Lesson on Circles: OzoBlockly Rotate and Move Blocks

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

There are no OzoBlockly movement blocks that tell Evo to go in a circle. Yes, Evo can *rotate in place* with the ***rotate*** block, but that does not make him move in a circle of some non-zero radius. Never-the-less, we can make Evo execute a nearly perfect circle by *combining* ***rotate*** and ***move*** blocks in a loop.

Consider the short program shown in Figure 1. Each rotate block is a 2^{o} rotation for Evo. Since the rotation is repeated 100+80 = 180 times, Evo will rotate a total of 360^{o}—a full circle. (We had to use two loops to get 180 rotations since 127 is the largest number Evo knows.) Each rotate is preceded by a move block in which Evo moves 2 mm in a straight line.

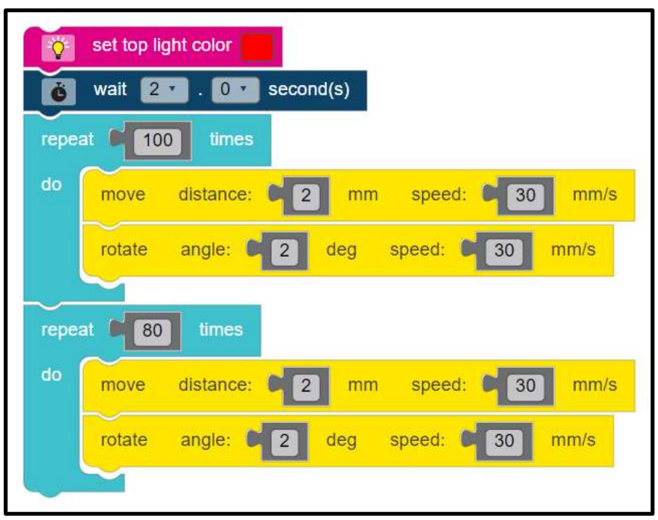

Figure 1

Exercise 1

Construct the program of Figure 1 in OzoBlockly, load the program into Evo, and place Evo on a small dot at the bottom of a piece of 8½" x 11" paper and facing the right. Run the program. Mark Evo's center on the paper when Evo is at the top of the circle. (The distance between the two dots will give you the circle's diameter, from which you can find the center of the circle.) Refer to Figure 2 for clarification. Did Evo move in a circle? Did Evo stop at the first dot, or did he over-rotate? or under-rotate? Mark Evo's center with a dot at his stop location.

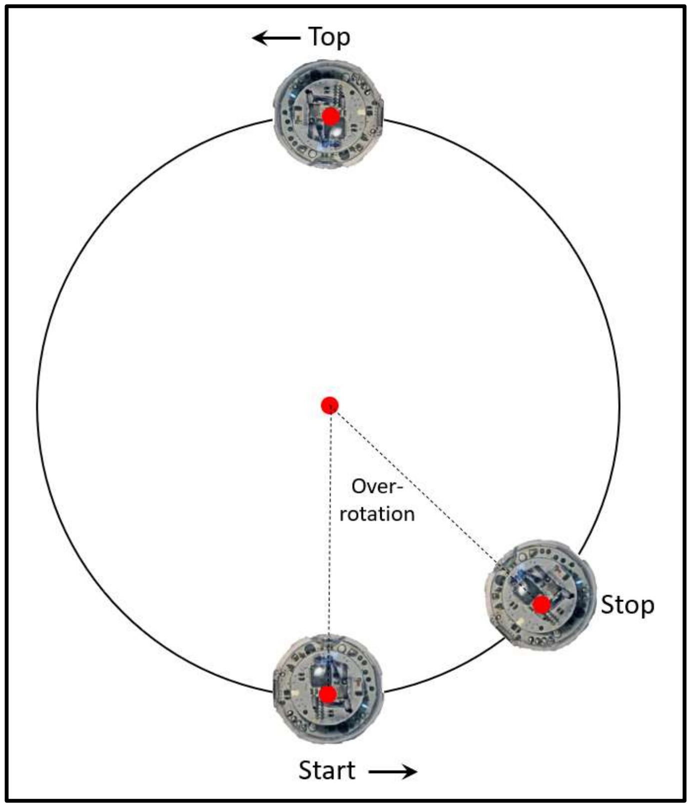

Figure 2

Let's assume that Evo over-rotated the circle. Why would he do this? The answer lies in the Ozobot Master Mode Glossary where it is stated: "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." Apparently, Evo rotated a bit more that 2^o in each rotate block. So, let's correct for the over-rotation.

Exercise 2

Use a straight-edge to draw a line from the center of the circle to the starting dot. Then draw a line from the center of the circle to the ending dot. Use a protractor to measure the angle between these two lines, as shown in Figure 2. For example, suppose that this angle is 40^o. This means that Evo executed an excess of 20 rotations of 2^o each. We could correct this by changing the loop count in the second loop block from 180 to 160. Use this logic with the angle you measured with the protractor to adjust the loop count in the second loop block. Then load the adjusted program into Evo, and run the program. Did Evo complete a full circle and stop at the starting point this time? If you want, you can tweak the loop count a little to improve this further. What would you need to do if Evo under-rotated the circle?

