OZOBLOCKLY SKILLS 3

Pair Programming

ESSENTIAL QUESTION

How can two people work together to code a single program?

OVERVIEW

Computer programming seems like a solitary endeavor, but it is possible to work in teams. In fact, working in teams creates more good ideas and finds the bad ones, plus any bugs in the code. It also gives students methods for listening, working together, and giving advice while coding.

The purpose of this lesson is to train young programmers to work as partners when programming, and develop great teamwork skills. Since many Ozobot lessons encourage group work, it's important to set the tone for how the group work should be accomplished.

LESSON OUTLINE

1. Students are introduced to the concept of Pair Programming and learn the tasks of a Navigator, and the tasks of a Driver, and how they differ.
2. In pairs, students complete the OzoBlockly program that will bring their Ozobot from Start to Finish on the Pair Programming map:
    a. Students must hit every color circle and not go out of bounds of each square,
    b. Students must switch between Navigator and Driver when Ozobot will enter a new square. Students practice each role twice.

PREREQUISITES

Knowledge of how to program Ozobot Bit and Evo using blocks from the Movement category. To practice, go through OzoBlockly Skills 1 and 2.

GROUPING

Groups of two students

GRADE LEVEL

Grades 2 and up

MATERIALS

- Tablet or computer with OzoBlockly editor ozoblockly.com/editor
- Ozobot Bit or Evo, one per group
- Charging cables, 1 per pair
- Printout of Pair Programming poster
- Printouts of Pair Programming Map for each pair

OZOBLOCKLY PROGRAMMING TOPICS

Free Movement: Forward, Backward, Turn, LED color

OZOBLOCKLY MODE

Modes 2 and up

DURATION

50 minutes

VOCABULARY

- *Ozobot Bit and Evo* - Little robot that can follow drawn lines, be programmed using visual codes or through the OzoBlockly programming language
- *OzoBlockly* - A visual editor which allows you to create programs by plugging blocks together. The blocks can be used to control Ozobot's behavior like movement, LED lights, etc.
- *Pair Programming* - two programmers work together, one as the Driver (at the computer) and the other as the Navigator (giving ideas and directing the course).
- *Steps* – a unit of movement forward
- *Rotate Left/Right* – turn left or right 90°, slight left or right 45°

QUESTIONS ABOUT THIS LESSON?

Please contact us at ozoEdu@ozobot.com

LESSON

PREPARATION

Watch the following videos from Code.org on Pair Programming and persistence. You will also begin the activity by showing these videos to the students:
https://www.youtube.com/watch?v=vgkahOzFH2Q
https://www.youtube.com/watch?v=eZqKqI8AvnA

Print out enough Pair Programming Maps for each pair of students. You can also print out one copy of the poster **Navigators** and **Drivers**, and keep it in sight of the students.

Give every pair of students a charging cable so their Ozobot Bit or Evo can be charged while students are immersed in programming. It would be sad if Ozobot ran out of juice on the final lap!

Finally, it is up to the teacher to decide if they will assign pairs, or let it happen naturally. It is beneficial to have both friends and acquaintances work as partners in pair programming. (*See Extension 1 at the end for ideas.*)

PART 1: INTRODUCE PAIR PROGRAMMING

Inform students that today's coding activity with Ozobots is about how two students can use one computer and one Ozobot together to create one program.

Students are introduced to the idea of Pair Programming from the founders of Generation Code. Watch the video:
https://www.youtube.com/watch?v=vgkahOzFH2Q

Show students the poster about Pair Programming, and hang it somewhere everyone can see. If they ever forget their roles, they can simply look up and find out.

Follow up that video with Persistence presented by Master Sand Sculptor Michael:
https://www.youtube.com/watch?v=eZqKqI8AvnA

Point out that failure and mistakes are for building a foundation. Failure is a part of learning, and it doesn't mean everything's over! It is a necessary step. It's very true that "frustration just means that there is something wonderful around the corner," especially in programming.

Now present the activity that students will be doing.

PART 2: PAIR PROGRAMMING ACTIVITY

Pairs program Ozobot to travel on its own through the Pair Programming Map.

RULES

- Start at the Start circle,
- Light the LED the same color as the box Ozobot is in,
- Ozobot must hit all the circles in the box. If they're numbered, they must be hit in order,
- Ozobot must enter boxes at the arrow, especially at the finish,
- Students must switch roles of Navigator and Driver every time Ozobot is supposed to enter a new box on the map it will move on,
- Students who become Driver cannot go back and change their partner's code unless they both agree there is a bug,
- To finish, Ozobot must land in the Finish box and do a fun light show or dance.

There are some key aspects students should know before they start:

1. **Debugging** must happen. That is, create a code, test it, and then change what doesn't work; repeat until the code accomplishes the task. Always test the code when you make a change. Code that doesn't work correctly or that causes a failure is a bug. Perhaps Ozobot turns the wrong way, or drives too far off track!

