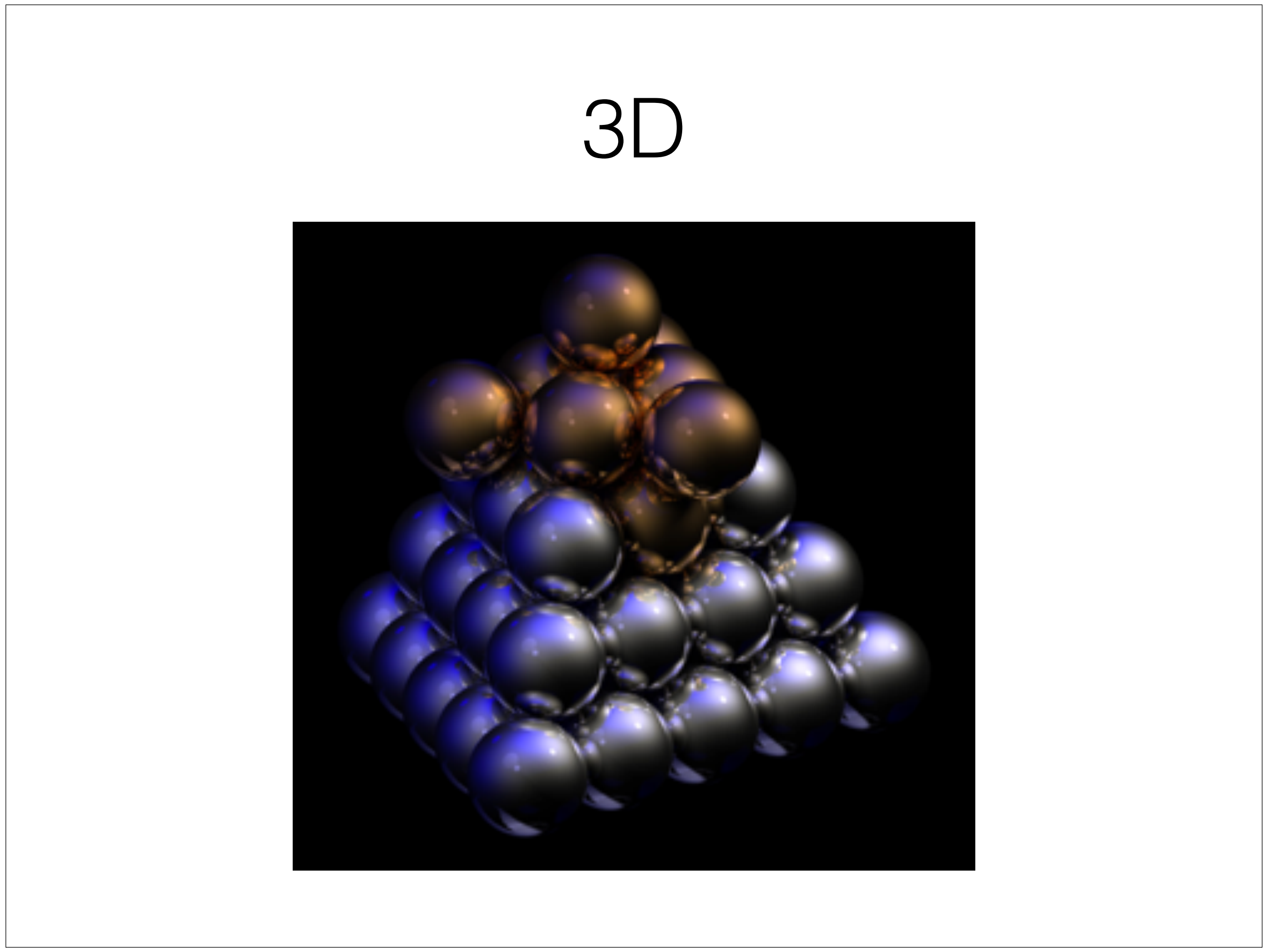
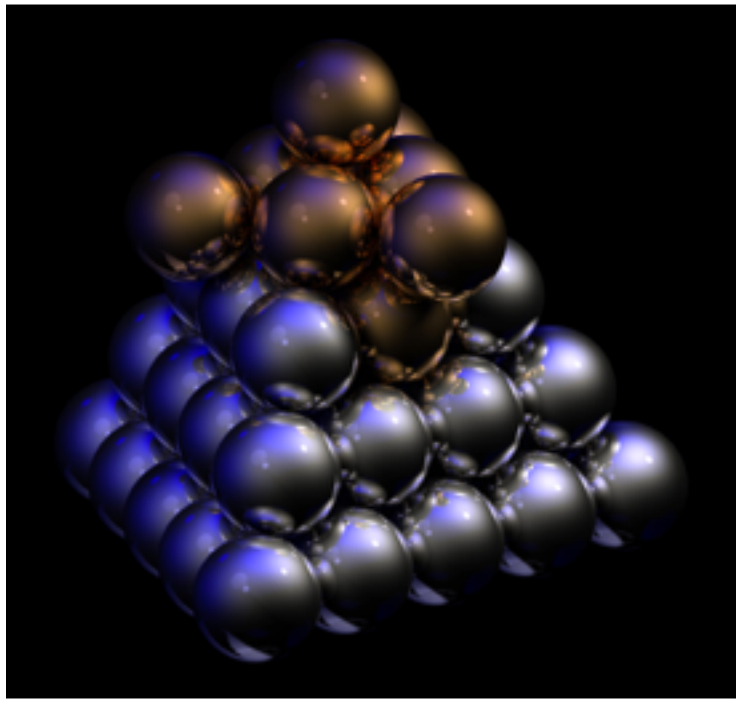
