MGAIA Assignment 1: Procedural Content Generation

Dinu Catalin-Viorel

1. Introduction

In this assignment, we want to familiarize ourselves with Procedural Content Generation by generating structures inside of Minecraft World. We used the same work environment as in the Generative Design in Minecraft Challenge (GDMC (Salge et al., 2018))

2. House architecture style

In order to increase the believability of the building in the Minecraft world, we choose to create a medieval-style house. This style gave us more liberty in designing the house to incorporate multiple environmental characteristics. So, in figure 1 we can see a default version of the house we want to integrate into our world.

Figure 1. A generated default house on a flat map

2.1. House characteristics

The first observation is that we want to create a building that doesn't follow the traditional box houses, so we extended our house to have an L shape configuration. To randomize the house's configuration as much as possible, we consider random values in different ranges for the dimensions of the house. We define a house by four dimensions, as shown in figure 2:

1. The lengths of the long walls of the house L_1 and L_2
2. The lengths of the short walls of the house W_1 and W_2

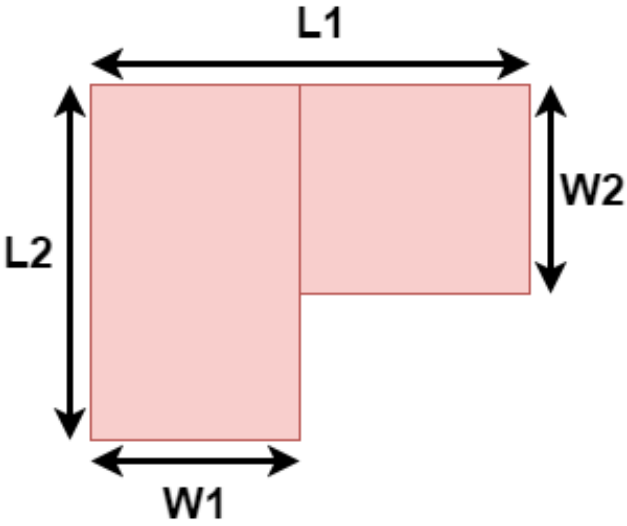

Figure 2. The dimensions of a generated house

Those dimensions provide multiple configurations that will result in different houses.
In addition to those dimensions, we consider all the possible rotations of the house. Those rotations can be seen in figure 3.

Additional to the L shape of the house, we have a custom rooftop composed of 2 sections:

1. A flat part
2. A triangle shape part that can incorporate a small attic.

3. House Generation

For different parts of the building process, we used a 3D filling algorithm(Alg. 1), which creates a list of coordinates containing similar block types.

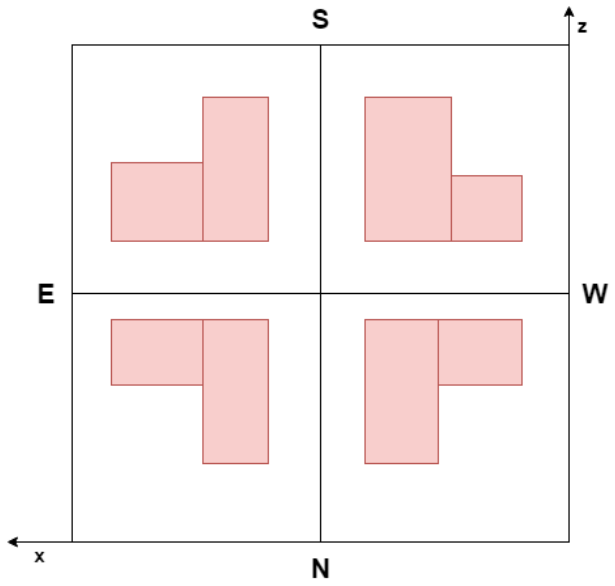

Figure 3. All possible orientations of a house

3.1. Area Selection

The first part of creating such a building consists of selecting an area that can fit such a structure. So, to incorporate this house into the environment, we generate a blueprint. Then we randomly select positions in the 100x100 building area to see if the area described by the point can fit the house described by the blueprint. An area is considered fit to incorporate such a house if:

- It doesn't contain any body of liquids(water/lava);
- The height that appears the most in the area is present for at least 10% in the whole area. We consider this as the main height ($\hat{h}$)
- The other heights in the area are ϵ apart from the main height($|\hat{h} - h| \leq \epsilon$), for our building, we consider $\epsilon = 4$, so the area consists of heights that are 4 blocks apart from the main height.

We consider the height to be the first block above the ground. This is achieved by considering the height map of the first air block and going down for any tree block we find in our path down.

