УДК 004.92

FRACTAL GRAPHICS FOR VIRTUAL ENVIRONMENT GENERATION

I.A. TREGUBOVA, A.O. MAKOGANIUK, CH.I. MASHTALIR

O.S. Popov Odessa national academy of telecommunications,
1, Kuznechna St., Odessa, 65029, Ukraine
pisma-irine@ukr.net, rabotamao@i.ua, christina.mashtalir@gmail.com

ФРАКТАЛЬНА ГРАФІКА ДЛЯ ГЕНЕРАЦІЇ ВІРТУАЛЬНОГО СЕРЕДОВИЩА

І.А. ТРЕГУБОВА, А.О. МАКОГАНЮК, Х.І. МАШТАЛІР

Одеська національна академія зв'язку ім. О.С. Попова,
1, вул. Кузнечна, м. Одеса, 65029, Україна,
pisma-irine@ukr.net, rabotamao@i.ua, christina.mashtalir@gmail.com

Abstract. *Progress in hardware and software development is impressively fast. The main reason of computer graphics fast improvement is a full experience that can be reached though visual representation of our world. Therefore, the most interesting problem of it is a realistic image with high quality and resolution, which often requires procedural graphics generation. The article analyzes simplicity of a fractal and mathematics abstraction. Mathematics describes not only accuracy and logic but also beauty of the Universe. Mountains, clouds, trees, cells do not fit into the world of Euclidean geometry. They cannot be described by its methods. But fractals and fractal geometry solve that problem. Fractals are fairly simple equations on a sheet of paper with bright, unusual images, and, above all, they explain things. Fractal is a figure in the space, which consists of statistical character as the whole. It is self-similar, and therefore looks 'roughly' same and does not depend on its scale. So, any complex object can be called a fractal, if it has the same shape, as the parts it consists of. Fractal is abstract, and it helps to translate any algebraic problem into geometric, where solution is always obvious. A lot of researches in the field of fractal graphics has been carried out, but there are still issues that deserve considerable attention and more perfect solutions. The main purpose of the work is codes development with object-oriented programming languages for fractal graphics rendering. The article analyzes simplicity of a fractal and mathematics abstraction. Procedural generation was described as a method of algorithmic data generation for 3D models and textures creation. Code was written with general-purpose programming language Python, which renders step by step creation of fractal composition and variations of fractal images. Fractal generation used for modeling is part of realism in computer graphics In summary, procedural generation is very important for video games, as it can be used to automatically create large amount of game content. The random generation of natural looking landscapes is based on geometric computer generated images Created compositions can be used in computer science for image compression, in medicine for the study of the cellular level of organs, etc.*

Keywords: *fractal graphics, procedural generation, program code, 3D models.*

Анотація. Прогрес у розробці апаратних та програмних засобів вражає своєю неймовірною швидкістю. Основна причина одночасного швидкого вдосконалення комп'ютерної графіки – це потреба одержання повного досвіду, який можна досягти завдяки візуальному уявленню нашого світу. Тому найцікавішою її проблемою є реалістичне зображення з високою якістю та великою здатністю, яке часто вимагає процедурної генерації графіки. У статті проаналізовано простоту фрактальної та математичної абстракції. Математика відображає не тільки точність та логічність, а і незрівняну красоту. Гори, хмари, дерева, живі клітини – все виходить за рамки звичної евклідової геометрії. Їх не можливо описати її методами. Тоді на допомогу прийшли фрактали та наука при них. Фрактали –це достатньо прості математичні рівння на листі паперу з яскравими, дивовижними зображеннями , що дають повне уявлення про об'єкт. Фрактал є локальним об'єктом у просторі, який можно розкласти на схожі та ідентичні елементи, що зменьшуються. Це самоподібний об'єкт, а це означає, що його найменьші елементи є копіями найбільших елементів. Отже, фрактальним можно вважати любий об'єкт складної структури, якщо він має такуж форму, як і одна або більше його складових. Фрактал дозволяє мислити образно, а будь-яку алгебраїчну задачу перевести в область геометрії, де правильна відповідь завжди очевидна. Проведено досить багато досліджень в області фрактальної графіки,але ще залишилися питання, задачі та проблеми, що заслуговують значної уваги та більш досконалого вирішення. Метою даної роботи є розробка кодів з використанням різних об'єктно-орієнтованих мов програмування для візуалізації об'єктів франтальної графіки. В статті описана процедурна генерація як метод генерації алгоритмічних даних для створення 3D-моделей та текстур. Це важливо для відеоігор, оскільки його можна використовувати для автоматичного створення великої кількості ігрового контенту та є важливим моментом при моделюванні природних ландшафтів, заснованих на геометричних зображеннях.

