# Thea Render Instancing Tutorial

<table>
<thead>
<tr>
<th>Revision</th>
<th>Author</th>
<th>Reason for Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>11/09/12</td>
<td>Christina Psarrou</td>
<td>Initial version.</td>
</tr>
</tbody>
</table>

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Copyright © Solid Iris Technologies
INTRODUCTION

In computer graphics, geometry instancing is the practice of rendering multiple copies of the same object in a scene at once\(^1\). This technique is primarily used for objects such as trees, grass or other models which can be represented as repeated geometry without appearing unduly repetitive. So, whenever there is a need to render the same object in many copies, the instancing method can be used.

Figure 1: Instancing with Thea Render (render by Juan Carlos Uribe)

INSTANCING AND THEA RENDER

Thea Render is a render engine that allows users to render multiple copies of selected objects, with the use of its embedded instancing brush. You can find this tool along with its options at the Settings panel, at Tools tab (see figure 2).

As it is seen in figure 2, the user can choose the desired instance -object- to generate and the canvas -surface- on which the copies will be placed. Apart from these basic selections, a lot of other options, allow the user to create more effects for fulfilling specific needs, such as different sizes, directions or angles of the new copies. Your created instances, will be saved in a Package and you will be able to see it at the Tree View.

For deeper understanding of each parameter, an analytic description is following, along with some visual examples that will help to distinguish the results and the effects that each option can lead to.

Figure 2: Instancing Tool in Thea Render
**Options Analysis & Case Study Example**

**Selection**

**Instance:** this option, allow the user to specify the object that will be generated by the program (initial object). After selecting the desired object by clicking on it at the Viewport or at the Tree View, just click at the rotating arrows button at the left side of the instance selection, in order to apply your object to instancing tool (you will now see its name next to it like on figure 3).

![Figure 3: Selection of Instance](image)

**Canvas:** this is the main surface where the instances will be placed. After selecting your desired object that will be used as a canvas, click at the rotating arrows next to canvas option to apply it (as you see in figure 4).

![Figure 4: Selection of Canvas](image)

**Tip:** instances are placed on the canvas according to their **Pivot Point**. Most of the times, it is more useful to place pivot point of the instance at the point that will be adapted to the canvas (for example at the bottom of a grass object). In order to change your pivot point, while having your object selected, press “p” key to enter in the pivot mode, change your axes accordingly and by pressing “p” again, exit pivot mode. You can also enter in the pivot mode, from the corresponding option of the toolbar at OpenGL Viewport (as you see in figure 5).

![Figure 5: Enter in Pivot Mode](image)

In our case study that will complement the options explanation, we will try to cover with grass a surface with slopes and also “plant” some flowers on it. For that purpose, we will use a surface as a canvas (figure 6), a grass patch as an instance (figure 7) and a flower as instance too (figure 8).

![Figure 6: Sloped Surface for Canvas](image)  ![Figure 7: Grass Patch for Instance](image)  ![Figure 8: Flower for Instance](image)

**Note:** apart from instance and canvas selection, that user needs to define for using the instancing tool, the other four basic tool buttons for creating and erasing instances, are located at the bottom of the Instancing Tool window (see figure 2 and 9).
Tip: In case your canvas is an infinite plane, populate button is not functional (automatic population cannot be carried out due to the surface infinity), so it is advised to use the brush tool.

**Populate:** after selecting your desired settings, by clicking on the populate button, your instances will automatically be placed on the canvas according to your specifications.

**Brush:** by clicking on the brush, the cursor becomes a brush and you can manually click and drag it on your canvas (at the Viewport) and create instances on the selected areas.

**Erase:** by clicking on the erase button, the cursor becomes an erase tool, and by clicking and dragging it on the surface, you erase from these areas the generated instances.

**Clear:** by hitting the clear button, you can erase all the instances that have been created at once.

**Direction**

**Direction:** this option defines if instances will be directed by canvas surface normal (0%), by global z-axis (100%) or somewhere in between. In figure 11, by using a very simple example, we can see these options, for an inclined surface. Brown cubes have direction 0% and they are adapted to the surface while purple-blue cubes have direction 100% and they are parallel to z-axis, without taking under consideration the angle of the surface.

Tip: a normal to a surface at a point is the same as a normal to the tangent plane to that surface at that point.

In our case study example, when changing the direction percentage for generating grass on the selected surface, we have the following results.
For zero direction the instance (grass) is placed exactly by surface’s normal. It means, that the instances are placed in such way to follow the curvature of the surface.

For the half value in the direction percentage, the instances are oriented between the z-axis and the surface normal. Actually, for grass, a percentage around 10-20% would create a more realistic effect, as grass follows the surface curves, but it is also directed towards the sky.

