INNOVATIVE MIXED REALITY ADVANCED MANUFACTURING ENVIRONMENT WITH HAPTIC FEEDBACK

A Thesis

Submitted to the Faculty

of

Purdue University

by

Jesse C. Satterwhite

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science in Electrical and Computer Engineering

August 2018

Purdue University

Indianapolis, Indiana

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Zina Ben-Miled, Co-Chair

Department of Electrical and Computer Engineering

Dr. Hazim El-Mounayri, Co-Chair

Department of Mechanical Engineering

Dr. Christian Rogers

Department of Computer Graphics Technology

Dr. Tamer Wasfy

Department of Mechanical Engineering

Approved by:

Dr. Brian King

Head of the Graduate Program

To Mom, Dad, and Rachel, who made it all possible.

To Ally, who made it all worth it.

ACKNOWLEDGMENTS

I'd like to first thank my thesis committee at IUPUI for their help in making this possible. Dr. Zina Ben-Miled, Dr. Hazim El-Mounayri, Dr. Christian Rogers, and Dr. Tamer Wasfy have all aided in getting me to this point. In particular, I'd like to thank Dr. El-Mounayri and Dr. Rogers, who have aided me in my research since the summer of 2015.

I'd also like to thank John Grove, Mounir Karmada, and Saurabh Shirolkar, each of whom aided with the MR-AVML study. Saurabh has worked with me on various studies with his section of ME 54600 for the last three semesters at IUPUI, and has been a tremendous help because of that. It was a pleasure serving as John and Mounir's MURI mentor during the 2017 - 2018 academic year.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure Page

ABBREVIATIONS

AVML	Advanced Virtual Manufacturing Lab
VR	Virtual Reality
MR	Mixed Reality
CNC	Computer Numerically Controlled

GLOSSARY

eLearning — Software designed to teach users, with the intention of serving as a replacement to traditional classroom learning.

G-code — The programming language of CNC machines, defined by a categorizing letter followed by a number.

Haptic feedback — A simulated sense of touch provided during virtual and mixed reality environments.

Unity3D — A 3D environment rendering engine commonly used to create VR and MR projects; the current rendering engine partnered with Microsoft on the HoloLens.

ABSTRACT

Satterwhite, Jesse C. M.S.E.C.E., Purdue University, August 2018. Innovative Mixed Reality Advanced Manufacturing Environment with Haptic Feedback. Major Professors: Zina Ben-Miled, Hazim El-Mounayri.

In immersive eLearning environments, it has been demonstrated that incorporating haptic feedback improves the software's pedagogical effectiveness. Due to this and recent advancements in virtual reality (VR) and mixed reality (MR) environments, more immersive, authentic, and viable pedagogical tools have been created. However, the advanced manufacturing industry has not fully embraced mixed reality training tools. There is currently a need for effective haptic feedback techniques in advanced manufacturing environments.

The MR-AVML, a proposed CNC milling machine training tool, is designed to include two forms of haptic feedback, thereby providing users with a natural and intuitive experience. This experience is achieved by tasking users with running a virtual machine seen through the Microsoft HoloLens and interacting with a physical representation of the machine controller.

After conducting a pedagogical study on the environment, it was found that the MR-AVML was 6.06% more effective than a version of the environment with no haptic feedback, and only 1.35% less effective than hands-on training led by an instructor. This shows that the inclusion of haptic feedback in an advanced manufacturing training environment can improve pedagogical effectiveness.

1. INTRODUCTION

Recent advancements in virtual reality (VR) and mixed reality (MR) have led to immersive, authentic, and viable pedagogical tools. These tools range from the traditional basic headset interaction, to more complex setups that include various physical feedback systems, to mixed reality environments that use physical objects as the primary method of interacting with the virtual world. Each of these methods is suitable for an eLearning environment, a tool designed to provide information and training to users without the need for an instructor [1], [2], [3]. The differences between these environments lies in the amount of haptic feedback they provide. Traditionally, environments rely on more feedback [4]. For example, having students press a physical button instead of reaching to touch a virtual button. Both achieve the same result, but more feedback is provided with the physical button. These interactions are considered to be more realistic [5].

Virtual reality and mixed reality tools are also utilized in industry. In the field of advanced manufacturing, these environments aid their users with a variety of tasks. These tasks include machine maintenance, complex assembly, factory planning, quality assurance, and training. However, these environments often lack the advanced feedback methods that have been shown to improve the effectiveness of other virtual environments [6]. This is specifically true for computer numerically controlled (CNC) machine simulation tools [7], [8], [9], [10]. Further, the immersive environments that do exist for training in advanced manufacturing are primarily VR [11], with very few designed for MR [6].

In such cases where environments are immersive in nature, the required setup is often expensive and bloated with advanced hardware [12]. While some tools are designed to bypass the increasing hardware costs in VR and MR environments, such as in [4], their feedback methods do not provide specific enough sensations to improve immersion in advanced manufacturing environments. It is therefore desirable to find a middle ground between using the most advanced hardware for specific feedback and the most cost-effective alternatives for less-specific feedback. This can be defined via a tier system for VR and MR environments. The 3 tiers for categorizing feedback and requirements for VR and MR systems is as follows:

- Tier 3: smaller scale, minimal space required, less-accurate haptic feedback techniques.
- Tier 2: smaller than room-scale but requires more space than standing-room-only (e.g. Tier 1 environments), accurate haptic feedback techniques.
- Tier 1: room-scale, advanced haptic feedback techniques for near true-to-life sensations.

The most advanced virtual environments strive to fit into the Tier 1 category, while alternative tools look to fit into Tier 3 to compensate for the rising cost of Tier 3 systems. Few commercial products intend to fit into Tier 2, despite it featuring benefits of a more cost-effective setup with effective feedback techniques. For example, Boeing's implementation of MR in industry - one of the leaders of the technology in advanced manufacturing - requires additional cameras and tracking systems in its lab-scale environment. This qualifies it as a Tier 1 system [13].

Feedback techniques implemented in VR and MR environments are commonly defined as haptic force feedback. Haptic feedback is often provided through vibration motors and force feedback gloves [3], [4], though it can also be implemented via physical object interaction within virtual environments [1], [2], [3]. It is therefore clear that two different types of haptic feedback exist: one style is force feedback caused by a users actions; the other style is a constant force that can be felt regardless of the user's actions. When typing on a keyboard, force feedback is felt with every key typed, alerting a user that they are successfully entering data. When not typing at all, the keyboard itself can still be felt and recognized as a physical object. These two types of feedback fit into the two styles described previously. These two styles of feedback will hereafter be referred to as active and passive haptics, respectively, and will adhere to the following definitions:

- Active haptics: reactionary feedback that presents a different sensation when a user performs an action, but otherwise provides nothing.
- Passive haptics: static feedback that, when felt by the user, provides the same sensation regardless of user action.

Most environments implement one style of haptic feedback or the other, but, as will be discussed in Chapter 2, incorporating both into an environment can be more effective than only having one [3].

Taking each of these points into account, it is clear that several gaps exist in the field of VR and MR environments in advanced manufacturing. In particular, both mixed reality training tools and immersive CNC training tools are lacking in industry. Thus, the industry has yet to fully take advantage of recent advancements in technology. Further, in VR and MR, there exists a dearth of Tier 2 systems that take advantage of advanced haptic feedback techniques provided in a cost-effective package. Bringing together all of these points, one solution to these gaps is the MR-AVML, a CNC machine training tool designed for use with the Microsoft HoloLens. This proposed tool can be tested and compared against both an immersive environment without haptic feedback and a traditional hands-on training in order to assess the effectiveness of haptic feedback techniques in an advanced manufacturing training environment.

The remainder of the thesis is organized as follows. Related works is the focus of Chapter 2. In Chapter 3, the motivation and design of the proposed MR-AVML system is discussed. Chapter 4 discusses the implementation of the MR-AVML. Chapter 5 then describes the design of the usability and pedagogical studies conducted on the proposed system. The results of these studies are included in Chapter 6 and are discussed in Chapter 7. Conclusions and direction for future work are provided in Chapter 8.

2. LITERATURE REVIEW

In this chapter, a review of the efficacy of eLearning tools, the benefits of haptic feedback, and the current state of virtual environments in advanced manufacturing is presented. The proposed mixed reality environment features elements of each of these areas. It is therefore necessary to validate that a gap exists therein that the MR-AVML will be able to fill.

2.1 Benefits of Haptic Feedback

Haptic feedback has been shown in numerous environments to improve pedagogical effectiveness. RoboStage is one such example of a pedagogical environment improved by haptic feedback. It consists of several robots and two perpendicular screens that form the virtual scene of an immersive foreign language learning environment. In order to interact with the RoboStage environment, students were required to command the robots with voice controls or physically move them to complete a given task. For example, have the robot move to a specific location, or have it speak to another robot. During the study associated with RoboStage, a baseline keyboard and mouse version of the simulation was created. Two groups of students used the baseline environment, while two more groups used the mixed reality version with robots. The results of this study indicated that the mixed reality version with haptic feedback greatly increased the authenticity of the environment compared to the non-immersive version. It also motivated the students to learn [1].

The benefits of haptic feedback are further validated by two additional studies with a focus on interaction with physical objects. It was found during a study by Hoffman that having users interact with virtual objects by moving a physical counterpart makes environments feel more realistic [2], including both passive and active haptic feedback can improve environments further than just the inclusion of passive feedback. In their study, users interact with a control panel in VR and receive haptic feedback through two methods:

- Active haptic feedback: a force-feedback glove provides sensations when the user presses a virtual button.

- Passive haptic feedback: a wooden slab that is roughly the shape, size, and position of the virtual control panel.

Three separate groups were tested for this study: one with only active feedback, one with only passive feedback, and one with both. Results showed that the inclusion of passive feedback was more effective than active feedback, but that both combined provided the most realistic and intuitive environment of the three. This result was obtained via eight trials in which virtual panel input errors were measured. The virtual environment's lighting darkened after each trial, forcing subjects to rely more on the haptic feedback sensations to accurately interact with the panel. Errors were significantly reduced for the mixed group, followed by a notably more errors in the active-only group than the passive-only group [3].

From these three studies, it can be concluded that the inclusion of haptic feedback in learning environments improves their realism. While it has been argued that using high end hardware can improve immersion in environments, haptic feedback can be included without the need for expensive equipment. Novy echoes this belief in his research [5]. It is possible to create an immersive pedagogical environment with a cost-effective haptic feedback that improves on less immersive environments.

2.2 Virtual and Mixed Reality in Advanced Manufacturing

In the last few years, advanced manufacturing industries have started to embrace the growing technologies of virtual and mixed reality. These technologies have not been fully embraced yet, but their use is growing. Wright provides an overview of the current state of VR and MR in advanced manufacturing in his eBooks [11] and [6], respectively. In each, he lists out five categories that the technologies are most commonly used for.

Virtual Reality [11]:

- Training.
- Factory planning.
- Maintenance.
- Automation.
- Assembly.

Mixed Reality [6]:

- Complex assembly.
- Maintenance.
- Expert support.
- Quality assurance.
- Automation.

From these lists, as well as Wrights explanations, it can be concluded that the current landscape of VR and MR in the industry focuses on planning and training in virtual reality, whereas mixed reality applications are primarily designed to aid shop floor workers through various tasks. Observing prominent examples from industry should substantiate this conclusion.

One of the most widely known examples of MR in the manufacturing industry comes from Boeing, which has been testing a mixed reality tool on tablets that provides mechanics with digital instructions, parts and arrows to augment the real-world view [13]. Based on what the mechanic is working with at any moment, cameras and tracking systems note what the mechanic is doing and provide the appropriate aid [13]. As another example, Volvo has a similar system that enables mechanics to see instructions and information about what they are working on via the Microsoft HoloLens [6]. Both of these tools fit under the previously listed category of complex assembly. Similar tools exist for maintenance and quality assurance, as well [6]. One tool that does not fit into the listed MR categories in advanced manufacturing is the LKDF Interact, created by EON Reality for Volvos Selam Vocational Training Centre. This application provides users with training on the proper maintenance of diesel engines. The tool itself is strictly meant for training, and as a result does not give instructions during the maintenance process itself [14]. For virtual reality, a prominent example cited by Wright is the Fronius VR welding training tool [11]. In this environment, users utilize physical replicas of welding equipment tracked by virtual welding stations to learn welding processes step by step in individual exercises and gradually develop a technical feel for the correct parameter settings [15]. Another commercial VR training tool comes from Pratt & Whitney Customer Training, which recently invested in an immersive tool for aviation engine manufacturing [16].

2.3 CNC Simulation Software

One area that could benefit greatly from virtual reality systems in advanced manufacturing is the specific field of training on computer numerically controlled (CNC) machines. These machines are expensive and require a lot of training to operate correctly. It is therefore the priority of many companies to minimize the amount of training time required on a physical machine by implementing more cost-effective virtual systems. Indeed, several applications have been created to address this need, but the most popular among them are non-immersive systems.

CNC Simulator Pro has several features that make it effective as a training tool, including built-in CAD/CAM modeling functionality, real-time part machining, and training options for multiple CNC machines [7]. Another tool is the CGTech Vericut CNC Machine Simulation, which boasts a realistic approach to machining simulation with an advanced collision checking process that alerts users when they have entered a G-code program that could potentially cause damage to the CNC machine. This feature allows trainees to learn not to make expensive mistakes in a low-pressure environment [8]. The above two environments could be considered virtual reality due to the way they realistically simulate the machining process, but they are non-immersive. Other non-immersive environments include the Predator Virtual CNC Machine software [9] and the Cutviewer [10].

2.4 Comparison to the Proposed System

The three previous sections describe technologies and tools that are effective. Virtual reality and mixed reality tools perform well in advanced manufacturing, and training tools can be effective even without immersive components. However, as discussed in Section 2.1, the effectiveness of training environments can be enhanced with the inclusion of haptic feedback. This is what the proposed MR-AVML will address. While the existing CNC simulation and training tools work well, there is potential for improvement with the inclusion of immersive technologies. Further, as discussed in Section 2.2, mixed reality training tools are not commonplace. With the MR-AVML, we will show that MR training tools can be effective. Designing, implementing, and evaluating the proposed MR-AVML is the focus of this thesis.

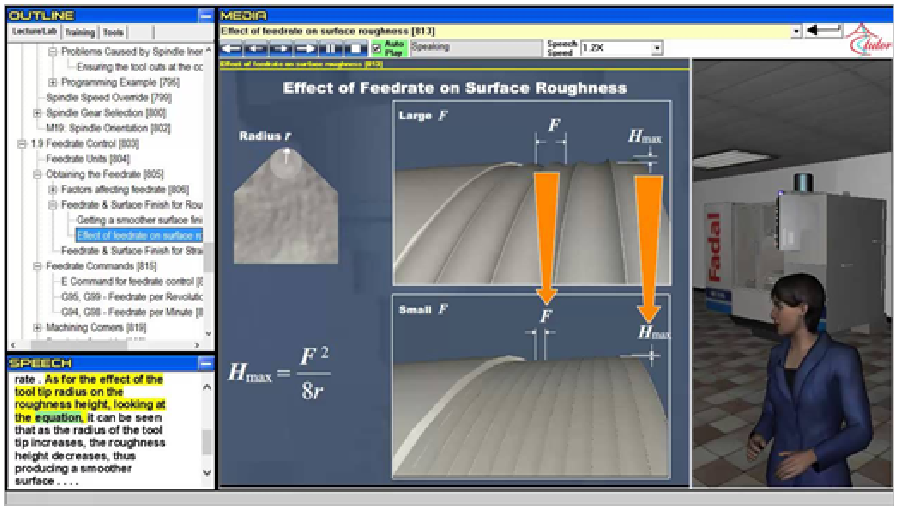
3. PROPOSED MIXED REALITY ENVIRONMENT

The MR-AVML leverages the knowledge gained during the design and development of the two previous versions of the software - the non-immersive AVML and the immersive VR-AVML - to provide a system that benefits from both natural input methods and an immersive environment. This chapter presents the design of the MR-AVML. It begins with an overview of each version of the Advanced Virtual Manufacturing Lab to provide context to the systems discussed. This is followed by a discussion of the motivations behind the design and development of each version. The design of the MR-AVML will also be discussed in-depth, including the software and hardware requirements, and the philosophy behind its design.

3.1 Non-Immersive Virtual Reality Environment: the AVML

The original AVML was developed in 2006. It was written using Visual Basic and designed as a web-based application accessible through a custom browser. The system provides a full class's worth of instructions on the use of the Fadal VMC-3016L in a "Lecture Lab" and a functional virtual replica of the aforementioned CNC machine in a "Training Lab". The goal was to provide a system capable of training users in a controlled environment that would give them the ability to learn how to use the machine without fear of harming either themselves or the machine [12].

Lectures are provided to the user through a virtual instructor who "reads" the lectures through text-to-speech software. While the lectures are playing, the display of the system shows users either informational slides about what the lectures are describing or a demonstration of a virtual user operating the CNC machine. Optional quizzes are provided during the lecture segments that allow users to assess the portions of the training that they need to review. Users should be provided with all of the operation and safety information needed to operate a Fadal VMC-3016L by the end



OUTLINE
Lecture/Lab
Training
Tools
Problems Caused by Spindle Iner
Ensuring the tool cuts at the cc
Programming Example [796]
Spindle Speed Override [799]
Spindle Gear Selection [800]
M19: Spindle Orientation [802]
1.9 Feedrate Control [803]
Feedrate Units [804]
Obtaining the Feedrate [805]
Factors affecting feedrate [806]
Feedrate & Surface Finish for Rou
Getting a smoother surface fini
Effect of feedrate on surface ro
Feedrate & Surface Finish for Stra
Feedrate Commands [815]
E Command for feedrate control [8
G95, G99 - Feedrate per Revolutic
G94, G98 - Feedrate per Minute [8
Machining Corners [819]
MEDIA
Effect of feedrate on surface roughness [813]
Auto Play
Speaking
Speech Speed
1.2X
Effect of Feedrate on Surface Roughness
Radius r
Large F
F
H_{max}
Small F
F
H_{max}
$H_{max} = \frac{F^2}{8r}$
Fadal
SPEECH

rate . As for the effect of the tool tip radius on the roughness height, looking at the equation, it can be seen that as the radius of the tool tip increases, the roughness height decreases, thus producing a smoother surface . . . .

Fig. 3.1. The "Lecture Lab" from the AVML.

of the lectures [17]. Figure 3.1 shows one of these lectures in progress. As shown in the figure, an outline of the lecture is in the top left. The virtual instructor is shown on the right, with a transcript of her lecture shown in the bottom left. The center of the screen displays either the lab or the lecture slides, depending on what is called for by the software at that moment.

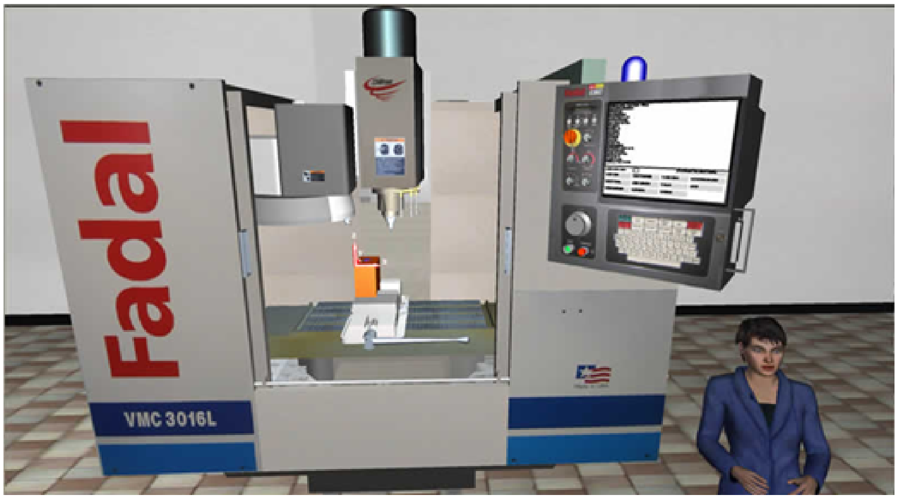


Fig. 3.2. The "Training Lab" from the AVML.

The ”Training Lab” segment of the AVML allows users to roam freely around a lab environment containing the CNC machine. During this experience, users are able to fully interact with the machine as they want, allowing for experiential learning with no fear of causing a machine-breaking mistake. Users are also able to view virtual demonstrations of the machine use during this segment with step-by-step instructions on several machine operations. These demonstrations can be selected from a list on the left side of the software display or by entering a specific request in a search bar at the top of the screen (e.g. ”Show me how to power on the Fadal VMC-3016L”). The full software shows the lab and instructor as seen in Figure 3.2. The remainder of the display is similar to what is shown in Figure 3.1, but with the outline replaced with a list of demonstrations the system can provide [17].

3.2 Immersive Virtual Reality Environment: the VR-AVML

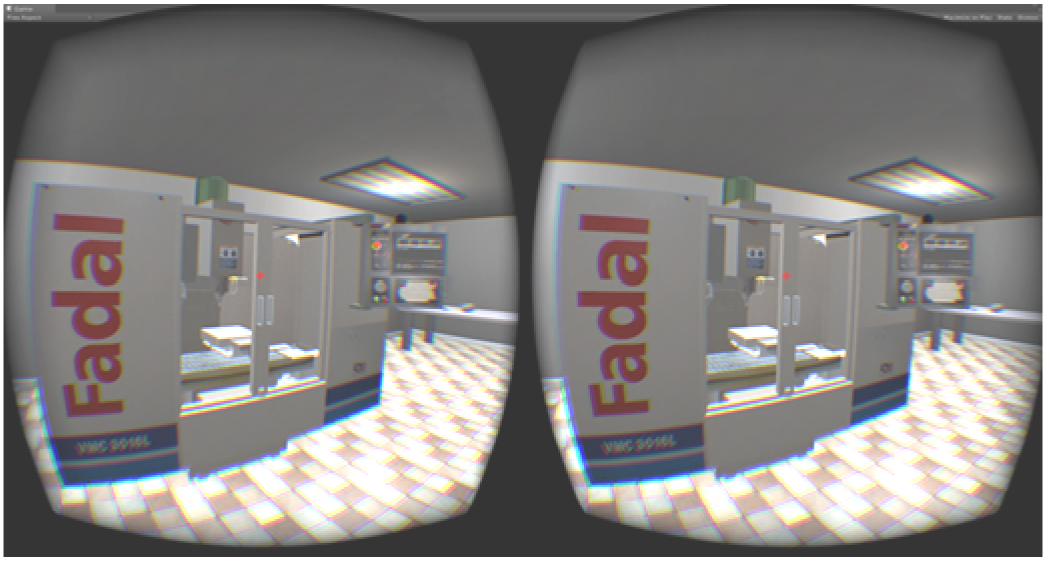
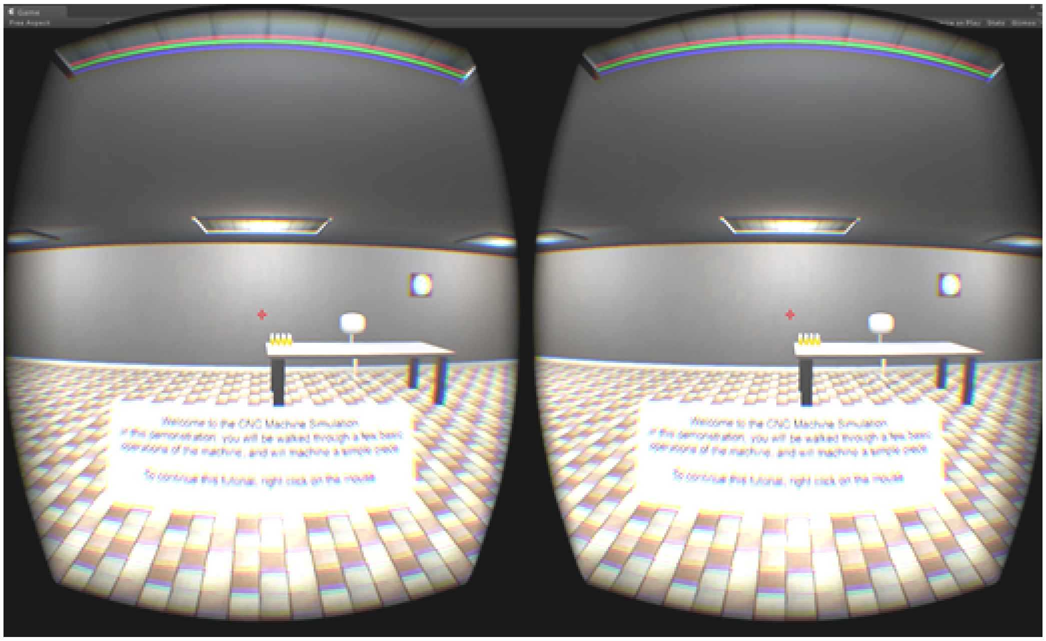


Fig. 3.3. The Fadal VMC-3016L as seen during the VR-AVML.

I developed the VR-AVML from 2015 to 2017. This version was created using the Unity3D game engine and designed to be viewed through the Oculus Rift virtual reality headset. Figure 3.3 shows the CNC machine as seen from the Oculus Rift. While its design is based on the original AVML, no code is shared between this version and its predecessor. However, models of the Fadal VMC-3016L and the surrounding lab are reused from the original AVML. The ”Lecture Lab” and ”Training Lab” portions of the original AVML were also truncated into a single experience in the VR-AVML. The system is designed as a Tier 3 VR system, defined here by the user sitting down at their computer and operating the system with a mouse and keyboard along with the VR headset. No additional sensors or tracking devices are required to operate the VR-AVML aside from the base Oculus Rift headset [18].



Welcome to the CNC Machine Simulation
In this demonstration, you will be walked through a few basic operations of the machine, and will machine a simple piece.

To continue this tutorial, right click on the mouse.

Fig. 3.4. Instructions displayed to users during VR-AVML operation.

The main tutorial in the VR-AVML provides instructions on how to power on the machine, navigate through the main menu, manually enter machine commands, and load code from a disk. Instructions are provided to the user through a text tutorial that they can look through at their own pace, rather than the virtual instructor from the AVML. The tutorial is displayed in the lower third of the user’s display, allowing them to see the instructions at any time without taking up too much of their vision. This is shown in Figure 3.4.

Inputs in the VR-AVML are handled through a cursor selection system. In the center of the user's vision is a red crosshair. When the user left clicks with the mouse, the object in the sight of the crosshair is ”selected”, initiating whatever action is associated with that object. This is how switches are flipped, how buttons are pressed, and how key commands are typed in the VR-AVML. It is also how manual G-code entry is handled.

The VR-AVML was designed to be a proof-of-concept that can be used to verify the pedagogical capabilities of virtual reality in advanced manufacturing. As such, its instructions focus on information needed to carry out its main tutorial. The framework allows for new tutorials. However, it currently does not have the same amount of lectures available as in the original AVML.

3.3 Mixed Reality Environment: the MR-AVML

Like the VR-AVML before it, the MR-AVML is created using Unity3D as its rendering engine. Its display, however, is unique in that it is rendered from the Microsoft HoloLens augmented reality headset. The HoloLens displays virtual items as holograms by projecting them into the physical environment. The HoloLens automatically measures the physical area it is in, allowing virtual objects to be tethered to their physical surroundings. It is through this that the CNC machine is placed in the room, allowing users to walk around and view the entire machine.

