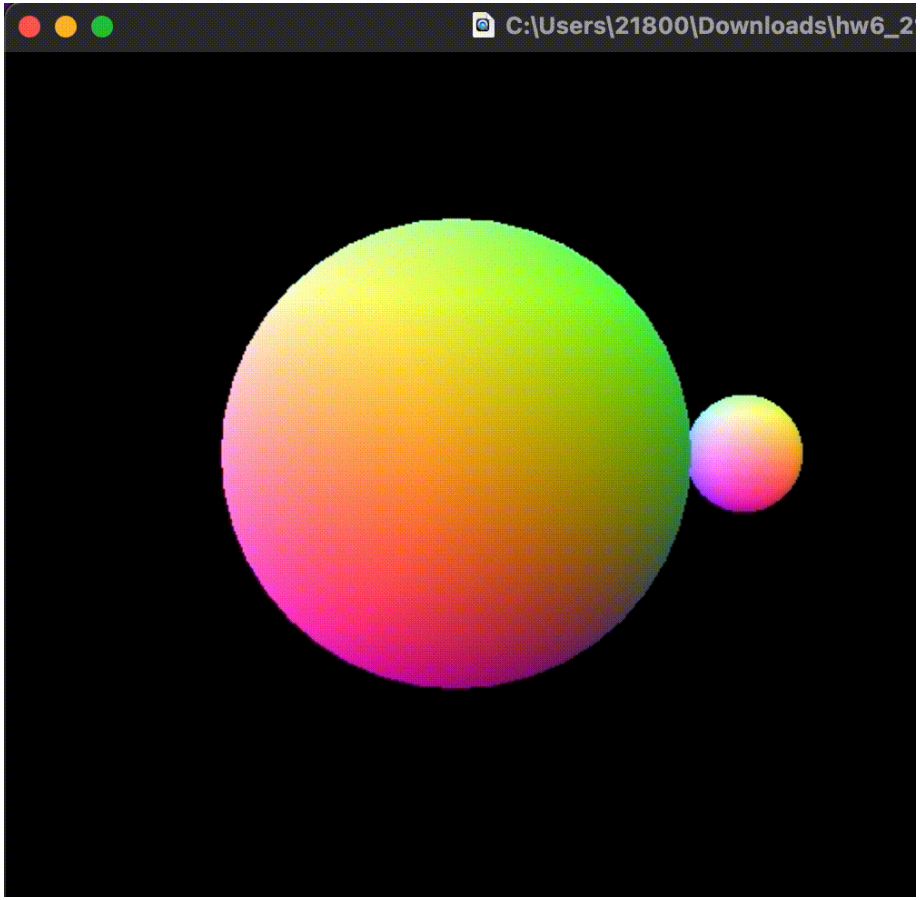
Computer Graphics
Orbital revolution

Sung Soo Hwang

Practice: Orbital revolution

- Use the spheres of “primi.h” to implement the orbital revolution.



