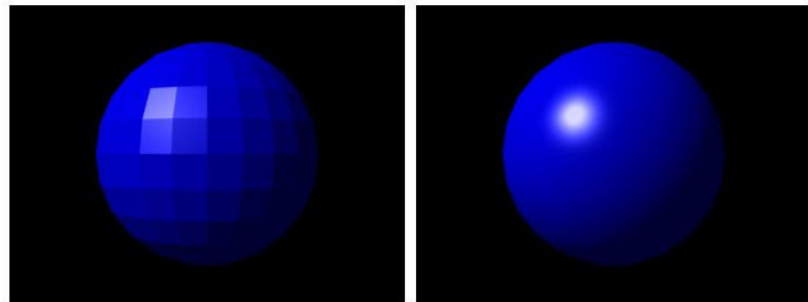


CSE 333/533: Computer Graphics

Lab **1** | Introduction to OpenGL Shaders

Introduction to OpenGL Shaders

- Shader is a user-defined program designed to run on some stage of a graphics processor.
- Shaders provide the code for certain programmable stages of the rendering pipeline.



Vertex Shader

- Vertex shader is responsible for deciding the spatial location where rendering takes place. It does this with the help of vertex attributes.

```
in vec3 vVertex;

uniform mat4 vModel;
uniform mat4 mView;
uniform mat4 vProjection;
uniform vec3 vColor;

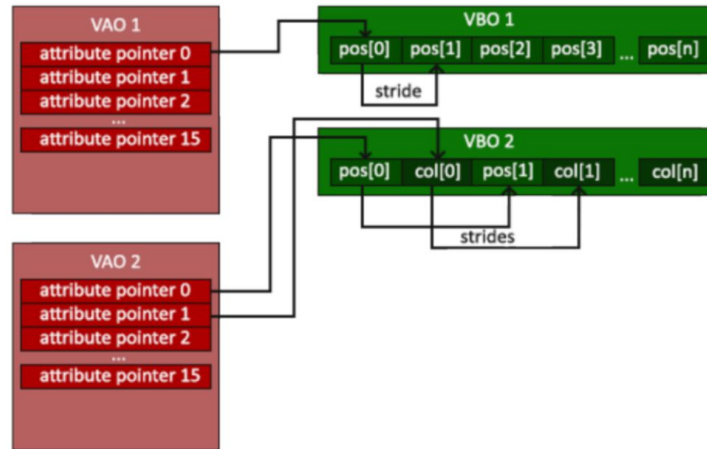
out vec3 fColor;
void main() {
    gl_Position = vProjection * mView * vModel * vec4(vVertex, 1.0);
    fColor = vColor;
}
```

Vertex Array Object

- Vertex Array Object (VAO) is an OpenGL Object that stores **all of the state needed to supply vertex data.**
- It stores the format of the vertex data as well as the **Buffer Objects** providing the vertex data arrays.

Vertex Buffer Object

- Vertex Buffer Object (VBO) is the common term for a normal Buffer Object when it is used as a source for vertex array data.



Fragment Shader

- Fragment shader is responsible for the appearance and colors of the rendered object (triangle in our case).

```
in vec3 fColor;  
out vec4 outColor;  
  
void main(void) {  
    outColor = vec4(fColor, 1.0);  
}
```

Exercise

- Use the shaders. Define all the vertices of the cuboid.
- Define the order in which the vertices must be used to form all the triangles required to complete the cuboid.

▼ Information
16.697 ms/frame (59.9 FPS)

