

FALCOENGINE — Built-in Materials Guide

A practical guide to choosing and using built-in materials: purpose, differences between similar materials, and selection rules for MeshRenderer, Terrain, DecalRenderer, ParticleSystem, Sky and Water.

Main principle Opaque lit geometry usually uses `DeferredStandard`. `ForwardStandard` is needed primarily for true transparency, mesh decals and special Forward tasks. Component materials `Terrain`, `ParticleSystem`, `Sky`, `Clouds` and `Water` should be used only through the corresponding component.

This material was prepared from a runtime audit of the built-in shaders. `DeferredStandard`, `ForwardStandard` and `UnlitStandard` tests: PASS.

Contents

1. Material and shader
2. Main object materials
3. Terrain and Triplanar
4. Decals
5. Vegetation
6. Particles
7. Sky and clouds
8. Water
9. System Deferred Light
10. Quick material selection

1. Material and shader

A `.shader` file defines the rendering algorithm: passes, lighting, blending, depth writing and available parameters. A `.material` file is an instance of a shader with specific textures, colors and switches.

In practice, a built-in material asset should be used as a template: create a copy in the project folder, assign the required textures and enable the corresponding **Has ... map** switches. A filled texture slot without the switch enabled does not activate the map.

Main PBR maps

- **Albedo** — base color of the surface, without painted-in highlights or shadows.
- **Normal** — fine relief without increasing the polygon count.
- **Metallic** — separates metals from dielectrics.
- **Roughness** — width and sharpness of reflection: a lower value means a smoother surface.
- **AO** — local shading of recesses, affecting the ambient and indirect component.
- **Emission** — the material's own glow.

2. Main object materials

Material	When to use	Limitations
<code>DeferredStandard</code>	Main PBR material for opaque objects: environment, walls, furniture, weapons and characters.	Not intended for true alpha blending.
<code>ForwardStandard</code>	Semi-transparent surfaces, mesh decals and objects that require Forward rendering.	Recalculated per light source; do not use in bulk instead of Deferred without need.
<code>UnlitStandard</code>	Screens, markers, decorative images and elements whose color must not depend on light.	Does not give a physically lit surface.
<code>UnlitStandardAdditive</code>	Flashes, lasers, energy effects and halos on a dark background.	Not suitable for glass or ordinary transparency.

DeferredStandard and ForwardStandard

Both materials support a similar PBR set. The difference lies in the rendering stage.

- **DeferredStandard** first writes surface properties into the G-buffer, after which lighting is computed for the scene. This is the main choice for opaque geometry and multiple light sources.
- **ForwardStandard** computes lighting directly while drawing the object. This path is required for alpha transparency and special Forward effects.

Selection rule Opaque object — `DeferredStandard`. Semi-transparent glass or a special Forward effect — `ForwardStandard`.

Unlit and Additive

`UnlitStandard` uses ordinary color and alpha blending, but does not react to Directional, Point and Spot lights.

`UnlitStandardAdditive` adds its color to the background: black practically disappears, while bright areas create a glow.

3. Terrain and Triplanar

Material	When to use	Limitations
<code>DeferredStandardTerrain</code>	Main opaque material of the Terrain component. Blends terrain layers and takes part in Deferred lighting.	Assigned to Terrain, not to an ordinary <code>MeshRenderer</code> .
<code>StandTerrain</code>	Forward variant of Terrain for Forward scenes and special compatibility requirements.	Potentially more expensive than Deferred with many light sources.
<code>DeferredTriplanar</code>	Rocks, caves, cliffs and meshes with poor UV unwrapping. Projects textures along world axes.	Not a Terrain material; layer blending requires vertex color <code>COLOR0</code> .

`DeferredTriplanar` is especially useful on steep slopes, where an ordinary UV texture gets stretched. Three sets of Albedo, Normal and Roughness are blended through the mesh's vertex colors.

4. Decals

`DeferredDecals.material` is applied only through `DecalRenderer`. The projector modifies an existing Deferred surface and is suitable for dirt, tracks, cracks, inscriptions and damage.

- The material must not be assigned to an ordinary `MeshRenderer`.
- A surface outside the projector volume must remain unchanged.
- Several overlapping projected decals currently have a known composition limitation.

If a decal is separate geometry with its own UV unwrapping, use `ForwardStandard` with **Mesh decals** mode enabled.

5. Vegetation

Material	When to use	Limitations
<code>NatureaGrass</code>	Grass and details created by the Terrain system.	Not intended for an ordinary Cube.
<code>StandTreeLeavs</code>	Foliage meshes, separate planes and intersecting vegetation cards.	The texture must have a correct alpha mask.

Both materials use alpha cutout: a pixel is either displayed or removed completely. For foliage this is usually more reliable than ordinary transparency, since it reduces sorting problems and allows the surface to take part in the depth test.

6. Particles

`Particles.material` is assigned through `ParticleSystem`.

- **Additive = off** — smoke, dust and soft particles with ordinary transparency.
- **Additive = on** — fire, flashes, sparks and energy effects.
- **Z-Write = off** — typical setting, so that particles do not hide geometry and other particles behind them.

7. Sky and clouds

Material	When to use	Limitations
<code>SkyboxProcedural</code>	Dynamic sky with time of day, sun, stars and	Requires more setup than a ready-made cubemap.

	textured cloud layers.	
SkyboxStandart	Static cubemap panorama when sky dynamics are not needed.	Needs a correct cubemap without swapped faces.
Clouds / CloudSystem	Standalone dynamic cloud system with Coverage and Density.	Better to use the component's internal material with the automatically assigned noise texture.

Skybox materials are assigned as **Sky Material** in the Lighting window, not to a MeshRenderer component.

8. Water

`StandardWater` is a hidden system shader created by the Water component. A separate material asset is usually not required.

- Supports waves, foam, reflection, refraction, caustics and depth-based coloring.
- `ForwardStandard` can create a blue semi-transparent plane, but does not replace a full Water component.

9. System Deferred Light

`StandardDeferredLight.shader` cannot be assigned to an object. It reads the G-buffer and computes lighting for the Deferred scene. Its operation is verified indirectly through DeferredStandard, Deferred Terrain, Triplanar and projected Decals.

10. Quick material selection

Task	Recommended choice
Ordinary opaque object	DeferredStandard
Semi-transparent glass	ForwardStandard
Object that does not react to light	UnlitStandard
Glow or VFX	UnlitStandardAdditive
Ordinary terrain	DeferredStandardTerrain
Forward terrain	StandTerrain
Rock or mesh without good UVs	DeferredTriplanar
Projected dirt or tracks	DeferredDecals + DecalRenderer
Terrain grass	NatureaGrass
Foliage on cards	StandTreeLeaves
Smoke, fire, sparks	Particles + ParticleSystem
Dynamic sky	SkyboxProcedural
Static panorama	SkyboxStandart
Dynamic clouds	CloudSystem
Full water surface	Water

Before testing For lit materials, assign the textures, enable the corresponding **Has ... map** switches, disable Realtime GI for the first check, and rotate the Directional Light by 180 degrees. The lit side should follow the light source.