

Color Vision Lesson Overview

Title	Color Vision
Grade Level Target / Range	11-12
Subject Area(s)	biology, anatomy, engineering, computer science
Time Required	Part 1: 1.5 - 2 hours Part 2: 3+ hours (students building the robot and/or doing their own programming will add significant time)
Group Size	3 students per group
Materials Needed *Note that Parts 1 and 2 can be taught independently from one another if necessary.	<p>Part 1: Human Color Discrimination*</p> <ul style="list-style-type: none"> ● Display screen for opening video ● Pre-prepared goggles with red, green, or blue color filters fixed in front of the lenses (one goggle for each student) <ul style="list-style-type: none"> ● Rosco E-Colour+ #736 Twickenham Green ● Rosco E-Colour+ #106 Primary Red ● Rosco E-Colour+ #071 Tokyo Blue ● Training Software (online) ● Color Game (online) ● Color Discrimination Chart (one per group) <p>Part 2: Robot Color Discrimination*</p> <p>(one set of each per group)</p> <ul style="list-style-type: none"> ● Pre-installed mBlock software or access to web version (it is recommended to use the Chrome browser for the web version of mBlock). ● Downloaded programs: color_vision_communication and color_vision_movement ● Makeblock mBot robot ● Me Color Sensor V1 ● four AA batteries ● colored paper ● tape ● Student worksheet - evaluating sensor behavior
Assessments	Science Content Awareness Questions (Teacher Version - make a copy)

	Science Content Awareness Questions (Student View) Optional Project (Student Instructions) Optional Project (Rubric)
Expendable Cost Per Group	<i>Prices are Estimates as of January 2022</i> Goggles with replaceable facets - \$15 each Filters cut to place into goggles (Red , Green , Blue) - \$30 Makeblock mBot Robot - \$80 Me Color Sensor V1 - \$8.50 Four AAA Batteries - \$2.00
Key Words	sensory biology, engineering, color vision, color discrimination, robotics, block programming
<u>National Educational Standards</u>	Next Generation Science Standards (NGSS) International Society for Technology in Education (ISTE)
<u>State Specific Educational Standards</u>	Indiana Science Standards Ohio Science Standards West Virginia Science Standards

Introduction

Sensory biology as a field of study is concerned with how organisms obtain information about their environment. Specifically, how sense organs transduce energy from external stimuli to neural electrical signals. This unit examines the sense of color vision. Many animals have evolved color vision, which is the ability to discriminate between things based solely on the wavelengths of light they either emit or reflect. The mechanisms underlying color vision are well understood. Studies across organisms have led to a rich understanding of how animals vary in their color perception ability and its importance in their daily behavior (Cronin et al., 2014). There are many examples of color being used to evoke desired behaviors, such as human-made media advertisements that use color to get consumer attention, and the development of artificial lures colored to attract different species of fish. The biological principles behind the detection of color have also been applied to engineering problems. For example, color sensors are used to check the colors of products in factories, allowing products to be automatically sorted and assessed for their type or quality.

The purpose of this challenge-based unit is to introduce students to the biological principles underlying color vision in animals and use engineering design to reinforce the principles learned. The unit also demonstrates to students how the working principles of color vision can be applied to autonomous robots. Groups of students participate in two activities that build foundational knowledge of how humans perceive color and how robots discriminate color to meet their engineering challenges. In the first activity, students engage as a group with a computer game. Each group member wears a different set of goggles that transmit distinct wavelengths of light. The group can only determine the color of objects by *collaboratively*

functioning as a human eye. Students learn to program a robot to navigate a colored path using light sensors in the second activity. The light sensor works analogously to the human eye, consisting of different light detectors registering the amounts of different wavelengths of light. By comparing the amount of light across these detectors, the sensor determines the color of the light received.

This unit was created collaboratively with faculty from the University of Cincinnati College of Arts and Sciences, College of Engineering, and School of Education. Combining biology with engineering activities provides students with a unique opportunity to understand the parallels between animal and robot behavior and sensory/sensor function and addresses broad Next Generation Science Standards (NGSS Lead States, 2013) and International Society for Technology in Education Standards (International Society for Technology in Education, 2022).

Investigating / Essential Questions

- How does the human eye identify color?
- How can a robot be made to detect color?

Learning Objectives

1. Explain how the human eye discriminates wavelengths of light.
2. Explain how a computer screen can fool you into seeing yellow when all it produces are red, green, and blue wavelengths of light.
3. Develop a computer algorithm based on an analysis of sensor comparisons to guide robot behavior and explain its parallels to human color discrimination.
4. Explain the advantage of using multiple sensors in biological and robotic color discrimination
5. Develop and test a robot that navigates using sensor comparisons.

Prerequisite Student Knowledge

Basic biology course, some understanding of algorithmic thinking.

Instructional Summary

Part 1 Human Color Discrimination: Students work in groups of three, each wearing goggles with different color filters (red, blue, or green) and serve as proxies of the three different cone cells present in the human eye. Each filter blocks out all (or, most) but one select range of wavelengths. The students' task is to decide collaboratively what color their student group is viewing.

