Space Systems Design Studio SP21 Report

Kyra Lee, kl627*
Cornell University, Ithaca, NY, 14853

I. Introduction

My role on the publicity team was to assist in creating and modeling a 3D environment reminiscent of what the CubeSat would resemble when the holograms were mounted on it. This included developing accurate computer renditions of the analog holograms users would choose from, implementing computations of the CubeSat's physically accurate movements, simulating the spatial surroundings, enhancing overall realism of elements like retroreflective sail material, and establishing a connection between this environment in Unity and the website through WebGL. In addition, I was responsible for programming the triggered events that occur when a user chooses between holograms to view on the website in a seamless manner, utilizing timed transitions to allow fading mechanisms when the hologram mounted on the CubeSat changed. Such modeling of holograms on the CubeSat was important to the mission to promote civilian involvement in this endeavor by allowing them to choose which holograms would enter space. Visualizing these holograms on an accessible web platform allows for such.

II. Major Assignments

a. Simulated accurate spatial environment in Unity

This includes correct earth axial tilt, atmospheric 'glow' for cinematic effect, revolving at a proportional speed, proper sun positioning relative to the earth (evokes a flare when the camera is on it), and constellations for flair. Revolving script is attached to the Earth object, along with the texture and the point light that emits the glow. Sun is used as a directional light, and one can adjust its brightness if needed. Constellations instantiated using a skybox.

b. Recorded all holograms

Each of the 7 holograms was recorded using a DSLR camera, attempting to make each recording mimic what the hologram would act like in space for seamless integration into the environment. Used a box with the insides painted black to absorb light, taped the holograms with blue masking tape to clearly indicate its edges for editing purposes, and carefully moved a lamp in lateral movements (similar to how the sun would act if the holograms were mounted on the moving CubeSat). The camera was set to Aperture Priority and kept still on a tripod as the lamp was moved to best show the hologram's "movement" and color changes. Must be recorded in a pitch black room wearing all black clothing, with as little reflection (from jewelry, shiny curtain rods, etc) as possible.

Holograms were cropped to accurate scaling afterwards, and enhanced using filtering by increasing contrast, sharpening, increasing vibrance, or any other tweaks necessary to best show the hologram. The videos were then duplicated, reversed, and added to the end of the original clip to create a circular, seamless video that doesn't have a distinct start or end cut.

c. Oriented CubeSat in environment

The pre-existing CubeSat CAD model with added textures for realism had issues transferring to Unity, namely with its axes positioning - thus we childed the CubeSat to a small sphere (invisible to the user), and moved the sphere so the CubeSat was the main focus of the camera. The earth's outline is barely in the corner, enough so to see its revolution/orient the user, but not enough to distract from the main event.

d. Placed holograms on CubeSat

After all 7 holograms were recorded, we used Unity's VideoTexture feature to create canvas objects of each of the hologram videos (scaling and cropping as necessary). Each hologram was then carefully placed on the four sides of the CubeSat (ie; the CubeSat would only display four of the same holograms at once), taking up most of the cube face to be as accurate as possible. This created the illusion of a glass hologram being mounted on the CubeSat, since the video's transparency can be adjusted to reduce complete opacity. Since the videos seem to be on an infinite loop, it appears that the sun makes them waver and move, just as intended with the lamp movements.

The issue arose of how to best organize the seven holograms in the hierarchy so that they would be positioned correctly at all times, but not visible until the user chooses them. After considering some data structures, the determined solution was to 1) position/scale the four hologram VideoTextures the way they would appear on the CubeSat, 2) child each of the four textures to a singular canvas named according to the hologram it holds, 3) set the canvas .isVisible property to False (subsequently rendering the four textures invisible), and 4) set the canvas back to visible when the user chooses. Each of the parent hologram canvases are childed to the CubeSat.

e. Created shader for transitional purposes

Shader slowly "dissolves" the VideoTexture, allowing for easier transitions between holograms appearing/disappearing off the CubeSat when the user chooses a new one. When the property on the shader is True, it begins slowly dissolving the hologram off the CubeSat, setting the hologram's corresponding canvas to invisible. The CubeSat is now a blank space waiting for the user to pick the next hologram, which will also be faded on to retain smooth flow.

f. Mimicked spin behavior of CubeSat in space

Spin script attached to the CubeSat allows the developer to adjust the X, Y, Z force elements of the cube's spin, placing more emphasis on whichever force will influence a spin that is most accurate in regards to the light movement on the hologram. Making the CubeSat "spin" (quotation marks attached because it is more like randomized circular movement) enhances environmental realism and makes the sun's reflections on the holograms look less contrived.

g. Completed integration with the website

Original plan was to record videos of the program build for each hologram, and then invoke videos on the website when a user clicked a button to view a specific one. However, the team member in charge of the website decided the environment disrupted the website's flow, so used WebGL to connect Unity to only show the CubeSat with mounted holograms with a transparent background (the website will be the background, almost like a floating/moving .png). CubeSat starts out blank, when a user chooses a button on the right of the WebGL canvas the corresponding hologram fades on. When the user wants to switch to a new hologram, the old one fades off and the new one fades on. The CubeSat revolution slows when transitioning to emphasize the swap (and enhance optimized efficiency), and speeds up again when the new hologram is fully loaded (the faster spin matches the light movement of the hologram).

