LUMA

BY PITSCO EDUCATION

TEACHER'S GUIDE

Welcome to LUMA!

LUMA is a programmable robot that's ready to go out of the box! As students work alongside LUMA, they will have the opportunity to explore the real-world applications of robotics and coding. Not only will students learn the basics of coding and computer science, but they will also be introduced to a variety of careers as they program the robot to complete tasks such as move, draw, flash lights, and more. These careers are not your traditional professions associated with robotics; students will learn how robots can help restaurants, architecture firms, events companies, and many more. As a culminating project, students will decide which industry is the best fit for LUMA's skills, combining what they've learned with their persuasive abilities.

LUMA's resources make it easy for students to work through the content. Video instructions are provided for students in the web app at **luma.pitsco.com**. Students will be guided through each of the Activities, and additional resources are provided as well.

We're excited for you and your students to experience LUMA. Let's get started!

Content

All LUMA content modules are delivered to students through LUMA's programming app and are arranged in Units. Content module types include Lessons, Activities, Explorations, and Challenges. It is recommended students complete the content modules in the order they are presented in the Pacing Guide on page 11 because concepts and skills will build from module to module.

- Lessons – Lessons are slide shows that introduce students to computer science concepts and skills. These can be presented by the educator or viewed by the student. Teachers can download the PowerPoint files at **Pitsco.com/LUMA**. Students can access and download the files through the Resources section of the app. These can be referenced at any time in a student's journey through the content. Refer to the Pacing Guide on page 11 to see how Lessons can be used to introduce concepts that are used in Activities and Challenges.

- Activities – Activities guide students through building specific LUMA programs that apply computer science concepts and skills. They are presented through videos in the pull-out Activity Panel in the app. Students are instructed to save the program files they build to their device. Activities should be completed in order because some programs from previous Activities will be reloaded and used in later programs.
- Explorations – Explorations are open-ended time for students to try their own ideas and create their own programs. Some Explorations are wide open while others are themed to help students explore ways robotics, technology, and computer science can benefit society.
- Challenges – Challenges present students with a problem to solve using LUMA. They are designed to be completed in teams but can be done individually as well. When completing a Challenge, students are given specific criteria and constraints that must be followed. All Challenges contain the same four objectives:
    - Plan – Students plan out their solution to the Challenge following the criteria and constraints.
    - Program – Students create their program using the LUMA App.
    - Put to the test – Students test their solution to the Challenge and make modifications to their program as needed.
    - Prove – Students present their solution to others.

LUMA's content also comes with a Coding Journal for students to complete as they work through Activities, Explorations, and Challenges. The Coding Journal is a place for students to take notes, answer questions, plan Challenges, and reflect on their learning. The Coding Journal is provided in two formats: an editable PDF and a PowerPoint file. The journal can be printed and distributed to students or completed using digital tools.

Programming App Sessions and Codes

Students use LUMA's programming app to view content and build their programs. When a student launches a session in the app, they will get a unique code at the top of the screen. This code is used to track the student's progress through the content as well as the current program they are working on without having to log in and track any identifying data. At the beginning of their first session, students should write this code down. The accompanying Coding Journal provides a location to do this on the first page. When students are ready to continue their session, they should click **I have a code** in the app and enter their code. Their progress through the content and their last program will be recalled.

Characters and Careers

As students complete the LUMA content, they will meet different animated characters. LUMA's inventors are MJ Livingston and Sharon Harvey; they guide students through the basics of using the robot and the programming app. Every Activity is presented by a different character who introduces students to their career and company. Each character wants to use LUMA to help them in their job. As students complete Activities, they will reflect on different tasks and jobs the robot could complete to help that character in their career field. After completing all Activities, students are asked to pick which career field they think LUMA fits best and explain their reasoning. For more information, see the Culminating Project section in this guide.

Pacing Guide

MODULE	TIME (MIN)
Impacts of Technology	25
Hardware and Software	15
Computing Devices	15
Activity 1 – Hello LUMA	90
Keeping a Coding Journal	20
Activity 2 – Getting Around	120
Challenge – A Maze Zing Drive Challenge	120
Loops – For Loop, True Loops	15
Algorithms & Pseudocode	15
Troubleshooting and Debugging	15
Activity 3 – Light It Up	120
Exploration: Light Work	90
Activity 4 – Sounding Off	120
Exploration: Environmentally Sound	90
Decomposing Problems	20
Simplifying Code	15
Activity 5 – Drawing Lines	120
Team Coding	20
Challenge – Robotic Artist	120
Senses and Sensors	15
Conditionals-If Statements	15
Variables	20
Activity 6 – Toeing	120
Exploration: Sensing Fun	90
Making Technology Accessible	10
While Loops	10
Activity 7 – Pressing My Buttons (HCI)	120
Exploration: Pressing On	90
Data Gathering	15
Activity 8 – Seeing Is Believing	120
The Design Process	25
Challenge: Follow the Leader	120
Reusing Code and Attribution	10
Activity 9 – Obstacle Detection	120
Exploration: Going the Distance	90
Activity 10 – Ultrasonic Theremin	120
Visual Data	30
Presentations	25
Challenge: LUMA Is on the Job	120

Classroom Implementation

LUMA and its content are designed to be flexible and can be implemented in a multitude of different ways. You'll need to determine what works best for you and your students. Here are a few examples:

