The Usage of PCG Techniques Within Different Game Genres

Martin Dahrén

Computer Science
Bachelor's Thesis
15 ECTS
Spring 2021
Supervisor: Steve Dahlskog

Abstract

Procedural Content Generation (PCG) has become more common in usage in game development nowadays, with the motivation of finding new ways to make games replayable, new ways for games to be played or for generating content during development. This paper explores the question how often Procedural Generation is used in practice and furthermore how often it is used by different game genres and how they use PCG in their particular games. This paper will try to answer these questions through both an industry review, discovering which games have used Procedural Generation and also through a literature study to find out what kind of research has been done within the area of PCG and how Game Developers could utilize that in the future. The findings were that even if the usage of PCG differentiated between genres, certain areas like Level Generation and entity instancating were more commonly using Procedural Generation compared with others such as Puzzle generation, Plot generation and Dynamic Systems. The literature study gave a perspective that there are plenty of research done within PCG on how to create new, different and unique ways to generate content, but it is usually in forms of prototypes and not ready to be used in games yet.

This gives the conclusion that game genres use Procedural Generation to maximize the user experience with what the game wants out from that genre or use it to make game development more efficient. However, certain genres such as Adventure-games and Role-playing-games could benefit from having PCG for parts of the games where it is not used today which means there is still room for potentially using Procedural Generation. But with that also comes a discussion about what areas of PCG can be improved to meet the needs of the developers and make them more willing to use PCG on areas where it is not currently used.

Table of Content

1.Introduction

Procedural Content Generation (PCG) refers to the creation of game content automatically using algorithms [7]. Due to games increasing in size, budget and scope, it has come in use for games more often in the recent years. The use of algorithms to generate gameplay, assets and other types of content for a game are appealing for developers that have huge games to create. PCG algorithms can be used to create new ways to play games, create new ways to create and develop games and offer more replayability [167]. With different types of content like AI, Levels, Quests, Models and other types of content that exist in games, it would naturally lead to different types of algorithms for creating such content.

This brings up questions on the other hand on how PCG is being used in practice? With so much usage within the game industry, there is a need for a clear view on when it is used, how often and what types of content that is being procedurally generated. What would make the picture even clearer is to explore how the PCG usage differs between the games genres. Finding out how these techniques are used and what techniques exist would also give a good view on what techniques are beneficial to use for a specific genre, while finding out if they are too cumbersome to use and thereby wasting time for other types of genres. Game genres have different goals when it comes to delivering their content and the study would help find out if games use PCG to either give unique approaches to these goals or if they help with development of these genres and therefore find out if PCG is being used to its fullest potential within Game Development.
The following research questions that the study is going to focus on are:

- How are different PCG techniques used in practice?
- How often are different PCG techniques used?
- How is the usage of different PCG techniques different, comparing game genres?
- Are there areas in games that have hidden potential if using PCG?

Surveys like this for PCG have been done in similar fashion. Surveys usually go through what type of content PCG could create in games and in what categories PCG can be used [2][8]. Related research seem also to focus solely on one aspect of PCG like Terrain [3], City generation [4] or Puzzle generation [5]. This study focuses on how different genres use PCG in practice, creating a good overview on how to use PCG techniques in theory and in practice with different genres.

This study will follow an analytical approach, with a literature study for collecting information about PCG and PCG techniques and an industry review of the usage of PCG techniques in different games. This dual approach was chosen to find out the correlation between PCG in what is being researched and how PCG is used in practice.

The contribution of this paper is to demonstrate what PCG can deliver, how it is used in practice today and to assess if there are further potential within PCG for the game industry in the future. Combined with data about different PCG usage can then be used to find out

what techniques can be of benefit for the genres that are not using them today. Finally it would contribute on what techniques should be researched further in PCG field so that these techniques can become easier or more appealing to be used for game development.

1.1 Previous Studies

Studies related to PCG have been done on different types of areas. Areas of study can either be an area of content, technique or general usage of PCG.
General studies include how PCG can be used for different parts of a game [2], where it explored what type of content exists in a game, how they can be generated with different techniques and what type of research should be done in the future.

Other surveys within PCG go more indepth on certain types of content or techniques. It could be very specifique and niched, such as how to generate a particular feature or going into certain PCG techniques.

As an example, one general study goes through techniques suitable for game development and group these according to the final content that they produce and where also classed whether they were assisted or non assisted techniques [9]. The study found out that assisted methods can help decrease development time for designers, since it can help create content in a detailed fashion due to guided generation, while non assisted methods were more suited for as a starting point for further design and for game content that is generated on the fly.

A study about different techniques for generating procedural dungeons found that a lot of techniques for generation of dungeons are search based [10].
Procedural Puzzle generation was also surveyed, when they studied different types of puzzles and stories and tried to find out on how to procedurally generate them [5].
A survey about City generation was covered and surveyed different techniques on how cities can be generated through PCG and evaluated them based on criterias like scale, realism, efficiency, control etc [11]. The result was that a lot of existing
surveys within PCG also goes through what usage certain techniques can have or how creative the usage is within PCG.

Search based PCG was also researched on how it can be applied to different types of content and how the computational creativity is with Procedural Generation [12].
The computational creativity survey found that usage of it within PCG is still in its early stage and present what kind of research can be done to use creativity models studied in computational creativity in PCG [13].

1.2 Procedural Content Generation Techniques

Techniques within PCG are divided into two different methodologies, Teleological (bottom up approach) and Ontogenetic (top down approach). Teleological are algorithms that create an accurate physical model of the environment and the generation process, then simply run the simulation and the result emerges in a similar way as in the nature [107].

Teleological algorithms are usually used for offline applications and even if the algorithms are an approximation, it produces emergent or unusual results. Types of algorithms within this methodology are the likes of Genetic Algorithms, Cellular Automata, Markov Chains, Raindrop Algorithms and Fractal Algorithms.
Ontogenetic Algorithms observes the end result of the Teleological Algorithms and attempts to reproduce those results with ad hoc algorithms [106]. Instead of simulation, it tries to get the end result without emulating the intermediate steps.
This methodology is usually more common in online/real time applications kind of games. Types of algorithms that use this methodology are the likes of L systems, different types of Fractal Algorithms like Perlin Noise, Simplex Noise, Diamond Square Algorithm, Midpoint Displacement Algorithm and algorithms for generating cities, mazes and dungeons.

Search based Procedural Generation is a type of PCG that does a special type of generate and test approach by grading the candidate content with numbers called a fitness function and then basing that fitness to produce more content with higher fitness [115]. The fitness is used to determine the suitability of the content to be used in a game and is decided by the developer creating the generator.

Procedural Generation has not only different techniques, algorithms and methodologies for generating content, but it also has different approaches on how to apply them. Examples are Online vs. Offline, Deterministic vs. Stochastic, Generic vs. Adaptive and Mixed initiative vs. Automatic [108, 117].
Online Generation is when a world is generated continuously so that the player never reaches the end. Offline generates content to a specific size and will not make it more of it if the player comes to the end of that content.
A Deterministic Generation generates the same content if the seed is the same, while a Stochastic Generation does not guarantee consistency [108, 117].
Adaptive Generation will take into account of the actions and choices the player makes for its generation, while generic is done without any input from the player [108, 117].
Mixed Initiative is that the user is asked for input and has influence over how the algorithm should perform, where as Automatic Algorithm generates content without any input and generates without any interruptions. An example of Mixed Initiative is when a developer generates levels for a game, the algorithm and the developer takes turns in generating new generations of the level until the developer is satisfied with the result [108, 117].

2.Method

To be able to demonstrate PCG usage among game genres in both theory and in practice, two research methods were chosen; a literature review and an industry review.
The reasons why these two were used was because a literature review gives us information on what PCG is, what kind of research that exists in this area and what areas in games is most commonly research on. This will provide insight on how much research has been done in the different areas of PCG and game areas, and what can potentially be done with it. The literature review will be presented first as it gives us a good overview of the area, but also gives us a better understanding of what kind of guidelines that should be set up for the industry review.
The industry review was used as a follow up, to demonstrate how games that exists in the market use PCG and allow more indepth examples on how and why games that exists in the market use it.
The combined data from both the industry review and the literature review will then be used to draw conclusions.

2.1 Literature Review

A literature review is a way to critically analyze a segment of a published body of knowledge through summary classification and comparison of prior research studies, review of literature and theoretical articles. It is a good way to synthesize evidence and also get a good understanding of the status of a research area [6].
The literature review follows guidelines chosen from the Software Engineering Group from Keele University and the Department of Computer Science from University of Durham [1], where the guidelines are fit to be used for Software Engineering. A literature review is divided into the following steps:

1. Formulating the problem (Planning)
   This step establish the planning of a literature review by developing a research plan, the research questions, inclusion and exclusion criterias and a way for handling collected sources.
2. Data collection
   Where the search for different kinds of sources are made. This is done by generating a search strategy, identify publication bias, prepare reference management and documenting the search.
3. Data evaluation
   The collected data is then valuated if it is relevant and valuable for the study. This is done trough study selection criterias, study selection process and abstract screening.

4. Data analysis
   Assess if the data collected is of good quality by checking the hierarchy of evidence and the quality instruments which are a checklist of factors that need to be evaluated.

The Research Plan was established by first defining what game genres exists and what kind of areas that exists in games that can be generated by PCG.
The research questions are already defined at the introduction of the paper and needs no further explanation.
The inclusion criterias was primarily that the data would come for studies about different ways (new or existing) that PCG can be used, should focus only on PCG usage within games, have been published during or after 2010 and language must be in English.
Exclusion criterias would be any studies of PCG that does not cover any area of game development, been published before 2010 and was in a language other than English.
The last part of the planning was to define how handle the collected sources and that was through adding a spreadsheet with definitions of title, keywords, abstract, when it was published and what game genre and game content the study covered. There is also a description of its quality and if it fits the study or not.

After the planning was done, the search for sources began and it was done following a search strategy. The search strategy was defined by searching on databases that was centered around technology papers like IEEE and ACM, due to the subject being heavily focused around technology. The search phrases and keywords that where used were: *PCG*, *Procedural Generation, Procedural Content Generation, Search based Procedural Generation, Machine learning,* any defined game genre like: A*ction Game, RPG, Role playing Game, Strategy Game, Simulation Game, Adventure Game* and any defined game area like: P*uzzle generation, Story generation, Level Generation, Terrain Generation, AI, Dynamic Systems* etc.
Each search is also documented with how many papers per search, the database, date and limits.

First, the sources are evaluated based on if the author of the paper is well known within the community of PCG. Then the papers are read through by their abstract then their conclusion, result, analysis and then discussion. The sources are valued based on how relevant they are to the study, how useful they are, how good the quality of the study is and what the strength and weaknesses are.

Lastly, the result are presented in the result section of the paper.

2.2 Industry Review

To have a better understanding on how PCG techniques are being used in practice and to get additional information from a more indepth source, an industry review was conducted after the literature review.

This works in a similar way as the literature review, but instead focusing on the actual usage within games that exists in the market. This was done as a parallel to the literature review to find out why do developers use PCG for their games and how do the usage of PCG differentiate between genres.
The industry review follows the same guidelines as the literature review. The guidelines are the same with planning, data collection, data evaluation and data analysis

The research plan was established by defining the game genres and what kind of areas that exists in games that can be generated by PCG, just like the literature review.
The research questions are already defined at the introduction of the paper and needs no further explanation.
The inclusion criterias was primarily that the game have to use some form of Procedural Generation, either under production or during gameplay, is a PC or a Console Game and the language of the source must be in English.
The limit was that the games did not involve offline or online player created content (unless the creators themselves use PCG in some form) [7]. It must be content that is generated based on algorithms or rules that the developers specify. It is also limited to games that are out in the market and are sold on PC/Consoles. The limitation was set to focus on games that big developers have developed for the game market and how PCG is used in the general game market. This limitation would exclude browser games, mobile games and other types of games that is not available from a major seller. An exeception to this limitation would be if a game uses Procedural Generation in an innovative way that has made the game garner attention from media. Last limition is that the sources can not be in any other language than English.
The last part of the planning was to define how handle the collected sources and that was through adding a spreadsheet and document the games and the collected data by their name, genre, PCG area type, release date, a description on how the PCG is used and the sources that are used to support the inclusion. This gives a good overview on what areas uses PCG and how they are spread among the different genres.

