TEACHER – OzoBlockly Stair Climb Challenge

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

In this teacher guide, the author's OzoBlockly solution to the "Stair Climb" challenge is discussed in detail. The complete code is shown in Figure 1. The main program appears in the upper left corner. It basically consists of five consecutive calls to functions (subroutines). It begins with some initialization, then Evo climbs up the stair, he turns around, climbs down the stair, and turns around again before stopping and turning off.

Values for the program variables are all initialized in the *Initialization* subroutine, making it easy to change them since they are all in one place in the program. For example, if the Ozomap had 5 stairs instead of 4, we could quickly change the value of the variable *numberOfStairs* to 5. This technique becomes particularly important when the length of a program is large. The rotation angle and speeds are also defined in the *Initialization* subroutine. The program begins with the top light red, gives the user 2 seconds to place Evo on the Ozomap, and then the top light turns green as Evo starts to move.

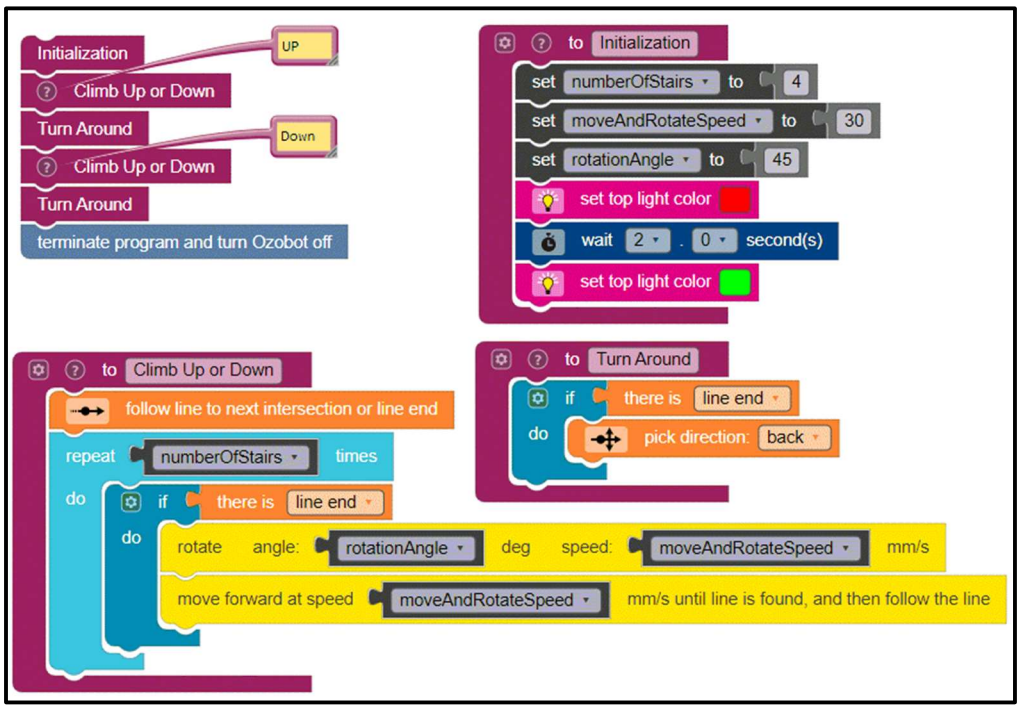

Figure 1

You will notice that the same function is used to both climb up and climb down the stairs. There is no need to make separate functions to climb up and climb down. Figure 2 helps clarify the reason for this. In this Figure, the Ozomap is shown both right side up and upside down. The stairs appear identical—so a single subroutine should easily handle both going up and going down the stairs.

