

S.O. Korda, A.G. Kovaleva  
Ural Federal University named after the first President of Russia B.N.  
Yeltsin  
Yekaterinburg, Russia

## PROCEDURAL CONTENT GENERATION

**Abstract:** This article presents the brief history of procedural content generation and clarifies the general meaning of this process. After the description of key events and developments in this field, several definitions of the term «procedural content generation» according to researchers are given. Based on these definitions, a variant of a single interpretation of the term is discussed in the paper.

**Keywords:** virtual worlds, procedural content generation, multimedia content creation.

С.О. Корда, А.Г. Ковалева  
Уральский федеральный университет имени первого Президента  
России Б.Н. Ельцина  
Екатеринбург, Россия

## ПРОЦЕДУРНАЯ ГЕНЕРАЦИЯ КОНТЕНТА

**Аннотация:** Данная статья представляет краткую историю процедурной генерации контента и разъясняет общее значение этого процесса. За описанием ключевых событий и разработок в данной сфере следуют определения термина «процедурная генерация контента», данные исследователями в данной области. На основе этих определений в конце статьи представлен возможный вариант формулировки термина, раскрывающий его значение в общем значении.

**Ключевые слова:** виртуальные миры, процедурная генерация контента, создание мультимедиа контента.

We live in the age of information that surrounds us constantly. This incredible amount of data is often perceived by us from a consumer point of view and there is no need to think about its source. But the content is always a result of some intellectual work, which means it has to be as much optimized as possible, especially if this content should be generated in large quantities.

There are plenty of methods to automate content generation in particular spheres. Procedural generation is one of them. It is mostly used by game developers when their goal is to obtain a large amount of data that does not have to be as complex as manually made objects. Therefore, it is reasonable to use techniques based on randomization. In these cases, they appeal to the use of procedural generation system, a complex of software solutions that implement procedural generation algorithms. Procedural generation system's running result is a set of specified type quasi-random data whose content is based on algorithms the system implements.

What is procedural content generation or PCG? It is complicated to give a single definition that gives the meaning of this method in accordance to every sphere of its application. Plenty of interpretations of procedural content generation are discussed in different research. Dwelling on the history of PCG and emphasizing the challenges that developers have tried to overcome using procedural algorithms helps to clarify its function and interpret its general meaning, before resorting to give any definitions.

### **History of PCG**

The first developments in the field of procedural content generation appeared in the late seventies. The most significant game which used algorithms to create levels was «Rogue» that came out in 1980. Its impact caused all the following games with the same mechanics (including procedurally generated levels) to be called «rogue-like games». The developers generally have chosen the generative approach for the level creation. In fact, these days, the memory requirements of a game with many different levels were too high so that the decision was made to generate the levels on the fly instead of writing them on a disk.

Darwin R. Peachey's paper Solid Texturing, published in 1985, was one of the first publications discussing procedural content generation.

Similar to today's normal mapping, Peachey proposed a technique that enabled two-dimensional textures to look three-dimensional [1].

The derived methods were also used in the animation movie sector. For example, Pixar's animation tool RenderMan offered procedural functions to define textures and materials algorithmically.

In 2004, the group .theprodukkt published a simple shooter called .kkrieger (Figure 1), having its origin in the demoscene. A particular category focuses on demos with a maximum file size of 4, 8 or 64 kb. This small footprint does not allow the usage of prefabricated assets such as graphics, music or models. Instead, those resources are generated procedurally during runtime or before the application starts. For example, in .kkrieger developers created animations, levels, textures, shaders and music based on a tool set offering procedural methods. The final release of .kkrieger had a size of 96 kb. The developers decided to publish their tools as the editor application called .werkzeug.



Figure 1 – Screenshot of kkriger's gameplay

Compton, Osborn and Mateas determine Computer Graphics as the origin of procedural generation and state that the discipline was connected to computer games with a publication of Intel. This particular paper surveyed generative techniques such as L-systems, Perlin noise and fractals [2].

Nowadays, most of the discussed utilities have appeared in modern game development tools and make procedural content generation easier. 3D models are constructed by deformation, cropping or merging primitives instead of assembling them triangle by triangle. In other words, developers focus on the procedural algorithms to generate content instead of tackling the low-level technical challenges. Game engines Unreal Engine 4, Unity

2017 and Cry Engine V provide tools to simply manipulate terrains and paint vegetation like trees or flowers onto the ground that speed up the level creation workflow. Developers focus on finding an algorithm to distribute vegetation objects over the level without taking into account objects collisions and graphical artifacts.

A remarkable achievement of today's game development tools is the procedural generation of textures and materials. The procedural generation of textures and materials is now applied by most of the big companies in the game engine market. In addition, instead of writing shaders by hand, modular tools were offered to graphic designers to construct materials by combining images, mathematical operations or shader functions to create a normal mapping effect as shown in Figure 2. The next game engine generation has begun to implement features to create even more complex objects.

