Fractal Trees with Springs

Gandhi Games

Table of Contents

How it Works

This section contains a brief summary of how the tree types are generated. For more information, see the 'Fractal Tree Information' documentation.

Branching Tree

The branching tree has a simple generation process, for each generation:

1. Create a branch
2. Split into two braches based on angle

L-System

The L-System starts with an Axiom (seed), for example: 'F'; and a set of production rules, for example 'F=F[+F-]X' and 'X=-FF'. Using these parts, a sentence is generated over a number of generations by applying the rules to the axiom and each successive sentence.

For example:

Gen 1 = F (initial Axiom)
Gen 2 = F[+F-]X (F changed into 'F[+F-]X')
Gen 3 = F[+F-]X[+F[+F-]X-]-FF (any subsequent F's changed into 'F[+F-]X' and X into -FF)

This process will continue for a number of generations until a final sentence is produced. This sentence is then used to generate a tree by looping through each character of the sentence and applying the following rules:

Character	Rule
C0, C1… Cn	Change all subsequent drawn lines to colour n.
F	Move forward in current direction and draw line.
G	Move forward in current direction without drawing a line.
-	Rotate direction left by n degrees.
+	Rotate direction right by n degrees.
[	Store current state.
]	Restore saved state.
!	Reverses the meaning of left and right.
|	Turn 180 degrees.

Space Colonization

Place a number of 'leaves in the scene'. These leaves should all be the child of a single GameObject and each leaf should have the ColonizationLeaf script attached. You can have the leaves generated for you by attaching ColonizationLeafGenerator to a GameObject in the scene. This will generate a number of leaves within a circle radius.

With the leaves placed the tree can grow. It does this by looping through each leaf:

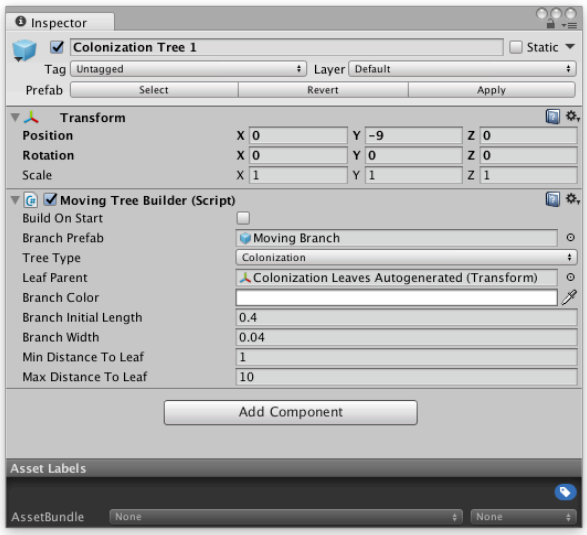
1. If any leaves are within a minimum and maximum distance, then the next branch direction is rotated towards that leaf (if a number of leaves are found then an aggregate direction is calculated).
2. Any leaves that are below the minimum distance are removed from the simulation.

These steps are performed until no suitable leaves are found.

How to Generate Trees

1. Add either the StationaryTreeBuilder or MovingTreeBuilder script to a GameObject (depending on whether you will be applying a force to the tree or not). Moving trees are computationally expensive and should only be used when necessary.
2. Add either the 'Stationary Branch' prefab or the 'Moving Branch' prefab (StationaryTreeBuilder must have the Stationary Branch prefab and the MovingBranchBuilder must have the Moving Branch prefab).
3. Choose your tree type.
4. Adjust the tree settings.

An example Space Colonization moving tree is shown below.



If you are creating a Space Colonization tree, then you will also need to setup the 'Leaf Parent'. You can create leaves either manually or by adding a ColonizationLeafGenerator script to a GameObject (check the Tree Demo for an example setup).

How to Apply Forces

When you have created a moving tree you can apply forces by calling the methods in MovingTreeBuilder:

- ApplyDirectedForce: applies a uniform force in the direction specified.
- ApplyPushForce: applies an explosive force that radiates from the position specified. Force is applied relative to the distance from the point.
- ApplyPullForce: applies an implosive force that radiates from the position specified. Force is applied relative to the distance from the point.

Examples:

```
treeBuilder.ApplyDirectedForce(direction, position, radius);
treeBuilder.ApplyPushForce(force, position, radius);
treeBuilder.ApplyPullForce(force, position, radius);
```

See DemoForceController script for an example of applying forces.

Inspector Variables

Select tree type in inspector to get tree specific options. The variables are outlined below.

Branching Tree

Branch Colour	The colour of the tree. The branching tree can only be one colour.
Generations	The number of iterations to grow. Larger numbers result in bigger trees.
Initial Length	The length of the initial tree trunk.
Length Multiplier On New Generation	The length of the branch is multiplied by the number on each successive generation. Enter a number less than 1 to reduce the size each generation.
Angle	The angle used when splitting a branch.
Branch Width	The width of all branches on the tree.

L-Tree

Auto Width	Adjust the width based on the colour index.
Mass Based on Width (moving trees only)	Adjust the mass based on the width of a branch.
Branch Initial Width	The initial width of the tree. This will be the width of all branches if auto width is false.
Generations	The number of iterations to process. Larger numbers result in larger sentences. The processing time grows exponentially.
Axiom	The initial seed.
L-Tree Rules	The rules to apply to the axiom and each successive sentence.
Branch Length	The length of the branches.
Angle	The angle applied on branching.
L-Tree Colour	The colours. Only the first colour will be used if the rule set does not include colour indices.

Space Colonization

Branch Colour	The colour of the branches.
Branch Initial Length	The length of the branches until a leaf is found.
Branch Width	The width of the branches.
Min Distance to Leaf	The minimum distance to nearby leaves. When a branch gets within this distance to a branch it is removed.
Max Distance to Leaf	Leaves further than this distance are ignored by the tree,