

User Documentation

Reference tool

Purpose:

To help lighting artists load the most recent versions of the files needed for a shoot, there is an Import Files feature. References to settings, layouts, props, and character animations from a user-specified directory are supported.

Features:

Asset Referencing: Make it simple to reference a variety of asset kinds, including props, settings, layouts, and character animations.

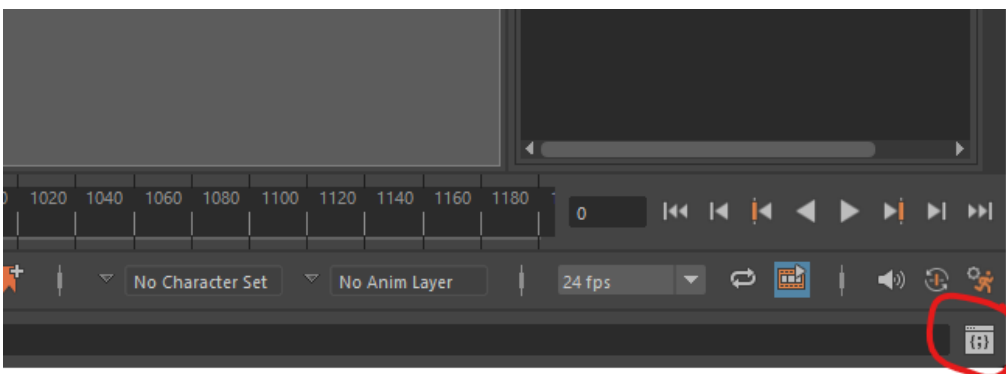
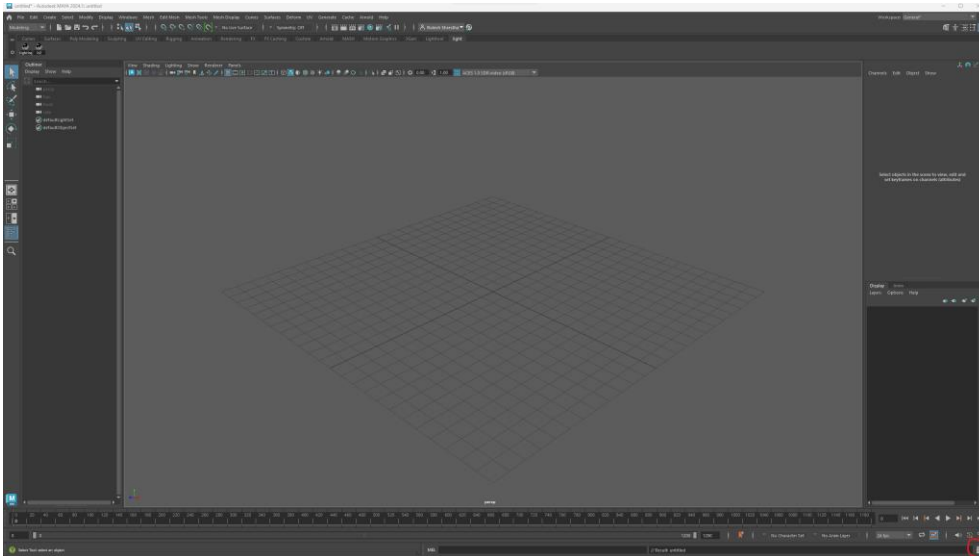
Version checking: Provides the ability to update out-of-date files and automatically looks for the most recent version of assets.

User Interface: An easy-to-use interface for managing and choosing reference materials. How to Use:

Using the Tool:

1. Running the code

1.1. Open Maya and run the script editor.

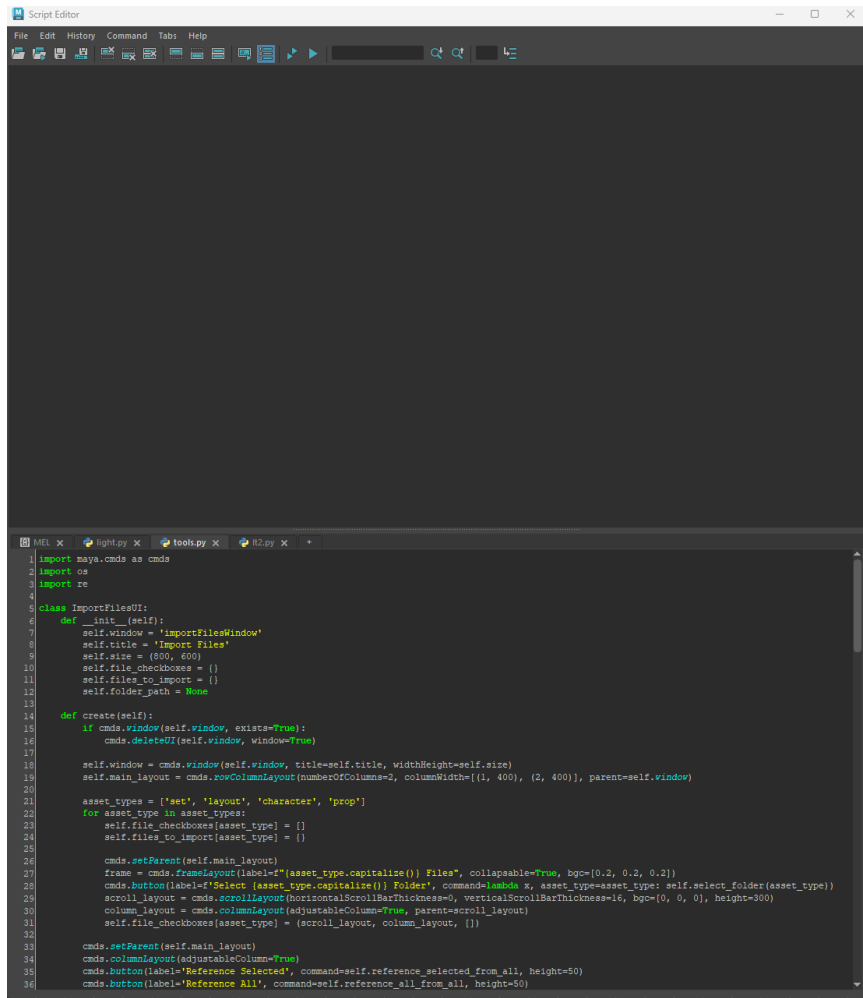


1.2. After you have opened the script editor run the script in Maya's Script Editor to open the Import Files tool.

Reference tool:

https://github.com/EmKinder/TD_Assignment2/blob/LightingTool/Lighting%20tool/Final%20Codes/Reference%20tool

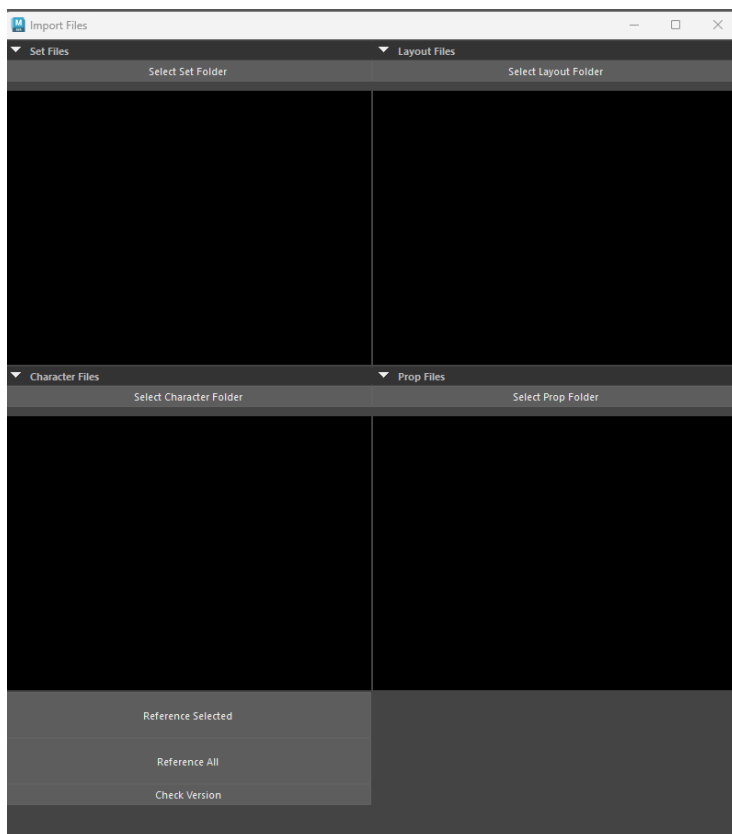
You can find the codes from the link above after you have successfully run the first code.



```
Script Editor
File Edit History Command Tabs Help
[Icons] [Search] [Run] [Stop] [Break] [Help]

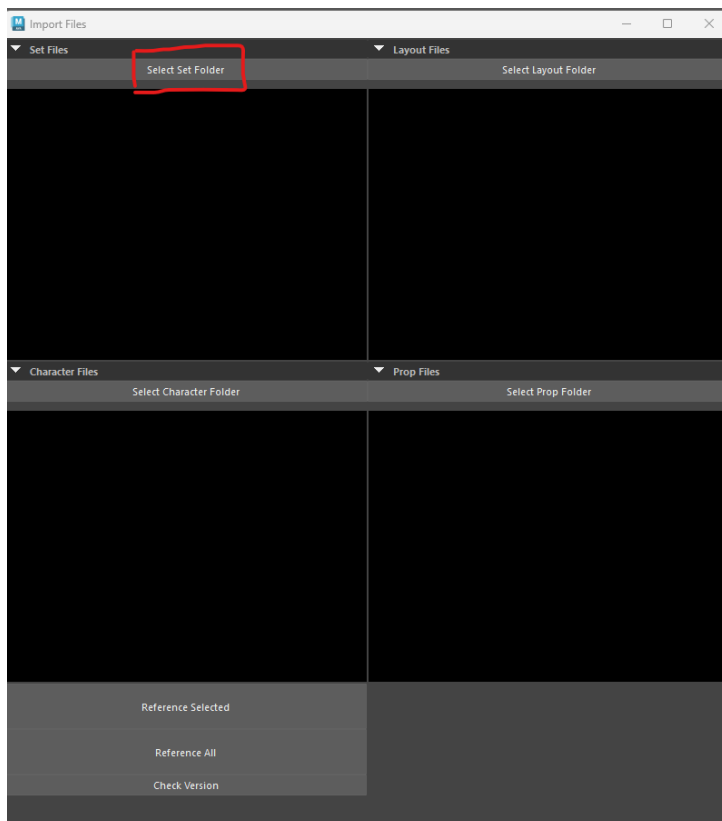
1 import maya.cmds as cmds
2 import os
3 import re
4
5 class ImportFilesUI:
6     def __init__(self):
7         self.window = cmds.window('ImportFilesWindow')
8         self.title = 'Import Files'
9         self.size = (800, 600)
10        self.file_checkboxes = {}
11        self.files_to_import = {}
12        self.folder_path = None
13
14    def create(self):
15        if cmds.window(self.window, exists=True):
16            cmds.deleteUI(self.window, window=True)
17
18        self.window = cmds.window(self.window, title=self.title, widthHeight=self.size)
19        self.main_layout = cmds.rowColumnLayout(numberOfColumns=2, columnWidth=[(1, 400), (2, 400)], parent=self.window)
20
21        asset_types = ['set', 'layout', 'character', 'prop']
22        for asset_type in asset_types:
23            self.file_checkboxes[asset_type] = {}
24            self.files_to_import[asset_type] = {}
25
26            cmds.setParent(self.main_layout)
27            frame = cmds.frameLayout(label=f'{asset_type.capitalize()} Files', collapsable=True, bgc=[0.2, 0.2, 0.2])
28            cmds.button(label=f'Select {asset_type.capitalize()} Folder', command=self.select_folder(asset_type))
29            scroll_layout = cmds.scrollLayout(horizontalScrollBarThickness=0, verticalScrollBarThickness=16, bgc=[0, 0, 0], height=300)
30            column_layout = cmds.columnLayout(adjustableColumn=True, parent=scroll_layout)
31            self.file_checkboxes[asset_type] = (scroll_layout, column_layout, [])
32
33        cmds.setParent(self.main_layout)
34        cmds.columnLayout(adjustableColumn=True)
35        cmds.button(label='Reference Selected', command=self.reference_selected_from_all, height=50)
36        cmds.button(label='Reference All', command=self.reference_all_from_all, height=50)
```

