CHAPTER II
THEORETICAL BASIS

2.1 Literature Review

In a research done by Blomberg et al. (2018) titled **"An Exploration of Procedural Content Generation for Top-Down Level Design"** the researchers would employ several types of procedural generation algorithm to its project by setting an input to the project and would set different sets of parameters and random numbers to generate a game object, which runs through a set of rules and functions dictated by the researchers which would define the how the object would be created and how the desired object would look like. Also, through these set of rules and functions in the procedural generation system, it would add a flexibility to the system and make it easier to tweak and add features to the system should any change happens later on during the development of a project.

According to a research done by Theobald Beyer (2017) titled **"Story Guided Procedural Generation of Complex Connected Worlds and Levels for Role Play Games",** the researchers mentioned that the manual creation of quality content in games are expensive and time consuming. Procedural generation in games makes this process less time consuming for human or even replacing the human work effort completely enables development teams to cut off development times in the creation of their projects and greatly benefit the game designers to focus their attention on designing their games instead of spending too much time on laborous tasks that comes with designing a game asset.

According to a research done by Cook, Colton, Gow, & Smith (2019), titled **"General Analytical Techniques For Parameter-Based Procedural Content Generators"** In a creation of a game asset in games, designers has to focus often on the concepts nd language of game design. These core concepts and game design patterns are often used to describe the commonly reoccuring parts of a design of the game that concerns and affects the gameplay inside of the game. These game design patterns provide a basis for which game elements would aid the game designers in creating and integrating the building blocks and inputs of the game into a procedural generation system. The researchers describes the process as a search based

approach to the object creation using design patterns such as objectives, inputs, generators, and parameters, this is done to provide a level of detail for any future systems to be built upon.

Based on a related research regarding the topic of procedural generation and autonomous design by Horton (2019) titled **"Assessing the Validity of Experience-Driven Procedural Generation in Horror Games"** where in which the researchers goes in depth on the process of designing and producing a video game in the horror genre through the process of procedurely generating its game content and laying out the steps and method on how the researchers goes and create the project, and came to the conclusion that in a creation of a procedureral generation system, the waterfall SDLC model which is contains the requirement stage up until the release stage. The waterfall SDLC allows the stages enacted and finished in the project to be revisited if any flaws are found in the system or if some features are lacking and needs further improvements.

A research done by Segarra (2019), titled **"Development of procedural cities generation in Unity3D using Houdini"** explains how procedurally generated 3D objects are built inside of Houdini according to various values such as the number of object and the type of object, and through these values, various geometries, colliders, and UVs will be generated to each objects. And these objects are going to be imported from Houdini directly onto the Unity game engine, where in which lies the advantages of using Houdini as an asset creation tool, which is its wide capacity to export the tools inside of itself on to other softwares. And this also enables Houdini to directly integrate itself within other 3D software and 3D engine, which in this case was the Unity engine, enabling efficient interactions between the two software.

Table 1 shows the literature works the writer drew inspirations from in the process of creating this project.

Table 1 Literature review

Author	Year	Remarks
Blomberg, Jemth, Lennar, Lilius-Lundmark, Johnson, & Svensson	2018	Procedural generation systems offers more flexibility to add tweaks and features in object creation process.
Theobald Beyer	2017	Procedural generation systems enables the object creation process to be less time consuming
Cook, Colton, Gow, & Smith	2019	In creating a procedural generation system, game design patterns needs to be identified in order to create more level of detail in a game object.
Jez Stuart Edward Horton	2019	The waterfall SDLC model is a method used to create a procedural generation system.
Segarra	2019	The Houdini engine would be used to create procedural generation system.

In conclusion, based on the literature works listed above, procedural generation systems offers more flexibility to the object creation process (Blomberg et al., 2018), makes the object creation process less time consuming (Beyer, 2017), and by identifying gmae design patterns, would create more level of detail to the game object (Cook et al., 2019), The waterfall SDLC model would be used as the method in creating the procedural generation system (Horton, 2019) and the procedural system would be created inside the Houdini Engine (Segarra, 2019).

