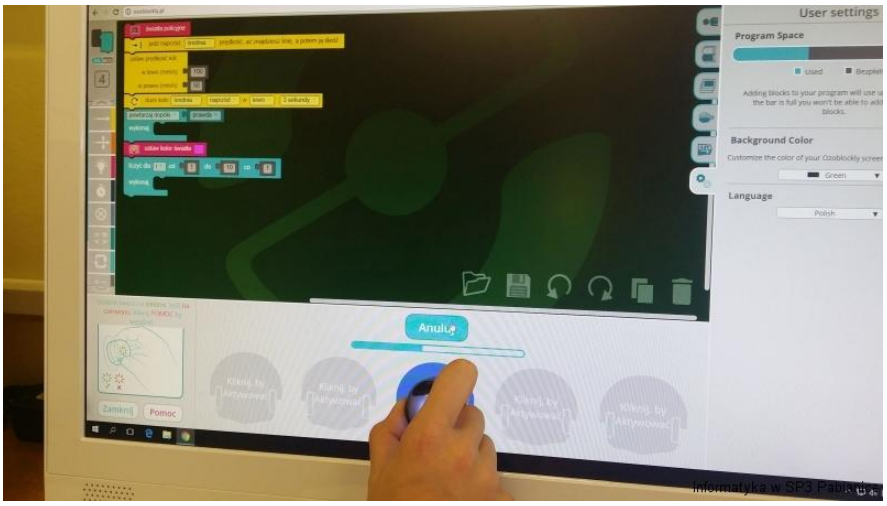
Ozobot Bit – the Fibonacci traveler

Coding Method:Color Codes, OzoBlockly
Subjects/Topics: Computer Science, Math
Grade Level:12
Duration:60 min
Required Ozobot Version: bit, evo

After completing Project No. 1 by Monika Schwarze "Cirkles, cirkles, cirkles, ...", the students learned the sequence of Fibonacci numbers.

Continuing the topic of Fibonacci numbers, students on the platform www.ozoblockly.pl (ozoblockly.com) worked with the code, and then with the Ozobot.bit educational robots.