Once again, the model of the Fadal VMC-3016L is borrowed from the original AVML. Some animations featured in the MR-AVML are reused from the VR-AVML, but no code is taken from the older versions. The design of the MR-AVML is similar to that of the VR-AVML, with the lecture and training segments of the AVML again combined into a single learning experience. Users view the virtual hologram of the CNC machine among the physical environment, and are able to interact with it using both a cursor selection system similar to the one used in the VR-AVML, as well as with a physical input method. The full design of the MR-AVML will be discussed at length in the following sections of this chapter.

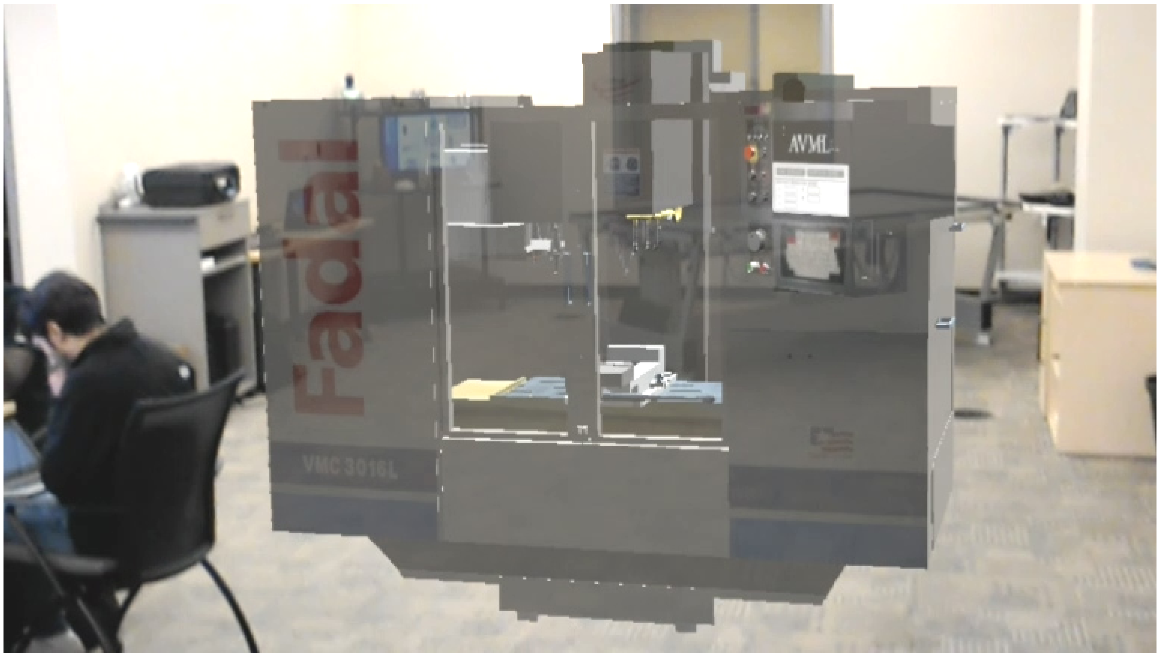


Fig. 3.5. Early implementation of the MR-AVML hologram without physical input.

3.4 Motivations

Development of the original AVML was motivated by the desire to provide a safer, cheaper alternative to CNC machine training. The web-based program allowed users to interact with a virtual recreation of a Fadal VMC-3016L while receiving lessons from a virtual instructor [12]. This enabled effective training without a physical machine or a physical instructor. There were three main benefits to this: first, the trainee would not need to work with a physical machine, eliminating the safety concerns associated with a new user operating the heavy machinery; second, a machine would not have to be dedicated to training while teaching a person how to operate it, meaning that machine would not have to be taken off of a workshop floor during that time; and third, not needing a physical instructor meant that the trained person who normally would be responsible for training could continue with their job as normal while a new person is being trained [12].

The VR-AVML was motivated by the same desires as the original AVML, providing an alternative to hands-on CNC machine training, with the additional motivation to validate the educational efficacy of virtual reality in advanced manufacturing. Leveraging the experience of the AVML and adding the immersive benefits of virtual reality would, in theory, provide a more engaging and effective learning experience for users. Using the same information from the original version, the VR-AVML would be able to show the value of VR [18].

The development of the MR-AVML was motivated by a desire to address the gap between the VR-AVMLs educational abilities compared to instructor training. The cursor input method of the VR-AVML, which relied on head movement to precisely align the cursor with the desired key to register an input, was found to be unnatural and cumbersome by most users during the system's usability study [18]. This cursor input method was designed to provide an alternative to using a hand tracking system to press small virtual keys closely grouped together. The hand tracking would provide a more natural input method, but the inaccuracies brought from this style of interaction would be frustrating for users. The cursor provides higher accuracy, therefore reducing potential frustration at the system. However, in practice, users felt it was unnatural [18]. The conclusion was that this input method could not be improved without significantly retooling the system's required hardware. The VR-AVML only required enough space for a user to sit in front of a computer with a mouse and a keyboard while wearing the Oculus Rift headset, defining it as a Tier 3 VR system. Adding hand tracking for virtual keyboard input would raise the cost of the environment significantly. However, even if the design were retooled, the lack of precision due to the nature of the input method being emulated would still compromise the system.

In a mixed reality system, these compromises are not present. By implementing a physical controller alongside a virtual machine, and allowing the user to see both simultaneously, inputs can be both precise and natural for users. Fully taking advantage of the benefits of mixed reality in this way will create a more immersive system and, therefore, begin to close the gap between virtual CNC training tools and instructor-led training. Further, validating the MR-AVML as an effective training system will show the capability of more cost-conscious Tier 2 systems as an alternative to the more advanced feedback methods in Tier 1 systems.

3.5 Software Requirements

- The application shall be programmed in C#.
- The application shall use Unity3D as its rendering engine.
- The application shall present users with a hologram of a Fadal VMC-3016L CNC machine.
- The tutorial shall include the same information as the VR-AVML, as will be discussed.
- The hologram shall be scaled to the size of its real-world counterpart automatically, with no input needed from the user.
- The hologram shall be placed in the correct position in the area in which the application is used.
- The hologram shall maintain its position as users navigate the environment.
- The hologram shall allow space for a physical controller to be seen in front of it.
- The application shall allow users to interact with the machine hologram.
- The machine shall have a functional power junction cabinet at the back of the machine, with a power switch and a power button.
- The machine shall power on only when the user activates the switch and button on the power junction cabinet.

- The machine shall have a recreation of the Fadal VMC-3016Ls software, including menus and the ability to read machine G-code.
- The machine shall have a manual entry mode in which users can enter their own G-code and see the appropriate response from the machine.
- Users shall be able to navigate through the menus to load previously-written G-code from a disk.
- Users shall be able to execute loaded code.
- The machine shall mill a small example piece after loaded code is executed.
- The application shall receive text input from the CNC controller via Bluetooth.
- The software shall interpret each text input as a corresponding key press.
- The machine shall respond appropriately to each key press.
- Key presses shall be debounced such that multiple inputs from a single key press do not occur.
- The application shall provide users with a tutorial on how to use the Fadal VMC-3016L CNC machine.
- The tutorial shall include instructions on machine startup, menu navigation, manual code entry, loading code from a disk, executing code, and using the emergency stop.
- The application shall take advantage of the features of the Microsoft HoloLens.
- The application shall use the HoloLenss room tracking for hologram positioning.
- The application shall use the HoloToolkits standard Input Manager for handling events specific to the HoloLens.

- The application shall provide the option of voice input for advancing the tutorial.
- The application shall use one of the standard HoloToolkit cursors for non-keyboard inputs.
- Additional software related to sending text data via Bluetooth shall be written in Python.
- The Python script shall interpret raw input data as key presses.
- The Python script shall send single characters to the HoloLens for each key press.
- The Python script shall run immediately upon its microcontroller powering on.

3.6 Hardware Requirements

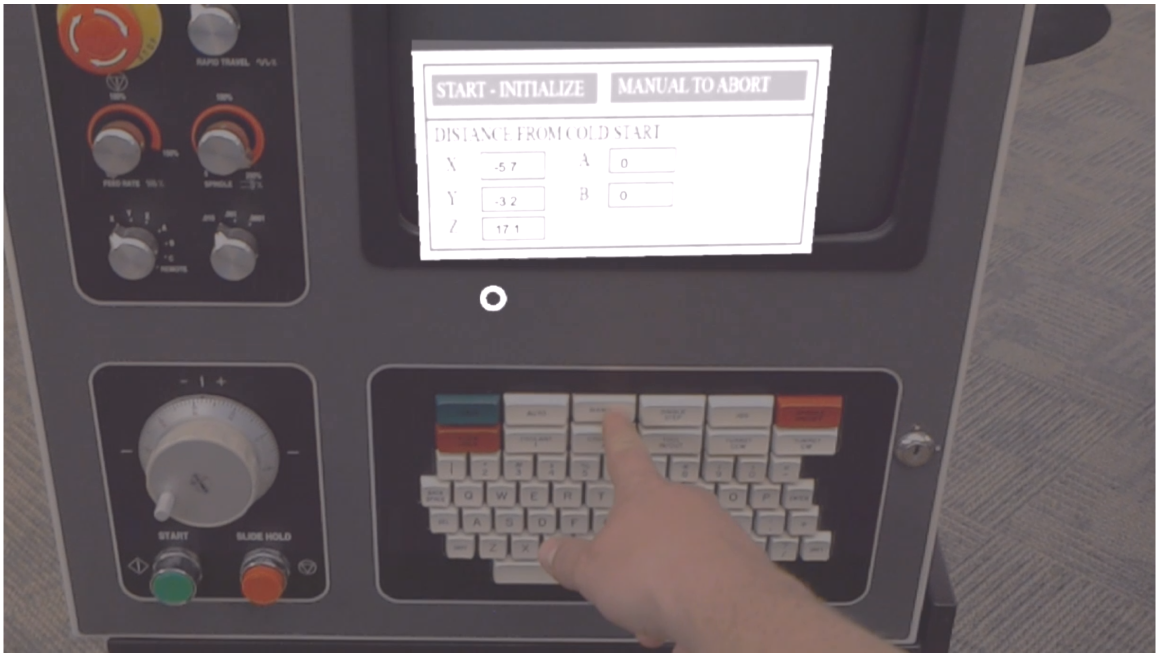
- The main application shall run on the Microsoft HoloLens.
- The additional Python script shall run on a separate Raspberry Pi Zero-W.
- A physical Fadal VMC-3016L CNC machine controller shall be integrated into the environment to provide both active and passive haptic feedback.
- Keyboard input shall be received from the keyboard on the Fadal VMC-3016L controller.
- The keyboard shall provide input as expected, giving one input for one key press.
- The keyboard shall be wired to the Raspberry Pi Zero-W to receive raw data.
- The Raspberry Pi Zero-W shall be placed inside the chassis of the controller and be able to receive power from this position.

- The controller shall be placed in the environment in a position that makes sense relative to where the hologram will be positioned.
- The environment shall allow enough room for users to navigate fully around the machine hologram and the physical controller.
- Marks shall be placed in the environment for consistent tracking.

3.7 Design Philosophy

Environments in mixed reality can incorporate both virtual components and physical components. Some MR tools take advantage of physical objects (e.g. Boeings and Volvos tools [13], [6]), but these tools only recognize physical objects rather than using the objects as an input method. Other MR tools mostly provide holograms inside a real-world environment, with no additional incorporation of physical items. In these cases, where a user is only manipulating holograms, there is no advantage to using MR over a room-scale VR environment. An ideal MR environment has physical components and virtual components that are linked where interacting with the physical piece causes changes in the virtual piece, and where manipulating the virtual piece changes how the physical piece is used. The MR-AVML was designed to fulfill this ideal version of a mixed reality environment.

The MR-AVML is ideal for a mixed reality environment because interactions with the Fadal VMC-3016L controller cannot be emulated accurately in virtual reality. As discussed in Section 3.4, while a CNC machine training environment can be created and used effectively in VR, the VR-AVML was found to suffer from unnatural code input [18]. The MR-AVML can instead incorporate a physical controller representation, enabling accurate user input that also feels natural compared to user expectations. Figure 3.6 shows the natural interactions possible in the MR-AVML. Figure 3.7 shows a user lining up the red crosshair in order to press keys in the VR-AVML, a process that was found to be unintuitive. The same commands are required in each of the AVML versions, but typing is more natural with the physical keyboard.



START - INITIALIZE

MANUAL TO ABORT

DISTANCE FROM COLD START

X -5.7

Y -3.2

Z 17.1

A 0

B 0

Fig. 3.6. Pressing a key in the MR-AVML.

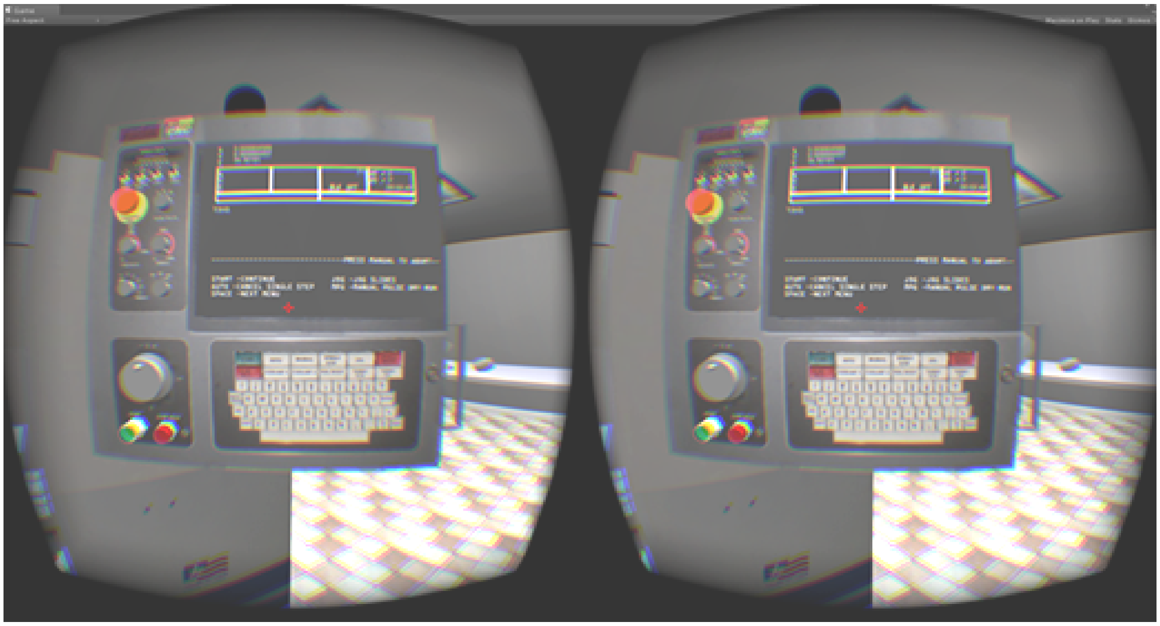


Fig. 3.7. Pressing a key in the VR-AVML.

Interactions with this physical controller shall be meaningful to the system as a whole, both influencing the hologram and being influenced by the holograms current state. Key presses in the MR-AVML are context sensitive, as they are in the real-world counterpart. Depending on what menu the machine is on, what mode it is in, or if it is powered on at all, any individual key press may navigate to another menu, initialize the machine, type one character in manual entry, execute code, or do nothing at all. The physical controller is integrated seamlessly into the virtual environment with a masking effect applied to the side of the machine model in Unity3D. Typically, holograms are overlaid on top of whatever is seen through the HoloLens. This includes instances where the hologram would normally be in the background, meaning a hologram can be viewed behind walls or other objects in cases where the hologram is misplaced. The applied masking allows the physical controller to be seen in front of the virtual machine when appropriate.

The controller is also designed to incorporate the two forms of haptic feedback defined in the Related Work section. The two types of haptics were defined as: active haptics, which provide reactionary force feedback; and passive haptics, which provide the same feedback regardless of the user interaction. These are both featured to an extent in the MR-AVMLs physical controller. Passive haptics are found in the controller itself. As a physical object in the environment, it can be touched and otherwise interacted with exactly as expected. Regardless of the user's actions, feedback felt from the controller itself is a constant. Active haptics are experienced whenever pressing keys on the controller - lightly pressing a key gives the sensation of physically touching an object, but a more forceful press provides an additional feeling, indicating that an input will be received. Physical keyboard input is usually associated with tactile feedback. However, for this project, physical keyboard input was chosen to fulfill the role of active haptics due to it fitting the definition of reactionary force feedback. The incorporation of these haptic feedback techniques take advantage of the mixed reality characteristics of the system, while also improving the systems immersion and teaching abilities.

The MR-AVML is designed to achieve these styles of haptic feedback in a limited space environment and with a minimal amount of hardware. No additional sensors are required in the simulation room, nor are they required on the hands of the user. The user's hands are also free of any additional force feedback tools. The only requirements are the HoloLens, the HoloLens software, and a keyboard sending data to the HoloLens, housed inside of a chassis that approximates the size, shape, and features of the Fadal VMC-3016Ls controller. The hardware requirements for the system were intentionally limited in order to define it as a Tier 2 system. Because of its room-scale requirements and the presence of some external components, it is more complex and demanding than a Tier 3 mixed reality system. However, the design is kept simple in order to be more cost-conscious than a Tier 1 system.

4. IMPLEMENTATION OF THE MR-AVML

The MR-AVML was designed according to the requirements and design philosophy. At a high level, two systems comprise the MR-AVML: the controller replica system and the simulation system. These systems are comprised of the following components.

Controller replica:

- A retrofitted Fadal VMC-3016L controller keyboard.
- A Raspberry Pi Zero-W.
- Software handling raw input data.

Simulation system:

- The Microsoft HoloLens.
- The simulation software.

Input handled by the controller replica is sent to the simulation, creating an immersive experience. Figure 4.1 shows a diagram of this high level design.

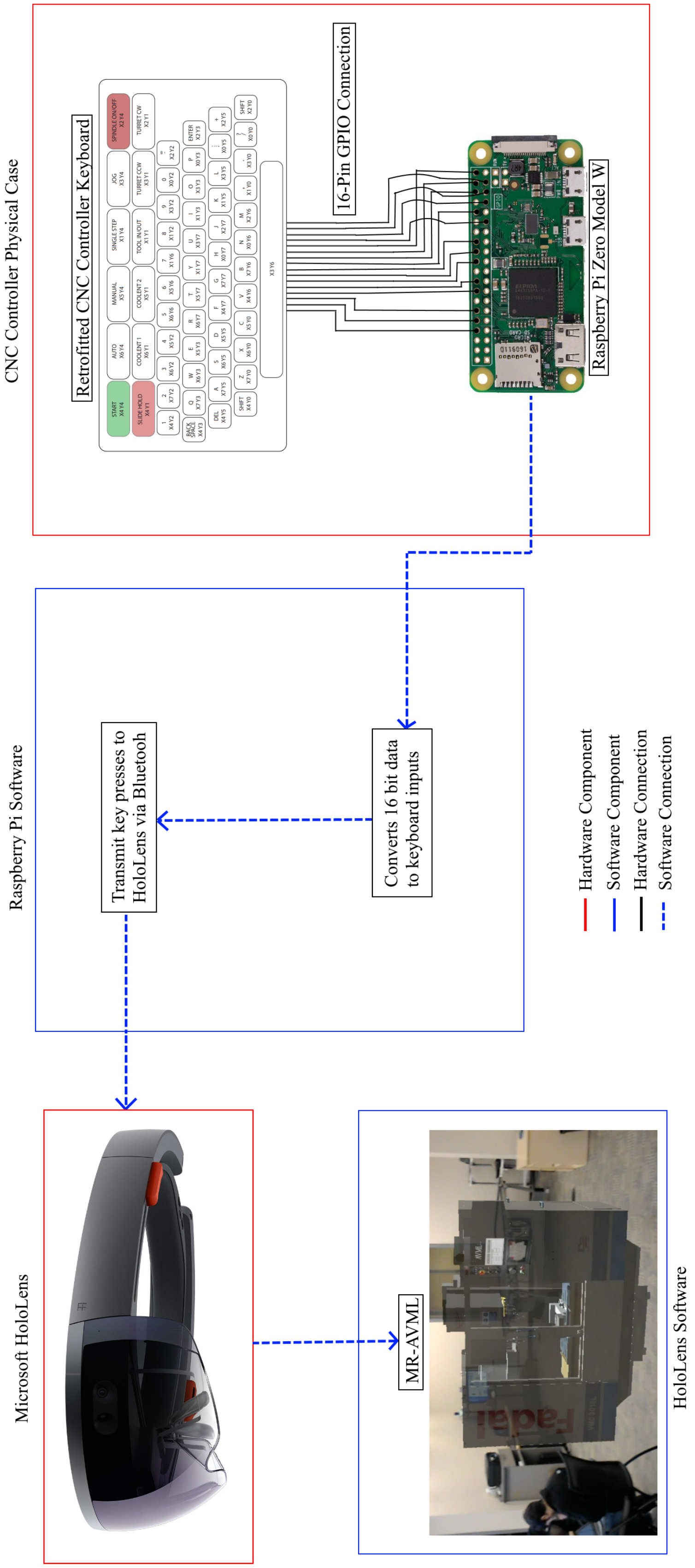

CNC Controller Physical Case
Retrofitted CNC Controller Keyboard
16-Pin GPIO Connection
Raspberry Pi Zero Model W
Raspberry Pi Software
Transmit key presses to HoloLens via Bluetooh
Converts 16 bit data to keyboard inputs
Hardware Component
Software Component
Hardware Connection
Software Connection
Microsoft HoloLens
MR-AVML
HoloLens Software

Fig. 4.1. High level diagram of the MR-AVML hardware.

The controller keyboard was retrofitted by disconnecting its wiring from the original controller motherboard and rewiring it to the Raspberry Pi Zero-Ws pins. The wiring consists of 16 pins in total: 8 pull up pins, labeled as X0 through X7; and 8 pull down pins, labeled as Y0 through Y7. Each of the 60 keys on the controller keyboard is connected to a unique permutation of two X and Y pins. Figure 4.2 shows a diagram showing each pin combination and their matching key.

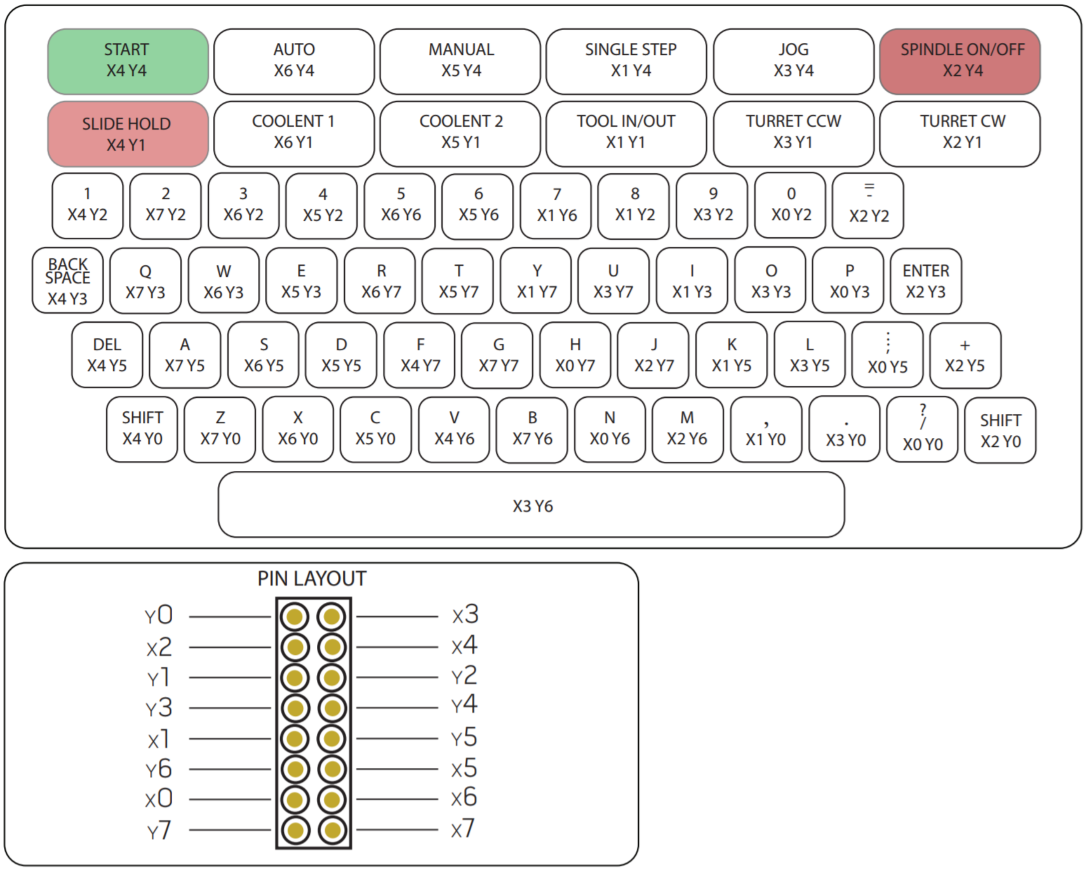


Fig. 4.2. Pin layout of the Fadal VMC-3016L keyboard.

When a key is pressed, the connection between that keys corresponding to the X and Y pins causes that combination to go low and high, respectively. The script on the Raspberry Pi monitors the pins for key presses by waiting for one of the Y pins to go high, and then checking each of the X pins for the one that went low. X pins are not checked prior to the detection of the high Y pin, since no X pin can go low without a Y pin going high. Once the X and Y pins are determined, the key pressed is selected via a lookup table, which translates the X and Y inputs to one character. This one character is then sent via Bluetooth to the HoloLens program.

