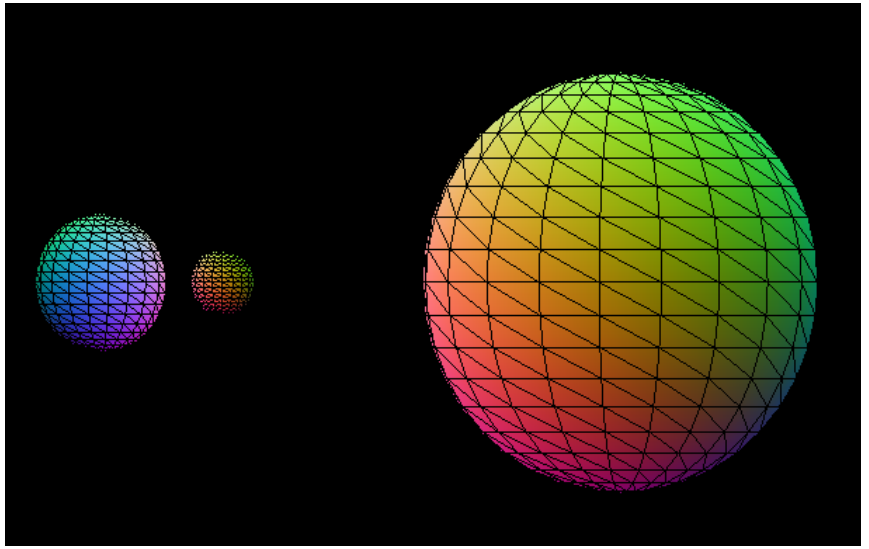
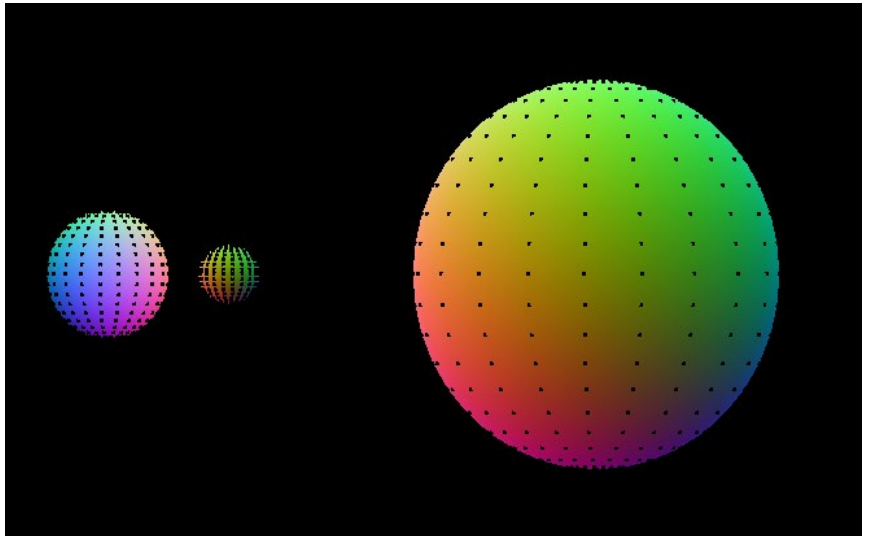
Homework 6

- Complete the solar system with the following additional requirements:
  - Draw the wireframes and the vertices of all the spheres when ‘w’ key and ‘v’ is pressed, respectively.
  - Do not assign an additional buffer object to store vertex positions for wireframes or vertices.
  - Use the following OpenGL functions for drawing a model as a wireframe, a colored object or points:

```
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);

glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);

glPolygonMode(GL_FRONT_AND_BACK, GL_POINT);
```





- How to generate object colors using vertex positions:

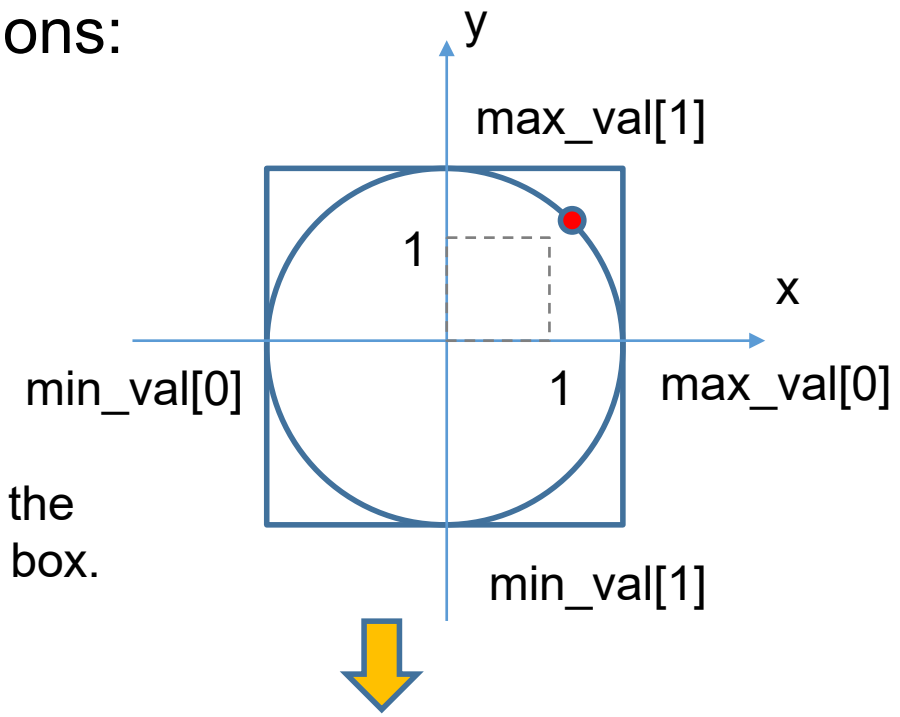
```
void get_color_3d_by_pos(GLvec& c, GLvec& p, GLfloat offset)
{
	GLfloat max_val[3] = { -INFINITY,-INFINITY,-INFINITY };
	GLfloat min_val[3] = { INFINITY, INFINITY, INFINITY };

	int n = (int)(p.size() / 3);
	for (int i = 0; i < n; ++i) {
		for (int j = 0; j < 3; ++j) {
			GLfloat val = p[i * 3 + j];
			if (max_val[j] < val) max_val[j] = val;
			else if (min_val[j] > val) min_val[j] = val;
		}
	}

	GLfloat width[3] = {
		max_val[0] - min_val[0],
		max_val[1] - min_val[1],
		max_val[2] - min_val[2]
	};

	c.resize(p.size());
	for (int i = 0; i < n; ++i) {
		for (int j = 0; j < 3; ++j) {
			int k = i * 3 + j;
			c[k] = std::fminf((p[k] - min_val[j]) / width[j] + offset, 1.0f);
		}
	}
}
```

