
Contents

Unit 1 Events and Sequences

- Lesson 1 The Secret of Codey Rocky
- Lesson 2 Press Buttons to Change Emotions
- Lesson 3 To Be an Animation Designer
- Lesson 4 Identify the Bug

Unit 2 Loops

- Lesson 1 The Steamed Bread Can't Jump
- Lesson 2 The Jumping Steamed Bread

Unit 3 Conditionals

- Lesson 1 The Racing Game I
- Lesson 2 The Racing Game II
- Lesson 3 Volume Bar

Unit 4 Functions

- Lesson 1 Good Morning! Functions
- Lesson 2 The Tiny Patroller I
- Lesson 3 The Tiny Patroller II

Unit 5 Variables

- Lesson 1 The Squirrel's Nuts Box
- Lesson 2 Mathematical Operations
- Lesson 3 The Bomb
- Lesson 4 Rock-Paper-Scissors

Unit 6 Inputs

- Lesson 1 Find the Blue Dot
- Lesson 2 Lucky Wheel
- Lesson 3 Bomb Disposal Expert

Lesson 4 Multifunction Switch

Unit 7 Magical Sensors

Lesson 1	Codey Rocky Can Do Addition
Lesson 2	Jumping Codey Rocky
Lesson 3	RC Car
Lesson 4	When Codey Meets Codey
Lesson 5	Volume Control
Lesson 6	Number Guessing
Lesson 7	I'm A Good Guesser
Lesson 8	Stoplight
Lesson 9	Sensing Motions
Lesson 10	Playing Sports
Lesson 11	The Greeter Codey
Lesson 12	Who Has the Fastest Hand?
Lesson 13	Speedy Colors
Lesson 14	Glowing Light

Lesson 1 The secret of Codey Rocky

Lesson Plan

Overview:

The goal of this lesson is to help students understand the concept of **Program** and what programs can do, as well as the basics of Codey Rocky and mBlock 5.

Teaching Objectives:

1. Understand the definition of Program and what Program can do.
2. Get to know Codey Rocky and its features.
3. Master the basics of mBlock 5.
4. Learn how to upload programs.

Preparation:

1. A whiteboard and a whiteboard marker (or you can use a blackboard and chalks)
2. One Codey and a Bluetooth dongle (or the USB cable) per student but it's fine if 2 or 3 students share one set.
3. A computer with installed mBlock 5 for each student but it's fine if 2 or 3 students share a computer.

Prior Knowledge

1. Computer basics;
2. Basic cognitive skills.

Teaching Procedure:

1. Warm up — The Secret of Codey Rocky

Introduce to students what Codey Rocky is: it's a tiny yet versatile robot.

The teacher can demonstrate the features of Codey Rocky through videos. Or the teacher can upload the programs to Codey Rocky in advance, making the robot perform such tasks as avoiding obstacles, following lines and more.

Ask students: **Apart from Codey Rocky, can you think of any other robots? What are those robots used for?** Pick one of the students to answer the question.

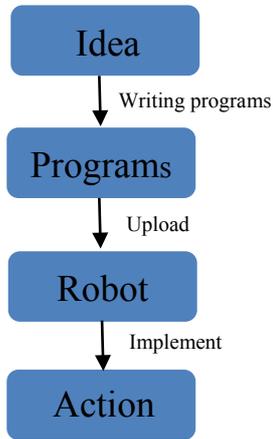
Sample answers: delivery robots, robots in logistics, security robots...

Ask again: How can those robots understand our instructions?

Give students some time for discussion.

2. New Concept — Program

The teacher says: Program is an artificial language that we use to tell robots what to do. We translate our instructions into a piece of program. Then we upload the program to the robot, making it do a variety of things as programmed.



Ask: Do you know what the answer is now? What's the secret of Codey Rocky?

Possible answer:

It's because Codey Rocky is uploaded with programs written by us.

3. Demonstration

The teacher asks students: Do you want to write code and upload the code to Codey Rocky? But before that, you need to get to know Codey Rocky. It's an adorable yet powerful robot.



Codey Rocky is an educational programmable robot. You can use software to code the robot, manipulating it to do a variety of things you can imagine. It's also a good companion that can help children learn to code. With mBlock 5, children are able to master the basics of coding and develop logical thinking as well as computational thinking. Also, Codey Rocky supports technologies like AI and IoT, which exposes children to the latest cutting-edge technologies.



Tell students: Combine Codey with Rocky and then you get a Codey Rocky. Now let's take a look at them one by one.

1) Codey: As the brain of the robot, Codey is equipped with a variety of sensors and programmable blocks. It can work individually and can also work with Rocky to perform more tasks. Now pick up your Codey. Let's take a look at what sensors it has.

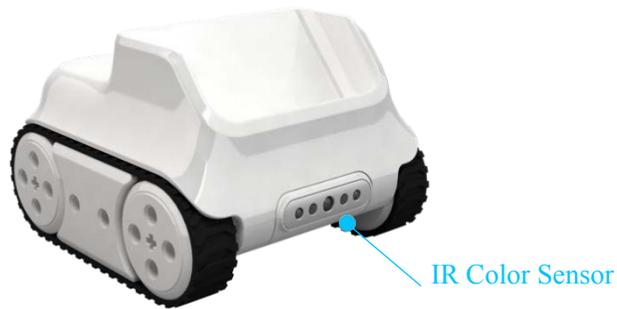


Tips:

You can use the picture above to show students what sensors Codey Rocky has and what purposes those sensors are used for.

Feature	Block Name	Applications and Core Functions
Input	IR transmitter and IR receiver	Facilitates communication between Codeys and remote controls
	Gear potentiometer	Adjusts the input signals
Input	Gyroscope	Detects how Codey moves and the angles.
	Buttons	Buttons can be programmed to control the facial expressions, motions, and sounds of Codey Rocky.
	Light Sensor	The sensor is used to measure the volume of sounds in the surrounding environments.
	Sound Sensor	The sensor is used to measure the light intensity of surrounding environments.
Output	Speaker	The sensor can be programmed to play music.
	RGB indicator	The indicator can glow in different colors.
	LED matrix screen	The images, texts and time displayed on the screen are all customizable.

2) Rocky serves as the chassis of Codey. It adds more abilities to Codey, like avoiding obstacles, identifying colors, following lines and more.



Tip:

Show the picture above to students. Let students know what sensors Codey Rocky has and what purposes those sensors are used for.

Features	Name	Applications
Input	IR Color Sensor	The IR Color Sensor integrates a color sensor, a grayscale sensor, an IR proximity sensor. By toggling the IR Color Sensor, you can make Codey Rocky perform a variety of fun tasks, like avoiding obstacles, following lines and more.
Input	Motor	The motor is used to control the motion of Rocky.

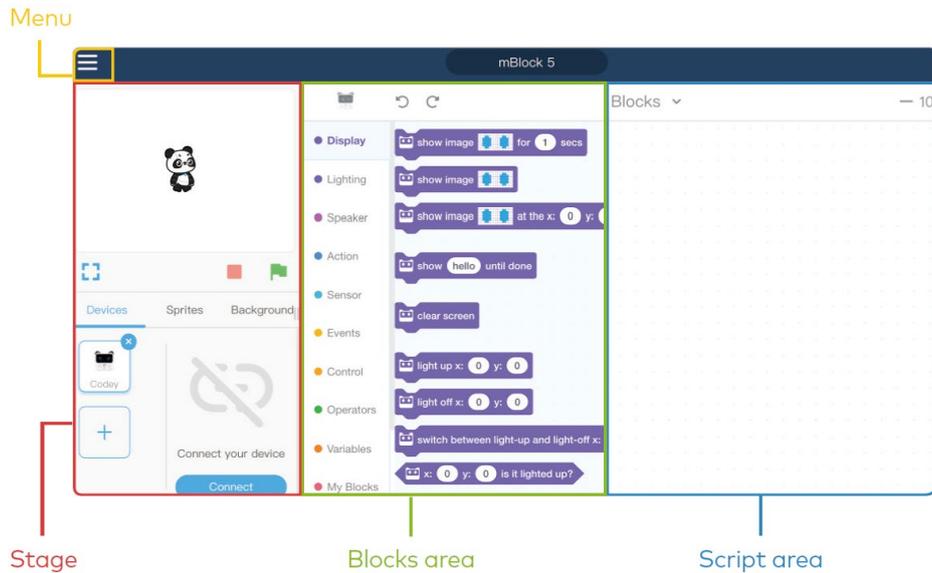
4. Imitate or Create

Have students practice how to code Codey Rocky to move!

Task 1: About mBlock 5

mBlock 5 is a programming tool which supports block-based and Python programming languages. It's developed based on the Scratch 3.0, an open-source software tool that's developed jointly by MIT and Google. Using mBlock 5, you can write programs that tell Codey Rocky or other robots to do whatever you would like. You can even take advantage of the software to create stories, games, and animations that are engaging and unique. Moreover, mBlock 5 exposes children to technologies, like AI, deep learning and model training. In a word, mBlock 5 can be one of the best options for first-time coders.

Have students open mBlock 5 PC and walk them through the interface.



Startup Interface

1. **Stage:** In this area, you can show your projects, connect devices and upload programs, add sprites and backgrounds.
2. **Blocks area:** You can find the blocks you need by color or category.
3. **Script area:** You drag blocks to this area to form programs.
4. **Device/Sprites/Backgrounds Setting area:** From here, you can find the devices, sprites, and backgrounds you need.

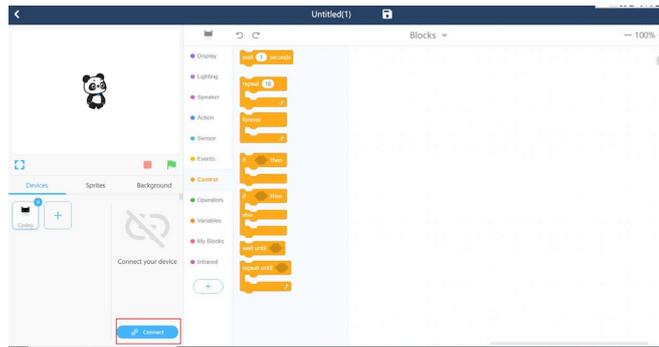
Task 2: Bring Codey Rocky to life

Have students practice how to make Codey Rocky move as programmed.

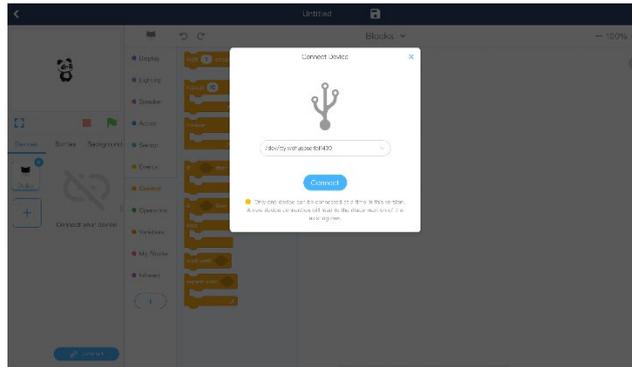
1. **Connecting to a computer:** Connect Codey to the computer via the USB cable. Then power on Codey.



2. Selecting the serial port: Open mBlock 5, click Connect, and select the correct serial port.

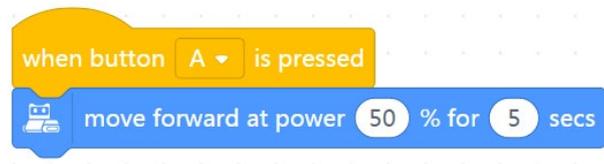


Click Connect



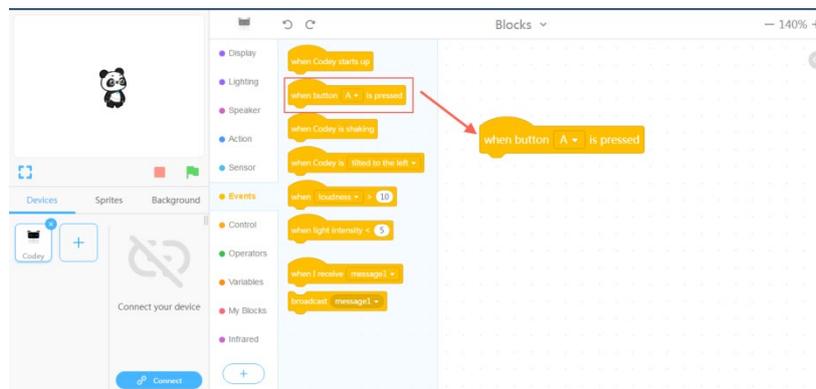
Select the serial port

3. Use mBlock 5 to create a piece of program as shown below:

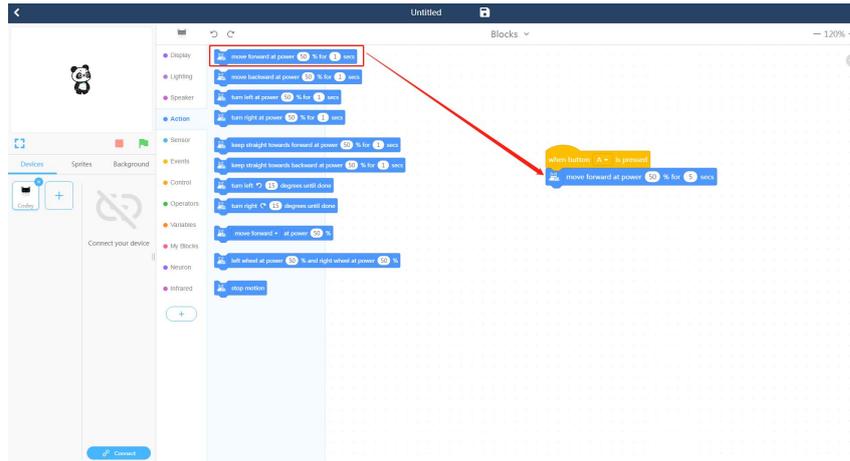


How to do:

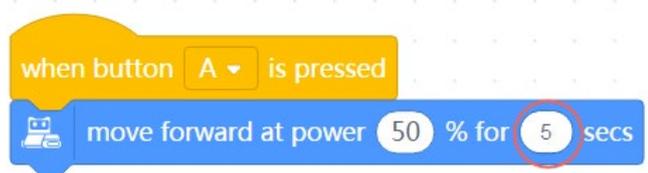
Step 1: Drag the **when button A is pressed** block out of the Event category.



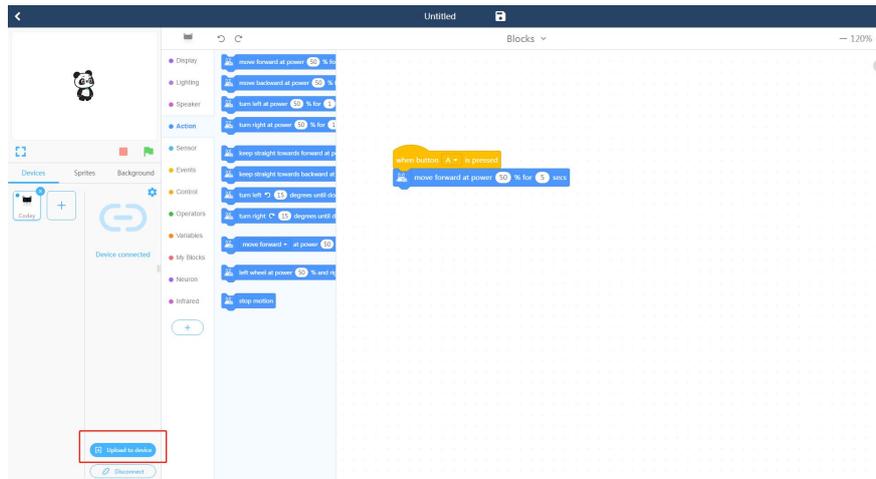
Step 2: Drag the **move forward at power 50% for (1) secs** block out of the **Action** category.



Step 3: Change the time to 5 seconds.



Step 4: Upload the program to Codey.



Click Upload to Device

Step 5. Unplug the USB cable and put Codey Rocky on the table. Press the button A and observe how Codey Rocky reacts. Have students work in pairs to complete the tasks above by writing programs.

5. Presentation

You can show some well-done projects or have volunteers present their works. When sharing the projects, students are supposed to give their answers to the questions proposed by you.

After one student presents his or her project, you can invite some students to comment on the project (what is good about the project and where to improve). Then give your own comments.

6. Wrap up

You need to give a summary of today's lesson.

Program: Program is an artificial language that we use to tell robots what to do. We translate our instructions into a piece of program. Then we upload the program to the robot, making it do a variety of things as programmed.

Ask students: Can you walk me through the interface of mBlock 5?

Sample answer: Stage area, Blocks area, Scripts area, Device/Sprites/Backgrounds Setting area.

7. Students' Self-review

Please find the attached self-review worksheets. Hand out the copies and ask students to spend a few minutes filling the worksheets.

Lesson 2 Press Buttons to Change Emotions -Events

Lesson Plan

Overview:

Learn how to write code using the Event blocks and create different facial expressions.

Teaching Objectives

1. Understand the concept of **Events**.
2. Master how to use the **Events** blocks in a program.
3. Use the **Events** blocks to create buttons that can function as you would like.

Preparation:

1. A whiteboard and a whiteboard marker(or you can use a blackboard and chalks) ;
2. One Codey and a Bluetooth dongle (or the USB cable) per student but it's fine if 2 or 3 students share one set;
3. A computer with installed mBlock 5 per student but it's fine if 2 or 3 students share a computer.

Prior Knowledge:

1. Got to know Codey Rocky;
2. Got to know the interface of mBlock 5;
3. Got to know how to write programs by dragging and dropping blocks.

Teaching procedure:

1. Review:

Ask students:

- 1) What can you learn from Codey Rocky?
- 2) By learning how to code, what technologies can you master?

Sample answers:

- 1) Able to master coding;
- 2) Able to master technologies like AI, IoT and more.

2. Explain New Concepts:

Expose students to the concept of **Event**. Tell students what an Event refers to. For example:

When it gets dark and we enter into a room, we need to turn on the light. To turn on the light, we need to press the light button. In this case, pressing the button is an **event** and that the light is turned on is the result.

Invite students to play a game, helping them have a better understanding of the concept.

3. Game—Follow the Instructions

Game rules:

You should:

- 1) Divide the students into 3 or 4 groups.
- 2) Draw some figures on the blackboard, like triangle, circle, square and star.
- 3) Define 3 or 4 events:

1. When you put your hand on the triangle;
2. When you put your hand on the circle;
3. When you put your hand on the square;
4. When you put your hand on the star.

The four events above trigger the actions below:

1. When you put your hand on the triangle — the 2nd group of students stands up;
2. When you put your hand on the circle— the 4th group of students stands up;
3. When you put your hand on the square— the 1st group of students stands up;
4. When you put your hand on the star— the 3rd group of students stands up.

Game procedure and teaching preparation:

- 1) Draw figures on the blackboard.
- 2) Divide students into groups and have them get ready for the game.
- 3) Put your hands on a shape randomly and check whether students react as required.
- 4) If students react as you expect, then you put your hand on another shape. If students fail to react as required, you need to repeat the game rules to students.
- 5) Repeat the game several times and speed up the process of switching between shapes.

You need to give a summary: In this case, the hand serves as an event. When the hand points to one shape, one specific group of students is expected to stand up as required.

Tips:

- 1) You can define the sequence number of groups;
- 2) You can customize the event. For instance, you can define the event as pointing to eyes or nose, or clapping hands 2 times or 3 times.

4. Demonstration

You need to demonstrate and give explanations. Then have students practice on their own.

Use the Event block to make Codey **【start up and smile】** .

Programs:



Explain to students:

The yellow block **【when Codey starts up】** is the Event: it means when Codey starts up;

The blue block **【show image】** is used to change the image displayed on the LED panel.

5. Practice

Task 1: Learn how to set events. Write programs to make Codey Rocky change its facial expressions based on the events. (when button A/B/ C pressed).

Tips:

- 1) Jump to the Task 2 if students are bright.

Have students work in pairs and practice themselves. Assist students when they need help.

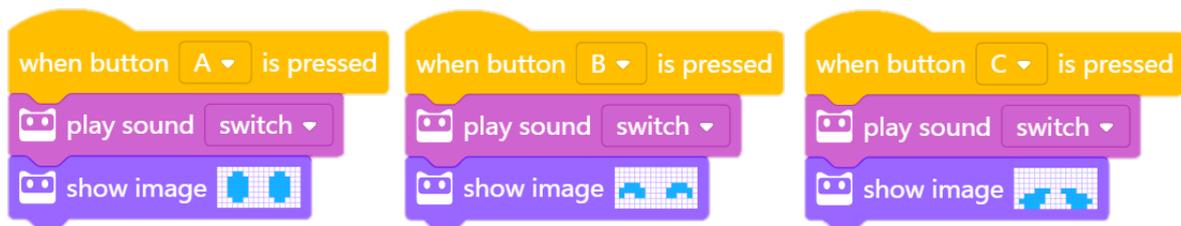
Task 2: Write programs to make Codey Rocky react in response to the events (when button A/B/C pressed), like changing its facial expression or making different sounds. Have students share their projects.

Tip :

- 1) If time allows, you can have students edit the images as they would like.

You can tailor the time limit based on your teaching purposes and the characteristics of students. Recommended time: 10 min

Here are the sample projects:



Program Story:

【When button A is pressed】 , then **【play sound switch】** and Codey open its eyes;

【When button B is pressed】 , then **【play sound switch】** and Codey smiles ;

【When button C is pressed】 , then **【play sound switch】** and Codey becomes sad.

Have students complete the tasks above in the form of pair programming.

Tell students that there will be a presentation session. Students are expected to share their projects with the whole class and give their answers to the following questions:

- 1) What is your project about?
- 2) Did you come across any problems? How did you solve them?

6. Presentation

You can show some well-done projects or have volunteers present their works. When sharing the projects, students are supposed to give their answers to the questions proposed by you.

After one student presents his or her project, you can invite some students to comment on the project (what is good about the project and where to improve). Then give your own comments.

7. Wrap up

You need to give a summary of today's lesson:

Remind students that an event is the beginning of a piece of program. When you write programs, the first thing you need to do is to select an event.

8. Students' Self-review

Please find the attached self-review worksheets. Hand out the copies and ask students to spend a few minutes filling the worksheets.

Lesson 3 To be an Animation Designer

Lesson Plan

Overview:

Understand the concept of Sequence and write programs to create animations.

Objectives:

1. Understand the concept of **Sequence**.
2. Master the basics of Sequence and learn how to create animations using **Sequence**.

Preparation:

1. A whiteboard and a whiteboard marker(or you can use a blackboard and chalks);
2. One Codey and a Bluetooth dongle (or the USB cable) per student but it's fine if 2 or 3 students share one set;
3. A computer with installed mBlock 5 per student but it's fine if 2 or 3 students share a computer;
4. Copies of A4 paper and pens (the amount depends on the number of students).

Prior Knowledge:

1. Mastered the basics of coding;
2. Knew how to use mBlock 5.

Teaching Procedure:

1. Review last lesson.

Ask students:

- 1) What is an Event?
- 2) Can you think of any events in daily life?
- 3) What events were used in the last

lesson?

Sample answers:

- 1) Event is an action that can cause things to happen;
- 2) Pressing the button leads to the light bulb lighting up.

In this case, pressing the button is an event and that the light bulb is turned on is the result.

3) Events that are used in the last lesson include: when program starts up, when button A/B/C pressed.

2. Explain New Concept—— Sequence

- 1) Ask students: What are the steps to put a watermelon into a refrigerator?
- 2) Sample answer: Open the refrigerator door and put the watermelon into the refrigerator.

Students might give you other answers.

3) Explain to students: To put the watermelon into the refrigerator, you need to take these steps: open the refrigerator, put the watermelon into the fridge, close the fridge door. If you are not taking the steps, you will be not able to put the water into the fridge.

- 4) Ask students: Can you think of any cases in which you need to follow a set of steps to

achieve something?

Tip:

Each example should only offer one specific order. It means that only when you follow one.

Leave some time for students to brainstorm. You can give an example: To drink water, you need to uncap the bottle, pour the water into your mouth and screw the cap of the bottle. If you don't take the steps, you won't be able to drink the water.

Have students play a game, helping them have a better understanding of the concept of Sequence.

3. Game ——I'm a Robot

Game rules:

You should act as a robot, walk from somewhere in the classroom to the blackboard and draw a smiley face on it. Invite students to give instructions (move forward, turn left, turn right and more) to the robot and write instructions on paper. You need to follow the instructions.

Game procedure and teaching preparation:

1. Introduce game rules to students and get students ready for the game.
2. Invite students to give instructions and write down the instructions on the A4 paper. Ask students to ensure instructions are arranged in the correct order (from top to bottom).
3. You should read the students' instructions from top to bottom and perform tasks as instructed.

Tips:

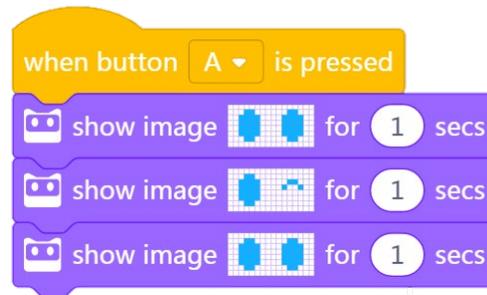
1. If students write instructions from left to right, teachers should still read instructions from top to bottom. In this case, there is a possibility that instructions can only be read from left;
2. When students' instructions are unclear, you still need to follow the instructions to make actions. For example, if the instruction is: turn left, move ahead by 4 meters, then the robot should execute the instruction like: turn left and move ahead. This is exactly how the software instructions are performed. When there is no specific setting for the time and the angle, the computer will read the simple instruction for turning left promptly and then read the instruction for moving ahead;
3. If it's necessary to make the instructions more specific, you can remind students about the fact that the robot lays down its two hands vertically. Therefore, when students are giving certain instructions, they need to make sure that the instructions are detailed enough. For instance, if the instruction for the robot is to pick up a pen, the instruction must include details: by which hand, the hand gesture, where to draw the smiley face exactly on the blackboard, etc.;
4. In consideration of the time limit and the ages of students, you can simplify the instructions. Anyway, the key point is clear: you need to make instructions specific and arrange them in the correct order if you want the robot to do things as you program.

Summary: When we are programming, we arrange the blocks in the order from top to bottom to form a set of steps. In this way, robots can follow the steps to perform a task. We refer to the set of steps as **Sequence**.

4. Demonstration

Create animations using the block **【show image () for () secs】**. The method is simple: Use the image of the block as the base, duplicate the block, and change the image slightly. Repeat the steps and then arrange those blocks in sequence.

Sample project: Winking Eyes



Program story:

Drag out the block **【show image () for () secs】** and edit the image to be a pair of open eyes;

Duplicate the block **【show image () for () secs】** but make the eyes wink;

Duplicate the block **【show image () for () secs】** once again, and this time make the winking eye open.

Add the Event block ——"When button A pressed"

Upload the programs to the device. When the button A is pressed, Codey will wink at you.

Explain to students:

Traditionally, animation designers would make animations by following these steps: Put a sheet of static drawing on the table first and unfold a new drawing paper on top of the first paper. Designers would outline the frame and then change the drawing bit by bit at a time. Then another piece of paper, outline the frame and change the drawing slightly again. Designers repeat the steps over and over again until they complete a series of pictures that are slightly different from each other. Then, they flip the drawings quickly to animate the pictures. Based on the same principle, we use the block "show image () for () secs" in our program to create animations.

5. Practice

Have students complete the following tasks.

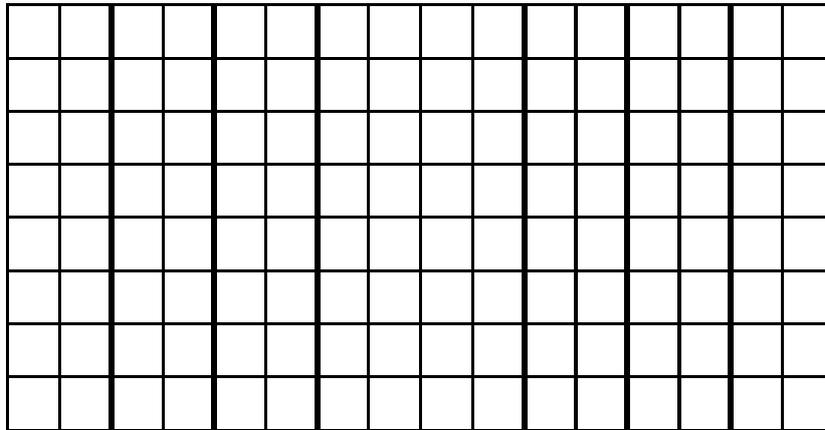
Task 1: Students are supposed to complete the project 【Winking Eyes】 as the teacher did.

Have students work in pairs and complete the task as you just did.

Task 2: Create Animations

Students are supposed to share their projects after they complete this task.

Ask students to work in pairs to complete Task 2. Students need to create an image by filling the following grid. Use the block 【show image () for () secs】 to create the same image; duplicate the block but change the image slightly; repeat the steps until a series of coding blocks are created. These coding blocks show different images and should be arranged in sequence.