1.3. After the code has been successfully run you should see this tool screen.

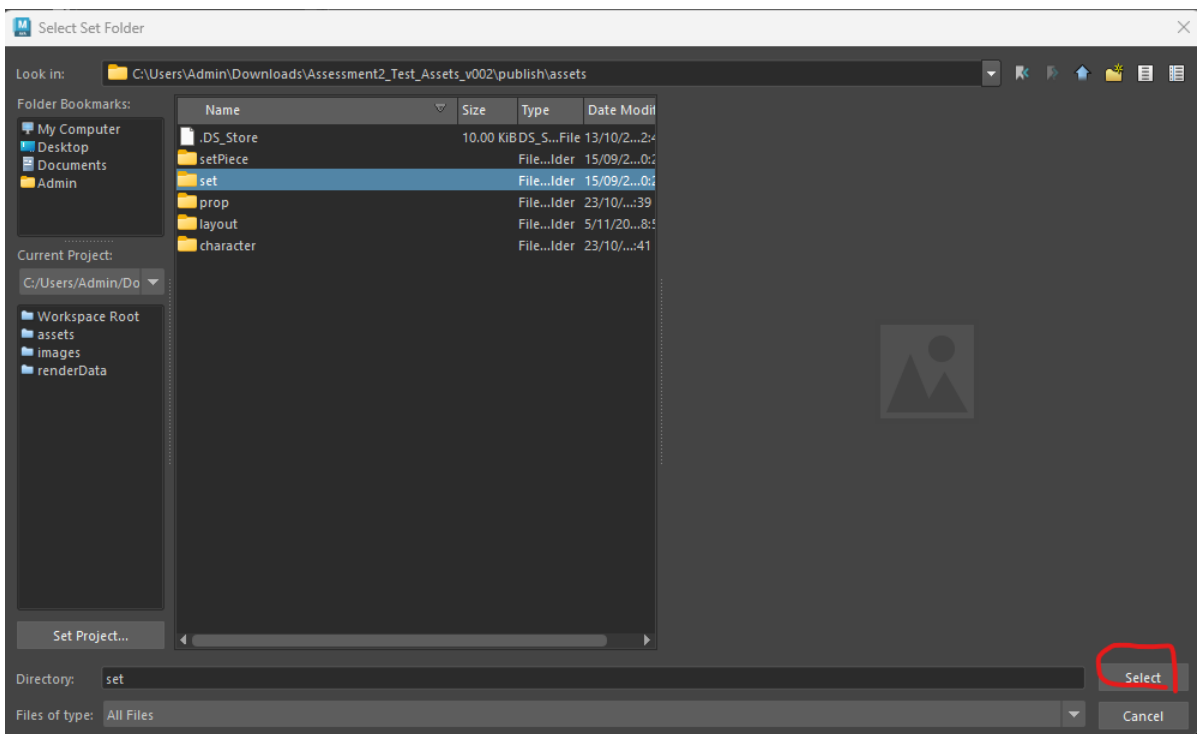


2. Selecting Folders

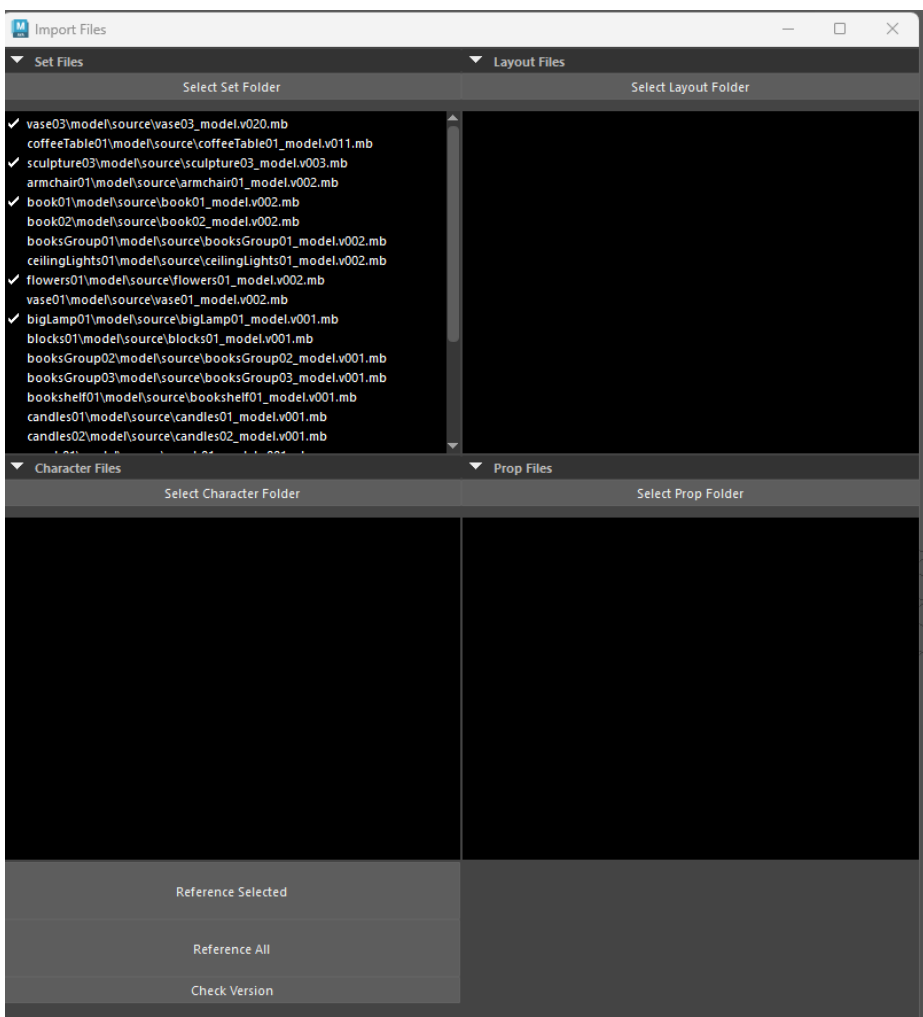
2.1. After the screen has opened, you will need to click on select set folder.



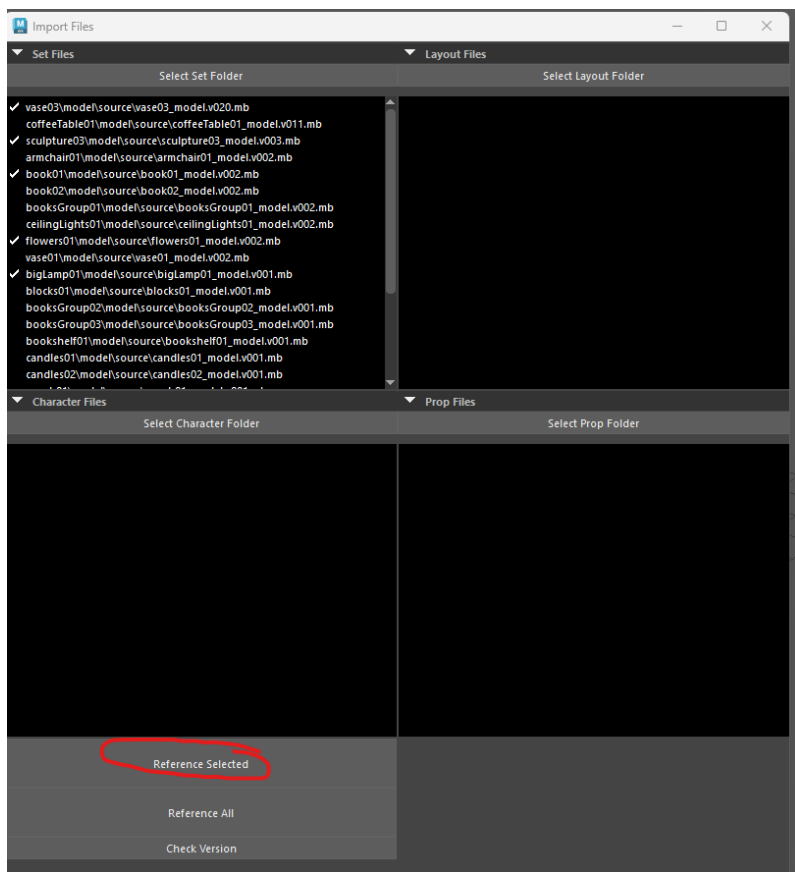
2.2. New window will pop up where you will need to search for your file location and after you have found your set folder press okay.



