



COLOR SEARCH

LESSON 1

Summary

Ozobot moves around a map by itself, ‘bounces’ off of colors, and ‘wins’ on the color red. Aim Ozobot right to get to the red the fastest.			
CS Topics: Loops and breaks, conditional logic, sensor input/output		Learning Outcomes: Use ‘repeat forever’ loops, Ozobot’s color sensor, and ‘self-driving’ movement.	
Grades: 2-5	Coding Level: Beginner	Map Style: Free Movement	Game Mechanic: winning

Introduction

Welcome to the first lesson in Ozobot’s [Elementary Computer Science with Game Design](#)! Students will quickly develop their coding knowledge and intuition by ‘deconstructing’ a game’s program to see what type of code creates each action, then create their own program. Leaders in computer science teach themselves new skills through exploring and modifying sample programs this way. These lessons are also great for teachers still learning coding, themselves. To learn more about this series of lessons, visit the [Elementary Computer Science, with Game Design README](#).

In this game called “Color Search”, students will discover how loops, if statements, and movement code blocks work together to make a game where Ozobot walks autonomously around a colored map until it lands on red. (All the info you need on the code is in ABOUT THE CODE, below.) The strategy for this game is to aim your bot to make it land on red the fastest.

After students see your demo of Color Search, the class will compare the actions of the bot to the program itself, then spend time modifying the movements and victory dance to improve the game. This engagement with the code helps students see the connection between loops, logic and movement, and give them the skill to build their own games that include the game mechanics of autonomous movement and winning.

If there is time, students can also experiment with making their own maps to use with their new program and challenge other students to play their map.

To get started, read each section of this lesson, below, before teaching the lesson. SETUP lists the preparation steps needed for this lesson, ABOUT THE CODE explains the programming concepts used, and LESSON OUTLINE is a simple step-by-step walkthrough for teaching this lesson. We recommend the teacher tries out the program and map before class to clarify the coding concepts and game play.

Questions or comments about this lesson? We'd love to hear from you! Email us at ozoedu@ozobot.com

SETUP

'COLOR SEARCH' OZOBLOCKLY PROGRAM ozo.bot/color-search

REQUIRED MATERIALS

- **FOR DEMO:** 1 Bit or Evo (pre-programmed, link above) & 1 printed map.
- 1 Bit or Evo per group,
- 1 printed game map per group (attached),
- 1 computer or tablet with wifi per group (Time-saving tip: set all devices to the program link, above, and calibrate all bots to the screen before class),
- 1 printed student worksheet per student (or write Q&A on the board) (attached).

STUDENT GROUPING

- Groups of 2-3 students are recommended, but 1:1 student to robot ratio is fine.

MAP SETUP

- Use one printed copy of the attached map. If you can't print in color, use markers to recreate the simple map, maintaining line thicknesses of the original.
- Remember to calibrate your bot to the black circle on paper before starting; this makes your bot able to see the colors properly. You don't need to calibrate again unless your bot isn't seeing the colors properly.

GAME PLAY STEPS

- **OBJECTIVE:** Find the best angle to launch your bot from to land on red the quickest.
- Start the program on your bot and set it on the start circle on the map, aimed at 10 o' clock, or top left.
- Watch your bot move forward on its own, bounce off of blue to go left, green to go right, and red to win.
- The program ends when the bot sees red. If the game runs too long for you, reset your bot on the start space and choose a better angle to start with.

DEMONSTRATION SETUP

- Set up your projector, or a main table, to show your Ozobot running the game program on the map so all students can see the bot's movements and reactions.

TIME MANAGEMENT

- This lesson is planned for 50 minutes. See the LESSON OUTLINE headers for a breakdown.
- If your classes are short (30 mins), or very busy, split the lesson in half; do the demo and code explanation first, and the code editing and playing second (with a little refresher on the program).

ABOUT THE CODE

'COLOR SEARCH' OZOBLOCKLY PROGRAM ozo.bot/color-search

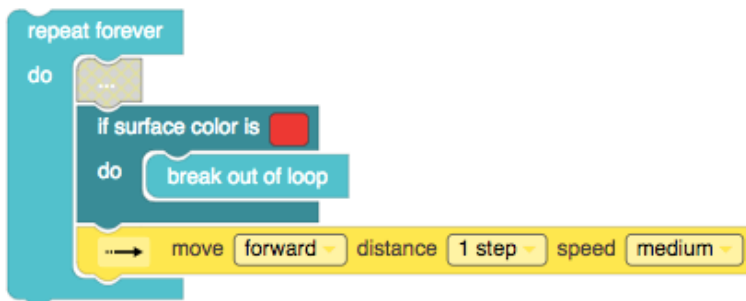
The coding concepts explained below make it possible for Ozobot to move forward, see the colors of the paper under it and react to them, repeat, and “win” when your bot sees red by cancelling the repeating code and ending the program. Read on to learn more.

