BuildingMakerToolset

Manual v1.01

Content

1) General Workflow

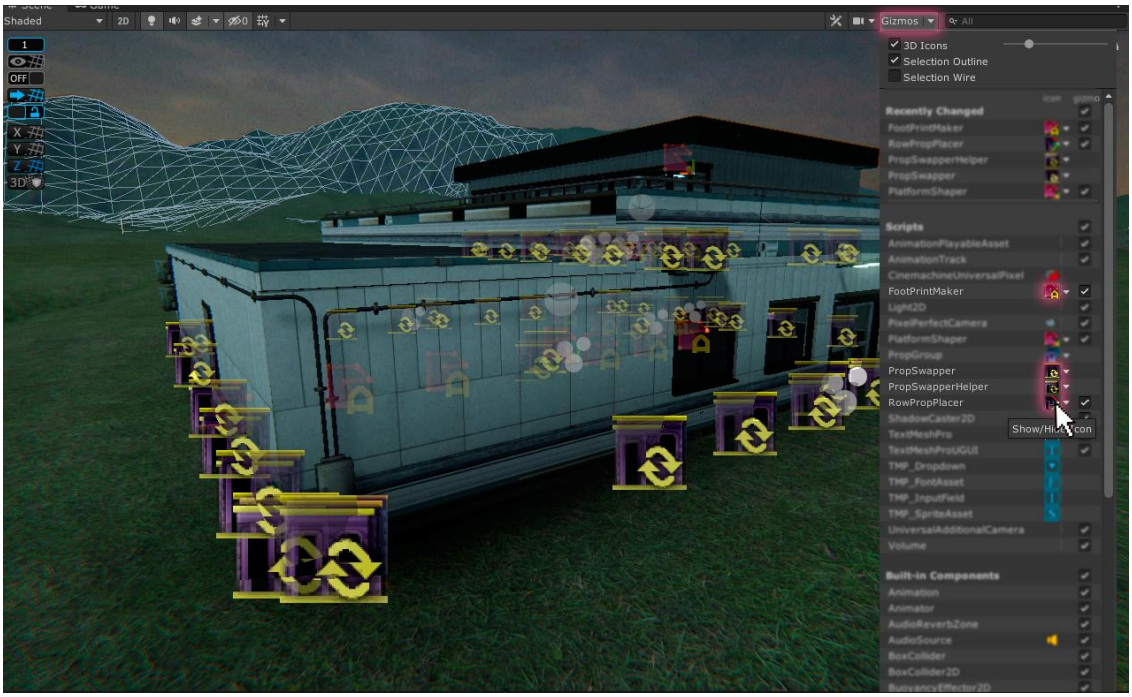
1. Use RowPropPlacer to place the walls of your building.
2. Use FootPrintMaker and PlatformShaper to create a floor and roof for your building.
3. Use PropSwapperHelper to add doors and wall trims.
4. (Optional) Use PropCombiner to combine meshes of similar material.

2) Integrating into existing Project

Gizmo setup

To get rid of the icon clutter, hide icons for the following scripts:

- RowPropPlacer
- PropSwapper
- PropSwapperHelper
- FootPrintMaker



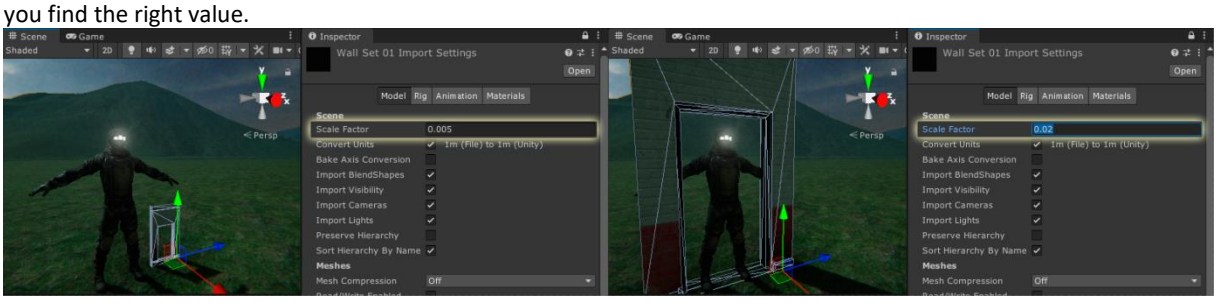
Scale setup

While creating your buildings, you need to make sure that the model scale is right.
There are a couple ways to scale the models, one of them is to rescale the Transform every time you drag a prefab into the scene. Since RowPropPlacer copies the scale to its placed objects, you don't need to rescale them after placing it. Another way, is to create the whole building first, then parent it to an empty GameObject and scale that GameObject until it fits.
Or you can change the Scale Factor of the model Asset, which is the most optimal way if you are creating a lot of buildings, since you skip the step of scaling the objects again and again. This toolset takes aware of the Scale Factor and it remains usable after you change it.

> Note: If you change the Scale Factor of the model Asset, buildings created prior to the change will become unusable. Make sure you have the right Scale Factor before you start creating buildings.

Changing the Scale Factor of the model assets:

Add a character mesh of your game to the scene as a scale reference. Find a wall prefab with a door and place it next to the character mesh. Select all source models under "BuildingHelper\Mesh\FBX" and change their Scale Factor until you find the right value.

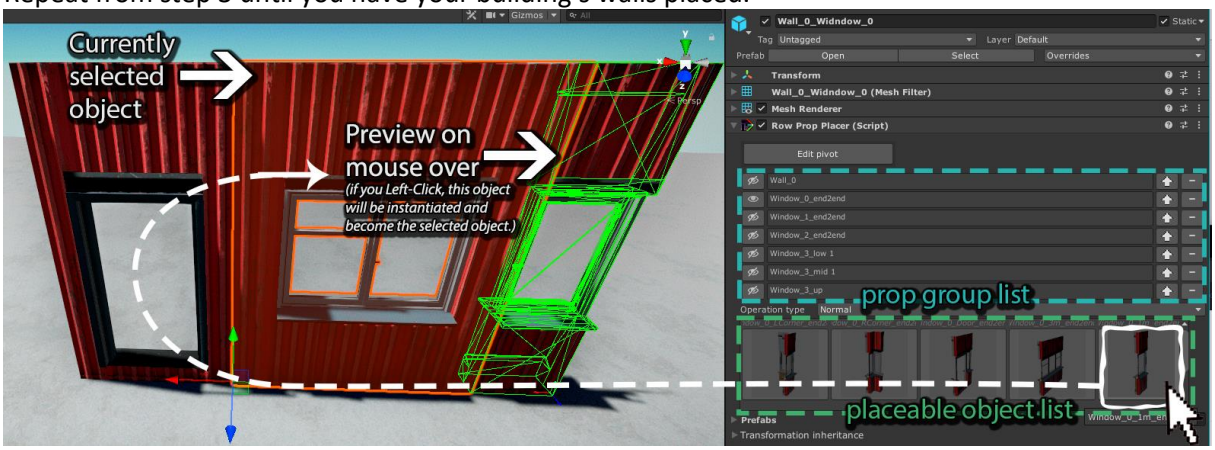


3) RowPropPlacer

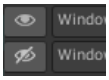
This tool provides a custom workflow to place props. Place walls, pipes, railings, etc. by clicking associated buttons in the Inspector. RowPropPlacer will automatically align the instantiated objects.

