

# iCreate Technology Review

Hannah Solorzano - Group 61

CS 461

Fall 2017

---

## Abstract

The iCreate virtual reality (VR) project requires the user to perform extremely precise gestures to create architectural structures using generative design. As such, it is impervious that the the technology constraints allow the user to perform exact gestures while having the program able to render the 3D object with vivid detail. The three technologies analyzed in this paper are: the user input controllers, the game engine/framework, and the graphics card (GPU).

---

## I. MAIN OBJECTIVE

Reviewed by Alannah Oleson.

Grade for reviewer: 1000

Overall structure - what is there looks good, but it's missing two sections - see my comment below.

Organization grade: 4/5 - would probably be 5/5 if it had the missing sections. Also no references at all.

Ideas grade: 4.5/5 - All your ideas are good, they're just not very well fleshed out. I do understand that this is a draft though, and you'll probably fill out these sections more when you write up the final. The sections definitely need more detail for the final submission. In addition, I think your paper would benefit from a whole lot of references. So your team doesn't know how accurate some of these things are - are there articles that compare them? Try to find someone else who did the work already.

Formatting grade: 5/5 - Looks good.

Spelling, sentences, grammar grade: 4/5 - There are minor spelling and grammar mistakes throughout, but overall this looks fine. Again, I'm reviewing this as a draft - I'd expect you'd want to re-read this before the final submission.

This is an inline comment.

The purpose of the iCreate

Why is it called iCreate? Does it have to do with Apple products? Just curious.

project is to allow people to create 3D architecture designs using generative design. This means that the user can instantiate a 3D shape as well as a trajectory path, and then the program will auto-generate multiple copies of the shape along the path, creating a more complex structure.

Two sections missing: Your role in the project; and your projected accomplishments for the rest of the year. Though these are really vague, they're up on the guidelines for the assignment on the class webpage.

## II. TECHNOLOGY 1 - GAME ENGINE

### A. Overview

The game engine that will be used in the development of this project is required to have an easy learning curve,

This seems to be a criteria rather than part of the overview

as well as the capability to render detailed graphics using a series of libraries.

This also seems to be a criteria rather than something for the overview.

### B. Criteria

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.

I think you may have reversed the overview and criteria section contents. In my paper, I consider "overview" to be the general description of the tech piece, and the "criteria" to be things that your ideal tech would have. So the tech choices you present should hit as many of those criteria as possible, and the disadvantages of the techs would be that they don't meet certain criteria.

### *C. Potential Choices*

1) *Choice 1 - Unity*: Unity, 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.

2) *Choice 2 - Unreal Engine*: Unreal Engine 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.

3) *Choice 3 - OpenGL*: Unlike the other two choices, OpenGL is not a game engine. Instead, it is an 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.

### *D. 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.

I don't know if I buy this discussion section. I'd really like to see the advantages and disadvantages of each here. This will also help you with the conclusion below.

### *E. 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.

### III. TECHNOLOGY 2 - GRAPHICS CARD

#### A. 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.

I think that this overview and criteria fit their sections really well! So maybe consider reworking your first overview/criteria sections to be more like these.

#### B. Criteria

The GPU is required to effortlessly render the constantly updating program. As such, the minimum requirements are at least a GeForce GTX 970 or an AMD Radeon R9 290. 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.

#### C. Potential Choices

1) *Choice 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.

2) *Choice 2 - NVIDIA TITAN Xp*: NVIDIA claims that this GPU 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.

3) *Choice 3 - Intel HD Graphics 520*: This is the standard GPU that comes with most i7 core CPU computers. Since this comes with the computer, it is not required to be a separate purchase.

#### D. Discussion

The team considered the budget for the project when considering which GPU to use. As the Intel HD Graphics 520 already comes with our computers, it appears to be more affordable. The NVIDIA TITAN Xp is clearly not an option as the budget is not large enough, and the GeForce GTX 970 is affordable, but it's rendering power is not strong enough.

#### E. Conclusion

In the end, the team decided to use the Intel HD Graphics 520 as we already possess this GPU.

This might require some additional justification rather than "we have it". It works for me, but Kevin and Kirsten might not be as into it

### IV. TECHNOLOGY 3 - USER INPUT CONTROLLERS

#### A. Overview

The third 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.

## B. 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.

## C. Potential Choices

1) *Choice 1 - Oculus Rift Touch Controllers:* The Oculus Rift Touch controllers 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.

If it's not known, this is something your should look up. Find sources and references that compare the sensitivities or something.

It is not known by the team members how precise this controller is.

2) *Choice 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.

3) *Choice 3 - Sony controllers:* The Sony controllers 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.

## D. 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.

## E. 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,

How did you find this? Did you test them all? LEt us know more about it here.

and there is no additional costs to including them.