# Immersive Multimodal and Procedurally-Assisted Creation of VR Environments

João Ferreira*
Faculdade de Engenharia,
Universidade do Porto

Daniel Mendes†
INESC TEC,
Faculdade de Engenharia,
Universidade do Porto

Rui Nóbrega‡
NOVA-LINCS,
Faculdade de Ciências e Tecnologia,
Universidade Nova de Lisboa

Rui Rodrigues§
INESC TEC,
Faculdade de Engenharia,
Universidade do Porto

**ABSTRACT**

We present VR Designer, a tool for expediting the creation 3D scenes inside VR. It uses controllers and voice commands to create and manipulate primitives and objects imported from openly available repositories. We use modifiers to accelerate repetitive tasks, resorting to procedural content creation techniques to automate the workflow. The tool allows non-expert users to quickly create scenes for contexts such as training or education. We also conducted a user study to validate VR Designer.

**Index Terms:** Computing methodologies—Computer graphics—Graphics systems and interfaces—Virtual reality; Human-centered computing—Human computer interaction (HCI)—Interaction techniques

## 1 INTRODUCTION

The increasing number of virtual reality (VR) applications has created an expedite need for more custom immersive scenes. This is important in areas were the need for new and updated content is high, such as in the training and education areas.

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 task. One of the largest components of this is the creation of VR environments which itself presents numerous challenges.

VR environment creation is commonly 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 authoring tools for the development of VR environments, especially in the case of users without specialized skills in the area.

In this paper we propose VR Designer, a VR authoring tool to create effective and personalized immersive 3D virtual content. The main advantages of the proposed tool are that (1) the entire scene modelling is executed inside VR, (2) makes use of voice commands and VR device affordances, (3) allows the import of objects from known available repositories, and (4) provides a set of scene Modifiers that procedurally allow the automation of repetitive tasks and multiple instancing.

The main advantage of this approach is that the authoring workflow is accelerated and can be used by a non-expert user to quickly create an immersive scene that can be used in custom scenarios such as training or education. Expert users can also use the tool to create fast initial drafts of a scene before fine-tuning it in a dedicated modeling tool.

In the next section we describe the state of the art and other projects related with this topic. The features and proposed techniques of VR Designer are described in section 3. In the end we present a user study conducted to validate the tool and its features.

*e-mail: joao.martins.ferreira@fe.up.pt
†e-mail: danielmendes@fe.up.pt
‡e-mail: rui.nobrega@fct.unl.pt
§e-mail: rui.rodrigues@fe.up.pt

## 2 RELATED WORK

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.

### 2.1 3D Representation

3D objects can be represented internally in different ways, each with their own advantages and disadvantages, being even possible to combine multiple representations to achieve more complex results. Some representations lend themselves especially well to some 3D modelling paradigms and techniques.

Some of the more common representations are: *Polygons*, 3D points connected to create the faces of objects; *Curves*, weighted control points that, most commonly NURBS [16], splines and Bézier surfaces; *Implicit Surfaces* [2], surfaces defined by mathematical formulas; *Constructive Solid Geometry (CSG)*, defining objects as compositions of primitive objects, using boolean operations; *Voxels*, the division of 3D space into a uniform grid, alike the pixels in 2D rasterized media; *Displacement Surfaces*, storing geometric information in a 2D texture, usually as a height or vector map.

Since we are targeting non-expert users, we will not allow for the control of complex elements and operations on objects' shapes. Instead, we will resort to the instantiation of existing objects, represented using polygon meshes, which users will be able to manipulate to create complete environments.

