PA4 Milestone

Zi Wang
Wenlin Mao

Accomplishment and Plan

We first established the entire system structure and separated abstractions such as Ray, Intersections, and Camera into different classes. Then we started to implement the camera algorithm. We wrote the readfile.cpp to read in data and create a Camera instance and bind it to a global variable. Then we implemented two primitives class: Sphere and Triangle. With polymorphism, we stored the Primitive class which is the superclass of Sphere and Triangle in the Scene objects and build the intersect method in both Sphere and Triangle class correspondingly. Thus, when raytracer calls Primitives‘ intersect method, it will run the correct intersect method based on the primitive type.

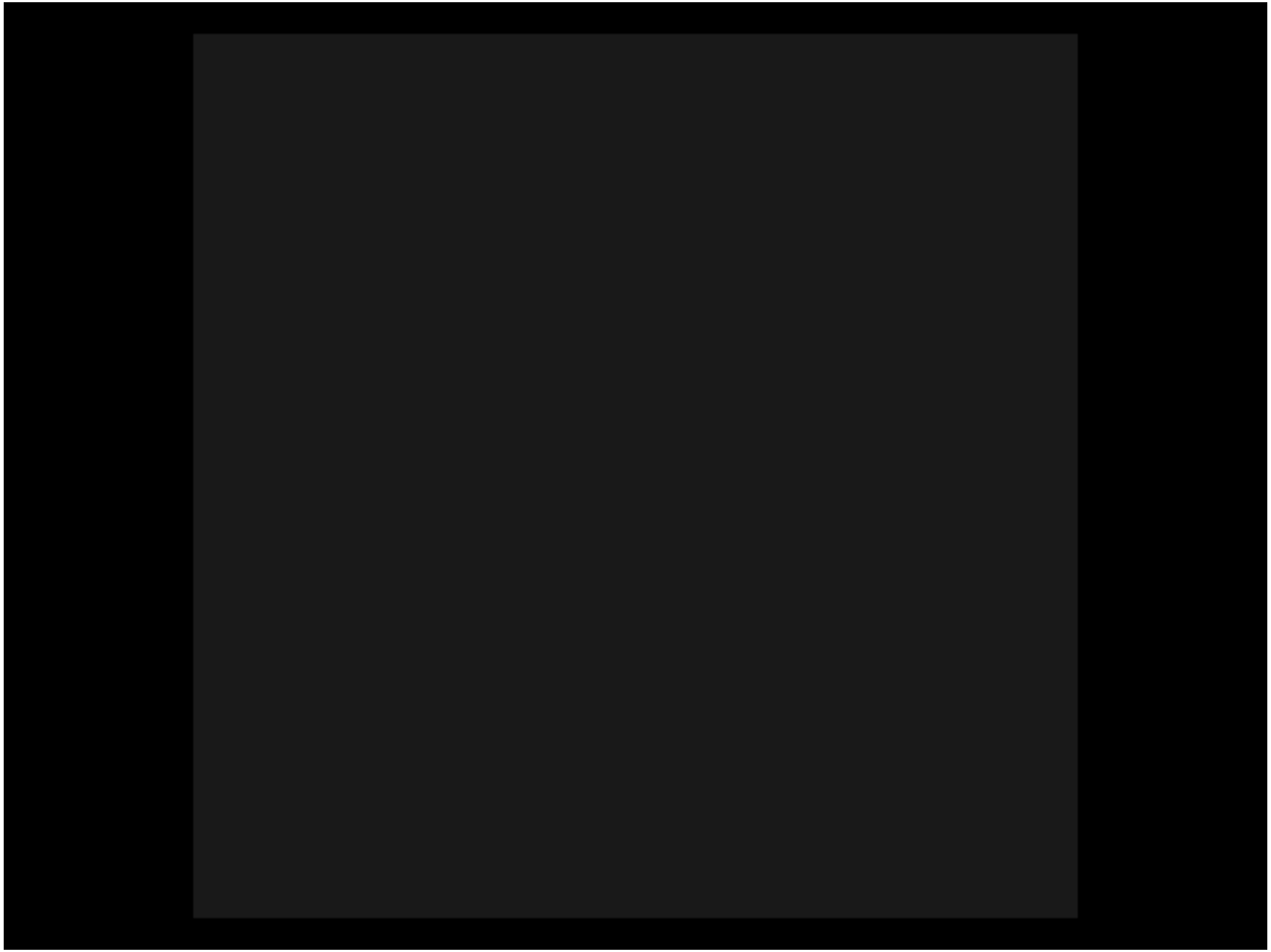
After that, we started to implement the mechanism of reading color in readfile.cpp. The readfile.cpp stored the color information and assigned them to the primitive object. Since we are not required to implement the light and shader, we just used the ambient term as our color.

We also implemented the transformation function. To do that we first read the transform command based on the write-up and store the transformation matrix in corresponding primitives. Then we applied transformation separately on Sphere and Triangle based on what the formula professor gave.

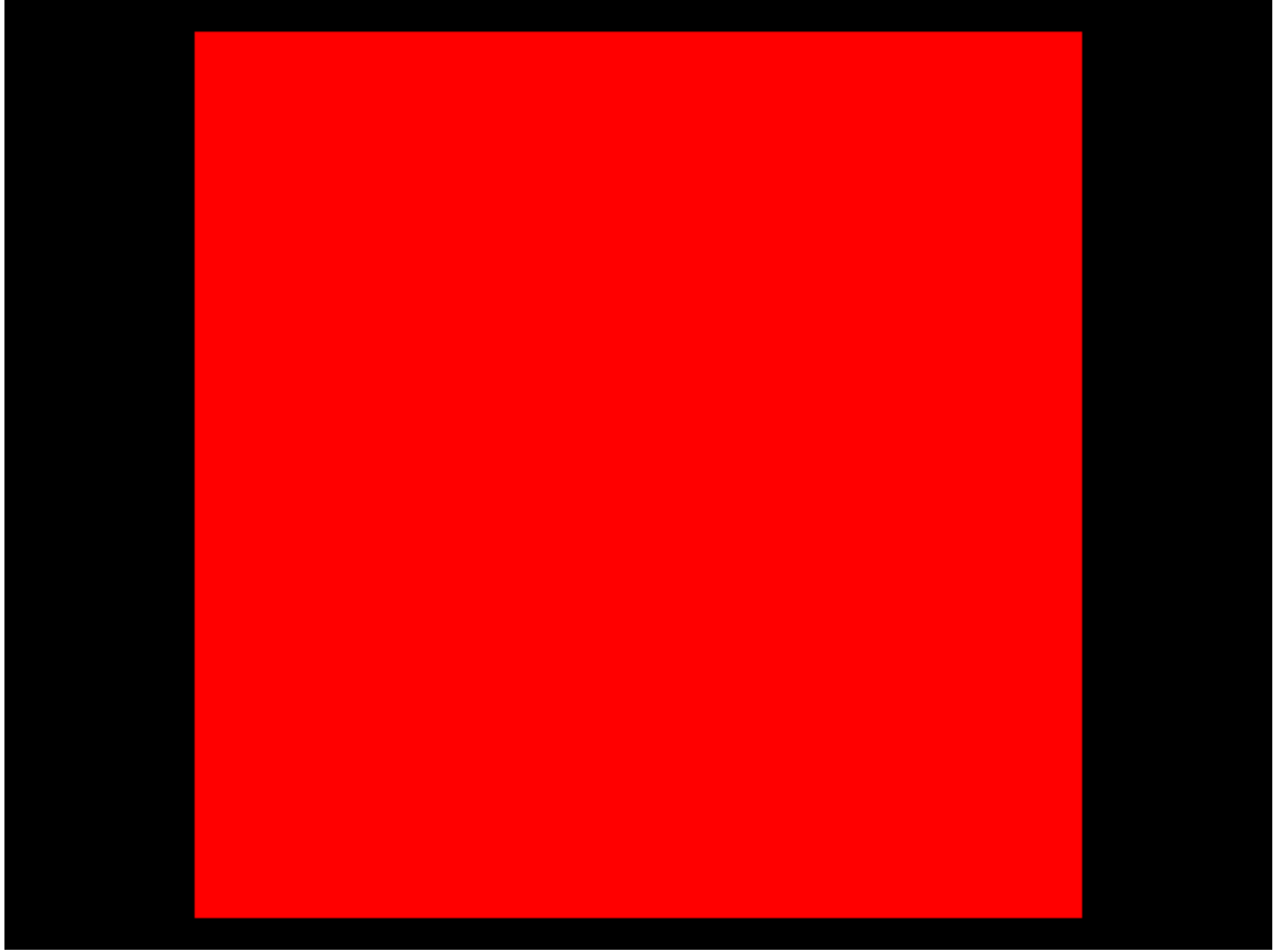
For the next step, we will first implement the color function using Intersection’s findColor method. And also implement the mechanism of reading light in readfile.cpp around March 13th. Then we will refactor the code and start to implement the acceleration structure to increase our program‘s speed by March 15th. If we have spare time, we will implement some other primitive structure for the extra credit.

