OpenGL Fundamentals

Coordinates, Clipping, Viewports & Projections

Learning Outcomes

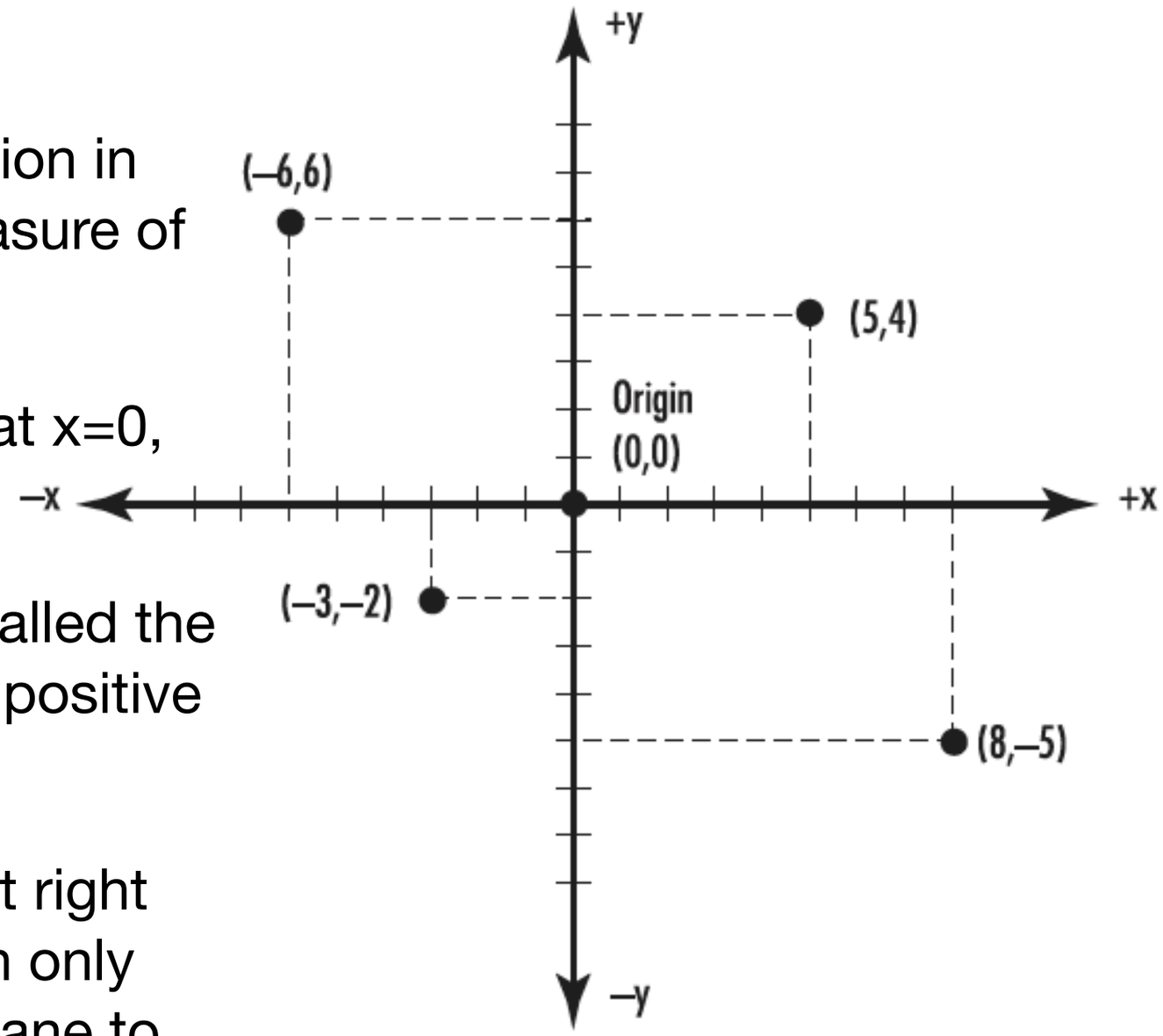
- Review 2D Cartesian co-ordinate system and its relevance to graphics
- Understand clipping and clipping regions
- Be able to distinguish between clipping regions and viewports and how they may be mapped
- Relate all of this to the glViewPort function.
- Visualise the 3D Cartesian co-ordinate system.
- Have encountered the following terms:
    - Vertex
    - Orthographic Projection
    - Perspective Projection

Coordinate Systems

- Before you can specify an object's location and size, you need a frame of reference to measure and locate against.
- When you draw lines or plot points on a simple flat computer screen, you specify a position in terms of a row and column.
- For example, a standard VGA screen has 640 pixels from left to right and 480 pixels from top to bottom.
- To specify a point in the middle of the screen, you specify that a point should be plotted at (320,240)—that is, 320 pixels from the left of the screen and 240 pixels down from the top of the screen.
- In OpenGL when you create a window to draw in, you must also specify the coordinate system you want to use and how to map the specified coordinates into physical screen pixels.