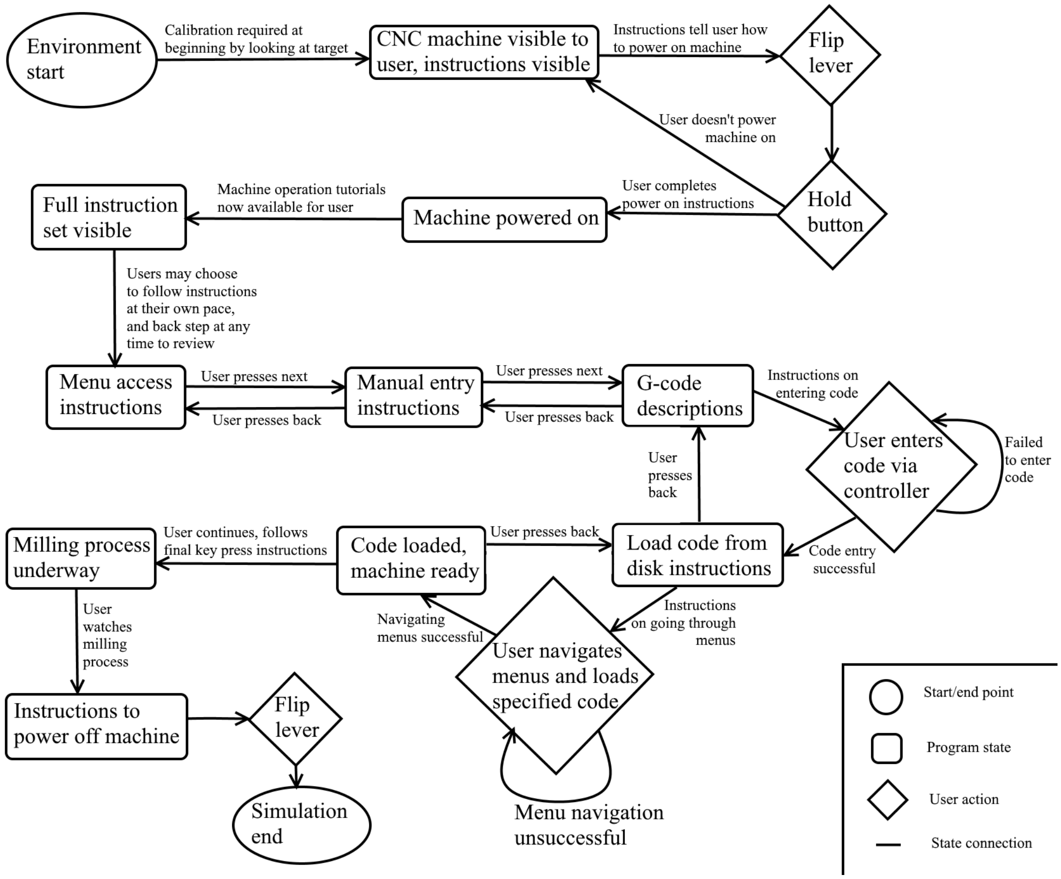

Environment start
Calibration required at beginning by looking at target
CNC machine visible to user, instructions visible
Instructions tell user how to power on machine
Flip lever
User doesn't power machine on
Hold button
User completes power on instructions
Machine powered on
Machine operation tutorials now available for user
Full instruction set visible
Users may choose to follow instructions at their own pace, and back step at any time to review
Menu access instructions
User presses next
User presses back
Manual entry instructions
User presses next
User presses back
G-code descriptions
Instructions on entering code
User enters code via controller
Failed to enter code
Code entry successful
User presses back
Load code from disk instructions
User presses back
Code loaded, machine ready
Instructions on going through menus
Navigating menus successful
User navigates menus and loads specified code
Menu navigation unsuccessful
User continues, follows final key press instructions
Milling process underway
User watches milling process
Instructions to power off machine
Flip lever
Simulation end
Start/end point
Program state
User action
State connection

Fig. 4.3. MR-AVML software flow diagram.

Several software systems work together in order to make up the MR-AVML simulation. The overall state machine process of the simulation is detailed in Figure 4.3. Internally, states are defined entirely by the user's progress through the tutorial. The states of the tutorial can be seen in Figure 4.3. The transcript of these tutorials follows:

1. Welcome to the AVML CNC Machine Simulation Test. In this brief demonstration you will be guided through some basic operations of the machine including milling a simple block of aluminum. Say ”NEXT” to continue.

2. In order to interact with the machine, you will use a combination of physical keyboard input and the cursor that tracks your head movement. Certain objects can be selected simply by looking at them. Say ”NEXT” to continue. Say ”BACK” to go back.

3. Let’s begin by powering ON the machine. The power junction cabinet is located on the back left side of the machine. This will feature a red lever with a small green button above it. Locate the power junction cabinet and place your cursor on the red lever until it rotates up. Then place your cursor on the green button until it presses inward. Return to the front of the machine once it has been powered ON. Say ”NEXT” to continue. Say ”BACK” to go back.

4. On the lower right side of the front of the machine you will be able to see the real physical CNC machine controller. A hologram screen overlays the real screen to display menu interfaces and useful information. Say ”NEXT” to continue. Say ”BACK” to go back.

5. The machine must now be initialized in order to mill parts. This involves navigating the Functions menus and loading the desired G-code module. From the start screen, press ”START” on the keyboard to compute the coordinate system for the drill. Now press ”MANUAL” to bring up the Quick Keys menu. Say ”NEXT” to continue. Say ”BACK” to go back.

6. The screen on the controller should now display the Quick Keys menu. This menu provides easy access to a number of commonly used operations. From the Quick Keys menu we can manually enter G-code. Press ”MANUAL” 2 times for G-code entry at runtime. Say ”NEXT” to continue. Say ”BACK” to go back.

7. G-code can now be typed out and processed to run the CNC machine. Let's test the following commands by typing them and pressing "ENTER": coolant ON (type "M08" "ENTER"); coolant OFF (type "M09" "ENTER"); spindle ON (type "M03" "ENTER"); spindle OFF (type "M05" "ENTER"); tool change (type "T1" "ENTER"). Say "NEXT" to continue. Say "BACK" to go back.

8. Let's return to the Quick Keys menu by pressing "MANUAL" then "AUTO". Now we will navigate to the Functions menu by pressing "SPACE". Say "NEXT" to continue. Say "BACK" to go back.

9. In order to access automated operations related to system memory, we must navigate to Disk. Press "0 (zero)" to access Disk Functions. From this menu, we can load G-code from a disk that has been previously imported on to the machine. To load a code file, press "1 (one)" for Disk to Memory. Say "NEXT" to continue. Say "BACK" to go back.

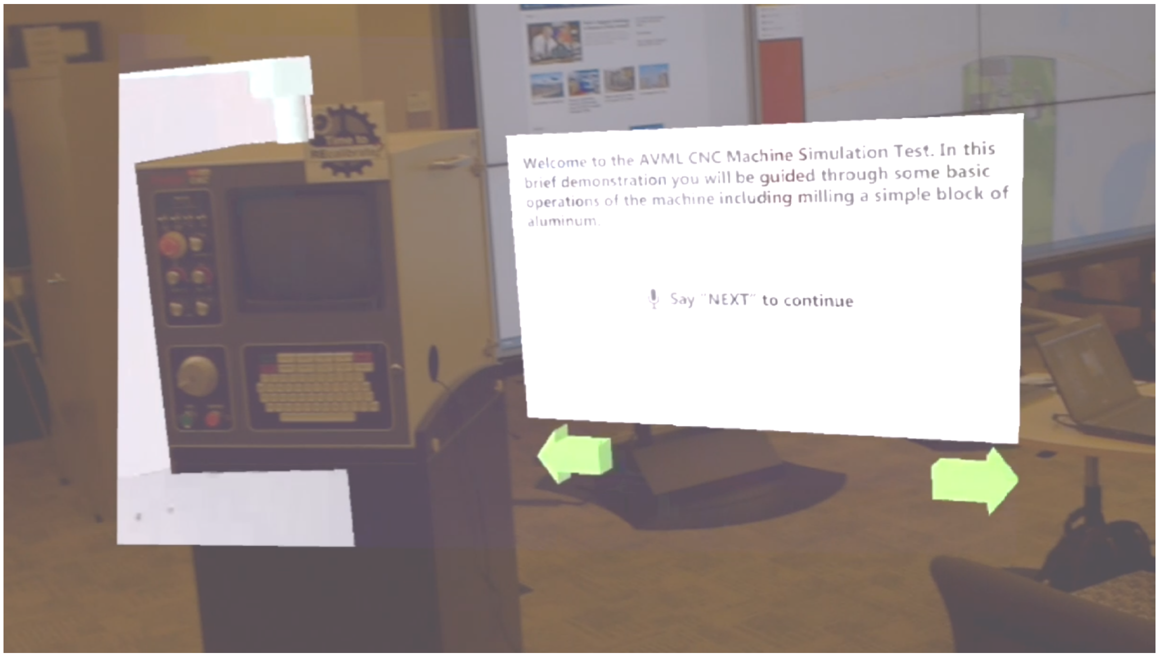
10. Now access the code file by first pressing "/?" then type the file name "a.txt". Press ENTER to load the file. Press "MANUAL" 3 times to navigate back to the main menu. Say "NEXT" to continue. Say "BACK" to go back.

11. To run the machine code file we just loaded, press AUTO. You should now see the machine perform a simple task of cutting a square into a block of aluminum. Say "NEXT" to continue. Say "BACK" to go back.

12. This concludes the AVML test. You may remove the headset now. Thank you for your participation.

The tutorial is the starting point in the MR-AVML. All systems are contextually based on the state defined by the tutorial. The tutorial only changes due to user input. Users are able to go forward or back in the tutorial in one of three ways: by using the HoloLens's voice recognition features and saying "next" or "back" out loud; by interacting with the forward and back arrow buttons inside the environment; or, during some steps, by completing the instructions provided in that tutorial step.

Several feedback methods are implemented in both the rendering and sound effects related to this tutorial, put in place because of the importance of the tutorial to the system. These will be discussed in their relevant sections.



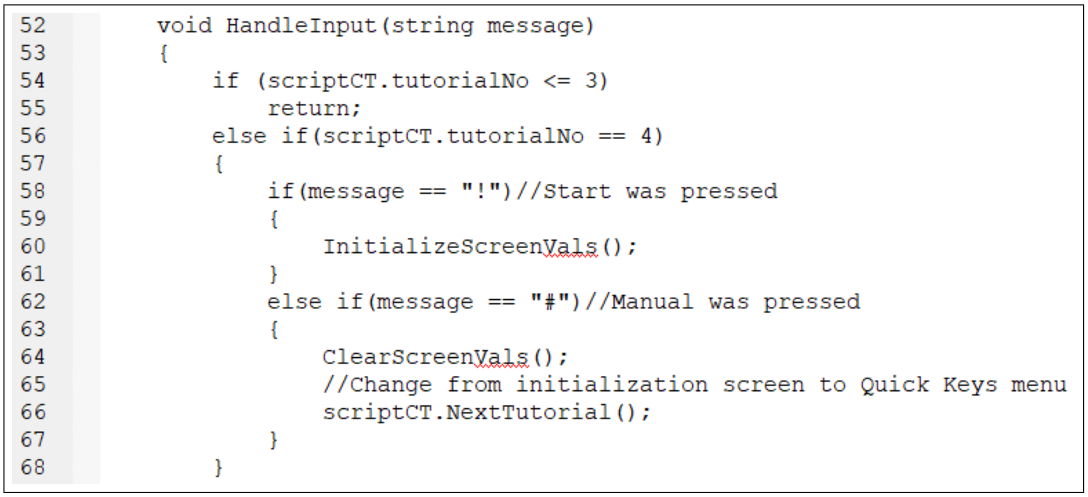
Welcome to the AVML CNC Machine Simulation Test. In this brief demonstration you will be guided through some basic operations of the machine including milling a simple block of aluminum.

Say "NEXT" to continue

Fig. 4.4. The tutorial hologram positioned next to the physical controller.

The second system receives the Bluetooth keyboard input. It parses the data it receives and, depending on the simulation's state, causes a specific reaction in the simulation environment. In all states prior to the ”Menu access instructions” state, all received keyboard inputs are ignored, as no inputs should cause actions while the virtual machine is in the ”powered off” state. Following this, only a few keys will cause reactions when pressed, with the majority still ignored in nearly all states. A piece of the code handling this input can be seen in Figure 4.5.

Menu navigation states, which consist of most available states in the simulation, only accept the inputs that will navigate them to other states in the simulation. This means the spacebar for moving between the main menu loop and the number keys for selecting submenus. The exception to this restriction is when the ”user enters code



```
    void HandleInput(string message)
    {
        if (scriptCT.tutorialNo <= 3)
            return;
        else if(scriptCT.tutorialNo == 4)
        {
            if(message == "!")//Start was pressed
            {
                InitializeScreenVals();
            }
            else if(message == "#")//Manual was pressed
            {
                ClearScreenVals();
                //Change from initialization screen to Quick Keys menu
                scriptCT.NextTutorial();
            }
        }
```

Fig. 4.5. Partial C# code for handling keyboard input in the MR-AVML.

via the controller” state and the ”user loads a specific code” state. During each of these states, all alphabet, number, and punctuation keys will display the pressed key in an input field. Pressing the ”delete” and ”backspace” keys cause the previously entered input to be removed from the input field string. The ”enter” key and ”Start” function key will cause the string entered in the input field to be interpreted as a manual G-code instruction execution - the process by which CNC instruction codes are entered line-by-line and executed as they are entered.

The third system in place handles all user input not related to the keyboard:

1. Voice recognition input.

2. Cursor input.

Both input methods are maintained by the HoloToolkit’s InputManager game object, a requirement for using the HoloLens’s features in Unity3D. Once voice input is recognized by the program (either ”next” or ”back”, as they are the only two words programmed with responses), custom code causes the tutorial instructions to either advance or go back, depending on the voice command received. Cursor input is handled using user gaze, rather than the traditional HoloLens method of recognizing the standard ”AirTap” gesture. Gaze was chosen over gesture control so gestures would

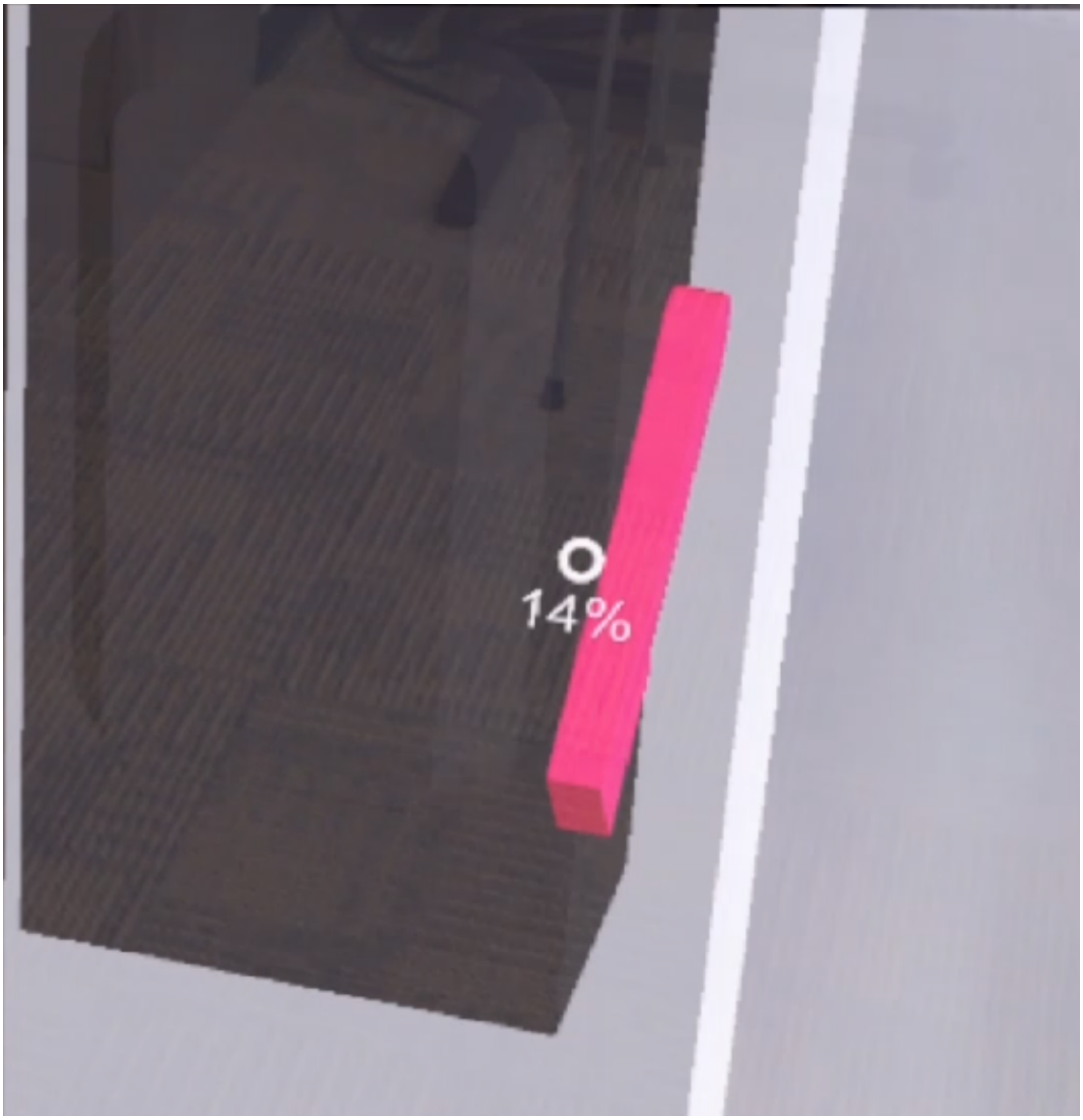
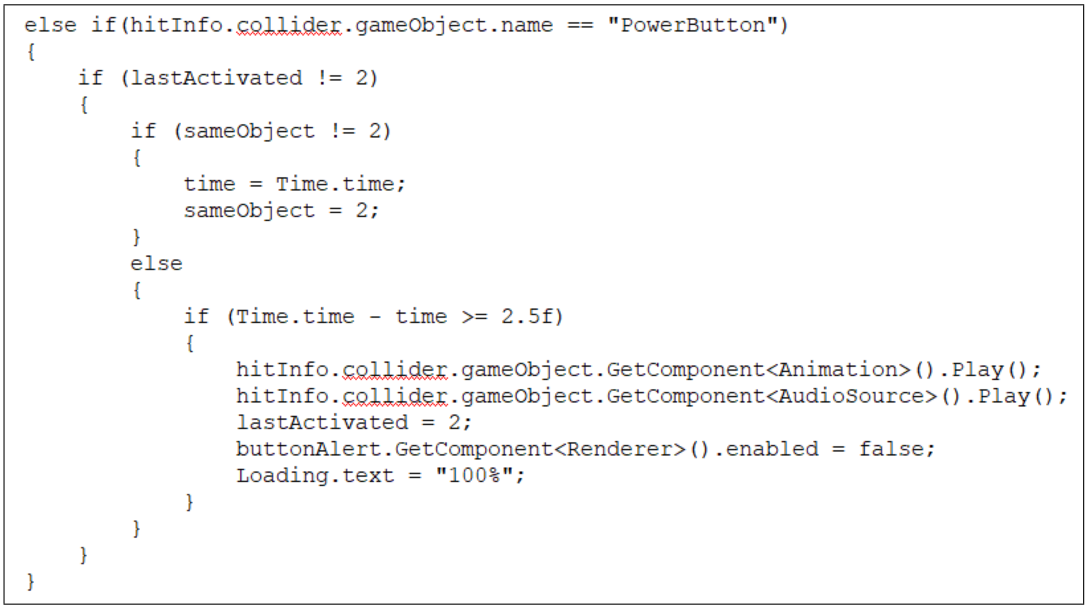


Fig. 4.6. An example of non-physical input method in the MR-AVML.

not be falsely recognized while the user interacts with the physical keyboard. Gaze is handled by sending a raycast from the user's "head" position to the first hologram that it comes in contact with. Upon contacting that hologram, the system checks if that hologram can be interacted with. This will be true if there is an animation or state-change action associated with the hologram in question. If interaction is possible, a timer starts to check how long the user holds their gaze onto the object. This is put in place to ensure that users don't accidentally cause an interaction by only briefly glancing at the hologram in question. After maintaining gaze on the hologram for long enough (1.5 seconds in most cases, 2.5 seconds in one case), the specified interaction occurs. Part of this code can be observed in Figure 4.7.



```
else if(hitInfo.collider.gameObject.name == "PowerButton")
{
    if (lastActivated != 2)
    {
        if (sameObject != 2)
        {
            time = Time.time;
            sameObject = 2;
        }
        else
        {
            if (Time.time - time >= 2.5f)
            {
                hitInfo.collider.gameObject.GetComponent<Animation>().Play();
                hitInfo.collider.gameObject.GetComponent<AudioSource>().Play();
                lastActivated = 2;
                buttonAlert.GetComponent<Renderer>().enabled = false;
                Loading.text = "100%";
            }
        }
    }
}
```

Fig. 4.7. Code from the C# script handling non-physical input.

Only four objects in the environment have interactions in this manner:

- The virtual power switch, which must be flipped to the "on" position before the machine will power on.
- The virtual power button, which must be held down until a "click" sound effect occurs while the power switch is flipped "on" to power up the machine.
- The virtual forward and back arrow buttons, which perform the same function as the "next" and "back" voice commands, respectively.

The virtual buttons for going forward and back in the tutorial were implemented as an alternative to voice commands for users who either have difficulty speaking English clearly and for users who are unable to speak loudly enough compared to their environment for the HoloLens microphone to detect their voice. With these input systems in place alongside the keyboard input manager, the user is able to control all aspects of the environment they need to in order to go through the full tutorial instruction set.

The fourth system handles the environment rendering. The machine hologram itself spawns in relative to the starting height and spatial position of the HoloLens when beginning the program. This is standardized by placing the HoloLens in the same position in the room prior to starting the program, then having the user put on the device after the environment initializes. The initialization of the program was handled remotely via a laptop paired to the HoloLens. This laptop was additionally used to monitor the user's actions in the environment. From this point, the virtual machine is fully visible to the user and is anchored to its spawn position, allowing the user to physically move around the environment to see the back of the machine. A mask and stencil effect is put in place in the environment to allow the physical controller to be seen as part of the environment, rather than as part of the physical background.

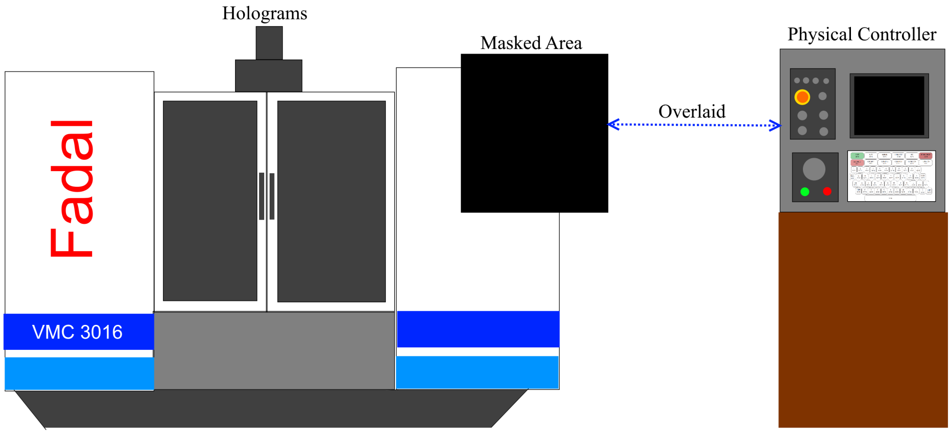


Fig. 4.8. The high-level diagram of the hologram overlaid on top of the controller.

The tutorial instructions are rendered directly next to the position of the physical controller. This position is static, meaning the tutorial can only be seen when the user is standing at the front of the machine. In the VR-AVML, the tutorial maintains its position in the lower third of the users vision at all times [18]. This design was decided against for the MR-AVML because the HoloLens's limited field of view. Currently, this field spans 35 degrees [19] and would have made the MR-AVML tutorial too small to read. Moreover, little space would have been left for the rest of the environment if more space is used for the instructions. The static positioning was chosen to allow users to read the tutorial easily. Further, any additional renderings rely on user input and the progression of the tutorial. Animations are in place for both flipping the virtual power switch and pressing the virtual power button, but neither will play prior to user input activating them. Additionally, the virtual screen of the machine will not be visible to the user before powering on the machine via the previous two steps. As previously stated, items displayed on this virtual screen will change depending on the user's selection via the physical controller keyboard and the progression of the tutorial. The raycast discussed in the user input section is visualized via a circular cursor that was included as part of the standard HoloToolkit. To further aid in letting the user know when they are in the process of selecting something with the cursor, a percentage counter is displayed below the cursor. This allows the user to see how much longer they need to maintain their gaze in order to cause a reaction in the system. There are also several G-code functions that are visibly showcased depending on user input. These are explained below:

- Set of particle effects that represent the CNC machines flood coolant function, activated by the user entering M08 during manual entry and deactivated by entering M09.

- An animation showing the drill bit spinning (as it would during milling), turned on by entering M03 during manual entry and turned off by entering M05.

- An animation for tool changing, activated by entering T followed by the tool number the user wishes to change to. For this MR-AVML iteration, only T1 causes a tool change, as it is the only tool available. Any other tool selection displays a notification on the virtual screen that the selected tool has not been initialized.

- A sequence of animations that plays during the "Milling process running" state, resulting in a milled piece.

The piece itself is "milled" by initially being rendered as a full piece with several additional game objects added on it, which are deleted from the environment instance as the drill bit passes over them. The purpose of each of these renderings is twofold: it emulates the actions of the actual Fadal VMC-3016L machine, while letting the user know that they have performed their intended action successfully. The second purpose is particularly important, as providing further context to actions makes the environment more natural, and therefore more immersive [5].

Aiding in providing users with an indication that their actions were performed successfully is the software that handles sounds in the environment. This is the final system of the proposed MR-AVML. Sound effects only play in the environment as a reaction to user input.

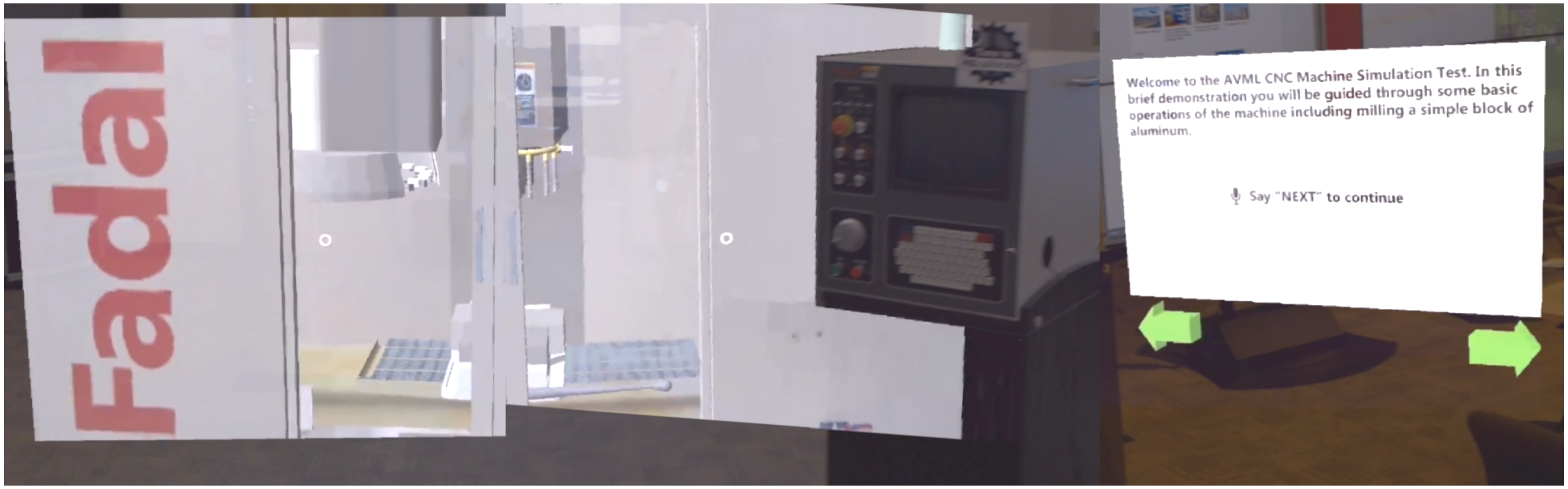
- When the power switch flips on, a sound effect plays to indicate it slotting into position.

- A click plays to indicate the machine has powered on after holding the power button down for a sufficient amount of time.

- When the machine begins executing loaded code and milling the piece, a whirring sound file plays to let the user know the main process was fully completed.

These sound effects are a nice indicator, but the most important sound effect plays when the user advances the tutorial instructions. As the tutorial can be advanced by following the instructions in that tutorial, rather than exclusively by the user manually advancing it, it is important to alert the user in some way that the environment has changed. This is handled by playing a page turn sound effect every time the tutorial changes - either by manual advancement or by following its instructions. As the first several steps in the tutorial can only be advanced manually, the user is given time to associate the sound with the tutorial changing. That way, when the sound effect plays outside of the context of manual tutorial advancement, the user immediately knows what in the environment changed. This type of feedback was imperative to include in the MR-AVML due to the previously discussed positioning of the tutorial. As the instructions can potentially be off screen when a change occurs, an audio cue allows users to always know when the change has occurred, whether they are actively looking at the tutorial or not.