RowPropPlacer Workflow:

1. Drag and drop a Prefab, with a RowPropPlacer-script attached, from your Project window into the scene.
2. With your placed object selected, scroll down in the Inspector and find the RowPropPlacer-script.
3. Click one of the icons to instantiate the associated Prefab.
4. Repeat from step 3 until you have your building´s walls placed.



The *Placeable object list* displays all the props added via PropGroups or regular Prefabs.



If Prefabs are organized in PropGroups, their visibility, for the *Placeable object list,* can be toggled. This is useful if you have many Prefabs and you want to filter them by groups.

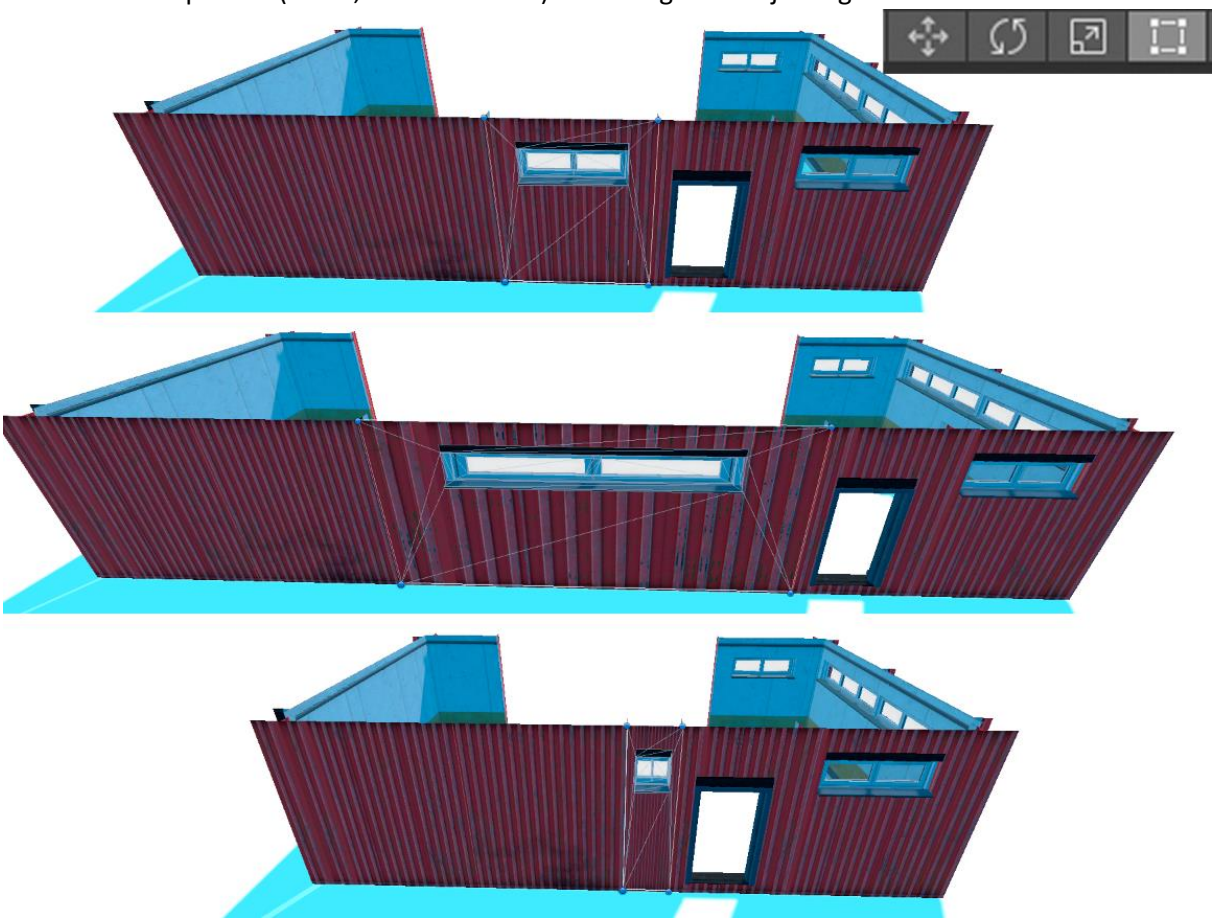
Tip: If you want to use custom Prefabs, you can drag and drop any Prefab or PropGroup on Placeable object list, to use them with RowPropPlacer.

Operation types determine Transform parameters for the GameObjects instantiated by RowPropPlacer.
Operation types:

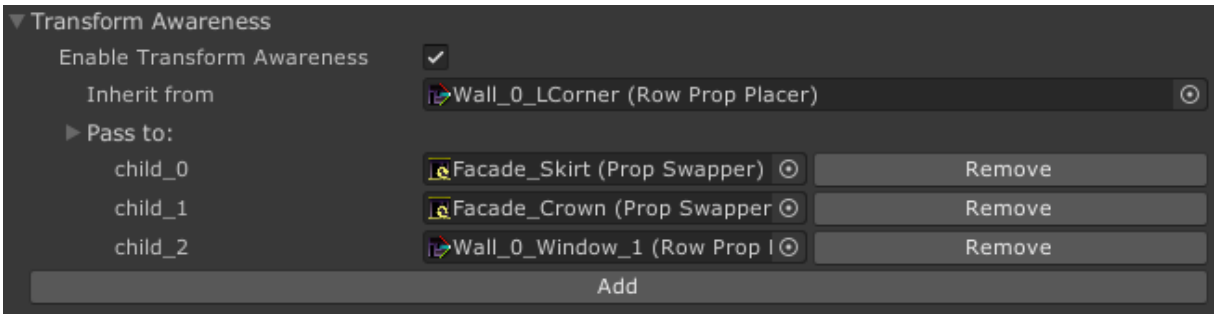
- Normal:
  The position, rotation and scale of the Placed-object is that of the Source-object-transform plus the offset determined by Source-object-spawnPivot.
- Reverse:
  The position, rotation and scale of the Placed-object is that of the Source-object-transform minus the offset determined by Placed-object-spawnPivot.
  (*Opposite direction of the normal operation*)
- Turnaround:
  The position, rotation and scale of the Placed-object is that of Source-object-spawnPivot plus rotation of 180 degrees on the world Y axis.