LOOPS

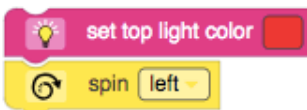
Loops allow a section of code to be repeated a certain number of times, or forever until a ‘break’ code is called. In this program, we use “**break**”. This code helps create autonomous movement and the ability to win.

Inside of this loop is code for checking the color of the paper or surface followed by a single short movement (see CONDITIONAL LOGIC, below, to learn how Ozobot checks color).

A single run of the loop takes less than a second to run. The surface color checks are extremely quick, then the bot moves forward a single step (about 1 cm) at medium speed. (If you are using Bit, the movements look continuous. On Evo, you can see the individual steps.) By repeating these codes we can create an autonomous movement where the bot moves on its own and reacts to different colors in different ways to stay inside of its map.



As you can see above, the **break** code is run when Ozobot lands on red. This means that the repeating code will end, and any code underneath **repeat forever** will run. Under the loop (**repeat forever**) is the victory dance (see below).



This dance can be anything - lights, movement, and, for Evo, sounds. Place LED colors before other blocks; if it's placed at the very end they won't be seen because the Ozobot will turn off. Use only 1 Light Animation (police, fireworks) since they add time to flash loading.

Be sure to set LED colors before any movements or timing. If you set an LED color at the very end of the program then you will not see that color appear.

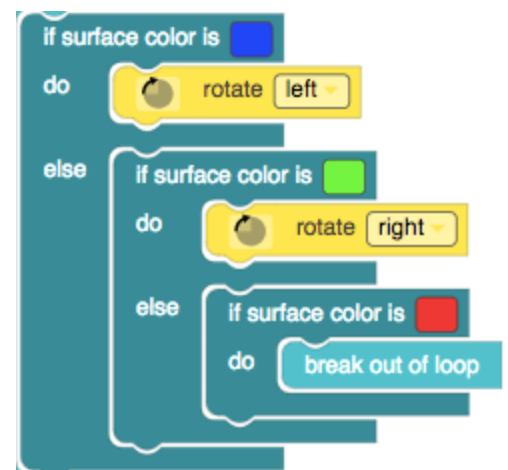
CONDITIONAL LOGIC

In programming, sometimes we want some code to run in a specific situation or condition. Programming languages make this easy by giving us “**if/else**” statements. These help create autonomous movement by reacting to color.

In OzoBlockly, we can check for the color of the paper underneath Ozobot by using this block: “**if the surface color is: (color)**”. Then, we can put in any action we want to have happen as soon as Ozobot see that color with its color sensor. If that color is not seen, then the actions in ‘do’ will not run.



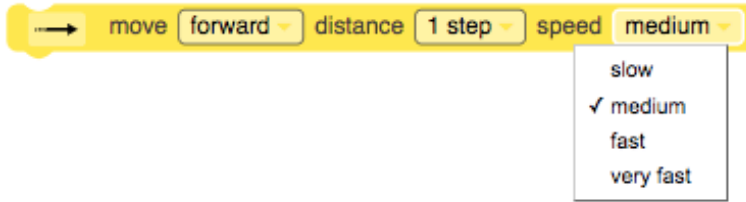
We can also create a specific action to run instead of doing nothing by using “else” with “if”. For this game, we want Ozobot to check for multiple colors, so we can put an “if” statement in an “else”. This “nesting” of blocks keeps code organized, and can save time while the code is running because the entire “if” block ends once any “do” section is run. That is, if blue is seen, then Ozobot doesn't check for green and red, it will continue with the loop instead.



Important: The paper's color can only be checked when that “if” code is run. This means that if your bot is performing the “move forward 1 step” code it will not be checking the paper's color. We need to wait until the movement is finished and running the “if” code to check the color of the paper or surface. This is why we only do one short movement between checking surface color.

MOVEMENT

The movement code blocks operate Ozobot’s two motors. Change the properties of the movement blocks by clicking the menu(s) on the block and select an option. 1 step is approximately 1 centimeter. Movement, combined with conditional logic and loops, create autonomous movement.