- LUMA in the classroom
    - 1-to-1: One LUMA per student working individually
    - Pairs: One LUMA per pair with students working collaboratively
    - Teams: Small groups of students working collaboratively
    - Combinations: Students could work individually through Lessons and Activities but work collaboratively on Explorations and Challenges.
    - Whole Class: Present the content from a projector or TV and walk through each Lesson and Activity together.
- LUMA in makerspaces and media centers
- LUMA in after-school programs
- LUMA in summer camps
- LUMA at home

Best Practices

- **Access to LUMA App** – Bookmark or favorite the web app so students can easily navigate to it.
- **Headphones** – When possible, students should use headphones when viewing video and audio content in the app to keep the classroom noise level to a minimum.
- **Whole-Class Lessons** – Lessons are different from the rest of LUMA's content because they are accessed through the Resources section of the app and do not follow in sequence of the Activities, Explorations, and Challenges. Because of this, consider presenting Lessons as whole-class presentations to everyone at the same time. This allows you to add your own teaching instruction, add clarification to confusing concepts, hold class discussions, and complete unplugged Activities together.
- **Activity Mini Challenges** – Some students will complete Activities more quickly than others. Every Activity concludes with a Mini Challenge. Have students who finish early work on these Mini Challenges to extend their learning and keep them busy.
- **Don't Be the Expert** – Try not to answer students' questions or solve their problems. Rather, when asked a question, respond with a question to help guide them to the answer. When troubleshooting and debugging code, be slow to point out errors. Let them fail and learn from their failures. Help students build grit and determination and experience the satisfaction of solving their own problems. Point them where to look and provide clues as frustration builds. Let other students who have had success guide students who are struggling. Jump in with answers and solutions as a last resort.
- **Identifying LUMAs** – If using multiple LUMAs in a classroom, it might be helpful to number them using a permanent marker on the underside of the chassis so students can use the same robot from day to day.
- **Use LUMA on the Ground** – Whenever LUMA is programmed to move, always place LUMA on the ground.
- **Create a Testing Area** – Consider setting up a designated area for students to test their LUMA robots. The area needs to be wide open and accessible to all. Mark off the area to create boundaries using painter's tape. The testing area might need to be modified for different Challenges. In larger classrooms, consider setting up multiple smaller testing areas. For example, you could designate each corner of the room as a testing area.

- **Challenge Groups** – Have students complete Challenges in pairs or small groups to give them a chance to build collaboration and communication skills. It is recommended to keep group size to a max of three students to keep all students engaged. When putting students in groups, consider breaking up students who are strong coders.
- **Team Roles** – When working in pairs or small groups, give students different roles. Have students rotate through different roles from project to project. Here are some examples of team roles:
    - **Team Lead** – In charge of making sure all aspects of a Challenge are understood, leads the team through planning, and keeps the team on task while completing the Challenge.
    - **Lead Programmer** – In charge of building the program in the app while incorporating feedback from others.
    - **Hardware Manager** – In charge of handling LUMA, starting and stopping programs, and connecting LUMA to the device.
    - **Engineer** – In charge of incorporating feedback from others to build attachments, tools, and decorations on LUMA using LEGO® Technic Beams.
- **LUMA's Shell** – It's not recommended to remove LUMA's shell unless necessary for maintenance. If the shell needs to be removed, it should be removed by an adult following the steps in the *LUMA User Guide*.
- **Battery Life** – Make sure LUMA is turned off when not in use. Turning on LUMA's pixel LEDs at 100 percent brightness will drain the batteries more quickly. When possible, use a brightness of 50 percent or less to preserve battery life. Consider using rechargeable AA batteries if LUMA is being used multiple hours in the day.
- **Coding Journal** – Distribute the Coding Journals to each student so they can keep notes, answer questions presented in the content, and reflect on what they've learned as they progress through the different Activities.
- **LUMA's Tools** – LUMA comes with two Allen wrenches for tightening the screws that hold LUMA together. It is recommended to collect these tools so students are not tempted to use them to take LUMA apart.
- **Drawing with LUMA** – When drawing with LUMA, it is recommended to:
    - Use a marker that fits snugly in LUMA's center hole such as a fine-point dry-erase marker. The marker should contact the drawing surface without a lot of wiggle room in the center hole.
    - You might want to avoid having students use permanent markers. It is easy to mark up LUMA's shell or chassis as the marker is inserted in the center hole. Non-permanent and dry-erase markers work well and should wipe right off of LUMA's shell.
    - If you decide to use a permanent marker, use caution when inserting and removing the marker.
    - Place a few extra layers of paper down to catch any ink that bleeds through the paper.
    - Tape paper down so it doesn't move as LUMA is drawing.
    - Use a large dry-erase board with dry-erase markers if possible.

Activities Overview

Activity 1 – Hello LUMA

Overview

In this Activity, students will work with Losaline Tuiaki, a game designer. Students will learn and practice uploading, running, and saving code on LUMA. For the first three Tasks, there is no actual programming on the learner's part. All programs are provided to them via the Activity Panel in the LUMA App. Students will be able to see the robot in action with its sensors, movement, and lights. In the final Task of this Activity, students will have an opportunity to create basic code as well as modify it.

Learning Objectives:

- Understanding the coding environment
- Transferring code to the robot
- Running code on the robot
- Saving code on the robot
- Creating and modifying basic code

Assessment

The main focus of this Activity is for students to understand how to upload, run, and save code using LUMA. By the end of the Activity, students should be comfortable doing these actions independently.

