FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Interactive Multimodal and Procedurally-Assisted Creation of VR Environments

João Pedro Martins Ferreira

Mestrado Integrado em Engenharia Informática e Computação

Supervisor: Rui Rodrigues

Second Supervisor: Rui Nóbrega

September 21, 2020

Interactive Multimodal and Procedurally-Assisted Creation of VR Environments

João Pedro Martins Ferreira

Mestrado Integrado em Engenharia Informática e Computação

September 21, 2020

Abstract

Virtual reality has become very popular all throughout the late 2010s due to the arrival of better and more affordable technologies to the consumer market, which in turn, has grown massively. This makes it so that virtual reality starts being used across many fields, from education to health, and the creation of virtual reality content spikes in demand. However, as we will show, the creation of virtual reality content has not managed to keep up with these advances, and research into new alternatives is needed.

In this dissertation we explore the gap that exists in the creation of better ways to develop virtual reality content, harnessing the power of virtual reality interaction and readily available systems. More specifically, we focus on how it is possible to help the creation of virtual reality environment using 3D model repositories, voice controls and procedural tools. We strive for a tool that makes the technical process behind the creation of virtual reality environments as intuitive as possible, leaving only the creative process to the user.

Put simply, our research goal is to find out if such a tool could be created that not only made it easier and faster for proficient users to make environment prototypes while instantaneously visualising the final product, but at the same time, made it possible for inexperienced users without 3D modelling abilities and even VR experience to also be able to create their own environments without the need for extensive training.

Our hope is that this project could be a stepping stone for the possible adoption of virtual reality into more domains.

We start by studying the foundations of 3D modelling and figuring what methodologies best suit our use case. We explore existing alternatives to better understand the existing gap in offerings, and speak with 3D modelling professionals to understand their needs and the viability of our project.

Following that, we define the key features of our prototype, introduce how we intend to test it and establish which tools and technologies we will use to develop it.

Next, we analyse the development process, exploring in great detail all the functionalities of our prototype and how to use them, and narrate the whole environment creation workflow using it.

Finally, we explain our testing methodologies, recount our testing procedure and summarize and interpret the gathered results, concluding that our prototype suitably responds to the established research questions.

Resumo

A realidade virtual alcançou uma popularidade bastante elevada nos últimos anos devido ao aparecimento de tecnologia melhor e mais barata no mercado do consumidor, que, por sua vez, tem crescido substancialmente. Isto fez com que o uso de realidade virtual se tenha disseminado para muitas áreas, desde a educação à saúde, e com que a criação de conteúdo para realidade virtual tenha cada vez mais procura. Essa criação de conteúdo, no entanto, tem tido dificuldades em acompanhar esses avanços e investigação em novas alternativas para o efeito precisa de ser feita.

Com este projecto, nós exploramos o espaço que existe na criação de melhores metodos para produzir conteúdo para realidade virtual, com a ajuda das capacidades de interação da realidade virtual e de sistemas disponíveis. Mais especificamente, o nosso foco é ajudar a criação de ambientes em realidade virtual, usando para o efeito repositórios de modelos 3D, controlos de voz e ferramentas procedimentais. Esperamos desenvolver uma ferramenta que torne o processo técnico de desenvolvimento de ambientes para realidade virtual o mais intuitivo possível, deixando assim apenas a parte criativa para o utilizador.

Posto de maneira simples, o nosso objectivo de investigação é descobrir se tal ferramenta pode ser feita, de modo a que, não só fizesse com que o processo de criação de protótipos de ambientes fosse mais fácil para utilizadores proficientes, mas também, fizesse com que fosse possível para utilizadores sem experiência em modelação 3D e até em realidade virtual, criarem os seus próprios ambientes sem a necessidade de aprendizagem extensiva.

A nossa ambição é que este projecto possa ser um degrau na direção da adopção da realidade virtual para mais domínios.

Começamos por estudar as fundações da modelação 3D e descobrir que metodologias melhor se encaixam com o nosso caso. Nós exploramos alternativas existentes de modo a melhor entender o vazio existente nas ofertas atuais, e falamos com modeladores 3D profissionais para perceber as suas necessidades e a viabilidade do projecto.

Depois disso, definimos as características chave do nosso protótipo, introduzimos com o planeamos testar e estabelecemos que ferramentas e tecnologias vamos usar para o desenvolver.

A seguir, analisamos o processo de desenvolvimento, explorando em grande detalhe todas as funcionalidades do nosso protótipo e como usá-las, e narramos todo o processo de criação de ambientes usando-o.

Finalmente, explicamos a nossa metodologia de testes, relatamos os nossos procedimentos e resumimos e interpretamos os resultados recolhidos, concluindo que o nosso protótipo responde adequadamente às questões de investigação estabelecidas.

"The whole point of human-centered design is to tame complexity,
to turn what would appear to be a complicated tool into one that fits the task,
that is understandable, usable, enjoyable."

Donald Norman

Contents

CONTENTS

List of Figures

List of Tables

Abbreviations

3D	Three-Dimensional
API	Application Programming Interface
DoF	Degrees of Freedom
FoV	Field of View
HCI	Human-Computer Interaction
HMD	Head-Mounted Display
HMI	Human-Machine Interface
SDK	Software Development Kit
VE	Virtual Environment
VR	Virtual Reality
WSR	Windows Speech Recognition

Chapter 1

Introduction

The concept of virtual reality is not a recent one. Prototypes of Head Mounted Displays (HMDs) have been made as far back as 1960 when Morton Heilig made his *Telesphere Mask*, a stereoscopic 3D vision device with stereo sound, and 1961 when two Philco Corporation employees made the *Headsight*, the first HMD with motion tracking. The concept itself goes even further back to the 1930s where it existed as creations of fiction literature. [Vir15]

Despite its long existence, it is in the early 2010s that VR suffers a great increase in attention with the launch of Oculus' famous crowdfunding campaign on August of 2012. This was one of the most important events in recent VR history, and triggered the explosion in VR interest we see today. Nowadays, driven both by the advances in technology and the appearance of low-cost alternatives that use one's mobile phone as a shortcut to achieving the functionality of an HMD, VR is at the highest point in popularity it has ever been, with plenty of VR devices being developed by many different companies. Current existing VR devices will be compared and discussed on section 2.1.1.

This growth in popularity can be observed in the related markets' figures, such as "Gaming in VR" which has grown steeply and with estimates putting it above the 14 billion USD mark by 2023 with an annual growth rate of more than 30% [Saw18]. This rise in popularity contributes to the appearance of new applications for the technology across many fields, where it brings about new innovative ways to tackle certain problems, as discussed later in section 2.1.2.

1.1 Context and Motivation

These new applications and the overall increase in the use of VR brings with it a necessity for more and better tailored VR content creation, which is, in general, a very time consuming and dispendious task. One of the biggest components of this is the creation of VR environments which itself presents numerous challenges, discussed in section 2.2.3.

Despite this, VR environment creation is still done using the same traditional 3D environment creation methods: screen-based 3D modelling software and game engines. Yet, although they have remained the *de facto* tools for the task for many years, many negative aspects about their use stand out:

- they tend to be very complex and have a steep learning curve, making it hard for newcomers or people without specialized skills to quickly adapt to them;
- they have unintuitive interfaces, with many shortcuts and mazes of sub-menus to memorize;
- and above all, specifically for VR, they do not provide a real sense of scale, making it so you need to step back and forth between VR and the 3D editor many times to double-check your work, which disrupts and slows down one's workflow.

All these factors support the need for research into the development of better tools for the development of VR environments, especially in the case of people without specialized skills in the area.

1.2 Research Questions

As an aid in guiding the research, design, and development of this work, the following research questions were elaborated indicating its main focus points:

1. How can VR and multimodal interaction approaches be used together to help the process of 3D environment creation?
2. Can we enable users without specific technical skills to easily and intuitively create VR environments?

Question 1 relates to our pursuit of new tools to assist in the process of creating 3D environments. As we will show along chapter 2, there is a lot left to explore along the lines of VR and multimodal interaction, for this very purpose.

1.3 Document Structure

Additionally to this introductory chapter, this document contains five more chapters:

Chapter 2 explores the state of the art around the fields of study of this project, and discusses related work and what aspects they leave open to be explored.

Chapter 3 will present our proposed solution and how we intend to test and develop it.

Chapter 4 will explain in detail the development process of the prototype and explore some of the problems faced, choices made and the features existent in the final product.

Chapter 5 will detail the testing process: what methods were used; what questions have been answered; what the overall results were.

Chapter 6 concludes the document by discussing the results obtained, future work, a brief summary and the expected impact of the project.

Chapter 2

State of the Art Review

To better understand the aim of this work, we first need to get an insight into the areas of study it pertains to, as well as recent work and relevant developments made on them.

Firstly, we will start by introducing the concept of virtual reality and exploring the current state of the technology, its applications and main research that has been done on the topic.

After that, we will proceed by introducing the topic of 3D modelling and delving into the methods for 3D models to be represented, the techniques used to model in 3D and finally focusing on the branch of 3D environment creation; we will explore all the challenges that come from doing so for VR and in what ways the process could be improved.

On a third note, we will highlight existing projects that try to tackle the same problem or that are similar to ours in some aspect.

And finally, we shall conclude by combining of all the arguments that support the development of our project.

2.1 Virtual Reality

Virtual Reality is a technology that allows the user to interface with a simulation through the use of stereoscopic 3D images and some form of motion tracking that compel the user to perceive a sense of depth and feel immersed in a digital environment. As best described in Encyclopaedia of Multimedia:

> "Virtual Reality is the technology that provides almost real and/or believable experiences in a synthetic or virtual way. To achieve this goal, virtual reality uses the entire spectrum of current multimedia technologies such as image, video, sound and text, as well as newer and upcoming media such as e-touch, e-taste, and e-smell. To define the characteristics of VR, Heim [Hei98] used the three "I"s, *immersion*, *interactivity* and *information intensity*" [Fur08]

2.1.1 VR Devices

As mentioned earlier, virtual reality has had a long evolution throughout the years, being that in the current years we have the largest amount of commercially available VR devices to choose from. Because of this, a quick comparison between them should be done and their main differences can be seen in table 2.1.

		Characteristics			
		Resolution	FoV	Refresh rate	Controllers
Tethered devices	Oculus Rift	1080x1200	95º	90 Hz	2 w/ 6 DoF
	HTC Vive	1080x1200	110º	90 Hz	2 w/ 6 DoF
	Playstation VR	960x1080	100º	60-120 Hz	2 w/ 6 DoF
	HTC Vive Pro	1440x1600	110º	90 Hz	2 w/ 6 DoF
	Oculus Rift S	1280x1440	95º	80 Hz	2 w/ 6DoF
	Valve Index	1440x1600	130º	80-144 Hz	2 w/ 6DoF and finger tracking
	Pimax 4K	1920x2160	110º	60 Hz	None
Untethered devices	Google Cardboard	Device dependant	100º	Device device	None
	Samsung Gear VR	1280x1440	96º	60 Hz	1 w/ 3DoF
	Oculus Go	1280x1440	110º	60-72 Hz	1 w/ 3DoF
	Oculus Quest	1440x1600	110º	72 Hz	2 w/ 6DoF
	HTC Vive Focus	1440x1600	110º	75 Hz	1 w/ 3DoF
	Google Daydream View	Device dependant	100º	Device dependant	1 w/ 3DoF

Table 2.1: Comparing the main characteristics of various VR devices

We have split the devices into two major categories: tethered and untethered devices, and for each, we have chosen some of the most popular solutions to compare. We will not be comparing between those two categories because tethered devices, as could be expected, tend to have the upper hand in terms of performance and, consequently, in terms of specifications. This is due to the fact that untethered devices mostly have inferior hardware driving them in order to make them portable. We will also focus more on the tethered side of the comparison as it is our use case.

Tethered devices

In the tethered devices, we have the three main competitors in the market right now: Sony, Oculus and HTC, being that Sony's Playstation VR is by far the most sold VR device currently, having surpassed the 4 million devices sold [Moo19], almost the same as the other two companies combined [sta18]. We have also included Pimax for their different approach to the development of their VR devices, opting for better resolution and FoV in exchange for refresh rates.

The Playstation VR has a unique position on the market, being the only VR device dedicated for a gaming console. Because of this, its comparatively low price, and the fact that Sony's Playstation 4 sold over 100 million units, it is easy to understand why Playstation VR became the market leader in sales despite being launched later than its competitors. It presents a resolution lower than other existing VR devices to make up for the fact that, being powered by a console, it will usually have inferior performance compared to a PC. It is also because of that, that it has a variable refresh rate, so more performance demanding games can be played at a lower refresh rate than lighter games.

The Oculus Rift and the HTC Vive are the two main competitors in the PC space. They have the same resolution and the same refresh rates, but the HTC Vive has a bigger FoV and a better tracking solution, being able to track over a 360º rotation while the Rift is limited to forward-facing. (An extra sensor can be bought for the Oculus to achieve 360º tracking.) Despite this, the fact that the Rift is cheaper, lighter and more comfortable to use and has better handheld controllers, have made it the sales leader in the PC market, as of 2019, followed by the Vive. Both companies have recently developed new solutions, the HTC Vive Pro and the Oculus Rift S. The Vive Pro is a clear upgrade from the Vive, with better per eye resolution, better tracking solution and built-in headphones. It is also more expensive than the Vive at roughly twice the price. The Oculus Rift S however, decided to go on a different direction, increasing the resolution only slightly and reducing refresh rates marginally, but developing a totally new inside-out tracking solution that foregoes the need for external lighthouses or external sensors. They have also chosen to keep the lower price of the original Rift.

The Valve Index is the latest competitor in the market. Developed by Valve, who had previous experience in developing the Vive in partnership with HTC, and who develops and maintains OpenVR (the currently most used VR SDK for its multi-platform support), the Index presents a list of new features that arguably make it the best VR device on the market. It has a resolution similar to the Vive Pro, a bigger field of view than its direct competitors, variable refresh rates up to 144Hz, and new and improved controllers with individual finger tracking.

Pimax has opted to pursue the highest resolution solutions being that their Pimax4K has a resolution of 1920x2160 per eye, and so, 4K resolution in total, thus the name. The use of higher resolution displays makes them have a crisper image when compared to other devices, but also makes them have lower refresh rates and be harder to drive with current hardware. Their solution does not include controllers, relying on other companies for that. They are also planning on releasing their new Pimax8K with a 4K display for each eye.

Untethered devices

On the untethered side, we have solutions like the Google Cardboard, Google Daydream View and Samsung Gear VR that rely on a mobile phone for their display, tracking solution and processor. Because of this, their resolution and refresh rate can vary greatly, although refresh rates tend to be on the 60Hz for most mobile devices. Samsung Gear VR is made specifically to work with Samsung Galaxy devices and, because of this, has a minimum known resolution of 1280x1440 per eye. All except the Google Cardboard have a controller but only the Daydream View one has gyroscopic tracking.

HTC and Oculus also have devices on this category, HTC Vive Focus, Oculus Go and Oculus Quest, all with similar specs and with internal displays, processor and tracking solution, thus not requiring any additional device. The Vive Focus and Oculus Quest have slightly higher resolutions and refresh rates.

2.1.2 VR Applications

Virtual reality, with its unique set of interaction possibilities and unparalleled visualization capabilities, makes many new ways to tackle different problems possible. It is because of this that this technology has branched into many other fields sparking new possibilities.

The study of these applications can help us better understand the use cases for our work. The use of VR technology has been observed in fields such as:

Education

Wang et al. [WWW+18] analyse the use of the technology in the field of construction engineering as an educational tool. More specifically they point out its application in architecture and design visualization, in training for construction health and safety, in training for operation and in structural analysis.

Mental Health

Jerdan et al. [JGvWK18] assesses the use of VR technology to aid in the cure of many mental health issues, such as anxiety, stress, depression, addiction, pain management and phobias. They exemplify with many cases such as the use of VR exposure therapy and Trier Social Stress Test done in VR. They also note the use of the technology as a simulation tool for learning.

As a conclusion they highlight the efficacy of VR in this area and observe that it is becoming increasingly used for health-related applications and is proving to be a valuable tool for public and mental health.

Architecture

Sepulveda et al. [SSFFR19] report on a project where they used virtual reality technology to help the city of Barcelona with an urban transformation project.

They employed the help of a virtual reality simulation of an existing urban space after the proposed changes, and presented it to the local population that wanted to participate in the project. They then gathered user feedback on the outlined changes and collected ideas proposed by the participants interactively in VR.

They found that VR greatly helped them by giving a level of realism to the proposed alterations and by providing a better visualization experience. They also discovered that “VR systems improve public motivation, implication, and satisfaction in urban decision-making processes.”

Manufacturing processes

Mujber et al. [MSH04] explain how VR technologies can be applied in several stages of manufacturing and what benefits it brings for each, based on existing projects. Stages like:

- Product design, where VR can be used as a visualization tool to evaluate the design of a product, compare alternatives, and even conduct ergonomic studies. It is also useful during prototyping to test and evaluate specific characteristics of the prototype;
- Operations management, where VR can be used to plan a new manufacturing system, to simulate the model and show its working and results to people and get feedback, and to train the employees earlier and more effectively;
- Manufacturing process, where VR can be used to aid the process of machining, assembly and inspecting through visualization and simulation of tasks.

It is clear that VR technologies have the potential to help and even revolutionize a wide range of fields through various interaction, simulation and visualization possibilities. The need for more VR content is, as a result, expected to increase as the demand for VR integration in more and more applications becomes apparent.

It is this increase in demand that we intend to help fulfil with our work.

2.1.3 VR Research

Virtual reality presents a new interaction paradigm that itself harbours a long list of research topics and problems to be tackled.

Notable research has been done on the topic of interaction, from the efficacy of a virtual laser pointer [MBN+02, Min95], multi ray projections from the fingers [MV18], to two-handed gestures [BBGV11] and the use of devices like electromyographic and inertial measurement armbands [HNV15]. Comparison between direct hand interaction or interaction through the means of a spatially-tracked controller has also been done [FRTT18], being the first less accurate but preferred by the user.

Despite the extensive research that exists in HCI and specifically in input technologies and techniques [HW07], the sub-field of interaction in VR has been less explored, due mainly to the lower levels of popularity up until recently. Despite that, efforts into establishing metrics and finding what makes a virtual environment effective have been made [SMR+03], as well as how to better classify and evaluate them [BGH02, BS10].

One topic of great relevance for VR research is that of the inherent motion sickness expectation that such a system carries [YZWZ19, Gol06a], how to better predict it [Gol06b] and how to better evaluate a virtual environment on that aspect [KPCC18].

Another topic of relevance for VR, particularly in our case, is that of voice interaction. Speech allied to gesture controls have multiple times been shown to aid in the 3D manipulation of objects [Hau89, HM93], and to be preferred by the user to exclusively gesture controls. The combination of gesture and speech controls have also been tested more deeply in VR applications, and been shown to be easy to adapt to and more efficient than traditional interaction methods [SZP+00]. Voice recognition can also be used simply as voice input when the user needs to input text in a VR context because it has been shown that it is an area that needs improvement, and alternative solutions like PizzaText [YFZ+18] have been developed.

Other research includes topics like the perception of presence in virtual reality, what factors contribute to it [SvdSKvdM01, MSPM10], and how that affects one's effectiveness and self-efficacy [SHCC18].

2.2 3D Modelling

3D modelling, as the name implies is the process of digitally modelling an object, using some dedicated software for the effect. It can be done in many different ways, both with different 3D representation paradigms and modelling techniques. Our interest is mainly in the modelling of 3D environments, but to better understand the processes of doing so, we first need to get an overview of 3D modelling as a whole.

2.2.1 3D Representation

To be able to do so, and although at the lowest level they end up being converted to similar representations for the computer to interpret, intermediary representations of the objects must exist. These representations can be done in different ways, being even possible to combine multiple of them. Each of these representations allows for different workflows on the user side, and have advantages and disadvantages which we will explore:

Polygons

The most common representation method, consists on having the model represented as a collection of vertices with 3D spatial coordinates, edges that each link two vertices, and faces that are defined by a closed-loop array of coplanar edges. Objects represented this way are usually know as polygonal meshes (Figure 2.1).

This method has the advantages of being easy to use and fast for computer to do calculations on, but it has the downside that the polygons are limited to being planar, thus being impossible to perfectly represent curved surfaces. To counteract this, many small polygons can be used to approximate a curved surface, which can end up greatly increasing the complexity of the model and consequently impact on its performance. Another way would be to interpolate light bounces between vertices [Um18b], which does not have as big of an impact, but can produce unrealistic or unintended results in some situations.

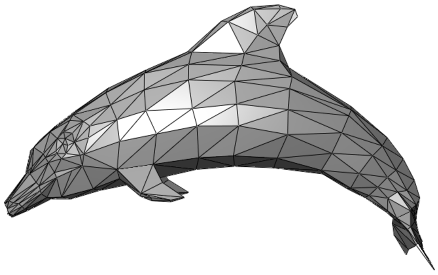
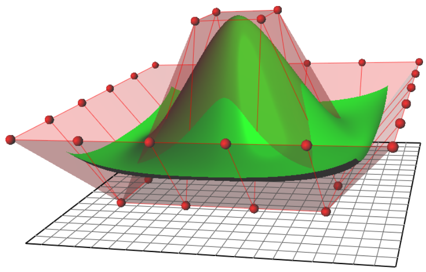


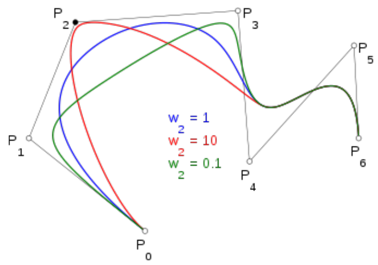
Figure 2.1: Example of a polygonal mesh

Curves

Other common representation methods exist that are based on curves, the most popular ones include NURBS (Non-Uniform Rational Basis Spline), splines and Bézier surfaces [Um18a, FH00, FVFH96a]. These have the ability to represent curved surfaces simply with a few weighted control points, being that the surface ends up as the interpolation or other mathematical function that is applied over these points to create a continuous surface (Figure 2.2a). Weights in the points or vectors associated with them can commonly be used to affect the way a surface follows them, changing its curvature (Figure 2.2b).



(a) NURBS surface



(b) Changing the weight of a point on a NURBS curve, Wojciech Muła

Figure 2.2: NURBS examples

Implicit Surfaces

Implicit surfaces are surfaces defined by mathematical equations (Figure 2.3). This method has the ability of representing very complex surfaces very efficiently, but at the same time, it has the downside of being limited to things that can be represented as a mathematical formula, and although it should be possible to represent most objects using a Fourier series function, as shown by Mousa and Hussein [MH20], the resulting formula would, in most cases, be extremely complex and computationally heavy.

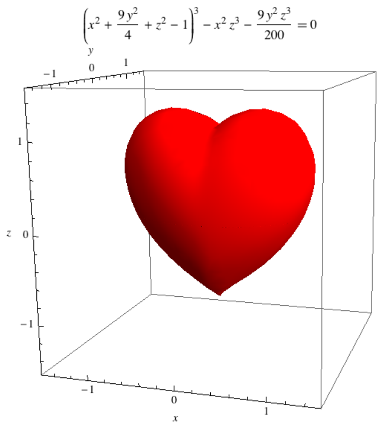


Figure 2.3: Example of an Implicit surface

CSG (Constructive Solid Geometry)

Constructive solid geometry, consists in using simple models, most frequently solid primitives, like cubes, spheres, cones, cylinders and the sort, and using Boolean operations to get to more complex models (Figure 2.5). It is mostly associated with Boolean and Parametric modelling because of this [FVFH96b].

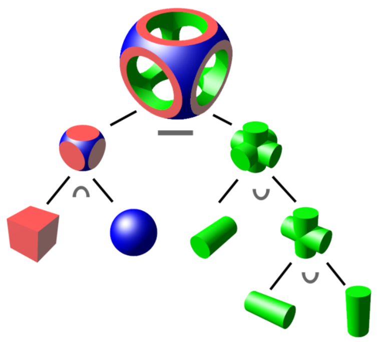


Figure 2.4: A CSG tree, Zottie

Voxels

Voxel representation consists in dividing the 3D space uniformly into a grid, similarly to pixels in a 2D image. In contrast to vertices and polygons, this representation usually does not have coordinates associated with the individual voxels, being instead inferred by their relative position to each other. This representation is usually associated with 3D sculpting software. The finer the grid, the more resolution an object represented by voxels will have.

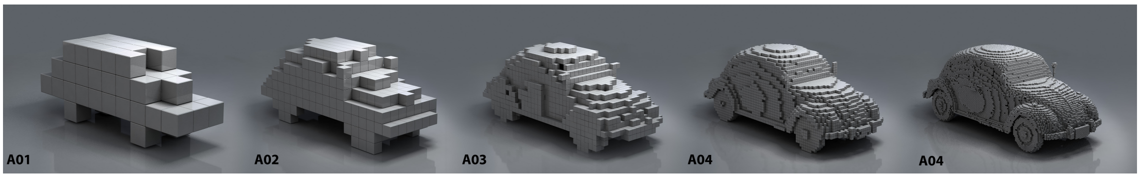
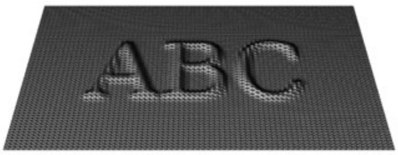


Figure 2.5: Voxels example, bilderzucht-blog

Displacement Surfaces

Displacement surfaces consist of storing geometric information in a 2D texture, usually as a height map or a vector map. Although this method usually produces results inferior to what could be done with polygonal modelling, it does so at a much lower cost for the CPU/GPU. This is why it tends to be heavily used for real-time rendering applications like video games. This method is also very common for sculpting applications, where displacing material on an object is abstracted as painting in a displacement texture.

