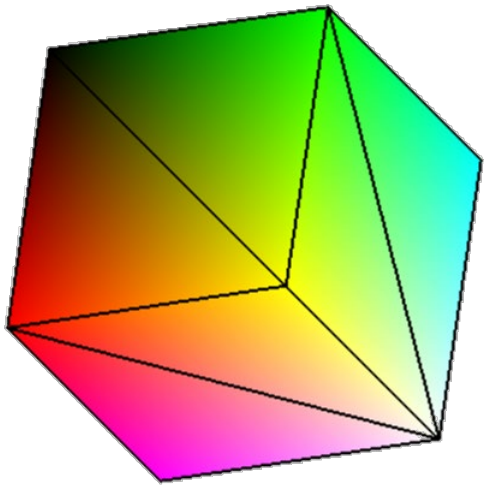
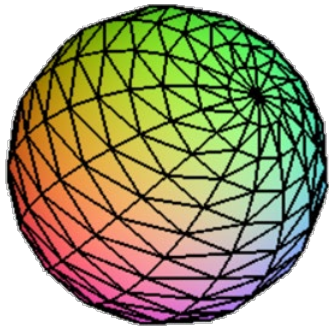


# Homework 8: Drawing Primitives

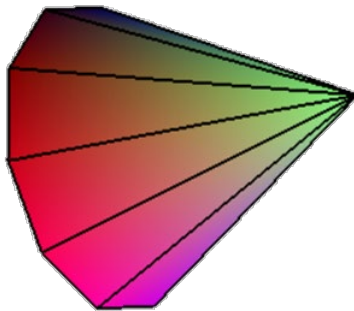
- Goal
  - To display five different types of primitives and a composite model composed of these as shown below:



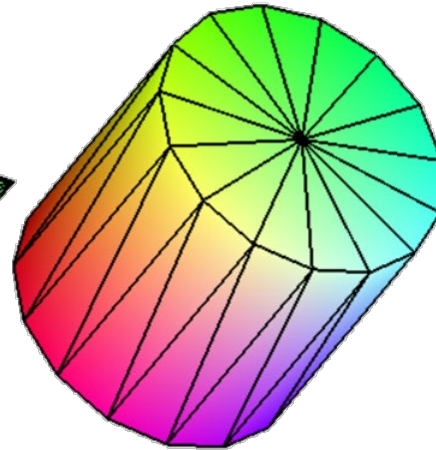
1. Cube



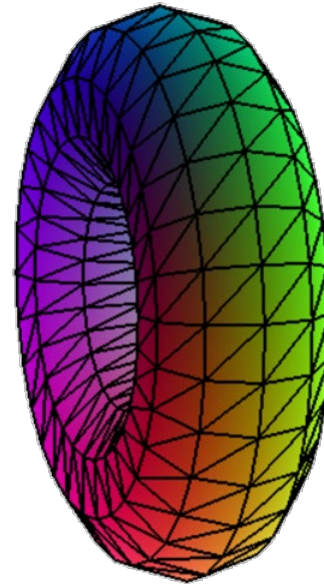
2. Sphere



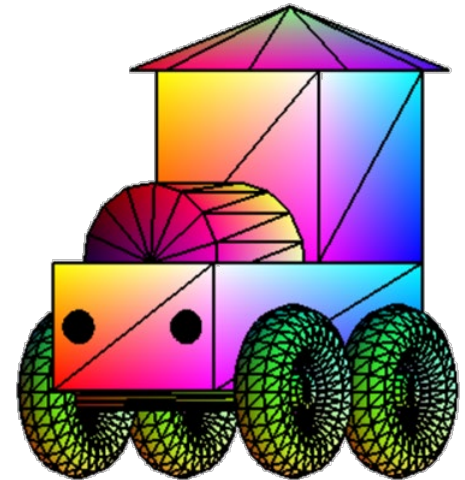
3. Cone