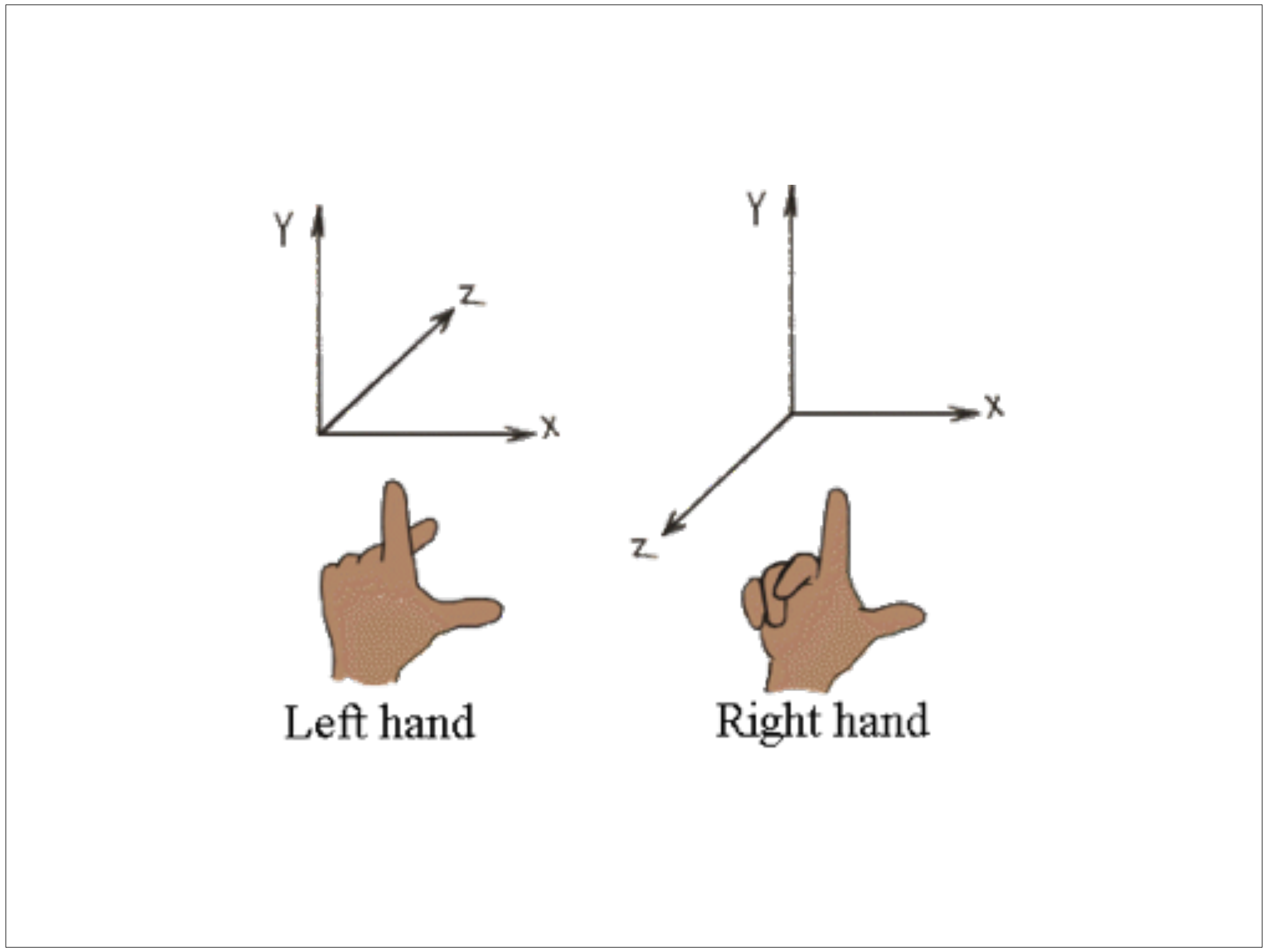


