

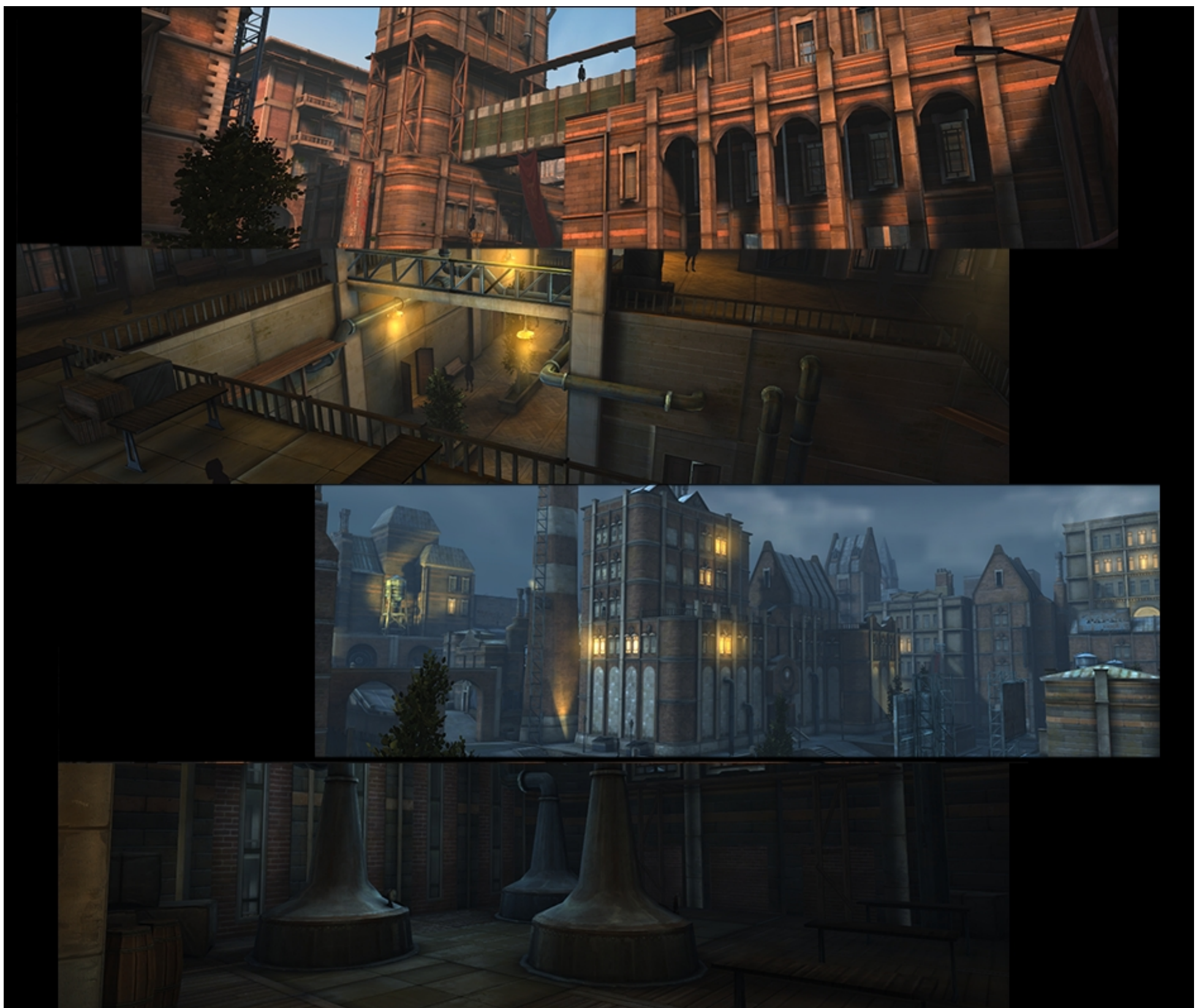
Quick Introduction

This is an alternate universe 1800s and steampunk styled industrial environment, containing two demo scenes, modular kits used to build them as well as preassembled buildings and structures.

High level of modularity is main strong side of this asset - if you are not going to be taking advantage of it and are looking for mostly standalone buildings that you are going to use as is without modifications, this package might not be for you.

Scenes

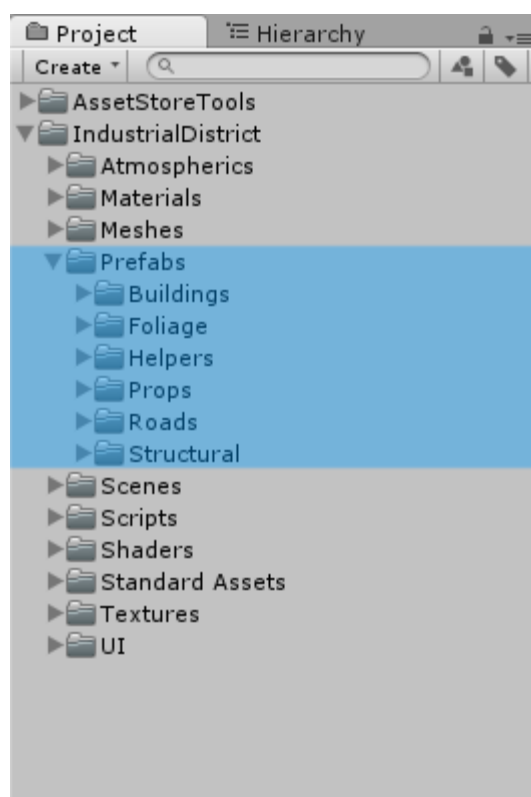
/Scenes/sc_A_Night.unity/ and */Scenes/sc_B_Dusk.unity/* contain two demo environments with different layouts and lighting. Be sure to check these first - they can provide quick insight about what to expect from this asset as a whole as well as context in which to use each individual prop or modular piece.



