INTERACTIVE 3D MODELING IN VIRTUAL REALITY

by

Darius Bigbee

A Thesis

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Master of Science

Department of Computer Graphics Technology

West Lafayette, Indiana

May 2019

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Esteban Garcia, Chair

Department of Computer Graphics Technology

Tim E. McGraw

Department of Computer Graphics Technology

Davin H. Huston

Department of Engineering Technology

Approved by:

Nathan Hartman

Head of the Graduate Program

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Author: Bigbee, Darius, L. MS
Institution: Purdue University
Degree Received: May 2019
Title: Interactive 3d Modeling in Virtual Reality
Committee Chair: Garcia, Esteban

Many applications have been developed for Virtual Reality (VR) during the new wave of VR technology. These new technologies make it possible to create 3D meshes in a virtual environment in real time. However, the usability of VR as a modelling tool is still a new area of research. This study's research created a VR 3D modeling tool that will provide the user with tools to interactively generate and edit 3D meshes in real-time and teach the users how to create 3D models. The study had two groups of participants, one group used Autodesk Maya, and another used the VR modeling tool. All participants were from Purdue University and all data was collected in the Polytechnic Institute. Both groups were given a task to create a teacup with the time it took to complete it recorded. The VR tool was evaluated with a SUS (System Usability Scale). The participants provided feedback and rated how difficult it was to use the application. With the SUS, it was determined that the application did not meet the industry standard average score of 68. However, further analysis on users' responses showed many areas to improve in the application. A few recommendations for future research include implementation of multi-selection, a undo and redo feature, and improvements of how the user interacts with the 3D meshes.

Keywords: 3D Modeling, Virtual Reality, Mesh, Geometry, Unreal Engine

CHAPTER 1. INTRODUCTION

1.1 Scope

3D applications such as Google Blocks and Tilt Brush allow creation of 3D models, but they are limited to the number of polygons that can rendered. These programs do not have many features and tools that allows for flexibility in generating 3D geometry. The goal is to create and test a program that can, generate meshes with minimal impact on performance and increase the flexibility so that any user is able to create their own geometry for any type of project. The VR application must be able to render the mesh in real-time to avoid lag that may cause the application to become unusable during the study. The participants that the study involves will be those who do not have much experience with 3D modeling. This will determine how well it is useable for any person who wants to use it in the future. The application will open the possibilities for future improvements that a researcher and/or developer might find useful.

1.2 Significance

In 3D modeling software, such as Autodesk Maya and 3ds Max, the process can be slow because the user must shape the mesh by manually manipulating vertices, edges, and faces with a mouse. The applications include tools to help the user quickly make changes to the geometry, such as beveling and extruding. The process of creating geometry in a program like ZBrush, which is revolved around sculpting, can be faster. The mesh, however, would most likely be unusable due to the high poly count (usually with millions of triangles for a single mesh). A high poly count can make it difficult for a mesh to be used in real-time, since it would require more time for the screen to render out a single frame which would produce lag.

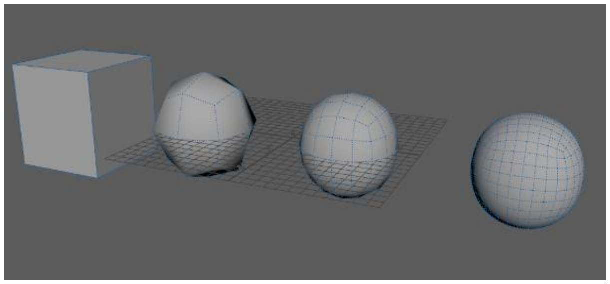


Figure 1. Mesh Poly Counts: 6, 36, 144, and 576

The topology would also make the mesh difficult to work with when texturing and rigging for animation. A mesh produced in ZBrush would require taking the mesh to another program to be optimized depending on its uses. The application produced for this study would be like Autodesk Maya and 3ds Max, in terms of functionality. VR gives the user the ability to simulate interactions with 3D objects, giving the user the illusion of being able to physically touch the objects, especially when implementing the sense of touch and sound. The significance of this study is to give the user a hands-on approach to creating and manipulating meshes to their needs using VR, while giving the user a different perspective when working.

1.3 Assumptions

The study will involve students that have little knowledge about 3D modeling. This is important because, the data analyzed can show how steep the learning curve is for new users. The VR application would need to be functional enough, so the users doesn't experience bugs or any other problems that might be present. Students know beforehand what VR is and how to use the VR equipment. The survey given at the end of the testing will be answered truthfully by the user giving accurate data. Since the identity of each participant in the study will not be taken or known, the tester can assume that the user's response will be honest.

1.4 Limitations

Participants who experience motion sickness when using the VR equipment may not complete the study. Only students of Purdue will be used as participants in this study. Most participants will also be enrolled and present in Geometric Modeling for Visualization & Communication (CGT 116) or Fundamentals of Imaging Technology (CGT 118) courses. The goal is to get participants who have little to no experience with 3D modeling. This includes both undergraduate and graduate students but there may be a few exceptions. For example, students in the undergraduate level (below 200 level) may have done some 3D modeling on their time outside of class that the tester may not be aware of. There is no guarantee that the user's opinion before using the application will be unbiased. Students may not know how to use the VR equipment and thus may have difficulty with using the application. The survey given could also limit the type of responses that are given from the participant, since they are limited to the text of the questions, statements, and categories of the survey itself.

1.5 Delimitations

The study will involve Computer Graphics Technology students who are currently enrolled in Purdue University. The testing of this study will be conducted at the Polytechnic Institute of Purdue University. Any students who are outside of Purdue and/or not in the classes mentioned will be excluded from this study.

1.6 Definitions

Autodesk 3ds Max, Maya, AutoCAD – the name of software trademark by Autodesk ("Autodesk | 3D Design, Engineering & Entertainment Software," n.d.)

ZBrush – a 3D sculpting software by Pixologic ("Pixologic : ZBrush - The all-in-one-digital sculpting solution," n.d.).

Head Mounted Display (HMD) – a typical virtual reality display system (Ma et al., 2009).

Mesh – a series of faces that share two or more vertices (Thomas Kohler, Johann Fueller, Kurt Matzler, & Daniel Stieger, 2011).

Polygon – a surface constructed using three or more points known as vertices (Thomas Kohler, Johann Fueller, Kurt Matzler, & Daniel Stieger, 2011).

Topology – the arrangement of polygon edges on a surface (Thomas Kohler, Johann Fueller, Kurt Matzler, & Daniel Stieger, 2011).

Vertices – points along the edge of polygon faces, usually at the intersection of two or more edges (Thomas Kohler, Johann Fueller, Kurt Matzler, & Daniel Stieger, 2011).

Virtual Reality (VR) – a dynamic 3D scenery modeled through computer graphics techniques (Pinho et al., 2002).

1.7 Summary

The study will include students that are in CGT 116 and 118 at Purdue University. The data gathered and evaluated will determine the usefulness of the application as a future VR 3D modeling tool for anyone who wants to learn 3D modeling. The study aims to create an application that is useful to users that do not have initial skills in 3D modeling but want to create their own models and/or 3D print.

CHAPTER 2. REVIEW OF LITERATURE

Identifying the right mechanics and features to include increases the usability of the application. For example, testing the time it takes for a user to model in VR versus 2D and how well they can adjust and use the tools provided. Studies have shown that modeling in a 3D workspace is faster and a more intuitive way to work than the traditional 2D workspace (Jackson & Keefe, 2016, p.1443). This chapter will discuss past software that used VR for 3D modeling, the possible mechanics that are needed to the application useable for 3D modeling, how user interface affects the user experience, interactions and different methods of navigating in a VR environment.

2.1 Previous Virtual Reality Applications

Virtual reality is a developing technology that has improved greatly over the past decades. Many VR applications have been developed, that changed the way VR is used. VR has been implemented into software like Tilt Brush, Drawing on Air, Cave Painting, Lift Off and Google Blocks. Recently Unreal Engine 4, a game development software, has even implemented VR into its engine, allowing developers to edit the 3D world in a VR space. A few of the mentioned software will be described further in the following sections.

2.1.1 DDDoolz

DDDoolz is a 3D voxel desktop sketching tool that explores the uses of VR in its development stage. DDDoolz was developed at Eindhoven University to be used for educational architecture purposes (Achten et al., n.d.). The purpose of the software was to make the creation of 3D models easy to manipulate in a VR environment. What the user experienced using the application determined whether the developers could use DDDoolz in the way it was intended to be used and achieve an appropriate outcome. The software was then used in a first-year CAAD (Computer Aided Architectural Design) class at Eindhoven University that ran for the first eight lecture-weeks of the year. The course was used to provide an understanding of CAAD and the uses of AutoCAD with an introduction to design, design processes, methods, and design thinking (Achten et al., n.d.). The sample of the study were beginning students. The course was split into two structures, two hours of CAAD and design theory, and four hours of exercises a week, with a different exercise each week. In the beginning, students did however, find the exercises given to them were too vague for them to self-evaluate designs, and only a limited number of the students were able to submit a finished design. In the last exercise the students could use DDDoolz in a short time and use all the tools available to them, such as being able to import Autodesk 3D studio (.3ds) models. For research purposes, DDDoolz serves as a platform to test various input devices, such as mouse controls, keyboard input, both keyboard and mouse, voice controls, flock-of-birds 6D mouse (a type of advanced gaming mouse), and also a pen and varied combinations of each. (Achten et al., n.d.). With the development of DDDoolz, the developers wanted to study 6 things from it: its usability for programs like AutoCAD, the overall design of the user interface, creating and manipulating 3D models in the virtual environment, navigation in the environment, use as an educational tool for students, and the future development of DDDoolz (Achten et al., n.d.). With those goals in mind, the future of the software can incorporate the findings to improve development of 3D VR software.

2.1.2 Lift-Off

Lift-Off was another 3D modeling software that used VR to create 3D models. The purpose for this software was to challenge the problems with modeling in VR: the limitations it puts on a user’s creativity and the software’s applicability. Lift-Off uses both 2D and 3D to generate 3D models. The process involves the user to begin with a 2D sketch that is imported into the application. With curves (Hermite) and an image processing algorithm, the curves are interactively lifted and used as a scaffold hold for the 3D model. The user would then sweep along the curves to create a surface for the 3D model (Jackson & Keefe, 2016). The user would have either motion controllers or a stylus to control where to place points. The software would then take each point and generate a curve along the path of each point. With those curves in place, a 3D model is generated. The use of curves gave the user more control and improved the ability to freehand 3D sketching. Immersive 3D modeling software has started to use sweeping, freehand, and gestural input types as the main way of creating 3D models, such software include HoloSketch, FreeDrawer, Surface Drawing, CavePainting, BLUI, Drawing on Air, and DragDrawing (Jackson, 2016). The developers of Lift-Off believed that 3D modeling in VR enhanced the artist spatial perception and give the potential to avoid depth ambiguity problems. For software like Maya and 3ds Max, the learning curve is quite high and usually takes the user time to learn all the tools and features available to them.