4. Cylinder



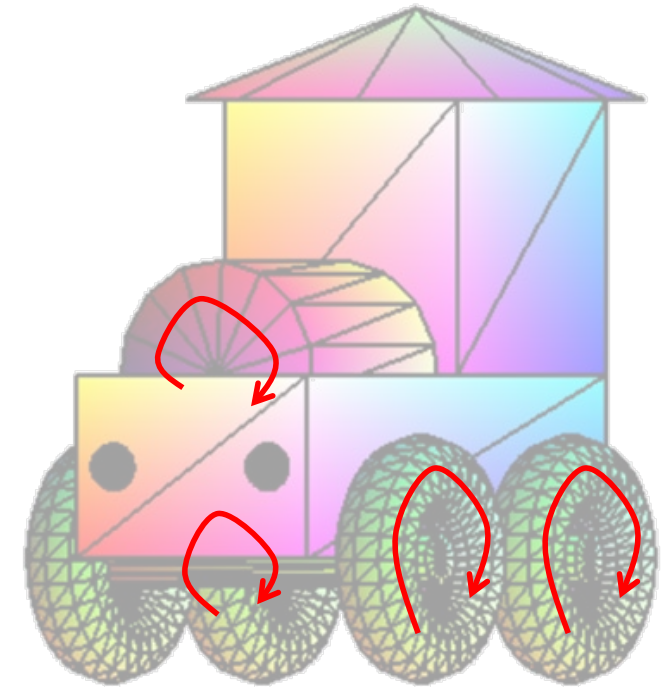
5. Torus



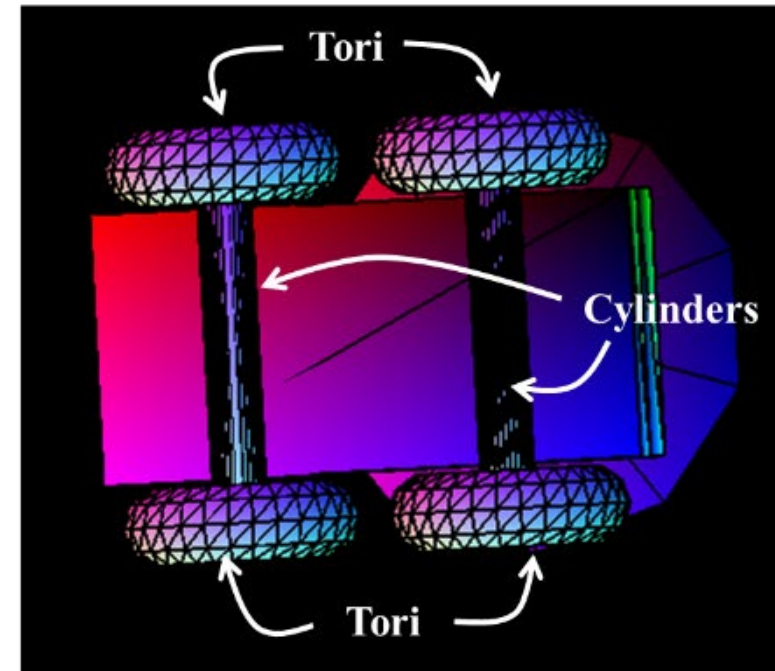
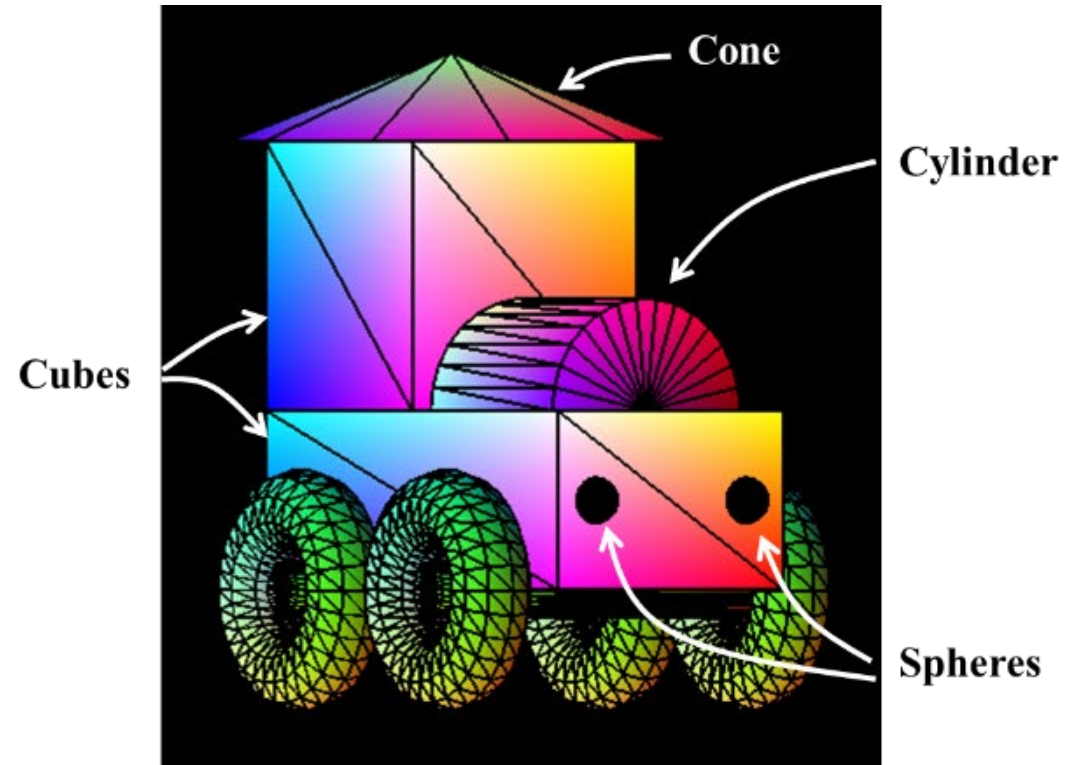
6. Composite Model