Folder structure

Folder structure for this package is pretty straightforward. *Materials*, *Textures*, *Shaders*, *Scripts*, *Prefabs*, Various *Meshes* and *Atmospherics* all go into their respective folders. Unless you want to tweak materials or meshes directly, you are looking for */Prefabs/* folder. It contains prefabs for every asset already set up with materials, collisions and layers.

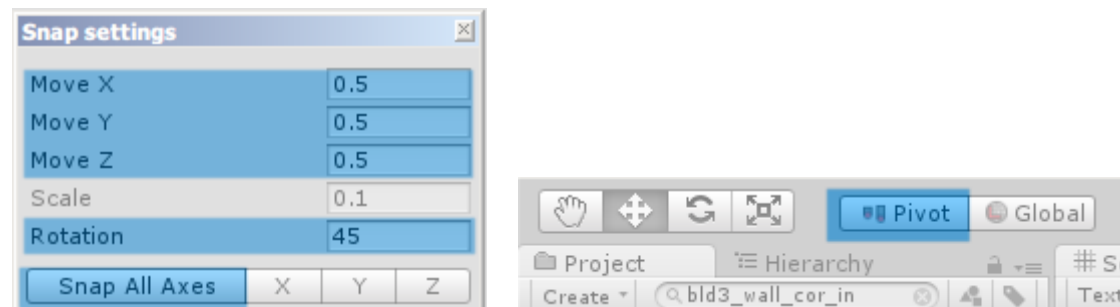
- ⤴ **Buildings** - containing modular pieces for buildings, preassembled structures as well as some towers and city walls
- ⤴ **Foliage** - containing trees and bushes
- ⤴ **Helpers** - containing various helper prefabs, such as human figure (for scale comparison) as well as some dummy geometry, rough blackout meshes and lighting rigs.
- ⤴ **Props** - containing various generic props, street props and some industrial assets such as distillery pots, turbines, gears etc.
- ⤴ **Roads** - containing modules for assembly of roads, bridges and pavements
- ⤴ **Structural** - various support structures for buildings as well as piping, reservoirs, platforms and various industrial pieces



.prefabs folder

Grid and Snap Settings

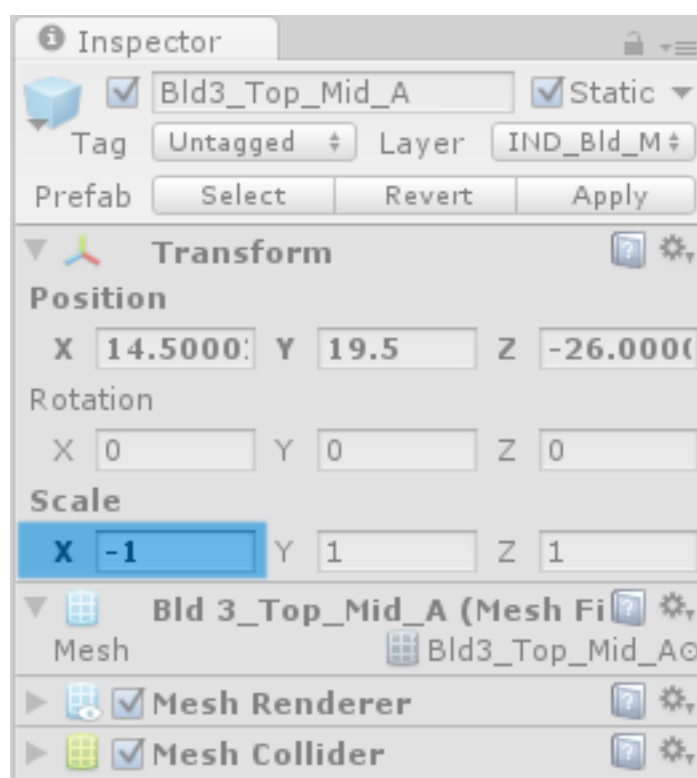
Usage of simple props is pretty straightforward - just drag and drop appropriate prefabs into your scene. Building, road and platform modules require simple grid setup though. It is accessed from Edit->Snap Settings. Use value of 0.5 units (50 cm) for move and values such as 5, 15, 45, 90 for rotation.



.snap and pivot settings

After placing object in scene hit "Snap All Axes" to align it to grid and use Ctrl+Drag to move it using specified increments. Make sure handle is placed at pivot point instead of object center. Keep holding Ctrl to rotate object using specified degree increments. Ctrl+D duplicates objects.

