DISDAIN: An Auto Content Generation VR Game

I. Gusti Ngurah Taksu Wijaya, Seng Hansun and Marcel Bonar Kristanda

Universitas Multimedia Nusantara, Gading Serpong, Tangerang, Indonesia; ign.taksu@student.umn.ac.id, hansun@umn.ac.id, marcel.bonar@umn.ac.id

Abstract

Objectives: To build a video game that implementing procedural content generation with a first-person shooter gener and using the newest gaming hardware technology, i.e. Virtual Reality. **Methods**: Some steps being incorporated in this study, starting from designing the video game, implementing the Procedural Content Generation and Grammars methods, implementing the game level generation, and evaluating the results using Game User Experience Satisfaction Scale (GUESS) and Procedural Content Generation (PCG) evaluation techniques. **Findings**: DISDAIN was successfully built using Grammars to generate the story and Room Generation Algorithm and Corridor Generation Algorithm to generate the maze-like game space. The game was evaluated by using the GUESS questionnaire which results on 77% satisfaction scale. **Application/Improvements**: The technique can be adapted to other heavy content games to reduce the production cost.

Keywords: Corridor Generation, Game User Experience Satisfaction Scale (GUESS), Grammars, Room Generation, Virtual Reality

1. Introduction

Video game is one form of media entertainment that require user to interact with an electronic device, a digital media played by using the computer, console, or mobile phone¹. Video games are everywhere. Everyday millions of people around the world play all sort of video game like Farmville, World of Warcraft, Call of Duty, The Sims, and Starcraft². That is why content within video game is important to keep people engaged², however the demand for better content keep increasing to the point where it is impossible to measure the budget needed to create such content³. But, one of the solutions to this problem is to implement Procedural Content Generation in a game⁴.

Procedural Content Generation (PCG) is a method where computer generate content by using a certain algorithm⁵. This method first appeared in the year 1980 by Michael Toy and Glenn Wichman's⁶. However, despite all efforts reduce the cost of development to create content there are not a lot of researchers conducted regarding this topic⁷.

This results in PCG which still has a few problems that yet not been solved. One of those problems is despite

being able to generate lots of content many players prefer hand-crafted content than computer generated content⁸. This is because it is hard to generate a content that would satisfy the artist and in return also satisfy the players². So, developers usually implement PCG on one type of content, which results in an uninspired content⁵.

One of the goals of PCG is to create a Multi-Level Multi-Content PCG which means it is able to generate all sort of content with high quality and high consistency⁵. One way to improve further development regarding this aspect is by generating two types of content within a game; one of those pair is a game narrative and its game space.

Based on the study above a study of PCG is created. However PCG is just a method and it requires a game to be implemented on. That game will be a first person shooter based Virtual Reality technology. The reason behind this decision is because it's the recent hype and technology breakthrough of the gaming world regarding Virtual Reality. So it is a good chance to explore the newest technology and first person shooter happens to be the top most popular genre of game for 10 years and it keep on evolving⁹. The game satisfaction scale is then measured using GUESS.

2. Procedural Content Generation

PCG is a method where computer generates content by using algorithm⁷. PCG is tied to computational aesthetics and computational creativity and is really important for human-computer interaction¹⁰. Games, Web 2.0, interface, and other design software are popular application that implemented PCG.

In terms of game content this includes: levels, maps, game rules, textures, stories, items, quest, music, weapon, vehicles, characters, etc. However, game content does not include game engine or behaviour of a Non-Playable Character⁷. To be considered a generator a computer must possess these set of skills¹¹.

- 1. Encapsulate knowledge,
- 2. Make structure,
- 3. Encode conditional option,
- 4. Make variance in structure, and
- 5. Able to limit itself.

2.1 Grammars

Grammars are a method with a concept that anything complex is formed from many small things. An example of this would be a galaxy has planets inside of it, a planet has countries, and a country has cities¹¹. Grammars are form of symbols and rules that decide how the symbols will be constructed¹². For example, there are three symbols A, B, and C that follow the following rules:

- 1. A->C,
- 2. A->B, and
- 3. B->CC.