(a) Displacement Map



(b) Displacement Surface

Figure 2.6: Displacement surface example, T-tus

2.2.2 3D Modelling Techniques

To better understand 3D modelling, we need to understand the main techniques used for 3D modelling. These can be very different from one another and be directed at different use cases and intended results. The most common techniques are:

Polygonal Modelling

Polygonal modelling is one of the most traditional ways of 3D modelling. By far the most used method, it can assume the form of various different techniques, like box modelling or contour modelling.

It works directly with vertices, edges and faces, which means that, because it is so low level, it is a very complex and time-consuming method. It also means that you are not confined by the tools you have and can model almost anything.

The modelling process usually starts with a solid primitive and through various operations like subdivision, extrusion, and simple translation of vertices, you get to the end result.

It is usually most commonly used for hard-surface modelling, but can be combined with techniques like subdivision or bevelling to achieve more organic results, which is known as cage modelling.

Digital Sculpting

Sculpting is one of the easiest modelling methods to learn. It is this way because it mimics the act of sculpting in real life; you add and remove material, engrave a texture, smooth surfaces, join models, basically anything you would be able to do to an actual piece of clay.

Because of this parallel with real life, this is the technique that tends to produce the most organic results/models. This method also takes the strain of worrying with edge flow and topology away from the artist, letting them focus better on their vision.

3D scanning and 3D reconstruction

In 3D scanning or reconstruction, the object is extracted algorithmically either from a 3D scan or from a set of 2D images (photogrammetry). Although not the most effective method, because its reliability depends on many factors to produce adequate results, it is by far the fastest and simplest method for replicating objects with considerable amounts of detail and/or complexity, depending only on processing power and time to produce its results.

Procedural Modelling

Procedural modelling refers to modelling where the design is done algorithmically instead of manually by the artist. User input comes on the form of changing parameters to an algorithm, setting up rules, or even through providing a mathematical formula. Notable examples of this are fractals, generative modelling and L-Systems.

Procedural manipulation can also be used as a tool to aid in the modelling process, for example in the randomization of parameters, deforming of models, or through the use of modifiers to perform tedious tasks.

Kitbashing

Kitbashing is the process of producing new models by combining existing models together. This is usually done to give detailed surfaces to simple objects or as the main process for creating 3D environments. Creating 3D environments usually consists in the placement of previously designed models around a scenery to reach an intended visual.

The term comes from the practice by the same name in which hobbyists or, more notably, movie prop designers would develop their unique models, mainly in science fiction, by combining construction kits from different models together.

Parametric Modelling

Parametric modelling is when a model is defined as the sum of the operations applied to it instead of its current state. Because of this, parameters about these operations can be changed at any time, altering the resulting model.

This modelling technique is very common in CAD software for its ability to produce very precise and easily adaptable models.

2.2.3 Environment Creation

Environment creation is a sub-branch of 3D modelling. It entails the design of bigger scenes composed of many 3D models instead of the design of a single 3D model. 3D Environment design usually entails worrying about the placement of objects, scale concerns, lighting, colours, and more.

Out of all the modelling techniques mentioned in section 2.2.2, some of them are better suited for the task of environment design than others, as methods like sculpting or box-modelling would be far too impractical, imprecise and time-consuming to be viable options for the task. Instead, methods like kit-bashing and procedural modelling are usually preferred.

3D model repositories and texture repositories are also often used to speed up the environment design process. They provide access to many fully polished 3D models and textures that eliminate the task of having to design them manually each time one wants to place props or texture something in an environment. Their use is especially common in some branches of environment modelling, like *archviz* (architecture visualization) where you need to develop a great number of renders very quickly and efficiently.

2.2.4 Consulting 3D modelling professionals

To gather more information about the main concerns when developing 3D environments for VR, we gathered feedback from various artists with some experience on the subject.

We contacted a team of 3D modellers, and through a spokesperson, we were able to relay our questions to the team and understand their thoughts on them.

When asked what the main problems they faced when designing virtual environments were, the overwhelmingly most common answer was the scale of objects. They all agreed that it was difficult to translate the sense of scale from VR to a 2D screen. Most of the times, even when working with real-life measurements, the final product does not correspond to the original vision. As one of the artists put it:

> "When designing environments for VR, we try to scale the objects and architecture very close to their real life-size. This seems like the most logical thing to do as we are looking to create realistic and proportionally accurate surroundings. However, when our team sees them in VR, there are always objects that feel out of place and need to be adjusted, either to occupy the assigned space better, or to establish a better affinity between themselves and their surroundings. [...] I find that realistic scaling doesn't always do much in creating expressive and absorbing environments in VR. I guess the brain interprets the sizes differently."

As we can interpret by this statement, alternating between testing an environment in VR and constructing it in a 2D screen-based modelling program can be a very frequent occurrence in order to achieve the intended results. This can prove to be an obstacle to maintaining a continuous workflow. Even using real measurements when creating models is not enough, as they do not always translate as intended into VR.

Another problem that was pointed out, was the difference in colour and lighting perception when viewing an environment in VR. Often enough colours and lights would have to be tweaked multiple times while alternating on and off of VR until they looked as intended.

The combination of theses issues really highlights the need for a VR environment creation tool that would eliminate all these problems, streamlining the workflow.

Questioning the artists about how helpful a tool that lets them design/prototype environments inside VR would be, the consensus was that it would easily improve their workflow. As remarked by one of them:

> "I can imagine how prototyping environments inside VR would be extremely helpful in predicting the end result of the environment. It would be like sketching on a sheet of paper, but instead you're sketching your ideas directly into the software. I expect this to result in more efficiency in creating the desired environment; a lot of time could probably be saved because there would be less unpleasant surprises at the time of assembly."

The case is then made for the problems that artists face when creating VR environments and how a tool that lets them do it in VR would be useful.

2.3 3D modelling in VR

There are a number of projects that explore 3D modelling in VR. To better understand how our proposed solution differs from them and what problems they fail to solve, we first need to learn about them. We tested and analysed a few of the most popular and feature-complete solutions available:

- Blocks [Goo] is an application developed by Google that lets you model 3D objects in VR. It is based on "paint" modelling and polygonal modelling techniques. It is very basic and thus simple to use. It does not allow the user to import or export any 3D models.

- Medium [Ocu] is an application developed by Oculus that allows you to create 3D models in VR. You do so with sculpting or kit-bashing modelling techniques. It has a library of objects to use for kitbashing and allows you to export the models to common 3D object formats.

- Ontlus [CPK+18] is a project developed by Chen et al. that lets you create and manipulate 3D objects in VR. Like our project, its motivation comes from the prevalence of 2D screen-based 3D modelling, but their approach focuses more in the creation of objects instead of environments and in collaboration, being that their solution allows multiple people to work simultaneously in the same scene. Their modelling is mainly based on sculpting and does not allow the importing or exporting of 3D objects.

- MasterpieceVR [Mas] is probably the most advanced VR 3D sculpting program in the market today. It offers more sculpting tools and possibilities than its competitors and also has kitbashing capabilities. It allows you to import images into the VR environment to use as references and finally allows you to export your models to common 3D formats. Like Ontlus, it has collaboration capabilities, with multiple users working in the same scene.

- Gravity Sketch [Gra] approaches modelling in VR differently. Instead of the usual sculpting modelling, Gravity Sketch is prominently based on curve modelling. It allows you to use Bézier and spline surfaces as well as bevelled curves to produce smooth modelling results. Like MasterpieceVR, it allows the importing of images to use as references, and allows the exporting of models into common 3D formats.

- Unbound [Unb], like Masterpiece VR, is mainly focused in sculpting modelling of 3D objects, although much more limited in terms of tools and capabilities, since it is still in early access. It has collaborative features and export capabilities.

- 3D palette [BBMP97] is a project by Billinghurst et al. Developed in 1997, that lets the user model 3D environments inside VR. It uses multimodal interaction approaches like the use of a digitizing tablet as an input. It is based on kitbash and polygonal modelling and shows that even in 1997, the search for better modelling paradigms, that were not based on 2D screens, existed.

- Maquette [Mic], developed by Microsoft, is the closest existing project, in terms of objectives, to our proposed solution. Like our solution, it focuses in the creation of 3D environments and is mainly based on kitbashing modelling, although it also has some sculpting capabilities. It has a local library of 3D objects and allows the user to export their creations. For text input it relies on a virtual QWERTY keyboard which is not the most practical solution. It is overall one of the most complete tools from this comparison.

2.4 Summary

With what we have learned, we planned our tool according to the following goals:

- Focus on the creation of environments, instead of individual objects, having no tools to create, edit or modify object meshes at a low level;
- Mainly based on kitbash modelling, with the ability to import 3D objects both from local directories and 3D repositories;
- Voice input and control capabilities;
- Procedural modelling capabilities;
- Ability to export the final result.

Comparing it to current offerings, as seen on Table 2.2, we can see that the feature set we want to offer with our solution is unique.

As was evidenced in this chapter, the development of environments for virtual reality is something that has been getting more and more demand recently. This is a process that has many problems, as explained by various artists, that need to be tackled if we intend to streamline the process to the point that even a regular user could do it without much effort. We have shown that there are many techniques and technologies that could help us in this process and that a tool that could harness them would be very welcome, even by experienced artists.

In conclusion, we can see that there are numerous, and strongly backed motivations that support the realization of this project.

	Blocks	Medium	Ontlus	Masterpiece VR	Gravity Sketch	Unbound	3D Palette	Maquette	Proposed Solution **VR Designer**
Low-level creation of objects	✓	✓	✓	✓	✓	✓	✓	✓	X
Creation of environments	X	X	X	X	X	X	✓	✓	✓
Importing of models	X	✓	X	X	X	X	X	✓	✓
Access to 3D repositories	X	X	X	X	X	X	X	X	✓
Procedural tools	X	X	X	X	X	X	X	X	✓
Collaboration capabilities	X	X	✓	✓	X	✓	X	X	X
Voice input	X	X	X	X	X	X	X	X	✓
Export creations	X	✓	X	✓	✓	✓	X	X	✓

Table 2.2: Comparison between existing VR 3D modellers

Chapter 3

Proposed Solution – VR Designer

Our proposed solution, aptly named **VR Designer**, is one that harnesses the interaction and visualization capabilities of VR technology to allow the user to interact with a virtual environment where they can, in an intuitive way, manipulate 3D objects with the intent of designing a 3D virtual environment.

In this chapter, we will define its intended features, structure its architecture and present which tools and technologies will be used.

3.1 Main Goals

Following what we learned from the state of the art research, we devised the following goals for our solution:

- Access to 3D model repositories to browse available libraries of 3D objects ready to place in the virtual world at will. To complement this, the ability to import 3D primitives or simple 3D objects.

- The use of voice recognition both as a way to input text for when it is necessary (e.g. search terms), and as a way to invoke commands, making the interaction with the virtual world as immersive as we are able to;

- The ability to use procedural tools to aid in the process of designing the environment. They can be both useful in the randomization of parameters about objects, in the deformation of models or to help in performing otherwise tedious tasks like the placement of objects in bulk;

- The ability to import objects from local directories, making it viable to integrate this project along with other programs in a complete VR environment modelling workflow;

- The ability to save and load the state of the scene as well as export it to a standardized format in order to use it with other programs.

3.2 Requirements

To help in organizing the development of our solution, we converted the goals of our prototype into a list of requirements:

1. The user can search for objects online using voice controls;
2. The user can import objects from online repositories;
3. The user can import objects from local directories;
4. The user can interact with and manipulate the imported objects;
5. The user can change properties like position, rotation, scale and type of an object;
6. The user can move around in the world;
7. The user can apply modifiers to the imported objects;
8. The user can group multiple objects into one;
9. The user can export the environment created;
10. The user can playtest the environment created.

3.3 Solution Architecture

The system architecture of our solution will resemble what can be seen on Figure 3.1. The user will interact with VR Designer through the VR headset, VR controllers and microphone, which we will explain in section 3.5. VR Designer uses the Unity 3D game engine and Steam VR and Windows Speech Recognition (WSR) as its primary libraries, and has access to online 3D object repositories to search for and download 3D models.

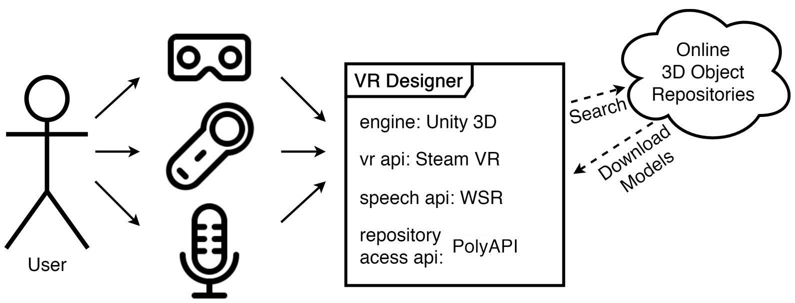


Figure 3.1: The planned system for our solution

The implementation of our solution will be organized into several modules according to the diagram in Figure 3.2.

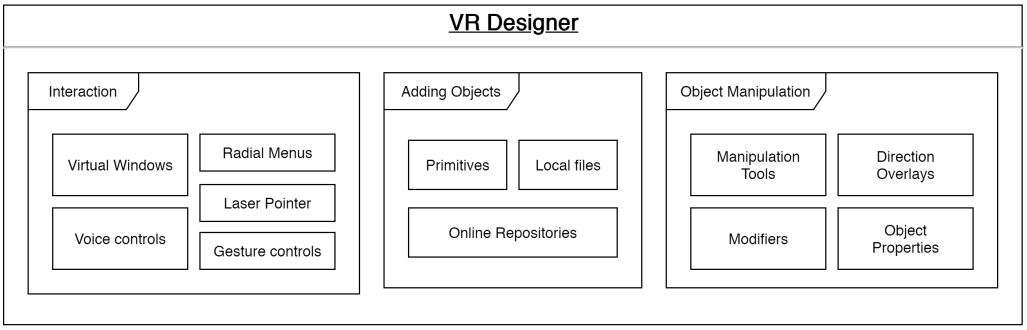


Figure 3.2: The building blocks of our solution

In this figure, we can see how each module relates to a different part of what makes our prototype work. Only the main modules of our prototype are represented, being that some minor modules will only be introduced in chapter 4. In that chapter, we will explore in detail how each module functions and their role in the prototype.

3.4 Tools and Technology

For the development of our solution, several different tools and technologies will be used.

3.4.1 Unity

Unity (also known as Unity3D) [Uni] is a game engine developed by Unity Technologies. It offers tools for the development of two-dimensional, three-dimensional, augmented reality and virtual reality games and experiences along with a very extensive scripting API in C# .

It was the chosen game engine for the development of our project for its integration with the SteamVR SDK (section 3.4.2), its very comprehensive online support and because we already had some previous experience with it.

3.4.2 SteamVR

SteamVR [Ste] is a VR platform developed by Valve that utilizes the Open VR API for cross-hardware support, allowing the development of vendor-agnostic VR applications. On top of that, it provides many building blocks for the development of VR applications. It is for these reasons that we chose to use SteamVR.

3.4.3 Windows speech recognition

Windows Speech Recognition (WSR) is a speech recognition API developed by Microsoft that features in every consumer-oriented build of Windows since Windows Vista. If features several libraries with different purposes each:

- Dictation recognizer. Listens to speech and attempts to determine what words or sentences were spoken;
- Keyword recognizer. Listens to speech and attempts to match what was said with to a list of registered keywords;
- Grammar recognizer. Extends on the keyword recognizer by using *XML* to define rules about the words or sentences to be recognized.

For our purpose, the *Dictation Recognizer* is the library we will be using, giving more liberty to what the user can say. This is crucial because of the uses we give to speech controls in this prototype.

3.4.4 Poly API

Poly API [Pol] is an API made by Google that allows us to interface with their free 3D models repository of the same name. Using this API, we can search Poly for specific models, retrieve thumbnails, and download models and view their information.

Most of the models in Poly and published under CC BY 🅭🅯 and are thus available for free to copy, modify and remix. For our prototype, we will be filtering the results by the objects that are, in fact, published under these terms.

3.5 Interface devices

When interacting with VR Designer, the user will have the following interface devices available to them:

- VR headset. The VR headset allows the user to look around and be immersed in the virtual world. We will develop our prototype for use with the HTC Vive HMD, seen in Figure 3.3, although it should be compatible with most existing HMDs;

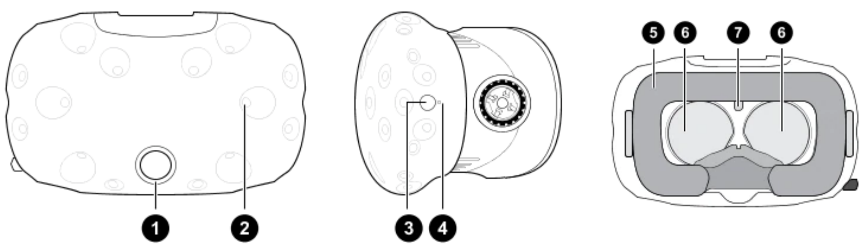


Figure 3.3: The HTC Vive HMD diagram

The basic elements of the HTC Vive HMD are:

1. A camera: Provides a live feed of what is in front of the player;

2. Tracking sensor: The combination of the tracking sensors on the HMD allow it to track both its position and orientation in 3D space;;
3. System button: A button that creates press and release events. Useful when there are no controllers available;
4. Status light: Indicates when the HMD is on or if there is an error preventing the HMD from being used;
5. Face cushion: Makes for a more comfortable fit to the face and prevents light bleed;
6. Lenses: Allow the user to see the HMD's screen from very close without straining the eyes;
7. Proximity sensor: Turns the screen on and off when the user puts on or removes the HMD.

- VR controllers. The controllers are the main source of user interaction in our system. It is using them that the user is able to perform most of the available operations in VR Designer. For this, the user can interact with the elements of the controller as seen in Figure 3.4. We optimized our prototype for use with the Vive wand controllers, however, it should be compatible with a great number of other VR controllers, needing only to remap the keys in Steam VR.

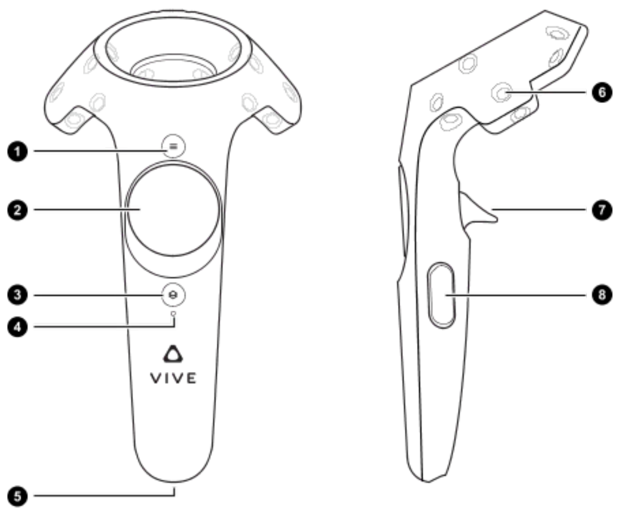


Figure 3.4: HTC Vive wand controllers diagram

The elements of the controller are:

1. Menu button: A button that creates press and release events;
2. Trackpad: A touch surface that the user can use to input XY positions and press and release events;

3. System button: A button that creates press and release events;
4. Status light: Indicates the on/off state of the controller as well as low-battery;
5. Charging port: Port used to charge the controller;
6. Tracking sensor: The combination of the tracking sensors on the controller allow it to track both its position and orientation in 3D space;
7. Trigger: A linear actuation input with additional press and release events when fully pressed;
8. Grip button: A button that is triggered by holding the controller with more intensity press and release events

- Microphone. The HTC Vive HMD comes equipped with a microphone in the underside, close to the nose rest (Figure 3.5), and because of that, sits near the user's mouth. We will use this microphone for all voice interactions in the VR Designer.

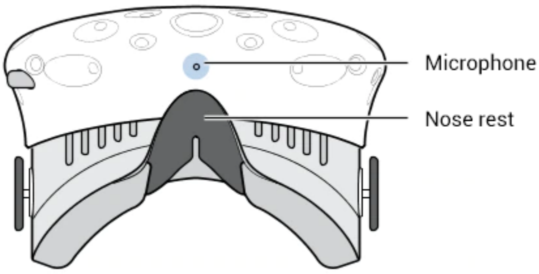


Figure 3.5: Diagram of the underside of the HTC Vive HMD

3.6 Summary

In this chapter, we introduce our solution by presenting its main goals, requirements and architecture, as well as tools used and technology employed. We also learn about the interface devices we intend to use.

In the next chapter, we will explore in detail each of the modules that make up our solution, and learn about its features and how to use them.

Chapter 4

Prototype Development

In this chapter, we will explore in detail the development process of the prototype, the different modules that make it up and the reasoning behind the decisions that were made along the way.

We will start by narrating an expected use case for our prototype to help better understand each modules' utilization. The analysis of the modules will be organised according to their function in the prototype, being split into three main groups: Interaction controls, Objects and World.

4.1 Expected Workflow

When developing our prototype, we had a specific workflow in mind that helped guide us in the development process. We will narrate the process a user would go through, using our prototype, to achieve their desired result, and indicate the respective sections where we explore in more detail the specific functionalities.

Let us say, for example, that a user would like to create a simple classroom scene, just 4 chairs in a line, the teacher's desk and chair, and a chalkboard on a wall, as seen in Figure 4.1.

The first step would be to import the necessary objects. To do this, the user would either be able to **import objects from local directories**, build them by **combining primitives**, or use speech controls to **search online for free 3D models to download** (section 4.3.2). For this, our prototype would need a **menu system** to choose between these and other actions (section 4.2.1), a way to **group simple objects** into more complex ones (section 4.3.6), a way to **input voice** and get it recognized to be used as a search query (section 4.2.3), and a way to **visualize and choose from the search results** (section 4.2.2).

The next step is to place the objects in the correct position. For this, we want the user to be able to manipulate the objects intuitively using **gesture controls to move and rotate** the objects in 3D space (section 4.3.4). The user would also have **tools available to scale, rotate and move the object** using the buttons and trackpad of the controllers (section 4.3.5), as well as a way to **choose the axis in which these transformations should be applied** (section 4.2.4).

To place the chairs for the students, the user should not need to repeat the same actions four times. Instead, the user should have available to them, a set of procedural tools that allow them to

Figure 4.1: An example of a scene.

expedite the process. Such tools would be the **modifiers** (section 4.3.7). As an example, an *Array modifier* would allow the user to replicate the same object a set number of times, with a chosen interval between them (section 4.3.7.2).

To make the scene feel more dynamic, the user would have the ability to place objects with physics interactions, like gravity and collisions. For this, two **types of objects** would exist: Static and Dynamic (section 4.3.1).

Our scene is now complete and we can spend some time adding details to bring it to life. To view how the scene mood changes under different lighting conditions, the user would also have the ability to **change the time of day** (section 4.4.2). All that is left, is to **play-test** it (section 4.3.3.3) before **exporting** it to use as desired (section 4.4.3).

4.2 Interaction Controls

In this section, we will explore the modules of the prototype that grant the user interaction capabilities. These modules are not specific to any distinct functionality, being instead a bridge for the user to access and interact with various functionalities. Some details of the modules in this section will be further analysed when examining the specific feature they are used in.

4.2.1 Radial Menus

Since the Vive wand controllers are very limited in the number of different buttons they have, it became apparent very quickly that other alternatives were needed to allow the user to access all the different features of the program. One such alternative were the radial menus.

The radial menus are a way to get the user to navigate menus and submenus in an as intuitive way as possible. The same way touchscreens came to change the interaction paradigm in mobile devices, forgoing the need for physical buttons, VR interaction and visualisation capabilities make it possible to design interaction systems with a minimal need for physical buttons.

To achieve this, the radial menu is made up of several 3D icons that appear around the hand of the user when they press the menu button on the controller, as can be seen in Figure 4.2. After this, to select an option, simply point the controller to the desired icon and press the select button. After selecting one of the options, the menu will close, and in the cases where the selected option corresponds to a sub-menu, another radial menu will appear with the new options showing. To go back up in the menu hierarchy, simply press the menu button again. If pressed while in the main menu, the menu button will close the radial menus.

To make it easier for the user to understand their actions, the radial menu is equipped with a feedback system where the icons will move when being pointed to, and text will show the name of the option the icon corresponds to. This interaction can be seen in Figure 4.2.

Figure 4.2: The main radial menu feedback system

Early tests proved the system to be effective and made it so that it became the main way of navigating the program's features. The menu options were hierarchically divided into 3 submenus:

- Interaction. In this submenu we will find options related to the way we want to interact with the world. The meaning of each option will be explained in more detail in section 4.3.3. The options in this submenu are:
  - Select Mode. Change interaction mode to selection mode. 4.3.3.2
  - Edit Mode. Change the interaction mode to edit mode. 4.3.3.1
  - Play Mode. Change the interaction mode to play mode. 4.3.3.3
- Add Object. In this submenu we will find options for adding objects to the environment. The meaning of each option will be explained in detail in section 4.3.2. The options in this submenu are:
  - Local Models. Import models from a local directory. 4.3.2.2
  - Search Online. Search online for 3D models to import. 4.3.2.3
  - Primitives. Import basic 3D models. 4.3.2.1
- Options Menu. In this submenu we will find options for actions that are needed sporadically, and thus are considered secondary compared to the others. They have been grouped here to reduce the clutter in the main menu. The options in this submenu are:
  - Change time of day. Toggle between different light settings and skyboxes 4.4.2.
  - Save world. Export the world created to an OBJ file. 4.4.3.

