PROGRAMMING WITH COLORS

AN OZOBOT TUTORIAL FOR THE HOUR OF CODE™

Hour of Code™

What is the Hour of Code? The Hour of Code is a nationwide initiative by Computer Science Education Week (www.csedweek.org) and Code.org (www.code.org) to introduce millions of students to one hour of computer science and computer programming. The idea is to introduce the concept of coding to students and letting them know that anyone can code!

The following is a planner created by Ozobot that suggests a lesson that can be used for the Hour of Code. This lesson is "unplugged", which means that the coding is not done on a computer or tablet, but on a piece of paper. Actually, you will be giving commands to Ozobot using OzoCodes, which are special color sequences. This is also a form of programming even if no computer is being used!

This tutorial introduces your students to programming and computational thinking, but it also explains what kind of robot Ozobot is. There is also a slightly different tutorial available that covers the same topics but focuses more on robotics. If you are interested, please also take a look at the tutorial "An Introduction to Programming and Ozobot – An Ozobot Tutorial for the Hour of Code"

The type of visual programming that your students are going to learn in this lesson is particularly well suited for younger students without any prior programming experience, since it is not intimidating and uses markers which kids are very familiar with. Therefore, this tutorial can be used very well as an introduction to programming for the Hour of Code. Please find more information about the Hour of Code™ here https://hourofcode.com.

What students will learn

- What kind of a robot is Ozobot?
- How does Ozobot sense its environment and move in it?
- How can you program Ozobot with colors?
- What do you have to pay attention to when using codes to program Ozobot?

Topics

- Robotics: line following and color sensing
- Physics: optics
- Computer science: visual coding and computational thinking
- Math: random behavior

Prerequisites

No programming experience or knowledge of Ozobot is required.

Standards

ISTE 1.c – Use models and simulation to explore complex systems and issues
4.b – Plan and manage activities to develop a solution or complete a project
6.a – Understand and use technology systems

CC Math 1. Make sense of problems and persevere in solving them.
2. Reason abstractly and quantitatively.

Age/Grade Level

Grades K-12

Grouping

Lessons can be done individually or in groups of two or three students.

Materials

- Ozobots (1 per group of about 3 students, make sure they are calibrated on paper and charged)
- Blank white paper, a few sheets per group
- Markers in colors black, red, blue and green (we recommend you use Ozobot markers, Sharpie's wide chisel tip or Crayola markers), one set per group
- One copy of printouts 1 and 2 per group, and one copy of either printout 3 or 4 depending on the student's age (3 is for ages 5-8 and 4 for ages 9+)
- Printout of OzoCodes reference chart (http://files.ozobot.com/stem-education/ozobot-ozocodes-reference.pdf)

Duration

1 hour

Tips

For more info on how to plan your event for the Hour of Code, please take a look at the Hour of Code website https://hourofcode.com/how-to/events.
You can also sign up your event there. Other great pages to help you plan are:
https://hourofcode.com/how-to
https://hourofcode.com/promote/resources

Beyond this Lesson

After completing this lesson, students should be comfortable programing Ozobot with markers, lines, and OzoCodes. As the next step, you may want to do another lesson using the OzoCodes. You can see all of these lessons in the OzoCodes section of the Ozobot Lesson Library here http://portal.ozobot.com/lessons/list/ozocodes.

Contact

Please contact us at ozoedu@ozobot.com with any question about Ozobot or this lesson plan.

LESSON

1. Introduction to Coding

"How many of you have heard of computer code? Today we are going to learn what coding is, and actually program a robot to move a specific way!"

1. As a class, brainstorm the steps for arriving at and crossing an intersection in a car.
   a. Start by allowing the kids to come up with ideas without input from you.
   b. Write the steps on white board, cards or paper strips so they can be moved around and edited instructions inserted.
   c. As the steps develop, question the students so they see where the steps need to be more specific. Add details to the instructions.
2. Show video: https://www.youtube.com/watch?v=THOEQ5soVpY about "What is coding?", or any of the inspirational videos from the Hour of Code website https://hourofcode.com/us/promote/resources#videos.