Upload the program to Codey. Then play the animation and have students share their projects with classmates.

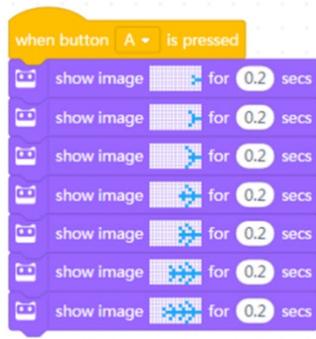
Tips:

1. Some students might accomplish the task ahead of time. Tell them to give their animations different events;
2. Or have them change how long the animation lasts;
3. If some students fail to accomplish the task on time, invite them to share one thing that happened when they were writing programs. It could be something funny, a challenge they confronted or a problem they have;
4. Students can share their works with the whole class, or they can take turns presenting the works to other groups ;
5. Teachers can tailor the time limit according to teaching purposes and the personality of the class.

Recommended time: 10 mins

Before students practice on their own, you can show them two sample projects: *A Growing Tree* and *Taking a Walk*.

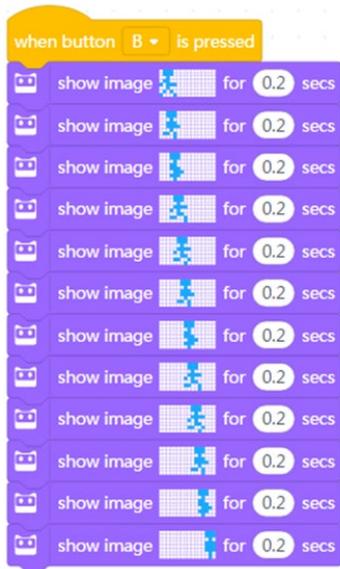
Sample project: A Growing Tree



Program Story:

Stand up the LED matrix display on the table. When the button B is pressed, the sapling will be growing.

Sample Project: Taking a Walk



Program Story :

When the button B is pressed, a kid will keep running from one side of the LED matrix display all the way to the other side.

Have students complete the tasks above in the form of pair programming.

Tell students that you will invite groups to present their works. And they need to answer these questions:

- 1) What is your project about?
- 2) Did you come across any problems? How did you solve them?

6. Presentation

You can show some well-done projects or have volunteers present their works. When sharing the projects, students are supposed to give their answers to the questions proposed by you.

After one student presents his or her project, you can invite some students to comment on the project (what is good about the project and where to improve). Then give your own comments.

7. Wrap up

You need to give a summary of today's lesson:

Remind students that Sequence refers to a series of steps which are carried out in order to complete a task. Sequence.

8. Students' Self-review

Please find the attached self-review worksheets. Hand out the copies and ask students to spend a few minutes filling the worksheets.

Lesson 4 Identify the Bug

Lesson Plan

Overview:

Understand the concept of **Bug** and learn how to identify bugs and fix the bugs.

Teaching objectives:

1. Understand the concept of Bug;
2. Know how to identify bugs and fix the bugs.

Preparation:

1. A whiteboard and a whiteboard marker(or you can use a blackboard and chalks) ;
2. One Codey and a Bluetooth dongle (or the USB cable) per student but it's fine if 2 or 3 students share one set;
3. A computer with installed mBlock 5 per student but it's fine if 2 or 3 students share a computer;

Prior Knowledge:

1. Mastered the basics of coding;
2. Mastered the basics of mBlock 5;
3. Knew how to write programs.

Teaching procedure:

1. Review

Ask students:

- 1) What is Sequence?
- 2) Can you think of sequences in daily life?

Sample answers:

- 1) Sequence refers to a series of steps which are carried out in order to complete a task.

For example: putting the watermelon into the refrigerator, washing hair and more.

2. Explain the new concept

Tell students what bug refers to.

Bugs are inevitable when we write programs. A "bug" means an insect in its literal sense. But in programming, a bug refers to a mistake that will lead to the failure of a program. It's like a typo in an article. You need to correct the errors because they might make your article difficult to understand.

3. The Behind Story

Tell a story to help students have a clearer understanding of the concept of **Bug**.

At the initial stage where computers were invented, unlike today's portable computers, they were extremely large in size back then. (Picture from Wiki).

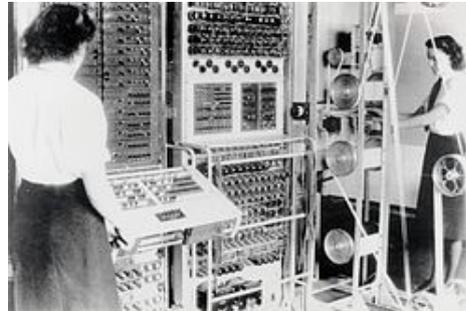


Figure 1 Colossus computers were used to decipher German codes during WW II

Once upon a time, a colossus computer failed to work. The whole team of programmers tried to identify the problem but to no avail. In the end, Grace Murray Hopper, a female programmer, identified what the problem was: a moth flew to the inside of the computer and caused the glitch. When they removed the moth, everything was back on the right track. It was the first bug in a computer program that was found and programmers affixed it to the logbook (see the picture above). Since then, the term bug becomes common in use when people refer to mistakes in a computer program. Naturally, Grace Murray Hopper was since then considered as the Mother of Debug.

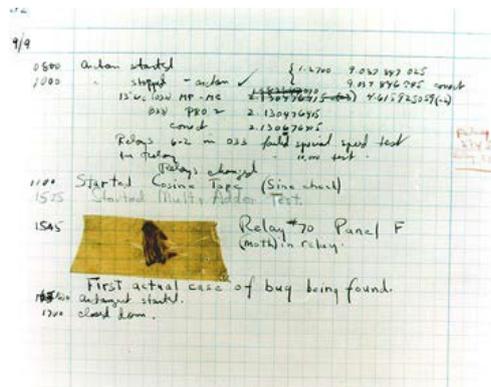


Figure 2 The moth caused the computer mistake and this is the first mistake in a computer program.

Move on to the next session. Have students learn how to identify bugs in programs.

4. Demonstration

Show students 3 sample projects:

Sample project 1 - The Car Key was Stolen!

Sample project 2 - The Bomb Can't Count Down!!

Sample project 3 - The Earthworm and the Bug

Remind students of one thing: In order to identify bugs in the programs, we need go over each line of code from top to bottom.

Task 1: Find bugs in the following programs and try to fix them:

1. The Car Key was Stolen!



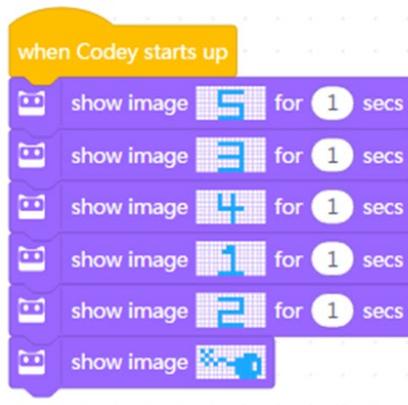
The teacher: tell students that the car key was stolen so they can't start the car. (lack of an event)

Students: Examine the program and try to find the bug.

The teacher: Ask students whether they identify the bug or not.

Students: The bug is found. Students need to add an Event block to the code.

2. The Bomb Can't Count Down!!



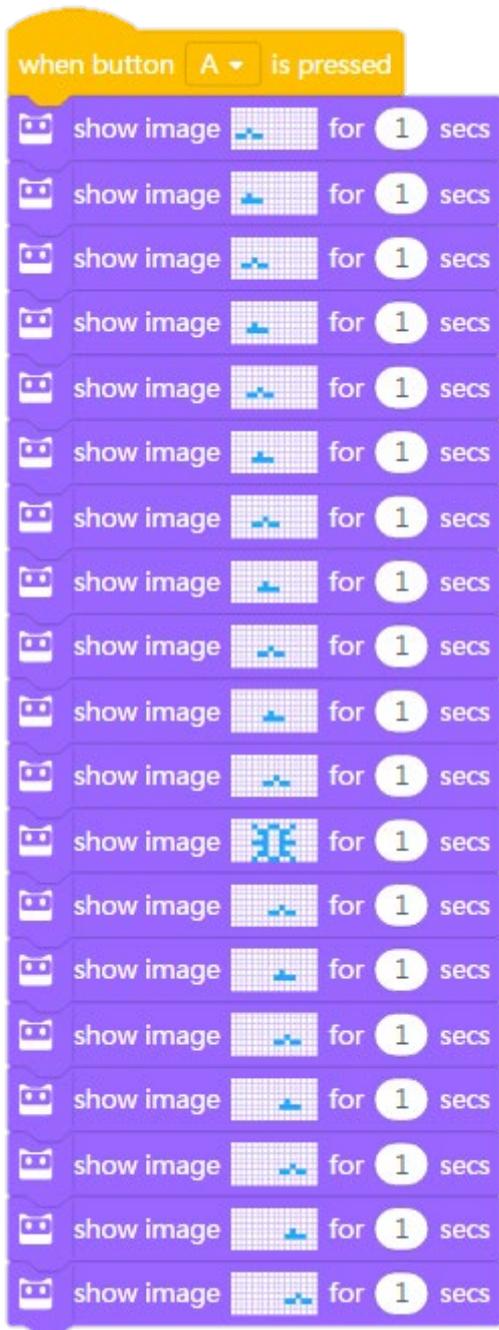
The teacher: Tell students the bomb can't count down and ask them to find the bug in the program.

Students: Try finding the bug in the program.

The teacher: Ask students whether they find the bug or not.

Students: The bug is found. Students need to rearrange the blocks and make sure the blocks are arranged in the correct order.

3. The Earthworm and the Bug



The teacher: Tell students a story: A little earthworm meets a big insect when crawling on the ground. Students need to remove the insect to let the earthworm keep moving forward (replace the insect image with an image of crawling earthworm). Ask students to find out the bug in the program.

Students: Examine the program and try to find the bug.

The teacher: Ask students whether they find the bug or not.

Students: The bug is found. Students need to replace the insect with an image of crawling earthworm.

Have students complete the tasks above in the form of pair programming.

Tell students that there will be a presentation session. Students are expected to share their projects with the whole class and give their answers to the following questions:

- 1) What is your project about?
- 2) Did you come across any problems? How did you solve them?

5. Presentation

You can show some well-done projects or have volunteers present their works. When sharing the projects, students are supposed to give their answers to the questions proposed by you.

After one student presents his or her project, you can invite some students to comment on the project (what is good about the project and where to improve). Then give your own comments.

6. Wrap up

You need to give a summary of today's lesson:

Remind students that bugs are inevitable when we write programs. So it's important to know how to identify bugs in programs and how to fix them. In mBlock 5, coding blocks are arranged in the order from top to bottom so we should go through each line of the code in the same order to find out bugs if programs are not running as expected.

7. Students' Self-review

Please find the attached self-review worksheets. Hand out the copies and ask students to spend a few minutes filling the worksheets.

Lesson 5 The Steamed Bread Can't Jump

Lesson Plan

Overview:

Use the Counting Loop block in programs to create engaging animations.

Teaching Objectives:

1. Understand the concept of Loops.
2. Know how to use the Counting Loop blocks.
3. Learn how to create animations using the Counting Loop block in your program.

Preparation:

1. A whiteboard and a whiteboard marker (or you can use a blackboard and chalks) ;
2. One Codey and a Bluetooth dongle (or the USB cable) per student but it's fine if 2 or 3 students share one set;
3. A computer with installed mBlock 5 per student but it's fine if 2 or 3 students share a computer.

Prior Knowledge:

1. Mastered the concept of Sequence and how to use the Event blocks in programs;
2. Knew how to find bugs and fix them.

Teaching Procedure:

1. Review:

Ask students:

- 1) What is Sequence?
- 2) Can you think of any situations where you need to stick to sequences in daily life?

Sample answers:

- 1) **Sequence** refers to a series of steps which are carried out in order to complete a task;
- 2) In daily life, we often need to follow sequences, like putting a watermelon into the refrigerator, washing hair and more.

2. Explain the New Concept

Take advantage of the PPT and explain the concept of counting loops and how to use counting loops in programs to students.

- 1) Tell students today they are going to learn about the concept of **Loop**.
- 2) Use a story to introduce the concept: A panda is going to plant trees and it needs to follow a set of sequences.
- 3) To plant a tree, the panda has to take these steps: dig a hole; drop down the sapling; cover it with soil; move forward 5 meters and plant another tree.
- 4) If the street is 20 meters long, the panda needs to repeat the steps 4 times. What steps will be repeated and how many times?
- 5) By repeat or loop, we mean the panda will carry out the actions over and over again. In

programming, a loop refers to a piece of code that is executed repeatedly by the computer.

3. Game - Tap to the Beat

Have students play a game, helping them have a better understanding of the concept.

Game procedure and teaching preparation:

1) A represents tapping your left leg with the left hand; B represents tapping your right leg with the right hand; C represents tapping your legs with both hands. Follow the beats:

ABABABC

Chaos might ensue when students are required to tap the table at a rapid speed. In this case, you can ask students whether it's necessary to simplify the beats. You can simplify the beats to be:

ABABABC Repeat two times

2) What if we add another beat? Add a beat D representing clapping the hands:

ABABABC ABABABC ABCABC ABABABC

ABABABD ABABABD ABDABD ABABABD

ABABABC ABABABC ABCABC ABABABC

ABABABD ABABABD ABDABD ABABABD

Ask students to tap their legs to the beats above. Or you can simplify the beats:

ABABABC **Repeat two times**

ABC **Repeat two times**

ABABABC

ABABABD **Repeat two times**

ABD **Repeat two times**

ABABABD

Repeat two times

Tips:

1) You can write down beats on the blackboard and mark the beats while working with students to figure out the rules;

2) Make sure the beats are simple if you are to design the rhythm yourself. Besides, don't make the game too long, otherwise students might lose patience soon.

You should give a summary:

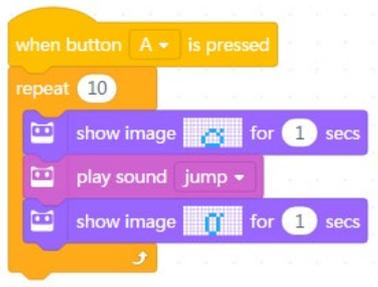
We use a loop to make the beats look neat and the game easier to play. In programming, the Repeat block is frequently used to make the code neat. With the Repeat block, Codey Rocky can carry out the same sequence of actions repeatedly. When the Repeat block is executed once, we count it as one cycle or one iteration.

4. Demonstration

Show students how to use the **repeat ()** block to create a fun project - The Steamed Bread Can't Jump

Story Line: The steamed bread is persisting in learning how to jump. It tried many times but failed to make it.

Program:



Ask students: What event is used in the program above? Can you use another event? How many times has the steamed bread tried?

5. Practice

Ask students to use the **repeat** block to create animations. Students can choose to complete one of the following tasks:

- 1) Improve on your sample program. Students can change the event, the sound or the number of times;
- 2) Students can change the image. The software has many built-in images. They can pick one image they like;
- 3) Give the animation a storyline.

Have students complete the tasks above in the form of pair programming.

Tell students that you will invite groups to present their works. And they need to answer these questions:

- 1) What is your project about?
- 2) Did you come across any problems? How did you solve them?

6. Presentation

You can show some well-done projects or have volunteers present their works. When sharing the projects, students are supposed to give their answers to the questions proposed by you.

After one student presents his or her project, you can invite some students to comment on the project (what is good about the project and where to improve). Then give your own comments.

7. Wrap up

You need to give a summary of today's lesson:

In programming languages, a Loop refers to a set of instructions that need to be executed repeatedly. When writing programs, we can use the Repeat block to replace those instructions that are executed repeatedly, making the code clean and concise. With the Repeat block, we can also create fun projects.

8. Students' Self-review

Please find the attached self-review worksheets. Hand out the copies and ask students to spend a few minutes filling the worksheets.

Lesson 6 The Jumping Steamed Bread

Lesson Plan

Overview:

Use the forever block to create fun animations.

Teaching Objectives:

1. Understand the concept of Infinite Loop.
2. Learn how to use the Forever block.
3. Use the Forever block to create your own animation.

Preparation:

1. One Codey and a Bluetooth dongle (or the USB cable) per student but it's fine if 2 or 3 students share one set;

2. A computer with installed mBlock 5 per student but it's fine if 2 or 3 students share a computer;

Prior Knowledge:

1. Mastered the concept of Loop;
2. Knew how to use the repeat block in programs;
3. Knew how to find bugs and fix bugs.

Teaching procedure:

1. Review

Ask students questions:

- 1) What is a loop?
- 2) What does a counting loop refer to?
- 3) Can you think of any counting loops in daily life?

Sample answers:

- 1) In programming, a loop refers to a piece of code that is repeated by the computer;
- 2) Counting loops refer to the code that will be cycled specific times;
- 3) Counting loops are everywhere in daily life, for instance, planting trees, brushing teeth and going to school.

2. Explain the New Concept

Take advantage of the PPT and explain the concept of counting loops and how to use counting loops in programs to students.

1) Tell students loops are divided into two types. One type of loops only repeats specific times while another type of loops cycles endlessly, like sunrise and sunset;

2) In programming, we use the **forever** block to repeat a piece of code endlessly.

You need to open mBlock 5 and drag out the **repeat** block and the **forever** block. Ask students what the difference is between the **forever** block and the **repeat () times** block. Remind student that the **forever** block has no bump at the bottom, which means another block can't be added to the end. This is because the

code inside the **forever** block will be executed endlessly.

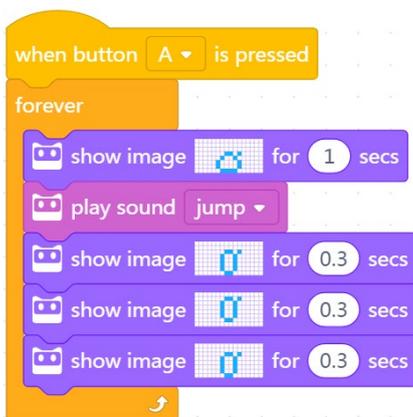


3. Imitate and Create

You should first remind students of the animation that was created in the last lesson and introduces a new story: One day, the steamed bread finally knows how to jump. It keeps hopping and can go anywhere.

Then have students use the forever block in the program to make the steamed bread jump up.

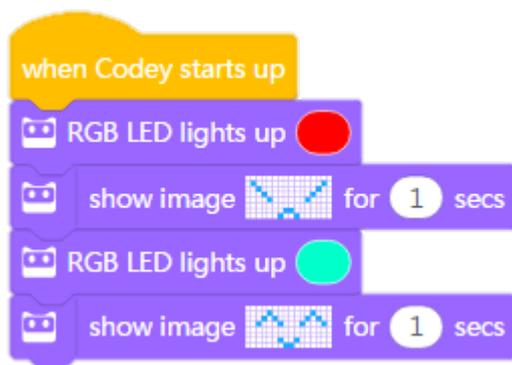
Sample program:



4. Imitate and Create

Have students complete the tasks above in the form of pair programming. Ask them to complete

- 1) Design two animations. Give each animation a specific event (button pressed). But use the **repeat** block to create one animation and use the **forever** block to program the other animation.
- 2) You can give the animations storylines, like one being *The Steamed Bread Can't Jump* and the other being *The Jumping Steamed Bread*; or the two animations can be mutually independent.
- 3) You can also spice up your animations by adding some sounds or lights effects.



Tell students that you will invite groups to present their works. And they need to answer these questions:

- 1) What is your project about?
- 2) Did you come across any problems? How did you solve them?

5. Presentation

You can show some well-done projects or have volunteers present their works. When sharing the projects, students are supposed to give their answers to the questions proposed by you.

After one student presents his or her project, you can invite some students to comment on the project (what is good about the project and where to improve). Then give your own comments.

6. Wrap up

You need to give a summary of today's lesson:

Remind students that a loop refers to a piece of code that is repeated by the computer. Ask students if they can think of any loops in daily life. You might get answers like the ebb and flow and the sunrise and sunset. Finally, conclude the lesson by calling students to create animations using the **forever** block in programs.

7. Students' Self-review

Please find the attached self-review worksheets. Hand out the copies and ask students to spend a few minutes filling the worksheets.

Lesson 7 The Racing Game I

Lesson Plan

Overview:

Have students master the concept of Conditional through playing games. Learn how to use the conditional blocks to enable Codey Rocky to identify colors and avoid obstacles. And have students complete the challenges.

Teaching Objectives:

1. Understand the concept of Conditional;
2. Know how to use the **if statement** block to create simple projects.

Preparation:

1. A Conditional Box. The box will be used in the game;
2. A whiteboard and a whiteboard marker (or you can use a blackboard and chalks);
3. Digital challenge cards (or copies);
4. One Codey and a Bluetooth dongle (or the USB cable) per student but it's fine if 2 or 3 students share one set;
5. A computer with installed mBlock 5 per student but it's fine if 2 or 3 students share a computer.

Prior Knowledge:

1. Knew how to find bugs and fix the bugs.

Teaching Procedure:

1. Review

Ask students:

- 1) What is a loop?
- 2) What is a counting loop?
- 3) What is an infinite loop?
- 4) When will the forever block be used?

Sample answers:

- 1) In programming, a loop refers to a piece of code that is executed repeatedly by the computer;
- 2) Counting loops refer to the code that will be cycled specific times;
- 3) Infinite loops refer to the code that will be cycled endlessly;
- 4) We use the **repeat** block or the **forever** block in programs if a piece of code needs to be repeated specific times or endlessly. Using the two blocks will make the code neat.

2. Explain the new concept

Take advantage of the PPT and explain the concept of Conditional and how to use conditional blocks in programs to students.

- 1) Tell students that today they are going to learn how to use the **if statement** blocks in programs. Tell a story: A panda is going to buy a cup of tea. It opens the door to see whether it's

raining.

2) If it's raining, it will put on its raincoat and then leave home. If it's not raining, the panda will go out straight away.

3) Conditional refers to the instructions that depend on whether something is true or false. The instruction will be executed only when the conditional statement is true; otherwise, the program will skip it or ignore it.

4) When programming, students can use the **if statement** block to make decisions.



5) First, add the **if statement** block. Then, add the instruction that will be executed when the condition is true. Outside the **if statement** block is the instruction that will be executed when the condition is false



3. Game—A Conditional Box

Have students play a game **Conditional Box**. Students will be able to get a better understanding of the concept "Conditional".

Game rules:

1) The box is full of paper strips. On those paper strips there write all kinds of conditional statements and specific instructions. Ask students to pick a paper strip randomly out of the box and read out what it says, make a decision and carry out actions based on the decision.

2) Ask students to write down the conditions on paper strips in advance and leave the actions part to you. In this way, we can avoid occasions where students might write down some impossible missions, like walking out of the classroom to leave school or jumping down from the 2nd floor. The chart attached is for reference. You can change the actions or add new actions. Print the chart and cut the paper into strips along the dashed lines. Fold the strips in half and put them all into the conditional box.

3) Remind students one thing: write down conditions that are easy to identify. This is to ensure that the game can go on smoothly. For instance, write conditions like these: if you have long hair; if you wear glasses; if you are in a black shirt; if your name has a letter A in it; if you were born in June; if someone puts up his or her hand; if someone claps his or her hands. If the conditional statement on a paper strip is hard to identify (for instance, "if it rains tomorrow" or "if the amount of your hair is an odd number"), ignore that paper strip.

4) Have students come to the podium to pick paper strips out of the box, or they can pass

around the box sitting on their seats.

Game procedure and teaching preparation:

1. Pick a paper strip out of the box and open it;
2. Read aloud what the paper strip says and make decisions. If the conditional statement is true, do as told correspondingly; if the conditional statement is false, then ignore the paper strip ;
3. Fold the paper in half and put it back into the box. Go back to the seat or hand the box to the next student.

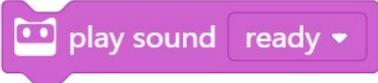
4. Demonstration and guided practice

Ask students to complete the challenges using the **if statement** block.

The storyline: Codey Rocky is taking part in a racing game. The game is starting soon. If Codey Rocky sees the green flag waving, it will start off at its top speed. Show students how to complete the Challenge 1.

Challenge 1—Start off when
the flag is waving

The game will begin soon. If Codey Rocky sees the green flag waving, it will start off at its top speed.

- When button A is pressed, Codey Rocky is getting ready at the starting line (play the sound ready).

- If the color detected is green, Codey Rocky will move forward at its top speed.



- The RGB LED will turn red once the program makes a decision based on the situation.

- Challenge:** Change the code to add facial expressions and sounds to Codey Rocky.

Follow the steps the Challenge Card above to show students how to create programs as shown below:



Remind students of the following things:

- 1) The **if statement** block includes a hexagonal dent, so we need to add a condition block (for instance, "if it is green?") inside the dent;
- 2) Remind students that the red block "RGB LED lights on with color red" should be put outside the **if statement** block. This is to help us assess whether the execution of the code comes to an end or not.

Have students work on the following extension activities:

- 1) Give Codey Rocky facial expressions and sounds when it is running;
- 2) If Codey Rocky identifies a red item, it will move backward.

Invite students to share their projects with the whole class.

5. Independent Practice

Give students a brief introduction to Task 2.

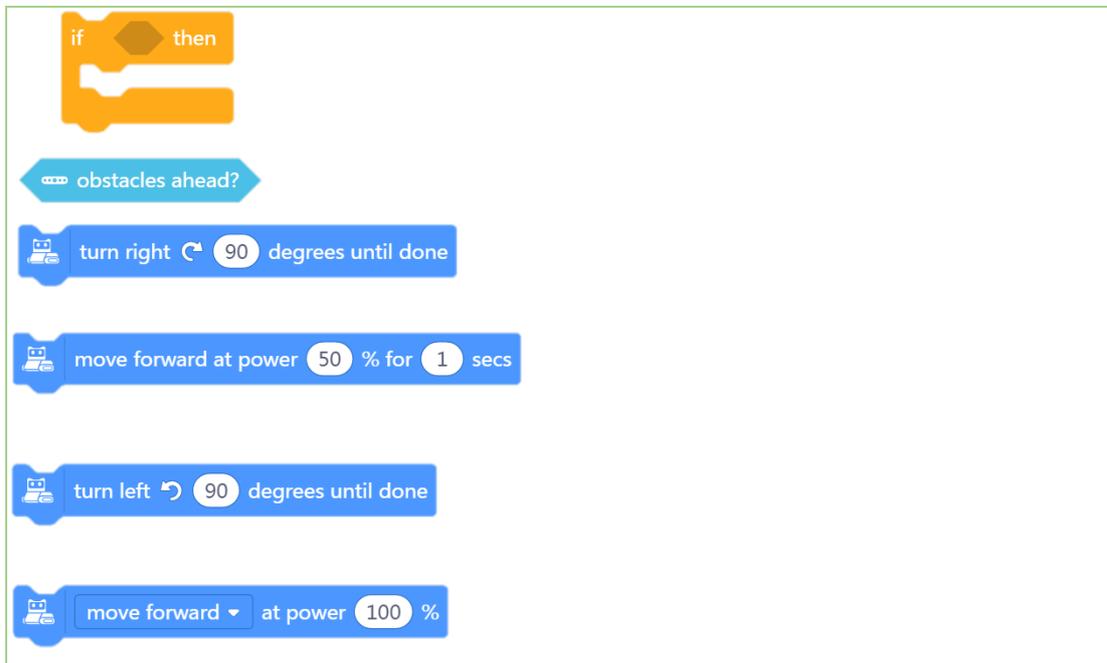
Have students complete Task 2 independently or in pairs.

Challenge 2 — Avoid the obstacles

When meeting an obstacle, Codey Rocky will avoid it and keep moving forward.

- Place an item in front of Codey Rocky


- When the button A is pressed, if Codey Rocky detects an obstacle, it will turn right by 90 degrees, move forward, turn left by 90 degrees and keep moving forward at a rapid speed.

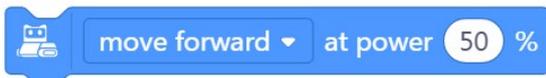


When students are working on the task, you need to

1) Make sure students put the "move forward" block and the "turn left/right" block with time limits inside the **if statement** block.



Not

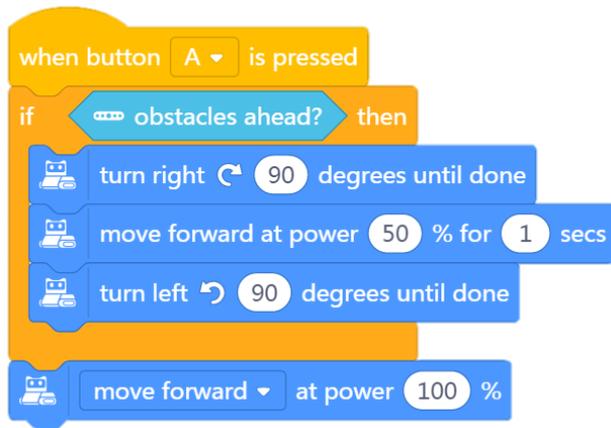


The first two blocks will stop when the time is up and then move on to run the next block. However, the third block will keep running with other blocks unless you use another action block in the programs. In this case, Codey Rocky will keep moving forward at power 50% until someone powers it off.