Transform Awareness

This is a very useful feature of RowPropPlacer and PropSwapper. It registers any changes to the Transform component (move, rotate or scale) and realigns its adjoining walls.

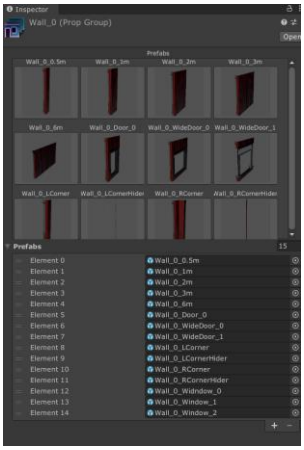


You can disable Transform Awareness in needed.



4) PrefabGroup

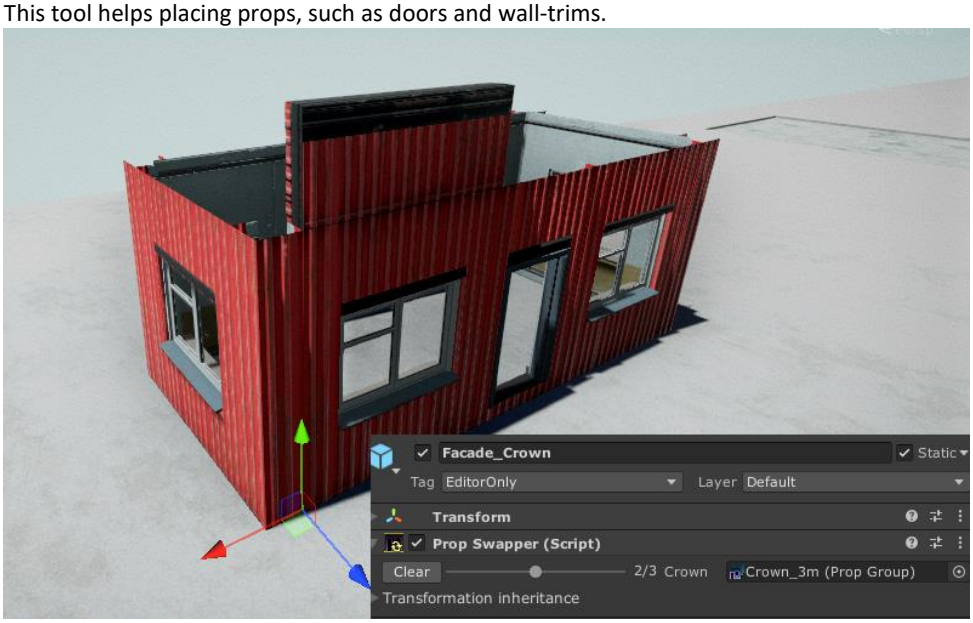
This is a Asset which holds a list of Prefabs.



It is used by RowPropPlacer and PropSwapper.
You can create a PropGroup with Right-Click>Create>BuildingHelperToolset>PrefabGroup
If you have a PropGroup selected, you can add Prefabs to it by drag and drop.

5) PropSwapper

This tool helps placing props, such as doors and wall-trims.



With the slider you pick one Prefab from the assigned PropGroup.
The position, rotation and scale of the placed-object is the same as the Source-object-transform´s.

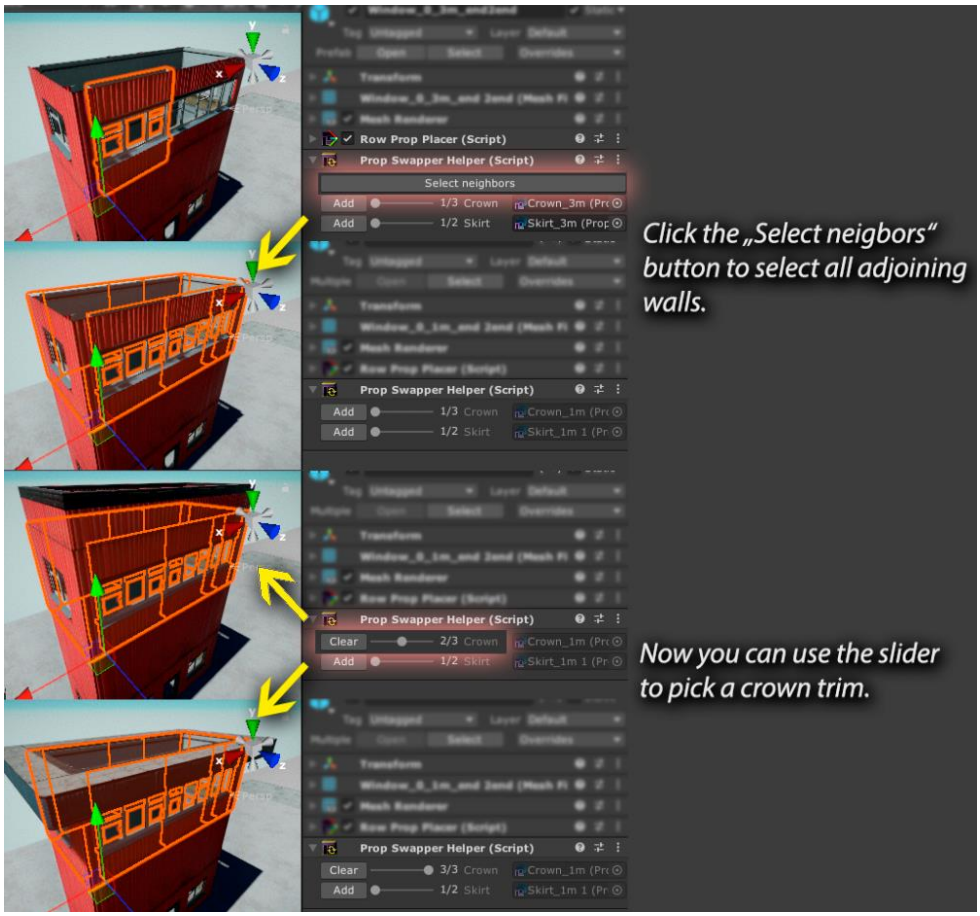
6) PropSwapperHelper

This tool helps to operate multiple PropSwappers at the same time.

