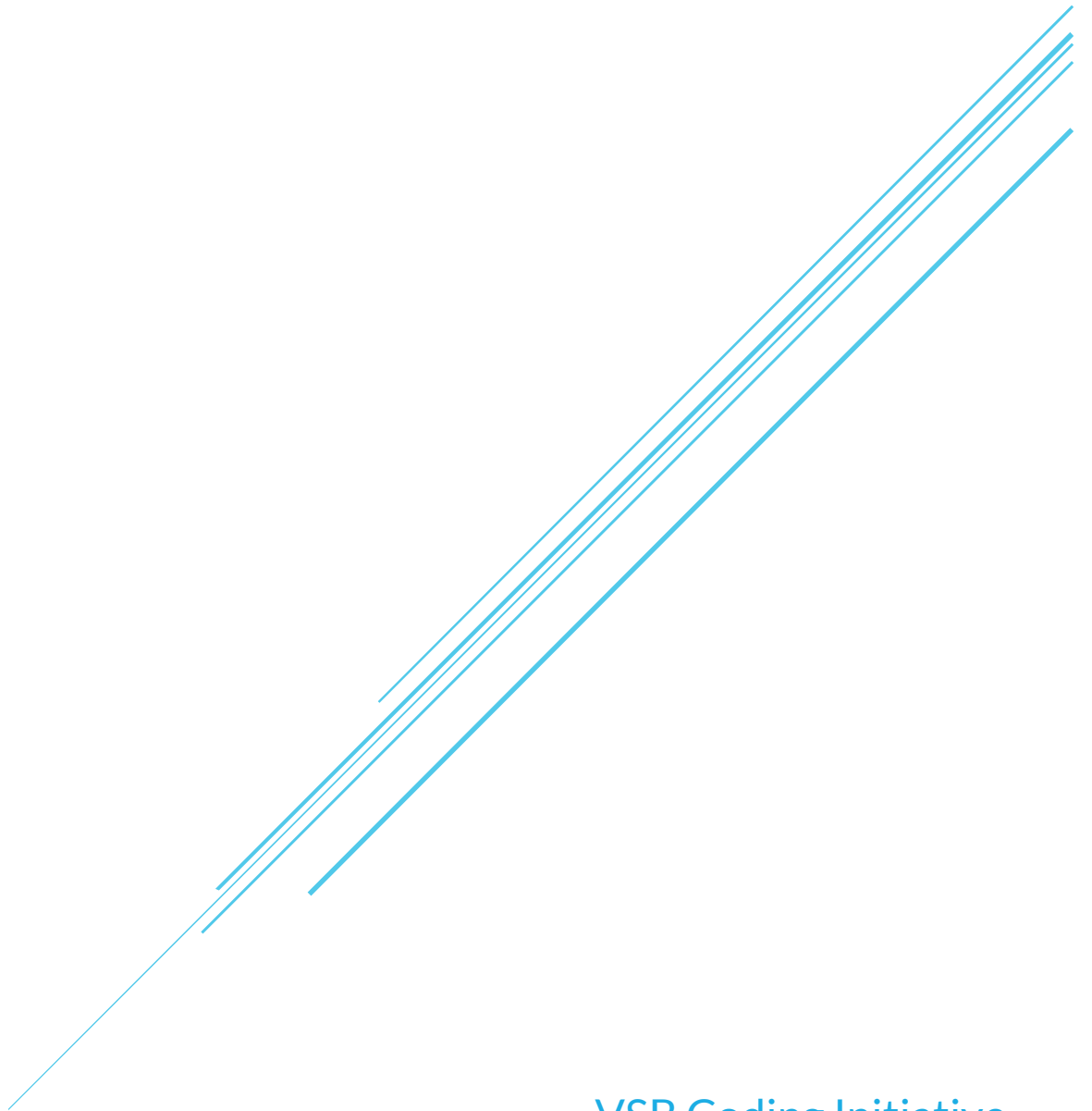


VSB TEACHER'S GUIDE

OZOBOT BIT

Basics



VSB Coding Initiative
Grade 5-7

CONTENTS

Guide Overview.....	2
BC Curriculum and Assessment	3
The Big Ideas	3
Connecting Curricular Content	4
Assessment.....	5
Get To Know Your Ozobot Bit	8
Hardware	8
Software and Coding.....	9
Basic Training - Colour Codes	10
Basic Training - OzoBlockly	28

GUIDE OVERVIEW

This Ozobot guide has been produced by the VSB District Coding Initiative Team to help you introduce and teach with the Ozobot Bits in your classroom!

The following sections will assist you in establishing a comprehensive Ozobot knowledge base in your classroom.

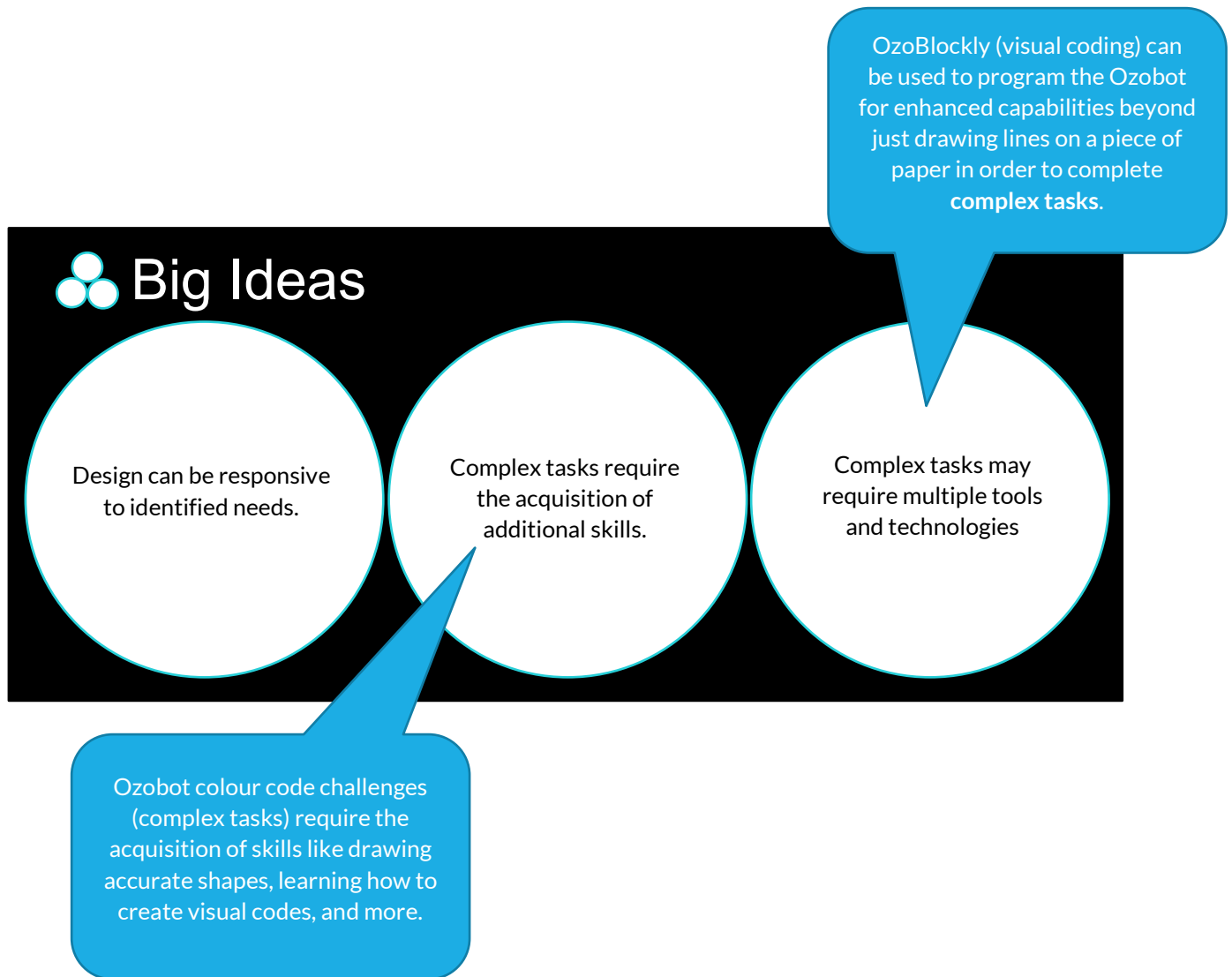
1. BC Curriculum and Assessment
This section introduces Ozobots connections to BC ADST curriculum and provides guidance for assessment.
2. Get to know your Ozobot Bit
This section will introduce you to the Ozobot Bit hardware and supporting software.
3. Basic Training – Colour Codes
This section contains the unit & lesson plans for teaching students how to use colour codes with their Ozobot Bits.
4. Basic Training – OzoBlockly
This section contains the unit & lesson plans to teach students how to use block coding with their Ozobot Bits.