Additional Comments

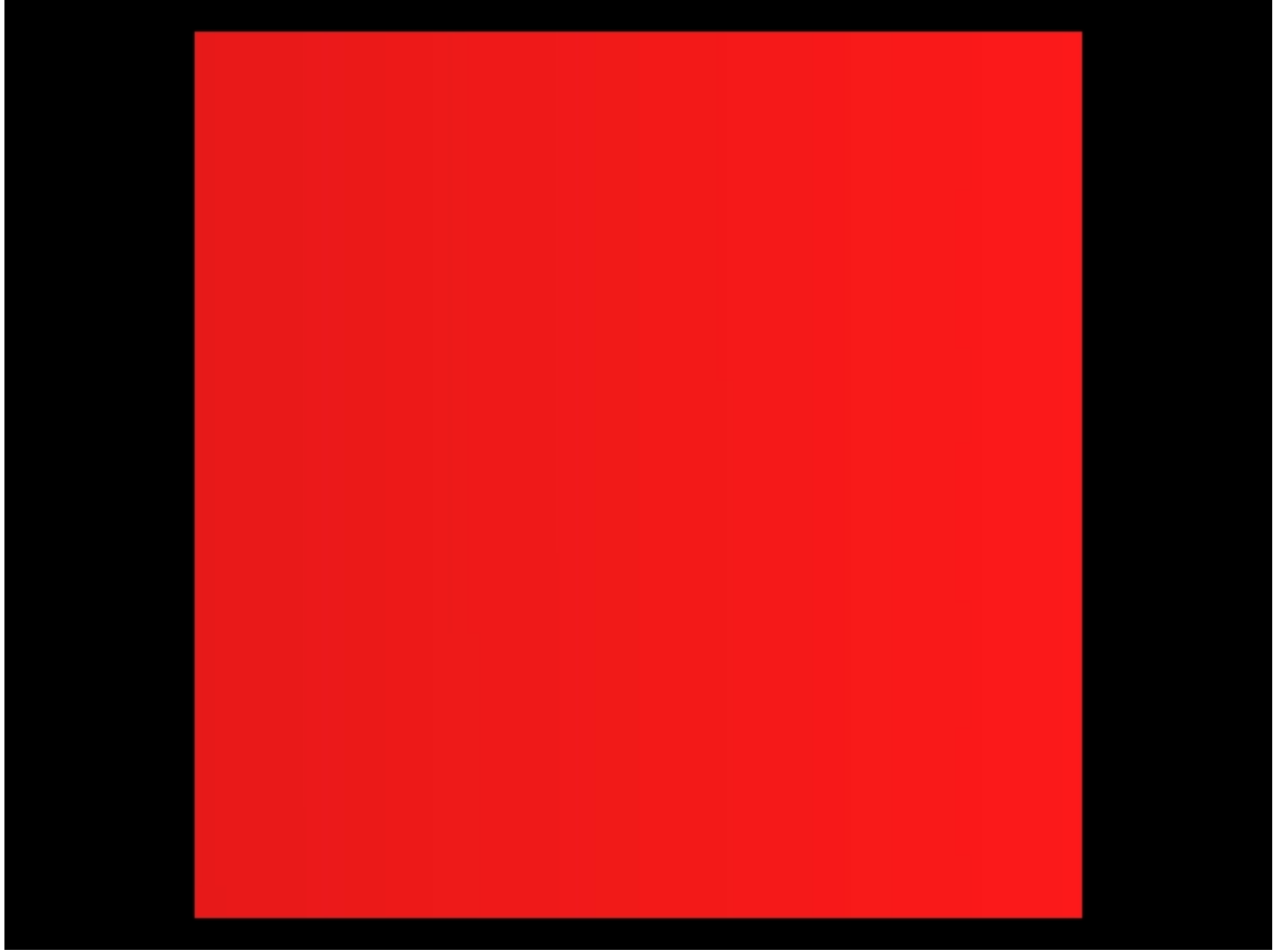
We also finished the transformation part of the assignment. Additionally, since we used constant ambient as our color, scene 1 have grey instead of red due to the ambient set in Scene1.test. We also include the red version of the scene1 by changing the ambient to 1 0 0.



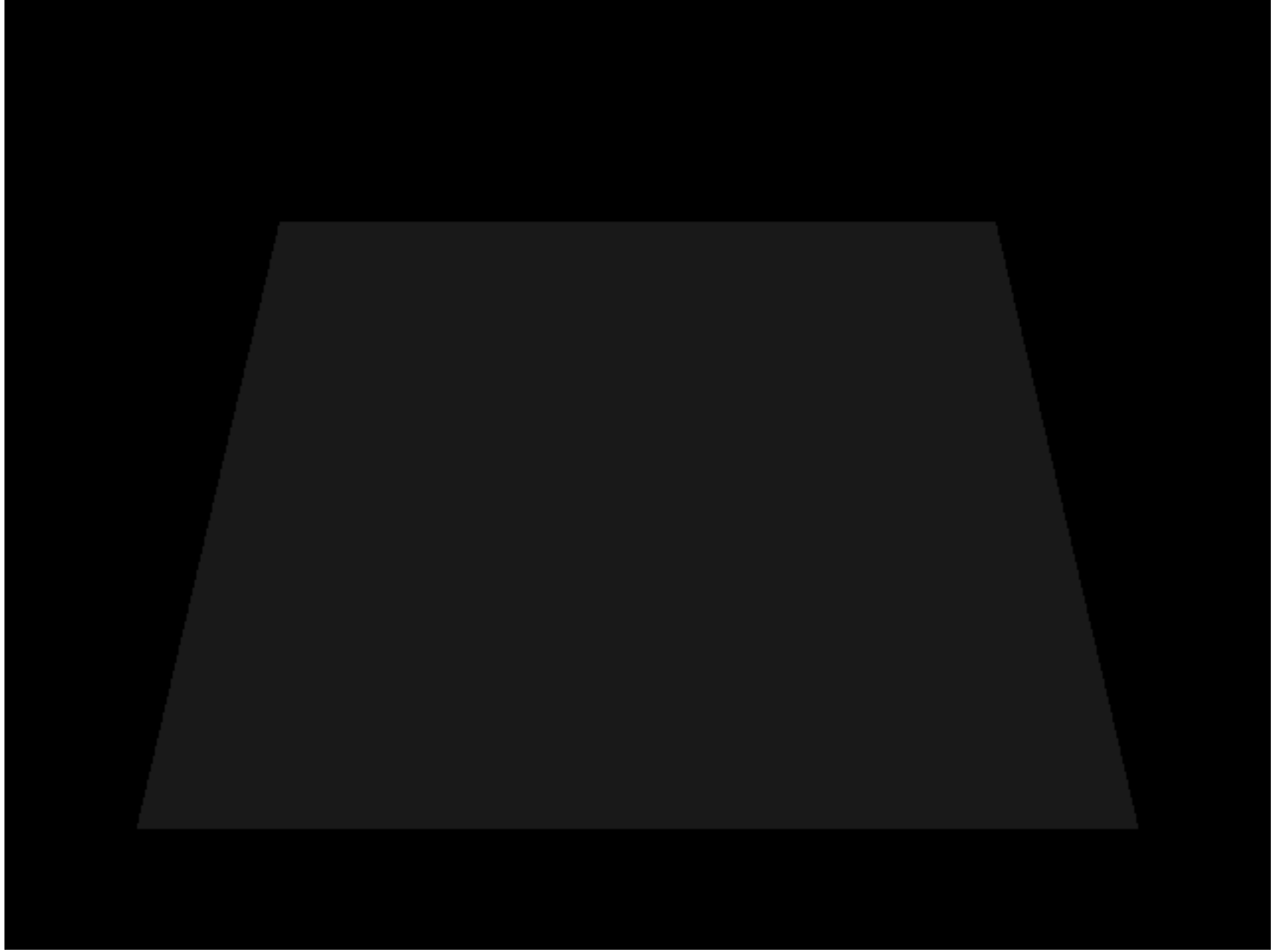
Our result with ambient 0.1 0.1 0.1 as default
Camera1 Scene1