2.2 Theoretical Basis

2.2.1 Waterfall SDLC

The method used in this project would be in the form of a waterfall SDLC. The waterfall SDLC method ensures that the end product is free of design flaws that was made during the development of the project, and due to the structure of the model, each stages does not overlap with one and another, and only begins and ends before beginning the next stage (Alshamrani & Bahattab, 2015). According to Kramer (2018), the waterfall SDLC model is used due to the nature of the project constantly shifting between different ideas and inputs with each iterations, this method allows parts of the system's interaction to be easily predicted and allows each stages of the project to be revisited if any changes happens during development.

In a research done by Istiqomah & Sudarmilah (2019), the authors used the waterfall SDLC as their development structure and overall design process in creating an android based game. In the early stage of the process, they started doing an analysis on the requirements of the system beforehand, which concluded in the usage of the software Construct 2, Lexis Audio Editor, and Coreldraw X7 as a tool to create the software and referencing several literature works, books, and interview results in creating the contents of the app.

2.2.2 Multimedia

According to Setiawati, Surya, Hasyim, & Dimyati (2017), Multimedia is a combination of multiple method of communication and transmission of messages, stemming literally from the latin word "multi" which means more than one and the word "media" which translates to a method of communcation. And according to Purba (2019), multimedia is a combination of multiple mediums of communication that changes the way people transmit and deliver information and can be split into several components, which are as the following:

1. Text

   Text is a combination of letters that formed a word or a sentence that explains or communicate information to other people

2. Image

   Image is a component of multimedia that delivers and communicate information in a form of a visual graphics, some examples are drawings or a computer rendered image.

3. Audio

   Audio is defined as a type of sound often stored in a digital form that delivers information in an audible form such as music or narrration.

4. Animation

   An Animation is a form of media that combines several amounts of images into one and displayed on consecutive order, creating the illusion of movement.

5. Video

   Video is defined as a combination of video and audio, by using animation's method of creating the illusion of movement by stiching several images into a consecutive order and the audible information stored in a form of audio forms video's audio visual form.

According to Setiawati et al. (2017), multimedia can be devided into 2 categories, which are as follows:

1. Linear Multimedia

   Linear Multimedia can be described as a form of multimedia that does not have a tool for control or method of controlling anything that can be operated by a user.

2. Interactive Multimedia

   Interactive Multimedia, are a form of media that is equipped with a tool or method of control and interaction that can be operated by the user, which allows the user to dictate how the media would operate.

According to Aminah (2018), multimedia has a varierty of usage and can accomodate various field of work, which are as follows :

1. The field of education

   Multimedia has the ability to better emphasize and visualize a complex subject matter that are hard to convey through words or other conventional method

2. The field of health

   Multimedia is able to inform complex health related information in a concise and more comprehensible way, and is able to promote better living lifestyle and health research.

3. The field of marketing

   Multimedia can be used as an effective way of presenting or promoting a product in short and concise manner and often is delivered in a more distinctive and attractive way.

4. The field of filmmaking

   Multimedia is a prominent component inside of filmmaking in creating an medium of entertainment such as the use of animation and computer generated imagery inside of movies.

5. The field of entertainment

   Multimedia can be used in creating a medium of entertainment and offering interactivity in the process, such as video games.

2.2.3 Procedural Generation

Game development is a laborous process that requires a significant amount of time to be spent on creating content, this includes terrains and game object that requires a significant amount of time by the game designers, with the quality of the content affecting the development's budget and time. And in this case, using the technique of procedural generation can help make the process much less tedious and lower the content creation cost in development (Soares De Lima, Feijo, & Furtado, 2019).

In a research done by Erath et al. (2017), procedural generation as a technique of 3d model creation has been used in many cases including the usage in the field of transport planning and intergrating it inside a micro simulation with a game engine rendering it in a semi automated pipeline, where in which the author uses this technique to create a traffic micro simulation inside of a virtual reality application to simulate driving behaviours.

According to a research about Procuderal Content Generation by (Amato, 2017), procedural content generation in games refers to the process of automatic or semi-automatic game content generation through algorithmic means. And the types of content that can be produced through the method of procedural generation can be split into the following:

1. Game Bits
   These are the basis and fundamental elementary units inside of a game, which does not affect the player's gameplay if it is considered independently. This category includes aspects of the game such as : textures, sound, structures, vegetation, fire, water, clouds or stones.
2. Game Spaces
   This part of the game represents a certain area or environment where in which the game takes place in, this typically cosists of different units of Game Bits. This category includes aspects of the game such as : indoor maps, outdoor maps, and water bodies such lakes, seas, and rivers.
3. Game System
   The game system cosists of parts of the gmae where in which the game's overall area of play with for the player, this type of objects cosists of things like the ecosystems, road networks and urban landscapes.