2. Ozobot Bit (Evo is better equipped for this) has hardware limitations so that it can turn off of a straight line quite quickly. Students will have to account for this imprecision. The next OzoBlockly lesson will actually teach students how to get deeper into the code and make Ozobot go straight!

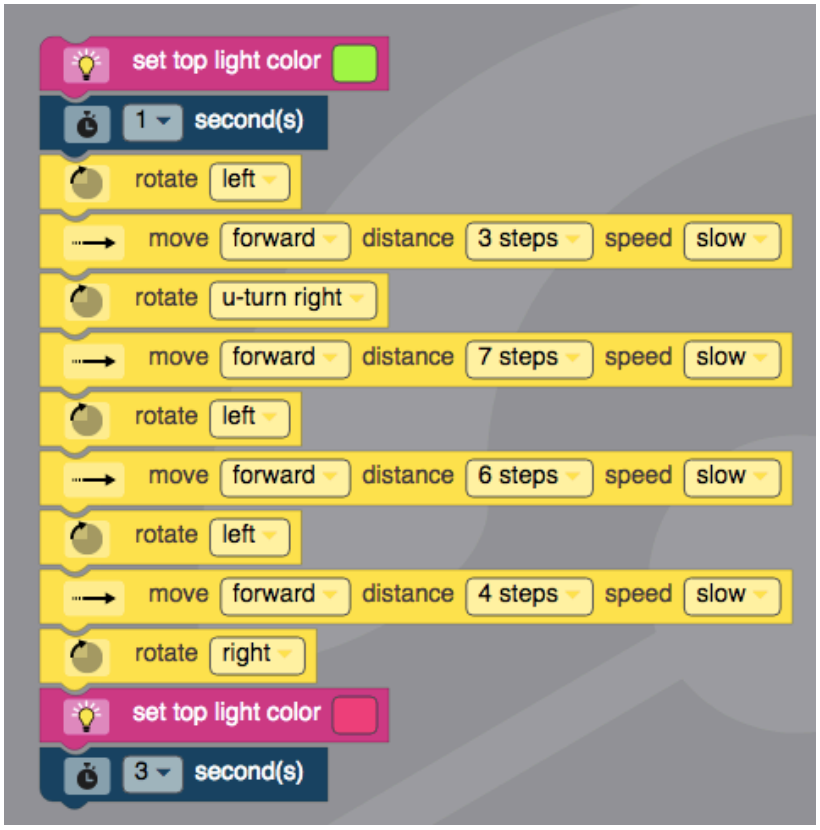
3. The location and the way that the Ozobot is facing at the start determines where it ends up and how well it hits the targets. Students should be aware that placing Ozobot is as important as writing correct code.

The word "debugging" has some interesting lore behind it. It's said that Admiral Grace Hopper, a computer scientist for the US Navy in the 1940s, pulled a moth out of her computer system (back then computers were the size of rooms). The machine couldn't work in that state!
However, the term dates back to Edison's time in the late 1800s. Still, programmers can't help but enjoy the image of pulling a literal bug out of their machine!

Now, students are ready to start! Students must choose who will drive and who will navigate first. There will be an equal amount of tries.

Once all groups have completed the code for their Ozobot they can demonstrate their maps for the whole class.

A note to the teacher: Sometimes the tendency to turn gets better or worse as Bit and Evo warm up with use. Inform students of this possibility, and, if possible, continue to charge Ozobot between tests as low battery can reduce performance.



```
set top light color 🟩
1 second(s)
rotate left
move forward distance 3 steps speed slow
rotate u-turn right
move forward distance 7 steps speed slow
rotate left
move forward distance 6 steps speed slow
rotate left
move forward distance 4 steps speed slow
rotate right
set top light color 🟥
3 second(s)
```

Example solution for the Green Box. Solutions will vary.

EXTENSION

1. In order to randomize pairs of programmers and to enforce the tasks for each role, consider the following exercise:
    a. Print out the posters as cards, 4 to a page,
    b. Print enough Navigators for one half of the class, and equal Drivers,
    c. Say you have 15 navigators – number them 1 through 15. Drivers, too
    d. Randomly pass out the cards while explaining the roles. Students must find the corresponding number to find their assigned partner for the day!
    e. Students keep cards at desk while they work,
    f. Cards can be reused for every class.
2. Got an odd number of students? Groups of three also work well. Have two navigators on each side of the driver and have students rotate roles. In this lesson we have 4 instances of coding – students can do Rock Paper Scissors for the last driver role.