2.1.3 Google Blocks & Tilt Brush

Google Blocks and Tilt Brush are the more recent VR applications that was created to allow its users to share their creativity. Both applications were developed and published by Google. Tilt brush is an application that allows artist to draw and paint in a 3D virtual environment. Tilt brush was released for Windows at the launch of the HTC VIVE and the Oculus Rift on February 24, 2017. Tilt brush can be used with a mouse and keyboard alongside the motion interfaces made for VR. The user can choose from a wide range of brushes and colors that are presented to them on a virtual palette. Brush strokes are generated from the movement of the handheld controllers (“Tilt

Brush by Google," n.d.). Tilt brush allows its users to share artwork as a room-scaled VR piece or as animated GIFs.

Google Blocks on the other hand is an application that was also released for the HTC VIVE and Oculus Rift and its focus was on 3d model creation in a VR workspace. It is stated on the Google Blocks webpage that creating models for a 3D world should be created in a 3D workspace, just how 2D art is created in 2D workspace. To give the user control, Google Blocks features 6 simple tools: shape, stroke, paint, modify, grab, and erase. These tools give the user the ability to insert, move, paint and modify shapes to create simple and complex models that can and/or publish to share with others. With Google Blocks, just like Tilt Brush, the user can move around the model as if looking at an object in the real-world. The models created with Google Blocks can be shared as object files (.obj) or animated GIFs as well.

2.1.4 Interactive swept surface modeling with motion-tracked controllers

Another application, created by professors at Purdue University, used VR to create a 3D modeling tool that used swept surface modeling to create geometry. The software used gestural motion and Hermite curves controlled by the motion controllers to create and modify geometry. The VR equipment used in this program allowed viewing and interactions with objects without the need for a keyboard and mouse. The aim of this application was to create a 3D modeling tool that would be similar to sculpting with clay and plastic (McGraw, Garcia, & Sumner, 2017). This application used the controllers to create a curve with the controller's position, controlling the curves endpoints and the orientation controlling the curves tangents. As the user moved, the controller's position and orientation changed the shape of the curve, which generated a segment of the mesh. The curves were created in a certain threshold to keep the curves from stacking as the controllers were moved. The curves created by the user allowed the formation of smooth swept surfaces, allowing organic modeling in VR.

2.2 Limitations of VR modeling Software

One limitation that some of these programs face, such as Google Blocks, are that models were limited to a low polygon count that does not allow for very detailed meshes. Other problems were implementing a user interface that would give the 3D modeler a feeling of immersion and the ability to navigate naturally around the environment. The limitations that Lift-Off faced was that the curves created by the user could no longer be edited once placed. The increase of fatigue from the user also plays a role, which can limit how long the user can keep using controllers to design. For Lift-Off a stylus was used to provide as much comfort as possible to the artist (Jackson & Keefe, 2016).

2.2.1 User Experience and Interface

One important thing for any software is to provide a great experience for the user. In terms of 3D modeling software, the user interface can determine what the user experiences. The creation of DDDoolz was developed with a simplified user interface in mind. When the program was first created there was no interface included but that was mainly because the developers thought that an interface less application would be suited for VR (Achten et al., n.d.). But afterwards, an interface was added to DDDoolz in the later versions to make it more suitable for the students that it was intended for. “In general, the interface supported seems to support a fast learning curve” (Achten et al., n.d., p. 7).

2.2.2 Interactivity

The monitor in which the user views the VR environment is the boundary between the real and digital world. The interaction and realism affects the amount of immersion the user experiences, if the interactivity is instant, the user will feel that the interface is alive and gives them a greater feeling of reality (Pinho et al., 2002). To make the user immersed, interactive frame rates and response time should be taken into consideration. A counteraction that was created for visualizations for interactive data was to reduce the visualization data to a degree that the system could handle (Ma, Gausemeier, Fan, & Grafe, 2009). In *Design of a Model of Human Interaction in Virtual Environments*, (Ma et al., 2009), they presented a generic model for human interaction in VR environments. They wanted the interactivity to go further than touch and movement but also have complex actions like in the real world. They took into consideration the scalability, extensibility, and ease of integration. The design of the software was used in the MAEVIF platform for Intelligent Virtual Environments for Education and Training. MAEVIF is a platform used for the development of education and training systems based on a VR environment (Ma et al., 2009). The platform is made up of two parts, an Agent-Based Intelligent Tutoring Subsystem (ABITS) and a Graphic and Interaction Subsystem (GIS). The ABITS is the central portion of the MAEVIF platform and GIS is the program that runs in the user's terminal (Ma et al., 2009). The agents of ABITS share and analyze information about actions performed in the VR environment. Those actions were evaluated to see if the objectives were done and are considered abstract. What this mean is not only is it important to know if an action was executed (the user opens a door) but also how an action was executed (the user forcibly pushes open the door or the user opened the door with his/her hand/shoulder). (Carlos Jerónimo J., Antonio A., Méndez G., Ramírez J, 2009). This is quite common in 3D modeling, where there are many tools provided but, ultimately there are many ways the user can reach his/her objective when creating a mesh. With many interaction devices available, the model would not be dependent on any one device. It stated to support any kind of device that would allow the user to perform their actions.

2.2.3 Navigation

Navigation in VR is a mechanic that is still being developed and experimented with. There are different methods of navigation that have been implemented into VR games and software. Those methods of navigation include having the user stay in place and a controller is used to navigate the VR environment, the player controls a character and moves him/her through the environment (first person VR game), or have the player move with the help of some device such as a treadmill or bicycle, that gives the user the feeling of movement while keeping them in place. In the case of DDDoolz, the software handled navigation by using an invisible viewpoint with a specified viewing direction that is attached to the mouse that was used as the default navigation tool (Achten et al., n.d.). Usually navigation is done with the use of buttons and/or key sequences and the use of metaphors for example. However, this would make the interaction look man-made and ineffective. In VR, simple tasks, such as looking, can be very complicated to create and look unnatural (Pinho et al., 2002). When the navigation needs to be fast and life-like, problems began to occur with the tools that are available.

In *Cyberpsychology & Behavior*, the writers discuss how the model they created wanted to accomplish the interaction between computer and human in a more direct way. Their idea was to map the user’s gestures to the movements inside the virtual world without the use of metaphors and sequence of buttons for users to interact with the objects in the world (Pinho et al., 2002). This model was used because as stated in the article, earlier studies had shown that for a complex VR application a generic interface is difficult to use. For virtual assembly, other examples of navigation were implemented for product developers. In virtual assembly the user could select assembly tools, grab components, and perform the trial in a virtual assembly environment. The user was able to use different types of interactions modes at different stages of assembly, which included hand interaction, speech, virtual menu, dialogue, etc. (Ma et al., 2009) In the interaction mode that used the virtual hand, a data glove and position tracker was mounted on the user's hand, gave the user the ability to interact with virtual environment, while also having a walk-through mode. This mode had two approaches of navigation, one was to use a flock of bird and mount it onto the user's head, tracking the user's head movement, and the other, with a Neowand, which is a device with a small patch of button, gave the user the ability to pan around the viewport using the up, down, left, and right buttons (Ma et al., 2009).

Another important aspect of VR is that the environment is a dynamic system (Pinho et al., 2002). A dynamic system is a system where the scenarios in a 3D VR environment are modified in real-time, such as the user's interactions and the objects placed in the world. A VR environment is basically a projected simulation that is modeled based on the real world as much as possible. Immersion could be important for a 3D modeling program to make the artist more connected to their work. Many other VR applications try to immerse the user so that they are interested and connected with the applications. An immersive game would give the player the feeling of being inside the game themselves and using VR would be the best approach when creating an immersive application.

2.3 Summary

Making the application immersive is an important focus when creating this application and that focus can determine if the users will have the ability learn 3D modeling using a VR application. The literature shows that there are many ways to implement a user interface for VR, but the solution depends also on the use of the software and what makes the user feel more comfortable and thus increase the applications usability. It also shows that there are many techniques to implement navigation into an application.

CHAPTER 3. METHODOLOGY

3.1 The Framework

The study collected data via a Qualtrics survey, in order to get an idea of what the participant experiences during the study and how they feel about the usefulness of the application. The variables that will be tested was the usability of the application and the difficulty of the given task. For this study, the results will be documented using the SUS, because it allows the evaluation of many types of products and services including software. The SUS is an industry standard, being referenced in over 1300 articles and publications(Affairs, 2013). The reason for using the usability test instead of a homemade questionnaire is because it can be easily given to participants, can be used on small sample sizes, and is valid (Affairs, 2013). SUS is also used for post-test questionnaire purposes and since this research uses VR, a post-test would be more beneficial to use rather than post-task because the user would have to answer a questionnaire after every task and this study only has one task. If there were many tasks the participant would have to take off and then put back on the equipment many times during the study, which could have fatigued the user quickly. Figure A.1 shows an example of the SUS survey given to the participants. The SUS consists of 10 Likert-scale questions which gives a score between 1-5. The survey includes the following questions:

- Which group the subject was in.
- Survey
    - I think that I would like to use this system frequently.
    - I found the system unnecessarily complex.
    - I thought the system was easy to use.

  - I think that I would need the support of a technical person to be able to use this system.
  - I found the various functions in this system were well integrated.
  - I thought there was too much inconsistency in this system.
  - I would imagine that most people would learn to use this system very quickly.
  - I found the system very cumbersome to use.
  - I felt very confident using the system.
  - I needed to learn a lot of things before I could get going with this system
- The amount of time it took to complete the task.
- What they would like to see improvements on, free response.
- How the difficult the task was to the participant.

Another benefit from using the SUS is that the score can be easily benchmarked with other scores of peers and competitors. With the test chosen for this research in mind the hypothesizes for this research include:

H_0: The application has a high usability as a 3D modeling program for VR based on the evaluations.

H_a: The application has a low usability as a 3D modeling program for VR.

H_1: The provided tasks were not difficult to complete for the participant.

H_{1a}: The provided tasks were difficult to complete.

3.2 Building the Application

Unreal Engine, created by Epic Games, was used for this study is because of its easy to use interface, has prebuilt objects that would make building a useable program easier, its visual scripting that makes coding faster, as well as the option to use C++ to create new objects that may not be implemented already. From Unreal, one of the default templates that will be used is the VR template which would ease the complexity of trying to implement VR into a visual studio C++ project. With the VR template, there are predefined settings that improves the overall performance when the program is finished or shipped.

