**General Optimizations during development:**

* **Optimized Models**

While selecting the models and the environment to use, i made sure to have models within a good low poly range. The hero models were not more than 20k polycount limit. The environment had a well lit roof to let a lot of the directional light into the room for Global Illumination to propagate.

* **Optimized Lighting**

The environment was marked static for lighting and it took around 10 minutes to bake the lightmaps for the Scene. The scene consists of 1 directional light and 4 emission bars in the room. The D light was real time whereas the emission bars were baked lights.

* **Functionalities handled by Events**

The functionality of most of what was used is present inside the UIManager class which contains a function called ‘Initialize’ which references each UI Element with their mirror in C# and a pertaining event for the Element. A few external functions were aso created which are referenced within the Initialize function.

**Challenges faced during development :**

* **Runtime Gizmos**

I have used a few assets from the asset store, especially Runtime Gizmos. I tried to build a system by using a cone model for the orientation of the axis and building a custom shader in shader graph for the selected axis. On further analysis, I figured out that it might take much longer to build the Runtime Gizmo, so I built the Transform manipulation through the UI. This definitely worked much faster but I feel it could have been more elegant with the Runtime Gizmo.

* **UI Elements for Runtime**

I have used IMGUI for a very long time and built a few tools for the editor using UI Elements. I have never tried using UI Elements for Runtime. I usually like to take up a challenge, so I built a usable runtime UI Elements interface for this project. The few things that I found to be challenging was to anchor Visual Elements. But after a few trials and errors, I kind of got used to it. Another thing is animating UI Panels. I usually like to tuck in UI panels by moving them off screen when not in use in Unity UI. I would have really liked to do the same for the UI Document.

Another problem I faced was the UI was scaling slightly differently on mobile devices. I'm guessing it's the way I would have placed the elements in their hierarchy. This cased a couple of iterations back and forth.

* **Usage of Scriptable Objects**

I really wanted to use SO for storing the data of the Models and their variants. I needed to quickly put together the objects in the scene so I used a monobehaviour to do everything within the UI Manager. But I feel I could have separated a few things just to make it more streamlined.

* **Post Processing on mobile**

For some reason a few post processing effects were not having the same output as the PC build. After a few iterations back and forth, I decided to remove those effects altogether.