



Cartesian Coordinate Translation, Enlargement & 1/4 Turn Rotation

Computer Science Concepts

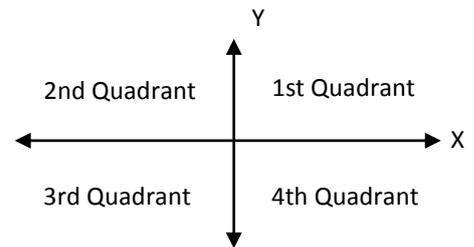
- X & Y coordinates to place sprites
- Reporting X & Y coordinate positions
- Using variables to adjust x and y (add, mul-

Program Aim

To explore translation of X & Y coordinates, decompose the program or recreate their own version by adapting the code to randomise the starting shape.

Maths Concepts

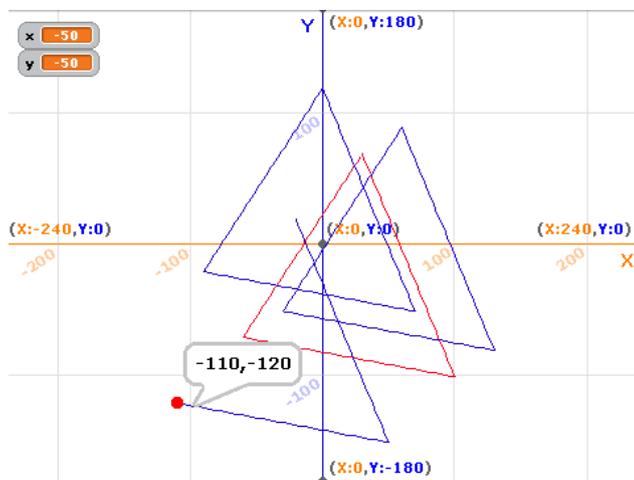
- Cartesian Coordinates
- Reflection of shapes into second third and fourth quadrants
- Translating shapes into other positions (add to x & y)
- Enlargement of shapes (multiply x & y)
- Simple rotation of shape through multiples of 90 degrees



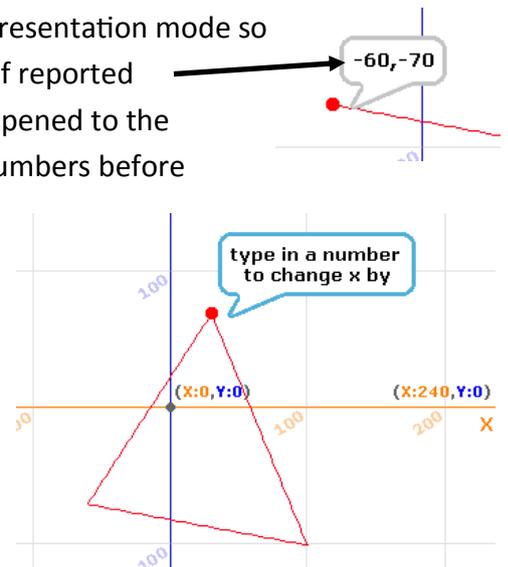
Activities

Exploring translation

This is the simplest activity which involves using the programs to explore what happens to X & Y when positive or negative numbers are added to all the X and Y coordinates. This is more of a supporting maths activity than a computing and maths activity. Download and load the **simple triangle translator** from code-it.co.uk into either Scratch 1.4 or Scratch 2.0. Although the code was designed in 1.4 it will upload and work perfectly well in Scratch 2.0. Run the program in presentation mode so that pupils are unable to see the code. Get pupils to note one pair of reported coordinates then add 10 to this number and ask them what has happened to the coordinates when the shape is translated? Repeat with positive Y numbers before moving on to ask what would happen to negative numbers. Can



pupils find numbers to add that would place the shapes wholly in a single quadrant? Pupils could mirror this activity off screen adding a fixed amount to all x coordinates and plotting the positions of the shapes.



how it works?

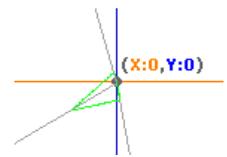
Finally allow pupils to see the code. Can they explain



