OzoBlockly Editor—**Lesson on Concurrency**—Student Guide

Ozobot as a Slot Car

by Richard Born, Associate Professor Emeritus, Northern Illinois University

Introduction

Slot car racing has been of interest to many for years and years. In this lesson you will learn how to design a program in the OzoBlockly Editor that makes Evo and Ari behave like a slot car on a slot car track. Also, since the OzoBlockly editor supports concurrency, *i.e., executing multiple tasks via* ***time-sharing***, you will learn how you can have more than one Ozobot racing at the same time on identical tracks.

Figure 1 shows the slot car track on which Ozobot will be racing. Ozobot is initially placed on the track in the direction shown by the arrow with its leading edge on the curved gray line. At every intersection Ozobot moves straight through, with the race ending at the *Finish* line.

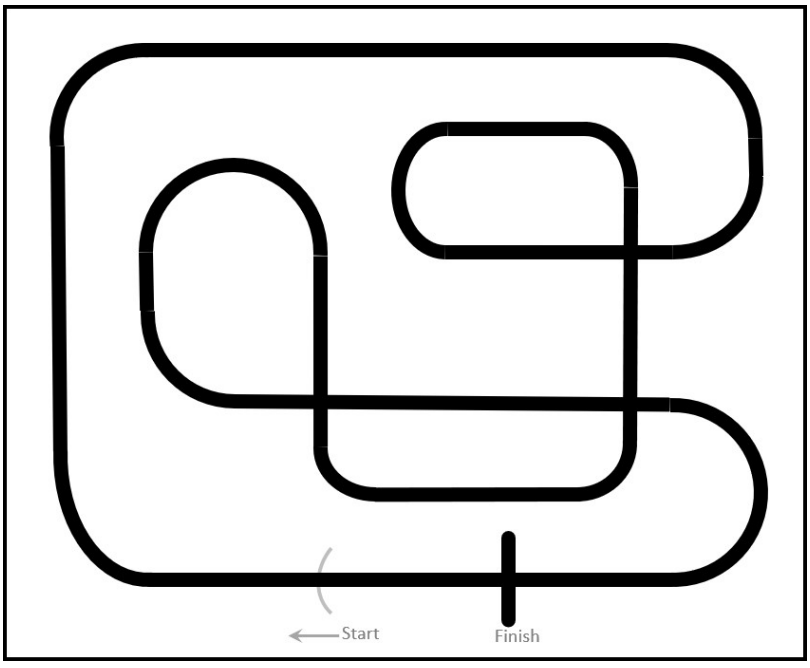

Figure 1

Slot Car Program Without Concurrency

Let's begin by constructing the slot car program without considering concurrency, as shown in Figure 2. Ozobot is placed at the start location on the Ozomap, and the program is started. The top light color is set to white to indicate that the program has begun execution.

The line-following speed is set to 50 mm/sec, though the OzoBlockly editor allows values from 20 mm/s to 90 mm/s. There is then a 2-second wait, and the top light is set to green (the color of a start flag), which will be the color while Ozobot moves through the slot car track.

You will find that there are seven intersections that Ozobot meets prior to finishing the race. Therefore, a repeat loops is executed seven times, with Ozobot following the line to the next intersection and picking the direction straight. Upon finishing the race, the top light color is set to red to indicate the finish, there is a 2-second wait, and the top light is set to white as the program terminates execution.

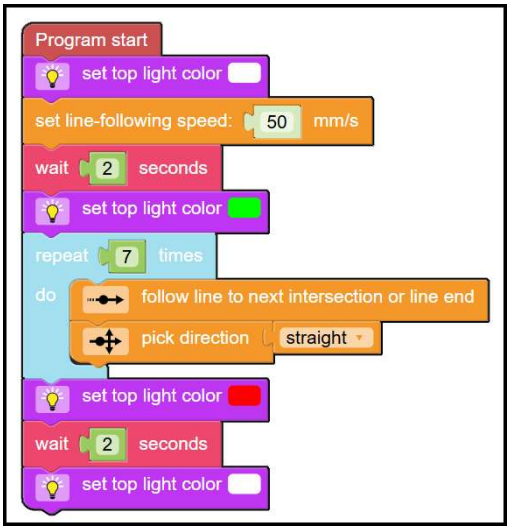

Figure 2

***Exercise 1**: Construct this program with the OzoBlockly Editor and run it either with and Evo or an Ari.*

Two Ozobots Running Concurrently on Two Identical Slot Car Racetracks

Now we are ready to study concurrency in which two Ozobots are racing against each other on identical tracks, as shown in Figure 3. This could be a pair of Evo's, a pair of Ari's, or one of each.

