Liberty Jail 3-Dimensional Rendering

Teancum Hoopes*

Brigham Young University

E-mail: teancum.hoopes@gmail.com

Abstract

This project was designed and carried out on a GitHub hosted website with the intent of expanding the understanding of the conditions of Liberty Jail. Liberty Jail functioned as an integral part of the history of the Church of Jesus Christ of Latter Day Saints. The goal of this project is to demonstrate how it might have been to be in Liberty Jail, as well as some of the impact of the conditions that were imposed during the months spent by the men inside.

The scripting language that was used to design this three dimensional model is called A-Frame. It is built on JavaScript which allows for websites to be used inside of Virtual Reality headsets or Augmented Reality headsets, this is then imported into the more common scripting language HTML (Hyper Text Markup Language). This is the ground work of the website as well as the various pieces of integration with being able to interact with the environment.

This paper serves to explain the process as well as the over all mechanisms to which the project was built as well as deployed. It will also cover some of the research involved to design and build the three dimensional model of the website. The virtual tour is hosted at `https://Thisguy217.github.io/start.html` Other projects are hosted here, however, they will not be examined during the course of this article, questions and other concerns are welcomed at the email listed above.

Introduction

The Liberty Jail serves as a somber reminder to members of the Church that no one is exempt from trials or tribulations. The events surrounding the imprisonment of the 6 men in Liberty Jail and the subsequent challenges are spoken of throughout the Church of Jesus Christ of Latter Day Saints world wide. Upon understanding this many people both members and non-members of the Church of Jesus Christ would understand if any of the men decided to renounce their religion in order to end the torment. However, the goal of this paper is not exclusively to retell the story, one may enjoy the tour for such. This paper is designed to explain the relevance as well as the process in which this project was built and deployed. The sections that follow will cover major topics and information concerning the build and questions are welcome at the provided email.

Rendering 3D Model

For the duration of this section I hope to elaborate and explore the process followed strictly from a technical side of incorporating historical information into the build out. It will hopefully portray the information necessary to both understand the process and if necessary build a similar site or application in the future. It will address aspects such as the syntax, the integration, as well as the historical authenticity of certain aspects. This will cover occasional information concerning health impacts, physics, and even over all weather conditions present during the time of imprisonment. Of course, most of that information will be found during the tour of the virtual site and is highly recommended for a better and more clear understanding of those details. If there are any other questions, it is also suggested to visit the historical site or to contact the local tour guides to receive answers to questions not addressed in this paper or the tour.

Structure of the Scripting Language

The website that was programmed was coded in A-Frame, a framework for developing virtual environments on the web.[1] Utilizing the structure of both HTML and JavaScript it is integrated into the HTML script, it does not require an installation, only a reference provided in the `<head>` section of the script. It can also be referenced locally or on a server, a local reference will require a download of the package in a form called "minified." A slang term used to describe the fact that the original JavaScript is very long and very complex and it is compressed for quick access, but can be just as versatile.

A-Frame's syntax is similar to HTML and easy to read. It requires opening and closing tags located inside the appropriate parent tags. A-Frame is accessible as a whole web page or as individual windows that are just as integrated and accessible. Depending on the purpose of the website it may provide useful to have individual windows available all at once. For my purpose it proved far more useful to integrate each room of the original Jail into their own web page. However, the goal of this section is not to explain the details provided in the documentation of A-Frame, but to give an over view of how the structure behaved.

An interesting note of both A-Frame and HTML is being able to generate tags that can encase groups of elements and give them names so that way large pieces of the script can be accessed and adjusted collectively. This is great when working with primitives. It helps to encapsulate sections of code and manipulate large sections of the room with minor adjustments to the script itself.

However, just like HTML, A-Frame comes with the assumption that when you perform this encapsulation that you must refer to the object as a whole. An example of this would be like this section here:

```
<!DOCTYPE html>
<html>
  <body>
    <a-scene>
```

```
      <!--Ungrouped Items-->
      <a-box position="0 0 -1"><\a-box>
      <!--Grouped Items-->
      <a-entity position="0 0 -1">
        <a-box><\a-box>
      <\a-entity>
  <\body>
<\html>
```

While the code may appear to have additional syntax that would be unnecessary, in most cases this method of grouping will ensure that sections of code will compile to similar objects when using primitives. This is less necessary when it is a previously rendered objects such a `.obj` or `.gltf` file. These would not require the `<a-entity>` tag to help compile the 3-D model down into a single line adjustment. Yet, the utility of the tag is still very useful for larger models. Something I learned very late in the project.