Mirroring objects is achieved by using negative scale. For example, if you want to mirror object on X axis, write "-1" as its scale value.



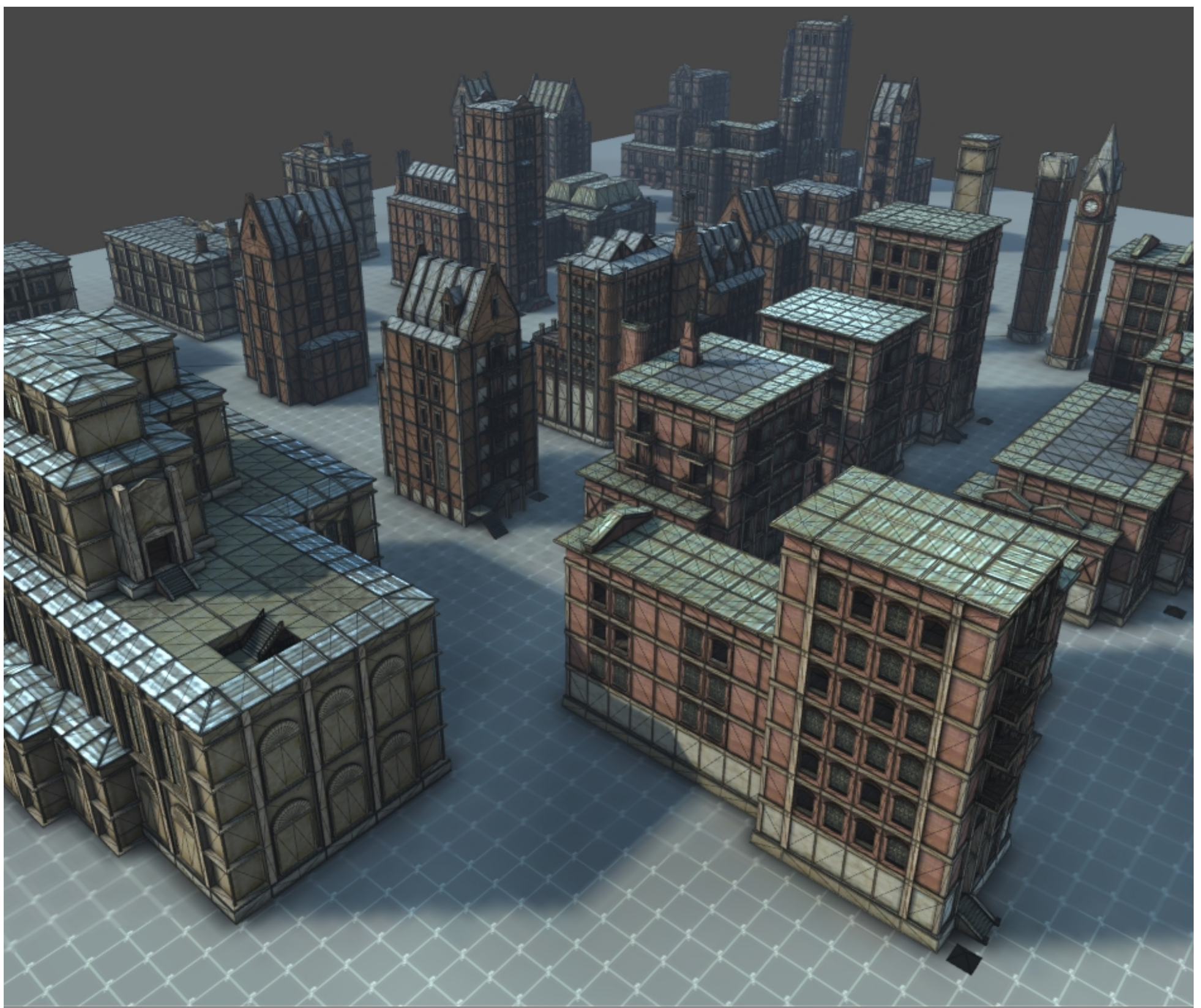
.mirroring geometry

Preassembled buildings

If you want to get started as soon as possible */Prefabs/Buildings/PreAssembled/* folder contains already built structures with basic interiors that you can simply drag and drop in your scene. You can then proceed to modify them as you see fit (add floors, remove interiors, change footprints, populate them with props etc etc)

/Prefabs/Buildings/PreAssembled/Background/ subfolder contains buildings that are used to fill backgrounds or areas that player can't reach. These have less details, are merged into single meshes and have no interiors nor collisions. They are also taller to make backgrounds more interesting compositionally.

For quick overview of aforementioned buildings, head to */Scenes/PreAssembled.unity/*. It is a scene that houses all prefabbed structures and provides nice overview + FPS controller for inspecting them better, checking interiors etc.

