Documentation OSB3D

How to add template blocks, building types, cars, and materials to the Unity project.

1. Assets for Road Generation

1.1 Adding Street Items

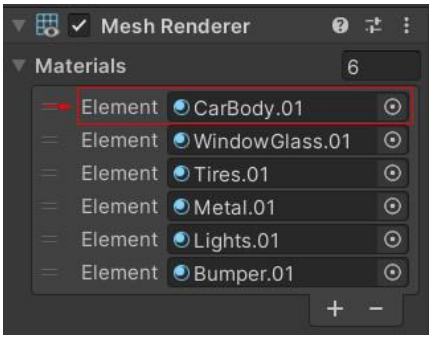
Add prefab to Resources > Prefabs > StreetItems
By default, the Street items are set with a minimum spacing of 3f, if larger items are used more spacing might be needed to avoid overlaps. Then change the variable itemSpacing in the RoadGenerator, set in Setup(). The script can be found in Scripts > CityGeneration.

1.2 Adding Cars, 3D models

Add prefab to Resources > Prefabs > Cars
By default, the Street items are set with a minimum spacing of 7.5f, if larger cars are used more spacing might be needed. Then change the variable carSpacing in the RoadGenerator script, set in Setup(). The car will get randomly assigned material, for more information on material selection in section 1.3.

1.3 Adding Cars, body material

Add prefab to the folder Resources > Materials> Cars
In the mesh model, the car body needs to be the first material in a list of materials. Additional materials will not be assigned, then the original material of the model will be used. It is therefore recommended to have tires and window glass as additional materials.



1.4 Adding Street Trees

Add prefab to Resources > Prefabs > StreetTrees

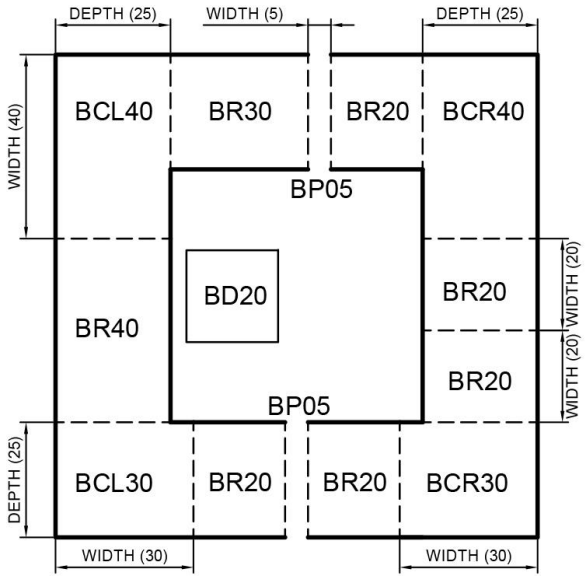
2. Assets for Block Generation

2.1 Included Building Types

The building prefabs are divided into folders by their building type. By default, the types included are BCC30, BCC40, BCS30, BCS40, BD20, BD30, BP05, BP10, BP15, BP20, BR20, BR30, BR40, BRV20. For information on how to add additional building types, see section 2.1.

The first letters in the code denote the building's position in the block and the following numbers are the buildings width. By default, the buildings are 25 units deep, except for the building types BD, BP and BRV which can have any depth. One unit in Unity is by default one meter.

BCL = Corner building left
BCR = Corner building right
BD = Detached building, for example the buildings inside the courtyard
BP = Passage, a part of the block that do not by contain a building by default and a door can be placed
BR = Row building in the block
BRV = Row building variant, has another depth than the standard row building



Example block for demonstration of building types (not implemented)

2.1 Adding Buildings, 3D-models

Add prefab to the folders for corresponding building type in Resources > Prefabs > Buildings
Either buildings can be added to the existing building types or new building types can be created. To create a new building type, create a new folder in Resources > Prefabs > Buildings, named with your new building type code, and place the prefab in the folder. Then add the new building type code in the List<string> buildingCodes in the BlockGenerator-script, it is set in Setup(). The script can be found in Scripts > CityGeneration. To get correct assignment of Materials, see section 2.2.

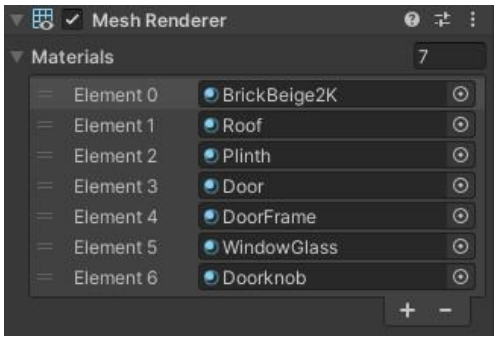
2.2 Adding Buildings, materials

Can be added to Resources > Materials > Buildings to the appropriate folder.