2. What can Ozobot do?

Ozobot is a miniature robot, the smallest of its kind, and there are a lot of things Ozobot can do. Ozobot's most elementary ability is to follow lines. You can draw a line, place Ozobot on it and it will follow the line.

Pick markers in black and other colors and draw lines (1/4", 6mm in width) on a white piece of paper. Turn Ozobot on by pressing the button on the side. Place Ozobot on the line.

What do you see and how does it work?

You can see that Ozobot follows the lines even if they bend or cross. How does the robot do this? Turn Ozobot over and see what's underneath. On the bottom, you can see 5 openings 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 see where the white and where the colored parts are and therefore knows where the line is.

What happens if Ozobot sees the different colors?

Ozobot's LED light shines in the same color as the line it is driving on. If you look at its sensors again, you will notice that the middle sensor is bigger than the others. Actually, the middle one is a color sensor. It can detect red, green and blue colors. And, since every color can be mixed using these three colors, Ozobot can see all of them. The one exception is black. It's impossible for Ozobot to shine a black colored line, so Ozobot is using a nice blue instead when driving on black lines.

3. Ozobot at intersections

Let's imagine that you are on a game show and there are three doors: door 1, 2 and 3. There is a prize hidden behind one of the doors, but which one should you pick? Let Ozobot decide. Use printout 1 and place Ozobot on the line at the start facing down. Which way does Ozobot go?

Ozobot has three options: take a left or right turn or go straight. Does Ozobot always take a left/right turn or always go straight? Repeat the experiment a few times. What can you observe?

Ozobot seems to take some left and some right turns and sometimes goes straight, but there is no logic behind those decisions. When you place Ozobot on the start, there is no way of knowing which way Ozobot will turn. Ozobot's decisions are random.

4. Program Ozobot with OzoCodes

Let's say now that you really want Ozobot to pick door number 2. How can we tell Ozobot to go straight at the intersection? Since Ozobot can see colors, we can use a color sequence to tell Ozobot what to do.

Take a look at printout 2 and place Ozobot on the line at Start 1 facing down. Observe Ozobot's behavior. What does Ozobot do after driving over the color sequence? Repeat the experiment several times.

As Ozobot drives along the line, the color sensor detects changes in color. When Ozobot sees the sequence blue/black/red, Ozobot knows to go straight at the next intersection. Ozobot has been programmed at the factory to know many different color sequences (i.e. codes) and execute specific moves whenever Ozobot encounters any of them. Take a look at the OzoCode reference chart to see some of them.

5. Order does matter

Use printout 2 again, but this time, place Ozobot on the line at Start 2 facing up. Does Ozobot go straight or is Ozobot choosing randomly again? Repeat the experiment several times.

Ozobot seems to choose the direction randomly again. But why does Ozobot not read the "go straight" command anymore? When Ozobot drives over the color sequence, Ozobot now reads red/black/blue, but this is the "go straight" command in reverse. In fact, this sequence is not used at the moment and Ozobot has been programmed to just ignore it. This is why Ozobot's behavior is random again – just like if there wasn't any code at all. Be careful though, some codes mean something totally different when they are read the opposite way. For example, the "snail dose" is the reverse of "nitro boost". You definitely don't want to confuse those two!

Some codes are symmetric, for example "Slow" or "Fast", so it doesn't matter if Ozobot reads them from left to right or right to left. But many codes are not symmetric, like the ones you have just tried out. Make sure that you rotate them according to how Ozobot reads them. All codes on the reference chart are oriented to be read from left to right.

