University of Michigan Department of Computational Medicine and Bioinformatics

0s, 1s, and Time Machines
**Association of Multicultural Scientists**
Science Career Day
May 12, 2018

Set up Ozobot and Ozoblockly

1. Download the Evo App (https://ozobot.com/play/apps)
   a. Open the App
   b. Select the orange button Join and create an account
2. Download the two .ozocode files provided
   a. TimeMachine.ozocode (Middle school adaptation)
      i. https://github.com/GWC-DCMB/ozobotLessons/blob/master/time_machine/TimeMachine.ozocode
   b. TimeMachine_Blanks.ozocode (High school adaptation)
      i. https://github.com/GWC-DCMB/ozobotLessons/blob/master/time_machine/TimeMachine_Blanks.ozocode
3. On a computer, visit https://ozoblockly.com (Google Chrome works best)
   a. Select Get Started
   b. On the far left panel, switch settings to evo and Level 4
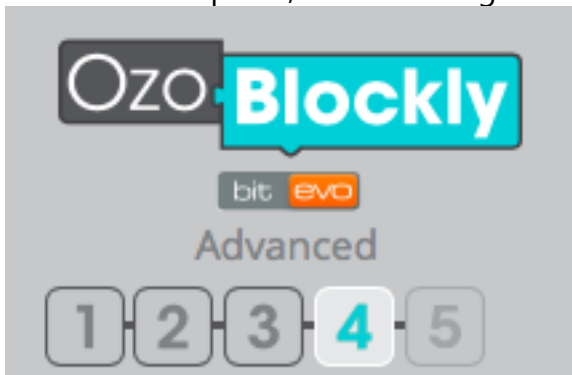
   c. On the far right panel login with your new account credentials
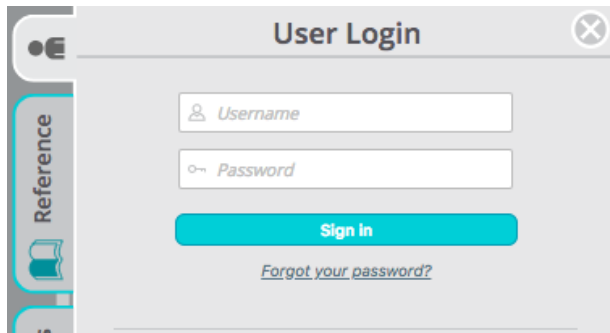
4. Under My programs, select an empty box
   a. You will be prompted to Save program. Name it "TimeMachine" or "TimeMachine_Blanks"
   b. Select the folder on the bottom

   c. Select Open file, then navigate to the .ozocode file on your computer and press open
   d. Select Save
   e. You will be prompted "Are you sure you want to save over this program?" Select Yes.
5. Turn on your Ozobot
6. Back on your phone or tablet, open the App and make sure your Bluetooth is on
   a. An Ozobot should automatically pop up under Nearby
   b. Select the Ozobot and choose connect on the bottom right
   c. Select the bottom left icon and choose Play with Code Ozoblockly from the menu
   d. Your programs from the browser will load and you can select TimeMachine
   e. Make sure AutoRun and Stay Awake are selected
   f. Your Ozobot should start executing the program when placed on the track

Objective
Use Ozobot to reinforce concepts of Boolean values and binary numbers where 0 is False and 1 is True. Use conditional statements and logic to accomplish the activity's goal of making the Ozobot follow the correct path.

Materials
Each group of 4-5 should have
- Ozobot
- Pack of markers (red, green, blue, black)
- Set of white labels to fix mistakes
- Laptop with the ozocode loaded into Ozoblockly.com
- Smartphone with the ozocode loaded into the Evo application
- Color print out of the track

Activity

First, how does Ozobot work? Ozobot follows lines and has sensors to identify the line color. Ozobot can sense black, green, red, and blue colored lines. If Ozobot reaches the end of the line, it stops moving forward. When Ozobot reaches an intersection, it will randomly choose a direction unless told otherwise. Ozobot listens to directions by executing code that you write with Ozoblocky.

