

# Transformations

---

OpenGL

# Learning Outcomes

---

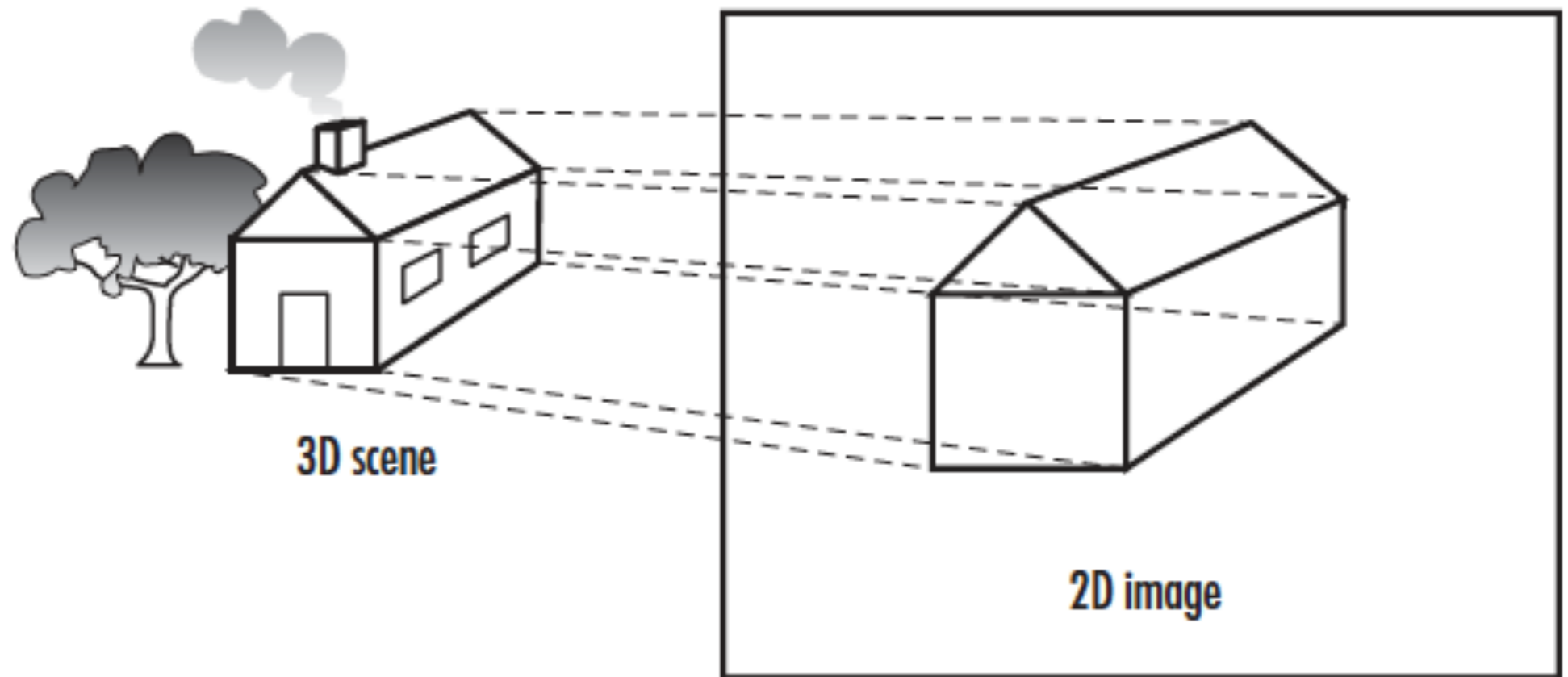
- Be aware of the difference between Orthographic and Perspective projections
- Understand transformations in general and specifically the difference between modeling, viewing and projection transformations.
- Understand the role of eye coordinates in transformation systems and in particular how they relate to viewing transformations.
- Within Modeling transformations, be able to work with rotation, translation and scaling
- Have encountered Model/View duality, and be able to relate it to the transformation terminology discussed above.

# Introduction

---

- We know how to draw points, lines, and various primitives in 3D.
- To turn a collection of shapes into a coherent scene, you must arrange them in relation to one another and to the viewer:
  - Move shapes and objects around within the a coordinate system.
  - Place and orient your objects within a scene
  - Describe an objects' dimensions (often around the origin) and then transform the objects into the desired position.