Cartesian Coordinate Translation, Enlargement & 1/4 Turn Rotation P2

Exploring Enlargement

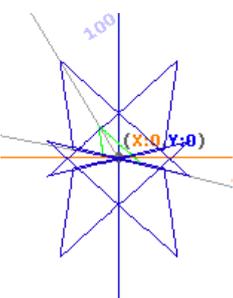
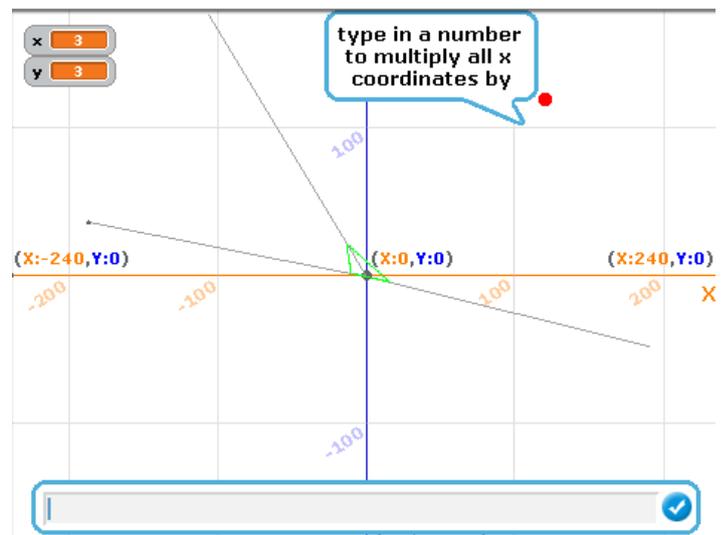
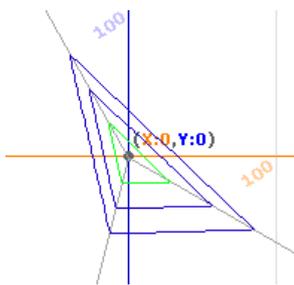
This activity involves using the program to explore what happens when the x and y coordinate values are multiplied. Download and load the **enlargement1** code from code-it.co.uk into either Scratch 1.4 or Scratch 2.0. Although the code was designed in 1.4 it will upload and work perfectly well in Scratch 2.0. Ask pupils if adding a number to the coordinate values (moved) translated the shape then what would you need to do to the x and y coordinates to enlarge it? (Answer multiply it) Run the program in presentation mode so that pupils are unable to see the code. Give pupils time to enlarge the shape. They are only limited by the size of the screen. Can they also shrink the shape? (multiply x and y by a decimal fraction less than 1 such as 0.5) Now open



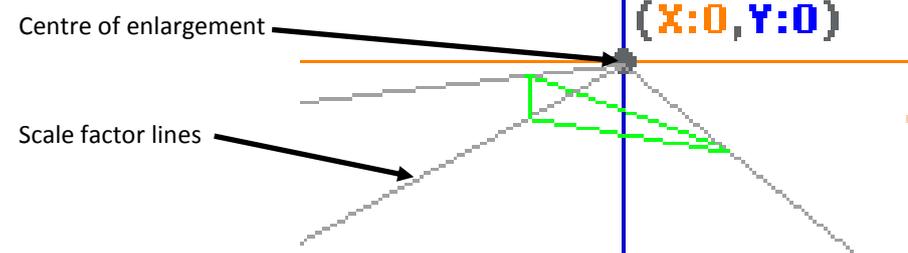
enlargement2 note with pupils how this draws a line from the centre of enlargement (0,0 in this case) through each pair of coordinates. Give pupils

time to explore enlarging shapes what do they notice about these lines? (If x and y are enlarged equally the shape coordinates will always lie on these lines) What do pupils think will happen if

they multiply by a negative number? Why not explore a negative and a positive number?



If you wanted your pupils to just explore enlargement in just the first quadrant first then change the range of random generated numbers to positive numbers only. When exploring multiplying by negative numbers you may wish to use **enlargement3** which has a reverse scale factor line



drawn at 180 degrees.



Cartesian Coordinate Translation, Enlargement & 1/4 Turn Rotation P3

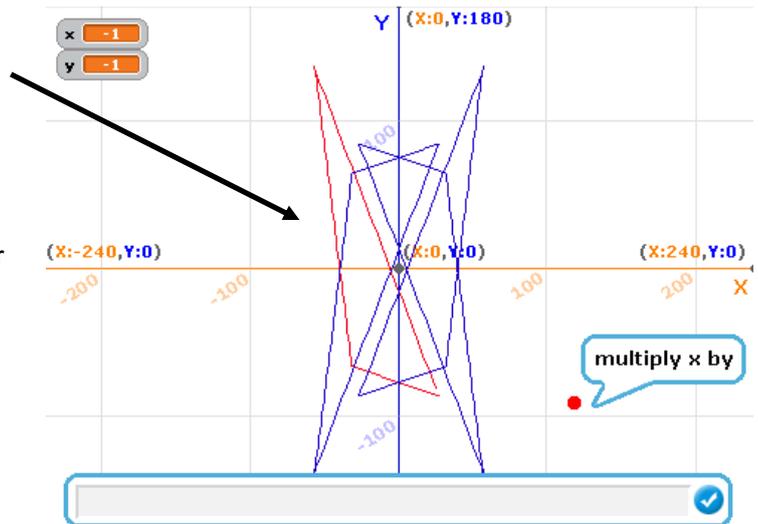
Exploring 1/4 Turn Rotation

Pupils may already have discovered that by multiplying either x or y or both by a negative 1 then the shape will rotate and mirror for 90° and 270° but rotate correctly for 180° degrees as shown.

