Generative Design In Minecraft (GDMC)

Tai-Wei Kuo, Tsu-Chuan Su, Wei-Cheng Chen, Wen-Yuan Wu

Advisor：Tsung-Che Chiang

National Taiwan Normal University

{lsktw22@gmail.com, subarya@subarya.me,
lapor.chen@cardinalblue.com, jw910731@gmail.com}

June 13, 2023

1 Abstract

GDMC (Generative Design in Minecraft) provides a method using the GDMC HTTP Interface to retrieve and modify block information in Minecraft game maps. Based on this, a settlement is established within the designated official map. The overall evaluation criteria revolve around four standards set by the official guidelines[1] [1]. These standards include adaptability to the environment and terrain, functionality in embodying cognition, storytelling that is believable and evocative, and visual aesthetics of the settlement and its structures. Additionally, GDMC offers bonus challenges, such as the Chronology Challenge, where participating teams can generate a story for the settlement and include it in in-game books within the storage containers. The research will utilize Agent-based Systems, Poisson Disk Sampling [2][3], A-star algorithm, and other techniques to accomplish the competition objectives.

2 Introduction

2.1 GDMC Rules

The GDMC competition involves generating a settlement within the Minecraft game world based on specific rules. These rules dictate various aspects of the generated settlement, including adaptability to the environment, functional features, storytelling elements, and visual aesthetics. The competition also provides additional challenges, such as creating a historical narrative for the settlement.

2.2 Principles of Settlement Generation

In Minecraft, players often choose areas with favorable conditions for their initial settlements, such as flat regions with water and trees that provide essential resources like food and shelter. However, the designed settlements should not only meet players' in-game needs but also consider the principles of real-life settlement design. These principles can be summarized as follows:

1. Availability of a water source, or the inclusion of irrigation wells or water storage facilities within the settlement.

2. Adequate brightness level (light level > 7) to prevent hostile creatures from spawning in the area. The brightness calculation takes into account the light source's brightness and the Manhattan distance between the light source and the target block [2].

[1]https://gendesignmc.wikidot.com/wiki:2022-settlement-generation-competition
[2]https://minecraft.fandom.com/wiki/Light

3. Presence of tree-covered areas since wood is a crucial material for early-game construction and tool crafting.
4. Adjacent buildings should have a maximum height difference of 1, considering players' jumping capabilities.
5. Preference for relatively flat terrain, as it facilitates the construction of houses and facilities.
6. In general, players tend to avoid settling and constructing buildings in special or ecologically unfavorable biomes. However, most standard biomes are suitable for settlement and construction, except for these exceptional cases: snowy biomes, cold biomes, temperate biomes, hot biomes, ocean biomes, cave biomes, Nether, and End biomes. Each biome category can further be divided into specific subtypes based on terrain and flora/fauna characteristics [3]. Most players typically choose initial spawn points in snowy, cold, temperate, or hot biomes to facilitate initial exploration and construction.

2.3 Quantifying the Quality of Settlement

In previous studies [1] [4], the quantification of settlement suitability based on environment and design factors has been proposed. The evaluation criteria for settlements in the GDMC competition can be categorized into four main standards:

1. Adaptability to the Environment and Terrain: This criterion assesses how well the generated settlement utilizes and harmonizes with the surrounding environment and terrain. Factors such as terrain elevation changes, proximity to water bodies, presence of natural features like hills or forests, and integration of the settlement into the landscape are considered.
2. Functionality in Embodying Cognition: This standard evaluates the functionality of the settlement in terms of meeting the cognitive needs of players. It includes factors such as the organization and layout of buildings, logical connections between different areas within the settlement, and the placement of functional structures like farms, storage areas, and paths.
3. Believable and Evocative Storytelling: The storytelling aspect assesses the narrative and immersive qualities of the settlement. Participants can create a historical background or story for the settlement, incorporating elements like signs, books, or structures that tell a compelling story and enhance the player's experience.
4. Visual Aesthetics: The visual aesthetics criterion focuses on the overall appearance and beauty of the generated settlement. Factors such as architectural design, landscaping, use of decorative elements, color schemes, and overall visual composition are considered.

