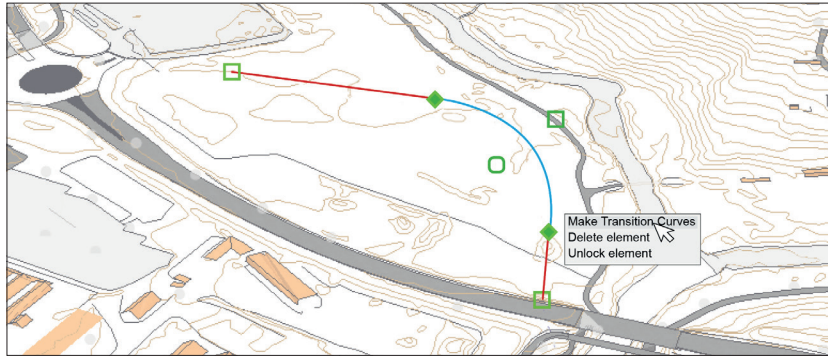
2.



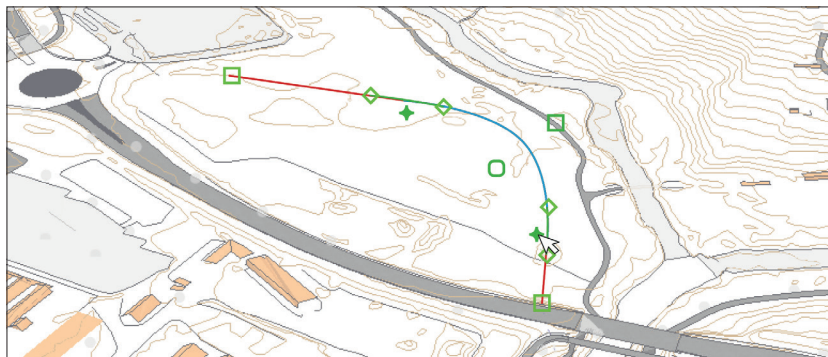
3.



4.



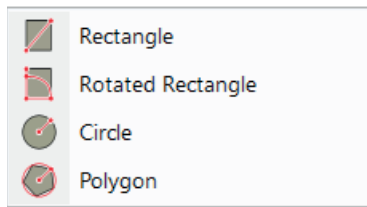
5.



When something is selected (1) you can get a menu of options when right clicked (2). This can make corners/straight lines into arcs (3). Similar can you right click when several points are chosen to get alternatives what to do with them all (4). They can for example be made into transition curves (5).

Using Patterns

Dilemma: Sometimes software offers shortcuts, or methods to do things quicker, one way could be to give the user a pattern that they predict the user will use and lets them use it. What kind of patterns are there and is this relevant for our line tool?



Screenshot of SketchUp tools (SketchUp)

Discussion: Some normal patterns are for the user to put out a square, circle or a star since these are common. Will pattern like these be of relevance to Novapoint users, or should they just be forced to draw out these by hands if they need them? If there is good automatic feedback like in SketchUp that shows how the lines are related to each other and length this might not be so much harder than just selecting the square or circle in the first place.

It is again important to not overdo something and give the user too many options if they are given patterns. There are few if any people working on an infrastructure project that needs to draw stars at a regular basis, so this should not be necessary to do.

Today in Novapoint's alignment tool in AutoCAD, it is possible to put out a linestring that automatically gets an arch in each corner, or the user could place certain constellations of the line elements like a "line,transition curve, transition curve, line"-element.

User insight: I was told that it is easy for an unfamiliar users to misuse the line-arc-line tool in Novapoint's AutoCAD alignment, and create unsolvable geometries.

I observed that the line-arc-line tool was frequently used and it is much faster to do than drawing the arc manually between each line. Editing it afterward would probably be easy.

Design suggestions:

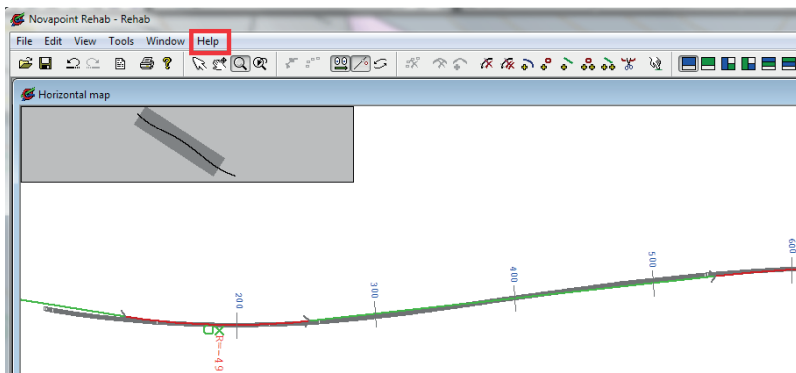
This should not be a priority, however patterns like circles and squares might be useful eventually.

The possibility to place lines with automatic arcs in between can be of use and should also be considered in the future.

Fitting through points

The dilemma: How to make a line fit through given points and some of the uses for this.

The examples: This is the tool today in Novapoint AutoCAD which can construct a line given a set of data points.



Bentley 3D Rail Track is another program that can do this, however a user that would choose to normally work with an alignment design in Novapoint, would prefer this program if the line was based on a dataset, as it was more powerful and he also had the option to weight the importance of the various points.

Freehand drawing of a new line or nudge an existing line use the same mathematics as the fitting through points. The computer would use the input from the user (a set of points where the mouse passed) using this as a base it would calculate how this can be interpreted as lines and arcs.

An example of free hand drawing exist in SketchUp or Illustrator. Earlier they have worked with a free drawing and nudging tool for Novapoint, though these were never implemented - this means that a lot of the mathematic behind it exists already.

The first part about taking in a dataset to make a line, would probably be so complicated and with so many settings that it

would better work as a separate task, or at least a subtask of the new line tool. So it should not be prioritized.

Nudging and free drawing are other things that will be useful in the main line tool both as means to create and edit the line. These will make it able for the users to work more organic and sketch thing up quickly.

User insight: One example where this kind of concept would be valuable is where you have data measured on site and want to implement those data points as a base for a new path of way.

Design suggestions:

- Implement a function to make a line fitting through points.
- Use the same principles to create a free drawing and a nudging tool.

A to B Optimizing

Dilemma: If the user knows the start and the beginning of a line that will be used as a centerline for a road for example, it is a possibility that given all the right parameters, the computer could analyze the various possible paths and figure out what would be optimal.

Discussion: The software “Quantum” is an existing software that does this optimization for the user, and is good at it. However this is something that will be out of scope for this master's thesis, but in a far future this is also something that would be beneficial for Vianova Systems to have implemented in their system.

This is something that is very useful and something we should have eventually, but again not something that is worth prioritizing as this would demand a lot of work. This would likely be in a separate task though as it will probably have a lot of input and settings that is irrelevant for the rest of the line tool.

Design suggestions:

- This function should not be a priority.

Supporting Points and Lines

Dilemma: How can the user work with lines and geometry that are not a part of the final project, but aid the working process of constructing the model.

Discussion: Solidwork also initiates "tasks" were the user sketches, some of the line it draws are reference lines that are only visible when the sketch is open, but not otherwise.

Another example can be found in SketchUp where the measure tool can leave the measure lines in the model and then be used as reference lines later when the user is working.

There should be possible to make sort of lines or points to work as an aid in constructing the real lines. These should be possible to draw in any view and only be visible as the task is active. At a later point, those supporting line should be able to be converted to normal lines with a feature type.

User insight: I know that there is use for this in road projects as I was shown road models with lines working as supporting lines for creating the road.

I know that Vianova Systems are considering making some kind of unspecified line type (feature type) for landscape architects. Sometimes the same line can be used as many things like both the edge of the road and the edge of the payment. This could be solved by having a "feature less" line type that can be copied and given a feature type later.

Design suggestions:

- There should be possible to make supporting lines to aid in the construction of the project lines.

Lines from the Model

Dilemma: Sometimes the existing model already has a line or geometric elements that the user want to have as base for their new line. How can the user get use the existing geometry in the model to create new lines?

Discussion: We have existing lines that the user might want to use. Either these could be at the same place (except maybe a change in feature type), the user might want to copy it and move the whole line somewhere else, or he/she might want to make an offset of it.

Notice that there is a big different between offset and copying. In particular an offset is a much more tricky operation for the computer to manage, in particular if the offset isn't simply just in one horizontal or vertical direction. The geometry can "grow" out of bound, the arc can get too little when it is scaled down, and what happens then.

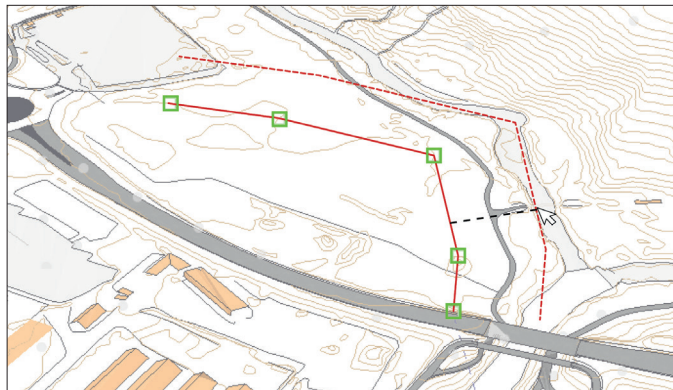
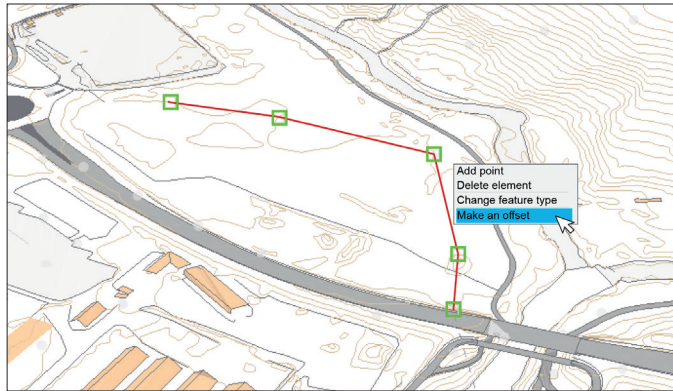
In Road net there is a particular feature in the property window for lines that let's the user decide if they want to keep the old or make a new line when they're editing a line in the network.

Another scenario of existing lines in the model is where surfcases intersects or parts of a solid 3D-figure.

User insight: Many have expressed their need for this, for example road engineers working in the city needs to extract the lines of the existing features, and both area planners and landscape architects needs to be able to get the line from the road. Often will they might they also need to have a longer section of a line than the road model offers, or shorter one. How this can be done will be talked more about in the next section.

Design suggestions:

- Make an effortless way to select an object in the model and extract a line from it, either manipulate it directly, through a copy, or an offset.



