Lowpoly Mesh Generator

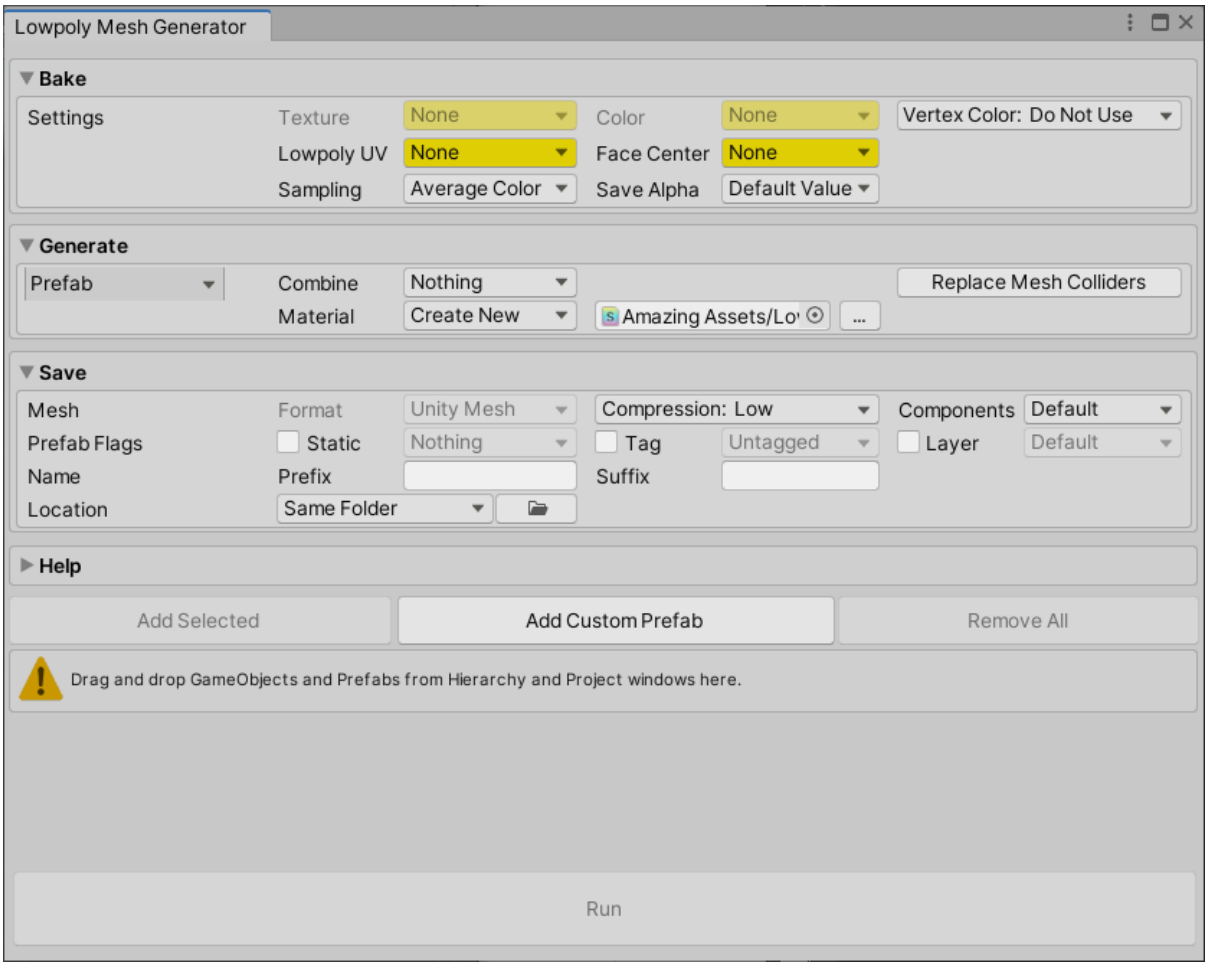
amazingassets.world

TABLE OF CONTENTS

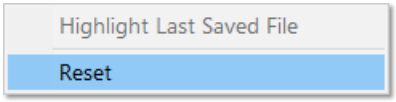
QUICK START

Open **Quick Start** scene from **Lowpoly Mesh Generator \ Example Scenes** folder. Scene contains simple ***House*** object.

