

Wonder Dash Teacher Guide

Scroll Quest challenges for Dash

Table of Contents

Level 1 Guide

[Wonder Workshop - Hello World](#)

[Wonder Workshop - Celebrations](#)

[Wonder Workshop - Go Bot](#)

[Wonder Workshop - Stroll Patrol](#)

[Wonder Workshop - Clap and Dance](#)

[Wonder Workshop - Press to Express](#)

Level 2 Guide

[Firefly Lagoon - Firefly Watch](#)

[Firefly Lagoon - Water Wiggle](#)

[Firefly Lagoon - Marco Polo](#)

Level 3 Guide

[Dragon Reach - Thrill Ride](#)

Level 4 Guide

[Yeti Pass - Snowman](#)

[Yeti Pass - Crazy Skates](#)

Level 5 Guide

[Mount Ashburn - Volcanic Trek](#)

[Mount Ashburn - Random Volcano](#)

Level 6 Guide

[Galaxy Lake - Spaceship Rescue](#)

[Galaxy Lake - Robot Translator](#)

Level 7 Guide

[Kong City - Doing Donuts](#)

[Kong City - Spinning Dash Game](#)

[Kong City - Butler Dash](#)

Level 8 Guide

[Forgotten Jungle - OMG Poppies!](#)

[Forgotten Jungle - Ancient Robot](#)

Level 9 Guide

[Dry Gulch Desert - Square Dance](#)

[Dry Gulch Desert - Desert Dash](#)

[Dry Gulch Desert - Dune Buggy](#)

Level 10 Guide

[Big Cat Canyon - Itchy Lion](#)

Level 11 Guide

[Castle Creepenstein - Dashlight](#)

[Castle Creepenstein - The Listener](#)

[Castle Creepenstein - Reacher Creature](#)

[Castle Creepenstein - Ogre of Darkness](#)

Level 1: Hello World, Celebrations, Go Bot, Stroll Patrol, Clap and Dance, Press to Express

Concepts covered

- Sequencing commands left to right
- Connecting commands
- Dragging commands to desktop
- Transitions: Auto, Instant
- Animations: Celebration, Take off, Stop, Turn, Dance
- Sounds: Using pre-recorded sounds and phrases from the Sound menu.
- Lights: Programming colored lights, i.e., magenta to turn on and off.
- Move: Programming distance for Dash to move forward or backward in cm.
- Connecting a command back to an earlier command in a loop.

Key terms

Code: One or more sequenced commands programmed on the Wonder app to be carried out by Dash.
Other forms of the word include: coder (noun), coding (verb)

Programming: The act of creating a code or program for Dash to carry out. Other forms of the word programming include: programmer and program (noun).

Command: An instruction you code for Dash to follow.

Cue: Cues tell Dash when to transition to the next command, e.g., auto, instant, clap, press top button, press button 1, 2 or 3.

Auto: Dash will go to the next state after finishing the current action.

Instant: Dash will go to the next state immediately.

Animation: To bring Dash to life with movements, sounds or a combination of both, e.g, celebrate, push, kiss, burp

Sounds: A command that allows you to play a pre-recorded sound to be recalled within a program.

Move: A command that allows you to move Dash forward or backward in cm.

Extension questions

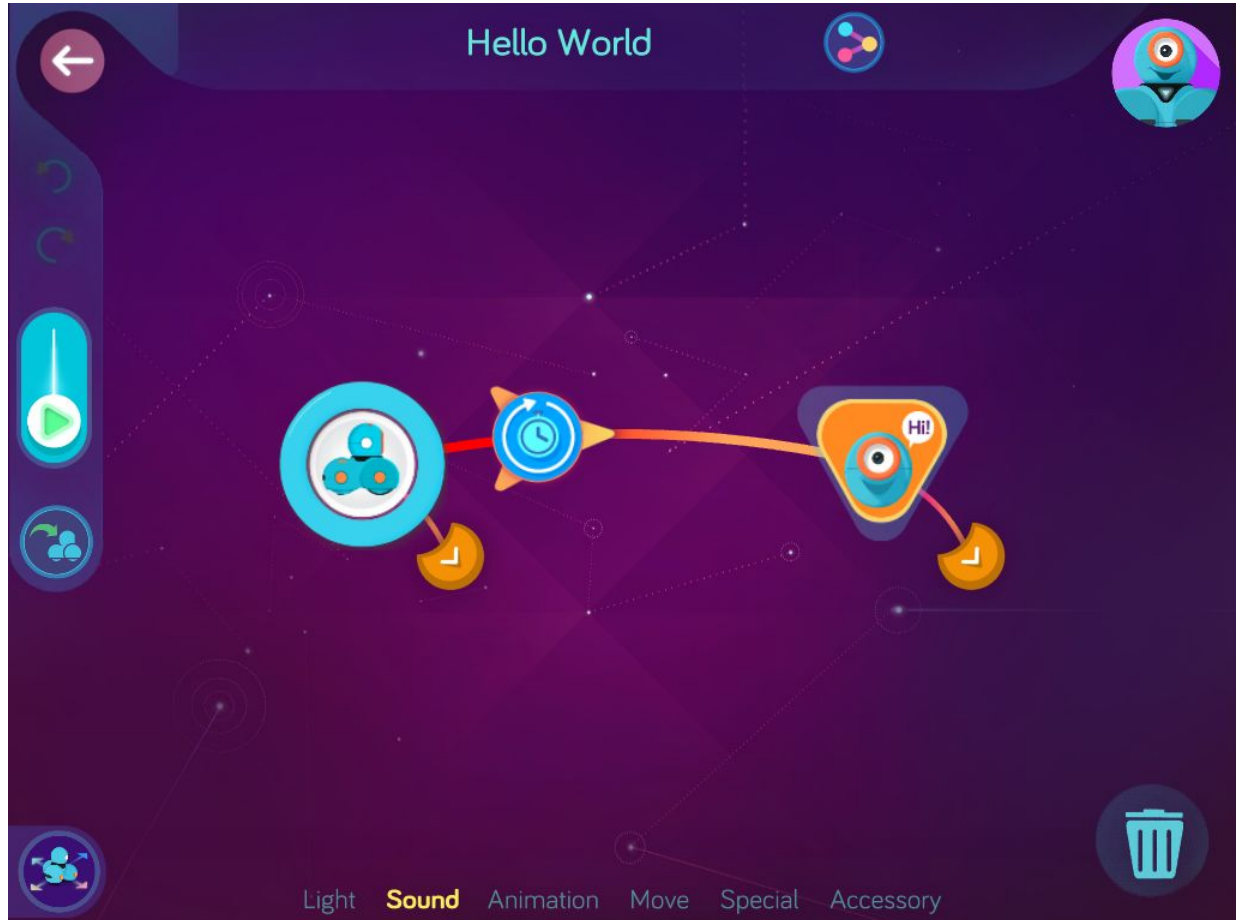
1. What did you learn about coding from this first set of challenges?
2. In what direction is a Wonder app code typically sequenced? What is the exception?
3. What are some different cues that can be used between commands? Explain the purpose of each. (Possible answers: instant, auto, clap, press top button, press buttons 1, 2 or 3)
4. What is the difference between an Animation and a Sound?
5. What are some different ways in which Dash can move? (Response: forward, backward, and turning around.)
6. What is the difference between the function of the Take Off versus the Move Forward command?

Hello World

Zone: Wonder Workshop
Challenge Progression 1

Solution

Make Dash automatically say, "Hi."



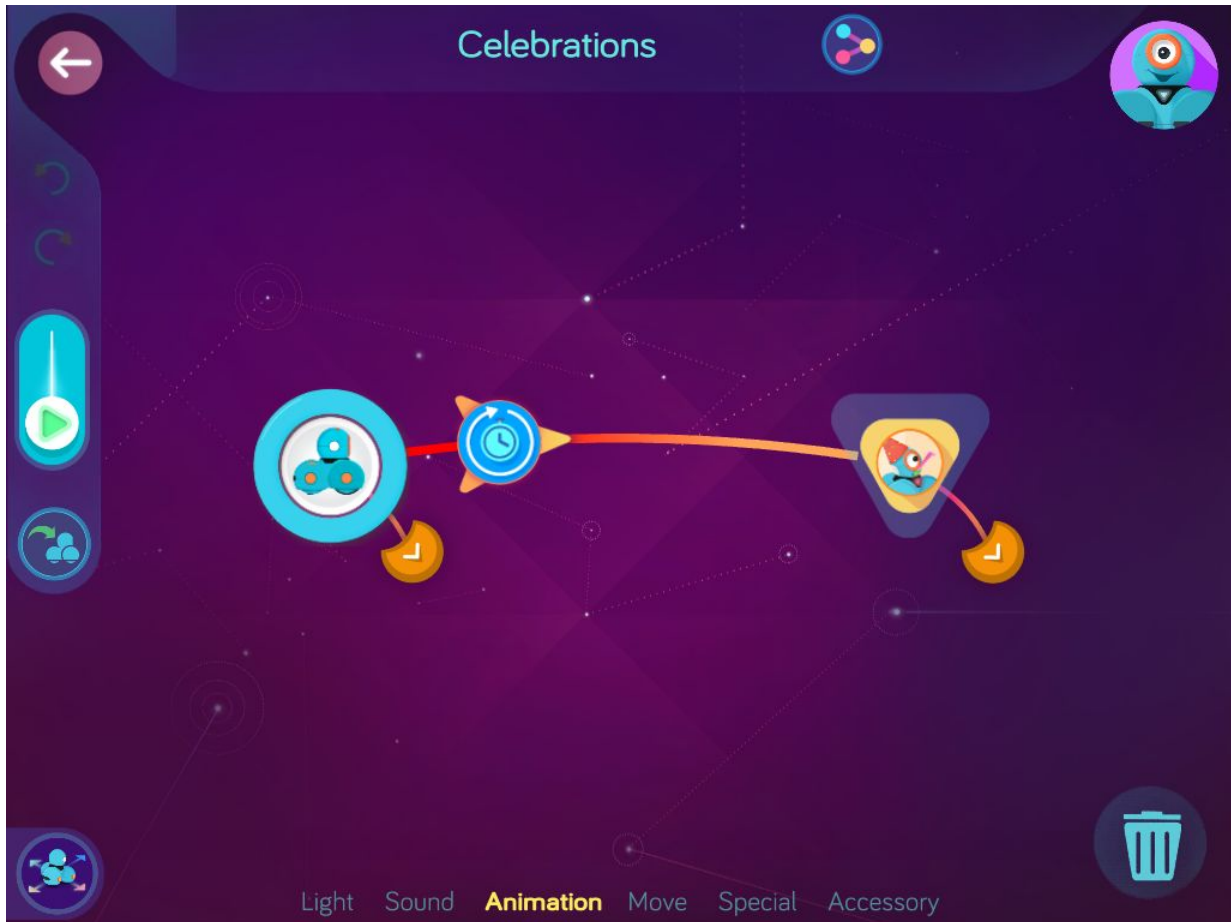
Wonder key: jkqv

Celebrations

Zone: Wonder Workshop
Challenge Progression 2

Solution

Make Dash automatically celebrate.



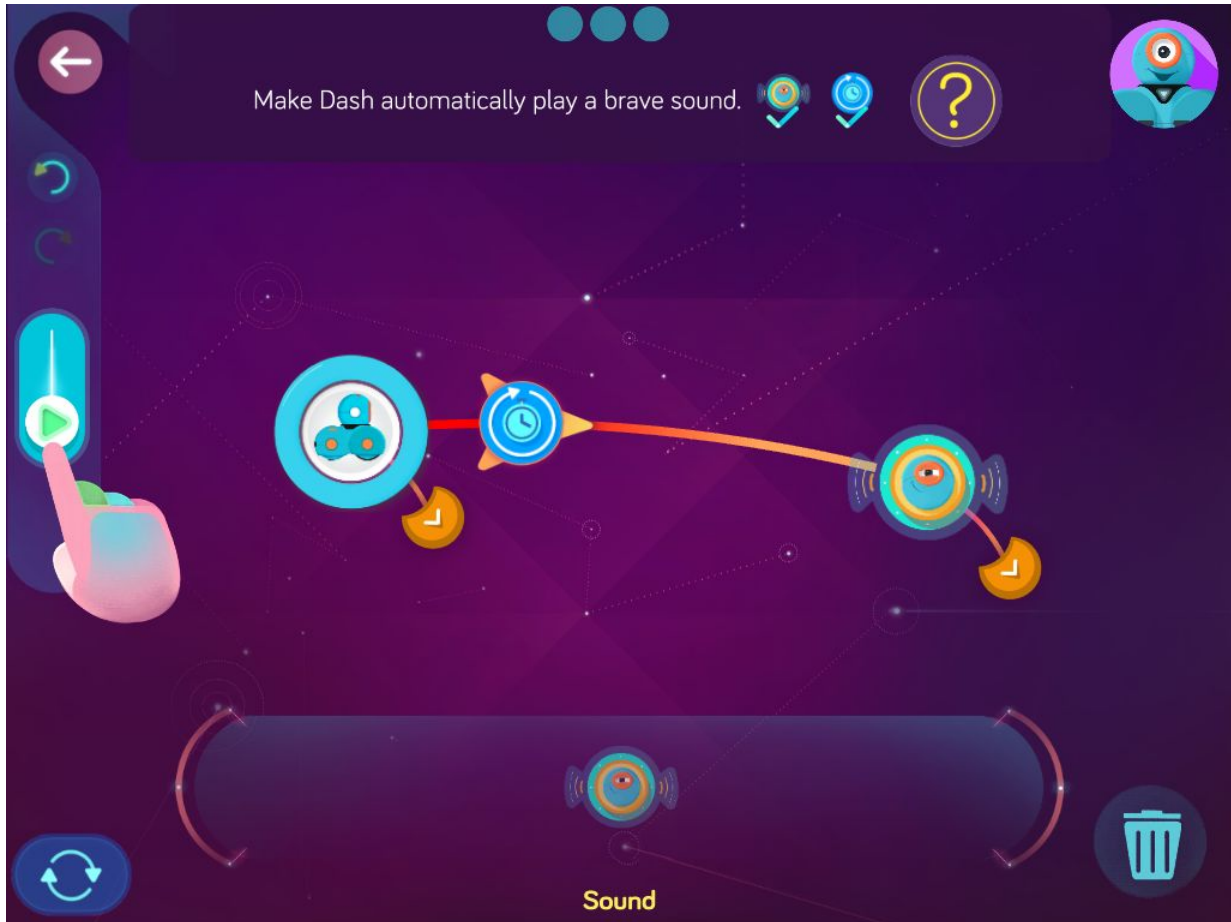
Wonder Key: ysod

Go Bot

Zone: Wonder Workshop
Challenge Progression 3

Solutions

Step 1: Make Dash automatically play a brave sound.



Step 2: After the sound, make Dash automatically take off.

