

**PROCEDURALLY GENERATING NEW GAME MECHANICS TO PREVENT
EXHAUSTING GAME REPETITION AND CREATE PERCEPTUAL
UNIQUENESS IN A ROUGELIKE GAME**

**BY
CHRISTOPHER TERRY**

**A Thesis Submitted in Partial Fulfillment of the Requirements
For the Degree of Bachelor of Science in The
Department of Mathematics and Computer Science**

Clafin University

**Orangeburg, South Carolina
Month Year**

**Dr. Ramaier Sirarm
Thesis Advisor**

**Enter Faculty Name Here
Committee Member**

**Enter Faculty Name Here
Committee Member**

**Dr. Jessica Myles
Faculty Liaison**

**Received by: _____ DATE: _____
Dr. Karina Liles, Chairperson, Department of Mathematics and Computer
Science**

**Received by: _____ DATE: _____
Dr. Verlie Tisdale, Acting Dean, School of Natural Sciences and Mathematics**

**Received by: _____ DATE: _____
Dr. Verlie Tisdale, Vice Provost for Academic Affairs**

ACKNOWLEDGEMENTS

[the Acknowledgments page is optional. It is a nice place to thank the faculty, friends, and family members who have helped you reach this point in your academic career. This section should be no more than one page in length.]

THESIS STATEMENT

“So your algorithm may generate 18,446,744,073,709,551,616 planets. They may each be subtly different, but as they player is exploring them rapidly, will they be *perceived as different?*” Compton (2016) This is a question refereed to the “10,000 Bowls of Oatmeal” problem that plague games that use procedural generation. What is procedural generation? The very simple definition is computer generated content through the use of algorithms. An online article from MIT Van Brummelen (2020) uses the game “No Man’s Sky” where the players can explore 18 quintillion (18,000,000,000,000,000,000) unique planets and moons as an example. Players lose interest in a game when there is no reason to keep playing. Too much predictability and no variety might lead to player boredom over repetition and optimizing the fastest way to beat these challenges as they become less of a challenge and more an inconvenience to the player discourage them from experimenting. Video games with random probabilities could take away predictability and discourage players from taking risks because of the chance of failure. These are arguments against procedural generation as it does might not challenge veteran players or frustrate new players. One of the most punishing and sometimes unfair video game genre that uses procedural generation like a mechanic are rouge-likes. Inspired by the game of the same name, Rogue, they are categorized as dungeon crawlers that procedurally generate dungeon levels with layouts changing with all items and enemies being randomly placed within those layouts furthermore having the player permanently die and have them start from the beginning of the game. The challenges of most rouge-likes is make fair deaths and offering new challenges when players start new runs or

continuing runs. The major criticisms of the genre is the repetition and randomness. Randomness comes from the level design of enemy, room, and item placement being different; moreover, repetition is provided by level structure as the player will transverse the same environments throughout multiple runs. To quote a game design Reddit comment by turtle_dragonfly “Lots of variety, but not *interesting* variety.” This is directly referencing “10,000 bowls of oatmeal” problem from “So you want to build a generator...” tumblr article by Kate Compton(2016). The article uses the metric of perceptual uniqueness and differentiation to describe why most games with procedural generation can fail to be “memorable.” This study aims to solve the issue of fixing the problem of perceptual uniqueness and differentiation in rouge-like games and there procedural generation without making everything in the game random lacking cohesion and keeping enough predictability so that players can still reach the goal of the game. Code will be modified from watabou’s Pixel Dungeon, a “Traditional rouge-like game with pixel-art graphics and simple interface. The question that should be asked Is there a way to change procedural generation so

ABSTRACT

This study aims to answer the question of whether or not a version of procedural generation can be used to create content with perceptual uniqueness in rogue-like games. Procedural generation is the use of algorithms to make new content for video games and other computer art media. Perceptual Uniqueness is how different is some video game element like characters and items is that it is completely different from one another. The roguelike game genre are games based off Rogue that creates procedurally generated dungeons that the player must transverse to the final level with the risk of dying and starting from the beginning. This study plans to modified a fork of Pixel Dungeon, a traditional pixel-art roguelike game, to experiment and showcase more forms of procedural generation.

