

# Group 61: iCreate - Generative Design in Virtual Reality Preliminary Design Document

Rhea Mae Edwards  
Software Developer

Nabeel Shariff  
Software Developer

Hannah Solorzano  
Software Developer

CS 461  
Fall 2017

---

◆

## Abstract

The purpose of this document is to represent further research on different aspects related to the creation of the senior capstone project generative design in virtual reality. Topics that are discussed within this paper are types of headsets, a variety of programming languages used, a handful of software that can be used along with a virtual reality program, type of developing environments, user interfaces, functionality of different types of APIs, GPUs, frames, and user controls in relation to the project. Each of these topics described consists of three different options and how they are common and also differ from one another. Out of the three options, one of the options has been chosen to be the initial choice of being implemented for the generative design project. Overall, this document identifies components of the problem the project consists of, identifies technologies for building solutions, researches alternatives, analyzes trade-offs using criteria, and works toward persuading its reader of the best choice based on its criteria.

# TABLE OF CONTENTS

<b>1</b>	<b>Introduction</b>	<b>4</b>
<b>2</b>	<b>Headsets</b>	<b>6</b>
2.1	Overview . . . . .	6
2.2	Criteria . . . . .	6
2.3	Potential Choices . . . . .	6
2.3.1	HTC Vive . . . . .	6
2.3.2	Oculus Rift . . . . .	6
2.3.3	PlayStation VR . . . . .	6
2.4	Discussion . . . . .	7
2.5	Conclusion . . . . .	7
<b>3</b>	<b>Programming Languages</b>	<b>8</b>
3.1	Overview . . . . .	8
3.2	Criteria . . . . .	8
3.3	Potential Choices . . . . .	8
3.3.1	C# . . . . .	8
3.3.2	C/C++ . . . . .	8
3.3.3	Java . . . . .	8
3.4	Discussion . . . . .	9
3.5	Conclusion . . . . .	9
<b>4</b>	<b>Connective Software</b>	<b>10</b>
4.1	Overview . . . . .	10
4.2	Criteria . . . . .	10
4.3	Potential Choices . . . . .	10
4.3.1	Dynamo . . . . .	10
4.3.2	Grasshopper - Iris VR . . . . .	10
4.3.3	Leap Motion . . . . .	10
4.4	Discussion . . . . .	10
4.5	Conclusion . . . . .	11
<b>5</b>	<b>User Interface</b>	<b>12</b>
5.1	Overview . . . . .	12
5.2	Criteria . . . . .	12
5.3	Potential Choices . . . . .	12
5.3.1	Non-diegetic UI . . . . .	12
5.3.2	Spatial UI . . . . .	12
5.3.3	Diegetic UI . . . . .	12
5.4	Discussion . . . . .	12
5.5	Conclusion . . . . .	12
<b>6</b>	<b>Operating System</b>	<b>13</b>
6.1	Overview . . . . .	13
6.2	Criteria . . . . .	13
6.3	Potential Choices . . . . .	13
6.3.1	Microsoft Windows . . . . .	13
6.3.2	Apple Macintosh . . . . .	13
6.3.3	Linux . . . . .	13
6.4	Discussion . . . . .	13
6.5	Conclusion . . . . .	13
<b>7</b>	<b>Distribution Method</b>	<b>14</b>
7.1	Overview . . . . .	14
7.2	Criteria . . . . .	14
7.3	Potential Choices . . . . .	14
7.3.1	Steam Store . . . . .	14
7.3.2	Oculus Home . . . . .	14