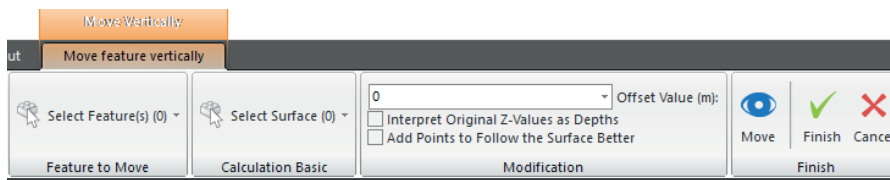
Demonstrating offset, notice that is is something else than copying.

Getting Data from the Model

Dilemma: How to use parts of the existing geometry when constructing a new line?

Discussion: Snapping is one of the most important tool when placing something in a 3D model. This is already an existing feature in some Novapoint Base element like the measure tool, and is something they are currently working to improve. This way of getting information of from the model is looked closer at in the constrain section later. However notice that there might be some challenges with the snapping tool, when the user is working in the vertical/horizontal alignment mode, in particular if it is done directly in the 3D view.

One of the most important tool when it comes to getting data from the model is to get height from an existing terrain. This can be seen as the same operation as happens in the tark of moving an feature vertically



I also have to mention the get height task from Novapoint base. There is also the "o" function in Landscape that is very important at do a lot for the terrain shaping tool. Both of these are ways to get the current terrain height.

A question is how the line should react if the underlying terrain si updated. For example if a line has its height set by elevation of the terrain when it was made, should it follow the terrain if the user decides to move the line horizontally?

User insight: The areal planners told me it would be useful to have the line automatically get the height of the terrain.

Design suggestions:

- There should be possible to use a given terrain as a base for the height of the line.
- How the snapping tool can work with the alignment should be furthered tested and evaluated.

Editing Lines

This section looks at how to change the line and adjust it.

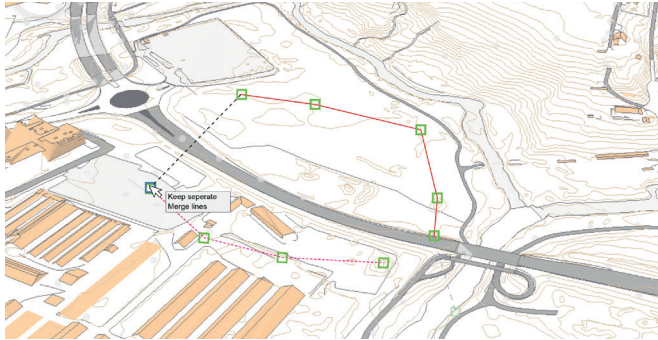
Editing tools

Dilemma: How to change the line. Looking at some normal editing tool

Discussion: In 3D-modelling programs, there are usually a stable trio of editing tool called move, rotate and scale. These geometrical operations can all be done to a 3D geometry along any axis. but when we work with a horizontal and vertical alignment these can only be done in one of the two planes that defines the lines. If you try to scale or rotate the line in another direction this would lead to a change in how the horizontal and vertical presentation would have been. It might be possible to do this, but it could lead to complications. This means that this normal trio works slightly different on the alignment tool than usual, but it does not mean that they don't have their uses, as the user till can use the operations along either of the two plans. They might have their uses, but that does not mean that the line tool is useless without them, so this might be features that can be delayed a bit.

Other important editing tools that people has been requested is line cutting and line merging. These tools should work smoothly and could be done pretty dynamical by just clicking on the line to cut it, or moving an endpoint to another to merge them.

In addition to these tools we might want a separate eraser tool in addition to the possibility to select something and then click on delete.



Example of how merging can work, when placing a point on top of another end point you get asked if you want to merge, or not. Could also merge by selecting to end-point and choose to merge them and then a straight line will bridge them together.

A possibility is to make a “fake” alignment tools, this means that there are only a polyline that has been given layers of arc and transition curves on them. This means that it can be easier to adjust the line in 3D. The arch/transition curves between them are simply quick things with a set size each of them. This might later be changed to a real alignment with no way back. The positive side about this face is that it is easy to adjust, however the arcs/transition curves will have to be at the same chainage point both in the horizontal and vertical position.

Design Suggestions:

- A glue/merge tool is important to implement.
- There should be tools to delete parts of the lines
- It could be useful with both rotate and scale in the horizontal and vertical direction; however, this is not something that needs to be prioritized.
- Might want to test out the “fake” alignment tool.

Line in a Table

Dilemma: Novapoint's alignment tool in AutoCAD works with a presentation of the line in the form of a table, where the line has an alternative to the graphical presentation, and the user is also able to edited the line by directly change the values in the table. Should this be done in Novapoint base, and then how?

Discussion: In Novapoint Base today has a polyline the opportunity to adjust the nodes directly by putting in new numbers in the list, and interpolate the z-value. As it is today we could make it easier to change these values by for example having the opportunity to select more nodes that can be changed at the same time, maybe using the scroll-method or drag on some kind of slide to adjust the value up and down in addition to just typing the new value.

If the line a vertical/horizontal alignment, the details of the property gives the option to switch between 3 tables: Horizontal, Vertical and point list. These can not be changed or edited on directly, like the similar table in used in Novapoint's alignment tool in AutoCAD. It would have been useful if the tables in Novapoint Base be changed and adjusted. In addition to the change in number it might be necessary or advisable to include a few of the functions that are worth to keep alive and which are found in the old alignment tool. These extra options for the alignment table in Novapoint Base could be made available by right clicking.

Two important aspects with the current alignment tool is that it tells if the geometry is a bad geometry, and that it has the opportunity to use an undo history so that the user can regret his or her input. Both these aspects will be looked closer at in connection to other aspects of the line tool.

Design suggestions:

- There should be made an adjustable table for the alignment tool.

Properties

Alignment

Point List alignmentGeometry

	Chainage	X	Y	Z	Length, 2D	Length, 3D	Gradient
1	0,000	990...	121...	35,3...			
2	5,000	989...	121...	35,4...	5,000	5,001	1,69 %
3	10,000	989...	121...	35,5...	5,000	5,001	1,69 %
4	15,000	989...	121...	35,6...	5,000	5,001	1,69 %
5	20,000	989...	121...	35,7...	5,000	5,001	1,69 %
6	25,000	989...	121...	35,7...	5,000	5,001	1,69 %
7	30,000	989...	121...	35,8...	5,000	5,001	1,69 %
8	35,000	989...	121...	35,9...	5,000	5,001	1,69 %
9	40,000	989...	121...	36,0...	5,000	5,001	1,69 %

Properties

Alignment

Horizontal elements alignmentGeometry

	Element	From	To	Radius	Length	Parameter	Start X	Start Y	End X	End Y
1	Straight Line	0,000	93,343	0	93,343		99000,910	1212548,850	98952,918	1212628,911
2	Clothoid	93,343	109,343	0,000 > ...	16,000	21.909	98952,918	1212628,911	98945,964	1212643,264
3	Arc	109,343	143,159	30,000	33,816		98945,964	1212643,264	98955,138	1212673,978
4	Clothoid	143,159	159,159	-30,000 ...	16,000	21.909	98955,138	1212673,978	98968,825	1212682,166
5	Straight Line	159,159	273,793	0	114,633		98968,825	1212682,166	99072,033	1212732,054
6	Clothoid	273,793	289,793	0,000 > ...	16,000	21.909	99072,033	1212732,054	99085,720	1212740,242
7	Arc	289,793	312,647	30,000	22,854		99085,720	1212740,242	99095,881	1212760,099
8	Clothoid	312,647	328,647	30,000 ...	16,000	21.909	99095,881	1212760,099	99094,515	1212775,990
9	Straight Line	328,647	561,396	0	232,749		99094,515	1212775,990	99054,080	1213005,200

Alignment Design Professional - 40 LINES\Line1 - Calc. basis ok

Object Edit View Insert Modify Tools Data Window Help

No.	Element	Type	Radius	Length	Param.	Bearing	Arc type	E1	M1	E2	M2
1	Line	—x—x	30,343			2,111		99000,910		1212548,850	
2	Transition	—x—x	0,000	16,000	21,909						
3	Arc	—x—x	-30,000	33,816			Long	98951,741		1212654,240	
4	Transition	—x—x	30,000	16,000	21,909						
5	Line	x—x—	114,633			0,450					
6	Transition	—x—x	0,000	16,000	21,909						
7	Arc	—x—x	-30,000	22,854			Long	99099,890		1212745,520	
8	Transition	—x—x	-30,000	16,000	21,909						
9	Line	x—x—	232,749			1,748				99054,080	1213005,200

Pick Object c Recalculate All OK - Draw Cancel

Calc OK HORIZONTAL TERRAIN Fill=0% Cut=0%

Example of how the details of Horizontal and vertical alignment lines looks like. This is similar to the form in AutoCAD (at the bottom) that are used today, so we could try to make it possible to edit the line directly in the property window.

Communicating the Lines

The computer is communicating with the user through the screen. This chapter about what the user sees and how they interact with the line tool. This is closely connected with the navigation, and therefore some aspects that could have fit here has already been explored.

Right-click-menu/widget

Dilemma: Both widget and right-clicking-menus has been used to be able to work in the 3D views of Novapoint Base, how can they be used in the new line tool.

Discussion: In the road net the right-click-menu strategy is used. When the user is right-clicking a menu over his/her choices are presented as a menu, these are to a great extend the same as the one in the Ribbon. A menu like the right-click-menu will also appear when the user has various choices, for example if there are more than one object under the mouse cursor it will ask which one of them the user want to select.

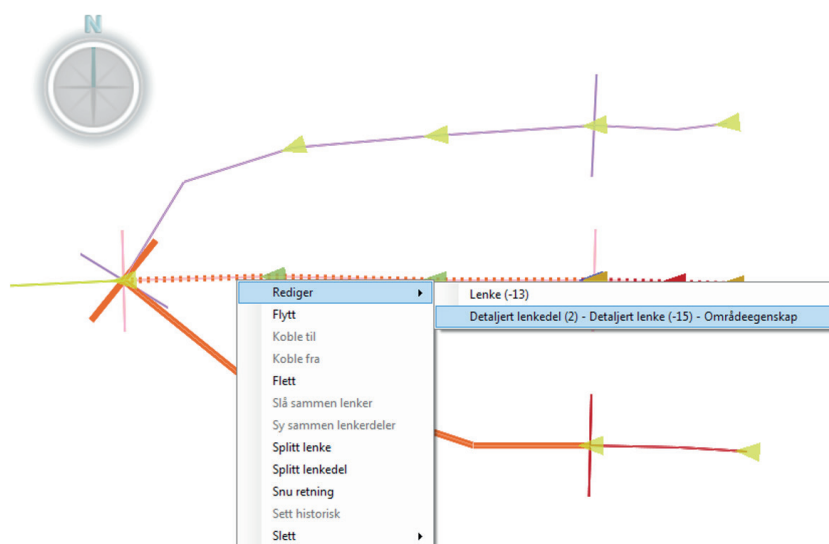
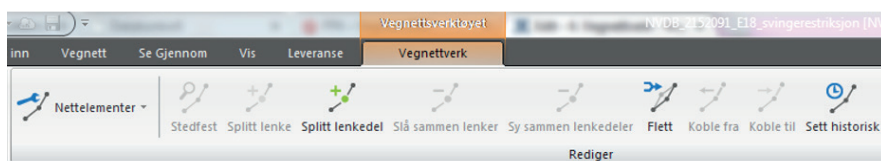
The widget strategy of terrain shaping uses only the widget that provides hot keys that varies depending on where in the process the user are. Terrain shaping has no button alternative like rod net.