- Requirements

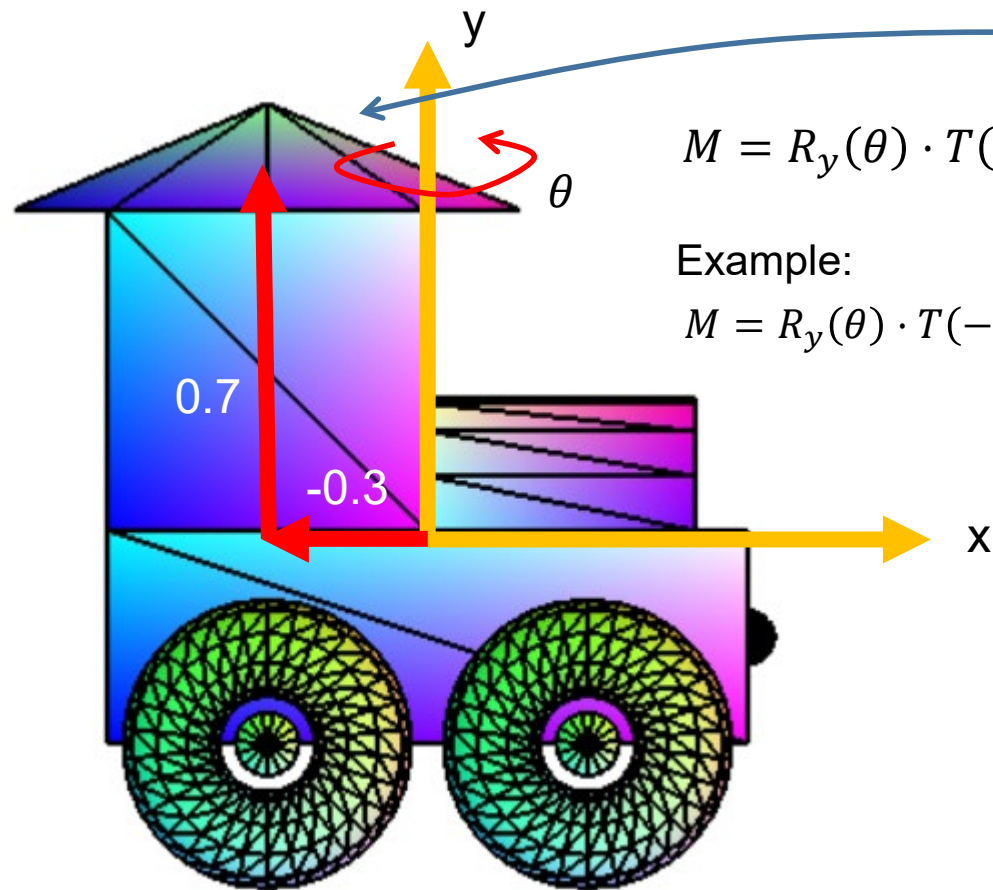
- When the user presses '1', '2', '3', '4', '5', and '6', your program should show a cube, a sphere, a cone, a cylinder, a torus, and a composite model, respectively.
- Rotate the current model being shown on the screen about an arbitrary axis. In the case of the composite model, the tires and the shafts of the model and the front cylinder body must also be rotated.
- When 'w' is pressed, toggle the wireframe of the current model on and off.
- When 'o' and 'p' is pressed, toggle on the orthographic projection and the perspective projection, respectively.



