

## Project Details

*The publisher is interested in releasing a 3D action platformer for mid-range devices. To prepare for production, automation tools must be developed which can batch process 3D models and textures to reduce poly counts and texture sizes, and palettize/stylize textures based on color comps and filters.*

*Using the 3D Game Kit from the Asset Store as a prototype:*

- *Reduce the LOD poly counts for all environment assets by 50%*
- *Reduce the texture sizes for all environment assets by 75%*
- *Create a palletizer which can recolor the original textures based on color comps*
- *Create a stylizer which can process textures using filter(s)*

*Demonstrate the following:*

- *Before and after poly counts for individual assets and the rendered game view in Level 1*
- *Before and after texture file sizes and texture memory usage in Level 1*
- *Before and after images showing original assets and reduced assets*
- *3 each palletization and stylization renders*

*The following assets were used in this project:*

- [3D Game Kit](#)

**The best video I've ever made:**

<https://youtu.be/cElw4b7rWd4>

## Analysis

Due to the required features, some questions that came up included:

- Which program should be used for poly reduction?
- How should the palletizer and colorizer be set up?

## Approach

After doing research into scripting with Maya/Python and learning more about Photoshop's automation capabilities the following process was implemented:

- Write up a script using python in Maya that takes advantage of Maya's polyReduce and FBXExport functions
- Use Photoshop actions and batch automation to handle the colorization and palletization
- Comp together a beautiful video with Premiere Pro



*Game view render of the scene after reduction.*

## Poly Reduction

There was only one short python script that handled the poly reduction. As soon as the user runs it, it prompts them to choose a directory to read FBX files from. It then walks through the entire directory searching for, poly reducing by 50%, and finally exporting as an FBX, overwriting the original file.

```
for root, dirs, files in os.walk(cleanPath):
    print root, dirs, files
    for fileName in files:
        if fileName.endswith('.fbx'):
            print (fileName)
            filePath = root + "/" + fileName
            cmds.file(filePath, open = True, force = True)
            obj = cmds.ls(geometry = True)
            for item in obj:
                try:
                    cmds.polyReduce(p = 50, n = cmds.select(item))
                except:
                    pass
            pm.mel.FBXExport(f = filePath)
```

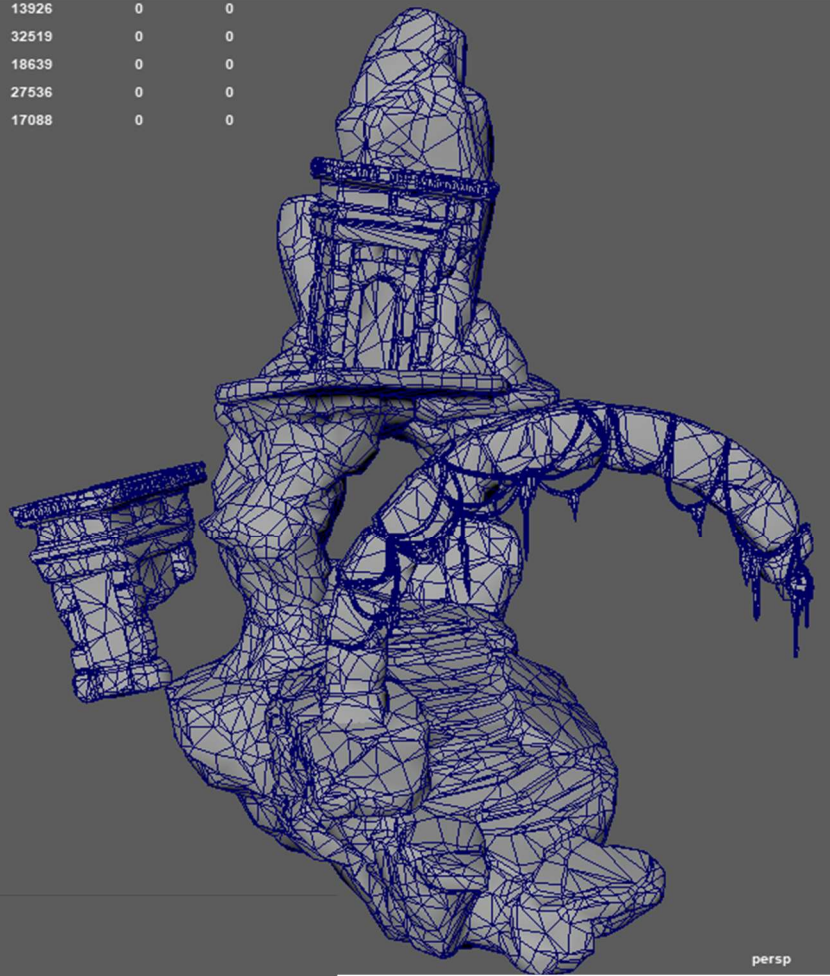
*The main loop of the script, showing how it traverses, reduces, and exports.*