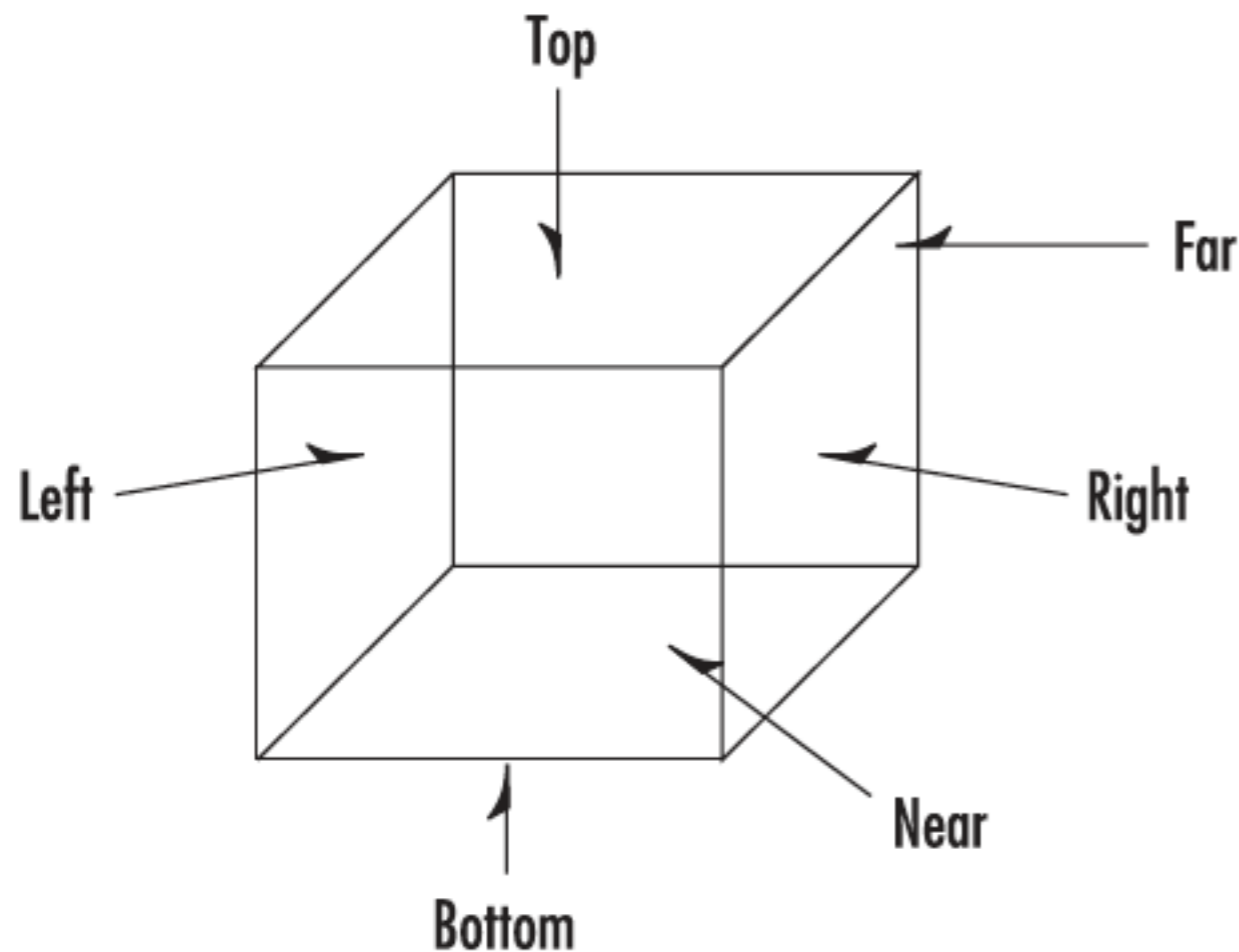
# Projections: Getting 3D to 2D



- Pixels on a screen have only two dimensions.
- The 3D coordinates used to create geometry are flattened or projected onto a 2D surface
- A bit like tracing the outlines of some object behind a piece of glass with a marker.
- By specifying the projection, we specify the viewing volume to be displayed in a window and how it should be transformed.

# Orthographic Projections

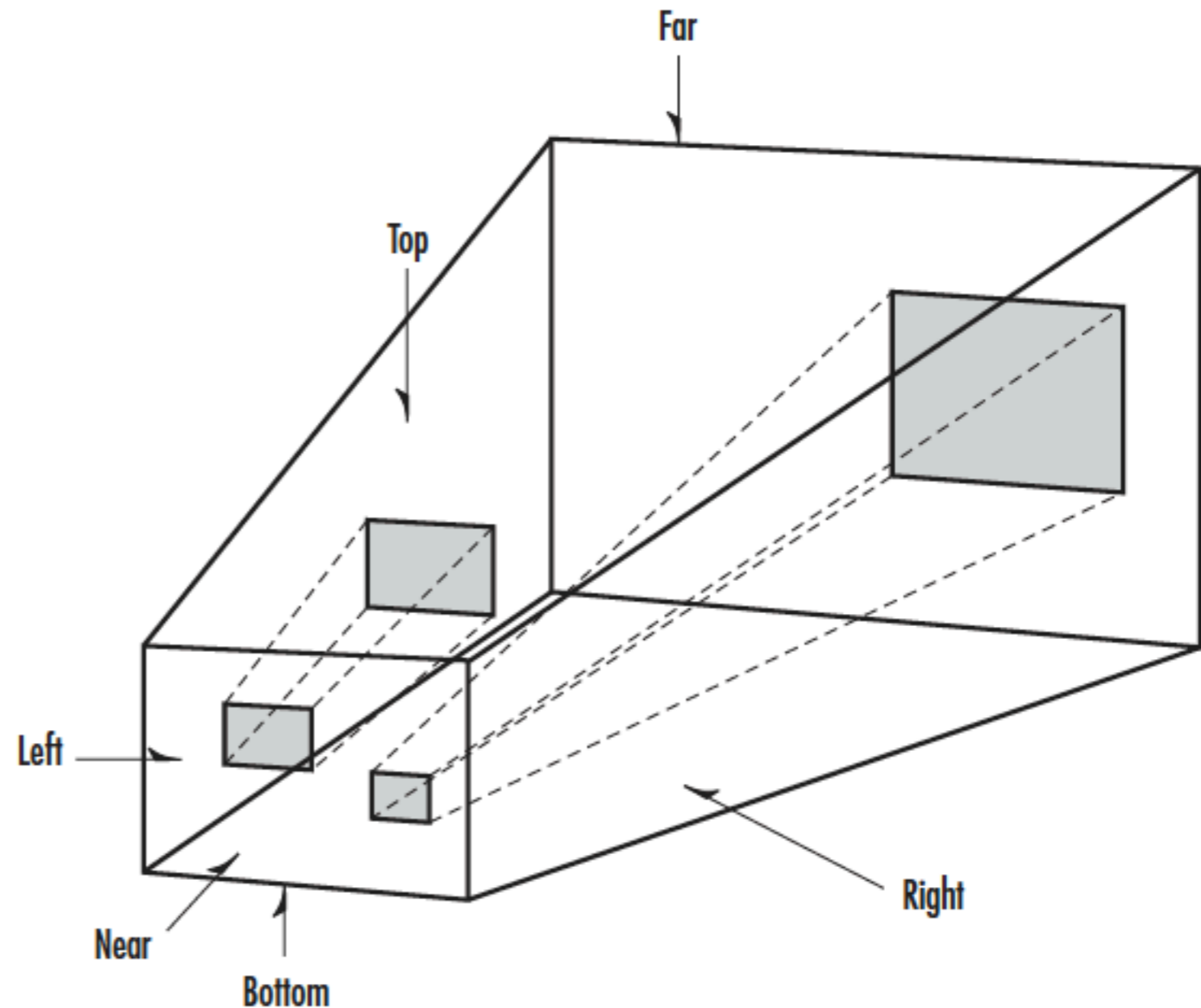
---



- This projection is generated by specifying a square or rectangular viewing volume. Anything outside this volume is not drawn.
- Furthermore, all objects that have the same dimensions appear the same size, regardless of whether they are far away or nearby.
- This type of projection is most often used in architectural design, computer-aided design (CAD), or 2D graphs.
- Specify the viewing volume in an orthographic projection by specifying the far, near, left, right, top, and bottom clipping planes.
- Objects and figures that you place within this viewing volume are then projected (taking into account their orientation) to a 2D image that appears on your screen