### 2.2 3D Modelling Techniques

Several techniques or abstractions exist when trying to model in 3D. Knowing them is useful for quickly understanding the general workflow of an application and what techniques are more suited for specific uses. Some of the more common modelling techniques are:

*Polygonal modelling* [17] consists in modelling objects by directly affecting the vertices in a polygonal 3D object; *Digital sculpting*, consists in modelling, using brushes to either add or remove material akin to real life sculpting; *3D scanning and reconstruction* consists in using techniques like photogrammetry [9, 13] to gather 3D information about one or multiple objects and replicating them in a digital

978-0-7381-1367-8/21/$31.00 ©2021 IEEE
DOI 10.1109/VRW52623.2021.00012

format; *Procedural modelling* consists in modelling through the use of algorithms, sets of rules or even mathematical formulas; *Kitbash modelling* consists in producing new models by combining other, usually simpler, models together; *Parametric Modelling* [7] consists in describing the models as a sum of operations, and because of that, their parameters are able to be changed at any time to get different results.

For our approach, a mixture of *kitbashing* and *procedural modelling* will be at the core of the workflow. *Kitbashing* will be reflected in the placement and grouping of existing models and primitives, and *procedural modelling* through the use of sets of modifiers to automate complex or monotonous tasks.

### 2.3 3D modelling in VR

There are a number of previous works that explore 3D modelling in VR, using different techniques. For instance, the seminal work by Billinghurst et al., 3D Palette [1] lets users model 3D environments inside VR using multimodal interactions, such as digitizing tablets. It is based on kitbash and polygonal modelling, and shows that the search for better modelling paradigms, not based on 2D screens, exist for quite a while now. Make VR [8] allows users to create a 3D scene in VR using two 6DOF controllers. Users can place primitive shapes or more complex objects, and modify them using CAD operations like booleans, sweeps, cuts and deformations. DIY World Builder [19] is an immersive world creation tool where users are able to create 3D environments in VR. To achieve this, it uses a 6DOF controller and a mobile phone's touchscreen to select models from a list, place and manipulate them.

Lift-Off [6] is an example of an approach that uses curves to model objects. It uses bimanual spatial input to make 3D models from free-hand sketches. The sketched curves are swept into surfaces that make-up the final objects. Mendes et al. [11] proposed a set of approaches to tackle CSG based modeling in VR. They investigated the use of gesture- and menu-based controls for such task. Chen et al. [3] developed Ontlus, which lets users create and manipulate 3D objects in VR. Their approach focuses on the creation of objects based on sculpting instead of environments, and supports synchronous collaboration.

Some commercial solutions have also been proposed. Blocks [4] is an application that lets users model 3D objects in VR. It is based on "paint" modelling and polygonal modelling techniques. Maquette [12] 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 users to export their creations. For text input it relies on a virtual keyboard. Medium [14] also allows users to create 3D models in VR with sculpting or kitbashing modelling techniques. It too has a library of objects ready to use and allows users to export the models to common 3D object formats. MasterpieceVR [10] offers more sculpting tools and possibilities than usual, and also has kitbashing capabilities. It lets users import images into the VR environment to use as references and to export created models to common 3D formats. It also has collaboration support, allowing multiple users to work in the same scene. Unbound [18] is another solution focused on sculpting modelling of 3D objects, and also supports collaboration. Gravity Sketch [5] approaches modelling in VR differently. Instead of the usual sculpting modelling, it is prominently based on curve modelling. It uses Bézier and spline surfaces, as well as bevelled curves to produce smooth modelling results. It also allows importing of images to use as references, and exporting models into common 3D formats.

In out approach, we focus on kitbashing by providing support to free online repositories of 3D models. However, instead of using a keyboard in VR, we support voice input to perform the queries, as suggested by Henriques et al. [15]. Additionally, to expedite the creation of environments, we allow users to specify sets of simple rules to procedurally generate the content.

## 3 VR Designer

Our approach, named VR Designer, is one that harnesses the interaction and visualization capabilities of VR technology to allow the user to use a 3D immersive interface to design a 3D virtual environment in an intuitive way and with reduced effort. Its main goal is to provide easy tools to populate an immersive environment resorting to: asset repositories, natural interfaces for object selection and manipulation, and rule-based procedural content generation to replicate, mutate and distribute elements on the scene.

The main interaction devices used in VR Designer are an HMD and a set of controllers (currently a HTC Vive headset and wands, although these can be mapped to others) and a microphone. For rendering, overall device management, VR Designer uses the Unity 3D game engine and Steam VR. In terms of natural interfaces, part is based on speech recognition, for which Windows Speech Recognition (WSR) is used.

