

DOI <https://doi.org/10.15407/usim.2019.03.052>  
УДК 303.721;004.03142

**O.S. BULGAKOVA**, Ph. D (Eng.), Associate Professor,  
V.O. Sukhomlynsky Mykolaiv National University  
Nikolska str., 24, Mykolaiv, 54030, Ukraine  
sashabulgakova2@gmail.com

**A.V. KUDRIAVTSEV**, Master degree of Computer Science,  
V.O. Sukhomlynsky Mykolaiv National University,  
Nikolska str., 24, Mykolaiv, 54030, Ukraine  
extosis.vt@gmail.com

**V.V. ZOSIMOV**, Ph. D (Eng.), Associate Professor,  
V.O. Sukhomlynsky Mykolaiv National University  
Nikolska str., 24, Mykolaiv, 54030, Ukraine  
zosimovvv@gmail.com

**V.O. POZDEEV**, Doctor (Phys.-Math.), Chief of the Department,  
V.O. Sukhomlynsky Mykolaiv National University  
Nikolska str., 24, Mykolaiv, 54030, Ukraine  
pozdeevval@gmail.com

## ALGORITHMIC MODIFICATIONS IN PROCEDURAL GENERATION SYSTEMS

---

*The Modified Diamond Square algorithm is presented, as a result of which a 3-dimensional map of the fractal surface is obtained. This method of visualization using voxels allows to generate the relief structures (caves, canyons, quarries) that cannot be generated using a regular elevation map. The result of using the modified algorithm is shown on the landscape construction.*

**Keywords:** Procedural generation, 3D, Diamond Square algorithm, 3-dimensional map, computer graphics, game system

### Introduction

Procedural content generation is one of the most relevant and actively developing areas of the research in the field of multimedia, in particular in the game industry. The procedural generation of content (PGC) means the automatic and semi-automatic creation and dynamic change of various components of games, including game objects and levels, two-dimensional and three-dimensional graphics, effects, sounds, music, characters, plots, etc. [1].

The role of procedural generation algorithms for three-dimensional structures in computer graphics today is quite large. They are used to generate the effects of smoke, fog, etc. From the point of view of computer graphics, procedural generation is indispensable for creating artificial clouds, mountains, the surface of the sea, and the like. That is why the current research is the algorithms for generating three-dimensional structures to work with them.

Procedural generation is a type of computer algorithms that automatically generates content in

real time in accordance with the established rules, laws and restrictions [2–4]. Unlike content, when the designer manually determines each object and all parameters of the content being created, during procedural generation, the result can be somewhat unpredictable and can generate different content every time it is launched. One of the advantages of procedural generation is a reduction in the development costs of various commercial products. In the gaming industry, procedural generation is used in design as key objects of gaming systems.

However, it is easy to make mistakes in understanding and applying procedural generation: it is very important to understand that this is not a tool for solving all problems. It can be used to obtain a variety of digital content, or to introduce an element of randomness into objects that are long and difficult to do manually.

## Procedural generation algorithms

Consider some procedural generation algorithms.

**Perlin noise**, sometimes also classical Perlin noise is a mathematical algorithm for generating a procedural texture using a pseudo-random method.

Perlin noise is a gradient noise consisting of a set of pseudo-random unit vectors located at the specific points in space and interpolated by the smoothing function between these points, is the most used. The Perlin noise function generates textures using the pseudo-random number generation method, but all texture details are the same size. This property makes Perlin noise easily manageable; Many scalable copies of Perlin noise can be used to create a wide variety of procedural textures. Synthetic textures using Perlin noise are often used in CGI to make computer-generated visual elements – such as object surfaces, fire, smoke, or clouds – appear more natural, by imitating the controlled random appearance of textures in nature [5–6].

**Voronoi diagram.** In mathematics, a Voronoi diagram is a partitioning of a plane into regions based on distance of the points in a specific subset of the plane. That set of points (called seeds, sites, or generators) is specified beforehand, and for each seed there is a corresponding region consisting of all points closer to that seed than to any other.