3.2. Remove patches of trees

After we select an area, we have to prepare it for the structure. The first step is to remove any tree area inside the selected area.

Algorithm 1 3D filling algorithm

Input: $\mathbf{x}_{start}$, $\mathbf{y}_{start}$, $\mathbf{z}_{start}$, **directions, valid_types**
Result: visited $\leftarrow$ **a set of all positions with type in the valid_type set**
visited $\leftarrow [(x_{start}, y_{start}, z_{start})]$
$i \leftarrow 0$
while $i \leq$ len(visited) **do**
  current $\leftarrow$ visited[i]
  for direction $\in$ directions **do**
    position $\leftarrow$ current + direction
    if (position $\notin$ visited) $\wedge$ (type(position) $\in$ valid_types) **then**
      visited $\leftarrow$ visited $\cup$ [position]
    end if
  end for
  $i \leftarrow i + 1$
end while
Return: visited

To compute the trees in the area, we used the 3D fill algorithm to compute all the tree blocks of a patch. After that, we will convert those blocks to air. In order to increase the believability of the building, we are considering removing all the patches of trees that have some parts inside our selected area. Still, we restricted it to the whole 100x100 building area. This behaviour makes the area feel like the house owner cleared some area for himself.

3.3. Levelling the area

Secondly, for the house, we need a flat area to build the structure, so we consider three sub-areas of the selected structure position:

1. The area that contains only the main height ($\hat{h}$), which won't be modified at all.
2. The area that contains heights higher than the main height ($\hat{h} < h$). This area will be levelled down to the main height, maintaining the same top non-air block the same(this will increase the believability of the area). Also, we consider adding an extra block to increase the height of the border of the area if the outside one of the heights of the adjacent blocks is at least higher by two blocks.
3. The area that contains heights smaller than the main block($\hat{h} < h$). We will create a foundation for this area by raising the area using the same building blocks used for the house. In addition, to increase believability, we also converted the existing blocks in the area with adjacent air blocks, so it looks like the foundation is well fixed in the ground.

In figure 4, we can observe how an area looks after transformation.

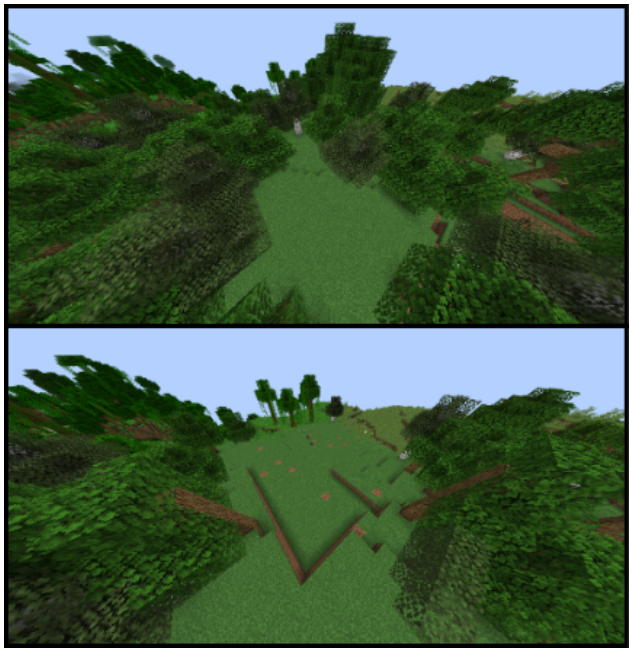

Figure 4. A before and after preparing an area to generate a house

4. Interior design and building blocks

We adapt our building to use the wood type for the building that appears the most in the 100x100 building area. The main stone-like block used for the building is computed randomly from a possible list. In figure 5, we can observe several generated houses in different areas with different types of trees. To select the spot for the **door**, we consider a

Figure 5. Different houses generated in different biomes with different types of woods

random position on the walls, a spot that is not a corner or adjacent to corners.

For **windows**, for any wall position that is not a corner, the door adjacent to corners or adjacent to the door has a probability of being a window or not.