2) Remind students that the block "move forward at power 100%" should be put outside the **if statement** block. After Codey Rocky turns left, the piece of programs surrounded by the **if statement** block will come to an end, and the outer block starts to run. The outer block has no time limits.

3) Have students think about one question: What will happen if they encircle the whole piece of code with the **forever** block?

Sample program:



6. Wrap up

You need to give a summary of today's lesson:

Remind students that conditional refers to the instructions that depend on whether something is true or false. The instruction will be executed only when the conditional statement is true; otherwise, the program will skip it or ignore it. In the **if statement** block, there is a hexagonal hole. We need to put the condition code (like "color is green?" or "obstacles ahead?") inside the hole.

7. Students' Self-review

Please find the attached self-review worksheets. Hand out the copies and ask students to spend a few minutes filling the worksheets.

Lesson 8 The Racing Game II

Lesson Plan

Overview:

Use more than one **if statement** blocks or nest the **if statement** block inside the **repeat** block to create projects. And complete two challenges using the comparison operators.

Teaching Objectives:

1. Learn how to use more one **if statement** block in programs;
2. Learn how to use the **if statement** block and the **repeat** block to make your code concise.
2. Learn how to use the comparison operators inside the **if statement** block.

Preparation:

1. Digital challenge cards (or copies);
2. One Codey and a Bluetooth dongle (or the USB cable) per student but it's fine if 2 or 3 students share one set;
3. A computer with installed mBlock 5 per student but it's fine if 2 or 3 students share a computer.

Prior Knowledge:

1. Knew how to use the **if statement** block;
2. Knew how to use the **repeat** block;
3. Knew how to find bugs and fix them.

Teaching Procedure:

1. Review:

Ask students:

- 1) What does conditional refer to?
- 2) Can you think of any examples in daily life? You have to decide whether to carry out certain actions depending on the situation?
- 3) What features does the **if statement** block have?
- 4) In the last lesson, what kind of **if statement** blocks did you use?

Sample answers:

- 1) The instruction will be executed only when the **if statement** is true; otherwise, the program will skip it or ignore it;
- 2) Whether I take the umbrella or not depends on whether it's raining;
- 3) In the **if statement** block, there is a hexagonal hole. We need to put the condition block inside the hole.
- 4) Codey Rocky will assess whether the color is green or not (or whether there are any items in its view).

2. Challenge: The Racing Game II

In today's lesson, the racing game continues. During the race, Codey Rocky needs to refuel at service stations for plenty of times. However, the ways it pulls into the station might vary from time to time. For instance, its head might face towards the left, the right or face forward:



Ask students:

- 1) What should Codey Rocky do to ensure that it can re-enter the race track in any case?
- 2) How should we write programs to ensure that Codey Rocky is able to re-enter the race track in any case?

Give students some tips: "You might use the following coding blocks. You can use them multiple times in your programs."



- 3) How to make your program concise? You can give students a hint by telling them they can use the **repeat** block.



Sample answers:

- 1) There are three situations. When Codey Rocky faces to the left, the thing it should do is to turn left 90 degrees; when Codey Rocky faces to the right, the thing it should do is to turn right 90 degrees; when Codey Rocky faces forward, the thing it should do is to turn 180 degrees or turn left(right) 90 degrees two times;

- 2) Ask students to figure out the solution: Use the **if statement** block three times – if there is an obstacle ahead, then Codey Rocky turns right 90 degrees until done;

- 3) Repeat the code "**if obstacles ahead then turn right 90 degrees**" three times.

Explain the Challenge Card:

Give a brief description of the challenge in case students might fail to notice some important details.

Challenge 1—Service Station

During the race, Codey Rocky needs to refuel at service stations for plenty of times. However, the ways it pulls into the station might vary from time to time. Write programs to ensure that Codey Rocky is able to re-enter the race track in any case after it refuels.

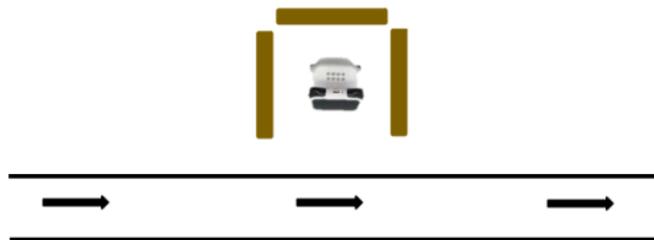
Use books or other items to encircle your Codey Rocky as shown below. This is to simulate a scene in which Codey Rocky pulls over at a service station. The position of Codey Rocky will be one of the following randomly:



Write programs: when the button A is pressed, if Codey Rocky meets an obstacle, it will keep making turns until it finds the exit. You need to use the if statement block more than once.



In front of the Exit is the race track. Once finding the Exit, Codey Rocky will turn left and run forward at its top speed.



Challenge: Give Codey Rocky facial expressions, sounds and lights effects.

Sample program:

```
when button A is pressed
if obstacles ahead? then
  turn right 90 degrees until done
if obstacles ahead? then
  turn right 90 degrees until done
if obstacles ahead? then
  turn right 90 degrees until done
move forward at power 50 % for 1 secs
turn left 90 degrees until done
move forward at power 100 %
```

Tips:

1) Remind students they should make the IR color sensor face forward.



Infrared Color Sensor
Multiple sensors are integrated into one single Infrared Color Sensor, including a color sensor, a grayscale sensor, an IR proximity sensor and more.

2) You can decide whether to have students work on the following extension task depending on the progress.

Extension task : The Tunnel

Introduction:

Codey Rocky is on track. But there's a new problem. There is a tunnel ahead and Codey Rocky has to turn on the RGB indicator and slow its speed.

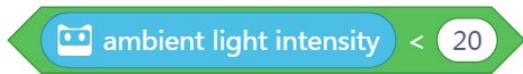
At the bottom right corner of Codey is a black dot. The black dot is the Light Sensor which is used to measure the light intensity of environments. When the light intensity is less than one specific value, the RGB indicator will turn white. And Codey Rocky will lower its speed.

Ask students: what is the condition in this case?

Sample answer: whether the light intensity is less than one specific value.

Explain to students: The hexagonal block we used in the previous tasks can help us assess

whether the **if statement** is true or false, for instance, whether there is an obstacle ahead or not, whether the color is red or not. Besides, we use the Comparison Operators block to evaluate two values. For example, the block "**when light intensity < 20**" helps us assess whether the light intensity exceeds 20. If the light intensity is below 20, then the result will be true. Otherwise, the result will be false.



The hexagon blocks return **Boolean** values (with only two possible values: true or false). A Boolean has only two values, 1 if true and 0 if false. If the if statement is true, then it will return the value - true. Otherwise, it will return the value - false. Therefore, we also refer to the hexagon blocks as Boolean blocks.

Demonstration:

1) Drag this comparison block from the Operators category;



2) Drag the "**ambient light intensity**" block into the left dent;



3) Input a value in the right-hand dent;



Have students complete the following challenge and share what they've done with the whole class.

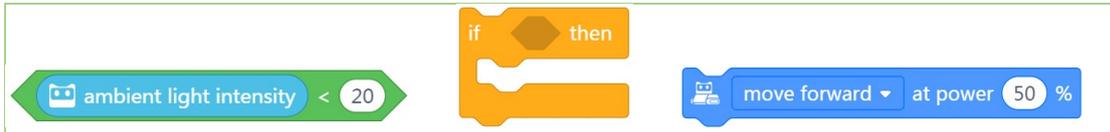
Challenge 2 —The Tunnel

When entering a dark tunnel, Codey Rocky will turn on the indicator light and slow its speed.

- Stick a piece of black paper on the Light Sensor of Codey Rocky.

- When the button A is pressed, Codey Rocky will move forward at its top speed.

- If the light intensity is below 20, Codey Rocky will turn on its white RGB indicator and drive at a lower speed.



Challenge: Add sounds and facial expressions to Codey when it's in a dark environment.

Have students consider one question: How to make Codey Rocky more sensitive to a dark environment? In other words, how to make Codey Rocky turn on the RGB indicator instantly the light intensity gets low? (sample solution: set the value as 40)

3. Presentation

You can show some well-done projects or have volunteers present their works. When sharing the projects, students are supposed to give their answers to the questions proposed by you.

After one student presents his or her project, you can invite some students to comment on the project (what is good about the project and where to improve). Then give your own comments.

4. Wrap up

You need to give a summary of today's lesson:

- a. How to nest the **if statement** block inside the **repeat** block;
- b. The **Comparison Operators** blocks;
- c. Boolean values;
- d. Where the Light Sensor is located;

5. Students' Self-review

Please find the attached self-review worksheets. Hand out the copies and ask students to spend a few minutes filling the worksheets.

Lesson 9 Volume Bar

Lesson Plan

Overview:

Complete challenges by nesting the **if statement** block inside the **forever** block.

Teaching Objectives:

1. Learn how to nest the **if statement** block inside the **forever** block in your code;
2. Know how to accomplish challenges using the **if...then...else** block.

Preparation:

1. Digital challenge cards (or copies);
2. One Codey and a Bluetooth dongle (or the USB cable) per student but it's fine if 2 or 3 students share one set;
3. A computer with installed mBlock 5 per student but it's fine if 2 or 3 students share a computer.

Prior Knowledge:

1. Mastered how to use the **if statement** block;
2. Knew how to use the **forever** block;
3. Grasped how to find bugs and fix them.

Teaching Procedure:

1. Review:

Ask students:

- 1) Can you give some examples of comparison operators blocks?
- 2) What is a Boolean value? Why are the hexagonal blocks also called Boolean blocks?
- 3) Where is the Light Sensor located?

Sample answers:

- 1) There are three types of comparison operators blocks: $< = >$;
- 2) A Boolean has only two values, 1 if true and 0 if false. If the **if statement** is true, then it will return the value - true. Otherwise, it will return the value - false. Therefore, we also consider the hexagon blocks as Boolean blocks;
- 3) The Light Sensor is at the bottom right corner of Codey Rocky. It's a black dot.

2. Challenge: Volume Bar

In the last lesson, we learned how to nest the **if statement** block inside the **repeat** block. Today, we will learn to accomplish a new challenge by combining the **forever** block with the **if statement** block. You'll create a Volume Bar of your own and learn how to make it fluctuate in response to the sound volume.

Codey Rocky wins the game and people are applauding. The volume bar on the screen will change based on the loudness of the applause.

Define the volume range:

- 1) when volume > 20;
- 2) when volume falls among the range of 10~20;
- 3) when volume <10.

The higher the loudness is, the higher the volume bar will be. Remind students of one thing: The block "and" will be used for assessing whether the volume falls among the range of 10~20.

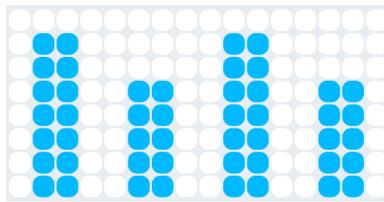
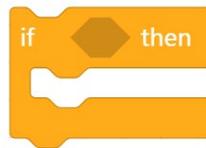


Have students accomplish the challenge in pairs or on their own. To ensure that Codey Rocky is able to keep track of the volume, we need to encircle the code with the **forever** block.

Challenge 1—Volume Bar

Codey Rocky wins the game and people are applauding. The volume bar on the screen will change based on the loudness of the applause.

- When Codey Rocky starts up, if the loudness exceeds 20, the volume bar will reach its top height.



- If the loudness falls between 10 and 20, the volume bar will go down. In this case, you need to use the "and" block.



- If the loudness is below 10, the volume bar will fall to its lowest height.
- Encircle all the code with a "forever" block to make Codey Rocky keep detecting the sound around it.



- Challenge:** Make the RGB LED indicator change its color based on the loudness.

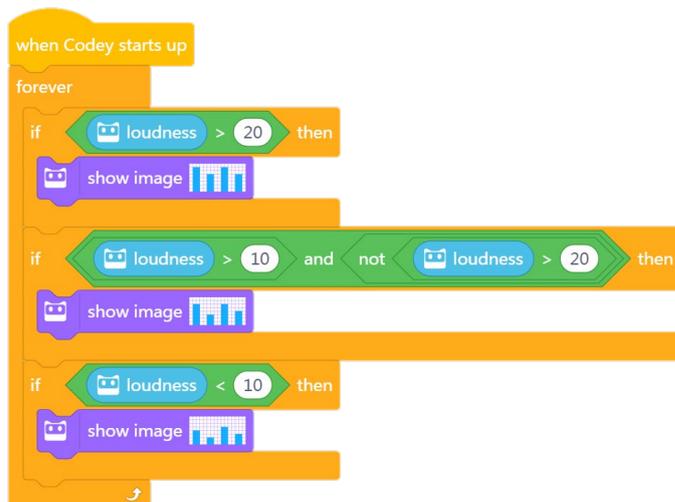
Students might wonder what they should do if they want to define the range as " $10 \leq x \leq 20$ ". Guide student to figure out the solution: use the "not" block in the program. For instance, the code for defining the range " $x \leq 20$ " looks like this:



If we want to evaluate whether the loudness falls among the range of $10 < x \leq 20$, we need to write programs using the combination of blocks as shown below:



Sample program:



Next, get students involved in the following extension tasks:

- 1) The RGB indicator will change its color based on the volume;
- 2) Make the Volume Bar more sensitive by subdividing the value range of loudness.

Invite students to share their projects with the whole class.

Extension task: The Kitten

If time allows and students are bright, you can get students involved in the following extension task.

Give a brief description of the task.

Extension Task -The Kitten

Imagine Codey is a cute cat. When you approach, it will follow you; otherwise, it will stay there waiting for you.

- The IR Color Sensor is used to measure the intensity of reflected IR lights. The closer an item is to Codey Rocky, the higher the light intensity will be.



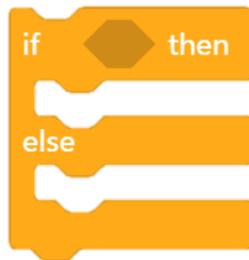
Infrared Color Sensor

Multiple sensors are integrated into one single Infrared Color Sensor, including a color sensor, a grayscale sensor, an IR proximity sensor and more. You can toggle the sensor up or down to make Codey Rocky perform a variety of tasks, like avoiding obstacles, following lines.

- When Codey Rocky starts up, the LED Panel will display the intensity value of the reflected infrared light.



- Put the show block inside the forever block. In this way, Codey Rocky is able to monitor the reflected infrared lights in real time.
- Add an **if statement** block to the end of the show block inside the **forever** block. If the light intensity is beyond a certain value, (customizable) Codey Rocky will run forward at its top speed. Otherwise, Codey Rocky will just stay there. In this case, you need to use the "if...then...else" block and the Operator block ">".



- Challenge:** If the loudness exceeds one certain value, Codey Rocky moves forward or turns left at the greatest power.

Clarify the following points to students:

1. When using the "show" block to display the light intensity value on the LED screen in real time, we should avoid using those blocks including time limits, such as "move forward for 1 sec", "turn left until done", "play sound until done". This is because these blocks will lead to the failure of Codey Rocky to monitor data in real time.

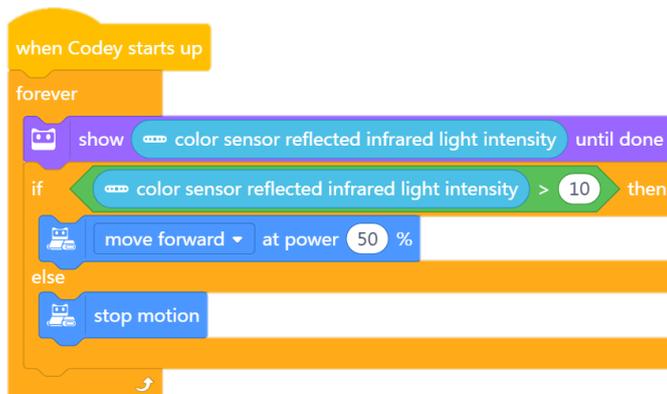
2. Through monitoring and displaying the reflected infrared light intensity in real time, we can customize the conditionals. That is to say, we can when the reflected infrared light intensity exceeds a certain value will Codey Rocky move forward.

3. Make sure all the programs are surrounded by the "**forever**" block;

4. Make sure the IR Color Sensor faces forward when it is sensing obstacles.

Guide students through figuring out the answer: **What if the IR color sensor faces down?**

Sample Program:



3. Presentation

You can show some well-done projects or have volunteers present their works. When sharing the projects, students are supposed to give their answers to the questions proposed by you.

After one student presents his or her project, you can invite some students to comment on the project (what is good about the project and where to improve). Then give your own comments.

4. Wrap up

You need to give a summary of today's lesson:

a. To enable Codey Rocky to monitor the environment in real time, we can use a combination of the **if statement** block and the **forever** block;

b. How to use the "**and**" block;

5. Students' Self-review

Please find the attached self-review worksheets. Hand out the copies and ask students to spend a few minutes filling the worksheets.

Lesson 10 Good Morning! Function

Lesson Plan

Overview:

Use Functions to create a boot program for Codey.

Teaching Objectives:

1. Understand the concept of Functions.
2. Grasp how to apply the Function block (My Block) in your code.
3. Create a function and call the function to give Codey a custom boot animation.

Preparation:

1. Function paper strips (you can print the attached paper strips);
2. A whiteboard and a whiteboard marker (or you can use a blackboard and chalks);
3. One Codey and a Bluetooth dongle (or the USB cable) per student but it's fine if 2 or 3 students share one set;
4. A computer with installed mBlock 5 per student but it's fine if 2 or 3 students share a computer.

Prior Knowledge:

1. Knew how to upload programs to Codey;
2. Mastered how to use the **Event** blocks in programs;
3. Knew how to find bugs and fix them.

Teaching Procedure:

1. Review:

Ask students:

- 1) What is a conditional?
- 2) How did we use the conditional blocks to assess conditions in the last lesson? Ask students if they could come up with any examples.

Sample answers:

1) The **if statement** block is used to help a computer make decisions based on situations. If the condition is true, the instruction will be executed; if the condition is false, the instruction will be skipped or ignored.

2) We used the conditional blocks to assess the following situations:

- Whether the color detected by the color sensor is green;
- Whether the light intensity exceeds 20;
- Whether the reflected infrared light intensity exceeds 10.

2. Explain the New Concept — Functions

Take advantage of the PPT and clarify the concept Functions to students.

- 1) Introduce the new concept — **Functions**:

In daily life, we often need to take a set of steps do some things more than once, like washing hair. In most cases, washing our hair needs three steps: shampoo your hair, massage hair to form foam and rinse foam off. But if we don't use a phrase "**wash hair**" to describe the set of steps, how will the situation be like?

1) When a friend comes to ask the panda out, it'll say: "I have to shampoo my hair, massage my hair to form foam and rinse off the foam. So, wait."

2) When your mom asks you to have breakfast, you'll say: "I'm shampooing my hair, massaging my hair to make it foam and rinsing the hair. Wait.";

3) Got caught in the rain, you murmur to yourself, "I have to take shampoo, massage my hair to form foam and rinse off the foam."

But in daily life, we don't need to describe each step. We often give the group of steps one single name. We use the name to represent the whole sequence of steps and call the name when referring to the steps. For example, we name the sequence of actions above as **wash hair**.

The situation will be totally different:

1) When your friend comes to ask you out, you are saying this time, "I am about to wash hair. Wait.";

2) When your mom is calling you to take breakfast, you tell her, "I'm washing hair, wait.";

3) Got caught in the rain, you murmur to yourself, "I've got to wash hair."

The process of using a single name to represent a group of actions is called **defining functions**. The name we use to represent a sequence of instructions is what we call a function. For example, "wash hair" can be regarded as a function. It represents a sequence of actions including "shampoo the hair, massage the hair to make it foam, rinse the foam". In programming, we use a function to name a set of instructions and we can call the function whenever we need to use the instructions in the code.

Tips:

The function name should be easy to understand, which otherwise could lead to some trouble when we call the function. Take washing hair as an example: If we give the sequence of steps a function name "have dinner", it could be misleading. Try to imagine. When your friends come to ask you out, you tell them "I'm having dinner" while you are actually washing hair in the bathroom. Undoubtedly, this will make your friends confused.

To create a function, you need to first give the function a proper name. Then, you need to add commands under the function to define the function.

Have students play a game. Help them gain a better understanding of how functions can work.

3. Game—Good Morning! Functions

Introduction:

The *Morning Functions*, as the name suggests, are a group of functions that indicate what you do in the morning every day, from getting up to leaving home for school. For instance, after you get up in the morning, you need to brush teeth—>wash face—>put on clothes—>put on shoes—>take breakfast—>carry the schoolbag—>leave home—>go to school. Each function includes a series of

actions. For instance, the function "brush teeth" refers to the following actions: take the toothbrush, squeeze the toothpaste onto the toothbrush, brush the teeth all around, rinse the mouth. Then have students consider a proper name to represent the sequence of actions. After defining the function, students can call the function to test how it works.

Game steps and teaching preparation:

1) The teacher prints the first chart in the file, The Morning Functions. Cut the print into pieces along the dash lines. These are called paper strips of functions;

2) The teacher writes down the following words on the whiteboard: **Event -When the alarm clock goes off;**

3) Have each student pick the paper strips randomly;

4) Make the game rules clear: "Everyone has a piece of paper strip in hand, and each of them represents a group of actions. Give the group of actions a name yourself; I will later call the functions under the event **When the alarm clock goes off**; When the function of yours is called, you need to read aloud the detailed steps and act them out";

5) Have students read out loud the names of those functions. Then list the names on the blackboard;

6) After putting down all the functions names on the whiteboard, the teacher needs to ask students in what order those functions should be called. Then the teacher lists the names of those functions under the event "When the alarm clock goes off" in the order as students request;

7) Ask students to figure out why some functions are not called and why some functions are called multiple times. Have students complete the programs. (The reason might be that some functions don't belong to the process). The following chart shows some sample functions:

<p><u>Oversleep</u> cover head with the quilt; wait for 5 seconds; uncover the quilt; stretch out one hand; turn off the alarm clock.</p>	<p><u>Have a look at the time</u> take up the alarm clock; have a look at the time; put down the clock.</p>	<p><u>Get up</u> stretch yourself; uncover the quilt; sit up.</p>
<p><u>Put on slippers</u> put on the slipper to the left foot; put on the slipper to the right foot; stand up.</p>	<p><u>Walk</u> lift the left foot; stretch forward the left foot and step on the ground; lift the right foot; stretch forward the right foot and step on the ground.</p>	<p><u>Brush teeth</u> take the toothbrush; squeeze the toothpaste; rinse the mouth; brush the teeth all around; rinse the mouth.</p>
<p><u>Wash up</u> turn on the tap; cup the water with hands; wash up with the water; } ^{3times} turn off the tap; dry your face with the towel</p>	<p><u>Take off the coat</u> If it's a T-shirt, take it off from top to bottom; If the coat has buttons, unbutton it first, then the left arm comes out of the left sleeve, next the right arm comes out of the right sleeve.</p>	<p><u>Take off pants</u> pull down the pants; the left leg comes out of the pants; the right leg comes out of the pants.</p>
<p><u>Take clothes off</u></p>	<p><u>Put on the coat</u></p>	<p><u>Put on pants</u></p>

take off the coat; take off pants.	get the left arm into the left sleeve; get the right arm into the right sleeve; if it's a T-shirt, get your head into the shirt; if the coat has buttons, fasten the buttons.	get the left leg into the pants; get the right leg into the pants.
<u>Dress up</u> put on the pants; put on the coat.	<u>Carry the schoolbag</u> get the right arm through one strap; get the left arm through the other strap; shoulder the schoolbag.	<u>Lace the shoes</u> make a circle of the shoelace with the left hand; make a circle of the shoelace with the right hand; tie the two circles
<u>Put on shoes</u> get the left feet into the left shoe; get the right feet into the right shoe; lace the left shoe.		

Tips:

1) The three functions "Take clothes off" "Put on clothes" and "Put on shoes" include nested functions, so it might be a little bit challenging for students to understand the concept of nested functions. In consideration of this, you could skip the nested functions. That is to say, we only use the paper strip "Dress up" in the game, ignoring the paper strips of "Put on the coat" and "Put on pants". However, if students are bright enough to understand the concept of nested functions, then it's fine to introduce the concept to students. A nested function refers to the process where a function is called by another function;

2) In some cases, the function names might be inaccurate. For example, students might name the group of actions "lift the left foot; stretch forward the left foot and step on ground; lift the right foot and step on ground" as "Gymnastics". Have students read the steps aloud, act them out and rename the function. Next, have students call the renamed function again in the program.

3) Sometimes, the function names might be hilarious. For example, students might give the group of actions "cover head with the quilt; wait for 5 seconds; uncover the quilt; stretch out one hand; turn off the alarm clock" an amusing name, like "Annoying". Anyway, the name doesn't have to be "Oversleep". In a word, you need to encourage students to bring whimsy into the class;

4) If students are not willing to act out or the time is not enough, skip the acting out part. Just let students read out the actions.

Wrap up:

To take advantage of functions, you need to take three steps:

1. Create a function and give the function a name;
2. Define the function;

3. Call the function in the program.

By playing games, we already knew what a function is and mastered how to use functions to carry out a series of actions. Now let's dive into the world of programming and start to learn how to use the Functions blocks in our code!

4. Demonstration

Have students work in pairs or on their own to complete the challenges.

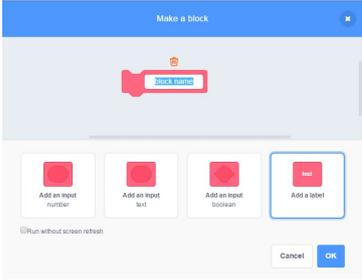
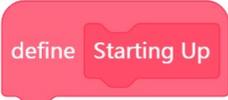
Unveil the task: Create a startup function for Codey Rocky, making sure the function will automatically run when Codey Rocky starts up.

Guided practice: Open the mBlock 5 and connect Codey to the software. Follow the instructions to complete the challenge.

**Challenge 1-
The Startup Function**

Create a startup function for Codey Rocky.

- Click My Blocks at the category bar and select Make a Block.

- Click Make a Block. Create a function and give it a name.

- Then, the define starting up block will appear in the Scripts area.

- What programs should be run when Codey Rocky starts up? Design programs under the **define starting up** block.
- After defining the function, you can directly call the function by adding the **starting up** block to the bottom of the event block when Codey Rocky starts up.


Sample program:



5. Imitate and create

Have students use functions to create a startup animation for Codey.

Students can:

- 1) Improve on the sample project. You can change the animation or the sound;
- 2) Select an image from the built-in images and refine the image as you would like. Then create their own animation using the image.
- 3) Give their own animation a compelling storyline.

Have students complete the tasks above in the form of pair programming.

Tell students that there will be a presentation session. Students are expected to share their projects with the whole class and give their answers to the following questions:

- 1) What is your project about?
- 2) Did you come across any problems? How did you solve them?

6. Presentation

You can show some well-done projects or have volunteers present their works. When sharing the projects, students are supposed to give their answers to the questions proposed by you.

After one student presents his or her project, you can invite some students to comment on the project (what is good about the project and where to improve). Then give your own comments.

7. Wrap up

You need to give a summary of today's lesson:

Remind students what a function refers to in programming: In programming, **Function** is a custom coding block. Function refers to a set of instructions that can be called repeatedly in the code. Tell students that functions are important because coders can use functions to make the code neat.

8. Students' Self-review

Please find the attached self-review worksheets. Hand out the copies and ask students to spend a few minutes filling the worksheets.

Lesson 11 The Tiny Patroller I

Lesson Plan

Overview:

Use functions in the code to make Codey Rocky perform tasks and apply mathematical knowledge in the game.

Teaching Objectives:

1. Master how to use the function blocks (My Block) in your code;
2. Know how to create a function and call the function;
3. Able to complete the challenge by applying functions and mathematical knowledge.

Preparation:

1. Maps (students working in pairs can share one map);
2. One Codey and a Bluetooth dongle (or the USB cable) per student but it's fine if 2 or 3 students share one set;
3. A computer with installed mBlock 5 per student but it's fine if 2 or 3 students share a computer;
4. Pencils, rulers, notebooks and more.

Prior Knowledge:

1. Knew how to upload programs to Codey;
2. Understood the concept of Functions and mastered how to use functions;
3. Knew how to find bugs and fix them;
4. Knew the relationship between distance, speed and time.

Teaching Procedure:

1. Review:

Ask students:

- 1) What is a function?
- 2) In the last lesson, how did you use functions?

Sample answers:

1) In programming, **Function** is a custom coding block. Function refers to a set of instructions that can be called repeatedly in the code.

2) Gave a sequence of actions a name and called the function when playing games; we used functions to create the startup animation for Codey.