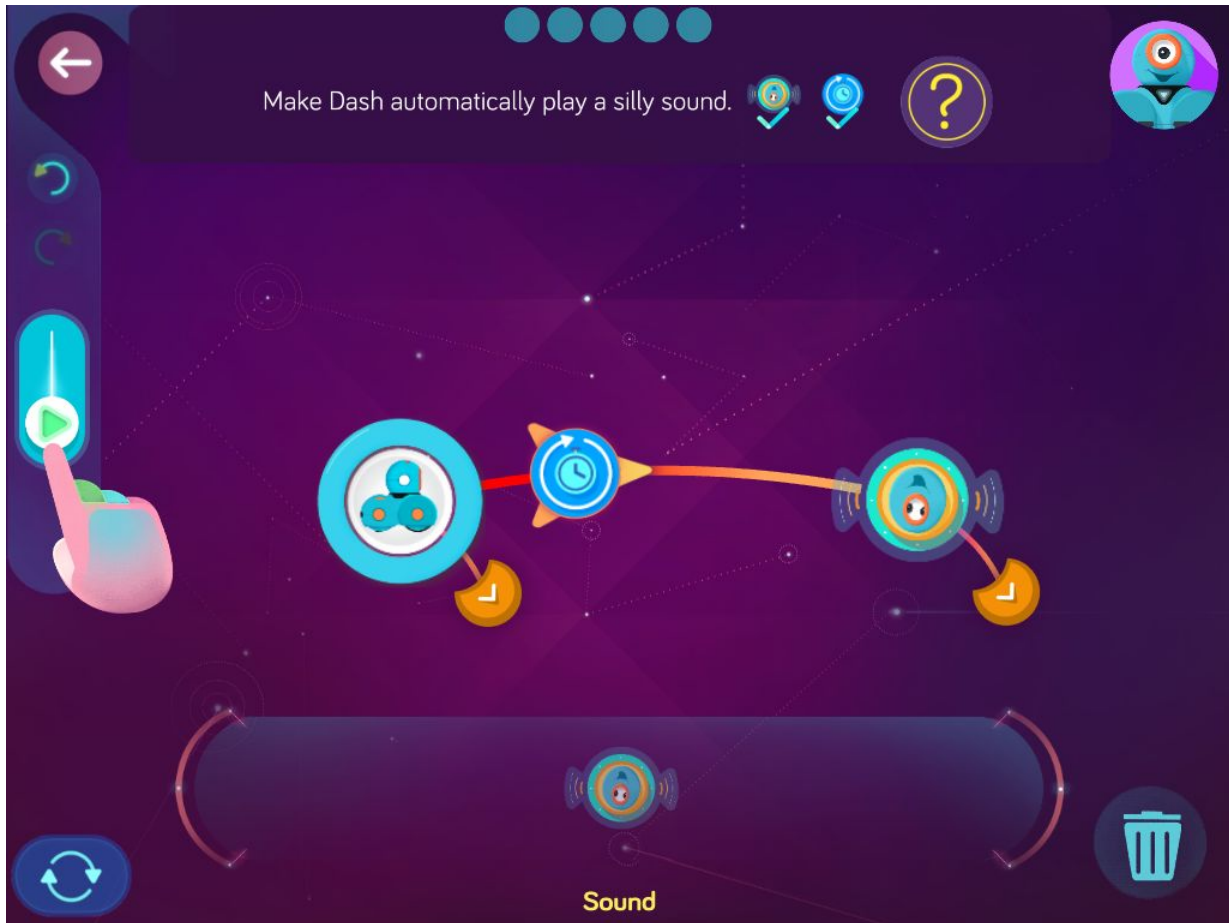


Stroll Patrol

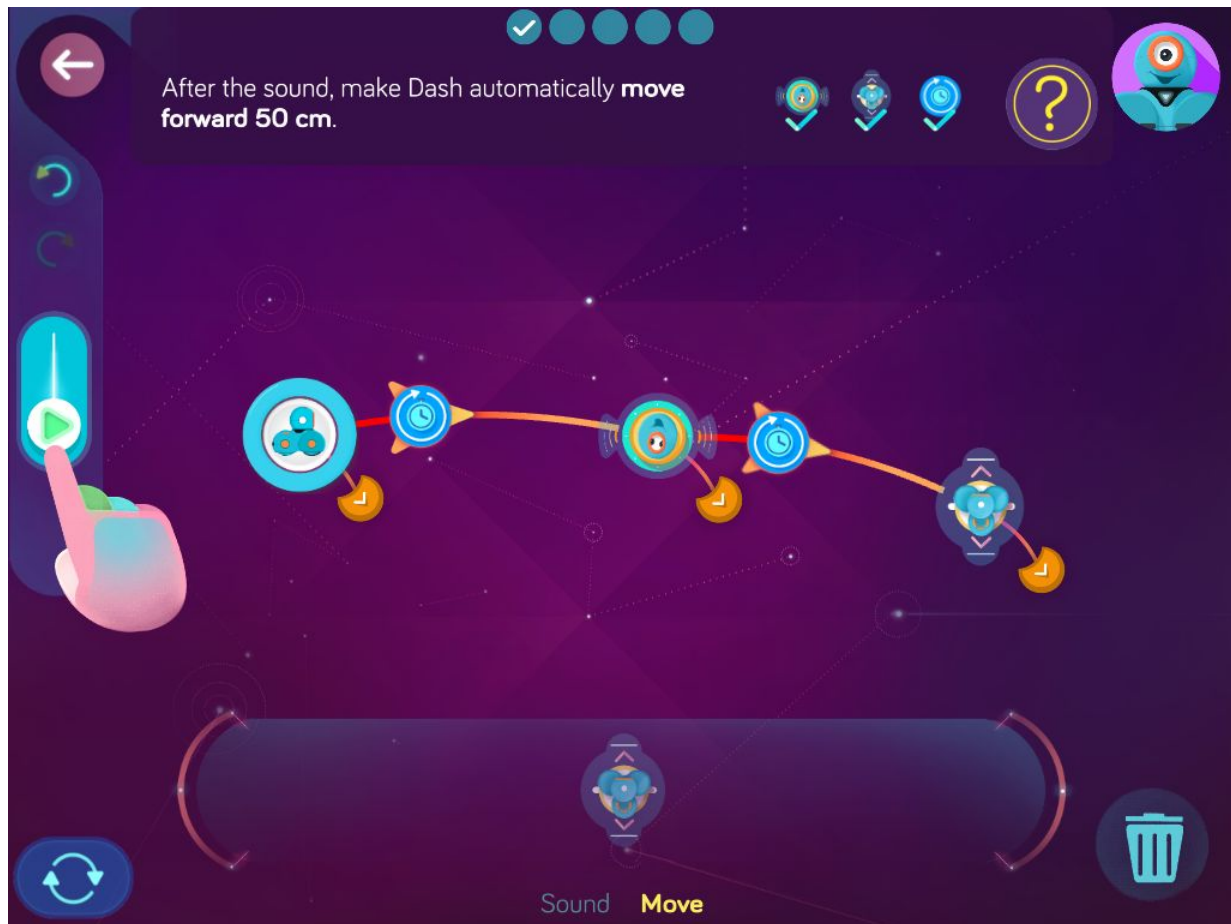
Zone: Wonder Workshop
Challenge Progression 4

Solutions

Step 1: Make Dash automatically play a silly sound.



Step 2: After the sound, make Dash automatically move forward 50 cm. *Click on the move forward command to program distance.



Step 3: After the sound, make Dash instantly move forward.

The image shows a Scratch script editor interface. At the top, there is a navigation bar with a back arrow, a progress indicator (four blue circles, the first two are checked), and a help icon (a question mark). Below the navigation bar, the text reads: "After the sound, make Dash **instantly** move forward." The main workspace contains a script with three blocks: a Sound block, a Move block, and another Sound block. A hand cursor is pointing at the Sound block. The bottom of the interface features a palette with "Sound" and "Move" categories, a trash icon, and a refresh icon.

Step 4: After moving forward, make Dash automatically turn around.

The image shows a programming interface for the Dash robot. At the top, a task instruction reads: "After moving forward, make Dash automatically turn around." The interface includes a top bar with a back arrow, a progress indicator (three checked marks and two unchecked), and a user profile icon. A central workspace contains a sequence of programming blocks connected by arrows: a 'Start' block, a 'Move Forward' block, a 'Turn Right' block, a 'Wait' block (set to 0.00), another 'Turn Right' block, a 'Move Forward' block, and a 'Turn Right' block. A 'Sound' block is also present. A bottom bar shows 'Sound', 'Animation', and 'Move' categories, with 'Animation' selected. A trash icon is located in the bottom right corner.

Step 5: After turning around, make Dash automatically play the silly sound again.



Wonder key: eshu

Clap and Dance

Zone: Wonder Workshop
Challenge Progression: 5

Solutions

Step 1: *Make Dash automatically turn the lights off.*



Step 2: After turning the lights off, make Dash play a silly sound when a clap is heard.

The screenshot shows a programming environment with a dark purple background. At the top, there is a navigation bar with a back arrow, a progress indicator (four circles, the first is checked), and a help icon (a question mark). Below the progress indicator, the text reads: "After turning the lights off, make Dash play a silly sound when a **clap** is heard." The main workspace contains a sequence of four blocks connected by red arrows: 1. A 'When light is off' block with a lightbulb icon. 2. A 'Wait 2 seconds' block with a clock icon. 3. A 'Play a sound' block with a speaker icon and a 'clap' sound icon. 4. A 'Say for 2 seconds' block with a speech bubble icon. A hand cursor is pointing at the 'Play a sound' block. At the bottom, there are 'Light' and 'Sound' labels, a trash can icon, and a refresh icon.

Step 3: After the sound, make Dash instantly dance to the left.

The image shows a programming interface for a character named Dash. At the top, a text box contains the instruction: "After the sound, make Dash **instantly** dance to the left." Above this text are four status indicators: two green checkmarks and two blue circles. To the right of the text are several icons: a lightbulb, a speaker, a play button, a question mark, and a character icon. Below the text is a workspace with a grid background. A sequence of programming blocks is connected by arrows. The blocks include: a 'When green flag clicked' block, a 'Play sound' block, a 'Wait' block, a 'Move Dash to x: -100 y: 0' block, and a 'Say Dash for 2 seconds' block. A hand cursor is pointing at a play button on the left side of the workspace. At the bottom, there is a palette with 'Light', 'Sound', and 'Animation' categories. The 'Animation' category is highlighted in yellow. A trash can icon is also visible in the bottom right corner.

Step 4: After dancing left, make Dash automatically turn the lights off again.



Wonder Key: zupp

Press to Express

Zone: Wonder Workshop
Challenge Progression: 6

Solutions

Step 1: Make Dash automatically turn on magenta lights.



Step 2: After turning on the lights, make Dash laugh when the top button is pressed. Then make Dash automatically turn on magenta lights again.

The image shows a Scratch script editor interface. At the top, a text box contains the instruction: "After turning on lights, make Dash **laugh** when the **top button is pressed**. Then make Dash automatically turn on **magenta lights** again." Above this text are five progress indicators, with the first one checked. To the right of the text are icons for a character, a lightbulb, a clock, and a speech bubble, each with a checkmark. A question mark icon and a character icon are also present.

The main workspace shows a Scratch script for a character named Dash. The script starts with a "When green flag is clicked" event block, followed by a "Play sound" block (a green triangle icon). A hand cursor is pointing at this block. The next block is a "When top button is pressed" event block, followed by a "Laugh" block (a speech bubble icon). The final block is a "Turn on magenta lights" block (a lightbulb icon). The script is connected to a character icon of Dash.

At the bottom of the workspace, there is a "Light" block with a lightbulb icon and an "Animation" block with a play button icon. A trash can icon is also visible in the bottom right corner.

Step 3: After turning on the lights, make Dash give a kiss when button 1 is pressed. Then make Dash automatically turn on magenta lights again.

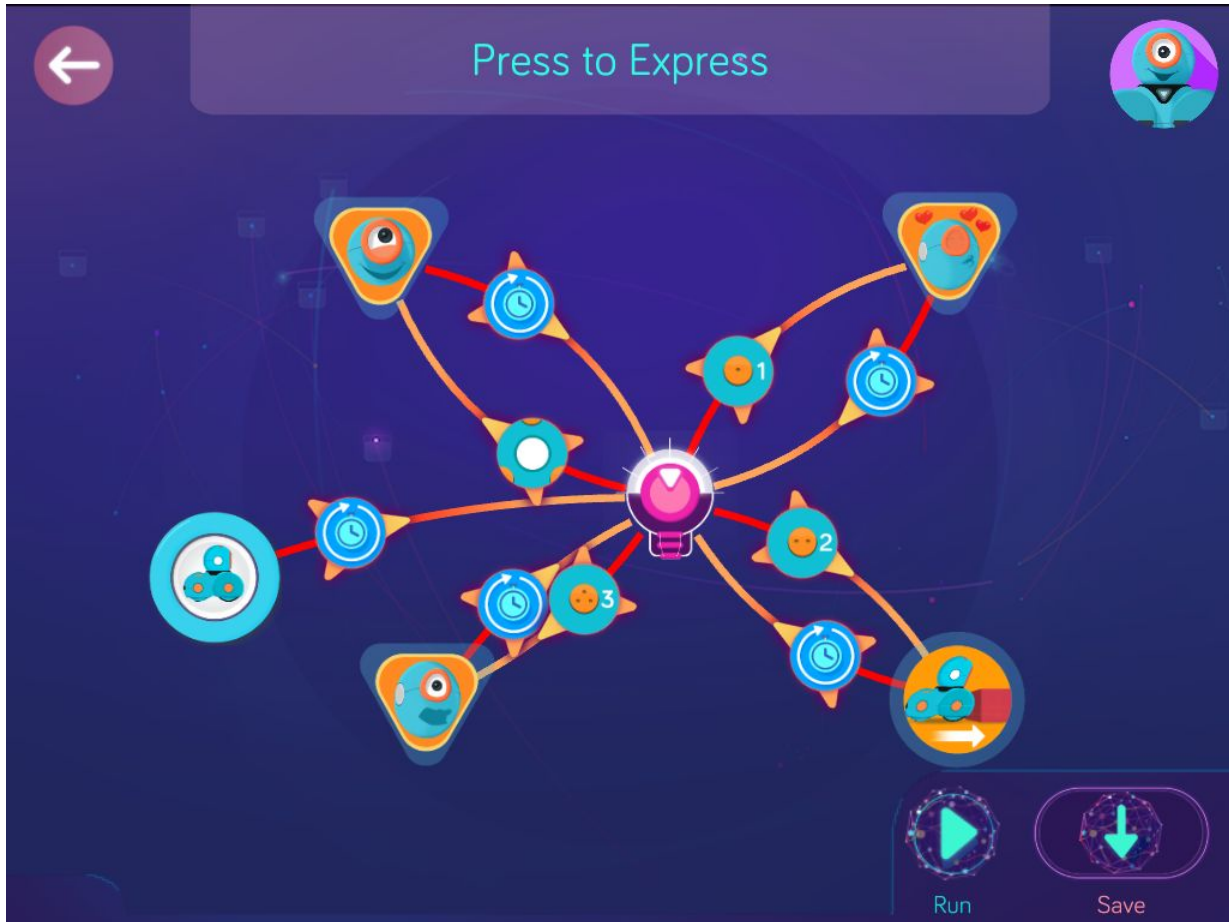
The screenshot shows the Blockly programming environment for the Dash robot. At the top, a text instruction reads: "After turning on lights, make Dash give a kiss when button 1 is pressed. Then make Dash automatically turn on magenta lights again." The workspace contains a sequence of blocks: a "Light" block (magenta), a "Kiss" block (orange), and another "Light" block (magenta). The "Kiss" block is connected to the "Light" block via a "When button 1 is pressed" block. A "Light" block is also connected to the "Kiss" block via a "When button 1 is pressed" block. The "Light" block at the bottom is currently selected, and its "Light" and "Animation" properties are visible. The interface includes a play button on the left, a trash can on the right, and a question mark icon in the top right corner.

Step 4: After turning on the lights, make Dash push when button 2 is pressed. Then make Dash automatically turn on the magenta lights again.

After turning on lights, make Dash **push** when **button 2 is pressed**. Then make Dash automatically turn on **magenta lights** again.

Light Animation

Step 5: After turning on lights, make Dash burp when button 3 is pressed. Then make Dash automatically turn on magenta lights again.



Wonder Key: t8ho

Level 2: Firefly Watch, Water Wiggle, Marco Polo

Concepts covered

- Program the **Random** cue to randomize a sequence of events.
- Program colored lights to turn on, off or randomly sequence, i.e., orange, magenta, green, blue and yellow.
- Program **Wait for** cue to set a time when Dash will transition to the next command.
- Program **Voice** cue to make Dash turn toward the speaker.
- Record and program original sounds
- Program Dash speed moving forward at cm/s.
- Program wheel differential to make left or right turns.

Key terms

Wait for: putting a hold on the program until a new cue or command is introduced.

Sensors: a device that detects and responds to changes within the environment.

Extension questions

1. How did the random cue affect the programming of the orange, magenta and green lights in Firefly Watch?
2. In Firefly Watch: After each colored light, you programmed Dash to wait for 1.0 s then turn the lights off again. What would have happened if you had used the auto cue instead or wait for?
3. How did you program Dash to turn left in step 2 of Water Wiggle? Based on this information, how would you program Dash to turn right?
4. In Marco Polo, which cue was programmed to make Dash turn toward you when you spoke?
5. How did you program Dash to say, "Polo" in the Marco Polo challenge? What else can you program Dash to say? Can you program a new game?

Firefly Watch

Zone: Firefly Lagoon

Challenge Progression 1

Solutions

Step 1: *Make Dash automatically play a curious sound then automatically turn the lights off.*

The screenshot shows a challenge interface for 'Firefly Watch'. At the top, there are four blue progress dots. Below them, the instruction reads: 'Make Dash automatically play a curious sound then automatically turn the **lights off**.' To the right of the instruction are three icons: a lightbulb, a speaker, and a play button, each with a checkmark. Further right is a question mark icon and a character icon. The main workspace contains a sequence of four blocks connected by arrows. The first block is a character icon (Dash) with a play button icon below it. The second block is a sound icon with a play button icon below it. The third block is a character icon (Dash) with a play button icon below it. The fourth block is a lightbulb icon with a play button icon below it. On the left side, there is a vertical toolbar with a back arrow, a play button, and a trash can icon. At the bottom, there are icons for 'Light' and 'Sound'.

Step 2: After turning the lights off, make Dash randomly choose to turn on orange, magenta, or green lights.