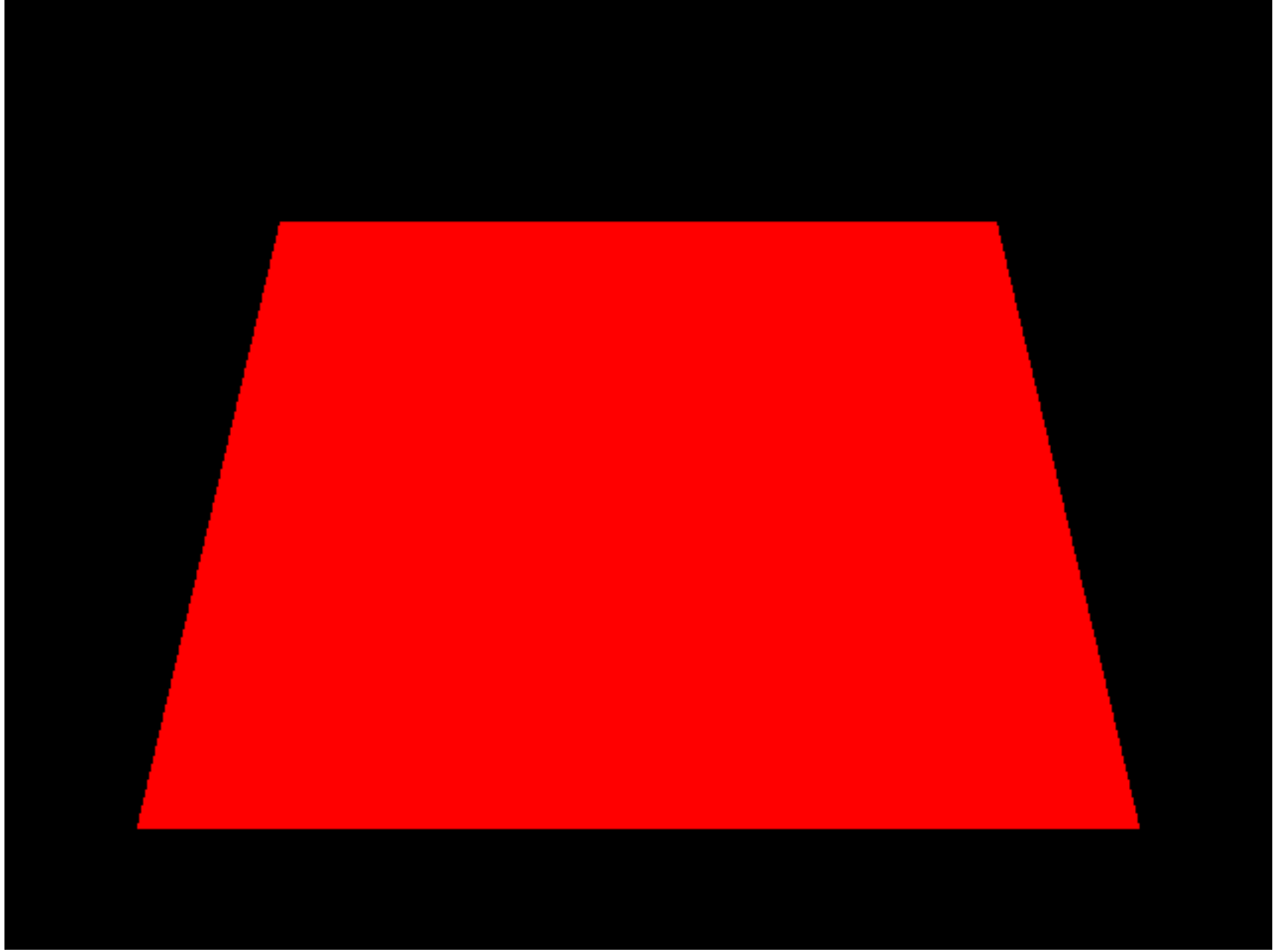
Our result with ambient 1 0 0
Camera1 Scene1



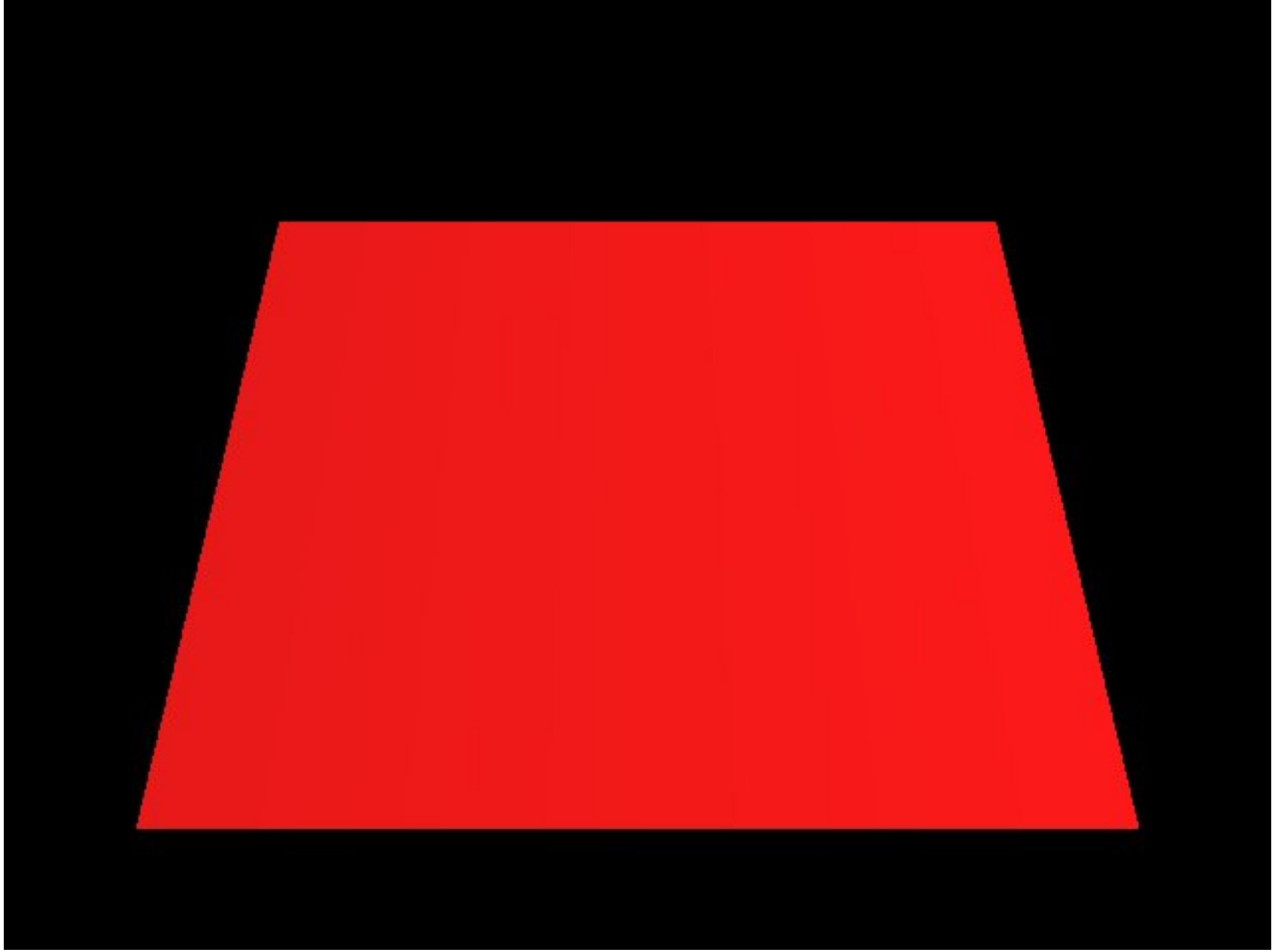
Given Solution
Camera1 Scene1



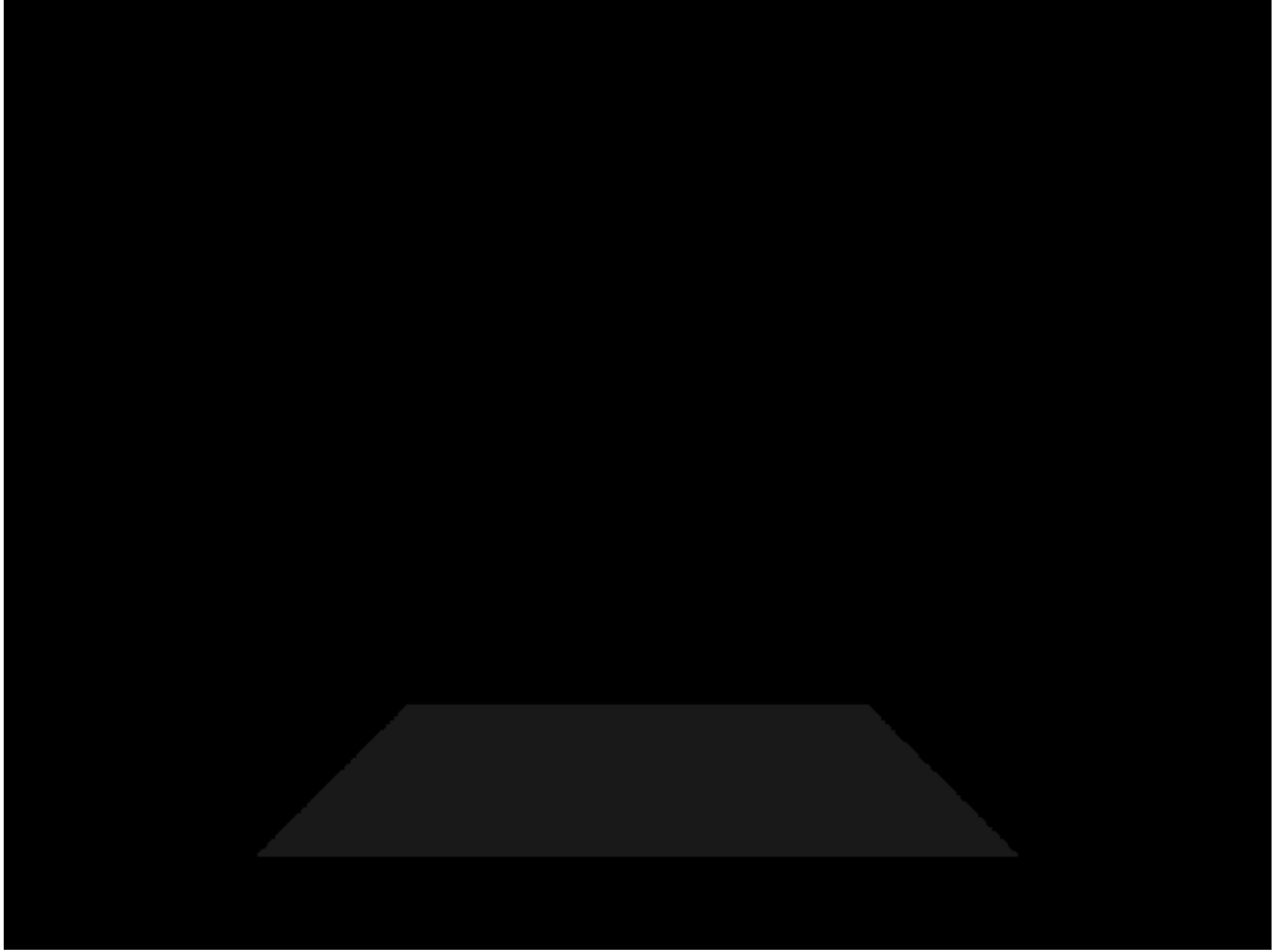
Our result with ambient 0.1 0.1 0.1 as default
Camera2 Scene1