	3
7.3.3 Executable File . . . . .	14
7.4 Discussion . . . . .	14
7.5 Conclusion . . . . .	14
<b>8 Graphics Card</b>	<b>15</b>
8.1 Overview . . . . .	15
8.2 Criteria . . . . .	15
8.3 Potential Choices . . . . .	15
8.3.1 GeForce GTX 970 . . . . .	15
8.3.2 NVIDIA TITAN Xp . . . . .	15
8.3.3 AKiTiO Node Thunderbolt 3 . . . . .	15
8.4 Discussion . . . . .	15
8.5 Conclusion . . . . .	15
<b>9 Development Environment</b>	<b>16</b>
9.1 Overview . . . . .	16
9.2 Criteria . . . . .	16
9.3 Potential Choices . . . . .	16
9.3.1 Unity . . . . .	16
9.3.2 Unreal Engine . . . . .	16
9.3.3 OpenGL . . . . .	16
9.4 Discussion . . . . .	16
9.5 Conclusion . . . . .	16
<b>10 User Controls</b>	<b>17</b>
10.1 Overview . . . . .	17
10.2 Criteria . . . . .	17
10.3 Potential Choices . . . . .	17
10.3.1 Oculus Rift Touch Controllers . . . . .	17
10.3.2 Hands . . . . .	17
10.3.3 Sony controllers . . . . .	17
10.4 Discussion . . . . .	17
10.5 Conclusion . . . . .	17
<b>11 Conclusion</b>	<b>18</b>
<b>References</b>	<b>19</b>

# 1 INTRODUCTION

There are many tools that our team will have to use in order to fulfill all of the requirements for our senior capstone generative design virtual reality program project. A few of the tools that we will have to implement are a headset, a programming language to write in, and the possibility of including some kind of other software that can be used with our finalize virtual reality program in order to intensify such a virtual user experience. Other tools for our project includes a developing environment, the structure of the program's user interface, functionality of an application programming interfaces (APIs), graphics processing units (GPUs), the development of the program's user controls, and frames.

The generative design virtual reality program our team will be creating involves the construction of simple 3D structures initiated by the user and further development by the software program. Such a program involves the user placing simple 3D object in some environment, and then setting constraints for the program to then build some structure by using the simple 3D object. These tools of a headset, programming language, and connective software, will have to satisfy such a overall functionality and purpose.

The options that we have chosen to research for headsets are the following:

- 1) HTC Vive
- 2) Oculus Rift
- 3) PlayStation VR

The options that we have chosen to research for programming languages are the following:

- 1) C#
- 2) C/C++
- 3) Java

The options that we have chosen to research for connective software are the following:

- 1) Dynamo
- 2) Grasshopper - Iris VR
- 3) Leap Motion

The options that we have chosen to research for user interfaces are the following:

- 1) Non-diegetic UI
- 2) Spatial UI
- 3) Diegetic UI

The options that we have chosen to research for the Operating Systems for development are the following:

- 1) Microsoft Windows
- 2) Apple Macintosh
- 3) Linux

The options that we have chosen to research for distribution methods are the following:

- 1) Steam Store
- 2) Oculus Home
- 3) Executable File

The options that we have chosen to research for GPUs are the following:

- 1) GeForce GTX 970
- 2) NVIDIA TITAN Xp
- 3) Intel HD Graphics 520

The options that we have chosen to research for developing environments are the following:

- 1) Unity
- 2) Unreal Engine
- 3) OpenGL

The options that we have chosen to research for user controls are the following:

- 1) Oculus Rift Touch Controllers
- 2) Hands
- 3) Sony controllers

This document will further discuss the options stated above, and which ones out of each of the topics we have chosen to initial use in our project implementation. Each section will describe each of the options individually, and then compare each of them in regards to our project requirements and preferred implementation choices.

## 2 HEADSETS

### 2.1 Overview

In order for our virtual reality program to be useful and actually be implemented for its full purpose, there will need to be a headset for us to download our program onto. A headset is the main piece of hardware that our team will need in order for us to properly test and use our virtual program with. Choosing a headset that will fulfill all of our needs and requirement is important, and highly sufficient enough for us for our project.

### 2.2 Criteria

Headsets are wore by individual users, allowing them to view computation altered reality with or through a screen, such as virtual and augmented reality. The price of a headset can greatly vary from just tens of dollars to a couple thousands of dollar depending on the complexity and usability of the device. Every headset has its own set of a variety of uses, where some headsets can be more relevant for some purposes and software program than others.

### 2.3 Potential Choices

#### 2.3.1 *HTC Vive*

The HTC Vive provides a decent space for its users to move within a room or in other words "play area". This headset allows a maximum of 15 feet for its user to move from its base-stations while in use. Such a play area does not have to be a perfect square to work effectively either. These boundaries built for the HTC Vive have been made to be fairly flexible. The HTC Vive also can detect objects and can possible even replace any obstacles within its play area, preventing any overlap or object confusion within its visuals. This specification also helps with the safety and real-life physical movements. [1]

A downside with the HTC Vive though, involves this idea of its use of effectively tracking its user. According to the Gadget review of 2015, "Here's EXACTLY How The HTC Vive Works," there is a certain way the headset should be worn, begin "mounted above head height, [...] faced down at a 30-45 degree angle." [1] Such accuracy can be very particular in its use for a user to wear.

