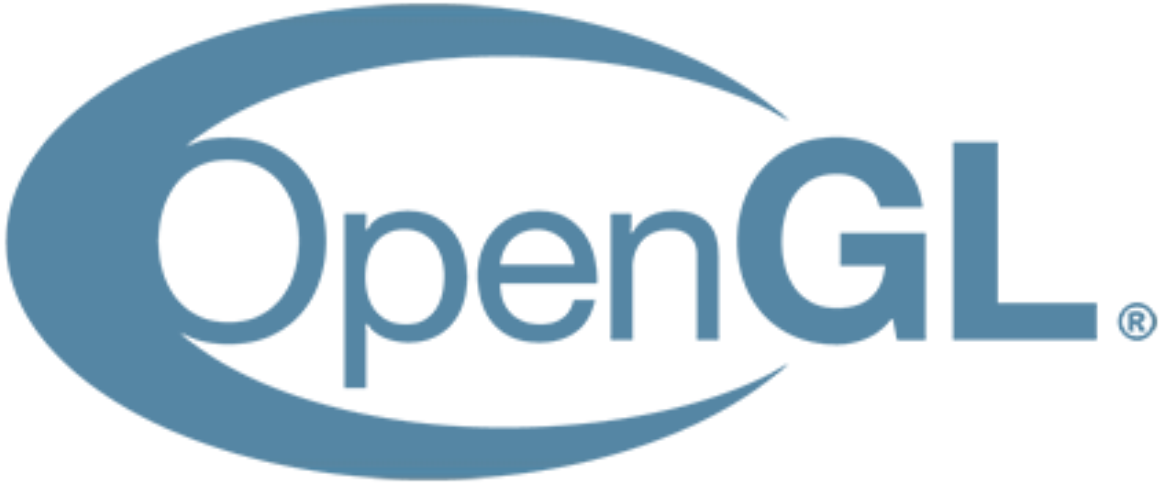
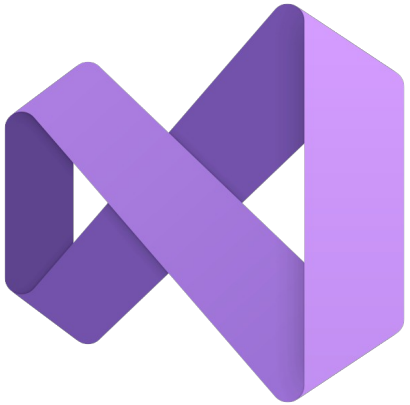
INTRODUCTION TO 3D GRAPHIC PROGRAMMING WITH C++ AND OPENGL

