Grades 3–5, 6–8

15–30 minutes

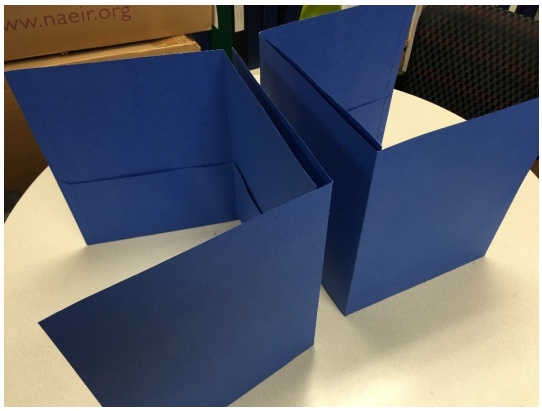
MY ROBOT FRIEND

DESIGN CHALLENGE

Program a “humanoid robot” to build a structure.

SUPPLIES AND EQUIPMENT

- ☐ Building blocks (e.g., LEGO or Duplo blocks, wooden blocks, Lincoln Logs, disposable cups; select materials with different colors and shapes)
- ☐ Flat surface or table
- ☐ Barrier wall (cardboard, stacked books, folders)
- ☐ Stopwatch (optional)



Wall barrier setup using folders.

GETTING READY

Set up a programming station for each pair of participants. Each station should have two sides with at least a 1' x 1' area on each side for building a structure. Separate the two building areas with a barrier wall, or partition, that blocks the view between the participants at the station.

Each side needs to have the exact same selection of blocks.

INTRODUCTION

- **What is a robot?** (You will probably get a wide variety of answers—this is good!) Let participants know that if we asked a bunch of robotics experts what a robot is, we would get a wide variety of answers as well. Because of the complexity and variety of robots, it is nearly impossible to agree on one standard definition of what a robot is. Instead, we can talk about how a robot works.
- **How does a robot work?** A robot is a machine that performs three basic functions: sensing, thinking, and acting.
  - **Sensing:** Robots use sensors to gather information about their surroundings. Sensors are devices that detect some type of input from the physical environment, including light, heat, pressure, motion, and sound. A robot’s sensors act just like our human senses do, giving us information about our surroundings.

MacGillivray Freeman’s DREAM BIG

INTRODUCTION (CONTINUED)

- **Thinking:** Robots have a brain called a microcontroller. Similar to our human brain, a robot's brain gathers information from its senses to make decisions on how to move or act. The robot's brain contains a set of instructions developed by a human programmer; these instructions tell the robot how to move and when to move.
- **Acting:** Acting is the fun part of the robotic process. This is where the robot comes to life to move the way it was designed to move. It might navigate around obstacles, pick up an object, or build a car.

■ **How are robots programmed?** (Again, you may have a variety of answers. The goal of this question is to help participants understand that engineers are the people who breathe life into robots through programming.) Let the whole group know that engineers control a robot by creating a set of instructions (a program) for the robot. The robot then follows the instructions so that it knows how to act or react when it encounters changes in its environment. A well-designed program can make a robot appear to be smart and to work on its own; this is all thanks to the engineer who programmed the robot's instructions.

INSTRUCTIONS

1. **Designate teams and team roles:** Divide the participants into teams of two. Have each team choose who will be the programmer and who will be the robot. Each team needs identical materials. Materials should contain blocks of different shapes and colors.
2. **Introduce the design challenge:** The challenge for each team is to program their humanoid robot to build a structure.
3. **Directions for the programmer:** Remember, a program is a set of instructions created by an engineer that tells the robot what to do. Build a structure using the blocks in front of you. At the same time, give your robot instructions on how to build the same structure. Since the robot cannot ask you any questions, your instructions need to be clear and simple so that the robot understands exactly what it needs to do.
4. **Directions for the robot:** Follow your programmer's instructions the best you can. As the robot, you can't talk back to the programmer to ask any questions about the instruction. Just follow the instructions as you understand them.
5. **Begin building:** As the teams are building, be sure to monitor each station to ensure that the robot is not asking for clarification or peeking to see the programmer's structure.
6. **The unveil:** After each pair finishes building, have each team unveil and compare their two structures.
7. **Ask questions:** Was the program clear? How could you improve the instructions? Can you figure out when the structures started to look different? What details should programmers have included?

ACTIVITY VARIATIONS

Google self-driving car. Credit: Google.

Competition: Have a group of participants judge the structures and pick a winning team for the activity. Judges choose a winner based on which team's structures are most alike.

Teamwork: Add a second programmer to each team (making teams of three). The two programmers will have to work together to accomplish the task.

Trade roles: Switch the programmer and robot roles for another round of building so that each participant experiences being a programmer and a robot.

Programming language: Give teams a few minutes to create their own robot vocabulary before programming. They could give blocks different names based on size or color and discuss how they can best describe how and where to place the blocks. This common language between the programmer and robot will make it easier for the robot to understand the instructions.

