What is/are your name(s)? What assignment group are you?

Neha Yelgireddy and Aaron Lin. A5 Group 28

Which framework did you select (Fitbit, Alexa, A-Frame)?

We selected A-Frame.

Q1: How, if at all, does this framework support package and library management?

A-frame doesn’t specifically have any built-in package or library management. According to the A-frame documentation, we can see how A-frame supports certain CDN imports and how we can use npm to install more dependencies. An example of a library is how we use spacetime.js to handle the time zones in this project.

Q2: How, if at all, does this framework support principles for code separation, like Model-View-Controller? Would the separation principles effectively support creation of a larger application? Why or why not?

A-frame doesn’t specifically support Model-View-Controller, but it does support principles for code separation in other ways. For example, A-frame really utilizes its HTML components to modulate the logic, UI, and data. In this project, we can see how the HTML handles the UI elements while the clock.js file handles all the logic regarding converting the time. In our opinion, we do not think the separation principles would effectively support the creation of a larger application because there are many limitations. With larger applications, managing the data as well as the interactivity can become too complex and inefficient.

Q3: In lecture, we discussed a few design recommendations for each respective device. How, if at all, does the framework support the recommendations for that device? What recommendations are left to the developer to decide how or whether to implement?

A-frame aligns with the design recommendations for larger displays rather than small, wearable devices. A-frame is primarily used for building virtual reality experiences as it supports event listeners and can also be used on TVs and VR headsets. Because larger displays and VR experiences need UI elements to be perceived from farther away, A-frame needs to support text and UI elements so that they can be seen. Some recommendations that are left up to the developer to decide whether to implement is how accessible they want to make the interactions. This because A-frame does not support any conventional accessibility features so it is up to the developer to decide how accessible they want to make it.

Q4: What appear to be benefits to using web technologies for development on this device? Conversely, what advantages might a native framework have over the framework you used?

The benefits to using web technologies for development are that it can be hosted on almost all platforms through browsers. This means that you don’t really have to download anything as it will all get updated immediately. Additionally, A-frame specifically uses HTML and JavaScript which is simpler to use to build with. An advantage of a native framework is that it supports better performance as they can fully use CPUs to handle the various tasks and they also have more optimized memory management.

Q5: What did you find easy and challenging about development in this framework?

When working with this framework, we found it very simple to use HTML to create simple but effective VR interactions. Also, because it runs in the browser, it was very easy to debug because we could see the instant updates in the browser. Something that was challenging was working with Cordova and being able to get it on our mobile devices. For example, the app would sometimes get stuck on the loading screen when trying to launch on the phone. We integrated Cordova specifically on Aaron’s MacBook and iPhone, to which we also found several challenges. For example, to support Cordova and iOS support, a few things had to be installed, such as CocoaPods via Ruby (gem), XCode on macOS, and the latest macOS/iOS update. Additionally, we found that Aaron’s MacBook had several outdated softwares needed to be upgraded in order to continue with Cordova development, and that XCode actually required an Apple Developer Account and Signing/Certification, which made things more complex. After finally dealing with Apple Dev Certification and working with dev mode in iOS, we found that our XCode build settings were also outdated. This provided further challenge as we had to do more research on Apple’s Development Guidelines, however, we were able to get to the adjust our build settings to one just right for developing on Aaron’s iPhone. After getting through those obstacles, we finally saw our Cordova App successfully deployed and working on Aaron’s iPhone!