# Perspective Projections

- This projection adds the effect that distant objects appear smaller than nearby objects.
- The viewing volume is something like a pyramid with the top shaved off.
- The remaining shape is called the frustum.
- Objects nearer to the front of the viewing volume appear close to their original size, but objects near the back of the volume shrink as they are projected to the front of the volume.
- This type of projection gives the most realism for simulation and 3D animation



# Understanding Transformations

---

- *Projection* : the process of migrating 3D data down into 2D data
- We refer to the projection whenever we want to describe the type of transformation (orthographic or perspective) that occurs during projection.
- However, projection is only one of the types of transformations that occur in OpenGL. Transformations can also:
  - rotate
  - move
  - stretch, shrink, and warp

# OpenGL Transformation Terminology

---

- Three types of geometric transformations occur between the time you specify your vertices and the time they appear on the screen:
  - viewing,
  - modeling
  - projection.

---

<b>Transformation</b>	<b>Use</b>
Viewing	Specifies the location of the viewer or camera
Modeling	Moves objects around the scene
Modelview	Describes the duality of viewing and modeling transformations
Projection	Sizes and reshapes the viewing volume.
Viewport	A pseudo-transformation that scales the final output to the window

---



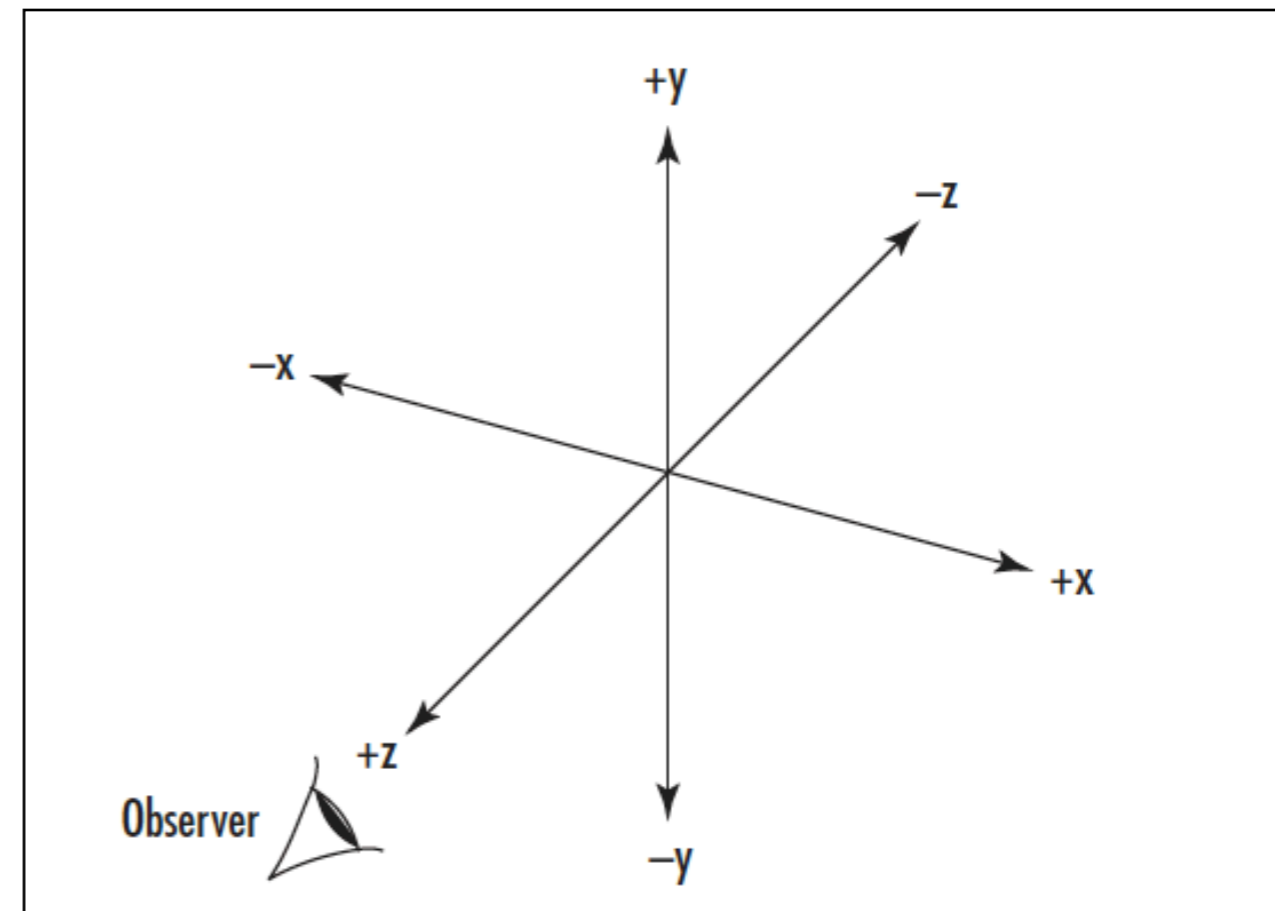
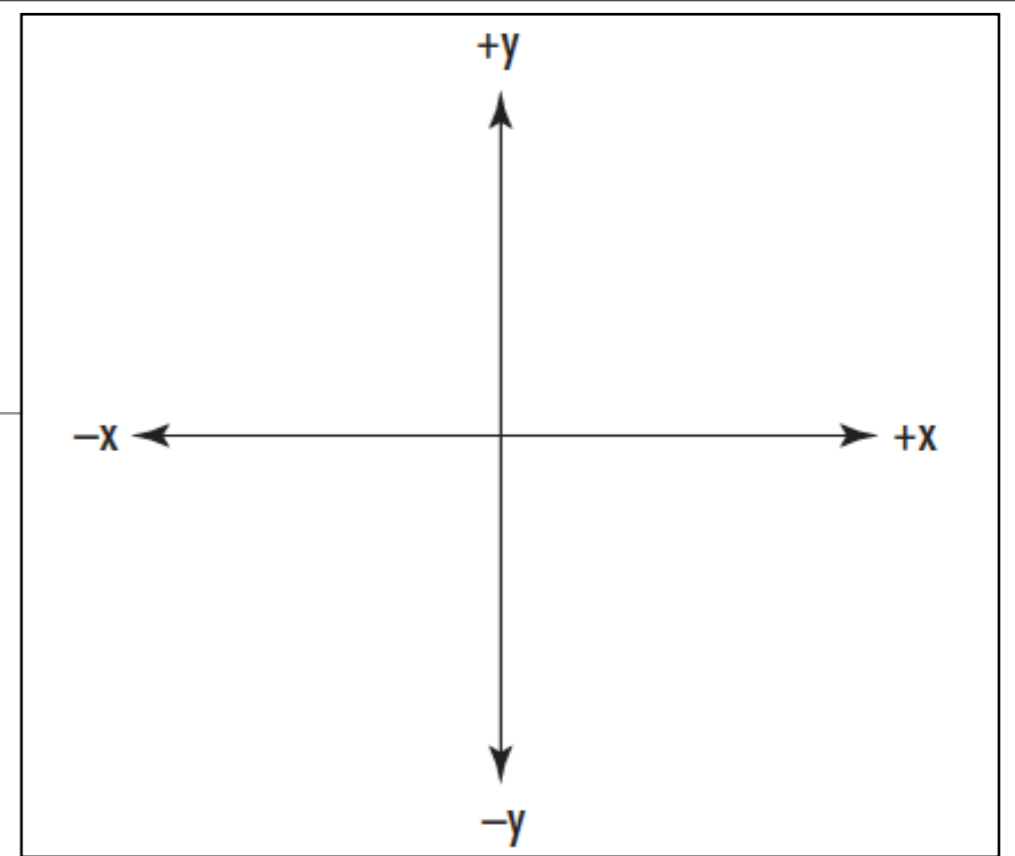
# Eye Coordinates

