HOW TO PROGRAM ROBOTS

ESSENTIAL QUESTION

How can students program robots with an easy-to-use block-based code editor?

OVERVIEW

This lesson is specially designed for the Hour of Code™ event to show students how easy it can be to program and use the robots called **Bit** and **Evo** by Ozobot through completing game levels. **You don't need robots to complete this lesson**. Students will play games that are designed to show students how to use a block-based editor, OzoBlockly, to program a simulated Bit or Evo, our line-following, color sensing robots that measure at only 1 cubic inch!

While playing two consecutive 10-level games, students learn how to command a simulated robot to move, set the LED light and use Loops (a basic computer science concept). For more information on how to use these robots in your classroom for computer science and integration into all subject areas, such as math, art, language, and science, visit ozobot.com/stem-education/stem-classroom-kit.

LESSON OUTLINE

1. *(Optional) Students create a papercraft robot to help them make the connection between sequential coding logic and a robot's actions.*
2. In Shape Tracer 1, students play the tutorial levels 1-10 to introduce them to OzoBlockly and programming concepts.

   Concepts:
   - Visual programming basics
   - Movement forward
   - Turn or rotate
   - Setting LED colors
3. Shape Tracer 2 follows the same format as Shape Tracer 1

   Further concepts:
   - Loops
   - Nested Loops (optional)

PREREQUISITES

No prior knowledge of coding or Ozobot robots is required.

GRADE LEVEL

Grade 2 and up. Older students may move more independently through the levels of the tutorials. The tutorials are appropriate for all levels to learn the OzoBlockly programming language.

GROUPING

Individuals or groups of two or three students

MATERIALS

- Tablet or computer for each student to access Ozobot Games games.ozoblockly.com
- *(Optional) Print outs of the papercraft robot, one for each student. The large size is intended for young students and the small size is helpful for saving paper.*
- *(Optional) Graph paper for students to track their "robot" movement.*

OZOBLOCKLY PROGRAMMING TOPICS

Free Movement, Light Effects, Loops

DURATION

1 hour

VOCABULARY

- *Bit or Evo by Ozobot* - Little robots that can follow drawn lines or be programmed using drawn codes or through the OzoBlockly language.
- *OzoBlockly* - A visual editor which allows to create programs by plugging blocks together. The blocks can be used to control the bot's behavior like movement, LED lights, etc.
- *Rotate Right or Left* – Approximate 90° turn
- *Rotate Slightly Right or Left* – Approximate 45° turn
- *Steps* – a unit of movement forward or backward

QUESTIONS ABOUT THIS LESSON?

Please contact us at ozoEdu@ozobot.com

LESSON

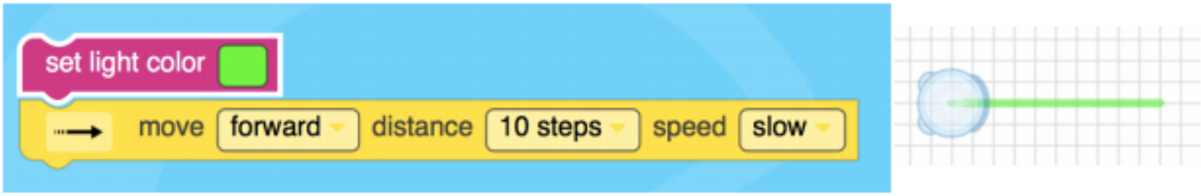
OPTIONAL PAPERCRAFT

Print out the papercraft outlines (the small version is for saving paper, and large is best for younger students) and let students color the blank sides, then fold and tape it. Tell students that they will simulate programs by flipping their paper robot to show the right color on top, and move it on the desk as their code says. Note: students should follow the code blocks, and not the picture in the simulator. Students can use graph paper to make measured movements with their paper robot.

1. SHAPE TRACER 1

LEVEL 1

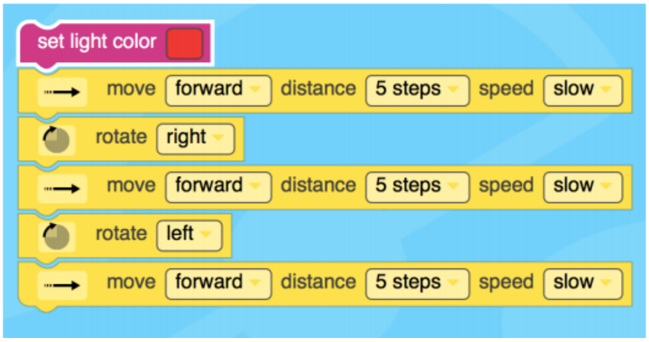
With all students on Level 1, guide them through the interface. First, the Goal pop-up tells students what they have to do. The image on the right is the simulator, and the simulated Ozobot robot must trace the line exactly. To build the program, click the left category menu, then the block you want. The blocks must click together to create a program. Students can click the drop-down menus or color squares in the blocks to change the settings of each block. To run the program, click "Run" under the simulator.



```
set light color 🟩
move forward distance 10 steps speed slow
```

LEVEL 2

Explain that Ozobot's robots can make turns. For this we use the "Rotate" block found in "Movement".



```
set light color 🟥
move forward distance 5 steps speed slow
rotate right
move forward distance 5 steps speed slow
rotate left
move forward distance 5 steps speed slow
```

Have the students experiment with movement and rotation. Question the students, "how far does the bot move when programed to turn a "slight left" versus a "left" turn?"

Discuss 45° rotation versus 90° rotation. Challenge the students to find how many degrees are in a U-Turn. Students solve Level 2.

