HCU HafenCity Universität Hamburg

ALFRED-WEGENER-INSTITUT HELMHOLTZ-ZENTRUM FÜR POLAR- UND MEERESFORSCHUNG

Master of Science in Geomatics
Specialisation in Hydrography

MASTER THESIS

Immersive Virtual Reality Visualisation of the Arctic Clyde Inlet on Baffin Island (Canada) by Combining Bathymetric and Terrestrial Terrain Data

by
Mona Caroline Lütjens

28.05.2018

1st Examiner: Prof. Dr.-Ing. Thomas Kersten – HafenCity Universität, Hamburg, Germany
2nd Examiner: Dr. Boris Dorschel – Alfred-Wegener-Institut, Bremerhaven, Germany

Declaration §22(5) ASPO 2015

I declare, that this Master Thesis – in the case of group work the respective marked parts of the work – has been completed independently without outside help and only the defined sources and study aids were used.

Literally or copied passages or passages analogous in sense from different text sources are marked by referencing the respective sources.

Hamburg
28th of May 2018

(Mona Caroline Lütjens)

Acknowledgments

I would like to use this space to thank the people who have advised and supported me both on a professional and personal level during my thesis.

First of all, I would like to thank my thesis advisors Prof. Dr.-Ing. Thomas Kersten and Dr. Boris Dorschel for their guidance, support and motivation. I would like to thank Thomas Kersten for his advices during frequent meetings and for giving me access to required data. In particular, I would like to thank Boris Dorschel for giving me the chance to take part in the research cruise MSM66, for making it such an unforgettable experience and for his continuous support even when time was sometimes short.

I would like to give special thanks to Felix Tschirschwitz whose door was always open for me whenever I ran into trouble or had a question considering the VR application. Thank you for your creative ideas and solutions to improve the application.

Moreover, I would like to thank Tanja Dufek for her support regarding backscatter processing, Simon Dreutter for his information related to data acquisition and guidance on multibeam processing. Additionally, I thank Simon Deggim for his time finding solutions for further VR problems.

I owe my sincere gratitude to all interviewees who were involved in the user survey for this thesis project. Without their passionate participation and input, the validation of the application could not have been successfully conducted.

Furthermore, I would like to thank all my friends and colleagues for their technical and personal support, especially Eike Barnefske for his input on accuracies and Brendon Duncan for his advices concerning Unreal Engine 4.

Last but not least, I must express my deep gratitude to my parents and to my brother for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of writing this thesis.

Abstract

Due to recent advances in hardware and software technologies, virtual reality (VR) is becoming ubiquitous, finding its use in more and more professional applications apart from the gaming industry. Up to now, VR could be successfully implemented for virtual surgery, virtual therapy, flight and vehicle simulations and cultural heritage. Through VR it is possible to virtually explore computer-generated environments and to be immersed into most unique and remote areas without leaving the current real-life situation.

While geographical datasets are commonly projected in a top-down view onto a 2D surface leading to cognitive challenges and ambiguous interpretations, virtual reality could become a helpful tool to visualise terrain data in a more intuitive way. This thesis explores the advantages, limitations and practical usages of such visualisations. Furthermore, the thesis explores the feasibility of virtual reality presentations in terms of accuracy and performance and investigates potential benefits for combining terrestrial and bathymetric datasets.

Against this background, the water body Clyde Inlet located on eastern Baffin Island, Canada has been visualised in virtual reality. A methodology has been conducted to merge terrestrial and bathymetric datasets and to import vast digital terrain models into the virtual reality software Unreal Engine 4. In this work, the digital terrain model comprising Clyde Inlet was computed using the open source datasets of ArcticDEM, Canadian DEM, IBCAO as well as the bathymetric dataset acquired during the research cruise MSM66 in 2017. The result offers the possibility to observe the terrain with or without the water surface in VR by using three different ways of locomotion. Moreover, distinct measurements can be performed, and terrain textures can be altered. Further results include the accurate import of terrain data and ample performance. With the aid of a user survey, the usability and utility of terrain visualisations in VR for various disciplines was assessed.

Contents

List of Figures VI
List of Tables XI
List of Abbreviations XII

1 Introduction 1
1.1 Research Questions 2
1.2 Overview of the Expedition and Investigation Area 2
1.3 Structure of the Thesis 4
2 Theoretical Principles 5
2.1 Basic Principles of Sonar Mapping Systems 5
2.1.1 Multibeam Bathymetry 6
2.1.2 Multibeam Backscatter 8
2.2 Basic Principles of Virtual Reality 10
2.2.1 Explanation of Terms 10
2.2.2 Functionality of the HTC Vive 11
2.2.3 Basics of Unreal Engine 4 13
3 Data Acquisition and Processing 16
3.1 Materials 16
3.2 Hydrographic Data Acquisition and Processing 18
3.2.1 Hydrographic Data Acquisition 18
3.2.2 Processing of Multibeam Bathymetry 21
3.2.3 Bathymetric Surface Modelling 25
3.2.4 Backscatter Analysis 27
3.2.5 Quality Considerations 29
3.3 Terrestrial Data Processing 32
3.3.1 Processing of Digital Elevation Models 32
3.3.2 Quality Considerations 33
3.4 Combining Bathymetric and Terrestrial Terrain Data 34
3.4.1 Height Map 35
3.4.2 Splat Map 37
4 Development of Virtual Reality Application 38
4.1 Creating the World 38
4.2 Designing the Environment and Landscape 40
4.2.1 Lightening 40
4.2.2 Landscape Material and Foliage 41
4.2.3 Water Surface 44

4.2.4 Seagulls 45
4.3 Developing Locomotion 45
4.4 Evolving Dynamic Interactions 47
4.4.1 Flying and Teleportation Locomotion Display 47
4.4.2 Elevation, Distance and Compass Measurements 48
4.4.3 Height Difference and Slope Measurements 49
4.4.4 Menu 50
4.4.5 Controller setup and input 52
5 Results 54
5.1 Terrain and Environment 54
5.2 Motion Controller and Locomotion 58
5.3 Measurements 59
5.4 Menu 59
5.5 System Requirements 62
6 Discussion 63
6.1 Consideration of Accuracy 63
6.2 Consideration of Performance 65
6.3 Consideration of Usefulness 69
6.3.1 Usability of VR Application 69
6.3.2 Utility - Advantages, Limitations and Potential Applications 70
7 Conclusion and outlook 75
8 Bibliography 77
9 Appendix 81

List of Figures

Figure 1-1: Research vessel Maria S. Merian....3
Figure 1-2: Overview of Clyde Inlet....3
Figure 2-1: Geometry of transducer arrays for a Mills Cross configuration....6
Figure 2-2: Depth measurement and footprint size....7
Figure 2-3: Specular reflection and scattering....8
Figure 2-4: Interaction of pulse at oblique incidence angle and generation of a snippet...9
Figure 2-5: Reality-virtuality continuum....10
Figure 2-6: HTC Vive with HMD, controllers and base stations....11
Figure 2-7: Overview of Lighthouse tracking technology for the HTC Vive....12
Figure 2-8: Landscape architecture in Unreal Engine 4....15
Figure 3-1: Track lines coloured according to CTD-station number & line number....21
Figure 3-2: Diagram of processing multibeam bathymetry data....21
Figure 3-3: Seabed before the refraction correction....23
Figure 3-4: Seabed after the refraction correction....24
Figure 3-5: Removing outliers with the Subset Editor....24
Figure 3-6: Weighting schemes of surface interpolation....25
Figure 3-7: DEM of Clyde Inlet....26
Figure 3-8: Backscatter mosaic of Clyde Inlet....27
Figure 3-9: ARA result of one patch for port and starboard side....28
Figure 3-10: ARA result of Clyde Inlet....29
Figure 3-11: Standard deviation of surface....30
Figure 3-12: Processing workflow of terrestrial DEMs....32
Figure 3-13: Gaps in ArcticDEM are closed....33
Figure 3-14: Processing workflow for combining bathymetric & terrestrial data....35
Figure 3-15: Final DEM showing the IBCAO data (pink), interpolated transition (blue) and masks for critical locations (yellow)....36
Figure 3-16: Splat map....37
Figure 4-1: Overview for the development of the virtual reality application....38
Figure 4-2: Snow coloured automatically according to splat map....42
Figure 4-3: Rainbow colour ramp....43
Figure 4-4: Distant landscape without normal texture and with normal texture....44
Figure 4-5: Surface motion created from the texture sample and its result....45
Figure 4-6: Green area around the user shows all possible places that can be accessed..46
Figure 4-7: Compass texture indicating West, North, East and South....48
Figure 4-8: Key bindings for the left and right motion controller....52
Figure 5-1: The terrain of Clyde Inlet visualised in the VR application....54

Figure 5-2: Visualisation of the RGI dataset in UE4 ... 55
Figure 5-3: Default landscape scenery in UE4 ... 55
Figure 5-4: Landscape scenery in UE4 without the water surface depicting the default natural terrestrial colouration and the backscatter colouration ... 56
Figure 5-5: Landscape scenery in UE4 without the water surface depicting the Hillshade colouration ... 56
Figure 5-6: Landscape scenery in UE4 without the water surface depicting the Hillshade colouration and the depth related rainbow colouration ... 56
Figure 5-7: Lightning effects under water and above the water surface ... 57
Figure 5-8: Grass foliage and sun disc with bloom effect. Purple Alpine catchfly and common Heather foliage. Northern Androsace foliage ... 57
Figure 5-9: Seagulls up close and distributed in the world in UE4 ... 57
Figure 5-10: The left and right motion controller in VR ... 58
Figure 5-11: Flying display and teleportation display ... 58
Figure 5-12: Height difference, slope distance and slope measurements ... 59
Figure 5-13: Menu page “Basic Information” ... 60
Figure 5-14: Menu page “Positioning” ... 60
Figure 5-15: Menu page “Landscape” ... 60
Figure 5-16: Menu page “Controls” ... 61
Figure 5-17: Menu page “Screenshot” ... 61
Figure 5-18: Menu page “Exit Application” ... 61
Figure 6-1: Distribution of height measurements and height differences between the DEM and the VR application ... 64
Figure 6-2: Overall LOD system including automatic LOD generation ... 67
Figure 6-3: GPU cost of various passes ... 68
Figure 6-4: Degree of user-friendliness of the VR application ... 70
Figure 6-5: Summary of advantages and potential applications of VR ... 74
Figure 9-1: Processing of bathymetric and terrestrial terrain models ... 81
Figure 9-2: Level Blueprint: Activate Post Process Volume ... 81
Figure 9-3: Landscape Material: Colour snow according to splat map ... 82
Figure 9-4: Landscape Material: MPC switches between the default "Natural" view and "Hillshade" view ... 82
Figure 9-5: Landscape Material: Colour bathymetry according to splat map ... 82
Figure 9-6: Landscape Material: MPC switched between backscatter, rainbow colouration and Hillshade ... 83
Figure 9-7: Landscape Material: Backscatter colouration using the RGB splat map ... 83

Figure 9-8: Landscape Material: Rainbow colouration according to the terrain height...83
Figure 9-9: The entire Landscape Material 84
Figure 9-10: Blueprint “BP_Birds”: get object name of active birds and save it 85
Figure 9-11: Main Game Instance – Macro “Show Birds” 85
Figure 9-12: Main Game Instance – Macro “Hide Birds” 85
Figure 9-13: Persistent Level Blueprint: Navigation Mesh is being moved constantly....85
Figure 9-14: Blueprint “MotionControllerPawn”: Flying locomotion logic......85
Figure 9-15: Blueprint “Waypoint”: get name and location of waypoint and save it......86
Figure 9-16: Main Game Instance: Event “SwitchLevels”......86
Figure 9-17: Main Game Instance: Event “SwitchWaypoints” 86
Figure 9-18: Main Game Instance – Macro “ChangeWaypoints” 87
Figure 9-19: Widget "IngameMenuWidget2": On button clicked, the variables “Levels” and “Waypoints” will be initialised and passed to the Macro......87
Figure 9-20: Widget "IngameMenuWidget2": Macro casts to Game Instance Class to fire the event “SwitchLevels” which will load the corresponding Level 87
Figure 9-21: Widget "VelocityWidget" 87
Figure 9-22: Blueprint “MotionControllerPawn”: Set the location of the “Velocity Widget” relative to right controller and rotation relative to headset......87
Figure 9-23: Blueprint “MotionControllerPawn”: Create the function “ShowVelocity” to spawn the widget and “HideVelocity” to destroy the actor......88
Figure 9-24: Blueprint “MotionControllerPawn”: Toggle between flying and teleportation display using the Boolean “EnableTeleportation” 88
Figure 9-25: Blueprint “MotionControllerPawn”: The value for speed is being coloured yellow when its above 346 km/h and red when above 1382 km/h......88
Figure 9-26: Material “CompassLetters_M”......89
Figure 9-27: Widget "ElevationWidget”......89
Figure 9-28: Material “LaserMat” 89
Figure 9-29: Blueprint "BP_MotionController": Line trace from the motion controller with a length of 10 km......90
Figure 9-30: Blueprint "BP_MotionController": The source and target point of the laser Particle System are set to the start and end point of the line trace....90
Figure 9-31: Blueprint "BP_MotionController": Receive values for the elevation and distance of the hit point and pass it to the widget......90
Figure 9-32: Blueprint "Marker": Function “Set Marker1” 91
Figure 9-33: Blueprint "Marker": Function “Show Symbols”......91
Figure 9-34: Blueprint "Marker": Function “Set Location Symbols” 91

Figure 9-35: Blueprint "Marker": The rotation of the symbols is constantly adjusted ... 91
Figure 9-36: Blueprint "Marker": Macro “ChangeScale” ... 92
Figure 9-37: Blueprint "Marker": Determine the height difference, slope distance and slope between indicators and pass it to the Widget ... 92
Figure 9-38: Blueprint “MotionControllerPawn”: Attach the Widget on top of the second indicator and change its height according to distance from user ... 92
Figure 9-39: Widget “IngameMenuWidget”: Access second menu within first Widget ... 93
Figure 9-40: Widget “IngameMenuWidget”: on button hovered and clicked the corresponding Canvas Panel of the second menu is made active ... 93
Figure 9-41: Blueprint “BP_IngameMenu”: Cast to the first menu widget and bind events to make the second menu visible/invisible ... 93
Figure 9-42: Blueprint “MotionControllerPawn”: Activate Scene Capture Component when the Positioning menu page is switched on ... 93
Figure 9-43: Widget “IngameMenuWidget2”: Change the colouration of the terrain by inserting a parameter inside the MPC when the check box is activated ... 94
Figure 9-44: Widget “IngameMenuWidget2”: Change the visibility of the water surface and the collision on check state change ... 94
Figure 9-45: Widget “IngameMenuWidget2”: Check which colouration is active in the scene and tick the corresponding check box once the menu is opened ... 94
Figure 9-46: Widget “IngameMenuWidget2”: When the check box for the left controller is checked, the right check box is unchecked ... 94
Figure 9-47: Widget “IngameMenuWidget2”: Once the screenshot has been initialised on button click, both menu widgets are closed ... 95
Figure 9-48: Widget “IngameMenuWidget2”: Once the button to quit the game has been pressed, the function “Quit Game” is called to end the application ... 95
Figure 9-49: Blueprint "BP_MotionController": Create two functions that translate motion controller press and release input to left mouse button clicks ... 95
Figure 9-50: Blueprint "BP_MotionController": Laser spline stretches from motion controller to the widget once the widget is hit by the laser ... 95
Figure 9-51: Blueprint “MotionControllerPawn”: Allocate the pink and orange painted logo texture to the respective controller ... 96
Figure 9-52: Blueprint “MotionControllerPawn”: The motion controller trigger axis handles the flying movement ... 96
Figure 9-53: Blueprint “MotionControllerPawn”: Right face button 1 and 3 control the speed of the flying movement by doubling or halving it ... 96
Figure 9-54: Blueprint “MotionControllerPawn”: Right face button 2 and 4 toggle between the flying and teleportation movement ... 96

Figure 9-55: Blueprint “MotionControllerPawn”: The right trigger will activate the teleporter 97
Figure 9-56: Blueprint “MotionControllerPawn”: The right shoulder button executes the screenshot command 97
Figure 9-57: Blueprint “MotionControllerPawn”: The grip button toggles the visibility of the locomotion display 97
Figure 9-58: Blueprint “MotionControllerPawn”: The left trigger sets two measurement points onto the landscape 97
Figure 9-59: Blueprint “MotionControllerPawn”: The left shoulder button opens or closes the menu 98
Figure 9-60: Questionnaire for usability survey 99

List of Tables

Table 3-1: Overview of used datasets concerning....16
Table 3-2: Summary of hydrographic survey....18
Table 3-3: Summary of technical specifications of Kongsberg Simrad EM 122....19
Table 3-4: Technical data of Seabird SBE911plus CTD probe....20
Table 3-5: Accuracy assessment of multibeam system EM122....29
Table 3-6: Spatial resolution....30
Table 3-7: Elevation measurements of terrestrial terrain during processing steps....34
Table 4-1: Parameters of imported landscape in Unreal Engine 4....39
Table 5-1: Minimum and recommended system requirements for the application....62
Table 6-1: Specifications of three landscape sizes related to the terrain of Clyde Inlet...66
Table 9-1: Elevation differences between the DEM and the VR application....98

List of Abbreviations

ARA	Angular Range Analysis
ASTER	Advanced Spaceborne Thermal Emission and Reflection Radiometer
AVG	Angular Varying Gain
CARIS	Computer Aided Resource Information System
CDEM	Canadian Digital Elevation Model
CPU	Central Processing Unit
CTD	Conductivity Temperature Depth
DEM	Digital Elevation Model
EPSG	European Petroleum Survey Group Geodesy
FMGT	Fledermaus Geocoder Toolbox
GDEM	Global Digital Elevation Model
GeoTIFF	Georeferenced Tagged Image File Format
GLONASS	Global Navigation Satellite System
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GPU	Graphics Processing Unit
HDCS	Hydrographic Data Cleaning System
HIPS and SIPS	Hydrographic Information Processing System and Sonar Information Processing System
HMD	Head-mounted Display
HTC	High Tech Computer Corporation
IBCAO	International Bathymetric Chart of the Arctic Ocean
IDW	Inverse Distance Weighting
IHO	International Hydrographic Organization
LOD	Level of Detail
MBES	Multibeam Echosounder
MPC	Material Parameter Collection
MRU	Motion Reference Unit
NSIDC	National Snow and Ice Data Center
OTPS	OSU Tidal Prediction Software
PNG	Portable Network Graphics
RGI	Randolph Glacier Inventory
RMS	Root Mean Square
RTK	Real-Time Kinematic
SIS	Seafloor Information System
SVP	Sound Velocity Profile
TPU	Total Propagated Uncertainty
UE4	Unreal Engine 4
VR	Virtual Reality
WGS84	World Geodetic System 1984

1 Introduction

Alongside humanity's increase in knowledge, contents of the earth's surface whether tangible or intangible have been portrayed in maps. During prehistoric times, knowledge was kept in minds for it was too small to require recording and to develop charting techniques. But as time passed and the horizons of the world expanded beyond own communities, mental maps started to become inefficient and surroundings were depicted as rock paintings or engravings. In fact, earliest proven depictions of cartographic elements date back to the Neolithic and Bronze age (6000 – 700 BCE) (Misra & Ramesh, 1989).

With rising cultural heritage, representations of the earth's surface improved and maps were not anymore limited to the terrestrial terrain. Evidence for measuring the seafloor by sounding line date back to ancient civilisations. The depth of the sea was measured by lowering a weighted rope to the seafloor and measuring its length (Oleson, 2008). This rudimentary method became obsolete as the first acoustic sounding systems were developed in the early 20th century (Lurton, 2010). Thenceforward, echo sounding has proven to be an efficient and accurate method to depict the topography of the seafloor.

Over the time, as charting technologies advanced, maps became much more accurate and obtained geographically correctness. Visualisations of the earth's surface are not only used for navigation purposes but also to better understand geospatial relationships. However, geospatial problems and questions are often 3D in nature, yet data has to be depicted onto a 2D surface. Reducing the dimensionality will force users to compensate the missing dimension in their heads creating cognitive challenges. Nowadays, to help users, three-dimensional representations are obtained through physical reconstruction, hill shading or virtual models that receive the simulated 3D effect by turning it on the monitor screen.

Another method to depict terrain in three-dimensions is by using virtual reality (VR). Opposed to virtual models, the user is immersed in a computer-generated environment and becomes an actor rather than a spectator. Movements such as walking or head movements in real world are transferred to corresponding motions in the virtual environment letting the user immerse and decouple from reality.

VR has been developed over the last 20 years but was primarily used by experts such as the U.S. Air Force and NASA for applications such as training, simulation and research. Nevertheless, in the 1990s a common interest of the broader public existed but was soon diminished due to limited graphics and unpleasant quality (Steinicke, 2016). The revolution of VR came with rapid advancements in computer technology. 2016 became the year of virtual reality for the general public as several companies brought low-cost high-quality head-mounted systems to the market. Technological progresses, however, are mainly owed to the wealthy video game industry which can effort to invest so much in pioneering industry. Therefore, many applications are designed for video games and only recently also other domains of application find use (Fuchs, 2017). VR has already been successfully adopted for virtual surgery, virtual therapy to treat e.g. phobias, flight simulations for pilots, vehicle simulations for e.g. product conception, and cultural heritage to experience sites at their glory times or to draw attention to their archaeological relevance (Gutiérrez, Vexo, & Thalmann, 2008). At the HafenCity University a couple of virtual reality applications have been already evolved, amongst those the development of a virtual museum of the historic town house "Alt-Segeberger Bürgerhaus" which provides additional knowledge and serves moreover as educational material (Kersten, Tschirschwitz, & Deggim, 2017).

Against this background, the thesis will investigate the potential of virtual reality for other disciplines. On the basis of bathymetric data collected during the research cruise MSM66, the Canadian water body Clyde Inlet shall be visualised above and beyond the water line in virtual reality. The virtual reality application should provide furthermore, several functionalities which enables the user to interact with the landscape.

1.1 Research Questions

Throughout the thesis the following research questions shall be answered:

- How accurately can the data be visualised in a VR application?
- How demanding is the visualisation of large scale terrains in VR for computers?
- How beneficial is the combination of bathymetric and terrestrial terrain data?
- What are the advantage and limitations of displaying terrain data in an immersive 3D compared to a 2D presentation?
- In which scientific disciplines could the immersive 3D presentation find use?

1.2 Overview of the Expedition and Investigation Area

The expedition MSM66 started the 22nd of July 2017 in Nuuk, Greenland and ended the 28th of August 2017 in Reykjavik, Iceland. The overall track line of the cruise is shown in Figure 1-2. The cruise was organised by the Alfred-Wegener Institute, Helmholtz Centre for Polar and Marine Research and conducted by chief scientist Dr. Boris Dorschel. The main objectives of the cruise were palaeoceanographic research and the mapping of glacial seafloor features to study the Late Pleistocene and Early Holocene dynamics of the Laurentide Ice Sheet. The Laurentide Ice Sheet is the largest ice sheet at the end of the Last Glacial Maximum (roughly 20,000 years ago) which covered much of Canada and northern United States. Researchers are in hopes that by understanding past ice sheet behaviour, future predictions of ice sheet changes can be made (Dorschel, n.d.).

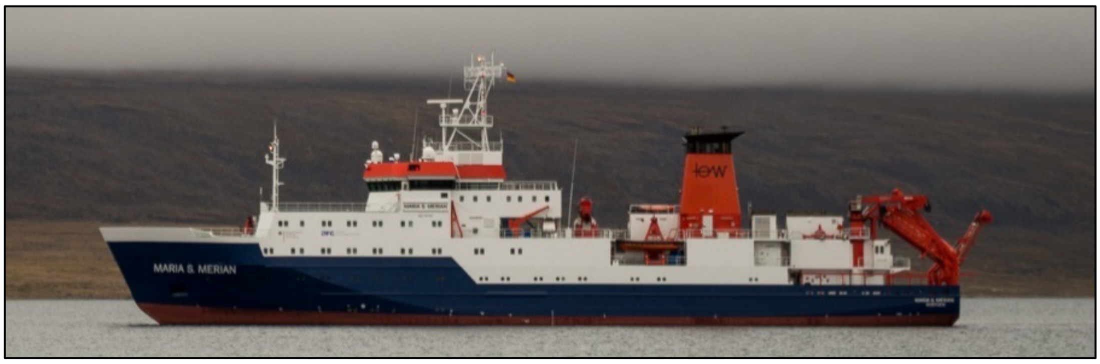


Figure 1-1: Research vessel Maria S. Merian (photographed by Simon Dreutter 15.08.2017)

The expedition was operated with the German research vessel Maria S. Merian (Figure 1-1), which was built from 2003 until 2005 at the Kröger Shipyard, Schacht-Audorf and is in service since February 2006. It is financed by the German government and owned by the Ministry of Education, Science and Culture, Mecklenburg-Vorpommern, represented by the Leibniz Institute for Baltic Sea Research. The ice margin vessel is named after the German entomologist Maria Sibylla Merian and can accommodate both 23 crew members and scientists. It has a length of 94.76 m and a width of 19.20 m. The vessel is operated by Briese Schiffahrts GmbH & Co.KG and available for other institutes as well (Briese Schiffahrts GmbH & Co. Kg., 2017).

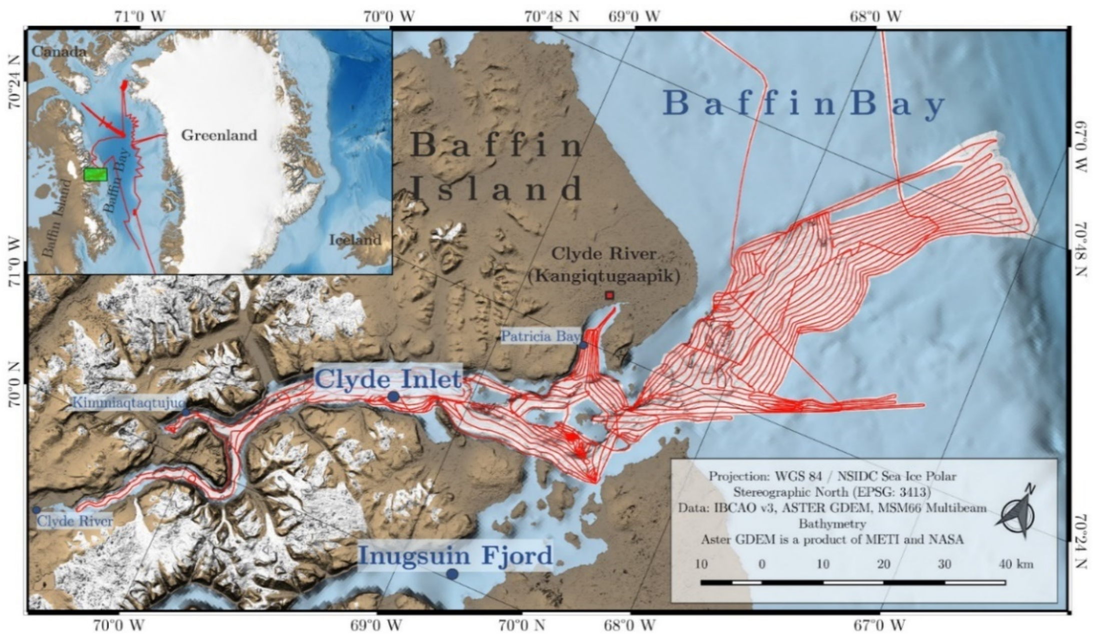


Figure 1-2: Overview of Clyde Inlet showing track lines (red) of cruise MSM66 (created in QGIS)

The area under investigation measures 160 km x 80 km and comprises the water body Clyde Inlet on eastern Baffin Island, Qikiqtaaluk Region, Nunavut, Canada. The elevation ranges from -847 m at the shelf edge to 1894 m at Baffin Island's interior plateau. Clyde Inlet is located in the Baffin Mountains which are part of the Arctic Cordillera mountain range – a north-eastern flank of the Canadian Arctic Archipelago. The Clyde Inlet system is a 120 km long fjord system that stretches from the Baffin Bay south-westwards towards Barnes Ice Cap. It includes three major geographic features which are branching roughly 20 km from the mouth of the system as seen in Figure 1-2.

The northern branch is called Patricia Bay and has a length of 12 km. At its sheltered cove the Inuit hamlet Clyde River (Inuktitut: Kangiqtugaapik or "nice little inlet") with approximately 1000 inhabitants is located (Municipality of Clyde River, n.d.).

The most southern branch is the Inugsuin Fjord with a length of about 90 km and the westward fjord is called Clyde Inlet. It has a length of 90 km as well but branches after roughly 60 km into two arms. The northern arm (12 km long) is called Kimmiaqtaqtujuq which is a hunting place of the Inuit. The southern branch is 40 km long and has the water stream Clyde River at its head which outflows from Generator Lake at the southern end of Barnes Ice Cap (Government of Canada, 2016) (Wykes & Inuit Heritage Trust Inc., 2007).

This thesis examines only the fjord Clyde Inlet including Patricia Bay. Hence, the terminology of "Clyde Inlet" will refer in the following chapters to the branching fjord and not to the entire system.

1.3 Structure of the Thesis

After the introduction was given in chapter 1, chapter 2 explains at first the basic principles of acoustic systems with regards to multibeam bathymetry and backscatter theory. Moreover, virtual reality will be briefly explained with respect to the used hardware (HTC Vive) and software (Unreal Engine 4). Chapter 3 describes the acquisition and processing of the multibeam bathymetry data and the combination with other digital elevation models. Subsequently, the development of the virtual reality application will be covered in chapter 4 and all results will be presented in chapter 5. Chapter 6 then discusses the results with respect to the research questions made in chapter 1 on the basis of a brief user survey. Finally, the conclusion sums up the findings at the end of the thesis in chapter 7.

2 Theoretical Principles

The second chapter briefly outlines basic principles regarding multibeam echosounders considering their acquisition of bathymetry and backscatter data. Likewise, the term virtual reality will be explained and the hardware as well as software used for this thesis introduced. Hence, only information relevant for this thesis will be covered.

2.1 Basic Principles of Sonar Mapping Systems

A sonar is a system that uses sound waves to navigate, communicate, detect, locate or map features in the water column, onto or within the seabed. Sonar is an acronym of Sound Navigation and Ranging and draws thus parallels with the electromagnetic radar (Radio Detection and Ranging). There are two types of sonars: active sonars and passive sonars. While active sonars transmit signals and receive echoes from a target to determine its position, passive sonars just intercept noises that are radiating from a target (Lurton, 2010).

In order to locate depth measurements accurately, sonar mapping systems have to receive additional information from ancillary systems such as positioning and attitude systems. The positioning system is used to determine the geographical position of the ship and to transform local coordinates into a global frame. Therefore, GNSS receivers are deployed on the ship which determine their own location and velocity by measuring pseudo ranges to satellites and receiving decoded navigation messages from satellites along a line of sight. The position is calculated using a minimum of four satellites (Hofmann-Wellenhof, Lichtenegger, & Collins, 2001).
To compensate for vessel motions such as roll, pitch, yaw and heave, motion sensors such as inertial navigation systems are furthermore mounted on the ship. Inertial navigation systems consist of three accelerometers to measure the linear acceleration and three angular rate gyroscopes to measure the angular motion. Knowing the starting position and integrating the acceleration twice, the trajectory of the moving vessel can be determined. Using satellite and inertial data in a combined system, will provide continuous and precise positioning results. GNSS dropouts, shadowing effects and MRU inertial drifts can be minimized (Hofmann-Wellenhof, Lichtenegger, & Collins, 2001).

All sensors need to be aligned in a vessel coordinate system, relative to a reference point on the ship. A measured point surveyed with an acoustic system, thus, refers to the vessel's frame and can be transformed to geographical global coordinates through GNSS. It is furthermore important that time offsets (latency) between depth measurement and positioning is as small as possible.

2.1.1 Multibeam Bathymetry

Multibeam echosounders are active sonars and were developed from single beam echosounders (SBES) to record the bathymetry. Apart from SBES, MBES emit instead of a single beam, a fan of sound waves to increase the bottom coverage. To emit and receive sound waves, transducers are used which are commonly mounted on the ship's hull. They can both transmit and receive sound waves by converting electrical energy into sound and vice versa (Ingham, 1992). Mostly, transducers are deployed in an orthogonal Mills Cross configuration meaning that the emitting transducer is located in fore and aft direction and the receiving hydrophone consisting of several transducers is placed athwartships (Figure 2-1).

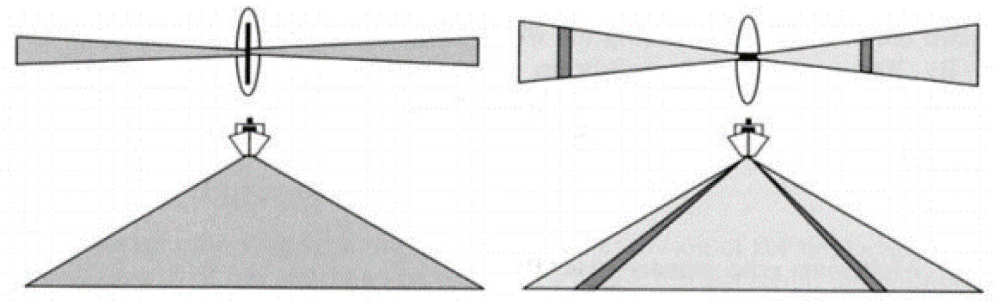


Figure 2-1: Geometry of transducer arrays for a Mills Cross configuration. Left: Transmitter in fore and aft direction emits sound as broad as possible across-track and narrow as possible along-track. Right: Hydrophone placed athwartships receiving sound in segments (Lurton, 2010)

The transmitter generates sound pulses athwartships as broad as possible and in the fore aft direction as narrow as possible. The sound lobes travel with speed of sound through the water column until they are reflected by the seafloor. Most energy is concentrated in the main sound lobe which is defined by -3 dB points. This limit specifies where acoustic energy has fallen to half. The angle between that limit and the main axis is called beam angle and defines the ensonified area on the seabed: the footprint (Figure 2-2). The footprint furthermore increases with increasing depth (Jong, Lachapelle, Skone, & Elema, 2010).

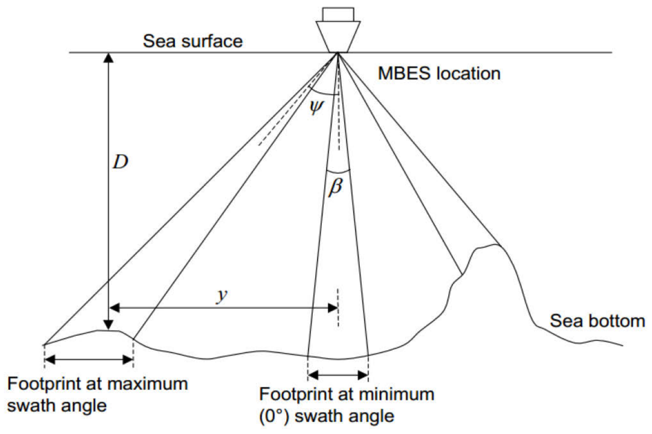

Sea surface
MBES location
D
Ψ
β
y
Sea bottom
Footprint at maximum swath angle
Footprint at minimum (0°) swath angle

Figure 2-2: Depth measurement and footprint size depending on the swath angle Ψ (Jong, Lachapelle, Skone, & Elema, 2010)

Upon return, the reflected sound is received in segments by the hydrophone resulting in a high number of soundings for each transmitted pulse. Segments are multiple smaller beams each having a beam width of around 1 – 3 degrees. The depth of each beam is determined at reception by looking at each time-angle pair to reconstruct the trajectory of the wave and the position of the footprint (Lurton, 2010). Using the two-way travel time ΔT, the water sound velocity c and the incident angle Ψ, the water depth D at the across-track position for each sounding y can be calculated as follows (Jong, Lachapelle, Skone, & Elema, 2010):

$$D = \frac{1}{2} c \Delta T cos \Psi \qquad (2\text{-}1)$$

$$y = \frac{1}{2} c \Delta T sin \Psi \qquad (2\text{-}2)$$

The above-mentioned equations only reflect the case in which sound velocity is constant in the entire water column. This, however, is rarely the case as the sound velocity depends linearly on temperature, salinity and pressure. Continuous changes in the sound velocity will progressively refract the sound wave and thus change its initial direction. To account for this, sound velocity profiles need to be acquired and more complex equations are needed to reconstruct the acoustic path. Sound velocities can be measured by using a CTD (Conductivity, Temperature, Depth) probe. A CTD is an electronic device which is lowered into the water and measures conductivity, temperature and pressure throughout the water column to compute the salinity, depth and sound velocity (Lurton, 2010).

Typical frequencies for deep water echosounders range from 12 kHz to 30 kHz and for shallow waters 100 – 200 kHz. It shall further be noted, that to achieve high accuracy, a high resolution as well as a narrow beam is needed. The higher the frequency, the shorter is the wavelength and the narrower the beam width. However, the higher the frequency the higher is also the absorption (Lurton, 2010).

The list of error sources considering multibeam systems is long. Major errors are (Jong, Lachapelle, Skone, & Elema, 2010):

- System measurement errors due to the system's electronics
- Depth measurement error due to beam width
- Beam angle error due to system's electronics (increases with swath angle)
- Acoustic propagation errors due refraction
- Effect of attitude errors (increase with swath angle)
- Transducer misalignment error
- System calibration errors
- Random errors such as external noise or reverberation of seafloor

2.1.2 Multibeam Backscatter

Multibeam echosounders do not only measure the travel-time of soundwaves to determine the bathymetry, they also record the intensity of the backscattered signal. As a slanted sound wave hits the seafloor, most of the signal is scattered in all directions. Some parts will also penetrate into the seafloor depending on the acoustic impedance. The signal with the highest strength is called specular reflection and the signal returning to the sonar is called backscattered signal (Figure 2-3) (Lurton, 2010).

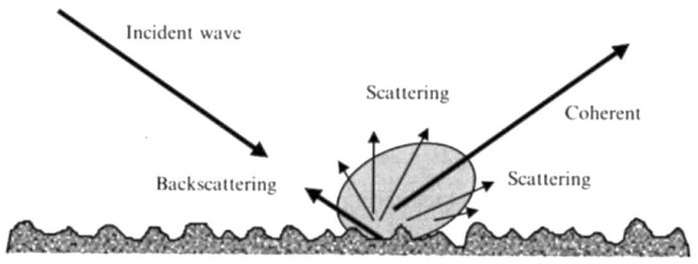


Figure 2-3: Specular reflection and scattering of an incident wave by a rough surface (Lurton, 2010)

Each bottom type has a different acoustic impedance related to the hardness, softness and roughness of the material and thus controls the angle and amplitude of the scattered signal. While rough materials scatter acoustic energy more homogenously and therefore yield a more stable intensity independent from the incidence angle, smooth fluid materials will provoke a mirror like response and a strong specular response in symmetrical direction. Hard seafloors such as rocks furthermore result in high echoes with respect to soft sediments. Acquiring these information, can be used to determine bottom type and seafloor characteristics. The backscatter strength is an initial method to interpret habitat as a relative rather than a direct measuring tool (Lurton, et al., 2015).

Multibeam systems furthermore collect backscatter data as a time-series of acoustic measurements along each beam which is referred to as “snippet”. At oblique incident angles, the acoustic pulse interacts with seafloor well before it reaches the beam maximum axis resulting in much more extended returns than in near nadir sections. The elongated echo is the snippet as seen in Figure 2-4 (Lurton, et al., 2015).

The backscattered strength, however, is not only dependent on the seafloor characteristics but also on the frequency and incidence angle. Close to the nadir, the specular reflection is the backscattered signal which returns with the highest echo level from the seafloor. At oblique incidence angles, the intensity now mainly depends on the impedance contrast and scattering increases because fewer elements of the seafloor point back towards the echosounder. At grazing angles, the backscatter response collapses (Lurton, et al., 2015).

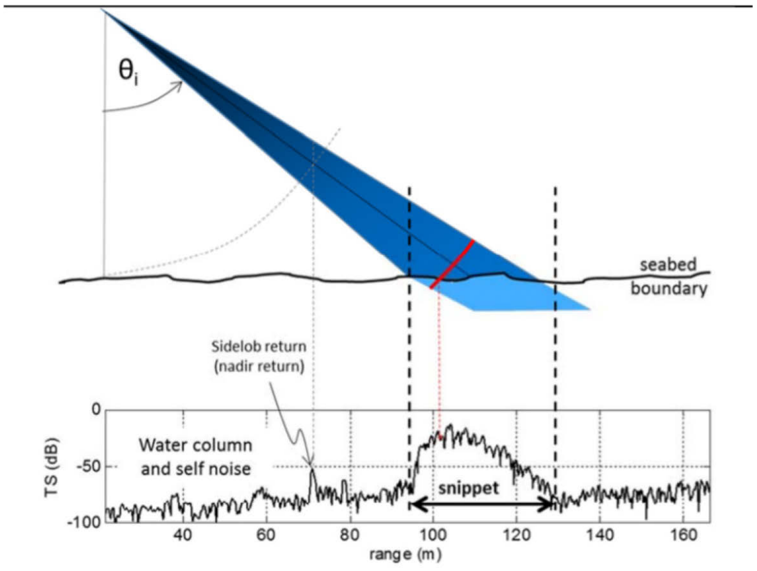


Figure 2-4: Interaction of a pulse at oblique incidence angle and the generation of a snippet. (TS: target strength) (Lurton, et al., 2015)

In order to access the backscatter information, the signal has to be firstly filtered to attenuate noise and amplified using static gains to compensate for transmission loss. Moreover, all influences from the sensor such as transmission level, reception sensitivity, beam aperture, signal duration and frequency need to be removed because they impact the received echo intensity. Then the measured signal intensity can be translated to the backscattered strength (Lurton, et al., 2015).