Створені, на основі розроблених кодів, композиції можуть використовуватись в інформатиці при стисканні зображення, у медицині для дослідження клітиного рівня внутрішніх органів, тощо.

Ключеві слова: *фрактал, процедурна генерація, програмний код, 3D модель.*

1 INTRODUCTION

The world of mathematics always tends to be abstract. Mathematics describes not only accuracy and logic, but also beauty of the Universe. Mountains, clouds, trees, cells do not fit into the world of Euclidean geometry. They cannot be described by its methods. But fractals and fractal geometry solve that problem [1].

Fractals are fairly simple equations on a sheet of paper with bright, unusual images, and, above all, they explain things. Fractal is a figure in the space, which consists of statistical character as the whole. It is self-similar, and therefore looks ‘roughly’ same and does not depend on its scale. So, any complex object can be called a fractal, if it has the same shape, as the parts it consists of. Fractals describe the real world better, than classic mathematics and physics methods. Fractal is abstract, and it helps to translate any algebraic problem into geometric, where solution is always obvious. Discovering of fractals gives us explanation of the flora, landscapes, natural phenomena, blood vessels [2].

Fractal geometry studies and describes fractals, and fractal graphics render them. The main task of fractal geometry is to provide an answer, an understanding of nature and the whole universe.

Everything around us flows and at the same time changes with incredible speed, humanity needs new impressions, feelings and knowledge based on some new discoveries. With computer science progress, entertainment growths too. People want new unforgettable experience and that is possible with computer technologies improvement. Progress in hardware and software development is impressively fast. But as for full experience can be reached though visual representation of our world, the most important thing is computer graphics. The main and most interesting problem is a realistic image with high quality and resolution, and they can be achieved by procedural graphics generation.

Fractal graphics study everything around us. It can render the shape of the cloud not by the sphere, mountains not by the cones, flashes not by straight line. Fractal graphics can “grow” plants, which do not exist at all, but their uniqueness is beautiful [3].

A lot of researches in the field of fractal graphics has been carried out, but there are still issues that deserve considerable attention and more perfect solutions.

The main purpose of the work is codes development with object-oriented programming languages for fractal graphics rendering. Created compositions can be used in computer science for image compression, in medicine for the study of the cellular level of organs, for the programming of children's development games, in multiplication and cinematography, in modeling, etc.

Fractal-generating software. Fractal compositions can be created by special fractal graphics software or by programming languages. Fractal-generating software is any type of graphics software that generates images of fractals [4]. There are many fractal generating programs.

Fractal images are generated by special software. There is wide variety of program environments. The basic program environment includes filters or plugins, modules, and libraries. For example, powerful toolset Blender has modification for fractal creation. More complex, composite fractal images are created in combination with other graphics programs, such as Photoshop. Required fractal rendering can be achieved by script compilation in 3ds Max or Autodesk Maya. Also, fractal images can be retrieved via free web-interfaces. Both 2D and 3D fractals can be generated with web-resource Fractal Lab, which is written in Java and has an open-source Jwildfire fractal graphics editor with tools and features as in widely known software named Apophysis, which render fractal effects too. In computer graphics realistic images can be achieved by fractal modelling. The main and the most important problem is a realistic image creation with

high resolution and quality. In turn, image with high quality and necessary resolution can be received by procedural graphics generation. In computer science, procedural generation is a method of algorithmic data generation. And in computer graphics, it is called random generation and is commonly used for 3D models and textures creation. Procedural generation is very important for video games, as it can be used to automatically create large amount of game content, landscapes, etc.

Fractals are used for fractal antennas creation in telecommunication. Fractal antennas provide broadband performance in a small form factor. In turn, they are used for marine, airborne vehicles, or personal devices. Also, in the field of network technologies, many studies have been carried out to show the self-similarity of traffic transmitted by different kind of networks [5]. This is especially true of speech, audio and video services. At present, research and development of fractal compression of traffic transmitted over networks is being conducted, with the aim of more efficient communication of information.

