OZOBLOCKLY SKILLS 5

Turn Accurately

ESSENTIAL QUESTION

How can we make Ozobot turn accurately without a line?

OVERVIEW

Just like Ozobot Bit (and to a lesser extent Evo) gets help keeping straight, it can also be programming to make an exact 90 degree turn. By discovering how to tune Ozobot in detail, students will develop troubleshooting skills in computer science and gain deeper control of their Ozobot's abilities.

This lesson directly follows *OzoBlockly Skills 4.* Students will program turns with the advanced mode, and can optionally integrate functions from Lesson 4. The culminating activity challenges students to use their new movements and turns to draw an exact right angled figure, as if Ozobot is on a catwalk showing off its moves!

LESSON OUTLINE

1. Students learn how to use the "rotate XX degrees" from Mode 4 to bias the wheel movement in order to make Ozobot turn a true 90 degrees.
    a. Optionally, students can discover how to make a function out of their program. This function can then be created instead of the OzoBlockly "rotate left/right" command when it is essential that Ozobot turns 90°.
2. Students are challenged to design OzoBlockly code wherein Ozobot returns to the exact original place on the grid. Creativity is encouraged!

PREREQUISITES

OzoBlockly Skills 1-4

GROUPING

Two to three students

GRADE LEVEL

Grades 2 through 12

MATERIALS

- Tablet or computer with OzoBlockly editor ozoblockly.com/editor
- Ozobot Bit or Evo, one per group
- Printouts of the grid, one per group
- Pencil to mark Ozobot's path

OZOBLOCKLY PROGRAMMING TOPICS

Free Movement, Functions (optional)

OZOBLOCKLY MODE

Modes 3 and up

DURATION

30 minutes, 50 with extension

VOCABULARY

- *Ozobot Bit and Evo* - Little robot that can follow drawn lines, be programmed using visual codes, or through the OzoBlockly programming language
- *OzoBlockly* - A visual editor which allows you to create programs by plugging blocks together. The blocks can be used to control Ozobot's behavior like movement, LED lights, etc.
- *Pair Programming* - two programmers work together, one as the Driver (at the computer) and the other as the Navigator (giving ideas and directing the course).
- *Rotate left/right*– turn 90° left or right
- *Function* - a computer programming concept wherein a code that is needed often is saved under its own name to be used any time
- *Function Call* – the code block that represents the function and is used in the main program.

QUESTIONS ABOUT THIS LESSON?

Please contact us at ozoEdu@ozobot.com

LESSON

PART 1: GUIDED CLASS ACTIVITY

Your Ozobot can now go forward, but can it turn accurately?

The best way to see if the Ozobot deviates from a straight line is to have a grid to check it against. Hand out the grid page from below to each group.

Let's get started:

STEP 1

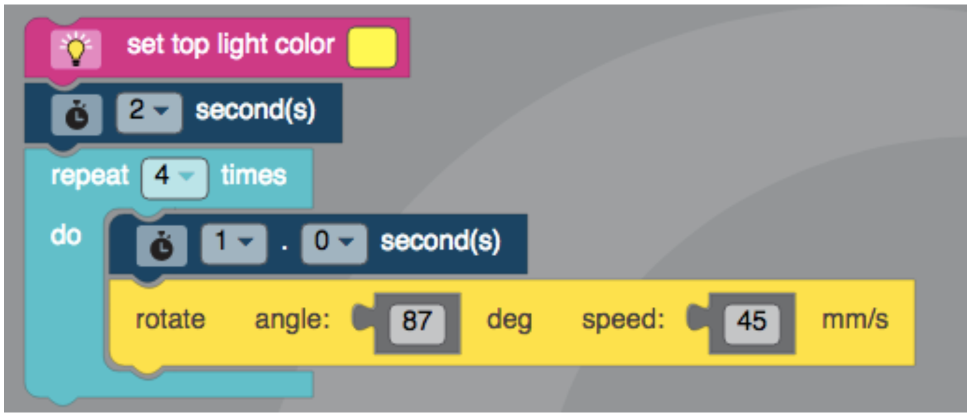
Begin with placing students in **Pair Programming** pairs or groups.

When you see this, ROLES SWITCH have your pair programmers switch Pair Programming roles.

