

9. In another part of the scene or a new scene, create another 2D circle, this one about 0.5 units in diameter.
10. Click the **Macro/Sweep** tool in the **Polygon Sweep** toolbar. The last path used, if any, will automatically be attached to the circle. Open the path library again, and click the path we just added. Click the Macro/Sweep tool again to execute the sweep. After sweeping, right-click the Macro/Sweep tool, and enable the **Bend** option.



3.17 Geometry Paint



Geometry Paint tool

The **Geometry Paint** tool, located next to the **3D Paint** tool on the main toolbar, is useful for pasting copies of one or more objects onto the surface of another object. For example, you could paint trees onto a landscape, or feathers on a bird. You can add some randomness to the copies by instructing the tool to vary the scale of the copied objects, or even vary the objects themselves by selecting multiple objects as the paint source.

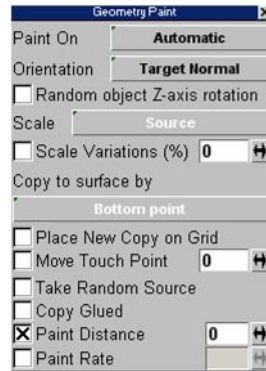
Note: Although the source object can be anything, the target must be a **polyhedral** object. For example, you can paint a NURBS object onto a polyhedron, but not vice-versa.

To use the **Geometry Paint** tool:

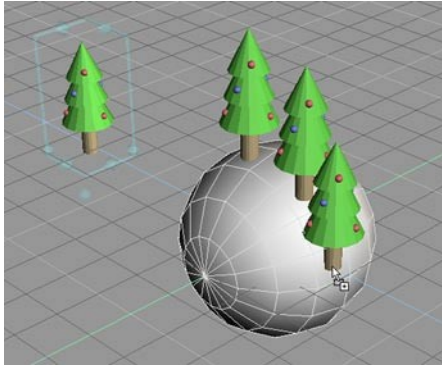
1. Select the object to be used as the “paint” source.

2. Activate the **Geometry Paint** tool.
3. Paint with the source object by doing one of the following:
 - Click a point on the surface of the target object to place a copy of the source object at that location.
 - Click and drag to create and move a copy of the source object on the surface of the target object. While dragging, the copy will follow the geometry of the target. Release the mouse button to stop dragging and place the copy.
 - Click and drag while holding down the CTRL key to paint continuously. The CTRL key may be pressed or released anytime while dragging to toggle continuous painting mode.

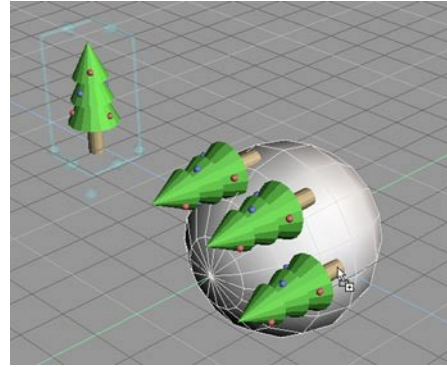
Right-click the Geometry Paint tool to open the **Geometry Paint settings panel**.



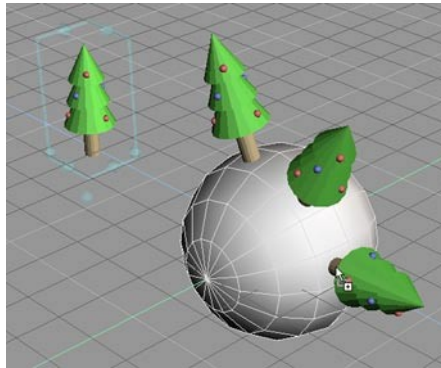
- **Paint On:**
 - **Object:** Source object is placed on surface of target object.
 - **Grid:** Source object is placed on grid.
 - **Automatic:** Clicking on an object will act like “Paint On Object.” Otherwise, it will act as “Paint on Grid.”
- **Orientation:**
 - **Source:** No rotation is performed. The placed object inherits the same rotation as the source object.
 - **Target:** The placed object inherits the same rotation as target object.
 - **Target Normal (default):** The orientation of the placed object is the same as the normal at the touch point on the target object.