Full percentage of the direction value, makes the instances to be oriented along the global z-axis. As we see, the grass instances, are totally horizontal (they just follow the height of the surface and not its normals) and in our case, with grass as an instance, the result is not very realistic. The final choice of the right value depends on the instance object that is used.

by changing this percentage, the instances will have random varying orientation related to the canvas normal. This time, by increasing the percentage, instances will use as fixed point their pivot point and they will rotate all around it, in all directions. Figure 15 shows the results of a simple example. Red squares have normal 0% and they are all at the same direction and tilt as the initial object, orange squares have 50% normal, so they are rotated somehow from their initial position, while blue squares that have 100% normal, are rotated completely differently.
In our case study, while changing the normal percentage, we have the following results.

<table>
<thead>
<tr>
<th>Normal</th>
<th>Description</th>
<th>Image</th>
</tr>
</thead>
<tbody>
<tr>
<td>0%</td>
<td>For normal 0% (all other values are also zero), instances are generated at the same direction as the original initial grass patch.</td>
<td><img src="image" alt="Figure 17: Normal 0%" /></td>
</tr>
<tr>
<td>50%</td>
<td>For half the percentage, instances are changing their directions and are placed randomly on the canvas (with their pivot point to stay pinned on the surface).</td>
<td><img src="image" alt="Figure 18: Normal 50%" /></td>
</tr>
<tr>
<td>100%</td>
<td>For full percentage of the normal value, instances are rotated even more in all possible directions. For grass, large normal perturbations create a rather unrealistic effect, while for other instances, this may be useful.</td>
<td><img src="image" alt="Figure 19: Normal 100%" /></td>
</tr>
</tbody>
</table>

**Roll**

**Roll**: this option enables the user to roll the instances; rotate them around the z (longitudinal) axis. For 0% roll, all instances will be the same as the initial one (see brick squares at figure 21) while for 100% roll, the objects are rolled around their z-axis (see wooden squares at figure 21).

![Figure 20: Roll around middle axis](image)

![Figure 21: Different Rolls](image)

At our case study, we use this time as instance, the flower model, which is not symmetrical around its z-axis, to see better the roll perturbation while using the instancing brush.
For zero roll, instances are generated exactly as the initial one.

By increasing the roll percentage, the flowers are rolling by their z-axis as they are generated on the canvas.

Scale: this option gives the possibility to the user to change the size of the generated instances. For scale 0%, the instances will be exactly the same as the initial one. By increasing this percentage though, the size of the new instances will be different. In figure 24, we see that the red squares that have scale 0% are all the same to the initial one, while the yellow squares, that have 50% scale are smaller or bigger than the initial one. Higher percentage, leads to bigger difference between the smallest and the largest created instance.

By using again as instance the flower model, we can see the results by changing the scale value.

For zero scale percentage, all flowers have the exact same size as the initial instance.

By increasing this percentage, instances start to have different sizes. The bigger the percentage, the bigger the difference between the smallest and the largest instance will be.
Scale: 100%
With full scale percentage, the instances are having even bigger deviation from their initial size, as we also see in figure 27.

<table>
<thead>
<tr>
<th>Placement</th>
<th>Minimum Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum Distance: 5 meters</td>
<td>In our example, by giving 5 meters as a minimum distance between new copies, the flowers will be placed in such way that there will be no other instance in radius of 5 meters around each one of them, as we can also see in figure 28.</td>
</tr>
<tr>
<td>Minimum Distance: 1 meter</td>
<td>If the minimum distance is smaller, instances come closer to each other as we can also see at figure 29.</td>
</tr>
<tr>
<td>Minimum Distance: 0.1 meters</td>
<td>The smaller the minimum distance is, the closer the instances come. For grass, very small minimum distance is desired, in order to assure better coverage, while for trees or flowers it depends on the result we need to create.</td>
</tr>
</tbody>
</table>
**Front Side**

**Front Side:** this option, which is enabled by default, means that instances will be generated at the front side of the canvas object. When disabled, instances will be placed at the back side.

In our example, we use these two options for the flower instance and we see the results below.

<table>
<thead>
<tr>
<th>Front Side: Enabled</th>
<th>Figure 31: Front Side</th>
</tr>
</thead>
<tbody>
<tr>
<td>For front side option enabled, all the instances (flowers) are generated on the top side of the surface, as we can see in figure 31.</td>
<td><img src="image" alt="Figure 31: Front Side" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Front Side: Disabled</th>
<th>Figure 32: Back Side</th>
</tr>
</thead>
<tbody>
<tr>
<td>When disabling this option, the instances (flowers) are now placed at the opposite side of the surface, as we see in figure 32.</td>
<td><img src="image" alt="Figure 32: Back Side" /></td>
</tr>
</tbody>
</table>

**Snap to Grid**

**Snap to Grid:** this option allows the user to place the instances along to the selected axes at specific distances.

For a better understanding of this option we will use a simple example of small cubes with green, red and blue colors, which represent the colors of the axes as well (x-axis: red color, y-axis: green color and z-axis: blue color).

