# openGL-1

Daniel Haitink and Robin Entjes

# openGL

```
m_funcs->glGenVertexArrays(1, &VAO);
m_funcs->glBindVertexArray(VAO);
```

Generate VAO and bind it to use it.

```
m_funcs->glGenBuffers(1, &vertices);
m_funcs->glBindBuffer(GL_ARRAY_BUFFER, vertices);
```

Generate buffers and bind them

# openGL

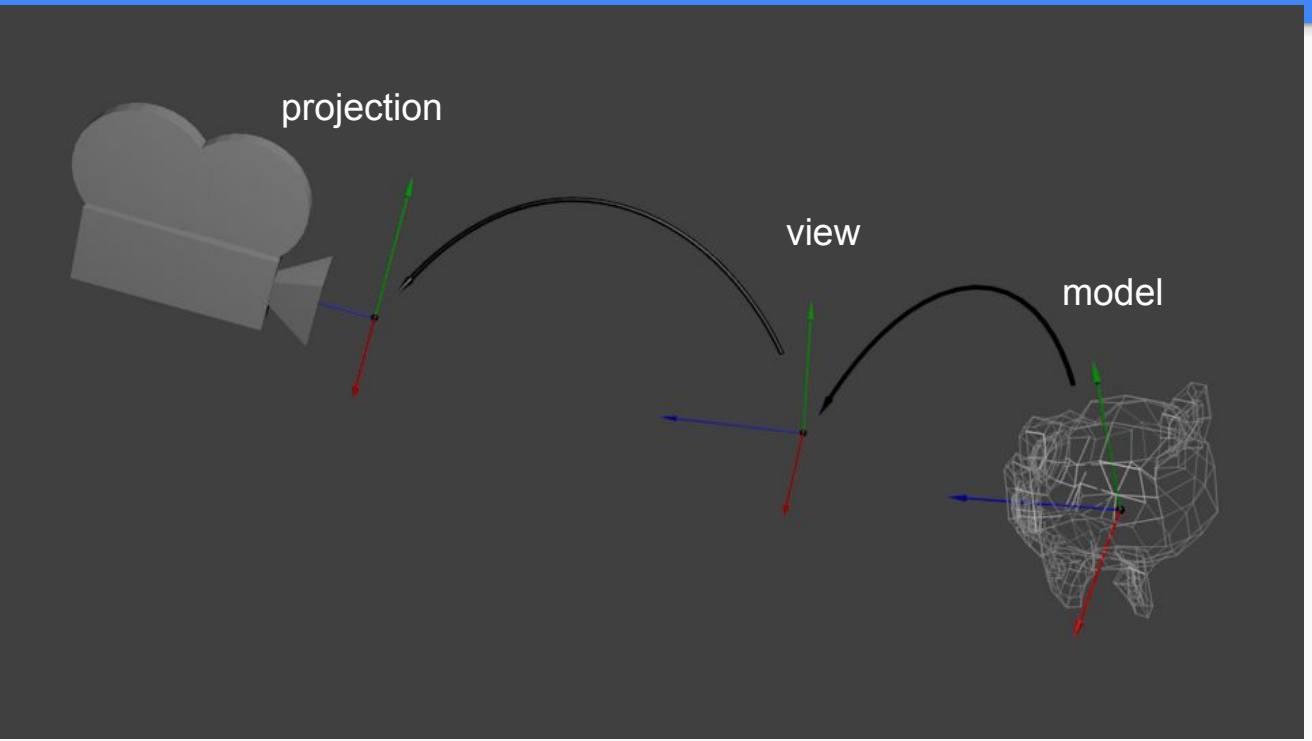
```
m_funcs->glBufferData(GL_ARRAY_BUFFER, sizeof(QVector3D) *
objectVectors.length(),objectVectors.data(), GL_STATIC_DRAW);
```

Fill buffer with data.

```
 m_funcs->glEnableVertexAttribArray(0);
 m_funcs->glVertexAttribPointer(0,3, GL_FLOAT, GL_FALSE, 0,0);
```