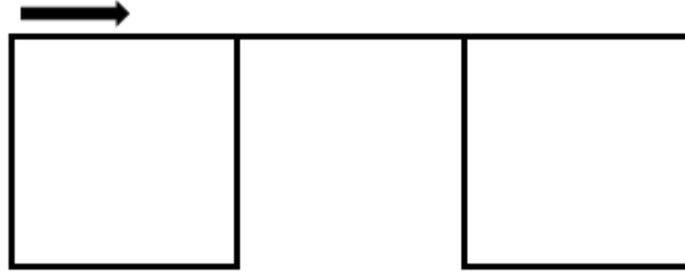
2. Challenge: Patrol the 1st Floor (Beginner)

Take advantage of the map and the Challenge Card to make the rules clear to students. Tell students that they are going to learn how to use functions to accomplish the following challenge;

About the challenge:

Imagine Codey Rocky is a security guard. It is patrolling the passages in the building to make sure that all the properties safe. Now, it's patrolling on the 1st floor. Here's the roadmap of the 1st

floor:



How should we write programs to make sure that Codey Rocky can patrol along the route as shown in the map above? We need to use the functions that we just learned yesterday. Take on the challenge now!

Challenge : Patrol the 1st Floor

Give a brief description of the challenge in case students might fail to notice some important details. Or have students accomplish the challenge as the Challenge Card instructs and share their outcomes with the whole class.

Challenge 1-
Patrol the 1st Floor

Imagine Codey Rocky is a security guard. It is patrolling the passages in the building to make sure that all the properties safe. Now, it's patrolling on the 1st floor.

- Design programs to make Codey Rocky follow the black lines as shown below.

- Create a function and name it as Square.

- You might need to use the following blocks.

 move forward at power 50 % for 1 secs

 turn right  15 degrees until done

- Figure out how much time it takes Codey Rocky to drive along the square-shaped line and along the linking line.
 - ✓ Measure the length of the side of the square.
 - ✓ Measure the length of the linking line.
 - ✓ Calculate the driving distance that Codey Rocky can reach per second.
(For instance, if Codey Rocky is programmed to keep moving forward at power of 50% for 1 second, what is the driving distance then?)
 - ✓ Calculate how much time it takes Codey Rocky to drive along a square.
 - ✓ Calculate how much time it takes Codey Rocky to drive along the linking line.
- Call the function Square two times under the event block when button A pressed.
- Challenge: If the loudness exceeds one certain value, Codey Rocky moves forward or turns left at the greatest power.

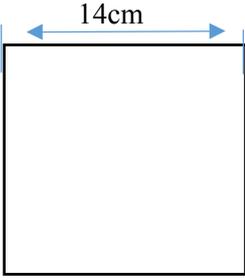
Instructions for students:

1) The first step is to measure the length and width of the roadmap. Based on the measurements, you need to figure out how much time it takes Codey Rocky to complete the route.

 move forward at power 50 % for 1 secs

In the coding block above, there are two variables. One is the power and the other is the time. In most cases, we only need to adjust either of the two variables. For instance, we define the power as 50% and keep changing the time; or we give the time a specific value and keep changing the power level.

For example: The side length of the square in the map is 14cm and Codey Rocky is programmed to move forward at power 50%. Then the speed of Codey Rocky is 7cm/sec, so the time it will take Codey Rocky to complete one side will be 2 seconds ($14 \div 7 = 2$). This method applies to the rest of the sides. Use this method to calculate how much time it will take Codey Rocky to complete the square.

	<p>The side length of the square is 14cm. If Codey Rocky is programmed to move forward at power 50%, then it will take Codey Rocky 2 seconds to complete one side.</p>
	<p>After Codey Rocky completes one side of the square, it needs to turn right 90 degrees to move on to the next</p>

2) the motor of Codey Rocky is a DC motor so Codey Rocky is not able to make turns or follow lines precisely. In this case, it's fine that Codey Rocky follows the line in a rough manner;

3) Codey Rocky is programmed to move forward and turn right. After Codey Rocky repeats the steps four times, it will go back to the initial position;

4) Since there are two squares in the roadmap, you are supposed to call the function Square at least twice in the code.

Sample program:

```

when button A is pressed
  Square
  move forward at power 100 % for 2 secs
  Square

define Square
  repeat 4
    move forward at power 100 % for 1 secs
    turn right 90 degrees until done
  
```

3. Presentation

You can show some well-done projects or have volunteers present their works. When sharing the projects, students are supposed to give their answers to the questions proposed by you.

After one student presents his or her project, you can invite some students to comment on the project (what is good about the project and where to improve). Then give your own comments.

4. Wrap up

You need to give a summary of today's lesson:

Remind students what a function refers to in programming: In programming, Function is a custom coding block. Function refers to a set of instructions that can be called repeatedly in the code. Tell students that functions are important because coders can use functions to make their code neat. In addition, considering we need to do operations when writing programs, we might use mathematical knowledge.

5. Students' Self-review

Please find the attached self-review worksheets. Hand out the copies and ask students to spend a few minutes filling the worksheets.

Lesson 12 The Tiny Patroller II

Lesson Plan

Overview:

Increase the difficulty of last lesson's task by complicating the map. Have students learn how to write programs using more complicated functions and mathematical knowledge.

Teaching Objectives:

1. Master how to use the function blocks (My Block) in your code;
2. Know how to create a function and call the function;
3. Able to complete the challenge by applying functions and mathematical knowledge.

Preparation:

1. Maps (students working in pairs can share one map);
2. One Codey and a Bluetooth dongle (or the USB cable) per student but it's fine if 2 or 3 students share one set;
3. A computer with installed mBlock 5 per student but it's fine if 2 or 3 students share a computer;
4. Pencils, rulers, notebooks and more.

Prior Knowledge:

1. Knew how to upload programs to Codey;
2. Understood the concept of Functions and mastered how to use functions;
3. Knew how to find bugs and fix them;
4. Knew the relationship between distance, speed and time.

Teaching Procedure:

1. Review:

Ask students:

- 1) What is a function?
- 2) In the last lesson, how did you use functions?

Sample answers:

1) In programming, Function is a custom coding block. Function refers to a set of instructions that can be called repeatedly in the code.

2) We used functions to make Codey Rocky perform task as programmed; we learned how to calculate how much time it took Codey Rocky to complete the route and how to adjust programs

2. Challenge: Patrol the 2nd floor (Advanced)

Show the map to students and explain the rules:

1) Tell students that they are going to learn how to use functions to accomplish a more complicated task.

About the challenge:

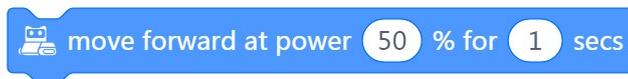
In the last lesson, we already had Codey Rocky patrol the 1st floor. In today's lesson, the task might be more challenging. Codey Rocky has to patrol the 2nd floor. Here's the roadmap of the 2nd

- You need to create two functions, Upper square and Bottom square.
- Figure out how much time it takes Codey Rocky to drive along the square-shaped line and along the linking line.
 - ✓ Measure the length of the side of the square.
 - ✓ Measure the length of the connecting line.
 - ✓ Calculate the driving distance that Codey Rocky can reach per second. (For instance, if Codey Rocky is programmed to keep moving forward at power of 50% for 1 second, what is the driving distance then?)
 - ✓ Calculate how much time it takes Codey Rocky to drive along a square.
 - ✓ Calculate how much time it takes Codey Rocky to drive along the connecting line.
- Call the function Square two times under the event block when button A pressed. Using the repeat block will make your code concise.
- Challenge:** Give Codey Rocky facial expressions, sounds and light effects.

Instructions for students:

1) Remind students that they should create two functions, Upper square and Bottom square, in their code :

2) Measure the length and width of the roadmap. Based on the measurements and the speed of Codey Rocky, students need to figure out how much time it takes Codey Rocky to complete the route.

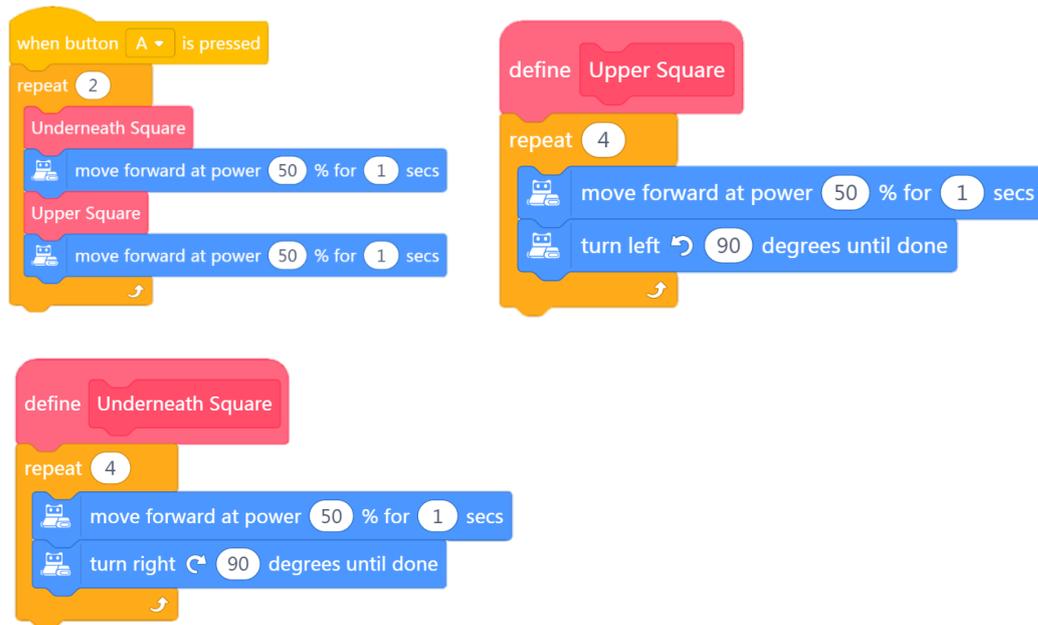


3) The motor of Codey Rocky is a DC motor so Codey Rocky is not able to make turns or follow lines precisely. In this case, it's fine if Codey Rocky follows the line in a rough manner;

4) There are multiple ways to make Codey Rocky take the route as shown above. Have students work on their own to figure out the solution; or have students complete the challenge using the following pseudocode.

Bottom square: move forward 1 second turn right by 90°	Repeat four times	Upper square: move forward 1 second turn left by 90°	Repeat four times
When button A pressed: bottom square move forward 1 second upper square move forward 1 second	Repeat two times		

Sample program:



3. Presentation

You can show some well-done projects or have volunteers present their works. When sharing the projects, students are supposed to give their answers to the questions proposed by you.

After one student presents his or her project, you can invite some students to comment on the project (what is good about the project and where to improve). Then give your own comments.

4. Wrap up

You need to give a summary of today's lesson:

Remind students what a function refers to in programming: In programming, Function is a custom coding block. Function refers to a set of instructions that can be called repeatedly in the code. Tell students that functions are important because coders can use functions to make their code neat and create engaging projects.

5. Students' Self-review

Please find the attached self-review worksheets. Hand out the copies and ask students to spend a few minutes filling the worksheets.

Lesson 13 The Squirrel's Nuts Box

Lesson Plan

Overview:

Use variables to make Codey Rocky perform tasks as programmed.

Teaching Objectives:

1. Understand the concept of Variable.
2. Learn how to create a variable and use the variable in the code to create projects.

Preparation:

1. A whiteboard and a whiteboard marker (or you can use a blackboard and chalks) ;
2. One Codey and a Bluetooth dongle (or the USB cable) per student but it's fine if 2 or 3 students share one set;
3. A computer with installed mBlock 5 per student but it's fine if 2 or 3 students share a computer.

Prior Knowledge:

1. Knew how to upload programs to Codey;
2. Mastered how to use the Event blocks and the concept of Function;
3. Knew how to find bugs and fix them.

Teaching Procedure:

1. Review

Ask students:

- 1) How were functions used in a program in the previous lesson?
- 2) What's the difference between a function and a loop?

Sample answers:

- 1) Functions were used to make Codey Rocky walk in the pattern of squares and dance.
- 2) A loop is a piece of code that is repeated in one specific place while a function can be called over and over again in different place in the code.

2. Explain the New Concept

Take advantage of the PPT to explain to students what a variable refers to and how to use variables.

- 1) Tell students that they are going to learn a new concept in today's lesson: Variable.

A variable is a container which stores information that can be changed.

Imagine that a variable is a box. You can put a value into the box and replace it with another value anytime. It works like a scoreboard which is used to record the scores during a contest. As the contest proceeds, the scores change so you can change the scores on the board as well.

In computer programming, a variable is a storage location paired with an associated name. When a variable is created, the computer will leave a storage location to store the variable and give the location a symbolic name. By calling the variable name, programs can read and change the value in the storage location.

It's important to give a variable a proper and simple name because this helps to ensure the variable can work properly. For instance, if we name the scoreboard as "Red Team", then the scoreboard will represent the scores that the red team wins in the match. When the red team wins a three-point shot, the number on the scoreboard will change accordingly. A new value will replace the previous one

The purpose of a variable is to **assign values**, **change information** and **compare values**. The first one is **Assignment**. You can copy a value into a variable. For example, a variable "Red Team" is created to store the scores of the red team, and the initial value of the variable is set to be 0. The second one is **Changing Information**, which means you can change the value stored in the variable by adding, subtracting, multiplying or dividing them. For instance, if the red team makes a successful shot, then the value in the variable "Red Team" will increase by 2. The third one is **Comparing values**. You can read the current value and compare it with another value. By comparing values, you are able to make decisions. For instance, the red team wins when the scores of the red team exceed the scores of the blue team.

You can find plenty of examples of variables in real life. One example can be the Health Points of characters in games. Similarly, the health points of characters are changing over time in games. The amount of money in your wallet and the number of candies can be both examples of variables. We can read the values and change the amount of money and candies by adding or subtracting them. Moreover, we can compare the values. For instance, if the amount of money is greater than the price of a certain product, then we know we can afford the product; if the amount of candies in the jar exceeds a certain number, then we are able to share them with friends.

3. Game -The Squirrel's Nuts Box

Have students play a game to gain a better understanding of the concept.

Storyline:

There is a little squirrel. It has a nuts box. Every day, the squirrel puts nuts into the box and takes nuts out of the box, so the number of the nuts keeps changing all the time. One day morning, the little squirrel gets up, opens the box and finds out that there are 10 nuts left. The amount of the nuts will change based on different situations.

For instance, the squirrel is hungry and eats two nuts, so there are 8 nuts left. Then the squirrel picks another 5 nuts so there are 13 nuts in total. Now let's play a game. Try figuring out how many nuts are left.

Game steps and teaching preparation:

1) The teacher writes down the number 10 on the blackboard and tells students there are 10 nuts left in the box.

2) The teacher prepares some paper strips and each paper strip describes a situation of how the number of nuts changes (as shown below).

3) Have students take turns picking a paper strip out of the box. (If there are too many students, invite some of them to pick paper strips randomly; if there are only a few students, then all the students have a chance to pick paper strips.)

4) Have students read out what the paper strips say and navigate them to figure out the amount of the nuts left based on the information. Tell students to write down the result on the blackboard.

In the morning, the little squirrel eats 2 nuts for breakfast.
If the maximum temperature exceeds 5°C, the squirrel will pick 10 nuts in the wild.
At noon, the squirrel eats 3 nuts for lunch.
In the afternoon, a little bird pays a visit and gives the squirrel 5 nuts.
Today is the monkey's birthday. The squirrel gives him 3 nuts as a gift.
In the evening, a rat steals 4 nuts.
The squirrel plays the rock-paper-scissors game with a friend three times. If he loses the game, he gives away a nut; if he wins the game, he gets a nut; if it's a draw, then no gain or loss for both sides.
In the evening, the squirrel eats only 1 nut to keep fit.
It's hot today. The squirrel buys a straw hat from Aunt Bear. It costs him 7 nuts.
If the amount of the nuts in the box is less than 5, the squirrel will pick 10 nuts outdoors.
If the amount of the nuts in the box exceeds 12, the little squirrel will turn in a circle happily.

Tips:

The order of how the paper strips are picked affects the calculated results. In some cases, students might get a negative number after they calculate the nuts. If students have already learned about the concept of a negative number, tell them to replace the previous value with a negative number; if they haven't learned the concept yet, then the teacher should revise the conditions on the paper strips to give the squirrel opportunities to gain more nuts.

Wrap up:

- Ask students the following questions:

1. What's the variable in the game? How did you name it?
2. What's the initial value for the variable?
3. How did you do operations with the variable?
4. How did you make a comparison between the variable and other values?

Sample answers:

1) The amount of the nuts is changing so the variable in the game is the Nut. Give students freedom to name the variable. Of course, the teacher can give some hints: the number of nuts or Nut

2) At the very first beginning, there are 10 nuts in the box so the initial value of the variable is 10.

3) The operations of addition and subtraction were applied in the game.

4) Comparisons were made between the current number of the nuts and the requested number.

For instance, if the number of the nuts is 5, then we have to tell whether the number is greater than

- What is a variable?

Imagine that a variable is a box. You can put a value into the box and replace it with another value anytime. When you need to use a variable in your code, the box will be opened and the value that's stored in the box will be taken out. You can do multiple operations with the value to output a new value and replace the new value with the previous one.

4. Demonstration

Help students create, set and use a variable. Then unveil the task and have students accomplish the task by following the instructions on the Challenge Card.

Instructions: Demonstrate how to write the following program:



Instructions:

1) How to create a variable: Open the mBlock 5, go to the category **Variables** and select **Create Variable**.

Name the variable:



2) Set a value:



3) Use the variable:



Have students try the task. Make the rules clear to students.

About the task:

Have students work in pairs or on their own to accomplish the task.

Challenge - Assign a value to the variable

Codey Rocky is standing on the stage and greeting the audience.

Codey Rocky turns left 70 degrees and then turns right 140 degrees. Finally, it returns to the starting point. To make Codey Rocky turn specific degrees, you need to create a variable named "angle".

Set the angle as 70.



Make Codey Rocky turn left by specific degrees as programmed.



Assign a new value to the variable "angle". As programmed, Codey Rocky will turn right 140 degrees and then turn left 70 degrees.

Add facial expressions, sounds and light effects to Codey Rocky.

Tips:

1) After we create a variable, a group of new blocks will be available:

Variable block: 



Block for setting the value:

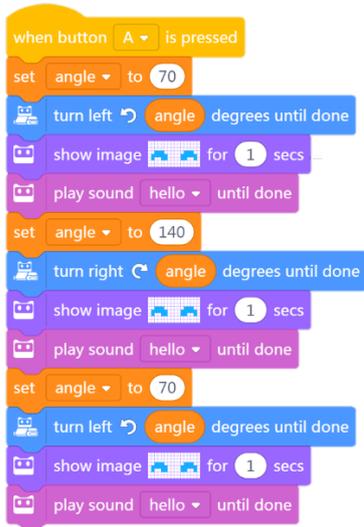


Block for increasing/decreasing the value: (a negative number indicates a decrease)

2) Give the variable a proper name. Make it easy to understand;

3) In the program, the variable is used three times.

Sample program:



Discussion:

I want to make Codey Rocky move forward at a certain speed and make it last for 1 second.

How should I add another variable and how to use it?

6. Imitate and Create

Ask students to use variables to control Codey Rocky. Students can select one of the following tasks:

- 1) Improve on the sample project. Students can reset the value and the facial expressions.
- 2) Students can try different combinations of Event blocks and variables, making Codey Rocky drive in the way as they would like;
- 3) Design a program that includes 3 variables. Use the coding blocks they learned previously to create programs.

Have students complete the tasks above in the form of pair programming.

Tell students that there will be a presentation session. Students are expected to share their projects to the whole class and give their answers to the following questions:

- 1) What is your project about?
- 2) Did you come across any problems? How did you solve them?

7. Presentation

You can show some well-done projects or have volunteers present their works. When sharing the projects, students are supposed to give their answers to the questions proposed by you.

After one student presents his or her project, you can invite some students to comment on the project (what is good about the project and where to improve). Then give your own comments.

8. Wrap up

You need to give a summary of today's lesson:

Remind students that a variable can be considered a container that is used to store information that can be changed. And a variable has three functions: assigning values, changing information and comparing values.

Lesson 14 Mathematical Operations

Lesson Plan

Overview:

Use variables to compare values.

Teaching Objectives:

1. Understand the concept of Variables;
2. Learn how to assign values to variables and how to do operations with the values.

Preparation:

1. A whiteboard and a whiteboard marker(or you can use a blackboard and chinks) ;
2. One Codey and a Bluetooth dongle (or the USB cable) per student but it's fine if 2 or 3 students share one set;
3. A computer with installed mBlock 5 per student but it's fine if 2 or 3 students share a computer.

Prior Knowledge:

1. Knew how to upload programs to Codey;
2. Understood how to use Loops and mastered the concept of Functions;
3. Knew how to create and use variables.

Teaching Procedure:

1. Review

Ask students:

- 1) What is a variable?
- 2) What are the three functions of a variable?

Sample answers:

- 1) A variable is a container that is used to store information that can be changed;
- 2) Assigning values, changing information and comparing values.

2. Challenge: Mathematical Operations

In this session, students are going to take on two challenges. Task 1 is about addition and subtraction; Task 2 is about multiplication and division.

About the task:

When studying mathematics, we often need to do operations. We can also do operations with the variables. It's like we are taking an exam. If we get the answer correct, then we get one point; if we get the answer wrong, then we lose one point.

Instead of simply assigning values to variables, we can also do operations with variables and store the calculation results. For instance, we can embed Operators blocks in the Variables blocks. In addition, we can increase numbers in multiples, like $1 \times 2 = 2$, $2 \times 2 = 4$, $4 \times 2 = 8$ With variables, we are able to simply the math expressions. Therefore, with the help of Codey Rocky, students may find it easier to master the

mathematical operations.

Task 1: Addition and Subtraction

Give a brief description of the challenge in case students might fail to notice some important details.

Or have students accomplish the challenge as the Challenge Card instructs and share their outcomes with the whole class.

Challenge 1- Add and Subtract

Codey Rocky is learning how to do mathematical operations. Now it masters how to add 1 to a value and subtract 1 from a value.

- Codey Rocky would like to start with the number 0, so the value of the variable "number" is set to be 0 when it starts up.

- When the button A is pressed, the number increases by 1.

- When the button B is pressed, the number decreases by 1. A negative number refers to subtracting.

- If you would like to make the LED screen show the calculation result, you can add the block "show () until done" under each event block.

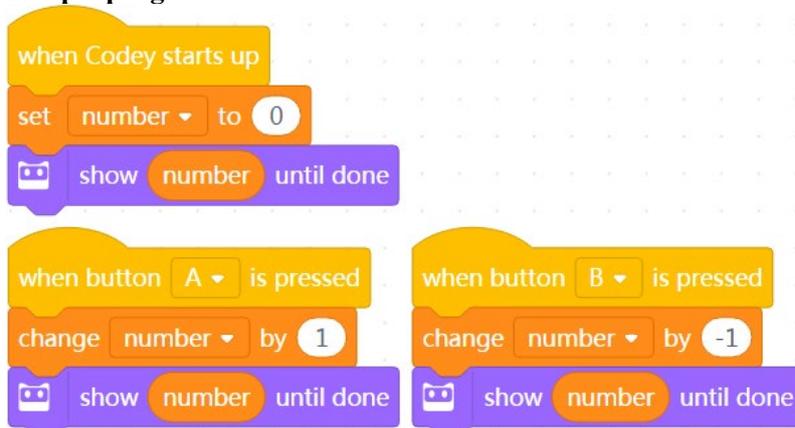

Tips:

1) In the "change (number) by ()" block, a negative number refers to subtracting.



2) The parameter dent has three shapes: hexagon, rectangle, and round rectangle. A variable block can only fit into the round rectangle;

Sample program:



Task 2: Multiply and Divide

Give a brief description of the challenge in case students might fail to notice some important details. Or have students accomplish the challenge as the Challenge Card instructs and share their outcomes with the whole class.

Tips:

1) We need to use the Operators blocks to get calculation results. Then we can replace the

Challenge 2- Multiply and Divide

Codey Rocky also learned about multiplication and division.

- Assign an initial value to the variable.
- When the button A is pressed, the number is multiplied by 2. We need to use the multiply block to get a calculation result. Then, assign the result to the variable.

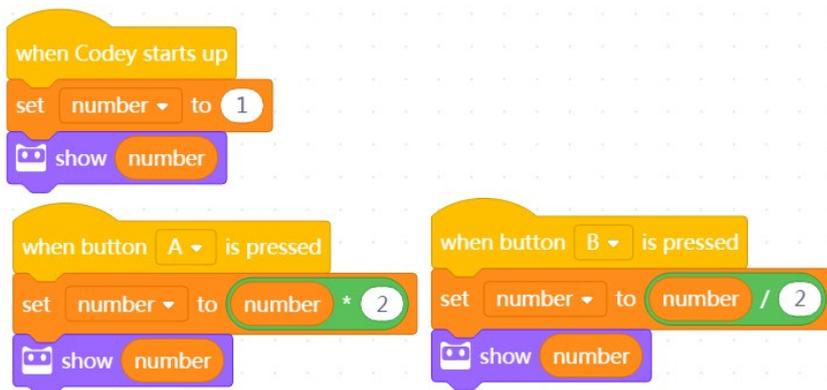


- When the button B is pressed, the number is divided by 2.
- Have the the LED screen show the calculation result each time. There is one thing that needs attention: the LED display can only show a value ranging from -999 to 9999.
- Challenge: Change the initial value and the factor (replace the number 2 with another value). Observe how the calculation result changes.

previous value with a new value by reassigning the calculation result to the variable.

2) Make sure the variable you use under the three different events (when Codey Rocky starts up; when the button A is pressed; when the button B is pressed) is the same one.

Sample program:



Discussion:

If we have a closer look at the values on the LED display, we will notice one thing: once a calculation result appears in the form of decimal, the subsequent calculation results will continue to show up in the form of decimals.

3. Presentation

You can show some well-done projects or have volunteers present their works. When sharing the projects, students are supposed to give their answers to the questions proposed by you.

After one student presents his or her project, you can invite some students to comment on the project (what is good about the project and where to improve). Then give your own comments.

4. Wrap up

You need to give a summary of today's lesson.

We can store a calculation result in a variable and can call the value of the variable repeatedly in the math expressions.

5. Students' Self-review

Please find the attached self-review worksheets. Hand out the copies and ask students to spend a few minutes filling the worksheets.

Lesson 15 The Bomb

Lesson Plan

Overview:

Make Codey Rocky perform tasks as programmed by comparing the values of variables.

Teaching Objectives:

1. Understand the concept of Variables;
2. Know how to increase or decrease the value of a variable.

Preparation:

1. A whiteboard and a whiteboard marker(or you can use a blackboard and chalks) ;
2. One Codey and a Bluetooth dongle (or the USB cable) per student but it's fine if 2 or 3 students share one set;
3. A computer with installed mBlock 5 per student but it's fine if 2 or 3 students share a computer.

Prior Knowledge:

1. Knew how to upload programs to Codey;
2. Understood how to use Loops and mastered the concept of Functions;
3. Knew how to create and use variables.

Teaching Procedure:

1. Review:

Ask students:

1) How do you assign a value to a variable when you are doing operation with the variable?

Sample answer: We can use the Operators blocks to store calculation results in a variable.

2. Challenge: Emergency - Explosion

Take advantage of the PPT to explain the concept of Variable and how to use variables.

In this session, students are expected to take on two challenges.

About the task:

In the last lesson, we learned how to do operations with the values of a variable using the buttons on Codey. In today's lesson, we are going to figure out how to use Loops in the code to keep the value changing. Take timers as an example here. For every 1 second, the number shown on the timer will increase by 1. With the time passing, the number is getting greater. Similarly, we can take advantage of this principle to create a countdown bomb.

Task 1: The Countdown Bomb

Give a brief description of the challenge in case students might fail to notice some important details.

Or have students accomplish the challenge as the Challenge Card instructs and share their outcomes with the whole class.

Challenge 1-The Countdown Bomb

Game rule: Codey is counting down. During the period, anyone who is holding Codey has to give an animal name. The game ends the moment the bomb explodes.

- ❑ When Codey Rocky starts up, the initial value of the variable "time" is set to be 30.
- ❑ When the button A is pressed, the "timer" will start counting down. After 1 second, the value of "time" will decrease by 1.

```

when button A is pressed
  change time by 1
  
```

Repeat the piece of code 30 times.

```

repeat 30
  
```

Add sounds effects to make the game more intense and write code to make the LED screen show how much time is left.

```

play sound step_1
show time until done
  
```

- ❑ During the countdown period, anyone who is holding Codey has to give an animal name and then passes Codey to the next person. Animal names cannot be repeated.
- ❑ When the value of "time" becomes 0, the countdown ends and the bomb explodes. At this moment, the RGB LED lights up in red.
- ❑ Challenge: Reset the initial value of the variable "time" to make the game time longer or shorter.

Sample Program:

```

when Codey starts up
  set timer to 30
  show timer

when button A is pressed
  repeat 30
    wait 1 seconds
    change timer by -1
    play sound step_1
    show timer
  show image [Bomb]
  play sound explosion
  RGB LED lights up [Red]
  
```

Task 2: The Explosive Number

Give a brief description of the challenge in case students might fail to notice some important details.

Or have students accomplish the challenge as the Challenge Card instructs and share their outcomes with the whole class.

