Zack Khan
Eashaan Kumar
FIRE 170

Final Report

1. **Introduction**

VirtuArt is a virtual reality gallery that museum visitors can contribute to. The gallery takes full-advantage of the added layer of depth that Virtual Reality brings to the table, allowing visitors to add lines that hold their position in 3D space, as well as add 3D objects that move within the VR environment. The animations and movements of the 3D objects depend on each object. For example, a bird will fly around in a circle, whereas a volcano will erupt and spit out lava. We chose these 3D objects to make the space more immersive and to ensure that our app utilizes the alluring animations that Unity and VR can bring.

We made VirtuArt to address the major deficiencies in the current art museum experience: a lack of engagement with visitors and a lack of technology incorporated within exhibits. With VirtuArt, we extend the power to contribute to the collection to ordinary visitors, a right that previously was only reserved to professional artists.

For design choices, we chose to add a pointer (reminiscent of a laser pointer) to allow for visitors to easily aim their controllers towards the desired area where they want to paint lines or place objects. We purposely made this pointer white in order to contrast it from the darker environment. We also added a VR representation of the controller, which appear as blocks in the 3D space, to teach the visitors how to properly use the Oculus Touch controllers. All of our design choices focused on ease of use, and making things intuitive. Having an intuitive design reduces the amount of instruction time that is needed to introduce new visitors to Virtu Art, saving crucial time that the visitor could instead be using to create their art and tell their story.

The most difficult parts of the project were incorporating the drawing feature, as well as animating the 3D objects to be interactive. The drawing feature required mathematical calculations related to linear algebra (Matrix operations in a 3D space). Luckily, Zack was able to find tutorials online to help guide him through the process. Furthermore, the tutorials online were all geared towards using the HTC Vive, which has a different development process from the Oculus Rift headsets and Oculus Touch controllers we have in the lab. Zack had to find resources online to help him map HTC Vive controller commands to the Oculus Touch controllers.

Eashaan spent a significant amount of time incorporating the interactive 3D objects. Since he had to incorporate custom animations and movements for each individual object, this task took a significant amount of time. Furthermore, Eashaan also helped incorporate the multiplayer portion of the app, using the Unet framework to allow multiple players to make art within the gallery. This was very difficult since we both had no experience with networking, yet we were still able to implement this feature through a long series of trial and error. Although Eashaan has prior experience with Unity, networking was something he had not touched within Unity before, and since Zack had very limited experience with Unity previously, implementing this feature was a significant learning experience for both of us.

This experience developing the app taught us how to properly manage teams with members of different skill levels. Although Eashaan and I had different levels of exposure with Unity, we were still able to divide tasks up efficiently and learn new Unity features and skills. This will no doubt be invaluable in the workforce or with other student groups, where skill levels vary with different technologies.

A significant amount of thought was put into both the design and implementation of this multi-faceted project In this report, we will go in depth on the context of the project, such as our motivations for completing the project and as well as offer an analysis of our design choices such as the pointer mentioned above. We will also go in depth on how we implemented this complex project using a technology known as Ray Casting, and give our overall insights and conclusion about the project as a whole.

2. **Background**

We have done substantial research to ensure that there is a clear and prevalent need for our project. Many of the studies we read, such as the NMC Horizon Report: 2016 Museum Edition, indicate the importance of museum visitors formulating fond memories and connections. The various papers we read all indicated that, in order to generate a base of visitors that continue to visit the museum regularly, they must generate fond memories at the museum that cause them to want to return. Furthermore, Michael Fischer of Stanford wrote a paper regarding implementing Virtual Reality within an art museum, and states the importance of collaboration and social communication within the VR space.

When visitors use our project, they create fond memories while creating their own art. They form a connection to the museum, as their work will eventually get curated within a virtual collection, that they can come back and see anytime. This experience puts the visitors into the position of an artist, which engages them and encourages them to want

to continue creating their own art (propelling them to visit the museum on multiple occasions).

Our project incorporates a multiplayer approach, allowing multiple users to be within the same VR space. Taking inspiration from Fischer's paper, we wanted to ensure that visitor did not feel isolated within the VR space: especially if they are new to VR technology. Being able to see a representation of another human within the virtual space makes the experience more comforting, which helps with creating fond memories for the users.

As stated in the introduction, we aim to address the absence of immersion within art museums. Multiple surveys and online articles, such as a Business Insider article titled "Why awesome museums are still boring", highlight the sentiment that museums are not catered to the visitor experience. Jake Barton, an expert in the field of museum experience and designer/ founder of the museum consulting firm [Local Projects](#), states that "the museum has failed to capture humans' deep-seated love of storytelling."

