Alexander Kyu
Programmable User Interfaces

FP4 Writeup

Part 1:

My website and AR App, WebdARw, allows users to draw in three dimensional space and upload it to the web application side to view! This can be as simple as a three dimensional drawing, two dimensional drawing on a detected plane (not yet implemented at the time of submission), or capturing point clouds of the surrounding area (also not yet implemented at the time of submission).

The web application allows users to view any uploaded model, download it, or share it via sending a link to anyone! This sharing of doodles allows for creative sharing of ideas, thoughts, or just creates fun moments for people! The target audience is anyone with an iPhone for the AR App component (because I didn't make the app compatible with Android yet and didn't have time to make it compatible unfortunately) and anyone with access to the web for the web application component.

Part 2:

AR App:

- Change color to draw with. This is just a button to open the color option and then select the button.
- Change the gradient with a secondary color chosen. This is similar to the primary color selection. This gradient unfortunately doesn't get saved to the cloud (technical issues implementing this).
- Change size of the drawing tool. This is just a slider.
- Clear and Undo buttons allow you to either completely clear your drawing or to just undo the last action you did.
- Upload the drawing by typing in the top selection box and hitting the upload button next to it.

Web App:

- Search for the specific drawings using the search bar above.
- Open and view the drawing by clicking on the first button on any drawing "card."
- Download any drawings in an *.fbx file using the second button on any drawing "card."
- Copy the link of any drawings as a url by clicking the third button

Part 3:

AR App:

- Unity - This has a lot of support for developing AR apps already
- Firebase SDK support for Unity Applications - This allows for use of the Google Firebase realtime DB and storage that I use for uploading.

Web Application:

- MUI - This has a lot of UI elements to quickly implement the UI and inputs of the web application.
- Firebase - Again, this is used to download and stream information from the Google Firebase I'm using.
- Three JS - I use this to display the model in three dimensions in the web application.
- React Router DOM - I used this to change the search params so that people can share drawings with each other!

Part 4:

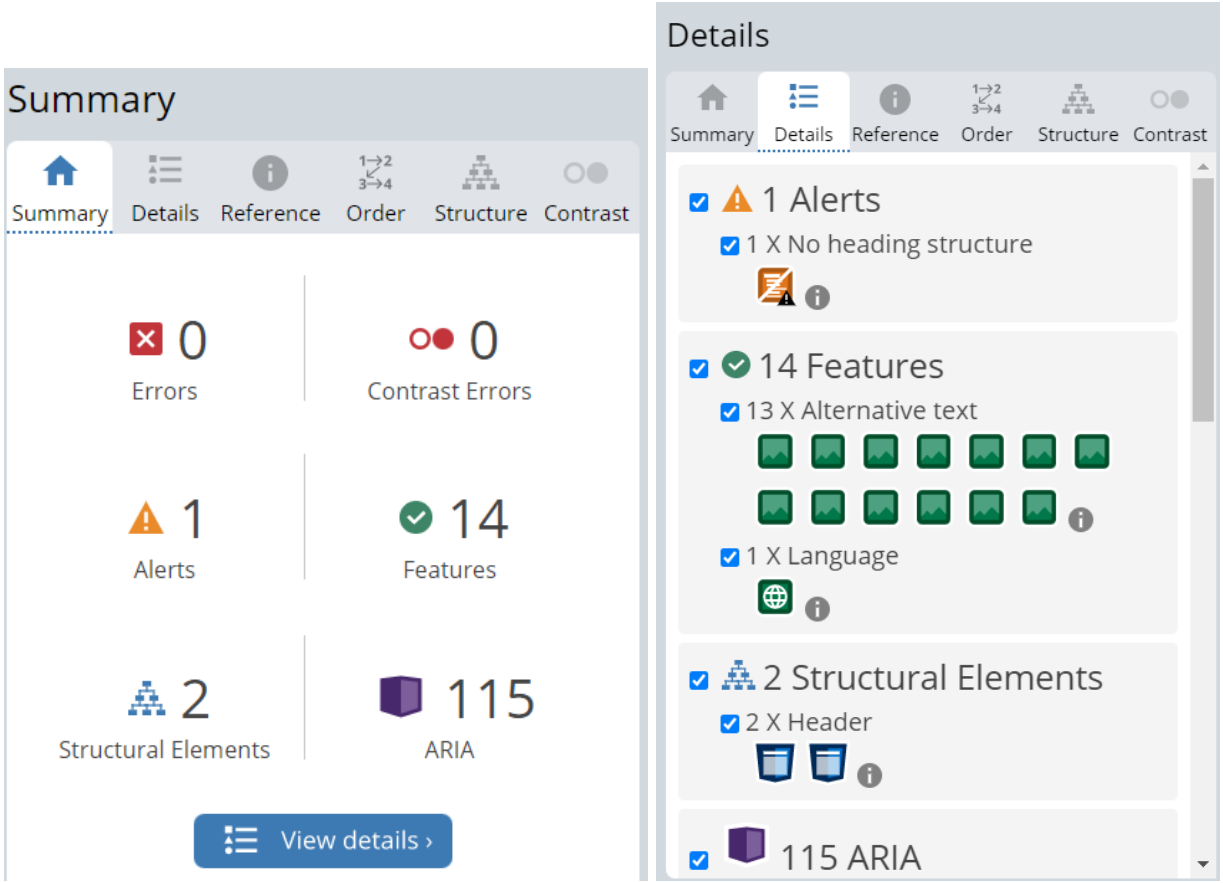
I initially started this project wanting to just get into AR applications. I first started with a waypoint application, but wasn't compelled by its use case. I pivoted to this drawing application. While I didn't have many iterations due to the complexity of implementation, I did take inspiration from my web application from thingiverse (of course very much simplified).

Part 5:

The major challenge for me was getting all of the components to work together. I initially got the drawing AR component to work, but wasn't able to export it in a three dimensional file format, so I had to convert the Line Renderers to a tubular mesh and then export as an FBX file (which Unity doesn't even have a built in method for this, so I had to find some other way of doing that). Then I had to figure out how to communicate with Google Firebase which I got to work quickly in the Unity Editor, but spent hours trying to build AND connecting on the deployed iOS app.

On the web application side, the hardest part was figuring out how to get the Three JS library to take web urls and show the corresponding FBX file. I had to get a blob version of the file, then make a local url to display it as the method for using the download url didn't work.

WAVE Accessibility



Responsiveness:

The web application should work on both mobile devices and computers! You can try and phone and computer for compatibility.