For the interior design, we have two different approaches, one for the attic and one for the ground floor area:

1. **Attic**;
   For the attic, we mapped the whole area of the attic using the 3D fill algorithm, and for any block inside this area, we have a probability of placing a block from a predefined list of blocks(figure 6);
   We also consider two cases,

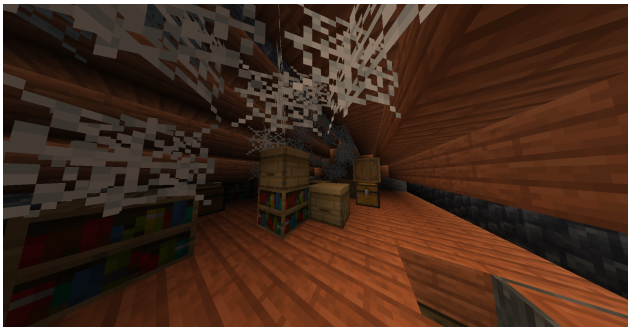

Figure 6. A random generated attic

- solid block below the selected block
- not solid blocks below the selected block

In figure 7, we can see all the blocks used in order to create an old dirty attic.

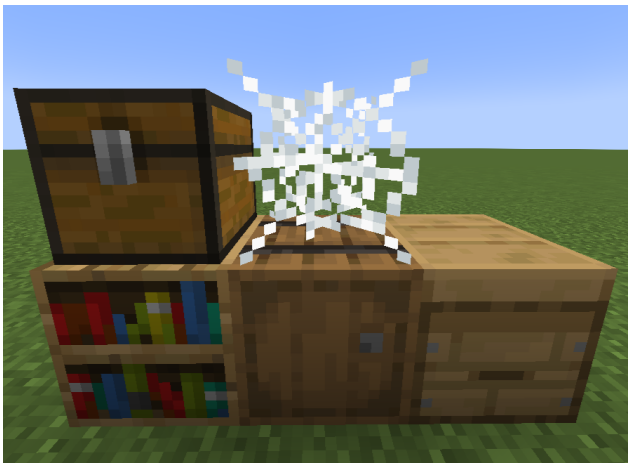

Figure 7. A list of blocks used to fill the interior of attic

2. **Ground floor**;
   Secondly, we have the ground floor, where we will randomly select areas of 3x3 or 2x2 that don't touch each other and fill them with predefined random structures (figure 8). In figure 9, we can observe several generated ground floors.

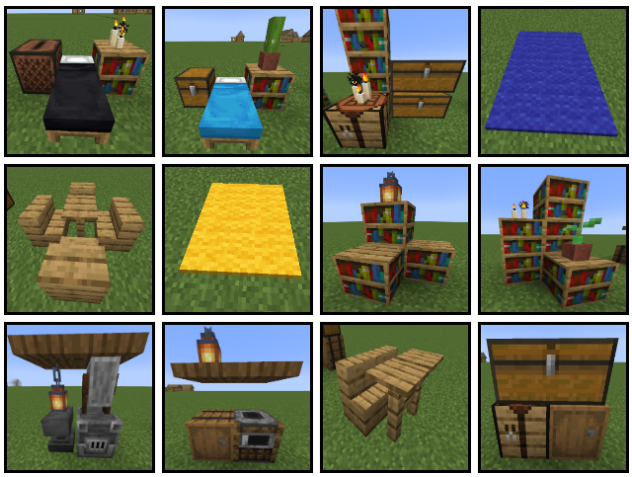

Figure 8. A list of all possible 2x2 and 3x3 structures that can appear inside a house

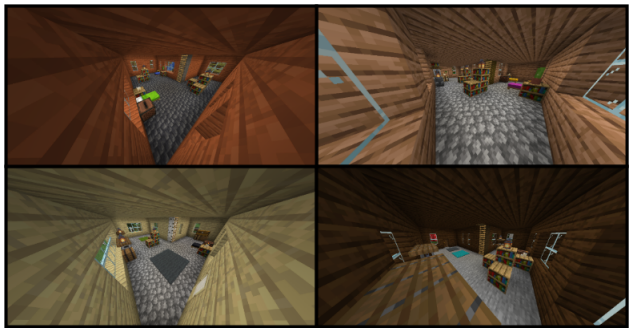

Figure 9. Several generated interior designs

5. Discussion and future work:

Our way of selecting the building blocks for the structure doesn't consider areas with different biomes with no trees. For example, for a dessert, we will create a default house, even though it may be better to have another predefined set of blocks. In future work, we want to map every biome to a specific set of possible building materials.
An interesting future addition would be to create additional types of buildings (towers, shops etc.) and to generate several of them and some paths between them creating a small village.
We also want to develop the algorithm for levelling the area of the house such that it can incorporate even more extreme cases like mountains and huge mines.
However, we observe in our building that there is some variation and adaptability in the current state of algorithms.

References

Salge, C., Green, M. C., Canaan, R., and Togelius, J. Generative design in minecraft (gdmc) settlement generation competition. In *Proceedings of the 13th International Conference on the Foundations of Digital Games*, pp. 1–10, 2018.