Report 3

David Aldenbro Pontus Lüthi
Sofia Andersson ~~Leonard Petersson Hoogendoorn~~
Linus Dinesjö Alexander Skalski
Elias Floreteng Gustaf Sundmark

Project Group 20, DD1369
7th May 2023

1 Abstract

When a disaster happens it is important to gain information on how difficult it will be to rescue people. Today this can be done with drone imagery and estimations. I-CONIC has created a tool that takes the drone videos and converts them to 3D-models. We will build upon this solution by creating tools to measure heights, lengths, areas and volumes in the 3D-models generated by the software that I-CONIC has created. The goal is for the interface to be user-friendly and allowing for a variety of measurement-types such as points, lines and polygons. This project is coded in C++ and uses wxWidgets and OpenGL to create the user interface and the measuring tools. We will create both the user interface that allows easy selection of the different measuring tools and the measuring tools themselves. This project is done by a group of 7 (formerly 8) students from KTH (Kungliga Tekniska högskolan) that are studying Computer Science, under supervision by Dr. Christopher Peters, and in collaboration with the company I-CONIC. Since the last report test cases and notes on change management have been added. For further notes please see §**??**.

Contents

1 Abstract **1**

2 Glossary **5**

3 Introduction **6**

3.1 Project summary 6
3.2 Purpose 8
3.3 Company background 8
3.4 Market analysis 9
3.5 User group 13
3.6 Project overview 14
3.6.1 Intended functionality of finished program 14
3.6.2 Requested features 14
3.6.3 Program class overview 15
3.6.4 Dependencies of the project 17
3.6.5 Current state of the project 18
3.7 User-friendly Interface 18
3.7.1 Good Design 19
3.7.2 Bad Design 19
3.7.3 Accessibility 19
3.8 Scope of the project 20
3.8.1 Limiting the scope 20
3.9 Members and skillsets 21
3.10 License 22

4 Requirements **22**

4.1 Project modules 22
4.2 Business requirements 23
4.3 Functional requirements 24
4.3.1 Interface module 26
4.3.2 Display module 29
4.3.3 Functionality module 31
4.4 Non-functional requirements 34
4.4.1 Interface module 34
4.4.2 Display module 34
4.4.3 Functionality module 35
4.4.4 External module 36

4.5 Use cases . . . 37
4.6 Testing and verification . . . 39
4.6.1 Test cases . . . 39
4.7 Traceability matrix . . . 46

5 Project Plan **48**
5.1 Software engineering methods . . . 48
5.1.1 Scrum . . . 48
5.1.2 Code review . . . 50
5.1.3 Pair programming . . . 50
5.1.4 Refactoring . . . 51
5.1.5 Prototyping . . . 51
5.1.6 Management methods and organizational structure . . . 52
5.1.7 UML diagrams . . . 54
5.2 Course deadlines . . . 54
5.3 Project timeline . . . 55
5.3.1 Deliverables . . . 55
5.4 Communication . . . 56
5.4.1 Services overview . . . 56
5.4.2 Within the group . . . 57
5.4.3 With the client . . . 57
5.4.4 With our mentor . . . 59
5.5 Documentation of meetings . . . 59
5.5.1 Internal meetings . . . 65
5.5.2 Mentor . . . 65
5.5.3 I-CONIC . . . 66
5.6 Responsibility roles . . . 66
5.6.1 Reflection on the responsibility roles . . . 67
5.7 Project journal and reflections . . . 67
5.7.1 Problems with hardware and software requirements . . . 67
5.7.2 Sprint 1 . . . 68
5.7.3 Sprint 2 . . . 68
5.7.4 Departure of a group member . . . 68
5.8 Change management . . . 69
5.8.1 Appointment of group leader . . . 69
5.8.2 Reevaluation of time distribution . . . 69
5.8.3 Client rescinding a requirement . . . 69

6 Risks **70**
6.1 Difficulty in development . . . 71
6.2 Production issues . . . 71
6.2.1 Risk encounter . . . 71
6.3 Software/hardware compatibility issues . . . 71
6.3.1 Risk encounter . . . 72
6.4 Goal misalignment with stakeholders . . . 72
6.5 Component functionality issues . . . 72
6.6 Group member unresponsiveness . . . 73
6.7 Communication breakdown with I-CONIC . . . 73
6.8 Team member disputes/disappearance . . . 73
6.8.1 Risk encounter . . . 73
6.9 Finalization risks . . . 74

References **75**

A Test case images **77**
A.1 Polygon tests . . . 77
A.2 Line tests . . . 79
A.3 Point tests . . . 82
A.4 Other tests . . . 84

2 Glossary

Word	Definition
Drone	A remote controlled flying robot, often mounted with a camera.
Aerial imagery	Images taken from the air, in this document specifically by a drone.
3D-model	In this document, aerial imagery that has been given associated height values for each pixel through I-CONIC:s software.
4D-models	A model consisting of multiple 3D-models that span across multiple points in time. A 3D-video.
Backend-systems	Software not presented to users.
CV	Computer vision.
GitHub repository	GitHub hosts Git repositories online to facilitate collaboration and ease of development. A Git repository is a version controlled folder.
GUI	Graphical user interface.
Icon	A small image that represents an application or a function.
OOP	Object-oriented programming.
Photogrammetry	The science of creating 3D-models from images.
UX	User experience.
Shape	In this document, a collection of points that represent data the user wants to know.
Measurement	In this document, specific data such as length and area of a shape.
Line	A measurement shape consisting of straight line-segments between points, used for measuring length.
Polygon	A measurement shape consisting of a polygon demarcated by points, used for measuring area and perimeter length.
Point	A measurement type consisting of a single point used for measuring heights.

Table 1: Glossary

3 Introduction

This introductory section aims to give an overview of the project and what it is about. It will also explain who we are and why we are doing this project. We will do this by presenting background information on our client, similar products, and the project. Additionally, we will be evaluating our team members and the scope of the project Thus far we have done research on similar projects and gotten a good idea of how the finished project may look. We have also made substantial progress in the project with only certain features remaining to be implemented.

3.1 Project summary

When disasters like earthquakes or floods happen it is important to quickly get an overview of the situation so that further risks can be assessed and first responders can get to work. The company I-CONIC has developed a technology that can efficiently create 3D-video models (see Figure 2) of an area using drone footage, something which can give vital information when surveying a disaster site. However, only seeing a 3D-model can be difficult for a person to parse. The end user therefore requires a tool to create measurements within the 3D-model. We have been tasked with creating a measuring system that would give users of I-CONIC's technology the capability to measure distances, heights, areas, and volumes in the models through a simple 2D-image interface (see Figure 13a for reference). This could possibly be used to measure the heights of destroyed buildings or the width of possible access roads. A simple architecture diagram can be found in Figure 1 that uses the modules defined in §4.1.

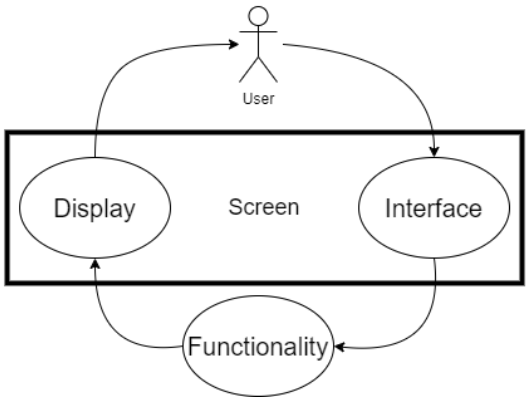
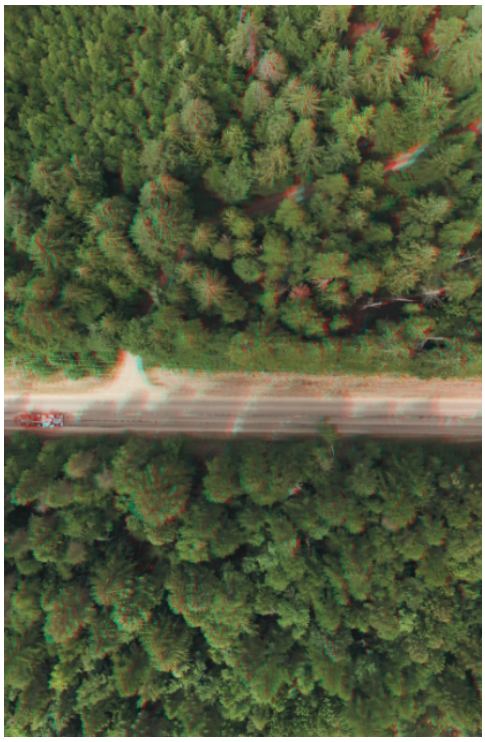
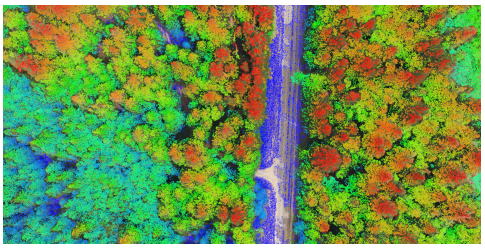


Figure 1: A simple system architecture

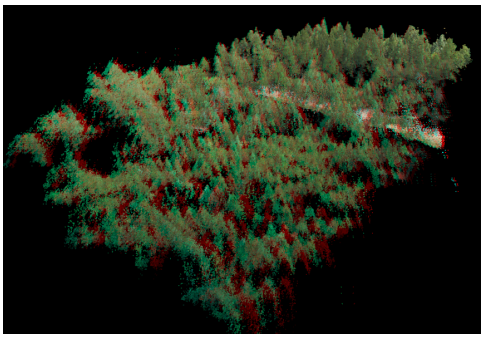
This will be accomplished by extending a project that I-CONIC has supplied us with that already implements rendering and basic user input. Coding will be done inside of C++ classes created by I-CONIC and will therefore be modular in the sense that the code can be transferred to other I-CONIC projects. So far the GUI has been extended so that all the necessary shapes can be created, selected, and modified by adding and moving their points. Additionally, the measurement data can be viewed both in the toolbar for the selected shape and in a side panel where all data is displayed. Lastly, the ability to delete shapes has been implemented. Going forward some more complex features such as volume calculations and height profile rendering remain to be done.



(a) Overhead stereoscopic image blending two video frames to give a 3D effect.



(b) Height map where colors indicate height value.



(c) 3D-model made of particles.

Figure 2: 3D-imagery generated by I-CONIC.

3.2 Purpose

The purpose of this document is to provide a comprehensive and coherent plan for the development and implementation of a measuring software for I-CONIC. The purpose of this report is to define multiple things: business requirements to understand what the need of the project is and what deems it a success as well as functional requirements to understand what needs to be implemented for the program to work properly. The requirements will take into consideration what the most prioritized requirements are to aid in making a viable working product within the time limit. By thoroughly defining the requirements for the project we aim to reduce any risks of uncertainty or confusion both in the project group as well as in the communication with I-CONIC. A lack of clear requirements would make it more difficult to progress as it would not be clear what the next steps in the project includes and what should be implemented or worked upon. The report will also explain certain use cases to exemplify the functional and business requirements.

This report also aims to explain the methodology we have chosen for this project, as well as explain why we have chosen this method for the project. The document will also explain how we have divided the work amongst ourselves to work efficiently as a group. It is important to have a clearly defined methodology in a bigger project to keep the planning and division of work organized as this greatly reduces unnecessary and time consuming discussions within the group to organize what should be clear from the beginning.

3.3 Company background

I-CONIC Vision AB, or I-CONIC for short, is a self-proclaimed "Swedish start-up with the ambition to revolutionize how video is processed and how video can be perceived and utilized." The company is founded by multiple Ph.D. recipients in image processing fields and have received multiple grants for their groundbreaking work (I-CONIC Vision AB, 2023).

I-CONIC prides themselves on their use of drones and software to create 3D and 4D models of real-life terrain. As such they have already developed a sophisticated backend-system for rendering 3D models with drone imagery (see Figure 3). Currently, I-CONIC is working on a United Nations supported project whose goal is to use drones to get quick aerial overviews of regions struck by disasters such as earthquakes or floods (I-CONIC Vision AB, 2022a).



Figure 3: I-CONIC pipeline for generating 3D-models.

The company has also worked with KTH students before on three different occasions in a similar context. Thanks to this, they were prepared with a set of tasks for us to implement prior to our first meeting. I-CONIC is well aware of the capabilities and limitations of us as computer science students; this knowledge is and has been helpful when negotiating realistic goals for the project at hand.

3.4 Market analysis

A market analysis is used to identify customers, competitors and more about the market such as the supply and demand of a service. There are a couple of similar services available that perform tasks similar to what this project hopes to do. For example: Google Earth and Lantmäteriet (only in Sweden) both allow users to make 3D measurements. Although, at the time of writing there is no availability in Lantmäteriet's measurement tool to make measurements on buildings, it is strictly bound to measurements of the ground. On the other hand, the desktop version of Google Earth, called Google Earth Pro, does have options to measure buildings and areas. It does lack some measurements especially regarding volumes but does contain some features expected to be implemented in this project such as height profiles and generic polygon area measurements.



Figure 4: Measurement Google Earth

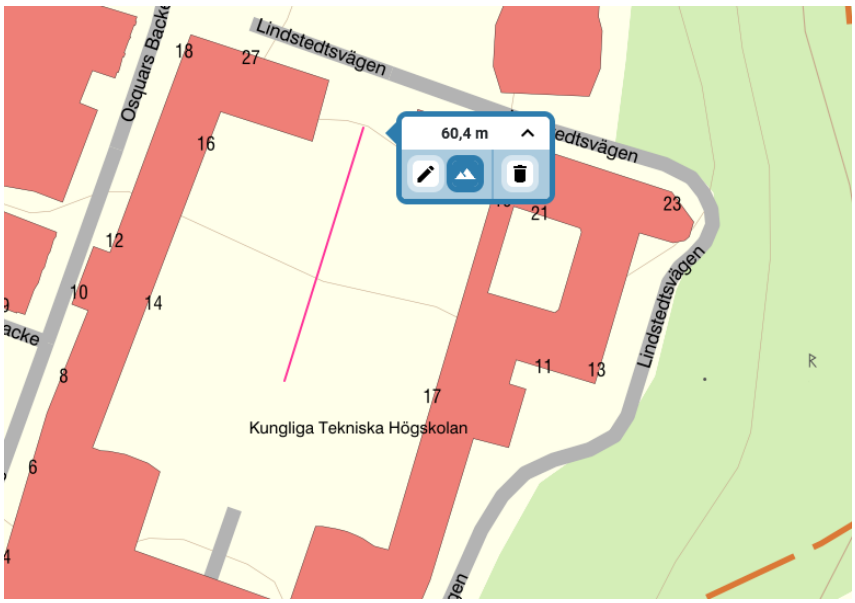


Figure 5: Measurement Lantmäteriet

The company has also provided the previous KTH-project (see Figure 6) which utilizes the wxWidgets library and encouraged us to familiarize with it in order to understand the usage of it in context. This project did not include precise 3D measurements but was meant to give a larger view of the area captured by the drones in context of a bigger map of the surroundings.

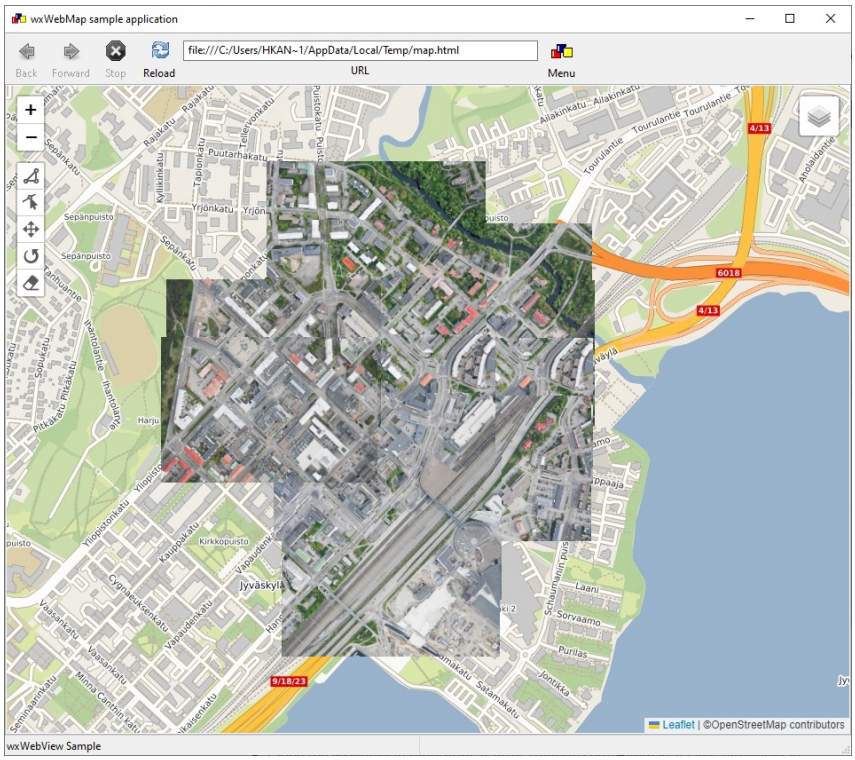


Figure 6: Previous KTH Project

There are other companies that offer similar drone services like I-CONIC does. For example Inkonova offers 3D mapping using drones through laser mapping, though they specialize in underground environments (Inkonova, 2023). There is also Spotscale who uses drone footage to make 3D models of buildings, but they appear to be focusing on aiding in insulating buildings and similar tasks (Spotscale, 2023). It does also appear this process takes longer than the solution I-CONIC has developed.

There is also a company called Pix4D whose product is very similar to that of I-CONIC. They develop 3D models from drone footage as well as other forms of footage such as mobile cameras. The software used to make measurements appear to include most if not all the features specified for this project. Pix4D was also engaged in disaster response in the explosion in the 2020 Beirut explosion (Karpowicz, 2022).

