

Procedural generation

In [computing](#), **procedural generation** is a method of creating data [algorithmically](#) as opposed to manually, typically through a combination of human-generated assets and algorithms coupled with computer-generated randomness and processing power. In [computer graphics](#), it is commonly used to create [textures](#) and [3D models](#). In video games, it is used to automatically create large amounts of content in a game. Depending on the implementation, advantages of procedural generation can include smaller file sizes, larger amounts of content, and randomness for less predictable gameplay. Procedural generation is a branch of [media synthesis](#).

Contents

Overview

Contemporary application

- Tabletop role-playing games

- Video games

 - Early history

 - Modern use

- Film

See also

References

Further reading

External links

One example of procedural generation, here used to generate realistic looking tree models. Different models can be generated by changing both deterministic parameters and a random seed.

Overview

The term *procedural* refers to the process that computes a particular function. [Fractals](#) are geometric patterns which can often be generated procedurally. Commonplace procedural content includes [textures](#) and [meshes](#). [Sound](#) is often also procedurally generated, and has applications in both [speech synthesis](#) as well as music. It has been used to create compositions in various genres of [electronic music](#) by artists such as [Brian Eno](#) who popularized the term "[generative music](#)".^[1]

While software developers have applied procedural generation techniques for years, few products have employed this approach extensively. Procedurally generated elements have appeared in earlier video games: *[The Elder Scrolls II: Daggerfall](#)* takes place in a mostly procedurally generated world, giving a world roughly two thirds the actual size of the [British Isles](#). *[Soldier of Fortune](#)* from [Raven Software](#) uses simple routines to detail enemy models, while [its sequel](#) featured a [randomly-generated level](#) mode. [Avalanche](#)

Studios employed procedural generation to create a large and varied group of detailed tropical islands for *Just Cause: No Man's Sky*, a game developed by games studio Hello Games, is all based upon procedurally generated elements.

The modern demoscene uses procedural generation to package a great deal of audiovisual content into relatively small programs.

New methods and applications are presented annually in conferences such as the IEEE Conference on Computational Intelligence and Games and Artificial Intelligence and Interactive Digital Entertainment.^[2]

Particularly in the application of procedural generation with video games, which are intended to be highly replayable, there are concerns that procedural systems can generate infinite numbers of worlds to explore, but without sufficient human guidance and rules to guide these, the result becomes "procedural oatmeal", a term coined by writer Kate Compton, in that while it is possible to mathematically generate thousands of bowls of oatmeal with procedural generation, they will be perceived to be the same by the user, and lack the notion of perceived uniqueness that a procedural system should aim for.^[3]

Contemporary application

Tabletop role-playing games

Using procedural generation in games had origins in the tabletop role playing game (RPG) venue.^[4] The leading tabletop system, *Advanced Dungeons & Dragons*, provided ways for the "game master" to generate dungeons and terrain using random die rolls, expanded in later editions with complex branching procedural tables. Strategic Simulations under license from TSR released the *Dungeon Master's Assistant*, a computer program that generated dungeons based on these published tables. *Tunnels & Trolls*, also published by TSR, was designed primarily around solitary play and used similar procedural generation for its dungeons. Other tabletop RPGs borrowed similar concepts in procedural generation for various world elements.^[5]

Video games

Early history

Prior to graphically oriented video games, roguelike games, a genre directly inspired by *Dungeons & Dragons* adopted for solitaire play, heavily utilized procedural generation in the same manner that tabletop systems had done. Such early games include *Beneath Apple Manor* (1978) and the genre's namesake, *Rogue* (1980). The procedural generation system in roguelikes would create dungeons in ASCII- or regular tile-based systems and define rooms, hallways, monsters, and treasure to challenge the player. Roguelikes, and games based on the roguelike concepts, allow the development of complex gameplay without having to spend excessive time in creating a game's world.^[6]

A procedurally generated dungeon map in the videogame *NetHack*

1978's *Maze Craze* for the Atari VCS used an algorithm to generate a random, top-down maze for each game.^[7]

Some games used pseudorandom number generators. These PNGs were often used with predefined seed values in order to generate very large game worlds that appeared to be pre-made. *The Sentinel* supposedly had 10,000 different levels stored in only 48 and 64 kilobytes. An extreme case was *Elite*, which was originally planned to contain a total of 2^{48} (approximately 282 trillion) galaxies with 256 solar systems each. However, the publisher was afraid that such a gigantic universe would cause disbelief in players, and eight of these galaxies were chosen for the final version.^[8] Other notable early examples include the 1985 game *Rescue on Fractalus* that used fractals to procedurally create, in real time, the craggy mountains of an alien planet and *River Raid*, the 1982 Activision game that used a pseudorandom number sequence generated by a linear feedback shift register in order to generate a scrolling maze of obstacles.