For both of the strategies it is a good thing that user don't have to move their mouse away from where they are working and up to the Ribbon, but instead get various options available depending on the selected objects. For example the option to change feature type, copying it, etc.

The widget demands the use of the key board and the eye would have to go up to the widget to look for the options until the user know the keys by heart. The right-click-solution gives the user the option to only use the mouse, the menu is where the

mouse already are, however it demands several mouse-clicks to activate a single function, compared to the widget where only the hot key is needed.

If the two solutions are going to be merged there has to be an assessment of which functions should be a widget function, which a right-click-one and which only accessible from the Ribbon. The widget might fit to functions that are used often and inside of other operation, or for changing between common working modes, for example between working in the horizontal plan and the vertical plan. The right-click can be used by slightly less used methods, or be checking of settings like the snapping setting. However when there are several menus available it is important that everything can be accessible from a place where



Click to place point
 Click (not selectable) + drag to make a selection rectangle
 Click (on selection) + drag to move selection

Roll mouse wheel to adjust value

Shift click: add/deselect from selection

O elevation from object/surface
L set slope
H set elevation (Z)
K set distance
F descend / ascent
N snap On / Off
I insert point
C close line

Q draw line
A draw arc
S draw transition curve
W edit mode
T trim
G cut
M merge
Y copy
U offset
V vertical / horizontal / other mode
? set arc

D free drawing
X nudging
Z eraser
E scale
R rotate
B edit values
P select from task / visible
? constrain mode thing
? (measure mode)
? (preset values)
? (linestring <-> alignment)

Ctrl+Z undo
Ctrl+Y redo
Ctrl+M select All
Del delete selection
Ctrl+'+' increase label size
Ctrl+'-' decrease label size
Ctrl+C copy
Ctrl+X cut
Ctrl+V paste
Ctrl+Tab change view window

This widget box shows most functions I find relevant for a fully implemented smart line tool in Novapoint Base.

The first batch of quick keys are from various tasks in Novapoint Base, so the already established quick keys.

The rest is given random letters, from this it is apparent that widget cannot be done with this many options. So the widget must be used in another way than having everything available at the same time.

the user will find it, and not only hidden in a part of the right-click menu that only reveal itself if certain conditions are met.

All of the menu solutions have the fault that they might be crowded if there comes to many functions as the tool develops. This is one reason to think twice for every solution implemented if they are really necessary.

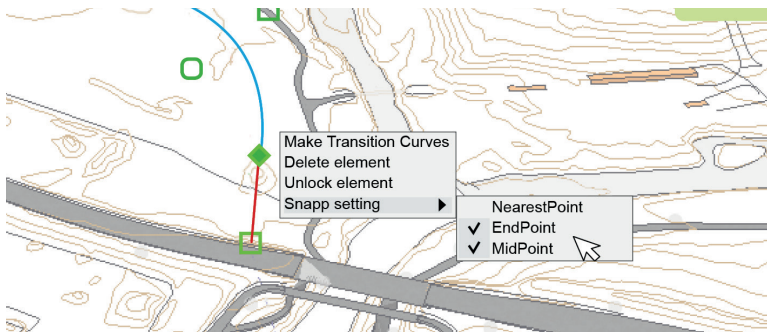
User insight: Both the terrain shaping and rod net have gotten good feedback from their users for their workflow.

When showed the widget solution people have really appreciated the quick keys of the widget as it is clear that it cater to expert users I have got many positive feedback that this seems like a nice way to work, and people like that it cater to an expert user, and that they can work efficient.

Design suggestions:

- Combine the use of both widget and right-click-menu, but have a good reflected discussion over how the workflow in Novapoint Base should be

- Have a group to look at these to make a design document to make sure that Novapoint Base gets a coherent design



Example of right-click-menu and selection of snap settings

The Line Elements

Dilemma: The line plays the main role in this tool, how does its graphical presentation communicate with the user?

Discussion: The main way for the line to be able to communicate anything is through color, thickness, feature type, and by the points and draggers coming along with it. There are a lot of information that the line could be communicating to the user through these variables like its feature type, what kind of line elements it consist of, its selection status, if it is constrained or not, warnings etc.

As the communication assets of the line is limited, it can hardly show all this information at the same time. In Novapoint Base the settings of how objects are drawn on the screen can be adjusted in the presentation settings. However even if the presentation setting would let the user change between that type of information, there should be a quicker way for the user to switch between preset settings.

Another way the line could communicate with the user is by making a "ghost" line to give a preview of what will happen when the user is about to place or edit a line.

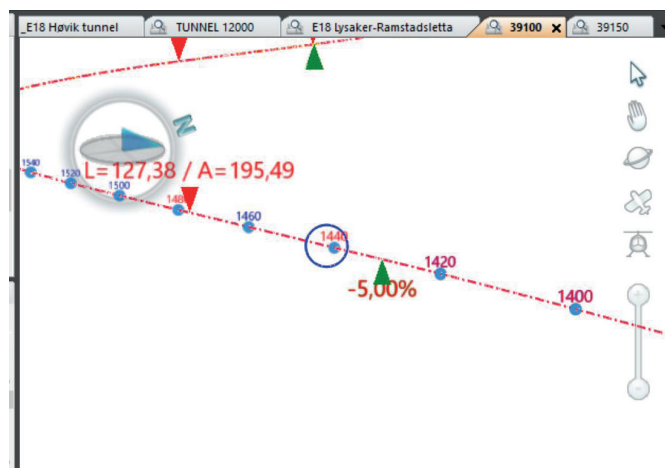
Design suggestions:

- The line should be able to present various information and the user should be able to quickly change between relevant settings.

The Labels

Dilemma: There are many labels that can be used to describe the line, how should they work and how do they communicate with the user?

Discussion: The labeling on the 3D alignment line in Novapoint Base today is hardly readable. It has symbols over and under the line to indicate the horizontal/vertical line, however as there is nothing to indicate which sections of the line these numbers are true for they get more useless. There are also few variations in the markings, which make it challenging to see the difference in the various values.



Various user groups would also have different preference for how the line should or shouldn't be labeled, for example are there several different methods to show the slope both in percent and ratio.

It is important with good markings, but if it becomes too much the whole screen will seem cluttered and the user will have a hard time interpreting what's going on. One way of handling the many possible labels is to make them dynamic so that as

the camera is far away hardly any of them are present, but as the user zooms in more and more of the labeling get visible and increases in size. The size of the label should be easily adjustable because the scale in which these project is conducted in change a lot, the scaling could be done through a hot key like Ctrl+'+' and Ctrl+'-'.

Other thing that could get labeled and marked are various constrains that are used, for example with a lock.

User insight: When asked if they wanted many labels and what kind of labels the Areal planners said that they would like as little as possible, however they might want a small marking for the area that would be possible to expand to get the full information of the area.

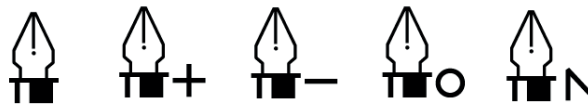
Design sugesstions:

- The kind of information shown through the labels should be adjustable.
- The labels should be scalable.
- The label needs to be readable.

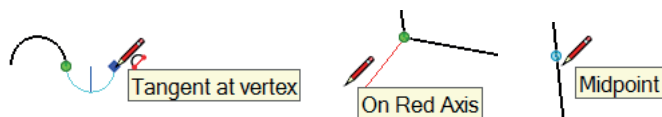
Mouse Cursor

Dilemma: The mouse cursor is an important tool for showing what the user is doing. How can it be utilized when working with a line?

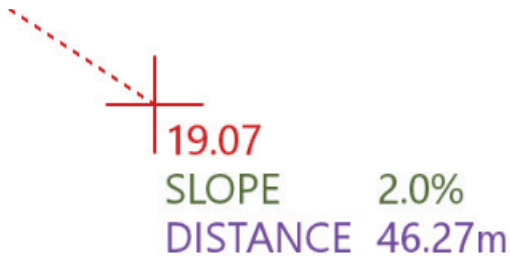
Discussion: Here are some examples of how the mouse has been used to communicate to the user:



In Illustrator the various tools have different mouse shapes, and depending on what it interact with, or if the user holds in certain keys the tool change mood and a small symbol is placed beside the main cursor to indicate which mode. (Harmer, 2013)



In SketchUp coloring and a small textbox tells the user about how it is about to snap thing in place and which relationship it would have to the geometry around it. (Picture from SketchUp help center)



In terrain shaping in Novapoint Base there are a dynamic numbers that shows the current settings and how what the new point placed would look like.

All of these strategies can do their work at communicating with the user. Having a mouse change slightly like adding a symbol like the illustrator pen, or a small text box like SketchUp depending on what it will do when the mouse is pressed, gives the user better control.

A good think about the Terrain shaping strategy is that it gives the user easy access to the most relevant control data. These labels would most likely only be visible when the line tool is used, and we might let the user adjust it in the visual properties or in the option menu of the line tool.

The good thing about mouse cursor is that this is the place where the user is looking so then he/she doesn't need to move his glance. Another thing is again to not go overboard and clutter it. Make sure that only the most important is there and don't do too much strange, the user needs to understand why it's changing

Design suggestions:

- The mouse should get fitting and dynamic changes that can help the user in his or her job.

Making it Smart

Making the software smart by having it help the user, and do thing automatically. The computer has to react to the user's input and facilitate their workflow.





Constrains

Dilemma: How can things be connected, and the user have control over the line and that it works as intended? This is where constrains comes in.

Discussion:

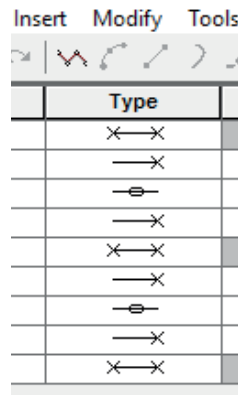
Lines can mainly have relationships with other lines, points or surfaces. These can be established when the user makes the line, for example the line can snap into an end point of another line or follow a surface with a given distance above it. After a line is placed out, there is a question if these relationships are permanent or not. If they are permanent, the line will have to be updated if the objects it is related to are changed.

