Creating SpeedTrees Start-to-Finish

Step 1: Copy the Speedtree5 folder from the network to somewhere on your local drive and create shortcuts for easy access to the Modeler and Compiler (you need both).

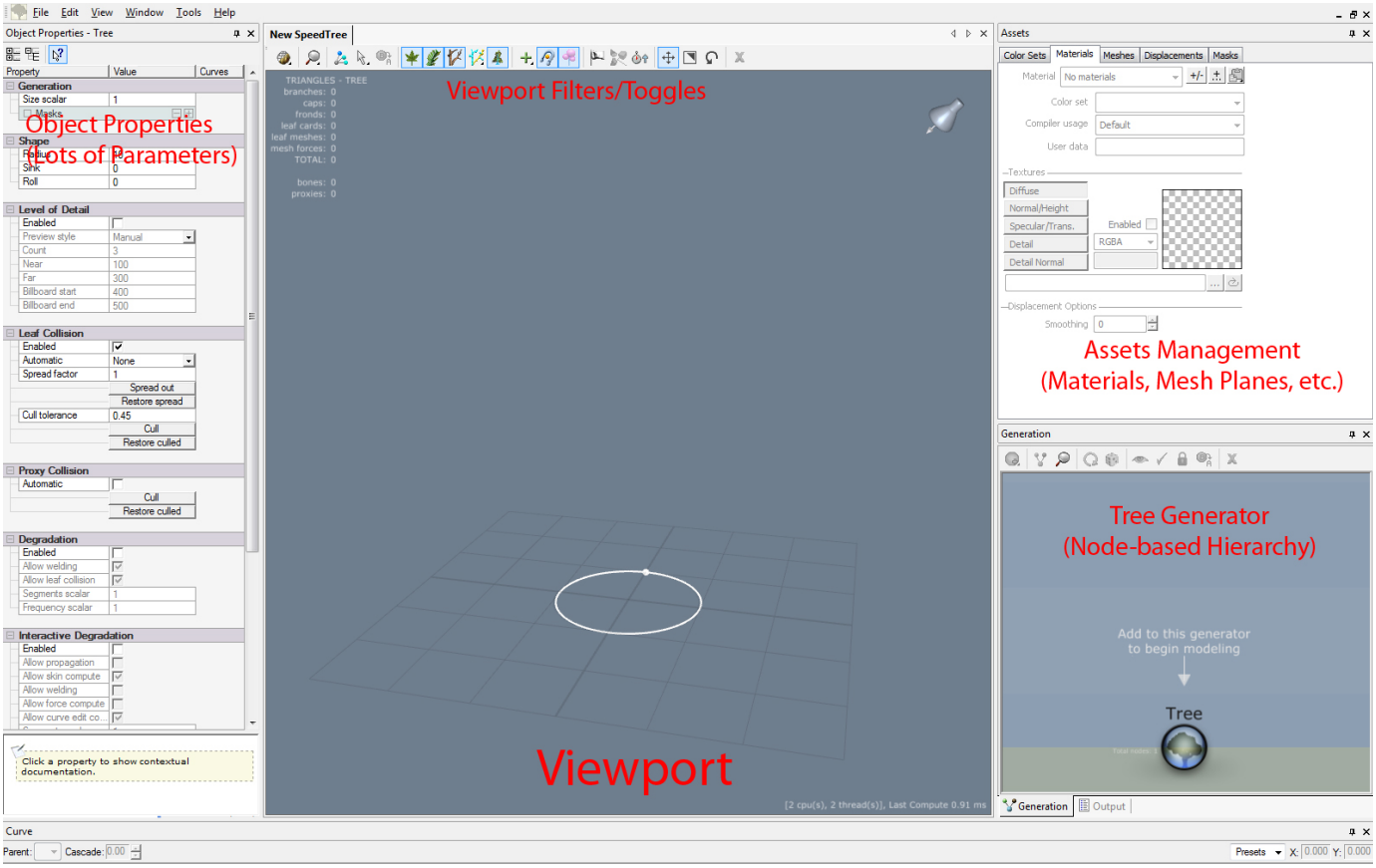
Network Location: \\SERVER\Place

DO NOT RUN SPEEDTREE FROM OR SAVE TO THIS LOCATION. MAKE A COPY ON YOUR OWN DRIVE.

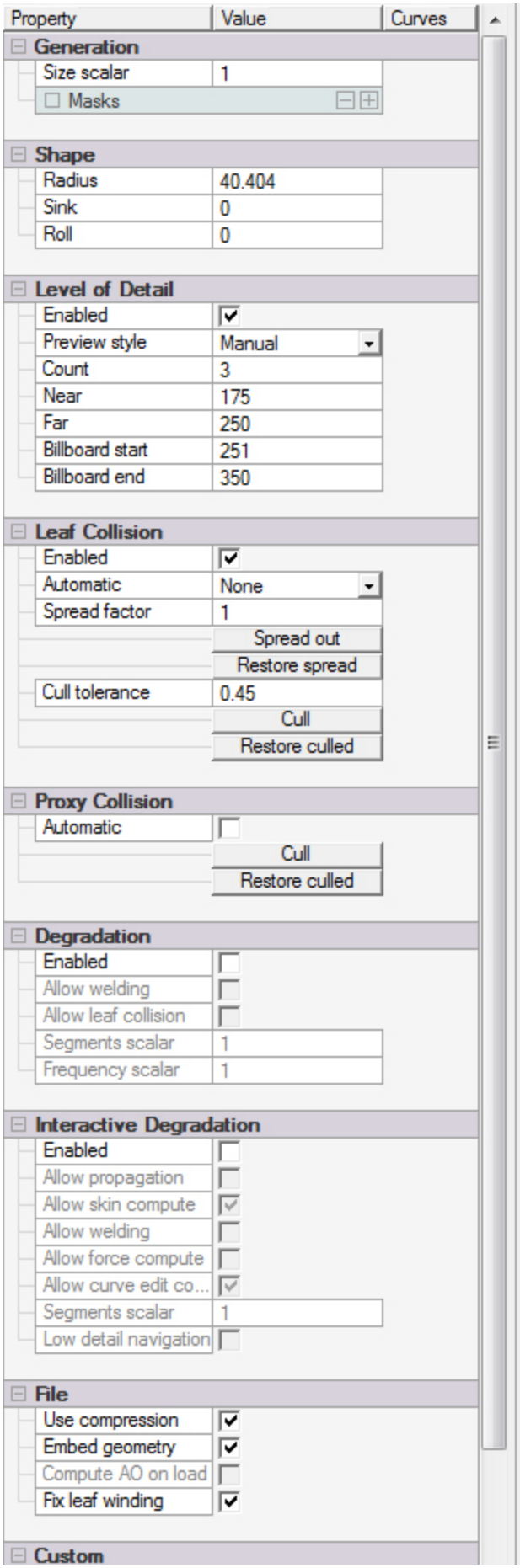
You will have two folders inside named Program and Trees; the latter of the two includes the tree library which will serve as a starting point for most of your trees (Only use trees with a _RT suffix as they are optimized for game engine use).

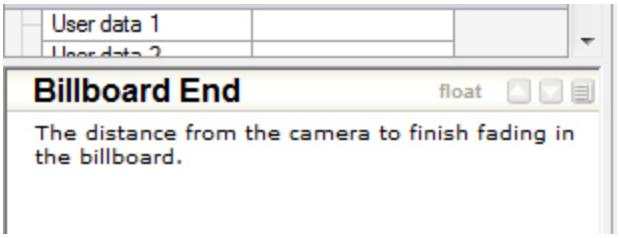
IMPORTANT: Do not save over tree library trees, once opening a tree library model for customizing save a copy elsewhere

Step 2: Launch the SpeedTree Modeler and you will be presented with what may or may not seem like a complicated interface so in the event of the former here is an interface breakdown:



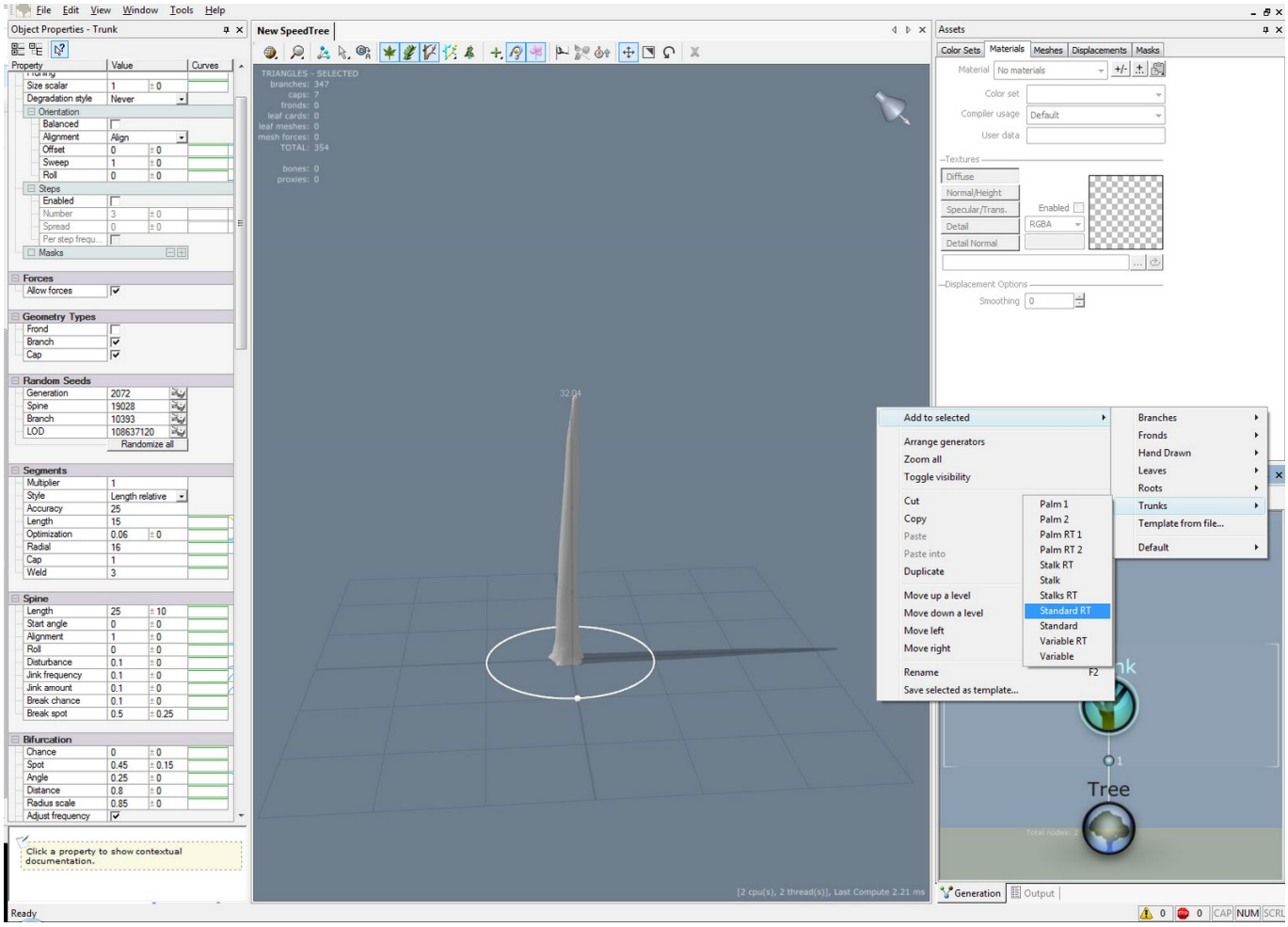
Now the first thing we're going to do is start off making a tree from scratch before you get into starting off with pre-existing trees from the tree library. There are a set of parameters to take a look at before you generate so select your Tree node from the generator section and you should see these parameters on your left. Tweak them as you see fit but here is a breakdown:


Property
Value
Curves
Generation
Size scalar
1
Masks
Shape
Radius
40.404
Sink
0
Roll
0
Level of Detail
Enabled
Preview style
Manual
Count
3
Near
175
Far
250
Billboard start
251
Billboard end
350
Leaf Collision
Enabled
Automatic
None
Spread factor
1
Spread out
Restore spread
Cull tolerance
0.45
Cull
Restore culled
Proxy Collision
Automatic
Cull
Restore culled
Degradation
Enabled
Allow welding
Allow leaf collision
Segments scalar
1
Frequency scalar
1
Interactive Degradation
Enabled
Allow propagation
Allow skin compute
Allow welding
Allow force compute
Allow curve edit co...
Segments scalar
1
Low detail navigation
File
Use compression
Embed geometry
Compute AO on load
Fix leaf winding
Custom

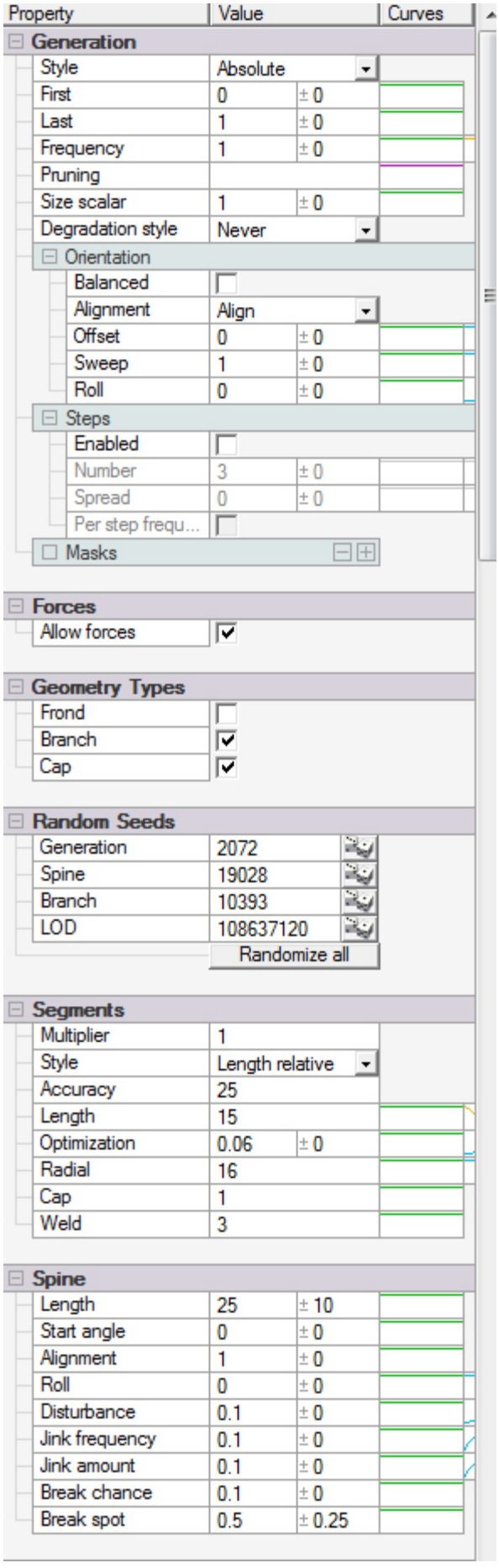


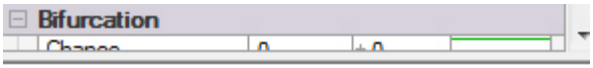
- Generation
  - Size scalar: The total uniform scale of a tree.
- Shape
  - Radius: The radius of the tree area (The white circle in the viewport, useful if you're making clusters of trees/plants).
  - Sink: How far down the base of a tree/plant sits relation to the origin.
  - Roll: The rotation of the tree relative to the origin.
- Level of Detail (Should be enabled by default for game trees)
  - Preview style (Manual/Near and far): Toggles between manual LOD previewing or using distance to preview LODing.
  - Count: How many levels of LODs you want. Should probably stay at 3.
  - Recommended Near/Far/BB start/BB end Settings: 175/250/251/350
- Leaf Collision (Keeps leaf sprites from clipping into each other, handles how far spread out they are or culls them out if they clip with other leaves).
  - Spread factor: How much to spread out leaf sprites.
  - Cull tolerance: Tolerance level before a clipping leaf sprite is removed.
- Degradation (Wholesale optimization flags for the entire tree)
  - Allow welding: Disables or allows welding on the entire tree.
  - Allow leaf collision: Disables or allows leaf collision.

