

---

# HW2 Modeling and Viewing Transformations

---

**Jiayao Zhang**  
Department of Computer Science  
King Abdullah University of Science and Technology  
jiayao.zhang@kaust.edu.sa

For further references see [Shadertoy Example](#)

## 1 Requirement 1

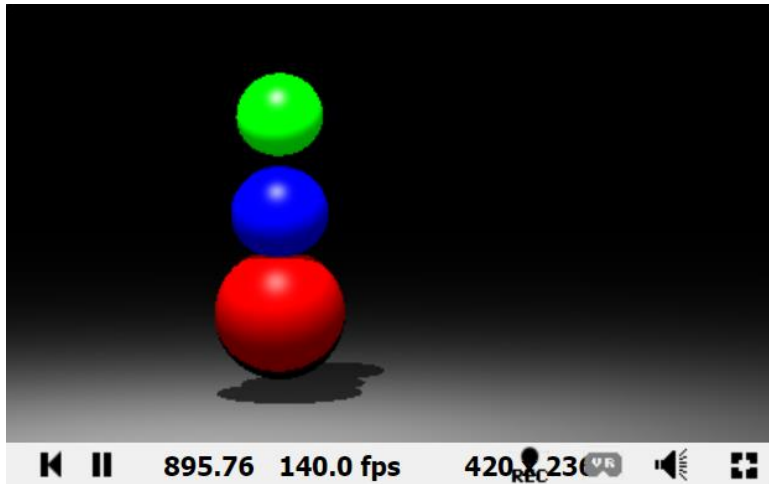


Figure 1: The ray transformation

Once the object is selected:

- move it in a view-aligned plane up, down, top, and bottom.

```
void mainImage( out vec4 fragColor, in vec2 fragCoord )
{
    vec2 mouse = iMouse.xy/iResolution.xy;
    vec3 wasd = chan3(.5).xyz;
    vec3 pos = vec3(mouse.x, mouse.y, 0.0);

    //!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!task 1!!!!!!!!!!!!!!
    obj_pos[1].xy += (mouse.xy - vec2(0.5))*10.;
```

- rotate it around its own center point in all three world coordinate axes (the center point of the object remains in the same position)

```
obj_pos[1] .x = 0.5 * cos(iTime);
obj_pos[1] .z = 0.5 * sin(iTime);
obj_pos[2] .x = 0.5 * cos(iTime);
obj_pos[2] .z = 0.5 * sin(iTime);
obj_pos[3] .x = 0.5 * cos(iTime);
obj_pos[3] .z = 0.5 * sin(iTime);
```

- scale it (center point of the object remains in the same position)

```
obj_prop[1].x = mouse.x *1.5;
```

- reset transformations

```
if(texelFetch( iChannel0, ivec2(KEY_b,2),0 ).x > 0.0){
    obj_prop[1].x = 0.5;
    obj_pos[1].xy = vec2(0.5);

    obj_pos[1] .x = 0.3 ;
    obj_pos[1] .z = 0.1 ;
    obj_pos[2] .x = 0.6 ;
    obj_pos[2] .z = 0.5 ;
    obj_pos[3] .x = 0.9 ;
    obj_pos[3] .z = 0.9 ;
}
```

## 2 Requirement 2

Implement simple camera controls:

- pan camera horizontally and vertically

```
vec3 wasd = chan3(.5).xyz;
```

- rotate camera around the scene from equal distance (horizontally and vertically)

```
vec3 vpp = wasd* 0.5 + vec3((-1.0 + uv * 2.0) * vec2(2.0, 2.0) * vec2(1.78*sin(iTime), 1.0 * sin(iTime)), 0);
```

- reset camera to the default position

```
if(texelFetch( iChannel0, ivec2(KEY_r,2),0 ).x > 0.0){
    vpp = wasd* 0.5 + vec3((-1.0 + uv * 2.0) * vec2(2.0, 2.0) * vec2(1.78, 1.0), 0);
}
```

### 3 Requirement

Example of additional options beyond 80%:

- perform zooming operation

```
vec3 vpp = wasd* 0.5 + vec3((-1.0 + uv * 2.0) * vec2(2.0, 2.0) * vec2(1.78*sin(iTime), 1.0 * sin(iTime)), 0);
```

- make the scene dynamic where either camera flies a certain path or the scene objects are animated (eg. implement simple physics)

```
vec3 vpp = wasd* 0.5 + vec3((-1.0 + uv * 2.0) * vec2(2.0, 2.0) * vec2(1.78*sin(iTime), 1.0 * sin(iTime)), 0);

if(texelFetch( iChannel0, ivec2(KEY_r,2),0 ).x > 0.0){
    vpp = wasd* 0.5 + vec3((-1.0 + uv * 2.0) * vec2(2.0, 2.0) * vec2(1.78, 1.0), 0);
}
```

```
//Task 1
obj_pos[1].xy += (mouse.xy - vec2(0.5))*10.;

obj_prop[1].x = mouse.x *1.5;
obj_pos[1].x = 0.5 * cos(iTime);
obj_pos[1].z = 0.5 * sin(iTime);
obj_pos[2].x = 0.5 * cos(iTime);
obj_pos[2].z = 0.5 * sin(iTime);
obj_pos[3].x = 0.5 * cos(iTime);
obj_pos[3].z = 0.5 * sin(iTime);
```

- modify the position of the light source

```
light.xz += vec2(sin(iTime), cos(iTime))*5.;
```