---

- Eye coordinates are from the viewpoint of the observer, regardless of any transformations that may occur;
- Think of them as “absolute” screen coordinates.
- Represent a virtual fixed coordinate system that is used as a common frame of reference.
- Most transformations are described in terms of their effects relative to the eye coordinate system.

# Eye Coordinates

- The eye coordinates are represented as seen by the observer of the scene (that is, perpendicular to the monitor)
- Positive  $x$  and  $y$  are pointed right and up, respectively, from the viewer's perspective.
- Positive  $z$  travels away from the origin toward the user, and negative  $z$  values travel farther away from the viewpoint into the screen.



# Viewing Transformations

---

Place the point of observation anywhere you want and look in any direction - like placing and pointing a camera at the scene.

- The viewing transformation is the first to be applied to a scene - used to determine the vantage point of the scene.
- By default, the point of observation in a perspective projection is at the origin (0,0,0) looking down the negative z-axis (“into” the monitor screen).
- This point of observation is moved relative to the eye coordinate system to provide a specific vantage point.
- When the point of observation is located at the origin, as in a perspective projection, objects drawn with positive z values are behind the observer.
- In an orthographic projection, however, the viewer is assumed to be infinitely far away on the positive Z axis, and can see everything within the viewing volume.

# Modeling Transformations

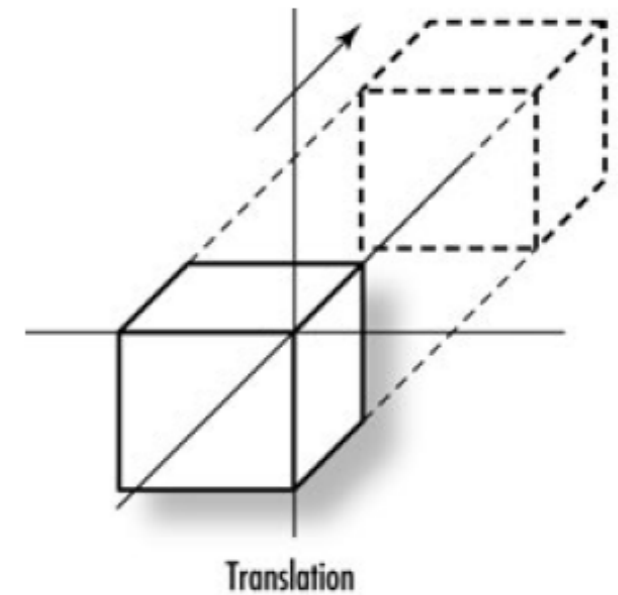
- Used to manipulate your model and the particular objects within it - move objects into place, rotate them, and scale them.
- Three of the most common modeling transformations:

1. Translation - an object is moved along a given axis.

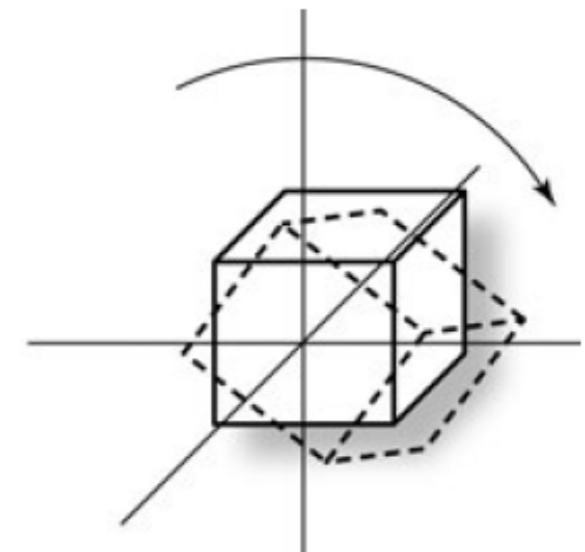
2. Rotation - an object is rotated about one of the axes.