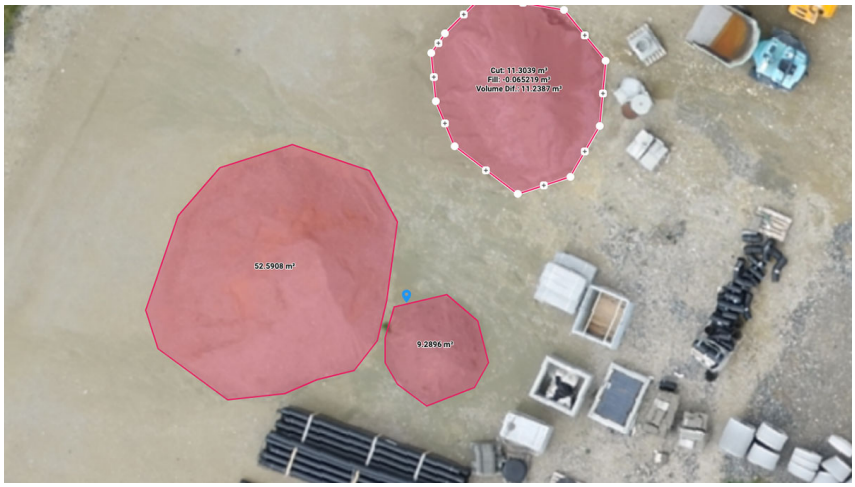


Figure 7: Pix4D UI measuring volume of a pile of sand



Figure 8: Pix4D 3D model of arena

According to I-CONIC, one of their main selling points is the speed at which they can create the models and use them in comparison to their competitors. Google Earth is based on a substantial amount of data, such as satellite imagery, and will therefore be considerably slower than I-CONIC's drone based solution. Pix4D on the other hand can use footage more easily obtainable, though does not appear to focus on speed to as far of an extent as I-CONIC. Therefore there appears to be some gap in the market which I-CONIC could fill.

Since these companies and I-CONIC have very similar goal products this project is likely to incorporate some of the features in a similar manner. For example a height profile of lines are incorporated well by both Lantmäteriet where it is easily available

and well designed and Google Earth where the data is easily read, see Figure 9 along with Figure 10.

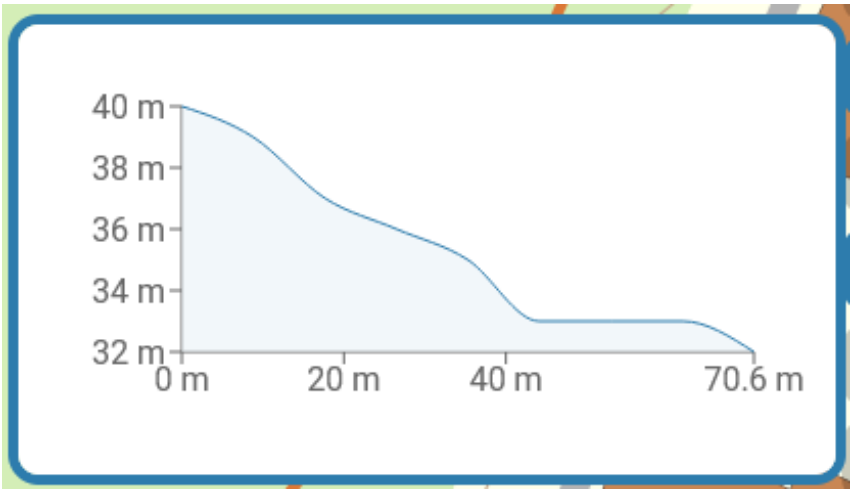


Figure 9: Height profile Lantmäteriet



Figure 10: Height profile Google Earth

3.5 User group

I-CONIC has a product that is meant to be used in high intensive environments and by first responders who do not have time to look at a manual. As noted previously they are currently conducting a project for use in disaster zones, and they have previously done projects in relation to satellites and homeland defence (I-CONIC Vision AB, 2022b). Therefore, it is fundamental that the product is reliable — i.e. not prone to crashing — and easy to use.

3.6 Project overview

This project began in December 2022 and will continue until around May 2023, further details are in §5. The methodology used will be Scrum using 3–4 week sprints, further details are in §5.1.1.

3.6.1 Intended functionality of finished program

The program should have two general modes of interaction: the move-mode and the measure-mode. In the move-mode, the user can use their mouse to click and drag which moves the loaded image around as well as use the scroll-wheel to zoom in and out — we refer to the free cartography software Google Maps for an example of this functionality. It is the goal of the project to implement the measure-mode which is entered by either creating a new shape or selecting an already existing shape. In the measure-mode, the program has a specific shape "selected" which allows the user to add points to it, move points of the shape, delete points of the shape, or move the entire shape.

Regardless of which mode the program is in, the measurements (i.e. coordinates, lengths, areas, and/or volumes) of the shapes should be viewable.

Additionally, when in the move-mode the user should have the option to read shapes from or write shapes to a file.

3.6.2 Requested features

I-CONIC has identified that their end-users need more than just the 3D models of a given environment, they have also discovered that a tool to make measurements using the data from the drones would be beneficial. An example they mention is the ability to find accessible routes at disaster sites by using the tool to measure the size of different gaps and obstacles.

The solution put forth by I-CONIC was to create a software application that uses the data gathered by the drones to let the users create different measurements.

Therefore, the project aims to create a system that allows users to take measurements in the height-maps that I-CONIC creates. Specifically, I-CONIC has requested the user to be able to place points on the map that allows them to see the height coordinate of a pixel; and to be able to draw lines by placing multiple points to calculate lengths and also display a height profile (for reference see Figure 9); and also to be able to draw polygons by placing multiple points to calculate areas and encapsulated volumes. These shapes should be modifiable and able to be deleted once created.

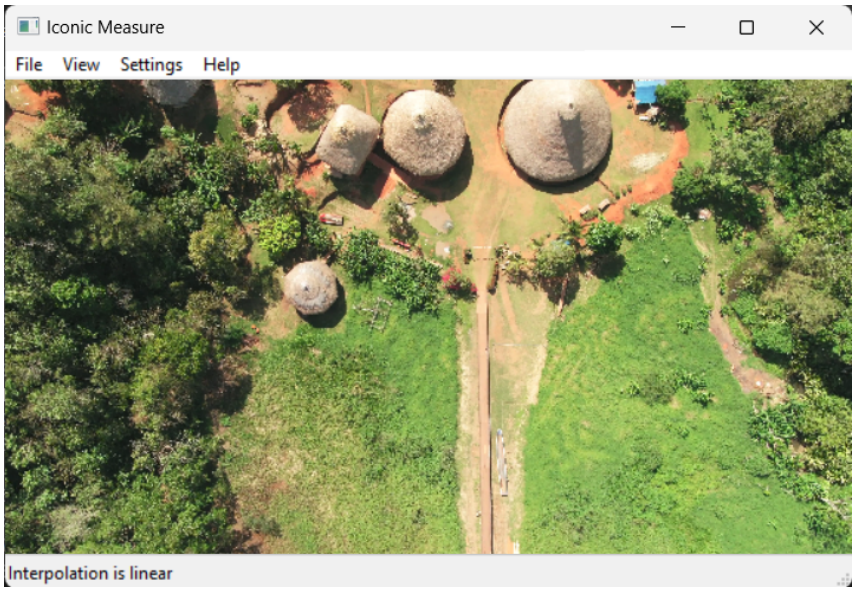


Figure 11: The base program created by I-CONIC

Additionally, I-CONIC wants the measurements to be save-able meaning that they should be able to be read from and written to files, specifically of the WKT file format.

I-CONIC has provided a base program (see Figure 11) to build upon that already implements the aforementioned move-mode, in addition to their API that handles certain computations. This is not a "final product" *or* prototype that we are building on — instead, our goal is to extend their rendering classes so that they can handle measurements.

3.6.3 Program class overview

The system that this project aims to create can generally be considered to be very simple. There is no database and no network communication; it is a standalone piece of software. In Figure 12 a class diagram of the system is presented, an explanation of the classes can be found in Table 2. Note that there is a component denoted GUI that contains multiple different classes. These are grouped since they together fill the role of communicating with the user.

Certain parts of this program were created by I-CONIC. This includes large parts of the GUI (excluding the SidePanel), and large aspects of the MeasureHandler and Geometry classes. The coding in the project focused on creating the Shape class and its derived classes, extending the MeasureHandler, and ensuring interoperability and loose coupling between the components of the program.

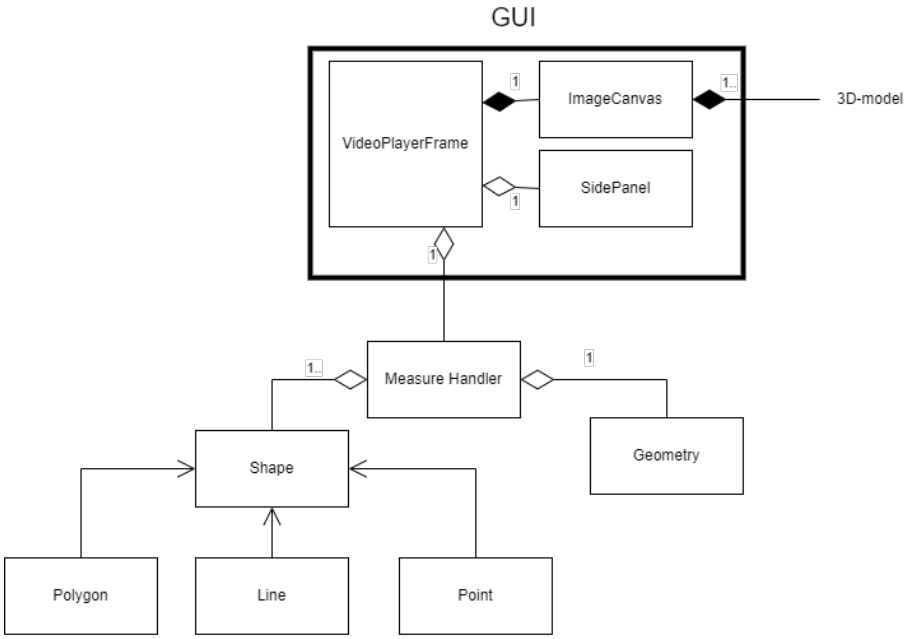


Figure 12: Class diagram describing the program

Class name	**Description**
VideoPlayerFrame	The class that defines the layout of the GUI. Extends the wxWidgets class wxFrame. Predefined by I-CONIC with additions made in the context of the project.
ImageCanvas	The class that handles interaction with and drawings on the loaded image/3D-model. Extends the wxWidgets class wxGLCanvas. Predefined by I-CONIC with additions made in the context of the project.
SidePanel	A class that allows for displaying measurement data created by the program. Created in the context of the project.
MeasureHandler	The class that handles creation and modification of the shapes in the program. It also aids with drawing the shapes and displaying their measurements. Predefined by I-CONIC but heavily extended in the project.
Geometry	The class that handles conversions between different coordinate systems. Predefined by I-CONIC.
Shape	The base class that defines what a shape is in the program and what they can do. Extended into the Polygon, Line and Point classes. Created in the context of the project.

Table 2: The code-classes of the program

Dependency	Description
Visual Studio	A development environment to build and structure C++ programs.
wxWidgets	The graphical library used for window management.
Boost	A collection of libraries used primarily for pointer-management and geometrical calculations.
OpenGL	Bindings used for rendering lines and dots that represent the shapes of the measurements.
Doxygen	A documentation tool generated from correctly formatted C++ comments.
I-CONIC:s API	An extensive tool for a wide variety of photogrammetry use cases. Will be used minimally in the project.

Table 3: Summary of software dependencies enforced by the client

3.6.4 Dependencies of the project

The base program that I-CONIC has provided us with (which can be seen in Figure 11) has certain software requirements. The most obvious one being wxWidgets that handles the graphical window of the program. To draw graphics in the wxWidgets window the code also uses OpenGL bindings — this creates a hardware requirement for the users of the program since their graphics cards need to be compatible. Additionally, multiple third party C++ libraries are used from the boost-collection, most notably boost-geometry for calculations.

I-CONIC has provided a system — the third party software Doxygen which works well with the language C++ that the program is to be coded in — for code documentation along with clear instruction on how to use it. They have also provided a GitHub repository where the base program that will be extended exists that already implements fundamental features such as rendering images in a window and a basic menu bar. Access to I-CONIC:s API has also been provided which means that we can use their state of the art photogrammetry functionality. Additionally, testdata — an image with its associated heightmap — is supplied. These aspects allow development to focus on the core aspects of the project.

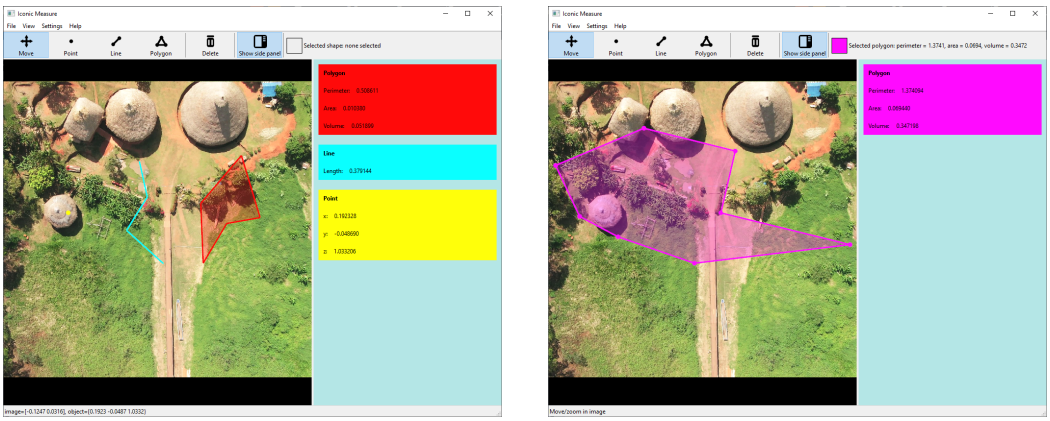
As noted in the previous section the program requires OpenGL compatibility. Our experience with installing and running the code tells us that only NVidia graphics cards with OpenGL functionality actually work.

The program also only works on Windows systems, although Linux interoperability is considered by I-CONIC.

3.6.5 Current state of the project

A detailed description on the state of the project can be found in §4.3, but a general overview will be given here.

In the start of the project we were given a base project that we were told to extend (see Figure 11). Note that the aerial image in the program is supplied test data loaded from a file. This project already had the move-mode implemented and as such we could focus immediately on the measure-mode functionality. So far, we have implemented the functionality to create different shapes by selecting a shape to create through a toolbar and by placing points on the image (see Figure 13a or 13b), as well as the capability to render said shapes. Recently, the functionality to display measurement data from the shapes has been added in addition to creating tools allowing for the editing of existing shapes. These tools include deleting entire shapes, adding points to shapes in arbitrary locations, as well as moving existing points in shapes. Remaining features include certain measurement calculations and presentations, the capability to delete points in shapes, and being able to save shapes to file.



(a) A display of every measurement type.

(b) A selected polygon.

Figure 13: Images of the current state of the program

3.7 User-friendly Interface

I-CONIC requested that the application has a user-friendly interface. This section will describe both generally what a user-friendly interface means and what it means for this specific project.

3.7.1 Good Design

The goal is an interactive product that is usable (Sharp, Rogers & Preece, 2019). To accomplish this it is necessary that simple operations are easy to access. That an operation is easy to access means that it is on the menu bar or one menu item deep. Examples of some common simple operations are save, delete and search. It is also important that frequently used operations are easy to access.

For our project this means that all the basic measuring tools should be easy to access when the program is in measuring mode.

Another design aspect is the icons (Sharp et al., 2019). Icons need to be recognisable for the user. They can do this by representing physical objects that either do the same thing (a floppy disc representing save), being an analogy (a scissor representing cut) or using similar icons to other programs. In our project we will be using icons that are similar to measuring and drawing tools from other programs.

It is also important to use menus (Sharp et al., 2019). A program with 100 different icons all on the screen where 70 of them are rarely used is just cluttering the user's screen. The different functions need to be divided into menus to make a good user interface. For our project this means that functions that are rarely used should be in a menu.

3.7.2 Bad Design

An interactive product can be difficult to use due to many different reasons (Sharp et al., 2019). One of the main things that makes a program difficult to use is a lack of or incomplete instructions. An example of this would be a program where all the icons use the same image and there is no text next to the icon nor when you hover the mouse over the icon. A program is also difficult to use if it is difficult to do simple tasks. An example of this would be a word processor where you have to go through five different layers of menus every time you want to make a new sentence. For our project this means that icons for different functions should be distinct and have a tooltip that shows when the mouse hovers over it.

3.7.3 Accessibility

Accessibility has two different perspectives to consider (Sharp et al., 2019). Accessibility is most often associated with creating solutions for disabled people, but accessibility is also important for abled people. There are four main areas to consider for accessibility, these are vision, sound, motor functions and cognitive. All of

these areas are spectrums, for example vision can be anywhere from having some difficulty with separating red from green to not being able to see at all. For our project this means that we should use existing functions to make screen objects instead of manually drawing squares on the screen because the existing functions have inbuilt support for accessibility tools such as screen readers.

3.8 Scope of the project

Due to the circumstances under which this project takes place — of which I-CONIC is well aware of — there are considerable limitations in scope and expectations. Primarily, the developers in the project are only second year computer science students and as such are far from experts in the field of software development. Additionally, the factor of time plays a major part in the limitations of this project given that it is part of a course that 1. Has a set end date, 2. Has additional work on the side such as presentations and 3. There are additional courses aside from this one that also occupies the developers' time.

Based on these factors the scope of the project has — in collaboration with I-CONIC — been limited to what is believed to be a reasonable size. The primary focus is on creating and displaying measurement data from shapes whose lines are straight. In early meetings with I-CONIC further features were discussed such as circular shapes and implementing a stereoscopic mouse sprite that would — given the user is wearing compatible 3D-glasses — make the mouse look like it is hovering at different distances from the screen depending on the height-map data at the current mouse position. These ideas are not prioritized and will be considered only if sufficient time is left after all other features are fully implemented.

3.8.1 Limiting the scope

Towards the end of the project it was realized by the project group that it would be difficult and highly stressful to try to complete the entire project as envisioned at the beginning. The workload in other courses as well as other coursework in this course towards the later parts of the project happened to be more than expected. Because of this certain features were down prioritized. This includes rendering heightprofiles and calculating volumes.

3.9 Members and skillsets

This group is comprised of people with different areas of expertise and background. In the following table, a summary of the more relevant skillsets of each member are presented.