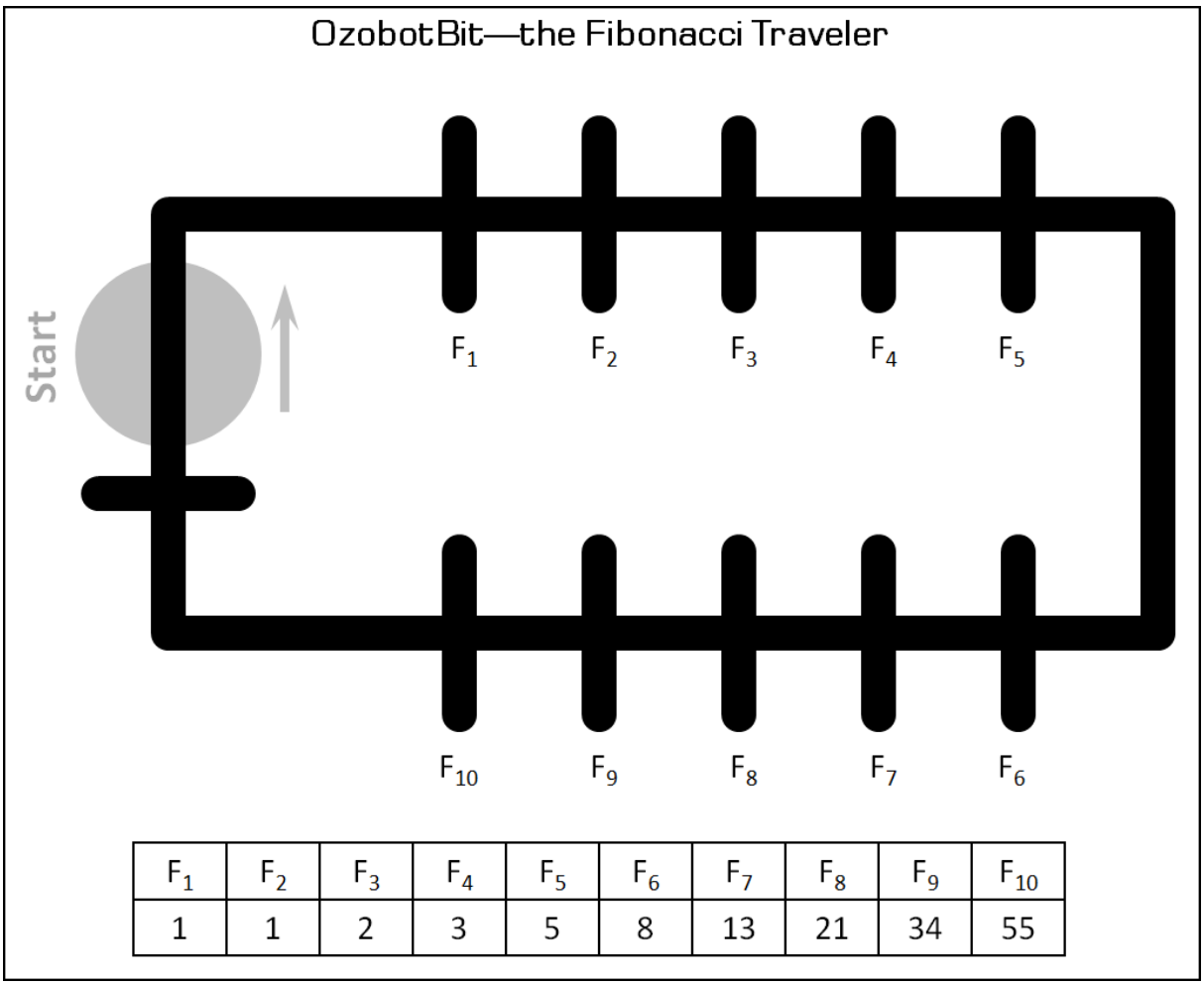


Challenge Requirements

Code - https://ozoblockly.com/editor?lang=en&robot=bit&mode=2#wx4bgo

1. www.ozoblockly.com - use Mode 4 (Advanced).
2. Ozobot Bit will start on the gray circle in the direction shown by the arrow.
3. Ozobot Bit will generate a random number between 1 and 10, as he will be computing only Fibonacci numbers between F_1 and F_{10}.
4. Anytime that Ozobot Bit is in motion on the maze, he will show a solid BLUE light.
5. Anytime that Ozobot Bit reaches an intersection, he will go straight through and not turn.
6. If Ozobot generates the random number 7, for example, then he will stop at the intersection labeled F_7, and will blink GREEN 13 times since $F_7 = 13$.
7. He will then start moving again (with a solid BLUE light), continue to the “finish” line, cross it, and repeat the cycle over-and-over, ad infinitum, or until you remove him from the maze or his battery is dead..

9. Be sure to calibrate Ozobot Bit on paper before starting the maze.



F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}
1	1	2	3	5	8	13	21	34	55

Description of creating the code by Richard Born Associate Professor Emeritus Northern Illinois University

One of the most fascinating sequences in the study of mathematics is the Fibonacci sequence, named after a 12th century Italian mathematician. The first two numbers in the sequence are both defined to be 1. Every number thereafter is the sum of the two preceding numbers. The first ten numbers in the sequence then become as shown in Figure 1. We see, in general, that for n > 2,

$F_n = F_{n-1} + F_{n-2}$. *(Equation 1)*

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	…
1	1	2	3	5	8	13	21	34	55	…

Figure 1

Although there are many algorithms to compute the numbers in the Fibonacci sequence, in this challenge lesson we are going to use a technique known as recursion. This is by no means the most efficient way, but it will help you to learn how

to use OzoBlockly functions and find out what recursion is. **With recursion, a function calls itself**. If you studied the Depth First Tree Traversal example on OzoBlockly.com, you saw recursion there; the function (or procedure) searchSubtree called itself twice.

Equation 1 above for the Fibonacci sequence indicates that we could use recursion to compute successive numbers in the sequence. We can find the n^{th} Fibonacci number if we know the $(n-1)^{st}$ and $(n-2)^{nd}$ numbers in the sequence. *Figure 2* shows OzoBlockly code for a recursive function that can be used to compute Fibonacci numbers.

Figure 2

We have named the function **fibonacci**. The function has an input we call **x**, which is the index number of the term that we wish to compute in the Fibonacci sequence. The function consists of three separate if-return statements. If x is one, we simply return the value 1, which is by definition the value of the first term in the sequence. If x is two, we again return the value 1, which is by definition the value of the second term in the sequence. If x>2, we use recursion to call **fibonacci**$_{x-1}$ and **fibonacci**$_{x-2}$, which are added together. Note that once a function has returned a value, no further statements within the function are executed. Also note that the OzoBlockly code "to fibonacci with x" is the **function** while the OzoBlockly code "fibonacci with x" is a statement that **calls** the **fibonacci** function. You are encouraged to use the **fibonacci** function of Figure 2 in programming your solution to this challenge.

To get a feel for how the recursive **fibonacci** function works, you can study the recursive calling tree for the sixth term in the sequence F_6. This tree appears in *Figure 3*. Note that each of the leaves shown in green in the figure returns the value 1. With eight such leaves, the value of F_6 then turns out to be 8. There are five levels of recursion. You can imagine the inefficiency of this process as the value of n increases in computing Fn. Yet recursion does have a beautiful simplicity in its concept, and we can certainly program Ozobot Bit using this recursive algorithm.

