

IniTree 0.2.3
by inimArt.com

Quick help guide

iniTree 0.2.3

iniTree is a Quartz Composer custom plugin by Luca Palmili (inimArt), that let you draw an N-ary tree with some particular effects, ideals for VJ and realtime performances.

more info:

inimart.com
[linkedIn](#)

inimart

Tree creation

The algorithm uses recursion. Here is what it does in pseudocode (for n-ary tree):

```
main{
  //...
  max_levels = 10 //the max num of levels
  tree(initLength, 1);
  //...
}

tree(length, level){
  //...
  if(level<max_levels){ //is the end of the recursion?
    n = 3 //it is a ternary tree
    for(x=0; x<n; &lt;++){
      pushMatrix()
      rotate ((360 / n)*x) along y axis
      rotate (aperture) along z axis
      drawLine(0,0,0, length,0) //from 0,0,0 to 0,length,0
      translate(0,length,0)
      tree(length/2, level+1) //recursive call
      popMatrix()
    }
  }
}
```

Tree customization

Vertices are calculated every time you change the number of branches, of levels, or the tree opening. It is possible to customize the aspect of the tree in several ways*:

- **Size matters. Colors too:**

The lines / points / 3D elements of the tree (leaves or 3D branches) width, as well as their colors, can be constant or shrink their size according to their depth (witch gives you a better perception of the tree in the 3D world).

- **A lot of elements to play with:**

You can draw the tree using a variety of elements: *lines*, for a wireframe style; *points*, for enhance the connections between branches; *polygons*, to perk up the tree with elegance objects; *3D branches*, to have a more robust look and feel; *3D leaves*, to add realism. All these elements are personalizable in terms of size and colors.

- **Grow up or shrink down, (almost) limitless:**

You can change in real time the number of branches and the depth of the tree. You will see your tree grow up, shrink down, or expand up to a huge number

of vertices and polygons. A smooth animation allows the tree changes to be more realistic and nice to see.

Changing the opening value of the tree will result in a very elegant opening-flower-like animation.

- **Proportion matters, too:**

The proportion between different branch levels length can be modified, allowing you to give more importance to the begin or the end of the tree.

- **Completely symmetrical, or incomplete and chaotic:**

Decide if you want a complete n-ary tree or if some branches will not continue to grow. Branches can be uniformly distant between each others, or can be randomly arranged.

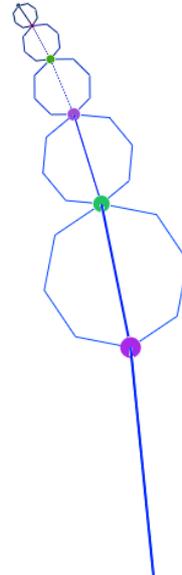
- **Texturize your world:** Now you can apply a texture on tree branches, as well as on leaves.

- **To be vain serves a mirror:**

You can also choose to draw a mirror-copy of the tree, rotated by 180 degrees. So you can double the fun.

All these possibilities of customization resulting in a huge variety of different tree-styles. You will get a hi-tech style, a style inspired by nature, pure fantasy, with strong contrast, elegant, symmetrical, minimal, rich in detail.

Let's see what you can do!



*Some features are Pro version-only

plugin inputs

height

The height of the tree. Be careful with this value. Remember that for a n -ary tree with height h , if you don't check **rndCut** option, the number of leaves will be n^h , and the total number of nodes in a perfect n -ary tree is $(n^{(h+1)} - 1)/(n - 1)$.

opening

The angle between parent and child branches (higher values will result in a more open tree).

branchesNum

The number of children for each branch. Be careful with this value if **rndCut** option is not checked (see Height input description).

branchesRatio

Each level of the tree will vary the length of the branches compared to the previous level:

- -1: branches near the root of the tree will be shorter
- 0: all branches will have the same length
- 1: branches near the root of the tree will be longer

rndRot

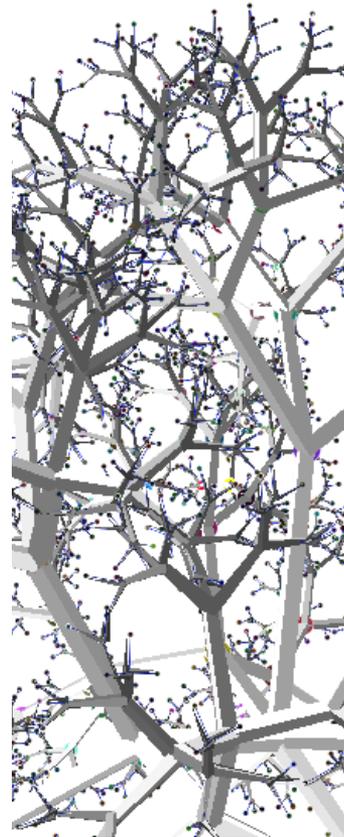
If enabled, the angle between branches on the same level will not be equally distant, resulting in a more realistic tree. This parameter doesn't take into account the space distribution of all the branches, so if you use rndRot, it will be more likely to have branches overlap.