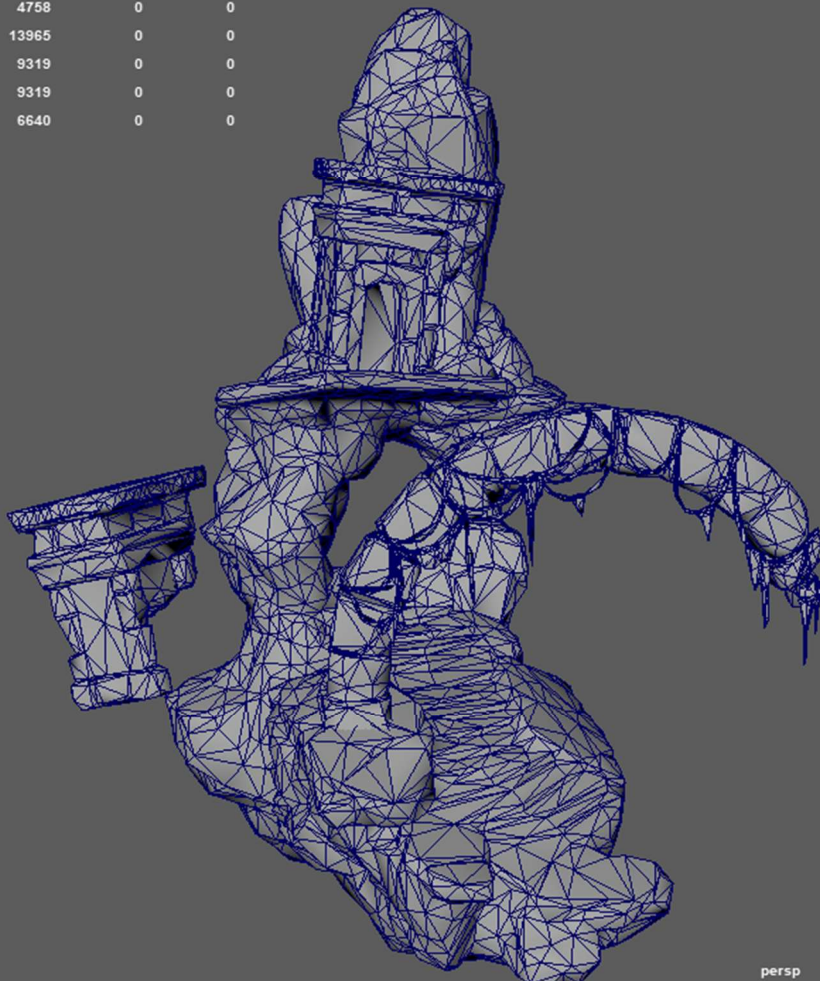
*Before (top right) and after (bottom left) of the 50% poly reduction on an alien statue's largest LOD.*



Verts:	13926	0	0
Edges:	32519	0	0
Faces:	18639	0	0
Tris:	27536	0	0
UVs:	17088	0	0



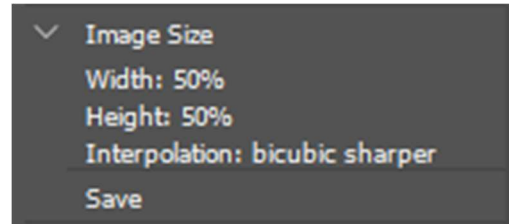
Verts:	4758	0	0
Edges:	13965	0	0
Faces:	9319	0	0
Tris:	9319	0	0
UVs:	6640	0	0



