An Evaluation of Procedurally Generated Cities and its Impact on Rapid Prototyping

Kevin Conaghan
School of Design and Informatics
Abertay University
DUNDEE, DD1 1HG, UK
Word Count: 3131

ABSTRACT

Context/Background: Changes during development of a game can be costly if undetected in the pre-production phase. A possible solution is using procedural generation in the pre-production phase rapidly prototype in a level or map, similar to what would be built in the production phase. This would make it easier to spot any potential problems or improvements.
Aim: To create a tool that can procedurally generate cities to compare design pipelines and analyze the effectiveness of testing relevant game mechanics in the city environment.
Method: A 3D prototype will be created using Unreal Engine. One type of city will be the minimum requirement for this project: the city will be modular in both size and number of buildings. The focus of this project will be generating realistic buildings which will be dynamic and replaceable. A new development pipeline will be created to compare against existing development pipelines.
Results: The project will analyze the effectiveness of the city in testing relevant game mechanics and will also determine if the city will be a viable tool for rapid prototyping. The prototype will undergo a series of performance tests while it is generating the city. Two questionnaires will be handed out: one for testers to compare the generated city with other cities in games and one for asking about the effectiveness of the prototype design pipeline.
Conclusion: The project will investigate various procedural methods and compare different design pipelines to create a procedural city and compare the effectiveness of the new design and development process.

Keywords

Procedural Generation, City, Design Pipelines

1. INTRODUCTION

1.1 Project relevance

Today Procedural Generation is used in almost every aspect of games development. The reason it is so popular is due to its ability to create a large quantity of assets or levels in a short space of time than compared to if they were handcrafted. This allows small independent games companies to compete with AAA games without the budget of a AAA company. One of the more popular games that uses procedural generation is *No Mans Sky* (Hello Games, 2016). It uses procedural generation to create a universe filled with billions of stars and planets, and within those planets, beautiful scenery and wildlife for players to explore. To create a system this complex is an impressive technical achievement for a small studio like Hello Games. Most likely the most popular game to use procedural generation is *Minecraft* (Mojang, 2011). Minecraft is a sandbox game that uses procedural generation to create worlds in a box style. The game has since sold over 122 million copies across all platforms. After the release of Minecraft, many games started using procedural generation to create sandbox worlds including *Terraria* (Re-Logic, 2011) and *Starbound* (Chucklefish, 2016). *The Sinking City* (Frogwares, 2019) makes use of procedural generation to create a city in a matter of hours where the buildings and different areas of the city are completely customizable even after generation. *Middle Earth: Shadow of Mordor* (Warner Bros, 2014*)* uses procedural generation for its Nemesis system; a system that creates a unique relationship with the player and the game's orcs, to generate unique enemy orcs with traits relevant to the player and the style they play the game. The enemy can even remember when the player last fought them and if they were injured the enemy will have a scar that is also procedurally modelled into the enemy character model.

1.2 Project focus

Procedural generation has been an invaluable tool in the past creating both beautiful scenery and enjoyable levels that can compete with handcrafted levels. This project will look at different procedural generation methods and how they can be applied to create a dynamic city. The result of this project will be a 3D prototype of a complete city that can be edited after creation for further design purposes. It will also have various game mechanics that can be tested within the city.

Aerial View of Glasgow (Scotland Now, 2015)

This project will be evaluated using several performance tests and a questionnaire for testers. The project will also create a unique development pipeline to be compared against current development pipelines to test effectiveness. The data will be presented to convey which parts of the prototype excelled and which parts still need improvement.

1.3 Aim

The current aim of this project is to create a tool that procedurally generates cities: to compare development pipelines and to analyze the effectiveness of testing relevant game mechanics in the city environment.

1.4 Objectives

The project objectives are:

- To asses current procedural generation methods and choose appropriate method.
- To build a suitable system for testing.
- To compare current design pipelines and compare using the system to determine effectiveness.
- To investigate current development pipeline and identify areas of development.
- To evaluate whether the prototype system enhances the overall development & design process.
- To evaluate whether it is a viable system to test game mechanics.