4.2.2 Windows

While the radial menus are useful in navigating and choosing options that would fit well in a menu context, some actions require more control and more information to be given to the user. To tackle this we developed virtual windows that behave very similar to 2D application windows, but in 3D space.

To interact with a window, simply activate the laser pointer, as will be explained in section 4.3.4.2, use it to point to the window and press the grab button to click on elements of the window. Different elements will respond differently to being hovered and pressed to give the player some feedback about their actions.

Three different types of windows are used in *VR Designer* (Figure 4.3) and each will be further explained in the section of their respective functionality:

- Search online window, seen in Figure 4.3 left (section 4.3.2.3);
- Properties window, seen in Figure 4.3 centre (section 4.3.2.3);
- Modifiers window, seen in Figure 4.3 right (section 4.3.7.1).

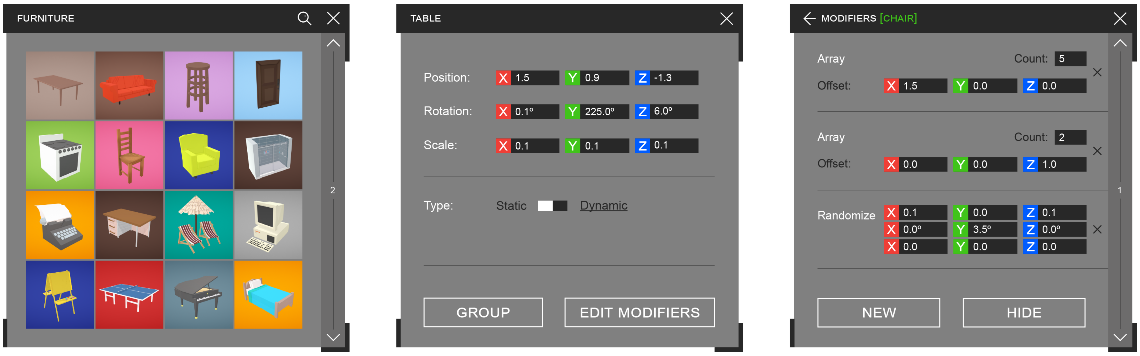


Figure 4.3: The three windows used in our prototype

4.2.3 Speech Recognition

Seeing as, while in VR, the user does not have access to a keyboard, text input is another interaction problem for which we have to find a solution. Possible solutions would be to develop a virtual keyboard or an alternative along the lines of PizzaText [YFZ+18], but we decided to go with speech-based text input.

We went with WSR because it integrates directly with Unity and because, being that it is locally processed, it achieves lower and more stable latencies than other solutions.

WSR is a stream-based speech-to-text engine, which means it constantly outputs hypothesis of what it thinks the user is saying. It also listens for pauses and inflexion in speech patterns to understand when the user finished talking and so output the final result.

In our system, we wait for this final result to be received in order to grab the text and consider it usable. However, sometimes the speech recognition fails to identify the perceived text as final, in which case the user would need to repeat itself. To prevent this, when it is the case that the speech recognition engine finishes without outputting a final result, we grab the last hypothesis it recognized and use it instead. This makes it so that the use of speech feels much less cumbersome.

4.2.4 Direction Overlays

This is a module with the objective of allowing the user to specify the axis along which they want a transformation to be executed. It gives the user, not only the ability to quickly change between directions, but also a clear visualisation of what their choice is and how it will affect the object.

Two different direction overlays exist, although they work very similarly. In fact the only difference between the two is that one has 4 options: *All*, *X*, *Y* and *Z*, while the second has only 3 options: *X*, *Y* and *Z*. This is because for some tools, the option *All* would not produce any desired results. Because of this, the scale tool uses the 4-option overlay, seen in Figure 4.4 (left), while the direction tool uses the 3-option overlay, seen in Figure 4.4 (right), since rotating in all axis at the same time would be impractical.

To enable the overlay, simply hold your finger without moving it on the controller trackpad while you have a tool selected. After around one second of holding, the overlay associated with

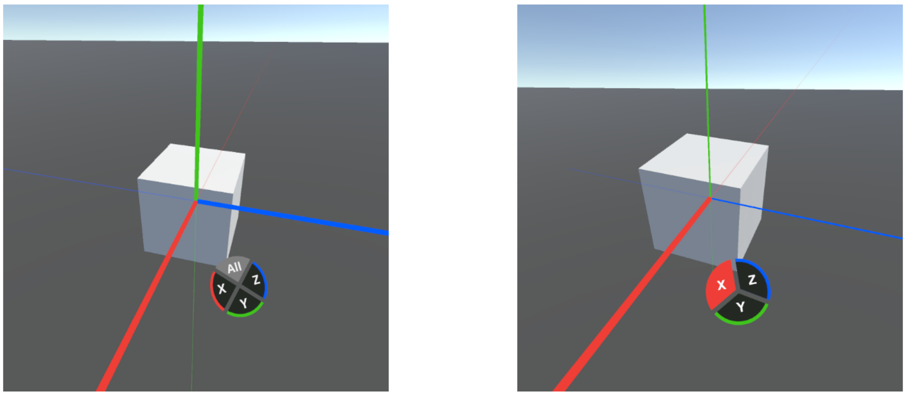


Figure 4.4: Direction overlays. Four options on the left, three options on the right

the tool selected will appear showing the available options. To select one of the options, simply touch the trackpad in the place of the desired option and either press it or release your finger and wait around 1 second for the overlay to hide itself again.

When hovering any of the options, the overlay will respond visually by highlighting which one is selected. At the same time, a visualisation of the selected option is shown over the target object in the form of a three-dimensional axis colour-coded to match the overlay options, as seen in Figure 4.5.

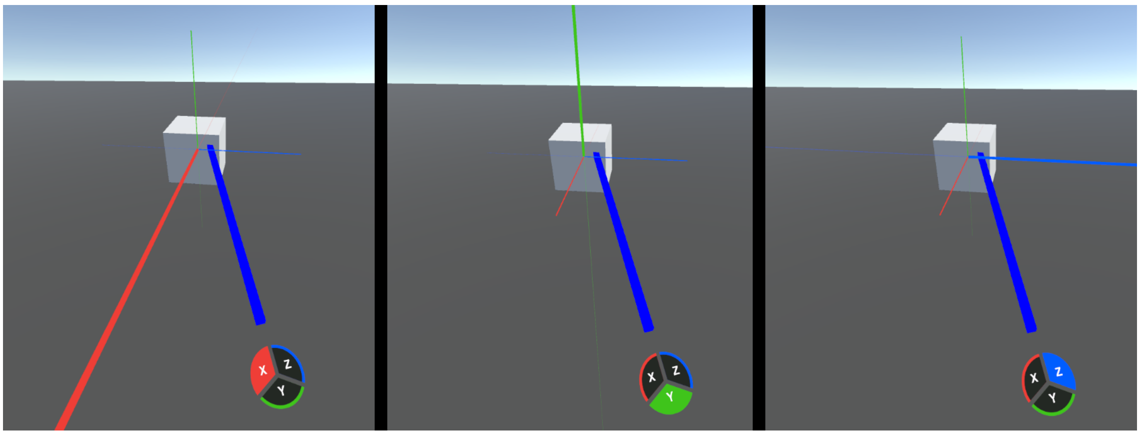


Figure 4.5: Visualisation of the chosen axis

4.3 Objects

In this section, we will analyse the modules and functionalities directly related to placing objects, manipulating them, and changing their properties.

4.3.1 Object Types

To enable the creation of more dynamic scenes, two different types of objects were created, each with different physical properties and interaction capabilities. The type of an object can be changed any time through the properties window, and this process will be explained in section 4.3.3.2. The two object types are *Static Object* and *Dynamic Object*, and their differences are listed in detail below.

4.3.1.1 Static Objects

These are the default type of objects. Whenever a new object is added to the scene, it always starts by being a static object. This is the ideal object type for furniture, buildings or in general motionless objects. These objects are characterised by the following key properties:

- They are motionless. Whenever they are placed in the world, they can not move unless the user moves them intentionally or it changes to being a dynamic object. This means that objects of this type are even allowed to float.
- They do not respond to collisions with other static objects. Because of this, static objects are able to be placed intersecting each other. This can be a powerful tool to allow the creation of more complex objects out of the junction of several simple ones. They still collide with dynamic objects normally, making it possible to place a dynamic object on top of a static object, for example.
- During play mode, they are not able to be grabbed, working as if they were unchangeable parts of the world.
- They intercept teleportation arcs while in play mode. A user cannot teleport through a static object. This is useful to keep players inside buildings or to prevent them from teleporting inside objects.

4.3.1.2 Dynamic

This is the ideal type for objects to exist as if they were placed on top of the world instead of being an intrinsic part of it. In other words, objects of this type function as if they were props. Dynamic objects are characterised by the following set of key properties:

- They are kinematic. They respond to gravity and other forces applied to them, have intrinsic velocities and can even be thrown.
- They collide with every other object in the world and even the player. Because of this, they can be nudged by being pushed, unlike static objects.
- They can be grabbed while in play mode. This makes environments feel much more interactive and alive.

- They do not intercept teleportation arcs. Dynamic objects function as if they were props placed in the world and because of this, they do not interfere with the player's actions and movement.

4.3.2 Importing Objects

Despite all the features we already talked about, this prototype would not have much purpose without the ability to add objects to the world. This module focuses on doing just that, being one of the most important modules in the prototype.

There are three different methods of importing objects into the environment. Each of them can be used to achieve slightly different results and this difference is explained in more detail below.

4.3.2.1 Import Primitive Objects

This is the simpler of the three import methods. With this feature, the user is able to place simple geometric shapes in the world by selecting from a list of 5 options. Despite being the simplest, these objects can also be the most versatile, since the player can join several of them to create more complex objects. The 5 options for primitives are:

- Cube. Translating and scaling them, 5 cubes together could constitute a table (table top and four legs);
- Sphere. Could be a simple beach ball;
- Cylinder. Together with a sphere, one could make a lamp post;
- Plane. Useful to make walls, divisions and surfaces;
- Pyramid. On top of a cube could be a house.

These options appear in the form of a radial menu, as seen in Figure 4.6 (left), making it very easy and intuitive for the user to select the desired one.

After selecting the desired primitive, it will appear in front of the player's field of view. At this point, the player can interact with the object as described in previous sections and manipulate it at will. This can be seen in figure 4.6 (right).

4.3.2.2 Import Local Objects

Another method of adding objects to the world is to import them locally. This method allows the user to import models from local folders on their computer. Currently, only the *OBJ* file extension is supported and the file must be placed in a specific folder. In the future, a module could be developed that would allow the user to navigate through their directories and import a more extensive list of file types, but we considered it to be unnecessary for this prototype.

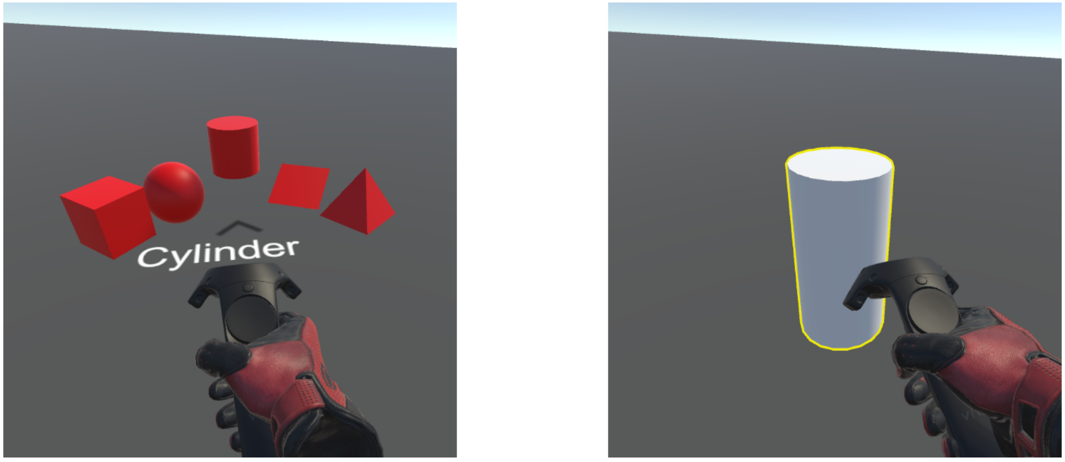


Figure 4.6: Primitives radial menu. And object after selection

4.3.2.3 Search Online for Objects to Import

The last, and in fact, the most important method to add objects to the world, is to search free online repositories for them. This is one of core functionalities of our prototype and because of that, we spent a lot of time trying to get it right.

To search for an object on an online repository, simple select the appropriate option in the *Add Object* radial submenu (section 4.2.1). After this, the speech recognition system will activate and start listening for the player's voice. A heads-up element will appear in front of the user, prompting them to say what type of object they are looking for. The user simply needs to speak out loud what they want to be searched, and the speech recognition system will translate it into text. This interaction can be seen in Figure 4.7.

Figure 4.7: Using speech to search for objects online

After this, a window will appear in front of the user where they will be able to see the results of their search. The window will show thumbnails for 16 results at a time, with the ability to go to the next and the previous page, and to change the search query. This window can be seen in Figure 4.8 (left), and will be explained in more detail in the end of this section.

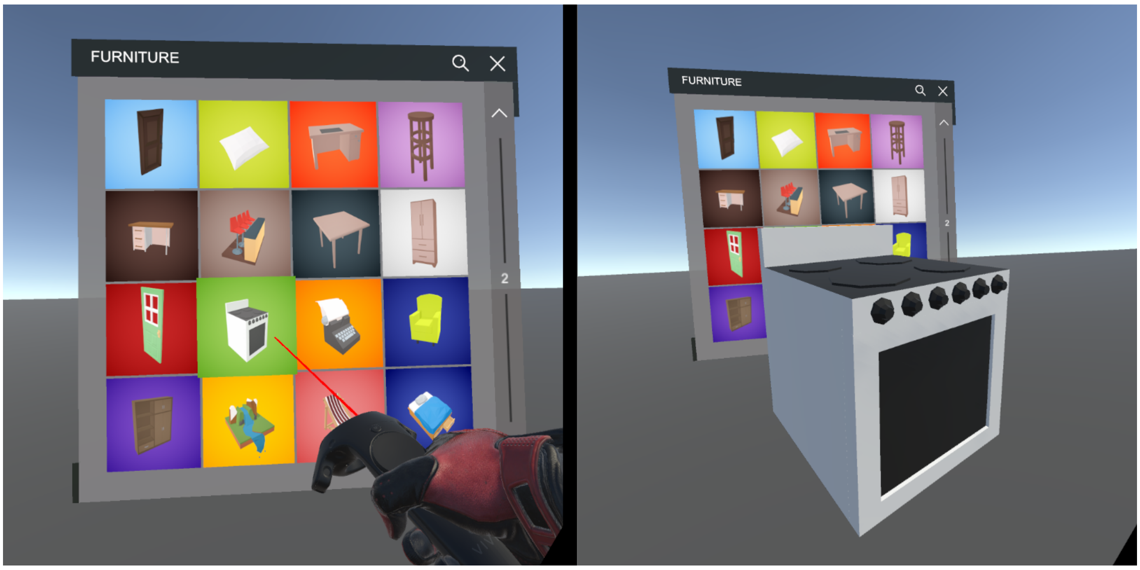


Figure 4.8: Using the search online window

At this point the user simply needs to choose which of the objects they want to import and select it. After selecting it, it will be imported into the world and appear in the field of view of the user, as can be seen in Figure 4.8 (right). Because different objects can have a very wide range of sizes, special care was put into making sure that the imported object was never bigger or smaller than specific set sizes.

For the online repository, we chose Poly by Google for a couple of reasons:

- It has a very well documented and easy to use API, which made the job of searching and importing objects much easier. If not for this, the development of a web scraper would probably have been needed, in order to extract information from other 3D repository websites;
- The results are more consistent. Compared to other repositories, the results from search queries present a higher number of usable results and with a more cohesive style. This can be very important if we do not want objects to feel out of place in the environment we are creating;
- Poly features a curation system maintained by Google, which allows us to filter out the non-curated results and obtain an even higher proportion of usable objects;
- Creative commons license. The *Poly API* allows us to filter results by their distribution licensing, and because of this, the user can feel safe when using the models available.

Search Online Window This is the window used to search for objects online. It opens after the user inputs a speech query after pressing the *Search Online* option in the *Add Objects* submenu. Doing this will make the window appear in front of the user, in the direction they are facing. It features the following elements, each with a specific function:

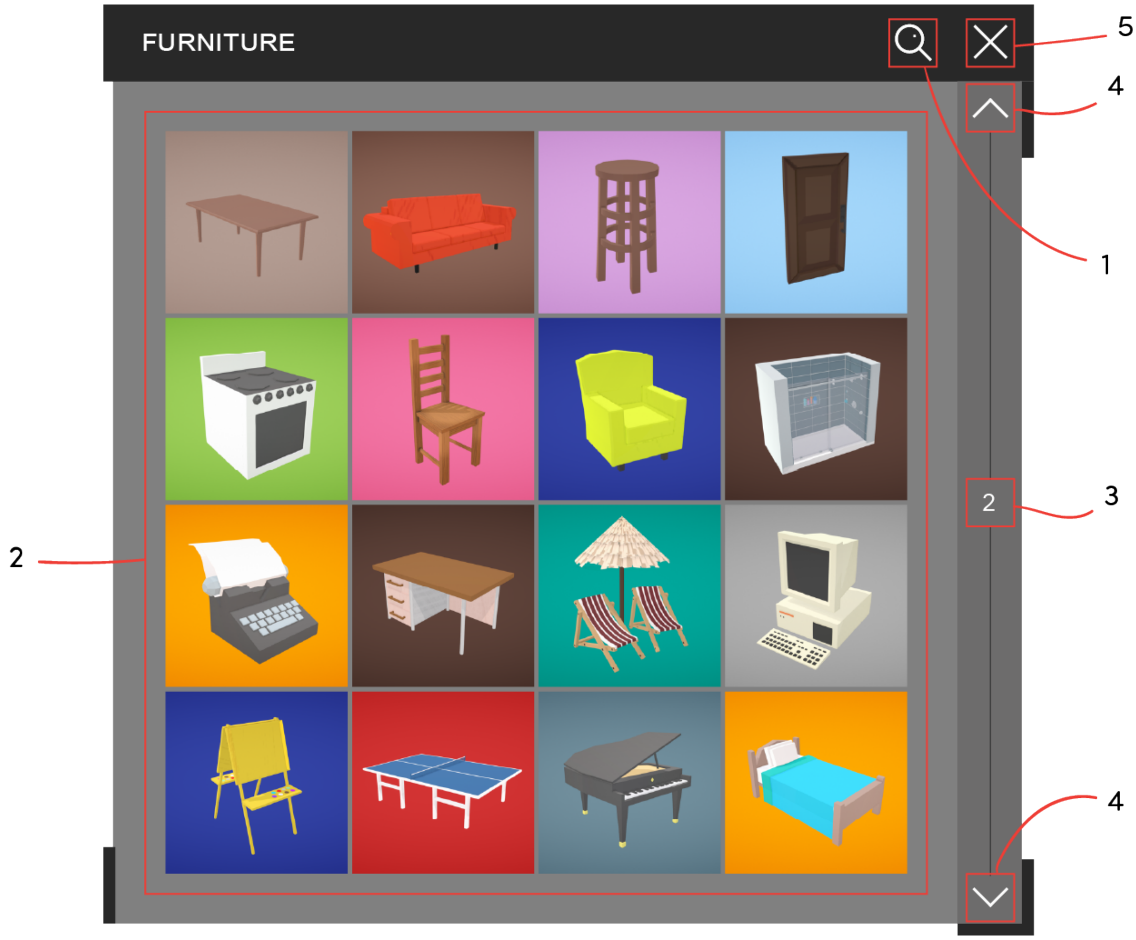


Figure 4.9: The search online window

1. Search button. This triggers the speech recognition and allows the user to change the current search query;

2. A 4 by 4 grid of object thumbnails. These are the results of the search query. Clicking one of them will import it into the world, making it appear in the field of view of the player, as explained previously in this section;

3. Page arrows. These arrows can be used to go to the next and previous pages of the search results. When a previous or next page are not available, the corresponding arrows will not show;

4. Page indicator. This number indicates the current page, or 0 if no results are available for the current search query;

5. Close button. Clicking this button will close the window.

4.3.3 Interaction modes

The way the user interacts with the environment around them is divided into three different modes. Although some interactions are similar between different modes, this separation makes the process feel more organised and thus, easier to learn and understand. Each of the three modes is explained in depth below.

4.3.3.1 Edit

This is the main mode of operation for the program. In this mode the user can move and orient objects by simply picking them up, using one of two methods that will be explained in section 4.3.4, and placing them in the desired position.

Additional controls for scaling and fine-tuning an objects' rotation exist in this mode, and are explained in section 4.3.5. This is the mode that should feel more natural to use and is the one that allows the user to be the most creative. This mode can better be explained as being the one used to make a rough mock-up of the environment.

4.3.3.2 Select Mode

In this mode the user is no longer able to move the objects using gesture controls. This mode's main purpose is to allow the user to select an individual object and modify each property independently. This is useful when an object is already nearly perfect but a slight alteration is needed, for example, rotating it one degree in the Z axis.

While in this mode, the user has access to tools that allow them to do that and much more, through the use of what we call a *Properties Window*. This mode can better be explained as being the one used to do controlled modifications and the final polishing to the environment.

Properties Window In this window, the user can see the current properties of the selected object. This window opens when the user is in *Select Mode* (section 4.3.3.2) and uses the laser to select an object. When doing so, this window will appear above the object and facing the user. This window feature the following elements:

1. Object name. This shows the user the name of the object that is selected. This can be either the name of the primitive, the file name of an object imported from a local file, or, in the case of it being imported from a repository, the name it had associated with it. Many times, this name will closely resemble the search query it resulted from;

2. Spatial indicators. These tell the user the current position, rotation and scale of the object selected. By clicking them, the user is able to use the controller trackpad as a scroll wheel to increase and decrease each individual component, fine-tuning the spatial properties of the object;

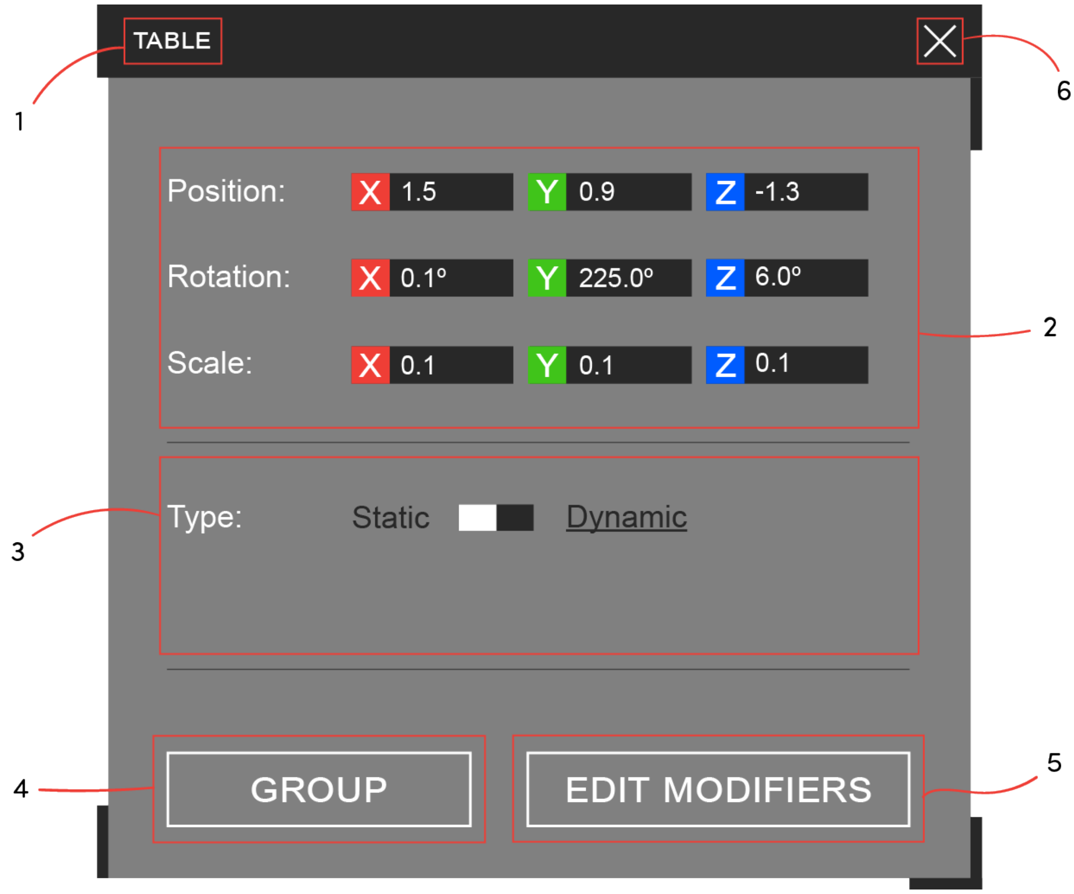


Figure 4.10: The properties window

3. Type toggle. This section allows the user to view and change the type of the selected object. Clicking any of the text elements will make them become the selected type, while clicking the switch in the middle will toggle between them. The change takes effect instantly, so if a static object is floating when pressing this or in a slanted surface, changing its type will make gravity act on it immediately;

4. Group button. This button allows the user to group two or more objects into a single one. Clicking it will let the user use the laser to select a secondary object to join to the one currently selected. Object grouping will be explained in more detail in section 4.3.6;

5. Edit modifiers button. Clicking this button will replace this window with a *Modifiers Window* associated with the currently selected object. This window will be explained in detail in section 4.3.7.1, just below;

6. Close button. Clicking this button closes the window.

4.3.3.3 Play

In this mode the user loses all selection and modification abilities. When in it, the user goes from being the creator to being a player and experiences the environment as an end-user would.