- How to draw the composite model



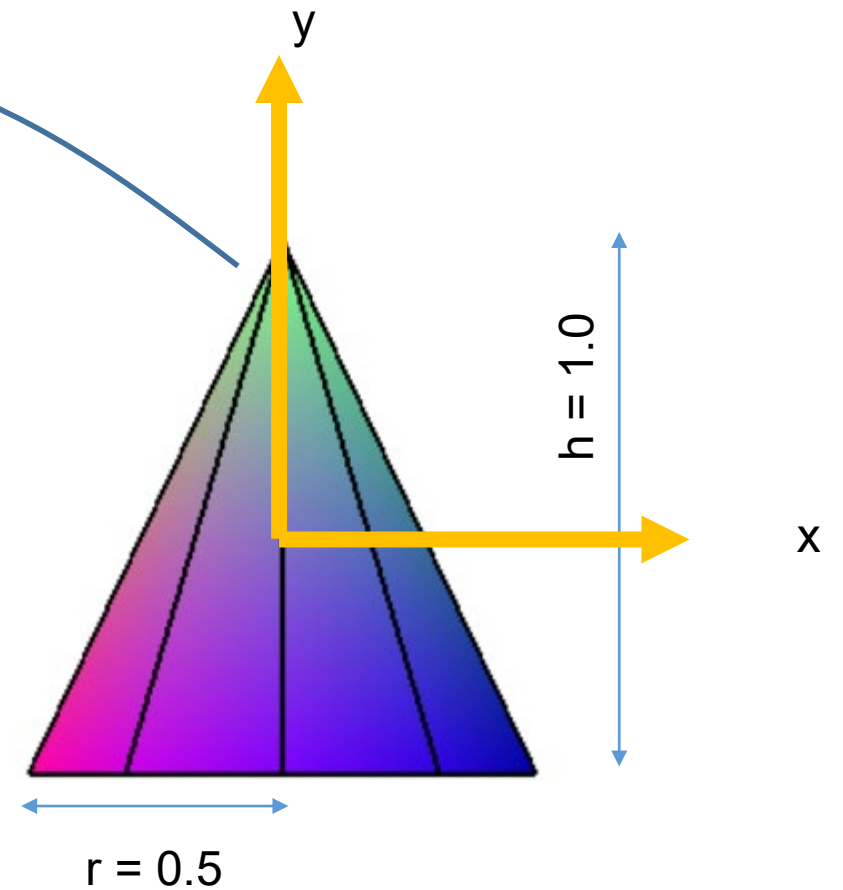
- How to draw the composite model



$$M = R_y(\theta) \cdot T(x, y, z) \cdot S(s_x, s_y, s_z)$$

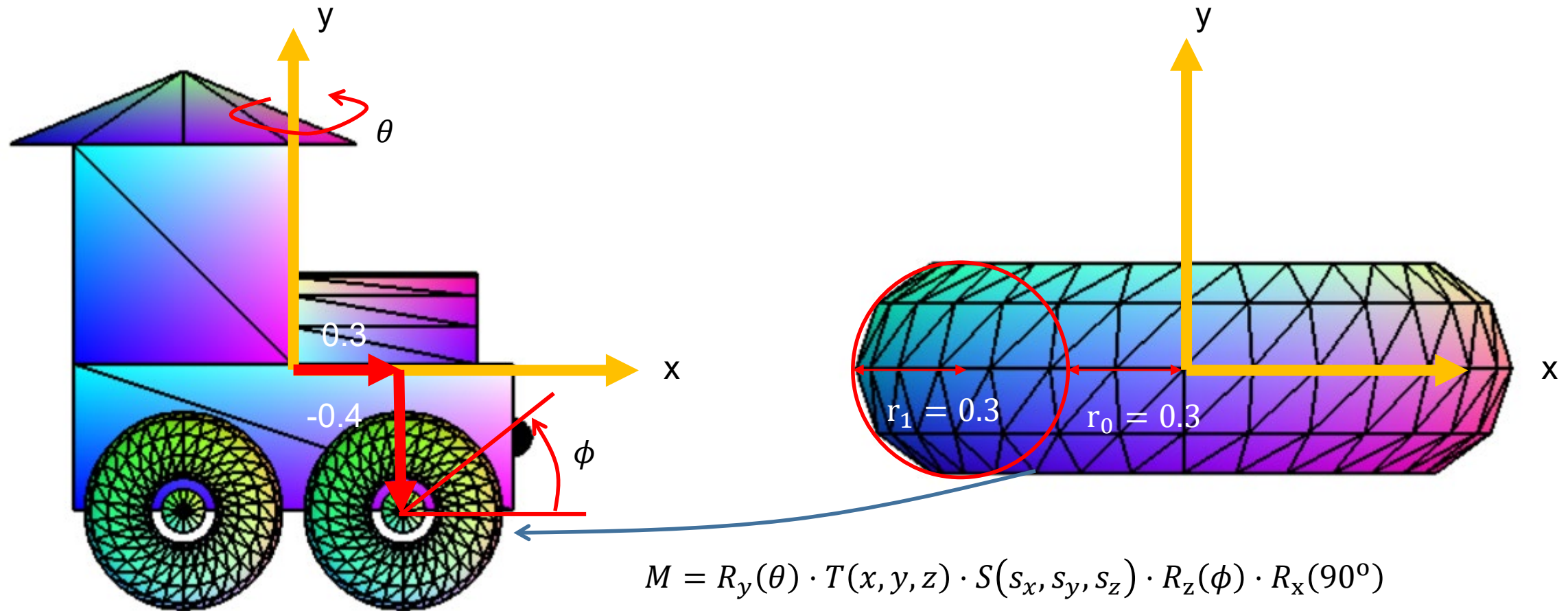
Example:

$$M = R_y(\theta) \cdot T(-0.3, 0.7, 0.0) \cdot S(1, 0.2, 1)$$





- How to draw the composite model



$$M = R_y(\theta) \cdot T(x, y, z) \cdot S(s_x, s_y, s_z) \cdot R_z(\phi) \cdot R_x(90^\circ)$$

Example:

$$M = R_y(\theta) \cdot T(0.3, -0.4, 0.4) \cdot S(0.3, 0.3, 0.3) \cdot R_z(\phi) \cdot R_x(90^\circ)$$

- How to do the orthographic / perspective projection

```
#version 430
```

```
in vec4 vPosition;
```

```
in vec4 vColor;
```

```
out vec4 fColor;
```

```
layout(location=1) uniform mat4 M;
```

```
layout(location=2) uniform mat4 V;
```

```
layout(location=3) uniform mat4 P;
```

```
void main()
```

```
{
```

```
    gl_Position = P * V * M * vPosition;
```

```
    fColor = vColor;
```

```
}
```

Vertex shader

```
int projection_mode = 0;
```

It must be properly modified  
according to user's keyboard  
input.

```
int width = glGet(GLUT_WINDOW_WIDTH);
```

```
int height = glGet(GLUT_WINDOW_HEIGHT);
```

```
double aspect = 1.0 * width / height;
```

```
mat4 V = lookAt(vec3(0, 0, 5), vec3(0, 0, 0), vec3(0, 1, 0));
```

```
mat4 P(1.0);
```

```
if (projection_mode == 0) {
```

```
    P = parallel(1.2, aspect, 0.01, 10.0);
```

```
}
```

```
else
```