The next step was to prepare the strategy to retrive information about the PCG usage in these different games. The primary way of retrieving data was to get it from developer sources. Conference recordings, interviews, articles or blog posts that describe the games' PCG function or usage was always the best way to get high quality data, due to the source coming directly from the people who have developed the systems for the games.
Secondary sources that do not originate/cite the developers of the games were chosen afterwards. Secondary sources were only used when either there were no sources from the developers or a need to strengthen the data for a particular game. This could also be from articles, videos, archives, blog posts etc. that would go indepth about how a PCG system in a game works or how it is used. The data from secondary sources had to be more carefully analysed due to the varying quality.

The analysis is going through the data covering what part of games and genres where PCG is being used the most, what each genre uses PCG for and why certain areas of PCG is not that common in certain genres or at all in games.
In this case we cover why PCG is used more in certain genres, how certain genres make use of PCG and what parts of the games are more common to be used with PCG. This is compared with the result from the literature review to discover where there are potential for new uses of PCG in both genres and game areas.
Important things to think about during the analysis is to follow the research questions that led to the industry review and identify and test any rival explanations to avoid bias.

The result is also covered under the result part of the paper.

3.Result

The result starts with the literature study and its findings. Next section covers the industry review which is divided into each genre that are covered to describe the finding in finer details. Source tables will be attached as separate appendix.

3.1 Literature Study Result

After the search with the mentioned criterias, the result was divided into what kind of area in games and what category of PCG they cover. When it comes to the literature study, multiple papers where found to indicate what PCG techniques can be used, for different types of areas in games.
There were plenty of papers showcasing different types of techniques and algorithms where the game industry is already using Procedural Generation. They also showcased new ways to generate this type of content.

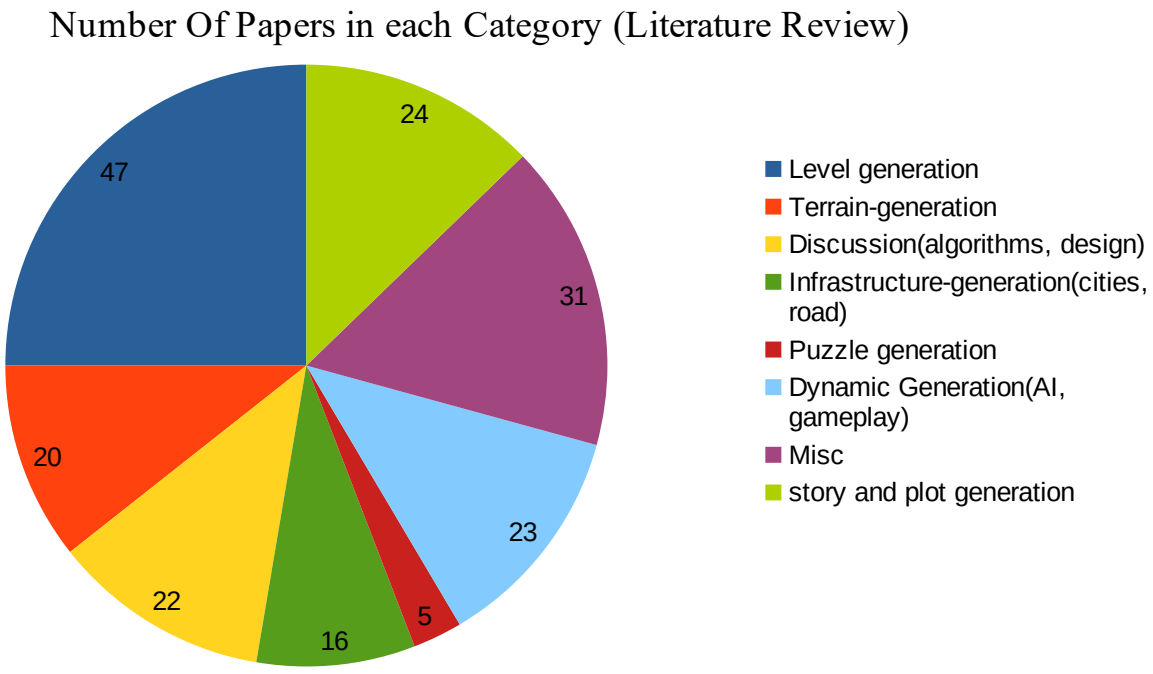

Table 1

3.1.1 Terrain Generation

Terrain Generation refers to generating virtual terrain that makes up the game world. It can be a used as background for games or be the virtual world that player often traverses. Terrain is one of the most common area to be used with Procedural Generation and can be used for every genre, the most common being in Action Games, RPG Games and in Strategy Games.
Despite being so common in game development, not a lot of papers about Terrain Generation was yielded from the literature review with 20 papers covering this area.

3.1.1.1 Noise-based Approaches

The most traditional way of generating terrain is through a noise based approach to make a heightmap, with methods such as Diamond square Algorithm, Simplex Noise, Perlin Noise [15][16].
The problems noise based approaches have, is that they are difficult to influence and are more suitable for only generating a basic terrain that can only be generated by the height. There where a lot of papers that showcases techniques to combat these limitations.

3.1.1.2 Advanced Approaches

One of them was about using Software Agents to make the generation more controllable and can give designers more parameters to design the terrain [17]. Another technique showcases on how to generate volumetric terrains with layered materials and is customizable [168].
One paper showcased semantic constraints to help designers express their intent and deliver the wanted outcome from the terrain [159]. Another method that can be of benefit for designers is by using interactive procedural sketching so they can quickly layout a world while still having control over the terrain features [14].

3.1.1.3 Realistic Approaches

More realistic approaches can use a hydrology based method to generate terrain by creating rivers and then making land from these rivers, creating terrain that looks more like something from the real world where the location of mountains are based on the placements of rivers [21].
Another realistic approach is to use satellite images, where images from real world satellietes can be used to train algorithms to produce more realistic terrain [22] or use it to show how to apply textures to the terrain [23].

3.1.1.4 Story Driven Generation

A story driven game can generate maps by a generator that builds continents based on user specification and then build structures like towns and unique terrains in a sequence until it reaches a complete world solution. This can create replayability to story driven games, but still adhere to the the required story [18]. Another approach for story based map generation is to use story space as a collection of plot points and use the relations between the plot points and locations to generate stories and map configurations for these generated stories [173].

3.1.1.5 Remaining Uses

There are also other uses for terrain techniques other than just terrain. Examples from the review include caves that can be generated with noise functions and marching cube algorithms [19], cellular automata [161] or with volumatric terrain based on voronoi diagrams and delaunay triangulations [162]. Planets could also be generated with procedural noise [20].
Techniques can also be combined with each other. Noise techniques can not generate structures like arches, overhangs or caves but with grammar and particle based system they can be added after the noise has generated the terrain [24].

The papers gathered showcased that Terrain Generation research focused on getting more control on what is being generated and covering areas that other techniques can not do, like generating archways or more relaistic terrain.

3.1.2 City, Building and Road Generation

City Generation, Building Generation and Road Generation refers to generating objects like buildings, roads and infrastructure for cities in games. These can appear in any game genre where a city takes place or needs road networks. The review yielded 16 papers from this area.

3.1.2.1 Cities and Roads

City Generation and Road Generation are usually linked and worked at the same time due to cities having roads as feature awhich is discussed in one paper where an agent based algorithm was used to generate cities (buildings, roads, water etc) and output evolved road networks based on parameters [25]. Other ways to create cities is through wave function collapse algorithm [32], using satelliete imagery and global scale data to generate a three dimensional approximation of an urban area [33] and vectorizing gridded urban simulation data to make it usable games [34]
Other ways to generate roads and cities is through L systems [26], but also through shorthest path algorithm, minimum spanning trees, templates and tensor fields [103]. Road generation by themselves uses pattern extraction from real world road layout with patch based approach [104] or with space colonisation [27].

3.1.2.2 Buildings

Building Generation are generating buildings that can be entered and interacted with. Papers in this area talks about generating buildings with novel shape grammar [28], making buildings with planar shapes and two dimensional boolean that can then be manipiulated by dragging objects around [29] and an agent based model for facades and interiors of buildings using two dimsional floor plan and semantically defined architectural styles as input [30]. There is one paper that uses multiple techniques and algorithms to make a fully textured 3D model of a house [31]. Lastly, a paper developed an algorithm called The Specific Room Placement (SRP) to generate a layout buildings with interiors based on room type(public, private) and their connection [168].

The papers go trough various techniques that can be used for generating cities, roads and buildings. Some papers are about automatic generation, some for making realistic depictions and some are for aiding the designer in the creation. This shows that this type of generation will need multiple PCG types to be able to deploy in games in a more realistic and easy to use kind of way for the game developers.

3.1.3 Level Generation

Level Generation is the generation of the areas the player can traverse through. Levels exists for every game genre, but most common is in Action Games (platformers, shooters, arenas),RPGs (dungeons) and Strategy Games (strategy maps).
Level Generation were the most frequent subject covered by the the research papers discussing PCG with 43 papers of the 187 found.

The papers found, showcases different algorithms and techniques used for generating levels in games.
Wave function collapse is a constraint based algorithm that is based on that a quantum particle can exist in multi state when unobserved. As soon as it is observed, the possibilites disappear and the function collapses. It works by extracting local patterns from input images, process the patterns into index to speed up constraint checking, incrementally generates the output image by growing a partial assignment and finally renders the total assignment back into an image in the same format as the input [35]. This method has quickly becoming popular in the PCG world and papers on how to augment the method has appeared, describing phenomen like using growing grid neural networks for map generation [36].

Another way to create Level Generation is Answer Set Programming. It is a declarative programming method that can be used to describe what problem is to be solved instead how it is to be solved. This method can be combined with graph grammar [43] or with video game description language [44] to generate dungeon like levels.

Other types of algorithms include Graph based approaches like Graph grammar or Generative Grammar which can be used to generate 2D levels [45], dungeons [43] and action adventure levels that fit with the objectives and missions of the game [46].
Search based Procedural Generation is a generate and test technique that searches and optimize content based on previously generated content. Papers that have been written about search based techniques have done it to generate strategy maps [47], generate mazes based on user properties [48], generating race tracks based the interactivity of the users [169] and generating levels for shooter games with generated classes [49].

Other types of techniques that can be used to generate levels are linear constraints together with Satisfiability Modulo Theories (SMT) Solvers [50], Machine Learning [51, 52], Cellular Automata [53], Deluged Novelty Search Local Competition algorithm [166] and

how Random Room Placement, BSP Rooms, Random Point Connect and Drunkard's Walk algorithms can be used together [55].

Some papers have researched on how using PCG could aid the designer in creating levels with various techniques. Mixed initiative can be used by using genetic algorithms to help game designers create dungeons based on their ideas [37,38]. Other Mixed initiative approaches are using a model transformation approach to partly automating the process of level design [39]. Other examples of procedural tools is a design centric maze generation tool that creates mazes based on inputs from the designer on what type of mazes they want [42] and algorithm that generates dungeons that can change size, difficulty and content of the dungeons while generating a playable level [54].

Tower Defense Games have a unique method of being generated with that they focus on generating the paths, monster sequence and tower generation. The way to generate these can be different like using Monte Carlo search to test the levels are playable and properly designed [40] or search based algorithms to generate roads [41].
One paper goes through on how to generate balanced strategy maps with different kinds of algorithms like Graph Grammar for layout, Terrain Generation with a combination of algorithms and placing out strategic features with Evolutionary Search Algorithm [165].

The papers show in both the sheer number of papers and the wide variety of ways that levels can be generated in, that Level Generation is the most common way to generate content for games. The papers showcase that Level Generation can mean anything from shooter levels, platform levels, strategy maps, mazes, dungeons and tower defense maps.