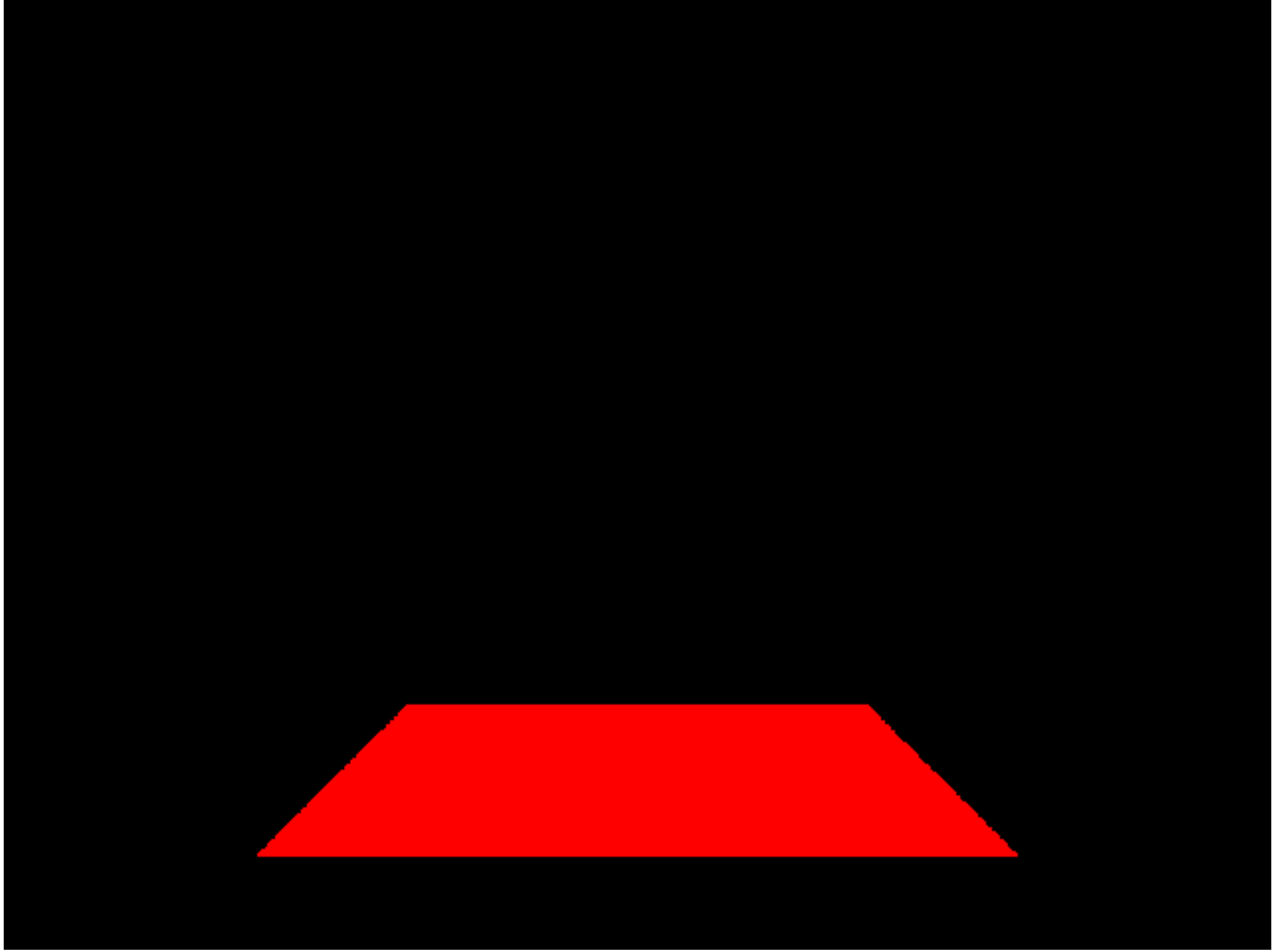
Our result with ambient 1 0 0
Camera2 Scene1



Given Solution
Camera2 Scene1



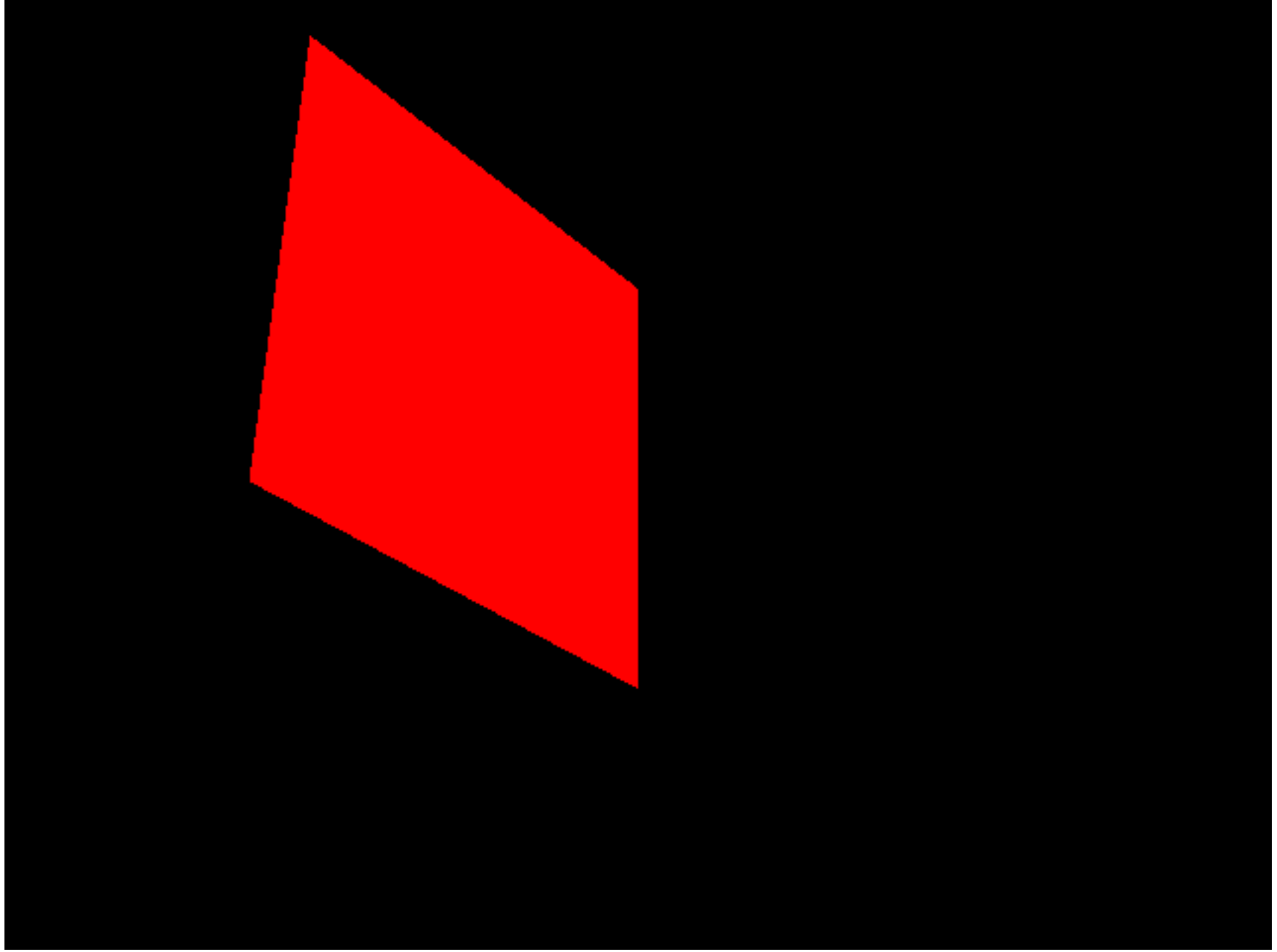
Our result with ambient 0.1 0.1 0.1 as default
Camera3 Scene1



