Advanced Rendering

MATERIALS, POSTEFFECTS AND SCENE COMPOSITION

New Features

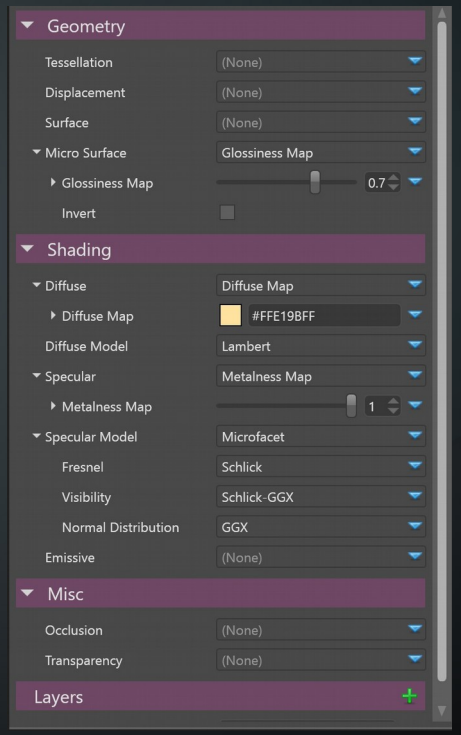
- New **Material** System
  - Allowing multi-layered PBR materials
- **PostProcessing** Effects
  - Along a simple and powerful ImageEffect API to ease the development of post effects
- **Scene Composition**
  - Allowing complex composition of multiple scene

Materials Overview

- A Material can be described by
  - **Color** and (**micro)Surface** Attributes
    - DiffuseMap, NormalMap, GlossinessMap, SpecularMap…etc
  - **Shading** Attributes
    - Diffuse Lambert, Specular GGX, Transparent…etc.
  - A **Composition of Attributes** by inheriting, reusing and combining basic or complex materials

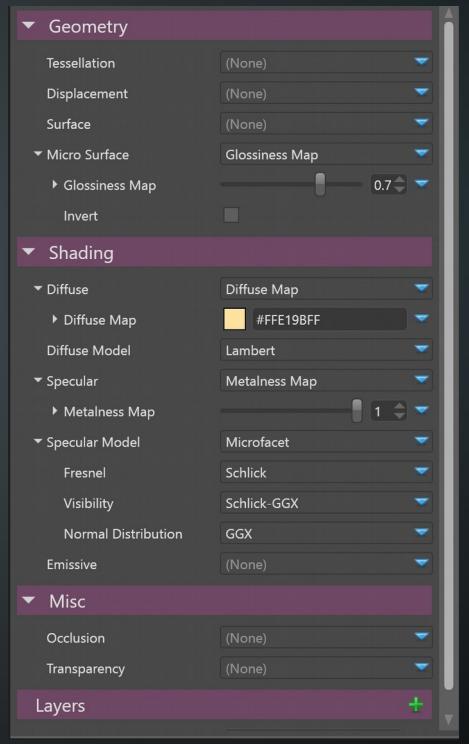
Material Geometry Attributes

- **Tessellation**
  - Flat or PN with support for Adjacent Edge Average
- **Displacement**
  - Displacement Maps: At Vertex, Tessellation or Pixel stage
- **Surface**
  - Normal Maps
- **Micro-Surface**
  - Glossiness Maps