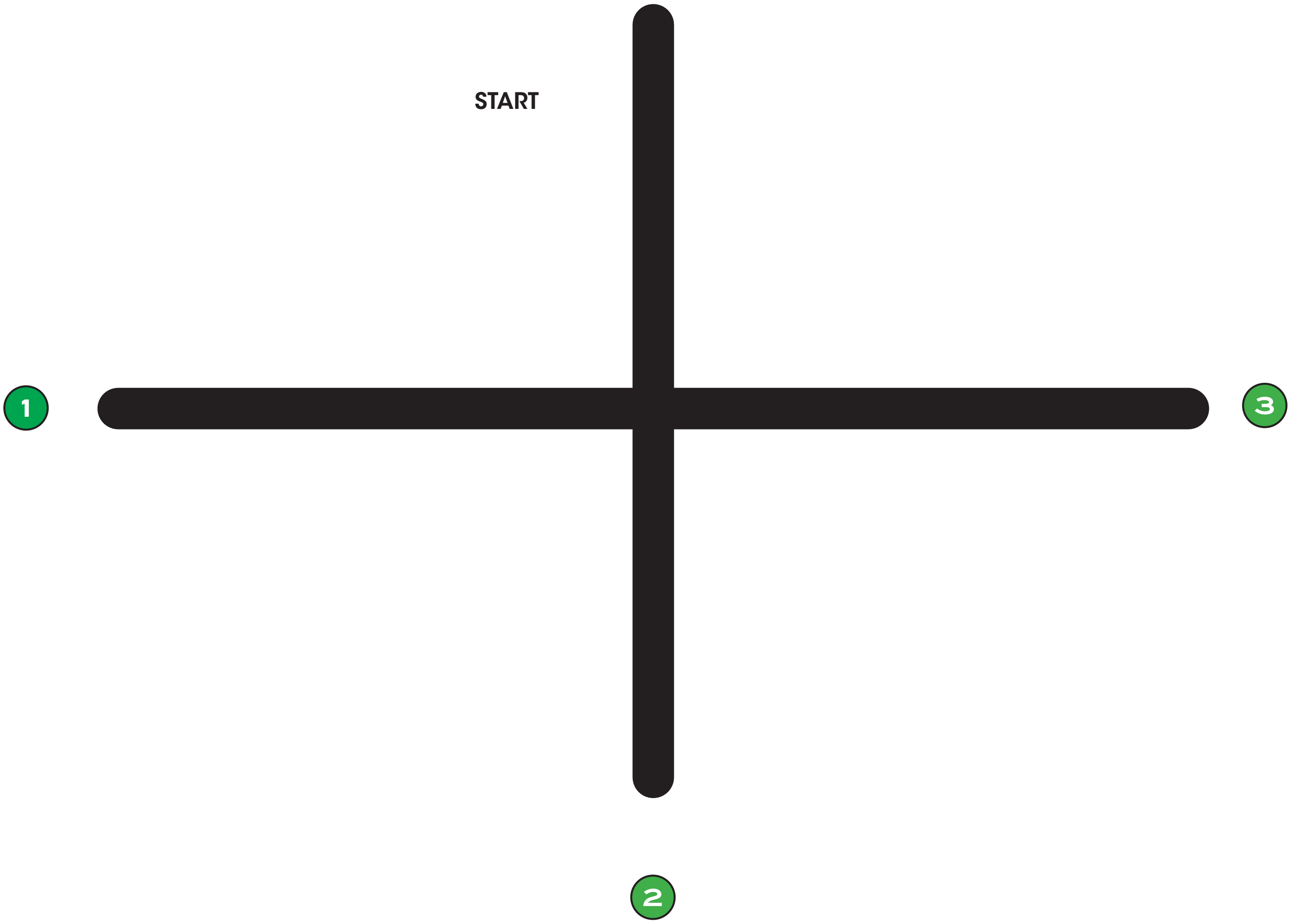
6. Help! Ozobot needs to get to school!

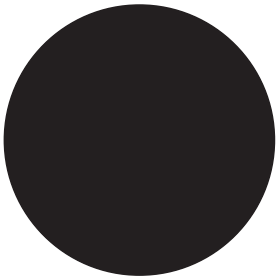
Use printout 3 to help Ozobot get to school, or printout 4 if you would like even more of a challenge.

Take a look at the printout. On the top is a house (two houses on printout 4) and Ozobot needs to take kids to school from this house. (If you are using printout 4, pick any of the two houses for Ozobot to start from.) But Ozobot might get stuck in front of a construction site, or not read the traffic signs, or simply not know the way to school. So it is up to you to guide Ozobot with the help of codes. Make sure that, when Ozobot starts from home, Ozobot always arrives at school.

