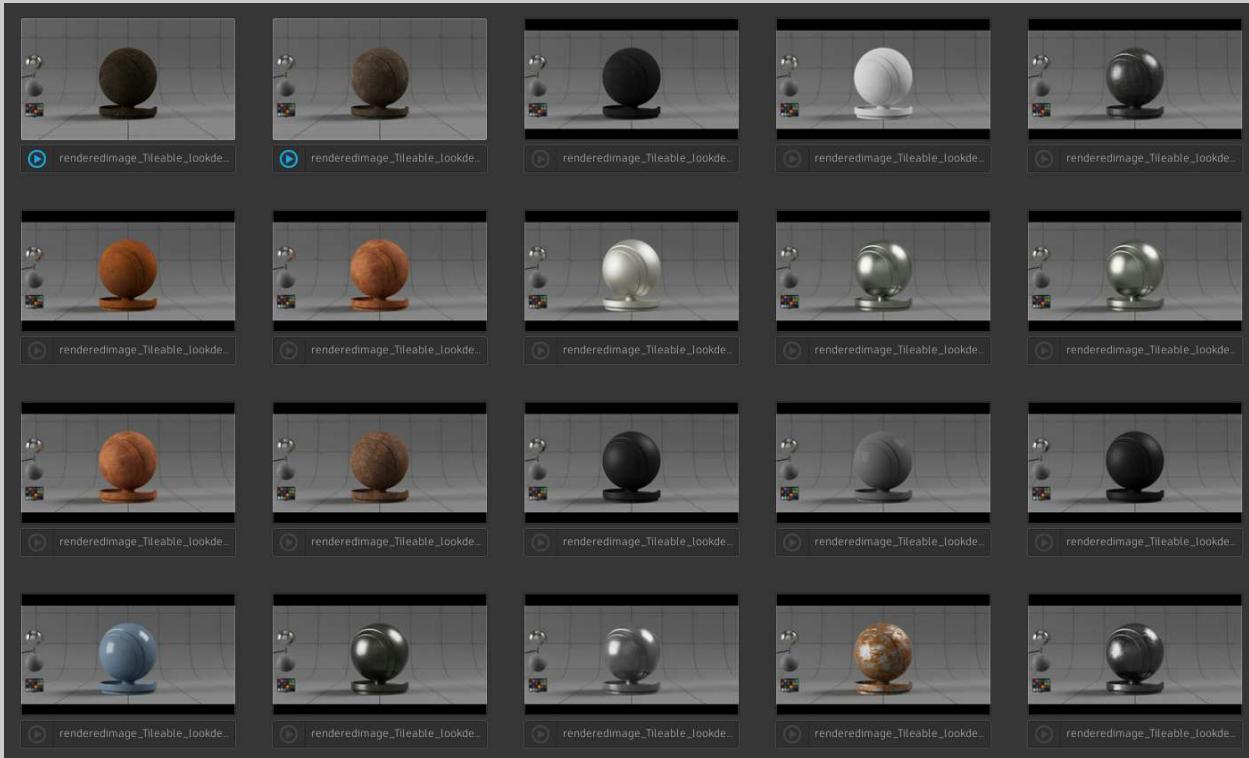


VFX Tech Reel (2014-2021)

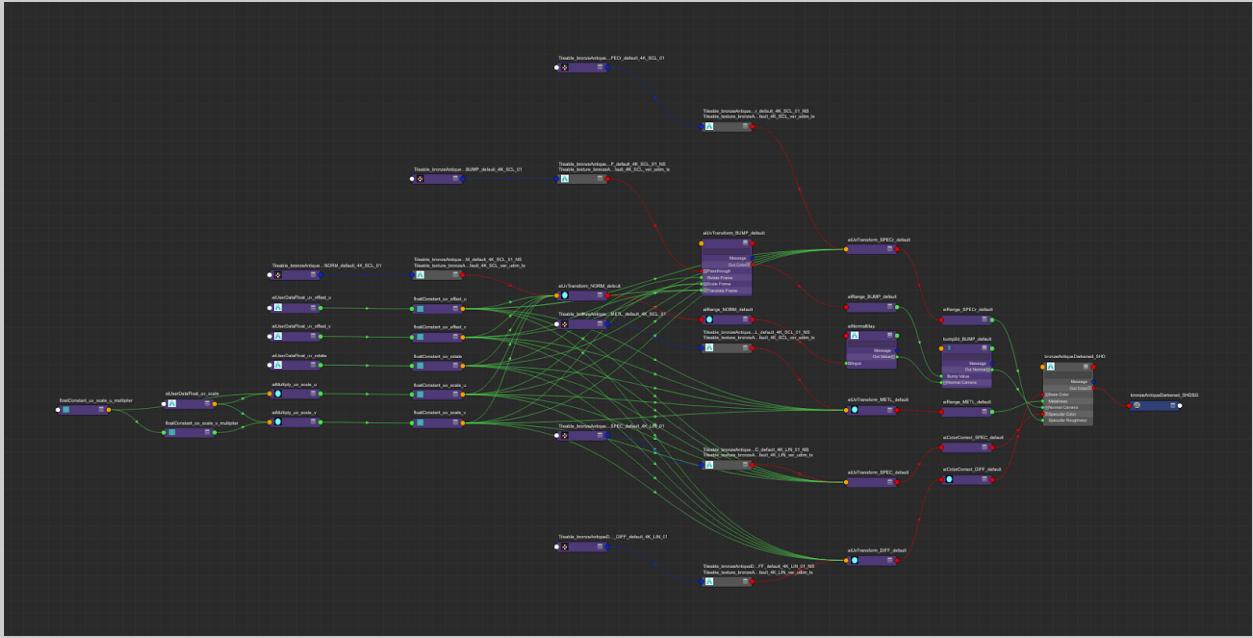
General Tools

Project Based Material Library

Batch extract maps from substance sbs files with substance-automation-toolkit. Create maya shader network automatically using Python with published texture maps.