Your students know **rotate right** and **rotate left** from **Movement** in Mode 3. Just like we can make wheel speeds specific, we can make turns specific. Start students off with Mode 4 **Movement**, use **rotate angle: 90**. Make speed 45, as it was with our "go straight" lesson.

The first test of turning bias comes with a simple turn code. Have your students turn their Ozobot 90 degrees 4 times. Does it face the same way?

Now allow students to troubleshoot by changing "90" to other integers. They should get Ozobot facing exactly the same way!



```
set top light color
2 second(s)
repeat 4 times
do
    1 . 0 second(s)
    rotate   angle: 87 deg   speed: 45 mm/s
```

FYI a turn has a maximum of 127 and -128 minimum in OzoBlockly

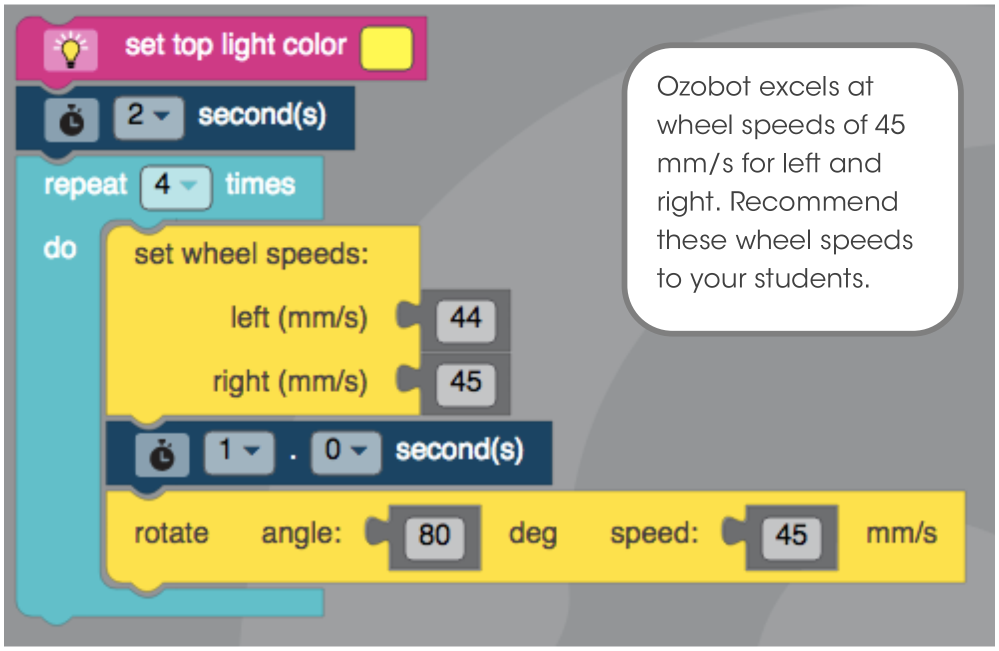
ROLES SWITCH

Step 2

Sometimes, the forward movement will affect the degree of the turn.

Have students code a straight movement from Lesson 4, testing on the grid to get as straight as possible. They could make the goStraight function if they know how.

Let's see what happens when we include our new right angled turns to draw a square with our new straight movement. Mark the starting square for the Ozobot on the grid, then try the following:



```
set top light color
2 second(s)
repeat 4 times
do  set wheel speeds:
        left (mm/s) 44
        right (mm/s) 45
    1 . 0 second(s)
    rotate angle: 80 deg speed: 45 mm/s
```

Ozobot excels at wheel speeds of 45 mm/s for left and right. Recommend these wheel speeds to your students.

We use the time delay to have time to place Ozobot straight on the grid!

Did the students' Ozobots return to the same original square, and facing the same way?

Ozobot may need an adjusted amount of degrees. Have students tweak the code until Ozobot faces the same way!

