

Exploring Turtle Geometry through Polygons and Spirals with TurtleArt

Cynthia Solomon

Introduction

Lately I've been thinking about spirals; in particular how to get Logo Turtles to make spirals. Seymour Papert first invented these turtles in 1969 at Bolt, Beranek and Newman (BBN) and at the MIT AI Lab. The need for them was in reaction to Seymour and me teaching a year-long Logo-math class to twelve-year olds. Logo was the programming language initially designed for children by Papert and a small group of BBNers (Wally Feurzeig, Danny Bobrow, Dick Grant and me). Seymour felt the children did well but could have done even better if they had a concrete object to interact with.

In the sixties Seymour was also steeped in work on robotics at MIT with Marvin Minsky, his close collaborator. Seymour didn't think of the turtle as a robot nor as an automaton, just a thing to control. But turtles had pens and could leave a trace of their paths. In 1970 a whole new world of computerized drawing, graphics and animation opened up to children when both floor and display turtles were set up in an elementary school where Seymour and I taught 10-year olds in a year-long computer class.

The basic turtle commands in Logo are: move FORWARD or BACK a number of steps and turn LEFT or RIGHT a number of degrees. Two of today's block languages, Scratch and Snap! and their offshoots have three commands: move, turnleft, and turnright. TurtleArt, on the other hand, is devoted to turtle behavior and follows the original Logo commands of forward, back, left and right.

"The 'turtle language' provides a formal system for describing many geometric objects; we think vastly superior to Cartesian coordinates as an introductory path into geometry" (From *20 things to do with a computer*). A powerful idea emerged: the turtle will eventually return to its starting state if it moves a fixed amount and turns a fixed amount repeatedly. The exception occurs if it turns 0 or 360 degrees whereupon it will walk in a straight line and never get back to its starting state (The Total Turtle Trip Theorem). The turtle's state is described by its position and heading. A display turtle's position is where it is on a grid, which is marked by the intersection of horizontal (x) and vertical (y) lines. The center position has x as 0 and y as 0.

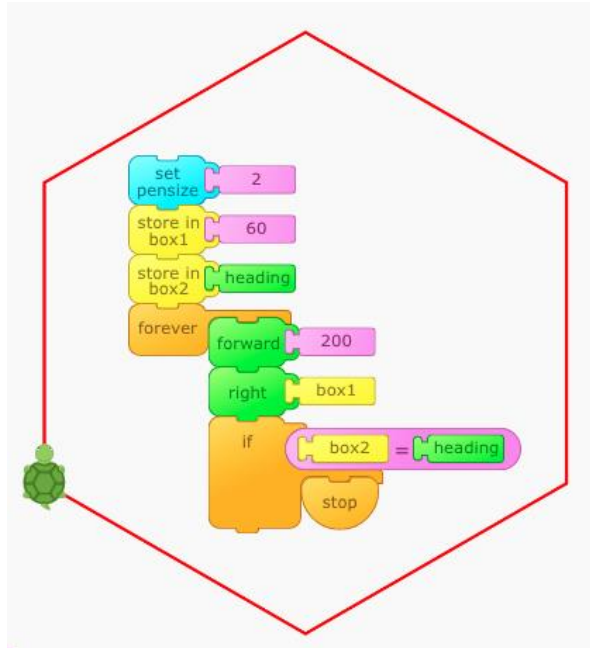
Regular Polygons

I have to admit I love making the turtle draw polygons whether they are regular polygons or star polygons. You may wonder what's a turtle and what does Logo have to do with it?

The turtle I have in mind takes two forms. It lives on a graphics screen and it lives on the floor as a wheeled mechanical creature. A turtle understands and obeys a cluster of commands.

Logo was the first computer language that had commands I could give to the turtle.

I can make polygons in different sizes and colors. I can make designs out of polygons and then capture the screen results on paper or have the floor turtle draw on paper. I can now transfer the designs to other materials by using digital devices such as a computerized embroidery machine, a laser cutter, or a 3D printer.



Look how I squeezed in a stop rule! I bet you wondered why I stored the turtle's starting heading. It is because of the total turtle trip theorem which states that the turtle will return to its starting position and heading if it moves a constant amount and turns a constant amount repeatedly.