The learning activities in this guide should on average take 5 classes to complete.

BC CURRICULUM AND ASSESSMENT

We chose to include Ozobots in the VSB Coding Carts as they are an engaging, easy to use, and a comprehensive platform that closely aligns to our BC ADST Curriculum. (<https://curriculum.gov.bc.ca/curriculum/adst/7>)

THE BIG IDEAS



CONNECTING CURRICULAR CONTENT

Below are some examples and instances of how Ozobots connect with the BC ADST curriculum;

BC ADST 7 Content	Ozobot Connections
Computational Thinking (Students are expected to know the following)	
Simple Algorithms that reflect computational thinking (sequencing of events) <i>for sorting, searching, sequence, selection, and repetition; specific statements to complete a simple task; cryptography and code breaking (e.g., cyphers)</i>	<ul style="list-style-type: none"> - Planning a pathway using lines and sequences of colour codes, for example, (sequence of colours) red + green + blue = "snail dose" speed are examples of simple algorithms.
Visual Representations of problems and data <i>graphs, charts, network diagrams, info graphics, flow charts, lists, tables, or arrays</i>	<ul style="list-style-type: none"> - Representing patterns or relationships creates visual representations of data. - Ozobot pathways and colour codes (patterns) are visual representations of data.
Visual Programming	<ul style="list-style-type: none"> - OzoBlockly is an example of visual coding; block coding.
Robotics	
Main components of robots: sensors, control systems, and effectors <i>sensors: "sense" – the parts of the robot that allow it to gather information about its environment that guides its behaviour</i> <i>control systems: "think" – the part of the robot that determines the robot's behavior</i> <i>effectors: "act" – the parts of the robot that do the work</i>	<ul style="list-style-type: none"> - The Bit has a colour sensor and line sensors. - The Ozobot has a microprocessor that reads sensors and controls effectors. - The wheels and LEDs on the Ozobot are effectors.
Various ways that objects can move <i>straight line, back-and-forth, round-and-round, zigzag, fast and slow, fixed distances in set patterns</i>	<ul style="list-style-type: none"> - The colour codes and lines in addition of OzoBlockly Code will let the Bit travel along a way

ASSESSMENT

When students (individually and in groups) demonstrate their ability to solve challenges and assignments with Ozobots then they are showing proficiency in their abilities and skills (such as, coding, collaboration, problem solving, etc.) in the Curricular Content of Computational Thinking and Robotics.

We recommend that teachers assess students' **proficiency** in the *Curricular Competencies* rather than the *Curricular Content*; the content is used for students to demonstrate their proficiency in the more important overarching *Curricular Competencies*, *Curriculum Big Ideas*, and *Core Competencies*.

We have inserted (page 6 and 7) example rubrics that focus on assessing students' proficiency in the ADST Curricular Competencies through the learning and completion of Ozobot activities outlined in this guide.

For more details and in-depth explanations of assessment please refer to our VSB Learning Services.

EXAMPLE OF POSSIBLE SELF - ASSESSMENT RUBRIC

STUDENT COMPETENCY SCALE	BEGINNING to acquire knowledge, skills, strategies and processes.	DEVELOPING the ability to apply knowledge, skills, strategies and processes.	APPLYING knowledge, skills, strategies and processes consistently.	EXTENDING knowledge, skills, strategies and processes creatively and strategically.
Empathize	With support from my teacher or my peers, when Pair Programming I was respectful, talked to my partner, explained what I was doing, thought ahead and made suggestions, and switched roles often.	With some occasional help from my teacher or my peers, when Pair Programming I was respectful, talked to my partner, explained what I was doing, thought ahead and made suggestions, and switched roles often.	When Pair Programming I was respectful, talked to my partner, explained what I was doing, thought ahead and made suggestions, and switched roles often.	When Pair Programming I was respectful, talked to my partner, explained what I was doing, thought ahead and made suggestions, and switched roles often. I also helped support other pairs of programmers that needed extra support.
Define	After completing the SPACESHIP DRIVER'S LICENSE KNOWLEDGE TEST, I can explain and show how the ozobot follows paths and reads colour codes.	After completing the SPACESHIP DRIVER'S LICENSE KNOWLEDGE TEST, with some occasional help from my teacher or my peers, I can explain and show how the ozobot follows paths and reads colour codes.	After completing the SPACESHIP DRIVER'S LICENSE KNOWLEDGE TEST, I can explain and show how the ozobot follows paths and reads colour codes.	After completing the SPACESHIP DRIVER'S LICENSE KNOWLEDGE TEST, I can explain and show how the ozobot follows paths and reads colour codes. I can also support others.
Ideating	With some support from my teacher or peers, I completed the challenges and assignments by using the assigned paths and colour codes. With some support from my teacher or peers, I generated potential ideas and added to other's ideas.	With some occasional support from my teacher or peers, I completed the challenges and assignments by using the assigned paths and colour codes. With some occasional support from my teacher or peers, I generated potential ideas and added to other's ideas.	I completed the challenges and assignments by using the assigned paths and colour codes. I generated potential ideas and added to other's ideas.	I completed the challenges and assignments by using the assigned paths and colour codes and helped others who needed assistance. I generated potential ideas and added to other's ideas. I also assisted others who needed support.
Prototyping	With some support from my teacher or my peers, I used the information given to me (colour code keys) to draw new programs for the ozobot. I tested my colour codes as I completed the lessons and tried out different ideas, especially if the first one didn't work.	With some occasional support from my teacher or my peers, I used the information given to me (colour code keys) to draw new programs for the ozobot. I tested my colour codes as I completed the lessons and tried out different ideas, especially if the first one didn't work.	I used the information given to me (colour code keys) to draw new programs for the ozobot. I tested my colour codes as I completed the lessons and tried out different ideas, especially if the first one didn't work.	I used the information given to me (colour code keys) to draw new programs for the ozobot. I tested my colour codes as I completed the lessons and tried out different ideas, especially if the first one didn't work. I helped others test and program their ozobots too.
Testing (debugging)	With help from my teacher or my peers, if I made a mistake with my colour code I debugged (fixed the code). I tested my code With some help from my teacher or my peers, I learned from any mistakes that were presented (trouble shoot) With some help from my teacher or my peers, I made appropriate changes and tested my code again	With some occasional assistance from my teacher or my peers, if I made a mistake with my colour code I debugged (fixed the code). I tested my code With some occasional assistance from my teacher or my peers, I learned from any mistakes that were presented (trouble shoot) With some occasional assistance from my teacher or my peers, I made appropriate changes and tested my code again	If I made a mistake with my colour code I debugged (fixed the code). I tested my code I learned from any mistakes that were presented (trouble shoot) I made appropriate changes and tested my code again	If I made a mistake with my colour code I debugged (fixed the code) and I also helped others debug. I tested my code and helped others test their code. I learned from any mistakes that were presented (trouble shoot) and helped others learn from their mistakes. I made appropriate changes and tested my code again and helped others change and test their code.
Making	With support from my teacher or peers, I can create lines and colour codes that function properly.	With occasional support from my teacher or peers, I can create lines and colour codes that function properly.	I can create lines and colour codes that function properly.	I can create lines and colour codes that function properly and help others with their code (line and colour) creations.
Sharing	With support from my teacher or peers, I can share (show and explain) my ideas and ozobot work with my peers and my teacher.	With occasional support from my teacher or peers, I can share (show and explain) my ideas and ozobot work with my peers and my teacher.	I can share (show and explain) my ideas and ozobot work with my peers and my teacher.	I can share (show and explain) my ideas and ozobot work with my peers and my teacher and assist others with communicating their work.