Appendices A, B, and C include the main three C# classes used in the MR-AVML. Namely, BluetoothInput.cs, CustomGazeHandler.cs, and ChangeTutorial.cs are presented in full.



Welcome to the AVML CNC Machine Simulation Test. In this brief demonstration you will be guided through some basic operations of the machine including milling a simple block of aluminum.

Say "NEXT" to continue

Fig. 4.9. Full MR-AVML machine hologram with controller and tutorial.

5. ASSESSMENT DESIGN

In order to assess the pedagogical capabilities of the MR-AVML, a study was designed to match the study conducted in [18]. By duplicating this study with the MR-AVML substituted for the VR-AVML, it will be possible to compare the results of the two studies. This comparison will help determine the extent of the benefits of the mixed reality input methods and the haptic feedback added. In this chapter, the setup of the MR-AVML training session, the usability study, and the pedagogical study will be detailed.

5.1 Training Session Setup

To maintain parity with the VR-AVML study [18], the MR-AVML was designed to provide the same information. A transcript of the VR-AVML tutorials follows:

1. Welcome to the CNC Machine Simulation. In this demonstration, you will be walked through a few basic operations of the machine, and will machine a simple piece. To continue this tutorial, right click on the mouse.

2. To move around the environment, you will use the keys: [W], [A], [S], [D] to move, [X] and [Z] to adjust your height. You can look around the environment by turning your head and by moving the mouse. If you ever need to review a previous tutorial step, click the middle mouse button. Right click to continue the tutorial.

3. In order to complete this simulation, you will need to interact with the environment using the red crosshair that is in the center of your vision. By aiming this crosshair towards the desired object and left clicking on the mouse, you can interact with certain objects. Right click to continue.

4. The main object you will be interacting with is the CNC Machine's controller. This is a virtual keyboard that you will use to enter input. In order to differentiate between the virtual keyboard and the physical instructions, physical keys will be denoted with brackets. Right click again to continue.

5. To begin practicing interacting with the environment, we will first power on the machine. To the left of your starting position, and the left side of the CNC Machine, you will find the milling machine power junction cabinet. This will feature a red lever with a small green button above it. Right click once you have found the power junction cabinet.

6. To power the machine on, line up the crosshair with the lever and left click to flip it into the ON position. After this, aim to the green button, then click and hold for two seconds to power on the machine. This is the method to power on a CNC machine from complete shutdown. Right click once more to continue.

7. Now that the machine has been powered on, we will proceed to the front of it. From the front, you can see into the machine, where the metal block that we will machine in the next step is located. Right click again to continue the tutorial.

8. On the right side of the front of the machine is the CNC Machine controller. In addition to the keyboard, the controller has a screen that displays useful information, such as the current coordinates where the drill is located in centimeters, and a display of the instructions that you have typed so far. Right click to continue.

9. Instructions for the CNC Machine, commonly referred to as G-Codes, follow the format of a letter followed by a number, such as: X25 to advance the drill in the X plane. Multiple commands can be entered at once in some scenarios, such as: X25Y25 to move in the X and Y planes together. Right click to continue.

10. Other commands that work for this machine include: X, Y, and Z coordinates (X25Y25); Coolant (M7, M8, M9); Tool spinning (M3, M4, M5); Tool change (i.e. T5 for tool 5). These codes can be entered in numerous orders to create many different designs. Right click again to continue.

11. In order to run the machine, we will now need to initialize the machine, navigate through the Functions menus, and load the appropriate G-Code. After loading, we will run the G-Code and machine a simple piece. Right click to continue.

12. The first step in running the machine is to initialize the coordinate system of the CNC machine's drill. From the start screen, you can press the green "Start" button in order to accomplish this. Press the "Start" button to proceed.

13. Now that the coordinates have been initialized, we can move from this startup screen to the Quick Keys menu by pressing the "Manual" key. Press the "Manual" key in order to continue.

14. Before we proceed further into menu navigation, let's take a look at some manual entry commands in order to get a better understanding of what the machine will be doing once it begins executing a program. Press "Manual" twice to go to manual entry.

15. We will now enter some commands to perform some of the CNC machine's basic functions. First, we'll change the tool that is currently being used. Using the CNC machine controller's keypad, enter "T2H2" and press the green "Start" button.

16. Note that, during manual entry, the first command will require you to press "Start"; all others will only require you to hit "Enter". This is important to keep in mind. Now, let's turn on the machine's flood coolant by typing "M08" and pressing "Enter".

17. You can now see that the flood coolant has been turned on. Now, you can turn on the spindle, which causes the machine's drill to spin rapidly. This is what allows it to machine parts. Type in "M03" and press "Enter" to turn on the spindle.

18. Now the spindle should be turning. This is the state the machine should be in when running - the spindle on, the coolant running, and a proper tool selected. For now, let's set the machine back to its original state by turning off the spindle and coolant and resetting the tool. Type "T1H1M05M09" and press "Enter".

19. The coolant and spindle should now be off, and the tool should have changed back to the tool in the first slot. The machine is now in the state that it was when we first turned it on. Press "Manual" and then "Auto" to return to the Quick Keys menu.

20. The screen of the CNC machine controller now displays the Quick Keys menu. This menu provides easy access to a number of commonly used operations. The operation we will be using, however, is on the Functions menu. To navigate to the Functions menu, press the space bar on the CNC machine controller.

21. The Functions menu offers access to numerous options and operations that could not be found on the Quick Keys menu. One of these is the "Disk" option, which takes us to the menu for any function related to disks inserted into the machine. We can see that the "Disk" option is related to the number 0; Press the 0 key on the CNC machine controller to continue.

22. After pressing the 0 key, we are now in the Disk Functions menu. This menu gives access to any operations related to disks. From here, we can load G-Code from a disk that has been previously inserted into the machine. To load a code file, press the 1 key for "Disk to Memory".

23. From this screen, we can load G-Code from a file. In order to do so, we will need to type in the name of the file by first typing a "?" by pressing the "Shift" key followed by the "/?" key on the CNC machine controller. Then, access the file by typing in "a.txt", the file name. After the name has been typed, press the "Enter" key.

24. The code has now been successfully loaded from the file. In order to run the code, we will need to navigate back to the main screen of the controller. Press the "Manual" key 3 times to go back.

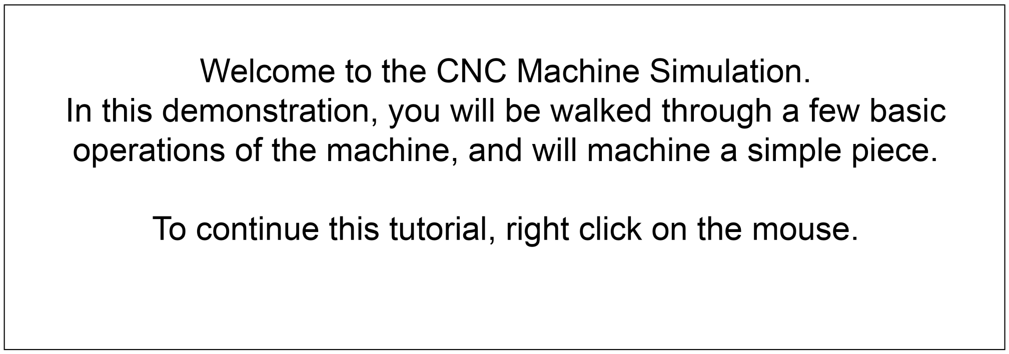
25. We are now on the screen for manually entering commands. However, since we have already loaded a series of G-Code commands into memory, we do not need to enter anything. Instead, we will run the code we have loaded by pressing the "Auto" key. Press the "Auto" key to continue.

26. The machine has now began drilling into the metal block placed inside the CNC machine's base. It will continue to machine this piece until it has finished the design. After it has finished, or whenever you are ready, you may conclude this demo by powering off the machine. To do this, simply flip the lever used to power the machine on.

27. Additionally, at any point during operation, you can immediately stop the machine by pressing the "Emergency Stop" button. This is the big red button near the top left of the CNC machine controller. You may try doing this now or simply right click again to end this tutorial.

28. This concludes the AVML demo. You may now remove the Oculus Rift headset.

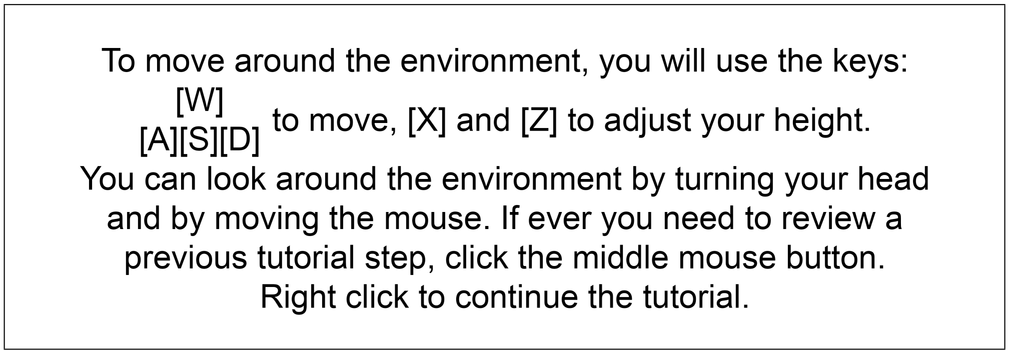
The texture files used in the VR-AVML are provided below. Note that some typos that were corrected in the transcript can be found in these textures. They are presented here as they were included in the VR-AVML study:



Welcome to the CNC Machine Simulation.
In this demonstration, you will be walked through a few basic
operations of the machine, and will machine a simple piece.

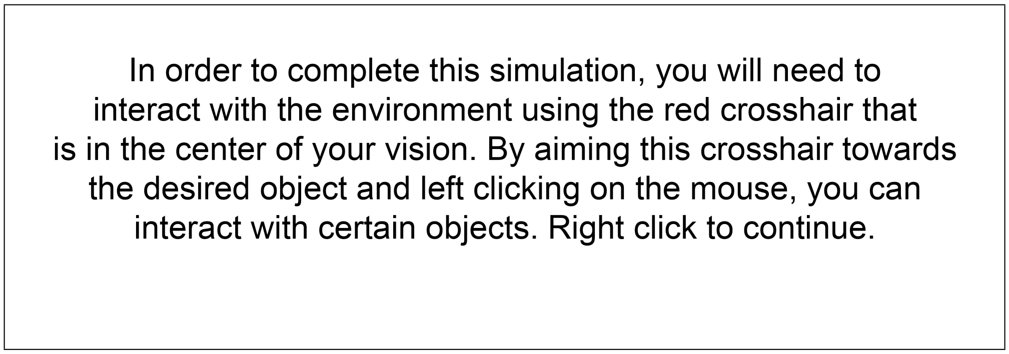
To continue this tutorial, right click on the mouse.

Fig. 5.1. Tutorial 1 from the VR-AVML.



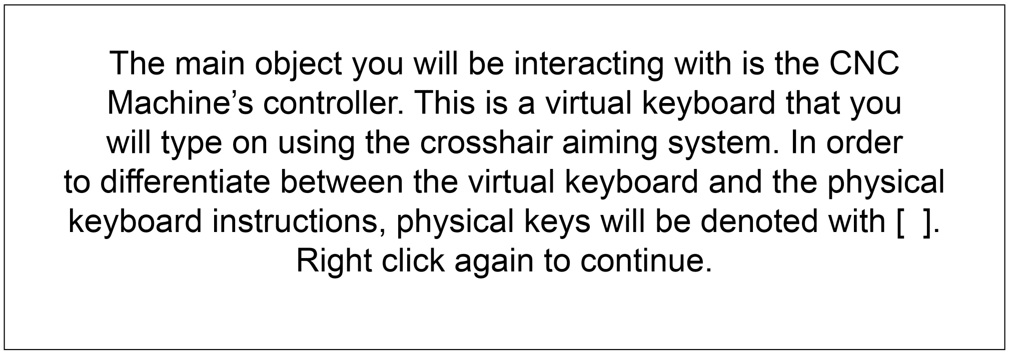
To move around the environment, you will use the keys:
[W]
[A][S][D] to move, [X] and [Z] to adjust your height.
You can look around the environment by turning your head
and by moving the mouse. If ever you need to review a
previous tutorial step, click the middle mouse button.
Right click to continue the tutorial.

Fig. 5.2. Tutorial 2 from the VR-AVML.



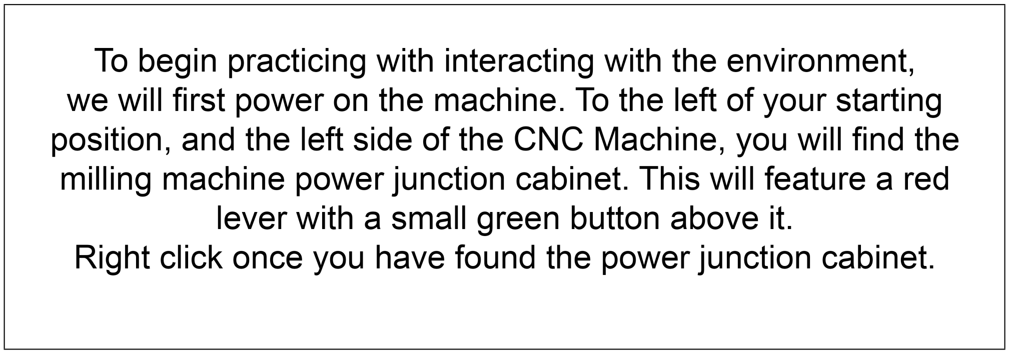
In order to complete this simulation, you will need to
interact with the environment using the red crosshair that
is in the center of your vision. By aiming this crosshair towards
the desired object and left clicking on the mouse, you can
interact with certain objects. Right click to continue.

Fig. 5.3. Tutorial 3 from the VR-AVML.



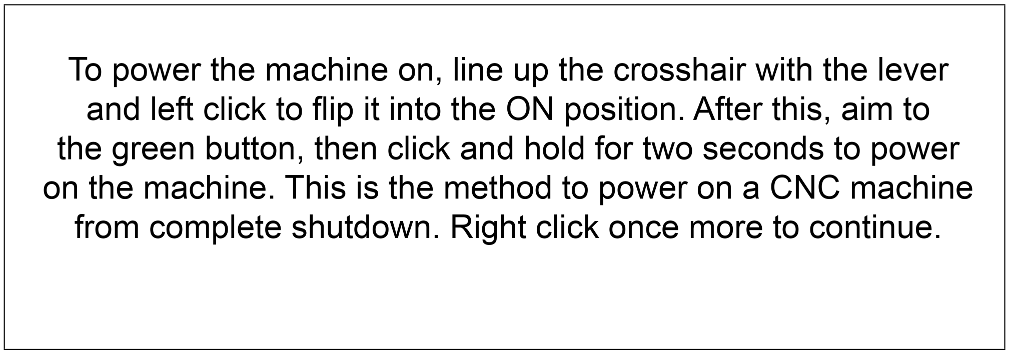
The main object you will be interacting with is the CNC Machine's controller. This is a virtual keyboard that you will type on using the crosshair aiming system. In order to differentiate between the virtual keyboard and the physical keyboard instructions, physical keys will be denoted with [ ]. Right click again to continue.

Fig. 5.4. Tutorial 4 from the VR-AVML.



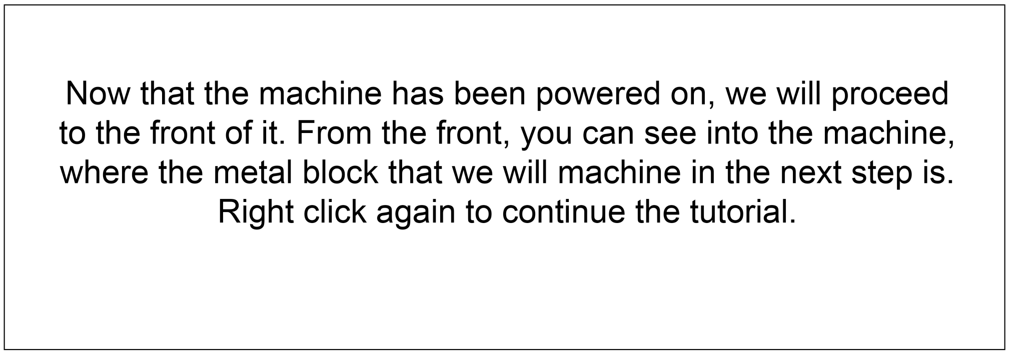
To begin practicing with interacting with the environment, we will first power on the machine. To the left of your starting position, and the left side of the CNC Machine, you will find the milling machine power junction cabinet. This will feature a red lever with a small green button above it.
Right click once you have found the power junction cabinet.

Fig. 5.5. Tutorial 5 from the VR-AVML.



To power the machine on, line up the crosshair with the lever and left click to flip it into the ON position. After this, aim to the green button, then click and hold for two seconds to power on the machine. This is the method to power on a CNC machine from complete shutdown. Right click once more to continue.

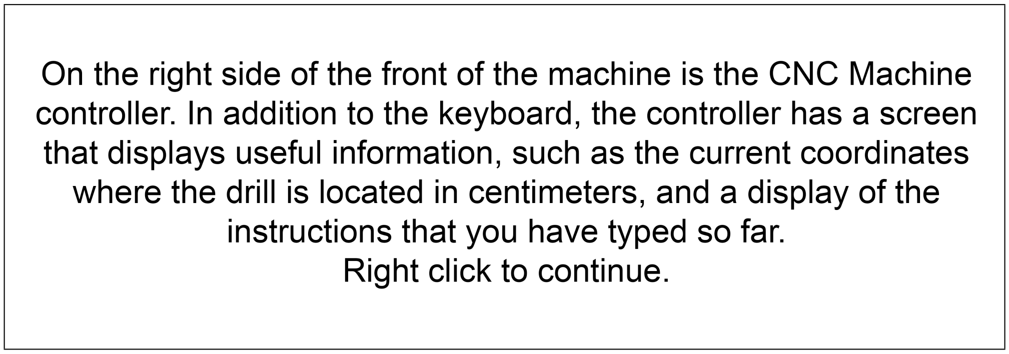
Fig. 5.6. Tutorial 6 from the VR-AVML.



Now that the machine has been powered on, we will proceed to the front of it. From the front, you can see into the machine, where the metal block that we will machine in the next step is.

Right click again to continue the tutorial.

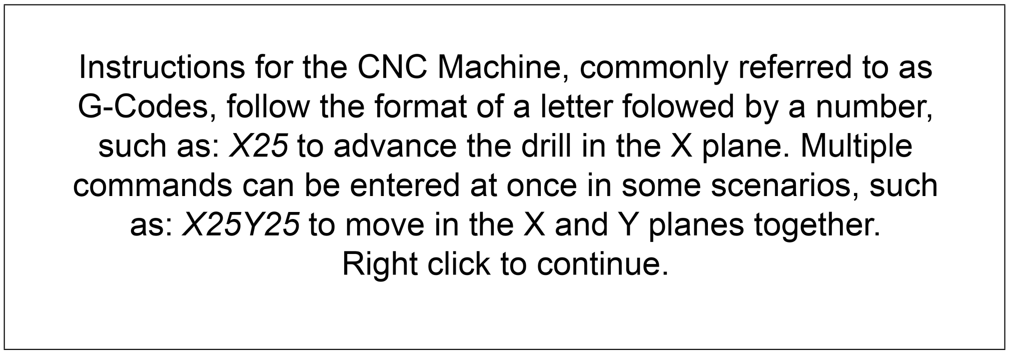
Fig. 5.7. Tutorial 7 from the VR-AVML.



On the right side of the front of the machine is the CNC Machine controller. In addition to the keyboard, the controller has a screen that displays useful information, such as the current coordinates where the drill is located in centimeters, and a display of the instructions that you have typed so far.

Right click to continue.

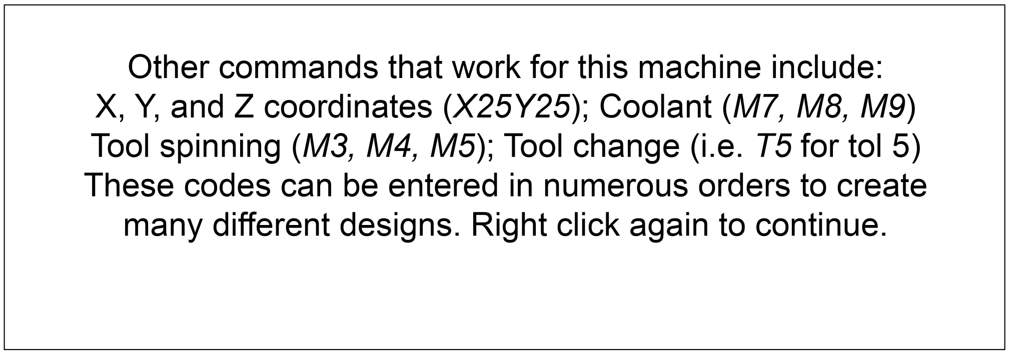
Fig. 5.8. Tutorial 8 from the VR-AVML.



Instructions for the CNC Machine, commonly referred to as G-Codes, follow the format of a letter folowed by a number, such as: *X25* to advance the drill in the X plane. Multiple commands can be entered at once in some scenarios, such as: *X25Y25* to move in the X and Y planes together.

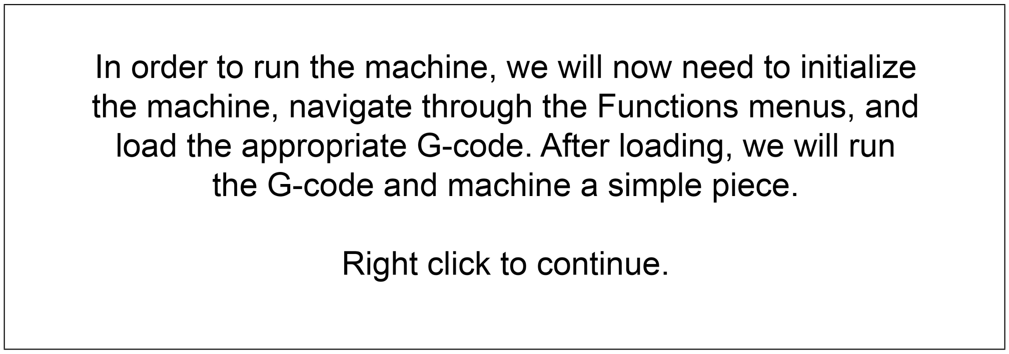
Right click to continue.

Fig. 5.9. Tutorial 9 from the VR-AVML.



Other commands that work for this machine include:
X, Y, and Z coordinates (*X25Y25*); Coolant (*M7, M8, M9*)
Tool spinning (*M3, M4, M5*); Tool change (i.e. *T5* for tol 5)
These codes can be entered in numerous orders to create many different designs. Right click again to continue.

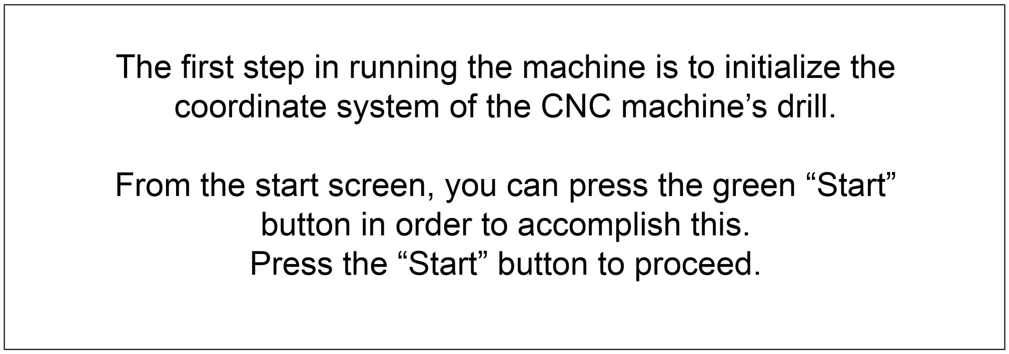
Fig. 5.10. Tutorial 10 from the VR-AVML.



In order to run the machine, we will now need to initialize the machine, navigate through the Functions menus, and load the appropriate G-code. After loading, we will run the G-code and machine a simple piece.

Right click to continue.

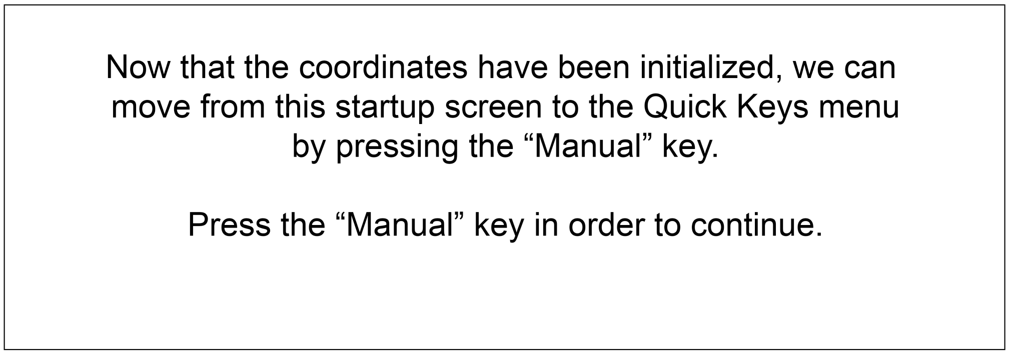
Fig. 5.11. Tutorial 11 from the VR-AVML.



The first step in running the machine is to initialize the coordinate system of the CNC machine's drill.

From the start screen, you can press the green "Start" button in order to accomplish this.
Press the "Start" button to proceed.

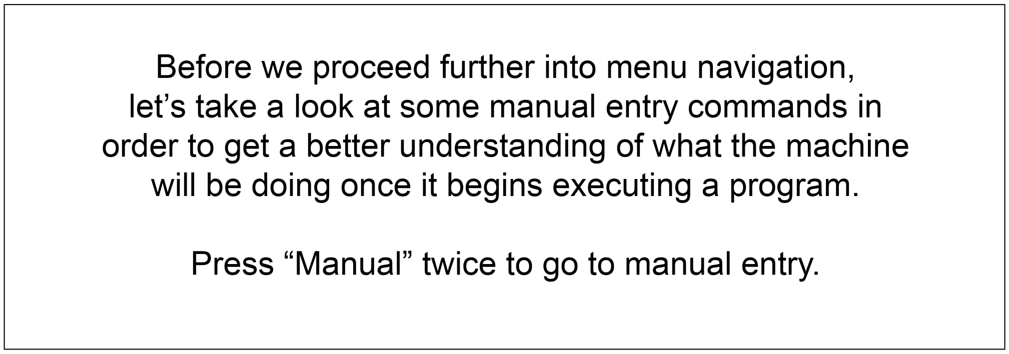
Fig. 5.12. Tutorial 12 from the VR-AVML.



Now that the coordinates have been initialized, we can move from this startup screen to the Quick Keys menu by pressing the “Manual” key.

Press the “Manual” key in order to continue.

