

Построение классического дерева Пифагора

Черкашин Александр Михайлович

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

Целью исследования являются технология построения фрактала «классическое дерево Пифагора». В скрипте использованы язык Python и библиотека Pygame для рисования. Скрипт выводит изображение фрактала.

Ключевые слова: Классическое дерево Пифагора, Pygame, Python.

Drawing a classic Pythagorean tree

Cherkashin Alexander Mihailovich

Sholom-Aleichem Priamursky State University

student

Abstract

The purpose of the research is the technology of constructing the fractal "classical Pythagorean tree". The script uses the Python language and the Pygame library for drawing. The script outputs an image of a fractal in the form.

Keywords: classic Pythagorean tree, Pygame, Python.

1 Введение

1.1 Актуальность исследования

Статья описывает возможность написания скрипта для рисования фрактала.

1.2 Цель исследования

Написание скрипта для рисования фрактала - классическое дерево Пифагора.

1.3 Обзор исследований

Т. Мунз рассматривает проблемы визуальным перекрытием ветви дерева Пифагора [1]. Ф. Бецк визуально анализировал дерево Пифагора использовал различные метрики параметров длина, ширина [2]. Л. Теиа описывает строение дерева Пифагора [3]. И. Рао исследовал оптимальный алгоритм дерева Пифагора [4]. П. Патлакас показал применение фракталы в дизайн [5]. К. В. Максименко-Шейко описал конструктивный подход построение уравнение теория R функций для объектов фрактальной геометрии [6].

2. Результаты и обсуждение

Скрипт написан на языке Python и использует библиотеки Numpy, Pygame. Функция `conv_hsv_rgb(h, s, v)` преобразует цветовую модель от HSV до RGB [7].

Переменная `screen_size` — размер окна, `mv` — координаты мыши от 0 до 1 по отношению к окну, `uv` — размер окна деленный на 2 (центральная точка).

В данной статье для построения фракталов, в скрипте определено функция `draw_fract(a, b, dep = 1)`, где `a, b` — точки, `dep` — глубина рекурсия.

Алгоритм:

1. $c_x = b_x - a_x$;
 $c_y = a_y - b_y$
2. $d_x = b_x - c_y$;
 $d_y = b_y - c_x$
3. $e_x = a_x - c_y$;
 $e_y = a_y - c_x$
4. $f_x = e_x + (c_x - c_y)$;
 $f_y = e_y - (c_x + c_y)$
5. Рисовать линии `[a, b, d, e]`
6. `draw_fract(e, f, dep - 1)`
7. `draw_fract(f, d, dep - 1)`