In post-processing several corrections have to be applied as well. The angle varying gain (AVG) compensates for the influence of the seafloor relative to the incidence angle to normalise measurements across the swath. A slant range correction will then determine the horizontal location of each data sample (Lurton, et al., 2015).

2.2 Basic Principles of Virtual Reality

2.2.1 Explanation of Terms

The oxymoron virtual reality can be defined as a computer-generated digital environment either realistic or fantastical that can be experienced and interacted with as if it was real. Jerald (2016) defines the goal of virtual reality "to completely engage a user in an experience so that she feels as if she is present in another world such that the real world is temporarily forgotten, while minimizing adverse effects [health effects such as motion sickness]". Next to the perception of believable places and the ability to interact with objects, virtual reality must also offer real-time rendering while changing the perspective corresponding to walking or head movements (Steinicke, 2016).

In order to achieve an illusionary world that can be believed, two essential factors have to be understood: immersion and presence. Immersion relates to the physical characteristics of the user interface of a VR application. Presence, on the other hand, defines a psychological state in which the user deeply senses the environment and feels involved with. It is the "awareness of being immersed in a virtual world while having a temporary amnesia or agnosia of the real world" (Jerald, 2016). The greater the immersion, the higher is the potential of presence (Jerald, 2016).

Apart from the term virtual reality, many other terms related to virtual experiences exist: augmented reality, augmented virtuality or mixed reality. To distinguish and classify these terms, a virtuality continuum can be introduced as illustrated in Figure 2-5 (Milgram, Takemura, Utsumi, & Kishino, 1994).

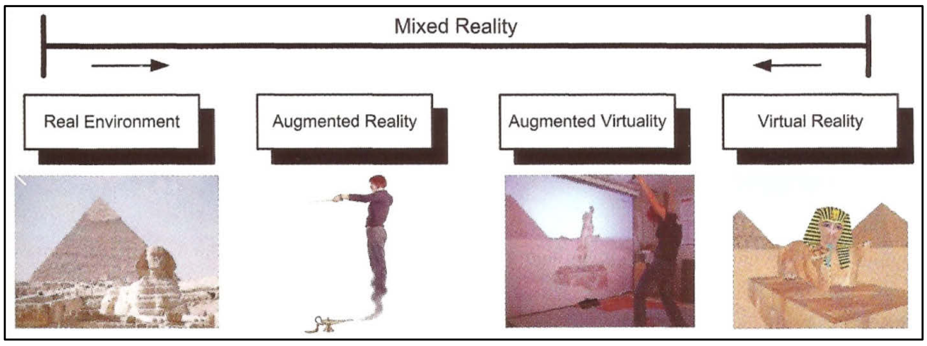


Figure 2-5: Reality-virtuality continuum (Gutiérrez, Vexo, & Thalmann, 2008)

On the one end of the continuum, the real environment is positioned which does not contain computer-generated interfaces. On the other end, virtual reality as fully immersive environment is situated. In between, mixed realities occur which combine reality with virtuality. Augmented reality contains more reality than virtual images referring to examples where new furniture can be placed into a room to see if they fit (Gutiérrez, Vexo,

& Thalmann, 2008). Augmented virtuality, however, contains more computer-generated images. Examples can be immersive films capturing the world in 360° from one or more specific viewpoints (Jerald, 2016).

2.2.2 Functionality of the HTC Vive

The HTC Vive is a product that was developed in partnership between HTC Corporation providing the hardware and the video game developer Valve Corporation providing the software knowledge in terms of SteamVR (McCaffrey, 2017). The virtual reality system was released mid-2016 and consists of a head-mounted system (HMD), two wireless controllers and two base stations as seen in Figure 2-6.

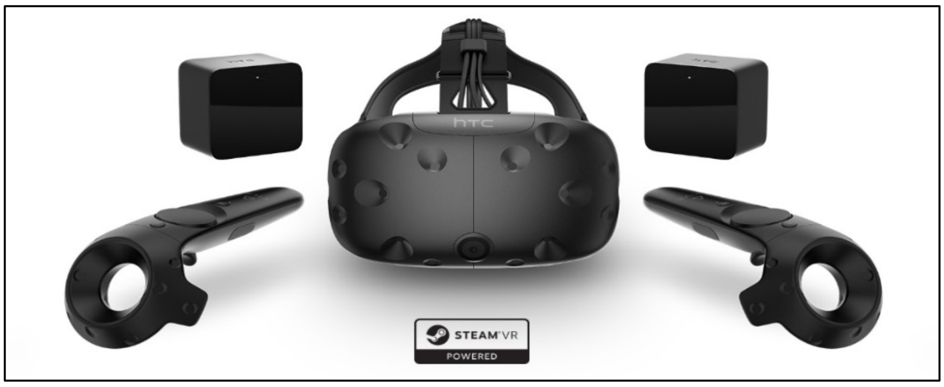


Figure 2-6: HTC Vive with HMD, controllers and base stations (Valve Corporation, 2016)

The headset is the heart of the system and it should provide stereoscopic vision, a large field of view, high resolution and gaze immersion to enable a great immersion for the user. The field of view for two moving human eyes amounts 210°, however, the majority of HMDs just possess a field of view of around 100 degrees (Fuchs, 2017) such as the HTC Vive which provides 110 degrees (HTC Corporation, 2018).

The HTC Vive headset ships with a resolution of 1080 x 1200 pixel for each eye amounting to 2160 x 1200 pixel for both eyes (HTC Corporation, 2018). Most VR system have the same resolution and to receive a relation to the resolution of such systems, the visual acuity of a human eye has to be taken into consideration. To obtain an average visual acuity, a 25 cm wide screen positioned 35 cm away from the eyes must obtain 2500 pixels horizontally for a 40° field of view and 6000 pixels for a 100° field of view which is unfortunately way too high for current technologies (Fuchs, 2017).

The refresh rate of the headset yields 90 Hz (HTC Corporation, 2018) which makes a framerate of 90 frames per second possible. The frame rate is expressed in frames per second (FPS) and is the frequency at which full-resolution rendered images (frames) are being displayed (Jerald, 2016). The time to render a frame is expressed in milliseconds. Critical values for perceiving images as motion are below 20 – 30 FPS (Fuchs, 2017). The headset, furthermore, provides a front facing camera and virtual play area boundaries for safety reasons. Moreover, it consists of several SteamVR tracking sensors, an accelerometer and a gyroscope.

The two motion controllers also consist of the SteamVR tracking sensors and additionally contain input buttons such as the multifunctional trackpad, grip buttons, dual-stage trigger, system button for SteamVR and a menu button (HTC Corporation, 2018).

The tracking configuration is managed through the Lighthouse technology developed by Valve. The two base stations are placed in the corners of the room opposite of each other, thus, eliminating shadowing and covering a 360° play area. Each station exists of stationary infrared LEDs and a pair of active laser which spin either horizontally or vertically with an opening angle of 120° each. After the LEDs have flashed one of the active lasers sweeps a beam across the room. Then the LEDs flash again and the other laser scans the room. This pattern is repeated in a very high frequency. The controller and the headset are covered with photosensors which can receive the light and convert it to digital pulses. The flash light acts as time stamp to indicate when the sensors should start counting time until they are being hit with the laser (refer to Figure 2-7 lower image).

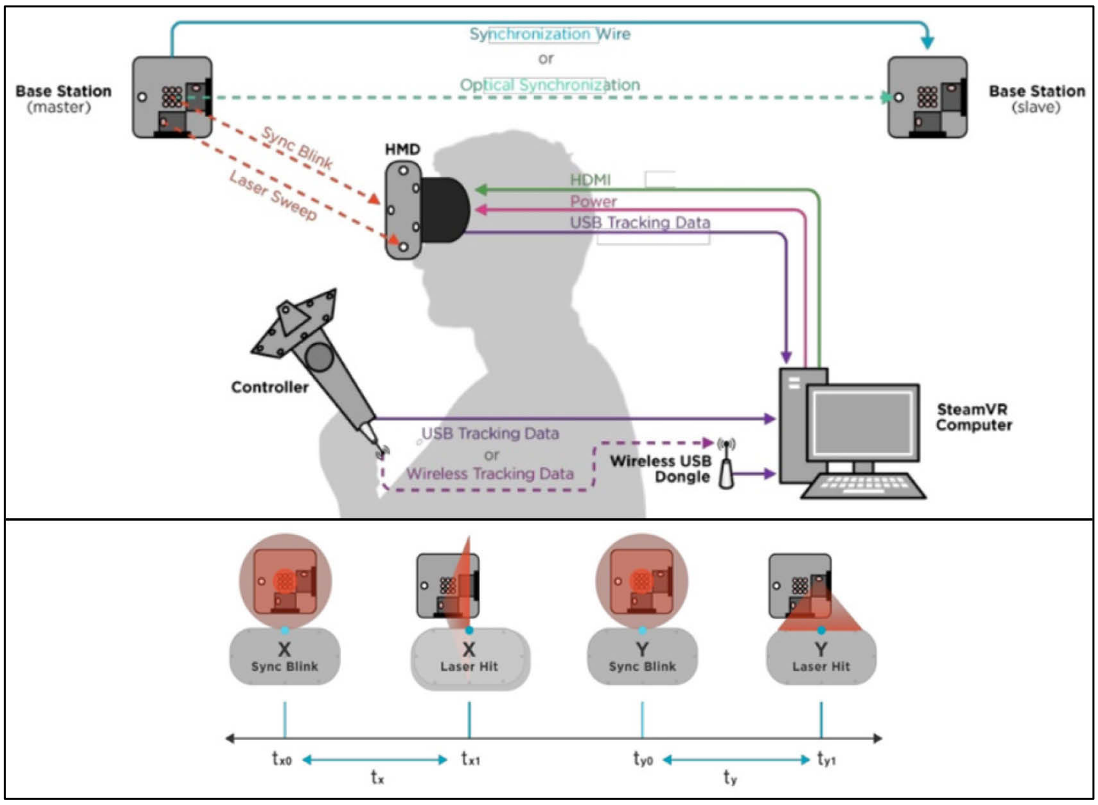


Figure 2-7: Overview of Lighthouse tracking technology for the HTC Vive (upper image) and reference signalling (lower image) (Bruey & Ciuffo, 2017)

The time and the location of the sensor are aggregated and sent to the computer where all sensors and all times are being analysed. Using the SteamVR software, the exact position relative to the base stations of the headset or controller can be thus mathematically determined. If enough sensors are hit, the exact pose and rotation can be furthermore computed (Buckley, 2015). An overview of the technology is given in Figure 2-7.

2.2.3 Basics of Unreal Engine 4

Unreal Engine 4 is one of the world leading game engines developed by the game and software development corporation Epic Games. The game engine offers a catalogue of creation tools used to develop games or other applications for mobiles, personal computers, virtual reality or augmented reality. It ships for free and developers have access to the entire source code. Furthermore, Unreal Engine 4 provides free project templates, sample contents, learning projects and a variety of other resources to enable a quick and easy start. Coding can be done either by using the visual scripting language Blueprints or traditionally by using C++. While C++ grants access to the more hidden features of the engine, Blueprints are way easier to learn and are used throughout this thesis' project (Epic Games, Inc., 2018i).

In the following, only a general overview will be given on the parts that are relevant for this thesis. In light of this, the below-mentioned topics will be briefly covered:

- Blueprints
- Unreal Motion Graphics (UMG)
- Materials and texture mapping
- Texture and world coordinates
- Architecture of landscapes

Blueprints

Blueprints is a way of visually connecting events, functions or variables with wires to create complex functionalities. The two most used types of Blueprints are: Level Blueprints and Class Blueprints. Level Blueprints are global and possess level-wide functionalities meaning that they can also communicate across Class Blueprints. By default, every map or world is equipped with one Level Blueprint. On the other hand, Class Blueprints contain their own variables, components and functionalities and are thus used locally.

Class Blueprints can inherit from several parent classes. The most important is probably the Pawn class which is the base class for all actors that can be possessed. Most commonly, they are the physical representation of the user and rotate and move according to the users input. A Camera component is attached and provides the user's point of view.

The two main events to execute code are Event BeginPlay and Event Tick. The Event Tick function fires every frame and every piece of code which branches off of the event will be executed on every game loop. Thus, this function can be very expensive in terms of performance. The Event BeginPlay function only fires once the corresponding Blueprint starts (Tavakkoli, 2016).

Unreal Motion Graphics (UMG)

To show users large amount of information, typically 2D user interfaces (UI) are being implemented. Unreal Engine therefore provides the Unreal Motion Graphics tool which contains a graphical editor to hierarchically design menus or other text-based features. The tool is retrieved within Widget Blueprints to design information, apply text or add functionalities such as buttons or sliders and so on. Since 2D head-locket UIs provoke a variety of problems in 3D worlds using HMDs, information should be displayed onto 3D objects which are placed within the scene (McCaffrey, 2017).

Materials and texture mapping

In computer graphics, 3D worlds are created out of 3D polygons which are made up of triangles or squares. When arranged to a mesh, they form 3D surfaces. The most basic and efficient polygon, however, is the triangle specified by three vertices that compose it. Graphic hardware takes these triangles and draws them coloured to the screen at very high speeds. The process of generating 2D projections of 3D shapes and assigning colour, lights, shadows or reflections to the triangle is called rendering (Gutiérrez, Vexo, & Thalmann, 2008).

To colour 3D shapes or landscapes in Unreal Engine 4, the Material Editor is needed. This tool drives visual elements which can be basically applied as paint to surfaces. UE4 offers a wide range of possibilities to adjust next to the colour also the shininess, roughness, opacity etc. of a material in order to create characteristics such as a mirror, a wooden floor, metallic structure and much more. A material is therefore the combination of different visual settings and possibly one or more textures. A texture is defined as a two-dimensional image which can be mapped onto a 3D shape. This process is also called texture mapping.

There are several types of textures to control the look. A diffuse map is the most common type and controls the main colour of a surface. A normal map is an image that contains a 3D vector for each pixel. This vector represents the normal of each pixel and provides a 3D look of the otherwise flat surface. The 3D look is enabled when light rays hit the surface and reflect with different angles related to the normal vectors. Another type is the specular map which shows where the software should highlight colour to define the shininess of a surface. The most common map is the height map which is a black and white image where dark values represent lower elevations and white values represent higher elevation (Tavakkoli, 2016).

Texture and world coordinates

Texture coordinates control how an image gets mapped to a 3D shape. The coordinate system is defined by the UVs where U represents the horizontal location and V the vertical location. The coordinate system runs from 0 to 1 with its origin in the lower left corner of the texture. Each vertex of a 3D shape has values that correspond to the coordinates of the texture and therefore links the correct position of the texture to the 3D shape.

The world coordinate system of Unreal Engine 4 is a cartesian three-dimensional coordinate system (X,Y,Z or R,G,B) where 1 Unreal Engine unit (UU) relates to 1 cm (Tavakkoli, 2016).

Architecture of landscapes

Landscapes in Unreal Engine 4 consist of a Landscape Actor which holds the entire landscape as seen in Figure 2-8. Each landscape furthermore is made up of several squared components which are the basis for rendering, visibility calculation and collision. The size of the components is defined upon landscape creation. The vertices where the components meet, are being duplicated.

Each component, in turn, consists of either 1 or 4 (2 x 2) sections which are the basis for the LOD (Level of Detail) calculation. The size must be a power of two (max. 256 x 256) in order to store LODs in mipmaps of the texture. The smallest elements of the landscape are the Quads which are squared as well and are specified by 4 vertices that compose it (Epic Games, Inc., 2018h).

Landscapes from external sources can be imported into UE4 by using one of the two supported formats: 16-bit grayscale *.PNG or 16-bit *.RAW format in little-endian byte order (Epic Games, Inc., 2018j).

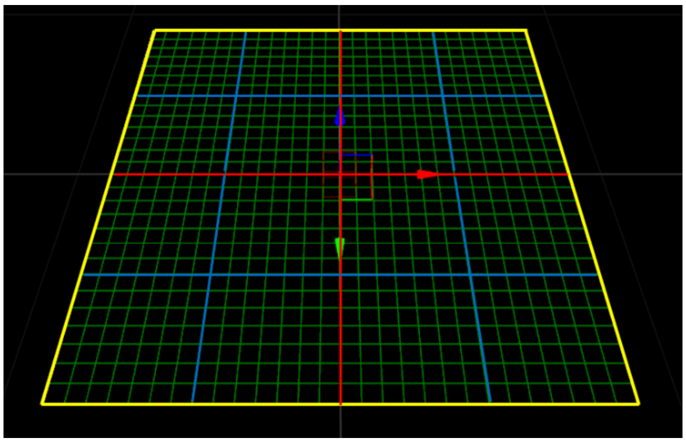


Figure 2-8: Landscape architecture in Unreal Engine 4. The landscape consists of a Landscape Actor (yellow), four Landscape Components (red) with each four Landscape Sections (blue) and individual Quads (green) (Epic Games, Inc., 2018h)

3 Data Acquisition and Processing

This chapter covers the acquisition and processing of hydrographic and terrestrial datasets. It lists all used elevation models and outlines the acquisition of hydrographic data in detail considering deployed instruments, parameters and settings. Furthermore, the processing of data and the combination of both terrestrial and hydrographic datasets is depicted. In the end a final product will be obtained which is used for the development of the virtual reality application delineated in chapter 4.

3.1 Materials

The bathymetry and backscatter data were collected from 11th until 19th of August 2017 during the expedition MSM66 conducted with the vessel Maria S. Merian (see section 1.2). The acquisition system consists of the Kongsberg EM122 MBES in combination with the positioning system Kongsberg Seapth320 and the CTD probe SBE911plus as further described in the subsection 3.2.1. To be specific, MBES lines "0422_[..]" until "0601_[..]" were used in this thesis.

For the generation of the terrestrial and residual bathymetric terrain, three DEMs were utilised as listed in Table 3-1. The fourth dataset – RGI – was implemented to allocate glaciers to the terrain texture.

Dataset	Resolution	EPSG	Download links
Arctic DEM	2 m	3413	http://data.pgc.umn.edu/elev/dem/setsm/ArcticDEM/geocell
	5 m	3413	http://data.pgc.umn.edu/elev/dem/setsm/ArcticDEM/mosaic
CDEM	~ 20 m	4269	http://maps.canada.ca/czs/index-en.html
IBCAO	500 m	3996	https://www.ngdc.noaa.gov/mgg/bathymetry/arctic/grids/version3_0
RGI	varying	4326	https://www.glims.org/RGI/rgi60_files/04_rgi60_ArcticCanadaSouth.zip

Table 3-1: Overview of used datasets concerning their resolution, coordinate reference system (EPSG) and download link

The ArcticDEM is an open data product distributed by the research facility Polar Geospatial Center (PGC) at the University of Minnesota initiated by the US-American National Geospatial-Intelligence Agency (NGA) and National Science Foundation (NSF). The aim of the initiative is to generate high quality elevation data in remote locations covering all regions beyond 60° N. The stereoscopic imagery is acquired by the vendor Digital Globe, Inc. using panchromatic bands of the optical satellites WorldView-1, WorldView-2, WorldView-3 and GeoEye-1. The data is pre-processed by applying auto-correlation techniques to overlapping pairs of images. ArcticDEM can be obtained in 2 m or 5 m resolution and yields both a vertical and horizontal accuracy of 4 m. The data is referenced to the WGS84 ellipsoid and is projected to the NSIDC Sea Ice Polar Stereographic North coordinate system (EPSG:3413) (Polar Geospatial Center & Regents of the University of Minnesota, 2017).

The Canadian Digital Elevation Model (CDEM) is derived from the outdated Canadian Digital Elevation Data (CDED). The data is acquired by the altimetry system of the Ministry of Natural Resources Canada and elements of the National Topographic Data Base (NTDB). The CDEM is related to the North American Datum 1983 and elevations are referenced to mean sea level. All values representing water surfaces such as lakes, oceans, estuaries, are assigned an elevation value of zero. The measured altimetric accuracy ranges from 5 – 15 m and the spatial resolution yields 0.75 arc seconds (Government of Canada, Natural Resources Canada, 2016).

The International Bathymetric Chart of the Arctic Ocean (IBCAO) was initiated in 1997 and conducted so far by 24 institutions in 10 countries to establish a data base that contains all bathymetric data beyond 64° N. IBCAO is furthermore supported by the Intergovernmental Oceanographic Commission (IOC), the International Arctic Science Committee (IASC), the International Hydrographic Organization (IHO), the General Bathymetric Chart of the Oceans (GEBCO), and the US National Geophysical Data Center (NGDC) (now the National Centers for Environmental Information (NCEI)). The DEM consists of data from various research ships and fishing vessels and thus provides a varying quality. The most recent version (3.0) contains a DEM with a resolution of 500 m. The data is related to the WGS84 ellipsoid and projected to the IBCAO Polar Stereographic coordinate system (EPSG: 3996) (Jakobsson, et al., 2012).

Unlike the above-mentioned datasets, the Randolph Glacier Inventory (RGI) is a vector dataset distributing the outlines of worldwide glaciers. It is a collection of worldwide available outlines from various databases such as GLIMS (Global Land Ice Measurements from Space), DCW (Digital Chart of the World), WGI (World Glacier Inventory) and many other sources. The production was initiated by the Fifth Assessment Report of the Intergovernmental Panel on Climate Change (IPCC AR5) (RGI Consortium, 2017).

3.2 Hydrographic Data Acquisition and Processing

In order to achieve optimum results, it is necessary to acquire data as precise as possible. Therefore, most accurate and reliable sensors such as multibeam, GNSS and MRU sensors should be deployed. Furthermore, all sensors need to be aligned in vessel coordinate system as good as possible to minimize geometric offset errors. Additionally, the processing of the acquired data has to be performed thoroughly to provide, eventually, datasets that represent the ocean floor in the best possible way.

Regarding the next sections, all relevant instruments will be introduced in terms of their specifications and set parameters during acquisition. Moreover, the processing of multibeam bathymetry and backscatter data including their final products will be outlined. In the end, the data will be examined for their accuracy and quality.

3.2.1 Hydrographic Data Acquisition

For the data acquisition, the Kongsberg software SIS (Seafloor Information System) was used. It is a real time software and constitutes an interface between echosounders, positioning systems and attitude sensors (Kongsberg Maritime AS, 2013). A summary of key information regarding the hydrographic acquisition is presented in Table 3-2.

Key aspects of the hydrographic survey	
MBES aperture angle	130°
MBES beam spacing	Equidistant
Mean survey speed	9 knots
Total hours of surveying	163 hours
Total cruise track length	2760 km
Total number of CTDs	11
Depth range	-20 to -714 m

Table 3-2: Summary of hydrographic survey

Multibeam echosounder Kongsberg Simrad EM 122

The multibeam echosounder EM 122 is used for seabed mapping and imagery to full ocean depth ranging from 20 m until 11000 m. It logs both multibeam bathymetry and backscatter data at the same time. Both data types are co-registered which means that they are geographically referenced together so that backscatter snippets will always be located at the correct place on the seafloor. The nominal sonar frequency is 12 kHz. With an aperture angle of 150° at most, the swath width will be six times the water depth which amounts to a swath coverage of more than 30 km for deep, flat ocean floors (Kongsberg Maritime AS, 2011). For this survey an aperture angle of 130° was chosen to get the best quality-coverage ratio. Moreover, the spacing between each adjacent track line was chosen to ensure a 30% swath coverage. In water depths below 100 m a swath coverage of 100% was chosen for safety reasons.

The linear transducer arrays for transmission and receiving are separate units in a Mills cross configuration with beamwidhts of 2° each. The transmission fan is furthermore split in several sectors with independent active beam steering related to the vessels roll, pitch and yaw movement. All soundings will be fitted to the best possible line perpendicular to the survey line ensuring uniform sampling and 100% coverage. To code the different transmit sectors, the frequencies range from 10.5 to 13.5 kHz (Kongsberg Maritime AS, 2011). The EM 122 receives additionally data from the Motion Reference Unit (MRU) for roll, pitch, yaw and heave and is thus as far as possible insensitive against ship movement (Briese Schiffahrts GmbH & Co. Kg., 2017).

The system has up to 432 soundings per ping, with a beam spacing of either equidistant or equiangle. Enabling dual swath mode, two swaths per ping cycle are generated to achieve a maximum of 864 soundings per ping. During this mode the transmit fan is being duplicated and transmitted with a small difference in along-track tilt to ensure a similar sounding density both along-track and across-track. Regarding the acquisition on cruise MSM66, the dual swath mode was set to dynamic with an equidistant beam spacing to assure a uniform sampling of the seafloor in across- and along-track direction. The ping rate depends on the water depth and the ranges of the outer beams. Thus, it was set to automatic. The maximum ping rate is 5 Hz (Kongsberg Maritime AS, 2011).

The total system root mean square (RMS) accuracy is specified by the manufacturer as seen in Table 3-3 assuming accurately aligned transducers and sensors such as positioning, vessel motion and sound speed systems (Kongsberg Maritime AS, 2011).

Technical data	
Frequency	12 kHz
Depth range	50 – 11000 m
Max. coverage	150° (6x water depth)
Beam width	2° x 2°
Beam spacing	Equidistant or Equiangle
System RMS accuracy	<0.2 % of depth (from 0°-45° re the vertical)
	<0.3 % of depth (between 45°-60°)
	<0.6 % of depth (between 60°-70°)
Max. number of soundings/ping	864
Effective pulse length	2 – 100 ms
Max. ping rate	5 Hz
Range sampling rate	3 kHz (25 cm)

Table 3-3: Summary of technical specifications of Kongsberg Simrad EM 122 (Kongsberg Maritime AS, 2011)

Positioning system

All echosounders on board the Maria S. Merian receive data from the Kongsberg Seapath 320 system. This integrated navigation system combines both satellite and inertial navigation data (Briese Schiffahrts GmbH & Co. Kg., 2017). The satellite positioning system includes two fixed baseline GNSS antennas which are able to receive both GPS and GLONASS satellite data. Additionally, the Seapath 320 system receives differential correction support from the OmniSTAR satellite-based augmentation system (SBAS) through the Trimble GPS SPS855 device. The overall positioning accuracy therefore ranges between 0.7 – 1.5 m (Briese Schiffahrts GmbH & Co. Kg., 2017).

Sound velocity profiles

To acquire sound velocity profiles during the survey, the CTD probe SBE911plus from the company Sea-Bird Scientific was used. The probe on board was mounted on a sampling rosette equipped with further sensors such as a biospherical PAR (Photosynthetically Active Radiation) light sensor, a fluorometer, an acoustic ground sensor to improve the handling of lowering the rosette and several water tubes to collect water samples on the way (Briese Schiffahrts GmbH & Co. Kg., 2017). Details of the technical specifications of the CTD-probe are listed in Table 3-4.

Technical data		
Sampling speed	24 Hz (24 samples/sec)	
Max. sampling depth	6800 m	
Measurement range	Conductivity	0 to 7 S/m
	Temperature	-5 to +35°C
	Pressure	0 to full scale (=10000 psia)
Initial accuracy	Conductivity	± 0.0003 S/m (Siemens/meter)
	Temperature	± 0.001 °C
	Pressure	± 0.015% of full scale range

Table 3-4: Technical data of Seabird SBE911plus CTD probe (Sea-Bird Scientific, 2017)

For the investigated area of Clyde Inlet, eleven sound velocity profiles were taken as seen in Figure 3-1. After each sampling, the sound velocity profile was inserted in the acquisition software SIS and applied to the following track lines until the next sound velocity profile was conducted. Figure 3-1 shows that CTD-stations were rather homogeneously distributed throughout the survey area, however, a large part of the survey was executed with only one sound velocity profile (41-1).

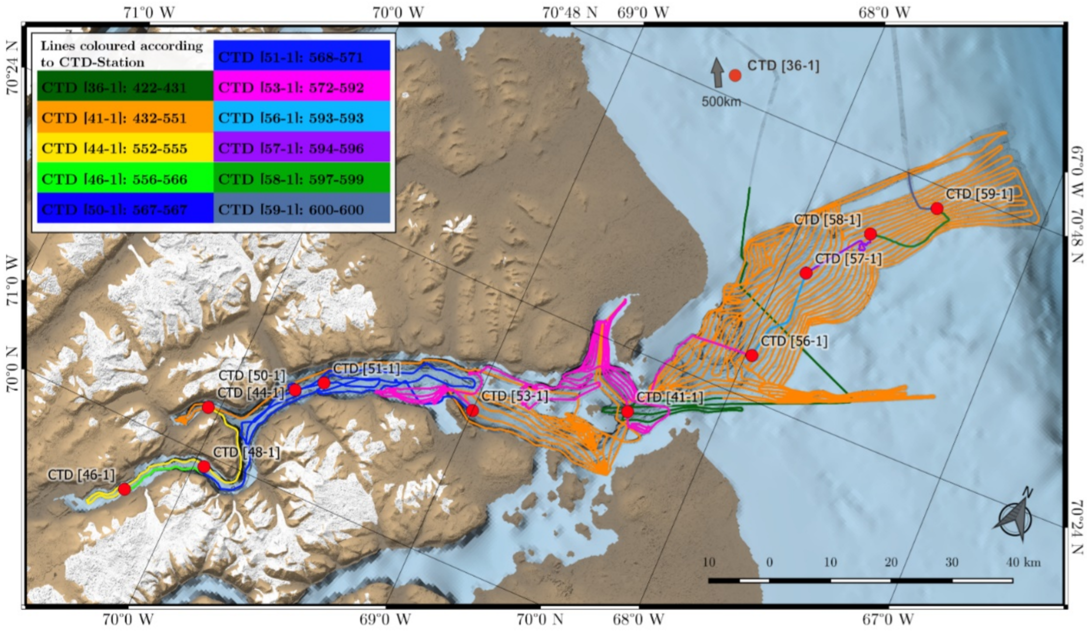


Figure 3-1: Track lines coloured according to the station number where sound velocity profiles were sampled and applied to the acquisition. [CTD-station number: line number] (created in QGIS)

3.2.2 Processing of Multibeam Bathymetry

The acquired multibeam bathymetry data was processed using the software CARIS HIPS and SIPS 10.3. Since bathymetric data is prone to various systematic and random errors as stated in section 2.1.1, it is the hydrographers task to eliminate most errors and to distinguish outliers and artefacts from true seafloor features. Most software packages including CARIS HIPS and SIPS provide two possibilities for data cleaning: either manually or by using automated methods. One of the most famous automated method is the CUBE algorithm. However, it was not possible to create the required TPU (Total Propagated Uncertainty) due to missing parameters and documentation of the vessel. Thus, the bathymetric data was cleaned manually using the processing steps as listed below in Figure 3-2.

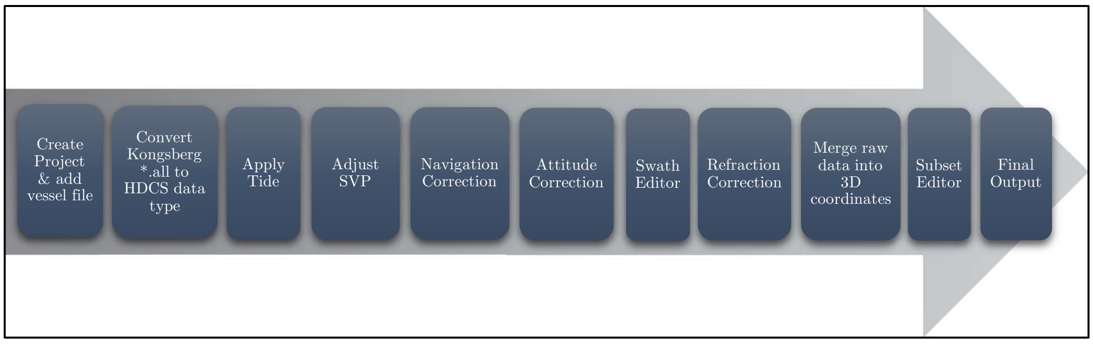


Figure 3-2: Diagram of processing multibeam bathymetry data in CARIS HIPS and SIPS

At first the project was created, and the vessel configuration file imported. The vessel file contains information about the sensor alignment in the local vessel coordinate system and is needed when merging raw data into 3D coordinates. The information was retrieved from the alignment report "Parkerreport" executed by the company Parker Maritime AS in April 2017.

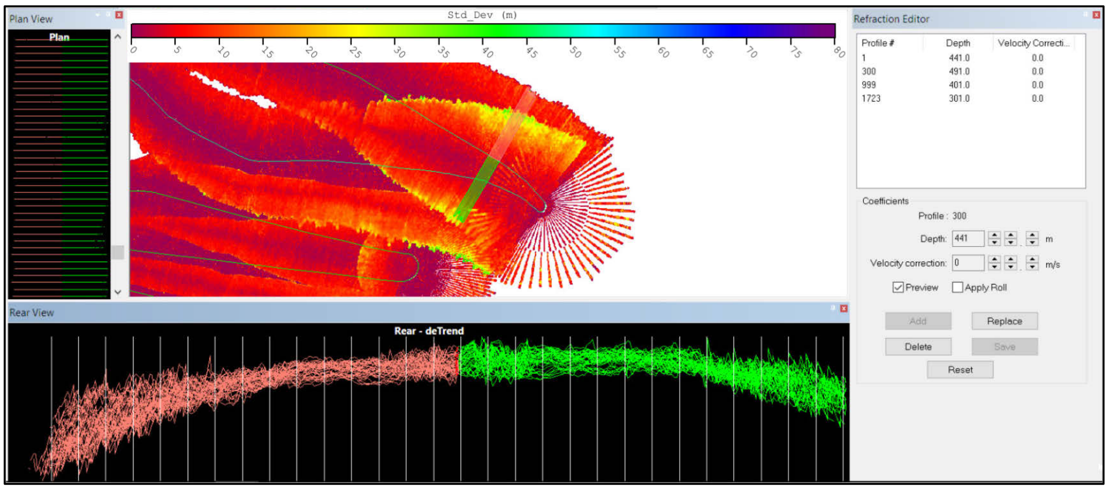
Then, the Kongsberg multibeam *.all raw data format had to be converted into the CARIS HDCS data structure which separates the original file into number of distinct files each storing different types of information such as SVP, gyro, GPS or observed depths information and so forth (CARIS, 2013). After the successful import, tide data was applied. Due to the fact that RTK positioning was not available in high latitudes, tide data generated through GNSS could not be used. Therefore, the predicted tidal model OTPS (OSU Tidal Prediction Software) was applied instead. The OTPS-tide is a globally predicted barotropic tide model developed through the Oregon State University (OSU) (Egbert & Erofeeva, 2010).

Sound velocity profiles were already applied during acquisition whenever a new profile was conducted. However, they were applied only for the next track line and not for the track prior the CTD sampling. Consequently, tracks just prior the CTD sampling might have been recorded with a sound velocity profile sampled far away in time or distance. This can be seen in Figure 3-1 for lines 422-431 which were recorded with a SVP sampled 500 km off the survey area. Comparing both sound velocity profiles at CTD station 36-1 and 41-1, it can be said that the velocity at station 36-1 is generally higher with a mean value of 6 m/s. This might be due to higher salinity or temperature values at station 36-1. To minimize resulting refraction errors in the data, the SVP at station 41-1 was applied retroactively to lines 422-431.

Moreover, sound velocity profiles from CTD station 41-1 and 44-1 were compared with each other since track lines within the northern branch of Clyde Inlet (Kimmiaqtaqtujuq) were recorded with a SVP which was sampled 5 days ago and in a distance of 70 km. The mean value of the difference between both stations amounts to 1.1 m/s. High differences of up to 11 m/s were found close to the water surface. Such fluctuations are not unusual in surface layers due to influences by wind, sun, and the melting of icebergs or glaciers. However, as this value is rather small it can be neglected, and no additional SVP corrections were applied throughout the survey.

To detect navigation errors, the whole survey data was examined for abrupt changes in direction and vessel speed. Only occasionally small GNSS jumps were detected and deleted. The positional gap was linearly interpolated using adjacent measurements. The Attitude Editor was used to validate all motion data of the vessel during the survey. No errors were found and the whole data looks plausible. By using the Swath Editor, all soundings of each track line were observed, and coarse artefacts were rejected.

Furthermore, data recorded along the shelf edge had to be corrected for refraction errors due to missing sound velocity values. The applied sound velocity profile (41-1) contains values only to a depth of 419 m. The depth along the shelf edge, however, ranges from 400 m until 714 m. Refraction errors are most distinctive in outer beams of the swath which tends to bend upwards or downwards the higher or lower the applied sound velocity is with respect to the true velocity. Figure 3-3 shows the swath curvature in rear view and the surface colouring according to the standard deviation. It can be seen that the standard deviation is much higher in the slant ranged beams where the curvature is most distinctive.



Profile #	Depth	Velocity Correcti...
1	441.0	0.0
300	491.0	0.0
999	401.0	0.0
1723	301.0	0.0

Coefficients

Profile : 300

Depth: 441 m

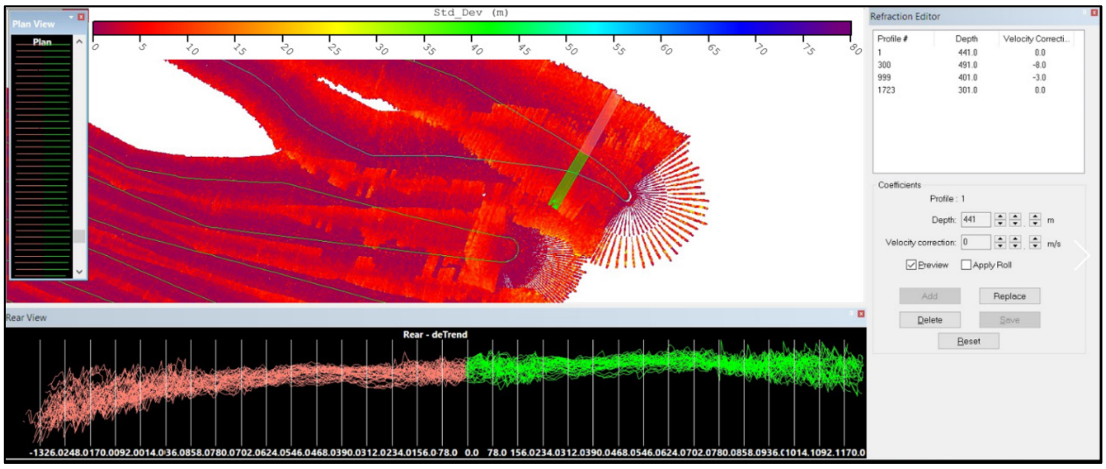
Velocity correction: 0 m/s

[✓ Preview] Apply Roll

Figure 3-3: Seabed before the refraction correction in CARIS HIPS and SIPS. Outer beams are curved downwards (bottom window) and yield a much higher standard deviation (upper window).

Through the Refraction Editor, the SVP can be altered manually for the required line segments in order to achieve a better refraction solution. This was done for each line section crossing the shelf edge. CARIS HIPS and SIPS interpolates linearly between inserted velocity corrections in order to prevent sound velocity jumps within a line (CARIS, 2013). In Figure 3-4 the corrected seafloor after the adjustment can be seen. Furthermore, it is noticeable that the standard deviation is much lower in the outer beams. The corrected values range from approx. -4 m/s for a depth of 450 m to approx. -10 m/s for a depth of 700 m.

After completing all line-based corrections and data cleaning operations, all raw data had to be merged into 3D coordinates. The merging process in CARIS HIPS and SIPS converts along- and across-track depths into XYZ-coordinates by combining information such as navigation, sensor alignment, tide, SVP, refraction correction, observed depths and so on (CARIS, 2013).

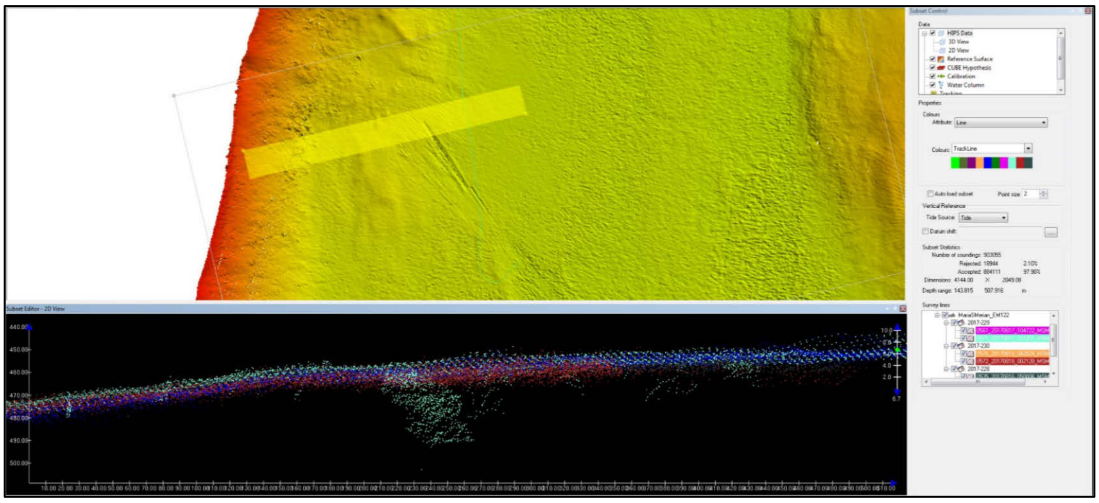


Profile #	Depth	Velocity Correcti...
1	441.0	0.0
300	491.0	-8.0
999	401.0	-3.0
1723	301.0	0.0

Coefficients
Profile : 1
Depth: 441 m
Velocity correction: 0 m/s
[✓ Preview] Apply Roll

Figure 3-4: Seabed after the refraction correction in CARIS HIPS and SIPS. Outer beams are corrected (bottom window) resulting in an improved standard deviation (upper window).