Part 2 Robot Color Discrimination: Student groups are tasked with programming a robot to follow a bicolored path constructed of paper using a color sensor to keep the robot in the middle of the path. The Me color sensor consists of three different light detectors registering the amount of red, green, and blue light.

Instructional Plan

Human Color Discrimination - Part 1

1.1 Introduction and Motivation

Play [this video](#) for students. Ask them just to watch the first time. Play it a second time and ask them to write down what they notice and what they wonder as they watch the video. Ask for student noticings first and write them down for all to see. Then ask for wonders. Write them down as well. Use the student notices and wonders as a lead into the essential question for the lesson: *how does the human eye identify color?*

1.2 Procedure

1.2.1 Color Training Program- 45 min

Link: <https://colorvisiontraining.anvil.app>

Students must try to understand how the human eye determines colors using only the three cone cells present in the human eye. Each student will put on a different colored goggle to represent one of the cone cells.



Students first use the ColorTraining program to familiarize themselves with the process of assigning color designations. The program shows four rectangles on the screen: one large colored rectangle at the top and three smaller rectangles below (Figure 1). The three squares are colored or black. Using a dropdown box, the students can select a different color for the large rectangle.

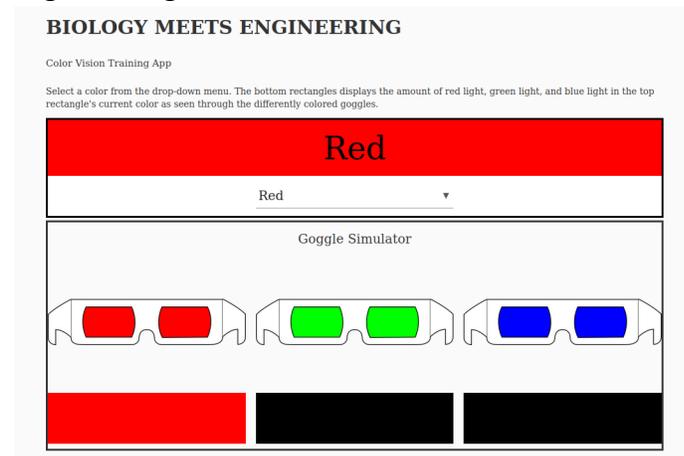


Figure 1. Representative image of the color training activity. The upper rectangle is the color to be identified and how this color is associated to the pattern of perceived brightness across three different colored goggles is shown below.

The bottom rectangle displays the amount of red light, green light, and blue light in the top rectangle's current color (and as seen through the red, green, and blue goggles). For example, if the students select red for the rectangle color, the left square would be bright red. The two other rectangles would be black. This is because the red color shown does not contain green and blue light. If they choose yellow for the rectangle, the rightmost square would be dark. The left and middle rectangles would be bright red and green, respectively. Another way of thinking about the function of the rectangle at the bottom is to consider them as showing the level of stimulation provided by the top color to the red, green, and blue color cones in the retina. We perceive a yellow color when light stimulates the red and green cones, but not the blue cones. Therefore, when selecting yellow, the rightmost (blue) rectangle is darkened.

Students can familiarize themselves with how various colors (top rectangle) look through their colored goggles using this training program. The rectangles at the bottom can be used in two ways. First, removing their goggles momentarily, the rectangles show how the top color looks through other students' goggles. For example, yellow (at the top) would look bright red and green to students wearing red and green goggles, respectively, but to a student wearing blue goggles, it would look black (or very dark blue). Therefore, students can use the rectangles at the bottom to explore how the color at the top can be encoded by the perceived brightness across three different cone cells (that is, colored goggles). A second way to use the rectangles is for students to keep their goggles on and explore how the three squares' brightness changes as a function of color. ***They should notice that a single color "channel" is not sufficient to distinguish colors.*** For example, the leftmost rectangle is equally bright for a yellow or red rectangle to a student wearing red goggles. This means that this student (or analogous cone) cannot distinguish between these colors alone. They need multiple color channels for this, especially for secondary colors (cyan, yellow, and magenta).

After students have been allowed to explore how different colors look through their goggles, they can be asked to identify the color in the large rectangle without looking at the label shown. The pull-down menu at the top of the screen (or the up/down arrows on the keyboard) changes the color of the rectangle. After the students are proficient at identifying primary colors (red, green, blue), they examine secondary colors (cyan, yellow, and magenta). These are created simply by mixing two of the primary colors. Therefore, they will be seen by two out of the three students as bright, and by the third as dark.

Table 1 is a key that relates the pattern of what the three students see – bright (+) or dark (-) – to the actual color of the rectangle on the screen. It may be helpful for students to have a [blank copy](#) to fill in as they work through this training program.

Table 1. Key for the color vision discrimination activities.