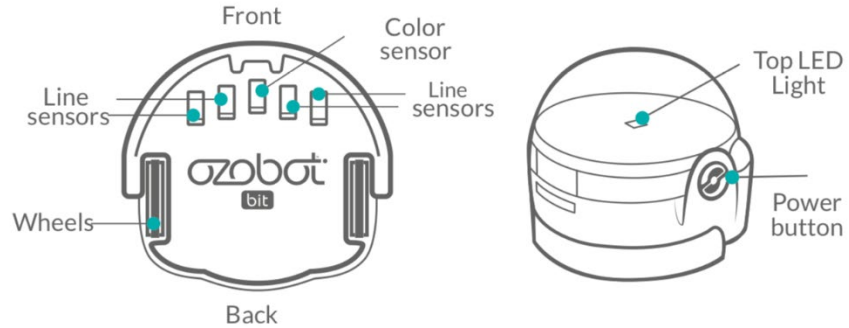
COMPUTATIONAL THINKING

	Criteria for proficient performance	Student Self-Evaluation Reflection - Explanation - Evidence	Teacher Feedback Evidence of how the program meets or exceeds criteria and areas that need improvement
Decomposition	I can explain an ozobot's multi-step path by defining each step separately.		
Pattern Recognition	I can recognize patterns in the colour code (how you put the colours in a certain order).		
Abstraction	I can recognize the colour code blocks represent complex behavior		
Algorithmic Thinking	I can combine code blocks to create a multi-step process		
Logical Reasoning	I understand if I create code blocks the ozobot will follow the instructions given.		

GET TO KNOW YOUR OZOBOT BIT

HARDWARE

Watch this video <https://youtu.be/OK8UytImflw> in order to get a video overview of the OzoBot Bit.



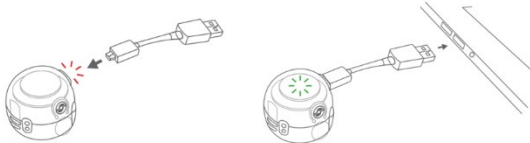
POWER ON & CHARGE

Power on or off with a single click of the bot's button.

DO! Try turning your bot on. Say hi!



DO! Charge a bot.
If a bot's LED lights **blink red**, it needs to charge. LEDs turn **solid green** on a full charge. Bots require up to 60 minutes to fully charge.



DO! Calibrate your bot to the black spot following steps below

- 

1 Make sure your bot is powered off, then place your bot on black circle.
- 

2 Press and hold your bot's power button for 2 sec. until the top LED light blinks white. Then, release power button.
- 

3 Your bot will move on the circle.
- 

4 You bot blinks green when calibrated. If your bot blinks red, start over from Step 1.

SOFTWARE AND CODING

2 WAYS TO CODE

How do Evo and Bit help teach coding *and* build creative confidence? With **two ways to code!**

- 1 Screen-free with **Color Codes**
Draw lines and Color Codes with markers, and your bots follow your lines and code commands.



- 2 Online with the **OzoBlockly editor**
Drag and drop blocks of code together to program your bots.



*You will explore the OzoBlockly editor *after* you learn the basics of Color Coding.

BASIC TRAINING – COLOUR CODES

LESSON 1: 1: DRAWING LINES AND MAPS

LESSON 2: FREE-DRAWING COLOUR CODES (ASSESSMENT)

Basic Training Lesson 1: Drawing Lines and Maps

QUICK GUIDE

PREPARATION

1. **FIND** some videos of robots to show students (we have some suggestions on page 2 of the lesson).
2. **FIND** student handbook for each student.

PART ONE: INTRODUCING ROBOTS (OPTIONAL)

1. **EXPLAIN:** A robot is a machine that can be programmed to do certain tasks.
2. **ASK:** “Do you have any robots at home?” “Have you seen any robots?” “What was their job?” “What did they look like?”
3. **SHOW:** Videos of robots in action.

PART TWO: INTRODUCING OZOBOT

1. **EXPLAIN:** Ozobot is a robot you can program yourself! One way to program Ozobot is to use markers, paper and color codes to tell Ozobot where to go and what to do.
2. **DISTRIBUTE:** Ozobots (one per student or group of no more than 3)
3. **DO:** Give students a minute to observe the different parts. Point out the power button and practice turning it on with your students.

PART THREE: CALIBRATING OZOBOT

1. **DISTRIBUTE:** Calibration Dot Templates and black marker (in student handbook)
2. **EXPLAIN:** Calibration is adjusting something to make sure it is accurate. Ozobot needs to be calibrated to make sure it can tell the difference between the white and colored part of the paper.
3. **DO:** Have students fill dot in with black marker and practice calibrating.

PART FOUR: OZOBOT FOLLOWS LINES

1. **DISTRIBUTE:** Handout 1 (In student handbook) and black markers.
2. **DO:** Students place their bot anywhere on the line. It will stop at a line break.
3. **EXPLAIN:** Ozobot’s optical sensors are Ozobot’s “eyes.”
4. **DO:** Tell students to help Ozobot get from the start to finish by drawing lines that are “just right” for Ozobot to follow (about $\frac{1}{4}$ in (5mm)). Point out the chisel tip if using Ozobot markers.

PART FIVE: TIPS FOR DRAWING CORNERS AND CURVES

1. **DISTRIBUTE:** Handout 2 (In student handbook).
2. **EXPLAIN:** When you draw corners and curves, it is still necessary to make the line $\frac{1}{4}$ in (5mm) thick for Ozobot to follow it.
3. **DO (OPTIONAL):** Have students practice free-drawing corners and curves by turning the paper over and drawing their own path.

PART SIX: DISCOVER OZOBOT’S LANGUAGE

1. **DISTRIBUTE:** Handout 3 (In student handbook).
2. **DO:** Have students fill in the Color Code boxes according to the instructions on the handout.
3. **ASK:** “What is each color code telling Ozobot to do?”

PART SEVEN: ORDER MATTERS

1. **DISTRIBUTE:** Handout 4 (In Student Handbook)
2. **DO:** Have students fill in the Color Code boxes according to the instructions on the handout.
3. **ASK:** Which color code does Ozobot react differently to from when it reads it backward?
4. **EXPLAIN:** Some codes are symmetric (has the same meaning both ways Ozobot goes over it). Some codes, though, are non-symmetric (have different meanings when Ozobot reads it from the opposite way).

PART EIGHT: HOW DOES OZOBOT MAKE A CHOICE?

1. **DISTRIBUTE:** Handout 5 (In Student Handbook), direct attention to the first intersection.
2. **EXPLAIN:** Ozobot has three choices when it gets to the intersection: go left, right, or straight. Place Ozobot at the start and see which color it “chooses.” Repeat several times.
3. **ASK:** “Does Ozobot always choose the same color?” “Can you know which way Ozobot will turn?”