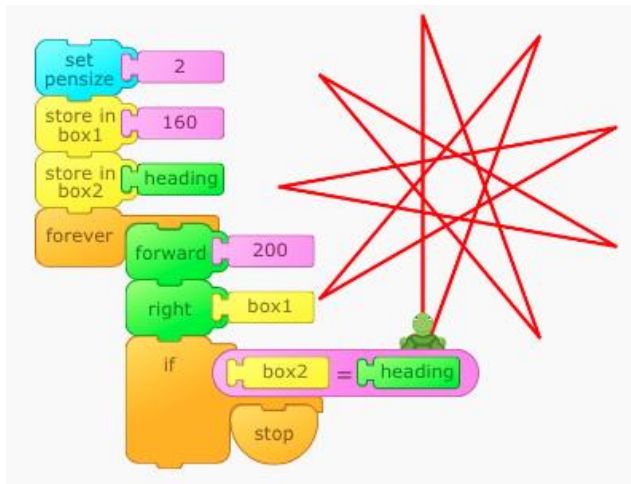
Test the theorem out by being a turtle and walking in a path where you move and turn constant amounts. Try walking in a square or a circle and jot down the commands you, the turtle, need to follow.

The stop rule is expressed by the if block. When the value stored in box 2 (the starting heading) is the same as the turtle's current heading, the job is done.

Play around. Make a square, a triangle, an octagon, a circle. Make the polygons in different sizes and colors.

Star Polygons

The same stack of blocks can make star polygons too. You just have to find the angles to do the job you want. Here is a stack of blocks in TurtleArt for making polygons. You have to change the value in box1 to create different polygons. Below is a star polygon. (My favorite)



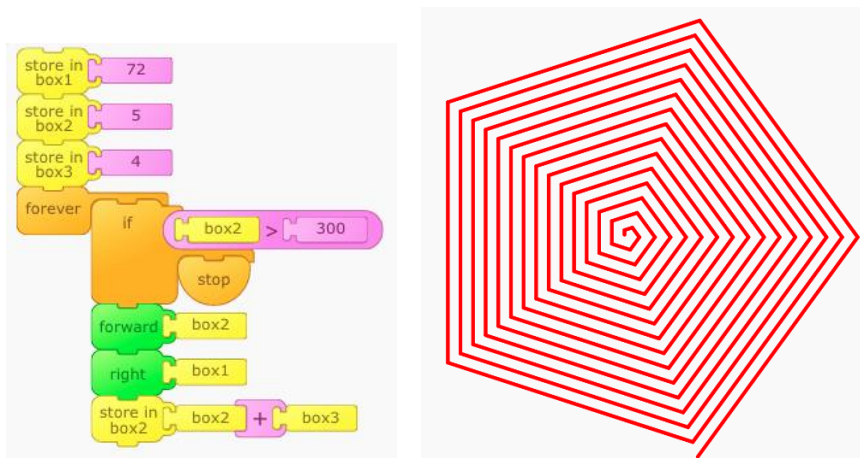
A clue: If you want to find an angle for the turn: with regular polygons their lines do not cross so their total turning is 360 degrees. Thus a regular polygon turns a total of 360 degrees. Notice the turtle makes external (not internal) turns.

There are two values you want to modify: the forward input and the left or right input.

Make 5-pointed, 8-pointed, multi-pointed stars. How do you find these angles? You know the total of the angles has to be greater than 360. Why? In pursuing possible angles, try doubling or tripling the angles of regular polygons.

