kriativ.tech

Edição Nº 9 – 25-11-2021
ISSN Print: 1646-9976 | ISSN Online: 2184-223X |
http://www.kriativ-tech.com
DOI: 10.31112/kriativ-tech-2021-10-66

Procedural generation: diversification algorithms for level generation in an RPG video game

Ricardo Baptista
Assistant Professor at ISTEC-Porto/ CITECA –
ricardo.baptista@my.istec.pt

Renato Santos
Computer Degree at ISTEC-Porto - renato.santos.727@my.istec.pt

Abstract: *It is necessary to provide some gameplay diversity to nurture a growing desire to play again when it comes to simple video games. To this end, we can focus on one of the most prominent components, the videogame environment itself.*

Procedural generation algorithms assist both the developer in the development process and the player by providing new experiences each time he plays. However, studying and choosing the correct algorithm for the desired video game style is essential for success since if the algorithm does not fit the intended purpose, the feeling when playing will not be the same.

This article intends to approach some of the videogame genres and present a procedural level generation algorithms study, followed by the analysis of the algorithm implemented in a videogame developed as a personal project.

Keywords: *Procedural generation, videogames, RPG, development, algorithm*

1. Introduction

Currently, technology is present in many lives, and, consequently, videogames are accessible to more and more people, either on mobile devices, such as smartphones or on consoles, personal computers, among others. Such growth inaccessibility is more notorious to the younger generation, with an increasing number of children getting used to the technological environment and the fun present in video games.

Considering the level of gameplay intended to provide the player, the use of procedural generation algorithms optimizes the entire development process, reducing considerably the content amount that must be placed manually.

Back to School is a videogame that fits into the rogue-lite RPG genre, developed by the author of this article with the aid of Godot Engine. This article aims to present conceptualization and taxonomy around the topic of video games, initially, which helps a proper understanding of the latter meticulous investigation conducted. Such investigation aided the development of Back to School and, primarily, the procedural generation algorithm implemented.

2. State of Art

Firstly, it is of the utmost importance to define particular concepts to ensure the proper understanding of this article, addressing the video game concept.

A videogame is, basically, a virtual game, where the player interacts through peripherals, like joysticks, keyboards, mice, connected to a computer or console and with a visual component when in the presence of a monitor ([1]). Videogames can also be seen as art, narrative, educational tool, and more, although a video game is primarily a game (Esposito, 2005).

The existence of games predates video games; for example, board or card games were played physically and in person. However, they now exist in digital format, and, in addition to these classics, we also have the emergence of new genres that would be impossible to obtain if not digitally, genres that will be addressed later.

2.1 Brief videogame history

Human beings always try to answer questions of primacy, and, of course, they have already tried to answer a particular question, "Which is the first video game?". Many assume the famed Pong (1972), from Atari, as the first videogame due to its popularity at the time, although the actual first videogames are already seven decades-old since their first public demonstration, with Bertie the Brain, in 1950 (Bateman, 2014), and Nimrod, in 1951 (Donovan, 2010). These would lead to the interest in videogame development, and two decades later, a multiplayer, multiprogram prototype emerged, "The Brown Box," accomplished by a group of developers led by Ralph Baer at Sanders Associates (Edwards, 2012). This system would later be finished and commercialized by Magnavox in the United States in 1972 and abroad with the Odyssey in the following year. Odyssey would then be considered the first domestic console (Edwards, 2012). Shortly afterward, in 1972, Atari published Pong, initially as an arcade game, which led Magnavox to sue Atari due to its resemblances compared to one of them already existing, 28 games published by Magnavox. However, around the 1975 Christmas season, Atari released a domestic version of Pong, and in 1977, Atari commercialized the Atari 2600, a domestic console with controllers, removable cartridges, and colored games, triggering the second generation of consoles. Nonetheless, there came a rough time for the videogame industry, from the oversaturation of the console market, the competition with computer games, and the large number of games that did not live up to expectations, like the videogame E.T., a video game based on the classic movie of the alien lost on Earth. Atari published this videogame in 1982, and it is still seen as the worst videogame ever, causing organizational changes within Atari.