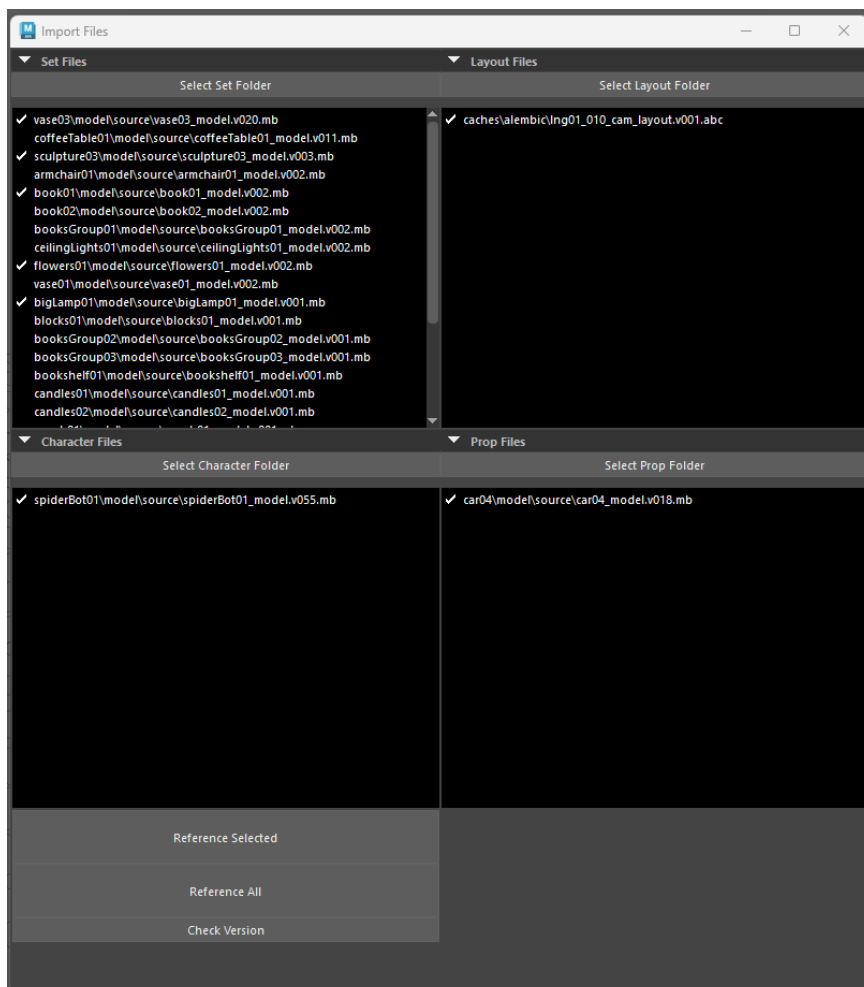
2.3. There you can tick the files you want to import and untick the one you don't want to import.



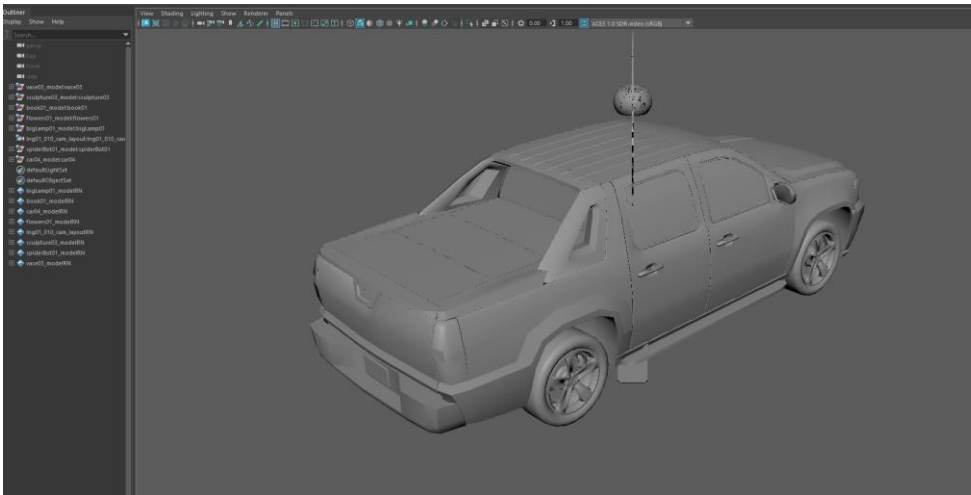
2.4. Click on Reference Selected to reference all the files you have selected.



2.5. Select folders for layout, Character, and Prop it's optional but you can select for all folders and mass select it for every file.