There are already some snapping methods implemented into the Novapoint Base's measure tool, but looking at other program it could have been many more variants. Solidworks is a program with a rich library of snapping and mechanism for making constrains. These tools includes standard features like snapping to end- and midpoints, but also tools to that lets the line be drawn parallel, tangential or snapped to an given grid.

Distance:	<input type="text" value="3,25"/>		Distance from baseline to position
Slope:	<input type="text" value="3%"/>		Slope from baseline to position
Angle:	<input type="text" value="100g"/>		Angle between baseline and direction
Location:	<input type="text" value="67,50"/>		Location along baseline

A suggestion for what parameters that can be used to fixate a point to a line.

In the alignment design tool in AutoCAD there are one column called (fixation) type that tells how the line element is fixated, for example that the element is going through two fixed points or if it is an arc that is fixated in a fixed point. This shows that we could let the user lock and fixate the points defining the line to make them stable. These could be fixation could be made in the 3D room or locked in particular relationships with other geometries.



User insight: There isn't hard to find situations where users needs constrains, either to have control over how the line is made regarding to existing geometry, to make sure certain relationships stay the same, or to lock the position/geometry of an object to make sure it isn't unintentionally changed. One example we could mention is the scenario of the water and sewerage users, where a sewerage were going to be placed along a road.

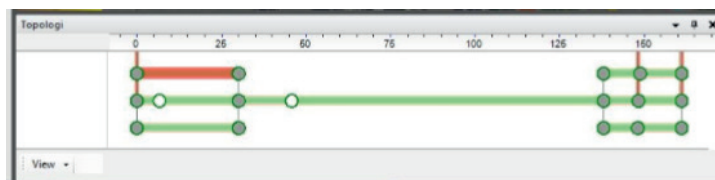
Design suggestions:

- It should be possible to lock elements and give then certain relationships towards other existing geometry. It should also be a fast operation to change these parameters and unlock them if needed.

The Topology

Dilemma: The structure of the model and how various objects are connected.

Discussion: The topology is about the models structure. In Road Net there is a separate window that shows how the various lines/roads are connected in a simplified schema.



Similar relationships between objects in the model in Novapoint Base exist when a lake edge is connected to the lake surface object.

In the line tool there would be more connection introduced through the constrains that were just discussed. These can be made by either connection to the overall room, a surface or along a line (a corridor).

Similar topology windows can be found in software like Maya, and there the relationships can be adjusted and changed directly in the topology window, which could also be a possibility for a topology window in Novapoint Base, however this is not the case for the window in Road Net.

User insight: The topology window gives the Road Net users good visual feedback about how the network is built up, which can be easier to read that since all the lines are simplified.

Design suggestions:

- There should be made a Topology window where the user can get an overview over how the objects are related and edit these relationship directly in this view.

Different Modules

Dilemma: Different users and tasks have different need for their line tool. How can the computer help in the various tasks?

Discussion: Today Novapoint offers several modules for different user groups which includes functions tailored for the various groups needs.

The line is a basic geometrical figure that almost all users of Novapoint needs to work with in one way or another. It is a basic function that needs to work. However how the different user groups want to use the line tool and what type of lines they will make are widely different. An area planner works mainly in the plan window using straight lines, and has to follow certain regulations for how the geometrical line should be looking, this is a completely different strategy compared to how the road engineer needs to have full control over the curvature and how the horizontal figure works together with the vertical.

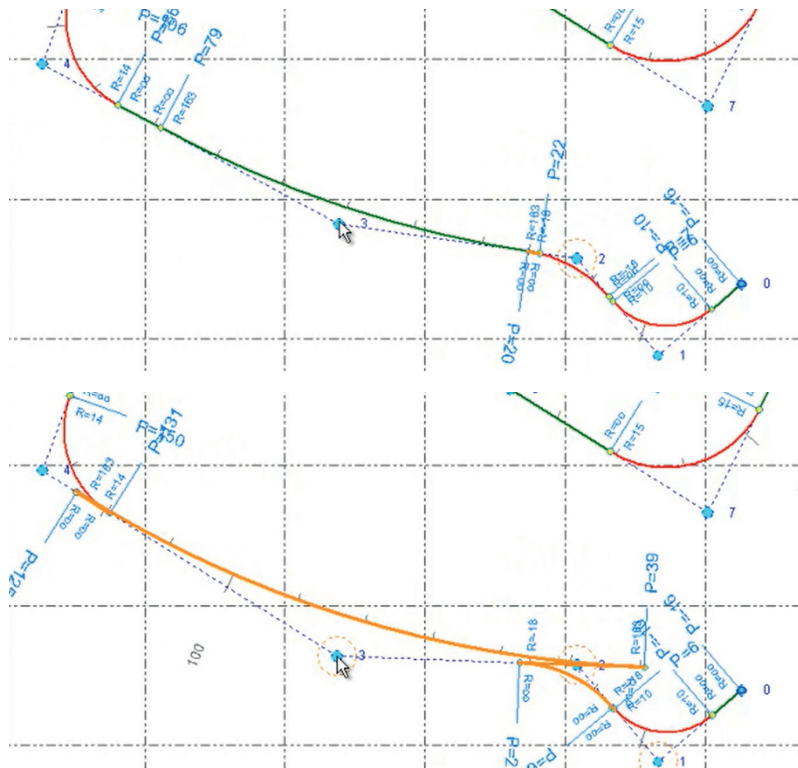
The basic principles for making a line would stay the same whomever the user are, but as they have different needs there should be possible to adjust the tool to fit each user. This might be done using various setting, or filter. Like the autocorrect language in Word, which all use the same writing program as base, but corrects the text depending on what language the user writes in. Theses personification of the tool can come in the form of autocorrecting, warnings and particular features that are special for the user group in question. An area planner would need a way to define the areas, which the other groups don't need in their tool.

Another thing that can be used to determine what functions should be available and a fit warning system are the feature types and how this can be used to differentiate between the users.

User insight: The landscape architect fighting against today's line tool because it is not made for their need.

Design suggestions:

- Make various version or filters of the line tool through which make the tool specially fitted for the user.
- Feature type can also be used to tell the computer what is, and isn't possible options when working with the line.



Automatization

Dilemma: How can the computer help the user doing things for them? It should understand what the user want to do and cater for it, and give warning if something goes awry.

Discussion: If the tool allows different moduls, it allows for the computer to give suggestions for how to work with the line depending on the typical regulations and workflow that is relevant for the modul and it's users. This can for example be giving warnings if a center line of the road doesn't follow the handbook which decribe the valid geometry.

Topology plays also an important role when it comes to automatization. The model should be updated if a line is edited and other lines depends on it. This might lead to some conflicts as changes in one place can lead to changes in the dependant geometry that are non inside valid parameters or would create impossible geometry. These problems the computer would have to share with the user and explaining what the problem is and give suggestion for how to fix it.

Many lines are made to be used in other tasks, for example to make roads. There is possible that a road model could be updated as the line is changed so that the user could get the imidiatly vizual feedback of how a change in the line would change the road. Another in function that would be particular useful to get an running update on as the line is edited is the mass calculation. Knowing the mass calculation can mean millions in saved in project cost. The importance of this has become apparent, as this has been the example most people I've talked with have brought up as something that would be useful for a smart line tool.

Pattern can be used to make geometry quickly, if it is a geometriacal figure like a square or constructing combinations of certain line elements. These could be updated and adjusted according to how the user use them, so if the computer notice the user change the pattern it might try to mirror this change

the next time the user uses the pattern. This can help the user be even more efficient, however if this isn't done right it can end up introducing mistakes that the user will not be able to notice.

User insight: Areal planner wanted several things automatized like splitting lines when it is appropriate, and connect the area descriptions to the corresponding lines.

Design suggestions:

- There should be a simple feedback system that the user can see about mistakes.
- Use the topology to update the whole model if something is changed. Eventually this should also mean updating functions like road constructions and mass calculation.
- Might also consider implementing patterns and templates to make the workflow more efficient.

Other Aspects

These are simply other aspects that I have had a hard time putting under one of the other category.

Feature Type

Dilemma: Feature type is something all object in Novapoint Base has. What does this mean for the line tool.

Discussion: Each object has a feature type. A list of every feature type can be found in Vianova System's object catalogue. This catalogue shows the properties to each feature type.

Earlier supporting lines were mentioned, this could work as a new feature type, or maybe a new more "featureless" geometry.

The screenshot displays the 'Feature Catalog' interface. At the top, there are navigation links for 'QuadriDCM', 'Easy Access', and 'Feature Catalog'. The main heading is 'Feature Catalog' with a 'Learn more »' link. A search bar contains the text 'line'. Below the search bar, there are two tabs: 'Feature Types (74)' and 'Attribute Types (106)'. The 'Feature Types' tab is active, showing a list of feature types. The 'RoadCentreLine' feature type is selected, and its details are shown in a panel on the right. The details panel includes a list of attributes and a table of properties.

Attributes:

- centreLine Curve
- attachment Struct (0..*)
- attachment Association (0..*)
- genericProperty Struct (0..*)
- propertyDescription WString
- propertyName WString
- valueBool Bool
- valueDateTime DateTime
- valueInteger Integer
- valueReal Real
- valueString WString
- information WString (0..*)
- link WString (0..*)
- metadata Struct
- validTimePeriod Struct

Properties:

No	0
PID	1310035
Description	line centered between two road edges
Short Name	
Start Date	Not defined

Screenshot of the Feature Catalog (Quadri^{DCM})

A question about the feature type would be what to do if the user would want to use different types of lines in one task. Should the default be the same? Should the user get a prompt if different line types is trying to merge, where they select which to go for?

User insight: People using the existing alignment tool in Novapoint Base found it strange that the only linetype available is reference type.

Both landscape architects and area planners uses a lot of different line type. Therefore, they were looking for a way to be able to select the correct feature. There is already possible to start writing things to have it pop up as a feature type, another way of doing it could be to use inspiration from software like Microsoft Words. Here a list of recent used fonts appears on the top of the list with all tasks. Including something like this would ensure that there is a quick way to change between frequently used feature types.

Design suggestions:

- Allow for different type of feature type in one task, and more flexible choice for feature type
- Have quick select options for the user when selecting feature types.

Multiple Elements

Dilemma: How to work with multiple lines and elements at the same time in the same task?

Discussion: Today there is possible to select more than one object in Novapoint base by holding down Ctrl as new objects are selected.

A relevant example from another program could be how Solidworks lets the user start a sketch were the user can draw several different line in the same sketch and edit them.

One problem that one of the user pointed out was the trouble today that if you had several of the same object and wanted to change a parameter of them in the property window you had to select one at the time, and for every object that was selected they would have to scroll down to the relevant parameter again for each time.

The options for what the user can do should and could change depending on what are selected, for example if two endpoints are selected the merge option is activated and the user can click on it to activate it.

In the vertical view the user could be able to change the view and maybe the active line as well by a simple click, just as the screen can be changed between the cross section views.