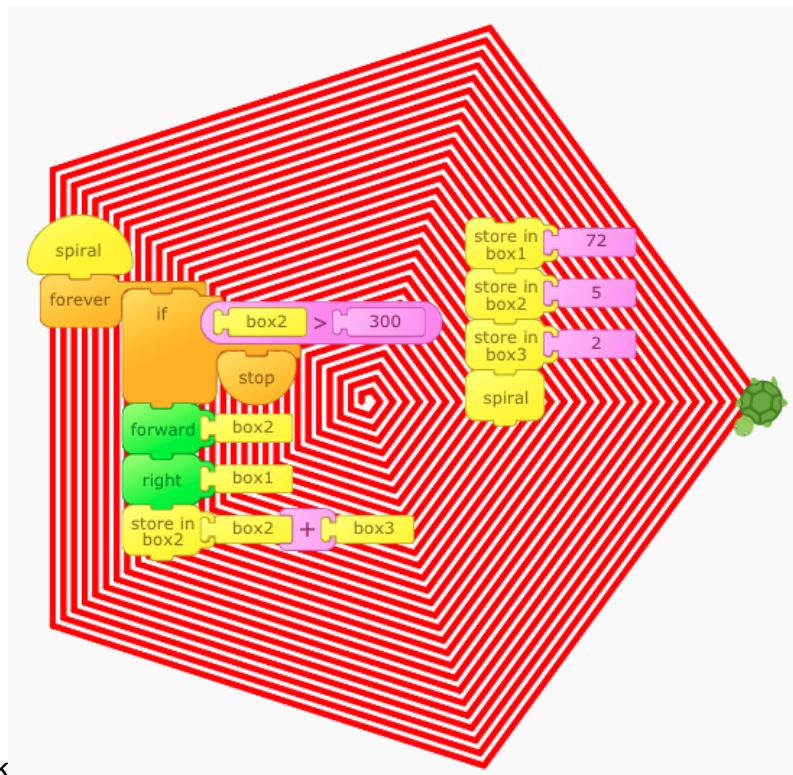
Spirals

My enthusiasm for polygons led me to making the turtle draw spirals.

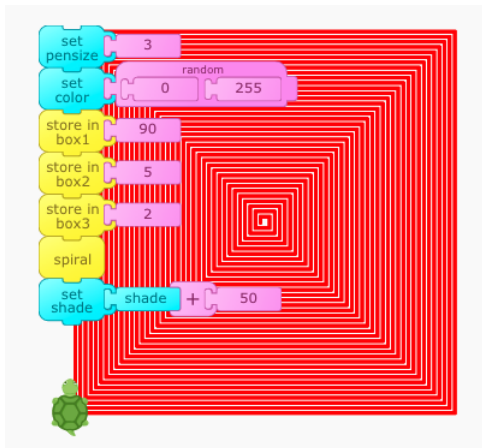
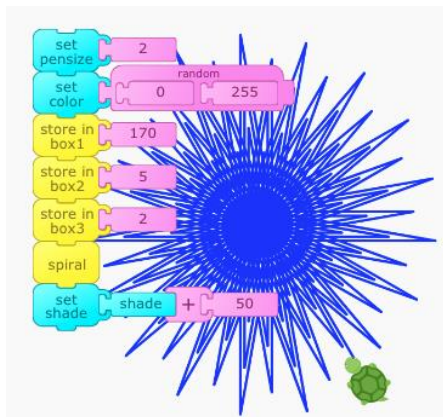


Notice: This stack has a different stop rule. The job is done when the forward step is greater than 300 or some other specific number.

Experiment with color and hue and see how the perception of the geometric shape changes. See what happens when you almost make a square spiral, but change the value from 90 to 92.



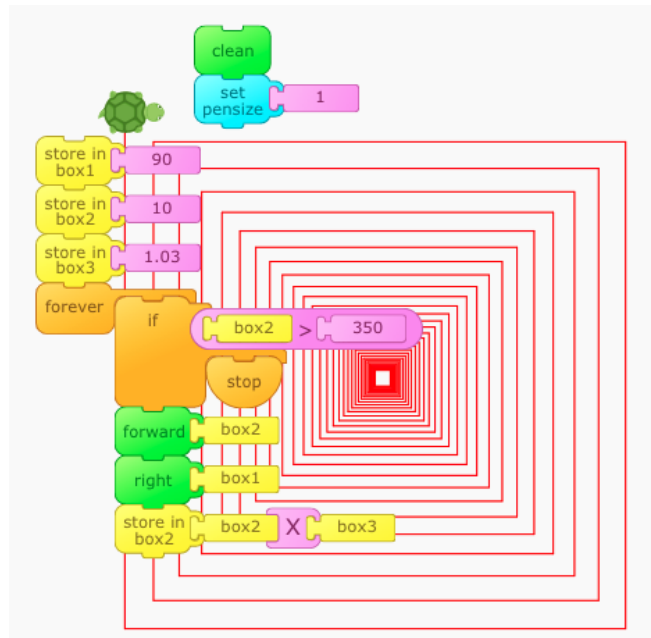
You can make a spiral block and then use it.



box 1 contains the angle of rotation
box 2 contains the forward step
box 3 contains the amount the forward step increases

Try your favorite polygon angles with the spiral blocks.

Here is another way to make turtle spirals. Instead of adding or subtracting a fixed amount to/from the forward step multiply the forward step by a small amount. What happens if you try a large amount?