If you select a GameObject with a PropSwapperHelper component, the PropSwapperHelper-inspector will collect all PropSwapper components in its hierarchical children and display them in groups. The “Select neighbors”- button will add adjoining walls to the selection. So that you can, for example, pick a wall-trim for the whole building with just one slider.

> Note: The “Select neighbors”- button only available if RowPropPlacer is attached to the same GameObject.

PropSwapperHelper Workflow:



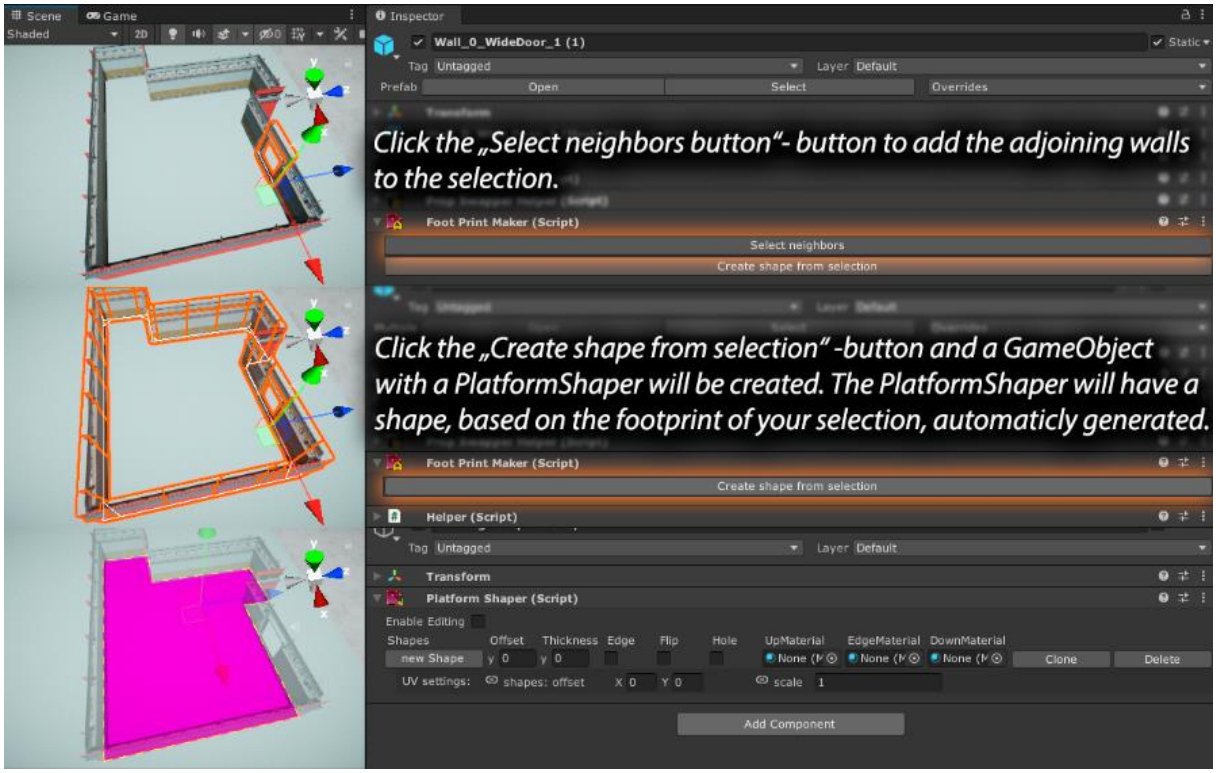
The grouping logic works like this:

1. All children-PropSwapper objects of all selected PropSwapperHelpers are collected into one group.
2. All PropSwappers in this group are getting a tag, with is the name of their assigned PropGroup, where all characters after and including this “_” are removed. (For example: *“Crown_1m, Crown_3m, Crown_6m”* will be tagged: *“Crown”*). The group is splited if the tag is different.
3. The groups are further splited where the slider position is different.
4. Now the groups are displayed in the inspector ready to be worked with.

7) FootPrintMaker

Creates a PlatformShaper with a shape based on the footprint of your Building, which can be used as the floor or ceiling.

Workflow:



Note: The "Select neighbors"- button is only available if RowPropPlacer is attached to the same GameObject.

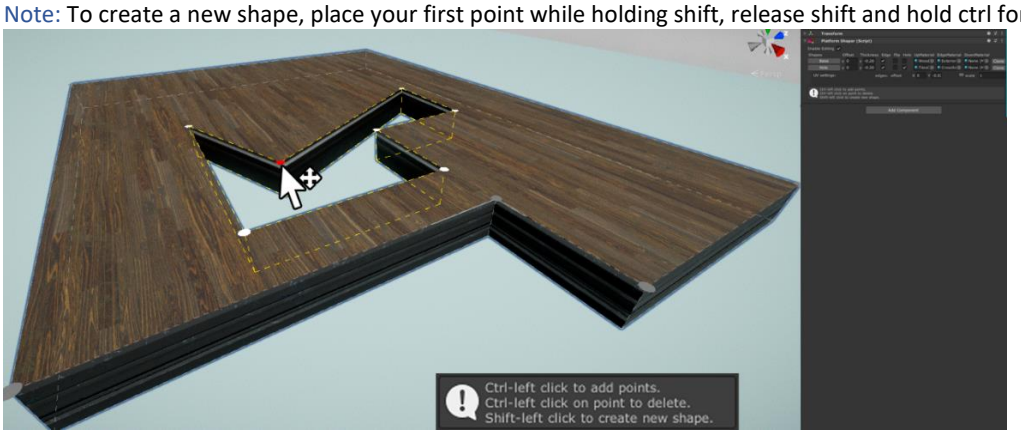
8) PlatformShaper

Mesh generator for making platforms.