PART NINE: TELLING OZOBOT WHICH WAY TO GO

1. **HAND OUT:** Color Code Reference Sheet (Optional: project it for the whole class)
2. **EXPLAIN:** “Unless you tell Ozobot what to do when it reaches an intersection, it will make a random choice about which way to go.”
3. **DO:** Have students choose a color code that will cause Ozobot to always choose [green] and test it out.

PART TEN (ACTIVITY): CODING CHALLENGE

1. **DISTRIBUTE:** Handout 6 intermediate (In Student Handbook)
2. **EXPLAIN:** Use the color codes in the boxes to help Ozobot get from home to school without getting stuck or lost! Note: You may need to point out that each color code is used only as many times as indicated next to the color codes in the blue box.

REFLECT/REVIEW

ASK: “What is the difference between a symmetric and non-symmetric color code?”

“Do you have any tips for your classmates about how to make sure Ozobot reads the color code in the correct order?”

“Which non-symmetric color codes were similar when they were read backward and forward? Which were the most different?”

BASIC TRAINING LESSON 1: DRAWING LINES AND MAPS

MATERIALS (PER GROUP)

- Set of black, green, blue, and red markers (Optional: include other bright colored markers you already have to test Ozobot's color-sensing capabilities)
- Handouts 1, 2, 3, 4, 5, and 6 intermediate (In Student Handbook)
- Plain large white paper or butcher paper
- 1 Bit (charged and calibrated)
- Calibration Dot Templates (In student Handbook)
- White Avery™ mailing labels (optional- for re-do's)
- Color Codes Reference Sheet (set of laminated sheets in the coding cart) (also can put on projector screen if you have in your classroom)
<http://play.ozobot.com/print/guides/ozobot-color-codes-reference.pdf>

PREPARATION

- Print a copy of each handout for each individual or group (in Student Handbook).
- Queue up some exciting robot videos for Part One. Some suggestions are given, but you may share any video you feel your class will enjoy. If your class is already very familiar with robots, feel free to skip to Part Two: Introducing Ozobot.

PART ONE: INTRODUCING ROBOTS

EXPLAIN: A robot is a machine that can be programmed to do certain tasks. They are used for many different things, including keeping humans safe from dangerous or boring jobs, getting into small places, doing jobs faster and better than humans can, and much more. From exploring space and the ocean to helping out with household chores, robots can be found all over. They come in all different shapes and sizes.

DO: Show some videos of robots in action (Feel free to find your own videos/articles/materials that are suitable for your individual classes' interests!)

Amazon's speedy shipping robots: goo.gl/daVH7m

The Mars Curiosity Rover: goo.gl/1mjUrb

A robot that goes to school in place of a student with a medical condition:

goo.gl/EnVgKa

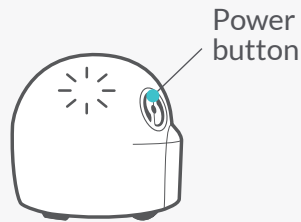
This train doesn't run on tracks; it has been programmed to follow lines:

goog.gl/dKvajH

PART TWO: INTRODUCING ROBOTS

EXPLAIN: Ozobot is a robot you can code! For the next few days, we will be learning one way to program Ozobot: using markers and paper.

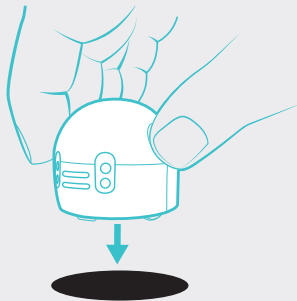
DO: Pass out Ozobots and give students a minute to observe the different parts. Point out the power button (you may need to show this on an Ozobot without the skin on so the button is more visible). Practice turning it on and off with your students.



PART THREE: CALIBRATING OZOBOT

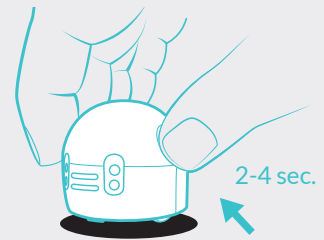
Paper Calibration

1 Make sure bot is powered off, then place it in the black circle.

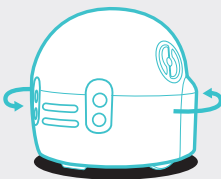


2

Press & hold bot's Power Button for 2-4 sec. until top light blinks white. Then, release Power Button.

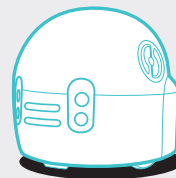


3



Your bot blinks green when calibrated. If bot blinks red, start over from step 1. Evo also spins during calibration.

4



Press the Power Button to turn Bit back on. Evo powers on automatically after calibrating.

EXPLAIN: Calibration is adjusting something to make sure it is accurate. Ozobot needs to be calibrated to make sure it can tell the difference between the white and colored part of the paper. It is important that before we begin we practice calibrating Ozobot, because it will help it follow lines and read Color Codes properly. Ozobot will need to be calibrated whenever you begin using it, change paper or playing surface, or lighting conditions change.

DO: Have students practice calibrate by: Drawing a black dot slightly bigger than Ozobot, then placing Ozobot on the dot

Holding the power button (about 2-4 seconds)until Ozobot flashes white, and

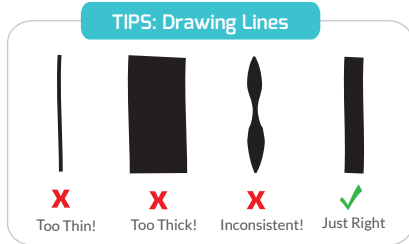
Letting go of the power button, being sure it is successful (the light flashes green after it moves outside the circle).

Since this is such an important part of Ozobot's proper functioning, you may want to go around the classroom to check on student progress.”

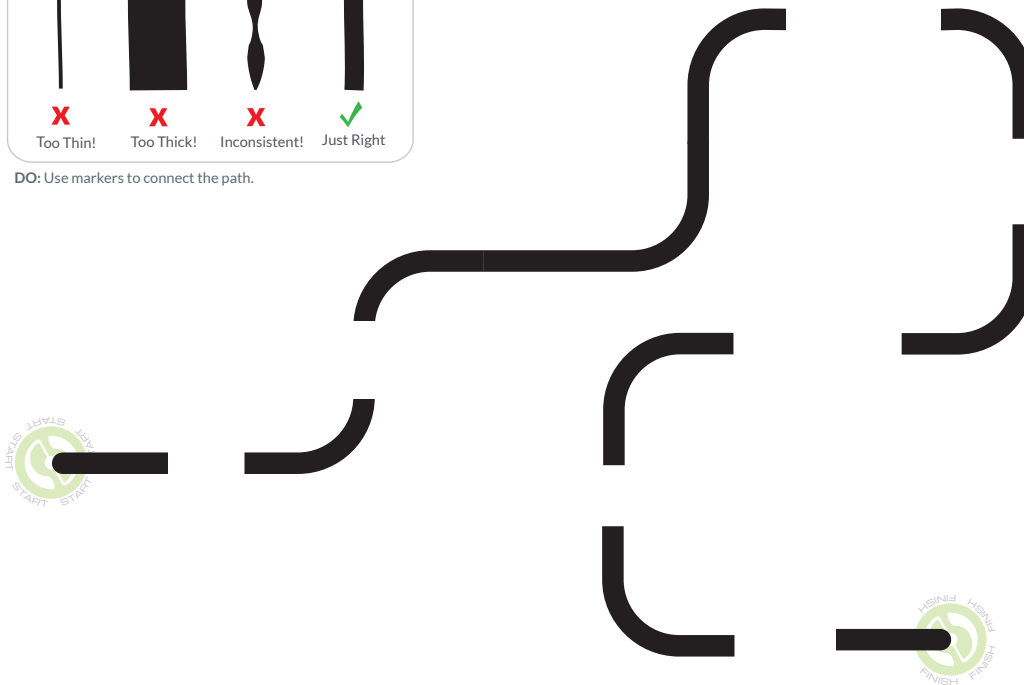
PART FOUR: OZOBOT FOLLOWS LINES

DISTRIBUTE: Handout 1 (in Student Handbook) + One black marker per student

Name: _____



DO: Use markers to connect the path.