Šimon Potočňák

WHAT IS OPENGL

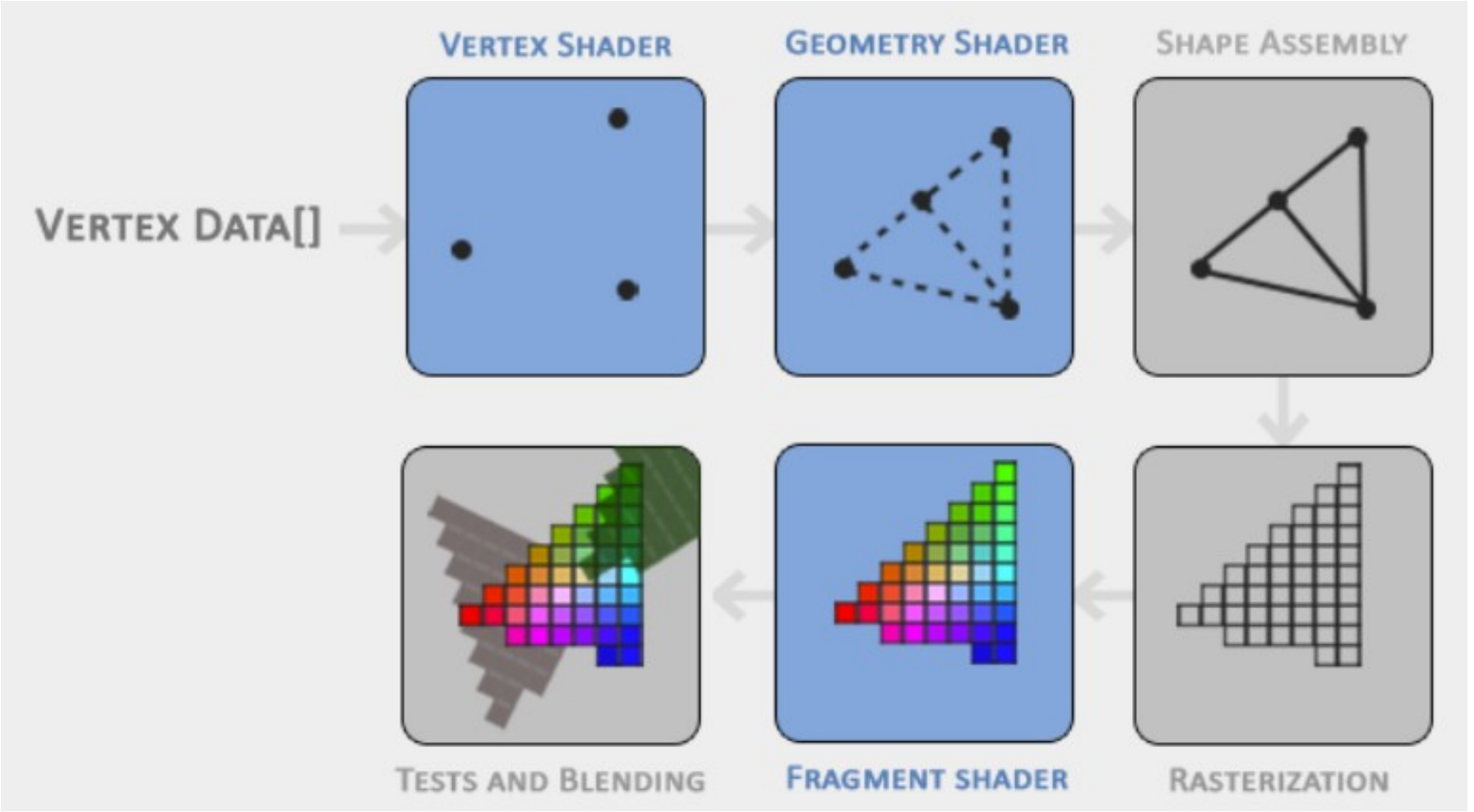




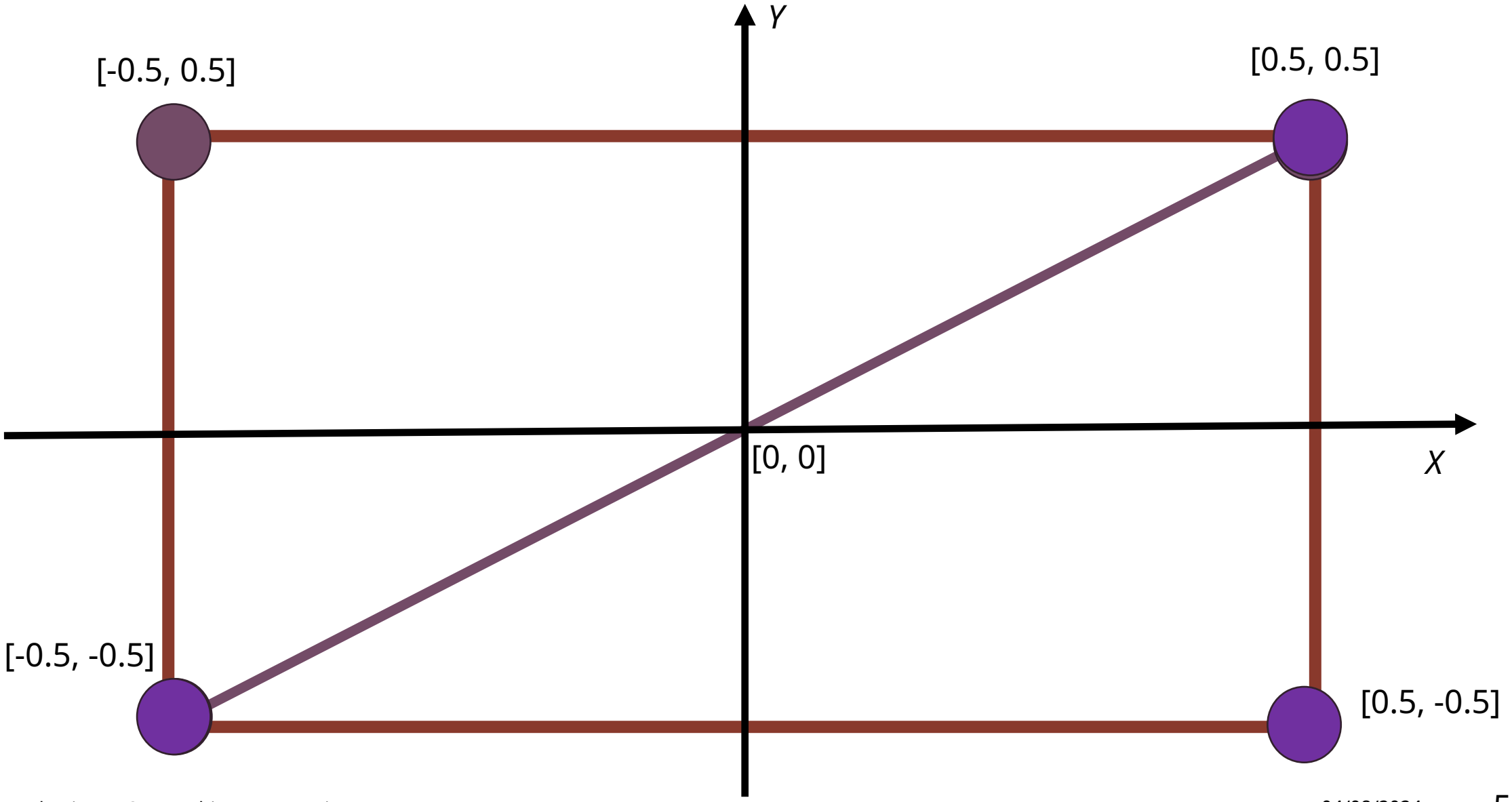
KHRONOS®
GROUP

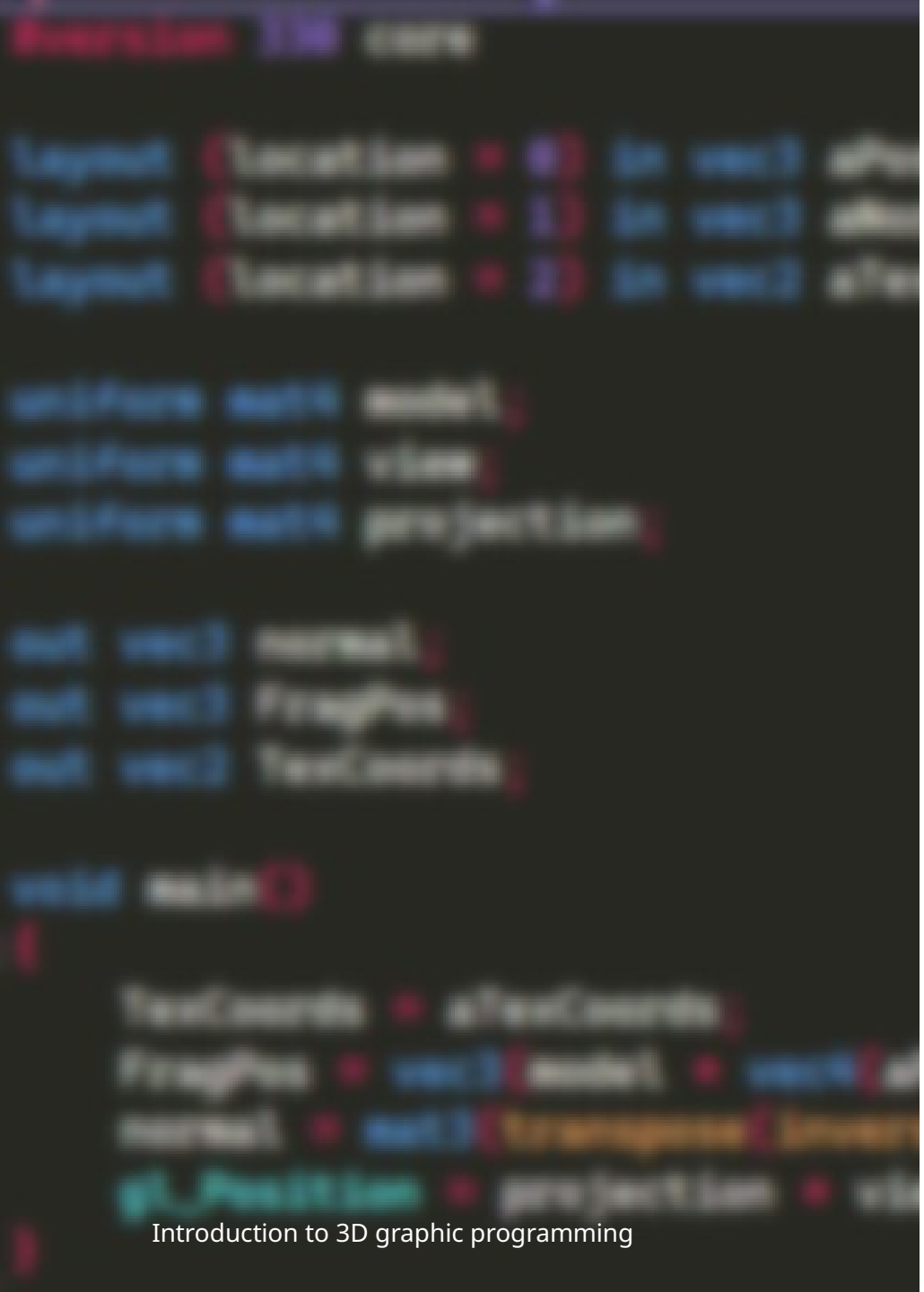
HOW THINGS ARE DROWN

SHADERS


VERTEX DATA[]
VERTEX SHADER
GEOMETRY SHADER
SHAPE ASSEMBLY
RASTERIZATION
FRAGMENT SHADER
TESTS AND BLENDING

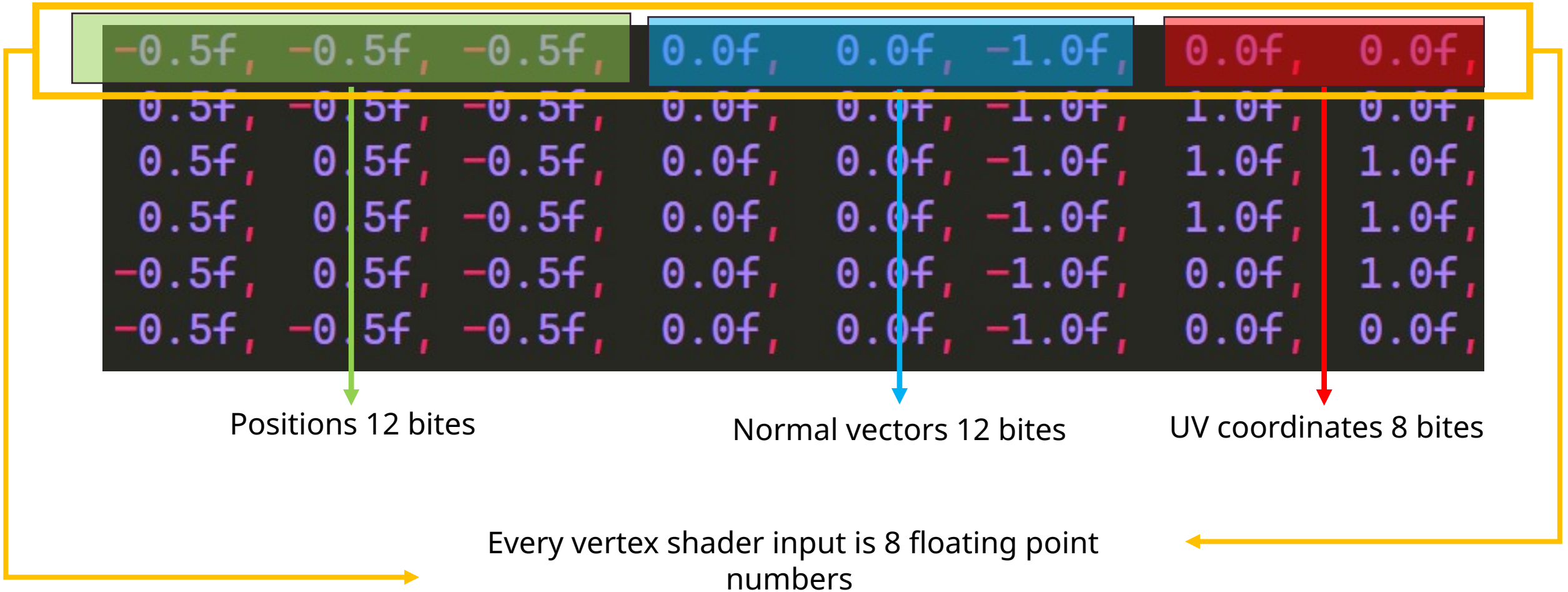
VERTEX SHADER





LET'S WRITE ONE OURSELVES

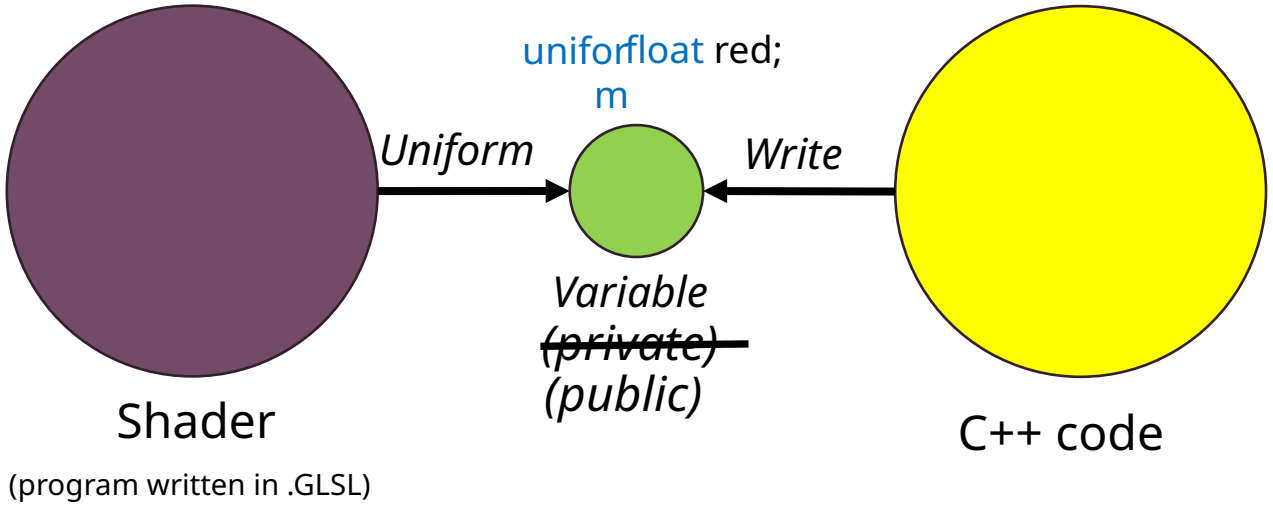