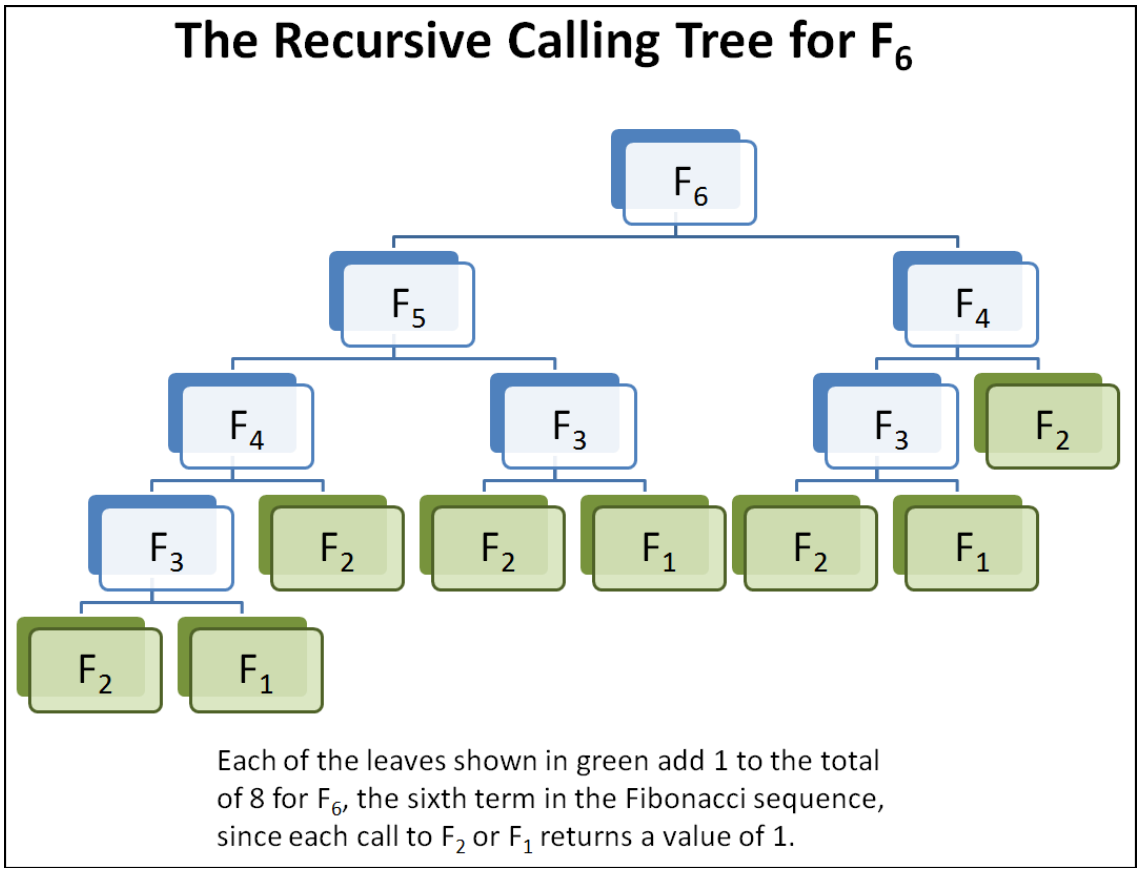
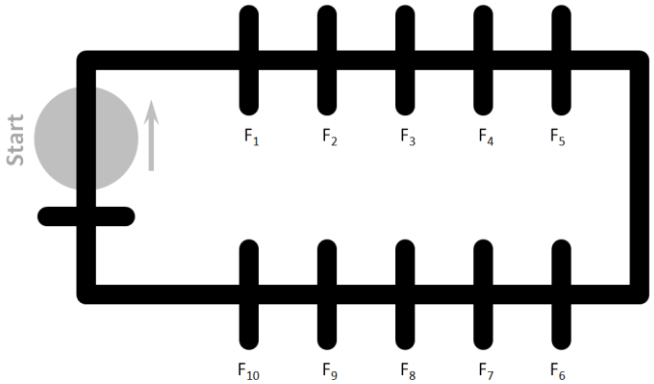


Figure 3

So how can we turn Ozobot Bit into a "Fibonacci Traveler"? For ready reference Figure 4 shows the maze that we will use.



F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}
1	1	2	3	5	8	13	21	34	55

Figure 4

Have fun with this challenge lesson! You will have a great feeling of accomplishment when you have turned Ozobot Bit into a perfect "Fibonacci Traveler"!