The image shows a Scratch script editor with a dark purple background. At the top, there is a navigation bar with a back arrow, a progress indicator (one checkmark, three dots), and a character icon. Below the navigation bar, a text box contains the instruction: "After turning the lights off, make Dash randomly choose to turn on orange, magenta, or green lights." To the right of the text box, there are icons for lightbulbs in orange, magenta, and green, each with a checkmark. A question mark icon is also present. The main workspace contains a script starting with a "When green flag clicked" block, followed by a "Turn all lights off" block, and then a "Randomly choose" block. The "Randomly choose" block has three options: "orange", "magenta", and "green". Each option is connected to a "Turn light on" block of the corresponding color. The "Turn light on" blocks are connected to a "Sound" block. At the bottom of the workspace, there are icons for "Light" and "Sound", and a trash can icon.

Step 3: After each colored light, make Dash wait for 1.0 s then turn the lights off again.

The image shows a programming interface for the Dash robot. At the top, there are four colored circles (green, yellow, blue, red) with checkmarks, indicating completed steps. Below them is a text instruction: "After each colored light, make Dash wait for 1.0 s then turn the lights off again." To the right of the text are icons for light blocks (grey, yellow, green, pink) and wait blocks (blue, purple, blue, blue). A question mark icon and a Dash robot head icon are also present. The main workspace contains a sequence of blocks: a Dash robot head, followed by three blue wait blocks (1.0s), then a grey light block, a yellow light block (1.0s), a green light block (1.0s), and a pink light block (1.0s). The interface includes a left sidebar with navigation and play buttons, a bottom bar with a trash icon and labels for "Light" and "Sound", and a top right corner with a Dash robot head icon.

Step 4: After turning the lights off, make Dash randomly choose to turn around then automatically play the sound again.



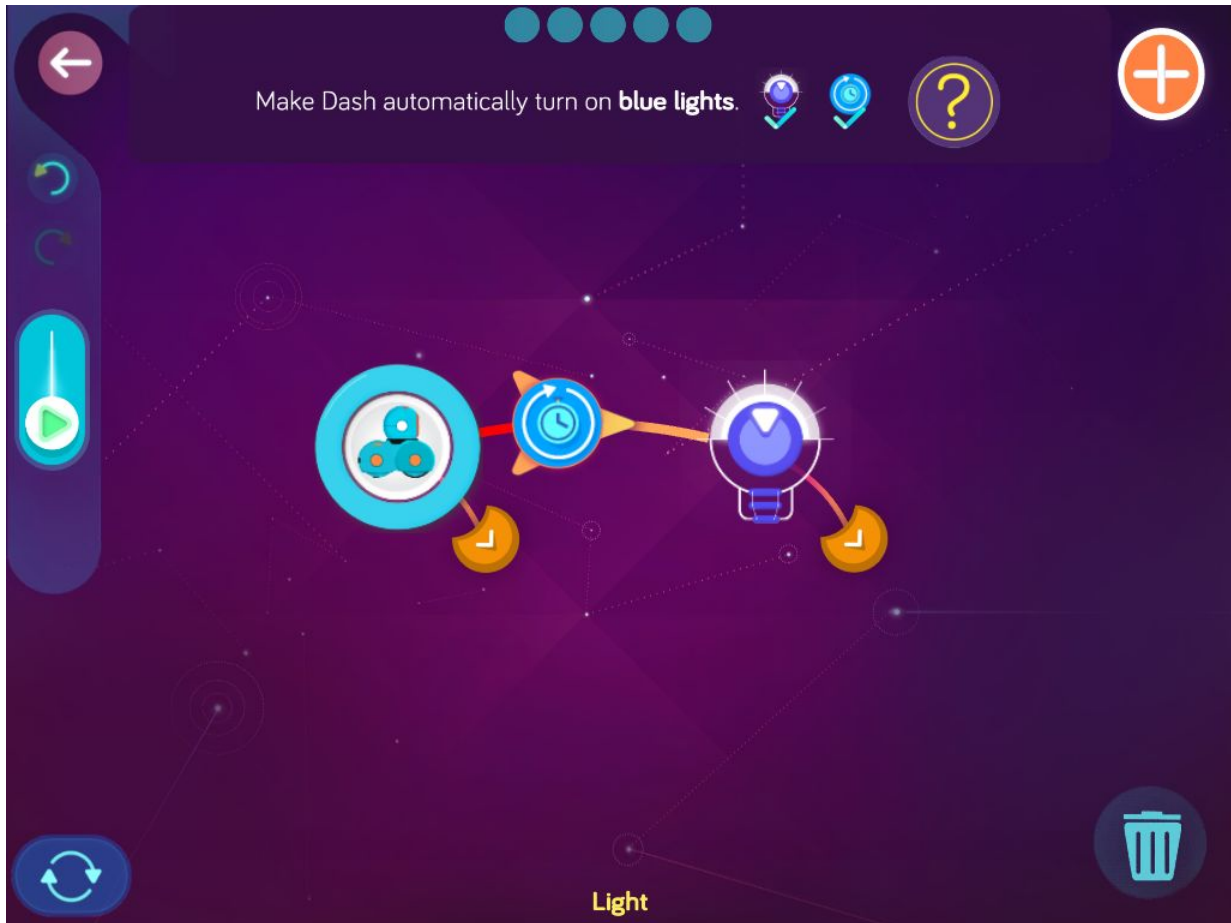
Wonder Key: vtqc

Water Wiggle

Zone: Firefly Lagoon
Challenge Progression 2

Solutions

Step 1: *Make Dash automatically turn on blue lights.*



Step 2: After turning on the light, make Dash automatically turn left with the left wheel set 30 cm/s and the right wheel set to 40 cm/s.

After turning on the light, make Dash automatically **turn left** with the left wheel set to **30 cm/s** and the right wheel set to **40 cm/s**.

Light Move

Step 3: After turning left, make Dash instantly play the Tugboat sound.

The image shows a Scratch workspace with a sequence of four event blocks connected by arrows:

- 1. **When green flag clicked** (Event block)
- 2. **Turn left 90 degrees** (Motion block)
- 3. **Play sound Tugboat sound instantly** (Sound block, highlighted in yellow)
- 4. **Say Hello for 2 secs** (Speech block)

The workspace includes a top toolbar with a back arrow, a progress indicator (5 circles, 2 checked), a help icon, and a plus sign. A text box at the top left contains the instruction: "After turning left, make Dash **instantly** play the **Tugboat** sound." A bottom toolbar contains a refresh icon, a legend with "Light", "Sound", and "Move" categories, and a trash icon.

Step 4: After the sound, make Dash wait for 1.0 then turn right with the left wheel set to 40 cm/s and the right wheel set to 30 cm/s.

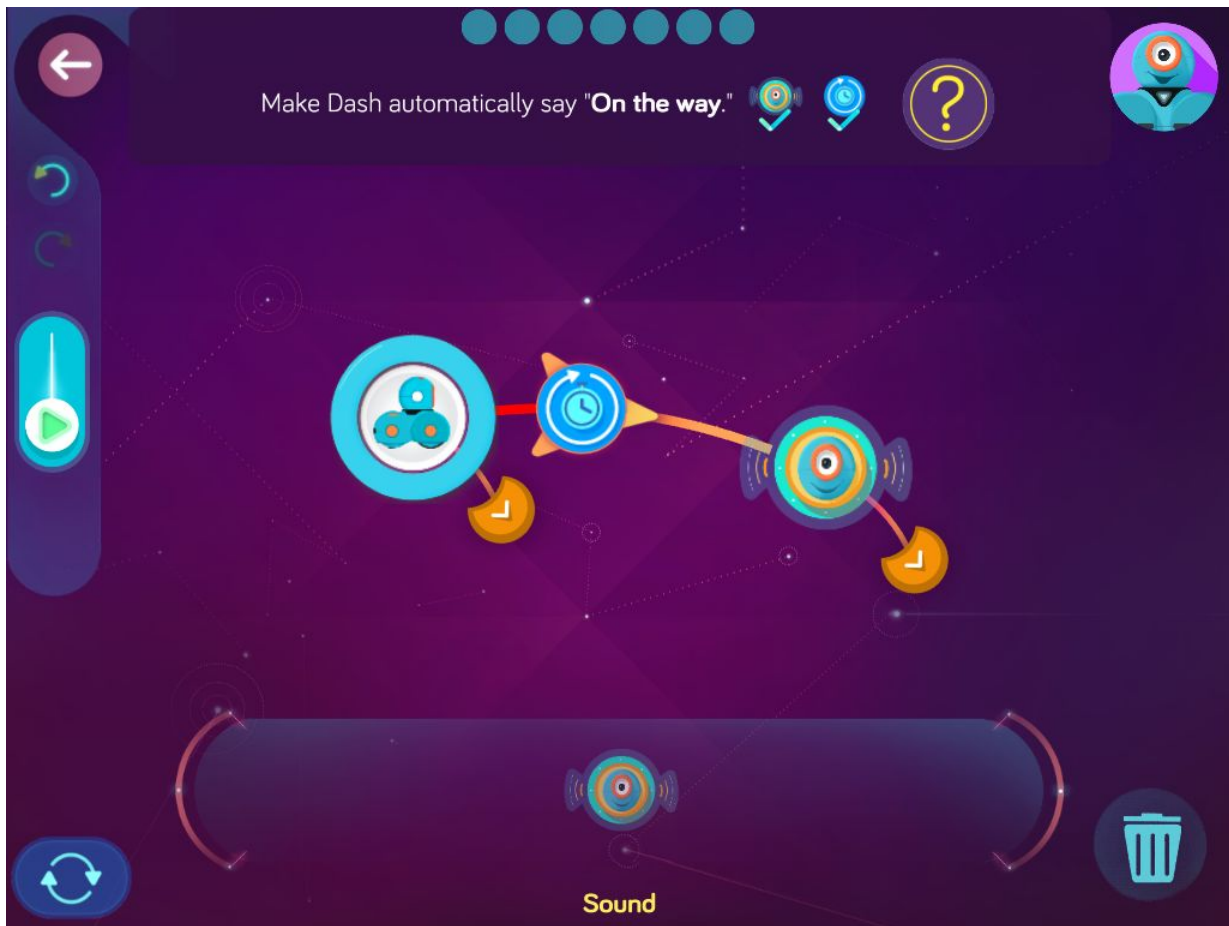
The image shows a programming interface for the Dash robot. At the top, there are five progress indicators: three green checkmarks and two blue circles. Below them is a text box with the instruction: "After the sound, make Dash wait for 1.0 s then turn right with the left wheel set to 40 cm/s and the right wheel set to 30 cm/s." To the right of the text are several icons: a lightbulb, a speaker, a play button, a question mark, and a plus sign. The main workspace contains a sequence of five blocks connected by arrows, starting from a lightbulb icon. The blocks are: a lightbulb icon, a speaker icon, a "1.0s" wait block, and a "Move" block with directional arrows. A trash can icon is in the bottom right. At the bottom center, the words "Light", "Sound", and "Move" are listed, with "Move" highlighted in yellow. On the left side, there are navigation icons: a back arrow, a refresh icon, a play button, and a circular arrow icon.

Marco Polo

Zone: Firefly Lagoon
Challenge Progression 3

Solutions

Step 1: Make Dash say, "On the way!"



Step 2: After speaking, make Dash instantly turn on green lights.

The image shows a Scratch script editor interface. At the top, a text box contains the instruction: "After speaking, make Dash **instantly** turn on green lights." Above this text is a progress indicator with seven blue circles, the first of which contains a white checkmark. To the right of the text are four icons: a lightbulb with a checkmark, a lightbulb, a speech bubble with a checkmark, and a speech bubble. Further right is a yellow question mark icon and a circular profile icon of a blue robot head.

The main workspace features a dark purple background with a grid of faint dotted lines. A script is being built on a blue circular icon representing the robot Dash. The script consists of three event blocks connected by orange arrows:

- The first block is a "When spoken" event block.
- The second block is a "Play sound" block, with a small orange arrow pointing to the right.
- The third block is a "Wait 0.00 seconds" block, with a small orange arrow pointing to the right.
- The fourth block is a "Turn on green lights" block, with a small orange arrow pointing to the right.

Each block has a small orange arrow pointing to the right, indicating the flow of the script. On the left side of the workspace, there is a vertical toolbar with a play button and a refresh button. At the bottom of the workspace, there are two labels: "Light" in yellow and "Sound" in white. A trash can icon is located in the bottom right corner.

Step 3: Make Dash turn to you when you say, "Marco!".

Make Dash to **turn to your voice** when you say "Marco!".

Light Sound **Move**

The image shows a Scratch script editor interface. At the top, there is a progress indicator with seven circles, two of which are checked. Below this is a text box containing the instruction: "Make Dash to **turn to your voice** when you say 'Marco!'." To the right of the text box are several icons representing different Scratch blocks, including a question mark icon. The main workspace displays a sequence of blocks connected by a red line. The blocks are: a 'when green flag clicked' block, a 'play sound' block, a 'say Marco! for 2 seconds' block, a 'turn 180 degrees' block, and another 'say Marco! for 2 seconds' block. The background is a dark purple space with a grid of stars and a faint outline of a planet. On the left side, there are navigation icons for back, forward, and play. At the bottom, there are icons for undo, redo, and a trash can. The text 'Light', 'Sound', and 'Move' is visible at the bottom of the workspace.

Step 6: After saying, "Polo," make Dash instantly move forward 50 cm.

The image shows a Scratch script editor interface. At the top, a text box contains the instruction: "After saying 'Polo,' make Dash instantly move forward 50 cm." Above this text are five checkmarks and a question mark icon. The script area contains the following blocks in order:

- When green flag clicked (blue circle with a white play button)
- Say "Polo" for 2 seconds (blue speech bubble)
- Play sound "Polo" (blue speaker icon)
- Say "Polo" for 2 seconds (blue speech bubble)
- Move forward 50 cm (blue arrow)

At the bottom of the editor, there are labels for "Light", "Sound", and "Move" with corresponding icons. A trash can icon is also visible in the bottom right corner.

Level 3: Thrill Ride

Concepts covered

- Program to **Push Dash Backward** as cue for next command.
- Program to **Move Dash Forward** as cue for next command.
- Program Dash to move forward as fast as possible.

Key terms

Move Forward: Dash will go to the next state when moved forward.

Push Backward: Dash will go to the next state when pushed backward.

Extension questions

1. What new cues were introduced in the Thrill Ride? (Push Forward, Move Backward)
2. Brainstorm ways in which this cue could be used in another type of amusement park ride.
3. When programming the Thrill Ride, did all of your commands sequence from left to right? Explain.
4. What is the top speed that Dash can be programmed to move forward? Backward?
5. In Thrill Ride, you set both of Dash's wheels at the same speed. What would have happened if you had set the left and right wheels at two different speeds? Try it.

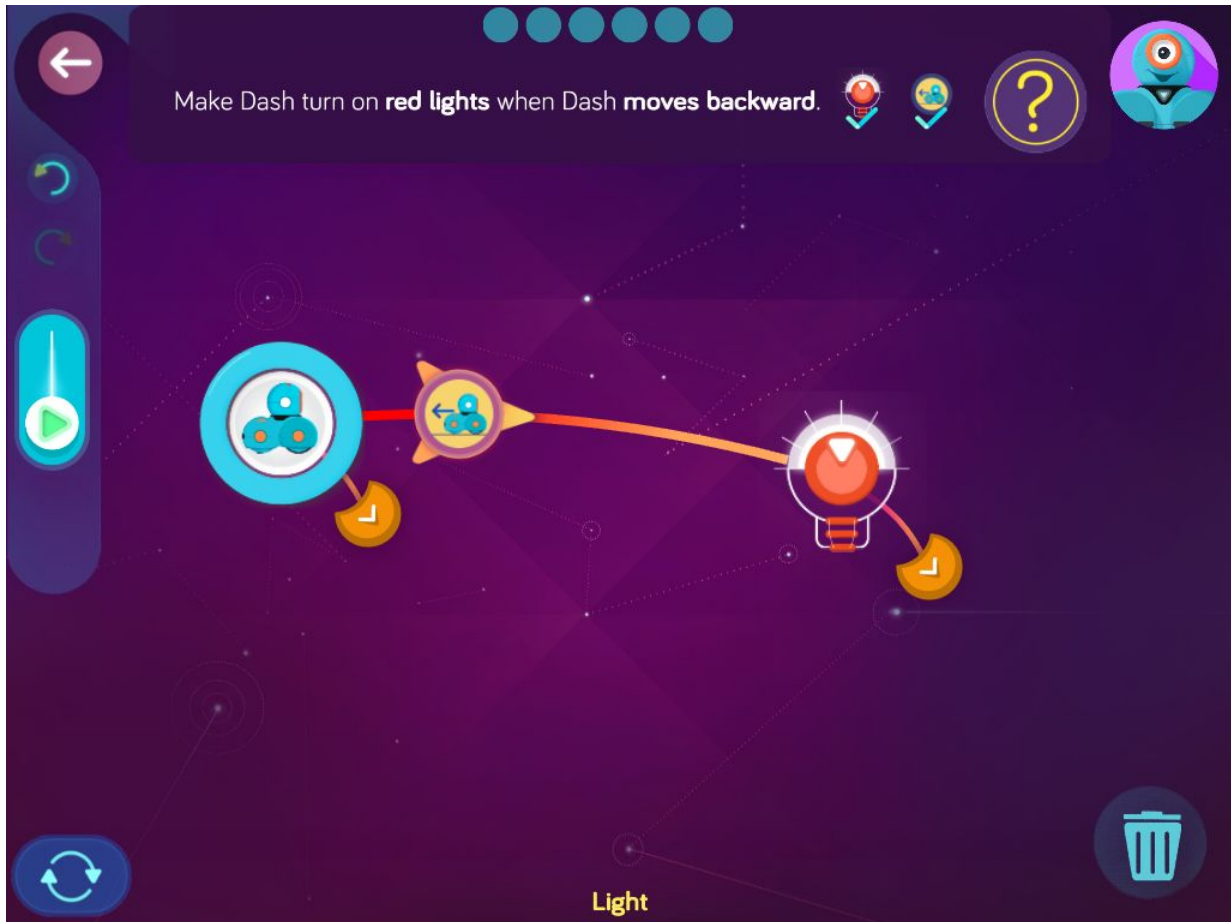
Thrill Ride

Zone: Dragon Reach

Challenge Progression 1

Solutions

Step 1: *Make Dash turn on red lights when Dash moves backward.*



Step 2: After the red light, make Dash play a brave sound when Dash moves backward again.

