FractalTree Documentation

Release v0.2.4

Pixelwar

Jul 26, 2020

Table of Contents

CHAPTER 1

Installation

```
pip3 install Tree
```

1.1 Development Version

```
git clone https://github.com/PixelwarStudio/PyTree.git
cd PyTree
pip3 install .
```

CHAPTER 2

Examples

CHAPTER 3

Commandline

3.1 Usage

```
tree-cli [OPTIONS]
```

Options:

-l, --length	The start length of tree.
-b, --branches	Add a branch with a scale and a angle.
-s, --sigma	Add randomness to scale and angle.
-a, --age	Indicates how many time the tree should be iterated.
-p, --path	The path for saving the tree.

-c1, –color1 The starting color given as r g b. -c2, –color2 The end color given as r g b. -t, –thickness The start width of the first branch. –help Show this message and exit. –show Shows a image of the tree.

3.2 Examples

CHAPTER 4

Reference

4.1 core

class Tree.core.**Tree**(*pos=(0, 0, 0, -100)*, *branches=None*, *sigma=(0, 0)*)

The standard tree.

__init__(*pos=(0, 0, 0, -100)*, *branches=None*, *sigma=(0, 0)*)

The contructor.

Parameters

- **pos** (*tupel*) – A tupel, holding the start and end point of the tree. (x1, y1, x2, y2)
- **branches** (*tupel/array*) – Holding array/s with scale and angle for every branch.
- **sigma** (*tuple*) – Holding the branch and angle sigma. e.g.(0.1, 0.2)

draw_on(*canvas*, *stem_color*, *leaf_color*, *thickness*, *ages=None*)

Draw the tree on a canvas.

Parameters

- **canvas** (*object*) – The canvas, you want to draw the tree on. Supported canvases: svgwrite.Drawing and PIL.Image (You can also add your custom libraries.)
- **stem_color** (*tupel*) – Color or gradient for the stem of the tree.
- **leaf_color** (*tupel*) – Color for the leaf (= the color for last iteration).
- **thickness** (*int*) – The start thickness of the tree.

get_branch_length(*age=None*, *pos=0*)

Get the length of a branch.

This method calculates the length of a branch in specific age. The used formula: length * scale^age.

Parameters **age** (*int*) – The age, for which you want to know the branch length.

Returns The length of the branch

Return type float

get_branches()
Get the tree branches as list.

Returns

A 2d-list holding the grown branches coordinates as tupel for every age. Example: [[(10, 40, 90, 30)], [(90, 30, 100, 40), (90, 30, 300, 60)], [(100, 40, 120, 70), (100, 40, 150, 90), ...], ...]

Return type list

get_node_age_sum(*age=None*)
Get the sum of branches grown in an specific age.

Returns The sum of all nodes grown in an age.

Return type int

get_node_sum(*age=None*)
Get sum of all branches in the tree.

Returns The sum of all nodes grown until the age.

Return type int

get_nodes()
Get the tree nodes as list.

Returns

A 2d-list holding the grown nodes coordinates as tupel for every age. Example: [[(10, 40)], [(20, 80), (100, 30)], [(100, 90), (120, 40), ...], ...]

Return type list

get_rectangle()
Gets the coordinates of the rectangle, in which the tree can be put.

Returns (x1, y1, x2, y2)

Return type tupel

get_size()
Get the size of the tree.

Returns (width, height)

Return type tupel

get_steps_branch_len(*length*)
Get, how much steps will needed for a given branch length.

Returns The age the tree must achieve to reach the given branch length.

Return type float

grow(*times=1*)
Let the tree grow.

Parameters **times** (*integer*) – Indicate how many times the tree will grow.

move(*delta*)
Move the tree.

Parameters **delta** (*tupel*) – The adjustment of the position.

move_in_rectangle()
Move the tree so that the tree fits in the rectangle.

4.2 draw

class Tree.draw.**PilDrawer**(*tree*, *canvas*, *stem_color=(255, 255, 255)*, *leaf_color=(230, 120, 34)*, *thickness=1*, *ages=None*)
A drawer class for drawing on PIL/Pillow images.

class Tree.draw.**SvgDrawer**(*tree*, *canvas*, *color=(255, 255, 255)*, *thickness=1*)
A drawer class for drawing on svg documents.

group
Saves the groups created for every age.

Type list

draw()
Draws the tree.

Parameters **ages** (*array*) – Contains the ages you want to draw.

Index

Symbols

D

G

M

P

S

T