PlatformShaper Workflow:

1. After you have placed the walls of your building, select one of the outer walls, scroll down in the inspector and find FootPrintMaker-script, click the “Select neighbors”- button and click “Create shape from selection”-button.
2. Edit the generated shape if needed.
3. Add a floor material to the “UpMaterial”-field and change the shapes name to “Floor”.
4. Click the “Clone”-button and change the height-offset of the new shape.
5. Change the name of the new shape to “Roof” and enter some value for the thickness.
6. Add a roof material to the “UpMaterial”-field and a ceiling material to the “DownMaterial”- field of the roof shape.
7. If you want to create a second floor, create a new shape with the same offset and thickness as your roof shape and toggle it as hole. Then you can build a stair using RowPropPlacer.

Note: To create a new shape, place your first point while holding shift, release shift and hold ctrl for the other points.



PlatformShaper Inspector

1. Enables shape editing and **locks selection if checked**.
2. Shape name.
3. Shape height offset.
4. Shape thickness. If zero no backfaces will be generated.
5. If checked and thickness is not zero, edge/side faceses will be generated.
6. If checked, all face normals will be flipped.
7. If checked, shape will cut a hole in overlapping shapes.
8. Top face material.
9. Edge/side faces material.
10. Bottom face material.
11. Copy of shape button.
12. Delete of shape button.
13. Shape UV map parameters

PlatformShaper UV Settings

1. Up/Down face independency toggle. If enabled, up and down face share the same UV parameters.
2. UV position offset.
3. UV scale.
4. UV scale axis independency toggle. If disabled axis scale can be applied independently.

9) PropCombiner

With this tool you can combine several Meshes of the same material into one Mesh. LOD Groups are taken into account and are also combined. PropCombiner will only combine objects that are marked as static, dynamic objects will be ignored. When combining, the renderers of the original meshes are simply deactivated, other components such as colliders or lights are not affected and retain their functionality. Combined Meshes will have their lightning UVs recalculated and can be used with baked lightning.

PropCombiner Workflow

1. Finish your building.
2. Create a new GameObject with PropCombiner attached and make it the parent of all objects that make up your building.
3. Click the “Pack” button to combine the building´s meshes.
4. If you want to make some changes to the building click “unpack”, it will reverse the combining action.

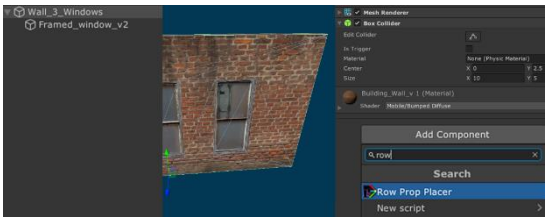
10) Using BuildingMakerToolset with other modular sets

You can combine this toolset with any modular set and make it a lot more useful. In this example I will add RowPropPlacer, PropSwapper and FootPrintMaker to a set of models from the Asset Store.

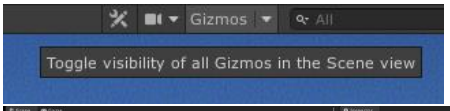
This are the prefabs I am using:

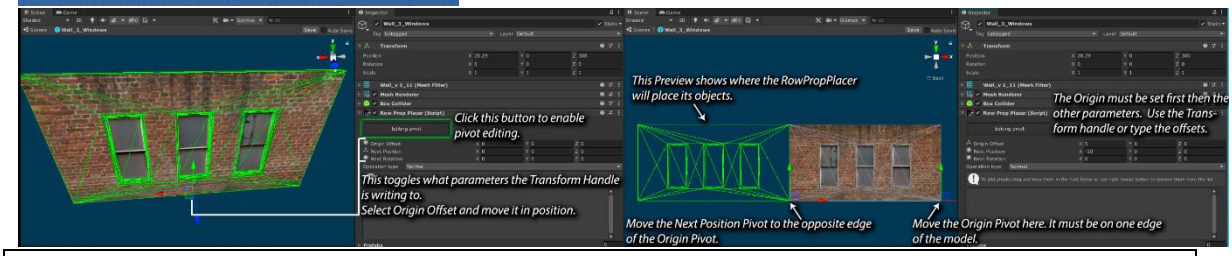
1. Import your models and make sure the Scale Factor is correct.

2. Open one of the prefabs in **Prefab editing Mode** and add a **RowPropPlacer**-component to its **root** GameObject.

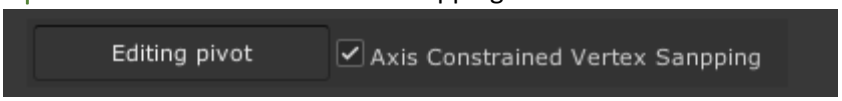


3. **Enable your gizmos** and set up the pivots:



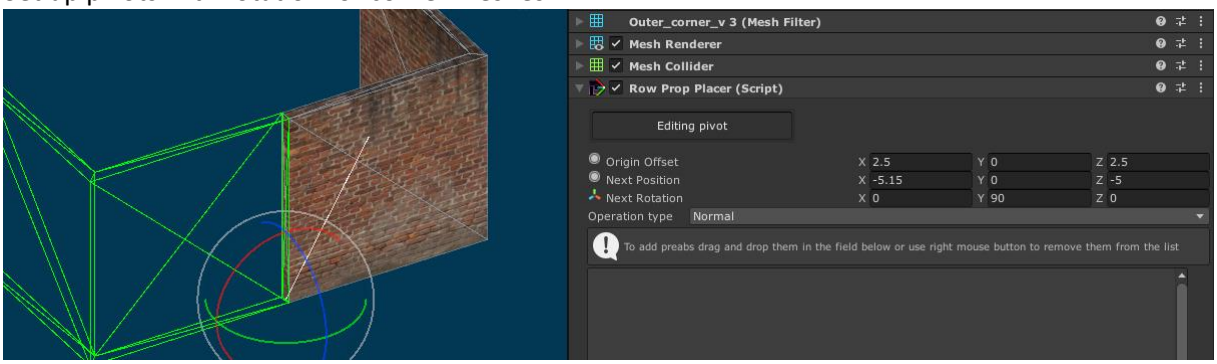


Tip: Use Axis Constrained Vertex Snapping.