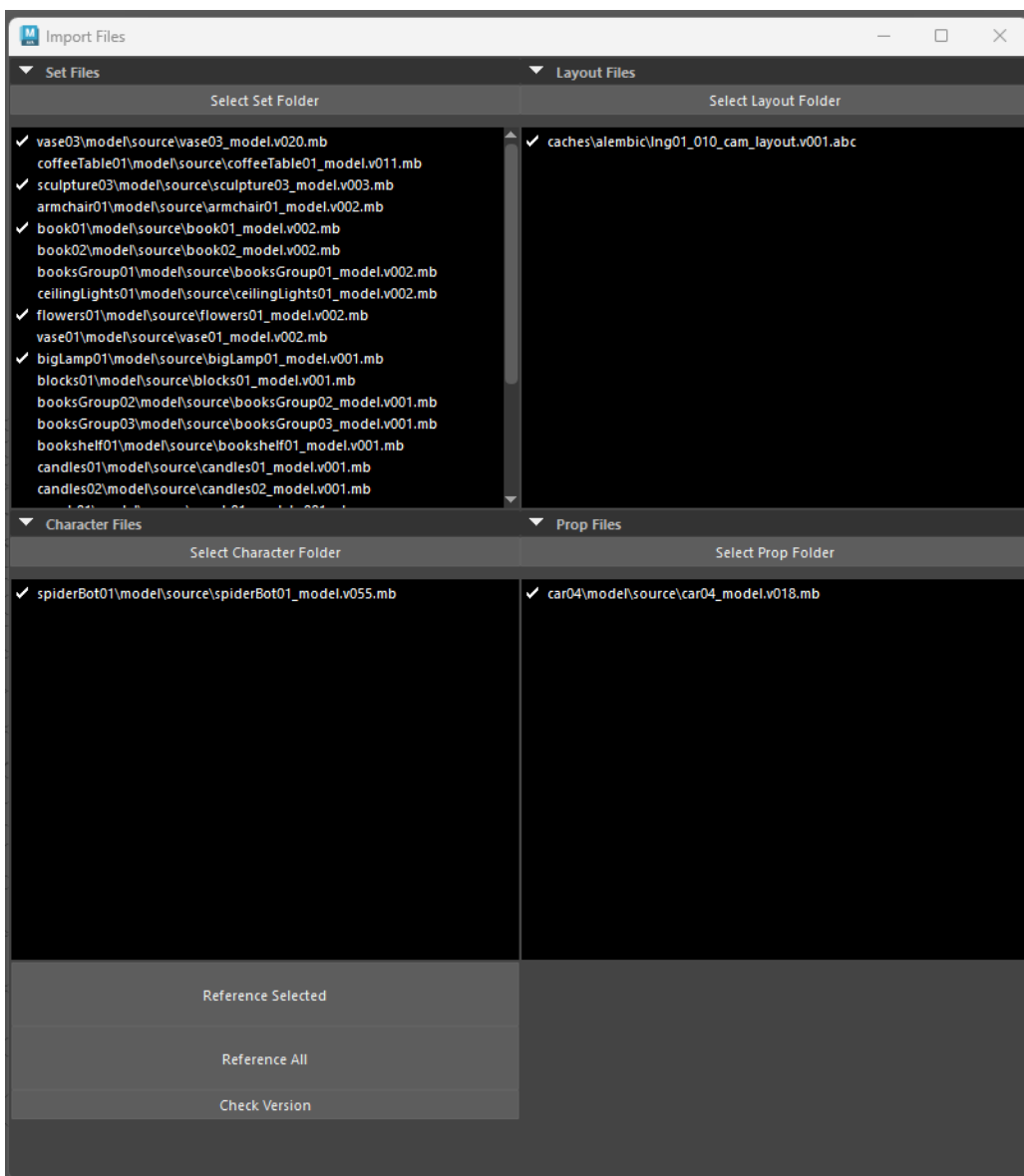


2.6. Now it will bring all the scenes you have selected.

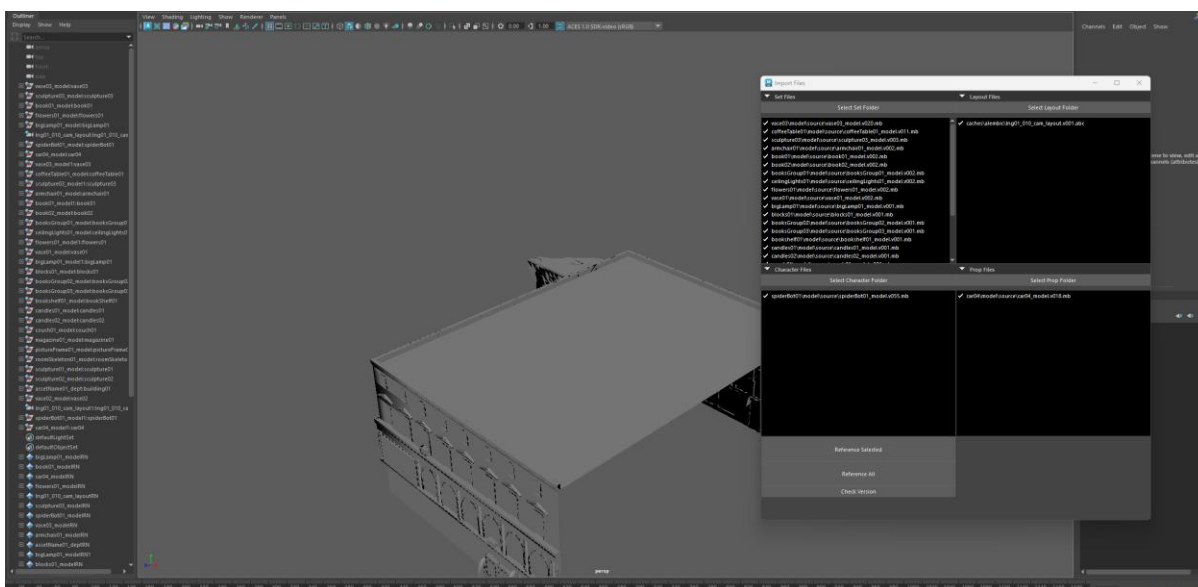
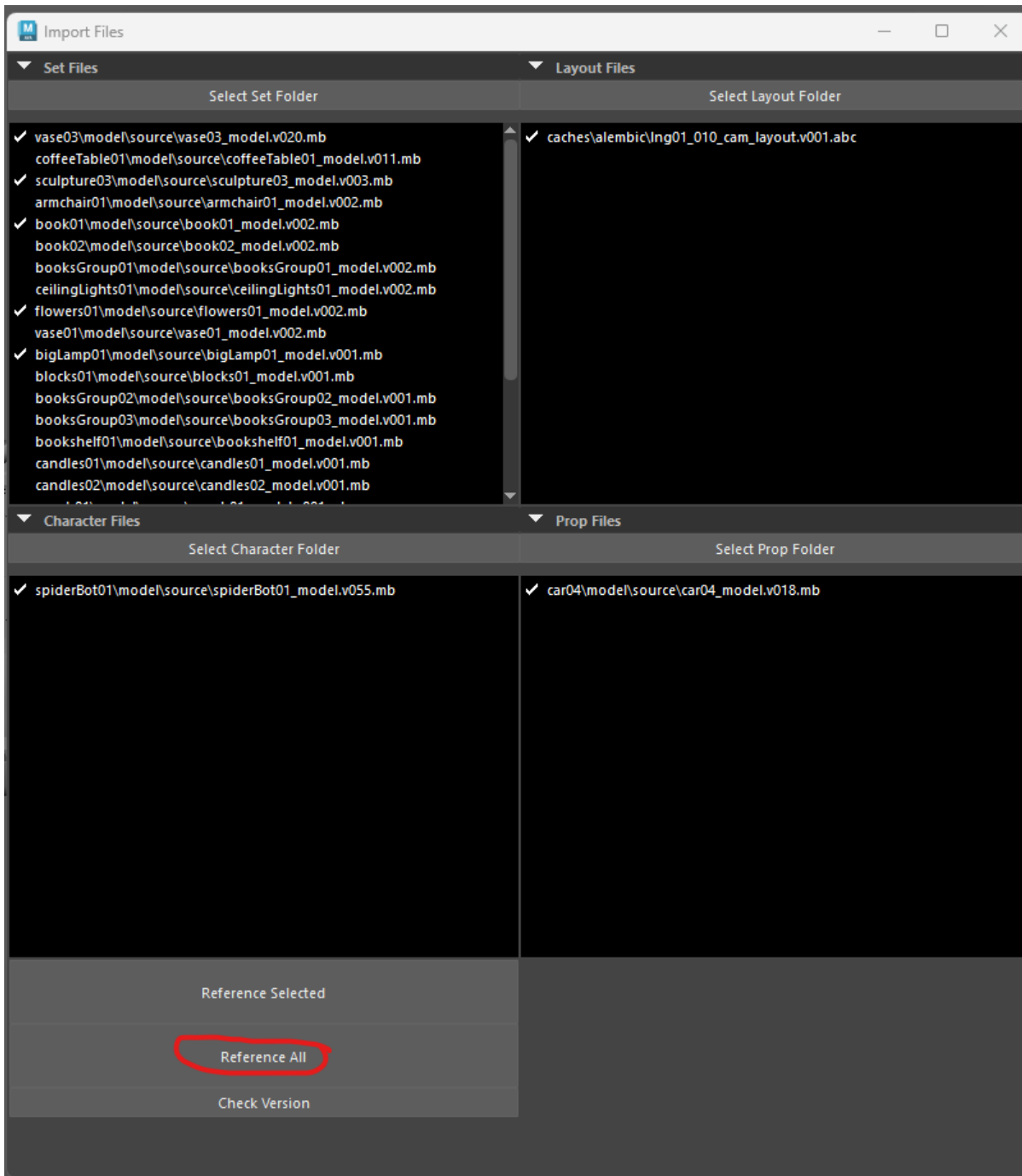


3. Importing everything

3.1. Select all the folders for the project you want to import.

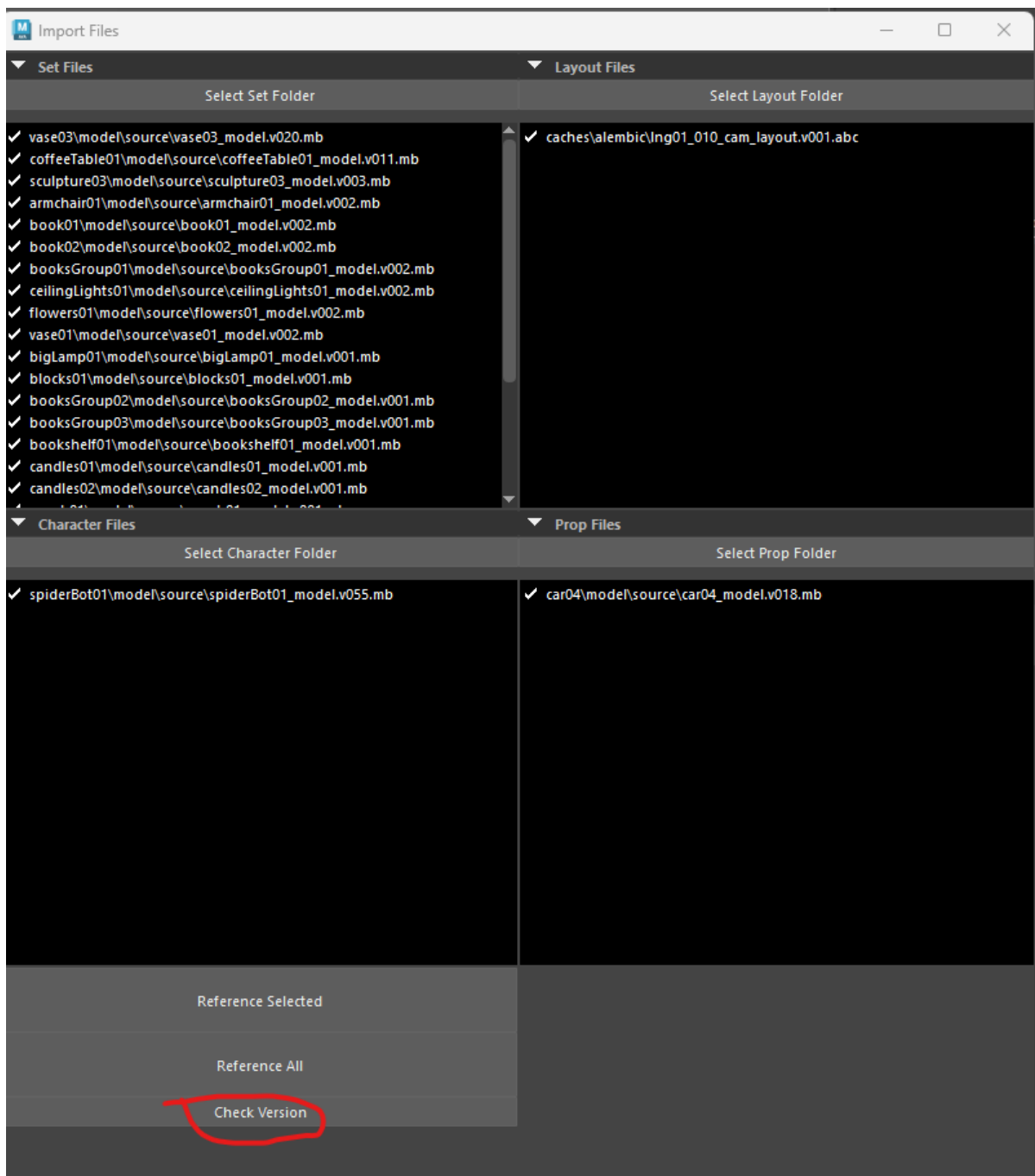


3.2. Then Click on Reference All button which will reference all the file that is available into the scene.

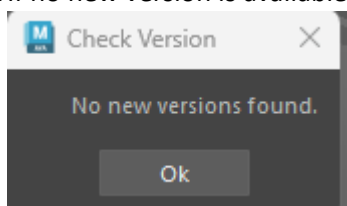


4. Check Version

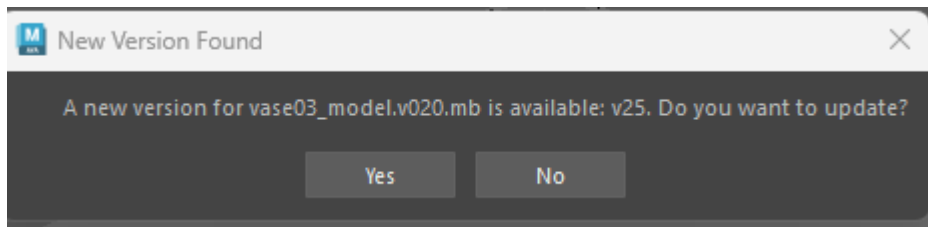
4.1. To check for the latest version, click on the latest version button.



