Displaying Pokemon Stats as a Hexagonal Graph or Bar Graph

By Fernando G. Berry

The basic formula for determining the length of an element or distance of a point from center is:

$$S_{base} \div S_{max} * D$$

Where S_{base} is the base stat of a pokemon, S_{max} is the maximum possible stat value, and D is the maximum available distance of a given element or chart.

For a bar chart, this is all that is really needed to calculate as D can just represent an element height or width.

A hexagonal chart is different as the distance represents a vector from the center of the chart to the maximal point that is $60°$ from a reference angle.

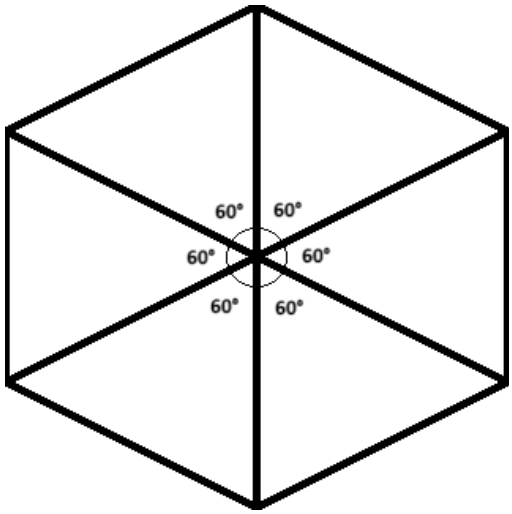

Figure 1. Angles of a hexagonal chart

The solution of this problem is to use the trigonometric functions of $sin()$ and $cos()$ to find the $opposite$ and $adjacent$ of a given $hypotenuse$ equaling D.

$$opposite = sin(\Theta) * D$$
$$\&$$
$$adjacent = cos(\Theta) * D$$

We can then define the $opposite, adjacent$ as the x, y coordinate and apply an offset to re-zero the point of origin (PO) to the center of a drawing screen. A $-30°$ offset will be applied to each angle since angles are being referenced to a cartesian coordinate system.

A function will be invoked to perform the required mathematics and return a pair of coordinates based off of angle, distance, and PO arguments. This function will ignore the $sin()$ and $cos()$ processing of $\pm$ 90° angles. It will instead treat the distance as the y coordinate. An inversion of the y coordinate will be required for angles greater than 180°. All angles provided as arguments are expected to be $30°, 90°, 150°, 210°, 270°, \ or \ 330°$ after the angle offset is provided.

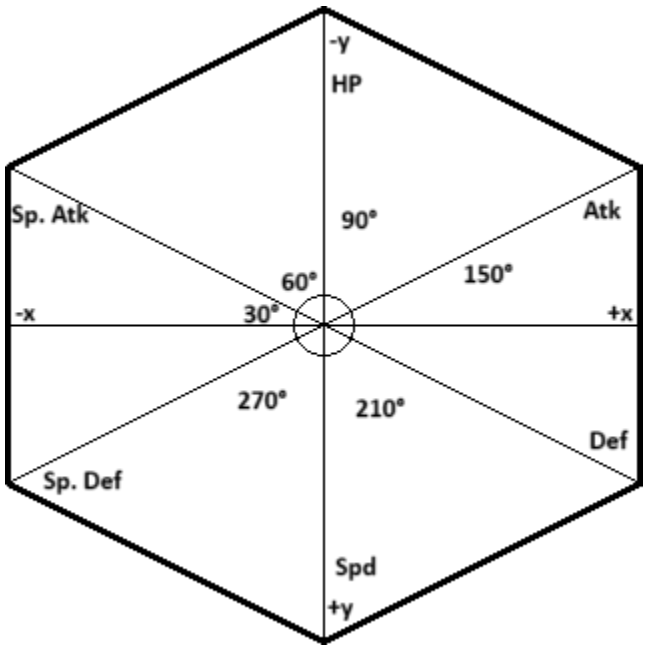

Figure 2. Labeled Hexagonal Chart showing inverted x,y axis, stat labels, and angles

The function is expected to behave as follows:

```
Declare Function getCoordinate( number angle, number distance, array offset )
  Declare array coordinate and assign an array with indices 0 and 1 set to 0
  Check if number angle > 180 degrees
    true: assign number distance = negative number distance
    false: continue
  Check if number angle is 90 or 270 degrees
    true: assign array coordinate index 1 = negative number distance
    false:
      assign array coordinate index 0 = sin(number angle) * number distance
      assign array coordinate index 1 = cos(number angle) * number distance
  assign array coordinate index 0 = array coordinate index 0 - array offset index 1
  assign array coordinate index 1 = array coordinate index 1 - array offset index 1
  return array coordinate
End of Function
```