To accomplish this, fill in the codes from the list of codes to use into the empty spaces on the road. Make sure to use stays only as many times as indicated. Once you have filled in all spaces, turn Ozobot on and place Ozobot on the line at a "Place here" marker. Does Ozobot arrive at school? Repeat this a couple of times, and alternate the starting house if you are using printout 4. If you found you have made a mistake, take another copy and fill in the codes differently. Have fun!

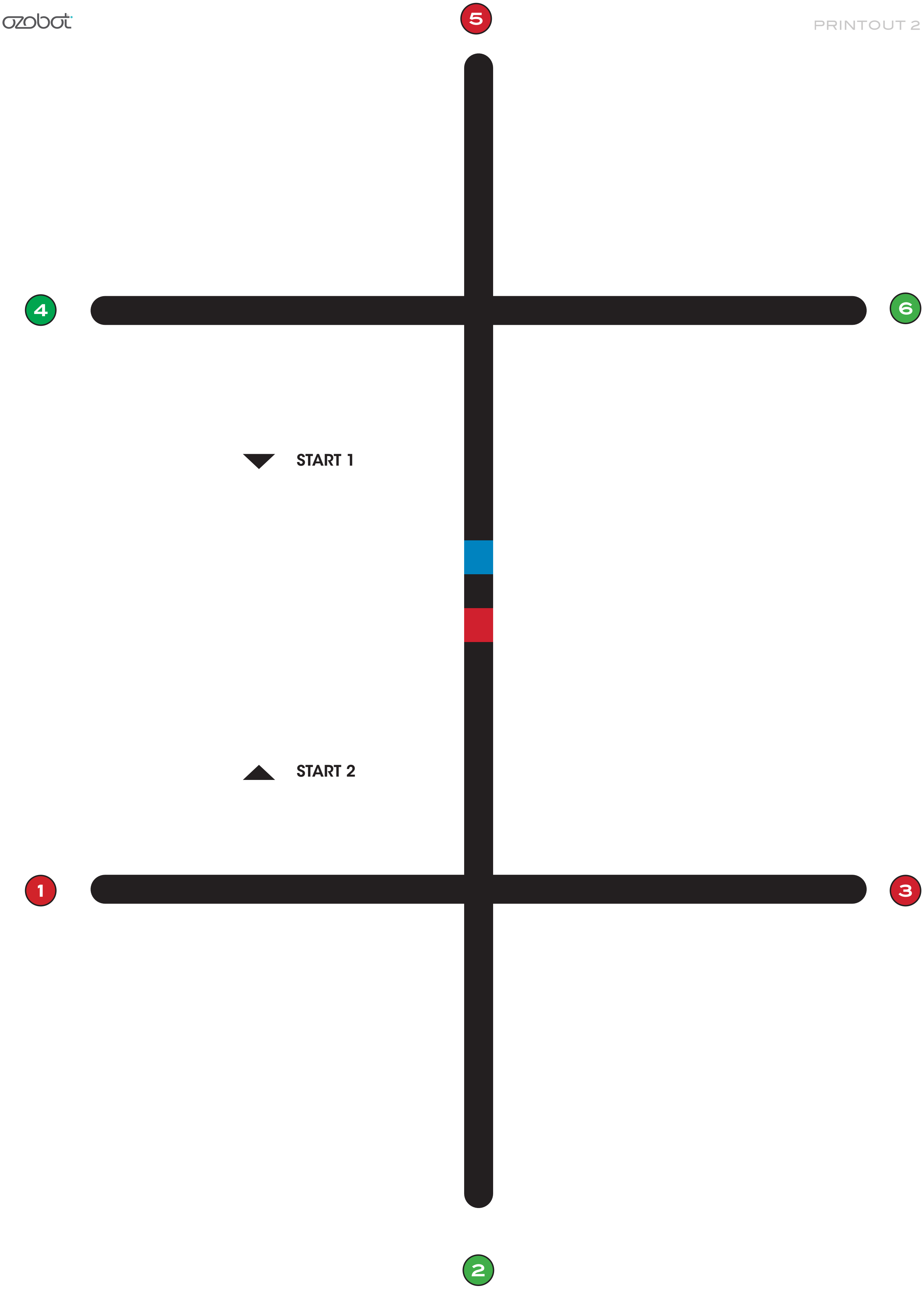
PRINTOUT 1

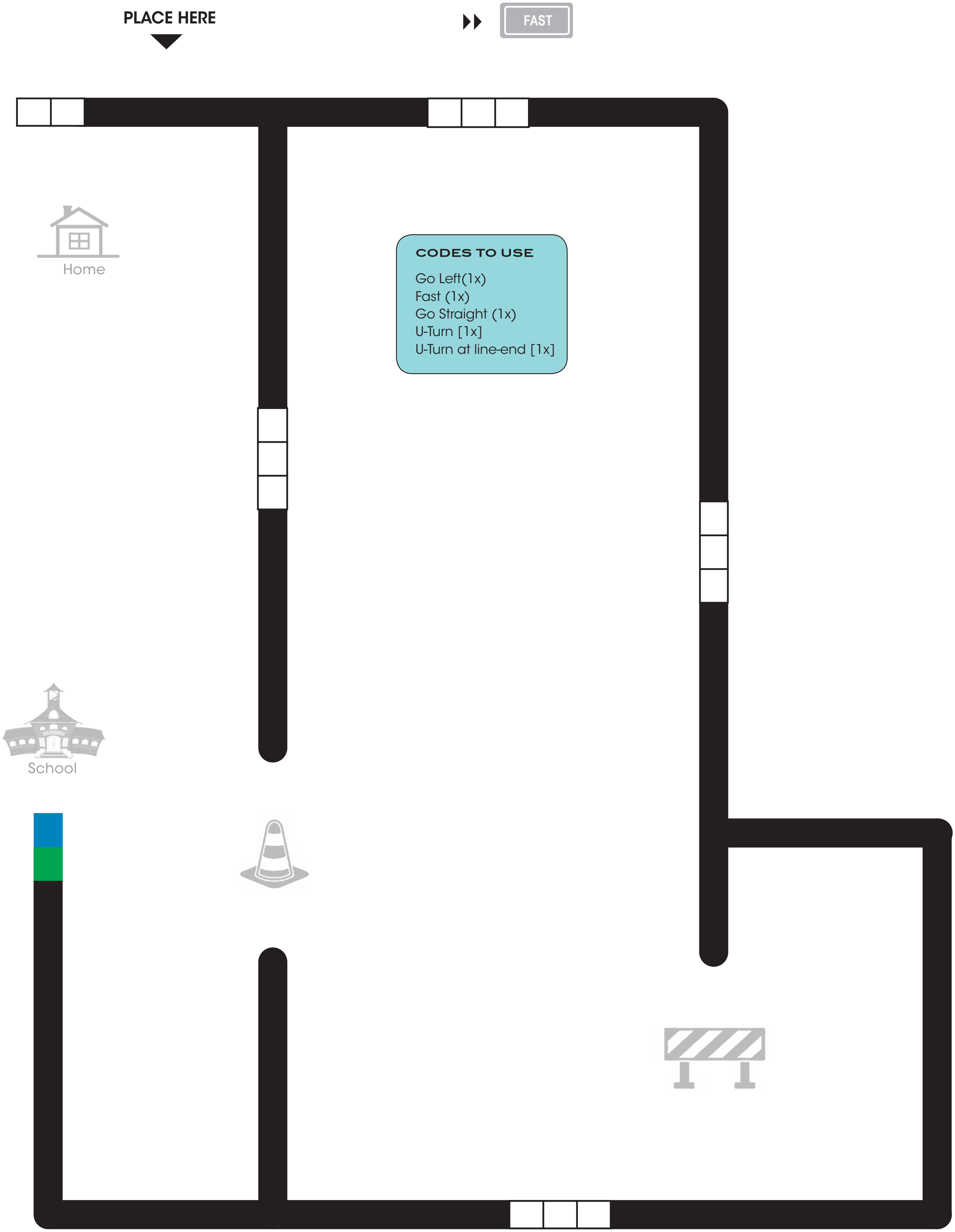