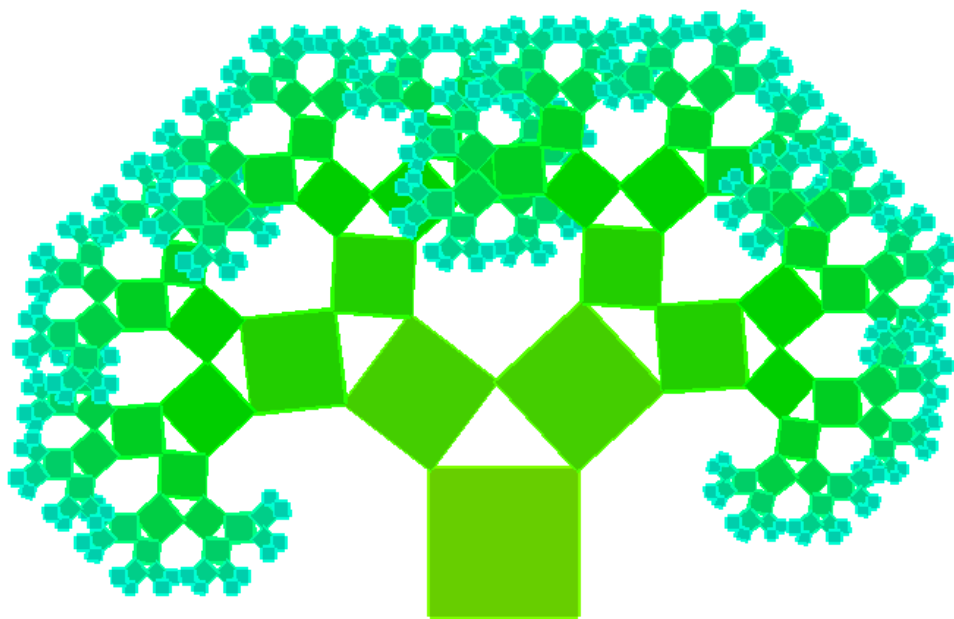


Рисунок 1. Фрактал дерево Пифагора

Листинг 1. Алгоритм фрактал дерево Пифагора

```

1  #!/usr/bin/python3
2  # -*- coding: utf-8 -*-
3  import pygame
4  import pygame.locals as lgm
5  import numpy as np
6  screen_size = np.array([800, 600])
7  mv = [0, 0]
8  glob_dep = 9
9  def conv_hsv_rgb(h, s, v):
10     c = v * s
11     if h < 0:
12         return np.array([0, 0, 0])
13     h = h / 60
14     x = c * (1 - np.abs(h % 2 - 1))
15     if 0 <= h <= 1:
16         return np.array([c, x, 0])
17     elif 1 < h <= 2:
18         return np.array([x, c, 0])
19     elif 2 < h <= 3:
20         return np.array([0, c, x])
21     elif 3 < h <= 4:
22         return np.array([0, x, c])
23     elif 4 < h <= 5:
24         return np.array([x, 0, c])
25     elif 5 < h <= 6:
26         return np.array([c, 0, x])
27     return np.array([0, 0, 0])
28 def draw_fract(a, b, dep = 1):
29     if (dep < 1):
30         return
31     c = np.array([0, 0])
32     c[0] = b[0] - a[0];
33     c[1] = a[1] - b[1];
34     d = np.array([b[0] - c[1], b[1] - c[0]])
35     e = np.array([a[0] - c[1], a[1] - c[0]])
36     f = np.array([e[0] + (c[0] - c[1]) * mv[0],
37                 e[1] - (c[0] + c[1]) * mv[1]])
38     gdw.lines(screen, conv_hsv_rgb(90 + (glob_dep-dep) / glob_dep * 90, 1.0, 1.0) * 255, True, [a, b, d, e], 5)
39     gdw.polygon(screen, conv_hsv_rgb(90 + (glob_dep-dep) / glob_dep * 90, 0.9, 0.9) * 255, [a, b, d, e])
40     draw_fract(e, f, dep-1)
41     draw_fract(f, d, dep-1)
42 if __name__ == '__main__':
43     pygame.init()
44     screen = pygame.display.set_mode(screen_size)
45     clock = pygame.time.Clock()
46     gdw = pygame.draw
47     pygame.display.set_caption("Fractal")
48     dcycles = False
49     screen.fill(np.array([1.0, 1.0, 1.0, 1.0]) * 255)
50     uv = screen_size / 2
51     while dcycles==False:
52         mouse = pygame.mouse.get_pos()
53         mv[0] = mouse[0] / screen_size[0]
54         mv[1] = mouse[1] / screen_size[1]
55         for event in pygame.event.get():
56             pygame.display.update()
57             if event.type == lgm.QUIT or (event.type == lgm.KEYDOWN and event.key == lgm.K_ESCAPE):
58                 dcycles = True
59         pygame.display.flip()
60         screen.fill(np.array([1.0, 1.0, 1.0, 1.0]) * 255)
61         draw_fract(uv + np.array([-50, 200]), uv + np.array([50, 200]), glob_dep)
62         clock.tick(60)
63     pygame.quit()

```

3 Выводы

В статье был описан скрипт, который использует алгоритм для рисования классического дерева Пифагора, в результате был нарисовано фрактал.

Библиографический список

1. Munz T. et al. Overlap-free drawing of generalized pythagoras trees for hierarchy visualization //2019 IEEE Visualization Conference (VIS). IEEE, 2019. С. 251-255.
2. Beck F. et al. Generalized pythagoras trees: A fractal approach to hierarchy visualization //International Joint Conference on Computer Vision, Imaging and Computer Graphics. Springer, Cham, 2014. С. 115-135.
3. Teia L. Anatomy of the Pythagoras' tree //Australian Senior Mathematics Journal. 2016. Т. 30. №. 2. С. 38-47.
4. Rao Y. et al. A new algorithm of geometric fractal in dynamic geometry //2012 Fourth International Conference on Digital Home. IEEE, 2012. С. 13-17.
5. Patlakas P. Maths & Crafts. 2014.
6. Maksymenko-Sheyko K. V., Sheyko T. I. Mathematical modeling of geometric fractals using R-functions //Cybernetics and Systems Analysis. 2012. Т. 48. №. 4. С. 614-620.
7. HSL and HSV. // Википедия URL: https://en.wikipedia.org/wiki/HSL_and_HSV#To_RGB (дата обращения: 2021-08-24).