MASTER THESIS

Thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Engineering at the University of Applied Sciences Technikum Wien - Degree Program Game Engineering and Simulation Technology

Using Procedural Content Generation via Machine Learning as a Game Mechanic

By: Bernhard Rieder, BSc

Student Number: 1610585006

Supervisors: Dipl.-Ing. (FH) Alexander Hofmann
Dr. Jichen Zhu
Dr. Santiago Ontañón

Philadelphia, March 14, 2018

FH University of Applied Sciences
TECHNIKUM
WIEN

Declaration

“As author and creator of this work to hand, I confirm with my signature knowledge of the relevant copyright regulations governed by higher education acts (see Urheberrechtsgesetz /Austrian copyright law as amended as well as the Statute on Studies Act Provisions / Examination Regulations of the UAS Technikum Wien as amended).

I hereby declare that I completed the present work independently and that any ideas, whether written by others or by myself, have been fully sourced and referenced. I am aware of any consequences I may face on the part of the degree program director if there should be evidence of missing autonomy and independence or evidence of any intent to fraudulently achieve a pass mark for this work (see Statute on Studies Act Provisions / Examination Regulations of the UAS Technikum Wien as amended).

I further declare that up to this date I have not published the work to hand nor have I presented it to another examination board in the same or similar form. I affirm that the version submitted matches the version in the upload tool.“

Philadelphia, March 14, 2018 Signature

Kurzfassung

Blah blah blah, das ist meine Kurzfassung über die Verwendung von Prozeduraler Inhaltsgenerierung mit Machine Learning als eine Spielmechanik, blah blah blah

Schlagworte: Prozedurale Inhaltsgenerierung, Machine Learning, Spielemechanik, Künstliche Intelligenz, Spieleentwicklung

Abstract

Blah blah blah, this is my outline about the use of procedural content generation via machine learning as a game mechanic, blah blah blah

Procedural Content Generation (PCG) is an essential topic in modern games. Notably, it is a very crucial topic for independent game developers due to a low budget, where PCG can generate much content for less effort. As the importance of PCG for game development increases, researchers explore new avenues for generating high-quality content with or without human involvement. Here is where Machine Learning comes into play and extends the capabilities of PCG. Procedural Content Generation via Machine Learning (PCGML) systems can be trained on its own and evolve if they do not generate usable output and offer a broad application possibility. One promising way of using PCGML is the use as a game mechanic. Therefore, this research will address and focus on the possibilities and development process of how PCGML can be used as a game mechanic and is going to provide a firrst demonstration of its use.

Abstracts can vary in length from one paragraph to several pages, but they follow the IMRaD format and typically spend:

- 25% of their space on importance of research (Introduction)
- 25% of their space on what you did (Methods)
- 35% of their space on what you found: this is the most important part of the abstract (Results)
- 15% of their space on the implications of the research (Discussion)

Keywords: Procedural Content Generation, Machine Learning, Game Mechanic, Artifial Intelligence, Game Development

Acknowledgements

Many thanks to mr Alexander Hofmann who gave me the possibilty to write my master thesis abroad at the Drexel University. Also many thanks to the Drexel University and Dr. Michael Wagner who gave me the opportunity to be a part of their team while i was writing my master thesis there. Lastly, many thanks to Dr. Santiago Ontanon and Dr. Jichen Zhu who supported me, ...

Contents

1 Introduction

Procedural Content Generation (PCG) is an essential and aspiring topic in modern games and is extensively used for decades (Togelius et al., 2011). Therefore, further research on different kinds of PCG is necessary to provide new exciting techniques for games development. Notably, it is especially a very crucial topic for small independent game developer studios due to a low budget, where PCG can generate much content for less effort and human resources. With this in mind, more and more storage will be available on a Personal Computer (PC) or console in the future according to Moore's Law, and it is getting hard to design a various range of content in a short amount of time. While gamer and players will be getting used to massive amounts of content because of big gaming companies which can establish a broad range of new content without the use of PCG, the small development teams will not keep up as smooth as the market leaders. Here is where Machine Learning (ML) comes into play. PCG is getting much more accessible and powerful with the help of ML which combined form the new impressive technique of Procedural Content Generation via Machine Learning (PCGML) (Summerville et al., 2017).
A PCGML system opens a lot of new possibilities due to the fact that it uses machine learning. For example, it can be trained on its own and evolve if they do not generate usable output (Summerville et al., 2017). Furthermore, the system could also be trained by some designers with unique input or by a regular user with their creative input (Summerville et al., 2017). PCGML can be used for so many aspects of a game since it can learn from simple examples and instructions. Most current work on PCGML focuses on creating designed content like unlimited amounts of unique levels (Summerville et al., 2017). But there are some open problems which needs to be addressed to utilize the whole power of PCGML. For this reason, one of an open problem is the use of PCGML as a game mechanic which is a promising approach for evolving the overall player experience in games, which could guide the games industry and development into a new future of content acquisition (Summerville et al., 2017).