Answer Key

Task 1

What do you think the first demo code will do?

Possible answers include comments about LUMA moving, detecting objects, and flashing lights.

(the following questions are answered on computer)

Does LUMA always pick the clearest path?

The following answer is a very detailed response. Students' responses will vary based on their understanding and skill level.

LUMA uses its ultrasonic sensor to detect the nearest object in both the left and right directions. It then flashes its light to indicate which direction has the farthest distance to an obstacle. However, LUMA might have difficulty detecting some objects if there is no flat edge to reflect the ultrasonic pulse back to the sensor. Round objects or objects placed at an angle to LUMA can cause the sensor to not detect the object. So depending on the obstacle, LUMA might not always pick the clearest path.

What happens if LUMA approaches an obstacle at an angle?

The following answer is a very detailed response. Students' responses will vary based on their understanding and skill level.

LUMA might have trouble detecting the object if the obstacle is at an angle. LUMA's ultrasonic sensor emits a pulse of sound that cannot be heard by human ears. The sound pulse reflects off the obstacle and back to the sensor. This is how LUMA knows something is in front of it. But, if the object is at an angle, the pulse might not reflect back to the sensor, causing LUMA to miss the obstacle. This is similar to how stealth technology works.

Task 2

What did LUMA draw? What do you think of it?

Possible answers include a smiley face and students' opinion about the drawing.

Task 4

List four to five messages you want to share using lights.

Students should have a list of four to five messages as well as the associated light pattern. For example: Meet at the door = blink red on right three times.

Differentiation

If students need additional support during the final Task, create one to two messages with the students and demonstrate how to change the code to display the message with lights.

Activity 2 – Getting Around

Overview

In Activity 2, students will join restaurant owner and chef Raul Bustamonte in his new pizza restaurant, Take a Slice. Students will learn how to navigate the robot around the restaurant by programming it to go forward and backward as well as make various types of turns. Activity 2 will culminate in a Challenge with students using LUMA to navigate a course while avoiding obstacles.

Learning Objectives:

- Changing the speed of the robot
- Moving the robot forward and backward with basic navigational commands
- Stopping the robot
- Making rotation turns
- Finding multiple solutions to a coding problem

Assessment

Students can be assessed on how well they are able to complete the final Task. Are they able to code their robot to navigate all of the obstacles? Are they accurately using all four blocks as given in the Challenge: a **Program Start block**, **motor speed blocks**, **wait blocks**, and a **Program End block**?

Answer Key

Task 1

What other types of movements should a restaurant robot have?

Possible answers could include cleaning floors, serving pizzas, washing dishes, seating customers, and so on.

Task 2

If LUMA drives in reverse at half the speed it drives forward, how many seconds do you think it will take for it to return where it started?

If LUMA drives forward for five seconds and drives in reverse at half the speed, it will take 10 seconds for LUMA to return to its starting point.

Task 3

Can you guess what shape LUMA will drive when this program is run?

LUMA will drive in a triangle. However, if a student guesses another shape with an appropriate rationale, it could also be considered correct.

*What would LUMA do if you changed the three **move for distance blocks** to 50 centimeters instead of 100 centimeters?*

LUMA would move in a small triangular pattern.

What would happen if you changed the direction of spin from clockwise to counterclockwise on each ***spin*** ***block?***
LUMA would move in the opposite direction.

How would you need to change the code to drive in a square?
The angles of the turns would need to change to 90 degrees and an extra side would need to be added in.

Task 4
How could you change the code you just created to move LUMA in a rectangle?
By changing the code for two parallel sides to a shorter or longer length.

Task 5
What are three examples of when you could use LUMA to move in a circle or arc at Take a Slice?
Possible answers: spreading sauce, making a round pizza crust, moving objects from one point to another on a pivot, passing materials to customers such as napkins, menus, and so on.

Differentiation
Based on the students' ability levels, the Challenge can be made more complex or simplified by creating a course with more or less obstacles, respectively.

A Maze Zing Drive Challenge

Overview
Students program LUMA to complete a maze. The robot must begin at the starting line of the maze and find its way to the finish line without crashing into any of the maze walls or cross any of the maze boundaries.

Assessment
Students should successfully program LUMA to navigate the maze without crashing into any walls or crossing any maze boundaries. Students should also document their plan for the robot to complete the maze in their Coding Journal.

Differentiation
For students needing an extra challenge, refer students to the Bonus Points section of their Challenge.

For students needing extra support, consider an easier maze to complete.

Activity 3 – Light It Up

Overview
Students will work with Peggy Samuelson, an event specialist, in Activity 3 as they learn to set the color and brightness of the controller's RGB LEDs. Through a variety of tasks in association with planning a large event, they will learn to flash the LEDs in different patterns as well as using lights and movement together. The Challenge for this Activity will be to create their own flashing light show.

Learning Objectives:
- Controlling the pixel LEDs on LUMA
- Communicating information using indicators
- Understanding algorithms
- Using pseudocode to plan a program
- Troubleshooting and debugging a program

Assessment
The most natural assessment for this Activity will be the final Task. How well are students able to

successfully complete the Task based on the criteria provided of creating a light show and incorporating in an element of creativity? While creativity can be a challenging element to grade, it can be more concrete by requiring students to share their planning process as well as an explanation as to why they feel they met the requirements for this Challenge.