Step 3: Right click on Tree on the lower right corner in the Tree Generator area of the interface and go to Add to Selected->Trunks->Standard RT (again always use RT for game trees). You should create something like this:



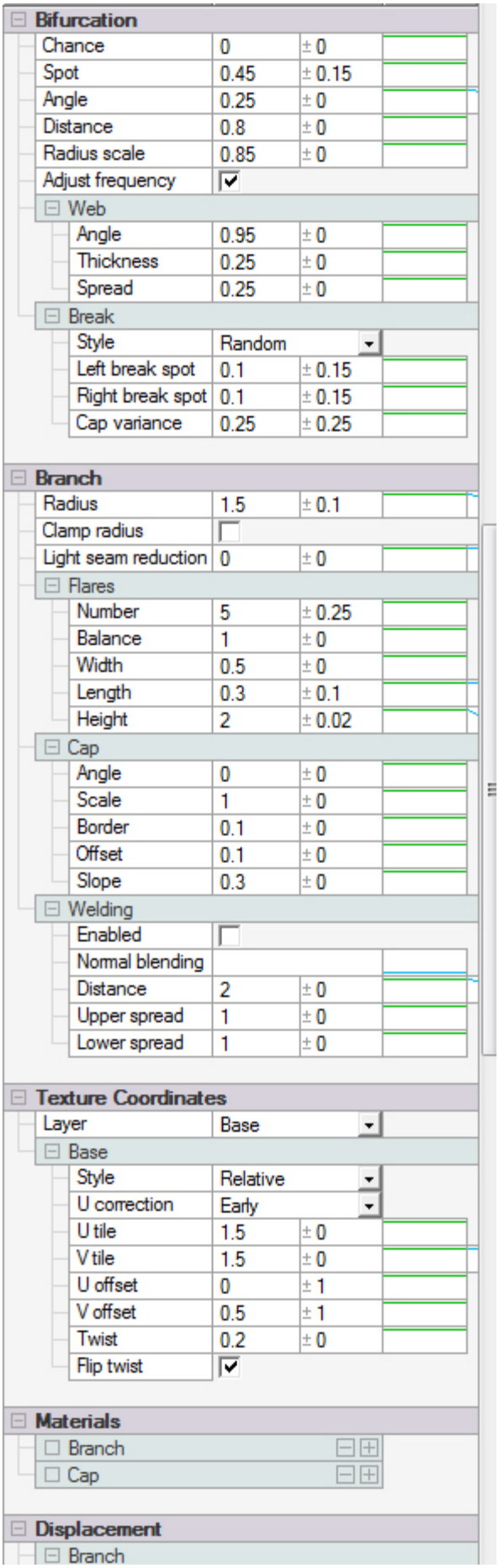
Take a look at the Trunk parameters on the left and play around with them to see how they function. Here's a breakdown to provide some clarity, also keep in mind that most parameters have a curve that if you click you can tweak a Bezier curve to get very distinct values that may have a falloff or be based on a certain area of the tree. They are extremely useful, accommodate yourself with them.

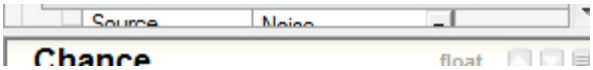

Property
Value
Curves
Generation
Style
Absolute
First
0
± 0
Last
1
± 0
Frequency
1
± 0
Pruning
Size scalar
1
± 0
Degradation style
Never
Orientation
Balanced
Alignment
Align
Offset
0
± 0
Sweep
1
± 0
Roll
0
± 0
Steps
Enabled
Number
3
± 0
Spread
0
± 0
Per step frequ...
Masks
Forces
Allow forces
Geometry Types
Frond
Branch
Cap
Random Seeds
Generation
2072
Spine
19028
Branch
10393
LOD
108637120
Randomize all
Segments
Multiplier
1
Style
Length relative
Accuracy
25
Length
15
Optimization
0.06
± 0
Radial
16
Cap
1
Weld
3
Spine
Length
25
± 10
Start angle
0
± 0
Alignment
1
± 0
Roll
0
± 0
Disturbance
0.1
± 0
Jink frequency
0.1
± 0
Jink amount
0.1
± 0
Break chance
0.1
± 0
Break spot
0.5
± 0.25



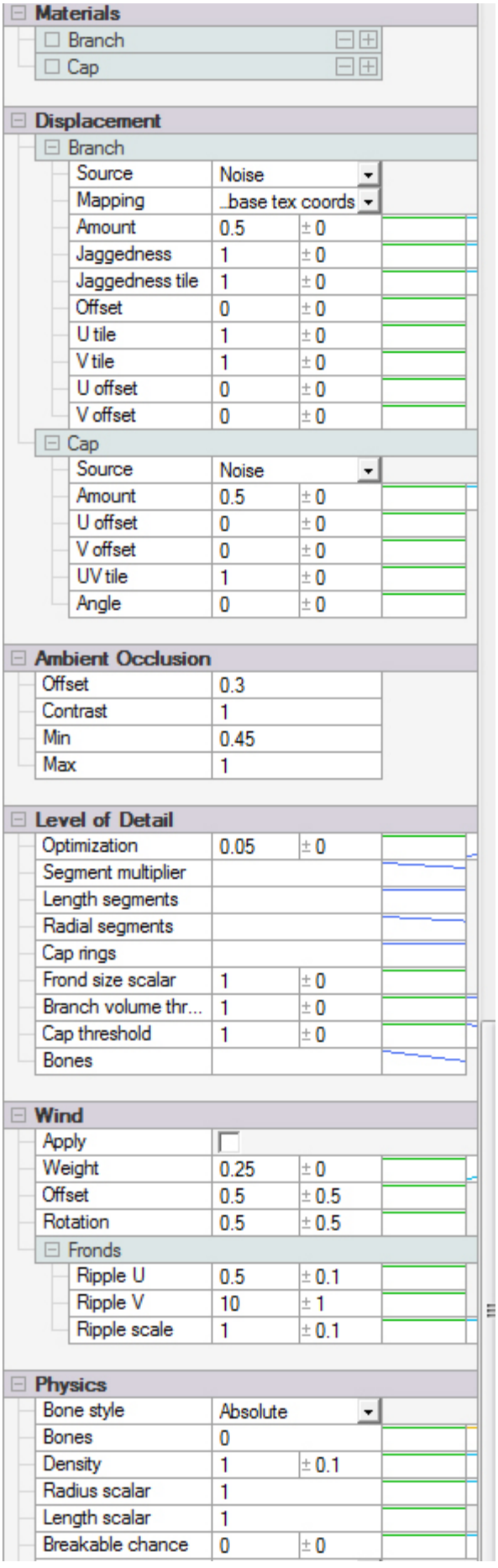
Note: You'll notice to the right of each value is a + or - box to enter inputs as well. This isn't 100% but I believe that is a plus or minus range for when you perform random seed operations so it stays within that set value in a positive or negative direction of your main input value. (Ex. If you look at length its value is 25 with a +- of 10 so when randomizing it'll never go lower than 15 but never higher than 35). Test this out and make sure that's an accurate assessment. It does effect the tree as soon as you enter values however.

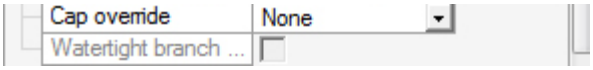
- Generation
  - First: The lowest point that a node can appear on a parent (this makes more sense on branches/fronds, for trunks it just seems to slide the trunk closer to the edge of the radius on one axis)
  - Last: The highest point that a node can appear on a parent (this makes more sense on branches/fronds, for trunks it just seems to slide the trunk closer to the edge of the radius on one axis)
  - Frequency: Multiplier to control how many instances of a node are created (not that useful for trunks, much more so for fronds/branches)
  - Pruning: Just a curve that helps you optimize your tree, play with it.
  - Size scalar: Uniform scale for the trunk.
  - Degradation style (Always/Never/Default): Whether or not to obey degradation settings you set in the master Tree node.
- Orientation
  - Offset/Sweep/Roll: All seem to rotate the trunk as if it sat upon a turntable.
- Steps
  - Number: Multiplier to create varied instances of the same Node (not that useful for Trunks)
  - Spread: Spreads out the steps.
- Segments
  - Multiplier: Uniform scale of length, radial and cap segments.
  - Accuracy: Not really sure how to explain (From the help box: The maximum fidelity of spines).
  - Length: Number of Vertical Edge Loops (cut down to optimize, especially for branches that are barely seen, but also avoid trunks specifically having hard angle breaks that make it look low poly).
  - Optimization: Cuts down polygons, more useful on branches as Trunks are generally already pretty low poly. Very useful overall however.
  - Radial: Horizontal edge loops (cut down for optimization)
  - Cap: Whether or not to have a cap on the trunk.
  - Weld: How many segments compose the fused part of welded branches (disabling welding REALLY helps cut down on polys and the difference in appearance is not significant).
- Spine
  - Length: The length of a node (height of a trunk or length of branches)
  - Start Angle: Tilts the trunk or branch.
  - Alignment: 0 to 1 interpolation between a random up direction and the world up direction.
  - Roll: Rotates in a turntable manner.
  - Disturbance: Randomizes the angles of each curve point to make interesting trunk/branch shapes.
  - Jink frequency: The rate of "jink" to apply to a node.
  - Jink amount: The amount of "jink" - Not sure exactly what this means but it seems to distort and twist the trunk node. Play with it to create your own understanding.
  - Break chance: 0 to 1 chance of a branch/trunk breaking as if it was snapped off.
  - Break spot: How far down the branch/trunk the break occurs.


Bifurcation
Chance 0 ± 0
Spot 0.45 ± 0.15
Angle 0.25 ± 0
Distance 0.8 ± 0
Radius scale 0.85 ± 0
Adjust frequency
Web
Angle 0.95 ± 0
Thickness 0.25 ± 0
Spread 0.25 ± 0
Break
Style Random
Left break spot 0.1 ± 0.15
Right break spot 0.1 ± 0.15
Cap variance 0.25 ± 0.25
Branch
Radius 1.5 ± 0.1
Clamp radius
Light seam reduction 0 ± 0
Flares
Number 5 ± 0.25
Balance 1 ± 0
Width 0.5 ± 0
Length 0.3 ± 0.1
Height 2 ± 0.02
Cap
Angle 0 ± 0
Scale 1 ± 0
Border 0.1 ± 0
Offset 0.1 ± 0
Slope 0.3 ± 0
Welding
Enabled
Normal blending
Distance 2 ± 0
Upper spread 1 ± 0
Lower spread 1 ± 0
Texture Coordinates
Layer Base
Base
Style Relative
U correction Early
U tile 1.5 ± 0
V tile 1.5 ± 0
U offset 0 ± 1
V offset 0.5 ± 1
Twist 0.2 ± 0
Flip twist
Materials
Branch
Cap
Displacement
Branch



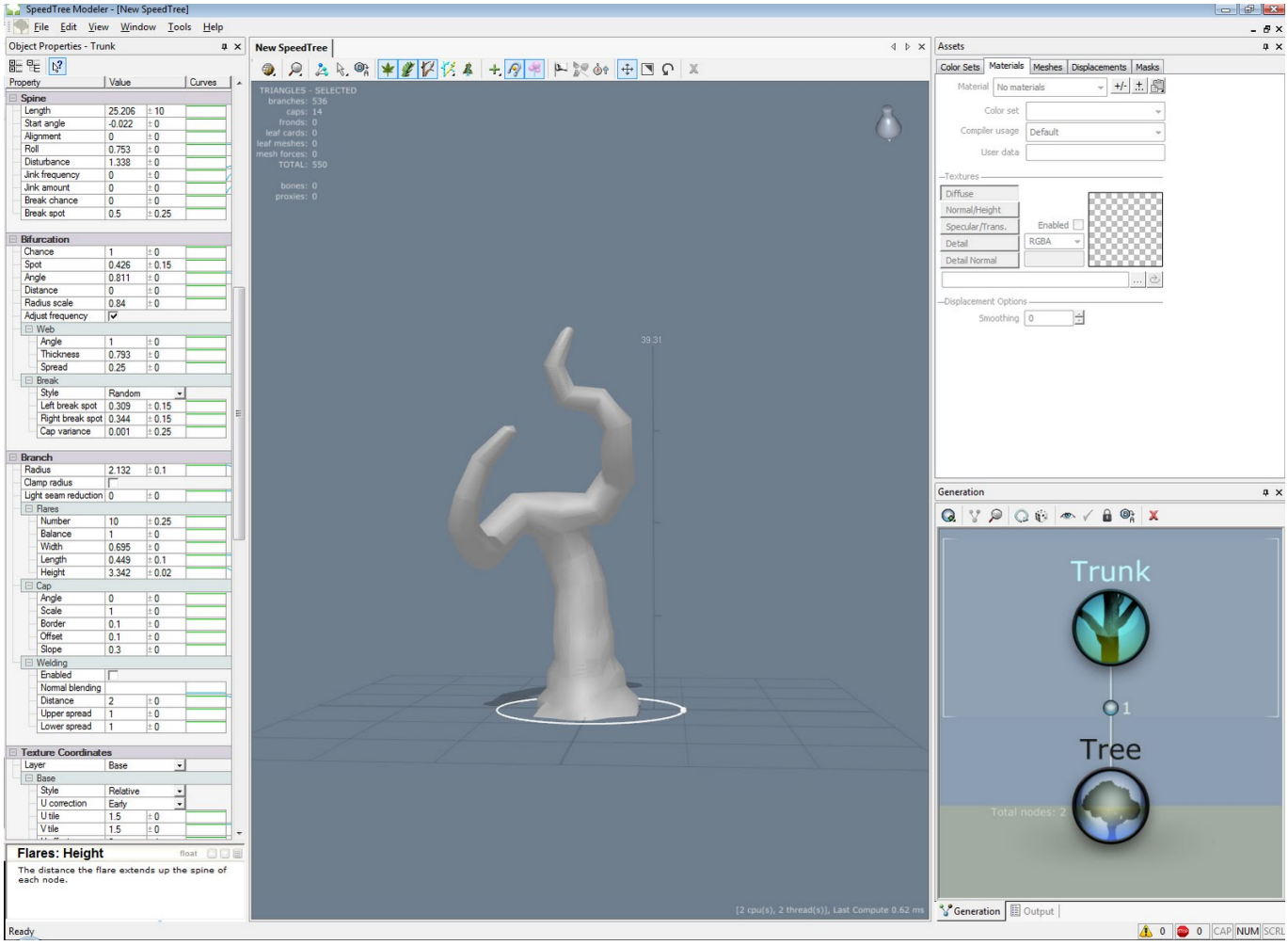
- Bifurcation: The forking of a trunk or branches into smaller ones. Play around with all of the parameters at your disposal if you want branches/trunks to split.
- Branch: Similar parameters to the Spine section.
  - Radius: The thickness of a trunk or branch.
  - Clamp radius: Allows or disallows the radius to exceed the scene radius.
  - Flares: Controls the base of the trunk and how it flares to help with ground/root integration.
    - Number: Behaves really strangely, seems to act like more of a randomizer than setting how many flares the base actually has.
  - Welding: Should probably be disabled in most instances, it saves significantly on polys.
- Texture Coordinates: Common settings for tweaking the UVs, unfortunately to my knowledge there's no way to manually adjust UVs without exporting to MAX but usually the UVs are pretty decent.
  - NOTE: Make sure to adjust your UV tiling in SpeedTree. If you try to adjust bark UVs in Unreal using a texture coordinate it will break the UVs and you'll see a nasty seam. If your bark looks bad try changing the style to V Relative and experiment with tilings (they don't always have to be the same).
- Materials: Apply created materials here from the drop down.
  - Branch: This is the material applied to the branch/trunk itself.
  - Cap: This is applied to the end caps. Generally you'll just want to apply the same material to both Branch and Cap. Some trees from the library have cap materials and those should in most cases be disregarded.