1.1 Idea

PCGML is a relatively new method and technique for creating different kinds of content in modern video games for PC, gaming consoles up to mobile devices. Most current work focuses mainly on replicating designed content to provide the player with infinite and unique variations on gameplay (Summerville et al., 2017). Another great and innovative possible use of PCGML is its use as the main mechanic of a game, e.g. presenting the PCGML system as an adversary or toy for the player to engage with (Summerville et al., 2017).

The paradigm of using PCGML as a game mechanic is a relevant and promising topic which is not addressed by now (Summerville et al., 2017). Therefore, it needs detailed analysis on how it could be used best in games. For example, design of mechanics could include enticing the player to generate content that is significantly similar to or different from the corpus the system was trained on, or identify content examples that are outliers or typical examples of the system (Summerville et al., 2017). Or players could also train PCGML systems to generate examples that possess certain qualities or fulfill certain objective functions, teaching the player to operate a model by feeding it examples that shape its output in one direction or the other (Summerville et al., 2017).

Treanor et al. (Treanor et al., 2015) illustrated the following various design patterns for developing a game mechanic with Artificial Intelligence (AI) which could be used for an exemplary PCGML system: "AI as Role-Model", "Trainee", "Editable", "Guided", "Co-Creator", "Adversary" or "Spectacle". Everyone of them provide a great guiding principle for designing and implementing a PCGML game mechanic.

1.1.1 Advantages

As already mentioned, PCGML can offer an unlimited amount of content when is comes to designed content generation which is also applicable for game mechanics. There is a good amount of replay value with PCG mechanics in general due to the fact of procedural generation itself but with the help of ML this is going to increase significantly. For example, players could play a game e.g. 10 times and experience different ways of fulfilling objectives every time. In particular, players could also emerge emotional feelings for a PCGML system which is used as a trainee and remains throughout the whole game. Hence, this could create positive and magnificent memories for the players and thus for the game experience and the game itself.

1.1.2 Challenges

One of the major challenges in creating a PCGML game mechanic is the design of the mechanic which should fulfill some crucial requirements of game design to offer a good player experience. As well, the machine learning part is going to be a challenging part since it might take a lot of tweaking to get a fully working AI algorithm.

1.2 Desired Goals

It is important to note that the main idea of this master thesis is to create game mechanics which rely on the principles of PCGML rather than creating a generic PCGML game mechanic generator.

With this in mind, it is expected to provide a first insight in the use of PCGML as a game mechanic in modern games. The primary goal is to demonstrate the possibilities as well as the development process of game mechanics when it comes to the use of PCGML and also how games should work when using PCGML.

Additionally, there are some further questions which need to be addressed by this thesis. It should impart some theoretical and practical knowledge of PCG, ML, PCGML, and PCGML as a game mechanic. Talking about theoretical and practical knowledge which means that it should show how these concepts are going to be implemented from scratch and which dependencies are given and needed for a fully working implementation.

Furthermore, it should provide a good overview and function as a primer for developing proprietary PCGML game mechanics in a specific game engine or other environments. Especially, a focus on implementations in commonly used game engines is desired since most of the independent game developers are using game engines instead of creating their own engine because that is often a long process of development.

A substantial goal for this work is a fully working game with PCGML as the core game mechanic which acts as a perfect example of what is possible with this kind of functionality. It is considered to playtest the game by different kinds of people where every feedback and idea will be evaluated to increase the usability of the PCGML game mechanics. Also, since video games in general are performance-heavy applications, it should cover a performance report as a point of reference for future implementations and uses. As an additional point, it should include an outlook of the opportunities of PCGML game mechanics in future games and work, which should also function as motivation for future work in this field of research.

Generally speaking, it should be an overall guideline for bringing PCGML game mechanics into a game.