Each of these criteria contributes to the overall quality of the generated settlement and will be assessed by the competition organizers and judges.

Additionally, the GDMC competition offers bonus challenges that participants can choose to take on. One such challenge is the Chronology Challenge, where teams can generate a coherent story for the settlement and create in-game books to store the narrative within the settlement's storage containers. This adds an extra layer of creativity and storytelling to the competition.

To generate the settlement, various techniques and algorithms can be employed. Agent-based systems can be used to simulate the behavior of different entities within the settlement, such as villagers or animals, to create a more dynamic and realistic environment. Poisson Disk Sampling can be utilized for distributing trees or decorative elements in a natural and visually pleasing manner. The A-star algorithm can assist in pathfinding and optimizing the layout of paths and roads within the settlement.

By combining these techniques and adhering to the evaluation criteria, we can create high-quality and visually appealing settlements within the Minecraft game world for the GDMC competition. Additionally, we provide the source code[4] and the video of the generator [5].

[3]https://minecraft.fandom.com/wiki/Biome
[4]https://github.com/NTNU-GDMC/GDMC
[5]https://youtu.be/j5x7trujfp0

3 Research Method

The method of settlement construction mainly The process of establishing the settlement can be divided into the following parts, please refer to the flowchart Figure 1.

1. Initialize each part of the system, then analyze the resources and terrain of the area. Generate the average height map, standard deviation height map, water map, resource map and biome map. We quantification the resources into numeral data that agent can analyze. For example, wood and stone.

2. The main loop. At the start of the round, each agent will use their own analyze function such as terrain smoothness, distance to water source. After the analyze the terrain, building and other numeral data. The agent will choose the best place to do their assigned job. Like place a building, or connect two buildings with a path.

3. After the interaction of agents and blueprint. The decoration agent will sample the area with Fast Poisson Disk Sampling [2][3] and decorate the settlement with decoractions like street light.