KEYWORDS AND ABBREVIATIONS

[if needed, list any acronyms or abbreviations you use in your text]

Think about keywords or abbreviations that are not common knowledge to the general public.

Rogue – AKA Rogue: Exploring the Dungeons of Doom – An RPG Dungeon crawling video game developed around 1980 for Unix-based minicomputers

RPG(s) – Role-Playing Game(s)

Roguelikes – A style of RPG that is characterized by dungeon crawling through procedurally generated levels with permanent death that would require players to start from the beginning of the game with the philosophy that every run (

Procedural Generation – The use of algorithms to generate new content for video game and other computer art media.

Perceptual Uniqueness - Feeling that a piece of content is not identical to the last on memorable components (Players, Important characters, important places, Enemies, Enemy Bosses)

Perceptual Differentiation – Feeling that a piece of content is not identical to the last on non-memorable components (Trees, Doors, furniture, etc.)

Video Game Mechanics – rules that the player and enemies have to follow during gameplay

LIST OF TABLES

Table 1. A Sample Table Showing Correct Formatting	5
--	---

Follow this format listing the caption and the page number for which the table can be found in your document. This page is optional and dependent upon if you have tables or not.

LIST OF FIGURES

Figure 1. Figure caption goes here.....xx

Follow this format listing the caption and the page number for which the figure/image can be found in your document. This page is optional and dependent upon if you have figures/images or not.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	i
THESIS STATEMENT	ii
ABSTRACT	iv
KEYWORDS AND ABBREVIATIONS	v
LIST OF TABLES	vi
LIST OF FIGURES	vii
TABLE OF CONTENTS	viii
INTRODUCTION	1
Background of the Study	3
Purpose of the Study	4
Research Question(s)	4
LITERATURE REVIEW	5
Summary and Conclusions	8
METHODOLOGY	9
Population and Sampling Procedure	4
Research Design and Development Procedures	9
Instrumentation (if applicable)	9
Data Analysis Plan (if applicable)	9
Limitations	10
Ethical Procedures & Considerations	10
Summary	10
RESULTS	11

Study Results	11
DISCUSSION AND CONCLUSIONS	12
Summary of the Findings	12
Global Impact of Computing Solution on Individuals, Organizations & Society	12
Recommendations	12
Conclusions	12
REFERENCES	13
Appendix A: Title of Appendix	16

INTRODUCTION

Video games are very predictable after long hours of play that replayability can be a chore.. Video games are generally made for players seeking fair challenges that are designed to have enough predictability that players can solve through trial and error. Too much predictability might lead to player boredom over repetition and optimizing the fastest way to beat these challenges as they become less of a challenge and more an inconvenience to the player discourage them from experimenting. Video games with random probabilities could take away predictability and discourage players from taking risks because of the chance of failure. These are arguments used for and against procedural generation in video games as it is considered unpredictable. What is procedural generation? The very simple definition is computer generated content through the use of algorithms. An online article from MIT uses the game “No Man’s Sky” where the players can explore 18 quintillion (18,000,000,000,000,000,000) unique planets and moons. One of the most punishing and sometimes unfair video game genre that uses procedural generation like a mechanic are rouge-like. Inspired by the game of the same name, Rogue, they are categorized as dungeon crawlers that procedurally generate dungeon levels with layouts changing with all items and enemies being randomly placed within those layouts furthermore having the player permanently die and have them start from the beginning of the game. The challenges of most rouge-like is make fair deaths and offering new challenges when players start new runs or continuing runs. The major criticisms of the genre is the repetition and randomness. Randomness comes from the level design of enemy, room, and item placement being different; moreover, repetition is

provided by level structure as the player will transverse the same environments throughout multiple runs. To quote a game design Reddit comment by turtle_dragonfly (2023) “Lots of variety, but not *interesting* variety.” This is directly referencing “10,000 bowls of oatmeal” problem from “So you want to build a generator...” article by Kate Compton (2016). The article uses the metric of perceptual uniqueness and differentiation to describe why most games with procedural generation can fail to be “memorable.”