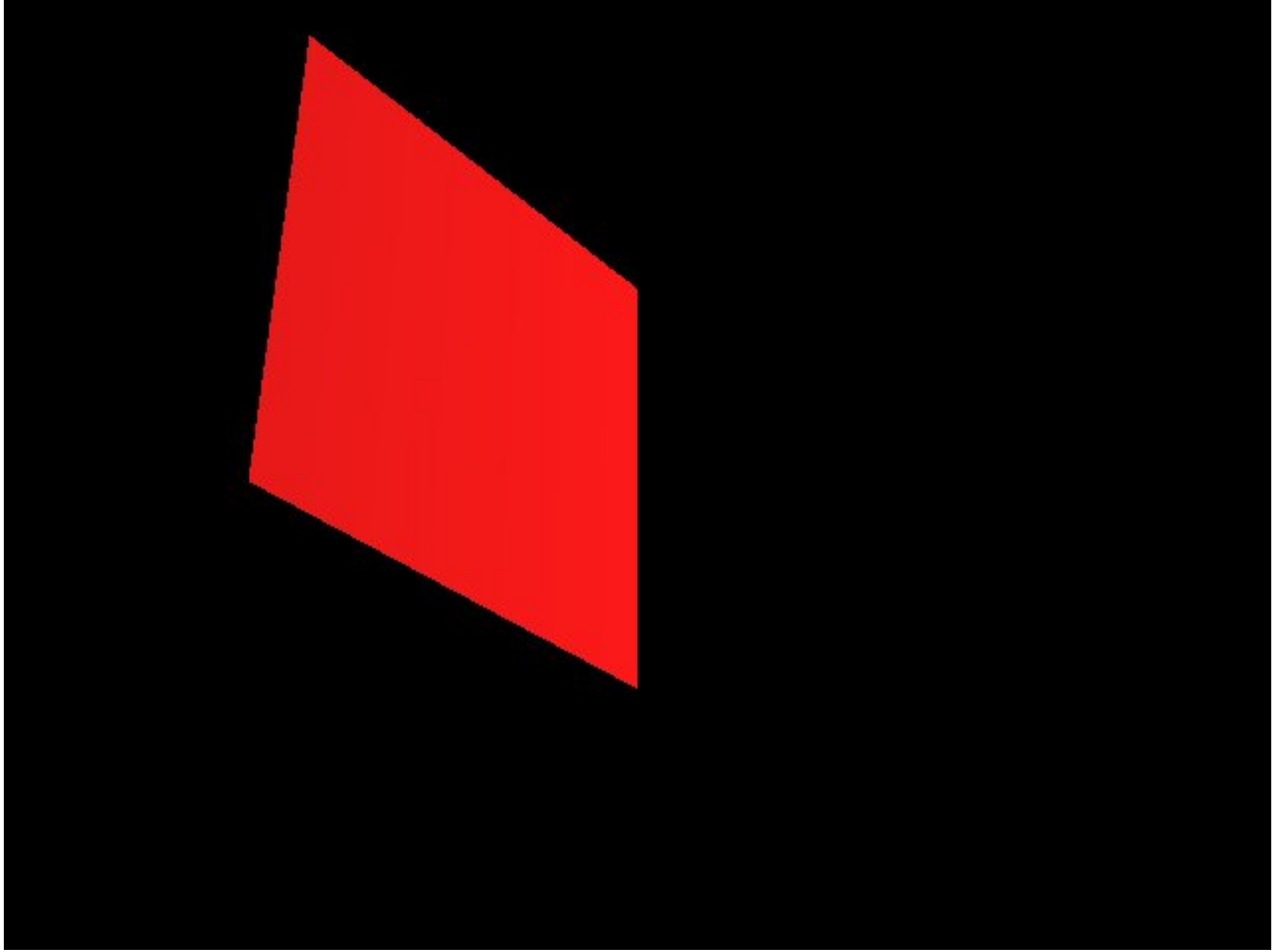
Our result with ambient 1 0 0
Camera3 Scene1

Given Solution
Camera3 Scene1

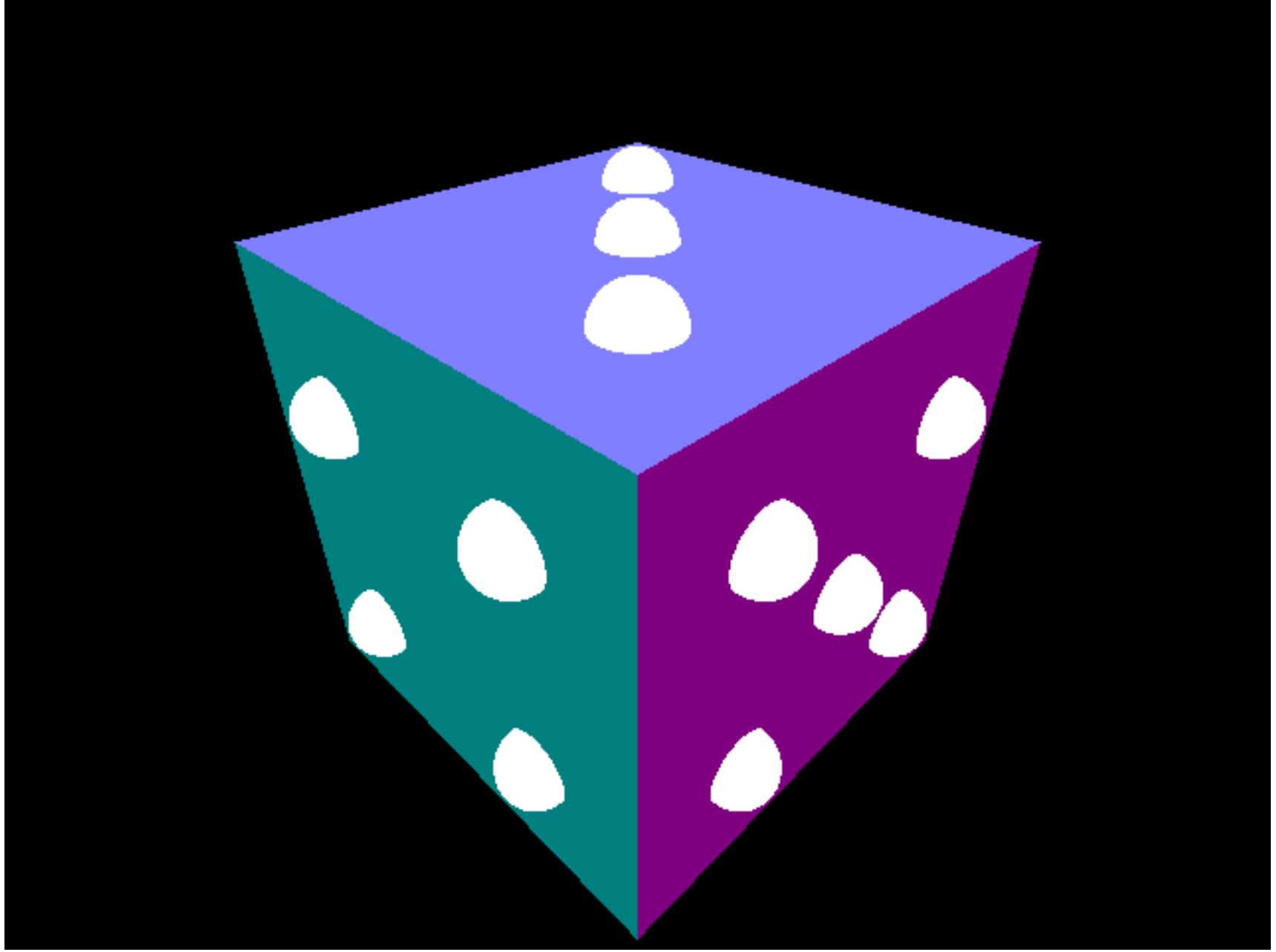
Our result with ambient 0.1 0.1 0.1 as default
Camera4 Scene1



