

Using Xgen as a Population method



Oluwaseun Ademoye

Fall 2016

Overview

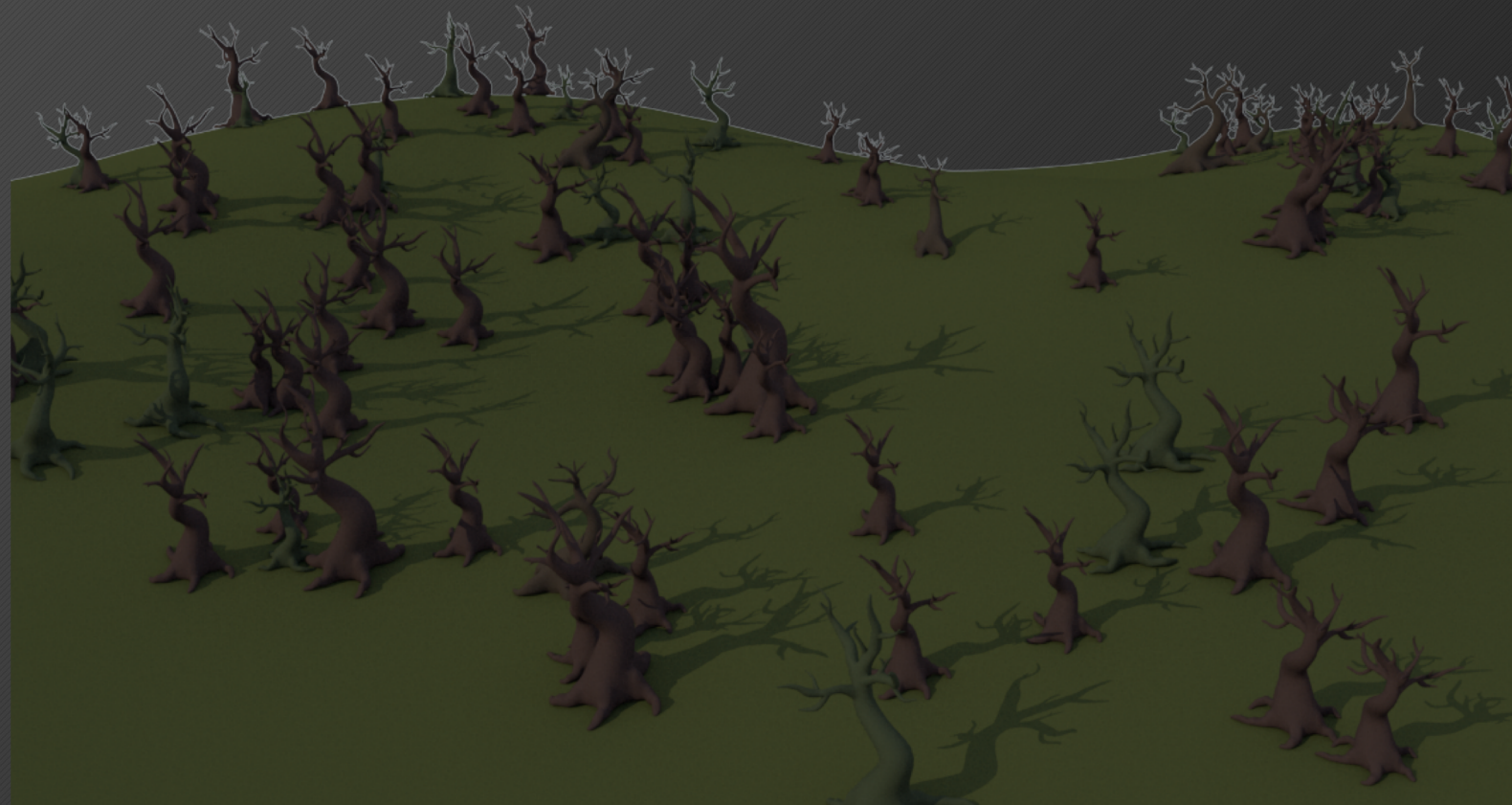


Xgen lets you populate the surface of a mesh with geometry either uniformly or randomly. The plug in allows for the control of the disbursement with attributes and expressions that give more direct control over the desired effect.

“For set dressing, XGen lets you quickly populate large-scale environments, including grass savannas, forests, rocky landscapes, and debris trails”

-Autodesk Knowledge Network

For this paper we will be using 3 tree models to quickly populate a forest



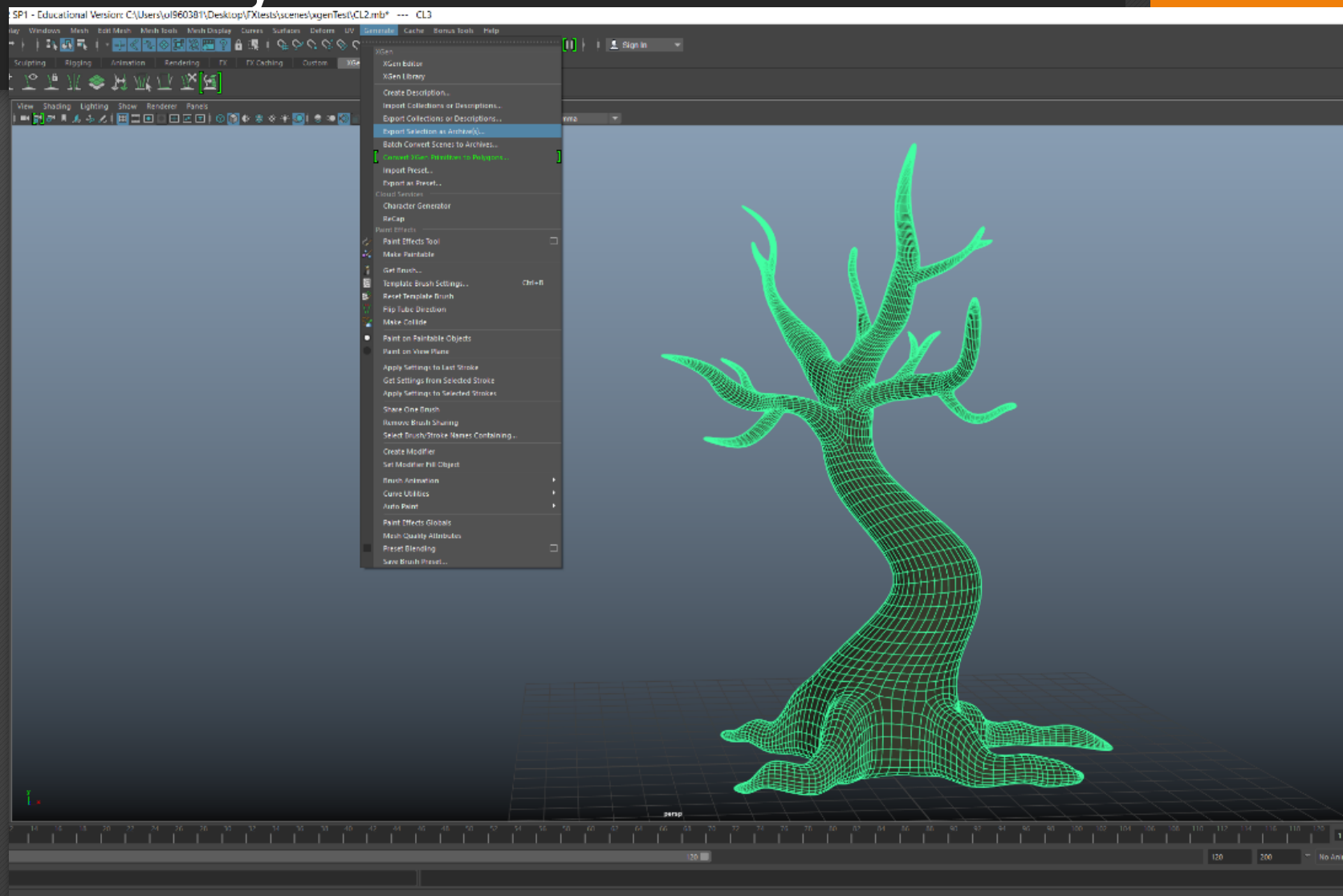
Get your models ready

The first step for XGen is to prepare the models you want to use to populate your surface.

Export your models as archives to your local folder either one at a time or multiple at once

- Generate > Export selection as Archive
- Generate > Batch Convert scenes to archives

YOUR MODEL FILES SHOULD BE SAVED AS MAYA BINARY





Export Selection as Archive(s)



Destination Folder: Local Archives (\${PROJECT}/xgen/archives) ▼

C:/Users/ol960381/Desktop/FXtests/xgen/archives



Archive Name: forestTree1

Level of Detail: Disabled ▼

Include Animation: ☐

Help...

Export

Cancel



Batch Convert Scenes to XGen Archives



Scenes to be Converted:

Browse...

Remove

C:/Users/ol960381/Desktop/FXtests/scenes/xgenTest/CL2.mb
C:/Users/ol960381/Desktop/FXtests/scenes/xgenTest/CL3.mb
C:/Users/ol960381/Desktop/FXtests/scenes/xgenTest/CLE1_1.mb

Destination Folder: Local Archives (\${PROJECT}/xgen/archives) ▼

C:/Users/ol960381/Desktop/FXtests/xgen/archives



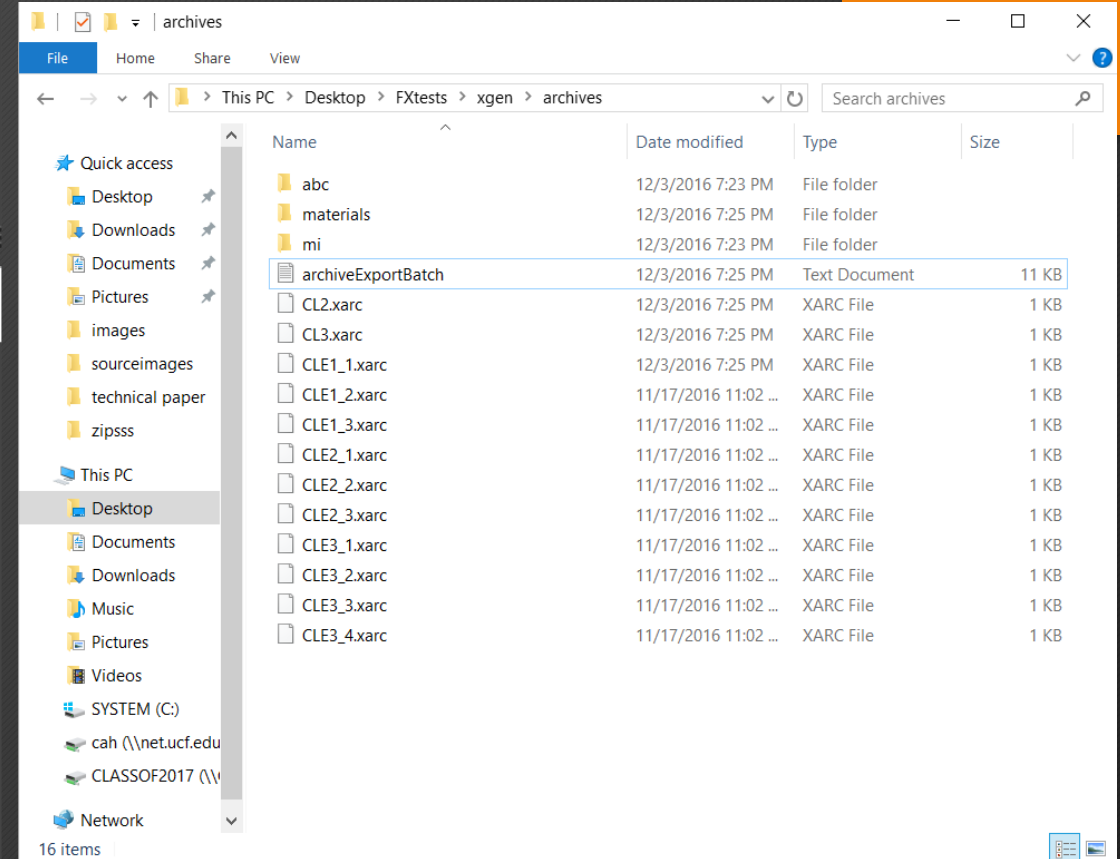
Level of Detail: Disabled ▼

Include Animation: ☐

Help...