One final note of the syntax of A-Frame that more is seen in the final rendering, is the light on seams. While objects may be integrated together and generate excellent primitive objects. They lack a clear defined beginning and end point. This creates some difficulty when rendering and a more realistic effect is needed, such as when there are two primitives side by side, they will appear as one rather than two individual primitives. This can be helpful in reducing the code necessary to run, and speed up loading times, but makes it difficult for also quick building process for the or coder rendering the environment.

Integration of JavaScript

While A-Frame has a great deal of versatility and options for build out in building three dimensional models, it lacks a great deal of interactive features. Especially understanding the multiplicity of platforms and devices it makes it hard to ensure that each receives the same interactivity. This would require an understanding of JavaScript, that I do not have.

However, it was necessary in building out this particular model to ensure the ability to move from one room to the next and allow the tour to function as a whole. I hope to describe some of the ways that I used it and the over all structure needed for JavaScript to function inside of A-Frame.

Interestingly, A-Frame despite the build being centered on JavaScript, it does allow for continued build out with either packages being referenced or installation of the packages needed. Many of the larger built A-Frame scripts prefer to install a physics engine to ensure that interaction will allow for standard things like throwing a ball or dropping a box. Yet, other interactions such as being able to do specific things such as moving from one page to another, from location to location within a map, may require building a script within the HTML script. I will not show the exact details of the scripts that I applied in my web pages, but I can explain the concept. If there are questions or other curiosities revolving on the syntax or structure of JavaScript the reader is urged to research in other sources such as YouTube.

As the project has come to life, and is still classified as living, it continues to see various forms of development. Features are added, some are taken away, and others only make it so far as the mind of the creator before becoming obsolete. However, A-frame provides a great deal of versatility in design and build out for integrating into HTML code and JavaScript.

Build out of Interaction

A great perk of how A-Frame is built is allowing for interaction with the user. Despite the challenges though, it allows for so many things like games, story-telling, animations, and more. There are different ways of doing this and integrating it, and it varies from platform to platform. It should be noted that this challenge is only necessary when dealing with various platforms and users all together. I will go through some of the process needed for this project as well as build for deployment.

The first challenge that needed to be surmised was how to do what is called "cross

platform compatibility" or in other words how to set up the web page to be accessible on multiple devices and provide the same user experience. Due to the design of the build it became easier for access to mobile devices to lock locomotion controls. This will limit the user from being able to navigate the model, but it does come with some interesting perks in that it allows them to view the full room, it is surprisingly small for the actual dimensions. The second thing was to ensure that for the windowed pages that the fonts and sizes remained the same across mobile devices and desktop browsers, as well as internal browsers of devices such as the Oculus Quest.

The second challenge presented for this build was integrating the ability to navigate the model when it required transporting between rooms or other locations. Normally this can be done by many methods, but the more intriguing of them require coding new packages that allowed for transport. Not very many are available for integration directly into A-Frame, but this did not stop me from finding new methods. Due to some of the issues with cross platform and such it actually became easier to create HTML files that have `<iframe>`, or bits of code that allow for the integration of a video or other protocol directly into the HTML file. For example the rooms are localized in their own files, but the access is only made available in an `<iframe>`. Upon completion of this integration other possibilities were available for being tackled, such as with cross platform play where iOS browsers and platforms do not allow for audio to be automatically played. This opened up more options for creating availability on more platforms than before.

Of course, once these challenges were over come there were other other challenges with ensuring the design of the building matched historical information. There was, and is, a continuing search for checking the information provided through historians as to the dimensions, designs, and other features present in the jail. A future section will cover some of the measurements and designing process, but this will be brief as it is something that is explicitly written in the code.