A final validation has been done by using the Subset Editor. This editor is area-based and is used to delete outliers of processed data. In Figure 3-5 artefacts caused through the turning manoeuvre of the vessel can be seen. The yellow box envelopes the area which is visible in the bottom of the figure. This area shows all depth values coloured in the respective track lines. The light blue depth values are clearly visible as outliers and were therefore rejected. In the end, the whole survey was successfully processed and cleaned, ready to be computed to a final digital elevation model as described in the following section.



Properties
Colours
Attribute: Line
Colours: TrackLine
Auto load subset Point size: 2
Vertical Reference
Tide Source: Tide
Datum shift:
Subset Statistics
Number of soundings: 903055
Rejected: 18944 2.10%
Accepted: 884111 97.90%
Dimensions: 4144.00 X 2049.00
Depth range: 143.815 507.916 m

Figure 3-5: Removing outliers with the Subset Editor in CARIS HIPS and SIPS

3.2.3 Bathymetric Surface Modelling

To visualise bathymetric data, a digital elevation model (DEM) was computed in CARIS HIPS and SIPS. A digital elevation model is a three-dimensional representation of the Earth's topography. It is computed through interpolation or extrapolation of sections where no sampled elevation data exists. Two types of DEMs can be distinguished: irregular vector-based models and regular raster-based models.

Vector models consists of vertices which are connected by edges to form a triangular network. They are thus called triangulated irregular network (TIN) and are mainly used for discrete and irregular distributed points (Zhou, Lees, & Tang, 2008). On the other hand, raster models store elevation data in a regular pattern of squared cells. Such models are therefore also known as regular or gridded DEMs. The size of the cells remains constant and defines the geometric resolution, whereas the number of rows and columns defines the spatial extent. Raster models are best used for depicting continuous data such as precipitation or surface elevations (Schwartz, 2005). The digital elevation model derived in this thesis is therefore based on the raster model.

In CARIS HIPS and SIPS, the gridded surface was generated by combining a swath angle weighting and an inverse distance weighting algorithm. The first method of the two-fold computation uses a weighted function based on a beam's incidence angle. Inner beams are given higher weight and contribute more to the calculation of the surface than outer beams. This function is particularly important in areas with adjacent or overlapping track lines. In Figure 3-6 (left) the applied swath angle weight diagram can be seen. Beams with an intersection angle of 90° - 75° with the seafloor receive a weight of 1.0. The weight then decreases linearly (CARIS, 2013).

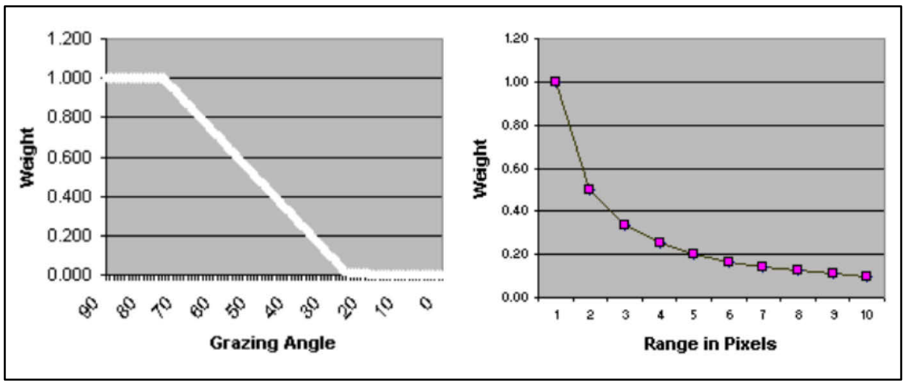


Figure 3-6: Weighting schemes of surface interpolation in CARIS HIPS and SIPS; Left: Incidence angle weighting function, Right: IDW interpolation function (CARIS, 2013)

The other weighting algorithm is based on an inverse distance approach, also called IDW (Inverse Distance Weighting). The weight a sounding receives is inverse proportional to the distance of the interpolated cell, meaning that closer soundings will obtain higher weights than soundings which are further away. The number of soundings which are weighted for the interpolation of a cell, is defined by the area of influence (CARIS, 2013). The higher the area of influence is, the higher is the smoothing of the surface. For the DEM generated in this thesis, the size was set to 9 x 9 pixels to retain relevant landforms but also to keep the storage capacity as low as possible. In comparison with a 5 x 5 pixel size influence area, no major differences could be observed. The storage size, however, was much higher. In Figure 3-6 (right) the range weighting scheme can be seen.

Additionally, the resolution of the surface must be defined. Therefore, the data density and the accuracy of the multibeam system was taken into consideration. The accuracy of the multibeam system amounts to 1.7 m as explained in section 3.2.5. The data density must be evaluated both in along-track and across-track direction. The along-track data density depends on the vessel speed, depth and ping rate and was observed to be 1 – 3 m in shallow depths of 100 – 200 m and 4 – 7 m in depths of 300 – 700 m. The across-track data density depends on the aperture angle, number of beams, depth and the beam spacing (equidistant or equiangle). The density ranges from 1 – 2 m in shallow depths to 3 – 6 m in deeper depths. The final resolution of the surface was chosen to the highest possible resolution of 5 m due to the high accuracy of the multibeam system and the fact that most areas within the surface have a density higher than 5 m.

In the end, the digital elevation model was computed as seen in Figure 3-7 and could be exported as GeoTIFF for further work.

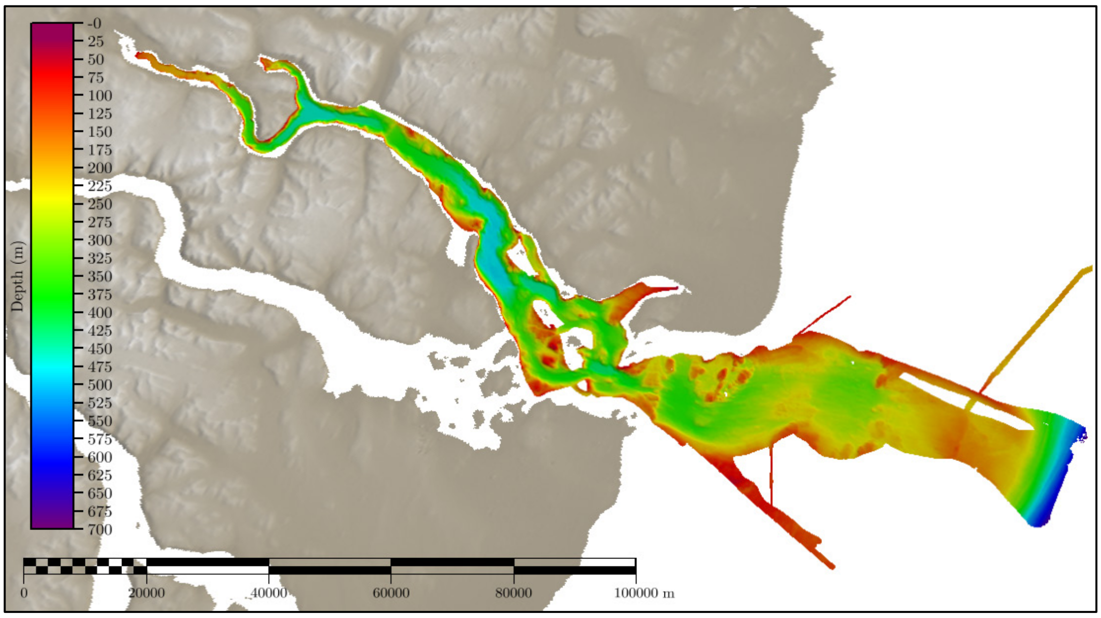


Figure 3-7: DEM of Clyde Inlet in CARIS HIPS and SIPS with ASTER background data

3.2.4 Backscatter Analysis

To acquire information about the sediment arrangement and its variability, multibeam backscatter data was processed using the FM Geocoder Toolbox (FMGT) from the software package Fledermaus version 7.8.0. The dataset was split into five areas of same sizes to account for high computing demands. The following parameter adjustments and processing steps were adopted for every area. Afterwards all areas were merged in QGIS.

At first, Kongsberg *.raw data files were imported into FMGT. Backscatter data was then extracted and mapped into mosaic cells. While mosaicking, several corrections and adjustments were applied. They include radiometric corrections based on sonar type and bottom topography, geometric corrections, angle varying gain (AVG) adjustments and anti-aliasing to allow the assemblage of mosaic cells at any required resolution.

The AVG was applied over a window of 300 pings using the adaptive algorithm. The adaptive algorithm is a combination of two other available methods, namely flat and trend. The flat method smooths out small variations in backscatter level whereas the trend algorithm searches for a trend in the signal to smooth around the trend line. The adaptive method is a combination of both and preferred for terrains with varying slopes.

To control the mosaic generation, two blending algorithms were applied for overlapping samples. If two lines fall into the same cell, both are blended equally by 50% each. However, samples of near nadir sections were blended using a weighting function. Outer beams will be prioritized if their distance from nadir is greater than 25% of the half swath distance. This option cleans most of the nadir artefacts and is most effective for survey lines with great overlap (QPS, 2018).

Subsequently, backscatter data was computed and mapped into mosaic cells with a resolution of 20 m (see Figure 3-8). This resolution was chosen due to the fact that imported images in UE4 are restricted to a maximum of 8192 pixels. The width of the study area (161285 m) divided by the maximum number of pixels (8192) yields 20 m for each pixel.

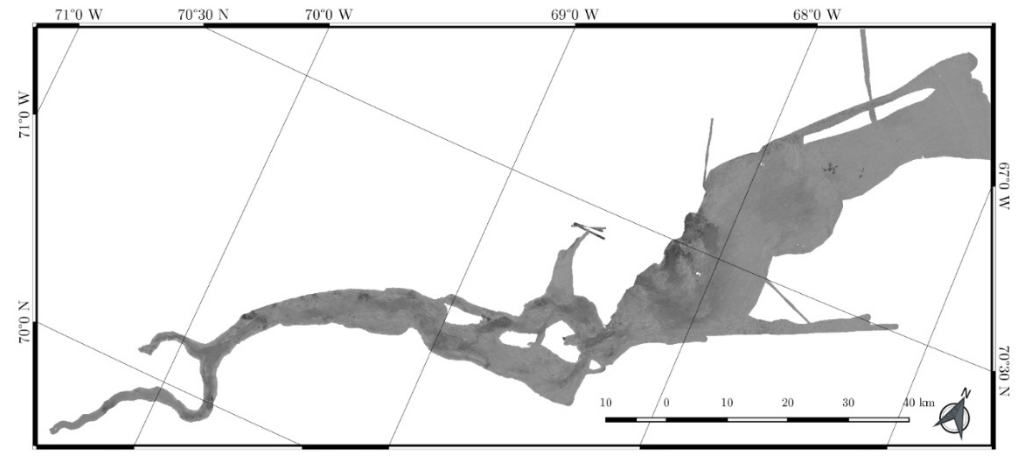


Figure 3-8: Backscatter mosaic of Clyde Inlet (created in QGIS)

After the mosaic generation, the Angular Range Analysis (ARA) could be computed. ARA is a method for seafloor characterization and is based on the premise that each sediment has a unique angular response. ARA compares the surveyed backscatter response to an expected response based on the Jackson model. The Jackson model is a very complex model which computes the acoustic response curve as a function of incidence angle vs. returned backscatter intensity. It is thus very sensitive as it depends on many parameters such as the acoustic frequency, sediment properties and topography. When the ARA matches a curve in the best possible way, it indicates which sediment properties the seabed might have (QPS, 2018). In Figure 3-9 (left) the modelled curve (blue) is fit to the reflected backscatter signals implying that the seabed might be clayey sand on the port side and gravelly coarse sand on the starboard side.

The results of the ARA are furthermore stored in patches for port and starboard side as seen in Figure 3-9 (right). Each patch contains 30 consecutive pings being averaged for classification (QPS, 2018). The characterization, thus, appears in a rectangular pattern.

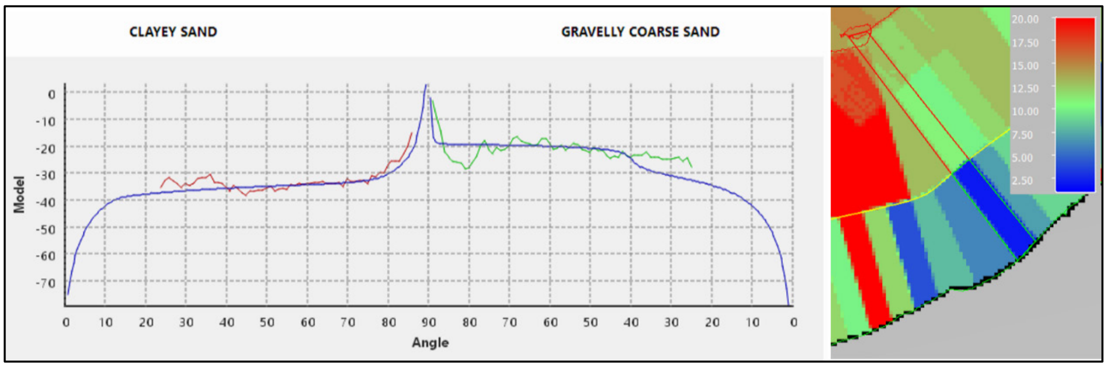


Figure 3-9: ARA result of one patch for port and starboard side in FMGT. The modelled curve (blue) is fit to the received angular responses (red, green) indicating the sediment type

In Figure 3-10 the ARA result for the whole terrain is depicted. ARA defines 20 groups of sediment types ranging from gravel to sand, silt and clay (QPS, 2018). It can be seen that most of the terrain was classified as clay and sand whereas close to the steep sided flanks sediment was classified as gravel.

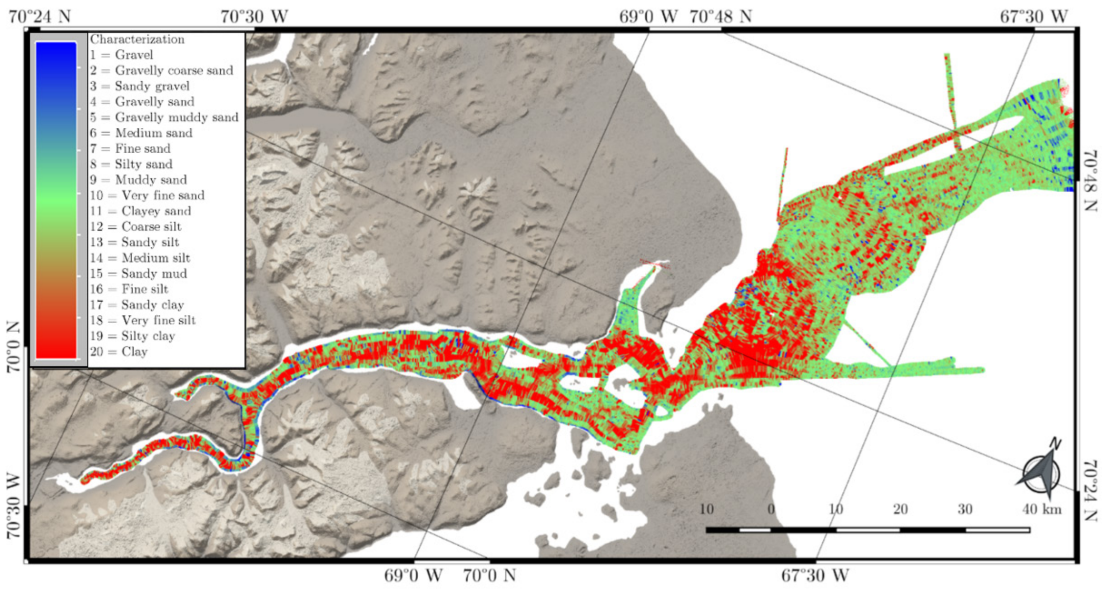


Figure 3-10: ARA result of Clyde Inlet (created in QGIS using ASTER background data)

3.2.5 Quality Considerations

Bathymetry

The accuracy of the multibeam system EM122 depends on the water depth and aperture angle as listed in Table 3-5. The greater the depth and aperture angle, the worse is the accuracy. Regarding the average water depth of 280 m, the accuracy for nadir beams amounts to 0.56 m, for middle beams to 0.8 m and for outer beams to 1.7 m. However, the total accuracy of a measured point is hardly accessible due to missing specifications of deployed instruments, latency and various systematic and random errors. Taking at least the accuracy of the navigation devices and the multibeam system into account, a measured point would have, at best, an accuracy of 1.2 m in mean water depth.

Accuracy / Depths	**0.2% of depth (from 0°-45°)**	**0.3% of depth (between 45°-60°)**	**0.6% of depth (between 60°-70°)**
50 m	0.1 m	0.2 m	0.3 m
280 m (mean)	0.6 m	0.8 m	1.7 m
700 m	1.4 m	2.1 m	4.2 m

Table 3-5: Accuracy assessment of multibeam system EM122 under given specifications (Kongsberg Maritime AS, 2011)

With regards to the precision of the data, the mean standard deviation yields 0.9 m and has thus the same magnitude as the accuracy. The standard deviation represents the distribution of soundings from the mean. If both the accuracy and precision amounts to approx. 1 m in average water depth, it can be stated that seafloor features smaller than this value would not be visible in the data. They would be within the measurement noise. Figure 3-11 shows the standard deviation of the surface.

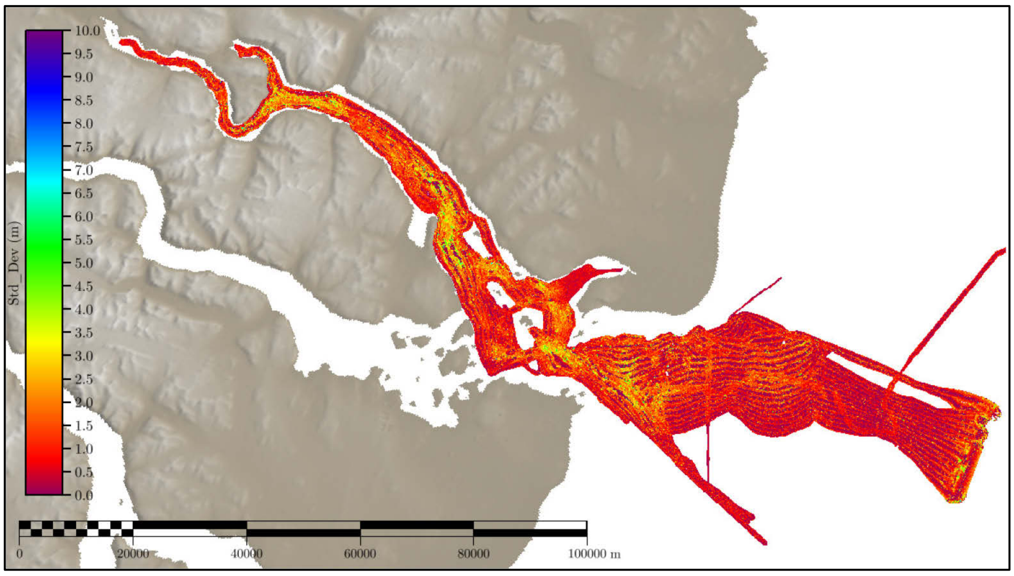


Figure 3-11: Standard deviation of surface in CARIS HIPS and SIPS with ASTER background data

The spatial resolution of data points can be calculated in along and across-track direction of the vessel. The across-track resolution depends on the water depth and the beam width. Taking the mean water depth of 280 m and an overall aperture angle of 130° into consideration, the total swath coverage yields 1201 m as listed in Table 3-6. With a beam width of 2°, the footprint in the nadir beam amounts to 9.8 m. Dividing the total swath coverage by 432 beams, a rough estimate for the distance between the centre points of the beams on a flat ocean floor is achieved. The beam spacing in equidistant survey mode amounts to 2.8 m and is thus one hundredth of the depth. Consequently, each midpoint of a beam will be covered at least three times.

Depth	**Footprint in nadir**	**Footprint in outer beam**	**Swath coverage**	**Beam spacing (across-track)**	**Ping rate**	**Swath spacing (along-track)**
50 m	1.8 m	9 m	215 m	0.5 m	2 Hz	2.3 m
280 m	9.8 m	51 m	1201 m	2.8 m	0.5 Hz	9.26 m
700 m	24 m	127 m	3002 m	6.9 m	0.3 Hz	14 m

Table 3-6: Spatial resolution

On the other hand, the spatial resolution in along-track direction depends on the vessel speed and the ping rate. The ping rate was set to dynamic (section 3.2.1) and is thus varying with the water depth. For the average water depth, a ping rate of 0.5 Hz could be examined. The resulting beam spacing in along-track could be computed by multiplying the ping rate with the mean vessel speed (see Table 3-2). The distance between two beams in nadir amounts to 9.26 m resulting in an overlap of 11% for the average depth.

In the end, the data was assessed based on the IHO (International Hydrographic Organization) standard S-44. The S-44 standard is imposed by the IHO and particularly used when compiling nautical charts from bathymetric data to provide safety for navigation. The S-44 standard is divided into four orders of accuracy requirements: Special Order, Order 1, Order 2 and Order 3. The first two orders are relevant for harbour surveys in shallow waters. Order 2 applies up to a depth of 200 m and Order 3 is used for offshore areas (International Hydrographic Organization, 2005). The data in this thesis falls under the category of Order 3. Thus, the accuracy requirements are least accurate yielding a minimum depth accuracy of 6 m and a minimum horizontal accuracy of 164 m concerning a depth of 280 m (International Hydrographic Organization, 2005). In CARIS HIPS and SIPS uncertainty values contained in the surface are compared to the S-44 standard. The result shows that 99% of the sampled points are in conformance with S-44. However, it is important to state that all standard specifications must be met in order to be compliant with S-44, meaning that all uncertainties must be known. Due to missing documentation as described before, the TPU could not be computed. Nevertheless, the above made calculation provides a first indication on the quality of the data.

Backscatter

The veracity of the ARA result depends on a variety of factors starting with the system calibration, applied corrections and filters. Furthermore, the sonar frequency plays an important role since inappropriate frequencies might penetrate the seabed resulting in reflected signals from underlying sediment. Variations in bathymetry additionally influences the intensity of the backscattered signal and its effect cannot always be completely removed (Dufek, 2012). Moreover, FMGT averages all angular responses within one patch consisting of 30 pings (QPS, 2018). If the morphology exhibits strong variations within those 30 pings, averaging might lead to incorrect classification results.

To obtain more reasonable ARA results, it is of advantage to ground-truth the data with sediment probes. This can be conducted in combination with a beampattern correction which might lead to more secure results. Both processing steps were not realised since they would fall beyond the scope of this thesis.

In conclusion, it should be stated that only an estimate of the sediment distribution was possible. The computed ARA result yields a first impression of the morphology of Clyde Inlet.

3.3 Terrestrial Data Processing

In this section, the processing of terrestrial data is described. To obtain an adequate terrestrial terrain model, a combination of the ArcticDEM with 2 m resolution, ArcticDEM with 5 m resolution and CDEM with 20 m resolution (see 3.1 for references) was utilised. The ArcticDEMs were chosen due to their uniquely high resolution in arctic polar regions. The CDEM is apart from ASTER GDEM, the only free available DEM with next higher resolution of about 20 m. The CDEM was used over ASTER GDEM, because its data set appears much smoother and not so blocky. Furthermore, the ASTER data shows a lot of outliers at the water surface which would make the processing significantly more difficult. In contrary, the water surface in the CDEM is cleaned and contains only values of zero.

3.3.1 Processing of Digital Elevation Models

The processing was conducted using the application ArcMap from the software package ArcGIS 10.4.1 for Desktop. An overview of the processing workflow is given in Figure 3-12. The complete workflow including bathymetric data sets is shown in appendix Figure 9-1.

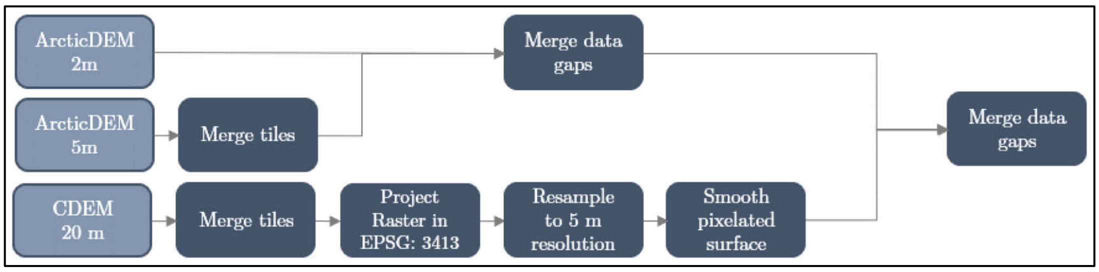


Figure 3-12: Processing workflow of terrestrial DEMs

To ensure a good representation of the terrestrial topography, a digital elevation model with a high spatial resolution should be used. Unfortunately, several memory errors occurred while working with the ArcticDEM(2 m), which is not surprising since a single 100 km x 15 km strip has a size of 1.5 GB. For this reason, the ArcticDEM(5 m) was chosen as the main elevation model. The source data set consist of nine tiles covering the investigated area. In ArcMap all tiles were merged to one raster. However, several gaps were exhibited within the data. These gaps were partially filled with the ArcticDEM(2 m) using the Mosaic-Operator. All remaining gaps were filled with the CDEM. The CDEM, however, needed to be processed beforehand.

First of all, its tiles had to be merged and the projection transformed to the NSIDC Sea Ice Polar Stereographic North projected coordinate system (EPSG:3413). This coordinate system was chosen, because it depicts the fjord in the most compact way which is essential for the work in Unreal Engine 4. The transformation was performed using the nearest neighbour assignment which will not change cell values unlike the bilinear or cubic interpolation (ESRI, 2018). Simultaneously, the resolution was changed to 5 m to match the ArcticDEM. Moreover, the maintaining pixelated surface was slightly smoothed through the Focal Statistics tool, because spatial aliasing is clearly visible in Unreal Engine 4. Focal Statistics performs a neighbourhood operation and computes in this case the average for the output cell after a predefined input neighbourhood (ESRI, 2016). A circular neighbourhood with a radius of 2 cells was chosen to just minimize the aliasing effect without smoothing the data too much. In Figure 3-13 the difference between the original, pixelated CDEM and the smoothed CDEM is shown.

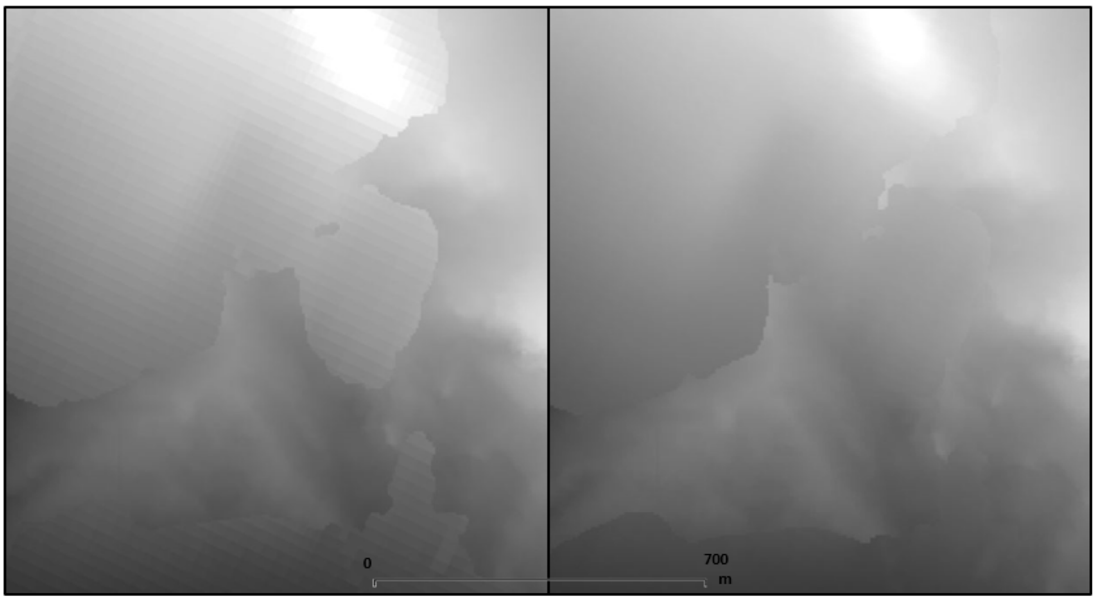


Figure 3-13: Gaps in ArcticDEM are closed with pixelated CDEM (left) and smoothed CDEM (right)

In the end, all three source DEMs were merged to a final terrestrial elevation model of good quality. The processing considering the linkage with bathymetry data will be examined in section 3.4.

3.3.2 Quality Considerations

The accuracy of the terrestrial elevation model varies between 4 m and 15 m as stated in the official papers distributed by the producers of the ArcticDEM and CDEM (refer to section 3.1). The ArcticDEM achieves an astonishing accuracy of 4 m, whereas the accuracy of the CDEM varies between 5 m – 15 m. The difference in accuracy and thus the height difference is in part greatly noticeable. This can be seen in Figure 3-13 where the transitions between CDEM and ArcticDEM are partly strongly visible. Additionally, height differences at five locations were measured yielding differences of up to 11 m (see Table 3-7).

Elevation in [m]	**68°58'23.63''W 70°5'32.50''N**	**70°19'23.18''W 69°49'35.81''N**	**70°37'27.21''W 70°20'2.04''N**	**69°47'53.55''W 70°22'30.17''N**	**68°43'50.42''W 70°32'30.48''N**
Source ArcticDEM 5m	786	751	382	469	125
ArcticDEM 5m merged	782	752	384	469	126
Source CDEM	797	744	373	470	123
CDEM transformed (3413)	797	744	373	470	123
CDEM Focal Stats (radius 2)	797	744	375	470	123
Final Clyde Inlet DEM	789	753	384	469	126
Differences after processing	+3	+2	+2	0	+1

Table 3-7: Elevation measurements of the terrestrial terrain during several processing steps

Moreover, it was shortly tested to what degree the elevation would change while conducting the processing steps. Table 3-7 presents elevation changes of about 1 m – 4 m while merging the data. The transformation into another reference system does not change the elevation when the nearest neighbour assignment is used. Also, the Focal Statistics tool is barely changing the elevation since the radius of just two cells is rather small. In the end, the elevation of the processed elevation model changes with regards to the original source dataset only about a couple of meters which is acceptable. However, it is recommended if possible to use datasets from the same source to inhibit harsh transitions.

3.4 Combining Bathymetric and Terrestrial Terrain Data

In this section bathymetric datasets will be combined with terrestrial datasets to obtain a final height map and splat map that can be imported into UE4. A height map is a greyscale image that stores elevation data where white areas represent the highest elevation and black areas represent the lowest elevation. A splat map, on the other hand, can be a greyscale or RGB (Red, Green, Blue) image and colour codes information to ensure a correct placement or blending of textures.

3.4.1 Height Map

The computation was conducted using the softwares ArcMap and QGIS version 2.18.13. An overview of the workflow is given in Figure 3-14 and for the complete workflow see Figure 9-1.

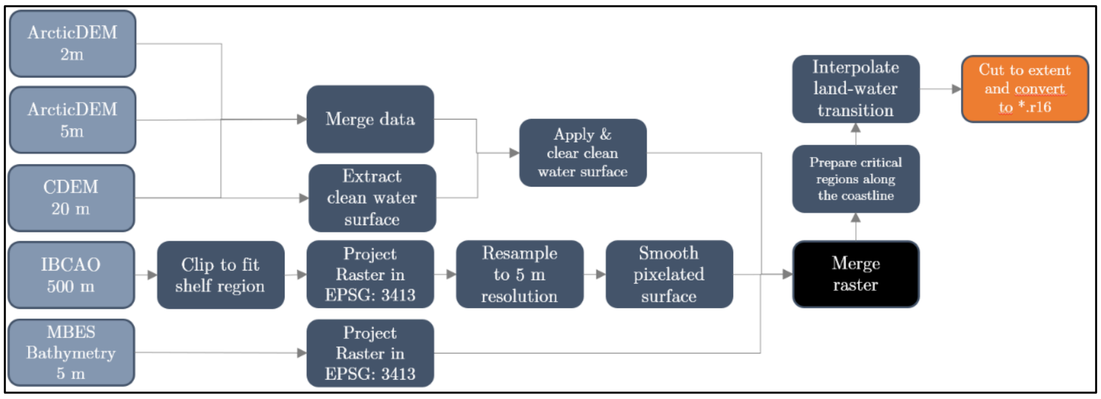

ArcticDEM 2m
ArcticDEM 5m
CDEM 20 m
IBCAO 500 m
MBES Bathymetry 5 m
Merge data
Extract clean water surface
Clip to fit shelf region
Project Raster in EPSG: 3413
Resample to 5 m resolution
Smooth pixelated surface
Project Raster in EPSG: 3413
Apply & clear clean water surface
Merge raster
Prepare critical regions along the coastline
Interpolate land-water transition
Cut to extent and convert to *.r16

Figure 3-14: Processing workflow for combining bathymetric and terrestrial elevation data

At first, the transition from terrestrial elevation data to MBES bathymetry data had to be addressed. Due to the fact that the sampled bathymetric data never reaches the shore and the satellite imagery data never penetrates the water surface, a gap at the transition will be always present and must be interpolated to prevent sharp edges. The elevation data processed so far, however, yields values between 1 m – 4 m for the water surface, which would be represented in Unreal Engine 4 as terrain above water. To be able to interpolate the transition, all values representing the water surface must be assigned to the value of "NoData". To simplify the reclassification of values, the already cleaned water surface of the CDEM was used to extract all water bodies. All cells with a value of 0, could be easily assigned to "NoData", by using the Reclassify tool in ArcMap. A disadvantage of using the CDEM is, that it has a poorer resolution than the ArcticDEM leading to a less precise location of the shoreline which was, however, accepted. The extracted water surface was, thus, merged with the preliminary elevation model described in section 3.3.1 leading to an elevation model with empty values for all water bodies.

The transition within the fjord is relatively narrow ranging from approx. 30 m to 1000 m. To fill the transition with IBCAO data would be unreasonable due to its poor resolution of 500 m. However, the region from the coastline to the shelf edge is only covered to one fourth with MBES bathymetry data. Therefore, IBCAO data was applied here. Like the CDEM before, the IBCAO data had to be transformed to the projection with the EPSG code 3413 and was resampled to a resolution of 5 m. Furthermore, the data was slightly smoothed through the Focal Statistics tool. In this case a neighbourhood radius of 50 cells was applied. Additionally, to ensure that the location of the coastline is maintained, IBCAO data was only applied approx. 1000 m beyond the shore. A polygon was drawn manually with 1000 m spacing from the coast and the IBCAO data was cropped according to this mask.

At this point, the terrestrial elevation model was merged with the IBCAO and the MBES bathymetry data as illustrated in Figure 3-14. To close the gaps within the transition the tool Fill-Nodata from QGIS was used. This tool, however, computes the value for the output cell depending on the neighbourhood cells. Hence, cell values between the mainland and islands would obtain positive values and would not be separated with each other. Also, in locations with a wide transition, shorelines would be shifted much further waterward than acceptable. To account for these problems, several masks in critical locations where prepared as seen in Figure 3-15. Cell values which represent the water surface and are covered by the masks, receive an elevation value of -5 m. This value was chosen to ensure a relatively smooth transition but to also maintain the shoreline as far as possible. All remaining gaps where then closed by interpolation.

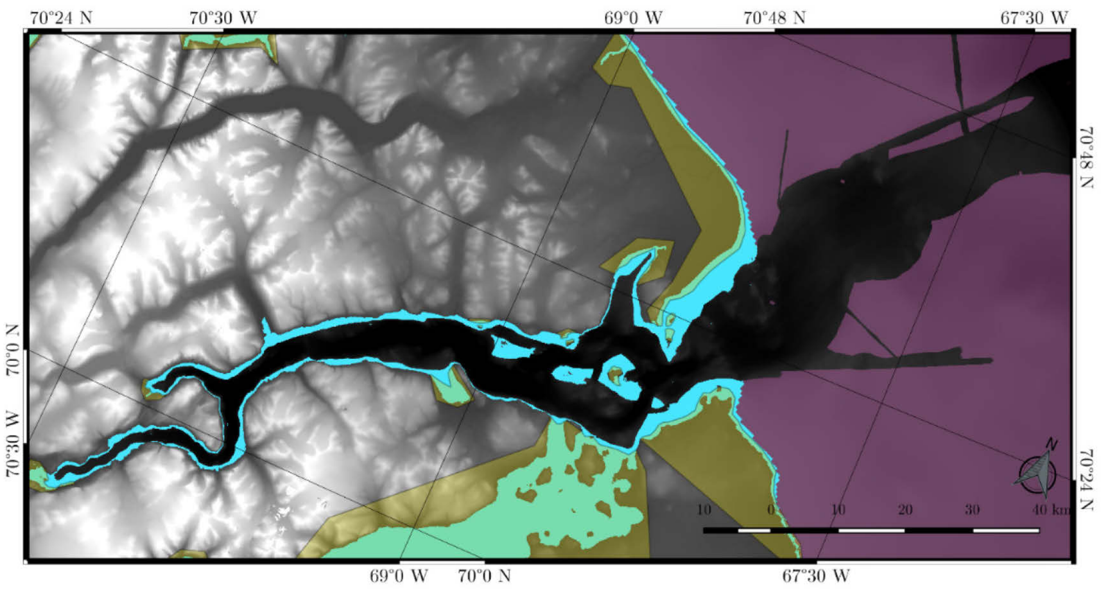


Figure 3-15: Final DEM showing the IBCAO data (pink), interpolated transition (blue) and masks for critical locations (yellow) (created in QGIS)

The finalised digital elevation model was cropped to an extent that fit 16 x 8 tiles each containing 2017 x 2017 pixel. The ratio of 2:1 for the number of tiles was chosen to ensure a correct scaling when importing into UE4. Additionally, the tile size of 2017 x 2017 pixel was chosen because it yields the best tested performance as described in section 6.2. Giving the resolution of 5 m each pixel, the overall extent of the area amounts to 161.285 km x 80.645 km. Furthermore, the elevation model had to be converted into a format supported by UE4. Therefore, the software L3DT (Large 3D Terrain) v.16.05 was used to convert the data to the file format *.r16.

3.4.2 Splat Map

The splat map was created in ArcGIS 10.4.1 for Desktop. Using the Reclassify Tool, all five terrain features such as glaciers, terrestrial terrain, transition, MBES and IBCAO data were assigned with a different greyscale value as illustrated in Figure 3-16. Thus, each feature could be allocated with its respective texture in UE4 as described in section 4.2.2. At which processing step each landscape body is extracted and reclassified is indicated in Figure 9-1.

The water body was split into three categories to differentiate the interpolated area as well as the two differently resolved datasets: MBES and IBCAO. Moreover, the spatial expansion of the RGI data had to be furthermore shrunk because its low resolution would result in pixelated contours which would swap over rocky slopes resulting in an unrealistic representation of glaciers. All contours were also later on traced using the painting tool in UE4 to assure a realistic glacial flow (refer to Figure 4-2).

In the end, all features were merged and the final map was cropped to the above mentioned extent, resampled to the maximum size of 8192 x 4096 pixel and exported as PNG.

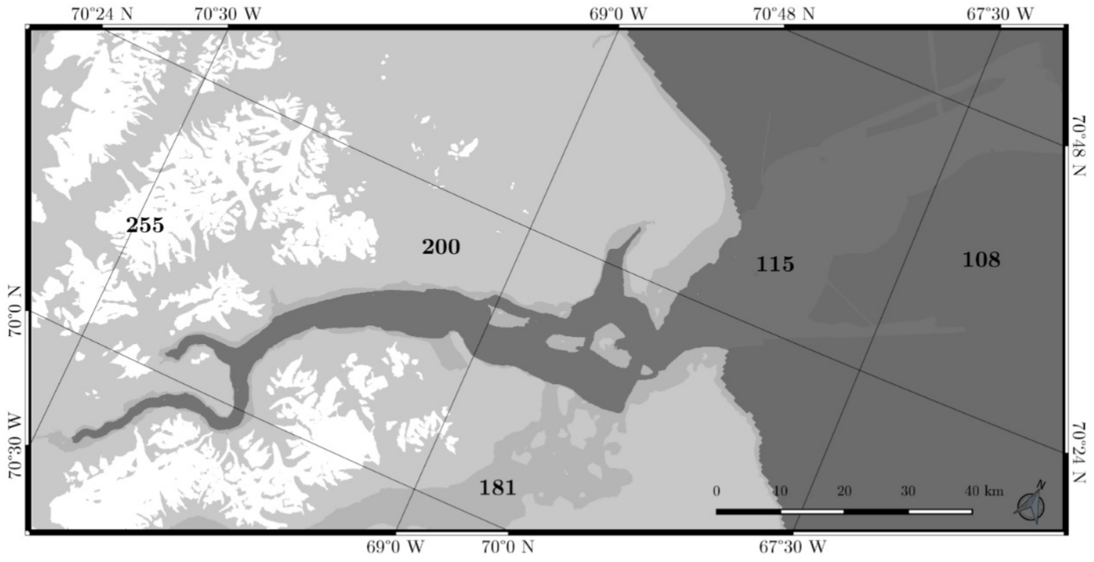


Figure 3-16: Splat map with greyscale values for glaciers (255), terrestrial terrain (200), transition (181), MBES data (115) and IBCAO data (108) (created in QGIS)

4 Development of Virtual Reality Application

This chapter covers the development of the virtual reality application using the software Unreal Engine 4 version 4.17.2. The digital elevation model as well as the splat map computed in chapter 3 have been used to generate and texturize the three-dimensional terrain. At first the DEM was imported and adjusted to ensure an adequate performance while running the application. Then the environment and landscape texture have been designed to provide both a realistic and functional view of the terrain and scenery. In order to navigate within the world, three locomotion systems have been developed. In the end, several measurement functionalities and other settings have been evolved to interact with the terrain. An overview of the workflow is given in Figure 4-1.

