

Nosy Robot

Summary

Description

Turn Dash into a pesky little, nosy robot who always wants to know what's going on.

Learning Procedure

In the Nosy Robot, students will program Dash's nosy expressions and actions over a series of **5 challenges**. Dash will repeat a series of actions (a wink as a customized eye pattern - open eye lights, close eye lights, open eye lights and a sound), when a conditional is met in the **When** block. By nesting a set of repeated actions in a **function** bracket, a **Call** block will activate this **function** when the **conditional** is met, i.e., *Object Behind, Hear Voice and Obstacle in Front*. The main objective of this puzzle is for students to understand that a **function** incorporates a set of instructions or actions that can be called on over and over again.

Concepts Covered

- **Connecting blocks** - students will learn how to connect **When** and **Call** blocks to activate a **function**, a set of instructions.
- **Function**
 - students will learn how to program a set of actions/instructions as a **function** that can be **called** on to be repeated over and over again.
 - students will learn that functions can be named, i.e., "Nosy"
- **Call blocks** - students will learn to program **When** blocks with variables as **input**, that when met, set off **Call** blocks which activate the **Function** or set of instructions, i.e., winking, sound.
- **Variable** - students will learn that information in a command block that can be modified or changed, i.e., the conditional in a **When** block.
- **Other command blocks used in this puzzle:** Start, Eye pattern, Turn to Voice, Look Towards Voice

In App

Vocabulary

Functions: A type of procedure or routine

Call block: allows a set of functions to run

When block: the When command executes when a specified event occurs

Variables: a value that can change

Reference Questions?

1. What types of everyday activities are composed of a simple set of actions that are repeated over and over again? Examples: pedaling a bike, swimming, jumping rope, etc.
2. What is the name of the functions in this program? Why is it important to name a function?
3. What happens to the Call blocks when you name a function?
4. In this program, what conditions activated the function?
5. How are the When conditions for Object Behind and Hear Voice programmed differently than Obstacle in Front? Why do you think this is so? (Suggested Response: There is no movement from the Drive bank following the When condition of Obstacle in Front.)

Activity Extensions

1. Annoying Robot

In this activity, the objective is to program the *most annoying robot!* Dash is a pesky little robot who always wants to know what's going on. Students will collaborate in small groups to program a series of instructions as a **Function** Dash will repeat when called on. Students must program random variables in the **When** block to **Call** the **Function**. Tell students that their function must include 3 actions from the Drive bank and one Sound from the Sound bank (words or sound).

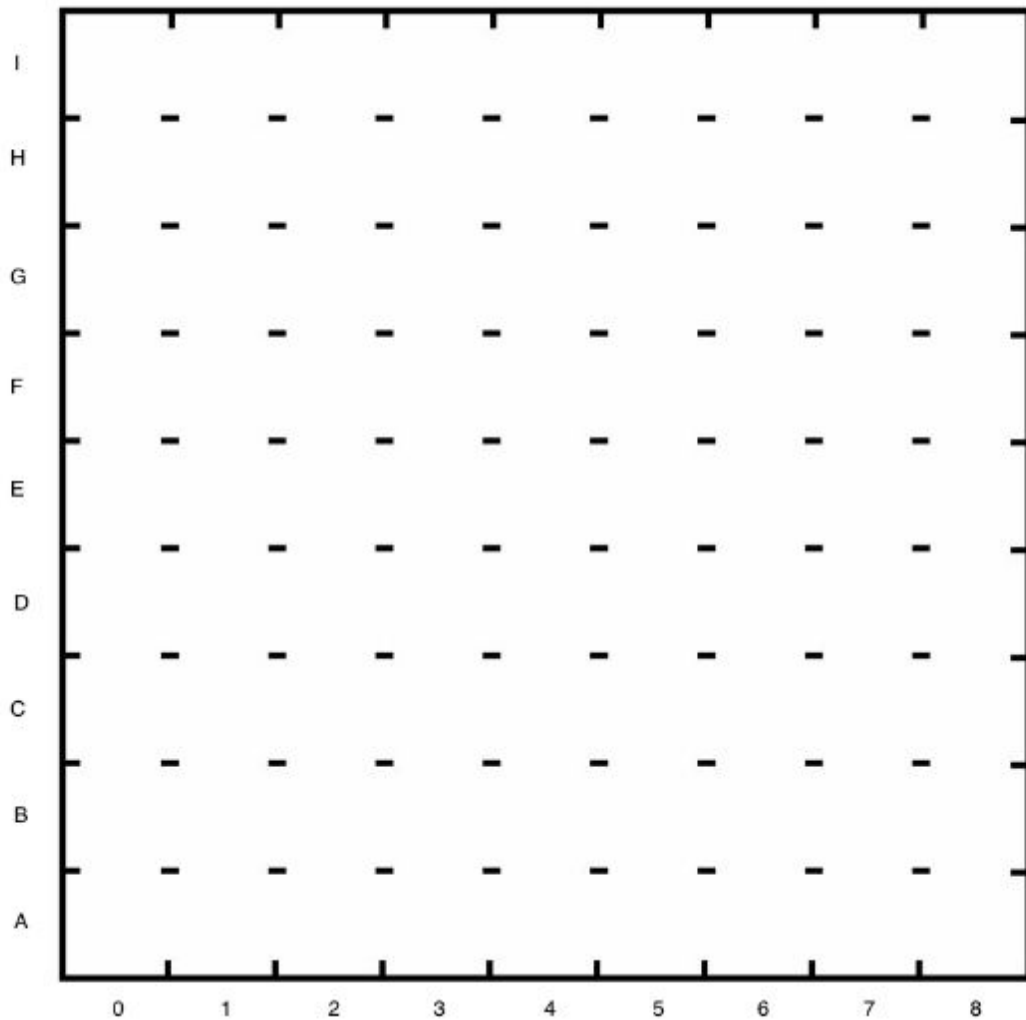
Call groups up one at a time to have Dash perform their group's annoying function for the class. Have classmates vote on the most annoying function. Example: Dash turns in right and left arcs and then sings a line from an annoying song like, John Jacob Jingleheimer Schmidt, or repeats an annoying phrase, like, "I know you are but what am I?" or makes a repeated siren sound.

