Visualizing our galaxy using Virtual Reality and big data

Simon Holk

DEPARTMENT OF DESIGN SCIENCES
FACULTY OF ENGINEERING LTH | LUND UNIVERSITY

2020

MASTER THESIS

Visualizing our galaxy using Virtual Reality and big data

Simon Holk

LUND
UNIVERSITY

Visualizing our galaxy using Virtual Reality and big data

Published by
Department of Design Sciences
Faculty of Engineering LTH, Lund University
P.O. Box 118, SE-221 00 Lund, Sweden

Subject: Interaction Design (MAMM01)
Division:
Supervisor: Mattias Wallergård
Co-supervisor: Oscar Agertz
Examiner: Joakim Eriksson

Abstract

Big data is becoming ever increasingly important in every kind of scientific field, and data visualization is a big part of gaining an understanding of the data. Exploring data builds on intuition but exploring some data on a 2D screen can remove some of the intuition that is gained by exploring in 3 dimensions. Navigation of data visualization programs are often complex and slow to move through. With the rise of VR, we can explore data faster with greater intuition. The process for developing the system was done iteratively starting with a meeting discussing the current flaws of visualization methods and what one would like to see in a Virtual Reality tool. Then the iterative process proceeded by implementing the features and improvements discussed. The tool was then tested by the targeted users and then discussion was held on what can be improved. This continued until the final testing phase in which targeted users evaluate how the tool compare to other more standard 2D-tools used in their field.

Keywords: Virtual Reality, data visualization, big data, galaxy

Sammanfattning

Big data blir alltmer viktigt i alla fält som handskas med data och visualisering utgör en stor del av att få en förståelse av data. Att utforska data bygger på intuition men att utforska data på en 2D skärm kan försämra intuitionen då en del data naturellt befinner sig i 3D rymden. Navigationen av visualiseringsprogram är ofta komplext och långsamt att manövrera. Med det ökande användandet av VR så kan vi utforska data snabbare och få en bättre förståelse. Processen för att skapa systemet genomfördes iterativt med ett möte som start för att diskutera brister med visualiseringsmetoder idag och vad man skulle vilja se i ett Virtuell Verklighet-verktyg. Den iterativa processen fortsatte genom att implementera funktionerna och förbättringarna som diskuterats på möten. Verktyget blev sedan testat av de tänkta användarna och diskussioner hölls därefter om vad som kan förbättras. Denna process fortsatte tills sista test-stadiet där de tänkta användarna testade och utvärderade hur verktyget står sig gentemot andra standardverktyg i 2D.

Nyckelord: Virtuell verklighet, big data, galaxer

Acknowledgments

I would like to thank the VR lab in Lund university for their enthusiasm and dedication to VR which inspired me to do better. I would also want to dedicate a big appreciation for my supervisor Mattias Wallergård which has been following along patiently throughout this long process and getting the ball rolling at multiple occasions. I also want to thank him for bringing me in to this project. His dedication to his students is remarkable and I still remember my first lectures with him to this day and how he managed to spark my interest in design science, something I had no interest in before starting university. I would also like to thank Joakim Eriksson for the advising and help he has given me throughout my last two years of university. The VR lab would not be the same without this duo. I would also like to thank Oscar Agertz for great lectures about space and his help in testing my prototypes. I enjoyed your and Mattias talks about old pop music, movies and experiences. Also, a thanks to Andy Tang for following me along on my VR adventure. Finally, big thanks to Henrik Garde at the Humanities Lab, Lund University, for contributing with a solution, of which the rest of the application has been built upon.

Stockholm, March 2020

Simon Holk

Table of contents

List of acronyms

FoV	Field-of-view
HMD	Head-mounted-display
VR	Virtual Reality

1 Introduction

In this section I will go through the background and development of Virtual Reality and big data. It will thereafter end with the goal of the study.

1.1 Background

In the 20th century Virtual Reality (VR) as we know it came under development starting with the Sword of Damocles 1968 (Brown University, 2019). VR is an immersive artificial environment that can emulate the world or even completely different environments making interactions similar to interactions in real life. With the rise of complex technologies like computers there came a challenge in how to navigate these kinds of environments. It has seen continuous improvements from being console based until the WIMP paradigm which was a big step in human computer interaction (Booth, 1980). There is still a lot that can be done to improve the human computer interaction and with the rise of VR there also arise challenges but also great opportunities in the way we interfere with software and entertainment. VR saw a lot of development throughout the century but limited by the capability of the technologies at that time it never really got to the point in which consumers could see themselves utilizing the technology. In the end 90s the movie "The Lawnmower man" was created which showed the potential of where VR was heading but the 90s and the century still ended on a sour note because of the quick capitalization of the hype. Hardware like Virtual Boy was far from the reality that the movie provided (Foiles, 2011). The VR craze that developed by that time was laid on ice while development was indirectly and directly happening in the background.

1.2 The state of Virtual Reality

With the 21th century the computing and screen technology has matured enough. VR devices are now more advanced but also more affordable. Companies like Facebook, Sony, Google, Samsung, and HTC have put tons of investments in VR in which Facebook invested 2 billion dollars buying Oculus in 2014 (Cipresso & Giglioli & Raya & Riva, 2018). New applications like games have seen a rising utilization of VR, one which are Beat Saber which had a very successful release. As with most newer technologies the entertainment business has been fast to jump on the use of VR. Most experiences are therefore associated with gaming, however there are other fields expanding such as education (Domingo & Bradley, 2017), medicine (Li et al., 2017), elderly care (Neri et al., 2017) and data visualization. Thanks to the advances in affordability, VR has gained a boost in interest as a research tool (Castelvecchi, 2016).

1.3 Big Data

Another field that has arisen in recent years thanks to computational power and the internet are big data. (Sheng & Amoah & Wang, 2017). One of our biggest problem with all this data is how to make sense of it all. This is done in various fields like in machine learning in which you transform the representation of data into something more understandable, but one can also utilize the human via data visualization. With data visualization one can facilitate understanding of the data and communicate these understandings (Allen, 2018).

Big data gives us many great possibilities but also many challenges when it comes to human computer interaction.

1.4 Data Visualization for research

Even with the rise of 3D and VR visualization the 2D screens still see a very large use and researchers have questioned the viability of utilizing VR for data visualization since 2D visualization seems good enough for the job (Akpan & Shanker, 2018). With fields like machine learning growing at an incredible rate visualization has also become an important part in this field for tasks like feature engineering. While machine learning can help you represent data in different ways to solve a problem the data visualization is often used as a tool to get key insights into the data. When exploring data, one often needs a basic understanding of the domain to find new understandings from the data. Data visualization has two main purposes; sense-making and communication (Soegaard & Friis Dam, 2013). In this thesis I focus on the sense-making aspect of the visualization process.

1.5 Previous work

There have been comparisons made between 2D and 3D visualization in which the conclusion has been that both are valuable tools (Akpan & Shanker, 2018). In this thesis, more focus will be put on the exploration of visualization in VR.

1.6 Purpose and Goal

The aim of the master thesis is to take an explorative approach on how VR can be used as a tool for big data visualization. How the process can be made in such a way that it can become a valuable tool set both in usefulness, but also in a usability sense. The hopes are that it can show the usefulness of VR as a tool in everyday work such as in visualization and also give ideas for visualization of big data. The main questions to be answered is:

- How can VR be used to navigate through a big amount of data?
- How does VR as a tool distinguish itself from 2D visualization tools?

1.7 Limitations

While the VR application developed is supposed to work for any kind of data that fits into the 3D realm, the data used in this thesis will be from the field of astrophysics. Other forms of data can be mentioned but the extensive testing and development will revolve around astrophysics data.

2 Theory & Technical Background

The second chapter will give a brief presentation of the theory used in this paper to give a necessary understanding of the process and the following result.

2.1 Virtual Reality

Virtual Reality (VR) in general can be specified as a computer-generated virtual world involving some kind of interaction. VR technology is constantly advancing and can be immersive to various degrees. For this thesis the VR device of choice will be a Head-mounted display (HMD). As the name implies an HMD is a display mounted in front of the eyes of the user to simulate being in the virtual environment. An HMD can be tracked allowing different degrees of freedom (DoF) of movement where 6 DoF basically follows normal human head movements. (HTC Corporation 2020)

2.1.1 **HTC Vive**

In the development of the application various HMDs have been used. One of the main one has been HTC Vive. Due to the different HMDs being similar in nature knowing how HTC Vive works will give the reader an idea how the rest of the headsets works. The components HTC Vive use which can be found at (HTC Corporation, 2020) is:

- 1 head-mounted display - The display attached to the users face to put them in the virtual environment simulated in 6DoF.

- 2 hand-controllers - Controllers that are tracked with 6DoF allowing the user to interact with the environment. Some main buttons are the “trigger button” used by the index finger and a touch pad used by the thumb moving in 2D.

- 2 lighthouse beacons - Set up to allow tracking of the HMD and controllers. As long as the user stays inside the range of the lighthouses, they can experience room-scale tracking. Note that some VR devices can differ here using so called inside-out tracking which let them be tracked in room-scale without the use of lighthouse beacons. For example, Samsung Odyssey uses inside-out tracking.