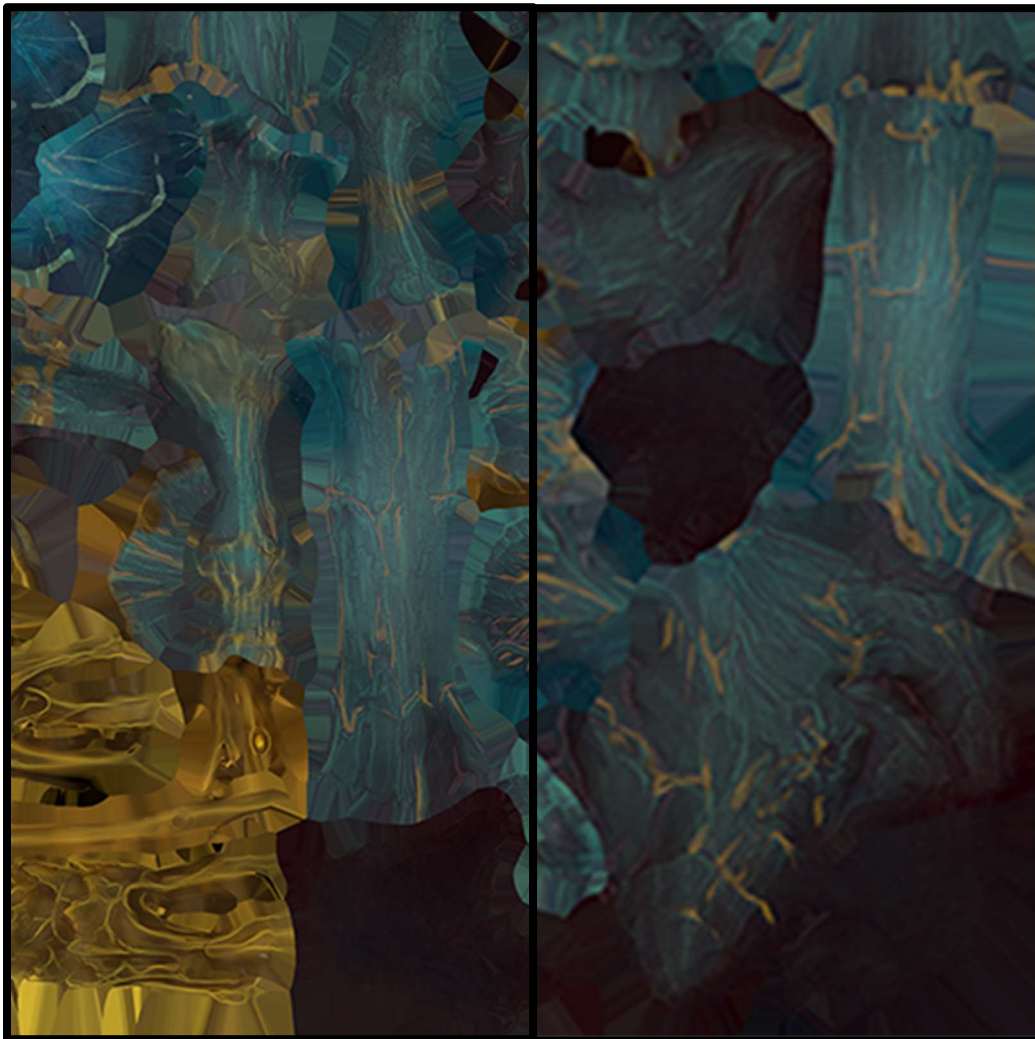
*Before (top right) and after (bottom left) of the 50% poly reduction of a very large monument's largest LOD.*

## Texture Reduction

The resizing was a very simple action in Photoshop that reduces the image's width and height by 50% to achieve the 75% size reduction requested. The images are then saved. After that, Photoshop's automation > batch is used to process all of the image files.

