I. Abstract

As part of its mission to preserve the past and safeguard access to it today and long into the future, the Yale Institute for the Preservation of Cultural Heritage (YPCH) is partnering with ICOMOS (International Council on Monuments and Sites) and CyArk (A prototype project to test the concept of a digital archive of 3D survey information of endangered world heritage sites) by working in the Project Anqa initiative.

Project Anqa is an emergency program for recording high risk physical cultural heritage, such as historic buildings and artifacts in the Middle East and North Africa regions. Named after the Arabic word for phoenix, Project Anqa deploys teams of international professionals and pairs
them with local experts to digitally document at-risk sites into 3D models before they are destroyed or altered.

Project Anqa aims to make the gathered data available online through the web for both cultural heritage scholars and the general public to benefit from this highly 3D detailed models.

This senior project aims to build an interactive platform through which users can best engage and learn from 3D Cultural Heritage models delivered by Project Anqa via web. For this purpose, various WebGL frameworks such as threeJS, blend4web and A-frame as well as emerging technologies such as VR and the upcoming WebVR standard were tested and applied to create a prototype of the final product. Additionally, a description of the user stories for the 3D Model Interaction Platform, explaining the end user experience and the user’s interaction with the contents of the website is provided.

II. Introduction

One of the core objectives of Project Anqa is to deliver 3D data of cultural heritage sites over the internet through a website. Currently, there are a variety of platforms that manipulate and deliver 3D content over the web; some of the platforms considered were Sketchfab, Thingiverse and their competitors. Nevertheless, none of these existing solutions are a good fit for Project Anqa.

Sketchfab is at the moment the most popular platform for sharing and publishing 3D content, with other cultural heritage projects such as CyArk utilizing its software and maintaining a presence on the site making it the prime candidate solution for Anqa’s site. However, its current
functionality is limited to simple HTML annotations overlapping the 3D models and restricts the ways in which the user can transform and save displayed models. It also requires monthly fees to be paid in order to upload any model bigger than 50MB in size and in order to be able to place more than 5 annotations per model and use all of the viewer’s customization features.

Thingiverse, another platform for 3D content, provides access to applications within its website to allow users to edit their models. However, the platform is geared towards 3D printing and lacks a unified 3D viewing experience, instead relying in disjointed apps for each different type of object to be printed.

Other competitors such as Flux.io and Cl3ver are geared towards more specialized users such as engineers and architects and charge greater fees for their services.

Even if there existed a third-party solution for Project Anqa 3D content delivery, the young and volatile nature of the 3D web content space challenges the longevity of any third-party solution. As an example, Goo Technologies, one of Sketchfab’s top competitors recently discontinued its service, forcing its users to migrate to other services.

In addition to the avoiding the drawbacks of using a third-party service, developing an in-house 3D content platform for Anqa’s website provides many benefits. Project Anqa will retain control of the 3D content distribution, will be able to tailor model transformation specifically to the needs of the general public and cultural heritage professionals.
For general audiences, Anqa aims to make available an experience where curated information is shown to the user via static elements such as annotations, textures, visual cues (arrows, markings) as well as dynamic such as animated transitions.

For cultural heritage experts, an enhanced platform will allow for the creation of new 3D models and annotations, with the purpose of submitting them for addition to the Project Anqa database and website.

Having an in-house 3D platform will facilitate the construction of the website’s backend for managing the submission of new information to the website.

**III. Frameworks and Technology**

Having decided on building an in-house 3D platform, the next step was to explore different existing technologies and frameworks in order to see if they were fit for Project Anqa’s purposes.

Due to its ubiquity and well documented frameworks, the WebGL standard for browser graphics was the chosen language for building the 3D platform.

The frameworks considered and explored were:

- Blend4web
- A-Frame
- ThreeJS
The target technologies for deployment were traditional 2D rendering of 3D graphics in a browser window and Mobile VR using the Google Cardboard headset.

**WebGL**

Since 2011 WebGL has been the modern standard for displaying 3D graphics in a browser, superseding Flash as the de facto way of delivering graphics over the internet.