USE TO CALIBRATE

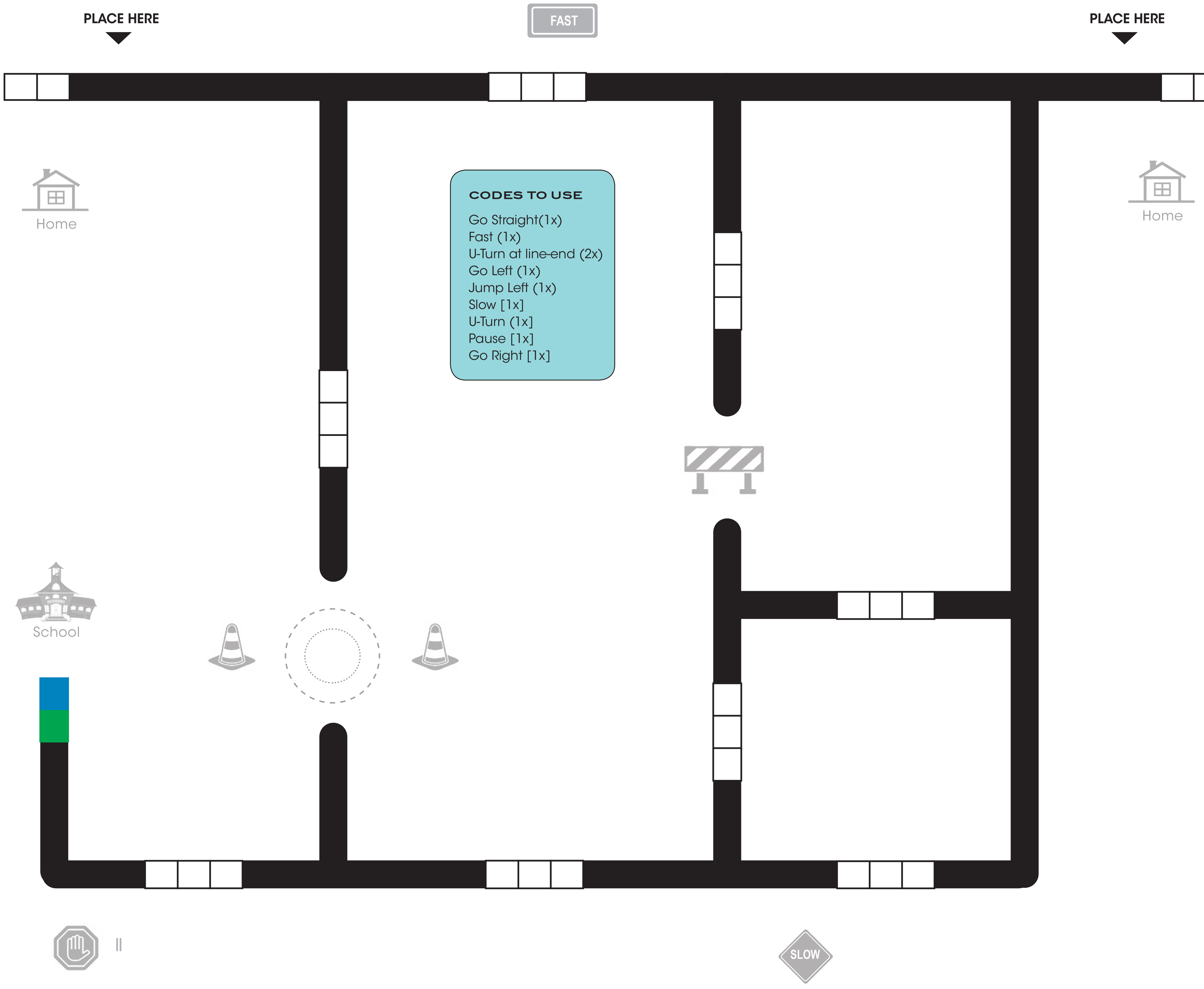
PRINTOUT 2





CODES TO USE

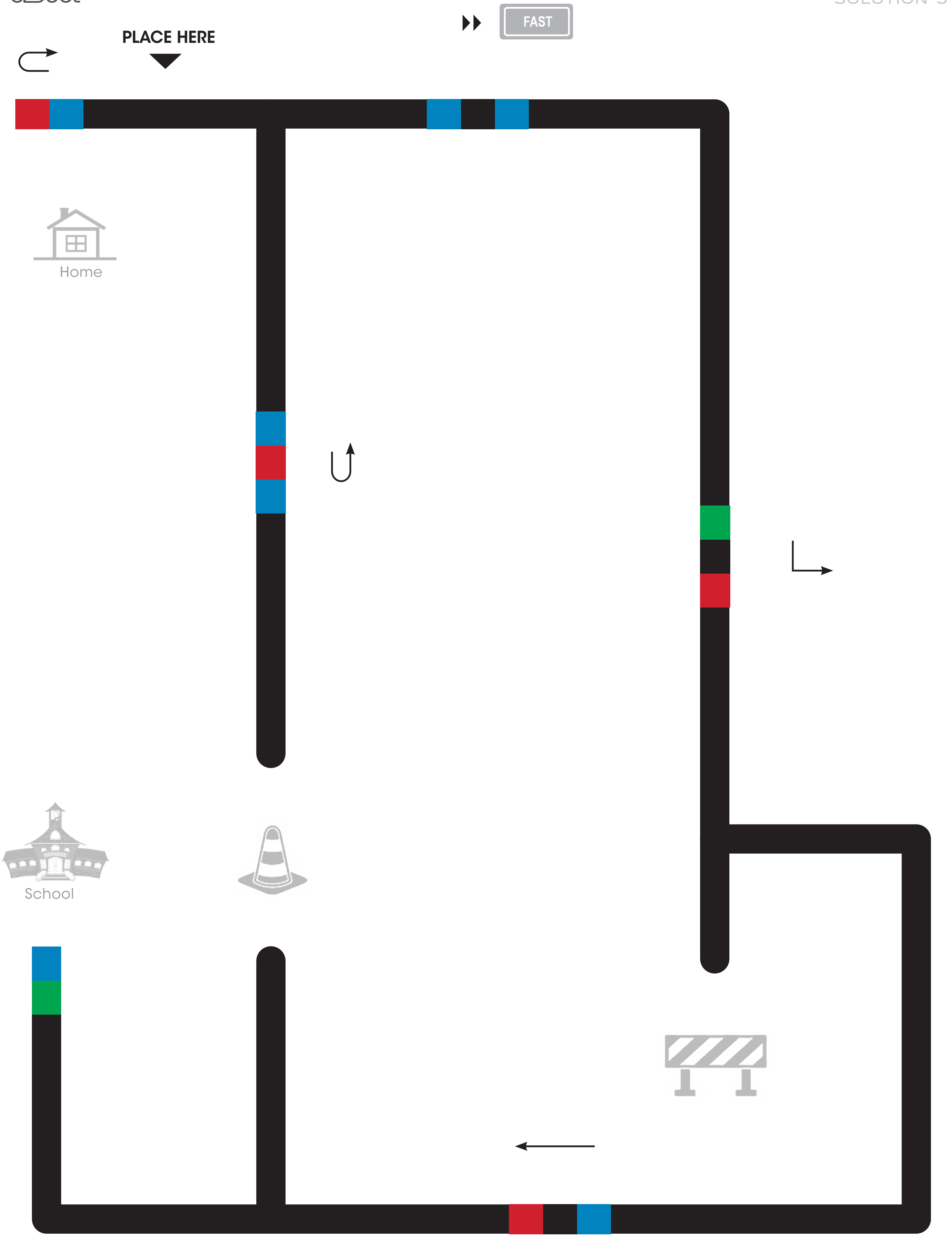
Go Left(1x)
Fast (1x)
Go Straight (1x)
U-Turn [1x]
U-Turn at line-end [1x]