		Actual color of shape on screen							
		Red	Green	Blue	Yellow	Magenta	Cyan	White	Black
Student 'photoreceptors'	Red	+	-	-	+	+	-	+	-
	Green	-	+	-	+	-	+	+	-
	Blue	-	-	+	-	+	+	+	-
		+ = shape appears bright - = shape appears dark							

1.2.2 Color Game Software

Link: <https://colorvisiongame.anvil.app>

After student groups practice identifying colors using the training program, they are asked to execute the ColorGame software to reinforce learned concepts and collaborative interaction. While wearing their goggles, groups open the software. They then see nine boxes of different primary or secondary colors. Below those boxes is a word indicating the color of the box to be identified. All boxes of that color must be selected, with the group's scores determined by the time needed to identify all appropriately colored boxes correctly. Thirty seconds are allotted per screen, with twelve total screens. The score for each screen is a measure of the percentage remaining on the clock after all the correct boxes are selected. The total score for the game is the total across all screens. An incorrect answer (selecting the incorrectly colored box) incurs a penalty.

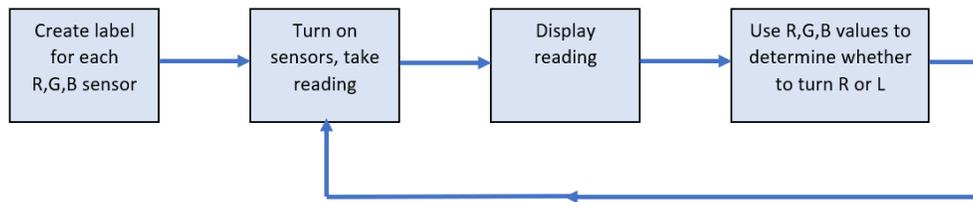
Part 2: Robot Color Discrimination

2.1 Introduction and Motivation

Explain to students that their task is to program a robot to follow a bicolored path constructed of paper while using a color sensor to keep the robot in the middle of the path. Students may need some [background information](#) on the color sensor. The Me color sensor consists of three different light detectors registering the amount of red, green, and blue light.

Give students some time to assemble the mBot robot according to the [manufacturer's instructions](#) and install the Me Color Sensor V1 in place of the Line follower sensor depicted in the generic instructions. Students should not over-tighten the screws when installing the color sensor because the caster wheel rests on some of the electronic components on the sensor board. This may cause the wheel not to move. Students also construct a bicolored path using colored paper and tape such that each side of the path is a different color.

Ask students to come up with a general outline for what a program would need to communicate to the robot in order for it to complete the task. Students should have main components such as:



Once students have a plan, proceed with the pre-written code unless you have time for students to explore writing their own code. Instructions to allow students to write the code from scratch can be found in [this document](#).

2.2 Procedure

Currently, the color sensor can only be used in the so-called Upload mode. This means that programs need to be uploaded to the robot using a USB cable. We can then disconnect the cable, and the robot is entirely autonomous. From then on, every time we switch on the robot, it will run the uploaded program without communication with your computer. The fact that the sensor can only be used in Upload mode makes programming the robot harder. The main issue is that one cannot observe sensor values while our program is running and the robot is not connected using a USB cable. This makes it challenging to pick the correct sensor values to which to respond. To solve this issue, we will proceed in two steps. First, students will run a program that can show the current values returned by the sensor while the robot is connected using a USB cable. This allows them to pick values for the program. Second, they can use these values to construct a program that follows the track, at which point the robot is no longer connected to the computer using a USB cable.

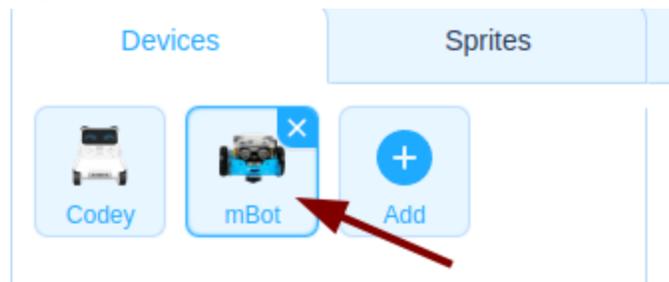
2.2.1 Observing the Sensor Values

To successfully program the robot to follow the path, students need to know the sensor values for the colors of paper they use. This is necessary because when the robot is placed on a bicolored path it will need to be programmed to turn right or left depending on whether the sensor detects too much of one color versus the other. This programming results in the robot finding and maintaining movement along the path midline.

To identify the sensor values for the different sheets of colored paper, students first run the colorvision communication program. This program allows the students to obtain readings from the Me Color sensor. The sensor returns three values from 0 to 225 that correspond to the amount of red, green, and blue light detected. The program continuously reads out the sensor values and sends them to the computer. This allows the students to experiment with the sensor and get to know the values it returns when the robot is placed on different colors of paper. Specifically, students obtain sensor values as follows:

1. Open the program in mBlock v5.
2. Connect the robot to the computer using the USB cable.
3. In the bottom left corner of the mBlock v5 window, click on the Devices tab.

4. Click Add and select mBot
5. While in Upload mode, click Connect
6. Once connected, update the firmware of the robot by selecting Update under Settings.
7. Once complete, use the slide button to select the Upload mode.
8. Next, click Connect to connect to the robot through USB. The color sensor can only be used in Upload mode. After uploading the program, students can disconnect the cable, and the robot is entirely autonomous.
9. Drag the color_vision_communication.mblock program (Figure 2, left panel) into the main Block programming tab. Or use File > Open from your computer. Click Upload. This will upload the program to the robot. If the program does not seem to load, select the mbot from the devices tab.