Perceptual uniqueness and differentiation is not only used in games with procedural generation but can be metric for many other games for longevity and replayability. A more recent video game that has failed this metric by the general public is Suicide Squad: Kill the Justice League being panned for repetitive missions and uninteresting endgame challenges that many people stop playing at that point or never got there and refunded the game. There was no reasons for players to come back because they believe they have seen everything the game have to offer. As quoted from IGN’s Cardy (2024) “Suicide Squad: Kill the Justice League is a repetitive and bland looter-shooter that, despite an engaging story, never stays fun for long enough.” Another reason it might have failed is the premise of the game stated in the title; the Suicide Squad, a group of super criminals, kills the Justice League, a group of superheros consisting of beloved characters such as Superman, Batman, and The Flash. There has to be a reason for players to keep playing that game after finishing it and the game must look appealing because a good gameplay loop, the main thing player do from the start of the game to the end, can only keep certain players invested to keep playing. The problem is not variety but the differences in variety that would keep players engaged to discover more. It is like going to restaurant to eat a

new menu item however it tastes like another menu item. There has to be something that keeps customers wanting to come back for more.

Background of the Study

Rogue: Exploring the Dungeons of Doom is a dungeon crawler video game made by Michael Toy and Glenn Wichman with later credits to Ken Arnold. The term dungeon crawl stems from a scenario mostly found in fantasy table top role-playing games, such as dungeons and dragons, introduced as far back as 1975 with Gary Gygax's Solo Dungeon Adventures. Dungeon crawl have the characters navigate a labyrinth where they will battle various monsters, avoid traps, solve puzzles, loot treasures. Rogue was developed for Unix-based minicomputers in 1980 and later ported to other systems with other modern ports being design by other parties with open-source code. The game allows players to control a character as they explore several levels of a procedurally generated dungeon seeking the game win condition the Amulet of Yendor on the bottom level of the dungeon. Along the way, players must fend off monsters while collecting *treasures* or items "such as weapons, armor, potions, scrolls, and other magical items." The game is turn-based on a square tile grid giving players time to determine the next best move for survival. The game philosophy is that every run is different that even the developers of the game might not be able to beat it first try. This game was so popular that it created a niche game genre that has had different takes on this philosophy of everybody's game is different. Brewer, N. (2017).

In 2010, The open-source pixel dungeon was released on google play by developer watabou written in java and categorized as a "Traditional roguelike game with pixel-art graphics and simple interface." Pixel Dungeon is said by the developer to be

inspired by Brogue, a roguelike with more visual 26-level and easy to pick up version of Rogue. Pixel Dungeon had inspired multiple forks of the game including the popular Shattered Pixel Dungeon, arguably considered the best version because of it balancing and consistent updates. The problems with all roguelikes is the repetitive formula that comes with the idea of permanent death and lost of in game progress to start from the very beginning. Watabou. (2014)

Purpose of the Study

The purpose of this study will be finding a way to make more perceptual uniqueness in a rouge-like game that would encourage players to continue playing a new playthrough or run; additionally, clear predictability for every run so players are not confused or grow frustrated enough to stop playing the game.

Research Questions

The research questions guiding this study are:

- Can Perceptual uniqueness be created procedurally?
- Could there be a way to make the death penalty in rouge-likes less punishing?
 - Could there be a new rule (mechanic) that can be added to allow players to continue?
- Could new challenges for the player be procedurally generated and add perceptual uniqueness?

LITERATURE REVIEW

The issue with most rouge-like game whether they are traditional or not is the lack interesting variety that can keep players playing for hours and fairness concerning the randomness of levels. It is the sense that once player have found the formula on how the game works it will become repetitive and boring to replay. I wanted to know why procedural generation can fail in this regarding replayability and unique playthroughs.

So you want to build a generator...

This is a blog essay that is advice for making procedural generators. The essay starts with the question “What are you making?” If the answer is a video game than you must break down elements of that game to find out what is really being generated. Those elements are refereed to as “artifacts” in the essay. Artifact examples include “Procedural birds, generated stories, animated dance choreography, gazpacho recipes, RPG quests, chess variants. “Good artifacts are concrete properties” with set parameters while “bad artifacts” have items and attributes that break it.