2. BACKGROUND

2.1 Development pipeline

The traditional pipeline for the game development process can be shown below in Figure 2.1:

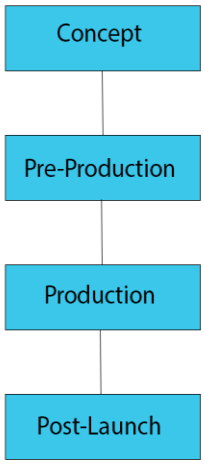

Figure 2.1 A Standard Development Pipeline

In the concept phase an idea is formed. The pre-production phase is where the idea is developed into a solid prototype to show off. In this phase artists work on concept art, programmers work on prototype game mechanics and designers concept the level design and the story. This is where a minimum viable product would be created. The production phase is where everything has been decided and the team works together to create the final product. In this phase, if there are any problems or if a change is required, it will come at a great cost in time and resources. In the post-launch phase, the developers will work on any minor bugs and any down-loadable content (DLC) that has been concepted. If designers were to use procedural generation, they could save time in the pre-production phase with level design and environment placement. The designer can test mechanics effectively in a more detailed level or map using procedural generation than if a designer created a prototype level manually. Shown below is a level in the pre-production phase, figure 2.2, and the level in production phase, figure 2.3:

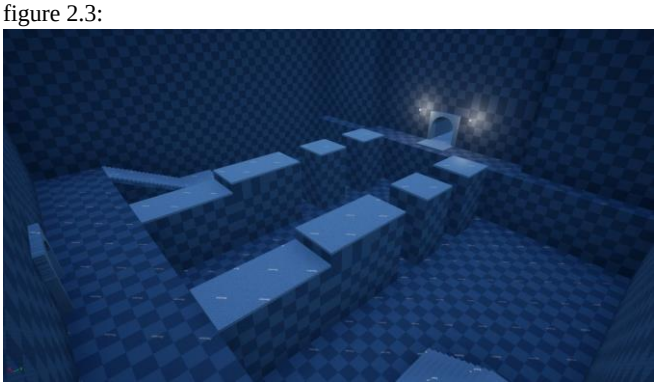

Figure 2.2 Pre-Production level created by Twitter user Vicente Quesada

Figure 2.3 Production level created by Twitter user Vicente Quesada

Currently, artists expend a majority of the development time, especially in a 3D game. An example of a basic modelling pipeline is shown below in figure 2.4:

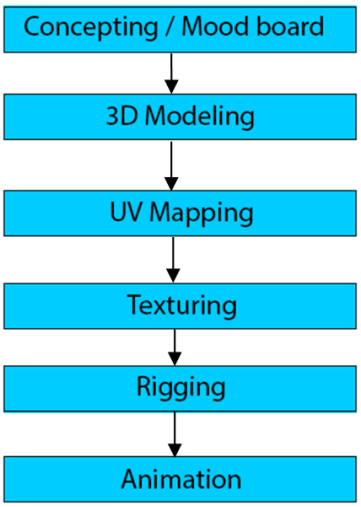

Figure 2.4 A basic modelling pipeline that an artist would use.

This can be a time-consuming process as models can be complex. This process would be repeated until all the models for the game have been completed. With this workflow in mind a team of artists would take months at a time to create and model a city such as New York. Even if the city was fictional, more time would be spent on mood boards and creating the overall style of the city. Also, during this time designers will often have to wait for assets to be created before they can be placed in the level.
This issue will be investigated in this project and the aim is to reduce cost of change in the production phase. However, these pipelines are a standard that most game developers use and generally works well within the industry.