The most difficult task to complete was choosing a starting point to build the application. Since Unreal has its own editor and its own way of rendering, a lot of time and research went into looking at the API documentation. The C++ classes found in the editor all derive from a class called UObject, which was created by Epic Games. This class is a default class that is used by the engine's garbage collection (which is a process to delete unreferenced objects created in code to free computer memory) and other important aspects of the engine. Looking at the static mesh documentation helped in some areas, such as locating the data for the vertices, normal, and UVs. With this information, an editor module was created that allowed the visualization of the vertices. But unfortunately, the meshes themselves could not be modified without major changes to the engine code. Looking at how some components were rendered in Unreal helped create a new class that rendered out a dynamic mesh that could be modified in real-time.

The dynamic mesh component implemented a link-list data structure, otherwise known as Doubly-Connected Edge List or DCEL for short. These are the few different types of data-structure that 3D programs use:

Vertex-Vertex - Stores the location and other vertices connected in the same face.

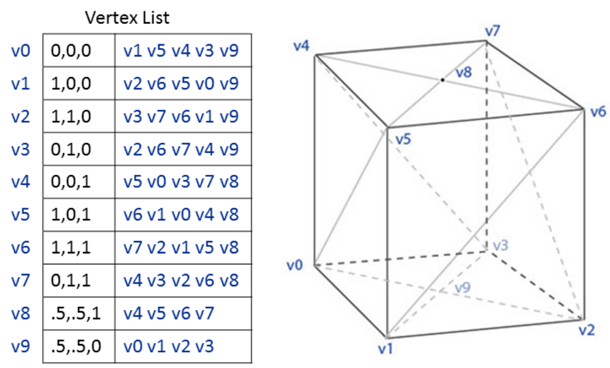


Figure 2. Vertex-Vertex Mesh

Face-Vertex - The Face stores the vertices that belong to it and the vertex store its location and the faces it is referenced in.

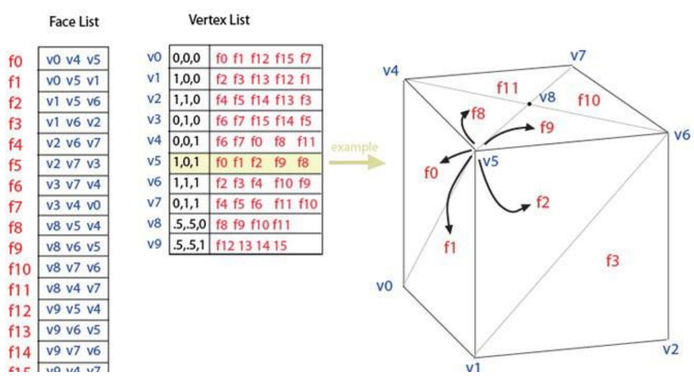


Figure 3. Face-Vertex Mesh

Winged-Edge - This is much more complex with the addition of edges; the face stores the edge index. The edge stores the vertices that makes the edge (total of 2), the faces it is referenced in and the other edges connected to the same face. The vertices store its location and the edges it is referenced in.

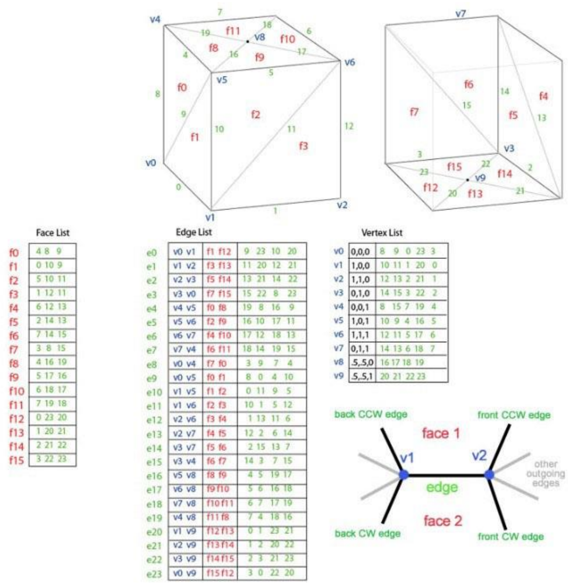


Figure 4. Wedged Edge Mesh

Doubly Connected Edge List (DCEL) – In this structure the vertex stores the half edge it is the origin of, the face stores one of the half edges that creates it, and the half edge stores the origin, the next and previous half edge and the face it is a part of.

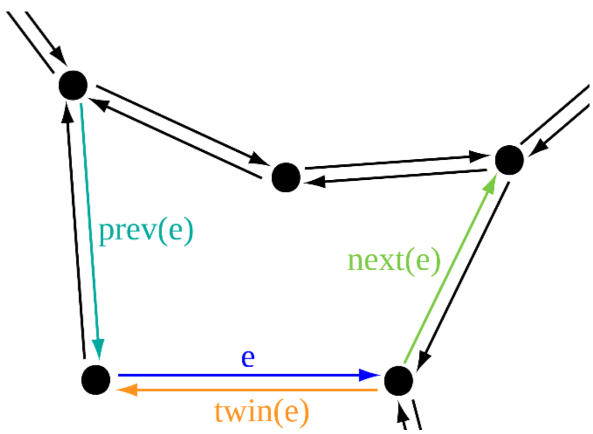


Figure 5. Doubly Connected Edge Mesh

From those, the DCEL was the best representation of a mesh to use because the amount of information stored would take up less computer resources while keeping enough information to easily implement functions to keep track of changes and modify the mesh. Creating a new component gave the application the ability to implement the DCEL structure and the freedom to determine how the geometry will render and be modified in real-time. With DCEL, rendering became as simple as having a list of faces that had a reference to one edge that made up that face, and with that reference loop to the next edge that the first edge reference and keep going until reaching the first edge again and breaking the loop. DCEL allowed for more control to find any edge, vertex, and face that would need to modify in the given mesh. For the controls both controllers were used for different functions, the left controller had the menu functions while the right controller had all the function for the interactions with the objects and points. The trackpad on the left controller was used to navigate the menu and since there were sub menus the trigger was used to go back to the home menu. The right controller had more functionality than the left, the grips were used to grab the objects and points, the trigger was used to select the points in the different editing modes, and the trackpad was used to navigate the VR environment. There was also the application menu button that was used to toggle the mode on the right controller in order to toggle interaction with the UI elements.

3.3 Sampling

The study had a sample size of 48 participants. Most of the students were from CGT 116 and 118, a few others were included in the study to get a wider range of results. The group that use Maya consisted of 15 participants while the group that used the VR application had a sample size of 32. The participants were able to schedule the time that wanted to the study since it was voluntary to participate.

3.4 Testing

The sample was split up into two different groups with the first group using Autodesk Maya. The instructions given to each participant were a set of steps that each user used in order to create a teacup, starting from a simple primitive to the complete model. To keep the tools used as consistent as possible in Maya and the VR application, the steps used tools that were like the functions created in the application. The participants were timed from start to finish and that time was then recorded in the Qualtrics survey. The second group used the VR application. The devices the user used was the HTC VIVE. This group completed the same teacup task as the other group. They too were given a set of instructions to complete the task. After completing the study, each user completed the Qualtrics survey, which included the SUS, to give feedback about their experience with the software, and any improvements the participants suggested. The testing area was setup in a room located in the Polytechnic Institute. Figure. 6 shows the setup of the room, the test area was approximately 8 feet and 2 inches by 11 feet and 7 inches in size, with the sensors for the VIVE placed approximately 8 feet and 2 inches from the floor. This gave the participants enough room to work, they also were given the choice to stand or sit while using the application.

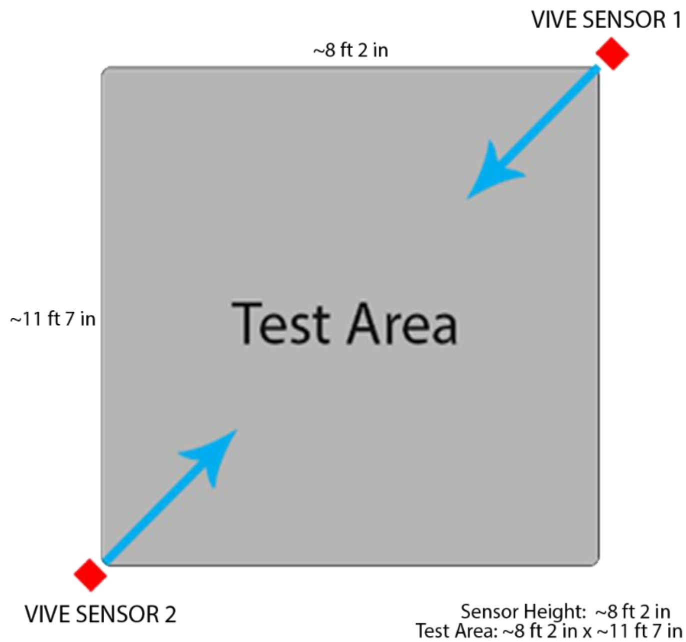


Figure 6. Testing Area

3.5 Data Analysis

Data was collected using a Qualtrics survey that the students completed after completing the task provided to them. Since the study is using SUS, there is a specified way the score must be calculated. First each SUS question, the choices are scored from 1-5, with Strongly Disagree at 1 and Strongly Agree at 5. The answers were then converted by taking each question that is odd numbered and subtracted by 1 and the even numbered questions were subtracted from 5 since those are considered the negative worded questions. This gave each user a score of 0- 40, after the scores are added to together. The converted scores were then multiplied by 2.5 to give an overall score between 0-100. The overall score is not based on a percentage but a percentile. The score will then be normalized to produce a percentile ranking, scores above the average score is considered above average while score below are below average (Brooke J., 1996).

3.6 Summary

The methodology covered the usability tests that will be used for this study. It also determined the best strategy to gather and interpret the data to get the best overview of the usability of the software. The next chapter will discuss the results that were found from the data collected through the survey.

CHAPTER 4. PRESENTATION OF THE DATA AND FINDINGS

This chapter will discuss the results from the data collected. The data collected included survey scores form the questions answered, the time it took for both groups to complete the task, and the score how on difficult the user felt the application was.

4.1 Results

4.1.1 Time

For the first group of participants only the time to complete the teacup was collected and the results from the data determined that the average time to make a teacup in *Maya* was 21 minutes and 20 seconds. The minimum time it took a participant was 6 minutes and 20 seconds while the maximum time was 34 minutes and 56 seconds. Table 1 shows the times that were gathered from the 15 participants.