Name	**Relevant skills**
David Aldenbro	• UI • C# • OOP
Sofia Andersson	• C# • LaTeX • 3D modeling
Linus Dinesjö	• C • C++ • UX
Elias Floreteng	• UX • Version Control • Code Style
Pontus Lüthi	• C++ • OOP
~~Leonard Petersson Hoogendoorn~~	• ~~C#~~ • ~~C~~
Alexander Skalski	• C# • Visual Studio
Gustaf Sundmark	• C • C++ • wx • CV

Table 4: Relevant project skills for group members

3.10 License

This code supplied in the base program created by I-CONIC and used as a starting point for the project is the property of I-CONIC. Additional code written by the project group in the context of the project in regards to extending the supplied program is owned by the project group who give I-CONIC full rights to use and modify the code without disclosing source. Any and all people who requests any code in this project whether written by I-CONIC or the project group for usage and/or inspection should contact I-CONIC. The project group is not liable for any problems the software might cause.

4 Requirements

This section will define the project requirements. It will also define how we will do our testing. We will define the requirements using the stakeholders request we got from I-CONIC. We will also hold meetings with I-CONIC to ask for clarifications and make or receive additional requests. We have defined the business, functional and non-functional requirements. We have also defined different use cases. Finally we have created test cases and linked these to the associated requirements through a traceability matrix which can be found in §4.7

4.1 Project modules

There are multiple facets that have been identified regarding this project. To distinguish and departmentalize the project the following modules have been crafted:

Module	Description
Interface	A big part of the project will have to do with user interaction. This means accepting input from the user and validating it.
Functionality	The core functionality that the client desires is encapsulated in this module, namely the system for calculating measurements. Additionally, it includes handling the measurements and geometries such as reading and writing them from and to files.
Display	When the user has made a request of data to the system that data should be presented. A big part of the project will be to make sure that the data is presented in an effective manner.
External	The program that is coded in this project is not meant to be a final product. It is instead meant to be a development environment for us to extend their code that already displays images to also handle creating and displaying measurements. Therefore, the project needs to be coded in a modular manner where classes and files can be transferred to other programs.

4.2 Business requirements

I-CONIC is, fundamentally, a tech startup created by people with a specific solution in mind. Because of this, the requirements received from I-CONIC were more practical and detail oriented than general, business requirements. Nevertheless, in discussions with them it became quite clear what their general goals with this project are, which are noted in the following table together with their associated module.

Business requirements	Priority	Description
BR1, Interface	High	Create a user-friendly interface for taking measurements of real environments based on aerial photography.
BR2, Functionality	High	The user should be able to measure heights, distances, areas and volumes.
BR3, Display	Medium	The relevant system information such as measurement values should be presented in an easy to understand manner.

In terms of ranking the priority of the business requirements it should be noted that they are all quite interlinked in the sense that having two without the third would result in a quite useless program. However, we deem the aspect of clear and effective

presentation of shapes and measurements to be of slightly lesser importance and therefore give it only medium priority.

4.3 Functional requirements

In this section the functional requirements of the project are detailed. The functional requirements are characterized by being directly implementable as features of a program. They are grouped by their associated module and are color coded as specified in Table 6 and ranked as specified in Table 7.

Color	**Explanation**
Red	Problem encountered. In the case of a problem there will be a reference to a specific subsection of §5.7 for further reading.
White	Not fulfilled.
Green	Fulfilled.

Table 5: Functional requirements color coding explanation

Priority	**Description**	
	Priority is based on a number of criteria including pertinence to business requirements, the degree of expected added value, dependencies, and expected "cost" of implementation in relation to the previously listed factors.	
	Labels	
	High	The requirement is high priority to fulfill.
	Medium	The requirement is medium priority to fulfill.
	Low	The requirement is low priority to fulfill.
Complexity	**Description**	
	Complexity is a measurement of the developers' estimation of how difficult the requirement will be to implement and meet.	
	Labels	
	High	The requirement is highly complicated to fulfill.
	Medium	The requirement is modestly complicated to fulfill.
	Low	The requirement is not complicated to fulfill.
Progress	**Description**	
	Progress is a measure of how far in development a requirement is to being fulfilled.	
	Labels	
	Not implemented	There has not been any work done to begin implementing the requirement.
	Partially implemented	There has been some work done towards meeting the requirement but it is not close to completed.
	Basically implemented	There has been done work towards meeting the requirement but it is at most only halfway achieved.
	Functionally implemented	There most vital parts of the requirement have been met.
	Fully implemented	The requirement has been fully met.

Table 6: Functional requirements property explanation

4.3.1 Interface module

The requirements in this section tie to BR1.

FR1	User should be able to select shape type to create with toolbar.
Description	The user should have a clear way to select what type of measurement they want to perform: polygon, line, or point. This should be achievable through a toolbar with a button for each measurement type.
Sub-requirements	**FR1.1** Switching to a new shape type should deselect the current shape, and in the case of it being incomplete, it should be deleted. **FR1.2** Options for shape types should include Polygonshape, Lineshape and Pointshape. **FR1.3** Selection may only occur while image is loaded.
Priority	High
Complexity	Low
Progress	Functionally implemented

FR2	User should be able to create a new shape of specified type.
Description	Creating shapes to get specific measurements is the fundamental usage of the program. Having selected what shape to create the user can start clicking on the image to create points that make up a shape.
Sub-requirements	**FR2.1** Creating a new shape requires a shape type to be selected. **FR2.2** A newly created shape is automatically selected. **FR2.3** Initial click inside the image bounds should create a starting point for the shape. **FR2.4** Further clicks not in the exact same point should create further points for the shape. **FR2.5** Visuals should be updated throughout the shape creation process to indicate the shape's current look. **FR2.6** Completing the creation is done by right-clicking the mouse. **FR2.7** If the shape is incomplete when ending creation (one point for line, one or two points for polygon) the shape should be automatically deleted.
Priority	High
Complexity	Medium
Progress	Functionally implemented

FR3	User should be able to select a shape that has been previously created.
Description	By clicking on or inside of a shape that shape should become selected which will highlight it and allow the user to perform operations on it.
Sub-requirements	**FR3.1** Selecting a shape deselects the previously selected shape, if there were one. **FR3.2** Clicking inside of a polygon should select that polygon. **FR3.3** Clicking on a line should select that line. **FR3.4** Clicking on a point should select that point. **FR3.5** If multiple shapes could be selected from the same action, one of them should be chosen. **FR3.6** Selection may only occur in the Move mode defined by I-CONIC. **FR3.7** Deselection may occur with right-click, "Escape", or by clicking in a place without a shape.
Priority	High
Complexity	Medium
Progress	Functionally implemented

FR4	User should be able to modify selected shape.
Description	The client has requested the ability to add, remove and move points that make up a shape. Additionally, the entire shape should be movable as a unit.
Sub-requirements	**FR4.1** A selected polygon or line should be possible to add points to by dragging the middle node of a side. **FR4.2** Points of a shape should be moveable. **FR4.3** Points of a shape should be deletable. **FR4.4** The entire shape should be moveable as one unit.
Priority	Medium
Complexity	High
Progress	Basically implemented

FR5	User should be able to delete selected shape.
Description	The client has requested the ability to remove created shapes.
Sub-requirements	**FR5.1** Pressing "Delete" or a button on screen should delete the selected shape. **FR5.2** Before deletion occurs, a prompt should appear requesting confirmation. **FR5.3** The prompt should be easy to disable.
Priority	Medium
Complexity	Low
Progress	Partially implemented

FR6	User should be able to save created shapes to a file.
Description	The client has requested the ability to save shapes to the WKT file format.
Sub-requirements	**FR6.1** The user presses a button and saves all shapes to a WKT file. **FR6.2** If no shapes have been created the program creates a prompt to inform the user of this and does not try to save to file.
Priority	Low
Complexity	Medium
Progress	Not implemented

FR7	User should be able to load previously saved shapes from a file into the program.
Description	The files created according to FR6 should be able to be loaded and the shapes should function as if they had been created by the user.
Sub-requirements	**FR7.1** The user loads a WKT file and the shapes are loaded into the system.
Priority	Low
Complexity	Medium
Progress	Not implemented

4.3.2 Display module

The requirements in this section tie to BR3.

FR8	The program should display the created shapes.
Description	The shapes will be rendered for the user using OpenGL. Polygons and lines will be drawn with linesegments and points will be drawn as circles.
Sub-requirements	**FR8.1** The linesegments of polygons and lines should be drawn as straight lines. **FR8.2** Points should be drawn as circles. **FR8.3** Each shape should have only one associated color. **FR8.4** Shapes should not blend into each other and remain separate entities even while overlapping.
Priority	High
Complexity	Low
Progress	Fully implemented

FR9	The program should display measurement information.
Description	Displaying the heights, distances, areas and volumes that are measured are an important part of the program.
Sub-requirements	**FR9.1** The area of a polygon should be displayable. **FR9.2** The length of a polygon's perimeter should be displayable. **FR9.3** The volume of a polygon should be displayable. **FR9.4** The length of a line should be displayable. **FR9.5** The 3D profile of a line should be displayable. **FR9.6** The coordinates of a point should be displayable.
Priority	High
Complexity	Medium
Progress	Functionally implemented

FR10	The program should display lines between the mouse and the shape that is currently being created.
Description	This feature will give the user a way to see what their input will accomplish without them needing to perform it by drawing temporary lines.
Sub-requirements	**FR10.1** For line shapes, a temporary line should be drawn from the last created point to the mouse cursor while it is being created. **FR10.2** For polygon shapes, temporary lines should be drawn from the start point of the polygon and last created point of the polygon to the mouse cursor while it is being created.
Priority	Low
Complexity	Low
Progress	Fully implemented

FR11	The program should make the selected shape appear highlighted.
Description	To aid the user experience it is important that the selected shape stands out from the rest.
Sub-requirements	**FR11.1** The selected shape should be highlighted and uniquely distinguishable from other shapes. **FR11.2** The corner points of polygons and lines should be visible when the shape is highlighted. **FR11.3** The midway-points of each line segment should be visible when the shape is selected. **FR11.4** Highlight measurement data regarding selected shape.
Priority	Medium
Complexity	Medium
Progress	Functionally implemented

4.3.3 Functionality module

The requirements in this section tie to BR2.

FR12	User should be able to add points to shapes.
Description	Adding points to shapes lets the user customize the precise measurements desired.
Sub-requirements	**FR12.1** Add point to shape while creating it should work by simply clicking and new points appear as indicated by FR10. **FR12.2** Adding points to a finished and selected shape should work by adding points in the middle of linesegments and moving them to desired locations.
Priority	Medium
Complexity	Medium
Progress	Fully implemented

FR13	The program should calculate lengths of polygon perimeters.
Description	Using boost::geometry functions the perimeters of polygons should be calculated.
Sub-requirements	-
Priority	Medium
Complexity	Low
Progress	Fully implemented

FR14	The program should calculate lengths of lines.
Description	Using boost::geometry functions the lengths of lines should be calculated.
Sub-requirements	-
Priority	High
Complexity	Low
Progress	Fully implemented

FR15	The program should calculate areas of polygons.
Description	Using boost::geometry functions the areas of polygons should be calculated.
Sub-requirements	-
Priority	High
Complexity	Low
Progress	Fully implemented

FR16	The program should calculate volumes bounded by polygons.
Description	Using boost::geometry functionality together with I-CONIC expertise the volume bounded by a polygon should be calculated.
Sub-requirements	**FR16.1** The "ground-level" of the volume must be determined. **FR16.2** The bounded volume should be numerically approximated.
Priority	Medium
Complexity	High
Progress	Not implemented

FR17	The program should calculate 3D-profile of line.
Description	By sampling the height coordinate of the points along a line a height-profile can be presented showing how the height varies over a path.
Sub-requirements	**FR17.1** A specific window needs to be created to show this measurement. **FR17.2** The 3D-profile should be sampled with approximately 1 pixel step size.
Priority	Medium
Complexity	High
Progress	Not implemented

FR18	The program should triangulate polygons so the perimeter points are in an appropriate order.
Description	The program should not accept polygons that are jumbled or folded over themselves. Therefore an algorithm that reorders the points of a polygon is needed so that calculated areas and rendering are correct. I-CONIC will aid with this.
Sub-requirements	**FR19.1** Points in a polygon shape should be rearranged to make sure that rendering and calculations happen correctly.
Priority	Medium
Complexity	High
Progress	Fully implemented

4.4 Non-functional requirements

In this section the non-functional requirements of the project are detailed. The non-functional requirements are characterized by not being directly implementable as features of a program and being more general and design oriented in nature. They are grouped by their associated module. If a non-functional requirement or sub-requirement is currently achieved, this will be indicated via the color green.

4.4.1 Interface module

The requirements in this section tie to BR1.

NFR1	The program should be intuitive to interact with.
Description	It is important that the measuring system is simple to use and easy to understand since the intended use case scenario is in a stressful environment.
Sub-requirements	**NFR1.1** Recognizable and/or industry standard icons should be used where possible. **NFR1.2** Contradictory or non-intuitive aspects of the interface, as found in user testing, should be reworked. **NFR1.3** No documentation or instructions should be required to use the measurement system. **NFR1.4** Accomplishing common actions should take minimal amount of clicks.
Comment	This requirement requires user testing to fully verify. This has yet to be performed.

4.4.2 Display module

The requirements in this section tie to BR3.

NFR2	The program should display measurement data in an easy to read manner.
Description	It is important that the values calculated by the program are easy to read since the intended use case scenario is in a stressful environment.
Sub-requirements	**NFR2.1** Values should be written on a plain background with well contrasting foreground and background colors.
Comment	Care has been put on making sure that well contrasting colors are used.

4.4.3 Functionality module

The requirements in this section tie to BR2.

NFR3	The program should be appropriately efficient.
Description	Since the intended use case scenario is in a stressful environment it is important that the calculations and operations that the program performs are efficient.
Sub-requirements	**NFR3.1** Recurring computation times that are noticeable are to be minimized.
Comment	Stresstesting has not been performed, but for normal usage the program is appropriately efficient.

NFR4	The program should handle non-rational and "unexpected" user inputs.
Description	To minimize crashing and unexpected behavior the user input should be prepared for unintended usage.
Sub-requirements	**NFR4.1** To minimize inefficiency the program should not allow too many measurements at once. **NFR4.1** The program should not allow multiple points of the same shape having the exact same coordinates.
Comment	Certain user inputs give rise to unclear error screens. This requirement is not achieved.

NFR5	The program should be stable and not prone to crashing.
Description	Since the intended use case scenario is in a stressful environment it is important that the program is reliable and does not add stress.
Sub-requirements	-
Comment	All encountered crash scenarios have either been corrected or documented. However, proper reliability testing has not been done.

4.4.4 External module

NFR6	The project should be well-documented.
Description	I-CONIC has requested that the project is documented so that they can later implement the measuring system in other programs.
Sub-requirements	**NFR6.1** Relevant aspects of the program should be documented such that a person with no experience with the program's code, but sufficient knowledge in the language, libraries and other tools used, is able to understand and work on it. **NFR6.2** The documentation tool Doxygen should be used for documentation, though other methods may be employed as well. **NFR6.3** The code should generally be written in a self-explanatory manner such that it could be understood without comments.
Comment	Doxygen documentation have been used consistently and efficiently.

NFR7	The project should be modular and compatible with integration in larger software packages and similar programs.
Description	Since I-CONIC intends to use this code in other programs it is important to code it in an easy-to-implement manner.
Sub-requirements	**NFR7.1** The product may not only work in a standalone setting. **NFR7.2** The product should integrate, or be able to integrate with, other I-CONIC tools and software.
Comment	The code has been refactored to ensure loose coupling.

NFR8	The program should be executable on Windows system with OpenGL compatible graphics cards.
Description	So that I-CONIC can run and make use of the code created in this project it should not have further system dependencies than the base project that they gave us.
Sub-requirements	-
Comment	The program has the same requirements as at the beginning of the project.

4.5 Use cases

Below are listed a set of goals that users of the program may want to achieve with it. They are listed with the functional requirements (see §4.3) that must be met for the goal to be achievable and the business (see §4.2) and non-functional requirements (see §4.4) that need to have been met for this to be achieved in an acceptable manner. The row will be marked green if that goal is currently achievable.

Use case label	Use case description	Required FR	Associated BR	Associated NFR
UC1	Create a polygon to measure area.	FR1, FR2, FR8, FR9, FR10, FR11, FR12, FR15, FR18	BR1, BR2, BR3	NFR1, NFR2, NFR3, NFR4
UC2	Create a polygon to measure circumference.	FR1, FR2, FR8, FR9, FR10, FR11, FR12, FR13, FR18	BR1, BR2, BR3	NFR1, NFR2, NFR3, NFR4, NFR8
UC3	Create a polygon to measure volume.	FR1, FR2, FR8, FR9, FR10, FR11, FR12, FR16, FR18	BR1, BR2, BR3	NFR1, NFR2, NFR3, NFR4, NFR8
UC4	Create a line to measure distance.	FR1, FR2, FR8, FR9, FR10, FR11, FR12, FR14	BR1, BR2, BR3	NFR1, NFR2, NFR3, NFR4, NFR8
UC5	Create a line to view a height profile.	FR1, FR2, FR8, FR9, FR10, FR11, FR12, FR17	BR1, BR2, BR3	NFR1, NFR2, NFR3, NFR4, NFR8
UC6	Create a point to view its coordinates.	FR1, FR2, FR8, FR9, FR11	BR1, BR2, BR3	NFR1, NFR2, NFR3, NFR4, NFR8

Use case label	Use case description	Required FR	Associated BR	Associated NFR
UC7	Select a shape.	FR3, FR8, FR11	BR1, BR3	NFR1, NFR3, NFR4, NFR5, NFR8
UC8	Deselect a selected shape.	FR3	BR1, BR3	NFR1, NFR4, NFR8
UC9	Move a created polygon.	FR3, FR4, FR8, FR11, FR16	BR1, BR2, BR3	NFR1, NFR3, NFR4, NFR5, NFR8
UC10	Move a created line.	FR3, FR4, FR8, FR11, FR14, FR17	BR1, BR2, BR3	NFR1, NFR3, NFR4, NFR5, NFR8
UC11	Add point to a created polygon.	FR3, FR4, FR8, FR11, FR12, FR13, FR15, FR16, FR18	BR1, BR2, BR3	NFR1, NFR3, NFR4, NFR8
UC12	Add point to a created line.	FR3, FR4, FR8, FR11, FR12, FR14, FR17	BR1, BR2, BR3	NFR1, NFR3, NFR4, NFR8
UC13	Move a point in a created polygon.	FR3, FR4, FR8, FR11, FR13, FR15, FR16. FR18	BR1, BR2, BR3	NFR1, NFR3, NFR4, NFR5, NFR8
UC14	Move a point in a created line.	FR3, FR4, FR8, FR11, FR16, FR17	BR1, BR2, BR3	NFR1, NFR3, NFR4, NFR5, NFR8
UC15	Delete a point in a created shape.	FR3, FR4, FR8, FR11	BR1, BR2, BR3	NFR1, NFR3, NFR4, NFR5, NFR8
UC16	Delete a shape.	FR3, FR5, FR8, FR11	BR1, BR3	NFR1, NFR4, NFR5, NFR8
UC17	View updated measurements after shape has been modified.	FR8, FR9	BR3	NFR2, NFR3, NFR8

Use case label	Use case description	Required FR	Associated BR	Associated NFR
UC18	Load correct shape data from file and view their measurement data.	FR7, FR8, FR9, FR13, FR14, FR15, FR16, FR17	BR1, BR2, BR3	NFR1, NFR2, NFR3, NFR5, NFR8
UC19	Load incorrect shape data from file.	FR7	BR1	NFR1, NFR4, NFR5, NFR8
UC20	Store shape data to file.	FR6	BR1, BR2	NFR1, NFR3, NFR4, NFR5, NFR8
UC21	Take the code from this project and use it in another program.	-	BR1, BR2, BR3	NFR6, NFR7, NFR8

Table 7: Use cases

4.6 Testing and verification

To ensure correct functionality of the program the test cases defined in §4.6.1 will be used to guide development and testing. Testing will be done by trying to achieve the goals detailed and correct functionality will be determined by making sure that already achieved goals stay achievable.