After the red light, make Dash play a brave sound when Dash **moves backward** again.

Light Sound

Step 3: After the brave sound, make Dash automatically turn on red lights again.

The screenshot shows a programming interface with a dark purple background. At the top, there is a navigation bar with a back arrow, a progress indicator (five circles, the first two are checked), and a character icon. Below the progress indicator, the text reads: "After the brave sound, make Dash automatically turn on **red lights** again." To the right of this text are four icons: a lightbulb, a speech bubble, a sound icon, and a play button, each with a checkmark. A yellow question mark icon is also present.

The main workspace contains a sequence of four event blocks connected by arrows. The first block is a "When green flag clicked" event. The second block is a "When sound finished playing" event, with "brave" selected in the sound dropdown. The third block is a "When light is on" event, with "red" selected in the light dropdown. The fourth block is a "When sound finished playing" event, with "brave" selected in the sound dropdown. Each block has a yellow "L" icon below it.

At the bottom of the workspace, there are two labels: "Light" in yellow and "Sound" in white. A trash can icon is located in the bottom right corner. On the left side, there is a vertical toolbar with a play button and a refresh button.

Step 4: After the red light, make Dash turn on **green lights** when Dash **moves forward**. After the green light, make Dash automatically play a silly sound.

After the red light, make Dash turn on **green lights** when Dash **moves forward**. After the green light, make Dash automatically play a silly sound.

Light Sound

Step 6: After moving forward, make Dash wait for 3.0 s then stop. Then make Dash wait for 1.0 se and turn on red lights again.



Wonder Key: ajcc

Level 4: Snowman, Crazy Skates

Concepts covered

- Programming and customizing eye ring patterns to create an expression, i.e., a smile
- Programming the **Listener** to wait for a cue
- Programming a clap as a cue for the next command
- Programming a loop

Key terms

Eye Ring: a customizable circle of lights on the front of Dash's face.

Moving Basics: Dash will begin spinning, turning, or going straight depending on the speed you choose for each wheel.

The Listener: works at the same time as the main code and waits until a cue causes it to change to a new state.

Loop: a set of instructions that a computer program repeats.

Extension questions

1. How can Dash's eye ring be customized to resemble a smile?
2. What other expressions can you create for Dash using the eye ring?
3. How were you able to make Dash switch back and forth between two different eye ring patterns?
4. Which cue did you use to program the time delay between eye ring patterns?
5. In Step 3 of Crazy Skates, how did you create a loop? What happened in the program when you did?

Snowman

Zone: Yeti Pass

Challenge Progression 1

Solutions

Step 1: Make Dash switch back and forth between the two eye ring patterns every 0.1s.



Step 2: After the *Listener*, make Dash turn off the eye lights when a **clap is heard**. Then make Dash automatically turn the **lights off**.

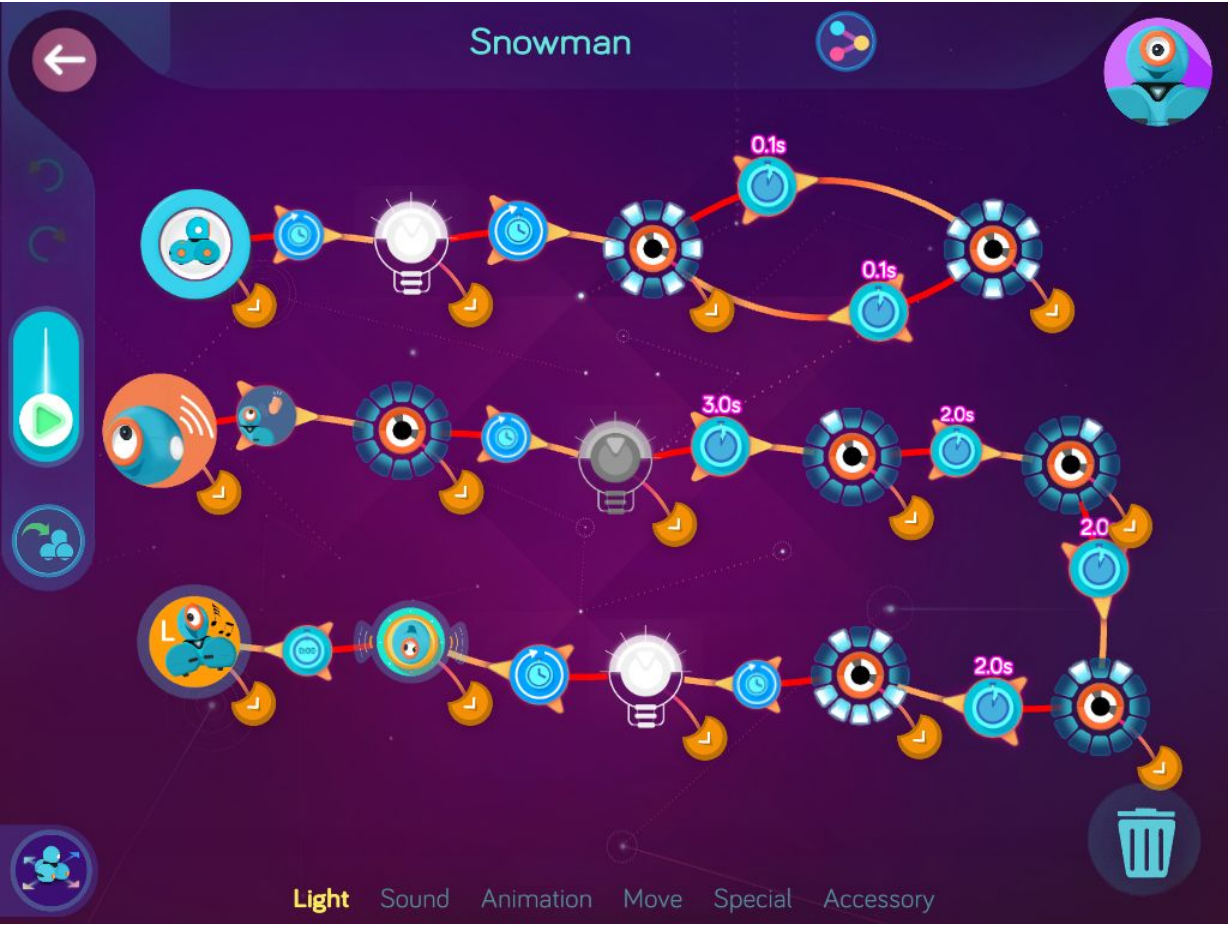
The image shows a programming interface for the Dash robot. At the top, there is a row of five blue circles, with the first one containing a white checkmark. Below this, a text box contains the instruction: "After the **Listener**, make Dash turn off the eye lights when a **clap is heard**. Then make Dash automatically turn the **lights off**." To the right of the text box are several icons: a lightbulb, a speaker, a robot head, and a lightbulb, each with a checkmark below it. A question mark icon is also present. The main workspace is a dark purple grid with two rows of blocks. The top row contains: a blue circle with a robot head, a blue circle with a lightbulb, a white lightbulb icon, a blue circle with a lightbulb, a blue circle with a speaker, a blue circle with a lightbulb, a blue circle with a speaker, a blue circle with a lightbulb, a blue circle with a speaker, a blue circle with a lightbulb, and a blue circle with a speaker. The bottom row contains: a blue circle with a robot head, a blue circle with a lightbulb, a blue circle with a speaker, a blue circle with a lightbulb, and a blue circle with a speaker. The blocks are connected by red lines. A trash can icon is in the bottom right corner. The text "Light Special" is at the bottom center.

Step 3: After turning the eye lights off, make Dash turn on one eye light after waiting 3.0 s.

After turning the lights off, make Dash turn on one eye light after waiting **3.0 s**.

Light Special

Step 6: After Dash sings, "Hello, hello, hello, hello...", make Dash instantly dance left.



Wonder Key: d52i

Crazy Skates

Zone: Yeti Pass

Challenge Progression 2

Solutions

Step 1: Make Dash automatically turn on white lights, turn on three eye lights, and play the "Let's go!" #3 sound.

The image shows a Scratch-style programming interface for a challenge. At the top, there are four blue progress dots. Below them, a text box contains the instruction: "Make Dash automatically turn on **white lights**, turn on three **eye lights**, and play the **Let's go! #3** sound." To the right of the text are four icons: a white lightbulb, three eye icons, and a speaker icon, each with a checkmark. A question mark icon and a character icon are also present. The main workspace features a sequence of seven blocks connected by arrows: a 'When green flag clicked' block, a 'Turn on white light' block, a 'Turn on eye light' block, a 'Turn on eye light' block, a 'Turn on eye light' block, a 'Play sound' block, and a 'Play sound' block. The 'Light' and 'Sound' categories are visible at the bottom. A trash can icon is in the bottom right corner.

Step 2: Edit the first **Moving Basics** state (the one on the left) to: left wheel 40 cm/s, right wheel 30 cm/s. Then edit the other **Moving Basics** state: left wheel 30 cm/s, right wheel 40 cm/s.

The screenshot shows a programming environment with a dark purple background. At the top, there are four colored circles (green, blue, red, blue) and a checkmark icon. Below them is a text box with instructions: "Edit the first **Moving Basics** state (the one on the left) to: left wheel 40 cm/s, right wheel 30 cm/s. Then edit the other **Moving Basics** state: left wheel 30 cm/s, right wheel 40 cm/s." To the right of the text are several icons: a lightbulb, a gear, a play button, a stop button, a question mark, and a character icon. The main workspace contains a sequence of blocks connected by arrows. The blocks are: a 'Light' block, a 'Sound' block, a 'Move' block, a 'Light' block, a 'Sound' block, a 'Move' block, a 'Light' block, a 'Sound' block, a 'Move' block, a 'Light' block, a 'Sound' block, a 'Move' block, a 'Light' block, a 'Sound' block, a 'Move' block, a 'Light' block, a 'Sound' block, a 'Move' block, a 'Light' block, a 'Sound' block, a 'Move' block. The 'Move' blocks have durations of 0.4s and 0.2s. At the bottom, there are icons for 'Light', 'Sound', and 'Move', and a trash can icon.

Step 4: Create a loop by connecting the two new Moving Basics states with two Wait For cues set 2.0.



Wonder Key: t19k

Level 5: Volcanic Trek, Random Volcano

Concepts covered

- Programming distance in centimeters per second
- Programming a spin using wheel differential
- Programming the **Picked Up** cue
- Programming **Put Down** cue
- Programming the **Random** cue for a variable number of seconds between 0 and 5.
- Programming a custom sound using record in the sound command
- Using buttons as cues, e.g., top, 1, 2 and 3 buttons.

Key terms

Picked up: a cue to program Dash to go to the next state after being picked up.

Put Down: a cue to program Dash to go to the next state after being put down.

Extension questions

1. Explain how would you program Dash to move ahead 30 cm.
2. In step 3 of the Volcanic Trek, how were the wheels set in order to make Dash spin?
3. Which direction did Dash spin in the Volcanic Trek? Compare the spin command to a turn command. How are the wheels set in each case?
4. In step 2 of Random Volcano, you used the random cue. Explain how it was used and how it affected Dash running the program over again, at least, 3 times.
5. In the final steps of Random Volcano, buttons 1, 2 and 3 were used as cues. When did you need to push these buttons? What happened if you did not?

Volcanic Trek

Zone: Mount Ashburn

Challenge Progression 1

Step 1: Make Dash automatically turn on blue lights then automatically move forward 20 cm/s.

The image shows a programming interface for a challenge. At the top, there are five blue progress dots. Below them is a text instruction: "Make Dash automatically turn on **blue lights** then automatically **move forward 20 cm/s**." To the right of the text are three icons: a blue lightbulb, a blue lightbulb, and a blue lightbulb, each with a checkmark. Further right is a yellow question mark icon and a character icon. The main workspace is a dark purple grid with a dashed path. A sequence of four icons is connected by arrows: a character icon, a blue lightbulb icon, a blue lightbulb icon, and a blue lightbulb icon. Below the icons are yellow L-shaped corner markers. On the left side, there is a vertical toolbar with a back arrow, a refresh icon, a play button, and a trash can icon. At the bottom, there are two labels: "Light" and "Move".

Step 2: After Dash starts moving, make Dash change the light color every 3.0 s. Make Dash change the lights to **green** then **yellow** then **magenta** then **red**.

After Dash starts moving, make Dash change the light color every **3.0 s**. Make Dash change the lights to **green** then **yellow** then **magenta** then **red**.

Light Move

Step 4: Link the **Listener** state to the **Stop** state using the **Picked up** cue.

The screenshot shows a programming interface with a dark purple background. At the top, there are five progress indicators: three green checkmarks and two blue circles. Below them is a toolbar with icons for a crossed-out lightbulb, a purple lightbulb, a blue lightbulb, a blue lightbulb, a blue lightbulb, a blue lightbulb, a blue lightbulb, and a question mark. A character icon is in the top right corner.

The main workspace contains a sequence of states and actions connected by arrows. The sequence starts with a character icon, followed by a state with a blue lightbulb and a '0:00' timer. This is followed by a state with a blue lightbulb and a '3.0s' timer. The sequence continues with several states, each with a lightbulb of a different color (red, blue, green, yellow) and a '3.0s' timer. The sequence ends with a state with a blue lightbulb and a '3.0s' timer.

At the bottom, there are four categories: **Light**, **Sound**, **Move**, and **Special**. A trash can icon is in the bottom right corner.

Random Volcano

Zone: Mount Ashburn

Challenge Progression 2

Solutions

Step 1: *Make Dash wait for 1.0 s then turn the lights off.*



Step 2: Make Dash wait for a random time up to 5.0 s then turn on red lights.

The image shows a programming interface for a Dash robot. At the top, a progress bar consists of 10 blue circles, with the first one containing a white checkmark. Below this, a text instruction reads: "Make Dash wait for a random time up to 5.0 s then turn on red lights." To the right of the text are four icons: a red lightbulb, a grey lightbulb, a blue circular arrow, and a purple robot head. Further right is a yellow question mark icon and a circular profile icon of a robot head. The main workspace is a dark purple grid with a dashed white grid. A sequence of four blocks is connected in a line: 1. A blue circular block with a robot head icon. 2. A blue circular block with a clock icon and the text "1.0s". 3. A grey lightbulb icon. 4. A purple circular block with a clock icon and the text "5.0s". 5. A red lightbulb icon. Each block has a small yellow L-shaped icon at its bottom right corner. On the left side, there is a vertical toolbar with a back arrow, a refresh icon, a play button, and a circular arrow icon. At the bottom center, the word "Light" is written in yellow. At the bottom right, there is a trash can icon.

Step 3: After the red light, make Dash automatically play the sound of an erupting volcano. Use a **custom sound** to record your own voice making the sound of an erupting volcano.