3.1.4 Story and Narrative Generation

Story Generation is about generating stories for games to make them more replayable or create stories based on changing circumstances. Stories can be featured in every genre, but they are more common in RPG and adventure games. The review came up with 24 papers about story generation, which is a lot in this area compared to how often it is used in games.

3.1.4.1 Story and Narrative

When it comes to story generation there where a lot of papers on how to generate story and narrative in a procedural way. Examples include generating narrative in three steps, context free grammars, planning and simulation [56], language models based on Neural Networks [57] and dynamic story based on both Handcrafted Event Network and Dynamic Artificial Social Network [58]. These three approaches generates stories based on what characters and relations that has been generated and makes these generator dynamic and scalable. Mystery based story games that the players have to solve can be generated by Monte Carlo Tree Search systems [70]. Another paper have researched on how using different

techniques under certain elements like character generation and social simulation can be used to support the storytelling [71].
Other types of use for story based Procedural Generation are procedurally generated natural language retellings of playtroughs from play trace data [109], using context free grammar to make game character generate discourse and decisions [72] and procedurally generate satellite sentences to make interactive stories have choices while still being authorly sound [73].
Other techniques include using grammatical evolution to improve grammar based stories to find multiple solutions [59] and using constrained story generation with a two layered architecture where one layer works with human authored probabilistic graphs model scenes and the other layer determined the sequence of scenes to be generated [108].
Techniques can also be used to aid game designers to create stories like Mixed Initiative tools to help generalize and enhance grammar based stories [60] and using layered hierarchical graphs to help writers describing the plot of a game [61].

3.1.4.2 Languages

Languages can be generated with a simulation based approach to generate fictional game language that characters in game can speak [62]. Fictional folklore in games can also be generated with Grammar directed Algorithm [63]. Even history of a game world can be generated with state machines and replacement grammar [69].

3.1.4.3 Quests

Quests that exist in games can also be generated. One paper goes through on how to generate quest based on the structure from different games [64], while another research on how to generate quests that can be solved in a multitude of ways [65].

3.1.4.4 Personalities

NPCS (non playable characters) can be generated into game world with unique backgrounds, personality and relations based on how and where in the world they are generated. Approaches to this method includes Mixed Initiative [66], Answer Set Programming [67] and Interactive Storytelling Models [68].

Story and narrative generation can be generated with a lot of techniques and cover various different areas of games. The research covered everything from story, quests, narratives, dialogues, game characters, history and languages.

3.1.5 Puzzle Generation

Puzzles are defined as logical and conceptual challenges that require problem solving skills rather than reflex skills or hand eye coordination. They can either be included as part of a game experience or be the entire gameplay loop. They can be featured in most genres of games. The most common being Adventure Games and the most uncommon one being in Strategy Games. The review produced very few studies about PCG in puzzles with only five papers about it.

One of the papers talks about bringing Puzzle Generation with a narrative aspect to it for Adventure Games [112]. The main goal was to generate narrative puzzles to make Adventure Games replayable, as they are story driven games where you solve puzzles by exploring the space and objects and how to manipulate them. They solved this by making a tool that designers could use to specify characters and objects with attributes and let the tool generate puzzles with these objects.
Another study focused on using a Machine learning technique called Parameterized Generative Adversarial Networks to generate puzzle for puzzle games [113].
The last few papers talks about generating Physics based Puzzles. One discusses how using Machine Learning to reduce the number of simulations that needs to be performed by an evolutionary approach [114]. Another talks about generating physics based puzzles with Estimation of Distribution Algorithms (EDA) which evaluates probability tables with a simulation based fitness function [171]. The problem with PCG involving physics is that it requires simulation of the puzzle to ensure feasibility and stability of the puzzle. This approach is computationally expansive and the paper employs machine learning to reduce it successfully.

A common theme with Puzzle Generation seems to be that it relies on more advanced algorithms like Evolutionay Algorithms and Machine learning. This is probably due to that puzzles are a more complicated and involved parts to generate for a game that has a lot of factors involved that can break the generation if not taken into account, therefore requirng more complicated algorithms.

3.1.6 Dynamic Generation

Dynamic Generation refers to Procedural Generation to create Dynamic Systems and agent behaviours like weather, AI, gameplay and group and crowd behaviour.
These types of systems can be applied to every genre also, but occurs more often in Action Games, RPG and in Simulation Games.
The literature review covers 23 papers about the subject.

3.1.6.1 Gameplay Adaption

Gameplay adaption was the most common of these papers. These papers covers how PCG can be used to balance the gameplay, considering the player's skill level and gameplay approach. This approach of personalization of content has shown to increase player engagement [89]. Player modeling is an approach to generate content based on what the player is doing with the game content. One example shown to be used is an infinite runner to generate models based on experienced and new players [87]. Missions in a game can be generated based on player skill with a Player Modeling Approach [86] and even entire games can be generated with this approach [88]. Even Experience Driven Procedural Content Generation (EDPCG) can with help of Learning Based PCG know preferences from developers and play testers during the games development and generate content that fits the players preference [77]. One paper wrote about a prototype for using Semantic

Gameplay Descriptions to generate complex and immersive game worlds for the player [170].
Reinforcement Learning, which is a category of Machine Learning that makes agents learn the way to act in a specific situation, is another a way to generate gameplay. Stages for turn based RPGs can be generated with this approach so that the stages can be more diverse and less frustrating to play [90]. Multiplayer Shooter Games can also use deep learning to help with the balance of the game, either by the Level Generation and Weapon Parameters [91] or with Games Character Classes Parameters [92].

3.1.6.2 Adaptive Difficulty

Adaptive difficulty was also a common topic within this area. From tailoring the level design and gameplay experience after the individual player by utilising player models [110], evolving optimal 2D levels through incorpoating Interactive Evolutionary computation and agents [74], using Multi Swarm models and ecosystem mechanisms to provide more flexible game balance [75], to using Rhythm Group Theory to adapt 2D levels after player experience [76].

The remaining papers showcase on how to use PCG for AI behaviours. This includes topics like a implemention of Exploration Algorithms for making the NPC react more realistic via a heuristic approach [111], how to make a living eco system with the help of evolving A life [163] and how to generate behaviours for NPCs with the help of artificial evolution [174]. Bosses and their attack patterns can be generated with the help of generative grammars to generate programs for agent behaviours [172].
The final paper showcases on how crime and vignette encounters were generated in **Spiderman (2018)** [78].

The result showcases that Dynamic Systems are not as researched as other areas of PCG. As with the Puzzle Generation category, it needs more advanced algorithms for it to work and it is difficult and time consuming to make it work properly. The majority of papers are therefore mostly focused on adaptive difficulty, making the range of research limited.

3.1.7 Miscellanous Generation

This section covers areas of PCG that had only one or few papers to justify having their own sections.

3.1.7.1 Development Usage

Few papers covered building blocks that make up a game suchs as 3D models, textures, sounds, lightning and physics. With 3D models there where papers about Inverse Procedural Modeling [79], Inverse Modeling approach that learns L system represention from images [80] and a system that can create models from shape grammars with a node based setup [81]. There is also papers about generating rocks by evolving L systems in 3D [160], making 3D models and textures with Interactive Genetic Algorithm from user feedback [175] and on how to create 3D models by combining other 3D models with the help of Spatial Relation Graphs (paper made Lego models with this method) [176].

When it comes to textures, two papers talked about an example based framework that to produce texture based on the model that is being used [82] and using Local Random Phase Noise to create procedural textures [177].
Music and sound effects can be generated to help enchance the game based on the action of the player, like Experience Driven Music Generation which is based on what is happening in the gameplay [83] and a Data Driven Method to create sound from Soft Body Objects (cloth, rope etc) [84].
Character animations can also be procedurally animated based around the environment. One paper showcases a physics based controller together with inverse kinematic solver and random noise movement to get realistic human movement [93].
Even lightning effects can be generated for windows and buildings [85].

3.1.7.2 Game Usage

In terms of gameplay aspects, there is a few papers that cover these aspects like items, skills, weapons, animations and vegetation.
A few papers have been written on how to generate plants and vegetation with the help of PCG. Examples include using mathematical models to simulate leaf growth for different patterns and shapes [94], using different algorithms to generate trees [95], evolutionary algorithms that can replicate flora life cycle to generate 3D environments [96] and using Search Based PCG to generate different types of flowers in the game petals [97].
One paper had been written about on how to use a search based approach on role play games skill systems where the algorithm generates and suggest new character abilities for them that they can pick and choose from after every RPG battle [117]. Another paper in the area of RPG described two different rule based algorithms to generate items and their stats based on how player used them [98].
There were also papers that described how to generate weapons for multiplayer games like one paper discussed using genetic algorithms to continuously generate new weapons and used a fitness value to base on what it would generate [116]. Two other papers wrote about on how to generate weapons in **Unreal Tournament 3** by either simulation based approach [99] or evolutionary search algorithm [100] to change the weapon parameters and try to balance the weapons out.
The remaining papers were about placing out enemies in stealth games with a Monte Carlo approach [101], how the developers of **FINAL FANTASY XV** made an AI that could take enhanced photos of your gameplay to get stylish ways to represent gameplay [102] and how battle formations for troops in a strategy game can be generated with the help of coevolutionary genetic algorithms [164].

3.2 Industry Review Result

The industry review result was divided into both by primary game genre and what PCG area is being used. Genre was a difficult topic to be defined due to this area lacking consistent classification and agreed-upon criterias. The definition of what genre a game is classified under in this paper is based on the core gameplay of the game rather than visuals

and narrative features. This paper classifies five major genres used to classify games in this paper: Action, Adventure, RPG, Simulation and Strategy. Each game that uses PCG are then classified based on the genre or the subgenres that the game has. Another problem arrived on how to define games that have elements from two or more primary game genres. Example of this is Action-Adventure games and Action-RPGs. If a game covered two primary genres they where counted by the core gameplay they covered most in the game (an Action RPG game that had more to do with RPGs than with Action Games counted as a RPG game). The type of areas being used with PCG, where there is a multitude of primary genres, is going to be covered both in the result and in the discussion sections.. Each genre type and their PCG usage is covered in subsections in each chapter.

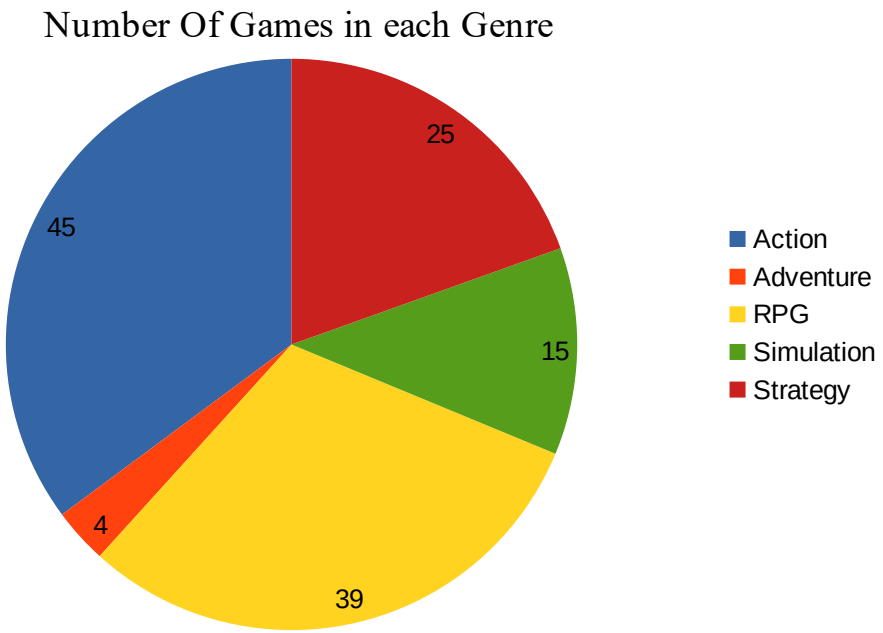

Table 2

Number of Game Genres in each Generation Category

	Level-generation	Design of content	Instansiating	user mediated	Dynamic	plot and puzzle
Action	22	12	20	1	7	0
Adventure	1	0	0	0	1	2
RPG	26	8	12	0	2	1
Simulation	11	1	2	5	3	3
Strategy	22	0	2	0	0	4

Table 3

3.2.1 Action Games

Action Games are defined as games where physical challenge that require hand eye coordination and motor skill to overcome the challenge.
Primary sub genres that are using PCG in practice are games in the Shooter genre, Platform genre and Survival genre. There is also a few games in the Rhythm genre that uses PCG techniques.
As we can see from the sample of games used in this study, it is seen that Procedural Generation has been mostly used for Level Generation, Design of Content and instansiating in game entities.
Shooters or the First Person Shooter genre uses Procedural Generation to generate the levels, place out enemies and generate different types of guns. Examples of this is **Galactic Arms Race** that uses algorithms to constantly evolve weapons based on how much the player uses them [119].
Shooters also uses Procedural Generation to make a more intresting level to play. Either by random generating it or by dynamically spawn encounters during the playthrough. A good example of this is the **Left 4 Dead** game which has an AI director that dictates when waves of zombies should spawn, where the zombies should spawn and what type of pickups should spawn based on how the players are doing and what situation they are in [120]. Other action games that uses directors to determine the spawning of enemies are games like **Risk of rain 2** [179], **Deep Rock Galactic** [180] and **Alien Swarm** [181].