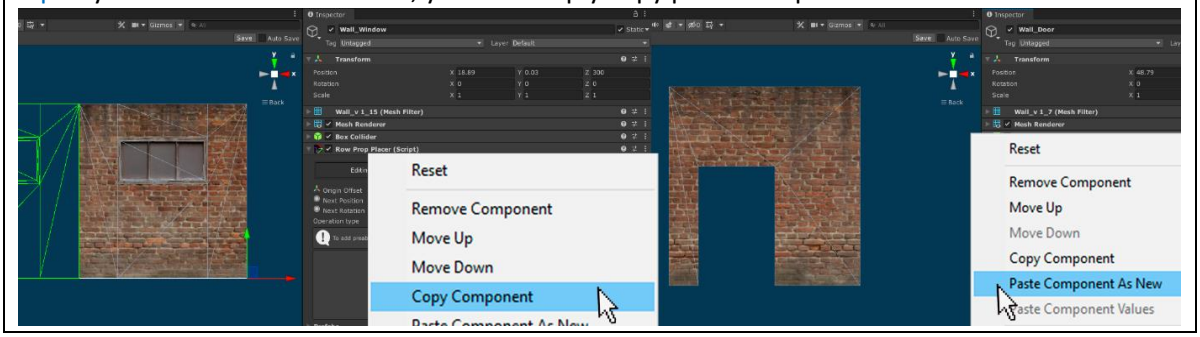
Grab the Axis of the Move Gizmo that you want to move, drag the cursor to any Vertex of the selected Mesh and the Gizmo will snap to it, but only on the grabbed axis.

4. Set up pivots with rotation for corner meshes:

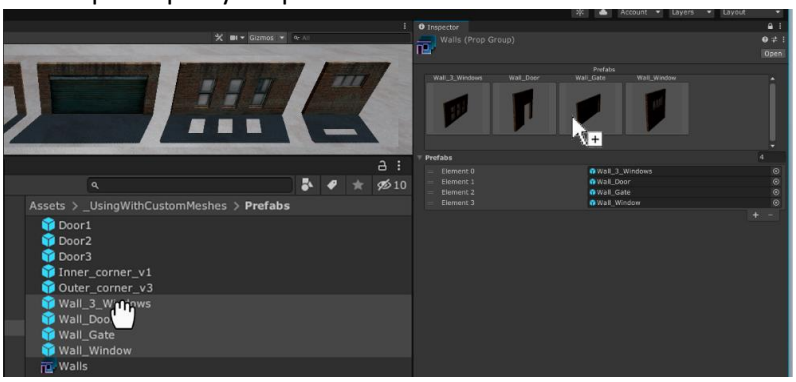


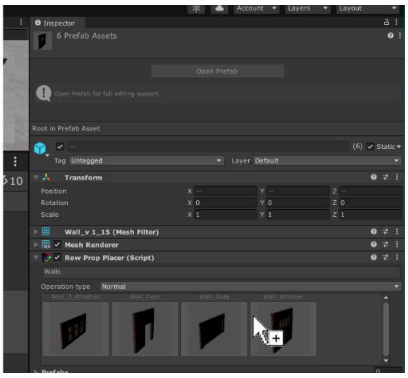
5. Repeat this steps for the other Prefabs.

Tip: if your mesh size is the same, you can simply copy paste components:

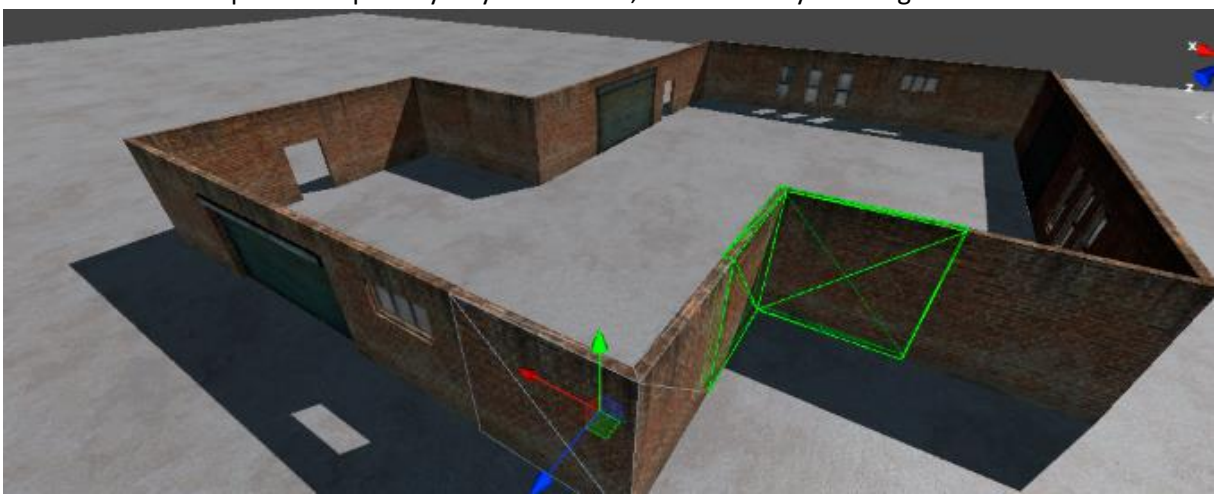


6. Create a PropGroup named Walls. Add your prefabs to it by drag and drop. Than drag and drop the PropGroup to your prefabs.

