Mtree – Tree creation

About Mtree

Mtree is a tree creation tool that allows easy high quality tree creation inside unity editor.

Mtree consist of:

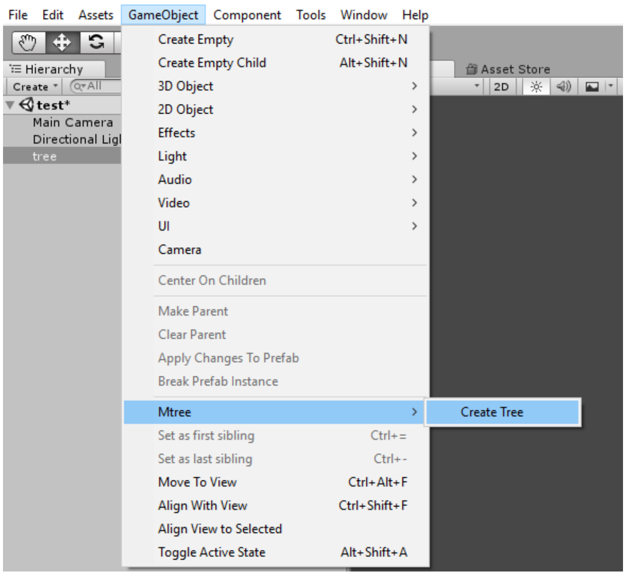
- User friendly step by step tree creation.
- Branches obstacle avoidance to make trees that adapt to the environment.
- Automatic material creation.
- Shader that supports wind vertex displacement, fast translucency, leaf color variation, and instancing.
- Automatic levels of details (LOD) creation.
- Automatic billboard creation.
- Automatic ambient occlusion bake into the vertex colors.
- Prefab creation.

Getting started

Create a tree

In order to create a tree, go to *GameObject/Mtree* and select *Create Tree*.

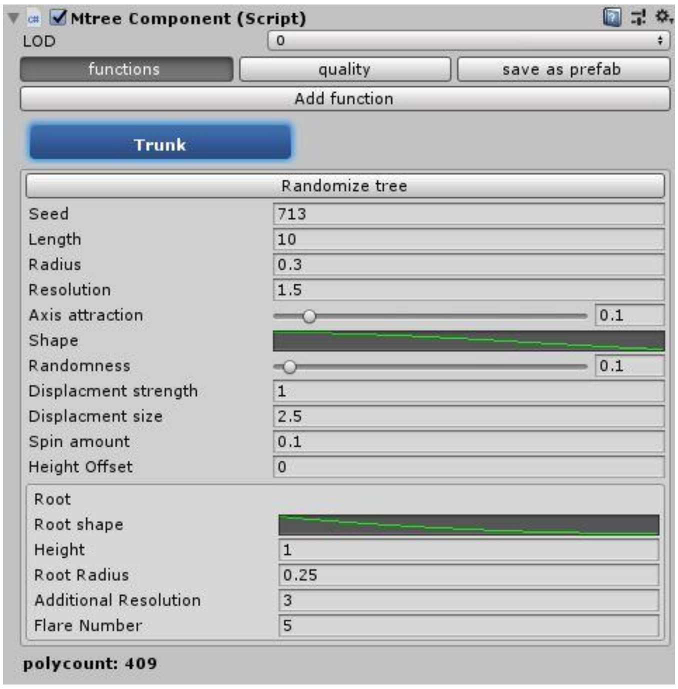
You can also add an empty object, and add the component *MtreeComponent.*



The Mtree Component

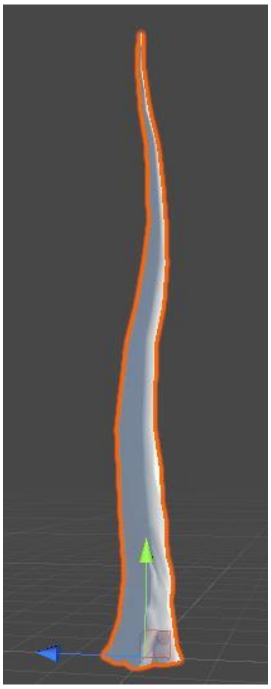
The Mtree component is what will guide the creation of the tree, and allow to save it as a prefab.

It can be found in the inspector when a tree is selected.



LOD	0
Seed	713
Length	10
Radius	0.3
Resolution	1.5
Axis attraction	0.1
Shape	
Randomness	0.1
Displacment strength	1
Displacment size	2.5
Spin amount	0.1
Height Offset	0
Root	
Root shape	
Height	1
Root Radius	0.25
Additional Resolution	3
Flare Number	5

polycount: 409



There are three tabs on the Mtree component:

- Functions – To create the tree level by level.
- Quality – To control the quality and polycount of the tree.
- Save as prefab – To export the tree as a prefab once it is finished.