However to truly rotate without mirroring 90° (x, y) becomes (y, -x)

180° (x, y) becomes (-x, -y) (rotated correctly)

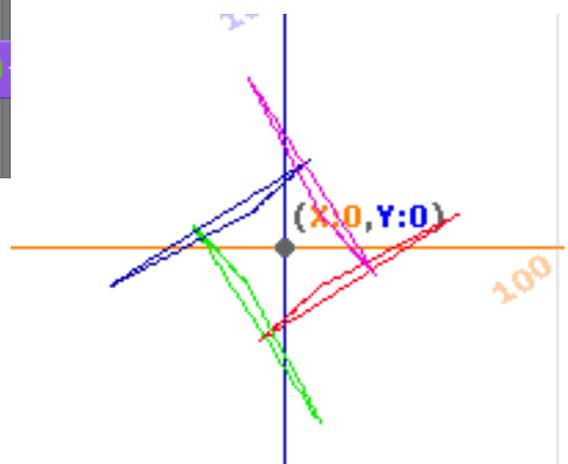
270° (x, y) becomes (-y, x) So to rotate a quarter and three quarter turns we need to swap the x and y coordinates as well as make one set of coordinates negative which can be done by multiplying it by -1.



```

when I receive rotatechoice
ask type 1= rotate 90 2=rotate 180 3 = rotate 270 and wait
if answer = 1
set pen color to
glide 1 secs to x: y1 * +1 y: x1 * -1
say join join x position of Sprite1 , y position of Sprite1
pen down
glide 1 secs to x: y2 * +1 y: x2 * -1
say join join x position of Sprite1 , y position of Sprite1
glide 1 secs to x: y3 * +1 y: x3 * -1
say join join x position of Sprite1 , y position of Sprite1
glide 1 secs to x: y1 * +1 y: x1 * -1
pen up
  
```

Please note, mathematicians would assume that the direction of rotation would be anticlockwise by default where Scratch rotates clockwise by default. A default anticlockwise rotation would lead to the 90° and 270° values identified above being swapped.



In this code to rotate 90° the x and y values have been swapped and the y value has been converted into a negative by multiplying it by -1

Pupils can explore this using **rotate** download



Cartesian Coordinate Translation P2

Examining the code with a view to randomising the location of the original base shape

Give pupils time to work with the simple triangle translator programs. Then show them the code. Challenge them to explain what is happening in each section either to a partner or to you. Could they find a way to randomise the triangle drawn? There are plenty of examples of this in the enlargement and rotate code.

Decomposing the main elements of the program before building their own version

This activity is only relevant for your most able pupils in KS2 although it would be relevant to many more in KS3. I would recommend that pupils complete the Cartesian Coordinate plotting and programming work found at <http://code-it.co.uk/scratch/cartcoordinates/cartesiancoordinates.html> and other Scratch programming modules that use variables first.

You could use the decomposed planner question sheet available or alternatively this can all be done verbally with pupils instead. Whatever method you use, you need to be satisfied that pupils have an idea how shapes can be drawn using glide to X & Y blocks or go to X & Y blocks combined with pen blocks. They also need to understand that they need a mechanism to report the coordinate positions and add to the X & Y coordinates. Once this understanding is in place pupils can puzzle out the rest.

```
when clicked
clear
pen up
set pen color to red
glide 1 secs to x: 30 y: 70
say join join x position of Sprite1, y position of Sprite1 for 1 secs
pen down
glide 1 secs to x: 100 y: -100
say join join x position of Sprite1, y position of Sprite1 for 1 secs
glide 1 secs to x: -60 y: -70
say join join x position of Sprite1, y position of Sprite1 for 1 secs
glide 1 secs to x: 30 y: 70
pen up
```

You can also use X & Y position blocks in place of the x position of sprite blocks and go to X & Y blocks in place of the glide

Part 1



Cartesian Coordinate Translation P3

Part 2

```
pen up
forever
  ask type in a number to change all the x coordinates by and wait
  set x to answer
  ask type in a number to change all the y coordinates by and wait
  set y to answer
  pen up
  set pen color to blue
  glide 1 secs to x: 30 + x y: 70 + y
  say join join x position of Sprite1 , y position of Sprite1 for 1 secs
  pen down
  glide 1 secs to x: 100 + x y: -100 + y
  say join join x position of Sprite1 , y position of Sprite1 for 1 secs
  glide 1 secs to x: -60 + x y: -70 + y
  say join join x position of Sprite1 , y position of Sprite1 for 1 secs
  glide 1 secs to x: 30 + x y: 70 + y
  say join join x position of Sprite1 , y position of Sprite1 for 1 secs
  pen up
  go to x: 30 y: 70
```

Please feel free to adapt this planning in whatever way you see fit. Tell me if you find any better ways of teaching it and I will add your adaptations and credit you on the planning. It is crying out for a version written using functions or procedures but for most pupils that would be KS3.

My thanks to Dr Barry Hobbs for Maths advice

All mistakes are of course my own

Phil Bagge