UNDERSTANDING VERTEX DATA



Every vertex shader input is 8 floating point numbers
32 bites

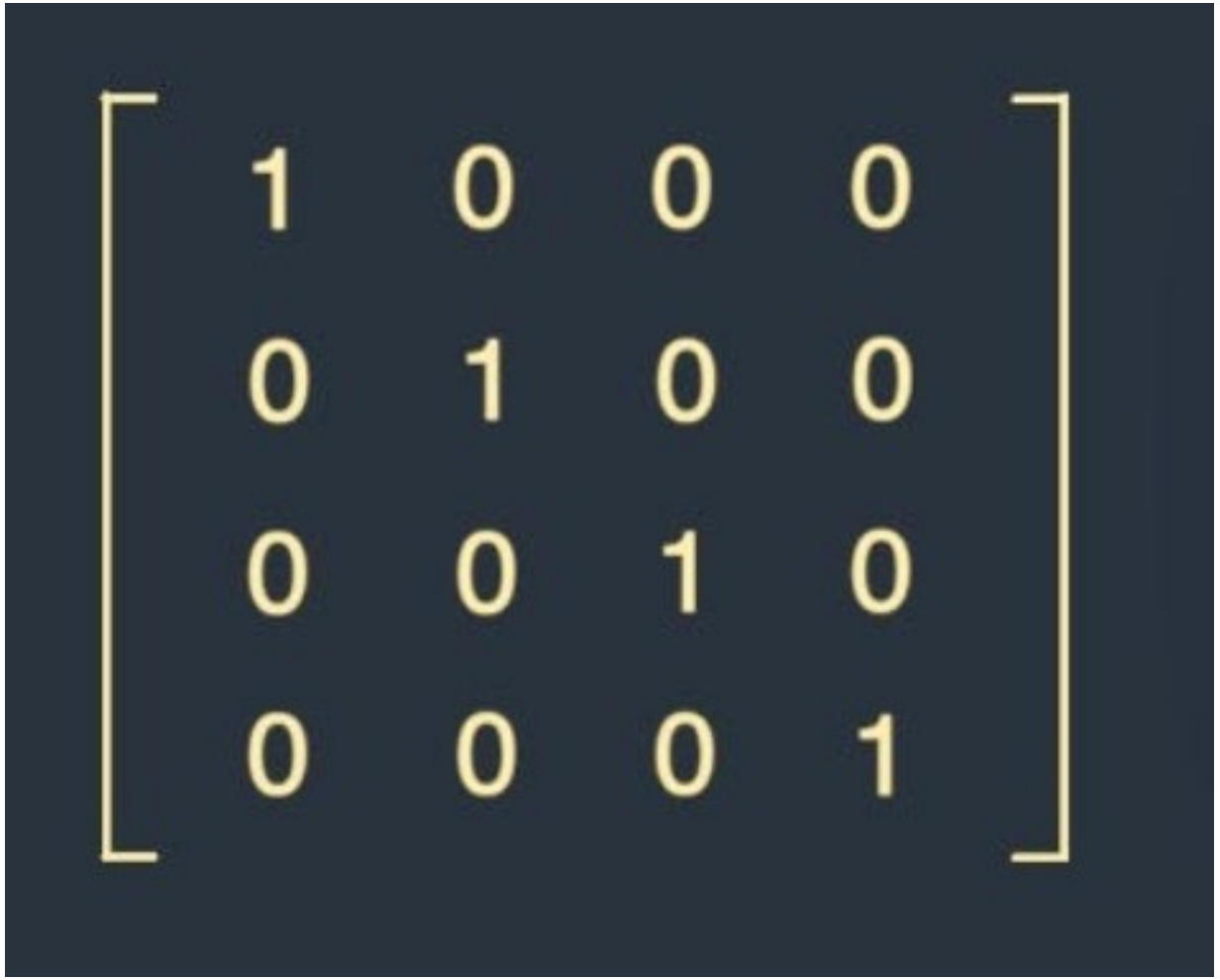
FRAGMENT SHADER

UNIFORMS

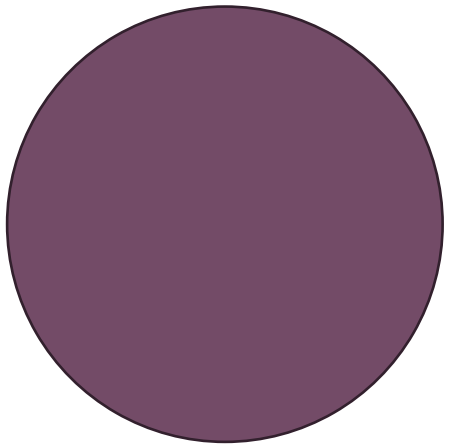

uniforfloat red;
m
Uniform
Write
Variable
~~(private)~~
(public)
Shader
(program written in .GLSL)
C++ code

3D SPACE

MATRICES



MOVING POINTS



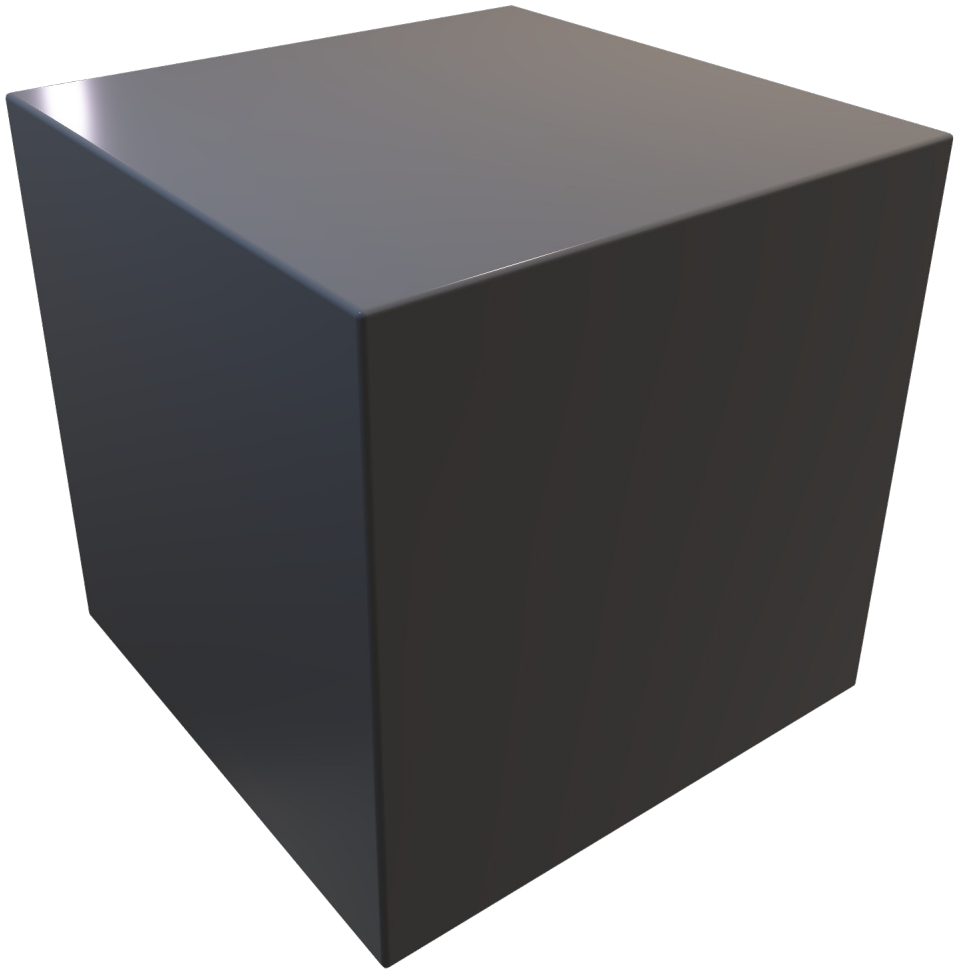
Point in space
vec4(vec3(0.0,0.0,0.0))