These rules can then be used to any form of symbols that follow the rules, the symbol itself represent the potential event of a quest game¹². One of the advantages of using grammars is that it is easy to understand and implement. Grammars can also be used to generate other kind of content such as game space or game object¹³.

2.2 Level Generation

While grammars will be used to generate the narrative content, to generate the game space or the game level a method by Jessica R Baron will be used, that is by separating the level creation into two-part Room generation and Corridor generation¹⁴. This method is used to create indoor maps which are a structure and rooms that are set into one interconnected space using corridor, that is call a dungeon².

- Room Generating Algorithm is a method/algorithm to create room environment in a game procedurally¹⁴. There are two algorithms that can be used to generate rooms; first is Random Room Placement which places the room randomly within a set game space and second is Binary Space Partitioning that divide the game space into section and areas and then create one room in each area, and
- 2. Corridor Generating Algorithm is the next step; this is the method where the corridor to connect the rooms from the Room Generating Algorithm is created. There are three algorithms that can be used. However, if the room was generated by using Random Room Placement method it can only implement one of the two algorithms, while there are three possible methods to be used when the room was generated by Binary Space Partitioning. The method that can be used in any of the Room Generating Algorithm are: Random Point Connect where the room are connected in a brute force method and Drunkards Walk when the corridor is created randomly until the two room connect or after the set number of vertices reach its limit, the last method can only be used if the Random Generating Algorithm uses is Binary Space Partitioning and that is Binary Space Partitioning Corridor.

3. GUESS and PCG Evaluation

The Game User Experience Satisfaction Scale or GUESS is a questionnaire design to measure the satisfaction scale of a game through nine subscales and 55 items in total¹⁵. GUESS had been tested and validate with 450 unique video games from different genres, meaning this questionnaire is applicable to any sort of video game to measure its satisfaction scale.

GUESS is based on seven point Likert scale with a respond anchored from 1 = Strongly Disagree to 7 = Strongly Agree. GUESS has 55 statements or items and nine subscales or dimensions, which are: Usability/Playability, Narratives, Play Engrossment, Enjoyment, Creative Freedom, Audio Aesthetic, Personal Gratification, Social Connectivity, and Visual Aesthetics.

PCG needs to be evaluated because it allows the better understanding of the generator capability and the content

it generates¹⁶. There are two methods to evaluate PCG which are:

- Bottom-Up Evaluation: This method directly asks the player regarding the content the PCG generates. In¹⁶ recommends the using of Game Experience Questionnaire in regard of this evaluation, and
- 2. Top-Down Evaluation: This method is done by turning content generated into expressive range, this allow the measurement of certain aspect of the content and the behaviour of the generator can be seen from the result.

4. Game Design Document

Since the goal of every designer is to create a successful product¹⁷, we started this section with the Game Design Document implemented in this study. Then, the implementation results, i.e. the Disdain game will be explained briefly and followed by the analysis and evaluation using PCG evaluation techniques and GUESS.

The game is called Disdain, which mean a feeling of disgust and the Formal Elements that are present in the game can be described as follows:

- 1. Player: Single Player
- 2. Objectives: Kill all enemies/Find a way out the maze/ Gather all the hidden object in the level
- 3. Procedures
 - a. Select Campaign to start the game.
 - b. Complete the tutorial and proceed to the next level.
 - c. Depending on the generator and rule generated, finish the objective given which can kill all the enemies, find a way out, or gather the hidden objects.
 - d. If the player health reach zero, the player will go to a game over scene.
 - e. If the player finishes the objective, the player will proceed to the main menu with the leader board updated.
- 4. Rules
 - a. Player needs to have Virtual Reality Headset in order to play.
 - b. Player has to use motion control to move.
 - c. In each level the game space, objective, and narrative are generated and player have to finish the given objective in order to progress.