Open Lowpoly Mesh Generator (LMG) editor window **Unity Main Menu \ Windows \ Amazing Assets \ Lowpoly Mesh Generator**.

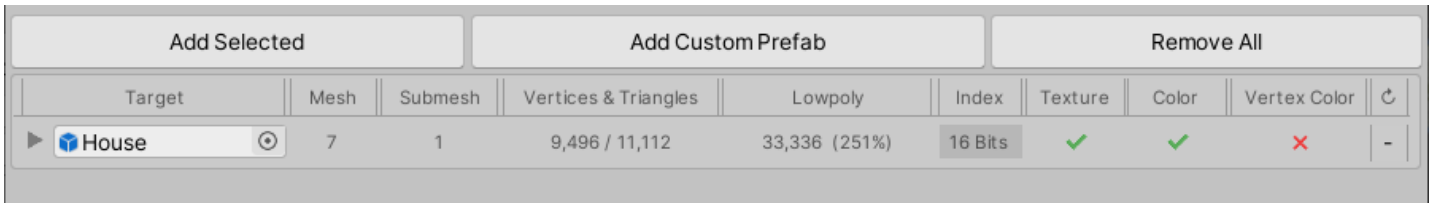


Use context menu to reset window and load default settings.

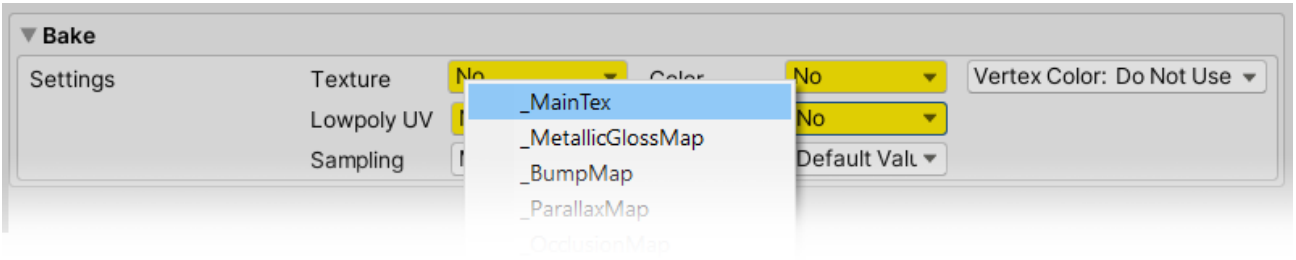


Add ***House*** object to the LMG editor window by drag & drop it from the Hierarchy window or by clicking on the **Add Selected** button (if object is selected).

LMG window will display ***House*** object and its mesh/material info.