Standards: CCSS.ELA-LITERACY.SL.1.1.B; CCSS.ELA-LITERACY.SL.2.1.B; CCSS.ELA-LITERACY.SL.3.1.C; CCSS.ELA-LITERACY.SL.4.1.C; CCSS.ELA-LITERACY.SL.5.1.C

2. Dash Nose Grids

The object of this activity is to create a **function** which will repeat, beginning with Dash inside any square in row A and ending with Dash in any square in row I. Create a 9 x 9 grid similar to the one used in the *Wonder League Robotics* competition using masking tape and a large space. Instruct students to program a **function** that includes three moves and a sound. In the first move, Dash will drive forward one square. In the second move, Dash will turn (left or right) one square over. In the third move, Dash will drive forward straight down one square. This function must be repeated with a **When** (setting the conditional) and **Call** block. Students will create this program in the "Create New" section of the Blockly App. *Fourth grade math students incorporate Algebraic Thinking, specifically patterns in functions. Fifth grade math students incorporate Geometry, plotting on "X" and "Y" axis.

Standards: CCSS.MATH.CONTENT.4.OA.C.5; CCSS.MATH.CONTENT.5.G.A.1



3. Dash Nose Patterns

In this activity, students will create hands-on **Functions** with patterns and stamps as an analogy, which will allow students to connect the word “function” with a concrete, visual meaning. First, ask students to choose 3 shapes, e.g., circle, square, triangle. Then have students create a pattern by hand, drawing their selected three shapes over and over again, at least 10 times. Now ask students to think of a shorter way to repeat the pattern. Using a concrete model of a function, students will construct a pattern stamp made from foam board and fun foam. These materials can be affordably purchased at the Dollar Tree. Have students cut out their 3 shapes from fun foam and adhere them in the proper pattern sequence to a rectangular shaped cutout of foam board. Using an inkpad, have your students stamp out the same pattern 10 times. Discuss how and when **functions** are helpful in programming. How is this stamp similar to a function? How is it different? This video shows how to create stamps.

<https://www.youtube.com/watch?v=WhXTYMUWzal>

Standards: CCSS.MATH.CONTENT.3.OA.D.9; CCSS.MATH.CONTENT.4.OA.C.5

4. Dash Nose Squares!

In this challenge, Dash will learn how to draw a square. Working together in small groups of 3 or 4, students will draft a series of step-by-step instructions to teach Dash how to draw a square, using the proper angles, lines and measurement in cm. Students will then translate these instructions into a programmable **Function** that Dash can follow over and over again incorporating **When** (programming a conditional) and **Call** blocks in “Create New” section of the Blockly App. Remind students to name their square-drawing function. In order for Dash to draw a square, students will need to create a contraption that enables Dash to hold a marker in an upright position making direct contact with the drawing paper or they may create a drawing apparatus for Dash using Legos, rubber bands and building brick extensions, as in this model.