3. Scaling - dimensions of the object are increased or decreased by a specified amount

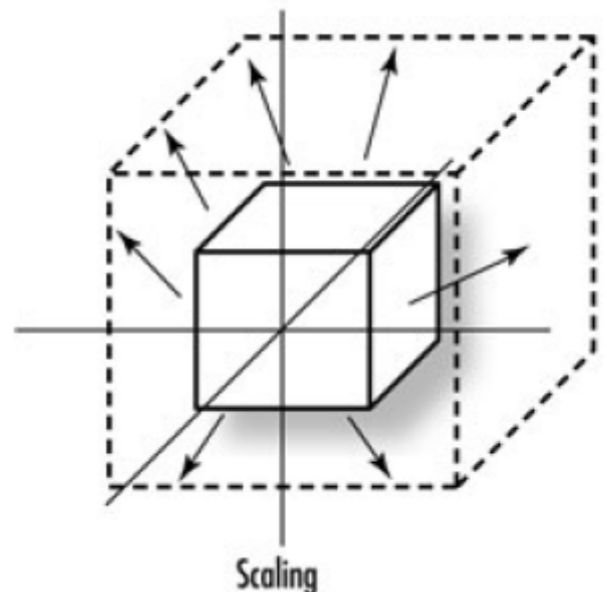
(Scaling can occur non uniformly - the various dimensions can be scaled by different amounts - to stretch and shrink objects)



Translation

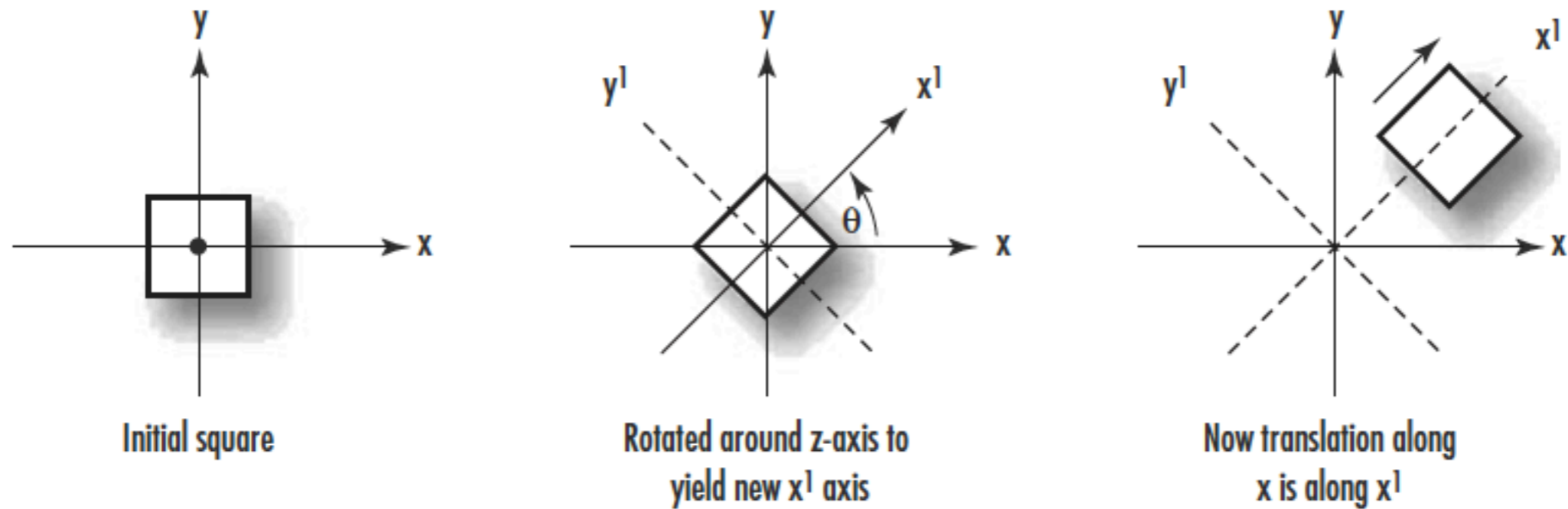


Rotation



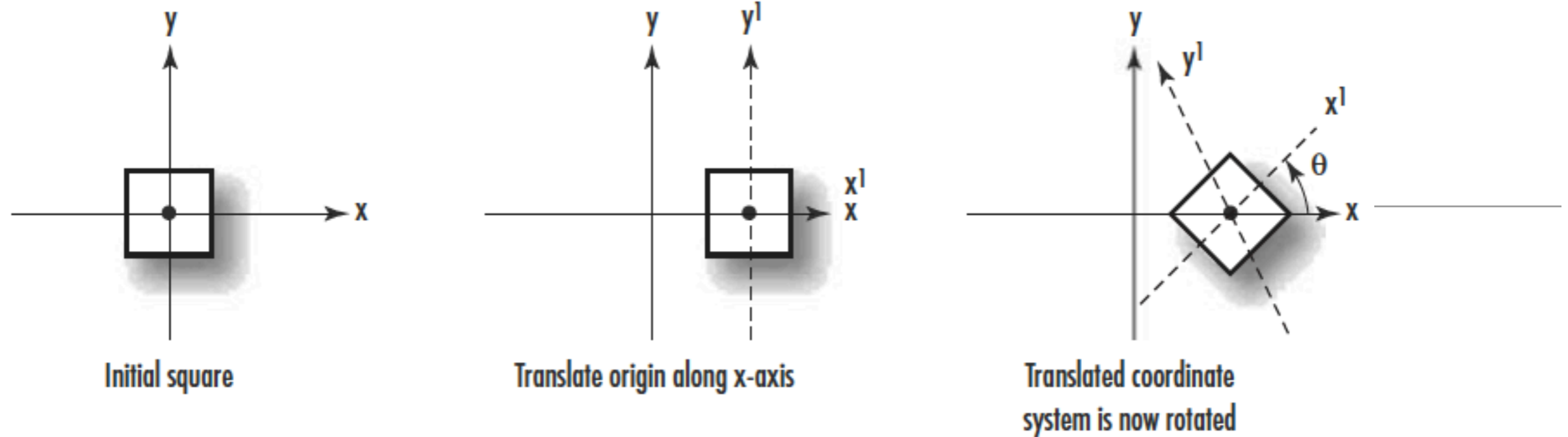
Scaling

- The final appearance of the scene can depend greatly on the order in which the modeling transformations are applied - particularly for translation and rotation.