Convert

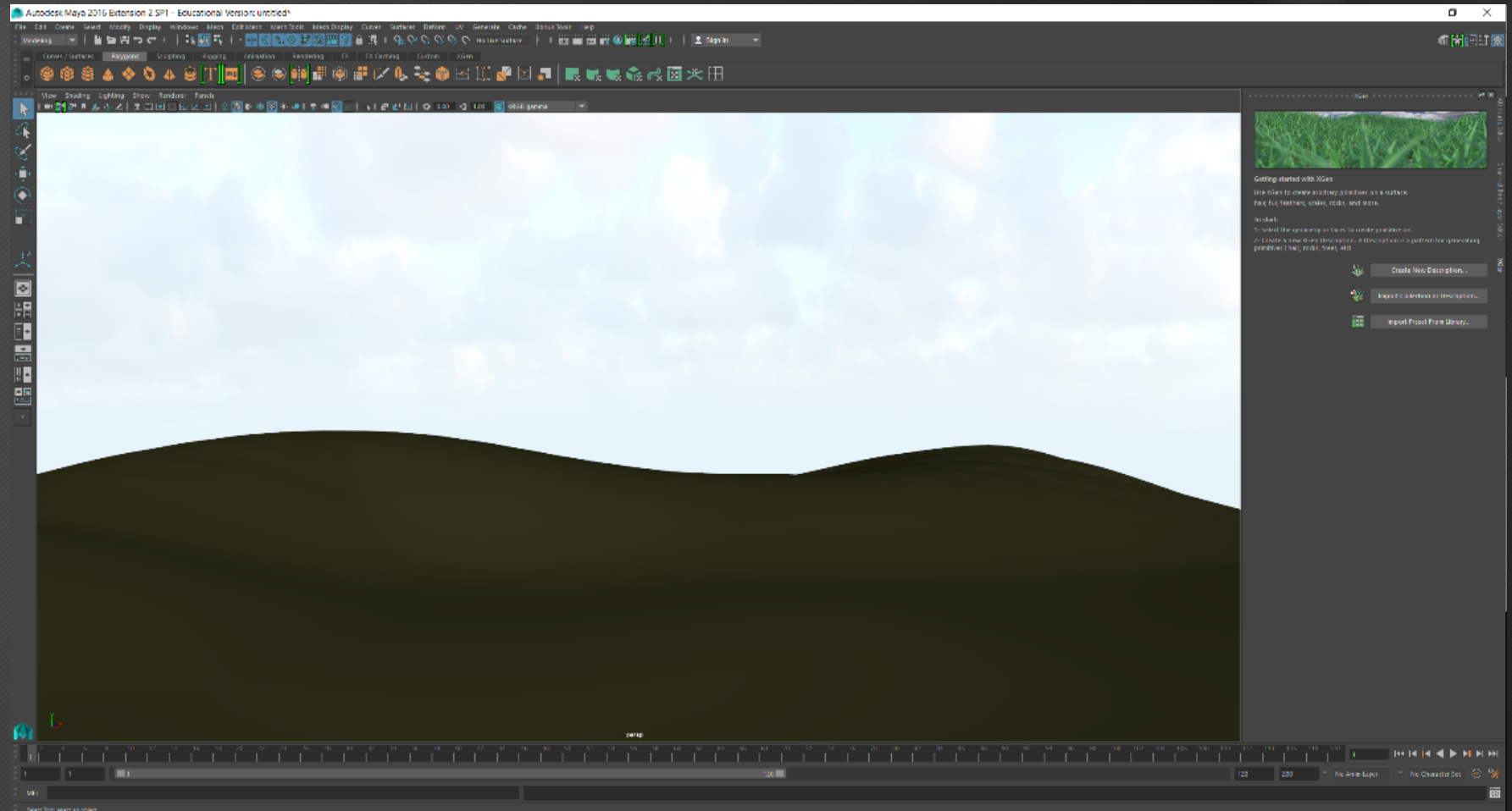
Cancel



Browse to the location of your archives

Getting Your Scene Ready

- Create a description
 - Select the mesh you want to populate and click “create description” in the XGen menu
- The name of the collection should be appropriate to what you’re making



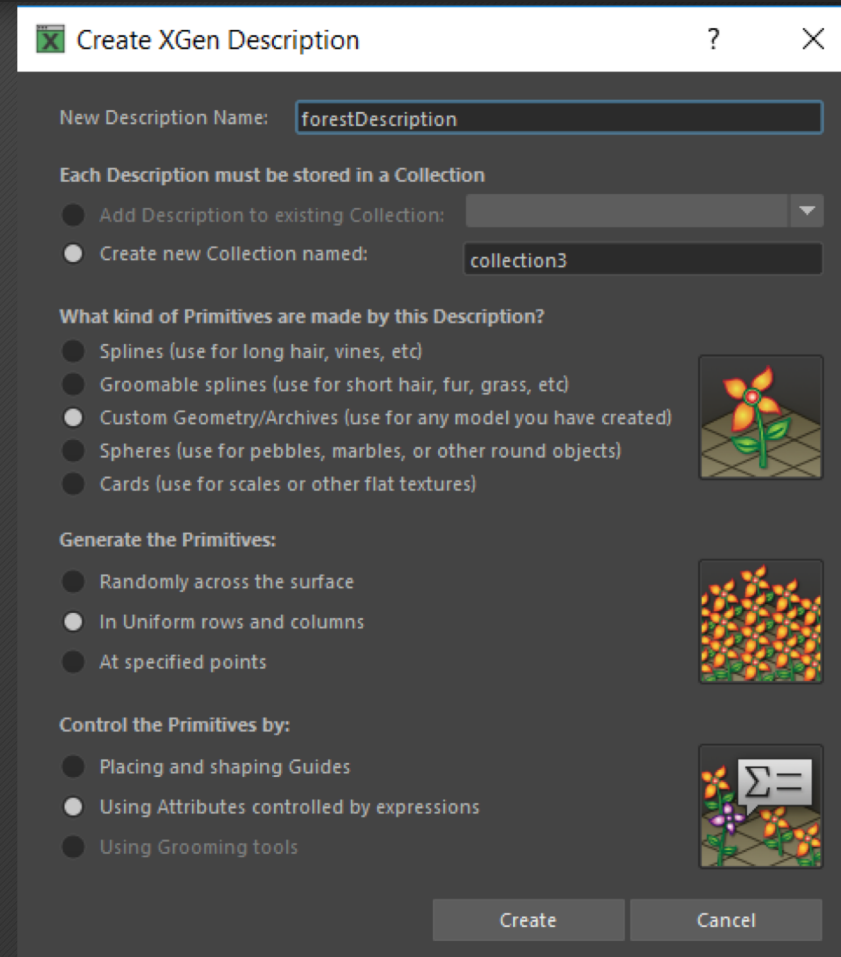
Getting Your Scene Ready

Select the type of primitives used for the description

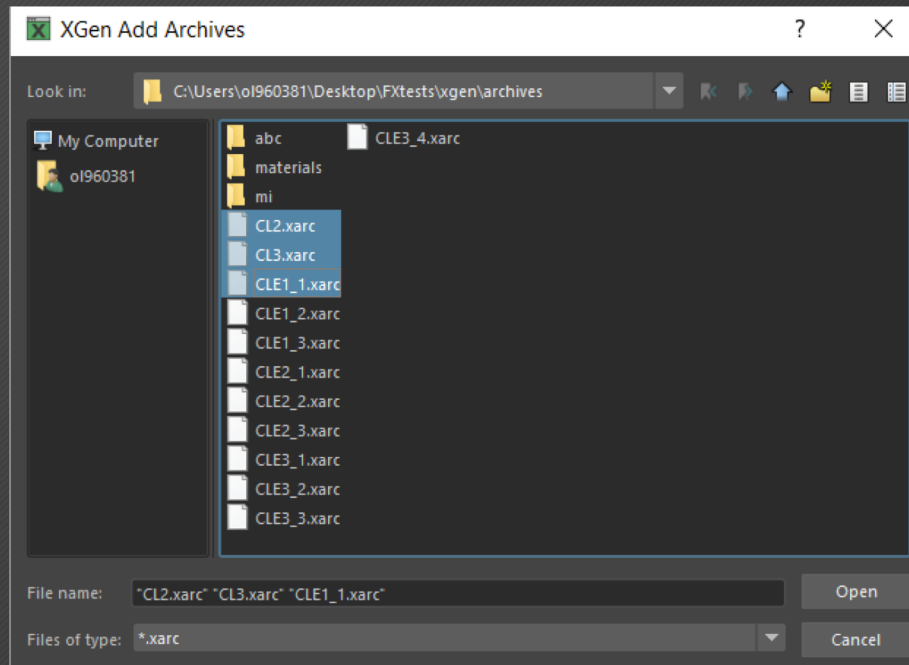
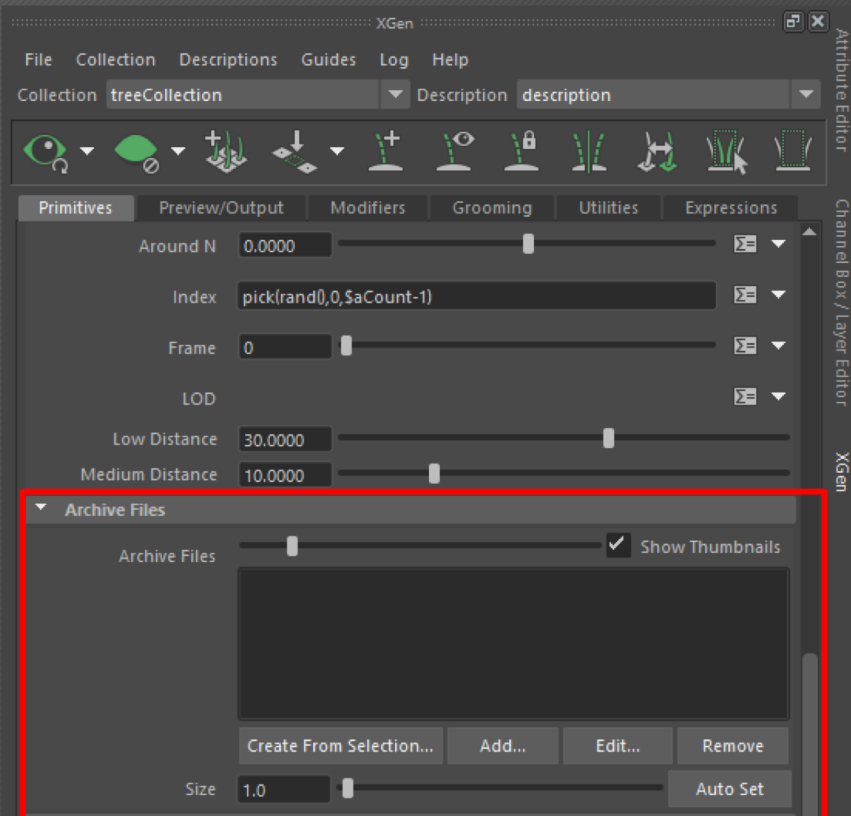
- for the purpose of this paper, we will be using Custom Geometry

Then we will choose to generate the primitives: “In uniform rows and column” and to control the primitives by: “Using attributes controlled by expressions”

- these options can be altered later

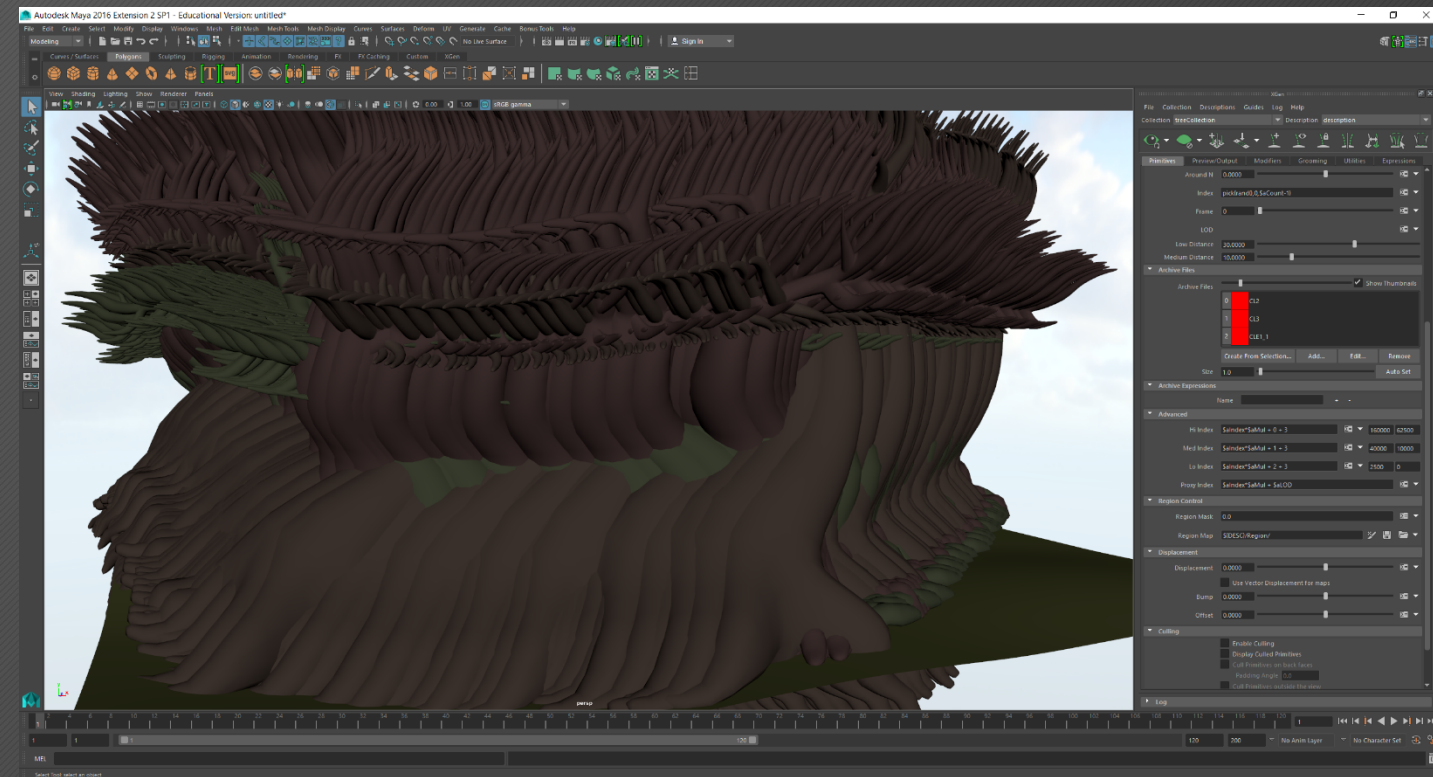


Adding Objects to the Description



- Under the “Primitives” tab in the XGen menu, scroll down to Archive Files and select “Add...”
- Navigate to the “xgen” folder in your local file directory and select the archives you wish to add

