**Getting Started**

At the root of the repository there is a Builds folder which contains a PC build of the tool. You can run 3DMeshVisualizer.exe to test and use the tool.

Also at the root of the repository is the 3DMeshVisualizer folder which is the Unity project folder used when creating the tool. The project was created in Unity 2020.3.5f1.

**How To Use The Tool**

To expand the UI, click the icon in the top left corner of the screen. The UI will allow you to select which assets you would like to apply to the model. It will also allow you to activate different lighting and post processing effects.

You can rotate the model by holding down left-click, scale by holding down right-click and translate by holding down middle-click and then dragging the mouse.

**Project Layout**

**Scene**

The scene to use is the VisualizerScene and contains all of the objects and functionality for the tool.

**Model Movement**

The rotating, scaling and translating of the model is handled by the ModelMovementController located in the Assets/Scripts folder. This is attached to the 3DMeshRenderer GameObject in the scene.

**Effects**

The lighting and post processing effects are set up and controlled by the EffectsMananger located in the Assets/Scripts folder. This is attached to the EffectsManager GameObject in the scene.

**Model Assets**

Being able to switch textures, meshes and materials is handled by the ModelAssetsController located in the Assets/Scripts folder. This is attached to the 3DMeshRenderer GameObject in the scene.

**UI**

The UI was primarily set up using the UI Builder which can be opened in Unity by double clicking on the MainUITemplate.uxml located here: Assets/UI/MainUITemplate.uxml.

The UI code is all handled in the UIController script located in the Assets/Scripts folder. This is attached to the GameObject UIDocument in the scene. The UIController handles dynamically setting up the ListView which holds the effect and asset options available to the user. It also handles registering the appropriate events so when effect/asset options are selected, the correct controller is notified and can handle the input.