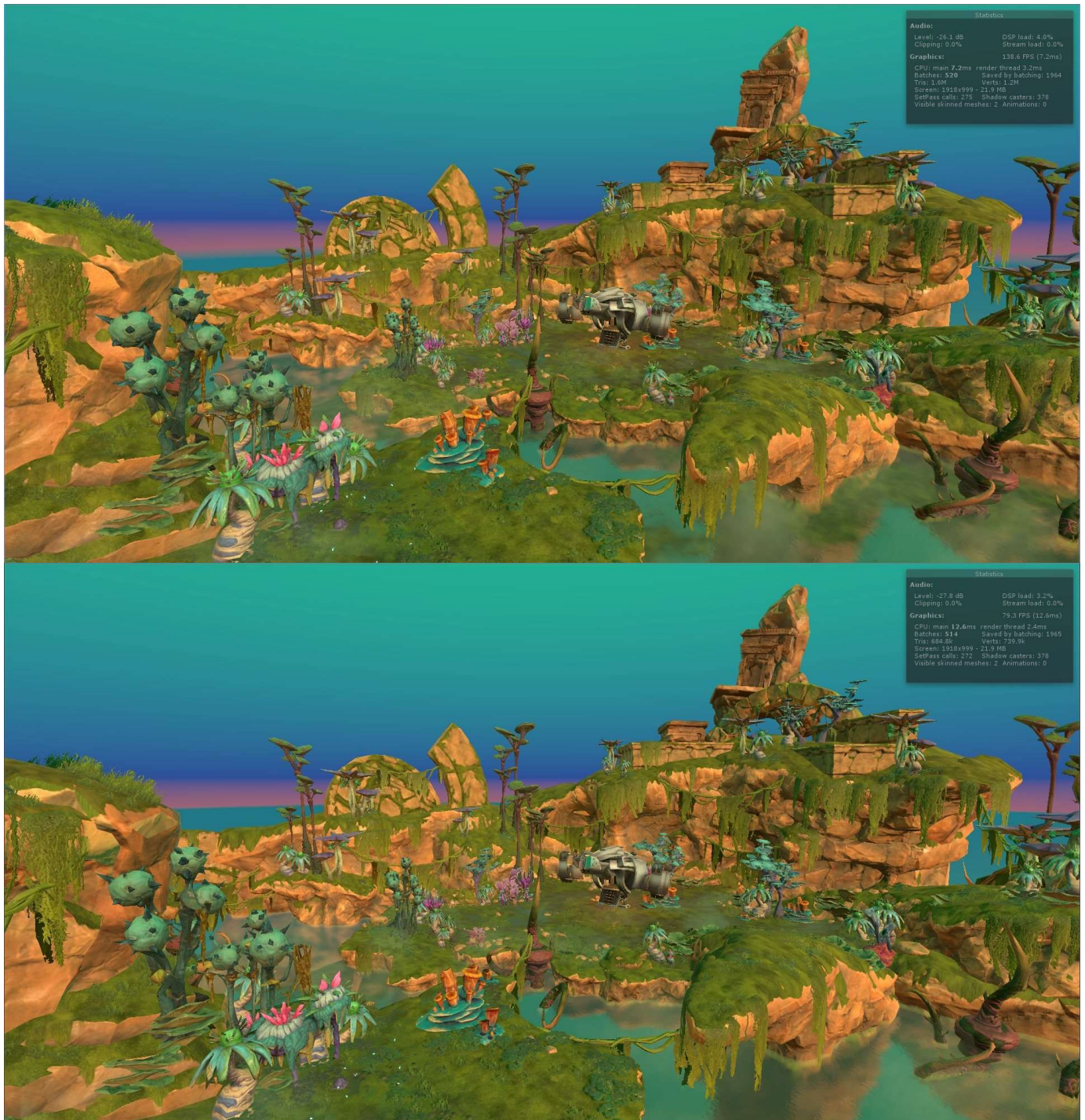


*The resizing Photoshop action.*



*A comparison of a corner of one of the vegetation textures before (left) and after (right) the size reductions.*



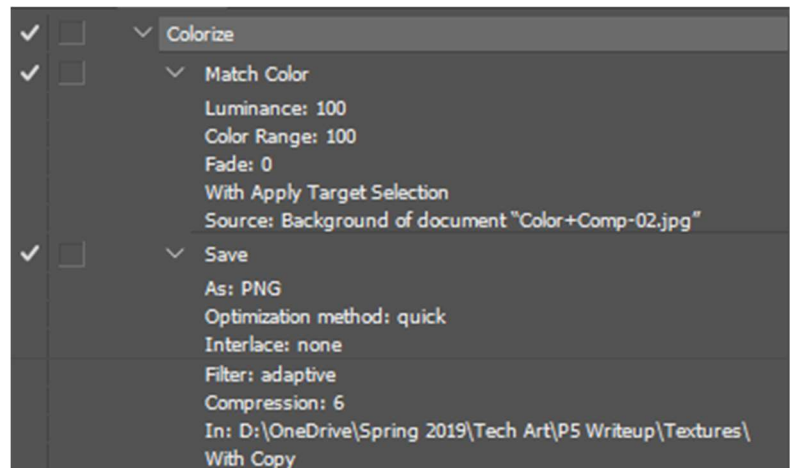


*A comparison of before (top) and after (bottom) the texture reductions.*

## Palletizer

The recoloring was done with 3 separate actions in Photoshop, one for each of the chosen images used to recolor the albedo textures. The actions start with the use of the match color adjustment, each action's source image within the match color options is tied to the reference images which are already open within photoshop.

The actions end with a save, just in case something goes wrong afterwards. The actions are then each run with the automation > batch feature in Photoshop to process whole directories. Ideally, for all of these actions, there would be separate folders for each action with all of the images you want to process due to the nature of the action's save overwriting the original files.