Using Procedural Content Generation via Machine Learning as a Game Mechanic

A solution to solving the issue could be making a procedural generator and developing it as a game mechanic. What is a game mechanic? Game mechanics are the rules that governs the actions of a player and the game’s response to those actions. In a blog on the medium from Trinh (2022) “The 4 Elements of Game Mechanics”, there are four main elements to understanding the framework of game mechanics:

- Quantity – Mechanics represented by a number, i.e. Resources, Currency, and Time

- Spatial – Mechanic representing the affect on space, i.e. World positioning, Tangible Collision for characters and props, and Intangible: Inventory or storage
- State – Mechanic representing additional rules sets such as limits or improvements to Player movement, it includes win conditions and losing conditions or if an object is off or on.
- Action – Mechanics represent change this include resource replenishment and consumption or Unlocking a door or opening a locked chest or how characters move in the world.

Trinh uses this framework to describe a health system from Nintendo's Two-Dimensional Super Mario Games.

“Big or small are state mechanics that (1) apply to Mario, (2) visually represent Mario's health points (quantity), and (3) affect how tall Mario is when he is standing (spatial). When Mario collides with another entity (spatial), that entity ay change Mario's health state to big, to small , or to dead (action).”

I want to use these elements to describe some of the game mechanics in the rouge-like game Pixel Dungeon. A lot of mechanics that appear in Pixel Dungeon also appears in other rouge-likes including the original Rogue and other turn-based dungeon crawlers. The first system is the nutrition or hunger system that effects the player health. There are three states that the player can be in with this system: Not Hungry, Hungry, and Starving. Nutrition is hidden to the player but this is a Hungry is a warning state to let the

player know to have food, a quantity element, available in storage, a spatial element. If the player does nothing about the hunger state after couple more turns, they will enter the state of starving which make the player lose 1 health every 33.33 turns average and natural regeneration, an action regaining health quantities without get hit by enemies or other damaging objects, is halted until the player eats healthy food. This will force players to manage inventory and hunger to avoid this debuff, negative modifier, mechanic. I have only described one mechanic in a plethora of other mechanics found in the game. If I were to give this mechanic a type based Adams & Dormans five types explained in Rieder (2018) *Using Procedural Content Generation via Machine Learning as a Game Mechanic* of Physics, Internal Economy, Progression Mechanisms, Tactical Maneuvering, and Social Interaction. This mechanic is a debuff mechanic that effects health so it would fall under Internal Economy as it involves health and inventory, since it involves “transaction like collecting and consuming.” For Physics of motion and forces, we can include the turn system since every player action revolves around turns. Progression Mechanisms, “controlling the players progress in the game world” would be the hero leveling mechanic, the strength leveling mechanic, and the item discovery [mechanic in most traditional roguelikes where the player must find out what a consumable item does]. In Pixel Dungeon, Tactical Maneuvering could include the ability to throw items near parts of the environment, planting traps for enemies, or surprise attacks which require positioning to catch an enemy off guard to have the chance to do more damage. Finally, Social Interaction, “rules that govern play-acting of a player”, should be the merchant that shows up after every boss floor in a room near the entrance of the first of every new dungeons stage.

In the third chapter of *Procedural Content Generation in Games: Constructive generation methods for dungeons and levels*, there is a full explanation for how most traditional roguelikes, adventure games, and RPGs decide to procedurally generate dungeons.

“Procedural generation of dungeons refers to the generation of the topology, geometry, and gameplay-related objects of this type of level. A typical dungeon generation method consists of three elements:

1. A representational model: an abstract, simplified representation of a dungeon, providing a simple overview of the final dungeon structure.
2. A method for constructing that representational model.
3. A method for creating the actual geometry of dungeon from its representational model.”

The chapter is about most traditional dungeon crawlers however it leads a discussion around other procedural generation games including *InfiniTux*, *Infinite Mario Bros* open-source game and the commercial rogue-lite, rogue-like-like, platformer game *Spelunky*.

Shaker, N., Togelius, J., & Nelson, M. J. (2016),

Summary and Conclusions

To conclude these thoughts, the major flaw in procedural generation is change between multiple playthroughs. If players want to keep playing a game there has to be a driving force of change after multiple deaths in game or multiple clears of the game.

METHODOLOGY

The purpose of this study will be finding a way to make more perceptual uniqueness in a rouge-like game that would encourage players to continue playing a new playthrough or run; additionally, clear predictability for every run so players are not confused or grow frustrated enough to stop playing the game.