- d. Player has two modes, the movement mode where the player can move the character and the combat mode where the player can't move but able to shoot the enemy.
- e. Player can defeat the enemy by shooting the enemy until their health reach zero and the enemy can kill the player by attack until the player health reach zero.
- f. Player loses if the health number reach zero.
- 5. Resource
 - a. Player Health: Shows the player of the character remaining health number, start from 100.
 - b. Ammunition: Shows remaining ammunition in the magazine.
 - c. Mini-map: Shows the location of the player in the maze.
- 6. Conflict
 - a. Finish the objective and stay alive.
- 7. Boundaries
 - a. When player is in movement mode, the character cannot perform any combat action, while the player in combat mode the character cannot perform any movement action.
 - b. Game Space of the level is shape as dungeon.
- 8. Outcome
 - a. Total Kills Score: Shows the player the amount of kills they got from one game.
 - b. Time Played: Show the player the duration for them to finish the level.

The Dramatic Elements of the game is as follows:

- 1. Challenge
 - a. Get as many kills as possible.
 - b. Complete the objective as fast as possible.
- 2. *Play*: Rule-based play meaning game has rules, how to play, limitation, and instruction.
- 3. *Characters*: Player plays as Mike, a retired soldier who is disabled to an accident that happened in his time of duty which resulted in the loss of both of his leg.
- 4. *Premise*: Mike is a retired soldier, his traumatic experienced in his time of duty haunt him and each night he is plague by nightmare, in his dream he is haunted by monster trying to kill me him, in each dream he have to find a way out, in each dream he have to wake up, in each dream it feel as though his very life is on the line.
- 5. *Story*: The Nightmare that Mike experience is different each time he sleeps. In each dream he has to

find a way to wake himself up, in each dream there's a strange figure that help him this figure will tell Mike what to do in order for him to wake up. This is where the grammars method comes into play, for it generates the exposition and objective of what the figure will tell Mike.

6. *World Building*: The game world is a dream world, it is dark and the environment is a maze fill with horrors. This symbolizes Mike eternal torment as he stuck forever in this nightmare each time trying to find his way out.

5. Implementation Results

Based on the Implementation that was done, the game had been created and the following is a screenshot taken directly from the game.

Figure 1 shows the main menu of the game; however the screenshot was taken through a monitor and not the Virtual Reality Headset. Player can start the game by choosing the campaign button under the title. To give a perspective of the main menu environment, Figure 2 contains the image of the environment of the player in the main menu.



Figure 1. Main menu display.





Figure 2 shows the environment of the main menu screen. Here player can look around and see the credits, leader board, and last game status. There is also the campaign button to start the game and the exit button to exit the player from the game. When the player start the game the player will be transported to a tutorial scene. Figure 3 shows the first scenery of the game.



Figure 3. Tutorial scene 1.

After a few seconds a figure will appear that will teach the player of the game controls, such is shown in Figure 4.



Figure 4. Tutorial scene 2.

After finishing the tutorial the player will then transported into a different scene, or nightmare scene where the player will receive objective to completed and enemy to face. The map or level in this time is generated and Figure 5 show one of the map generated by the generator in the image shown the objective is to gather the hidden objects and the map is generated using the Random Room Placement for creating the rooms and Drunkard's Walk for generating the corridor that connects the rooms.



Figure 5. Generated dungeon.

Other than the game space the narrative is also generated by a generator. Figure 6 shows an example one of the possible narrative being generated by the game in Bahasa Indonesia.



Figure 6. Generated story.

After that the player has to complete the objective of the level. In the context of Figure 6 the objective of the level is to search for hidden object scatter around the maze. If the player has succeeded, the player will be thrown back into the main menu scene as shown in Figure 1 and Figure 2. If the player losses or was killed by the enemy than the player will be thrown to the game over scene, as shown in Figure 7.



Figure 7. Game over screen.

