90 minutes

Grades 6–8, 9–12

Operation PolarEye

DISCOVER e
LET'S MAKE A DIFFERENCE

Polar bear habitats are melting! Practice coding without computers, and then code a drone to map polar bear habitat.

Background

Operation PolarEye had an urgent mission: gather information about polar bears and their habitat and use it to save them from extinction due to climate change. This innovative collaboration was a partnership between engineers at Northrop Grumman and wildlife biologists at San Diego Zoo Global.

Engineers designed a remote-controlled, Unmanned Aerial System (UAS) that can withstand the harsh conditions of the Arctic and carry different kinds of cameras to collect a range of data without disturbing the bears. It can actually recognize polar bears as well as take multispectral images of ice! To learn more, go to https://now.northropgrumman.com/landing/environmental-tech/

Materials

- ☐ Disposable cups (10 per team)
- ☐ Blank paper
- ☐ Pencils or pens
- ☐ Graph paper, 1/2" ruled
- ☐ Small buttons or similar markers (no bigger than 7/16" diameter) to represent a drone

High-res visual
5-band multispectral
Dual radar ice mapping
4K video
Infrared video
Polar bear recognition

Instructions: Part 1

The ingenuity behind Operation PolarEye has computer coding at its core. Coding, or programming, is the set of instructions we give computers so that they know what we want them to do. Part 1 of this activity demystifies coding for students by having them write their own “code” for a human “robot,” rather than a computer, to follow. In Part 2, they build on this knowledge by coding instructions for a hypothetical drone to map polar bear territory.

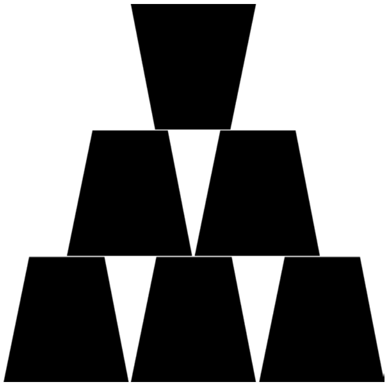
Using a predefined “robot programming vocabulary” (see below), students write a program that instructs the student “robot” to build a pyramid out of plastic cups. To write this code, they connect symbols with actions, just like engineers who build code for a UAS, drone, or any computer-based technology. Students also debug code as part of the process.

1 For preparation, write the programming symbol key and draw the 6-cup pyramid on the board (see diagrams below).

2 Ask students how robots know what actions to take. Explain that they are programmed to follow specific step-by-step instructions and are programmed just like computers.

3 Tell students they will write code, using only the 6 arrow symbols on the board, that a human “robot” will use to build a cup pyramid. Their goal is to design a program that gets the 6-cup pyramid built in as few steps as possible.

↑	— Pick Up Cup
↓	— Put Down Cup
→	— Move 1/2 Cup Width Right
←	— Move 1/2 Cup Width Left
↻	— Rotate Cup 90˚ Clockwise
↺	— Rotate Cup 90˚ Counterclockwise



4 Begin with a simple example: Stack 2 cups and hold your hand in the air above the stack. Explain that you need them to write a program to tell a robot to get the cups next to each other on the table. Ask the class for the first instruction. The correct answer is “pick up a cup.” (Note: For this activity, “pick up a cup” means to lift the cup above the highest cup already in the stack. Also, a robot may only work with each cup once in a program—there’s no way to instruct the robot to return to a cup that’s already been placed.) With your hand still in the air, ask for the next move. You may have to remind the class that one step forward is only half the width of a cup. Once you’ve placed the cup, ask the class to help you write the symbols on the board to create a program for a robot to follow properly.

5 Split the class into teams of 3 or 4. Each team should choose one robot. Send the robots to the “robot waiting zone” in another part of the classroom while the “programmers” start coding. Robots can use their time to make sure they understand the rules. The programmers should work together to figure out the moves the robot needs to follow to create a 6-cup pyramid. The idea is to make the instructions as efficient as possible by using the fewest steps. Once they have the steps figured out, the team should translate the steps into a written program using the arrow symbols. They should test their program to make sure it works as they expect, and then fix the bugs as needed.