Matrix that will move point to the right

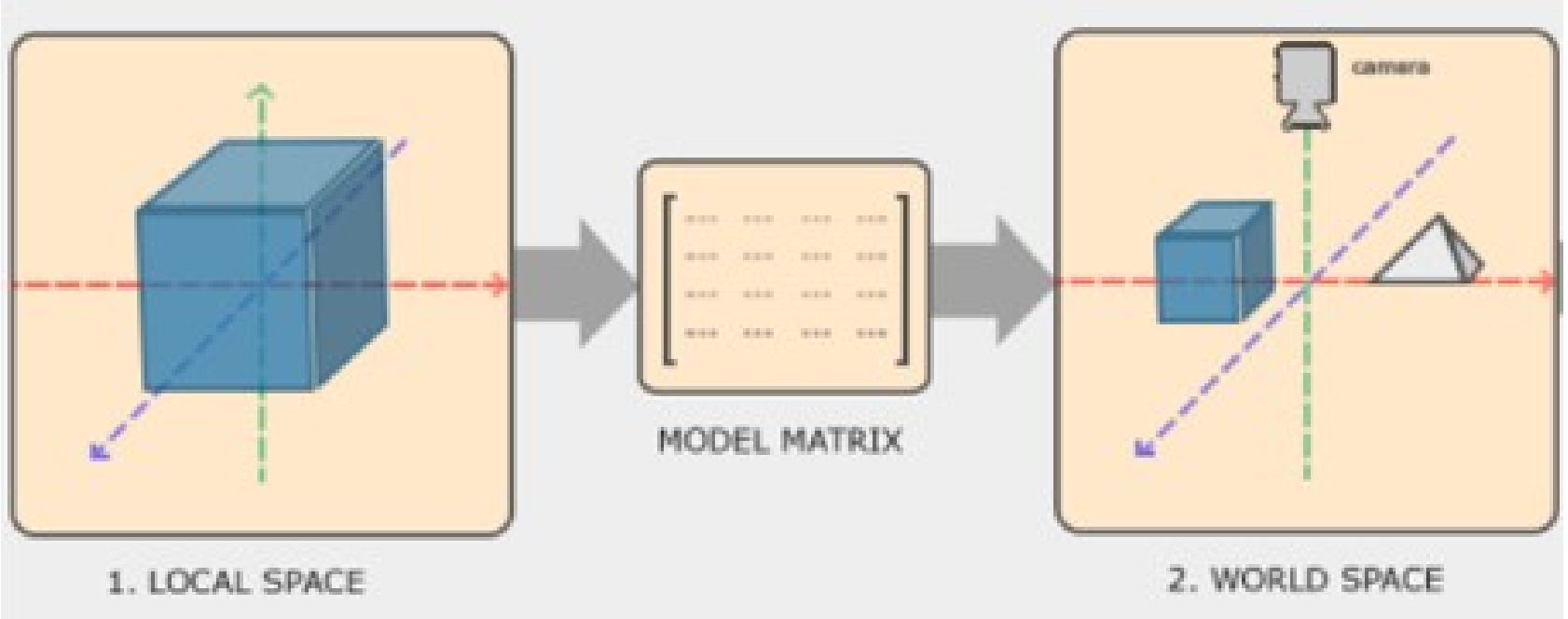
Point in space
vec4(vec3(1.0,0.0,0.0))