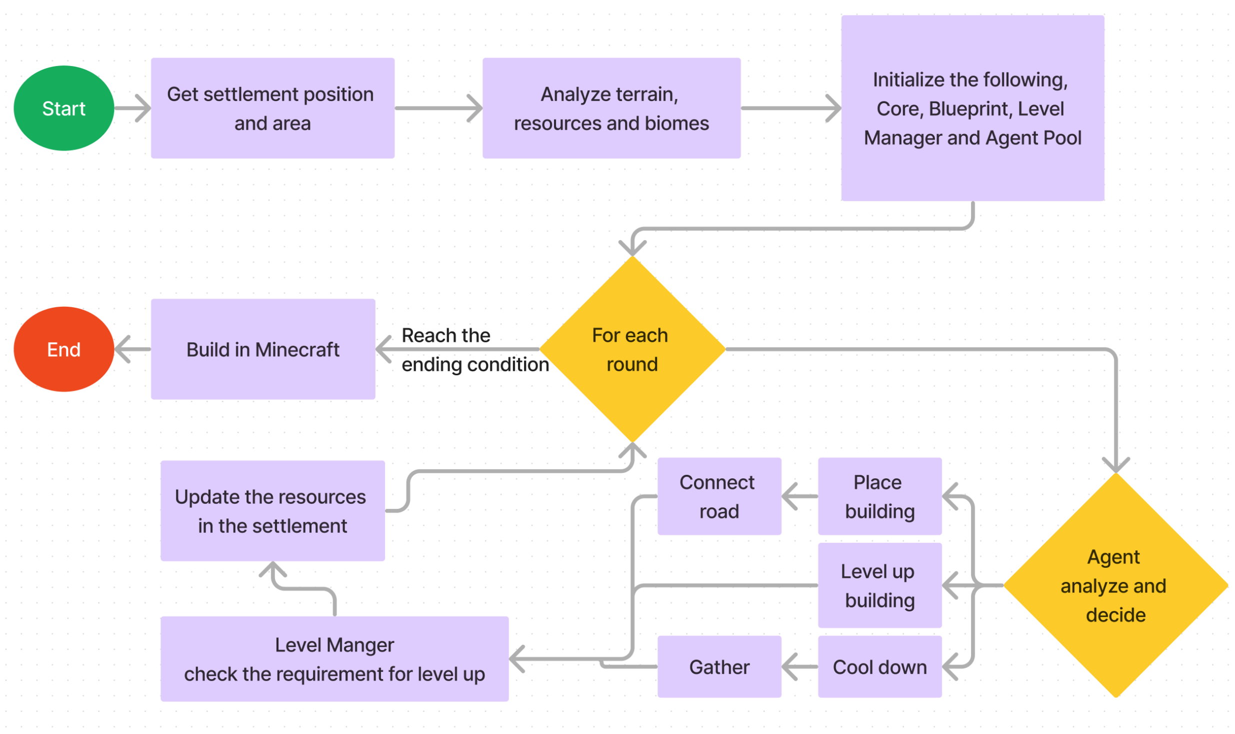

Figure 1: Flow chart of the program

3.1 Agent-based Model

We tried to plan the settlement as if it is an optimization problem. But it does not work when we combine many complex problems together, like building distribution, road connection and adaptability[6] to terrain and biome. In the end, we think this settlement generation cannot be seen as an optimization problem easily. We decide to use agent-based model to solve this problem. Make the agent simulate the behaviors of a player. To make building like what a player might do.

3.2 System Structure

Our system mainly divided into three parts, Core, Agent Pool and Level Manager.

3.2.1 Core

Core serves as the store of all datas, including: Map information, like terrain, resource, biome. System information, like level, resources we have, resource limit, blueprint and road network. Core provides relative search and update methods to agents. Allow the agents to analyze with these datas.

[6]https://gendesignmc.wikidot.com/wiki:2022-settlement-generation-competition

3.2.2 Level Manager

Level Manager manages the level up system for core. Provide query methods to search level up condition and the build limits of the settlement.

3.2.3 Agent Pool

Agent Pool is responsible of generating and managing all agents. Make every agent run in every round.

3.3 Analyzation of Terrain, Resource and Biome

Since core is responsible of recording every information, we need to pre-process raw datas from Minecraft and convert them into information we need.

3.3.1 Read

After we read data from Minecraft with package GDPC [7]. We can get 1. Height, 2. Height without walkable blocks, 3. Height without leaves and 4. Height without liquid. We can also query the information of a block in a specific position and analyze the surface of the map.

3.3.2 Analyze Terrain

We use the height map without leaves to analyze. We pre-process and calculate the two dimensional prefix array for height. Let us calculate the average height of an arbitrary area in O(1) time complexity. The benefit of using this height map is that we can make the use of the space under the leaves. Usually, there's enough space under the leaves allows player to walk through, even we break the leaves, it doesn't affect the terrain too much.

3.3.3 Analyze Resource

To analyze an area, we use blocks on the surface. Like leaves, stone, iron ore, grass, water and others. We make distribution graphs for these information, to let us determine the characteristics of the area.

3.4 Analyze Biome

We have categorized biomes into seven major types: Snowy, Forest, Jungle, Savanna, Dark oak forest, Desert, and Badlands. We have created biome distribution maps for each of these biomes to facilitate the querying of their respective distribution patterns.

3.5 Resource Storage and Level System

3.5.1 Resource storage

We make the agents simulate the gather and consume of the resources. If the resource is not enough for building, the agent will gather the needed resource and put it into the storage. And for the special buildings, like sawmill. They generate resources each round.