Practice: Orbital revolution

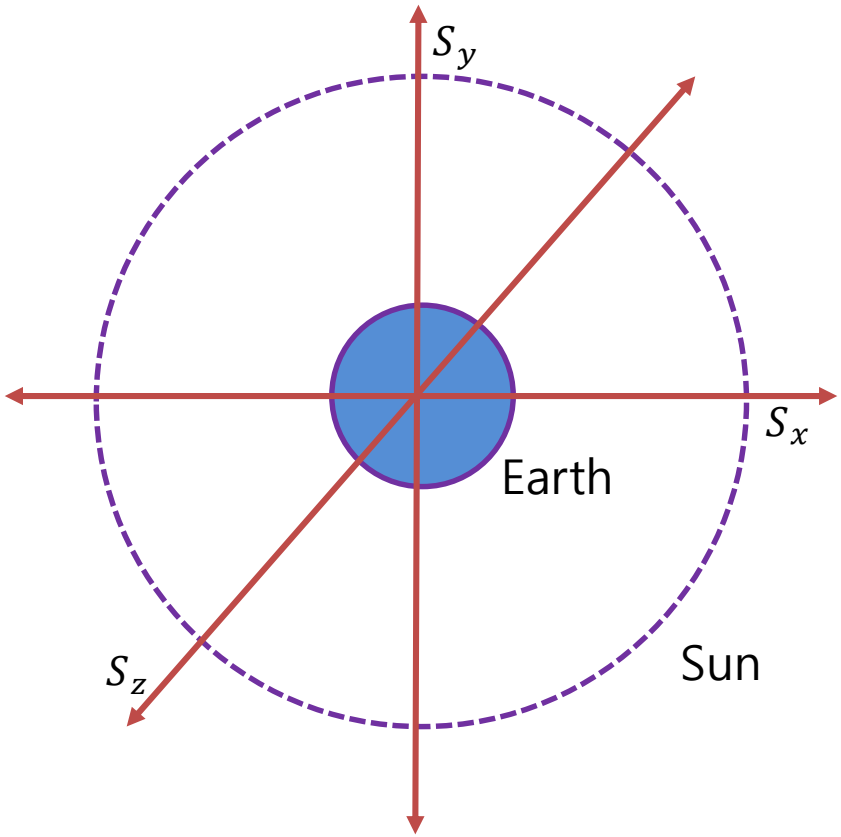
Orbiting around the sun

Self-rotation

$$T_{earth} = R_y \cdot T(0.8,0,0) \cdot R_y \cdot S(0.25)$$

Translation to
The earth position

Scaling



Practice: Orbital revolution

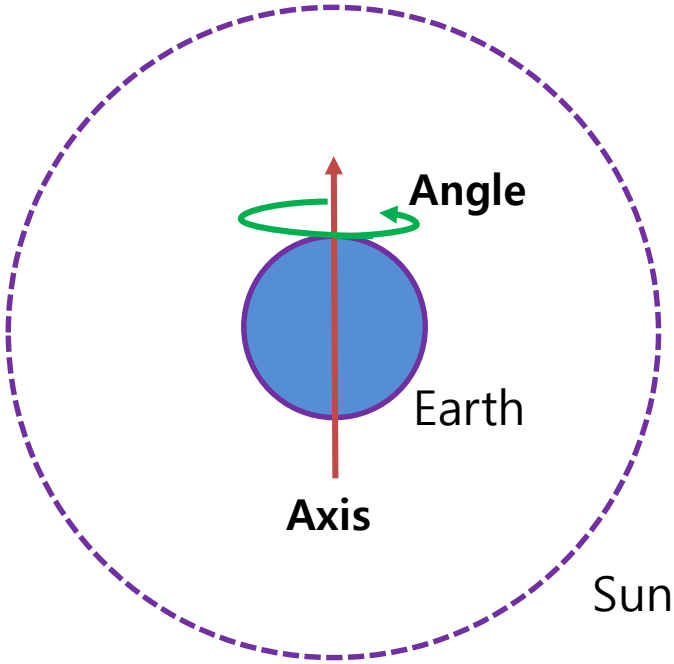
Orbiting around the sun

Self-rotation

$$T_{earth} = R_y \cdot T(0.8,0,0) \cdot R_y \cdot S(0.25)$$

Translation to
The earth position

Scaling



Practice: Orbital revolution

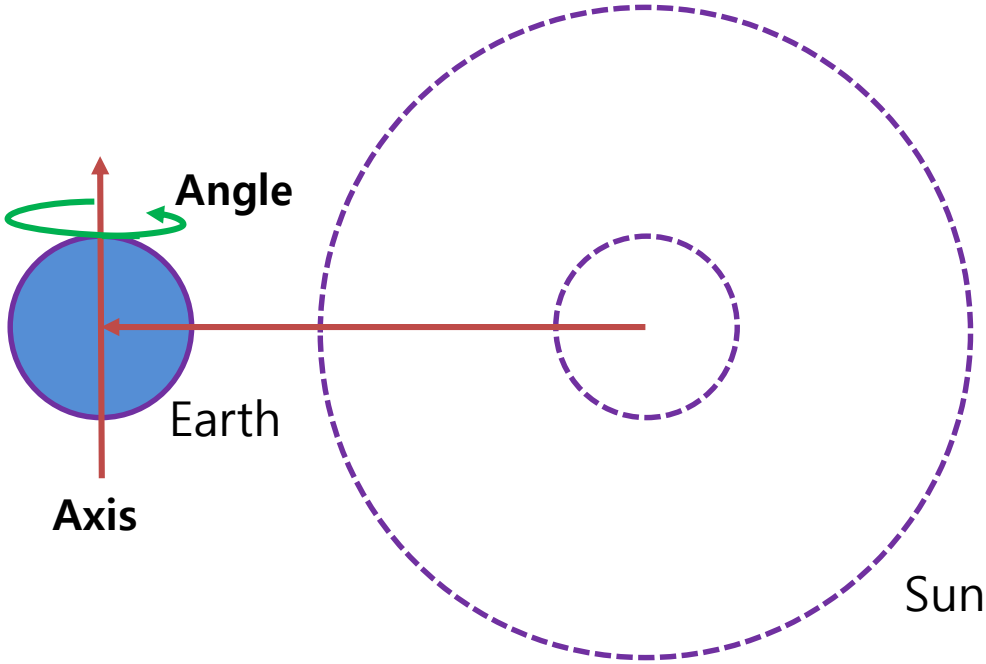
Orbiting around the sun

Self-rotation

$$T_{earth} = R_y \cdot T(0.8,0,0) \cdot R_y \cdot S(0.25)$$

Translation to
The earth position

Scaling



Practice: Orbital revolution