My bot made it from start to finish!

DO: Have students turn their bots on and place it anywhere on the start. It will stop at a line break.

EXPLAIN: Have students turn Ozobot over (while it is powered on) and see what's underneath. On the bottom, you can see 5 openings (BIT) with lights shining out of them. An optical sensor lives in each of these openings. These sensors are Ozobot's "eyes." Each of the sensors sees how bright the paper underneath is. This way, Ozobot can differentiate the black and white parts of the paper to know where the line is!

DO: With just markers and paper, you can create your own line for Ozobot to follow! To make sure Ozobot can see and follow your line, it needs to be about ¼ in (5mm). Tell students to help Ozobot get from the start to finish by drawing lines that are "just right" for Ozobot to follow. If you are using Ozobot markers, you can point out how the flat end of the chisel tip is the perfect thickness to be able to draw a line in one stroke.

PART SIX: DISCOVER OZOBOT'S LANGUAGE

DISTRIBUTE: Handout 3 (in Student Handbook)

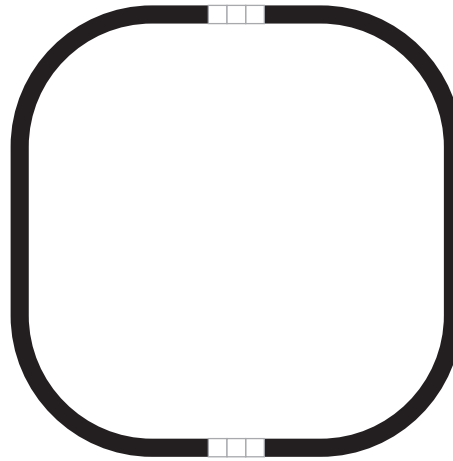
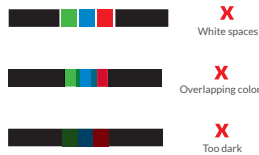
DO: Color in the code boxes
Fill in the Color Code boxes!

Name: _____

Top box:  Bottom box: 

My bot _____ when it read this Color Code 

Tips for Drawing Color Codes



My bot _____ when it read this Color Code 

ozobot  edu
HANDOUT 3

EXPLAIN: Ozobot can see many colors, but when certain combinations of colors (red, green, blue, and black) form patterns in the form of small, subsequent boxes, they give an instruction to Ozobot.

DO:

1. Have students color in the top set of boxes (blue-green-blue) and bottom set (blue-red-blue), challenging them to try to fill in the box completely yet without going over it more than once since Ozobot won't be able to read the colors if they are too dark.
2. Have students turn Ozobot on and watch as Ozobot navigates the lines and reads the codes (notice how Ozobot flashes the color of the box it is reading).
3. If Ozobot is unable to read the codes of some of the students, you can give them a new sheet of paper OR cover the code with a white sticker and have them try again. Make sure the colors are not overlapping or too dark.

ASK: What do you think these patterns (commands) are telling Ozobot to do?
Answer: blue-green-blue: turbo speed & blue-red-blue: u-turn

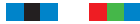
EXPLAIN: Ozobot has one optical sensor that senses colors (for Bit, it is the middle one in the row of openings, and for Evo it is the opening between the wheels).

PART SEVEN: ORDER MATTERS

DISTRIBUTE: Handout 4 (in student Handbook)

Do: Use the color codes below to fill in the blank boxes. Then, have Ozobot read the color codes from both ends of the lines. Which Color Code means something different when it is read backward? Why?

Name: _____



I discovered that # _____ means something different when Ozobot reads it the other way!

DO:

1. Have students color the top code box (blue-black-blue), again reminding them to try not to make the boxes too dark. Then, have them color the second set of boxes (blue-green-red), left to right.
2. Have students place Ozobot on the beginning of the top line and watch as it follows the line and reacts to the Color Code.
3. Starting from the opposite end of the line, watch as Ozobot follows the line and reacts to the color code.
4. Repeat steps 2 & 3 with the bottom line.

ASK: Which code does Ozobot react differently to from “Start 2”? Answer: the bottom code. From start one, it goes really fast, but on the way back, it goes really slow.

EXPLAIN: Order matters! Some codes are symmetric (meaning either way you turn it, Ozobot reads the same order of colors). The left code is symmetric. Either way Ozobot reads it will be red first, then blue, then red again. Some codes, though, are not symmetric, like the second one. When you started from the top, Ozobot read blue-green-red, which means “nitro boost.” When you started from the bottom it read red-green-blue, which means “snail dose.”

PART EIGHT: HOW DOES OZOBOT MAKE A CHOICE?

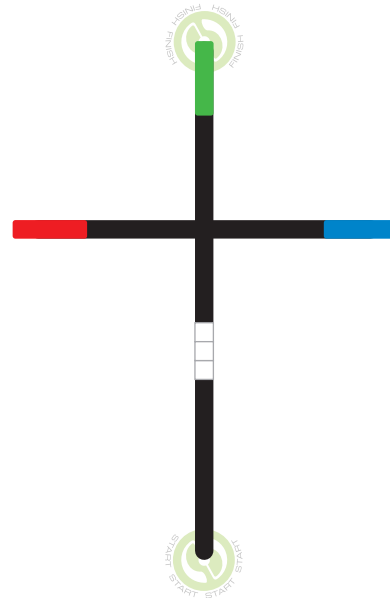
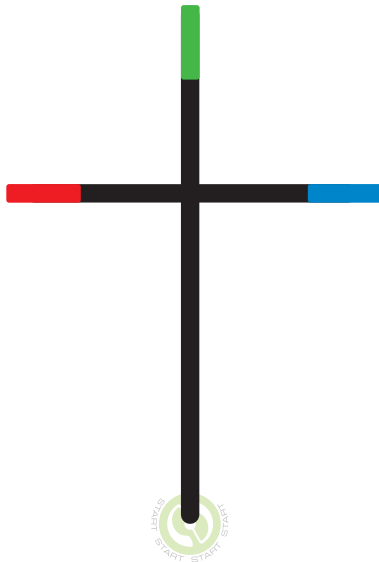
DISTRIBUTE: Handout 5 (in Student Handbook) and markers

Ozobot has three choices: Go left, right, or straight.
 DO: Place Ozobot at the start and see which it chooses. Repeat several times to see if Ozobot always chooses the same way.

DO: Look at your Color Code Reference Sheet.
 Which Color Code can you add to the blank boxes so that your bot always reaches the finish?

Name: _____

Part 1 →



azobot edu
HANDOUT 5

EXPLAIN: When you put Ozobot at the start, it has three choices: to go left (to red, right to blue, or straight to green. Let's see what Ozobot chooses.

1. Place Ozobot on the start.
2. Press the power button to turn Ozobot on.
3. Watch which color Ozobot chooses.
4. Repeat several times .

ASK: Was there any way to know which color your bot would choose?

EXPLAIN: Ozobot's decisions are a bit like flipping a coin. There is no way to know before the coin is tossed whether it will land on heads or tails. Similarly, which way Ozobot turns is a matter of chance. Just like there is no way to know for sure which side a coin will land on, Ozobot's decisions are random with no logic behind them.