The screenshot shows the Dash programming interface with a sequence of blocks: a light block, a 1.0s delay block, a light block, a 5.0s delay block, and a sound block. A microphone icon with the number 1 is connected to the sound block, indicating a custom sound recording. The interface includes a top navigation bar with a back arrow, a progress indicator (7 circles, 2 checked), and a help icon. A toolbar on the left contains a play button and a refresh button. A legend at the bottom identifies 'Light' and 'Sound' blocks. A trash icon is in the bottom right corner.

Step 4: After the erupting volcano sound, make Dash return to Start when the top button is pressed.



Step 5: After the lights are off, make Dash turn on orange lights when button 3 is pressed. Then make Dash wait for 1.0 s before turning the lights off again.

After the lights are off, make Dash turn on **orange lights** when **button 3** is pressed. Then make Dash **wait for 1.0 s** before turning the **lights off** again.

Light Sound

Step 6: After the orange lights, make Dash turn on yellow lights when button 2 is pressed. Then make Dash wait for 1.0 s before turning the lights off again.

The image shows a programming interface for the Dash robot. At the top, there is a progress bar with seven circles, the first five of which are checked. Below this is a text instruction: "After the orange lights, make Dash turn on yellow lights when button 2 is pressed. Then make Dash wait for 1.0 s before turning the lights off again." To the right of the text is a palette of programming blocks, including a light block, a wait block, and a button block. A question mark icon is also present. The main workspace contains a flowchart with several nodes: a microphone icon with a '1' in a blue circle, a light icon with an orange glow, a wait block labeled '5.0s', a button icon with a '3' in a blue circle, a light icon with an orange glow, a wait block labeled '1.0s', a button icon with a '2' in a blue circle, a light icon with a yellow glow, a wait block labeled '1.0s', and a light icon with a yellow glow. The flowchart is connected by orange arrows. On the left side, there is a vertical toolbar with a back arrow, a refresh icon, a play button, and a circular arrow icon. At the bottom, there is a legend with 'Light' and 'Sound' labels, and a trash can icon.

Step 7: After the yellow lights, make Dash turn on green lights when button 1 is pressed. Then make Dash automatically play the "Tah dah!" sound.

After the yellow lights, make Dash turn on **green lights** when **button 1** is pressed. Then make Dash automatically play the "Tah dah!" sound.

Light Sound

Step 7: After the “Tah dah!” sound, make Dash turn the lights off again when the top button is pressed.



Wonder Key: axrg

Level 6: Spaceship Rescue, Robot Translator

Concepts covered

- Customizing eye lights
- Programming lights
- Adjusting the speed of a spin
- Cuing eye lights with a button press
- Programming Dash to randomly choose a sound to play
- Programming and recording a custom sound

Key terms

Eye Lights: 12 customizable white LEDs on the front of Dash's face.

Lights: Dash's ear lights, tail lights and spotlight

Extension questions

1. What is the slowest rate of speed you can adjust Dash to spin? What is the fastest?
2. How many different light sources are on Dash? Which lights can be programmed in different colors? Which lights can be programmed to resemble a facial expression, like a smile?
3. How many eye lights does Dash have? How can you program all of Dash's eye lights to turn off? On?
4. In the Robot Translator, what was the easiest/fastest way to replace each happy sound with a custom sound?
5. What was the purpose of the random cue in the Robot Translator?

Spaceship Rescue

Zone: Galaxy Lake

Challenge Progression 1

Step 1: Make Dash automatically turn the lights off then automatically turn off all eye lights.

The image shows a Scratch-style programming interface for a challenge. At the top, a progress bar consists of five teal circles. Below it, a text box contains the instruction: "Make Dash automatically turn the **lights off** then automatically **turn off all eye lights**." To the right of the text are three icons: a lightbulb, an eye, and a speech bubble, each with a checkmark. Further right is a question mark icon and a character icon. The main workspace features a sequence of five objects connected by a red line: a blue robot head, a blue clock icon, a grey lightbulb, another blue clock icon, and a blue eye icon. Each object has a yellow L-shaped 'lock' icon below it. On the left side, there is a vertical toolbar with a back arrow, a play button, and a refresh button. At the bottom, there is a 'Light' label and a trash can icon.

Step 2: After the eye ring, make Dash automatically spin slowly with the left wheel set to 20 cm/s and the right wheel set to -20 cm/s.

After the eye ring, make Dash automatically spin slowly with the left wheel set to 20 cm/s and the right wheel set to -20 cm/s.

Light Move

Step 3: After spinning slowly, make Dash turn on two eye lights when button 1 is pressed.

The image shows a programming interface for a character named Dash. At the top, there is a progress bar with five circles, the first two of which are checked. Below this, a text box contains the instruction: "After spinning slowly, make Dash turn on two eye lights when button 1 is pressed." To the right of the text box are several icons representing different actions: a lightbulb, two eye icons, a circular arrow, and a question mark. The main workspace features a sequence of eight blocks connected by a red line. The blocks are: 1. A 'When green flag clicked' block. 2. A 'Spin slowly' block. 3. A 'Turn on light' block. 4. A 'Turn on eye lights' block. 5. A 'Turn on eye lights' block. 6. A 'When button 1 is pressed' block. 7. A 'Turn on eye lights' block. 8. A 'Turn on eye lights' block. On the left side, there is a vertical toolbar with a play button and a refresh button. At the bottom, there is a 'Light' label, a 'Move' label, and a trash can icon.

Step 5: After turning on four eye lights, make Dash turn on six eye lights when button 3 is pressed.

The image shows a Scratch script editor interface. At the top, there is a progress bar with six circles, the first four of which are checked. Below the progress bar, a text box contains the instruction: "After turning on four eye lights, make Dash turn on six eye lights when button 3 is pressed." The script area contains the following sequence of blocks:

- When green flag clicked, go to the top of the screen.
- When green flag clicked, play sound: lightbulb.
- When green flag clicked, turn on eye light 1.
- When green flag clicked, turn on eye light 2.
- When green flag clicked, turn on eye light 3.
- When green flag clicked, turn on eye light 4.
- When green flag clicked, turn on eye light 5.
- When green flag clicked, turn on eye light 6.
- When green flag clicked, turn on eye light 7.
- When green flag clicked, turn on eye light 8.
- When green flag clicked, turn on eye light 9.
- When green flag clicked, turn on eye light 10.
- When green flag clicked, turn on eye light 11.
- When green flag clicked, turn on eye light 12.
- When green flag clicked, turn on eye light 13.
- When green flag clicked, turn on eye light 14.
- When green flag clicked, turn on eye light 15.
- When green flag clicked, turn on eye light 16.
- When green flag clicked, turn on eye light 17.
- When green flag clicked, turn on eye light 18.
- When green flag clicked, turn on eye light 19.
- When green flag clicked, turn on eye light 20.
- When green flag clicked, turn on eye light 21.
- When green flag clicked, turn on eye light 22.
- When green flag clicked, turn on eye light 23.
- When green flag clicked, turn on eye light 24.
- When green flag clicked, turn on eye light 25.
- When green flag clicked, turn on eye light 26.
- When green flag clicked, turn on eye light 27.
- When green flag clicked, turn on eye light 28.
- When green flag clicked, turn on eye light 29.
- When green flag clicked, turn on eye light 30.
- When green flag clicked, turn on eye light 31.
- When green flag clicked, turn on eye light 32.
- When green flag clicked, turn on eye light 33.
- When green flag clicked, turn on eye light 34.
- When green flag clicked, turn on eye light 35.
- When green flag clicked, turn on eye light 36.
- When green flag clicked, turn on eye light 37.
- When green flag clicked, turn on eye light 38.
- When green flag clicked, turn on eye light 39.
- When green flag clicked, turn on eye light 40.
- When green flag clicked, turn on eye light 41.
- When green flag clicked, turn on eye light 42.
- When green flag clicked, turn on eye light 43.
- When green flag clicked, turn on eye light 44.
- When green flag clicked, turn on eye light 45.
- When green flag clicked, turn on eye light 46.
- When green flag clicked, turn on eye light 47.
- When green flag clicked, turn on eye light 48.
- When green flag clicked, turn on eye light 49.
- When green flag clicked, turn on eye light 50.
- When green flag clicked, turn on eye light 51.
- When green flag clicked, turn on eye light 52.
- When green flag clicked, turn on eye light 53.
- When green flag clicked, turn on eye light 54.
- When green flag clicked, turn on eye light 55.
- When green flag clicked, turn on eye light 56.
- When green flag clicked, turn on eye light 57.
- When green flag clicked, turn on eye light 58.
- When green flag clicked, turn on eye light 59.
- When green flag clicked, turn on eye light 60.
- When green flag clicked, turn on eye light 61.
- When green flag clicked, turn on eye light 62.
- When green flag clicked, turn on eye light 63.
- When green flag clicked, turn on eye light 64.
- When green flag clicked, turn on eye light 65.
- When green flag clicked, turn on eye light 66.
- When green flag clicked, turn on eye light 67.
- When green flag clicked, turn on eye light 68.
- When green flag clicked, turn on eye light 69.
- When green flag clicked, turn on eye light 70.
- When green flag clicked, turn on eye light 71.
- When green flag clicked, turn on eye light 72.
- When green flag clicked, turn on eye light 73.
- When green flag clicked, turn on eye light 74.
- When green flag clicked, turn on eye light 75.
- When green flag clicked, turn on eye light 76.
- When green flag clicked, turn on eye light 77.
- When green flag clicked, turn on eye light 78.
- When green flag clicked, turn on eye light 79.
- When green flag clicked, turn on eye light 80.
- When green flag clicked, turn on eye light 81.
- When green flag clicked, turn on eye light 82.
- When green flag clicked, turn on eye light 83.
- When green flag clicked, turn on eye light 84.
- When green flag clicked, turn on eye light 85.
- When green flag clicked, turn on eye light 86.
- When green flag clicked, turn on eye light 87.
- When green flag clicked, turn on eye light 88.
- When green flag clicked, turn on eye light 89.
- When green flag clicked, turn on eye light 90.
- When green flag clicked, turn on eye light 91.
- When green flag clicked, turn on eye light 92.
- When green flag clicked, turn on eye light 93.
- When green flag clicked, turn on eye light 94.
- When green flag clicked, turn on eye light 95.
- When green flag clicked, turn on eye light 96.
- When green flag clicked, turn on eye light 97.
- When green flag clicked, turn on eye light 98.
- When green flag clicked, turn on eye light 99.
- When green flag clicked, turn on eye light 100.

At the bottom of the script area, there are two buttons labeled "Light" and "Move". A trash can icon is located in the bottom right corner of the script area.

Robot Translator

Zone: Galaxy Lake

Challenge Progression 2

Solutions

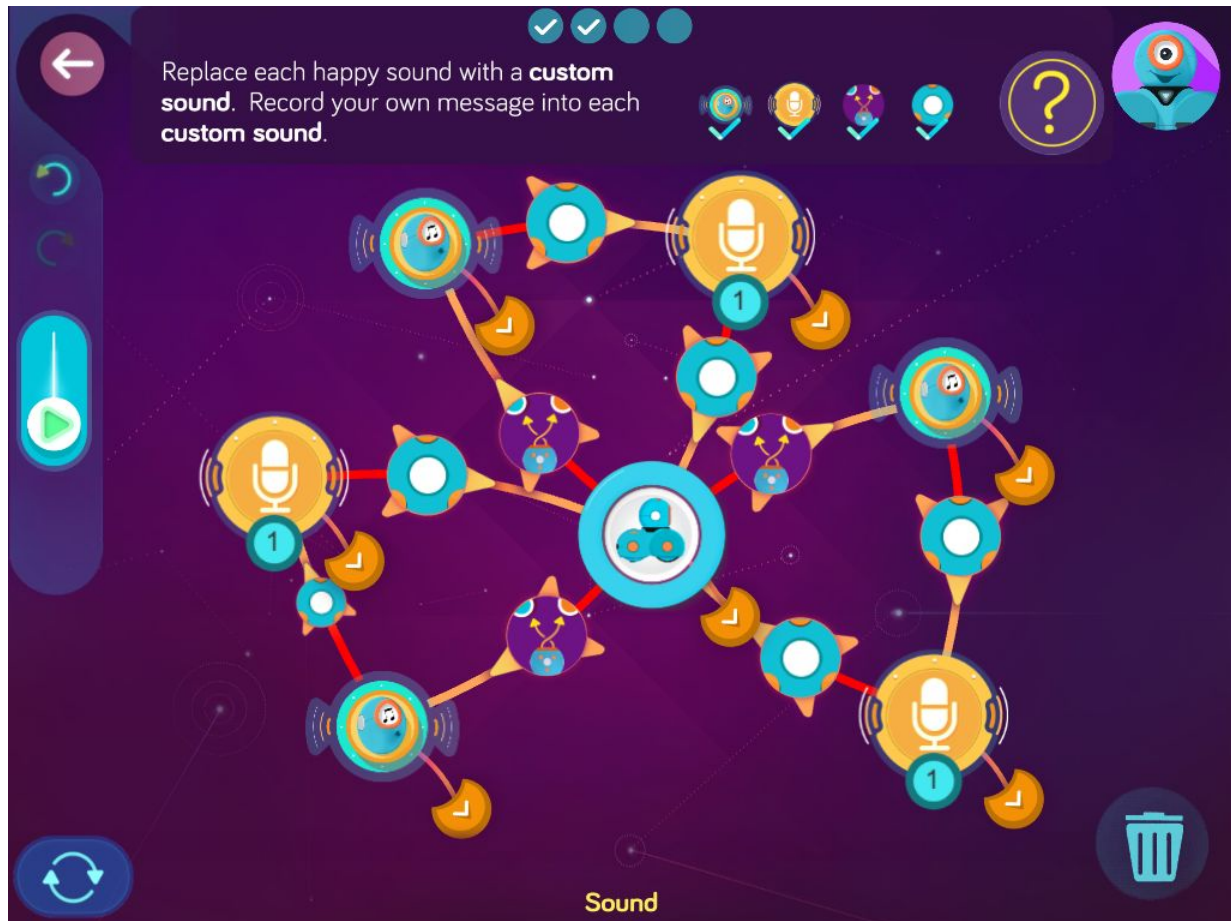
Step 1: *Make Dash randomly choose an FX sounds to play.*



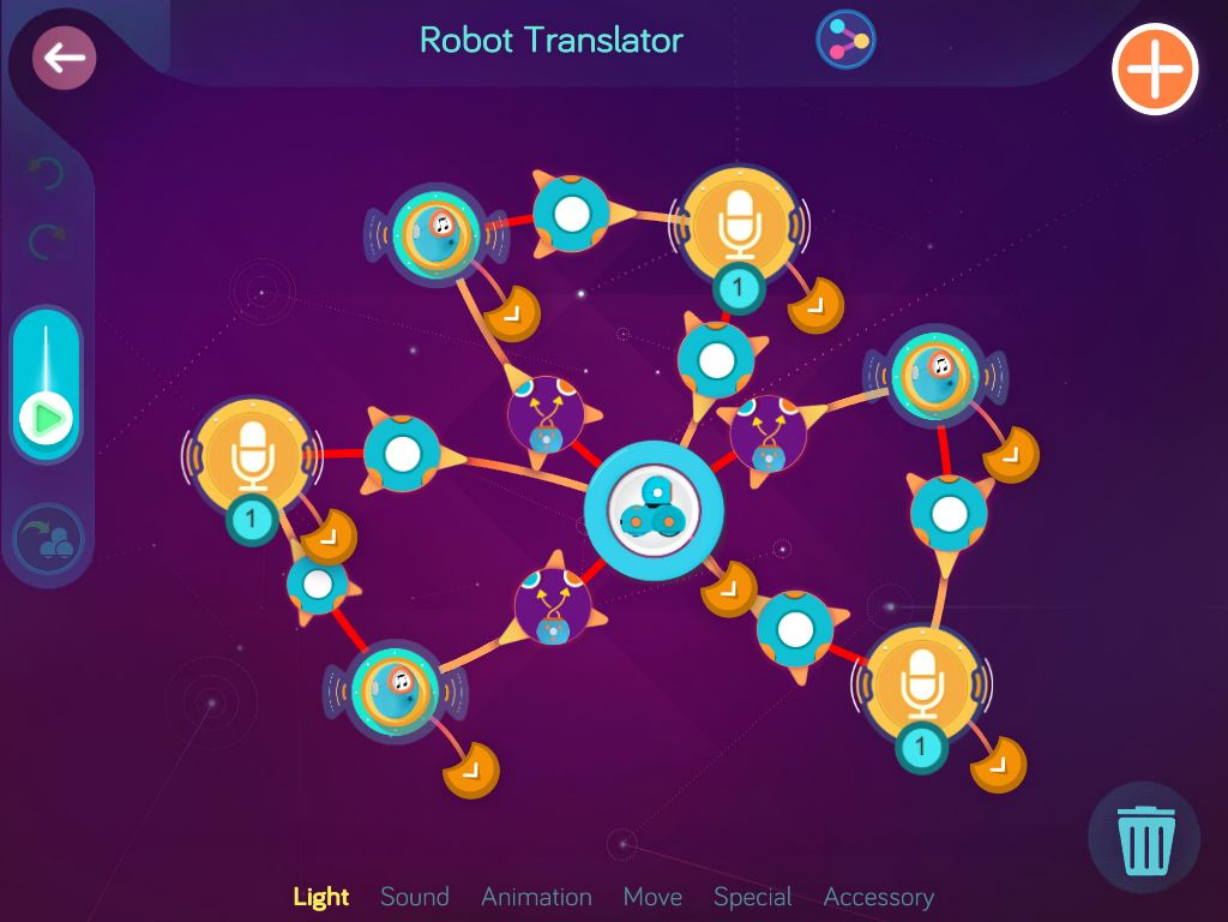
Step 2: After each FX sound, make Dash play a happy sound when the top button is pressed. Then make Dash return to **Start** when the top button is pressed again.