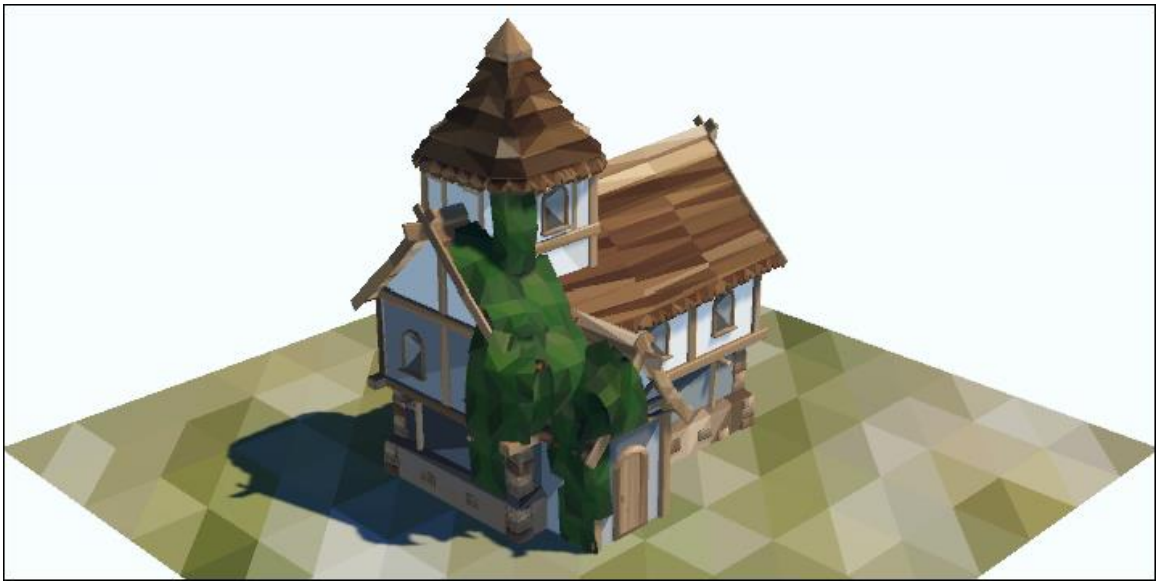
Inside **Bake** group choose texture and color properties that will be baked inside mesh vertex color.
List displays all available texture and color properties currently used by mesh materials.



For this **Quick Start** tutorial use:

- **_MainTex** for Built-in render pipeline.
- **_BaseMap** for Universal RP.
- **_BaseColorMap** for HDRP.

Click on the **Run** button.
LMG will convert ***House*** object into a lowpoly (flat-shaded) style mesh, create material for it using vertex color shader and instantiate ready to use prefab in the scene in the same position as the source object.



To see generated mesh, move it inside Scene view or hide source object

Generated prefab is the exact copy of the source ***House*** object but with lowpoly meshes replaced in MeshFilters.

From the context menu choose **Highlight Last Saved File** to navigate Project window to the directory where LMG has saved generated files.

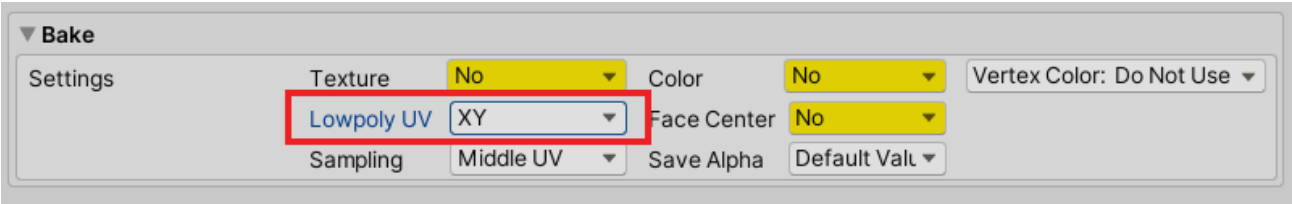
Generated lowpoly mesh has the same triangle count as the source mesh but vertex count always equals to 3 * triangle count. As triangle consists of 3 vertices and each one needs to be unique.

LMG editor window displays vertex & triangle count for original mesh and vertices count for lowpoly meshes.

Target	Mesh	Submesh	Vertices & Triangles	Lowpoly	Index	Texture	Color	Vertex Color
House	7	1	9,496 / 11,112	33,336 (251%)	16 Bits	✓	✓	✗
b		1	3,123 / 4,852	14,556 (366%)	16 Bits	✓	✓	✗
g		1	88 / 66	198 (125%)	16 Bits	✓	✓	✗
p		1	1,112 / 1,344	4,032 (262%)	16 Bits	✓	✓	✗
r		1	2,219 / 2,280	6,840 (208%)	16 Bits	✓	✓	✗
w		1	63 / 37	111 (76%)	16 Bits	✓	✓	✗
w		1	2,770 / 2,333	6,999 (152%)	16 Bits	✓	✓	✗
C		1	121 / 200	600 (395%)	16 Bits	✓	✓	✗

All lowpoly style and flat-shaded meshes created manually or generated using scripts use this vertex/triangle ratio.