3D



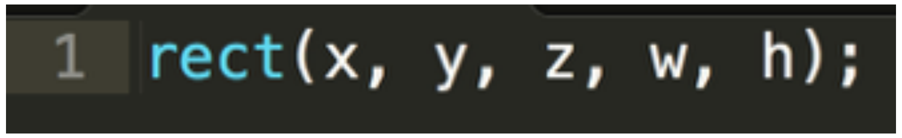
In three-dimensional space, a third axis (the z-axis) refers to the depth of any given point. How far in front or behind the window does a pixel live?
we're talking about here is how to use the theoretical z-axis to create the illusion of three-dimensional space

as soon as we give ourselves over to the illusion of 3D space,
a certain amount of control must be relinquished to the P3D renderer.
You can no longer know exact pixel locations
as you might with 2D shapes, because the 2D locations will be adjusted to create the illusion of 3D perspective.

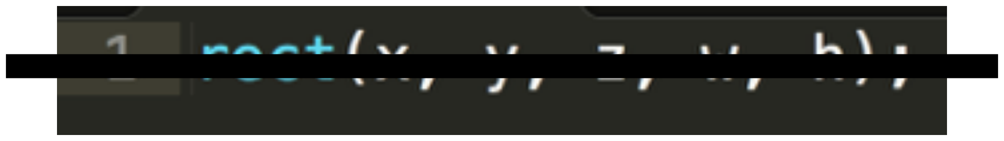


If you point your index finger in the positive y direction (down) and your thumb in the positive x direction (to the right), the rest of your fingers will point towards the positive z direction