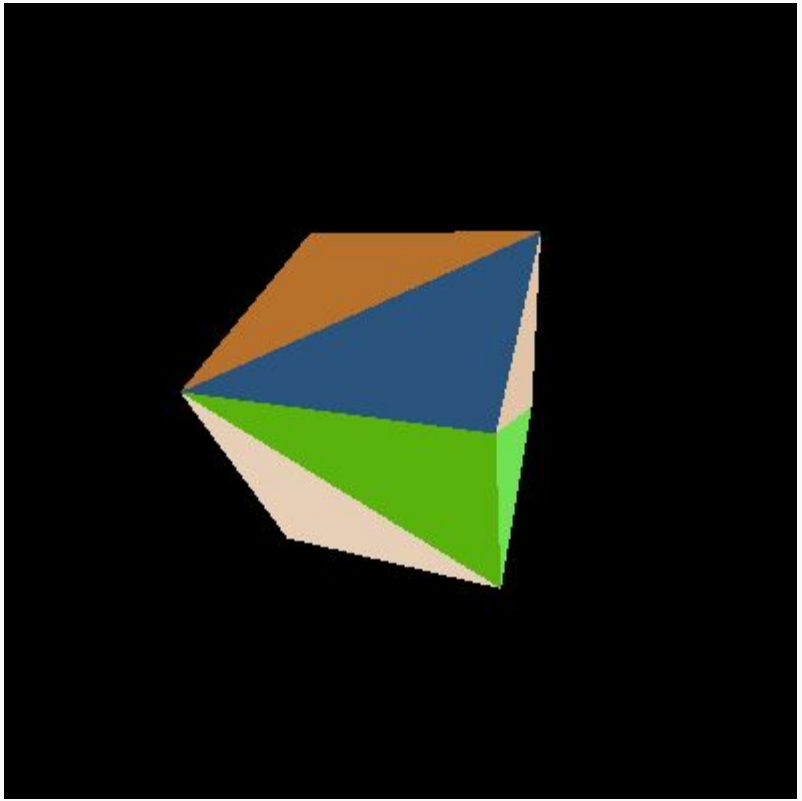
Set to attribute pointer

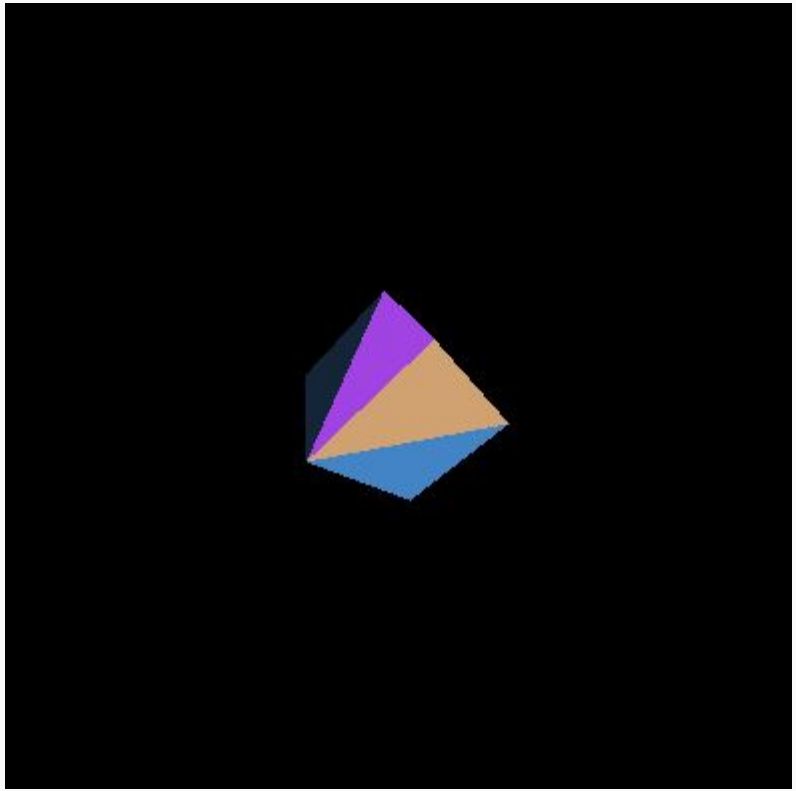
# MVP




in glsl:

gl_Position = projection * view * model * vec4(posAttr, 1);





our cube in openGL

# Spheres

- bind normals instead of colors
- give more uniforms (per ball)
  - color
  - material
  - lightPosition
  - normalMatrix = normal of view*model
  - eye position

# Phong shading

$$I = I_a k_a + I_i(k_d(L \cdot N) + k_s(R \cdot V)^n)$$

Or: S = Sm*max((N•H), 0)$^n$

fColor = vec4(materialColor*I,1)