Challenge 2- The Explosive Number

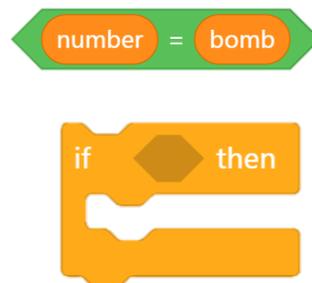
Game rule: When the button is pressed, Codey will generate a number randomly. Two players play the game rock-paper-scissors. Anyone who loses the game presses the button and the number on the LED screen will decrease by 1. Repeat the steps. When the LED screen shows a number that equals the randomly generated number, Boom! The bomb explodes.

- You need to create two variables: bomb and number. The variable "bomb" stands for the randomly generated number. The variable "number" represents the number that keeps changing during the game (starting from 0). The bomb will explode once the two variables output the same value.
- When Codey Rocky starts up, the initial values of the two variables are 0.
- When the button A is pressed, the value of "bomb" is set to be a random number ranging from 1 to 20.



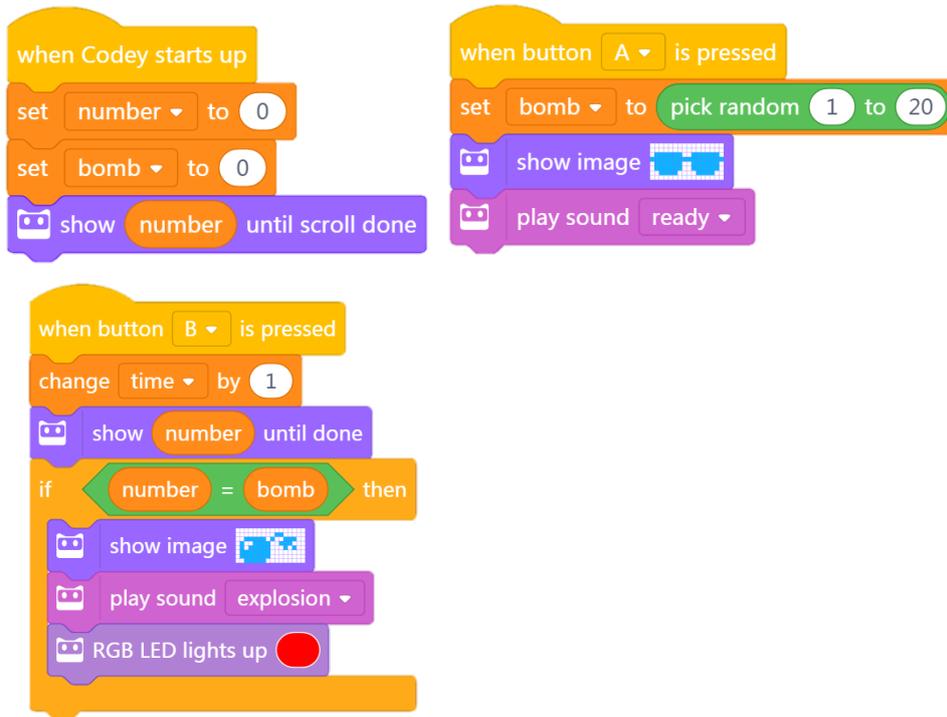
Meanwhile, Codey puts on his sunglasses and plays the sound "ready".

- Two persons play the game rock-paper-scissors. Anyone who loses the game presses the button B. When the button is pressed, the value on the LED display increases by 1.
- If the value of the variable "number" equals the "bomb" value, Codey will display an image of bomb and plays the sound accordingly. To assess whether the two values equal each other, you might need to use the operator block comparison and the if statement block in your code.



- Challenge:** Change the "random" range of the variable "bomb".

Sample program:



3. Presentation

You can show some well-done projects or have volunteers present their works. When sharing the projects, students are supposed to give their answers to the questions proposed by you.

After one student presents his or her project, you can invite some students to comment on the project (what is good about the project and where to improve). Then give your own comments.

4. Wrap up

You need to give a summary of today's lesson.

In today's lesson, we learned how to make a comparison between two variables. The result can be placed in the **if statement** block as the condition. And with a combination of a loop and Variables blocks, we can keep the value of a variable changing.

5. Students' Self-review

Please find the attached self-review worksheets. Hand out the copies and ask students to spend a few minutes filling the worksheets.

Lesson 16 Rock-Paper-Scissors

Lesson Plan

Overview:

Make Codey Rocky perform tasks as programmed by comparing the values of variables.

Teaching Objectives:

1. Understand the concept of Variables;
2. Make comparisons between variables.

Preparation:

1. A whiteboard and a whiteboard marker(or you can use a blackboard and chalks) ;
2. One Codey and a Bluetooth dongle (or the USB cable) per student but it's fine if 2 or 3 students share one set;
3. A computer with installed mBlock 5 per student but it's fine if 2 or 3 students share a computer.

Prior Knowledge:

1. Knew how to upload programs to Codey;
2. Understood how to use Loops and mastered the concept of Functions;
3. Knew how to create and use variables.

Teaching Procedure:

1. Review

Ask students:

- 1) By using the if statement block and the loop block, what abilities can you add to a variable?

Sample answer: We can increase or decrease the value of a variable.

2. Challenge: Rock-Paper-Scissors

About the task:

In daily life, we often play a game Rock-Paper-Scissors with our friends. As the name suggests, "rock" "paper" and "scissors" represent three different hand gestures. And the game rule is simple: the "rock" can be used to hammer the "scissors" so the "rock" beats the "scissors"; the "scissors" can cut the "paper" so the "scissors" beats the "paper"; the "paper" is used to wrap the rock so the "paper" beats the "rock". Explain the rules to students and have students code Codey Rocky to play the game with them.

Give a brief description of the challenge in case students might fail to notice some important details.

Give a brief description of the challenge in case students might fail to notice some important details.

Or have students accomplish the challenge as the Challenge Card instructs and share their outcomes with the whole class.

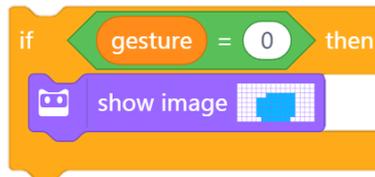
Challenge- Rock-Paper-Scissors

Using Codey Rocky, you can play the game rock-paper-scissors with friends.

- When Codey Rocky starts up, the initial values are all set to be 0.
- Codey Rocky changes its hand gestures randomly. The numbers 0, 1 and 2 are used to represent Rock, Scissors and Paper respectively. When Codey is shaken, the variable "gesture" will be one of the three numbers randomly.



If the value is 0, the LED screen will show an image of a fist.



If the value is 1, the LED screen will show an image of scissors.

If the value is 2, the LED screen will show an image of paper.

- If Codey wins the game, press the button A and the variable "win" change its value by 1. Meanwhile, Codey shows a smiley face and plays the sound "laugh".
- If Codey loses the game, press the button B and the variable "lose" changes its value by 1. Meanwhile, Codey shows a sad face and plays the sound "sad".
- If it is a draw, press the button C and the variable "draw" changes its value by 1. Meanwhile, Codey looks calm and hums.
- In some cases, Codey might steal a glance at its chance of winning. When the light intensity exceeds 2, the chance of winning will appear on the LED screen in the form of decimals.

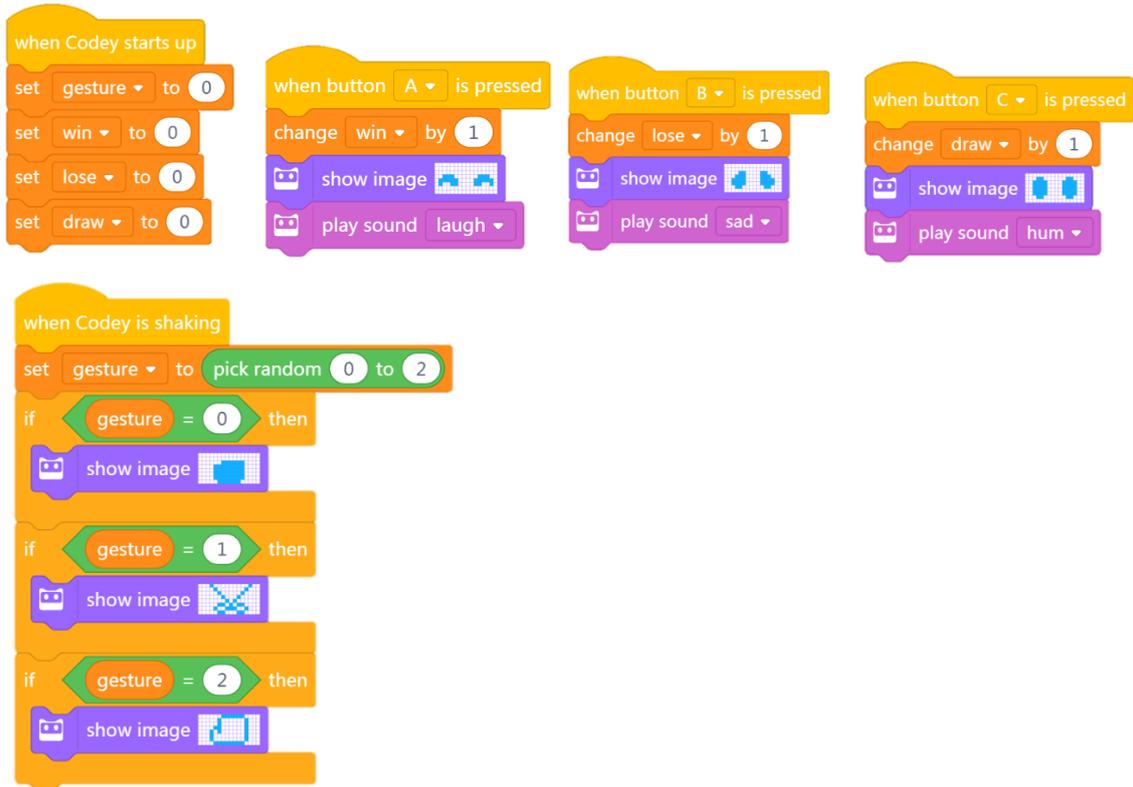


Tips:

- 1) When coding Codey Rocky, students can use six Event blocks at most;
- 2) To get the chance of winning, students need to add multiple Operators blocks to the code.



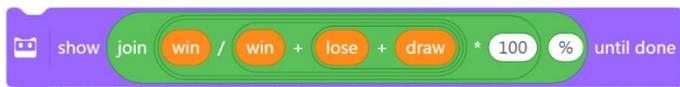
Sample program:



Discussion:

1) How to show the winning probability on the LED display in the form of percentages.

Sample answer:



2) How to show the winning probability in the form of ratio? Sample program:



Extension Task: IR Battle

The extension task is optional (based on the situations of students).

Challenge 2- IR Battle

Game rule: When the button is pressed, Codey transmits an infrared signal; if another Codey receives the signal, the health points go down by 1; the game ends when the health points fall to 0.

- The ears of the Codey are preset with an IR receiver and an IR transmitter, which facilitates the wireless communication between two Codeys.



- Select the block "send IR message" in the category "Infrared". Have two Codeys send each other different messages. In this way, they can identify each other. For instance, two Codeys send messages "red" and "blue" respectively. The messages serve as the bullets.



- Once Codey A receives the message from Codey B, Codey A gets shot and its health points reduce by 1. You can use the if statement block to assess whether Codey receives an IR message (gets shot). Once Codey gets shot, it will make a surprised sound and the health points will fall by 1.

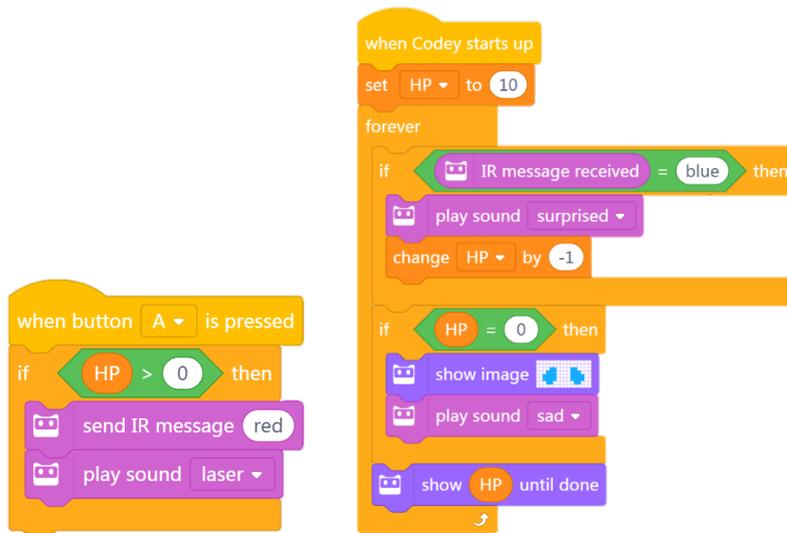


- When Codey starts up, the initial value of the health point is 10.
- The next step is to use the forever block to keep detecting whether Codey gets "bullets" from its "enemy". If Codey gets shot, the health point increases by "-1" and plays the sound "surprised". The game is over when the health points fall to 0 and Codey looks so sad.
- When the button A is pressed, if Codey still has health points left (the value is greater than 0), it can send IR messages and play the sound "laser".
- Challenge:** If it is a battle of three, then how to change the programs?

Tips:

1. When coding Codey Rocky, students can use six Event blocks at most.

Sample program:



Discussion:

- 1) If there are 3 teams, then how will you rewrite the program?
- 2) How to use the **repeat until** block to create programs?

3. Presentation

You can show some well-done projects or have volunteers present their works. When sharing the projects, students are supposed to give their answers to the questions proposed by you.

After one student presents his or her project, you can invite some students to comment on the project (what is good about the project and where to improve). Then give your own comments.

4. Wrap up

You need to give a summary of today's lesson.

We've learnt to assign a value to a variable in different ways. We could make an assignment by typing in a value in a variable block. We could also call a value of another variable, do mathematical operations, and then assign the calculated result to the first variable.

5. Students' Self-review

Please find the attached self-review worksheets. Hand out the copies and ask students to spend a few minutes filling the worksheets.

Lesson 17 Find the blue dot

Lesson Plan

Overview:

Gain a better understanding of how the physical buttons and the LED display work.

Teaching objectives:

Students will

1. Learn about physical buttons and their applications in our everyday life;
2. Understand how LED display works and complete game tasks;
3. Apply what they learn about the LED display to create simple programs.

Preparation:

1. Whiteboard and markers;
2. One Codey and a USB cable per student (OR two students share one set);
3. A computer with mBlock installed per student (OR two students share one set).

Difficulties & Key points:

1. Learn about the LED display and buttons
2. LED coordinate grid system
3. Compare variables

Teaching Procedures:

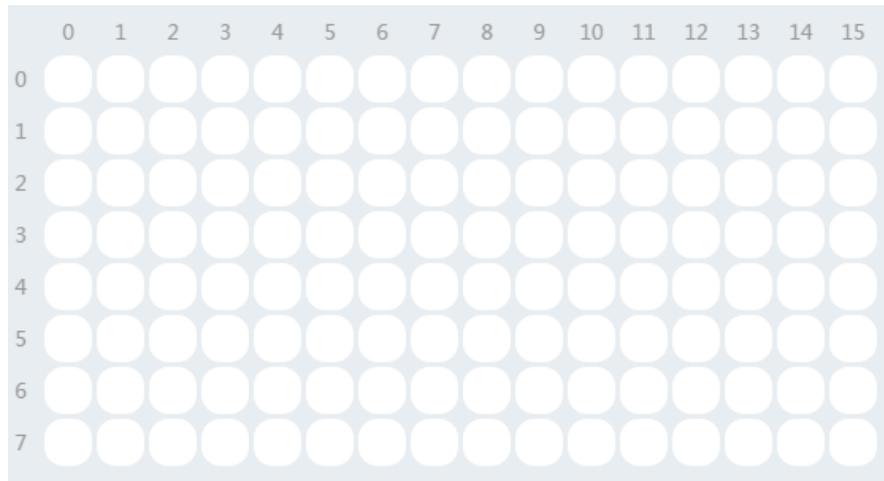
1. Lead-in Game

Have students play a game: Plotting Points. The task is to find the missing blue dot.

Introduce the game:

The rectangular coordinate system is closely related to our life. For example, when a teacher wants to pick one student to answer questions, there are many possible ways to do that. One way is to call the student's name. Another way is to point out the position where the student sits. If a teacher says "Let's have the student who's sitting at Row 3 Column 4 answer this question", students could easily spot who their teacher is referring to. The latter way is an application of the rectangular coordinate system in our daily life. Here are two number lines in the picture below. The horizontal line at the top shows the column number while the vertical

one on the left shows the row number. Both lines start from 0, and value increases from top to bottom and left to right. Have students take a close look at the following grid.



Introduce the task: Fill the grid based on the following coordinates.

3, 2	3, 3	3, 4	3, 11	3, 12	3, 13	4, 1	4, 2
4, 3	4, 4	4, 5	4, 10	4, 11	4, 12	4, 13	4, 14
5, 1	5, 4	5, 5	5, 10	5, 11	5, 14		

Demonstration:

1) The first coordinate is (3, 2), indicating the dot is at Row 3 Column 2. Using the coordinate, we can locate the dot.

2) Fill the grid with pencils.

Tips:

1) You can print out the grid and ask students to fill the grid based on the coordinates.

2) You can replace the image with one you design and redefine the coordinates.

Ask students: What image did you get in the end?

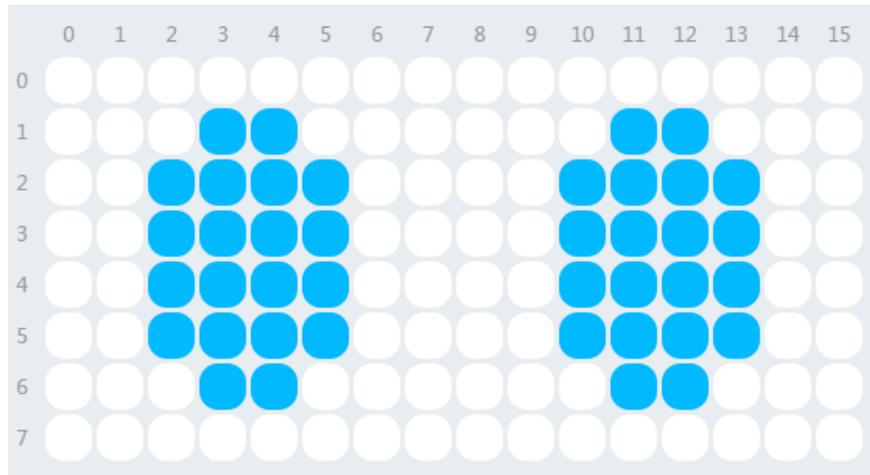
Possible answer: A smiley face

Summarize: In the game, we learned to locate the dots based on the coordinates. And we learned that coordinate systems are closely related to mathematics and are useful in our

everyday life.

2. Explain New Concept

The face of Codey comprises an 8X16 LED display. The display has 128 LEDs and each LED has its specific coordinate. We can easily locate the LED light based on the coordinates.



If we want to program the LED display, we need to use these coding blocks linked to the LED display. We can find them under the Looks category.

A list of Scratch coding blocks from the Looks category. The blocks are: 'show image [LED icon] for 1 secs', 'show image [LED icon]', 'show image [LED icon] at the x: 0 y: 0', 'turn off screen', 'show hello', 'show hello until scroll done', 'show hello at x: 0 y: 0', 'light up x: 0 y: 0', 'light off x: 0 y: 0', 'switch between light-up and light-off x: 0 y: 0', and 'x: 0 y: 0 is it lighted up?'. A red box highlights the 'light up' and 'light off' blocks. A blue arrow points from this box to a text box on the right.

Type in numbers to turn on/off a particular LED light.

Tips:

- 1) On the right-hand side of the drawing editor, we can see the icon  .
Click the icon to see the coordinates.

Applications:

In our daily life, LED displays are around us. For instance, they are used in those rolling advertising billboards which we often see in the streets.

Codey has three buttons: Button A, Button B and Button C. We can program each button to add different abilities to them. In the following task, we will use the three buttons.

Explain to students: "When you are programming the three buttons, you need to use these coding blocks linked to buttons. You can find them under the category **Events**:

Pull the drop-down menu to select the button you need.

And we can find the following block under the category Sensing.



The two blocks above can help us program the buttons. The difference is that the yellow block is an Event and the blue block is a Boolean (a Boolean needs to work with an if/then block).

3. Coding Task

The task is called **Find the Blue Dot**. The task requires good memory so it might be somewhat challenging.

1) Goal: When button C is pressed, a blue dot appears on the LED screen and then disappears, and a new blue dot appears in the left corner of the screen. Write programs to move the second blue dot to the position where the first blue dot is placed.

2) Gameplay tips: Pressing button A moves the blue dot to the right; pressing button B makes the blue dot move upwards. When you make a right move, the LED screen displays a face wearing sunglasses; when you make a wrong move, the screen displays a crying face and the game is over.

4. Explain Game Rules

- Upload the program to Codey Rocky (Remind students that they should not see the programs at this stage/OR the teacher can help upload programs for students).

- The blue dot can move forward but cannot move backward. One wrong step will end up in losing the game.

- Press button C to start the game.

- Press button A and button B to move the blue dot. Press button C to restart the game after you win.

- Calculate how many times you win the game.

5. Independent Practice

- Upload programs

- Start the game

- Ask students to raise hands when the task is completed.

6. Working Principle Analysis

After students complete the task, invite 6 students to challenge each other to see who completes more levels. Have one student be the referee. The top 3 students will gain game coins.

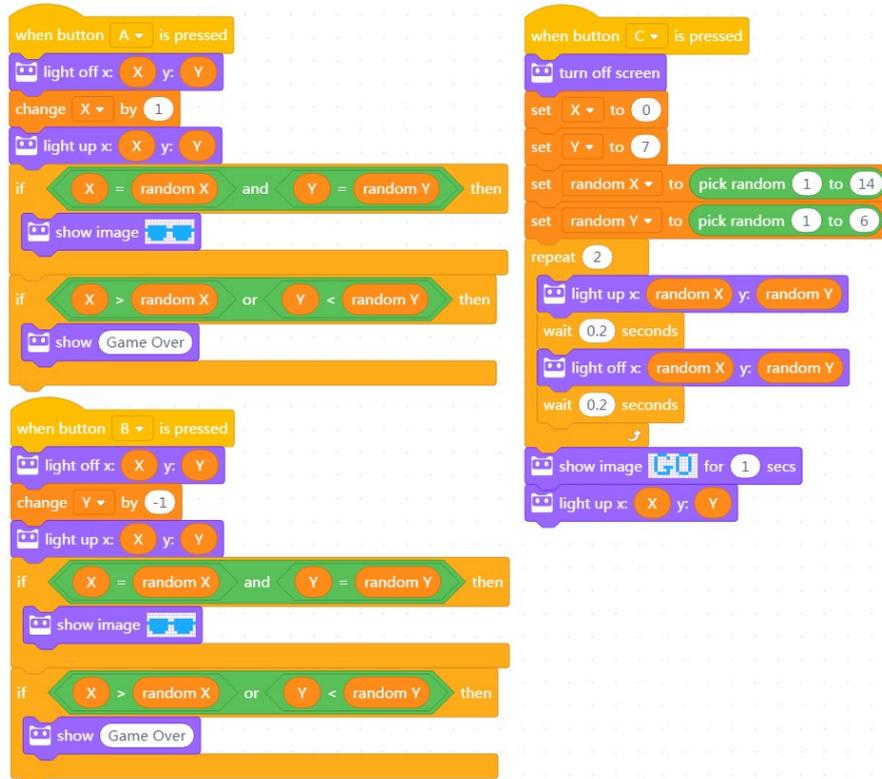
Ask students: "How does the program decide whether the blue dot already moves to the correct position?"

Possible answer: "The program memorizes the original position where the blue dot is placed. Then the program decides whether the blue dot gets to the correct position based on its memory."

Explain to students: "That's right. Since each dot has a coordinate, the program can easily locate the dots. That's why the program can make the correct decision."

Give further explanations: Create variables "X" and "Y" to store the position of the blue dot in the left corner. Create variables "random X" and "random Y" to store the position of the randomly generated blue dot. Press button A and button B to light up the moving dot by changing the value of X and Y respectively. Meanwhile, the program decides whether the current coordinate matches the value of the randomly generated blue dot. It matches, a face wearing sunglasses will be shown on the LED screen; otherwise, Game Over will be shown

on the screen.



From the programs above, we can know that the programs for button A and button B share some similarities. The only difference is that pressing button A changes the value of the X coordinate but pressing button B changes that of the Y coordinate.

7. Wrap-up

Each dot on the LED screen has a coordinate. A coordinate is made up of numbers, so we can create variables to store the coordinates. Then by pressing buttons to change the values, we can light up the LEDs however we like.

Lesson 18 Lucky Wheel

Lesson Plan

Overview:

Complete tasks using physical buttons and LED displays.

Teaching Objectives:

Students will

1. Learn about LED displays and their applications;
2. Understand how LED displays work and complete tasks using physical buttons and LED displays.

Preparation:

1. A whiteboard and markers (OR a blackboard and chalks);
2. One Codey Rocky and a USB cable per student (OR two students share one set);
3. A computer with mBlock installed per student (OR two students share one set).

Prior Knowledge:

This lesson requires students to

1. Use physical buttons;
2. Understand what LED displays are used for and how to use them.

Teaching Procedures:

1. Review

Briefly review what students learned in the previous lesson: functions and applications of physical buttons and LED displays.

2. Task one: Lucky Wheel

Briefly describe the rules to students. This can help those who have no patience for written texts better understand the rules. Of course, students can follow the instructions on the Challenge Card to complete the task step by step. After they finish, ask students to present their works to the class.

Use Codey Rocky to control the spinning of the wheel on the stage.

Codey Rocky	Sprite-Wheel	Sprite-Arrow
1. Press the button A to make the wheel spin and button B to stop the wheel. At the same time, either a start or a stop icon will be shown on the LED display.	1. When button A is pressed, the wheel starts spinning; when button B is pressed, it stops spinning.	1. Place the arrow near the wheel.

Connect Codey Rocky to mBlock 5

1. Connect via Bluetooth or a USB cable.

Add/Delete Sprites

1. Add the sprite Wheel.
 - Click "+" under Sprites-Click "My Sprites"-Choose the image we need;
 - Name the sprite as "Wheel";
 - Change its name in Costumes settings as well.



2. Add the sprite Arrow1.
 - Name the sprite as "Arrow";
 - Reset the size of the sprite to fit the stage.



3. Delete the sprite Panda.

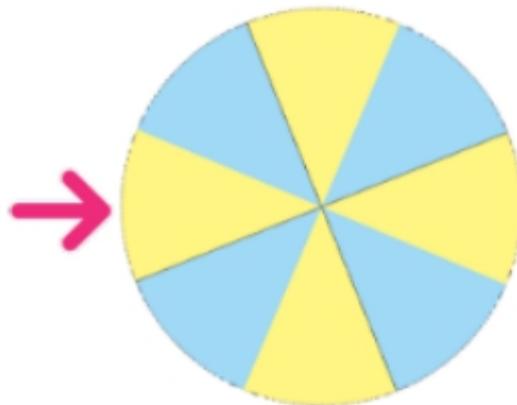
Write Programs

1. Device: When Codey starts up, a stop icon will be shown on the LED display.
2. Device: When button A is pressed, a start icon will be shown on the LED display and a message "Start" will be sent.

3. Sprite-Wheel: When receiving the message "Start", it keeps turning right 20°.
4. Device: When button B is pressed, a stop icon will be shown on the LED display and a message "Stop" will be sent.
5. Sprite-Wheel: When it receives the message "Stop", it will stop all the scripts.

Tips: You can use a triangle as the start icon and a square as the stop icon.

Referential Stage Design



3. Task Two: Catch the Apples

Catch the Apples

Use Codey to control the monkey to catch the falling apples.

Codey Rocky	Sprite-Monkey	Sprite-Apple
<ol style="list-style-type: none"> 1. When a button is pressed, Codey will send messages to control the monkey. 2. It shows directions on its LED display. 	<ol style="list-style-type: none"> 1. When the message is received, the monkey will move. 2. Set a conditional to tell whether the monkey touches the apple. 	<ol style="list-style-type: none"> 1. The apple falls randomly from the top until it reaches the bottom.

Connect Codey Rocky to mBlock 5

1. Connect via Bluetooth or a USB cable.

Add/Delete Sprites

1. Add the sprite Monkey

- Set its size to 50%.



2. Add the sprite Apple

- Reset the size of the sprite to fit the stage.



3. Delete the sprite Panda

□ Write Programs

1. Device: When button C is pressed, Codey Rocky broadcasts the message "Start".

2. Sprite-Apple: When it receives the message "Start", the apple repeats the following instructions: it moves to a random position and its y coordinate is set as 180; it repeats moving downward (its y coordinate increases by -10) until its y coordinate is less than -170 or it hits the monkey.