2.2 Procedural generation

In the games industry procedural generation is used in every area of development including art, level and audio. Procedural generation is a powerful tool for developers as it can instantly generate assets and levels of near infinite size, which saves time and resources. Games developers have been using procedural generation for years now but it was first mainly used to generate levels and then it branched out into other departments such as: 3D modelling, character attribute generation and loot systems in MMO games.
Procedural generation can be also used for character modelling and realistic animations. An example of procedural character modelling is its use in *No Man's Sky* (Hello Games, 2016).. It uses procedural modelling to create unique alien wildlife for the players to interact with. Figure 2.5 shows the possible parts the wildlife could be made up from:

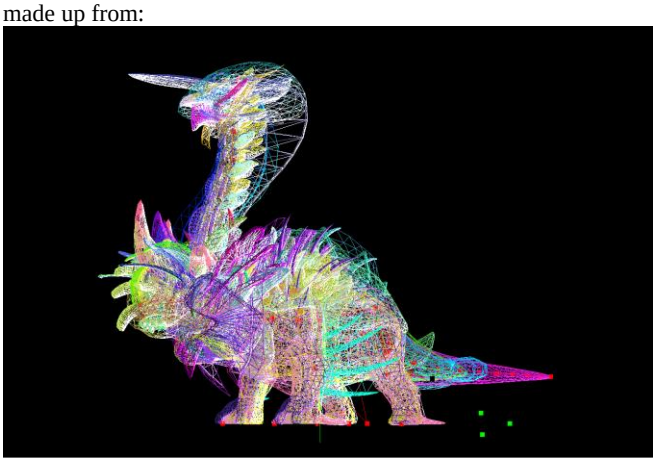

Figure 2.5 No Man's Sky's procedural character modeling. Image taken from Kotaku

Procedural generation also has its disadvantages. Sometimes level generation can seem repetitive like in games such as *Diablo and Darkest Dungeons.* They use procedural generation to generate dungeons and after playing, the level's art assets become familiar and repetitive which then has a negative impact on the gameplay. A possible solution that will be implemented into the prototype is:

- Human attention will be required for the cities as the node network will be placed manually and the buildings will be editable after generation.

2.3 Occlusion

Occlusion may be used to detect if the buildings overlap with each other. If an object is occluded, then it is blocking or intersecting another object. Müller proposes using an occlusion query test in the procedural modelling of buildings to test if a shape is occluded with another shape. The test can return three categories (full, partial or no occlusion) and the results are then used to determine which shape should be placed next. This technique will be investigated as a possible way to check if buildings overlap with each other. If the prototype is ahead of schedule, then it will be used in the procedural modelling of the buildings. (Müller et al., 2006)

2.4 Past work

Frogware's *The Sinking City* uses a node system to procedurally generate a city with pre-designed buildings. The nodes create the road network and different districts such as: residential, industrial and the city district. The districts can be defined when the nodes are connected. This process defines which types of buildings should spawn in which place. After the city has be generated the designers can analyze the city and replace the buildings, making it a valuable tool that will save both time and resources. The game is expected to release in March 2019.

CityGen is an interactive system that procedurally generates cities and was created in 2008 by George Kelly. It uses several techniques to generate the cityscapes and was intended to be used for games and other graphical applications. (CityGen, 2007)

Müller proposes using L-systems to draw the road network and then subdivides them to create the lots for the buildings. The buildings are created and placed using a stochastic grammar. A stochastic grammar is a random probability distributer which is a possible technique that can be used for generating cities. Müller also uses grammars to procedurally model buildings and create complex buildings without the need for an artist. This way of modelling buildings will be investigated as an additional task for the prototype. (Müller et al., 2006)

Perlin noise has been used in various projects to generate cities. Olsen proposes using Perlin noise to generate cities because of its more natural form. The idea behind this was that if everything in the project was created using this procedural technique then it would be fast to implement. (Olsen and Frank, 2017)