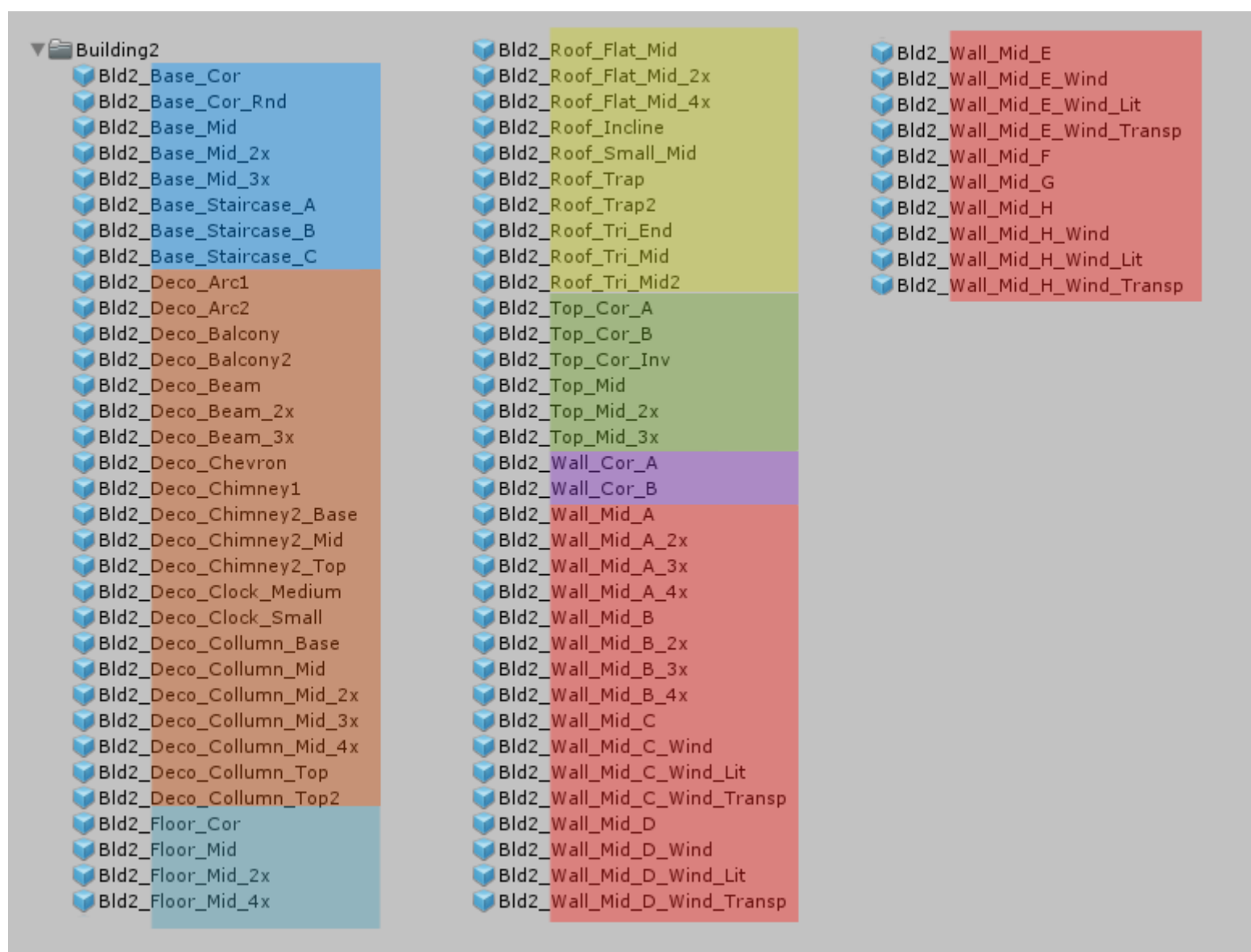


.scene with preassembled buildings