4.2. If no new version is available

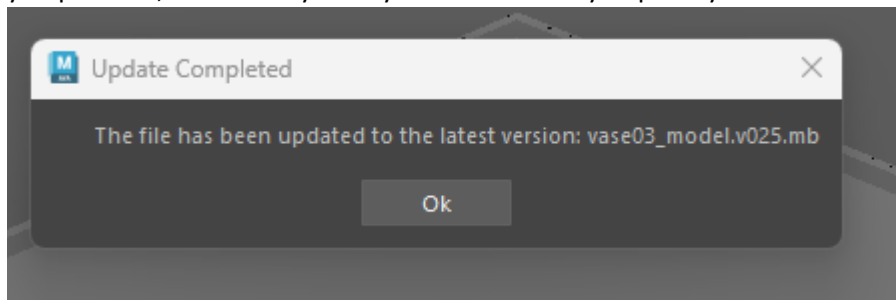


This popup will be shown but if some changes are there.



This popup will be shown which will classify which model has been changed and which is the latest version available.

4.3. If you want to change it to the latest version press yes or if you want to stay in the current one press no. If you press no, it will take you to your scene but if you press yes.



This popup will be shown which means you have updated to the latest version, and you can press ok to return to your scene.

Issue in tool: After you have updated to the latest version the folder will reset sometime so you might have to again select the folder if that happens.

Note: Ensure that your file paths and naming conventions are consistent for the version checking to work correctly. The tool is compatible with both Windows and Unix file paths.

Video Link: <https://youtu.be/RpvhFJSOTxQ>

This video link contains a small demonstration of how I have used the tool.

Lighting Tool for Maya

Purpose:

The Lighting Tool is a packed feature tool made specifically for Maya that makes it easier to create, control, and render lighting in a scene. With its user-friendly interface, artists can effortlessly add different kinds of lights, modify their attributes, control lighting settings, and generate scenarios.

Features:

Light Creation: Add different types of lights to your scene, such as point, directional, spot, and area lights.

Light Management: Delete, group, and rename lights or groups of lights within your scene.

Parameter Controls: Adjust the intensity, colour, position, rotation, and scale of selected lights.

Presets and Templates: Save and load custom lighting presets to streamline your workflow.

Real-Time Preview and Rendering: Open a preview window for real-time lighting feedback and perform quick renders.

Scene Interaction: Highlight lights for better visibility and isolate lights to focus on individual elements.

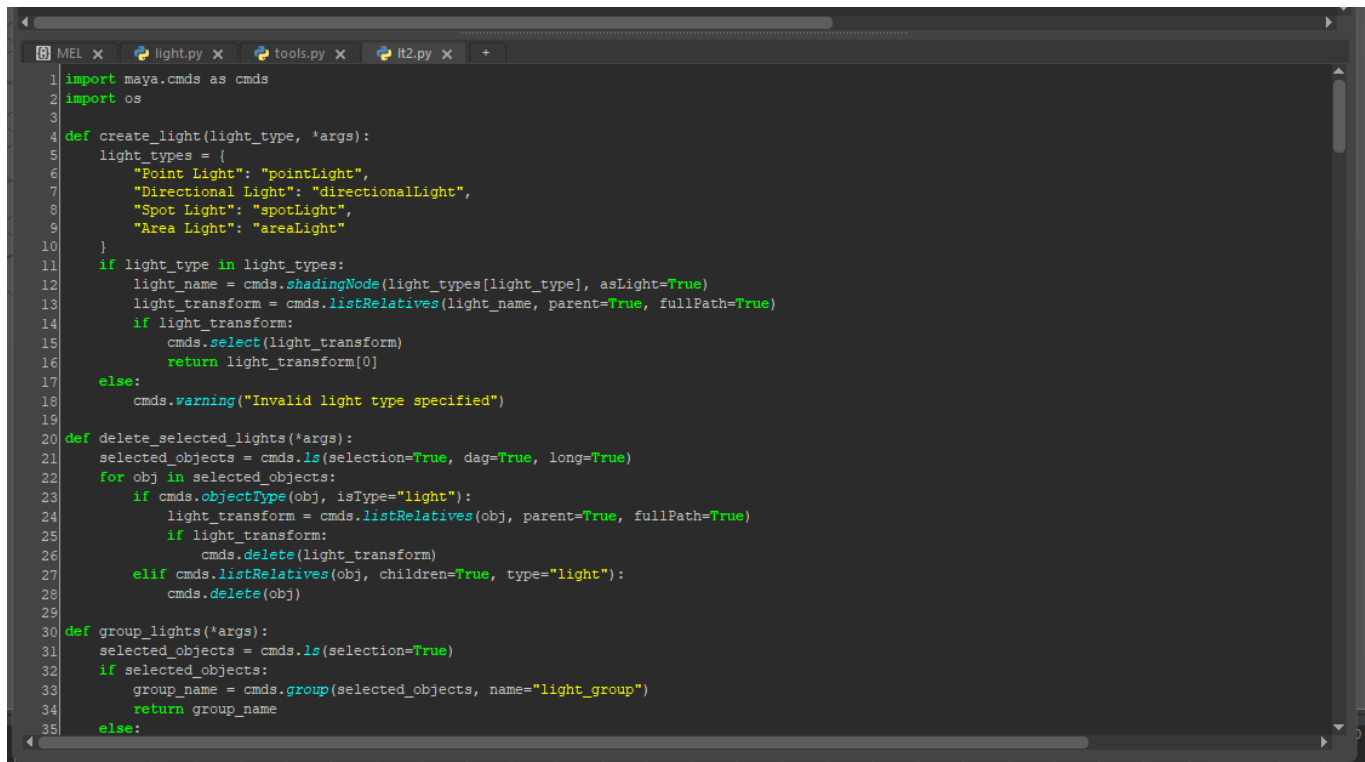
Render Quality Settings: Easily switch between different render quality settings for Arnold renderer.

How to Use:

1. Launch the Tool:

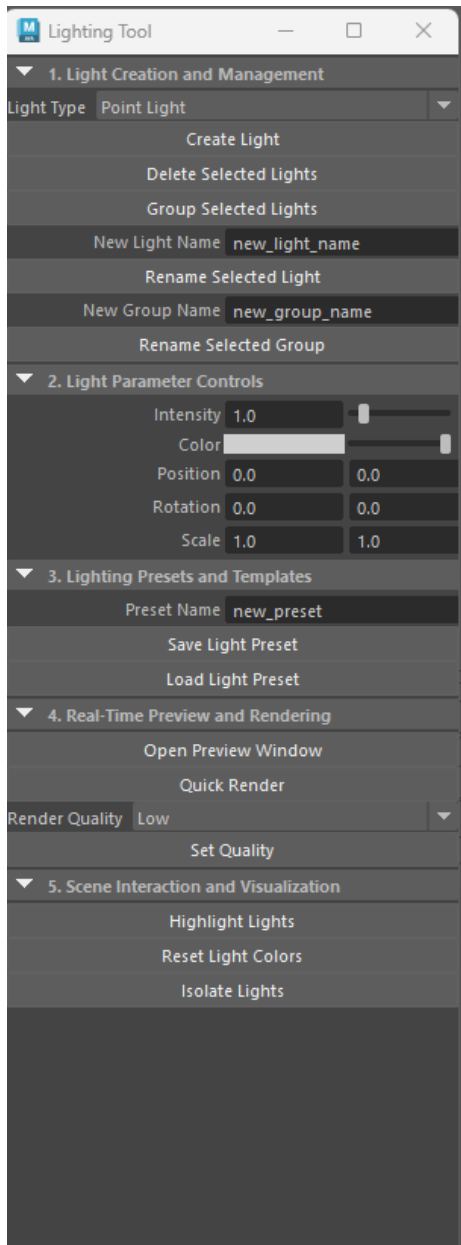
1.1. Run the script in Maya's Script Editor to open the Lighting Tool.

Lighting tool: https://github.com/EmKinder/TD_Assignment2/blob/LightingTool/Lighting%20tool/lightcreationv4