The use of Lindenmeyer-Systems (L-Systems) is a popular technique to use in procedural generation and was first created by Aristid Lindenmayer as an algorithmic way of simulating plant development. L-Systems is a system that rewrites in parallel and simultaneously to replace the predecessor. Parallel production can have a beneficial impact on the performance of the formal properties of the rewriting systems (Lindenmayer and Prusinkiewicz, 2004). This technique will be investigated to generate the city using pre-defined rules. Another technique that is very popular is Grammars (Shaker, Togelius and Nelson, 2016), it uses a left-hand side (LHS) and right-hand side (RHS) system to distribute its set of rules and replace the predecessor, for example if:

1. A ⟶ AB
2. B ⟶ *b*

Then the *"initial string is A, in the first rewriting step the A would be replaced by AB by rule 1, and the resulting string will be AB. In the second rewriting step, the A would again be transformed to AB and the B would be transformed to b using rule 2, resulting in the string ABb. The third step yields the string 'ABbb and so on."* (Shaker, Togelius and Nelson, 2016).

This method will be investigated and compared against grammars for implementing the placement of buildings.

3. METHOD

3.1 Execution

The finished prototype for this project will consist of a 3D city that is procedurally generated and is capable of editing after generation. The project will also contain a design pipeline that will be created for the prototype. Different procedural techniques will be explored such as *The Sunken City's* node system to achieve the desired result. The city will be separated into different areas: city centre, urban areas and industrial area: this will make the city more realistic. The node system will be implemented first to create a road network and will make use of splines to achieve this.

After the road network has been placed, the buildings will then be generated at the side of the road using grammars. The prototype will also make use of occlusion to ensure the buildings do not overlap. The buildings will be saved as a prefab upon generation which will be capable of editing in Unreal Engine's editor: this will allow the designer to replace any buildings in the scene generated with either another generated building or one an artist has modelled.

The main goal is to make the city editable after generation. This will allow designers freedom to customize the city to make it more immersive and feel more like a city. The prototype will have some pre-defined buildings, however if the additional tasks are achieved then the buildings will also be procedurally modelled which would make every building unique and create a more realistic city.

The project is ambitious but within the timescale a prototype can be developed which includes: a node system to create the road networks and the generation of a basic city environment.

3.2 Design pipeline execution

After the prototype has been developed a new design pipeline will be created that will utilize the prototype and procedural generation in general. The main goal of the design pipeline is to develop a new and efficient pipeline that will help designers and artists cut down their production time which will increase overall polish of a game. The new design pipeline will have a similar process to pipelines mentioned above only it will make the designers more efficient in their work. An example of a workflow is shown below in figure 3.1:

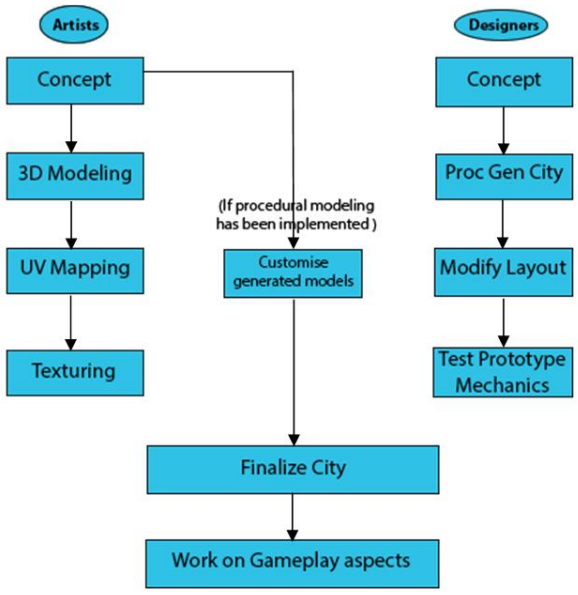

Figure 3.1 An initial plan for the design pipeline

3.3 Potential problem areas

A potential area that could result in a problem is Unreal Engine's error handling. If Unreal cannot find an object in the scene it will crash the whole application. This can become time consuming if error handling is not carried out properly. Buildings will have to be stored in an appropriate data structure to ensure that they can be checked that they are in the scene.