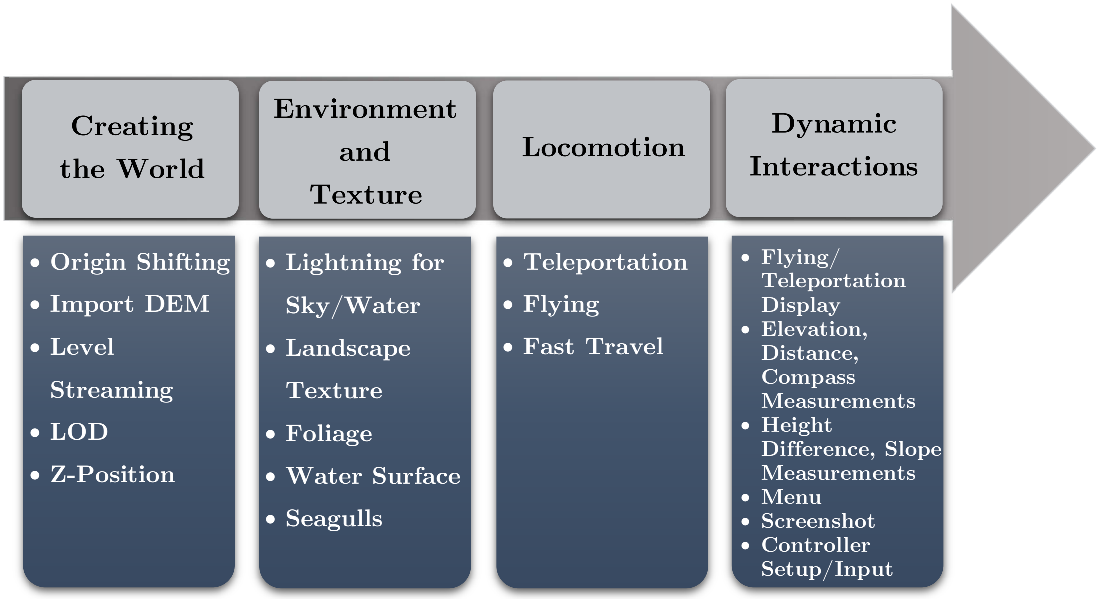


Figure 4-1: Overview of the workflow for the development of the virtual reality application

4.1 Creating the World

World Composition and origin shifting

After a new blank project has been created in UE4, the feature World Composition was enabled in order to create tiled worlds. World Composition is a tool specialised to manage large worlds in a simple way. One of its settings is origin rebasing which through activation will change the world's coordinate origin when the user moves too far away from the current origin. Unreal Engine 4 has a hard-coded maximum world size (WORLD_MAX constant) which currently amounts to 20 km. In order to circumvent world limits, origin rebasing was enabled which resets the origin after an arbitrary amount of about 5 km to the current position of the user. When world translation is initialised, the offset of the origin shift will be added to all registered Actors in the world (Epic Games, Inc., 2018a).

Import and scaling

The height map processed in section 3.4.1 was imported in UE4 as a tiled landscape with 2017 x 2017 pixel each tile. To ensure a 1:1 scale for real life representation of the terrain, a specific scaling factor had to be applied. To calculate the X and Y scaling factor, the overall width of the terrain (161,285 km) had to be divided by the product of 2017 (tile size) and 16 (number of tiles in X extent). The result was multiplied by 100 and yields 499.8 which is the same for both, X and Y scale due to their 2:1 ratio. To adjust the vertical scale, the total vertical range (2741 m) had to be multiplied by the constant 0.1953125 (Den, 2015) yielding 535.4. Subsequently, all tiles of the elevation model could be imported – each as a new level. Levels store all objects and everything that can be seen on that particular part of the landscape. A summary of the landscape architecture is given in Table 4-1.

Landscape parameters	
Number of tiles	128
Overall resolution	32,257 x 16,129 pixel
Total components	32,768
Number of components	256 x 128
Sections per component	2 x 2 sections
Section size	63 x 63 quads
Number of triangles per tile	8,128,512
Size of tile	10 km x 10 km (2GB)

Table 4-1: Parameters of imported landscape in Unreal Engine 4

Level streaming

Level streaming was implemented to load and unload only relevant parts of the landscape to account for memory issues when realizing large seamless worlds. As seen in Table 4-1, one tile already takes up 2 GB of space. Through level streaming, only loaded tiles will be rendered and loaded into memory. To initialise level streaming, all tiles were assigned to a new layer that holds information about the streaming distance so that all tiles can inherit that information (Epic Games, Inc., 2018a). Distance based level streaming will load and unload tiles according to their distance from the user. The user, thus, can only see contiguous parts of the landscape, whereby distant parts will be invisible. For this project, a streaming distance of 10 km was chosen to have only adjacent tiles being loaded into memory.

Level of Detail

Level of Detail (LOD) is furthermore used to save up memory and increase the performance. The main approach encompasses the reduction of complexity of a 3D model as the user moves away. Basically, the number of triangles of the mesh will be reduced. When up close, the user will see everything in full detail. Because of the fact, that level streaming will hide unloaded tiles, LODs were used to show hidden landscape parts in a much lower resolution and thus enable an overview of the entire landscape of Clyde Inlet with a feasible performance. In UE4, each level can have up to four LOD distance based streaming tiles, however, only with a valid license of the software Simplygon (Epic Games, Inc., 2018a). Without this licence, UE4 creates only one LOD per level using a fixed triangle reduction. For each tile, the triangles were decreased by 99,97% to a total amount of 2048 triangles. The streaming distance was set to a maximum of 100 km and applies after the level streaming distance, meaning that all low-resolution tiles will be loaded after the level streaming distance of 10 km. Now even most distant parts of the landscape will be visible in the application.

Z-position

During import, the position of the landscape could not be adjusted. Therefore, the landscape was placed unsupervised in the absolute world coordinate system. To ensure the ease of use, the z-position of the landscape was adjusted afterwards so that zero elevations will correspond with the zero point of the z-axis. The entire landscape and all LOD levels were, thus, shifted by 523.5 m (2741 m divided by 2 and subtracted by 847 m).

4.2 Designing the Environment and Landscape

4.2.1 Lightening

Sky

Lightning for this project has been realised using three Actors: Atmospheric Fog, Directional Light and Sky Light. All of these interact with each other to provide a realistic landscape scene. The Atmospheric Fog simulates atmospheric light scattering and drives the sun effect in the sky. It literally, creates a sun disc and ray scattering around the disc. Furthermore, it creates the horizon and changes the colour of the sky relative to the altitude of the sun (Epic Games, Inc., 2018d). The Directional Light is an infinite light source which therefore facilitates parallel shadows as if the objects where hit by distant sun rays. This light is thus ideal for simulating sunlight. The temperature of this light was set to 5500 Kelvin to provide a warmer scene. The Sky Light is designed to receive light from distant parts of the landscape and projects it back to the scene. It mainly lightens up shadow areas (Epic Games, Inc., 2018e).

Water

To apply under water effects, a Post Process Volume has been stretched over the entire landscape covering the area under water. Post Process Volumes are used to tweak the overall look of a scene and thus implements several effects. These effects, however, are only applied when the user is within the bounds of the volume (Epic Games, Inc., 2018f). To create the under-water impression the above listed effects have been used (Figure 9-2):

- Scene colour tint: To create a blueish tint to the scene
- Bloom: To simulate a blurrier effect of the sun
- Lens Flare: To create a blueish lens flare tint
- Vignette: To darken the edges of the image
- Depth of field: To apply a blur to the scene after 10 m which gradually increases with distance

4.2.2 Landscape Material and Foliage

Landscape Material

The landscape Material for Clyde Inlet was designed to allow the user to switch between several different colourations. For the terrestrial terrain, the user can choose between either a natural texture or a greyscale texture which represents a Hillshade. The hydrographic sampled data can be either coloured as a Hillshade as well, according to its backscatter values or charted in rainbow colours relative to the terrain height. The transition zone as well as the IBCAO bathymetry are illustrated as another greyscale value to differentiate them from the MBES data. All landscape textures and functionalities are contained in one single landscape Material which is assigned to every level in the world.

To create the natural terrestrial colouration, the package "Photorealistic Landscape Pack2" developed and released by Gokhan Karadayi in September 2015 was implemented. This set provides 15 diffuse textures, 15 normal textures, 9 specular textures and 3 height map textures. It furthermore provides the blending logic of textures and the automatic slope-based colouration for cliffs and rocks. This algorithm will colour rocky textures only for very steep terrain. In the end, 5 grass and ground textures, 1 snow texture and 1 rock texture including their respective normal, specular and height map textures were used.

The snow was painted automatically according to the RGI dataset, colour coded by the splat map created in section 3.4.2. Before the splat map texture could be used, its Compression Settings had to be set to Grayscale and the setting sRGB had to be disabled. Only then, greyscale values could be accessed, and textures matched to the respective grey value. Moreover, the tiling of the splat map texture had to be adjusted in UV direction so that it will have the same extent as the entire landscape. Therefore, the U direction was tiled by 1/32,257 and the V direction by 1/16,129 (see Table 4-1 for the divisor). Using a Custom Material Expression and C++ code, all values smaller than 255 were allocated with grass, ground and rock textures. Every part of the splat map that equals 255, was painted as snow (refer to Figure 9-3 for the Blueprint coding). However, due to the stretching of the texture, all contours received a pixelated structure and had to be traced manually to obtain a realistic natural view (Figure 4-2).

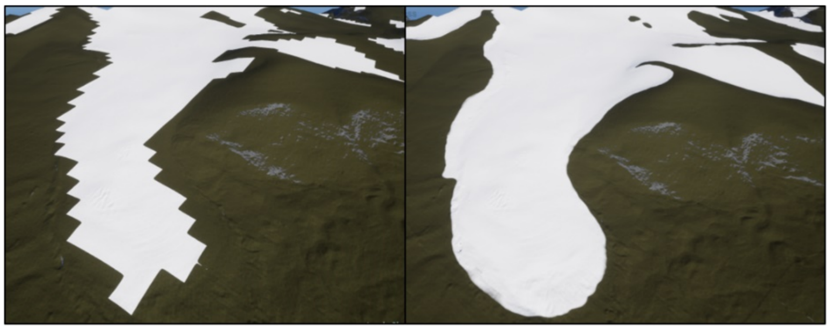


Figure 4-2: Snow coloured automatically according to splat map. Left: without manually painting. Right: with manually contour tracing

To switch between "Natural" and "Hillshade" view, a Material Parameter Collection (MPC) was implemented. This asset stores any desired number of scalar or vector parameter that can be accessed and changed during running application. Hence, it is mainly used to drive effects such as wetness, sun angle, snow amount or terrain colour (Epic Games, Inc., 2018b). The MPC used in this project contains four different scalar parameters. The "BaseColourTerrestrial" parameter drives the switch between "Natural" and "Hillshade" view. Its default value is set to 0 and the default colouration is the "Natural" texture. When the MPC is switched to a higher value than 1, the "Hillshade" will be activated as seen in Figure 9-4.

The different regions of the bathymetry were coloured in a very similar fashion as the snow colouration. Also here, the splat map was used in combination with a Custom Material Expression and C++ code (refer to Figure 9-5).

All regions below greyscale value 108 received the transition colouration, all regions between 109 and 155 received the colouration for the MBES data and all regions above 116 received the greyscale colouration for the IBCAO dataset. The different MBES textures were handled using the MPC parameter "BaseColourBathy". The default texture is the backscatter colouration. When the MPC is switched to a number equal to 1, the rainbow colouration is enabled. When switched to a number higher than 1, the Hillshade will be activated (Figure 9-6).

The backscatter was coloured using a yellow tone representing sand, ochre representing clay and blue representing gravel. To assign those values to its corresponding position, the backscatter image processed in section 3.2.4 was used as a RGB splat map. Consequently, each RGB node was assigned to ochre, yellow and blue, respectively (Figure 9-7).

On the other hand, the rainbow colour ramp (Figure 4-3) will be stretched over an arbitrary height range. The minimum and maximum height value can be set during running application through parameters in the MPC (refer to Blueprint code in Figure 9-8). The overall minimum value, however, is 0 m and the overall maximum value is -700 m.

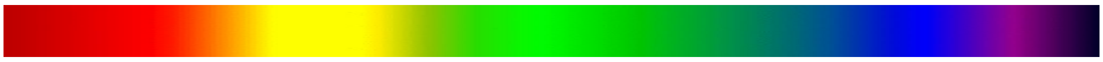


Figure 4-3: Rainbow colour ramp used to colour the terrain corresponding to a given height range. Shallow elevation will be coloured red and deep elevation will be coloured dark purple

In the end, both the entire terrestrial texturing and the bathymetric texturing were linearly interpolated so that they appear above and beyond the zero elevation, respectively. This could have been solved also easily using the splat map. However, as mentioned before due to the stretching of the texture, the splat map will yield pixelated contours that do not provide a decent look. The Linear Interpolate node, thus, reveals a smooth cut corresponding to the terrain height. The entire landscape Material can be seen in the appendix in Figure 9-9.

Moreover, it should be noted that the Sampler Source for each texture was set to “Shared: Wrap”. This was induced in order to exceed the maximum number of textures specified for one Component. The limit is 16 textures per Component including the masks for each layer.

Furthermore, a light version of the Landscape Material had to be assigned to each LOD tile. Light version because only one grass texture was applied to represent the “Natural” view. Because LOD tiles do not share an overall Landscape Actor that holds the information about the entire tiles and their overall extent, the splat maps cannot be stretched over the entire landscape. Therefore, each LOD tile had to be provided with a splat map representing its corresponding section. The greyscale as well as the RGB splat map for the backscatter colouration were split into 16 x 8 mosaics and assigned to each related LOD tile.

Additionally, a Normal texture for each level (that was created during the LOD process automatically) needed to be assigned to the material. The Normal texture provides further information about the relief of each specific tile. Applying these, yield a much more in-depth view of the topography of the terrain even though the resolution of each LOD might be very low. The difference between a landscape with and without applied normal texture can be seen in Figure 4-4.

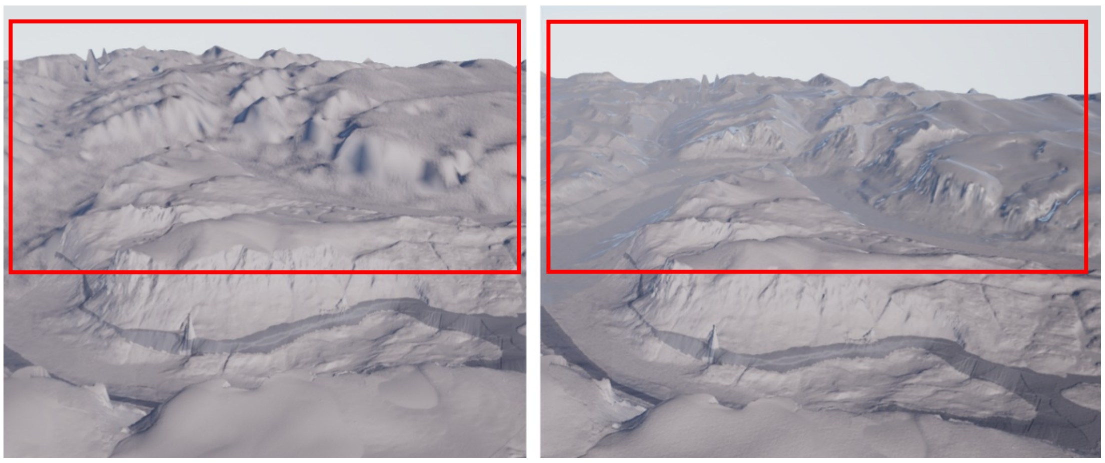


Figure 4-4: Distant landscape without normal texture (left) and with normal texture (right)

Foliage

The “Photorealistic Landscape Pack 2” ships with an integrated foliage system consisting of three different grass types and two different flowers – Papaver and Artemis. However, only the Artemis flower was used as it shall represent the purple Alpine catchfly which is native to Canada (iNaturalist Canada, 2016a). Furthermore, two other flowers were assigned to the foliage system: common Heather and a white flower representing Northern Androsace. Both appear in northern Canada (iNaturalist Canada, 2016b&c) and were imported from the package “Open World Demo Collection” developed and released by Epic Games in May 2015.

The foliage logic was realised using the GrassTool which consists of two main Actors namely Landscape Grass Type and landscape Material. The Landscape Grass Type holds all used grass and flower meshes and regulates amongst others the density of the distribution and the cull distance of the foliage. The cull distance is the distance at which objects will not be drawn to the screen anymore based on their distance from the user. This will optimize the scene as objects in further distance will be considered unimportant and will not be rendered (Epic Games, Inc., 2018c). On the other hand, several nodes were used inside the landscape Material to spawn the foliage at a specific layer.

4.2.3 Water Surface

The water surface added to this project was imported from the open source project “Water Planes” developed and released by Epic Games in January 2015. Epic Games’ project contains a collection of water surfaces for lakes and oceans. For this thesis the Blueprint “LakeWater” was used and dragged over the entire landscape extent. It contains a Static Mesh Component that represents the water surface. To create surface motion, the texture illustrated in Figure 4-5 (left) and several Panner expressions were used to pan the texture coordinates in the UV direction under a certain speed. Moreover, several parameters such as wave scale, speed and roughness were set to achieve a realistic look as seen in Figure 4-5 (right).

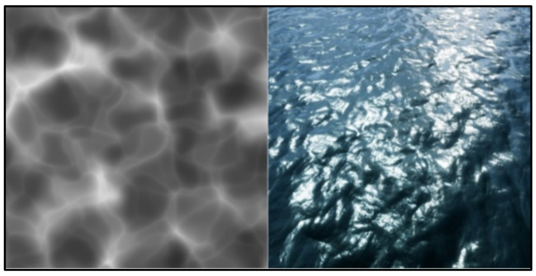


Figure 4-5: Surface motion created from the texture sample (left) and its result (right)

4.2.4 Seagulls

To create an increased immersion in the world, several seagulls were placed in the landscape. These seagulls were imported from the open source project "Landscape Mountains" which was developed and released by Epic Games in August 2014. Each bird is made up of a Static Mesh, several textures and a Material that contains the movement logic such as flapping motion and wing twist. To spawn birds in the world and have them fly forward, a Particle System was induced. Such a system is used for visual effects and controls in its basics settings the direction of movement, velocity, spawning rate, size, life time and colour over life. The particle emitter was added in a Blueprint Class which then again was placed at four different locations in the landscape. Birds placed at each different location will either fly in a straight or circular pattern. Furthermore, the amount of birds varies per location. Additionally, it was ensured that only those birds will be rendered that are located above a loaded tile. All others will be destroyed and deleted from memory. For the functionality refer to Figure 9-10 - 9-12.

4.3 Developing Locomotion

Teleportation

To navigate within the world, three different locomotion systems have been utilised. The first system is the Motion Controller Teleportation which teleports the user forward by an arbitrary amount between 1 m and approx. 25 m. This system is the default navigation system of the Blueprint Virtual Reality Template and was migrated to the Clyde Inlet project. It consists of several Static Meshes of an arrow, a beam and a flat cylinder. While a button is pressed, an arced beam appears which traces the landscape. At its end, the arrow and the cylinder show up and rotate according to the user's hand motion. On button release, all meshes disappear and the user teleports to the arrow's last position. This teleportation system is the only navigation system that keeps the user onto the ground and prevents motion sickness most of all.

The navigation logic also contains a valid location check, that proves if the user is allowed to navigate to the indicated position or not (Figure 4-6).

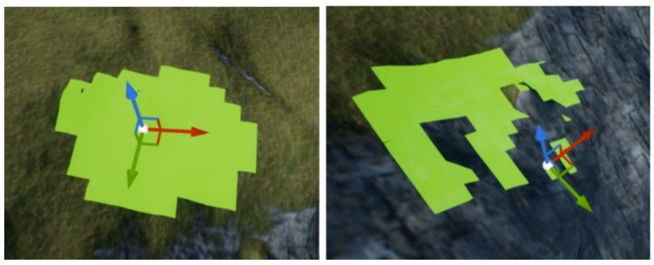


Figure 4-6: The green area around the user (user indicated by the coordinate axis) shows all possible places that can be accessed at a defined radius. If the user moves forward, the Navigation Invoker updates and provides new accessible places. However, no access is granted at very steep terrain (right)

That way it is impossible for the user to navigate on very steep terrain or to jump into deep chasms. Therefore, this type of locomotion requires a Navigation Mesh that filters valid destinations. On default, the Navigation Mesh must be stretched over the whole terrain to compute navigation data for every location the user shall reach. However, when using large terrains this can be very cost effective. Hence, a Navigation Invoker Component was added to the Pawn which computes the navigation data only around the user. To be able to invoke valid destinations, a Navigation Mesh still has to envelop the landscape, only that through this method the navigation is not being computed for the entire terrain. A lot of troubles had been caused while world origin rebasing is being initialised. Therefore, the Navigation Mesh was reduced in size and attached to the user as well (Figure 9-13).

Flying

The second navigation system is the flying movement which does not restrict the user to the ground as the name already implies. Its basic function is the Add Movement Input node which receives the pitch and yaw motion of the right controller and translates it to the virtual movement (Figure 9-14). As one might indicate, the pitch motion yields the up- and downward movement while the yaw motion handles the left and right movement. However, it should be noted that an exact upward or downward motion is not possible. The user will always tend to move in a certain left or right direction when flying up- or downwards. The reason for this could not be identified. Additionally, the speed can be changed during running application. The default speed is 43 km/h (12 m/s). To address the movement a Floating Pawn Movement Component had to be added to the Pawn. Here, the acceleration was increased to 80 m/s^2 and the deceleration increased to 160 m/s^2.

Fast Travel

The Fast Travel navigation system is a convenient alternative to the above-mentioned systems. It allows the user to be quickly transported to a distant waypoint without having to teleport or fly to the location. Throughout the landscape 12 waypoints have been distributed along the fjord and an overview point grants access to a top down view of the entire landscape. The waypoints are accessible within the menu (refer to 4.4.4). On button click, the user will be spawned instantly at the selected location.

To achieve this, 13 "Waypoint" Actors have been placed in the world. Each Actor saves its name and location within a variable that is being called inside a Game Instance Class (Figure 9-15). The advantage of this globally accessible class is that it can store and transfer any data between levels. So, the location and name of each waypoint will not be deleted when switching between levels. The Game Instance Class contains two custom events. The first event loads the activated streaming level (Figure 9-16) and the second event calls the name and location of the selected waypoint and transports the user to that specific waypoint (Figure 9-17 - 9-18). At some locations where the user will be spawned in high altitudes, the water surface will be disabled automatically to obtain a clear view of the bathymetry. The first event of the Game Instance class will be fired inside the menu Widget (Figure 9-20), when a waypoint has been selected (Figure 9-19). The second event fires on Event BeginPlay when the level has been loaded. Hence, first the level will be loaded and then the user will be transported to that location.

4.4 Evolving Dynamic Interactions

4.4.1 Flying and Teleportation Locomotion Display

The flying and teleportation locomotion display has two layouts which switch with the locomotion. The flying layout indicates at which speed the user moves, whereas the teleportation layout displays the distance of the jump.

The display is created inside a Widget which is being attached to the right motion controller. The Widget holds the layout and text of the display. It furthermore consists of a WidgetSwitcher which toggles between the flying and teleportation layout. The values for speed and distance are communicated via variables inside functions which can be called to pass correct real time values (Figure 9-21). In order to display the Widget in the 3D environment it had to be stored as a Widget Component inside a Blueprint Class. This Component provides a surface in the 3D world on which the Widget is being rendered. Normally, in 2D environments Widgets are just rendered to the screen and do not need a special render target which is then placed into the scene.

Inside the Pawn Class, the location and rotation of the Widget is being adjusted (Figure 9-22). Additionally, a Boolean was used to indicate whether the layout has to switch to the flying or teleportation display. Values for the speed are accessed, converted from UU to km/h and then passed to the layout of the Widget (Figure 9-24). Moreover, all values higher than 346 km/h are painted yellow and all values higher than 1382 km/h are coloured red to indicate the rate of smoothness the application will run at (Figure 9-25). Because the higher the speed, the more often the system has to shift the origin as well as load and unload levels. All of this cost computational power and slows down the performance. It is thus recommended to navigate with a maximum speed of 346 km/h to prevent system lags. The distance of the teleportation arc is also passed to the Widget, however, only when a valid teleportation destination has been found. Furthermore, two functions were created to spawn and destroy the Widget (Figure 9-23). On Event BeginPlay the "ShowVelocity" function is called to spawn the Widget when the application starts.

4.4.2 Elevation, Distance and Compass Measurements

The elevation, distance and compass measurements are as well displayed in one Widget which is this time attached to the left controller. The compass indicates the direction the user is facing. With the help of a laser, all parts of the landscape within range can be accessed and the elevation above mean sea level as well as the distance to the hinted point retrieved.

As explained above, the values for the elevation and distance are stored in functions and are changed in real time. The compass consists of a texture (Figure 4-7), frame and a material. The material is used to pan the texture in the U direction so that only that letter is centred on which corresponds to the rotation of the user. The adjacent letters will be faded out (Figure 9-26). The material is placed as an image inside the Widget and set as a dynamic material to enable the panning movement in U direction (Figure 9-27).

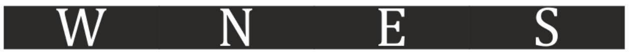


Figure 4-7: Compass texture indicating West, North, East and South. The texture centres always on the respective letter according to the user's rotation

As mentioned above, the Widget is stored as a Widget Component inside a Blueprint Class. It is attached to the left controller and spawned when the application starts, just like stated before.

To set up the laser beam, first of all, a material was created which generates an image that seamlessly spans horizontally and is brighter in the inside than the outside (Figure 9-28). This material was added to a Particle System to create the beam itself. The most important settings are the type, speed, size and colour of the beam. The type was set to "Target", to allow the beam being rendered from a source point to the target point.

Moreover, source and target point are set to user specified to control both settings via Blueprints. The colour is defined as red and the speed as 0 to obtain an instantaneous beam.

The Particle System, in turn, was added as a Particle System Component to a Blueprint Class. Within the Blueprint, a function was created that does a simple line trace to detect objects that are being hit by a line that starts at a point and ends at another given point in the world. The object and the location of the hit are returned. In this case the line trace starts at the left motion controller and travels 10 km in a straight line from the controller (Figure 9-29). The source point of the laser Particle System is hence defined as the starting point of the line trace and the target point is defined as the end point of the line trace. Thus, the laser is stretched at a length of 10 km and attached to the left motion controller. To visualise the hit point in the world, a Static Mesh Component in a shape of a sphere was added. The sphere receives the hit location and moves to that location instantly. When no intersection occurs, the sphere will turn invisible. Moreover, the size of the sphere was scaled with distance so that the user is able to see it also in great distance. The closer the sphere gets to the user, the smaller is its size (Figure 9-30).

Subsequently, the value for the elevation at the hit point gets updated and passed to the Widget. Also, the distance between the user and the hit point is passed. When no hit occurs both the value for the elevation and the distance receive the value 0 (Figure 9-31). In the end, two more functions were programmed to hide and show the laser which will be needed later on.

4.4.3 Height Difference and Slope Measurements

Moreover, measurements between two set points can be performed. Using the left motion controller, two points can be placed onto the terrain. A display will pop up and displays the calculated height difference, slope distance and slope in degree between the two indicators. To clarify which point has a higher elevation than the other one, a plus and minus symbol will be placed on top of each point.

Two spherical meshes symbolising the two indicators as well as the plus and minus meshes are stored in a Blueprint. Two functions are then used to position each indicator at the respective hit location of the laser. When the function for the first indicator fires, the second indicator will be destroyed, if it was already placed in the scene (Figure 9-32). Subsequently, the symbols are placed according to the elevation of the indicators (Figure 9-33 - 9-34). They furthermore rotate so that they always face to the user (Figure 9-35). The size of the indicators, symbols as well as the height distance between the symbols and the indicators are scaled respectively to the distance from the user (Figure 9-36). Thus, the greater the distance from the user, the larger is the size of the meshes and the height distance between the meshes. This was conducted so that every mesh is still visible even in great distance. Values for the height difference, slope distance and slope are passed to the Widget which is stored in a Blueprint Class (Figure 9-37). Every time the second indicator is positioned in the scene, the Widget is spawned on top of that indicator. Also here, the size and the height distance varies with distance from the user (Figure 9-38). When the first indicator is placed while the second one was still active, both the second indicator as well as the Widget vanishes.

However, due to unknown reasons the indicators will not spawn at indicated places after the user navigates to another waypoint. A lot has been tried, but to no success. The only thing to prevent this problem is to change the “Auto Receive Input” of the Actor “Marker” inside the Blueprint “Marker” to “Player 0”. Then the application has to be started, cancelled and the input changed back to “Disabled”. Only then everything works as described.

4.4.4 Menu

The menu contains several options to help the user find information and execute functions. It comprises six main options:

- Basic Information: provides an overview of the data that has been used for the generation of the terrain
- Positioning: holds the fast travel navigation system (refer to section 4.3) and an overview map as an aid to orientation
- Landscape: includes the functionality to change the colouration of the terrain (refer to section 4.2.2) and a switch to toggle the visibility of the water plane
- Controls: gives an overview of the input of the left and right controller
- Screenshot: allows the user to do a screenshot of the scene
- Exit Application: exits the application

The menu furthermore features a two-piece design consisting of two Widgets that incorporate two layouts. Both Widgets are stored as a Widget Component within a Blueprint Class. The first menu layout contains only buttons for the six main options. While hovering or clicking on a button, the second menu layout appears next to the first layout and contains the respective sub options. All sub options that belong to a main option are located on a Canvas Panel which serves as a container for text, buttons and so on. A Widget Switcher is now able to toggle between the six Canvas Panels and makes the selected Panel visible. Therefore, the second menu Widget needs to be accessed within the first menu Widget to link the index of each Canvas Panel to the respective button (Figure 9-39). Furthermore, an Event Dispatcher is utilised and called every time a button is hovered on, clicked or unhovered (Figure 9-40). The event, which toggles the visibility of the second menu Widget, is bound within the Blueprint Class, because only here the visibility could be accessed (Figure 9-41).

In order to navigate within the menu, an additional laser system has been implemented. To interact with the Widget in 3D space, A Widget Interaction Component had to be activated once the application starts. It simulates e.g. a mouse button click by using a motion controller trigger press instead. Two functions simulate the left mouse button press and release functionality and are later on called when the respective key of the motion controller is fired (Figure 9-49). To create the laser beam itself a Spline Component and a Spline Mesh Component were added to the motion controller. The spline is activated once the menu is visible and only if the spline is hovered over the Widget. If the spline hits the Widget it stretches from the right motion controller to the Widget otherwise it will shrink to invisibility (Figure 9-50). Once the spline hits a button, it can be clicked with the motion controller to execute the corresponding menu option.

Positioning

As stated above, the Positioning menu option contains the fast travel navigation system as well as an overview map to provide orientation. The overview map shows a small map section and an arrow icon that indicates the direction the user is facing. The icon is a Paper Sprite Component and is attached to the Camera Component to receive the rotation of the user. It is thus always connected to the user but made invisible for the Camera Component so that the user would never see the icon in the world.

A Scene Capture Component 2D is furthermore attached to the user and placed at a height of 20 km above the user. The Scene Capture Component 2D acts like a camera and captures an image of the scene in a top-down view. It is thus taking a snapshot on every frame of its view frustum depicting the arrow icon and its surroundings (Epic Games, Inc., 2018g). The image is translated into a texture which is then fed into a Render Target and placed on the Widget. While the Capture Component is very expensive considering the fact that it is rendering an image every frame at a very large view frustum, it is only activated while the respective menu page is being switched on (Figure 9-42).

Landscape

The Landscape menu option does not only contain the functionality to change the colouration of the terrain (Figure 9-43), it also offers the possibility to change the visibility of the water surface. However, the visibility state alone will not change the collision settings, meaning that a laser beam would still be blocked by the water surface even though the surface has been made invisible. Therefore, all collisions will be disabled once the water surface is switched off (Figure 9-44). Furthermore, every time the menu is closed and reopened, the ticks of the checkboxes from a prior selection remain (Figure 9-45).

Controls

The left and right description of the controller input are each placed on a different Canvas Panel that is toggled when the corresponding check box is ticked (Figure 9-46).

Screenshot

When a screenshot is initialised both menu Widgets are closed and reopened after a small intentional delay once the screenshot is performed. To close and open both menus two Event Dispatchers have been applied that listen and execute their functions once the button for the screenshot has been pressed (Figure 9-47).

Exit Application

To exit the application the simple function "Quit Game" was called once the button to quit the application has been pressed (Figure 9-48).

4.4.5 Controller setup and input

Controller setup

By default, hand meshes are spawned in virtual reality to represent the user's hand. They allow the user to interact with the world by pointing, touching or grabbing objects. However, for this project realistic hands are undesirable since they do not possess buttons. Therefore, a HTC Vive motion controller mesh and texture allocated by SteamVR was imported to the project. The mesh depicts the motion controller in a realistic way with all its buttons. Furthermore, the Vive logo of the texture was painted orange and pink to distinguish the left from the right motion controller. The texture was hence allocated to the respective controller (Figure 9-51).

Controller input

To provide an overview of the key bindings for the left and right motion controller, refer to Figure 4-8.

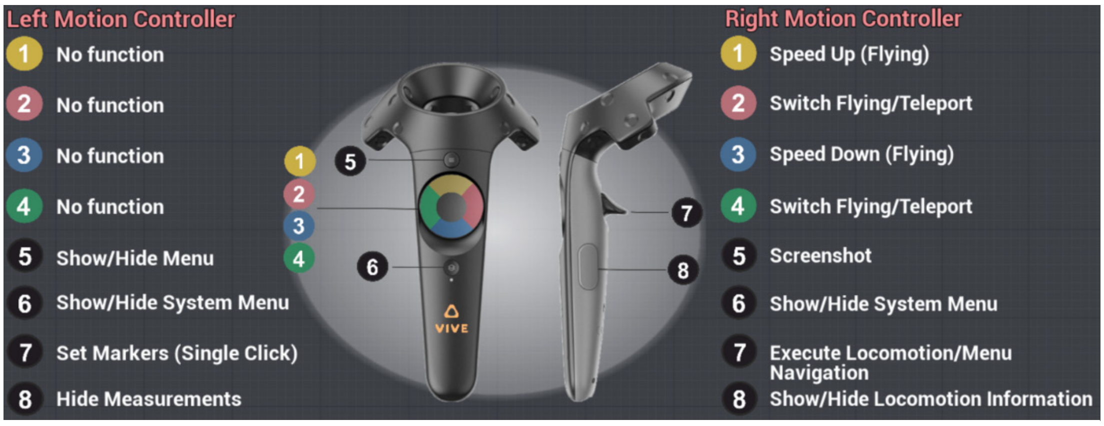


Figure 4-8: Key bindings for the left and right motion controller

The right controller mainly operates the locomotion such as flying and teleportation. The default locomotion is the flying movement. By pressing the right motion controller trigger button, the flying function fires and moves the user forward as long as the trigger is pushed. While releasing the trigger button, the user will decelerate and eventually stop

(Figure 9-52). To increase or decrease the speed, the yellow face button will double the speed each time it is pressed. The blue face button will halve the speed when pressed (Figure 9-53). To change from flying to teleportation movement, the user has to press the green or the red face button (Figure 9-54). Once the teleportation movement is selected, the right trigger will activate and execute the teleportation movement when pressed and released (Figure 9-55). Unlike the flying movement, the teleportation movement is initialised on every trigger click and release whereas the flying movement executes while the trigger is pushed down and not released. Whenever the menu is activated, the right trigger will be used to navigate within the menu and to click on buttons or check boxes. The right shoulder button is used to take a screenshot of the scene with a 2K resolution (Figure 9-56). Furthermore, it is possible to hide or show the flying and teleportation locomotion display (refer to section 4.4.1) by clicking the right grip button (Figure 9-57).

The left motion controller is mainly used to execute measurements and to enable or disable the menu. Once the left trigger button is pressed and released a measurement point is set onto the landscape. Whenever the trigger is pressed again, another measurement point is located on the landscape. A display listing the height difference, slope distance and slope will appear over the second measurement point. To hide the display, the left grip button has to be clicked (Figure 9-58). To open or close the menu, the left shoulder button should be pressed. At the same time as the menu opens, the measurement laser of the left motion controller disappears and a laser to interact with the Widget appears at the right motion controller. Furthermore, both measurement and locomotion displays attached to the right and left controller are being disabled. Once the menu is closed, everything switches back to the default setup (Figure 9-59).

5 Results

This chapter covers the results of the virtual reality application. Results comprise the visualisation of the terrain stretching from the head of Clyde Inlet until the continental margin of Baffin Bay. All in all, five different datasets have been used to create the terrain above and beyond the water surface with the best possible resolution. Furthermore, several functionalities and measurement tools were adopted to interact and work with the terrain. The following sections will illustrate the landscape and interactive tools of the virtual reality application. Moreover, recommendations on the system requirements are given.

5.1 Terrain and Environment

The terrain of about 160 km x 80 km is portrayed and can be entirely seen in the VR application. Figure 5-1 illustrates a top-down view of the terrain. The texture of the bathymetry dataset measured during the MSM66 cruise as well as the terrestrial terrain dataset can be altered. The default colouration of the terrestrial terrain is the natural, greenish colour including the glacial RGI dataset. The default colouration of the MSM66 bathymetry dataset is the ARA backscatter result. The IBCAO dataset as well as the transition between terrestrial terrain and bathymetry are coloured dark blue and dark grey to clearly demarcate the MSM66 dataset.

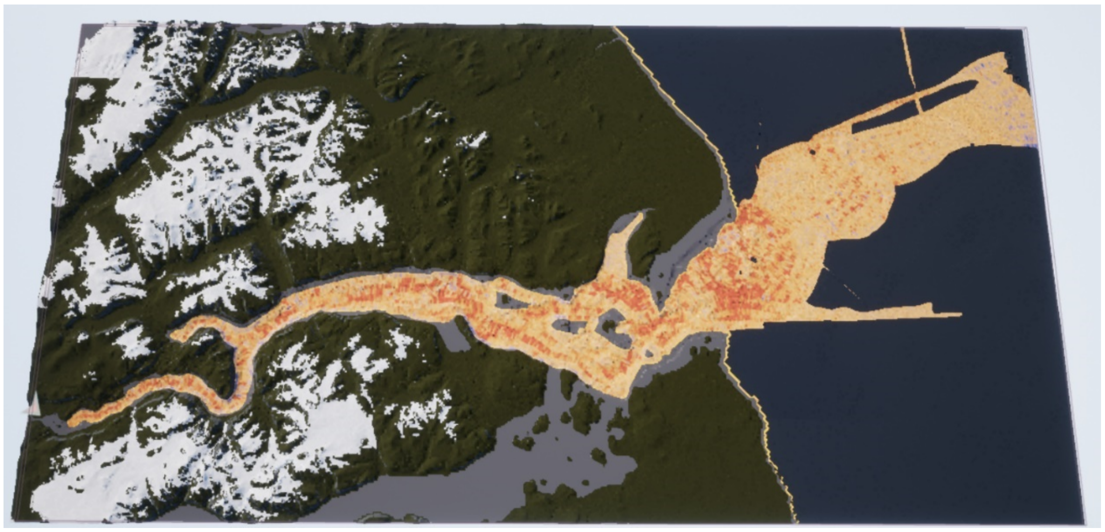


Figure 5-1: The terrain of Clyde Inlet visualised in the VR application with the default natural and ARA backscatter colouration

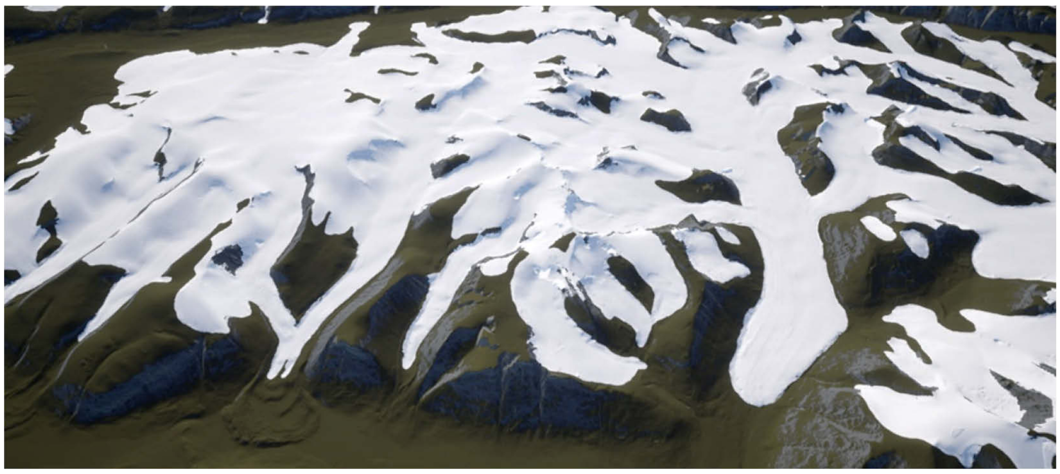


Figure 5-2: Visualisation of the RGI dataset in UE4

Figure 5-2 depicts a section of the RGI dataset and its numerous glacial tongues which lie perfectly in the valleys derived from digital elevation models ArcticDEM and the Canadian DEM. As described in section 3.4.2, all glacier outlines have been manually traced to smoothen the contours. The combination of the RGI dataset with the DEM, furthermore accentuates generations of moraines and glacier retreat. Additionally, the terrestrial terrain can be visualised as a Hillshade to depict the naked terrestrial surface. On the other hand, the bathymetry dataset collected during the cruise MSM66 can be visualised in three different modes. All landscape alterations are presented in the following figures. Furthermore, Figure 5-8 illustrates the applied foliage consisting of three grass and flower types. To increase the degree of immersion seagulls were placed at several spots throughout the landscape. Figure 5-9 shows the mesh of the birds and the distribution within the environment.