2D Cartesian Coordinates

- Cartesian coordinates are specified by an x coordinate and a y coordinate.
- The x coordinate is a measure of position in the horizontal direction, and y is a measure of position in the vertical direction.
- The origin of the Cartesian system is at x=0, y=0.
- The x and y lines with tick marks are called the axes and can extend from negative to positive infinity
- Two axes (or two lines) that intersect at right angles define a plane. In a system with only two axes, there is naturally only one plane to draw on.

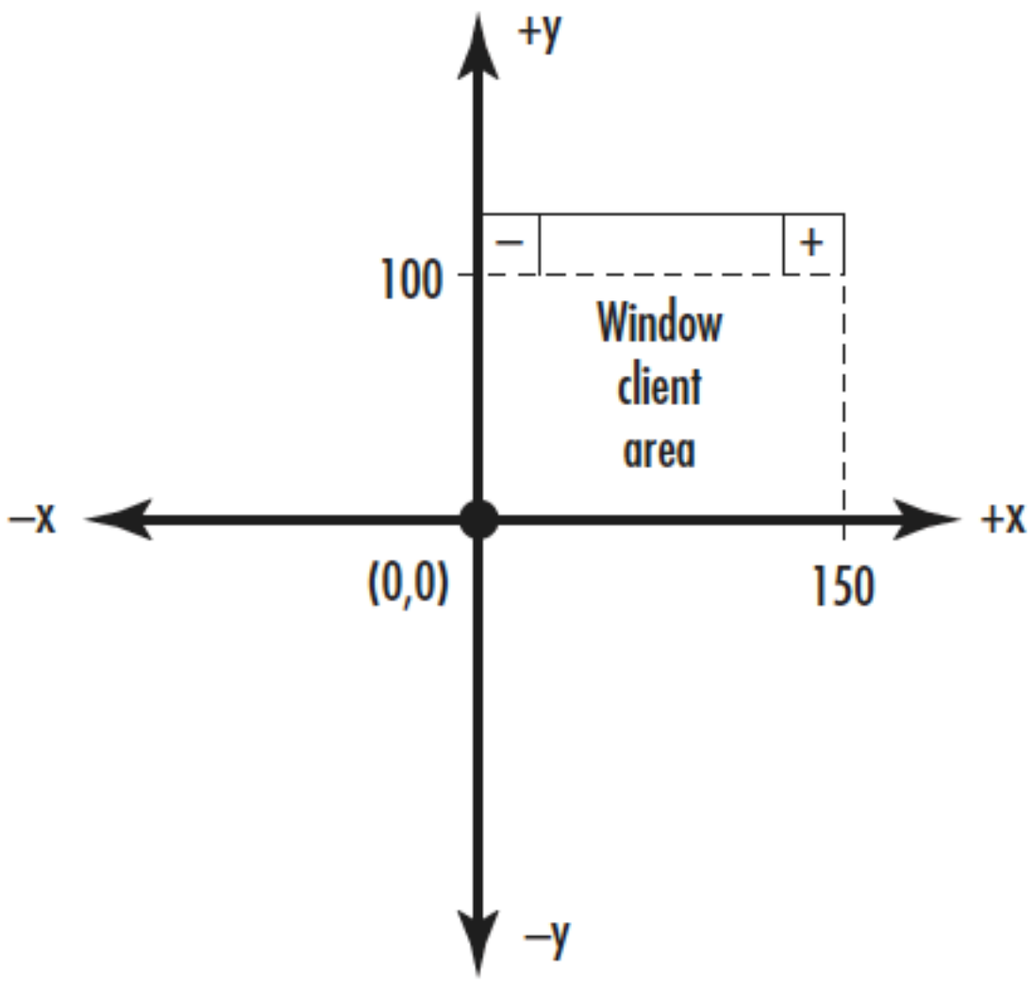


Coordinate Clipping