Additionally, VR Designer provides a back-end to access 3D repositories, which in its current iteration uses the *Poly API*[1], 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 for specific models, retrieve thumbnails, and download models and view their information.

The following sections describe the various functionalities provided by VR Designer.

### 3.1 Adding Objects

There are three different methods of adding objects to the environment. Each of them can be used to achieve slightly different results.

#### 3.1.1 Instancing Primitives

With this feature, the user is able to place basic geometric shapes in the world by selecting from a list of five options: cube, sphere, cylinder, plane and pyramid. These can later be combined to create more complex objects. These options appear in the form of a radial menu, as seen in Fig. 1 (middle). After selecting the desired primitive, it will appear in front of the user's field of view. At this point, the user can interact with the object and manipulate it at will. This can be seen in Fig. 1 (right).

#### 3.1.2 Import Local Objects

This method allows the user to import models from local folders on their computer. Currently, only the *OBJ* file format is supported and the file must be placed in a specific folder.

#### 3.1.3 Search Online for Objects to Import

The most relevant method to add objects to the virtual world is to directly import them from free online repositories, by allowing the user to query the repository by voice, and to select from the query


Figure 1: Main radial menu, primitive selection sub-menu and object after selection

[1]Poly: https://poly.google.com/


Figure 2: Using speech to search for objects online

results which object to import. Currently, we have only integrated our prototype with Google's Poly repository, but more can be added in the future.

After selecting the appropriate option in the *Add Object* radial submenu, the speech recognition system will activate and start listening to the user'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 (Fig. 2). The user can now speak out loud what they want to be searched (e.g. "furniture"), and the speech recognition system will translate it into text. The query is then forwarded to the repository, and the results of the search will appear in a window in front of the user.

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 Fig. 3 (left).

At this point users can choose which of the objects they want to import. 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 Fig. 3 (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. Nevertheless, the user can then transform the object if necessary.

### 3.2 Selection and Manipulation

Objects in VR Designer can be selected in two different ways: by grabbing or by using a laser.

#### 3.2.1 Grabbing

To grab an object, the user must first move the hand close to it. If the object is available to be grabbed, it will be highlighted, and the user will have to press the "grab" button in the wand to pick the object. At that point, the hand and highlight disappear (to make it easier to see the object and surroundings when manipulating it), and the object will follow the hand movements until the "grab" button is released (Fig. 4, top).


Figure 3: Using the search online window

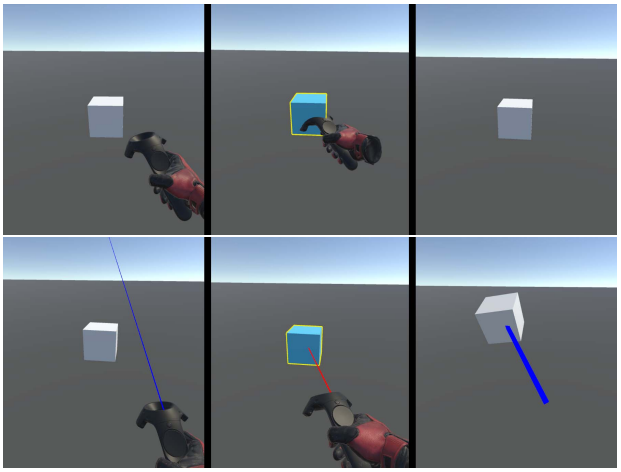

Figure 4: Selecting objects. Grabbing (top): Approach-Hover-Grab; Laser (bottom): Enable-Point-Attach

#### 3.2.2 Laser

If the "grab" button is slightly pressed without any object close to the hand, a blue beam "laser" is activated, and can then be pointed to an object. If the object is grabbable, it will be highlighted and the laser will turn red. By fully pressing the "grab" button, the object will be picked and become attached to the laser, which turns blue and thicker (Fig. 4, bottom). It will then follow the laser according to the user's hand motion. This method has the advantage of being able to manipulate objects at a much longer distance, but it is not very precise and does not allow much control over the rotation of the object. Additionally, the user is able to move the object close or further away along the laser, by performing scroll motions in the trackpad of the controller, where scrolling up will make the object go away from the user and scrolling down will make it come closer.

