Programming Evo to Make Regular Polygons
with Move and Rotate Blocks

by Richard Born
Associate Professor Emeritus
Northern Illinois University
rborn2@niu.edu

Introduction

A ***regular polygon*** is a two-dimensional figure characterized by two features:

1. All of its interior angles are equal, *i.e.*, it is *equiangular*.
2. All of its sides are equal in length, i.e., it is *equilateral*.

Figure 1 shows some examples of common regular polygons. An *n*-gon polygon has *n* equal sides with *n* equal internal angles. An equilateral triangle and square are probably the most well-known polygons. The headquarters of the United States Department of Defense is a 5-gon building aptly known as the Pentagon. Beehive cells are known for their hexagon shape. Throughout the world, a red octagon warns us to stop. Some countries mint decagon and dodecagon coins.

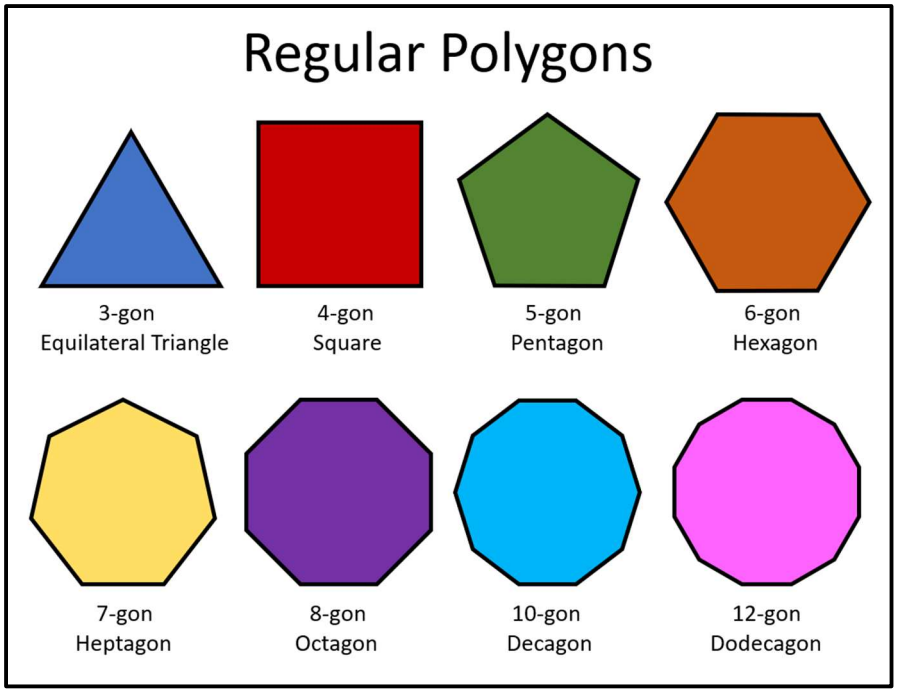


Figure 1

Some Polygon Geometry

An *exterior angle* of a polygon is the angle between any one of its sides and a line that is extended from the previous side. As an example, Figure 2 shows the exterior angles for a regular pentagon. It can be proven that *the sum of the exterior angles of a polygon always add up to 360°.* Therefore, each exterior angle of a regular polygon with *n*-sides must be $360^o/n$. The exterior angle of the regular pentagon of Figure 2 is $360^o/5 = 72^o$.

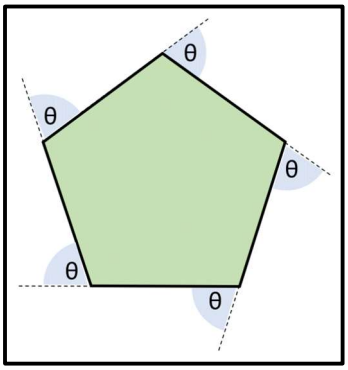


Figure 2

Programming Evo to Make a Regular Polygon Using Move and Rotate Blocks

Now let's suppose that we want to make Evo make a regular polygon using move and rotate blocks. Consider the regular pentagon shown in Figure 3 as an example. With Evo located at the start position facing the direction shown by the arrow, the OzoBlockly program is started. A *move block* specifies the distance that Evo is to move—the length of any side of the regular pentagon. When Evo reaches the vertex in the lower right corner, it must rotate left by the number of degrees in the exterior angle, *i.e.*, 72^o. This is accomplished by the *rotate block*. The rotate and move blocks can be placed inside a *repeat 5 times* loop to force Evo to make the entire regular hexagon.