the progression of a square rotated first about the  $z$ -axis and then translated down the newly transformed  $x$ -axis

The square is rotated with respect to the origin first



square is first translated down the x-axis and then rotated around the z-axis

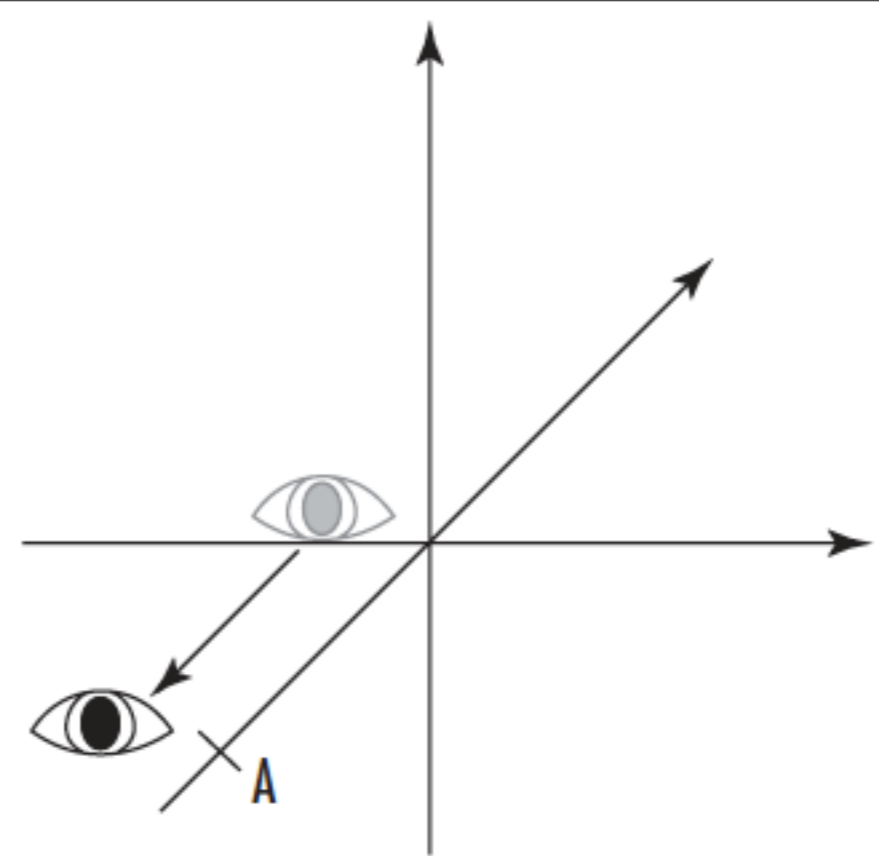
The difference in the final dispositions of the square occurs because each transformation is performed with respect to the last transformation performed

after the square is translated, the rotation is performed around the newly translated origin

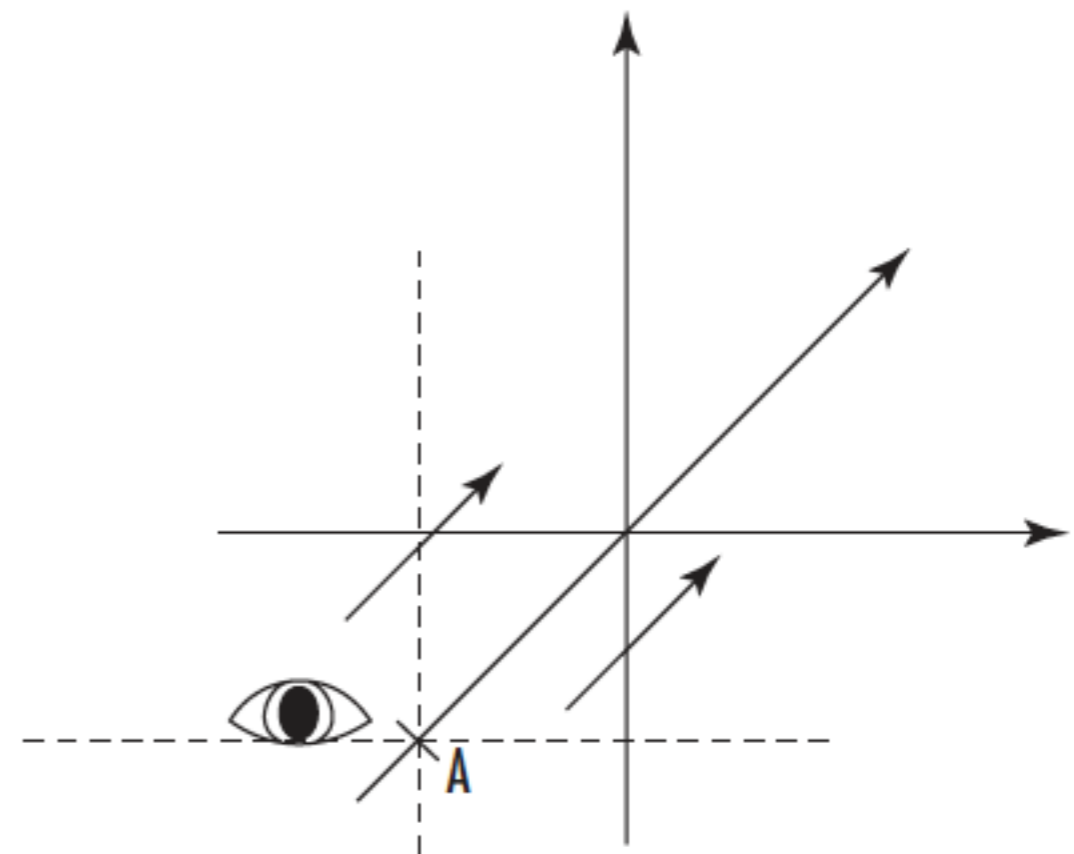
# Viewing Transformations

---

- The viewing transformation is essentially a modeling transformation that you apply to a virtual object (the viewer) before drawing objects.
- New transformations are repeatedly specified as you place more objects in the scene.
- By convention, the initial transformation provides a reference from which all other transformations are based.



