# Coordinate Spaces and the Viewport

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

## 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.

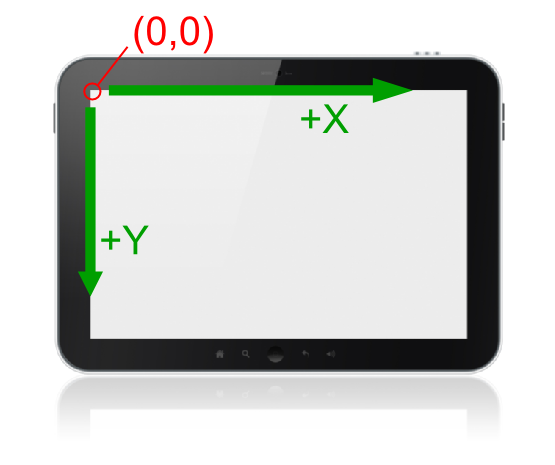


Figure 1Screen space

## World Space

The world space is arranged just like the screen space. X increases to the right and Y increases going down. This space is completely imaginary, so the location of its origin can change depending on the format of the game. In the case of *Super Asteroids*, the origin is at the top left of the world bounds.

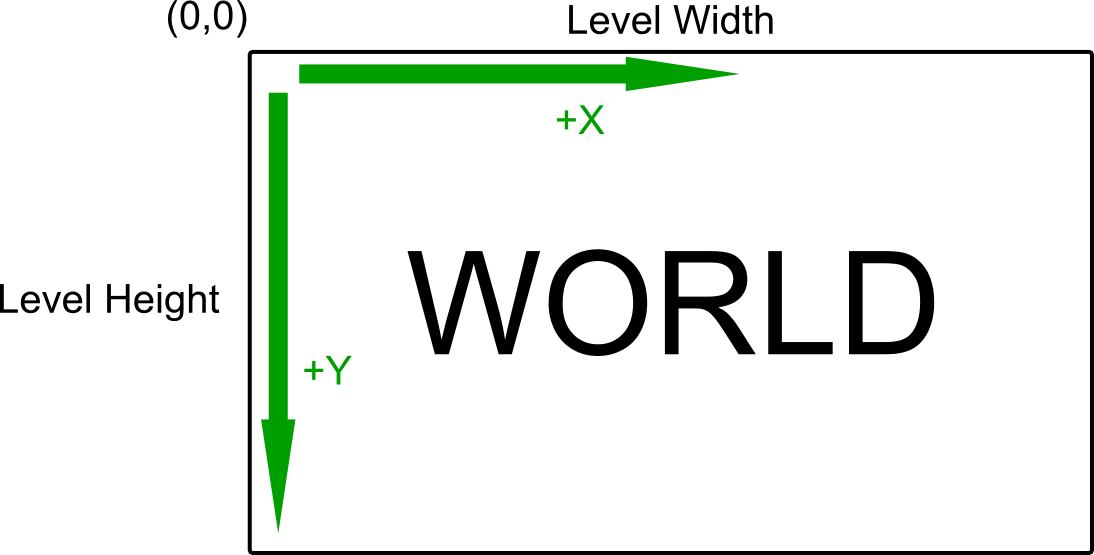
For 2D games, the world is best visualized as a rectangle. In *Super Asteroids*, the world width and height are always set to the width and height of the current level.

Figure 2World space

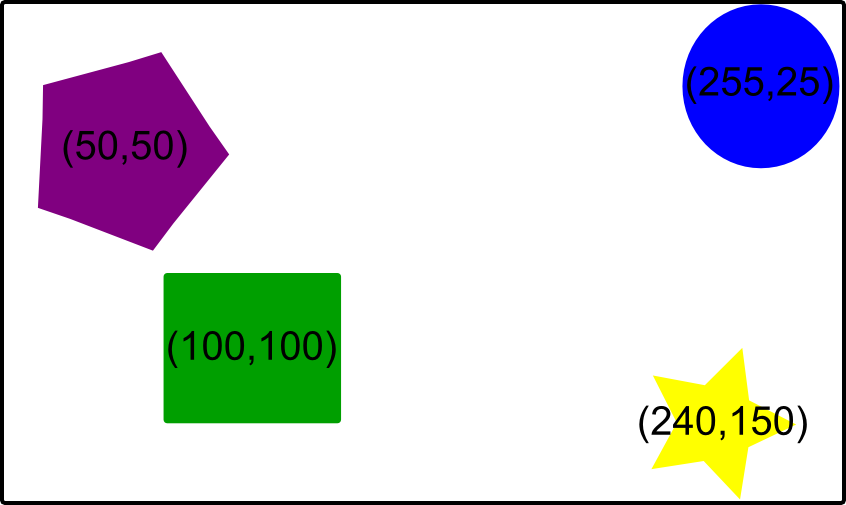
Almost every game element needing to be drawn to the screen has a position (x,y) within the world coordinate system. Even some invisible game elements have world positions. Figure 3 shows several game objects with their approximate world coordinates. Please note that object world positions are best measured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

Figure 3World space object coordinates

## View Space and Viewports

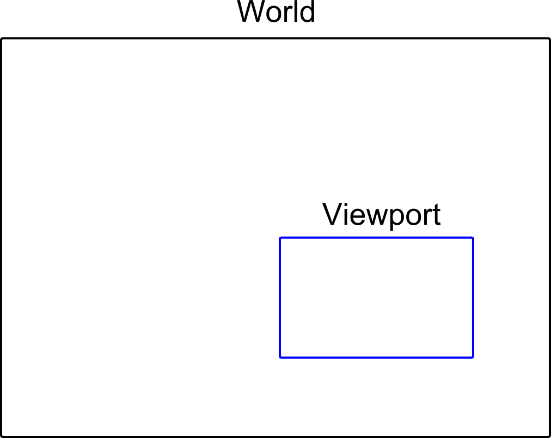
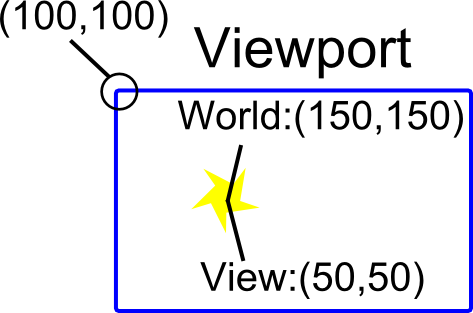
Most game worlds are larger than can be shown on the screen at one time. The view space is the window into the portion of the world to be seen at one time. A game viewport is the component of the game used to simulate the view space. The world and viewport are both depicted by rectangles in the 2D game environment.

Figure 4World and viewport

In essence, 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.

In the simplest case, the viewport has the same dimensions as the screen and the object’s screen coordinates are the same as its view space coordinates and 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:



**obj Screen pos = obj world pos – viewport world pos**

Figure 5 World and viewport coordinates

As you may be aware, Android devices have varying screen sizes. If you were required to create *Super Asteroids* such that its look and feel was the same across all possible screen sizes, you would have to have use a standard viewport size. Luckily, this is not a requirement for this project. Instead, your viewport should always be the same dimension as the current Android view hosting the game. These dimensions can be accessed through the **DrawingHelper** class. Thus, to find the drawing coordinates of a game object, you simply need to translate from world to view space.

## 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.