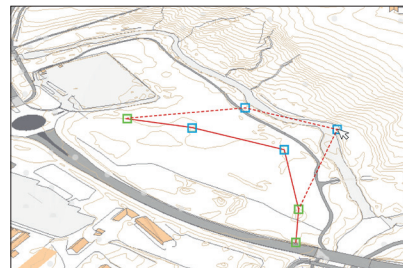
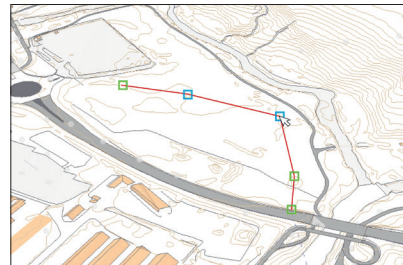
User insight: Areal planners needs to be able to work with several lines at the same time to create the areas. A possibility could be to create new tasks for each topology. In particular for areal planners they sometimes want a partial connection between the various topologies.

Design suggestions:

- Use Ctrl to select/deselect more objects/line elements
- Let the user be able to select more objects and then change the property of all of them if the user change it, or at least have the same property selected when several of the same types of objects/elements are selected.
- Change between the lines in the task in the vertical window.
- Have dynamic option of possible functions depending on what are chosen.

Properties					
Veranda					
Back Point List edge					
	Chainage	X	Y	Z	Length, 2D
1	0	98606,250	1213116,780	31,600	
2	1,024	98607,050	1213117,420	31,600	1,024
3	3,971	98608,940	1213115,160	31,600	2,946
4	13,546	98601,510	1213109,120	31,600	9,575
5	16,467	98599,660	1213111,380	31,600	2,921

Curve metadata	
Curve Capture Date	26.04.2009
Metadata	
medium	In air [2]
heightReference	Top [2]
positionQuality	
accuracy	320 mm
accuracy/HeightSymmetry	230 mm



Example how more than one element can be moved at the same time.

Undo

Dilemma: To let the user be able to regret an action and correct their mistakes, are of essential for good user experience, but how?

Discussion: From most modern software undo is a natural inclusion and helps the user feel more secure in their work, since they can always click undo if an operation fails.

Word and Photoshop are typical software that are known for their undo and history systems. They also has a history in today's alignment tool in AutoCAD alignment.

Some software lets the user undo selections as well as actions. This is something the new line tool will have to take a stance towards. One strategy to solve this is to let this be a setting the user might change at will.

User insight: From most people I talked to undo was an essential function if they were going to work in a line tool. Recently they finally have to include the undo function in the Terrain shaping tool with great positive response. The user really wants it, however in most tasks in Novapoint Base the undo function is lacking.

Design suggestions:

- Include Undo and redo, let the user use Ctrl+Z and Ctrl+Y as key buttons, as these are the standard buttons.
- Include a simple history, which might contain, and an adjustable buffer to make sure not all the work disappears, or the program turns slow.
- Let the user select if the undo should affect selections or not, with the default being that it includes selection.

The Design Brief

To sum up the findings of the chapter I will present a short design brief that shows some the key design aspects which a final line tool needs to implement in a satisfying manner.

Usability

The user should easily be able to understand and use the tool and be able to manipulate the alignment in the 3D environment.

Performance

The computer shouldn't be slowed down as the user work on the line.

Clarity

The tool should avoid graphical clutter, and the user should see clearly what happens and why.

Dynamic

As the line is made and edited, features that depends on the line should be updated as well.

Accuracy

The user needs to feel that they are in control, and be able to adjust the line by 0,1 millimeter accuracy.

Flexibility

Different use and user groups and should be able to solve all of their problems and get the help they need solving it.

Efficiency

The user should be able to work in a straightforward manner, and be able to have a swift workflow.

5. CONCEPTS

The main concepts and ideas for the new line tool and how it can be developed.

The Concepts

This chapter will present the concept of how the new line tool in Novapoint Base could look like, or rather it will suggest a plan for how to develop the tool and sketch up an desired end result. It became apparent throughout the project that I could not just simply try to describe the idealized line tool, it would be too complex, and it would be hard to get it validated without thoroughly implemented and tested it. Therefore I choose to present a sketch of this idealized state, and supplement this with a more concrete plan of how to implement simpler versions of this and describe a plan for development.

Last chapter analyzed various aspects of the line tool presented some design suggestions, these concrete solutions will not necessary be mentioned in this chapter. This doesn't mean that the suggestions should not be implemented, but that it would take unnecessary much space in this report to repeat it.

How to Develop it

A suggestion for how to develop the tool would be to follow these 7 steps:

1. A basic poly line tool in 3D
2. Be able to handle more than a single line in a task
3. Adding an arc to the line
4. Adding transition curves
5. Line from points
6. Control tools
7. Making it smarter

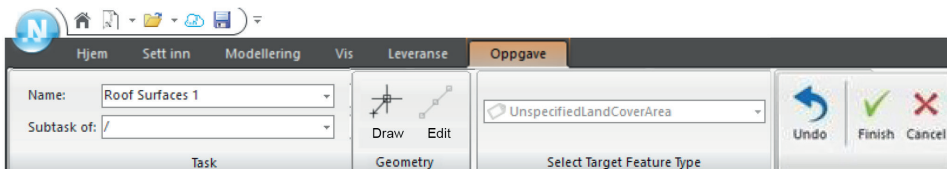
This implementation sequence would start with a simple, but still usable line tool, and later be developed with increased functionalities, and refined through testing and user feedback. It is worth mentioning that many of the features could have been implemented in another order. For example some feature for making the line tool smarter, for example by warning against certain mistakes could be implemented already after the basic linestring tool is implemented. This could be to warn about a too steep slope when a landscape architect is making a ramp.

1. A basic polyline tool in 3D

First thing that should be implemented is a basic linestring tool, that lets the user create and edit a linestring. This should be fairly easy to do as there are already existing drawing procedures in Novapoint Base for this. To make a simple tool like this even more useful an easy way to export this to AutoCAD should be made so the users can sketch up a line in the Novapoint Base environment and then refine the line in AutoCAD. It is worth to comments to that some of the functions included in this and lather phase, have been included because they can be used to test out how the user would interact and work with the line in 3D.

1.1 Making and adjusting a basic polyline

This could be based upon the same principles as the ruler or the line creation in the long section task. That means we can use the already implemented snap functions and the undo function. The edit functions lets the user move already placed points to new positions. The ribbon for this task would look something like this:



Ribbon:

1. Task

Gives the task a name and places it in the model's hierarchy.

2. Mode

Changes between a draw mode and an edit mode.

3. Select Feature type

Change and selects the feature type of the line.

4. Finish

Ends the tasks and lets the user undo his/her actions.

Right click (in the various views):

Set the snap settings

Hot key:

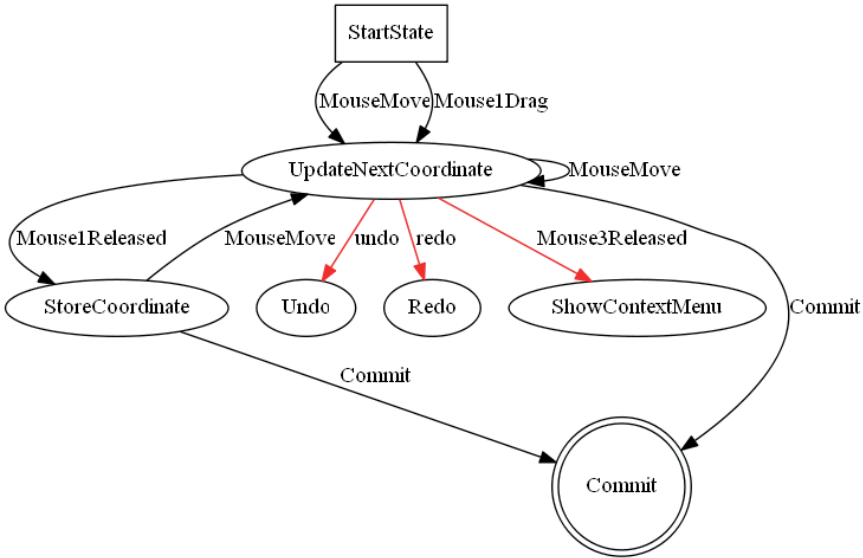
Ctrl+z: undo

In addition draw/edit should eventually get hot keys so user can quickly change between these modes, but these are not necessary from the very beginning.

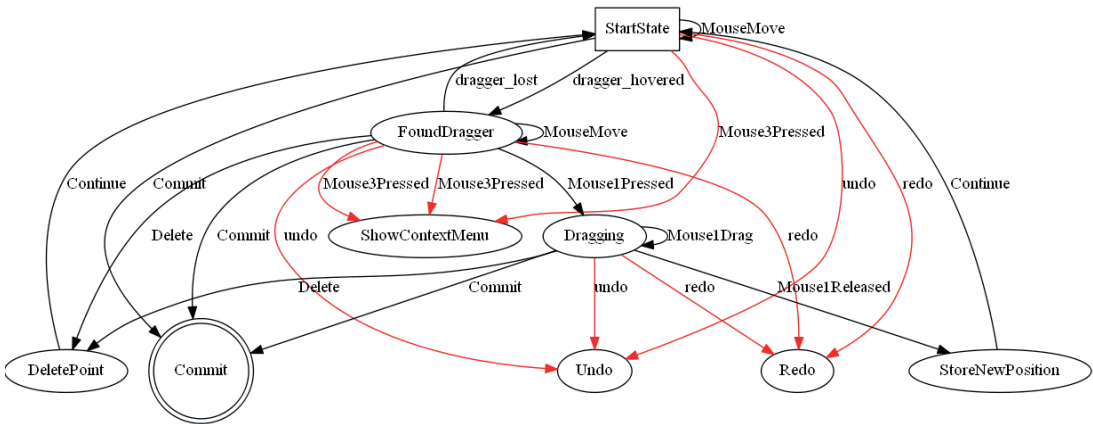
Finite state machine and prototype

The plan was to have the first part of the line tool implemented, and be able to test how user would be working with it in 3D. Due to a miscommunication were both the programmer and I thought the other one were working on the prototype, this did not happen. The dialog I had with the programmer were however fruitful, and gave me a good insight in how Novapoint work. One important aspect is that the program works using finite state machine. This means that the computer goes in different states were it waits for certain inputs to change to another state. On the next page are two finite states I made to illustrate how the finite state machine to the draw function or the edit function of the basic polyline tool could look like.

Draw a line



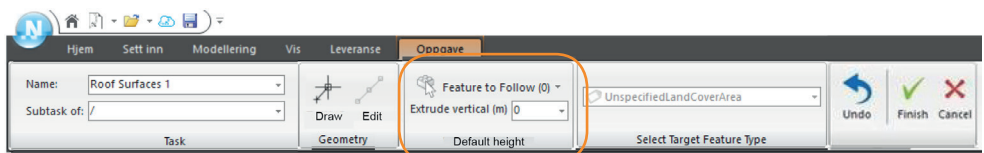
Edit a line



Finite State Machines (FSM) of both draw a line and edit a line.