Materials
Branch
Cap
Displacement
Branch
Source Noise
Mapping _base tex coords
Amount 0.5 ± 0
Jaggedness 1 ± 0
Jaggedness tile 1 ± 0
Offset 0 ± 0
U tile 1 ± 0
V tile 1 ± 0
U offset 0 ± 0
V offset 0 ± 0
Cap
Source Noise
Amount 0.5 ± 0
U offset 0 ± 0
V offset 0 ± 0
UV tile 1 ± 0
Angle 0 ± 0
Ambient Occlusion
Offset 0.3
Contrast 1
Min 0.45
Max 1
Level of Detail
Optimization 0.05 ± 0
Segment multiplier
Length segments
Radial segments
Cap rings
Frond size scalar 1 ± 0
Branch volume thr... 1 ± 0
Cap threshold 1 ± 0
Bones
Wind
Apply
Weight 0.25 ± 0
Offset 0.5 ± 0.5
Rotation 0.5 ± 0.5
Fronds
Ripple U 0.5 ± 0.1
Ripple V 10 ± 1
Ripple scale 1 ± 0.1
Physics
Bone style Absolute
Bones 0
Density 1 ± 0.1
Radius scalar 1
Length scalar 1
Breakable chance 0 ± 0



- Displacement: Use a displacement map to drive the geometry of the selected node.
    - Play around with the parameters within this category to get the desired effect. Have no experimented much with this.
- Ambient Occlusion: Parameters for tweaking AO, haven't really used AO too much for SpeedTrees so not sure how useful it actually is.
- Level of Detail: A variety of parameters and curves to tweak how much the tree is cut down poly-wise when being generated.
- Wind: Tweak wind parameters for determining how the tree appears to move with simulated wind.
- Physics: If you want breakable/destructable trees the settings can be set here but for what we're doing we'll probably never or very rarely use tree physics (It's probably very expensive).

Step 4: Take some time playing around with these parameters, just reading a description of what they do is not going to teach you how to use them. Try to create something interesting while still having a relatively low poly count without the tree looking polygonal. This is what I came up with:

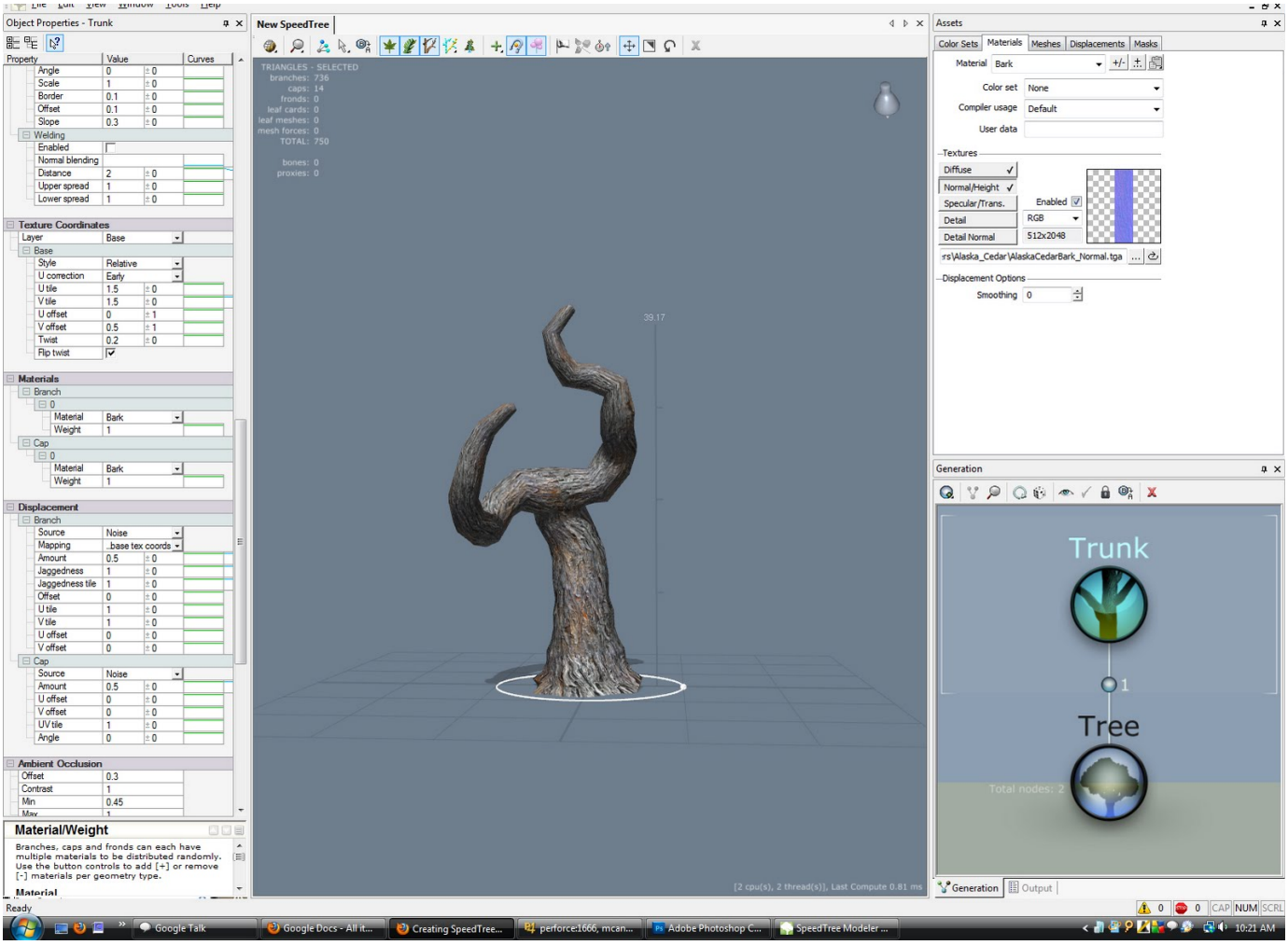


This was created mostly by tweaking Bifurcation settings and Disturbance and Jink parameters under the Spine section. As you can see it looks pretty good at a low poly count of 550 (you could probably get rid of some of the polygonal look on the right fork by increasing the length under Segments.

Step 5: Now that you have a trunk we need to get some textures on there to get it looking good. On the top right of the interface click the Materials tab and hit the +/- button. A screen will come up with a blank list. Click 'Add' and name this Bark.

You can add any texture files to the various channels available to you but for our stuff we'll mostly only use Diffuse and Normal. Navigate to the tree library and select a bark. I went into the Conifers/Alaskan_Cedar folder and used that bark diffuse and normal and loaded it in using the '...' button under the channel buttons.

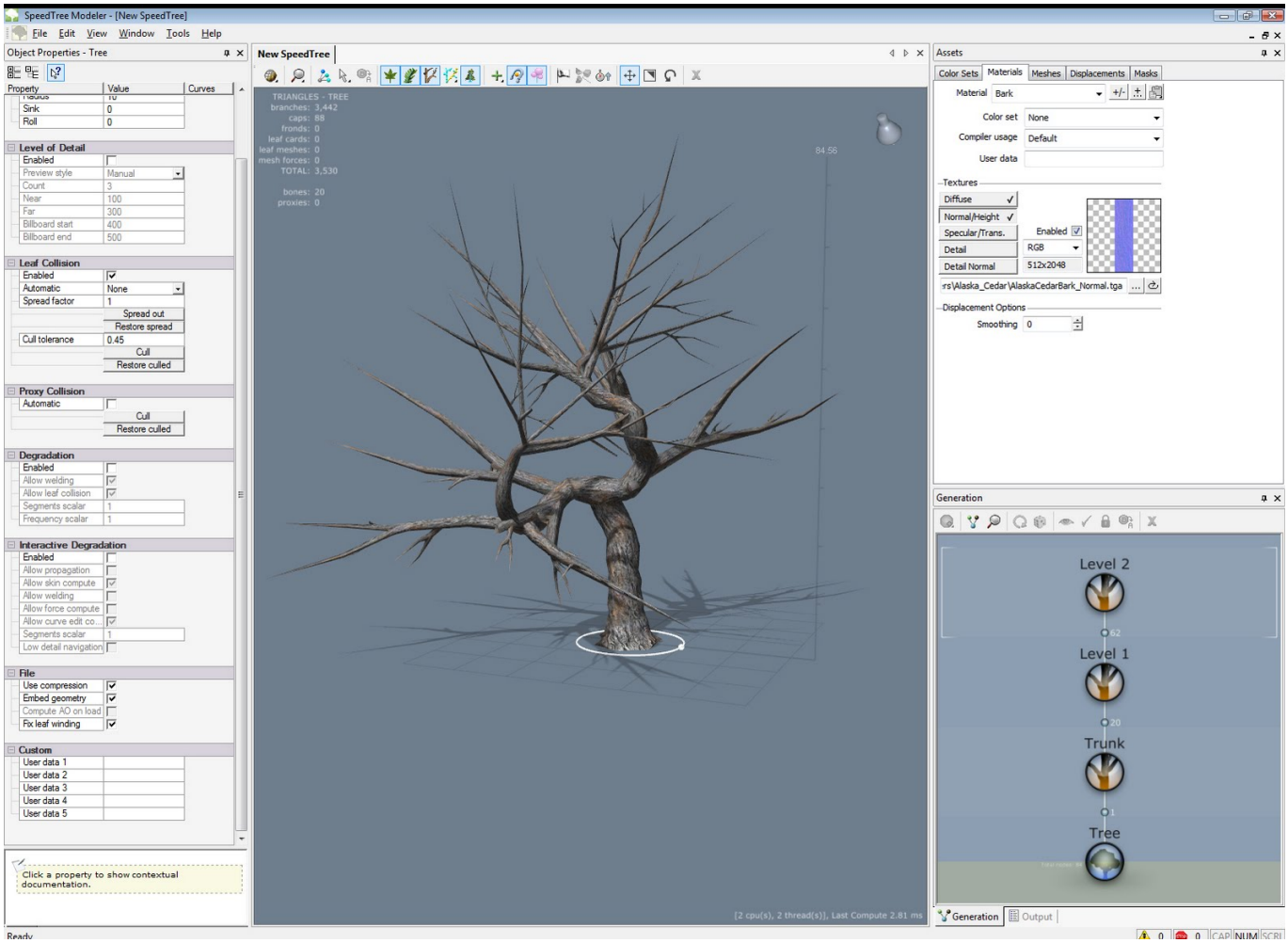
After the Diffuse and Normal maps are loaded into the material they need to be assigned to the Trunk. On the left navigate to the Materials panel and drop down Branch and Cap. From the Material dropdown list drop it down and choose the Bark material for both Branch and Cap. If you used Alaskan Cedar as well you should see something like this:



Step 6: Let's get some branches on this bad boy. Right click on Trunk in the Generator panel and navigate to Add to Selected->Branches->Standard RT and if your tree looks anything like mine the initial result should be something pretty chaotic looking.

The parameters for Branches are identical to that of the Trunk. Before you get too far however navigate to the Branch section and uncheck Enabled under Welding (this cuts down on polys quite a bit, there's no need for the branches to be sewn in with the tree).

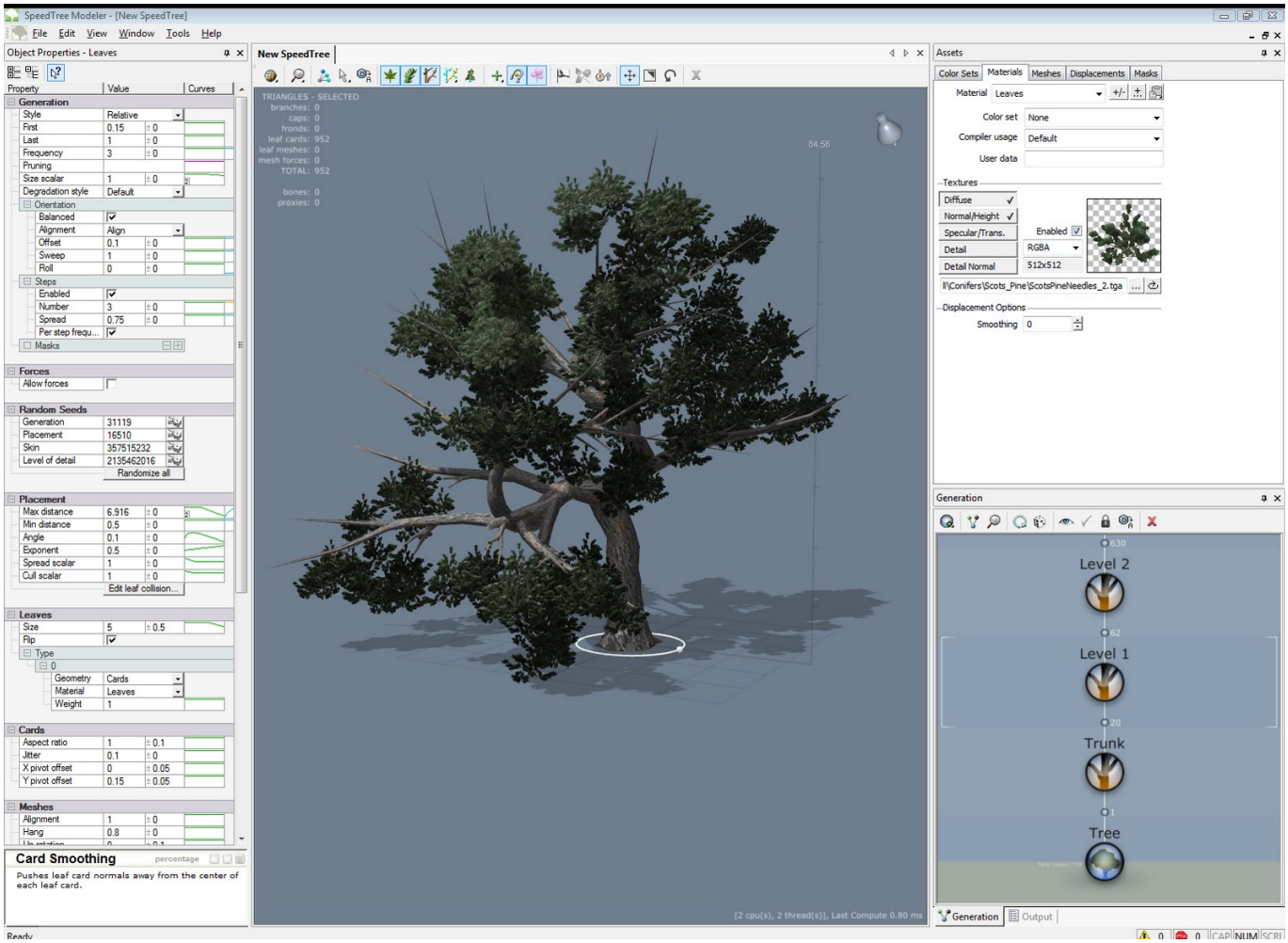
Tweak the parameters of Level 1 and Level 2 of the branches to get something that looks interesting.