LEVELS 3-10

After students have discussed and learned the basic movement and programming, **students can complete the 10 levels on their own.**

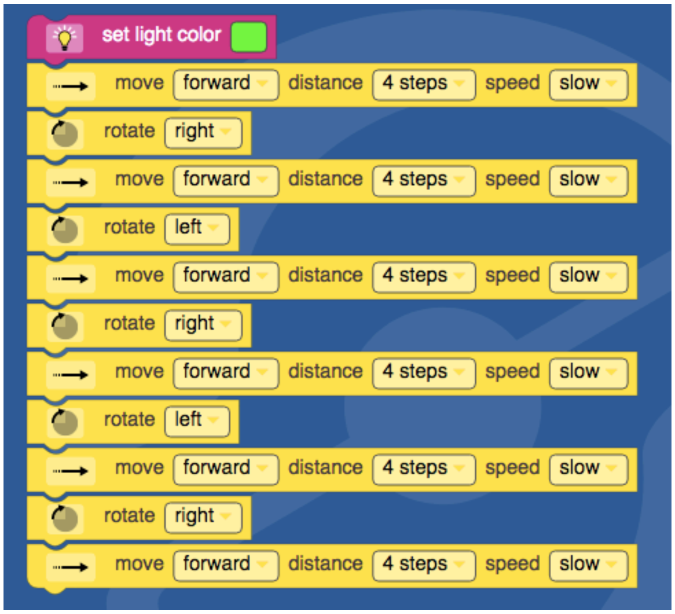
When you program Bit or Evo, you will get the programs onto the robot using flashing color from the screen. That's done using Ozobot's special Flash Loading.

How does **Flash Loading** work? The bottom of **Bit** and **Evo** has light and color sensors that it uses to see printed or drawn black lines, special drawn color sequences called OzoCodes, and the rapid colors from the screen. During Flash Loading, **Bit** or **Evo** memorizes the program built from the code blocks and can run that program anytime – just double-click the power button to have the bot execute it!

2. SHAPE TRACER 2

LEVEL 1

Ask the students for ideas about how they could get their Bit or Evo to trace the stair step shape. Write some ideas on the white board. Let students explore and find solutions to the puzzle. Many students will program like this:



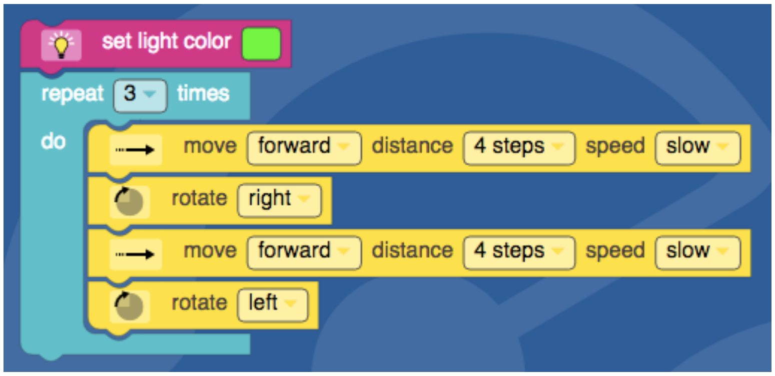
```
set light color 🟩
move forward distance 4 steps speed slow
rotate right
move forward distance 4 steps speed slow
rotate left
move forward distance 4 steps speed slow
rotate right
move forward distance 4 steps speed slow
rotate left
move forward distance 4 steps speed slow
rotate right
move forward distance 4 steps speed slow
```

Question
"Do you see any patterns in the order and types of blocks?"

Answer
"The blocks start to repeat; the same blocks are used over and over."

Demonstrate this program and ask the students if they see any combination of blocks that repeat.

Explain that this is the purpose of the "Loop" block. Any combination of blocks that repeat can be placed inside a loop block. All you need to do is tell the loop how many times to repeat the commands. The correct way to program this level looks like:



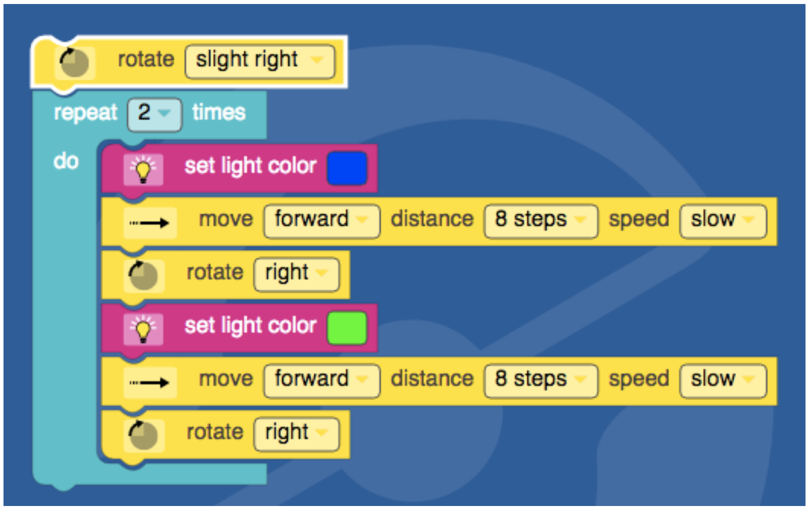
```
set light color 🟩
repeat 3 times
do
    move forward distance 4 steps speed slow
    rotate right
    move forward distance 4 steps speed slow
    rotate left
```

LEVEL 2

This level involves the same skills as level 1. Students solve level 2.

LEVEL 3

Begin this level by letting the students try and figure out the puzzle in groups or individually. After some time experimenting, ask the class what solutions they found. Many will have found solutions like this:



```
rotate slight right
repeat 2 times
do
    set light color 🟦
    move forward distance 8 steps speed slow
    rotate right
    set light color 🟩
    move forward distance 8 steps speed slow
    rotate right
```

Question
Why is using a loop a better way to program?

Answers
Because you use fewer blocks (lines of code) and do less work.

Because it is easier to change the program if you need to modify it.

Because you are less likely to make a mistake if you don't have a long list of blocks.