In Japan, the domestic videogame industry started revitalizing around 1985 by introducing the Nintendo Entertainment System (NES), or Famicom. The NES improved sound, gameplay, and 8-bit graphics - each pixel can store 8 bits of colors data compared to previous consoles. There was also what could be considered the first console war, where the NES and Sega were rivals and battled to see who took first place in console and video game innovation and sales (Greene, 2015).

With the technological evolution, the advances in videogames also became apparent, introducing three-dimensional graphics. From there came the era of the

famous PlayStation, launched in 1995, the Saturn, launched in 1995, and the Nintendo 64, launched in 1996, a relatively short rivalry since Sony, and their PlayStation console, had the clear advantage due to its third-party support.
Currently, we have three main consoles, PlayStation and Xbox, that maintain a healthy rivalry, and the Switch as a portable and more versatile console. Lastly, personal computers also became one of the main ways of playing video games, matched with the consoles referred to previously.

2.2 Videogame Taxonomy

The genre of a video game is a classification associated with it, usually defined by a set of gameplay challenges independent of the narrative or content of the virtual world, such as shooter games do not cease to be it just because of the context. (Adams & Rollings, 2006)
The first attempts to define taxological categories for videogames focused on sorting catalogs and books. A catalog for the Atari 2600, in 1981, used eight titles:

- Skill Gallery
- Space Station
- Classics Corner
- Adventure Territory
- Race Track
- Sports Arena
- Combat Zone
- Learning Center

From there, several methods of genre separation of videogames were approached. However, there is still a discrepancy present between authors such as Wolf (2001), Ap-perley (2006), and Despain (2009). We then assume the following main genres:

- Action
- Adventure
- Fighting
- Puzzle
- Racing
- Roleplaying (RPG)
- Simulation
- Shooter
- Sports
- Strategy

2.2 RPG and Roguelite

There are several video game genres, and a single game can often be assigned more than one of these genres. These genres are often interconnected in a certain way, and it is possible to consider a game as being Action and Adventure or Roleplaying and Strategy, for example.
In the taxonomy of games, Roleplaying is a game classification in which the player assumes a character in a fictional context, predefined by a master who defines and controls all aspects of the level and story (Apperley, 2006). Players take responsibility for playing these roles within the narrative through literal representation or a structured decision-making process with full respect for character development.
Nevertheless, some say that RPG videogames do not demonstrate the true splendor of an RPG board game, since "in most computer RPGs, the computer replaces the role of the Dungeon Master (DM), not only in facilitating the players' actions within the fantasy world but in creating the fantastic environment in which those actions take place." (Apperley, 2014)
One of the relevant aspects of the RPG genre is the level of challenge imposed on the player, which increases interaction, supporting immersion. Mostly, this genre does not assume a direct competitive aspect but feeds the curiosity and desire to

evolve the character where, sometimes, it can have a comparative value between players, through specific forums, for example.

Sometimes, in this type of game, the most exciting factor is the learning and evolution of the character and the player himself, understanding the game better and better, exploring new paths and methods at crucial moments, which can lead to a different outcome.

Roguelike (or roguelike) is a subgenre of RPG characterized by procedurally generated dungeon crawl navigation, turn-based play, grid movement, and permanent character death. Many video games of the roguelike genre reflect on fantasy narratives influenced by the RPG board game, Dungeons & Dragons (Barton, 2007). Around the year 2000, the rogue-lite (or roguelike-like) subgenre appears through independent developers, indies. These developers started to incorporate some roguelike elements in genres that are not generally related to this one, thus forming the basis of this new subgenre.

That is, the rogue-lite type refers to all subgenres that use some aspects of the roguelike subgenre (for example, procedurally generated dungeons), typically accompanied by a more complex look, requiring less strategy and some persistent elements after the character's death.