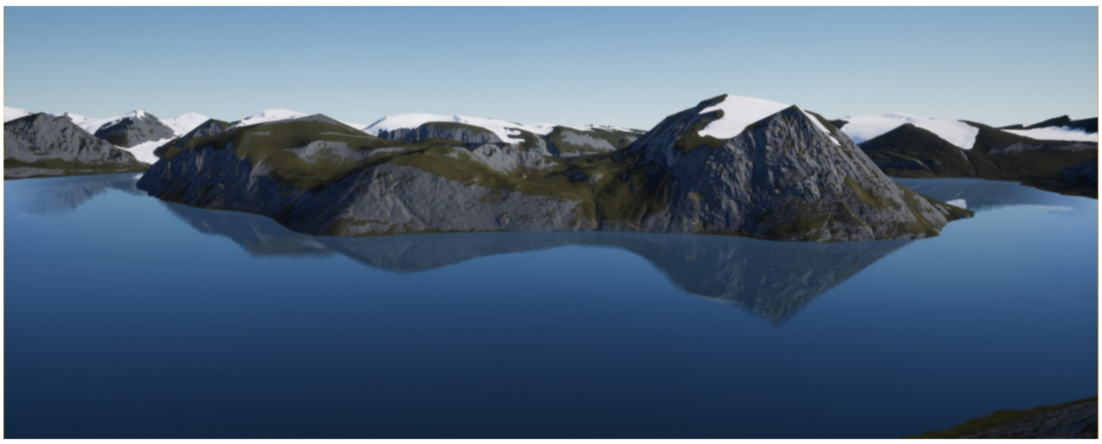


Figure 5-3: Default landscape scenery in UE4 with the natural terrestrial colouration and the water surface

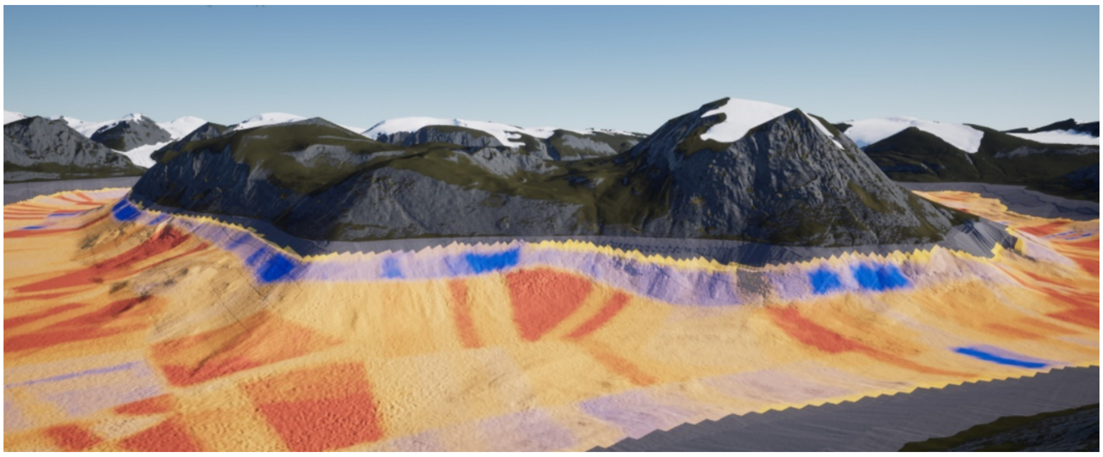


Figure 5-4: Landscape scenery in UE4 without the water surface depicting the default natural terrestrial colouration and the bathymetric ARA backscatter colouration

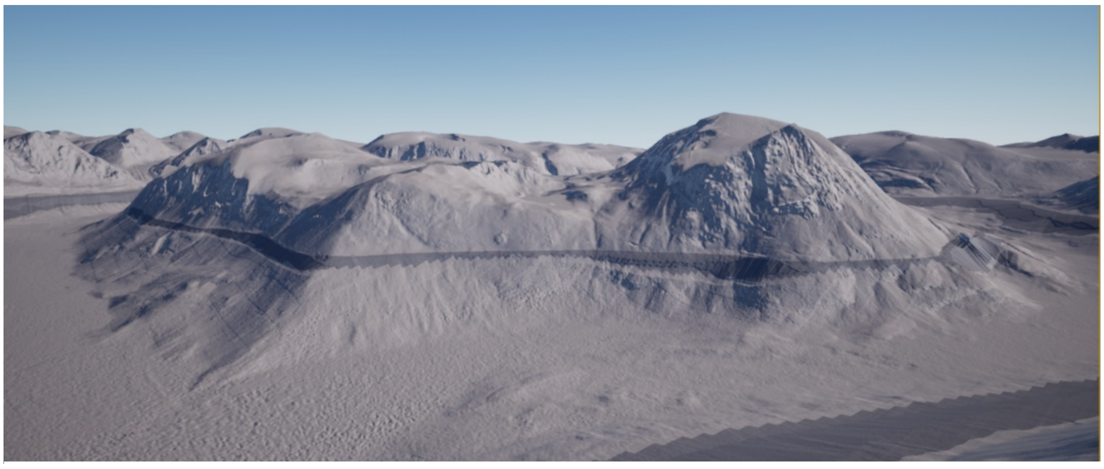


Figure 5-5: Landscape scenery in UE4 without the water surface depicting the Hillshade colouration for both the terrestrial as well as the bathymetry terrain. The interpolated dark grey transition zone is clearly visible

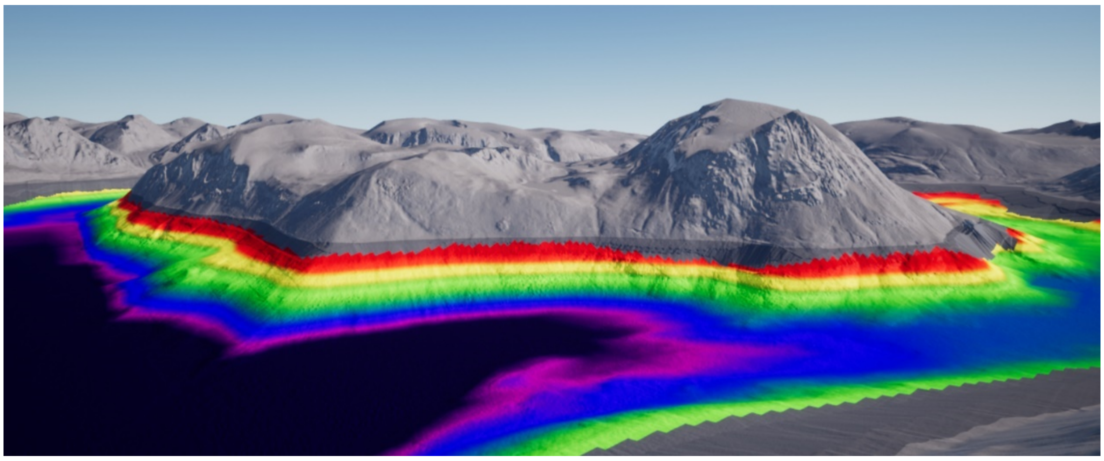


Figure 5-6: Landscape scenery in UE4 without the water surface depicting the Hillshade colouration for the terrestrial terrain and the depth related rainbow colouration for the bathymetry. In this presentation, the rainbow colour ramp stretches from -100 m until -450 m

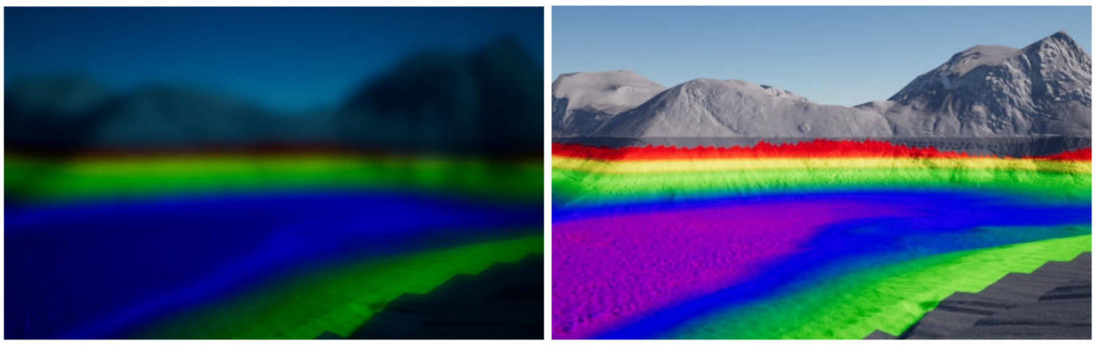


Figure 5-7: Left: Lightning effects under water with an activated water surface. Right: Lightning effects with an inactivated water surface which is the same as above the water surface

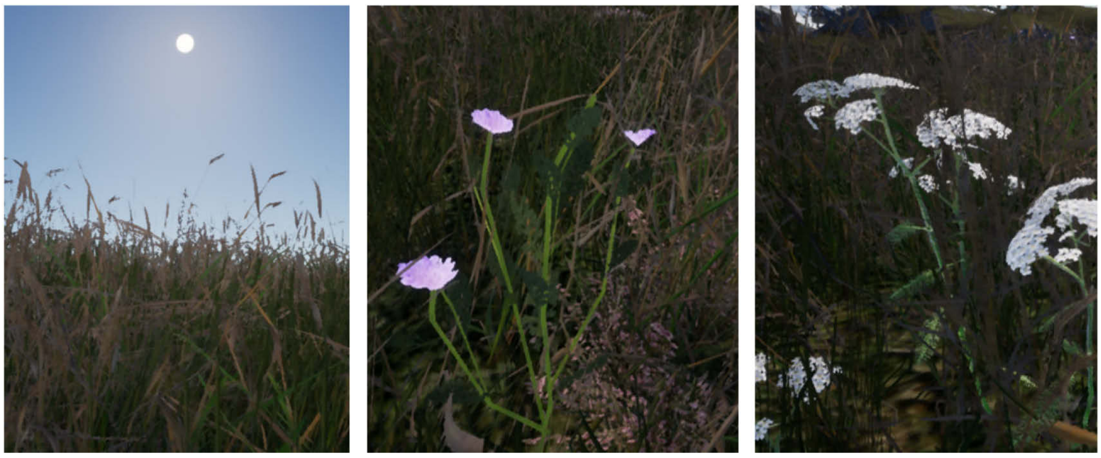


Figure 5-8: Left: Grass foliage and sun disc with bloom effect. Middle: Purple Alpine catchfly and common Heather foliage. Right: Northern Androsace foliage

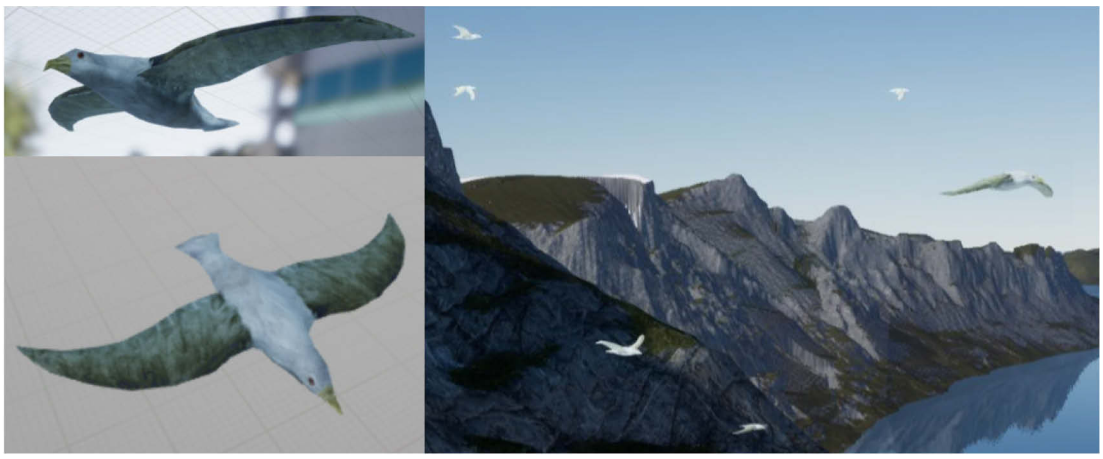


Figure 5-9: Seagulls up close (left) and distributed in the world in UE4 (right)

5.2 Motion Controller and Locomotion

Figure 5-10 shows both motion controllers in VR as the user is holding them in real life. Hands or other parts of the user's body, however, are not represented in VR. The implemented functions are described in the following. The right motion controller is identified by a pinkish logo, whereas the left controller utilises an orange logo. The right controller holds the flying and teleportation locomotion display, while the left controller holds the elevation, distance and compass measurement display. As the user hovers with the left controller and the attached red laser over the landscape, the elevation of and the distance to the hovered spot are updated in the display. The compass indicates the user's rotation which points now westwards.

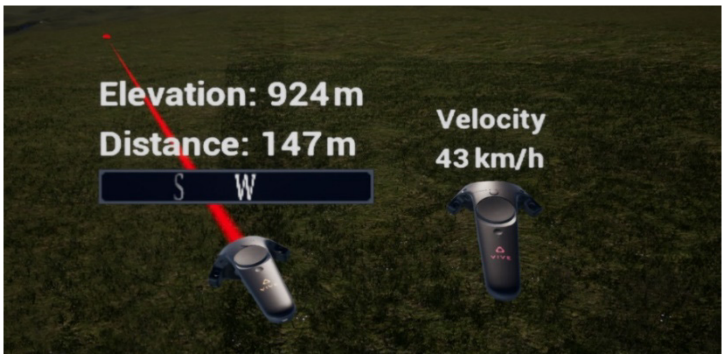


Figure 5-10: The left and right motion controller in VR. The left controller holds the elevation, distance and compass display whereas the right controller holds the flying and teleportation display

While switching between the flying and teleportation movement, the locomotion display also changes. The flying displays indicates the velocity of the movement in km/h (Figure 5-11 (left)). On the other hand, the teleportation display specifies the distance to the proposed location. An arc to the indicated location is formed while the right trigger is pushed. When the trigger is released the user jumps to the location of the arc endpoint (Figure 5-11 (right)).

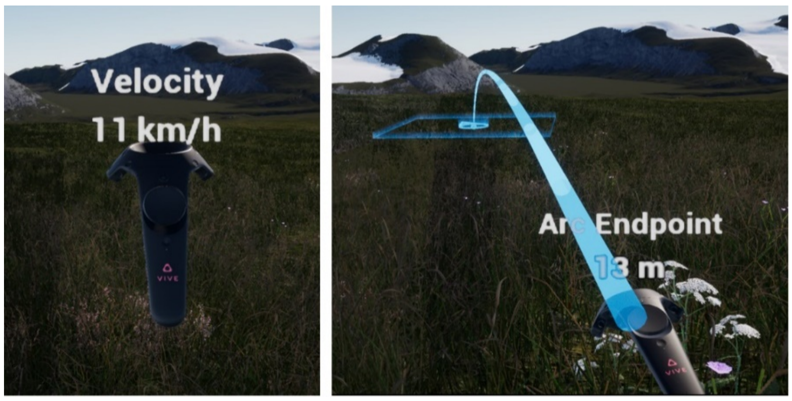


Figure 5-11: Left: Flying display showing the velocity in km/h. Right: Teleportation display showing the distance to the arc endpoint

5.3 Measurements

The height difference, slope distance and slope measurements between two measurement points are depicted in Figure 5-12. To visualise which point has a higher elevation than the other one, a plus and minus symbol have been introduced. The measurements can be executed between any arbitrary points. However, when a new measurement is being initialised, the display of the previous measurement vanishes.

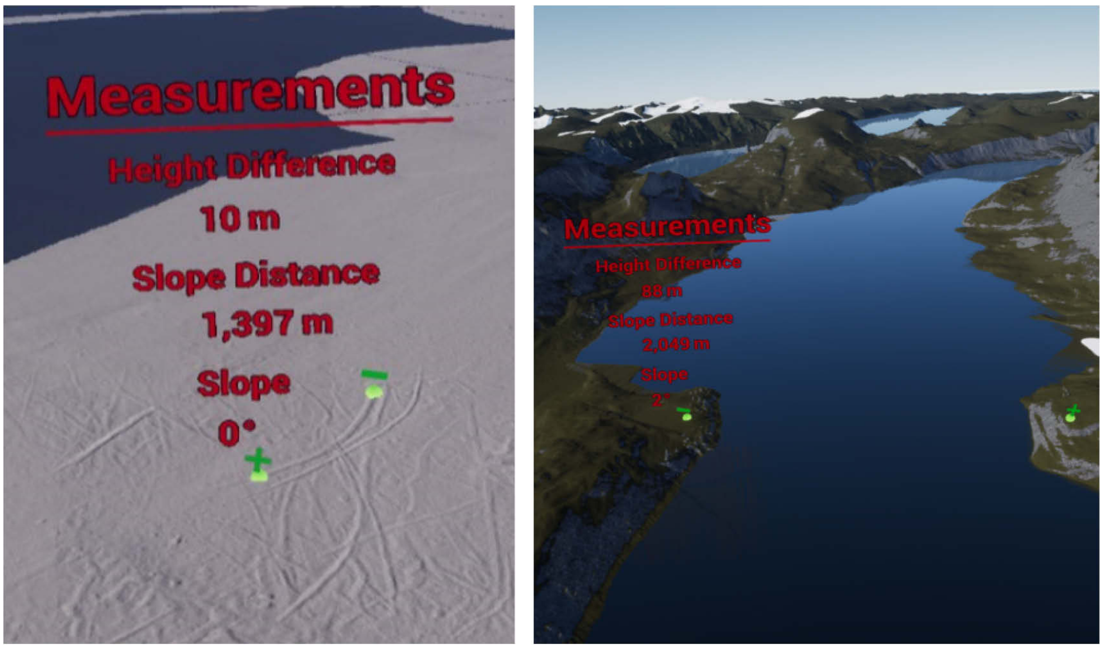
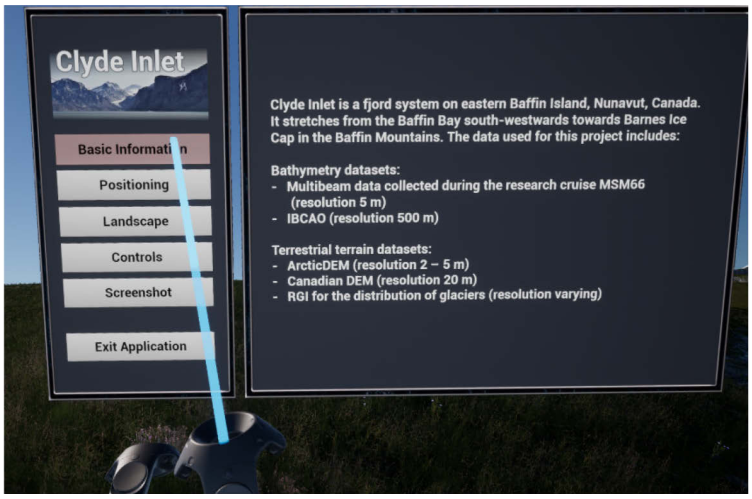


Figure 5-12: Height difference, slope distance and slope measurements between two measurement points. Left: along an iceberg scour and right: across the fjord

5.4 Menu

The two-piece menu is attached to the left motion controller and the right controller is used to navigate within the menu. As the menu is opened, the user is not able to execute movement anymore. Moreover, both controller displays and the red laser are disabled and a light blue laser on the right controller shows up. The first menu layout holds six main options which when selected open the second menu layout. The second right menu layout distributes the respective sub options or information. The following figures introduce the menu pages.



Clyde Inlet is a fjord system on eastern Baffin Island, Nunavut, Canada. It stretches from the Baffin Bay south-westwards towards Barnes Ice Cap in the Baffin Mountains. The data used for this project includes:

Bathymetry datasets:
- Multibeam data collected during the research cruise MSM66 (resolution 5 m)
- IBCAO (resolution 500 m)

Terrestrial terrain datasets:
- ArcticDEM (resolution 2 – 5 m)
- Canadian DEM (resolution 20 m)
- RGI for the distribution of glaciers (resolution varying)

Figure 5-13: Menu page “Basic Information” provides information of the fjord and utilised datasets

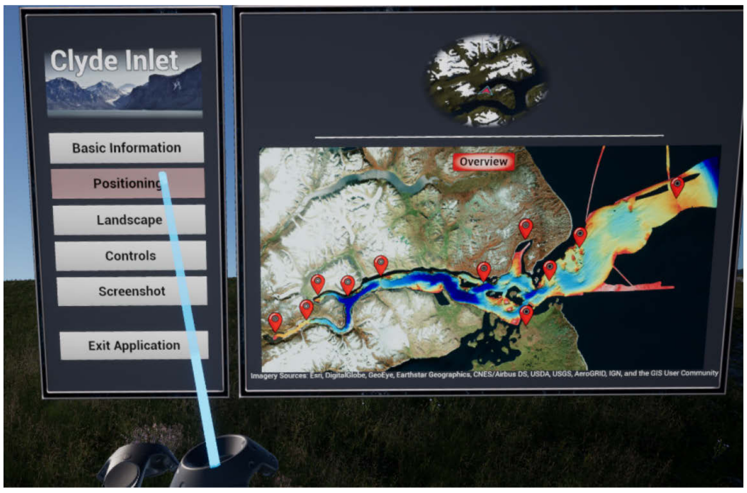
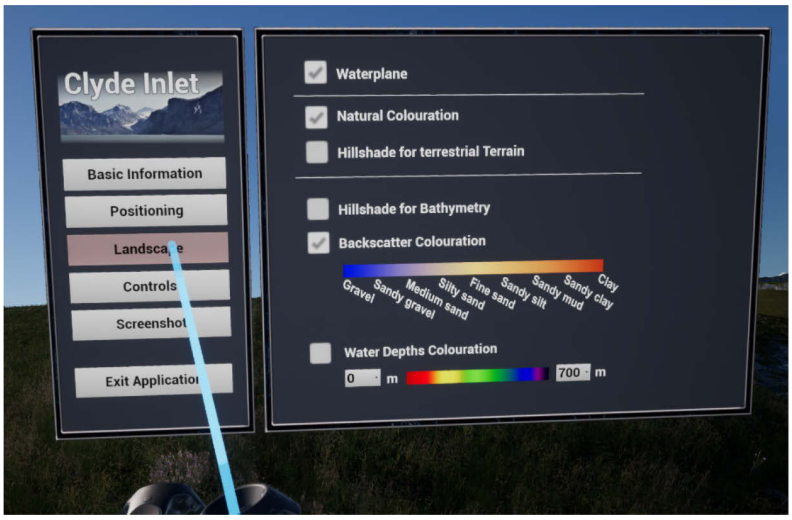

Clyde Inlet
Basic Information
Positioning
Landscape
Controls
Screenshot
Exit Application
Overview
Imagery Sources: Esri, DigitalGlobe, GeoEye, Earthstar Geographics, CNES/Airbus DS, USDA, USGS, AeroGRID, IGN, and the GIS User Community

Figure 5-14: Menu page “Positioning” holds the fast travel navigation system (bottom) as well as an aid to orientation (top)



✓ Waterplane

✓ Natural Colouration

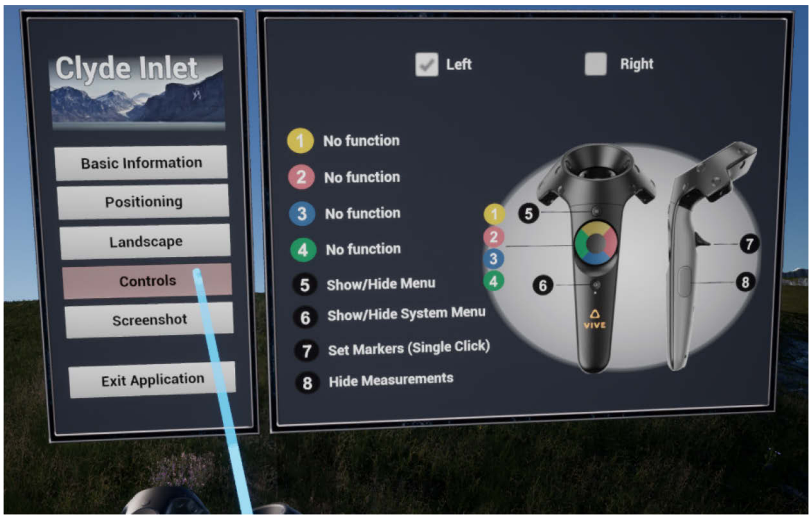
☐ Hillshade for terrestrial Terrain

☐ Hillshade for Bathymetry

✓ Backscatter Colouration

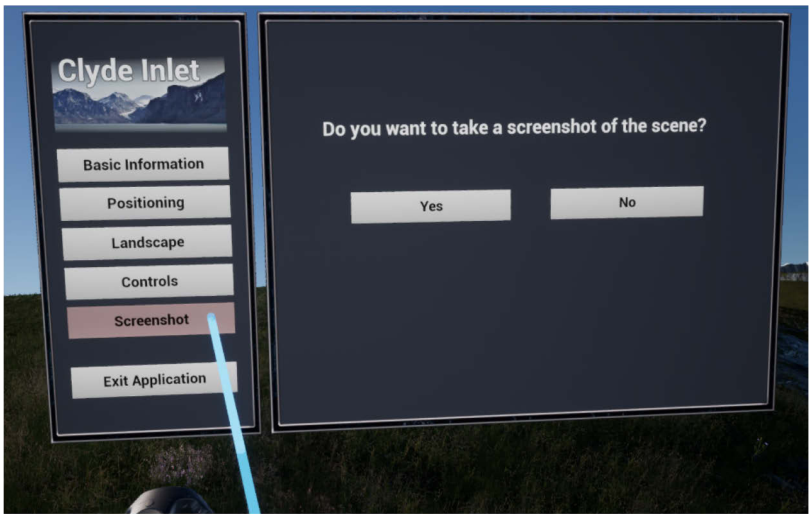
☐ Water Depths Colouration

Figure 5-15: Menu page “Landscape” holds checkboxes to toggle the visibility of the water surface and to change the landscape colouration



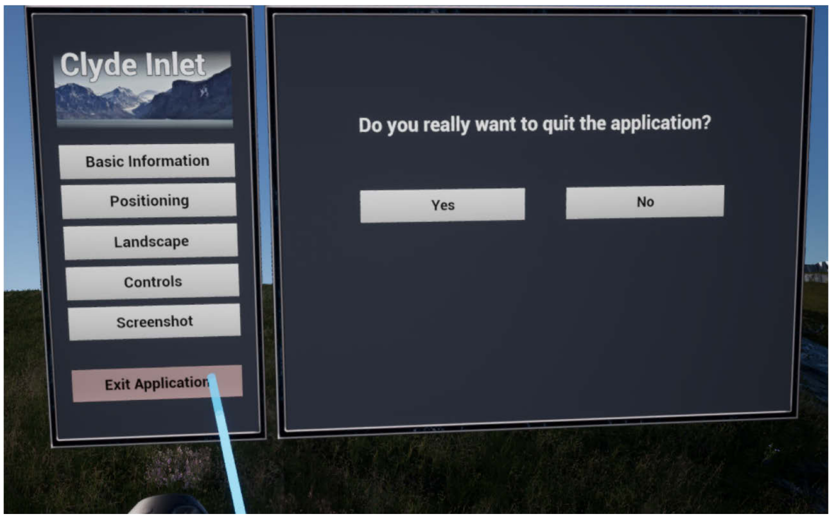
1 No function
2 No function
3 No function
4 No function
5 Show/Hide Menu
6 Show/Hide System Menu
7 Set Markers (Single Click)
8 Hide Measurements

Figure 5-16: Menu page "Controls" illustrates the key bindings for the left and right motion controller



Do you want to take a screenshot of the scene?

Figure 5-17: Menu page "Screenshot" offers the possibility to take a screenshot



Do you really want to quit the application?

Figure 5-18: Menu page "Exit Application" offers the possibility to exit the application

5.5 System Requirements

The application was tested on two desktops and one laptop computer with varying CPUs and GPUs. Since only three computers were tested, the following recommendations should only be used as rough indications rather than guidelines. Table 5-1 states minimum and recommended requirements for the CPU, GPU and RAM (Random-Access Memory). Unfortunately, one of the desktops could not run the application properly. The deployed Intel® Xeon® E5540 with 2.53GHz took about 15 min to compile shaders. Shaders have to be compiled when the project opens, the application starts and when the user navigates to distant waypoints and several new levels have to be loaded. Thus, a smooth running is not fulfilled.

The minimum requirements were adopted from the laptop which had an extremely high CPU usage during the VR gameplay. Nevertheless, the application was running even if not quite as smoothly as with the other desktop. The loading of tiles after the user navigated to a distant waypoint, took more than 10 seconds. Unlike the laptop, the desktop took only approximately 4 seconds to load the landscape tiles. The minimum requirements for the graphic card are the same as specified by the HTC Corporation (Gepp, 2017). The best performance is achieved with one of the desktops whose system specifications are thus classified as recommended requirements.

Hardware	**Minimum Requirements**	**Recommended Requirements**
CPU	Intel® Core™ i7-6700HQ 2.60 GHz (Turbo: 3.50 GHz)	Intel® Core™ i7-6700K 4.00 GHz (Turbo: 4.20 GHz)
GPU	NVIDIA GeForce™ GTX 1060	NVIDIA GeForce™ GTX 1080
RAM	16 GB	16 GB

Table 5-1: Minimum and recommended system requirements for the application

6 Discussion

This section discusses the results of the virtual reality application presented in chapter 5. The first subchapter examines the accuracy of the terrain while the second subchapter addresses the challenges regarding the performance of the application. In the end, the usefulness will be evaluated through interviews with users from six different career fields. The usability as well as the utility considering advantages, limitations and improvements will be analysed. Finally, a brief list of potential applications for virtual reality environments for various disciplines will be presented.

6.1 Consideration of Accuracy

The horizontal and vertical accuracy depends on the datasets being implemented. Every dataset has been conducted using different instruments and system setups. Therefore, an overall accuracy can hardly be accessed since it is varying with each dataset. Nevertheless, approaches to determine the accuracy of the MBES dataset as well as the processed terrestrial dataset have been made in sections 3.2.5 and 3.3.2.

With regards to the MBES dataset, the accuracy yields at best 1.2 m in mean water depth taking into account the system specifications of the echosounder and the positioning devices. However, rarely are the specifications stipulated by the manufacturers met in real world applications. Also, a lot more factors influence the accuracy of a measured point such as the accuracy of the sound velocity determination, the sensor offset in the vessel coordinate system, latency error estimates or tide errors. All error sources and uncertainties add up to the TPU of a surveyed point and would result in a much higher value than 1.2 m. As stated before, due to missing specifications the TPU could not be computed.

With regards to the terrestrial terrain, the accuracy varies between 4 m to 15 m with respect to the used DEMs. Transition zones between one and the other DEM are thus in part strongly visible. To smooth the edges, the Focal Statistics tool can be applied. However, this tool smoothens the dataset just globally and consequently changes the elevation of places which are not even affected by harsh transition zones. If the accuracy of the terrain altitude is less important than a coherent terrain, the Focal Statistics tool might be the solution. However, if the accuracy should be maintained as good as possible every additional processing step should be scrutinized. A local interpolating tool would be the key to the problem though it would end up in a lot of manual work. In the end, harsh transition zones and other artefacts of the terrestrial dataset were not diminished nor averted to maintain the global accuracy as good as possible.

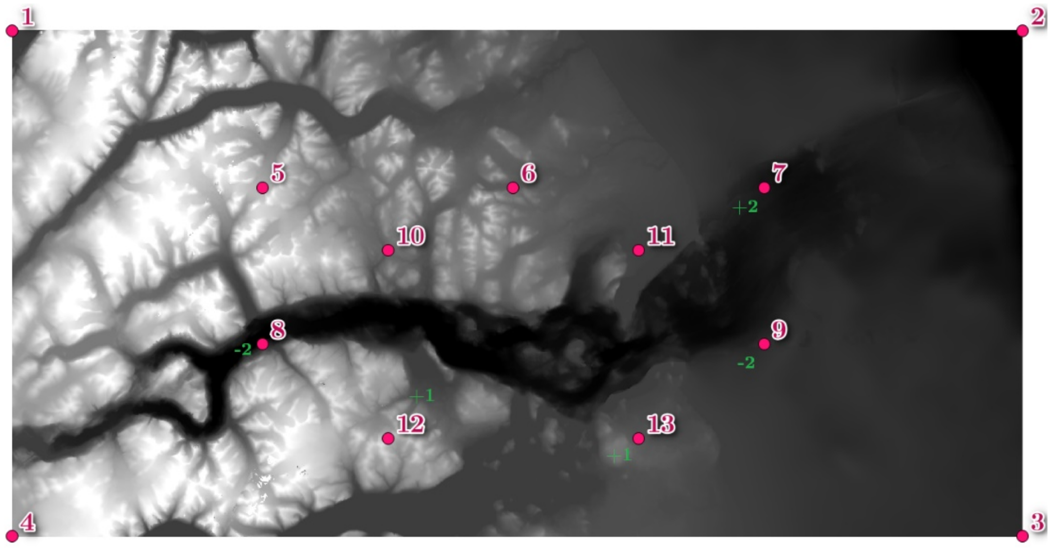


Figure 6-1: Distribution of height measurements (pink) and height differences (green) between the DEM and the landscape in the VR application

Furthermore, several elevations before and after the import into UE4 were measured to detect potential accuracy losses. In total, thirteen elevations were measured as seen in Figure 6-1. 8 out of 13 locations yield the same elevation before and after the import. However, 2 locations are 1 m higher and 3 locations possess a height difference of 2 m (refer to the appendix Table 9-1). Possible reasons for the discrepancies are an inaccurate conversion to the *r.16 format by L3DT or an erroneous scaling during import. Most likely, the latter induced the elevation differences. Since the width and height of the landscape is calculated as indicated in the following equations, the results of both will never receive an exact ratio of 2:1:

$$X\ extent: (16 * 2017 - 15) * 5\,m = 161285\,m \tag{6-1}$$
$$Y\ extent: (8 * 2017 - 7) * 5\,m = 80645\,m \tag{6-2}$$

*(Extent: (number of tiles * number of pixels – (number of tiles-1) * resolution))*

The cause lies in the architecture of the landscape. For each tile one pixel has to be subtracted meaning that if 16 tiles should be used, 15 pixels have to be subtracted and if 8 tiles should be used, 7 pixels have to be subtracted. This results in an inaccurate ratio. Dividing the width by the height, a ratio of 1.999938 results. This leads to a slightly different scaling value of 0.015 which might be the reason for the elevation variations of 1 m – 2 m. For further projects it is recommended to use either square landscape formats or just one tile in total. Resulting drawbacks, however, are a decrease in performance for large terrains if only one tile is used or if the landscape has to be created larger than necessary to maintain the square shape.

The accuracy of the conceived measurement tools such as elevation, slope distance and slope measurements have been tested on several objects. All measurements are rounded and accurate to one meter.

6.2 Consideration of Performance

In addition to the degree of illusion and immersion, the performance is a key factor for a pleasant VR experience. Because a low degree of performance, does not only frustrate the user, but can also cause visually induced motion sickness so called cyber sickness. Apart from physically induced motion sickness, cyber sickness occurs due to visual motion alone and can be stopped by closing the eyes. Dropping frames and motion tracking lag are two crucial effects caused by low performance that can lead to symptoms of nausea, dizziness, headache, sweating, and in worst cases vomiting (Jerald, 2016).

The performance is typically measured in either frame rate or milliseconds, although milliseconds is mostly the preferred measure due to the fact that the frame rate behaves highly non-linear and can thus slant the perception of performance (Dunlop, 2003). A VR application should run ideally at a frame rate which complies with the refresh rate of the HMD yielding 90 Hz regarding the HTC Vive (refer to 2.2.2). In any case, frame rates should exceed 60 FPS (16.7 ms) for stereoscopic images in order to view a continuous flow (Fuchs, 2017).

To emphasize how demanding VR systems are, an ordinary application that runs with 60 FPS on a screen with a resolution of 1920 x 1080 pixels will produce 124 million pixels per second. A VR system, however, needs to render two screens at resolutions of 1080 x 1200 each adding up to 2160 x 1200 pixels. Because the glasses are close to the human's eyes, lenses that distort the images are mounted in order to obtain a high field of view and to focus objects far in the distance. To counteract lens distortion, the software renders images with the opposite barrel distortion. Admittedly, barrel distortion loses some of the perceived resolution of the image. Therefore, UE4 renders 1.3 – 1.5 times the display's resolution. In total, using stereoscopic images and a factor of 1.5, VR systems produce at 60 FPS 233 million pixels per second and optimally at 90 FPS 350 million pixels per second (McCaffrey, 2017).

Because VR systems are so demanding and low performances have such a negative effect to the user, maintaining a good performance is thus a great challenge for the developers and designers of VR applications. To make VR experiences furthermore accessible to a wide audience, an application needs to properly work also on low or mid-end hardware. This provokes a trade-off between performance and visual representation which needs to be carefully weighted.

Regarding this thesis project, a great challenge was to work with a terrain size of 160 km x 80 km. The highest resolution at which the landscape can be represented amounts to 5 m yielding more than 1040 million triangles if visualised at full resolution in UE4. This would be impossible to handle for the university's best computer available to develop this thesis project (see section 5.5 for system specifications).

One way to cope with this large landscape is to reduce the resolution. However, this causes lack of details which would decrease the user's experience and the utility of working with the terrain.

On the other hand, the landscape can be split into what is called streaming levels to only load in parts of the world in detail that are actually needed. In order to choose a level size, one has to consider the performance trade-off between component size and the total number of components. Small component sizes allow quicker LOD transitions and enable the occlusion of more terrain. But smaller component sizes require more components which necessitates more draw calls and thus increases the overall CPU cost. Epic therefore recommends a maximum of 1024 components and proposes several landscape sizes that maximize the area while minimizing the number of components (Epic Games, Inc., 2018h). From these recommendations, three landscape sizes where chosen and tested as illustrated in the following table with specifications related to the terrain of Clyde Inlet.

Size of level [pixel]	**Size of level [km]**	**Number of levels**	**Number of components**	**Components per level**	**Size of component [m]**
1009 x 1009	~ 5x5	450 (30x15)	28,800	64 (8x8)	~ 20x20
2017 x 2017	~ 10x10	128 (16x8)	32,768	256 (16x16)	~ 20x20
8192 x 8192	~ 40x40	8 (4x2)	8,192	1024 (32x32)	~ 40x40

Table 6-1: Specifications of three landscape sizes related to the terrain of Clyde Inlet

The largest recommended size yields 8192 x 8192 pixel and comprises a component size of 40 x 40 meters. It is thus twice the size than the other two options which both yield component sizes of 20 x 20 meters. In return, the number of components is much less for the 8192p level which - one might think - could reduce the CPU cost. In fact, utilizing a level size of 8192 x 8192 pixel will reduce the variability of the streaming distance, meaning that big parts of the landscape are always loaded in full detail which reduces the overall performance. The smaller the level size, the greater is the area of the terrain that can be unloaded. The tested frame rate yields therefore 14 FPS (71 ms) which is in any case insufficient.

While considering whether a level size of 1009 x 1009 or 2017 x 2017 is more adequate, the desired streaming distance has to be considered. The greater the streaming distance, the greater is the area of detailed terrain that is loaded and thus visible. Now, since the level of detail for this project could not be adjusted automatically and since the reduction of triangles is fixed to 99.97%, a streaming distance of 10x10 km seemed to be appropriate. Of course, it is possible to manually set several degrees of triangle reductions and this would be the ideal way, but it would also drastically increase the workload and was thus not accomplished. The streaming distance of 10 x 10 km proved to be the right distance at which the user would barely spot the difference between high and low detailed terrain.