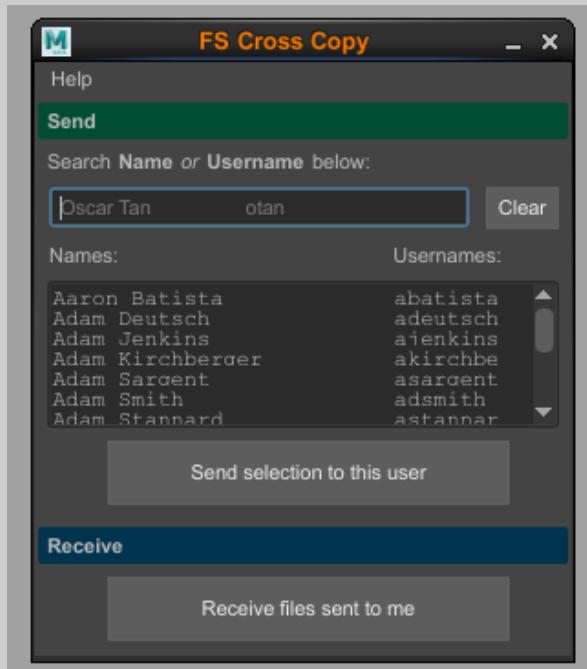


All shading network are generated with Python procedurally:



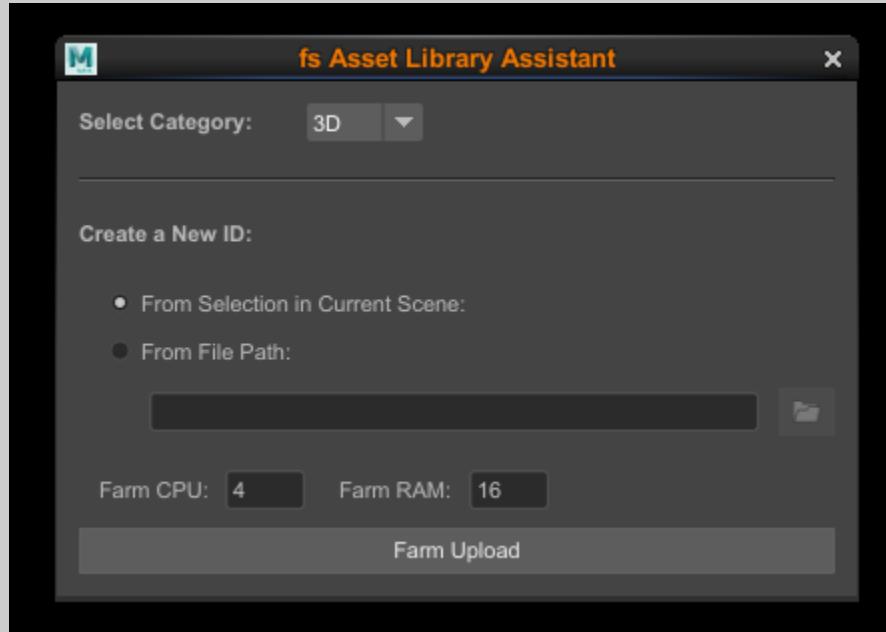
Maya Crosscopy

Sharing scene contents in-between users. Send selected maya scene elements to designated colleagues by type searching their names.



In-House Asset Library Manager

A maya GUI tool to upload specified geometry to in-house web-based asset library.



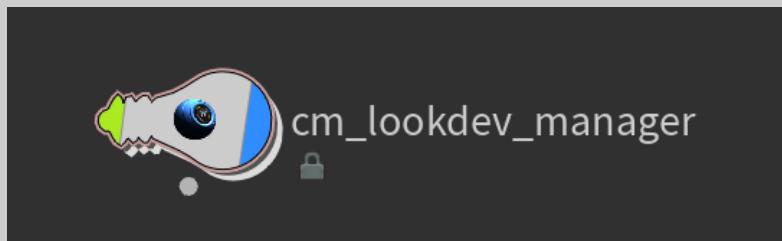
The screenshot shows the "AssetLibrary" interface. The top navigation bar includes tabs for 2D, 3D (selected), Stills, Materials, Ref, and Collections. On the left, there are two expandable sections: "Tags" (listing categories like new, anatomy, animals, etc.) and "Filters" (listing Fx, Quality, Type). The main area displays a search results grid with 237 items, sorted by popularity. Each result card shows a thumbnail image, an ID (e.g., 253444, 252951, 252915, 252911, 252899, 252879, 251908, 251904, 251900), and three small icons for selection, preview, and addition. On the right side of the interface, there are several control buttons: a plus sign, a magnifying glass, and minus signs.

Lookdev and Lighting Tools

Houdini Lookdev Manager

By using shotgun and framestore internal APIs, this tool will fetch pipeline published Alembic geometry and generate placeholder shader networks for lookdev artists. This Houdini digital asset also contains several turntable lighting presets.