6. Analysis and Evaluation

When the game is finished, the next step is conduct testing of the application. The total of minimum sample needed in doing a research is 30^{18,19}. There are two evaluations being conducted, first is the PCG evaluation and second is evaluation of the player satisfaction scale with GUESS. The samples are obtained from people around the age of 15-40 which are label as teenager and young adult²⁰.

6.1 Bottom-Up Evaluation

There's two PCG implemented in the game, first is to generate a story or narrative, second is to generate the game space or level. To evaluate the narrative generated the Bottom-Up Evaluation was used. The Game Experience Questionnaire was used to measure the quality of the generator, using only the immersion aspect since narrative or story effect player immersion and no other aspect in the Game Experience Questionnaire²¹. The result can be seen in Table 1.

Sl. No.	Not at all	Slightly	Moderately	Fairly	Extremely
1.	0	9	9	12	0
2.	1	4	9	12	4
3.	2	0	7	16	5
4.	1	4	2	12	11
5.	1	7	0	11	11

 Table 1.
 Game experience questionnaire's results

Based on Table 1, the quality result of the narrative content generated by the generator can be calculated by the following formula:

From this result the narrative score is 0.721333, which can be concluded that player fairly enjoy the content generated by the generator. Figure 8 is the result of 356 contents generated turn into heat map and converted into a 2d histogram.

6.2 Top-Down Evaluation

For the game space content generator two metric was used to measure the behaviour of the generator that is Density: Measure the space used to create the level and Leniency: Measure the difficulties for the player to navigate the level, the higher it is the easier it is to navigate.

As seen from Figure 8 the generator has a high tendency to generate a map with high Density and a very low



Figure 8. Heatmap of game space generator

Leniency meaning the game space is wide and very hard to navigate. Other than that there's also sign of biased of the generator to generate content with High Leniency and High Density meaning the game space generated is wide and easy to navigate.

Lastly is the result on the Game User Experience Satisfaction Scale as can be seen in Table 2.

Table 2.GUESS's results			
Subscales	Average Score		
Usability/Playability	0.77		
Narratives	0.71		
Play Engrossment	0.73		
Enjoyment	0.80		
Creative Freedom	0.76		
Audio Aesthetic	0.85		
Personal Gratification	0.76		
Visual Aesthetic	0.77		
All Scale	0.77		

Table 2.(UESS's results
-----------	----------------

This result shows the scores of each subscales and the final result, the game mostly received a Good respond from the samples with the exception of narrative which score fairly good. The game satisfaction scale scores a 0.77 which mean the game is good.

Conclusion 7.

Based on the research that had been done to design and develop a first person shooter game with procedural content generation based on virtual reality technology, the following are conclusions that can be drawn from the result.

- 1. The game was successfully designed and developed by using Grammars to generate the story and Room Generating Algorithm and Corridor Generating Algorithm to generate the game space, all that while using the Oculus Rift Virtual Reality. Implementing the grammars by setting up the symbol and rules and simply calling the symbols and rules when a story requires to be generated, the objective result will then be given to the level generator to generate the game space that is fitting for the objective, and
- 2. Based on the PCG evaluation it can be concluded that the narrative generator that used grammar method to generate the story reach a fairly good quality with a score of 0.721333. While the game space generator has a high density and low leniency biased when generating its content. The game overall, based on the result of GUESS, reach a satisfaction scale of good (0.77) with the majority of its subscales scoring good with the exception of narrative which scored fairly good.

References 8.

- 1. What is a Video Game? A Short Explainer. Date accessed: 09.03.2016. https://www.thewrap.com/what-is-a-videogame-a-short-explainer/.
- 2. Hendrikx M, Meijer S, Velden JVD, Iosup A. Procedural content generation for games: A survey, ACM Transactions

on Multimedia Computing, Communications and Applications. 2011; 1–24.