Answer Key

Task 1 (answered within the learning platform)

What problem do you see with the code as it's written?

Answers may vary, but should include the light will turn on and off too quickly to be seen.

Task 2

After you've tested the code, evaluate how it did. What can you do to make this code even better?

Answers may vary. Possible answers include changing the length of time for each light or repeating the pattern multiple times.

Task 3

Write pseudocode to program LUMA's front two pixels to flash lights of two different colors back and forth continuously. Remember, pseudocode has no rules other than to write it in a way that makes sense to you.

The nature of pseudocode is that it makes sense to the author of it to help with programming. So, students may have a wide variety of responses. A possible pseudocode could be:

```
Turn on light 1
Wait for .5 seconds
Turn off light 1
Turn on light 2
Wait for .5 seconds
Turn off light 2
Repeat
```

Write two to three sentences about how well your code matches the pseudocode you created. Were they the same? Were there any parts you didn't include but should have?

Students should be able to accurately reflect on their pseudocode and how well it matches their code on-screen.

Differentiation

For students who have not yet mastered the skills in this Activity, it may be helpful to provide a coding framework where they can plug in various elements. For example, giving students the blocks of code that they'll need or providing the first few blocks of code to get them started.

Exploration: Light Work

Overview

This Exploration is an open-ended opportunity for students to explore LUMA and its programming capabilities. Students should be encouraged to select a specific task to complete and some sample ideas are given to students.

Idea Starters:

- Program LUMA to act out a story
- Program LUMA to do a specific task or job
- Program LUMA to express your feelings
- Program LUMA to help someone else

Activity 4 – Sounding Off

Overview

In this Activity, students work with sound engineer Arif Noori to learn a few basics around LUMA's sound capabilities. Students will learn how to program the robot to produce sounds for a variety of situations such as a truck backing up, creating a catchy tune, and alerting a user that their food is finished cooking. Some sounds will be built in and chosen from a drop-down or imported function from the library. Students will also learn how to play individual notes or tones. For their final Challenge, students will take the knowledge they've learned in the Activity to create a new jingle for an ice cream truck in order to attract customers. While the jingle is played, the LED lights will also be incorporated to increase customer interest.

Learning Objectives:

- Producing sounds using the robot controller
- Manipulating tones using the speaker buzzer to produce music
- Storing a value in a variable
- Performing arithmetic operations in code
- Using for loops

Assessment

Students should be assessed on how closely they met the criteria of the Challenge. The jingle should be four to seven seconds long and the LEDs should be incorporated into the jingle. Students can also be graded on the planning process and how well they can reflect on the differences between their plan and the actual results. Students should be able to accurately identify what worked and what didn't work and the changes they made as a result.

Answer Key

Task 1 (answered within the learning platform)

Think about the coding you just completed. What would be the best answer to these questions?

About how many seconds does LUMA back up? 10 seconds

How could you change the distance that LUMA backs up? There are at least three ways without adding any new blocks to the program.

1. Change the move speed
2. Change the wait time
3. Change the number of times the loop repeats

Task 2

What are three other ways you could use a similar program with your customers?

Answers will vary, but should include reasonable ways that music could be incorporated into various tasks that would appeal to customers. Some possible answers include incorporating music or tones into crosswalk signals, alerting users with music when an appliance cycle is complete, or playing musical notes when a customer enters or leaves a business.

Task 3

Brainstorm three to four ideas of tunes you think would be a good fit. Think about what sounds go well together. Decide how you want the lights to be included. What colors? How long will they flash?

Answers will vary. Students should have record of a reasonable plan that would fit the criteria of the project. The plan should include both lights and music.

Differentiation

For students who need an extra challenge, have them incorporate movement into their final project. LUMA can drive around as it plays the music and flashes the lights.

For students who need a simpler task, reduce the criteria of the final Challenge to just music. Or have students complete the music first and then give them the additional layer of the lights.

Exploration: Environmentally Sound

Overview

This Exploration is an open-ended opportunity for students to explore LUMA and its programming capabilities. The questions posed to students is: What can LUMA do to help our environment? Students should be encouraged to select a specific task to complete and some sample ideas are given.

Idea Starters:

- Program LUMA to collect trash. Use sound and lights to alert people to move out of LUMA's way.
- Program LUMA to deliver safe drinking water to a location. Use sound and lights to signal a successful delivery.
- Program LUMA to simulate testing air quality in different locations. Use sound and lights to warn people if the air quality is bad.

Activity 5 – Drawing Lines

Overview

Activity 5 introduces students to Keisha Hill, an architect with Build It Up! architecture firm. In this Activity, students will combine what they've done so far and program LUMA to drive in a specific pattern. Students will learn to use LUMA to draw specific shapes such as triangles and squares. These shapes will be used as the basics to draw a picture, which will simulate creating blueprints for the architecture firm. Students will also incorporate LEDs and sound as the robot draws. Students will use functions to create different robot features and then call those functions from the main program to complete a driving course. Students will also use loops to repeat navigational algorithms in order to shorten code and to make it more efficient. In the final Challenge, students will add on to the blueprints with additional details.

Learning Objectives:

- Writing a function or procedure to simplify code
- Simplifying repeated behaviors using for loops
- Making code more efficient
- Combining multiple robot features
- Applying previously written code in new situations

Assessment

Students should be assessed on their answers within their Coding Journals as well as how closely they are able to meet the criteria of the Challenge. They should be able to successfully add at least one feature to the house that LUMA has already completed in Task 4.