Table 1. Maya Times

20:47	20:40	21:07	21:00	06:20	20:50	25:35	34:56
13:33	22:07	23:00	24:42	26:28	27:27	11:35	

As for the second group of participants the times were also recorded as shown in Table 2. The minimum amount of time it took a participant to complete the task was 7 minutes and 56 seconds and maximum time it took was a total of 37 minutes and 26 seconds, with an average time being 21 minutes and 8 seconds.

Table 2. VR Times

20:24	32:23:00	37:26:00	19:00	10:00	16:30	26:53:00	23:00
20:48	10:00	8:37	26:00:00	17:50	7:56	31:11:00	16:29
26:05:00	15:30	23:06	14:05	21:24	28:02:00	24:32:00	19:27
20:43	36:09:00	24:00:00	12:00	19:02	23:15	21:17	23:05

4.1.2 Survey Results

Other than the times that were recorded, the second group had scores collected from the SUS portion of the survey. However, due to human error the last question ("I needed to learn a lot of things before I could get going with this system") that is found on a SUS survey was not added to the Qualtrics survey. The scores that are shown are set in a range if the last question was answered with strongly disagree or strongly agree for all responses.

Table 3 and Table 4 shows score for all the participant. The percentile rankings for Table 3 participant are shown in Figure 8 and the rankings for Table 4 are shown in Figure 9. Table displays the scores if the last question was answered with strongly agree while Table 4 shows the scores if the question was answered with strongly disagree. Looking at Table 3, the average SUS score was 33.98 which is well below the industry SUS average of 68. The minimum score was 15 with the maximum being 57.5 and had a median of 30. 12 participants were above average while 20 were below average. 53% of the participants were below the 90th percentile and 43% were below the 75th percentile.

Table 3. SUS Scores (Strongly Agree #10)

32.5	30	25	40	37.5	30	55	57.5
52.5	30	47.5	20	20	30	27.5	15
42.5	32.5	22.5	25	40	52.5	55	25
25	20	27.5	42.5	22.5	47.5	25	32.5

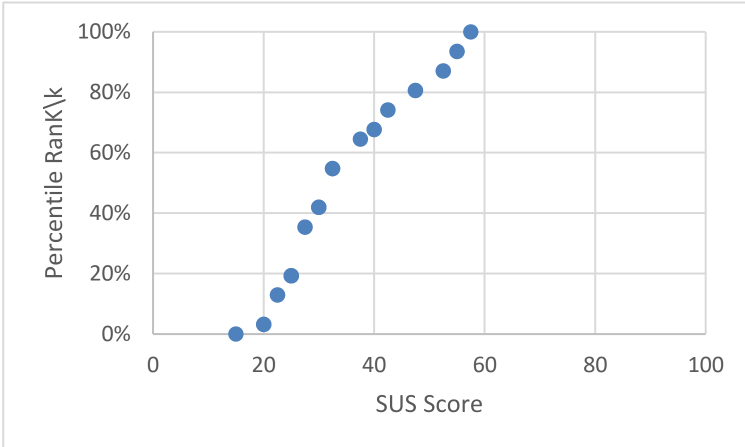


Figure 7. Percentile Rankings (Strongly Agree #10)

Now looking at Table 4, if all the answers for question 10 were answered with strongly disagree the average SUS score was 46.48 with a minimum of 27.5, maximum of 70, and a median of 42.5 with the same number of students both above and below the average. With that, the average SUS score was in the range of 33.98 – 46.48 which unfortunately is still below the industry standard average of 68. 68% of participants were below the 90th percentile, 55% were below the 75th percentile.

Table 4. SUS Scores (Strongly Disagree #10)

32.5	30	25	40	37.5	30	55	57.5
52.5	30	47.5	20	20	30	27.5	15
42.5	32.5	22.5	25	40	52.5	55	25
25	20	27.5	42.5	22.5	47.5	25	32.5

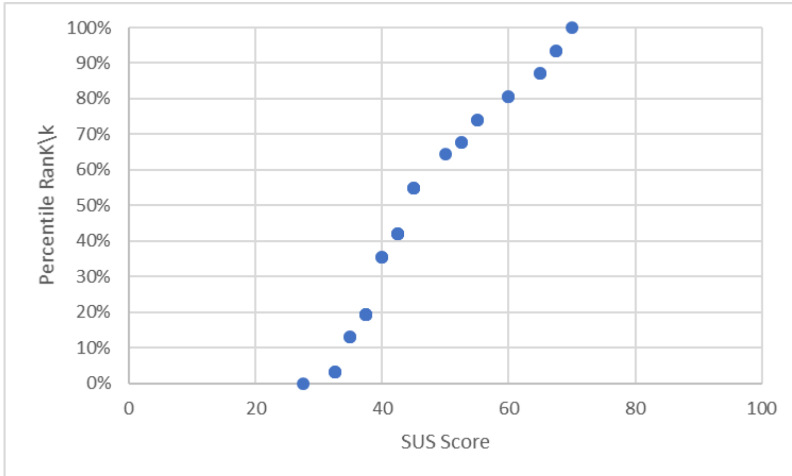


Figure 8. Percentile Rankings (Strongly Disagree #10)

The participants also rated how difficult the application was to use at the end of the Qualtrics survey. A rating of very easy was counted as 0 and very difficult was counted as 4. The average rating for the difficulty of the application was 2.5.

4.2 Problems Encountered

There were many problems that were encountered during the testing phase of the research. One problem was the equipment used to measure the time for the participants to complete the given task. A simple stopwatch used on the clock app on windows 10 and the stopwatch application present on android phones made it difficult to move back and forth from the program to the clock since the tester had to switch from each program in order to stop and start the time. This made it also difficult to have a consist start and end time even though the time difference were only a couple seconds. In some cases, the tester also forgot to start the time, since it took more time than others to explain how the controls worked and the application and task itself. There were also encounters with glitches that made it impossible for some users to complete the task and few had to restart, one glitch caused the collision on the menu to conflict with an invisible collision that was impossible to determine the cause but, may have been caused by the add vertex function in the application. The final problem was the missing of a question from the survey the tester created but somewhere between the creation of the survey and the beginning of testing the question may have been removed while recreating the survey for the purposes of gathering data, which was not realized until after the data was collected. It would have been beneficial to have the participants retake the survey but due to time constraints, the previous times collected, and participant confidentiality it was problematic to have the participants to redo the surveys.

4.3 Task Comparisons

Shown in Figure 9 are examples of the teacups that were created inside of Maya and the VR application. The teacups created shows a similarity in the structure and design, this suggested that the VR application was useable to complete the task given to the participants in about the same amount of time. In some cases, the teacups created in the VR application had a better design then the Maya versions, not including the smoothness and density of the later since subdivision was not implemented. Features that were available to the participants included: four different edit modes: object, vertex, edge, and face, extrusion of edges and faces, deleting faces, creating faces, and the merging of vertices. The polycount for the teacups created with Maya were about 490 - 492 triangles, while the polycount for the teacups created in the VR application was about 400- 420 trianlges which depended on the level of detail placed in the handle. The participants who used Maya did have more ways to create the handle than the VR users did.

4.4 Summary

The data gathered from the surveys was used to determine the usability of the application as well as compared how well the participants did. The next chapter will discuss the problems encountered, the conclusions the data represented and the future research that should be done based on the feedback from the participants.

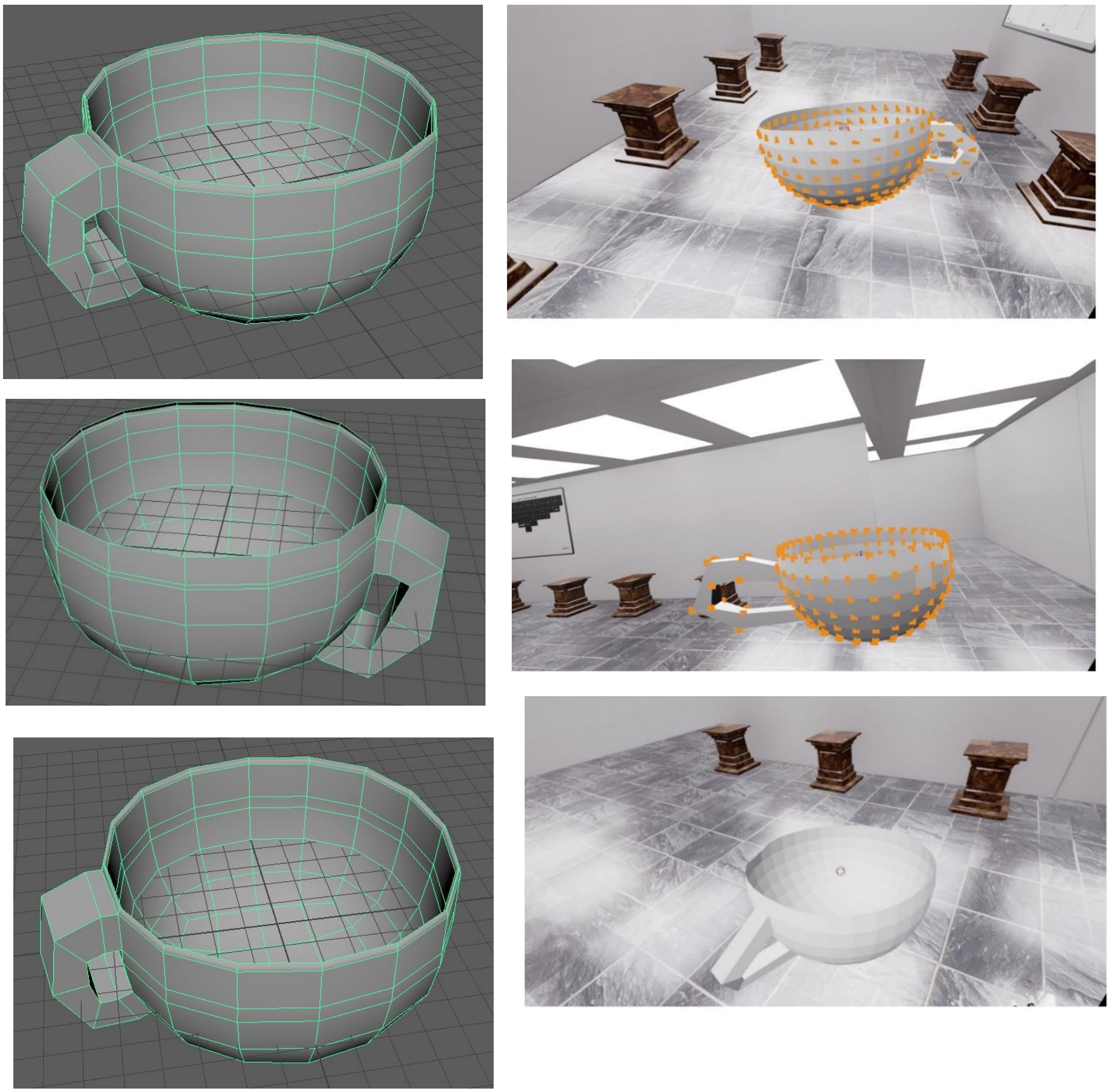


Figure 9. Teacups created in Maya (Left). Teacups created in the application (Right)

CHAPTER 5. CONCLUSIONS AND RECOMMENDATIONS

This chapter will discuss the potential problems that were encountered with the application and the ways to fix these problems for future research. It will also the possible improvements and fixes for problems given from the responses from the participants.