#### 2.3.2 *Oculus Rift*

One thing about the Oculus Rift, this headset has Windows 10 cross-device compatibility, where streaming through the device is a possibility when connected to a Xbox One. The Oculus Rift has similar head motion controls with its user when moving through a play area, being about a 3-foot by 3-foot square in any given space. This headset can also connect to a computer for further use allowing future installs and upgrades for its software. [2]

Also, the game engines Unity and Unreal Engine are supported by the Oculus Rift.

#### 2.3.3 *PlayStation VR*

The PlayStation VR is a fairly new headset which has not been out on the market for barely even a year. The PlayStation VR is highly compatible with game consoles, begin a true console-based user experience. This headset also provides "HDR (high dynamic range) - compatible TVs" meaning its support with HDR - compatible games. Also for a consumer's perspective on the headset, the price one pays for the device is the best value provide for how much it costs, which is high consumer's benefit. [3]

## 2.4 Discussion

	Virtual Reality Capability	Abundant Available Software Resources for New Users	"Large" Play Area
HTC Vive	✓	✓	✓
Oculus Rift	✓		✓
PlayStation VR	✓		

Comparison of Headset Technology Options

## 2.5 Conclusion

	Virtual Reality Capability	Abundant Available Software Resources for New Users	"Large" Play Area
HTC Vive	✓	✓	✓

HTC Vive Headset Choice

We chose the HTC Vive headset setup because its play area specifics and available resources we have through the internet and personal user experiences in using the device. With our virtual reality program in relation with generative design, using the HTC Vive provides a great amount of physical space for a user to move and design their elements within the program, and hardware specifics that allow a user to get a maximum virtual reality experience.

### 3 PROGRAMMING LANGUAGES

#### 3.1 Overview

Before writing any code, we need to figure out what programming language we will want to write in. Basis of the basics.

#### 3.2 Criteria

Individually, programming languages were made with their own purpose with it comes to coding. There are programming languages that are more appropriate to write in when it comes to coding for a virtual reality program than others. Each language can also provide its own set of unique functionality. Also, when it comes to what a developer is programming, there are more sources out there that can help with such programming problems depending on the language one uses.

#### 3.3 Potential Choices

##### 3.3.1 C#

A note from C# developers is their love with the language of C# "for begin pleasant to use and well-designed". C# was originally designed by Microsoft for developing apps. The language is also highly recommended to be used for creating games through the Unity game engine being easy to start with, which is relevant to our team's over project, in relation to writing a virtual reality program. C# is a high level language, where there is a focus more on programming than the little details of the language itself, which can be slightly annoy for using other programming languages out there. [4]

On the scalability side, C# is easy to maintain and fast for being a statically typed language. Also, there are many resources and examples written in C# for developers to reference and use. [4]

##### 3.3.2 C/C++

C++ is a language that allows its programmer to have a lot of power and control over its computers resources through the act of coding. With sufficient knowledge in using the language, its programs can be created cheaply with high speeds and ability to run compare to other programming languages. C++ also allows its programmers "to develop game engines, games, and desktop apps. Many AAA title video games are built with C++." [5]

A downside of coding in C++, is its beginner friendliness. C++ is a lower level language that can be very complex, especially in regards to brand-new programmers coding a virtual reality program such as every member in our team. Such a complex with the language, also makes it a little difficult to maintain in general.