These regions are called Voronoi cells. The Voronoi diagram of a set of points is dual to its Delaunay triangulation [7].

As implied by the definition, Voronoi cells can be defined for metrics other than Euclidean, such as the Mahalanobis distance or Manhattan distance. However, in these cases the boundaries of the Voronoi cells may be more complicated than in the Euclidean case, since the equidistant locus for two points may fail to be subspace of codimension 1, even in the 2-dimensional case.

A weighted Voronoi diagram is the one in which the function of a pair of points to define a Voronoi cell is a distance function modified by multiplicative or additive weights assigned to generator points. In contrast to the case of Voronoi cells defined using a distance which is a metric, in this case some of the Voronoi cells may be empty. A power diagram is a type of Voronoi diagram defined from a set of circles using the power distance; it can also be thought of as a weighted Voronoi diagram in which a weight defined from the radius of each circle is added to the squared distance from the circle's center [8].

The most common algorithm that gives one of the most realistic results is the **Diamond-square** (or square-diamond) algorithm, an extension of the one-dimensional midpoint displacement algorithm to a two-dimensional plane.

The diamond-square algorithm begins with a 2D square array of width and height  $2n + 1$ . At first, the four corner points of the array must be set to initial values. The diamond and square steps are then performed alternately until all array values have been set.

The algorithm starts with a 2D grid, then, from four initial values, randomly generates a height map, ordered as a grid of points, so that the whole plane is covered with squares.

During the square steps, the points located on the edges of the array will have only three adjacent values set rather than four. There are a number of ways to handle this complication – the simplest being to take the average of just the three adjacent values. Another option is to «wrap around», taking the fourth value from the other side of the array. When used with consistent initial corner values this method also allows

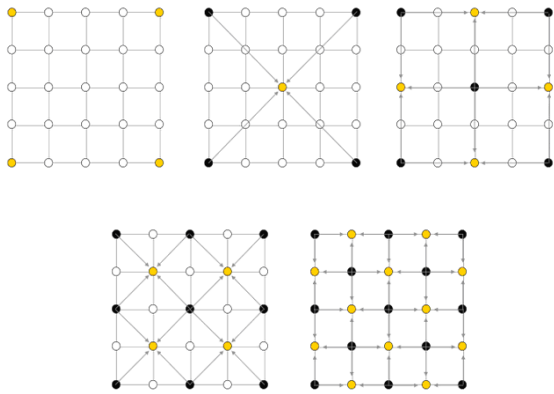


Fig. 1. The steps involved in running the Diamond-square algorithm on a 5× 5 array

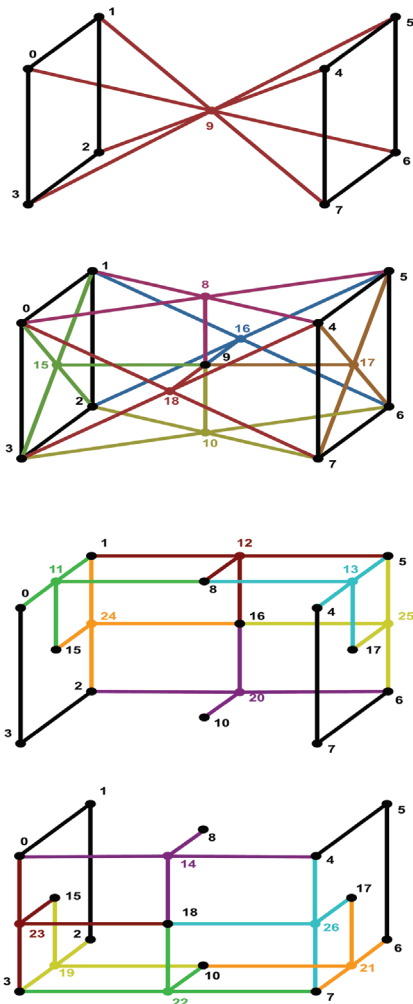


Fig. 2. Visualization of the Modified Diamond-square algorithm

generated fractals to be stitched together without discontinuities.

