Fachhochschule Luzern HSLU

Studiengang Digital Ideation, Bachelor
4. Semester

Procedural Generation in the age of AI

Simon Hischier

May 26, 2018

Inhalt

1 Abstract

This work is juxtaposition between procedural generation with manually written algorithms and approaches with Machine Learning (ML). The question leading through this article therefore is: How does content creation in games differ between Procedural Content Generation (PCG) and ML content creation? ML and Neural Network (NN) are booming in big data and image recognition, only very few works address the artistic side like googles deep dream and this work takes a look at balancing procedural solutions and ML solutions. The work first defines a few important terms and then lists different problemsolutions with procedural and with ML examples. We defined the basic terms, discussed the desire to controll the artistic vision, worked through various examples for both approaches and listed possible solutions to some problems occuring with either procedural or ML tasks. The goal of this work is to create awareness for creative ML utilisation and to highlight challenges in both systems.

2 Definitions

2.1 Procedural Content Generation

The term PCG defines a form of content generation that is automated. The content is created through algorithmic processes and with few to no human interaction. This allows for the creation of bigger, more random looking, unique content with less to no artist interaction.[1]

2.2 Artificial Intelligence

In 1956 the Book Automata Studies[2] layed the ground work for Artificial Intelligence (AI) and the Dartmouth summer research project on artificial intelligence marked the key event "to nail the flag to the mast." McCarthy is credited for coining the phrase "artificial intelligence" and solidifying the orientation of the field[3]. The name AI was used ever since for various applications. Ever since this key event AI is defined based on the goal that is tried to beeing achieved. Bernard Marr lists them as following:[4]

1. Build systems that think exactly like humans do ("strong AI")
2. Just get systems to work without figuring out how human reasoning works ("weak AI")
3. Use human reasoning as a model but not necessarily the end goal

Marr referes with "strong AI" and "weak AI" to the paper written by John Searle where he defines a strong AI of beeing able to think and have a mind and a weak AI that can only act like it thinks and has a mind. The paper is also known for the "Chinese Room" argument[5]. More terms for a AI classification exist like Artificial General Intelligence (AGI) and the Artificial Superintelligence (ASI). These classifications are not further defined in this work.

Bernard Marr continues to list the various definitions for AI. While the dictionaries list AI as a definition, companies lack a clear definition and Marr extrapolates a definition from the companies research field. The dictionary definitions and his extrapolated definitions are listed here:

1. **The English Oxford Living Dictionary** "The theory and development of computer systems able to perform tasks normally requiring human intelligence, such as visual perception, speech recognition, decision making, and translation between languages."

2. **Merriam-Webster**

   (a) A branch of computer science dealing with the simulation of intelligent behavior in computers.

   (b) The capability of a machine to imitate intelligent human behavior.

3. **The Encyclopedia Britannica** "artificial intelligence (AI), the ability of a digital computer or computer-controlled robot to perform tasks commonly associated with intelligent beings."

4. **Amazon** defines it as "the field of computer science dedicated to solving cognitive problems commonly associated with human intelligence, such as learning, problem solving, and pattern recognition."

5. **Google AI** "create smarter, more useful technology and help as many people as possible"