The main purpose of this mode is to allow the user to play-test the environment they are creating without the need to switch to another application to do so.

Different types of objects have different behaviours in this mode as explained in section 4.3.1.

4.3.4 Object selection and manipulation

Objects in VRDesigner can be selected in two different ways. We will explain how each of them functions and their main advantages and disadvantages.

4.3.4.1 Grabbing

The first of the two is the grabbing method. This method should feel the most natural of the two as well as the most intuitive.

To select an object, simply move your hand close to the object. If your hand is close enough and the object is available to be grabbed, it will shift to a blue colour and a yellow highlight will appear around it.

Press the grab button to have it attach to your hand, similarly to how you would pick an object in real life. The hand model, as well as the highlights, will disappear to make it easier to see the object when manipulating it, and make it easier to see how it fits in its surroundings before releasing it. The whole process can be seen in Figure 4.11.

After grabbing it, simply move your hand to the desired position and the object will follow both the hand's position and rotation. Some objects can only be grabbed while not in play mode.

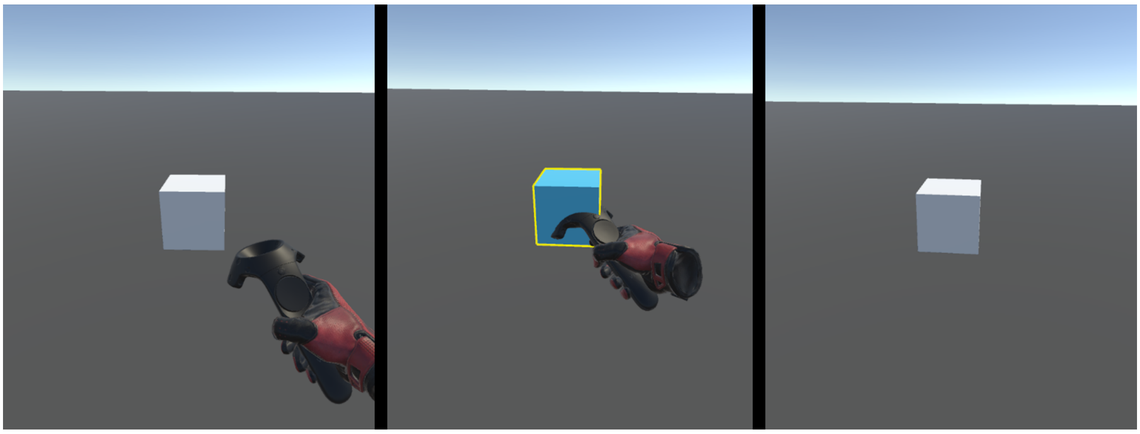


Figure 4.11: The process of grabbing an object. Approach - Hover - Grab

4.3.4.2 Laser

The second method is what we call the laser, for its aesthetic similarities to a real LASER. This method is activated when the grab button is slightly pressed while not being near any grabbable object. When doing this a blue beam shooting forward from your hand appears. The beam can be used to point to any object and, similarly to the grab method, when the object is available to be picked, it will turn blue and obtain a yellow highlight around it.

At the same time the laser should also turn red as another indicator that you can pick up the object. To do this, simply press further the grab button making the object attach to the laser. The laser will turn back to blue, but this time being slightly thicker, signalling that the object has been attached successfully. This process can be seen in Figure 4.12. When attached, you can then wave your hand around and the object will follow the laser to wherever you point.

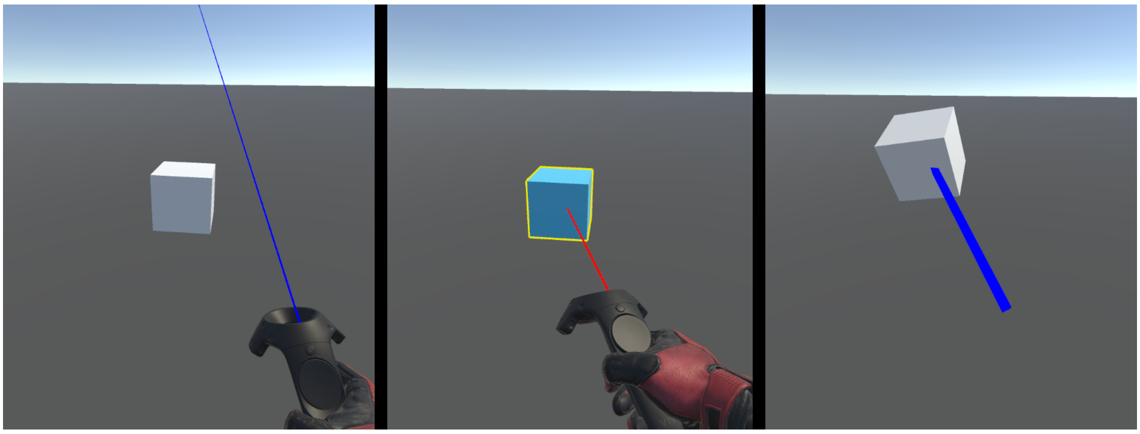


Figure 4.12: The process of *lasering* an object. Enable - Point - Attach

This method has the advantage of being able to manipulate objects at a much longer distance, but in turn has a couple of disadvantages, such as being less precise and not allowing much control over the rotation of the object. These disadvantages, however, are counteracted by some object transformation tools available to the user that will be explained in section 4.3.5.

4.3.5 Tool-based transformations

To aid in the user in the making of the perfect environment, some additional tools were created to give the user more creative control over the objects and make them easier to transform. These tools are only available in edit mode and while the user is grabbing or *lasering* an object. When these conditions are met, the menu button can be used to toggle between the available tools instead of opening the main radial menu.

After the wanted tool is selected, the user can now use the trackpad on their controller to use the tool. Different tools may respond to different gestures on the trackpad depending on what kind of action they represent. Three different tools exist: Distance, Scale and Rotation.

4.3.5.1 Distance Tool

This tool is only available while *lasering* an object. With this tool, the user is able to move the object close or further away along the laser. To do this, the user simply needs to use the trackpad in the controller in a scroll motion, where scrolling up will make the object go farther and scrolling down will make it come closer, as can be seen in Figure 4.13.

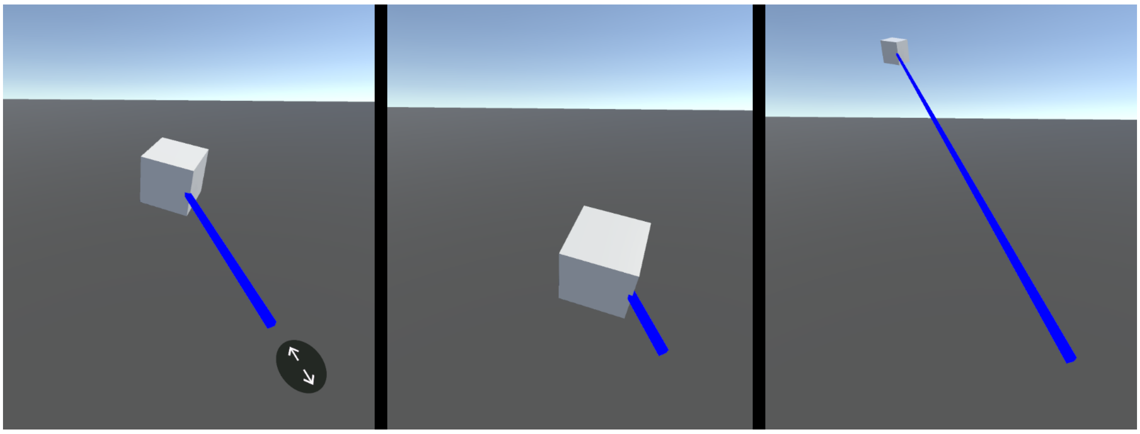


Figure 4.13: Distance tool being used

4.3.5.2 Scale Tool

This tool is available both when grabbing and *lasering* an object. With this tool, the user is able to change the size of the object they are holding. Similarly to the Distance tool, this can be done by simply using the trackpad in a scroll motion. Scrolling up makes the object bigger, while scrolling down makes it smaller. This behaviour can be seen in Figure 4.14. The scroll motion is designed to have acceleration, and so, faster movements will result in bigger and faster changes to the object, while slower movements will allow the user to control the object with more precision.

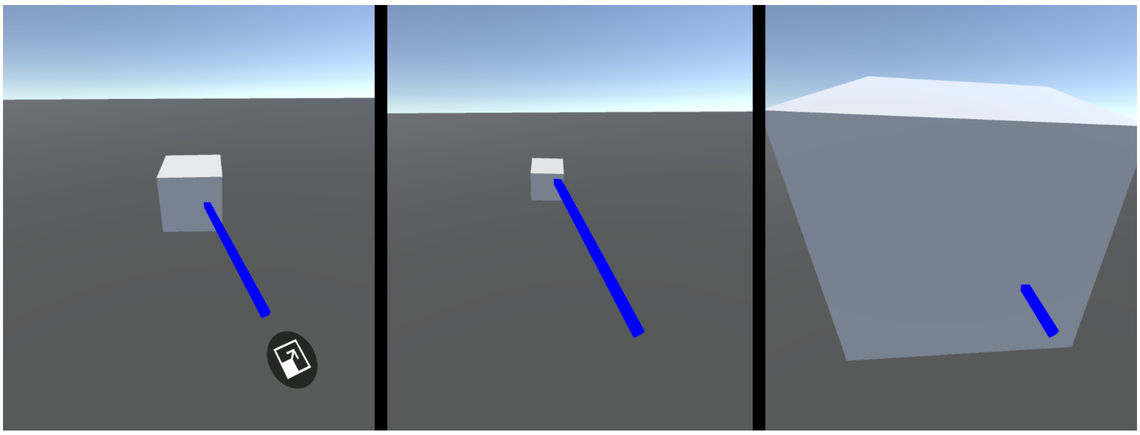


Figure 4.14: Scale tool being used

By default the scaling is done uniformly in all 3 axes, but it is possible to select each axis individually to achieve non-proportional scaling. This can be done by simply touching the trackpad without dragging for just under one second, at which point a direction selection overlay will appear above the controller with the options to do so, as was explained in section 4.2.4.

4.3.5.3 Rotation Tool

This tool is available both when grabbing and *lasering* an object. With this tool, the user is able to change the rotation of the object they are holding. Differently to the other two tools, the trackpad functions as a radial input instead of a scroll input. This means that, to affect the rotation of the object, the user must rotate their finger circularly around the trackpad to offset the object by the desired amount. As an example, if the user wishes to rotate an object by 180^o, then all they need to do is to select the rotation tool and do a 180^o circular motion on the controller's trackpad. This behaviour can be seen in Figure 4.15.

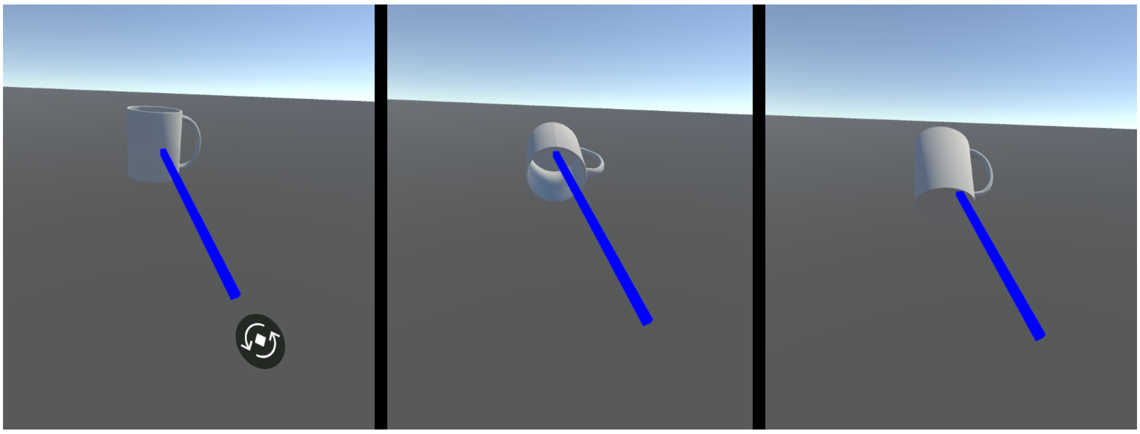


Figure 4.15: Rotation tool being used

By default the tool rotates around the *X* axis, but similarly to the scale tool, it is possible to select other axes by enabling the direction overlay in the same manner. The functioning of this overlay was explained in section 4.2.4.

4.3.6 Grouping objects

When creating an environment, the ability to group objects may prove to be very useful. The ability to group objects, gives the user the power to:

- move or edit objects together;
- add a modifier to a collection of objects;
- build more complex objects out of simple ones.

In its current state, grouping objects together effectively turns them into a single object. For future work, the ability to manage object groups and ungroup parts of them through a dedicated *Window* could be added. This would make groups an even more useful and powerful tool.

4.3.7 Procedures / Modifiers

Procedures or modifiers are tools the user can employ to turn tedious or specialized tasks into something that can easily be applied to any object.

4.3.7.1 Modifiers Window

This window is used to edit, add and remove the modifiers applied to the selected object. This window can be opened through the *Properties Window* as explained in section 4.3.3.2. It features the following elements:

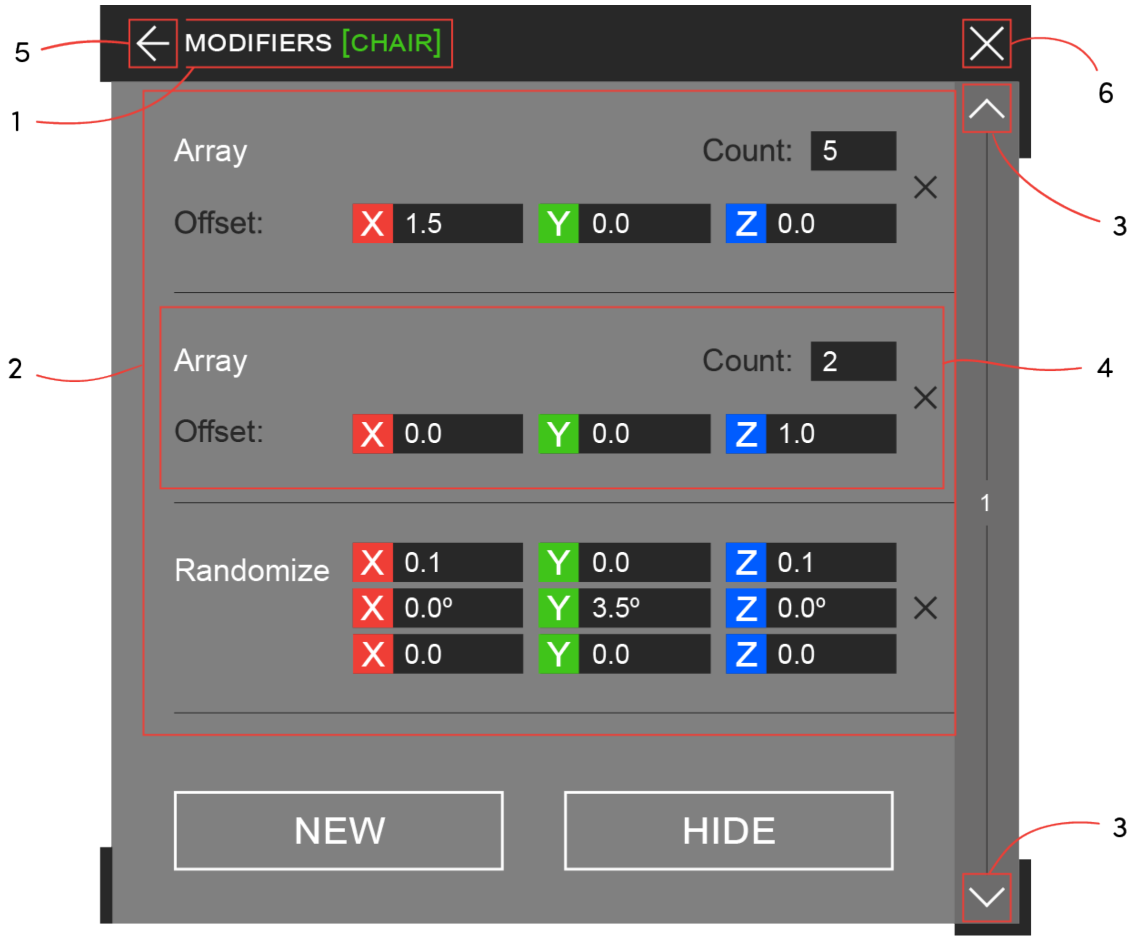


Figure 4.16: The modifiers window

1. Object name. Similar to the *Properties Window*, this indicates the name of the selected object;

2. Modifiers. At any time, the user can see up to three of the modifiers currently applied to the object;

3. Page arrows. Similarly to the *Search Online Window*, clicking these arrows takes the user to the next or previous page of three modifiers. When no more pages are available in one of the directions, the arrows will disappear accordingly;

4. Modifier Element. Different modifiers can have different properties and thus different elements to edit them. Detailed information about each of the available modifiers can be found in the next sections;

5. Back arrow. Clicking this arrow takes the user back to the *Properties Window* they came from;

6. Close button. Clicking this button closes the window.

Although more ideas for modifiers existed, only the ones below ended up being implemented. For future work, additional modifiers could be added.

4.3.7.2 Array Modifier

An array modifier is a tool that allows the user to replicate an object multiple times with controlled spacing between them. An example of its use would be in the creation of a classroom. Instead of the user having to place every table and chair individually, simply using 2 array modifiers could create an n by n grid of the intended object.

Below we see the properties of an array object:

Figure 4.17: The array modifier

1. Name of the modifier. This helps the user distinguish between modifiers;

2. Amount indicator. With this element, the user can indicate how many replicas of the original object they want;

3. Positional offset. Changing this, will change the distance between each of the replica objects.

4.3.7.3 Randomize Modifier

This modifier simply works to randomize properties of the object. If applied after an array modifier, it will affect each of the replica objects differently.

Figure 4.18: The randomize modifier

1. Name of the modifier. Used to identify the modifier;
2. Maximum offsets. These vectors indicate the maximum amount the object can be affected by. If the modifier has a 1.0 in the x of position, then it will randomize the object's position between +1.0 and -1.0 of its current position.

4.4 World

In this section, we will analyse the modules related to navigating the world, changing it as a whole, and when complete, exporting it.

4.4.1 Teleportation

To move around the virtual environment, the player is equipped with the capability of teleportation. Similarly to many other VR applications, pressing the trackpad on the controller makes a teleportation arc appear, as seen in Figure 4.19. This arc can then be aimed by pointing with the hand, to precisely choose the teleport location. Releasing the trackpad will trigger the teleportation instantly. A brief black screen will flash to help prevent the induction of nausea in the user.

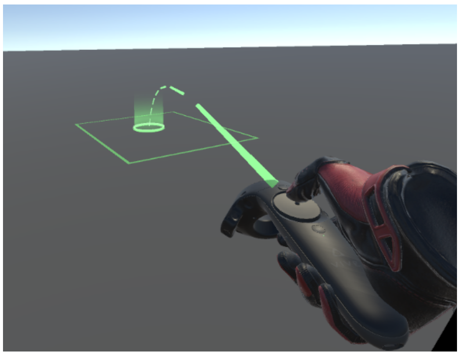


Figure 4.19: Teleportation arc

The teleportation arc interacts differently with different types of objects and depending on the current interaction mode. These interactions are explained in section 4.3.1.

4.4.2 Lighting

When creating an environment, different lighting conditions can drastically affect the perception of the space, and lead to different decisions at the time of building. Because of this, we gave the user the ability to toggle between 4 lighting conditions. Each a specific time of day, being thus called Dawn, Noon, Dusk and Night. Changing to different times of day changes two aspects of the lighting, as seen in Figure 4.20:

- Sun light. Changing the time of day will alter its orientation, position and intensity. This light is mostly responsible for the specular component of the illumination.
- Skybox. Changing the time of day will toggle between 4 different HDRIs, each representing a different time of day and emulating the appropriate light conditions. This is mostly responsible for the ambient component of the illumination.

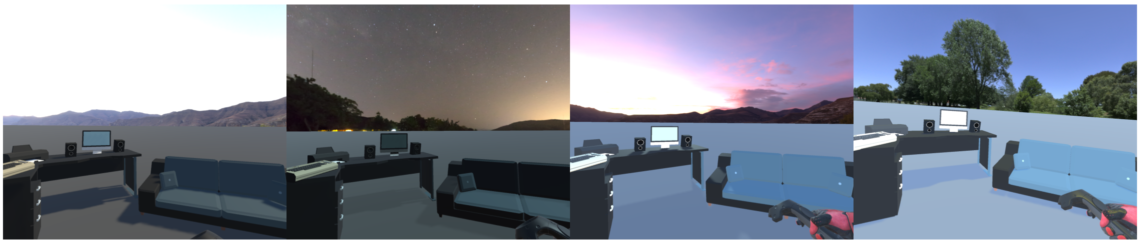


Figure 4.20: The four different light conditions available

4.4.3 Saving / Exporting the world

At any point, the user has the ability to export the created environment through the respective option in the radial menu (section 4.2.1). Doing so will create an OBJ file at a specific directory in the program's directory. The file will be named with the timestamp of when the export action is called. In this file, only the objects placed by the user will be stored.

For future work, the ability to choose the file name and save directory of the OBJ file could be a quality of life improvement for some use cases.

4.5 Proposed Workflow

In this section, we will go through narrating the originally planned workflow from section 4.1, but this time detailing how a user would go about executing it, using our prototype.

Using the same example, of a user that would like to create a simple classroom scene, with 4 chairs in a line, the teacher's desk and chair and a chalkboard on a wall.

The first step would be to import the necessary objects. To do this, the user would open the *Radial Menu*, and go the *Add Object* option, followed by the *Search Online* option. Doing this will start the *Speech Recognizer* and all the user needs to do is say what object they want to search for. This can be seen in Figure 4.21.

Figure 4.21: Initiating voice controls to search for online objects

At this point, a window will appear in front of the user, showing thumbnails of the search results. The user can navigate between pages to try to find the object that most closely matches their vision, at which point, simply clicking on it, will import it into the world. After the first object is imported, clicking on the search icon on the top of the window, will allow the user to search for other objects. Repeating this procedure for each object, the user can quickly import all the necessary objects into the scene. This can be seen in Figure 4.22.

Figure 4.22: Importing objects from a repository to the scene

The next step is to place the objects in the correct position. For this, the user goes into *Edit Mode* to be able to grab the objects using the hand or the laser. Once grabbed, the user can scale, rotate and move the object as desired. Doing this for each of the objects, the user can quickly create a rough first build of the scene. This can be seen in Figure 4.23.

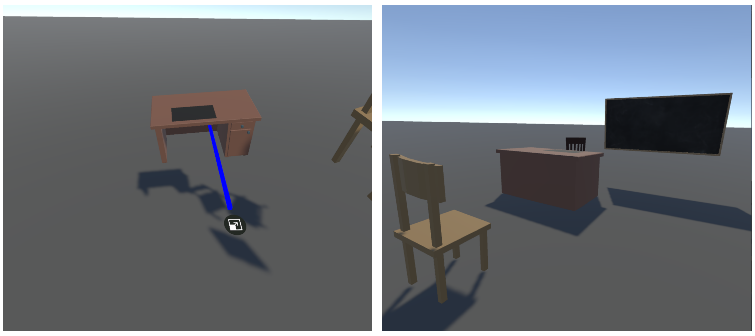


Figure 4.23: Positioning the objects

To place a wall behind the chalkboard, the user can import a plane through the primitives menus. To do this, the user needs to open the *Radial Menu*, select the *Add Object* option, followed by the *Primitives* option, and finally, select the *Plane* from the available primitives. The only thing that is left is to scale it to match the size of a wall and place it behind the chalkboard. This can be seen in Figure 4.24.