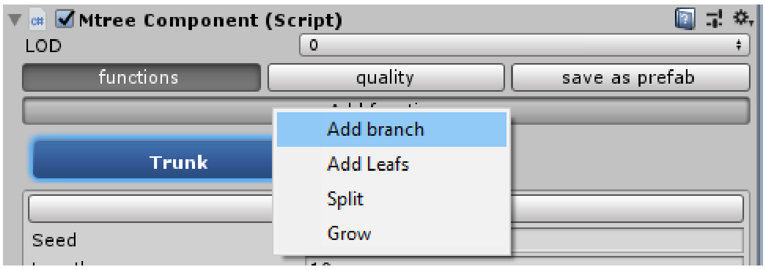
Functions tab

It is the most important tab, the one in which the tree is effectively created.

A tree follows an intuitive structure. First there is the trunk, then the branches, and finally the leafs. To each step is associated a function, and each function has parameters.

To select a function and change its parameters, click on it.

To add a function, click on *Add function* and select the function you wish to add. The function will be added on top of the previously selected function.



To remove a function, click on the minus inside the function you wish to remove.

Add Branch Function

This function adds branches the active level of the tree; is the trunk function is selected, the branches will be added to the trunk, and if a branch function is selected, the new branches will be added on the selected ones.

Add Leafs Function

This function adds leafs to the active level of the tree. If a tree has a leaf function, it will have two sub meshes, and two materials.

Split and Grow Functions

Those functions are used to have a greater control over the tree. They are not used often.

The Split function adds tiny splits to the selected levels. Those splits are very short branches that can be grown.
The Grow function make the extremities of the active level of the tree grow. It is mostly used after a Split function, but can be applied any other function except a Leafs function.

The Branch function is nothing more than a Split and Grow functions combined into one, with less parameters exposed.

The Functions parameters

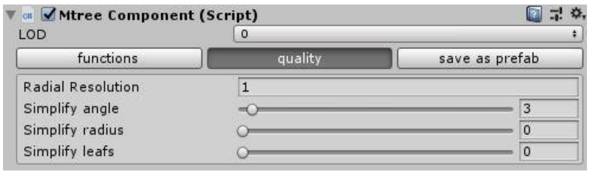
Each function has a set of parameters which control how the function affects the tree. Playing with the parameters is probably the best way to discover their uses.

Here are the most important parameters:

- **Seed** – Changing the seed will give another result of the function, like throwing dices two times will give two different numbers. To change the seed of all functions at once, you can click on *Randomize tree.*
- **Length** – controls the length of the branches/trunk generated.
- **Resolution** – Increasing the resolution will give a more complex looking tree, with a higher number of triangles. Be warry as the poly-count rises quickly.
- **Randomness** – A higher value will give the tree a more jagged appearance.

Quality tab

It is a seldom visited tab, it allows control over the quality of the tree. The settings of the quality tabs vary with the level of detail (LOD) of the tree. To visualize each LOD level, change the *LOD* settings.



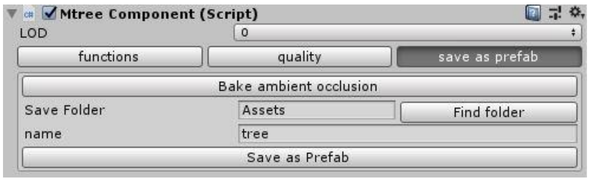
LOD 0
Radial Resolution 1
Simplify angle 3
Simplify radius 0
Simplify leafs 0

- Radial resolution – Controls the amount of vertices that go around each branch. Lower is more performant.
- Simplify angle – Controls how much straight branches are simplified. Higher is more performant.
- Simplify radius – Threshold of the thinnest branch drawn. Higher is more performant.
- Simplify leafs – Controls the density of leafs after their creation. Higher is more performant.

In most uses, the default values of each LOD level are good enough.

Save as prefab tab

The trees are created with the Mtree component. When you are satisfied with the tree and consider it finished, **you must save it as a prefab using this tab**, otherwise the tree will not be optimized.



LOD 0
Save Folder Assets
name tree

Saving the tree as prefab will generate all 4 LOD levels with ambient occlusion, as well as render a billboard. The meshes are saved in the specified folder, and a prefab with a LOD group is created.

You can visualize the result of the ambient occlusion pass by clicking on *Bake ambient occlusion*. Note that the ambient occlusion information will be lost if you change the tree afterwards.

You can Select the folder to which save the prefab by clicking on *Find folder.*

Save the tree as a prefab by clicking on *Save as Prefab* and wait until the process is over. You can find the prefab in the specified location. Once the tree is saved, only use the prefab, you can delete the tree with MtreeComponent attached.

Exhaustive list of function parameters.