4.6.1 Test cases

Below follow test cases that we have identified to be fundamental to ensure correct functionality of the program. Each of them have a State associated with them that is also represented by the color of the test case as described in the following table:

Color	Explanation
Red	Faulty (thought to be fulfilled but has in testing proven to not be)
Yellow	Not prioritized (due to time constraints)
White	Not fulfilled
Green	Fulfilled

Table 8: Test case color coding explanation

TC1	Create a triangle and display its measurements
Steps	Select polygonshape, click in three different positions on the image, deselect shape.
Expected result	A triangle should appear, and its area, perimeterlength, and enclosed volume should be displayed.
State	Faulty
Comment	The displayed values are incorrect. More specifically the volume calculation is incorrect. Example image in §A.1.

TC2	Create a line of three points and display its measurements
Steps	Select vectortrainshape, click in three different positions on the image, deselect shape.
Expected result	A line should appear, and its length and 3D-profile should be displayed.
State	Not fulfilled
Comment	The 3D-profile is not displayed. Example image in §A.2.

TC3	Create a point
Steps	Select pointsshape, click once on the image.
Expected result	A point should appear, and its coordinates should be displayed.
State	Fulfilled
Comment	Example image in §A.3.

TC4	Select a polygon
Steps	Select move mode, click inside of an already created polygon.
Expected result	The polygon should look selected and function as such.
State	Fulfilled
Comment	Example image in §A.1.

TC5	Select a line
Steps	Select move mode, click on an alreadty created line.
Expected result	The line should look selected and function as such.
State	Fulfilled
Comment	Example image in §A.2.

TC6	Select a point
Steps	Select move mode, click on an already created point.
Expected result	The point should look selected and function as such.
State	Fulfilled
Comment	Example image in §A.3.

TC7	Deselect a shape
Steps	Have a shape selected, press RMB or the Escape key.
Expected result	The currently selected shape should be deselected.
State	Not fulfilled
Comment	Pressing the Escape key does nothing.

TC8	Move a polygon
Steps	Select a polygon, click inside of it and then drag it with LMB.
Expected result	The polygon should keep its shape, be moved to the new location, remain selected, and its data should be recalculated.
State	Not prioritized
Comment	-

TC9	Move a line
Steps	Select a line, click on it and then drag it with LMB.
Expected result	The line should keep its shape, be moved to the new location, remain selected, and its data should be recalculated.
State	Not prioritized
Comment	-

TC10	Move a point
Steps	Select an already created point, click on it and then drag it with LMB.
Expected result	The point should be moved to the new location, its measurement data should be updated, and it should remain selected.
State	Fulfilled
Comment	Example image in §A.3.

TC11	Add a point to a polygon
Steps	Select an already created polygon, click in the middle of a linesegment and drag the new point to desired location with LMB.
Expected result	The polygon should have a new point dividing the chosen linesegment in two with all other points unchanged, its data recalculated and the polygon should remain selected.
State	Fulfilled
Comment	Adding points is not done exactly as specified but the intended result is achievable. Example image in §A.1.

TC12	Add a point to a line
Steps	Select an already created line, click in the middle of a linesegment and drag the new point to the desired location with LMB.
Expected result	The line should have a new point dividing the chosen linesegment in two with all other points unchanged, its data recalculated and the line should remain selected.
State	Fulfilled
Comment	Adding points is not done exactly as specified but the intended result is achievable. Example image in §A.2.

TC13	Move point of polygon
Steps	Select an already created polygon, click on one of its points and then drag it with LMB.
Expected result	The point should be moved to the new location, the polygon's data should be recalculated and the polygon should remain selected.
State	Fulfilled
Comment	Example image in §A.1.

TC14	Move point of line
Steps	Select an already created line, click on point and then drag it with LMB.
Expected result	The point should be moved to the new location, the line's data should be recalculated and the line should remain selected.
State	Fulfilled
Comment	Example image in §A.2.

TC15	Delete point of polygon
Steps	-
Expected result	-
State	Not prioritized
Comment	Steps for this functionality have not been prototyped.

TC16	Delete point of line
Steps	-
Expected result	-
State	Not prioritized
Comment	Steps for this functionality have not been prototyped.

TC17	Delete a polygon
Steps	Select an already created polygon, then press the delete button in the toolbar or the Delete key.
Expected result	The polygon should be removed from screen, its data should be deleted from the system and it should be deselected.
State	Fulfilled
Comment	Example image in §A.4.

TC18	Delete a line
Steps	Select an already created line, then press the delete button in the toolbar or the Delete key.
Expected result	The line should be removed from screen, its data should be deleted from the system and it should be deselected.
State	Fulfilled
Comment	Example image in §A.4.

TC19	Delete a point
Steps	Select an already created point, then press the delete button in the toolbar or the Delete key.
Expected result	The point should be removed from screen, its data should be deleted from the system and it should be deselected.
State	Fulfilled
Comment	Example image in §A.4.

TC20	Reject a polygon of less than three points
Steps	Select polygonshape, click in only one or two different positions on the image, deselect shape.
Expected result	The polygon should not be created.
State	Fulfilled
Comment	Example image in §A.1.

TC21	Reject a line of only one point
Steps	Select lineshape, click in only one position on the image, deselect shape.
Expected result	The line should not be created.
State	Fulfilled
Comment	Example image in §A.2.

TC22	Load correct shape data from file
Steps	Click button to load shape data, select file containing correct data.
Expected result	The shapes should be loaded in and be displayed.
State	Not prioritized
Comment	-

TC23	Reject incorrect shape data from file
Steps	Click button to load shape data, select file containing incorrect data.
Expected result	The program should display an error message regarding faulty data and the shape data should not be loaded.
State	Not prioritized
Comment	-

TC24	Store shape data to file
Steps	Click button to store shape data, select where to store it and what name the file should have.
Expected result	The program should suggest a fitting file name and once confirmed by the user, store the data to a file in the WKT file format.
State	Not fulfilled
Comment	-

TC25	Ignore line-selection when clicking inside imaginary polygon
Steps	Select move mode, click inside the polygon that would exist if the first and last points of an already created line connected.
Expected result	The line should not be selected and if something was selected it should be deselected (same as clicking another empty part of the image).
State	Fulfilled
Comment	This test case relates to an earlier version of the software where this was an issue. In the current implementation this is not an issue at all. Example image in §A.2.

TC26	Ensure polygon mouse-selection is precise
Steps	Select move mode, click empty space nearby an already created polygon in the image.
Expected result	The polygon should not be selected and if something was selected it should be deselected (same as clicking another empty part of the image).
State	Fulfilled
Comment	This test case relates to an earlier version of the software where inconsistent polygon mouse-selections where an issue. More precisely the polygon selection-area extended upwards, allowing the polygon to be selected by clicking above its visible representation. Example image in §A.1.

TC27	Uninitiated user test
Steps	Acquire suitable test person and ask them to perform tasks in order to verify user friendly design.
Expected result	Hopefully the test indicates that the program is user friendly. Otherwise, appropriate steps should be taken to ensure user friendliness.
State	Not fulfilled
Comment	User testing not performed.

TC28	I-CONIC approval for integration into codebase
Steps	Present code to I-CONIC and allow them to inspect it.
Expected result	Given the close communication that has been kept with I-CONIC the integration should go without major issues.
State	Not fulfilled
Comment	Project not yet finished.

4.7 Traceability matrix

The traceability matrix can be found in Table 9. In the figure BR stands for business requirement (see §4.2), FR stands for functional requirement (see §4.3), NFR stands for non-functional requirement (see §4.4), and TC stands for test case (see §4.6.1). The structure of the matrix is based on the modules defined in §4.1. The meaning of the color coding is the same as for the test cases (see Table 8).

Module	BR	FR & NFR	TC
Interface	BR1	FR1	TC1 TC2 TC3
			TC20 TC21
		FR2	TC1 TC2 TC3
			TC11 TC12 TC20 TC21
		FR3	TC4 TC5 TC6
			TC7 TC25 TC26
		FR4	TC8 TC9 TC10
			TC11 TC12 TC13
			TC14 TC15 TC16
		FR5	TC17 TC18 TC19
		FR6	TC24
		FR7	TC22 TC23
		NFR1	TC27
Functionality	BR2	FR12	TC1 TC2
			TC11 TC12 TC13
		FR13	TC1 TC8
			TC11 TC12 TC13
		FR14	TC2 TC9
			TC12 TC14 TC16
		FR15	TC1 TC8
			TC11 TC13 TC15
		FR16	TC1 TC8
			TC11 TC13 TC15
		FR17	TC2 TC9
			TC12 TC14 TC16
		FR18	TC1 TC8
			TC11 TC13 TC15
		NFR3	TC27
		NFR4	TC27
		NFR5	TC27
Display	BR3	FR8	TC1 TC2 TC3
		FR9	TC1 TC2 TC3
			TC8 TC9 TC 10
			TC11 TC12 TC13
			TC14 TC15 TC16
			TC17 TC18 TC19
		FR10	TC1 TC2 TC4 TC5
			TC11 TC12 TC20 TC21
		FR11	TC4 TC5 TC6
		NFR2	TC27
External	-	NFR6	TC28
		NFR7	TC28
		NFR8	TC28

Table 9: Traceability matrix

5 Project Plan

This section will define the project plan. Seeing as this is a complex project that requires collaboration both within the group and between the group and external parties it is important to have a structure to work within. The project plan includes both the project goals and course deadlines. This section deals with detailing important dates and deadlines while also explaining the methodology and communication pathways employed in the project. We will also define the software engineering methods we will use, which, among others, include Scrum which I-CONIC has specifically requested. We will also define how we will communicate within the group and outside the group and documented important communication such as meetings and selected written communication. This section ends with comments and reflections on how the project is going and changes that have been made.

5.1 Software engineering methods

This section will explain what software engineering methods we are using for this project. It will also explain why we are using these methods and how we will be applying them. We will do this by researching different software engineering methods. We will also take into account methods we were recommended to use by both I-CONIC and the lectures.

5.1.1 Scrum

Thus far we have chosen to use the software engineering method Scrum. We have also done research on how to use Scrum.

Standard aspects of Scrum

Scrum in the traditional sense has three different roles: the Scrum Master, the Product Owner and the Scrum Team. The Scrum Team being a set of individuals tasked with completing a set of features in a set amount of time known as a sprint. There is also the Product Owner, which helps the team identify and prioritize different features (usually a key stakeholder in the project). Finally there is the Scrum Master whose purpose is to keep the team focused and productive, adhering to Scrum principles. Important parts of Scrum also include: a backlog of tasks, daily Scrum Team meetings, and a sprint retrospective meeting at the end of the sprint (International Scrum Institute, 2021).

Specifics regarding Scrum is that the sprints should be fixed and time-boxed, lasting for only a couple weeks. Another thing to note is that the daily meetings should be short, about 15 minutes (Meyer, 2014).

Reasoning for the use of the Scrum

We chose Scrum as the iterative method we would utilize while working on our project. This choice was largely due to our client requesting it, saying that they would be available for regularly scheduled meetings to review our work. This combined with the fact that we received a comprehensive list of project requirements from our client which works as a product backlog makes for perfect conditions to employ sprint cycles. As mentioned earlier in the report, we will have meetings with our client I-CONIC every three to four weeks, each sprint cycle will therefore also be of the same length.

Differences between our methodology and Scrum

However, we will not be using all parts of the traditional Scrum methods. For example, we will not be having daily 'stand-up' meetings. This due to the fact that we as a group do not have the time to do so, since we have other courses that also require our attention. Instead we will have weekly meetings of a longer nature (about an hour) to catch up with each other's work and work through eventual issues that has surfaced.

At the start of each sprint we will pick features from the backlog of tasks that shall be done during that sprint. These could be features straight from I-CONIC or some discovered to be necessary from the previous sprints. The focus for each sprint will be determined together with I-CONIC during the review-meeting of the previous sprint. This is in order for I-CONIC to have a say in what they prioritize the most during that particular sprint. With this approach we adhere to agile principles by enabling the requirements to be changed throughout the project thanks to incremental development and frequent client involvement.

A major part of Scrum that we have omitted has been the Scrum Master. This was in part due to our focus on roles, one of which where Scrum, but where the task of Scrum Master was not decidedly a part of the role, instead focusing on researching Scrum. This has lead to the team not being as productive or focused on the task, as no one have had the expressed focus of guiding the team forward. To increase our productiveness, one solution could therefore be to introduce a Scrum Master. (This has since been done, see §5.8).

Finally, the sprint retrospective has been omitted. This is mostly done to save time, as a reflection of the work that has been done can be done in other contexts, such as via text or in the weekly meetings.

5.1.2 Code review

Code review is a method where other developers review code written by a developer to identify potential bugs, performance issues, lack of consistency, or other potential improvements. The reviewer can also suggest changes regarding documentation, clarity and structure.

A code review can be done in any stage of the development of the software and is often done quite frequently. In our case we have implemented this through the use of GitHub. When we have finished a task during a sprint, we create a pull request with our work and assign two or three reviewers who are tasked with looking through the changes before accepting them, thereupon merging the changes with the main branch. Most often something requires fixing or has to be improved upon prior the the merge, which has been detected through the use of this method (GitLab, 2023).

The previously mentioned code review has mostly been done within our own group, but since this project is for a client who has quite specific needs regarding the software they have also been included in the code review process. Albeit not as frequently as within the group itself. This has been done close to, or at the end of each sprint, where the client has been allowed the opportunity to give feedback regarding the code or request change to the software regarding the new work that has been made during the sprint. Through this we have gotten feedback regarding not only functionality that they want, but also code conventions that they would like us to follow, but had not previously informed us of.

5.1.3 Pair programming

Pair programming is a method where two developers work concurrently on the same piece of code. One of the developers write the code while the other reviews the code in real-time. Pair programming of course also includes discussing in the pair to reach solutions. This method helps reduce the amount of bugs and errors earlier in the development reducing the risk of them reaching later stages. It also help code readability and best practice and the reviewer can help the one writing to be consistent and follow standards.

Not only does it help prevent errors and the like, but can also aid in knowledge deficiencies. We have taken advantage of this in some cases where one team member

is more proficient in some aspect that another member has to work around. This is good in multiple regards. Work during pair programming will be done both faster and better, also, we can spread the knowledge regarding the subject, allowing the latter member to work more on this later or further spread the knowledge in the group (GitLab, 2021).

Additionally, this has helped to ensure that everyone in the group has the opportunity to contribute to the project. Because of certain hardware dependencies of the project (see §3.6.4) some group members have had a hard time writing any code themselves. Through pair programming everyone has gotten a chance to come with ideas and feel like they helped.

5.1.4 Refactoring

Code refactoring is a software engineering method where the existing structure of the code, and code itself, is improved upon without adding anything new to the code base. The reason this method exists is to improve upon the software code base. This can be done to improve the code base in multiple aspects: readability, maintainability, possibility to build upon, and reduce time required when working on the code base.

This has been done in different scales a couple of times in the project. Since the group is quite new to bigger projects, refactoring has been required occasionally when it has become apparent that the planned structure of the code has been deviated from. It has also been required a couple of times as a result of the client's code review where they have not been satisfied with the structure of the code as it would make it more difficult to incorporate the work with their other projects (Wikipedia, 2023).

5.1.5 Prototyping

Prototyping in the context of software development is the process of developing the software as prototypes of parts of the final product. This method can be useful for allowing testing of features and parts of the product without requiring it to be fully built, such as in the case of the requirements not being finalized or fully known, or to allow independent testing of modules before their dependencies have been finalized.

This is achieved by the partial implementation of system modules, allowing the evaluation of these modules early in the development cycle, and therefore avoiding resources being unnecessarily devoted to work that is not pertinent to the final product, as envisioned or later conceived by the customer. Additionally, prototyping allows for

testing of modules early in development or design, again allowing resources to more effectively be dedicated to producing a satisfactory end product (GeeksForGeeks, 2023; Martin, 2023).

Prototyping is typically done in a number of steps. This initially starts with analysis and requirement gathering. In this phase, business requirements and potentially corresponding functional and non-functional requirements are formulated based on the needs of the customer (GeeksForGeeks, 2023).

In the next step, an initial design can be drawn up based on the results obtained in the first step. This design is then turned into a basic initial prototype. The development of an early prototype allows for initial testing and evaluation. This allows the customer to give feedback on the prototype, avoiding scenarios where significant resources are dedicated to building a product which is not actually what the customer desired, since potential misunderstandings or misalignment in requirements can be discovered early. Early user testing and evaluation can also be performed on the early prototype, which allows flaws to be discovered before significant resources have been invested into their development (GeeksForGeeks, 2023).

Based on the results from the testing and evaluation of the initial prototype, a more thorough prototype is then built, or the initial prototype is refined. This step and the previous steps can then be repeated until satisfaction (GeeksForGeeks, 2023).