3.5.2 Settlement Level

In order to make agents have a clear target to make strategy, we create the level system, limits the maximum building in each level. When the settlement have enough buildings and resources, the settlement will level up and increase the limit and unlock new building types that allow the settlement to progress.

3.5.3 Building Level

Beside the settlement level, we design many level for building. Allow the agents to upgrade old buildings. Upgraded building will increase the resource it generates and the building scale.

[7] https://github.com/avdstaaij/gdpc

3.6 Methodology of Building Distribution

The steps performed by the Build agent are as follows:

1. Obtain Available Area List: Depending on the assigned building size, the agent retrieves a list of available areas from the blueprint.

2. Analyze Pre-Designated Building Areas: The agent analyzes all pre-designated building areas and assigns a score to each area based on various criteria. The area with the highest score is selected.

3. Select Highest Scoring Area for Construction: The agent chooses the area with the highest score as the location for construction and updates the results on the blueprint.

3.6.1 Blueprint

Using an empty map to simulate the settlement we planned the placement of buildings and roads, and call it blueprint. To reduce the computational load of the unused area. The unit of the blueprint is a 2 by 2 area. Although this method sacrifice the details of the map. Consider the performance of the program. We think this method is better.

3.6.2 Analyze Area

Analyze the biome map and choose the building that fits the area the best. For example, the building in the dessert biome would be different from the plain biome.

3.7 Building Generatation

3.7.1 The Material Replacing of Building

In the game, the material differs from different biome. The wood has the most obvious differences between biomes. To make the material of the building fits the area it is placed in. While building, the blocks will be replaced with the material that can be found in the area.

3.7.2 Easily replaceable material of building

To facilitate the construction of residential areas in different biomes, we initially employ a manual building approach to create the structures. Moreover, We utilize the in-game structure blocks to access and store the buildings within Minecraft, generating NBT (Named Binary Tag) files. This approach offers several advantages, including the ability to store the structures in the NBT file format, which enables convenient material swapping to adapt to different biome environments.

3.8 Road Network

Roads serve as significant pathways connecting building to building are essential infrastructure for residents' daily movement. In settlements, roads are indispensable foundations for facilitating transportation. The planning and establishment of road networks have a profound impact on the function of accessibility. Therefore, we cannot overemphasize the importance of road planning.

We consider the road network as an undirected graph, where buildings are treated as nodes, and the pathways between two buildings are regarded as edges. This allows us to understand the connectivity of roads and enables us to handle them accordingly. In addition, the buildings within a settlement are interconnected, meaning we can perceive them as connected components forming the settlement.

3.8.1 Monitoring Building Events

Once a building is constructed, the Core emits a notification, which is indicating the occurrence of the building event. The road agent notified the building event is responsible for connecting to the respective building. The road agent establishes connections between the new building and other existing buildings. If it connect the road successfully , it means that it will create an edge on the road network. However, if the connection fails, the building will remain on the road network, awaiting further processing.

3.8.2 Building Selection for Road Connectivity

When the Road agent decides which building to connect to, it uses a random selection method that takes into account the distance. Buildings that are closer have a higher chance of being chosen. This ensures that the final outcome is both logical and varied.

3.8.3 Popular Road

To create a visually cohesive and orderly road network, inspired by the idea that popular roads in reality become main arteries (such as highways), a concept is implemented where, upon the establishment of an edge, the hot value of all coordinates along the path is incremented by 1. This hot value encourages the road agent to preferentially traverse the roads.

3.9 Road Connection

To find a usable path within a short period of time, the road agent uses the A-star algorithm to connect roads, and the search process and results are optimized by employing the following methods.

3.9.1 Node Expansion

Nodes are expanded in all four directions: east, west, south, and north. However, the following situations will be considered impassable:

1. Due to player's jump height being 1 block in Minecraft, any height difference of 2 or more blocks is considered impassable.
2. There is already a structure on the blueprint, so avoid directly passing through it.
3. Avoid artificial structures because it is impolite to arbitrarily destroy other people's belongings.