Platformers primarly uses Procedural Generation for generating its levels like Infinite Runners (**Infinite Mario Bros**) [121] and Rougelikes (**Risk of Rain 1, Spelunky**) [122, 123].

Survival games use Procedural Generation in a similar way as to Shooter Games. They use them for generating the world/levels that are being played, spawn enemies and place out resources. Examples of survival games include open world survival games like **7 Days to Die, State of Decay series, Dont Starve, Valheim, Minecraft** etc.[124, 125, 126, 127, 190] These games usually have randomly generated maps that you spawn in when you start, randomizes the spawn of resources and spawn enemies under certain conditions.

Rhythm Games have also used Procedural Generation, such as **Audiosurf** [128] and **Beat Hazard** [129], where users can add in their own music to the game and the game generates a level for them. It also makes it one of the few times where Procedural Generation is used for user mediated content.

Action Games uses Procedural Generation during the development of the games, primarily in Open World Games, PCG is used to generate terrain or cities for the games the players are going to traverse over. Examples include **Spiderman (PS4)** [182] and **Agents of Mayhem** [183] for creation of cities and the **Just Cause** Game [184] for creation of terrain.

3.2.2 Adventure Games

Aventure Games are defined as games where the player solves various puzzles by interacting with characters or the environment and has very little action based challenges. This type of genre is more focused on exploration and puzzle solving and have a more emphasis on story and character. Subgenres include Text Games, Point and Click Games, Puzzle Adventure Games, Narrative Games, Walking Simulators, Visual Novels and Interactive Movies.
This genre does not have a lot of games that uses Procedural Generation, but when it does use it, it is usually for Dynamic Systems or Puzzle and Plot Generation.
One example is **Facade** which uses AI to decide how the character should react to you based on what you have typed going into the characters [130].
Another is **Heaven's Vault** which uses Procedural Generation to determine how puzzles should be generated [131].
A unique case of Procedural Generation is **AI Dungeon** which is a Text Adventure Game that generates story with the help of AI, based on what the player types in text box [132]. The game also has a way for players to create their own settings where the AI generates the setting based on what the player types into the game.

3.2.3 RPGs

Role Playing Games are games that centers around a character or a party of characters who completes quests and tasks to level up and grow stronger through experience. Role Playing Games usually have well defined worlds and more focus on story and setting,
Subgenres for this are action RPGs, MMORPGS, tactical RPGs, Roguelikes/Roguelites, Sandbox RPGs and First Person Party based RPGs.
The most common use of Procedural Generation in RPG games seems to be in Runtime-Random Level Generation, Design of Content, instanciating of ingame entities. A more moderate use with Dynamic Systems and very little for Plot and Puzzle Generation.
RPG primarly uses Procedural Generation for levels, dungeons and environment that the player can explore. Examples include Action RPGs like the **Diablo** games [133]and **Path of Exile** [134]. Other games that feature random generation of levels or dungeons include the **Persona** games [185] and **Remnant from the Ashes** [186].

Roguelike/Roguelites uses Procedural Generation in one form or another for both Level Generation and item spawning. Games like the **Binding of Isaac** [135], **Slay the Spire** [136], **Noita** [137] and **Crypt of the Necrodancer** [138] use Procedural Generation for levels, enemy spawning and what items the player gets.

Tactile RPG uses Procedural Generation for item statistics and random levels. The **Disgea** games [139] have items that have random statistics on them, that is depending on their rarity. These items have an "item world", that is a series of random levels that level up the item once the "item world" is completed, allowing the player to customize their character

even more. Another example of this is the **Borderlands** games, which uses Procedural Generation to generate guns with different models, stats and attributes [118]

Procedural Generation is used also to help developers to create the worlds during development time. **Elder Scrolls IV: Oblivion** and **Elder Scrolls V:Skyrim** [140, 141] used different kinds of algorithms like Erosion Algorithms to help create the open world of the games. Other types of games that uses procedural techniques during development are games like **Horizon: Zero Dawn** which developed a procedural tool so the designers could paint on the terrain and procedurally place out vegetation on the ground [142]. The newer Assassin's Creed games like **Assassin's Creed Valhalla** [143] uses procedural techniques to place vegetation throughout the world. MMO RPGs like **Star Wars Galaxies** and **EVE online** used Procedural Generation to generate their world as the massive size would have made making them manually too time consuming [144, 145].

Dynamic Systems can also be used to create more life like by making NPCs more belivable with there own goals and schedules. Such is the case with both **Elder Scrolls IV: Oblivion** and **Elder Scrolls V:Skyrim** with a system called radient AI where NPCs act dynamically by giving them goals and then let the system decide on what action they should take.

3.2.4 Simulation Games

Simulation Games are games which are designed to simulate reality in some form or another. Usually they are designed to closely simulate a real or fictional activity. This genre often has no strictly defined goals and are usually allowed control of the character or environment freely.
Subgenres within this genre are Construction and Management Simulation, Life Simulation, Sports genre, Vehicle genre and Wargame genre.
Simulation Games have a fair usage of Procedural Generation Level Generation, instansiating of in game entities being common and moderate usage of Dynamic Systems. It also has more usage of user mediated content and Plot and Puzzle Generation than most other genre.
Exampels of games within the simulation genre are **Spore** which simulates life and ecosystems in the game [146]. It allows the player to create their own creature with an editor that uses Procedural Animation to ensure to make it walk and navigate correctly.

Racing Games like **Dirt 4, Gran Turismo 5** and **Trackmania Turbo** have an optional mode to play on randomly generated tracks [147, 148, 187]. These tracks can be shared to other players to either race for better times or to play with other people.
Dwarf Fortress and **Rimworld** are Construction and Managemant Games which generate an entire world at the start and then have settlers settle down to build a base and to survive as long as possible [149, 150]

Other types of Construction games like **Factorio** [189] and the **Transport Fever**-series [188] generate just the world for the player to navigate and build in.

3.2.5 Strategy Games

Strategy Games is a genre that emphasize skillfull thinking and planning to achieve victory and usually includes warfare with tactical and strategic consideration, exploration and economy management. Players must plan ahead to outplay their opponent through good planning and management of resources to win in this type of game.
Subgenres that often uses PCG for their games are 4X (Explore, Expand, Exploit, Exterminate), Artillery, Turn Based Strategy, Grand Strategy and Real time Strategy Games.

A common use of PCG in Strategy Games is Runtime-Random Level Generation which a good majority of Strategy Games uses when they are using PCG. **Civilization** games is a 4X strategy game that have always had random maps for every new playthrough [151]. **Bad North** [152] uses wave function collapse algorithms to generate its levels. **Stellaris** [153] is a grand strategy 4x game that uses Procedural Generation to generate entire galaxies. The **Worms** games [187] features a mode where the map gets randomly generated.

Strategy Games also seem to take advantage of Procedural Plot and Puzzle Generation where it is mainly used to generate quests in world or character personalities in the world. Examples of usage within this area include the **Crusader King** games with randomized personalities and traits of the leaders based on the traits from the family tree [154].
The **Mount & Blade** games [155] uses it to randomly generate quests due to the games not having any linear story to follow.

4.Analysis and Discussion

As a result of the literature review, there were a lot papers that cover different areas and topics of PCG.
The most common topic to cover in these papers was in the area of Level Generation whether it is general Level Generation for areas like 2D platformers or dungeons, Terrain Generation for areas like Open World Games or City Generation to generate roads and cities for games.
Other areas that had quite few papers written about them were Plot and Puzzle Generation. While Puzzle Games had very few papers, Story and Narrative Generation was covered a lot, with topics such as Quest Generation, Narrative Generation, History Generation and NPC Personality Generation.
Instanciating of in-game entities was also covered quite widely with topics from 3D models and textures to weapons, vegetation and RPG skills.
Papers covering Design of Content tools was also covered a lot in the found research papers, with multiple of them talking about creating Mixed Initiative tools to help game designers create their ideas.
Dynamic Systems had a few papers too, most of them covering topics of Gameplay Adaption and Dynamic Difficulty. Other areas were about AI and Dynamic Encounters.
The only area where no research on PCG was found was User Mediated Content Generation.

The result of the industry review shows that there are certain areas in games that more commonly use PCG than others. A lot of games uses PCG in a similar way with some exceptions. Primarily Procedural Generation is used for Runtime-Random Level Generation, Design of Content and Instanciating of In-game Entities.
Both Runtime-Random Level Generation and Instanciating of In-game Entities are used for giving the games more random elements or replayability. Random levels/worlds are usually common among RPGs to make them more interesting to explore. Different kinds of items are also a common part of games to be procedurally generated because it gives a incentive to playing the game in hopes of finding more loot that is stronger or more interesting than what the player already has. It also cuts down the time for developers when creating unique weapons for the player to discover and find.
With Design of Content, game developers take this approach to aid game designers or to produce content in a bigger scale. Generally it is used for creating worlds like terrain, roads, cities and vegetation to either be used in game or for the game designers getting a start so they can modifiy it. It can also be used in unique ways like how **Assassins's Creed Odyssey** used Procedural Generation to generate all of its conversations in the game, so the designers later could tweak them to make them feel more alive[178].

Other areas of games that can have procedural content are a lot less used like Dynamic Systems, User Mediated Content and Puzzle and Plot Generation.
Dynamic Systems are usually used with AI, in order for it to react more dynamically to situations based on what has happened. Good examples of this is **Elder Scrolls** 4 and 5

with radient AI and the **S.T.A.L.K.E.R** series who have AI life systems, meaning that the NPC AI is not scripted and reacts based on rules and goals of the characters instead.
Puzzle and Plot Generation are commonly used to generate story for procedural worlds or for randomizing orders of how to solve a puzzle. A few games use it to generates it quest in games where stories are nonlinear.
User Generated Content is a very rare area within Procedural Generation to exist in games, due to the fact that a lot of player creation tools do not need Procedural Generation to make them work. A few exceptions exist like **Spore** with player created creatures that have procedural animations, racing games like **Dirt 4** and **Trackmania Turbo** to help players create race tracks and **AI Dungeon** where you can write in ideas and the generator creates scenarios for you. The reason that those games are using it, is because for the games to create that type of content, it is only possible through a procedural way.

Comparing the results from both the literature review and the industry review, we see that from both sides Level Generation, Design of Content and Entity Instanciating is featured a lot in both the industry review and the literature review. Level Generation is very common in games where is is mostly used when generating levels in Rougelikes, Survival Games, Strategy Games, RPG games and Multiplayer Maps. In the literature review, Level Generation research was the most common type of research with 47 papers about Level Generation and 20 about Terrain Generation.
Entity Instanciating is also featured heavily in games with examples such as item generation for RPGs, Gun Generation and spawning of NPCs and enemies. The literature review have a lot of papers covering topics such as Weapon Generation, Guard Placement, Character Generation and RPG Skill Generation.
With Design of Content there were a lot of games showcasing on how they used PCG to generate worlds for the designers during the games production. Same thing with the literature review, where papers were written on how generators could be more intuitive for designers and help them design content more easily.

