CODERS
Missouri State University

Lesson Name: Cutebot Obstacle Avoidance

School District:

Teacher Name:

Intended Grade Level: All

Differentiation: N/A

Accommodations:

Make accommodation appropriate for your classroom.

Material(s):

- 1 cutebot with battery pack (w/ batteries)
- 1 micro:bit with USB cable
- 1 ultrasonic sensor

Time allotment (varies according to grade level):

- TBD

Alignment with CODERS module(s):

(Cutebot)

MLS Standards:

4.AP.M.01 – Decompose (break down) large problems into smaller, manageable subproblems to facilitate the program development process.

5.AP.M.01 – Decompose (break down) large problems into smaller, manageable subproblems and then into a precise sequence of instructions.

4.AP.PD.01 – Use an iterative and collaborative process to plan the development of a program that includes user preferences while solving simple problems.

5.AP.PD.01 – Use an iterative and collaborative process to plan the development of a program that includes other perspectives and user preferences while solving simple problems.

Objectives:

- Code the cutebot to implement obstacle avoidance functionality by utilizing the ultrasonic sensor.

Background information / Activation of Prior Knowledge:

- Access to MakeCode classroom.
- Basic computational thinking skills.
- Basic knowledge of block programming.

Introduction / Anticipatory Set:

ENTRANCE TICKET (pre-writing):

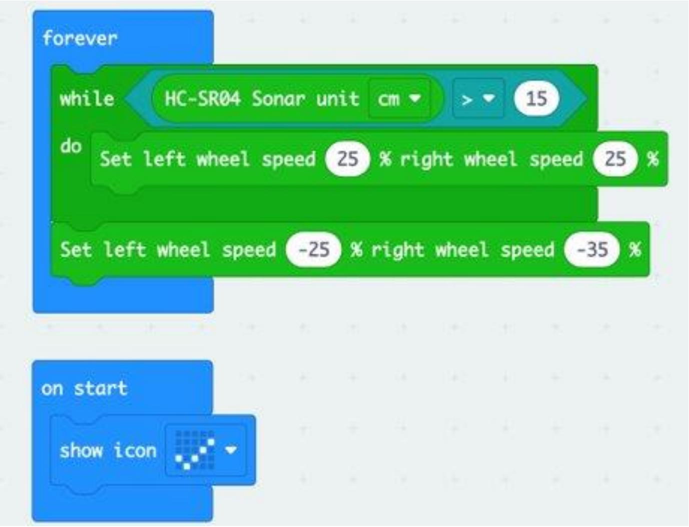
Teaching (I do):

Introduce the goal of this lesson (to implement obstacle avoidance in a robot car) and how you will do it (by using a micro:bit and cutebot with an ultrasonic sensor and a short block program). Group students into pairs and have students write out the steps for their program in addition to drawing a flowchart for their program. Once each student is done writing, have partners trade instructions and code the program based only on the instructions their partner provided them. After the majority of students have finished you can either have partners revise their instructions or you can lead the class through the "Guided Practice (We do)" section to ensure everyone has a working solution to the problem.

Guided Practice (We do):

1. **Code**
   a. Name the project "Obstacle Avoidance".
   b. The on start event will show an icon this allows the programmer to know when the new program has been successfully flashed and can assist when debugging.
   c. Then in a forever loop, while the HC-SR04 Sonar unit detects an object greater than 15 centimeters away, the cutebot should set the left and right wheel speeds equal to each other.

d. If the sonar detects an object closer than 15 centimeters set the left and right wheel speeds to a negative speed (so the cutebot moves in reverse). The wheel speeds should be different from each other though so that way the cutebot reverses but also turns as it reverses.



```
forever
  while HC-SR04 Sonar unit cm > 15
  do
    Set left wheel speed 25 % right wheel speed 25 %
  Set left wheel speed -25 % right wheel speed -35 %

on start
  show icon
```

2. Test the program

a. Plug the micro:bit into a computer and flash the program to the micro:bit. Then insert the micro:bit into the cutebot and turn on the battery pack.

3. Debugging

a. In the previous section, were any issues found in the program? Did the cutebot move too quickly? Did the cutebot run out of space? Did the cutebot successfully avoid all obstacles?

b. Go back to MakeCode and make the changes needed to fix the bugs observed in the testing phase.

c. Test the program again and repeat the debugging process if needed.

Group / Independent Practice (You do):

1. **Reflection**

    a. Have students write a reflection on their first draft of instructions. Was it enough for their partner to successfully implement it? If not, why? If it was enough, then how can it be revised to be more concise?

Notes, Reflections, Attachments