This is powered by a high-end computer allowing for smooth tracking of 60 FPS or above which is preferable for a pleasant experience.

Figure 2.1: HTC Vive headset, controller and lighthouse in the background. (ETC-USC, 2016)

2.2 Unity

As engine for the VR application Unity has been chosen. Unity is a widely used 3D engine made for game development but also other uses that involve graphical rendering. Unity was chosen due to the following reasons:

- It allows rapid prototyping due to ease of use
- The authors' past experience in Unity
- Simple cross-platform support allowing a wide array of VR headsets for testing.

2.2.1 **Vertices, & Meshes**

The VR program works mainly with vertices but also meshes to display data. Vertices are points in space that include info such as position and color. It also has information such as the normal, but due to not being needed for our visualization, an abstraction has been made by not using the normal. Vertices can be connected with each other using edges. These edges in turn can be “filled in” basically creating a surface called a face. These three ingredients together create a mesh (Blender). These terms are common in game development and one of the main problems working with meshes is to keep the geometric objects complexity down which entails using as little vertices, edges and faces as possible while not losing fidelity of the object. In data visualization meshes can more commonly be used with fewer data points while if you wanted to display millions of data points performance would quickly degrade (Unity). In figure 2.2 you can see how meshes are built out of the different components.

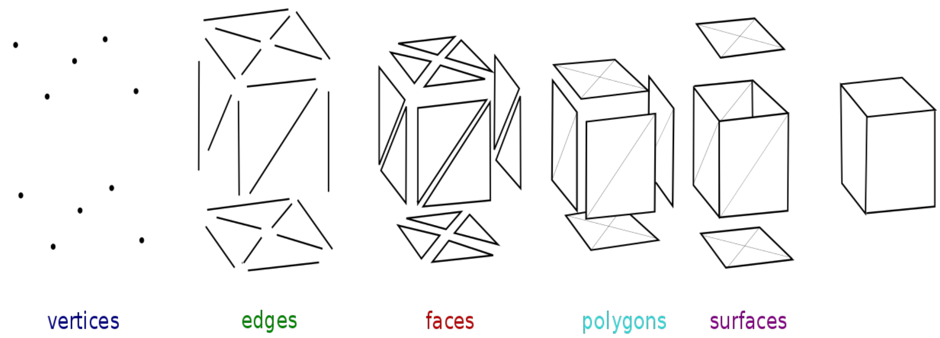


Figure 2.2: Image showing vertices, edges, faces and how they are combined into one object. (Rchoetzlein, 2009)

2.2.2 **Gameobjects**

In Unity everything is made up of gameobjects, which can be seen as any other objects in programming languages. These gameobjects can have components which can be simple scripts giving behavior to the object or storing values, rendering meshes, physics behavior and more.

2.3 Tabular data

The data that the VR program is accessing has to be stored and, in this case, it was stored in a tabular fashion. Tabular data means that data is stored in different rows. Each column represents a property of said object. A property can also be called variable. The rows are a collection of those variables for a single object that we call “data point". A property of tabular data is that each record shares the same set of variables. It’s one of the most simple and oldest form of storing data (Krishnamurthi & Lerner & Politz, 2016).

2.4 Pointcloud

To visualize the tabular data in 3D, the program uses the Pointcloud technique. This technique functions by using the variables of the tabular data to map it into 3D space. This works most naturally with data that is 3D in nature and have variables that represent 3D space. The tabular data is fed into the Pointcloud algorithm which will then pick the three given columns representing positions and display a point at that X, Y, Z-location (Levoy & Whitted, 1985). By displaying only a point in space one will display the data point in its smallest reasonable form saving rendering performance which is crucial for VR applications. Nothing stops the algorithm from being extended and instead of creating data points one could create gameobjects with a mesh to display more data. Due to the nature of vertex points being just a point in space there is limited ability to display extra information about variables. Points can be displayed in different colors though and can therefore be used to visualize extra data that is not 3D.

2.5 Exploratory design methodology

Due to the project being exploratory in nature, the evaluation of the program during the development process has been crucial to gain an understanding of its usage. Most of the results comes from the discussions and evaluations that has been held throughout the process.

2.5.1 **Focus group**

One of the main evaluation forms has been a less formal focus group. Focus group can be seen as an interview/discussion in group letting me collect qualitative data. Focus groups are great for bouncing ideas and getting information from different perspective as the group often, and in this case too, has people with different experiences (Blank, 2020).

2.5.2 **Observation**

Another evaluation form that was used was observations. This basically entails letting the user test new prototypes while being observed. It's important that the observers are focused and have time to write down observations such as flaws or new ideas. It's a great way to get information that might not usually get conveyed just from a conversation with the user.

2.5.3 **Bodystorming**

Bodystorming helps in the design of new ideas by doing physical activity. This lets you physically experience what you're about to design that might give new ideas and realizations that might be hard to find in less physically intense design methods (Barry, 2010). With VR it becomes simple, one can just grab a pair of controllers, close the eyes and imagine the interaction.

2.5.4 **Interaction design**

Throughout the thesis I have followed interaction design principles and user-centered design, which means a heavy focus on user involvement throughout the whole development process and regularly trying to understand what the user needs with various methods. In user-centered interaction design there is usually four main stages that one goes through iteratively during development (Rogers & Preece & Sharp, 2011):

Identifying needs

In order to design something that supports the user in their domain you first have to understand what they want. This can be done in various ways such as interviews, focus groups, surveys, observations, etc. Without understanding the users need, one risk to design something that nobody finds any use for.

Create design alternatives

This involves finding ideas on how to meet the users' need. There are various tools to specify down design alternatives. One can do conceptual design which entails what the products should do, behave and look like. Another part of this step is the physical design for example colors and icons of menus.

Building interactive prototypes

Once I know the user's requirement and have a design in mind, I can start prototyping an interactive experience that the end user can interact with. The prototypes can be in various degrees of details while in this project the implementation is often higher fidelity being implemented straight into Unity itself.

Evaluation

This is the process of determining the usability of the product and if it fits the users need. The purpose can be to see if the design alternative was actually a good solution, but also to evaluate that the thing the user claims as a need actually is a need because sometimes the user can have a hard time knowing what they actually want. The evaluation step is important to steer the process in the right direction and demands a heavy user involvement.

These steps are repeated iteratively. Doing the process iteratively let us discover problems at an early stage without spending countless of resources in improving something that was not important. New ideas can also spring from trying out newer versions of a product. Design are rarely on the mark the first time and has to be constantly evaluated by the user who will in fact use the product in the future.

3 Method

This chapter goes into the process in the development of the visualization tool and also the different stages

3.1 Survey

A survey of the field of VR but also visualization was done to gain a solid understanding in how the field is currently looking and the challenges that are involved in VR visualization. Some of these studies are displayed as references in which you can dig deeper into towards the end of the paper.

3.2 Targeted user

For this project the targeted user has been set to be people working with astrophysics on an expert level. The user should have used some sort of data visualization technology before. A basic understanding of visualization is needed, otherwise it will be hard for the user to determine the usefulness compared to 2D visualization. This report has focused on astrophysics PhD students and professors, but nothing stops the program from being used on other expert groups in the future. The users experience with VR range from first time users to those who have used it a few times.

3.3 Initial meeting

This meeting focused on how visualization of big data works as it is today and what the possibilities are in the field of VR. This was done with the insight from different field experts in fields such as humanities, design sciences, cognition, and astrophysics. The data that was decided as a focus point was astrophysics data for which the astrophysics department was responsible for. It was also decided to start the development process by building upon an already started prototype used for visualizing archeology on a 2D screen using Pointcloud in Unity. And from there remake it into VR. It was also discussed about the usability issues of regular 2D software and there were key insights given into how visualization was already used for sense-making for visualization of galaxy data.

3.4 Development process

The process was done in an iterative fashion, which can be broken down into three main stages. The two first stage was worked on concurrently during meetings with the focus group and test user. The first stages involve idea generation and planning, and the second stage involve evaluation and improvements. Regular meetings were held in which I was supposed to supply new versions of the software. The purpose was to evaluate the new features and improvements that had been made on the software by the domain expert. The other purpose was the planning and idea generation stage which was the planning of new features that seemed to be missing based on the ideas generated.

The third stage was the development process in which I designed and implemented the software while at the same time doing extensive testing. Some of the testing overlapped with the first stage since some bugs are bound to show up with more users testing the software.