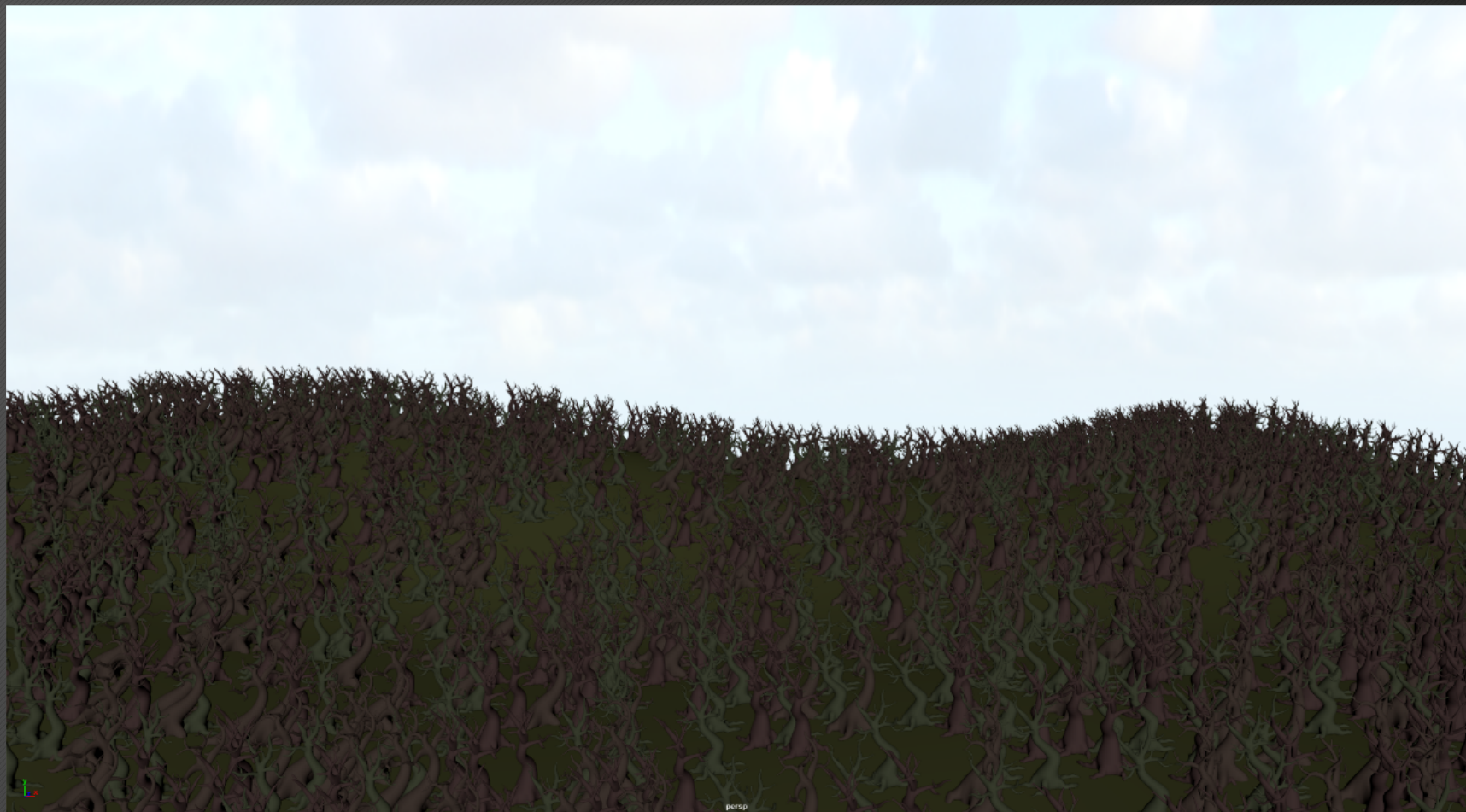
Start of your Populated Mesh



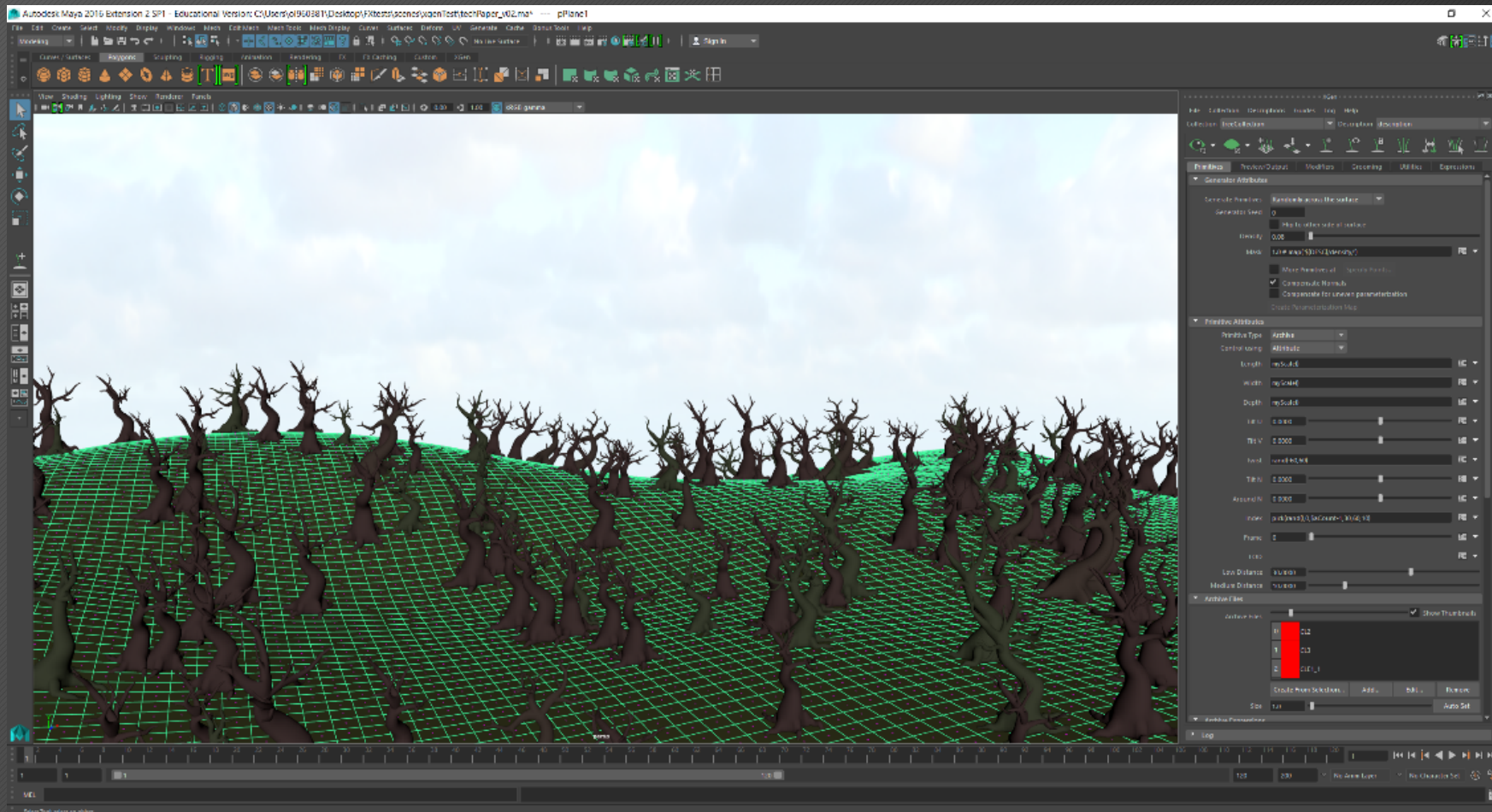
The generator attributes start with the primitives in uniform rows and columns. The spacing attribute slider will adjust how close the models are to each other; the higher the number, the further apart they'll be from each other and the fewer models will be placed in the scene.

If we change the generator to be random across the surface, the control comes from the density attribute. The higher the density, the more objects there will be.

Start of your Populated Mesh

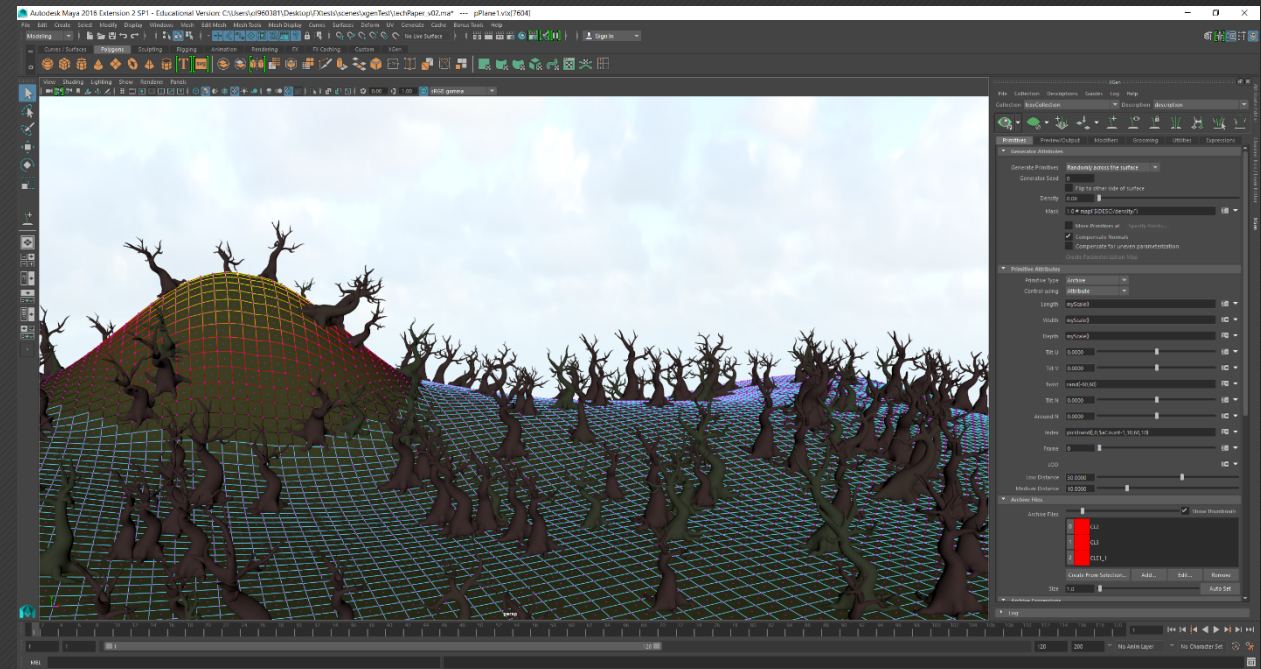
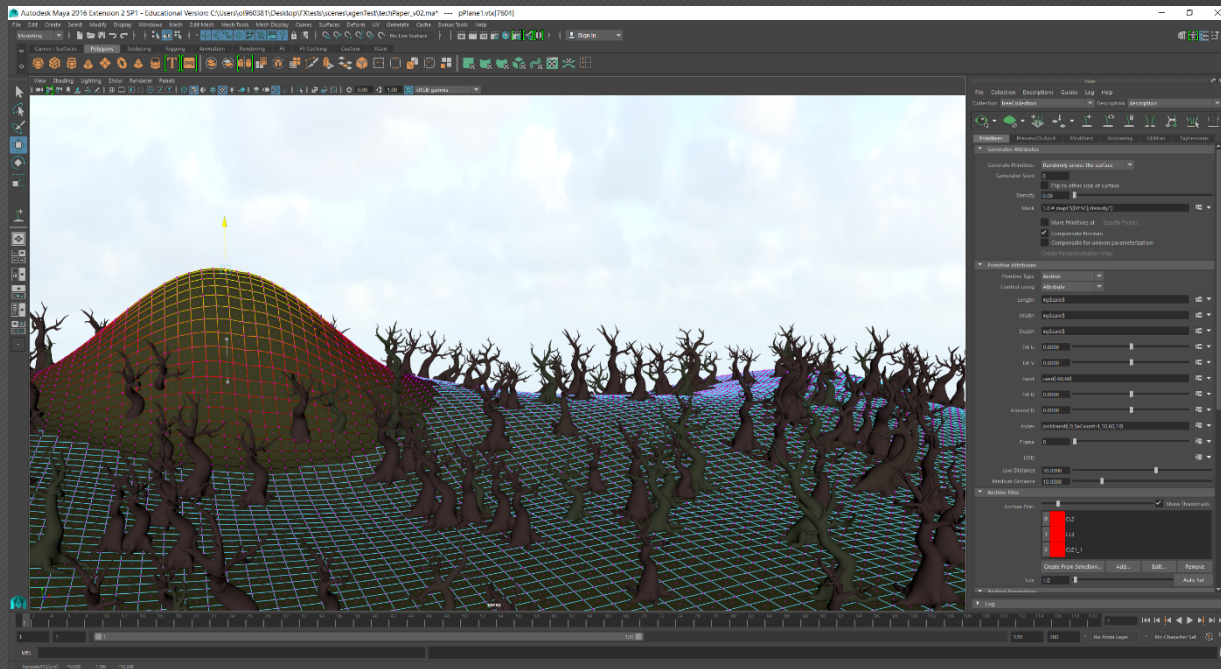