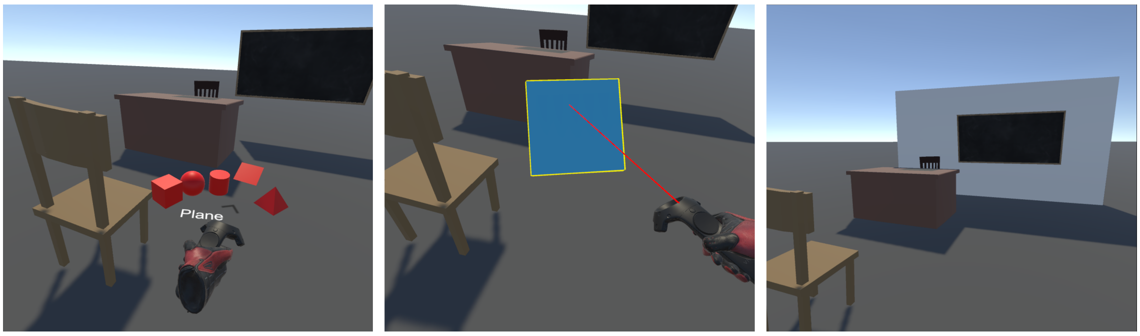


Figure 4.24: Using a plane primitive as a wall

Now, our classroom is almost complete, but it missing some chairs for the students. To populate the room, the user goes into *Select Mode* and selects the chair. This will open the *Properties Window* where the user can then click to open the *Modifiers Window*. In this window the user can an array modifier with a count of 4 and an offset in the negative *Z* direction. This can be seen in Figure 4.25.

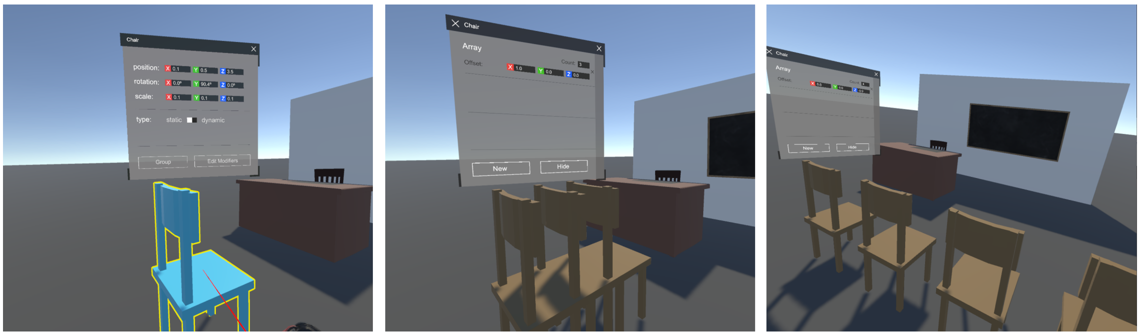


Figure 4.25: Applying an array modifier to an object

Our scene is now complete and we can spend some time adding details to bring it to life. All that is left, is to go into *Play Mode* and test our newly created scene. It can be seen in Figure 4.26.

A video of this process can be seen online[1].

[1]Online resources: https://tutamkhamon.com/thesis/content/

Figure 4.26: The complete scene

4.6 Summary

This chapter explored the individual modules that make up our prototype and how they would be used together to achieve our goal of creating a virtual environment using VR tools, speech recognition and online 3D repositories.

In the next chapter, we detail the testing procedures and methodologies, as well as the results obtained and what can be learned from them.

Chapter 5

Testing and Evaluation

In this chapter we will detail all the testing done to evaluate this project, the methodologies used, the results obtained, and what conclusions were gathered.

The objective of this dissertation, as set forth by the research questions (section 1.2) is two-fold:

- To understand how VR and multimodal interaction can help the field of 3D environment creation. How does it compare to existent alternatives both in ease of use and in speed?
- To find out if, using the intuitiveness inherent to voice and VR gesture controls, it is possible to allow users without technical skills and even without VR experience, to quickly learn and be capable of creating their own 3D environments.

To research both of these objectives, a prototype was developed as detailed in chapter 4 and tested according to the tests listed in section 5.1 and explained in section 5.2. The tests were then carried as detailed in section 5.3 and the results gathered and analysed in section 5.4. The conclusions reached were then put forth in section 5.5.

5.1 Solution Evaluation

To evaluate our work, several tests have been planed:

- Firstly, to evaluate the individual features of the prototype, a list of simple tasks will be created for users to replicate. They will be timed and the results compared to a control group of 3D modelling professionals using their 3D software of choice. The users will vary in experience both of 3D modelling and VR.
- After testing individual features, the prototype will be tested as a whole. For this the users will be asked to replicate a list of scenarios according to some established rules. Once again they will be timed, and compared to a control group. The scenes will be given in a different order for each user, preventing order bias.

- Finally questionnaires will be given to the users with questions based on their subjective opinion of the tool.

In section 5.2, these tests will be further detailed.

5.2 Testing Methodologies

To put the developed prototype to the test on the objectives established, we created an array of tests with varying degrees of complexity and objectives. The methodologies behind these tests will be detailed in this section.

5.2.1 Simple Tasks

To test how intuitive the prototype we created is, and if, in fact, it allows non-experienced users to create 3D environments, we devised a set of tasks, consisting of simple interactions that required the use of the different features of the prototype.

The tasks, as seen in Figure 5.1, were:

ST1 Firstly, to create walls for a room using the add primitive feature. In this test, the test subject was expected to, using the plane object primitive (section 4.3.2.1), show their abilities at interacting with a simple object and manipulate its position, rotation and scale to achieve a pre-determined result;

ST2 Next, to create a rectangular table suitable for 10 seats, again using primitives. In this test, the test subject was expected to use a combination of 5 cube primitives and, scaling them appropriately, create a table with a top and four legs. In the end, the test subject simply needed to use the grouping feature to group the 5 primitives into a single object;

ST3 Following that, to search online for a chair and place it by the table. In this test, the test subject was expected to use the voice controls to search for a chair, choose one to import from the results, and scale it and move it to fit with the table;

ST4 And finally, to multiply the chairs as to have 5 on one side of the table. In this test, the test subject was expected to use the selection mode to access the properties of the object, add an array modifier and edit it to accomplish the task.

Each of the tasks focused on evaluating different individual features of the prototype, and these were what we paid most attention during their execution. To help keep our evaluation organized, we defined the key points of each task as being, respectively:

ST1 Importing of primitives, interacting with an object, and manipulating its position, rotation and scale;

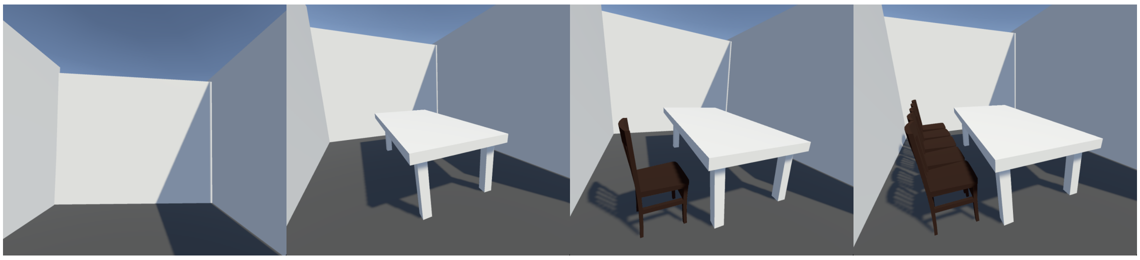


Figure 5.1: The sequence of the four simple tasks

ST2 Kitbash modelling, using simple objects together to create more complex objects and object grouping;

ST3 Using voice controls to search for objects, importing objects from online repositories and simple window-based interactions, like selecting and clicking elements;

ST4 Using modifiers to do procedural tasks, and complex window-based interactions, like managing modifiers and editing values.

To have a baseline against which to compare the test subject's performance, we got 3D modelling professionals to do equivalent tasks in a 3D modelling software of their choosing, while being timed. Our objective for this was to compare the time each task took to complete on our prototype, against this baseline. With this, we can see which specific features bring an advantage to our prototype in competing with traditional modelling software.

During this test, by observing the struggles, or lack of them, that the test subjects might go through, we could also gain some insight, together with the questionnaire, into the usability and intuitiveness of the different systems.

5.2.2 Environments Creation

The main focus of our prototype has been, since the beginning, to allow the use of virtual reality, along with voice controls, online 3D repositories and procedural tools, to help in the process of environment design and creation. Not only that, but our goal was to make it a better experience, faster, and easier to learn than traditional 3D modelling tools. With that in mind, we developed a batch of tests that consisted of just that, 3D environment creation.

For this, we again resorted to 3D modelling professionals for the creation of test scenarios. We asked them to create three different 3D environments, using only objects from 3D repositories, and timed how long it took them to do it.

Each of the scenes would then be shown to the test subjects, as a guideline to what they had to replicate. Every scene had specific object requirements, and were only considered complete when all of them were present. Extra elements could be added, but, being that time was of the importance, the test subjects were advised not to misuse too much time polishing the scenes.

5.2.2.1 EC1 – Museum scene

This is the most simple scene of the three, as seen in Figure 5.2, and all that was required was for it to have the following elements:

- Two walls;
- Three paintings distributed by the walls;
- One bench;
- A security camera;
- A sculpture;
- A fire extinguisher.

Figure 5.2: Museum test scene

5.2.2.2 EC2 – Bedroom scene

The next scene was the bedroom of a child, as seen in Figure 5.3. The required elements were:

- A bed;
- A book shelf;
- A desk;
- A lamp;
- A trashcan;
- A floor mat;
- A window;
- Some type of toy.

Figure 5.3: Bedroom test scene

5.2.2.3 EC3 – Classroom scene

The last scene was that of a classroom and part of the corridor attached to it, as seen in Figure 5.4. The required elements were:

- The teacher's table;
- The teacher's chair;
- The blackboard;
- A bookshelf;
- A clock;
- A door;
- Four chairs and tables for the students;
- Nine lockers in the corridor.

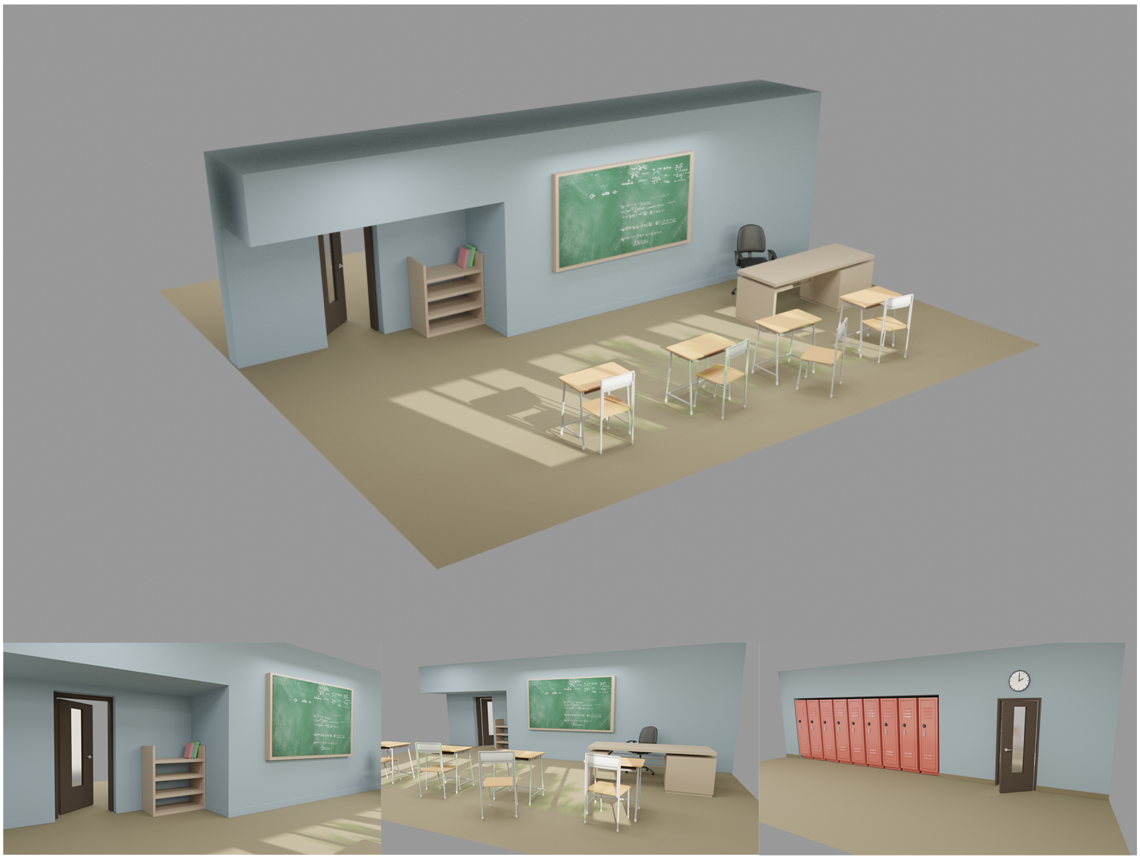


Figure 5.4: Classroom test scene

5.2.3 Questionnaire

A questionnaire was created to help gather extra information about the test subjects and their side of the tests.

It starts by collecting information about the test subject's previous experience with VR, 3D modelling and speech controls.

After that, the subject is asked a series of quantitative questions pertaining to their tests and the overall experience using the tool.

Finally, it ends with a more open-ended and subjective section where the test subjects can express their opinion about *VR Designer*, what they think could be improved and if they would see themselves using a more complete version in the future. The full questionnaire can be seen in Appendix B.

5.3 Testing Procedure

Eight different test subjects with varying degrees of VR experience and 3D modelling proficiency were gathered. Each of them was given an id number at the beginning of the test to later allow us to cross-reference results from different tests, while still keeping test subject anonymity. All of them went through the same testing procedure which was divided into 5 different steps as will be explained.

5.3.1 Declaration of consent

To begin the tests, the test subjects were informed about the purpose of the study they were volunteering for, and how their information was going to be gathered and treated. They were warned about any discomfort they might feel during the use of VR, and about their rights during the test, including that of leaving at any point and withdrawing from the study. They finally were asked to confirm their acknowledgement of all that was explained, and their consent to proceeding to the tests.

5.3.2 Introduction to VR

If the subject was new to VR, some time was taken to let them get familiar with the basics: moving their head to look around; moving the controllers to affect the virtual world around them; physically moving around; all the buttons present on the controllers. We proceeded to the next step as soon as the subject felt accustomed and comfortable with their virtual presence.

5.3.3 Familiarisation with *VR Designer*

After that, it was time to get the test subjects into *VR Designer*. Once that was done, we started by explaining the controls to them and, one by one, test all of the features of the prototype. To help organize this step, a checklist was constructed detailing all the tasks the test subject would have to go through. This list was followed in order and can be seen below:

1. Teleport around the scene;
2. Open and navigate the *Radial Menus*;

3. Add a primitive to the scene;
4. Search for an object using voice controls;
5. Import an asset from the search results;
6. Change into *Edit Mode*;
7. Close the *Search Online Window*;
8. Grab and move the object around;
9. Manipulate the objects' scale, position and rotation;
10. Change into *Selection Mode*;
11. Open the *Properties Window* of the object;
12. Modify values in the *Properties Window*;
13. Group the imported object and the primitive;
14. Open the *Modifiers Window* for the object;
15. Add a modifier and edit its values;
16. Export the scene;

After all the steps were completed, the subjects were given 5-10 minutes to explore the prototype's capabilities. This was important to get the test subjects familiar with all the features available to them so that they could complete the tests to the best of their abilities. To clarify, one of our goals was to test that subjects could learn to use *VR Designer* in a considerably short timespan, not that they could use it without having learnt how to. For this reason, we find that this learning process was important for the reliability of the results obtained.

5.3.4 Simple tasks test

In this step, we both wanted to see how well the test subjects retained the information they just learned, and how they compared to the control group, as explained in section 5.2.1. After resetting the scene, we instructed the subjects to do each of the 4 tasks in order and timed their duration. Between each task, we instructed the subject to export the scene as to have it available for later analysis. The subjects were informed that their performance would be timed, but were encouraged to complete them at a comfortable pace without the need to rush them.

This test was important in setting up the next, more complex test, because, not only did it serve to gather results on the usability and intuitiveness of the tools, but also introduced the subject to the complete workflow needed to create a scene.

5.3.5 Scene creation test

For this test, the test subjects were shown a scenario that had been previously prepared by a professional 3D modeller, using models from free online 3D repositories. They were told to replicate them, paying attention to all the objects present in the scene. They were also given the list of the required objects for the scene to be considered complete, and were allowed to consult it at any point. The time taken to complete the scene was measured and the completed scene was exported for later analysis.

This was repeated for each of the three scenes, although the order in which they were given to the test subjects changed, as to avoid order bias in the results.

5.3.6 Questionnaire

After both tests were complete, the test subjects were taken out of VR and asked to fill a brief questionnaire regarding their experiences with *VR Designer*, as explained in section 5.2.3. The complete questionnaire can be seen in Appendix B.

5.4 Results

In this section we will analyse the results we obtained in the two tests, together with data from the questionnaires, to see how our prototype answers the research questions established. We need to be mindful of the small size of the test group, meaning that our conclusion should be taken as indications of what would be expected from a larger group. This will, therefore, work as a preliminary study, requiring a more extensive evaluation in the future.

5.4.1 Experience Questionnaire

In the second section of the questionnaire (section B.2) we gathered information the test subjects' previous experience with 3D modelling, use of VR and use of speech controls, to better help us characterize our sample. The results were:

5.4.1.1 3D Modelling experience

To understand our test subjects level of proficiency with 3D modelling, we asked the following questions:

- How much experience with 3D modelling and environment creation do you have? The answers ranged from 1 to 5, 1 meaning *No experience at all* and 5 meaning *A lot of experience*. The results can be seen in Figure 5.5.
- Have you done any profession 3D modelling before? This was a binary question with the answers being simply *yes* and *no*. The results can be seen in Figure 5.6.

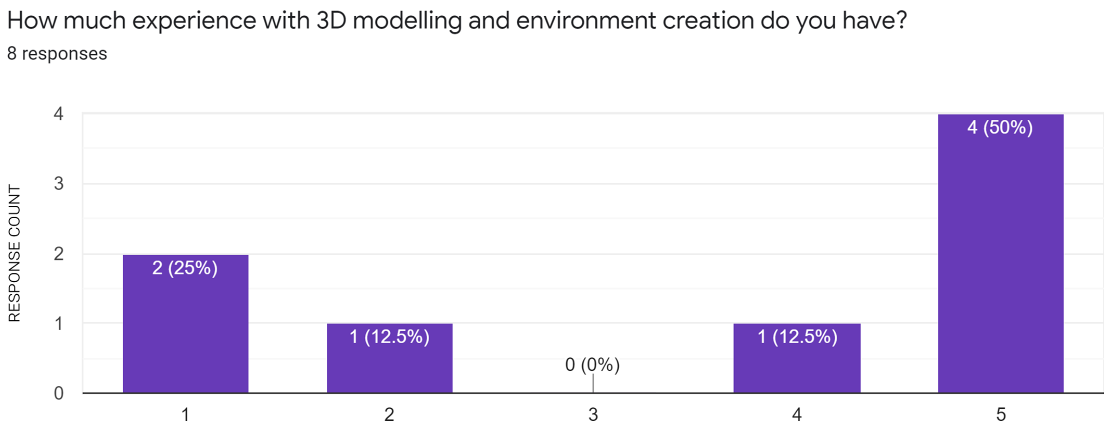


Figure 5.5: Distribution of modelling experience in our test group

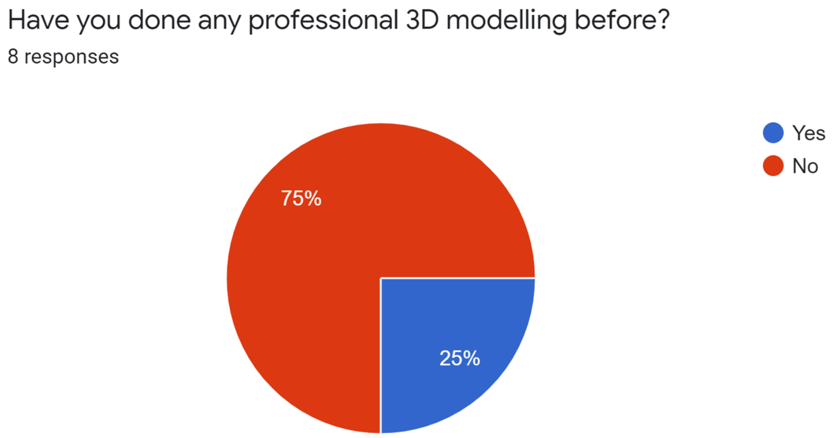


Figure 5.6: Professional modelling experience in our test group

Upon analysing this data, we can conclude that our user sample is adequately diverse. We can see, however, that they tend to group up at the extremities. Because of this, when analysing the test results, we will be referring to the group of people in the 1 and 2 levels as *Modelling Inexperienced*, and the group in the 4 and 5 levels as *Modelling Experienced*. As could be expected, all the subjects who had previous professional 3D modelling experience fell in the group with an experience level of 5.

5.4.1.2 VR Experience

To understand the test subjects' level of experience with VR, we asked the following question:

- How much experience do you have with Virtual Reality? The answers ranged from 1 to 5, 1 meaning *No experience at all* and 5 meaning *A lot of experience*. The results can be seen in Figure 5.7.

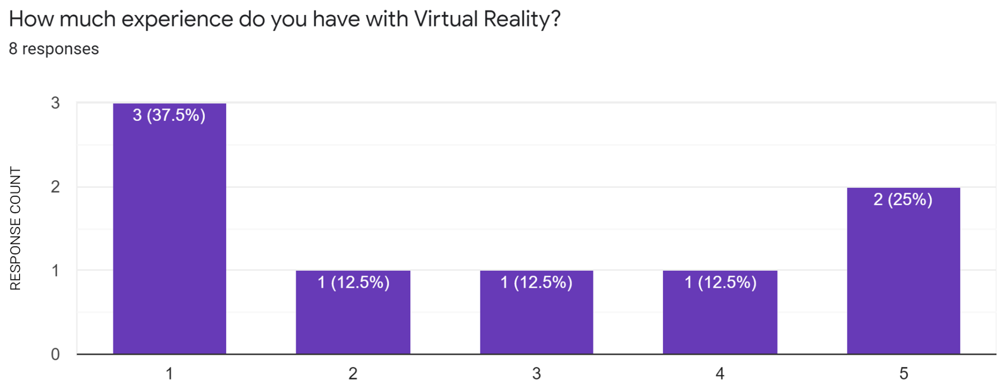


Figure 5.7: Distribution of VR experience in our test group

Again we can see that our group is sufficiently diverse, but again, having a skew to the extremities. When analysing the test results, we will be referring to the group of people in the 1 and 2 levels as *VR Inexperienced* and the group in the 3 to 5 levels as *VR Experienced*.

This polarisation of the study group in both of the results can be good when analysing the results because, with it, we have the ability to compare extreme use cases. In fact, the user group was chosen having this in mind and people with and without experience in both areas were chosen purposefully.

5.4.1.3 Speech Controls Experience

To understand how accustomed our test subjects were with the use of speech controls, we asked the following question:

- How often have you used voice controls in your life? The answers ranged from 1 to 5, 1 meaning *Never* and 5 meaning *Quite regularly*. The results can be seen in Figure 5.8.

Here we see a slightly diverse sample group, with a heavy skew toward the inexperienced end. Having this in mind, when analysing the results of the tests, we will be referring to the group of people in the 1 and 2 levels as *Speech Inexperienced* and the group in the 3 and 4 levels as *Speech Experienced*.

5.4.1.4 User Sample Analysis

To have a better understanding of the distribution of our sample group, we can present it as a Venn diagram, as can be in Figure 5.9. For this diagram we can see that all of our subjects with VR

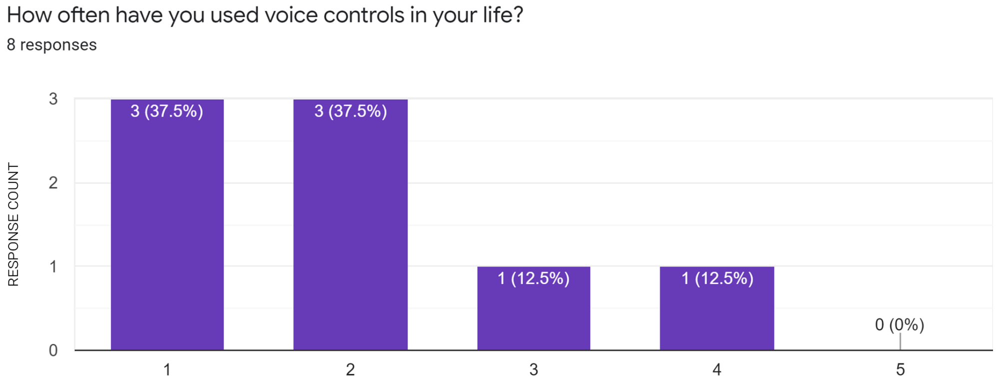


Figure 5.8: Distribution of speech controls experience in our test group

experience also have 3D modelling experience. We need to keep this in mind when analysing our results because it will skew the results into having VR experience and Modelling experience pointing to the same conclusions. We can also observe that 1 subject has experience in all of the three aspects while 2 subjects are inexperienced at all three.

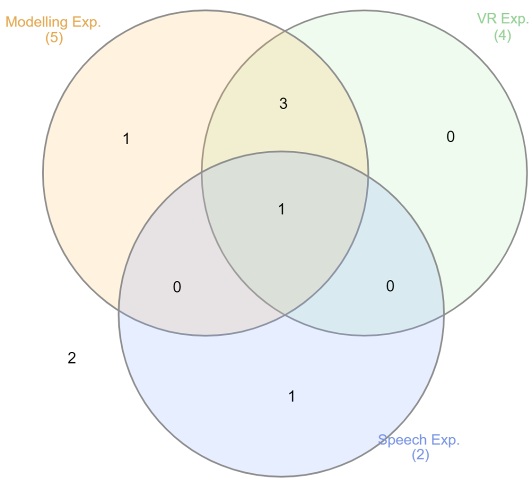


Figure 5.9: Venn diagram of the distribution of our sample group in relation to their experience in 3 different aspects

5.4.2 Simple Tasks

The *Simple Tasks* test was made to analyse to which tasks our prototype brings benefits when compared with traditional 3D modelling tools. The full results of this test can be seen on Table A.2. Each of the tasks will be analysed in depth.

