Coordinate Spaces and the Viewport

To successfully complete *Super Asteroids*, you need to be familiar with several coordinate spaces.

1. Screen Space

Almost every device with a screen uses the same coordinate system. This system places the origin (X=0, Y=0) at the top left corner of the screen. X increases to the right and Y increases moving downward. This means X decreases to the left and Y decreases going up.

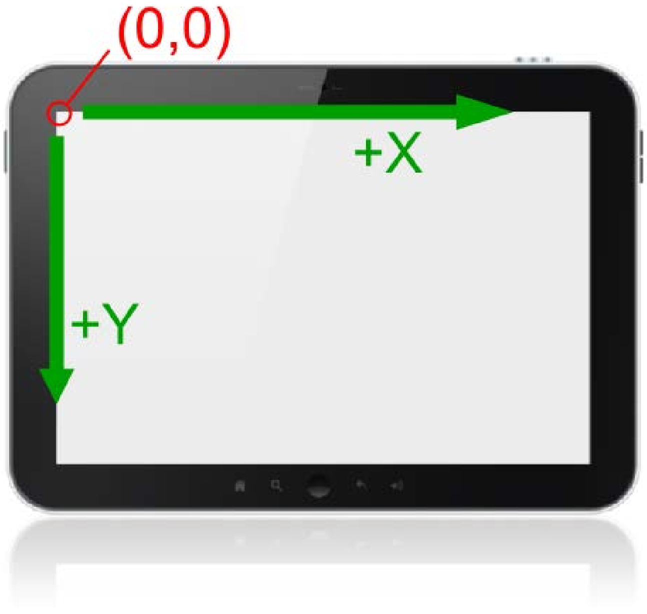


Figure 1 - Screen space

2. World Space

The “world” consists of all objects on the current level of the game (ship, asteroids, background objects, projectiles, etc.), regardless of whether they are currently visible on the screen. The “world space” is the coordinate system used to define the positions of the game objects in your program’s internal data structures (i.e., in your “model”). The world space is arranged just like the screen space. The origin is at the top left of the world space. X increases to the right and Y increases going down.

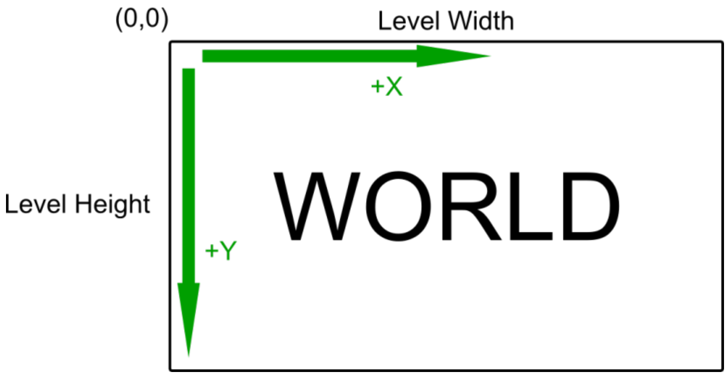


Figure 2 - World space

For 2D games, the world is best visualized as a rectangle. In *Super Asteroids*, the world width and height are always set to the width and height of the current level.

Almost every game element needing to be drawn to the screen has a position (x,y) within the world coordinate system. Even some invisible game elements have world positions. Figure 3 shows several game objects with their approximate world coordinates. Please note that object world positions are best measured from the center of the object. One exception is the *Super Asteroids* viewport. Its position is measured from its top left corner.

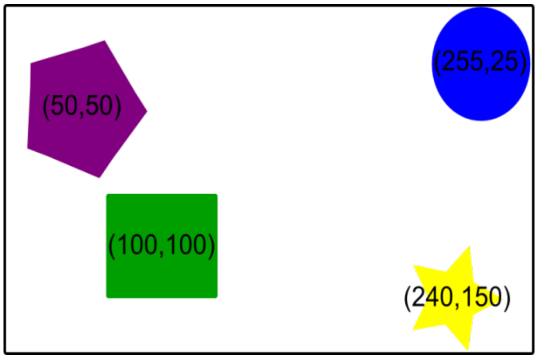


Figure 3 - World space object coordinates

The Viewport

Most game worlds are larger than can be shown on the screen at one time. The viewport defines the portion of the world that is currently visible on the screen. The world and viewport are both depicted by rectangles in a 2D game environment.

The location and size of the viewport determines what is seen on screen. If an object does not intersect the rectangle representing the viewport, it should not be seen on screen. If an object does intersect the viewport rectangle, the object is drawn to the screen by converting its world coordinates to screen coordinates.

For *Super Asteroids*, the viewport has the same dimensions as the screen, and an object's screen coordinates are calculated by subtracting the viewport world position from the object's world position. This calculation is known as a **translation** and looks like this:

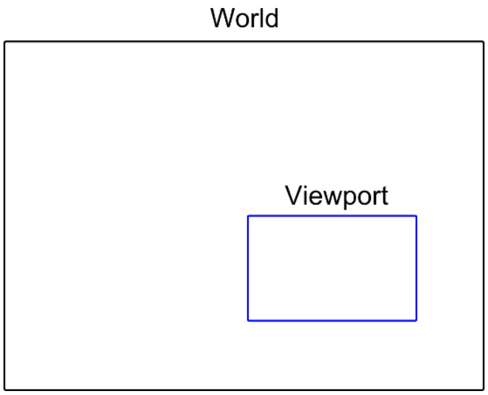
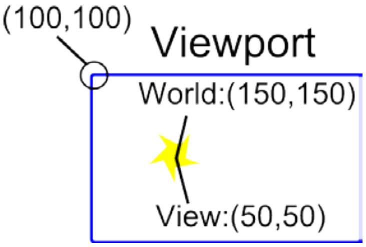


Figure 4 - World and viewport



obj Screen pos = obj World pos – viewport World pos

Figure 5 - World and viewport coordinates

For *Super Asteroids*, your viewport should always have the same dimensions as the current Android view hosting the game. These dimensions can be accessed through the **DrawingHelper** class.

Please note:

The Android view hosting the game in the ship builder has different dimensions compared to the view hosting the game during actual gameplay.