Today we will use the power of code to turn our Ozobots into Time Machines! We're going to travel in time to see the bright futures ahead of each of you! While on the Time Machine, you can keep your future self from running into problems, or pitfalls, along the way. These pitfalls are in the form of intersections which lead to line ends. You must use code to avoid these line ends and keep your Time Machine moving forward. You will travel in time from present day in Wolverine pathways, through your undergraduate career, and to your future career. To create your Time Machine, use what you've learned about binary numbers, Boolean values, and conditional statements.

Middle school adaptation**:** The code is already written and students must color in the blanks. The goal is to color the lines such that they satisfy the conditional statements so the Ozobot can travel in time while helping our future-selves avoid pitfalls. For example, if Ozobot is approaching an intersection we need to tell it which direction to turn so it does not reach the end of the line (representing a pitfall). Students should read the code and figure out what it is doing so they know which sections should be which colors. Verify students' logic before they color in the map. Make sure the students neatly color in the map so the Ozobot does not get confused. Then they can test their robot by placing the Ozobot at the present time and executing TimeMachine.ozocode.

High school adaptation**:** The code is written but has blank conditional statements. However, the building blocks needed to complete the code are laid out. Students must fill in the conditional statements and color in the blanks and to travel in time while helping our future-selves avoid pitfalls. For example, if Ozobot is approaching an intersection we need to tell it which direction to turn so it does not reach the end of the line (representing a pitfall). Students should write code and color the blanks accordingly to prevent Ozobot from reaching a pitfall. Hints: Red, Green, and Blue should be used. Verify students' logic before they color in the map. Make sure the students neatly color in the map so the Ozobot does not get confused. Then they can test their robot by placing the Ozobot at the present time and executing the completed version of TimeMachine_Blanks.ozocode.

Facilitate the activity
1. Split students into groups of 4-5 with one Ozobot per group
2. Review how the Ozobot senses colors and follows lines
3. Questions and prompts to guide problem solving (before looking at the code):
   a. Which direction should Ozobot turn at each pitfall to continue moving forward in time?
   b. Brainstorm how you can make the Ozobot do this.
   c. How can you use conditional statements to achieve your goal?
   d. Students should write pseudocode using conditional statements in their lab notebooks
4. Look at the code as a group.
   a. High school adaptation: Students should figure out what the code is doing and then complete the conditional statements using the blocks available. They should decide what color each missing track should be and check their answers with the facilitator before proceeding.

    b. <u>Middle school adaptation:</u> Students should figure out what the code is and check their answers with the facilitator before proceeding.
5. Color in the missing colors for the turns so that the ozobot will not encounter any pitfalls.
6. Color in speed and cool move codes using the OzoCodes reference sheet. Encourage the students to be creative!
7. After coloring in the missing track, including the speed and cool move codes, you track is ready for testing! Execute the ozocode and place your Ozobot at the start of the track to start time travelling!

<u>Tips and Tricks</u>
- When you open the .ozocode file you may need to scroll around or zoom out to find the code on your screen
- You can view the Ozoblockly code translated to JavaScript on the right side of the browser

Instructional Concepts and Definitions

- Pseudocode - description of the instructions you want to code for a computer program, essentially an outline of the algorithm
- Source code - human-readable instructions for the computer which are translated into machine code by a compiler or interpreter (our source code is written in Ozoblockly here)
- Machine code - binary code instructions the computer can understand
- Binary - a number system with only two numbers: 0 and 1
- Boolean - a type of variable that is either true or false
- Conditional Statements (aka if statements) - a statement defining what condition must be satisfied for a specific action to be done
- For loop - allows instructions to be repeated multiple times (our for loop iterates the same conditional statements for 100 times)
- Variable - the way you store and retrieve information (our variable is color)