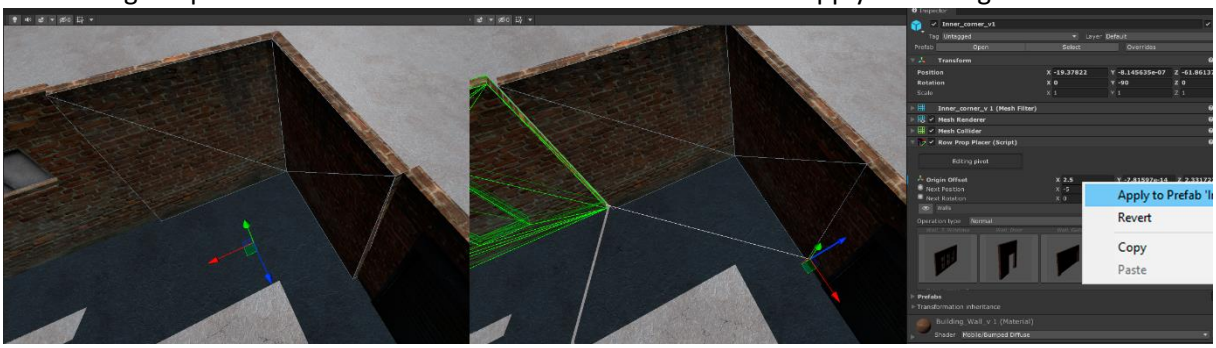




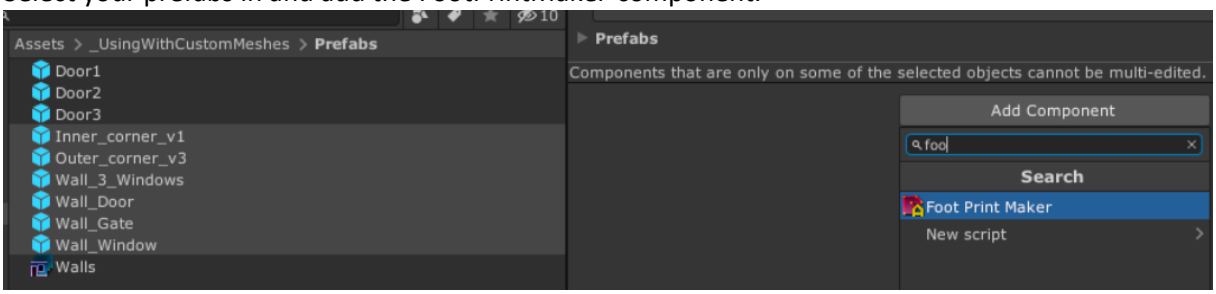
7. You added RowPropPlacer capebility to your Prefabs, now test it by creating some walls.



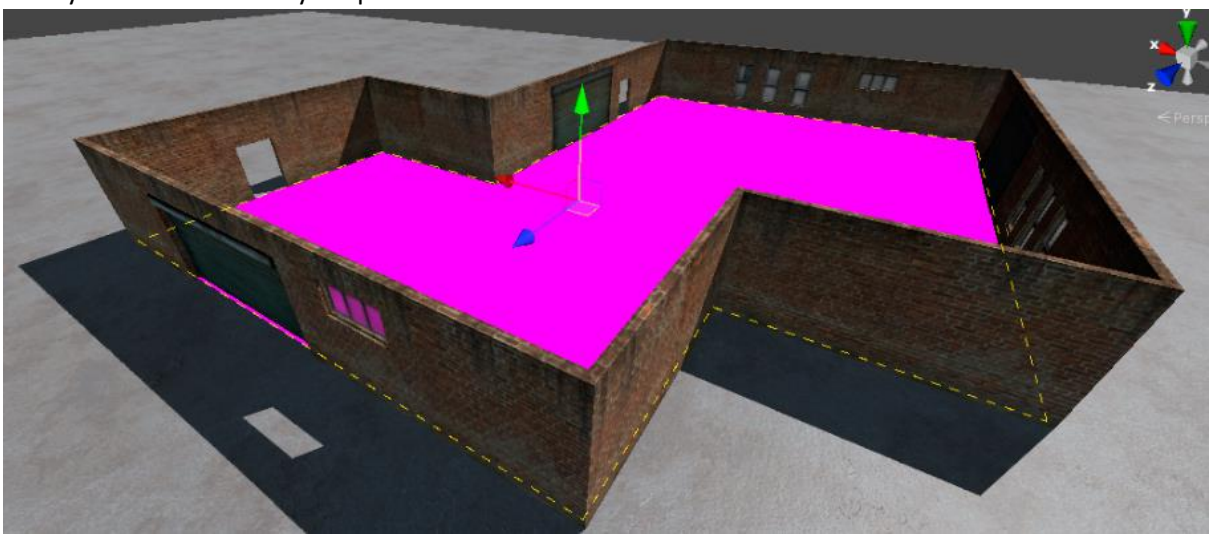
8. Fix missaligned pivots if needed. You can fix them in the scene and apply the changes.



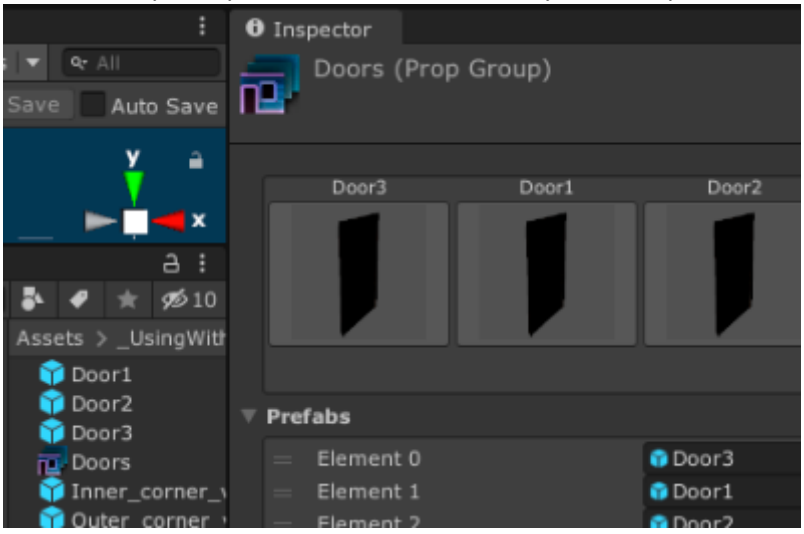
9. Select your prefabs in and add the FootPrintMaker-component.



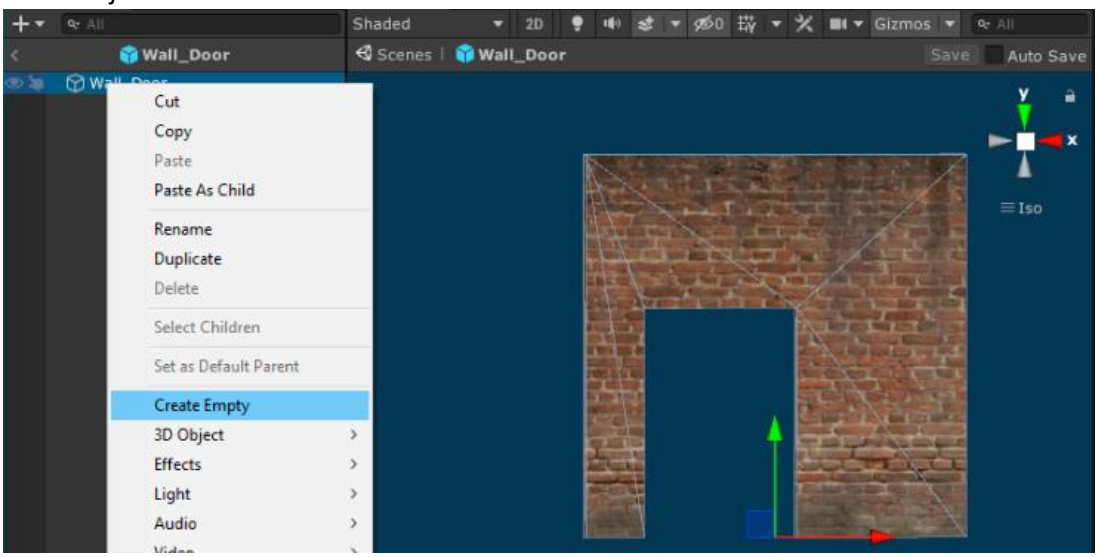
Now you can use it with your prefabs.