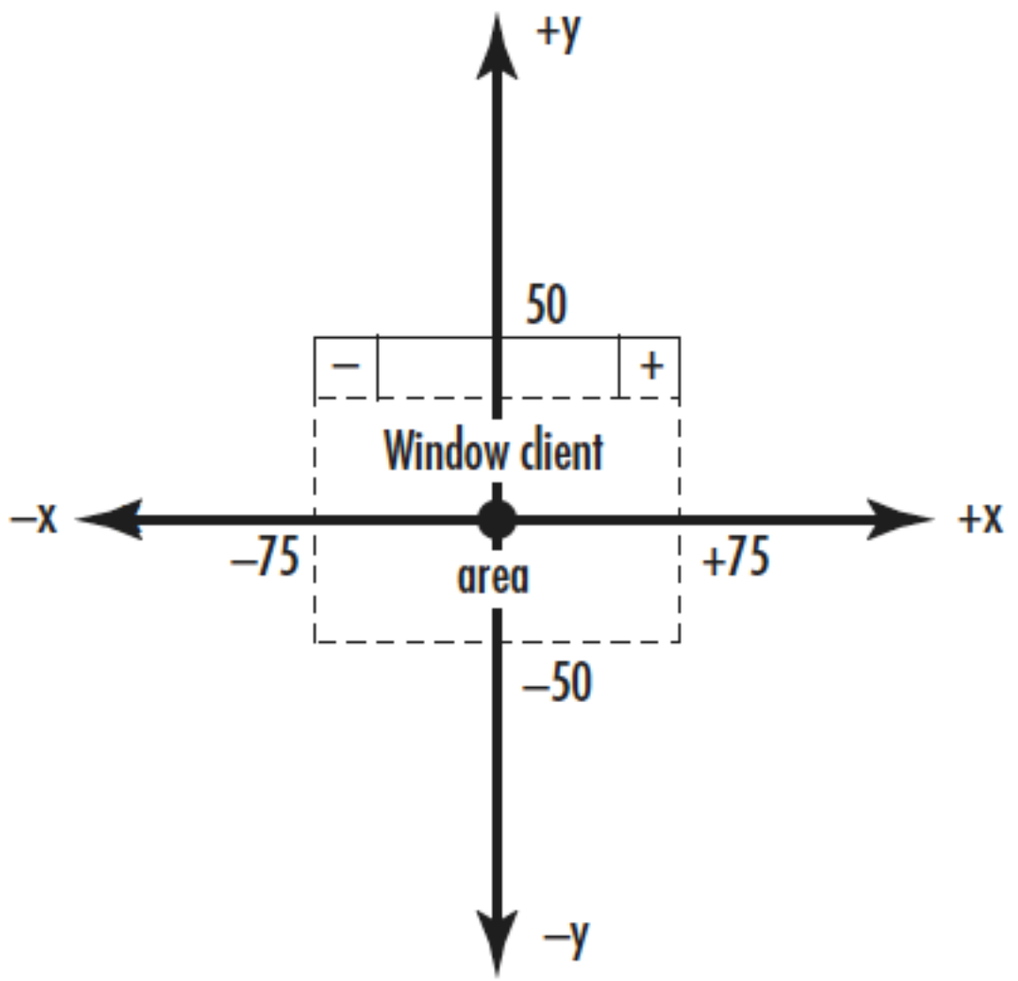
- A window is measured physically in terms of pixels.
- OpenGL must be instructed how to translate specified coordinate pairs into screen coordinates.
- Do this by specifying the region of Cartesian space that occupies the window.
- This region is known as the clipping region.
- In two-dimensional space, the clipping region is the minimum and maximum x and y values that are inside the window.

Two Clipping Regions

- x coordinates in the window range left to right from 0 to +150
- y coordinates range bottom to top from 0 to +100.
- A point in the middle of the screen would be represented as (75,50).