With our project, we allow the visitors to tell their own stories through the art they create within the virtual space. What is even more impactful is the ability for visitors to eventually see the stories laid out by past visitors, and seeing how stories change over time. The fact that the VirtuArt gallery will constantly be changing as new visitors arrive again provides yet another reason for visitors to continue visiting the Phillips collection on a regular basis.

Current museums try to employ a "mixed-media" approach to try to engage and immerse visitors, such as by using audio-tours or mobile apps. However, none of these approaches allow the visitors to tell their own stories like VirtuArt does.

There is a dire need to address this lack of immersion. As stated by Nicole of the Phillips Collection during our first class, the typical demographic of Phillips Collection visitors is elderly white women. Younger audiences require high levels of immersion. With the significant level of immersion that VirtuArt brings to the Phillips, we are confident that our project can bring in younger audience members, and turn them into recurring visitors.

There is not a significant amount of work in the area of creating VR galleries within art museums, let alone VR galleries that let the visitors add content. The idea is quite novel. In terms of VR gallery applications on the market today, the top VR gallery application is Google's Tilt Brush. Tilt Brush has an array of features for customizing VR

art and creating truly unique pieces of VR art. However, Tilt Brush lacks a multiplayer component, as well as 3D interactive objects. VirtuArt has both of these key features, and is more strongly suited for the art museum environment.

3. **Analysis and Design**

Our project (on a high level) can be broken into three major components that work together cohesively: the drawing/line feature, the interactive 3D objects, and the surrounding environment. All of our design emphasized ease of use, as we want visitors to be able to jump right into the VR experience with as little instruction as possible. We initially wished to finish the curation and storing work this semester, but we were unable to do so in a timely fashion. We instead prioritized networking and adding a social component to our app.

The drawing component requires significant input from the Oculus Touch controllers. It allows the users to draw lines in 3D space as they move the Touch controllers. We implemented the Color Picker in order to easily allow the users to change the colors of their lines. We scaled the Color Picker to be very large and noticeable so that users could easily make slight variations in their colors.

The interactive 3D objects design were also focused on being easy to switch through. Through a simple press of a single button on the Oculus Touch controller, visitors can easily cycle through their desired object. In regards to the animations, the animations are automatic, and do not require any set-up by the visitor. The design of these objects is very bright, and each object is very distinct and different from another. This allows the user to create more vibrant and unique works of art.

We focused intensively on making the surrounding environment as easy to understand as possible. We used a room with 4 walls to not overwhelm the visitor with too much space, but still give the visitor enough room to add whatever objects and lines they desire to the space. Furthermore, the VR representation of the Oculus Touch controllers, (appearing as blocks with labels that show the visitor how to use the application) plays into our overall focus on a design that is easy to use. The blocks have button on them that indicate the placement of buttons on the Touch controllers. Under each button is a label explaining what each button does. For example, next to the left block there is a label for the button titled "Paint" which shows that the top button on the right controller is used for the painting feature.

4. **Implementation**

Our implementation was divided into two parts: Zack implemented the painting tool and Eashaan implemented the object selection tool and the networking. To implement these features, we had to use some basic Unity concepts to our advantage. For example:

Raycasting

Raycasting is a way to find what object the user is currently looking at. We take the starting point as the position of the camera and point the ray in the direction of the "transform.forward" of the camera, ie the camera's normal vector. Then, the ray is creating using Unity's Physics class and if it hits an object, the method returns the collider of that object. This is useful to calculate where the the user the pointing the left touch controller for selection colors and where to spawn objects.

First Person Controller

Because we didn't always have a VR-ready PC at our homes, we had to figure out a way to test the features with a player that could be controlled through the mouse and keyboard. That is why we used the Unity's default First Person Controller to move around in the scene and test networking. This allowed us to get a feel for how it would feel to wear a headset without having access to one. The drawbacks were that we had to implement the same code twice: one for the VR player and one for the First Person Controller.

Drawing

This involved some concepts from Linear Algebra and Matrix Manipulation. The reason for so much math is that drawing is essentially live-matrix creation. As the users move their right touch controllers around, our code instructs Unity's Line Renderer object to add more triangles to the line with a certain width, color/material, and direction.

Unet

Unet was difficult to debug. Eashaan was responsible for this part and it was very tough trying to learn Unet for the first time. Unet is Unity's own networking system that, if learned correctly, can easily allow the developers to add multiplayer features. The main problems were that to test multiplayer, we had to build a standalone game every time, which was very time consuming. Each gameobject that could be spawned over the server needed to have a Network Identity script attached to it and have it's "Has Local Player Authority" box checked (or unchecked it not the player). Before spawning any object, it had to be registered with the master server, which took Eashaan a lot of debugging to figure out. But the most exciting part was have it finally working on LAN, ie on any computer connected to umd-secure wifi.

5. **Testing**

We followed an iterative development cycle, constantly testing new features as we created them. For example, as soon as the drawing feature was polished, we tested the feature with various peers within the lab. We asked a wide range of students, from undergraduate students to older, graduate students.