10. Once the uploading is finished, click on the Sprites tab to bring up the program running on the side of the computer. Several blocks should be flashing. Each time a block flashes, a message from the robot is received.
11. The panda stage window will show the values of the red, green, and blue channels of the color sensor (36, 5, and 6 in Figure 2, right panel). As long as the program is running, the values red (RD variable), green (GR variable), and blue (BL variable) values will be updated.

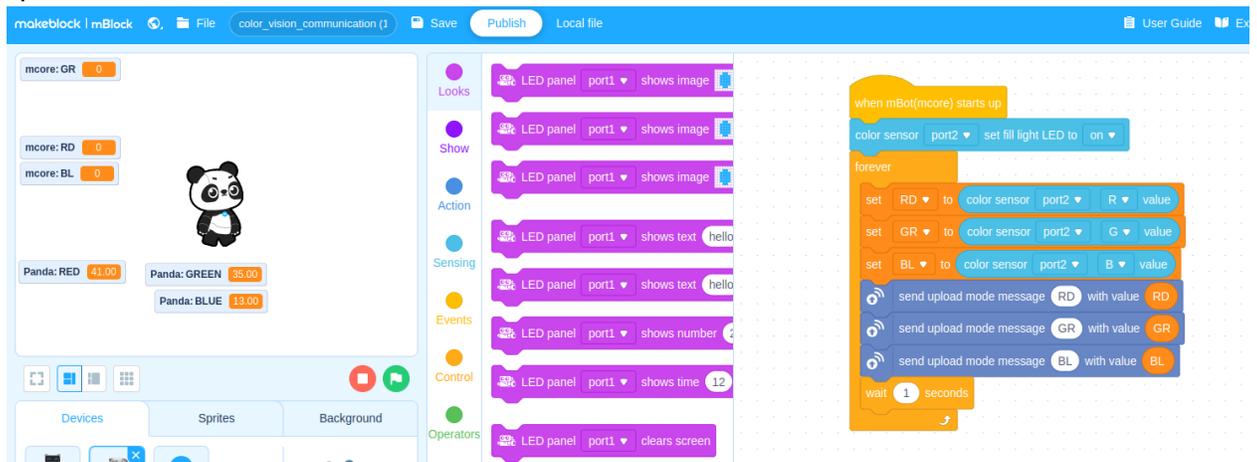


Figure 2. Visualization of Me Color sensor values. The mBlock code for the program color_vision_communication.mblock is shown (left panel). The Panda stage window showing the output values of red, green, and blue detectors within the color sensor are illustrated (right panel).

Give students some quiet think time followed by time to discuss as a group what each part of the program does. After hearing from a few groups, clarify any misconceptions. The program

works as follows. The sensor features white LEDs, which can be switched on and off. When the mBot starts up the program turns on the white LEDs, and a uniform light hits the floor (under the robot). Lighting the floor with a fixed and broad-spectrum light ensures the colors are easier to discriminate. Inside the forever loop, the values of the color sensor are assigned to three variables RD (red), GR (green), and BL (blue). The first three blocks are used to send data to the computer if the robot is attached using a USB cable. For example, 'set RD to color sensor port 2 R-value' tells the robot to read the amount of red light detected by the sensor plugged into port 2 and store its value in a variable called RD. The subsequent 'Send upload mode message RD with value RD' sends the RD values to the computer and attaches a RD label to the values stored on the computer. The robot waits one second before taking its next sensor reading.

The sensor values recorded from the colored paper are used to maintain the robot on the midline. Therefore, students next evaluate the sensor values when the robot is left, right, or on the midline. To do this, students place the center of the Me Color sensor at different positions across the midline and at each position (see Figure 3) and [record the values of the sensor](#) (see example in Table 2).