positive comes towards you

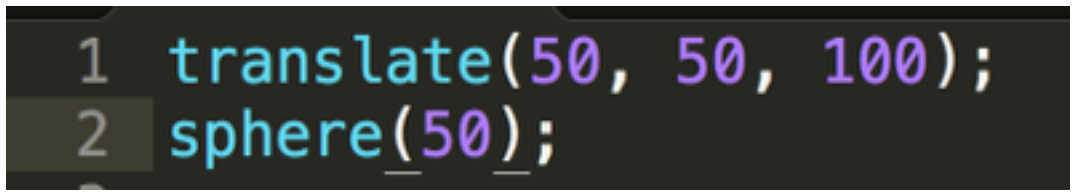


```
rect(x, y, z, w, h);
```



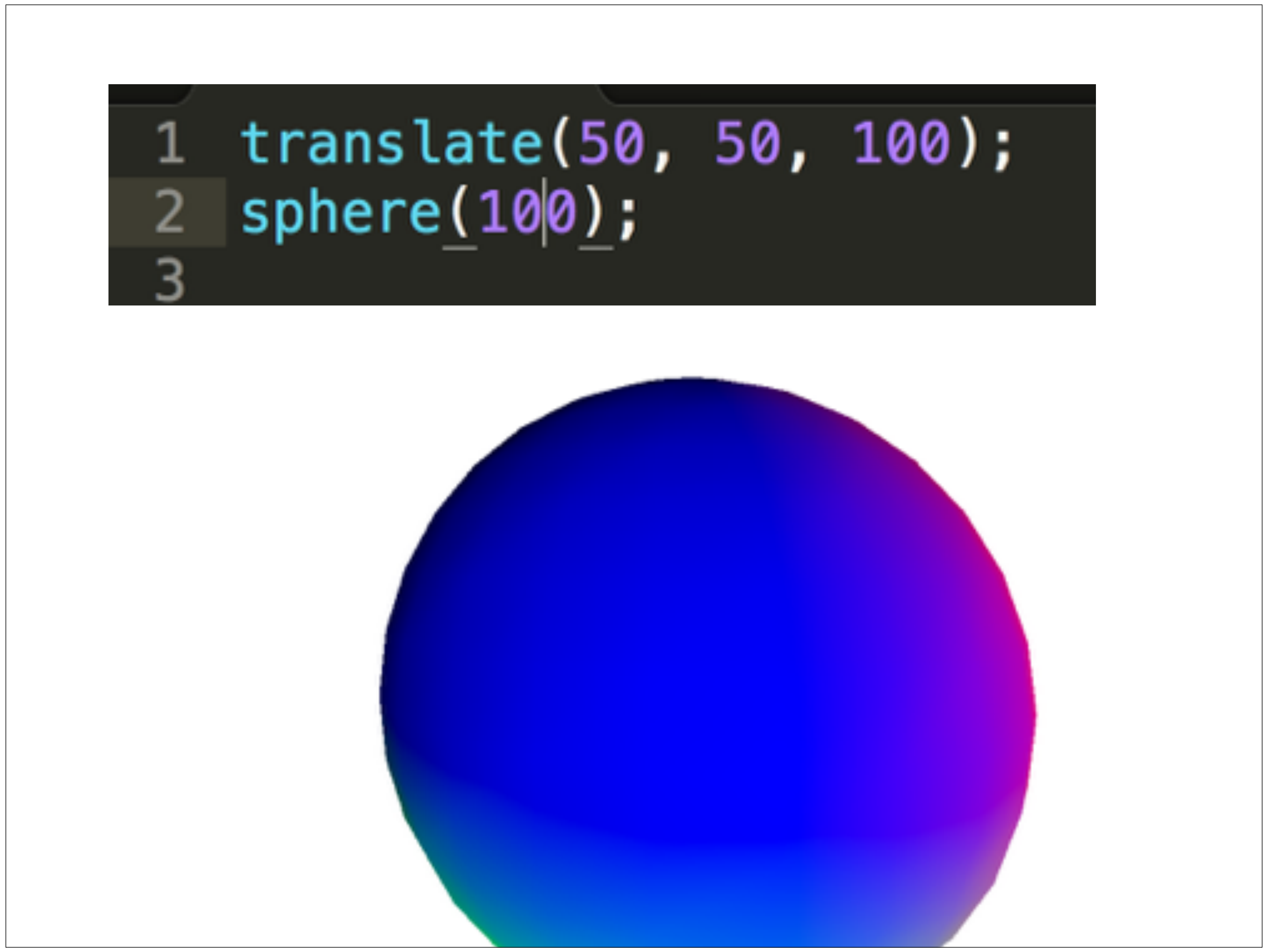
~~`rect(x, y, z, w, h);`~~

it's often simpler to specify vertex locations using a standardized unit of measure (i.e. 1 pixel)
and relative to a point of origin (0,0,0).
The size and position of the shape is then set using matrix transformations: translate(), rotate(), and scale().