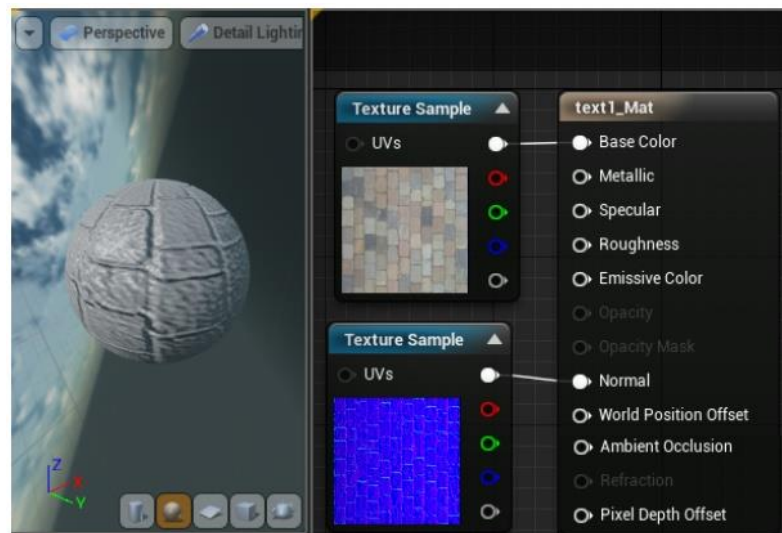


Figure 2 - The Unreal Engine's material editor by Epic Games

### Definition of PCG

PCG is not only a subject of research in computer science. Prusinkiewicz and Lindenmayer emphasize the growing interests in other communities caused by the interdisciplinarity, affecting natural sciences like biology [3].

Hendrikx define procedural generation as an alternative to manual design but emphasize the need for a possible parameterization so that designers may edit the generated object. He also introduces the abbreviation PCG-G (Procedural Content Generation for Games) in order to separate PCG for games from other fields such as simulations or movies [4]. This shows that PCG-based methods, algorithms and tools may be

applied to a large variety of fields. Pixar has already been named the company using procedural content generation in RenderMan. Additionally, Disney's research provides publications mentioning procedural techniques, for virtual terrain editing, showing that there are advantages of automated content generation in the movie industry.

Ruben M. Smelik defines procedural content generation as «any kind of automatically generated asset based on a limited set of user-defined input parameters» [5].

Roden and Parberry call these kinds of algorithms amplification algorithms, taking a small set of input parameters to transform them to a larger set of output data [6].

Togelius formulates a definition by an antithesis, saying that procedurally generated content does not correspond to content that is generated by users even if they make use of procedural algorithms since they have to be manually parameterized [7].

Shaker is more exact and defines PCG by giving examples of PCG like a software tool to generate random dungeons without any user input and a map editor that lets users place items.

Jonas Freiknecht and Wolfgang Effelsberg give their own definition of PCG: the automatic creation of digital assets for games, simulations or movies based on predefined algorithms and patterns that require a minimal user input [8].

Procedural content generation is a relatively young sphere of computer science. Although its principles, which were founded at the beginning, still take place in modern PCG systems. As it is demonstrated in this paper, at the beginning developers were motivated to use procedural generation in order to fit the memory requirements. In other words, PCG was a forced measure to minimize resource. However, nowadays there is an opposite situation: data storage has become much cheaper and capacious enough to keep enough amount of data to store everything. Developers, not being chained by technical aspects of memory, now use PCG to obtain incredible amount of content, like virtual worlds, high-resolution textures, etc.

The analysis of different researches and implementations in the field of PCG clearly shows its complexity and variety. As it turned out it is complicated to give a single universal definition of what procedural generation is. Considering all reviewed definitions and summarizing them, procedural generation may be defined as a complex of methods and

algorithms, which are intended to generate data automatically, instead of designing it in a manual way.

## REFERENCES

1. Peachey D.R. Solid Texturing of Complex Surfaces. In Proceedings of the 12th Annual Conference on Computer Graphics and Interactive Techniques, San Francisco, CA, USA, 22–26 July 1985; ACM: New York, NY, USA, 1985; pp. 279–286.
2. Compton K., Osborn J.C., Mateas M. Generative methods. In Proceedings of the Fourth Procedural Content Generation in Games Workshop, Chania, Greece, 14–17 May 2013.
3. Prusinkiewicz P., Lindenmayer A. The Algorithmic Beauty of Plants; Springer Science & Business Media: Berlin, Germany, 2012.
4. Hendrikx M., Meijer S., Van Der Velden J., Iosup A. Procedural Content Generation for Games: A Survey. *ACM Trans. Multimedia Comput. Commun. Appl.* 2013, 9, doi:10.1145/2422956.2422957.
5. Smelik R.M., Tutenel T., de Kraker K.J., Bidarra R. A declarative approach to procedural modeling of virtual worlds. *Comput. Graph.* 2011, 35, 352–363.
6. Roden T., Parberry I. From artistry to automation: A structured methodology for procedural content creation. In Proceedings of the International Conference on Entertainment Computing (ICEC 2004), Eindhoven, The Netherlands, 1–3 September 2004; pp. 301–304.
7. Togelius J., Kastbjerg E., Schedl D., Yannakakis G.N. What is procedural content generation? Mario on the borderline. In Proceedings of the 2nd International Workshop on Procedural Content Generation in Games, Bordeaux, France, 28 June 2011; ACM: New York, NY, USA, 2011.
8. Freiknecht J., Effelsberg W. A Survey on the Procedural Generation of Virtual Worlds. *Multimodal Technologies Interact.* 2017, 1, 27.