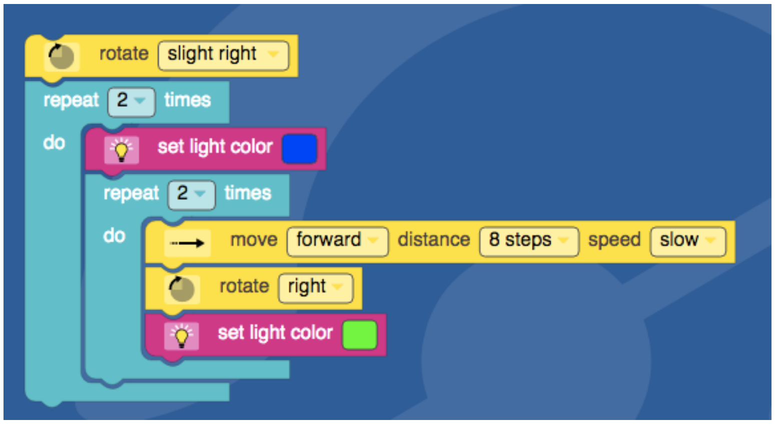
This is a perfectly valid solution and you can have students move on to the next level if you like. At this point though, you do have the option to introduce the concept of nested loops. There is a solution that is more efficient than the one above!

Optional: Understanding Nested Loops - *What are nested loops? Nested loops are constructed by putting one repeat block (Loop) inside another to use the fewest number of blocks to complete the task. This video from code.org briefly explains nested loops:* http://www.youtube.com/watch?v=2C0pdRhlZhk

When you create nested loops, the process of running the code begins in the outer loop. When the run hits the inner loop, all those commands are taken to completion before the run once again starts on the outer loop. The total amount of times the whole run will loop is the product of the numbers in both inner and outer loops.

Nested Loop Demonstration

1. *Let students take another look at the program above that uses only one loop. They will probably have realized that the commands almost repeat, but there is a slight difference in the commands that prevent them from using fewer blocks. The difference is the change in color.*
2. *Create this program to demonstrate to the class:*



```
rotate slight right
repeat 2 times
do  set light color 🟦
    repeat 2 times
    do  move forward distance 8 steps speed slow
        rotate right
        set light color 🟩
```

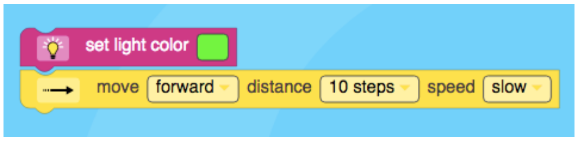
3. *Allow the students to watch yours play so they can see how the program runs through the blocks. Play the solution several times and ask volunteers to explain what is happening.*
4. *Have the students program their Ozobot simulator with a nested loop.*

LEVELS 4-10

After students have discussed and learned about movement and loops, allow students to **complete the 10 levels**.

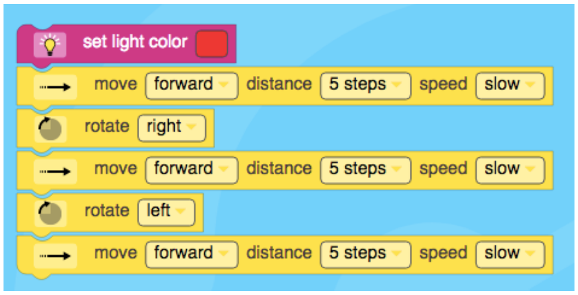
SHAPE TRACER 1 ANSWER KEY

Level 1



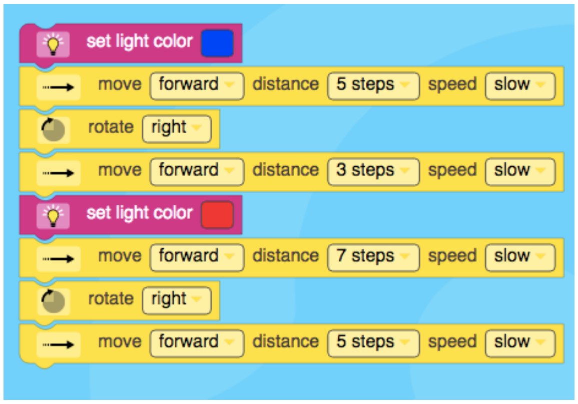
```
set light color 🟩
move forward distance 10 steps speed slow
```

Level 2



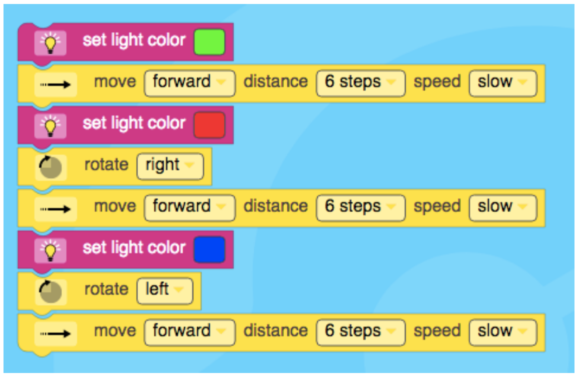
```
set light color 🟥
move forward distance 5 steps speed slow
rotate right
move forward distance 5 steps speed slow
rotate left
move forward distance 5 steps speed slow
```

Level 3



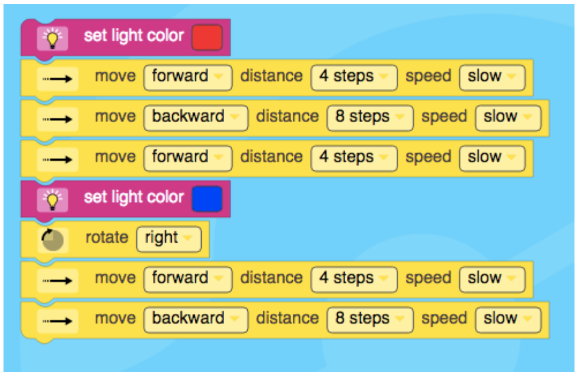
```
set light color 🟦
move forward distance 5 steps speed slow
rotate right
move forward distance 3 steps speed slow
set light color 🟥
move forward distance 7 steps speed slow
rotate right
move forward distance 5 steps speed slow
```