Modern use

Though modern computer games do not have the same memory and hardware restrictions that earlier games had, the use of procedural generation is frequently employed to create randomized games, maps, levels, characters, or other facets that are unique on each playthrough.^{[9][10]}

In 2004, a PC first-person shooter called *.kkrieger* was released by a German demo group. It is entirely contained in a 96 kilobyte executable for Microsoft Windows that generates hundreds of megabytes of 3D and texture data when run. According to one of the programmers, "it was a complete failure as far as the game side was concerned (mostly because no-one involved really deeply cared about that aspect)."^[11]

Naked Sky's RoboBlitz used procedural generation to maximize content in a less than 50 MB downloadable file for Xbox Live Arcade. *Will Wright's Spore* also makes use of procedural synthesis.

Procedural generation is often used in loot systems of quest-driven games, such as action role-playing games and massive multiplayer online role playing games. Though quests may feature fixed rewards, other loot, such as weapons and armor, may be generated for the player based on the player-character's level, the quest's level, their performance in the quest, and other random factors. This often leads to loot having a rarity quality applied to reflect when the procedural generation system has produced an item with better-than-average attributes. For example, the *Borderlands* series is based on its procedural generation system which can create over a million unique guns and other equipment.^[12]

Many open world or survival games procedurally create a game world from a random seed or one provided by the player, so that each playthrough is different. These generation systems create numerous pixel- or voxel-based biomes with distribution of resources, objects, and creatures. The player frequently has the ability to adjust some of the generation parameters, such as specifying the amount of water coverage in a world. Examples of such games include *Dwarf Fortress* and *Minecraft*. An artifact of the procedural generation around these games is that if the space that the player is allowed to explore is not limited, the randomness of the procedural generation will start to produce more noise than content; this is exemplified in the idea of the "Far Lands" within some earlier versions of *Minecraft*, where the usual smooth transitions between biomes was replaced with haphazard formations.^[13]

Procedural generation is also used in space exploration and trading games. *Elite: Dangerous*, through using the 400 billion known stars of the Milky Way Galaxy as its world basis, uses procedural generation to simulate the planets in these solar systems. Similarly, *Star Citizen* uses the technology for its planets, to create a collection of seamlessly-loaded planet-sized planets among its hand-crafted universe. *I-Novae Infinity* features a plethora of planets which are procedurally generated between which the player can travel via space ships. *Outerra Anteworld* is a video game in development that uses procedural generation and real world data to create a virtual replica of planet Earth in true scale. *No Man's Sky* features a universe containing 18 quintillion planets which are procedurally generated on the fly as the player encounters them, including their terrain, weather, flora, and fauna, as well as a number of space-faring alien species. This

universe is defined by the use of a single random seed number to their deterministic engine, assuring that the same content will be generated at the same places for all players, which enables players to share discoveries using only knowledge of the locations of the planets in the virtual galaxy.^{[14][15]}

Film

As in video games, procedural generation is often used in film to create visually interesting and accurate spaces rapidly. This comes in a wide variety of applications.

One application is known as an *imperfect factory*, where artists can rapidly generate many similar objects. This accounts for the fact that, in real life, no two objects are ever exactly alike. For instance, an artist could model a product for a grocery store shelf, and then create an imperfect factory to generate many similar objects to populate the shelf.

MASSIVE is a high-end computer animation and artificial intelligence software package used for generating crowd-related visual effects for film and television. It was developed to create fighting armies of hundreds of thousands of soldiers for Peter Jackson's *The Lord of the Rings* films automatically.^[16]

Coherent noise can be extremely important to procedural workflow in film. Simplex noise is often faster with fewer artifacts, though an older function called Perlin noise may be used as well. Coherent noise, in this case, refers to a function that generates smooth pseudo-randomness in **n** dimensions.

See also

- Computational creativity
- Fractal landscape
- Fractional Brownian motion
- Generative art
- Linear congruential generator
- List of games using procedural generation
- Media synthesis (AI)
- Noise, Perlin noise, Simplex noise
- Procedural animation
- Procedural modeling
- Procedural texture
- Random dungeon
- Random map
- Scenery generator

References

1. Brian Eno (June 8, 1996). "A talk delivered in San Francisco, June 8, 1996" (<http://www.inmotionmagazine.com/eno1.html>). *inmotion magazine*. Retrieved 2008-11-07.
2. "Artificial Intelligence and Interactive Digital Entertainment" (<http://www.aiide.org/>). AIIDE.org. Retrieved 12 June 2016.