Trunk

Parameter	*Description*
Seed	Changing the seed will give another result of the function, like throwing dices two times will give two different numbers.
Length	The length of the generated trunk.
Radius	The radius of the trunk.
Resolution	The amount of points per unit of length.
Axis attraction	How much the tree is drawn to its original axis. Prevents the trunk from diverging too much.
Shape	The variation of the trunk radius with its length.
Randomness	How irregular the trunk looks.
Displacement strength	How much noise affects the geometry of the trunk.
Displacement size	How large is the noise affecting the geometry of the trunk.
Spin amount	How much the trunk is twisted.
Height offset	How much the trunk goes inside the ground. Helps when a tree is placed on an uneven ground.
Root Shape	The evolution of the radius of the trunk near the ground.
Height	The height to which the root goes.
Root Radius	The radius of the trunk on the ground.
Additional Resolution	How much more resolution to add to the trunk near the ground.
Flare number	Number of flares near the ground.

Add Branch

Parameter	*Description*
Seed	Changing the seed will give another result of the function, like throwing dices two times will give two different numbers.
Number	Approximate number of how many branches to add.
Length	The length of the branches.
Resolution	The amount of points per unit of length. **Warning:** polygon count is very sensible to this parameter.
Randomness	How irregular the branches look.
Radius	The start radius of the branches, relative to the radius of what they grow on.
Split proba	The probability for a branch to fork into multiple branches. **Warning:** polygon count is very sensible to this parameter.
Max splits number	The max number of branches a branch can fork into.
Angle	The angle of the branches from what they grow on.
Up attraction	How much the branches tend to grow toward the sky.
Gravity strength	How much gravity affects the branches.
Start	The minimum height from which the branches are created.

Add Leaf

Parameter	*Description*
Mesh type	The geometry of the leaf.
Number	Approximately how many leafs to add.
Size	The size of each leaf.
Max branch radius	The maximum radius of the branches from which the leafs are created.
leaf weight	How much the leafs are drawn to the ground.
Override normals	If true, the leafs normal will be determined by their position on the tree. If false the default leaf mesh normal are used.

Split

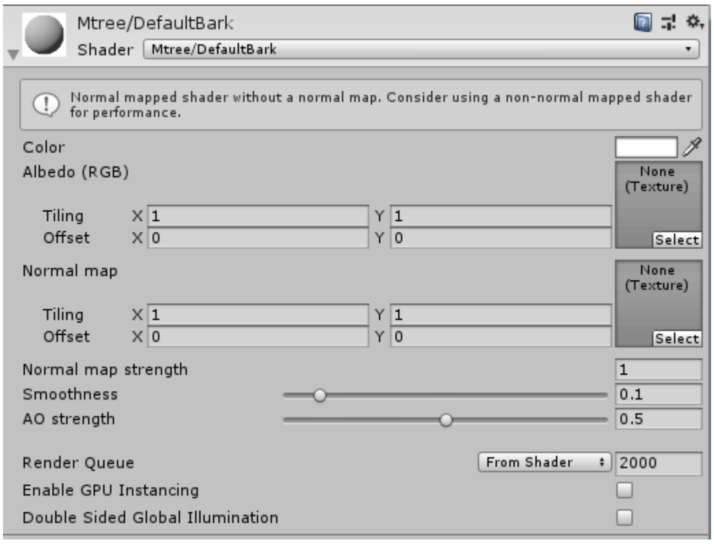
Parameter	*Description*
Seed	Changing the seed will give another result of the function, like throwing dices two times will give two different numbers.
Number	Approximately how many splits to add.
Split angle	The angle of the splits from what they grow on.
Max splits at a time	The max number of splits that can occur on the same location.
Split radius	The radius of the split relative to the radius of its parent.
Start	The height
Height spread	The minimum height from which the splits are created.

Grow

Parameter	*Description*
Seed	Changing the seed will give another result of the function, like throwing dices two times will give two different numbers.
Length	The additional length to grow the parent.
Resolution	The amount of points per unit of length. **Warning:** polygon count is very sensible to this parameter.
Split proba	The probability to fork into multiple branches. **Warning:** polygon count is very sensible to this parameter.
Split angle	When forking, the angle between the newly created branches.
Shape	The evolution of the radius with the length.
Split radius	When forking, the radius of the newly created branches.
Max splits at a time	The max number of splits that can occur on the same location.
Randomness	How irregular the branches look.
Up attraction	How much the branches tend to grow toward the sky.
Gravity strength	How much gravity affects the branches.

Bark shader

By default, when adding a tree, a material is created using the Mtree bark shader.



