

# Procedural Building Name: Bergen House

Yanni He | VSFx 721 | Project 1 | Spring 2023

Houdini version: 19.0.383

Date: 4/19/2023

## Render

Renderer: Mantra

Average render time: 15 min per frame

Image resolution: 1920 x 1080

Number of lights: 2 (one distant light and one environment light with HDRI)

## Sampling

Min ray samples: 1

Max ray samples: 9

Noise level: 0.01

Global quality: 1

Diffuse quality: 1

SSS quality: 1

Reflection quality: 1

Refraction quality: 1

Volume quality: 1

## User's Manual

This tool can be used to customize the appearance of a house in the Bergen, Norway. Customizing the appearance takes place in the top level of the Controls node. A description of each control is as follows:

Side Window Gap/Front Window Gap: This parameter controls the proportion of windows. When this value is 0, there will appear all windows and keep normal proportion, when the value is higher, the proportion of windows will change.

\* Side window gap are set as random value, when this value exceeds 0.5 and is close to 1, all windows on the side will disappear.

Roof Height: This parameter controls the height of the roof. The default value is 10, but if you input 0 value, the roof will disappear.

Width: This parameter controls the width of the whole building. When the building gets wider, two different kinds of windows will generate along the x axis randomly. If the building gets slimmer, the remaining windows that don't fit will vanish.

Each Floor Height: This parameter controls the height of each floor. Recommend Value is between 10-20.

Depth: This parameter controls the depth of the whole building. As the building gets longer, two different kinds of windows will generate along the z axis randomly. If the building gets flatter, the remaining windows that don't fit will vanish.

Floor Num: This parameter controls the total number/amount of floors. When this value get higher, more floors will be generated and the whole building will change the height automatically.



# Beyond the Requirements

## - Color Variation



The houses in Bergen, Norway have distinct difference colors, but limited to red, yellow, white and brown. For the scattering, my purpose was to random the color of main part of each house but without changing the color of tiles, doors and windows.

My solution is grouping each part of the house first. `group:base_1` `group:basefram` `group:roofface_1` `group:tiles_1`

Then using **Pack** and **Attribute** adjust Integer nodes. I set the random value from 0-4 on each points so that when all the houses are copied, each house will have one attribute named **color**.

|    | P[x]     | P[y]    | P[z]     | color |
|----|----------|---------|----------|-------|
| 0  | -500.0   | 41.9526 | 0.873433 | 2     |
| 1  | -357.915 | 29.6884 | 0.618098 | 3     |
| 2  | -215.831 | 53.6808 | 1.11761  | 1     |
| 3  | -73.7465 | 37.3687 | 0.777999 | 1     |
| 4  | 68.338   | 29.6683 | 0.61768  | 3     |
| 5  | 210.422  | 35.2111 | 0.733079 | 2     |
| 6  | 352.507  | 33.0414 | 0.687906 | 2     |
| 7  | 494.591  | 47.889  | 0.997027 | 4     |
| 8  | 636.676  | 56.7232 | 1.18095  | 4     |
| 9  | 778.76   | 44.9922 | 0.936717 | 4     |
| 10 | 920.845  | 32.3749 | 0.67403  | 3     |
| 11 | 1062.93  | 33.1489 | 0.690144 | 4     |
| 12 | 1205.01  | 58.4485 | 1.21687  | 0     |

Then I used **Unpack** node(\*this step is to restore the color of tiles on the roof, door and window) and transfer the color attribute generated before. So I can random and adjust the color of the main part of each house but not influencing other parts.

## - Randomize the Position of Windows on Each Floor



There are two different types of windows randomly appearing on every floor (except the first floor). I created two windows first, then gave them one attribute named `window`. The value is an integer from 0-1.



Then using For Each Loop to controls each floor generating with iteration from 0. For the points on the line, I gave them the same attribute '`window`', but different random value.



(Special Thanks for Professor Deborah and Tempest teaching me this way to do the randomization.)

## - Change the proportion of windows

I used one attribute wrangle node to control the proportion of all windows, which can make a gap between each window. I created one channel(filter) which can remove the points on the line.

```
VEXpression
1 if (@ptnum = (@ptnum)/chf("filter")){
2   removepoint(0,@ptnum);
3 }

VEXpression
1 if ( random(@ptnum) < chf("filter")){
2   removepoint(0,@ptnum);
3 }
```



(The Parameter "Front Window Gap" From 0-1)



(The Parameter "Side Window Gap" From 0-1)

## - Randomize the Scale of Each House When Scattering Using VEX

```
VEXpression
1 @pscale = fit01(rand(@ptnum),0.5,1);
```

## - Texturing the Main Part

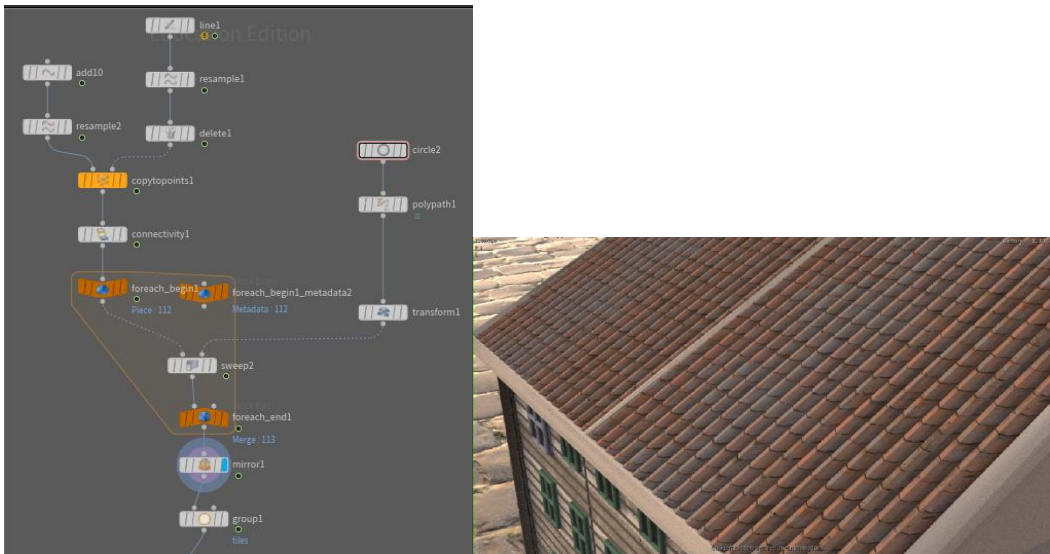
Using UV Texture and UV Quick Shade on each part. The UV tiling ensures that even if the main building is stretched, the UV will not be affected.





## - Making the Tiles On the Roof

Using for each loop to create each tile model on the roof.



- In addition to width, depth, and height of the house, I also created height of each floor, roof height variables.

## Reference

### Houses in Bergen Norway