##### 3.3.3 Java

According to Codementor, "this general-purpose language [of Java] was designed to be easier to use than C++." Companies have used Java to develop desktop apps and website backend systems. Being more related with virtual reality programs, Java is highly used in developing Android apps. Java is beginner friendly and can be fairly easy to use to optimize performance within a written program, but can be a frustrating the use from the start, due to its great lengths in code when writing. [6]



### 3.4 Discussion

	Able to Use for Virtual Reality Programs	Simple to Learn for Beginners	Many Available Resources
C#	✓	✓	✓
C/C++	✓		✓
Java	✓		✓

Comparison of Programming Language Technology Options

### 3.5 Conclusion

	Able to Use for Virtual Reality Programs	Simple to Learn for Beginners	Many Available Resources
C#	✓	✓	✓

C# Programming Language Choice

We chose to use C# as our main programming language because mainly with our initial chosen game engine, Unity, for our group to code in, C# is the more widely use language to code in. Many of the tutorials and examples when coding in Unity are in C# , along with C# begin the most often recommended programming language when creating games through the Unity game engine. [4]

## 4 CONNECTIVE SOFTWARE

### 4.1 Overview

Suggested by our client, was a handful of separate software and development tools that we as a team, can think of using with our program once we get our basic implementation working for our main project. The idea behind these additions, is to further our basic idea of a program and provide a more diverse user experience with our generative design virtual program. There are hopes also that adding a single or multiple additional software and tools will provide a more realistic experience for the program's users overall.

### 4.2 Criteria

There are only certain types of software that will be able to connect with a virtual headset, and especially depending on the one our team decides on using. Along with the additional software's ability to integrate, the software's purpose is to enhance and benefit the main program in some way. Such enhancement can be done by the addition of useful features or accessibility and functionality of the program overall.

### 4.3 Potential Choices

#### 4.3.1 *Dynamo*

Dynamo is a software that is useful to implement in regards to "easing the modeling, visualization, and analysis tools" for most connected devices. In relation to our generative designing virtual reality project, Dynamo can help users "generate sophisticated designs from simple data, logic, and analysis" due to its structural work-flow in the software along with is such implementation. [7]

#### 4.3.2 *Grasshopper - Iris VR*

Grasshopper - Iris VR helps provide exported "surfaces, meshes, and breps" that can be found very useful in our generative design virtual reality program. This software can also help apply material and color to a structure or object in a given program. In regards in providing a greater designing ability with generative design virtual reality program for our users, this addition can be helpful in providing such capabilities. [8]

#### 4.3.3 *Leap Motion*

Through the cameras of a headset and infrared light, Leap Motion can help integrate a user's real hand motions within a virtual reality program. Leap Motions is rated to be fairly simple to use and implement by a developer, which is a great benefit for beginner programmers wanting to just simply create a highly realistic program through virtual reality in a short amount of given time. [9]

### 4.4 Discussion

	Capable with the HTC Vive	Movement Relevant	Visually Relevant
Dynamo	✓		✓
Grasshopper - Iris VR	✓		✓
Leap Motion	✓	✓	

Comparison of Connective Software Technology Options

## 4.5 Conclusion

	Capable with the HTC Vive	Movement Relevant	Visually Relevant
Leap Motion	✓	✓	

### Leap Motion Connective Software Choice

We chose Leap Motion as our initial main software to connect our main virtual program to because it allows its users to use their bare hands when maneuvering through a virtual reality program. Allowing this type of availability for our user with generative designing will be a very convenient addition, and allow our users to experience a more realistic building of a virtual physical structural design with their own hands in virtual reality.

## 5 USER INTERFACE

### 5.1 Overview

A user interface (UI) is the way the user interacts with the software system. Virtual Reality (VR) has allowed designers and developers to break the limits of UI and literally think outside the box. Since the team will be using the Unity game engine [10], the interface will depend on the UI options available through Unity.

### 5.2 Criteria

The criteria for choosing the appropriate UI is based on maximizing the quality and comfort of the user's experience. The user should be able to have easy access to essential tools through an intuitive UI. The UI should also work seamlessly for both controller and gesture output, and allow the user to maximize their efficiency when designing 3D models.

### 5.3 Potential Choices

#### 5.3.1 *Non-diegetic UI*

A non-diegetic UI is an UI that is overlaid on top of a screen, usually in the form of a heads-up display (HUD) [11]. It doesn't exist within an environment, but makes sense for the user in the context of viewing the application. Examples of non-diegetic UI include health bars and scores.

#### 5.3.2 *Spatial UI*

Spatial UI is meant to place the UI within the environment itself, being attached to the virtual camera of the headset, allowing the user's eyes to focus on the UI [11]. An example of spatial UI is a virtual notification appearing in front of the user in VR.