5.1 Interpretation

From the results, the application was not user friendly since the average was well below the industry standard average SUS score of 68. This tells the researcher that more focus needs to be done to make the user interface for the application easier to work with. The percentile ranking shows the SUS scores collected on a grading curve and with many of the scores well below the average, there are major problems with the usability of the application. To get a better understanding of the scores it might have been best to have a SUS survey for the Maya group as well. Having a SUS survey for the that group could have given the score a better comparison than the industry average, since both groups were filled with mostly novices to 3D modeling. With a difficulty average of 2.5 out of 4, overall the participants did find the application somewhat easy to use. With average times of 21 minutes and 20 seconds (*Maya*) and 21 minutes and 8 seconds (VR), we can conclude that the time it took to complete the teacup in *Maya* and in VR took roughly the same amount of time.

5.2 Future work

The feedback gathered from the participants shows the problems that future researchers should investigate to make the application more user friendly. A few problems or suggestions made by participants focused more on the usability of the application and how the interactions worked with the different object modes.

5.2.1 Improving Selecting

There was no way for the users to multiselect points, which was not a feature, and many participants felt that it should have been. The participants had to select each mesh component one at a time in every edit mode, selecting in edge mode was even more problematic since the collision for each line was a sphere at its midpoint, making them hard to select. One participant mentioned, “Multi-select or drag-selecting is largely what I was aching to have, so I could get a lot of messy modeling done fast; I'm so used to selecting many things at once with computer files, text, etc., that it's a carry-in desire that I have”. The selecting of each individual point made it feel tedious to participants and increased the amount of time it took for participants to select many mesh components. A couple ways to implement multi-selection is to select points just by holding the trigger while sweeping over them instead of pressing the button on each one or incorporate a way to draw a selection box around mesh components on the current mesh that is selected only as suggested by a participant. As for the edges that were difficult to select, it would probably be best to use a collision box instead of a sphere. The collision box could be subclass just like the sphere and only create these when the mode is set to edges and this could also apply to faces to allow selecting and grabbing them easier. Adding vibration feedback on the motion controllers to know whether they are hovering over a point could also make it easier for users to know if they are above a mesh component. Already given a way to select the borders (edges not connected to another edge) of a mesh, other ways to expand the current selection features would be to give users the ability to select loops and rings for edges and faces.

5.2.2 Improving Grabbing

As mentioned, grabbing edges and faces was difficult for participants and a possible solution was suggested. The participants could only grab one mesh component at a time, so moving multiple points at a time should be added to application by using some sort of array tracking each selected point and then offsetting their position by their distance from the grabbed point. Grabbing the mesh itself was also difficult since the collision used for the mesh was a small sphere with a radius of 3. The sphere was smaller than the mesh because of conflicts with the collision on the points. If the collision was applied all over the mesh, trying to grab a point would have caused the participants to grab the mesh instead. One possible solution to this would be to allow grabbing the mesh only available in object mode by disabling collision and then reenabling when back in object mode. The collisions for the points are only created when the mode is set, which would result in no conflicts with collision detection if this solution is applied to the application. On top of that, it would also benefit participants to use complex collision for the mesh instead of using a collision sphere to allow for more accurate grabbing. The controls should also be changed from the grip to another button since many participants did not realize the controllers had grips on the sides and some found it uncomfortable to grab that way. As of now only the position of the controller affects the mesh components when grabbed and allowing the orientation to as well should be added.

5.2.3 Undo and Redo Feature

Many participants felt that an undo and redo feature should have been implemented into the application. There was the possibility to delete a mesh and restart when the participant wanted to redo the teacup from the beginning but allowing them to take a few steps back would have been much better. The original plan was to add such feature but due to time was not implemented into the final version. To add this feature, the recommendation would be to keep track of each state of the mesh when its being modify and when undone set the mesh's state to that saved state and only delete states after that if the mesh is modified after being undone. The same would also apply to the redo feature to reload the next state if the mesh was not modified beforehand. Keeping track of each state could be done only for the current mesh but that could cause problems if there are multiple meshes and the user wanted to go undo a mesh later. So, a solution to this would be to have each mesh or a class keep track of the mesh's states. Also, important to note is add a way to clear the history of the mesh at some point to free up computer resources as well, by having a max amount of states allowed and removing the older states as more are added after the max and give the user the ability to clear all history like other software.

5.2.4 Other Possible Features

There were many other points that were brought up that could possibly be implemented into the application. The list below shows many of the feature's that the participants suggested: more well guided instructions and tutorial videos for the participants to follow along with, add a 2D view mode for orthographic modes, trace movements in a drawing mode, constraints of the points while moving (possibly through a widget), simpler way to create and delete objects, ability to zoom in and out of frame. Fortunately, many of the participants did like the way the menu interface was created for the application, which was a disk that the user simply touched the trackpad to navigate it. With units of measurement in Unreal Engine in cm and since most 3D programs have sort of unit system, allowing the user to scale and have precise measurements for the 3D mesh would an important feature to add. For example, the teacups that were created in the VR application were larger than a real-world teacup and if they were exported the teacup would be huge in other 3D applications. With precise measurements and a mesh that is quite small, the ability to zoom and scale the mesh would be beneficial to the user. But the problem with this feature would affect the ability to grab the mesh components because the collisions for each component would be overlapping each other if the sizes of the collision boxes are too large and very close to one another. A solution to get around that would be to have relative size for the components that would depend on the size of mesh and/or distance from the user.

5.3 Summary

This chapter concludes the overall data collected and problems that arose during testing. This section also provided solutions to the features that the participants had trouble with and did not like. Making these changes based on the feedback given could improve the overall usability of the application and make it more user friendly.

REFERENCES

Achten, H., Vries, B. DE, Achten, H. H., nl, tue, Vries, B., Jessurun, J. A., & tue nl, bwk. (n.d.). DDDoolz A Virtual Reality Sketch Tool for Early Design.

Blocks - Create 3D models in VR. (n.d.). Retrieved April 18, 2018, from https://vr.google.com/blocks/

Carlos Jerónimo J., Antonio A., Méndez G., Ramírez J. (2009) Design of a Model of Human Interaction in Virtual Environments. In: Lopez Jaquero V., Montero Simarro F., Molina Masso J., Vanderdonckt J. (eds) Computer-Aided Design of User Interfaces VI. Springer, London

Ching-Shoei Chiang. (2017). The Tangent Medial Circles Inside the Region Defined by Hermite Curve Tangent to Unit Circle. *Advances in Technology Innovation, 2*(1), 08-12.

Dachille Ix, F., Qin, H., Kaufman, A., & El-Sana, J. (n.d.). Haptic Sculpting of Dynamic Surfaces. Retrieved from http://www3.cs.stonybrook.edu/~qin/research/fd-i3D1999.pdf

Echeverria, V., Falcones, G., Castells, J., Granda, R., & Chiluiza, K. (2017). Multimodal collaborative workgroup dataset and challenges. In *CEUR Workshop Proceedings*. https://doi.org/10.475/123

Huang, T. C., Chen, M. Y., & Lin, C. Y. (2017). Exploring the behavioral patterns transformation of learners in different 3D modeling teaching strategies. *Computers in Human Behavior*, 1–9. https://doi.org/10.1016/j.chb.2017.08.028

Jackson, B., & Keefe, D. F. (2016). Lift-Off: Using Reference Imagery and Freehand Sketching to Create 3D Models in VR. *IEEE Transactions on Visualization and Computer Graphics*. https://doi.org/10.1109/TVCG.2016.2518099

Liang, J. S. (2012). Modeling an immersive VR driving learning platform in a web-based collaborative design environment. *Computer Applications in Engineering Education*, *20*(3), 553–567. https://doi.org/10.1002/cae.20424

Lmaati, E. A., Oirrak, A. El, & Kaddioui, M. N. (2010). A 3D search engine based on 3D curve analysis. *SIViP*, *4*, 89–98. https://doi.org/10.1007/s11760-008-0091-2

Ma, D., Gausemeier, J., Fan, X., & Grafe, M. (2009). Virtual Reality & Augmented Reality in Industry. *The 2nd Sino-German Workshop*, (1), 215. https://doi.org/10.1007/s13398-014-0173-7.2

McGraw, T., Garcia, E., & Sumner, D. (2017). Interactive swept surface modeling in virtual reality with motion-tracked controllers. *Proceedings of the Symposium on Sketch-Based Interfaces and Modeling - SBIM '17*, (1), 1–9. https://doi.org/10.1145/3092907.3092908

Palamar, T. (2015). Mastering Autodesk Maya 2016: Autodesk Official Press, John Wiley & Sons, Incorporated, 2015.ProQuest Ebook Central, https://ebookcentral.proquest.com/lib/purdue/detail.action?docID=4180376.