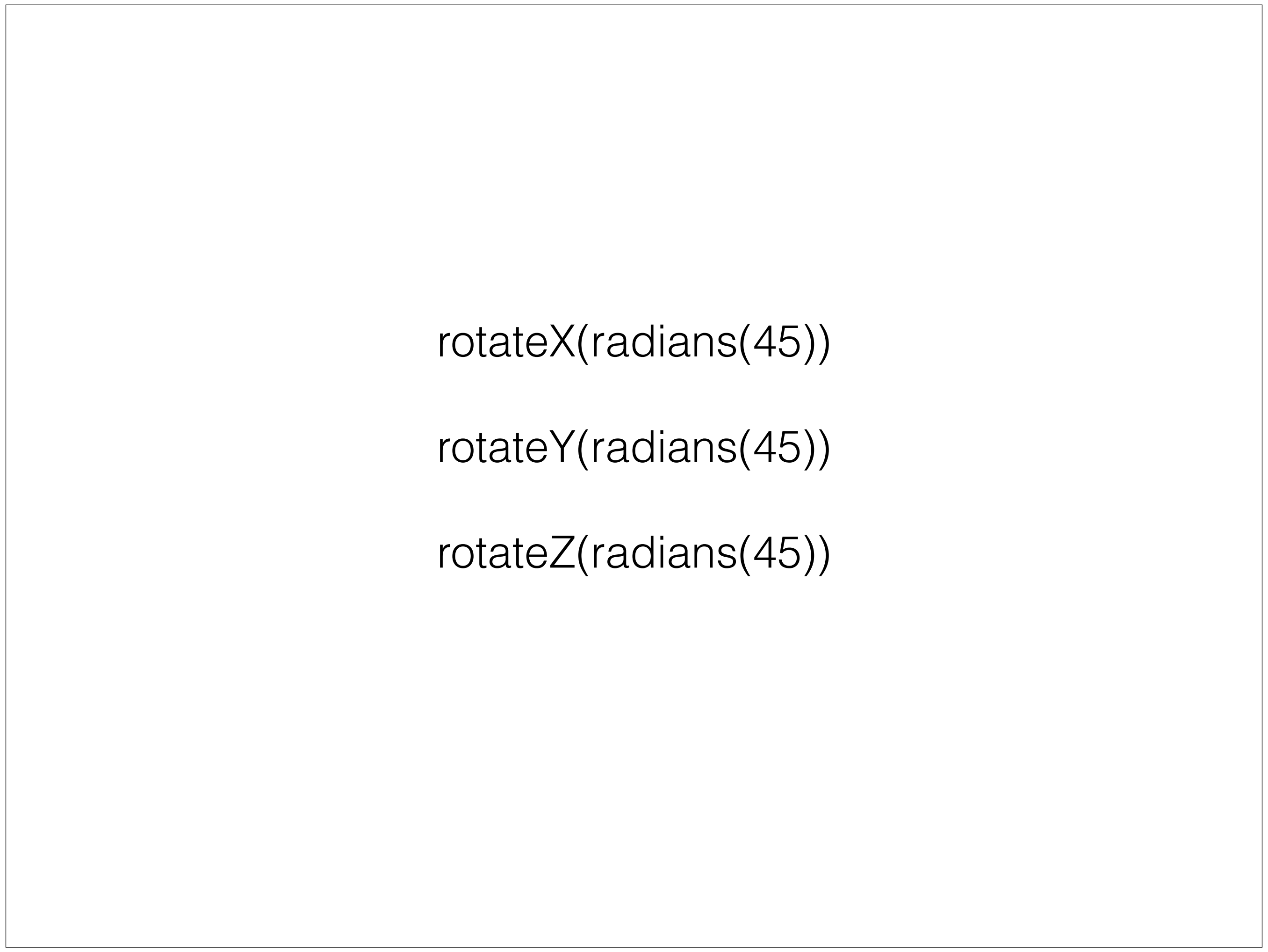
```
translate(50, 50, 100);
sphere(50);
```

can pass things sizes, but not positions. use translate() + rotate to position
In order to specify 3D coordinates for shapes, you have to use translate()
The size and position of the shape is then set using matrix transformations: translate(), rotate(), and scale().



```
translate(50, 50, 100);
sphere(100);
```

can pass things sizes, but not positions. use translate() + rotate to position
In order to specify 3D coordinates for shapes, you have to use translate()
The size and position of the shape is then set using matrix transformations: translate(), rotate(), and scale().



```
rotateX(radians(45))

rotateY(radians(45))

rotateZ(radians(45))
```