The image shows a Scratch workspace with a dark purple background. At the top, there is a text box with the instruction: "After each FX sound, make Dash play a happy sound when the **top button is pressed**. Then make Dash return to **Start** when the **top button is pressed** again." Above the text box are four colored circles (green, blue, red, yellow) and a checkmark icon. To the right of the text box are four sound block icons: a blue sound block with a checkmark, a blue sound block with a checkmark, a purple sound block, and a blue sound block with a checkmark. A question mark icon is also present. In the top right corner, there is a character icon of Dash. The main workspace contains a central character icon of Dash surrounded by a network of smaller character icons connected by red lines. At the bottom center, the word "Sound" is written. In the bottom right corner, there is a trash can icon. On the left side, there is a vertical toolbar with a back arrow, a play button, and a refresh button.

Step 3: Replace each happy sound with a **custom sound**. Record your own message into each **custom sound**.



Step 4: Play with Dash: Press the top button until you hear each of your recorded sounds.



Wonder Key: 56dz

Level 7: Doing Donuts, Spinning Dash Game, Butler Dash

Concepts covered

- Programming wheel differential for turns
- Programming a cue for when Dash gets stuck
- Programming the **Top Button Released** cue
- Programming Dash to move a set distance in cm.
- Programming Dash to sense an obstacle in front

Key terms

Stuck: a cue for Dash to go to the next state when stuck.

Top Button Released: a cue to program Dash to go to the next state when you release the big white button on Dash's head.

Extension questions

1. When making a left turn, which wheel turns faster? The left or right wheel. Explain why.
2. How is programming a right turn different than programming a left turn?
3. How is programming a turn different than programming a spin?
4. Which direction did Dash spin when you programmed the turn for -45 degrees in Butler Dash? Which direction did Dash spin when you programmed the turn for 45 degrees?
5. What is the farthest degree that you can set Dash to spin in either direction?

Doing Donuts

Zone: Kong City

Challenge Progression 1

Step 1: Make Dash automatically play a silly sound, play a transport sound, then spin right (left wheel 80 cm/s, right wheel 10 cm/s).

The screenshot shows a programming environment with a dark purple background. At the top, there is a instruction box with a back arrow on the left and a character icon on the right. The instruction text reads: "Make Dash automatically play a silly sound, play a transport sound, then spin right (left wheel 80 cm/s, right wheel 10 cm/s)". To the right of the text are four icons: a checkmark, a sound icon, a wheel icon, and a checkmark. Below the instruction box is a sequence of five circular action blocks connected by a red line. The first block is a "Play Sound" block with a speaker icon and a "Silly" sound icon. The second block is a "Play Sound" block with a speaker icon and a "Transport" sound icon. The third block is a "Spin" block with a circular arrow icon and a "Right" direction icon. The fourth block is a "Spin" block with a circular arrow icon and a "Right" direction icon. The fifth block is a "Spin" block with a circular arrow icon and a "Right" direction icon. On the left side of the screen, there is a vertical toolbar with a play button, a volume slider, and a refresh button. At the bottom, there is a "Sound" label in yellow and a "Move" label in white. A trash can icon is located in the bottom right corner.

Step 2: After spinning right, make Dash wait for 3.0 s, then spin left (left wheel set to 10 cm/s, right wheel set to 80 cm/s.) Then make Dash wait for 3.0 s and spin right again.

After spinning right, make Dash **wait for 3.0 s** then spin left (left wheel **10 cm/s**, right wheel **80 cm/s**). Then make Dash **wait for 3.0 s** and spin right again.

Sound Move

Step 3: While spinning left or spinning right, make Dash say, "Oh no!" if Dash gets stuck.

While spinning left or spinning right, make Dash say "Oh no!" if Dash gets stuck.

3.0s

3.0s

Sound Move

The image shows a Scratch script editor interface. At the top, there are five progress indicators, two of which are checked. A navigation arrow is on the left. The main workspace contains a sequence of blocks: a 'Say' block with the text 'Oh no!', a 'Wait' block set to 3.0s, and another 'Say' block. The workspace also features a robot icon and various tool icons. The bottom of the interface includes a 'Sound' and 'Move' section, a trash icon, and a refresh button.

Step 4: When Dash says, "Oh no!" make Dash instantly stop.

The image shows a Scratch script editor interface. At the top, a text box contains the instruction: "When Dash says 'Oh no!' make Dash instantly stop." Below this, a series of blue checkmarks indicates that previous steps in the sequence have been completed. The main workspace displays a sequence of Scratch blocks connected by orange arrows. The sequence starts with a "When Green Flag Clicked" block, followed by several "Say" blocks with durations of 3.0s. A "Sound" block is also present. The final block in the sequence is a "Stop" block, which is highlighted in red, indicating it is the current step being worked on. The interface includes a left sidebar with navigation icons (back, forward, play, volume), a top bar with a question mark icon, and a bottom bar with "Sound" and "Move" labels and a trash icon.

Spinning Dash Game

Zone: Kong City

Challenge Progression 2

Solutions

Step 1: Make Dash sing the *Inputs and Outputs* song when the top button is pressed.



Step 2: When Dash starts singing, make Dash instantly turn on **white lights**. Then make Dash automatically spin right (left wheel 30 cm/s, right wheel -30 cm/s).

The image shows a programming interface for the Dash robot. At the top, a text box contains the instruction: "When Dash starts singing, make Dash instantly turn on **white lights**. Then make Dash automatically spin right (left wheel 30 cm/s, right wheel -30 cm/s)." Above this text are four colored circles (green, blue, blue, blue) and a question mark icon. Below the text, a sequence of programming blocks is connected by arrows. The first block is a "When I click the green flag" block. The second block is a "When I hear a sound" block. The third block is a "Turn on white lights" block. The fourth block is a "Spin right" block. The interface includes a play button on the left, a trash can icon at the bottom right, and a legend at the bottom with "Light" (blue), "Sound" (yellow), and "Move" (blue) categories.

Step 3: Make Dash say, "Oh no!" when the top button is released.

Make Dash say "Oh no!" when the top button is released.

Light Sound Move

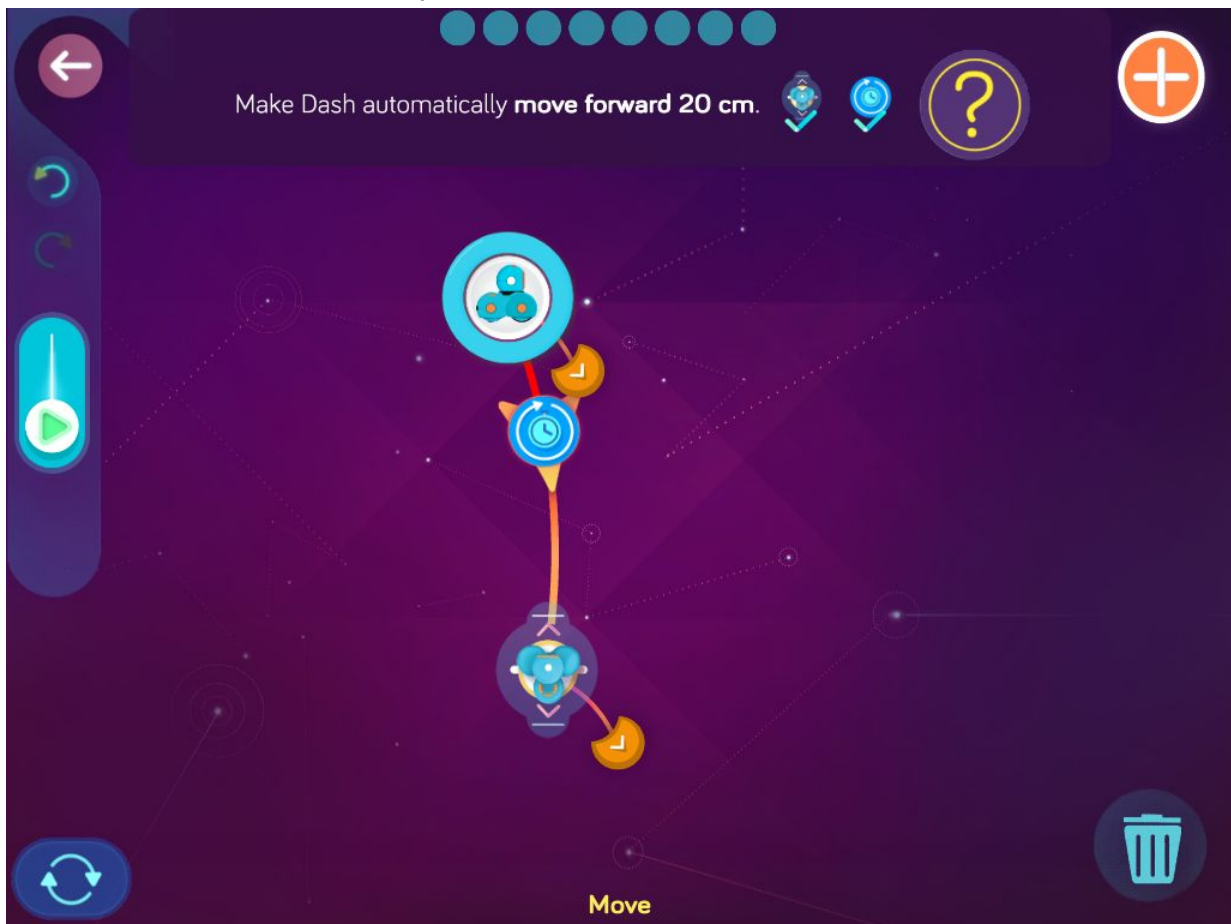
Butler Dash

Zone: Kong City

Challenge Progression 3

Solutions

Step 1: Make Dash automatically move forward 20 cm.



Step 2: After moving forward, make Dash randomly choose to spin -45 degrees. Then make Dash automatically move forward again.



Step 3: After moving forward, make Dash randomly choose to spin 45 degrees. Then make Dash automatically move forward again.

The image shows a Scratch script editor interface. At the top, there is a progress bar with 10 blue circles, the first two of which are checked. Below the progress bar, a text box contains the instruction: "After moving forward, make Dash randomly choose to spin 45 degrees. Then make Dash automatically move forward again." To the right of the text box are several icons, including a question mark and a character icon. The main workspace features a dark purple background with a grid and a central character icon. A script is being built with the following blocks: a "Move" block, a "Spin 45 degrees" block, and another "Move" block. The "Move" blocks are connected to the "Spin 45 degrees" block. The "Move" block at the bottom is highlighted with a yellow border. The word "Move" is written in yellow text below the highlighted block. On the left side, there is a vertical toolbar with a back arrow, a refresh icon, a play button, and a redo icon. On the right side, there is a trash can icon.

Step 4: After moving forward, make Dash randomly choose to spin 180 degrees. Then make Dash automatically move forward again.

The image shows a Scratch script editor interface. At the top, there is a progress bar with 10 circles, the first three of which are checked. Below the progress bar, a text box contains the instruction: "After moving forward, make Dash randomly choose to **spin to 180 degrees**. Then make Dash automatically move forward again." To the right of the text box are four small icons representing different block types: a robot head, a clock, a play button, and a question mark. The main workspace displays a script starting with a "Move" block, followed by a "Spin 180 degrees" block, and then another "Move" block. The "Move" block is currently selected. The interface includes a back arrow, a play button, a trash can, and a refresh button.

Step 5: While moving forward, make Dash smile if an obstacle is seen in front.

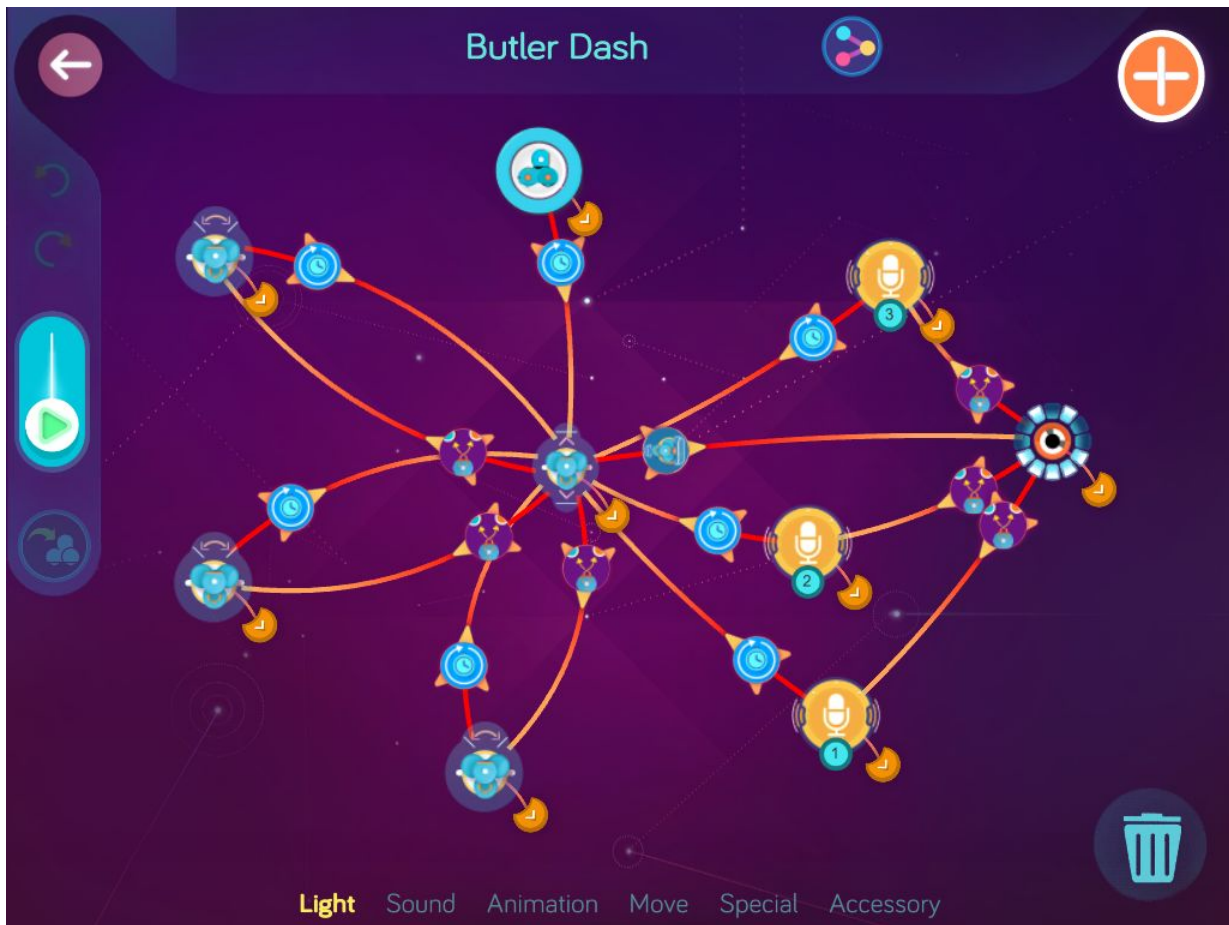
While moving forward, make Dash smile if an **obstacle is seen** in front.

Light Move

Step 6: Edit the custom sound to make Butler Dash offer some food.

The screenshot shows a programming environment with a character named Butler Dash. The interface includes a top navigation bar with a back arrow, a progress indicator (five blue checkmarks followed by three grey circles), and a character icon. A text box at the top reads: "Edit the **custom sound** to make Butler Dash offer some food." Below this is a palette of blocks, with a 'Sound' block highlighted. The main workspace contains a behavior tree diagram with a central character icon and several branches. A 'Sound' block is highlighted in the tree, and a '1' icon is visible below it. At the bottom, there are labels for 'Light', 'Sound', and 'Move', with 'Move' highlighted in yellow. A trash can icon is in the bottom right corner.