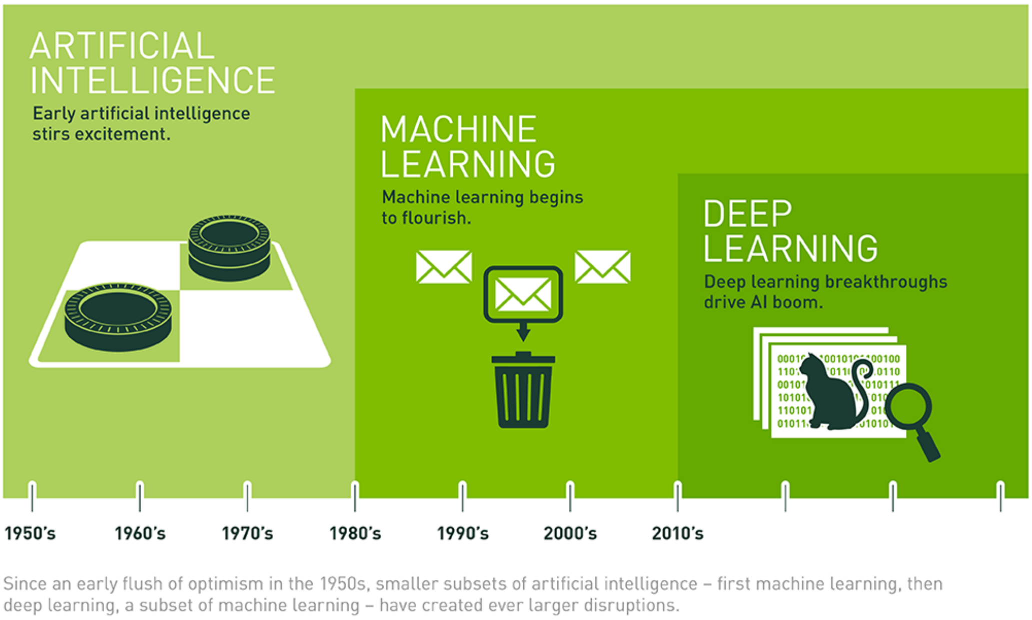
6. **Facebook AI Research** "advancing the file of machine intelligence and are creating new technologies to give people better ways to communicate."

7. **IBM**'s three areas of focus are "AI Engineering, building scalable AI models and tools; AI Tech where the core capabilities of AI such as natural language processing, speech and image recognition and reasoning are explored and AI Science, where expanding the frontiers of AI is the focus."[4]

This work is using the definition of the term AI as listed in *The Encyclopedia Britannica*. The fields described in this work can be categorized in the 3. objective listed by Marr "Use human reasoning as a model but not necessarily the end goal"[4]

2.3 The Difference between AI, Machinelearning, Deeplearning

As shown before, AI is not a new phenomena and even ML is known for quite some time. There is not a great difference between AI, ML and deep learning. They are subsets of the previous. While AI still is a very broad term ML and deep learning are closer together. Deep learning introduces neuron simulation to ML by creating discrete layers, connections and directions of data propagation[6]. The technology started to grow significantly around the year 2010. This seems to be related to the growing computational power and the immense amount of collected and stored data[6]. The illustration is showing the relationship between the 3 terms.



§

Figure 1: Levels of AI as Image[6].

2.4 Often used Neural Networks

2.4.1 Convolution Neural Network

The Convolution Neural Network (CNN) describes a method to reduce input data and enhance NN performance. The convolution allows for faster calculations and reduces memory requirements. This advantages make CNN very desirable for image recognition and voice recognition tasks and are widely used in these fields.

2.4.2 Recurrent Neural Networks

As explained in subsection 2.3 ML is a broad term and includes a variety of models. The Recurrent Neural Networks (RNN) are networks for tasks where we need some kind of persistance. If we want to classify videoframes the network should have some kind of consistency.[7] A network should persist the last seen data and not reclassify items every frame. Reclassifying without previous context could lead to different recognitions in every frame for the same object.

2.4.3 Long short-term memory

RNN are good for persisting very recent information. Sentences are a great example: "Ships are built to float on *water*". The RNN is great in filling the end of this sentence. Problems arise when the information is needed a lot later. The more information is inbetween the contextual references the more unreiable a basic RNN gets. Books for example can have references on the last page to the very beginning. For such tasks a Long short-term memory (LSTM) model is the perfect fit. The LSTM network was introduced by Hochreiter Sepp and Urgen Schmidhuber[8]. A LSTM is a specialized version of a RNN which is designed for these kind of tasks. Almost all RNN tasks can be achieved with a LSTM RNN[7].

3 The artistic vision and the generation

Generating content for games is a fundamental artistic choice for game developers. The generation in various forms is linked to a decrease in the artistic vision. Designers have to step away from micro-controlling game parts like environment, shapes, colors, enemy behaviours etc. Therefore games do include PCG in different ways and in various depths. Big studios tend to stick to more controlled experiences and have more (human-)resources to ensure this vision. We define a list of various depths of PCG:

1. **No generation** An example here might be *Super Mario Bros* (Nintendo, 1985) where everything was handplaced, drawn and animated. Mr. Miyamoto explains some details of the level creation in *Super Mario Bros.* Level 1-1[9].