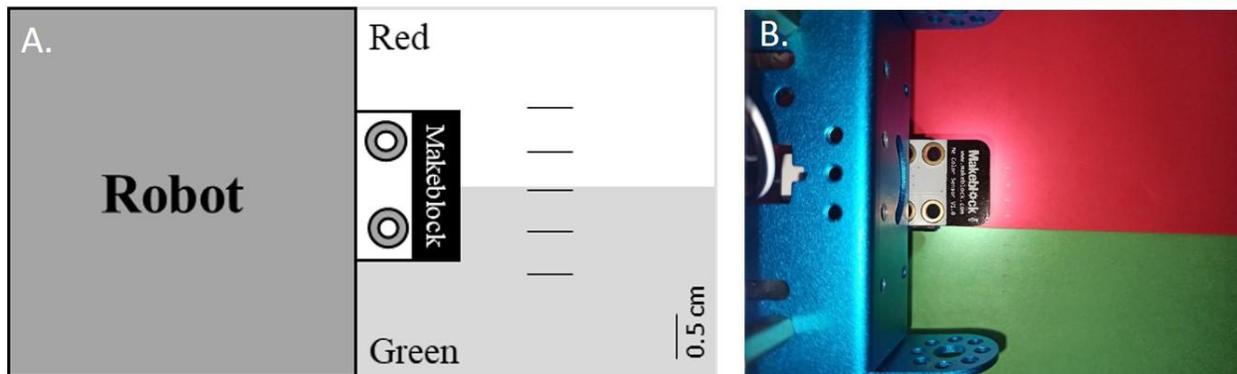


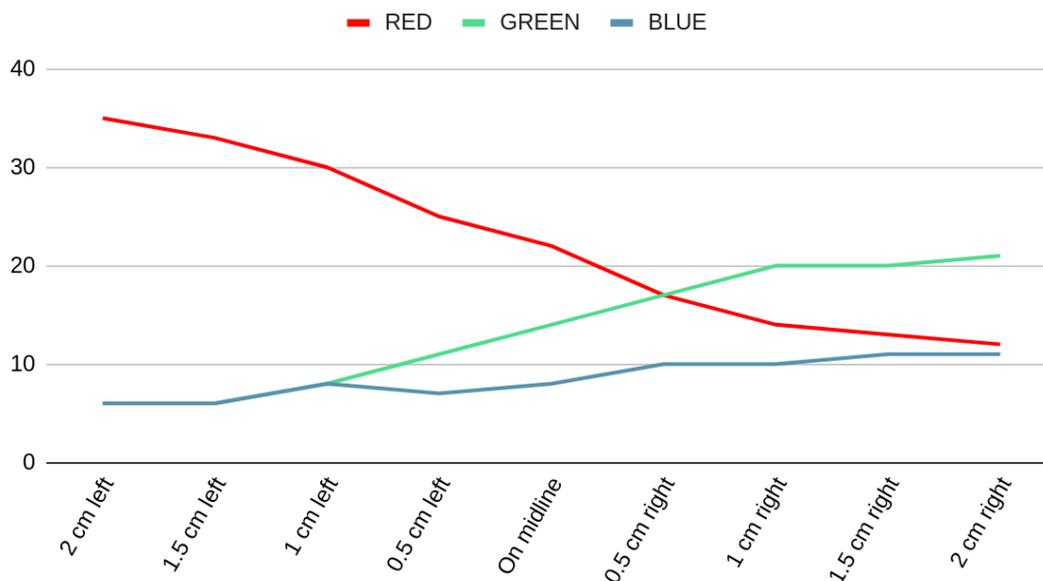
Figure 3. mBot robot. (A) Schematic of mBot (B) Actual mBot example. Top view of the front of the robot showing the sensor face down. The bicolored track has red paper on the robot's left and green on its right. Measurements were recorded while moving the robot from left to right in steps of 0.5cm, starting with the middle of the sensor directly on the midline.

Table 2. Example values recorded from the color sensor positioned at different locations on the bicolored path

Position	RED	GREEN	BLUE
2 cm left	35	6	6
1.5 cm left	33	6	6
1 cm left	30	8	8
0.5 cm left	25	11	7
On midline	22	14	8
0.5 cm right	17	17	10

1 cm right	14	20	10
1.5 cm right	13	20	11
2 cm right	12	21	11

Activation of color channels



There are a few things to observe. The students' values might be different depending on the color of the paper used. The maximum values (35 and 21) correspond to the color of the red and green papers, respectively. However, notice that for each paper, all channels have a value larger than 0, indicating that each paper also reflects some amount of other wavelengths of light. This recorded data can be used formatively by the teacher to generate guiding questions to foster student understanding. See the sample questions below.

- What color do the maximum values of 35 and 21 correspond to?
- Why do all of the channels have a value greater than 0?
- How would your values change if you switched the colors on the left and right of the line?

2.2.2 Running the robot

Now that students have some baseline data, they will need to develop their computing algorithm and program the robot to follow the midline. Give students time to write a rule that would direct the robot when to turn right or left based on their data. Check student work before moving on. Students should have something similar to the example explanation below.

In the above example, the set-up consisted of red paper at the left and green at the right. From Table 1, it could be decided that the robot can stay in the center of the path using the following rules, based on the difference between the red and green channels.

If RED - GREEN > 14: turn right

If RED - GREEN < 0: turn left.

