

INSTANCE PAINTER

MANUAL V1.0.1

OVERVIEW

InstancePainter allows you to easily paint objects onto other objects in your scene. The most common application for this would be creating dense forests with minimal effort. InstancePainter is made to be flexible and feature-rich for all needs.

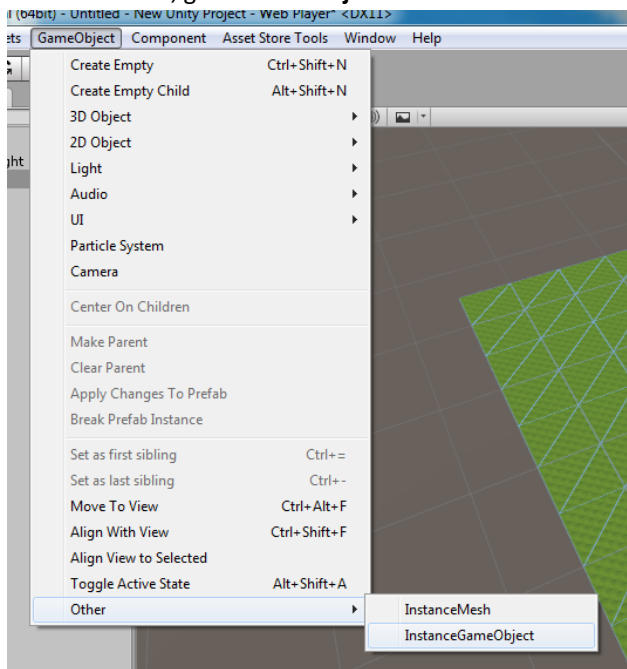
Some of the features included are:

- Paint simple meshes and any type of game objects onto other objects
- Paint multiple object types at the same time
- Paint on any surface--regardless of its angle
- Easily add subtle variation to each instance to avoid obvious repetition
- Select and modify individual instances
- Add color to objects
- Skew objects
- Simplify hierarchy
- Animate trees and grass with efficient wind animations
- Alternative rendering method with built-in culling, billboarding and LOD management
- Comes with fully textured tree and grass models

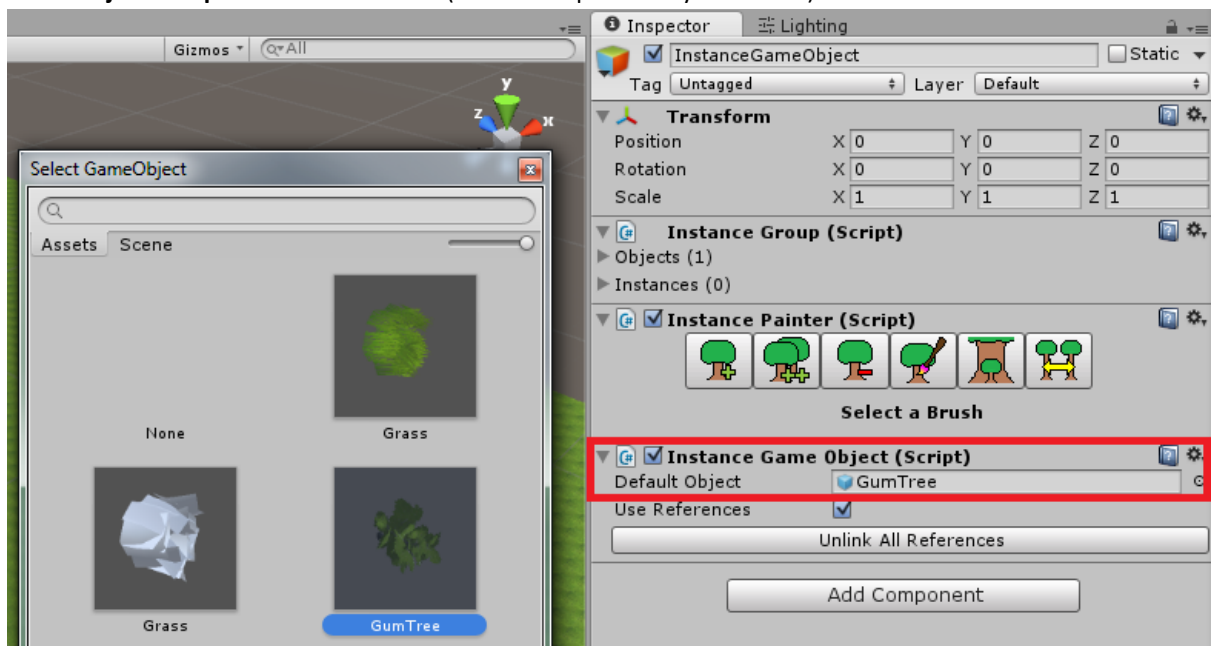
QUICKSTART GUIDE

In this quickstart, we will paint some tree prefabs onto a flat plane.