Answer Key

Task 1 (answered within the learning platform)

Why is using a loop more efficient, or better, instead of repeating lines of code? List two to three reasons.

Possible answers could include: Takes less space on screen, requires less memory to run, easier to make modifications, reduces errors

Task 2

What do you think about reusing code? Is it easier than just writing the code from scratch or harder? Which do you prefer and why?

Answers will vary. Regardless of student's opinion, they should be able to justify their answer with reasonable support.

Task 3 (answered within the learning platform)

What are the steps LUMA will need to take to draw a triangular roof at the top of your house?

A list of steps to create a triangle should be listed. Sample answer: LUMA should draw a line, turn, and then draw another line.

How would you change the program if you wanted a roof that was angled on the sides and flat on the top? (answered within the learning platform)

Answers will vary, but a list of steps to create a roof with a flat top should be listed. Sample answer: Draw a short line, turn to be parallel with the top of the house (square), draw a line, turn, and draw another short line.

Differentiation

To increase the rigor of the Challenge, have students create more than one add-on to the house or provide them with specific tasks to create such as draw a tree or a doghouse.

For students who need more support, have them physically draw what add-on they would like to do before programming. This visual support creates a more concrete example for them of what they would like to code.

Robotic Artist Challenge

Overview

In this Challenge, students program LUMA to draw their first and last initials on a piece of paper. They will use LUMA's LEDs to indicate the type of movement the robot is making. When the robot completes the last letter of their initials, it should play a custom song or sound effect created by the student.

Assessment

Students should be assessed on how well they're able to program LUMA to complete the Challenge. Additionally, students can be evaluated on their planning and reflection before and after the final Task.

Differentiation

For students needing an extra challenge, refer students to the Bonus Points section of their Challenge.

For an easier task, have students complete a single initial or provide them with an outline of the letter they need to create.

Activity 6 – Toeing the Line

Overview

In Activity 6, students work with Mei Chan, an automotive designer from the Curiosity Cove Theme Park, to complete various Tasks relevant to the theme park industry. First, students will program LUMA to interact with a line on the floor using its reflected light sensor to simulate food being delivered to guests. In Task 2, LUMA will be programmed to stay within a boundary in order to enhance the environment within the park. Next, students will begin working with simulating different attractions that guests can experience. LUMA will be programmed to count the number of times it encounters a boundary to simulate a game similar to bumper cars. In Task 4, students will code LUMA to follow a line utilizing its light detection sensor. In the Challenge, students will take the code from the previous Task and improve the ride experience for guests.

Note: This Activity utilizes a physical course that needs to be built prior to jumping into this Activity. For instructions on building this course, see the Building the Testing Field video in Activity 6 resource. Build time takes approximately 30-45 minutes.

Learning Objectives:

- Use conditionals to affect robot behavior.
- Investigate sensors and how they collect data about the environment around them.
- Nest conditionals inside other conditionals or while loops.
- Use variables to track iterations through a loop and to set thresholds.
- Practice creating pseudocode to guide coding.

Assessment

Students can be assessed on their responses within the Coding Journal, especially their work within Task 5 (the Challenge). Students should be able to show how the data they collected guided their future trials in order to create a smoother path.

Answer Key

Task 1

Create pseudocode for LUMA to move forward until it detects a line.

Answers will vary. A sample response is:

```
Move forward.
Has a line been detected?
If yes, stop. If no, repeat first two steps.
```

Did the robot move forward the way you wanted it to? Why or why not? What changes do you need to make in the future?

Students should be able to explain what LUMA did, as well as reasonable explanations and changes they'd like to make.

Record what you saw happening with the red LED light on LUMA's reflected light sensor as you moved it between the light and dark surfaces.

The red LED light turns on when it is over a light colored surface. When it's over a dark colored surface, the LED turns off.

What happened when you ran the code? What changes need to be made to the code? Record your observations.

Students should record observations of LUMA as well as recommend changes that need to be made.

Task 2

Can you think of other examples of if-else conditions you encounter in your life? List two to three examples.

Examples will vary. They should all have an if-then statement followed with an else. For example: If it's raining, then take an umbrella or raincoat. Otherwise, or else, leave it at home.

Record your observations of LUMA as it runs the code. What happens?

Students should record their observations of what LUMA does. LUMA should move within the boundary established with dark tape.

Task 3

What are some ways LUMA can communicate to tell us the number of boundaries it has encountered?

This is left up to the creativity of the student. Some answers may include flash LEDs, make a noise, spin in circles, and so on.

Task 4

Record your observations as LUMA runs the code. What changes would you like to make?

Students should record their observations and recommended changes. Some possible answers could include:

LUMA followed the line, but I want it to move smoother.

LUMA went on the course, but I want it to go faster.

Task 5

Compared to the duck-themed ride, is LUMA driving smoother? Why or why not? Record your observations.

Students should compare their current observations with the results of the previous Task and provide a reasonable explanation. This code should provide a smoother drive than that in Task 4.

As you make changes to the code, record them in the following table. Use them to help you make decisions on what percentages to change and by how much.

Answers will vary based on the percent students modify. Look for rational thinking – are students randomly choosing numbers or are they using previous results to decide what numbers to select next?