3.4.1 **Evaluation & improvements**

During the evaluation I mainly used observations and focus group discussions with at least one domain expert. This was done after each iteration of the program was done to let the users test the new functions. First observations was made by letting the test person try out the program. It usually followed these steps in the observation process. The user is immersed in the program and freely explore for a while trying to gauge what has been added. This was done to see how the more subtle changes was received and talking about them immediately could influence the testers opinion. The test person was encouraged to speak out about the experience they were having and ask questions when needed. After this the user went through the new features trying them out, again speaking their thoughts while comparing how they would usually do in a 2D program and how this feature does it different. At this point new ideas started to pop up as the tester gets accustomed to the program and I have made enough observations about potential pain points and wants. The testing was mostly done by the domain expert and sometimes done by other participants. Some with expertise in VR but not in the target domain, and some with no experience with VR but work in the target domain. Since exploring data can be seen much as a pipeline process it’s crucial to keep track on the flow. More so since current VR-devices have limited inputs and data visualization tends to have a lot of features. Observations are written down during and after the observations, taking what has been done in this step into the idea a planning stage.

After the test and observation there is discussions held about how well the features add to the whole experience and if some feature can be improved. This was done most of the time with a domain expert, the author himself and experts in various fields ranging from design science to social sciences. The discussions were done in free form format with some steering of the directions at times. This because the talks often tend to go towards idea generation, so to evaluate what has to be done probing questions was asked to evaluate how the new changes fit into the toolset for the domain expert.

3.4.2 **Idea generation & planning**

This phase is very interconnected with the evaluation phase as the test users tends to find out new ideas and wishes throughout the evaluation process. Therefore, an idea generation stage is held afterwards in which there is held discussions about features that the test person felt was missing. Except for generating new features there is also discussions about ways to interfere with the tools in a way that fit the VR medium. Throughout the phase various tools to help with design generation was used such as whiteboards to write down and illustrate ideas and body storming to try to map out interactions that can work for VR. VR and bodystorming has a very strong synergy which lead to bodystorming being an active part throughout the whole meeting phase. A new meeting is then generally planned, and improvements are prioritized to decide which should be in the new release until the next meeting.

3.4.3 **Development phase**

In this phase the development of the software began with implementation of the improvements and fix the problems in the Unity prototype that was decided upon in the meeting. This phase typically lasted for around 2 weeks. The focus throughout this phase has been mostly on implementing new features which entails both prototyping the interaction and implementing it, but also to ensure that the software remains optimized.

4 Result

Throughout the design part of this thesis the process in developing the visualization software is displayed.

4.1 Initial meeting

It was decided that I would use a software that was already made in Unity for visualizing data, but which had no VR-support at that point. There was a thorough review of the program to gain an understanding how it visualizes data, the underlying code and to gain an idea on how the work will proceed.

Since it was about astronomy data a short session was held by the astrophysics expert. Here it was shown what kind of data is interesting and why it’s important to visualize this kind of data to gain an understanding of our galaxy. Some focus was put on dwarf galaxies. For example a theory is in gaining an understanding of these galaxies, one could gain an understanding in how our own galaxy was created.

It was also decided in what kind of speed I would go throughout the project, setting bi-weekly meetings for evaluation and discussion. The first improvements were decided which was to add VR-support and a basic grab function to navigate the galaxy.

Figure 4.1: Whiteboarding by a domain expert to explain problem areas.

4.2 Iteration 1

4.2.1 **Improvements**

Here are the improvements that was worked on throughout this iteration:

1. Add VR support to the already existing visualization tool that currently only works for 2D screens.

2. Add support to navigate through the data in VR. This was done by letting the user walk throughout the limit set up by the VR-device. Another function added to help was the use of the grab tool that let the user grab the data.

3. Since visualization tools are often heavy with functions there was a need to make an easy way to navigate. This was done by adding a menu that contains all the tools. This menu was then bound to one of the controllers.

4.2.2 **Development**

Since Unity3D is such a fast and simple tool to prototype most of the implementations will start right away in Unity instead of other Lo-fi prototypes. The program was built on top of an already existing program using Pointcloud technology. The original purpose was to visualize old buildings created by the humanities department at Lund University, now modified to visualize star data.

The first thing that was added was VR support as the old software was only made for 2D screens. The data that was visualized was simulated data of the milky way galaxy stars. With the VR implementation the user could walk through the data and visualize it from different angles and positions. The current visualization capacities allowed the user to see the 3-dimensional positions of the stars in the galaxy represented by vertices in the x, y, z coordinates in Unity.

The user could also see the ages of the stars as it was represented by the colors of the vertex points. The user could also use the controllers trigger button to grab the data to rotate it and move it through the different dimensions. This gave the user a very quick way to look at specific points of the data in a quick manner as it happened at the flick of a wrist.

Throughout the development it was noted that already with only one tool a big chunk of the controller was already occupied. There is only a few buttons on the controller and using the trigger buttons already occupied around 30% of them. So, another thing that was added to future proof the navigation of tools was a menu attached to one of the controllers hence occupying one of the controllers. This was thought of as a good tradeoff since it reduces complexity while also giving us a way to navigate through different tools instead of trying to fit it all into the physical VR-controller.

To make selections on the menu, raytracing was used to let the other controller work as a pointer which can be seen in figure 4.3b. The controller then worked both as a pointer for the menu but also had multiple other purposes depending on the current tool selected. The user could see where they were pointing with help of a red circle that appeared at the end of the raytracing beam when it collided with the menu. The buttons will also be highlighted when hovered over by changing color to a slightly brighter color and enlargement by a small degree.

In figure 4.2 the look of the galaxy data in its current state is displayed.

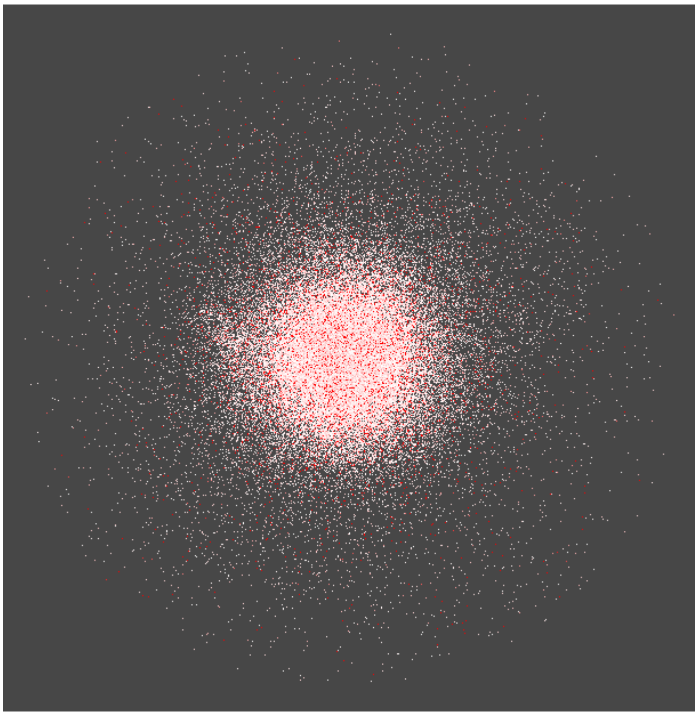
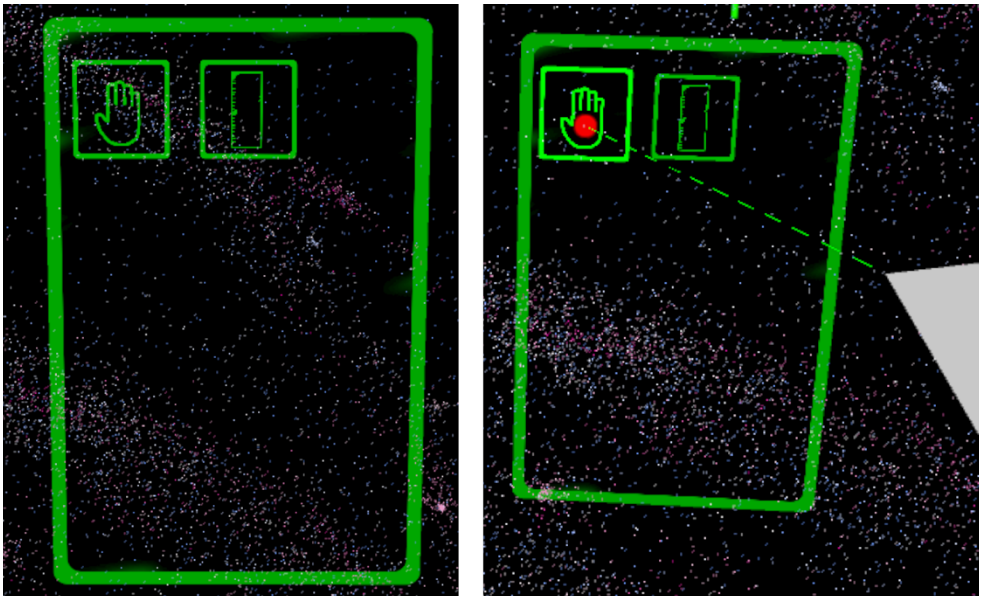


Figure 4.2: The galaxy data represented by a Pointcloud.



(a)

(b)

Figure 4.3: (a) The menu without highlight. (b) The menu with highlight.

4.2.3 **Evaluation**

While evaluating the prototype the following things were noted:

1. The new navigation while simple was a very powerful way to navigate through the data. By closing on the point, you want to explore and holding the data while rotating your hand let you see the data from all different directions in a very effective way.