Pinho, M. S., Dias, L. L., Antunes Moreira, C. G., Khodjaoghlanian, E. G., Becker, G. P., & Duarte, L. M. (2002). A User Interface Model for Navigation in Virtual Environments. *Cyberpsychology & Behavior*, *5*(5), 443–449. https://doi.org/10.1089/109493102761022869

Thomas Kohler, Johann Fueller, Kurt Matzler, & Daniel Stieger. (2011). Co-Creation in Virtual Worlds: The Design of the User Experience. *MIS Quarterly*, *35*(3), 773–788. https://doi.org/10.1300/J066v17n01

Tilt Brush. (n.d.). Retrieved December 2, 2018, from https://www.tiltbrush.com/

Virtual reality. (2002). In E. D. Hirsch, J. F. Kett, & J. S. Trefil, The new dictionary of cultural literacy, Houghton Mifflin (3rd ed.). Boston, MA: Houghton Mifflin. Retrieved from https://search.credoreference.com/content/entry/hmndcl/virtual_reality/0

APPENDICES

Example of a System Usability Survey

1. I think that I would like to use this system frequently.

1. Strongly Disagree	2.	3.	4.	5. Strongly Agree
○	○	○	○	○

2. I found the system unnecessarily complex.

1. Strongly Disagree	2.	3.	4.	5. Strongly Agree
○	○	○	○	○

3. I thought the system was easy to use.

1. Strongly Disagree	2.	3.	4.	5. Strongly Agree
○	○	○	○	○

4. I think that I would need the support of a technical person to be able to use this system.

1. Strongly Disagree	2.	3.	4.	5. Strongly Agree
○	○	○	○	○

5. I found the various functions in this system were well integrated.

1. Strongly Disagree	2.	3.	4.	5. Strongly Agree
○	○	○	○	○

6. I thought there was too much inconsistency in this system.

1. Strongly Disagree	2.	3.	4.	5. Strongly Agree
○	○	○	○	○

7. I would imagine that most people would learn to use this system very quickly.

1. Strongly Disagree	2.	3.	4.	5. Strongly Agree
○	○	○	○	○

8. I found the system very cumbersome to use.

1. Strongly Disagree	2.	3.	4.	5. Strongly Agree
○	○	○	○	○

9. I felt very confident using the system.

1. Strongly Disagree	2.	3.	4.	5. Strongly Agree
○	○	○	○	○

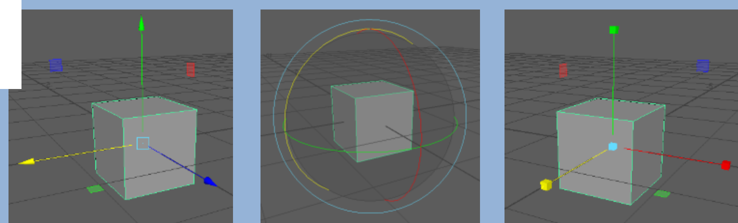
10. I needed to learn a lot of things before I could get going with this system.

1. Strongly Disagree	2.	3.	4.	5. Strongly Agree
○	○	○	○	○

Maya Instructions that were used by students who used Maya to complete the given task.

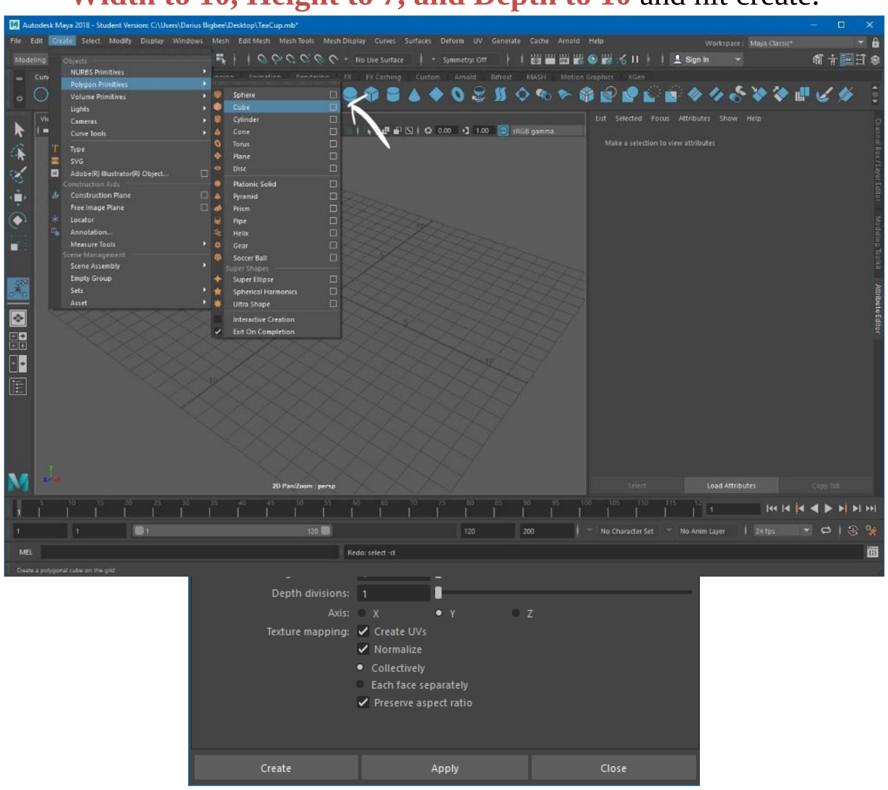
Modeling a Tea Cup in Maya

Tips:
W to move, **R** to rotate, **S** to scale

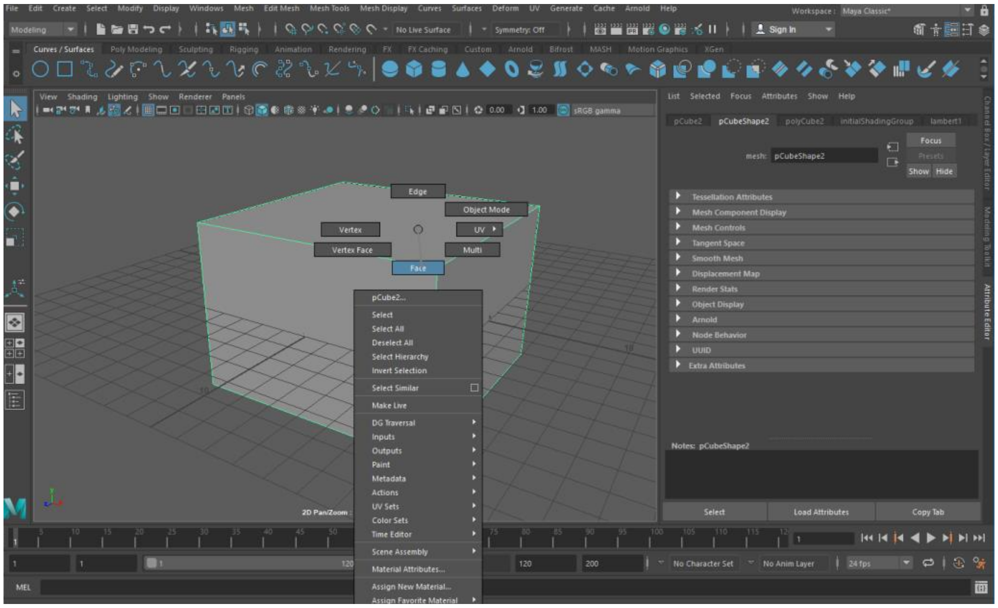


Hold **Shift** to select multiple components and **Alt** to deselect a component (Vertex, Edge, Face, etc.). To navigate the viewport **Alt+ (Left Mouse Button)** Rotate, **Alt + (Middle Mouse Button)** Pan, and **Scroll Wheel** Zoom. To select different modes, hold the **Right Mouse Button**.
☐ means to click the option box if seen in instructions when available.

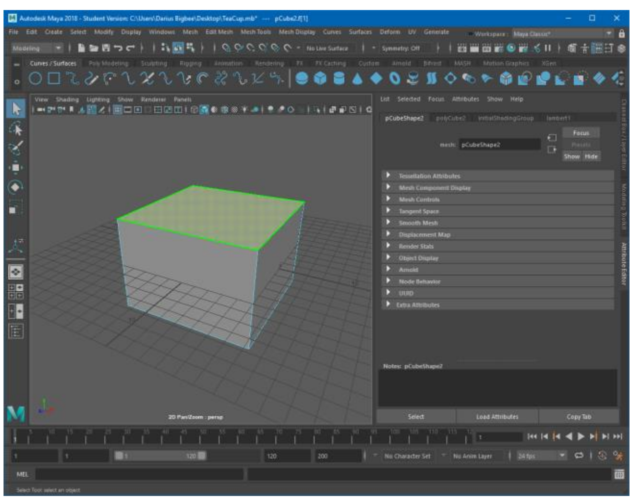
Create a Cube **Create->Polygon Primitives->Cube** ☐ and select the option box. Set the **Width to 10, Height to 7, and Depth to 10** and hit create.

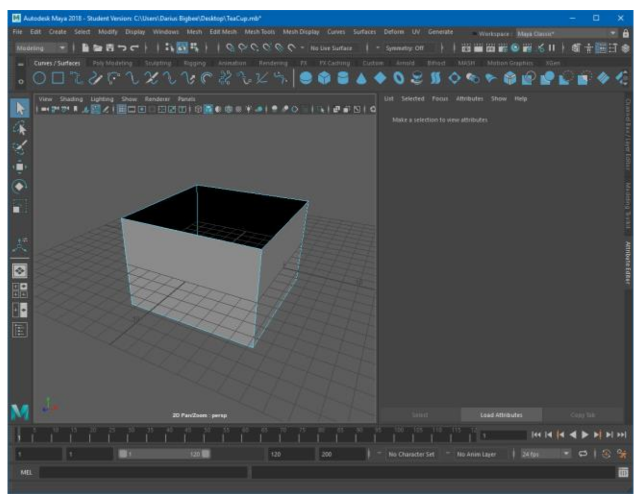


Holding the right mouse button over the cube and highlight **Face** to enter **Face mode**.

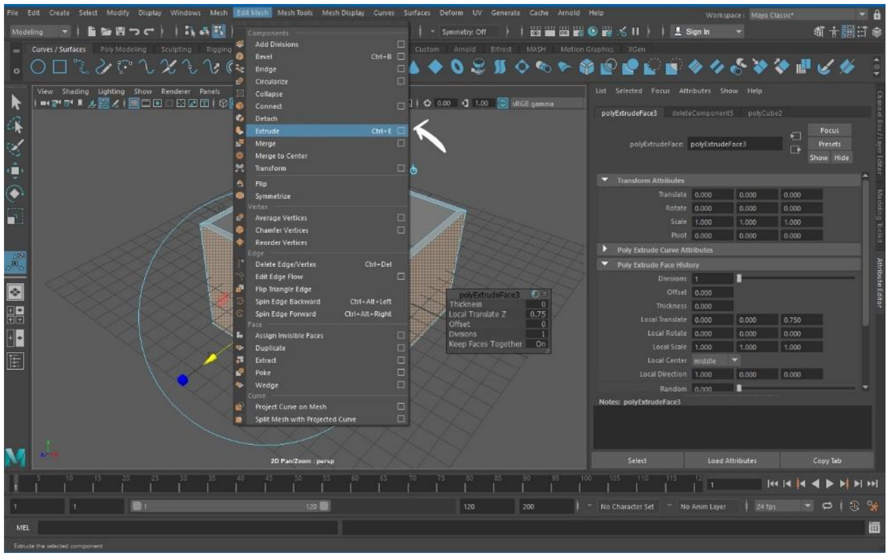


While in **Face mode** select the topmost face and then hit **Delete** to remove the face.