1.3 Proceedings

1.3.1 Approach

As said before, one goal of this thesis shall be the support of small and independent game developers with an introduction into PCGML game mechanics in a game engine like Unreal Engine or Unity. For doing so, it is going to address all important topics which are dependent on building PCGML game mechanics and their use in game engines. It is attempted to start with the central fields of interest like game mechanics, PCG and ML to create awareness for this topics in the a beginning. Afterwards, all the beforehand discussed topics shall be combined into PCGML and furthermore into PCGML game mechanics. In particular, theoretical usage is not only the most important subject which is the reason for providing at least some conceptual implementations on PCG, ML and PCGML. The implementation of a PCGML game mechanic with subsequent playtests as an evaluation of the concepts is also a necessary matter which should complete the introduction.

1.3.2 Agenda

The agenda will be split into two parts. The first one is a scientific-informal part about getting to know more about the foundation of PCGML and its use as a game mechanic. Since PCGML is a relatively new theme in game development, it focuses on topics regarding core knowledge of PCG and ML separately and game mechanics to act as a base for further research on PCGML as a game mechanic. Following topics shall be a part of the informal research:

- Game mechanics and their use in games.
- Necessary and important theory of PCG and ML which is dependent for PCGML with a constant focus on game mechanics, like types of PCG and some of the most used learning and training models of ML.
- The conceptual use of PCG and ML in a game engine as well as best practices, other approaches and possible issues when using PCG and ML in games and a game engine.
- Overview of possible game mechanics with PCGML.

The second part of the agenda deals with the central scientific problem of this master thesis. It addresses every aspect of PCGML and discusses how to use PCGML as a game mechanic in modern games with a focus on the maximum possible benefit for game developers. This part shall contain the following fields of research:

- Theory of PCGML and its methods in general.
- Research on different PCGML implementations and practical usage possibilities in a game engine.
- Comparison of PCGML methods regarding their use in PCGML as a game mechanic.
- Conceptual implementation of possible PCGML game mechanics in a game engine and subsequent evaluation as well as a detailed comparison.
- Development of a game with one of the best-evaluated PCGML game mechanic as the central game mechanic of the game.
- Proof of concept with playtest sessions and evaluation of its feedback.
- Research summary with meaning of PCGML as a game mechanic for the future of games.

1.3.3 Methodological Considerations

Just as important as the agenda are some methodological questions which need to be raised and answered at both research and implementation time, like:

- Which PCGML techniques are best for a game? Or which learning and training models for PCGML have the greatest advantage?

- Which programming languages fit best for the use with PCGML in conjunction with a game engine and which game engine should be used?
- Is it better to use an online or offline version of PCGML? Related to this field is the question of requirements on hardware and software regarding PCGML as well if multithreading needs to be minded.
- Which game mechanics could be implemented in PCGML and suits a game?
- What evaluation criteria shall be used for the playtesting session?

1.4 Thesis Overview

Finish and write this section afterwards the thesis is finished!

1.5 Target Group

This thesis is dedicated to advanced game developers who are interested in using PCGML game mechanics in their game. The theoretical part assumes a basic knowledge of game design and mechanics, PCG and ML since it will not be explained everything in detail. In particular, specific topics of PCG and ML which contribute to the use of PCGML as a game mechanic will be discussed and handled in more detail.
The practical part concentrates primarily on programming in different programming languages like C++ which makes it necessary for the reader to be familiar with programming. Special algorithms used thorough the chapters will be covered in detail whereas basic algorithm knowledge is assumed. Furthermore, it does not require special game development back-end skills since it addresses the use of the technique in game engines.

2 Game Mechanics

Starting this chapter with a quick insight on the Mechanics-Dynamics-Aesthetics (MDA) framework which was introduced by (Hunicke et al., 2004), helps to understand the foundation and the correlation of game mechanics in video games. In general, the MDA framework describes the division of gaming experience emergence into three dependent parts, starting with "Rules" followed by "System" and concluded with "Fun" (Hunicke et al., 2004). These fundamental parts can be represented by the designs of "Mechanics", "Dynamics" and "Aesthetics" in a game (Hunicke et al., 2004). Therefore, a large amount of gaming experience is made out of mechanics and a game will not be fun at all if their mechanics are not properly thought through even if it has amazing graphics (Adams & Dormans, 2012). Consequently, game mechanics are acting as one of the most important roles in game design which is the reason to create awareness for this topic in the beginning of the thesis.