OPENGL MATHEMATIC LIBRARY

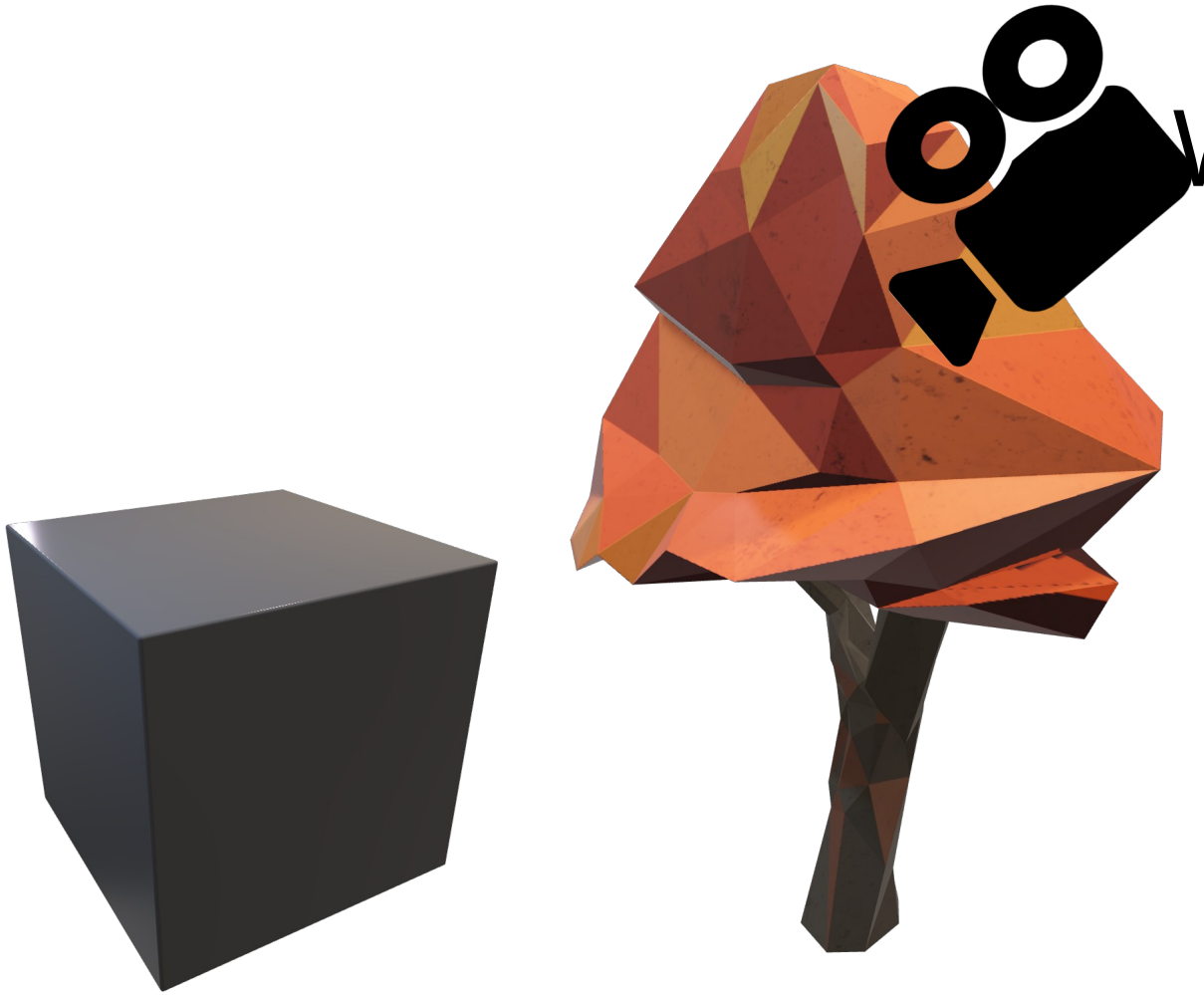
LOCAL SPACE



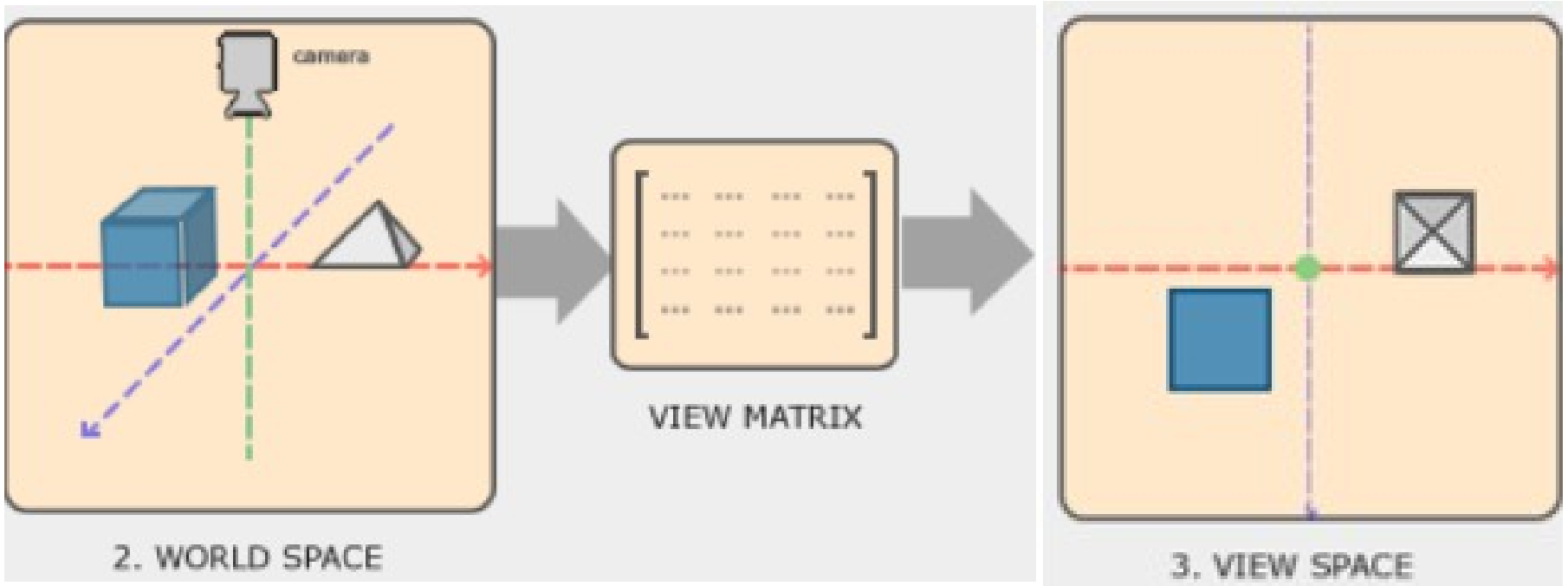
WORLD SPACE



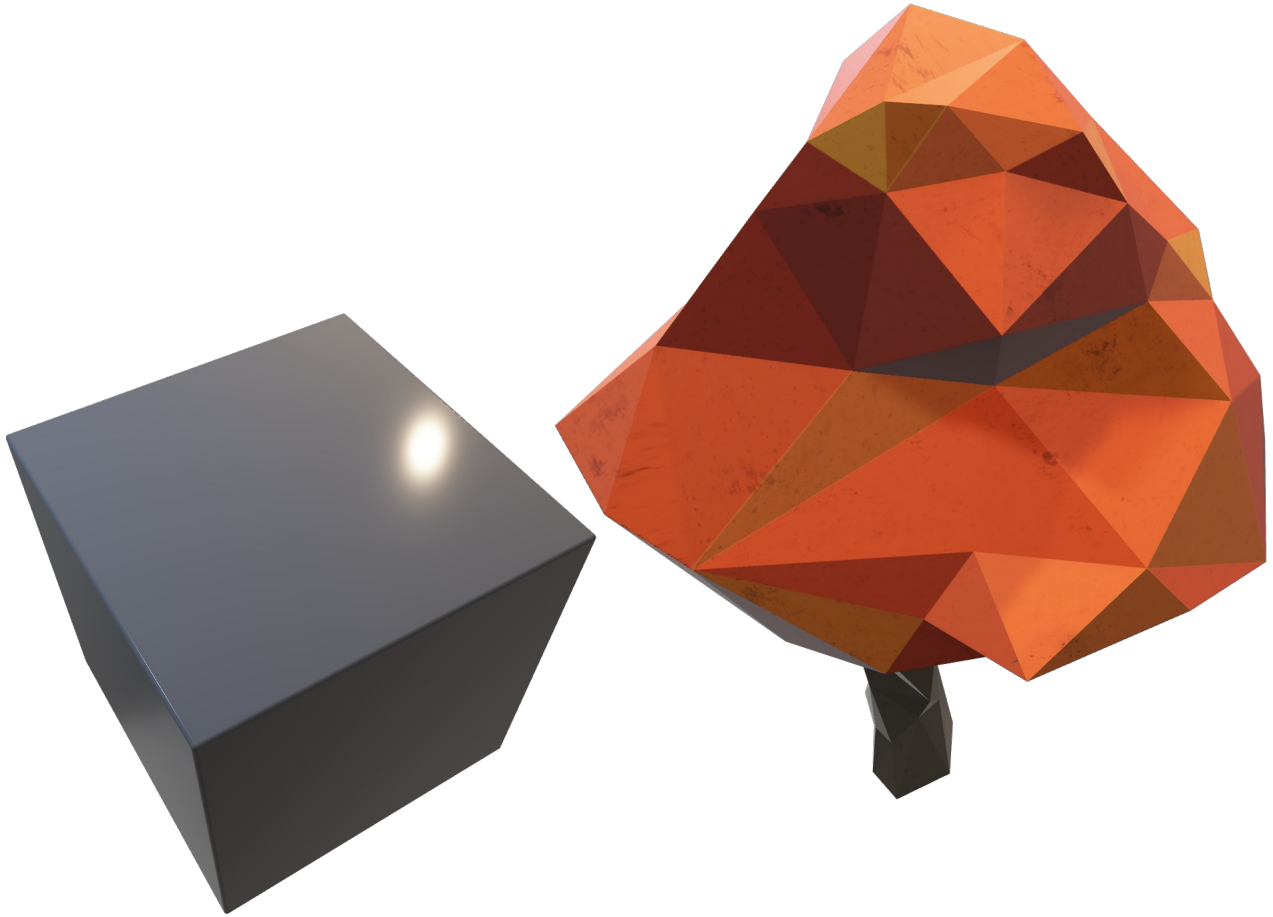
WORLD SPACE



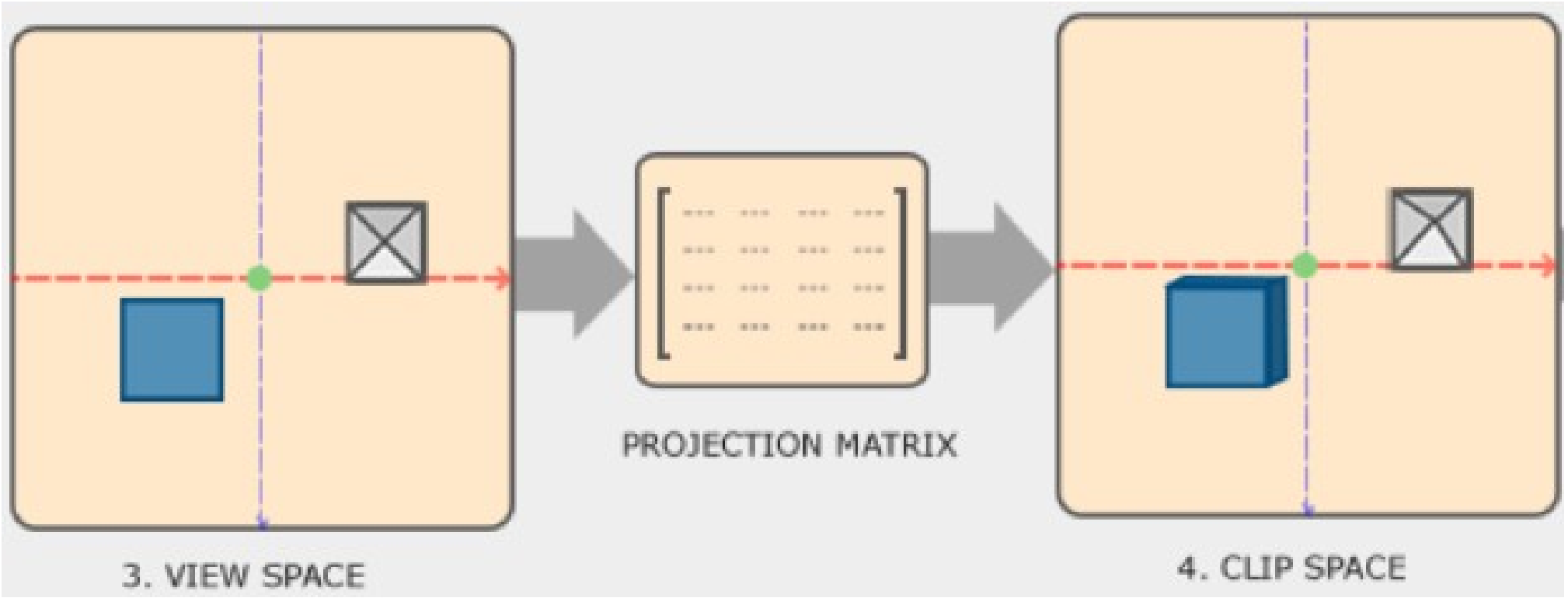
VIEW SPACE (ILLUSION OF THE CAMERA)



