

Tree[d] → SketchUp Tutorial

by Wyatt Thompson

November 11, 2010

Ruby Plugins Required:

1. [OBJ Importer](#) (TIG)
2. [TT-Cleanup](#) (ThomThom)
3. [UV Toolkit](#) (ThomThom)
4. [Loose to Groups](#) (Chris Fullmer)
5. [Component/Group Tools](#) (Matt666)
6. [Component Comparison](#) (ThomThom)

Thank you to all the ruby authors who make this process possible!

The first 5 steps are required to import a tree component into SketchUp with textures.

1. Create a tree in tree[d].
2. Export to obj.
3. Copy the required png/jpg textures from the tree[d]/texture folder into the same folder as the obj and mtl created in the previous step. You can open the mtl file in a text editor to see which images are required.

Example file structure: File folder

- *.obj
- *.mtl
- *.png

4. Start a new SketchUp file.
5. From the Plugins menu, select OBJ Importer - Import OBJ [with textures]. Select the scale (feet or meters works best for trees). Select the proper YZ axes orientation.

The following steps attempt to make the file size of each tree component more manageable. These steps will take time to process so you may need to weigh the time spent vs the need for a more manageable model.

6. Clean up the model - From the Plugins menu, select TT-cleanup – Clean-up Entire Model. Select Yes for all the dialogs.
7. Create groups - Group the trunk/branches and leaves separately. Select by material is a fast way to do this, but there are several methods/plugins that you could use.
8. Paint the leaves - Open the Leaf group, select all the leaves, and select UV Toolkit from the Plugins menu. Select Frontface to Backface.
9. Simplify leaves – Select all the leaves and select Loose Geometry to Groups from the Plugins menu. With the groups still selected, run Convert in Single Components from the Plugins menu. Choose Yes when asked to convert all the entities to components. (You may have to run it multiple times until you get the option to select Yes, since the plugin doesn't seem to work 100% of the time.) With the leaf components still selected, click Similar Objects from the Plugins menu and click Replace Similar Objects to eliminate redundant components. Be patient because these operations will take a while.
10. Purge the model.
11. If you don't plan to render, edit a leaf component and hide the edges.

Ultimately, the file size will depend on the complexity and variation of the tree created in tree[d]. To create the smallest 3D tree, turn Leaves Off and set Cross-Geometry to On. When individual leaves are necessary (for a foreground tree, for example), start with the lowest number of leaves suitable for your purpose and set the leaf-size minimum/maximum to the same number. This will allow you to create a single leaf component instead of hundreds in Step 9.

Using the default tree, I was able to obtain the following results.

Steps 1-5	6MB	12,000 faces
Step 6-10	3.5MB	7,400 faces
Leaf Size Min/Max = Same	2.5MB	7,000 faces
Cross-geometry On	1.8MB	2,100 faces

Known Issues:

Tree[d] utilizes vertex colors to create gradations in the color of the leaves based on the color palette you select in the Branch dialog. Each vertex of the leaf face in the tree[d] model is assigned a color, and the colors blend together across the face of each leaf. At this time, SketchUp is not able to import vertex colors so the color variations are not possible to recreate (at least using the above method; it may be possible using a different export format from tree[d]). There are two possible workarounds:

1. Export a png image from tree[d] and create a 2D face-me component in SketchUp.
2. Modify the leaf texture in SketchUp and use it to repaint the leaf components. This won't have the same gradation effects, but it can be used to introduce variation into the model. Use [Random Select](#) by TIG to select random leaf groups/components to modify.