<https://www.makewonder.com/play/ideas/23>

Standards: CCSS.ELA-LITERACY.L.2.3; CCSS.ELA-LITERACY.L.3.3; CCSS.ELA-LITERACY.L.4.3; CCSS.ELA-LITERACY.L.5.3

Educational Standards

CC Mathematical Practices:

1, 2, 4, 5, 6, 7, 8

CC Language Arts Standards

SL.1.1.C, SL.2.1.C

SL.1.3, SL.2.3

SL.3.1, SL.3.3, L.3.6, SL.4.1, L.4.6, SL.5.1, L.5.6

CSTA K-12 Computer Science Standards

- CT.L1:3-03. Understand how to arrange information into useful order
- CT.L1:6-01. Understand and use the basic steps in algorithmic problem-solving.
- CT.L1:6-02. Develop a simple understanding of an algorithm
- CPP.L1:3-04. Construct a set of statements to be acted out to accomplish a simple task.
- CPP.L1:6-05. Construct a program as a set of step-by-step instructions to be acted out.
- CT.L2-03. Define an algorithm as a sequence of instructions that can be processed by a computer.
- CT.L2-06. Describe and analyze a sequence of instructions being followed.

Next Generation Science Standards NGSS

- 3-5-ETS1-2. Generate and compare multiple possible solutions to a problem based on how well each is likely to meet the criteria and constraints of the problem.

Solutions

Challenge 1

Connect the blocks so that when Dash sees an **Obstacle in Front**, Dash will **wink** and then make a **custom sound**. Record your own voice and make Dash say something nosy like, **Whatcha' doing?** or **What are you talking about?**

When Start

Eye Pattern All on

START ▶

Reset

What's going on?

Connect the blocks so that when Dash sees an **Obstacle in Front** Dash will **wink** and then make a **custom sound**. Record your own voice and make Dash say something nosy, like **Whatcha doing?** or **What are you talking about?**

The image shows a Scratch 'Eye Pattern' block with the 'All on' radio button selected. The eye pattern is a purple square with an orange border and a black center. The inner ring is filled with white segments, representing an open eye. The block is part of a script starting with a 'When Start' block. Below the block is a 'START' button and a 'Reset' button. A text box at the bottom provides instructions for connecting blocks to trigger a 'wink' and a 'custom sound' when an 'Obstacle in Front' is detected.

When Start

Eye Pattern All off

START ▶

Reset

What's going on?

Connect the blocks so that when Dash sees an **Obstacle in Front** Dash will **wink** and then make a **custom sound**. Record your own voice and make Dash say something nosy, like **Whatcha doing?** or **What are you talking about?**

The image shows a Scratch 'Eye Pattern' block with the 'All off' radio button selected. The eye pattern is a purple square with an orange border and a black center. The inner ring is filled with black segments, representing a closed eye. The block is part of a script starting with a 'When Start' block. Below the block is a 'START' button and a 'Reset' button. A text box at the bottom provides instructions for connecting blocks to trigger a 'wink' and a 'custom sound' when an 'Obstacle in Front' is detected.

When Start

Eye Pattern All on

START ▶

Reset

Connect the blocks so that when Dash sees an **Obstacle in Front** Dash will **wink** and then make a **custom sound**. Record your own voice and make Dash say something nosy, like **Whatcha doing?** or **What are you talking about?**