Now, reset LMG editor window settings and instead of baking **Texture** and **Color** properties inside mesh vertex color, choose baking of **Lowpoly UV -> XY**.



Click on the **Run** button.

This will generate similar flat-shaded mesh, but now it will have new UVs that allow mesh to automatically render any material in lowpoly style.

Note, generated mesh now uses material from the source ***House*** object, but it is rendered in lowpoly style. Additional vertex color material is not required.
Changing diffuse texture inside material will have immediate effect.

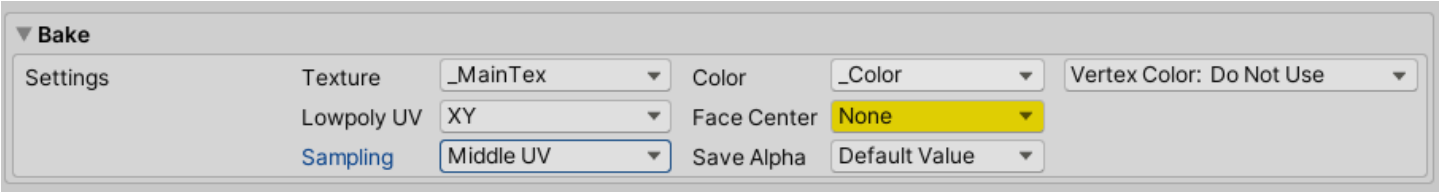
Baking **Texture/Color** or **Lowpoly UV** has its usage cases and advantages.

- \+ After baking **Texture/Color** inside vertex color there is no more need in diffuse texture and it can be removed from a project and build.
  \+ Multiple meshes can be combined into one mesh and be rendered in one draw-call.
  \- If two identical meshes use different textures for rendering, it is necessary to generate two meshes.

- \+ Baking **Lowpoly UV** is useful for dynamically changing meshes to avoid constant texture baking as they will automatically render any material in lowpoly style.
  \+ No need to generate multiple lowpoly mesh variations in the cases when they use different diffuse textures.
  -/+ Even diffuse texture is still in use by materials, there is no need to have them in high resolution as the same lowpoly rendering effect can be achieved with 512 or 128 pixel size texture.
  \- Similar meshes cannot be combined into one mesh, if they are using different materials or diffuse textures.

EDITOR WINDOW

BAKE



- **Texture & Color** – Texture and color properties that LMG editor tool will bake inside vertex color. Those properties are read from material used by a mesh.
  If property with such name is not available inside mesh used material, LMG editor will display it with ❌ icon and white color will be baked instead.

> Note, Run-time API allows baking any textures inside mesh vertex color without using material. Explained in chapter Run-time API.