### 3.3 Editing Individual Properties

Modifying individual object properties is useful for fine-grained adjustments (for example, rotating an object one degree in the $Z$ axis), for controlled modifications and final polishing of the environment. These adjustments can be made through the use of a *Properties Window*, which appears above the object and facing the user after an object is selected (Fig. 5).

The first area of the window allows the user to change the object's position, rotation and scale, by using the controller trackpad as a scroll wheel to increase and decrease each individual component. The type toggle is used to make objects static or dynamic (reactive to other elements, such as collisions and gravity). The button on the bottom left 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. This ability allows users to move or edit objects together, add a modifier to a collection of objects, and build more complex objects out of simple ones. Lastly, the *Edit Modifiers* button allows the configuration of modifiers to be applied to the object/group (see 3.4).

### 3.4 Applying Modifiers

Modifiers are tools the user can employ to turn tedious or specialized tasks into something that can easily be applied to any object. The *Modifiers Window* (Fig. 6, accessible through the "Edit Modifiers" button of the properties window) is used to edit, add, and remove the modifiers applied to the selected object. Currently, VR Designer

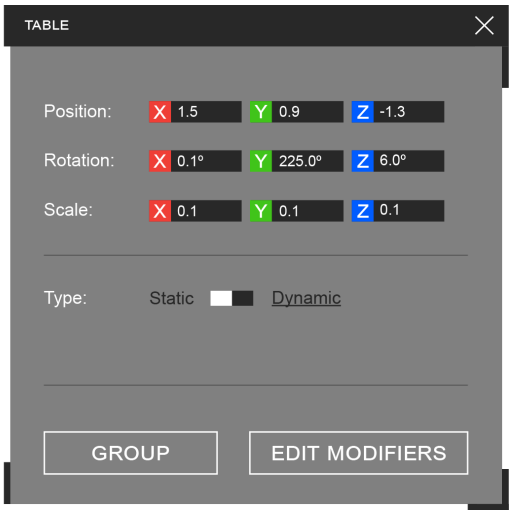


Figure 5: The properties window

supports two modifiers, *Array* and *Randomize*, and we intend to add more in the future.

### 3.4.1 Array Modifier

An *Array* modifier allows the user to replicate an object multiple times (the *Count* property) with controlled spacing between them (the *Offset* property). 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, using two array modifiers could create a grid with $N \times M$ instances of the intended object. Fig. 7 exemplifies the creation of a row of chairs.

### 3.4.2 Randomize Modifier

The *Randomize* modifier allows to randomize placement properties of the object. If applied after an *Array* modifier, it will affect each of the replica objects differently. The three vectors available allow

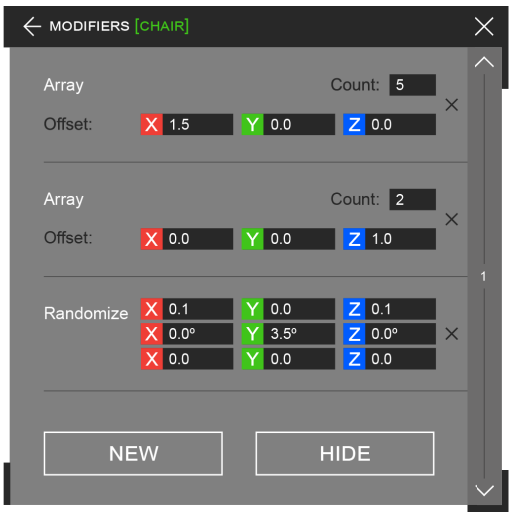


Figure 6: The modifiers window

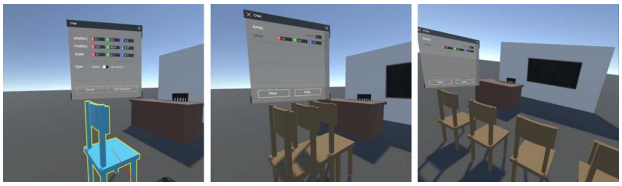

Figure 7: Applying an array modifier to an object

users to specify the maximum amount the object can be affected by this modifier regarding position, rotation and scale, respectively. For example, if the modifier has a value of 1.0 in the position's *X* axis, then it will randomize the object's position in said axis between +1.0 and -1.0 in relation to its current position.