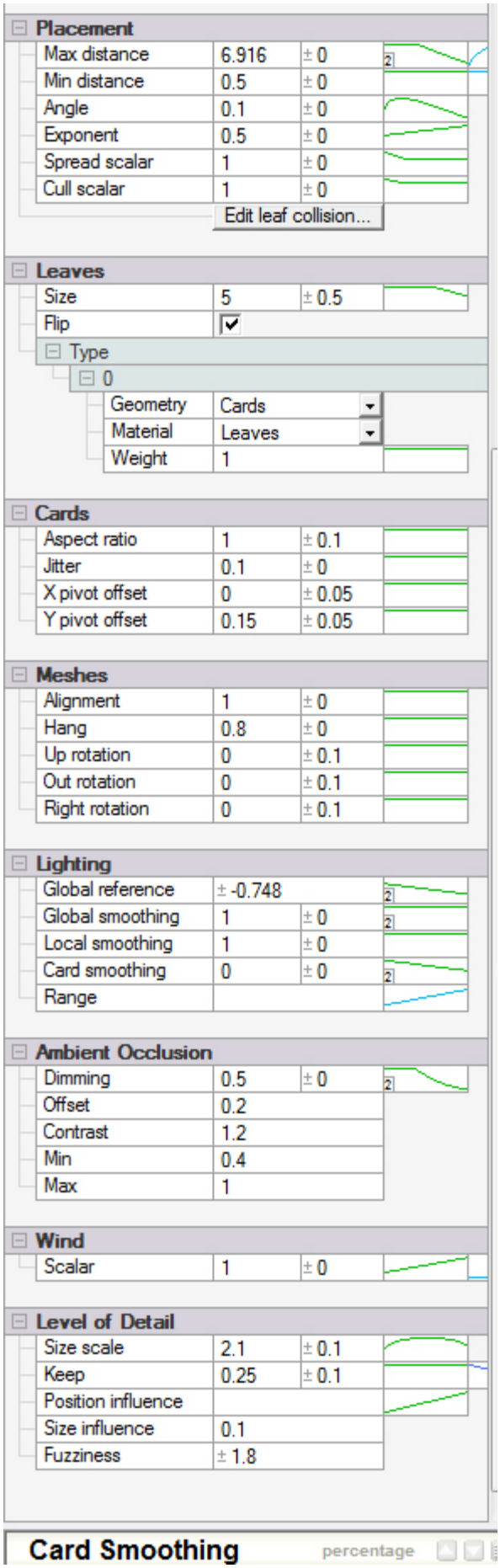


This took into account most of the parameters outlined but it took less than 5 minutes to get this thing looking pretty visually interesting. Now let's give it some life.

Step 7: Create another material and call it Leaves. Select a diffuse and normal from the height library and load it into the material. I chose the needles located at: Speedtree5\Trees\SpeedTree_Model_Library_v5.2_Full\Conifers\Scots_Pine\ScotsPineNeedles_2.tga

After those are loaded in right click on Level 2 in the generators panel and navigate to Leaves->Standard RT. On the left scroll down to the Leaves section, drop down the Type sub-menu and select the Leaves material from the dropdown. You should have something similar to this if you used the same texture. A breakdown of new parameters follows.


SpeedTree Modeler - [New SpeedTree]
File Edit View Window Tools Help
Object Properties - Leaves
Property Value Curves
Generation
Style Relative
First 0.15
Last 1
Frequency 3
Pruning
Size scalar 1
Degradation style Default
Orientation
Balanced
Alignment Align
Offset 0.1
Sweep 1
Roll 0
Steps
Enabled
Number 3
Spread 0.75
Per step frequ...
Masks
Forces
Allow forces
Random Seeds
Generation 31119
Placement 16510
Skin 357515232
Level of detail 2135462016
Randomize all
Placement
Max distance 6.916
Min distance 0.5
Angle 0.1
Exponent 0.5
Spread scalar 1
Cull scalar 1
Edit leaf collision...
Leaves
Size 5
Flip
Type
0
Geometry Cards
Material Leaves
Weight 1
Cards
Aspect ratio 1
Jitter 0.1
X pivot offset 0
Y pivot offset 0.15
Meshes
Alignment 1
Hang 0.8
Card Smoothing
percentage
Pushes leaf card normals away from the center of each leaf card.
Ready
New SpeedTree
TRIANGLES - SELECTED
branches: 0
caps: 0
fronds: 0
leaf cards: 952
leaf meshes: 0
mesh forces: 0
TOTAL: 952
bones: 0
proxies: 0
84.56
[2 cpu(s), 2 thread(s)], Last Compute 0.80 ms
Assets
Color Sets Materials Meshes Displacements Masks
Material Leaves
Color set None
Compiler usage Default
User data
Textures
Diffuse
Normal/Height
Specular/Trans.
Detail
Detail Normal
Enabled
RGBA
512x512
II\Conifers\Scots_Pine\ScotsPineNeedles_2.tga
Displacement Options
Smoothing 0
Generation
630
Level 2
62
Level 1
20
Trunk
1
Tree
Generation
Output

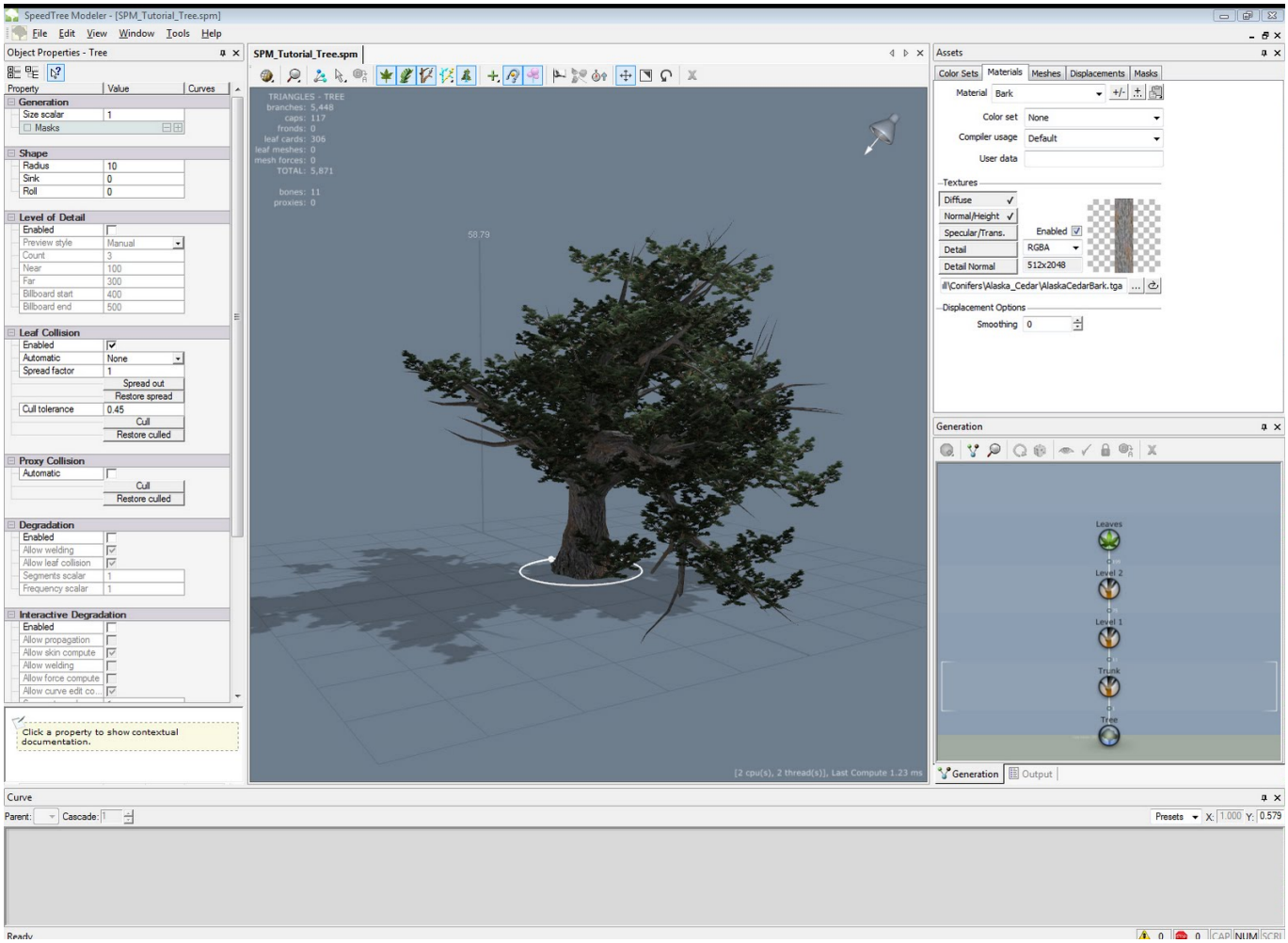

Placement
Max distance 6.916 ± 0
Min distance 0.5 ± 0
Angle 0.1 ± 0
Exponent 0.5 ± 0
Spread scalar 1 ± 0
Cull scalar 1 ± 0
Edit leaf collision...
Leaves
Size 5 ± 0.5
Flip
Type
0
Geometry Cards
Material Leaves
Weight 1
Cards
Aspect ratio 1 ± 0.1
Jitter 0.1 ± 0
X pivot offset 0 ± 0.05
Y pivot offset 0.15 ± 0.05
Meshes
Alignment 1 ± 0
Hang 0.8 ± 0
Up rotation 0 ± 0.1
Out rotation 0 ± 0.1
Right rotation 0 ± 0.1
Lighting
Global reference ± -0.748
Global smoothing 1 ± 0
Local smoothing 1 ± 0
Card smoothing 0 ± 0
Range
Ambient Occlusion
Dimming 0.5 ± 0
Offset 0.2
Contrast 1.2
Min 0.4
Max 1
Wind
Scalar 1 ± 0
Level of Detail
Size scale 2.1 ± 0.1
Keep 0.25 ± 0.1
Position influence
Size influence 0.1
Fuzziness ± 1.8
Card Smoothing
percentage

> Pushes leaf card normals away from the center of each leaf card.

As you can see there are a variety of new parameters to work with. The Generation area is the same as the other elements as well as some of the others but it's important to acclimate yourself with the new ones.

- Placement: Used to tweak the location of Leaf sprites.
  - Max distance: How far from the branches to spawn leaf cards (sprites).
  - Min distance: How close from the base of the branches to spawn leaf cards. (If you go too low Speedtree will crash).
  - Angle: Rotates the cards into different orientations.
  - Exponent: Randomly scatters leaves between the Min and Max values (I think).
  - Spread scalar/Cull scalar: Relates to the leaf collision settings. Click the edit leaf collision settings and play around with how it spreads and culls leaves based on if they're intersecting with other leaf cards.
- Leaves
  - Size: Controls the scale of each individual leaf card. (Almost always will need tweaked).
  - Flip: X-axis flip.
  - Type
    - Geometry: Select a leaf card mesh to apply if one is loaded.
    - Material: What material is applied to the leaf cards.
    - Weight: No idea, haven't seen this setting change anything.
- Cards
  - Aspect ratio: The ratio between width and height.
  - Jitter: Offsets the cards from their initial position.
  - X pivot offset: Self-explanatory.
  - Y pivot offset: Also self-explanatory.
- Meshes: These parameters control custom leaf sprite meshes. You probably won't use them too often. Play with them when you do and form your own conclusions about their utility.
- Lighting
  - Global reference: The area used to calculate global smoothing.
  - Global smoothing: Interpolation from 0 to 1 between local and globally smoothed leaf cards.
  - Local smoothing: Interpolation from 0 to 1 between individual and local smoothed leaf cards.
  - Card smoothing: Pushes normals around from the center to edges of the leaf card.