leafPolyStart

Where the drawing of the leaves/polygons between branches should start

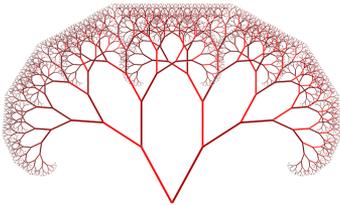
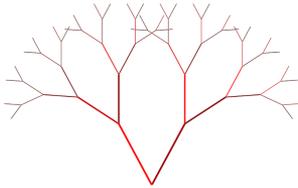
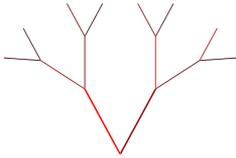
- 0 means that leaves should start from the tree root.
- 1 means that leaves should stay only at the end of the tree.

You can enable/disable the drawing of leaves/polygons using drawLeaf/drawPoly input.



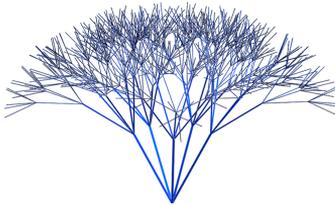
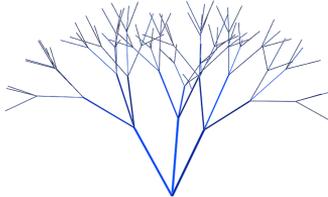
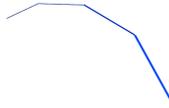
height

3 / 5 / 10



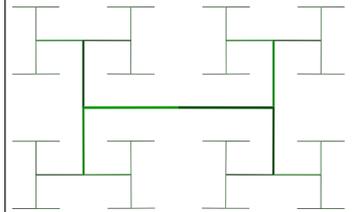
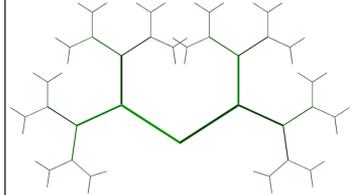
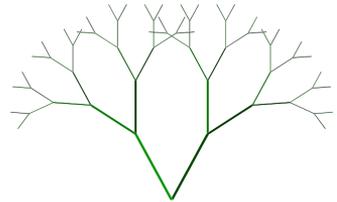
branchesNum

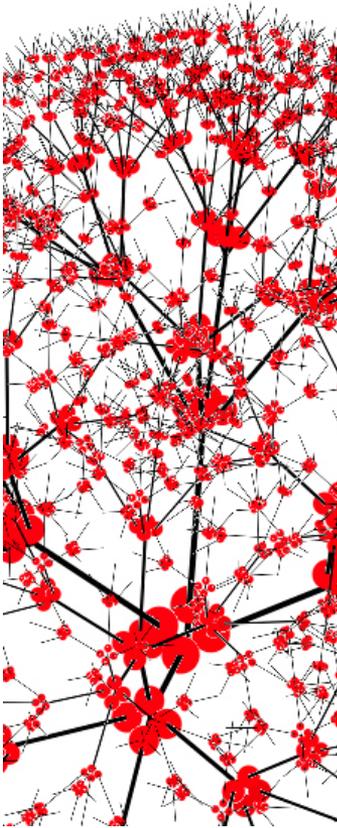
1 / 3 / 5



opening

0.5 / 1.0 / 1.5





pointPos

Points position within the branch. Each branch b has got one point p , you can choose to draw it at the beginning of the branch, at the end, or before or after the branch starts/ends.

- >1 : the point p will be after b ends, following b direction
- 1 : the point p will be at the end of the branch b
- 0 : the point p will be at the start of the branch b
- <1 : the point p will be before b starts, following b direction

growDelay

Each time you increase or decrease the height of the tree, the new branches will grow or shrink following a smooth animation, with a duration proportional to this value.

drawLines

Enable/disable the drawing of branches using lines.

drawPoints

Enable/disable the drawing of points between branches.

drawLeaves

Enable/disable the drawing of leaves between branches, starting from **leaf-PolyStart** position on the tree.