Fig. 1 shows the steps involved in running the diamond-square algorithm on a 5×5 array [7].

### Modified Diamond-square algorithm

The modified Diamond Square algorithm is the relief generation method, as a result of which we obtain a 3-dimensional map of a fractal surface that can be used in future projects. The following describes the structural modifications and an example application.

#### Algorithmic Modifications

The algorithm begins with a 3D grid, then, with eight initial values, pseudo-randomly generates a height map, ordered in the form of a 3-dimensional grid of points, so that the whole surface is covered with cubes.

The modified diamond-square algorithm starts with a three-dimensional array. The top surface of the points of the array is set to initial values as false, the lower surface of the points is true. Unlike the original algorithm, in which there are only two steps: diamond and square, the new algorithm has three steps: center, diamond and square. The center, diamond, and square steps are performed in turn until all array values are set. Table 1 shows stepwise differences between the diamond-square and modified diamond-square algorithms.

The following describes the stages of the algorithm, Fig. 2:

Step 1. Initialization of corner points. The assignment of values to them is true.

Step 2. Step center.

Step 2. Step diamond.

Step 3. Step square.

Step 4. Sequential, recursive calls itself eight times, until the values of all points are found.

This modification of the algorithm works on the principle of density, that is, there is a limit in the algorithm, by which the algorithm will fill the coordinate with the voxel, otherwise it will leave the free space, Fig. 3.

Table 1. Stepwise differences between the algorithms

Step	Diamond-square algorithm (2D)	Modified Diamond-square (3D)
The Centre step	—	For each cube in the array, there is a median point in which the average value of eight angular points is set + a random value*
The diamond step	For each square in the array, set the midpoint of that square to be the average of the four corner points + a random value	For each cube in the array, there are reference points where the average value of five points is set: four corner points and a center + a random value*.
The square step	For each diamond in the array, set the midpoint of that diamond to be the average of the four corner points + a random value.	For each cube in the array, vertices are set, which are assigned the arithmetic average of corner points + a random value*

\*this is a value added to the points decrease; after the height limit, the random value tends to zero.

This method of visualization using voxels allows to generate such relief structures as:

- caves
- careers
- rocks
- dumps
- canyons

and other structures that cannot be generated using a normal height map.

The algorithm presented in this paper has certain disadvantages:

- speed: the algorithm works with three-dimensional space, it needs to do approximately  $n^3 - n^2$  more actions than the original algorithm;
- control: the original algorithm differed from others in that it has only one way to control it through a random factor multiplier, which is an advantage because the less algorithm controls it has, the more realistic it can show, but in a modified algorithm this is a problem as, in most cases, it creates «flying» structures that need to be removed using a separate algorithm.
- memory: although the algorithm presented in the work uses the boolean data type, it uses a lot of RAM to process.

**Software Implementation**

In the Unity environment [9–12], demonstration for building game levels was created, which allows to construct maps based on the Modified Diamond