A video game that sparked the growth of indie roguelites was Spelunky (2008). This one was developed by Derek Yu, aiming to mix the deeper gameplay offered by roguelikes, which allows enjoying different levels whenever to play, with the ease of understanding of platform games. As a result, Spelunky is a platform game that embodies the notion of permadeath (permanent death), where the character navigates randomly generated caves.

In particular, for smaller independent developers, the nature of the procedurally generated levels allows them to deliver quite a few hours of content without spending many resources and time developing detailed worlds. This question also allows developers to devote more time to other complex aspects of the game and its interaction system that are part of the pleasure of playing. McMillen, the developer of The Binding of Isaac, states that incorporating roguelike elements into other video game genres can be difficult, as they have to overcome complex interfaces that roguelikes tend to have. However, when a method of doing so is found, "it becomes an increasingly beautiful, deep, and everlasting design that allows generating a seemingly dynamic experience for players, so that each time they play your game, they are getting a new adventure." (McMillen, 2012)

3. Procedural Level Generation

When we talk about levels in video games and their generation, we refer to the player's scenario. These scenarios can be presented in various ways, 2D or 3D, more realistic or less realistic, linear or open-world, predefined or procedural, and many more.

In this chapter, the most important is procedural generation, that is, a type of scenario generation where a newly completed scenario will never be the same as previous scenarios. The arrangement of certain elements, such as rooms or objects, or even the textures of the scenario and other game constituents are mostly randomly generated, with only a few predefined parameterizations.

This type of implementation allows developers not to worry so much about the design, limiting the number of constituents to be placed manually, directing developers' focus to other goals.

Procedural generation algorithms can be presented in several ways and can obtain different results with minimal change. For the development of Back to School, several algorithms and solutions were explored, looking for the most relevant result. That said, the following algorithms were explored:

- Random Room Placement
- Árvore BSP
- Cellular Automata

3.1 Random Room Placement

Being the most straightforward technique that has been studied, it is used, in some way, in several classic roguelikes. This technique boils down to creating a rectangle – the room – in a position and with random size, and each room, if positioned successfully, must be added to a list. If it creates a room intersecting with one that has already been placed, that room is discarded, and it happens a next algorithm iteration. This process repeats until the desired number of rooms is reached, or it is impossible to create more rooms.

Once we have all the rooms created, we should try to connect them with corridors. What must be ensured is that all rooms have at least one link, and we can solve this, for example, with an algorithm called Random Point Connect (RPC), where we iterate through the entire list where the rooms were stored and link one with a particular index to the one with the index directly after. Although this may be a straightforward solution, it makes the traversable path linear, which may not be the intended result of the developer.

Picture 1 - Visual representation of how the RRP algorithm draws rooms.

3.2 BSP Tree

Binary Space Partitioning (BSP) is the second studied method used to subdivide a space recursively until reaching the desired limit of divisions or there is no more space for divisions. Each subdivision results in two rectangles, the 'right child' and the 'left child,' and the divisions can be done horizontally or vertically, but with a minimum size depending on the minimum size of the rooms that can be created. The starting rectangle is the root, and all subdivisions are saved as branches, hence the tree structure. At the algorithm ends, rooms will create in the last tree structure subdivisions, always limiting the size of the rooms according to the size of the subdivision itself. The connection between rooms must, once again, be guaranteed, with each room having at least one connection.

Picture 2 - Visual representation of how the BSP algorithm divides the space and draws the rooms (Esquerda, 2013)

3.2 Cellular Automata

Lastly, *Cellular Automata* was first conceptualized by John von Neumann and Stanislaw Ulam in the 1940s (Schiff, 2011), whose concept, after two decades of study, gave rise to the Game of Life in the late 1960s. John Conway, a British mathematician, invented this machine that operates a two-dimensional array of square cells (Adamatzky, 2010). Each cell has a specific state, and a set of rules is used to define its state, depending on the neighboring cells.

Differentiating from the methods previously presented, Cellular Automata, applied to the generation of levels, is a way to generate levels that resemble caves, with their deformities and an appearance close to natural. For this, the developer must apply a set of rules, whether new or previously tested, and with that, we can expect a trend of organic structures, and the isolated cells tend to disappear after a few steps.

