3D Lightboard: A Space for Organizing and Sharing Ideas

Brian Tang

University of Wisconsin - Madison

1 PROJECT DESCRIPTION

3D Lightboard will allow users to design, visualize, and share their ideas in a 3D space. Brainstorming and designing ideas has always been limited to 2 dimensions (Whiteboards, Google Jamboard, Sticky Notes). With the emergence of the coronavirus pandemic, people have largely resorted to collaborating and working remotely. 3D Lightboard seeks to solve this problem and create a unique space where people can collaborate on designing ideas without being restricted to a desktop monitor or laptop screen. Taking inspiration from Google Tiltbrush as seen in Figure 1, the space will allow people to intuitively draw and design creations in a 3D space. Among the features to be added are:

(1) A free draw 3D space
(2) A 3D graphing space with grid-lines which drawings will snap to
(3) Saving, loading, and sharing spaces
(4) Integration with normal computers
(5) Simultaneous collaboration (peer to peer or server based)

In a world where VR becomes increasingly prevalent, it's important to have tools and software to take advantage of the unique experience provided in a virtual 3D space. I believe this project will have a large impact in the way people work through ideas and problems and even make certain tasks more intuitive. Applications that may directly benefit from this project include creating 3D engineering designs, working through math problems, brainstorming ideas, teaching classes, visualizing graphs, planning projects, and more.

Figure 1: A visualization of Google's VR Tiltbrush App.

2 DISCUSSION

As the scope of this project is relatively small, and due to constraints of me being off campus, I've decided to work on this project by myself. I already have access to an Oculus Rift S which I will use with Unity to develop this project. While I am experienced at designing and programming my ideas, my experience with Unity and C# is limited. That being said, I plan to look at existing code and tutorials for creating a paintbrush to better understand how these systems work together. I have very limited experience with 3D modeling and art, though this shouldn't be much of a problem since there will be very few assets to create. This project is not without its shortcomings though, as some of the features (such as simultaneous collaboration and cross platform support) are outside the scope for this semester.

2.1 Timeline

If the project goals and milestones are hit, the final product will allow users to draw in a 3D space, save and load their spaces, and collaborate with others on projects. The timelines will be as follows (because I'll have more time towards the end of the semester):

(1) ~~**November 15**: Finish a minimum viable product for the drawing mechanic.~~
(2) ~~**November 30**: Add loading and saving rooms/drawings.~~
(3) **December 5**: Add erasing, undoing, colors, 2D surface snapping, and 3D grid-lines.
(4) ~~**December 8**: Add different stroke types/materials (lines, ribbons, shapes, and planes)~~
(5) **December 10**: Add basic peer to peer multiplayer support.

In the case where my schedule becomes overwhelmed or major obstacles appear, I can consider having a basic 3D drawing, saving, and loading system a bare minimum success. This is something I intend to continue working on after graduating this semester, so even if things don't work out, there are more opportunities in the future.

3 WEEKLY PROGRESS

3.1 Week 1

I have made the following contributions in the past two weeks:

(1) Finished setting up the Unity VR development environment.
(2) Done some research on TiltBrush to get a better sense of the direction of the project.
(3) Downloaded and looked through the source code of an existing VR drawing asset.

The main challenge I've run into is that SteamVR is failing to render hand devices with my Oculus, meaning I'm unable to use that package for development. This means that the drawing asset I downloaded is unable to be used, and I will have to create basic VR functions using Oculus development tools.

The current plans are to go through some Unity-Oculus tutorials to get some base functionality working (like hand manipulation). I

will then use the source code from the drawing asset I downloaded as a template for creating my own drawing mechanics from scratch.

3.2 Week 2

3.2.1 Contributions.

(1) Finished Unity+Oculus hand presence tutorial.
(2) Got the existing VR drawing asset set up and working with Oculus.
(3) Identified more issues.

3.2.2 Challenges.
The main issues with this existing drawing asset are that the drawing material is difficult to work with (draws more like a paintbrush than a marker), and the fact that there is no plane-snapping makes drawing words and diagrams unintuitive. Saving and loading drawings is built into this tool, but saving multiple drawings is something I need to add. Also, I'm not a fan of this UI, so I will likely change it into a menu that can be toggled with a button press. A smaller change involves swapping the presence of controllers with maybe a marker and eraser.

3.2.3 Plans.
Next week, I plan to add new stroke materials and the capability to save multiple drawings. I may also change the UI if I can get around to it.
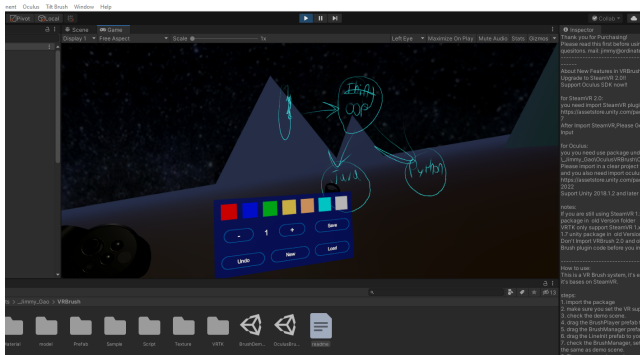

Figure 2: The drawing package

3.3 Week 3

3.3.1 Contributions.

(1) Added new drawing materials that are easier to write with.
(2) Added ability to save, load, and delete multiple files.