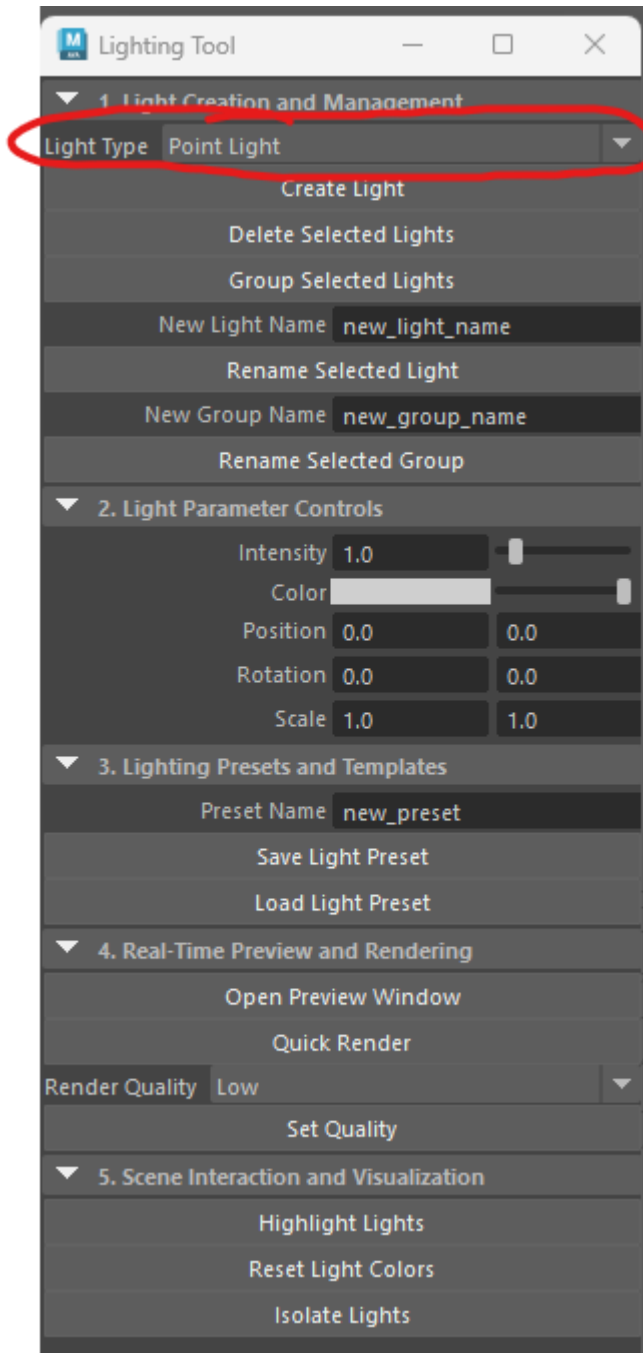
A screenshot of a code editor window showing a Python script for creating and managing lights in Maya. The script is titled 'lightcreationv4' and is located in a file named 'light.py'. The code is written in Python 3 and uses the 'maya.cmds' module for Maya commands. It defines three main functions: 'create_light', 'delete_selected_lights', and 'group_lights'. The 'create_light' function takes a 'light_type' and optional arguments, and creates a light of the specified type. The 'delete_selected_lights' function deletes all selected lights. The 'group_lights' function groups all selected lights into a single group named 'light_group'. The script also includes a warning message for invalid light types.

```
1 import maya.cmds as cmds
2 import os
3
4 def create_light(light_type, *args):
5     light_types = {
6         "Point Light": "pointLight",
7         "Directional Light": "directionalLight",
8         "Spot Light": "spotLight",
9         "Area Light": "areaLight"
10    }
11    if light_type in light_types:
12        light_name = cmds.shadingNode(light_types[light_type], asLight=True)
13        light_transform = cmds.listRelatives(light_name, parent=True, fullPath=True)
14        if light_transform:
15            cmds.select(light_transform)
16            return light_transform[0]
17        else:
18            cmds.warning("Invalid light type specified")
19
20 def delete_selected_lights(*args):
21     selected_objects = cmds.ls(selection=True, dag=True, long=True)
22     for obj in selected_objects:
23         if cmds.objectType(obj, isType="light"):
24             light_transform = cmds.listRelatives(obj, parent=True, fullPath=True)
25             if light_transform:
26                 cmds.delete(light_transform)
27             elif cmds.listRelatives(obj, children=True, type="light"):
28                 cmds.delete(obj)
29
30 def group_lights(*args):
31     selected_objects = cmds.ls(selection=True)
32     if selected_objects:
33         group_name = cmds.group(selected_objects, name="light_group")
34         return group_name
35     else:
```

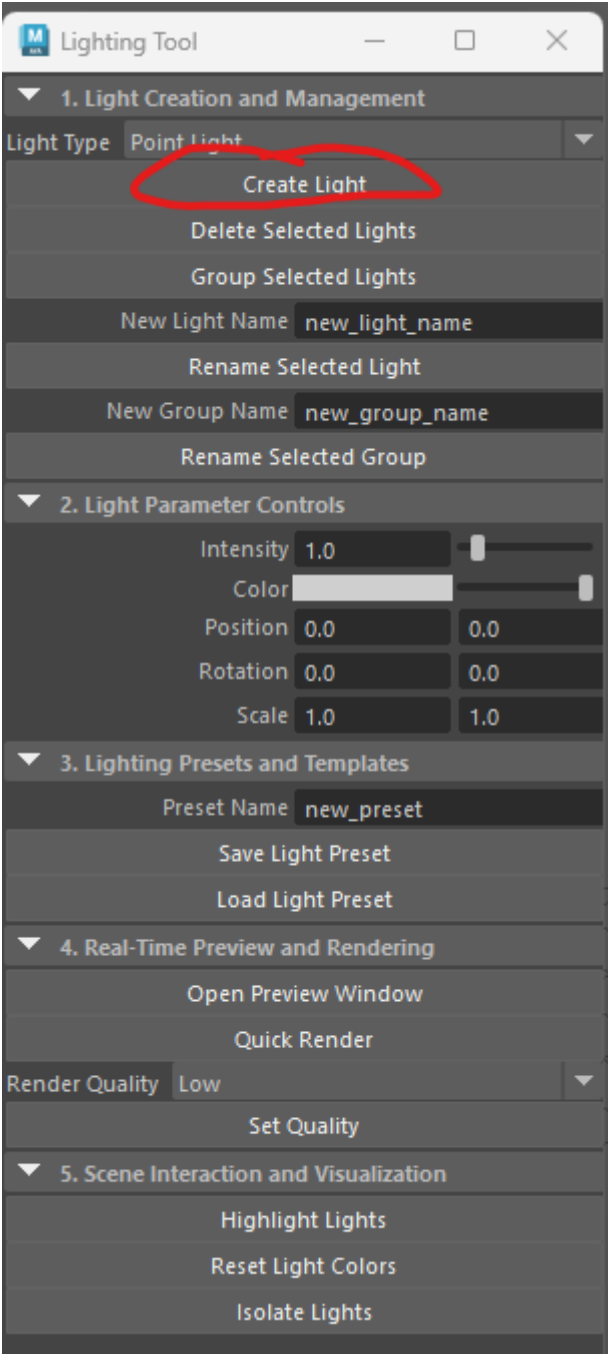
1.2. After the code has been run UI should popup