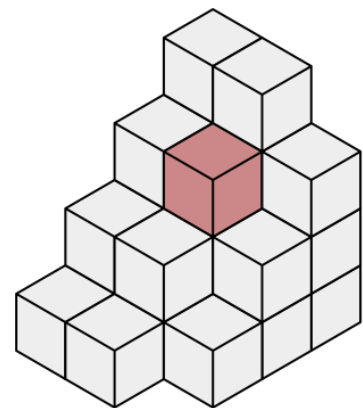


Fig. 3. Rendering voxels

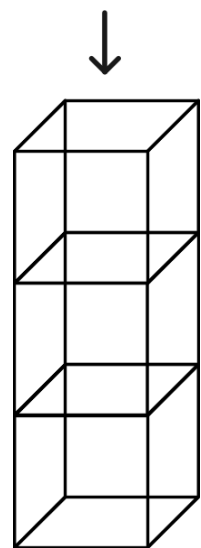
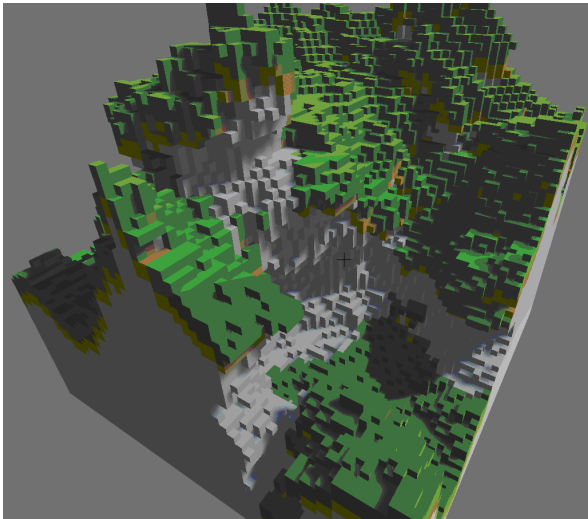


Fig. 4. The principle of auxiliary algorithm operation



*Fig. 5.* The result of the algorithm

Square algorithm. As a result of its work, we will obtain a 3-dimensional map of the fractal surface that can be used in future projects.

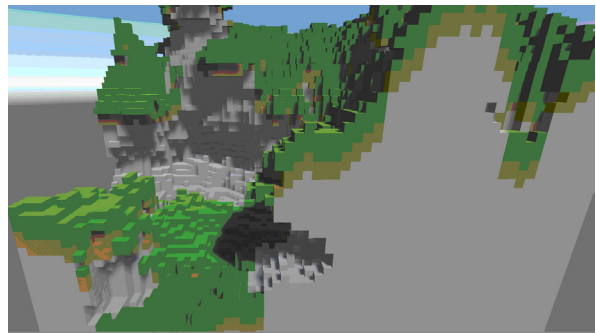
The source programme serves to demonstrate the operation of the algorithm, and is based on three scripts: the Modified Diamond Square algorithm, auxiliary processing algorithms, and a camera script.

To demonstrate the result of the work performed by the algorithm, it is necessary to be able to move relative to the model made, for this the camera object was selected and the displacement script was attached.

The result of the algorithm is rather unstable, it has single “flying blocks” and “standing blocks”; a script-handler was created to remove them. Also, to visualize the vegetation in this structure — the convenience of inspecting overhanging structures and caves, a script was created that script as rays from top to bottom, with the next blocks as occupied, Fig. 4.

## REFERENCES

1. Hendriks, M., 2013. “Procedural Content Generation for Games: A Survey”. *ACM Transactionson Multimedia Computing, Communications, and Applications*, 9 (1), pp. 1–22.
2. Togelius, J., Shaker, N., Nelson, M., 2014. *Procedural Content Generation in Games: A Textbook and an Overview of Current Research*. Berlin: Springer, 142 p.
3. Nelson, M., 2008. “Recombinable Game Mechanics for Automated Design Support”. *Proceedings of the Fourth AIIDE Conference*, pp. 84–89.



*Fig. 6.* The result of the algorithm on the other side

After fixing the auxiliary script on the game object, the parameters of the generating surface are entered in the LENGTH field, and the start button of the game is pressed, waiting for some time, depending on the computer’s capabilities, we get a voxel structure that resembles a steep slope or hollow. The result is shown in Fig. 4–5.

## Conclusions

The use of generation algorithms in computer technology is still a fairly new direction. Their use allows you to solve many current tasks for today. One of the main uses is 3D graphics.

The developed algorithm is described. In Unity, the demonstration for building game levels is created, which allows to construct a 3-dimensional map of the fractal surface based on the Modified Diamond Square algorithm. The presented algorithm operates on the principle of density. It means that the algorithm has a limit on the excess of which the algorithm will fill this voxel coordinate, otherwise — free place.

The presented algorithmic and programmed solution are useful for those working in the field of computer graphics and the computer game development.