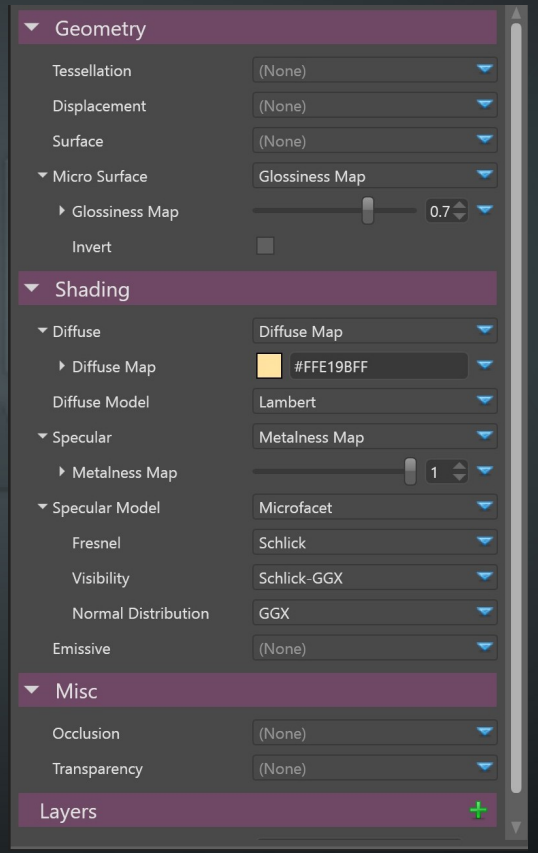
Material Shading Attributes

- **Diffuse**
  - Base Color
  - Shading Model (Lambert…)
- **Specular**
  - Color: **Metalness** or **Specular** workflow both at the same time
  - Shading Model: Microfacet with all variations (NDF, Visibility, Fresnel) with latest GGX/Disney models…



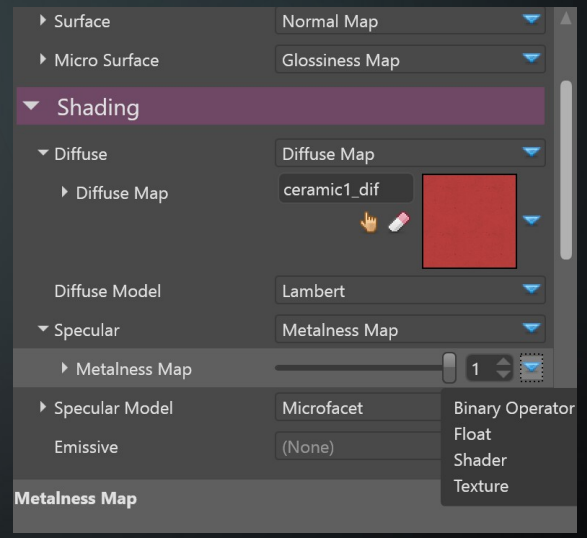
Material Shading Attributes

- **Emissive** Shading models
  - Emissive maps, Heat maps…
- **Transparency**
  - Cutoff, Additive blending…
- Misc
  - **Occlusion** maps
  - **Cavity** maps