Consequently, both level sizes were tested with the above-mentioned streaming distance and each yield a frame rate of 44 FPS (22 ms) while navigating through the world. As stated before, in order to see a continuous flow at least 60 FPS should be obtained as proposed by Fuchs (2017). Even though the recommendations are not met by both options, stuttering or other flickering effects could not be identified while navigating slowly through the terrain. However, when the speed increases, stuttering is noticeable each time a new level is loaded, or the world origin is shifted. The terrain that is made up of smaller level sizes caused a higher frequency of stuttering due to the fact that more levels are constantly loaded and unloaded while moving. Therefore, in the end, the 2017 x 2017 size was chosen which stutters only occasionally and only when navigating at high altitudes or with great velocity. Motion sickness or any other form of discomfort caused by the stuttering could not be identified by various test persons.

Further approaches implemented to maintain the performance are shortly listed in the following. At first, the automatic LOD system by UE4 is used which radially reduces triangles of the landscape with the distance of the user. Figure 6-2 presents six grades of details for all loaded levels starting with the highest LOD represented by white colour and ending with the lowest LOD represented by pink colour. Unfortunately, no documentation about the degree of reduction could be found. All landscape parts which are visualised beyond the streamed in levels (and therefore possess the triangle reduction of 99.97%) are represented by cyan and thus hold the least level of detail.

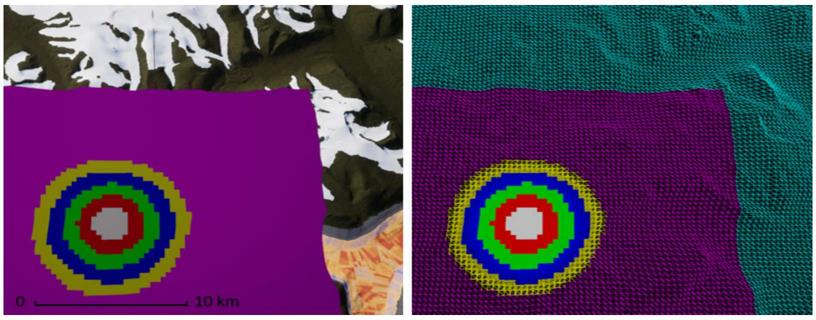


Figure 6-2: Overall LOD system including automatic LOD generation by UE4 for all loaded Levels (white, red, green, blue, yellow and pink) and manual LOD generation for all landscape parts beyond loaded Levels (cyan)

Moreover, each foliage contains at least 3 LODs which need to be automatically computed relative to the distance of the user. Also, the cull distance had to be kept small. Additionally, costly tools such as the overview map are only computed when really needed.

In the end, two rendering techniques were scrutinised to further optimize the application by just changing the way a frame renders. To render a frame, many pieces work together. In broad terms, the main tasks for the CPU are to query the user input and to submit draw calls to be carried out by the GPU. The GPU, hence, renders the draw calls and displays the image on the screen. An image, however, exists of many vertices which are grouped to triangles and later on shaded to obtain a realistic world. The pipeline of how vertices are processed to create the final image can be altered regarding two main methods: forward versus deferred rendering. The more common approach is the forward rendering technique. Simply put, a vertex shader firstly takes all vertices of a geometry and turns them into an appropriate form for the fragment shader. The fragment shader then calculates the final colour of the pixels/fragments. But the drawback is that lights are even applied to fragments that are hardly affected by a particular light source. Also, geometries will be shaded just to find out that they will be covered by other geometries and thus be invisible to the user.

On the other side, the deferred rendering technique gathers all geometries and fragment descriptions in so called G-buffers first and shades fragments later in a combined approach. In this way, covered objects will not be rendered. Furthermore, this method provides fancy techniques regarding reflections and shadow effects which certainly can be very demanding (McCaffrey, 2017).

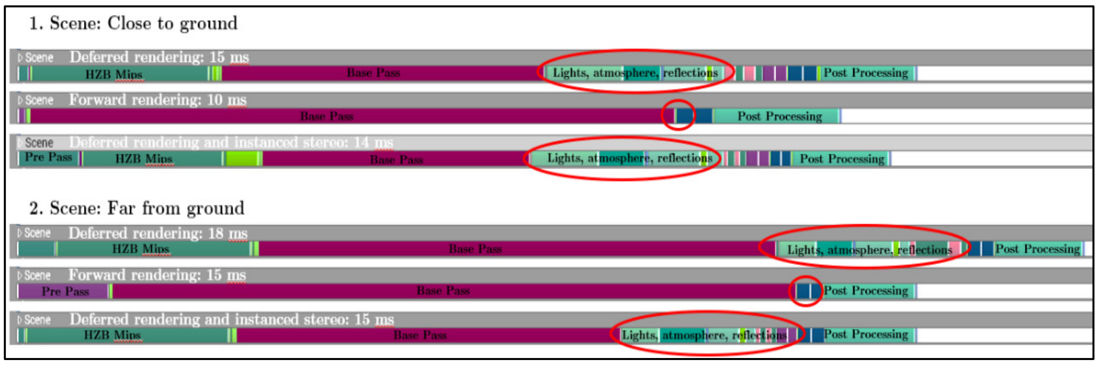

1. Scene: Close to ground
Scene Deferred rendering: 15 ms
HZB Mips
Base Pass
Lights, atmosphere, reflections
Post Processing
Scene Forward rendering: 10 ms
Base Pass
Post Processing
Scene Deferred rendering and instanced stereo: 14 ms
Pre Pass
HZB Mips
Base Pass
Lights, atmosphere, reflections
Post Processing
2. Scene: Far from ground
Scene Deferred rendering: 18 ms
HZB Mips
Base Pass
Lights, atmosphere, reflections
Post Processing
Scene Forward rendering: 15 ms
Pre Pass
Base Pass
Post Processing
Scene Deferred rendering and instanced stereo: 15 ms
HZB Mips
Base Pass
Lights, atmosphere, reflections
Post Processing

Figure 6-3: GPU cost of various passes (PrePass, Base Pass, Lights, Post Processing) for two scenes with three different rendering techniques. The first scene is close to the ground whereas the second scene depicts an overview of the entire world. The first rendering method is the deferred technique followed by the forward technique and the combination of deferred and Instanced Stereo technique

While testing the forward rendering technique, frames were rendered three to five milliseconds faster than with the deferred method (refer to Figure 6-3). When investigating the GPU cost of the various passes in a frame it turns out that the forward rendering technique does not contain heavy light passes such as shadows and reflections (see red circles in Figure 6-3). This is also visible when running the application. There are no reflections on the water surface nor shadows from high mountain peaks. For this reason, the deferred method was implemented even though the performance is slightly worse. To account for the performance loss, the Instanced Stereo tool was enabled which draws images for both eyes at the same time rather than having them rendered independently (McCaffrey, 2017). The number of draw calls submitted by the CPU is hence reduced resulting in an increased performance on the GPU side by one to three milliseconds.

6.3 Consideration of Usefulness

This section covers considerations regarding the usefulness of virtual reality applications based on the in this thesis developed immersive visualisation of Clyde Inlet. As defined by Nielsen (2012), the main goal of a user-experience design is to have a high usefulness. To achieve this, the application must not only provide a high usability but also a high utility. While the usability examines the ease of use and user-friendliness of an interface, the utility assesses whether the interface is beneficial and solves a real user need (Nielsen, 2012).

Against this background, the utility was tested for various field of works to discover the benefits as well as the limitations of virtual reality visualisations specific to each surveyed occupation. Another aim was to identify improvements and potential applications and to determine whether virtual reality would in fact be applied in the everyday working life.

The analysis was conducted on the basis of interviews with ten participants each representing one of the following occupations: hydrography, geology, geography, archaeology, civil engineering and lastly tourist industry. The interviewees have a variety of functions such as professor, doctoral candidates, "Diplom" engineer, general manager and come from different universities and authorities. Each participant had time to use the VR application for 20 – 30 minutes and would afterwards answer four questions regarding the usability and six questions considering the utility (refer to Figure 9-60 for the questionnaire).

6.3.1 Usability of VR Application

The handling and user-friendliness of the VR application were perceived as overall pleasant as seen in Figure 6-4. All of the interviewees found the application easy to use, although some participants found it at the beginning slightly less easy due to their inexperience with VR systems. But once all functionalities have been learned, the application was referred to as “very user-friendly”. Additionally, everybody received a good sense of orientation by checking the compass as well as the overview map. However, some interviewees preferred to have the overview map visualised continuously without having to open the menu. Also, the overview map was considered by one participant slightly too small. With regards to motion sickness, no participant felt uncomfortable while navigating through Clyde Inlet. One participant felt slightly uneasy while flying very quickly close to the ground. Another interviewee felt a bit insecure when looking down a cliff. In conclusion, the VR application was considered as user-friendly and comfortable to use. Serious cases of motion sickness or any form of discomfort could not be discovered.

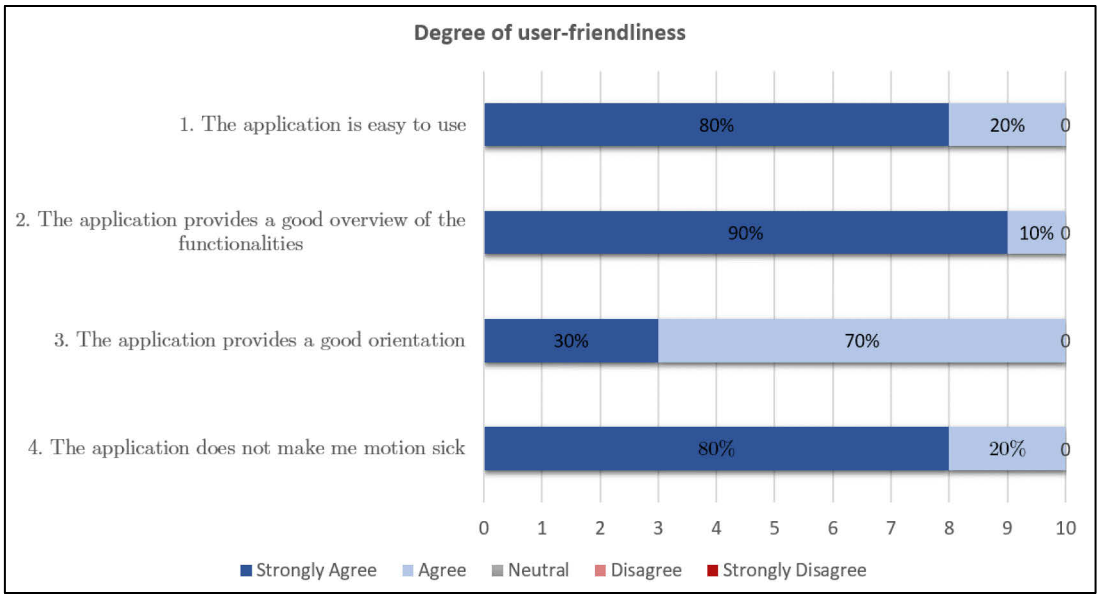


Figure 6-4: Degree of user-friendliness of the VR application accessed using a five-tier Likert scale

6.3.2 Utility - Advantages, Limitations and Potential Applications

Advantages

The advantages of the VR application are manifold and diversely to each professional field. Yet, for all disciplines the VR application enabled a better impression of the terrain data compared to a 2D presentation. Moreover, the linkage between bathymetry and terrestrial terrain was considered by everyone helpful as the terrain is regarded as a whole and would provide additional information to the orientation. Also, all functionalities and measuring tools seem to help the viewer get necessary information from the terrain and are useful to seize the dimensions.

Taking a look at the field of hydrography, the VR visualisation was considered beneficial to directly identify “ship wrecks, piles of stones or other hazardous obstacles”. Since outliers can be very easily determined even if they are very small, such an immersive 3D visualisation would be also suitable for certain forms of quality analyses. Moreover, lots of time could be saved when outliers or hazardous material can be easily identified. Additionally, it was stated that a VR visualisation would be preferred, when just a quick impression of the data shall be received, “because in a VR application one would not have to scroll in and out a lot which can be very time consuming”. However, one participant remarked that for very elongated terrain sizes a general idea of the terrain is difficult to obtain.

Furthermore, the VR application was considered helpful for interpreting the terrain, especially regarding backscatter analyses. The combination of backscatter results with the elevation as the backscatter image is draped above the 3D terrain would yield additional information: “As an example: correlations between backscatter intensity and slope of the terrain could be examined in an easy way”.

Some hydrographers stated that the application would also be beneficial to support the exchange between scientists such as biologists, geographers, geologists and so forth. It could be helpful especially when scientists can communicate and work simultaneously within the application. This would yield a great advantage since many scientists live at different places in the world. Moreover, it was suggested that due to the 3D presentation many misunderstandings or misinterpretations can be avoided especially considering geological analyses.

As for hydrographers, also for civil engineers, advantages lie in the presentation of data. Since the presentation is very intuitive and self-explanatory, VR systems could be used to convince decision-makers or be displayed at civic participations when new projects are presented. One participant suggested that it might be helpful as an informational tool to inform the public about the exact proposed changes regarding the currently controversially discussed deepening of the river Elbe. Hence, misconceptions could be playfully eliminated. Moreover, the combination of bathymetry and terrestrial datasets are regarded useful when working with flood zones or the constructions of embarkments such as quay walls.

With regards to archaeologists, visualising the terrain as a whole is always then considered beneficial when questions related to the coastline sought to be answered such as “where ancient harbours might be located at the Red Sea?”. The location of sand deposits, shallows, quay wall relicts, wrecks as well as the morphology of the coastline are only a few of many more indications considered when looking for suitable sites.

Advantages of virtual reality visualisations are that archaeologist receive a better impression of sites and receive a feeling of dimensions which is not possible when investigating simulations. Moreover, VR is regarded as a cost-effective tool to reconstruct places or buildings when the original sites cannot be visited anymore because they either do not exist anymore, are located under water, within prohibited areas or when the trip is considered too dangerous or expensive. A virtual immersive reconstruction is, however, not only preferred over a physical reconstruction due to financially reasons but it also offers the possibility to investigate visual axes in a very easy way. Ancient visual axes are observed to check whether e.g. fortified towers can be seen from neighbouring towers or to verify a location that is bequeathed to be the last point before a community where vessels can be seen far in the distance entering the fjord's mouth. These are only two examples of possible hypotheses that are hardly checked using 2D visualisations or simulations. A benefit over physical 1:1 reconstructions are moreover that virtual applications can easily adjust e.g. weather conditions. Sites or buildings can thus be easily observed by rain, sunlight, fog, snow, night, moonlight or the like.

From a touristic point of view, virtual applications could be used to display trips, especially educational journeys. They could help with the explanation of e.g. the formation of fjords or just be simply used to visualise hiking trips. Potential travellers could receive a more in-depth perception of the journey. For example the guide of ship cruises can show his guests the visited underwater area without the need for diving.

Limitations and improvements

A great disadvantage over traditional software packages is the very limited possibility to import or export geospatial datasets. It is currently not possible to export other formats than plain *.png screenshots while running the application. Raster datasets can only be imported as *.png, *.bmp, *.jpg or *.jpeg but without spatial reference since no coordinate transformation algorithm is implemented. The only way to provide a sort of spatial link is to import raster datasets with the same extent, scale and position as the terrain in UE4. However, as mentioned in several chapters before, the greatest resolution is only 8192 pixels resulting in a very pixelated display when stretching the image over the entire terrain. On the other hand, to import or export vector datasets, UE4 provides only the formats *.csv and *.json. Nevertheless, it is always possible to manually place objects in the world with spatial reference since the coordinate origin of UE4 is by default placed at the top left corner of the terrain. Taking the X-Y distance from the top-left corner to a point and transferring the distances to another software (such as QGIS for example) one could access the coordinates. By far this would be very time-consuming and inefficient for most applications.

Applying a coordinate transformation algorithm is generally possible, nevertheless it will arouse quite a challenge since the origin of the coordinate system in UE4 changes after an arbitrary distance. To overcome this, one would have to use either a very small terrain of less than 5x5 km and disable world origin rebasing or initiate world origin rebasing manually which is very work-intensive especially for larger terrains.

Almost all interviewees regarded the import and export possibilities as insufficient for the work. Especially, the import of vector datasets comprising cores, pipelines or other constructions were considered necessary. Also, the ability to access coordinates are viewed as indispensable when certain parts of the landscape need to be examined, noted for other investigations or simply to navigate to specific points. But several participants stated that coordinates not necessarily need to be applied when the terrain just need to be analysed.

Further suggested improvements are to include also backscatter greyscale mosaics, seismic data as well as water column data. Additionally, all scientific disciplines requested cross profiles. For archaeologists, area and especially volume calculations are regarded to be highly beneficial since this information is used to determine how much soil needs to be dug out or filled back in of the investigated site. Another suggestion was to manually select the information to be displayed at the left motion controller to retrieve different information from the terrain such as e.g. decibel values of backscatter data.

Other limitations of the VR application are, inter alia, the transition zone between terrestrial and bathymetric data. An improvement would be to collect missing data via airborne LiDAR (Light Detection and Ranging) systems or smaller vessels if necessary. Also, the VR application is not yet ready to harbour multiple users simultaneously.

Potential applications

Potential applications for VR visualisations in the field of hydrography are backscatter analyses and quality assessment as outliers are nicely visible. Furthermore, it can be used for the presentation of wrecks or pipelines and possibly utilised for track planning. Moreover, it can serve as a platform to improve the exchange between scientists and be beneficial for geological interpretations. Additionally, 3D immersive representations are useful to give an insight into construction projects that can be used for civic participations.

For the field of civil engineering, a virtual reality application might be useful for inspections and maintenance of quay walls or be used for site selections for offshore wind parks “because you might be able to better visualise glacial channels in the North Sea”. It can be also suitable for the preparation and control of constructions since it provides a better overview especially when the project contains lots of details. Of particular use, is furthermore the ability to show/hide or change various construction designs for decision making processes.

In archaeology, 3D immersive visualisations can be used for reconstructions of places or buildings when a site cannot be visited. It is useful to check assumptions and solve questions such as “Why would Romans construct their weir system in an Egyptian valley which is surrounded by mountains from which it is easily attackable?”. Examining the terrain and visual axes play hence an important role.

For travel agencies, virtual reality might be beneficial for planning and preparing educational journeys. Also, it would be effective to showcase trips at fairs in order to attract potential customers.

In conclusion it can be said that all interviewees see the potential in virtual reality. Figure 6-5 provides a summary of key aspects. However, all participants could only imagine implementing it into their work under certain circumstances:

- When the import of data is easy and not so time-consuming
- When the VR setup is much simpler and would just require glasses and controllers
- When the export of geospatial information is possible and quick
- When depending on the specific tasks additional functionalities are implemented

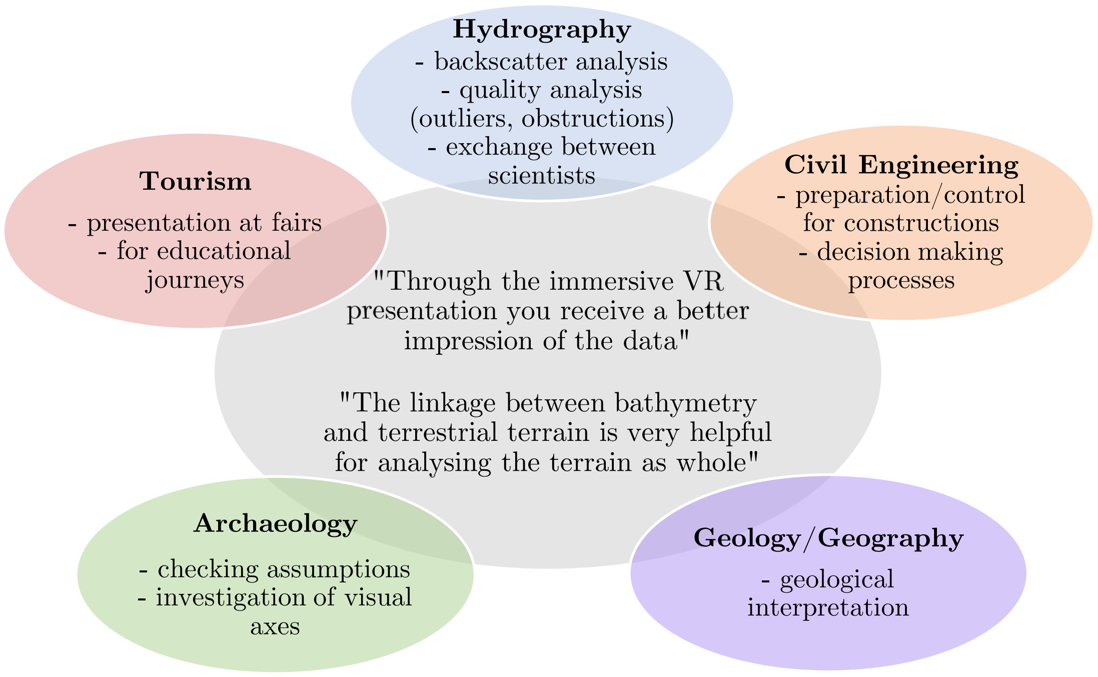


Figure 6-5: Summary of advantages and potential applications of VR for specific field of works

7 Conclusion and outlook

The objective of this thesis was to develop a virtual reality visualisation of the water body Clyde Inlet located on eastern Baffin Island, Canada by combining terrestrial and bathymetric terrain data. On the basis of this approach, questions regarding the accuracy and performance of this visualisation were examined. Furthermore, the combination of bathymetric and terrestrial datasets was evaluated in terms of possible benefits. Since 3D immersive visualisations are still novel to various scientific and economic fields, a brief study has been conducted to determine the potential of virtual reality based on representatives from six different occupational areas. In the end, possible applications were proposed in view of each specific field.

Throughout the thesis project, a method for visualising vast, high resolution terrain in virtual reality has been developed. To create a consistent elevation model above and beyond the water line, several datasets of different resolutions and projections have been utilised. These include excerpts from the ArcticDEM and CDEM covering the terrestrial terrain as well as bathymetric records obtained during the research cruise MSM66 and provided by IBCAO. The combination of all datasets yields the major component of the virtual reality application – a 160 km x 80 km large landscape which can be observed using three different ways of locomotion. Further results are the ability to conduct distinct measurements as well as the alteration of different terrain textures.

Over the course of several test measurements, no major loss of accuracy could be revealed throughout the entire processing chain. Nevertheless, squared terrain sizes or single terrain tiles can be recommended for small datasets as they prevent slight inaccuracies while importing terrain data into the software Unreal Engine 4. However, with respect to great datasets yielding vast territories of high resolution, these options are not advisable in terms of performance costs. Since poor performance is a major contributor to motion sickness, a combination of highly detailed streamed in landscape tiles and low detailed distant landscape parts were proven necessary if great datasets ought to be used. Moreover, if special lightning or shadow effects are not required, the forward rendering method shall be recommended.

Despite noted minimum requirements of 60 FPS by Fuchs (2017), a framerate of 44 FPS was proven sufficient for virtual reality applications that do not necessitates fast movements such as the in this thesis developed project. On the basis of user statements, no motion sickness nor any other form of discomfort could be discovered. However, no long-term evaluations over several hours have been observed which might provide opposite results. In the end, 90 FPS should always be aimed for future applications as advised in standard literature (Jerald, 2016). Overall, the application was considered user-friendly.

With regards to the utility of virtual reality visualisations, findings of the user survey demonstrate an increased perception of terrain data compared to a 2D presentation. With VR, it is possible to obtain a great overview and observe the terrain from different perspectives. Moreover, dimensions are more easily seizable as the user is able to freely navigate around and measure distances. Another advantage over 2D presentations is the lack of misguided assumptions that arise from a reduced dimensionality. Especially, slope related investigations can be unambiguously examined, and outliers or hazardous obstacles are nicely detectable. Likewise, the combination of bathymetric and terrestrial datasets is especially beneficial for geological interpretations of landforms, archaeological inquiries of coastal and river zones and generally yields an increased orientation.

In view of the above, several potential applications emerge for the following specific field of works. For hydrography, virtual reality visualisations are particularly helpful for combining several datasets with the elevation of the terrain such as backscatter analyses. Also, it can be used as an addition to quality checks and track-planning. Furthermore, geographers and geologists might use VR for geological interpretations of landforms whereas archaeologists find use in the reconstruction of sites to examine the terrain and to investigate visual axes. Civil Engineers might visualise several construction designs for decision making processes or use VR for site selection and planning of constructions such as offshore wind parks. The tourist industry could use immersive visualisations to show cast journeys at for example fairs to attract potential customers. Lastly, virtual reality could possibly be used to support the exchange between scientists especially when they live all around the world.

However, still a number of limitations need to be overcome to arrive at a fully operational virtual reality application that could affectively be applied in the everyday working life. Most importantly, the import and export of geospatial data need to be improved and a spatial reference system need to be developed. Another disadvantage is the time-consuming workflow of the project as well as tedious hardware setup. Nevertheless, the latter shall be solved in nearby future as current developments aim at wireless headsets, non-peripheral sensory systems (besides glasses) and eye tracking systems (Jerald, 2016).

In conclusion, with the aid of the developed application, the potential of virtual reality for various occupations could be successfully demonstrated. There will be no doubt that virtual reality finds use in future applications when the above-mentioned limitations are to be overcome.

8 Bibliography

Briese Schiffahrts GmbH & Co. Kg. (2017). *Maria S. Merian - Handbuch*. Retrieved January 30, 2018, from https://www.ldf.uni-hamburg.de/merian/technisches/dokumente-tech-merian/handbuch-msm-deutsch-20171127.pdf

Bruey, D., & Ciuffo, M. (2017). *Sneak Peek: Designing Tracked Objects for Steam VR.* (M. Denno, Editor) Retrieved May 15, 2018, from https://www.slideshare.net/MorganDenno/sneak-peek-designing-tracked-objects-for-steam-vr

Buckley, S. (2015). *This Is How Valve's Amazing Lighthouse Tracking Technology Works*. Retrieved May 15, 2018, from https://gizmodo.com/this-is-how-valve-s-amazing-lighthouse-tracking-technol-1705356768

CARIS. (2013). *User Guide CARIS HIPS and SIPS 8.1.* Retrieved February 02, 2018, from https://spaces.awi.de/confluence/download/attachments/29492612/CARIS_HIPS_and_SIPS_User_Guide_v81.pdf

Den, L. (2015). *World Machine to UE4 using World Composition*. (Epic Games, Inc.) Retrieved April 04, 2018, from https://wiki.unrealengine.com/World_Machine_to_UE4_using_World_Composition

Dorschel, B. (n.d.). Forschungsschiff MARIA S. MERIAN - Reisen Nr. MSM65 – MSM68. Institut für Geologie Universität Hamburg - Leitstelle Deutsche Forschungsschiffe. Retrieved January 19, 2018, from http://epic.awi.de/40384/26/msm65-msm68-expeditionsheft.pdf

Dufek, T. (2012). Backscatter Analysis of Multibeam Sonar Data in the Area of the Valdivia Fracture Zone using Geocoder in CARIS HIPS&SIPS and IVS3D Fledermaus. (Master's thesis). Retrieved from 10013/epic.40792

Dunlop, R. (2003). *FPS Versus Frame Time*. Retrieved April 28, 2018, from https://www.mvps.org/directx/articles/fps_versus_frame_time.htm

Egbert, G., & Erofeeva, L. (2010). *OSU Tidal Prediction Software (OTPS).* Retrieved January 29, 2018, from http://volkov.oce.orst.edu/tides/otps.html

Epic Games, Inc. (2018a). *World Composition User Guide*. Retrieved April 03, 2018, from https://docs.unrealengine.com/en-us/Engine/LevelStreaming/WorldBrowser

Epic Games, Inc. (2018b). *Material Parameter Collections*. Retrieved April 05, 2018, from https://docs.unrealengine.com/en-us/Engine/Rendering/Materials/ParameterCollections

Epic Games, Inc. (2018c). *Volumes Reference*. Retrieved April 06, 2018, from https://docs.unrealengine.com/en-us/Engine/Actors/Volumes

Epic Games, Inc. (2018d). *Atmospheric Fog User Guide*. Retrieved April 06, 2018, from https://docs.unrealengine.com/en-us/Engine/Actors/FogEffects/AtmosphericFog

Epic Games, Inc. (2018e). *Lighting the Environment*. Retrieved April 06, 2018, from https://docs.unrealengine.com/en-us/Engine/Rendering/LightingAndShadows

Epic Games, Inc. (2018f). *Post Process Effects*. Retrieved April 11, 2018, from https://docs.unrealengine.com/en-us/Engine/Rendering/PostProcessEffects

Epic Games, Inc. (2018g). *Scene Capture 2D*. Retrieved April 17, 2018, from https://docs.unrealengine.com/en-us/Resources/ContentExamples/Reflections/1_7

Epic Games, Inc. (2018h). *Landscape Technical Guide*. Retrieved April 29, 2018, from https://docs.unrealengine.com/en-us/Engine/Landscape/TechnicalGuide

Epic Games, Inc. (2018i). *What is Unreal Engine 4*. Retrieved May 16, 2018, from https://www.unrealengine.com/en-US/what-is-unreal-engine-4

Epic Games, Inc. (2018j). *Creating and Using Custom Heightmaps and Layers*. Retrieved May 17, 2018, from https://docs.unrealengine.com/en-us/Engine/Landscape/Custom

ESRI. (2016). *How Focal Statistics works*. Retrieved February 18, 2018, from http://desktop.arcgis.com/en/arcmap/10.3/tools/spatial-analyst-toolbox/how-focal-statistics-works.htm

ESRI. (2018). *Project Raster*. Retrieved February 18, 2018, from https://pro.arcgis.com/en/pro-app/tool-reference/data-management/project-raster.htm

Fuchs, P. (2017). *Virtual Reality Headsets - A Theoretical and Pragmatic Approach.* London: CRC Press.

Gepp, M. (2017). *Here Are The VR System Requirements Needed To Run The HTC VIVE*. Retrieved April 24, 2018, from https://blog.vive.com/us/2017/10/16/8697/

Government of Canada. (2016). *Clyde River*. Retrieved January 20, 2018, from http://www4.nrcan.gc.ca/search-place-names/unique/OADOM

Government of Canada, Natural Resources Canada. (2016). *Canadian Digital Elevation Model Product Specifications.* Retrieved February 19, 2018, from http://ftp.geogratis.gc.ca/pub/nrcan_rncan/elevation/cdem_mnec/doc/CDEM_product_specs.pdf

Gutiérrez, M. A., Vexo, F., & Thalmann, D. (2008). *Stepping into Virtual Reality.* London: Springer-Verlag.

Hofmann-Wellenhof, B., Lichtenegger, H., & Collins, J. (2001). *Global Positioning System* (5 ed.). New York: Springer-Verlag Wien GmbH.

HTC Corporation. (2018). *SPECS & DETAILS.* Retrieved May 15, 2018, from https://www.vive.com/uk/product/#vive-spec

iNaturalist Canada. (2016a). *Silene suecica.* (Canadian Wildlife Federation (CWF) and the Royal Ontario Museum (ROM)) Retrieved April 06, 2018, from http://inaturalist.ca/taxa/485190-Silene-suecica

iNaturalist Canada. (2016b). *Common Heather.* (Canadian Wildlife Federation (CWF) and the Royal Ontario Museum (ROM)) Retrieved April 06, 2018, from http://inaturalist.ca/taxa/119450-Calluna-vulgaris

iNaturalist Canada. (2016c). *Androsace septentrionalis.* (Canadian Wildlife Federation (CWF) and the Royal Ontario Museum (ROM)) Retrieved April 06, 2018, from http://inaturalist.ca/taxa/75436-Androsace-septentrionalis

Ingham, A. (1992). *Hydrography for the Surveyor and Engineer* (Vol. 3). Oxford: Blackwell Scientific Publications.

International Hydrographic Organization. (2005). *Manual on Hydrography.* Monaco: International Hydrographic Bureau.

Jakobsson, M., Mayer, L., Coakley, B., Dowdeswell, J. A., Forbes, S., Fridman, B., . . . Weatherall, P. (2012). *International Bathymetric Chart of the Arctic Ocean (IBCAO) Version 3.0.* Retrieved February 19, 2018, from https://www.ngdc.noaa.gov/mgg/bathymetry/arctic/2012GL052219.pdf

Jerald, J. (2016). *The VR Book: Human-Centered Design for Virtual Reality* (Vol. 1). New York: Association for Computing Machinery and Morgan & Claypool. doi:10.1145/2792790

Jong, C. d., Lachapelle, G., Skone, S., & Elema, I. (2010). *Hydrography.* Delft: DUP Blue Print.

Kersten, T., Tschirschwitz, F., & Deggim, S. (2017). Development of a Virtual Museum including a 4D Presentation of Building History in Virtual Reality. In D. Aguilera, A. Georgopoulos, T. Kersten, F. Remondino, & E. Stathopoulou, *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XLII-2/W3, 3D Virtual Reconstruction and Visualization of Complex Architectures* (pp. 361-367). Nafplio, Greece.

Kongsberg Maritime AS. (2011). *Product description EM 122 Multibeam Echo Sounder.* Retrieved January 30, 2018, from http://epic.awi.de/45364/1/Kongsberg_302440ae_em122_product_description.pdf

Kongsberg Maritime AS. (2013). *SIS Seafloor Information System.* Retrieved January 30, 2018, from https://www.km.kongsberg.com/ks/web/nokbg0397.nsf/AllWeb/A269870356C3A572C1256E19004ECFA9/$file/164878ac_SIS_Product_specification.pdf

Lurton, X. (2010). *An Introduction to Underwater Acoustics - Principles and Applications* (Vol. 2). Berlin: Springer.

Lurton, X., Lamarche, G., Brown, C., Lucieer, V., Rice, G., Schimel, A., & Weber, T. (2015). *Backscatter measurements by seafloor-mapping sonars. Guidelines and Recommendations.* Retrieved from http://geohab.org/wp-content/uploads/2013/02/BWSG-REPORT-MAY2015.pdf

McCaffrey, M. (2017). *Unreal Engine VR Cookbook.* Boston: Addison-Wesley.

Milgram, P., Takemura, H., Utsumi, A., & Kishino, F. (1994). Augmented reality: a class of displays on the reality-virtuality continuum. *SPIE - Telemanipulator and Telepresence Technologies, 2351*, pp. 282-292.

Misra, R., & Ramesh, A. (1989). *Fundamentals of Cartography* (Vol. 2). New Delhi: Ashok Kumar Mittal.

Municipality of Clyde River. (n.d.). *Welcome to Clyde River Kanngiqtugaapik ("Nice Little Inlet").* Retrieved January 17, 2018, from http://clyderiver.ca/

Nielsen, J. (2012). *Usability 101: Introduction to Usability.* Retrieved May 06, 2018, from https://www.nngroup.com/articles/usability-101-introduction-to-usability/

Oleson, J. (2008). Testing the Waters: The Role of Sounding-Weights in Ancient Mediterranean Navigation. In R. Hohlfelder, *The Maritime World of Ancient Rome* (pp. 119-176). University of Michigan Press.

Polar Geospatial Center & Regents of the University of Minnesota. (2017). *Introduction to ArcticDEM.* Retrieved February 19, 2018, from https://www.pgc.umn.edu/guides/arcticdem/introduction-to-arcticdem/

QPS. (2018). *FMGT.* Retrieved March 10, 2018, from https://confluence.qps.nl/display/FM780/FMGT

RGI Consortium. (2017). *Randolph Glacier Inventory – A Dataset of Global Glacier Outlines: Version 6.0: Technical Report, Global Land Ice Measurements from Space.* Colorado, USA: Digital Media. doi:https://doi.org/10.7265/N5-RGI-60

Schwartz, M. L. (2005). *Encyclopedia of Coastal Science.* Dordrecht: Springer.

Sea-Bird Scientific. (2017). *SBE 911plus CTD.* Retrieved January 31, 2018, from http://www.seabird.com/sbe911plus-ctd

Steinicke, F. (2016). *Being Really Virtual: Immersive Natives and the Future of Virtual Reality.* Cham, Switzerland: Springer International Publishing.

Tavakkoli, A. (2016). *Game Development and Simulation with Unreal Technology.* Boca Raton: Taylor & Francis Group.

Valve Corporation. (2016). *HTC Vive.* Retrieved May 15, 2018, from https://store.steampowered.com/app/358040/HTC_Vive/

Wykes, T., & Inuit Heritage Trust Inc. (2007). *Kangiqtugaapik.* Retrieved January 18, 2018, from http://ihti.ca/eng/place-names/NU27F_Kangiqtugaapik.pdf

Zhou, Q., Lees, B., & Tang, G.-a. (2008). *Advances in Digital Terrain Analysis.* Berlin Heidelberg: Springer-Verlag.

