God Rays and Shadow Maps

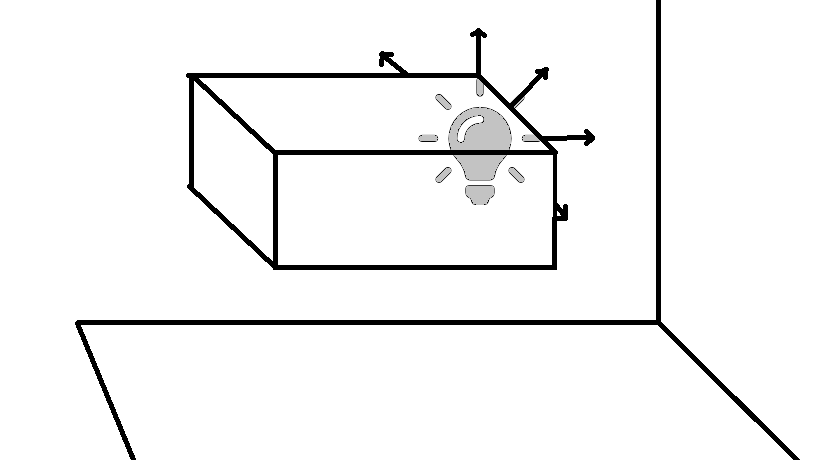
# Overview

We will implement:

1. A screen-space shader simulating crepuscular rays around objects occluding a point light.
2. Shadows using a shadow map rendered from the point of view of a directional light.

These will be rendered onto the same scene separately, one after the other.

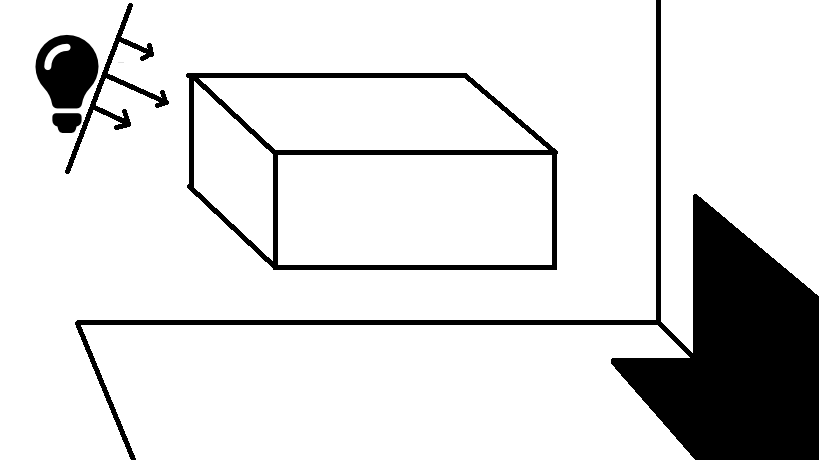
# God Rays



Also known as sun shafts or crepuscular rays, these will be rendered in a three pass process. The first pass renders the scene geometry solid black into an FBO. The second pass applies a screen-space shader to the FBO, drawing rays that attenuate outward from the light source. The third pass renders the geometry again using the proper shaders, then blends it together with the FBO to achieve the final image.

SOURCES: <http://http.developer.nvidia.com/GPUGems3/gpugems3_ch13.html>, <http://fabiensanglard.net/lightScattering/>

# Shadow Map



This is a non-raytraced method for drawing shadows cast by a directional light. A depth texture is rendered from the perspective of the directional light, with a frustum that encompasses the bounds of the actual camera. The depth map’s coords are then transformed into the actual camera’s space. The fragment shader for this scene samples the transformed depth texture and determines whether the fragment is further than the nearest occluder. If so, the pixel does not receive Phong shading (although ambient light is still applied).

SOURCES: <http://www.opengl-tutorial.org/intermediate-tutorials/tutorial-16-shadow-mapping/>