WebGL is a JavaScript API consisting of control code written in JavaScript and shader code implemented in the OpenGL Shading Language. WebGL overcomes the complexities of working with the low-level OpenGL specification, while keeping the benefits of running close to the hardware, allowing the browser to directly interface with the GPU.

WebGL uses the HTML5 canvas element, which can be accessed through the traditional document object model (DOM) of a website. All modern desktop browsers are WebGL compatible as well as the most popular mobile browsers such as Safari, Chrome and Firefox. Utilizing WebGL requires no extra installation steps as it comes integrated with the users’ browsers.
**Choosing a Framework: Blend4web vs. A-frame vs. ThreeJS**

*Blend4web*

Blend4web is an open-source framework for developing WebGL-based 3D content in a browser. It relies and leverages heavily in using Blender, the popular open-source 3D content creation tool.

Blend4web allows for easy export of a Blender scene to a JSON (JavaScript Object Notation) and a binary file that are then loaded by the web browser. Its most appealing feature is the ability to export a 3D application packaged in a single HTML. Alternatively, the Blend4web player can be embedded into an existing website using an *iframe* element. It also includes easy to use support for VR experiences, positional audio, physic engines and model annotation.

Reputable organizations with objectives similar to Project Anqa’s have used blend4web to build their own 3D interactive experiences. For an example, refer to NASA’s Experience Curiosity.¹

As part of this project, a demo of a Blen4web viewer was created using a model of the Syrian Nur al-Din Bimaristan site, obtained by Project Anqa. The viewer displays the model at the full resolution of 500,000 faces.

¹ Experience Curiosity - https://eyes.nasa.gov/curiosity/
The content creation process consisted of:

1. Importing the model to Blender
2. Creating empty objects and relating them to the model in a parent-child relationship,
3. Enabling Meta-Tag functionalities
4. Typing the text content
5. Positioning the annotations on the model
6. Exporting to JSON and .bin format.

In spite of its good performance and the rich features for content available with Blender, the framework suffers from the necessity of having to export back to a blender (.blend) file to manipulate and save user input. While this is not a limitation for the purposes of presenting
content to a general audience, developing an information capture interface for cultural heritage experts would have been unnecessarily complicated. Additionally, threeJS and A-frame also offer support for importing blender scenes, trumping the main advantage of using Blend4web.

*ThreeJS*

ThreeJS is another popular framework for using WebGL in browsers maintained by a community of more than 650 contributors. The core ThreeJS library allows to programmatically generate 3D scenes. Extensive contributions from the community constantly increase ThreeJS functionality, allowing for animation, importing OBJ, blender files and Virtual Reality support.

Its transparent implementation allows for easy customization and manipulation of objects. This feature makes it easy to capture user input using JavaScript and store it.

From the experiences obtained working with this framework on this project’s prototypes, the final platform for Project Anqa will ultimately be built utilizing ThreeJS and its modular libraries.

*A-frame*

A-frame is a new open-source framework for WebGL specifically purposed to build Virtual Reality Experiences. It is being developed by Mozilla in an effort to incentivize the adoption of Virtual Reality by making it easier to build VR experiences and deliver them over the internet.
It uses the provisional JavaScript WebVR standard and is built on top of threeJS.

A-frame is based on an Entity-Component-System (ECS), a software architecture pattern very popular in developing games. ECS puts composition over inheritance and allows for greater flexibility when customizing the 3D scene.

The framework was first released in December of 2015 as v0.1.0 and is constantly being updated each week, with the most recent stable release being v0.5.0, which is the one used in the demos included in this project.

The A-frame framework could be a viable option for implementing interactive experiences with the Project Anqa models, however it poses the problem of learning the ECS paradigm, coding scenes in A-frame language, an unusual mixture of HTML5 tags and JavaScript Scripts. This overcomplicates the development of the main 2D application, making it hard to create 3D Scenes directly using A-frame. Since VR support is the main feature of the framework, a better option is to develop an application using ThreeJS and then connect it to an A-frame shell.

Two A-frame demos exploring its applicability to Project Anqa are included in this project.
IV. Prototype Implementations

For this project, a website hosting the demos and prototypes developed through the semester was built. The technologies used were expressJS server logic, Handlebars for rendering, Git for version control, Bitbucket as a repository and Heroku for hosting. The website can be accessed through the referenced link².  