2. You don't always want to see all data as it might be in the way therefore it would be great to be able to select some of the data and only look at the selection.

3. The menu was interesting but since there was not a lot of functions to utilize it for, there was no extensive testing done. The green color that was chosen felt sci-fi and fit with the galaxy data.

4. It was observed that the navigation of the data was very efficient compared to the already made 2D version.

5. Since one of the hands are used for the menu it means that if you're left-handed it will be harder to navigate. Since all testers are right-handed this was set as a low priority.

6. The users wanted to be able to save different perspectives so that they could export it and show to another user.

4.3 Iteration 2

4.3.1 **Improvements**

Here are the improvements that were worked on throughout this iteration:

1. Added a way to select some specific data in the 3D-space for visualization while hiding the rest of the data. This was done via a so-called box select that works much as a mouse pointer does when you try to drag to mark objects on the desktop but in 3D-space.

2. Added a way to measure distances in the data set by having a 3-dimensional ruler.

3. Added a function that let you mark interesting points by using the existing box selection tool but this time making the box stay in place instead of selecting.

4. Added a new button for tool configuration which lets you change settings on the current tool you are using.

5. Added a new feature in which you can spawn a 2D histogram which shows the distribution of stars based on a specified data feature which at this point was star age.

4.3.2 **Development**

When developing a way to select data it was decided to implement a similar selection tool as already implemented in various 2D user interfaces but make it fit into 3D space (figure 4.4). The user selects the start and endpoints getting visual feedback with a transparent cube to show where the user is marking. A calculation is then made based on the width of the cube and its center point to find all the data points which was inside the area.

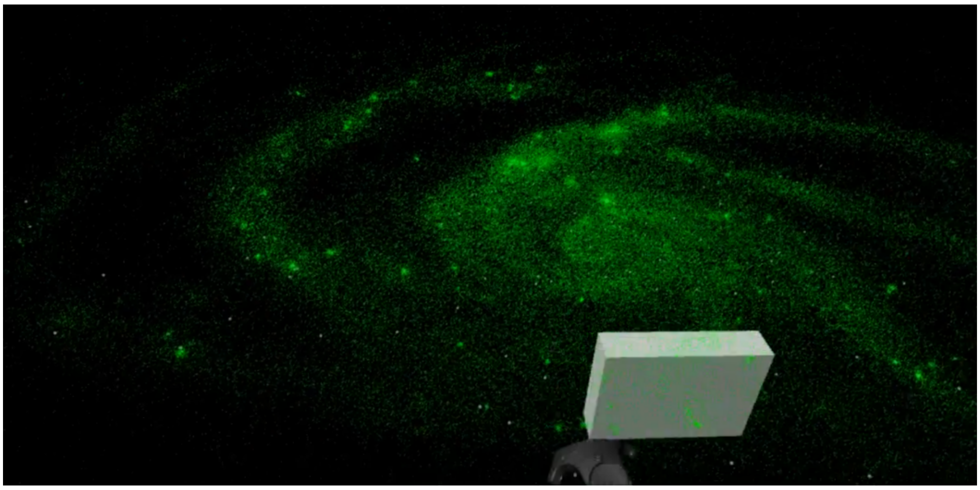


Figure 4.4: Box selection tool used to isolate stars from the galaxy data.

A tool was made called the measure tool. The interaction works almost exactly like the selection tool, but instead of making a selection the distance between the two end points was instead displayed (figure 4.5).

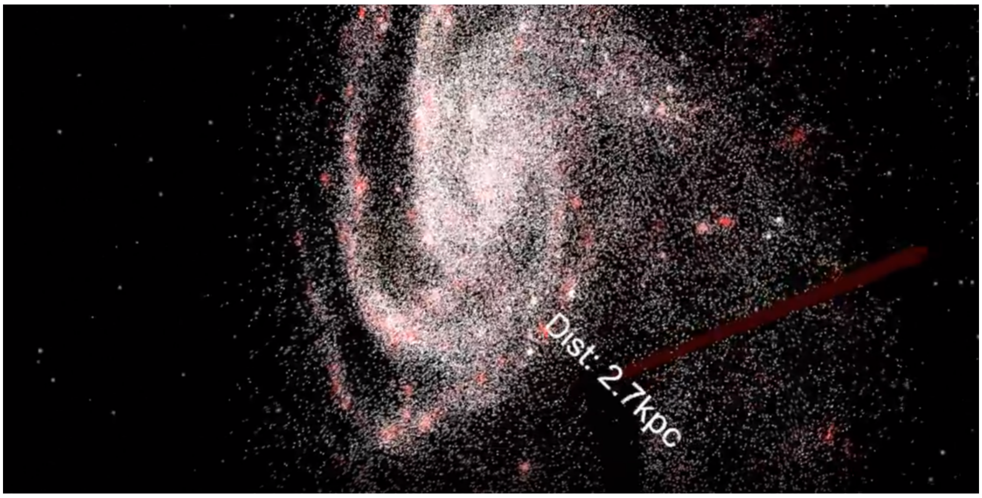


Figure 4.5: Measuring tool used to show distances, useful to measure distances between stars.

A concern when the measure tool was created was the visibility of the text. This is a valid concern in VR in which the text readability varies depending on the HMD used. Therefore, a new menu was added which lets the user customize the current tool selected. For the measuring tool this entails changing color, and size of lines, but also the support to change the metric used (figure 4.6).

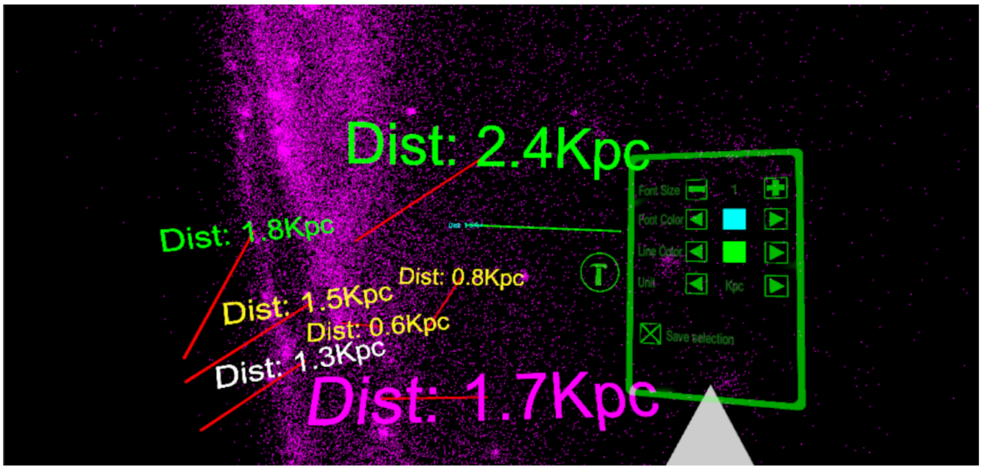


Figure 4.6: Displays customization of the measuring tools giving different colors and sizes.

With the box selection tool and the measurement tool there came up an opportunity to save these values as “marks” to show interesting locations of the data. Therefore, for both these tools there was added an option to save the box marking and the measurements. The usual behavior was for the selections and measurements to just disappear after you released the grip button at the endpoint.

Another way to select the data was also added through a 2D-histogram. With this you could select a range of data values that you wanted to visualize. This in contrast with the box selection which selects based on 3D-space, the histogram lets the user select based on data that is not suited for 3D-selection. In this way you could show only stars that was within a specific age-range. This function can be seen in figure 4.7.

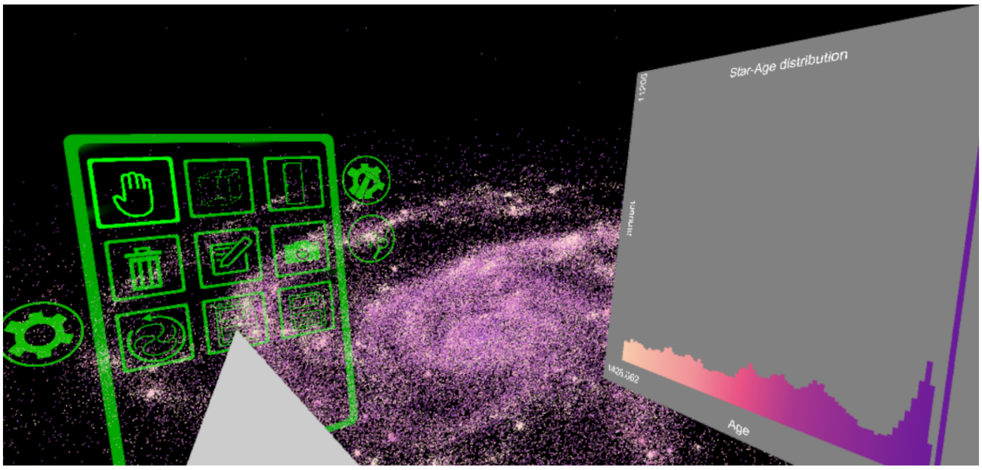


Figure 4.7: A 2D diagram that displays the distribution of the star ages. This can be used to filter out stars outside of a specific range.