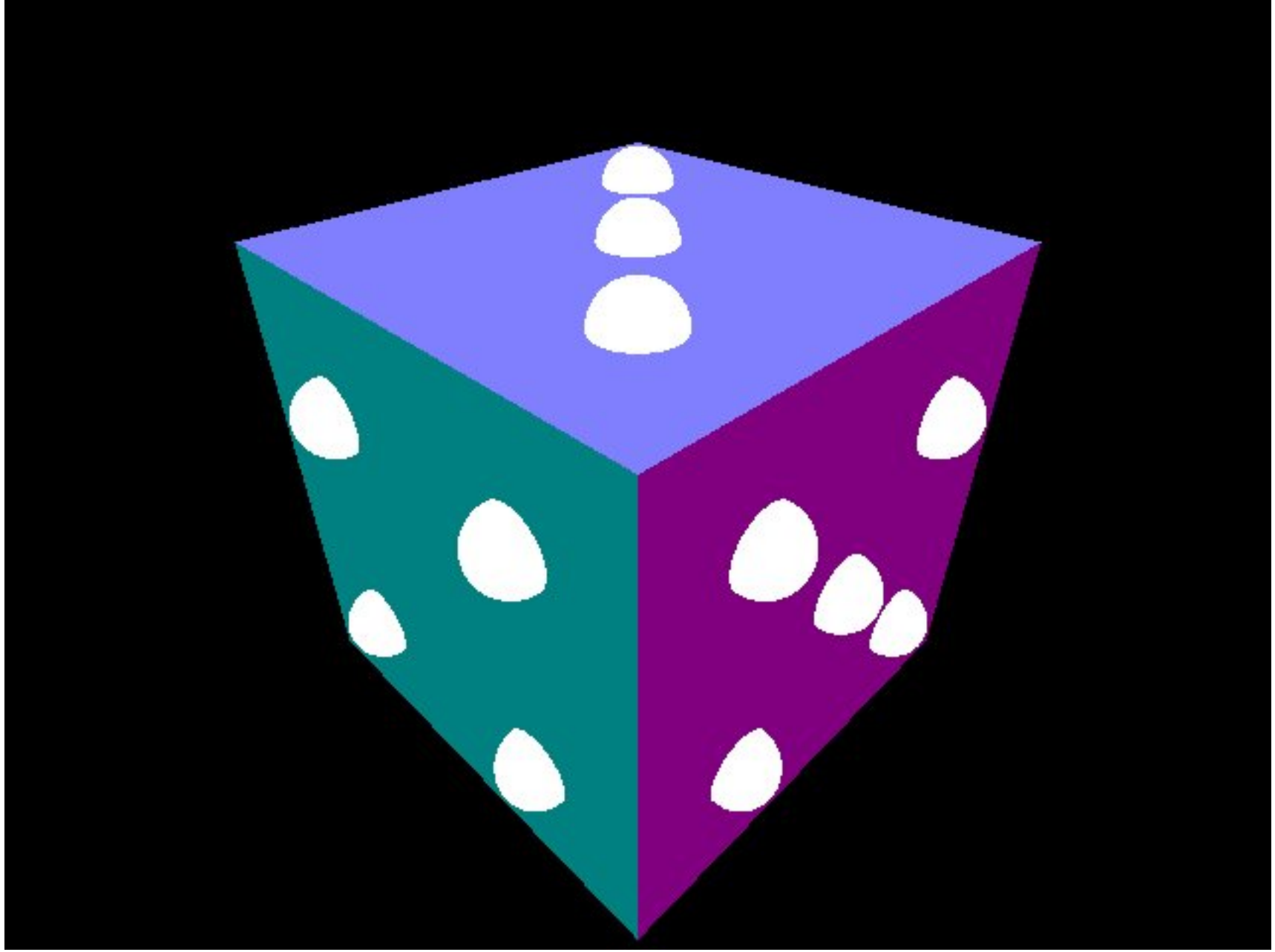
Our result with ambient 1 0 0
Camera4 Scene1