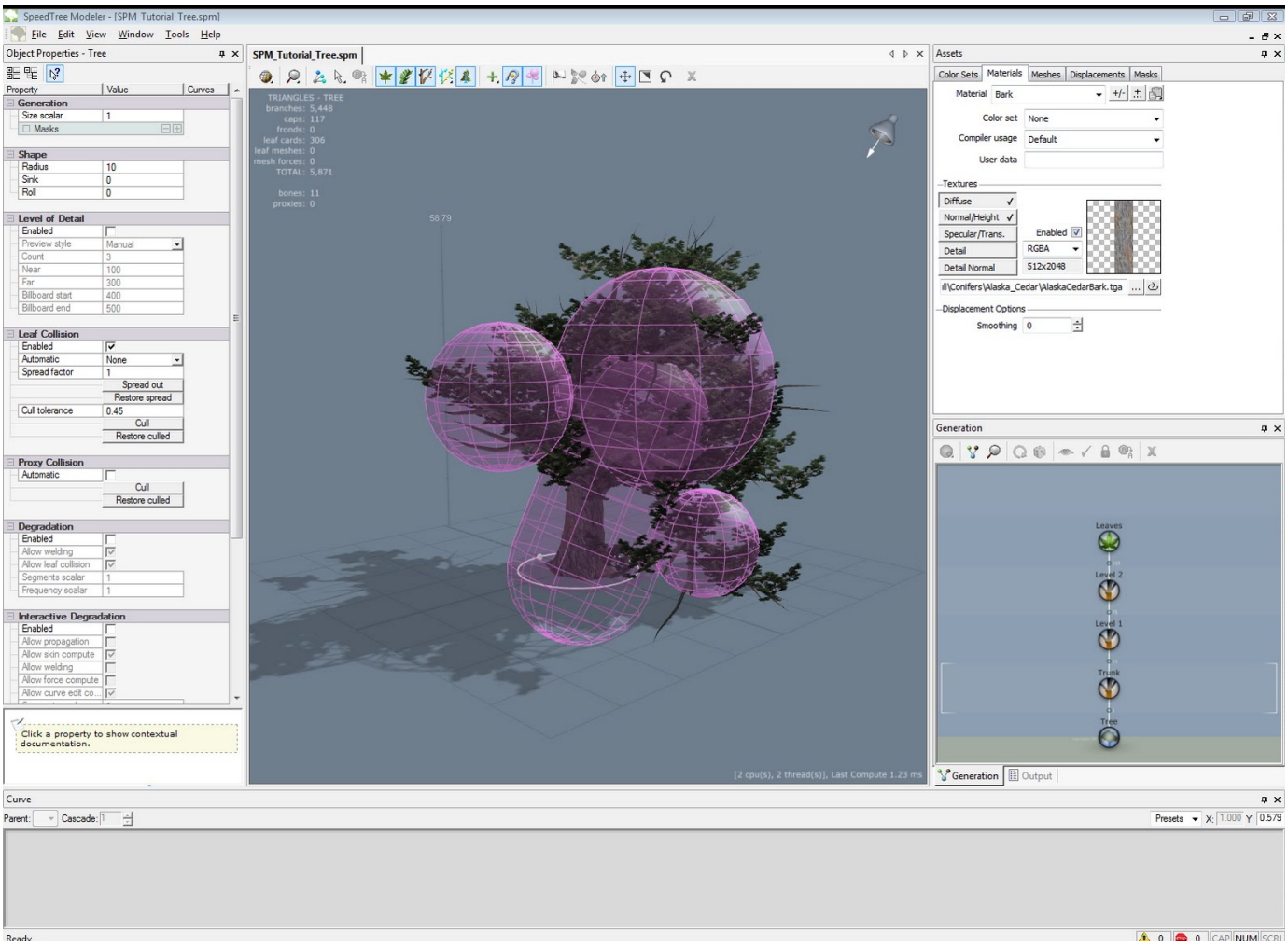
Play with all these new parameters and get something you like. Here's what I came up with:



Step 8: The tree is in a pretty good spot to prepare for the game engine. We need to generate collision for the tree and doing so is pretty simple. You can manually create collision by right clicking and picking either Sphere or Capsule under 'Add collision object.' Or you can automatically generate it from the Tools->Generate Collision Primitives from the top menu. Tweak these settings to try to get something close. If it's not close select individual primitives and use the parameters screen to scale them or move them around.

Note: For capsules, the hemisphere at the top and bottom are disregarded in Unreal so if the generated capsules are larger than you want they're probably just about right.

Toggle the visibility of your collision prims from the pink icon at the top of your center viewport. (It looks like 3 pink balloons).

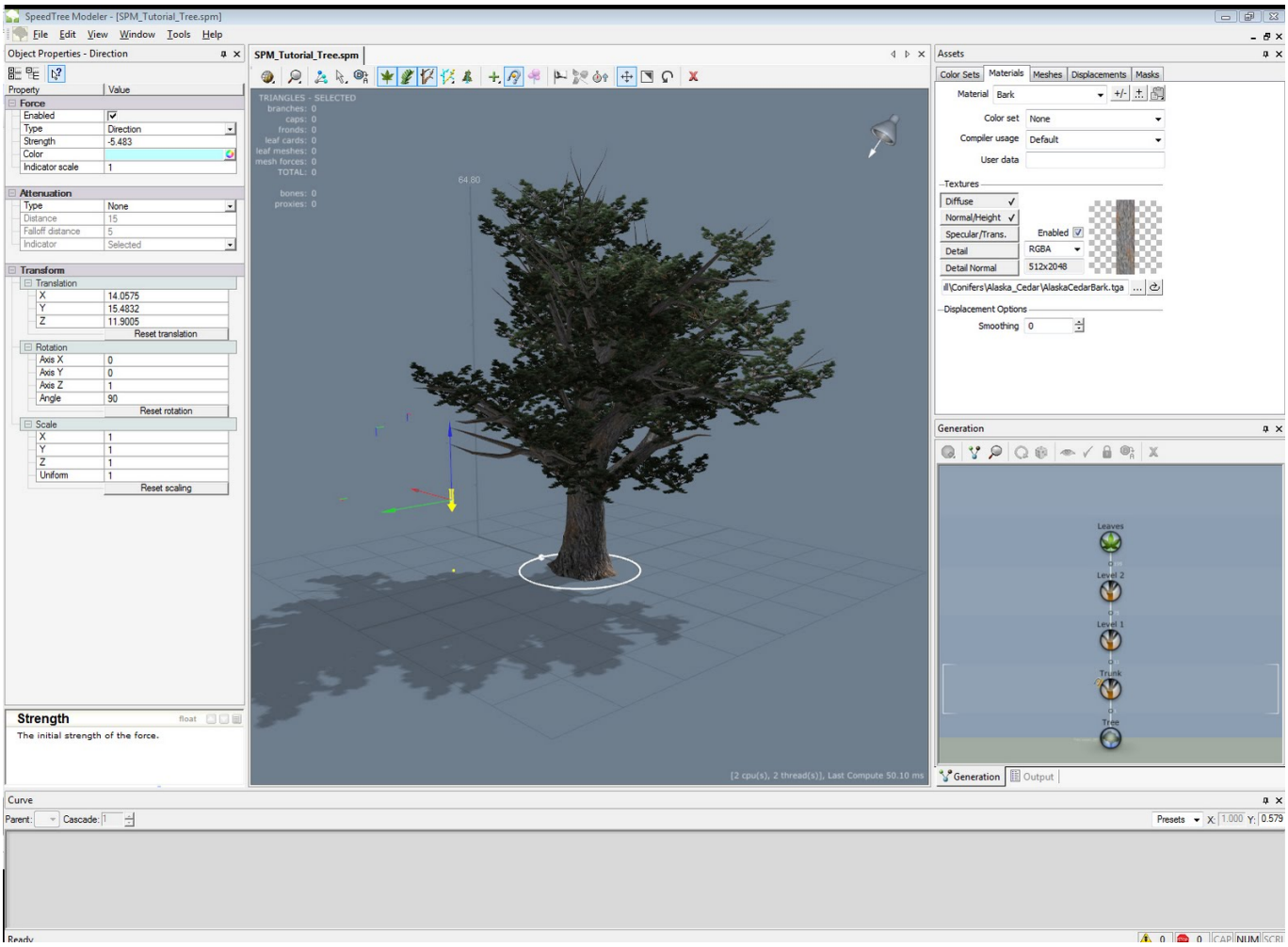


That pretty much does it for the SpeedTree Modeler. There are a variety of other cool features that I'll cover now. (Keep in mind you'll have to regenerate collision if you make significant tweaks).

Adding Forces: Forces behave sort of like Deformers in Maya or how Bend/Melt/etc modifers work in Max. To add a force select a node (for this demo select the Trunk) then right click in the viewport go to Add Forces->Direction and you should see a little downward arrow icon/widget pop up in your viewport where you clicked.

You'll get a variety of parameters on the right with it selected but the only one I tend to use is Strength but play around with the other ones to see what kind of effects you can get. I gave this direction force a negative value to straighten up my tree. It is much more believable and less gnarled and crazy looking than before.

Experiment with all the forces especially Twist and Gnarl, you can create some really cool stuff. Keep in mind they can be applied to any node and you can layer forces together, try adding some Twist forces to your branches and just have fun with it.



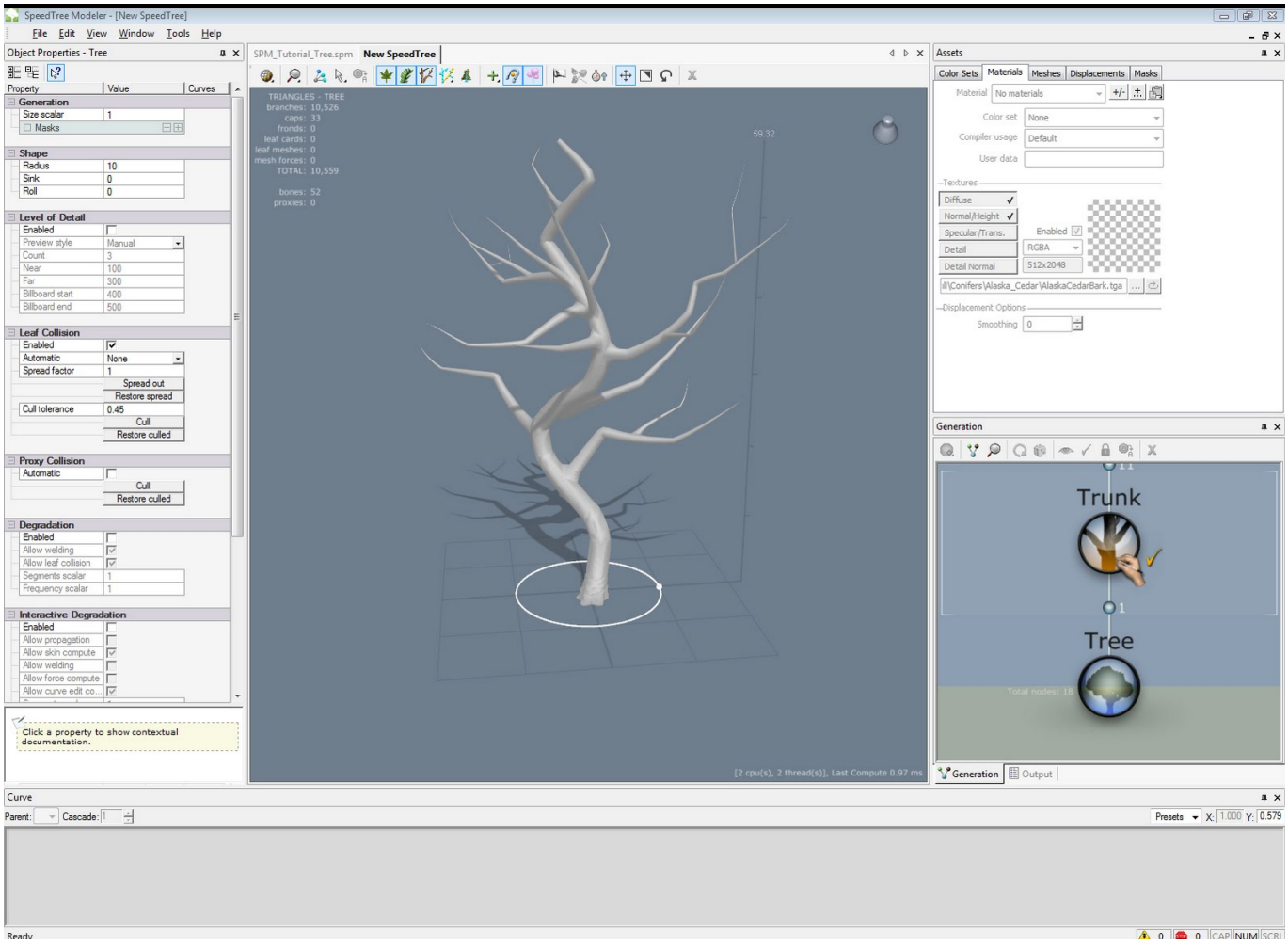
Save this tree as we will come back to it in a little while.

Hand Drawn Trees: This is really simple but extremely powerful. The hand-painting tools allow you to paint very complicated and/or specific shapes while maintaining the power of SpeedTree's parameters.

Create a new document and right click on your Tree Master node go to Hand Drawn->Trunk. Now in the viewport hold Spacebar (keep it held down) and click-drag to draw your shape. You can also use a tablet which is awesome.

For branches you can just click on the trunk and draw outward and it will automatically add your branch nodes, same thing for drawing level 2 branches. Manually you can just right click on your Trunk node and go to Hand Drawn->Branch.

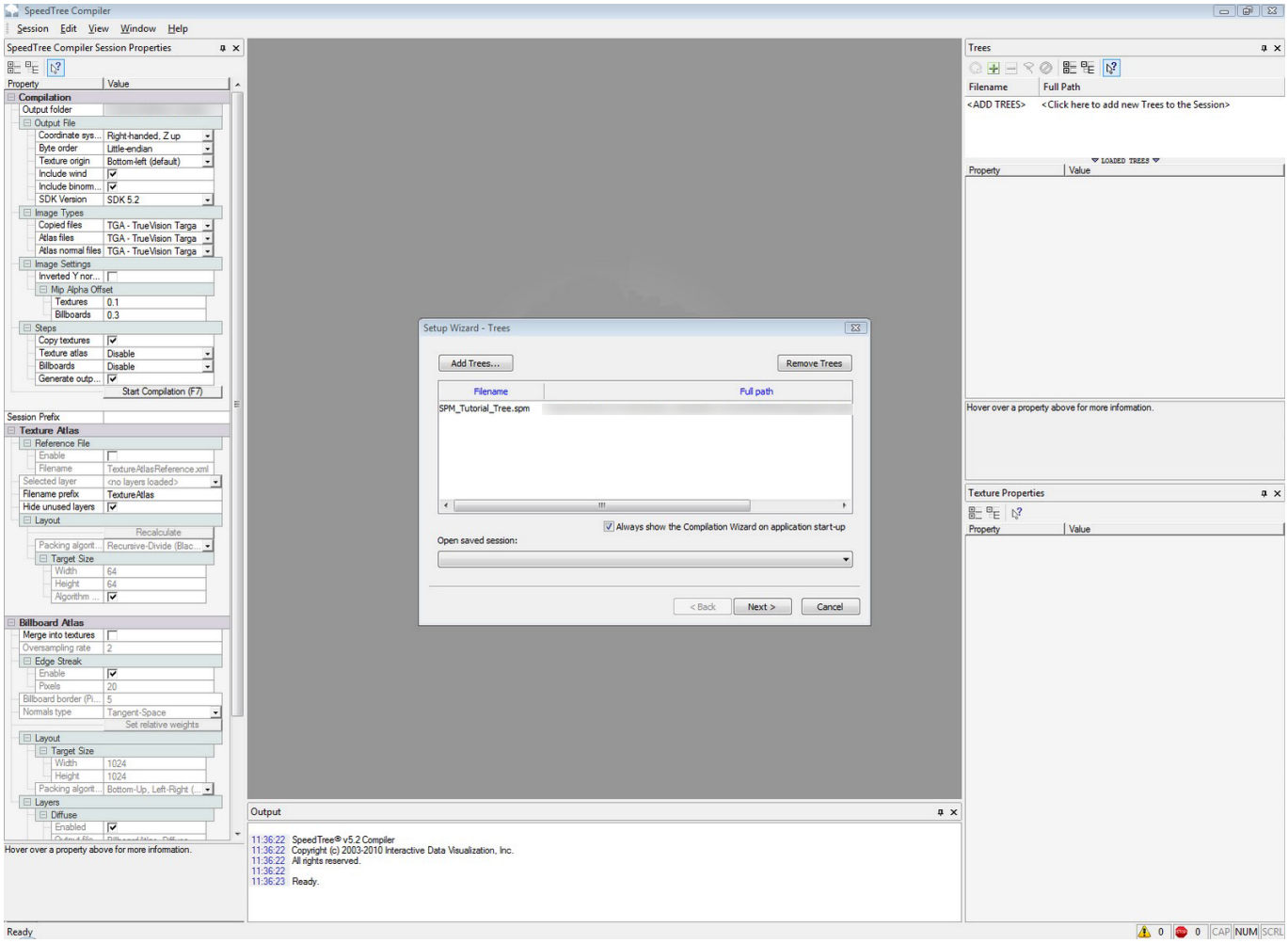
Experiment with this and see what you can get. Keep in mind Hand Drawn trees almost always need significant optimization so those Length, Radial and Optimization parameters under Segments will be very important. Here's what I got in only about a minute.