Level 4



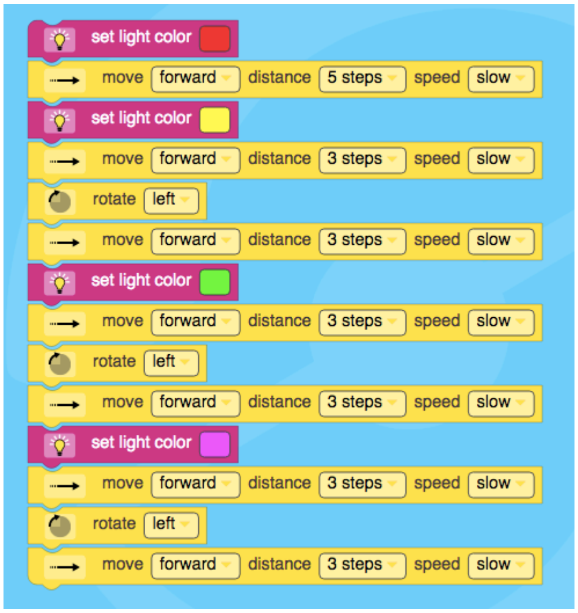
```
set light color 🟩
move forward distance 6 steps speed slow
set light color 🟥
rotate right
move forward distance 6 steps speed slow
set light color 🟦
rotate left
move forward distance 6 steps speed slow
```

Level 5



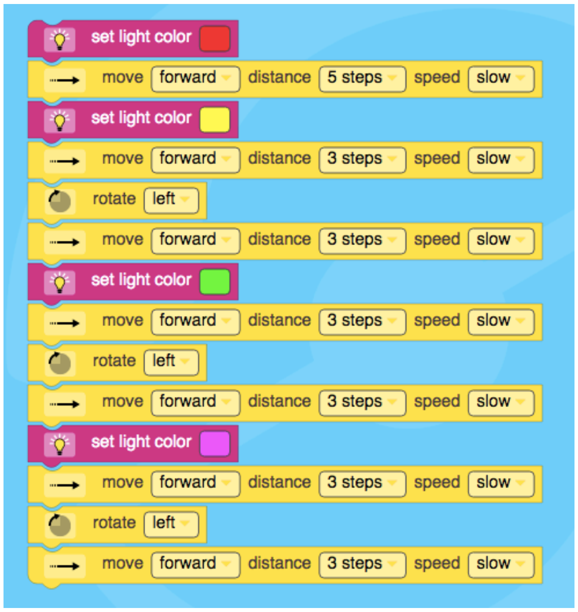
```
set light color 🟥
move forward distance 4 steps speed slow
move backward distance 8 steps speed slow
move forward distance 4 steps speed slow
set light color 🟦
rotate right
move forward distance 4 steps speed slow
move backward distance 8 steps speed slow
```

Level 6



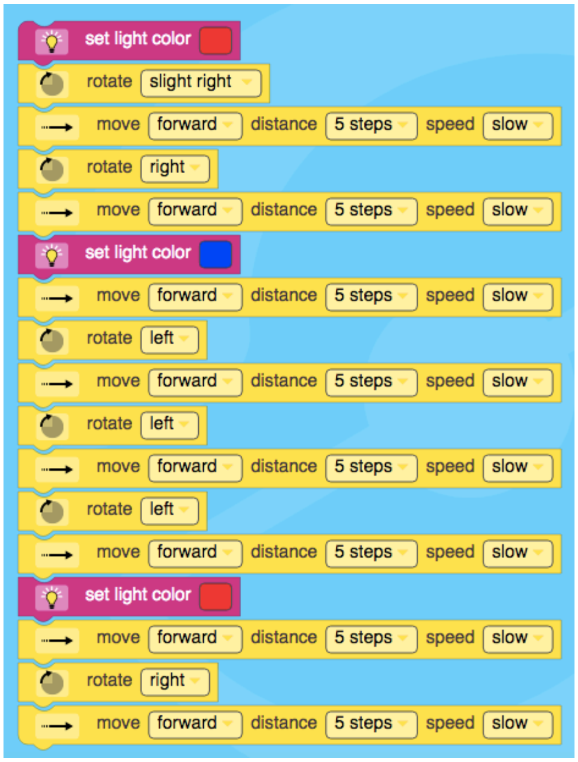
```
set light color 🟥
move forward distance 5 steps speed slow
set light color 🟨
move forward distance 3 steps speed slow
rotate left
move forward distance 3 steps speed slow
set light color 🟩
move forward distance 3 steps speed slow
rotate left
move forward distance 3 steps speed slow
set light color 🟪
move forward distance 3 steps speed slow
rotate left
move forward distance 3 steps speed slow
```

Level 7



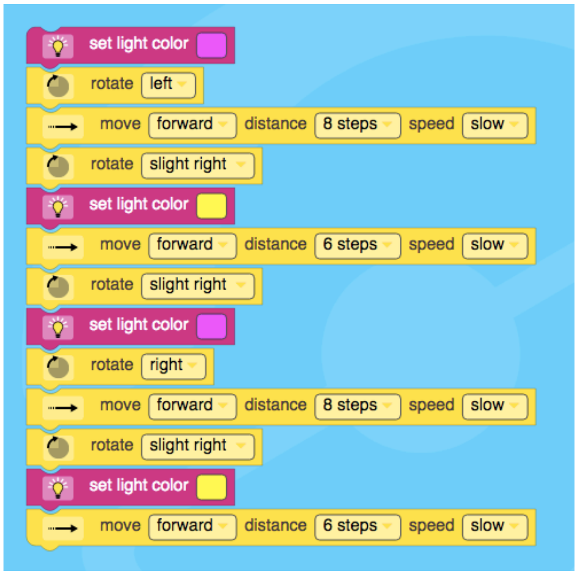
```
set light color 🟥
move forward distance 5 steps speed slow
set light color 🟨
move forward distance 3 steps speed slow
rotate left
move forward distance 3 steps speed slow
set light color 🟩
move forward distance 3 steps speed slow
rotate left
move forward distance 3 steps speed slow
set light color 🟪
move forward distance 3 steps speed slow
rotate left
move forward distance 3 steps speed slow
```