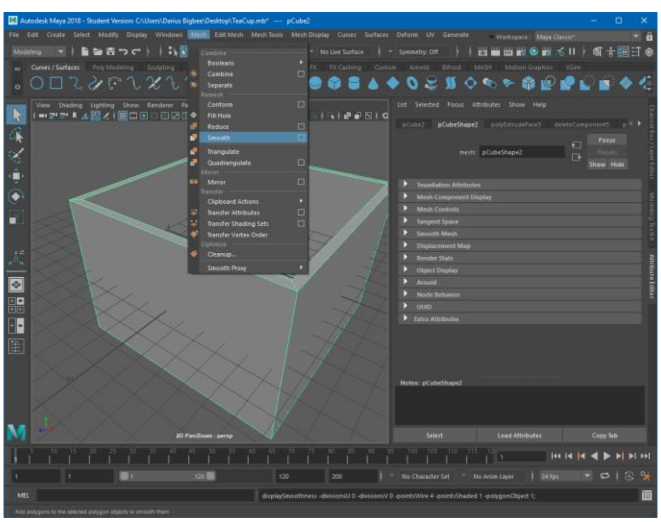
Next go back to **Object mode** again holding the right mouse button (same way we entered face mode). Next extrude the cube by going to **Edit Mesh->Extrude**. A small menu

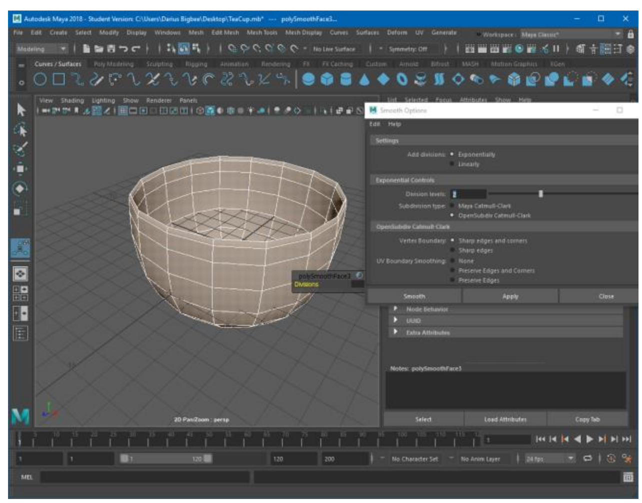


should appear, set the **Local Translate Z** to **.75** to give the cup thickness.

Press **w, r, s or click into an empty space** in the viewport to exit extrude mode.

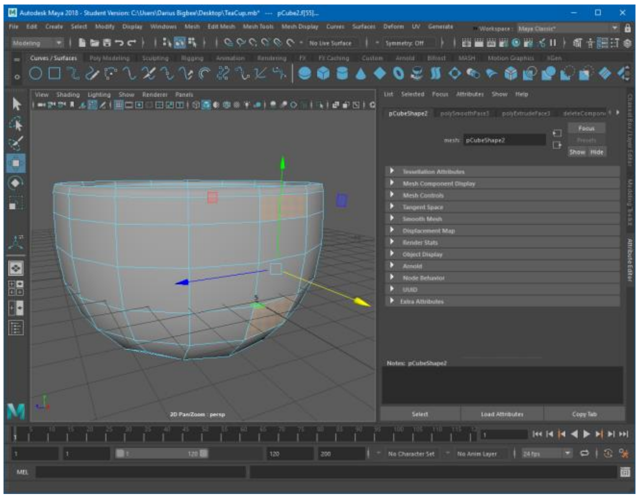
Go back to object mode and now we will subdivide the mesh to allow more detail to added namely the handle. To Subdivide go to **Mesh->Smooth□**. Set the **Division Levels to 2** and hit apply.

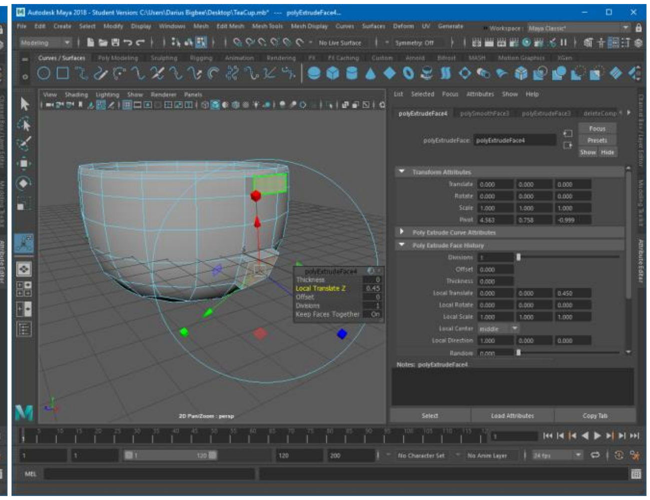




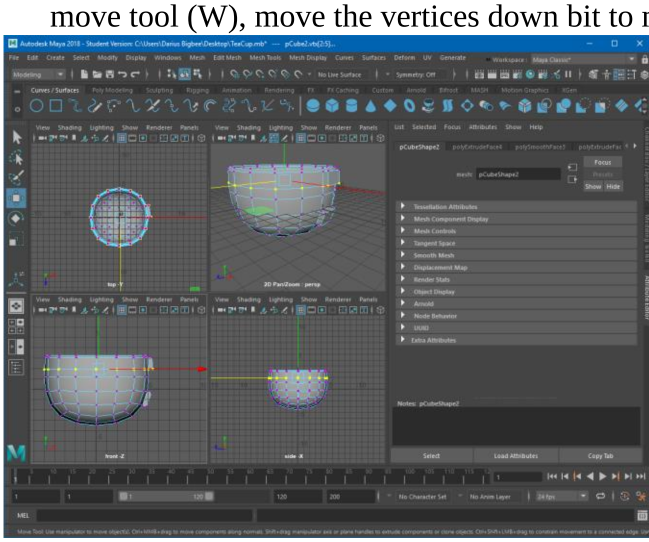
Again, hit **w, r, or s to enter object mode and exit smooth mode**, we will now create the handle for the cup

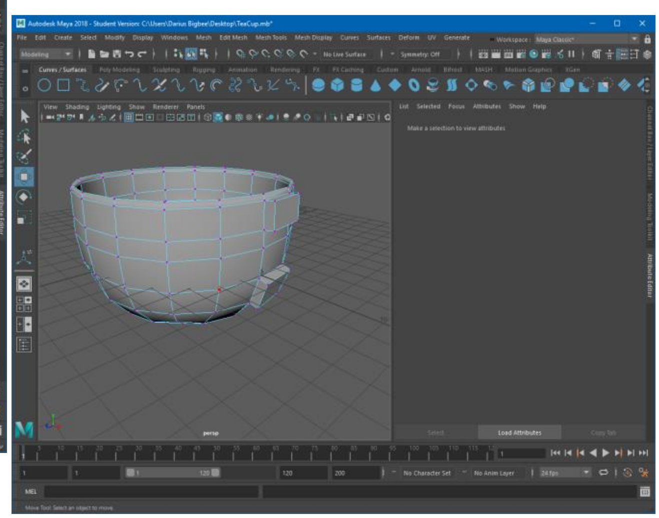
In **face mode**, select two faces on the side of the cup, one at the row before the top and one at the bottom on the side. With the faces selected **Extrude** them with a **Local Translate Z** of **.45**.



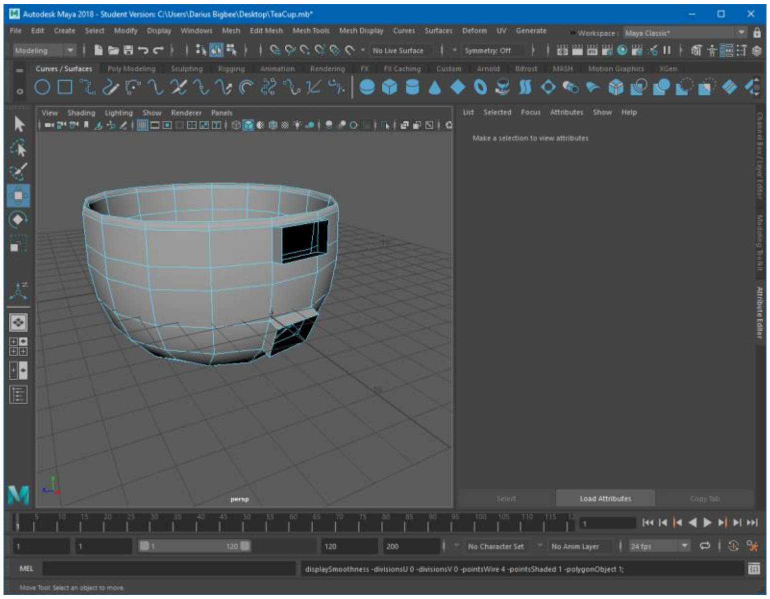


Now we want the extruded faces to be about the same size relative to each other. Enter **Vertex mode** holding the right mouse button. **Tap** the **Space Bar** to toggle open other viewports where we can see the front, top, perspective, and side of the mesh. Holding the left mouse button in an empty area, select the vertices that make up the bottom of the top extruded face. With the move tool (W), move the vertices down bit to make the extruded face about the same size.