We said at the beginning of this lesson that "Evo execute a nearly perfect circle". Why did we say "*nearly perfect"*? Let's address that question by studying Figure 3. The figure shows a portion of a circle C and its center. Since we are using rotations of nominal value 2^o, a sector of that size has been constructed as shown. Since the angle is small, the length of arc AB and chord AB are very nearly the same. This is clarified in the foreground enlargement of the blue shaded region. In contrast, arc DE is a little longer than chord DE, since the arc is curved and the chord is a straight line. The bigger the angle, the greater the difference in the arc and chord lengths. (Consider the arc and chord on a 60^o sector, for example.) For small rotations of just a few degrees, we can approximate—nearly perfect—the circumference C of the circle by multiplying the number of rotations by the distance moved for each rotation. Since we know that $C = \pi D$, then we can find the diameter D of the circle by dividing the circumference by π, i.e., $D = C/\pi$.

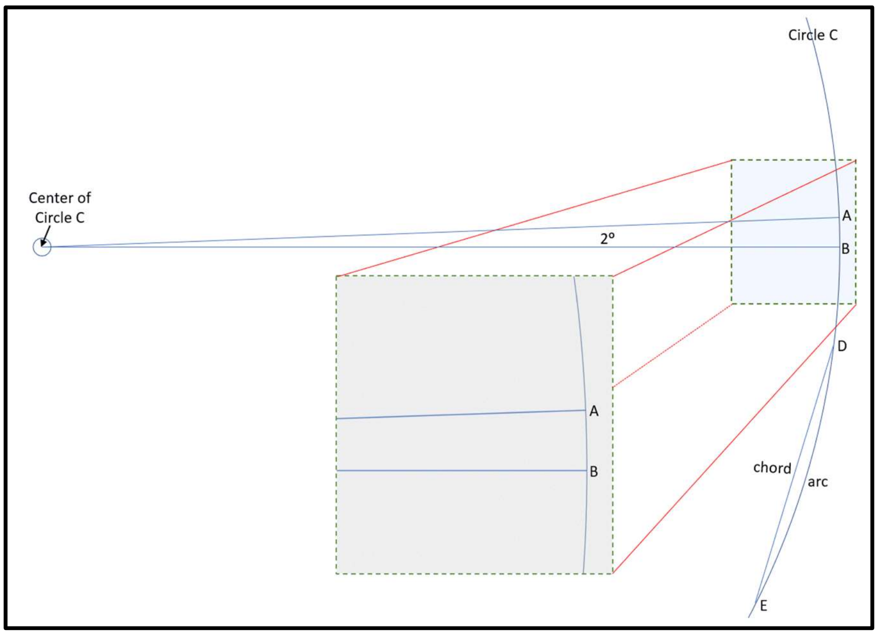

Figure 3

Exercise 3

Modify your program so that each rotation is 3^o and each move is 4 mm. Adjust your program for over-rotation or under-rotation, as needed. Tweak the number of rotations if necessary, so that Evo starts and stops at the same location.

- *Measure the diameter of the resulting circle.*
- *Using our discussion of Figure 3, calculate the theoretical diameter.*
- *Do the two results agree? If not, what is the percent difference?*

Exercise 4

Modify your program to make Evo's circle movement a little more exciting. Adjust the program so that:

- *Evo's top light turns on just before starting each rotation and turns off immediately after completing each rotation. (The light should not be on while Evo is executing a move block.)*
- *Evo plays a MIDI note number 100 for 10 ms just before the light turns off each time.*

You can view the short video that accompanies this lesson to see the final result from this modification.

Optional Project

This optional project is for students who enjoy creating robotic art. Figure 4 shows a long exposure photo of Evo traversing the circle of Exercise 4.

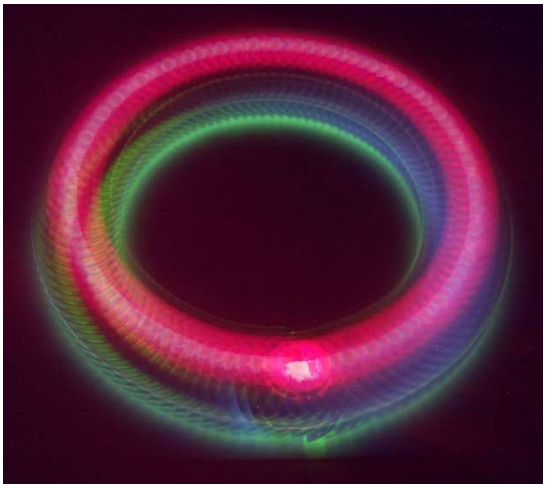

Figure 4

The photo was 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.