When a satisfactory result has been reached, the focus can then be shifted to turning the (now refined) prototype into the final product and delivering it to the customer (GeeksForGeeks, 2023).

The exact nature of prototyping employed depends on the exact prototyping model used. For example, one model may employ the creation of multiple quick low-effort prototypes, with the intention of discarding most of them. On the contrary, another model may instead focus on developing one prototype and refining it until satisfaction. There are advantages and disadvantages to each of the various prototyping models. The development of multiple quick prototypes may offer more breadth, which prevents narrowing down on and refining a sub-optimal initial prototype. However, it also results in more resources arguably "wasted" on developing prototypes which get discarded (GeeksForGeeks, 2023; Martin, 2023).

5.1.6 Management methods and organizational structure

For the purposes of the project, we have employed management methods and an organizational structure best believed to fit the project requirements. In terms of

management methods, a flat organizational hierarchy has been employed within the group, meaning that all group members may be involved in decision making, and are largely responsible for the work that they are involved in. This method largely follows the method known as "Ömsesidig anpassning" (Mutual adjustment), where individuals take it upon themselves to complete work and cooperate as necessary to succeed at the set goals (Mintzberg, 1979).

This method is often highly efficient when possible, as it induces a sense of shared responsibility for the work, while also allowing for flexibility as necessary to overcome potential challenges and reach the set goals. However, one major drawback of this method is that it is not efficient in larger organizations or teams, and is best suited to small organizations/teams (Jacobsen & Thorsvik, 2019). For the group size of initially 8, and later 7 members, this has not posed any issues.

Another method that has been employed, in this case less by choice, is "Standardisering av kunskap" (standardization of knowledge). This is a method that suits itself well to knowledge-intensive work, where other methods such as direct oversight, standardization of tasks or standardization of results, may not be suitable. In these cases, typically where the workload involves novel challenges that require knowledge-intensive work, the standardization of knowledge may instead be more efficient. This is done by ensuring that all actors are sufficiently knowledgeable and able to handle the challenges at hand, and giving them sufficient freedom to do so, relying on their expertise to achieve results. For example, in a hospital, this may be achieved by hiring professionals in their fields who can be trusted to perform as required, and allowing them the necessary freedom to perform treatments as they see fit, based on their knowledge and experience in the field (Jacobsen & Thorsvik, 2019).

Since all group members have as a part of the computer science programme taken basic programming courses and some more advanced computer science courses, knowledge was naturally standardized in the group from its inception.

The method of "Horisontella samarbetsarrangemang" (horizontal collaboration arrangements) has also been partly employed throughout the project. This method is largely based on "ömsesidig anpassning" as described above, but more formalized. In knowledge-intensive work where varying expertise is required to overcome challenges, people with different knowledge may need to be brought in to collectively work on certain tasks in a semi-dynamic manner. This is often done in the form teams, which are made up of people selected for their knowledge and abilities, as relevant to the task(s) at hand. These teams can be both short-term and long-term,

depending on the requirements (Jacobsen & Thorsvik, 2019). In the context of this course, mini-teams have been formed for certain course and project objectives, such as specific development requirements, report writing/research, document formatting (LaTeX) and presentation preparation and creation.

5.1.7 UML diagrams

UML diagrams are a way to create visual models of a system, often used in software development. UML diagrams offer great expressibility of the relations between different components of the system, or even for more general aspects such as the organization itself. There are multiple different kinds of UML diagrams, including use case diagrams and class diagrams which have been used in this project (Nishadha, 2022; Lucidchart, n.d.).

5.2 Course deadlines

This project has four main sets of deadlines. There are reports to turn in — such as this one —, there are presentations to give, there is a seminar to give (as well as two to review), and there are Scrum deadlines imposed by I-CONIC. The first three of these are imposed by the context of this project being part of a course and, in the words of the course examiner, the "**deadlines are firm**" (Hussain, 2022), and will therefore be handled separately.

The strict project deadlines are of utmost importance since failing to meet them would result in not passing the course. They are presented in the following table:

Date	Deadline description	Comment
2023-01-23	Project Report 1	This deadline has passed and the report was turned in on time.
2023-02-20	Seminar Review 1	This date has passed and the group was present to review.
2023-03-20	Project Report 2	This deadline has passed and the report was turned in on time.
2023-03-23	Half-time Presentation	This deadline has passed and the presentation was held successfully.
2023-04-17	Seminar Presentation	This deadline has passed and the seminar was held.
2023-05-07	Project Report 3	Substantial work is currently being done regarding this deadline.
2023-05-08	Seminar Review 2	Preliminary work has been done regarding this deadline.
2023-05-15	Final Presentation	Preliminary work has been done regarding this deadline.

5.3 Project timeline

As detailed in §5.1.1 work will be conducted in an agile manner meaning that detailed planning has not been employed. However, it is worth noting the fundamental steps that are employed in this project.

After the initial meeting with I-CONIC we began the work to analyze what the project will be about. When this had been done the workload was divided so that it could be assigned both to individual people and also individual sprints. There are no clearly defined plans for what will be done in each sprint specifically, but the goal is for each sprint to end with a completed deliverable as described in §5.3.1.

5.3.1 Deliverables

Below are listed a set of prototypes that are viewed as deliverables in this project. The list of functional requirements that are associated with each prototype are cumulative, meaning that P2 also has the requirements of P1. The functional requirements can be found in §4.3.

Prototype label	Prototype description	Associated requirements	Delivered
P1	A basic prototype that has fundamental functions	FR1, FR2, FR3, FR8, FR9, FR10, FR11, FR12.1	Yes, 2023-02-23
P2	A MVP that allows for most requested measurements to be taken	FR5, FR9, FR13, FR14, FR15	Yes, 2023-04-02
P3	A more developed prototype that supports more functionality and allows for more user control	FR4, FR12.2, FR17	No, estimated early-middle May 2023
P4	A finished product that implements all requested functionality	FR6, FR7, FR16, FR18	No, estimated to not be completed

5.4 Communication

To ensure that work in the project progress smoothly and predictably it is pivotal to have good communication. Relevant communication can be split into two categories: communication within the group and communication with people outside of the group. Notes on how to handle lapses in communication are in §6.6 and §6.7.

5.4.1 Services overview

A variety of services are used for communication within the project. At first only a few channels of communication were employed, mainly the voice and text communication platform Discord, but this has since grown to sustain different needs of both the group, client and mentor. The currently employed communication services are presented in the following table:

Service	Use case
Discord	• Internal communication • Internal meetings
Trello	• Keep track of tasks and their current progress
Slack	• External communication with I-CONIC
Github	• Keep track of code • Keep track of issues/bugs
Google Meet	• External meetings with I-CONIC
E-mail	• External communication with mentor
Zoom	• External meetings with mentor

5.4.2 Within the group

Since this group contains seven different people it will take work to ensure that everyone knows what they are supposed to be doing at every step of the project. For most cases we will use written communication to relate information such as reminders and questions with each other using a private Discord server. This will generally occur in text messages viewable of all group members to aid with transparency and giving opportunity for all members to contribute with their knowledge. We hope for responses in this context to occur within a day. For more information intensive communication such as decision making and work delegation we will have verbal meetings (also using Discord) to ensure that everyone has a say in the matter.

5.4.3 With the client

Communication with our client I-CONIC began via e-mail but has since moved on to a Slack workspace, since they preferred it as a communication medium. Additionally – as is described under §5.1.1 – we will have meetings with them every three to four weeks to give status update on our progress; these meetings will be held over Google Meet as per I-CONIC's request. We believe that it is of upmost importance to have clear and honest communication with I-CONIC since we want the project to result in something useful to them in the end, which may not happen if we base our decisions on misunderstandings. So far, I-CONIC have been very swift in responding to e-mails and Slack messages. In addition to this, they have initiated some communication. During the course of the project I-CONIC has continuously been integrating our code with their code base and through this they have noticed

some aspects of the code that they wanted changed. Feedback like this has been very helpful, since it ensures both that the client gets a product they like, and that we grow as programmers.

Additionally we have sometimes presented different options to I-CONIC in order to get insight in what aspects of the program they value the most. The most relevant example of this is regarding the UI of the program, where we presented different concept art (see Figure 14) to them during the second sprint review. I-CONIC preferred information presented in the tool bar over the side panel, which allowed us to re-prioritize our efforts to accommodate for that during the third sprint.

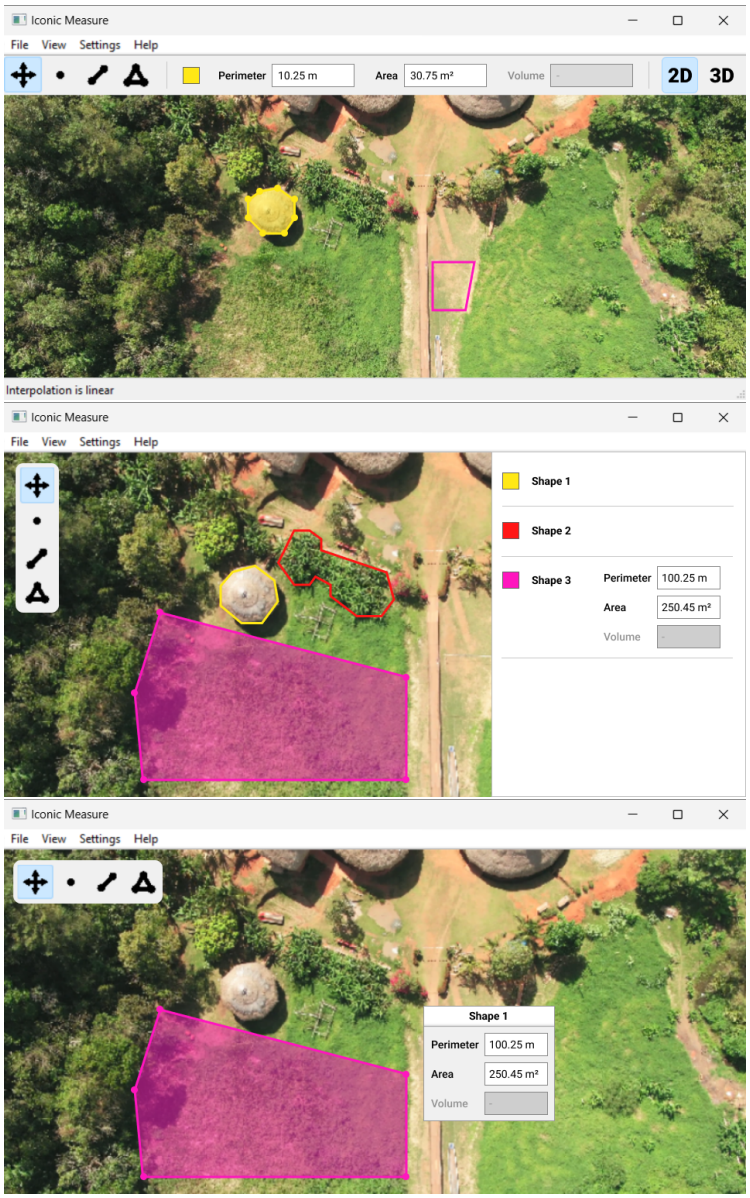


Figure 14: Concept art

We hope that future communication is similarly efficient, but we will still attempt to give them as much time to respond as possible and of course respect their working hours.

5.4.4 With our mentor

Communication with our mentor Christopher Peters has been predominantly via e-mail, with occasional Zoom-meetings for both mandatory and voluntary meetings. Early in the project we believed that communication with our mentor would not be all that important, something that we have since realized to be a faulty assumption. This course is generally quite vague in what its instructions and having someone with experience in the field to ask questions to has proven helpful. As in the case with emailing I-CONIC, we will give our mentor as much time to respond as possible and respect their working hours.

5.5 Documentation of meetings

Aside from the non-negotiable course deadlines there have also been meetings throughout the project. These are presented in the following large table and are commented on in subsequent sections.

Color	Explanation
Cerise	Internal group meetings
Cyan	Meetings with mentor
Orange	Meetings with client

Table 10: Test case color coding explanation

Date	Meeting description	Comment
2022-12-14 13:00-14:00	Initial meeting with course mentor	This meeting was a great introduction to the course and gave insight into how it functions. It also introduced us to our mentor who has been a big help so far in the project. It was conducted physically on the KTH Campus.

Date	Meeting description	Comment
2022-12-14 14:00-14:30	First internal meeting	This was the first time all the members of the group met. During this meeting the upcoming meeting with I-CONIC was discussed as well as a showcase of a project they had sent us took place. This meeting was conducted physically on the KTH Campus.
2022-12-20 09:30-10:30	First contact and project startup meeting	This meeting was very useful as it gave us great insight into I-CONIC as a company and what their product is and does. It was conducted physically in their offices.
2023-01-17 14:00-15:00	Second meeting with course mentor	This meeting gave some clarity regarding the required contents of Project Report 1. It was conducted physically on the KTH Campus.
2023-01-18 16:30-18:00	Internal meeting in preparation for sprint startup meeting	This meeting acted to gather our collective thoughts that we had gotten over the last month and to prepare for the upcoming sprint startup meeting. This meeting was conducted virtually over Discord.
2023-01-25 17:15-18:00	Internal meeting going over the stakeholders request	This meeting we went over the stakeholders request document we received from I-CONIC and discussed the feasibility of the requested features. This meeting was conducted virtually over Discord.

Date	Meeting description	Comment
2023-01-30 10:00-11:00	Sprint 1 startup meeting	This meeting functioned as the proper project initialization meeting after the holidays and exam period. The time between this and the last meeting gave us an opportunity to investigate the code that we had been given and figure out what specific information we were lacking. In accordance with advice given by I-CONIC a Trello board was created to keep track of tasks. It was conducted virtually via Google Meet.
2023-02-07 16:00-17:00	Internal meeting about work division for sprint 1	This meeting acted to distribute tasks for the upcoming sprint. In this meeting we split the group into two teams: those focused on the GUI part (Interface and Display module) and those focused on the functionality part (Functionality module).
2023-02-17 9:30-11:00	Internal meeting regarding upcoming assignments	This meeting was initiated by a mandatory survey in relation to this course. During the meeting the upcoming Project Report 2 and Halftime presentation were discussed. This meeting was conducted virtually over Discord.
2023-02-20 11:30-12:00	Meeting with course mentor regarding course work	This meeting touched on similar topics as the previous internal meeting and led to a lot of questions being answered. This meeting was conducted virtually over Zoom.
2023-02-23 10:15-11:00	Fourth meeting with course mentor	This meeting touched on similar topics as the previous one. It was conducted virtually over Zoom.

Date	Meeting description	Comment
2023-02-23 15:00-16:00	Sprint 1 review	This meeting was mainly focused on presenting our progress to I-CONIC. Simultaneously, we discussed what had gone well and asked questions regarding things that could be improved, taking advantage of the software development experience that I-CONIC has. In the end, the planning for sprint 2 also took place. The meeting was conducted virtually via Google Meet.
2023-03-17 14:15-16:00	Communication Workshop	This workshop conducted in the context of this course acted as an internal meeting for beginning the work on the Half-time presentation. This meeting was conducted physically on the KTH Campus.
2023-03-17 17:00-18:00	Internal meeting about assigning tasks	This meeting acted as a panic button when we realized the impending deadlines of Project Report 2 and the Half-time presentation were soon to arrive. This meeting was conducted virtually over Discord.
2023-03-19 10:00-11:15	Internal meeting going over Project Report 2	This meeting functioned as a soft, internal deadline for Project Report 2 in which we went over the contents of it and gave feedback on other peoples work. We also discussed the departure of a group member and how that will affect the project. This meeting was conducted virtually over Discord.

Date	Meeting description	Comment
2023-03-22 15:30-17:15	Internal meeting in preparation for the Halftime presentation	This meeting functioned to make sure that the presentation the following day went smoothly. This meeting was conducted virtually over Discord.
2023-03-23 11:30-11:50	Internal meeting about end of sprint 2	This meeting acted to regroup after the end of a stressful period. Work was divided and a deadline was set for 27/3 where another meeting will take place to collect the efforts of the sprint. Additionally, the initial work on our seminar was assigned. This meeting was conducted virtually over Discord.
2023-03-27 13:00-13:20	Internal meeting in preparation for sprint 2 review	This meeting was a short check in in the spirit of a stand-up where work was reported and future external meeting plans were discussed. This meeting was conducted virtually over Discord.
2023-04-02 8:00-9:20	Fifth meeting with course mentor	This meeting was an opportunity for us to get feedback on our second Project Report and to see how other groups had written theirs. This meeting was conducted virtually over Zoom.
2023-04-02 10:00-10:35	Sprint 2 review	This meeting was the culmination of the second sprint where the progress was presented to I-CONIC. In this meeting a lot of clarifications regarding implementation issues where made and some work was delegated to I-CONIC based on its complexity. This meeting was conducted virtually over Google Meet.

Date	Meeting description	Comment
2023-04-07 11:00-11:15	Internal meeting delegating work for the coming days	This meeting acted to divide work tasks for the coming days. The main focus was on making sure that the seminar would be planned in good time, but some people were also delegated to working on the final report. This meeting was conducted virtually over Discord.
2023-04-12 14:00-15:30	Internal meeting in preparation for the coming seminar	This meeting was a check in on the work that had been conducted over the previous days and was where the topics of the seminar were divided among the group members. A deadline for being prepared was set for the coming Sunday where a rehearsal was planned to take place. This meeting was conducted virtually over Discord.
2023-05-01 15:00-15:50	Internal meeting in preparation for the coming report deadline.	In this meeting the remaining work in the project was discussed and the workload for the coming week was delegated, with focus on the upcoming report deadline. This meeting was conducted virtually over Discord.
2023-05-02 14:00-14:40	Sixth meeting with course mentor	This meeting was an opportunity for us to get some guidance for what the third report should contain. Additionally, the final presentation and its structure was discussed. This meeting was conducted virtually over Zoom.

Date	Meeting description	Comment
2023-05-05 11:00-11:45	Seventh meeting with course mentor	This was an obligatory meeting in the context of the course that touched on similar topics as the earlier meeting. This meeting was conducted virtually over Zoom.
2023-05-06 11:00-11:40	Internal meeting for clarifying remaining work for report 3	In this meeting the remaining work for the third report was clarified and delegated. This meeting was conducted virtually over Discord.
2023-05-06 11:40-12:00	Internal teambuilding meeting	In an attempt to alleviate the stress of the final parts of the semester, some venting and friendly discussions were had. This meeting was conducted virtually over Discord.
2023-05-10 14:00-14:25	Sprint review 3	This meeting was a presentation of the progress of the third sprint for the client. They were pleased. Resources for the final presentation were also discussed. This meeting was conducted virtually over Google Meet.