<table>
<thead>
<tr>
<th>We are enabling the snap to grid option and we enter the following parameters, for each square that we use as an instance, thus, creating three packages:</th>
</tr>
</thead>
<tbody>
<tr>
<td>• For generating the green square, we use X step=0, Y step=2 and Z step=0.</td>
</tr>
<tr>
<td>• For the red square, we have: X step=2, Y step=0 and Z step=0.</td>
</tr>
<tr>
<td>• For the blue one, we set: X step=0, Y step=0 and Z step=2.</td>
</tr>
</tbody>
</table>

At figure 33, we have the three instance packages, and we see that the green cubes are snapped along the Y axis (with step of 2 meters), the red cubes at X axis of the canvas (with step of 2 meters) and blue cubes along its Z axis (with step of 2 meters). Their distance from the axis center is defined by the step we have set. If we
use a larger distance, for example 5 meters, we see at figure 34 the way that instances are placed. They follow the dimensions of the canvas but they snap to the grid of it with the desired step.

**Miscellaneous**

**Tool Radius**

**Tool Radius (pixels):** this option allows the user to change the size of the brush or erase tool. By increasing it, instances can be created in a larger area than the default one and correspondingly, instances will be erased in larger areas by one click on the instances at the Viewport.

In our example, we use the “Brush” to create some flowers and the “Erase” to delete some grass.

**Tool Radius for Brush**

By specifying the pixels of your tool, you can adjust the brush tool and generate instances to your desired areas by clicking and dragging your cursor. Instances will cover the area included in the red square (tool radius), like in figure 35.

**Tool Radius for Erase**

You can also erase specific areas of generated instances, by using the erase tool. By increasing the pixels of the tool radius, you are able to erase bigger surfaces. In the right figure, we are erasing the grass by clicking and dragging the cursor on the instances.

**Population**

**Population:** this is the amount of the desired instances the user wants to generate on the canvas. The population of instances can vary depending on the scene, the selected instance and the canvas size. For creating a grass or a carpet, large population promises wider coverage of the surface, while for trees or flowers for example, user may need just a small amount of them.

In our example, we will use the grass patch as an instance to fill our surface while using different populations of them. Tip: You need to use a small minimum distance as well, to allow your grass patches to come close enough and cover the surface.
If we use as desired population the value 50, as we see in figure 37, for this particular canvas and the desired effect, this amount is quite small.

By increasing the number of instances, the surface is covered by more grass patches and starts to look more realistic.

With the use of a very large population number, the whole surface is covered by grass, as it is seen in figure 39.

**Modified Density**

Modified Density: this option, allows the user to create even more complex scenes, since there is the possibility to enter a pattern (by adding a texture image), which will define the areas that instances will be placed. In general, for a given texture, instances are placed only over the white areas, while black ones stay empty. For gray areas, the application will probabilistically decide whether to place instances or not, based on the brightness level. It is better to use gray-scale images, to be able to specify and control better the areas you want or not to place instances.

Apart from choosing the desired texture, there is also the option to set the desired texture scale, by changing the numerical value next to the modified density option (see figure 40), for handling the gray parts of the image and increasing or decreasing the possibility to place an instance on them. At the next example, we use different values to experiment with this parameter. For easier observation we have applied the gradient texture on the canvas too.
Modified Density: 0
For a zero value, the whole area is considered to be black and the possibility to add an instance is zero as well, so no single instance is created, as we see too in figure 41.

![Figure 41: Zero Possibility](image)

Modified Density: 0.2
By increasing this value, some lighter areas are now used for generating the instances, as we see in figure 42.

![Figure 42: 0.2x Possibility Scale](image)

Modified Density: 2
As we are increasing more this value, the possibility for darker areas to start being used as active canvas is getting higher. For this value, as we see in figure 43, almost only the very dark area is not covered by grass.

![Figure 43: 2x Possibility Scale](image)

Modified Density: 100
This is the maximum value that we can set, and is making even the most dark areas (except the pure black ones) to receive instances, as we can also see in figure 44.

![Figure 44: 100x Possibility Scale](image)

By using our case study example again, we use another texture (it is black and white only, so there is no need to change the scalar value) with a specific pattern as you can see in figure 45, to create some grass areas and place flowers in between them. At the following images we see the necessary settings along with the final results.
After specifying the instance, the canvas, all the other perturbations etc., we can enter the population as well as the pattern, as seen in the image below. The result is shown in the figure 45, where grass is only generated on the specific white areas of the texture.

By inverting the colors of the previous texture (we could have used another texture as well) we can place instances to the empty areas, flowers for example. We see that the result is the creation of an exact pattern of combined grass and flowers, that fills the whole canvas.

By this way you can create unique designs, like carpets for example (see figure 49), by creating different instance packages each time and filling different areas of your canvas.

References