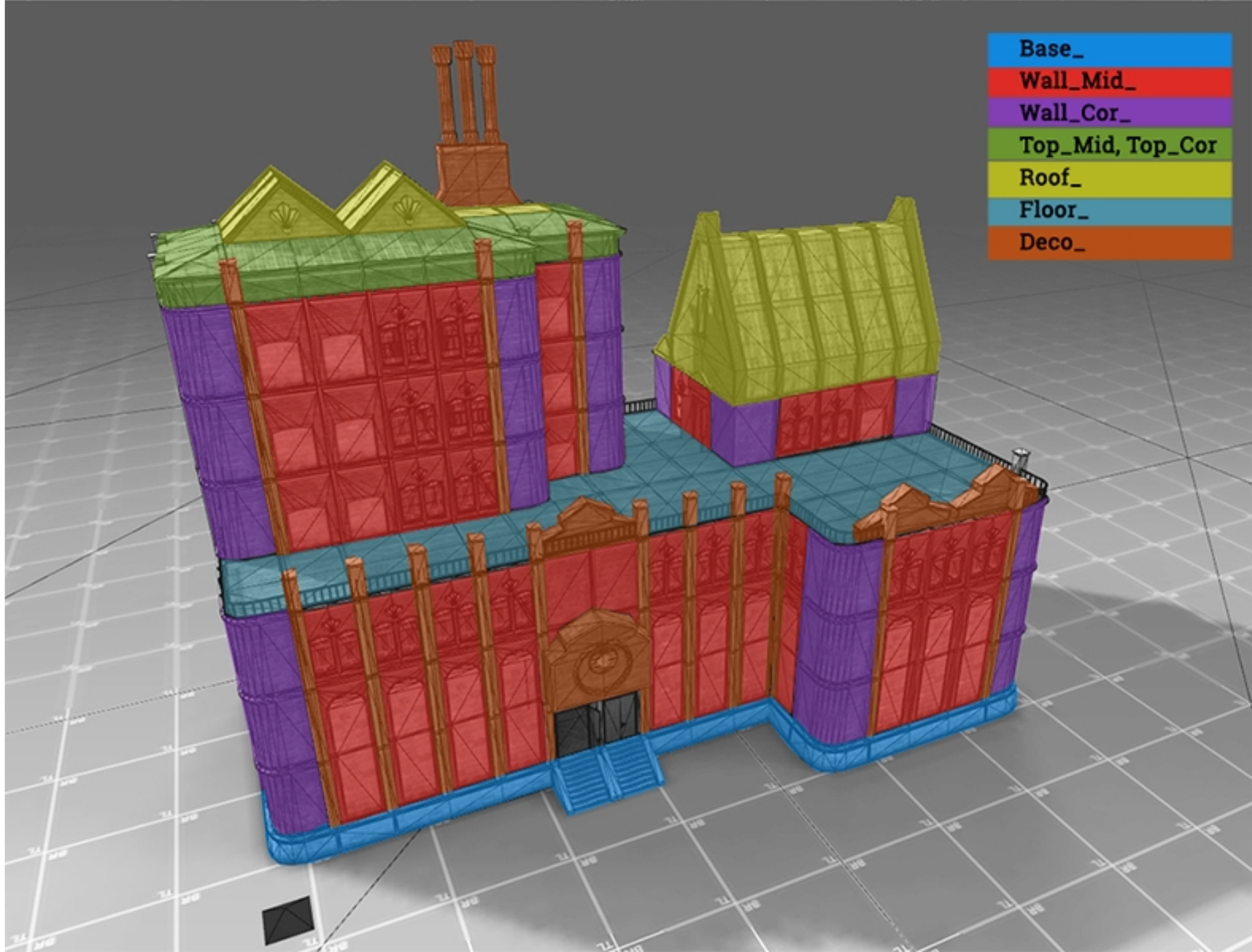
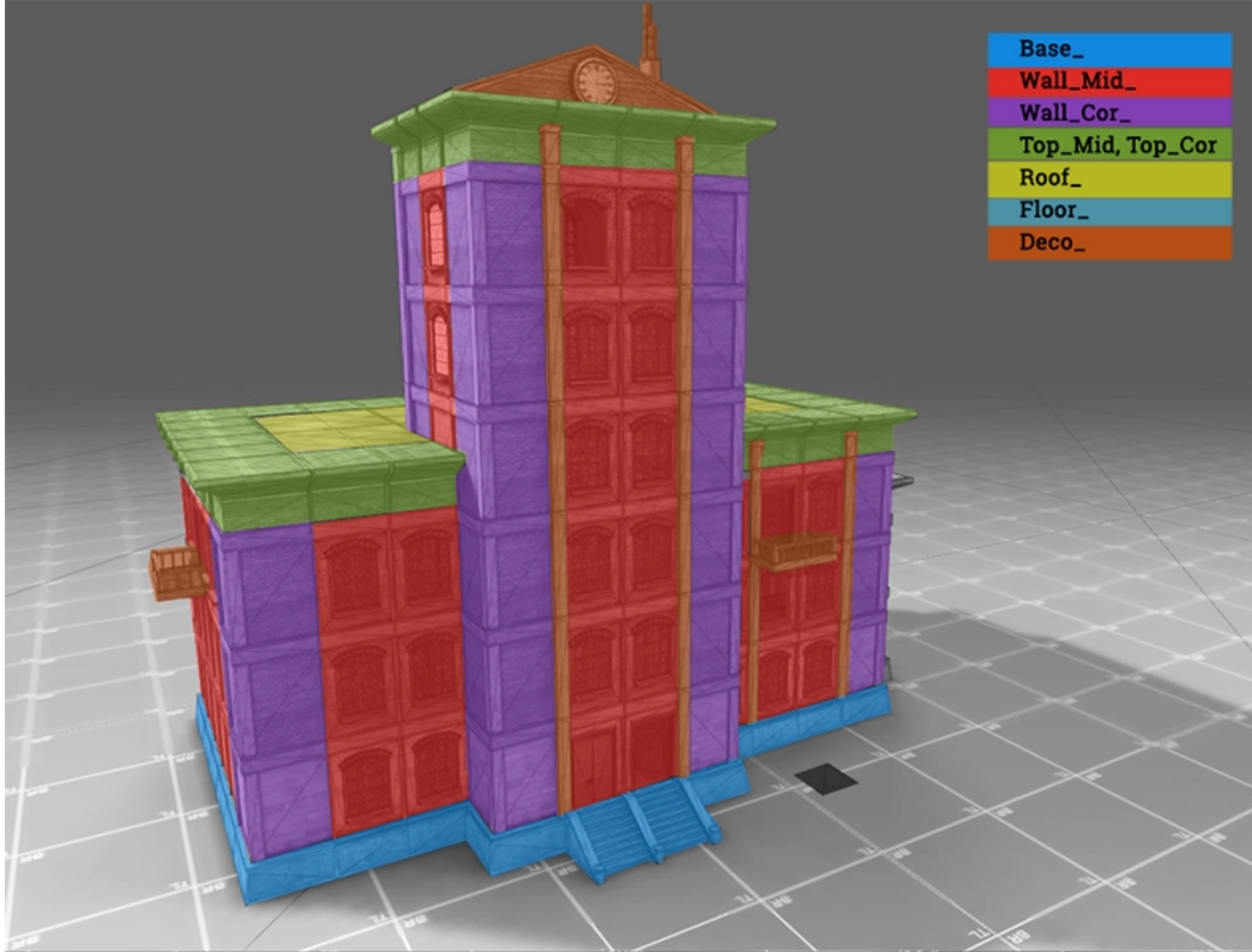
Modular Assets: Buildings