4. Game Scenarios

   This part represents the system running the flow of a series of developments, determining how, where or when events would occur, how different game objects will interact with one and another such as puzzles, storyboard, and the concept of the levels.

5. Game Design

   This is the part that determines the rulesets of the game, what the player can or cannot do, how the game system will flow from one point to the next, and setting up objectives for the player to where each has to be done in order to progress through.

Also in a research by (Amato, 2017), other types of taxonomy on what kind of content generated through procedural generation is being proposed, which are as follows:

1. Online-Offline Generation

   This first type of procedurally generated content examines whether the content is generated online (while the game is running) or offline (during the development of the game). One example of an online Procedural Content Generation is when a player enters a stage or opens a door of an object or structure, the game would instantly generate its interior, rooms, walls and decorations. On the other hand, an example of an offline Procedural Content Generation is when an algorithm constructs the basic internal layout of the structure, which then are modified and refined by the designer before the game is completed. With the 2 methods in mind, online Procedural Content Generation must comply with the basic requirements of creating a valuable content, it must be fast, and the content it creates must be qualitatively acceptable.

2. Necessary-Optional Content

   This type of procedurely generated content refers to the importance of the content, whether or not the generated content is necessary or optional for the game. Necessary content can be described as a

content required by the player to progress through the game. For example, monsters that must be defeated or dungeons that must be completed. On the other hand, optional content is everything that the player could choose to not do or consider, such as weapons and items that could be ignored. Another difference between the 2 types are the required quality of the content generated, necessary content must always have an excellent quality and fucntionally correct, because things such as an unpassable dungeon, an undefeatable monster, or an incorrect set of rules can cause the player to be unable to progress through the game.

3. Control Degrees

This type has a distinction that depends on the type of generation algorithm used and how it can be parameterized. In one case, the algorithm can be used to randomly generate a number as an input and on the other the algorithm can be used as a multi dimensional vector that contains the parameters taht sepcifiy the properties of the generated content. For example, in generating a dungeon, the algorithm can generate an input parameter for the number of rooms, branching corridors and item locations the dungeon has, with more degrees of control it has, the more the generated content can be customized, tweaked and controlled.

4. Generation Deterministic or Stochastic

This type distinction relates to the degree of randomness during the build process. It is possible to hve deterministic algorithms that will generate the same content with the same input parameter or if it would generate different content even with the identical input parameters.

5. Constructive or Generative-with-test

This type has a distinction that refers the output of the algorithm. Constructive algorithms can generte content and end their execution, then producing the output result. For example is the use of fractals to generate terrains, a generative with test algorithm will incorporte a genertion mechanism and a test, which means that during the execution of the algorithm, each content instance is being tested according to criterias depending the design of the game. If the test fails, then it is fully or partially discarded and regenerated again, and the process continues until the content passes the test and has suficient quality.

2.3 Tools used

2.3.1 Houdini

Houdini is a 3D graphics software package developed by Side Effects Software released in 1996 it was one of the main appliactions used in the video game and film industries in generating procedural content, and has since been in development since then until the version 17.5. This software has been primarily used in films and TV (Saario, 2019). Houdini is not only limited to procedurl generation, it also allows a simulation of fire, fluids, particles, volumes, lightning, cloths, fur, crowds, and complex physics, rigging of a character and perform animations (Luis & García, 2018).

According to Wassynger (2018), Houdini is a node based 3-Dimensional environment that allows users to easily and procedurealy build computer generated assets. Houdini's usage for procedural generation can be seen in a research by Chang & Luoh (2019) in generating dimensional, granular elements such as dust, debris, and cobwebs to enhance storytelling and ambiance in the movie Toy Story 4.

And according to Kekre (2017), Houdini's tools are mostly implemented as operators, which is benefitial on creating highly detailed objects to be constructed with a very few steps and encouraging non-linear workflow and allows combining existing operators. These operators are divided into several main groups, which are as follows:

1. SOPs (Surface Operators)

   Surface Operators can be used to deal with geometric structure by influencing surfaces, vertices, points, normals. Surface Operators also provide capabilities to dealt with static volumes and object velocities. These operators play a big role in procedural modelling.