Orientation: Source



Orientation: Target

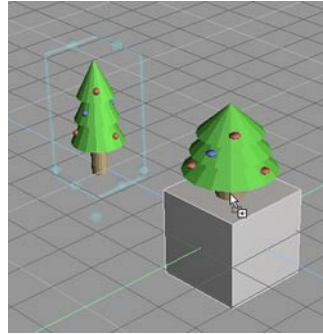


Orientation: Target Normal

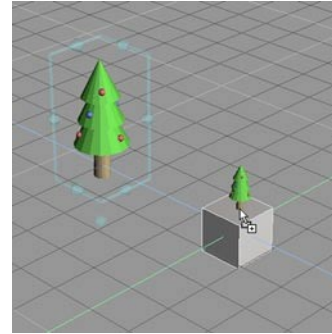
- **Random object Z-axis rotation:** When enabled, the placed object is randomly rotated about its Z-axis.
- **Scale:**
 - **Source:** No scaling is performed. The placed object inherits the same scale as the source objects.
 - **Target:** The placed object inherits the same scale as target object.
 - **Target, Proportional:** The placed object inherits the same scale as target object and keeps the source object's proportions.



Scale: Source



Scale: Target



Scale: Target Proportional

- **Scale Variations:** When checked, the placed object is randomly proportionally scaled up or down. The value represents the maximum scaling in percent.



Scale Variations

- **Copy to surface by:** There are two possible settings for mapping the touch point of the placed object onto the target object:
 - **Center Point:** The center of placed object is mapped onto the touch point.
 - **Bottom Point:** The bottom of placed object is mapped onto the touch point.
- **Place New Copy on Grid:** This works only when the source object is painted on the grid. If checked, the copied object is placed on the grid with its Z coordinate set to zero. Otherwise, the copies will have the same Z co-ordinate as the original object.
- **Move Touch Point:** When checked, the position of the placed object is adjusted by the value specified, in the Z-direction of the placed object.
- **Take Random Source:** If a group of objects is selected as the paint source, and Take Random Source is checked, the placed object is chosen randomly from one of the objects in the group, instead of placing the whole group. The selection of objects from the group is limited to top-level.



Random Source

- **Copy Glued:** If checked, all the copied objects are converted into a group after exiting the tool.
- **Paint Distance:** If checked, you can define the offset used for CTRL+drag object painting. If the offset is set to zero (default), the distance between copies will be taken from the distance between the original object and the current copy.
- **Paint Rate:** If checked, you can define how often the object should be copied in object painting mode. A value of 1 means that the object is copied on every single CTRL+drag step. The higher this value, and the slower you move the cursor, the less space there will be between each copy.

3.18 Modeling with Procedures

New command-based architecture in trueSpace makes it possible to create objects procedurally – i.e. executing a series of commands that automatically create your object instead of performing modeling tasks manually. There are many advantages to defining an object procedurally:

- **Compact:** A three-line procedure can create an entire forest of trees that can be stored in a small fraction of the space that polygonal representations would require.
- **Dynamic:** You can easily add parameters (for instance “age” for an Oak tree) to a procedural object which can be animated and constrained to other variables to change the nature of an object.
- **Flexible:** You can create procedural objects in a number of ways (plugins, scripts, Link Editor structures), and you can define exactly how they work to fit the specific needs of your project.

You can create procedural objects in trueSpace via a couple of different methods. One is to use a **script object**. Script objects can define a model by creating a mesh from scratch, or by creating and attaching sub-objects (e.g. creating a series of spheres to represent a molecule). By exposing various script attributes you can let the user of your object easily change how the object is created. ConeObj, GearObj,