2. Light Creation and Management:

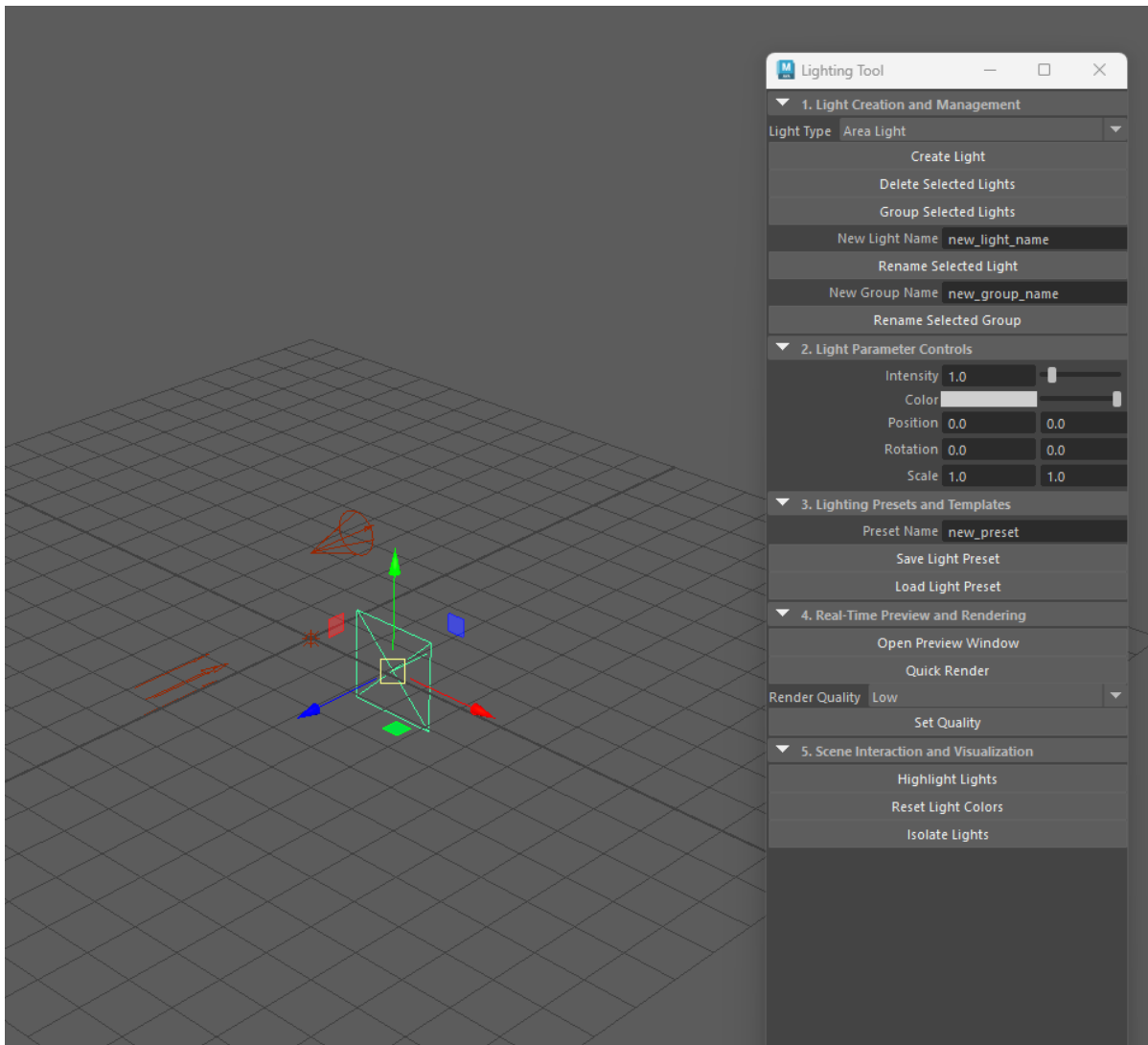


2.1. Through Drop down menu select the type of light you want to create

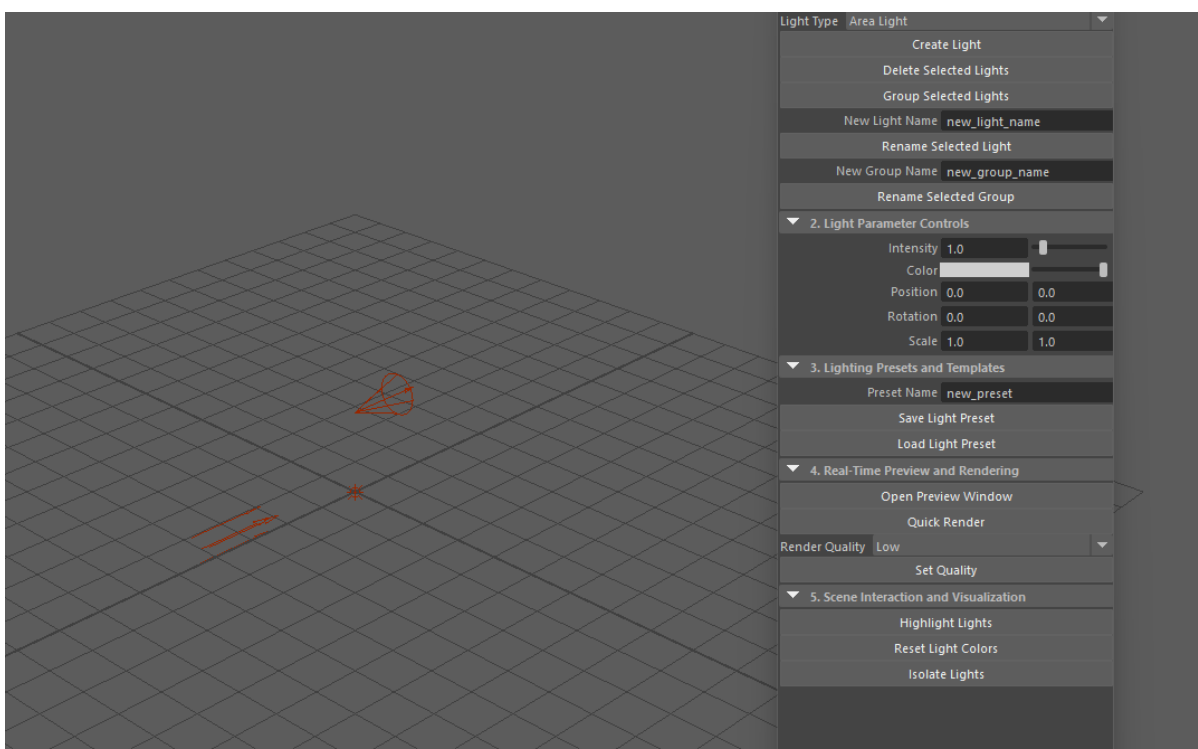


Click create light to create the light in scene.

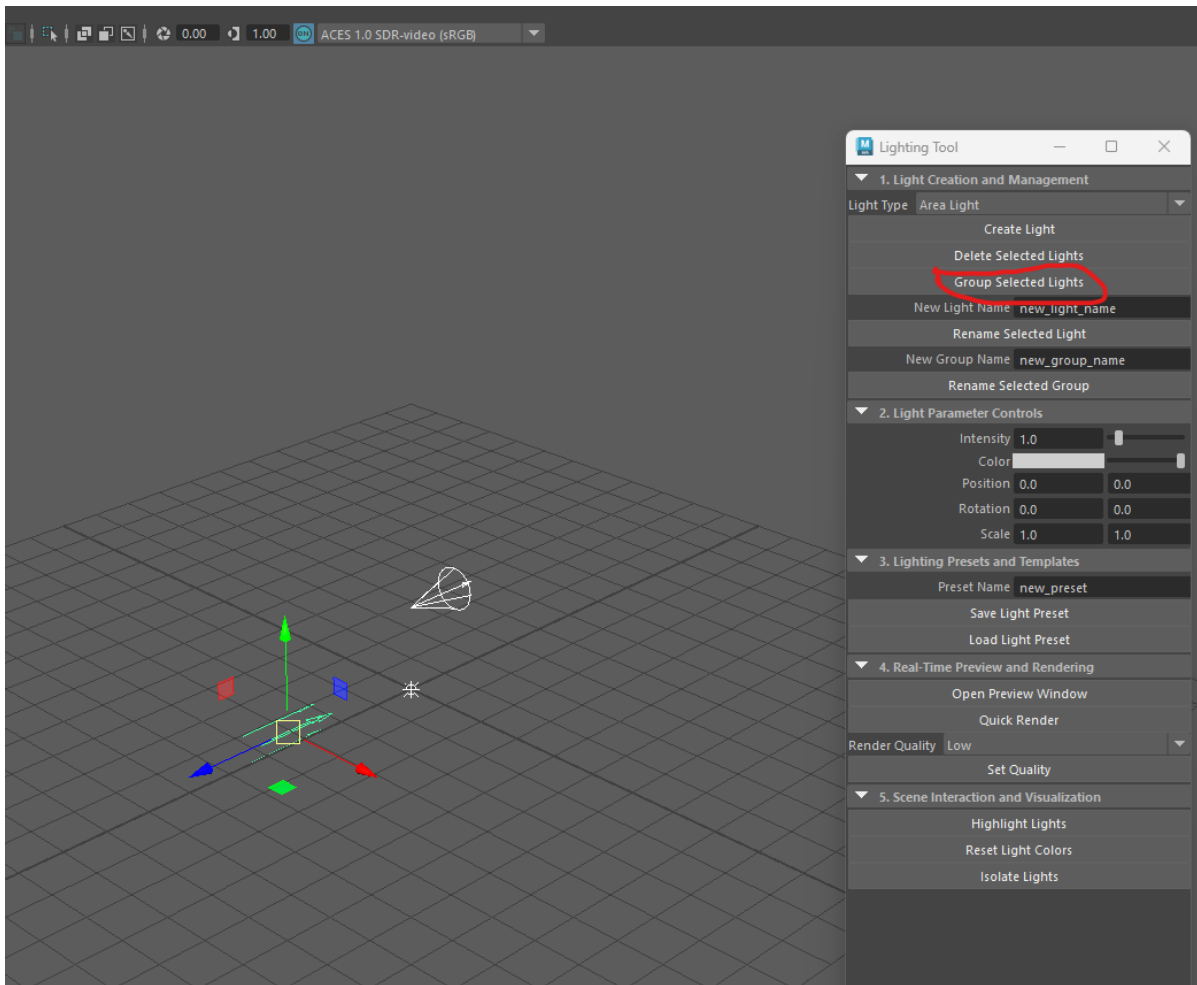
2.2. Select any light you want to delete, and press delete the selected light to delete the light.



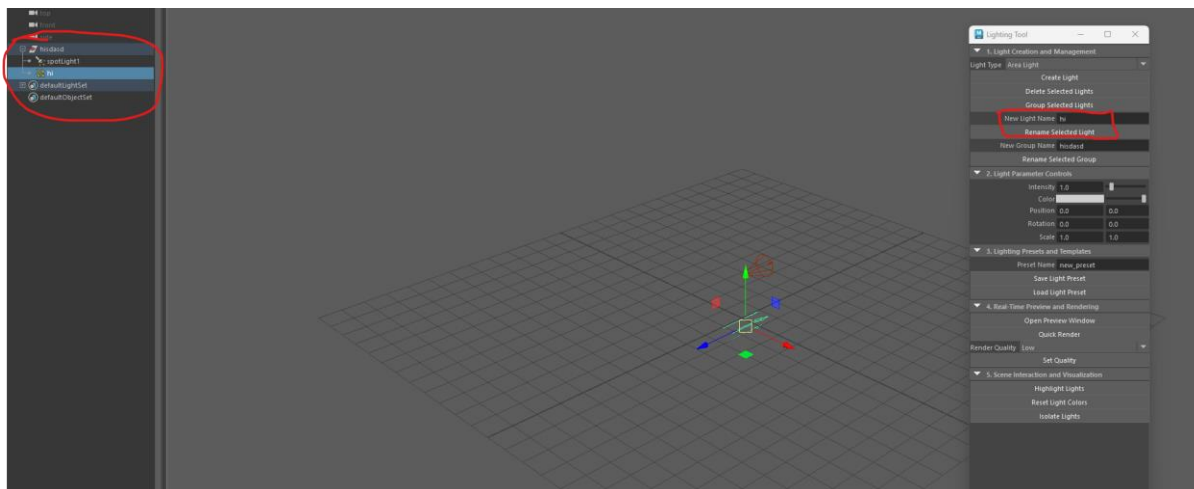
Which will then remove the selected light.



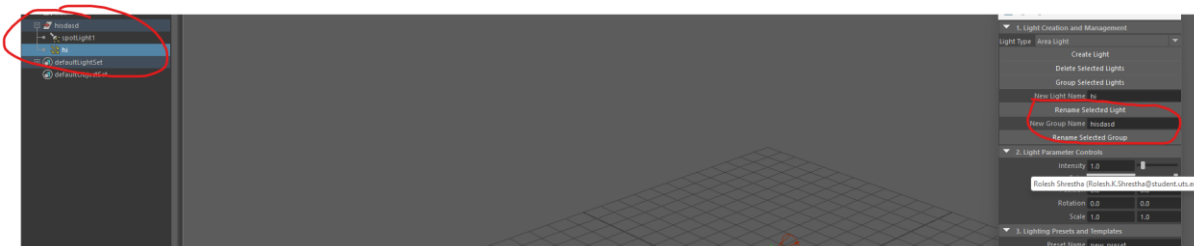
2.3. Select multiple lights and then click on group selected lights to group them.



2.4. To rename the name of light click on the light you want to rename and type the name you would like to keep then press rename.



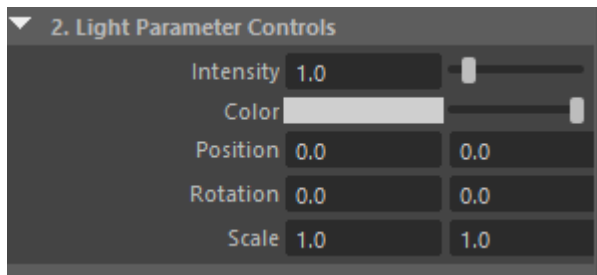
2.5. To change the group name, click the group and type the name you want to rename then press rename group name.



