City Generation Using Modularity

Yven Lommen
yvenlommen@gmail.com

Niels NTG Poldervaart
n.t.g.poldervaart@umail.leidenuniv.nl

Floris Tomassen
floristomassen@gmail.com

Arvindeva Wibisono
arvindeva@gmail.com

Ricardo Blikman
ricardo@blikman.com

ABSTRACT
Procedural Content Generation (PCG) is a wonderful field within game AI. It aims to produce content within games algorithmically. Applications of this are seemingly endless, including quest generation for open-world games, level generation and dynamic narrative generation. In this report we describe our approach of building settlements for the AI Settlement Generation Challenge 2021 for the game of Minecraft. We explain how we analyzed the urban architecture of Barcelona to make modules and produce city blocks. We also show some of the results of settlements (or cities) our algorithm produces. A downside of our chosen city architecture is that the terrain needs a flat land, so adaptability to the environment was no option even though it is desired for the competition.
KEYWORDS
GDMC, Procedural Content Generation, Minecraft, Settlements.

1. INTRODUCTION

The game Minecraft was released in 2011 and has been a global phenomenon ever since. The voxel-based building blocks that make up the world of Minecraft can be edited by the player, creating a free sandbox of nearly limitless possibilities. At its core its a survival game where you try to survive by growing crops, eating livestock and building a settlement. However, an equally well-known game mode is *Creative*, where players are free to fly around the world and build whatever they want with unlimited resources. A popular thing to do within the Minecraft community is building cities and recreating real-world buildings, sometimes to scale. Creative mode is even so popular that there exists a *GDMC AI Settlement Generation Challenge*[1], where contenders have to design an algorithm that generates settlements. In this report, we share our design, implementation and preliminary results for this competition. In Section 2 we firstly discuss the rules of the competition. In Section 3 we explain how we approached the implementation of our settlement generation algorithm. In Section 4 we show some examples and pictures of cities that were generated by our algorithm. Finally, in Section 5 we will conclude our findings and recap the most important aspects of our report.

[1]Organised by New York University since 2018 https://gendesignmc.wikidot.com/wiki:settlement-generation-competition

2. AI SETTLEMENT GENERATION CHALLENGE

The AI Settlement Generation Challenge is hosted by Generative Design in Minecraft (GDMC) where the goal is to write an algorithm that generates an interesting settlement for a given (unknown) Minecraft map [3]. The competition judges the algorithm by applying it to three different maps which are unknown to the participants beforehand. So the real challenge lies by making the algorithm adaptive to a unknown environment. In previous years, the algorithm has to apply to a given map with a size of 256x256.

The competition nicely describes the challenge for this report, we can formulate the research question as:

Can we design a algorithm that generates believable settlements which are in line with the AI Settlement Generation Challenge standards?

In the upcoming paragraphs we will share the ideas and design principles we used for our solution to this challenge.

3. METHOD/APPROACH

Although the AI Settlement Generation Challenge has some clear restrictions and encourages participants to make an adaptive AI, we did slightly diverged from these guidelines. We studied the infrastructural architectures of some major cities in the world such as Paris, Barcelona, Amsterdam and Rio de Janeiro. What we found was that major cities had some great potential to be generated on large scale by algorithms because there was great reproducibility within the general architecture. We eventually chose to generate a settlement with the same architectural design of Barcelona, specifically the Eixample neighbourhood. How this architecture looks will be discussed in Section 3.1. Our algorithm was based on modular streets and buildings such that a city could be relatively easily generated with standardized building blocks. We chose for a grid of 256x256 (based on previous years of competition evaluation) and let our algorithm build a city within this grid. One of the advantages of the

modules we built is that we can change the default materials with other random ones such that a wide variety of buildings could be built even though the modules are the same. This makes for a colourful and varying city, the modules are further discussed in Section 3.2. One downside of our algorithm however is that it is not adaptive, the algorithm flattens the area which it wants to builds on.

In this report procedural content generation (PCG) is the central point of attention within the context of the course of Modern Game AI Algorithms. We got the idea of the AI Settlement Generation Challenge from the first lecture on PCG. We also used the second lecture on PCG for ideas to inspire us in our approach. We finally looked at the lecture about experimentation for our approach. When designing the city blocks and pre-built modules (further discussed in Section 3.2) we had to incorporate the rotation of these modules within city blocks. For this part a lot of experimentation and trial and error was used to achieve the desired result.

3.1 Barcelona Architecture

The city of Barcelona is a huge metropolis with the iconic Sagrada Família in the centre of it. One of the reasons for the fame of this basilica (besides its beauty) is the fact that it never finished, this fits our theme of procedural content generation quite well. But that is not the only eye catcher in Barcelona. The *Eixample* section, which makes up a large part of the city, the city is designed in square blocks[1]. Figure 1 shows an aerial view of the city where you can see the building blocks. These building blocks all look the same at first glance but are still different in height and content. Each of these building blocks can include but are not limited to shops, restaurants or apartments. This city design inspired the fundamentals of our algorithm, which generates a city which also shows these kinds of city blocks.

Figure 1: Aerial view of Barcelona.