For correct material assignment, the material order in the mesh renderer of a building needs to be:

1. Façade
2. Plinth
3. Door
4. Window frames
5. Roof

Any additional material, such as material for window glass, doorknob, or stair, should come after, then the original material of the 3D-model will be used. By default, the door and window frame are set to the same material. This is controlled in the SetBuildingMaterials() method in the BlockGenerator class. This does not apply to passage-prefabs (BP05, BP10, BP15, BP20). For those, the only material assigned is the façade material and they are handled by the DoorPlacement class.

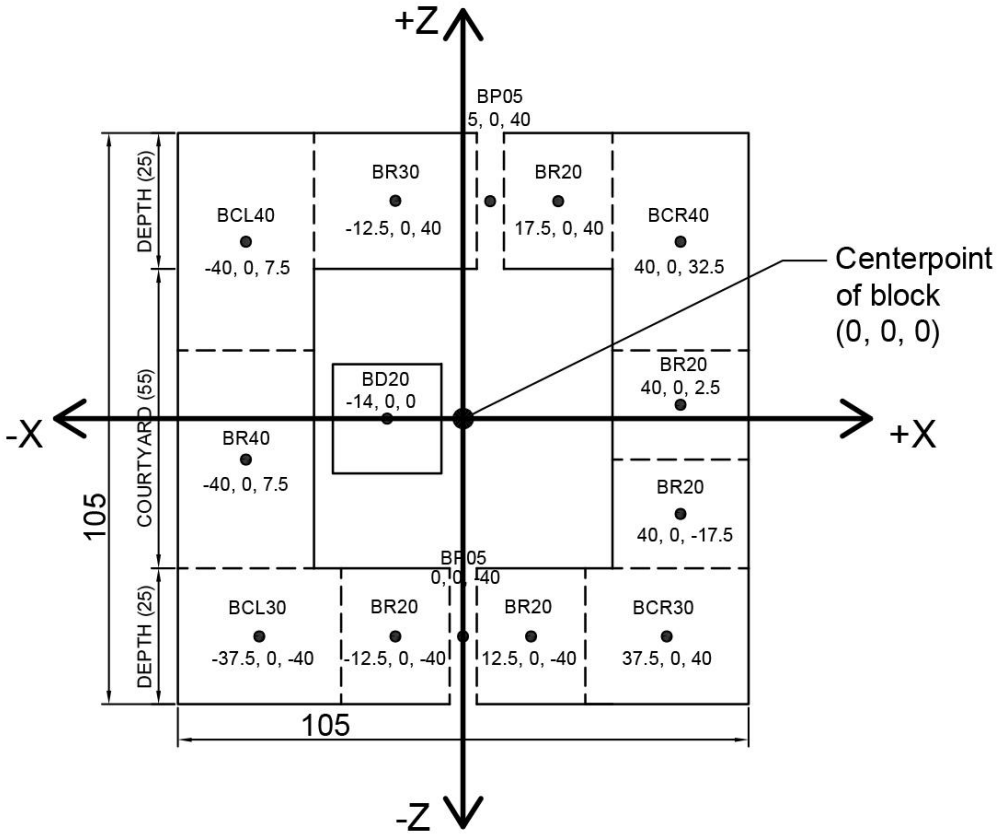


2.3 Block Template files

Can be added to Resources > BlockTemplates
Needs to be a .txt-file with comma separated values. The first row in the file is assumed to be a header and will be ignored. Subsequent rows entail information on the buildings in the block, with a row for each building or passage in the block. Each row should include the following information in the following order, without spacings;

buildingCode,positionX,positionY,positionZ,rotationY

The first value, buildingCode, is which type of building should be placed, read more about building types in the section 2.1 and 2.2. The following values are that buildings position and rotation in the block, when the center of the block is placed at (0, 0, 0), where the coordinates are in the order of (x, y, z). The coordinate should note the building's origin point, the origin points of the buildings included are at the center of the building in the x/z direction and in the bottom of the building in the Z direction. By default, a block is assumed to have the size of 105 *105 units. On the picture below the positions of the example template from 2.1 is demonstrated.



2.4 Template block with generated terrain

The positionY value, for the y coordinate, should be denoted as t in the template if the buildings are going to be placed on a generated terrain. If a generated terrain is going to be used for the block, all buildings in the block should have t as the y-coordinate.

Terrain variables can be set in the Unity Editor on the “Block Terrain”-script. The variables are:

Resolution of the terrain. Lowest is preferred to avoid performance issues.
Scale of the Perlin noise used to generate the terrain.
Min Height of the terrain, by default set to 0.
Max Height of the terrain, by default the lowest value is 0.5. This can be changed in the script.
Edge Distance for the gradient used to blend the edges of the terrain.
Material to use for the terrain.