## 3.5 Example Workflow

To provide a practical workflow example, let us consider that a user would like to create a simple classroom scene, with four chairs in a line, the teacher's desk and chair, and a chalkboard on a wall.

The first step is to import the necessary objects. To do this, the user opens the *Radial Menu*, and goes to the *Add Object* option, followed by the *Search Online* option. Doing this starts the *Speech Recognizer* and the user can say what object they wanted to search for.

At that 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 matched their vision, at which point, by clicking on it, will be imported into the world. After that, 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.

The next step is to place the objects in the correct position. For this, the user can go 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 can quickly yield a rough first build of the scene.

Figure 8: The complete example scene.

To place a wall behind the chalkboard, the user can create a plane through the primitives menus. To do this, the user opens 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 left is to scale it to match the size of a wall and place it behind the chalkboard.

Now, the classroom is almost complete, except for some chairs for the students. To populate the room, the user can go into *Select Mode* and select the chair. This opens the *Properties Window* where the user can then click to open the *Modifiers Window*. In this window the user adds an array modifier with a count of 4 and an offset in the negative *Z* direction. The completed scene can be seen in Fig. 8.

## 4 User Evaluation

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.

To validate our work, we conducted a user evaluation. Our main objectives were: (1) to understand how VR and multi-modal interaction can help the field of 3D environment creation, namely how it compares to existent alternatives both in ease of use and in speed; (2) to find out if, using the voice and gestures, it is possible for users without technical skills to quickly learn and be capable of creating their own 3D environments.

For this, we carried out two main tasks, one focused on individual features (Simple Task), and the other in the overall workflow (Environments Creation). We included general users and 3D modelling professionals to assess the differences in interaction and outcomes.

### 4.1 Simple Task

To put the developed tool to the test on the objectives established, we created a simple task with four steps (Fig. 9), each consisting of simple interactions that required the use of the different features of the prototype.

(1) The first step consisted of creating four walls of a room. This required the use of primitives, interacting with objects and manipulating them. (2) The second step was to create a table suitable for 10 seats, manipulating and grouping five cube primitives. This step tested the ability to do kitbash modelling to create complex objects out of simpler ones, using our prototype. (3) The third step was to import a chair from an online 3D repository using voice controls, and manipulate it to place it by the table. (4) The fourth step was to use the modifiers' feature of the prototype to automate the replication and placement of five chairs on one side of the table.

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 step 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, and combined with a questionnaire, we could gain some insight into the usability and intuitiveness of the different systems.

### 4.2 Environments Creation

After testing individual features, the prototype was tested as a whole. For this, participants were asked to replicate a list of scenarios according to some established rules. Once again they were timed, and compared to a control group. The scenes were given in a different order for each user, preventing order bias. 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.

EC1 **Museum:** This was the simplest scene of the three. The required elements were: two walls, three paintings distributed by the walls, one bench, a security camera, a sculpture, a fire extinguisher (Fig. 10 Left).

EC2 **Bedroom:** The next scene was the bedroom of a child. The required elements were: a bed, a book shelf, a desk, a lamp, a trashcan, a floor mat, a window, some type of toy (Fig. 10 Middle).

EC2 **Classroom:** The last scene consisted of a classroom and part of the corridor attached to it. The required elements were: the teacher's table and chair, the blackboard, a bookshelf, a clock, a door, four tables and chairs for the students, nine lockers in the corridor (Fig. 10 Right).

### 4.3 Method

All participants went through the same testing procedure, which was divided into five different steps.