3.2 Modular buildings

The algorithm that we designed, uses modular buildings and streets to generate a random city in the style of Barcelona as described in the previous paragraph. Within Minecraft we can save pre-built structures such that we can save them (as NBT files [2]). These pre-built structures are from now on referred to as modules which the algorithm can access for city generation. Our algorithm uses the following components to place these modules on:

- City blocks of 64x64 Minecraft blocks.
- Streets (or roads) of 22x64 Minecraft blocks between building blocks.
- Intersections of 22x22 Minecraft blocks between streets.

The city blocks upon which buildings are placed actually have eight buildings on them, one on each corner (corner buildings) and a building between each pair of corner buildings (middle buildings). This resulted in four types of modules:

- Corner building modules for the first floor.
- Corner building modules for upper floors (so non first floors).
- Middle building modules for the first floor.
- Middle building modules for upper floors.

The corner building modules have a size of 21x21 whereas the middle building modules have a size of 22x21 which results in eight buildings on a city block. It also leaves us with an empty centre of a city block with an exact size of 22x22. We have also designed special for these centre which we refer to as courtyards.

These building blocks also have a unique feature before placement. The building blocks are randomly chosen and can differ from the actual module from which they are based on. For the first floor modules, the material from which the building is randomly chosen by the algorithm along with the door. The upper floor modules in addition have randomly selected balconies. The height of each building (number of floors) is randomly chosen within a range of [6,9] using a uniform distribution.

The building blocks are separated by streets and intersections which also have their own modules.

The streets are generated in a similar fashion to the city blocks. A street module consist of sixteen sidewalk structures, six to eight asphalt structures based on the amount of crosswalks present in the street and anywhere from zero to sixteen objects on the sidewalk. There are two types of street modules, horizontal streets and vertical streets. This distinction has been made due to placement problems rotation brought with itself. To connect the street modules to each other and the outside world the intersection module and the broken street module were created. The design of all street related structures were based on real life footage from example, Barcelona through the use of Google Earth [2]. The outline of the city is created in the perimeter module. The outline consists of a deteriorating sidewalk and wild growth.

- Horizontal street module
- Vertical street module
- Intersection module

[2]Google,(n.d.). Eixample. Retrieved from https://earth.google.com/

- Broken street module
- Perimeter module

4. RESULTS

Because there are no clear defined metrics to analyse our results, the results are in the form of pictures of our generated world. In Figure 2 the street view in a generated city is shown, this street view is from the same height as the top of the buildings. In this picture you can see a wide variety of buildings which use the same module, this shows that there can be still wide variety in pre-built modules. In Figure 3 a street view from ground level is shown which especially shows the streets.

In Figure 4 the entire skyline of a generated city is shown. The nine city blocks vary in their size and color which makes for a unique and interesting skyline. Finally, in Figure 5 a top view from a generated city is shown where nine city blocks are again clearly shown. Each one of these nine city blocks consists of eight corner and middle buildings. Both Figure 4 and 5 highlight the differentiation in buildings based on the same module.

Figure 2: Street view from the same height as the top of the buildings.

Figure 3: Street view from ground level.

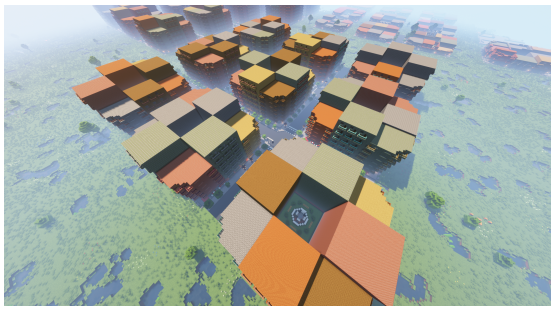

Figure 4: View of the skyline of a generated city.

Figure 5: Top view of generated city.

5. CONCLUSION

To recap, our research question was stated as:

Can we design a algorithm that generates believable settlements which are in line with the AI Settlement Generation Challenge standards?

In our approach we diverged from the idea of settlement and moved towards the direction of urban architecture. Where we looked at the city design of Barcelona and the use of city blocks. These city blocks in our design, as well as the streets and intersections, consist of pre-built modules. Using these modules we could build a wide variety of buildings in different heights and with different materials using only a few modules. However, the downside of our urban architecture is that is not quite adaptive to the environment. Terrains need to be flattened before we build our city.

6. DISCUSSION

Although we managed to create a believable settlement within the time given there are a lot of other interesting options to explore. The world of Minecraft is a vast, fun and an interesting one. There are many options to explore but like all research we must stop somewhere.

As everyone in our group never worked with coding in Minecraft before we all learned a lot during the process. With our increase in knowledge we now know some things could be done in a better designed way, including more procedural content generation. An example of this is to use any form of noise on the asphalt surface to make it more realistic looking than plain grey.

In Minecraft there are limitless possibilities for increased detailing and polishing. An example of this would be to increase of objects in the city (e.g broken down cars, traffic signs, house decoration). However our focus was on creating a finished product. In future work, adaptability to a previously unknown environment could be more in the focus. Even though our results are great, this is an aspect which we would like to put more thought on.

7. REFERENCES

[1] *The Peculiar Architecture And Design of Eixample, Barcelona*, 2013.

[2] Minecraft Wiki. *NBT format.*

[3] New York University. *Settlement Generation Challenge 2021*, 2021.