**Challenges**

**UI**

Challenge 1:

I had never used the UI Toolkit or UI Builder before so I had to familiarize myself with the new way to set up the UI and how to handle the objects in code. I read up on the new UI in Unity Documentation and watched a couple tutorials to get me started. I also made a test project to play around with the UI and practiced manipulating the objects in code.

Challenge 2:

Once I got familiar with the UI I had to decide how I wanted to structure the layout. I reviewed some menu UIs that I had used previously to get ideas on what I liked. After reviewing some UIs and drawing a mockup, I decided the look and layout for the UI that I wanted.

Challenge 3:

I decided to use a ListView to store the list of effects, materials, textures and meshes. This way I would not need to hand set up a lot of different options in the UI Builder and it would be easier to change and add to the options. It took a while to figure out how to use a ListView and how to recycle it so I only needed one. I was able to figure it out by reading the Unity Documentation and testing out how the itemsSource, makeItem and bindItem worked.

**Model Movement**

Challenge 1:

I tried a few different ways of letting the user translate, scale and rotate the object.

My first attempt was having 3 UI buttons where the user could click “Translate”, “Rotate” or “Scale”. Clicking on one of those buttons determined what would happen to the model when the user left-clicked and dragged the model. I didn’t like this implementation because the UI took up too much of the screen space.

My second attempt was to move the movement UI into the main UI that had all of the effect and model options. This fixed the issue of taking up screen space but it still didn’t feel good because the user had to navigate in the menu every time they wanted to switch the movement type.

The solution I settled on was to change from using UI buttons to decide the action of the left-click and instead have left-click Rotate, right-click Scale and middle-click Translate. This freed up the UI and allowed the user to always have access to deciding how they interacted with the model.

**Lighting**

Challenge 1:

It took a while for me to decide how I wanted the user to interact with the lights. I thought about allowing the user to dynamically add lights to the scene but that wouldn't feel good unless the user could also adjust settings and placement for the lights they added. The Technical Test document which outlined the guidelines for the tool mentioned that the UI should be simple and intuitive so I felt having so many controls would go against the tool’s goal. I decided to go with a simplistic solution where I preset a few lights and allowed the user to toggle them on and off. It’s a bit simple but it does showcase how lights will affect the model.

**Post Processing**

Challenge 1:

I had a similar issue with Post Processing as I did with Lighting. I originally was going to enable a bunch of different effects and allow the user to toggle them and adjust their primary settings. This solution also had the issue of complicating the UI. Instead, I created a handful of effects that I liked and allowed the user to choose between those preset effects. This showcases how Post Processing can affect the screen and displays some use cases for the feature.

**Making an iOS Build**

Challenge 1:

I had never made an iOS build before so I wanted to take the opportunity to learn. I watched a few tutorials and was able to generate a build from Unity. However, I ran into a roadblock where I needed a way to run XCode. I tried to figure out a workaround but everything I read said I needed t be on iOS to get a build and test it on an iPhone. Since I didn’t have a way around this, I was not able to make the build.