Moving the observer



Moving the coordinate system

# The Modelview Duality

---

- The viewing and modeling transformations are the same in terms of their internal effects as well as their effects on the final appearance of the scene.
- The distinction between the two is made purely as a convenience for the programmer: there is no real difference visually between moving an object backward and moving the reference system forward.
- The viewing transformation is a modeling-like transformation that is applied to the entire scene, where objects in the scene will often each have their own individual model transformation, applied after the viewing transformation.
- The term modelview indicates that these two transformations are combined in the transformation pipeline into a single matrix—the modelview matrix.



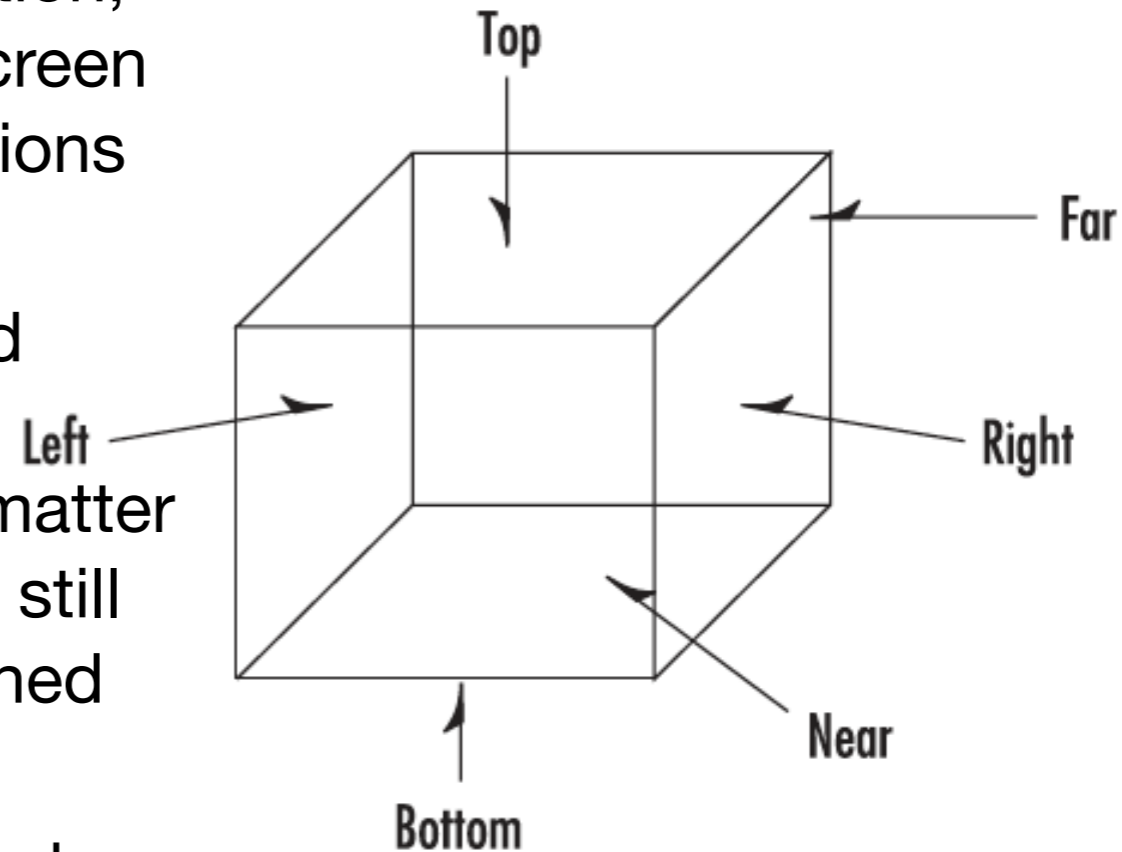
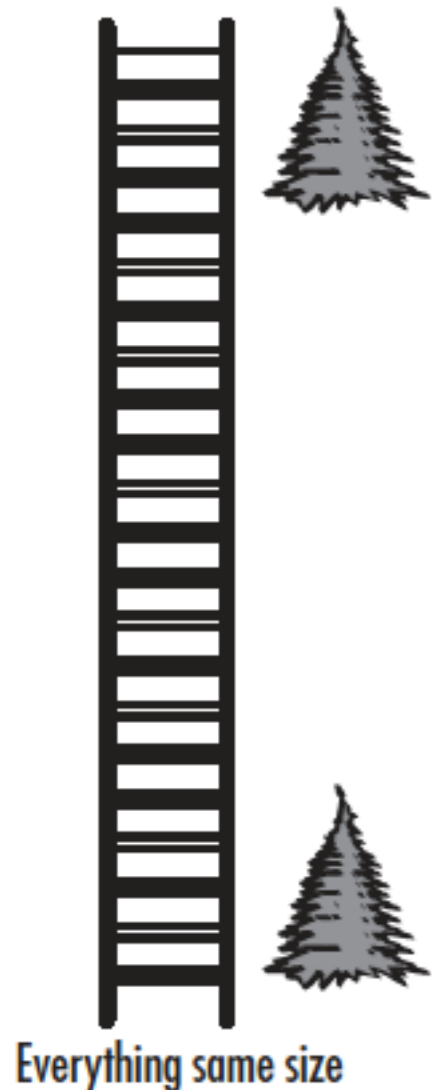
# Projection Transformations