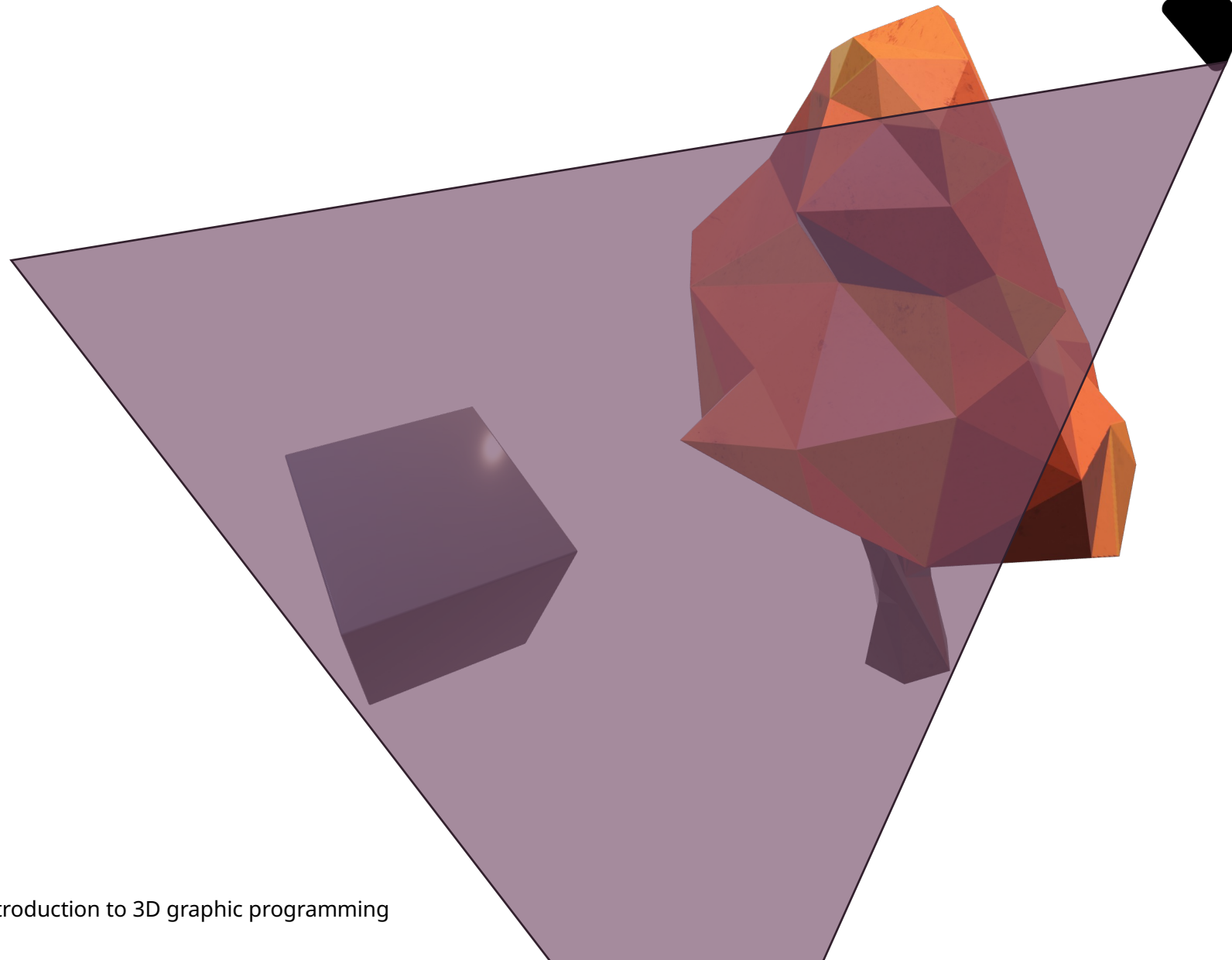
VIEW SPACE



PROJECTION MATRIX

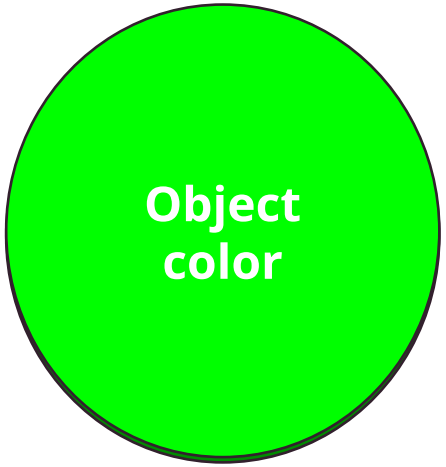


CLIP SPACE

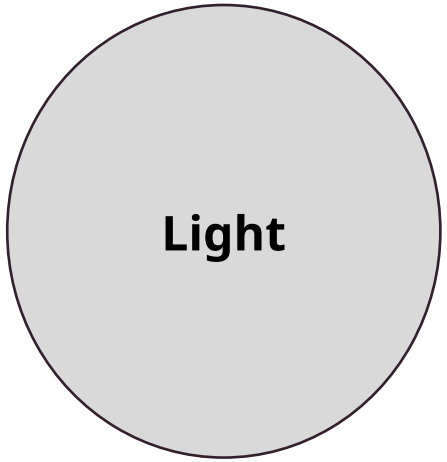


LIGHTNING

OBJECT COLOR AND LIGHT COLOR

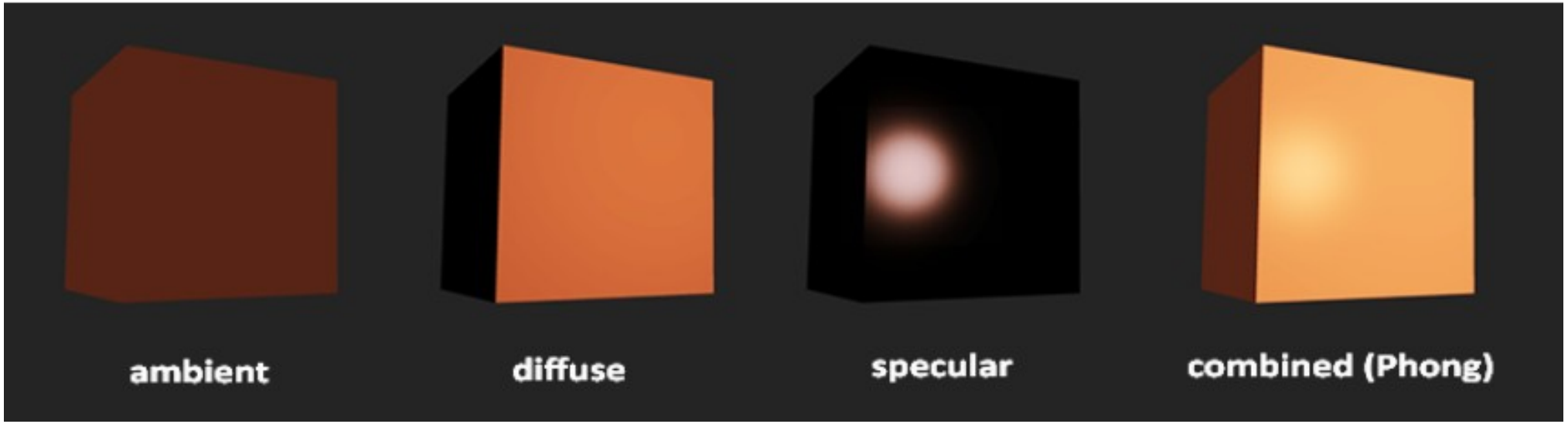


(1.0, 0.5, 0.31)



(0.7, 0.7, 0.7)