The tool node:



Python module script:

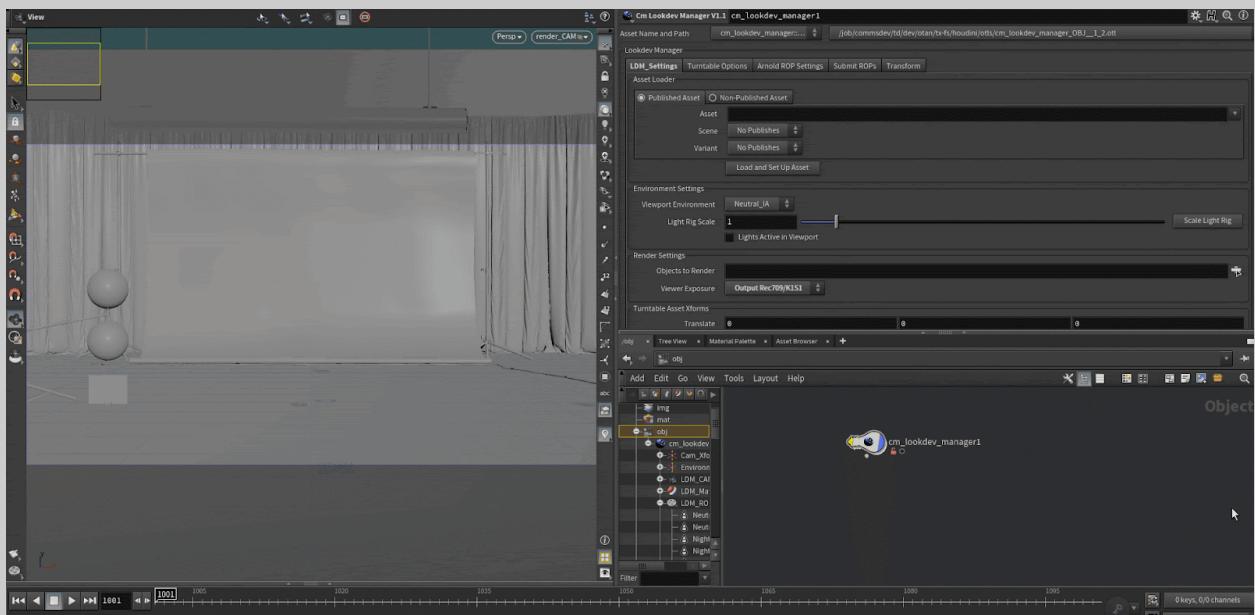
The screenshot shows a Python code editor window with the following code:

```
import os
import commstd
import sgtk

# ~~~~~#
# ~~~~~#
# Lookdev Manager
# ~~~~~#
# ~~~~~#
# Lookdev Manager supplies artist with several calibrated light rigs along with robust turntable
# along with template building of your lookdev setup with shaders based on the material token.
#
# ~~~~~#
# ~~~~~#
# set_data_parms:
# Setting all relevant turntable settings which dynamically drives current camera, along with ro
# lights and model rotations.
def set_data_parms():
    with hou.undos.disabler():
        current_node = hou.pwd()
        # current_node = hou.node('/obj/Lookdev_Manager1')
        frame_dict = {}
        no_mo.blur_list = []
        total_frame_amount = 0
        starting_frame = current_node.parm('LDM_StartFrame').eval()
        model_rotation_amount = current_node.parm('LDM_Model_Rotations_Multi').eval()
        for index in range(1, model_rotation_amount+1):
            index_dict = {}
            rot_amount = current_node.parm('LDM_Model_Frame_Length_'+str(index)).eval()
            rot_range_x = current_node.parm('LDM_Model_Rotation_Range_'+str(index)+x).eval()
            rot_range_y = current_node.parm('LDM_Model_Rotation_Range_'+str(index)+y).eval()
            cam_toggle = current_node.parm('LDM_Model_Camera_Override_Toggle_'+str(index)).eval()
            cam_path = current_node.parm('LDM_Model_Camera_Override_'+str(index)).eval()
            first_frame = starting_frame + total_frame_amount
            end_frame = starting_frame + total_frame_amount + rot_amount
            index_dict['cam_toggle'] = cam_toggle
            index_dict['cam_path'] = cam_path
            index_dict['keyframes'] = [first_frame, end_frame-1]
            index_dict['rotation_range'] = [rot_range_x, rot_range_y]
            index_dict['mode'] = 'asset'
```

The code is a Python script for a Houdini node named "Lookdev Manager". It defines a function `set_data_parms()` that sets up various parameters for a turntable setup. The script uses Houdini's `hou` API to interact with nodes and parameters.