Step 8: After smiling, make Dash automatically choose to play another custom sound that offers a different kind of food or drink. Then make Dash automatically move forward again.



Wonder Key: inyq

Level 8: OMG Poppies, Ancient Robot

Concepts covered

- Program Dash using **Sleep** Animation
- Programming commands as a loop
- Programming a clap as a cue

Key terms

Loop: a set of instructions that a computer program repeats over and over again

Extension questions

1. What is a loop and how was it used to program OMG Poppies?
2. How was Dash programmed to sleep in OMG Poppies?
3. How was Dash programmed to awaken?
4. What is the difference between the 4 eye light patterns programmed in Ancient Robot?
5. Explain the steps to program Dash to turn off a series of 5 eye lights (one at a time) each time you clap?

OMG Poppies

Zone: Forgotten Jungle
Challenge Progression 1

Step 1: Make Dash automatically go to sleep then automatically turn the lights off. Then make Dash automatically turn off all eye lights.

The image shows a Scratch-style programming interface for a game. At the top, a text box contains the instruction: "Make Dash automatically go to **sleep** then automatically turn the **lights off**. Then make Dash automatically **turn off all eye lights**." Below this, a sequence of four actions is connected by arrows: 1. A 'Go to Sleep' block with a character icon. 2. A 'Turn Lights Off' block with a lightbulb icon. 3. A 'Turn Off All Eye Lights' block with an eye icon. 4. A 'Wait' block with a clock icon. The interface includes a toolbar on the left with a play button and a trash can on the right. At the bottom, there is a 'Light Animation' section with a lightbulb icon and an eye icon. The background is dark purple with a starry pattern.

Step 2: After turning off the eye lights, make Dash wake up when a clap is heard.

The image shows a Scratch script editor interface. At the top, there is a navigation bar with a back arrow, three progress indicators (the first is checked), and a help icon. Below the navigation bar, a text box contains the instruction: "After turning off the eye lights, make Dash wake up when a clap is heard." To the right of this text are several icons representing different Scratch blocks: a sound block (clap), a lightbulb icon, an eye icon, and a character icon (Dash). The main workspace shows a sequence of four Scratch blocks connected by arrows: 1. A "When green flag clicked" event block. 2. A "Wait 2 seconds" block. 3. A "Lightbulb" block with a downward arrow, indicating the light is turned off. 4. A "Character" block with a character icon (Dash) and a downward arrow, indicating the character wakes up. The background is a dark purple space with stars and a grid. At the bottom, there are icons for "Light" and "Animation" (highlighted in yellow), and a trash can icon.

Ancient Robot

Zone: Forgotten Jungle
Challenge Progression 2

Solutions

Step 1: *Make Dash turn on more eye lights each time you clap. Make Dash finish with a smile.*



Level 9: Square Dance, Desert Dash, Dune Buggy

Concepts covered

- Programming sound effects, e.g., car trouble
- Practicing commands and functions learned in Scroll Quest thus far.

Extension questions

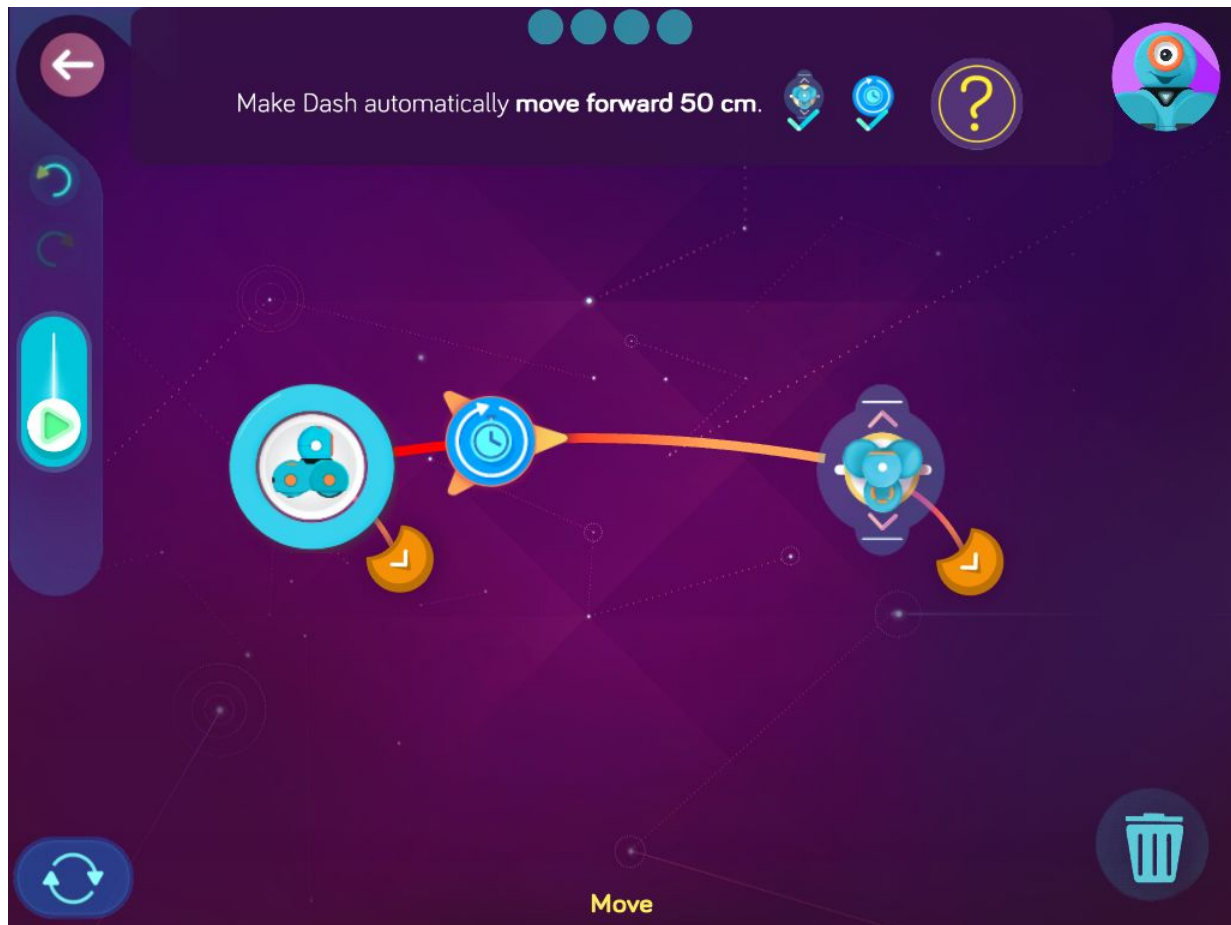
1. Which direction does Dash spin when set to 90 degrees? How can you reverse the spin?
2. Which commands in the Square Dance were part of a loop?
3. Describe two ways in which the top button was used as a cue in Desert Dash.
4. Describe the commands in the loop that were programmed for Dune Buggy.
5. In Step 4 of Dune Buggy, Dash is programmed to wait for 2.0 s and then stop. What other cue could have been used here in place of the wait for cue? How would it change this program?

Square Dance

Zone: Dry Gulch

Challenge Progression 1

Step 1: *Make Dash automatically move forward 50 cm.*



Step 2: After moving forward, make Dash spin to 90 degrees.

The image shows a programming interface for a character named Dash. At the top, there are four progress indicators, with the first one checked. Below them is a text instruction: "After moving forward, make Dash automatically spin to 90 degrees." To the right of the text are three icons representing different types of blocks: a character block, a movement block, and a spin block. A question mark icon and a character icon are also present in the top right.

The main workspace contains a sequence of five blocks connected by arrows, representing a script for Dash:

- 1. A character block (Dash) with a yellow arrow pointing right, indicating a movement action.
- 2. A spin block with a blue circular arrow icon, indicating a spin action.
- 3. A character block (Dash) with a yellow arrow pointing right, indicating a movement action.
- 4. A spin block with a blue circular arrow icon, indicating a spin action.
- 5. A character block (Dash) with a yellow arrow pointing right, indicating a movement action.

At the bottom, there is a "Move" label and a trash can icon. On the left side, there are several control icons: a back arrow, a refresh icon, a play button, and a circular arrow icon.

Step 3: After spinning right, make Dash automatically play a silly sound.

The screenshot shows the Scratch programming environment with a sequence of blocks for Dash the robot. The blocks are connected in a sequence: a 'When green flag clicked' block, a 'Spin right' block, a 'Move' block, another 'Spin right' block, another 'Move' block, and finally a 'Play sound' block. The 'Play sound' block is highlighted in yellow, indicating it is the current block being edited. The interface includes a top bar with a back arrow, a progress indicator (four circles, the first two are checked), a help icon (a question mark), and a Dash character icon. A vertical toolbar on the left contains a play button and a refresh button. A bottom toolbar contains a refresh button, a 'Sound' label, a 'Move' label, and a trash icon.

After spinning right, make Dash automatically play a silly sound.

Sound Move

Step 4: After the silly sound, make Dash automatically move forward 50 cm again.



Wonder Key: ffrg

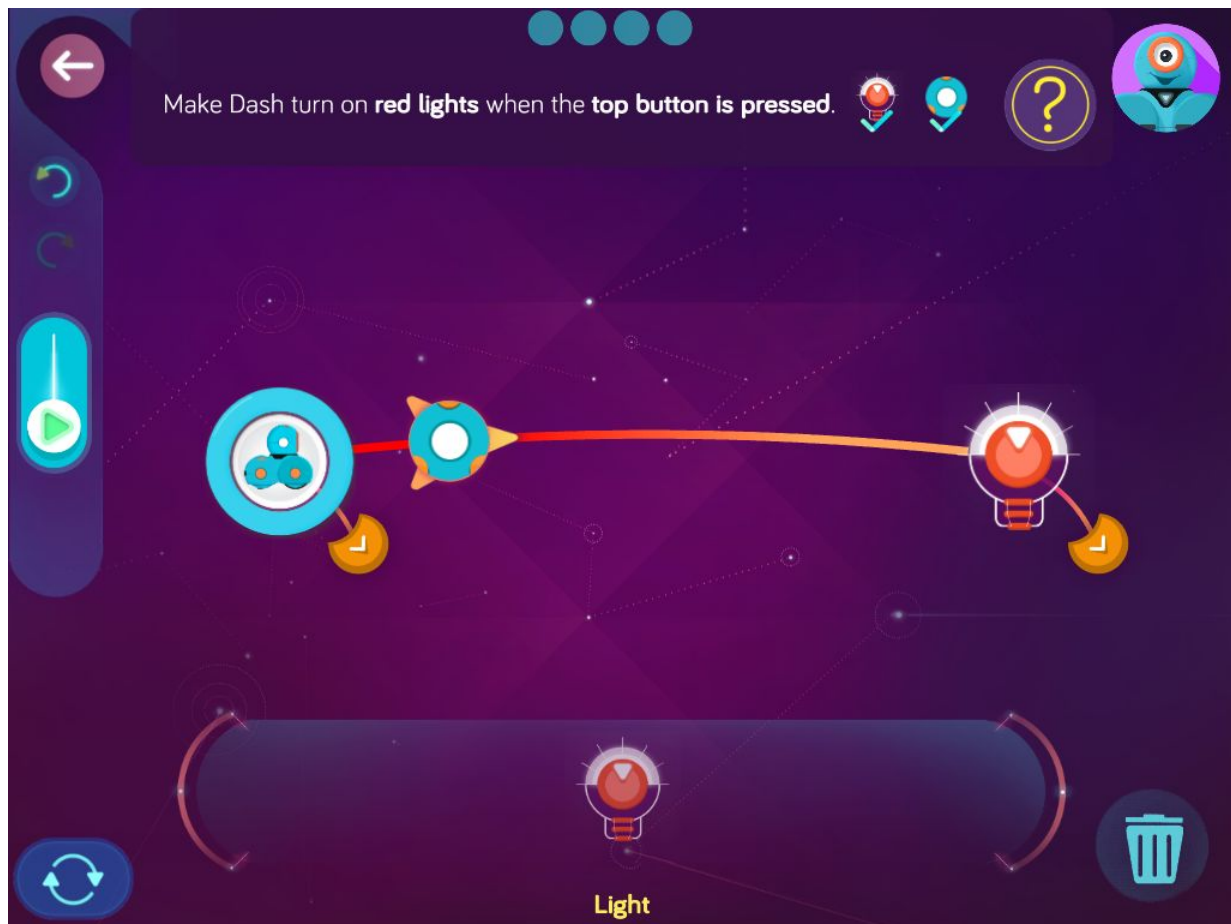
Desert Dash

Zone: Dry Gulch

Challenge Progression 2

Solutions

Step 1: Make Dash turn on red lights when the top button is pressed.



Step 2: After turning on red lights, make Dash automatically take off.

The image shows a Scratch script editor interface. At the top, a text box contains the instruction: "After turning on red lights, make Dash automatically take off." Below this, a script is being built on a dark purple background. The script consists of four blocks connected in a sequence:

- When green flag clicked** (Event block)
- Turn on red lights** (Light block)
- Wait 2 seconds** (Time block)
- Make Dash take off** (Animation block)

Each block has a small yellow 'L' icon at its bottom right corner, indicating it is a block that can be locked. The interface includes a top navigation bar with a back arrow, a progress indicator (four circles, the first is checked), and a help icon. On the left, there are icons for undo, redo, and a play button. At the bottom, there are icons for undo, redo, and a trash can. The text "Light Animation" is visible at the bottom center of the workspace.

Step 3: After taking off, make Dash turn on green lights when the top button is released. Then make Dash automatically play the tire squeal sound.

The image shows a programming interface for the Dash robot. At the top, there is a text instruction: "After taking off, make Dash turn on **green lights** when the **top button is released**. Then make Dash automatically play the **tire squeal** sound." Above this text are four status indicators: two green checkmarks and two blue circles. To the right of the text is a palette of blocks including a lightbulb, a sound icon, and a question mark. The main workspace contains a sequence of blocks connected by a red line: a "when top button released" block, a "turn on green lights" block, and a "play tire squeal sound" block. A "Dash" robot icon is positioned at the end of the sequence. On the left side, there is a vertical toolbar with a back arrow, a play button, and a refresh button. At the bottom, there is a legend with "Light" (green), "Sound" (yellow), and "Animation" (blue) categories, and a trash can icon on the right.

Step 4: After playing the tire squeal sound, make Dash instantly move forward with both wheels set to 80 cm/s.



Wonder Key: ymyr

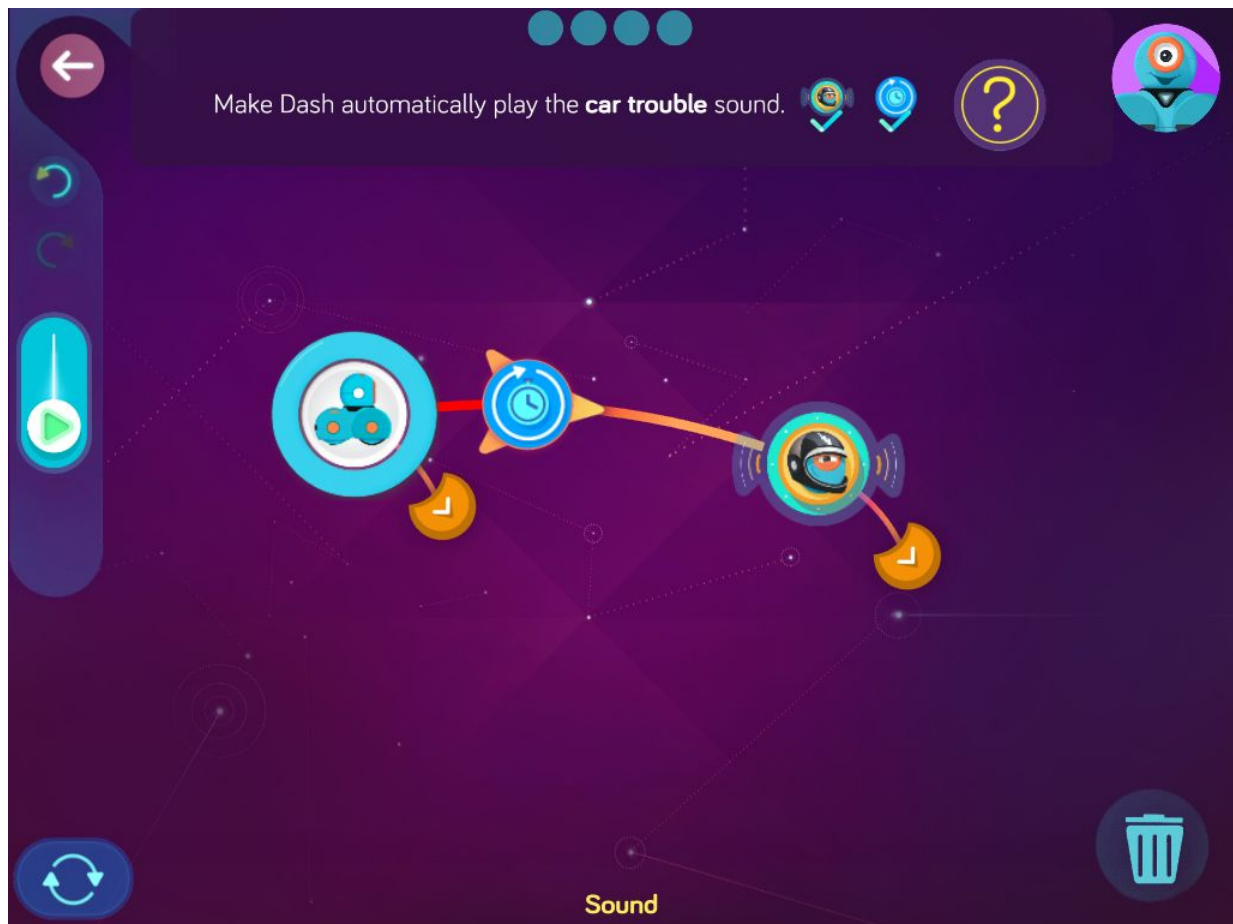
Dune Buggy

Zone: Dry Gulch

Challenge Progression 3

Solutions

Step 1: *Make Dash automatically play the car trouble sound.*



Step 2: After playing the **car trouble** sound, make Dash automatically return to **Start**.