4.3.3 **Evaluation**

While testing the prototype the following things were noted:

1. The box-selection was a very intuitive way to select, but some testers had the problem to get a feel for how big the selection was. The nature of this problem could easily be further discussed. Some problems being small data points, clumsy controllers and relatively low fidelity. This could be some of the main causes.

2. The ruler while an enjoyable addition did not provide too much actual value at this iteration.

3. The addition of selection and marking of data naturally lead to the desire of saving these objects so you could show to a research college.

4. The tool configuration was appreciated. There was a problem in that the user didn't really seem to understand that the menu was changing based on the tool. This could be because of the nature that changing menus are

confusing for the user, but also due to the menu not giving enough information.

5. The histogram was extensively used by the testers and seemed to be a crucial addition to the visualization program rivaling the usefulness of the box selection tool which was the tool the testers requested the last iteration.

6. One of the testers were colorblind and suggested different gradients. This does not only help colorblind people, but different gradients can enhance the users understanding of the data. Flipping between different colors and contrasts frequently is an important way to make discoveries.

Below in figure 4.8 the design of the selection tool can be seen in which a cube will be created based on the start point and end point of the controller.

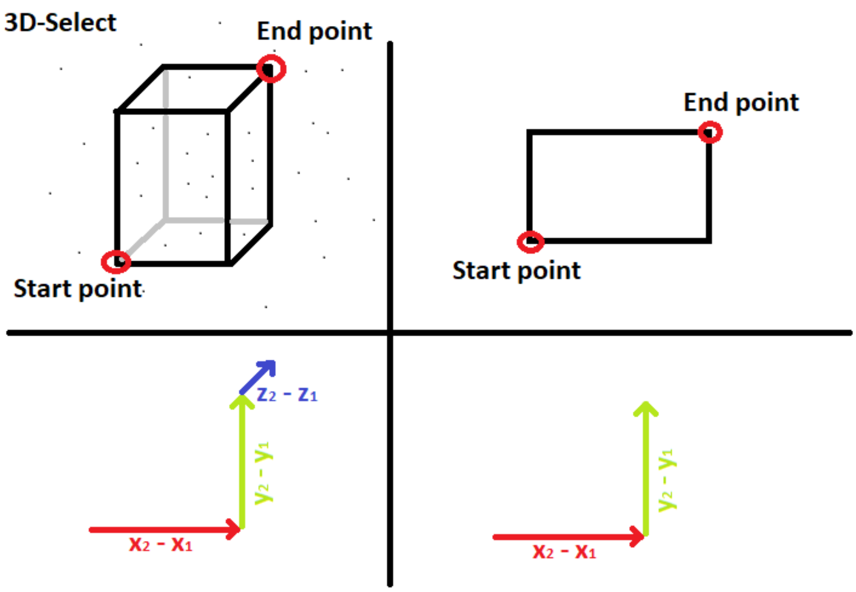


Figure 4.8: The design of the box selection tool.

4.4 Iteration 3

4.4.1 **Improvements**

Here are the improvements that were worked on throughout this iteration:

1. Saving and loading of the marks and measurements in the tool.

2. Deletion of the marks and measurements that is occupying the 3D space.

3. Support for different gradients that can be created inside the Unity editor whenever needed. Added a way to flip between the different gradients on the menu.

4.4.2 **Development**

The task of implementing a save and load function was shown to be a non-trivial task. There are a lot of processes and objects involved in the save and load function and instead of just saving a gameobject as it was you would manually have to "unpack" the gameobject and store its values. To then load you would have to grab these values and re-assemble all the gameobjects that was in the Unity Scene. The complexity would just increase with time as new features was added. While the feature was added and worked fine in this iteration it was also decided that this function would be put on-hold as the benefits was not worth the extra development time cost.

The deletion tool was trivial, and it involved removing saved marks by pressing them with your controller to remove.

Support for new gradients was added. The user could through the Unity editor create new gradients to display their data in. This was done by adding the desired colors and put them in the right order. The program then filled in the in-between colors creating a gradient of the colors. These gradients could then be chosen in the menu inside the program.

4.4.3 **Evaluation**

While testing the prototype the following things was noted:

1. Saving and load features wasn't used except when the testers was told to test the tools. It was discussed that it was a nice feature to show others your discoveries. In a consumer version this is a crucial feature. The purpose of

this thesis though is to test VR for visualization and for that purpose save and load have less meaning as they aren't used in the shorter tests we are doing.

2. Different gradients were greatly appreciated, and it was observed that the testers sometimes got an "aha"-moment when they tried different gradients for the data. This showed the importance of the color representation of data.

3. A feature request from one of the testers was to add a log-scale for the graph as a lot of data is naturally better displayed in a log-scale instead of linear-scale.

4. Up until this point whenever you wanted to explore the data from the beginning you had to restart the program. As a quality-of-life improvement it was decided that there should be a way to reset the galaxy data inside the program.

5. Since the 2D histogram was a success there was some interest in seeing 3D style graphs in the VR environment.

Figure 4.9 displays the use of different colors to display the same data.

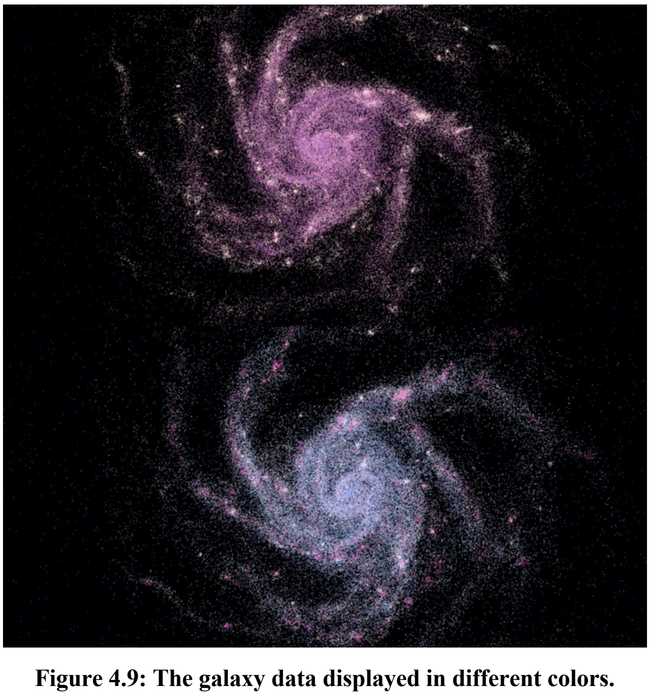


Figure 4.9: The galaxy data displayed in different colors.

4.5 Iteration 4

4.5.1 Improvements

Here are the improvements that was worked on throughout this iteration:

1. A button to reset the galaxy to its original state

2. A way to reshape the 2D histogram to make it smaller or larger to fit better into the view.

3. Added support for a log-scale for the histogram

4. Added a 3D-graph that showed the density of stars based on the X and Y positions. The higher number of stars in a specified x, y-range the taller the staple became.

4.5.2 Development

The 3D-graph is a representation of the density of stars in the x, y-plane. As the development of this feature proceeded it became obvious that not too much details could be displayed using this 3D-graph. The default test size was 100x100 bars and with that details you could see the shape of a spiral galaxy. But this resulted in 100x100 = 10000 cubes which all have 6 faces each resulting in 60000 faces and 10000 gameobjects. The reason for the cubes being game objects was for the purpose of being able to select data just like with the 2D-histogram. This resulted in severe drops in framerate and the future of 3D-graphs was put on hold.

The other implementations were trivial with exception of the log-scale. Here a custom log-scale was implemented, and the creator are still not sure if it’s 100% accurate, but for the purpose of this experiment the accuracy matters less as it’s the functionality that is primary goal to be evaluated. There was no added support to display negative values.

4.5.3 Evaluation

While testing the prototype the following things were noted:

1. Reshaping function of graph was rarely used, the user seemed to rather just move around the graph in 3D-space instead of reshaping the scale of the graph.

2. 2D graphs are often placed right outside the user's field of view and when they must look, they turn their head slightly. When they feel finished with the graph, they tend to just move it away placing it further away.

3. Log-scale seemed to be used more than linear scale.

4. 3D-graph was never really used more than for looking, the program tended to drop frames.

5. The testers wanted a new way to select data as the current way is a bit clumsy and doesn't allow fine details selection. Therefore, to take advantage of the natural movements of the human body a sort of brush select was suggested in which you can "paint" over the data points. This could be a huge advantage compared to visualization software in 2D.

6. The tester wanted to expand more on the contrast of data. Some suggestions were making the data you didn't select transparent instead of disappearing to be able to see the context of which the data belongs to. They also wanted to be able to change the amount of transparency.

7. The menu was really starting to become cluttered and the testers seemed to have problem finding tools as there was no good structuring. There was talk about the overview zoom and filter details on demands which later gave the author the idea of categorizing the different tools in a similar pattern.