3. Sprite-Monkey: When receiving the message "Start", the monkey will move to the initial position as set.



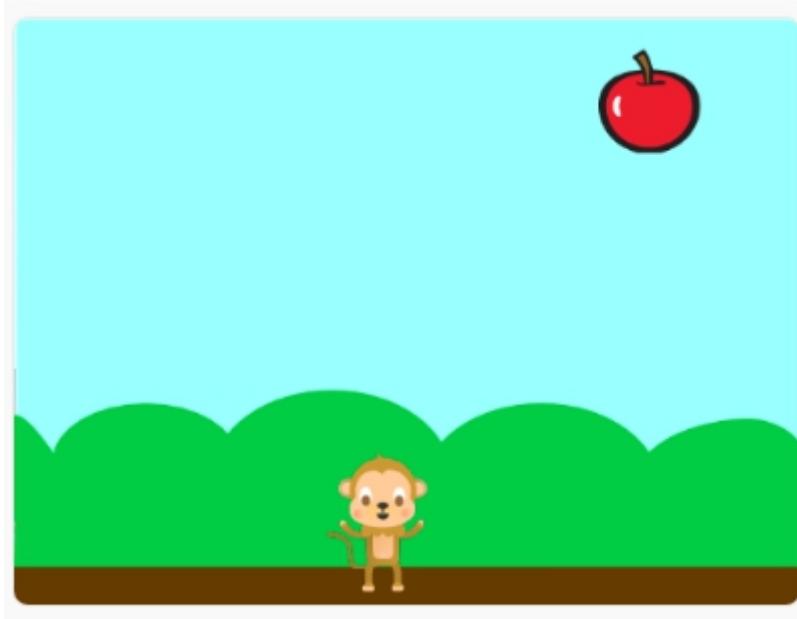
4. Device: When Codey starts up, it repeats the following instructions: When button A is pressed, it broadcasts "Move leftward", and a left arrow is shown on the LED display; when button B is pressed, it broadcasts "Move rightward", and a right arrow is shown on the LED display.

5. Sprite-Monkey: When the message "Move leftward" is received, it points in the direction of -90°

and then moves 10 steps. When the message "Move rightward" is received, it points in the direction of 90° and then moves 10 steps.

point in direction -90

□ Referential Stage Design



4. Independent Practice

- ① Have students work in pairs to complete the tasks.
- ② Ask students to raise their hands when they finish.

5. Presentation

Give students time to present their projects.

6. Wrap-up

In this lesson, we completed a few tasks using the physical buttons. We learned to use physical buttons to control sprites. Also, the LED display brought a better visual effect to our programs.

Lesson 19 Bomb Disposal Expert

Lesson Plan

Overview:

Learn about Neuron Funny Touch through games.

Teaching Objectives:

Students will

1. Learn about Neuron Funny Touch and its applications;
2. Learn to connect Neuron blocks to Codey Rocky;
3. Understand how Funny Touch works and complete tasks.

Preparation:

1. A whiteboard and markers (OR a blackboard and chalks);
2. One Codey Rocky, a Neuron Funny Touch block and a computer with mBlock

installed per student.

Difficulties & Key Points:

1. Learn about Neuron Funny Touch;
2. Connect Neuron blocks to Codey;
3. Compare values of a variable.

Teaching Procedures:

1. Situated Learning

In the previous lesson, we have learned about Codey Rocky's sensors. In this lesson, students will learn about Codey Rocky's friend-Neuron Funny Touch.

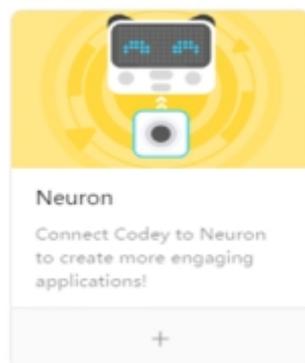


2. Explain New Concept

Connect Devices

Pogo pins are used in Neuron blocks. The left side of a neuron block can be stuck to Codey and Rocky.

Open mBlock 5. At the bottom of the Blocks area, there is a "+" button. To use the Neuron extension, you need to click "+" first, then select Neuron in the Extension center, and Click "Add".

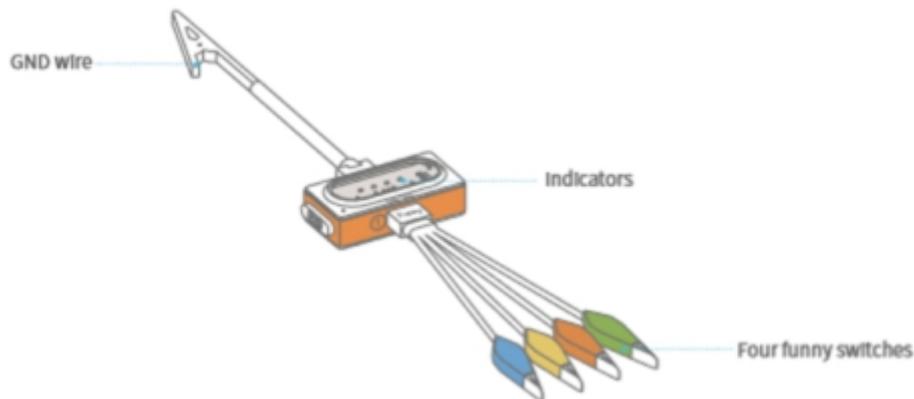


After you add the extension, you'll see the Neuron category at the bottom of the Blocks area. You can click the Neuron category to drag the blocks you'll need to the Scripts area.



Before moving on to the coding session, you should first introduce the Neuron Funny Touch block. The Funny Touch block could turn any conductive materials like a banana or

water into a touch switch. It contains a GND wire and four funny switches in different colors. In the middle lies the main part of the Funny Touch block, the GND wire is connected to slot 2 and the funny switch to slot 1. The light indicator will turn on once the circuit between the ground wire and the clips is closed up. If we hold the GND wire in one hand and one of the alligator clips with the other hand, we will also see an indicator being turned on. That is because our bodies are conductive.



Applications

Given that the Funny Touch block can turn any conductive materials into touch switches and that fruits are conductive, some people may use it to make a fruit piano while others make a game controller. People can invent fun gadgets with help of the Funny Touch and all sorts of conductive materials.

You need to explain to students: When the Funny Touch block is used, they need the following coding block.



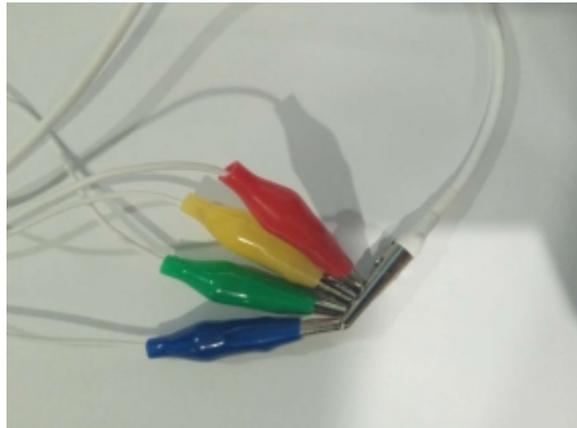
Click on the drop-down menu, we can choose the alligator clip needed. There are four alligator clips, respectively in red, yellow, blue and green. Conditionals can be used to tell which clip is touched.

3. Coding Task

The first task today is **Dispose of the Bomb**. Students may see bomb disposal scenes in films. The bomb disposal experts in the film would have to figure out the circuit of a time bomb and cut the right wire to prevent a massive blast. If they cut the wrong wire and trigger

the bomb, the experts and hundreds of people may lose their lives. In most cases, if the experts fail to stop the bomb within given time, the bomb will also explode. In this task, students will use Codey Rocky and Neuron to complete a bomb disposal task.

1) Goal: When button A is pressed, a countdown starts on the display. Students need to find out the right wire to cut before time is up. Connect the alligator clips of the Funny Touch block according to the example below.



2) How to Play: Press button A to start the game. Codey will give clues with its RGB LED, and students need to follow the clues to find out the right wire to cut.

4. Explain the Rules

- Upload the program to Codey Rocky (Remind students that they should not see the programs at this stage/OR you can help upload programs for students).

- Connect the four Funny Switches (Alligator Clips) to the GND Wire.

- Press button A to start the game.

- When the game begins, Codey will start counting down and give us clues with its RGB LED.

- If students manage to defuse the bomb, they complete the task; otherwise, they fail the task.

5. Independent Practice

- Upload the program to Codey Rocky.

- Have students start handling the "bomb".

- When time is up, ask students to share their progress and results.

6. Working Principle Analysis

Ask the students: After complete the task, do you know why the program can help us tell

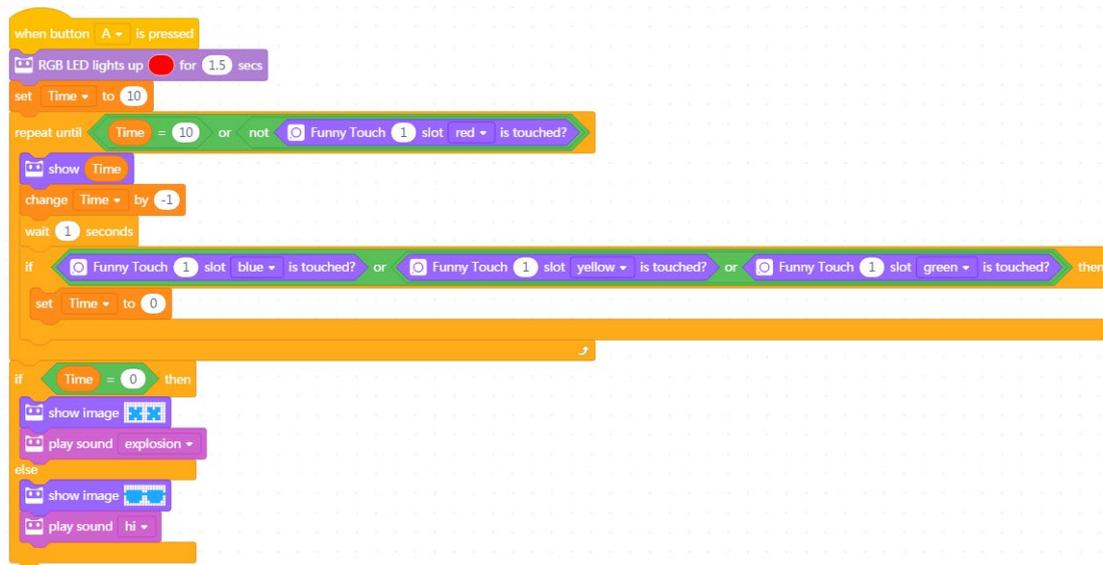
which wire is cut?

They may say: Because we use the "Not []" operator block.

Then you can explain to them: We use the **Funny Touch [1] slot [Color] is touched?** block to decide which switch has been touched. When we nest it into the Operator Block "Not []", we could realize the opposite result, that is, it tells us which switch has not been touched.



Also, you need to explain how to create a countdown: We used Variables blocks to make the countdown, **Loop** blocks to keep the values changing, and the Control block **wait [] seconds** to define the loop time. At the beginning of the game, clues were shown through Codey's light indicator. We used conditional blocks in the code to decide whether the Loop should be stopped: If we manage to disable the bomb before the countdown timer reaches 0, we manage to disable the bomb and win the game; otherwise, we lose the game.



7. Wrap-up

Neuron Funny Touch block could turn any conductive materials into input modules, which allows us to add more abilities to Codey Rocky. Only when the circuit between the GND Wire and the Funny Switch is closed can they work properly.

Lesson 20 Multifunction Switch

Lesson Plan

Overview:

Complete tasks using the Funny Touch block.

Teaching Objectives:

Students will

1. Learn about Neuron Funny Touch and its applications;
2. Understand how Funny Touch works and complete tasks.

Preparation:

1. A whiteboard and markers (OR a blackboard and chalks);
2. One Codey Rocky and a USB cable per student (OR two students share one set);
3. One computer with mBlock installed per student (OR two students share one set).
4. Sets of Neuron Funny Touch and sheets of Funny Touch Display Paper.

Prior Knowledge:

This lesson requires students to

1. Use physical buttons;
2. Understand how touch switches work and how to use them.

Teaching Procedures:

1.Review

Briefly review what students learned in the previous lesson: functions and applications of touch switches.

2.Task One: Codey Rocky Loves Music

Briefly describe the rules to students. This can help those who have no patience for written texts better understand the rules. Of course, students can follow the instructions on the Challenge Card to complete the task step by step. After they finish, ask students to present their works to the class.

Codey Rocky Loves Music

Using the Funny Touch block, you can make the instruments on the stage react to what

Codey does.		
Codey Rocky	Sprite—Panda	Instrument-Drum-snare /cymbal / highhat / kit Instrument-Drum
1. Use Funny Touch to enable Codey to broadcast messages to the sprites.	1. Place Panda in the middle of the stage.	1. The instruments on the stage will react to the messages they receive.
<input type="checkbox"/> Connect Codey Rocky to mBlock 5		
1. Connect via Bluetooth or a USB cable.		
<input type="checkbox"/> Add/Delete Sprites		
1. Add the sprite drum-snare		
<ul style="list-style-type: none"> ● Reset the size of the drum-snare to make it fit the stage; 		
		
<ul style="list-style-type: none"> ● Select the drum-snare, click Costumes, then delete the third costume. 		
2. Add the sprite drum-cymbal		
<ul style="list-style-type: none"> ● Reset the size of the drum-cymbal to make it fit the stage; 		
		
<ul style="list-style-type: none"> ● Select the drum-cymbal, click Costumes, then delete the third costume. 		
3. Add the sprite drum-highhat		
<ul style="list-style-type: none"> ● Reset the size of the drum-highhat to make it fit the stage; 		



- Select the drum-highhat, click Costumes, then delete the third costume.
4. Add the sprite drum
- Reset the size of the drum to make it fit the stage;



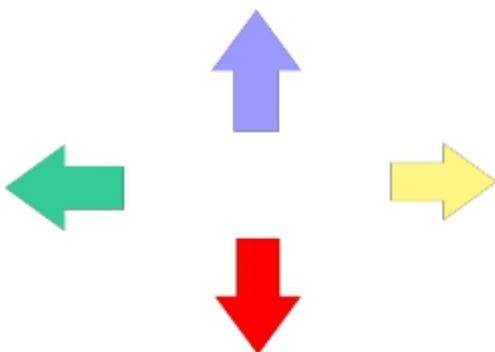
- Select the drum, click Costumes, then delete the third costume.
5. Add the sprite drum kit
- 1 Reset the size of the drum kit to make it fit the stage;



- Select the drum kit, click Costumes, then delete the third costume.

Connect Neuron to Codey Rocky

1. Connect Neuron to Codey;
2. Connect the four Funny Switches to the Funny Touch Display Paper.



Funny Touch Display Paper

□ Write Programs

1. Device: When Codey starts up, it repeats the following steps: if the blue Funny Switch is touched, Codey broadcasts "DRUM"; if the yellow Funny Switch is touched, Codey broadcasts "DRUM2". (Follow the steps to define other colors.)

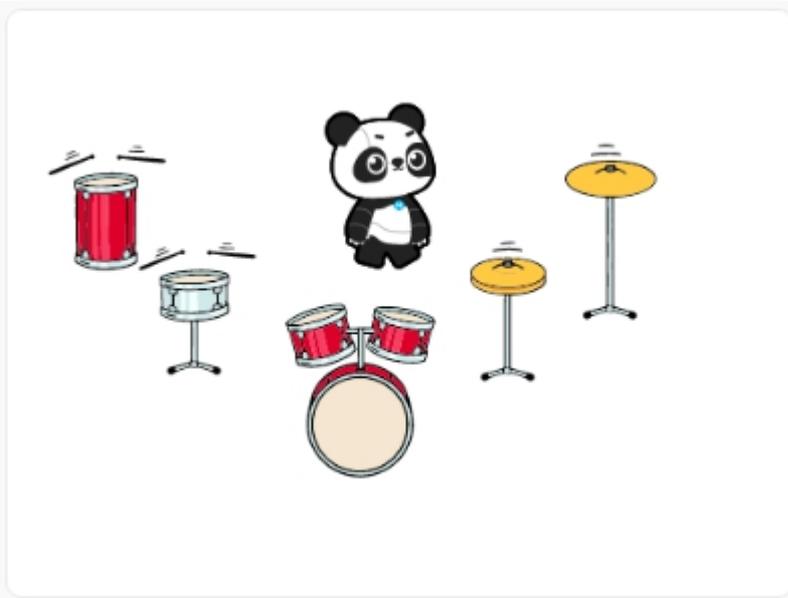


2. Sprite-Instrument-Drum: When receiving the message "DRUM", the drum changes to next costume, plays a sound and then changes to the original costume. With the same steps, you can complete the programs for the rest of the musical instruments.



Tips: When students run the program, remind them the GND wire of the Funny Touch block must have contact with their skins.

□ Referential Stage Design



□ Extension Task

1. Codey has three buttons. Give students time to figure out how to program the three buttons to serve as a Touch Switch and choose a proper background for the stage.

3. Task Two Beetle Hunter

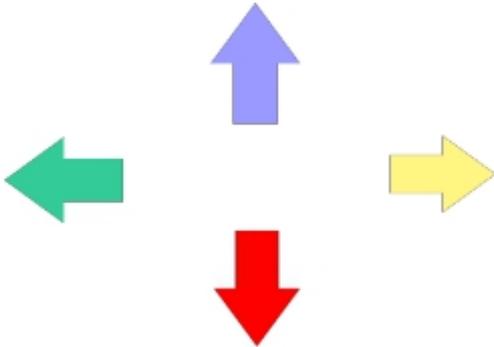
Beetle Hunter			
Use Codey and a keyboard to control two sprites. One needs to act as the chaser and the other sprite tries to avoid being caught.			
Codey Rocky	Sprite-beetle	Sprite-bird	Background- Jurassic
1. Use Funny Touch to broadcast a message and control the moves of one sprite.	1. The beetle moves based on the message that it receives and tries to avoid being caught by the bird.	1. Use the keyboard to control the bird.	N/A

Connect Codey Rocky to mBlock 5

1. Connect via Bluetooth or a USB cable.

Connect Neuron to Codey Rocky

1. Connect Neuron to Codey;
2. Connect the four Funny Switches to the Funny Touch Display Paper.



Funny Touch Display Paper

Add/Delete Sprites

1. Add the sprite beetle
2. Reset the size of the beetle to make it fit the stage.



3. Add the sprite bird
4. Reset the size of the bird to make it fit the stage.



5. Delete the sprite Panda

□ Write Programs

1. Device: When Button A is pressed, Codey Rocky broadcasts "Start".
2. Sprite-bird: When it receives the message "Start", the bird goes to a starting position.



It repeats the following steps: if it is on the edge, the bird bounces back (to stop it from moving off the edge of the stage); if the left arrow key is pressed, it points in the direction of 90° and moves -10 steps; if the right arrow key is pressed, it points in the direction of -90° and moves -10 steps; if the up arrow key is pressed, it points in the direction of 0° and moves 10 steps; if the down arrow key is pressed, it points in the direction of 180° and moves 10 steps.



3. Sprite-beetle: When receiving the message "Start", it goes to an initial position.



It repeats the following steps: if it is on the edge, the beetle bounces back (to stop it from moving off the edge of the stage); if it is hit by the bird, it broadcasts "End" and stops all the scripts.



4. When the message "End" is received, the LED display shows "Game Over".
5. Device: When Codey starts up, it repeats the following steps: if the blue Funny switch is touched, it broadcasts "Up". Program the rest of Funny Switch slots, making them broadcast the messages "Down", "Left", and "Right" when they are touched.

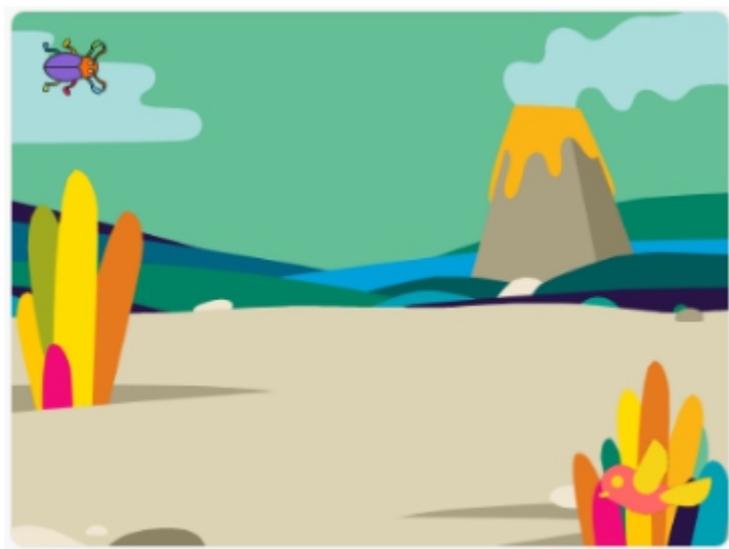
```
if [Funny Touch] 1 slot [blue] is touched? then
  broadcast Up
```

6. Sprite-beetle: When the message "Up" is received, the beetle points in the direction of 0° and moves 10 steps; when the message "Down" is received, it points in the direction of 180° and moves 10 steps; when the message "Left" is received, it points in the direction of -90° and moves 10 steps; when the message "Right" is received, it points in the direction of 90° and moves 10 steps.

```
when I receive Up
  point in direction 0
  move 10 steps
```

Tips: When students run the program, remind them the GND wire of the Funny Touch block must have contact with their skins.

Referential Stage Design



4. Independent Practice

- ① Have students work in pairs to complete the tasks;
- ② Ask students to raise their hands when they finish.

5. Presentation

Give students time to present their projects.

6. Wrap-up

In this lesson, we enabled Codey Rocky to interact with the stage sprites by programming and using the Neuron Funny Touch block.

Lesson 21 Codey Rocky Can do Addition

Lesson Plan

Overview:

Learn about IR sensors through games.

Teaching Objectives:

Student will

1. Learn about IR sensors and their applications in everyday life;
2. Understand what IR sensors are used for and complete tasks;
3. Understand how IR sensors work.

Preparation:

1. A Whiteboard and markers (OR a blackboard and chalks);
2. One Codey Rocky, a Bluetooth dongle (or a USB cable) and a Neuron Board per student (OR two students share one set);
3. One computer with mBlock installed per student (OR two students share one set).

Difficulties & Key Points:

1. Learn about IR sensors;
2. Use IR sensors;
3. Compare the intensity value of an IR sensors with the value of a variable.

Teaching Procedures:

1. Situated Learning

Ask students: There are various kinds of household appliances like televisions and air-conditioners at our home. Do you know how to control them?

Sample answer: We use remote controls to control them.

Ask them another question: Then how do remote controls work?

Sample answer: They send IR signals to household appliances.

Explain and ask: When we use a remote control to send an instruction to a television, the remote control emits an infrared radiation which will be received by the television. But do you know why we can't see the infrared radiation?

Students may need time to think about the question.

Explain to students: Infrared radiation is beyond the visible spectrum so it is invisible to our eyes. Besides its application in controlling household appliances, a remote control can also be used in line-following or obstacles avoiding robots.

2. Explain New Concept

The IR sensor is located at the lower front part of Rocky. It consists of an IR transmitter and an IR receiver. The IR sensor detects objects by emitting and receiving radiation. When the IR transmitter emits radiation, it reaches the object and then the radiation bounces back to the receiver. The closer an object is to the sensor, the higher the intensity of reflection will be. The IR sensor of Codey Rocky has a detection range. Therefore, if the object is located out of this range, Codey Rocky may fail to detect it.

Introduce a new block: To program the IR sensor, we'll need this **blue color sensor reflected infrared light intensity block** in our code.



3. Game Task

After all the introduction, it is time for students to complete the task "Codey Rocky Can Do Addition". Codey Rocky will be the judge to decide whether students get the calculations right.

Show students how to complete the task:

- 1) When button C is pressed, two numbers will be shown on the LED display.
- 2) Fast Calculation: Add them up yourself and keep the result in mind. Then place a magnetic board in front of the IR sensor of Codey Rocky. The IR intensity value will be shown on the LED display. Keep moving the board horizontally until the calculated result is shown on the LED display.
- 3) Check Calculation: Codey Rocky will check your calculation; if you get the calculation right, a "√" symbol will be shown on the LED display; if you get it wrong, a "xx" symbol will be shown on the LED display.

Explain the Rules

- Upload the program to Codey Rocky. (Remind students that they should not see the

programs at this stage/OR you can help upload programs for students.)

- Make sure the IR sensor face forward.
- Press button C to start the game. Any miscalculation will lead to a "game over".
- The score will be shown on the screen when the game is over.

Independent Practice

- Start to upload the program.
- Give students time to complete the task.
- Have students present their projects.

4. Working Principle Analysis

When students finish, ask them whether they find out how the intensity value changes.

Ask students: Congratulations, you completed the task. Do you know why the intensity value of the IR sensor changed and how Codey Rocky decided whether you got the calculation right?

Sample answers: IR sensors have certain detection range, so the closer the magnetic board is to the sensors, the higher intensity value will be. Codey Rocky runs the program written to sum the two random numbers up and compare the calculated value with the intensity value of the IR sensor.

Explain to students: Yes, you're right. Codey Rocky judges by comparing the intensity value of the IR sensor with the calculated value.

As for the three variables we used in our program, two of them were used to generate two random numbers, and the last one to record the sum of the two numbers.

Sample program:

```

when Codey starts up
  show image [ ]

when button C is pressed
  set Number 1 to pick random 1 to 30
  set Number 2 to pick random 1 to 70
  show Number 1 until scroll done
  wait 1 seconds
  show Number 2 until scroll done
  wait 1 seconds
  set Sum to Number 1 + Number 2
  show GO until scroll done
  reset timer

repeat until timer > 6
  show color sensor reflected infrared light intensity until scroll done

if color sensor reflected infrared light intensity = Sum then
  show image [✓] for 1 secs
else
  show image [✗] for 1 secs
  show image [ ]

```

Generate two random numbers (Remind students to set a value range for them in case that the sum of the two numbers might exceed the detection range), assign them to two variables, and have the LED display show the numbers.

Use the “timer” block to set a time limit, and program the LED screen to show the infrared light intensity data in real time.

Compare the value of the infrared light intensity with the sum value we added up: if they match, a “√” symbol will be shown on the LED display, which means we get the calculation right; otherwise, a “×” symbol will be shown, which means we get the calculation wrong.

Ask students: What happens if we use a black object instead of a magnetic board? Will the IR sensor manage to detect the black object? Try that with the program we just wrote.

Explain to students: It is widely known that black colors are quite good at absorbing IR lights. Thus, light can never escape a black hole. Even though infrared light is invisible but it still cannot stop itself from being absorbed by black colored objects.

5. Coding Task

Prepare students for the next task: We now know that IR sensors can be used to make robots avoid obstacles, follow lines and detect objects. Now Let's use it to do another task.

In the previous session, we used the IR sensor to enable Codey Rocky to do addition. What about other mathematical operations? Try to modify your program to make Codey Rocky do subtraction.

Tips:

Given that the IR sensor has a detection range, you need to make sure the sum of the two random numbers are within the range. To achieve this, you can make the value range of Number 1 always greater than that of Number 2.

6. Wrap-up

IR sensors work based on the reflection of infrared light and detect obstacles based on the intensity of the reflection. If it receives reflection, it means that there is an object ahead; if it does not receive any reflection, it means there is no objects ahead. It is widely known that black colors are quite good at absorbing IR lights, so an IR sensor cannot detect black objects. What “scares” the light most is a black hole, because it can never escape a black hole.

Lesson 22 Jumping Codey Rocky

Lesson Plan

Overview:

Complete tasks using IR sensors and stage sprites.

Teaching Objectives:

Students will

1. Learn about IR sensors and their applications;
2. Understand what IR sensors are used for and complete tasks.

Preparation:

1. A whiteboard and markers (OR a blackboard and chalks);
2. One Codey Rocky, a Bluetooth dongle (or a USB cable) and a Neuron Board per student (OR two students share one set);
3. A computer with mBlock installed per student (OR two students share one set).

Prior Knowledge:

This lesson requires students to

1. Learn about IR sensors;
2. Use IR sensor related coding blocks.

Teaching Procedures:

1. Review

Briefly review what students learned in the previous lesson: functions and applications of IR sensors.

2. Task One: Jumping Codey Rocky

Briefly describe the rules to students. This can help those who have no patience for written texts better understand the rules. Of course, students can follow the instructions on the Challenge Card to complete the task step by step. After they finish, ask students to present their works to the class.

Jump, Codey!

Codey Rocky needs to avoid obstacles on its way. Use its IR sensor to make sprites

avoid obstacles.

Codey Rocky	Sprite-Codey	Sprite-Obstacle FIRE
<ol style="list-style-type: none">1. Use the IR sensor to enable the sprite Codey to jump.2. When button A is pressed, the game starts.3. When the game is over, a sad face will be shown on the LED display.	<ol style="list-style-type: none">1. When the game starts, the sprite Codey appears in the middle of the stage.2. It avoids obstacles by jumping over them. Any contact with an obstacle means a game over.	<ol style="list-style-type: none">1. When the game starts, the sprite FIRE keeps moving from right to left.

Connect Codey Rocky to mBlock 5

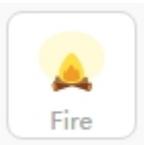
1. Connect via Bluetooth or a USB cable.