2.1 Definition

As already indicated, a game mechanic is a main concept with many underlying sub-concepts like dynamics, aesthetics, rules, systems, processes, procedures or data which all characterize the heart of a game besides story and technology (Adams & Dormans, 2012) (Schell, 2008). It also creates gameplay and the experience of playing a game. But besides, there is no concrete definition of what a game mechanic is. Nonetheless, there are some key concepts mentioned by different game designers which contribute to an interpretation of what a game mechanic can or shall be or do:

- Defines how a game is played, their objectives can be achieved or how to lose a game. Thus, mechanics are precisely designed, detailed, specified and implemented to fulfill playability. (Adams & Dormans, 2012) (Schell, 2008)
- Often used to indicate the most influential and affecting aspect of a game which is also mostly referred as core mechanic. (Adams & Dormans, 2012)
- Enables interaction and control of game objects and elements. (Adams & Dormans, 2012)
- Mostly hidden from the player, media-independent and easy to learn. For example, rules are more considered as printed and players are aware of them because they can see or read them whereby mechanics like an enemy damage model with its damage points are hidden. (Adams & Dormans, 2012)

- A game mechanic can also be seen as a meeting point for a designers question and their provided tools for answering that question by a player. (Stout, 2015)

2.2 Types of Mechanics

It is obvious that one tries to divide possible mechanics into concrete types since of their various possibilities and shared base ideas. For this purpose, (Adams & Dormans, 2012) summarized different types of game mechanics which are mainly used in games nowadays. They first categorized them into the following five types which are listed below with some related mechanics:

- **Physics**: Motion and forces like gravity, shooting, fighting, jumping, moving, driving or any other kind of position change. (Adams & Dormans, 2012)
- **Internal Economy**: In general, all game elements which involve transaction like collecting, consuming, harvesting, buying, building, upgrading, risking or customizing of resources like currency, ammunition, portions, power ups or other kind of items. Also the use of energy, health, lives, power, points, popularity or experience and management actions for team, resources or inventory. (Adams & Dormans, 2012)
- **Progression Mechanisms**: Usually the elements or mechanisms which are controlling the players progress in the game world. For example, quests, missions, competitions, tournaments, races, challenges, levers, switches, locks, keys or special items which allow a player to defeat an AI. (Adams & Dormans, 2012)
- **Tactical Maneuvering**: Is mainly used in strategy games but also in roleplay or simulation games and often deals with the placement of elements on a map like in chess. Mechanics are for instance internal tactics where a player gains offensive or defensive advantage, also team tactics and management of resources and buildings. (Adams & Dormans, 2012)
- **Social Interaction**: Refer to rules that govern play-acting of a player or strategic actions of forming allies to defeat bosses or other allies like in roleplay games. Further mechanics would be e.g. reward of giving gifts, inviting new friends to join the game, competition between players or in particular mechanics in a co-op game where at least two players are forced to work together to achieve an objective. (Adams & Dormans, 2012)

In addition, all prior mentioned mechanics can be subdivided into discrete and continuous mechanics in terms of their internal values (Adams & Dormans, 2012). For example, internal economy is mostly discrete since it is mostly represented by a simple integer value because e.g. a player cannot pick up half of a portion — either the portion is picked up completely or not (Adams & Dormans, 2012). In contrast, continuous mechanics make use of high precision values for accuracy and is continuously calculated throughout the game like the movement of a character (Adams & Dormans, 2012).

Furthermore, every type can also be used to categorize game genres in which they are used the most. The distinction can be seen in table 1.

	Game Mechanics				
Game Genres	Physics	Economy	Progression	Tactical	Social
Action	x	x	x		
Strategy	x	x	x	x	x
Roleplay	x	x	x	x	x
Sports	x	x	x	x	
Vehicle Simulation	x	x	x		
Management Simulation		x	x	x	x
Adventure		x	x		
Puzzle	x		x		
Social Games		x	x		x

Table 1: Game Genres and their related Game Mechanics (Adams & Dormans, 2012)

But since the overview of (Adams & Dormans, 2012) is no universal taxonomy for game mechanics, there is another great approach to categorize them as described by (Schell, 2008). Following rather similar types to (Adams & Dormans, 2012)'s approach are used which also correlate to some parts described in the MDA framework:

- **Space**: Every game takes places in some kind of game spaces. Spaces can be continuous or discrete, consists of dimensions and can have bounded areas that may or may not be connected. The mechanics of Tic-Tac-Toe are a good example for this kind of mechanics which are taking place in a discrete space. (Schell, 2008)
- **Time**: Contains mechanics which are using time, clocks, races or controlled time. A popular example for this kind of mechanics is the game Superhot which tweaks the time to create a unique game experience. (Schell, 2008)
- **Objects, Attributes, States and Actions**: If these terms are compared to the structural elements of a sentence then the game objects represent the nouns, attributes and states are their adjectives and actions are the verbs of a game mechanic. This paradigm represents most of the mechanics which are used for interaction with game elements. (Schell, 2008)
- **Rules**: Combines all spaces, times, objects, actions and their consequences, constraints and the goals to form the behavior of the game. (Schell, 2008)

- **Skill**: Shifts the focus to the players and focus on their physical, mental and social skills. That means it includes mechanics like dexterity, coordination, memory, observation, puzzle solving, reading or fooling an opponent or coordinating with teammates . (Schell, 2008)

2.3 Considerations with Procedural Content Generation and Machine Learning

This chapter shall state some crucial considerations for the next chapters since PCG and ML game mechanics are not visible used in big game titles and therefore need some special attention on their implementation in a game. One of the good things is that there are dozens of possibilities for mechanics which should not create a big problem in coming up with new and novel ideas for new mechanics. With certainty, the focus of implementing such mechanics will lie on the introduction to the player and their ability for interactions due to the fact that PCG and ML mechanics could confuse some players. Therefore, the implemented mechanics should kept as easy as possible if user interaction is needed instead of creating complex but novel and unusual mechanics. A good starting point is to design the mechanics as soon as the main gameplay concept is set and adhere to the design stages of concept, elaboration and tuning during development (Adams & Dormans, 2012).

It is necessary to list some possible design flaws which need to be avoided since game mechanics shall amaze people instead of frustrate them during playing a game. In addition, a lot of detailed planning is made to come up with new extraordinary mechanics where plans about their proper introduction are missing (Pears, 2018). For this reason, it is relevant to address some common mistakes and their possible improvements:

- Do not introduce all mechanics of a game as fast as possible because players need time to learn and get used to mechanics. For this reason, just introduce one mechanic at a time! (Pears, 2018)

- Do not introduce mechanics when the player has no time to explore them. They need time in their own pace to explore the mechanics otherwise they will not enjoy their new ability. (Pears, 2018)

- Use and create feedback loops for game mechanics otherwise players will not know what to do with them. For example, if someone uses a portion and there is no obvious visualization for the use of it then the player does not know for what to use it. (Adams & Dormans, 2012)
  Sometimes feedback is one of the most important elements which can be seen in the concept of the basic grammar model introduced by (Koster, 2013). This model can be applied to most of popular games. It loops the concepts of a mental model, intent, input, actual model and rules, state change and feedback (Koster, 2013). Where the mental model of

a player assumes how a game works and what their intentions for the input and the actual input does, what then really happens with their input in terms of applying core mechanics, concluded with a feedback for their inputs (Koster, 2013). If no feedback would be given then the player could never update their mental model and cannot progress through a game. Feedback can be given in a simple binary or even complex way (Koster, 2013).

- Besides feedback loops, do not forget to provide the player with directions for parts of your mechanic which are or could not be obvious (Pears, 2018). Further tutorials should be easily accessible if they are needed because there is nothing more frustrating to a player than being confused (Doan, 2017).
- For core mechanics does apply: provide clear rules on how to be successful, create a natural interaction but do not forget to challenge the player and provide possibility for natural progression of their skills, properly guide the player towards successfully completing their in-game objectives with directions and feedback, allow the player to move naturally from objective to objective without the necessity of using the core mechanic and provide options besides the core mechanic. (Doan, 2017)
- In general, the skill of a player will grow over throughout the game which means that the difficulty curve shall match the player's skill throughout a game. (Doan, 2017)

3 Procedural Content Generation

3.1 Theoretical Introduction

Theory of PCG in general.

3.1.1 Types

(Shaker et al., 2016) methaphors for PCG in pcg book intro
- as a tool
- ...