4. Togelius, J., 2011. “Search-Based Procedural Content Generation: A Taxonomy and Survey”. IEEE Transactions on Computational Intelligence and AI in Games, 3 (3), pp. 172–186.
5. Perlin, K.H., [online]. Available at: <[https://en.wikipedia.org/wiki/Perlin\\_noise](https://en.wikipedia.org/wiki/Perlin_noise)> [Accessed 12 Aug. 2018].
6. Perlin, K.H. Making noise, [online]. Available at: <[noisemachine.com](http://noisemachine.com)> [Accessed 20 Dec. 2018].
7. Voronoi diagram, [online]. Available at: <[https://en.wikipedia.org/wiki/Voronoi\\_diagram](https://en.wikipedia.org/wiki/Voronoi_diagram)> [Accessed 27 Nov. 2018].
8. Arya, S., Malamatos, T., Mount, D., 2002. “Space-Efficient approximate Voronoi diarmmams”. Proc. 34th Ann. ACM. Symp. Theory of Comput. (STOC), pp. 721–730.
9. Edelsbrunner, H., 1987. Algorithms in Combinatorial Geometry, EATCS Monographs on Theoretical Computer Science, Springer-Verlag, 379 p.
10. Unity on Linux: Release Notes and Known Issues. Unity Community, [online]. Available at: <<https://forum.unity.com>> [Accessed 27 Nov. 2018].
11. Bulgakova, O., Kyslychenko, P., 2016. “The large scalable applications architecture in object-oriented programming context”. Geometric modeling and information technologies, 1, pp. 52–56.
12. Bulgakova, O., Nesterenko, K., 2018. “Features of three-dimensional modeling in geoinformation systems on the basis of perlin noise”. NUS Journal, 3, pp. 59–80.

Received 24.03.2019

*О.С. Булгакова*, кандидат техн. наук, доцент,  
Миколаївський національний університет імені В.О. Сухомлинського,  
вул. Нікольська, кв. 24, 54030, м. Миколаїв, Україна,  
sashabulgakova2@gmail.com

*А.В. Кудрявцев*, магістр,  
Миколаївський національний університет імені В.А. Сухомлинського,  
вул. Нікольська, 24, 54030, м. Миколаїв, Україна  
extosis.vt@gmail.com

*В.В. Зосімов*, кандидат техн. наук, доцент,  
Миколаївський національний університет імені В.А. Сухомлинського,  
вул. Нікольська, 24, 54030, м. Миколаїв, Україна,  
zosimovvv@gmail.com

*В.О. Поздєєв*, д-р фізико-математичних наук, зав. кафедрою,  
Миколаївський національний університет імені В.О. Сухомлинського,  
вул. Нікольська, кв. 24, 54030, м. Миколаїв, Україна,  
pozdeevval@gmail.com

## АЛГОРИТМІЧНІ МОДИФІКАЦІЇ В СИСТЕМАХ ПРОЦЕДУРНОЇ ГЕНЕРАЦІЇ

**Вступ.** Процедурна генерація незамінна для створення штучних хмар, гір, поверхні моря тощо. Саме тому нинішніми дослідженнями є алгоритми генерації тривимірних структур для роботи з ними.

На відміну від того, коли дизайнер вручну визначає кожен об'єкт і всі параметри створеного контенту, під час процедурного генерування результат може бути дещо непередбачуваним і може генерувати різний вміст при кожному запуску. Однією з переваг процедурної генерації є зниження витрат на розробку різних комерційних продуктів.

Процедурна генерація може бути використана для отримання різного цифрового контенту або для створення елемента в об'єкті, який довго і складно робити вручну.

**Мета статті** — представлення модифікованого алгоритму *Diamond Square*, в результаті роботи якого можна отримати тривимірну карту фрактальної поверхні.

**Методи.** У середовищі *Unity* була створена демонстрація для побудови ігрових рівнів, яка дозволяє будувати карти на основі модифікованого алгоритму *Diamond Square*. В результаті своєї роботи отримуємо 3-мірну карту фрактальної поверхні, яка може бути використана у майбутніх проектах.