Below you can see the result of the iteration. Figure 4.10 displays the new diagram while figure 4.11 displays the log scale of the histogram.

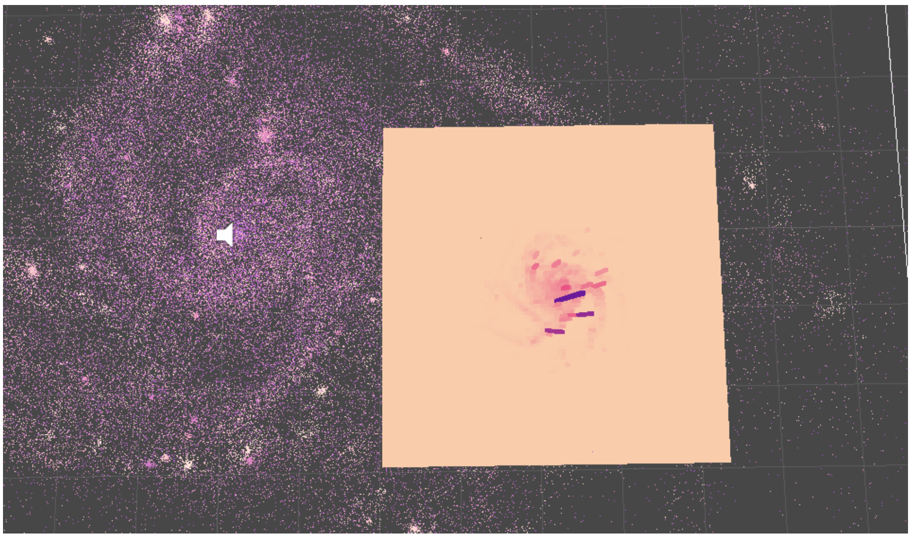


Figure 4.10: A diagram displaying the data distribution over the X and Y planes.

Figure 4.11: The histogram displayed using a log scale instead of linear.

4.6 Iteration 5

4.6.1 Improvements

Here are the improvements that was worked on throughout this iteration:

1. Paint-brush selection letting the user select data based on using a paint brush in 3D space.

2. Added support for transparency in the data points shader.

3. Added an option to let the user just “dim” the data points not selected showing them in a transparent gray scale.

4. Restructured menu by grouping tools into different categories for better usability.

4.6.2 Development

The paintbrush let the user select all data points that are within a certain radius from the controllers' center point. The selection area therefore forms a sphere in which the points are selected. The data points are not actually gameobjects but are stored in arrays in the program. Therefore, the collider can't be used and instead the arrays will have to be searched through to find the data points within the controllers' range.

This can create some optimization problem as the paintbrush is searching through all the data every frame and it also has to know which points it has gone through. The latter is solved by a HashMap which let us make that check in constant time. The former hasn't been a problem yet as the data in the program are relatively small and the graphic performance tends to be the bottle neck, not the calculations under the hood. The search for the data points are done in linear time, one could probably optimize more by sacrificing memory to store the points in three different arrays and sort each by x,y,z and then doing some binary search to pinpoint the data within range. Other changes could be made which might affect the fluidity of the program such as not doing the check every frame. Change the settings so that fixedUpdate happens less often. There is always more that can be done, but since it's not a problem for our scope no further optimization has been done.

The program uses a custom shader to show the color of each datapoint, but it didn't have transparency support, so it was added. You could now dim out the data that was not in your selection to compare the selected and unselected data.

The menu got restructured into sub-menus such as "zoom" where all selection tools were listed, "filter" where the 2D-graph tools were listed and then a general tab for the rest of the tools. At least one further sub-menu could be added, further enhancing the usability.

4.6.3 Evaluation

While testing the prototype the following things were noted:

1. Observing the testers, you could see that the brush was perfect for the galaxy data as you could paint around the dwarf galaxies giving a more fine-tuned selection instead of putting it in a box.

2. The paintbrush was very intuitive and could be seen as an advantage over 2D-screen where a similar complex selection would be harder to make.

3. Things were starting to fall apart, and the program at this point is littered with quite a few bugs which has to be fixed. This is due to so many features that manipulate the underlying data in so many different ways, some might conflict and therefore the data behaves in unexpected ways.

4. Due to the testers' former experience of the program already it's hard to say if the new sub-menus have done their job. This since the testers had already learned the old layout, but the new one changes it around. It doesn't seem to go slower though. The 2D-histogram might even go faster as unnecessary settings that was accompanying the function was removed as they were never used and only cluttered the menu.

5. It was talked about further exploring colors, changing background color being one way.

Below is a set of pictures showing the new features from this iteration. Figure 4.12 shows the new menu layout. Figure 4.13 shows the new dimming function. Figure 4.14 shows the brush selection tool that lets the user paint the data they want to select.

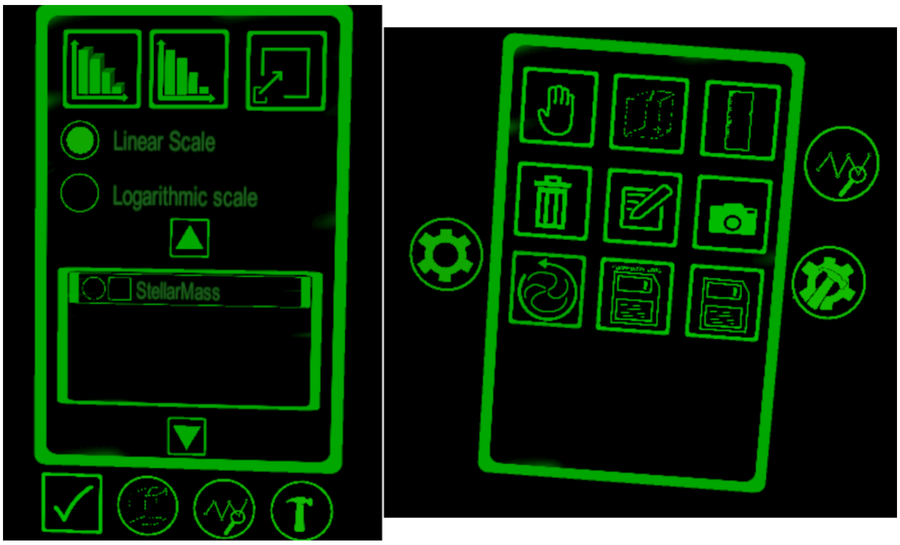


Figure 4.12: To the left is the menu after the redesign while the old menu is to the right.

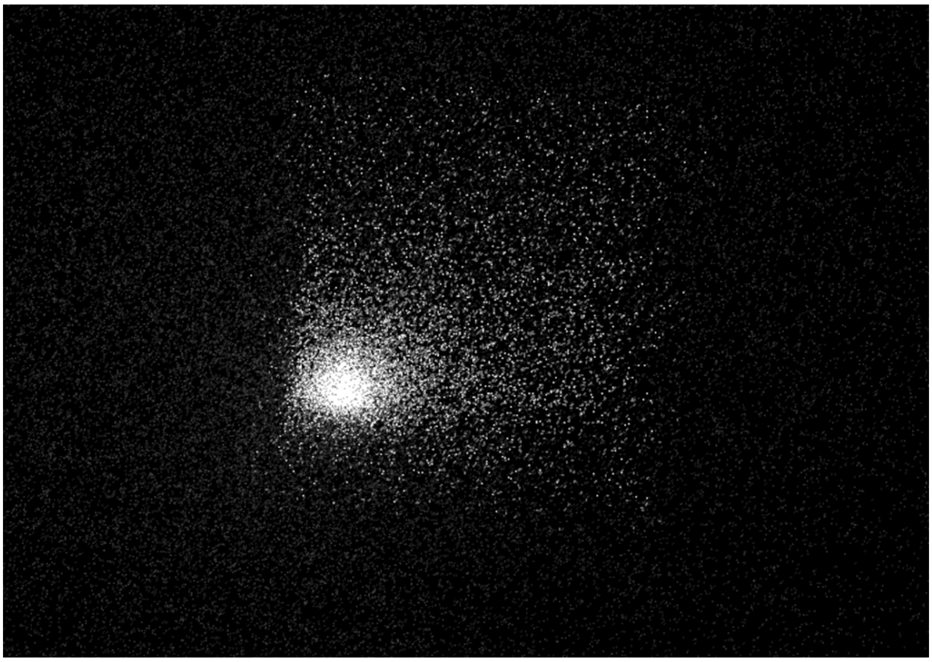


Figure 4.13: The selected data is highlighted while the other is dimmed out to not lose out on the context.