5.4.2.1 ST1

The first task (ST1) was to have the subject create 4 walls of a room. This task tests the use of primitives, the subjects's interaction with objects and their manipulation.

The results of this task can be seen in Table 5.1.

	Task id	C1	C2	101	102	103	104	105	201	202	203
Create walls for a room	ST1	0:35	0:29	1:52	1:31	1:10	1:03	0:58	1:51	2:20	2:13

Table 5.1: Time measurements of ST1

Plotting this data on a boxes and whiskers chart, grouping people based on their VR and 3D Modelling experience (Figure 5.10), we can see that subjects with more 3D and VR experience outperform the ones without. We can also see that the control group, that is, 3D professionals using their 3D modelling software of choice, greatly outperforms all of the groups. This tells us that the use of primitives for modelling and their manipulation in 3D space, at least in strict geometric terms, is not one of the advantages our prototype brings.

Modelling experience and VR experience, at a first glance, seem to show the same advantage when compared to their inexperienced counterparts, but remembering that the VR experienced group is contained in the modelling experienced group, not many conclusions can be reached with that information.

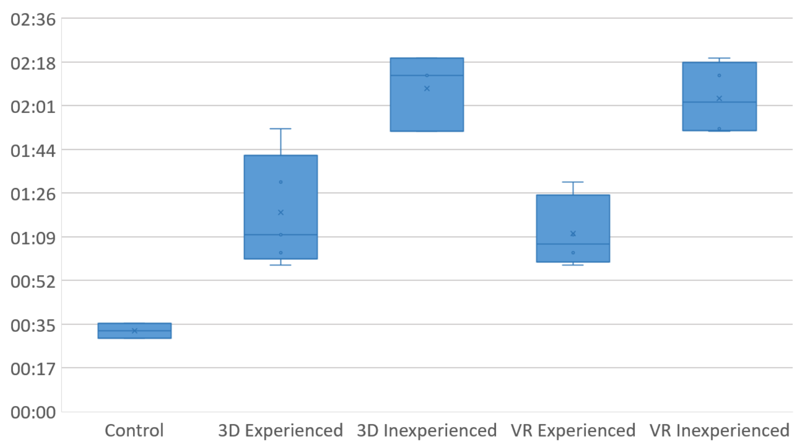


Figure 5.10: Box chart of the results of ST1 grouped by experience groups

5.4.2.2 ST2

The second task (ST2) was to have the subject create a table, using the combination of four cube primitives. This task tests the ability to do kitbash modelling, to create complex objects out of simpler ones, using our prototype.

	Task id	C1	C2	101	102	103	104	105	201	202	203
Create a table using primitives	ST2	1:23	2:00	3:53	2:58	2:16	2:35	2:35	5:12	6:01	4:44

Table 5.2: Time measurements of ST2

The results of this task can be seen in Table 5.2.

Plotting this data on a box chart, grouping people based on their VR and 3D Modelling experience (Figure 5.11), we can see results very similar to the previous task. Control outperforms experienced subjects who, in turn, outperform the inexperienced ones. And again modelling experience and VR experience seem to show the same advantage when compared to their inexperienced counterparts.

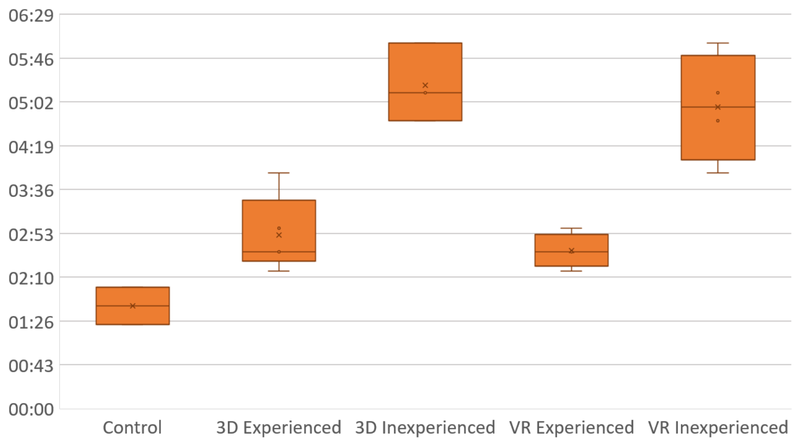


Figure 5.11: Box chart of the results of ST2 grouped by experience groups

5.4.2.3 ST3

The third task (ST3) was to have the subject import an object from an online 3D repository. This is one of the core functionalities of our prototype, and the one we expect to have the best chance at competing with the control group.

The results of this task can be seen in Table 5.3.

	Task id	C1	C2	101	102	103	104	105	201	202	203
Place an imported chair by the table	ST3	2:35	2:52	0:41	0:35	0:33	0:24	0:30	0:32	0:23	0:41

Table 5.3: Time measurements of ST3

Seeing as, to use of the feature being tested, the subject is required to use speech controls in our prototype, the data is again plotted on a box chart, but this time comparing the speech experience of the subjects as well, as seen in Figure 5.12. With this we expect to find out if previous experience at using voice controls affects the performance of this feature.

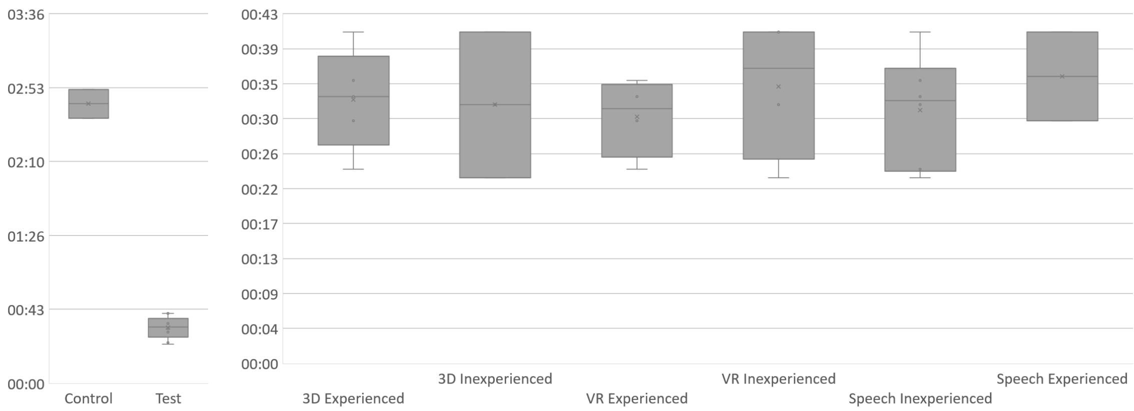


Figure 5.12: Box chart of the results of ST3 grouped by experience groups

As can be observed in the chart, and confirming our expectations, the test subjects consistently outperform the control. However, unlike the previous tests, experience in the three relevant aspects seems to not have any positive or negative effect on the results. Our read on this is that, the 2 main mechanics that this feature relies on, speech controls, and virtual windows, are intuitive in their nature. Speech controls because that is the human's most natural way of communication and the virtual windows because we, in the 21st century, have been exposed to the point and click paradigm inherently associated with digital windows so much, that the step of learning how to use them in VR becomes trivial.

Overall, these results contribute to answering our first research question: How can VR and multimodal interaction approaches be used together to help the process of 3D environment creation? We will discuss this in more detail in the conclusion in section 5.5.

5.4.2.4 ST4

The fourth task (ST3) was to have the subject use the modifiers feature of the prototype to automate the replication and placement of an object at random intervals.

The results of this task can be seen in Table 5.4.

	Task id	C1	C2	101	102	103	104	105	201	202	203
Place 5 chairs on one side of the table using modifiers	ST4	0:43	0:25	0:40	0:53	0:51	0:43	0:50	1:45	1:37	1:52

Table 5.4: Time measurements of ST4

Plotting this data on a box chart yields the results seen in Figure 5.13. Analysing the chart, we can see that subjects with 3D and VR experience are only slightly slower at the task then the control group. We also observe that the 3D and VR inexperienced groups are drastically outperformed by the control and experienced groups.

Looking further into these results, and looking back into the distribution of our test group, as seen in the Venn diagram in Figure 5.9, we can see that the difference between the *Modelling Inexperienced* and the *VR Inexperienced* groups, is of only one subject that is modelling experienced but VR inexperienced. Taking this into account, if we look into comparing the 3D Inexperienced group with the VR Inexperienced group on the results of this task, we can see that the dramatic difference in the lower limit of the box is due to that one subject difference. Knowing this, we can safely assume that the massive differences in the completion time seen in these results is due to the modelling experience of some subjects and not due to their VR experience. This agrees with our previous findings that virtual window-based interactions, as is the case with this feature, suffer no observable advantage or penalty from the level of the user's VR experience.

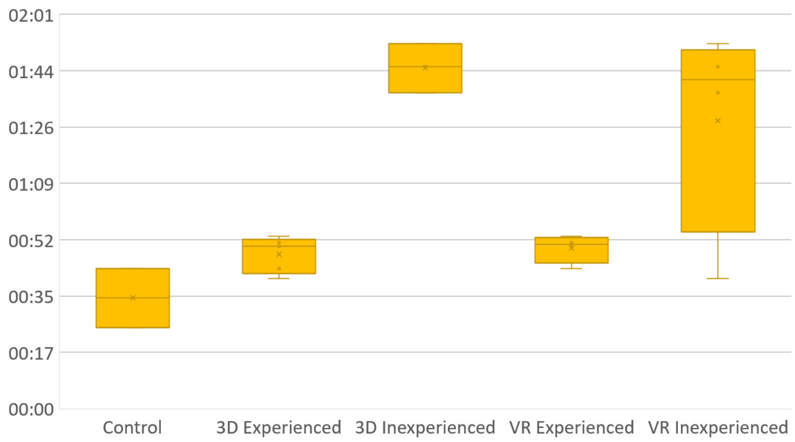


Figure 5.13: Box chart of the results of ST4 grouped by experience groups

5.4.2.5 STtotal

The STtotal is the sum of the completion times of all the previous tasks. Studying this metric, we hope to find if the possible time gains in some of the tasks are big enough to outweigh the potential time losses in others. Overall, the tasks were designed to simulate the basic elements of building a 3D environment, and so, these results reflect what performance, timewise, could be expected from our prototype.

The results of this task can be seen in Table 5.5.

	Task id	C1	C2	101	102	103	104	105	201	202	203
Simple Tasks	STtotal	5:16	5:46	7:06	5:57	4:50	4:45	4:53	9:20	10:21	9:30

Table 5.5: Total time measurements of the simple tasks

Plotting, once again, the data into a box chart (Figure 5.14) shows the results between the control group and both of the experienced groups to be very close, with some subjects even managing to be faster.

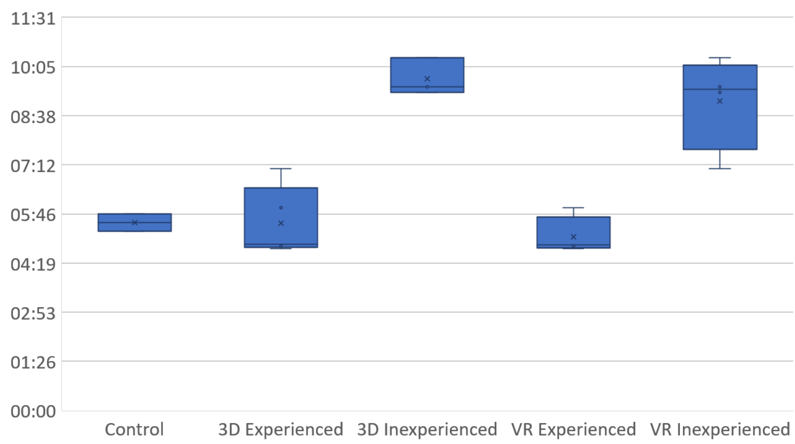


Figure 5.14: Box chart of the sum of the results from all the tasks (STtotal) grouped by experience groups

The speech experience vs inexperience comparison was omitted from this chart since it did not manage to cause negligible differences in time in any of the tests. Our conclusion is therefore that previous experience in the use speech controls brings no advantage to the user while using VR Designer. As explained before, our view is that speech being the natural mean of communication that it is, the adaptation process is almost inexistent. The inexistence of a positive or negative correlation is most apparent when plotting the STtotal times vs the speech experience levels in as scatter plot, as seen in Figure 5.15.

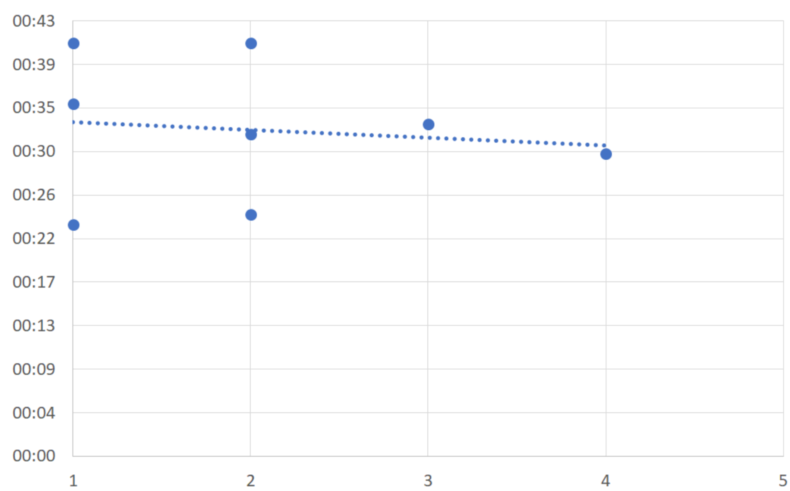
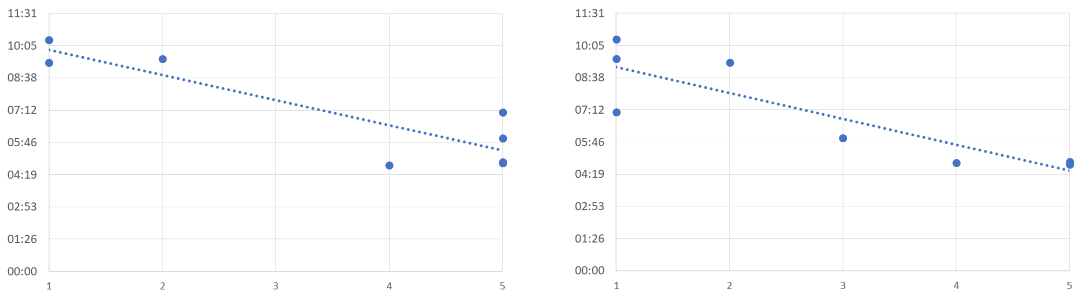


Figure 5.15: Scatter plot of the speech experience vs the total test (STtotal) time

As to modelling experience and VR experience, we can see that both groups obtain considerable leads when compared to their inexperienced counterparts. Our assumption is that this lead is mainly due to the modelling experience and not as much due to the VR experience, but being that the group of VR experienced subjects is contained by the group of modelling experienced subjects, further testing with a group of VR experienced, but not modelling experienced subjects, would be needed to confirm this.

Plotting the 3D and VR experience vs the time taken to complete the test, as seen in Figure 5.16, shows, in both cases, an apparent negative correlation between the two variables.



(a) Modelling Experience

(b) VR Experience

Figure 5.16: Scatter plot of the subject's experience vs simple tasks test time

Overall, we can see that our prototype allows users inexperienced in 3D modelling to perform simple environment creation tasks and experienced users to match the efficiency they would have using their software of choice. It is important to note, that despite some subjects having previous experience both in VR and in 3D modelling, this was their first time using our prototype. Because of this, it is reasonable to predict that their times would improve over a more prolonged usage.

5.4.3 Environment Creation Test

The *Environment Creation* test was designed to simulate the intended use cases for the prototype we developed. In this test, the subjects were instructed to replicate three test scenes and to include all the present objects, as detailed in a list given to them. This list can be seen in Table A.1. Their time to completion was measured and can be seen in Table A.3.

Since the three scenes were functionally identical, instead of analysing each scene individually, we are going to, instead, analyse the three scenes altogether as well as the total time taken to complete them. Figure 5.17 shows the scene completion times comparing different experience groups.

We can see that similarly to the *Simple Tasks* test, users with 3D and VR experience achieve lower times when compared to their inexperienced counterparts. However, in this test, even the totally inexperienced managed to achieve times lower than those of the control group.

This proves that our prototype achieves what we set out to do and answers positively to the second research question established: Can we enable users without specific technical skills to easily and intuitively create VR environments?

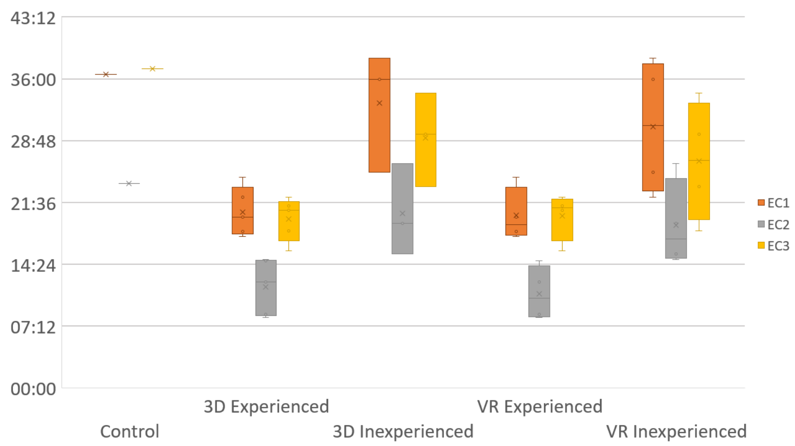


Figure 5.17: Box chart of the environment creation test results

Analysing the total times (Figure 5.18), shows us the same results, but this time with less variability in the results. This is due to the learning process that occurred during the test, being that users would tend to spend more time searching for things in their first scene, and less in the second and third. Because the scenes were given in a different order to each subject to prevent order bias, this means that some subjects would get the EC1 as their first, when they are still getting to know the ropes, and others as their last, when they are more proficient with the prototype, causing the high variability in times observed if Figure 5.17. Summing all the times together cancels out this variability, as seen in Figure 5.18.

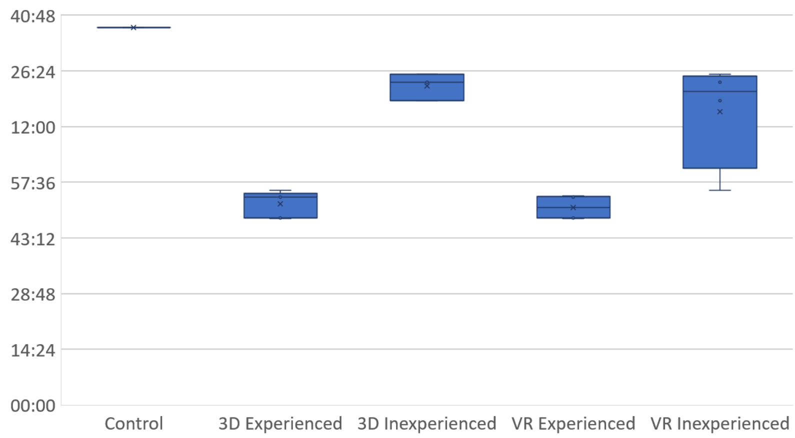


Figure 5.18: Box chart of the total environment creation times

If instead, we take the ratios between the time it took the subject to complete a scene and the control time, and grouped the scenes by their test order for each subject, we get the chart as seen in Figure 5.19. In it, we can clearly see the decrease in relative time from each scene to the consecutive one.

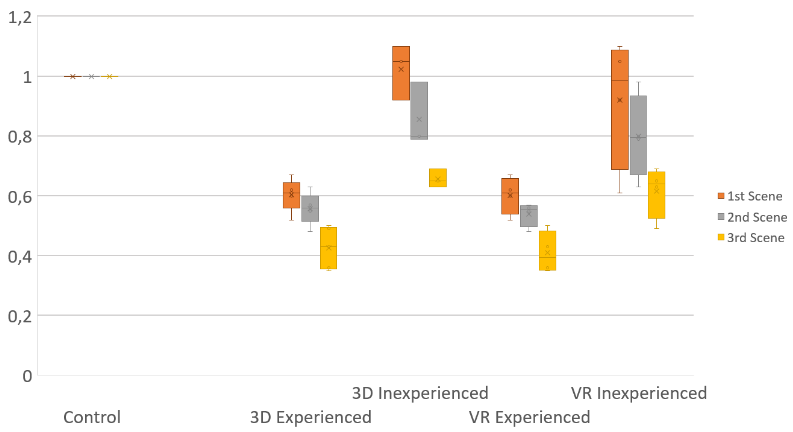
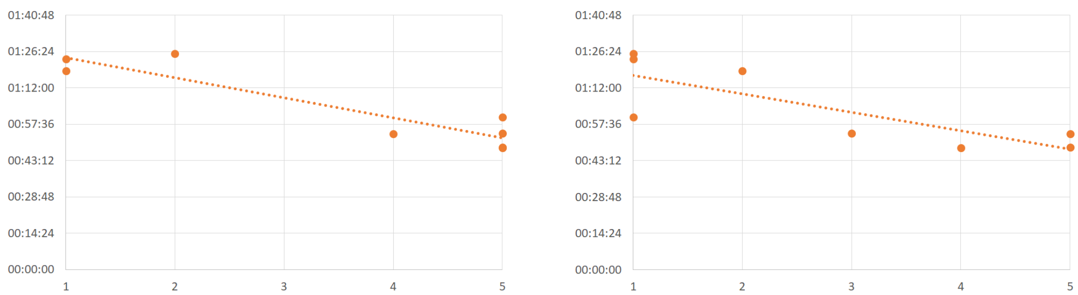


Figure 5.19: Box chart of the total environment creation times

And again, if we plot the 3D and VR experience vs the time taken to complete the test, as seen in Figure 5.20, we get an apparent negative correlation between the two variables.



(a) Modelling Experience

(b) VR Experience

Figure 5.20: Scatter plot of the subject's experience vs environment creation test time.

5.4.4 Usability Questionnaire

Section 3 of the questionnaire (section B.3), enquires the subjects about the usability of the prototype.

5.4.4.1 Interaction and Manipulation

To get some insight about what the subjects thought about the object interaction and manipulation, we devised the following three questions, with answers ranging from 1, meaning *Very Hard* to 5, meaning *Very Easy*:

3.1 How easy did you find it was to interact with the objects around you?

3.2 How easy did you find it was to manipulate the objects around you?

3.3 How easy did you find it was to choose the direction you wanted to manipulate an object?

The results obtained can be seen in Figure 5.21. We can see that we got no negative answers, with question 3.1 getting all max answers. This shows us that the overall object interaction fulfilled the requirements of being easy to learn and intuitive to use.

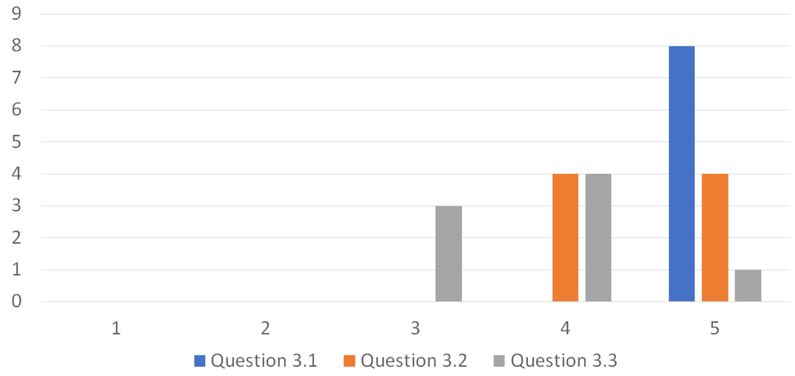


Figure 5.21: Results of questions 1, 2 and 3 of section 3 of the questionnaire

5.4.4.2 Access to 3D repositories

One of the core functionalities of our prototype is the ability to search for and import 3D objects from free online repositories. To see what the users thought about the usability of this feature, we asked them:

3.4 How useful did you find the direct access to free 3D repositories to be?

3.5 How easy did you find it was to import objects?

For both questions, the answers ranged from 1 to 5, where in question 3.4, 1 meant *Not Useful* and 5 meant *Very Useful*, and in question 3.5, 1 meant *Very Hard* and 5 meant *Very Easy*.

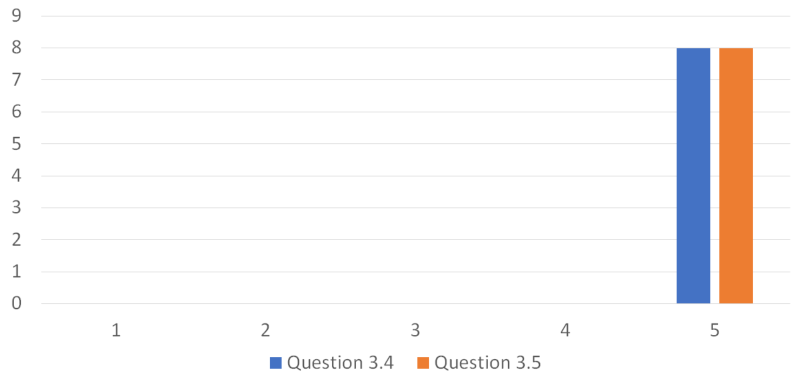


Figure 5.22: Results of questions 4 and 5 of section 3 of the questionnaire

The results for these can be seen in figure 5.22, and are overwhelmingly positive which validates our efforts into making this core feature as accessible as possible.