Both the industry review and the literature review results featured variation and plenty of examples of on how Level Generation and Entity Instancing can be done with PCG. There is a difference between the study and the review when it comes to Design of Content with the industry review having results primarily from games that features Open World Design, mostly from Action Adventure Games and Action RPG. The literature review contained more varied types of content design tools that could be used to help designers like tools to help generate quests, stories, dungeons and 2D levels.

A major difference between the literature review and the industry review is the reviewing of the usage of PCG for Dynamic System and Puzzle and Plot Generation. The numbers of cases was quite small compared to more results in these categories from the literature review. In the Dynamic System area there was a couple of instances within the realm of AI in the industry review, while the literature review contained more examples of PCG for Dynamic System and most of them was about gameplay adaption and adaptability difficulty. Examples of this was balancing of Multiplayer Games, missions based on player

skills and RPG encounters for gameplay department and balancing levels after player experience in the adaptability department.
Within Plot and Puzzle Generation, there were a lot more in the literature review than in the industry review. No cases of Puzzle Generation exist in the games while there is a few in the research area. Plot Generation had a lot more papers from the literature review than in the industry review and the literature review covered more varied topics than what was used in games from the industry review. This includes generating puzzles for narrative games and how to generate physics based puzzles.
The one area that the industry review had more of than the literature review was the area of User Mediated Content. The literature review had no research paper specifically about User Mediated Content while the industry review explored that in three games. These games usually used PCG to generate models and animations the player had created and this specific area was covered in research like how to generate models with L systems and how textures can be procedurally generated. But even then, these paper were also few with only one to two papers in the area.

When it comes to PCG usage within certain genres, the major factors of how often PCG is being used for different genres, are dependant on how necessary Procedural Generation is for the part that genre emphasis, how common the genre is and how easy is it to implement. Action Games and RPG are incredibly common genres in the game market and are usually mixed with other genres, which means that Procedural Generation in this genre is a common.
Simulation and Strategy Games are less common genres, but they still feature some Procedural Generation due to certain subgenres using it enhance their gameplay with random Level Generation like race tracks for racing games and random worlds for construction games.
Adventure Games rarely features Procedural Generation due to the element that defines Adventure Games are very hard to get good results with current techniques that exist. Even if developers wanted to use it for Adventure Games, it would be too much time investment that it would be better to make an ordinary standard story. Adventure Games are also quite niche with very few games being made in these genre in comparison to the likes of Action and Role Playing genre.
What areas that are being used with PCG are not specific to any specific genre. While certain genres priorities different areas depending on what type of gameplay that genre emphasis, there is no area for any genre that is impossible to implement with PCG. There is just less justification to implement PCG for the puzzles in a action game unless the game is focused on it or if game developers find I good reason to add it.

When genres are combined, it is usually the genre that is the most dominating that controls what kind of PCG that gets used or the most common areas of each major genre is used in these combinations. Action-adventure games combines the puzzle solving of adventure games and obstacles that require elements of action games to overcome. These games usually use PCG in the same way as Action games like level-generation and enemy spawning, but also adventure games such as puzzle generation. Similar thing happens in

Action-role playing games where combinations of PCG used in similar areas existing in both action games and RPG games.

The reason the usage of Procedural Generation is like it is now, is because of the random nature of Procedural Generation and how unpredictable the outcome of the content can become. When it comes to develop games, most developers want to know that the content that they create works properly in their game. The major reasons why developers would not want to use Procedural Generation either for ease of development or to produce content during runtime is because the techniques or methods are unintuitive to use, hard to control and that result generated from different methods can be hard to integrate into a complete and consistent part of the game [105]. The biggest reason as to why random Level Generation, Design of Content and Entity Instanciating is the most common types of Procedural Generation is because the content they generate can be a lot more controlled and that the techniques for producing this type of content has become a lot more easier to integrate into games and are a lot more controllable. In the other areas it is rare to see PCG being used to generate their content. This is due to that these areas cover a more complicated and advanced type of content that has more compontents that can break as well as the techniques/methods in this field for generating this type of content is still too under-developed and is too unintuative to use.

Research within PCG show promise of new ways to generate content, but most of these papers showcases prototypes of the techniques/generators and most of these papers mentions how they need further research to be used in a game.
From the literature review and what each genre want to emphasize, it would seem that Level Generation and Instanciating of Entities is commonly used and commonly researched and each genre uses these areas to their full potential.
With Content Design, games uses them a lot but there is still potential to have new ways to generate content for this area, especially for other subgenres of Action and RPG like Platformer Games and Dungeon Generator Tools. Adventure Games could also benefit from using Content Design more.
The largest potential would be more PCG usage in areas such as Story, Puzzles, AI and Dynamic Systems. From the papers we see potential on usage from Plot and Puzzle Generation, espacially for games in RPG genre and the Adventure genre.
With Dynamic Systems, every genre have potential to be better using PCG. From the papers we see plenty of examples on how gameplay and difficulty can be adapted and be made to be more enjoyable to play.
User Mediated Content is rare due to Player Created Tools that need requirement of PCG for it to be used is uncommon in games. Even then, most User Mediated Content Generation use techniques such as Model Generation. User Mediated Content would benefit to have more research done in this area to help develop better tools for Player Creation Tools.

In areas where PCG is not used today and where it would be beneficial to use, more research should be done to produce tools and techniques that make it easier for developers

and designers to add PCG elements to these under-utilized areas like Story Generation and Dynamic Systems. The reason why these areas are so under-utilized is because that the areas are much more complex to generate procedurally and has a lot more factors that has to be taken into account. In order to make the generated content feel sufficient for the game that is being developed.
When trying to procedurally generate AI or story, developers have to keep track of more areas that has to be taken into account for to make sure that the generation generates correctly and does not corrupt or feel out of place in the game.
Reasons why PCG has started to become more commonly used is because of new tools that help developers with the development of certain areas in games. Examples of these new tools include Speedtree [156], Houdini [157] and Substance Painter [158] that has become commonplace in the workflow of games. Tools for more complicated areas of games would make them more appealing for developers and publishers to use them with the promise of increased creativity, speeding up work and reducing cost of development.

Limitations of this study was that even if a lot of games have been gathered from the industry review, there is still potential games that uses PCG that have missed being researched either due to a lack of documented information or because the games are obscure. Missed games could be using PCG, but could not be part of the industry review due to lack of legitimate sources or lacking in documentation. This could have been migitated with interviews where the developers could have been interviewed about the systems used for PCG and how they used them. This could not have been done due to being too big of a task for a one man team to find contacts and interview them one by one.
Another limitation from this study was that researching algorithm use in different games was quite difficult, which made this study drop researching it entirely. Very few games document on what type of algorithms they use to generate the content. Most of them only mention what area of content they use PCG for and what it does, but rarely do they share how the algorithms work. This is in contrast to the literature review where research papers talks about different usage of different PCG algorithms and techniques. This again could have been solved by conducting interviews with game developers about their use of PCG algorithms.

5.Conclusion

This paper was written to find out how Procedural Generation is used in different genres of games, find out what papers have been written about different approaches to Procedural Generation and find if Procedural Generation is used to its full potential or if there is more that Procedural Generation can do to benefit different types of genres.

The results found that while Procedural Generation is being used as a tool for creativity, replayability and ease of development, certain genres still have areas that could innovate and benefit by using Procedural Generation.
Most games use Procedural Generation for random Level Generation, Design of Game Content and Instantiating of Entities. Dynamic Systems, User Mediated Content and Puzzle and Plot Generation are less common to be used in games. Genres were divided into Action Games, Adventure Games, RPG, Simulation Games and Strategy Games.
It was found that Level Generation is heavily featured in a lot of games and is common in a lot of genres.
Design of Content is most found for development of Action games and RPG.
Instanciating of In-game Entities where most common in Action Games but existed in the other genres too.
Dynamic Systems had few instances of usage in games compared to other areas, but existed consitently through every genre.
User Mediated Content had only a few uses in Simulation and Strategy Games.
Plot and Puzzle Generation had also few games using it and only a single game had it for every genre except Action Games.

The literature review showcases a lot of papers on research for Level Generation, Entity Instansiating and Content Design meaning that there is plenty of choice to use PCG in these areas, but there was also a lot of papers in Dynamic Systems and Plot and Puzzle Generation meaning that PCG has more potential usage for games to generate new and interesting content.

Future research for Procedural Generation should focus on making Procedural Generation techniques and methods for Dynamic Systems, AI, Story and Puzzle generation more appealing for developers and designers to use, so that they can find more innovative use for them in games.

If a more indepth review of this kind should be conducted, the suggestion would be to have a larger selections of games to study on, conduct interviews with developers to get a deeper insight as to way they choose Procedural Generation for their content and also study more indepth on what types of techniques game developers used for their games to find out what techniques are more common and uncommon in the game development world. Either through looking at the games source code, interviewing developers or through company/developer sources like blogpost and conferences.

References

[1] *Barbara Kitchenham a, O. Pearl Brereton a , David Budgen b , Mark Turner a , John Bailey b , Stephen Linkman , "Systematic literature reviews in software engineering – A systematic literature review", Proceedings of the Third International Symposium on Empirical Software Engineering and Measurement, ESEM 2009, October 15 16, 2009, Lake Buena Vista, Florida, USA, October 2009*

[2] *Mark Hendrikx, Sebastiaan Meijer, Joeri Van Der Velden, and Alexandru Iosup, "Procedural Content Generation for Games: A Survey", Multimedia Comput. Commun. Appl. 9, 1, Article 1 (February 2013), 22 pages.*

[3] *Ruben M. Smelik, Klaas Jan de Kraker, Saskia A. Groenewegen, Tim Tutenel and Rafael Bidarra, "A Survey of Procedural Methods for Terrain Modelling", Netherlands Organization for Scientific Research and the Netherlands ICT Research and Innovation Authority*

[4] *George Kelly and Hugh McCabe, "A Survey of Procedural Techniques for city generation", January 2006. Available:*http://www.citygen.net/ *[accessed April 2021]*

[5] *Barbara De Kegel and Mads Haahr, "Procedural Puzzle Generation: A Survey", IEEE Transactions on Games , Volume: 12, Issue: 1, March 2020*

[6] *Claes Wohlin, "Guidelines for Snowballing in Systematic Literature Studies and a Replication in Software Engineering", EASE '14: Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering, May 2014, Article No.: 38, Pages 1–10*

[7] *Julian Togelius, Emil Kastbjerg, David Schedl and Georgios N. Yannakakis, "What is Procedural Content Generation? Mario on the borderline", PCGames 2011 June 28, 2011, Bordeaux, France.*

[8] *Daniel Michelon De Carli, Fernando Bevilacqua, Cesar Pozzer and Marcos Cordeiro d'Ornellas, "A Survey of Procedural Content Generation Techniques Suitable to Game Development", Games and Digital Entertainment (SBGAMES), 2011 Brazilian Symposium, November 2011*

[9] *Daniel Michelon De Carli, Fernando Bevilacqua, Cesar Tadeu Pozzer and Marcos Cordeiro d'Ornellas "A Survey of Procedural Content Generation Techniques Suitable to Game Development", Games and Digital Entertainment (SBGAMES), November 2011*

[10] *Breno M. F. Viana and Selan R. dos Santos "A Survey of Procedural Dungeon Generation" 2019 18th Brazilian Symposium on Computer Games and Digital Entertainment (SBGames), October 2019*

[11] *George Kelly and Hugh McCabe "A Survey of Procedural Techniques for City Generation" School of Informatics and Engineering,Institute of Technology, Blanchardstown, Dublin 15, Ireland, January 2006*

