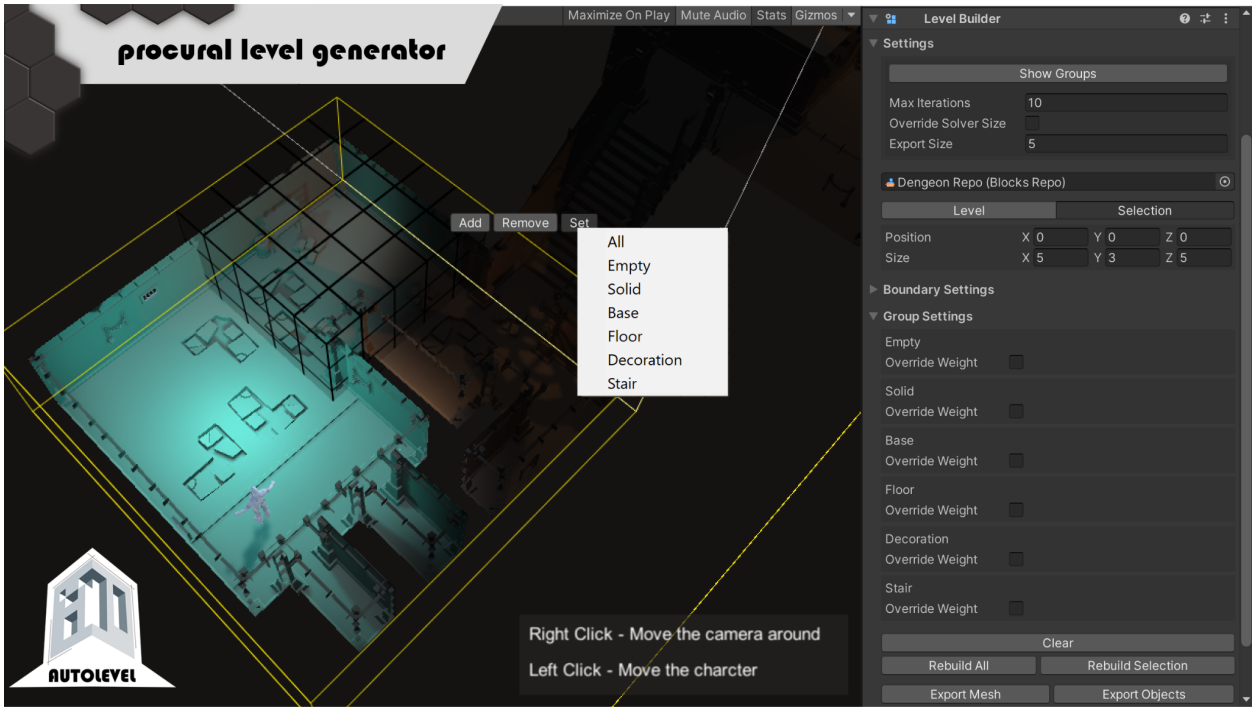
AutoLevel



Free procural level generator based on WFC algorithm for unity.

The target of this package is to create a procural level generator that is controllable, easy to use, and fast to some extent by leveraging the power of the WFC. WFC is a powerful algorithm in terms of procural generation. However, it does not give the user control over the generation process.