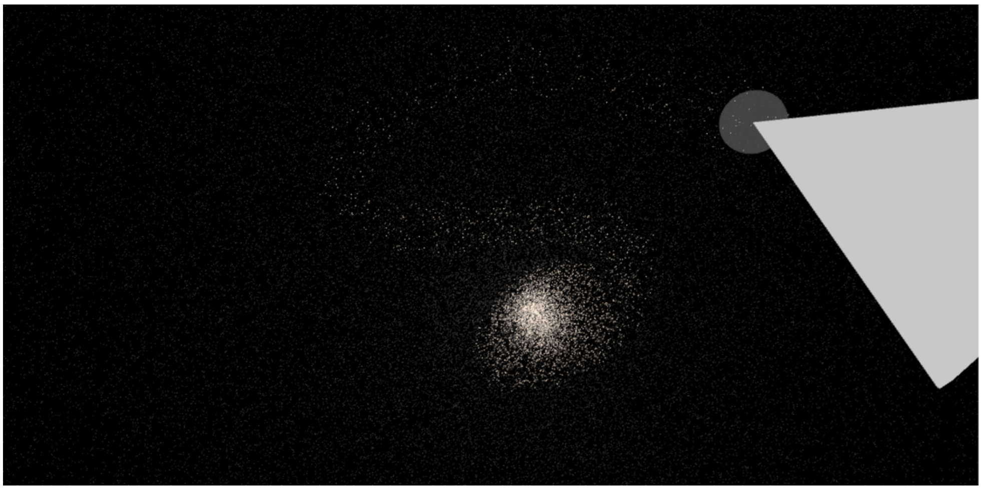


Figure 4.14: The brush selection tool letting the user select data in a non-linear way.

4.7 Iteration 6

4.7.1 Improvements

Here are the improvements that was worked on throughout this iteration:

1. Created harmony between the different functions and restructure the underlying data so no conflicts will occur creating unexpected behaviors.

2. Added an apply button to lock-in some changes to the data. Reset galaxy still goes back to the original state.

3. At this point you could only display the position of the data point and one other variable of the data. This due to the limit of the data being stored as a vertex point only. Therefore, when the number of data points selected was small enough a function was added that turned them into actual gameobjects.

4.7.2 Development

When the data points selected was just a few thousands they got turned into gameobjects. In this way I could display not only one variable in color coding, but also store variables in the size of the gameobject. Further ways to visualize variables could be in the shape of the variables for categorical data and transparency suited for numerical data.

The apply button was a natural extension to the underlying structure fix. Before you had to keep the actual graph in the view to still display only the data in the variable range. Now you could apply this change removing the graphs. Before you also had to make a change in the data before you could actually see the new gradient when you changed, but now when you press apply the gradient will change without the need to transform the data.

One of the structure changes involved the features that made copies of the original data array but transformed based on what the feature did. This copy was originally done for the 2D-histogram because the changes was supposed to be temporary so you could play around with the 2D-histogram with ease. Now instead of having a copy of data, there has been put on constraints only, saving the ranges making a conditional statement.

4.7.3 Evaluation

1. The program seemed to be relatively bug-free and all the major conflicts has been sorted out.

2. Adding gameobjects for smaller subset shows potential for displaying more data about the variables. One could use shape, size, color, transparency and more to display different variables of the stars.

3. Full support for the gameobjects had not been given yet, so it only displayed the same variable for both coloring and size which means you can't visualize multiple variables yet. It still gives more insight into the data.

Below are the end results when smaller subsets are displayed as gameobjects as in figure 4.15.

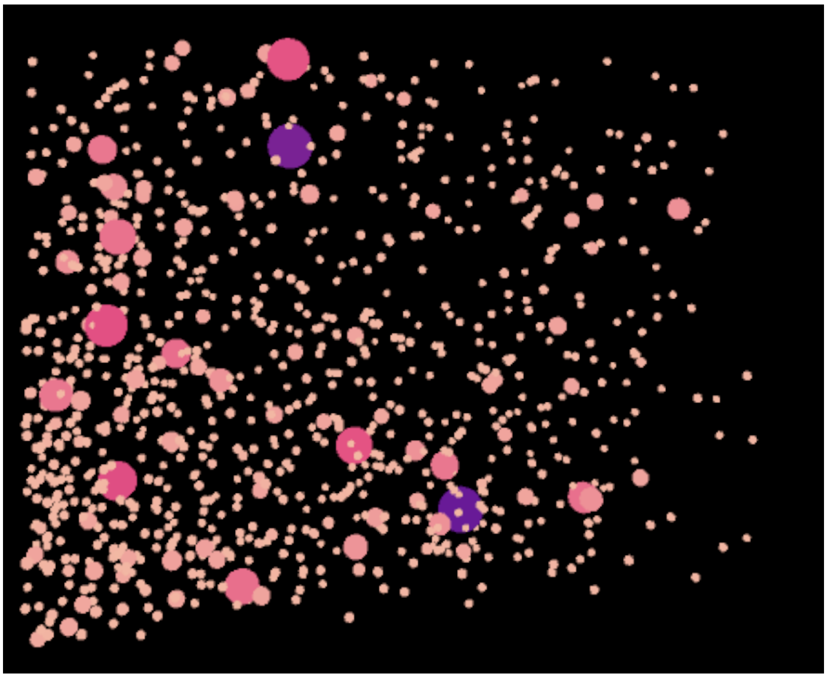


Figure 4.15: When a small enough subset of data has been selected they get turned into gameobjects displaying data in both size and color.

4.8 Final state of program

The final prototype of the program was made up of a visualization environment of data representing the milky way. As proof of concept the amount of data displayed was only 100K, but it can work with a few million data points. Here are the final set of features that was created for the program:

- **Moving the data** - A grab tool allowing the user to grab the data with their hands using the trigger button to move around and rotate

- **Moving around the data** - Allowing for room-scale allowing the user to walk around the data watching from different perspectives.

- **Reset data** - Resets the data to its original form in an instant due to the data already being loaded into the program from file.

- **The Left-hand menu** - A menu used to navigate throughout all the different tools in the user's arsenal moving along the users left hand.

- **Measuring tool** - A tool to measure distances in the 3D space. Working as a measuring tape allowing to measure distances between data points.

- **Gradient selection** - Setting to allow the user to change the gradient of the data displaying it in different color tones.

- **Dimming of data** - Setting to allow the unselected data to still show up but in dimmed form displaying some context for the user.

- **3D box selection tool** - Allows the user to cut in the data selecting only interesting data points to look at while hiding the background noise.

- **Free-from selection tool** - A tool allowing the user to "draw" in 3D space to select data in a very controlled and flexible way.

- **Selection tools settings -** Settings allowing you to instead of selection data, draw out cubes working as a mark to mark interesting points.

- **Annotation tool** - Tool letting you annotate the marks you put out in the 3D space.

- **Deletion tool** - Allowing deletion of marks.

- **Multi-variable support** - Support for storing multiple variables in each data point allowing you to toggle between them to display in the color gradient.

- **2D graph creation -** Creates a 2D histogram to show distribution of the variable chosen for the data points. Later support to reshape the size of the graphs was added.

- **2D graph linear/log scales** - Allow the user to toggle between displaying linear or logarithmic scale of the graphs.

- **2D graph selection** - Allows the user to interact with the 2D histogram to select different ranges of data points to visualize. Allowing filtering of data in a more numerical form instead of geographical.

- **3D graph creation** - Creates a graph showing the distribution of the data points in the X- and Y-axis.

- **Screenshot tool** - Allowing the user to take screenshots of what they see.

- **Displaying of data in gameobjects -** Sets a threshold so that when enough of the data points are filtered out the remaining few data points will be turned into gameobjects letting us display more information of the variables of the data point not only in colors, but also in size.

- **Saving and loading** - Allows saving and loading of the data, although it was created early so support for later functions are not supported.

- **Transparency support** - Added the possibility to use transparency when displaying the data points but also created a tool in which you can increase and decrease transparency.

- **Support for temporary changes and locked in changes** - Some changes are not locked by default allowing for rapid exploration of data, but changes can be “locked in” being applied to the real underlying data. This allows for chain filtering and selection.

- **Undo button (Deprecated)** - An undo button was created to allow for the user to undo what they have done; due to the smaller scope of this project and the maintenance of an undo button it lost support in the middle of the project and was disabled.

4.9 Focus group and expert evaluations

The results written down here can be seen as a summary of discussions held regularly throughout the development process.

VR in its current state works excellent for basic visualization to explore and get a feel for the data being a complement to more detailed visualization in a 2D program.

VR gives a better sense of depth of the data and let you manipulate the data based on depth with more ease than in 2D visualization.

It is easier to get an overview of the data and see interesting points being immersed in VR.

There is a design problem that has to be solved how to fit all the tools needed to do more advanced visualization. New users had some problem navigating all the tools to find what they needed.

Some problem can be seen with smearing pixels mostly because the low fidelity of the older VR headsets, clarity increases significantly with more modern HMDs.

Navigating data geometrically is very intuitive in VR and one can navigate the data in a very fast and precise way.

Rendering more than a few vertices makes the program lag and therefore gives an unpleasant experience. In my case I visualized the Milky Way which has 100 of billions of stars which could obviously not be rendered but just a very small subset. In VRs current state visualizing smaller representative subset of the data might be one way forward.

The paintbrush selection tool is a unique way to select data points geometrically via VR, hard to replicate in a 2D environment. This kind of tools could be explored further.