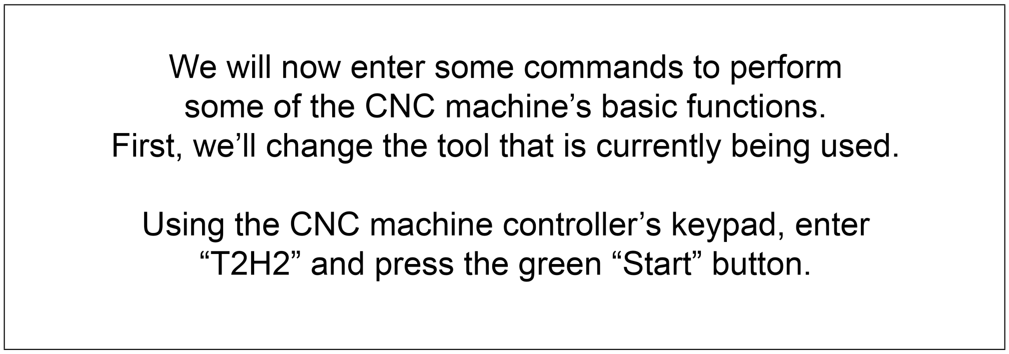
Fig. 5.13. Tutorial 13 from the VR-AVML.



Before we proceed further into menu navigation, let’s take a look at some manual entry commands in order to get a better understanding of what the machine will be doing once it begins executing a program.

Press “Manual” twice to go to manual entry.

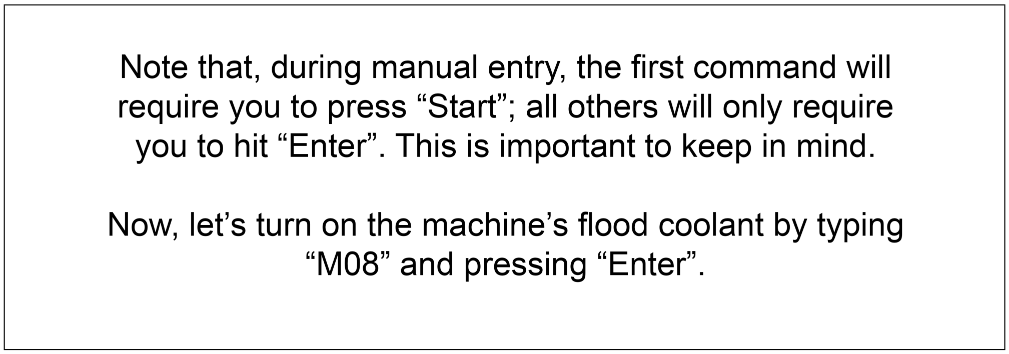
Fig. 5.14. Tutorial 14 from the VR-AVML.



We will now enter some commands to perform some of the CNC machine’s basic functions. First, we’ll change the tool that is currently being used.

Using the CNC machine controller’s keypad, enter “T2H2” and press the green “Start” button.

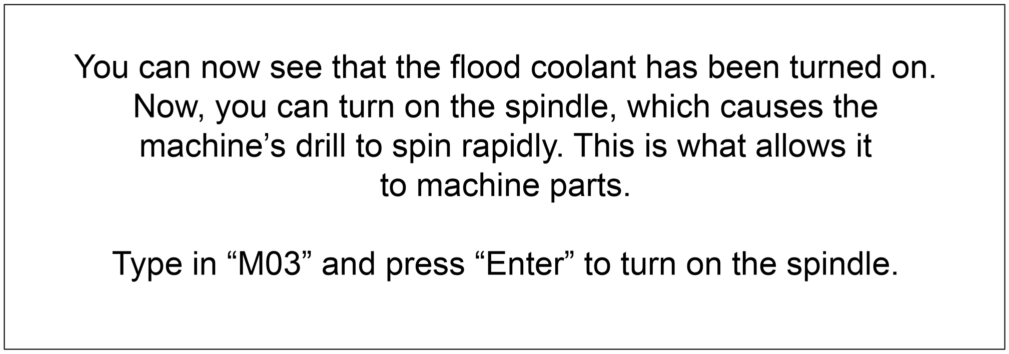
Fig. 5.15. Tutorial 15 from the VR-AVML.



Note that, during manual entry, the first command will require you to press “Start”; all others will only require you to hit “Enter”. This is important to keep in mind.

Now, let’s turn on the machine’s flood coolant by typing “M08” and pressing “Enter”.

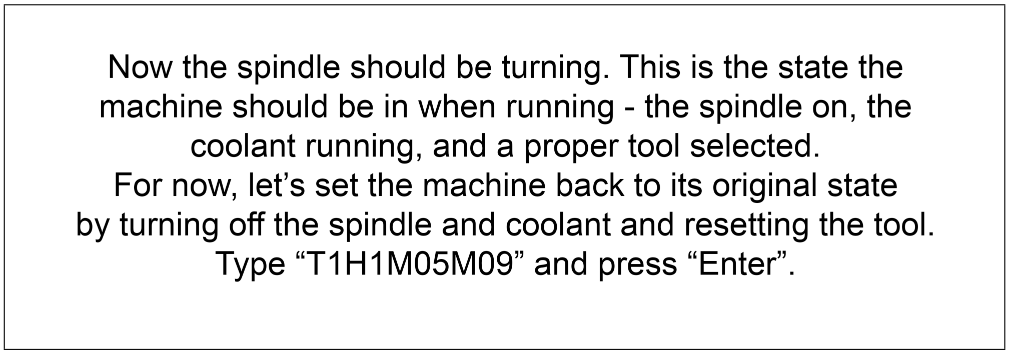
Fig. 5.16. Tutorial 16 from the VR-AVML.



You can now see that the flood coolant has been turned on. Now, you can turn on the spindle, which causes the machine’s drill to spin rapidly. This is what allows it to machine parts.

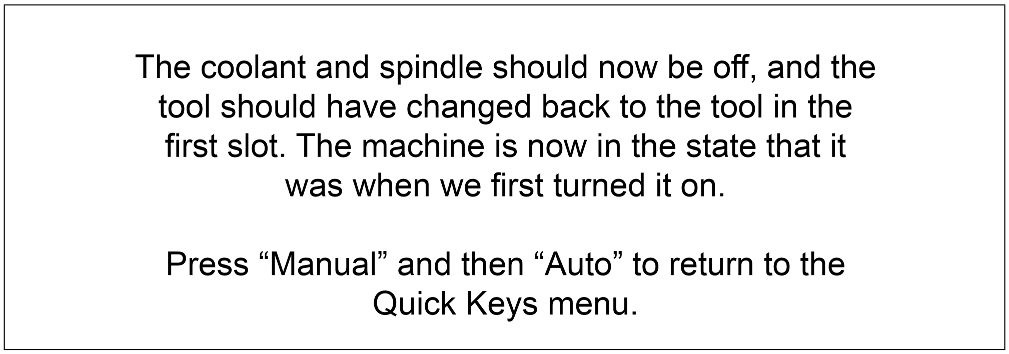
Type in “M03” and press “Enter” to turn on the spindle.

Fig. 5.17. Tutorial 17 from the VR-AVML.



Now the spindle should be turning. This is the state the machine should be in when running - the spindle on, the coolant running, and a proper tool selected.
For now, let’s set the machine back to its original state by turning off the spindle and coolant and resetting the tool.
Type “T1H1M05M09” and press “Enter”.

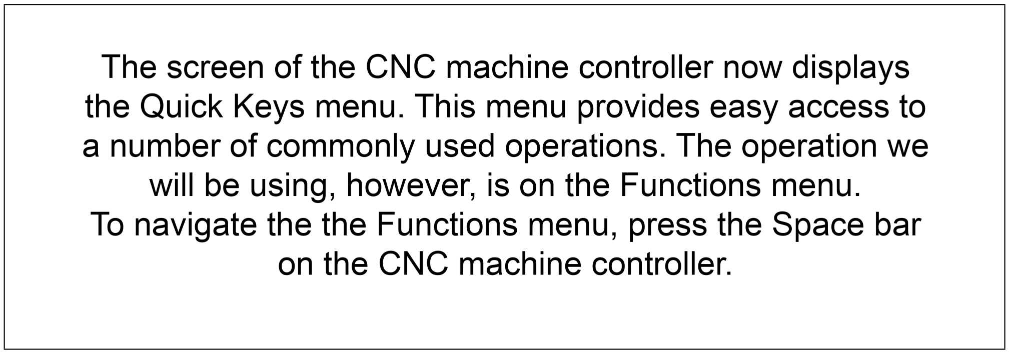
Fig. 5.18. Tutorial 18 from the VR-AVML.



The coolant and spindle should now be off, and the tool should have changed back to the tool in the first slot. The machine is now in the state that it was when we first turned it on.

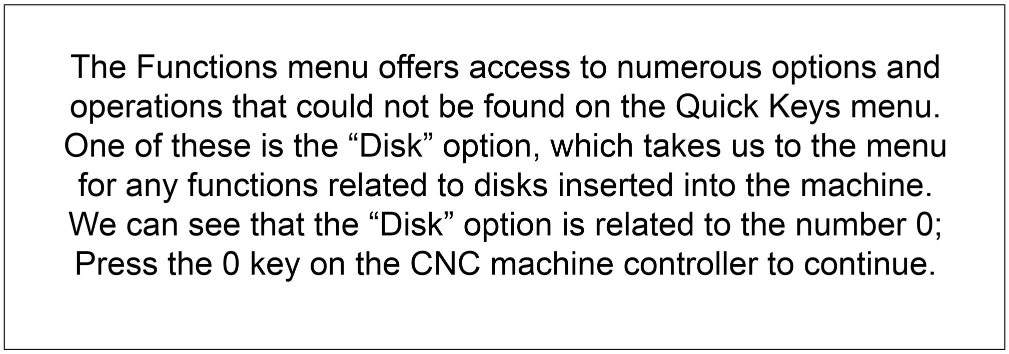
Press “Manual” and then “Auto” to return to the Quick Keys menu.

Fig. 5.19. Tutorial 19 from the VR-AVML.



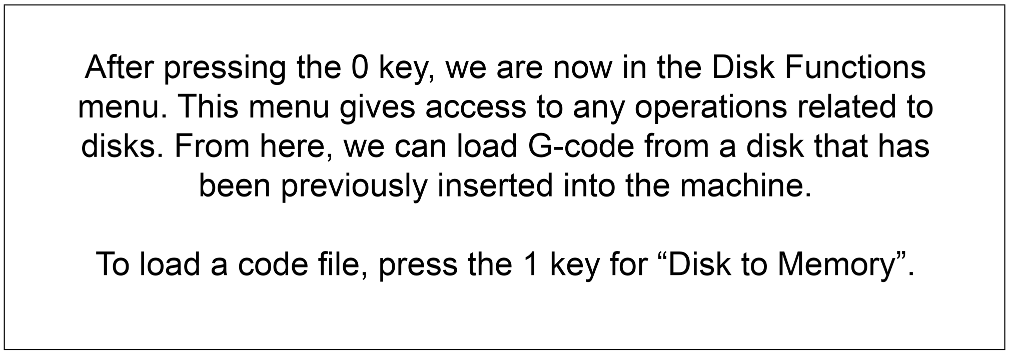
The screen of the CNC machine controller now displays the Quick Keys menu. This menu provides easy access to a number of commonly used operations. The operation we will be using, however, is on the Functions menu.
To navigate the the Functions menu, press the Space bar on the CNC machine controller.

Fig. 5.20. Tutorial 20 from the VR-AVML.



The Functions menu offers access to numerous options and operations that could not be found on the Quick Keys menu. One of these is the “Disk” option, which takes us to the menu for any functions related to disks inserted into the machine. We can see that the “Disk” option is related to the number 0; Press the 0 key on the CNC machine controller to continue.

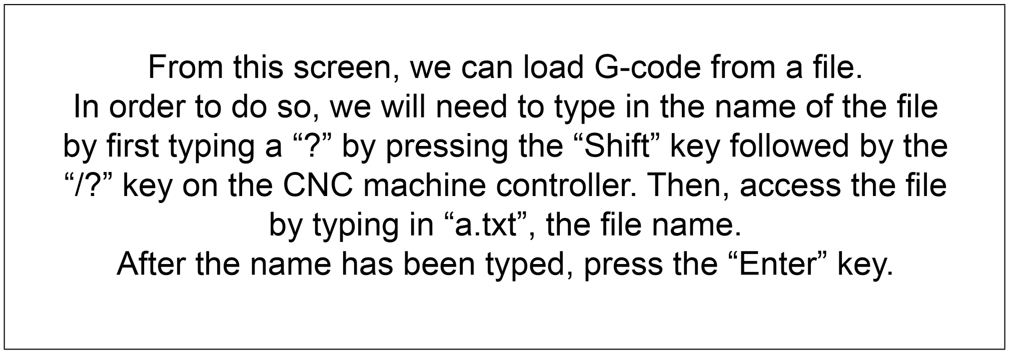
Fig. 5.21. Tutorial 21 from the VR-AVML.



After pressing the 0 key, we are now in the Disk Functions menu. This menu gives access to any operations related to disks. From here, we can load G-code from a disk that has been previously inserted into the machine.

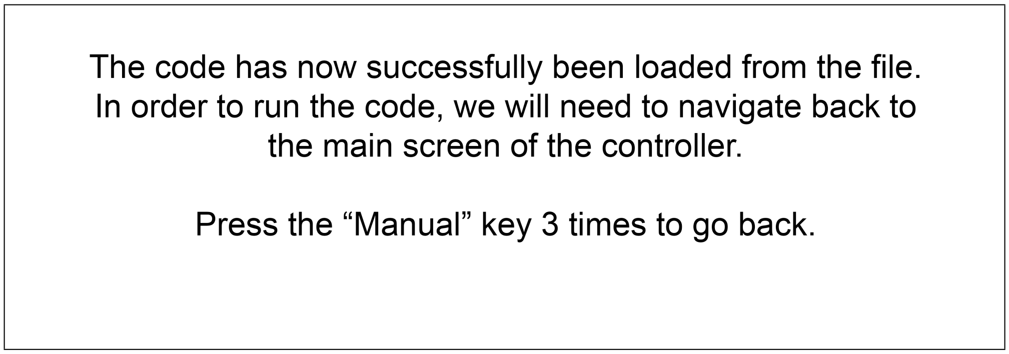
To load a code file, press the 1 key for “Disk to Memory”.

Fig. 5.22. Tutorial 22 from the VR-AVML.



From this screen, we can load G-code from a file.
In order to do so, we will need to type in the name of the file by first typing a “?” by pressing the “Shift” key followed by the “/?” key on the CNC machine controller. Then, access the file by typing in “a.txt”, the file name.
After the name has been typed, press the “Enter” key.

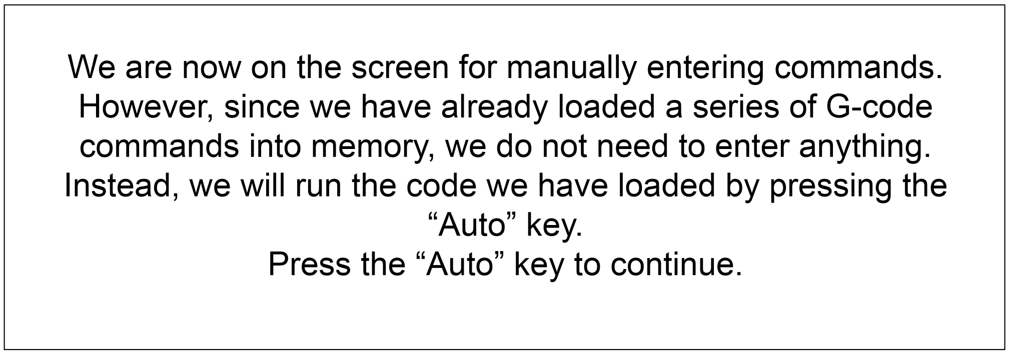
Fig. 5.23. Tutorial 23 from the VR-AVML.



The code has now successfully been loaded from the file. In order to run the code, we will need to navigate back to the main screen of the controller.

Press the “Manual” key 3 times to go back.

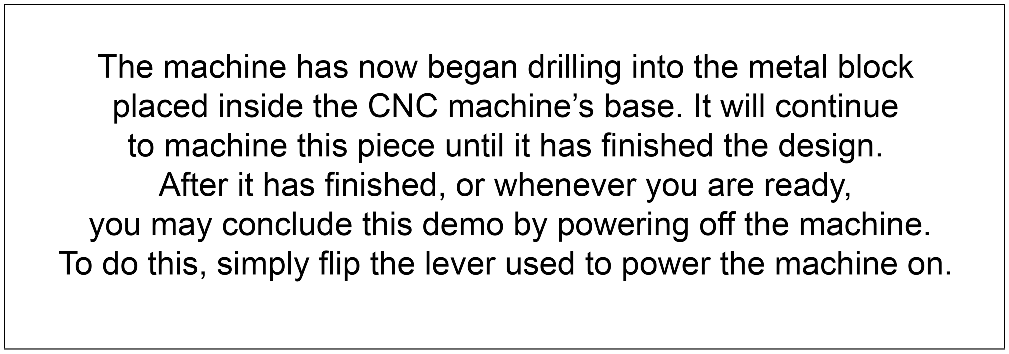
Fig. 5.24. Tutorial 24 from the VR-AVML.



We are now on the screen for manually entering commands. However, since we have already loaded a series of G-code commands into memory, we do not need to enter anything. Instead, we will run the code we have loaded by pressing the “Auto” key.

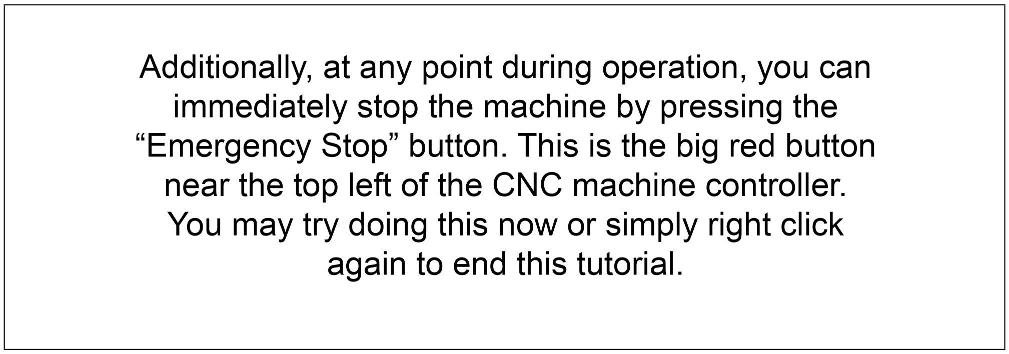
Press the “Auto” key to continue.

Fig. 5.25. Tutorial 25 from the VR-AVML.



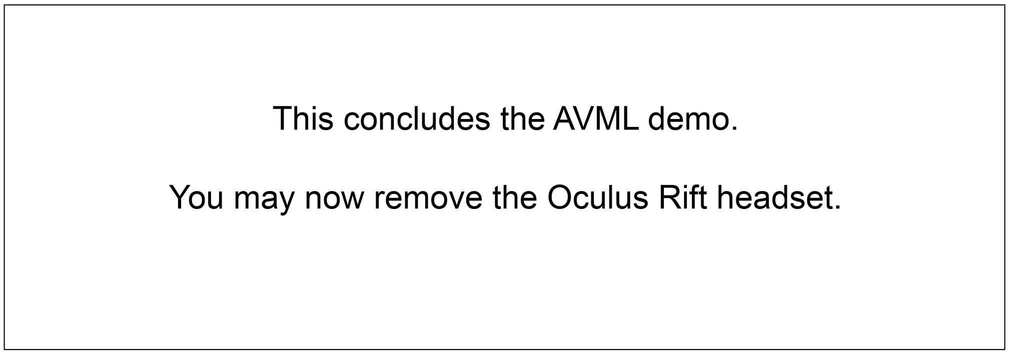
The machine has now began drilling into the metal block placed inside the CNC machine’s base. It will continue to machine this piece until it has finished the design. After it has finished, or whenever you are ready, you may conclude this demo by powering off the machine. To do this, simply flip the lever used to power the machine on.

Fig. 5.26. Tutorial 26 from the VR-AVML.



Additionally, at any point during operation, you can immediately stop the machine by pressing the “Emergency Stop” button. This is the big red button near the top left of the CNC machine controller. You may try doing this now or simply right click again to end this tutorial.

Fig. 5.27. Tutorial 27 from the VR-AVML.

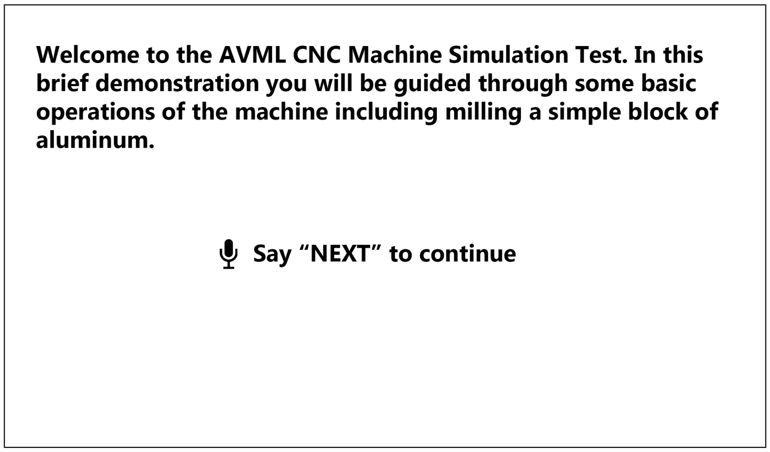


This concludes the AVML demo.

You may now remove the Oculus Rift headset.

Fig. 5.28. Tutorial 28 from the VR-AVML.

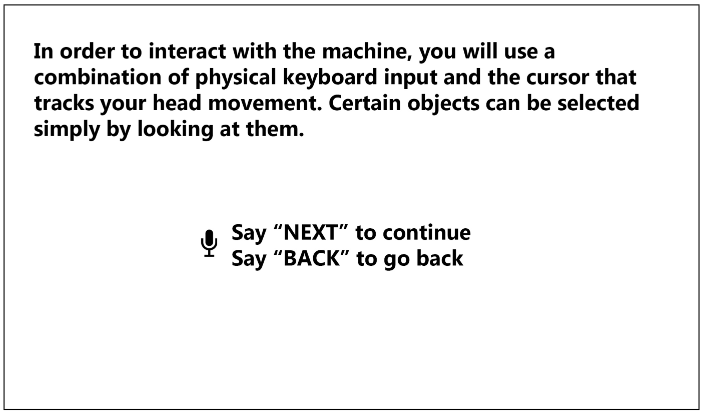
The exact wording of the original tutorial was changed, and some instructions were removed entirely. The only information removed, however, was related to movement controls and input methods specific to the VR-AVML, none of which applied to the MR-AVML. A series of manual entry slides were also condensed into a single slide to expedite the training process while providing the same basic information - providing users with a few G-code commands. The end result was twelve total tutorial slides. These tutorials are the same as what was listed in Chapter 4. The texture files used for the tutorials in the MR-AVML are presented below:



Welcome to the AVML CNC Machine Simulation Test. In this brief demonstration you will be guided through some basic operations of the machine including milling a simple block of aluminum.

Say "NEXT" to continue

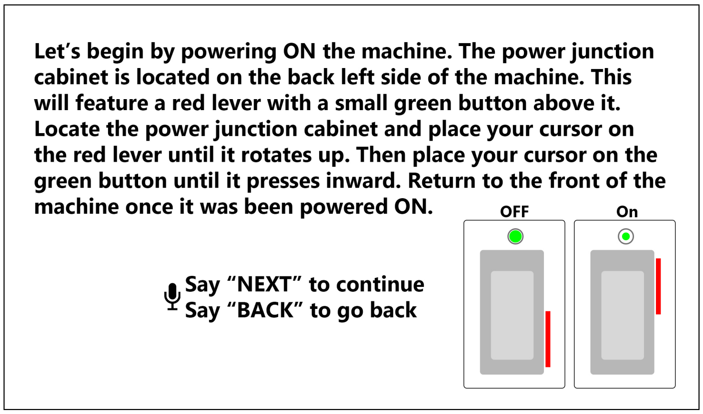
Fig. 5.29. Tutorial 1 from the MR-AVML.



In order to interact with the machine, you will use a combination of physical keyboard input and the cursor that tracks your head movement. Certain objects can be selected simply by looking at them.

🎙 Say “NEXT” to continue
Say “BACK” to go back

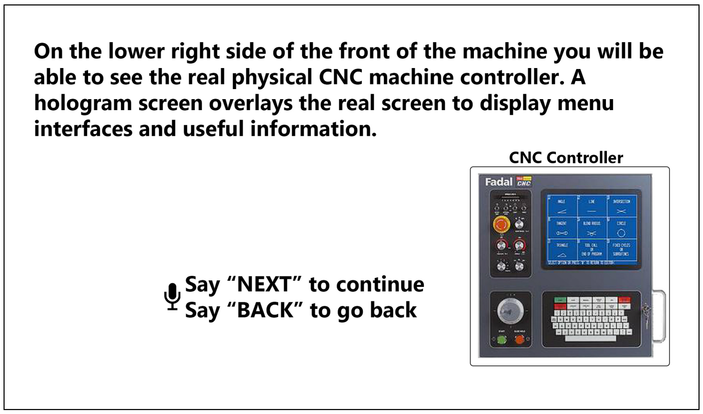
Fig. 5.30. Tutorial 2 from the MR-AVML.



Let’s begin by powering ON the machine. The power junction cabinet is located on the back left side of the machine. This will feature a red lever with a small green button above it. Locate the power junction cabinet and place your cursor on the red lever until it rotates up. Then place your cursor on the green button until it presses inward. Return to the front of the machine once it was been powered ON.

🎙 Say “NEXT” to continue
Say “BACK” to go back

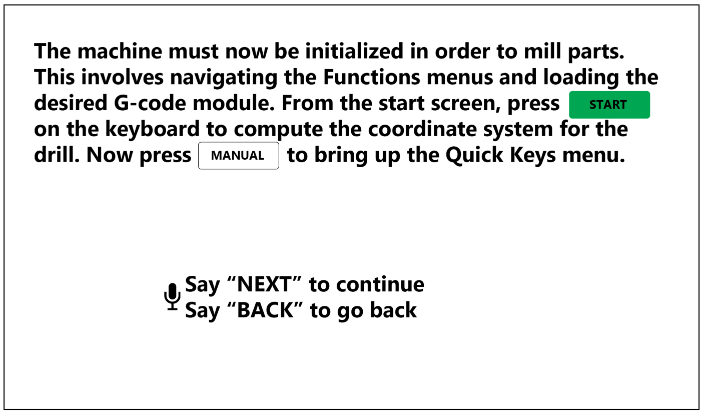
Fig. 5.31. Tutorial 3 from the MR-AVML.



On the lower right side of the front of the machine you will be able to see the real physical CNC machine controller. A hologram screen overlays the real screen to display menu interfaces and useful information.

🎙 Say “NEXT” to continue
Say “BACK” to go back

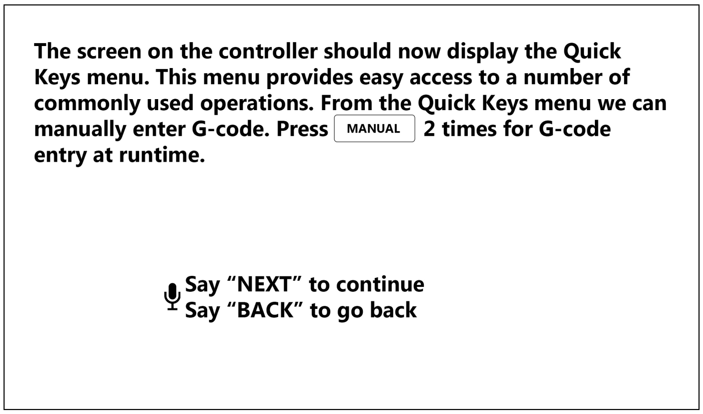
Fig. 5.32. Tutorial 4 from the MR-AVML.