² http://rem62-cpsc-490.herokuapp.com/
The first was implemented in A-frame with the purpose of learning the basics of the language and envisioning how the final product could be enhanced by incorporating a Virtual Reality component.

The demo consisted of a single scene, with two loaded .OBJ models, one with its corresponding MTL material file, a rendered text element, a textured plane and a cube that responds to user clicks.

The demo was tested using the Google Cardboard VR headset on an iOS mobile Chrome browser. Even though WebVR is not yet fully compatible with iOS, the VR mode of this demo fully works since it is based on WebGL standards. This demo can be experienced through the referenced link\(^3\) in 2D from any browser, using the WASD or ARROW controls to navigate in First Person mode, and in VR by using a mobile phone in conjunction with a VR headset, or through visiting the prototype’s website using a dedicated VR console such as the Vive or Oculus.

\(^3\) http://rem62-cpsc-490.herokuapp.com/demo1
Similarly, demo 2 implements a different scene in A-frame experimented with different sky and floor textures, and can be accessed through the referenced link⁴.

Next, demos 3, 4 and 5 were implemented using threeJS with the purpose of learning the framework and testing its reliability in loading high resolution models and the functionality of the VR libraries available for threeJS.

Demos 4 and 5 load the full resolution models of the Nur al-Din Bimaristan site and respond to the user cursor events by changing the emissive material of the object.

⁴ http://rem62-cpsc-490.herokuapp.com/demo2
A prototype of the interface and functionalities that will be made available to cultural heritage experts to contribute information to the project is implemented in two modalities. The first one is exclusively for 2D use and has more features such as importing JSON and OBJ files, as well as projecting decals into object surfaces. The second one is the full screen version of the interface with VR functionality enabled. A transform helper was experimented with when doing this implementation. The final product will combine the best features from these examples.

Finally, a Virtual Reality environment was implemented using 3 models provided by Project Anqa from the Syria site. The VR scene implements a sky with controllers to adjust the scene’s illumination. It also experiments with “porting” spheres that allows Google Cardboard users to move around the scene. From earlier demos, it incorporates the ability to respond to user actions, however for speed, the models first underwent vertex collapsing to a lower resolution.
2D Editing Interface implementing different colored and sized decals, exporting and importing capabilities.

Full Screen / VR enabled Editing Interface implementing the transform helper for translations, rotations and scaling.
An Orbit view of the implemented VR scene, note the 2 models in gold and silver, the purple port spheres and the adjustable sky and luminance.
A first-person view of the scene with the gaze-controlled cursor in action, causing the text rendering to become visible.

V. Discussion of Intended User Functionality

Having implemented the aforementioned prototypes and tested their capabilities and limits, the next step is to build a platform that fulfills the needs of Project Anqa.

There are three main features that the final platform must fulfill:
1. Effective visualization of cultural heritage information

2. Production and capture of new information by cultural heritage experts using the site

3. Exporting of original and marked models.

For effective visualization, a non-full screen visualizer window with a predefined set of options containing curated information will be the most effective. This can be implemented either with threeJS or blender4web. A VR experience that automatically guides the user through space will be especially relevant for big models such as those of the entire site, as this will allow users to contextualize the cultural heritage artifacts and appreciate the site as a whole rather than through disjointed models.

The production and capture of new information will be implemented in an editor-type interface with threeJS, as the framework has proven robust enough to handle the task. The backend handle the JSON objects produced by the framework and will allow the user to save its work for future modification. Experimenting with more sophisticated VR platforms such as Oculus, Vive or Daydream and incorporating the use of controllers is something to be explored.

Finally, for exporting the models, currently there exist exporters for OBJ and JSON formats but not for the widely used Collada format. Implementing such an exporter will enhance the cross-compatibility of the final product.
VI. Acknowledgements

- Prof. Holly Rushmeier for her guidance and support in undertaking this project and her engaging teaching of computer graphics.
- Saima Akhtar, for coordinating Project Anqa.
- Goze Akoglu, for her valuable end user input.

VII. References