3.9.2 Constraint of Search

It is very time-consuming to perform a full map search on a size of 1000x1000. Therefore, as a first step, we have imposed constraints on the search scope. Since taking detours is not an ideal outcome, we have selected two nodes as focal points and drawn an elliptical range to only search for nodes within that range. The method of check is straightforward: When connecting node A and node B, if the intermediate node being searched is node C and $\overline{AC} + \overline{CB} > 2\overline{AB}$, we consider it as being out of range and discard the search for this node. Doing so effectively avoids the problem of having disconnected nodes at close proximity that would result in a complete map search, significantly reducing unnecessary search time.

3.9.3 Actual Distance and Heuristic Function

We use the Manhattan distance as the distance calculation function, which make the road connect horizontally and vertically between building and building in the road network. The Euclidean distance, on the other hand, tends to make diagonal connections. Due to the more favorable results observed with the Manhattan distance, we have chosen to adopt it consequently.

Additionally, we have incorporated some weighting factors into the distance calculation. The adjustments we made are as follows:

1. To avoid excessive inclines in the roads, we mark the height (y-coordinate) dimension as the weight of 2, in order to make the agent avoid traversing paths with significant changes.
2. Agent strive to traverse the existing road network as much as possible. Therefore, we divide the distance by (1 + hot value) to make the agent tend to walk the paths with higher hot value, such as simulating the preference for traveling on highways.

3. Aim to minimize travel on water surfaces. Hence, we assign a weight of 10 to paths on the water, emphasizing that the agent should take detours of 10 additional steps on land rather than swimming through water.

4 Research Findings

Figure 2: Plains of Two Mixed Biomes.

Based on our implemented environmental analysis, during the construction phase, resource assessment is conducted in the vicinity of the building site. The agent prioritizes selecting locations with abundant resources nearby. As observed in the figure above, the building located in the rear right utilizes a different type of wood compared to the nearby structures.