Mtree/DefaultBark
Shader Mtree/DefaultBark

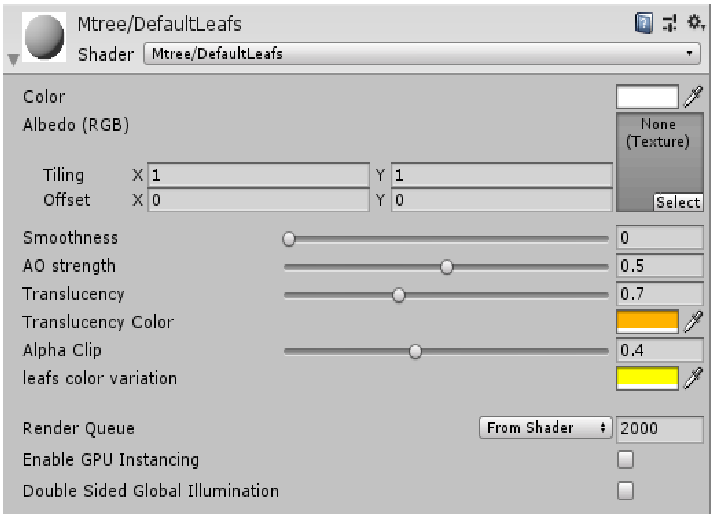
Normal mapped shader without a normal map. Consider using a non-normal mapped shader for performance.

Color
Albedo (RGB) None (Texture) Select
Tiling X 1 Y 1
Offset X 0 Y 0
Normal map None (Texture) Select
Tiling X 1 Y 1
Offset X 0 Y 0
Normal map strength 1
Smoothness 0.1
AO strength 0.5
Render Queue From Shader 2000
Enable GPU Instancing □
Double Sided Global Illumination □

- **Color**: The color tint of the material.
- **Albedo**: The color texture.
- **Normal map**: The normal texture.
- **Normal map strength**: The intensity of the normal map.
- **Smoothness**: How reflective the material is.
- **AO strength**: The intensity of the baked ambient occlusion.

Leafs shader

When adding leafs to a tree, a leafs material is created using the Mtree leafs shader.



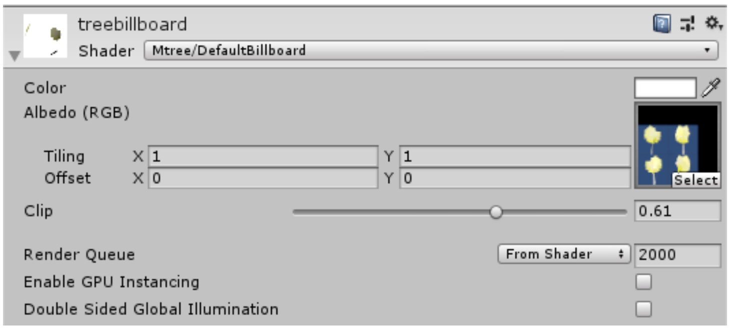
Mtree/DefaultLeafs
Shader Mtree/DefaultLeafs

Color
Albedo (RGB) None (Texture) Select
Tiling X 1 Y 1
Offset X 0 Y 0
Smoothness 0
AO strength 0.5
Translucency 0.7
Translucency Color
Alpha Clip 0.4
leafs color variation
Render Queue From Shader 2000
Enable GPU Instancing □
Double Sided Global Illumination □

- **Color**: The color tint of the material.
- **Albedo**: The color texture.
- **Smoothness**: How reflective the material is.
- **AO strength**: The intensity of the baked ambient occlusion.
- **Translucency**: How much light can go through the leafs.
- **Translucency Color**: The color tint of the backlit leafs.
- **Alpha clip**: Threshold at which a pixel is considered transparent.
- **leafs color variation**: Tint applied randomly to leafs.

Billboard shader

When saving a tree as prefab, a billboard of the tree is created, and has a material using the Mtree billboard shader.



treebillboard
Shader Mtree/DefaultBillboard

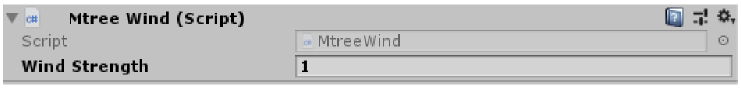
Color
Albedo (RGB)
Tiling X 1 Y 1
Offset X 0 Y 0
Clip 0.61
Render Queue From Shader 2000
Enable GPU Instancing □
Double Sided Global Illumination □

- **Color**: The color tint of the material.
- **Albedo**: The color texture. This texture is automatically assigned when creating the prefab.
- **Clip**: Threshold at which a pixel is considered transparent.

Wind vertex displacement

Both the bark and leafs shaders use vertex displacement to simulate wind. The wind intensity can only be controlled when the tree is saved as prefab.

A saved tree prefab has a *Mtree Wind* component.



Mtree Wind (Script)
Script MtreeWind
Wind Strength 1

To change the wind intensity, change the *Wind Strength* parameter.