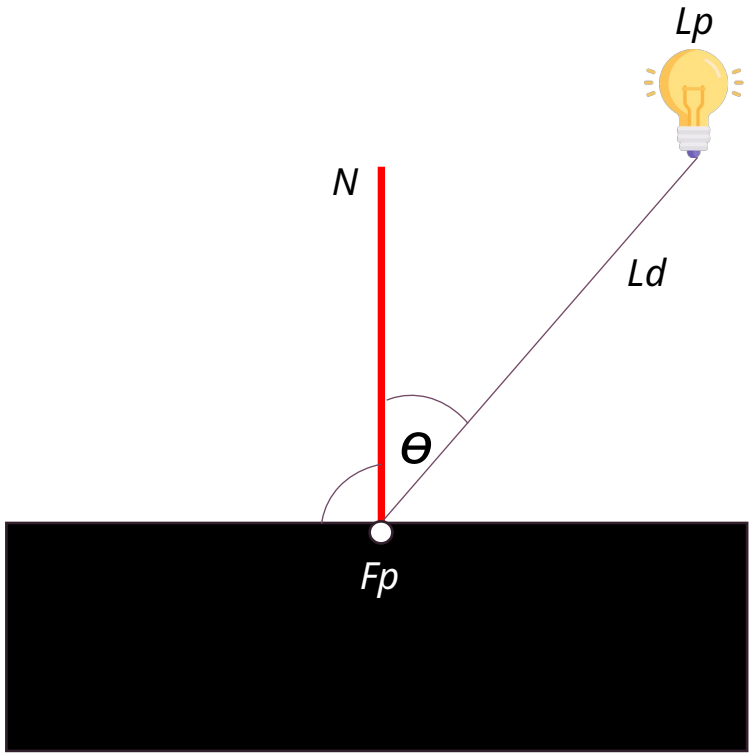
PHONG LIGHTNING MODEL



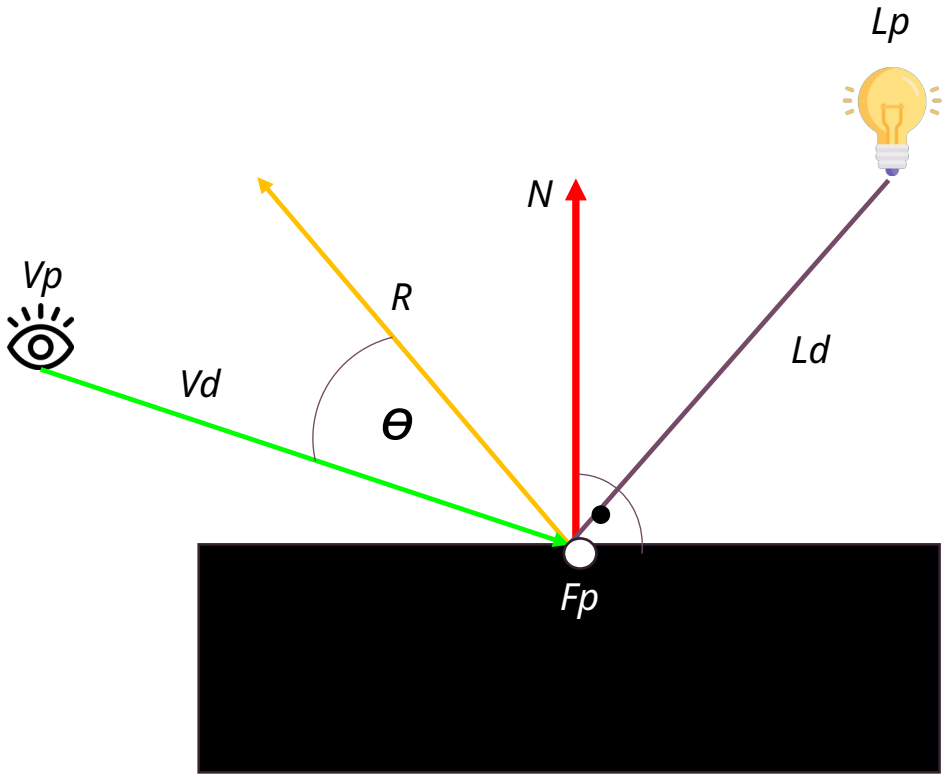
AMBIENT

$$AmbientStrength = lightColor * K$$
$$FragmentColor = AmbientStrenght * LightColor * Object\,color$$

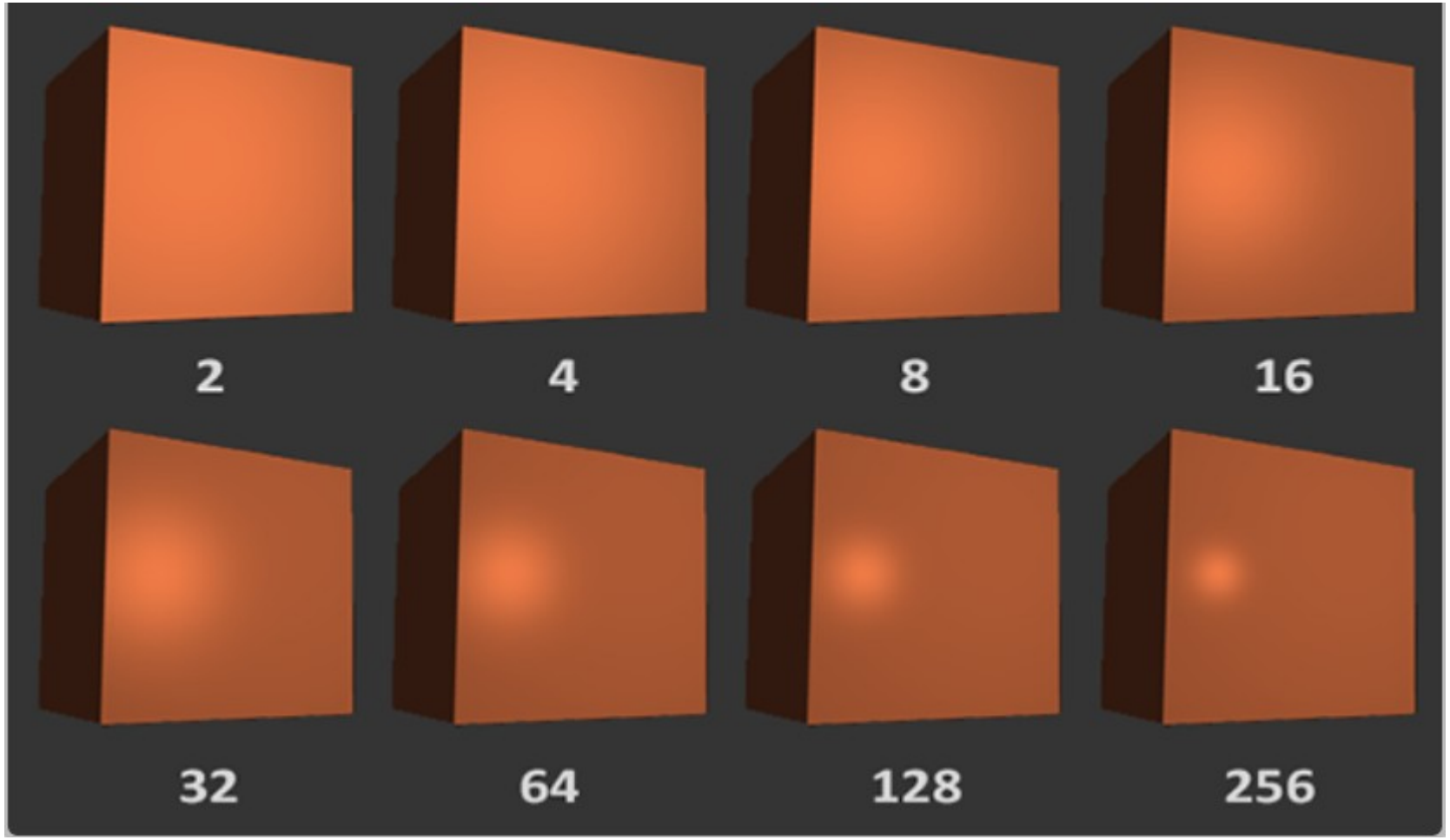
DIFFUSE LIGHT



SPECULAR LIGHT

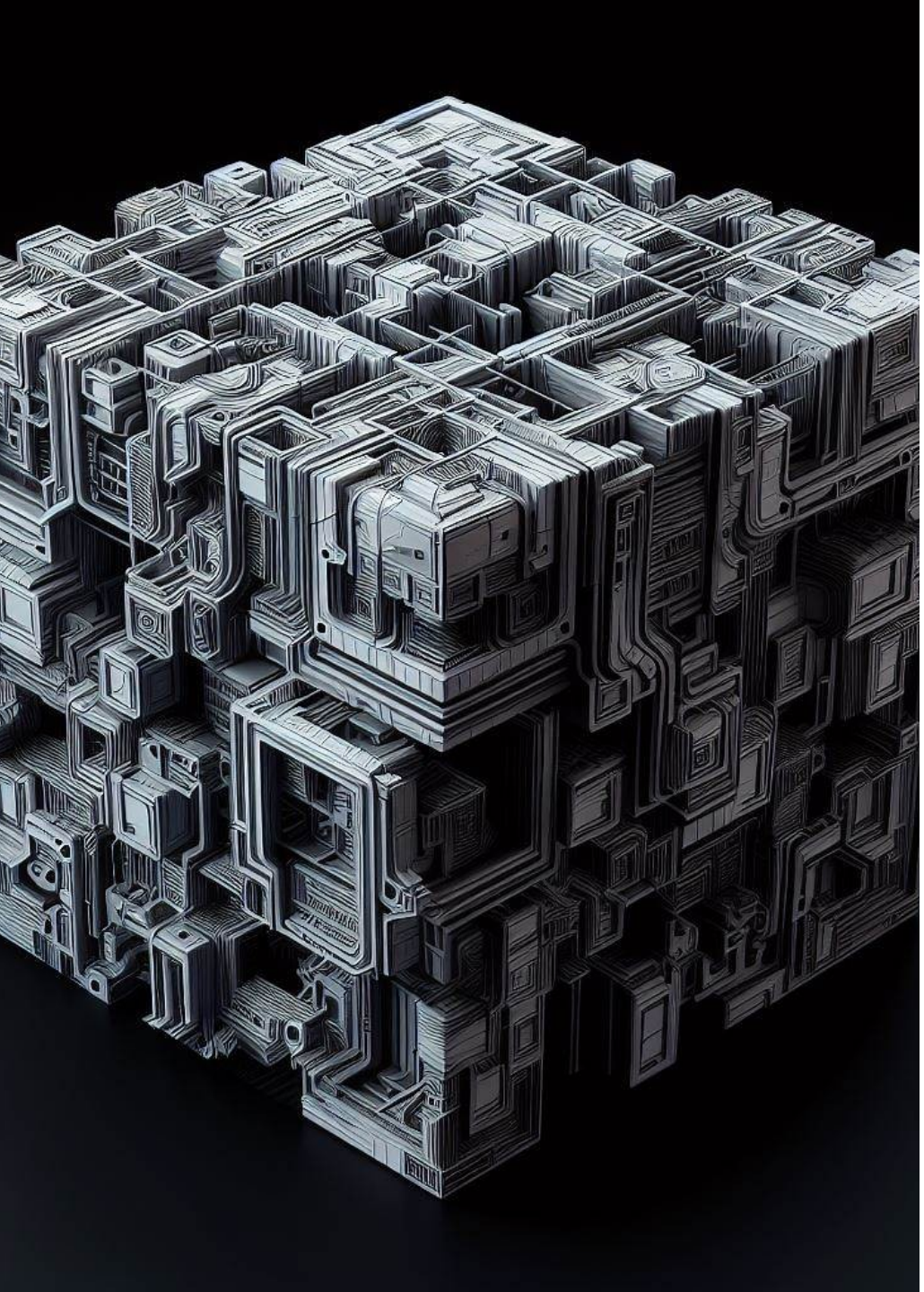


WHY 16 ?