1.2. Giving the line a default height

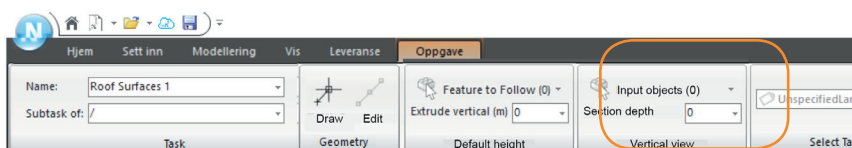
The line should have the possibilities to get a default height. This should work so that the height of the line is given from the feature it is following. This would work the same way as the existing tool in Novapoint base “move vertically”, It should be able to give it an offset as well



The trouble with this is how it should work compared to snapping, but for now I think we should simply be able to turn default height on/off. Then it will either goes to the default height or the snapped height.

1.3. Having a vertical view of the line

I would like to have the line task open a new view that shows the vertical height of the line. Working in the same way as the long section, except it shows the active line and it's possible to select it's points and adjust the height.



It should have the possibility of selection which objects that shows up in the vertical view and maybe the section depth as well.

1.4. Changing between ways of moving points

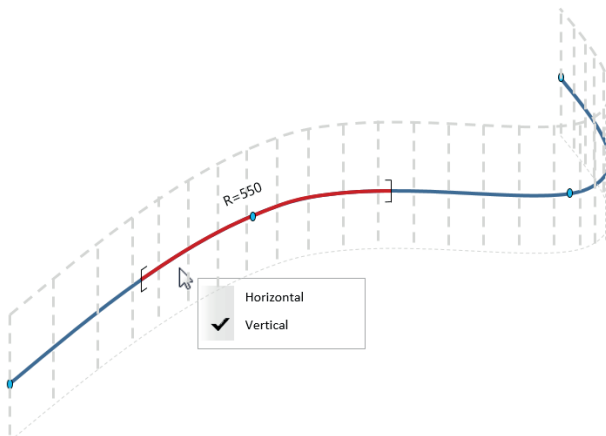
We could apply an restriction to the plane and vertical window to make sure that when a point is moved in the plane window the height won't change without the user's knowledge, and similar in the point can only be moved up or down.

I would like to be able to shift between various modes of moving the points in 3D to simulate similar restrictions:

1. Using the standard snap method (without other restriction).
2. Restrict the movement to the horizontal plane.
3. Restrict the movement to vertical.

This is one thing I would like to test to see how it works regarding giving the user more control over how to make and edit the line in the 3D view.

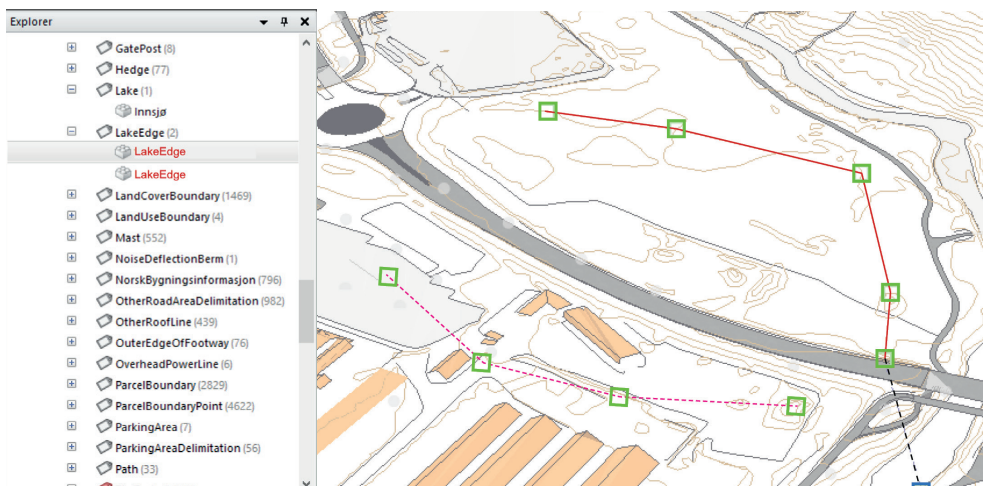
Hot keys can be used to switch between the two editing mode.



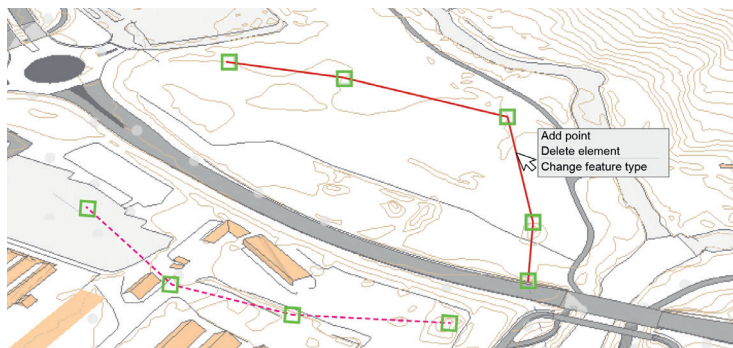
2. Be able to handle more than a single line

2.1. Being able to draw and editing more than one line

There should be possible to work with several line in the same task. This is essential to be able to use other earlier made lines to create new lines, and to make many lines more efficient.



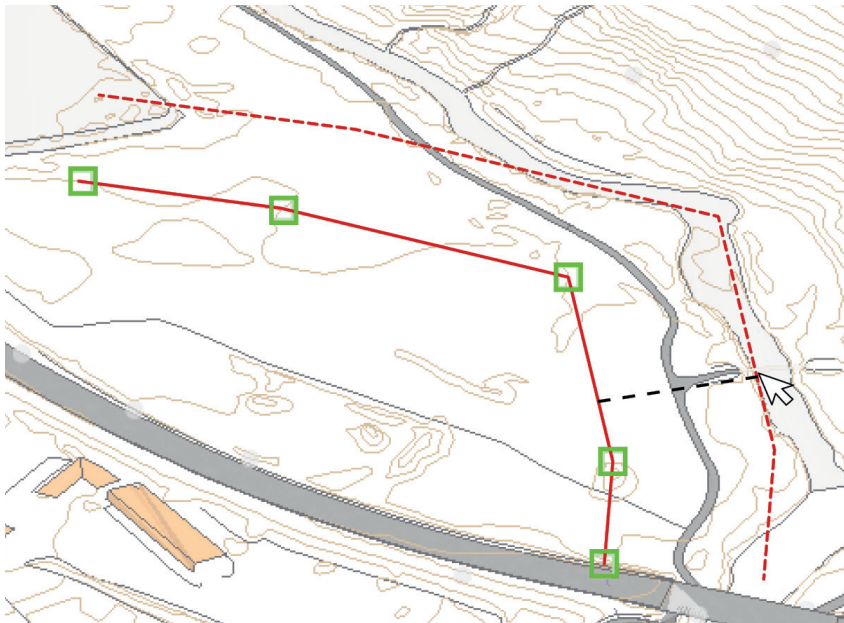
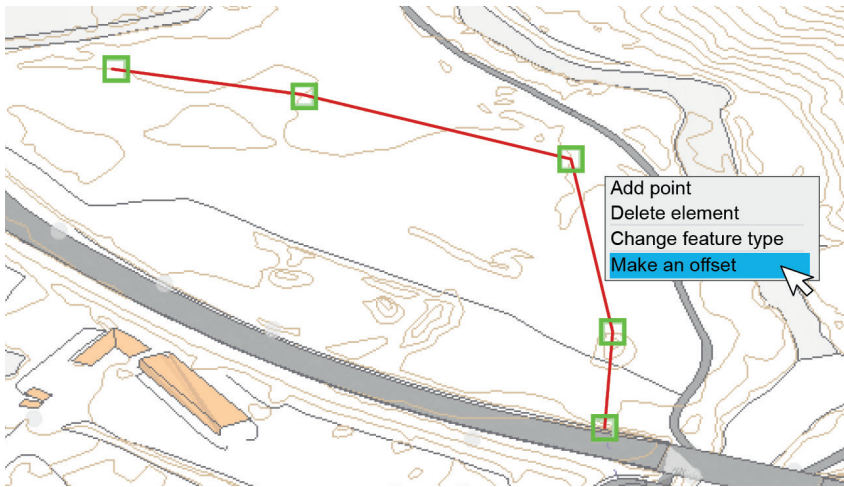
Example on two lines that are being worked on. The lines that are in the tasks are red in the Explorer, and the one chosen is marked, and will also be active in the vertical window.



By right clicking on an element on the line a menu should pop up with various possibilities like changing the feature type or editing the line.

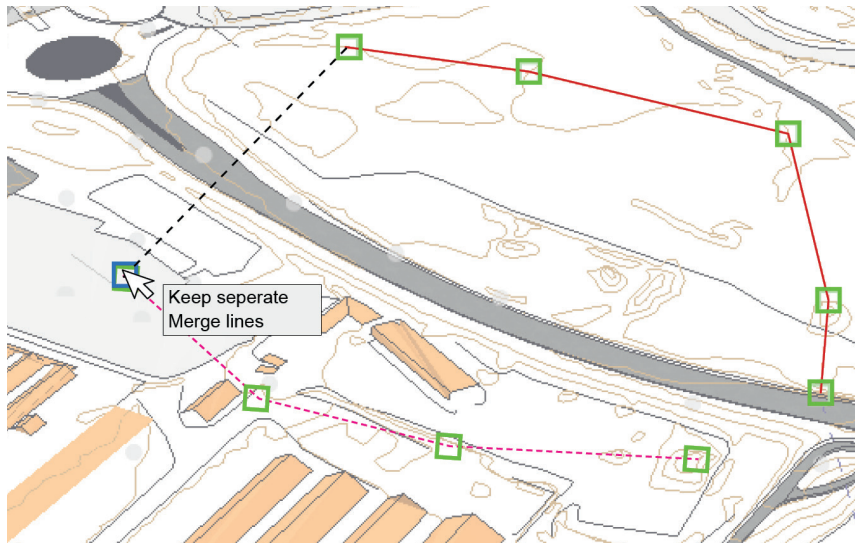
2.2. Copying existing lines / making offsets

There should be a possibility to make an offset from existing lines.



2.3. Append lines and glue them together

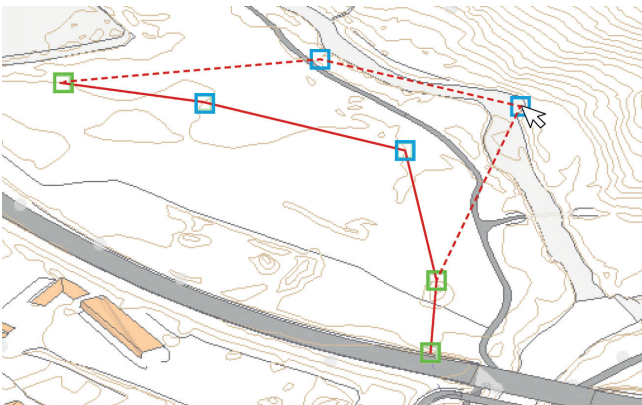
There needs to be a way to glue/merge various lines together.