Test #	Detect Dark	Detect Light	What Happened?
1	Left: % Right: %	Left: % Right: %	
2	Left: % Right: %	Left: % Right: %	
3	Left: % Right: %	Left: % Right: %	
4	Left: % Right: %	Left: % Right: %	
5	Left: % Right: %	Left: % Right: %	
6	Left: % Right: %	Left: % Right: %	
7	Left: % Right: %	Left: % Right: %	
8	Left: % Right: %	Left: % Right: %	
9	Left: % Right: %	Left: % Right: %	
10	Left: % Right: %	Left: % Right: %	

Differentiation

Students who struggle with the Challenge might need some support in understanding how the data they've already collected can guide their future trials. Pairing students who have not grasped this concept with those who have might be beneficial. Another option is doing a mini lesson in using data to make decisions within this trial.

Exploration: Sensing Fun

Overview

This Exploration is an open-ended opportunity for students to explore LUMA and its programming capabilities. Students are asked to think of ideas for how LUMA could improve Curiosity Cove Theme Park. They are encouraged to incorporate the reflective light sensor and let their creativity shine by utilizing LEGO elements or other elements to build attachments, costumes, and accessories for LUMA.

Idea Starters:

- Write a program for LUMA to become a new ride.
- Write a program for LUMA to perform a show.
- Write a program for LUMA to become a skill game for players to win prizes.
- Write a program for LUMA to serve food and drinks to guests.

Activity 7 – Pressing My Buttons (HCI)

Overview

In this Activity, students will be introduced to Isaac (Zac) Harrison from Green Tech Interfaces (GTI), who will help them integrate a user interface (UI) into LUMA's programming. In the first Task, the students will program LUMA to move in different directions based on button presses in order to feed chickens on the farm. Task 2 will have students adding on to the previous Task by adding a feature of pressing both buttons together to have LUMA go straight. This will allow LUMA to help a farmer lay a fence line. Next, students will help a farmer plant seeds by programming LUMA to follow a line and pause when buttons are pressed. Task 4 will add speed to the planting, which elevates LUMA's ability to interact with a user. Lastly, the students will be given a Challenge to integrate some feedback from the customer and modify the lights and sound feature.

Learning Objectives:

- Use while loops to repeat a series of commands while a certain condition is true.
- Use nested loops and compound conditionals (and, or, not).
- Read the value of the controller's button and program different tasks based on whether the button is pressed or not.
- Investigate HCI – Human computer interaction

Assessment

Students can be assessed on their responses in their Coding Journal. Additionally, students should be able to explain what a user interface is and how LUMA utilizes one throughout the Tasks.

Answer Key

Task 1

What would be the pseudocode for teaching LUMA to feed the chickens?

Answers will vary, but a possible answer is:
If the green button is pushed, move to the right
If the black button is pushed, move to the left
Continue the action until a new button is pushed

List two to three ideas about how this type of program could be useful on a farm.

Possible farm tasks should be listed with an explanation of LUMA moving in different directions based on button presses.

Task 2

How will we modify the code from Task 1 to meet the criteria for this new task of laying the route for the fence?

Possible answer: We'll need to add a function for when both buttons are pressed.

After running the code, draw a picture of your fence outline. What does it look like? How would the farmer use this fence on their land?

The image will differ based on the button presses the student inputted. Students should be able to explain their answer and possible uses on the farm.

Task 3

After running the program a few times, which planting pattern would be the most successful? Explain why. Are there any changes you would make to the program?

Answers will vary based on student's opinion. Student should be able to support their answer in a logical way. Suggested changes to the program will vary by student. Some examples may be: Add lights/sounds when paused to alert that a seed is being planted, change the speed of LUMA, and so on.

Differentiation

At the beginning of the Challenge, students are reminded that they did a similar code in A4T1 Backup Beeper. To add an extra challenge for students, encourage them not refer to that code until they have created their own code. Can they figure it out from what they understand about coding?

Exploration: Pressing On

This Exploration is an open-ended opportunity for students to explore LUMA and its programming capabilities. Students are asked to think of ideas for how LUMA could further support the farmer. They are encouraged to incorporate LUMA's buttons to create an interface for the farmer to tell LUMA what to do.

Idea Starters:

- Program LUMA to herd animals to the left or right using LUMA's buttons.
- Program LUMA to count chicken eggs and duck eggs using its buttons.
- Program LUMA to fertilize crops using the buttons to apply fertilizer.
- Program LUMA to harvest a crop by moving back and forth and using the buttons to turn around.

Activity 8 – Seeing Is Believing

Overview

In Activity 8, students are introduced to Jason Ingleton, the head of security at GuardianTech Industries. Because of all the new tech being created at GuardianTech, the Tasks in this Activity focus on detecting and alerting to unusual activity. Task 1 will have students program LUMA to act as a proximity alert by utilizing the ultrasonic sensor. Next, students program LUMA to alert the user to objects out of place. It will drive to an object, stop before running into it, and alert that there is an obstacle. In Tasks 3 and 4, LUMA does a little reconnaissance work. Task 3 will program LUMA to scan for objects and identify which one is closest. In Task 4, LUMA will drive to the closest object and alert the user of the object. Finally, students will wrap up with a Challenge to add on to Task 4 by programming LUMA to wait before scanning for a new object. This will incorporate an UI component and mimic the security guard investigating the object before allowing LUMA to find a new object.

Learning Objectives:

- Store sensor data in variables and compare variables.
- Use nested conditionals and loops to further define a robot's task.
- Compare nested conditionals vs compound conditions.

