Web-based Virtual Reality with A-Frame

Evaluating the Development Effort

Solange Gomes Santos
CISUC - Department of Informatics Engineering,
University of Coimbra, Coimbra, Portugal
solangegs@student.dei.uc.pt

Jorge C. S. Cardoso
CISUC - Department of Informatics Engineering,
University of Coimbra, Coimbra, Portugal
jorgecardoso@ieee.org

Abstract — **Virtual Reality has become an accessible technology but tools to create Virtual Reality experiences are only now becoming accessible to the general public. This work evaluates the development effort of creating a virtual tour based on 360º photos using the web-based VR framework A-Frame. To evaluate the framework we implemented a virtual tour with common interactive features and analysed the development process. We found that A-Frame can be a viable framework to develop VR experiences but there are still aspects that could be improved to reduce the required development effort.**

Keywords – virtual reality; a-frame; web programming; development.

I. Introduction

Virtual Reality (VR) is increasingly known and present in our daily lives. In recent years, this technology has been refined, and it has quickly become possible and easy to enjoy VR in the comfort of our own house using accessible equipment. Likewise, tools to create VR experiences have been created and have become increasingly accessible. There are already several tools that allow the creation of VR experiences such as Unity 3D [1], Unreal Engine [2] and React 360 [3]. These are, however, primarily targeted at knowledgeable programmers. In this project, we are interested in evaluating a web-based VR development framework named A-Frame [4].

A-Frame is an open-source web framework for building VR experiences based on HTML and JavaScript, originally developed within the Mozilla VR team. HTML is an easy to read and understand mark-up language, often learned by digital/multimedia designers, making A-Frame an accessible framework to non-programmers. Additionally, A-Frame is multi-platform, supporting VR headsets such as HTC Vive, Oculus Rift, Windows Mixed Reality, Daydream, GearVR and Google Cardboard.

The purpose of this work is to evaluate the capabilities of A-Frame and the required effort to create a 360º interactive virtual tour by a multimedia designer – someone knowledgeable in HTML, CSS, and introductory experience in JavaScript. We consider the use of 360º images as they represent an often-used solution for virtual tours (e.g., for museums, cities, real estate) and because 360º-based VR requires 3D modelling, which can be a very time-consuming task.

To serve as a case study, we developed a virtual tour for the Conimbriga Monographic Museum. Located in Condeixa-a-Nova, Portugal, the museum has a collection composed exclusively of archaeological material found in the ruins of the Roman city of Conimbriga. The current exhibition displays everyday objects organized by theme. Six of these objects have been classified as of national interest. The virtual tour highlights these objects of interest and allows users to interact with them to see additional details.

II. A-Frame

An A-Frame VR scene is defined in the body of the HTML document with an <a-scene> element, which represents the 3D scene's root and will contain all the necessary 3D objects in the scene. Simple 3D objects are defined with A-Frame's primitives; for example, to create a sphere, the <a-sphere> element can be used:

```
<a-sphere position="3 3 -10" radius="0.7" color="red"></a-sphere>
```

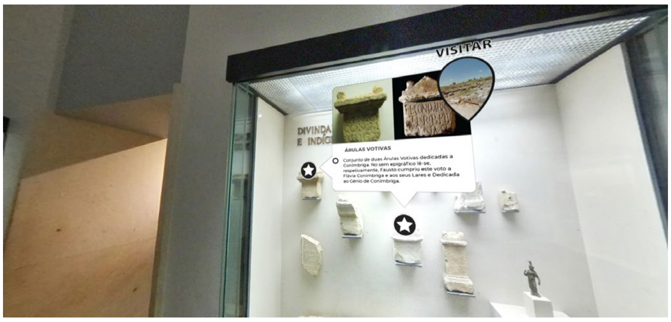
The addition of other elements such as lights, text, portals to other locations, is equally straightforward, as it follows the same logic. A-Frame emulates traditional HTML events such as "click", "mouseenter", "mouseleave" (on a mobile device, these are triggered by positioning a cursor shown at the centre of screen over an object in the scene). These events can be listened to and result in changes in the scene – for example, to change the colour of an object, to make it visible, to change its position, etc. Listening to events requires the use of an external component, which can be added by including a JavaScript file after the inclusion of A-Frame's JavaScript file.

A-Frame's functionality can be extended by creating external components which are executed during the render loop and can be used to create custom elements and behaviours according to the current state of the scene. External components are programmed using JavaScript.