NAVIGATOR

1. Sits next to driver
2. Never grabs mouse, keyboard or tablet
3. Answers driver's questions
4. Points out problems and bugs
5. Makes suggestions and plans big picture
6. Agrees on style

REMEMBER TO
SWITCH IT UP!

ozobot
PAIR PROGRAMMING

DRIVER

1. Sits at computer
2. Uses keyboard, mouse or tablet
3. Writes the code
4. Explains what he/she is doing out loud
5. Listens to advice of the navigator
6. Agrees on style

REMEMBER TO
SWITCH IT UP!

NAVIGATOR

1. Sits next to driver
2. Never grabs mouse, keyboard or tablet
3. Answers driver's questions
4. Points out problems and bugs
5. Makes suggestions and plans big picture
6. Agrees on style

REMEMBER TO
SWITCH IT UP!

ozobot

PAIR PROGRAMMING

DRIVER

1 Sits at computer

2 Uses keyboard, mouse or tablet

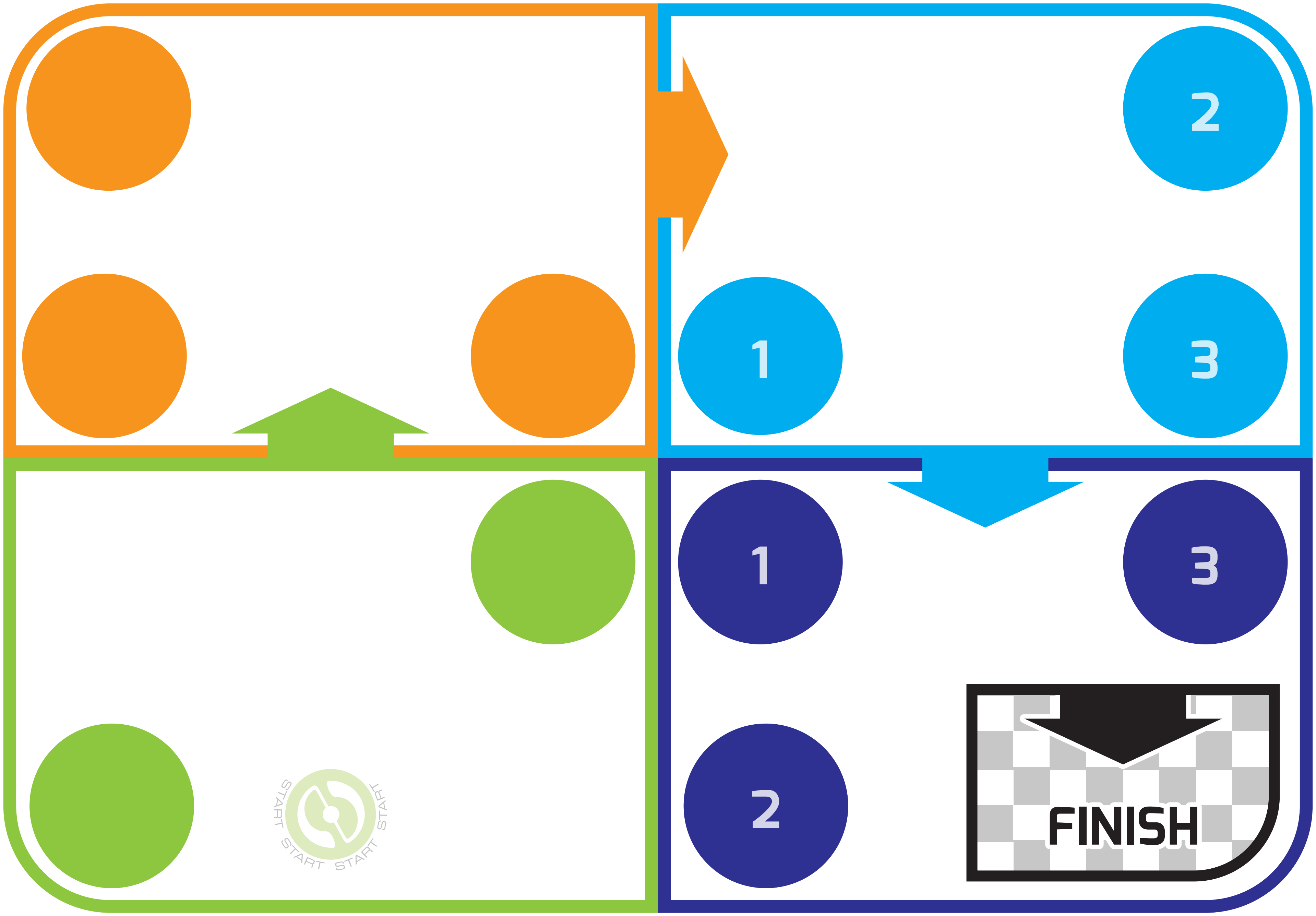
3 Writes the code

4 Explains what he/she is doing out loud

5 Listens to advice of the navigator

6 Agrees on style

REMEMBER TO
SWITCH IT UP!



ozobot PAIR PROGRAMMING MAP