5.4.4.3 Voice controls

One of the major contributors for the accessibility of our prototypes are the voice controls. To understand how they were received by the test subjects, we created the following question:

3.6 How easy did you find it was to search for objects using your voice?

The results can be seen in Figure 5.23. They are mostly positive, although some improvements could be made to the speech recognition in the future.

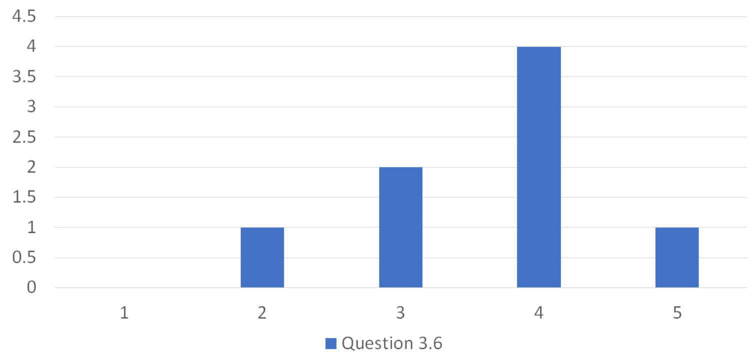


Figure 5.23: Results of questions 6 of section 3 of the questionnaire

5.4.4.4 Modifiers

Another big component of our prototype are the procedural tools, which we call modifiers. To see how users felt about them, we asked them:

3.7 How useful did you find the object modifiers to be?

The results can be seen in figure 5.24. Although mostly neutral, we feel like the modifiers module is the place where the prototype could use the most work. Either by having a wider range of modifiers to choose from, or having more editable parameters in the available modifiers.

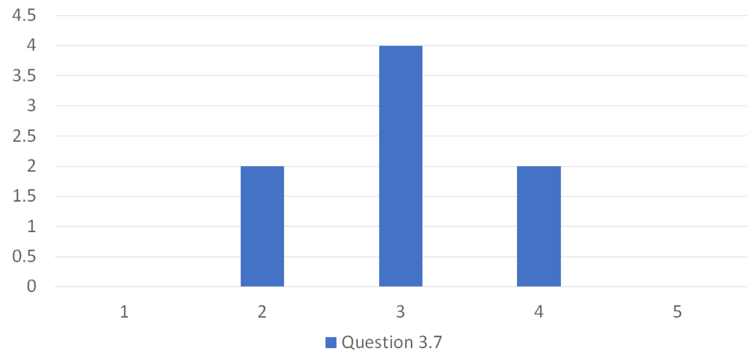


Figure 5.24: Results of questions 7 of section 3 of the questionnaire

This module alone could have almost as much work put into it as the rest of the prototype, but being that it is not the main focus of this thesis, we opted by implementing just the essentials, leaving improvements to this module for future work.

5.4.5 Open Ended Questionnaire

Section 4 of the questionnaire (section B.4), was a more open-ended section where the users could say what they thought about the overall experience with the prototype, if they would be interested in its future, and leave any remarks about improvements that could be done or things that they liked about it. We asked the following questions:

4.1 How useful did you find VRDesigner to be?

4.2 Do you see yourself using it in the future, if it is ever released?

4.3 Why or why not would you use it again in the future?

4.4 What do you think could be improved with the tool?

As can be seen by the results in Figure 5.25, a great majority of the users thought the prototype to be useful and would be interested in its future developments.

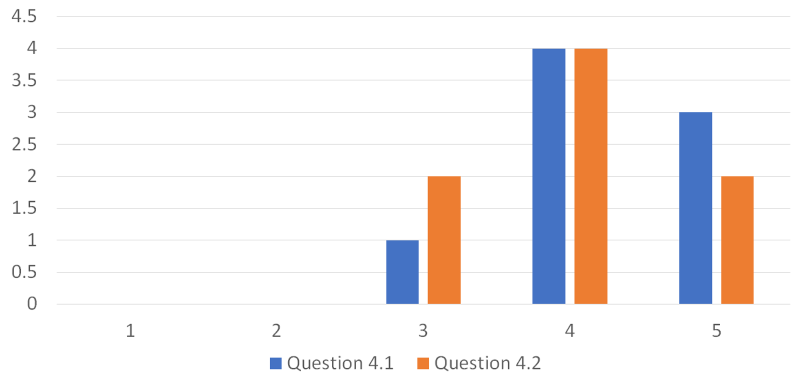


Figure 5.25: Results of questions 1 and 2 of section 4 of the questionnaire

Reasons for this surround the idea of making quick scenes to later polish on other software and the idea of making VR environments directly inside VR, as can be seen in some of the answers from questions 4.3 and 4.4:

- "I could use a tool like this to prototype worlds and make sure environment dimensions are correct for VR."
- "to make vr environments quick"
- "It would be good to make a quick scene and later improve it on other software."

For future work, the features the users thought could most be improved or that were lacking were:

- The access to more objects. This could be done by simply adding access to more online 3D repositories;
- Ability to change textures for the objects. Similarly to the 3D model repositories, we could allow users to access online free texture repositories such as CC0 Textures or Texture Haven, from which they could directly pick textures to use in their scenes.
- Improved speech recognition. While the implemented system is very good at understanding whole sentences or short phrases, single word recognition is a much more difficult task, because the word presents no context. This made it so that occasionally homophones could get detected instead of the intended word (e.g. share instead of chair). An attempt at solving this could be to cross-reference the speech recognition's hypothesis with the 3D repository database in order to disambiguate such cases.

5.5 Conclusions

Overall, we managed to make and test a prototype that attempted to answer the research questions we had established. However, our conclusions need to be taken with the reduced size of the test sample in mind, making us unable to make definitive claims about our results. Despite this, the results we obtained give us indications of the following:

1. How can VR and multimodal interaction approaches be used together to help the process of 3D environment creation?

   Our prototype is an example of a tool that integrates VR, speech control and procedural tools to do just that, as verified by the tests we conducted, and detailed and analysed along this chapter.

2. Can we enable users without specific technical skills to easily and intuitively create VR environments?

   From our testing, the answer to this question seems to be a yes, having achieved this with multiple people consistently. As conjectured, the unique combination of VR and speech controls make for an exceptionally intuitive experience that puts the creative process above the technical expertise.

Chapter 6

Conclusions

With this dissertation we hope to show that there is much that can done for the benefit of VR environment creation. We started by giving some insight into what is virtual reality and how its set of interaction and visualization capabilities makes it unique both in the problems it brings for its content creation and as a tool for tackling them.

After that we explored the world of 3D modelling and explained what are the particular challenges behind environment creation and specifically for doing so for VR. We consulted some 3D artists with experience in modelling environments for VR and gathered that it is a time-consuming process with the tools available today and that a tool with the feature set that we propose would be very welcome in their work. Then we reviewed the existing alternatives in the field and how they compare to what we propose.

Next, we detailed our proposed solution, list its core goals, divide it into building blocks, describe the tools and technologies we expect to use, and explain how we intend to evaluate it.

We followed by summarizing the development process and explaining in detail the different modules that make up our prototype, what features it possesses and walk through how to use it in a typical scenario.

Finally, we explain the testing procedures, we found that our prototype, when tested with our, admittedly small, test group, adequately answered our research questions, being that it: made it faster and more efficient for 3D modelling professionals to develop 3D environments; allowed people without 3D modelling skill and even VR experience to quickly and easily learn to make their own 3D environments. To reach definitive conclusions, a bigger test group would be needed.

For future work, access to online free texture repositories and more procedural tools could be considered. The more feature-complete the system is, the more liberty it would give the user, which in the end, translates to creative potential.

For the testing, as said before, a larger group of test subjects, especially containing subjects with VR experience but no 3D modelling experience, could allow us to extract more information from the tests performed.

In conclusion, we think that our project comes to fulfil a demand that exists for better tools for the creation of VR environments. We expect with it to show that it possible to lower the skill-floor needed for the creation of such environments and to increase its efficiency and streamline the process, accelerating this way the adoption of VR technologies into more domains.

References

[BBGV11] Amartya Banerjee, Jesse Burstyn, Audrey Girouard, and Roel Vertegaal. Pointable. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces - ITS '11*, page 11, New York, New York, USA, 2011. ACM Press.

[BBMP97] Mark Billinghurst, Sisinio Baldis, Lydia Matheson, and Mark Philips. 3D palette. In *Proceedings of the ACM symposium on Virtual reality software and technology - VRST '97*, pages 155–156, New York, New York, USA, 1997. ACM Press.

[BGH02] Doug A. Bowman, Joseph L. Gabbard, and Deborah Hix. A Survey of Usability Evaluation in Virtual Environments: Classification and Comparison of Methods. *Presence: Teleoperators and Virtual Environments*, 11(4):404–424, aug 2002.

[BS10] Cedric Bach and Dominique L. Scapin. Comparing Inspections and User Testing for the Evaluation of Virtual Environments. *International Journal of Human-Computer Interaction*, 26(8):786–824, jul 2010.

[CPK+18] Chien-Wen Chen, Jain-Wei Peng, Chia-Ming Kuo, Min-Chun Hu, and Yuan-Chi Tseng. Ontlus: 3D Content Collaborative Creation via Virtual Reality. In Klaus Schoeffmann, Thanarat H. Chalidabhongse, Chong Wah Ngo, Supavadee Aramvith, Noel E. O'Connor, Yo-Sung Ho, Moncef Gabbouj, and Ahmed Elgammal, editors, *24th International Conference on Multimedia Modeling*, pages 386–389, Bangkok, Thailand, 2018. Springer International Publishing.

[FH00] Gerald E. Farin and Dianne Hansford. *The Essentials of CAGD*. A K Peters/CRC Press, oct 2000.

[FRTT18] Lucas Figueiredo, Eduardo Rodrigues, João Teixeira, and Veronica Teichrieb. A comparative evaluation of direct hand and wand interactions on consumer devices. *Computers & Graphics*, 77:108–121, dec 2018.

[Fur08] Virtual Reality. In Borko Furht, editor, *Encyclopedia of Multimedia*, pages 968–968. Springer US, Boston, MA, 2008.

[FVFH96a] James D Foley, Andries Van Dam, Steven K Feiner, and John F Hughes. Representing Curves and Surfaces. In *Computer Graphics: Principles and Practice*, chapter 11, pages 471–532. Addison-Wesley Professional, 2nd edition, 1996.

[FVFH96b] James D Foley, Andries Van Dam, Steven K Feiner, and John F Hughes. Solid Modeling. In *Computer Graphics: Principles and Practice*, chapter 12, pages 533–562. Addison-Wesley Professional, 2nd edition, 1996.

[Gol06a] John F. Golding. Motion sickness susceptibility. *Autonomic Neuroscience*, 129(1-2):67–76, oct 2006.

[Gol06b] John F. Golding. Predicting individual differences in motion sickness susceptibility by questionnaire. *Personality and Individual Differences*, 41(2):237–248, jul 2006.

[Goo] Google. Blocks - Create 3D models in VR. Available at https://vr.google.com/blocks/.

[Gra] Gravity Sketch Limited. Gravity Sketch. Available at https://www.gravitysketch.com/.

[Hau89] A. G. Hauptmann. Speech and gestures for graphic image manipulation. *ACM SIGCHI Bulletin*, 20(SI):241–245, mar 1989.

[Hei98] Michael Heim. *Virtual Realism*. Oxford University Press, New York, USA, 1998.

[HM93] Alexander G. Hauptmann and Paul McAvinney. Gestures with speech for graphic manipulation. *International Journal of Man-Machine Studies*, 38(2):231–249, feb 1993.

[HNV15] Faizan Haque, Mathieu Nancel, and Daniel Vogel. Myopoint. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems - CHI '15*, pages 3653–3656, New York, New York, USA, 2015. ACM Press.

[HW07] Ken Hinckley and Daniel Wigdor. Input Technologies and Techniques. In Andrew Sears and Julie A. Jacko, editors, *The human-computer interaction handbook: fundamentals, evolving technologies and emerging applications*, chapter 9, pages 151–168. Human Factors and Ergonomics, 2 edition, 2007.

[JGvWK18] Shaun W Jerdan, Mark Grindle, Hugo C van Woerden, and Maged N Kamel Boulos. Head-Mounted Virtual Reality and Mental Health: Critical Review of Current Research. *JMIR serious games*, 6(3):e14, jul 2018.

[KPCC18] Hyun K. Kim, Jaehyun Park, Yeongcheol Choi, and Mungyeong Choe. Virtual reality sickness questionnaire (VRSQ): Motion sickness measurement index in a virtual reality environment. *Applied Ergonomics*, 69, 2018.

[Mas] MasterpieceVR. MasterpieceVR. Available at https://www.masterpiecevr.com/masterpiecevrhttps://store.steampowered.com/app/504650/MasterpieceVR/.

[MBN+02] Brad A. Myers, Rishi Bhatnagar, Jeffrey Nichols, Choon Hong Peck, Dave Kong, Robert Miller, and A. Chris Long. Interacting at a distance: Measuring the Performance of Laser Pointers and Other Devices. In *Proceedings of the SIGCHI conference on Human factors in computing systems Changing our world, changing ourselves - CHI '02*, page 33, New York, New York, USA, 2002. ACM Press.

[MH20] Mohamed H. MOUSA and Mohamed K. HUSSEIN. Surface approximation using GPU-based localized fourier transform. *Journal of King Saud University - Computer and Information Sciences*, apr 2020.

[Mic] Microsoft. Maquette. Available at https://www.maquette.ms/.

[Min95] Mark R Mine. Virtual Environment Interaction Techniques. Technical report, Department of Computer Science University of North Carolina Chapel Hill, NC 27599-3175, 1995.

[Moo19] Mariella Moon. Sony has sold 4.2 million PlayStation VR headsets. Available at https://www.engadget.com/2019/03/26/sony-4-2-million-playstation-vr-headset-sales/, mar 2019.

[MSH04] T.S. Mujber, T. Szecsi, and M.S.J. Hashmi. Virtual reality applications in manufacturing process simulation. *Journal of Materials Processing Technology*, 155-156:1834–1838, nov 2004.

[MSPM10] Ioannis Messinis, Dimitrios Saltaouras, Panayotis Pintelas, and Tassos Mikropoulos. Investigation of the Relation Between Interaction and Sense of Presence in Educational Virtual Environments. In *2010 International Conference on e-Education, e-Business, e-Management and e-Learning*, pages 428–431, Sanya, China, 2010. IEEE.

[MV18] Fabrice Matulic and Daniel Vogel. Multiray. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*, pages 1–13, New York, New York, USA, 2018. ACM Press.

[Ocu] Oculus. Medium. Available at https://www.oculus.com/medium/.

[Pol] Poly API | Google Developers.

[Saw18] Abhiskek Sawant. Virtual Reality (VR) in Gaming Market 2018 Leaders: Electronic Arts, Google, HTC, Oculus VR, Kaneva, Samsung, Sony, Leap-Motion - Reuters. Available at https://www.reuters.com/brandfeatures/venture-capital/article?id=32714, 2018.

[SHCC18] Yu Shu, Yen-Zhang Huang, Shu-Hsuan Chang, and Mu-Yen Chen. Do virtual reality head-mounted displays make a difference? A comparison of presence and self-efficacy between head-mounted displays and desktop computer-facilitated virtual environments. *Virtual Reality*, pages 1–10, nov 2018.

[SMR+03] Kay M. Stanney, Mansooreh Mollaghasemi, Leah Reeves, Robert Breaux, and David A. Graeber. Usability engineering of virtual environments (VEs): identifying multiple criteria that drive effective VE system design. *International Journal of Human-Computer Studies*, 58(4):447–481, apr 2003.

[SSFFR19] Mónica Sanchez-Sepulveda, David Fonseca, Jordi Franquesa, and Ernesto Redondo. Virtual interactive innovations applied for digital urban transformations. Mixed approach. *Future Generation Computer Systems*, 91:371–381, feb 2019.

[sta18] VR device shipments by vendor worldwide 2017-2019. Available at https://www.statista.com/statistics/671403/, dec 2018.

[Ste] SteamVR - Valve Corporation.

[SvdSKvdM01] Martijn J. Schuemie, Peter van der Straaten, Merel Krijn, and Charles A.P.G. van der Mast. Research on Presence in Virtual Reality: A Survey. *CyberPsychology & Behavior*, 4(2):183–201, apr 2001.

[SZP+00] R. Sharma, M. Zeller, V.I. Pavlovic, T.S. Huang, Z. Lo, S. Chu, Y. Zhao, J.C. Phillips, and K. Schulten. Speech/gesture interface to a visual-computing environment. *IEEE Computer Graphics and Applications*, 20(2):29–37, 2000.

[Um18a] Dugan Um. Parametric Line and Curve Theory. In *Solid Modeling and Applications*, chapter 4, pages 93–146. Springer International Publishing, Corpus Christi, TX, USA ISBN, 2 edition, 2018.

[Um18b] Dugan Um. Rendering Theory. In *Solid Modeling and Applications*, chapter 6, pages 173–192. Springer International Publishing, Corpus Christi, TX, USA ISBN, 2 edition, 2018.

[Unb] Unbound Technologies. Unbound Alpha. Available at http://unbound.io/.

[Uni] Unity Real - Time Development Platform | 3D, 2D VR & AR Engine.

[Vir15] Virtual Reality Society. History Of Virtual Reality - Virtual Reality Society. Available at https://www.vrs.org.uk/virtual-reality/history.html, 2015.

[WWW+18] Peng Wang, Peng Wu, Jun Wang, Hung-Lin Chi, Xiangyu Wang, Peng Wang, Peng Wu, Jun Wang, Hung-Lin Chi, and Xiangyu Wang. A Critical Review of the Use of Virtual Reality in Construction Engineering Education and Training. *International Journal of Environmental Research and Public Health*, 15(6):1204, jun 2018.

[YFZ+18] Difeng Yu, Kaixuan Fan, Heng Zhang, Diego Monteiro, Wenge Xu, and Hai Ning Liang. PizzaText: Text entry for virtual reality systems using dual thumbsticks. *IEEE Transactions on Visualization and Computer Graphics*, 24(11):2927–2935, nov 2018.

[YZWZ19] Mengli Yu, Ronggang Zhou, Huiwen Wang, and Weihua Zhao. An evaluation for VR glasses system user experience: The influence factors of interactive operation and motion sickness. *Applied Ergonomics*, 74:206–213, jan 2019.

Appendix A

Test results

A.1 Checklist

	101	102	103	104	105	201	202	203
Classroom								
Teacher's table	✓	✓	✓	✓	✓	✓	✓	✓
Teacher's chair	✓	✓	✓	✓	✓	✓	✓	✓
Blackboard	✓	✓	✓	✓	✓	✓	✓	✓
Bookshelf	✓	✓	✓	✓	✓	✓	✓	✓
Clock	✓	✓	✓	✓	✓	✓	✓	✓
Door	✓	✓	✓	✓	✓	✓	✓	✓
4x Chairs	✓	✓	✓	✓	✓	✓	✓	✓
9x Lockers	✓	✓	✓	✓	✓	✓	✓	✓
Museum								
2x walls	✓	✓	✓	✓	✓	✓	✓	✓
3x paintings	✓	✓	✓	✓	✓	✓	✓	✓
Bench	✓	✓	✓	✓	✓	✓	✓	✓
Camera	✓	✓	✓	✓	✓	✓	✓	✓
Sculpture	✓	✓	✓	✓	✓	✓	✓	✓
Fire extinguisher	✓	✓	✓	✓	✓	✓	✓	✓
Bedroom								
Bed	✓	✓	✓	✓	✓	✓	✓	✓
Book shelf	✓	✓	✓	✓	✓	✓	✓	✓
Desk	✓	✓	✓	✓	✓	✓	✓	✓
Lamp	✓	✓	✓	✓	✓	✓	✓	✓
Trashcan	✓	✓	✓	✓	✓	✓	✓	✓
Floor mat	✓	✓	✓	✓	✓	✓	✓	✓
Window	✓	✓	✓	✓	✓	✓	✓	✓
Toy	✓	✓	✓	✓	✓	✓	✓	✓

Table A.1: Checklist for the realisation of the scene creation test.

A.2 Results

A.2.1 Simple tasks

	Task id	C1	C2	101	102	103	104	105	201	202	203
Simple Tasks	STtotal	5:16	5:46	7:06	5:57	4:50	4:45	4:53	9:20	10:21	9:30
Create walls for a room	ST1	0:35	0:29	1:52	1:31	1:10	1:03	0:58	1:51	2:20	2:13
Create a table using primitives	ST2	1:23	2:00	3:53	2:58	2:16	2:35	2:35	5:12	6:01	4:44
Place an imported chair by the table	ST3	2:35	2:52	0:41	0:35	0:33	0:24	0:30	0:32	0:23	0:41
Place 5 chairs on one side of the table using modifiers	ST4	0:43	0:25	0:40	0:53	0:51	0:43	0:50	1:45	1:37	1:52

Table A.2: Time measurements of the simple tasks

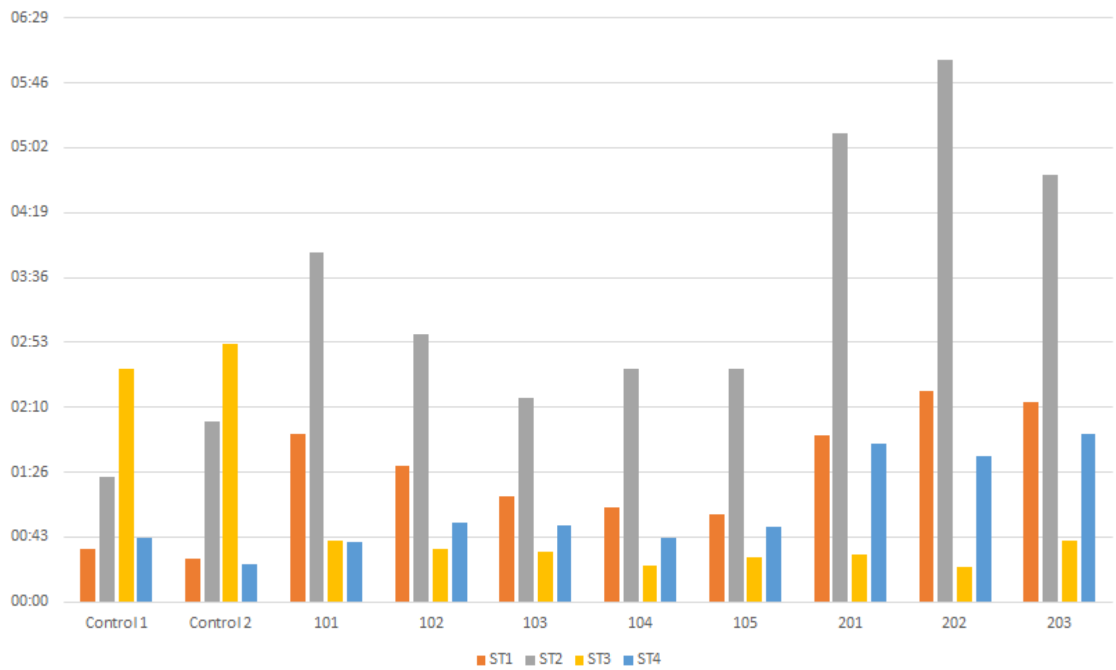


Figure A.1: Graph of the times taken for each task

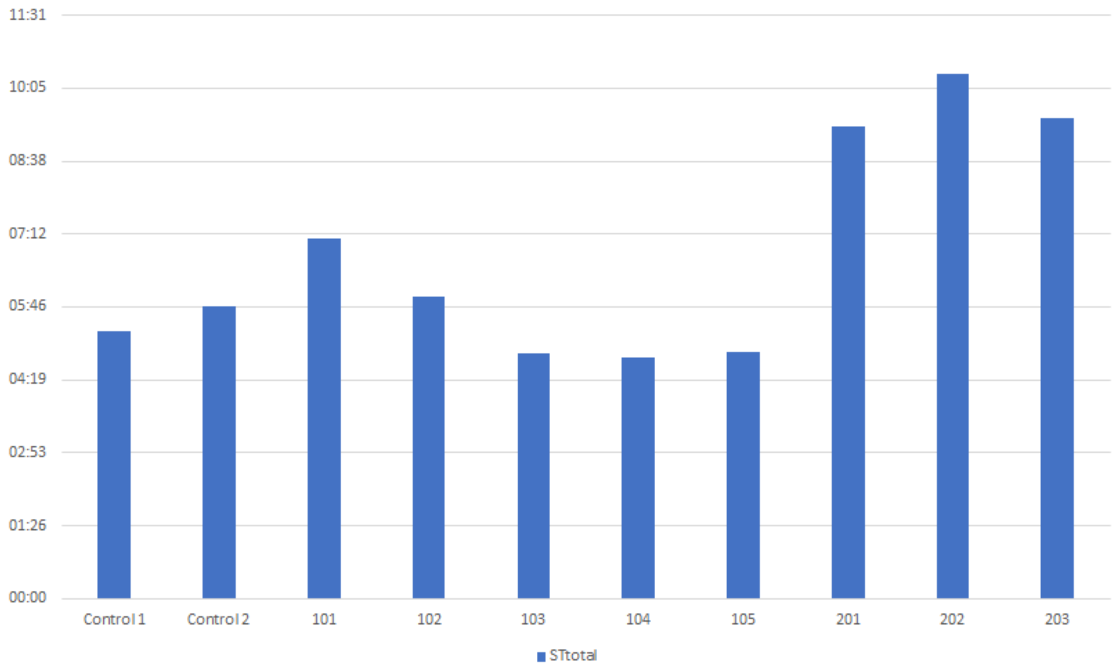


Figure A.2: Graph of the total time taken for the simple tasks

A.2.2 Environment creation

	Task id	Control	101	102	103	104	105	201	202	203
Modelling Environments	ECtotal	1:37:40	1:00:33	54:06	48:19	53:48	48:29	1:18:45	1:23:34	1:25:41
Classroom scene	EC1	36:35	22:13	24:36	19:57	18:13	17:39	25:10	38:25	36:00
Museum scene	EC2	23:52	15:01	08:15	12:23	14:51	08:35	19:12	15:36	26:12
Bedroom scene	EC3	37:13	18:19	21:15	15:59	20:44	22:15	34:23	29:33	23:29
Scene order		N/A	1-2-3	1-3-2	2-1-3	2-3-1	3-1-2	3-2-1	1-3-2	2-1-3

Table A.3: Time measurements of the creation of environments.