In the beginning, 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. Each participant filled an informed consent form.

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

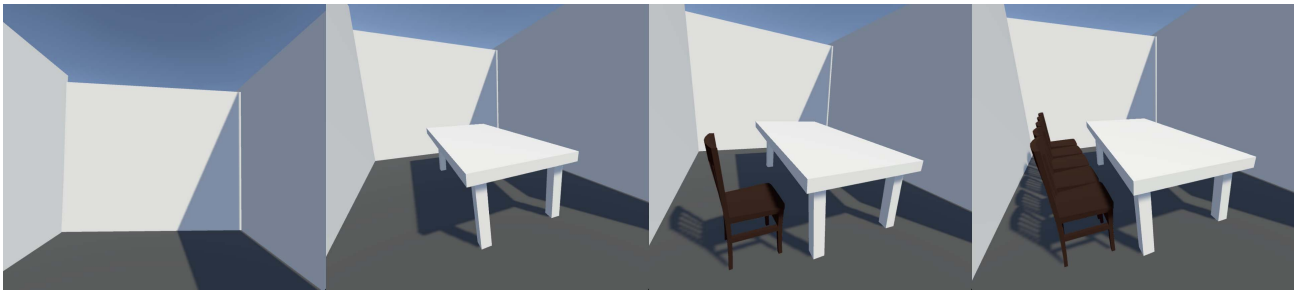

Figure 9: The sequence of four steps for the simple task.

Figure 10: Left: Museum test scene; Middle: Bedroom test scene; Right: Classroom test scene

moving around; experimenting 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.

After that, it was time to get the test subjects into *VR Designer*. We started by explaining the controls to them and, one by one, test all of the features of the prototype: moving inside VR; use of radial menus; adding primitives; importing assets by searching using voice controls; manipulating objects' scale, positioning and orientating; selecting and grouping objects; changing objects' properties and applying modifiers. After all the steps were completed, the subjects were given 5-10 minutes to freely explore the prototype. After resetting the scene, we instructed the subjects to do the four steps of the Simple Task and timed their execution. The subjects were informed that their performance would be timed, but were encouraged to complete them at a comfortable pace without needing to rush.

For the Environments Creation tasks, 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.

After all tasks were complete, the test subjects were taken out of VR and asked to fill a brief questionnaire regarding their experience with *VR Designer*. The questionnaire included questions about the participants' profile, quantitative questions regarding the overall experience using the tool, and 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.

## 4.4 Participants

The tests counted with eight participants. Five reported being highly familiarized with 3D modelling and environment creation, while three had no previous experience with such tasks. Half of the participants had experienced VR before, and only two participants had previously used speech controls. All participants with VR experience were familiarized with 3D modelling.

## 4.5 Results

In this section we will analyse the results we obtained in the two tests, together with data from the questionnaires, to evaluate the prototype in light of the goals set. Regarding task completion times, we compared the control group (3D modelling professionals using a 3D modelling software of their choosing) with participants using our prototype, grouped by reported experience with both 3D modelling software and VR.

### 4.5.1 Simple Task

The Simple Task was made to analyse to which operations our prototype brings benefits when compared with traditional 3D modelling tools. Overall, the task was designed to simulate the basic elements of building a 3D environment, and so, these results reflect what performance, time-wise, could be expected from our prototype (always considering simple scenarios). Fig. 11 shows the completion times of all steps. The results between the control group and both of the experienced groups are shown to be very close, with some subjects even managing to be faster.

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.

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.

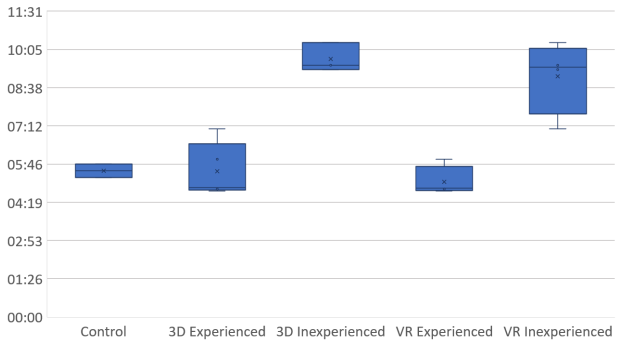


Figure 11: Time to complete the simple task, grouped by participants' previous experience.