2 TERRAIN GENERATOR

Imitation of natural land scape sand scenery generation can be done with Fractal generation software.

Advantages of procedural generation:

- Small file size;
- Large amount of content;
- Randomizing of content.

Fractal generation is used to build a graphical visualization of the terrain. You can render the fractals with algorithmic methods. Unwanted deterministic fractal surface. In the form of fractal surfaces, various phenomena can be represented, which are statistical self-similarity. A synchronously varying surface texture can create a realistic visual picture.

So, the main task is to implement terrain generator based on fractals. Procedural terrain generation requires a suitable programming language and a graphic library for correct generation of terrain. As the goal of our generator is to be used for the further game scene, it's better to select some game engine which provides 3D rendering, for example, Unity3d provides a wide variety of interfaces to create required 3D scene via Unity Scripting API.

There are two available languages: C# and JavaScript. Both are object-oriented languages, but as C# is more popular and more supported by engine, therefore, this language is preferable and was chosen for terrain generator.

The first step is a terrain object, which is a base for desirable landscape rendering. Terrain is created in Editor and then transferred to main logic script.

```
terrain = GetComponent<Terrain>();
```

The next step is a terrain configurations setup (e.g. resolution, size, height):

```
terrainData.heightmapResolution = (int)Width;
terrainData.size = new Vector3(Width, Depth, Height);
terrainData.SetHeights(0, 0, GenerateHeights());
```

After that, it's possible to start fractal generation as array of points in 2D space using desirable fractal formula for its computation.

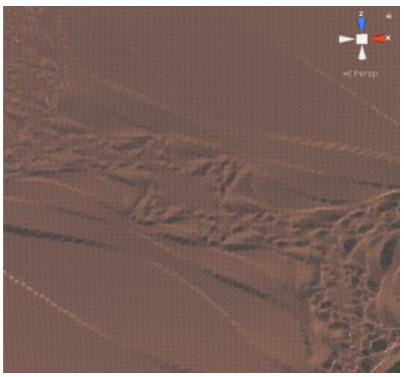
The next and most important part of terrain generation is to provide 3D representation of landscape, set all the heights and then render the result.

```
for (int x = 0; x < Width; ++x)
      {
for (int y = 0; y < Height; ++y)
         {
            heights[x, y] = CalculateHeights(fractal[x, y].x, fractal[x, y].y);
         }
      }
```

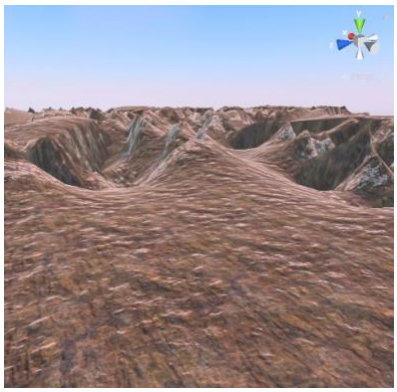
For heights calculation, there is simple pseudo-random generation of coordinates in space based on provided X and Y points, and global Width, Height and Depth defined as public values from the editor.

```
privatefloat CalculateHeights(float x, float y)
{
float xcoord = x / Width + Depth;
float ycoord = y / Height + Depth;
return Mathf.PerlinNoise(xcoord, ycoord);
}
```

Fig. 1 illustrates mountain scenery visualization rendered by use of the described above code.



a. Top view



b. Side view

Figure 1 – Generated Terrain

Other natural fractal compositions can be created in the same way

Rendering of fractals can be done by other object-oriented languages, e.g. Python [6].

You can use the graphics.py module to create elemental fractal rendering in Python and create a graphical user interface using tkinter package tools. However, in this work, the Turtle module was used to create graphic objects in a separate Python window.

Turtle graphics are vector graphics for complex shapes drawing from simple lines. Turtle graphics use a relative cursor (which is called 'Turtle') upon Cartesian plane [7]. It is a simple robot in the shape of a turtle that has such attributes, as:

Position in 2D space (X, Y)

Angle measured in degrees or radians.

Flag for pen that could be set to True/Down or False/Up.

The basic two-dimensional computer graphics in combination with the basic algorithmic elements were used to create a simple fractal. Initially, a data structure and related functions were repeating operations of the Turtle graphics system, and then this turtle is used to draw curves.