5.5.1 Internal meetings

Most internal communication has been written but some meetings have been conducted to ensure consensus on certain topics. These meetings have usually been spontaneous and planned at most a few days ahead.

5.5.2 Mentor

Most meetings with our mentor have been planned by them and invitations sent around a week ahead of time. These have been informative meetings mandatory in the context of the course. On a few occasions we have asked for an extra meeting to get clarity on what (for example) Project Report 2 was supposed to contain.

Aside from requesting extra meetings there has been some more e-mail communication. The most noteworthy being the submission of a draft of Project Report 2 in which we received a lot of feedback that we then took to heart and tried to adapt

to, and the e-mail informing them of the departure of one of the group members. Communication in this manner has been effective and there has not been any delays caused by any unresponsiveness.

5.5.3 I-CONIC

I-CONIC has from the very beginning of this project been very communicative. They responded quickly to the first e-mail we sent them in which they invited us to their office for a presentation of them as a company and what they wanted us to do. When it comes to specifying dates for meetings and such they have been very understanding of our limitations as students and for each meeting there has been a discussion regarding figuring out the optimal time. So far there has been three meetings conducted with I-CONIC and further meetings will take place as necessary.

In addition to planning meetings there has been contact with I-CONIC in regards to working out problems with software installation as well as asking for and receiving images to use in the Half-time presentation.

5.6 Responsibility roles

To divide responsibility between group members in this course that place practically all the responsibility on the students, we previously had designated roles for each group member. A person with a specific role will make sure that the work that the group as a whole does in that area follows deadlines and is up to the standard set by both the course grading criteria and I-CONIC. This was meant to lessen the burden on each individual group member.

Name	**Roles**
David Aldenbro	GUI, Functionality
Sofia Andersson	Documentation, LaTeX
Linus Dinesjö	C++, Report
Elias Floreteng	Git, GUI
Pontus Lüthi	Git, C++
~~Leonard Petersson Hoogendoorn~~	~~Methodology, Report~~
Alexander Skalski	Communication, Seminars
Gustaf Sundmark	Documentation, Functionality

Table 12: Responsibility roles

5.6.1 Reflection on the responsibility roles

Although the roles were meant as guide to how the responsibility of work should be partitioned, this has largely not been the case in reality. This could be due to a variety of factors, such as (but not limited to): a mismatch between the roles and the actual work, a general lack of engagement, and an unclear definition of each role.

Instead, work partitioning has mostly been taken care of during meetings and by individual initiative. Unfortunately, this has not been ideal due to a general lack of engagement stemming from structural issues. The result of this has been that work has been pushed forward, resulting in emergency meetings where tasks have been assigned and worked on rapidly in order to meet the strict deadlines. The tasks themselves are worked on individually or sometimes in pairs. This is so that there is less dependency on each other, as we are all mostly working from separate locations with different schedules.

Going forward we are abandoning the current responsibility roles seeing as they have served minimal purpose. Instead, we plan to increase communication and create clearer delegations of work. Either by letting certain people have the responsibility of delegating the work or by having more concrete discussions well in advance of deadlines.

5.7 Project journal and reflections

In this section comments on noteworthy things in the project are made. It includes problems that have been encountered, changes that have been or are going to be made, and reflections on the progress of the project.

5.7.1 Problems with hardware and software requirements

An initial setback was faced during the installation of the preexisting software which we are to build upon. More precisely, an OpenCL-compatibility issue with AMD graphics cards as well as integrated graphics caused some team members to be unable to run the existing application. While the issue has not been solved, it has been worked around by distributing work based on capabilities, i.e. having group members with AMD graphics cards work on non-programming tasks such as report writing, or work through pair programming in order to contribute to the software development. The application's GitHub-installation page was updated to reflect the fact that the program assumes the user is using a dedicated NVidia graphics card. Additionally,

not all of the application's dependencies were initially listed by I-CONIC, which added to the confusion.

5.7.2 Sprint 1

For development, progress was swift within the first sprint, to a degree that the client was surprised by the significant progress made. It is also worth noting that the initial work mainly dealt with visual/UI aspects, which made the progress a lot more apparent than later work which was more focused on back-end.

5.7.3 Sprint 2

In order to coordinate the group in preparation for the half-time presentation and the project report deadline, we divided the group into two groups: programming and report/presentation, both containing four people. The above mentioned deadlines in part coincided with exam season. Most group members did not wish to dedicate much or any time to the project development course until after the exams, which led to slow progress on the project as the remaining limited human resources were mostly dedicated to course specific tasks. This led to more work being pushed back until the final days before the submission and very slow progress on the project.

In general, the group has deemed that the second sprint has been unsatisfactory, and that steps are to be taken to prevent similar situations it from occurring again. Due to the high standards set for the expected quality of the group's work, there is a need for a time buffer before submissions to allow for drafts to be submitted for review before final submission. One step to be taken in order to avoid similar outcomes in the future is to better plan out the division of labor and assign group members certain tasks and/or responsibilities early on, in order to incentivize people to get started earlier. Future schedules will also be evaluated more carefully to better foresee periods where group members may be busy with other tasks, allowing the scheduling of project tasks in advance.

5.7.4 Departure of a group member

A recent major setback has been one of the group members announcing they will no longer study at KTH; leaving the group project with immediate effect. This obviously increased the workload of remaining group members. Shortly before this occurrence the group had held a meeting assigning each member responsibilities related to the project report as well as half-time presentation. The member quitting left gaps in our planning, leaving some remaining members with additional work.

5.8 Change management

Over the course of this project a number of changes, both in assignment and logistics, have occurred. These are detailed here together with what led up to them and the consequences that the changes have had.

5.8.1 Appointment of group leader

During the early parts of the project it was noted that members of the group felt uncertain of what they should be doing. This arose, at least partially, from the very loose, ad-hoc like structure of the group where no one assigned tasks and people were free to choose freely what to work on. To mitigate this, a groupmember took it upon themselves to begin dividing, delegating and following up on work tasks. This course shift has been deemed successful as more people now know what they should be doing.

5.8.2 Reevaluation of time distribution

In the beginning of the project the group was under the false assumption that coding would be the majority part of the project. This led to other aspects such as the reports being down-prioritized and stressed over close to deadline. To mitigate this the group has changed focus and therefore begun prioritizing course work over coding. This has led to a better division of work and less stress.

5.8.3 Client rescinding a requirement

In early discussions with I-CONIC they requested a feature that they would later take upon themselves to create. This occurred after both the project group and I-CONIC collectively came to the realization that it was a more difficult feature to implemented than previously expected. This led to less work being required to complete the project and therefore more time that could be allotted to other features.

6 Risks

This section will define the risks that are associated with this project. This section will also define what we will do if a specific risk happens. We will define the risks by analysing things that could happen during the project. We will also create contingency plans so that when something happens we are prepared to handle it. An overview of the identified risks can be viewed in Table 13 together with a short description of planned contingency plans and a risk level indicating a mixture of estimated likelihood of occurrence and scale of impact if realized. Following this the individual risks are explained in more detail. Thus far one major risk has been realized, which was a group member leaving. We handled this by dividing up the work that that person had volunteered to do and by contacting our mentor.

Table 13: Project Contingency Plan

Risk	**Contingency Plan**	**Risk Level**
6.1 Difficulty in development	Utilize group knowledge and consult I-CONIC if necessary	Low
6.2 Production issues	Communicate and reassign work if necessary	Medium
6.3 Software/hardware compatibility issues	Utilize hardware resources and consult I-CONIC if necessary	Medium
6.4 Goal misalignment with stakeholders	Maintain clear communication with stakeholders and adjust project as necessary	Low
6.5 Component functionality issues	Develop project in smaller parts and mitigate issues	High
6.6 Group member unresponsiveness	Wait and redistribute work if necessary	Low
6.7 Communication breakdown with I-CONIC	Wait and consult mentor if necessary	Medium
6.8 Team member disputes/disappearance	Continue working and redistribute work as necessary	High
6.9 Finalization risks	Mitigate issues during development and testing	High

6.1 Difficulty in development

In the case that certain aspects of the development part of the project prove too difficult to successfully be dealt with, we will as a first step use the collective knowledge of the group to attempt to find solutions. If this does not yield results, we will contact I-CONIC to seek further clarification and information about these aspects, in an attempt to better understand and be able to undertake them. If it still proves too difficult for satisfactory completion, we will arrange alternative solutions with I-CONIC, potentially involving implementing features in different ways or shifting the focus to other features or aspects.

6.2 Production issues

There are multiple risks involved with the production aspect of the project, both for the programming tasks and the course-focused tasks. For the course focused tasks, we plan to preemptively minimize risks by employing careful planning and a well defined production structure. In the event that certain parts of the project cannot be or have not been completed as planned, we plan to as a first solution employ communication to find potential solutions. If communication is unable to be established, work may be reassigned internally. For production-related issues regarding the development aspect, we plan to communicate well with I-CONIC to as a first measure avoid such issues, and as a backup option rearrange plans such that the project can be completed in a satisfactory manner.

6.2.1 Risk encounter

Towards the end of the second sprint the course workload mixed with the exam period proved to be detrimental to project progress. To ensure a passing grade in the course the progress on the project needed to halt so that mandatory course work could be completed. This occurrence was not foreseen and caused a big change in the planning. In the future, we will try to foresee stressful periods of this nature and work to lessen their impact.

6.3 Software/hardware compatibility issues

For hardware-related issues, we plan to utilize effective solutions to best handle the problem at hand. For example, we may use the combined hardware resources of the group to solve problems. For hardware issues and also software compatibility problems related to the software issued or required by I-CONIC, we will contact

them for further assistance if the problems cannot be solved internally. For software issues related to software not required by I-CONIC (e.g. third party libraries), we will first attempt internal solutions. If unsuccessful, alternative solutions will be discussed internally and planned to be implemented.

6.3.1 Risk encounter

This risk has been encountered multiple times. The most detrimental one being the fact that one group member does not have hardware capable of running the project. Additionally, multiple members of the group had issues with the initial installation of certain third party libraries. These issues have been mitigated by employing the planned contingency.

6.4 Goal misalignment with stakeholders

There is a possibility of encountering miscommunication or misalignment with stakeholders, particularly with I-CONIC and our mentor regarding both the project work and course work. Stakeholders may have varying degrees of experience, expertise, goals, or expectations, which can lead to the implementation of unnecessary or erroneous work. Fortunately, we are in a favorable position since both I-CONIC and our mentor have been unambiguous about their objectives, requirements, and feedback. Additionally, they have expressed their willingness to consider our own solutions. As a result, the likelihood of this risk materializing is deemed to be relatively low. However, in the case of it being realized it should be simple to solve through effective communication.

6.5 Component functionality issues

As a preemptive measure to major functionality issues, we plan to develop the project in smaller individually functional parts, such that the functionality of the project as a whole is not reliant on a single link and can be at least to some extent demonstrated independently (§5.3.1). This will avoid issues where the product may be completely unusable due to the failure or lack of a single component. It will also avoid issues that may not be detected until testing, as individual components can be tested independently to some extent. When functionality issues are encountered, the first measure will be to employ effective solutions resulting in the satisfactory mitigation of the issue. If this does not result in a satisfactory level of mitigation, project plans may have to be changed, either to delay certain components of the product or to alter them. This will be communicated with the client.

6.6 Group member unresponsiveness

In the case of unresponsiveness from members of the group we will give them a few days to respond. If this does not yield any response the group will have to continue with the work without taking the missing member(s) into account, which will mean that their responsibilities will be delegated among the other members. In the case of extensive unresponsiveness the group will inform the course mentor for the sake of honesty and adherence to the EECS Code of honour (*KTH School of Electrical Engineering and Computer Science Code of Honour*, 2019).

6.7 Communication breakdown with I-CONIC

There is a non-zero probability of a breakdown in communication with I-CONIC. In the event that we are unable to get timely responses, we plan to first wait a reasonable amount of time for their response, and then consult our mentor for further direction. Other aspects of the project and product which are not dependent on the communication with I-CONIC may continue.

6.8 Team member disputes/disappearance

In the event of internal disputes, disappearance of team members or personal conflict between group members, the project may suffer production consequences. In similar regard to 6.6 the group will handle such situations by continuing to work like normal but with redistribution of the tasks delegated to the person who is unable to perform their tasks (which may be because they, for example, left the group).

6.8.1 Risk encounter

We encountered a setback that involved one of our team members leaving the course. This initially posed a challenge to the project, as the team member was responsible for certain tasks that now had to be distributed between the other group members. However the fact that the group size remained as large as seven members meant that the workload would not be too heavy for the remaining members. The contingency plan involved reassigning the lost members' tasks by identifying which team members had the necessary skills and capacity to take on the additional tasks. We also made sure that the team member who had left the course provided all the necessary documentation and information needed for the tasks to be completed. Another effect this can have on the project is that the other risks are more likely to occur, as we have less workforce to complete the tasks. We may therefore expect

delays in the project, and the risk of not meeting the deadlines is increased. We will however try to mitigate this risk by communicating with I-CONIC and our mentor to ensure that we are on track with the project, as part of our contingency plan. In addition the group will also try to identify what the previous member was currently working on and try to complete those tasks, or find a suitable replacement.

6.9 Finalization risks

There is a noteworthy risk of encountering issues with the finalization of the product and project as a whole. The preemptive measures explained in section 6.5 should to some extent avoid the worst of the potential finalization risks, though it is still an aspect worthy of consideration. To avoid significant challenges first being detected during the finalization stage of the project causing major issues, we will leave some margin in the planning to deal with such challenges which may be encountered.

References

GeeksForGeeks. (2023). *Software engineering | prototyping model.* https://www.geeksforgeeks.org/software-engineering-prototyping-model/. (Updated: May 3, 2023. Accessed: May 6, 2023)

GitLab. (2021). *4 tips for agile remote pair programming.* https://about.gitlab.com/blog/2021/02/04/remote-pair-programming-tips/. (Accessed: May 6, 2023)

GitLab. (2023). *What is a code review?* https://about.gitlab.com/topics/version-control/what-is-code-review/. (Accessed: May 6, 2023)

Hussain, D. (2022). *Course grading system.* https://canvas.kth.se/courses/35958/pages/course-grading-system?module_item_id=577958. (Not publicly accessible, Accessed: March 18, 2023)

I-CONIC Vision AB. (2022a). *Ansur and i-conic win €1.5m project on real-time 3d.* https://i-conicvision.com/2022/09/13/ansur-and-i-conic-win-e1-5m-project-on-real-time-3d/. (Accessed: May 7, 2023)

I-CONIC Vision AB. (2022b). *I-conic wins two on-board processing contracts.* https://i-conicvision.com/2022/08/15/i-conic-wins-two-on-board-processing-contracts-2/. (Accessed: May 7, 2023)

I-CONIC Vision AB. (2023). *I-conic vision, about us.* https://i-conicvision.com/about-us/. (Accessed: May 7, 2023)

Inkonova. (2023). *Inkonova autonomous mapping and inspection robotics with automated analytics.* https://spotscale.com/. (Accessed: March 18, 2023)

International Scrum Institute. (2021). *What is scrum?* https://www.scrum-institute.org/What_Is_Scrum.php. (Accessed: March 18, 2023)

Jacobsen, D. I. & Thorsvik, J. (2019). *Hur moderna organisationer fungerar*. Vigmostad & Björke AS.

Karpowicz, J. (2022). *3d models from drone data support disaster efforts and define city mapping possibilities.* https://www.commercialuavnews.com/public-safety/3d-models-from-drone-data-support-disaster-efforts-and-define-city-mapping-possibilities. (Accessed: March 18, 2023)

Kth school of electrical engineering and computer science code of honour. (2019). https://www.kth.se/en/eecs/utbildning/hederskodex/inledning-1.17237. (Accessed: March 10, 2023)

Lucidchart. (n.d.). *Types of uml diagrams.* https://www.lucidchart.com/blog/types-of-UML-diagrams. (Accessed: May 6, 2023)

Martin, M. (2023). *Prototype model in software engineering.* https://www.guru99.com/software-engineering-prototyping-model.html. (Updated: April 8, 2023. Accessed: May 6, 2023)

Meyer, B. (2014). *Agile!: the good, the hype and the ugly*. Springer.

Mintzberg, H. (1979). *The structuring of organizations*. Prentice-Hall.

Nishadha. (2022). *Uml diagram types guide: Learn about all types of uml diagrams with examples.* https://creately.com/blog/diagrams/uml-diagram-types-examples/. (Updated: September 28, 2022. Accessed: May 6, 2023)

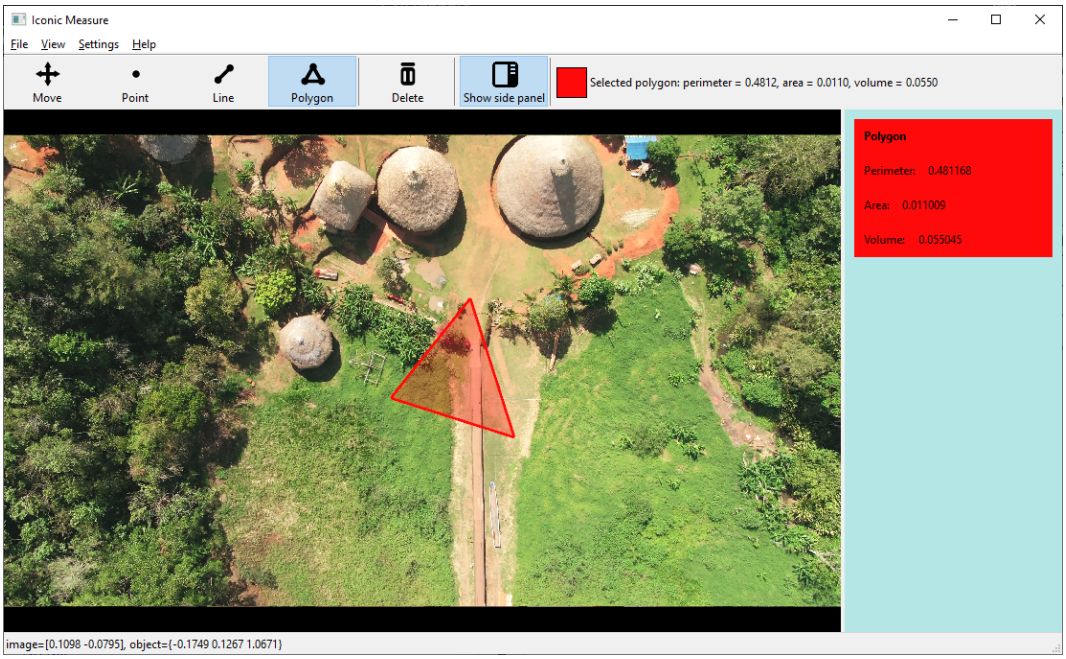
Sharp, H., Rogers, Y. & Preece, J. (2019). *Interaction design: beyond human-computer interaction* (5th ed.). John Wiley & Sons, Inc.