Have the programmer write out the instructions. The robot can only use the written instructions.

TROUBLESHOOTING

- The structures may become unbalanced and fall during the building. If the structure falls during the building, have teams restart their build from the beginning.
- During the build, the robot may lose track of the commands and the order they should be in. To avoid this, allow the robot to confirm that it has received and executed the command using a visual cue.

RELEVANT TERMINOLOGY

Humanoid robot: A robot with its body shape built to resemble that of the human body.

Program: A set of instructions a robot follows to perform a specific task.

Programmer: A person who creates and tests programs for devices including robots.

Programming: Creating a plan or schedule of activities and procedures to be followed.

Sensor: A device that detects some type of input from a physical environment. This input could be light, heat, pressure, motion, or sound. Sensors are used in robotics to help robots gather information about their surroundings.

GUIDANCE FOR YOUNGER CHILDREN

QUESTIONS TO ASK AFTER THE ACTIVITY

- Did your structures look the same? Why or why not?
- How could you improve your program to make it more accurate?
- Were there programs that worked better than other programs? Why?
- Was it harder to be the programmer or the robot? Why?
- How well did your team work together? How could you have improved your teamwork?

ENGINEERING CONNECTIONS

Robots are amazing machines that are designed, built, and programmed by engineers. Engineers often make the task of bringing robots to life look easy, but it is more difficult than it looks. This is especially true when it comes to programming a robot. It is easy to make a mistake when writing programs for robots. All it takes is one single command to be misspelled or placed in the wrong order, and the robot won't understand what it needs to do. To work around this issue, engineers test their programs many times. They do it in a safe environment that won't harm the robot, just to find mistakes. If a bad instruction is found, the engineers fix the problem quickly and then test the program again. They do this until the program works correctly.

SCIENCE CONNECTIONS

Programming is a key element of computer science. So what is computer science? Computer science is simply the study of computer technology. Like engineers with robots, computer scientists use programming to create a plan that the computer follows. These plans allow computers to do the amazing things we take for granted every day.

Writing a program for a robot is similar to creating a recipe for a cake. The chef who creates the cake recipe needs to clearly write out the instructions so that there is no confusion. If any instruction is out of order or confusing to the baker, or asks for the wrong ingredients, the cake won't taste very good. Computer programmers need to think carefully as they are creating their plans for a robot or a computer. If any command is out of place, it could spell disaster: the program or robot won't work.

GUIDANCE FOR OLDER YOUTH AND ADULTS

QUESTIONS TO ASK AFTER THE ACTIVITY

- What was your program (plan) before starting the challenge? Did your strategy work?
- What was the result of your program: did your structures look the same? Why or why not?
- How could you improve your program to make it more accurate?
- Did you learn from any of the other teams' strategies? Did you add any elements of their program into your program?

ENGINEERING CONNECTIONS

Just like engineering, programming at its core is all about problem solving. The task of a programmer involves much more than writing code. A programmer's true task is developing creative solutions to problems through the code they write. One of the biggest challenges that programmers face is developing programs that allow computer programs to learn on their own—essentially creating a computer program that can think like a human. In recent years, programmers have made huge strides tackling this problem because of advancements in computer technology. These advances in computer technology have led to a new type of programming called Machine Learning.

Machine Learning was born out of the idea that computers could learn from the data they were processing and make future decisions based on what they learn. Over the years, Machine Learning has advanced to the point where computers can now analyze large amounts of data very quickly. By learning from patterns that are emerging in the data, computers can make complex decisions very quickly. Advancements in Machine Learning have led to a number of new products and innovations, including Google's self-driving car, Netflix's recommendation system, and Facebook's newsfeed.

SCIENCE CONNECTIONS

In the activity, you probably noticed that the "robot" didn't always understand words in the way you meant them. Humans find ways to work around these miscommunications using complex processing abilities in our brain. To make sure computers can process information quickly and efficiently, computer scientists use coding languages, all of which are built using a system called binary. Binary is a system of logic that communicates using only zeros and ones. Each piece of information can only be one of these two options, which allows programmers to avoid any miscommunication like the kind you may have experienced during the activity. Programmers use the binary system to develop programs made up of many zeros and ones. Since computers use the simplified binary language, they can process information at very high speeds. This allows computers and robots to assist in tasks that would take much longer for humans to complete.

MacGillivray Freeman's DREAM BIG

ENGINEERING OUR WORLD

dreambigfilm.com

ACKNOWLEDGMENTS

Activity adapted from the Saint Louis Science Center.

Supplemental content adapted for *Dream Big* Activities by Carnegie Science Center.

CARNEGIE SCIENCE CENTER
One of the four Carnegie Museums of Pittsburgh

Funding for this *Dream Big* Activity was provided by a generous grant from the United Engineering Foundation.

Find more great activities at DiscoverE.org

DISCOVER
LET'S MAKE A DIFFERENCE

MacGillivray Freeman

ASCE
American Society of Civil Engineers

Bechtel