The machine must now be initialized in order to mill parts. This involves navigating the Functions menus and loading the desired G-code module. From the start screen, press START on the keyboard to compute the coordinate system for the drill. Now press MANUAL to bring up the Quick Keys menu.

Say "NEXT" to continue
Say "BACK" to go back

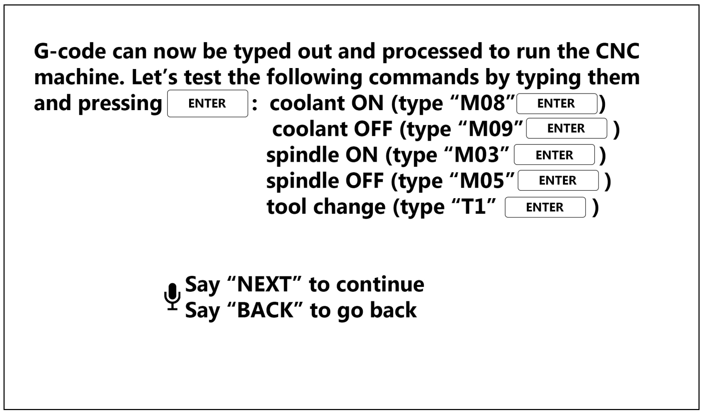
Fig. 5.33. Tutorial 5 from the MR-AVML.



The screen on the controller should now display the Quick Keys menu. This menu provides easy access to a number of commonly used operations. From the Quick Keys menu we can manually enter G-code. Press MANUAL 2 times for G-code entry at runtime.

Say "NEXT" to continue
Say "BACK" to go back

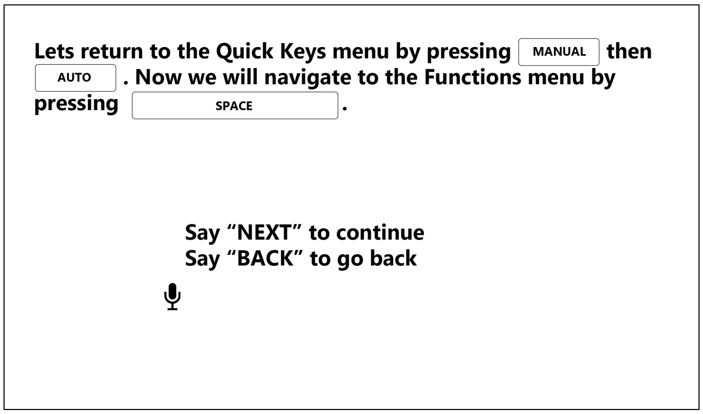
Fig. 5.34. Tutorial 6 from the MR-AVML.



G-code can now be typed out and processed to run the CNC machine. Let's test the following commands by typing them and pressing ENTER : coolant ON (type "M08" ENTER)
coolant OFF (type "M09" ENTER)
spindle ON (type "M03" ENTER)
spindle OFF (type "M05" ENTER)
tool change (type "T1" ENTER)

Say "NEXT" to continue
Say "BACK" to go back

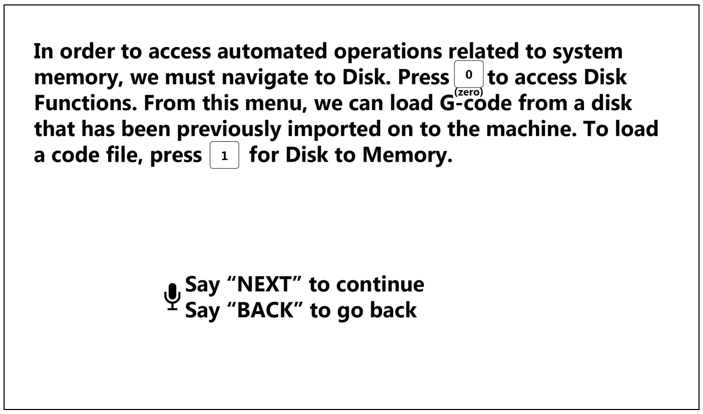
Fig. 5.35. Tutorial 7 from the MR-AVML.



Lets return to the Quick Keys menu by pressing MANUAL then AUTO . Now we will navigate to the Functions menu by pressing SPACE .

Say “NEXT” to continue
Say “BACK” to go back

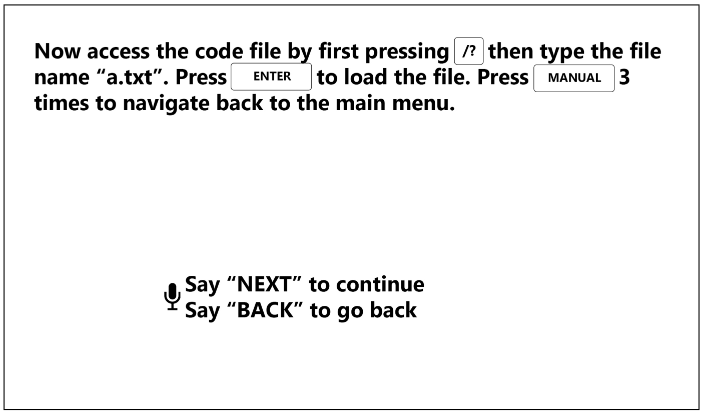
Fig. 5.36. Tutorial 8 from the MR-AVML.



In order to access automated operations related to system memory, we must navigate to Disk. Press 0 (zero) to access Disk Functions. From this menu, we can load G-code from a disk that has been previously imported on to the machine. To load a code file, press 1 for Disk to Memory.

Say “NEXT” to continue
Say “BACK” to go back

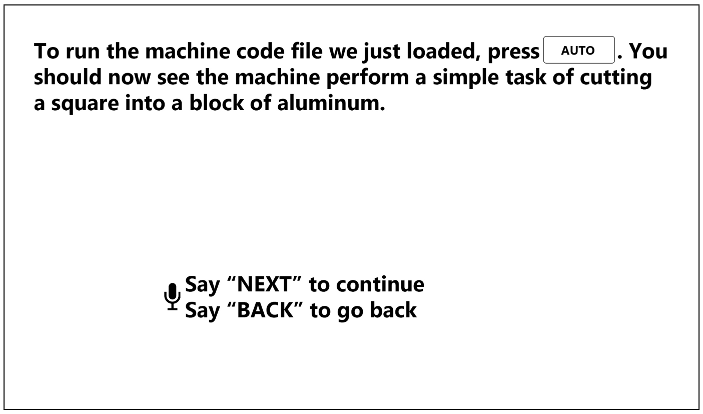
Fig. 5.37. Tutorial 9 from the MR-AVML.



Now access the code file by first pressing /? then type the file name “a.txt”. Press ENTER to load the file. Press MANUAL 3 times to navigate back to the main menu.

Say “NEXT” to continue
Say “BACK” to go back

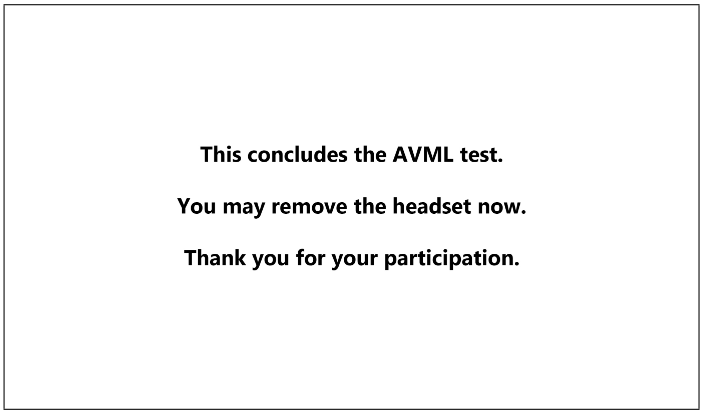
Fig. 5.38. Tutorial 10 from the MR-AVML.



To run the machine code file we just loaded, press AUTO. You should now see the machine perform a simple task of cutting a square into a block of aluminum.

🎙 Say “NEXT” to continue
Say “BACK” to go back

Fig. 5.39. Tutorial 11 from the MR-AVML.



This concludes the AVML test.

You may remove the headset now.

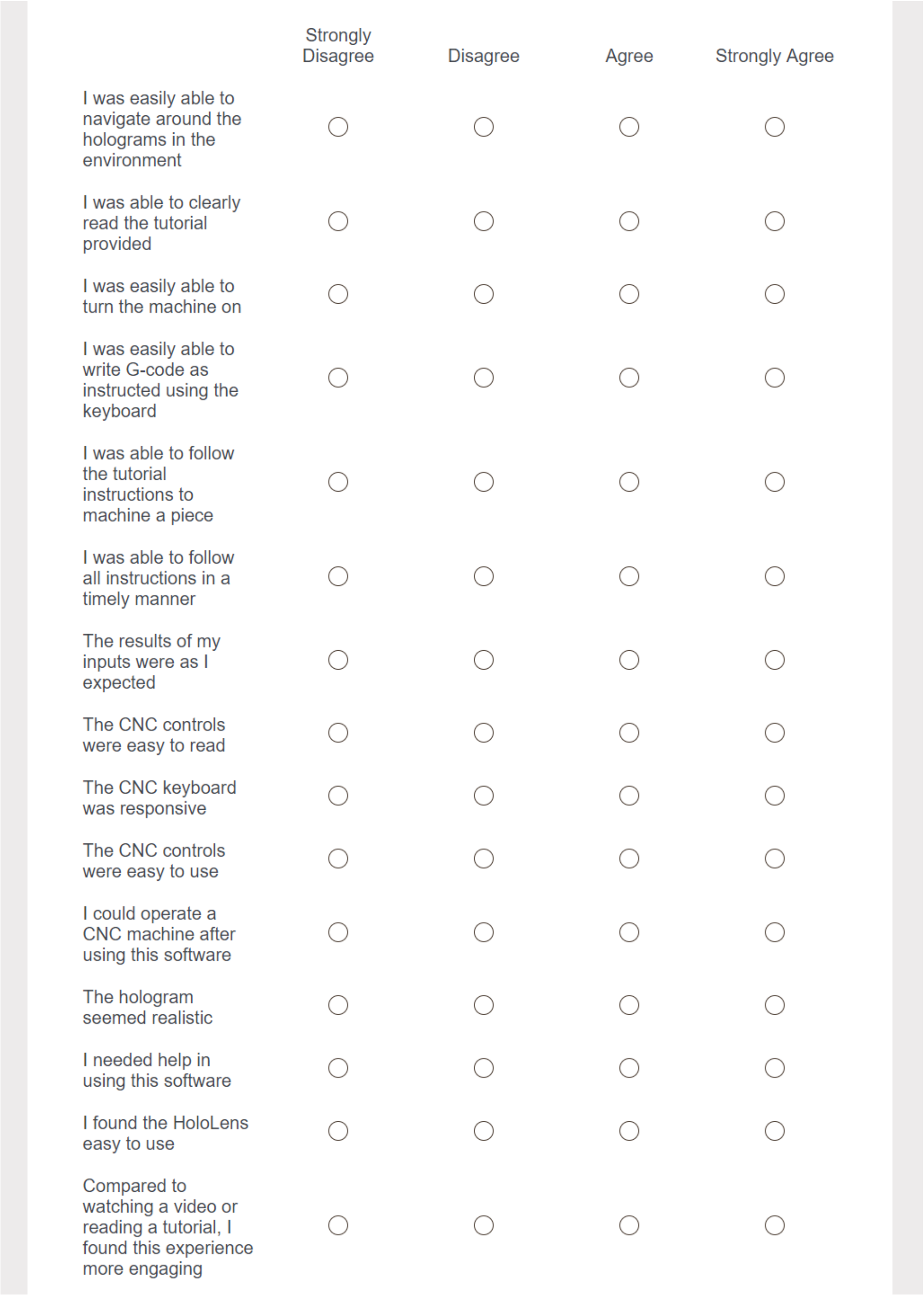
Thank you for your participation.

Fig. 5.40. Tutorial 12 from the MR-AVML.

Using these tutorial instructions for the MR-AVML demo, a training session was scheduled for a randomly selected half of the students in the spring 2018 section of the course ME 54600. The other half of the students received the same information through hands-on, instructor-led training. As for the MR-AVML group, each student was given time to go through the simulation individually. During the training, the progression of the students through the environment was monitored via a laptop projecting the environment rendered for the HoloLens. Monitoring was put in place in order to ensure that any major bugs encountered during the training session could be corrected. Beyond helping students when encountering bugs and assisting students with putting on the HoloLens at the beginning of the study, there was no further communication between the students and the researcher running the study. Each training session took between 7 and 10 minutes. Following the training, each student was asked to fill out a short survey to provide information for a usability study.

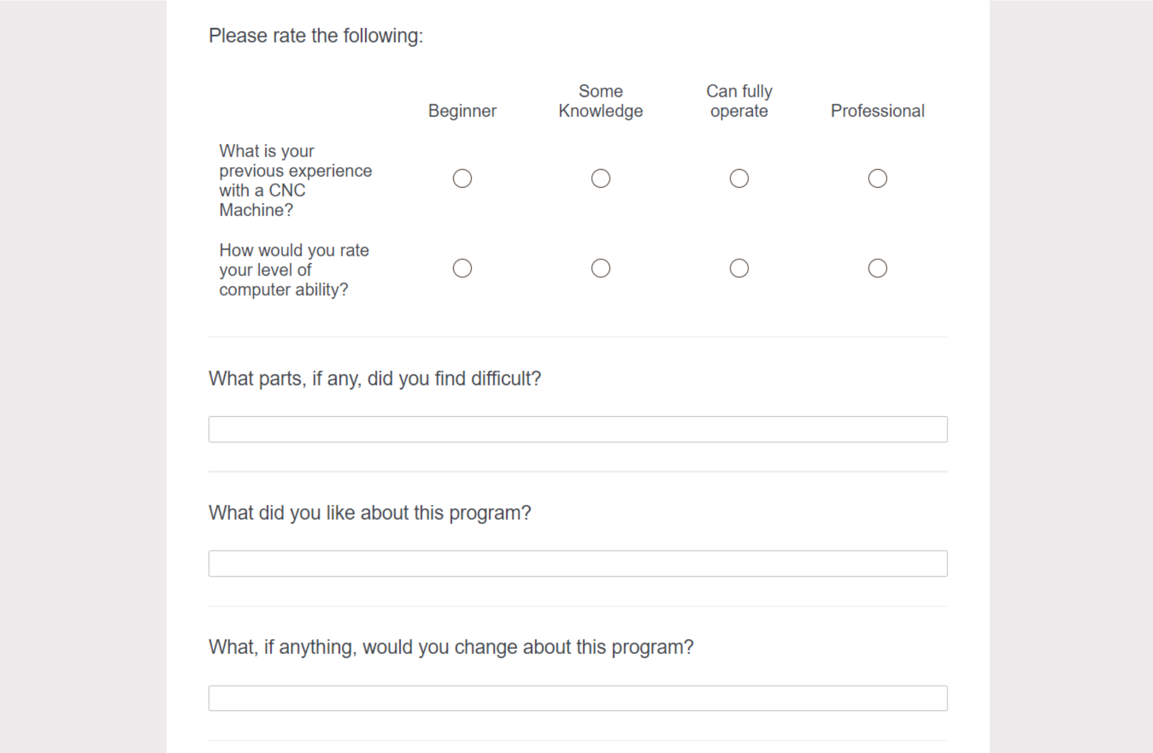
It's important to note the differences between the training sessions for the MR-AVML group and the instructor-led control group here. While the students in the MR-AVML group receive training individually with the HoloLens, the control group receives training instructions as a group. This should have a minimal effect on the results, but it is a key difference in the training between the two groups.

5.2 Usability Study Setup



	Strongly Disagree	Disagree	Agree	Strongly Agree
I was easily able to navigate around the holograms in the environment	○	○	○	○
I was able to clearly read the tutorial provided	○	○	○	○
I was easily able to turn the machine on	○	○	○	○
I was easily able to write G-code as instructed using the keyboard	○	○	○	○
I was able to follow the tutorial instructions to machine a piece	○	○	○	○
I was able to follow all instructions in a timely manner	○	○	○	○
The results of my inputs were as I expected	○	○	○	○
The CNC controls were easy to read	○	○	○	○
The CNC keyboard was responsive	○	○	○	○
The CNC controls were easy to use	○	○	○	○
I could operate a CNC machine after using this software	○	○	○	○
The hologram seemed realistic	○	○	○	○
I needed help in using this software	○	○	○	○
I found the HoloLens easy to use	○	○	○	○
Compared to watching a video or reading a tutorial, I found this experience more engaging	○	○	○	○

Fig. 5.41. 4-point Likert scale questions from the MR-AVML usability study.



Please rate the following:

	Beginner	Some Knowledge	Can fully operate	Professional
What is your previous experience with a CNC Machine?	○	○	○	○
How would you rate your level of computer ability?	○	○	○	○

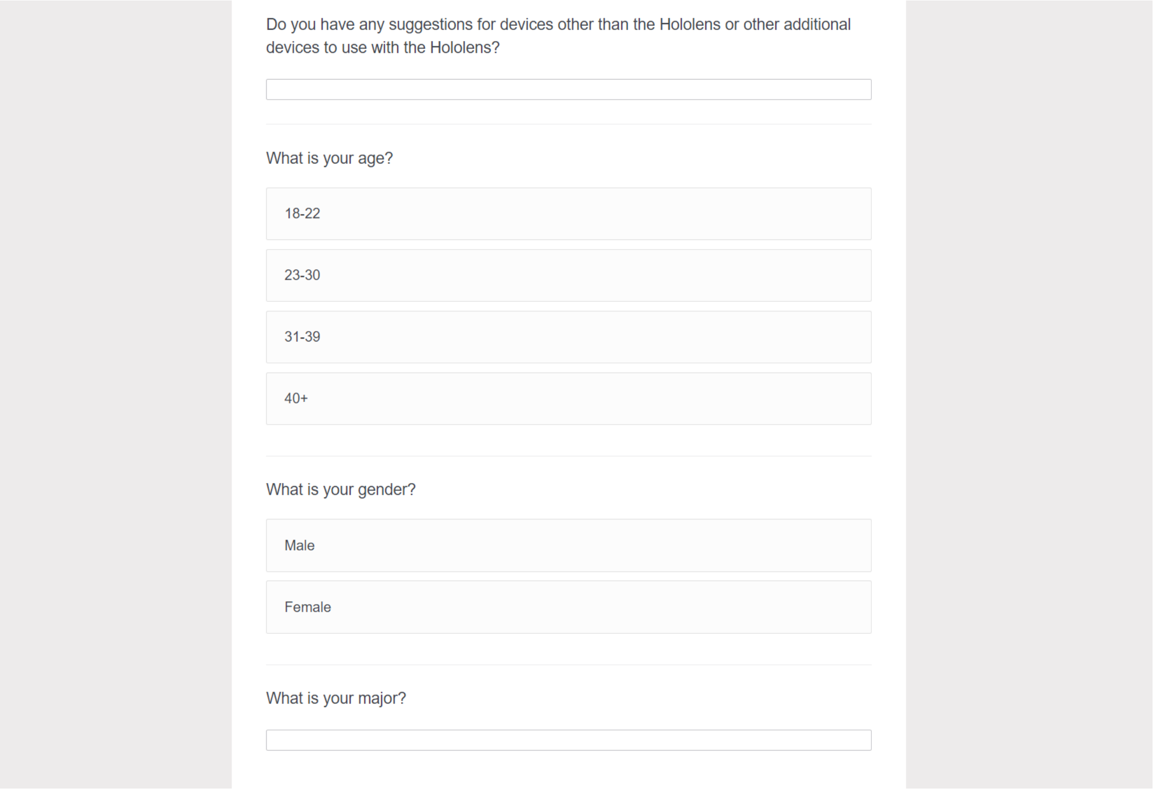
What parts, if any, did you find difficult?

What did you like about this program?

What, if anything, would you change about this program?

Fig. 5.42. Background/open-ended questions from the MR-AVML usability study.

In addition to studying the pedagogical capabilities of the MR-AVML, gathering data on the usability of the system as perceived by the users can be useful for improving the system during future iterations. To that end, a twenty-four-question survey was prepared for each student to take following the training session. Five of these questions focus on demographic information about the users. The other nineteen focused on ease of use, ability to complete the training, and overall opinion of the simulation. Of the nineteen, fifteen of the questions were scored using a four-point Likert scale, giving participants the option to answer ”Strongly Agree”, ”Agree”, ”Disagree”, or ”Strongly Disagree” to each question. A ”Neutral” option was intentionally avoided to ensure more useful feedback from the questions. The remaining four questions were open-ended, prompting a written response from participants.



Do you have any suggestions for devices other than the Hololens or other additional devices to use with the Hololens?

What is your age?

- 18-22
- 23-30
- 31-39
- 40+

What is your gender?

- Male
- Female

What is your major?

Fig. 5.43. Demographic questions from the MR-AVML usability study.

5.3 Pedagogical Study Setup

The pedagogical study was designed to provide concrete data on the teaching capabilities of the MR-AVML compared to that of the VR-AVML and hands-on training. After both the MR-AVML group and the hands-on training group completed their training sessions, the groups were asked to take the same eight question quiz. Having both groups of students complete the same quiz will allow the hands-on group to serve as a control for the study. The questions scored during the quiz are as follows: Questions one and eight require pressing specific buttons on the machine. Questions two, three, and seven require learning the layout of the machine software. Questions four, five, and six require learning manual G-code entry.

1. Power on the machine.
2. Navigate the main menu.
3. Import code from disk.
4. Turn on coolant.
5. Changing tools.
6. Turn ON/OFF spindle.
7. Run code.
8. Emergency stop.

Fig. 5.44. Questions from the MR-AVML pedagogical study.

The quiz was administered directly by a course instructor, who asked students to perform actions related to each of the questions on the physical Fadal VMC-3016L machine located on campus. The quiz was scored out of 20 possible points, with each question having a maximum of 2.5 points. Smaller deductions were enacted if a student made a minor mistake during each step or needed a small hint to complete the action. No points were awarded for any question in which a student needed to be directly told how to complete the given task.

The quiz itself is simple with eight total questions. It is a standard quiz that has been used by the course instructor for ME 54600 for three semesters. The content of the quiz is designed to be taught during a short training session. The shorter duration allows a group of students to individually train with the HoloLens version within a reasonable time frame. The steps covered still require a student to pay attention during the training session in order to perform well on the quiz. This is evidenced by some of the lower scores observed during the VR-AVML study [18].

The VR-AVML study also included a hands-on control group. Because the MR-AVML study mirrors the VR-AVML study, the MR-AVML study could have relied on the previous control group. However, including a second control group allows the control groups used during the two studies to be compared. By comparing the two controls, the grading of the two studies can be normalized.

6. ASSESSMENT RESULTS

Data was received from a total of 41 students during the April 2018 MR-AVML comparative study. Of those students, 21 were trained on the MR-AVML; the other 20 received training instructions directly from the course instructor. Training for the MR-AVML group took place on April 11th and April 16th, while all students in the control group were trained on April 11th. Each of the 21 students trained on the MR-AVML filled out a usability survey immediately following their individual experience with the system. On April 18th, all students were administered the same quiz on the physical machine that the control group was initially trained on. Data from this quiz is used for the pedagogical study. Additional data on the effectiveness of an immersive system without the use of active and passive haptic feedback comes from the mirrored study on the VR-AVML from [18]. This data will provide a basis for assessment of the benefits of mixed reality and haptic feedback.

This chapter introduces both the data gathered from the usability study and the data gathered from the pedagogical study. Once the data has been presented, a discussion of mixed reality in advanced manufacturing training environments is presented.

6.1 Usability Study Results

The following tables show the results for the usability study conducted immediately following the MR-AVML training session. A transcript of the questions in the study is provided in Figures 5.41, 5.42, and 5.43.

Table 6.1.
Usability Study - Multiple Choice Qualitative Questions.

Question No.	No. Strongly Disagreed	No. Disagreed	No. Agreed	No. Strongly Agreed
Q1	1 (4.76%)	0 (0.00%)	8 (38.09%)	12 (57.14%)
Q2	1 (4.76%)	2 (9.52%)	7 (33.33%)	11 (52.38%)
Q3	1 (4.76%)	2 (9.52%)	3 (14.28%)	15 (71.43%)
Q4	1 (4.76%)	0 (0.00%)	8 (38.09%)	12 (57.14%)
Q5	1 (4.76%)	0 (0.00%)	6 (28.57%)	14 (66.67%)
Q6	1 (4.76%)	1 (4.76%)	7 (33.33%)	12 (57.14%)
Q7	1 (4.76%)	0 (0.00%)	8 (38.09%)	12 (57.14%)
Q8	1 (4.76%)	1 (4.76%)	9 (42.86%)	10 (47.62%)
Q9	2 (9.52%)	0 (0.00%)	5 (23.81%)	14 (66.67%)
Q10	1 (4.76%)	0 (0.00%)	7 (33.33%)	13 (61.90%)
Q11	3 (14.28%)	5 (23.81%)	9 (42.86%)	4 (19.05%)
Q12	1 (4.76%)	1 (4.76%)	13 (61.90%)	6 (28.57%)
Q13	3 (14.28%)	6 (28.57%)	7 (33.33%)	5 (23.81%)
Q14	2 (9.52%)	1 (4.76%)	13 (61.90%)	5 (23.81%)
Q15	1 (4.76%)	1 (4.76%)	4 (19.05%)	15 (71.43%)

Table 6.2. Usability Study - Open Ended Qualitative Questions.

Question No.	Responses
Q16	PRO: ”none”
	PRO: ”none”
	PRO: ”Found nothing difficult”
	CON: ”The process was not difficult, just some of the language was harder to understand for my level of understanding.”
	CON: ”The textbox was not fully visible at all times.”
	CON: ”I didn’t speak loud enough for the voice inputs.”
	CON: ”progressing the tutorial using ”next””
	CON: ”walking as the screen sort of shakes”
	CON: ”The VR unit used experienced some visual glitches, but overall very cool!”

Continued on next page

Table 6.2 – continued from previous page

Question No.	Responses
	CON: ”Ensuring the headset is on correctly. A little difficult to comfortably wear the headset with glasses on.”
	CON: ”Couldn’t see very well, field of view was limited”
	CON: ”Hard to see things. Too zoomed in.”
	CON: ”voice command and the head set did not sit right”
Q17	PRO: ”I liked the interaction that I was able to do and it was even fun.”
	PRO: ”it was more interesting than a lecture”
	PRO: ”It was something I’ve never experienced before.”
	PRO: ”It felt like I was actually using a CNC machine”
	PRO: ”It is a new way to learn and I believe it should be used more often!”
	PRO: ”How interactive it was, and how almost life like it was.”
	PRO: ”shows you how milling can be done easily with the right codes”
	PRO: ”I was able to see the effects of my actions instantly”
	PRO: ”I have never received training over VR. Awesome experience”
	PRO: ”Very simple and easy to understand”
	PRO: ”The quickness of it”
	PRO: ”vr”
	PRO: ”It allowed for actually typing on a controller.”
	PRO: ”very engaging”
Q18	PRO: ”I can’t think of any changes”
	PRO: ”Nothing”
	PRO: ”none”
	PRO: ”The program seems to be fully functional to me”
	PRO: ”More implementation of VR training in Student courses. Really allows hands-on experience.”