Add/Delete Stage Sprites

1. Add the sprite C-codey-rocky1-a
 - Rename it as Codey;
 - Reset the size of the sprite to make it fit the stage.



2. Add the sprite bonfire
 - Rename it as Fire;
 - Reset the size of the sprite to make it fit the stage.



3. Delete the sprite Panda

Write Programs

1. Device: When button A is pressed, a smiley face will be shown on the LED display and Codey Rocky broadcasts "Start".
2. Sprite-Codey: When Codey receives the message "Start", the sprite Codey will appear in slightly left off the middle of the stage. The following code will run repeatedly: if the sprite

Codey touches the sprite Fire, it will broadcast "Game Over", give a sad face and stop all the scripts.

```
go to x: -90 y: -40
```

3. Sprite-Fire: When the message "Start" is received, the fire will move to the rightmost point of the stage and show up. Then it starts moving from there to the leftmost point of the stage within the next 3.5 seconds and disappears there.
4. Device: When Codey Rocky starts up, it repeats the following steps: if it detects obstacles, it will broadcast "Jump" and wait for 0.8 seconds.
5. Sprite-Codey: When the message "Jump" is received, Codey, mimicking a jumping locomotion, will move upward within 0.4 seconds, and then move back to its original position within 0.8 seconds.

```
glide 0.4 secs to x: -90 y: 110
```

```
glide 0.8 secs to x: -90 y: -40
```

6. Device: When Codey receives the message "Game Over", a sad face will be shown on the LED display.

Referential Stage Design



Extension Task

Use variables blocks to keep track of the score. You gain 1 point each time you help Codey avoid an

obstacle.

3. Independent Practice

- ① Have students work in pairs to complete the tasks.
- ② Ask students to raise their hands when they finish.

4. Presentation

5. Wrap-up

In this lesson, we programmed the IR sensor to sense our movement. When the IR sensor sensed our movement, the stage sprite executed a corresponding program.

Lesson 23 RC Car

Lesson Plan

Overview:

Learn about IR transmitters and receivers and complete coding tasks.

Teaching Objectives:

Students will

1. Learn about Codey Rocky's IR transmitters and receivers and where they are located;
2. Learn about IR transmitters and receivers and their applications in everyday life;
3. Complete tasks using Codey Rocky's IR transmitters and receivers.

Preparation:

1. A whiteboard and markers (OR a blackboard and chalks);
2. One Codey Rocky and a USB cable per student (OR two students share one set);
3. One computer with mBlock installed per student (OR two students share one set).

Difficulties & Key Points:

1. Understand how IR signals are emitted and received;
2. Learn about the Infrared blocks and complete tasks using these blocks;
3. Understand why the two sets of Codey Rocky will be uploaded with different programs.

Teaching Procedures

1. Situated Learning

Ask students: "How do you change channel when watching TV? And how do you turn on an air-conditioner and adjust temperature on it?"

Sample answers: We use remote controls to do those.



Another question for students: Remote controls are widely used in everyday life. We use them to control household appliances like televisions and air-conditioners, and even RC cars, robots, and sound devices. But do you know why remote controls can be used to control those things?

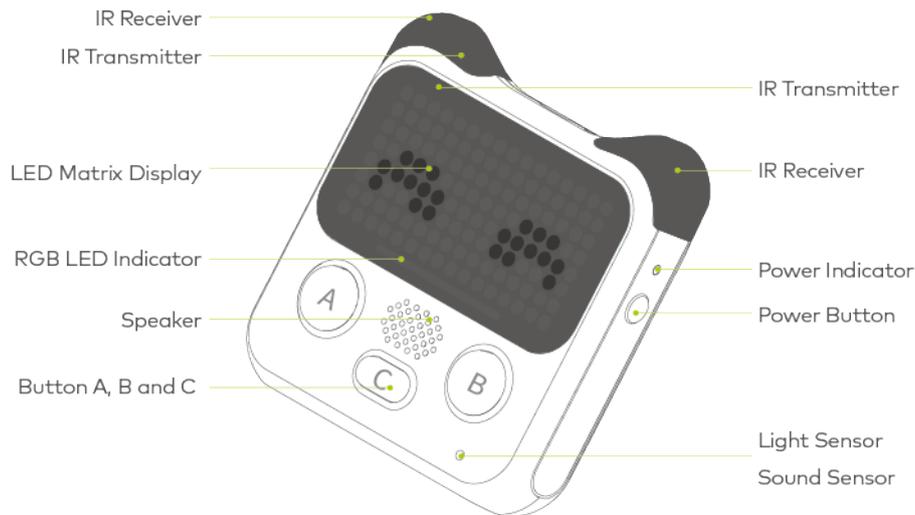
Give students some time to discuss and find out the answers. Sum up their answers and bring up today's topic: IR transmitters and receivers.

2. Explain New Concept

Introduce IR communication technology: IR communication is present in most remote controls. IR transmitters and receivers enable wireless data transmission. An IR transmitter sends out a modulated IR signal, which is picked up and demodulated by the IR receiver. That is how IR communication is carried out.

Ask students: Codey Rocky has its own IR transmitters and receivers. Do you know where they are?

Explain to students: Each Codey has two IR transmitters and receivers. The IR receivers are in Codey's ears. One of the transmitters is in its ear while the other in the LED display. Through programming, we could enable two Codey to "communicate" with each other.



Pros and cons of using IR transmitters and receivers:

- 1) Easy to use; no physical connection is needed;
- 2) Limited operating angle (a cone angle less than 30°); limited detection range and line-of-sight required the data transmission; high secrecy;
- 3) Not suitable in a place where there are many obstacles because a line of sight is required between the transmitter and receiver, and an obstacle between the transmitter and the receiver will interrupt the communication.

3. Time to Create

Write a story

Task Assigning: "Two sets of Codey Rocky can communicate with each other with the help of IR transmitters and receivers. Stretch your imagination and discuss with your partners, then create something fun with the IR devices.

Have students work in pairs to do brainstorming and write down their ideas. Walk around to keep in track of how students think. Invite 3-5 students to share their ideas. If the ideas are too complicated, provide advice on how to make the ideas realizable. Ask students to put down their ideas on paper and stick them to the Idea Wall.

Ask students: Remote controls can be used to operate televisions, do you know why?

Sample answers: Because there's an IR transmitter in the remote control, and an IR receiver in the TV set.

Give examples: We can use one Codey Rocky to control another by applying the IR

technology.

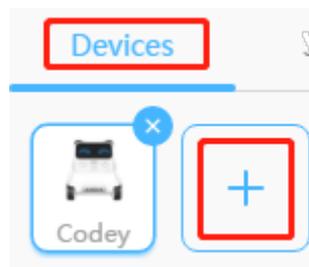
Show students the examples

Codey (Transmitter)	Codey 2 (Receiver)
Start up (IR Message: Start)	Move forward
When button A is pressed (IR Message: Left)	Turn left 90° and then move forward
When button B is pressed (IR Message: Right)	
When button C is pressed	
When loudness is greater than []	

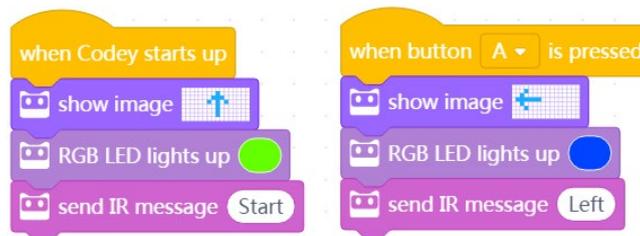
The sample program, **IR Transmitters and Receivers-Make Your Own RC Car**, can be found in the program files.

Explain how the program works: Shown in the table above are the IR messages between two sets of Codey Rocky. When Codey starts up, it sends the IR message "Start"; when Codey 2 receives the IR message "Start", it moves forward.

Click "+" under the Devices list, then add another Codey.



Program for the first Codey (i.e. Codey):)





Program for the second Codey (i.e. Codey2).

A screenshot of a Scratch-style programming environment showing a script for the Codey2 robot. The script starts with a yellow 'when Codey starts up' block. This is followed by an orange 'forever' loop block. Inside the loop, there are two 'if' blocks. The first 'if' block has a green arrow pointing to the right and is labeled 'IR message received = Start'. Its 'then' block contains: 'RGB LED lights up' (green dot), 'show image' (a blue and white image), and 'move forward' (at power 50%). The second 'if' block has a green arrow pointing to the left and is labeled 'IR message received = Left'. Its 'then' block contains: 'RGB LED lights up' (blue dot), 'show image' (a blue and white image), 'turn left' (90 degrees until done), and 'move forward' (at power 50%).

Tips for students

- 1) Use RGB LEDs as signs to indicate whether IR messages are successfully transmitted and received between the two devices;
- 2) Use numbers or English characters (An IR message should not be written in Chinese).

Independent Practice

- 1) Use "Event" blocks to allow Codey to send more IR messages to Codey2;
- 2) Add images, animation, lights or sound effect to Codey2.

Presentation

- 1) Have students upload their programs, and share with and learn from each other.
- 2) Invite volunteers to show their projects or recommend other groups' projects they find great. The volunteers should answer the following questions: a) What did you make? b) What inspired you? c) Did you come across any problems? d) If there's a yes to question c, how did you fix the problems? e) Did you receive any suggestions?

You should give feedback on their works.

4. Wrap-up

Give a wrap-up on today's lesson: IR communication technology and its applications.

You can expand students' horizon a little more by asking them: Can you think of other applications of IR communication technology?

If students are stuck, give them some examples. IR communication technology is used in cameras. A camera phone with IR communication features can wirelessly transfer photos to a printer or a flat panel display.

Preview with students what they are going to learn in the next lesson. In this lesson, they made two Codeys carry out one-way communication—using one to control the other—but it is possible to make a two-way talk between them. Have them imagine what they can achieve with the IR communication feature of Codey Rocky. What fun stories will happen to the two Codeys?

5. Students' Self-review

Have students complete the self-review report.

Lesson 24 When Codey Meets Codey

Lesson Plan

Overview:

Write stories or make animations using IR transmitters and receivers.

Teaching Objectives:

Students will

1. Know the location of Codey Rocky's IR transmitters and receivers and how to use the Infrared blocks;
2. Use Infrared blocks to facilitate communications between two Codeys;
3. Complete the story writing task by using IR transmitters and receivers.

Preparation:

1. One Codey Rocky and a Bluetooth dongle (or a USB cable) per student (OR two students share one set);
2. One computer with mBlock installed per student (OR two students share one set);
3. Textbooks for Students (or prepare task cards).

Prior Knowledge:

This lesson requires students to

1. Know Codey Rocky's IR transmitter and receivers are located and what they can be used for;
2. Use Infrared blocks to send and receive IR messages.
3. Create and use Variables blocks.

Teaching Procedures:

1. Review

Briefly review what students learned in the previous lesson: the location functions and applications of Codey's IR transmitters and receivers.

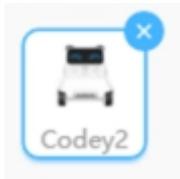
2. Task One: When Codey Meets Codey

In the previous lesson, you learned about IR transmitters and receivers. It is time to explore more and move on to today's first challenge.

Briefly describe the rules to students. This can help those who have no patience for written texts better understand the rules. Of course, students can follow the instructions on the Challenge Card to complete the task step by step. After they finish, ask students to present their works to the class.

In this session, students need to write a story of two Codeys based on the task description. Have them work and program in pairs. Ask them to act out their story when they finish.

The sample program, **IR Transmitters and Receivers-When Codey Meets Codey**, can be found in the program files.

When Codey Meets Codey	
One day, two Codeys met each other on the road, then a fun story started.	
Device-Codey	Device-Codey 2
<ol style="list-style-type: none"> 1. It sends an IR message "Hi". 2. When it receives the message "Hello", it makes a response and sends back a new message. 	<ol style="list-style-type: none"> 1. When it receives the IR message "Hi", it makes a response and sends an IR message "Hello".
<input type="checkbox"/> Connect Codey Rocky to mBlock 5	
<ol style="list-style-type: none"> 1. Connect via Bluetooth or a USB cable. 	
<input type="checkbox"/> Add Devices	
<ol style="list-style-type: none"> 1. Click "+" under Devices category to add another Codey (i.e. Codey2). 	
	
<input type="checkbox"/> Write Programs	
<ol style="list-style-type: none"> 1. Device-Codey: When button A is pressed, Codey shows a smiley face, laughs and sends an IR message "Hi". 	



2. Device-Codey2: When it starts up and receives the IR message "Hi", Codey2 waits for 1 second, makes a response (like a smiley face and/or a laughing sound), and then sends an IR message "Hello".



Remind students to use the "Forever" block to keep Codey2 detecting IR messages from Codey.



3. Device-Codey: When it starts up, Codey repeats detecting IR messages from Codey2. When it receives IR message "Hello":

- Codey waits for 1 second;
- Codey's RGB LEDs turn red;
- Codey plays sound "surprised";
- Codey sends back another IR message.

4. Complete the story.

- Why was Codey surprised?
- What message did Codey send back?
- How did Codey2 respond to the new message?

5. Upload the programs to two Codeys respectively.

□ **Tips**

Facial expressions, sound effects or light effects can be used as indicators telling us whether IR messages are successfully sent and received.

If students are stuck, help them contemplate one or two storylines, like "Codey asked in surprise, 'You got a new dress?'" or "Codey said in surprise, 'I found one dollar.'"

3. Task Two: Running Rabbit

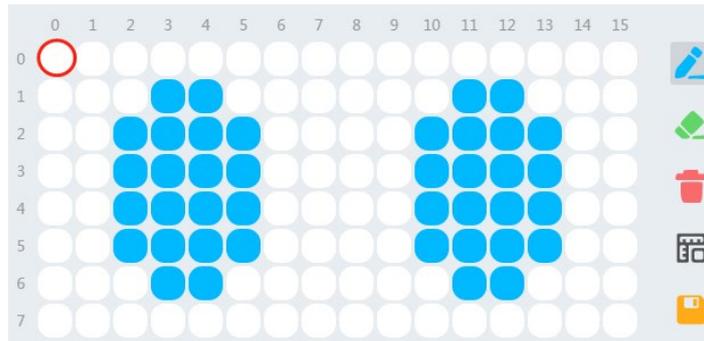
If time allows, move on to the second task, Running Rabbit.

First, introduce "image coordinates".

- 1) Show students how to use the block below:

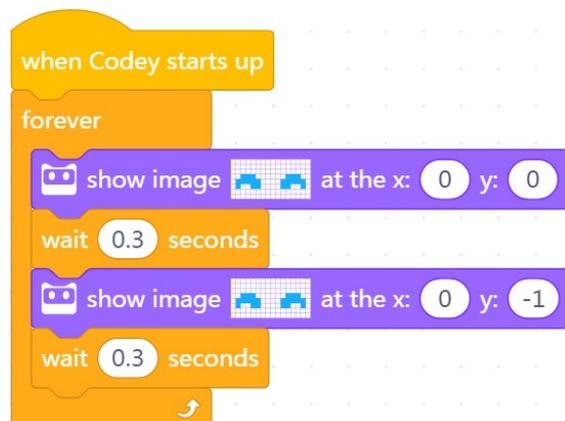


- 2) The block means that the image will be shown at a specific coordinate position. The point circled in red in the image below is the origin of coordinates.

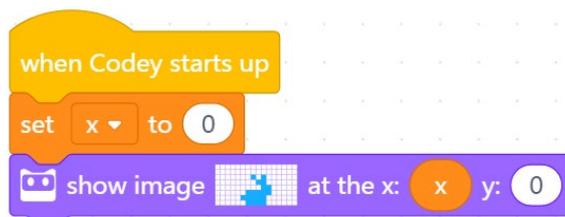


Tips: Click "  " in Image Editor to show coordinates.

Run the script below and see what happens.

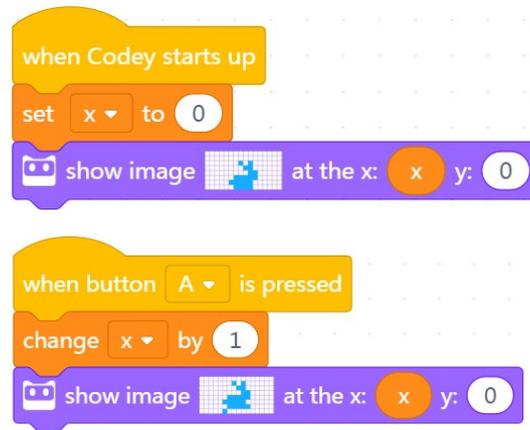


- 3) Use variables blocks to create an animation effect. In this task, we can create a variable "X" and set it as the x coordinate of the image.



- 4) Add another piece of code to achieve this effect: when a button is pressed, the

variable X will change its value. And when the variable reaches a specific value, the rabbit will run outside of the LED display.

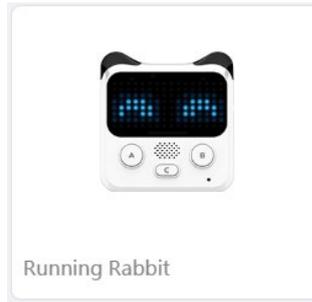
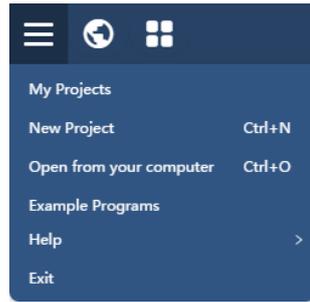


5) Mark down the value of the variable X when the rabbit reaches the rightmost place of the screen.

There is one more step before having students take the task—briefly describe the rules to students. This can help those who have no patience for written texts better understand the rules. Of course, students can follow the instructions on the Challenge Card to complete the task step by step. After they finish, ask students to present their works to the class.

The sample program, **IR Transmitters and Receivers-Running Rabbit**, can be found in the program files.

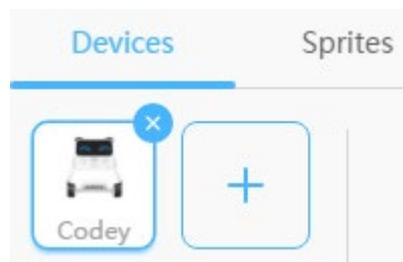
Running Rabbit	
<p>A little rabbit is running on Codey's LED display but the space is so small. Let's create more space for the rabbit, allowing it to run on the screens of two Codeys.</p>	
Device-Codey	Device-Codey2
<ol style="list-style-type: none"> 1. The rabbit runs from left to right. 2. When the rabbit reaches the rightmost position of the screen, Codey sends the IR message "A". 	<ol style="list-style-type: none"> 1. When receiving the IR message "A", the rabbit appears on the screen and runs left to right until it runs outside the screen.
<p><input type="checkbox"/> Connect Codey Rocky to mBlock 5</p> <ol style="list-style-type: none"> 1. Connect via Bluetooth or a USB cable. 	
<p><input type="checkbox"/> Open the Example Program</p> <ol style="list-style-type: none"> 1. Open the example program "Running Rabbit". 	



2. Read the program and upload it to Codey. Watch the animation of the running rabbit.

□ Add Devices

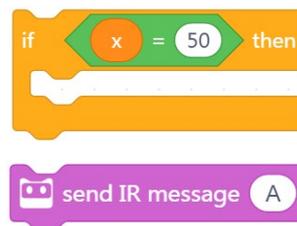
1. Click "+" under the Devices category to add another Codey (i.e. Codey2).



□ Write Programs

1. Device-Codey: Modify the program- Replace the **when Codey starts up** block with the **when button A is pressed** block. Have students think about why they need this replacement.

2. Device-Codey: When Codey starts up, it will keep track of the value of Variable X; if the value reaches 5 (that is to say, when the rabbit reaches the rightmost place of the screen), Codey will send an IR message "A".



3. Device-Codey2: When it starts up, Codey2 repeats detecting IR messages; when it receives the IR message "A", the rabbit appears on Codey2's screen and continues running.



Tips: Duplicate the script in the example program "Running Rabbit".

4. Upload the programs to two Codeys respectively. Then run the programs and see how they go.

Extension Task

Create an animation with the help of the IR communication features of two Codeys.

Ask students a question about task two: Why did you replace the **when Codey starts up block** with the **when button A is pressed** block. They may say, "With the replacement, we can replay the animation any time by pressing button A instead of restarting Codey."

4. Wrap-up

At this point, make sure to revisit what have been learned about IR transmitters and receivers, their functions and applications. To end, prepare students for the next lesson by giving a preview.

5. Students' Self-review

Have students complete the self-review report.

Lesson 25 Volume Control

Lesson Plan

Overview:

Learn about gear knobs and complete coding tasks.

Teaching Objectives:

Students will

1. Learn about gear knobs and their applications in everyday life;
2. Be able to describe the location of Codey Rocky's Gear Knob;
3. Complete tasks by using the gear knob.

Preparation:

1. A whiteboard and markers (OR a blackboard and chalks);
2. One Codey Rocky and a Bluetooth dongle (or a USB cable) per student (OR two students share one set);
3. One computer with mBlock installed per student. (OR two students share one set).

Difficulties & Key Points:

1. Create stories, make animations or design games using Codey's gear knob.

Teaching Procedures:

1. Situated Learning

To start, ask students whether any of them used earphones before and if they did, how they controlled the audio volume.

Then bring up today's topic. Some devices have a driving-plate-like component which is used to control volume—that is a gear knob.



A gear knob is an electronic component that we can toggle to change resistance. It can be

used to control the audio volume or the brightness of a light bulb.

2. Explain New Concept

Ask students the following question: "Codey has a gear knob. Do you know where it is?"

Show them after they give answers. Codey's gear knob is on its side.

Then throw another question: Do you know how to read the gear knob?

Students may say, "By dragging coding blocks and uploading programs." After hearing their answers, you can illustrate how to read the gear knob.

Method 1: Write the following code and upload it to Codey.



Method 2: Connect your device to Codey, tick the box before **the gear potentiometer value** block under Sensing category.



Make sure every student can see clearly the results you show: 1) the gear potentiometer value displayed on Codey's screen; 2) the value synchronously displayed on mBlock 5 stage.

3. Observation and Summary

Give student time to observe how the gear potentiometer value changes and have them answer the following questions:

1) When the gear knob is toggled upwards or in a clockwise direction, the value of the gear knob (); when the gear knob is toggled downwards or in an anticlockwise direction, the value of the gear knob ().

A decreases

B increases

2) According to your test, the maximum value of the gear knob is (), the minimum value is ().

A -100

B 0

C 100

The value of the gear knob is in the range of 0 to 100. When the gear knob is toggled

upwards, its value increases; when the gear knob is toggled downwards, its value decreases.

4. Time to Create

Ask students what can be done with the gear knob? Give them time to stretch their imagination and to complete the sentences below.

When the value of the gear knob increases, _____.

When the value of the gear knob decreases, _____.

As the value of the gear knob _____, _____.

Remind student to work in pairs. Walk around to keep in track of how students think. Invite 3-5 students to share their ideas. If the ideas are too complicated, provide advice on how to make the ideas realizable. Ask students to put down their ideas on paper and stick them to the Idea Wall.

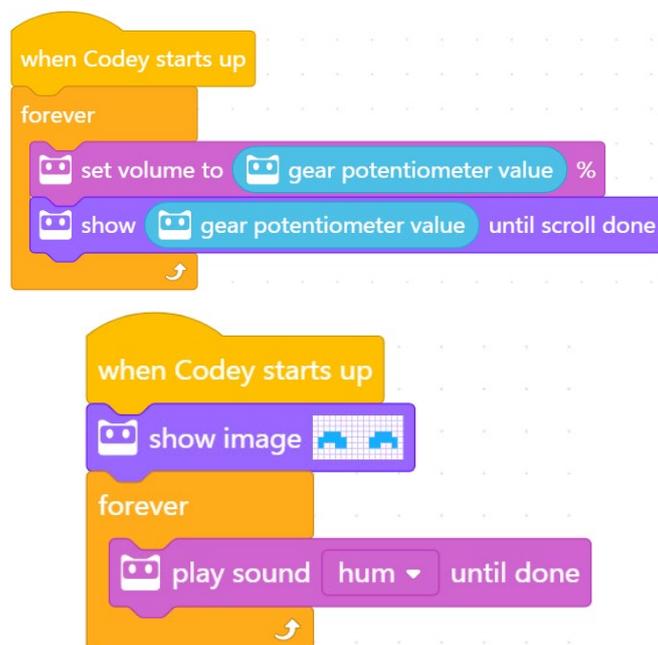
[Questions and Examples]

Questions and examples could help students along in their thinking. Run the example program and have students observe what happens when the gear potentiometer value changes.

The sample program, **Gear Knob-Volume Control**, can be found in the program files.

Sample purpose: Use the gear knob to adjust Codey's volume.

Sample program



Upload the programs to Codey and run them in class. Ask students what they notice

about Codey's audio volume. Some of them may realize when the gear knob is toggled upwards, Codey's volume goes up, and when the gear knob is toggled downwards, Codey's volume goes down.

Independent Practice

Ask students to write a program based on the sentences they completed in the earlier session. Remind them to use the gear knob. Help them simplify the overly complicated projects. If they are stuck, give them some more examples like controlling the size or location of a stage sprite with the gear knob.

Presentation

1) Have students upload their programs, and share with and learn from each other.

2) Invite volunteers to show their projects or recommend other groups' projects they find great. The volunteers should answer the following questions: a) What did you make? b) What inspired you? c) Did you come across any problems? d) If there's a yes to question c, how did you fix the problems? e) Did you receive any suggestions?

You should give feedback on their works.

5. Wrap-up

At this point, revisit what have been learned about gear knobs, their functions and applications. Again, asking questions is a good way to review. Have students think of the applications of gear knobs in everyday life. They may get at audio volume control or light bulb brightness control.

If time allows, you can further expand students' horizons by asking more questions. For example, "If the gear knob can be used to design games, what kind of a game will you design?" Any answers are welcome. To end, prepare students for the next lesson—Use the gear knob to design a game.

6. Students' Self-review

Have students complete the self-review report.

Lesson 26 Number Guessing

Lesson Plan

Overview:

Complete tasks using Codey's gear knob.

Teaching Objectives:

Students will

1. Describe the location of Codey Rocky's gear knob and use related blocks;
2. Complete tasks using the gear knob;
3. Complete extension tasks.

Preparation:

1. A whiteboard and markers (OR a blackboard and chalks);
2. One Codey Rocky and a Bluetooth dongle (or a USB cable) per student (OR two students share one set);
3. One computer with mBlock installed per student (OR two students share one set).

Prior Knowledge:

This lesson requires students to

1. Learn about the location and functions of Codey Rocky's gear knob;
2. Read the gear knob;
3. Use Variables blocks.

Teaching Procedures:

1. Review

Briefly review what students learned in the previous lesson: the location, functions and applications of Codey's gear knob.

2. Task One: Number Guessing

In the previous lesson, you highlighted Codey's gear knob and it is time for students to go further—use the gear knob to complete a task.

Briefly describe the rules to students. This can help those who have no patience for written texts better understand the rules. Of course, students can follow the instructions on the Challenge Card to complete the task step by step. After they finish, ask students to present

their works to the class.

The sample program, **Gear Knob-Number Guessing**, can be found in the program files

Number Guessing

Codey Rocky generates a random number between 0 and 100 and you need to guess the number.

Device: Codey

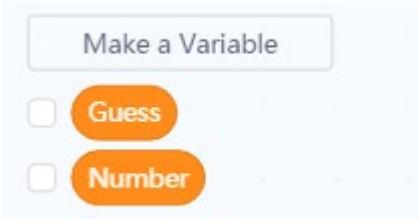
1. Generate a number between 0 and 100.
2. When the gear knob is turned, the guess is shown on the LED display.
3. When button A is pressed, Codey judges whether the guess is correct or not.

Connect Codey Rocky to mBlock 5

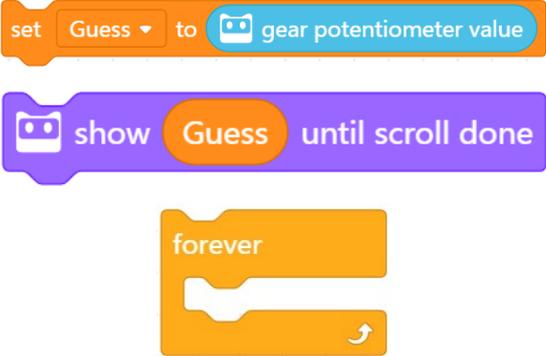
1. Connect via Bluetooth or a USB cable.

Write Programs

1. Make two variables, "Number" and "Guess".

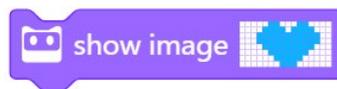

2. When Codey starts up, the two variables are initialized. The variable "Guess" with an initialized value of 0 refers to your guesses, and the variable "Number" refers to the random number picked by Codey from 0 to 100.


3. When you toggle the gear knob, your guess is shown on the LED display.


4. If button A is pressed, then Codey starts to check the condition: whether the guess is correct.



5. If the guess is correct, Codey repeats the following steps 3 times: it shows a heart on its screen and plays the sound "score".



