VR Jerusalem: Post Mortem

An experiment with VR and Stereographs

The Team : Who did What

- Jacob Westerback
    - Team Leader
        - Organized and Communicated with team
        - Managed Tasks
        - Collated efforts
    - Javascript Build
    - Late Programming/Testing
        - Wrote Subtitle System
        - Created Opening Scene
        - Managed Unity Content
        - Fake VR Build
    - Voice Talent and Audio Parser
    - Scriptwriter
    - Documentation
    - Presentations
- Matt Dieselman
    - Unity Expert
    - Created Camera System
    - Zoom Scripts
    - Early Programming/Testing
- Ryan Flanangan
    - Stereograph Expert
    - Photo Editor
    - Zoom Selector & Editor
    - Poster Creator and Printer
    - Presenations
- Chris Littlefield
    - Voice Talent
    - Scriptwriter
    - Audio Recorder and Parser
- Jeremy Astolfi
    - UI Creator
    - Reciticle Creator
- James Sasson
    - Voice Talent
    - Audio Recorder
    - Task Distribution / Whiteboard Leader
    - Early Programming/Testing
    - Documentation

Executive Summary

VR Jerusalem is an app that takes stereographs and presents them for google cardboard. This project takes a new spin on century old stereograph images, converting images to a virtual reality space. The idea is to give a new life to these images, and tell the story behind the moments captured in a radically different period than our own. The viewer will be presented with a historic photo. The goal is to emerge an audience in a historic time period in a area foreign to many.

By having these images available in an application, they can be much more accessible to the public, museums can put them on display, they can be more easily shared, and there is 0 risk of damage to the historical documents themselves.

Main Goal

Create a VR experience for Stereograph images provided to us by the University of Pennsylvania.

Links to Content

JS Build: https://people.rit.edu/sjc5283/slideshow/
Github Repository: https://github.com/solanin/VRJerusalem

Hopes For the Project

We had hoped to go so far as to include modern 360 images in our project. To accomplish this goal, we would have to have 360 degree image data, which was not available in the locations we wanted nor would it work inside of the same program without massive changes. 360 image data loads an entire zone based on the data, while we are simply loading two pictures. To correct for this, we would have had to completely rewrite our handling of the images. In the modern day, stereographs are no longer used so custom content would have to be created.

We also hoped that we could create a sort of ‘Museum” environment for the user to move around in and explore. This, however, was pushed back due to the lack of progress on presenting the Stereographs. If we were to move forward, this could be explored more thoroughly.

Start To Finish: A Look at Our Progress

The JavaScript Mockup: Jumping off

The first prototype we made for the project was the Javascript build. This was done as a proof of concept. The build is located here: https://people.rit.edu/sjc5283/slideshow/

The goal with this was to show the image in such a way the it could be seen on a phone with a google cardboard. This was accomplished with this build. The final build worked off of the same premise as this one.

After this build was made, the team decided to continue work in Unity because that platform would allow us to do more complicated 3D things in addition to the stereographs, such as 3D video of modern times, or walking around a virtual museum.

What We Accomplished

Our final build accomplished our main goal of providing a stereographic 3D experience of Jerusalem in the 1900's. While it is simple in nature, it did prove that we can create something that could be used in a museum exhibit on a phone using the Google Cardboard. We were also able to include audio and subtitles for those who wish to learn more about the images. We are happy that we can include the deaf friendly option of subtitles, as we weren't sure if it would be doable due to the nature of the project's 3D effect.

Our final build is located here: https://github.com/solanin/VRJerusalem

Road Blocks

There is more detail about the specific challenges that we faced in the Timeline section, but the major Roadblocks we faced were Version Control, Unity Builds, and 3D effect mixing.

Github

Version Control is a common issue in projects such as these, especially concerning the combination of Github and Unity. Our issue came with the ability to push. We had multiple times when we had to reset our git repository to a new one because one or more of our team members could not push their changes to the repo. Surprisingly, merging was not the issue, it was simply an issue with Github accepting the code.

Building to Android

Getting our Unity Builds onto android was also a major roadblock. We had a lot of builds that worked perfectly in the Unity editor, but when we built onto Android, it would break our camera or the layering, or crash immediately. This caused a lot of issues as we weren't sure why it was breaking until we moved to a new version of Unity and the problem ceased to exist. Ultimately we were able to build to android with 5.4 but we were never able to solve the initial problem as to do so we restarted the project.

3D Effects

One thing we did not realize till the final weeks of the project was that you cannot mix two different 3D effects and still get a 3D image. We tried to use the Google VR SDK to get a 3D effect in Unity. The logic behind this decision was that were we to incorporate other 3D technologies or effects we would need this system. However this system was fundamentally incompatible with the stereographs. This also meant that we could not get a UI created and thus no buttons to display the zoom images we made.