Вихідна програма служить для демонстрації роботи алгоритму і ґрунтується на трьох сценаріях: модифікованому алгоритмі *Diamond Square*, допоміжних алгоритмах обробки і сценаріїв камери.

**Результати.** В статті представлено модифікований алгоритм *Diamond Square*, в результаті роботи якого отримуємо тривимірну карту фрактальної поверхні. Результат використання модифікованого алгоритму показано на прикладі побудови ландшафтної конструкції.

**Висновки.** Запропонований метод візуалізації процедурної генерації на основі модифікованого алгоритму *Diamond Square* з використанням вокселів дозволяє створювати рельєфні структури (печери, каньйони, кар'єри), які неможливо створити за допомогою звичайної карти висот.

**Ключові слова:** *процедурна генерація, 3D, алгоритм Diamond Square, 3-вимірна карта, комп'ютерна графіка, ігрові системи.*

*A.C. Булгакова*, кандидат техн. наук, доцент,  
Николаевский национальный университет имени В.А. Сухомлинского,  
ул. Никольская 24, 54030, г. Николаев, Украина,  
sashabulgakova2@gmail.com

*A.B. Кудрявцев*, магистр,  
Николаевский национальный университет имени В.А. Сухомлинского,  
ул. Никольская 24, 54030, г. Николаев, Украина  
extosis.vt@gmail.com

*V.V. Зосимов*, кандидат техн. наук, доцент,  
Николаевский национальный университет имени В.А. Сухомлинского,  
ул. Никольская 24, 54030, г. Николаев, Украина,  
zosimovvv@gmail.com

*V.A. Поздеев*, д-р физико-математических наук, зав. кафедрой,  
Николаевский национальный университет имени В.А. Сухомлинского,  
ул. Никольская, кв. 24, 54030, г. Николаев, Украина,  
pozdeevval@gmail.com

## АЛГОРИТМИЧЕСКИЕ МОДИФИКАЦИИ В СИСТЕМАХ ПРОЦЕДУРНОЙ ГЕНЕРАЦИИ

**Введение.** Процедурная генерация незаменима для создания искусственных облаков, гор, поверхности моря и т.п. Именно поэтому нынешними исследованиями являются алгоритмы генерации трехмерных структур для работы с ними.

В отличие от того, когда дизайнер вручную определяет каждый объект и все параметры созданного контента, при процедурной генерации результат может быть несколько непредсказуемым и может генерировать разное содержание при каждом запуске. Одним из преимуществ процедурной генерации есть снижение затрат на разработку различных коммерческих продуктов.

Процедурная генерация может быть использована для получения разнообразного цифрового контента или для создания элемента в объекте, который долго и сложно делать вручную.

**Цель статьи** — представление модифицированного алгоритма *Diamond Square*, в результате работы которого можно получить трехмерную карту фрактальной поверхности.

**Методы.** В среде *Unity* была создана демонстрация для построения игровых уровней, которая позволяет строить карты на основе модифицированного алгоритма *Diamond Square*. В результате работы алгоритма получается 3-мерная карта фрактальной поверхности, которая может быть использована в будущих проектах.

Исходная программа служит для демонстрации работы алгоритма и основывается на трех сценариях: модифицированном алгоритме *Diamond Square*, вспомогательных алгоритмах обработки и сценариев камеры.

**Результаты.** В статье представлен модифицированный алгоритм *Diamond Square*, в результате работы которого получаем трехмерную карту фрактальной поверхности. Результат использования модифицированного алгоритма показан на примере построения ландшафтной конструкции.

**Выводы.** Предложенный метод визуализации процедурной генерации на основе модифицированного алгоритма *Diamond Square* с использованием вокселей позволяет создавать рельефные структуры (пещеры, каньоны, карьеры), которые невозможно создать с помощью обычной карты высот.

**Ключевые слова:** *процедурная генерация, 3D, алгоритм Diamond Square, 3-мерная карта, компьютерная графика, игровые системы.*