Start of your Populated Mesh



Let's say somewhere along the line you decide to edit the base mesh you already started populating with geometry. Since XGen uses instanced geometry to populate a mesh, the mesh can be changed and then repopulated without having to start the process over

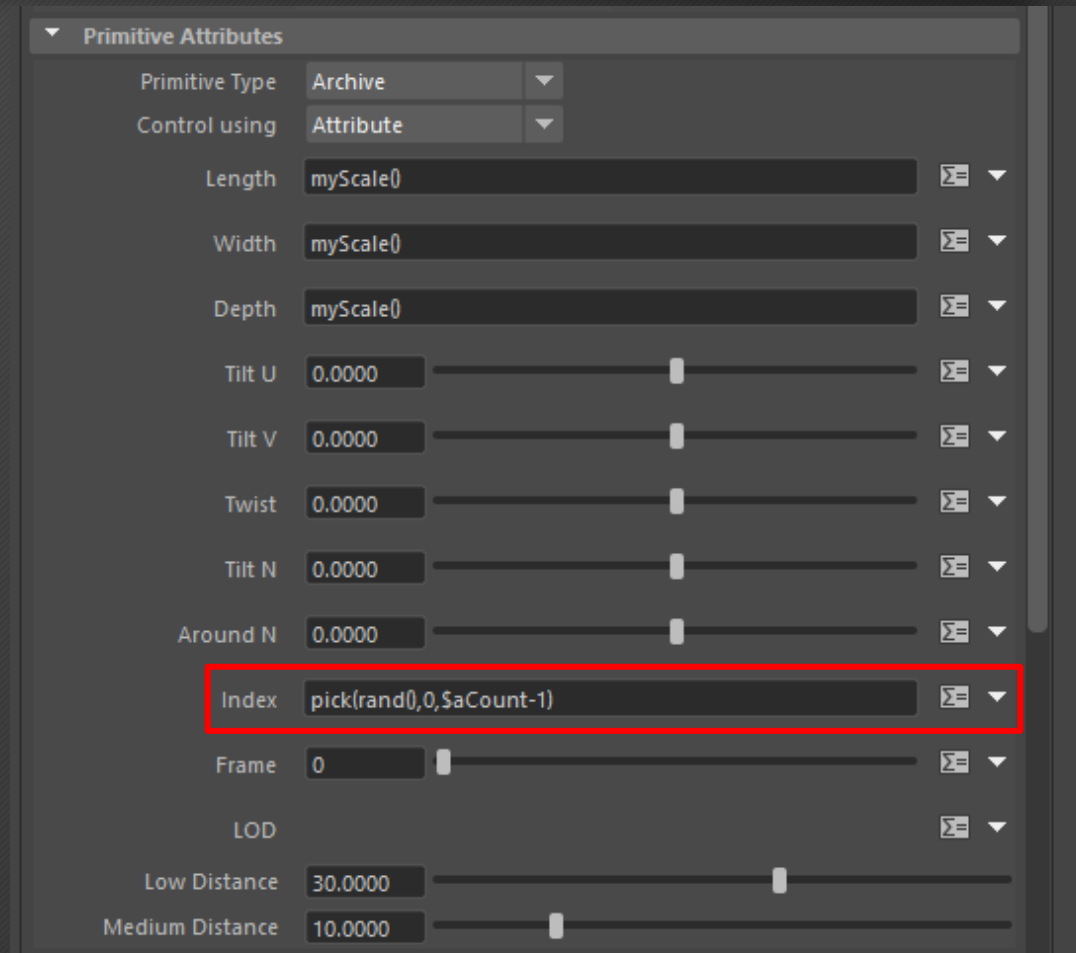
Start of your Populated Mesh



Changing the Distribution of Models

Depending on the number of models you archive in to the scene, by default XGen will evenly distribute the models. Since we have 3 models, each one will be generated 33.3% of the time.

To change this, we add values to the Index attribute under the Primitives tab.



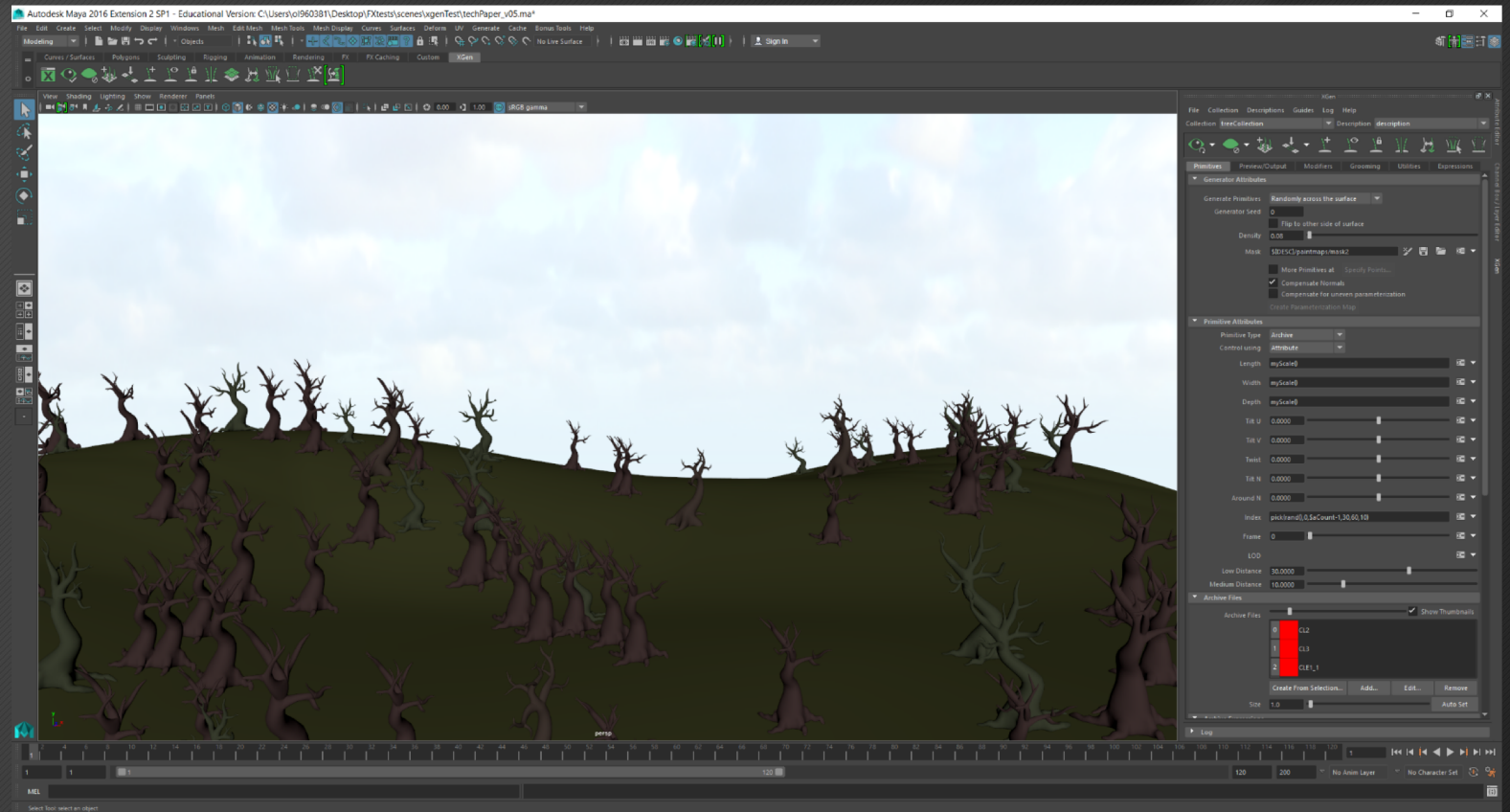
Changing the Distribution of Models

Adding values to the end of the pick() expression changes how often a model is generated. So if we write:

```
pick(rand(),0,SaCount-1,60,20,20)
```

the first model on the list will show up 60% of the time, and the other 2 will show up 20% of the time

Playing with these numbers lets us better define how we want our scene to be populated with our chosen models



Controlling the Scale of your Models

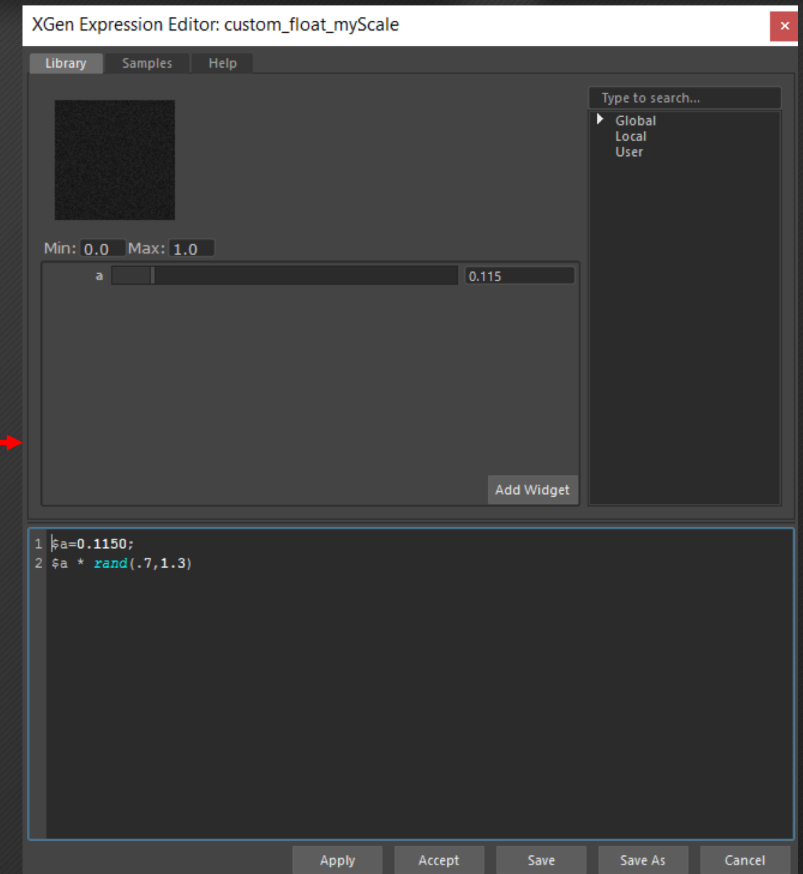
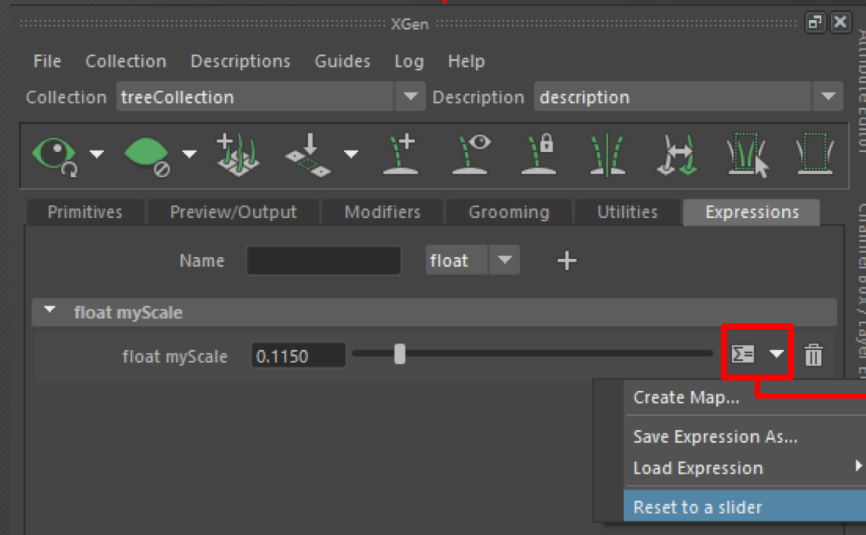
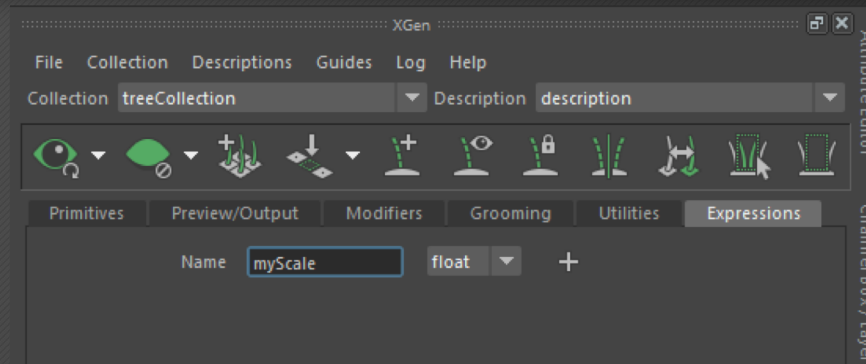
Typically you'll want to control the scaling of your models uniformly rather than using the 3 preset attributes of Length, Width, and Depth separately. To do so, we create a new expression for the uniform scale under the Expressions tab

- We have the option to set the expression to a slider for easier control

To give the scale of our models more randomness, multiply by a random value within a range with the following expression added to our myScale expression:

$a\$ * \text{rand}(.7, 1.3)$

- $a\$$ is the slider of the scale
- $\text{rand}()$ is the random expression
- $(.7, 1.3)$ are the range of values

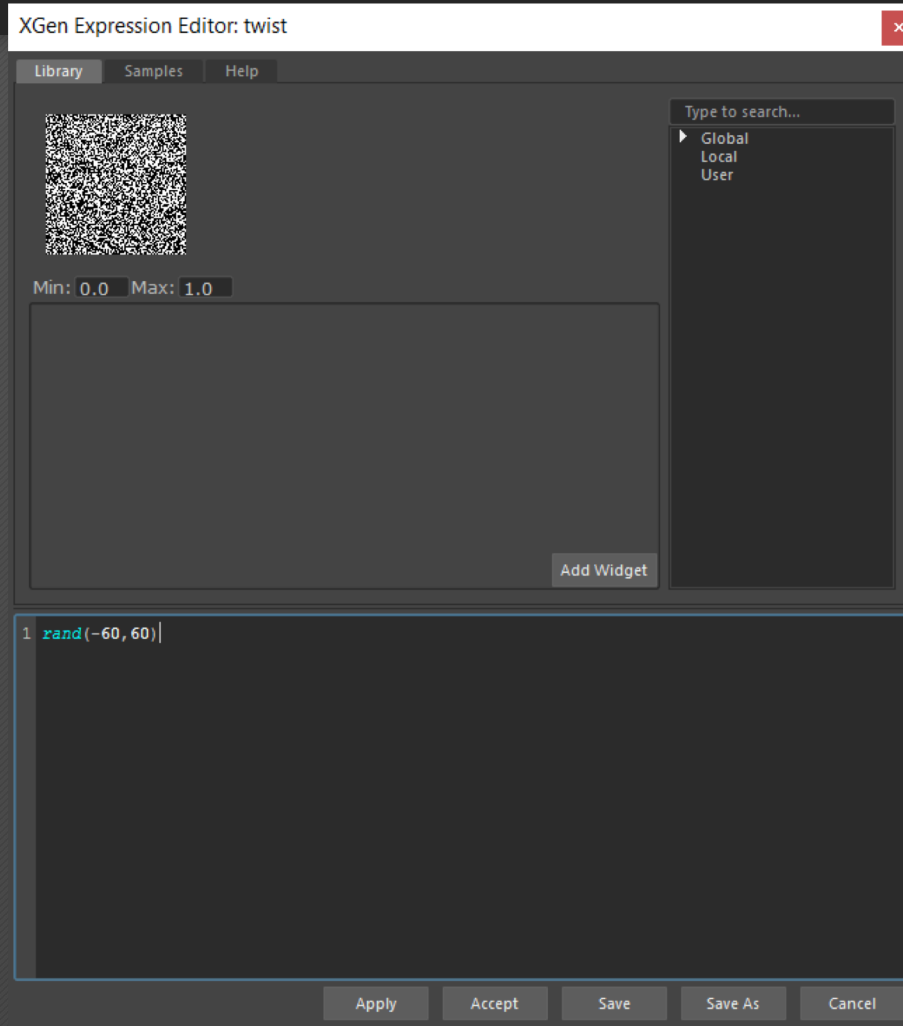


Controlling the Scale of your Models

Now the scale of our models has been randomized between 70% and 130% of the models' original size



Varying the Orientation of your Models



As you've probably noticed, the models that are generated are all oriented in the same direction. This can be easily fixed by adding a random expression to the Twist attribute.

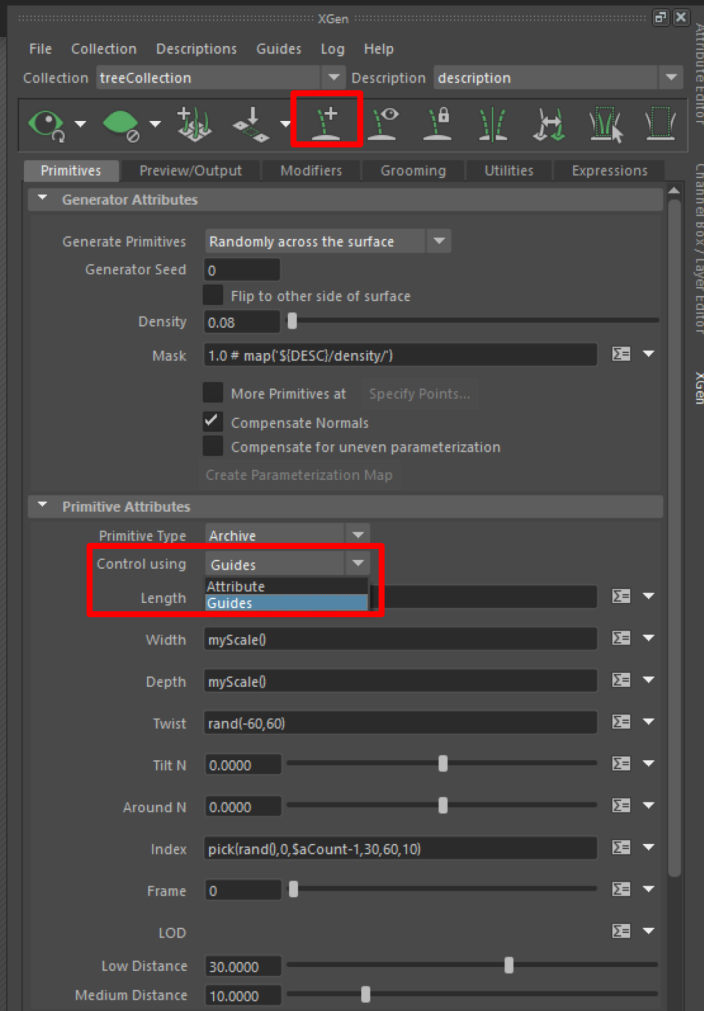
With the expression `rand(-60,60)` each tree is randomly rotated between -60 and 60 degrees

Using Guides

Another level of control over distribution can be achieved using guides. Guides can be used independently or in conjunction with expressions. Placing guides makes it easier to visualize the customization of disbursement amongst your scene. The guides can easily be manipulated using basic transforms.

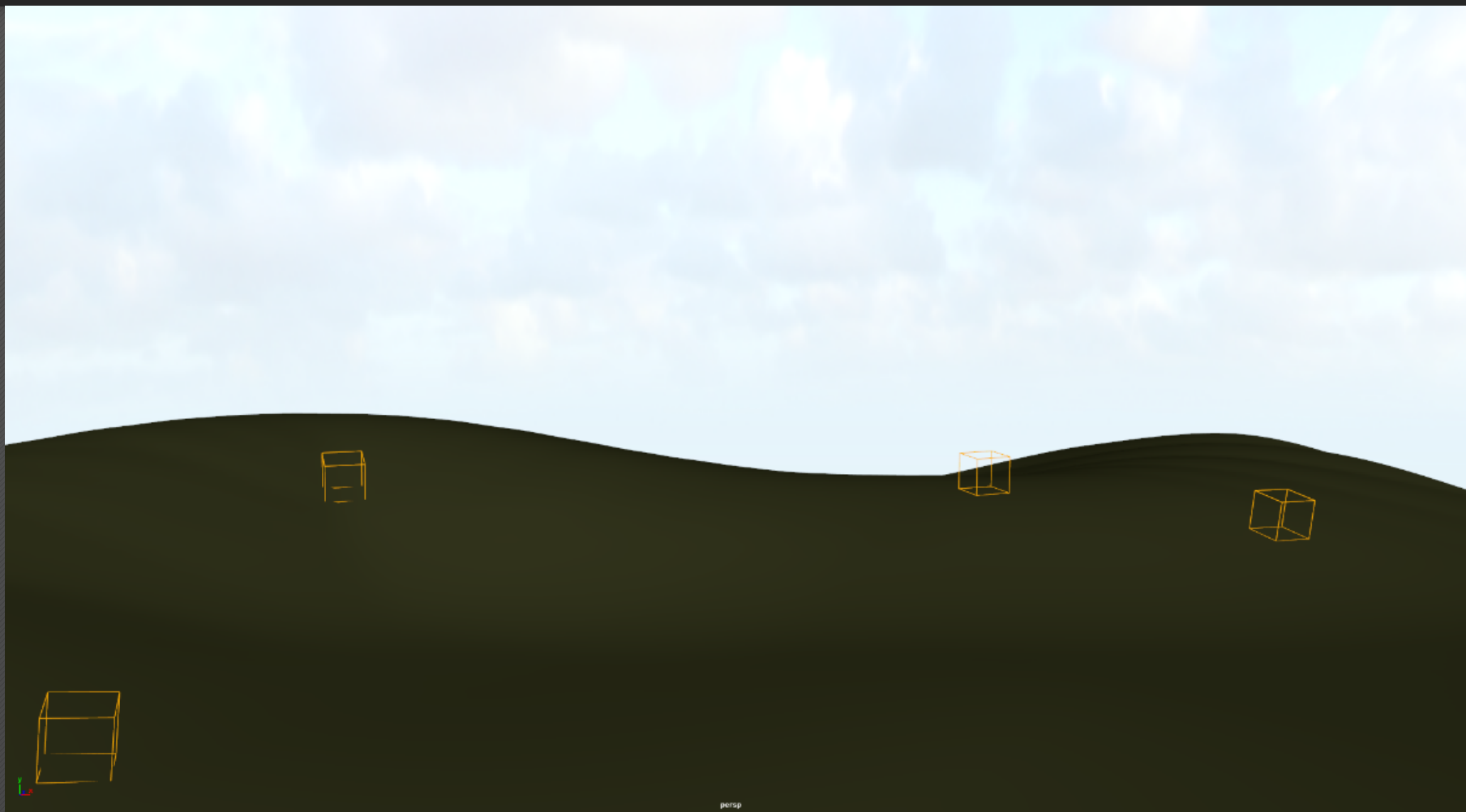
Like I mentioned before, the guides can be used in conjunction with expressions. So the expressions that we created for both the scale and rotation of the models is compounded with the transforms of the guides. The guides are also be affected by deformers to give more interesting transformations.

Using Guides



- You first have to change the primitives control type to using guides instead of attributes in order to place your guides.
- Then you can select the Add/Move Guides tool and place guides on your surface mesh

Using Guides



Using Guides



Using Guides



Using Guides



The guides add their transform values to the pre-existing attributes and expressions

Using Guides



Using Guides

Guides can also be transformed using deformers

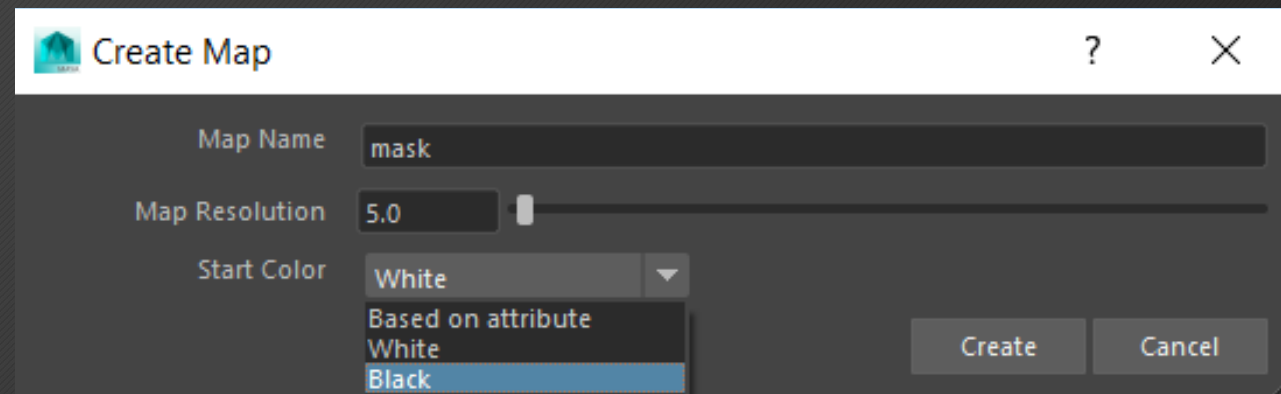
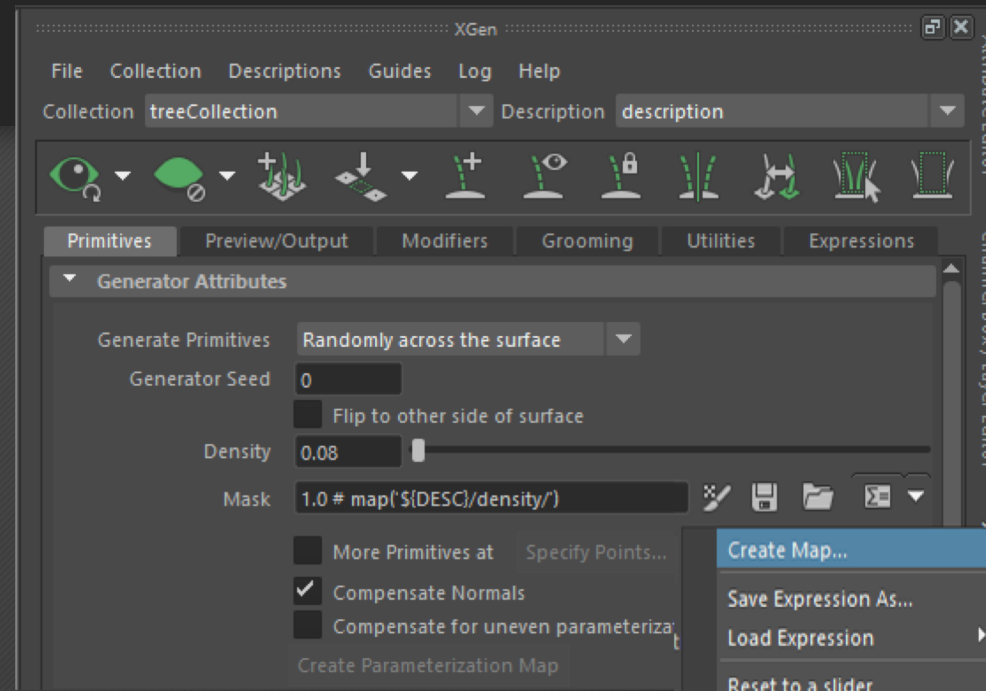
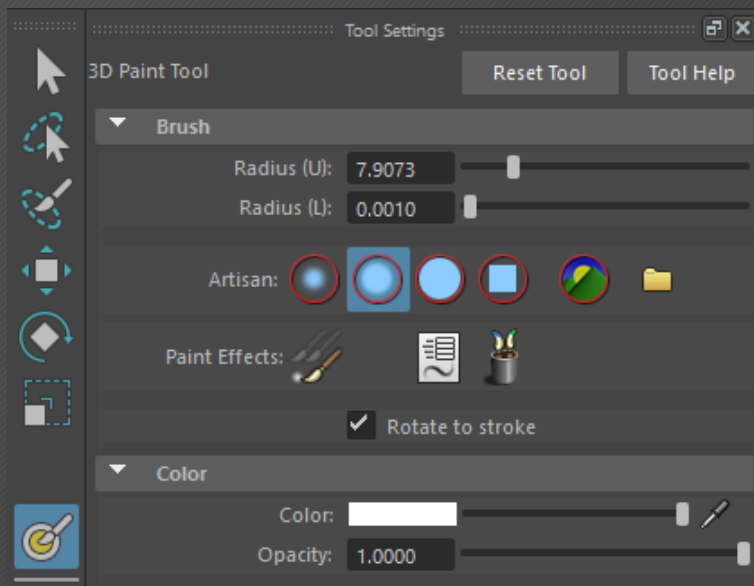