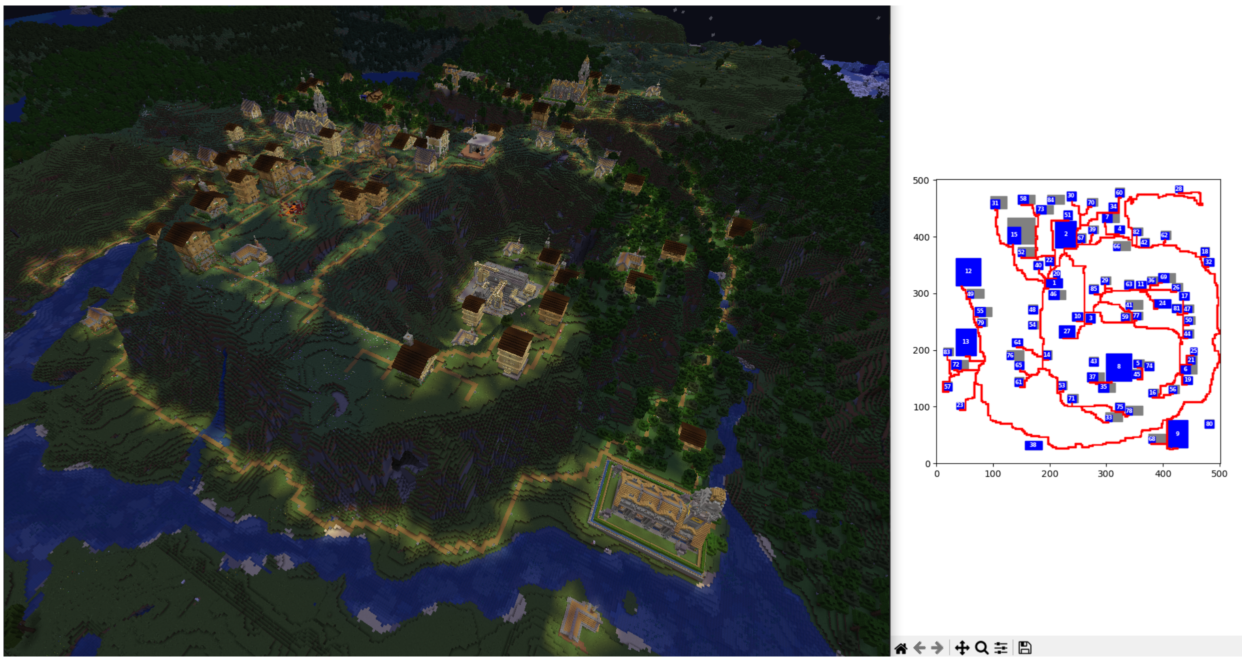

Figure 3: A plateau with a basin in the center. The blueprint for this residential area is located on the right side of the figure.

When determining the location for constructing buildings, we observed that the agent tends to prefer areas with relatively gentle terrain, such as plateaus, foothills, or basins. This approach helps avoid the issue of constructing buildings on steep cliffs. For road construction, we utilized the A-star algorithm. By configuring the weights appropriately, the roads exhibit a similar effect to those built by players. It's mainly because that Players typically choose routes that are flat and easily traversable. Taking the above figure as an example, the roads would encircle the outer perimeter of the plateau while ascending.

Figure 4: Coastal settlement where the majority of the buildable area is covered by water. It is evident that the agent, through analyzing the water area on the map, has arrived at the conclusion of avoiding construction on the sea and instead opting for building in flat areas along the coastline.