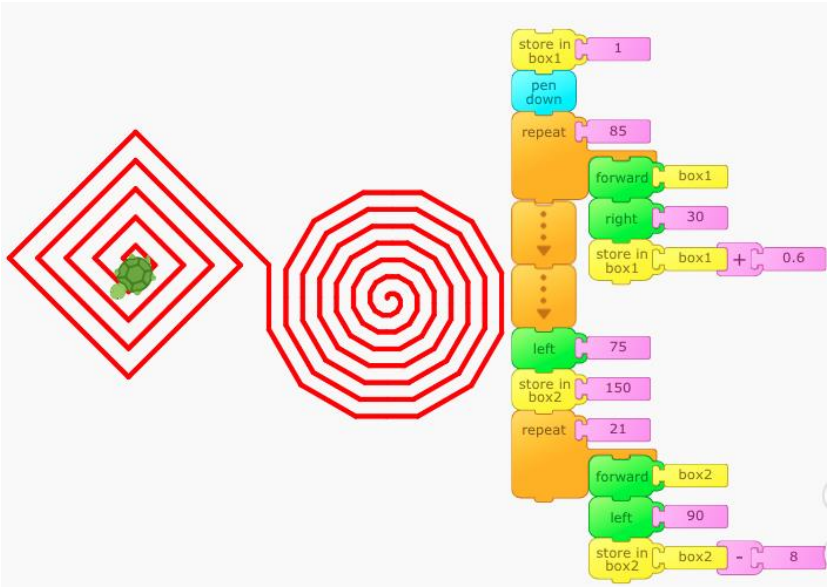


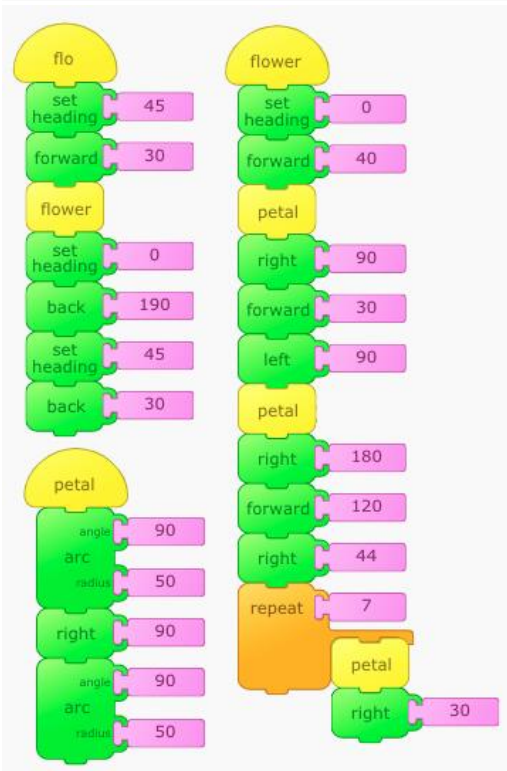
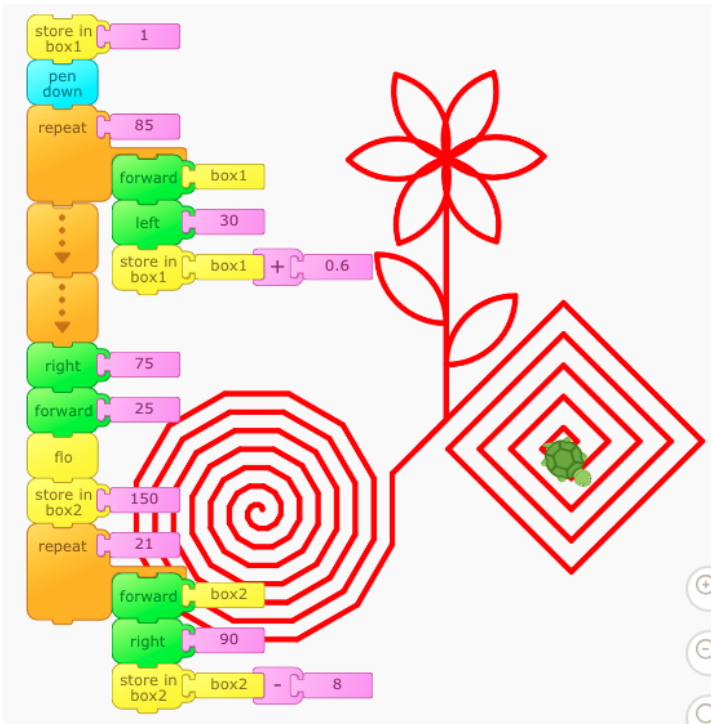
There are some names for this kind of spiral: exponential or logarithmic spirals. The growth is so fast that the increasing step size needs to be small.

Can you find ways to make the turtle draw a polygon by rotating in the opposite direction?
Can you get the turtle to make a spiral by starting with the longest side?

A Final Note

Here is a lovely spiral design by Artemis Papert. And notice her code is slightly different from mine.





Finally, Artemis changed the shape of the circular spiral by using the arc block.