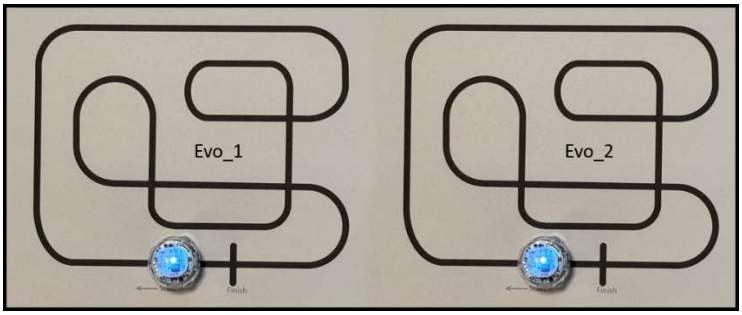

Figure 3

To accomplish this, we will be making use of additional blocks as shown in Figure 4. The blue block on the upper left is used to **define** a function, which consists of any number of blocks that *do something*. The blue block on the lower left is used to execute the blocks making up the function. This is often referred to as **calling** the function.

The red blocks in the center are identical in what they do. As implied by their name "run concurrently", they will run the code in the two enclosed blocks concurrently. The version of this in the bottom center of the figure visually makes it clearer that the code is run concurrently and not one after the other. Remember that concurrently does not mean simultaneously. Rather it means in a time-sharing fashion, with rapid shifting back-and-forth between the two.

The green *with actor* block on the far right is used to specify which device will execute the enclosed code blocks.

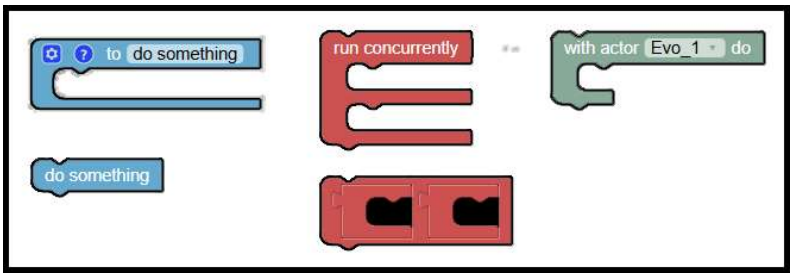

Figure 4

Armed with an understanding of the above blocks, it is straightforward to get two Ozobots to run concurrently on two identical slot car racetracks, as shown in Figure 5.

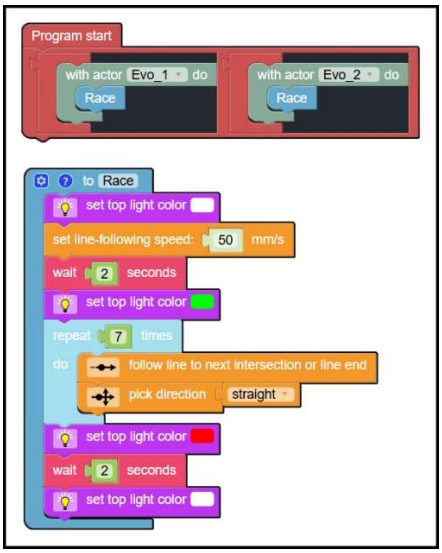

Figure 5

***Exercise 2**: Modify your program from Exercise 1 to get two Ozobots racing concurrently on two identical racetracks.*

Improving Visualization of Block Execution for Each Ozobot

You are undoubtedly aware that the Ozobot Editor can very briefly lightly flash each block in a program as it is executed. This is a particularly valuable tool when debugging your program. However, in the program of Figure 5, the flashing is quite confusing as two calls to the same function are occurring concurrently. One solution to this issue is to modify our program so that we have two functions containing identical blocks but with different names, say, *Race* and *Race 2*. The modified version of our program is shown in Figure 6. With this modified version, the flashing of blocks currently being executed visually separates out Evo_1 and Evo_2 flashing. It also shows that execution of code blocks is not simultaneous, but rather in a time-sharing fashion.