h. Optimized hologram transitions

After speaking with Dr. Muhlberger for a mid-semester progress meeting, I began working on the optimization of the transitions between holograms when a user chooses a new one to view. Per his suggestion, I worked closely with our website specialist to determine when the program was slowing down, and we discovered it was the Unity VideoPlayer function that takes a significant amount of time to load and animate. Because this was due to Unity and not our own use of data structures, we decided to slow the spin (as aforementioned in section Ig). By slowing the spin as the CubeSat transitions between VideoPlayer textures, we created an illusion that the entire environment was slowing down, which masks the buffering effect the transition previously had. Despite this being a less robust solution than I'd like, our lack of time only permitted something like this to remedy Unity's personal issue of VideoPlayer performance.

III. Performance

a. Points of merit

Utilized efficient data structures and programming techniques (coroutines, optimization) to attempt making the website as clean and fast as possible. My partner and I committed to making the environment as accurate as possible, from the earth's axial tilt to the positioning of the sun directional light, and although the environment was ultimately not used, it was good practice paying attention to detail for the future. We were also creative with how we modeled the holograms, which were challenging to mimic computationally because of their flatness combined with their perceived depth and color changing images.

b. Points of improvement

The transition between holograms is not as smooth as it could be, due to my limited understanding of the WebGL/Unity connection and lack of experience optimizing algorithms including coroutines. Because we used videos for holograms, this environment is not very robust - everything needs to be moved and positioned in conjunction with the hologram videos, which makes spinning the hologram difficult to look realistic. There were a couple snags in proportions of the CubeSat and the holograms because it was challenging to crop the videos accordingly without Unity placing black bars on the sides, which led to less accurately mounted holograms.

IV. Next Steps

Since my work on this team was purely to create a clear visualization of holograms for the mounting process, after the voting process is conducted there is no more continuation of this project. The website will stay live so those interested can see more information about the mission, so perhaps after voting is concluded we can simply display a CubeSat with the voted holograms mounted, without the user interaction option to switch between hologram options. A mobile application was being discussed previously in the semester, but that is relatively separate from my work done on Unity for the website with the intention of assisting voting processes.

V. Learning Goals

The learning objectives outlined at the beginning of this semester and my progress on them is as follows:

1. **Improve understanding of 3D model movement in a 3-axis environment**
   Was an integral part of overall development, but especially when creating the spin script that took X, Y, and Z forces into account when creating a randomized movement for the CubeSat. Taught me about how each axis affects 3D models, and how to alter emphasis on an axis to move the object accordingly.
2. **Learn how to best create dynamic materials that change based on its surroundings**
   Creating the retroreflective material for the Lightsail (first semester) was the most helpful in this, but simulating the sun's movement when recording the holograms forced me to take into account the sun's position and how it would affect the hologram appearance.
3. **Determine the most effective and secure way to create a voting platform**
   This task was removed from my list of responsibilities, but I was able to learn about how our website specialist was conceptually executing this task by tracking IP addresses and only letting each address vote once. Votes will then be stored in an external location (ie; a spreadsheet) for our team to eventually parse through to determine winners.
4. **Learn the relationship between Unity and web applications**
   After successfully deploying our environment on the website, I better understand how WebGL establishes a connection between Unity and the external site. The process of creating functions locally in Unity/Visual Studio so the site can call them upon a mouse event was much simpler than I thought, and it opened more avenues for communication between the interface of the site and the Unity build. I previously only created static builds in Unity or ones in which the user interacted with the Unity program directly, so it was interesting to see how Unity can be used in other mediums as a form of visualization.
5. **Understand the synthesis between physics/physical systems and programming in the process of creating an environment that is as accurate and functional as possible**
   Learned more about this when creating the spatially accurate environment. Although we ultimately decided not to use the environment, the process of making the earth's revolution and the sun's positioning helped me better understand how to accurately orient static physical objects in the surroundings without hindering CubeSat functionality and focus.

6. **Learn server communication in iOS apps**
   Unfortunately I did not have time to complete this task, but I hope to communicate with the next team member who works on this in the future semesters to learn more about their approach!

Overall, I am pleased with the amount of knowledge I obtained through this team this semester; almost every single one of my learning goals was achieved in some capacity. Much of my learning was through research or communication with other team members on their areas of expertise, which felt indicative of what it would be like to be on a software development team in the workforce. My main goal for myself was to expand upon my previously held Unity knowledge, which was certainly accomplished - I feel very comfortable with creating a Unity environment from scratch and implementing any desired functionality. As an added benefit, I was also able to learn more about web application development, which will surely be helpful in the future. This project has sparked an interest in cross-platform communication and overall optimization for me, which I have been reading about even after the conclusion of our deliverables, and I anticipate a continued interest in it over the summer. I am grateful for this team for offering the exposure to new topics and the chance to improve my current 3D modeling skills, and I look forward to the launch of our website and CubeSat.

VI. Acknowledgments

I want to take the time to thank Dr. Muhlberger for his guidance throughout the process, and my partners Cella and Amy for fostering such a supportive work environment.