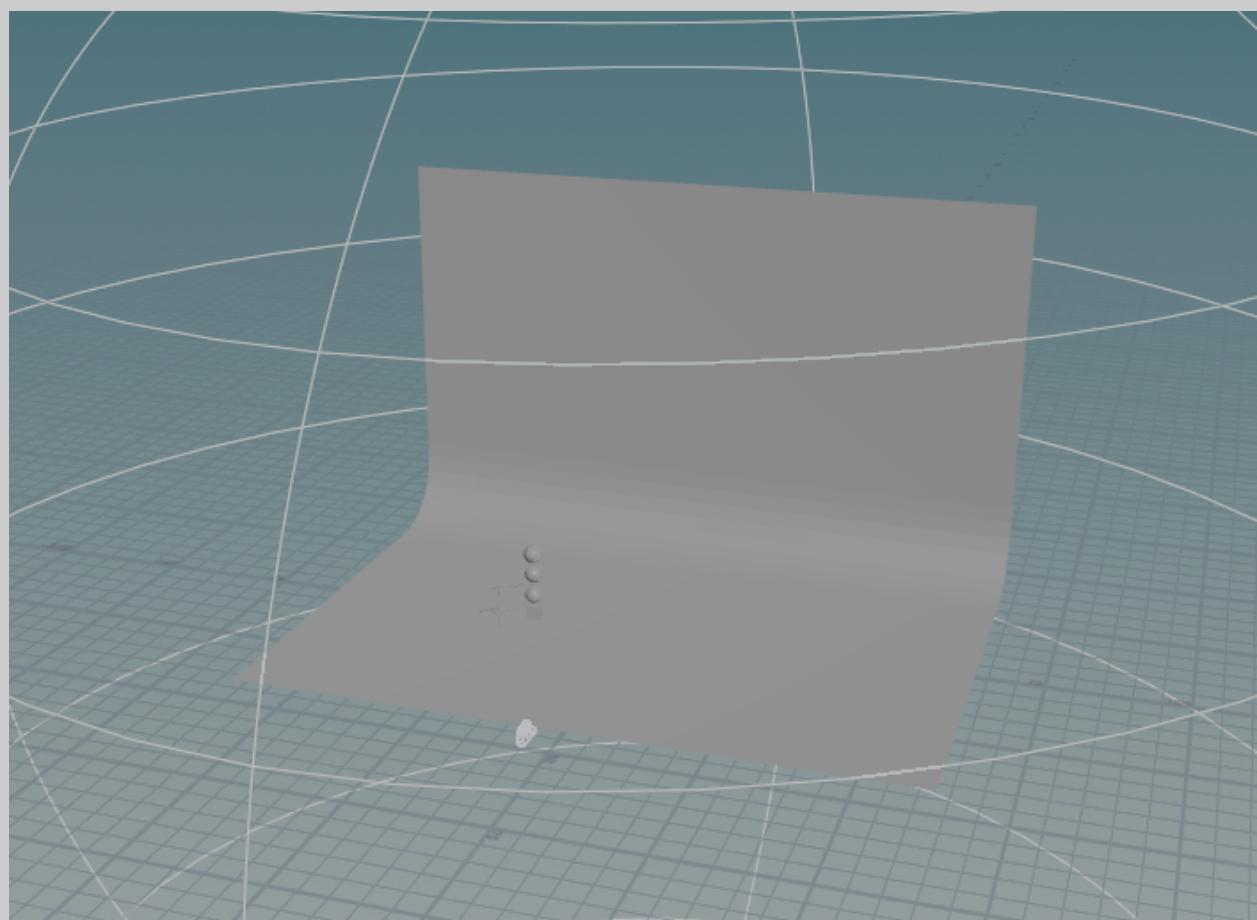
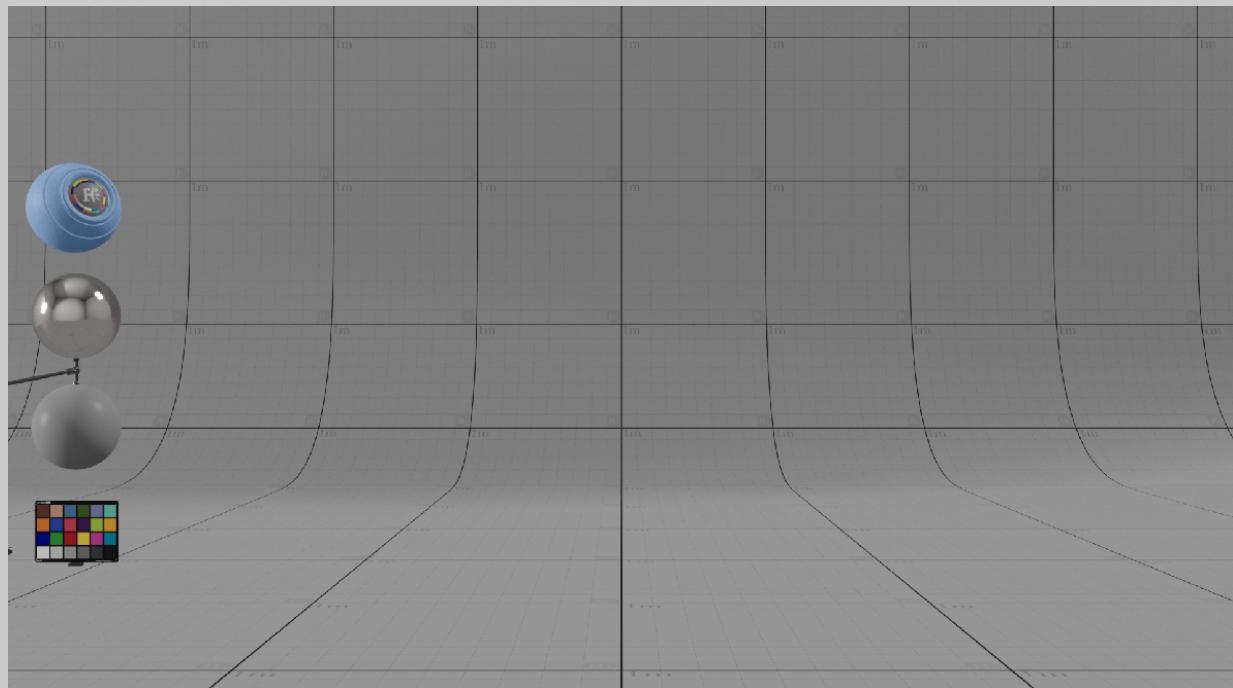
Parameter gui:



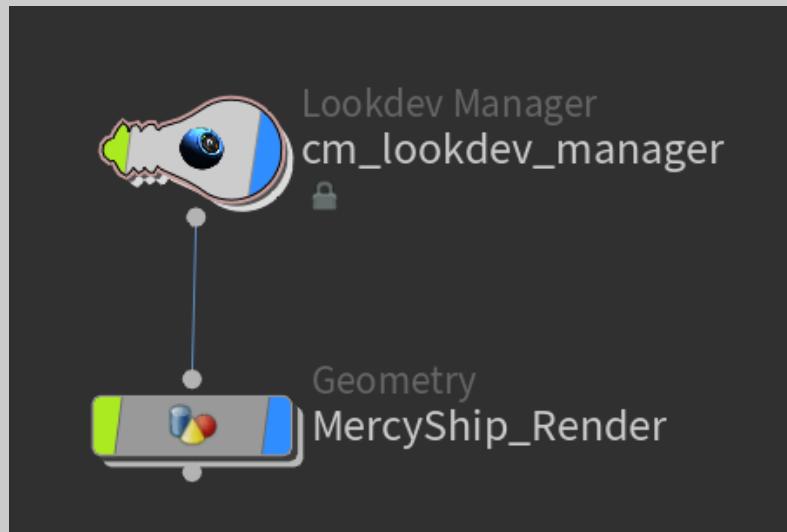
Inside of HDA node:



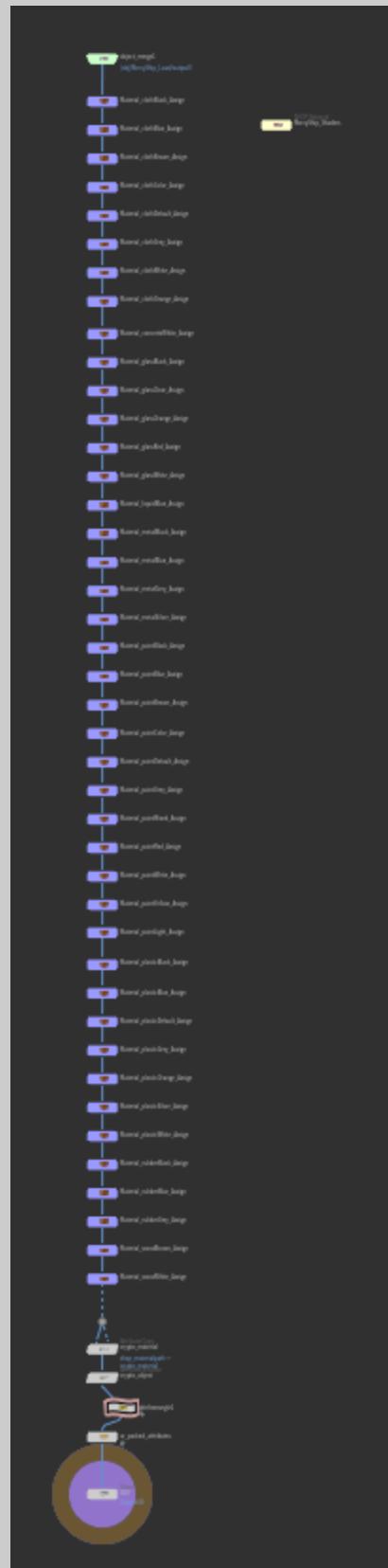
Empty turntable scene before importing models



This tool will generate a “*_Render” node automatically after select the model:



Inside of the “*_Render” node(all materials nodes are generated by script):



Final finished lookdev turntable render:



Personal Tools Dev

<https://github.com/tawibox/txmaya>

tx Node Finder

A tool allows the user to filter nodes by name or by node type.

