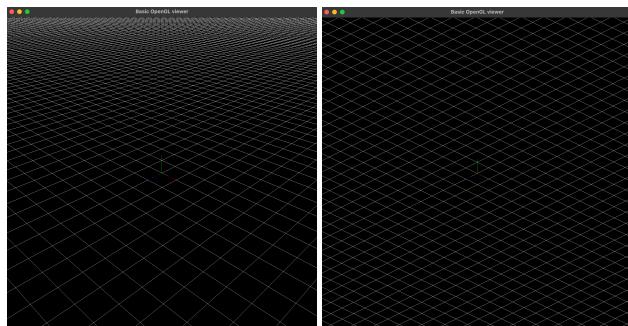


# ClassAssignment2 Report

2016025041 하태성

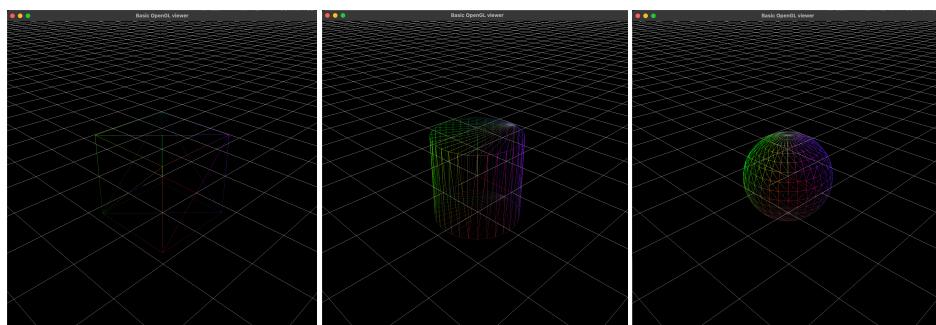
## - Manipulating Camera as in ClassAssignment1 + Draw Grid Plane

- Perspective & Orthogonal



## - Open obj file by Drag-and-Drop

- cube.obj / cylinder.obj / sphere.obj



## - Use glDrawArrays() or glDrawElement() / Print Information to stdout

```
def drawObject(varr, fvarr, snarr, fnarr, iarr):
    global Toggle_Shading

    glEnableClientState(GL_VERTEX_ARRAY)
    glEnableClientState(GL_NORMAL_ARRAY)

    if Toggle_Shading == ForcedShading:
        if varr.size == 0: return
        glNormalPointer(GL_FLOAT, 3*snarr.itemsize, snarr)
        glVertexPointer(3, GL_FLOAT, 3*varr.itemsize, varr)
        glDrawElements(GL_TRIANGLES, iarr.size, GL_UNSIGNED_INT, iarr)
    elif Toggle_Shading == ShadingNormal:
        if fvarr.size == 0: return
        glNormalPointer(GL_FLOAT, 3*fnarr.itemsize, fnarr)
        glVertexPointer(3, GL_FLOAT, 3*fvarr.itemsize, fvarr)
        glDrawArrays(GL_TRIANGLES, 0, int(fvarr.size/3))
```

```
▶ python main.py
File name : cube-tri-quad.obj
Total number of faces : 10
Number of faces with 3 vertices : 8
Number of faces with 4 vertices : 2
Number of faces with more than 4 vertices : 0
File name : cylinder-tri-quad-n.obj
Total number of faces : 45
Number of faces with 3 vertices : 22
Number of faces with 4 vertices : 21
Number of faces with more than 4 vertices : 2
File name : sphere-tri-quad.obj
Total number of faces : 524
Number of faces with 3 vertices : 88
Number of faces with 4 vertices : 436
Number of faces with more than 4 vertices : 0
```

## - Animating Hierarchical Model Rendering Mode

- at least 3 different meshes loaded from 3 different downloaded obj files
- Use os.path.join()