[12] *Julian Togelius, Georgios N.Yannakakis, Kenneth O.Stanley and Cameron Browne, "Search Based Procedural Content Generation: A Taxonomy and Survey" IEEE TRANSACTIONS ON COMPUTATIONAL INTELLIGENCE AND AI IN GAMES ,VOL.3,NO.3,SEPTEMBER2011*

[13] *Nuno Barreto, Amílcar Cardoso and Licínio Roque "Computational Creativity in Procedural Content Generation:A State of the Art Survey", Videojogos, November 2014*

[14] *Aitor Santamar´ıa Ibirika, Xabier Cantero, Mikel Salazar, Jaime Devesa and Pablo G. Bringas "Volumetric Virtual Worlds with Layered Terrain Generation", 2013 International Conference on Cyberworlds*

[15] Thomas J. Rose and Anastasios G. Bakaoukas ,"*Algorithms and Approaches for Procedural Terrain Generation" 2016 8th International Conference on Games and Virtual Worlds for Serious Applications, September 2016*
[16] *Tuomo Hyttinen, Erkki Mäkinen, and Timo Poranen, "Terrain Synthesis Using Noise by Examples", AcademicMindtrek '17: Proceedings of the 21st International Academic Mindtrek Conference, September 2017*
[17] *Aurimas Petrovas and Romualdas Baušys, "Automated Digital Terrain Elevation Modification By Procedural Generation Approach", 2019 Open Conference of Electrical, Electronic and Information Sciences (eStream), April 2019*
[18] *Elizabeth A. Matthews and Brian A. Malloy* ,"*Procedural Generation of Story Driven Maps", The 16th International Conference on Computer Games, July 2011*
[19] *Khalil satyadama, Reza Fuad Rachmadi and Supeno Mardi Susiki Nugroho, "Procedural Environment Generation for Cave 3D Model Using OpenSimplex Noise and Marching Cube", 2020 International Conference on Computer Engineering, Network and Intelligent Multimedia, 2020*
[20] *Ryan J. Vitacion and Li Liu, "Procedural Generation of 3D Planetary Scale Terrains", 2019 IEEE International Conference on Space Mission Challenges for Information Technolog, 2019*
[21] *Génevaux, J., Galin, É., Guérin, É., Peytavie, A., Beneš, B. 2013. Terrain Generation Using Procedural Models based on Hydrology. ACM Trans. Graph. 32, 4, Article 143 (July 2013), 10 pages.*
[22] *Georgi Ivanov, Magnus Håkon Petersen, Kristián Kovalský, Kristian Eng berg, and George Palamas. 2020. An Explorative Design Process for Game Map Generation Based on Satellite Images and Playability Factors. In In ternational Conference on the Foundations of Digital Games (FDG '20), Sep tember 15–18, 2020, Bugibba, Malta. ACM, New York, NY, USA, 4 pages.*
[23] *N. Mikuličić and Ž. Mihajlović, "Procedural Generation of Mediterranean Environments", MIPRO 2016, May 30 June 3, 2016*
[24] *Rahul Dey, Jason G. Doig, and Christos Gatzidis. 2017. Procedural Feature Generation for Volumetric Terrains. In Proceedings of SIGGRAPH '17 Posters, Los Angeles, CA, USA, July 30 August 03, 2017, 2 pages.*
[25] *Asiiah Song and Jim Whitehead. 2019. TownSim: Agent based city evolution for naturalistic road network generation. In The Fourteenth International Conference on the Foundations of Digital Games (FDG '19), August 26–30, 2019, San Luis Obispo, CA, USA. ACM, New York, NY, USA, 9 pages.*
[26] *Rahul Sharma, "Procedural City Generator", 5th International Conference on System Modeling & Advancement in Research Trends, 25th_27'h November, 2016*
[27] *Gabriel Dias Fernandes and António Ramires Fernandes, "Space Colonisation for Procedural Road Generation", 2018 International Conference on Graphics and Interaction (ICGI), November 2018*
[28] *Schwarz, M., Mueller, P. 2015. Advanced Procedural Modeling of Architecture. ACM Trans. Graph. 34, 4, Article 107 (August 2015), 12 pages*
[29] *Diego Jesus, António Coelho and António Augusto Sousa, "Towards Interactive Procedural Modelling of Buildings", Proceedings of the 31st Spring Conference on Computer Graphics, April 2015*

[30] *Ivan Silveira, Daniel Camozzato, Fernando Marson, Leandro Dihl, and Soraia Raupp Musse,"Real time Procedural Generation of Personalized Fac¸ade and Interior Appearances Based on Semantics", 2015 14th Brazilian Symposium on Computer Games and Digital Entertainment, November 2015*
[31] *Jonas Freiknecht and Wolfgang Effelsberg, "Procedural Generation of Multistory Buildings With Interior", IEEE TRANSACTIONS ON GAMES, VOL. 12, NO. 3, SEPTEMBER 2020*
[32] *Werner Gaisbauer, William L. Raffe, Jaime A. Garcia and Helmut Hlavacs, "Procedural Generation of Video GameCities for Specific Video Game Genres Using WaveFunctionCollapse (WFC)", CHI PLAY '19 Extended Abstracts: Extended Abstracts of the Annual Symposium on Computer Human Interaction in Play Companion Extended Abstracts, October 2019, Pages 397–404*
[33] *Xiaowei Zhang, Aly Shehata, Bedrich Benes, and Daniel Aliaga. 2020. Automatic Deep Inference of Procedural Cities from Global scale Spatial Data. ACM Trans. Spatial Algorithms Syst. 7, 2, Article 6 (October 2020), 28 pages.*
[34] *Chris Sexton and Benjamin Watson, "Vectorization of Gridded Urban Land Use Data", Proceedings of the 2010 Workshop on Procedural Content Generation in Games, June 2010*
[35] *Isaac Karth and Adam M. Smith. 2017. WaveFunctionCollapse is Constraint Solving in the Wild. In Proceedings of FDG'17, Hyannis, MA, USA, August* 14 17, 2017, 10 pages.
[36] *Tobias Møller, Jonas Billeskov, and George Palamas. 2020. Expanding Wave Function Collapse with Growing Grids for Procedural Map Generation. In International Conference on the Foundations of Digital Games (FDG '20), September 15–18, 2020, Bugibba, Malta. ACM, New York, NY, USA, 4 pages*
[37] *Alberto Alvarez, Steve Dahlskog, Jose Font, Johan Holmberg, Chelsi Nolasco and Axel Österman, "Fostering Creativity in the Mixed Initiative Evolutionary Dungeon Designer", Proceedings of the 13th International Conference on the Foundations of Digital Games, August 2018*
[38] *Alexander Baldwin, Steve Dahlskog, Jose M. Font, and Johan Holmberg. 2017. Towards Pattern Based Mixed Initiative Dungeon Generation. In Pro ceedings of FDG'17, Hyannis, MA, USA, August 14 17, 2017, 10 pages.*
[39] *Joris Dormans, "Level Design as Model Transformation: A Strategy for Automated Content Generation", Proceedings of the 2nd International Workshop on Procedural Content Generation in Games, June 2011*
[40] *Simon Liu, Li Chaoran, Li Yue, Ma Heng, Hou Xiao, Shen Yiming, Wang Licong, Chen Ze, Guo Xianghao, Lu Hengtong, Du Yu, and Tang Qinting. 2019. Automatic generation of tower defense levels using PCG: Full Paper. In The Fourteenth International Conference on the Foundations of Digital Games (FDG '19), August 26–30, 2019, San Luis Obispo, CA, USA. ACM, New York, NY, USA, 9 pages.*
[41] *Yu Du, Jian Li, Xiao Hou, Hengtong Lu, Simon Cheng Liu, Xianghao Guo, Kehan Yang and Qinting Tang, "Automatic Level Generation for Tower Defense Games", 2019 IEEE 3rd Information Technology,Networking,Electronic and Automation Control Conference (ITNEC 2019), March 2019*

[42] *Paul Hyunjin Kim, Jacob Grove, Skylar Wurster, and Roger Crawfis. 2019. Design Centric Maze Generation. In The Fourteenth International Conference on the Foundations of Digital Games (FDG'19), August 26–30, 2019, San Luis Obispo, CA, USA. ACM, New York, NY, USA, 9 pages.*
[43] *Thomas Smith, Julian Padget, and Andrew Vidler. 2018. Graph based Gener ation of Action Adventure Dungeon Levels using Answer Set Programming. In Proceedings of FDG, Malmö, Sweden, August 7 10, 2018 (FDG'18), 10 pages.*
[44] *Xenija Neufeld, Sanaz Mostaghim and Diego Perez Liebana, "Procedural Level Generation with Answer Set Programming for General Video Game Playing", 2015 7th Computer Science and Electronic Engineering Conference, September 2015*
[45] *Santiago Londoño and Olana Missura, "Graph Grammars for Super Mario Bros ∗ Levels", Sixth FDG Workshop on Procedural Content Generation, June 2015*
[46] *Joris Dormans, "Adventures in Level Design: Generating Missions and Spaces for Action Adventure Games", Proceedings of the 2010 Workshop on Procedural Content Generation in Games, June 2010*
[47] *Julian Togelius, Mike Preuss and Georgios N. Yannakakis, "Towards multiobjective procedural map generation", Proceedings of the 2010 Workshop on Procedural Content Generation in Games,June 2010*
[48] *Paul Hyunjin Kim and Roger Crawfis, "Intelligent Maze Generation based on Topological Constraints", 2018 7th International Congress on Advanced Applied Informatics, July 2018*
[49] *Daniel Karavolos, Antonios Liapis and Georgios Yannakakis, "A Multifaceted Surrogate Model for Search Based Procedural Content Generation", IEEE TRANSACTIONS ON GAMES, VOL. 13, NO. 1, MARCH 2021*
[50] *Jim Whitehead. 2020. Spatial Layout of Procedural Dungeons Using Linear Constraints and SMT Solvers. In Proceedings of the 11th Workshop on Proce dural Content Generation (PCG 2020), September 15–18, 2020, Bugibba, Malta. ACM, New York, NY, USA, 9 pages.*
[51] *Mariana Werneck and Esteban W. G. Clua ,"Generating Procedural Dungeons Using Machine Learning Methods", 2020 19th Brazilian Symposium on Computer Games and Digital Entertainment, Novemeber 2020*
[52] *Anurag Sarkar and Seth Cooper. 2020. Sequential Segment based Level Generation and Blending using Variational Autoencoders. In International Conference on the Foundations of Digital Games (FDG '20), September 15–18, 2020, Bugibba, Malta. ACM, New York, NY, USA, 9 pages*
[53] *Peter Ziegler, and Sebastian von Mammen "Generating Real Time Strategy Heightmaps using Cellular Automata", International Conference on the Foundations of Digital Games, September 2020*
[54] *William Forsyth, "Globalized random procedural content for dungeon generation", Journal of Computing Sciences in Colleges, Volume 32, Issue 2, December 2016*
[55] *Jessica R. Baron. 2017. Procedural Dungeon Generation Analysis and Adaptation. In Proceedings of ACM SE '17, Kennesaw, GA, USA, April 13 15, 2017, 5 pages.*
[56] *Chris Miller, Mayank Dighe, Chris Martens, and Arnav Jhala. 2019. Stories of the Town: Balancing Character Autonomy and Coherent Narrative in Procedurally Generated*