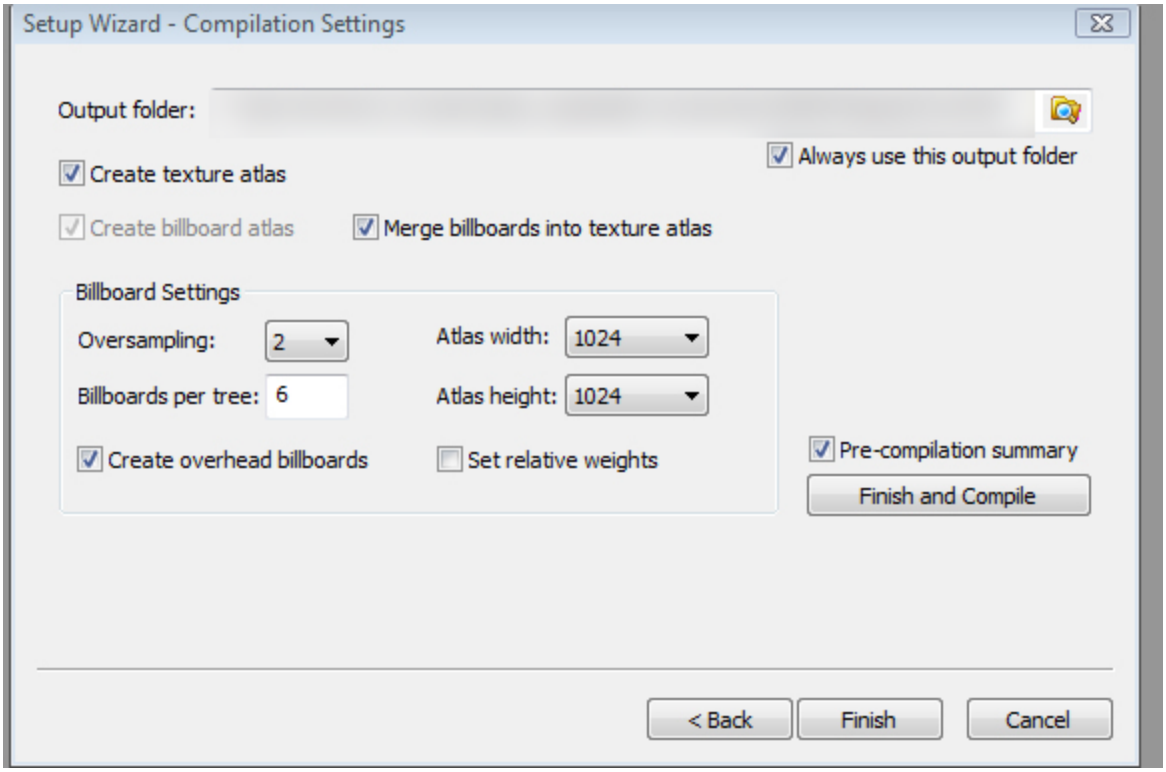
You don’t have to hand draw everything. You can take advantage of the procedural stuff as well such as hand-drawing a very interesting looking trunk but then using Standard RT branches.

Step 9: Now it’s time to get into the SpeedTree Compiler and get your tree ready for Unreal. Re-open your tree from before and if you need to re-generate collision from experimenting with forces do that. Once you’re done save the tree and open the Speed Tree Compiler.

You’ll see something like this. (I already clicked Add Trees... and selected our SPM file that we saved out).



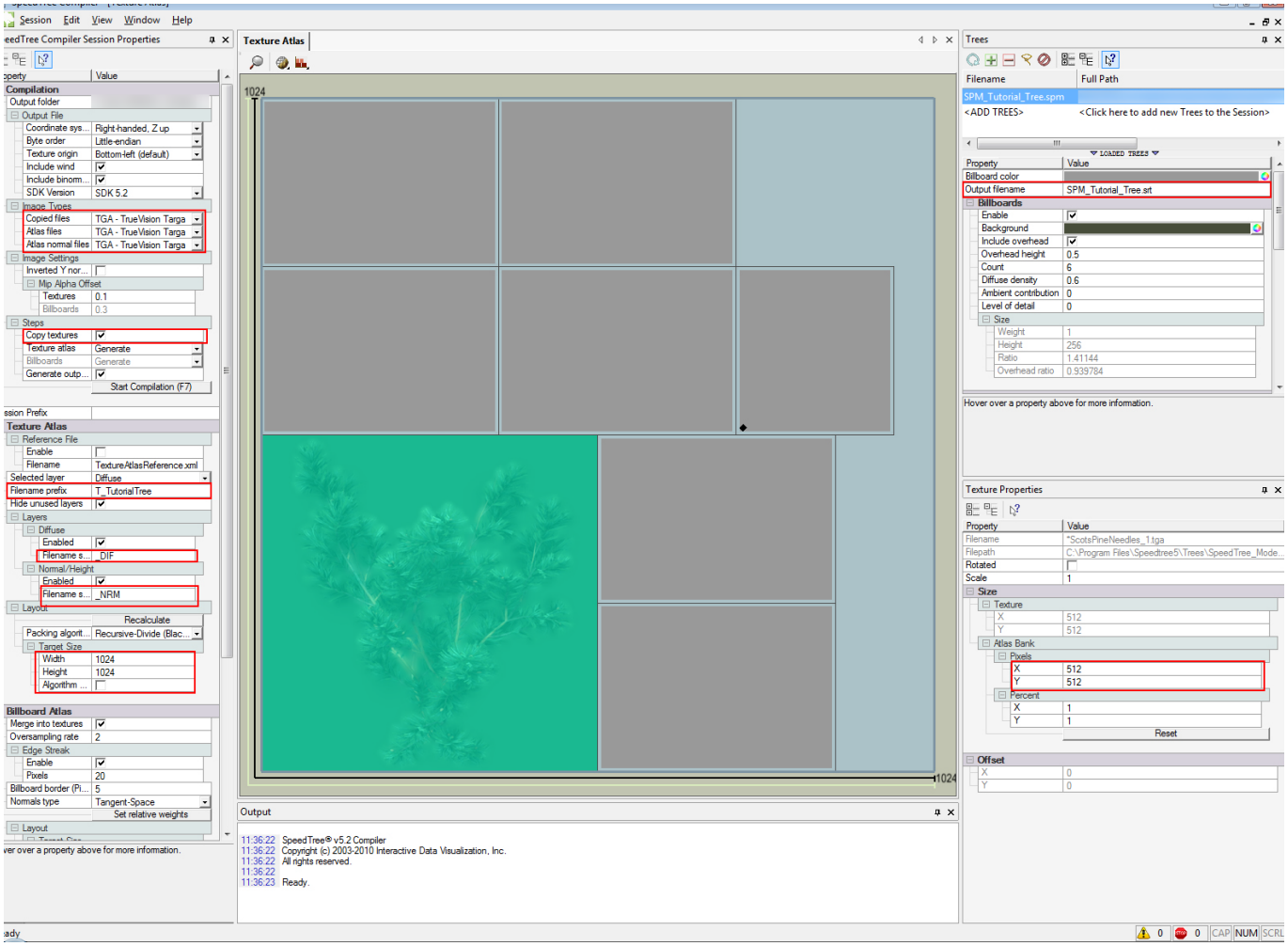
Step 10: After you add your tree to the list click ‘Next’ and set your output folder. Make sure your other settings match these:

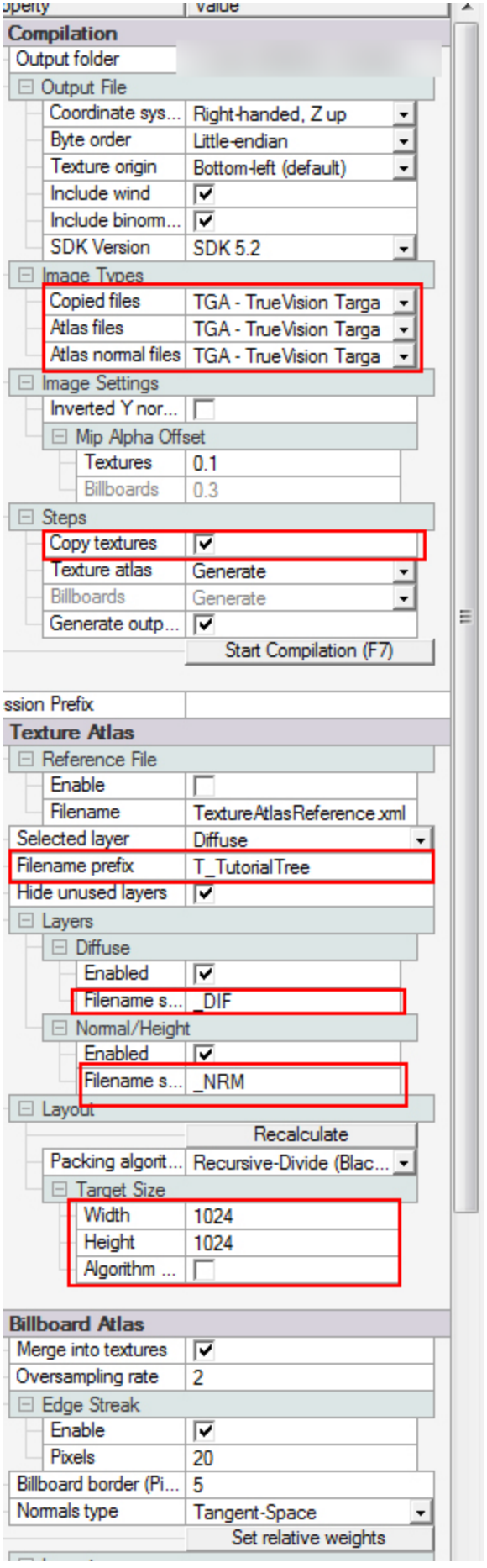


Step 11: You should see your working texture that's going to export in the middle and see a similar interface to the SpeedTree Modeler except there's parameters everywhere. The left side parameters handle the general compile settings for when you compile. The settings on the right handle selections for tweaking.

For this tree the billboard and sprite proportions are pretty good but a lot of times a sprite will be too big or the billboards too small so you change the X and Y values under Size->Atlas Bank on the right to make the sprite smaller.

The highlighted settings tend to get changed. The rest of the settings for the most part stay at their default values.

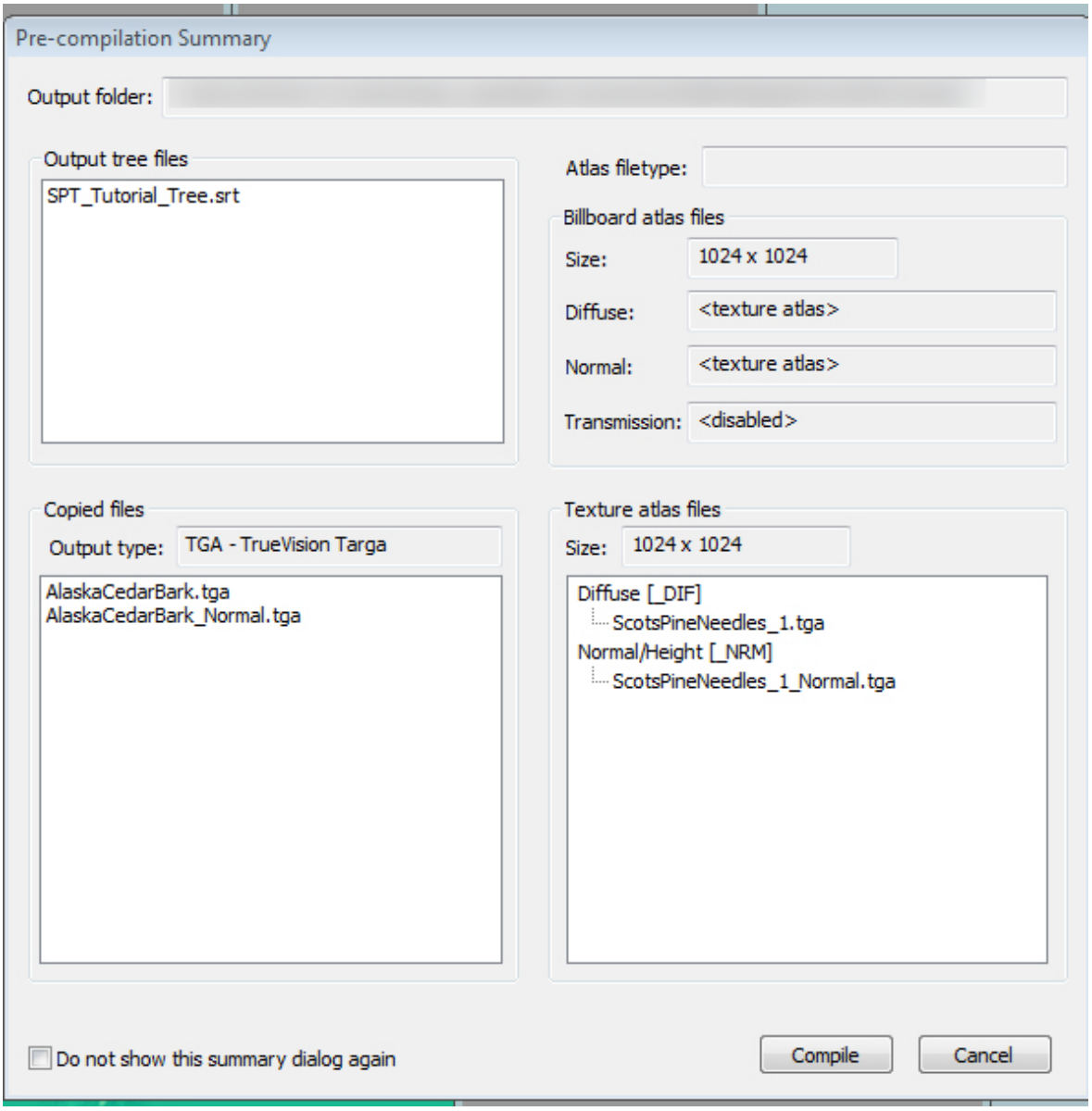

Session Edit View Window Help
eedTree Compiler Session Properties
Compilation
Output folder
Output File
Coordinate sys... Right-handed, Z up
Byte order Little-endian
Texture origin Bottom-left (default)
Include wind
Include binorm...
SDK Version SDK 5.2
Image Types
Copied files TGA - TrueVision Targa
Atlas files TGA - TrueVision Targa
Atlas normal files TGA - TrueVision Targa
Image Settings
Inverted Y nor...
Mip Alpha Offset
Textures 0.1
Billboards 0.3
Steps
Copy textures
Texture atlas Generate
Billboards Generate
Generate outp...
Start Compilation (F7)
ssion Prefix
Texture Atlas
Reference File
Enable
Filename TextureAtlasReference.xml
Selected layer Diffuse
Filename prefix T_TutorialTree
Hide unused layers
Layers
Diffuse
Enabled
Filename s... _DIF
Normal/Height
Enabled
Filename s... _NRM
Layout
Recalculate
Packing algorit... Recursive-Divide (Blac...
Target Size
Width 1024
Height 1024
Algorithm ...
Billboard Atlas
Merge into textures
Oversampling rate 2
Edge Streak
Enable
Pixels 20
Billboard border (Pi... 5
Normals type Tangent-Space
Set relative weights
Layout
ver over a property above for more information.
Texture Atlas
1024
1024
Output
11:36:22 SpeedTree® v5.2 Compiler
11:36:22 Copyright (c) 2003-2010 Interactive Data Visualization, Inc.
11:36:22 All rights reserved.
11:36:22
11:36:23 Ready.
Trees
Filename Full Path
SPM_Tutorial_Tree.spm
<ADD TREES> <Click here to add new Trees to the Session>
LOADED TREES
Property Value
Billboard color
Output filename SPM_Tutorial_Tree.srt
Billboards
Enable
Background
Include overhead
Overhead height 0.5
Count 6
Diffuse density 0.6
Ambient contribution 0
Level of detail 0
Size
Weight 1
Height 256
Ratio 1.41144
Overhead ratio 0.939784
Hover over a property above for more information.
Texture Properties
Property Value
Filename "ScotsPineNeedles_1.tga
Filepath C:\Program Files\Speedtree5\Trees\SpeedTree_Mode...
Rotated
Scale 1
Size
Texture
X 512
Y 512
Atlas Bank
Pixels
X 512
Y 512
Percent
X 1
Y 1
Reset
Offset
X 0
Y 0
0
0
CAP
NUM
SCRL