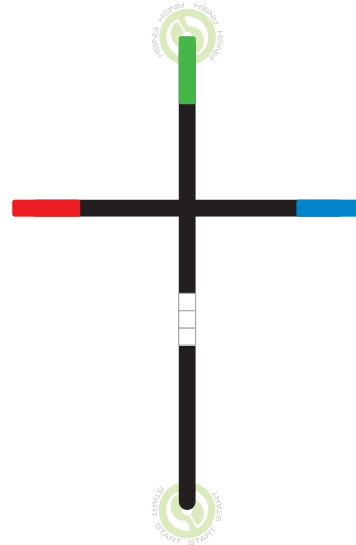
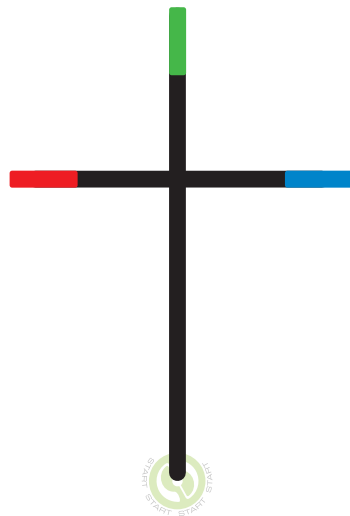
PART NINE: TELLING OZOBOT WHERE TO GO

DISTRIBUTE: Color code reference charts (in Coding Cart) and markers

Ozobot has three choices: Go left, right, or straight.
DO: Place Ozobot at the start and see which it chooses. Repeat several times to see if Ozobot always chooses the same way.

DO: Look at your Color Code Reference Sheet. Which Color Code can you add to the blank boxes so that your bot always reaches the finish?

Name: _____



ozobot edu
HANDOUT 5

DO: Have students take out Color Codes Reference Sheet (or display one on a projector) and look now at the second intersection with the blank code boxes. Now, using the handout with the blank code, ask them to find the code that will allow Ozobot to always reach the end of the line.

EXPLAIN: Unless you tell Ozobot what to do, when given a choice, it will choose randomly. If you want Ozobot to do something specific when it comes to another line, you must program it with a Color Code.

PART TEN: CODING CHALLENGE

DISTRIBUTE: Handout 6 intermediate (In Student Handbook)

azobot edu
HANDOUT 6 INTERMEDIATE

MISSION: Help Ozobot make it from home to school without getting stuck or lost! You have all the codes you need in the blue box. Use them only as many times as indicated in parentheses.

COLOR CODES TO USE

- GO STRAIGHT (20)
- FAST (20)
- TURN (LINE END) (20)
- GO LEFT (10)
- SLOW (10)
- TURN (10)
- GO RIGHT (10)
- PAUSE (10)
- LINE JUMP LEFT (20)

azobot edu
HANDOUT 6 INTERMEDIATE

EXPLAIN: The goal of this activity is to help Ozobot find the way from home to school without getting lost or stuck!

Use the Color Codes in the blue box (you'll have to use your Color Codes Reference Sheet to see what colors to use in the empty code slots). Use each code only as many times as it says in the box, and make sure each empty code slot is filled in. When you are finished, place Ozobot at home and see if it reaches school. Try it a few times. If Ozobot gets stuck anywhere or doesn't make it to school, get a new sheet or cover the codes you wish to change with blank white stickers and try again. Have fun!

Basic Training Lesson 2: Free-Drawing Color Codes

QUICK GUIDE

PREPARATION

1. **PRINT** one copy of each handout (7 and 8) (In Student Handbook) for each student or group.
2. **PRINT** and cut out sheets of “Spaceship Driver’s Licenses.” Optional
3. **TAKE OUT** board game: Ozobot Expedition to Neptune. included in Coding Cart

PART ONE: FREE DRAWING PRACTICE

1. **DISTRIBUTE:** Handout 7 (included in Student Handbook)
2. **EXPLAIN:** The tips on the top of the sheet show to draw color codes so Ozobot can read them.
3. **ASK:** Have students look over the visuals for the tips and then attempt to draw their own Color Codes from the boxes. *Optional: demonstrate and narrate the first Color Codes, reminding students to draw black lines on both sides of the Color Code.*

PART TWO: WHERE TO PLACE COLOR CODES ON A LINE

1. **DO:** Display the “Code Placement” portion of the Ozobot Tips Sheet (Laminated copies in Coding Cart).
2. **EXPLAIN:** For Ozobot to read a Color Code, the code must be drawn between straight, black lines. It should not be on a curve, too close to an intersection, or too close to another code.
3. **OPTIONAL DISCUSSION:** Why are these rules important? Do these guidelines tell you anything about how Ozobot works? How it is built?

PART THREE: SPACESHIP DRIVER’S LICENSE TEST (Optional)

1. **DISTRIBUTE:** Handout 8 (In Student Handbook)
2. **EXPLAIN:** Soon, you will be taking your Ozobot on a journey to Neptune using what you’ve learned about drawing lines and color codes. But, before you start driving in space, let’s review what you’ve learned about the “rules of the road” with this Spaceship Driver’s License Test, to get you own Spaceship Driver’s License!
3. **DO:** Have students complete the quiz individually or in groups. Go over the answers as a class, offering the opportunity to correct “mistakes” so all students can earn their Driver’s License, and then distribute one license to each student and have them fill in their names. Can connect this to **Assessment Rubric** in VSB Teacher’s Guide.

PART FOUR: OZOBOT EXPEDITION TO NEPTUNE

DO: Divide students into groups of 3-5 and read through instructions with students.

REFLECT/REVIEW

- ASK:** “What was the most difficult of drawing color codes without having boxes to color in?”
 “Do you have any tips for the rest of the class to be successful in free-drawing color codes?”
 “What did you like about Expedition to Neptune? Were there any frustrating parts?”

BASIC TRAINING LESSON 2: INTRODUCING COLOR CODES

MATERIALS (PER GROUP)

- 1 Ozobot Bit per group (charged and calibrated)
- Markers in colors black, red, light blue and light green (we recommend you use Ozobot markers. Alternatively, choose Sharpie's wide chisel tip, Crayola classic, or Crayola pastel), per person
- Handouts 7, and 8, one per person (In Student Handbook)
- Ozobot Expedition game (link below), per group (Laminated copy in coding cart)
- Spaceship Driver's Licenses (Optional)
- Lots of plain white 8.5" x 11" paper, per group of 3-5
- Timer, per group (you can have students use the clock on the iPads for timers)

PREPARATION

- This Game has been printed, cut and laminated and is in the coding cart.
- You will need to print a copy of Handout 7, Handout 8, and "Expedition to Neptune (found at <https://portal.ozobot.com/lessons/detail/mission-to-neptune>) for each individual or group.
- Optional: Cut out game components to have them ready for students to play. Alternatively, have each group cut out their own game components (see game instructions).
- Gather materials for game: a timer per group and a stack of 8.5x11 white paper
- Please note you will need one set of markers PER STUDENT playing the game.

PART ONE: FREE-DRAWING PRACTICE

Name: _____

Tips for Drawing Color Codes

X Overlapping colors
X White spaces
X Too dark
X Different sizes
X Codes on colored lines
✓ Codes on black lines. Single color square approx. 1/4"

TRY IT! Draw the following Color Codes between black lines and see if Ozobot can read them!

ozobot edu
HANDOUT 7

EXPLAIN: To make sure Ozobot can read the codes you draw on a line, there are several things to keep in mind (refer to tips on Handout 7):

1. Make sure each ColorCode has a black line before and after (except the U-Turn line end code, which we will not be working with today).
2. Ozobot cannot read codes when they are on colored lines, only black.
3. Code boxes must all be the same size, about $\frac{1}{4}$ in. (5mm) thick. If they are too thick or too thin, Ozobot won't be able to properly sense them.
4. Make sure there are no white spaces between the boxes or between the code and the line.
5. However, take care to not allow the colors to overlap.
6. Ozobot cannot read colors that are too dark, so try not to go over a color more than once.