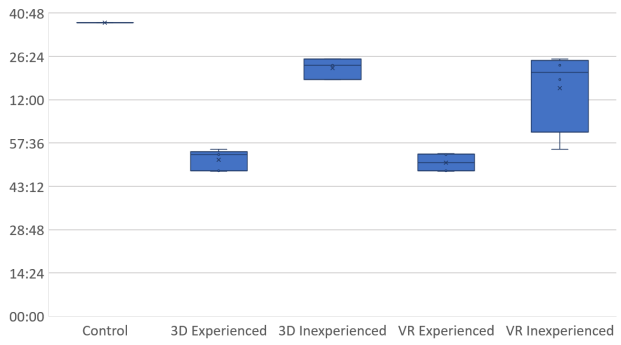


Figure 12: Box chart of the total environment creation times, grouped by participants' previous experience.

#### 4.5.2 Environment Creation

The Environment Creation tasks were 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. 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. Analysing the total times (Fig. 12), we can see that, similarly to the Simple Task, 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 shows that our prototype can enable users without specific technical skills to easily and intuitively create VR environments.

#### 4.5.3 User Feedback

The questionnaire enquired participants about the usability of the prototype. The results obtained can be seen in Table 1.

Regarding interaction and manipulation (questions 1, 2, and 3), we can see that we got no negative answers, with question 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.

One of the core functionalities of our prototype is the ability to search for and import 3D objects from free online repositories (questions 4 and 5). The results are overwhelmingly positive which validates our efforts into making this core feature as accessible as possible.

One of the major contributors for the accessibility of our prototypes are the voice controls (question 6). The results are mostly positive, although some improvements could be made to the speech recognition in the future.

Another big component of our prototype are the procedural tools, which we call modifiers (question 7). This module had mostly neutral answers, which can be justified by the fact that the scenes created did not have a significant use for them, and also due to the small number of available modifiers at the time. Nevertheless, we believe that in scenes with higher complexity, the mechanism can be quite useful, and therefore we expect to further explore this component in the future.

The questionnaire included an 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. In addition to questions 8 and 9 shown in Table 1, we asked participants why (or why not) they would use VR Designer again in the future (question 10), and what they thought that could be improved in the prototype (question 11).

As can be seen by the results of questions 8 and 9, a great majority of the users thought the prototype to be useful and would be interested in its future developments. 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 to questions 10 and 11: "I could use a tool like this to prototype worlds and make sure environment dimensions are correct for VR.", "to make VR environments quick", and "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: (1) Access to more objects - this could be done by adding access to more online 3D repositories; (2) 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[2] or Texture Haven[3], from which they could directly pick textures to use in their scenes; (3) Improved speech recognition - hile 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" versus "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.

[2]CC0 Textures: `https://cc0textures.com/`
[3]Texture Haven: `https://texturehaven.com/`

Table 1: Answers to the questionnaire. Reporting median and interquartile range. Higher values are better.

| **Question** | **Median (IQR)** |
|---|---|
| 1. How easy did you find it was to interact with the objects around you? | 5 (0) |
| 2. How easy did you find it was to manipulate the objects around you? | 4.5 (1) |
| 3. How easy did you find it was to choose the direction you wanted to manipulate an object? | 4 (1) |
| 4. How useful did you find the direct access to free 3D repositories to be? | 5 (0) |
| 5. How easy did you find it was to import objects? | 5 (0) |
| 6. How easy did you find it was to search for objects using your voice? | 4 (1) |
| 7. How useful did you find the object modifiers to be? | 3 (1.5) |
| 8. How useful did you find VRDesigner to be? | 4 (1) |
| 9. Do you see yourself using it in the future, if it is ever released? | 4 (1.5) |

## 5 Conclusions

VR Designer is an example of a tool that integrates VR, speech control and procedural tools to help the process of 3D environment creation.Overall, we managed to make the tool and test its feasibility, having shown that users without specific technical skills can easily and intuitively create VR environments, having achieved this 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. However, our conclusions are limited due to the reduced size of the test sample, making us unable to make definitive claims about our results. Despite this, the results we obtained give us indications that the application is working as intended.