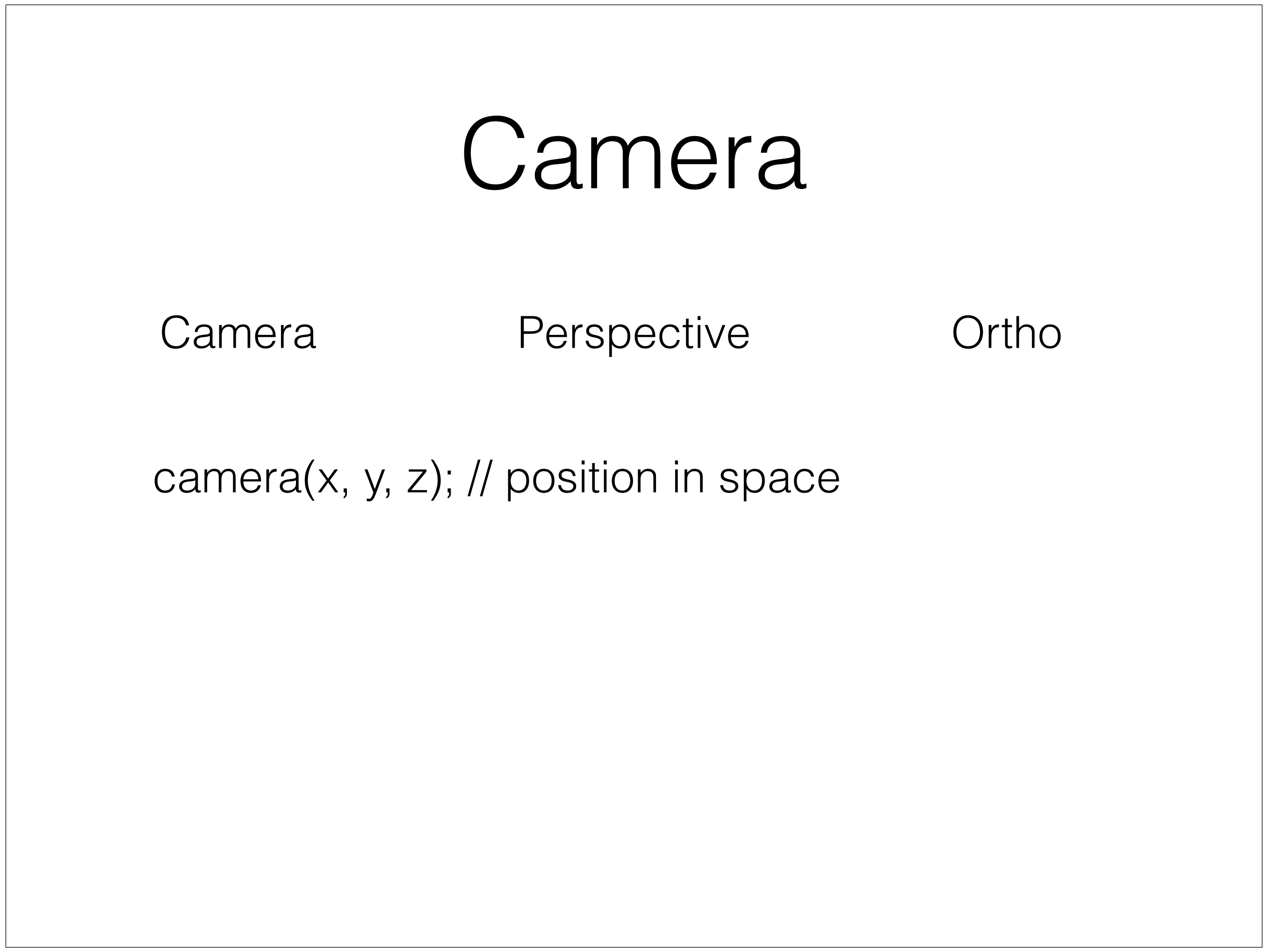
see demo 0

Lighting

Ambient　　Directional　　Point

can manipulate lighting to facilitate the 3D feeling and illusion of space
sphere doesn’t appear 3D until it’s lite

see code



Camera

Camera Perspective Ortho

camera(x, y, z); // position in space

Camera - our view of the scene; zoom in on objects - they get closer. rotate around them, etc

Camera

Camera Perspective Ortho

```
perspective(fov, aspect, near, far);
```

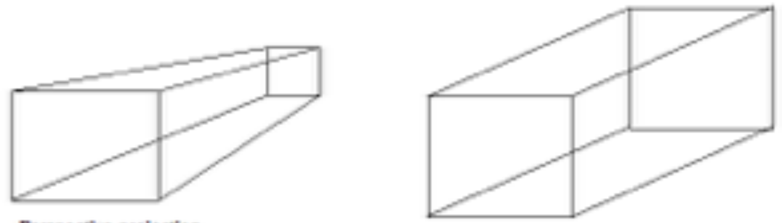
Perspective - Objects near to the front of the volume appear their actual size, while farther objects appear smaller.