Using Masks

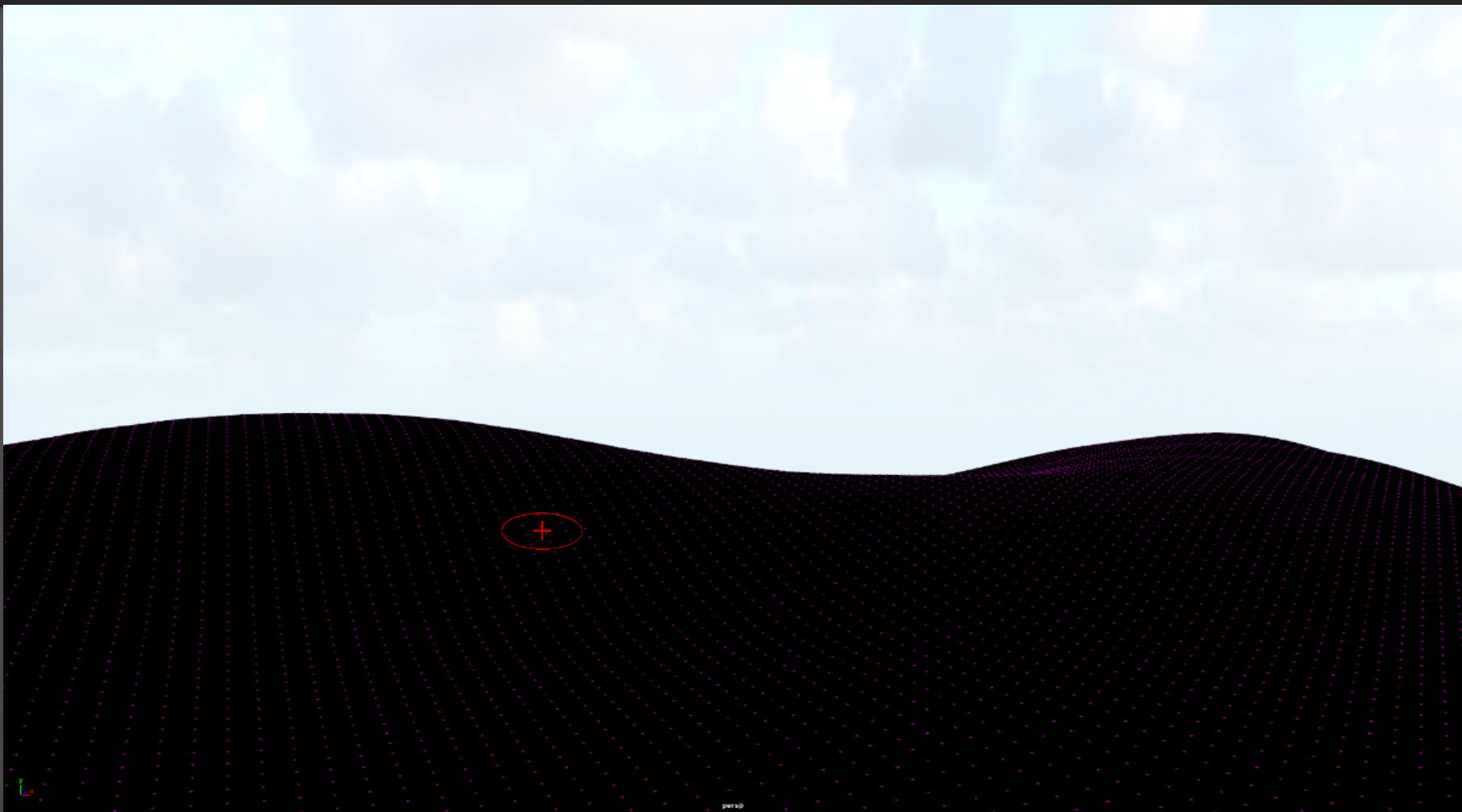
Another level of control in the populating of models on a surface comes from the use of masks. With masks you can hand paint paths and spots to populate with geometry. The mask attribute works along with the density attribute by either adding to it or removing from it. Since the mask uses per-face texture mapping, the surface being populated doesn't have to be UV'd beforehand. You do have the option to bring in a pre painted mask and use that instead of painting one in Maya

Using Masks

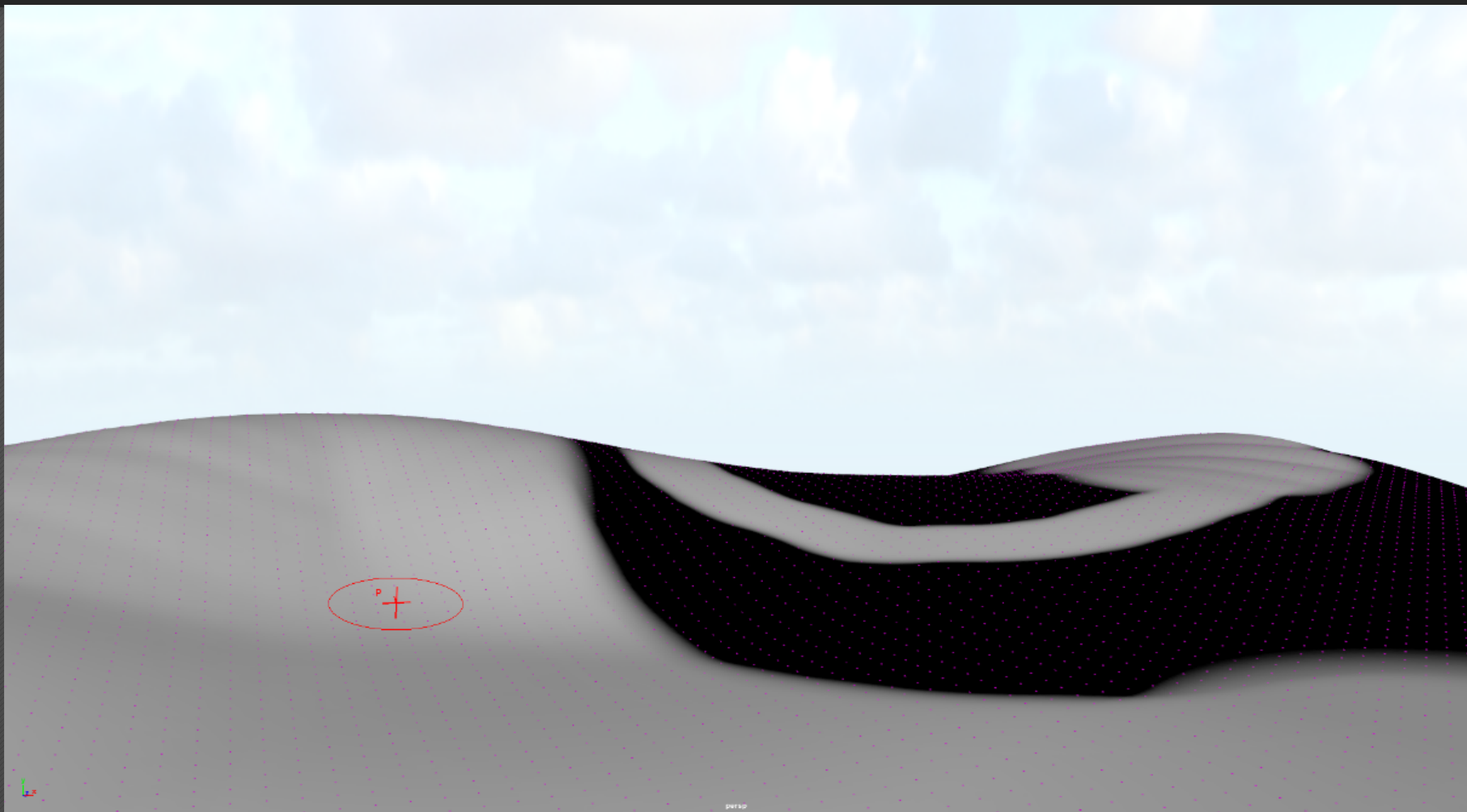
With the start color set to black, you can paint white onto the surface in the areas you want to populate. When you're satisfied with the look of the mask, you can hit the save button and repopulate the newly painted surface.