Historical Authenticity

Most of the information provided concerning the jail has been provided by one of the early historians hired by the Church of Jesus Christ, Andrew Jenson. Being accompanied by other historians, they travelled to Missouri and found the crumbling building. They recorded the dimensions, design, and other information they could collect from the building, despite having been fallen apart almost entirely, the information collected has allowed for succeeding historians, programmers, and even authors to render in the modern mind what they building may have looked like.

The information that was collected was recorded in his autobiography.[2] This information was used as well as website information from the Church of Jesus Christ provided on their web site.[3] These as well as the site guide[4] of the Liberty Jail Historic Site provided to missionaries to which I was given opportunity to study and collect information from. These serve as the main sources for the structure of the jail. However other bits of information were collected from personal visits, as well as talking to others who lived in the area currently that give a clear understanding of what the conditions of the jail itself were. I hope to explain briefly some of that.

Though some artist interpretation has taken place it will continue to be under development and designing the Jail to match more authentic images to what we have. The wife of the author provided information from her time in Missouri living in the local area, providing first hand account of weather, conditions, as well as confirming or rejecting ideas of the rendering that explain the design of the jail. During some visits information was also given concerning it, however, limited amounts of what was seen and felt can be portrayed into the design.

Deployment Using GitHub

The deployment of this project took place on a well regarded platform in the Computer Science community, GitHub. GitHub allows for amateur programmers as well as companies that produce software to design, build, distribute, and develop applications, website, scripts, code, and more. This section will briefly describe the method and the following subsection will provide additional details as to the exacts of utilizing GitHub for deployment.

GitHub provides a platform for individual users to generate a website, they could build multiple websites given the domain name owner ship, but a single website can be maintained without needing to without needing to purchase a domain. Other platforms allow for building websites and integration, but a open source and free to use availability is limited and difficult to come by. GitHub Pages provides both the ease of use, the ability to control in depth features, as well as a simple platform for distribution. Of course, documentation is provided on how to perform the specific deployment on GitHub I will address the succeeding features for deployment as they relate to this project.

This project did require building separate web pages, and using the structure and syntax of the GitHub terminal allowed for locating all necessary files of a website in the same format as how the standard computer is designed. The main web page runs on the index and from there through either `<a>` tag or through providing the direct URL HTML connections. As an aside, I am not sure at this time of other methods of connecting, but almost always the HTML file that someone wants to connect requires the exact URL to be given.

Upon completion of designing the structure of the website in the terminal it came to linking and designing the linking of the specific web pages to allow for access to each section. This is where the fore mentioned methods found in the subsection *Build out of Interaction*. However, to briefly sum it requires, for being able to ensure cross platform usage, to design the `<iframe>` window and integrate it into a separate HTML page. Both remain available but only one remains available for touring.

Conclusion

In conclusion, the process of building the Liberty Jail and truly stepping into the room gives a whole new and terrifyingly real perspective of what it would have been like. The challenges of building this and entering into the Liberty Jail and hearing the stories remind me of the challenges and reality as we well as the great blessings that come from this trying times. I hope that this will allow others to feel this and as time can be devoted perhaps more can be built and much better. Other things that may be explored in the future is designing other historic sites, or perhaps just religious stories being given. There are many options for other projects. Perhaps other things can be designed for building stories in 3D models.

Acknowledgement

The author thanks his wife for being so kind as to provide many of the resources used for the historical authenticity of this project as well as other information used for the tour and compilation of historical facts.

Needs more to fill out and finish!

References

(1) Introduction. `http://aframe.io`, Accessed: 2021-11-11.

(2) Jenson, A. *Autobiography of Andrew Jenson*; 1938.

(3) Liberty Jail. `https://history.churchofjesuschrist.org`, Accessed: 2021-11-11.

(4) Department, C. H. *Historic Liberty Jail:Historic Site Guide*; The Church of Jesus Christ of Latter Day Saints.

TOC Graphic

The process of rendering the Liberty Jail using the scripting language A-Frame, JavaScript, and HTML to design, render, and deploy the model for public use and enjoyment. This is a summary of the project for design and deployment in relation to a school project, and really only serves as a summary of the process as well as an overview of the research involved.