2. **Content generation in the game making phase.** ex. *The Elder Scrolls Oblivion* (Bethesda Softworks, 2006) used PCG to generated most of the world before the artists curated it[10]. An example of a widely used PCG algorithm middleware for game studios is SpeedTree[11].

3. **Gameplay (partially) definded or influenced** by PCG such as the sidequests for *The Elder Scrolls Skyrim* (Bethesda Softworks, 2011) which were endlessly generated[12] or Castles in *Rogue Legacy* (Cellar Door Games, 2013) which are generated procedurally but the game has some kind of continuosity and progress on top of the castle runs[13].

4. **Games almost completely generated** ex. *Dwarf Fortress* (Dwarf Fortress, 2006) doesn't stop at the map generation. It starts out generating the history of this world and everything that happened before[14].

Games and even game genres do fall into these different levels of PCG. A major role for this classification of games and game genres is the depth of artistic control or lack therefore.[1] Games that do rely more on PCG tend to focus more on the fun gameplay rather then an intriguing story and complex characters. Further categorization and different methods of PCG can be found in the Paper from van der Linden, Lopes and Bidarra[1].

4 Procedural Content Generation and AI

As explained in The artistic vision and the generation there are various genres and games falling into different levels of PCG. Due to the type of gameplay and game mechanics linked to PCG, games with high levels of procedural generation can easily be identified. We hope AI blurs the line between PCG and handcrafted more. A goal for AI components is to create PCG games that are less distinguishable from handcrafted counterparts. By extending the PCG with AI we hope to have a more natural and handcrafted feel to PCG type games. As ML gets used more in games, the applications starts to vary further.

5 Applications of Procedural Approaches

Procedural generation of content varies wildly and can affect most gameparts. In The artistic vision and the generation we made an attempt to categorize the different stages and depths of automation. However the reason to use PCG can range from necessity (memory limitations, more content to create then time to create it) to artistic choice (pattern and style generation). The PCG algorithms allow to shift the artists focus away from the detail work to broader concepts. An example might be the world biomes in *Minecraft* (Markus Persson, 2009) instead of exact placements of blocks and levelparts. PCG has multiple facettes and listing the different reasons to use PCG is not feasable in this work. What all procedural algorithms share in common is a tradeoff between disk space and computing time. A big advantage of PCG is the great influence on the outcome and the artists controll over what exactly gets generated. With PCG algorithms the artist has complete controll over the process of generating the drawback for this is an increasing amount of programming work and complexity. Another interesting property of PCG is the curiosity involved in finding out what the algorithm generated. Eventhough the artist has complete controll over **how** content is generated they don't controll **what** exactly is generated, leaving said room for curiosity. Some unsorted and randomly picked examples of PCG widely used are listed below.

5.1 Level Generation

Procedural level generation is around for a very long time. It was present during the arcade games because of the need to possibly play endlessly to get the high score. It got carried over to the personal computer, generating dungeons to provide unique experiences[1] and to counter floppy disk memory limitations. Currently one of the most known level generators is used in *Minecraft* (Markus Persson, 2009) where the endless worlds are used to feed the players curiosity. The strength of these systems is that worlds are replicable (see world-seeds in Minecraft), can be controlled and fine-tuned and don't necessarily need a lot of computational power. They are expandable, maintainable and errors in the level generation process can be checked and fixed. A drawback is the complexity (or lack therefore) of such generators. Worlds can look repetitive and empty.

5.2 Text to Speech

Procedural approaches of text to speech up until now were more on the functional side. Text to speech has great challenges and voices can sound very monotone, lack highlights and are in general not used in games. Games either hire voice actors to read the text examples or display the text as is.