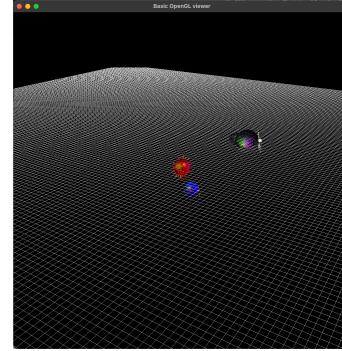
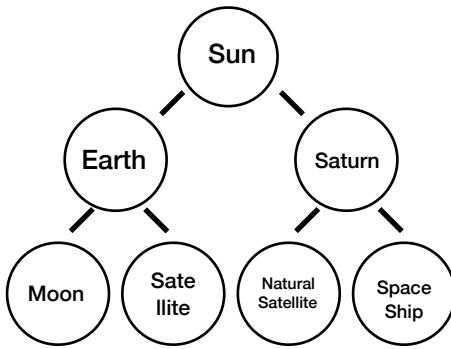
```

path_sun = os.path.join(".", "Sun.obj")
sun_varr, sun_fvarr, sun_snarr, sun_fnarr, sun_iarr = objectRender(path_sun)
path_earth = os.path.join(".", "Earth.obj")
earth_varr, earth_fvarr, earth_snarr, earth_fnarr, earth_iarr = objectRender(path_earth)
path_moon = os.path.join(".", "Moon.obj")
moon_varr, moon_fvarr, moon_snarr, moon_fnarr, moon_iarr = objectRender(path_moon)
path_satellite = os.path.join(".", "Satellite2.obj")
satellite_varr, satellite_fvarr, satellite_snarr, satellite_fnarr, satellite_iarr = objectRender(path_satellite)
path_spaceship = os.path.join(".", "Spaceship.obj")
spaceship_varr, spaceship_fvarr, spaceship_snarr, spaceship_fnarr, spaceship_iarr = objectRender(path_spaceship)
path_naturalite = os.path.join(".", "Natural_Satellite.obj")
naturalite_varr, naturalite_fvarr, naturalite_snarr, naturalite_fnarr, naturalite_iarr = objectRender(path_naturalite)

```

### - Hierarchical Model with Using OpenGL matrix stck

- 3 levels and each node have 2 childs / Animate - relative move



- Hyperlink : <https://youtu.be/3T9R72P8IHG>

### - Lighting

- 3 light source
- put at (10, 10, 10)  
/ (-10, 10, 10) / (10, 10, -10)
- directional light

```

glEnable(GL_LIGHTING)
glEnable(GL_LIGHT0)
glEnable(GL_LIGHT1)
glEnable(GL_LIGHT2)
glEnable(GL_NORMALIZE)

lightPos0 = (10., 10., 10., 1.)
lightColor0 = (1., 0., 0., 1.)
ambientLightColor0 = (.1, .0, .0, 1.)
glLightfv(GL_LIGHT0, GL_POSITION, lightPos0)
glLightfv(GL_LIGHT0, GL_AMBIENT, ambientLightColor0)
glLightfv(GL_LIGHT0, GL_DIFFUSE, lightColor0)
glLightfv(GL_LIGHT0, GL_SPECULAR, lightColor0)

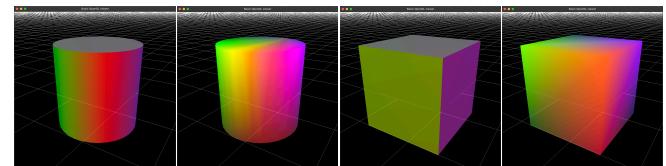
lightPos1 = (-10., 10., 10., 1.)
lightColor1 = (0., 1., 0., 1.)
ambientLightColor1 = (.0, .1, 0., 1.)
glLightfv(GL_LIGHT1, GL_POSITION, lightPos1)
glLightfv(GL_LIGHT1, GL_AMBIENT, ambientLightColor1)
glLightfv(GL_LIGHT1, GL_DIFFUSE, lightColor1)
glLightfv(GL_LIGHT1, GL_SPECULAR, lightColor1)

lightPos2 = (10., 10., -10., 1.)
lightColor2 = (0., 0., 1., 1.)
ambientLightColor2 = (.0, .0, 1., 1.)
glLightfv(GL_LIGHT2, GL_POSITION, lightPos2)
glLightfv(GL_LIGHT2, GL_AMBIENT, ambientLightColor2)
glLightfv(GL_LIGHT2, GL_DIFFUSE, lightColor2)
glLightfv(GL_LIGHT2, GL_SPECULAR, lightColor2)

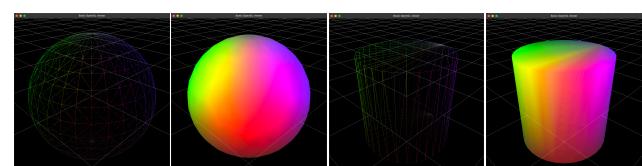
```

### - Etc & Extra Credits

- \* Toggle shading



- \* Toogle wireframe / solidmode



- \* Triangulation

- No gl\_QUADS & gl\_POLYGON
- Only Triangle mesh with gl\_TRIANGLES

### - All Requirement Implemented!!