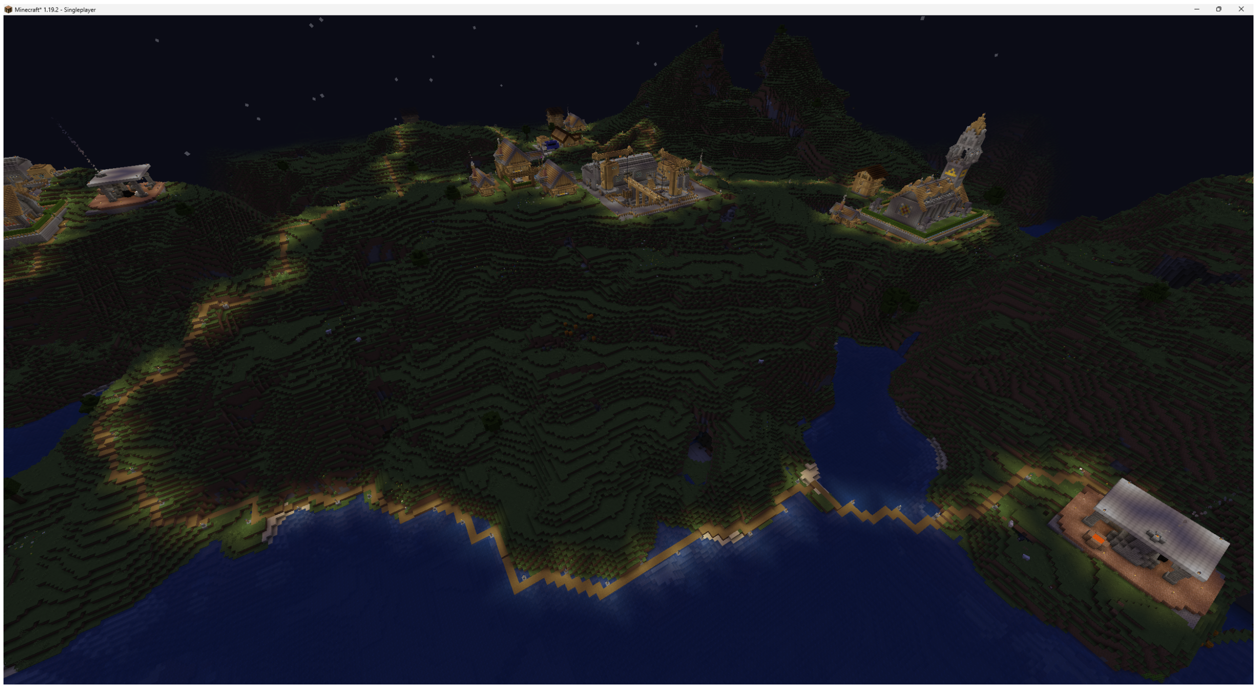

Figure 5: Coastal Highway
As shown in the figure, in order to connect the buildings on the plateau with those on the opposite side of the sea, the agent has designed a road along the coastline. This design avoids traversing steep mountain paths directly and, at the same time, minimizes proximity to build the road on water directly. Consequently, a road that follows the coastline was ultimately constructed.

Figure 6: Hybrid Biomes
Different biomes are characterized by unique architectural styles and building materials. For instance, in a desert biome, buildings are constructed using sandstone, while in forest and grassland biomes, the building material is the wood specific to those biomes.

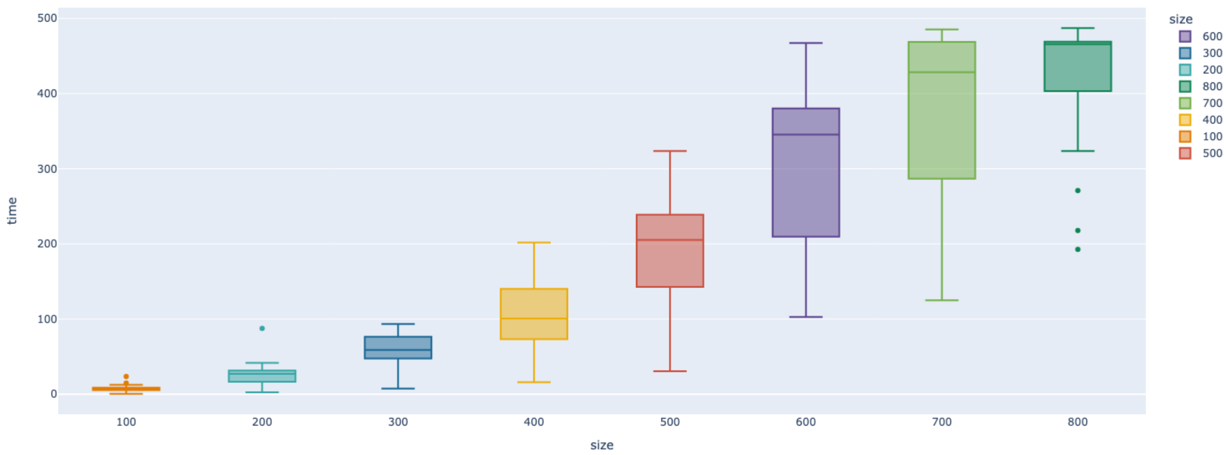

Figure 7: Size-Time Relationship Graph

When the map size is smaller, the Agent's speed in analyzing available areas and connecting roads is faster, and the time required for each round is also shorter, resulting in a reduced overall time spent. Additionally, according to this chart, we can observe that the longest execution time does not exceed 500 seconds. It's mainly because that we adhered to the maximum time limit of 600 seconds set by the GDMC competition and configured the longest execution time accordingly. Once it surpasses 465 seconds (7 minutes and 45 seconds), regardless of reaching the round limit, the loop is immediately terminated, and the analyzed data is sent to Minecraft to begin constructing the settlement.