Spotscale. (2023). *Spotscale buildings decoded.* https://spotscale.com/. (Accessed: March 18, 2023)

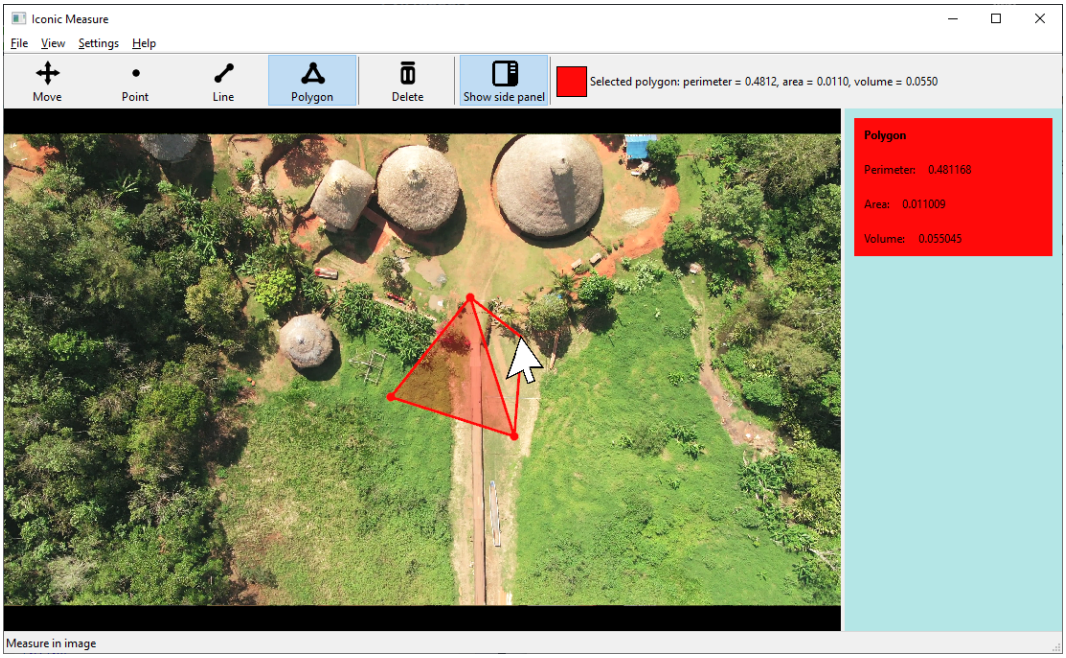
Wikipedia. (2023). *Code refactoring.* https://en.wikipedia.org/wiki/Code_refactoring. (Accessed: May 6, 2023)

A Test case images

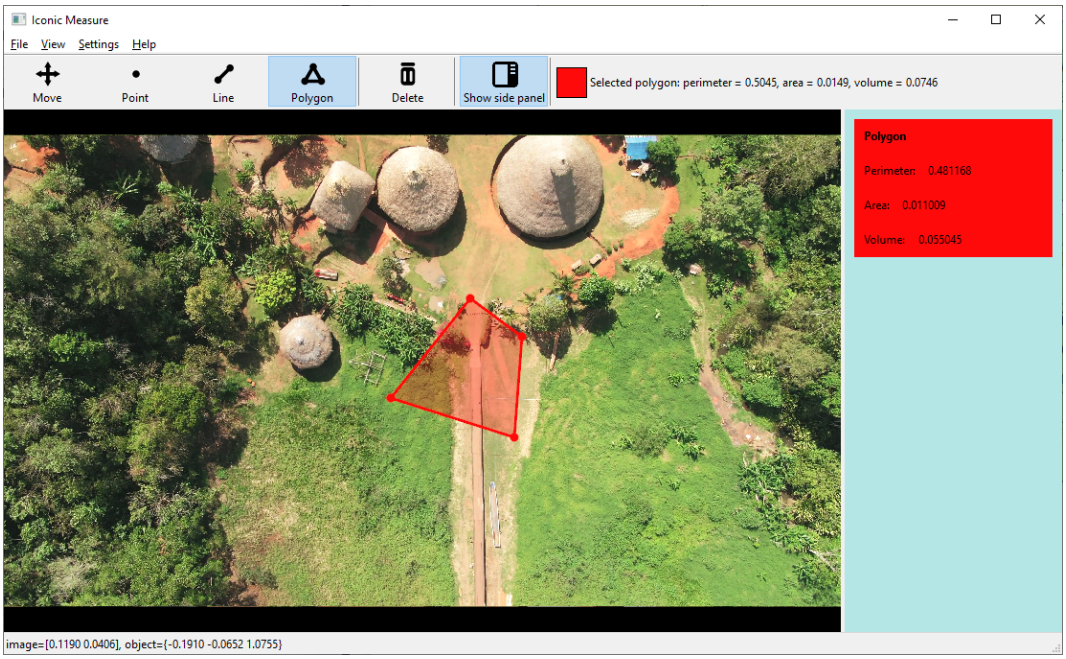
A.1 Polygon tests



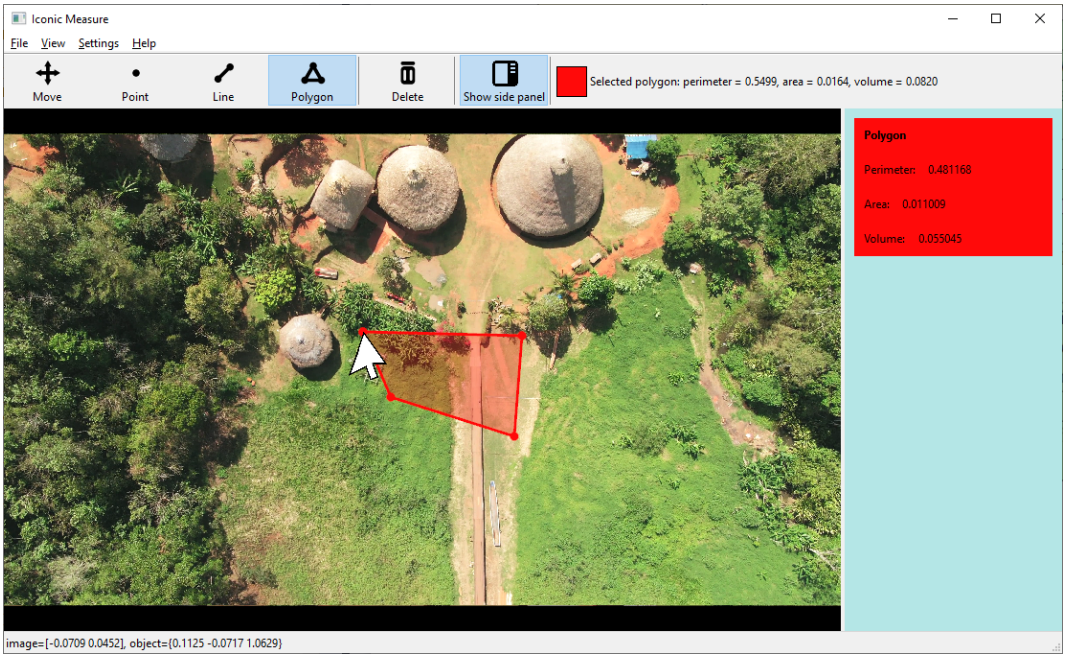
TC1: Creating a triangle. Note that the displayed measurement values are incorrect.



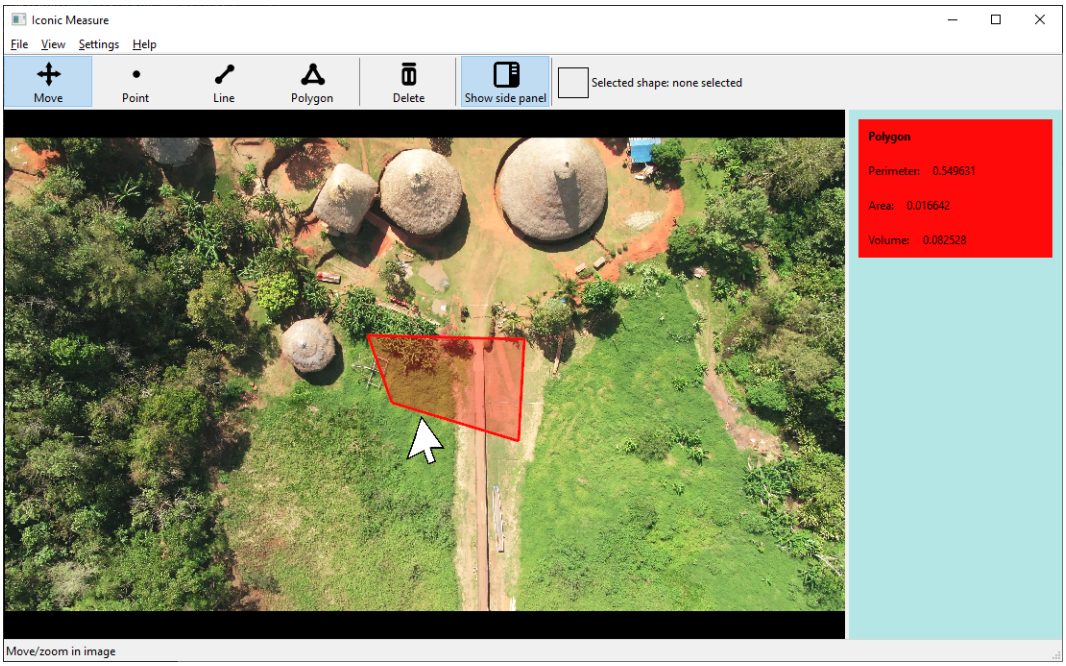
TC4: Selecting a polygon.



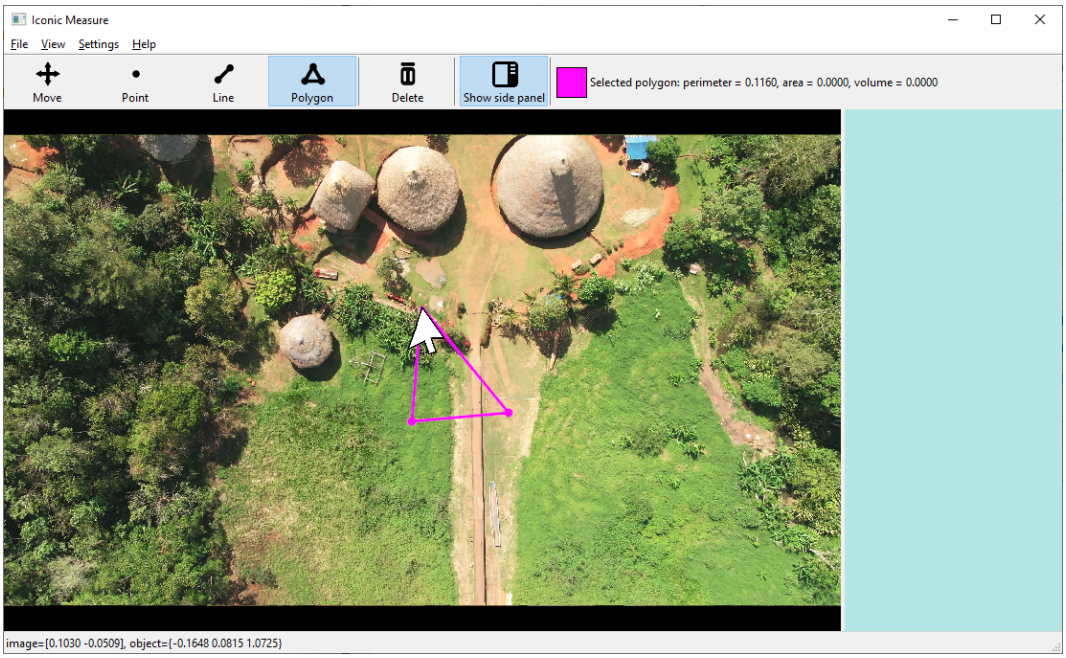
TC11: Adding a point to a polygon. Note that the measurement values are updated.



TC13: Moving a point of a polygon. Note that the measurement values are updated.

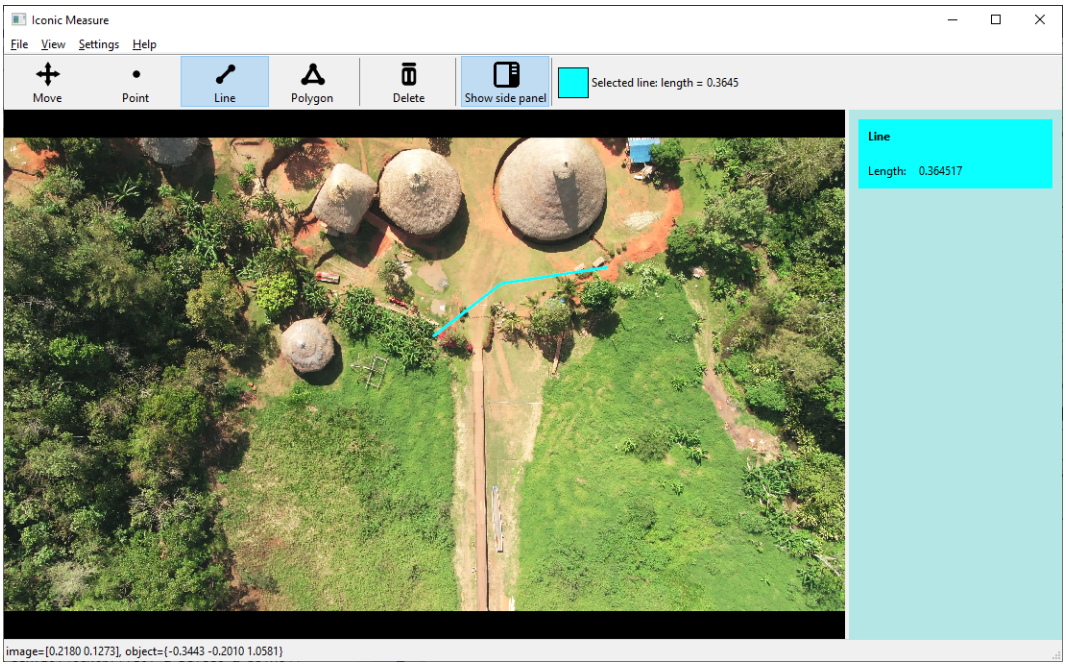


TC26: Trying, and failing, to select a polygon be clicking next to it.

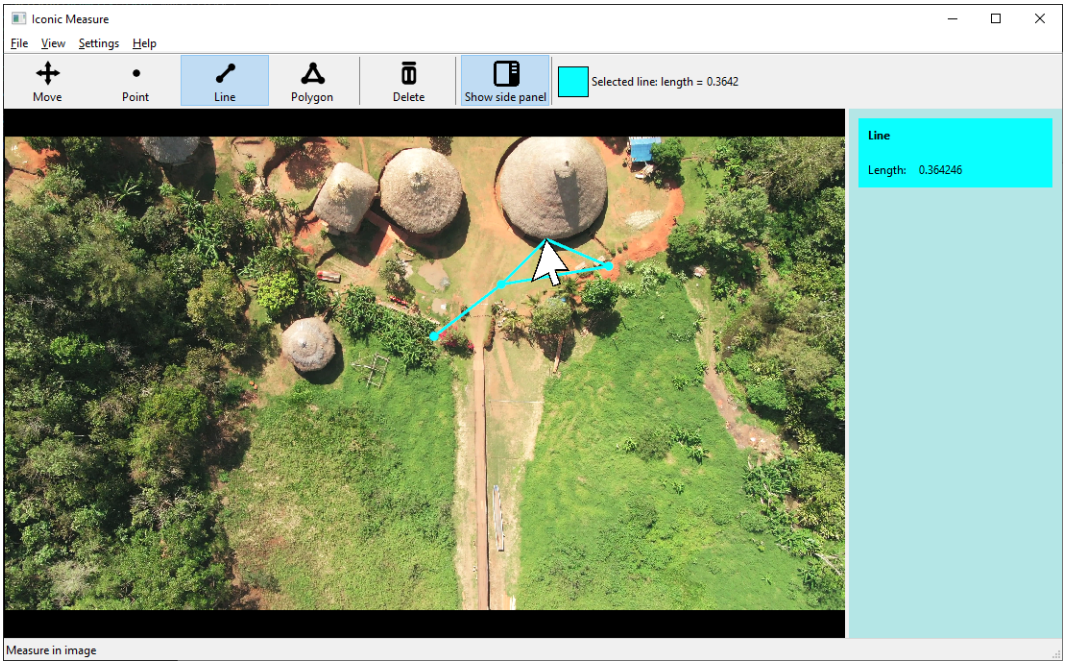


TC20: Attempting to create a polygon with only two points. When finishing this measurement the placed points disappear since the polygon is not finished.