#### 5.3.3 *Diegetic UI*

Diegetic UI provides a way for elements within the environment to display information to the user. This type of UI can be attached to elements within the virtual environment. A virtual holographic display on the user's controller is an example of diegetic UI.

### 5.4 Discussion

Non-diegetic UI is a very common UI and is probably the best option for non-VR applications. However, iCreate is a VR application, so the non-diegetic UI will be too close for the user to focus on with their eyes, causing great discomfort to the user. In contrast, spatial and diegetic UI offer an alternative that allows the user to comfortably interact with their VR environment without feeling discomfort, i.e. nausea.

Spatial UI will allow the program to relay information directly in front of the user. This type of UI is useful for showing the user information, but isn't as useful for receiving input from the user. In contrast, diegetic UI allows not only a way to display information to the user, but also receive input from the user. Compared to spatial UI, diegetic UI is attached to the environment. This means the user can directly interact with almost any element within the virtual environment if it has an UI attached.

### 5.5 Conclusion

The team has chosen diegetic UI as the main UI for the iCreate VR application. Diegetic UI can be used to attach small menus with sliders and buttons on the user's controllers, allowing the user to comfortably have their tools within arm's reach. Additionally, to relay notifications directly to the user, the team may use spatial UI, but the main UI will still be diegetic.

## 6 OPERATING SYSTEM

### 6.1 Overview

Apart from the VR headset's propriety operating system (OS), the VR application also needs to run on the OS that is running on the computer itself. The OS manages the resources needed by the headset and VR software to interact with each other.

### 6.2 Criteria

The appropriate OS chosen for iCreate's development should give the team access to the chosen software tools and should be highly compatible with those tools. Additionally, the OS shouldn't hinder the performance of the VR application, and should also be compatible with the headset and its proprietary software.

### 6.3 Potential Choices

#### 6.3.1 *Microsoft Windows*

Windows [12] is the most commonly used OS, and offers great compatibility with Unity, the HTC Vive headset, and VR applications. Additionally, there is a plethora of documentation related to the chosen software tools, facilitating the team's development.

#### 6.3.2 *Apple Macintosh*

Macintosh [13] is also widely used around the world, but is not as open as some of the other operating systems. It doesn't have great compatibility with VR applications, and only recently supported the HTC Vive with the latest Macintosh OS update [14]. It does however support Unity very well, and is also compatible with several design applications that can help the team with 3D development. Moreover, documentation for Unity and other software is present for Macintosh as well.

#### 6.3.3 *Linux*

Linux [15] offers the team the most freedom when it comes to programming. Moreover, Linux is also widely used and is open source. However, although Linux supports the HTC Vive, it does not offer stability and compatibility with Unity. Additionally, Unity is only available experimentally for Linux [16].

### 6.4 Discussion

Macintosh offers a great design experience for the development team, and Linux offers a more direct approach for development. However, both Macintosh OS and Linux are lacking when it comes to compatibility and stability of the software tools chosen by the team. In contrast, Windows is highly compatible with Unity and HTC Vive, and both are stable on Windows. Additionally, most of the team's experience with developing in Windows is much greater compared to the experience with developing on the other operating systems. Moreover, there is plenty of documentation for Windows about Unity, Leap Motion, and the HTC Vive headset, making Windows the most effective and efficient option.

### 6.5 Conclusion

The team has decided to develop the iCreate VR application on the Microsoft Windows operating system because it offers the most compatibility and support for the software tools chosen by the team, and also allows for the most efficient and effective development with a great deal of documentation available for Windows. Additionally, Windows is also the most commonly used OS for VR application use, and thus will give the team access to the widest share of users [17].

## 7 DISTRIBUTION METHOD

### 7.1 Overview

Once the VR application is developed, it has to find its way to the user via some channel of distribution. A distribution method is the way the user can access and install the VR application on their system.

### 7.2 Criteria

The criteria for choosing the appropriate distribution method has to take into account the simplest and most convenient way for the user to access and install the iCreate VR application. It should also be compatible for the chosen operating system, hardware, and software chosen for development.