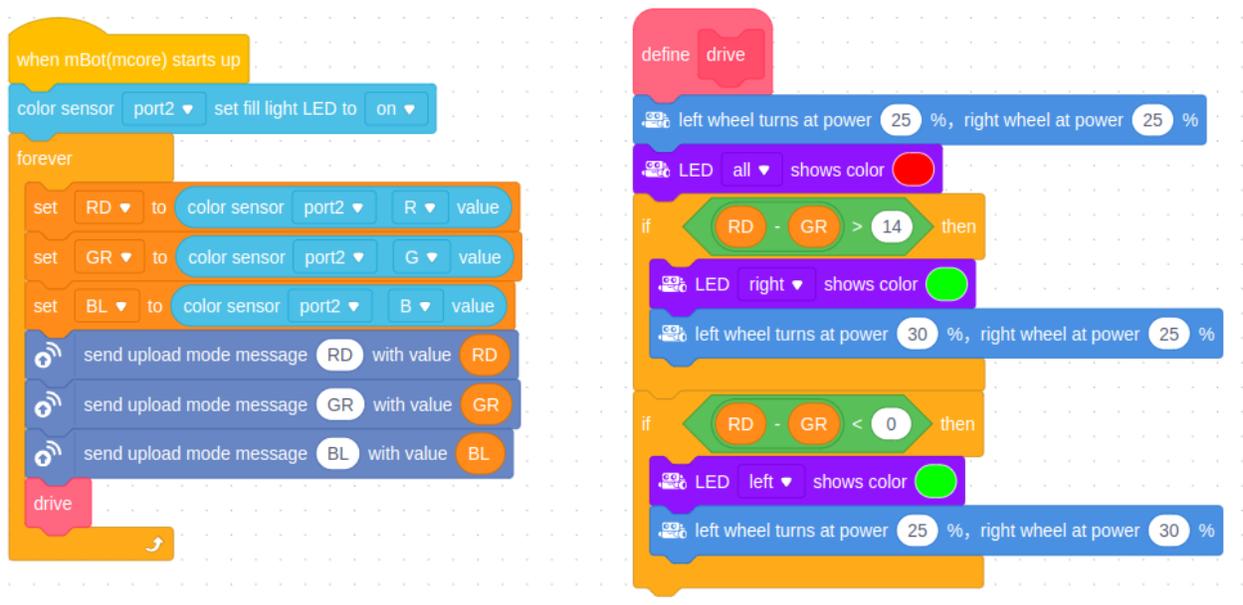
In this example, the value of 14 reflects the difference between the red and green values when the sensor is 0.5 cm left of the midline (25 - 1, in table 2). The RED or GREEN values could also be used in isolation to decide when to turn, for example,

- If RED > 25: turn right
- If RED < 17: turn left

However, by using multiple channels in deciding when to turn, the robot should be less sensitive to noise.

Next, students program the robot to follow the midline of the bicolored path using the following steps.

1. Follow steps 1-8 above, excluding steps 6 and 7 (found under 2.2.1).
2. Drag the color_vision_movement program into the main programming tab to upload it onto the robot (Figure 4). This program implements the rules derived above. Students edit the program to make sure the robot follows the rules they decided upon while collecting data.



Again, give students some quiet think time followed by time to discuss as a group what each part of the program does and how they might modify it to work with the rules they came up with. After hearing from a few groups, clarify any misconceptions. The program works as follows. The left side of the program handles reading the color sensor, as explained in Figure 4. The 'drive' block is where the action is, and its contents are expanded on the right side of the program. This block converts the sensor values (RD, GR, BL) into motor commands (motor

speeds). The block starts by setting both motors to the same (low) speed (in this example a power (speed) setting of 25% for each wheel). This makes the robot drive forward. Next, the program checks whether the difference between RD and GR is larger than a defined value (e.g., 14) and alters the speed of the motors to make the robot turn. So in this example, the robot should turn right. Therefore, if $RD - GR > 14$, the left wheel is sped up slightly (from 25% to 30%). This makes the robot turn right. Next, the robot checks whether $RD - GR < 0$, and if so, the robot should turn left by speeding up its right wheel.

Extensions

Several extensions to the lesson plan can be made. Students can experiment with

- (a) changing the wheel speed to mimic a fast-running bug,
- (b) altering the colors in the path to include cyan, yellow and/or magenta mimicking a more natural color environment, or
- (c) changing the nature of the ambient light which requires a more nuanced (and biologically realistic) analysis of the sensor output.

In the example program, all speeds are quite low (25% or 30%). Slower robots often perform better. However, students can experiment with the wheel speeds. Also, the difference between the wheels when turning is quite small (5%). Increasing this difference will increase the rate of turning. This might enable the robot to follow a winding path. However, it will also make the robot's path less stable. Altering the program to implement the students' rules requires changing the 'if' conditions (i.e., changing the hexagonal green blocks to reflect the rules).

Note: if the robot appears not to move, give it a slight push. Older batteries may cause the 25% speed to be too slow to allow the robot to start movement.

Second, the end and the start of the track can be constructed of a third color. When the robot detects this color, it can be programmed to turn 180 degrees to return to the other end of the path.

Finally, the white LEDs on the Me Color sensor can be switched off. Remember, the LEDs deliver a uniform fixed and broad-spectrum light onto the floor, making the sensor's ability to color discriminate easier. But, consider a real-life scenario where a robot is trying to detect whether the floor underneath it is red or yellow, but the light in the environment is variable or suboptimal. If the lighting is poor, the lamps overhead might not emit sufficient yellow light, and as such, the light reflecting from the floor will contain little yellow light. The sensor will have difficulty discriminating yellow from red.

These are just a few examples of many increasingly complex challenges students can give the robot to do.

Assessments

Consider checking student understanding through a knowledge check and an individual project.