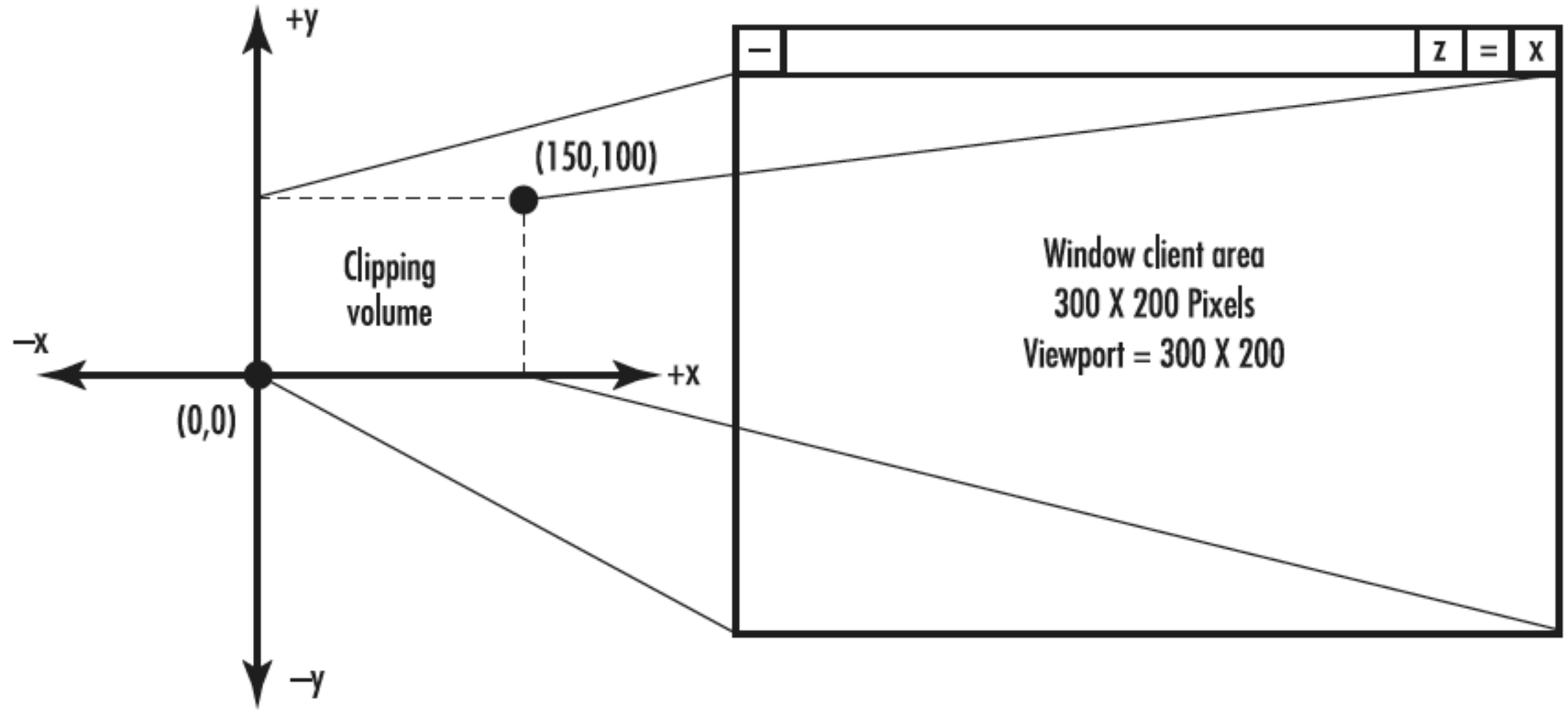


- x coordinates ranging left to right from –75 to +75
- y coordinates ranging bottom to top from –50 to +50. In this example,
- a point in the middle of the screen would be at the origin (0,0).



Viewports

- Rarely will the clipping area width and height exactly match the width and height of the window in pixels.
- The coordinate system must therefore be mapped from logical Cartesian coordinates to physical screen pixel coordinates.
- This mapping is specified by a setting known as the viewport.
- The viewport is the region within the window’s client area that is used for drawing the clipping area.
- The viewport simply maps the clipping area to a region of the window.
- Usually, the viewport is defined as the entire window, but this is not strictly necessary; for instance, you might want to draw only in the lower half of the window.