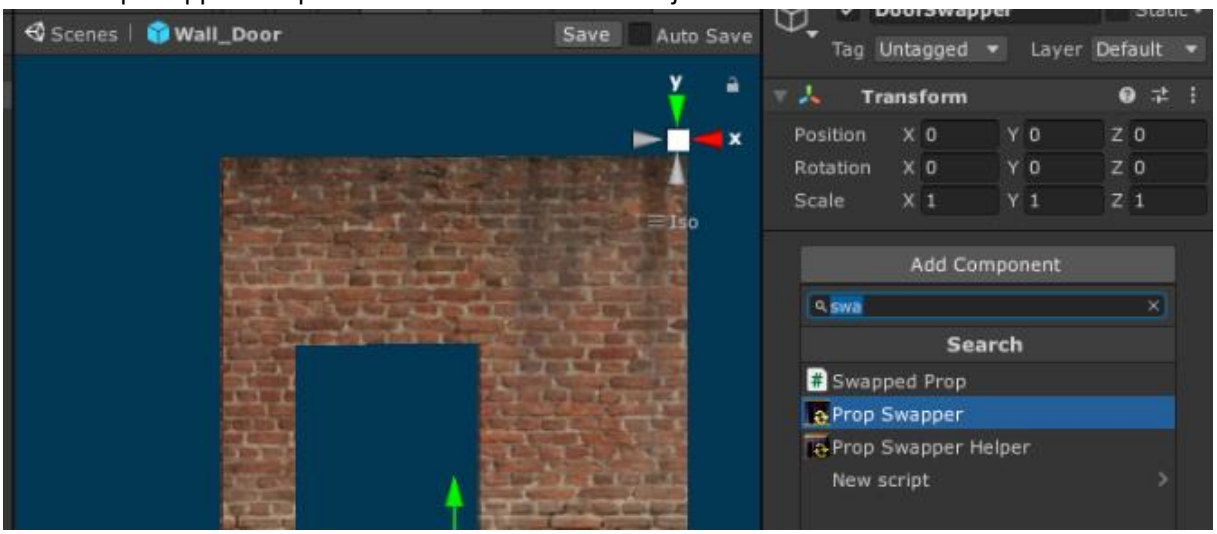
10. Create a PropGroup named “Doors” and add your door prfabs to it.



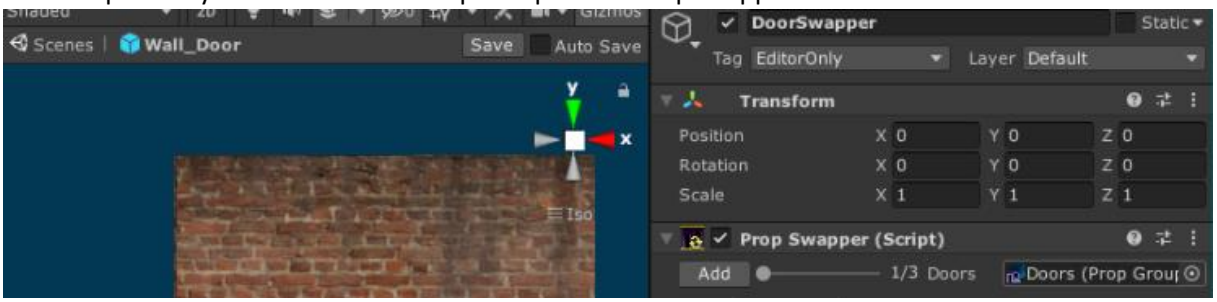
11. Open the prefab, with the door hole in the mesh, in the Prefab editing Mode and add a empty GameObject.



12. Add a PropSwapper component to the created GameObject.

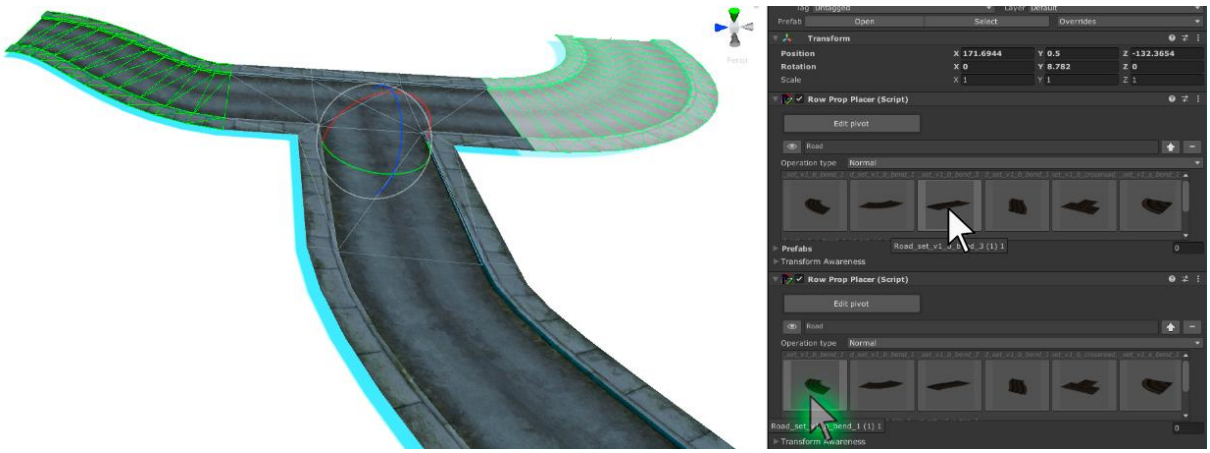


13. Add the previusly created “Doors” PropGroup to the PropSwapper.



14. If you need to move the PropSwapper Transform, you have to disable the Transform Awareness on the PropSwapper, because it only works if the local position is zero.
15. Now you can add doors with the PropSwapper. Add a PropSwapperHelper to the root Transform of the prefab to make it easier to use.

Tip: In some cases it is a good idea to add multiple RowPropPlacer components



That’s it, now you know how to upgrade your prefabs to use them with the BuildingMakerToolset.