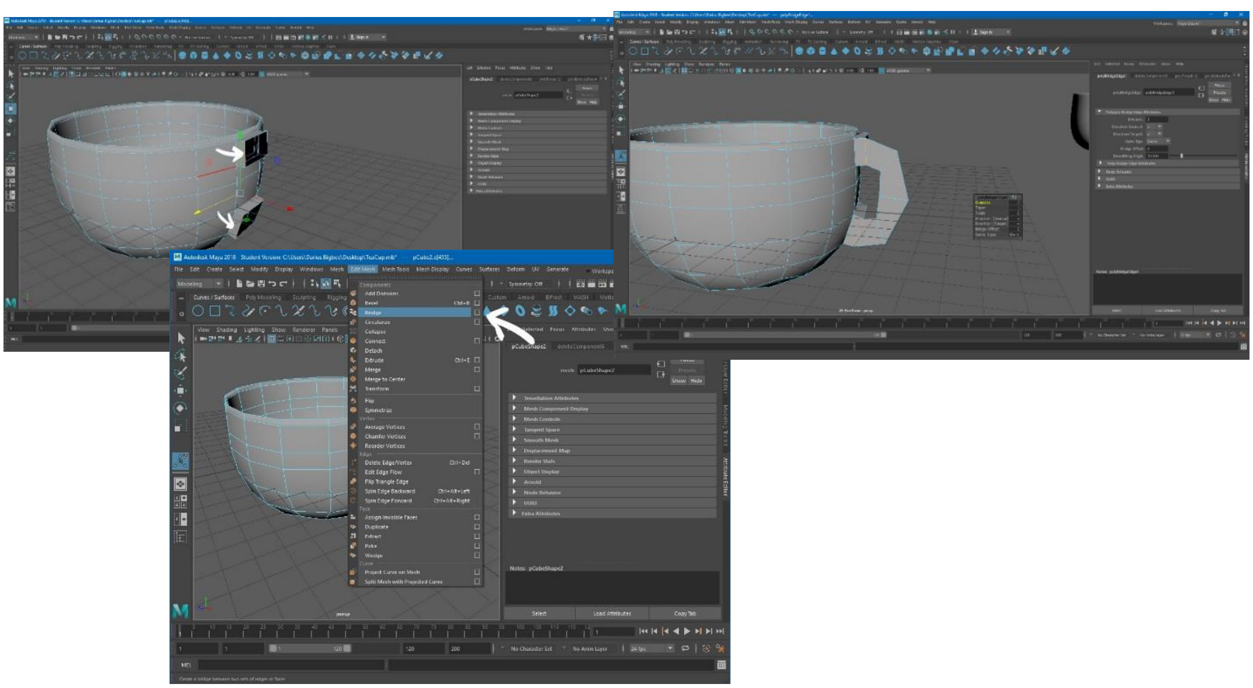


When done, **hover over the perspective window** and hit the **space bar** to maximize the window back to normal size. Go back to **Face mode** and reselect the extruded handle faces and

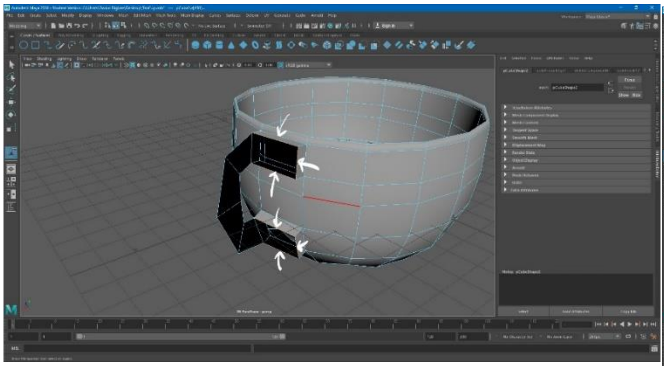
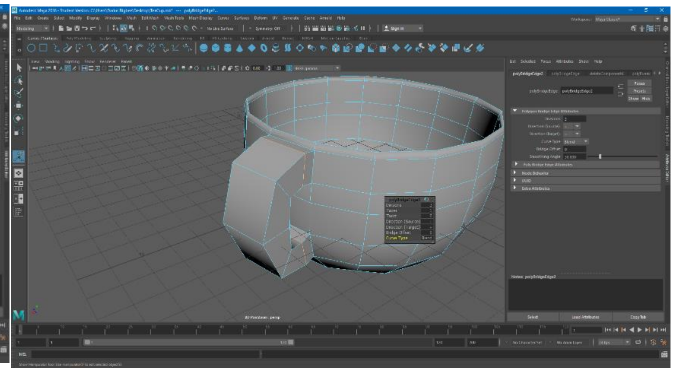


delete them.

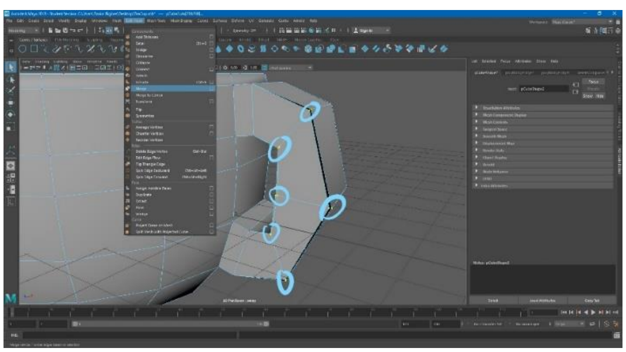
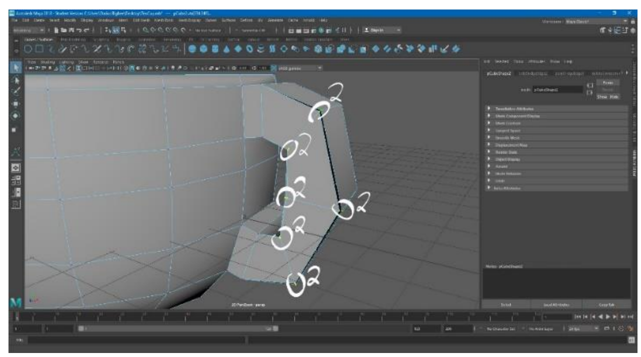
In edge mode, select an edge that makes up the deleted face. Go to **Edit Mesh->Bridge** to bridge the edges together. Set **Divisions** to **3**. Set the **Direction (Source)** and **Direction (Target)** to + from –. Set **Curve Type** to **Blend**. Exit bridge mode.



Now select the remaining edges of the holes and bridge those as well with the same settings mentioned before. Exit bridge mode when done.

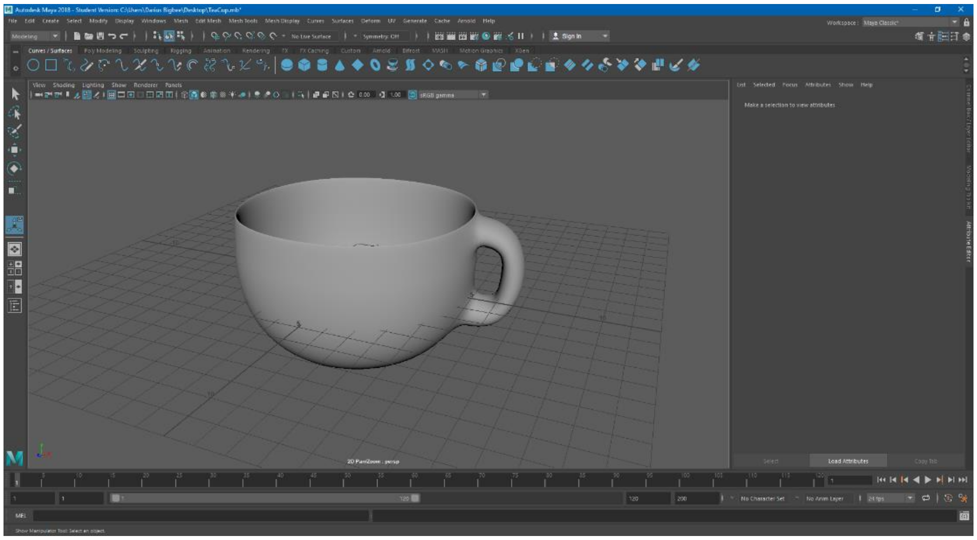



As you can probably see after bridging there are still holes in the handle of the cup. To fix we are going to merge those vertices. Select the vertices in vertex mode that are around the hole. Now go to **Edit Mesh->Merge** ☐ to merge the vertices in together. Set the **Threshold to .1**

and the vertices should combine, if not set the value higher. The threshold is the maximum distance a vertex should be to another before merging them into 1 vertex.

Back in object mode, smooth the mesh one more time with **Divisions** of **1** or **2.** Now our tea cup is finished.



Below is the IRB Exemption Letter for this study.

PURDUE
UNIVERSITY

HUMAN RESEARCH PROTECTION PROGRAM
INSTITUTIONAL REVIEW BOARDS

To:	GARCIA, ESTEBAN
From:	DICLEMENTI, JEANNIE D, Chair Social Science IRB
Date:	07/24/2018
Committee Action:(P100)	Determined Exempt, Category **(P100)**
IRB Action Date:	07 / 24 / 2018
IRB Protocol #:	1807020844
Study Title:	Interactive 3D Modeling in Virtual Reality

The Institutional Review Board (IRB) has reviewed the above-referenced study application and has determined that it meets the criteria for exemption under 45 CFR 46.101(b).

Before making changes to the study procedures, please submit an Amendment to ensure that the regulatory status of the study has not changed. Changes in key research personnel should also be submitted to the IRB through an amendment.

General

- To recruit from Purdue University classrooms, the instructor and all others associated with conduct of the course (e.g., teaching assistants) must not be present during announcement of the research opportunity or any recruitment activity. This may be accomplished by announcing, in advance, that class will either start later than usual or end earlier than usual so this activity may occur. It should be emphasized that attendance at the announcement and recruitment are voluntary and the student's attendance and enrollment decision will not be shared with those administering the course.
- If students earn extra credit towards their course grade through participation in a research project conducted by someone other than the course instructor(s), such as in the example above, the student's participation should only be shared with the course instructor(s) at the end of the semester. Additionally, instructors who allow extra credit to be earned through participation in research must also provide an opportunity for students to earn comparable

extra credit through a non-research activity requiring an amount of time and effort comparable to the research option.

- When conducting human subjects research at a non-Purdue college/university, investigators are urged to contact that institution's IRB to determine requirements for conducting research at that institution.
- When human subjects research will be conducted in schools or places of business, investigators must obtain written permission from an appropriate authority within the organization. If the written permission was not submitted with the study application at the time of IRB review (e.g., the school would not issue the letter without proof of IRB approval, etc.), the investigator must submit the written permission to the IRB prior to engaging in the research activities (e.g., recruitment, study procedures, etc.). Submit this documentation as an FYI through Coeus. This is an institutional requirement.

Categories 2 and 3

- Surveys and questionnaires should indicate
  - only participants 18 years of age and over are eligible to participate in the research; and
  - that participation is voluntary; and
  - that any questions may be skipped; and
  - include the investigator's name and contact information.
- Investigators should explain to participants the amount of time required to participate. Additionally, they should explain to participants how confidentiality will be maintained or if it will not be maintained.
- When conducting focus group research, investigators cannot guarantee that all participants in the focus group will maintain the confidentiality of other group participants. The investigator should make participants aware of this potential for breach of confidentiality.

Category 6

- Surveys and data collection instruments should note that participation is voluntary.
- Surveys and data collection instruments should note that participants may skip any questions.
- When taste testing foods which are highly allergenic (e.g., peanuts, milk, etc.) investigators should disclose the possibility of a reaction to potential subjects.

You are required to retain a copy of this letter for your records. We appreciate your commitment towards ensuring the ethical conduct of human subject's research and wish you luck with your study.

Ernest C. Young Hall, 10th Floor - 155 S. Grant St. - West Lafayette, IN 47907-2114 - (765) 494-5942 - Fax: (765) 494-9911

Figure A4. VR Application Demo

Figure A5. GitHub