### 7.3 Potential Choices

#### 7.3.1 *Steam Store*

The Steam Store [18] is most compatible with the HTC Vive. It also has a large number of users. Additionally, it supports the use of Oculus Rift for VR applications. Furthermore, Steam makes it slightly easier for users to access the VR application since the users on the Steam Store know exactly what their looking for. However, it also requires the use of a proprietary development kit, and involves a \$100 fee to join the program [19].

#### 7.3.2 *Oculus Home*

Oculus Home is similar to the Steam Store, but is exclusive to Oculus products, and does not support the HTC Vive [19]. This also limits the user base and development options. Although there is no fee, a developer account is required to publish on Oculus Home [10].

#### 7.3.3 *Executable File*

Distributing the VR application as a standalone executable program provides the most freedom and direct access to the user. However, since this distribution method does not involve a 3rd party store, it is the least secure method of distribution. Additionally, the team will have to find a way to make the iCreate application available for download for users.

### 7.4 Discussion

Since the team is using the HTC Vive as the main hardware to develop for, Oculus Home does not provide any benefit as it is only available for Oculus products. In contrast, both the Steam Store and executable file distribution methods make iCreate accessible for HTC Vive users, as well as for other headsets including the Oculus Rift. However, compared to the Steam Store option, the executable file distribution is free and gives the development team the most freedom and control for distribution. Moreover, the executable file can be made available for download directly from the iCreate website, so the team will not be limited to using a proprietary development kit and will not be required to pay any fees either.

### 7.5 Conclusion

The team has decided to distribute iCreate via an executable file, available for download directly from the iCreate website. For now, the VR app will be distributed for free, so there is no need to go through a 3rd party channel at the moment. In the future, should the team decide to distribute iCreate more aggressively and for profit, the Steam Store will serve as a good backup option.

## 8 GRAPHICS CARD

### 8.1 Overview

The graphics card, or graphics processing unit (GPU), in the computer that is running the program is important as the GPU is what allows the computer to render and clearly display graphics. Without the proper GPU, the user will not be able to use iCreate.

### 8.2 Criteria

The GPU is required to effortlessly render the constantly updating program. As such, the minimum requirements are at least a GeForce GTX 970 [20] or an AMD Radeon R9 290 [21]. These GPU's are not very powerful in terms of their rendering power, but they are still capable of running a VR program. In addition to a strong GPU, it should also be noted that the computer needs at least a i5 core central processing unit (CPU), 8GB of RAM, and a 1.3 HDMI port.

### 8.3 Potential Choices

#### 8.3.1 GeForce GTX 970

This graphics card is the least powerful GPU that is still capable to run VR programs. It is affordable and easy to use.

#### 8.3.2 NVIDIA TITAN Xp

NVIDIA claims that the NVIDIA TITAN Xp [22] is the strongest GPU currently on the market. It has more rendering power and produces highly detailed displays. It is not as affordable as the GeForce GTX 970, as it retails for \$1200.

#### 8.3.3 AKiTio Node Thunderbolt 3

The AKiTio Node Thunderbolt 3 [23] is a eGFX box that contains a high powered GPU. It also contains a 400W SFX power supply that provides more power to the GPU. This choice is more affordable than the other choices at \$250.

### 8.4 Discussion

The team considered the budget for the project when considering which GPU to use. As the AKiTio Node Thunderbolt 3 is both affordable and accessible, the team went with this option. The NVIDIA TITAN Xp is clearly not an option as the budget is not large enough, and the GeForce GTX 970 is affordable, but its rendering power is not strong enough.

### 8.5 Conclusion

In the end, the team decided to use the AKiTio Node Thunderbolt 3 as we already possess this GPU and we do not have the budget to acquire new graphics cards.

## 9 DEVELOPMENT ENVIRONMENT

### 9.1 Overview

The libraries that the chosen game engine should have must fully be able to complete a given task. This way, the team does not have to spend time reinventing the wheel and creating our own methods while designing the program's functionality.

### 9.2 Criteria

The game engine that will be used in the development of this project is required to have an easy learning curve, as well as the capability to render detailed graphics using a series of libraries.

### 9.3 Potential Choices

#### 9.3.1 *Unity*

Unity [24], a game engine known for its easy learning curve offers several tools for programming simulations and video games for platforms, computers, and mobile phones. It utilizes the C# programming language whose classes and object oriented items allow for the easier creation of 3D items and interface. Unity also offers an abundance of libraries and tutorials for virtual reality as there is a large user base who share their 3D objects and libraries so that others may use those items in their own project.