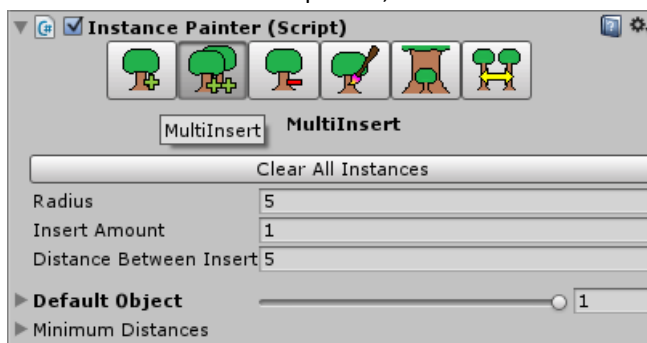
1. From the menu, go to **GameObject -> Other -> InstanceGameObject**.



- With the newly created **InstanceGameObject** selected, change the **Default Object** under the **Instance Game Object Component** to a **GumTree** (or another prefab of your choice).



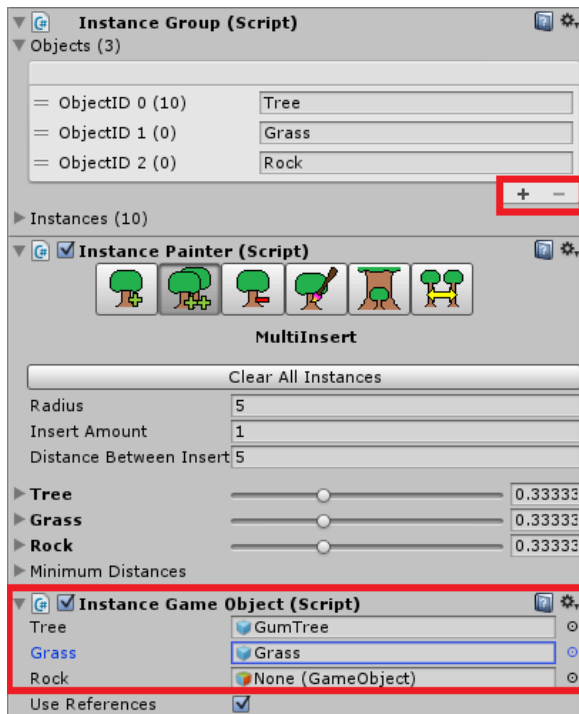
- In the **Instance Painter** component, select the **MultiInsert** tool (the icon with 2 trees with a ++).



- Now simply left-click and drag along the surface of an object in the scene, and the trees will paint on.



- You can continue adding more game objects via the **InstanceGroup** and **InstanceGameObject** components (as shown below).



GETTING STARTED

TERMINOLOGY

- Instance: A single object that has been placed in the scene. For example, this could be a single tree.
- Instance Object: A type of object that instances reference. For example, this could be the prefab of a tree.
- InstanceGroup: The component that stores a list all of the instances
- Instance Reference: The component that refers an instance GameObject to the InstanceGroup.

WHAT TO USE?

There are a few components required to start painting objects:

- InstanceGroup (stores a list of all of the instances)
- InstancePainter (performs the actual painting)
- InstanceMesh or InstanceGameObject (actually place the objects in the scene)

Alternatively, you can add ready-made game objects via the menu path: GameObject->Other.

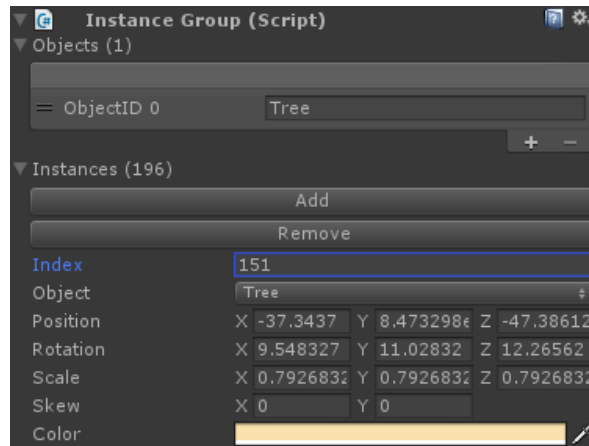
INSTANCEMESH OR INSTANCEGAMEOBJECT?