The image shows a Scratch workspace with a dark purple background. At the top, there is a navigation bar with a back arrow, a progress indicator (four circles, the first is checked), and a user profile icon. Below the navigation bar, a text box contains the instruction: "After playing the **car trouble** sound, make Dash automatically return to **Start**." To the right of the text box are two checkmarks and a question mark icon. The main workspace features a character named Dash on the left. A sequence of events and actions is connected by a red line:

- A "Click" event block (yellow circle with a lightning bolt) is connected to a "Play sound" block (blue circle with a speaker icon).
- The "Play sound" block is connected to a "Wait" block (blue circle with a clock icon).
- The "Wait" block is connected to a "Go to Start" block (blue circle with a location pin icon).
- The "Go to Start" block is connected to a "Click" event block (yellow circle with a lightning bolt).
- The "Click" event block is connected to a "Play sound" block (blue circle with a speaker icon).
- The "Play sound" block is connected to a "Wait" block (blue circle with a clock icon).
- The "Wait" block is connected to a "Go to Start" block (blue circle with a location pin icon).

At the bottom of the workspace, there is a "Sound" label and a trash can icon.

Step 3: While playing the **car trouble** sound, make Dash **take off** when Dash **moves forward** or **moves backward**.

The image shows a Scratch script editor interface. At the top, there are four progress indicators: two green checkmarks and two blue circles. Below them, a row of icons shows various Scratch blocks with checkmarks, and a yellow question mark icon. The main workspace contains a script for a character named Dash. The script starts with a 'When green flag clicked' event block, followed by a 'Play sound' block (labeled 'car trouble'). This is followed by two 'When Dash moves' blocks: 'When Dash moves forward' and 'When Dash moves backward'. Each of these blocks is connected to an 'Animation' block labeled 'Take off'. The script ends with a 'When Dash moves' block labeled 'When Dash moves' connected to a 'Play sound' block labeled 'car trouble'. The interface includes a left sidebar with navigation icons (back, forward, undo, redo, play, volume), a bottom toolbar with a refresh icon, and a bottom right corner with a trash icon. The text 'Sound Animation' is visible at the bottom of the workspace.

Level 10: Itchy Lion

Concepts covered

Programming Dash to turn toward your voice when you speak

Programming Dash to sense obstacles seen from close behind and cue the next state

Key Terms

Obstacle: an object or person blocking Dash in front or behind.

Extension questions

1. How did you program the obstacle cue to detect what was close behind Dash?
2. Can you think of a situation in which this cue would be helpful in a program?
3. Which animal sound is NOT included in the animal sound menu that you think should be?
4. How could you record this animal sound to add to your own program?
5. In the final step of Itchy Lion, Dash instantly Dances Left. What would happen if Dash had been programmed to automatically Dance Left? How are the instant and auto cues different? Explain.

Itchy Lion

Zone: Big Cat Canyon
Challenge Progression 1

Step 1: Make Dash automatically turn on **yellow lights** and automatically play the **lion** sound. Then make Dash instantly show the pattern in the **eye ring**.

The image shows a Scratch-style programming interface for a challenge. At the top, a text box contains the instructions: "Make Dash automatically turn on **yellow lights** and automatically play the **lion** sound. Then make Dash instantly show the pattern in the **eye ring**." To the right of the text are several icons: a lightbulb, a lion sound icon, an eye ring icon, a timer icon, and a question mark icon. Below the text is a sequence of five Scratch-style blocks connected by a red line. The first block is a 'When green flag clicked' block. The second block is a 'When green flag clicked' block with a clock icon. The third block is a 'Turn on yellow lights' block with a lightbulb icon. The fourth block is a 'Play lion sound' block with a lion icon. The fifth block is a 'Show eye ring pattern' block with an eye ring icon. At the bottom, there are two icons: a lightbulb labeled 'Light' and an eye ring labeled 'Sound'. A trash can icon is also visible at the bottom right.

Step 2: After the eye ring, make Dash turn to your voice when you speak.

The image shows a Scratch script editor interface with a dark purple background. At the top, there is a navigation bar with a back arrow, a progress indicator (a checkmark followed by four circles), and a character icon. Below the navigation bar, a text box contains the instruction: "After the **eye ring**, make Dash to **turn to your voice** when you speak." To the right of the text box are several icons representing different Scratch blocks: a lightbulb (Thought), a speech bubble (Say), a cat face (Say), a globe (Go to URL), a speech bubble with a checkmark (Say), a speech bubble with a checkmark (Say), a globe with a checkmark (Go to URL), and a question mark icon. The main workspace features a series of connected Scratch blocks: a 'Say' block with a cat face, a 'Say' block with a speech bubble, a 'Thought' block with a lightbulb, a 'Say' block with a speech bubble, a 'Say' block with a cat face, a 'Say' block with a speech bubble, and a 'Say' block with a cat face. A 'Go to URL' block is also present. The workspace is decorated with a grid of dotted lines and small icons. At the bottom, there are labels for 'Light', 'Sound', and 'Move' in yellow text, and a trash can icon.

Step 3: After Dash turns to the voice, make Dash automatically move forward 30 cm. Then make Dash automatically play another lion sound.



The screenshot displays the Scratch-like programming environment for the Dash robot. At the top, a progress bar shows five steps, with the first two completed. The instruction text reads: "After Dash turns to the voice, make Dash automatically **move forward 30 cm**. Then make Dash automatically play another **lion** sound." The workspace contains a sequence of blocks: a "Turn to Voice" block, a "Move Forward 30 cm" block, and a "Play Sound" block with a lion icon. A "Lion" sound icon is also visible in the top-right block palette. The interface includes a left sidebar with navigation and play buttons, a bottom status bar with "Light", "Sound", and "Move" categories, and a trash icon in the bottom right.

Step 4: After Dash plays the second lion sound, make Dash turn on magenta lights when an obstacle is seen close behind.

The image shows a Scratch script editor interface. At the top, there are five progress indicators: three green checkmarks and two blue circles. Below them, a list of available blocks is shown, with a magnifying glass over the 'When obstacle is seen close behind' block. The main workspace contains two script areas. The top script starts with a 'When green flag is clicked' event block, followed by a 'Play sound' block (lion sound), a 'Turn on light' block (magenta light), a 'Play sound' block (lion sound), a 'Turn on light' block (magenta light), a 'Play sound' block (lion sound), a 'Turn on light' block (magenta light), and a 'Play sound' block (lion sound). The bottom script starts with a 'When green flag is clicked' event block, followed by a 'Play sound' block (lion sound), a 'Turn on light' block (magenta light), a 'Play sound' block (lion sound), a 'Turn on light' block (magenta light), a 'Play sound' block (lion sound), a 'Turn on light' block (magenta light), and a 'Play sound' block (lion sound). The interface includes a left sidebar with navigation and playback controls, a bottom toolbar with 'Light', 'Sound', and 'Move' categories, and a trash icon in the bottom right corner.

Level 11: Dashlight, The Listener, Reacher Creature, Ogre of Darkness

Concepts covered

Review of concepts covered in previous Scroll Quest Challenges:

Turning lights on and off

Turning eye lights on and off

Programming a spin using wheel differential

Programming using the top button: pushed or released as a cue

Programming the Listener cue

New Concepts:

Programming **Search** cue

Programming the Listener to **Do Nothing**

Key Terms

Listener: this state waits until a cue causes it to put Dash into a new state.

Extension questions

1. In Step 2 of DASHLIGHT, you programmed Dash to spin by setting the left wheel to -20 cm/s and the right wheel set to 20 cm/s. What would have happened if you had set the left wheel to -20 and the right wheel to 40? Explain the results.
2. How would you rate Dash's speed at 20 cm/s? Fast? Slow? In between?
3. In The Listener, which cue prompted the Listener to move onto a new state?
4. In Step 3 of The Listener, when was the proper time to speak, so that Dash could hear you? (When the robot was paused. Dash cannot hear voice commands when its motor is running.)
5. In Step 3 of Reacher Creature, what was the purpose of the loop? Explain.
6. In Ogre of Darkness share some of the ways you programmed Dash to appear to be sleeping.

Dashlight

Zone: Castle Creepenstein

Challenge Progression 1

Step 1: Make Dash automatically turn the lights off then automatically turn off all eye lights.

The screenshot shows the 'Dashlight' challenge interface. At the top, a progress bar consists of five blue circles, with the first one filled. Below it, a back arrow is on the left, and a character icon is on the right. The main instruction reads: 'Make Dash automatically turn the **lights off** then automatically **turn off all eye lights**.' To the right of the text are three icons: a lightbulb, an eye, and a coin, each with a checkmark. A yellow question mark icon is also present. The central workspace features a sequence of four icons connected by a red line: a blue robot head (Dash), a blue clock icon, a grey lightbulb icon, another blue clock icon, and a blue eye icon. Each icon has a yellow L-shaped cursor below it. On the left side, there is a vertical toolbar with a play button and a refresh button. At the bottom, there is a trash can icon and the word 'Light' in yellow text.

Step 2: After turning off the eye lights, make Dash automatically say, "Oh no!". Then make Dash instantly start spinning with the left wheel set to -20 cm/s and the right wheel set to 20 cm/s.

After turning off the eye lights, make Dash automatically say "Oh no!". Then make Dash **instantly** start **spinning** with the left wheel set to -20 cm/s and the right wheel set to 20 cm/s.

Light Sound **Move**

Step 3: After spinning, make Dash turn on all eye lights when the top button is pressed.

After spinning, make Dash **turn on all eye lights** when the **top button** is pressed.

Light Sound Move

Step 4: After turning on the eye lights, make Dash instantly move forward with both wheels set to 20 cm/s.

After turning on the eye lights, make Dash instantly **move forward with both wheels set to 20 cm/s.**

Light Sound **Move**

Step 5: After moving forward, make Dash turn the lights off again when the top button is released.

After moving forward, make Dash turn the **lights off** again when the **top button** is released.

Light Sound Move

Step 2: After stopping, make Dash wait for 2.0 s then search again.

The image shows a programming interface with a dark purple background. At the top, there is a navigation bar with a back arrow, a progress indicator (four circles, the first is checked), and a character icon. Below the navigation bar, a text box contains the instruction: "After stopping, make Dash **wait for 2.0 s** then **search** again." To the right of the text box are several icons: a character, a lightbulb, a crossed-out character, a checkmark, and a question mark. The main workspace features a sequence of actions connected by orange arrows. The sequence starts with a character icon, followed by a blue circular icon with a clock, then a character icon with a hat, then another blue circular icon with a clock, then a lightbulb icon, then a blue circular icon with a clock, then a blue circular icon with a clock and the text "2.0s" in pink, and finally a blue circular icon with a crossed-out character. At the bottom, there are three tabs: "Light", "Animation", and "Move", with "Animation" selected. A trash can icon is located in the bottom right corner.

Step 3: Make Dash use the *Listener* to turn the lights off when a voice is heard. Then make Dash automatically say, "Oh no!" and turn off all eye lights.

The screenshot shows the Dash robot programming interface. At the top, there are four status indicators: two green checkmarks and two blue circles. Below them, a grid of available blocks is shown, including a lightbulb, a speaker, a lightbulb with a slash, and a lightbulb with a checkmark. The main workspace contains a sequence of blocks connected by orange lines. The sequence starts with a 'Listener' block (lightbulb with a checkmark), followed by a 'Say' block (speaker), then a '2.0s' delay block (lightbulb with a checkmark), and finally a 'Turn off all eye lights' block (lightbulb with a slash). A '2.0s' delay block is also placed between the 'Say' and 'Turn off all eye lights' blocks. The interface includes a left sidebar with navigation icons (back, forward, play, volume), a top right corner with a question mark and a plus sign, and a bottom right corner with a trash can icon. At the bottom, a category menu shows 'Light', 'Sound', 'Animation', 'Move', and 'Special', with 'Move' highlighted in yellow.

Reacher Creature

Zone: Castle Creepenstein

Challenge Progression 3

Step 1: Make Dash automatically play the Whistle #1 sound then instantly move forward with both wheels set to 15 cm/s.

The screenshot shows the Dash robot programming interface. At the top, a task instruction reads: "Make Dash automatically play the Whistle #1 sound then instantly move forward with both wheels set to 15 cm/s." The interface includes a navigation bar with a back arrow, a progress indicator (three blue circles), and a help icon (a question mark). The main workspace contains a sequence of four programming blocks connected by arrows: 1. A "Play Sound" block with a whistle icon. 2. A "Wait" block with a clock icon. 3. A "Move Forward" block with a speedometer icon. 4. A "Move Forward" block with a timer icon showing "0:00". A legend at the bottom indicates "Sound" and "Move" block types. A trash can icon is located in the bottom right corner.

Step 3: After pushing to get away, make Dash stop if Dash is put down. Then make Dash automatically say "Ahh" and play the Whistle #1 sound again.

After pushing to get away, make Dash **stop** if Dash is **put down**. Then make Dash automatically say **Ahh** and play the **Whistle #1** sound again.

Sound Animation Move

Wonder Key: tso4

Ogre of Darkness

Zone: Castle Creepenstein

Challenge Progression 4

Step 1: Make Dash automatically turn on **red lights**, show a pattern in the **eye ring**, play the **growl** sound, and **move forward**.

The screenshot shows a programming interface for a challenge. At the top, there are five blue progress indicators. Below them is a text box with the instruction: "Make Dash automatically turn on **red lights**, show a pattern in the **eye ring**, play the **growl** sound, and **move forward**." To the right of the text box are four icons representing different actions: a red light, a blue eye ring, a growl sound, and a forward movement. A question mark icon and a plus sign icon are also present. Below the text box is a sequence of six programming blocks connected by arrows, representing the solution to the challenge. The blocks are: 1. A 'when green flag clicked' block. 2. A 'set red lights to on' block. 3. A 'set eye ring to [red light pattern]' block. 4. A 'play growl sound' block. 5. A 'move forward' block. 6. A 'say for 2 seconds' block. At the bottom of the interface, there are labels for 'Light', 'Sound', and 'Animation'. A trash can icon is located in the bottom right corner.

Step 2: After moving forward, make Dash automatically turn on red lights again.

After moving forward, make Dash automatically turn on **red lights** again.

Light Sound Animation

Step 3: Make Dash use the listener to do nothing when an obstacle is seen close behind Dash.

Make Dash use the **listener** to **do nothing** when an **obstacle is seen** close behind Dash.

Light Sound Animation **Special**

Step 5: Make Dash turn off an eye light every 1.0 s. Make Dash finish with all of the eye lights turned off.

The image shows a programming environment with a dark purple background. At the top, there are five status indicators: four green checkmarks and one blue circle. Below them is a toolbar with icons for undo, redo, and a question mark. The main workspace contains a character named Dash with two eyes. A script is being built in the script area, consisting of several blocks: a 'when green flag clicked' block, a 'say' block, a 'wait 2.0s' block, a 'turn off eye light' block, and a 'wait 1.0s' block. The 'turn off eye light' block is highlighted. The palette at the bottom has categories: Light, Sound, Animation, and Special. A trash can icon is in the bottom right corner.

Make Dash turn off an eye light every 1.0 s.
Make Dash finish with all of the eye lights turned off.

Light Sound Animation Special