III. Approach

The approach used in this work was to analyse the learning and development process of a multimedia designer (the first author) with no previous experience with A-Frame or other VR development tools. This process was documented with consideration to the encountered difficulties, shortcomings and current flaws in A-Frame. The following steps were taken:

1. Learn A-Frame. Learning A-Frame through an existing online course and through the official A-Frame's documentation.
2. List virtual tour elements. Analysing existing virtual tours based on 360º photos to uncover the



ÁRULAS VOTIVAS

Conjunto de duas Árulas Votivas dedicadas a Conimbriga. No sem epigráfico lê-se, respetivamente, Fausto cumpriu este voto a Flávia Conimbriga e aos seus Lares e Dedicada ao Génio de Conímbriga.

Figure 1. Popup panel with information about two of the exhibited pieces.

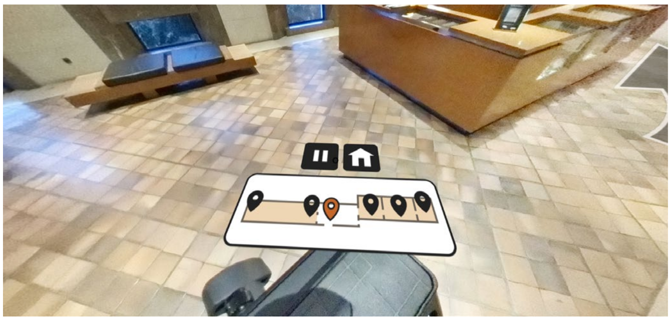


Figure 2. Bottom menu for jumping across scenes.

main elements that would be necessary to implement.

3. Test implementation. Implementing each main element separately and document the difficulties encountered.
4. Integrate. Integrating all the implemented elements in a virtual tour.

IV. Results

A prototype of the implemented virtual tour can be found at: https://conimbriga360.dei.uc.pt.

The following issues were found while going through the development process.

A. *Positioning Elements*

Positioning 3D objects in a scene in relation to objects in a 360º photo is a trial and error process. Although A-Frame provides a visual inspector that can facilitate this task, its current version lacks flexibility making it cumbersome to adjust the 3D objects. In addition, it is necessary to go back to the HTML file to correct the values for the 3D position, rotation, and scaling.

B. *Visually Complex Elements*

Although A-Frame is based on HTML, it does not support any form of CSS. All styling must be accomplished by composing simpler elements, often requiring image editing for simple effects – an example is creating an information popup panel with rounded corners (Fig 1). This is very different from the typical styling process of an HTML document through CSS.

C. *Complex Interactions and Animation*

Although there are various external components for animations and interactions, many situations still require creating custom components and, hence, JavaScript programming. An example of this is the bottom navigation menu that appear when users look down. The menu cannot be statically positioned because it must appear in the direction the user is facing, however, after it appears it should be static so that users can select a location (Fig. 2). This behaviour required developing a custom A-Frame component.

D. *Browser Compatibility*

A-Frame tries to be multi-platform and compatible with most browsers. However, not all features are equally well supported on all browsers. Although not the only one, the most salient issue is the fact that not all browsers support automatically entering VR mode when users traverse a portal (a link to another A-Frame scene in another HTML file). This causes a break in the VR experience requiring users to press a button on the screen to re-enter VR.

E. *Documentation*

A-Frame is still a young framework, leveraging novel standards such as WebVR. Although it already provides extensive documentation, there are still areas where documentation could be improved.

V. Conclusions

Virtual Reality is entering a stage where not only it has become accessible for consumption by the general public, but also where it is now accessible as a contents platform where almost anyone can create and distribute VR experiences.

In this work, we have evaluated the development effort of creating a virtual tour based on 360º photos with the A-Frame web-based VR framework.

Results have shown that a multimedia designer with experience in web programming (HTML, CSS, JavaScript) is able to create a VR experience using A-Frame. There are, however, aspects that result in greater than desired development effort. The lack of a CSS-like language for styling, a mature visual editor for positioning 3D objects, and browser support are the major shortcomings identified.

Acknowledgment

This work was financed by national funds through FCT – Fundação para a Ciência e a Tecnologia, I.P., within the project CISUC - UID/CEC/00326/2019. We thank Dr. José Ruivo and Dr. Virgílio Correia from the Conimbriga Monographic Museum for facilitating the access and for providing the necessary information about the pieces of national interest.

References

[1] Unity Technologies. (2019). Unity 3D. Retrieved March 26, 2019, from https://unity3d.com/

[2] Epic Games. (2019). Unreal Engine. Retrieved March 22, 2019, from https://www.unrealengine.com/en-US/what-is-unreal-engine-4

[3] Facebook Inc. (2019). React 360. Retrieved March 22, 2019, from https://facebook.github.io/react-360/

[4] A-Frame. (2019). A-Frame. Retrieved March 22, 2019, from https://aframe.io/