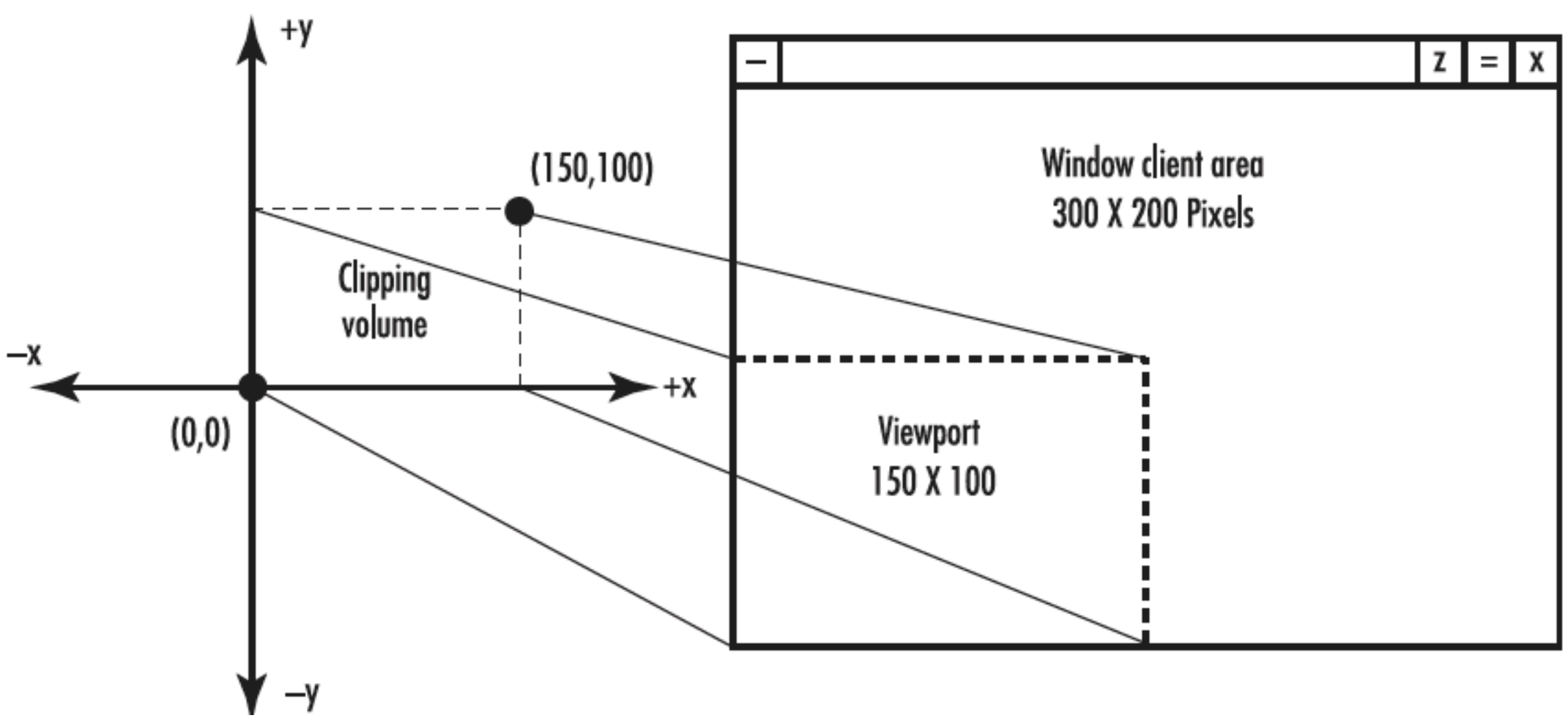


- A Window measuring 300×200 pixels with the viewport defined as the entire client area.
- Clipping area set to 0 to 150 along the x-axis and 0 to 100 along the y-axis, the logical coordinates would be mapped to a larger screen coordinate system in the viewing window.
- Each increment in the logical coordinate system would be matched by two increments in the physical coordinate system (pixels) of the window.

- A viewport that matches the clipping area.

- The viewing window is still 300×200 pixels, however, and this causes the viewing area to occupy the lower-left side of the window.

- You can use viewports to shrink or enlarge the image inside the window and to display only a portion of the clipping area by setting the viewport to be larger than the window's client area.

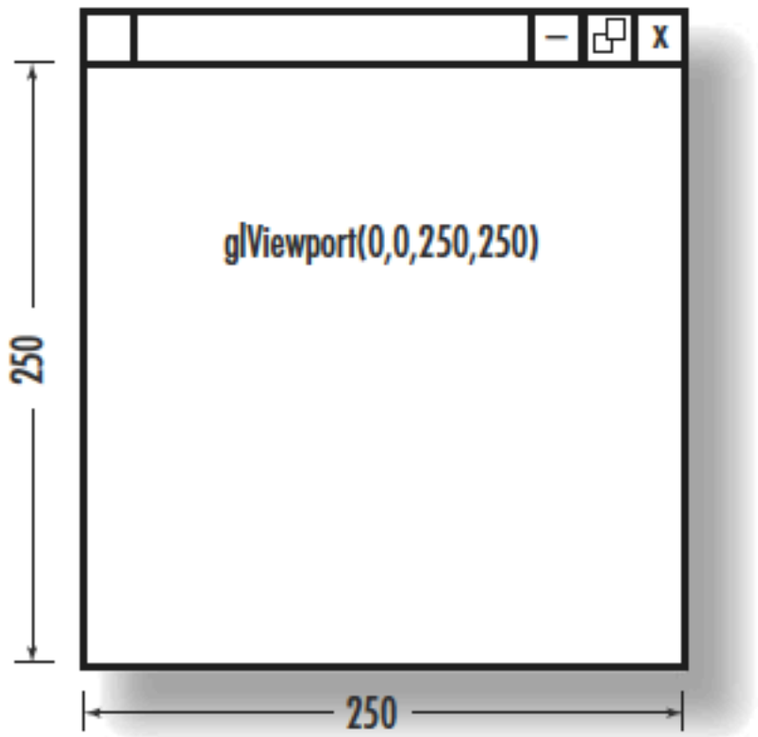


glViewport()