2.4. Adjusting several elements at the same time.

There should be possible to select more than just one point by for example using ctrl while clicking on points. When selected they can be moved around all together at the same time.

Properties					
Veranda					
Point List edge					
	Chainage	X	Y	Z	Length, 2D
1	0	98606,250	1213116,780	31,600	
2	1,024	98607,050	1213117,420	31,600	1,024
3	3,971	98608,940	1213115,160	31,600	2,946
4	13,546	98601,510	1213109,120	31,600	9,575
5	16,467	98599,660	1213111,380	31,600	2,921



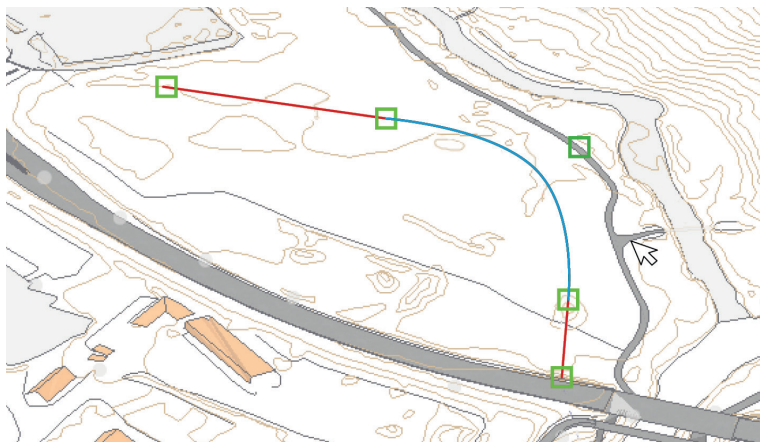
Explaining how the various elements can be selected. Here illustrated with a selection of two points that are moved a little.

3. Adding an arc to the line

We want to add arc to the mix and with this we split the line into a horizontal and a vertical description that combined makes the line. This would mean that the line has to be adjusted horizontally or vertically, but not both at the same time. Drawing arc could be done in various ways.

Properties								
Alignment								
← Back Vertical elements alignmentGeometry								
Element	From	To	Radius	Length	Parameter	Start Height	End Height	
1	Straight...	-172,081	-22,131	0	149,949		35,000	35,000
2	Arc	-22,131	-22,030	6,000	0,101		35,000	35,001
3	Straight...	-22,030	127,884	0	149,914		35,001	37,535
4	Arc	127,884	127,911	2,000	0,027		37,535	37,535
5	Straight...	127,911	865,358	0	737,447		37,535	40,000

The Property window showing the Vertical elements that now includes Arc elements.



Right click to make an arc from a corner.

4. Adding transition curves

Making transition curves with varying radius. This increases the complexity and we have to find a good way of making them editable.

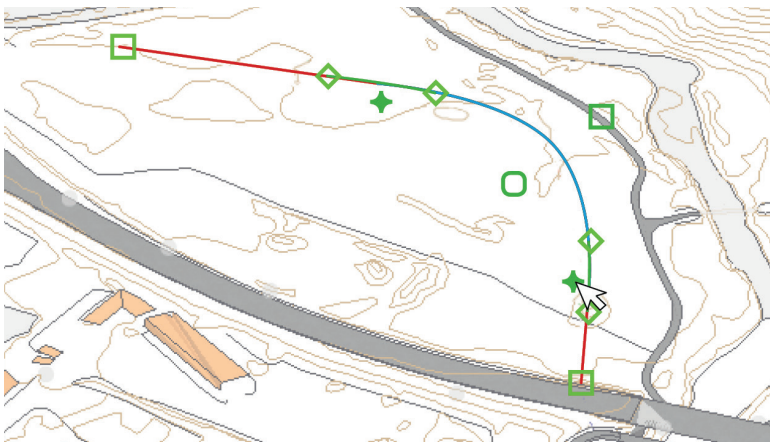
Properties

Alignment

Horizontal elements alignmentGeometry

	Element	From	To	Radius	Length	Parameter	Start X	Start Y	End X	End Y
1	Straight Line	0,000	93,343	0	93,343		99000,910	1212548,850	98952,918	1212628,911
2	Clothoid	93,343	109,343	0,000 > ...	16,000	<u>21.909</u>	98952,918	1212628,911	98945,964	1212643,264
3	Arc	109,343	143,159	30,000	33,816		98945,964	1212643,264	98955,138	1212673,978
4	Clothoid	143,159	159,159	-30,000 ...	16,000	<u>21.909</u>	98955,138	1212673,978	98968,825	1212682,166
5	Straight Line	159,159	273,793	0	114,633		98968,825	1212682,166	99072,033	1212732,054
6	Clothoid	273,793	289,793	0,000 > ...	16,000	<u>21.909</u>	99072,033	1212732,054	99085,720	1212740,242
7	Arc	289,793	312,647	30,000	22,854		99085,720	1212740,242	99095,881	1212760,099
8	Clothoid	312,647	328,647	30,000 ...	16,000	<u>21.909</u>	99095,881	1212760,099	99094,515	1212775,990
9	Straight Line	328,647	561,396	0	232,749		99094,515	1212775,990	99054,080	1213005,200

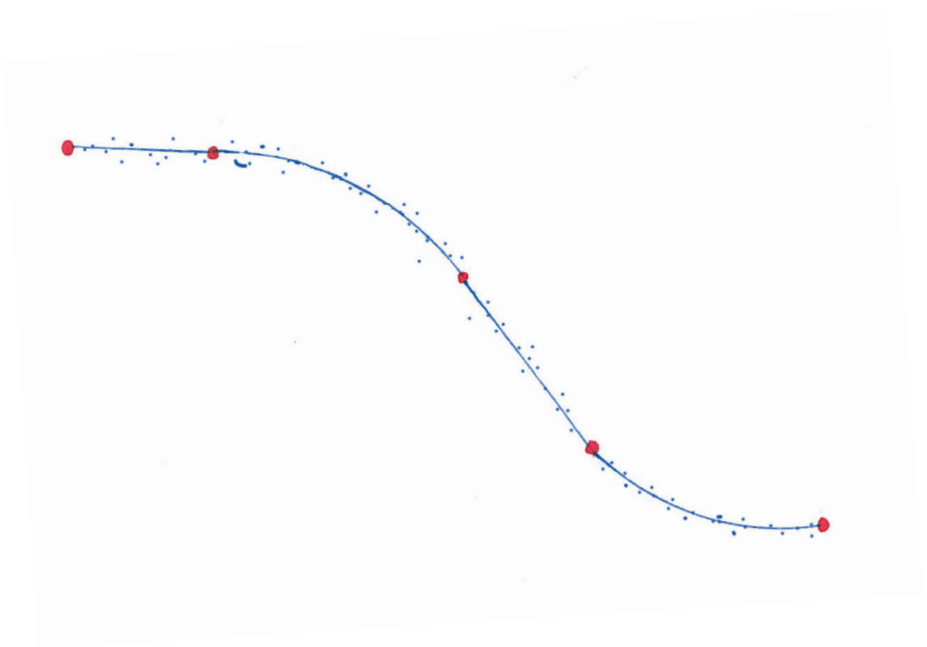
The Property window showing the Horizontal elements that now includes some transition elements (Clothoids).



Added transition curve

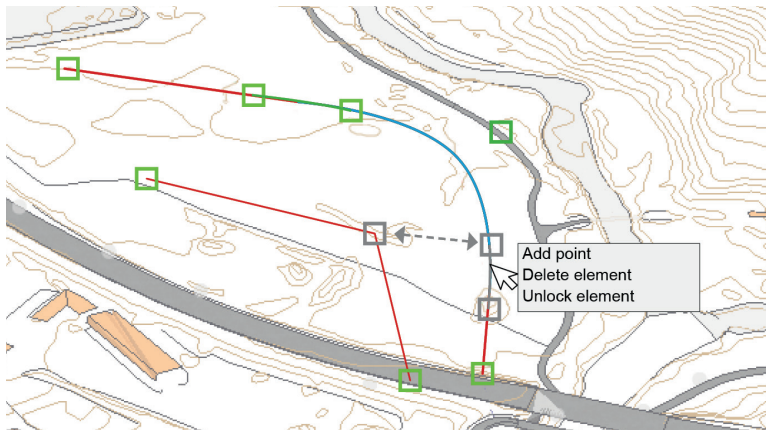
6. Line from points

We need a way to calculate a curved line given points, this can be utilized to create many useful tools like a nudging tool, derive a line from measured points or a freehand tool that can be translated into an editable line.



7. Control/restriction tools

We need some sort of control tools to make sure that things that should stay a certain way doesn't get changed, for example lock points in place or lock distances between lines.



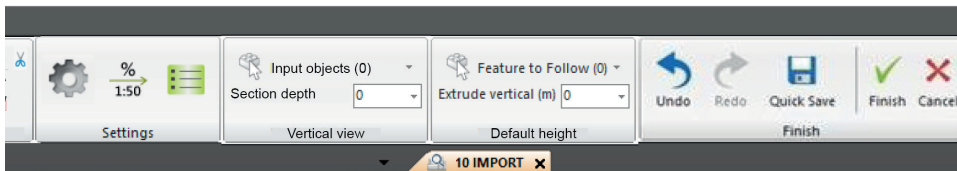
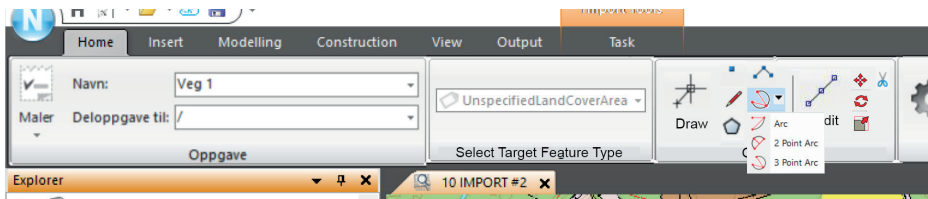
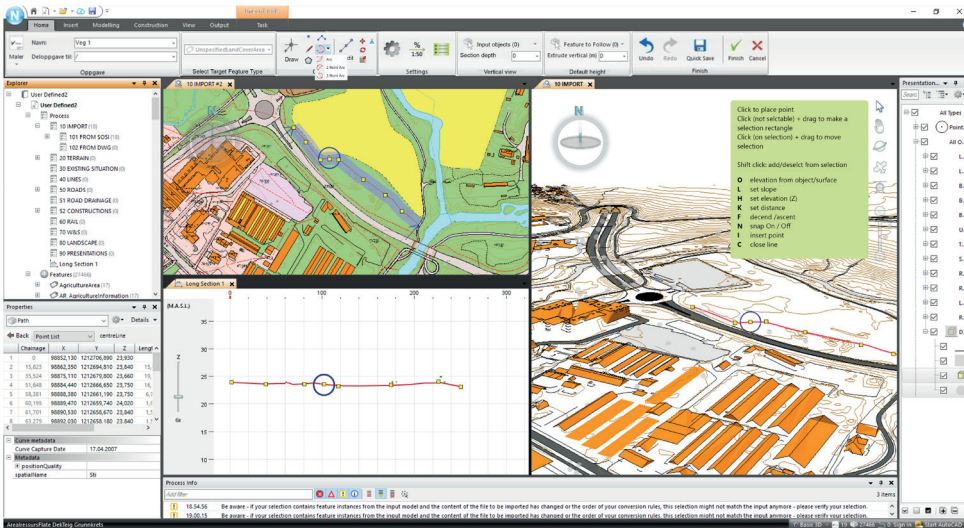
Explaining how elements could be locked (grayed out) to make sure they keep their position. There could also be locked relationships exemplified with the arrow between the two lines.