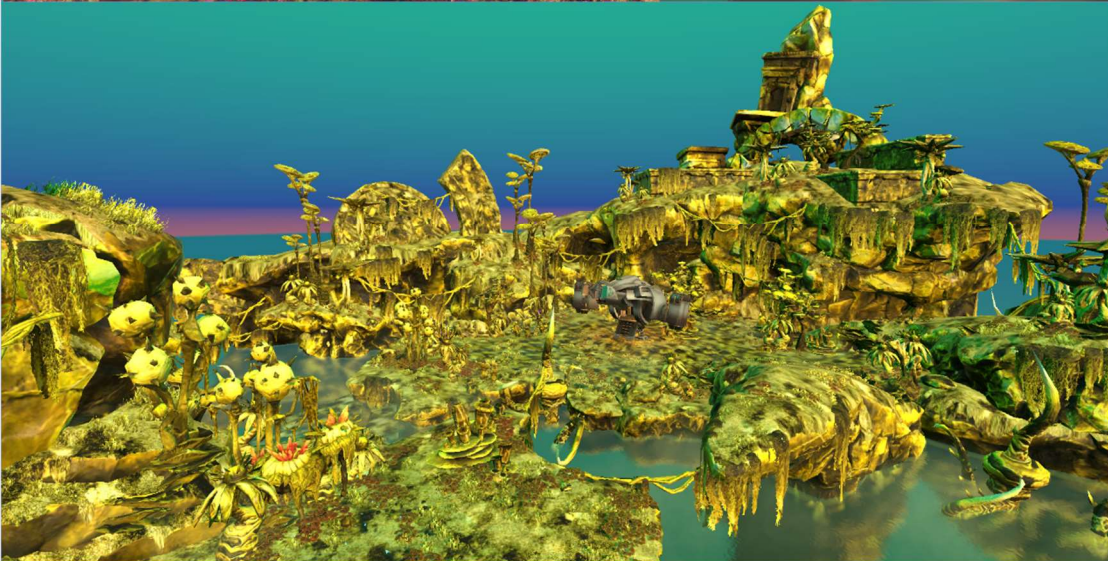
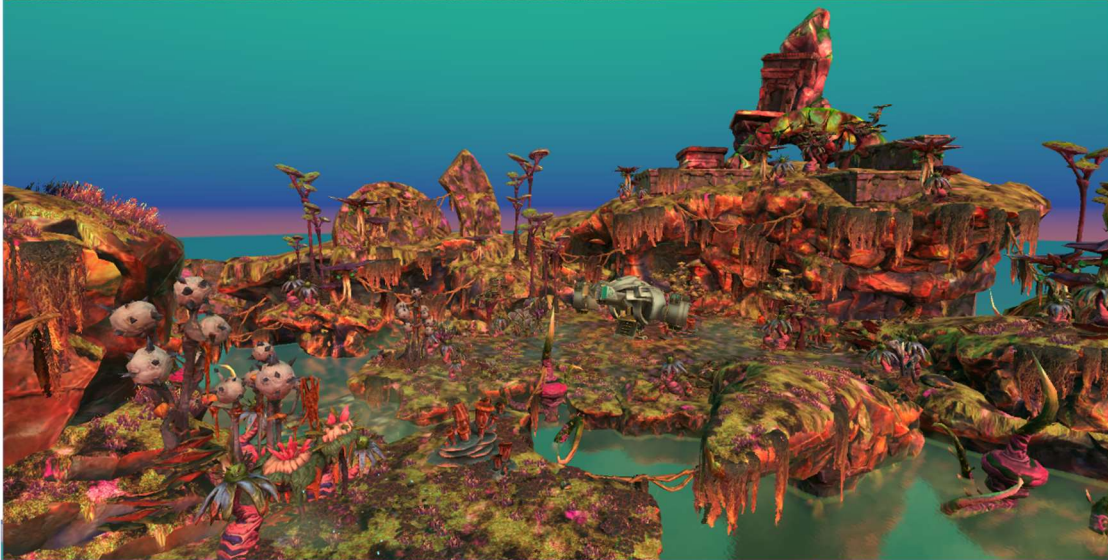
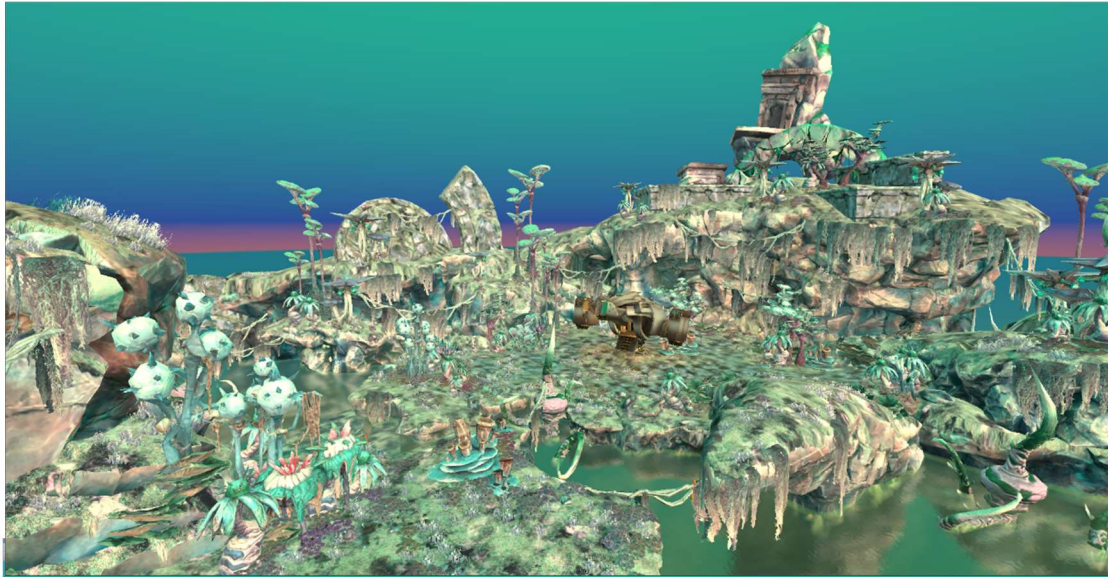