Reminder: OzoBlockly is sequential. One code block can run at a time. If you change this block to be ‘move forward 10 steps’, no surface color check is possible until those 10 steps are over! Use ‘1 step’ for this game program to make surface color checks more frequent. By making these checks frequent we create a kind of autonomous behavior.

If you would like your students to practice their angles in degrees for the turns you can have them go to Mode 4 Movement to get ‘rotate by xx degrees’ to replace the ‘rotate’ inside of green and blue surface color checks. Use the negative symbol in front of the number for degrees to make your robot rotate clockwise (right).



LESSON OUTLINE

DEMONSTRATE THE GAME - 10 minutes

Have one map and one programmed Ozobot ready to demonstrate to the class (see SETUP, above).

1. **Organize** students into pairs or groups.
2. **Explain** today's lesson to your students: *We will play a programmed game for Ozobot, then discover how it was coded. Then, we will edit the program to explore it. We'll learn loops, breaks, and the "if/else" codes that make Ozobot move on its own and "win".*
3. **Hand out** one student worksheet (attached) to each student. They'll write the answers down during your discussion of the demo. (Or you can do the Q&A on a board or screen.)
4. **Demonstrate** the game program as many times as you like. Explain how the game is played: *Ozobot will walk around by itself inside the map, 'bouncing' off of blue and green until it lands on red. Set Ozobot on the start symbol and aim it to get it to the red circle the fastest.*
5. **Students complete** sections A and B of the worksheet about Ozobot's movements and reactions on the map. We'll compare the answers to how the program is written, next.

DISCUSS HOW THE PROGRAM WORKS - 15 minutes

1. **Open** the OzoBlockly program (ozo.bot/color-search) on your main screen.
2. **Point out** the codes that make Ozobot see color: "if/else" statements (conditional logic);
 - a. "if" in coding is used like a sentence: "if Ozobot walks on blue, turn right" is an understandable English sentence! These codes control what Ozobot does on a paper's color.
 - b. "else" is only used when you want something to happen if Ozobot does NOT see that color. It's optional. Using only "if" means nothing will happen if that color is not seen.
 - c. the reaction to blue, the color of the border, helps Ozobot stay inside the map.
 - d. the reaction to green changes Ozobot's direction.
 - e. the reaction to red ends ("breaks") the loop, which ends the movement and therefore the game.
3. **Point out** the codes that make Ozobot walk around by itself: the loop and 'move forward';
 - a. The loop 'repeat forever' repeats the code inside of it over and over.
 - b. The forward movement is short so color checks happen often. By happening often, we can create 'autonomous' movement, or 'able to move by itself'.

- c. A single run of the loop takes less than 1 second.
4. **Point out** the codes after the loop: the victory dance;
 - a. A victory dance can be lights, movements, and sounds for Evo. Put lights before movement.
 - b. Light animations (police, rainbow, etc.) take longer to load, so use only one in a program.
5. **Allow** for any questions from students.
6. **Summarize** the program to remind students that they can now code autonomous movement, surface color reactions and winning!

STUDENTS EDIT THE PROGRAM – 25 minutes

1. Before having students edit the program, **seed** some programming ideas to make some meaningful edits, like make Ozobot go faster, make the turns wider, set LED colors in if statements.
2. **Hand out** one computer/tablet, Ozobot and map to each group of students.
3. **Students navigate** to the program link (or set each device to that page prior to class).
4. **Students load** the original program to their Ozobot without changing the code, and play.
5. **Students plan edits** on paper. If writing will take too long, students can plan edits verbally.
6. If there is time, **students can create** their own unique maps, which will allow them to explore how to program a game with the Winning mechanic.
7. **Ask** students to explain what edits they made to their program, and why.

COLOR SEARCH

STUDENT WORKSHEET

A. What does Ozobot do?

On blue, Ozobot _____

On green, Ozobot _____

On white, Ozobot _____

On red, Ozobot _____

B. Pseudocode

This 'pseudocode' is a simple English version of the program for the Color Search game. Read through, then fill in the blanks with the words you think belong there.

Set LED to white.

Wait 1 second.

Repeat forever:

If Ozobot is on blue,

If Ozobot is on green,

If Ozobot is on red,

Move forward 1 step.

Set LED to pink.

Spin left.

↘ Calibrate to paper ↙