Once the set of rules is decided, it is necessary to randomly populate the space with cells that represent the walkable space, and it is anticipated that a map covered, between 40% and 50%, with those cells will give the best results. A Cellular Automata algorithm was then studied, based on what one of the game's developers Helms of Fury, made publicly available, where a coverage value of 48% over 20,000 used steps. Finally, we must check connectivity, as isolated rooms are not uncommon. To do this, we can use a Flood Fill algorithm to map all the individual caverns and then use the Drunkard's Walk algorithm (Monaghan, 2019) to connect these caverns non-linearly. (Nazgum, 2019)

Picture 3 - Example of a cave obtained from the use of the Cellular Automata algorithm

4. Application

Back to School, a videogame whose prototype was developed as part of the graduation project, implements a procedural generation algorithm as one of its goals, hence the previous algorithms mentioned above. The final decision for the procedural level generation relapsed over an algorithm inspired by RRP and BSP Tree, thus developing a hybrid that fits the initial vision.

This algorithm is on a two-dimensional array, forming a 4x4 grid where each grid index has a randomly assigned value to maintain the unpredictability of the world in which the player places and thus make each exploration of the level a different adventure, with associated risks.

Picture 4 - Visual representation of the algorithm used in Back to School

The _ready() function, a function native to the engine executed first when the class is called, defines the main processes. It starts with the assignment of the Tileset that should be used. The art of each level should constitute another level of diversity.

Then, each room identities are defined according to the explanation given above. These rooms are then drawn with elements such as floors, walls, and doors placed using coordinates.

After finishing the room's design, the player places it in the starting room, and playing begins.

Currently, enemy rooms do not have random placement of opponents, but its implementation is one of the primary goals in one of the following iterations of the videogame, providing the player with more levels of diversity in order to increase gameplay.

5. Conclusion

Procedural generation algorithms are a great way to introduce diversity and character into a video game, increasing unpredictability for the player and maintaining a considerable level of enjoyment in the act of playing.

On the other hand, procedural generation can be more complex to implement, but it facilitates the development process, allowing the developer to focus on other aspects of due importance in the project, such as assets and design.

This work corresponded to a set of objectives for the analysis of procedural generation algorithms for game levels. From this analysis, it was possible to identify the best contributions of each one and proceed to the implementation of scenarios with a hybrid algorithm in a prototype of an RPG videogame.

6. Bibliography

Adamatzky, A. (2010). *Game of Life Cellular Automata*. Springer.

Adams, E., & Rollings, A. (2006). *Fundamentals of game design*. New Riders. p. 67

Apperley, T. H. (2006). Genre and game studies: Toward a critical approach to video game genres. *Simulation & Gaming*, 37(1), 6–23. https://doi.org/10.1177/1046878105282278

Atari 2600 VCS catalog - Atari (USA) - 1981 - English. (n.d.). Www.atarimania.com. Retrieved March 22, 2021, from http://www.atarimania.com/catalog-atari-atari-usa-co16725-rev-d_3_2.html

Baptista, R. J. V. (2017). Jogos Sérios para Treino e Certificação de Competências. *Repositorio-Aberto.up.pt*. https://repositorio-aberto.up.pt/handle/10216/110820

Barton, M. (2007, February 23). *The History of Computer Role-Playing Games Part 1: The Early Years (1980-1983)*. Www.gamasutra.com. https://www.gamasutra.com/view/feature/3623/the_history_of_computer_.php

Barton, M., & Stacks, S. (2019). *Dungeons and desktops: the history of computer role-playing games*. Crc Press.

Baron, J. R. (2017). Procedural Dungeon Generation Analysis and Adaptation. *In Proceedings of the Southeast Conference (ACM SE '17)*. ACM, New York, NY, USA, 168–171. DOI: https://doi.org/10.1145/3077286.3077566

Bateman, C. (2014, August 13). Meet Bertie the Brain, the world's first arcade game, built in Toronto - Spacing Toronto. *Spacing Toronto*. http://spacing.ca/toronto/2014/08/13/meet-bertie-brain-worlds-first-arcade-game-built-toronto/

Donovan, T. (2010). *Replay: The History of Video Games*. Yellow Ant.