/Prefabs/Buildings/Building(1-4)/ contains four distinct modular kits that can be assembled into many different buildings and layouts. In order to help identify function and relative placement of each module, they are broken down into several categories and are separated by prefixes. These are:

- ^ **Base_** - these form basement of buildings. copied horizontally.
 - ^ **Wall_Mid_** - walls of buildings. these include plain walls as well as walls with holes for windows/doors. Duplicate these horizontally or vertically to form width/height of main building walls
 - ^ **Wall_Cor_** - corners of buildings. stacked vertically to form height of building. If corner you'd like to form is not available you can use simple combination of two wall modules and a column module in-between them to cover up intersection.
 - ^ **Top_Mid_**, **Top_Cor_** - these modules form topmost part of walls and are used to transition from wall to roof modules. Duplicated horizontally.
 - ^ **Roof_** - various roof modules, including flat, triangle and trapezoid shape roofs.
 - ^ **Floor_** - floor modules. primarily used in interiors but can be used with open terraces as well. duplicated horizontally
 - ^ **Deco_** - set of decorative modules, including columns, balconies, ceiling support beams, chimneys etc. These are important since they add visual interest to buildings, help break up boring areas or even mask harsh transitions or lightmap seams.
- ^ Some modules also have **_2x**, **_3x**, **_4x** *suffixes*. These are basically several copies of same module attached together to reduce scene object count and make assembly process a bit faster.
- ^ */Prefabs/Buildings/Shared/* contains Shared items that are supposed to be reused across buildings. Standardized items, such as doors, staircases and their railings also fall into this category and are identified with **Gen_** prefixes.



05. overview of folder containing modular kit



Modular Assets: Towers and City Walls

/Prefabs/Buildings/Towers/ contains some tower modules as well as large walls and arcs to form city walls. These modules are:

- ⤴ **Arc_** - large arcs. these can be copied around to form arcades or used between walls to create city gates for example. Generally duplicated horizontally, but can be stacked on vertical axis as well to form Aqueduct-like structures.
- ⤴ **Gate_** - city gates. Used in conjunction with arcs.
- ⤴ **Tower_Base_** - basements for towers
- ⤴ **Tower_Mid_** - mid parts of towers. stacked vertically to create height of towers
- ⤴ **Tower_Top_** - topmost part of towers, be it simple tower roof or even clockwork.
- ⤴ **Wall_** - massive wall modules used to form external city walls. Use towers to redirect flow of walls.

Modular Assets: Roads

/Prefabs/Roads/ folder contains modular kit used to create roads, pavements and various street-related architectural structures.

- ⤴ **Road_** - actual road modules. Includes straight modules, corners, 3-Way modules as well as crossroads, inclines and dead ends. stacked horizontally
- ⤴ **Bridge_** - bridge modules. These are supposed to be used as continuation of roads and have roughly same footprints
- ⤴ **Deco_** - fences, pillars, fence bases and other decorative geometry
- ⤴ **Pavement_** - flat pavements with different patterns.
- ⤴ **Pod_** - modules used to form podium-like elevated structures. Good for making platforms for buildings, plaza-like areas etc to introduce variety in terrain and break line-of-sight of players. Stacked both horizontally and vertically.
- ⤴ **TramRails_** - railings. Place these on top of road modules to form train tracks.

Misc: Structural

/Prefabs/Structural/ folder consists of various useful visual and gameplay support structures:

- ⤴ **Pipe_Basic_**, **Pipe_Thick_** - pipe modules, thin and thick. Thicker pipes are mostly meant to be used in middle and backgrounds to add visual interest and 'industrialness' to buildings.
- ⤴ **Platform_**, **WoodPlatform_** - various wooden and metal platforms. They can be used as simple platforms for players to jump on, or assembled into more complex structures such as makeshift bridges between buildings, as means for players to scale large walls/structures vertically and so on.
- ⤴ **Lattice_** - metal supporting structure, used independently or as means to 'ground' other structures better
- ⤴ **Elevator_**, **Reservoir_** - Industrial style elevators similar to ones used on construction sites as well as some fluid reservoirs.