It's not too often you'll need to change these parameters, but if you do, altering the field of view ("fov") tends to have the effect of zooming objects in and out (as the viewing volume grows and shrinks) and changing the aspect ratio can skew the rendering of objects making the appear fatter or skinnier.

Camera

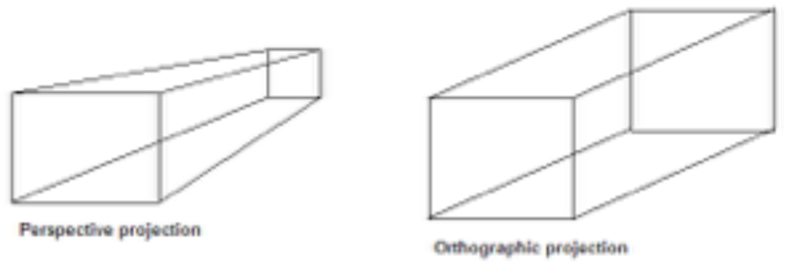
Camera Perspective Ortho

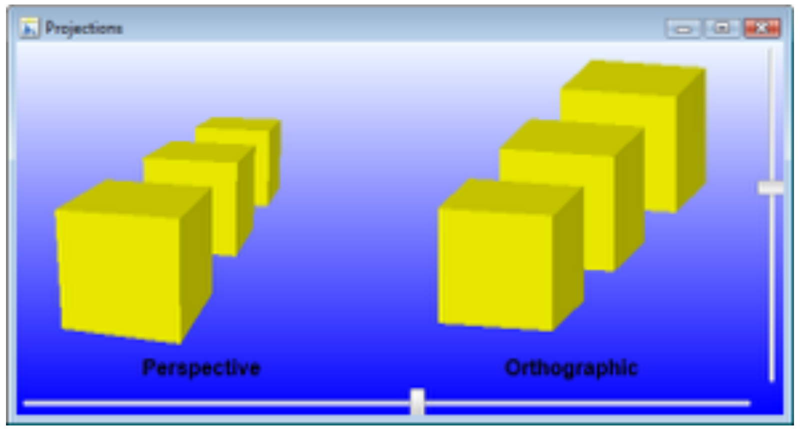
ortho(left,right,bottom,top,near,far)


Perspective projection Orthographic projection

Orthographic - all objects with the same dimension appear the same size, regardless of whether they are near or far from the camera.

Perspective versus Ortho





Orthographic - all objects with the same dimension appear the same size, regardless of whether they are near or far from the camera.