Materials Basics

The virtual world, similar to the real world, contains many different objects, and we can see and distinguish them due to the interaction of the their materials with light. It is materials that make objects green, transparent or, for example, mirror-like. PBR (Physically-Based Rendering) system was developed to ensure uniformity of materials and make them look as realistic as possible. This system considers the properties of each material and uses mathematical formulas to accurately simulate how light behaves when it interacts with the object's surface. In other words, it creates a realistic environment where light behaves as it would in the real world. This model is widely used in various engines, including UNIGINE.

In terms of UNIGINE, a **material** is a rule defining how the surface (or in some cases a volume) will look like: its color, interaction with lights, reflection and refraction parameters, etc. The material can imitate red plastic, yellow sand, transparent glass, shiny metal, muddy water and much more. The set of basic material properties includes:

- **States** that specify conditions, based on which the Engine applies corresponding shaders, textures, and parameters. For example, you can set the Water state to enable/disable sea foam. Depending on the selected state (foam / no foam), the engine will choose the required shaders.
- A set of **Textures**. As a rule, a base material has several textures. This is caused by the following reasons:
    - Shader may require more than one texture.
    - Different rendering stages user different textures.
    - A state may require a specific texture.
- **Parameters** — arguments passed to the shader that define how the shader will be used.
- **Shaders** — basis of materials, special programs that actually draw the material, taking into account the defined parameters, modes, textures, and so on. Each material has several shaders, and one of them is used depending on the selected mode and stage of drawing. In addition to the basic shader types — vertex, fragment and geometry shaders, there are other types.