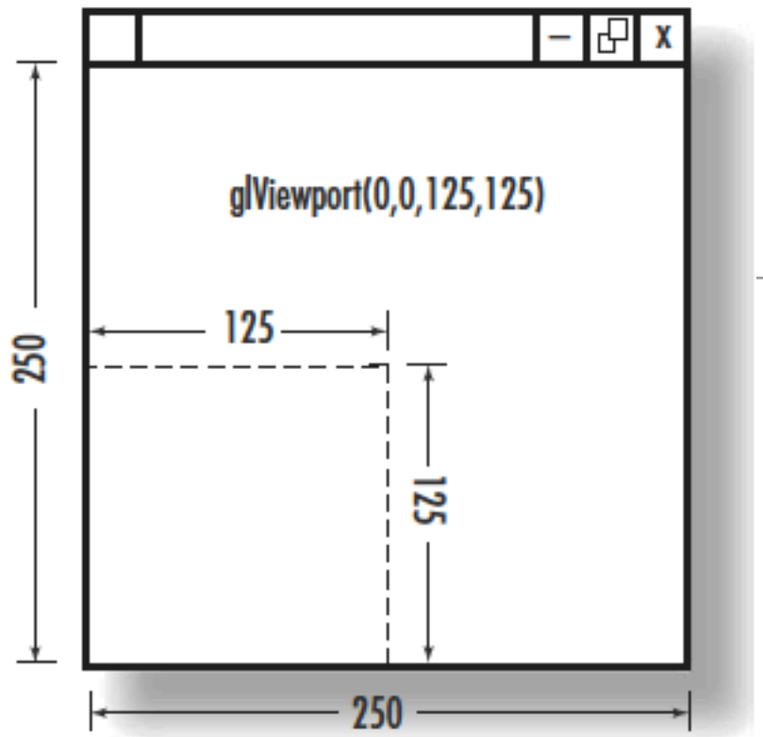
```
void glViewport (GLint x, GLint y, GLsizei width, GLsizei height);
```

```
void changeSize(int w, int h)
{
  GLfloat aspectRatio;
  //...

  glViewport(0, 0, w, h);

  //...
}
```

- The viewport defines the area within the window in actual screen coordinates that OpenGL can use to draw in.
- The x and y parameters specify the lower-left corner of the viewport within the window.
- Width and height parameters specify these dimensions in pixels.
- Usually, x and y are both 0, but you can use viewports to render more than one drawing in different areas of a window.



Window and viewport are same



Viewport 1/2 size of window

- The viewport defines the area within the window in actual screen coordinates that OpenGL can use to draw in.

- The current clipping volume is then mapped to the new viewport.

- If you specify a viewport that is smaller than the window coordinates, the rendering is scaled smaller

The Vertex—A Position in Space

- In both2D and 3D, when an object is drawn, it is composed of several smaller shapes called primitives.

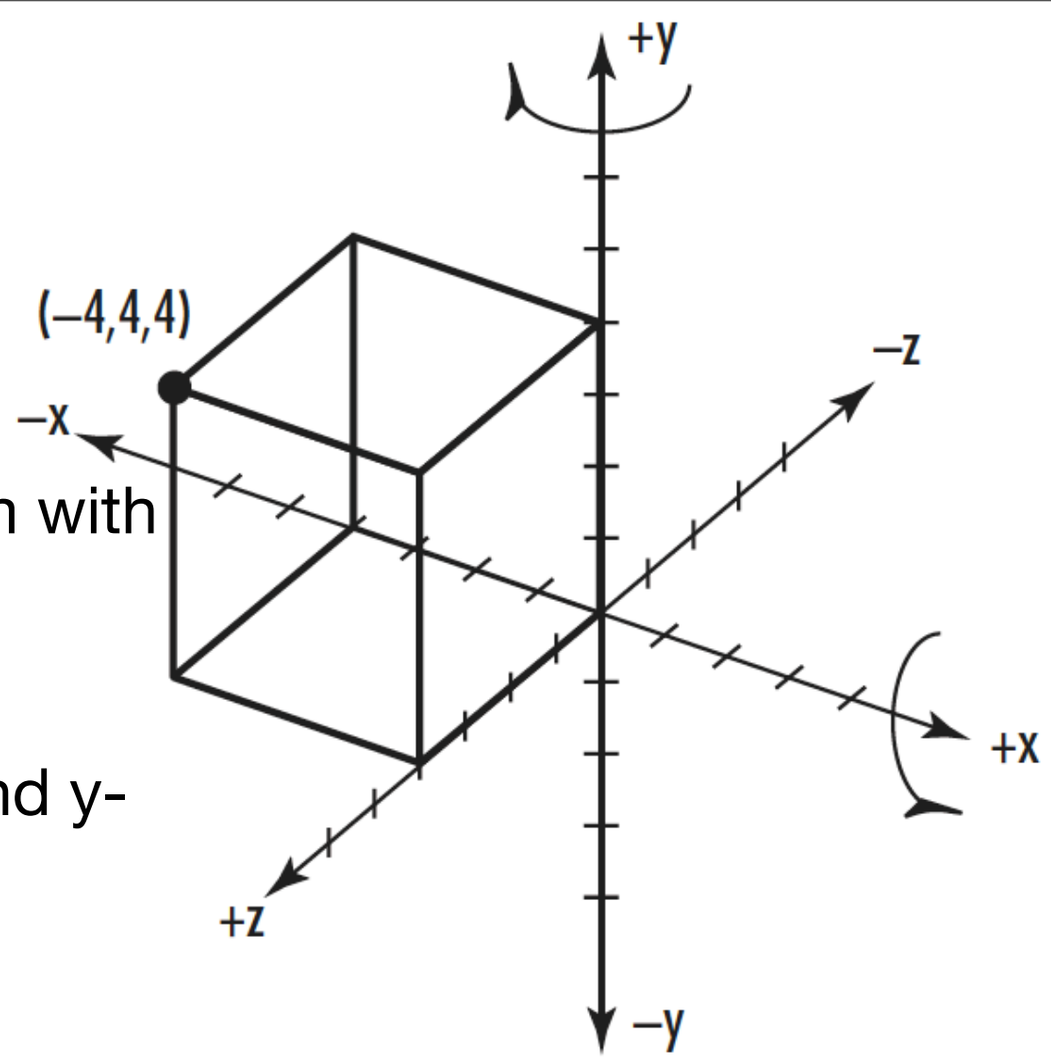
- Primitives are one- or two-dimensional entities or surfaces such as points, lines, and polygons (a flat, multisided shape) that are assembled in 3D space to create 3D objects.

