Cutebot/Micro:bit Challenge

Lesson Plan for Grades 9-12, Technology Robotics/Programming

Estimated activity time: 50 minutes

Prepared by Brandon Washington and Kathleen Hassey

OVERVIEW & PURPOSE

Students will use their knowledge of programming to program the Micro:bit and Cutebot (robot) to navigate the "race track" in the quickest time.

EDUCATION STANDARDS

International Society for Technology in Education (ISTE)

1.5. Computational Thinker. Students develop and employ strategies for understanding and solving problems in ways that leverage the power of technological methods to develop and test solutions.

1.5.d. Students understand how automation works and use algorithmic thinking to develop a sequence of steps to create and test automated solutions.

PREREQUISITES

1. Previous lessons on programming, debugging, loops and conditionals. Here are several lesson plans that cover the prerequisites: Computing Fundamentals (Programming 1), Data Handling and Selection, and Autonomous Obstacle Avoidance.

OBJECTIVES

1. Automatic Collision Avoidance: Students will work in pairs to create a program, to program the Micro:bit, and to make the Cutebot avoid collision. Students will work in pairs to use the best code to autonomously avoid obstacles in the path of the Micro:bit.

2. Line Tracing: Students will work in pairs to create a program, to program the Micro:bit, and to make the Cutebot follow the track. Students will work in pairs to determine maximum speed of each wheel to complete the turns in the track.

MATERIALS NEEDED

1. Micro:bit
2. Cutebot (Ultrasonic sensor included)
3. 3 double A batteries
4. Computer with type c-port input or USB adaptor
5. Internet access
6. Paper track 1 inch thick black line
7. Code bank resource sheet

VOCABULARY

1. Program: Set of steps in code to give a command to a computer.
2. Robot: A machine that is programmed by a computer.
3. Autonomous: Performed by a device capable of operating without direct human control.
4. Sensor: A device which detects or measures a physical property and records, indicates, or otherwise responds to it.
5. Conditional Code: Statement code that runs a certain program once a certain condition is met.
6. Loop: A code that repeats as many times as it is told to

STEPS FOR SUCCESS

Each student should have a copy. Make a visible poster of steps.

Steps for students to follow to ensure success:

Getting Started to Code

1. Make sure you have your materials.
2. Log into [https://makecode.microbit.org/#editor.](https://makecode.microbit.org/#editor)
3. Create a Team Name to name your code.

4. Once in the code, go to extensions and search "Cutebot" and double click on it to select it.
5. Begin your code by putting a team design in the start block using the "show leds in the basic drawer".

Ideas to Consider

6. Know the outcome: What do you want to happen? Make vehicle follow a track using a sensor and/or make vehicle turn with the track by changing wheel speeds.
7. Use your resources in the code bank and the information on the board about the sensors to help you build your code.
8. Use the forever loop to place your code.
9. Think through the steps in the track (forward turn right) forever loop.
10. What is the conditional statement? If the Cutebot senses the track with both eyes then it's going straight and both wheels are on the same speed. If the Cutebot only senses the track with 1 eye (track turns) then 1 wheel is slower. Complete for each wheel and each eye.
11. How do you make it turn right? Should 1 wheel go slower? Which one?
12. Test different wheel speeds and record which turn best find the fastest speed
13. Test your code to see if it works. If not, find your bug and fix it.

ACTIVITY

Steps of the lesson:

1. Break students into pairs.
2. Students will review conditional code: What is it? How did we use it in our code previously? Students can think in pairs and answer these questions. Choose a few students to share their answer. (**5 minutes**). Conditional code allows programmers to make the program interactive. If this condition is met, then a certain program will run.
3. Review the vocabulary with the students.
4. Explain the challenge and the learning objectives of the challenge. I can program the Cutebot to follow a track using a conditional code. I can program the Cutebot to recognize an object using sonar sensors and to stop when sensing the object.
5. Introduce the code bank to the students explaining each block of code and what it does. (**3 minutes**)

6. As a class, use the Steps for Success to build the code and test it straight then test it changing the one wheel to go slower to turn. It should take a few tries for students to get a baseline for the code and wheel speeds. (**15 minutes**)
7. Allow students in pairs to copy the code created by the class and to try the code on their tracks. Time each team's final code and the fastest time wins. **(20 minutes**)
8. Debrief with students. What wheel speed worked best? Why do you think slowing down one wheel makes the Cutebot turn? What is a sensor? How did the sonar sensor sense the object? (**5 minutes**)

ASSESSMENT

Each team should have the Cutebot programmed to avoid collusions and follow the track conducting various turns. They should explain their code: Why does the program run forever? What are the conditions to avoid collusion, and the 3 conditions used to track the line?

REFLECTION/MODIFICATIONS

1. Pair a student you know has issues with reading or letter recognition with students who are strong readers / strong coders with struggling coders.
2. Each team should get a handout that includes the steps for success, the code bank and the completed working code. Don't let them use it until the class creates the code on the board.
3. If you have older students or advanced learners, let them share the code in their groups instead of creating the code as a whole group. Then, the teacher should still test a few programs to show them how it works and determine the baseline code for the challenge.

IoT4Ag
The Internet of Things for Precision Agriculture
an NSF Engineering Research Center

This lesson plan is a product of the Internet of Things for Precision Agriculture, a National Science Foundation-funded Engineering Research Center (NSF Award Number EEC-1941529). Reach out to us at IoT4Ag@seas.upenn.edu and visit IoT4Ag.us for more information.