3.3.2 Challenges.
Luckily, I didn't run into any issues or challenges this week. Editing the scripts and adding UI elements was pretty straightforward. Adding multiple JSON files for saving and loading was a bit more difficult. Currently, there is no way to name files or search for them, but this would be a difficult UI to implement.

3.3.3 Plans.
The next steps that follow are to add a keyboard functionality when saving files and a better interface for loading. This is a less important change, and most likely I will focus on adding erasing and 2D snapping for next week.
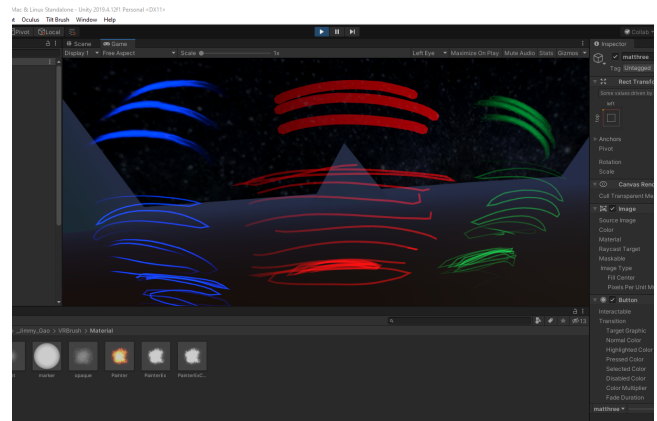

Figure 3: The different material/stroke types

3.4 Week 4

3.4.1 Contributions.

(1) Added code for grid-snapping
(2) Started eraser function
(3) UI additions

3.4.2 Challenges.
My headset wasn't working on the day I set aside to work on this project. I'll have to diagnose this as sometimes Oculus detects no headset for some reason. The eraser functionality is going to be more difficult than I had originally imagined. It seems there's no easy way to erase an area with the current implementation of the drawing tool. The next best thing I can go for is to select and erase entire lines based on the position of the controller.

3.4.3 Plans.
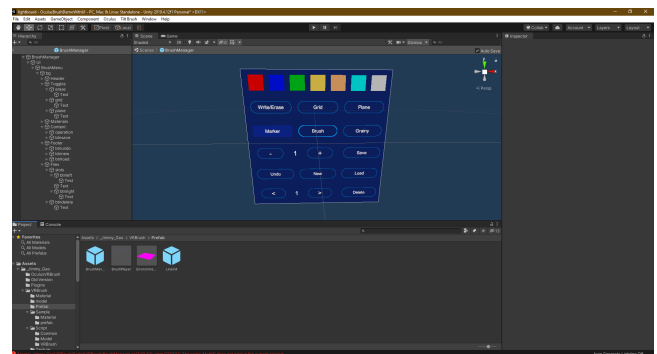I think I can afford to spend more time testing and working on these features, since I'm ahead of schedule.

Figure 4: The current state of the user interface

3.5 Week 5

3.5.1 Contributions.

(1) Tested and fixed code for grid-snapping
(2) Fixed bug with saving and loading multiple drawings

3.5.2 Challenges.
The erasing function is proving to be quite difficult and more complicated than I first imagined. Other than this, I'm making good progress and only need to complete multiplayer support and erasing.

3.5.3 Plans. I have quite a busy week coming up, so I won't be able to make any big contributions. I will likely work on minor UI quality of life improvements.

3.6 Week 6

3.6.1 Contributions.

(1) Identified issue with erasing implementation
(2) Looked at tutorial for multiplayer support with Unity VR

3.6.2 Challenges. For the sake of time, I will likely move the erasing function to a lower priority because of the way lines are constructed in the space. Multiplayer support seems quite difficult to implement, especially since I don't have the equipment to test it.

3.6.3 Plans. If the multiplayer implementation doesn't pan out, I will prioritize adding keyboard and directory saving/loading support. With this, I will change how the menu UI is accessed (via button instead of attached to controller). I may also consider adding glowing materials and changeable backgrounds (skyboxes). Finally, if I have extra time, I will revisit erasing (and multiplayer if it fails). Eventually, I hope to add more features and try to finalize the project as a free Oculus software.
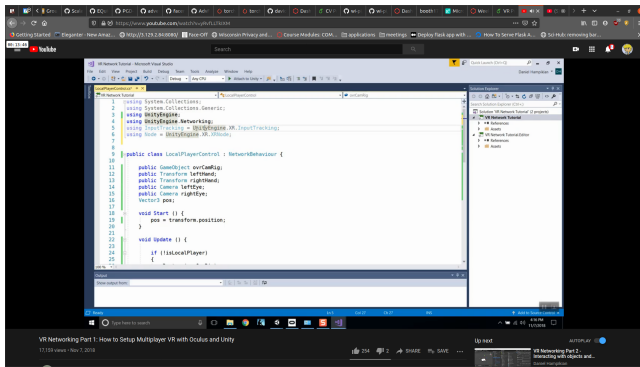

Figure 5: The Unity VR multiplayer tutorial