9 Appendix

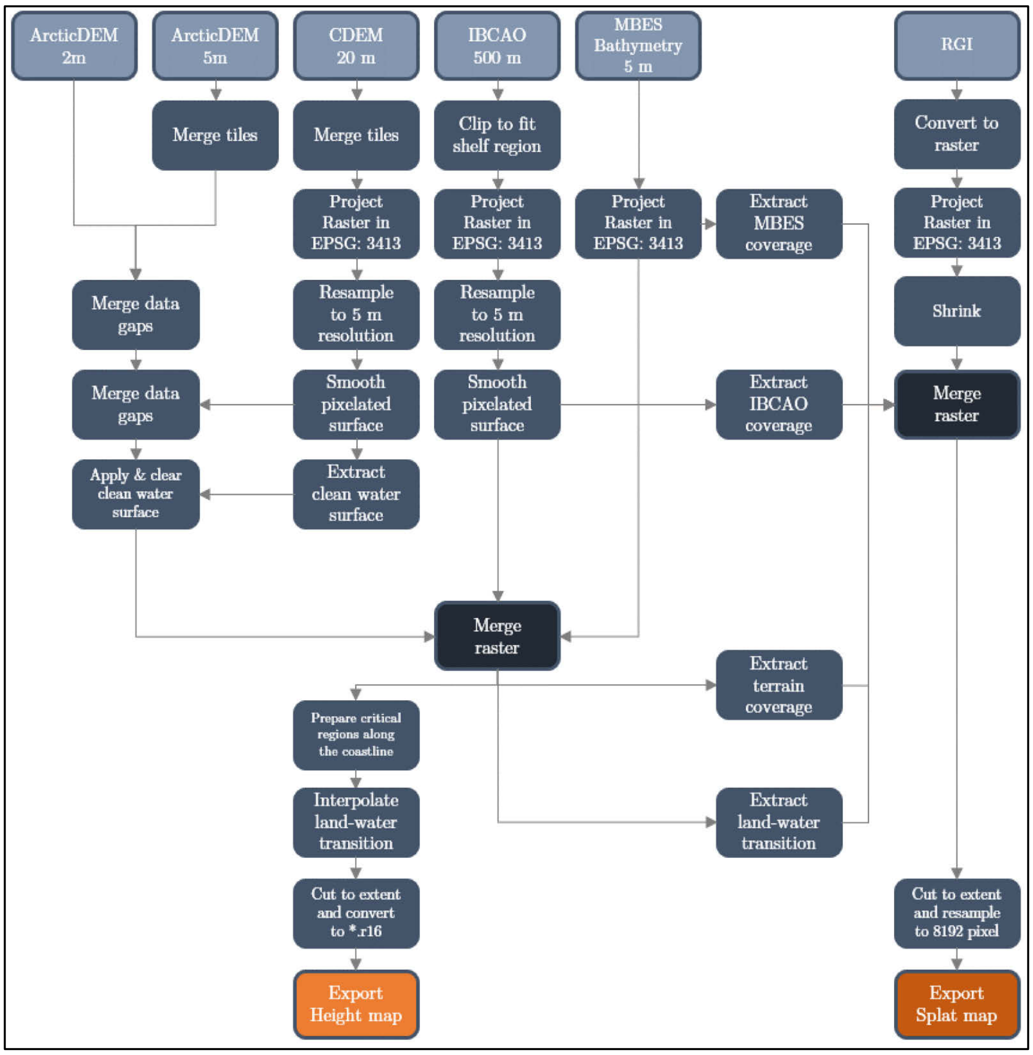
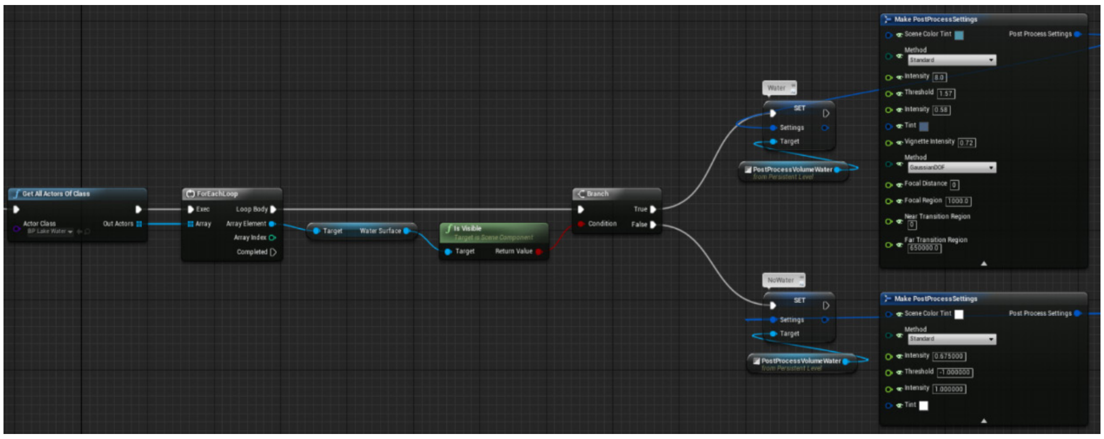

ArcticDEM 2m
ArcticDEM 5m
CDEM 20 m
IBCAO 500 m
MBES Bathymetry 5 m
RGI
Merge tiles
Merge tiles
Clip to fit shelf region
Convert to raster
Project Raster in EPSG: 3413
Project Raster in EPSG: 3413
Project Raster in EPSG: 3413
Extract MBES coverage
Project Raster in EPSG: 3413
Merge data gaps
Resample to 5 m resolution
Resample to 5 m resolution
Shrink
Merge data gaps
Smooth pixelated surface
Smooth pixelated surface
Extract IBCAO coverage
Merge raster
Apply & clear clean water surface
Extract clean water surface
Merge raster
Extract terrain coverage
Prepare critical regions along the coastline
Interpolate land-water transition
Extract land-water transition
Cut to extent and convert to *.r16
Cut to extent and resample to 8192 pixel
Export Height map
Export Splat map

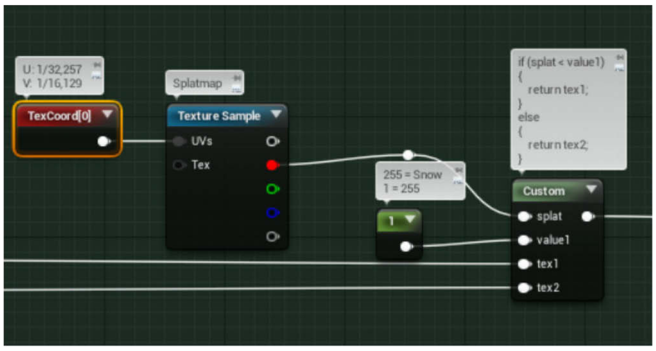
Figure 9-1: Processing of bathymetric and terrestrial terrain models to obtain height and splat map



Make PostProcessSettings
Scene Color Tint
Method Standard
Intensity 8.0
Threshold 1.57
Intensity 0.58
Tint
Vignette Intensity 0.72
Method GaussianDOF
Focal Distance 0
Focal Region 1000.0
Near Transition Region 0
Far Transition Region 650000.0

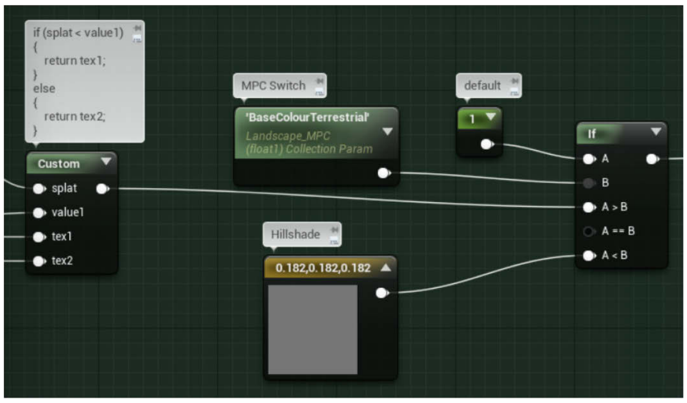
Make PostProcessSettings
Scene Color Tint
Method Standard
Intensity 0.675000
Threshold -1.000000
Intensity 1.000000
Tint

Figure 9-2: Level Blueprint: Activate Post Process Volume with water effects when the water surface is on and disable all under water effects when the water surface is off



```
if (splat < value1)
{
  return tex1;
}
else
{
  return tex2;
}
```

Figure 9-3: Landscape Material: Colour snow according to splat map. Every location that has a grey value smaller than 255 will be coloured as grass, ground or rock. Every location that has a grey value of 255 will be coloured as snow. Texture 1 is the output from the "Natural Terrain" BaseColor. Texture 2 is the snow texture



```
if (splat < value1)
{
  return tex1;
}
else
{
  return tex2;
}
```

Figure 9-4: Landscape Material: MPC switches between the default "Natural" view and "Hillshade" view. Hillshade will be activated when the MPC receives a value higher than 1

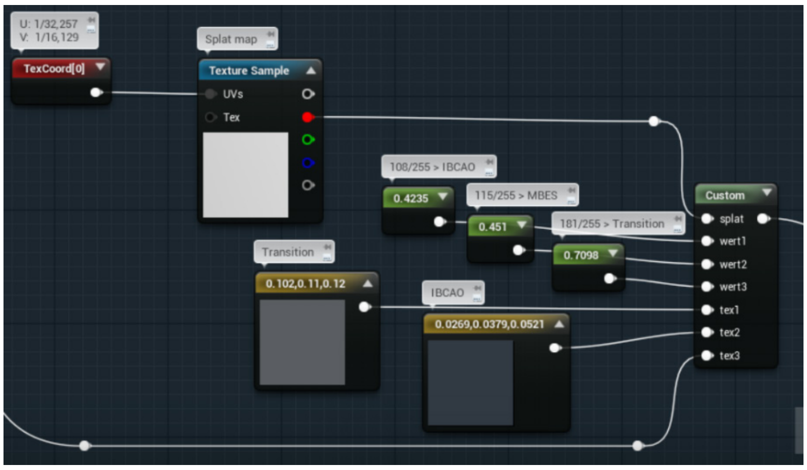


Figure 9-5: Landscape Material: Colour bathymetry according to splat map. Every location that has a grey value smaller than 108 receives IBCAO texture, every location that has a grey value between 109 and 115 receives MBES colouration and every location that has a grey value higher than 116 receives a texture for the transition

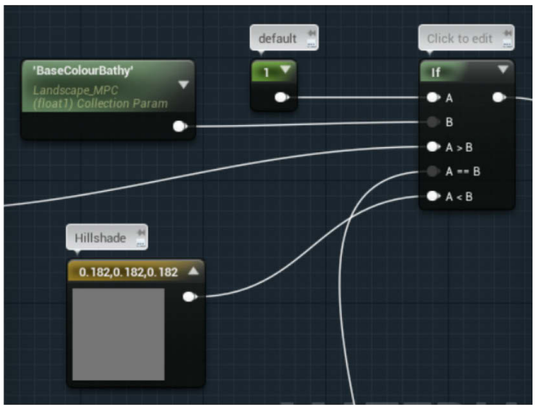


Figure 9-6: Landscape Material: MPC switched between backscatter (A>B), rainbow colouration (A==B) and Hillshade (A<B)

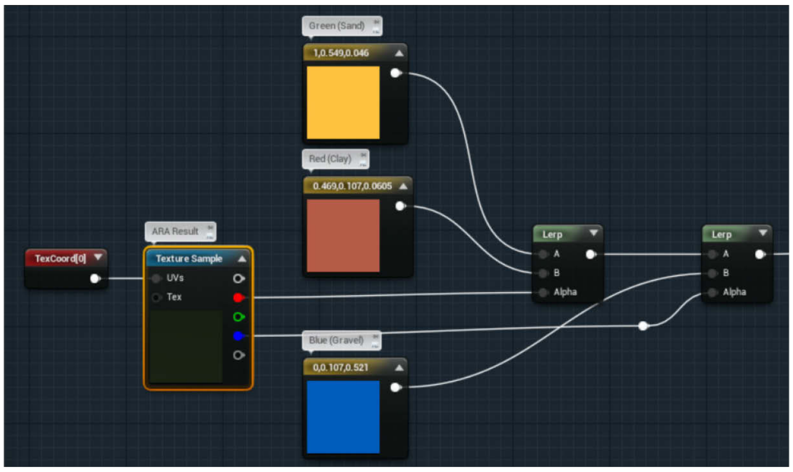


Figure 9-7: Landscape Material: Backscatter colouration using the RGB splat map from section 3.2.4. Red receives ochre colour, Green receives yellow colour and Blue receives another shade of blue

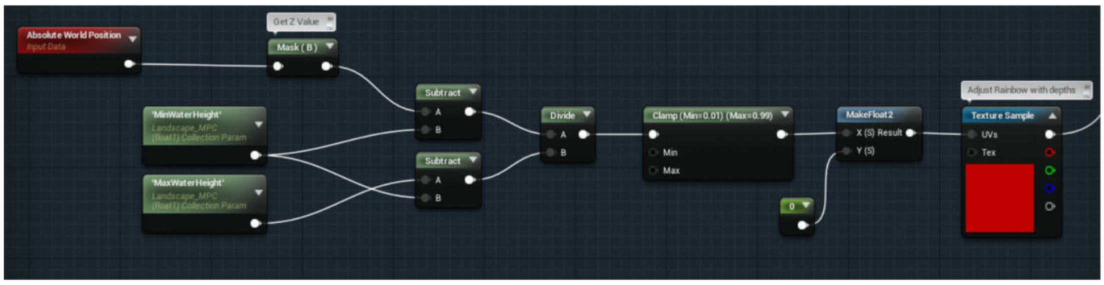


Figure 9-8: Landscape Material: Rainbow colouration according to the terrain height. The height range can be set during running application by choosing values for the MPC parameters “MinWaterHeight” and “MaxWaterHeight”

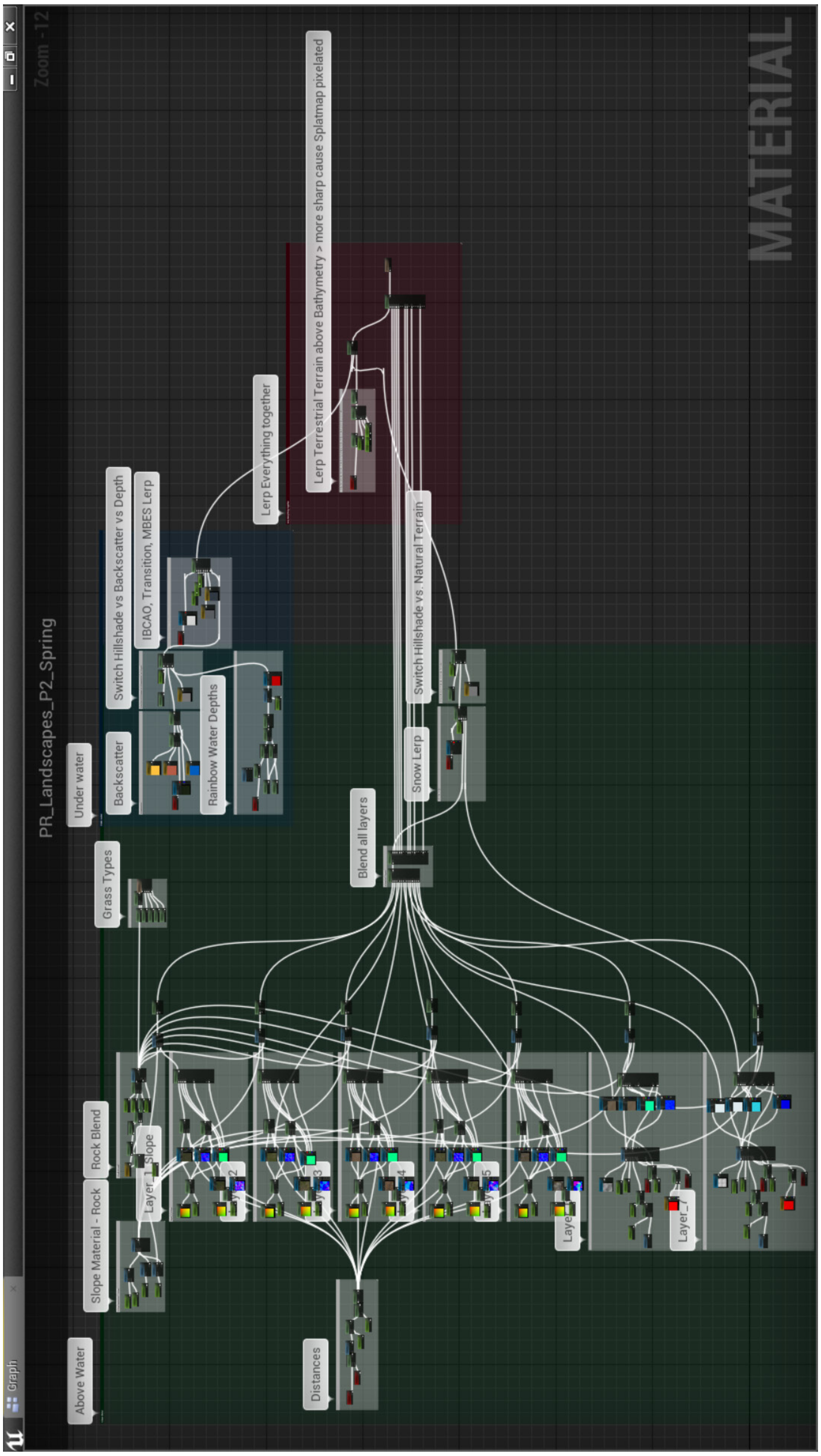


Figure 9-9: The entire Landscape Material

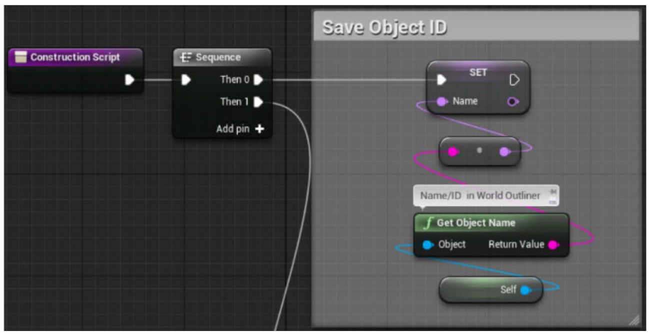


Figure 9-10: Blueprint “BP_Birds”: get object name of active birds and save it to variable “Name”

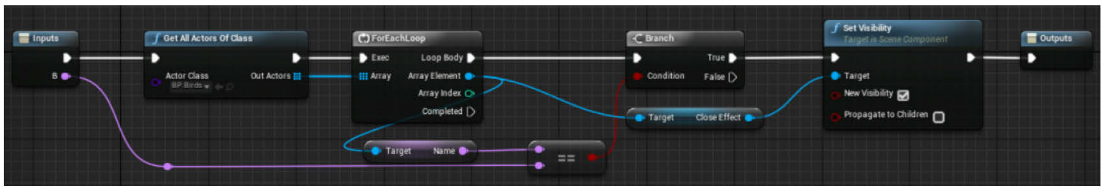


Figure 9-11: Main Game Instance – Macro “Show Birds”: Get name of active birds and make them visible

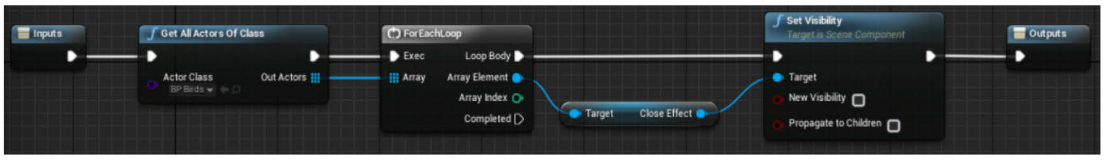


Figure 9-12: Main Game Instance – Macro “Hide Birds”: Get all birds and make them invisible

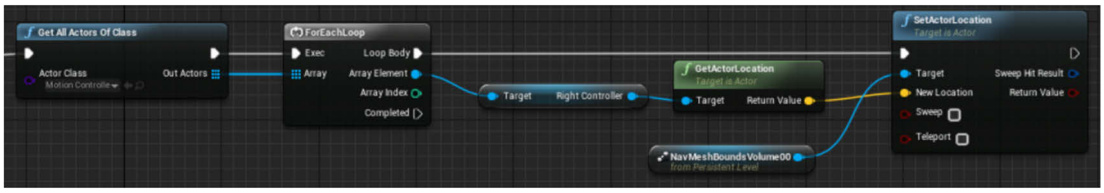


Figure 9-13: Persistent Level Blueprint: Navigation Mesh is being moved constantly to the user

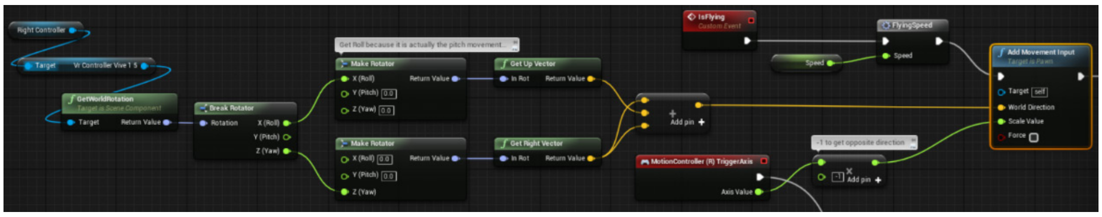


Figure 9-14: Blueprint “MotionControllerPawn”: Flying locomotion logic. Movement is received from right controller pitch and yaw movement. The speed can be changed during running application

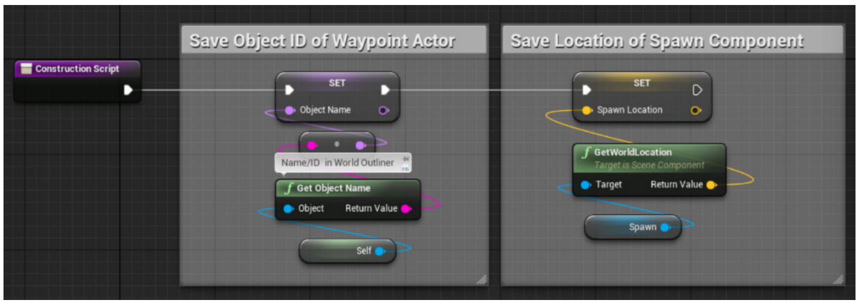


Figure 9-15: Blueprint "Waypoint": get name and location of each waypoint and save it to variable

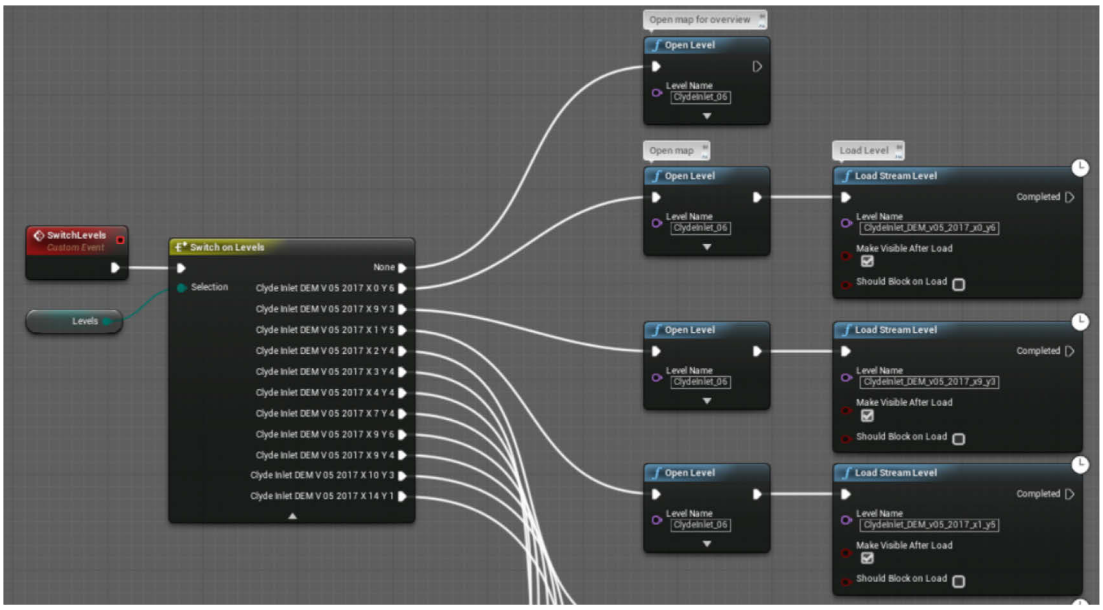

SwitchLevels
Custom Event
Levels
Switch on Levels
Selection
None
Clyde Inlet DEM V 05 2017 X 0 Y 6
Clyde Inlet DEM V 05 2017 X 9 Y 3
Clyde Inlet DEM V 05 2017 X 1 Y 5
Clyde Inlet DEM V 05 2017 X 2 Y 4
Clyde Inlet DEM V 05 2017 X 3 Y 4
Clyde Inlet DEM V 05 2017 X 4 Y 4
Clyde Inlet DEM V 05 2017 X 7 Y 4
Clyde Inlet DEM V 05 2017 X 9 Y 6
Clyde Inlet DEM V 05 2017 X 9 Y 4
Clyde Inlet DEM V 05 2017 X 10 Y 3
Clyde Inlet DEM V 05 2017 X 14 Y 1
Open map for overview
Open Level
Level Name
ClydeInlet_06
Open map
Load Level
Load Stream Level
Completed
ClydeInlet_DEM_v05_2017_x0_y6
Make Visible After Load
Should Block on Load
ClydeInlet_DEM_v05_2017_x9_y3
ClydeInlet_DEM_v05_2017_x1_y5

Figure 9-16: Main Game Instance: Event "SwitchLevels" receives input from "Levels" variable declared within the menu Widget and chooses the corresponding output. The output will load the map and Level

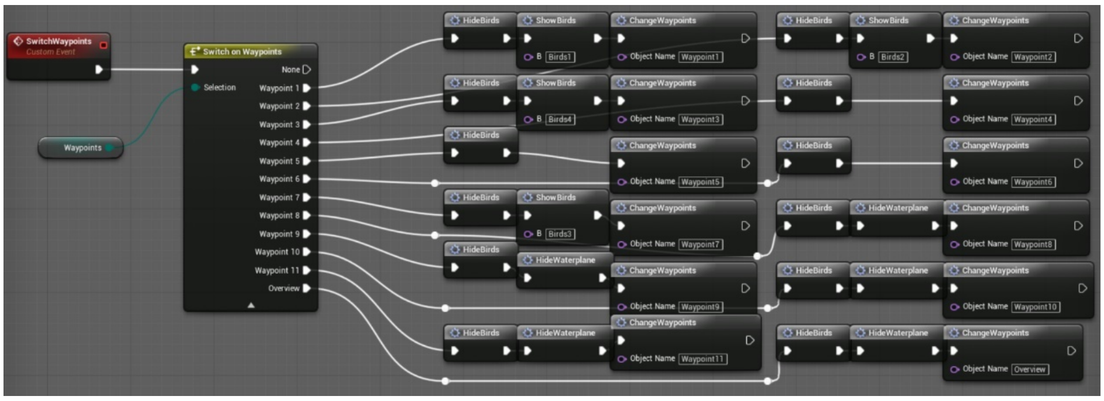


Figure 9-17: Main Game Instance: Event "SwitchWaypoints" receives input from "Waypoints" variable declared within the menu Widget and chooses the corresponding output. The output will hide/show birds or water surface and transports the user to the selected location

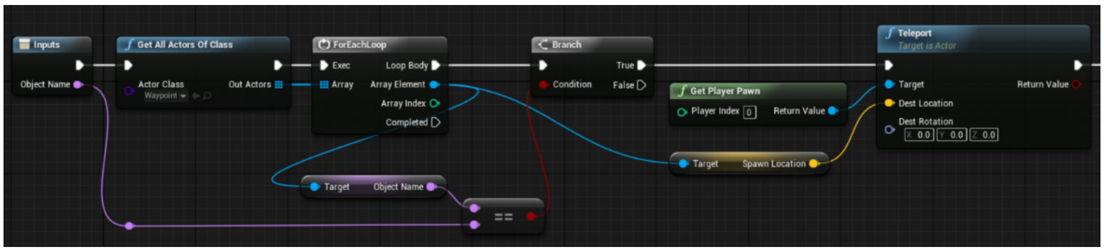


Figure 9-18: Main Game Instance – Macro “ChangeWaypoints”: This code receives the object name and location of the waypoint and transports the user to that waypoint

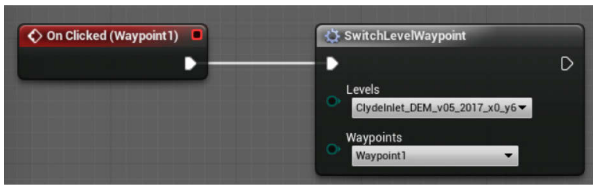


Figure 9-19: Widget "IngameMenuWidget2": On button clicked, the variables “Levels” and “Waypoints” will be initialised and passed to the Macro “SwitchLevelWaypoint”

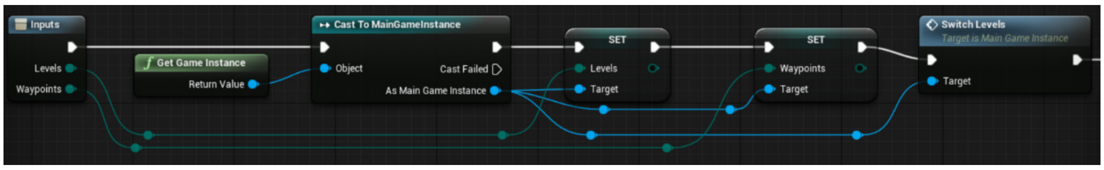


Figure 9-20: Widget "IngameMenuWidget2": Macro casts to Game Instance Class to fire the event “SwitchLevels” which will load the corresponding Level

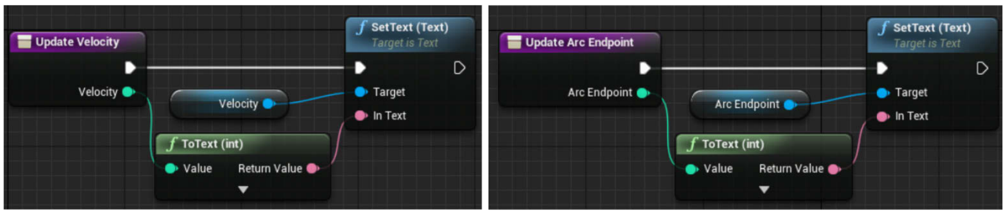


Figure 9-21: Widget "VelocityWidget": Function to update the value for the speed (left) and distance (right)

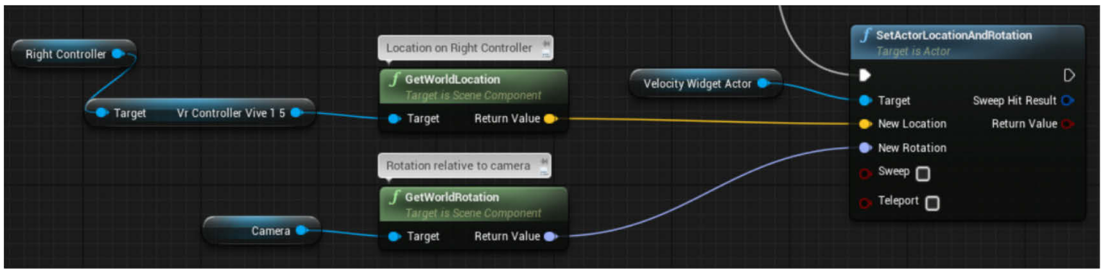


Figure 9-22: Blueprint “MotionControllerPawn”: Set the location of the “VelocityWidget” relative to the right controller and the rotation relative to the headset

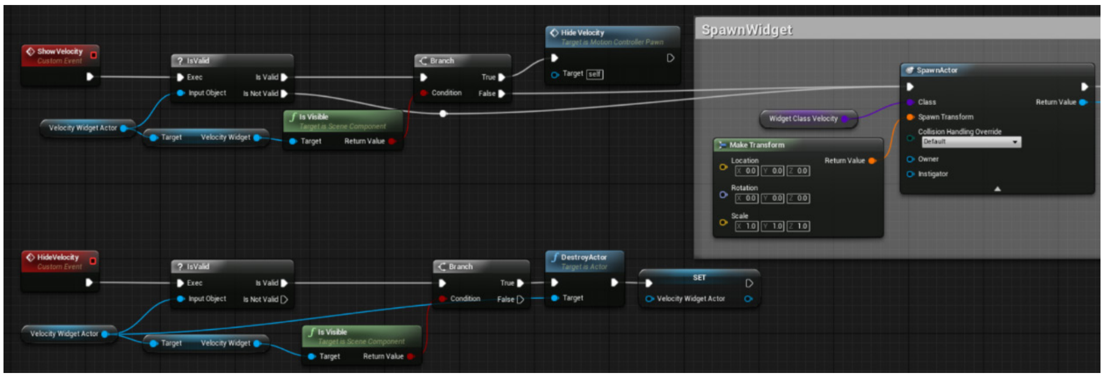


Figure 9-23: Blueprint "MotionControllerPawn": Create the function "ShowVelocity" to spawn the widget and "HideVelocity" to destroy the actor

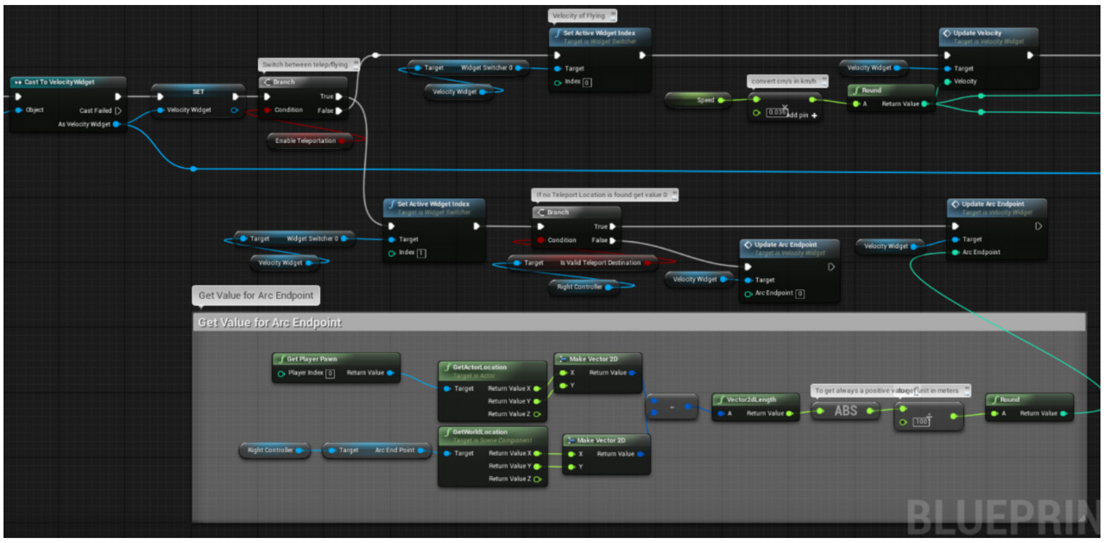


Figure 9-24: Blueprint "MotionControllerPawn": Toggle between flying and teleportation display using the Boolean "EnableTeleportation". Access values for the speed and distance of the jump and pass them to the function "UpdateVelocity" and "UpdateArcEndpoint"

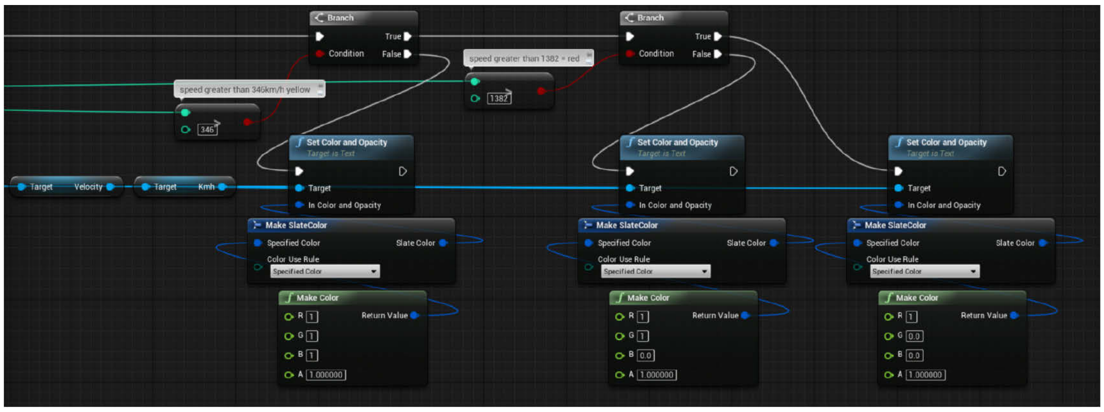


Figure 9-25: Blueprint "MotionControllerPawn": The value for speed is being coloured yellow when its above 346 km/h and coloured red when its above 1382 km/h

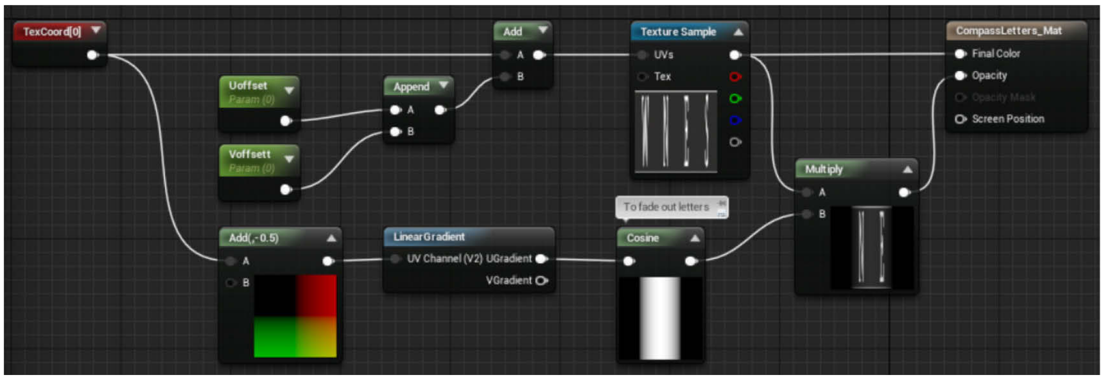


Figure 9-26: Material “CompassLetters_M”: The UV coordinates are accessed to be able to pan the texture later on. Also, the left and right edges fade out using the Cosine function

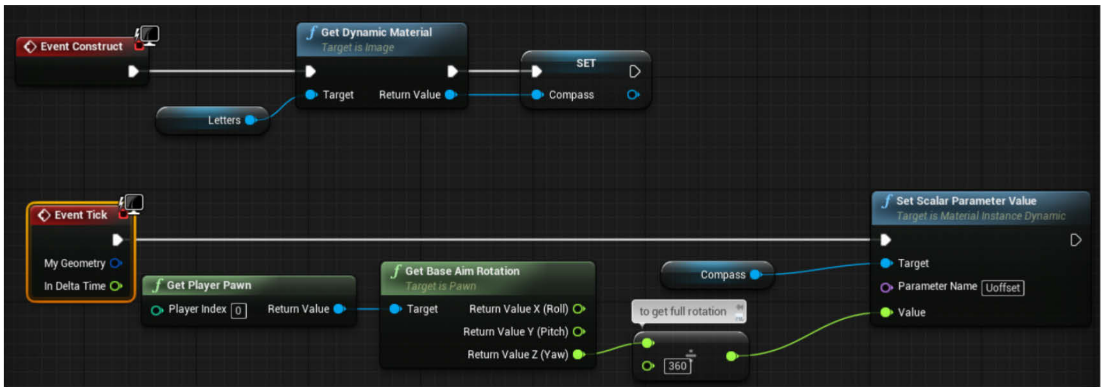


Figure 9-27: Widget "ElevationWidget": The widget image which holds the compass texture is being set as a dynamic material and panned in U direction according to the rotation of the user

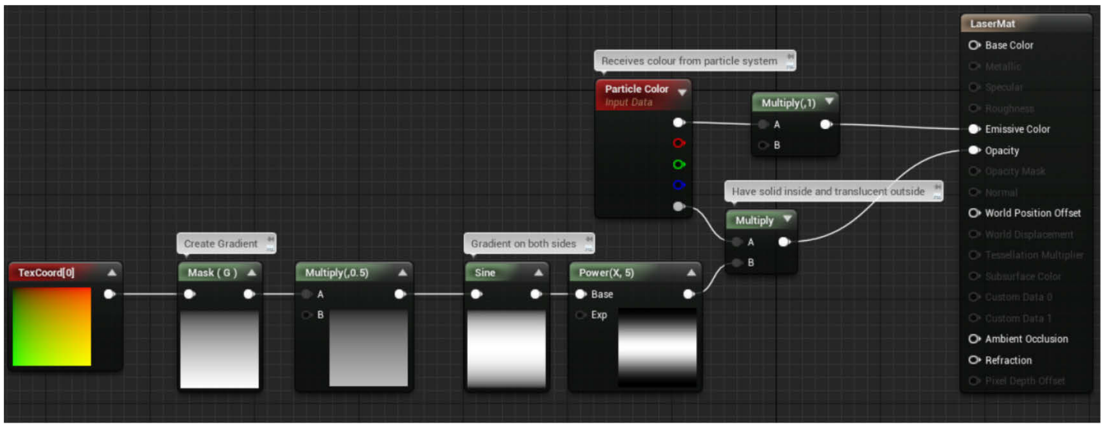


Figure 9-28: Material “LaserMat”: creates the texture for the laser beam

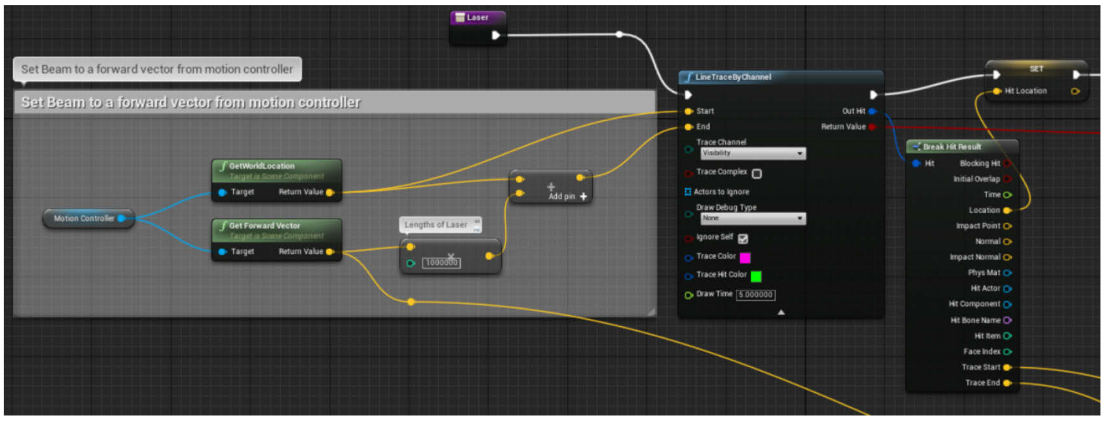


Figure 9-29: Blueprint "BP_MotionController": Line trace from the motion controller with a length of 10 km. The intersection with geometry is saved in the variable “Hit Location”

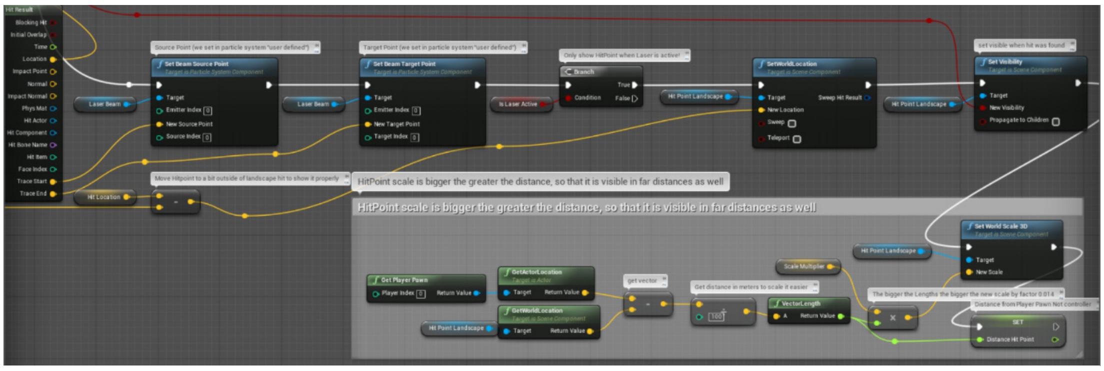


Figure 9-30: Blueprint "BP_MotionController": The source and target point of the laser Particle System are set to the start and end point of the line trace. A sphere indicates the hit point and gets visible only when a hit occurs. The greater the distance to the sphere, the bigger is its size

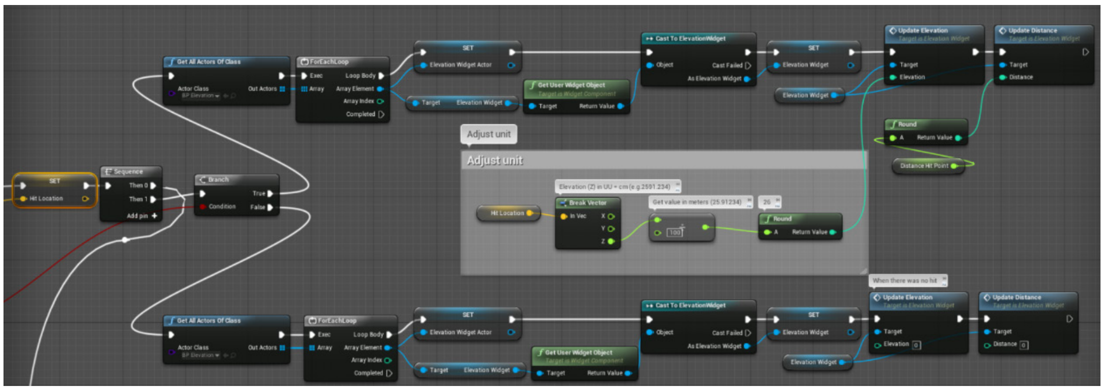


Figure 9-31: Blueprint "BP_MotionController": Receive values for the elevation and distance of the hit point and pass it to the widget. When no hint occurs, the values will be 0