tx Node Finder X

• by Name ● by Type DAG nodes only

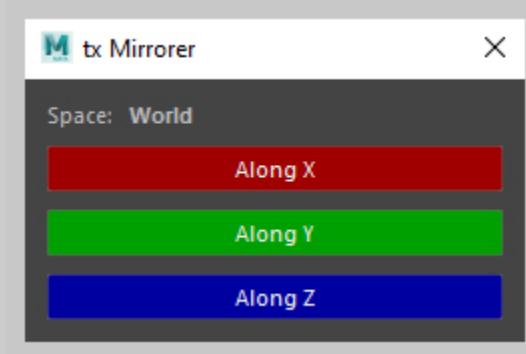
Refresh List Clear

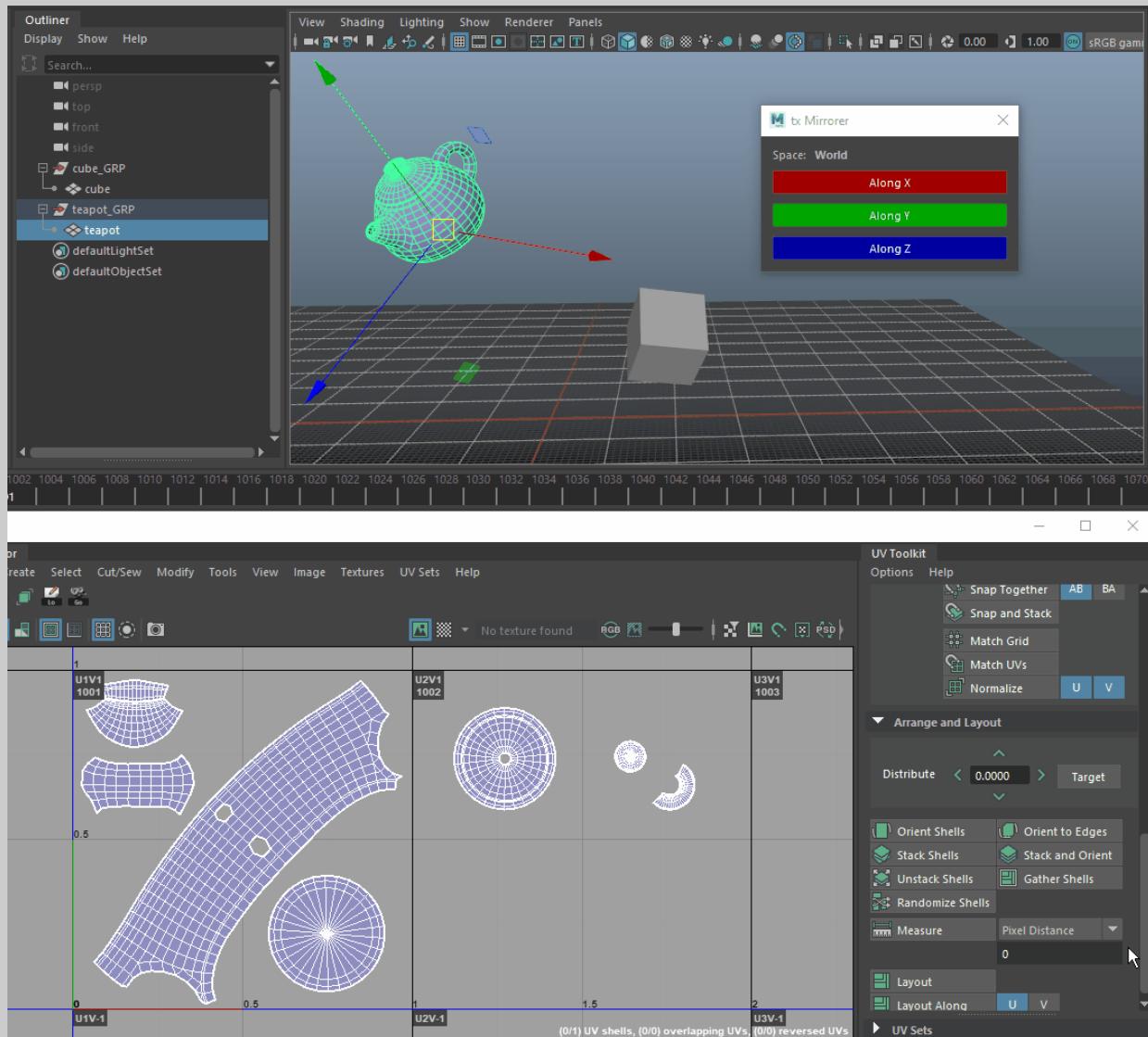
Name	Type	DAG	Long Name
front	transform	Y	front
frontShape	camera	Y	front frontShape
nurbsCircle1	transform	Y	nurbsCircle1
nurbsCircle2	transform	Y	nurbsCircle2
nurbsCircleShape1	nurbsCurve	Y	nurbsCircle1 nurbsCircleShape1
nurbsCircleShape2	nurbsCurve	Y	nurbsCircle2 nurbsCircleShape2
persp	transform	Y	persp
perspShape	camera	Y	persp perspShape
side	transform	Y	side
sideShape	camera	Y	side sideShape
teapot_A	transform	Y	teapot_A
teapot_AShape	mesh	Y	teapot_A teapot_AShape
teapot_B	transform	Y	teapot_B
teapot_BShape	mesh	Y	teapot_B teapot_BShape
top	transform	Y	top
topShape	camera	Y	top topShape



tx Mirrorer

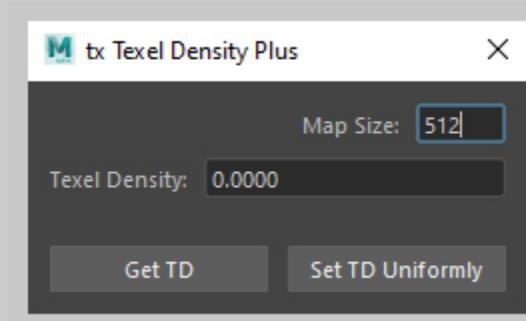
A world-space mirroring tool to mirror multiple geos and their UVs without losing transformations.