- For example, a three-dimensional cube consists of six two-dimensional squares, each placed on a separate face. Each corner of the square (or of any primitive) is called a vertex.

- These vertices are then specified to occupy a particular coordinate in 3D space.

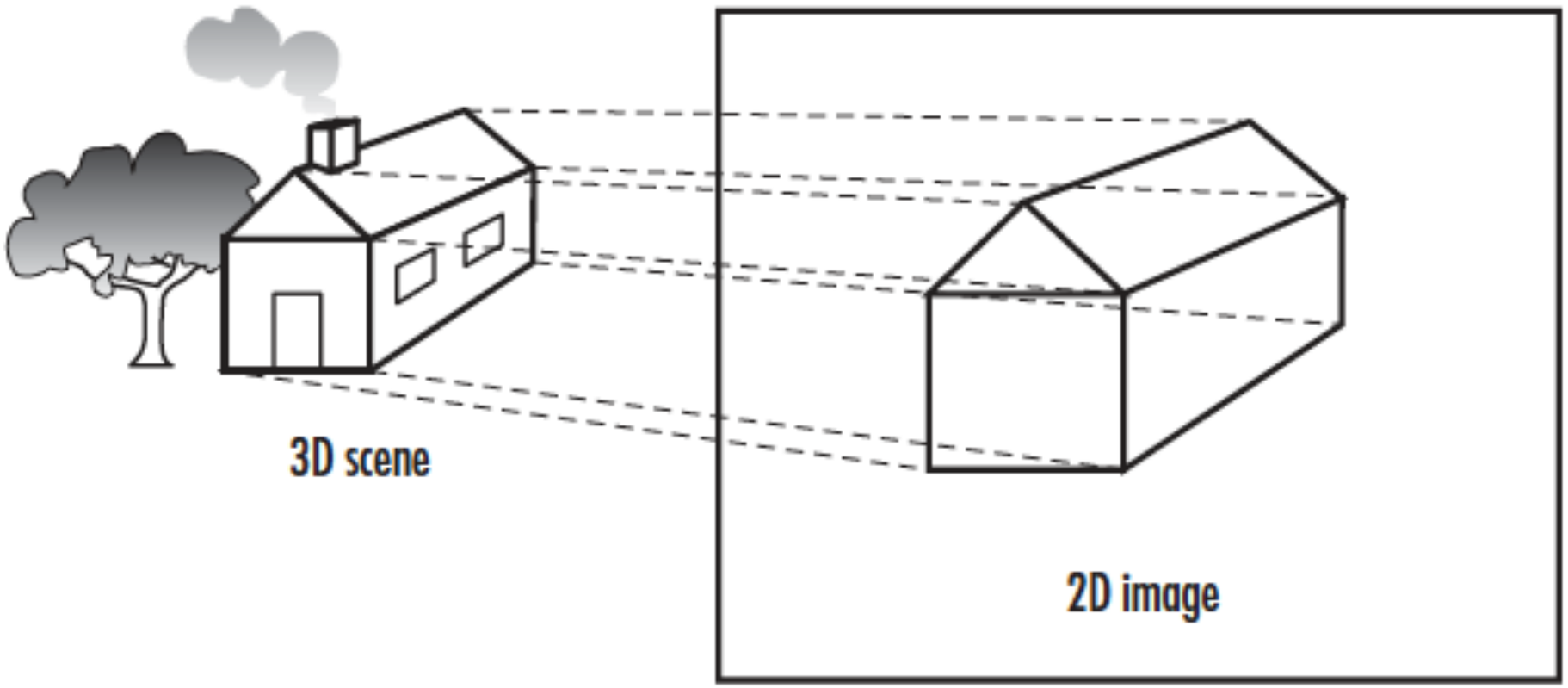
- A vertex is essentially a coordinate in 2D or 3D space.