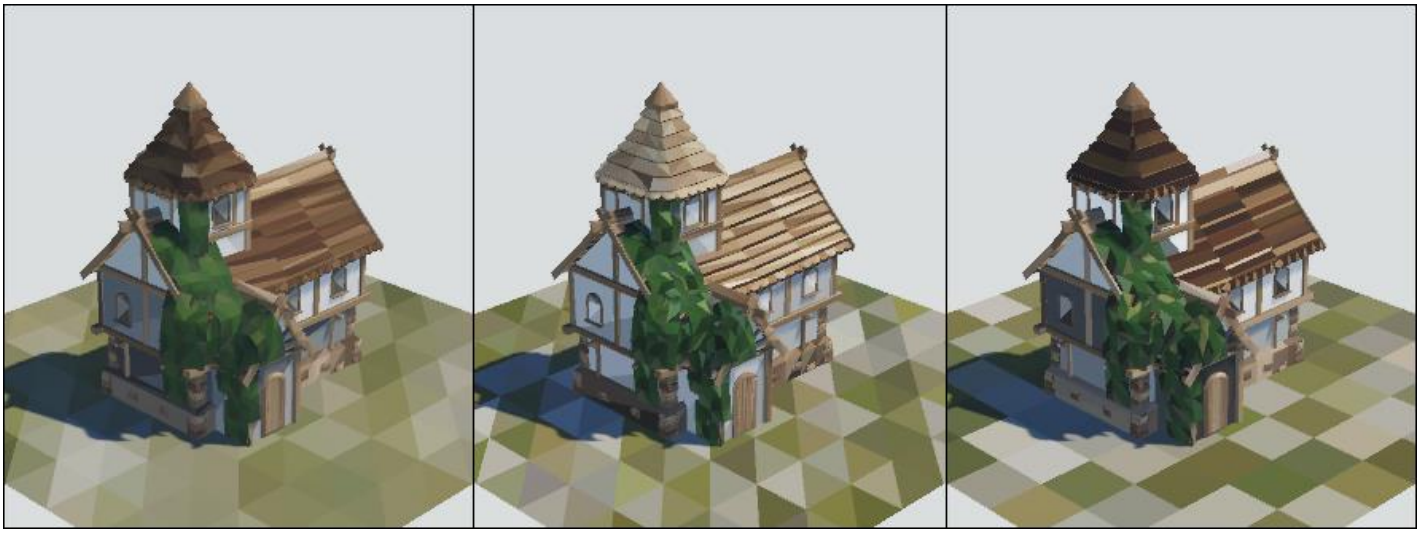
- **Vertex Color** – If source mesh already has vertex color, LMG can combine it with baked texture and color.

- **Lowpoly UV** – Generates new lowpoly UV0 and bakes it inside XY channel (overwrite original UV.xy) or ZW channel (original UV.xy is not changed).

- **Face Center** – Generates triangle face center and saves inside one of the UV channels. Can be useful for custom shaders.

- **Save Alpha** – Allows to choose alpha value for baked vertex color.
  Default means: texture's Alpha * color's Alpha * original vertex color's Alpha.

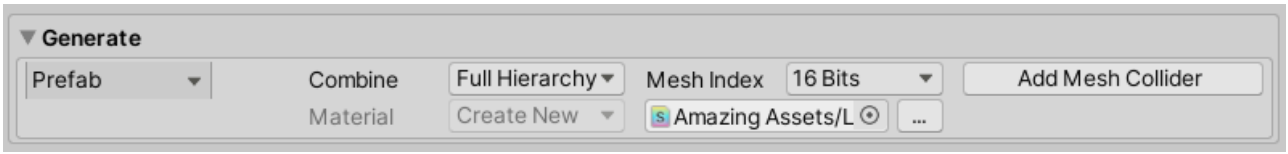
- **Sampling** – Style of the baked vertex color:
  - Average Color – LMG reads colors at each 3 vertices of a triangle and bakes average value.
  - Middle UV – Color is read from triangle's center.
  - First Index – Color of the first vertex of a triangle is used as bake color.



Average Color | Middle UV | First Index

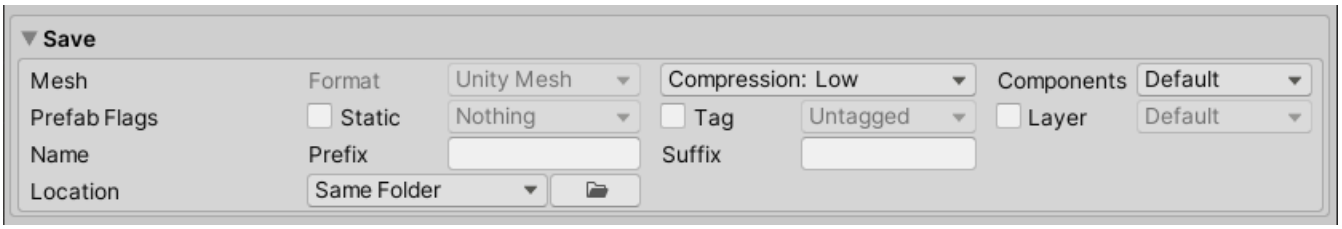
Note, generated **Lowpoly UV** can use only **Middle UV** or **First Index** types.