Material Attributes : Value providers

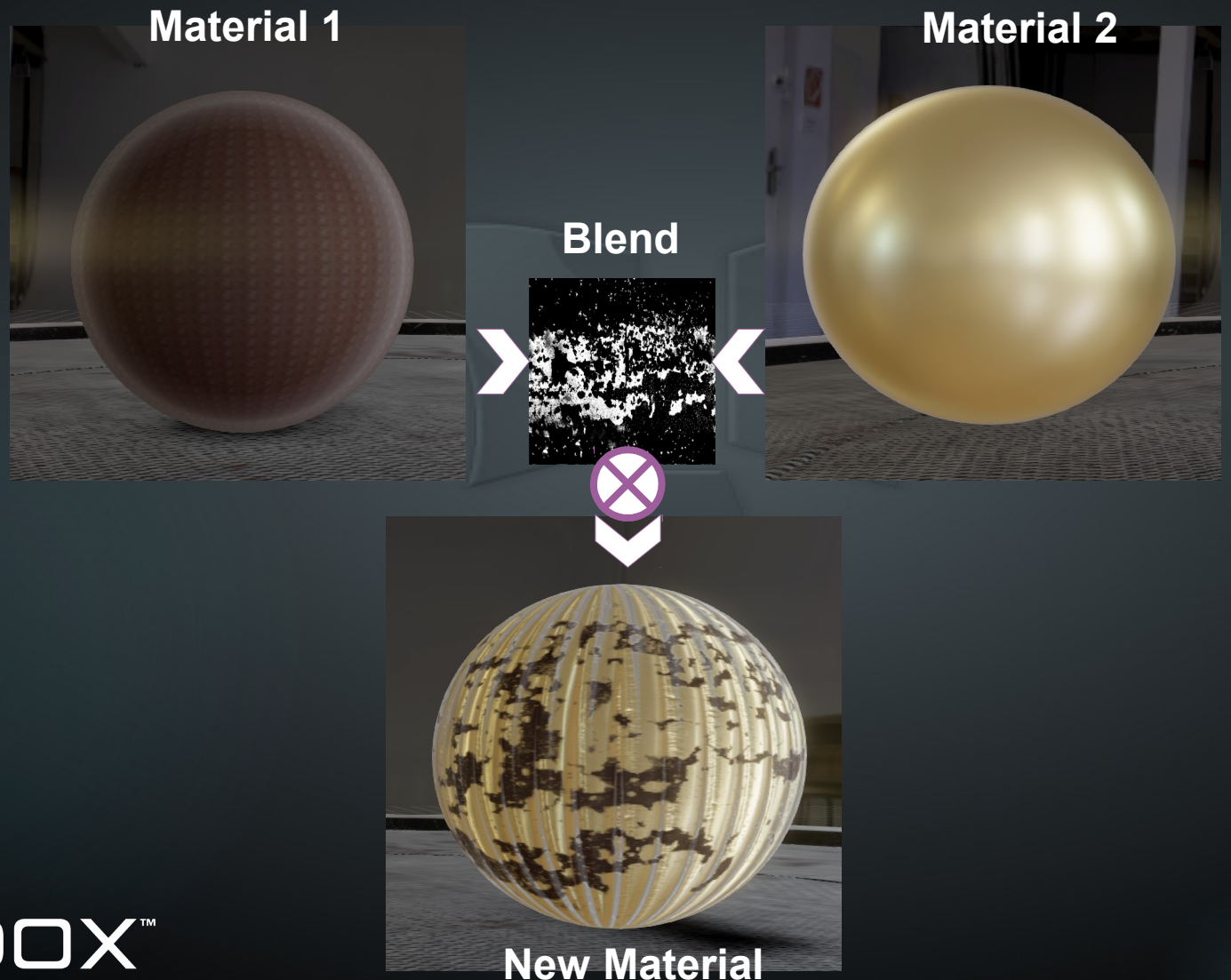
- Attributes are parametrized by “value” providers:
  - A **constant** value (e.g Metalness set to 1.0f)
  - From a **texture** (e.g. Diffuse Map)
  - From a **shader function** (e.g. Procedural maps)
  - From a **vertex attribute**
  - A **binary operator** between 2 values
- A value provider is either a **Color** or a **Float**



Material Composition: Layers

- We use a simple approach by using layers
- A Layer defines:
  - A reference to a **Material**
  - A **blending map** (e.g. float, texture…etc.)
  - Layer **overrides**: `uv` scale for the material…
- Layers can use different BRDF models (only working for forward/+ renderers)

Material Composition



Material Extensions

- Extensible
  - Leverage our modular shader language
  - Easily add new custom material attributes and shading models
  - Add specific blender function for a particular attribute (e.g. normal maps)
  - A material attribute can impact different stages
    - Vertex, Tessellation, Pixel
- Future
  - Allow to integrate graph node based Materials
  - Attributes squashing: combine textures at compilation time

PostEffects

- Comes with several streamlined post-effects
  - Depth Of Field
  - Bloom
  - Glares
  - Lens Flares
  - Tone Mapping
  - Anti-Aliasing, Vignetting, Film Grain…

Post-Effects API

- **Easy to use**
  - `PostFx.SetInput(…), PostFx.SetOutput(…), PostFx.Draw()`
- **Easy to extend** or develop new custom effects
  - Add for example your own new ToneMap operator…etc.
- **Efficient**
  - Easily combine many one-pass post-effects into an **optimized single shader**
- Supports both **Pixel and Compute shaders post-fx**

Scene Composition

- **Easy and powerful way to compose the rendering of your ga**me
- Layered based
- A **Graphics mixer** similar to an Audio Mixer
- **Render multiple Scene and combine them together**
  - From multiple cameras
  - To different outputs
  - By Filtering entities from the Scene
- **Integrated with Post-Effects**
- **Extensible**: Add your own custom renderer to the pipeline
- **Mix easily Deferred and Forward(+)** rendering in the same pipeline

Coming Next

- Add Deferred and Forward+ Rendering (previously deferred lighting)
- Physically based Camera and (area) Lights
- Real-time Global Illumination
- Screen-Space Reflection and Local Probes
- More post-effects (Motion Blur…etc.)