3D Cartesian Coordinates



- 3D extends the Cartesian coordinate system with a new axis, z.
- The z-axis is perpendicular to both the x- and y-axes.
- It represents a line drawn perpendicularly from the center of the screen heading toward the viewer.
- Can specify a position in three-dimensional space with three coordinates: x, y, and z.
- Example point (–4,4,4)

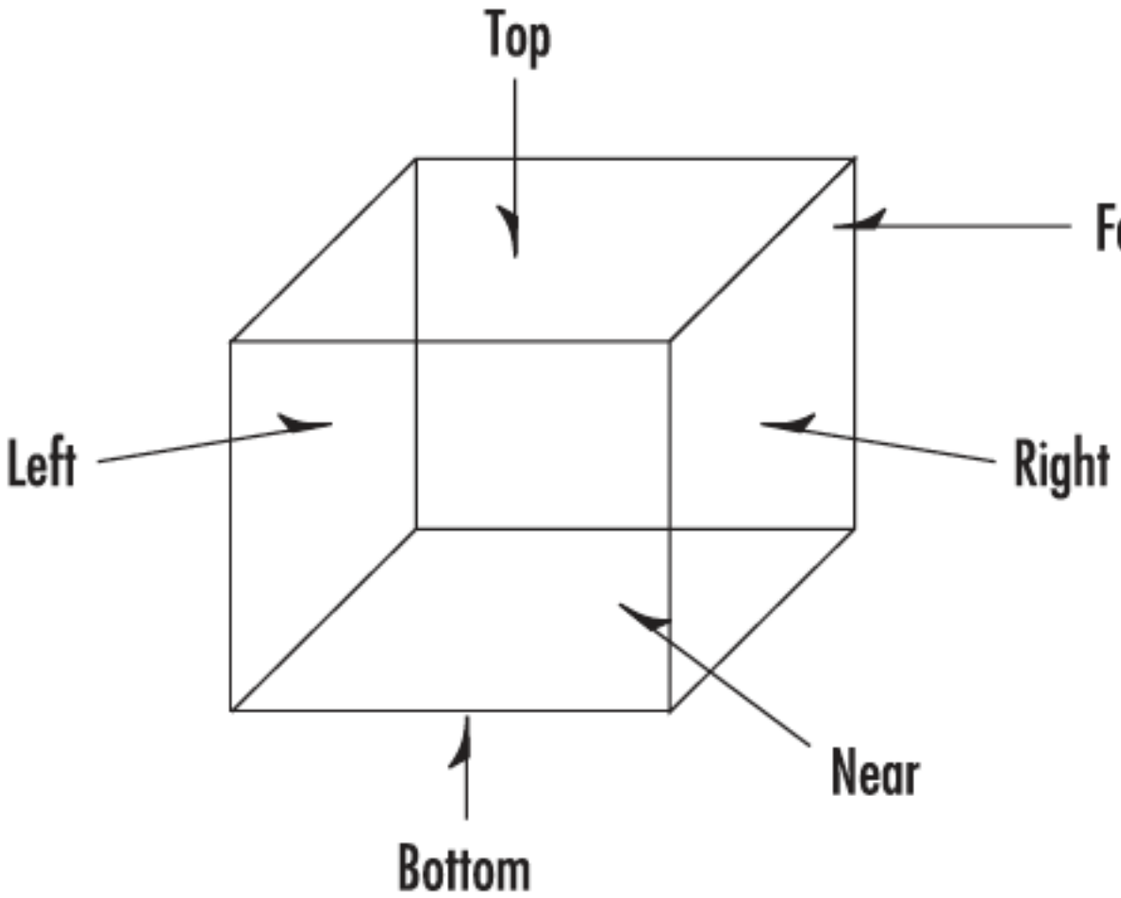
Projections: Getting 3D to 2D



- 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 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