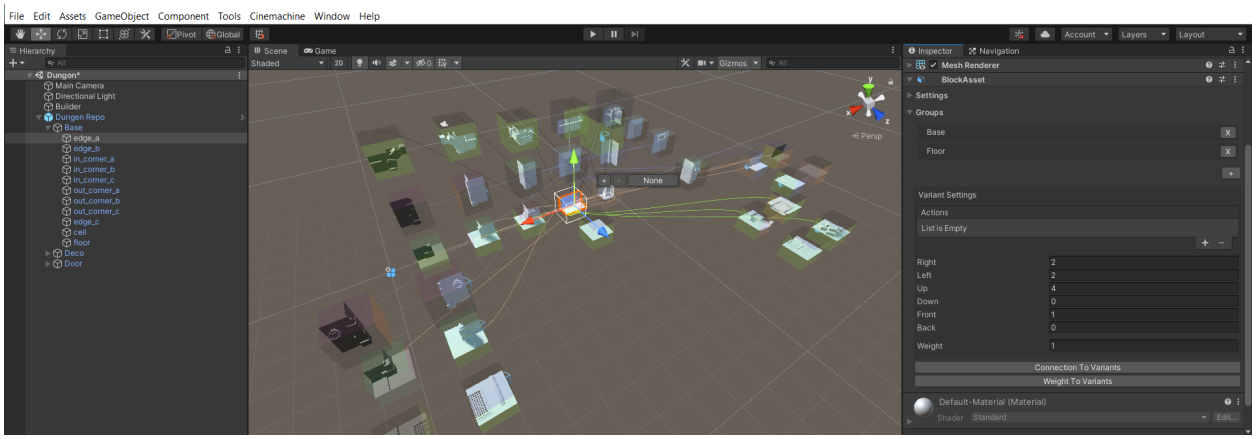
Features

- Runtime WFC solver
- Editor tools, build level inside the editor
- Set the weight for individual Blocks or Block Group
- Control the generation process, and constrain volume to groups of Blocks
- Constrain boundary to a single group or another builder boundary
- Ability to rebuild a given section of the level
- Block variants and Block group variants
- Big block support
- Mesh builder that supports multiple materials input, with the ability to subdivide the level into chunks
- fbx export
- Child objects export

Complete C# source code is provided.

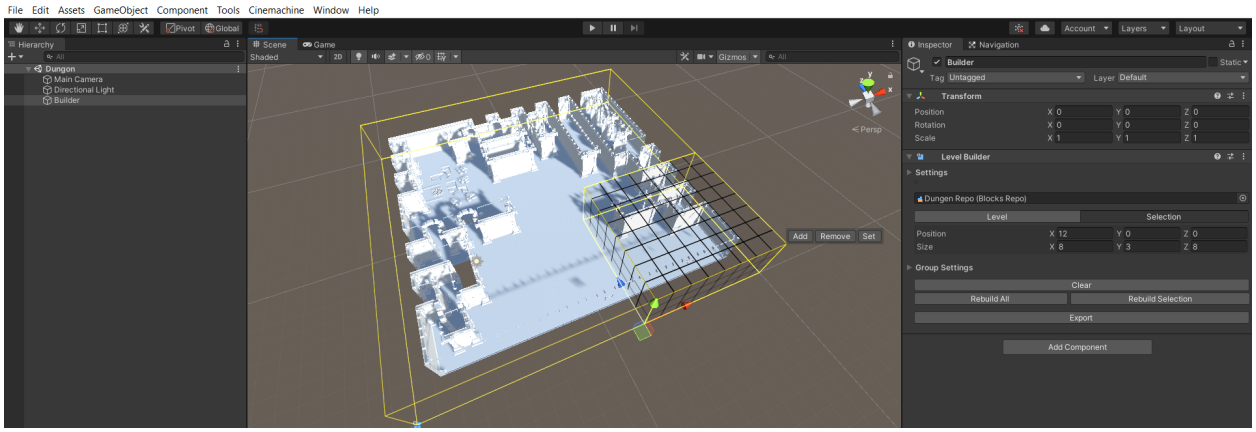
Usage

- Create the `Blocks Repo` by clicking 'GameObject/AutoLevel/Blocks Repo'
- Place Blocks under the `Blocks Repo` in the hierarchy. Make sure that the mesh is read/write enabled in the import settings, and the Blocks are in the range (0,0,0) (1,1,1), and their pivot is in the bottom back left corner.
- Add the `Block Asset` component to the Blocks
- Select a Block and start making connections in the scene view. Remember to change the editing mode to connection. After the connections are made, the `Blocks Repo` is ready to use



- Assign the `Blocks Repo` to the builder
- Use the selection handle to set block groups over the level. The level inspector provides a toggle to switch between controlling the level or selection bounds. Bounds can be controlled via inspector or by using scene handles. Toggle between the different handles using w,r,t
- Hit the rebuild button
- Hit `Export Mesh` to export the result to fbx file

Keep in mind that the Building performance depends on two factors the size of the builder and the number of blocks produced by the repo. You can get better performance by only building where needed using multiple builders. Also, using constraints can reduce the building time significantly.



Filling

When selecting a `Block Asset,` there is an option called filling in the scene view context menu dropdown. This will show handles to edit the block filling, red for empty and green for fill, similar to the marching cubes algorithm.

The filling has two roles. First, the connections will only be made to blocks with similar side patterns. Second, `Level Builder` can use that information to define the level rooms. The builder contains two built-in groups, the 'Empty' and 'Solid' groups, and you can use them to define the rooms and walls.

What's Next?

* Fine Block connection control
* Multithredead solving