---

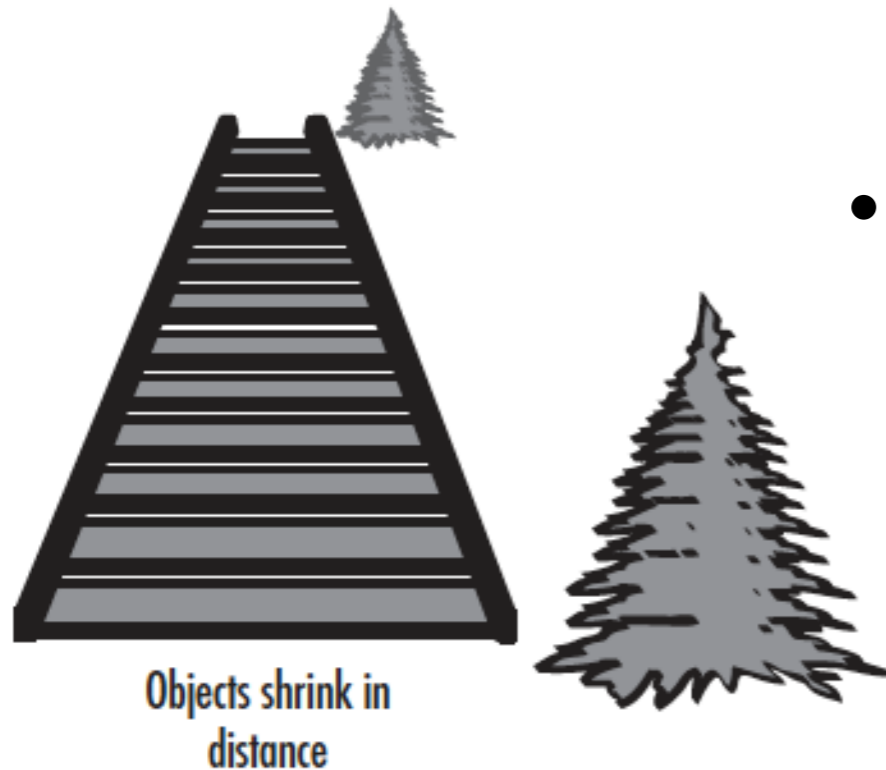
- The clipping planes are plane equations in 3D space that OpenGL uses to determine whether geometry can be seen by the viewer
- The Projection transformation is applied after the modelview transformation - it defines the viewing volume and establishes clipping planes.
- It specifies how a finished scene (after all the modeling is done) is projected to the final image on the screen.

# Orthographic Projection Transformation

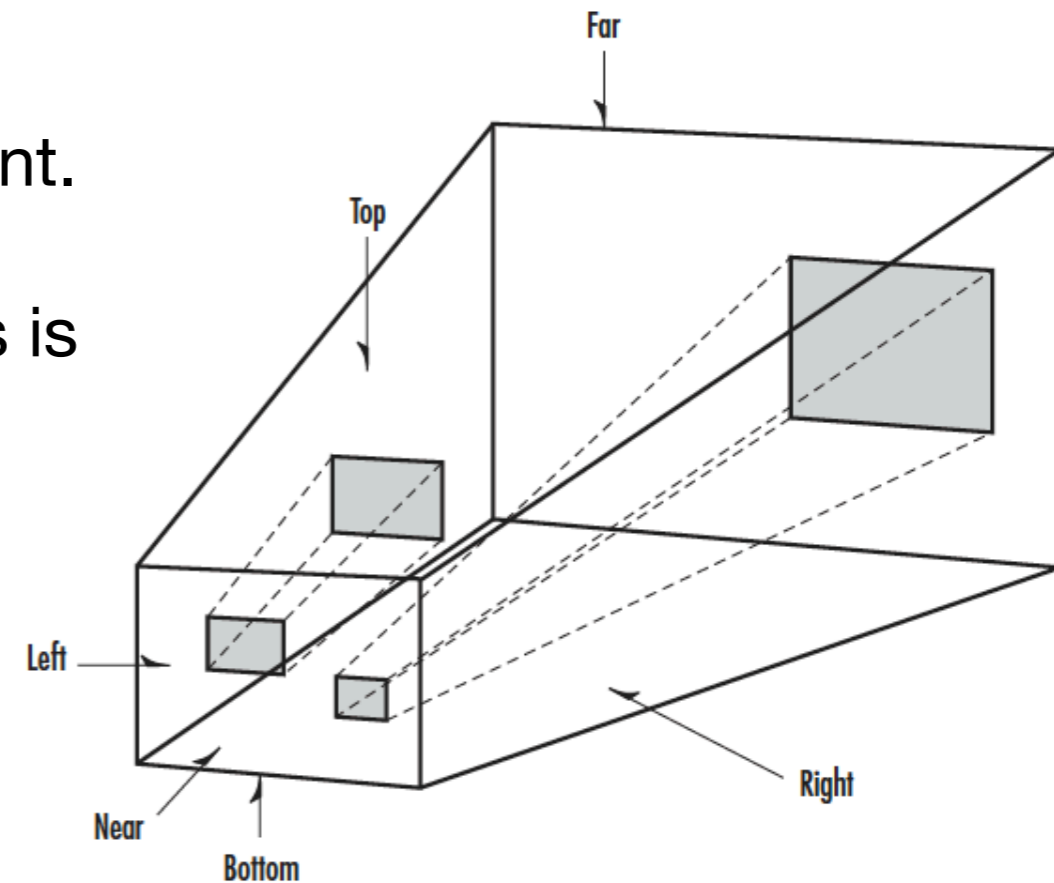
- Orthographic, or parallel, projection, all the polygons are drawn onscreen with exactly the relative dimensions specified.
- Lines and polygons are mapped directly to the 2D screen using parallel lines, which means no matter how far away something is, it is still drawn the same size, just flattened against the screen
- Used for 2D drawing purposes where you want an exact correspondence between pixels and drawing units
- Also for 3D renderings when the depth of the rendering has a very small depth in comparison to the distance from the viewpoint



# Perspective Projection Transformation



- A perspective projection shows scenes more as they appear in real life instead of as a blue-print.
- The trademark of perspective projections is foreshortening, which makes distant objects appear smaller than nearby objects of the same size.
- Lines in 3D space that might be parallel do not always appear parallel to the viewer.
- Most typical for 3D graphics



# Viewport Transformations

---

- Mapping to physical window coordinates - the last transformation.
- Usually, a one-to-one correspondence exists between the color buffer and window pixels, but this is not always strictly the case.
- In some circumstances, the viewport transformation remaps what are called “normalized” device coordinates to window coordinates