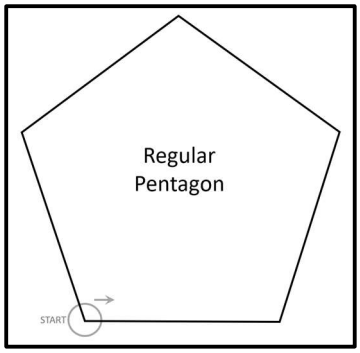


Figure 3

Improving the Shape of Evo's Regular Polygons

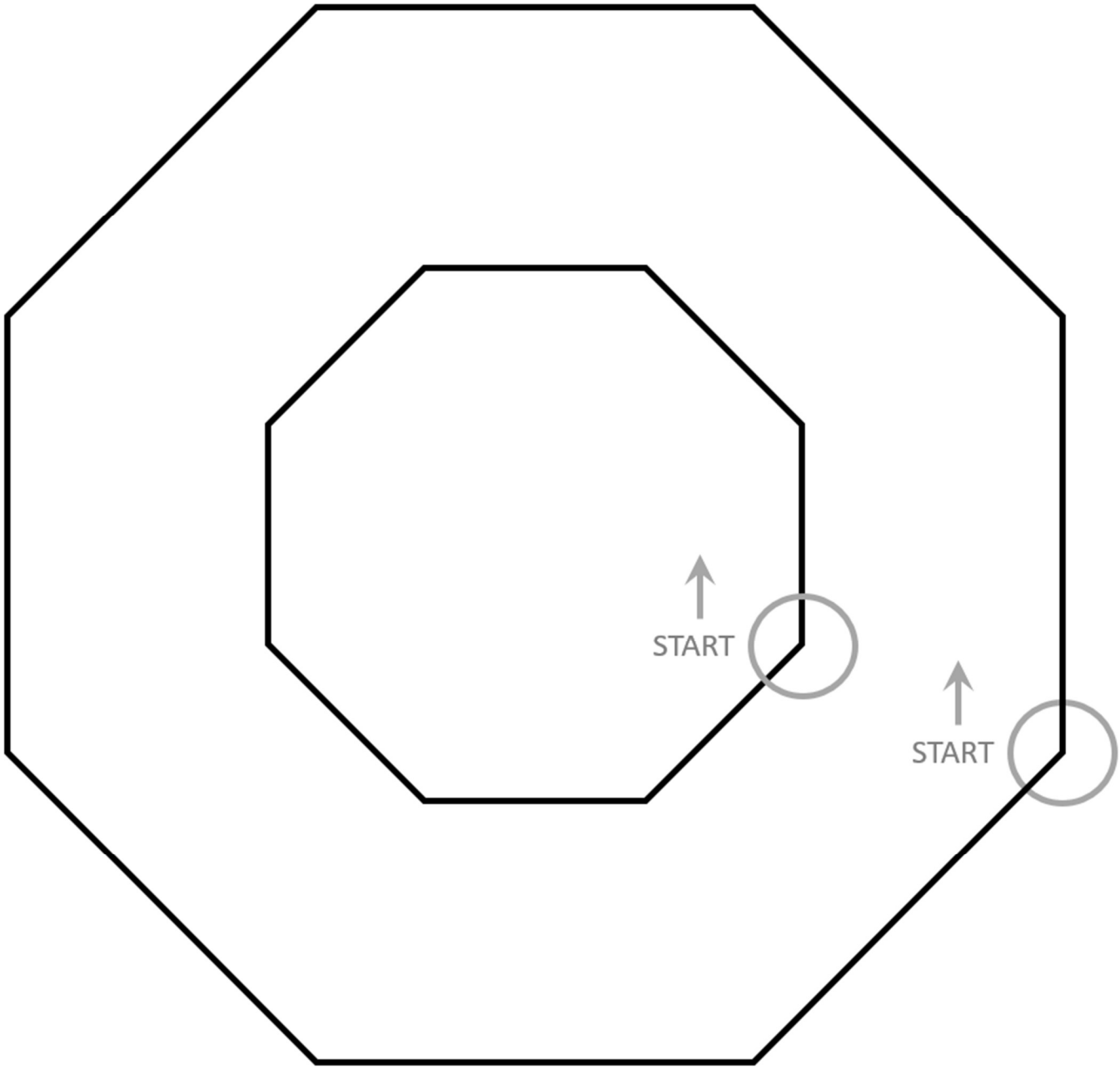
You may notice that upon completing the regular polygon, Evo has either under-rotated or over-rotated the angle through which it is supposed to rotate. As was discovered previously, Evo should rotate *exactly* 360^o while making any polygon. But, as pointed out in the OzoBlockly Glossary:

> Note that Ozobot performs the rotations with a limited precision due to traction differences between different operating surfaces (wheel slippage), inertia, speed calibration, and its hardware design limitations. The over- and under-rotation errors can be also varied when you change the rotation speeds. However, the advanced control over the robot available in the **Move** and **Rotate** blocks allows for tuning and biasing of the input values in order to compensate for the use conditions and various deviations observed in your Ozobot.