Below is a step by step creation of written code:

Standard library import:

```
import turtle
```

Animation pointer creation:

```
t = turtle.Pen()
```

Pointer's movement and rotation parameters:

```
t.left(360/sides + 1)
t.left(90)
```

Line width and color pallette:

```
colors = ["red", "yellow", "blue", "green", "orange", "purple"]
t.width (x*sides/200)
```

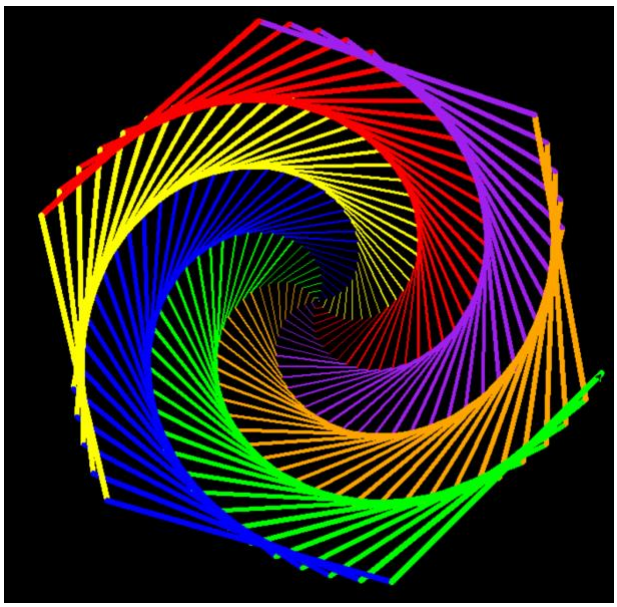
The result is written code, which shows basic animated fractal composition creation (Fig. 2.). It is possible to create a lot of variation of images by changing parameters.

Due to Python core simplicity, it is easy to implement tasks. The standard library includes a large amount of useful functions, so this program is the basis of a composite fractal image.

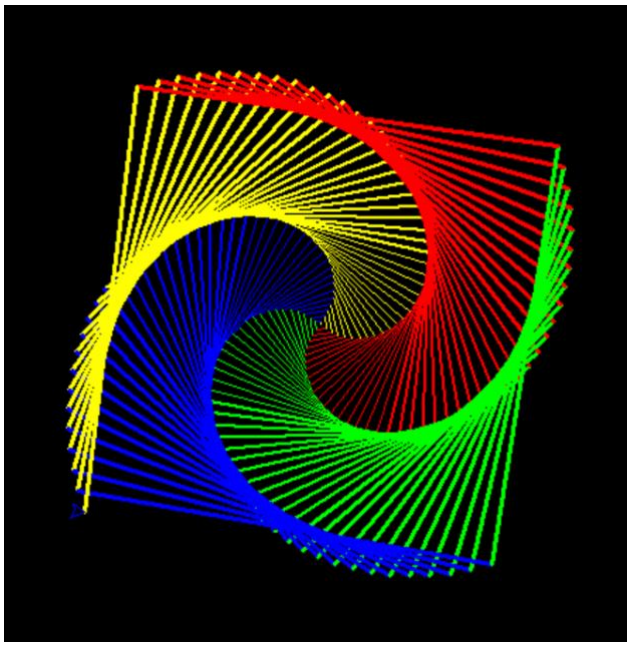
These compositions can be used in computer science when compressing images, for programming developing children's games, in medicine for the study of the cellular level of internal organs, in pedagogy and in other directions.

3 CONCLUSIONS

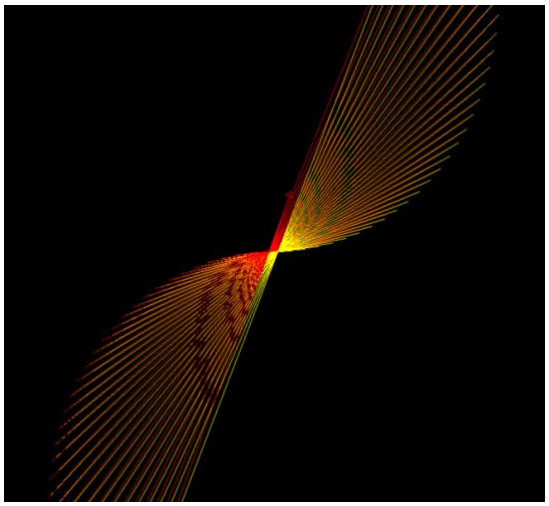
In this work application for 3D landscape models rendering was developed with object-oriented language C# and based on geometric, procedural generated objects, which provide abilities to implement landscape with fractals. Also, procedural generation is very important for video games, as it can be used to automatically create large amount of game content. Fractal generation used for modeling is part of realism in computer graphics.