3.1.2 Desirable Properties

(Shaker et al., 2016) (page 6 of chapter 1)
- speed
- reliability
- controllability
- expressivity and diversity
- creativity and believability

3.1.3 Taxonomy

(Shaker et al., 2016) (page 7 of chapter 1)
- online vs offline
- necessary vs optional
- degree and dimensions of control
- generic vs adaptive
- stochastic vs deterministic
- constructive vs generate-and-test
- automatic generation vs mixed authorship

3.1.4 Pitfalls and Development Considerations

3.2 Usage in Games

- mention speed tree as a tool

3.2.1 Popular Examples

For example, the open world game No Man's Sky fits over 18 quintillion planets onto a 6**GB!** (**GB!**) file with the help of PCG. https://iq.intel.com/no-mans-sky-and-the-technology-that-created-18-quintillion-planets/ https://www.digitaltrends.com/gaming/no-mans-sky-install-size/

3.2.2 Possibilities

3.3 In a Game Engine

How to use PCG and ML in a game engine? What are some of the best practices and approaches for using PCG and ML in a game?

3.3.1 Conceptual Implementation

Basic sample PCG and ML implementations in a game engine.

3.3.2 Possible Issues

Issues of PCG and ML in games and game engines.

3.4 Game Mechanics

3.4.1 Industry Examples

spelunki rogue legacy galactic arms race

3.4.2 Possible Mechanics

3.4.3 Summary

4 Machine Learning

4.1 Theoretical Introduction

Theory of PCG and ML in general.

4.1.1 Regression

4.1.2 Classification

4.1.3 Clustering

4.1.4 Reinforcement Learning

4.2 Learning Models

Which training models are used in ML? –> the 5 tribes of ML

4.2.1 Linear

4.2.2 K-Nearest Neighbor

4.2.3 Decision Tree

4.2.4 Support Vector Machine

4.2.5 Neural Networks

Artificial Neural Network

Convolutional Neural Network

4.3 Use Cases

4.3.1 Data Science

stick to game related stuff, like alpha go

4.3.2 Games

show and demonstrate current and existing work

Examples

Possibilities

4.4 In a Game Engine

How to use PCG and ML in a game engine? What are some of the best practices and approaches for using PCG and ML in a game?

4.4.1 Conceptual Implementation

Basic sample PCG and ML implementations in a game engine.

4.4.2 Possible Issues

Issues of PCG and ML in games and game engines.

4.5 Game Mechanics

4.5.1 Industry Examples

4.5.2 Possible Mechanics

4.5.3 Summary

5 Procedural Content Generation via Machine Learning

5.1 What is it About?

Theory of PCGML and its methods in general. Evaluation of PCGML hardware and software requirements

5.2 Current Use in Games

5.3 Possibilities

5.4 Difference to Usual Procedural Content Generation

5.5 Example Implementations

Theory of PCGML and its methods in general.

5.6 Learning Models

Comparison of PCGML learning models

5.6.1 Markov Chains

5.6.2 Artificial Neural Network

5.6.3 Bayes

look for a paper of "Guzdial"

5.7 In a Game Engine

research on different PCGML implementations and practical usage possibilities in a game engine

5.7.1 Conceptual Prototype

5.7.2 Possible Issues

6 Procedural Content Generation via Machine Learning Game Mechanics

Overview of possible game mechanics with PCGML.

6.1 First Considerations

6.2 Possibilities

Concepts of possible PCGML game mechanics.

6.2.1 Role-Model

A PCGML system replicates content which is generated by players of various levels of skill or generates content suitable for players of certain skill levels. New players are trained by replicating the content or by playing the generated content in the form of generative tutorial. (Summerville et al., 2017)

6.2.2 Trainee

The player trains a PCGML system to generate a piece of necessary content (e.g., part of a puzzle or level geometry). (Summerville et al., 2017)

6.2.3 Editable

Rather than training the AI to generate the missing puzzle piece via examples, the player changes the internal model's values until acceptable content is generated. (Summerville et al., 2017)

6.2.4 Guided

The player corrects the PCG system's output to fulfil increasingly difficult requirements. The AI, in turn, learns from the player's corrections, following the player's guidance. (Summerville et al., 2017)

6.2.5 Co-Creator

The player and a PCGML system take turns in creating content, moving towards some external requirements. The PCGML system learns from the player's examples. (Summerville et al., 2017)

6.2.6 Adversary

The player produces content that the PCGML system must replicate by generation to survive or vice versa in a "call and response" battle. (Summerville et al., 2017)

6.2.7 Spectacle

The PCGML system is trained to replicate patterns that are sensorically impressive or cognitively interesting. (Summerville et al., 2017)

6.3 Conceptual Prototypes (with evaluation)

A test implementation of PCGML game mechanics in a game engine.