3. Light Parameter Controls:

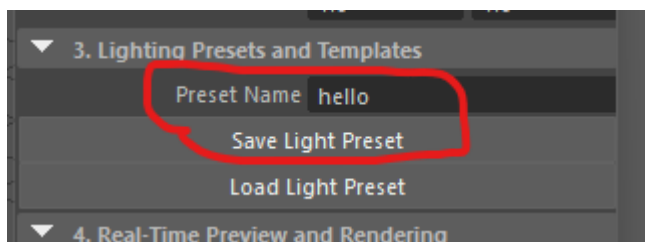
Adjust the sliders and color picker to modify the properties of the selected lights.

Use the position, rotation, and scale fields to numerically adjust these attributes.

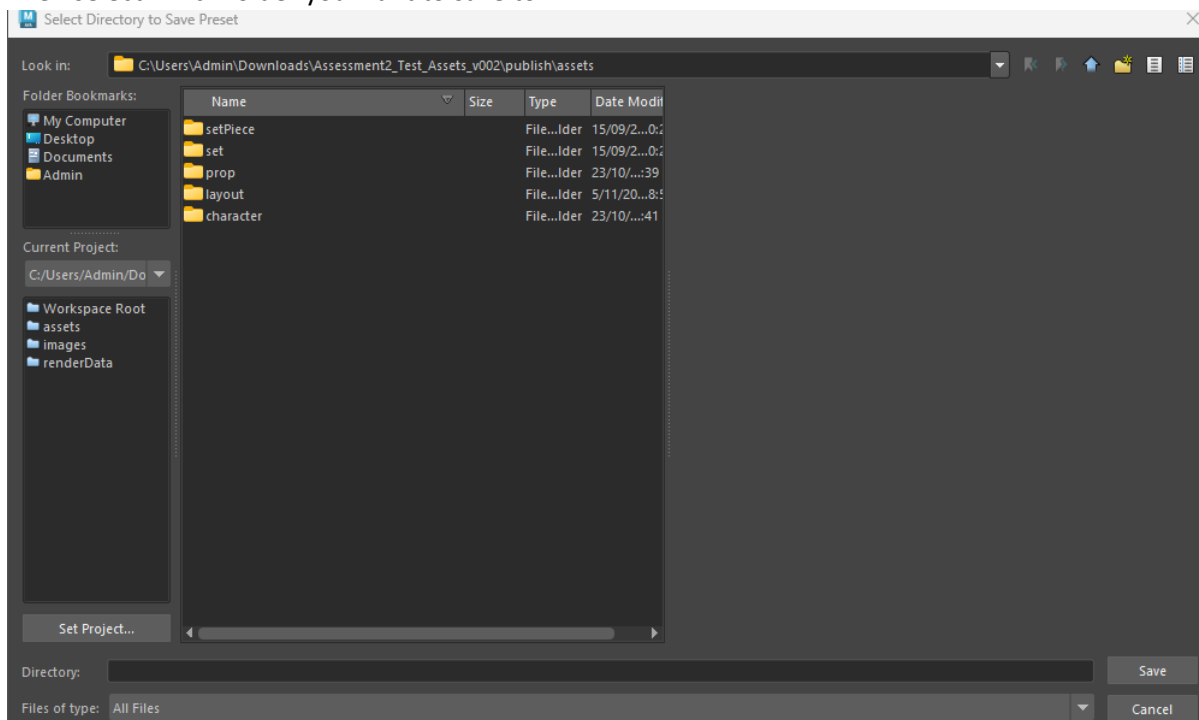


4. Lighting Presets and Templates:

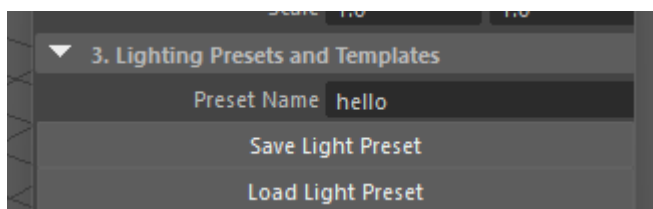
4.1 Enter the name you want to save your preset as and then click on save preset.



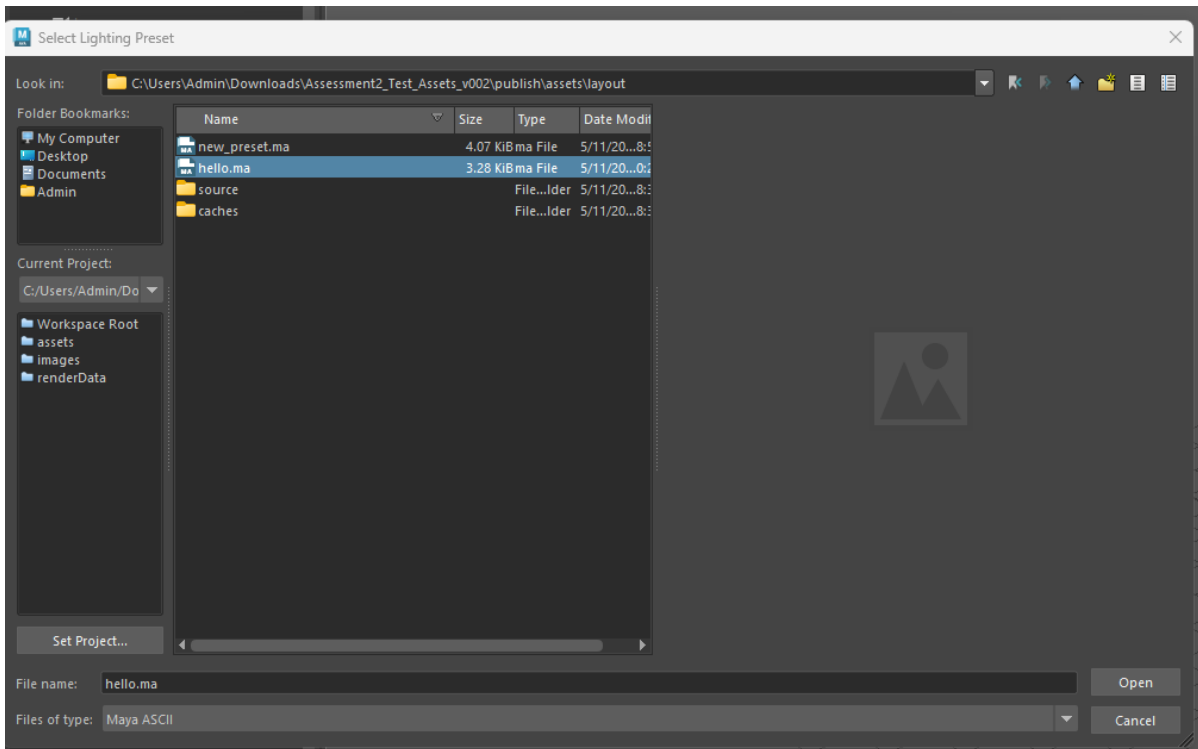
Then select which folder you want to save to.



4.2. Load Light Preset: Click this button to load a previously saved lighting preset.

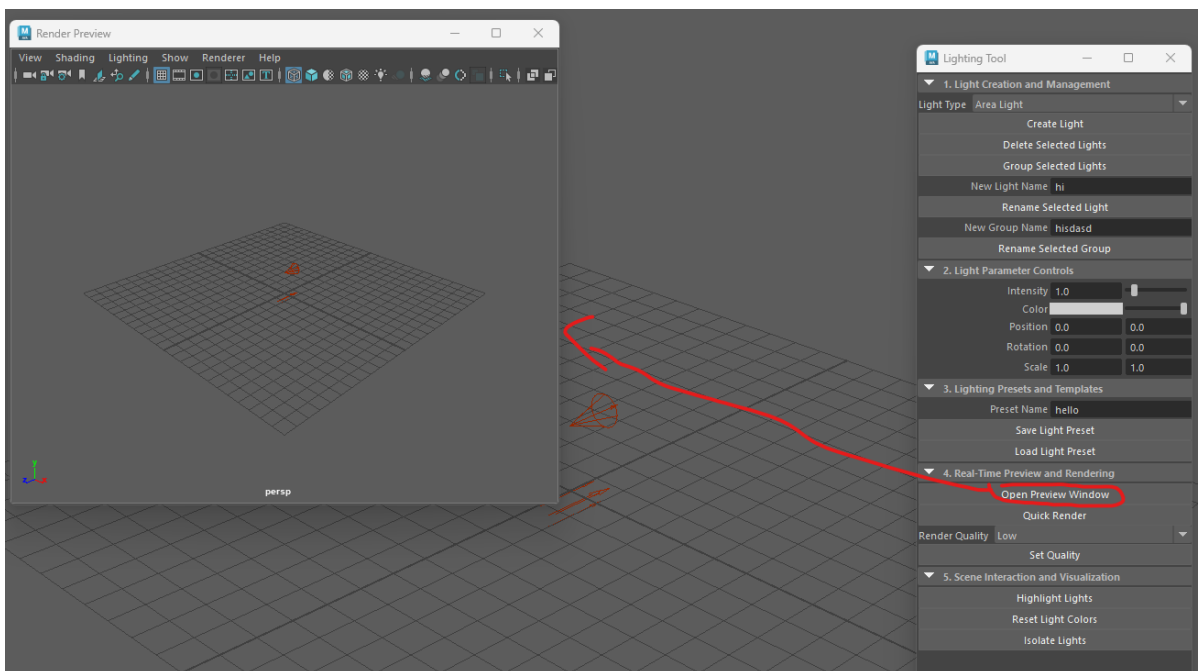


After you have pressed load click on the preset you want to open



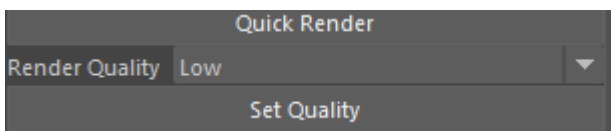
5. Real-Time Preview and Rendering:

5.1. Open Preview Window: Click to open a window that displays a real-time preview of the lighting.



5.2. Quick Render: Click to perform a quick render of the current view.

Set Quality: Choose a render quality from the dropdown and click "Set Quality" to apply it.



Note: This tool assumes Arnold is the active renderer for certain features. Ensure Arnold is set up correctly in your Maya environment to utilize all functionalities. The scene interaction and visualization tool is currently in development.