6 Applications of Neural Network systems

In comparison to the procedural approach of extending the algorithm the NN approach is a very top-down approach. The NN needs a lot of computing resources and a great set of datapoints. ML does need a lot more computational power then PCG but those numbers are rapidly falling as shown by AlphaGo Zero[15]. The user does not have to (neither can) finetune the parameters for the generation directly as is with the PCG. The users can choose different NN types and can only control the various ML controllers which come with the NN. There is no direct influence on the generation itself, the output can be changed by letting the ML model guess and finetune its weights until it gets the desired outputs. This changes the workflow significantly and to use the network a user has to know what outcome is desired. This stands in direct opposition to procedural algorithms where the output is a result of the written code. With NNs the artists decides on what the concept is which the NN should learn and he then teaches the NN to follow the concept. Testing such a NN and evaluating corner cases is very labor intensive and debugging is not a real option. The big drawback when creating a new model is the need for existing data. The procedural algorithms can be programmed to work with a random number and data can optionally be fed into the algorithm to shape the outcome while developing. For a NN to generate data in the desired way one has to first create a lot of examples that look like the desired outcome to train the network with. This is the biggest difference when working with NN instead of procedural algorithms. To show a few examples of various NN we list examples where the NN AI performs great. Note that the examples represent a small field of applications and the examples are randomly picked. This field is under extensive research and there are significant parts left out like image combination[16], style transfer, evolution of objects and many more.

6.1 Face representation in games

An area where AI is much more effective then handwritten procedrual approaches is face reconstruction and face mapping. Webcam feeds have no depth information and creating 3D faces from photos is a laborintensive work. Actual research shows ML is capable of reconstruction and position maping 3D avatars from single images. It's robust, fast and stable[17]. A large area in the games industry is avatar faces. Massive Multiplayer Games tend to have elaborate avatar creaton tools to customize the game character. Games like *Arma* (Bohemia Interactive, 2013) are using the users voice in multiplayer sessions and try to synchronize the avatars lips with the players spoken words. Online chats with avatar representations such as *VRChat* (VRChat Inc., 2017) have immersive VR worlds where players can walk and talk in a virtual environment. For streaming platforms like *Twitch* (Amazon, 2011) a webcam feed is part of the majority of streams. All these different games, genres and platforms would benefit from some kind of reconstruction of the players face. *(*Arma) could have the real users faces represented in real-time in the game, non human avatars could use AI to map the users lips and expressions on to the virtual avatars. Streamer could have their faces as 3D models or have an avatar instead of a live camera feed.

6.2 text to speech

Text is a big part of games. Extreme examples of games with word counts comparable to The Lord of the Rings book trilogy are present in the Role Play Game (RPG) and text adventure genre. For this games voicing every line is out of range for game studios and therefore players have to read most of the text. With a trained NN it is possible to have a voice actor read example texts and let the NN generate the voice from the written text in the game[18]. Although a demonstration used 24.6 hours of training data this number can probably be reduced. Another paper shows, that even singing can be generated with 16 and 35 minutes of sentences read out[19]. The later audio example lengths could potentially open doors for further applications of text to speech such as online chat using the users voice to generate spoken words for the other users.

6.3 Game AI

Something almost all games have is computer players controlled by AI. There are a variety of different solutions to mimic human behaviour. The more complex a game is the harder it is to write comprehensible game AI. Games like Go remain too hard to calculate a winning strategy. There are simply too many complex situations and too many possible paths to explore. In this field AI really showed its power and Alpha Go defeating the world champion was a major step in AI[20][15]. While Alpha Go did need more hardwarepower then the usual gaming computer provides, OpenAI demonstrated a less powerhungry AI for *Dota 2* (Valve Corporation, 2013). At the International Tournament 2017 OpenAI demonstrated an AI that beat Pro Players at the game reliably.[21] The rules were more strict then a normal game of Dota 2 but given enough time and resources, an AI that can beat Professional teams is likely. These types of games are a big challenge to program computer players for because they allow for wacky situations, situations never seen before and asymmetric or incomplete information about the other players. A strong field for ML is creating AI that always remains and adapts to the level of the human player. This is a colossal task and almost impossible to program by hand or with other approaches.

6.4 Anti-Cheating

Valve demonstrated a very significant increase in Anti-Cheat software effectiveness after they started to use deep learning. Old methods of hardcoding different checks did and does only lead to an arms race with the cheat providers. The new approach requires the company to maintain a powerfull server. A big drawback is, that the company has to own a more powerfull NN and have more complete information to not get outsmarted by cheat providers[22].

7 Level Generation compared

Level generation can be done via ML as shown by Adam Geitgey[23] or PCG depending on the desired output. Both approaches could potentially be combined for a wider variety. What both approaches lack is the carefully curated and manually designed human aspect of levels. The amount of detail and teaching in a single *Super Mario Bros.* (Nintendo, 1985) level[9] is still not feasible by both approaches.