DO: Have students look over the visuals for the tips and then attempt to draw their own Color codes from the boxes. Optional: demonstrate and narrate the first Color Code, reminding students to draw black lines on both sides of the Color Code.

This is a good time to go around and make sure students are experiencing some success in drawing their Color Codes. It may be helpful to carry around some white sticker labels so students can re-do Color Codes if Ozobot was unable to read them as drawn.

PART TWO: WHERE TO PLACE COLOR CODES ON A LINE

DO: Display the “Code Placement” portion of the Ozobot Tips Sheet (or use a printed copy to give to students).

EXPLAIN: For Ozobot to read a Color Code, the code must be drawn between straight, black lines. It should not be on a curve, too close to an intersection, or too close to another code.


PART THREE: SPACESHIP DRIVER’S LICENSE KNOWLEDGE TEST

DISTRIBUTE: Handout 8

Spaceship Driver’s Test


Name: _____

DO: Look at the images to determine whether or not you think Ozobot would be able to read the Color Codes on the maps. Check the box to indicate your decision. If you believe Ozobot wouldn’t be able to read the Color Codes, explain why in the blank.




Ozobot would read this!

Ozobot wouldn’t read this because _____



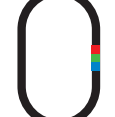
Ozobot would read this!

Ozobot wouldn’t read this because _____




Ozobot would read this!

Ozobot wouldn’t read this because _____




Ozobot would read this!

Ozobot wouldn’t read this because _____



Ozobot would read this!

Ozobot wouldn’t read this because _____



Ozobot would read this!

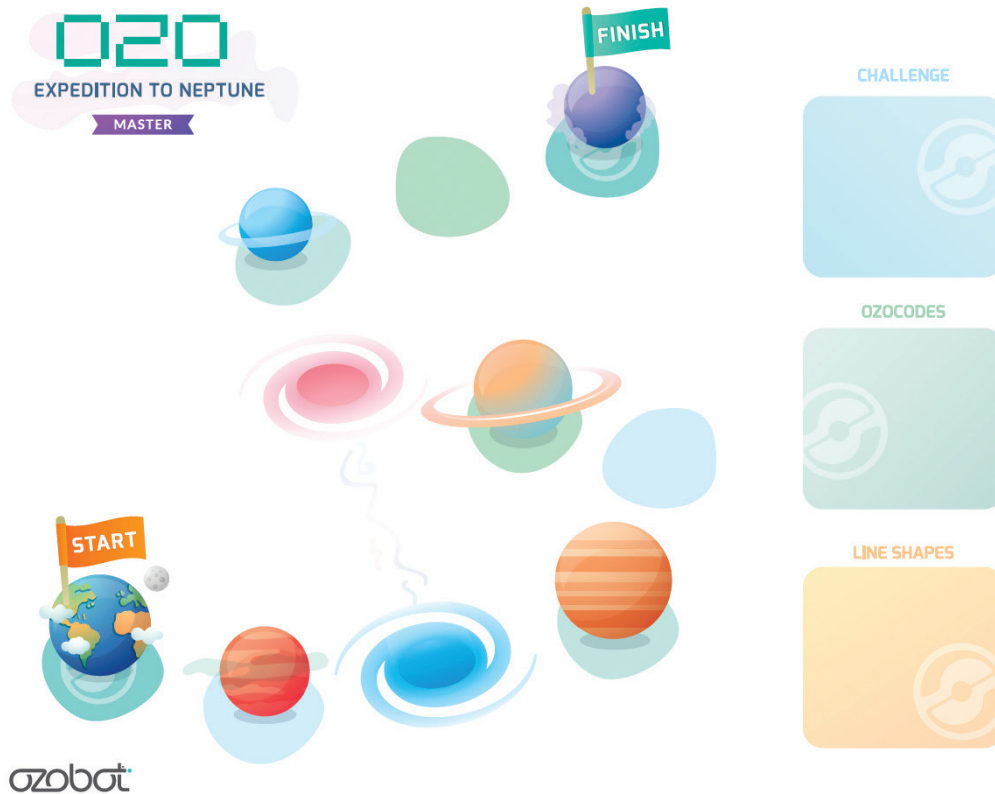
Ozobot wouldn’t read this because _____

EXPLAIN: Soon, you will be taking your Ozobot on a journey to Neptune using what you’ve learned about drawing lines and color codes. But, before you start driving in space, let’s review what you’ve learned about the “rules of the road” with this Spaceship Driver’s License Test, to get you own Spaceship Driver’s License!

DO: Have students complete the quiz individually or in groups. Go over the answers as a class, offering an opportunity to correct “mistakes” so all students can earn their Driver’s License, and then distribute one license to each student and have them fill in their names.

PART FOUR: EXPEDITION TO NEPTUNE

DISTRIBUTE: Mission to Neptune (Instructions, game board, game pieces, stacks of white paper, timer and one set of markers per student playing)



DO: Divide students into groups of 3-5, read through instructions with students, and begin the journey to Neptune!

REFLECT/REVIEW

ASK: “What was most difficult about drawing Color Codes without having boxes to color in?”
 “Do you have any tips for the rest of the class to be successful in free-drawing Color Codes?”
 “What did you like about Expedition to Neptune? Were there any frustrating parts?”

BASIC TRAINING – OZOBLOCKLY

LESSON 1: DISCOVER OZOBLOCKLY

LESSON 2: DRAG RACE (ASSESSMENT)



OZOBLOCKLY BASIC TRAINING FOR GRADES 6+

SUMMARY

This is the quickest orientation to using OzoBlockly with an Ozobot; students will open a premade program, edit it, then load it to their bot, all while learning the features of the code editor. You can follow this lesson to learn OzoBlockly yourself!

OVERVIEW

If your class is ready to use subject-based OzoBlockly lessons and don't have time to go through basic training, including the Shape Tracer games games.ozoblockly.com, or you simply want students prepared to use the editor itself, this 30-50 min interactive orientation is just what you need. It's also great for the educator to learn OzoBlockly!

Students begin by watching the OzoBlockly Tutorial video, then load an example program to practice **screen calibration**, **flash loading**, and **running a program** on an Ozobot. Next, students do the Rectangle Challenge where they edit the Square Walk program. This gets students comfortable with **creating**, **editing**, and **deleting** code blocks, and mixing blocks from other **Modes** in OzoBlockly.

Finally, students, have groups compete in a Drag Race by exploring **Mode 4 blocks** (the most customizable and detailed blocks in OzoBlockly), and practice **planning**, **computational thinking**, **debugging** and **design process**. If you need any help during this lesson, use the **Tutorial** on the right side bar of the editor.



TUTORIAL
BUTTON

LESSON OUTLINE

1. Students watch the OzoBlockly Tutorial video on their computer or device.
2. Students load the "Square Walk" program from **Examples in Mode 2**.
3. Students undertake the "Rectangle walk" **Challenge** by editing the code.
4. (*Optional task*) Students plan and execute a drag race program to compete.

PREREQUISITES

None

GROUPING

Groups of 2-3 students to 1 bot, or 1-to-1

MATERIALS

- Tablets or computers, 1 per group.
- Ozobot Bit or Evo, 1 per group
- Pencil, colored pencils
- 1 Printout of Student Handout per group (in Student Handbook)
- *(Optional) 1 Printout of Rectangle Challenge per group*
- *(Optional) 1 Printout of OzoBlockly Getting Started Guide per group*

GRADE LEVEL

Grades 6-12

DURATION

25 minutes, 50 with optional task

OZOBLOCKLY PROGRAMMING TOPICS

Free Movement, Lights, Loops, (plus, Sounds for Evo, Logic for Drag Race)

OZOBLOCKLY MODE

Modes 2-4

QUESTIONS ABOUT THIS LESSON?