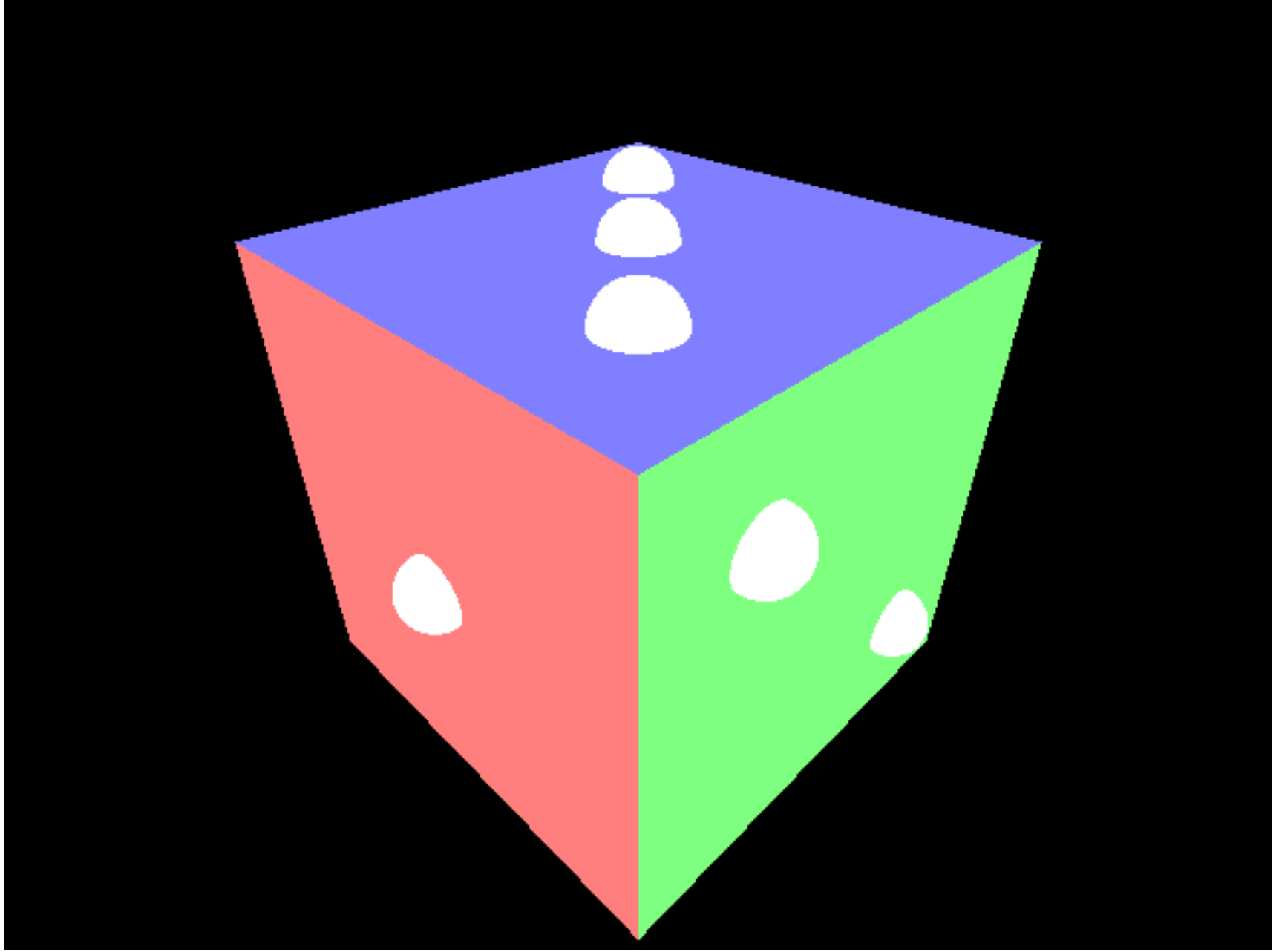
Given Solution
Camera4 Scene1



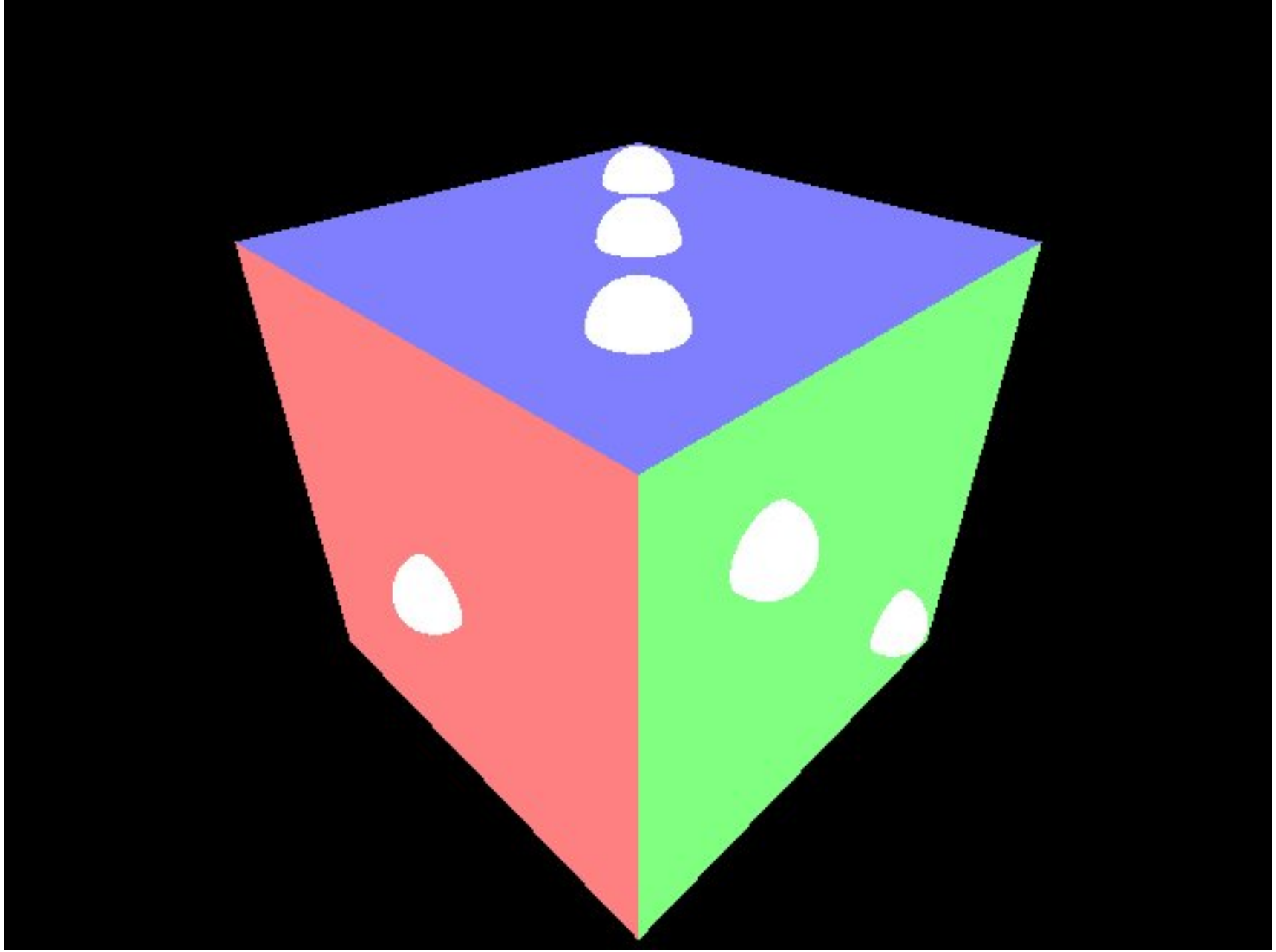
Our result
Camera1 Scene2



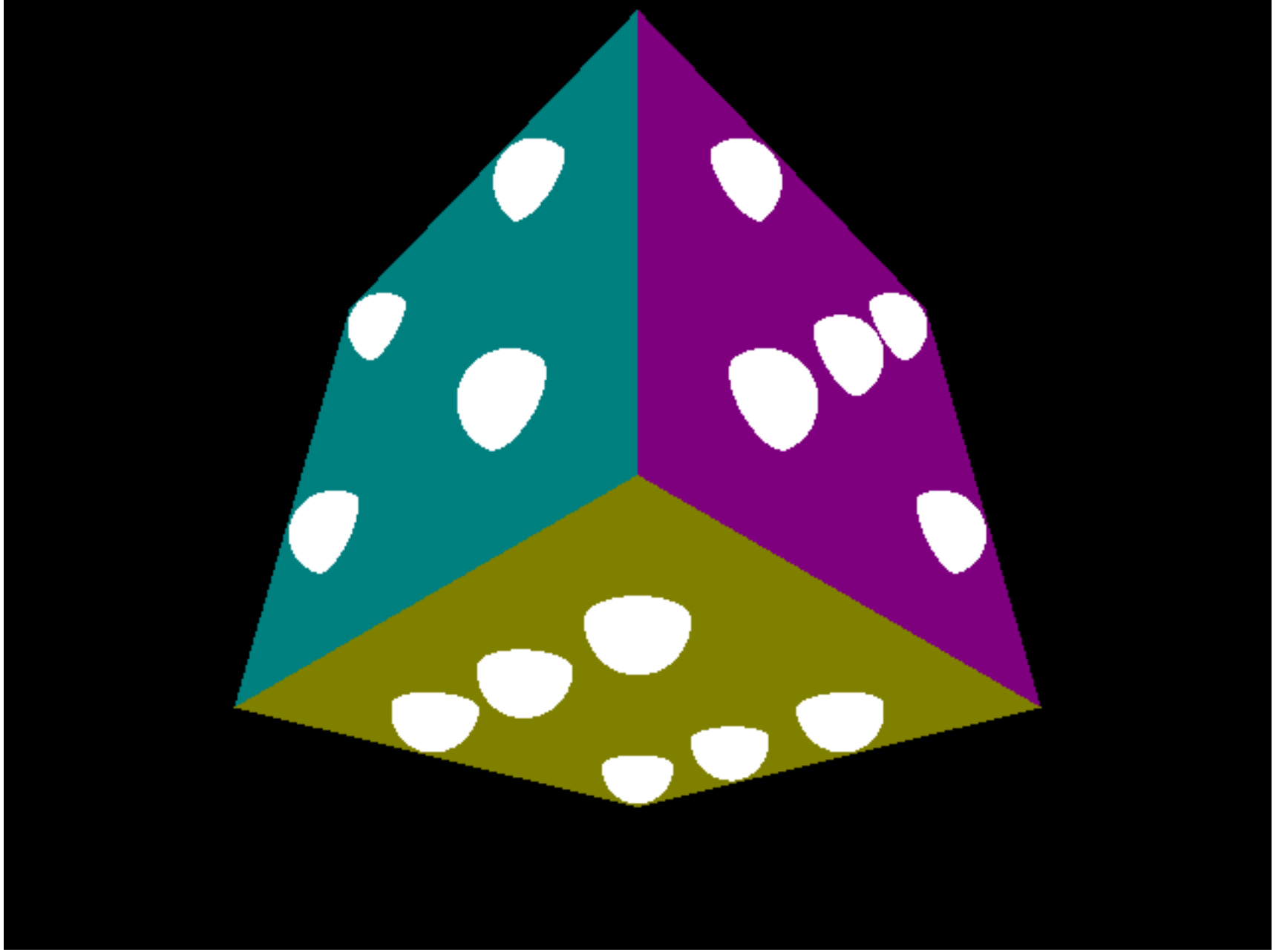
Given Solution
Camera1 Scene2



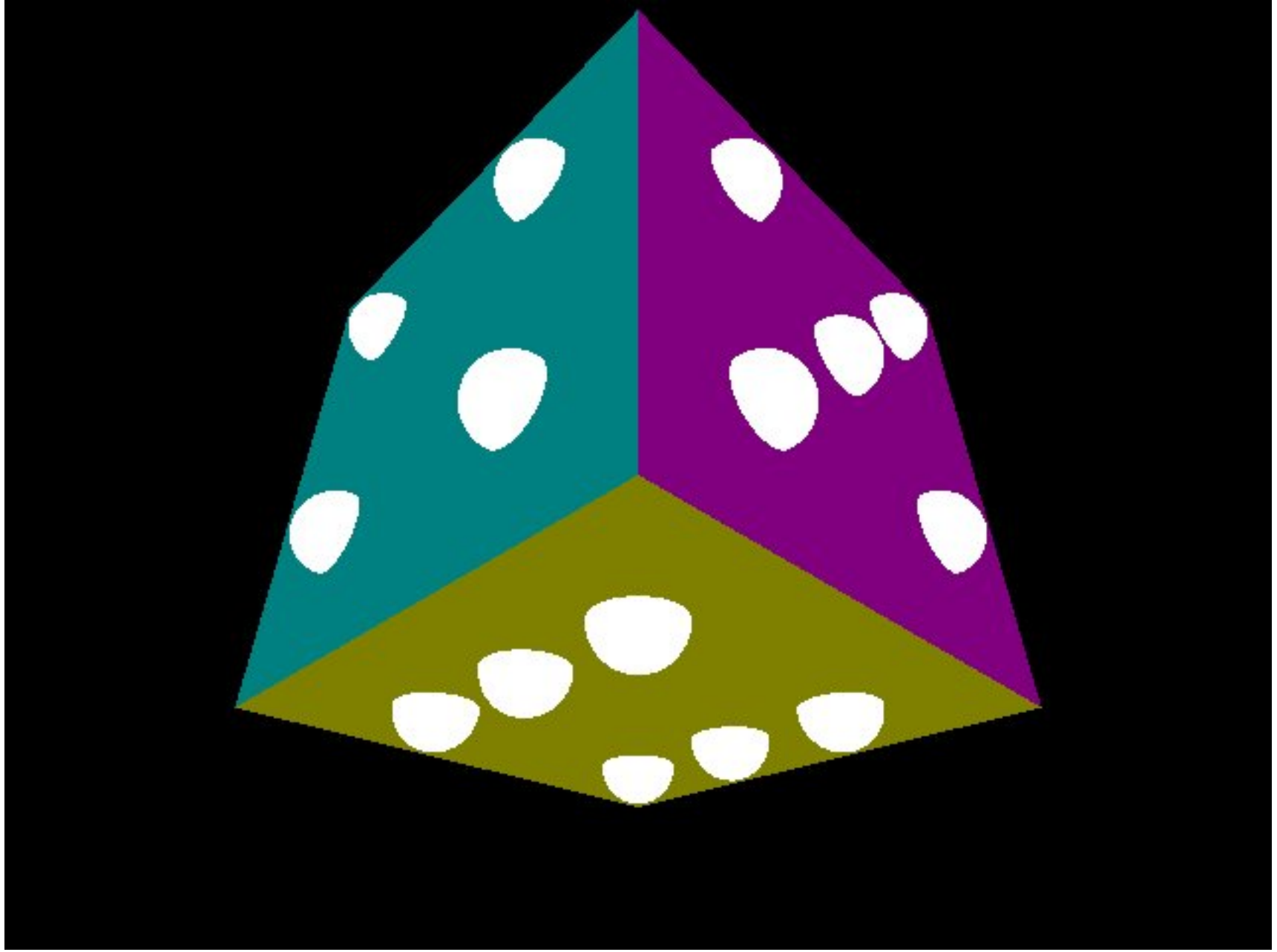