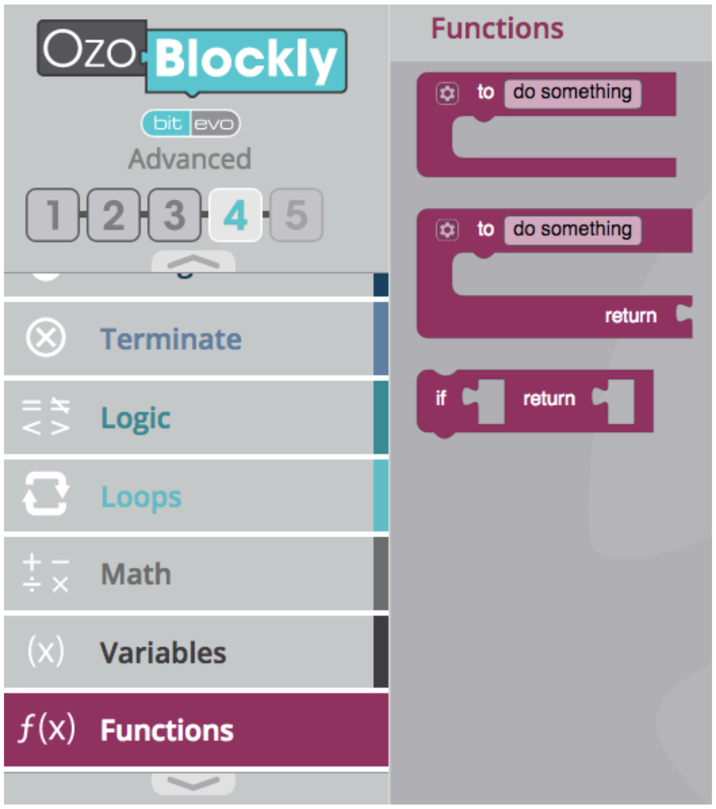
How to code U-turns and right turns: to turn right, set 90 to negative 90, "-90". This will turn the opposite way. How to turn 180? Blockly won't let me! It's okay, just call turnRight or turnLeft twice (or put it in a repeat 2 times loop).

*If students used **functions** in the last lesson, then you can also add in OPTIONAL Step 3 below. Students will make functions for turnLeft and turnRight. They could also code a U-turn.*

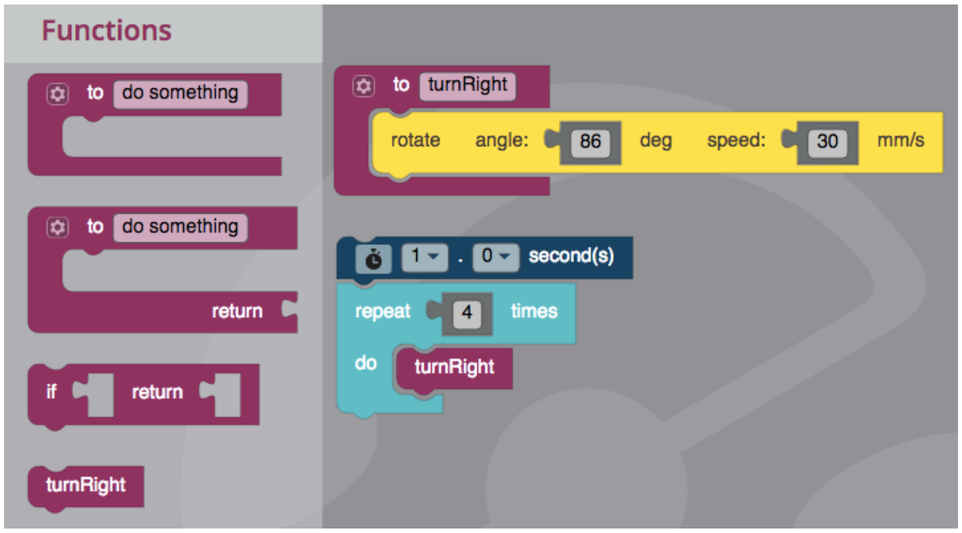
ROLES SWITCH

OPTIONAL Step 3

Remember: to make a function, go to OzoBlockly Mode 4, scroll to Functions, and choose the first option for 'to do something'. Change the name.



Once you make a function, that name appears in the list of Functions' options. You just made a new temporary command! Now, you can use it in your main program.



