Script 04

- Illumination: using different light sources.
- Illumination: using different materials.
- Shading: flat shading vs smooth shading vs Phong shading.
- Application example: randomly generated scene.

3.1 Illumination: using different light sources

Open the folder **01_ex_Illumination**.

Analyze the example. The scene has just **one sphere** – its color is gray and it is shiny – and **one point light source** – red light; placed at (10, 10, 2) –; there is also a **small ambient illumination component**.

Questions:

- **Remove the point light source** and keep the ambient component. What do you see?
- **Remove the ambient component** and keep the point light source. Do you notice any difference?

Tasks – Adding and animating light sources

- Animate the light source; it rotates counterclockwise around the YY axis.
- Add **a second point light source** to the scene placed at (-7.5, 7.5, 2); light color is green. It rotates clockwise around the YY axis, with smaller speed.
- Add **a third point light source** to the scene placed at (0, 3, 10); light color is blue.
- Add **a fourth point light source** to the scene placed at (0, -10, 0); light color is white.

Tasks – Different light sources

- **Add two similar spheres** to the scene: one on the left, the other on the right. Notice the illumination effects.
- **Keyboard**: use keys to **control the type of light source** – **point light** source, **directional light** or **spotlight**. Notice the illumination effects. IDEA: **add** or **remove** light sources from the scene.
- Add **shadows to the scene**. Notice the illumination effects. Use the keyboard to enable or disable shadows.

Tasks – Adding more models

- **Add three cubes** above the spheres. They should **rotate with different speeds** around their **vertical axis**.

3.2 Illumination: using different materials

In addition to light sources, illumination effects also depend on the **material properties** associated with each model.

Diffuse reflection, computed using the Gouraud method, can be used to render dull (not shiny) models – **THREE.MeshLambertMaterial**.

Diffuse and specular reflection, computed using the Phong method, can be used to render metallic (shiny) models – **THREE.MeshPhongMaterial**.

You can use the models and light sources of the previous example to experiment with different materials.

Tasks

- Change the **light sources** so that they emit **white light**.

- Assign different materials to the models using **THREE.MeshLambertMaterial** and **THREE.MeshPhongMaterial**.

- Compare the illumination effects.

Appropriate material parameters (R,G,B) can be obtained from **empirical tables**, such as the following one:

	Ambient			Diffuse			Specular			shininess
Bronze	0.21	0.13	0.05	0.71	0.43	0.18	0.39	0.27	0.17	25.6
Polished Bronze	0.25	0.15	0.06	0.4	0.24	0.1	0.77	0.46	0.20	76.8
Copper	0.19	0.07	0.02	0.7	0.27	0.08	0.26	0.14	0.08	12.8
Polished Copper	0.23	0.08	0.03	0.55	0.21	0.07	0.58	0.22	0.07	51.2
Chromium	0.25	0.25	0.25	0.4	0.4	0.4	0.77	0.77	0.77	76.8
Brass	0.33	0.22	0.03	0.78	0.57	0.11	0.99	0.94	0.81	27.9
Gold	0.25	0.20	0.07	0.75	0.60	0.23	0.63	0.56	0.37	51.2
Polished Gold	0.25	0.22	0.06	0.35	0.31	0.09	0.80	0.73	0.21	83.2
Polished Silver	0.23	0.23	0.23	0.28	0.28	0.28	0.77	0.77	0.77	89.6
Red Plastic	0.3	0.0	0.0	0.6	0.0	0.0	0.8	0.6	0.6	32.0
Very Shiny Blue	0.0	0.0	0.5	0.0	0.0	1.0	1.0	1.0	1.0	125.0
Gray	0.1	0.1	0.1	0.5	0.5	0.5	0.7	0.7	0.7	1.0

Note that the **shininess coefficient** has to be **multiplied by 256** (the maximum allowed value).

Tasks

- Instantiate some of the materials defined in the table above.
- Create several spheres and cubes and assign those materials.
- Compare the illumination effects obtained.

3.3 Shading: flat shading vs smooth shading vs Phong shading

Shading assigns a color to each pixel and is computed on the GPU pixel shader.

The simplest shading possibilities are:

- **Phong Shading:** the color assigned to each pixel is computed using the Phong illumination model and interpolated normal vectors.
- **Flat-Shading:** a constant color is assigned to all pixels representing a mesh triangle/face.
- **Smooth-Shading or Gouraud Shading:** the color assigned to each pixel is computed by averaging the color values associated with each triangle/face vertex.

In Three.js, you can choose how shading is carried out:

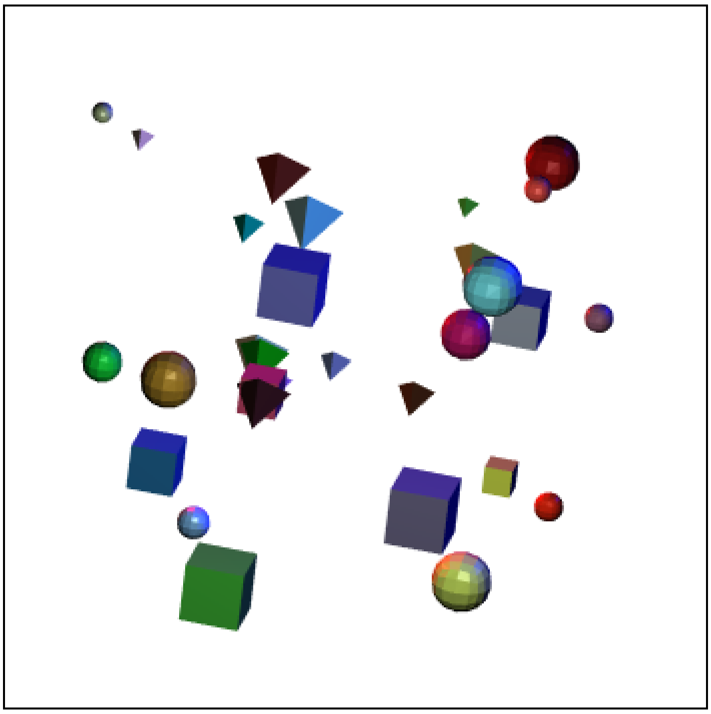
- **Phong Shading** is the default for **THREE.MeshPhongMaterial**.
- **Flat-Shading:** can be set (boolean attribute) for a **THREE.MeshPhongMaterial**.
- **Smooth-Shading or Gouraud Shading:** is the default for **THREE.MeshLambertMaterial**.

Tasks

- Compare the illumination effects obtained when using the different shading procedures.
- Use meshes defined by **more/less triangles**.

3.4 Application example: randomly generated scene

Create a new example, where the scene is randomly generated – see the figure below.



Goals

- Scene is made up of randomly chosen models: spheres, cubes, tetrahedra, etc.
- Model color, position and size are also randomly set.
- There are directional light sources illuminating the scene.
- Models rotate around the vertical axis of the scene.
- Each model also rotates around its own horizontal axis.

Suggestion

- Use the methods that allow for traversing the scene graph (**scene.traverse**) and for getting an array containing a scene graph's objects (**scene.children**).

Extra

- Add a new model / remove the last model using the keyboard.