- Iosup A. POGGI: Generating puzzle instances for online games on grid infrastructures, Concurrency and Computation: Practice and Experience. 2011; 23(2):158–71. https://doi.org/10.1002/cpe.1638.
- Smith G. The Future of Procedural Content Generation in Games. Experimental Artificial Intelligence in Games: Papers from the AIIDE Workshop; 2014. p. 53–57.
- Togelius J, Champandard AJ, Lanzi PL, Mateas M, Paiva A, Preuss M, Stanley KO. Procedural content generation: Goals, challenges and actionable steps, Artificial and Computational Intelligence in Games. 2013; 6:61–75.
- 6. Forsyth W. Globalized random procedural content for dungeon generation, Journal of Computing Sciences in Colleges. 2016; 32(2):192–201.
- Togelius J, Preuss M, Yannakakis GN. Toward Multiobjective Procedural Map Generation. Proceedings of the Workshop on Procedural Content Generation in Games. Monterey, California; 2010:1–8. https://doi. org/10.1145/1814256.1814259.
- 8. Tell Us Dammit: Most Annoying Video Game Tropes. Date accessed: 27.02.2014. https://www.kotaku.com.au/2014/02/ tell-us-dammit-most-annoying-video-game-tropes/.
- Kruse J, Sosa R, Connor AM. Procedural Urban Environments for FPS Games. Proceedings of Interactive Entertainment. Canberra, Australia; 2016. https://doi. org/10.1145/2843043.2843479.
- Yannakakis GN, Togelius J. Experience-driven procedural content generation, IEEE Transactions on Affective Computing. 2011; 2(3):147–61. https://doi.org/10.1109/T-AFFC.2011.6.
- Kate Compton. So you want to build a generator. Date accessed: 22.02.2016. http://galaxykate0.tumblr.com/ post/139774965871/so-you-want-to-build-a-generator.
- Short T, Adams T. Procedural Generation in Game Design. 1st Edition. New York: AK Peters/CRC Press; 2017. p.

1-336. https://doi.org/10.1201/9781315156378. PMCid: PMC5223622.

- Dormans J, Bakkes, S. Generating missions and spaces for adaptable play experiences, IEEE Transactions on Computational Intelligence and AI in Games. 2011; 3(3):216–28. https://doi.org/10.1109/ TCIAIG.2011.2149523.
- Baron JR. Procedural Dungeon Generation Analysis and Adaptation. Proceedings of the South East Conference. Kennesaw, USA; 2017. p. 168–71. https://doi. org/10.1145/3077286.3077566. PMid: 28689816, PMCid: PMC5573184.
- Phan MH, Keebler JR, Chaparro BS. The development and validation of the Game User Experience Satisfaction Scale (GUESS), Human Factors and Ergonomics Society. 2016; 58(8):1217–47. https://doi.org/10.1177/0018720816669646. PMid: 27647156.
- Shaker N, Togelius J, Nelson MJ. Procedural Content Generation in Games. 1st Edition. Springer International Publishing; 2016. https://doi.org/10.1007/978-3-319-42716-4.
- 17. Aziz MSA, Lindgaard G, Hamzah MS, Whitfield TWA. The design and evaluation of DACADE visual tool: Theoretical implications, Bulletin of Electrical Engineering and Informatics. 2018; 7(1):90–95.
- Sugiyono. Metode Penelitian Kuantitatif, Kualitatif dan R&D. Bandung:Alfabeta; 2012.
- Gunawan AB, Hansun S, Kristanda MB. Nolong.in: An Android based incident notification application with push notification technology, International Journal of Electrical and Computer Engineering. 2019; 9(1):486–96.
- Geiger AM, Castellino SM. Delineating the age ranges used to define adolescent and young adults, Journal of Clinical Oncology. 2011; 29(16):e492–3. https://doi.org/10.1200/JCO.2011.35.5602. PMid: 21482981.
- IJsselsteijn WA, de Kort YAW, Poels K. The Game Experience Questionnaire. Eindhoven: Technische Universiteit Eindhoven; 2013. p. 1–10.