Another potential problem is how to sub-divide the plots of land, so the buildings can be placed. If the plots between the road network is not divided correctly then potentially, some of the buildings will be inaccessible as other buildings will be blocking their way. Appropriate area calculations will be made to ensure the plots are sub-dived correctly and will keep the buildings in line with the road and not behind others.

3.4 Evaluation

The evaluation will consist of 3 sections:

i) The prototype will have a series of performance tests to ensure the impact of procedurally generating cities is viable for use in games development. Some variables that will be tested are: total time taken to create cities of various sizes; storage space required; number of vertices and texture data generated.

ii) A questionnaire will be handed out to testers evaluating how the city compares to cities in games regarding size, variety of buildings and structure.

iii) A second questionnaire will be handed to design students asking a series of questions about how efficient the new pipeline is and whether they would use it in certain genres of games. A standard design pipeline will be provided to allow comparison between the two.

The data collected from these evaluations will be portrayed in graphs that will help quantify the advantages and disadvantages of procedurally generating cities for rapid prototyping.

After the data has been collected it will be evaluated and if the requirements mentioned below are met then the project will be successful:

1. Successfully generate a fully editable city.
2. Testers will give positive feedback on the city in the questionnaire.
3. The new design pipeline will have positive comparison with the standard design pipeline.
4. The prototype will have good performance results and become a potentially viable tool for procedural generation in games development.

3.5 Additional tasks

If there is extra time the procedural modelling of buildings will be investigated further and implemented into the city. This will create more realistic buildings and will accelerate the design process further as the artists will not need to model all the details. City scenery such as: bins, traffic lights, street lamps and benches may be procedurally generated to further create an immersive city. In addition, some game mechanics will be created to test out in the city and see how well the city works with actual game elements. The mechanics will be relevant to the city environment to test the quality of the city in both design and impressiveness. Some mechanics may include: racing, climbing, checkpoints and pathfinding.

4. SUMMARY

Procedural generation is being used widely in games development as a tool to create almost infinite worlds. Smaller games development companies are creating more popular games due to tools such as procedural generation as it can save them time and resources.

The aim of this project is to assess the advantages and disadvantages of the use procedural generation as a method of accelerating the design process. The prototype will be created in the Unreal Engine and a design pipeline will be created. Both the prototype and design pipeline will be used to gather data: the prototype will collect both performance data and qualitative data through testers completing a questionnaire; the design pipeline will be compared against another and collect qualitative data through another questionnaire.

The results of the project will indicate if procedural generation can be used in a games environment to create a more efficient workflow and hasten the overall development of games development. This research will help smaller games development companies to compete with AAA companies as it will show that procedural generation can be utilized at a fraction of the cost than if someone were to handcraft a world.

5. REFERENCES

1. Lindenmayer, A. and Prusinkiewicz, P. (2004). *The Algorithmic Beauty of Plants*. Springer-Verlag.

2. Noor Shaker, Julian Togelius, and Mark J. Nelson (2016). *Procedural Content Generation in Games: A Textbook and an Overview of Current Research*. Springer. ISBN 978-3-319-42714-0.

3. Müller, P. et al. (2006) *Procedural Modeling of Buildings. ACM SIGGRAPH.*

4. CityGen (2007). Available at: http://www.citygen.net/ (Accessed: 3rd October 2018)

5. Olsen, N. and Frank, E. (2017). *Procedural city generation using Perlin noise.* Blekinge Institute of Technology.

6. Frogwares (2019). *The Sinking City* – PC [Video Game]. Frogwares.

7. Hello Games (2016). *No Man's Sky* – Playstation 4 [Video Game]. Hello Games.

8. Mojang (2011). *Minecraft* – PC [Video Game]. Mojang.

9. Warner Bros (2014). *Middle Earth: Shadow of Mordor* – Playstaton 3 [Video Game]. Warner Bros.

10. Scotland Now (2015). Available at: http://www.scotlandnow.dailyrecord.co.uk/lifestyle/glasgow-above-see-scotlands-largest-5851493 (Accessed: 1st October 2018)