Which one you use depends on your requirements. Generally, if you want to add scripts to individual instances, or you want collision on the objects, you should choose InstanceGameObject. If you just want to render vegetation with basic animation, InstanceMesh should do the trick.

If you want a more detailed explanation, check out the Component Reference section.

COMPONENT REFERENCE

INSTANCEGROUP



The InstanceGroup is the core component of instance painting. It is the component that stores the individual components, including their transform. Although it is possible to create and modify each instance using this component, you would usually use the InstancePainter component.

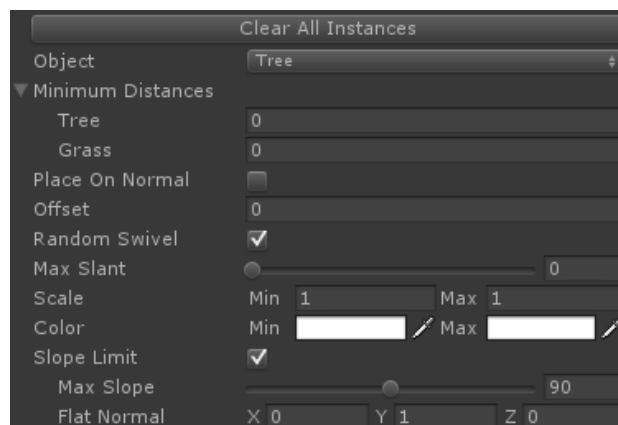
The most important thing that you must deal with in the InstanceGroup is the object names. Each instance refers to an object that you create here. For example, this may be a tree or grass.

INSTANCEPAINTER



InstancePainter is where you will actually paint instances into the scene. Below are details on each of the brushes.

INSERT



Use Insert to insert a single instance at a time. This is useful if you want precision in where you are placing the objects.

Minimum Distances allows you to specify the minimum distance between different object types.

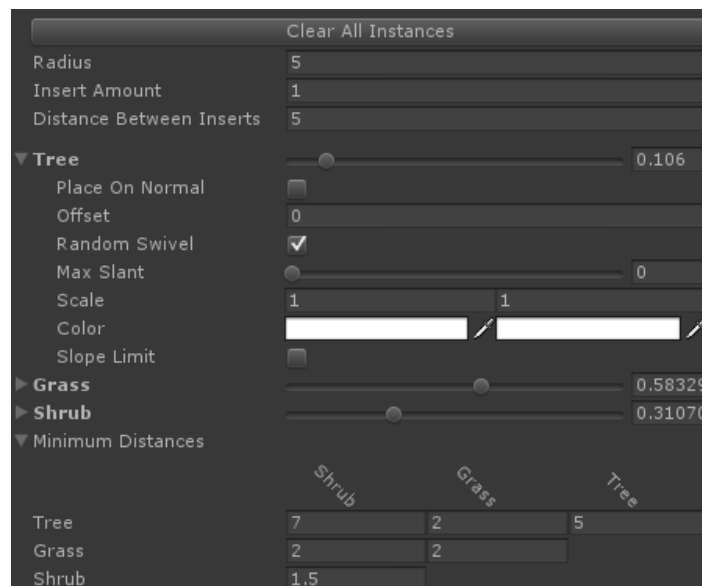
If Place On Normal is check, the instance will be oriented to be at a right angle to the surface it was placed on.

Offset is how far away from the surface it will be (vertically).

Max Slant will randomize the angle at which the instance will be (in degrees). Trees usually look good at 15 degrees.

Slope Limit will limit the surface incline that the instance can be placed. The Flat Normal specifies which direction is up (you usually don't want to change this). This option is more useful for MultiInsert.

MULTIINSERT



Use MultiInsert to quickly paint a large number of objects into the scene. Every object that you have added into InstanceGroup will have its own foldout. This allows you specify per-object information. For more information on these foldouts, look at the Insert brush section.

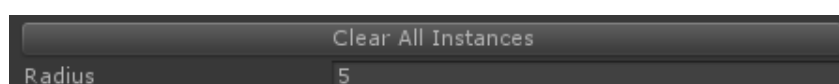
Next to each objects foldout is a slider that allows you to specify the amount of that object that will appear. For example, a value of 0 means it will never appear. 1 means it will always appear. 0.5 means it will appear 50% of the time.

The Insert Amount is how many objects will be inserted at a time.

Distance Between Inserts is how far you must move the cursor before more objects are inserted.

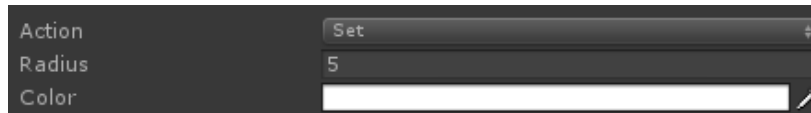
At the bottom is a matrix that allows you to specify the minimum distance between object types.