[Science Content Awareness Questions](#) (Teacher Version - make a copy)

[Science Content Awareness Questions](#) (Student View)

[Optional Project](#) (Student Instructions)

[Optional Project](#) (Rubric)

Supporting Activity Information / Background

Light is electromagnetic radiation with a wide range of frequencies/wavelengths, i.e., a wide spectrum. While a broad spectrum of wavelengths may be present in the environment, only a narrow range is visible to most animals. It is the particular mixture of wavelengths present within this range, and animals' ability to analyze them, that determines the perceived color. In humans, light enters the eye through the cornea, passes through the pupil, and finally is focused by the lens onto the retina. The retina contains rod and cone photoreceptor cells. The rod photoreceptor cells are important for peripheral vision and low light sensitivity, while three types of cone photoreceptor cells are used to perceive detail and color. The three types of cone cells are referred to as red, green, and blue cone cells. These color names refer to different wavelengths of light to which each of them is most sensitive. This difference in color sensitivity of the cone cells is due to their expression of different proteins, called opsins. These opsins form a complex with another molecule, called retinal, to form the pigment called rhodopsin that absorbs the light entering the eye. Differences in the amino acid sequence of the opsin proteins in the different cell types result in the cells' distinct sensitivities to wavelengths or colors of light; that is, it makes them more or less likely to absorb short (blue), medium (green) or long (red) wavelengths.

However, for an organism to have color vision, it is not sufficient to be sensitive to certain wavelengths of light. Rather, it means that the organism can tell the difference between the different wavelengths. To do this, the light absorbed by the rhodopsin in each cone cell is converted into electrical energy, and the amount of electrical excitation produced is compared by the eye and brain such that the color of the incoming light is determined. For example, consider a red object. That object looks red because it absorbs most wavelengths of visible light but reflects long wavelengths around 600-750 nanometers. This reflected light would stimulate the 'Red' cone cell (i.e., the longest wavelength sensitive cone in the human eye) because its rhodopsin absorbs those wavelengths, but not the 'Green' or 'Blue' cone cells because their rhodopsins do not. The brain interprets this pattern of activation of the three cone cells as red. Using the same logic, light reflecting off a yellow object will stimulate the 'Red' and 'Green' cones, but not the 'Blue' and forms a different pattern in the brain, resulting in the object being perceived as yellow. If all three cells are equally stimulated, the object is perceived as white (or if the light is low, gray).

There is a PhET simulation by the University of Colorado, Boulder that could be used to further student understanding. It can be found at this link:

https://phet.colorado.edu/sims/html/color-vision/latest/color-vision_en.html

National Educational Standards

NGSS	<p>Performance Expectations: <i>HS-LS1-2.</i> Develop and use a model to illustrate the hierarchical organization of interacting systems that provide specific functions within multicellular organisms. (Part 1) <i>HS-LS1-3.</i> Plan and conduct an investigation to provide evidence that feedback mechanisms maintain homeostasis. (Part 2 - <i>note that it is the robot that is maintaining the homeostasis</i>).</p> <p>Disciplinary Core Ideas: <i>LS1.A: Structure and Function</i></p> <ul style="list-style-type: none">● Multicellular organisms have a hierarchical structural organization, in which any one system is made up of numerous parts and is itself a component of the next level. (Part 1)● Feedback mechanisms maintain a living system’s internal conditions within certain limits and mediate behaviors, allowing it to remain alive and functional even as external conditions change within some range. Feedback mechanisms can encourage (through positive feedback) or discourage (negative feedback) what is going on inside the living system. (Part 2) <p>Crosscutting Concepts: <i>Systems and System Models</i> Models (e.g., physical, mathematical, computer models) can be used to simulate systems and interactions—including energy, matter, and information flows—within and between systems at different scales. (Part 1) <i>Stability and Change</i> Feedback (negative or positive) can stabilize or destabilize a system. (Part 2)</p> <p>Science and Engineering Practices:</p> <ul style="list-style-type: none">● Asking Questions and Defining Problems (Parts 1 & 2)● Developing and Using Models (Parts 1 & 2)● Planning and Carrying out Investigations (Part 2)● Analyzing and Interpreting Data (Parts 1 & 2)● Using Mathematics and Computational Thinking (Parts 1 & 2)● Constructing Explanations and Designing Solutions (Parts 2 & 2)● Engaging in Argument from Evidence (Parts 1 & 2)
-------------	---

	<ul style="list-style-type: none"> ● Obtaining, Evaluating, and Communicating Information (Parts 1 & 2)
<p>ISTE Standards</p>	<p>1.1 Empowered Learner</p> <ul style="list-style-type: none"> c. Students use technology to seek feedback that informs and improves their practice and to demonstrate their learning in a variety of ways. d. Students understand the fundamental concepts of technology operations, demonstrate the ability to choose, use and troubleshoot current technologies, and are able to transfer their knowledge to explore emerging technologies. <p>1.4 Innovative Designer</p> <ul style="list-style-type: none"> c. Students develop, test, and refine prototypes as part of a cyclical design process. <p>1.5 Computational Thinker</p> <ul style="list-style-type: none"> b. Students collect data or identify relevant data sets, use digital tools to analyze them, and represent data in various ways to facilitate problem-solving and decision-making. c. Students break problems into component parts, extract key information, and develop descriptive models to understand complex systems or facilitate problem-solving. d. Students understand how automation works and use algorithmic thinking to develop a sequence of steps to create and test automated solutions. <p>1.7 Global Collaborator</p> <ul style="list-style-type: none"> c. Students contribute constructively to project teams, assuming various roles and responsibilities to work effectively toward a common goal.