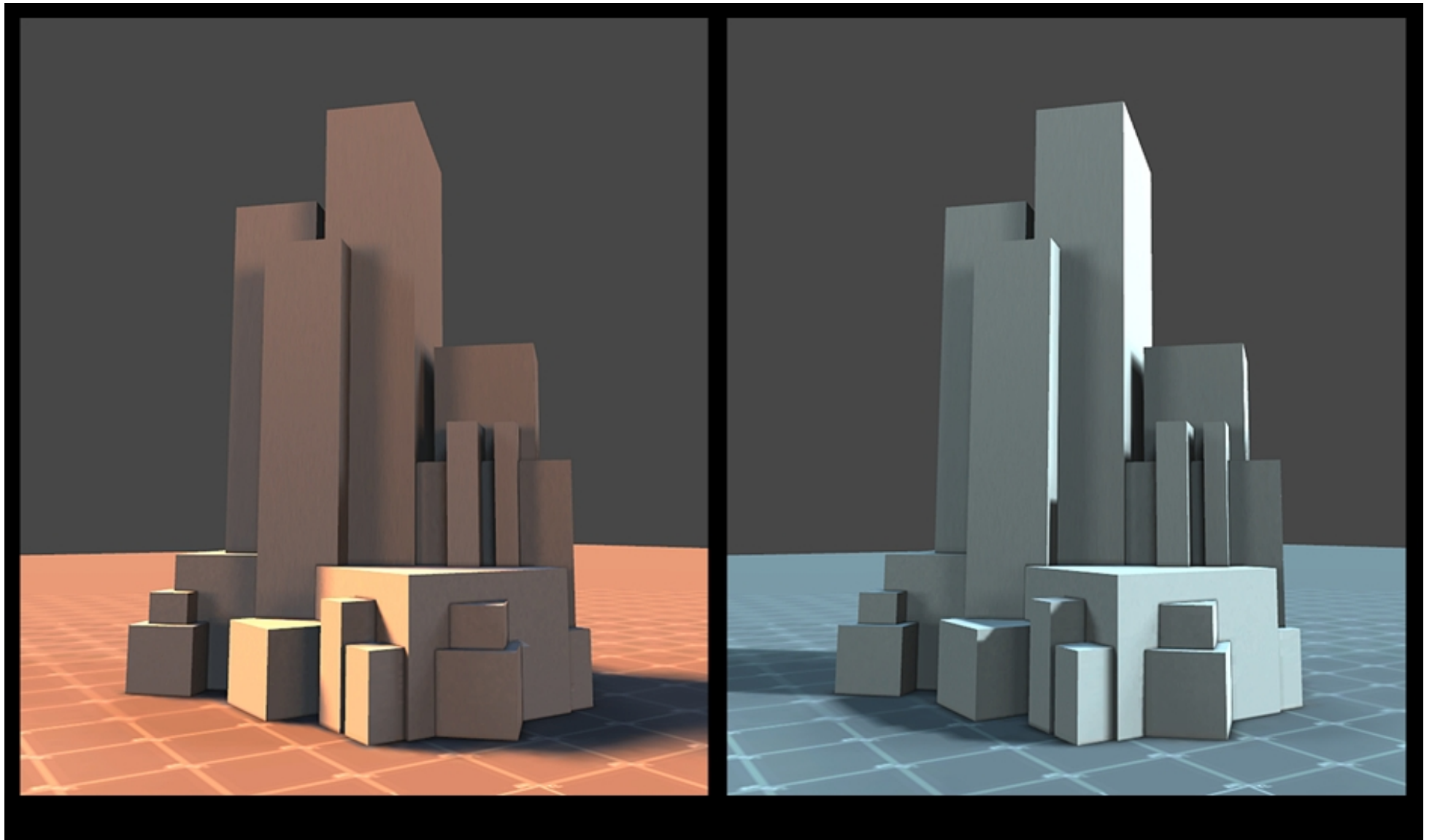
Misc: Props

/Prefabs/Props contains rest of props which have no usage rules whatsoever. These are *park benches, street lamps, fire hydrants, posters, crates, barrels, pallets*, various industrial props such as *turbines, gears, iron beams* etc.

Lighting

If you want to quickly get lighting setup similar to ones used in demo scenes check */Prefabs/Helpers/LighRig_Dusk* and *LightRig_Night*. These lighting prefabs consist of bunch of low-intensity lights to provide soft skylight approximation and one keylight for sun/moonlight. Simply drag& drop one of them in your scene and do a quick bake.

Since Unity Pro has built-in skylight and global illumination features, these rigs are only going to be relevant for Indie users.