Assessment

Students can be assessed based on their Coding Journal responses. Additionally, the work they do with the Challenge can also be assessed. Students should be able to add in a loop to the code from Task 4 that causes LUMA to pause and wait for the user to press a button.

Answer Key

Task 1

What did you notice about the measurements comparing the gauge to the ruler? Did the measurements on the gauge match the measurements on the ruler? Was one more accurate or exact than the other? Why do you think that is?

Generally, there will be some slight differences between the two measurements. This is usually due to the object that is being measured not being held completely still. Students' responses should show evidence of logical thinking and answers should be reasonable.

Task 2

What blocks of code will we use to program LUMA to drive to an obstacle, stop before running into it, and alert us that it found something?

start moving block, **stop moving block**, **play sound effect block**

Some more advanced students may also recognize the need for a **comparison block** and/or a **forever loop block**.

List three objects that work well with LUMA's ultrasonic sensor and three that don't work well.

Answers will vary based on objects available. Generally, objects that are solid (no gaps or empty spaces) and larger will work best. For example, a book will work well. A chair will not due to the empty space between the legs.

Why do you think some objects worked better than others?

	Work Well	Don't Work Well
1		
2		
3		

The sensor works on sound waves moving toward an object. If the waves are bounced back, the sensor recognizes an object is there. If there is empty space where the waves can pass through, the sensor won't detect the obstacle as well.

Did you notice any similarities or patterns between objects that worked well and others that didn't?
Large objects that are more solid and without gaps or spaces will work better.

Task 3

What condition (or situation) would cause LUMA to continue to rotate?
As long as LUMA has not rotated a full 360 degrees (or a complete circle), it will continue to rotate.

How many turns will LUMA execute in order to complete a full circle?
36 turns (It turns 10 degrees each time and a circle has 360 degrees. 360 divided by 10 is 36.)

How will we change the code from what it is currently to get LUMA to identify an object and drive to it?
Answers will vary based on students understanding of the code but should be something along the lines of LUMA will identify which object is closest and then move toward it. Because it already spins toward the closest object after scanning the area, we'll just need to add code to have LUMA move forward toward that object.

Task 4

Why do we want LUMA to subtract five centimeters from the distance it detected the object to be at?
By subtracting five centimeters from the distance, it will take LUMA close to the object, but not right next to the object. This could be very important if the object is dangerous; we need to keep LUMA safe!

Differentiation

If the Challenge is too rigorous for some of your students, consider printing out the solution code and cutting the blocks apart. Have students put the blocks in the correct order and test their code. This will provide them with the information they need to be successful but still allow them to develop their critical-reasoning skills to analyze the correct order those blocks of code need to be placed in.

Follow the Leader Challenge

Overview

Students program LUMA to detect an object in front of it and follow the object as it moves forward or reverse at different speeds. LUMA should try to keep a distance of 40 centimeters away from the object. As LUMA follows, it must stay inside a boundary line that students create. Students will use LUMA's pixels to indicate whether LUMA needs to slow down or speed up to stay within range of the object. If LUMA loses track of the object, it should spin in a circle until it detects the object again and then follow the object in this new direction. If LUMA reaches a boundary, it should stop, play a custom sound, turn around, and wait until the object is detected in front of it again.

Assessment

Students should successfully program LUMA to complete the task of following an object without crashing into the object or leaving the boundary. Students should also document their process in their Coding Journal.

Differentiation

For students needing an extra challenge, refer students to the Bonus Points section of their Challenge.

For students needing extra support, work with a small group of students to develop pseudocode with them. Brainstorm different blocks of code they need in order to complete the final Task.

Activity 9 – Obstacle Detection

Overview

In this Activity, students work with Russell Amos with EchoReach Emergency Response. Students will program LUMA to support emergency responders. First, students will program LUMA to carry supplies between emergency response camps. LUMA will follow a line and then stop when it reaches obstacles. Next, students will add on to this programming and have LUMA give an auditory and visual alert when it stops. In Task 3, students will program LUMA to drive around an obstacle of a specific size. Then, in the next Task, students will program LUMA to avoid obstacles of any sizes. Lastly, students will be challenged to program LUMA to count the number of obstacles it encounters.

Learning Objectives:

- Gathering data using multiple sensors
- Data-based decision-making

Assessment

Students can be assessed on their responses to the Coding Journal prompts. Additionally, as the coding becomes more and more difficult, it might be beneficial to find additional ways to grade students outside of the code they create. Grading students on their pseudocode or providing partial credit to code they created may be options.

Answer Key

Task 2

List at least one reason why you would want to use a function for a new section of code.

Possible answers:

Functions make it much easier to keep code organized by tasks.

It's helpful when you need to make changes to the code. You don't have to read through all your code to find the section you need to modify.

Task 3

What pseudocode would help LUMA navigate around an obstacle?

Students need to explain their plan for LUMA moving around the obstacle. A possible solution could be:

```
Turn right
Drive forward
Turn left
Drive forward
Turn left
Drive forward
Turn right
Return to following the line
```

Task 4

How can we use the ultrasonic sensor and the ScanObstacles function to guide it around the obstacle?

Answers will vary. Basically, LUMA will move a little bit and then scan to see if it's moved far enough. If not, it will continue moving forward a little more and then check again.