*One of the Photoshop palletizer actions, detailing all of the options chosen within.*



*The original scene with the reduced textures and models.*



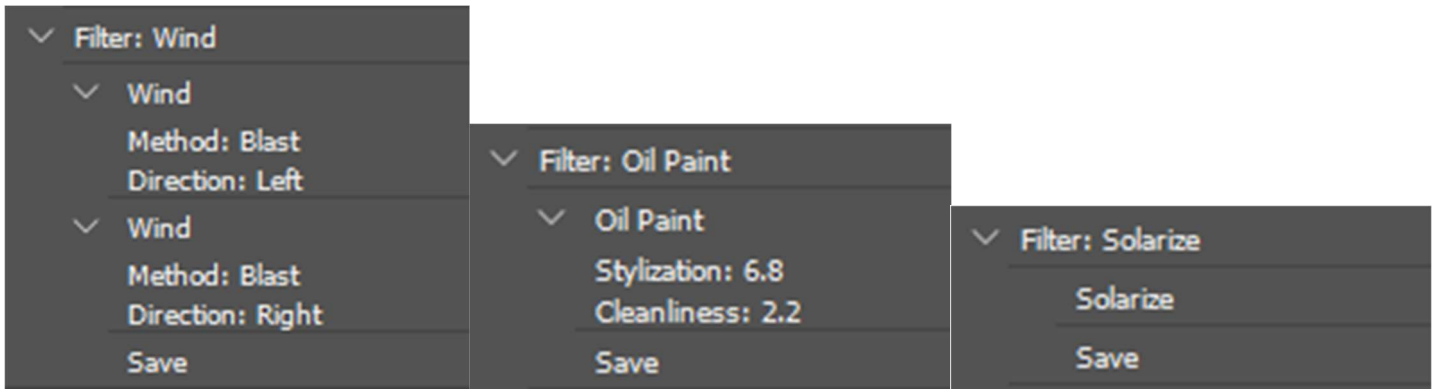


*The palletized scenes with reduced textures and models.*



## Stylizer

The stylizing was also handled with Photoshop actions, 3 separate actions again for each filter, and batch automation. The 3 filters chosen were wind, oil paint, and solarize for the distinctiveness and ability to showcase the effect easily. The wind filter was run twice, each time in a different direction for a less subtle look due to the lack of adjustment of the strength of the wind. The oil paint filter was cranked up higher for the same reason. Each end with a save, just in case. Batch automation is then used in the same way as the palletizer.