Please contact us at ozoEdu@ozobot.com

LESSON

STEP 1 DISCOVER OZOBLOCKLY - 5 MINUTES

Are your students working pairs or in groups? We recommend that groups have a max of 4 students per device and robot. Allow students to take turns at the device for each step of the lesson. It's up to you if each student tries calibration and loading.

For more on how students can make the program together, check out this Code.org video on pair programming: www.youtube.com/watch?v=vgkahOzFH2Q You can even share this video with students at the beginning. We also have a lesson that teaches Pair Programming: portal.ozobot.com/lessons/detail/ozoblockly-basic-training-3

We advocate for students getting into Driver and Navigator mentalities when working together. For the programming portion of this lesson, use a timer to have all students swap roles and each get a chance, or find natural breaking points to swap.

Hand out tablets or computers to students. Send students to ozoblockly.com.

Have each student watch the tutorial video on the page with low volume or muted. All of the information is written on the screen.

After students have watched the video, have them **click on Get Started** at the top part of the page. Point out the grey space, where code is placed or moved around freely, the icons in the bottom right: **save, undo, redo, duplicate, trash**. Next, students will load an example program to get coding quickly.



STEP 2 LOAD A PROGRAM - 15 MINUTES

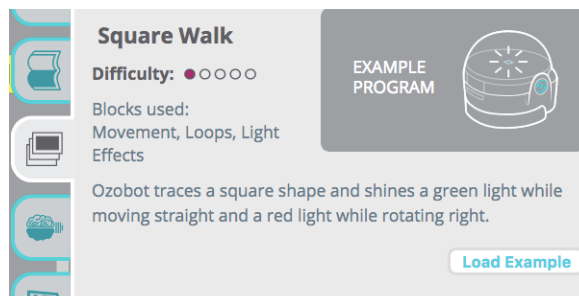
Direct all students to the logo. Select **Bit** or **Evo**, then **Mode 2** (see right) and **Examples** on the right side. Everyone will use "Square Walk" at the top (see below).



Students click 'Load Example'.

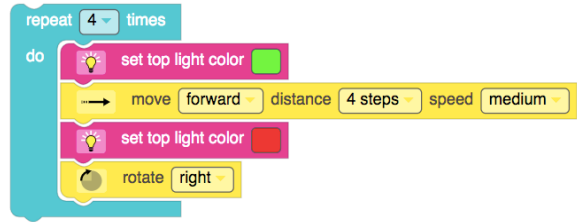
All students will now have a program where their bot draws a square with different colored sides. They should not edit this program.

Let students know, especially if they are using iPads and iPad minis, that they can move the program around by **clicking and dragging on the background only**; dragging the blocks will



move them.

Walk through the program for what the bot will do, and the logic behind loops:



1. **Repeat 4 times** means everything in the turquoise block will happen 4 times
 - a. The top LED goes green
 - b. The bot walks 1 step (about a centimeter) 4 times
 - c. The top LED changes to red
 - d. The bot turns 90° to the right

Another way to understand this is to read a-d 4 times.

Get ready to load this program into your bots!

Walk through calibration, flash loading and running the program on the bot as a class:

CALIBRATION

1. Make sure all students have 100% brightness and turned off auto-dim on their devices.
2. Click 'Flashing' or the lightning on the bottom left.
3. **Hold** the bot's power button for two seconds, until the LED blinks, and **let go**.
 - a. For Evo, the wheels will spin quickly before the sensors are ready to calibrate.
4. Immediately place the bottom of the bot on the white outline (see right) and wait until the bot's LED flashes green.
 - a. If it flashed red, try it again. Students may have tilted their bot away from the screen. Calibrate again as a class or let students follow the 'Help' instructions until all bots are green.



You and/or your students can also use the OzoBlockly Getting Started Guide

5. Tell students that they can click 'Help' for the steps in the future.
6. You only need to calibrate to the screen **once per day!**

TROUBLESHOOTING TIPS: If calibration failed, make sure brightness is 100% and students let go of the power button after two seconds!

FLASH LOADING

1. Make sure you chose **Bit** or **Evo** on the top left of the editor.
2. Your bot turned off. **Turn it back on.** (New updates keep Evo on.)
3. **Set the bottom** of the bot on the white outline for Flash Loading.
 - a. Only one outline should be activated (tap outlines to deactivate).
4. **Click on 'Load'** above the outline. This begins the flashing.

- a. **Hold the bot steady** on the colors until flashing is complete.
- b. The bot will **blink green** continuously, or it will blink red if it failed.
- c. If students need to do it a second time, let them do so following 'Help' or remembering the steps, or do it again as a class.

TROUBLESHOOTING TIPS: If loading failed, make sure calibration worked (flashed green afterwards), that the program isn't broken (look for random blocks hidden under the program or on the edge of the screen, and students don't tilt the bot from the screen while loading).

RUN THE PROGRAM

1. The bot is still on, so double-press the power button to run the program.

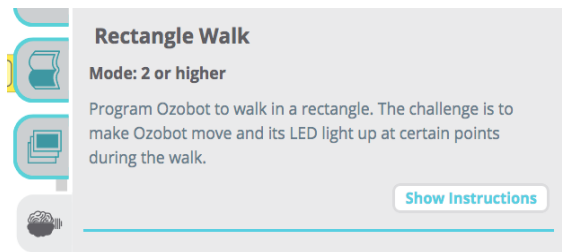
TROUBLESHOOTING TIPS: If running failed, make sure loading completed (students held the bot to the screen the whole time), that the bot is ON before double-pressing, and that double pressing is as fast as saying "one, two".

Students observe the program that their bots are performing – is it what they expected? They can run it multiple times; just turn it on again after the program completes and double press ("one, two")

STEP 3 RECTANGLE CHALLENGE - 5 MINUTES

Students keep their square walk program and **should not** edit it yet.

All students navigate to the 'Challenges' tab on the right and open 'Rectangle Walk'. They will click on 'Show Instructions' to open a one-page PDF with the steps for the challenge.



Rectangle Walk
 Mode: 2 or higher
 Program Ozobot to walk in a rectangle. The challenge is to make Ozobot move and its LED light up at certain points during the walk.
 Show Instructions

Or, hand out a printed challenge page. A copy is attached to this lesson.

Challenge

1. Students complete the challenges on the handout.
2. The challenges should be quick to complete. Have all students show you that they accomplished the challenge.
3. When students have completed the handout, they can explore OzoBlockly.

NOTE *students are often interested in the orange Line Navigation blocks – we will not be using any of those today.*

Now it's time to get creative!

OPTIONAL STUDENT PROJECT - 15 MINUTES + 10 FOR SHARING

Students are challenged to plan and build a drag race for their bot. Give students the template program link: <https://ozoblockly.com/editor#t7yfuc>. **In this program:**

1. "repeat forever" keeps Ozobot moving until "break out of loop",
2. when Ozobot sees red underneath, it will 'break out of loop' and end program.
3. at first, Ozobot gives you time to place the robot on the paper.
4. **Example Solution:** <https://ozoblockly.com/editor#s2sx9b>

Hand out the worksheet to each group for planning their program. This sheet also explains why their bot might swerve from a straight line, and how it can be fixed.

Students build a drag race track on a long white sheet of paper by coloring one end red (about 1 inch thick down the side of the page). Their bot stops when it sees red.

After 15 minutes of coding and testing, set up each group to compete against each other in heats. That is, the winner of a 2 way race will compete against the winner of the other 2 way race.

END OF LESSON