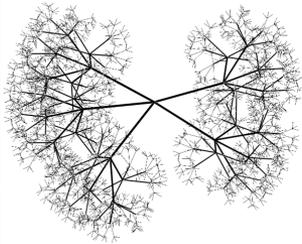
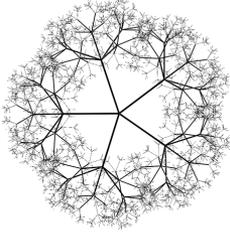
draw3D

Enable/disable the drawing of extruded branches with **3DWidth** thickness.

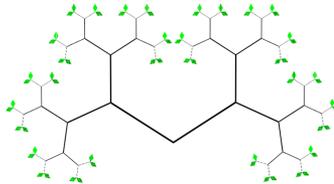
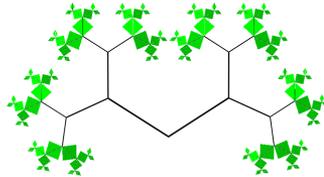
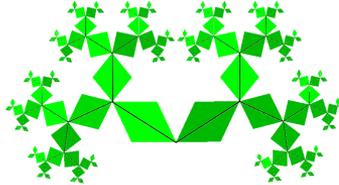
drawMirror

If checked, the tree is drawn two times: the second one rotated by 180 degree along X axis.

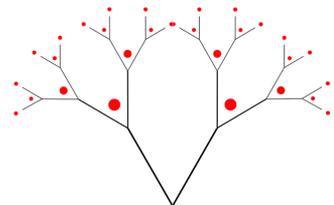
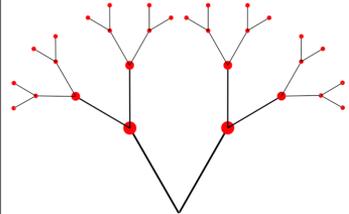
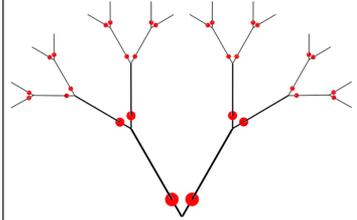
rndRot
FALSE / TRUE / TRUE



leafPolyStart
0.0 / 0.5 / 0.9



pointPos
0.2 / 1.0 / 1.3



drawPoly / polyDiv

Enable/disable the drawing of a polygons outline between the ends of a branch. The polygon will have **polyDiv** segments (3 for triangle, 10+ for circle, etc.) and **polyWidth** will be its stroke value.

lineWidth / leafWidth / pointSize / 3DWidth / polyWidth

Sets the width and the size of the tree elements.

autoSize

If checked, the width of the elements will become smaller depending on their height in the tree.

rndCut / rndCutVal

If checked, the tree will not be a complete tree: each branch will continue to grow only with a certain probability **rndCutVal**.

For example, if **rndCutVal** is:

- 0 : only the first level branches of the tree will be drawn
- n, with $n > 0$ and $n < 1$: the branch will be drawn with a probability of n
- ≥ 1 : all branches will be drawn (same effect of **rndCut** turned off)

line/point/leaf/3D/Poly-Col

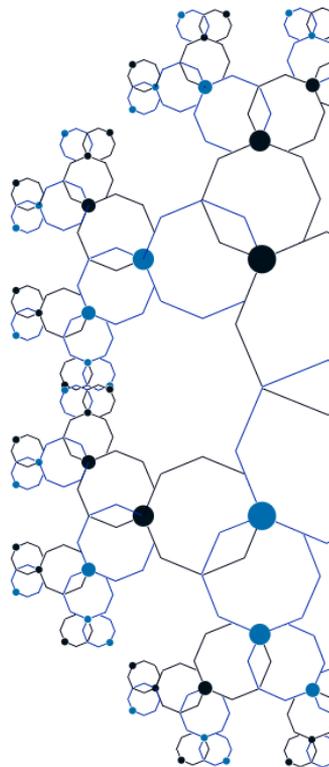
Sets the color of the tree elements.

dec-line/point/leaf/3D/Poly-Col

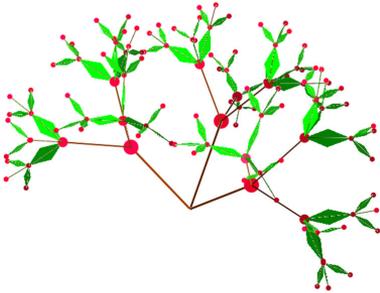
If checked, the color of the elements will become darker, depending on their height in the tree.

rndPointCol

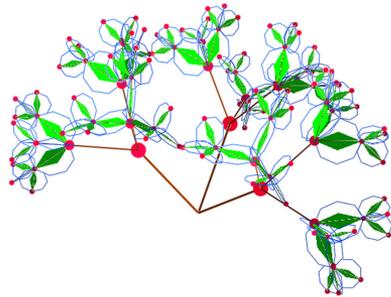
If checked, the points will have random colors (**pointCol** input will no longer be valid).