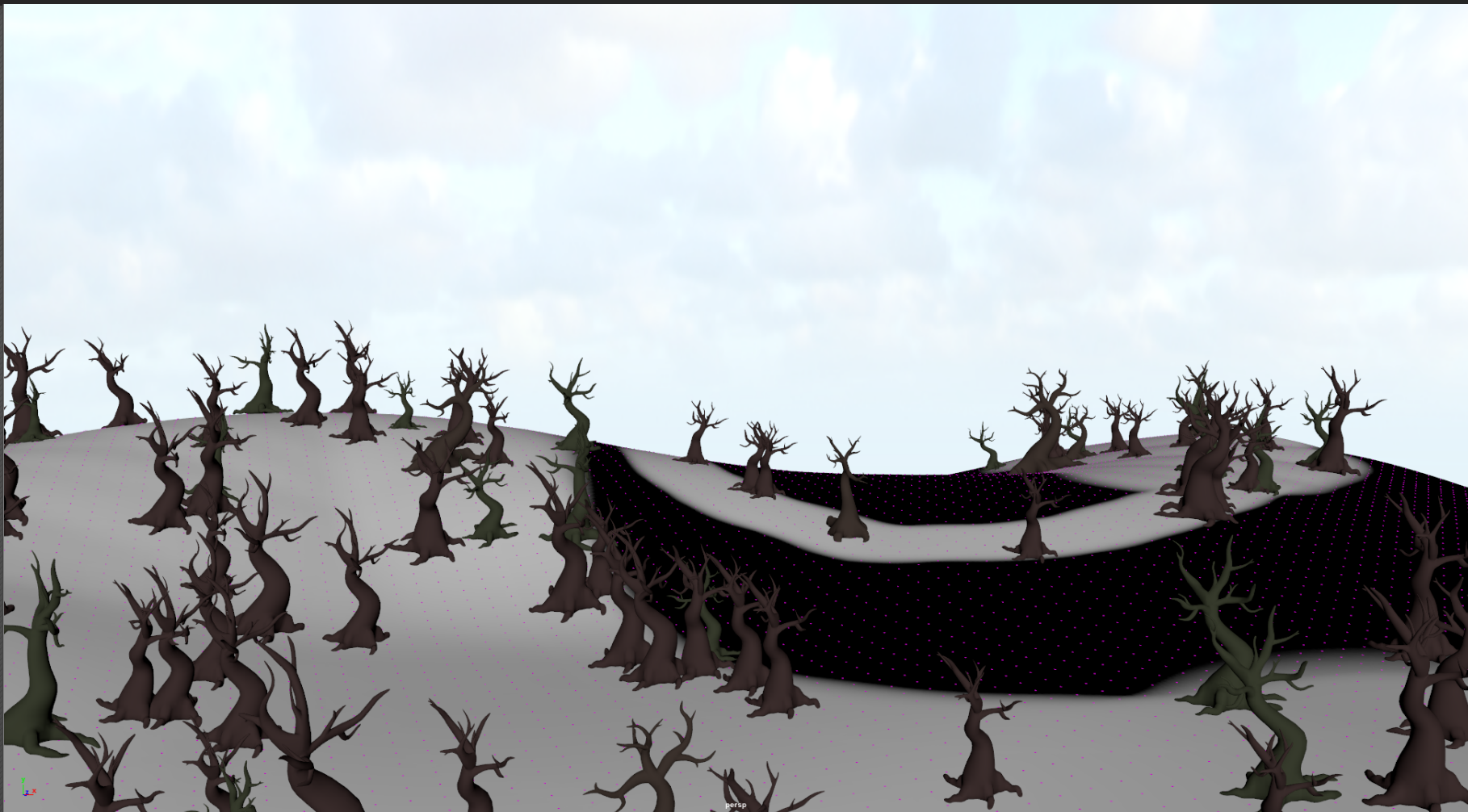
Using Masks



Using Masks



Using Masks

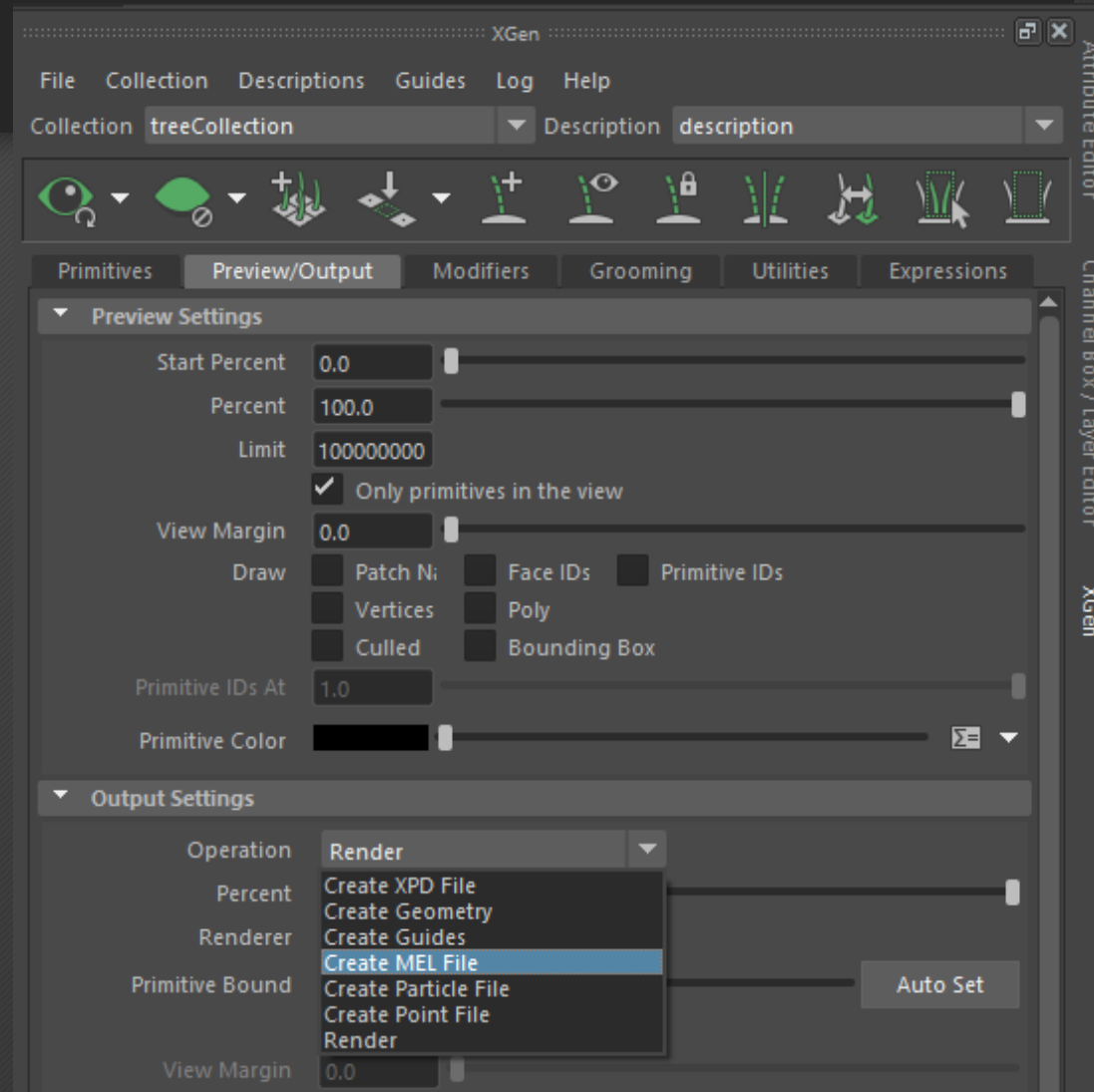


Finalizing your Scene

The final step in creating your populated xgen scene is to export the populated geometry. You'll notice that you can't actually select any of the objects that are populating your surface since it's all instanced geometry. So we need Maya to take all the transform information for all of the populated models and convert them into actual objects.

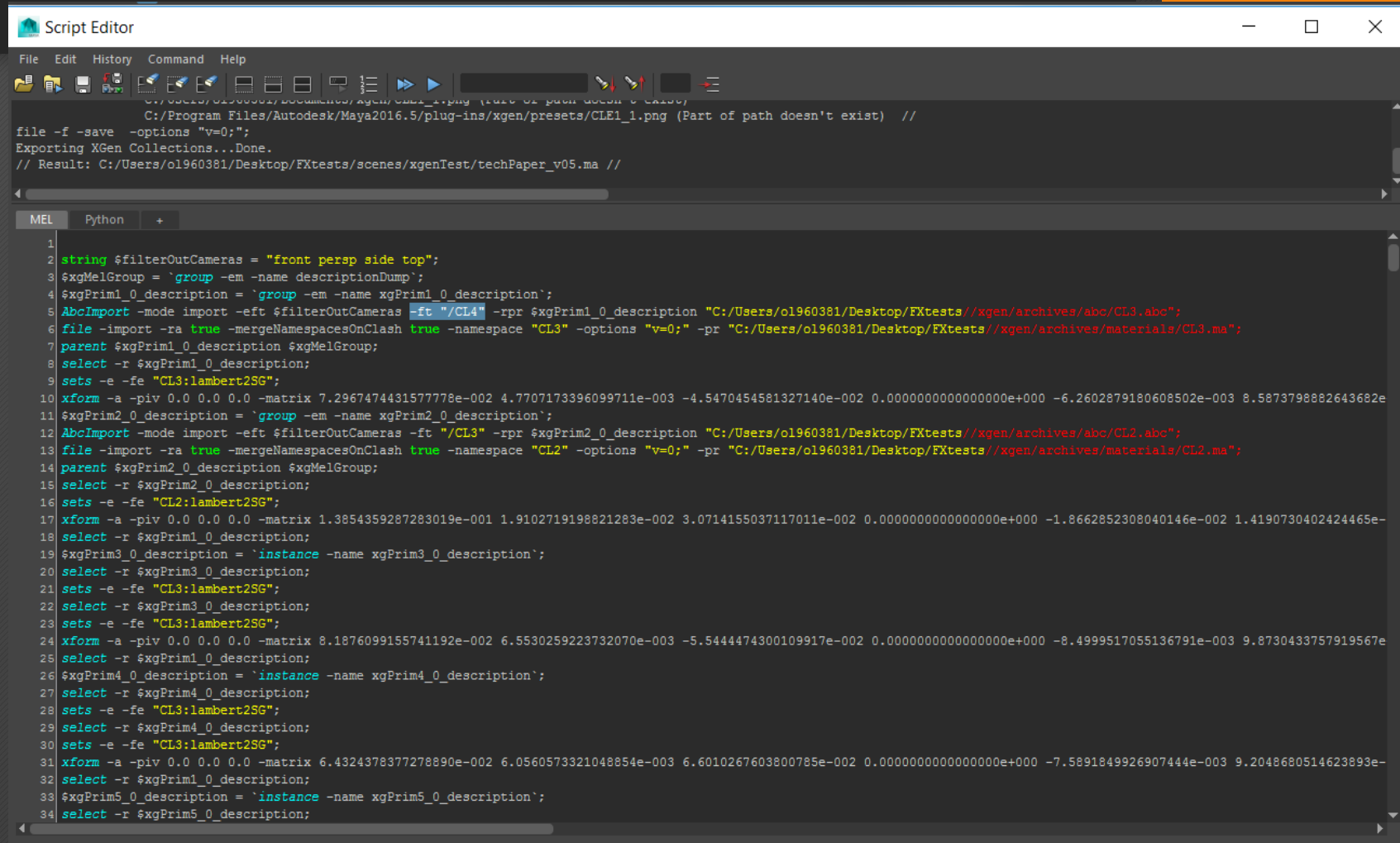
Finalizing your Scene

- In the Preview/Output tab under the output settings, change the operation to “Create MEL file”
- Open the script editor and load the script you just created



Finalizing your Scene

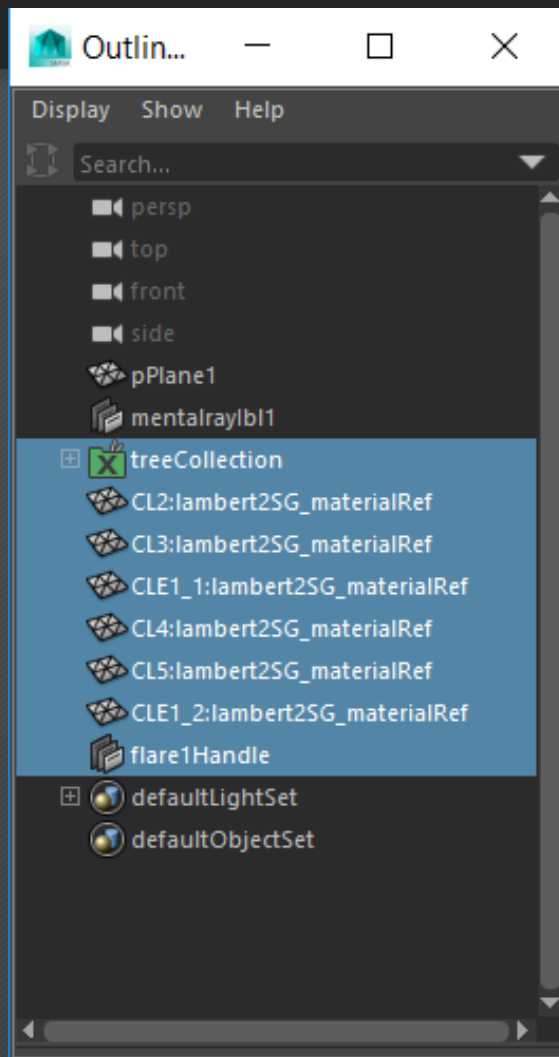
- For every archived file you are bringing in, Maya uses the AbcImport command to bring that model into the scene
- It uses an unnecessary flag that we need to delete called the -ft flag.
- So in each of the AbcImport lines we'll delete the flag "-ft '/MODEL NAME'"; where -ft is the flag and /MODEL NAME is the string argument used

The screenshot shows the Maya Script Editor window. The top menu bar includes File, Edit, History, Command, and Help. Below the menu is a toolbar with various icons. The main text area contains MEL code. The code starts with a comment about a path that doesn't exist, followed by an 'file' command to save options. Then, it shows the export of XGen collections. The main part of the code is a series of MEL commands to create groups, import ABC files, and set properties for several objects (Prim1 through Prim5). The code uses the 'AbcImport' command with various flags like '-ft' and '-mergeNamespacesOnClash'. The code is as follows:

```
Script Editor
File Edit History Command Help
C:/Users/ol960381/Desktop/FXtests/scenes/xgenTest/techPaper_v05.ma
file -f -save -options "v=0;";
Exporting XGen Collections...Done.
// Result: C:/Users/ol960381/Desktop/FXtests/scenes/xgenTest/techPaper_v05.ma //

MEL Python +
1
2 string $filterOutCameras = "front persp side top";
3 $xgMelGroup = `group -em -name descriptionDump`;
4 $xgPrim1_0_description = `group -em -name xgPrim1_0_description`;
5 AbcImport -mode import -eft $filterOutCameras -ft "/CL4" -rpr $xgPrim1_0_description "C:/Users/ol960381/Desktop/FXtests//xgen/archives/abc/CL3.abc";
6 file -import -ra true -mergeNamespacesOnClash true -namespace "CL3" -options "v=0;" -pr "C:/Users/ol960381/Desktop/FXtests//xgen/archives/materials/CL3.ma";
7 parent $xgPrim1_0_description $xgMelGroup;
8 select -r $xgPrim1_0_description;
9 sets -e -fe "CL3:lambert2SG";
10 xform -a -piv 0.0 0.0 0.0 -matrix 7.2967474431577778e-002 4.7707173396099711e-003 -4.5470454581327140e-002 0.0000000000000000e+000 -6.2602879180608502e-003 8.5873798882643682e-003;
11 $xgPrim2_0_description = `group -em -name xgPrim2_0_description`;
12 AbcImport -mode import -eft $filterOutCameras -ft "/CL3" -rpr $xgPrim2_0_description "C:/Users/ol960381/Desktop/FXtests//xgen/archives/abc/CL2.abc";
13 file -import -ra true -mergeNamespacesOnClash true -namespace "CL2" -options "v=0;" -pr "C:/Users/ol960381/Desktop/FXtests//xgen/archives/materials/CL2.ma";
14 parent $xgPrim2_0_description $xgMelGroup;
15 select -r $xgPrim2_0_description;
16 sets -e -fe "CL2:lambert2SG";
17 xform -a -piv 0.0 0.0 0.0 -matrix 1.3854359287283019e-001 1.9102719198821283e-002 3.0714155037117011e-002 0.0000000000000000e+000 -1.8662852308040146e-002 1.4190730402424465e-002;
18 select -r $xgPrim1_0_description;
19 $xgPrim3_0_description = `instance -name xgPrim3_0_description`;
20 select -r $xgPrim3_0_description;
21 sets -e -fe "CL3:lambert2SG";
22 select -r $xgPrim3_0_description;
23 sets -e -fe "CL3:lambert2SG";
24 xform -a -piv 0.0 0.0 0.0 -matrix 8.1876099155741192e-002 6.5530259223732070e-003 -5.544474300109917e-002 0.0000000000000000e+000 -8.4999517055136791e-003 9.8730433757919567e-003;
25 select -r $xgPrim1_0_description;
26 $xgPrim4_0_description = `instance -name xgPrim4_0_description`;
27 select -r $xgPrim4_0_description;
28 sets -e -fe "CL3:lambert2SG";
29 select -r $xgPrim4_0_description;
30 sets -e -fe "CL3:lambert2SG";
31 xform -a -piv 0.0 0.0 0.0 -matrix 6.4324378377278890e-002 6.0560573321048854e-003 6.6010267603800785e-002 0.0000000000000000e+000 -7.5891849926907444e-003 9.2048680514623893e-003;
32 select -r $xgPrim1_0_description;
33 $xgPrim5_0_description = `instance -name xgPrim5_0_description`;
34 select -r $xgPrim5_0_description;
```