Research Design and Development Procedures

The plan for this project is creating a modified version of watarbu's Pixel Dungeon and create a version of the game with enough perceptual uniqueness and differentiation that no runs can feel the same. However I do not want make the game too random for player that runs are unbeatable and not too predicable that it becomes boring and repetitive. I want to see how to improve procedural generation to prevent the same patterns showing up.

Instrumentation

I Plan to Fork and experiment with Watabou's Pixel Dungeon Source Code

For published instruments, provide:

- Name of developers and year of publication
- Pixel Dungeon Watabou (2015)
- Appropriateness to the current study
- Will be use to experiment with new techniques of procedural generation

Data Analysis Plan (if applicable)

Data might be collected by survey so see how much of the game has noticeable repetition after players have played a couple of runs or have beaten the game?

Limitations

The limitations of this project are the assets found in the original pixel dungeon and the java code base for the game. A question to be asked is whether or not procedural generation can be change to add more perceptual uniqueness that would help to keep players invested every new run.

Ethical Procedures & Considerations

Legal Issues: There should be any legal issues since I am using open-source code for this project since Pixel Dungeon uses the GNU General Public License Version 3 giving me the right to redistribute as long as I credit the original author and not pay wall the free software.

There are no ethnical or security issuses that could be raised since the project is open-source. The survey would be optional and would not effect results of the project.

RESULTS

Study Results

There are no study results found yet for conclusion.

DISCUSSION AND CONCLUSIONS

Summary of the Findings

Global Impact of Computing Solution on Individuals, Organizations & Society

Recommendations

Conclusions

REFERENCES

Follow APA 7th edition formatting when listing your references.

Your reference list and your in-text citations should align!

Van Brummelen , J., & Chen, B. (2020, July 17). *PROCEDURAL GENERATION: CREATING 3D WORLDS WITH DEEP LEARNING*. Procedural generation.

https://www.mit.edu/~jessicav/6.S198/Blog_Post/ProceduralGeneration.html

Brewer, N. (2017). Computerized dungeons and randomly generated worlds: From rogue to minecraft [scanning our past]. *Proceedings of the IEEE*, 105(5), 970-977.

Compton, K. (2016, February 22). *Kate Compton on tumblr: So you want to build a generator...* Tumblr. <https://www.tumblr.com/galaxykate0/139774965871/so-you-want-to-build-a-generator>

turtle_dragonfly (2023, October 13). What are the downsides of quasi-infinite procedural generation in a game like minecraft or no mans sky. Reddit. https://www.reddit.com/r/gamedesign/comments/17761c4/what_are_the_downsides_of_quasiinfinite/.

Rieder, B. (2018). Using procedural content generation via machine learning as a game mechanic. Austrian Marshall Plan Foundation.

Shaker, N., Togelius, J., & Nelson, M. J. (2016). Procedural content generation in games. <https://www.pcgbook.com/>.

Cardy, S (2024, February 2). Suicide Squad: Kill the Justice League Review. IGN. <https://www.ign.com/articles/suicide-squad-kill-the-justice-league-review>.

Trinh, S. (2022, January 12). *The 4 elements of Game Mechanics*. Medium. <https://stephentrinh.medium.com/the-4-elements-of-game-mechanics-fce3166123c2>

Watabou. (2014). *Watabou/pixel-dungeon: Traditional roguelike game with pixel-art graphics and simple interface*. GitHub. <https://github.com/watabou/pixel-dungeon?tab=readme-ov-file>

Watabou. (n.d.). *Watabou/PD-classes: Simple game library for Android*. GitHub. <https://github.com/watabou/PD-classes>

00-Evan. (2014). *00-evan/shattered-pixel-dungeon: Shattered Pixel Dungeon is an open-source traditional roguelike dungeon crawler with randomized levels and enemies, and hundreds of items to collect and use. it's based on the source code of Pixel Dungeon, by Watabou*. GitHub. <https://github.com/00-Evan/shattered-pixel-dungeon>

Davidslv. (2016). *DAVIDSLV/Rogue: Original rogue game (5.4.4)*. GitHub.

<https://github.com/Davidslv/rogue/tree/master>

Appendix A: Title of Appendix

If you are using a survey or particular documents, you can list them as an appendix.

YOUR CODE SHOULD BE INCLUDED AS AN APPENDIX