We listened intently to the feedback we gained from our peers while trying the project within the VR space. They all gave mostly positive feedback. Some students did mention, however, that they may have felt a slight degree of motion sickness while using the app. To address this, we slowed down the movement time to ensure that visitors were not moving too fast within the VR space.

In the Appendix, we have included two videos that demonstrate some of our testing. One of the students testing is an undergraduate student, while the other student is a graduate student. Both are testing separate features (the undergraduate student is testing the 3D object feature, whereas the graduate student is testing the line-drawing feature)

6. **Results**

The multiplayer component, using the Unet library was a brand new experiment that we had absolutely no background in. We experimented with other libraries as well, such as the Photon Networking library, but they were not successful. We chose this experiment because we felt that the networking feature was crucial to create a social and comforting VR experience. Luckily, this experiment was successful, and we were able to fully implement the feature with keyboard and mouse.

An experiment we carried out was to see what kinds of art people would make when first put into the VR environment. We found that the majority of people began with writing their names in the VR space, and then proceeded to randomly place 3D objects across the room.

We also were even able to have a working prototype done for the Maryland Undergraduate Research Day, where we presented our project. We received praise from both faculty and peers, and it was a very rewarding experience.

7. Conclusion

Overall, we are extremely proud of our work. We worked very efficiently as a team, met all of our deadlines, and made a project that met our initial deadlines. In regards to future work, we have not implemented a system to store the pieces of art that visitors create. In the fall, the project can be completed through developing a way in which the visitors 3D environment can be stored and curated so that other visitors can see their art as well. There are also a few bugs with networking that need to be fixed, such as allowing the changes to artwork to be shown live .

It was a great experience to be able to present our project at the Undergraduate Research Day, and we both learned a great deal about developing software projects on a team, as well as how to conduct and present research. Overall, we had a great experience with this project and are excited for its future.

8. Appendix

Testing video 1
https://drive.google.com/file/d/0Bx4GzUKVi5khaUZiaDh6MExpY3M/view?usp=sharing

Testing video 2
https://drive.google.com/open?id=0Bx4GzUKVi5khVlpYQkkzREcwSWs

System manual
All code is in the Asset folder in the unity main folder. We have created an ArtObject class that belongs to any object that can be placed via hand-controller. ArtHolder is the object that allows us to select from ArtObjects.

User Manual:
First download and unzip our project. Open Unity and click on "Open" to go to the folder where the unzipped contents are located. When you click on the folder, Unity might prompt you with a dialog box saying the the last time this project was opened in a different version of Unity and needs to make some changes. Click Ok and you will now enter the main Unity Scene Editor. Do not touch anything in the Hierarchy window or the Inspector window. To start the game, click on Edit > Project Settings > Player Settings. A panel is pop up. Go into Other Settings and make sure VR option is enabled. Then, Go back to the Project Settings panel and from the left scroll view, select Build For PC/Mac/Linux and hit Build and Run. It will prompt you to enter in a name for the executable and where to save it. That doesn't really matter. Hit ok, wait for it to build, and the game will automatically run.

Code Listing:
There is way too much code to be put into this report. However, all of our code can be found in the "_Scripts" folder in Assets folder in the main Unity project.

Bibliography

Freeman, A., Adams Becker, S., Cummins, M., McKelroy, E., Giesinger, C., Yuhnke, B. (2016). NMC Horizon Report: 2016 Museum Edition. Austin, Texas: The New Media Consortium

Volker Kirchberg and Martin Trondle (2012), Experiencing Exhibitions: A Review of Studies on Visitor Experiences in Museums, The Museum Journal

Bennett, Tony. 2005. The Birth of the Museum: History, Theory, Politics. Abingdon, Oxon, New York: Routledge.

Baur, Joachim, ed. 2010. Museumsanalyse: Methoden und Konturen eines neuen Forschungsfeldes. Bielefeld: Transcript Verlag

Falk, John H. 2009. Identity and the Museum Visitor Experience. Walnut Creek: Left Coast Press.

Unity3D API: https://docs.unity3d.com/ScriptReference/

Raycasting: https://unity3d.com/learn/tutorials/topics/physics/raycasting

UNet: https://docs.unity3d.com/2017.1/Documentation/Manual/UNet.html

Tilt Brush: https://www.tiltbrush.com/air/artists/peter-chan/

Painting Tutorial: https://www.youtube.com/watch?v=eMJATZI0A7c

American Museum of Natural History:
http://www.amnh.org/exhibitions/permanent-exhibitions/biodiversity-and-environmental-halls/milstein-hall-of-ocean-life

Oculus SDK: https://developer.oculus.com/downloads/

Business Insider Article:
http://www.businessinsider.com/why-awesome-museums-are-still-boring-2015-11