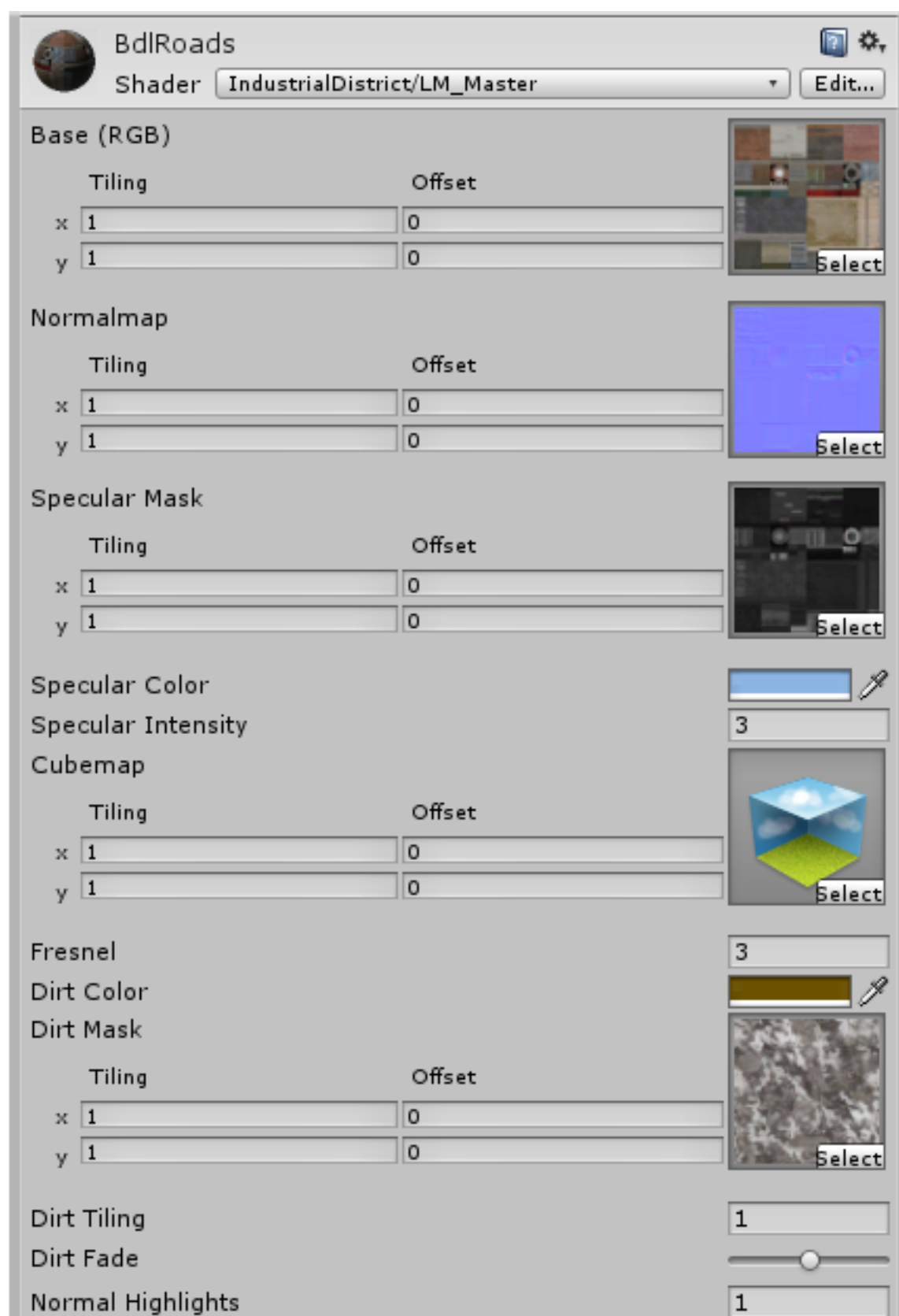


. prefab lighting rigs

Material/Shader

Majority of objects use `/Shaders/LM_Master.shader`. This is a simple surface (lambert) shader with couple of enhancements. Settings are as follows:

- ^ **Base, NormalMap, Specular Mask** - slots for appropriate diffuse, normal and specular maps.
- ^ **Specular Color, Specular Intensity** - color and intensity modifiers for specular. Please note that shader does not do actual specular calculations and relies on reflection cubemap for highlights.
- ^ **Cubemap** - cubemap for highlights. This can be actual blurred cubemap sampled from scene or even simple high-contrast image imported as cubemap.
- ^ **Fresnel** - fresnel multiplier for specular. higher values push specular towards edges.
- ^ **Dirt Color** - color for dirt layer. By carefully controlling color and intensity variety of effects can be achieved.
- ^ **Dirt Mask, Dirt Fade, Dirt Tiling** - input for black and white mask for dirt as well as intensity and tiling controls for it
- ^ **Normal Highlights** - This is a simple shader trick to get some of the normal detail back after lightmap bakes. If your project uses dynamic lighting or dual/directional lightmaps this is not necessary and can be set to 0.



Performance

Since package was designed with flexibility in mind, modules are relatively small. This means large levels are going to have high amount of objects in it. Depending your performance goals, you might want to address this situation by enabling occlusion culling, static batching or combination of both.

For Unity Indie users, mix of techniques such as hiding/unhiding interior objects dynamically, controlling camera clipping plane and disabling mesh renderers of distant props could be a viable alternative. Merging close objects together with a mesh combiner script of some sorts could also be helpful in keeping drawcalls under control.