6.3.1 Comparison of Different Mechanics

Comparison of implemented PCGML game mechanics prototypes.

6.4 Summary

what is the best game mechanic? why? etc

7 Game Prototype

Development of a game with one of the best-evaluated PCGML game mechanic as the central game mechanic of the game.

7.1 Considerations

7.1.1 Which Game Engine?

There are 2 main engines which are commonly used thorough the industry due to their fact of free to use.

7.2 Game Design

Designing Awesome AI for Games - Summary (GDC Talk)

1. Support the core experience
2. Watch people play and get in their heads
3. Identify broad behaviours
4. Start simple
5. Figure out what the brain gives you for free
6. Try going simpler before you go complex

7.3 Implementation

7.3.1 Technical Approach

Classes, Diagrams, why use what - eg. why use neural network over other and stuff like that!

7.3.2 Online or Offline?

7.4 Performance Measurements

7.4.1 Improvement Iterations

7.5 Playtest

7.5.1 Preparations

7.5.2 Evaluation

Evaluation of playtesters feedback.

7.6 Result

show the fucking game and the mechanics!

8 Conclusion

The meaning of the use of PCGML as a game mechanic for the future of games and gaming.

8.1 Summary

8.2 Research Result

8.3 Future Work

Bibliography

Adams, E. & Dormans, J., 2012. *Game Mechanics: Advanced Game Design*. 1st edition. Thousand Oaks, CA, USA: New Riders Publishing.

Doan, D., 2017. *GameDev Protips: How To Design More Meaningful And Engaging Game Mechanics*: Gamasutra. Available at: <https://www.gamasutra.com/blogs/DanielDoan/20170322/294224/GameDev_Protips_How_To_Design_More_Meaningful_And_Engaging_Game_Mechanics.php> [Accessed 12.03.2018].

Hunicke, R., Leblanc, M. & Zubek, R., 2004. MDA: A Formal Approach to Game Design and Game Research. , 1. Available at: <http://www.cs.northwestern.edu/~hunicke/pubs/MDA.pdf> [Accessed 12.03.2018].

Koster, R., 2013. *Theory of Fun for Game Design*. 2nd edition: O'Reilly Media, Inc.

Pears, M., 2018. *Game Design: Introducing Mechanics*: Gamasutra. Available at: <https://www.gamasutra.com/blogs/MaxPears/20180307/315172/Game_Design_Introducing_Mechanics.php> [Accessed 12.03.2018].

Schell, J., 2008. *The Art of Game Design: A Book of Lenses*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Shaker, N., Togelius, J. & Nelson, M. J., 2016. *Procedural Content Generation in Games: A Textbook and an Overview of Current Research*: Springer.

Stout, M., 2015. *Trinity, Part 3: Game Mechanics*: Gamasutra. Available at: <https://www.gamasutra.com/blogs/MikeStout/20150622/246683/Trinity_Part_3_Game_Mechanics.php> [Accessed 12.03.2018].

Summerville, A., Snodgrass, S., Guzdial, M., Holmgård, C., Hoover, A. K., Isaksen, A., Nealen, A. & Togelius, J., 2017. Procedural content generation via machine learning (PCGML). *CoRR*, abs/1702.00539. Available at: <http://arxiv.org/abs/1702.00539> [Accessed 12.03.2018].

Togelius, J., Kastbjerg, E., Schedl, D. & Yannakakis, G. N., 2011. What is procedural content generation?: Mario on the borderline. In: *Proceedings of the 2Nd International Workshop on Procedural Content Generation in Games*. Bordeaux, France: ACM, pp.3:1–3:6.

Treanor, M., Zook, A., Eladhari, M. P., Togelius, J., Smith, G., Cook, M., Thompson, T., Magerko, B., Levine, J. & Smith, A., 2015. *AI-based game design patterns*. Santa Cruz, CA: Society for the Advancement of Digital Games. Available at: <https://strathprints.strath.ac.uk/57219/> [Accessed 12.03.2018].

List of Figures

List of Tables

List of Code

List of Abbreviations

AI	Artificial Intelligence
ANN	Artificial Neural Network
CNN	Convolutional Neural Network
MDA	Mechanics-Dynamics-Aesthetics
ML	Machine Learning
PC	Personal Computer
PCG	Procedural Content Generation
PCGML	Procedural Content Generation via Machine Learning
SVM	Support Vector Machine