ERASE



Use Erase to remove instances from the scene. Clicking on the Clear All Instances button removes all of the instances from the InstanceGroup.

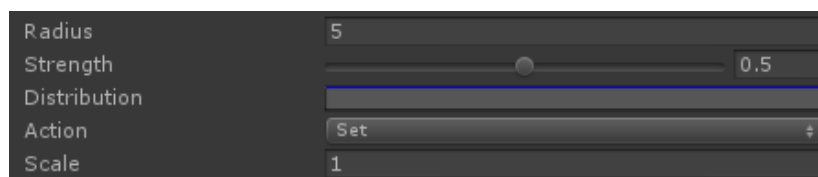
COLORIZE



Use Colorize to modify the color of existing instances. Action has four options:

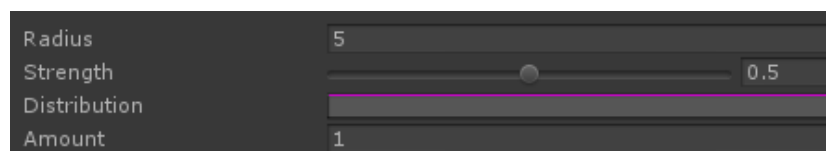
- Set: All painted instances will have the same color
- Set Range: All painted instances will have a color that is linearly interpolated between the two specified colors.
- Set Noise Range: Same as above but applied to individual color channels. This can create a more noisy effect.
- Add: Modify the current color. This can be good for making existing colors darker, lighter or adding a tint.

SCALE



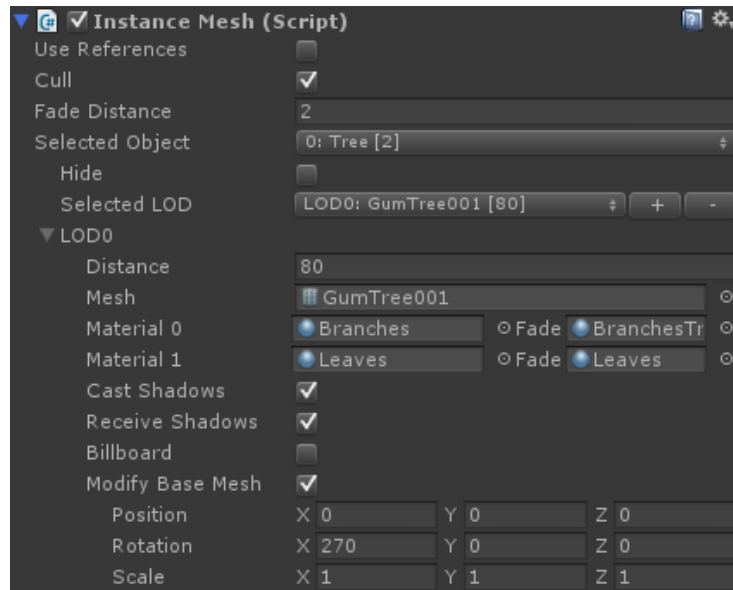
Use Scale to modify the size of the instances. This allows you to set, grow or shrink instances.

SCATTER



Use Scatter to make objects move away from each other. Use amount to specify how fast the objects will move.

INSTANCEMESH



InstanceMesh is a method of rendering each instance. Rather than separating each instance into a different GameObject, InstanceMesh will handle it all in the one GameObject (this may improve performance and workflow).

InstanceMesh also has Level of Detail (LOD) capabilities. This means that the further away an object is, the less detail you can put into the mesh. The Distance can be used to achieve this by setting the furthest distance from the camera that this LOD will appear (infinity means that it will always be visible).

Fading is enabled by adjusting the Fade Distance value. This will make LODs fade between each other. For this to work, you must specify an additional material that will be used when fading. The reasoning for this is that having a separate material for transparency will increase performance and reduce potential visual artefacts.

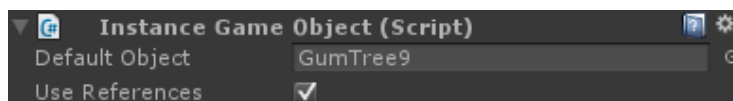
There is also a Billboard tick box which will make the LOD object always face the active camera.

Modify Base Mesh allows you to override the mesh transformation. This is useful if your mesh is not centered in the right place, or if it imported in the wrong direction, or if you want to quickly modify all instances. It is recommended that you turn this off if you don't need it as it will generate a new mesh (at initialization only).