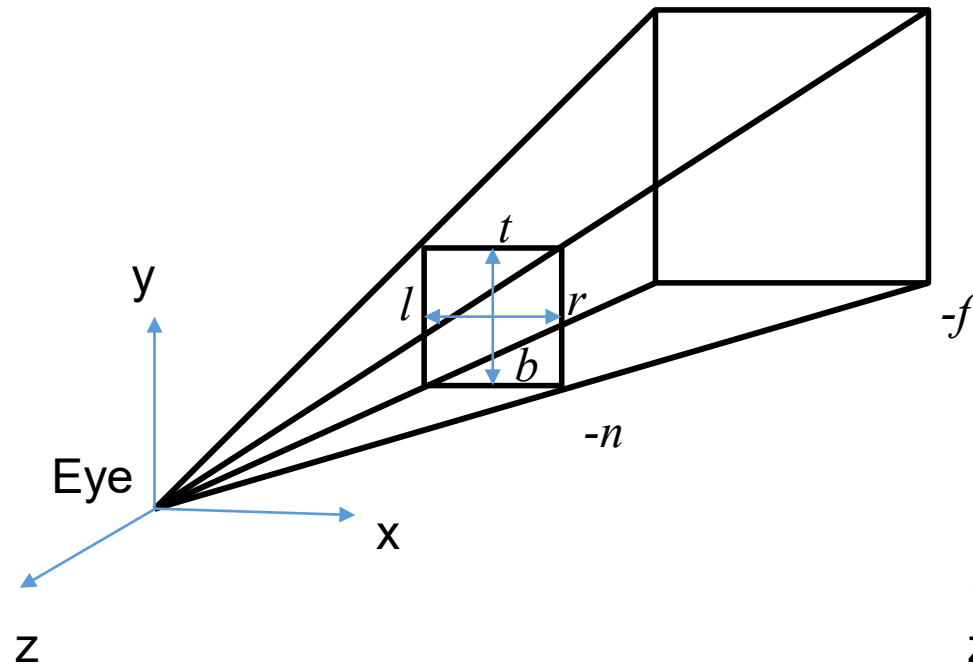
```
    P = perspective(M_PI/180.0*(30.0), aspect, 0.01, 10.0);
```

```
glUniformMatrix4fv(2, 1, GL_FALSE, value_ptr(V));
```

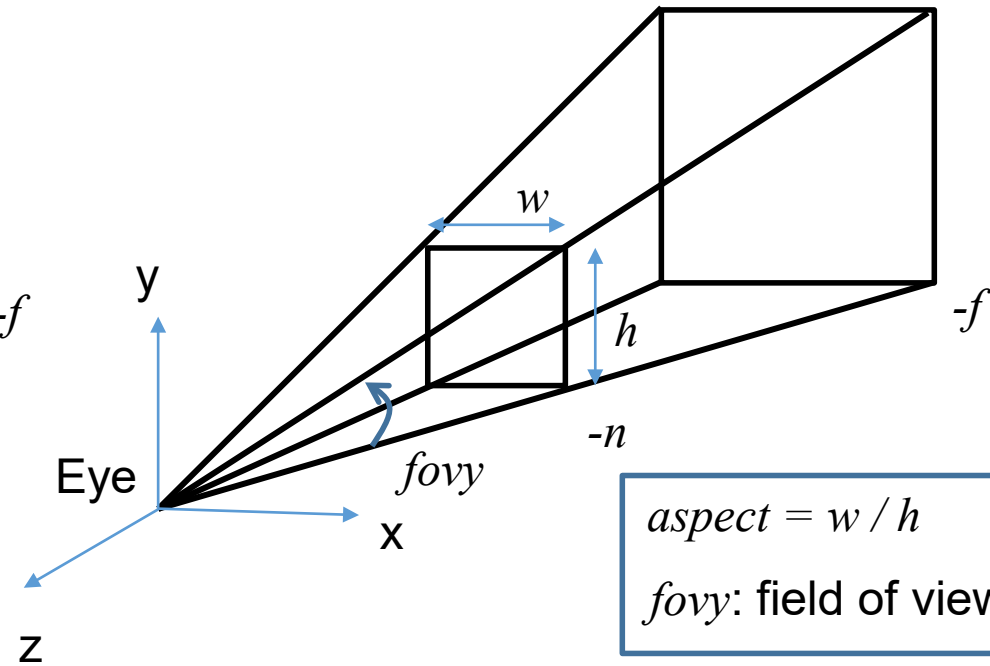
```
glUniformMatrix4fv(3, 1, GL_FALSE, value_ptr(P));
```



- How to specify the viewing frustum



`mat4 M = glm::frustum( l, r, b, t, n, f )`



$$aspect = w / h$$

*fovy*: field of view in the  $y$ -axis

`mat4 M = glm::perspective( fovy, aspect, n, f )`



- How to set the clear color to white: **glClearColor(1, 1, 1, 1);**
- Modify the call to `glPolygonOffset(...)` as follows:

`glPolygonOffset(-1, -1);`  `glPolygonOffset(1, 1);`

\* This is because the depth values are reversed by the projection transformation.

- 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.
    - Seven screen capture images that respectively show the cube, sphere, cone, cylinder, torus and composite model with the wireframe mode turned on
  - File name format
    - **hw8\_000000.zip**, where 000000 must be replaced by your own student ID.
- Due date: **To be announced later**