a. Animation of 6 sides



b. Animation of 4 sides



c. Animation of 2 sides

Figure 2 – Animation of fractal compositions using the Turtle module

Code was written with general-purpose programming language Python, which renders step by step creation of fractal composition and variations of fractal images. Created compositions can be used to create new innovative projects in construction and design, medicine, cinema and many other industries.

References

1 Горобець Ю., А. Кучко, І. Вавілов. Фрактальна геометрія в природничих науках / Ю. Горобець, А. Кучко, І. Вавілов // –Наукова думка. – 2008. – с. 232.

2 Трегубова І.А., К.О. Собко, Р.О. Гохман. Фрактальна графіка як сучасна технологія візуалізації / І.А. Трегубова, К.О. Собко, Р.О. Гохман. // Цифрові технології. – Україна, Одеса: ОНАТ ім О.С.Попова. – 2018. –№24. – с.111–117, .

3 Чудеса фрактальной графики [Электронний ресурс]. – Режим доступу: http://zaholstom.ru/?p=4452

4 Шляхтина С. Обзор решений для генерации изображений на основе фракталов и атракторов [Электронний ресурс]. – Режим доступу: https://compress.ru/article.aspx?id=21776

5 Стрелковская И.В., Т.И. Григорьева, И.Н. Соловская. Самоподобный трафик в очереди G / M / 1, определяемый распределением Вейбулла /И.В. Стрелковская, Т.И. Григорьева, И.Н. Соловская// Радиоелектроника та системи связи. – 2018. – № 61 (3). – с. 128–134, .

6 Frame M., and A. Urry. Fractal Worlds, Grown, Built, and Imagined. – Yale University Press. – 5 May2017.– p.129, ISBN978-0300197877.

7 Haeseler F. V., H.-O. Peitgen, and G. Skordev. Linear Cellular Automata, Substitutions, Hierarchical Iterated Function Systems and Attractors / F. V. Haeseler, H.-O. Peitgen, and G. Skordev// Fractal Geometry and Computer Graphics. –Springer Science & Business Media. – 7 May 2017. – p.3, ISBN9783642956782.

References

1 Gorobets, Y., A.Kuchko, and I. Vavilov. Fraktal′na heometriya v pryrodnychykh naukakh (*Fractal geometry in the natural sciences)*. Naukova dumka, 2008.

2 Tregubova, I.A., K.O. Sobko, and R.O. Gokhman. “Fraktal′na hrafika yak suchasna tekhnolohiya vizualizatsiyi(Fractal Graphics as modern imaging technology).”, *Digital Technology*, Odessa, O.S.Popov ONAT, №24, 2018, pp.111–117.

3 Yepishin, Aleksey, “Chudesa fraktal'noy grafiki (The wonders of fractal graphics).” *Zaholstom*, http://zaholstom.ru/?p=4452

4 Shlyachtina, S.,“Obzor resheniy dlya generatsii izobrazheniy na baze fraktalov i attraktorov (Review of solutions for image generation based on fractals and attractors).” *compress*, https://compress.ru/article.aspx?id=21776

5 Strelkovskaya, I.V.,T.I. Grygoryeva, and I.N. Solovskaya. “Samopodobnyy trafik v ocheredi G / M / 1, opredelyayemyy raspredeleniyem Veybulla (Self-similar traffic in G/M/1 queue defined by the Weibull distribution).”,*Radioelectronics and Communications Systems*, No 61(3), 2018,pp. 128–134.

6 Frame, M., and A. Urry, “Fractal Worlds”,*Grown, Built, and Imagined*, Yale University Press, 5 May, 2017, p.129,ISBN978-0300197877.

7 Haeseler, F. V., H.-O. Peitgen, and G. Skordev.“Linear Cellular Automata, Substitutions, Hierarchical Iterated Function Systems and Attractors.” *Fractal Geometry and Computer Graphics*. Springer Science & Business Media, 7 May2017, p. 3, ISBN9783642956782.