Minecraft Settlement Generator Description

Maël Galesne, Eduardo Hauck, Claus Aranha
University of Tsukuba, Department of Computer Sciences, Japan

UrbanSettlementGenerator[1] aims at generating modern cities, with a little extra here, given a Minecraft map. It generates houses and apartment buildings connecting all of them with roads. The buildings are generated according to a downtown/outskirts division, with apartment buildings placed towards the center and houses towards the edges of the map.

I started from the code of Eduardo Hauck's last year submission[2] and tried to improve it. I added new types of houses, the "LEGO" ones, new windows layouts for the buildings, floor lamps to roads and stairs to ease circulation.

Houses have walls made of double stone slab and pitched roofs made of wood. They have one oak door as the entrance on one random side, and glass windows on the walls perpendicular to the door wall. Houses are furnished with carpets, a table in the center of the room, a chandelier on the ceiling, and a bed, bookshelf and couch in the corners.

Apartment buildings have clay walls, 4 to 10 floors, with one apartment on each floor. The entrance to each building leads to a small corridor with a door to the ground floor apartment and a stairwell for reaching other floors. The apartments' furniture follows a similar fashion as in the houses, and windows are on the opposite wall to the building entrance.

To sum up, the generator first creates a height map of the first valid ground block for all coordinates x, z of the map. It then gets the width and depth of the space, and selects a percentage of the central area to become the city center. The remaining areas are divided into 4 sections to become neighborhoods. The center and the 4 neighborhoods go randomly through either binary or quadtree space partitioning. The partitions containing water, lava, etc or not reaching the set minimum size are eliminated as invalid. The remaining ones are ordered according to their steepnesses and become building lots. After all that, a building is generated on each lot, after checking if it does not intersect with any previous lot, until a certain minimum number of lots is reached for each of the 5 sections.

Before generating the building, the whole lot is flattened to the most occured height of it. Once it is done, generation proceeds at that height. The generation of buildings and houses are mostly hardcoded. For the houses, it is as follows: first, the land is cleared changing trees etc to air blocks. Then a random width and depth between a certain range, and the maximum height of the house are set. From that, floor and walls are generated, an orientation (N/S/E/W) is decided based on the position of the house, always facing the center of the map. The orientation will dictate the location of the entrance door, the entrance to the lot at the edge of it where roads will "connect", as well as the position of the windows. The orientation also sets the direction of the pitched roofs. Once the house is completed, the interior is furnished. For apartment buildings, generation happens in a similar way.

Once all the buildings have been placed, the last step is connecting them with roads. To perform this, we consider each lot as a node in a graph, and we create a minimum spanning tree to connect them. Distances between nodes are calculated with Manhattan

[1] https://github.com/djidanemael/UrbanSettlementGenerator
[2] https://github.com/ehauckdo/UrbanSettlementGenerator

distance. Once we have the minimum spanning tree, we use A* to actually find the path to generate the road. The heuristic function that gives the cost between the current and the target point is the Manhattan distance. The cost function between each node is $1+n^2$, where n is the difference of height between the current block and the next. We employed this function to try and find a path that is not too steep between two points (e.g. going around a mountain instead of climbing it).

The A* returns a sequence of coordinates 1 block wide where the pavement is to be generated. We complete the road on both sides of this path to make it 3 blocks wide. When doing that, we verify if the neighbouring block is not an invalid block (e.g. water), and if it is on top of air, we fill the blocks below recursively until getting to a valid ground block.

My addings are these:

"LEGO" houses made out of each type of wood. Their principle is to look like a lot of small rooms that we would have stacked. They have a various number of rooms in width, in depth and in height. High rooms are accessible with ladders. They have one oak door, a window next to it and a little path in front of them. They have a total of five different interiors from bedroom to kitchen. The roofs are made out of wood in the center and stone stairs on the outline.

Generating them is a bit different than houses. A random number of rooms is defined for the width, the same applies for the depth and so gives the dimensions of the final house. Then a loop goes from minimal to maximum height on each place where there will be a room and builds it with a random wood type and in either a random orientation or the same one for every room. For the interior, the first generated room of each floor is a bedroom and the others are random. This loop will eventually stop before the maximum height and add the roof on the top of the highest room in the place. Then two loops go through every rooms of the house. The first one adds the little path in front of them and the second one the ladders to reach them. For these ladders, in a lot of cases, they won't reach the ground. If this happens, it will try to add a ladder on an available side of the room below and so on. To finish the orientation of the whole house is defined by the side with the most doors on the ground floor and a path is generated on this side.

For the buildings, I changed the way windows are being generated so each building will have different windows layouts. The windows on the corridor side are the same on every floors but change following the size of the building. On the apartment side, they follow patterns following the size of the building and so should differ on each floor. For example, buildings 14 blocks wide will have three 2x2 windows and one 3x3 window on each floor while buildings 16 blocks wide will have two 2x2 windows, one 3x3 window and one 4x3 window in a different order each time.

For the roads, every 8 iterations of the loop generating a pavement block, a lamp block is added instead. A redstone block is added underneath to keep it alight. A stair is also generated when the road goes up or down by one block.