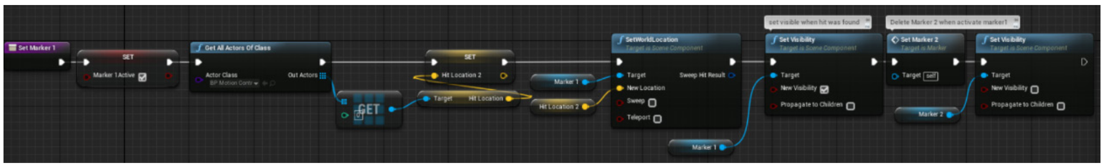


Figure 9-32: Blueprint "Marker": Function “Set Marker1” places the mesh “Marker1” at the location of the hit point, turns the mesh visible and disables the mesh that represents the second hit point

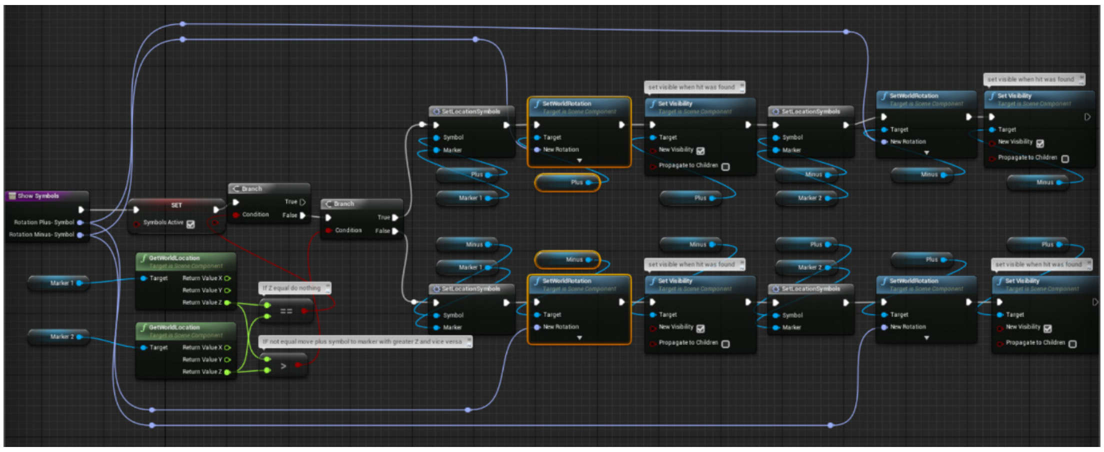


Figure 9-33: Blueprint "Marker": Function “Show Symbols” locates the plus and minus symbols to the corresponding marker (indicator). If “Marker 1” has a higher elevation than “Marker 2”, the plus symbol is placed on top of “Marker1” and vice versa. If they are both equal no symbols will be placed

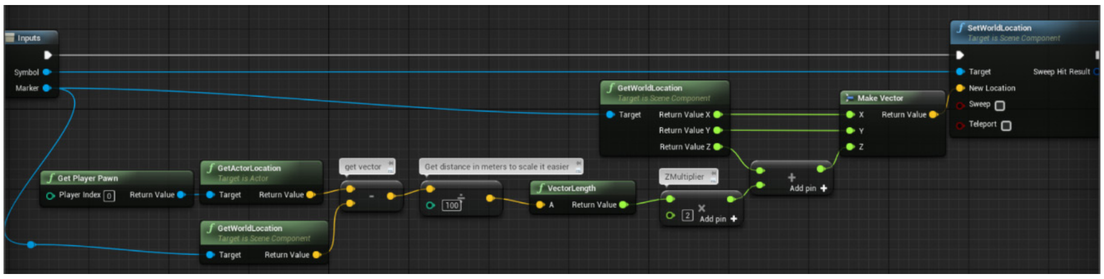


Figure 9-34: Blueprint "Marker": Function “Set Location Symbols” places the symbols on top of the indicators. The distance from the indicators changes with the distance from the user

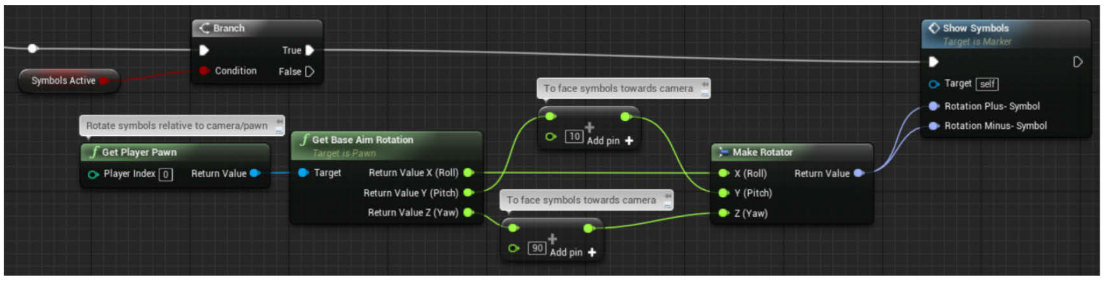


Figure 9-35: Blueprint "Marker": The rotation of the symbols is constantly adjusted so that the symbols face the user constantly

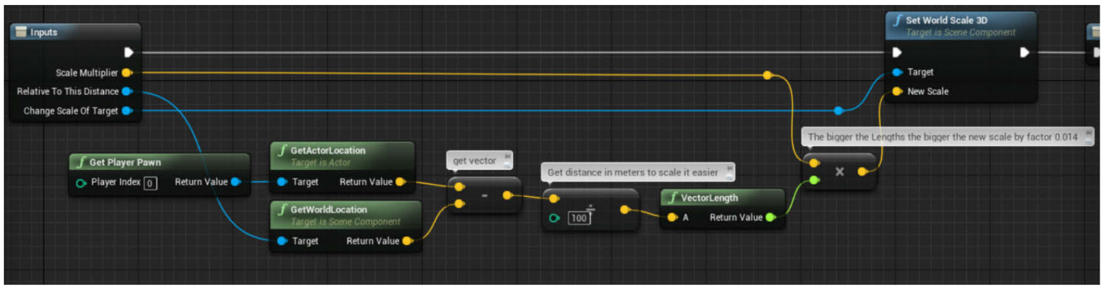


Figure 9-36: Blueprint "Marker": Macro “ChangeScale” changes the scale with the distance from the user

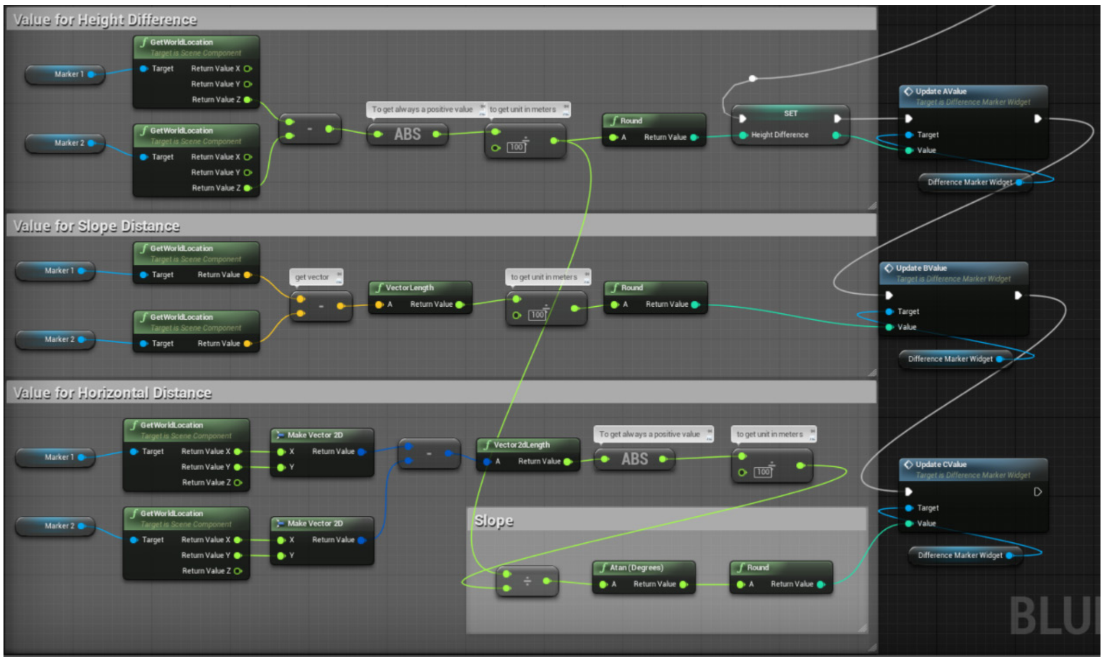


Figure 9-37: Blueprint "Marker": Determine the height difference, slope distance and slope between indicators and pass it to the Widget

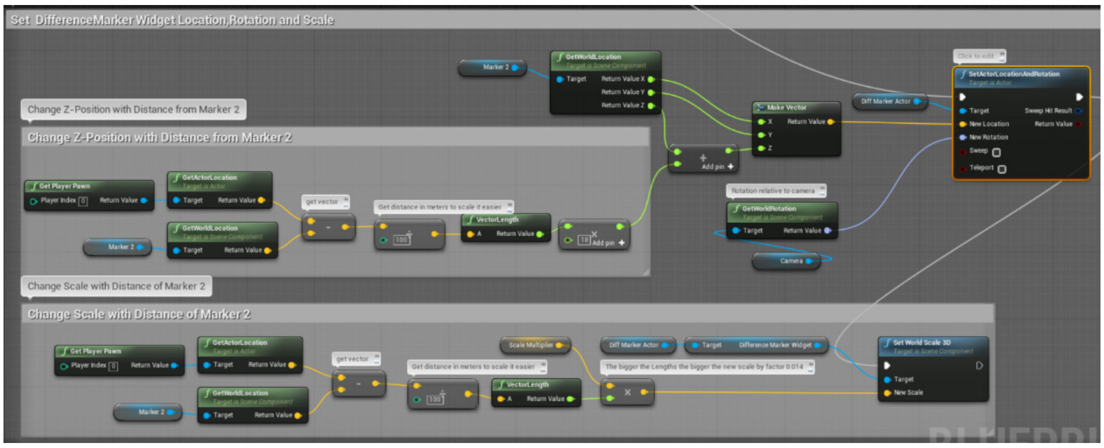

Set DifferenceMarker Widget Location,Rotation and Scale
Change Z-Position with Distance from Marker 2
Change Scale with Distance of Marker 2
Marker 2
GetWorldLocation
Make Vector
Add pin
Diff Marker Actor
SetActorLocationAndRotation
Target
New Location
New Rotation
Sweep
Teleport
Sweep Hit Result
Return Value
Rotation relative to camera
GetWorldRotation
Camera
Get Player Pawn
GetActorLocation
get vector
Get distance in meters to scale it easier
VectorLength
Scale Multiplier
Difference Marker Widget
The bigger the Lengths the bigger the new scale by factor 0.014
Set World Scale 3D
New Scale

Figure 9-38: Blueprint “MotionControllerPawn”: Attach the Widget on top of the second indicator and change its height according to the distance from the user. Rotate the widget according to the user

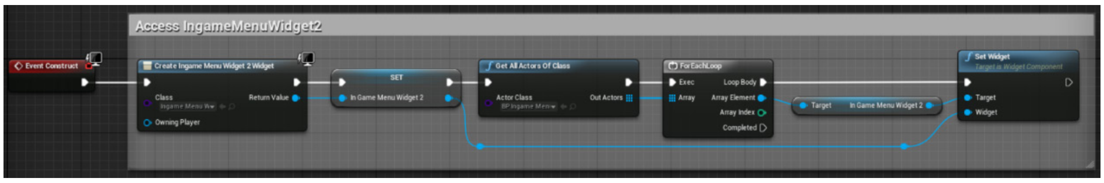


Figure 9-39: Widget “IngameMenuWidget”: Access the second menu Widget within the first Widget

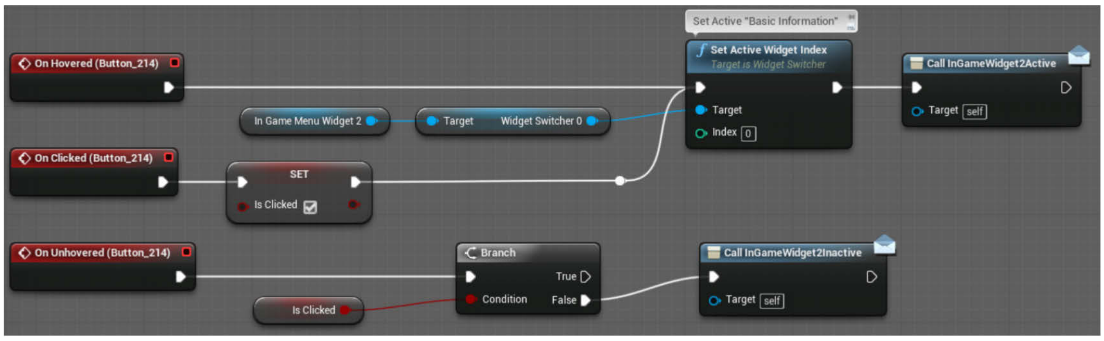


Figure 9-40: Widget “IngameMenuWidget”: on button hovered and clicked the corresponding Canvas Panel of the second menu is made active and an Event Dispatcher is called to make the second menu visible. On unhovered button, another Event Dispatcher is called to hide the second menu Widget

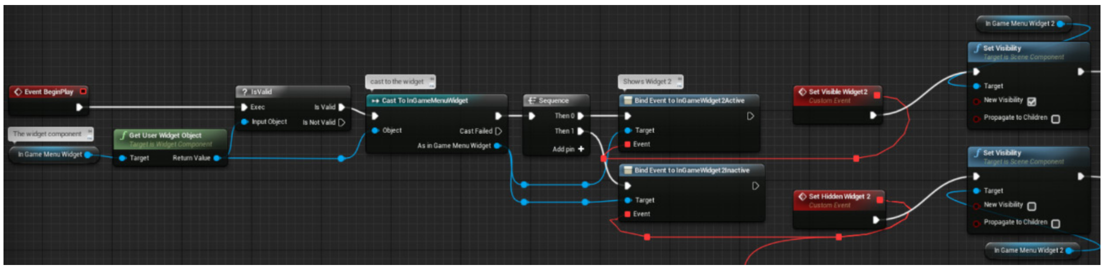


Figure 9-41: Blueprint “BP_IngameMenu”: Cast to the first menu widget and bind events to make the second menu visible/invisible

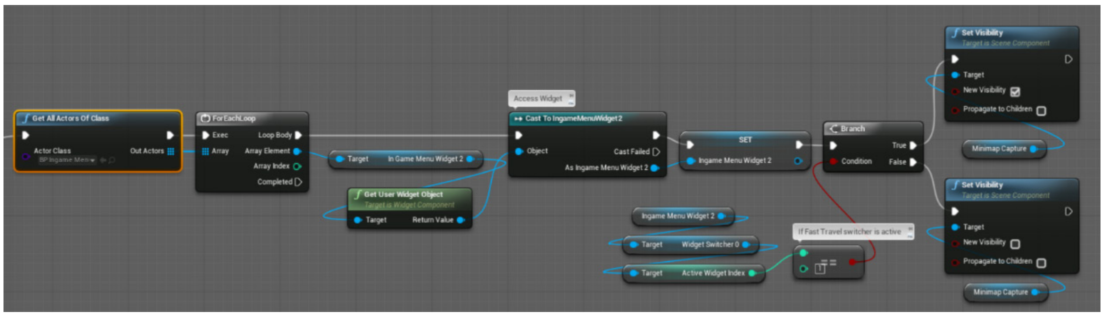


Figure 9-42: Blueprint “MotionControllerPawn”: Activate Scene Capture Component when the Positioning menu page is switched on

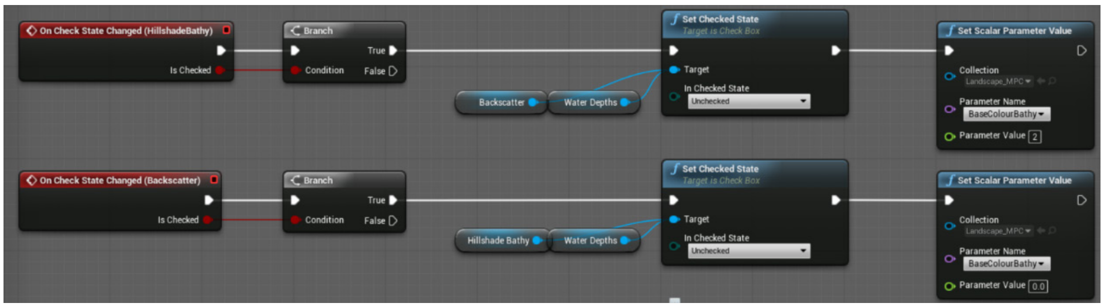


Figure 9-43: Widget “IngameMenuWidget2”: Change the colouration of the terrain by inserting a parameter value inside the MPC when the check box is activated. At the same time uncheck other checkboxes

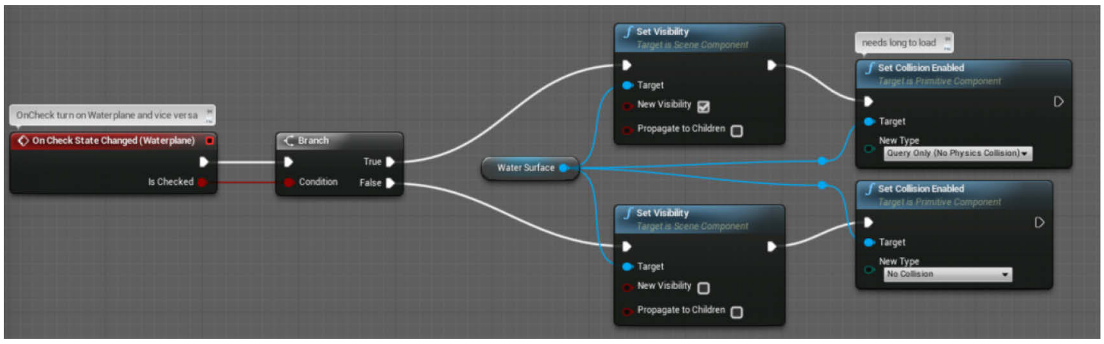


Figure 9-44: Widget “IngameMenuWidget2”: Change the visibility of the water surface and the collision on check state change

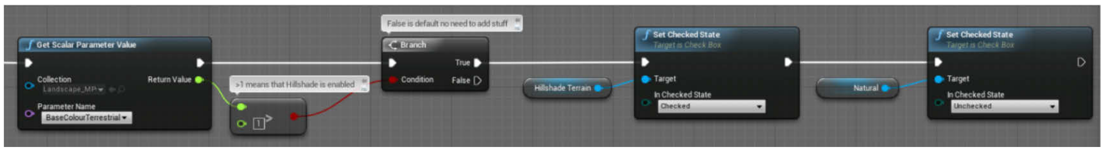


Figure 9-45: Widget “IngameMenuWidget2”: Check which colouration is active in the scene and tick the corresponding check box once the menu is opened. This Figure is an example for the terrestrial colouration

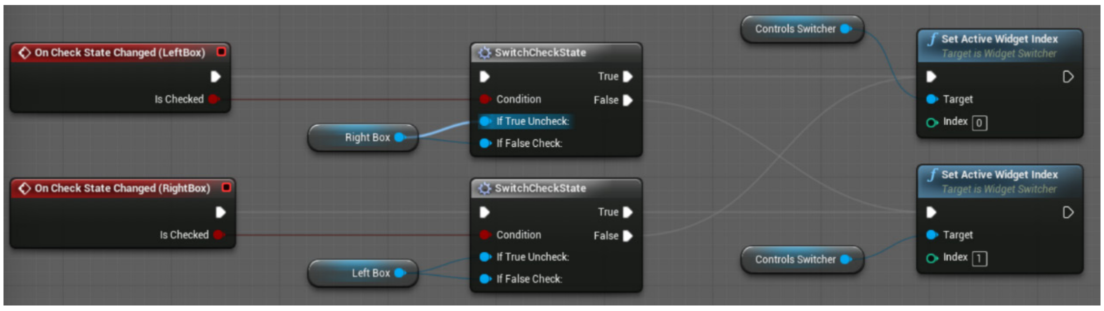


Figure 9-46: Widget “IngameMenuWidget2”: When the check box for the left controller is checked, the right check box is unchecked and the left CanvasPanel appears and vice versa

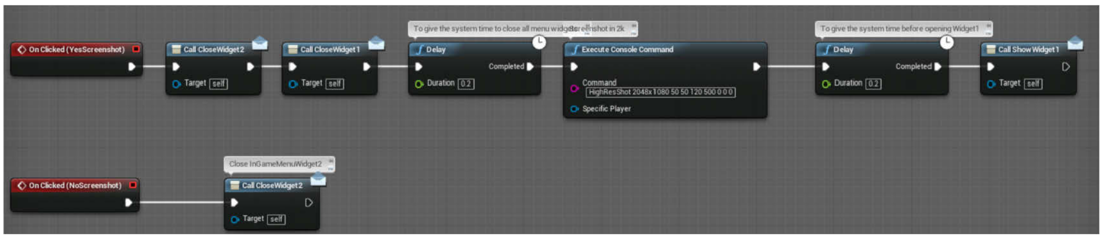


Figure 9-47: Widget “IngameMenuWidget2”: Once the screenshot has been initialised on button click, both menu widgets are closed, the screenshot command fires before the first menu widget is opened again

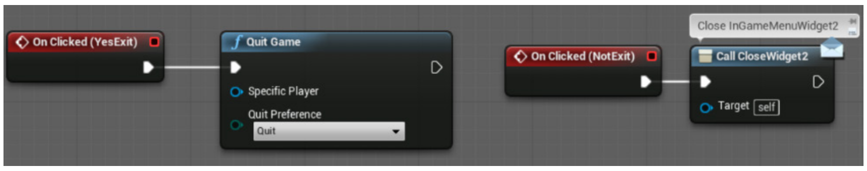


Figure 9-48: Widget “IngameMenuWidget2”: Once the button to quit the game has been pressed, the function “Quit Game” is called to end the application

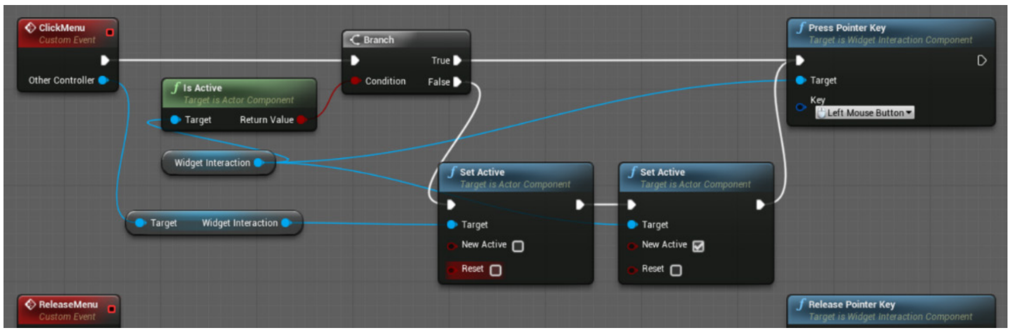


Figure 9-49: Blueprint "BP_MotionController": Create two functions that translate motion controller press and release input to left mouse button clicks

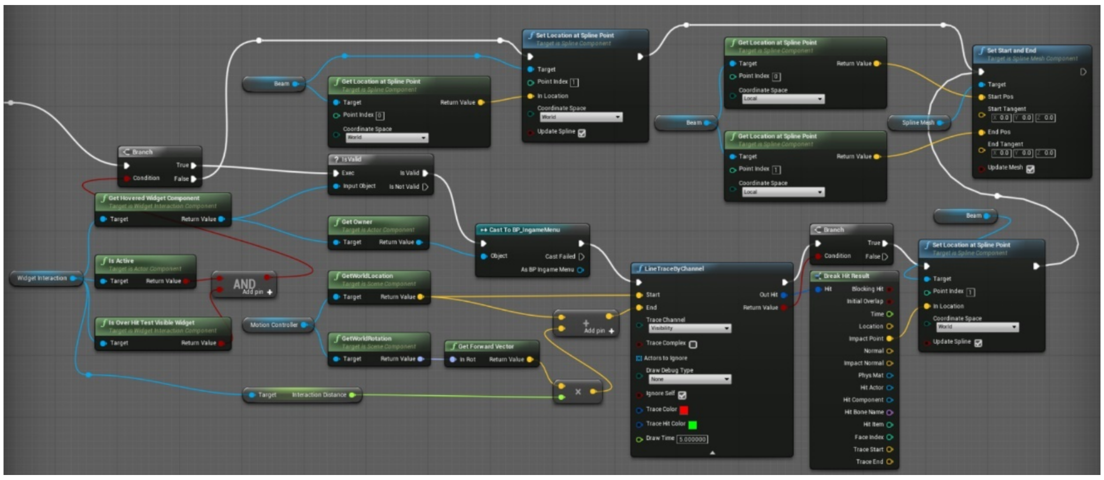


Figure 9-50: Blueprint "BP_MotionController": Laser spline stretches from motion controller to the widget once the widget is hit by the laser, otherwise it shrinks so that is not visible

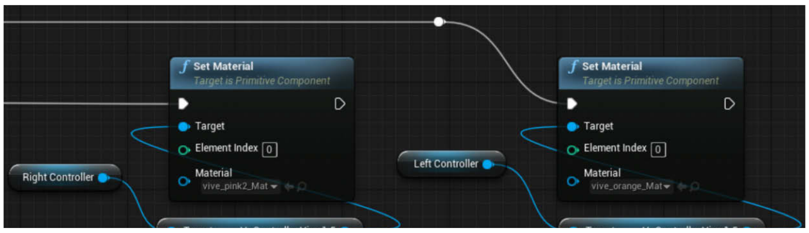


Figure 9-51: Blueprint "MotionControllerPawn": Allocate the pink and orange painted logo texture to the respective controller

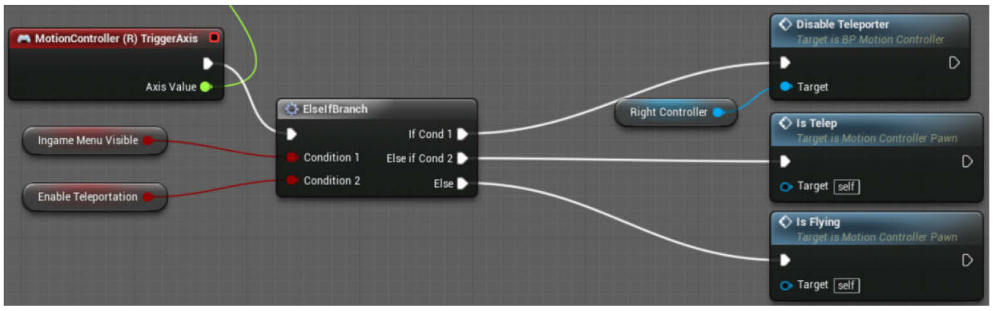


Figure 9-52: Blueprint "MotionControllerPawn": The motion controller trigger axis handles the flying movement which is disabled once the menu opens or the teleportation mode is switched on

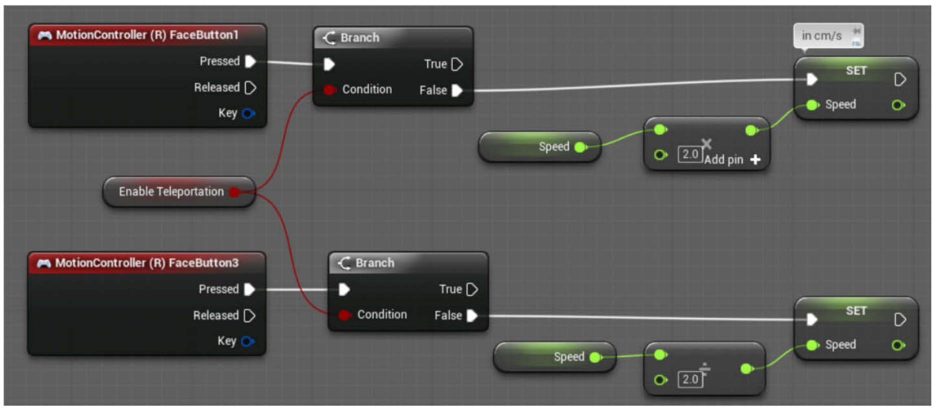


Figure 9-53: Blueprint "MotionControllerPawn": Right face button 1 and 3 control the speed of the flying movement by doubling or halving it. The speed can be controlled only when the teleportation mode is disabled

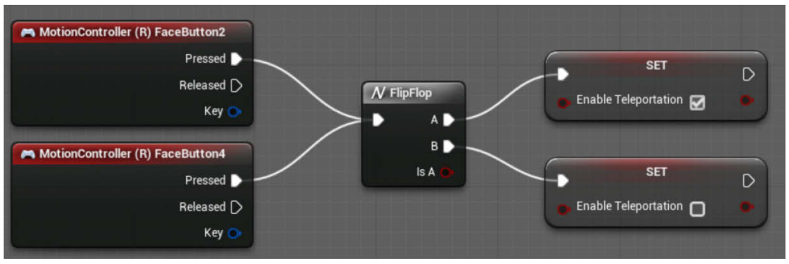


Figure 9-54: Blueprint "MotionControllerPawn": Right face button 2 and 4 toggle between the flying and teleportation movement

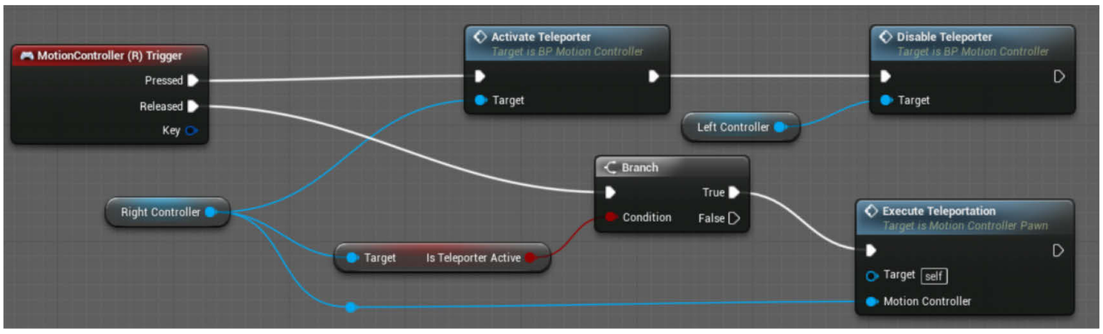


Figure 9-55: Blueprint "MotionControllerPawn": The right trigger will activate the teleporter while pressed and when the trigger is released the teleportation is executed

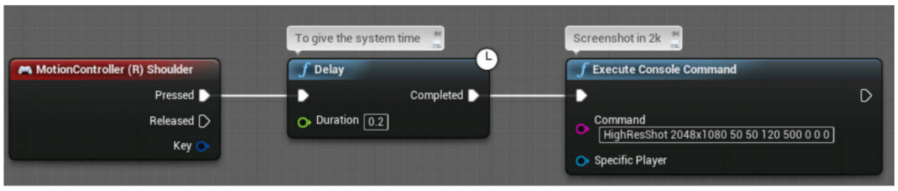


Figure 9-56: Blueprint "MotionControllerPawn": The right shoulder button executes the screenshot command

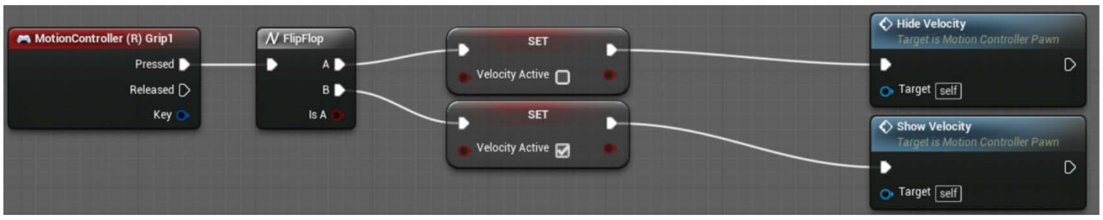


Figure 9-57: Blueprint "MotionControllerPawn": The grip button toggles the visibility of the locomotion display

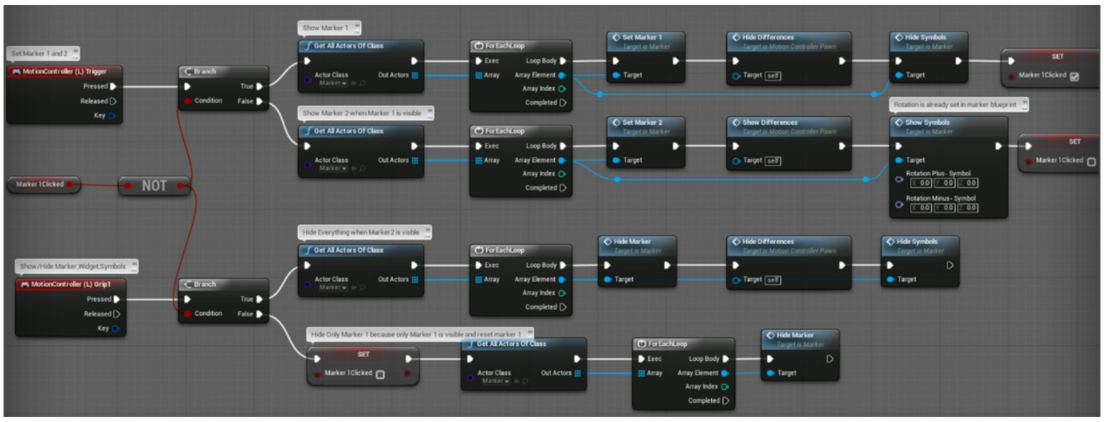


Figure 9-58: Blueprint "MotionControllerPawn": The left trigger sets two measurement points onto the landscape. While the first point is set, the height difference and slope display vanishes as well as the plus and minus symbols. While the second point is set, the height difference and slope display appears as well as the plus and minus symbols. The grip button hides the height difference and slope display

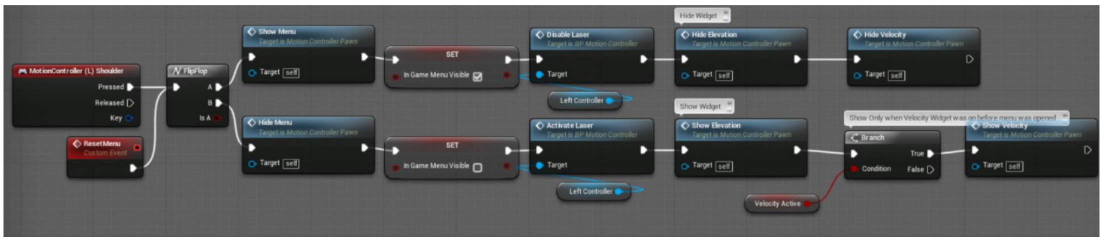
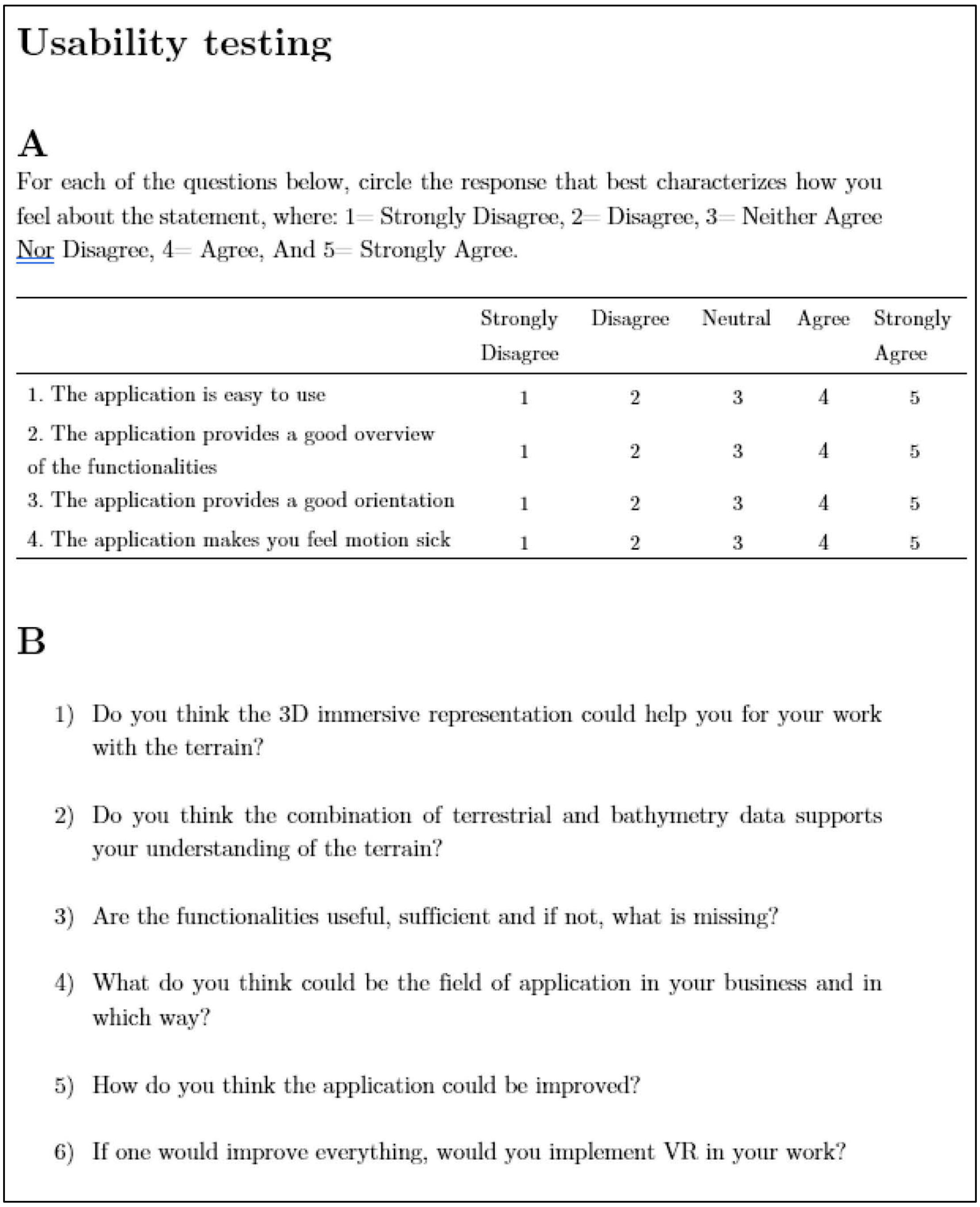


Figure 9-59: Blueprint "MotionControllerPawn": The left shoulder button opens or closes the menu. While the menu is opened, the left hand laser is being disabled and both widgets which are attached to the controllers are disabled as well. Once the menu closes, everything will be set back to the default state

ID	Coordinates	Elevation of DEM [m]	Elevation within VR [m]	Difference
1	71°15'42.12"W; 70°22'22.10"N	-5	-5	0
2	67°16'27.65"W; 70°58'06.66"N	-847	-847	0
3	66°27'41.60"W; 70°17'47.07"N	-125	-125	0
4	70°20'25.04"W; 69°43'18.79"N	805	805	0
5	70°00'42.47"W; 70°19'33.08"N	773	773	0
6	69°02'20.24"W; 70°28'29.52"N	207	207	0
7	68°03'04.16"W; 70°37'05.98"N	-240	-242	2
8	69°43'59.72"W; 70°07'20.69"N	-317	-315	-2
9	67°47'21.55"W; 70°24'41.83"N	-88	-86	-2
10	69°25'00.31"W; 70°19'09.75"N	266	266	0
11	68°26'23.11"W; 70°27'53.90"N	25	25	0
12	69°05'26.77"W; 70°04'26.61"N	915	916	1
13	68°07'25.46"W; 70°13'03.76"N	171	170	1

Table 9-1: Elevation differences between the DEM and the VR application



Usability testing

A

For each of the questions below, circle the response that best characterizes how you feel about the statement, where: 1= Strongly Disagree, 2= Disagree, 3= Neither Agree Nor Disagree, 4= Agree, And 5= Strongly Agree.

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
1. The application is easy to use	1	2	3	4	5
2. The application provides a good overview of the functionalities	1	2	3	4	5
3. The application provides a good orientation	1	2	3	4	5
4. The application makes you feel motion sick	1	2	3	4	5

B

1) Do you think the 3D immersive representation could help you for your work with the terrain?
2) Do you think the combination of terrestrial and bathymetry data supports your understanding of the terrain?
3) Are the functionalities useful, sufficient and if not, what is missing?
4) What do you think could be the field of application in your business and in which way?
5) How do you think the application could be improved?
6) If one would improve everything, would you implement VR in your work?

Figure 9-60: Questionnaire for usability survey