Property
Value
Compilation
Output folder
Output File
Coordinate sys...
Right-handed, Z up
Byte order
Little-endian
Texture origin
Bottom-left (default)
Include wind
Include binorm...
SDK Version
SDK 5.2
Image Types
Copied files
TGA - TrueVision Targa
Atlas files
TGA - TrueVision Targa
Atlas normal files
TGA - TrueVision Targa
Image Settings
Inverted Y nor...
Mip Alpha Offset
Textures
0.1
Billboards
0.3
Steps
Copy textures
Texture atlas
Generate
Billboards
Generate
Generate outp...
Start Compilation (F7)
ssion Prefix
Texture Atlas
Reference File
Enable
Filename
TextureAtlasReference.xml
Selected layer
Diffuse
Filename prefix
T_TutorialTree
Hide unused layers
Layers
Diffuse
Enabled
Filename s...
_DIF
Normal/Height
Enabled
Filename s...
_NRM
Layout
Recalculate
Packing algorit...
Recursive-Divide (Blac...
Target Size
Width
1024
Height
1024
Algorithm ...
Billboard Atlas
Merge into textures
Oversampling rate
2
Edge Streak
Enable
Pixels
20
Billboard border (Pi...
5
Normals type
Tangent-Space
Set relative weights

Take a look at the ‘Copy textures’ checkbox. What this will do is make a copy of the Bark textures you used and place them in your output directory. This will be useful for your first few trees, but if you start re-using finished Bark materials in Unreal you will need to uncheck it.

Step 12: Set up your settings to match the ones above. The filename prefix can be whatever. The final filename will be for example: T_TutorialTree_DIF and T_TutorialTree_NRM when it generates the two maps.

Step 13: Once all of your settings are in place click ‘Start Compilation (F7)’. It should bring up the following dialogue detailing what it’s going to create and copy (if you have ‘Copy textures’ checked). If everything look correct click ‘Compile”



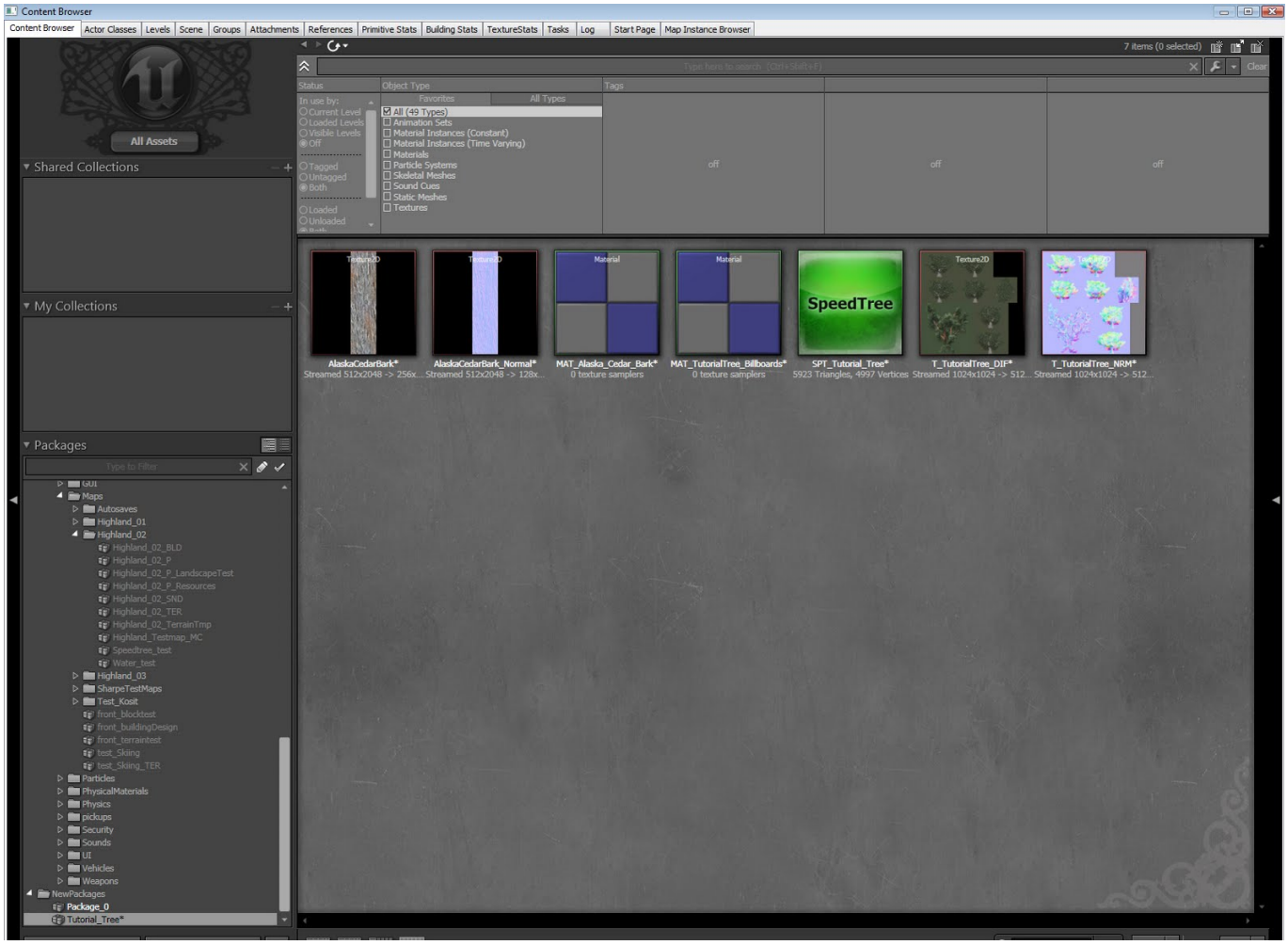
The ScotPineNeedles_1 will be duplicated and renamed to the name you set in the parameters: T_TutorialTree_DIF/T_TutorialTree_NRM

The compiler will run and the compile should run smoothly, after it does click Done and launch Unreal.

BILLBOARD NORMAL MAPS: A lot of time the normal maps on the billboards are too intense by default and because of that they light very bad in Unreal. They need to be manually adjusted. Take them into photoshop, select the 6 (or 7 if you did overhead too) billboard and fill those areas with normal blue.

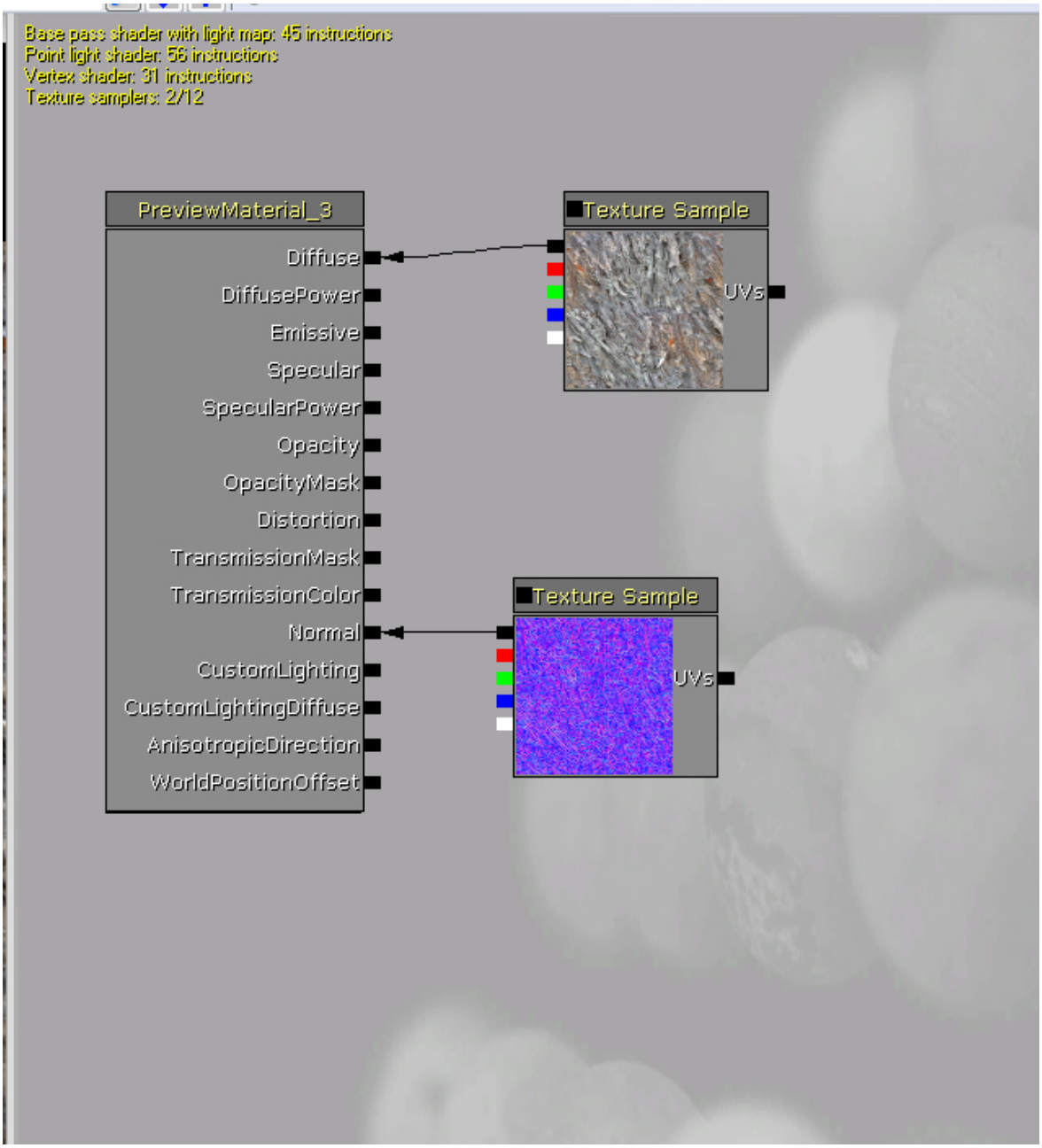
Alternatively, you can make a new layer on the normal map, select all the billboards then go to Layer->Layer Mask->Reveal Selected. Copy the normal map then go into CrazyBump and choose Paste Normal Map, drag the intensity to between 1 and 10 then copy that to clipboard. Take that into Photoshop, paste it above the mask layer then press CTRL+E and choose to Preserve the layer mask. You should see the billboards now replaced with the less intense normal while maintaining the intensity of the sprite areas of the normal map.

Step 14: Import everything into a package (for this tree there should be 5 total assets). REMEMBER TO CHANGE THE COMPRESSION SETTINGS FOR THE TWO NORMAL MAPS. Then create two materials: MAT_Alaska_Cedar_Bark and MAT_TutorialTree_Billboards.

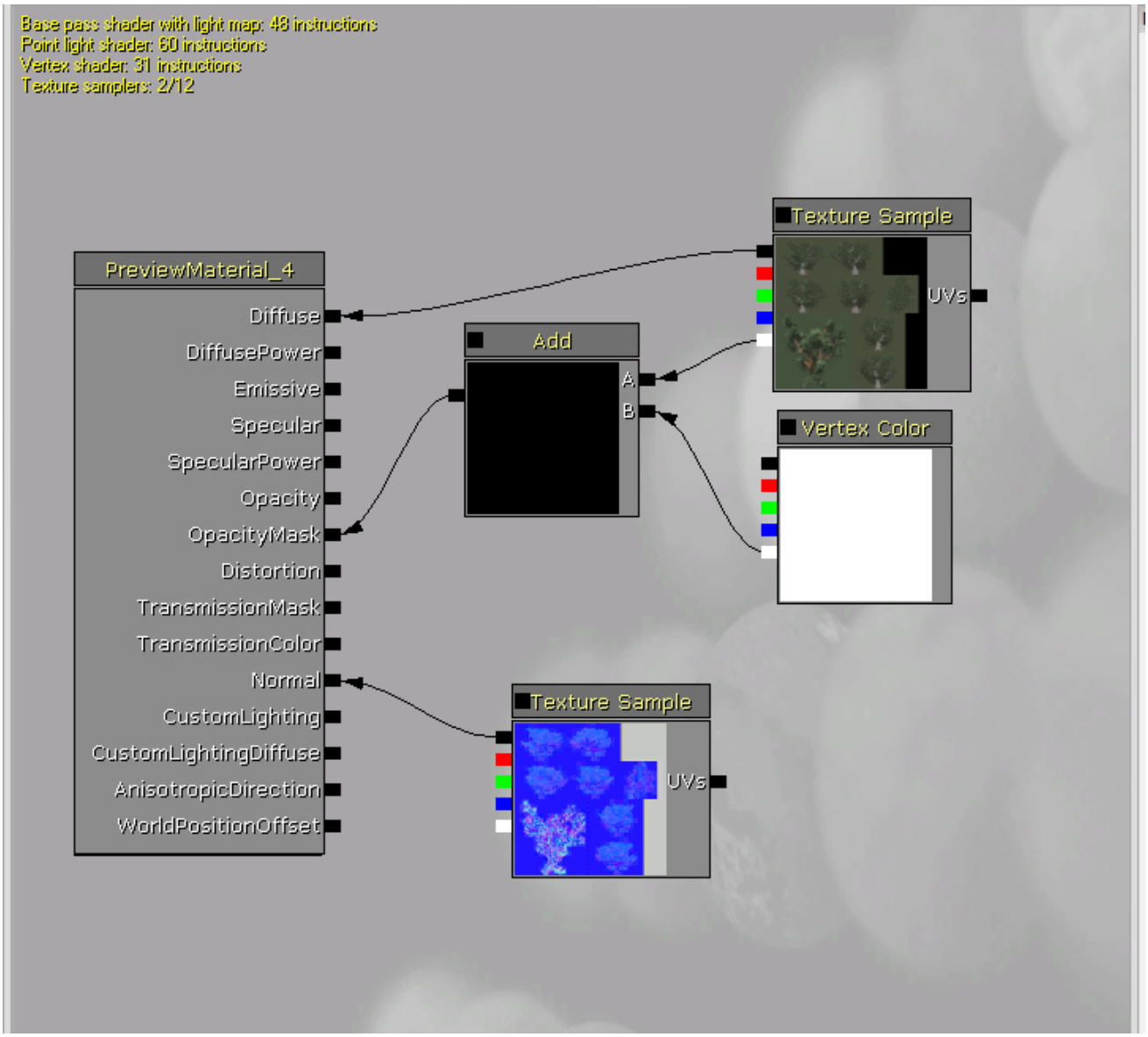


Step 15: Double click the MAT_Alaska_Cedar_Bark to launch the Materials Editor and re-create this simple material set-up for your bark.