State Specific Educational Standards

Ohio Science Standards	<p>AP.LO.3 Homeostasis (of the robot) AP.IC.2 (Sense of Sight)</p> <p>Cognitive Demands for Science <i>Designing Technological/Engineering Solutions Using Science Concepts</i></p> <ul style="list-style-type: none">● Requires students to solve science-based engineering or technological problems through the application of scientific inquiry. Within given scientific constraints, propose or critique solutions, analyze and interpret technical and engineering problems, use science principles to anticipate effects of technological or engineering design, find solutions using science and engineering or technology, consider consequences and alternatives, and/or integrate and synthesize scientific information. <p><i>Demonstrating Science Knowledge</i></p> <ul style="list-style-type: none">● Requires student to use scientific practices and develop the ability to think and act in ways associated with inquiry, including asking questions, planning and conducting investigations, using appropriate tools and techniques to gather and organize data, thinking critically and logically about relationships between evidence and explanations, constructing and analyzing alternative explanations, and communicating scientific arguments. <p><i>Interpreting and Communicating Science Concepts</i></p> <ul style="list-style-type: none">● Requires students to use subject-specific conceptual knowledge to interpret and explain events, phenomena, concepts, and experiences using grade-appropriate scientific terminology, technological knowledge, and mathematical knowledge. Communicate with clarity, focus, and organization using rich, investigative scenarios, real-world data, and valid scientific information. <p><i>Recalling Accurate Science</i></p> <ul style="list-style-type: none">● Requires students to provide accurate statements about scientifically valid facts, concepts, and relationships. Recall only requires students to provide a rote response, declarative knowledge, or perform routine mathematical tasks. This cognitive demand refers to students' knowledge of scientific facts, information, concepts, tools, procedures (being able to describe how), and basic principles.
-------------------------------	---

<p>Indiana Science Standards</p>	<p>AP.1.3 Homeostasis (of the robot) AP.7.2 Function of the eye</p> <p>Science and Engineering Process Standards (SEPS) SEPS.1 Posing questions (for science) and defining problems (for engineering) SEPS.2 Developing and using models and tools SEPS.3 Constructing and performing investigations SEPS.4 Analyzing and interpreting data SEPS.5 Using mathematics and computational thinking SEPS.6 Constructing explanations (for science) and designing solutions (for engineering) SEPS.7 Engaging in argument from evidence SEPS.8 Obtaining, evaluating, and communicating information</p>
<p>West Virginia Science Standards</p>	<p>S.HS.HAP.13 Apply the structure of the ear and eye to their function/dysfunction in relation to environmental perception. S.HS.ETS.2 Design a solution to a complex real-world problem by breaking it down into smaller, more manageable problems that can be solved through engineering.</p>

References

Cronin, T. W., Johnsen, S., Marshall, N. J., & Warrant, E. J. (2014). *Visual ecology*. Princeton University Press.

Indiana Department of Education. (2017). *Indiana Academics for Science. Science: Anatomy & Physiology*.
<https://www.in.gov/doe/students/indiana-academic-standards/science-and-computer-science-2016-2010/>

International Society for Technology in Education. (2022). *ISTE Standards: Students*.
<https://www.iste.org/standards/iste-standards-for-students>

NGSS Lead States. (2013). *Next Generation Science Standards: For States, By States*. Washington, DC: The National Academies Press.

Ohio Department of Education. (2018-19). *Ohio's Learning Standards and Model Curriculum Science*.
<https://education.ohio.gov/Topics/Learning-in-Ohio/Science/Ohios-Learning-Standards-and-MC>

Simulation by PhET Interactive Simulations, University of Colorado Boulder, licensed under [CC-BY-4.0](https://creativecommons.org/licenses/by/4.0/) (<https://phet.colorado.edu>).

West Virginia Department of Education. (2015). *Policy 2520.3C: Next Generation Content Standards and Objectives for Science in West Virginia Schools*.
<https://wvde.us/middle-secondary-learning/science/standards-and-guidance/>

Authors (in alphabetical order)

Anna DeJarnette is an Associate Professor at the University of Cincinnati, School of Education, Cincinnati, OH 45221.

Anna E. Hutchinson is a postdoctoral fellow and program coordinator at the University of Cincinnati, Department of Biological Sciences, Cincinnati, OH, 45221.

John E. Layne is an Associate Professor at the University of Cincinnati, Department of Biological Sciences, Cincinnati, OH, 45221.

Erin Milligan is an Education Consultant at Hamilton County Educational Service Center, Instructional Services, Cincinnati, OH, 45231.

Stephanie M. Rollmann is a Professor at the University of Cincinnati, Department of Biological Sciences, Cincinnati, OH, 45221.

Dieter Vanderelst is an Assistant Professor at the University of Cincinnati, Departments of Biological Sciences, Electrical Engineering, Mechanical Engineering, and Psychology, Cincinnati, OH, 45221.

Contributor, Supporting Program, Acknowledgements, and Classroom Testing

We thank the teacher and student participants in the University of Cincinnati (UC) Biology meets Engineering summer program for their feedback. We thank Gloria Ononye and Valecia Kelly for discussions of instructional frameworks. This material is based upon work supported by the National Science Foundation ITEST grant number 1759150 to SMR, JEL, AD, and DV.