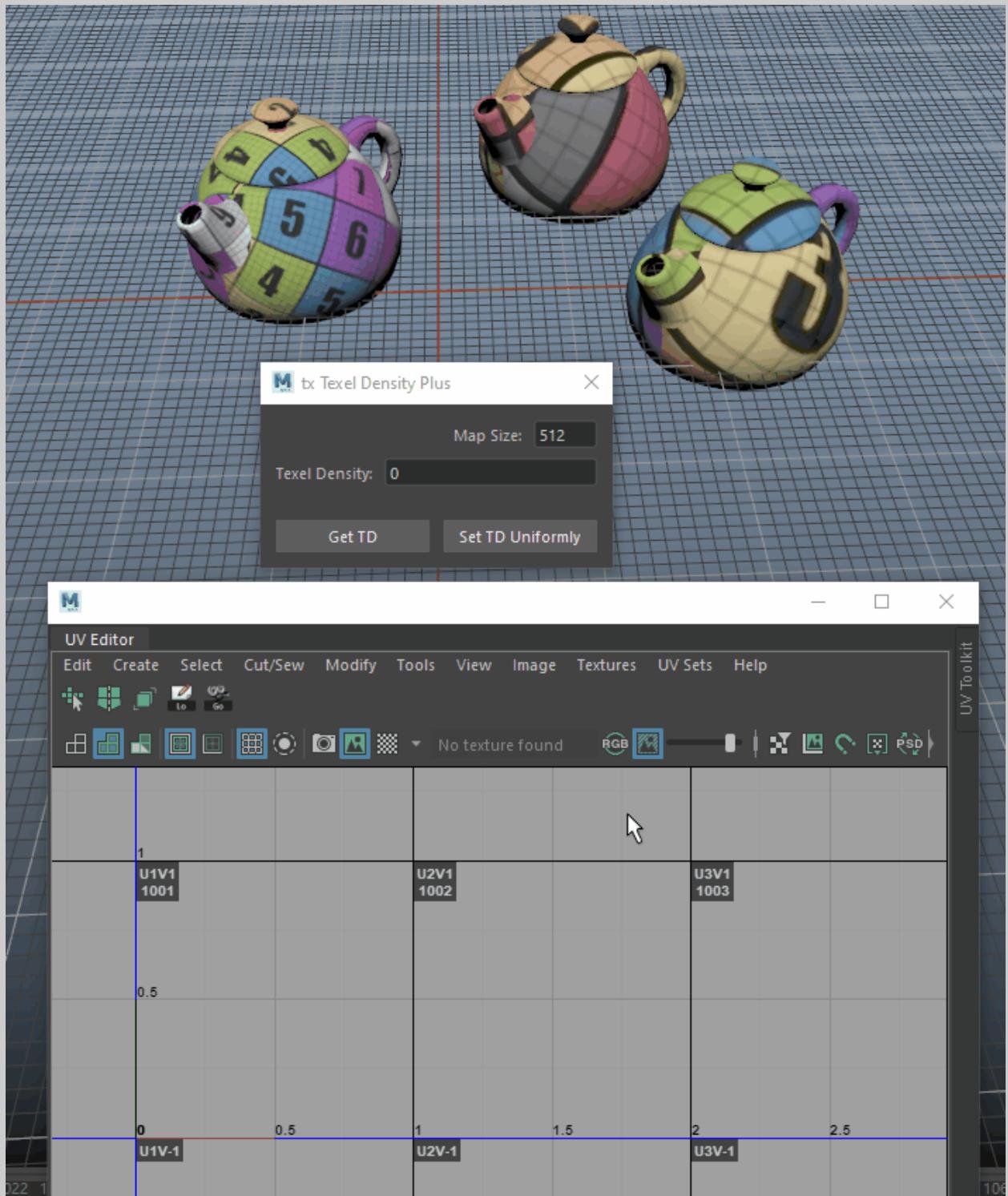




tx Texel Density Plus

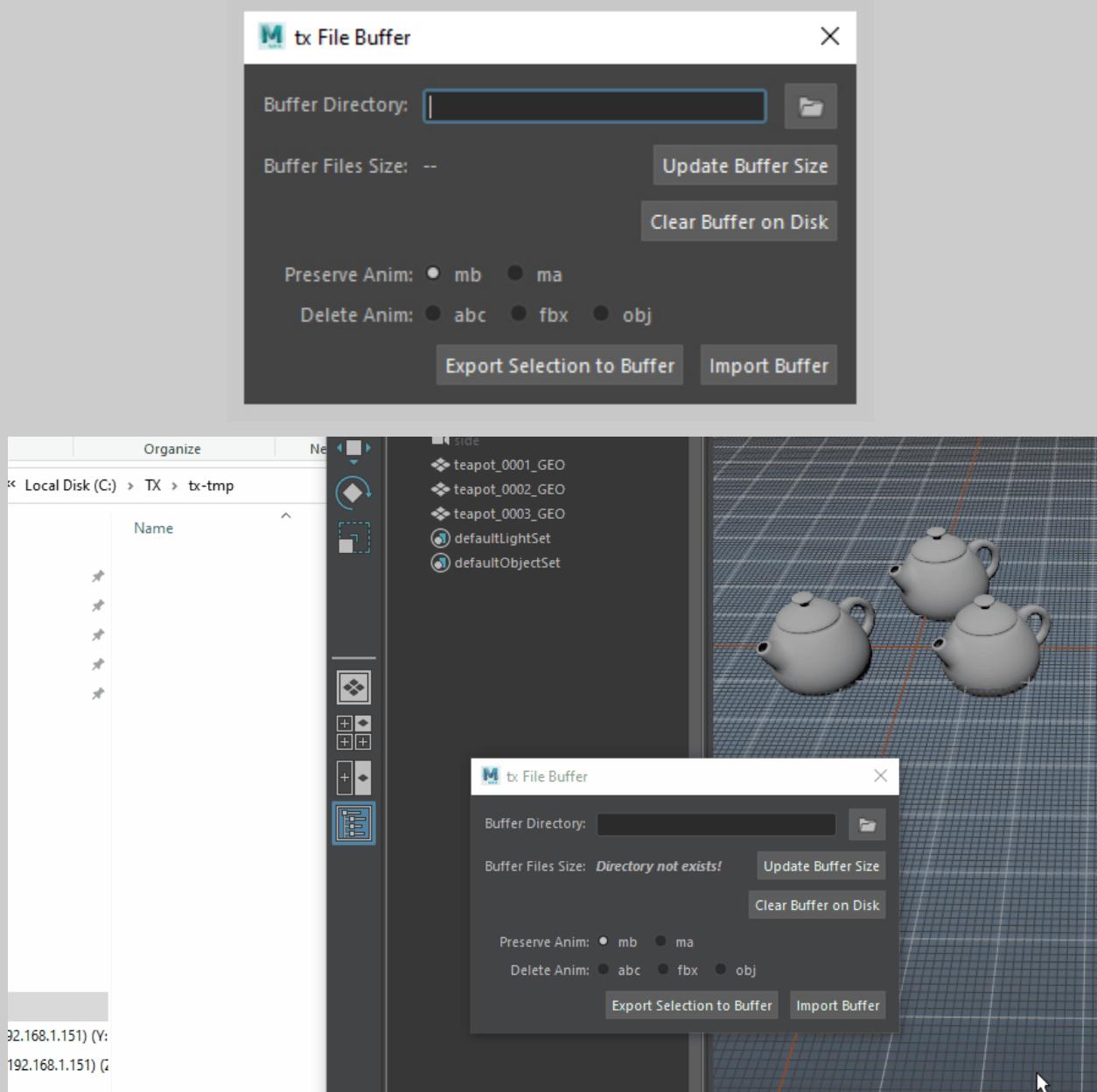
A tool to get selections' UV texel density and scale other selections' UV altogether based on the TD value.