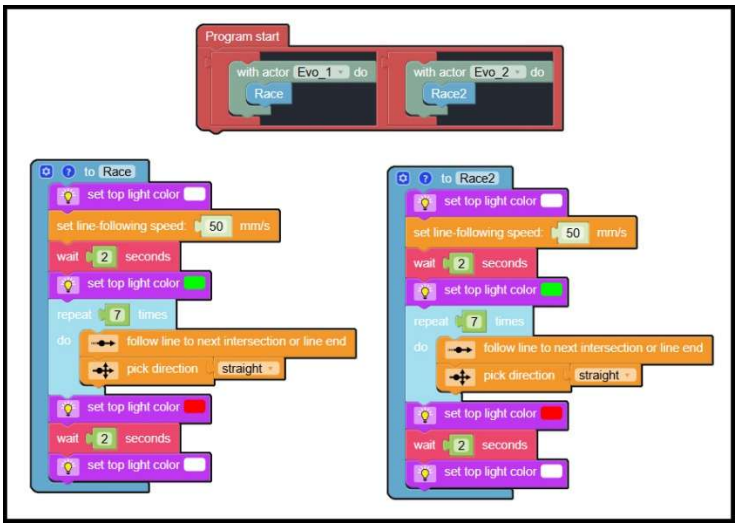

Figure 6

***Exercise 3**: Modify your program from Exercise 2 so that it separates flashing of blocks in the manner shown by the program in Figure 6.*

Nesting "Run Currently" Blocks

Now let's see how we can get **four** Ozobots to run concurrently on four identical slot car racetracks, as shown in Figure 7.

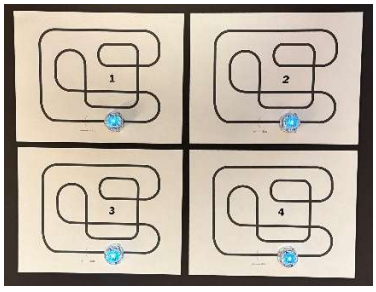

Figure 7

This can easily be accomplished by nesting two “run concurrently” blocks inside a “run concurrently” block, as shown in Figure 8. Two additional functions, Race3 and Race4, have been added. Then their respective function calls have been inserted into “ with actor…do” blocks Evo_3 and Evo_4.

Figure 8

An accompanying video shows this program in action, as the four Evo’s race on their respective tracks.

Exercise 4: *Modify your program from Exercise 3 so that you can view four Evo’s or four Ari’s race on four separate racetracks. The racetrack numbers in Figure 7 should correspond to Evo_1, Evo_2, Evo_3, and Evo_4. What do you notice when viewing the flashing of the blocks as the program executes? What is the order in which the Ozobots finish the race? Does this make sense when considering the fact that block execution is concurrent, not simultaneous?*

Exercise 5: *Can you think of a couple of ways to modify the four functions to get the four Ozobots to finish at closer to the same instant of time? Try out your modified program.*

Here’s a final quick note. You can change the name of devices such as Evo_1 and Ari_2 to whatever you wish. Just click on the pencil icon, as shown circled in Figure 9. A text box will pop up for you to enter the new name for the device.

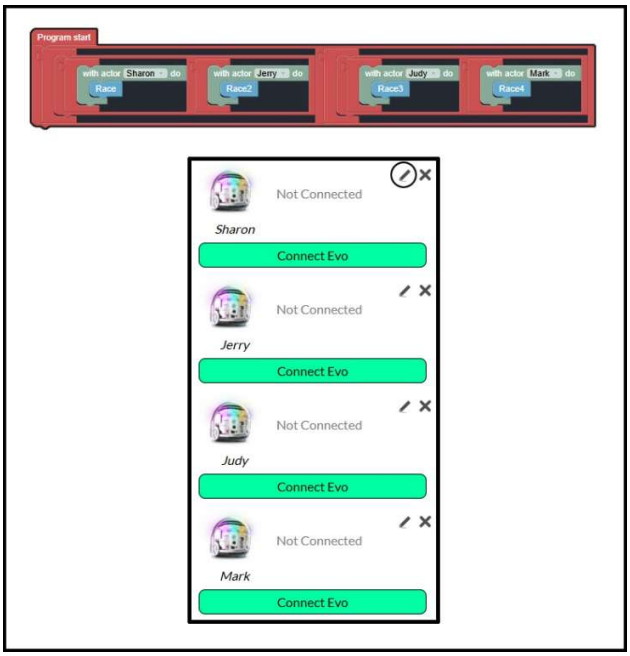
Program start
with actor Sharon do
Race
with actor Jerry do
Race2
with actor Judy do
Race3
with actor Mark do
Race4
Not Connected
Sharon
Connect Evo
Not Connected
Jerry
Connect Evo
Not Connected
Judy
Connect Evo
Not Connected
Mark
Connect Evo

Figure 9