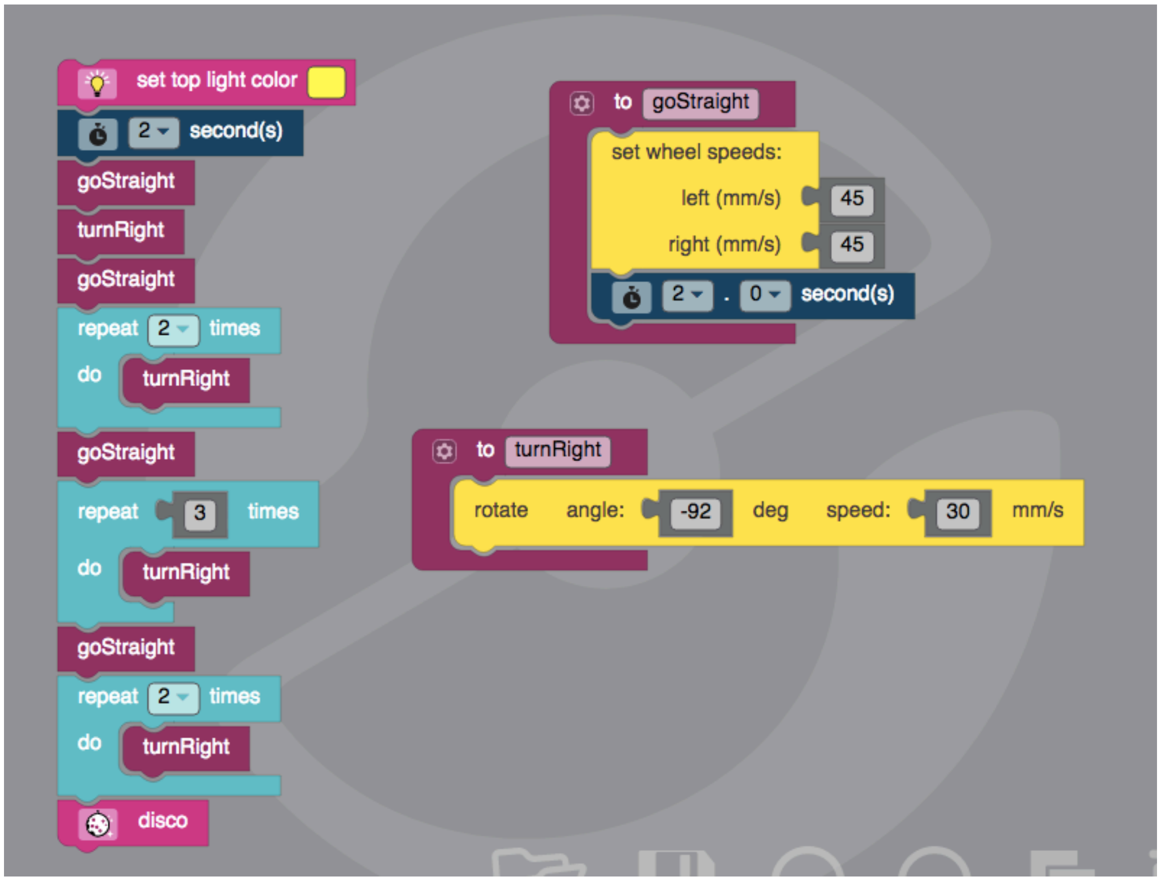
Left: the new turnRight function call. Right top: the turnRight function.
Right bottom: the program using turnRight function call

PART 2: STUDENT CHALLENGE

The students have all the tools they need to help Ozobot strut its stuff on the catwalk. **Pair Programming** students can switch roles at regular intervals or when they're ready, but everyone should have a chance to perform each role.

Students outline (and even decorate) their grid in a right-angled shape like a T or cross shape, or even an L, to show how accurate their bot is now. Teacher could even randomly assign right angled shapes for each group.

Then, run through the program, debugging as necessary. When the Ozobot is ready to perform, students can demonstrate their code running on their Ozobot.



```
set top light color
2 second(s)
goStraight
turnRight
goStraight
repeat 2 times
  do turnRight
goStraight
repeat 3 times
  do turnRight
goStraight
repeat 2 times
  do turnRight
disco
```

```
to goStraight
  set wheel speeds:
    left (mm/s) 45
    right (mm/s) 45
  2 . 0 second(s)
```

```
to turnRight
  rotate angle: -92 deg speed: 30 mm/s
```

Ozobot Catwalk on an L-shaped platform! This code uses functions.
For an example without functions, replace the magenta blocks on the left
with the corresponding blocks from inside function block on the right. Solutions will vary.

ozobot Testing Grid