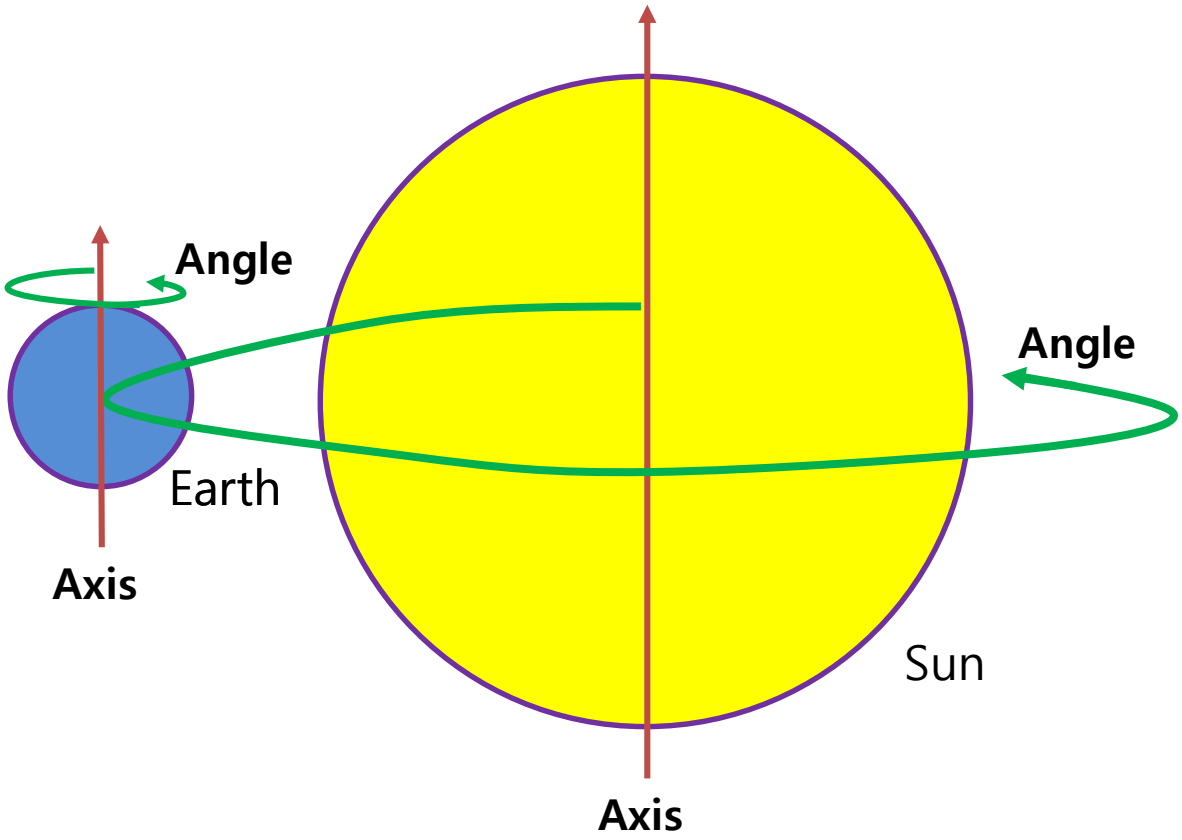
Orbiting around the sun

Self-rotation

$$T_{earth} = R_y \cdot T(0.8,0,0) \cdot R_y \cdot S(0.25)$$

Translation to
The earth position

Scaling



Practice: Orbital revolution

How to implement in OpenGL code.

Ex. GLfloat theta = 0.001f * clock();

```
mat4 T_earth(1.0f);
T_earth = rotate(T_earth, theta, vec3(0.0f, 1.0f, 0.0f));
T_earth = translate(T_earth, vec3(0.8f, 0.0f, 0.0f));
T_earth = rotate(T_earth, theta, vec3(0.0f, 1.0f, 0.0f));
T_earth = scale(T_earth, vec3(0.25f));
```

Orbiting around the sun

Self-rotation

$$T_{earth} = R_y \cdot T(0.8,0,0) \cdot R_y \cdot S(0.1)$$

Translation to
The earth position

Scaling