#### 9.3.2 *Unreal Engine*

Unreal Engine [25] is known for its highly detailed graphics and rendering capabilities. It is written in C++, another object oriented language, which allows for game objects to have more detailed specifications. This engine is much more difficult to learn, and there is a smaller casual user base as Unreal has only gone free to use for a short time. As such, there are less tutorials and user written libraries available for use.

#### 9.3.3 *OpenGL*

Unlike the other two choices, OpenGL [26] is not a game engine. Instead, it is a application programming interface (API) that specializes in communicating directly with the graphics processing unit (GPU). This is advantageous because it allows for more customization and control over how an object renders. This API is written in C, an object-oriented language, which uses classes to give objects more customization options such as position, color by pixel, shininess/reflectiveness, and smoothness. There are few libraries that add additional features, and customizing or creating our own library is a daunting task.

### 9.4 Discussion

The team considered the benefits of each game engine and thought about how important the number of libraries present will be once we begin to design the program. It was decided that it is important for the engine to have a large user base as this means that there are more tutorials, forums, books, or people to consult if we run into a problem or are struggling to figure out how to design a certain feature.

### 9.5 Conclusion

In the end, it was decided that Unity would be the best tool for us to design the iCreate program on. With its large user base, numerous libraries available, and easy learning curve, this tool would be the easiest engine to use for development. Compared to the smaller user bases of Unreal, or the more complicated library structure of OpenGL.



## 10 USER CONTROLS

### 10.1 Overview

This technology is how the user controls how they manipulate the 3D objects in the virtual space. Since this requires an extreme amount of precision, the technology that we end up going with needs to have pinpoint accuracy.

### 10.2 Criteria

Due to the nature of this program, the user needs to be able to grab and drag the points of an object, as well as sketch precise trajectory paths.

### 10.3 Potential Choices

#### 10.3.1 *Oculus Rift Touch Controllers*

The Oculus Rift Touch controllers [27] feature a small controller that fits in the palm of your hand, along with a piece that wraps around the entirety of your hand. It is designed to add touch sensitivity to increase the VR experience.

#### 10.3.2 *Hands*

This form of a controller is very basic as there are no physical controllers required. Certain types of VR programs use the headset to identify where the fingers of the hand are positioned, and allows the user to pinch, drag, and drop items with hand gestures.

#### 10.3.3 *Sony controllers*

The Sony controllers [28] are small analog type controllers that feature an analog stick and arrow keys. This controller is widely available, but is not as precise in detecting movement.

### 10.4 Discussion

The team wanted a controller that would maximize the level of preciseness that the user experiences. This is important because the preciseness of movements is what allows the user to create intricate structures. Affordability of the controller was also considered.

### 10.5 Conclusion

In the end, the team decided that we would implement the iCreate program to incorporate hand gestures. We found this to be the most precise of all the types of controllers, and there is no additional costs to including them.

## 11 CONCLUSION

In conclusion, the iCreate development team will be using components that are most useful and beneficial to both the users and the developers. The team has chosen to develop for the HTC Vive using the Unity 3D game engine, utilizing programs written in C#. Additionally, the team will be using Leap Motion to implement the user's hands as one of the primary methods of input, and the user interface will mostly consist of diegetic UI. Moreover, the team will be using an Nvidia GeForce GTX 970 (or better) video card as the primary dedicated graphics card. Finally, the iCreate application will be developed for the Windows operating system, and to be distributed as an executable file via a download page on the iCreate website.