RESULT

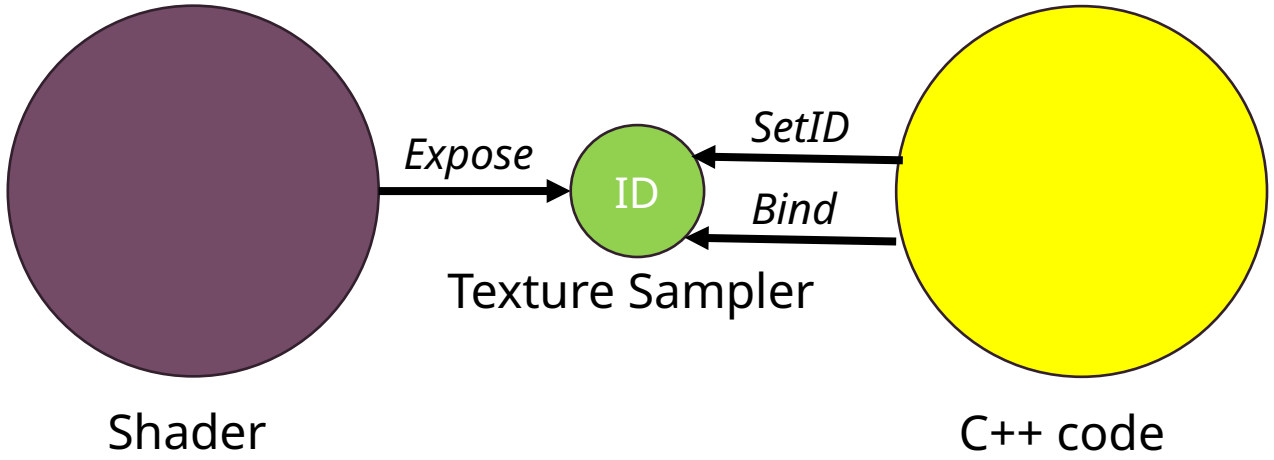
FinalColor = Ambient + Diffuse + Specular



LET’S CODE THIS

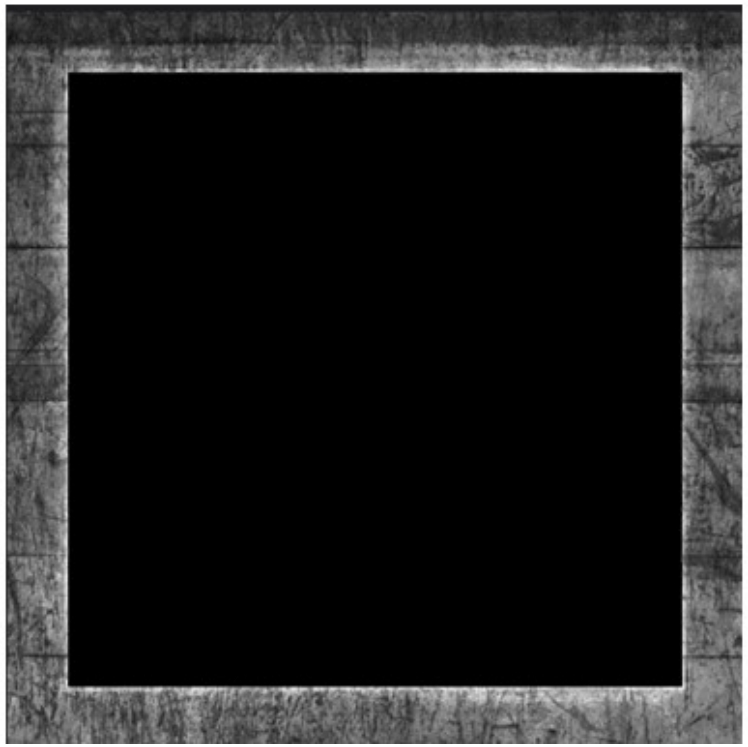
TEXTURES

TEXTURES



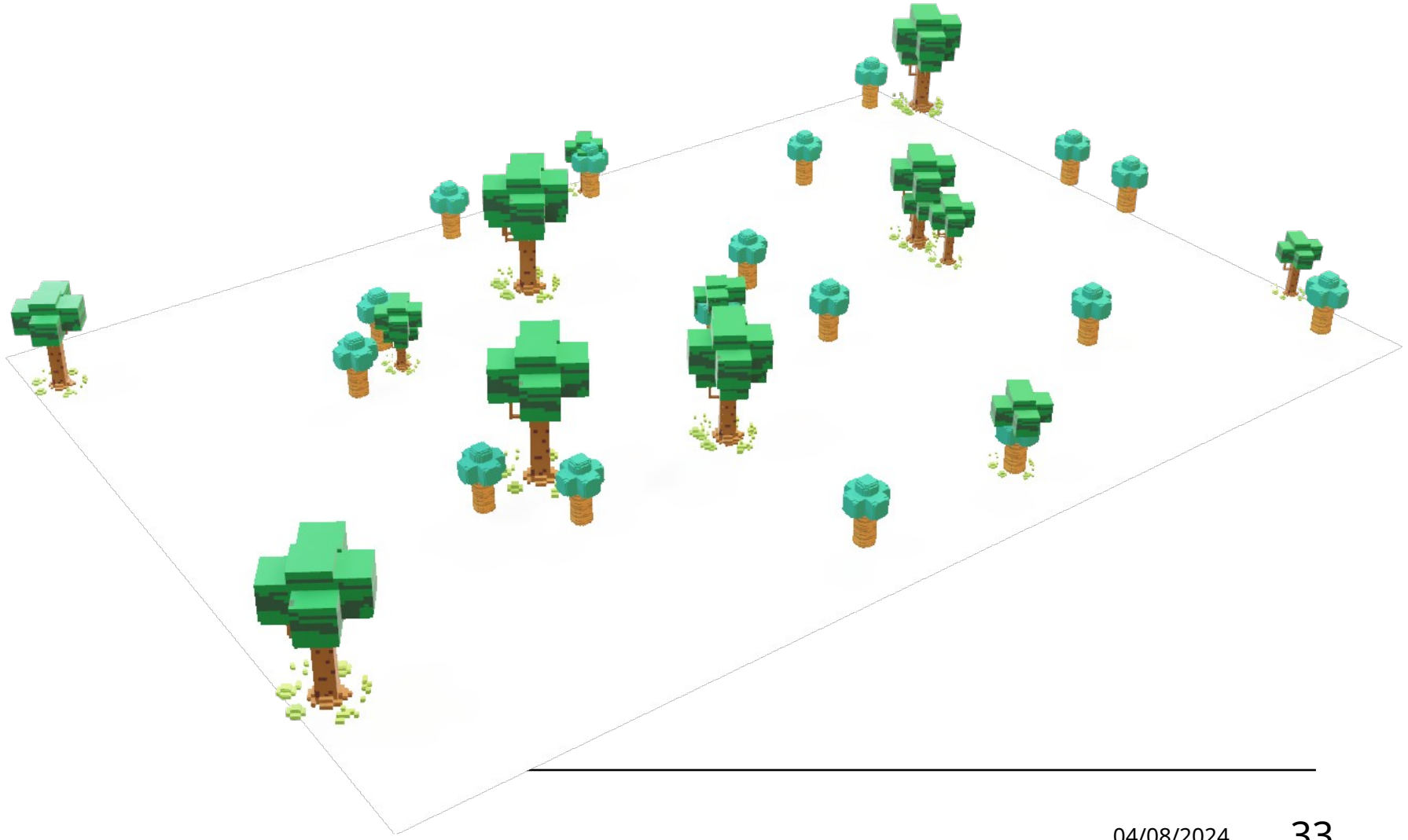
MATERIALS

Diffuse (does not have shiny spot)



Specular (has shiny spot)

MODEL LOADING



THANK YOU

Šimon Potočňák