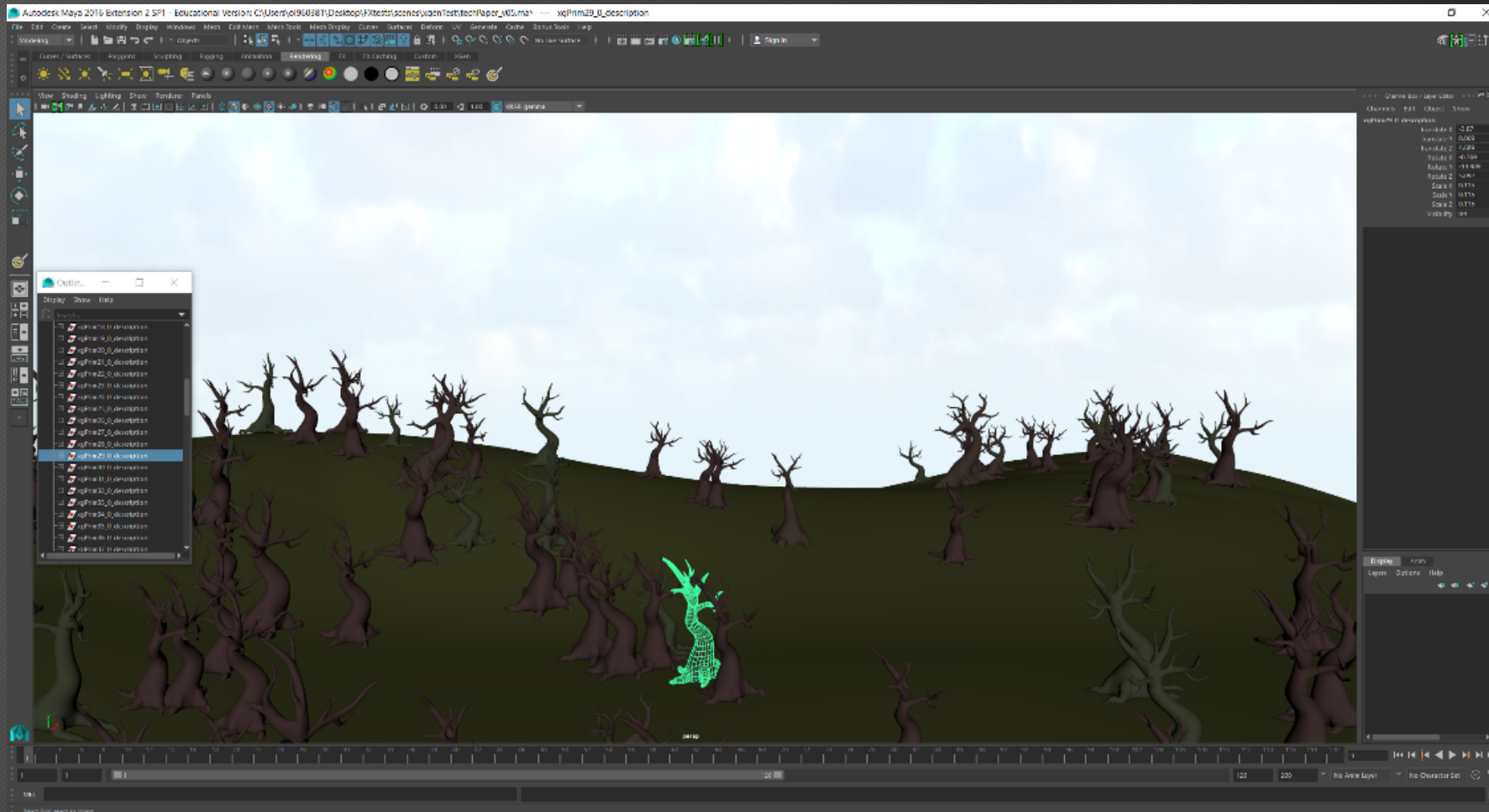

Finalizing your Scene



- Once we have our MEL fixed, we can start from scratch with the surface we want to populate
- We'll delete all of the archives, guides, and references associated with XGen from the outliner.

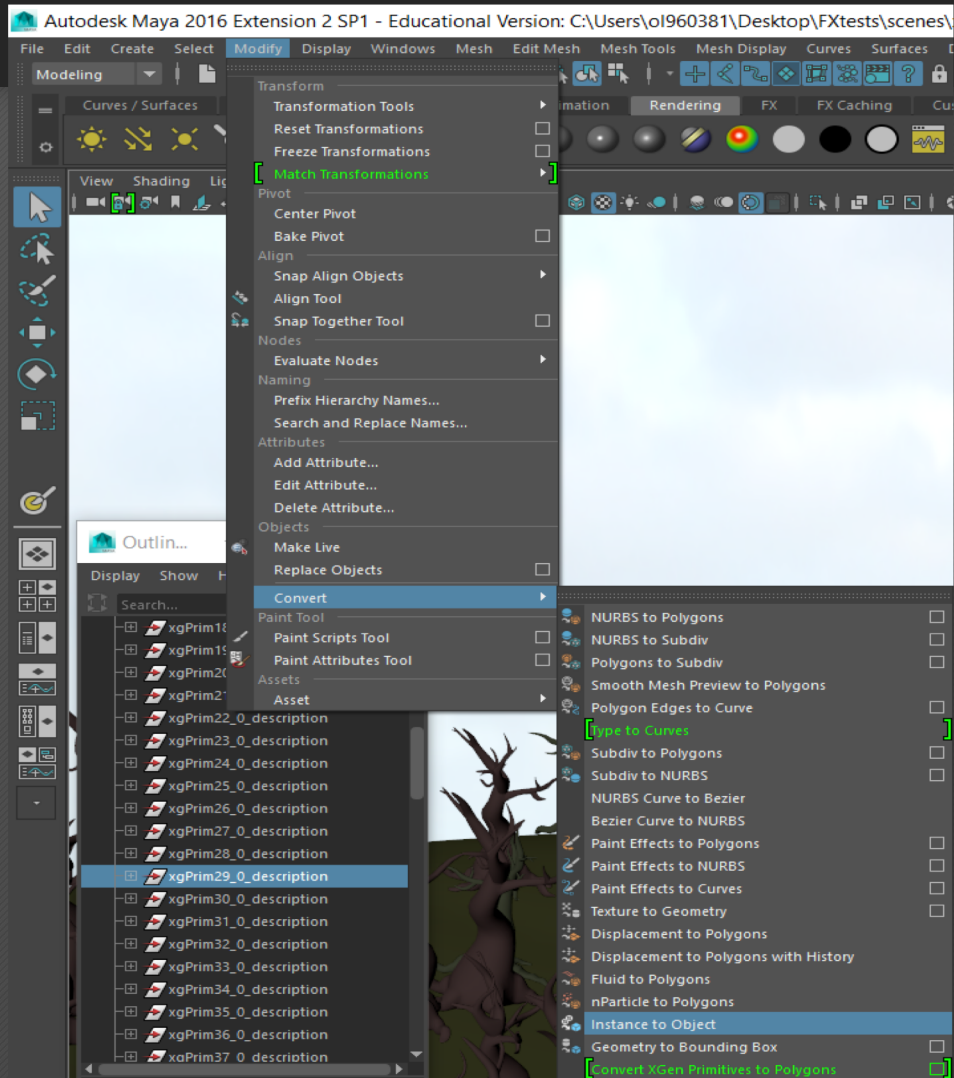
Finalizing your Scene

Yaaaay Rachel



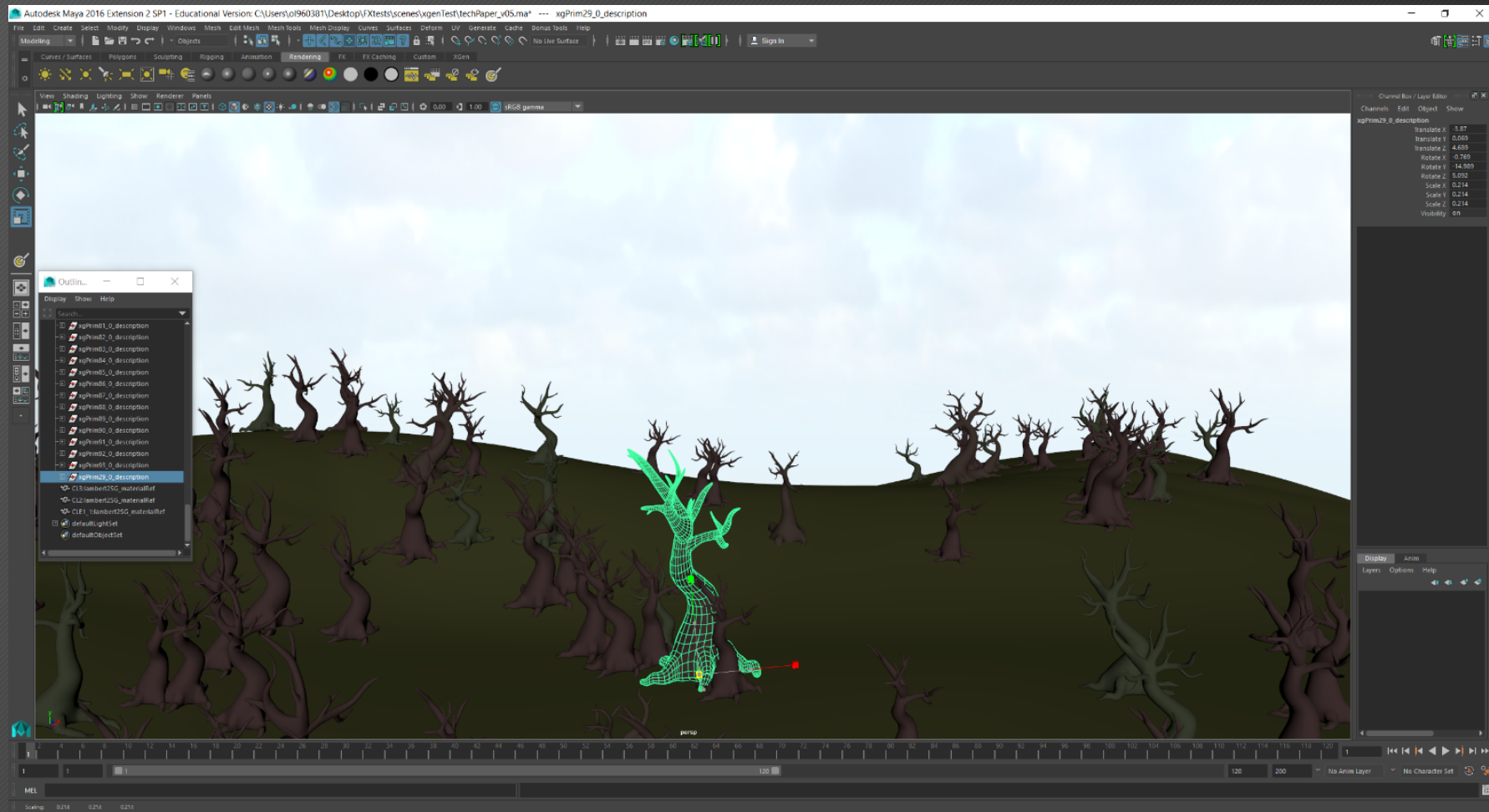
- To finish it off, we'll select our entire MEL script from the script editor and hit enter
- So now we have our scene populated with real geometry that can be transformed individually, since each model maintains its individual transform information from the XGen expressions

Finalizing your Scene



- The last thing we have to worry about is the fact that all the models are instances. To add that last level of detail, we can convert models from instances to objects
- Select either the single model or multiple models in the outliner and then under **Modify > Convert > Instance to Object**

Finalizing your Scene



Now that the object has been converted from an instance, it can be modified and transformed independently from the other tree models

Conclusion

XGen's populating methods make set dressing significantly easier with its procedural attributes. Things like grass, leaves, hair, and other things that require large numbers of assets can be made completely in minutes rather than having to hand place all of them.