Level 8



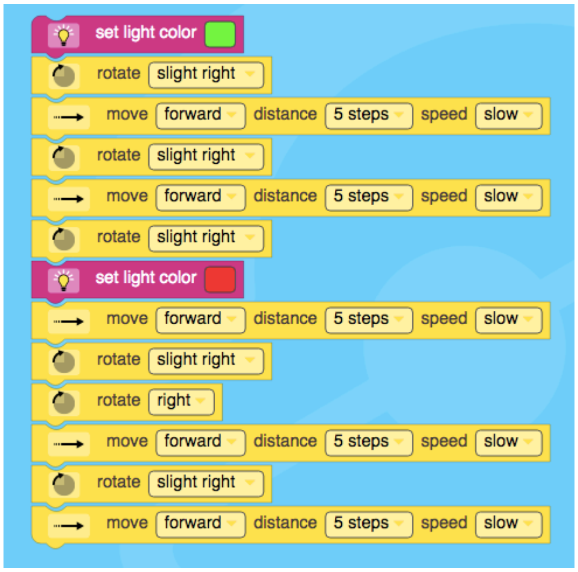
```
set light color 🟥
rotate slight right
move forward distance 5 steps speed slow
rotate right
move forward distance 5 steps speed slow
set light color 🟦
move forward distance 5 steps speed slow
rotate left
move forward distance 5 steps speed slow
rotate left
move forward distance 5 steps speed slow
rotate left
move forward distance 5 steps speed slow
set light color 🟥
move forward distance 5 steps speed slow
rotate right
move forward distance 5 steps speed slow
```

Level 9



```
set light color 🟪
rotate left
move forward distance 8 steps speed slow
rotate slight right
set light color 🟨
move forward distance 6 steps speed slow
rotate slight right
set light color 🟪
rotate right
move forward distance 8 steps speed slow
rotate slight right
set light color 🟨
move forward distance 6 steps speed slow
```

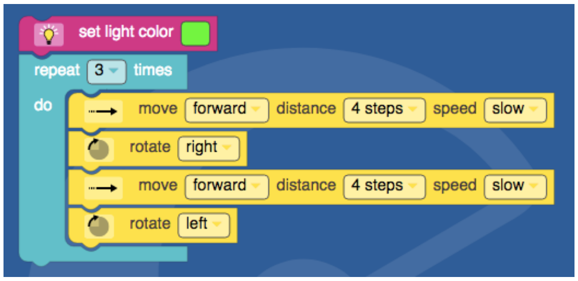
Level 10



```
set light color 🟩
rotate slight right
move forward distance 5 steps speed slow
rotate slight right
move forward distance 5 steps speed slow
rotate slight right
set light color 🟥
move forward distance 5 steps speed slow
rotate slight right
rotate right
move forward distance 5 steps speed slow
rotate slight right
move forward distance 5 steps speed slow
```

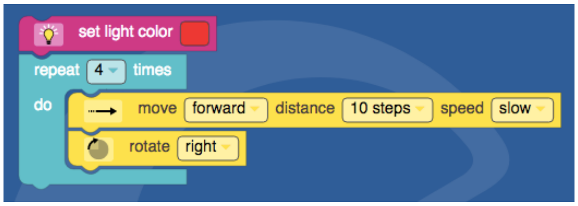
ANSWER KEY

Level 1



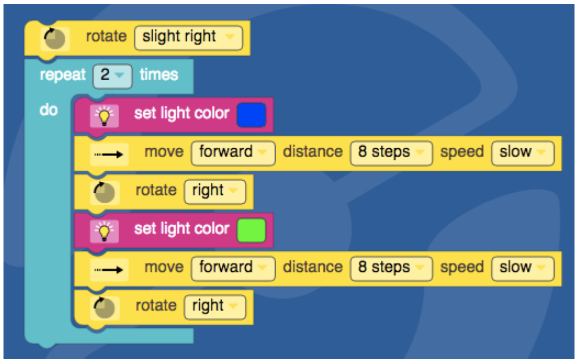
```
set light color 🟩
repeat 3 times
do
    move forward distance 4 steps speed slow
    rotate right
    move forward distance 4 steps speed slow
    rotate left
```

Level 2



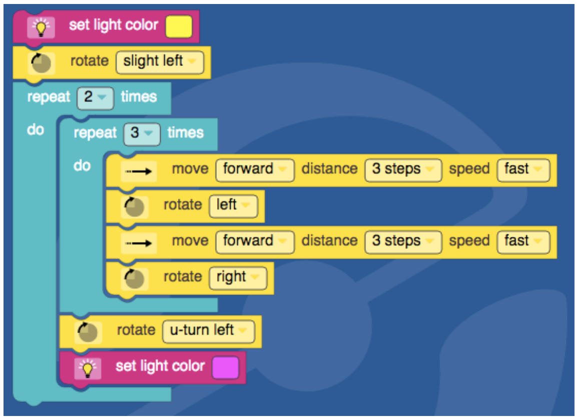
```
set light color 🟥
repeat 4 times
do
    move forward distance 10 steps speed slow
    rotate right
```

Level 3



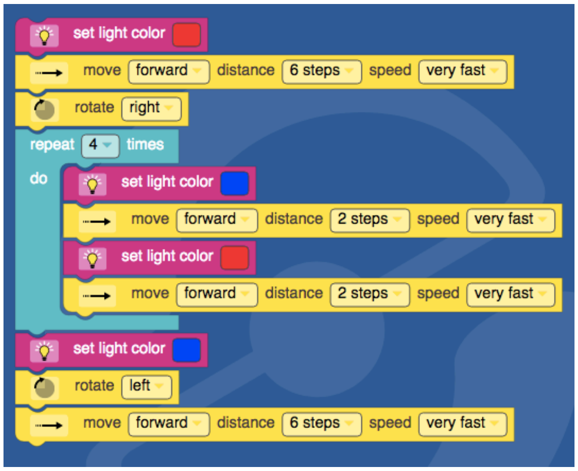
```
rotate slight right
repeat 2 times
do
    set light color 🟦
    move forward distance 8 steps speed slow
    rotate right
    set light color 🟩
    move forward distance 8 steps speed slow
    rotate right
```