GENERATE



- **Generate** – Generates full prefab file that is an exact copy of the source object but with generated lowpoly mesh and vertex color material. Or just mesh asset file.
- **Combine** - LMG can combine generated mesh submeshes or entire object's hierarchy into one mesh file.
- **Mesh Index** – Combined mesh index format.
- **Add/Replace Mesh Collider** – Adds or replaces MeshCollider's mesh inside generated prefab object, to be the same lowpoly mesh as in the used MeshRenderer.
- **Material** – Generated prefab can use material from the source object or LMG can create a new material with selected shader.

SAVE



- **Format** – In the current version LMG saves generated mesh only in ***.asset*** format.

- Mesh **Compression** setting allows reducing generated file size by lowering numerical accuracy of the mesh. Instead of 32-bit floats, lower size fixed number will be used to represent mesh data.

  Note, more compression introduces more artifacts in vertex data (*position*, *normal*, *uv, etc*).

- **Components** – By default generated mesh has all components that are available in the source mesh. Using this option unused components can be excluded from saved mesh and file size will be reduced.

- **Prefab Flags** – Overrides generated prefab's flags.

- **Name** – Adds prefix & suffix to the all generated file's names.

- **Location** – Generated files save location. Can be any folder on the hard drive, inside or outside of a project.

RUN-TIME API

Lowpoly Mesh Generator method can be brought into scope with this using directive:

C#
```
using AmazingAssets.LowpolyMeshGenerator;
```

Now Unity mesh class will have new extension method:

```
Mesh GenerateLowpolyMesh(Texture2D[] textures, Vector4[] texturesTilingOffset, Color[] colors,
                         int uvIndex, SamplingType samplingType,
                         SourceVertexColor sourceVertexColor, AlphaType alphaType,
                         bool mergeSubmeshes,
                         LowpolyUV saveLowpolyUV = LowpolyUV.None,
                         Buffer saveTriangleCenter = Buffer.None);
```

- `textures` – Array of the baked textures with the same size as the source mesh's submesh count. Each texture will be baked to the corresponding submesh by its index.
  In the case of providing only one texture, it will be used for all submeshes.
  Pass **null** to bake default white texture.

- `texturesTilingOffset` – Bake textures tiling & offset values. Array must be the same size as textures.
  Pass **null** to use default values.

- `colors` - Array of the baked colors with the same size as the source mesh's submesh count. Each color will be baked to the corresponding submesh.
  In the case of providing only one color, it will be used for all submeshes.
  Pass **null** to bake white color.

- `uvIndex` – Mesh UV index used for texture rasterization. By default it is 0. Can be in range of [0, 7].

- `samplingType` – Generated lowpoly color style.

  public enum SamplingType
  ```
  AverageColor = 0,
  MiddleUV = 1,
  FirstIndex = 2
  ```

Note, generated **Lowpoly UV** can use only **Middle UV** or **First Index** values.

- `sourceVertexColor` – Combines source mesh original vertex color with generated lowpoly color.

```
public enum SourceVertexColor
DoNotUse,
UseOriginal,
UseLowpoly
```

- `alphaType` – Generated lowpoly vertex color's Alpha value.

```
public enum AlphaType
DefaultValue = 0,
TextureAlpha = 1,
ColorAlpha = 2,
VertexColorAlpha = 3,
One = 4,
Zero = 5
```

- `mergeSubmeshes` – Combines generated mesh's submeshes.

- `saveLowpolyUV` – Generates lowpoly UV0 and saves it inside **xy** or **zw** channels.

```
public enum LowpolyUV
None = 0,
XY = 1,
ZW = 2
```

- `saveTriangleCenter` – Saves triangle's center position as Vector3 inside one of the mesh data buffer.

```
public enum Buffer
None = 0,
UV0, UV1, UV2, UV3, UV4, UV5, UV6, UV7,
Color,
Normal,
Tangent
```