

Computer Graphics

Light and Shadow

- 1) Create the **light.hpp** file containing a **Light** struct with the standard init(), destroy() and bind() functions. It should encapsulate all the data and functionality that is currently floating around in the Engine struct. Add the appropriate uniforms to our shaders (both default and shadow pipelines).

```
GLuint _shadow_width = 512;
GLuint _shadow_height = 512;
GLuint _shadow_texture; // cube map (6 textures)
std::array<glm::mat4x4, 6> _shadow_views;
glm::mat4x4 _shadow_projection;
// the following ones need to be uniforms in shaders as well
glm::vec3 _shadow_position = {1.0, 3.0, -0.5};
float light_range; // (new)
vec3 light_color; // (new)
```

- 2) Render two lights at once, each with their own shadow map. Place the lights in different positions such that the cast shadows are different.
Think about how to mix the light colors within the shader, since each light will contribute to **diffuse** and **specular** color portions of the current pixel differently!