*The 3 actions detailing the filters used.*



*The original scene with reduced textures and models.*





*The stylized scenes with reduced textures and models (from first to last:  
wind, solarized, oil paint).*



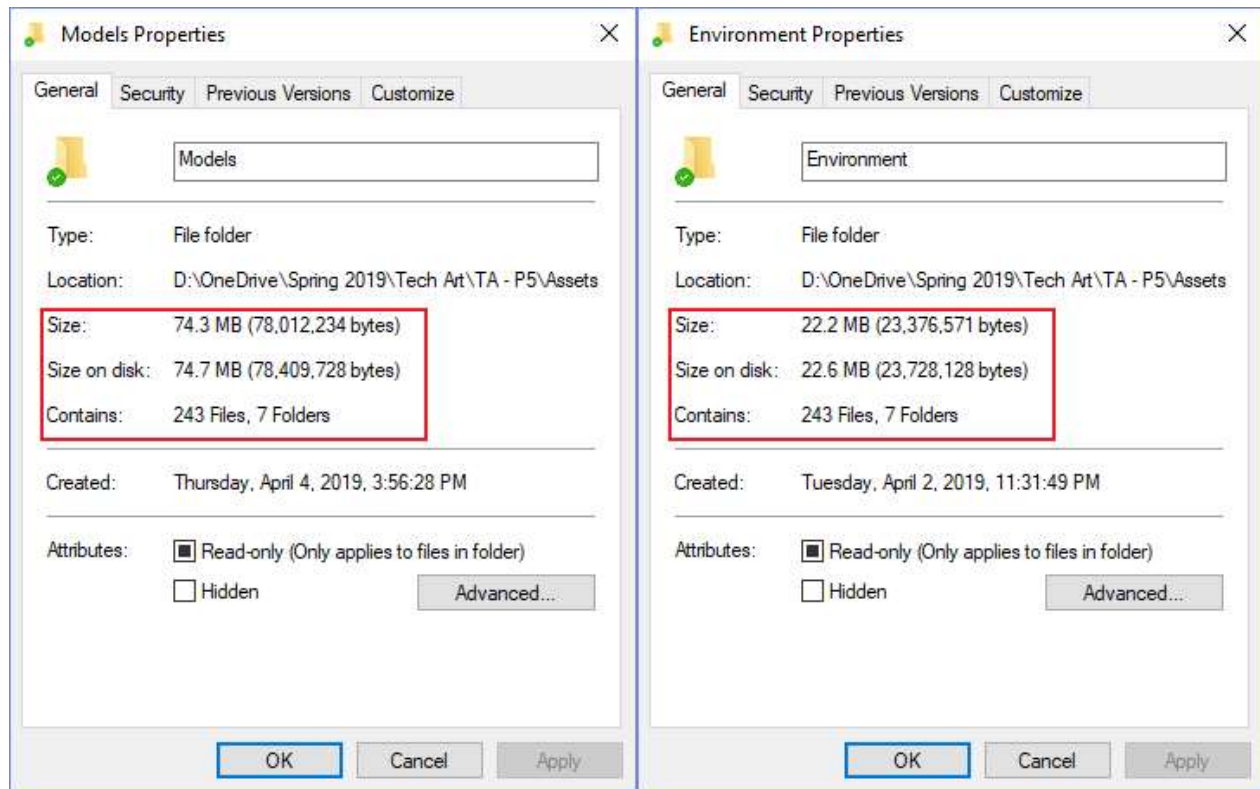
## Performance and Sizes

The texture reductions and poly reductions are significant, with GFX usage going from .95 GB to .52 GB and texture memory going from 1.39 GB to .94 GB. The texture folder size went from 782 MB to 230 MB and the model folder size went from 74.3 MB to 22.2 MB.