As an example, the author of this lesson found that the regular pentagon produced by Evo was always over-rotated. With just a little trial and error, it was found that a rotation of 69^o instead of 72^o produced an almost perfect pentagon.

Investigations

1. Your teacher will provide you with a full-page printed copy of one of the regular polygons that appears in Figure 1. Design an OzoBlockly program using move and rotate blocks in a loop that will make Evo "trace" the regular polygon once and then stop and turn off. Evo should start in the circled vertex facing the direction of the arrow. It is advisable to include a "wait" block at the start of the program. This will give you a few seconds to make sure that Evo is centered on the circle and facing the direction of the arrow before Evo starts moving. Finally, you need to keep in mind that Evo can only handle input values as large as 127. Therefore, if the move distance exceeds 127 mm or the rotate angle exceeds 127^o, you will need more that one move block or rotate block to handle larger numbers.
2. In the previous investigation, Evo was moving counterclockwise around the regular polygon. Modify the program and start direction so that Evo will move clockwise around the polygon.
3. Using a pair of Evos, place an Evo on each of the two octagons shown on the next page. Program them to trace the octagons in sync, starting at their respective start locations, and finishing their traces simultaneously. A video of this accompanies this lesson.


START
START

Optional Project

This optional project is for students who enjoy creating robotic art. Figure 4 shows a collage of regular polygons created by the author of this lesson. Each member of the collage was created by taking a *long exposure* photo of Evo as he traced the regular polygon.

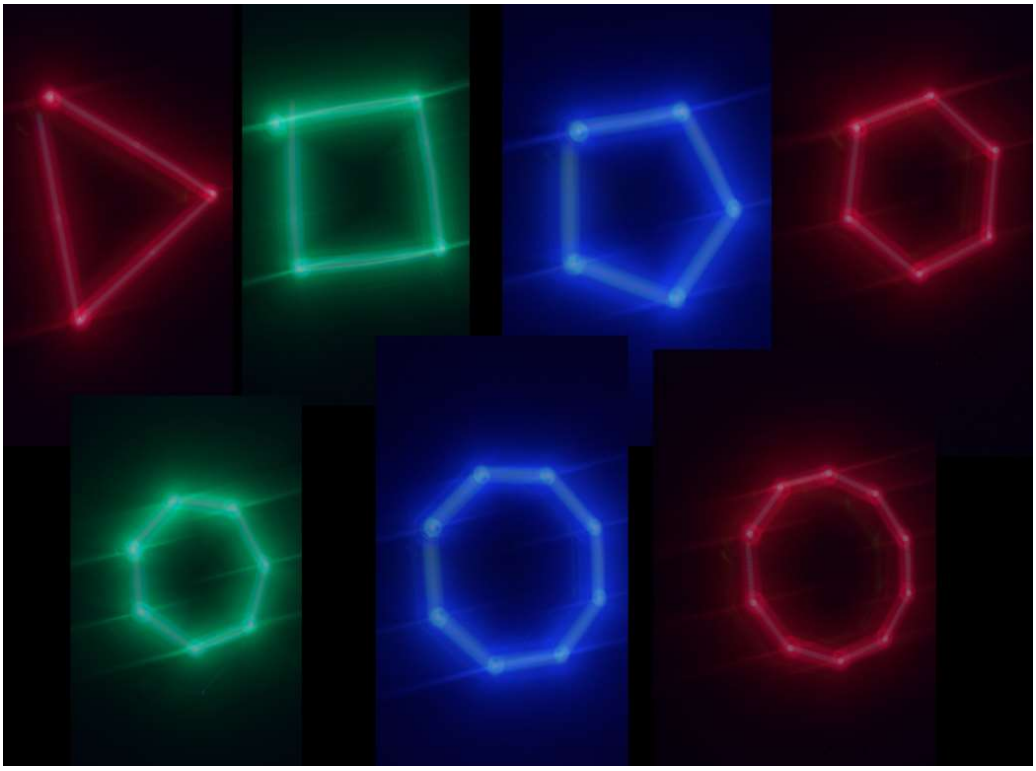


Figure 4

All of the photos were taken in a dark room using an iPhone and the app called "Slow Shutter". The settings were as follows:

- Capture mode is "Light Trail"
- Light Sensitivity is "1/16"
- Shutter speed is "Bulb"
- ISO is "Auto".

It is critical to keep the iPhone still during the exposure. A tripod or some other way to keep the iPhone steady is a must.