Worlds. In The Fourteenth International Conference on the Foundations of Digital Games (FDG ’19), August 26–30, 2019, San Luis Obispo, CA, USA. ACM, New York, NY, USA, 9 pages.
[57] *Jonas Freiknecht and Wolfgang Effelsberg. 2018. Procedural Generation of Interactive Stories using Language Models. In Proceedings of the International Conference on the Foundations of Digital Games (FDG ’20). ACM, New York, NY, USA, 8 pages*
[58] *Matheus R. F. Mendonça and Artur Ziviani. 2018. Network Based Procedural Story Generation. Comput. Entertain. 16, 3, Article 3 (September 2018), 18 pages.*
[59] *Erik M. Fredericks and Byron DeVries, ”(Genetically) Improving Novelty in Procedural Story Generation”, 2021 IEEE/ACM International Workshop on Genetic Improvement (GI), May 2021*
[60] *Eric LaBouve, Erik Miller, and Foaad Khosmood. 2019. Enhancing Story Generation with the Semantic Web. In The Fourteenth International Confer ence on the Foundations of Digital Games (FDG ’19), August 26–30, 2019, San Luis Obispo, CA, USA. ACM, New York, NY, USA, 11 pages.*
[61] *Iwona Grabska Gradzińska, Leszek Nowak, Wojciech Palacz, and Ewa Grab ska. 2021. Application of Graphs for Story Generation in Video Games. In 2021 Australasian Computer Science Week Multiconference (ACSW ’21), February 01–05, 2021, Dunedin, New Zealand. ACM, New York, NY, USA, 6 pages*
[62] *James Ryan, “Diegetically Grounded Evolution of gameworld languages”, Proceedings of 1st International Joint Conference of DiGRA and FDG, August 2016*
[63] *Jason A. Hall, Benjamin Williams, and Christopher J. Headleand “Artificial Folklore for Simulated Religions”, 2017 International Conference on Cyberworlds, September 2017*
[64] *Jonathon Doran and Ian Parberry "A Prototype Quest Generator Based on a Structural Analysis of Quests from Four MMORPGs”, Proceedings of the 2nd International Workshop on Procedural Content Generation in Games, June 2011*
[65] *Thongtham Chongmesuk and Vishnu Kotrajaras, ”Multi Paths Generation for Structural Rule Quests”, 2019 16th International Joint Conference on Computer Science and Software Engineering (JCSSE), July 2019*
[66] *Bas in het Veld, Ben Kybartas, Rafael Bidarra and John Jules Ch. Meyer, “Procedural Generation of populations for storytelling”, Sixth FDG Workshop on Procedural Content Generation, June 2015*
[67] *Erica Jurado, Kirsten Emma Gillam and Joshua McCoy. 2019. Non Player Character Personality and Social Connection Generation. In Proceedings of the 2019 Foundations of Digital Games Conference (FDG’19). FDG, San Luis Obispo, CA, USA, 7 pages.*
[68] *Artur O. R. Franco Jos ́e G. R. Maia and Joaquim A. M. Neto Fernando A. C. Gomes, ”An Interactive Storytelling Model for Non Player Characters on Electronic RPGs”, 2015 14th Brazilian Symposium on Computer Games and Digital Entertainment. November 2015*
[69] *Jason Grinblat and C. Brian Bucklew . 2017. Subverting Historical Cause & Effect: Generation of Mythic Biographies in Caves of Qud. In Proceedings of FDG’17, Hyannis, MA, USA, August 14 17, 2017, 7 pages.*
[70] *Corinna Jaschek, Tom Beckmann, Jaime A. Garcia, and William L. Raffe ”Mysterious Murder MCTS driven Murder Mystery Generation”, 2019 IEEE Conference on Games (CoG), Augusti 2019*

[71] *Max Kreminski, Devi Acharya, Nick Junius, Elisabeth Oliver, Kate Compton, Melanie Dickinson, Cyril Focht, Stacey Mason, Stella Mazeika, and Noah Wardrip Fruin. 2019. Cozy Mystery Construction Kit: Prototyping Toward an AI Assisted Collaborative Storytelling Mystery Game. In The Fourteenth Inter national Conference on the Foundations of Digital Games (FDG '19), August 26–30, 2019, San Luis Obispo, CA, USA. ACM, New York, NY, USA, 9 pages.*
[72] *Jonathan Lessard, Etienne Brunelle Leclerc, Timothy Gottschalk, Marc Antoine Jetté Léger, Odile Prouveur, and Christopher Tan. 2017. Striving for Author Friendly Procedural Dialogue Generation. In Proceedings of FDG'17, Hyannis, MA, USA, August 14 17, 2017, 6 pages.*
[73] *Aaron A. Reed, "Sharing Authoring with Algorithms", Proceedings of the The third workshop on Procedural Content Generation in Games, May 2012*
[74] *Daniel Wheat, Martin Masek, Chiou Peng Lam and Philip Hingston ,"Dynamic Difficulty Adjustment in 2D Platformers through Agent based Procedural Level Generation", 2015 IEEE International Conference on Systems, Man, and Cybernetics, October 2015*
[75] *Wen Xia and Bhojan Anand, "Game Balancing with Ecosystem Mechanism", 2016 International Conference on Data Mining and Advanced Computing (SAPIENCE), March 2016*
[76] *Henry D. Fern´andez B., Mikami Koji and Kunio Kondo, "Adaptable Game Experience Based on Player's Performance and EEG", 2017 NICOGRAPH International, September 2017*
[77] *Jonathan Roberts and Ke Chen, "Learning Based Procedural Content Generation", IEEE TRANSACTIONS ON COMPUTATIONAL INTELLIGENCE AND AI IN GAMES, VOL. 7, NO. 1, MARCH 2015*
[78] *David Santiago. 2019. Procedural System Assisted Authoring of Open World Content for Marvel's Spider Man. In Proceedings of SIGGRAPH '19 Talks. ACM, New York, NY, USA, 2 pages.*
[79] *Daniel G. Aliaga, İlke Demir, Bedrich Benes, Michael Wand, "Inverse Procedural Modeling of 3D Models for Virtual Worlds", SIGGRAPH '16: ACM SIGGRAPH 2016 Courses, July 2016*
[80] *Jianwei Guo, Haiyong Jiang, Bedrich Benes, Oliver Deussen, Xiaopeng Zhang, Dani Lischinski, and Hui Huang. 2020. Inverse Procedural Mod eling of Branching Structures by Inferring L Systems. ACM Trans. Graph. 39, 5, Article 155 (June 2020), 13 pages*
[81] *Pedro Brandão Silva, Pascal Müller, Rafael Bidarra and António Coelho, "Node Based Shape Grammar Representation and Editing" 2013*
[82] *Yiwei Hu, Julie Dorsey, and Holly Rushmeier. 2019. A Novel Framework For Inverse Procedural Texture Modeling. ACM Trans. Graph. 38, 6, Article 186 (November 2019), 14 pages.*
[83] *David plans and Davide Morelli, "Experience Driven Procedural Music Generation for Games", IEEE TRANSACTIONS ON COMPUTATIONAL INTELLIGENCE AND AI IN GAMES, VOL. 4, NO. 3, SEPTEMBER 2012*
[84] *Feng Su and Chris Joslin. 2019. Procedural Sound Generation for Soft Bodies in Video Games. In Proceedings of MIG '19: Motion, Interaction and Games (MIG '19), October 28 30, 2019, Newcastle upon Tyne, United King dom. ACM, New York, NY, USA, 12 pages.*

[85] *Jennifer Chandler, Lei Yang and Liu Ren, "Procedural Window Lighting Effects for Real Time City Rendering", Proceedings of the 19th Symposium on Interactive 3D Graphics and Games, February 2015*
[86] *Alexander Zook, Stephen Lee Urban, Michael R. Drinkwater and Mark O. Riedl, "Skill based Mission Generation: A Data driven Temporal Player Modeling Approach", Proceedings of the The third workshop on Procedural Content Generation in Games, May 2012*
[87] *Sander Bakkes and Shimon Whiteson, "Towards Challenge Balancing for Personalised Game Spaces", 2014 IEEE Games Media Entertainment, October 2014*
[88] *Norbert Heijne and Sander Bakkes. 2017. Procedural Zelda: A PCG Envi ronment for Player Experience Research. In Proceedings of FDG'17, Hyannis, MA, USA, August 14 17, 2017, 10 pages.*
[89] *Sauvik Das, Alexander Zook and Mark O. Riedl, "Examining Game World Topology Personalization", Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, April 2015*
[90] *SangGyu Nam and Kokolo Ikeda, "Generation of Diverse Stages in Turn Based Role Playing Game using Reinforcement Learning", 2019 IEEE Conference on Games (CoG), Augusti 2019*
[91]*Daniel Karavolos, Antonios Liapis and Georgios Yannakakis, "Learning the Patterns of Balance in a Multi Player Shooter Game", Proceedings of the 12th International Conference on the Foundations of Digital Games, August 2017*
[92] *Daniel Karavolos, Antonios Liapis and Georgios Yannakakis, "*Pairing Character Classes in a Deathmatch Shooter Game via a Deep Learning Surrogate Model"*, Proceedings of the 13th International Conference on the Foundations of Digital Games, August 2018*
[93] *Ben Kenwright ,"Generating Responsive Life Like Biped Characters", Proceedings of the The third workshop on Procedural Content Generation in Games, May 2012*
[94] *Dr. Monssef Alsweis and Prof. Oliver Deussen, "Procedural Techniques for Simulating the Growth of Plant Leaves and Adapting Venation Patterns", Proceedings of the 21st ACM Symposium on Virtual Reality Software and Technology, November 2015*
[95] *H. Nuić and Ž. Mihajlović, "Algorithms for Procedural Generation and display of trees", MIPRO 2019, May 20 24, 2019*
[96] *Carlos Mora, Sandra Jardim and Jorge Valente, "Flora Generation and Evolution Algorithm for Virtual Environments", 2021 16th Iberian Conference on Information Systems and Technologies (CISTI) 23 – 26 June 2021*
[97] *Sebastian Risi, Joel Lehman, David B. D'Ambrosio, Ryan Hall, Kenneth O. Stanley "Petalz: Search Based Procedural Content Generation for the Casual Gamer", IEEE TRANSACTIONS ON COMPUTATIONAL INTELLIGENCE AND AI IN GAMES, VOL. 8, NO. 3, SEPTEMBER 2016*
[98] *Ng Wai Foong, Chin Kim On, Rayner Alfred, Jason Teo, and Ag Asri Ag Ibrahim, "Interactive Procedural Generation for Items in Role Playing Game", 2017 IEEE 2nd International Conference on Automatic Control and Intelligent Systems (I2CACIS 2017), 21 October 2017, Kota Kinabalu, Sabah, Malaysia*
[99] *Daniele Gravina and Daniele Loiacono, "Procedural Weapons Generation for Unreal Tournament III", 2015 IEEE Games Entertainment Media Conference (GEM), October 2015*
[100] *Daniele Gravina, Antonios Liapis and Georgios N. Yannakakis, "Constrained Surprise Search for Content Generation", 2016 IEEE Conference on Computational Intelligence and Games (CIG), September 2016*