2. OBJs

   This is a type of node tht passes transform information, certain material and render settings, which are traditionally composesd of one or more SOPs. A collection of these nodes are known as an Object network, which are otherwise known as the "Scene level"

3. DOPs (Dynamic Operators)

   Dynamic Opeators are used for dynamic simulation of fluids, cloths, and rigid body interaction. A collection of Dynamic Operators forms Dynamic Operator simulation, which in turn provides controls such as caching options, and timeskip controls.

4. SHOPs (Shading Operators)

   Shading Operators can be used to create shading networks to represent different shaders and are responsible for the creation of materials.

5. POPs (Particle Operators)

   Particle Operators are used to manipulate particle systems by providing functionality to introduce, or control, forces, noise, and collisions.

6. CHOPs (Channel Operators)

   Chnnel Operators are used for manipulating time-based channel data such as animation curves or audio informations.

7. COPs (Composite Operators)

   Composite Operators are used to manipulate 2D pixel data. Like other mainstream compositing packages, there is funcitonality to composite images such as render passes and depth maps.

8. ROPs (Render Operators)

   Render Operators are used for building networks to represent different render passes and render dependencies. Different nodes can also represent different render jobs or launch different renderers.

9. VOPs (VEX Operators)

   VEX Operators are used for building nodes of any of the above types using a graphical representation of VEX.

2.3.2 Unity Engine

Unity Engine is a cross-platform game engine released in 2005 and was developed by Unity Technologies. Whereas most games are built to accommodate a very specific type of software development, the Unity Engine is considered to be a general purpose engine, meaning that although its default toolset is built to accomodate the creation of particular types of content, it is ubuilt to accommodate a range of possible project types, design methodologies and production workflows (Nicoll & Keogh, 2019).

In a research done by Nguyen & Dang (2017), the unity engine is used as a tool in the creation of a virutal reality and augmented reality learning environment through the use of the Vuforia package and Google Cardboard package for unity. And according to (Segarra, 2019), in order to implement a tool created in Houdini inside of the Unity, the Houdini Engine is required as bridge to properly integrate and execute tools created inside Houdini. The Houdini Engine Pipeline Roadmap can be seen in the figure below.

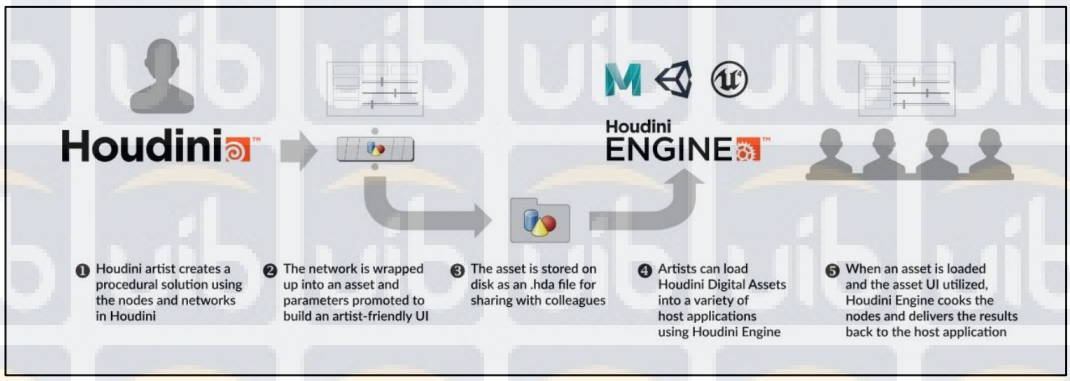

Figure 1 Houdini Pipeline Roadmap

According to Halpern (2019), Unity's default editor layout consists of the following:

1. Scene View
   The Scene View panel functions as a panel to navigate around a game scene inside of Unity
2. Game View
   The Game View displays renders of a game from the active camera's point of view
3. Asset Store
   The Asset Store panel is the online storefront of Unity Technologies and can be used to import objects to a unity project.
4. Hierarchy Window
   The Hierarchy Window displays a list of all objects in the current Scene in a hierarchical format.
5. Project Window
   The Project Window gives an overview of all the content in the Assets folder
6. Console Window
   The Console Window will display errors, warnings, and other output from a Unity application.
7. Inspector Window
   The Inspector Window is used to modify game objects such as scripts, meshes and colliders inside of a scene.