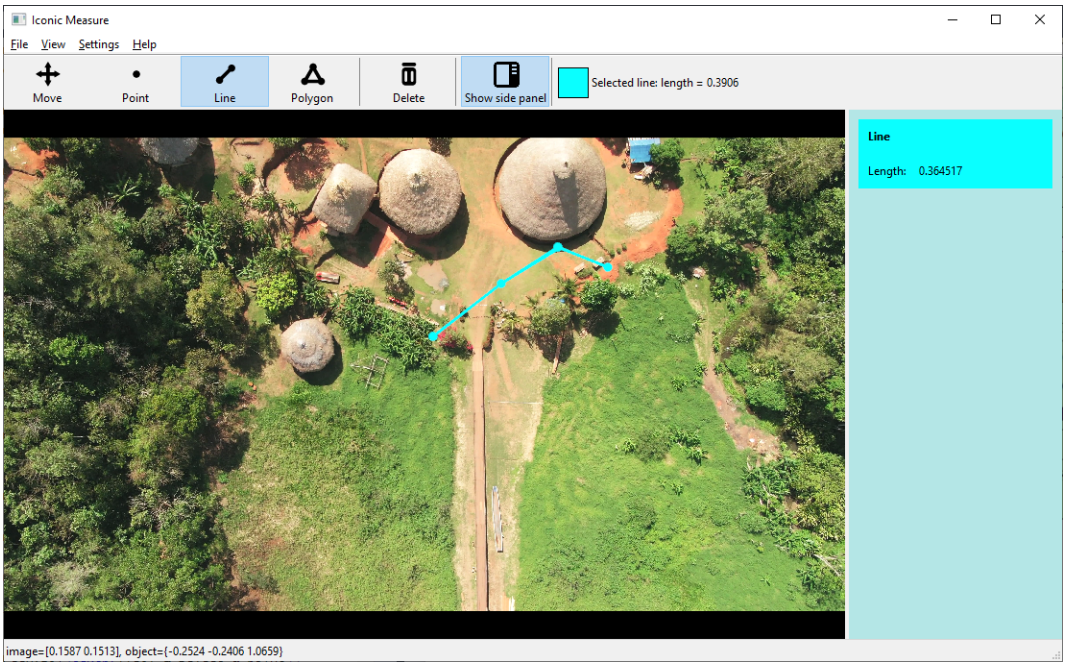
A.2 Line tests



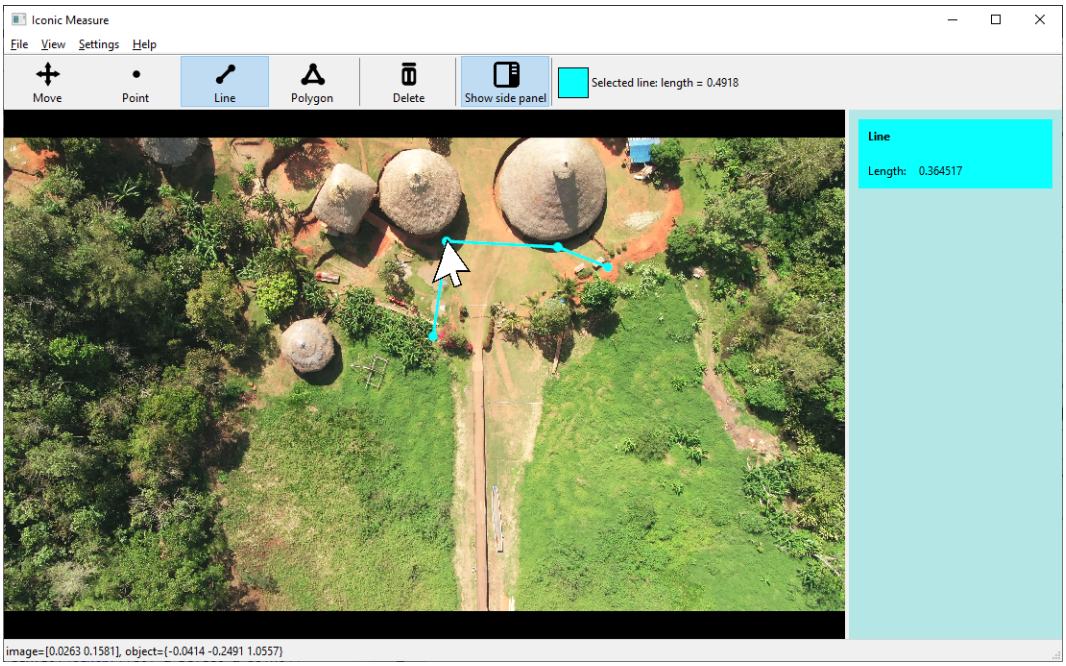
TC2: Create a line of three points. Note that the heightprofile of the line (see Figure 9 for reference) is not displayed.



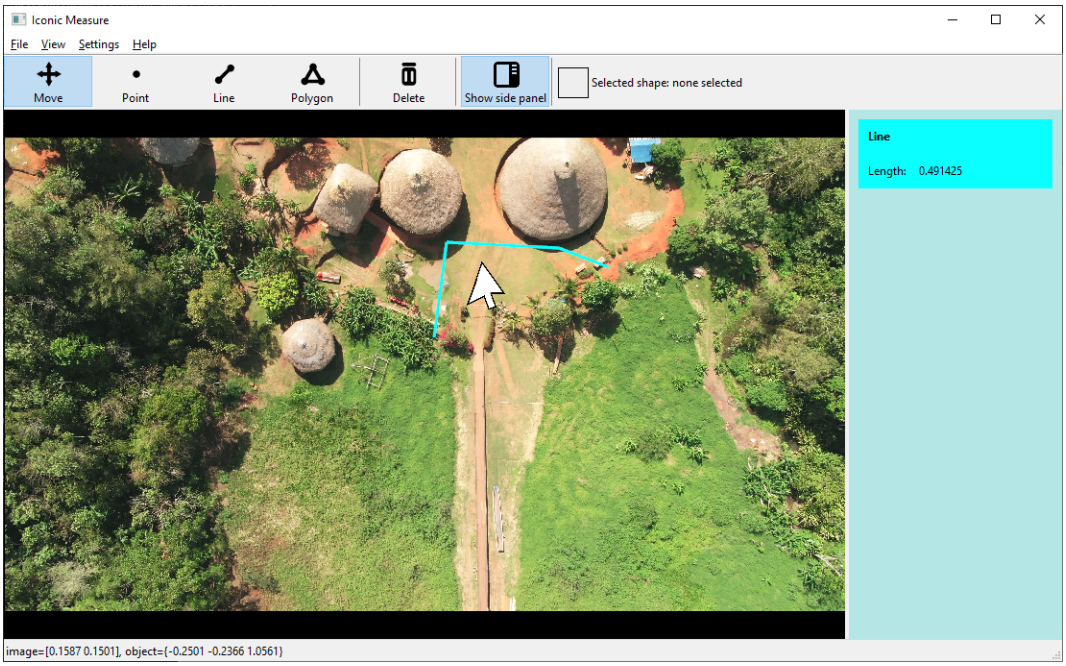
TC5: Selecting a line.



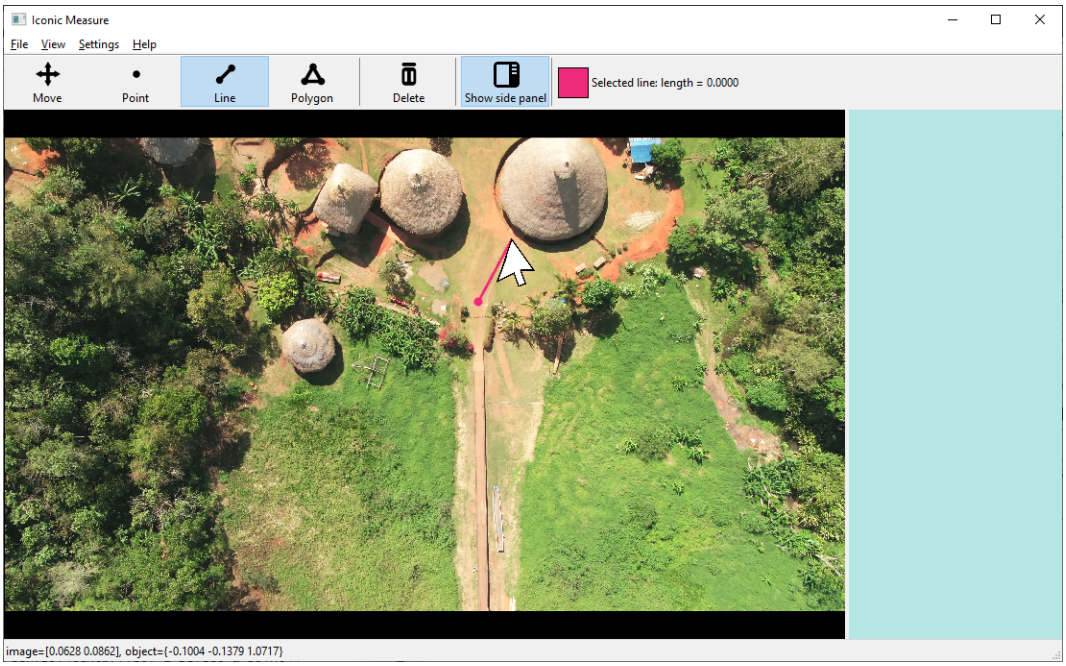
TC12: Adding a point to a line. Note that the measurement values are updated.



TC14: Moving a point of a line. Note that the measurement values are updated.

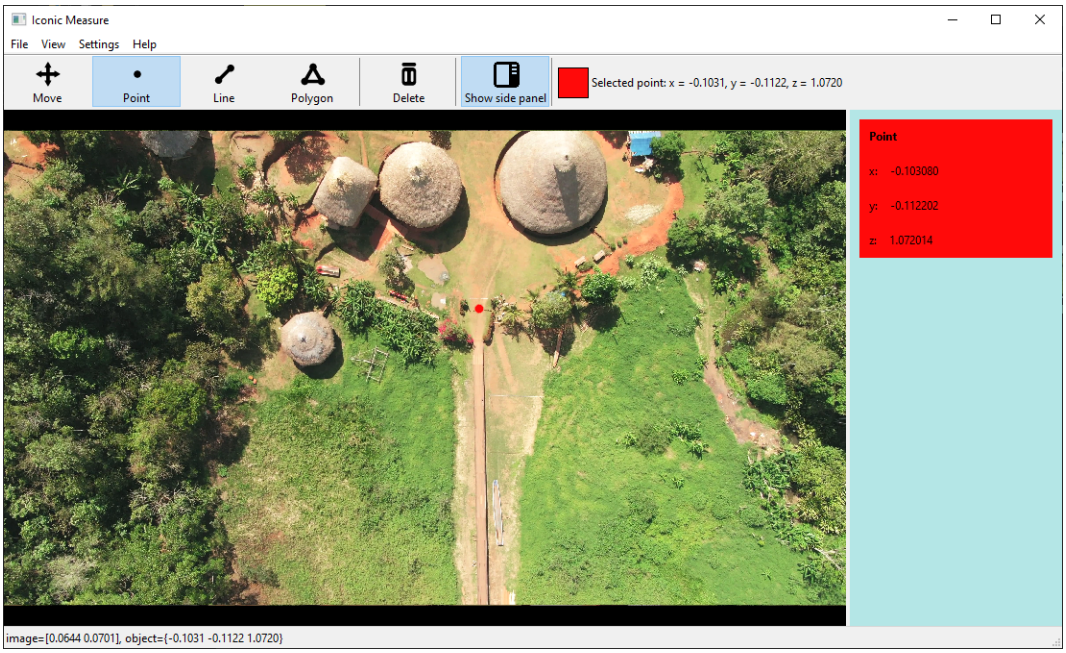


TC25: Trying, and failing, to select a line by clicking inside of it.

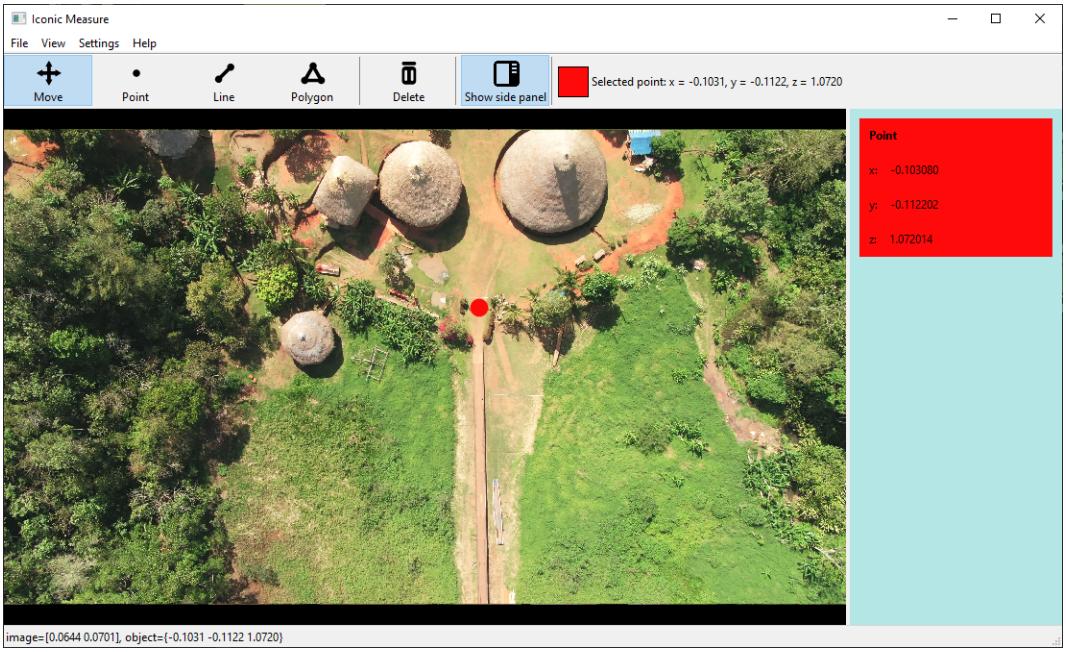


TC21: Attempting to create a line of only one point. When finishing this measurement the placed point disappear since the line is not finished.

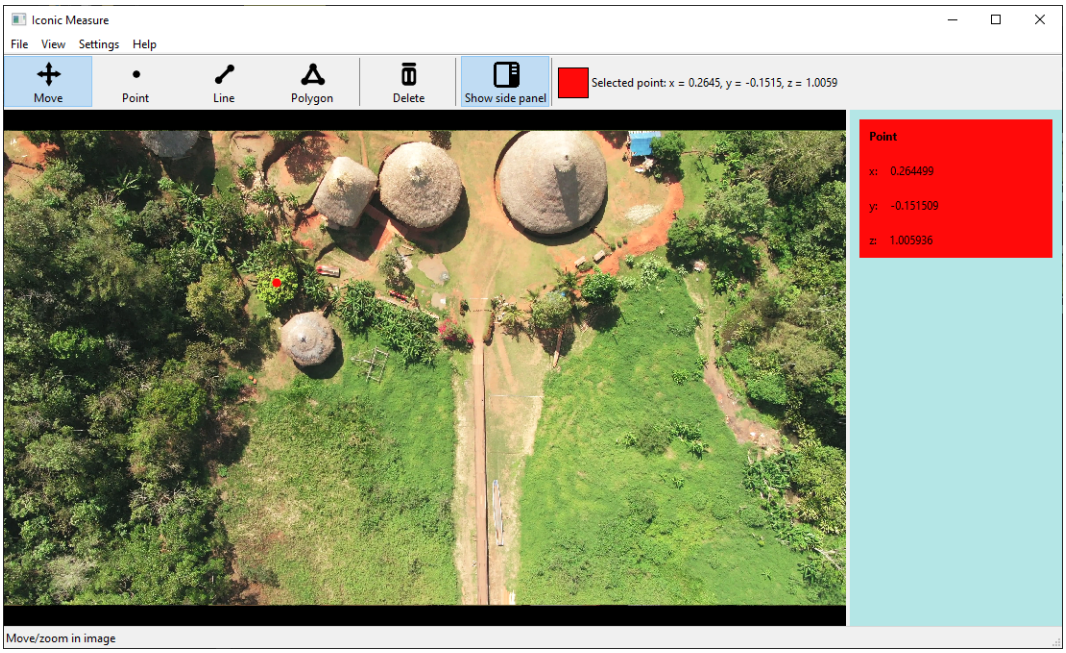
A.3 Point tests



TC3: Create a point.

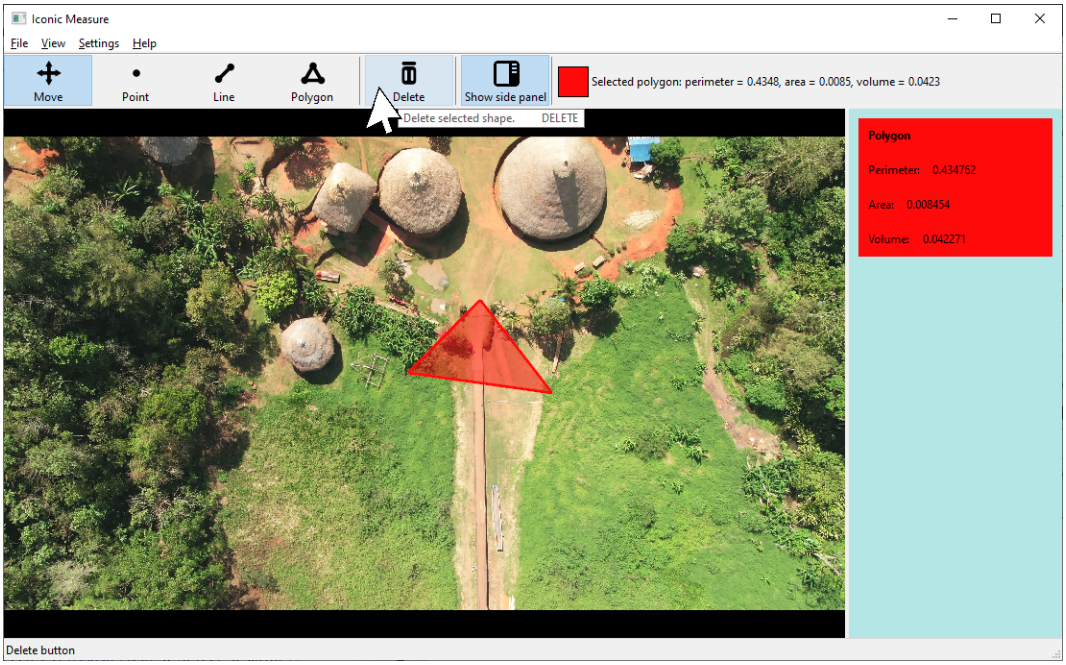


TC6: Selecting a point.



TC10: Moving a point. Note that the measurement values are updated.

A.4 Other tests



TC17–19: Deleting a shape. Deletion can occur either via the hovered button or the label indicated keyboard key DELETE.