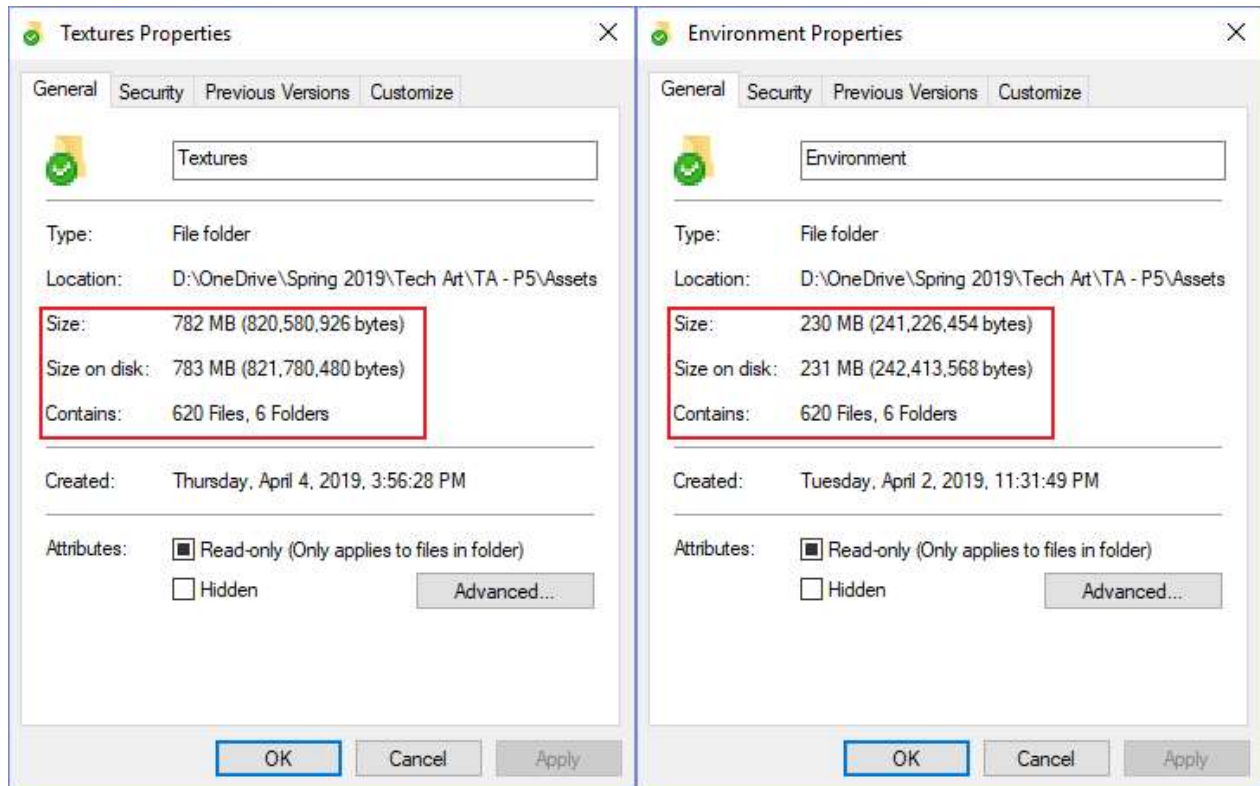
Recording was handled with OBS and editing/comping was done with Premiere Pro.

Simple	Used Total: 3.44 GB	Unity: 2.30 GB	Mono: 20.6 MB	GfxDriver: 0.95 GB	FMOD: 158.7 MB	Video: 0 B	Profiler: 11.0 MB
	Reserved Total: 3.83 GB	Unity: 2.67 GB	Mono: 34.0 MB	GfxDriver: 0.95 GB	FMOD: 158.7 MB	Video: 0 B	Profiler: 16.0 MB
	Total System Memory Usage: 4.95 GB						
	Textures: 1032 / 1.39 GB						
	Meshes: 2397 / 1.36 GB						
	Materials: 191 / 508.0 KB						
	AnimationClips: 88 / 18.9 MB						
	AudioClips: 412 / 155.8 MB						
	Assets: 11231						
	GameObjects in Scene: 19372						
	Total Objects in Scene: 78097						
	Total Object Count: 89328						
	GC Allocations per Frame: 564 / 29.0 KB						
Simple	Used Total: 2.30 GB	Unity: 1.59 GB	Mono: 20.3 MB	GfxDriver: 0.52 GB	FMOD: 158.7 MB	Video: 0 B	Profiler: 11.1 MB
	Reserved Total: 2.78 GB	Unity: 2.06 GB	Mono: 34.0 MB	GfxDriver: 0.52 GB	FMOD: 158.7 MB	Video: 0 B	Profiler: 16.0 MB
	Total System Memory Usage: 3.90 GB						
	Textures: 1052 / 0.94 GB						
	Meshes: 2323 / 0.66 GB						
	Materials: 195 / 511.0 KB						
	AnimationClips: 88 / 18.9 MB						
	AudioClips: 412 / 155.8 MB						
	Assets: 11258						
	GameObjects in Scene: 19372						
	Total Objects in Scene: 78027						
	Total Object Count: 89285						
	GC Allocations per Frame: 588 / 30.0 KB						

*Before (top) and after (bottom) comparison of the texture memory usage using Unity's profiler.*



*A comparison of the size reduction for the models, before (left) and after (right).*



*A comparison of the size reduction for the textures, before (left) and after (right).*