3. Cook, Michael (August 10, 2016). "Alien Languages: How We Talk About Procedural Generation" (https://www.gamasutra.com/blogs/MichaelCook/20160825/279938/Alien_Languages_How_We_Talk_About_Procedural_Generation.php). *Gamasutra*. Retrieved October 7, 2019.
4. Brown, Joseph Alexander; Scirea, Marco (2018). "Procedural Generation for Tabletop Games: User Driven Approaches with Restrictions on Computational Resources". *SEDA 2018: Proceedings of 6th International Conference in Software Engineering for Defence Applications*. International Conference in Software Engineering for Defence Applications. Rome, Italy. pp. 44–54.
5. Smith, Gillian (2015). *An Analog History of Procedural Content Generation* (http://www.fdg2015.org/papers/fdg2015_paper_19.pdf) (PDF). Foundations of Digital Games 2015. Pacific Grove, California. Retrieved October 7, 2019.
6. Hatfield, Tom (2013-01-29). "Rise Of The Roguelikes: A Genre Evolves" (<http://pc.gamespy.com/pc/ftl-faster-than-light/1227287p1.html>). *GameSpy*. Retrieved 2013-04-24.
7. "Maze Craze" (http://www.atarimania.com/game-atari-2600-vcs-maze-craze-a-game-of-cops--n-robbers_18243.html). *Atari Mania*.
8. Francis Spufford (October 18, 2003). "Masters of their universe" (<https://www.theguardian.com/weekend/story/0,3605,1064107,00.html>). *Guardian*.
9. Moss, Richard (January 1, 2016). "7 uses of procedural generation that all developers should study" (http://www.gamasutra.com/view/news/262869/7_uses_of_procedural_generation_that_all_developers_should_study.php). *Gamasutra*. Retrieved January 1, 2016.
10. Baker, Chris (9 August 2016). "'No Man's Sky': How Games Are Building Themselves" (<https://www.rollingstone.com/culture/news/no-mans-sky-how-games-are-building-themselves-w433492>). *Rolling Stone*. Retrieved 9 August 2016.
11. Giesen, Fabien (April 8, 2012). "Metaprogramming for madmen" (<https://fgiesen.wordpress.com/2012/04/08/metaprogramming-for-madmen/>). *The ryg blog*.
12. Kuo, Ryan (April 19, 2012). "Why Borderlands 2 Has the Most Stylish Guns in Gaming" (<http://blogs.wsj.com/speakeasy/2012/04/19/why-borderlands-2-has-the-most-stylish-guns-in-gaming/>). *Wall Street Journal*. Retrieved April 21, 2016.
13. Peckham, Matt (8 August 2016). "NO MAN'S SKY IS WILDLY AMBITIOUS, UTTERLY VAST AND A HUGE CHALLENGE TO THE VIDEO GAME INDUSTRY'S STATUS QUO" (<http://time.com/no-mans-sky/>). *Time*. Retrieved 9 August 2016.
14. Khatchadourian, Raffi (18 May 2015). "World without end : creating a full-scale digital cosmos" (<http://www.newyorker.com/magazine/2015/05/18/world-without-end-raffi-khatchadourian>). *Annals of Games*. *The New Yorker*. **91** (13). pp. 48–57. Retrieved 5 August 2015.
15. Wilson (16 July 2015). "How 4 Designers Built A Game With 18.4 Quintillion Unique Planets" (<https://www.fastcodesign.com/3048667/how-4-designers-built-a-game-with-184-quintillion-unique-planets>). *Fast Company*. Retrieved 9 August 2015.
16. "About Massive" (<http://www.massivesoftware.com/about.html>). Massive Software. Retrieved 12 June 2016.

Further reading

- Ebert, David S.; Musgrave, F. Kenton; Peachey, Darwyn; Perlin, Ken; Worley, Steve (2002). *Texturing and Modeling: A Procedural Approach* (3rd ed.). Morgan Kaufmann. ISBN 978-1-558-60848-1.
- Shaker, Noor; Togelius, Julian; Nelson, Mark J. (2016). *Procedural Content Generation in Games: A Textbook and an Overview of Current Research* (<http://pcgbook.com/>). Springer. ISBN 978-3-319-42714-0.

External links

- [The Future Of Content \(http://www.gdcvault.com/play/1019981/The-Future-of-Content\)](http://www.gdcvault.com/play/1019981/The-Future-of-Content) – [Will Wright](#) keynote on Spore & procedural generation at the [Game Developers Conference 2005](#)
 - [Procedural Content Generation Wiki \(http://pcg.wikidot.com/\)](http://pcg.wikidot.com/) – a community dedicated to documenting, analyzing, and discussing all forms of procedural content generation.
 - [Procedural Trees and Fire in a Virtual World \(https://github.com/intelzombi/Procedural-Trees-and-Fire-in-a-Virtual-World\)](https://github.com/intelzombi/Procedural-Trees-and-Fire-in-a-Virtual-World) – a white paper on creating procedural trees and procedural fire using the Intel Smoke framework
 - [A Real-Time Procedural Universe \(http://www.gamasutra.com/view/feature/3098/a_realttime_procedural_universe_.php\)](http://www.gamasutra.com/view/feature/3098/a_realttime_procedural_universe_.php) – a tutorial on generating procedural planets in real-time
 - [Search-based procedural content generation: a taxonomy and survey \(http://julian.togelius.com/Togelius2011Searchbased.pdf\)](http://julian.togelius.com/Togelius2011Searchbased.pdf)
-

Retrieved from "https://en.wikipedia.org/w/index.php?title=Procedural_generation&oldid=954075773"

This page was last edited on 30 April 2020, at 15:12 (UTC).

Text is available under the [Creative Commons Attribution-ShareAlike License](#); additional terms may apply. By using this site, you agree to the [Terms of Use](#) and [Privacy Policy](#). Wikipedia® is a registered trademark of the [Wikimedia Foundation, Inc.](#), a non-profit organization.