To increase performance, ensure that Cull is ticked. This only renders instances that are visible to the camera. This may cause some shadowing artefacts, because it doesn't take the shadow volume into consideration.

| Pros | Cons |
|---|--|
| <ul style="list-style-type: none"> • Should be faster to render in most cases • Clean hierarchy • Allows for coloring and skewing of instance • Built-in Level of Detail (LOD) • Adjustable fading between LODs • Easy billboarding | <ul style="list-style-type: none"> • Cannot select instance by clicking on them • Cannot attach scripts to individual instances • Only works with static meshes |

INSTANCEGAMEOBJECT

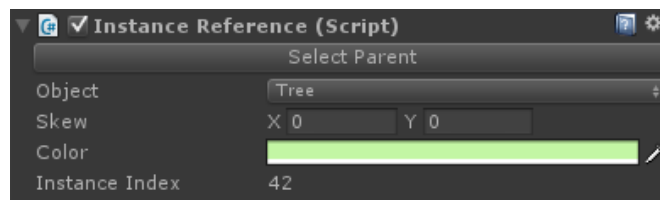


InstanceGameObject turns each instance into a Unity GameObject. Make sure the Use References is ticked so that the instances are visible.

It is best to use prefabs for the objects so that you can easily modify all of the instances at once.

| Pros | Cons |
|--|--|
| <ul style="list-style-type: none"> Easily select individual instance Different scripts for different scripts Make good use of prefabs | <ul style="list-style-type: none"> May be slower to render Can make the scene hierarchy messy No color or skewing |

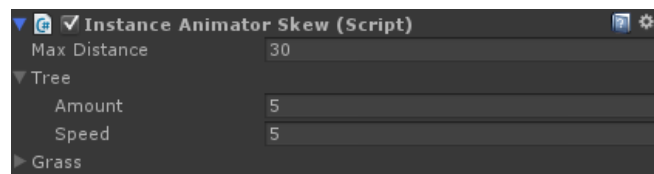
INSTANCEREFERENCE



Links the instance GameObject to the InstanceGroup. To see these, you must check that Use References are ticked in InstanceGameObject or InstanceMesh.

With this, you can modify properties of the selected instance. You can also modify the transform of the GameObject. The Select Parent button will take you to the GameObject that contains the InstanceGroup.

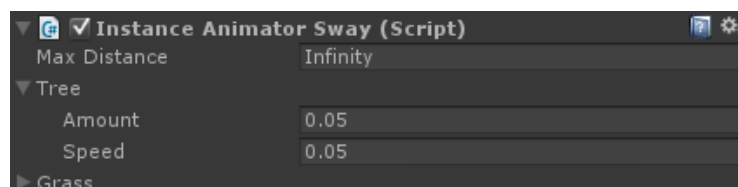
INSTANCEANIMATORSKEW



InstanceAnimatorSkew animates all instances by slightly skewing them. This can provide convincing animation for vegetation such as trees and grass. Skewing is mostly good for grass as the base will always be stationary (unlike swaying), whereas the top will move.

Modifying the Max Distance field will set the maximum distance that instances must be from the camera before they stop animating.

INSTANCEANIMATORSWAY



InstanceAnimatorSway animates all instances by slightly rotating them. This can provide convincing animation for vegetation such as trees and grass. Swaying is mostly good for trees as there will be no distortion of the mesh (unlike skewing).

Modifying the Max Distance field will set the maximum distance that instances must be from the camera before they stop animating.

FAQ

WHY DO MY TEXTURES FLASH WHEN USING SKEW?

Skewing can often mess up the normals of the mesh (which is one of the many reasons Unity doesn't support it out-of-the-box). It appears that this does not happen with Unity 5's new shaders. Alternatively, you can rewrite your shaders to renormalize the normals.

HOW CAN I IMPROVE THE PERFORMANCE OF INSTANCEMESH?

- Make use of LODs
- Use billboards
- Make sure Cull is ticked
- Have a maximum draw distance (ie don't use infinity)
- Try not to use animation where possible (animation causes recalculation of matrices which can be slow). Alternatively, you can specify a max distance from the camera that objects will be animated.
- Turn off lighting for distant LODs

DOES INSTANCEMESH SAVE DRAW CALLS?

No. Each instance has one draw call per sub mesh, which can be quite slow. Performance is made in other areas, such as less matrix rebuilding, LODs, efficient culling and less GameObjects for Unity to deal with. If Unity were to allow instancing from scripts, then I could easily modify the code to significantly reduce the draw calls (maybe we should nag UT for this feature?). But, as of Unity 5.0, this is not possible.

CONTACT

Email: stu_pidd_cow@hotmail.com