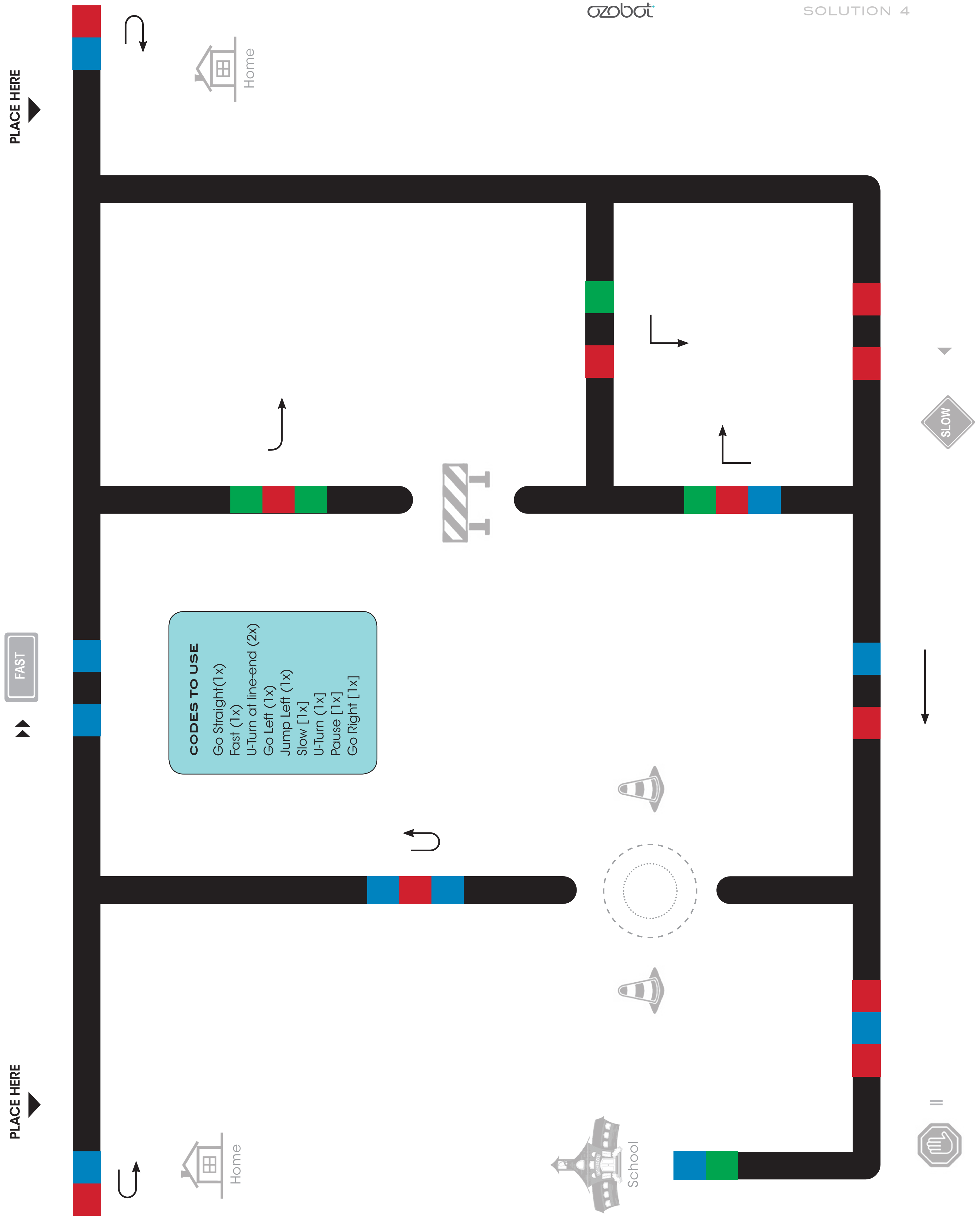


CODES TO USE

Go Straight(1x)
Fast (1x)
U-Turn at line-end (2x)
Go Left (1x)
Jump Left (1x)
Slow [1x]
U-Turn (1x]
Pause [1x]
Go Right [1x]

SOLUTION 3





CODES TO USE

Go Straight(1x)
Fast (1x)
U-Turn at line-end (2x)
Go Left (1x)
Jump Left (1x)
Slow [1x]
U-Turn (1x]
Pause [1x]
Go Right [1x]