Continued on next page

Table 6.2 – continued from previous page

Question No.	Responses
	CON: ”That the tutorial window wouldn’t stay in the same place but would remain a constant distance so I wouldn’t need to move”
	CON: ”Upgraded equipment to improve experience.”
	CON: ”If I had more time to explore myself”
	CON: ”try not to walk that far and closer mic”
	CON: ”I would like more information on the button function and how they work”
	CON: ”have larger field of view”
	CON: ”Make the textbox visibility more robust and show more arrows to where you should be going/looking.”
	CON: ”Better sound effects”
Q19	PRO: ”No”
	PRO: ”no”
	PRO: ”no”
	PRO: ”N/A”
	PRO: ”N/A”
	PRO: ”Not at this time.
	PRO: ”None that I can think of”
	PRO: ”none”
	CON: ”Oculus Rift utilizes hand controls that would be interesting to implement”
	CON: ”mic on the glass”
	CON: ”a handheld cursor”
	CON: ”Instead of yelling out NEXT, may be press a button on a hand toggle

Table 6.3.
Usability Study - Previous Knowledge Questions.

Question No.	Beginner	Some Knowledge	Can fully operate	Professional
Q20	12 (57.14%)	5 (23.81%)	3 (14.28%)	1 (4.76%)
Q21	1 (4.76%)	3 (14.28%)	15 (71.43%)	2 (9.52%)

Table 6.4.
Usability Study - Demographic Questions.

Q22	Age 18-22: 11 (52.38%)	Age 23-30: 9 (42.86%)	Age 31-39: 1 (4.76%)
Q23	Male: 21 (100%)	Female: 0 (0%)	
Q24	Mechanical Engineering: 19 (90.48%)	Mechanical Engineering and Physics: 1 (4.76%)	Biomedical Engineering: 1 (4.76%)

Table 6.5.
Pedagogical Study - Question Averages.

Question No.	MR-AVML Group Average	Control Group Average
Q1	2.3095 / 2.5 = 92.38%	2.45 / 2.5 = 98%
Q2	2.3809 / 2.5 = 95.24%	2.4 / 2.5 = 96%
Q3	2.5 / 2.5 = 100%	2.475 / 2.5 = 99%
Q4	2.4524 / 2.5 = 98.09%	2.45 / 2.5 = 98%
Q5	2.1667 / 2.5 = 86.67%	2.45 / 2.5 = 98%
Q6	2.3809 / 2.5 = 95.24%	2.325 / 2.5 = 93%
Q7	2.5 / 2.5 = 100%	2.5 / 2.5 = 100%
Q8	2.5 / 2.5 = 100%	2.35 / 2.5 = 94%
Final Score	19.1904 / 20 = 95.95%	19.4 / 20 = 97%

6.2 Pedagogical Study Results

For Table 6.5, refer to the pedagogical study questions listed in Figure 5.44. Each question in the pedagogical study is scored out of a possible 2.5 points, for a total of 20 possible points. The averaged data of both the MR-AVML group and the control group is shown in Table 6.5. The raw data, including comments from the instructor grading the quiz, can be found in Table 6.6.

Table 6.6. Pedagogical Study - Raw Data.

Name	Assessment Questions	Points Earned	Comments
Student 1 (Control)	Power on the machine	2.5 / 2.5	
	Navigate main menu	2.5 / 2.5	
	Import code from disk	2.5 / 2.5	
	Turn on coolant	2.5 / 2.5	
	Changing tools	2.5 / 2.5	
	Turn ON/OFF Spindle	2 / 2.5	inserted wrong m code
	Run Code	2.5 / 2.5	
	Emergency Stop	0 / 2.5	not sure where it was
		17 / 20	
Student 2 (Control)	Power on the machine	2.5 / 2.5	
	Navigate main menu	2.5 / 2.5	
	Import code from disk	2.5 / 2.5	
	Turn on coolant	2.5 / 2.5	
	Changing tools	2.5 / 2.5	
	Turn ON/OFF Spindle	2 / 2.5	inserted wrong m code
	Run Code	2.5 / 2.5	
	Emergency Stop	2.5 / 2.5	
		19.5 / 20	
Student 3 (MR-AVML)	Power on the machine	2.5 / 2.5	
	Navigate main menu	2.5 / 2.5	
	Import code from disk	2.5 / 2.5	
	Turn on coolant	2.5 / 2.5	
	Changing tools	2.5 / 2.5	
	Turn ON/OFF Spindle	2.5 / 2.5	
	Run Code	2.5 / 2.5	

Continued on next page

Table 6.6 – continued from previous page

Name	Assessment Questions	Points Earned	Comments
	Emergency Stop	2.5 / 2.5	
		20 / 2.5	
Student 4 (Control)	Power on the machine	2.5 / 2.5	
	Navigate main menu	2.5 / 2.5	
	Import code from disk	2.5 / 2.5	
	Turn on coolant	2 / 2.5	inserted G code instead of M code
	Changing tools	2 / 2.5	inserted wrong m code
	Turn ON/OFF Spindle	0 / 2.5	forgot the code
	Run Code	2.5 / 2.5	
	Emergency Stop	2.5 / 2.5	
		16.5 / 20	
Student 5 (Control)	Power on the machine	2.5 / 2.5	
	Navigate main menu	2 / 2.5	missed o and zero
	Import code from disk	2.5 / 2.5	
	Turn on coolant	2.5 / 2.5	
	Changing tools	2.5 / 2.5	
	Turn ON/OFF Spindle	2.5 / 2.5	
	Run Code	2.5 / 2.5	
	Emergency Stop	2.5 / 2.5	
		19.5 / 20	
Student 6 (MR-AVML)	Power on the machine	2.5 / 2.5	
	Navigate main menu	2.5 / 2.5	
	Import code from disk	2.5 / 2.5	
	Turn on coolant	2.5 / 2.5	
	Changing tools	2 / 2.5	missed o and zero

Continued on next page

Table 6.6 – continued from previous page

Name	Assessment Questions	Points Earned	Comments
	Turn ON/OFF Spindle	2 / 2.5	inserted wrong m code
	Run Code	2.5 / 2.5	
	Emergency Stop	2.5 / 2.5	
		19 / 2.5	
Student 7 (Control)	Power on the machine	2.5 / 2.5	
	Navigate main menu	2.5 / 2.5	
	Import code from disk	2.5 / 2.5	
	Turn on coolant	2.5 / 2.5	
	Changing tools	2.5 / 2.5	
	Turn ON/OFF Spindle	2.5 / 2.5	
	Run Code	2.5 / 2.5	
	Emergency Stop	2.5 / 2.5	
		20 / 20	
Student 8 (MR-AVML)	Power on the machine	2.5 / 2.5	
	Navigate main menu	2.5 / 2.5	
	Import code from disk	2.5 / 2.5	
	Turn on coolant	2.5 / 2.5	
	Changing tools	2.5 / 2.5	
	Turn ON/OFF Spindle	2.5 / 2.5	
	Run Code	2.5 / 2.5	
	Emergency Stop	2.5 / 2.5	
		20 / 20	
Student 9 (MR-AVML)	Power on the machine	2.5 / 2.5	
	Navigate main menu	2.5 / 2.5	
	Import code from disk	2.5 / 2.5	
	Turn on coolant	2.5 / 2.5	

Continued on next page

Table 6.6 – continued from previous page

Name	Assessment Questions	Points Earned	Comments
	Changing tools	2.5 / 2.5	
	Turn ON/OFF Spindle	2.5 / 2.5	
	Run Code	2.5 / 2.5	
	Emergency Stop	2.5 / 2.5	
		20 / 20	
Student 10 (Control)	Power on the machine	2.5 / 2.5	
	Navigate main menu	2.5 / 2.5	
	Import code from disk	2.5 / 2.5	
	Turn on coolant	2.5 / 2.5	
	Changing tools	2.5 / 2.5	
	Turn ON/OFF Spindle	2.5 / 2.5	
	Run Code	2.5 / 2.5	
	Emergency Stop	2.5 / 2.5	
		20 / 20	
Student 11 (Control)	Power on the machine	2.5 / 2.5	
	Navigate main menu	2 / 2.5	pushed manual button only once
	Import code from disk	2.5 / 2.5	
	Turn on coolant	2.5 / 2.5	
	Changing tools	2.5 / 2.5	
	Turn ON/OFF Spindle	2.5 / 2.5	
	Run Code	2.5 / 2.5	
	Emergency Stop	2 / 2.5	forgot the button then corrected
		19 / 20	
Student 12 (Control)	Power on the machine	2.5 / 2.5	
	Navigate main menu	2.5 / 2.5	

Continued on next page

Table 6.6 – continued from previous page

Name	Assessment Questions	Points Earned	Comments
	Import code from disk	2.5 / 2.5	
	Turn on coolant	2.5 / 2.5	
	Changing tools	2.5 / 2.5	
	Turn ON/OFF Spindle	2.5 / 2.5	
	Run Code	2.5 / 2.5	
	Emergency Stop	2.5 / 2.5	
		20 / 20	
Student 13 (MR-AVML)	Power on the machine	2.5 / 2.5	
	Navigate main menu	2.5 / 2.5	
	Import code from disk	2.5 / 2.5	
	Turn on coolant	2.5 / 2.5	
	Changing tools	2.5 / 2.5	
	Turn ON/OFF Spindle	2.5 / 2.5	
	Run Code	2.5 / 2.5	
	Emergency Stop	2.5 / 2.5	
		20 / 20	
Student 14 (Control)	Power on the machine	2.5 / 2.5	
	Navigate main menu	2.5 / 2.5	
	Import code from disk	2.5 / 2.5	
	Turn on coolant	2.5 / 2.5	
	Changing tools	2.5 / 2.5	
	Turn ON/OFF Spindle	2 / 2.5	missed o and zero
	Run Code	2.5 / 2.5	
	Emergency Stop	2.5 / 2.5	
		20 / 20	
Student 15 (MR-AVML)	Power on the machine	2.5 / 2.5	

Continued on next page

Table 6.6 – continued from previous page

Name	Assessment Questions	Points Earned	Comments
	Navigate main menu	2.5 / 2.5	
	Import code from disk	2.5 / 2.5	
	Turn on coolant	2.5 / 2.5	
	Changing tools	2 / 2.5	inserted wrong m code
	Turn ON/OFF Spindle	2 / 2.5	missed o and zero
	Run Code	2.5 / 2.5	
	Emergency Stop	2.5 / 2.5	
		19 / 20	
Student 16 (MR-AVML)	Power on the machine	0 / 2.5	forgot this function
	Navigate main menu	2 / 2.5	pushed manual button only once
	Import code from disk	2.5 / 2.5	
	Turn on coolant	2 / 2.5	inserted wrong m code
	Changing tools	0 / 2.5	forgot this code
	Turn ON/OFF Spindle	2 / 2.5	inserted wrong m code
	Run Code	2.5 / 2.5	
	Emergency Stop	2.5 / 2.5	
		13.5 / 20	
Student 17 (MR-AVML)	Power on the machine	2 / 2.5	pressed the wrong start button
	Navigate main menu	2 / 2.5	pushed manual button only once
	Import code from disk	2.5 / 2.5	
	Turn on coolant	2.5 / 2.5	
	Changing tools	2.5 / 2.5	
	Turn ON/OFF Spindle	2.5 / 2.5	
	Run Code	2.5 / 2.5	

Continued on next page

Table 6.6 – continued from previous page

Name	Assessment Questions	Points Earned	Comments
	Emergency Stop	2.5 / 2.5	
		19 / 20	
Student 18 (MR-AVML)	Power on the machine	2 / 2.5	pressed the wrong start button
	Navigate main menu	2 / 2.5	pushed manual button only once
	Import code from disk	2.5 / 2.5	
	Turn on coolant	2.5 / 2.5	
	Changing tools	2.5 / 2.5	
	Turn ON/OFF Spindle	2.5 / 2.5	
	Run Code	2.5 / 2.5	
	Emergency Stop	2.5 / 2.5	
		19 / 20	
Student 19 (MR-AVML)	Power on the machine	2.5 / 2.5	
	Navigate main menu	2.5 / 2.5	
	Import code from disk	2.5 / 2.5	
	Turn on coolant	2.5 / 2.5	
	Changing tools	2.5 / 2.5	
	Turn ON/OFF Spindle	2.5 / 2.5	
	Run Code	2.5 / 2.5	
	Emergency Stop	2.5 / 2.5	
		20 / 20	
Student 20 (Control)	Power on the machine	2 / 2.5	pressed the wrong start button
	Navigate main menu	2.5 / 2.5	
	Import code from disk	2.5 / 2.5	
	Turn on coolant	2.5 / 2.5	

Continued on next page

Table 6.6 – continued from previous page

Name	Assessment Questions	Points Earned	Comments
	Changing tools	2.5 / 2.5	
	Turn ON/OFF Spindle	2.5 / 2.5	
	Run Code	2.5 / 2.5	
	Emergency Stop	2.5 / 2.5	
		19.5 / 20	
Student 21 (MR-AVML)	Power on the machine	2.5 / 2.5	
	Navigate main menu	2.5 / 2.5	
	Import code from disk	2.5 / 2.5	
	Turn on coolant	2.5 / 2.5	
	Changing tools	2.5 / 2.5	
	Turn ON/OFF Spindle	2.5 / 2.5	
	Run Code	2.5 / 2.5	
	Emergency Stop	2.5 / 2.5	
		20 / 20	
Student 22 (Control)	Power on the machine	2.5 / 2.5	
	Navigate main menu	2.5 / 2.5	
	Import code from disk	2.5 / 2.5	
	Turn on coolant	2.5 / 2.5	
	Changing tools	2.5 / 2.5	
	Turn ON/OFF Spindle	2.5 / 2.5	
	Run Code	2.5 / 2.5	
	Emergency Stop	2.5 / 2.5	
		20 / 20	
Student 23 (MR-AVML)	Power on the machine	2.5 / 2.5	
	Navigate main menu	2.5 / 2.5	
	Import code from disk	2.5 / 2.5	

Continued on next page

Table 6.6 – continued from previous page

Name	Assessment Questions	Points Earned	Comments
	Turn on coolant	2.5 / 2.5	
	Changing tools	2.5 / 2.5	
	Turn ON/OFF Spindle	2.5 / 2.5	
	Run Code	2.5 / 2.5	
	Emergency Stop	2.5 / 2.5	
		20 / 20	
Student 24 (MR-AVML)	Power on the machine	2.5 / 2.5	
	Navigate main menu	2.5 / 2.5	
	Import code from disk	2.5 / 2.5	
	Turn on coolant	2.5 / 2.5	
	Changing tools	2.5 / 2.5	
	Turn ON/OFF Spindle	2.5 / 2.5	
	Run Code	2.5 / 2.5	
	Emergency Stop	2.5 / 2.5	
		20 / 20	
Student 25 (Control)	Power on the machine	2.5 / 2.5	
	Navigate main menu	2.5 / 2.5	
	Import code from disk	2.5 / 2.5	
	Turn on coolant	2.5 / 2.5	
	Changing tools	2.5 / 2.5	
	Turn ON/OFF Spindle	2.5 / 2.5	
	Run Code	2.5 / 2.5	
	Emergency Stop	2.5 / 2.5	
		20 / 20	
Student 26 (Control)	Power on the machine	2.5 / 2.5	
	Navigate main menu	2.5 / 2.5	

Continued on next page

Table 6.6 – continued from previous page

Name	Assessment Questions	Points Earned	Comments
	Import code from disk	2.5 / 2.5	
	Turn on coolant	2.5 / 2.5	
	Changing tools	2.5 / 2.5	
	Turn ON/OFF Spindle	2.5 / 2.5	
	Run Code	2.5 / 2.5	
	Emergency Stop	2.5 / 2.5	
		20 / 20	
Student 27 (Control)	Power on the machine	2.5 / 2.5	
	Navigate main menu	2.5 / 2.5	
	Import code from disk	2.5 / 2.5	
	Turn on coolant	2.5 / 2.5	
	Changing tools	2.5 / 2.5	
	Turn ON/OFF Spindle	2.5 / 2.5	
	Run Code	2.5 / 2.5	
	Emergency Stop	2.5 / 2.5	
		20 / 20	
Student 28 (Control)	Power on the machine	2.5 / 2.5	
	Navigate main menu	2.5 / 2.5	
	Import code from disk	2.5 / 2.5	
	Turn on coolant	2.5 / 2.5	
	Changing tools	2.5 / 2.5	
	Turn ON/OFF Spindle	2.5 / 2.5	
	Run Code	2.5 / 2.5	
	Emergency Stop	2.5 / 2.5	
		20 / 20	
Student 29 (MR-AVML)	Power on the machine	2.5 / 2.5	

Continued on next page

Table 6.6 – continued from previous page

Name	Assessment Questions	Points Earned	Comments
	Navigate main menu	2.5 / 2.5	
	Import code from disk	2.5 / 2.5	
	Turn on coolant	2.5 / 2.5	
	Changing tools	2.5 / 2.5	
	Turn ON/OFF Spindle	2.5 / 2.5	
	Run Code	2.5 / 2.5	
	Emergency Stop	2.5 / 2.5	
		20 / 20	
Student 30 (Control)	Power on the machine	2.5 / 2.5	
	Navigate main menu	2 / 2.5	pushed manual button only once
	Import code from disk	2.5 / 2.5	
	Turn on coolant	2.5 / 2.5	
	Changing tools	2.5 / 2.5	
	Turn ON/OFF Spindle	2.5 / 2.5	
	Run Code	2.5 / 2.5	
	Emergency Stop	2.5 / 2.5	
		19.5 / 20	
Student 31 (MR-AVML)	Power on the machine	2 / 2.5	pressed the wrong start button
	Navigate main menu	2 / 2.5	pushed manual button only once
	Import code from disk	2.5 / 2.5	
	Turn on coolant	2.5 / 2.5	
	Changing tools	2.5 / 2.5	
	Turn ON/OFF Spindle	2 / 2.5	missed o and zero
	Run Code	2.5 / 2.5	

Continued on next page

Table 6.6 – continued from previous page

Name	Assessment Questions	Points Earned	Comments
	Emergency Stop	2.5 / 2.5	
		18.5 / 20	
Student 32 (MR-AVML)	Power on the machine	2.5 / 2.5	
	Navigate main menu	2 / 2.5	pushed manual button only once
	Import code from disk	2.5 / 2.5	
	Turn on coolant	2.5 / 2.5	
	Changing tools	2 / 2.5	inserted wrong m code
	Turn ON/OFF Spindle	2.5 / 2.5	
	Run Code	2.5 / 2.5	
	Emergency Stop	2.5 / 2.5	
		19 / 20	
Student 33 (Control)	Power on the machine	2 / 2.5	pressed the wrong start button
	Navigate main menu	2 / 2.5	pushed manual button only once
	Import code from disk	2.5 / 2.5	
	Turn on coolant	2.5 / 2.5	
	Changing tools	2.5 / 2.5	
	Turn ON/OFF Spindle	2.5 / 2.5	
	Run Code	2.5 / 2.5	
	Emergency Stop	2.5 / 2.5	
		19 / 20	
Student 34 (MR-AVML)	Power on the machine	2.5 / 2.5	
	Navigate main menu	2.5 / 2.5	
	Import code from disk	2.5 / 2.5	
	Turn on coolant	2.5 / 2.5	

Continued on next page

Table 6.6 – continued from previous page

Name	Assessment Questions	Points Earned	Comments
	Changing tools	2.5 / 2.5	
	Turn ON/OFF Spindle	2.5 / 2.5	
	Run Code	2.5 / 2.5	
	Emergency Stop	2.5 / 2.5	
		20 / 20	
Student 35 (MR-AVML)	Power on the machine	2.5 / 2.5	
	Navigate main menu	2 / 2.5	pushed manual button only once
	Import code from disk	2.5 / 2.5	
	Turn on coolant	2 / 2.5	inserted wrong m code
	Changing tools	0 / 2.5	forgot this command
	Turn ON/OFF Spindle	2.5 / 2.5	
	Run Code	2.5 / 2.5	
	Emergency Stop	2.5 / 2.5	
		16.5 / 20	
Student 36 (MR-AVML)	Power on the machine	2.5 / 2.5	
	Navigate main menu	2.5 / 2.5	
	Import code from disk	2.5 / 2.5	
	Turn on coolant	2.5 / 2.5	
	Changing tools	2 / 2.5	inserted wrong m code
	Turn ON/OFF Spindle	2 / 2.5	inserted wrong m code
	Run Code	2.5 / 2.5	
	Emergency Stop	2.5 / 2.5	
		19 / 20	
Student 37 (Control)	Power on the machine	2.5 / 2.5	
	Navigate main menu	2.5 / 2.5	

Continued on next page

Table 6.6 – continued from previous page

Name	Assessment Questions	Points Earned	Comments
	Import code from disk	2 / 2.5	forgot the location but finally I hinted the he corrected
	Turn on coolant	2 / 2.5	inserted wrong m code
	Changing tools	2.5 / 2.5	
	Turn ON/OFF Spindle	2.5 / 2.5	
	Run Code	2.5 / 2.5	
	Emergency Stop	2.5 / 2.5	
		19 / 20	
Student 38 (Control)	Power on the machine	2.5 / 2.5	
	Navigate main menu	2.5 / 2.5	
	Import code from disk	2.5 / 2.5	
	Turn on coolant	2.5 / 2.5	
	Changing tools	2.5 / 2.5	
	Turn ON/OFF Spindle	2.5 / 2.5	
	Run Code	2.5 / 2.5	
	Emergency Stop	2.5 / 2.5	
		20 / 20	
Student 39 (Control)	Power on the machine	2.5 / 2.5	
	Navigate main menu	2.5 / 2.5	
	Import code from disk	2.5 / 2.5	
	Turn on coolant	2.5 / 2.5	
	Changing tools	2.5 / 2.5	
	Turn ON/OFF Spindle	2.5 / 2.5	
	Run Code	2.5 / 2.5	
	Emergency Stop	2.5 / 2.5	

Continued on next page

Table 6.6 – continued from previous page

Name	Assessment Questions	Points Earned	Comments
		20 / 20	
Student 40 (MR-AVML)	Power on the machine	2.5 / 2.5	
	Navigate main menu	2.5 / 2.5	
	Import code from disk	2.5 / 2.5	
	Turn on coolant	2.5 / 2.5	
	Changing tools	2.5 / 2.5	
	Turn ON/OFF Spindle	2.5 / 2.5	
	Run Code	2.5 / 2.5	
	Emergency Stop	2.5 / 2.5	
		20 / 20	
Student 41 (MR-AVML)	Power on the machine	2.5 / 2.5	
	Navigate main menu	2.5 / 2.5	
	Import code from disk	2.5 / 2.5	
	Turn on coolant	2.5 / 2.5	
	Changing tools	2.5 / 2.5	
	Turn ON/OFF Spindle	2.5 / 2.5	
	Run Code	2.5 / 2.5	
	Emergency Stop	2.5 / 2.5	
		20 / 20	

7. DISCUSSION

7.1 Usability Study Discussion

During the training session, one student encountered a bug that caused the holograms displayed on the HoloLens to disappear and never reappear. This bug was caught via monitoring the student's progress as described previously. The bug appeared roughly one third of the way through the training session. After the error was caught, the student was asked to restart their session. After restarting, this issue did not occur again. Moreover, several students had difficulty understanding the concept of the holograms maintaining their position in the physical space when moving. These students would stand perfectly still in place, even when an instruction asked them to walk around in the environment. As moving around is a requirement in order to perform the "power on machine" task (the power switch is located at the rear of the machine), an exception was made to the "no communication between the user and the researcher" rule outlined in the "Methodology" chapter. In future iterations, it may be necessary to include further instruction to users prior to starting the program about the need to physically walk around to fully experience the training. This would eliminate the need for any other communication outside of what is provided to the user in the program.

Regarding the data from the questions scored via Likert scale in Table 6.1, the majority of the responses were positive. Aside from question 13, which asked whether the user needed help with the software, each question had at least 61.91% positive responses. The lowest score of positive responses was question 13, which had 42.85% positive (in the context of the needed help question, Disagree and Strongly Disagree are considered to be the positive responses), followed by question 11 (asking about whether the user feels they could operate a real CNC machine after the training), which had 61.91% positive responses. Every other question had at least 85.71% positive responses, with 11 out of the 15 questions asked having the majority of their responses ranked as Strongly Agree. From this information, a few statements can be made.

- Students were able to complete the tasks asked of them with little difficulty.
- Despite needing help with the knowledge that they could navigate around the environment, students thought navigation was easy once they learned it was possible.
- The holograms were perceived to be realistic, but not as many students felt strongly that they were realistic. This is expected from a mixed reality system compared to similar usability responses in virtual reality [18], but does point to potential software improvements that can be made.
- The physical CNC controller was perceived to be responsive and easy to use.
- Students felt that the HoloLens itself was not found to be difficult to use, but was not completely easy to use, either.
- Students felt that, overall, the experience was positive and engaging.

The written statements in the open-ended questions in Table 7.2 support the above observations. The most common complaint was with the HoloLens itself. Students complained about the microphone, the field of view, and even the way the HoloLens fits. It is likely for these reasons that the HoloLens was not perceived to be completely easy to use. Aside from the microphone, which could potentially be improved with software settings, the other complaints about the HoloLens cannot be improved without a significant hardware revision. Some students noted small visual glitches or inconsistencies while moving around, explaining why some users felt that the holograms were less realistic. Reduced realism could also be attributed to the sound effects implemented in the environment, which received a complaint from one student. On a more positive side, students felt that the physical controls of the MR-AVML were beneficial to the environment. Many noted how they enjoyed the interactive nature of the system, with some directly mentioning the physical keyboard as a positive feature. They stated how the interactions were simple, responsive, and felt like actually working with a CNC machine. This relates directly to the original motivation of the MR-AVML, providing an experience that mimics the feeling of working with a CNC machine as closely as possible. Students also stated that the experience was engaging. Several had never experienced VR or MR at all and were engaged by the technology itself. Others noted that the immediacy of their real-world actions causing virtual reactions, as well as the hands-on nature of the program, made the system engaging to use.

Two other items in the system that received complaints were related to the non-physical input methods and the positioning of the tutorial. Complaints about the voice input have been mentioned somewhat previously, but it was perceived to be one of the larger issues with an otherwise successful environment. Students stated that they had difficulty speaking loudly enough for the microphone to consistently register their inputs. This is likely a software issue, rather than an issue with the HoloLens microphone itself, but it is something to be improved in future iterations. Students also requested the option for a handheld cursor. These complaints could be referring to a desire to interact with holograms directly with hand movements. However, this would not be possible without significant hardware additions, potentially moving the system to a Tier 3 definition. Adding a small handheld controller to interact with an item selected by the cursor without relying on the current gaze input system could address this complaint without such a significant revision. Another usability study in the future may be useful to test the benefits of a handheld cursor. As for the tutorial, as anticipated, some students took issue with its stationary nature, saying it would be easier to complete actions if the tasks were constantly in a visible position. While a tutorial constantly in the lower third of the user's vision would not work with the current HoloLens hardware for the reasons stated in the Chapter 3, it may be useful to either place multiple instances of the tutorial box in the environment, or create a tutorial that follows the user as they move around. This tutorial would need to stay outside of the user's vision for most of the experience, but shift to a visible position when it is detected that the user is actively attempting to look at it. This is another aspect that could be explored in future improved iterations of the system.