## REFERENCES

- [1] D. Peppiatt. (2015). *Here's EXACTLY How The HTC Vive Works* . [Online]. Available: <https://www.gadgetdaily.xyz/raspberry-pi-3-is-on-sale-now/>
- [2] M. Andronico, S. Smith. (2016). *What is the Oculus Rift?* . [Online]. Available: <https://www.tomsguide.com/us/what-is-oculus-rift,news-18026.html>
- [3] W. Fulton. (2017). *PlayStation VR (2017) review* . [Online]. Available: <https://www.digitaltrends.com/vr-headset-reviews/playstation-vr-2017-review/>
- [4] Codementor. (2016). *Why Learn C# ?* . [Online]. Available: <http://www.bestprogramminglanguagefor.me/why-learn-c-sharp>
- [5] Codementor. (2016). *Why Learn C++?* . [Online]. Available: <http://www.bestprogramminglanguagefor.me/why-learn-c-plus-plus>
- [6] Codementor. (2016). *Why Learn Java?* . [Online]. Available: <http://www.bestprogramminglanguagefor.me/why-learn-java>
- [7] J. Dellel. (2017). *Dynamo: Autodesk's answer to Computational Design* . [Online]. Available: <https://enterprisehub.autodesk.com/articles/dynamo-autodesk-s-answer-to-the-computational-design>
- [8] IrisVR. *Grasshopper* . [Online]. Available: <https://help.irisvr.com/hc/en-us/articles/220686068-Grasshopper>
- [9] A. Colgan. (2014). *How Does the Leap Motion Controller Work?* . [Online]. Available: <http://blog.leapmotion.com/hardware-to-software-how-does-the-leap-motion-controller-work/>
- [10] Unity Technologies. (2017). *Unity-Game Engine* . [Online]. Available: <https://unity3d.com/>
- [11] Unity Technologies. (2017). *Unity-User Interfaces for VR* . [Online]. Available: <https://unity3d.com/learn/tutorials/topics/virtual-reality/user-interfaces-vr>
- [12] Microsoft. (2017). *Windows — Official Site* . [Online]. Available: <https://www.microsoft.com/en-us/windows/>
- [13] Apple. (2017). *macOS High Sierra - Apple* . [Online]. Available: <https://www.apple.com/macos/high-sierra/>
- [14] L. Painter. (2017). *How to use VR on a Mac* . [Online]. Available: <https://www.macworld.co.uk/how-to/mac/how-use-vr-on-mac-3640213/>
- [15] Linux. (2017). *Linux.org* . [Online]. Available: <https://www.linux.org/>
- [16] N. Bard. (2015). *The State of Unity on Linux* . [Online]. Available: <https://blogs.unity3d.com/2015/07/01/the-state-of-unity-on-linux/>
- [17] Steam. (2017) *Steam Hardware & Software Survey: October 2017* . [Online]. Available: <http://store.steampowered.com/hwsurvey/>
- [18] Steam. (2017). *Welcome to Steam* . [Online]. Available: <http://store.steampowered.com/>
- [19] T. Andrade. (2016). *VR Games & Apps-Where to Sell?* . [Online]. Available: <https://virtualrealitypop.com/vr-games-apps-where-to-sell-2127697bb70c>
- [20] Nvidia. (2017). *GeForce GTX970* . [Online]. Available: <https://www.geforce.com/hardware/desktopgpus/geforce-gtx-970>
- [21] AMD. (2017). *AMD Radeon™ R9 Series Gaming Graphics Cards with High-Bandwidth Memory* . [Online]. Available: <http://www.amd.com/en-us/products/graphics/desktop/r9>
- [22] Nvidia. (2016). *NVIDIA TITAN Xp Graphics Card with Pascal Architecture* . [Online]. Available: <https://www.nvidia.com/en-us/design-visualization/products/titan-xp/>
- [23] AKiTiO. (2017). *AKiTiO Node — Thunderbolt™ 3 eGFX expansion chassis for eGPUs* . [Online]. Available: [www.akitio.com/expansion/node](http://www.akitio.com/expansion/node)
- [24] Unity. (2017). *Game Engine* . [Online]. Available: [unity3d.com/](http://unity3d.com/)

- [25] Unreal Engine. (2017). *Game Engine* . [Online]. Available: [unrealengine.com/en-US/what-is-unreal-engine-4.com/](http://unrealengine.com/en-US/what-is-unreal-engine-4.com/)
- [26] Group, K. (2017). *The Industry's Foundation for High Performance Graphics* . [Online]. Available: <https://www.opengl.org/>
- [27] Pino,N. (2017). *Oculus Touch review* . [Online]. Available: <http://www.techradar.com/reviews/oculustouch-controller>
- [28] PlayStationMove. (2017). *PlayStation Move Accessories* . [Online]. Available: <https://www.playstation.com/en-us/explore/accessories/vraccessories/playstation-move/>