8. Making it smarter

The users demands a smarter solution in Novapoint Base and its new line tool. The software has to be able to react to all of the user's input and facilitate their workflow, by not only adjusting the line in question, but the whole model. As the center line of a road is adjusted, the user would expect to be able to how the changes affects the whole model. The sewage system should follow the new path, and the cost of the mass transportation should be calculated. To be able to make this happen Vianova Systems would have to look at how the model data is handled and figure out how to conduct such heavy computation in real time.

Another aspect of making the line tool smarter is to create more specialized tasks in Novapoint Base for certain user groups. For example will the area planners need a tool where they can draw boundaries and connect them to defined areas. The planners would also benefit from a specialized task that are able to auto fix the line to fit the discipline's requirements

The Main Vision



The overall proposed solution of a line tool is a tool where the user easily can start drawing a line in both 2D and 3D. Using various drawing and editing tools the user can create lines and observe how it dynamically updates the tasks that are dependent on the line and the 3D-model. The software will be smart and help facilitate the user's work. This tool should fulfill the design brief and have a great performance, be clear and easy to operate, give the user dynamic feedback, be accuracy and flexible in use, and give the opportunity for an efficient workflow.

6. EVALUATION

An evaluation of the project and the resulting concept.

Evaluation of the Concepts

My concept of the final line tool is not highly detailed, but instead includes some guidelines for what we would need in the future and some suggestions about how to do this. The earlier part of my development plan is more detailed, as these are concrete suggestions about how to implement the most basic aspects of the tool, testing these on users would have given feedback that could have been used to further develop the line tool and would influence how the more complicated operations would have been addressed.

Sadly I didn't get to make prototypes I didn't get to conduct the user tests as planned. I did however get to show parts of the overall main concept to people and talk with several about my suggestions for developing the line tool. Talking with the programmer about prototyping revealed some aspects of the ideas that had to be slightly adjusted. I'm not sure how the computer will be able to work with the dynamic tool that should be the end result, because of the heavy computation that would have to be done fixing and updating the whole model. This is something that we would have to work out as we develop the tool to find fitting strategies of how to handle the base model of the software, in particular if this has to be changed. Given that we'll manage to develop a tool that is user friendly and reliable the people I've been in contact with show great interest in it.

I can illustrate some of the feedback from potential users by this one response from a landscape architect that had trouble figuring out how to work with Novapoint and AutoCAD to change a stream: "Working like this would have been so much more visual. Then I could have placed a basic line, and then put in a fall here and have a flat plateau there. By working like this, we could get the line construction done in a smooth operation. This tool would have been perfect for making a creek."

I also talk with Hallgeir Hjelle about my master's thesis, having my concept in front of me. Hallgeir is employed by Trimble, but not a part of Vianova System, he has been working with 3D lines for a long time, wrote a phd-thesis about how to geometrical modelling of roads in 3D, and he has also done some prototypes illustrating how lines can be worked on in 3D, which I have referred to earlier in this thesis. During my talk to him I got told about some of his thoughts about how a good line should work, and he pretty much described a lot of my earlier conclusions, and also when we started talking about some of the aspects that would be needed to be included he mentioned several, all of which I had already included in my plan for how to develop the line. Of course there were some new elements that was uncovered during this talk, as all my interviews and discussions throughout my studies have provided. This tells me that as we develop the line tool, we can never have everything under control and some challenges and problem will only be uncovered as the tool take form and users starts to work with it.

Overall, I think that my concept and ideas could have been more detailed, but as the project grew as I was working, I did not find time to explore all of the aspects in details or make room for every feedback and possible suggestions in my report. I believe have gotten the impression that my thesis and concept will be of great use as Novapoint System now starts to implement and work with the new line tool for Novapoint Base.

Evaluation of the Process

I'm sure that like almost all master students, I too feel that I could have done a much better job if only I could have done it all over again. This section will mostly include everything about the process I believe I could have done a better job with, but I did manage to do something so I want to just start this section by giving myself a pat on the shoulder to acknowledge that I survived and somehow manage to have something to hand in at the end.

Writing this project description, I was looking forward to explore a new topic, and engage with the people of Vianova Systems and their network of users to uncover a suitable solution for how the implementation of a line tool could be done in Novapoint Base. The meetings I've had with people has been the main driving force for me during this thesis. People has been engaging and really showed me a pride of their work, and shared their dreams and hope of how a new line tool could look like. I really wish that I had spent more time talking to people, and derive more of their enthusiasm into my master's thesis. However, though each meeting filled me with glee, I also got a huge load of feedback that I needed to process.

During my thesis, I discovered that a line wasn't simply a line. Making a line in a 3D software is something completely different from drawing a line on the paper. At times during my work on this thesis this has felt discouraging and I've had a hard time making myself motivated to but in the effort I feel this master's thesis deserved. I had only half a year to delve into this, and I'm sure that this topic without a problem could have been used a basis for a PhD. This means that I have not been able to explore everything in depth, but rather try to get an overview over the most important aspect.

One of the bigger obstacles for me during this thesis was the failure to create a proper prototype like the one I had wanted. This was due to a miscommunication and I blame myself for it. I should have given clearer directions, and double-checked that everything was going according to plan. However miscommunications happens, and there was sadly little to do to fix the problem in time to be able to get workable prototypes to test.

Something I could have done during this project was to get a deeper understanding of how the code and the underlying system of Novapoint Base worked. This could have helped me to better understand the limitations of the software, and I would have been able to code the prototype myself. However I had chosen from the very beginning that I wanted to mainly focus on the user interface trusting that the programmers of Vianova Systems could handle the coding part, so I never found time to delve into this aspect. If I had started with the master over again, this is one of the things I would have made sure to learn from the very beginning.

In the end I wish that I could have given 110 % from the very beginning, however I think that I still would have needed more time to refine my report and would be fully aware of how much that could still be included.

The Future

Finishing of this thesis, I will now hand over this report over to Vianova Systems. I know that they will start working on implementing a line for Novapoint Base in the near future, and I hope my findings and conclusions can be of use for them.

My suggestions for them would be to start with making a simple line tool that allows the user to quickly create a line string and develop the tool from there. In parallel and as an extension of the work with the line tool, someone should work on a design document for Novapoint Base, and figure out how the basic model should work as the software is getting smarter. The design document should outline how task should be implemented in Novapoint Base to make sure the user gets a coherent experience when working. A guideline will also make it easier to move functions from AutoCAD to Novapoint as some of the choices about how the user should work will already be given.

I believe that Novapoint Base soon will have a line tool that can be of help for those planning and designing new infrastructure project.

REFERENCES

List over books, articles and webpages that have been used as a source of information

The List

Some important source of information in this project have been internal documentation of Vianova Software, and various software. The internal documentations have been learning materials and various development files. Included in this list are also links to the homepage/help-page for the various systems studied in this Master's Thesis.

If nothing else is noted the pictures are screenshot of Novapoint, illustration taken from one of Vianova Systems internal documents or made in particular for this project.

Adobe. "Illustrator." <https://helpx.adobe.com/illustrator.html>.

Autodesk 3DS max. "Learn & Explore." <https://knowledge.autodesk.com/support/3ds-max/learn-explore>.

Autodesk AutoCAD. "Learn & Explore." <https://knowledge.autodesk.com/support/autocad/learn-explore>.

Autodesk Maya. "Learn & Explore." <https://knowledge.autodesk.com/support/maya/learn-explore>.

Bentley Rail Track. "Rail Track Design and Analysis Software." <https://www.bentley.com/en/products/product-line/civil-design-software/bentley-rail-track>.

Colborne, G. 2011. Simple and Usable: Web, Mobile, and Interaction Design: New Riders.

Dassault Systems. "Solidworks Help." <http://help.solidworks.com/2016/English/SolidWorks/sldworks/>.

Harmer, Tony. 2013. "Illustrator Pen Tool Chart." <http://www.academyclassblog.co.uk/pen-tool-reference/>.

Hjelle, Hallgeir. 2002. *Geometrisk modellering av veger i 3D*.

Lovdata. 2010. "Forskrift om tekniske krav til byggverk (Byggteknisk forskrift)." <https://lovdata.no/dokument/SF/forskrift/2010-03-26-489>.

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

Novapoint "Resource Center." <http://help.novapoint.com/>.

NS-EN ISO 9241-210:2010. 2011. *Ergonomics of human-system interaction - Part 210: Human-centred design for interactive systems (ISO 9241-210:2010)*: Standard Norge.

Oxford Dictionaries. "Line." Accessed 05.03.2016. <http://www.oxforddictionaries.com/definition/english/line>.

Quadri^{DCM}. "Feature Catalog." <https://www.quadridcm.com/FeatureCatalog>.

Robert, McNeel. "Rhinoceros for Windows User's Guide." <http://docs.mcneel.com/rhino/5/usersguide/en-us/index.htm>.

Rogers, Y., H. Sharp, and J. Preece. 2011. *Interaction Design: Beyond Human - Computer Interaction*: Wiley.

Russell-Rose, T., and T. Tate. 2013. *Designing the Search Experience: The Information Architecture of Discovery*: Morgan Kaufmann.

Shneiderman, B., and C. Plaisant. 2010. *Designing the User Interface: Strategies for Effective Human-computer Interaction*: Addison-Wesley.

Site3D. "Getting started." <http://www.site3d.co.uk/gettingstarted>.

SketchUp. "Help Center." <https://help.sketchup.com/en>.

Sommerville, I. 2011. *Software Engineering*: Pearson.

Toftøy-Andersen, Eli, and Jon Gunnar Wold. 2011. *Praktisk brukertesting*: Cappelen Damm Akademisk.

Trimble Quantm Software. "Alignment Planning Solutions." <http://www.trimble.com/alignment/index.aspx>.

Vianova Systems. "Vianova Systems." <https://www.vianovasystems.com/>.