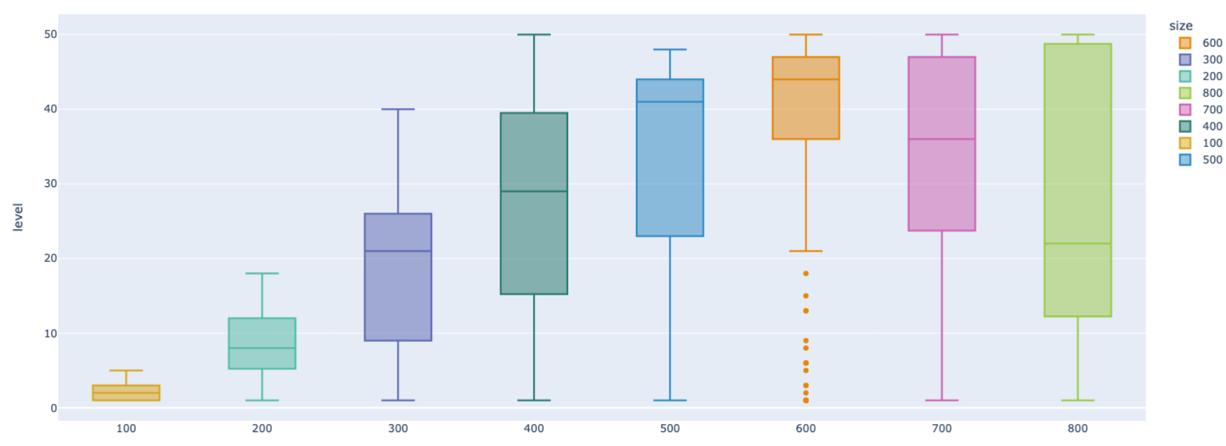

Figure 8: Size-Level Relationship Graph

In general, the leveling system varies on maps of different sizes due to the availability of space. The box plot clearly indicates a higher position of the median, which further demonstrates our effective utilization of the designated area without modifying the original terrain and objects in map. However, as the map size increases, there is a higher probability of encountering more extreme environments, such as rugged terrains and water surfaces. Consequently, this leads to a gradual decrease in the overall level distribution depicted by the box plot.

5 Future work

Currently, the road is connect on a 2d plane. Sometimes the building entrance is higher or lower than the actual road. The current solution is to raise the road to the height of the entrance, but this method makes unnecessary space under the road, which is not good for the adaptability. In the future, we want to implement the methods used in image processing, like Gaussian Blur, Neutral density filter, discrete cosine transform[5] and sharpen. Using these methods to smooth the terrain, allow the road agents to build more natural, smooth roads.

6 References

[1] Christoph Salge, Michael Cerny Green, Rodgrigo Canaan, and Julian Togelius. Generative design in minecraft (gdmc): Settlement generation competition. In Proceedings of the 13th International Conference on the Foundations of Digital Games, FDG '18, New York, NY, USA, 2018. Association for Computing Machinery.

[2] Robert Bridson. Fast poisson disk sampling in arbitrary dimensions. In ACM SIGGRAPH 2007 Sketches, SIGGRAPH '07, page 22–es, New York, NY, USA, 2007. Association for Computing Machinery.

[3] Daniel Dunbar and Greg Humphreys. A spatial data structure for fast poisson-disk sample generation. In ACM SIGGRAPH 2006 Papers, SIGGRAPH '06, page 503–508, New York, NY, USA, 2006. Association for Computing Machinery.

[4] Jean-Baptiste Hervé and Christoph Salge. Comparing pcg metrics with human evaluation in minecraft settlement generation. In Proceedings of the 16th International Conference on the Foundations of Digital Games, FDG '21, New York, NY, USA, 2021. Association for Computing Machinery.

[5] Melih C. Yesilli, Jisheng Chen, Firas A. Khasawneh, and Yang Guo. Automated surface texture analysis via discrete cosine transform and discrete wavelet transform. Precision Engineering, 77:141–152, 2022.