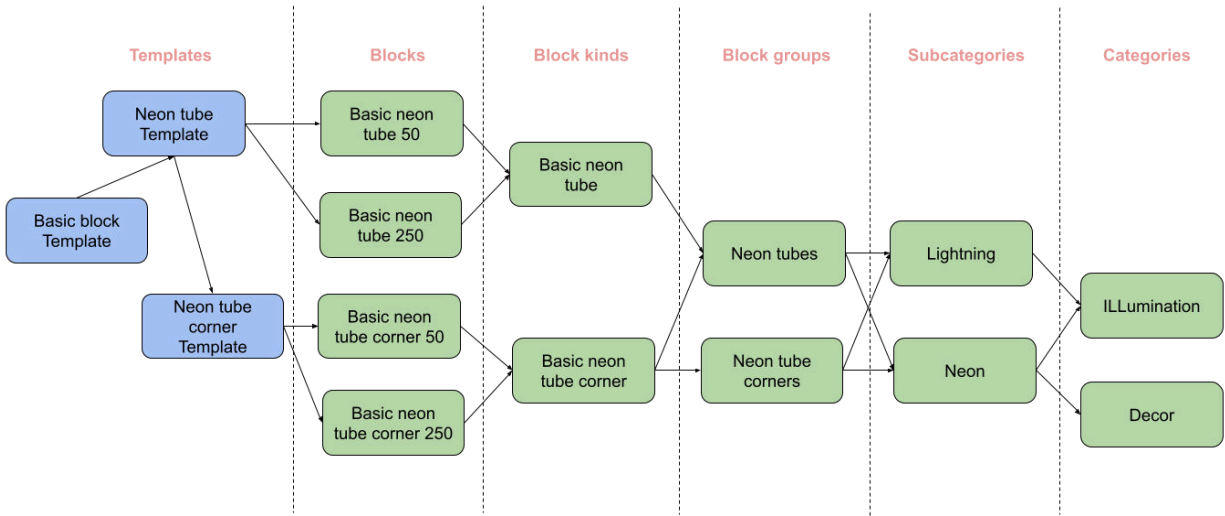


G-Screen

Data relations



Ordering rules

Categories [\(Definitions\)](#)

Categories are sorted by:

- 1) `CategoryDefinition.Order`, lower numbers go first. Order 0 is special, and will be treated as `int.MaxValue`.
- 2) `CategoryDefinition.Name` localization key.
- 3) [CategoryDefinition.Guid](#).

Subcategories [\(Definitions\)](#)

Subcategories are sorted by:

- 1) `SubcategoryDefinition.Order.`, lower numbers go first. Order 0 is special, and will be treated as `int.MaxValue`.
- 2) `SubcategoryDefinition.Name` localization key.
- 3) [SubcategoryDefinition.Guid](#).

Block groups [\(Definitions\)](#)

Block groups are sorted by:

- 1) `BlockGroupDefinition.Order`. lower numbers go first. Order 0 is special, and will be treated as `int.MaxValue`.
- 2) `BlockGroupDefinition.Name` localization key.
- 3) [BlockGroupDefinition.Guid](#).

Block Kinds [\(Definitions\)](#)

`BlockKindDefinition` represents blocks of the same type but different sizes. Block Kind can be assigned to block using `CubeBlockDefinition.BlockKind`.

Block groups can be configured in `BlockKindDefinition.BlockGroups`. Block kinds will be inserted to `BlockGroupDefinition.BlockKinds` automatically during the post-processing step.

Block kinds in Block Groups are sorted by:

- 1) `BlockKindDefinition.OrderingPriority`, lower numbers go first. Ordering priority 0 is special, and will be treated as `int.MaxValue`.
- 2) `BlockKindDefinition.Name` localization key.
- 3) [BlockKindDefinition.Guid](#).

Manual block kinds ordering

If really needed, you can manually set order of block kinds in the block group using `BlockGroupDefinition.BlockKinds`. But consider that this will also affect the final order of block kinds for which groups have been configured using `BlockKindDefinition.BlockGroups` since the insertion procedure is repeated for each block kind that will be inserted automatically to the group.

Automatic insertion rules

When manual order is specified for some block kinds in `BlockGroupDefinition.BlockKinds`, all other block will be inserted each separately using the following procedure in the previously specified sorting order:

- 1) Find the last block in `BlockGroupDefinition.BlockKinds` that has higher or the same ordering priority
- 2) Insert the block after it.
- 3) Take the next block that should be inserted
- 4) Return to step 1.

Block sizes (Definitions)

BlockSizeDefinition represents relative block size that is used to indicate block size on the size indicator. You can configure the icon, its name and BlockSizeDefinition.RelativeBlockSize that's a unique value used to compare the size with other block sizes.

Assignment block sizes to blocks

BlockSizeDefinitions are assigned to blocks of BlockKindDefinition by the real physical size CubeBlockDefinition.ExactBlockSize automatically. If we have less blocks in the Block Kind than BlockSizeDefinitions in the game, all sizes are simply assigned linearly, starting from the smallest block. Blocks with the same physical sizes get the same BlockSizeDefinition. In all other cases blocks with highest CubeBlockDefinition.ExactBlockSize get BlockSizeDefinition with highest BlockSizeDefinition.RelativeBlockSize value. Similarly with the smallest size blocks. Other block sizes are interpolated linearly and assigned for other blocks. This means that block sizes will be evenly distributed between blocks in the block kind based on block positions and corresponding block size positions.

Blocks sorting

Blocks are sorted inside BlockKindDefinition according to CubeBlockDefinition.ExactBlockSize in ascending order during the post-processing step.

Size indicator (UI)

Size indicator displays all blocks of the BlockKindDefinition as a list of corresponding block size icons. There are as many icons as blocks in the Block Kind. If there are more blocks than BlockSizeDefinitions in the game, there will simply be duplicate icons (Typically used only for Mod conflicts resolution). They serve as buttons to switch between blocks.