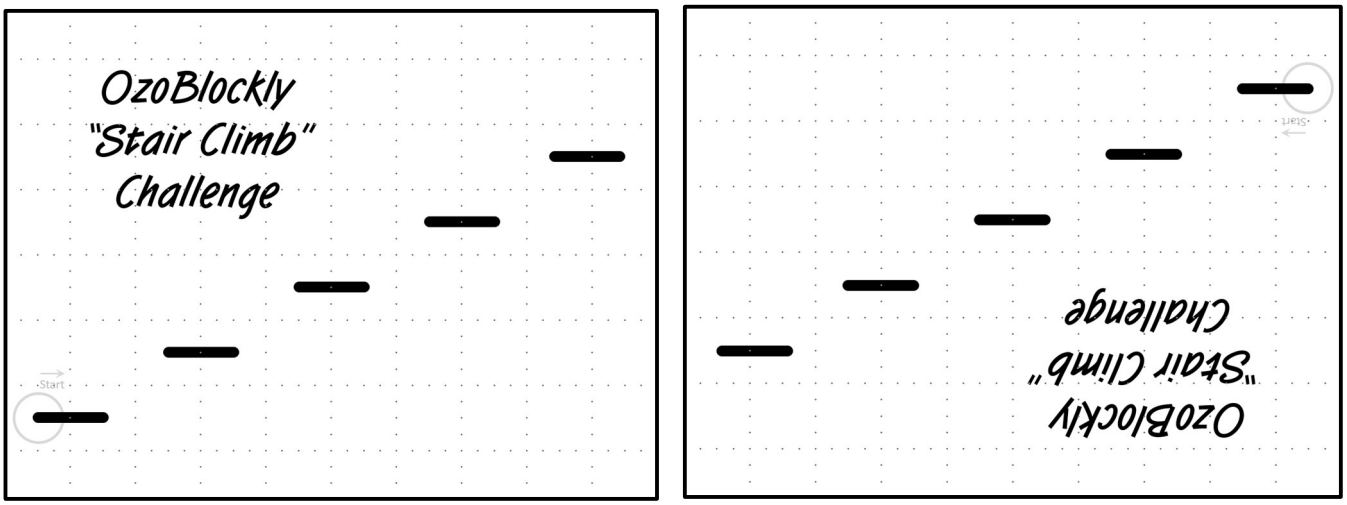

Figure 2

The *Climb Up or Down* subroutine begins with Evo on a landing, either the top or bottom landing of the staircase and facing the staircase. The orange *follow line to next intersection or line end block* brings Evo to the edge of the landing. The edge of the landing is a line end, so Evo starts a loop which repeats *numberOfStairs* times. Each time through the loop, Evo first rotates 45^{o} so that he is facing the edge of the next step. Then he moves forward until the line is found and follows the line. When the end of the landing is reached, Evo executes the *Turn Around* subroutine, which has him facing the staircase once again.

Note that a dotted grid is found on the Ozomap. The purpose of this grid is to give students a way to determine the angle through which Evo needs to turn each time to reach the end of the next step. There is no need to use a protractor. Evo needs to go 4 dots to the right and 4 dots up. This means that he must turn through an angle of 45^{o}. Note that the angle that Evo turns is dependent on a variety of things including the nature of the surface upon which Evo is moving. Therefore, the 45^{o} angle may need to be tweaked a bit.

In conclusion, we see that this solution to the *Stair Climb* challenge, involves the use of several computer science programming concepts:

- The use of variables to hold the values of program constants
- Subroutines to break up code into logical segments
- The use loops to handle tasks that must be repeated over and over
- The use of conditionals as needed to control the flow of logic
- Subroutine and variable names that support self-documentation
- Nesting of structures (*if* inside a *loop*)

In addition, the student gets a good workout on transitioning from free movement to line following.

Ways to Use This Challenge with Your Students

1. At the simplest level you can present the author's solution to the students and help them do a code review similar to that presented above.
2. You can present the challenge to the class without any indication of how to code it in OzoBlockly—and then let the students come up with their own solution. Students could then present their solutions to the class for a discussion of its advantages and disadvantages. (Students may even come up with a "better" solution than the author's solution! All the power to them!)
3. You could present the challenge to the students specifying that they must make use of the six programming concepts discussed in the previous section of this document.

Regardless of the approach that you choose, it is suggested that students view the accompany video that shows Evo perform the stair climb. This will give them a chance to view the ultimate goal of their work.

If class time is an issue, you may want to assign coding the challenge as a homework project and then have the students present their solutions as part of a class discussion.