Dungeon generation using BSP trees | eskerda.com. (2013, December 22). Eskerda. https://eskerda.com/bsp-dungeon-generation/

Edwards, B. (2012, May 27). *Inside the Magnavox Odyssey, the First Video Game Console*. PCWorld. https://www.pcworld.com/article/464739/inside_the_magnavox_odyssey_the_first_video_game_console.html

Esposito, N. (2005). A short and simple definition of what a videogame is. *In Proceedings of the 2005 DiGRA International Conference: Changing Views: Worlds in Play (DiGRA'05)*, (3). http://www.digra.org/digital-library/publications/a-short-and-simple-definition-of-what-a-videogame-is/

Esquerda, L. (2013). *Dungeon generation using BSP trees.* https://eskerda.com/bsp-dungeon-generation/

Greene, G. (2015, November 28). The art and legacy of the ’90s console war. *Venture Beat*. https://venturebeat.com/2015/11/28/the-art-and-legacy-of-the-90s-console-war/

History.com Editors. (2019, June 10). *Video Game History*. HISTORY; A&E Television Networks. https://www.history.com/topics/inventions/history-of-video-games

Jogo eletrônico. (2021). Britannica.com.br. https://escola.britannica.com.br/artigo/jogo-eletr%C3%B4nico/481214

Matthews, V. (2018, April 12). The Many Different Types of Video Games & Their Subgenres [Review of *The Many Different Types of Video Games & Their Subgenres*]. https://www.idtech.com/blog/different-types-of-video-game-genres

McMillen, E. (2012, November 28). *Postmortem: McMillen and Himsl’s The Binding of Isaac*. Game Developer. https://www.gamedeveloper.com/business/postmortem-mcmillen-and-himsl-s-i-the-binding-of-isaac-i-

Nazgum. (2019). *Procedural generation with Godot: Creating caves with Cellular Automata*. Abitawake.com. https://abitawake.com/news/articles/procedural-generation-with-godot-creating-caves-with-cellular-automata

Nintendo. (1987). The Official Nintendo Player’s Guide (1987). *In Internet Archive.* https://archive.org/details/The_Official_Nintendo_Players_Guide_1987/page/n151/mode/2up

Pearson, D. (2013, December 31). Where I’m @: A Brief Look At The Resurgence of Roguelikes [Review of *Where I’m @: A Brief Look At The Resurgence of Roguelikes*]. GamesIndustry.biz. https://www.gamesindustry.biz/articles/2013-01-30-where-im-a-brief-look-at-the-resurgence-of-roguelikes

Powell, K. (2019, March 10). What electronic games can teach us. Knowable Magazine. https://knowablemagazine.org/article/mind/2019/video-games-educational-benefits

Schiff, J. L. (2011). Cellular automata: A discrete view of the world. (Vol.45). John Wiley & Sons.

Valtchanov, V. & Brown, J. A. (2012). *Evolving Dungeon Crawler Level With Relative Placement. In Proceedings of the Fifth International C* Conference on Computer Science and Software Engineering (C3S2E '12)*. Association for Computing Machinery, New York, NY, USA, 27–35. DOI: https://doi.org/10.1145/2347583.2347587

van der Linden, R., Lopes, R., & Bidarra, R. (2014). Procedural Generation of Dungeons. *IEEE Transactions on Computational Intelligence and AI in Games*, 6(1), 78–89. https://doi.org/10.1109/tciaig.2013.2290371

Yee, N. (2006). The Demographics, Motivations, and Derived Experiences of Users of Massively Multi-User Online Graphical Environments. *Presence: Teleoperators and Virtual Environments* 2006; 15 (3), 309–329. DOI: https://doi.org/10.1162/pres.15.3.309