[101] *Qihan Xu, Jonathan Tremblay, Clark Verbrugge,"Procedural Guard Placement for Stealth Games", January 2014*
[102] *Prasert Prasertvithyakarn, Tatsuhiro Joudan, Hidekazu Kato, Seiji Nanase, Masayoshi Miyamoto, and Isamu Hasegawa. 2017. Procedural Photograph Generation from Actual Gameplay: Snapshot AI in FINAL FANTASY XV. In Proceedings of SIGGRAPH '17 Talks, Los Angeles, CA, USA, July 30 August 03, 2017, 2 pages.*
[103] *Y. I. H. Parish and P. Müller, "Procedural modeling of cities" in Proceedings of the 28th annual conference on Computer graphics and interactive techniques SIGGRAPH '01 ser. SIGGRAPH '01, New York, NY, USA:ACM, pp. 301 308, 2001.*
[104] *E. Teng and R. Bidarra, "A semantic approach to patch based Procedural Generation of urban road networks", Proceedings of PCG 2017 Workshop on Procedural Content Generation for Games co located with the Twelfth International Conference on the Foundations of Digital Games, 2017.*
[105] *Ruben Smelik, Tim Tutenel, Klaas Jan de Kraker and Rafael Bidarra ,"Integrating Procedural Generation and manual editing of virtual worlds", PCGames '10: Proceedings of the 2010 Workshop on Procedural Content Generation in Games, Article No.2, Pages 1–8, June 2010*
[106] *Gamasutra (August 6, 2008) "Random Scattering: Creating Realistic Landscapes"* https://www.gamasutra.com/view/feature/130071/random_scattering_creating_.php?page=2 *retrived: May 2021*
[107] *N. Shaker, J. Togelius, and M. J. Nelson, Procedural Content Generation in Games. Springer, 2016*
[108] *Riccardo Cantoni, Jacopo Essenziale, Manuel Pezzera, Renato Mainetti and N. Alberto Borghese, "Procedural constrained story generation based onPropp's and Fabula models", 2020 IEEE 8th International Conference on Serious Games and Applications for Health (SeGAH), Augusti 2020*
[109] *Christopher Antoun, Matthew Antoun, James Owen Ryan, Ben Samuel, Reid Swanson, and Marilyn A. Walker, "Generating Natural Language RetellingsfromProm Week Play Traces", 6th Workshop on Procedural Content Generation in Games, June 2015*
[110] *Sander Bakkes, Shimon Whiteson, Guangliang Li, George Viorel Vişniuc, Efstathios Charitos, Norbert Heijne and Arjen Swellengrebel ,"Challenge balancing for personalised game spaces", 2014 IEEE Games Media Entertainment, October 2014*
[111] *Muntasir Chowdhury and Clark Verbrugge, "Exhaustive Exploration Strategies for NPCs", Proceedings of 1st International Joint Conference of DiGRA and FDG 7th Workshop on Procedural Content Generation, 2016*
[112] *Clara Fernández Vara and Alec Thomson,"Procedural Generation of Narrative Puzzlesin Adventure Games: The Puzzle Dice System", Procedural Generation Content Workshop at Foundations of Digital Games 2012, May 2012*
[113] *Andreas Hald, Jens Struckmann Hansen, Jeppe Kristensen, and Paolo Burelli. 2020. Procedural Content Generation of Puzzle Games using Conditional Generative Adversarial Networks. In Malta '20: 11th Workshop On Procedural Content Generation (PCG 2020), September 15–18, 2020, Bugibba, Malta. ACM, New York, NY, USA, 9 pages.*

[114] *Leonardo T. Pereira, Claudio Toledo, Lucas N. Ferreira, and Levi H.S. Lelis "Learning to speedup Evolutionary content generation in physics based puzzle games", 2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI), November 2016*
[115] *Julian Togelius, Georgios N. Yannakakis, Kenneth O. Stanley and Cameron Browne, "Search based Procedural Content Generation", Proceedings of the 2010 international conference on Applications of Evolutionary Computation, April 2010*
[116] *Eric McDuffee and Alex Pantaleev ,"Team Blockhead Wars: Generating FPS Weapons in aMultiplayer Environment", Workshop Proceedings of the 8th International Conference on the Foundations of Digital Games, 2013*
[117] *Alex Pantaleev "In Search of Patterns: Disrupting RPG Classes throughProcedural Content Generation" Proceedings of the The third workshop on Procedural Content Generation in Games,May 2012,Pages 1–5*
[118] Gearbox Software, Borderlands 3, 2K Games, 2019 https://borderlands.com/en US/ retrived August 2021
[119] Evolutionary Games, Galactic Arms Race, 2010
[120] Turtle rock studios, Left 4 Dead 2, Valve Corporation, 2009
[121] https://web.archive.org/web/20090213180306/http://www.mojang.com/notch/mario retrived August 2021
[122] Hopoo Games, Risk of rain, Hopoo Games, 2013
[123] D. Yu and A. Hull, Spelunky, Mossmouth, 2009.
[124] The Fun Pimps, 7 days to die, The Fun Pimps, 2013
[125] Undead Labs, State of decay 2, Xbox game studios, 2018
[126] Klei Entertainment, Don't starve, Klei Entertainment, 2013
[127] Iron Gate Studio, Valheim, Coffee Stain Publishing, 2021 https://www.valheimgame.com/ retrieved August 2021
[128] Dylan Fitterer, Audiosurf 2, Dylan Fitterer, 2015
[129] Cold Beam Games, Beat Hazard 2, Cold Beam Games, 2019
[130] Procedural arts, Façade, Procedural arts, 2005
[131] Inkle, Heaven's Vault, Inkle, 2019
[132] Latitude, AI Dungeon, 2019 https://play.aidungeon.io/main/home retrived August 2021
[133] Blizzard North, Diablo 2, Blizzard Entertainment, 2000
[134] Grinding Gear Games, Path of exile, Grinding Gear Games, 2013
[135] Nicalis, The Binding of Isaac: Rebirth, Nicalis, 2014
[136] Mega Crit Games, Slay the Spire, Mega Crit Games, 2019
[137] Nolla Games, Noita, Nolla Games, 2020
[138] Brace Yourself Games, Crypt of the NecroDancer, Brace Yourself Games, Klei Entertainment, 2015
[139] Nippon Ichi Software, Disgaea 6: Defiance of Destiny, Nippon Ichi Software, 2021
[140] Bethesda Game studios, The Elder Scrolls IV: Oblivion, Bethesda Softworks, 2006
[141] Bethesda Game studios, The Elder Scrolls V: Skyrim, Bethesda Softworks, 2011
[142] Guerrilla Games, Horizon Zero Dawn, Sony Interactive Entertainment, 2017
[143] Ubisoft Montreal, Assassin's Creed Valhalla, Ubisoft, 2020
[144] Sony Online Entertainment, Star Wars Galaxies, LucasArts, 2003
[145] CCP Games, EVE Online, CCP Games, 2003

[146] Maxis, Spore, Electronic Arts, 2008
[147] Codemasters, Dirt 4, Codemasters, 2017
[148] Polyphony Digital, Gran Turismo 5, Sony Computer Entertainment, 2010
[149] Bay 12 Games, Dwarf Fortress, Bay 12 Games, 2006
[150] Ludeon Studios, RimWorld, Ludeon Studios, 2018
[151] Firaxis Games, Civilization VI, 2K Games, 2016
[152] Plausible Concept, Bad North, Raw Fury, 2018
[153] Paradox Development Studio, Stellaris, Paradox Interactive, 2016
[154] Paradox Development Studio, Crusader Kings III, Paradox Interactive, 2020
[155] TaleWorlds Entertainment, Mount & Blade II: Bannerlord, TaleWorlds Entertainment, 2020
[156] Interactive Data Visualization, Inc, Speedtree, https://store.speedtree.com/ retrived August 2021
[157] Side Effects Software Inc. , Houdini, https://www.sidefx.com/products/houdini/ retrived August 2021
[158] Adobe, Substance Painter, https://www.substance3d.com/ retrived August 2021
[159] *Ruben Smelik, Krzysztof Galka, Klaas Jan de Kraker, Frido Kuijper, Rafael Bidarra "Semantic constraints for Procedural Generation of virtual worlds", PCGames '11: Proceedings of the 2nd International Workshop on Procedural Content Generation in Games, June 2011*
[160] *Isaac M Dart, Gabriele De Rossi and Julian Togelius, "SpeedRock:procedural rocks through grammars and evolution", Proceedings of the 2nd International Workshop on Procedural Content Generation in Games, January 2011*
[161] *Lawrence Johnson, Georgios Yannakakis and Julian Togelius, "Cellular automata for real time generation of infinite cave levels", Proceedings of the 2010 Workshop on Procedural Content Generation in Games, June 2010*
[162] *Aitor Santamaría Ibirika, Xabier Cantero, Sergio Huerta, Igor Santos and Pablo G. Bringas, "Procedural Playable Cave Systems based on Voronoi Diagram and Delaunay Triangulation", 2014 International Conference on Cyberworlds, December 2014*
[163] *Dylan Robert Ashley, Valliappa Chockalingam, Braedy Kuzma and Vadim Bulitko ,"Learning to Select Mates in Evolving Non playable Characters", 2019 IEEE Conference on Games (CoG), August 2019*
[164] *André Siqueira Ruela and Frederico Gadelha Guimar˜aes, "Coevolutionary Procedural Generation of Battle Formations in Massively Multiplayer Online Strategy Games", 2014 Brazilian Symposium on Computer Games and Digital Entertainment, November 2014*
[165] *Jakub Kowalski, Radosław Miernik, Piotr Pytlik, Maciej Pawlikowski, Krzysztof Piecuch, Jakub S ̨ekowski, "Strategic Features and Terrain Generation for Balanced Heroes of Might and Magic III Maps", 2018 IEEE Conference on Computational Intelligence and Games (CIG), August 2018*
[166] *Alexandre Santos Melotti and Carlos Henrique Valerio de Moraes, "Evolving Roguelike Dungeons With Deluged Novelty Search Local Competition", IEEE TRANSACTIONS ON GAMES, VOL. 11, NO. 2, JUNE 2019*
[167] *Julian Togelius, Emil Kastbjerg, David Schedl and Georgios N. Yannakakis, "What is Procedural Content Generation? Mario on the borderline", January 2011*

[168] *Daniel Sanchez, Juan M. Solá Sloan and Elio Lozano Inca, "Procedural Generation Of Building Blueprints for Real Time Applications", Proceedings of the 2010 Spring Simulation Multiconference, SpringSim 2010, Orlando, Florida, USA, January 2010*
[169] *Luigi Cardamone, Daniele Loiacono and Pier Luca Lanzi, "Interactive Evolution for the Procedural Generation of Tracks in a High End Racing Game", Proceedings of the 13th annual conference on Genetic and evolutionary computation, July 2011*
[170] *Ricardo Lopes and Rafael Bidarra, "A semantic generation framework for enabling adaptive game worlds", Proceedings of the 8th International Conference on Advances in Computer Entertainment Technology, ACE 2011, Lisbon, Portugal, November 2011*
[171] *Lucas Ferreira and Claudio Toledo, "Generating Levels for Physics based Puzzle Games with Estimation of Distribution Algorithms", Proceedings of the 11th Conference on Advances in Computer Entertainment Technology, November 2014*
[172] *Eric Butler, Kristin Siu, and Alexander Zook. 2017. Program Synthesis as a Generative Method. In Proceedings of FDG'17, Hyannis, MA, USA, August 14 17, 2017, 10 pages*
[173] *Josep Valls Vargas, Santiago Onta˜nón and Jichen Zhu, "Towards Story Based Content Generation: From Plot Points to Maps", 2013 IEEE Conference on Computational Inteligence in Games (CIG), August 2013*
[174] *Everton Schumacker Soares and Vadim Bulitko, "Deep Variational Autoencoders for NPC Behaviour Classification", 2019 IEEE Conference on Games (CoG), August 2019*
[175] *DuMim Yoon and Kyung Joong Kim, "3D Game Model and Texture Generation using Interactive Genetic Algorithm", Proceedings of the Workshop at SIGGRAPH Asia, November 2012*
[176] *Jo Mazeika and Jim Whitehead. 2017. Solusforge: Controlling the Gen eration of the 3D models with Spatial Relation Graphs. In Proceedings of FDG'17, Hyannis, MA, USA, August 14 17, 2017, 4 pages.*
[177] *Gilet, G., Sauvage, B., Vanhoey, K., Dischler, J., Ghazanfarpour, D. 2014. Local Random Phase Noise for Procedural Texturing. ACM Trans. Graph. 33, 6, Article 195 (November 2014), 11 pages*
[178] GDCVault (March 2019), "Procedural Generation of cinematic Dialogues", https://www.gdcvault.com/play/1026381/Procedural Generation of Cinematic Dialogues, *retrived: May 2021*
[179] Hopoo Games, Risk of rain 2, Gearbox publishing, 2020
[180] Ghost Ship Games, Deep Rock Galactic, Coffee Stain Publishing, 2020
[181] Valve, Alien Swarm, Valve Corporation, 2010
[182] Insomniac Games, Spider Man, Sony Interactive Entertainment, 2018
[183] Volition, Agents of Mayhem, Deep Silver, 2017
[184] Avalanche Studios, Just Cause 2, Square Enix, 2010
[185] P Studio, Persona 5, Atlus, 2016
[186] Gunfire Games, Remnant From the Ashes, Perfect World, 2019
[187] Nadeo, Trackmania Turbo, Ubisoft, 2016
[188] Urban Games, Transport fever 2, Good Shepherd Entertainment, 2019
[189] Wube Software, Factorio, Wube Software, 2020
[190] Mojang Studios, Minecraft, Mojang Studios, 2011