6 Next the group will get the robot back to run the program. The programmers are not allowed to tell the robot what to do at all—the robot must exactly follow the instructions in the program. If the group notices a mistake, they are allowed to halt the program, send the robot back to the waiting zone, and fix the error before bringing the robot back to complete the challenge.

7 Once all the groups are done, have them compare their programs to see how many steps each one has.

8 Now have each group pick a new robot to go to the robot waiting area. Have the programmers draw a complex pyramid with 10 cups. Then have them write a program that will instruct the robot to build this pyramid. The object is to see how accurately the robot builds the pyramid just from the program without seeing the drawing.

Guiding Questions ?

How can you eliminate unneeded steps from your program?

How will you debug your program before providing it to your robot?

Why is it critical for a computer program to have absolutely no mistakes?

Instructions: Part 2

The second half of this activity gets students a little closer to using their coding ability to solve a pressing real-world problem. Instead of writing code for a robot to build a cup pyramid, they write code to move a drone systematically over polar bear habitat, creating a map of the terrain.

1. Generate interest and set context by showing students the short video about the scientists and engineers who went to Churchill, Manitoba, to test and gather initial data for Operation PolarEye at http://now.northropgrumman.com/landing/environmental-tech; click the video titled Episode 1: The Challenge.

2. Distribute several sheets of graph paper to each team. Explain that in this activity, students will code a hypothetical drone to map the icy habitat of polar bears. They will draw on their experience with coding a robot to build a pyramid to complete this new challenge.

3. Draw these symbols on the board, along with the actions they represent:

↑	– 1 Step Up	⇑	– 5 Steps Up
↓	– 1 Step Down	⇓	– 5 Steps Down
→	– 1 Step Right	⇒	– 5 Steps Right
←	– 1 Step Left	⇐	– 5 Steps Left

Guiding Questions ?

Is it more efficient to scan up and down, or across?

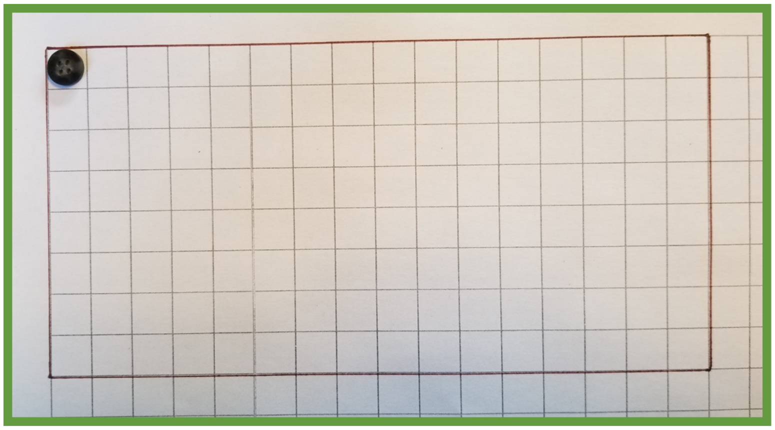
If you could scan diagonally, would that allow fewer steps?

Which is easier, coding the pyramid-making robot or the drone?

4 Tell teams to outline a rectangle on their graph paper that is 8 units by 16 units. Tell them that this area represents the polar bear habitat that their drone must map.

5 Announce the challenge: code the drone to map every square of the habitat, starting from the upper left square and returning to the same square. The drone can move horizontally and vertically, but not diagonally. It cannot backtrack. It can move one square at a time or 5 squares at a time. Try and program your drone to use the smallest number of steps possible.

6 Distribute one marker to each team that can be placed on the graph paper to help students keep their place. Tell students the marker represents the drone and to place it in one corner of their "habitat." Show them the coding symbols on the board, which they can use to code one step or five steps in each direction.



7 Give teams time to write their code. They can sketch their code on the graph paper in pencil, and then test it out on a blank habitat, using their marker. This way they can catch bugs in the code.

8 Once all the teams are done, see how many steps each one took and what directions were most efficient.

This activity was adapted from the Traveling Circuits lesson "My Robotic Friends," developed by Thinkersmith and published under the Creative Commons CC-BY-NC-SA license.