Overall, the main aspects of the MR-AVML have been received positively during the usability study. In particular, the effectiveness of the haptic controls and the way it mimics the real CNC machine it is modeled after were reviewed positively. The system was able to effectively provide an engaging and realistic training simulation for users.

7.2 Pedagogical Study Discussion

Overall, the results of the MR-AVML group compare favorably to the scores from the control group, with the average score between the two groups differing by only 1.05%. Observing the average scores of the individual questions will help to contextualize these results.

Out of the 8 individual questions, the MR-AVML group scored higher on average than the control group on 4 questions, less than the control group on 3 questions, and tied on 1 question. The 3 questions that had a significant difference between the two groups are Question 1 with a 5.62% difference between the two groups, Question 5 with a 11.33% difference, and Question 8 had a 6% difference. The remaining questions had either a minimal difference caused by an outlier or a negligible difference caused by the MR-AVML group having one extra participant.

7.2.1 Questions with Higher Scores from the Control Group

Question 2 asked students to navigate through the main menu. It has a difference of 0.76% between the two groups. Since this gap is minimal, it indicates that the training between the two groups in this area was roughly equivalent. Improving the tutorial instructions for menu navigation may close this gap completely. The other two questions had a significant gap.

Question 1, with an average score difference of 5.62%, asked students to power on the machine. Given that the MR-AVML training for this question is the only machine interaction that is entirely virtually emulated, it makes sense that this is one area in which the MR-AVML would be weaker than the hands-on, interactive training of the control group. This is an area of the MR-AVML that cannot be meaningfully improved without adding additional hardware to simulate physically flipping the power switch and pressing the power button. The tutorial instructions for this step can be refined, potentially reducing this gap. However, a significant improvement should not be expected without hardware additions.

The third question where the MR-AVML scored less than the physical system is Question 5. This question asked students to enter commands that change tools in the system. This question had a significant difference in average score between the two groups, with the MR-AVML group scoring 11.33% less than the control group. As described in Chapter 5, the tool changing protocol only allowed for one tool change command to be recognized, whereas the Fadal VMC-3016L allows for many tools. It is possible that not allowing for other tool changes caused a limited understanding of the command. This would explain the large score difference. Another potential explanation is that students understood that the physical machine could have more tools, but the instructions failed to explain the full extent of how tool changing works. In either case, this issue is one that can be fixed entirely by adjusting the MR-AVML software.

7.2.2 Questions with Higher Scores from the MR-AVML Group

Out of the 4 questions in which the MR-AVML group scored higher on average, only Question 8 had a significant difference between the two groups. Question 4 had a completely negligible difference between the two groups. The remaining two had a small, but non-negligible difference between the two groups.

Looking at Question 4, which asked students to enter the manual command that turns on the machine coolant, the two groups differ by only 0.09%. This difference is caused by the fact that the MR-AVML has one more student than the control group.

Question 3 asked students to import code from a disk loaded in the machine controller. The MR-AVML students in this question not only scored 1% higher than the control group, but also received a perfect score of 2.5/2.5. Given that this question asks students to perform several hands-on interactions with the machine controller, it makes sense that the MR-AVML group would perform well. The experience gained from interacting with the CNC machine controller replica allowed students to mimic the needed actions to complete the question. Question 6 also had a small gap of 2.24% between the two groups. This question asked students to turn the spindle (drill bit spin) on and off.

The one question that showed a large difference between the two groups with the MR-AVML scoring higher on average was Question 8. Question 8 asks students to locate and press the emergency stop button. The result of a 6% difference between the two groups with a perfect 2.5/2.5 score for all students in the MR-AVML group is somewhat unexpected. The tutorial instructions for the MR-AVML do not make any special mention of the emergency stop button. The most likely possibility to explain this anomaly is that not every student in the control group saw where the emergency stop button was located. Moreover, students in the control group were trained in a larger groups. The MR-AVML group, on the other hand, got to stand directly in front of the CNC machine controller for the duration of the training. This suggests students in the MR-AVML group were more likely to notice and learn about the emergency stop button as a result of being closer to the controller during training.

7.2.3 Other Data Points

The final question to be discussed is Question 7. This question tasked students with executing code that had been loaded into the machine. Results for both the control group and the MR-AVML group were the same, with all students from both groups scoring a perfect 2.5/2.5. In this case, the results show that the required actions were simple enough that all students were able to effectively be trained and remember that training in both groups.

From this analysis of the study data, a few assertions can be made about the MR-AVMLs pedagogical abilities.

- Some instructions, particularly the instructions about the tool change commands, needs improvement.
- The instructional capabilities of the MR-AVML are weaker when users have to interact with the holograms rather than the physical machine controller.
- With the exception of the tool change command, any task related to interacting with the physical controller was taught as effectively by the MR-AVML as the instructor-led training.

This is significant because it indicates that the technology utilized in the MR-AVML is approaching the levels needed to entirely replace instructor-led training.

7.2.4 VR-AVML vs. MR-AVML vs. Instructor-Led Training

The scores obtained previously for the VR-AVML study [18] are compared to the MR-AVML results in Table 7.1.

Table 7.1.
Pedagogical Study Questions.

Question No.	VR-AVML Group Average	Control Group Average
Final Score	17.978 / 20 = 89.89%	19.25 / 20 = 96.25%

Looking first at the difference between the control groups, hereafter referred to as the VR-control group and the MR-control group for the VR-AVML study and the MR-AVML study, respectively. The average score for the VR-control group was 0.75% lower than the MR-control group, and 0.30% higher than the MR-AVML group. The difference between the VR-control group and the MR-control group is small enough to be within the margin of error. This low percent difference allows the control groups to be equally used as baselines.

In [18], the 6.36% difference between the VR-AVML group and the VR-control group indicated that the technology is a viable CNC machine training tool. The MR-AVML improved on the VR-AVML and reduced the difference to 0.30%. As stated in the Section 3.4, the MR-AVML was designed specifically to include the same instructional material as the VR-AVML, edited to fit the control requirements of the HoloLens and the physical controller. The tutorial can be eliminated as a contributing factor in the improvement. Therefore, the passive and active haptic feedback introduced into the MR-AVML are the only factors that can explain the improvement. This shows that implementing haptic feedback techniques in advanced manufacturing virtual environments can be effective. Moreover, a Tier 2 system such as the one proposed may reduce the need for more expensive Tier 1 systems.

8. CONCLUSIONS AND FUTURE WORK

The addition of haptic feedback techniques allowed the MR-AVML system to outperform the VR-AVML. The gap between the VR-AVML and instructor led training was 6.36%. This difference was reduced to 1.05% by the MR-AVML. The MR-AVML was designed to show that mixed reality can be used effectively for CNC machine training. This reduction successfully showed mixed reality's effectiveness. Further, this project showed that Tier 2 virtual environments can provide a cost-effective training environment. Moreover, mixed reality allowed for an accurate input method. Overall, haptic feedback has been shown to make a significant difference for advanced manufacturing training tools.

8.1 Future Work

While the MR-AVML was successful based on the results of the pedagogical study, there are several potential improvements. These should be addressed in future iterations of the MR-AVML.

Some of the instructions in the tutorial need to be improved, especially the tool changing instructions. Menu navigation instructions should also be changed to ensure that they make sense to the user. Instructions about powering on the machine can also be improved.

Moreover, the voice recognition feature either needs to be modified or removed entirely. Programming already exists in the MR-AVML to advance the tutorial without the voice input, so only the tutorials would need to be changed if the latter option is selected. However, given the potential convenience of a hands-free, natural method to progress in the instructions, there should be an attempt to improve the voice input

Another addition to the system that should be considered is more measures for tracking. Some users complained about visual glitches while walking around the holograms of the MR-AVML. More complex tracking could prevent these glitches. Additionally, different input methods for interacting with the holograms should be tested. The HoloLens standard ”AirTaps” should still be avoided to avoid false inputs from interacting with the physical controller. Either a handheld device to press to immediately activate the hologram currently in the users gaze as a sort of click action, or new hardware implementation would be worth investigating.

Additional instruction sets should be programmed into the system. The Fadal VMC-3016L has several menus, functions, and milling capabilities that can be the subject of additional tutorials. Further, a ”sandbox” mode should be implemented that allows users to use the CNC machine freely. To complement this feature, there should also be a tooling check system implemented to alert users whenever a command they have entered would cause damage to the machine. These features should help improve the effectiveness of the environment in training.

REFERENCES

REFERENCES

[1] C.-W. Chang, J.-H. Lee, C.-Y. Wang, and G.-D. Chen, “Improving the authentic learning experience by integrating robots into the mixed-reality environment,” *Computers & Education*, vol. 55, no. 4, pp. 1572–1578, 2010.

[2] H. G. Hoffman, “Physically touching virtual objects using tactile augmentation enhances the realism of virtual environments,” in *Virtual Reality Annual International Symposium, 1998. Proceedings., IEEE 1998.* IEEE, 1998, pp. 59–63.

[3] C. W. Borst and R. A. Volz, “Evaluation of a haptic mixed reality system for interactions with a virtual control panel,” *Presence: Teleoperators & Virtual Environments*, vol. 14, no. 6, pp. 677–696, 2005.

[4] M. Achibet, A. Girard, M. Marchal, and A. Lécuyer, “Leveraging passive haptic feedback in virtual environments with the elastic-arm approach,” *Presence*, vol. 25, no. 1, pp. 17–32, 2016.

[5] D. E. Novy, “Computational immersive displays,” Master’s thesis, Massachusetts Institute of Technology, 2013.

[6] I. Wright, “What can augmented reality do for manufacturing?” May 2017, (Last date accessed: June 20, 2018). [Online]. Available: https://www.engineering.com/AdvancedManufacturing/ ArticleID/ 14904/ What-Can-Augmented-Reality-Do-for-Manufacturing.aspx

[7] “Cncsimulator pro - your virtual workshop!” (Last date accessed: June 20, 2018). [Online]. Available: http://cncsimulator.info

[8] “Cnc machine simulation - cgtech,” (Last date accessed: June 20, 2018). [Online]. Available: http://www.cgtech.com/products/about-vericut/machine-simulation/

[9] “Predator virtual cnc software for 3d cnc machine simulation, cnc verification, setup sheets and reverse engineering,” (Last date accessed: June 20, 2018). [Online]. Available: http://www.predator-software.com/Predator_CNC_Editor_Software.htm

[10] Lamson Global, “Cutviewer - cnc simulation and verification,” 2018, (Last date accessed: June 20, 2018). [Online]. Available: http://www.cutviewer.com/

[11] I. Wright, “5 virtual reality applications in manufacturing,” 2017, (Last date accessed: June 20, 2018). [Online]. Available: https://www.engineering.com/ResourceMain?resid=597

[12] D. K. Aw, “Development of the advanced virtual manufacturing lab (avml),” Master’s thesis, Purdue University, 2006.

[13] A. Wheeler, “Boeing’s ar tablet tool for assembly lines,” May 2015, (Last date accessed: June 20, 2018). [Online]. Available: https://www.engineering .com/ AdvancedManufacturing/ ArticleID/ 10069/Boeings-AR-Tablet-Tool-for-Assembly-Lines.aspx

[14] Eon Reality, “Augmented reality engine training,” 2018, (Last date accessed: June 20, 2018). [Online]. Available: https://www.eonreality.com/portfolio-items/augmented-reality-diesel-engine-training/ ?portfolioCats=611

[15] Fronius.com, “Virtual welding - welding technology - fronius,” 2018, (Last date accessed: June 20, 2018). [Online]. Available: https:// www.fronius.com/ en-au/ australia/welding-technology/our-expertise/welding-education/virtual-welding

[16] Pratt & Whitney, “Pratt & whitney investing in virtual reality training tools to enhance customer training technologies,” April 2017, (Last date accessed: June 20, 2018). [Online]. Available: http://www.pw.utc.com/News/Story/20170425-0200

[17] H. El-Mounayri, T. Wasfy, E. Fernandez, and J. Peters, “Innovative online course using an integrated virtual reality based e-learning tool with course management tool,” in *Innovative techniques in instruction technology, e-learning, e-assessment, and education.* Springer, 2008, pp. 183–189.

[18] C. Rogers, H. El-Mounaryi, T. Wasfy, and J. Satterwhite, “Assessment of stem e-learning in an immersive virtual reality (vr) environment,” *The ASEE Computers in Education (CoED) Journal*, vol. 8, no. 4, 2018.

[19] N. Statt, “This startup’s prototype ar goggles fix one of the hololens’ biggest issues,” January 2018, (Last date accessed: June 20, 2018). [Online]. Available: https://www.theverge.com/2018/1/9/16872110/realmax-ar-goggles-microsoft-hololens-prototype-field-of-view-ces-2018

APPENDICES

A. MR-AVML C# CODE - BLUETOOTHINPUT.CS

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using System.IO.Ports;

public class BluetoothTestinput : MonoBehaviour {

    public static SerialPort sp = new SerialPort("COM3", 38400, Parity.None, 8,
    StopBits.One);
    public string message, message1;
    public string message2, messageFinal;
    public TextMesh text;
    public TextMesh xval, yval, zval, aval, bval;
    public TextMesh filename;
    public Material[] menus;
    public GameObject screen;
    private int numCheck;
    public ParticleSystem coolant1, coolant2, coolant3;
    public GameObject spindle;
    public GameObject toolChanger;
    public GameObject plane;
    public GameObject drill;

    public GameObject tutorialBox;
    private ChangeTutorial scriptCT;


    void Start()
        {

        print("We are about to start\n");
            OpenConnection();
        print("back from opening\n\n");
        //Load in ChangeTutorial script to access tutorialNo var
    text.text = "";
    scriptCT = tutorialBox.GetComponent<ChangeTutorial>() as ChangeTutorial;
    screen.GetComponent<Renderer>().enabled = false;
    numCheck = 0;

}

void Update()
{
    message2 = sp.ReadExisting();
    if (message2.Length > 0)
    {
        HandleInput(message2);
    }
}



void HandleInput(string message)
{
    if (scriptCT.tutorialNo <= 3)
        return;
    else if(scriptCT.tutorialNo == 4)
    {
        if(message == "!")//Start was pressed
        {
            InitializeScreenVals();
        }
        else if(message == "#")//Manual was pressed
        {
            ClearScreenVals();
            //Change from initialization screen to Quick Keys menu
            scriptCT.NextTutorial();
        }
    }
```

```
else if(scriptCT.tutorialNo == 5)
{
    if(message == "#")//Manual was pressed
    {
        if (numCheck < 1)
            numCheck++;//Must be pressed twice to advance
        else
        {
            numCheck = 0;
            //Change from Quick Keys menu to manual entry screen
            scriptCT.NextTutorial();
        }
    }
}
else if(scriptCT.tutorialNo == 6)
{
    if (text.text == "Specified tool\nnot initialized.")
        text.text = "";
    if(message == "e")
    {
        string GCode = text.text;
        text.text = "";
        if(GCode[0] == ’T’)
        {
            if(GCode != "T1")
            {
                text.text = "Specified tool\nnot initialized.";
            }
            else
            {
                toolChanger.GetComponent<Animation>().Play();
            }
        }
        else if(GCode == "M08")
        {
            coolant1.Play();
            coolant2.Play();
            coolant3.Play();
        }
        else if(GCode == "M09")
        {
            coolant1.Stop();
            coolant2.Stop();
            coolant3.Stop();
        }
        else if(GCode == "M03")
        {
            spindle.GetComponent<Animation>().Play();
        }
        else if(GCode == "M05")
        {
            spindle.GetComponent<Animation>().Stop();
        }
    }
    else if(message == "b" || message == "d")
    {
        if (text.text.Length == 0)
            return;
        else
            text.text = text.text.Substring(0, text.text.Length - 1);
    }
    else
    {
        text.text += message;
    }
}
else if(scriptCT.tutorialNo == 7)
{
    if(message == "#")
    {
        screen.GetComponent<Renderer>().material = menus[6];
    }
    else if(message == "@")
    {
        //Change from manual entry screen to Quick Keys menu
        screen.GetComponent<Renderer>().material = menus[1];
        //numCheck to make sure users get back to Quick Keys first
        numCheck++;
    }
    else if(message == " " && numCheck > 0)
    {
        //Change from Quick Keys menu to the Functions menu
        scriptCT.NextTutorial();
        //reset numCheck for later use
        numCheck = 0;
    }
}
else if(scriptCT.tutorialNo == 8)
{
    if(message == "0")
    {
        //Change from Functions menu to the Disk Functions menu
        screen.GetComponent<Renderer>().material = menus[4];
        //numCheck to make sure users get to Disk Functions first
        numCheck++;
    }
    else if(message == "1" && numCheck > 0)
    {
        //Change from Disk Functions menu to Disk to Memory
        scriptCT.NextTutorial();
        //reset numCheck for later use
        numCheck = 0;
    }
}
else if (scriptCT.tutorialNo == 9)
```

```
{
    if (message == "e")//Enter was pressed
    {
        filename.text = "";
        //Change from Disk to Memory to Disk Functions
        screen.GetComponent<Renderer>().material = menus[4];
        //numCheck to make sure users get back to Disk Functions first
        numCheck++;
    }
    else if(message == "#" && numCheck > 0)//manual was pressed
    {
        if(numCheck == 1)//Manual pressed first time
        {
            numCheck++;
            //Change from Disk Functions menu to the Functions menu
            screen.GetComponent<Renderer>().material = menus[3];
        }
        else if(numCheck == 2)//Manual pressed second time
        {
            numCheck++;
            //Change from Functions screen to Quick Keys menu
            screen.GetComponent<Renderer>().material = menus[1];
        }
        else //Manual pressed last time
        {
            numCheck = 0;
            //Change from Quick Keys menu to manual entry screen
            scriptCT.NextTutorial();
        }
    }
    else if (message == "b" || message == "d")//backspace or delete
    was pressed
    {
        if (filename.text.Length == 0)
            return;
            else
                filename.text = filename.text.Substring(0, filename.text.
                Length - 1);
        }
        else if(message == "?")
        {
            filename.text += "/";
        }
        else
        {
            filename.text += message;
        }
    }
    else if(scriptCT.tutorialNo == 10)
    {
        if(message == "@")
        {
            //Change to final screen
            scriptCT.NextTutorial();
            //Play everything needed to simulate machining process
            coolant1.Play();
            coolant2.Play();
            coolant3.Play();
            spindle.GetComponent<Animation>().Play();
            spindle.GetComponent<AudioSource>().Play();
            plane.GetComponent<Animation>().Play();
            drill.GetComponent<Animation>().Play();
        }
    }
}

void InitializeScreenVals()
{
    xval.text = "-5.7";
    yval.text = "-3.2";
    zval.text = "17.1";
    aval.text = "0";
    bval.text = "0";
}

void ClearScreenVals()
{
    xval.text = "";
    yval.text = "";
    zval.text = "";
    aval.text = "";
    bval.text = "";
}


 public void OpenConnection()
 {
     if (sp != null)
     {
         print("port is not NULL\n\n");
         if (sp.IsOpen)
         {
             sp.Close();
             message = "Closing port, because it was already open!";
         }
         else
         {
         print("trying to open\n\n");
             sp.Open();
             sp.ReadTimeout = 500;
         print("opened");
             message = "Port Opened!";
         }
     }
     else
```

```
            {
                if (sp.IsOpen)
                {
                    print("Port is already open");
                }
                else
                {
                    print("Port == null");
                }
            }
        }

        void OnApplicationQuit()
        {
            sp.Close();
        }
    }
```

B. MR-AVML C# CODE - GAZEHANDLER.CS

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class GazeHandler : MonoBehaviour {

    public Camera camera;
    public Material[] tutorials;
    public GameObject tutorialBox;
    public GameObject buttonAlert;
    public TextMesh Loading;
    private ChangeTutorial scriptCT;
    private int sameObject;
    private int lastActivated;
    private float time;
    public GameObject screen;
    private bool powerOn;
// Use this for initialization
void Start () {
        buttonAlert.GetComponent<Renderer>().enabled = false;
        //sameObject: 0 is not touching anything, 1 is power switch, 2 is power
        button, 3 is NextArrow in tutorail, 4 is BackArrow in tutorial.
        sameObject = 0;
        //lastActivated is used as a sort of debounce to make sure the same object
        isn’t activated multiple times in a row.
        lastActivated = 0;
        time = 0;
        //Bool used to check whether the "flip power switch on" or "flip power
        switch off" animations should be played.
        powerOn = false;
        //Initialize loading text to blank.
        Loading.text = "";
        //Load in ChangeTutorial script to access tutorialNo var
        scriptCT = tutorialBox.GetComponent<ChangeTutorial>() as ChangeTutorial;
```

```
}

// Update is called once per frame
void Update () {
        RaycastHit hitInfo;
        if(Physics.Raycast(camera.transform.position, camera.transform.forward,
        out hitInfo, 20.0f, Physics.DefaultRaycastLayers))
        {
            if (hitInfo.collider.gameObject.name == "PowerSwitch")
            {
                if (lastActivated != 1) //Debouncing
                {
                    if (sameObject != 1) //If the user was not previously
                    looking at this object, restart the timer
                    {
                        time = Time.time;
                        sameObject = 1;
                    }
                    else
                    {
                        if (Time.time - time >= 1.5f) //User must continuously
                        look at the object for 1.5 seconds to activate it
                        {
                            if (powerOn == false)
                            {
                                hitInfo.collider.gameObject.GetComponent
                                <Animation>().Play("SwitchON");
                                hitInfo.collider.gameObject.GetComponent
                                <AudioSource>().timeSamples = 2;
                                hitInfo.collider.gameObject.GetComponent
                                <AudioSource>().Play();
                                screen.GetComponent
                                <Renderer>().enabled = true;
                                powerOn = true;
                                buttonAlert.GetComponent
                                <Renderer>().enabled = true;
                            }
                            else
                            {
                                hitInfo.collider.gameObject.GetComponent
                                <Animation>().Play("SwitchOFF");
                                hitInfo.collider.gameObject.GetComponent
                                <AudioSource>().timeSamples = 2;
                                hitInfo.collider.gameObject.GetComponent
                                <AudioSource>().Play();
                                screen.GetComponent<Renderer>().enabled = false;
                                powerOn = false;
                            }
                            lastActivated = 1; //Debounce setup
                            Loading.text = "100%";
                        }
                    }
                }
            }
            else if(hitInfo.collider.gameObject.name == "PowerButton")
            {
                if (lastActivated != 2)
                {
                    if (sameObject != 2)
                    {
                        time = Time.time;
                        sameObject = 2;
                    }
                    else
                    {
                        if (Time.time - time >= 2.5f)
                        {
                            hitInfo.collider.gameObject.GetComponent<Animation>
                            ().Play();
                            hitInfo.collider.gameObject.GetComponent<AudioSource>
                            ().Play();
                            lastActivated = 2;
                            buttonAlert.GetComponent<Renderer>().enabled = false;
                            Loading.text = "100%";
                        }
                    }
                }
            }
            else if(hitInfo.collider.gameObject.name == "NextArrow")
            {
                if (lastActivated != 3)
                {
                    if (sameObject != 3)
                    {
                        time = Time.time;
                        sameObject = 3;
                    }
                    else
                    {
                        if (Time.time - time >= 1.5f)
                        {
                            if (scriptCT.tutorialNo < scriptCT.maxTutorials)
                            {
                                tutorialBox.GetComponent<Renderer>().material =
                                tutorials[++scriptCT.tutorialNo];
                                tutorialBox.GetComponent<AudioSource>().Play();
                            }
                            lastActivated = 3;
                            Loading.text = "100%";
                        }
                    }
                }
            }
            else if (hitInfo.collider.gameObject.name == "BackArrow")
            {
                if(lastActivated != 4)
                {
                    if (sameObject != 4)
                    {
                        time = Time.time;
                        sameObject = 4;
                    }
                    else
                    {
                        if (Time.time - time >= 1.5f)
                        {
                            if (scriptCT.tutorialNo >= 1)
                            {
                                tutorialBox.GetComponent<Renderer>().material
                                = tutorials[--scriptCT.tutorialNo];
                                tutorialBox.GetComponent<AudioSource>().Play();
                            }
                            lastActivated = 4;
                            Loading.text = "100%";
                        }
                    }
                }
            }
            else
            {
                sameObject = 0;
                lastActivated = 0;
                Loading.text = "";
            }
            if(sameObject == 2 && sameObject != lastActivated)
            {
                Loading.text = ((int)(100 * ((Time.time - time) / 2.5)) + "%");
            }
            else if (sameObject != 0 && sameObject != lastActivated)
            {
```

```
                Loading.text = ((int)(100*((Time.time - time) / 1.5)) + "%");
            }
        }
}
}
```

C. MR-AVML C# CODE - CHANGETUTORIAL.CS

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class ChangeTutorial : MonoBehaviour {

    public Material[] tutorials;
    public int maxTutorials;
    public int tutorialNo;
    public GameObject progression;
    private BluetoothTestinput scriptBT;
    public GameObject screen;
    public GameObject arrow1, arrow2, arrow3, arrow4, arrow5, arrow6;

    // Use this for initialization
    void Start () {
        tutorialNo = 0;
        gameObject.GetComponent<Renderer>().material = tutorials[tutorialNo];
        scriptBT = progression.GetComponent<BluetoothTestinput>() as
        BluetoothTestinput;
    }

// Update is called once per frame
void Update () {

}

    public void NextTutorial()
    {
        if(tutorialNo < maxTutorials)
        {
            gameObject.GetComponent<Renderer>().material = tutorials[++tutorialNo];
            gameObject.GetComponent<AudioSource>().Play();
            UpdateScreen();
    }
}

public void BackTutorial()
{
    if (tutorialNo > 0)
    {
        gameObject.GetComponent<Renderer>().material = tutorials[--tutorialNo];
        gameObject.GetComponent<AudioSource>().Play();
        UpdateScreen();
    }
}

private void UpdateScreen()
{
    if(tutorialNo <= 4)
    {
        screen.GetComponent<Renderer>().material = scriptBT.menus[0];
        if(tutorialNo == 2)
        {
            arrow1.GetComponent<Renderer>().enabled = true;
            arrow2.GetComponent<Renderer>().enabled = true;
            arrow3.GetComponent<Renderer>().enabled = true;
            arrow4.GetComponent<Renderer>().enabled = true;
            arrow5.GetComponent<Renderer>().enabled = true;
            arrow6.GetComponent<Renderer>().enabled = true;
        }
        else if(tutorialNo == 3)
        {
            arrow1.GetComponent<Renderer>().enabled = false;
            arrow2.GetComponent<Renderer>().enabled = false;
            arrow3.GetComponent<Renderer>().enabled = false;
            arrow4.GetComponent<Renderer>().enabled = false;
            arrow5.GetComponent<Renderer>().enabled = false;
            arrow6.GetComponent<Renderer>().enabled = false;
            }
        }
        else if(tutorialNo == 5)
        {
            screen.GetComponent<Renderer>().material = scriptBT.menus[1];
        }
        else if (tutorialNo <= 7)
        {
            screen.GetComponent<Renderer>().material = scriptBT.menus[2];
        }
        else if (tutorialNo == 8)
        {
            screen.GetComponent<Renderer>().material = scriptBT.menus[3];
        }
        else if (tutorialNo == 9)
        {
            screen.GetComponent<Renderer>().material = scriptBT.menus[5];
        }
        else if (tutorialNo == 10)
        {
            screen.GetComponent<Renderer>().material = scriptBT.menus[2];
        }
        else if (tutorialNo == 11)
        {
            screen.GetComponent<Renderer>().material = scriptBT.menus[6];
        }
    }
}
```