Level 4



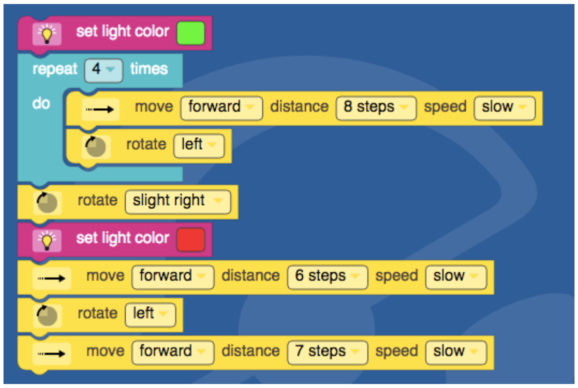
```
set light color 🟨
rotate slight left
repeat 2 times
do
    repeat 3 times
    do
        move forward distance 3 steps speed fast
        rotate left
        move forward distance 3 steps speed fast
        rotate right
    rotate u-turn left
    set light color 🟪
```

Level 5



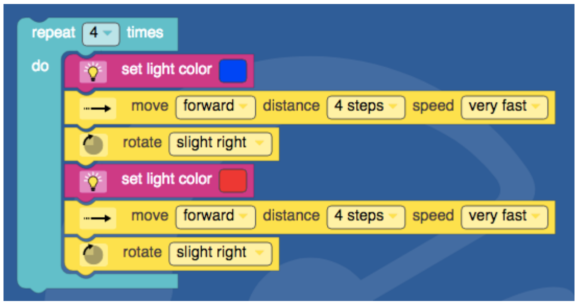
```
set light color 🟥
move forward distance 6 steps speed very fast
rotate right
repeat 4 times
do
    set light color 🟦
    move forward distance 2 steps speed very fast
    set light color 🟥
    move forward distance 2 steps speed very fast
set light color 🟦
rotate left
move forward distance 6 steps speed very fast
```

Level 6



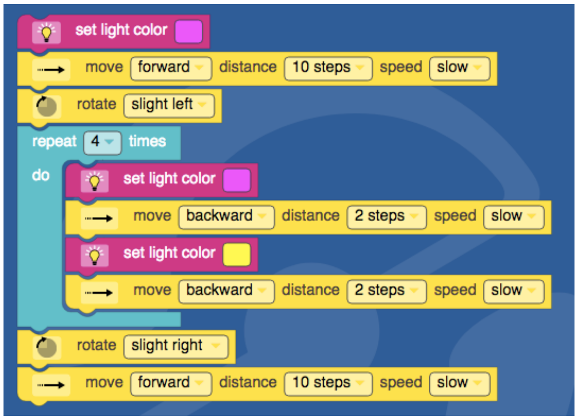
```
set light color 🟩
repeat 4 times
do
    move forward distance 8 steps speed slow
    rotate left
rotate slight right
set light color 🟥
move forward distance 6 steps speed slow
rotate left
move forward distance 7 steps speed slow
```

Level 7



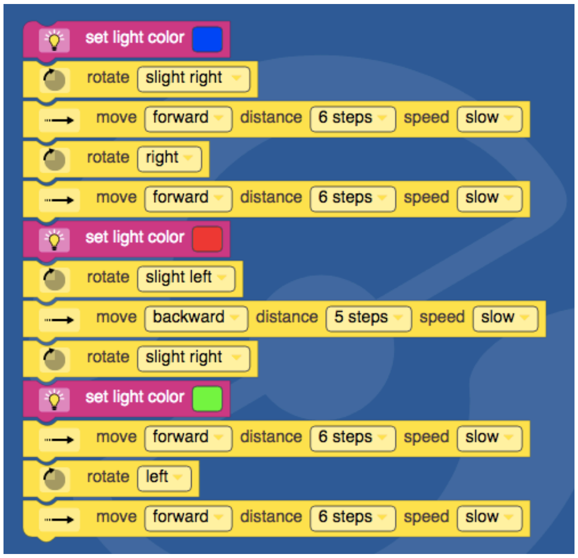
```
repeat 4 times
do
    set light color 🟦
    move forward distance 4 steps speed very fast
    rotate slight right
    set light color 🟥
    move forward distance 4 steps speed very fast
    rotate slight right
```

Level 8



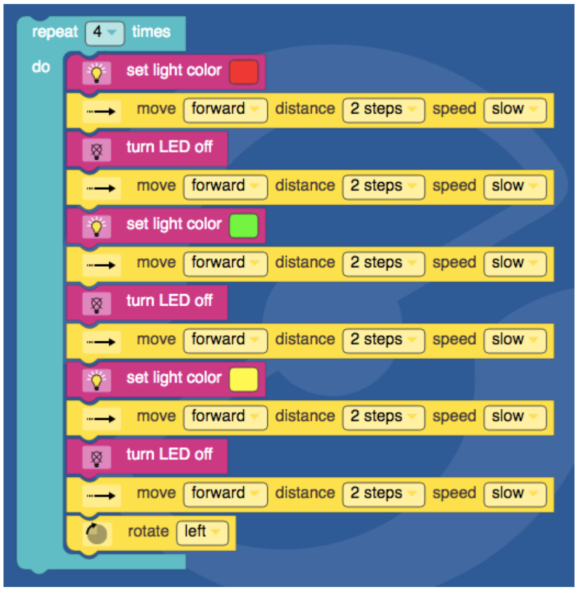
```
set light color 🟪
move forward distance 10 steps speed slow
rotate slight left
repeat 4 times
do
    set light color 🟪
    move backward distance 2 steps speed slow
    set light color 🟨
    move backward distance 2 steps speed slow
rotate slight right
move forward distance 10 steps speed slow
```

Level 9



```
set light color 🟦
rotate slight right
move forward distance 6 steps speed slow
rotate right
move forward distance 6 steps speed slow
set light color 🟥
rotate slight left
move backward distance 5 steps speed slow
rotate slight right
set light color 🟩
move forward distance 6 steps speed slow
rotate left
move forward distance 6 steps speed slow
```