Our result
Camera2 Scene2



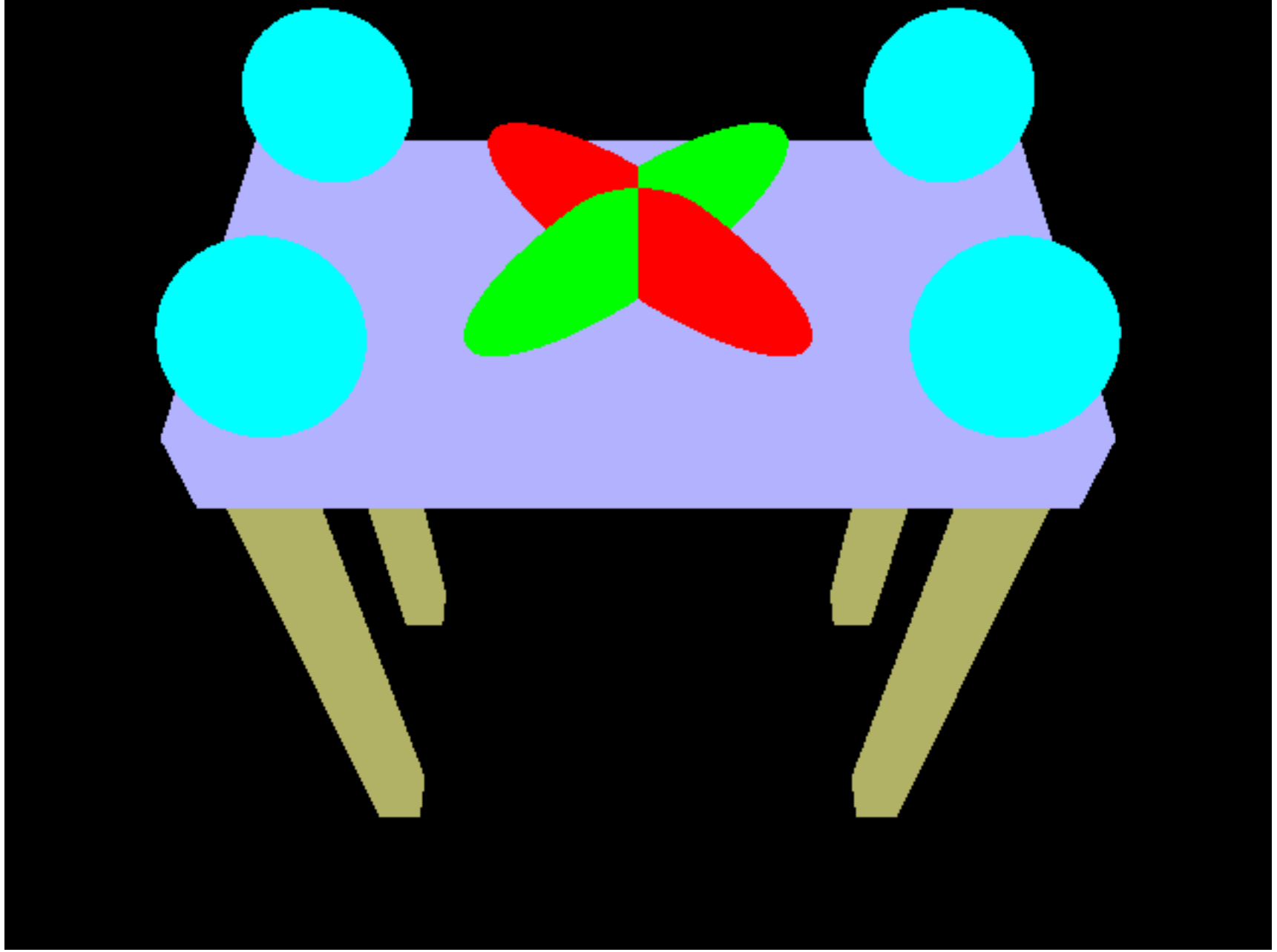
Given Solution
Camera2 Scene2



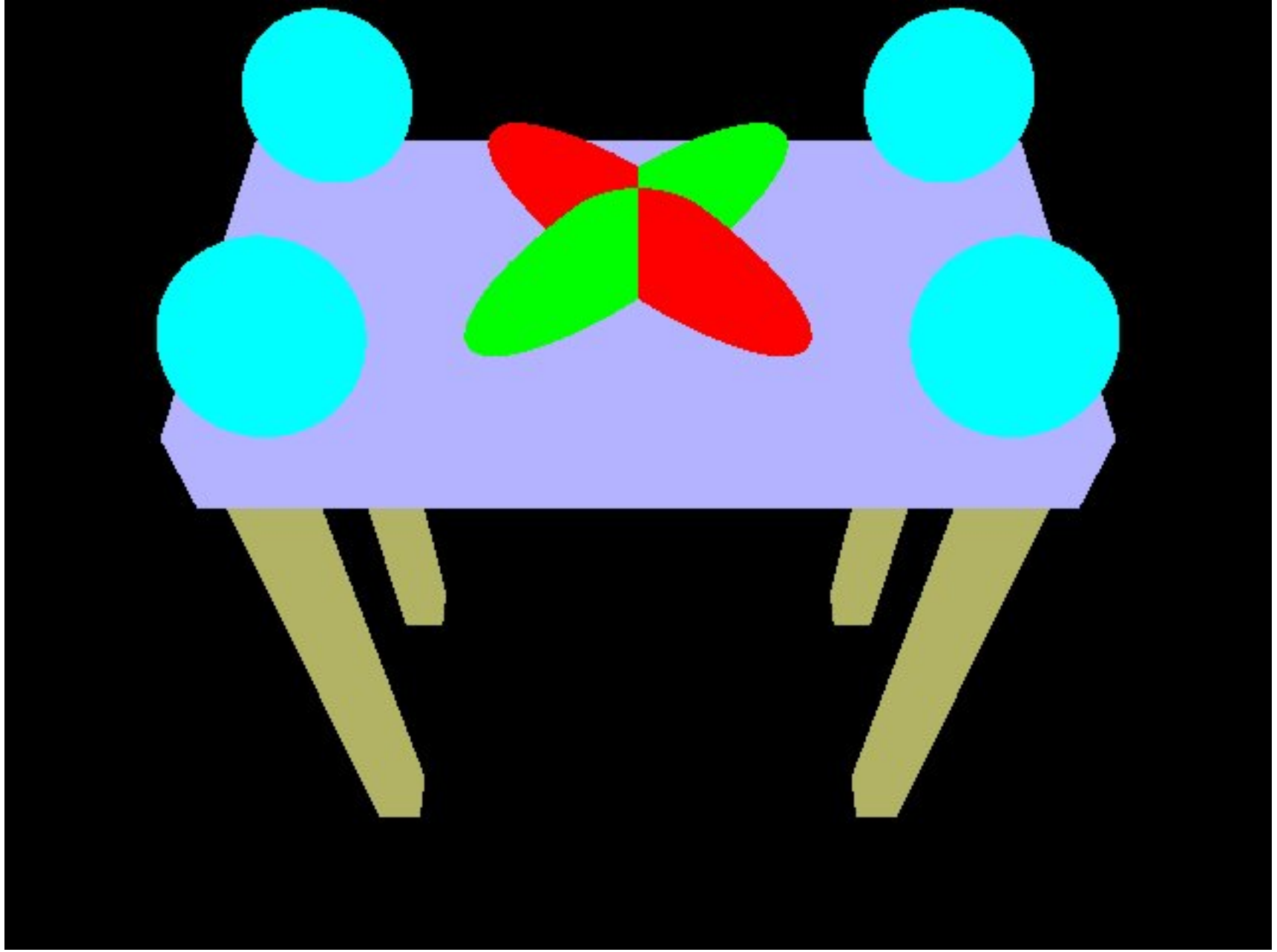
Our result
Camera3 Scene2



Given Solution
Camera3 Scene2



Our result
Scene3



Given Solution
Scene3