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 freedom it would give the user, which in the end, translates into creative potential.

## Acknowledgments

This work was partially supported by the ERDF – European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation - COMPETE 2020 under the Portugal 2020 Partnership Agreement, and by National Funds through the Portuguese funding agency, FCT - Fundação para a Ciência e a Tecnologia within project PAINTER with reference POCI-01-0145-FEDER-030740 - PTDC/CCI-COM/30740/2017.

## References

[1] M. Billinghurst, S. Baldis, L. Matheson, and M. Philips. 3D palette. In *Proceedings of the ACM symposium on Virtual reality software and technology - VRST '97*, pp. 155–156. ACM Press, New York, New York, USA, 1997. doi: 10.1145/261135.261163

[2] J. Bloomenthal, C. Bajaj, J. Blinn, B. Wyvill, M.-P. Cani, A. Rockwood, and G. Wyvill. *Introduction to implicit surfaces*. Morgan Kaufmann, 1997.

[3] C.-W. Chen, J.-W. Peng, C.-M. Kuo, M.-C. Hu, and Y.-C. Tseng. Ontlus: 3D Content Collaborative Creation via Virtual Reality. In K. Schoeffmann, T. H. Chalidabhongse, C. W. Ngo, S. Aramvith, N. E. O'Connor, Y.-S. Ho, M. Gabbouj, and A. Elgammal, eds., *24th International Conference on Multimedia Modeling*, pp. 386–389. Springer International Publishing, Bangkok, Thailand, 2018. doi: 10.1007/978-3-319-73600-6_38

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

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

[6] B. Jackson and D. F. Keefe. Lift-off: Using reference imagery and freehand sketching to create 3d models in vr. *IEEE Transactions on Visualization and Computer Graphics*, 22(4):1442–1451, 2016. doi: 10.1109/TVCG.2016.2518099

[7] P. Janssen and R. Stouffs. Types of parametric modelling. 2015.

[8] J. Jerald, P. Mlyniec, A. Yoganandan, A. Rubin, D. Paullus, and S. Solotko. Makevr: A 3d world-building interface. In *2013 IEEE Symposium on 3D User Interfaces (3DUI)*, pp. 197–198, 2013. doi: 10.1109/3DUI.2013.6550246

[9] W. Linder. *Digital photogrammetry*, vol. 1. Springer, 2009.

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

[11] D. Mendes, D. Medeiros, M. Sousa, R. Ferreira, A. Raposo, A. Ferreira, and J. Jorge. Mid-air modeling with boolean operations in vr. In *2017 IEEE Symposium on 3D User Interfaces (3DUI)*, pp. 154–157. IEEE, 2017.

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

[13] R. Nóbrega and N. Correia. Design your room: Adding virtual objects to a real indoor scenario. In *CHI '11 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '11, p. 2143–2148. Association for Computing Machinery, New York, NY, USA, 2011. doi: 10.1145/1979742.1979851

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

[15] P. B. Pascoal, D. Mendes, D. Henriques, I. Trancoso, and A. Ferreira. Ls3d: Lego search combining speech and stereoscopic 3d. *International Journal of Creative Interfaces and Computer Graphics (IJCICG)*, 6(2):18–36, 2015.

[16] L. Piegl and W. Tiller. *The NURBS book*. Springer Science & Business Media, 1996.

[17] M. Russo. *Polygonal modeling: basic and advanced techniques*. Jones & Bartlett Learning, 2006.

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

[19] J. Wang, O. Leach, and R. W. Lindeman. Diy world builder: An immersive level-editing system. In *2013 IEEE Symposium on 3D User Interfaces (3DUI)*, pp. 195–196, 2013. doi: 10.1109/3DUI.2013.6550245