Level 10

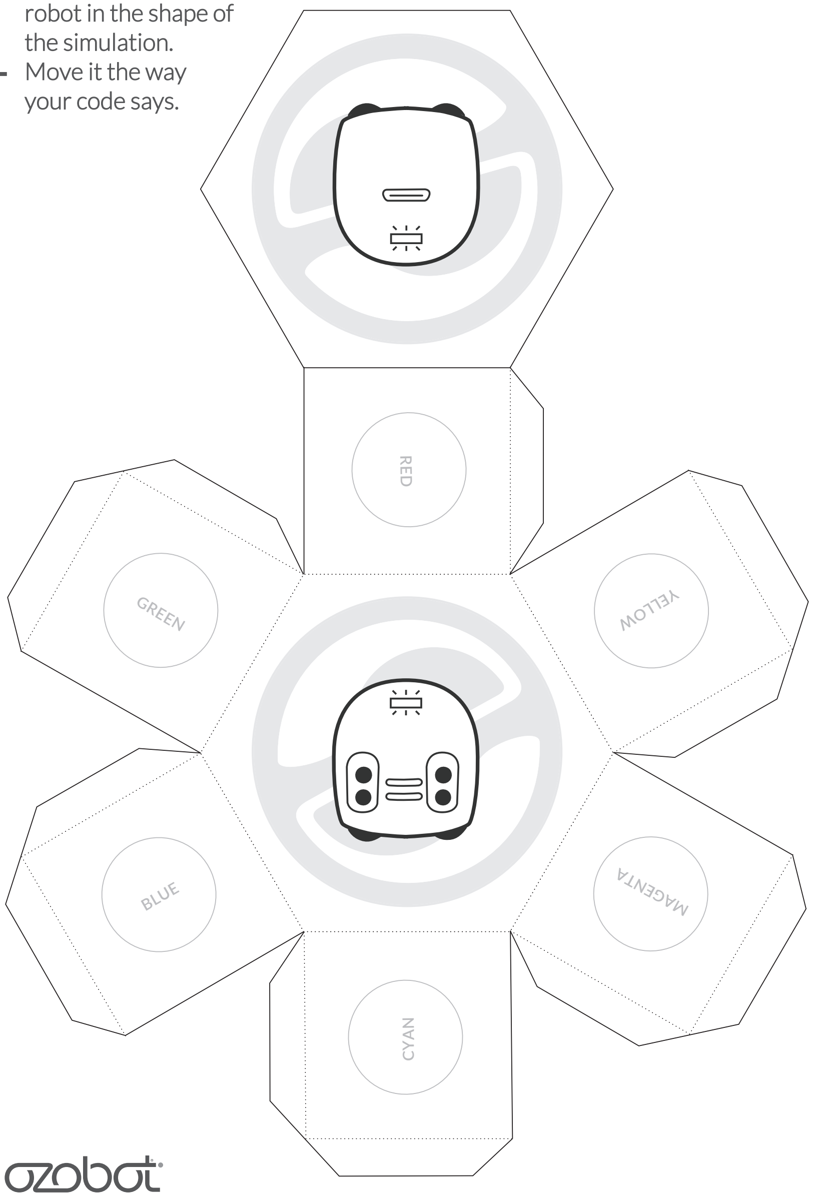


```
repeat 4 times
do
    set light color 🟥
    move forward distance 2 steps speed slow
    turn LED off
    move forward distance 2 steps speed slow
    set light color 🟩
    move forward distance 2 steps speed slow
    turn LED off
    move forward distance 2 steps speed slow
    set light color 🟨
    move forward distance 2 steps speed slow
    turn LED off
    move forward distance 2 steps speed slow
    rotate left
```

Papercraft Ozobot Robot

1. Cut along the outside lines.
2. Color the sides with crayons, markers or pencils.
3. Fold on the dotted lines.
4. Tape or glue edges together.
5. Move your robot by facing the front toward the direction your code blocks tell it to go, and flip it to the right LED color for each section.

TIP! Don't move the robot in the shape of the simulation. Move it the way your code says.