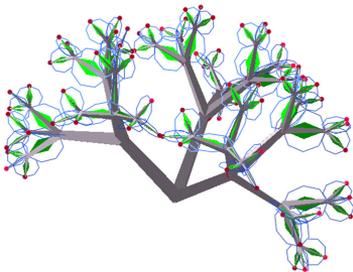
drawLines + drawPoints + drawLeaves



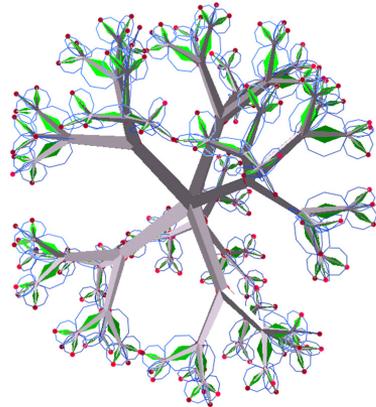
+ drawPoly

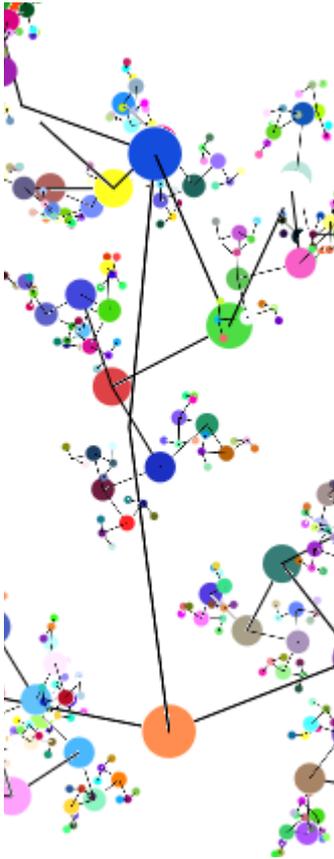


+ draw3D



+ drawMirror



**texture3DBody**

If present, the the provided image will be used as texture for each branch (only if **draw3D** is checked).

textureLeaves

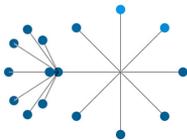
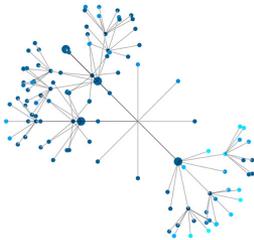
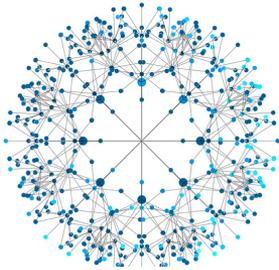
If present, the the provided image will be used as texture for each leaf (only if **drawLeaf** is checked).

blendingMode

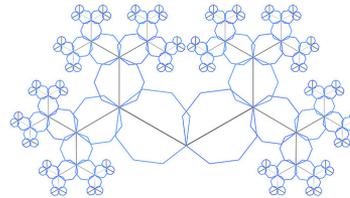
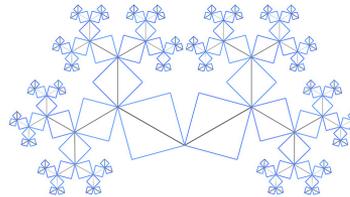
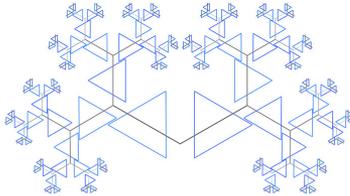
Different kind of OpenGL blending function, used for draw all the OpenGL primitives in iniTree:

- 0 : `glBlendFunc (GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA);`
- 1 : `glBlendFunc(GL_ONE, GL_ONE_MINUS_SRC_COLOR);`
- 2 : `glBlendFunc(GL_DST_COLOR, GL_ONE_MINUS_SRC_ALPHA);`

rndCut
FALSE / TRUE / TRUE
rndCutVal
1.0 / 0.5 / 0.2



polyDiv
3 / 4 / 8



texture3DBody + textureLeaf

