

Hands on Computer Science Activity 3

Topic: Programming

Learning Objective: Learn how to program a robot to be autonomous. Students will learn how to use the KISS Institute for Practical Robotics (KIPR) software suite and robotics kit in order to write code that will allow a robot to move autonomously using the motors and sensors built on it. Students will have a moving robot by the end of the lesson, and be able to relate it to real-world applications where robots are used.

Alignment with NGSS Grades 3-5

Science and Engineering Practices

Constructing Explanations and Design Solutions

- Apply scientific ideas to solve design problems.
- Generate and compare multiple solutions to a problem based on how well they meet the criteria of the design solution.

Performance Expectations and Disciplinary Core Ideas for Engineering Design

- 3-5-ETS-1-1 Define a simple design problem reflecting a need or a want that includes specified criteria for success and constraints on materials, time, or cost.

Materials:

- Projector/TV (For demonstration purposes)
- Laptop with Google Chrome browser
- KIPR Robotics Kit
 - Wallaby Controller Set & Charger
 - Demobot
- Micro-USB Cable
- Paper and Pens/Pencils

Detailed Description

- Activity
 - Get settled in the lab, review rules of the classroom and activity for the week.
 - Things to note for my class:
 - We will be typing A LOT, practice at home if needed!
 - Activities require thinking. Take your time, try not to get frustrated!
 - Discuss the topic of robots
 - Where do we see them in our everyday life?

- How are they useful? How can they help us make things easier or more convenient? How does it entertain us?
 - Show examples of Honda ASIMO, Disney animatronics, Amazon kimo robots, iRobot roomba, Google Home, etc.
 - Question to think about:
 - How are these robots being controlled?
 - What is one piece of pseudocode for how the robots move?
- Introduce the Demobot & KISS IDE (the software suite)
 - Wallaby Controller
 - Micro-USB Port
 - Charging Port
 - Battery Connection
 - USB Port
 - Motor, Servo, & Sensor Ports (Digital & Analog)
 - How to run a compiled program
 - Motors on the robot
 - Have students plug in the motors to the correct port and micro-USB cable to the controller and laptop
 - Open Google Chrome and click on the bookmark named “KISS IDE”
 - Features:
 - Wallaby is connected/disconnected
 - “KISS IDE” button -where we will write code for the robot(s)
 - Create new projects/work on existing ones
 - Save file
 - Compile file
 - Run file
 - Where to write all our code
 - Different colors and their meaning
- What is a programming language, pseudocode, and compiling?
 - Programming language - code that humans can read and understand, there are many programming languages out there to use (C, C++, Python, Java, etc)
 - Compiling - translation of human-readable code into machine language (code that a computer can read and execute)
 - Pseudocode - simple actions written in the human language that will be rewritten in a programming language to command the actions of the robot

- Run through a couple examples of pseudocode and real code on the board
 - Some examples:
 - Robot move forward one step → `forward(1);`
 - Robot turn right → `turnRight();`
 - Robot turn on → `turnOn();`
- Introduce the basics of C programming language and code available to use
 - We will use this equation a lot:
 - action + left parenthesis + argument(s) + right parenthesis + semicolon
 - Syntax - spelling and grammar of a programming language
 - Debugging - action of identifying and removing errors from code
- Students will write pseudocode for having a robot go around in a square.
- Demonstrate how to translate pseudocode to code using the KISS IDE
 - How many milliseconds are in a second? (Answer: 1000)
 - move forward for 5 seconds → `motor(0,70);motor (3,70);msleep(5000);`
 - turn right for 1 seconds → `motor(0,0);motor(3,100);msleep(1000);`
- Have students translate the rest of the pseudocode to code to control the robot
 - Emphasize debugging and trial and error
 - Remember to:
 - Click on “Compile” and run program on the Wallaby
 - Wallaby’s battery life is limited, remember to charge
- Robot Demonstrations
- Clean up! Save projects and properly shut down Wallaby
 - Pseudocode, why is it important to do that first before writing the actual code?
 - Why is syntax on code important? Debugging?
 - Does learning the basics now provide a better understanding of writing the code that will make the robot run the way we want it to?
 - What are the next steps after today?
 - From what we learned, describe how you now know how companies control the robots as we have seen at the beginning of class?
 - Push the chairs back to where they belong.

How will you conclude the lesson to enforce the learning objective:

I will ask students what they accomplished today and through what actions it took that lead up to it. I will ask a series of reflection questions (listed above) to allow students time to

understand the importance of syntax and debugging while writing code, in addition to relating their experience to real world applications.

What science process skills will this lesson exercise?

Logical thinking, problem solving, observation, prediction, experimenting

Safety precautions

None