8 Conclusion

In this work we learned about different applications for ML and PCG algorithms. The ML field is still under heavy research and most of the ML research is outside of the field of creative content generation. As ML gets more popular more fields will be researched. From the different usecases we learned that ML is a top-down approach to generate content and the PCG is a bottom-up approach to generate content. This showed, that ML takes a big upfront work before it can generate usefull gameparts and users have to know upfront what they want. Further the goal of ML was not to use as little hardware as possible which is one of PCGs strengths. The different strengths of PCG and ML suggest, that ML is an extension in the content generation toolbox rather then a replacement. ML is opening up a lot of new fields to content generation where classic PCG was insufficient. The computing requirements show, that some NN can operate in realtime already and that other NN can still be heavily optimized as the Alpha Go team showed. But at the moment most ML applications in games will still be used during the production of the game instead of during the playtime. Both approaches still lack the thought through creative possibilities possible by humans as explained with the *Super Mario Bros* level. ML is very hard to control and finetune and therefore depending on the area of application is a lot more labor intensive.

9 Referenzen und Akronyme

Acronyms

AI Artificial Intelligence. 2, 3, 5, 7, 8

CNN Convolution Neural Network. 4

LSTM Long short-term memory. 5

ML Machine Learning. 2–8

NN Neural Network. 2, 4, 6–8

PCG Procedural Content Generation. 2, 5, 6, 8

RNN Recurrent Neural Networks. 4, 5

RPG Role Play Game. 7

References

[1] Roland Van Der Linden, Ricardo Lopes, and Rafael Bidarra. Procedural generation of dungeons. *IEEE Transactions on Computational Intelligence and AI in Games*, 6(1):78–89, mar 2014.

[2] John McCarthy and Claude Elwood Shannon. *Automata Studies: Annals of Mathematics Studies. Number 34 - William Ross Ashby*. Princeton University Press, 1956.

[3] James Moor. The Dartmouth College artificial intelligence conference: The next fifty years. *Ai Magazine*, 27(4):87, 2006.

[4] Bernard Marr. The Key Definitions Of Artificial Intelligence (AI) That Explain Its Importance, 2018.

[5] John R. Searle. Minds, brains, and programs. *Behavioral and Brain Sciences*, 3(03):417, sep 1980.

[6] Michael Copeland(Nvidia). The Difference Between AI, Machine Learning, and Deep Learning? — NVIDIA Blog, 2016.

[7] Nery Sofiyanti, Dyah Iriani Fitmawati, and Andesba A. Roza. Understanding LSTM Networks, 2015.

[8] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 1997.

[9] Eurogamer. Miyamoto on World 1-1, 2015.

[10] Bethesda Game Studios. The Elder Scrolls 5: Skyrim, 2011.

[11] SpeedTree. SpeedTree Vegetation Modeling.

[12] Matt Bertz. The Technology Behind The Elder Scrolls V: Skyrim, 2011.

[13] Rich Stanton. The making of Rogue Legacy • Eurogamer.net, 2013.

[14] Alex J. Champandard. Beyond Terrain Generation: Bringing Worlds in DWARF FORTRESS to Life — AiGameDev.com, 2012.

[15] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. Mastering the game of Go without human knowledge. *Nature*, 550(7676):354–359, oct 2017.

[16] Fujun Luan, Sylvain Paris, Eli Shechtman, and Kavita Bala. Deep Painterly Harmonization. apr 2018.

[17] Yao Feng, Fan Wu, Xiaohu Shao, Yanfeng Wang, and Xi Zhou. Joint 3D Face Reconstruction and Dense Alignment with Position Map Regression Network. mar 2018.

[18] Jonathan Shen, Ruoming Pang, Ron J. Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, RJ Skerry-Ryan, Rif A. Saurous, Yannis Agiomyrgiannakis, and Yonghui Wu. Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions. dec 2017.

[19] Merlijn Blaauw and Jordi Bonada. A Neural Parametric Singing Synthesizer. apr 2017.

[20] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, jan 2016.

[21] Openai. More on Dota 2, 2017.

[22] John McDonald. GDC Vault - Robocalypse Now: Using Deep Learning to Combat Cheating in 'Counter-Strike: Global Offensive', 2018.

[23] Adam Geitgey. Machine Learning is Fun! – Adam Geitgey, 2014.

List of Figures

List of Tables