Showing data to onlookers can be hard as it's hard to compare the view inside of the HMD with what the other observers see on the screen. Some progress was made by adding multiplayer (which was later scrapped) and by doing green screen demos (which are non-trivial to setup).

5 Discussion

In this chapter I will discuss the results and the design process that was used throughout the process

One of the main discussion points in the evaluations was how it was easier to work with depth compared with standard 2D visualization. Standard 2D visualization got the problem that it uses mouse and keyboard. The mouse which is the main interaction tool are constrained to the X, and Y axis moving up and down, left and right, but lacking the ability to move back and forth in depth. Same problem shows up again but to a lesser extent in the 2D display. The 2D display shows everything in a flattened form but it still gives an illusion of depth.

Using VR, the ability to estimate depth increased compared with a 2D screen. Although the screen is still technically in 2D one theory why the ability to measure depth increased could be the ability to move one's head and therefore change perspective slightly. The data points close to the user moves faster, while the further back the stars is the slower, they move, creating a sense of depth (Ayeras, 2014). This in compared with a 2D screen that will always show the same perspective no matter how you move your head. This assumes small movements, but of course larger movements will increase the precision more. Another reason could be the closeness of the screen and that HMDs tends to trick the user to focus in a bit more distance while the focus point on a 2D screen will often end at where the screen starts.

Another advantage of VR is that when the user for example want to make selections of data geometrically, they can utilize the proprioception ability. Humans are remarkable at knowing exactly where their body parts are relatively in 3D space (Walker, 2014). Since selections are done with the controllers in the users' hands and because of the increased depth vision, selecting data based on geometrical depth becomes very natural. A basic box select in a typical 2D program will have a limited X, and Y but in most cases assume an infinite Z axis selecting everything that is inside the range of the X and Y of the box. Selecting in VR allows you to limit the depth you are selecting.

Navigating data in VR is very intuitive and fast to navigate. When presenting the experience to new people they could relatively quickly get the hang of the program and navigate the data. As moving our body and picking up stuff comes naturally ever since we were born this form of navigation works very well and could even be used to target younger users.

Some tools are not created equally though. There seem to be a correlation between the ease of use for tools with how much it mimics the movements and behaviors in the real world. For example, navigating data by picking it up almost everyone could do immediately. Navigating the menus was harder though, I would say that it's not unique to VR as visualization tools can sometimes be complex the more features you add. The main expert user who tested the program could easily pick up the program and start working without much help. This might also show that having some prior experience to visualization even if in 2D can carry over into the VR space.

Some of the problems facing visualization in VR comes from the equipment. This is positive in a way that the technology will always increase allowing for more visualization of more data with time. Some problem with details improved a lot when an upgrade was made to use the Valve Index to look at data.

5.1 Limitations and difficulties

One of the main limitations of the thesis is the lack of quantitative data. Although the study being exploratory in nature which goes hand in hand with qualitative data the author would have liked to do some kind of final test and try to quantify the user experience and the navigation speed of the data. The author has been very lucky though to have frequent access to various people from various faculties and expert users in the covered field to test the program regularly.

Too much focus was put on creating more new features. Some features tried to emulate some things that 2D screens does well and it might have been more fruitful to research some of VR stronger points. Such as utilizing the body as a tool and focus on quality of a very few tools. Most of the features was used once and never again while a few main tools were always used.

Due to VR being a new medium, users not only have to combat learning the equipment and getting used to a virtual environment. They also have to learn the program at the same time which can add additional complexity. For example, an older participant had problem with some buttons and gave up quickly having nothing to do with the program itself. There is also a wow factor for new users giving them a more positive experience immediately because it's their first time.

5.2 Future improvements

As touched upon before more exploration could be done to use the body as a tool to explore the data. Maybe almost completely getting rid of the menu and instead relying on intuition and exploration would be an interesting subject to test. Working on collaborative visualization is another great opportunity for VR.

Making more quantitative tests to see the speed in which a new user can use the visualization tool and use it to accomplish some goal would be interesting to confirm that VR can work faster with more ease.

6 Conclusion

The purpose of this master thesis was to explore how VR can be used as a tool to visualize big data using our galaxy as a reference. I wanted to explore in what new ways one can interact with the data and how it can complement old fashioned 2D screen visualization.

The conclusion can be drawn that there is a lot of potential to use VR for visualization of data. There are some ways to work with the data that can only be achieved through VR which would otherwise face a huge design obstacle working with regular keyboard and mouse. Potential to use intuition and playing with data is big and it might be VRs niche for the short-term while dwelling into the more quantitative analysis with progress in research and equipment.

People getting experience with the visualization program have been very enthusiastic about its future. With the rise of new devices with focus on better ergonomics, more clarity and innovative controllers, VR could be more and more integrated into the research field.

7 Reference List

Akpan, I. J., & Shanker, M. (2018). *A comparative evaluation of the effectiveness of virtual reality, 3D visualization and 2D visual interactive simulation: an exploratory meta-analysis*. SIMULATION, 95(2), 145–170.

Allen, W. L. (2018). *Visual brokerage: Communicating data and research through visualisation.* Public Understanding of Science, 27(8), 906–922.

Ayeras, I. (2014). *Introductory Psychology Blog (S14) C.* Retrieved February 21, 2020, from https://sites.psu.edu/intropsychsp14n3/2014/03/04/motion-parallax

Barry, C. (2010). *Bodystorming*. Retrieved February 21, 2020, from https://dschool-old.stanford.edu/groups/k12/wiki/48c54/Bodystorming.html

Blank, G. *Focus Groups.* Retrieved January 15, 2020, from http://www.cse.lehigh.edu/~glennb/mm/FocusGroups.htm

Blender. *Meshes.* Retrieved January 15, 2020, from https://docs.blender.org/manual/en/latest/modeling/meshes/index.html

Booth, C. (1980). *Alan Kay and the Graphical User Interface*. Retrieved March 14, 2020, from https://web.archive.org/web/20110713232927/http://www.lottiebooth.com/pdf/essay.pdf

Brown University. (2019). *The Present and Future of Virtual Reality.* Retrieved March 14, 2020, from https://blogs.brown.edu/virtualreality/the-present-and-future-of-virtual-reality/

Castelvecchi, D. (2016). *Low-cost headsets boost virtual reality's lab appeal*. Nature, 533(7602), 153-154.

Cipresso, P., Giglioli, I., Raya, M. A., & Riva, G. (2018). *The Past, Present, and Future of Virtual and Augmented Reality Research: A Network and Cluster Analysis of the Literature.* Frontiers in psychology.

Domingo, J. R., & Bradley, E. G. (2017). *Education Student Perceptions of Virtual Reality as a Learning Tool.* Journal of Educational Technology Systems, 46(3), 329–342.

ETC-USC. (2016). *CES2016_HTCVive_Pre_Winters*. Retrieved March 14, 2020, from https://www.flickr.com/photos/92587836@N04/24177102722/

HTC Corporation. *Vive.* Retrieved January 15, 2020, from https://vive.com.

Krishnamurthi, S., & Lerner, S. B., & Politz, G. J. (2016). *Intro tabular data.* Retrieved January 15, 2020, from https://papl.cs.brown.edu/2016/intro-tabular-data.html

Levoy, M., & Whitted, T. (1985) *The Use of Points as a Display Primitive.* Technical Report 85-022, Computer Science Department, University of North Carolina.

Li, L., Yu, F., Shi, D., Shi, J., Tian, Z., Yang, J., … Jiang, Q. (2017*). Application of virtual reality technology in clinical medicine.* American journal of translational research, 9(9), 3867–3880.

Foiles, L. (2011). *Top 5 Hardware Super Fails*. Retrieved March 14, 2020, from https://v1.escapistmagazine.com/videos/view/top-5-with-lisa-foiles/2796-Top-5-Hardware-Super-Fails

Neri, S. G., Cardoso, J. R., Cruz, L., Lima, R. M., de Oliveira, R. J., Iversen, M. D., & Carregaro, R. L. (2017*). Do virtual reality games improve mobility skills and balance measurements in community-dwelling older adults? Systematic review and meta-analysis.* Clinical Rehabilitation, 31(10), 1292–1304.

Rchoetzlein. (2009). *Elements of polygonal mesh modeling*. Retrieved March 14, from en.wikipedia.org

Rogers, Y., Preece, J., & Sharp, H. (2011). *Interaction design*. Hoboken, NJ: Wiley.

Sheng, J., Amankwah-Amoah, J., & Wang, X. (2017). *A multidisciplinary perspective of big data in management research.* International Journal of Production Economics, 191, 97-112.

Soegaard, M., & Dam, R. F. (2013). *Encyclopedia of Human-Computer Interaction*. Aarhus, Denmark: Interaction Design Foundation.

Unity. *Optimizing graphics performance.* Retrieved January 15, 2020, from https://docs.unity3d.com/Manual/OptimizingGraphicsPerformance.html

Walker, E. (2014). *Proprioception: your sixth sense.* Helix Magazine. Retrieved February 21, 2020, from https://helix.northwestern.edu/article/proprioception-your-sixth-sense