Maya XGen Notes

*Notes about getting xgen set up properly.*

**Setup**

* A collection is just a collection of descriptions.
* A description is a set of modifiers for the hair generation.
* Faces and objects can be added to a description through the Descriptions tab > Bind Patches > Add selected faces.

**Primitives**

* Typically the grooming hair option is best to give us the most control over the shape.
* Under “Primitive Attributes” you can turn off the “Face Camera” option to face the cards a specific way.

**AnimWires (modifier)**

* Anim wires are used to animate the hair and let it dynamically move.
* SETTING UP:

1. To set up an AnimWires modifier, first you select “Create maps” and generate points where the hair will be animated around.
2. After submitting it, select “Create Hair System” and check each box.
3. Select “Attach Hair System” with the hair system’s output curves group selected.

* The dynamics of the AnimWires hair system can be edited by selecting the hairSystem object in the hierarchy and changing attributes such as dampness, straightness, stiffness, etc.

**Wind (modifier)**

* Wind adds a small force to the hair splines to make them move.
* Uses noise and a vector direction.
* This does NOT require AnimWires.

**Interactive Hair Grooming**

* This is the newer version of descriptions in maya. It lets you have more control over the actual grooming process and updates the hair in realtime.

**Errors**

1. “Failed to evaluate subd. [Description], [Object]” This occurs when there are too many edges connected to a single vertex and it can affect AnimWires and other generation attributes. To fix, either clean up the topology or remove the faces from the description by going to the Descriptions tab > Bind Patches > Remove selected faces.
2. “Live mode is enabled but the reference frame wasn’t read yet: 1” Simply set the playback to frame 1 before previewing the hair.
3. “Trouble closing bend ptx file” Has trouble saving out to the ptx file. A simple maya restart usually fixes it.
4. No xgen information is loaded after opening the scene. This is because the path file can’t be found (Collection > Attribute Editor > Xg File Name). There seems to be an issue where Maya might just stop exporting an xgen file after it’s saved, and so far I can’t find a fix for it other than just referencing an older xgen file.

**Workflow**

* Each Maya scene typically saves its own “.xgen” file which contains the info about the xgen in the scene. If this isn’t happening automatically then an xgen file needs to be referenced in the collection’s attribute editor.
* To import xgen info into a new scene, be sure to select “Import collection” in the xgen window. Doing so will ask to create a new collection or replace the existing one. Any matching polygons from the collection will automatically have their descriptions applied.
* Import any grooming info by selecting “Import grooming” in the description’s options and choosing its grooming folder. Although, some settings such as bend and grooming info seems to need to be done manually.

**Editing Geometry**

* As far as I know, editing the geometry’s faces requires the generation to be redone.
* Moving/Scaling/Rotating an object in xgen will update the generated positions of the leaves/hairs, BUT the grooming guides seem to remain in place. Redoing the guides might be necessary for fixing.
* Removing or adding faces from the description also requires redoing the grooming.

**Duplicating Geometry**

* The only “true” method of duplicating geometry right now is doing each step manually: Duplicating the mesh, creating a new description, grooming the guides, and so forth. You could potentially save some work by keeping each tree together on the same description, but any problems with the mesh would mean redoing all the trees again.
* It’s also possible to take each frame of the generation, convert it to geometry, then import each frame and use them as a multi-target blendshape. Then, the tree would appear animated as the blendshape moves from 0-1, then resets back to 0 on the loop. (To convert to geometry, select “Generate” at the top menu and “Convert Xgen primitives to polygons. Make sure the animation is on the correct frame too.)
* <https://forums.autodesk.com/t5/maya-dynamics/convert-simulated-xgen-hair-to-animated-geometry/td-p/7408167>

**Xgen with Standin**

* An xgen description can be converted into a Standin file to make the scene run very fast without requiring a preview until render time (in Arnold).
* Converting xgen to standin:

1. Once we’re satisfied with the look and animation of our description, click File > Export Patches for Batch Render. (The standin uses information from this when being exported.)
2. Turn on the description’s preview and make sure “update automatically” is checked.
3. With the description selected, choose Arnold > Standin > Export Standin.
   1. If using animation, turn on “Sequence” and choose the frames that should be exported with it. Each frame of the standin will be exported as a separate .ass file.
   2. NOTE: During export, Arnold’s current render settings will be used. Make sure the settings are finalized before export. For example, we should choose if we want to export with motion blur or not.
4. Once exported, choose Arnold > Standin > Create Standin
5. To use the standin’s animation, open the attribute editor and select “Use File Sequence.” The “Frame” box should turn purple and automatically get each frame of the sequence. You can also try typing “=frame.”

* This is useful for things like trees that have a large amount of leaves and fields of grass.
* (Ex: For a large field of grass we can split it up into smaller patches and create a description for patch. We can then export each patch as a standin and hide standins that won’t be visible in the render view. This also means we’ll only need to use a few dozen standins in the scene view.)
* (Ex: For a tree we can simply export the whole object (leaves, trunk) as a standin and drop the standin in places the tree should be visible.)
* NOTE: Sometimes the animation preview of xgen doesn’t work. I still haven’t figured out the specific cause because it’s janky. If exporting was done correctly, try messing with updating the original description’s preview. (Also, I’m still not sure if exporting patches for batch render, turning on update automatically, etc. is required for a standin export.)

**Expressions**

* Expressions can be used in xgen to generate primitives and shader attributes.
* Expressions use global and local variables. A local variable can be defined by using the “$” sign and ending each line with “;”
* [Random Culling Example]
  + I want to randomly cull each primitive. XGen will run the expression *for each* primitive I want to get rid of. If the number is anything *except* 0, it will be culled.
  + Code:
    - $rand = rand(); #(Or rand(0,1))
    - Round($rand)
  + Maya doesn’t seem to like it when I use “int” or “float”, so I had to either do an “if, else” check or just round it.
* [Random Noise Example]
  + I want to randomly cull each primitive based on noise values I control.
  + Code:
    - $hi =50; #0.00,100.00
    - $lo =rand(-100,0); #-100.00,0.00
    - round(noise($lo, $hi))
* These expressions can also be used to affect the color of the shader. We can first determine the color by entering their RGB values, then we can randomize it by putting it directly into a noise expression.
  + Code:
    - $hi = 52.4648; #0.00,100.00
    - $lo = rand(-1,0); #-100.00,0.00
    - $a = noise($lo, $hi);
    - $color1 = [0.978,0.144,0.144]; #The value used in both the "G" and "B" spots of an RGB color.
    - $color2 = [0.227451,0.701961,1]; #Same thing, but the value we want to lerp to.
    - if ($a < 0) #Make sure $a is between 0-1.
    - {
    - $a \*= -1;
    - }
    - if ($a > .5) #Set the color to either value1 or value2.
    - {
    - $b = $color1;
    - }
    - else
    - {
    - $b = $color2;
    - }
    - $b
  + It’s also possible to use a lerp value to smoothly move between values.
    - $lerpValue = $colorValue1 \* (1 - $a) + $colorValue2 \* $a;
    - *(Function lerp (start, end, t)…*
    - *(start \* (1 – t) + end \* t)*