Our Work : Timeline

Stage 1

First Stage:

- Identify How to create the app.
    - JavaScript?
    - Unity?
    - Other Options
- Explore options of presentation
    - Museum Exploration?
    - Slideshow?
- Set-Up contacts
    - Github
    - Slack
- Restart Project Counter: 1

First Stage in Retrospect:

We decided on Unity and chose to discuss the presentation once we had some sort of demo. We set up a Github and a Slack channel and started working with what version of Unity we should use. We started with 5.4 with the DayDream expansion. We set up the Github and started to get familiar with the technology. We were Investigating Stereograph images and how they work. At this point Ryan began cleaning up the images. During this stage Jacob also made a simple mockup using Javascript.

Stage 2

Second Stage:

- Prototype 1
    - Images on Screen
    - Build to App
- Audio
    - Voice Recording
    - Deaf Friendly
- Image Splitting
    - How many images?
    - How to format the images?
- Restart Project Counter: 2

Second Stage in Retrospect:

Once we started working on a prototype, we realized that putting images on a screen in VR the way we wanted too was a bit harder than anticipated. To give a Stereographic effect, we need a separate image on each eye. We wouldn't figure out how to do this properly until about a week before Imagine RIT. We did, however, have a build ready and at least a few images on the screen. We also started working on Audio and the actual images to be in the final version. These were just initial tests to make sure we could do it easily. We were able to get a build with some UI and different images on each camera eye. However this would not build to the phone, it would only display in the unity editor. We attempted to start over and redid our project in 5.4 to attempt to fix some of these issues.

Stage 3

Third Stage

- Build Broken
    - Unknown Errors
    - Running on Unity, not on Phone
- Github Broken
    - No longer Committing
    - Unable to Merge
- Audio and Images Progressing
- Subtitle System
- Other Options
    - Unreal
    - Android Studio
- Restart Project Counter: 3

Third Stage in Retrospect:

This was the stage where we ran into our first major issue. Our build was completely destroyed and our Github wouldn't let us commit or pull in some cases. This lead to us having to start from scratch on the Unity App. It wasn't too hard to get back to where we were, but at this time we explored the other options of Unreal and Android Studio. We decided to stick with Unity, but moved to Version 5.6, as it had the Google VR package built in.

After switching to 5.6, this was the first time we were able to get the app to build on a phone. This is also the point at which the GUI version of the subtitle system was built.

Stage 4

Fourth Stage:

- Unity issues
    - 5.6 could not give us camera control
- Github was being very finicky
- Started from scratch in 5.4
- Built Fake VR system in 5.6
- Content

    - Finished Audio Recording
    - Finished Image Editing
- Began work on Posters
- Restart Project Counter: 5

Fourth Stage in Retrospect:

At this stage we were forced to restart again. After switching to 5.6 we discovered that we were not able to have the camera control that we needed to separate the images. We reached out to some experts - both professors at RIT and people Professor Jacobs put us in touch with but that did not help much. So we worked in parallel development again. Matt led the effort to go back to 5.4 and see if we could resolve the android build issues. Jake stuck in 5.6 to see if he could get the VR working with just one camera. Both efforts were successful. The 5.4 build was able to build to the phone and the cameras could be split - however the split camera only appeared in the editor and would not show in the android build. The 5.6 build also worked however the single camera would not give a stereographic effect when built with VR settings.

Final Build: Imagine RIT

Final Stage:

- Abandoned Google VR libraries
    - Went with the 5.6 Fake VR build
- Content
    - All audio Parsed
    - Subtitles Typed
    - And Posters Printed

Final Stage in Retrospect:

Imagine RIT went very well. People were interested and excited by our project. One person asked if we had a website where he could download the app. Some people were less impressed but on average the reaction was quite positive.

Lessons Learned

3D effects

One thing we did not realize till the final weeks of the project was that you cannot mix two different 3D effects and still get a 3D image. All of our attempts to get the google VR working with our split cameras were futile since even if we had gotten them working- the 3D effect would have been lost. This was a classic issue of overthinking the issue.

Parallel Development

We had to restart our project about 5 seperate times. This taught us that working in parallel was very important if we wanted to end up with something to show at the end of the semester. However since most of the code on this project was dependant on the camera and VR system, that development was constantly put on hold due to the errors we encountered. These dependencies were what ultimately held up development.

Team Size

In some ways the team was too big for the project, and the timeline too short. The workload was not balanced and fell much heavier on some members than others. It was often difficult to make sure that there was enough work to go around and that everyone had something to do. As such we often tried to take turns on the work to be done. On the flipside of this we did nearly run out of time on this project. It is also worth noting that the issues we had with github aggravated this problem. Had we overcome our hurdles sooner than we may have been able to spread the work wider and this may have been less of an issue. However this was not the case, and the hurdles were not overcome near the end of the project.

Future Development

Documentation

This documentation, our final power point on the project and the powerpoint about the image processing progress are all posted on our github. Furthermore, all of our code is documented - this includes a document on how to add more stereographs. This will help with any future development efforts.

Potential Future Goals

- Adding modern content
- New or different stereographs
- Zoom UI
- Museum environment to “walk” in
- Anything else you can think of!