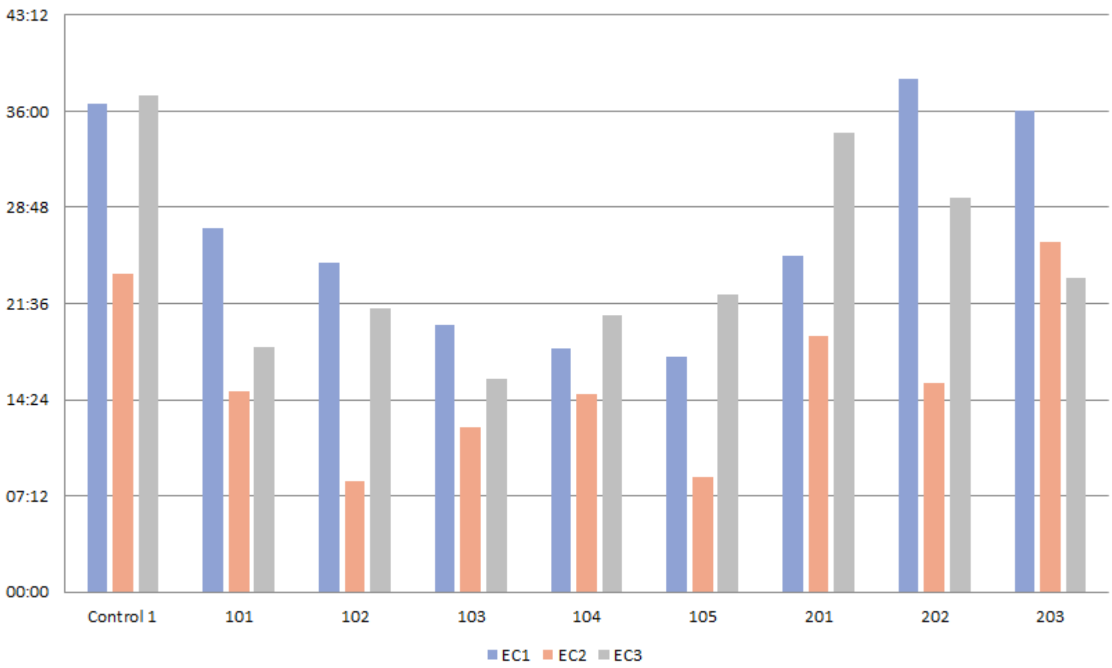


Figure A.3: Graph of the times taken to create each environment.

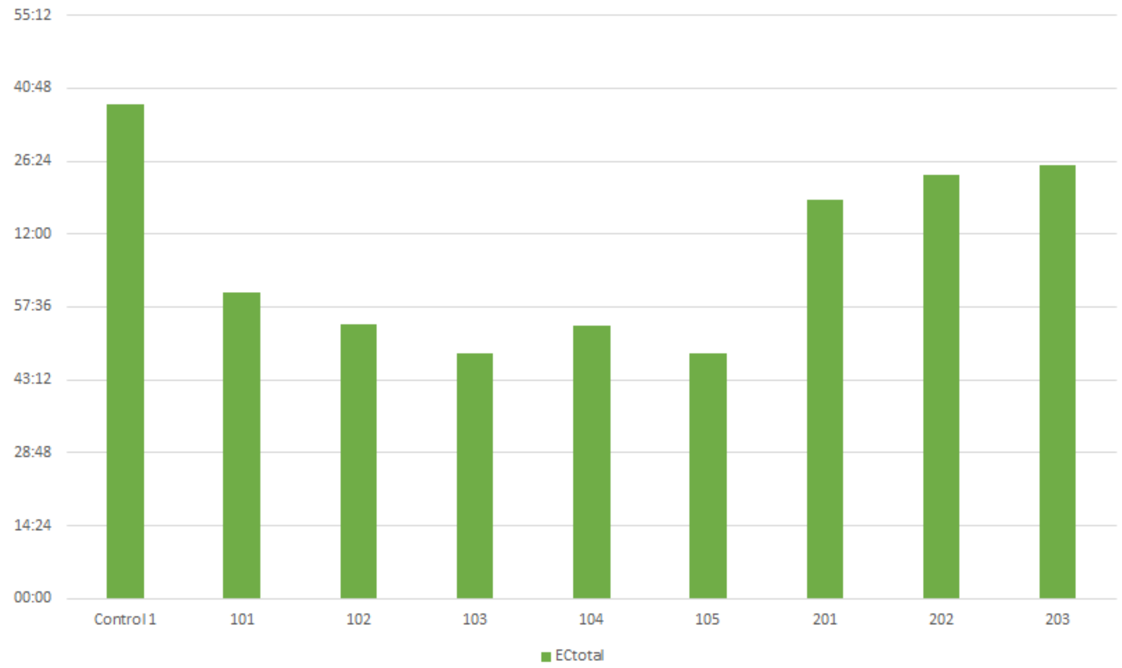


Figure A.4: Graph of the total time taken for creating the environments.

Appendix B

Questionnaire

B.1 Declaration of consent

First of all, we want to make sure you understand and consent to the gathering of information from your tests.

All information gathered during and after this test will be solely used for the development of a thesis dissertation and will be done so confidentially. No personal information will be gathered besides your answers to this questionnaire.

Virtual reality is known to cause nausea in people not used to it. Feel free to pause your test and rest at any point. You are also allowed to withdraw from the test at any stage, as well as opt not to have your test data collected.

You are free to ask questions about any part of the study you may feel need further clarification or information. By choosing "I consent", you agree to voluntarily participate in this study and acknowledge all your rights during it. You also confirm you feel sufficiently informed about it and acknowledge the fact that excerpts from your answers may be quoted in the said dissertation.

○ I consent

○ I do not consent

B.2 About you

This will let us know about your previous experience with 3D modelling and VR.

1. **Introduce here the id you were given.**

   ____

2. **How much experience with 3D modelling and environment creation do you have?**

No experience at all	1	2	3	4	5	A lot of experience

3. **Have you done any professional 3D modelling before?**

   ○ Yes

   ○ No

4. **How much experience do you have with Virtual Reality?**

No experience at all	1	2	3	4	5	A lot of experience

5. **How often have you used voice controls in your life?**

Never	1	2	3	4	5	Quite regularly

B.3 About your experience with the tool

Tell us how useful and easy to use you found the tool you just tested to be.

1. **How easy did you find it was to interact with the objects around you?**

Very Hard	1	2	3	4	5	Very Easy

2. **How easy did you find it was to manipulate the objects around you?**

Very Hard	1	2	3	4	5	Very Easy

3. **How easy did you find it was to choose the direction you wanted to manipulate an object?**

Very Hard	1	2	3	4	5	Very Easy

4. **How useful did you find the direct access to free 3D repositories to be?**

Not Useful	1	2	3	4	5	Very Useful

5. **How easy did you find it was to import objects?**

Very Hard	1	2	3	4	5	Very Easy

6. **How easy did you find it was to search for objects using your voice?**

Very Hard	1	2	3	4	5	Very Easy

7. **How useful did you find the object modifiers to be?**

Not Useful	1	2	3	4	5	Very Useful

B.4 Ending thoughts

1. **How useful did you find VRDesigner to be?**

Not Useful	1	2	3	4	5	Very Useful

2. **Do you see yourself using it in the future, if it is ever released?**

Not at All	1	2	3	4	5	Very Likely

3. **Why or why not would you use it again in the future?**
   How do you feel it could help you?

   ___

   ___

4. **What do you think could be improved with the tool?**
   What would you like to see changed or added? Where does it excels and where is it lacking? Add any suggestions you might have.

   ___

   ___

   ___

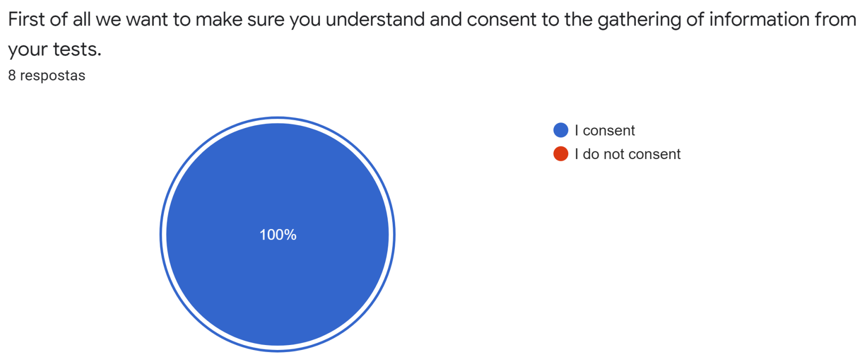
B.5 Thank you for your collaboration

Please hit submit if you are satisfied with your answers. If not, you can go back and do any modifications you'd like.

Appendix C

Questionnaire results

C.1 Declaration of consent



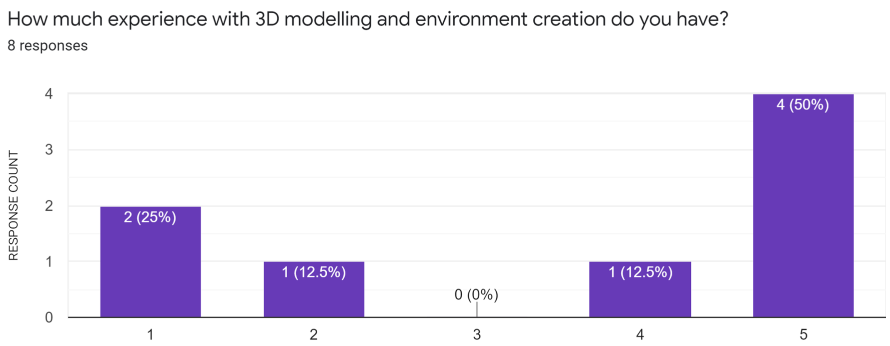
1.

Figure C.1: Section 1, Question 1

C.2 About you

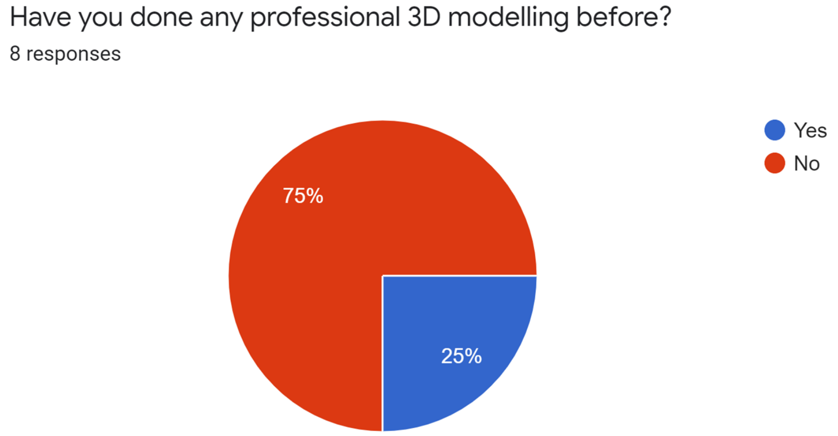
1. **Introduce here the id you were given.**

    - 101
    - 102
    - 103
    - 104
    - 105
    - 201
    - 202
    - 203



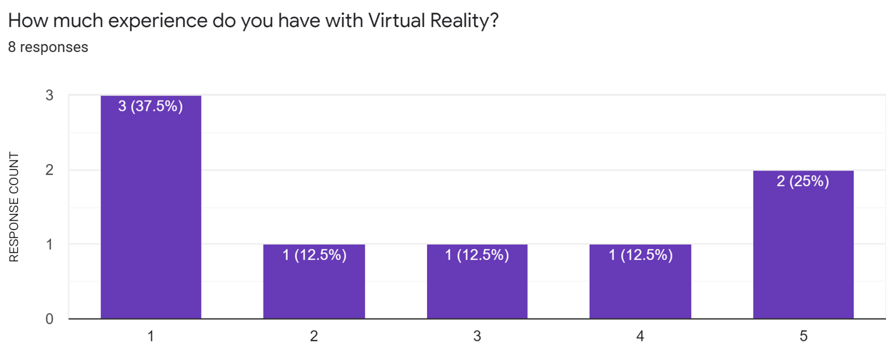
2.

Figure C.2: Section 2, Question 2



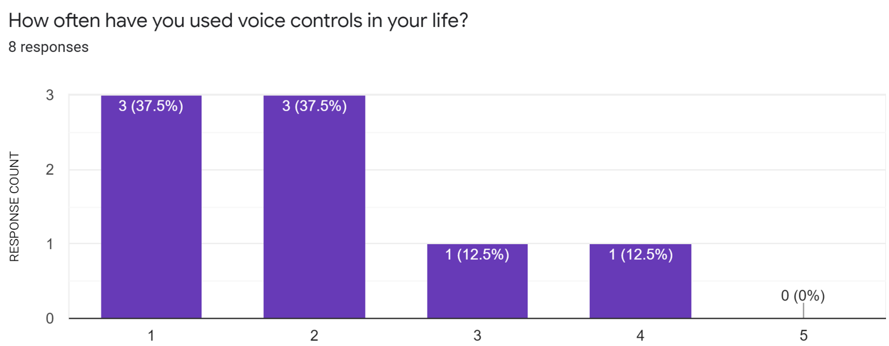
3.

Figure C.3: Section 2, Question 3



4.

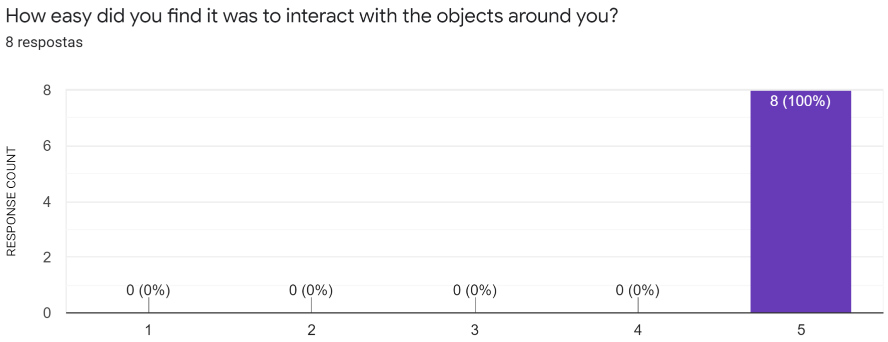
Figure C.4: Section 2, Question 4



5.

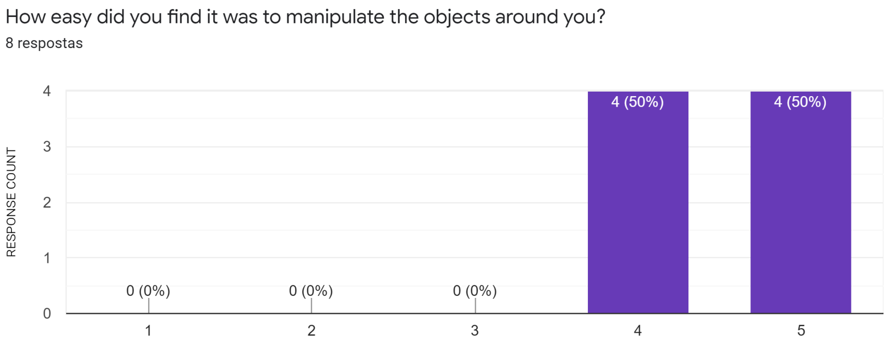
Figure C.5: Section 2, Question 5

C.3 About your experience with the tool



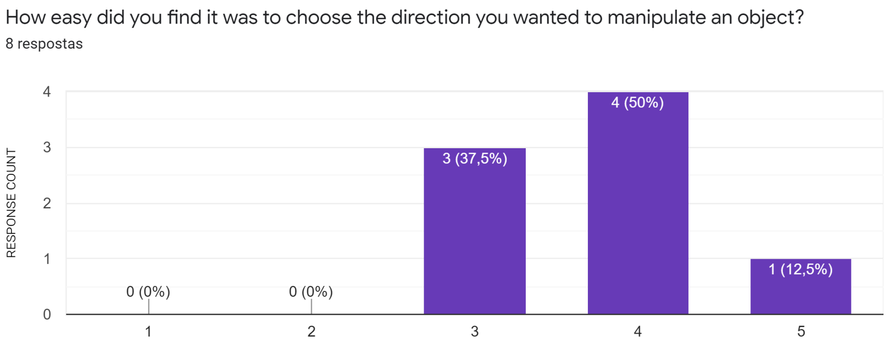
1.

Figure C.6: Section 3, Question 1



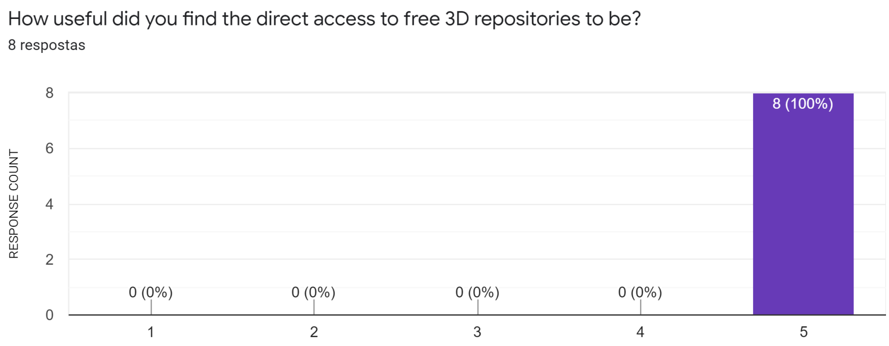
2.

Figure C.7: Section 3, Question 2



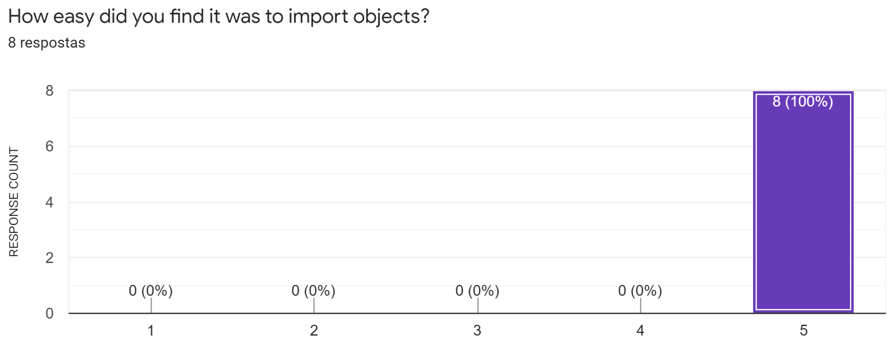
3.

Figure C.8: Section 3, Question 3



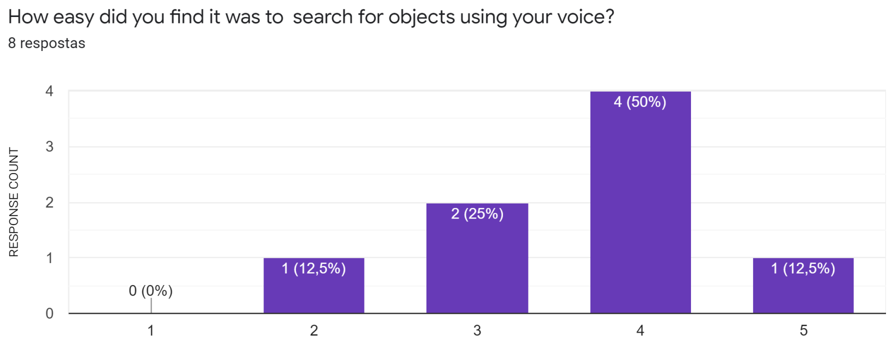
4.

Figure C.9: Section 3, Question 4



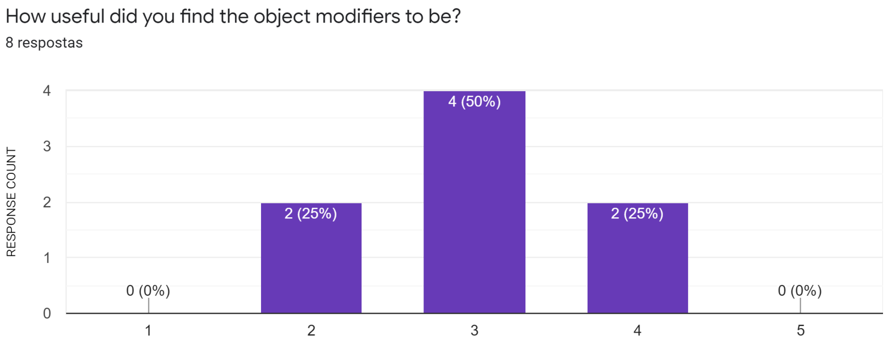
5.

Figure C.10: Section 3, Question 5



6.

Figure C.11: Section 3, Question 6



7.

Figure C.12: Section 3, Question 7

C.4 Ending thoughts

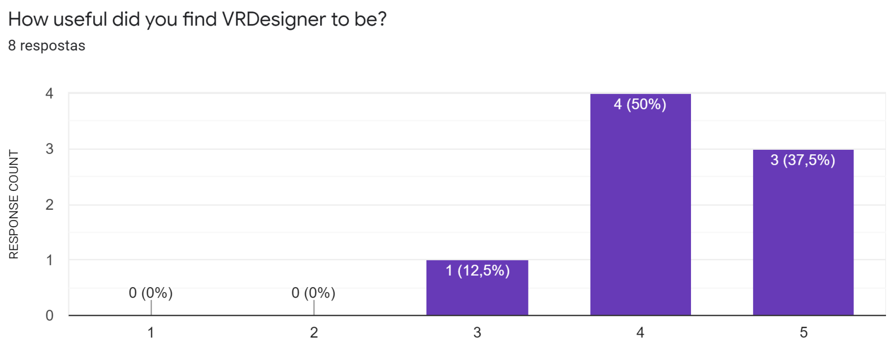
1. 

Figure C.13: Section 4, Question 1

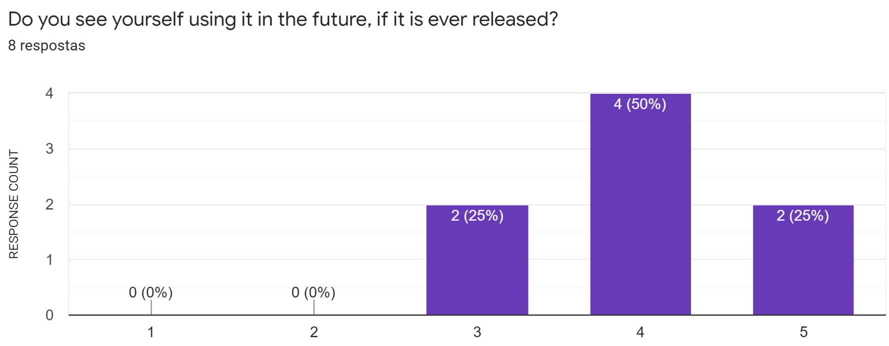
2. 

Figure C.14: Section 4, Question 2

3. **Why or why not would you use it again in the future?**

    - It would be good to make a quick scene and later improve it on other software.
    - to make vr environments quick
    - I could use a tool like this to prototype worlds and make sure environment dimensions are correct for VR.
    - It could be useful to make quick things but the fact that it's VR makes it a bit more cumbersome to set up. Maybe when VR is more mainstream/more convenient to use.
    - I think despite being faster to get results than a conventional method, it is more tiring physically.
    - Seems like a fun tool to use and is very easy to learn.
    - If I ever wanted to create 3D environments. It is very easy to use.
    - Maybe if I get a VR system for myself.

4. **What do you think could be improved with the tool?**

    - The voice controls were hard to use. Would be good to be able to change textures for the primitives. More modifiers would also be good. I like the way the menus and windows worked. Very intuitive.
    - better results when searching for assets. i like how fast it is.
    - Voice controls could be improved. More modifiers would be good. Access to more assets.
    - Add ways to change atmospheric settings. Also add ways to change textures.
    - More options.
    - It is very good. In my opinion this app could have a very promising future.