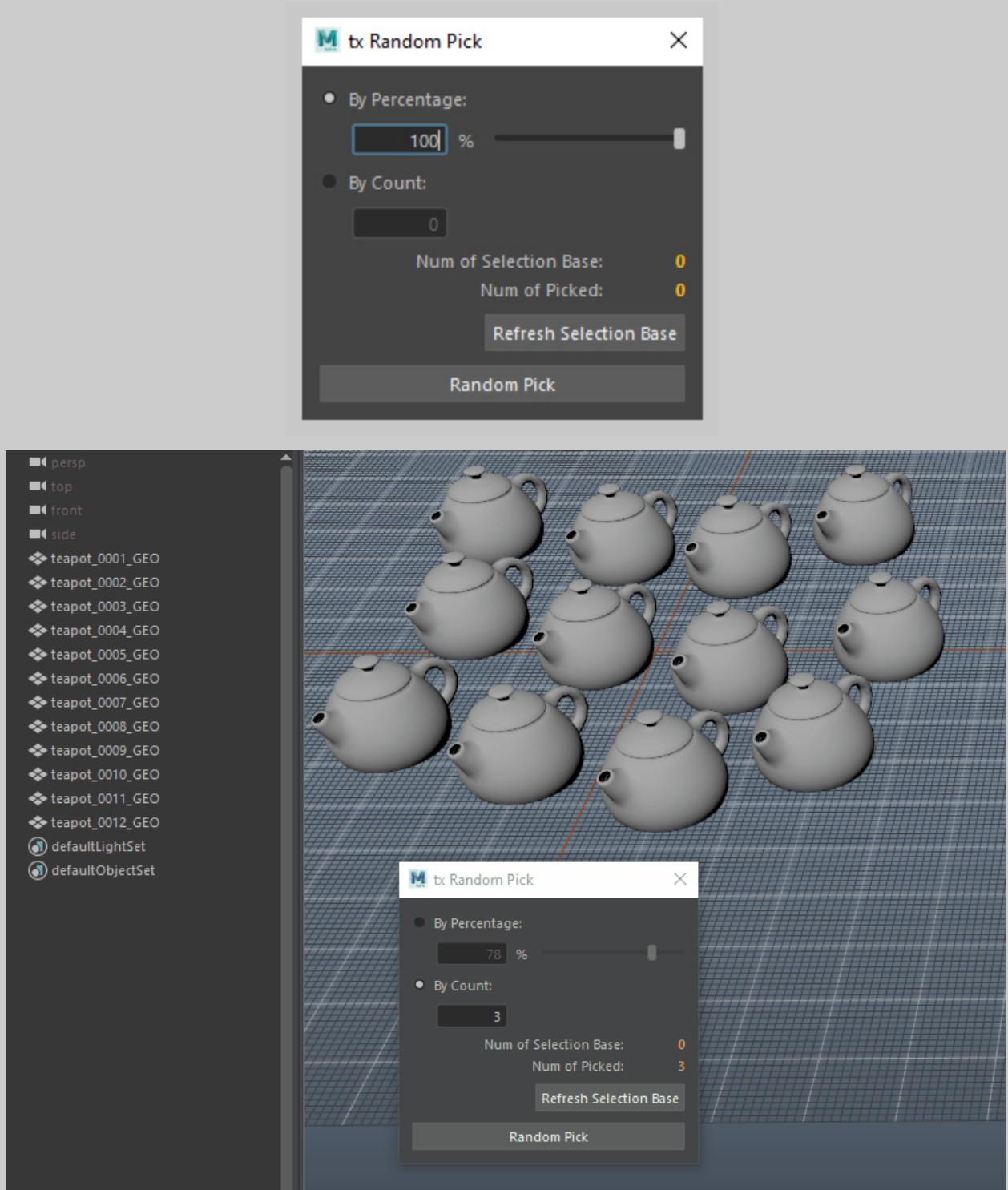
tx File Buffer

A tool allows users to export selection to a temporary folder and re-import back. Could be used for sending files in-between Mayas or cleaning up geometries.



tx Random Pick

A tool to randomly select some members from a group of items.



tx UV Batch Transfer

A tool to transfer UVs in-between two groups of geos with 1 click. (Based on topology)