Plug the AlaskaCedarBark into the Diffuse channel and the AlaskaCedarBark_Normal into Normal.

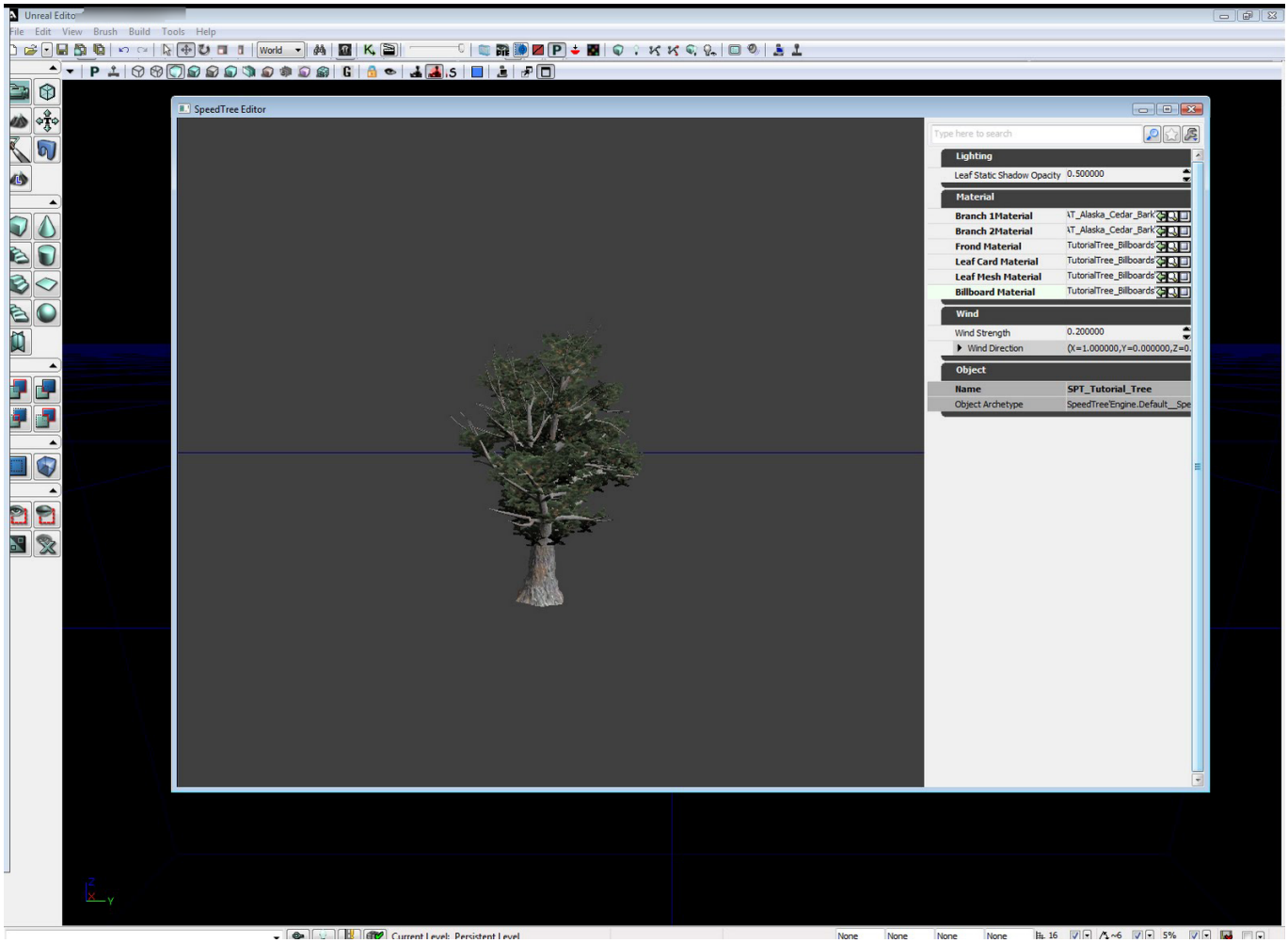


Step 16: Save that material and close it. Now double click the MAT_TutorialTree_Billboards. Set the Blend mode to BLEND_Masked and re-create the following material set-up:

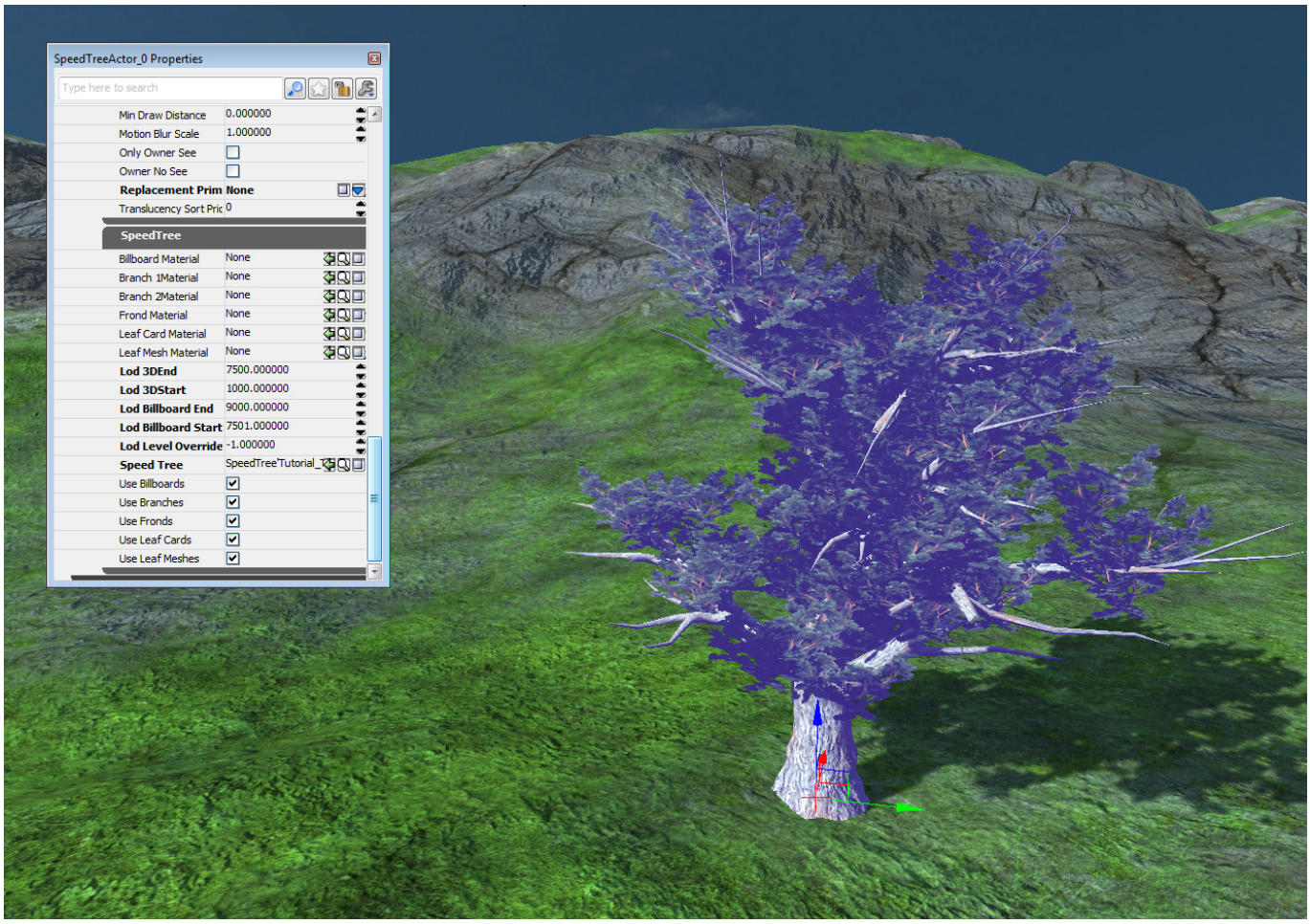


Plug T_TutorialTree_DIF into Diffuse and T_TutorialTree_NRM into Normal. Plug an Add into OpacityMask then plug the Alpha of your Diffuse into A and plug the Alpha of a Vertex Color into B. Make sure the Blend Mode is set to BLEND_Masked

Step 17: Save the Material and close the Materials Editor. Double click the SpeedTree Actor to launch the SpeedTree Editor. You will see 6 Inputs for different Materials. You want to plug MAT_Alaska_Cedar_Bark into the first 2 inputs and plug MAT_TutorialTree_Billboards into the remaining 4. You should see something like this:

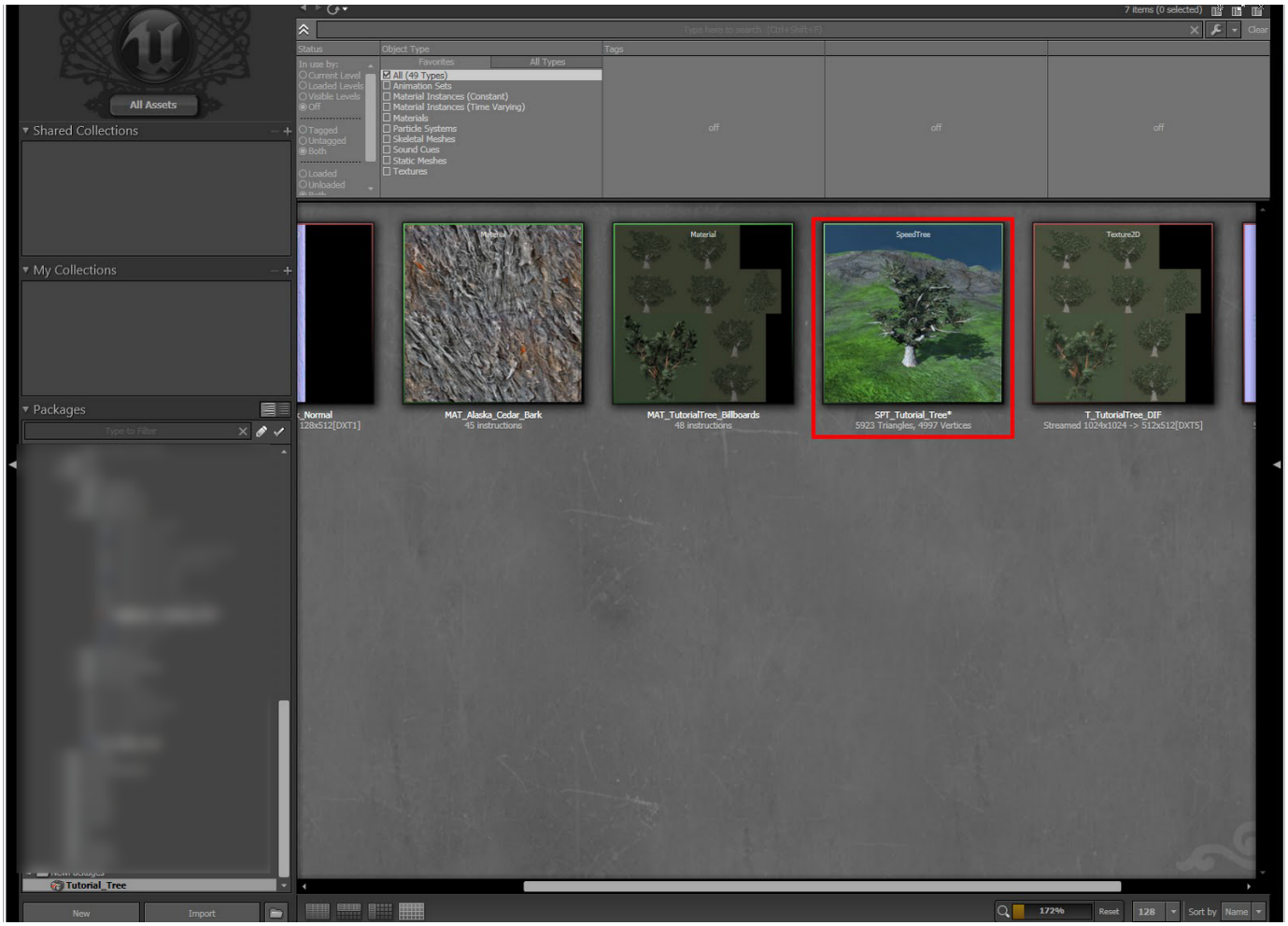


Step 18: Place the tree in a level and press F4 to bring up the properties. Drop down the SpeedtreeActor properties and go down to the SpeedTree section. In here you can override material settings on a per-instance basis for variation. You can also check off Leafs or Branches if you want a leafless or branchless version of the same tree. Take into account how big the level the tree will be in and adjust your LOD settings accordingly. A lot of the time you will want to set LOD Override to -1.



Step 19: SpeedTrees by default are very small, so at this point you might not even see your tree. The tree shown in the screenshot above has a scale of 20 which can be set in the number boxes at the bottom of the Unreal interface.

Step 20: To make it easy to see a lot of trees in your content browser you're going to want to make a thumbnail. Place it in a fairly blank level and maximize your perspective view. Now focus your tree in the shot. Right click on the SpeedTree actor in your Content Browser and click 'Capture Thumbnail from Active Viewport.' It should automatically update the thumbnail in the Content Browser to look something like this:



Step 21: Build lighting on your tree, go in-game and test the LODs out and collision to make sure everything is working how you intend it to. If everything checks out you should be good to go.

Congratulations you're done. Hopefully your tree won't get damaged by a 'nader.