When Start

When Dash Obstacle In Front

Eye Pattern All on

Eye Pattern All off

Eye Pattern All on

My sounds #1

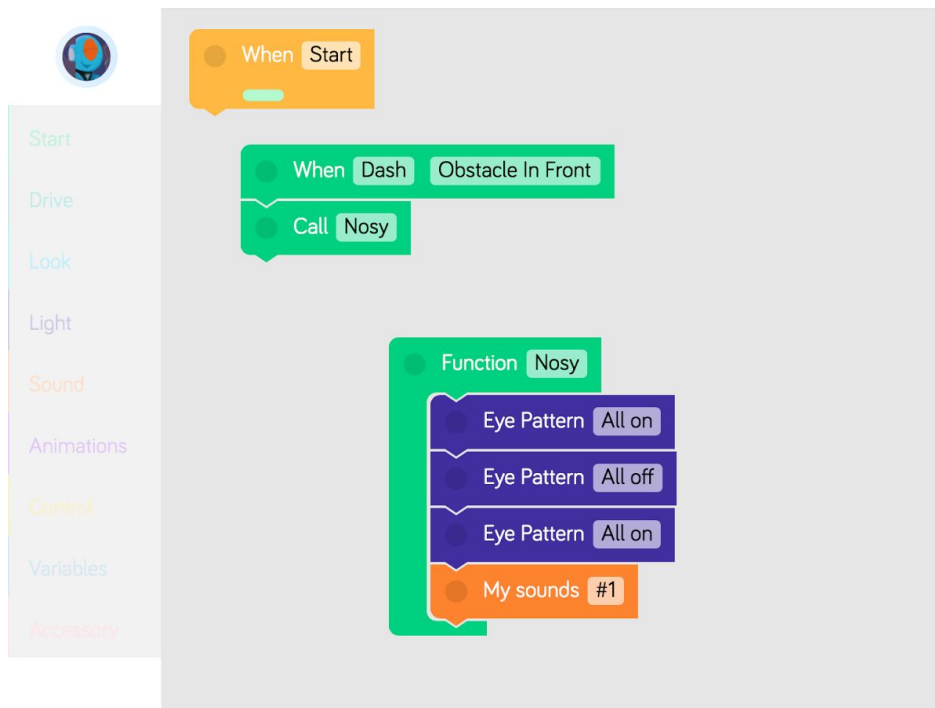
STOP ■

Need a hint?

Connect the blocks so that when Dash sees an **Obstacle in Front** Dash will **wink** and then make a **custom sound**. Record your own voice and make Dash say something nosy, like **Whatcha doing?** or **What are you talking about?**

Challenge 2

Now let's turn the wink and nosy lie into a **function**. Drag the **Eye Pattern** and **My Sounds** blocks into the **Function Nosy** loop. The put the **Call Nosy** block where the other blocks use to be, under **When Dash Obstacle In Front**.



Challenge 3

We want Dash to perform the same **Nosy Function** Dash looks toward voice. Drag a new

The image shows a Scratch script for a character named Dash. The script is organized into three main sections:

- Start:** A 'When Start' block followed by a 'Call Nosy' block.
- Obstacle In Front:** A 'When Dash Obstacle In Front' block followed by a 'Call Nosy' block.
- Hear Voice:** A 'When Dash Hear Voice' block followed by 'Turn to Voice', 'Look towards Voice', and 'Call Nosy' blocks.

A 'Function Nosy' block is defined with the following steps:

- Eye Pattern All on
- Eye Pattern All off
- Eye Pattern All on
- My sounds #1

A hint box at the bottom of the screen reads: "We want Dash to perform the same **Nosy Function** after Dash **looks towards voice**. Drag a new **Call Nosy** block from the menu and add it under **Look towards voice**."

Challenge 4

We've added a new **When** stack for when Dash detects an **Obstacle Behind**. Call the **Nosy function** at the bottom of this new stack.

The image shows a Scratch code editor with the following script:

- When Start** (orange block)
 - Call Nosy** (green block)
- When Dash Hear Voice** (green block)
 - Turn to Voice** (blue block)
 - Look towards Voice** (blue block)
 - Call Nosy** (green block)
- When Dash Obstacle In Front** (green block)
 - Call Nosy** (green block)
- When Dash Object Behind** (green block)
 - Turn Left 180** (blue block)
 - Call Nosy** (green block)
- Function Nosy** (green block)
 - Eye Pattern All on** (purple block)
 - Eye Pattern All off** (purple block)
 - Eye Pattern All on** (purple block)
 - My sounds #1** (orange block)

A hint box at the bottom reads: "We've added a new **When** stack for when Dash detects an **Obstacle Behind**. Call the **Nosy** function at the bottom of this new stack."

Challenge 5

You can name functions whatever you want. Tap the **Function Nosy** block and choose another name for it. You can call it anything you want. Notice that the **Call** blocks change the name automatically to match.

The image shows a programming environment with a sidebar on the left containing categories: Start, Drive, Look, Light, Sound, Animations, Control, Variables, and Accessory. The main workspace contains the following code blocks:

- When Start** (orange block)
- When Dash Object Behind** (green block) containing:
 - Turn Left 180** (blue block)
 - Call Bossy** (green block)
- When Dash Obstacle In Front** (green block) containing:
 - Call Bossy** (green block)
- When Dash Hear Voice** (green block) containing:
 - Turn to Voice** (teal block)
 - Look towards Voice** (blue block)
 - Call Bossy** (green block)
- Function Bossy** (green block) containing:
 - Eye Pattern Custom** (purple block)
 - Eye Pattern Custom** (purple block)
 - Eye Pattern Custom** (purple block)
 - My sounds #1** (orange block)

At the bottom, a text box reads: "You can name functions whatever you want. Tap the **Function Nosy** block and choose another name for it. You can call it anything you want. Notice that the **Call** blocks change the name automatically to match." The text box is flanked by left and right navigation arrows and a progress bar below it.