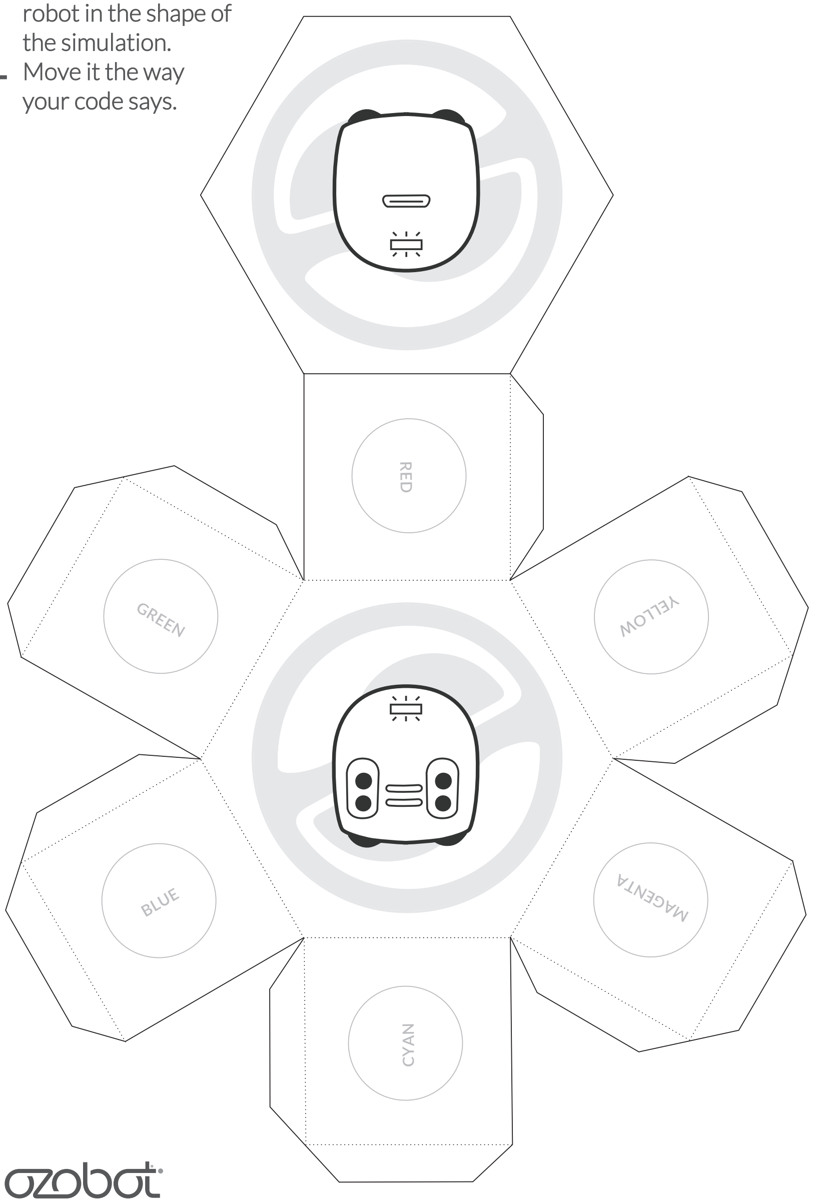


ozobot

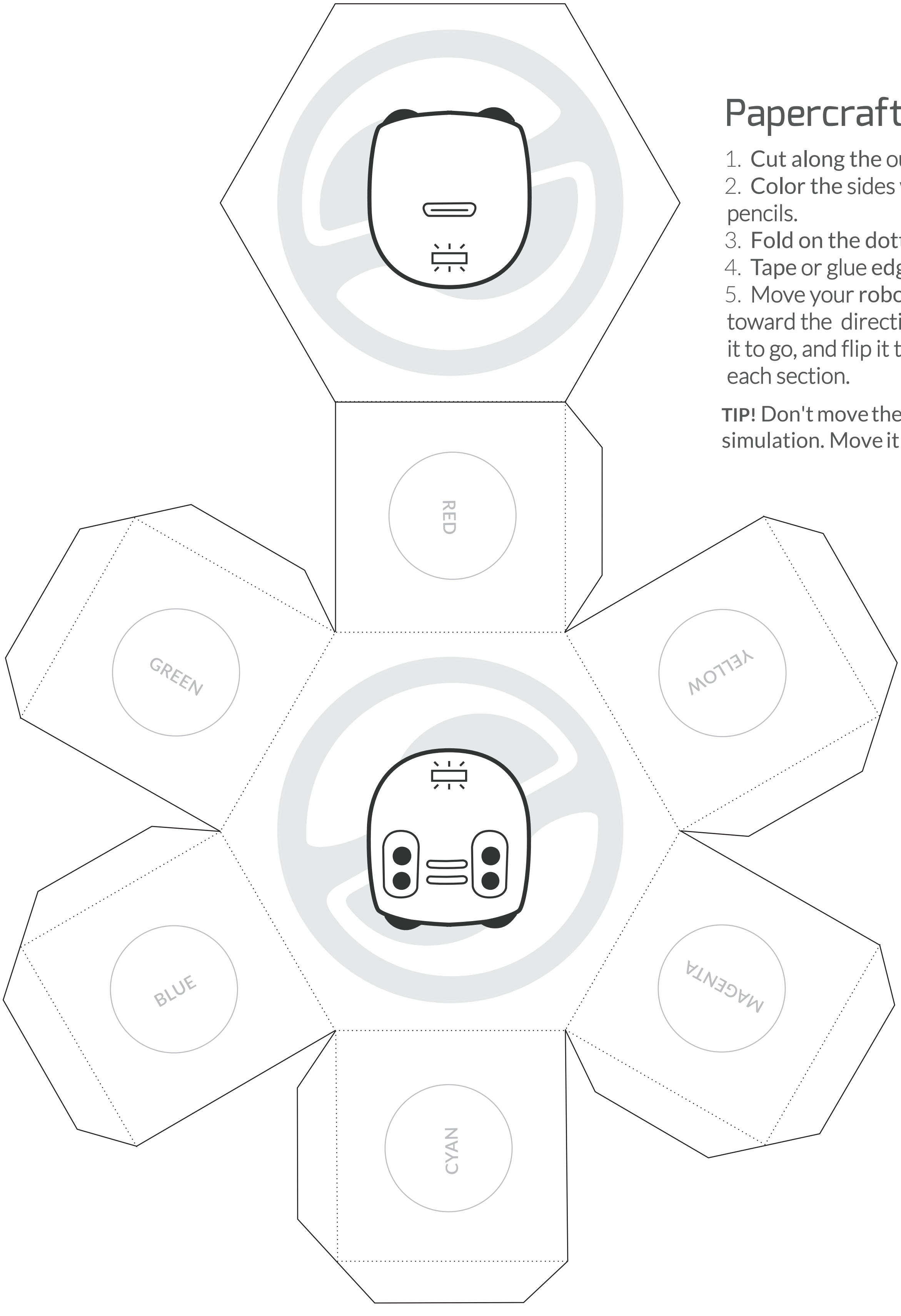
Papercraft Ozobot Robot

1. Cut along the outside lines.
2. Color the sides with crayons, markers or pencils.
3. Fold on the dotted lines.
4. Tape or glue edges together.
5. Move your robot by facing the front toward the direction your code blocks tell it to go, and flip it to the right LED color for each section.

TIP! Don't move the robot in the shape of the simulation. Move it the way your code says.



ozobot



Papercraft Ozobot Robot

1. Cut along the outside lines.
2. Color the sides with crayons, markers or pencils.
3. Fold on the dotted lines.
4. Tape or glue edges together.
5. Move your robot by facing the front toward the direction your code blocks tell it to go, and flip it to the right LED color for each section.

TIP! Don't move the robot in the shape of the simulation. Move it the way your code says.