Differentiation

If students are struggling to create the code for the Challenge, have them work in small groups. Or, partway through the time for working on the code, have students present their action plan for accomplishing the Challenge. Oftentimes, having group brainstorming sessions partway through a Challenge can provide the necessary inspiration needed to complete Tasks.

Exploration: Going the Distance

This Exploration is an open-ended opportunity for students to explore LUMA and its programming capabilities. Students are asked to explore other ways LUMA could help the emergency response team. Students are encouraged to use LUMA's reflected light sensor, ultrasonic sensor, buttons, pixels, and sounds to help during emergency situations.

Idea Starters:

- Program LUMA to follow a course to the hospital, flashing lights and sounding sirens to warn people to get out of the way.
- Program LUMA to be a search and rescue robot.
- Program LUMA to navigate a maze of paths to lead hikers home.

Activity 10 – Ultrasonic Theremin

Overview

In this Activity, students will work with Fatima Perez, the owner and DJ for Fusion Spin. LUMA will be programmed to be a theremin, which can be used in Fatima's gigs DJing. (Not familiar with a theremin? Check out **https://youtu.be/WYxv8r4x0JQ?si=tzGeyiFnbamHLsrV**.) In Task 1, students will use code to convert LUMA into a theremin by utilizing LUMA's ultrasonic sensor. They'll have some code already created for them that they'll insert into their programming and add to it. After students have LUMA working as a theremin, they'll add lights and the option for sound to only play with a button press. Tasks 3 and 4 will add programming for the user to change the color of the LEDs. Finally, the Challenge will have students adding additional options for the colors.

Learning Objectives:

- Use hexadecimal codes to represent colors
- Constraining data to a range of values

Assessment

Students can be assessed on their responses in their Coding Journal as well as how successful they are with the Challenge.

Answer Key

Task 1

How can you solve the problem of LUMA's slow response time to changing the pitch?

Answers that have reasonable logic would be acceptable. Something about changing the time associated with the **wait block** would be ideal.

How can you lower the range of pitches so it's easier to listen to?

Answers that have reasonable logic would be acceptable. Something about value of the **HighTone** variable would be ideal.

Task 3

How can you make the LEDs change colors based on user inputs on LUMA?

Utilizing the button presses to change the colors would be a suggested response. However, if students have an idea that has a reasonable explanation, that would be acceptable.

Task 4

You'll need to pick a color from the pixel color menu for each value of the color variable. You can pick whatever color you want for each value.

Students will complete this chart by selecting colors and recording the hexadecimal code as instructed in the video.

Color Variable Value	Pixel Color	Hexadecimal Code
1		
2		
3		
4		
5		
6		
7		
8		

Take a moment and think about what should be happening based on this new code. Record your thoughts in your Coding Journal.

Answers will vary. The audio of the instructions will give the following information:

LUMA will make different pitches based on the distance of the object in front of its ultrasonic sensor. The lights will depend on the distance – dimmer lights for short distances, brighter lights for farther distances. The pitches will only happen when the green button is pressed.

When the black button is pressed, LUMA will go into the Settings mode. As the green button is pressed, the robot will cycle through the different color options for its pixels. When the black button is pressed again, LUMA will save the color selection and exit the Settings mode. It will return to being a theremin again when the green button is held down.

Differentiation

For students struggling with the Challenge, encourage them to look at the pattern in the code when creating the options of the LEDs. Analyzing that, they can copy and paste code to add additional lights.

LUMA Is on the Job Challenge

Overview

Students choose a job or career that could be impacted by robots. Students use their creativity to figure out how LUMA could be helpful in whatever career they choose. Then, following the criteria and constraints, program LUMA to do at least one task related to this job.

Assessment

Students can be assessed by how well they met the criteria and constraints of the Challenge as well as the creativity of their solution.

Differentiation

For students needing an extra challenge, refer students to the Bonus Points section of their Challenge.

For students needing extra support, work with a small group of students to brainstorm different careers and tasks that they could select. Modify the criteria and constraints as needed to meet the skill level of the students.

Culminating Persuasive Project

After students have completed all the Activities, they're ready to decide which industry will be the best fit for LUMA. There are many ways students can present this information.

- Persuasive Essay: Have students use their persuasive writing skills to explain why LUMA should join that specific industry. Set criteria such as length, number of reasons, writing expectation, and so on. Students can submit their writing for evaluation and/or read their essays aloud to a group.
- Visual Presentation: Students use PowerPoints, Google Slides, Prezi, Pear Deck, or other digital presentation tools to share their idea as well as the rationale. Want to keep it simple? Have students create a poster explaining their decision and supporting ideas. Students can share presentations to the class or small groups.
- Film: Give students the opportunity to create a script explaining their decision and rationale to LUMA's inventors, Sharon and MJ. Students can create a video using the script and enhance it with footage of the robot, interviews with others, and music.
- Pamphlet: Have students create digital or paper pamphlets for members of the company that LUMA will be joining. Students can create visuals and write texts explaining how the robot will be a benefit for the company. Students should include tasks that the robot can complete as well as reasons why it's a good fit.
- Music and Poetry: Students use their reasons for why LUMA is a good fit for a company as the content for a poem or lyrics for a song or rap. Students can create the music themselves or use a tune already familiar to them.
- Comic Strip: Provide students with blank comic book frames and have them use characters from the Activities to explain which career LUMA will join and why.

LUMA

BY PITSCO EDUCATION