$0 \le r \le 1$

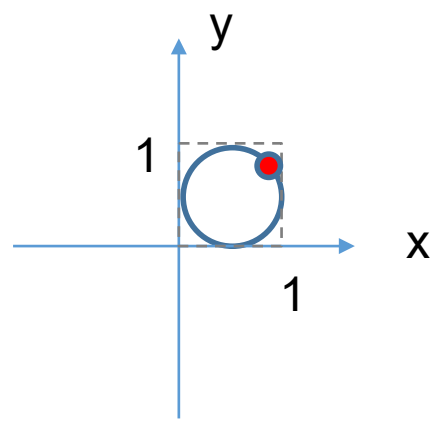
$0 \le g \le 1$

$0 \le b \le 1$

Compute the bounding box.



Normalize the coordinates.



- Explicitly setting the location of GLSL variables

```
#version 430

in vec4 vPosition;
in vec4 vColor;
out vec4 fColor;
layout(location=1) uniform mat4 T;

void main()
{
        gl_Position = T * vPosition;
        fColor = vColor;
}
```

```
glUniformMatrix4fv(1, 1, GL_FALSE, value_ptr(T_sun));
```

- Implementation of fragment shader

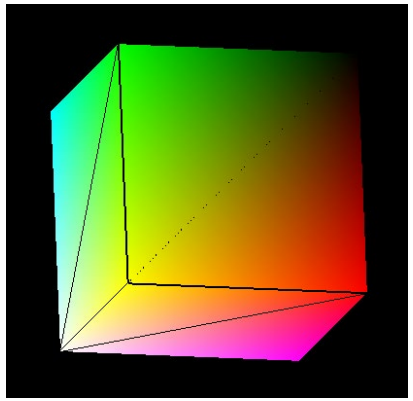
```
#version 430

in vec4 fColor;
out vec4 FragColor;
layout(location=2) uniform int draw_mode;

void main()
{
    switch(draw_mode)
    {
    case 0:
        FragColor = fColor;
        break;
    case 1:
        FragColor = vec4(0,0,0,1);
        break;
    }
}
```

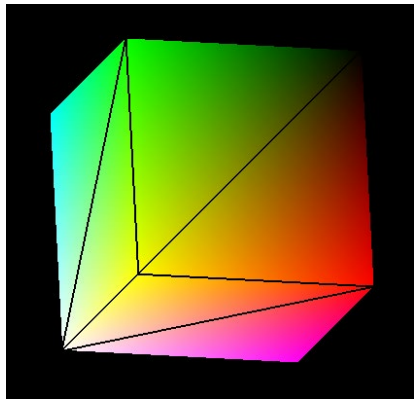
- Changing line width: `glLineWidth(width);`
- Changing point size: `glPointSize(size);`

- Example code:

```
void draw_sphere(const GLfloat* trans_mat)
{
     glUniformMatrix4fv(1, 1, GL_FALSE, trans_mat);
     glDrawArrays(GL_TRIANGLES, 0, num_of_vertices);
}
void display()
{
     ...
     ... Compute Transformations T_sun, T_earth, T_moon. ...
     glUniform1i(2, 0);
     draw_sphere(value_ptr(T_sun));
     draw_sphere(value_ptr(T_earth));
     draw_sphere(value_ptr(T_moon));

     if (show_vertices)
     {
          glPolygonMode(GL_FRONT_AND_BACK, GL_POINT);
          glPointSize(3);

          glUniform1i(2, 1);
          draw_sphere(value_ptr(T_sun));
          draw_sphere(value_ptr(T_earth));
          draw_sphere(value_ptr(T_moon));
     }
     ...
}
```

- Drawing wireframe lines clearly





```
void display()
{
     ...
     glEnable(GL_POLYGON_OFFSET_FILL);
     glPolygonOffset(1, 1);

     // Draw your colored objects here.
     ...
     glUniform1i(2, 0);
     draw_sphere(value_ptr(T_sun));
     draw_sphere(value_ptr(T_earth));
     draw_sphere(value_ptr(T_moon));

     glDisable(GL_POLYGON_OFFSET_FILL);

     if (show_wireframe)
     {
          glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
          glLineWidth(1);
          ...
     }
     if (show_vertices)
     {
          ...
     }
     ...
}
```

- What to submit:
  - A **zip file** that compresses the following files:
    - **Project source files** except libraries.
      - Clean your project before compression by selecting **Build → Clean Solution** in the main menu.
    - Three screen capture images for drawing spheres with wireframes, with points and without both wireframes and points, respectively
  - File name format
    - **hw6_000000.zip**, where 000000 must be replaced by your own student ID.
- Due date: **to be announced later**