Note: If the guess is right, the game restarts with the variable "Guess" returning to its initialized value and the "Number" variable could be any number between 0 and 100.

6. If the guess is lower than the number, a hint "<n" will be shown on the LED display for 1 second.
7. If the guess is higher than the number, a hint ">n" will be shown on the LED display for 1 second.
8. Upload the program to Codey. Watch how the program works and show it to students.

Extension Task

1. When a wrong guess is made, Codey plays the sound "embarrassed".
2. Improve the number guessing game. Usually, we would get a hint if we make a wrong guess. For example, if the generated number is 20, and your first guess is 30, you will get a hint that the generated number is in the range of 0 to 30. Then you move on and give your second guess 10, then you will get another hint that the generated number is in the range of 10 to 30. Follow the tips and keep guessing until you get it right.

Tips: Create two variables, "Maximum" and "Minimum". If your guess is lower than the generated number, the value of your guess will be assigned to the variable "Minimum"; if your guess is higher than the generated number, the value of your guess will be assigned to the variable "Maximum".

```

if (Guess < Number) then
  set Minimum to Guess
  play sound embarrassed
  show join join Minimum - Maximun until scroll done

```

3. Task Two: Change the Falling Speed

If time allows, move on to the second task, Change the Falling Speed.

Before having students jump into the task, brief them through the task. This task is actually an upgrade to the task Catch the Apples. Students are required to use Codey's gear knob to control the apples' falling speed.

1) In the task Catch the Apples, when button C is pressed, the game starts; when button A is pressed, the sprite monkey moves leftward; when button B is pressed, the sprite monkey moves rightward.

```

when Codey starts up
  forever
    if (button A is pressed?) then
      broadcast Move leftward
      show image [left arrow]
    if (button B is pressed?) then
      broadcast Move rightward
      show image [right arrow]
  when button C is pressed
    broadcast Start

```

2) Sprite-Monkey: When receiving the message "Start", it moves to the default position. When receiving the message "Move leftward", it points in the direction of -90° and then move 10 steps. When receiving the message "Move rightward", it points in the direction of 90° and then moves 10 steps.

```

when I receive Start
  go to x: -15 y: -135
  set size to 50 %

when I receive Move leftward
  point in direction -90
  move 10 steps

when I receive Move rightward
  point in direction 90
  move 10 steps

```

3) Sprite-Apple: When receiving the message "Start", the apple moves to a random position and falls from the top (any position where the Y coordinate is 180). It keeps falling down until it is caught by the monkey or falls on the ground (any position where the Y coordinate is less than -170). When it is caught by the monkey or falls on the ground, the apple returns to any position where the Y coordinate is 180.

```

when I receive Start
  forever loop
    go to random position
    set y to 180
    repeat until (y position < -170 or touching Monkey?)
      change y by -10
  
```

Task goal: Change the falling speed of the apple with the gear knob.

1) Create a variable "Speed" to control the falling speed of the apple.

2) The value range of the gear knob is 0-100. Assume that the falling speed of the apple is within the same range, the falling speed will get too fast if it goes over 10. Therefore, we need to narrow down the range. Divide the value of the gear knob by 10, round the quotient, assign the calculated value to the variable "Speed". Then the range of the variable "Speed" is

```

set Speed to round (gear potentiometer value / 10)
  
```

0-10,

3) The variable "Speed" determines the falling speed of the apple. The apple falls down only when Y coordinate decreases. Thus, we need to multiply the variable "Speed" by -1 to generate a negative value.

```

change y by Speed * -1
  
```

Give students time to complete the task and present their works

The sample program, **Gear Knob-Change the Falling Speed**, can be found in the program files.

Change the Falling Speed	
Change the falling speed of the apple by toggling the gear knob.	
Device-Codey	Sprite-Apple
<p>1. To narrow down the range of the variable Speed between 0 and 10, divide the value of the gear knob, then assign the calculated value to the variable.</p> <p>2. Use the gear knob to adjust the falling speed of the apple.</p>	<p>1. The variable "Speed" determines the falling speed of the apple, which means the falling speed can be controlled with the gear knob.</p>
<p><input type="checkbox"/> Connect Codey Rocky to mBlock 5</p> <p>1. Connect via Bluetooth or a USB cable.</p> <p><input type="checkbox"/> Open the Existing Program</p> <p>1. Open the program of Catch the Apples or rewrite the program with mBlock 5.</p> <p>2. Save the program as Change the Falling Speed.</p> <p><input type="checkbox"/> Write Programs</p> <p>1. Device-Codey: Create the variable "Speed".</p> <p>2. Device-Codey: Limit the range of the value of the gear knob within 0 to 10 and assign the value to the variable "Speed".</p> <div style="border: 1px solid #ccc; padding: 5px; margin: 10px 0;">  </div> <p>3. Device-Codey: Instead of showing Left/Right arrows, the LED screen displays the falling speed in real time.</p> <p>4. Sprite-Apple: The variable "Speed" determines the falling speed of the apple. The apple falls as its Y coordinate decreases. Multiply "Speed" by -1 to make it negative.</p> <div style="border: 1px solid #ccc; padding: 5px; margin: 10px 0;">  </div> <p>5. Upload the program and watch how it works</p> <p><input type="checkbox"/> Extension Task</p> <p>1. Add a variable "Score" to store the number of apples caught by the monkey.</p>	

4. Wrap-up

At this point, you need to revisit what have been learned in today's and the previous lessons, location, functions and applications of Codey's gear knob. To end, prepare students for the next lesson.

5. Students' Self-review

Have students complete the self-review report.

Lesson 27 I'm a Good Guesser

Lesson Plan

Overview:

Learn about color sensors and complete tasks.

Teaching Objectives:

Students will

1. Learn about color sensors and what they are used for;
2. Know where the color sensor of Codey Rocky is located;
3. Use the color sensor to complete game tasks.

Preparation:

1. Whiteboard and markers (OR blackboard & chalks);
2. One Codey Rocky and a USB cable per student (OR 2-3 students share one set);
3. A computer with mBlock installed per student (OR 2-3 students share one set);
4. Challenge cards "What My Favorite Color is?".

Difficulties:

1. How to identify the color sensor and use it to perform tasks;
2. How to program Codey Rocky to tell others what your favorite color is.

Teaching Procedure:

1. Situated Learning

Ask students: "What will you focus on when you are matching clothes?"

Possible answer: Style, Color, Size, Season...

Introduce a new topic: "When we are matching clothes, color matching is always an important factor that we'll consider. We identify colors with our eyes and match colors based on our sense of color. However, for those people who are deprived of the ability to see things due to incidents or diseases, they just can't see colors like we do anymore. In this case, they can only "feel" the clothes by touching them. So, what will you do to help them out? "

Brainstorm session. Ask students to share their ideas. Then give a summary of the students' answers and show them a product design - Talking Color Identifiers. Talking color identifier are designed to detect colors by saying the color names. They are perfect for people with low vision when they are buying clothes or matching colors.



Source: <http://www.thinker360.com/page/3738>

Introduce the Talking Color Identifier: "In the Talking Color Identifier, a color sensor is used to help detect colors. And today we are going to learn about how color sensors work and their applications. Codey Rocky also has a color sensor. Next, we are going to play a game, I'm a Good Guesser. In the game, we'll have to figure out what Codey Rocky's favorite color is and where the color sensor is located."

2. Lead-in Game

Steps:

1) Guess: Show students color cards: red, green, blue, yellow, cyan, purple, black and white. Ask them to find out which color is Codey Rocky's favorite. You can use the following card to give students some hints.

I'm a good guesser		
Codey Rocky: I like spring. In spring, everything comes back to life again and the air is fresh and clean. All I can see is the color___ , which really cheers me up.		
I guess the color that Codey Rocky likes is:	Because:	In fact, the color that Codey Rocky likes is:

Have students write down their own answers and ask representatives to share his or her idea to the class .

Note down their answers on the board.

2) Test: The animations on the LED display tells us which color is Codey Rocky's

favorite.

The teacher needs to upload programs to Codey Rocky beforehand (refer to program "Color Sensor - Codey Rocky's Favorite Color"). Toggle the color sensor to the front and place color cards in front of the color sensor. One card each time. Codey Rocky will react based on the color cards. If Codey Rocky laughs and animations linked to the color starts playing on the LED screen, then it means it likes the color; if Codey Rocky sounds like it's embarrassed and displays a face wearing sunglasses on the LED screen, then it doesn't like the color.

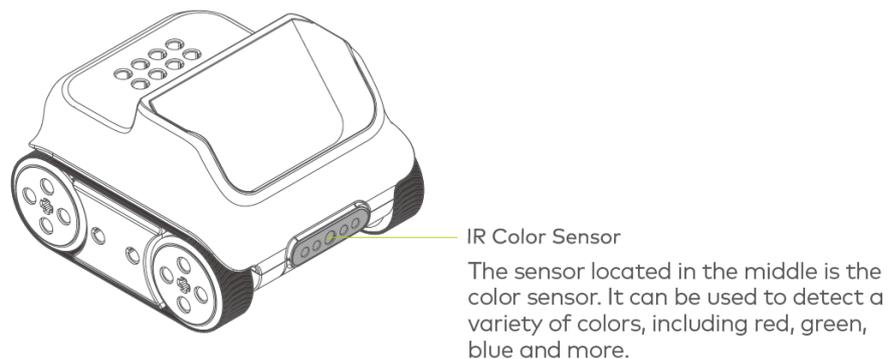
Before running the program, ask students to focus on the whole process of how Codey Rocky detects colors and to answer the following questions later:

- a) Where is the color sensor located?
- b) In spring, what color does Codey Rocky likes most?
- 3) At the end of the game, reveal the final answer.

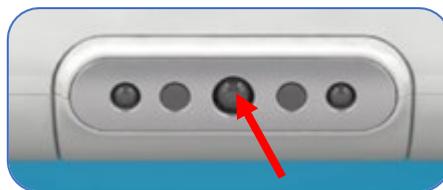
3. Explain New Concept

Ask students: " Who can tell me where the color sensor is?"

Give a description: "At the front lower part of Rocky there are several sensors, including a color sensor, a grayscale sensor, an IR proximity sensor and more."



The sensor located in the middle is the color sensor. It can be used to detect a variety of colors, including red, green, blue and more.



Programs:



Explain how to program the color sensor: "If we want to program the color sensor, we'll need to use these blue Sensing blocks. We can pull the drop-down menu to reset the color to be detected."



4. Coding Task

Navigate students: "You already know the location and functions of the color sensor. Now you can use Codey Rocky to create a game that you can play with your friends: let them

guess what your favorite color is. There is a table below. Write down hints in the table. For instance, if you like blue and white, then the hints could be like: **I like a sunny day because I can see the color _____ in a sunny day when I look up in the sky.** Upload the programs to Codey Rocky and define the animation effects. The animations on the LED screen will change based on the color th detects."

Encourage students to define the animations, sounds and lights as they would like. What you just did in the last session could be a good example for students to follow: Codey Rocky likes red flowers so when it detects red, it will move forward, laugh and display a blooming flower on the LED screen.

Ask students to present their works to the class and tell others to guess what color the speaker likes most. On the following table, write down the answers. Then run the program to test whether the

Or, you can tell students to find partners to share their works with. Partners should make a guess respectively and run the programs to test their speculations.

What is my favorite color?		
Hints:		
My favorite color is:	Because:	In fact, the color I like is:

5. Wrap up

Give a summary of where the color sensor is located and what it can be used for. This lesson already introduces the role of color sensors in helping visually impaired people match clothes.

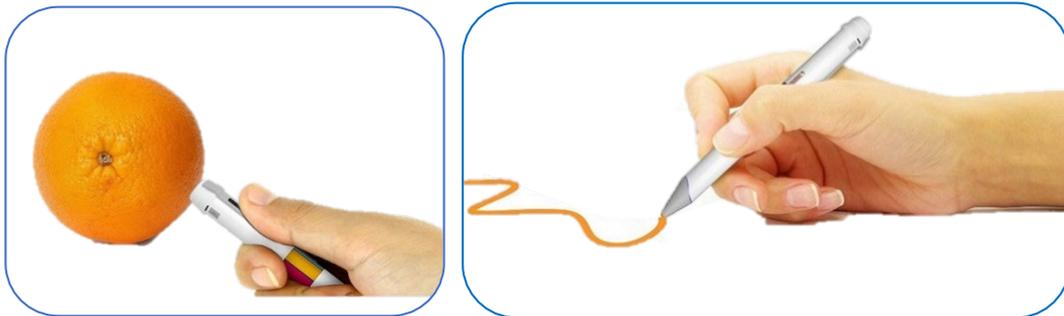
Have students brainstorm about other applications of colors sensors in our everyday life. You can give students some examples:

- 1) You buy a green dress and you want to match a top with the dress. In this case, a color sensor will help you capture the color of your dress and find matching colors;
- 2) A robot with a built-in color sensor can sing different songs based on the colors it

scans;

3) Color sensors are used in white canes to detect the colors of stoplights and tactile pavings, which could mean a lot for people with low vision;

4) Color picker pens have built-in color sensors that enable them to capture the color from the surface of an object. Using the pens, children and painters can draw whatever they like on paper or digital devices with the colors they capture.



Pictures from the Internet

To sum it up, color sensors are widely applied in the field of artificial intelligence. Color sensors capture colors in the real world and robots then react to the colors the sensors scan, which simulates the process of how our eyes recognize colors.

Lesson 28 Stoplight

Lesson Plan

Overview:

Apply the knowledge of color sensors to complete tasks.

Teaching Objectives:

Students will:

1. Know where the color sensor is located and how to program the color sensor;
2. Use the color sensor to complete tasks;
3. Use the color sensor to create stories, animations and games.

Preparation:

1. Whiteboard and markers
2. One Codey Rocky and a Bluetooth adapter (OR a USB cable) per student (OR two students share one set);
3. A computer with mBlock installed per student (OR 2-3 students share one set);
4. Challenge Cards

Prior Knowledge:

This lesson requires students to

1. Have a basic understanding of where the color sensor is located and its functions;
2. Know how to program the color sensor to detect colors;
3. Know how to use Broadcast.

Teaching Procedures:

1. Review

Briefly review what students learned in the previous lesson: location, functions and applications of the color sensor.

2. Task 1 Stoplight

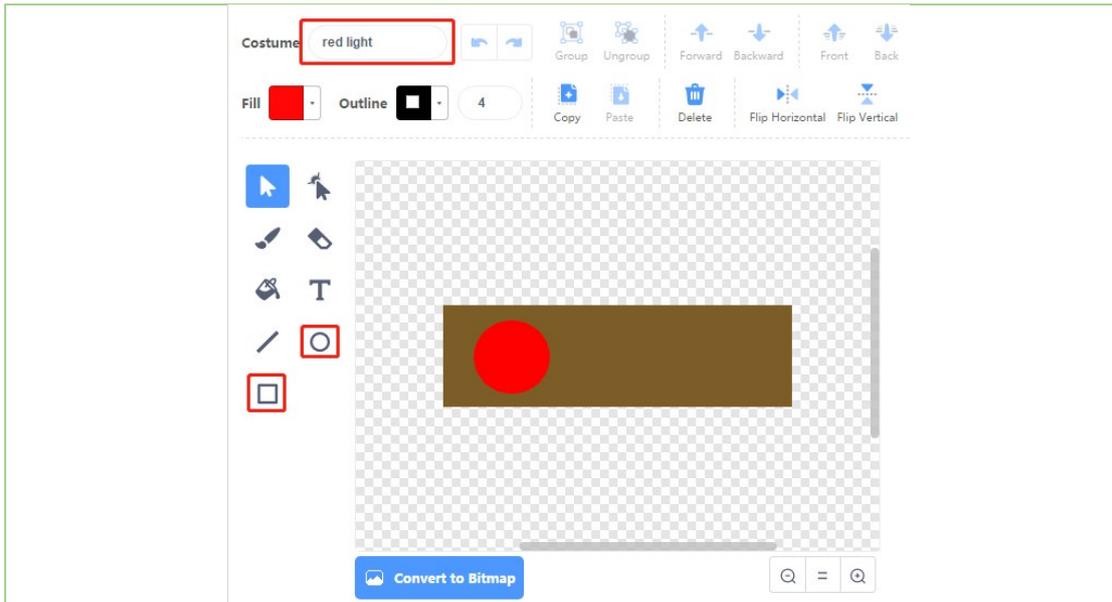
Navigate students: "In the previous lesson, we learned about the basics of color sensors. Next, we are going to use the color sensor to complete some challenges. Let's take a look at the first challenge we've got."

Briefly describe the rules to students. This can help those who have no patience for

written texts better understand the rules. Of course, students can follow the instructions on the Challenge Card to complete the task step by step. When they finish, students are expected to present their works to the class. For this challenge, students need to control the movements of the car on the stage using the color sensor. Have students work in pairs to complete this coding task.

Sample programs can be found in the program files, **Color Sensor - Stoplight**.

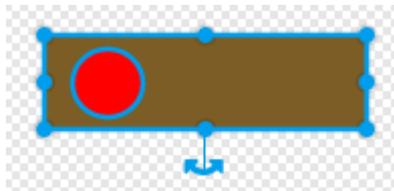
Stoplight		
<p>Turn Codey Rocky into a stoplight. The car on the stage will decide either to go or stop based on the commands given by Codey Rocky: red light for stop and green light for go.</p>		
Device-Codey	Stage Sprite-Car	Stage Sprite-Stoplight
1. Detect red and green cards. 2. Send messages "red light "or "green light".	1. When receiving the message "red light", the car will stop. 2. When receiving the message "green light", the car will move forward and bounce back when hitting the edge.	1. When receiving the message "red light", the stoplight turns red. 2. When receiving the message "green light", the stoplight turns green.
<p><input type="checkbox"/> Connect Codey Rocky to mBlock 5</p> 1. Connect Codey via Bluetooth or a USB cable.		
<p><input type="checkbox"/> Add Sprites and Background</p> 1. Under the Sprites tab, you'll see an icon "+". Click the icon to add the sprite C-mbot-happy2-a. Rename the sprite "Car" and drag it to the bottom of the stage.		
		
2. Add another sprite Empty. Click Costumes and use the tools  and  to draw a red light. Name the sprite "red light".		
		



Add a blank costume.



Click the sprite "red light" and click  to  the square and circle you just draw into the blank costume.



Change the color to green and drag it to the right-hand side of the square. Rename the green circle "green light".



Rename the sprite "stoplight" and drag it to the upper left corner of the stage.



3. Select the background Space.



Write Programs

1. Device-Codey: When Codey Rocky starts up, the code to detect colors will run repeatedly. When the color is green, a message "green light" will be broadcast and then the RGB LED will turn green.



When the color is red, a message "red light" will be broadcast and the RGB LED will turn red.

2. Sprite-Car: When receiving the message "green light", the following code to control the motion of the car will run repeatedly: "move 10 steps", "if on edge, bounce", "set rotation style left-right".



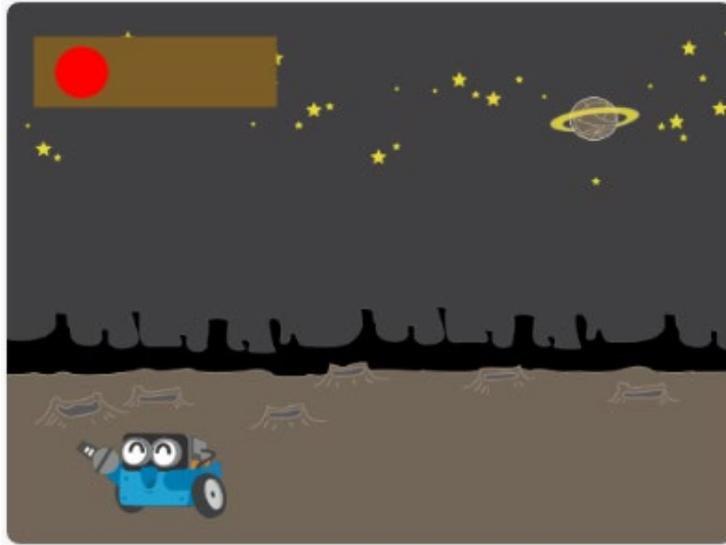
When receiving the message "red light", other scripts in this sprite will be stopped.



3. Sprite – Stoplight: When receiving the message "green light", the costume will be switched to the green light; when receiving the message "red light", the costume will be switched to the red light.

4. Upload the program to see how it works.

Referential Stage Design



Challenge

Use the color sensor to create new stories, animations or games.

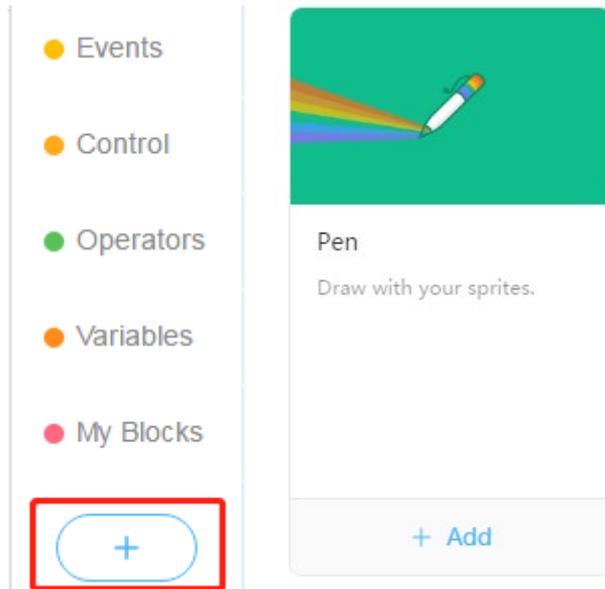
3. Task 2 A Blooming Flower

If time allows, have students take on Task 2.

Before students get started, introduce a new concept: Pen

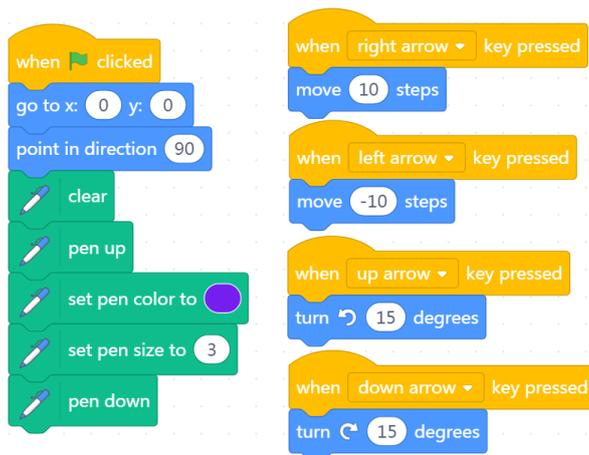
Each sprite has an invisible pen which comes in two states: Pen Up and Pen Down. If the Pen Up block is used, then the sprite will draw a trail wherever it goes. Of course, you can reset the width, color and shade of the trail. If the Pen Down block is being used, then the sprite will leave no trail. You can drag those red blocks out of the Pen category to define the pen.

By clicking the icon + as shown below, you can add the pen from the Extension Center.



You can use the red blocks as shown below to define the pen.

Try writing the following programs for the pen. Change the width and color of the pen each time to see what the difference is.



You can redefine the pen by resetting the parameters or using other blocks.

Using the pen and the Loop blocks, you can draw something fun.

The following session teaches students to use the tool pen to draw a flower.

1) First off, initialize the pen:

```

when I receive red
  clear
  pen up
  set pen size to 10
  set pen color to red
  point in direction 0
  go to x: pick random -150 to 150 y: 0
  pen down

```

Code



Stage Effect

2) Use a loop block in your code to draw half of the petal.

```

repeat 10
  move 5 steps
  turn 9 degrees

```

Code



Stage Effect

3) Add the **turn 90 degrees** block and another loop block to your code. Complete the petal.

```

repeat 2
  repeat 10
    move 5 steps
    turn 9 degrees
  turn 90 degrees

```

Code



Stage Effect

4) Add the **turn 72 degrees** block and another loop block to complete the flower.

```

repeat 5
  repeat 2
    repeat 10
      move 5 steps
      turn 9 degrees
    turn 90 degrees
  turn 72 degrees

```

Code



Stage Effect

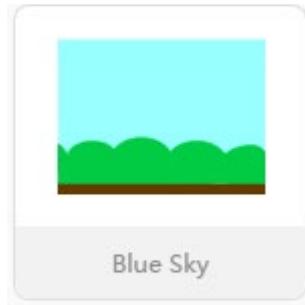
After students know how to draw a flower, briefly describe the rules of the task to students. This can help those who don't have patience for written texts better understand the steps.

The sample program can be found in the program file, **Color Sensor - A Blooming Flower**.

Tell students to complete the task by following the Challenge Card instructions. Ask students to present their works to the class.

A Blooming Flower	
Use the color sensor of Codey Rocky to detect colors and the sprite pen will paint flowers with the color that is detected.	
Device-Codey	Stage Sprite-Pen
1. Use the color sensor to detect colors and broadcast messages.	1. The sprite will draw corresponding flowers based on the messages it receives.
<input type="checkbox"/> Connect Codey Rocky to mBlock 5	
1. Connect Codey Rocky via Bluetooth or a USB cable.	
<input type="checkbox"/> Add Sprites and Backgrounds	
1. Click the sprite Panda to add the costume "Arrow1-a".	
	
2. Delete other costumes and leave the costume "Arrow1-a" there. Change the name of the sprite Panda to "Pen".	

3. Select the background "Blue Sky".



Write Programs

1. Device-Codey: When button A is pressed, if Codey Rocky detects items that are red, it will send the message "red" and the RGB LED will turn red.



2. Sprite –Pen: When receiving the message "red", the color of the pen will be set to red and the width will be set to 10. You can use multiple loop blocks in the code to draw red flowers on the stage.



3. Repeat the steps above to draw different colored flowers.

4. Upload the programs to Codey Rocky to test how they work exactly.

Challenge

1. Reset the looping times and rotation angles to draw flowers with different number of petals. For instance, you can draw flowers with 6 petals or 10 petals.



2. Use the tool pen to paint green leaves and stems.



4. Wrap up

Summarize what students learned in last and today's lessons: location, functions and applications of the color sensor.

Conclude the lesson with a preview of what students will learn in the next lesson.

5. Students' Self Review

Have students fill the self-review.

Lesson 29 Sensing Motions

Lesson Plan

Overview:

Learn about the gyroscope block and grasp a basic understanding of how gyroscopes work.

Teaching Objectives:

Students will

1. Learn about gyroscopes and their applications;
2. Understand how gyroscopes work and complete tasks using gyroscopes;

Preparation:

1. Whiteboard and markers
2. One Codey Rocky and a computer with mBlock installed per student
3. Lanyards

Difficulties & Key Points:

1. Learn about what a gyroscope can do;
2. Learn about the orientations.

Teaching Procedures:

1. Lead-in Game

Tell students that they are going to play a game to see who's faster at shaking hands. Have students compete with their deskmates first. In the end, ask the 3 fastest students to challenge each other.

Explain the rules:

- Upload the programs to Codey Rocky. (Remind students that they should not see the programs at this stage/OR the teacher can help upload programs for students)
- Attach the lanyard to Codey Rocky and remember to wear the lanyard around the wrist before shaking Codey Rocky.
- Before the game starts, press button A to clear the previous data.
- When the game is over, the LED screen will show the final scores.

- The teacher will stop the game when time is up.

Ask students: "You just had some fun in the game. Now, who can tell me why Codey is able to detect our movements?"

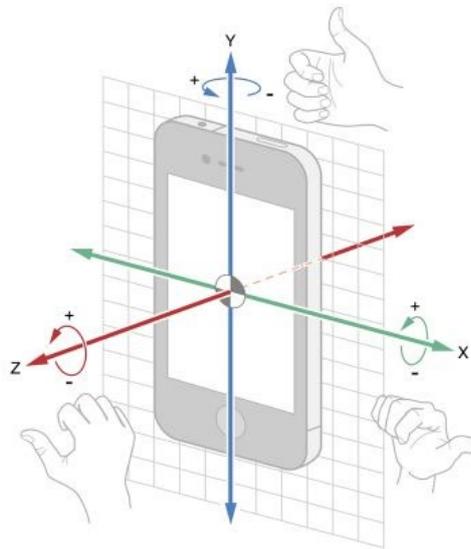
Possible answers: Because Codey has a sensor that enables it to detect our movements.

The sensor could be a motion sensor, like a gyroscope.

Explain to the class: Codey has a built-in sensor which is called gyroscope. With the gyroscope, Codey owns the ability to detect movements. For instance, in the game, we kept shaking Codey. And the gyroscope can measure the rotations of Codey when it is being shaken.

2.Explain New Concept

In the main board of Codey there is a sensor called gyroscope. The gyroscope can sense three-dimensional movements (as shown below). Therefore, we use the gyroscope to detect the rotational movements of Codey around the three axes and calculate the rotation angles. By taking advantage of the calculations, we can program Codey to perform different tasks.



Picture from the Internet

When programming the gyroscope, we will need coding blocks linked to the gyroscope. Most of the blocks can be found in the **Sensing** Category. And many of them have a drop-down list, which is quite clear in the picture below. By changing the options, we can code the gyroscope to detect the movements of Codey around a specific axis. In the following session,

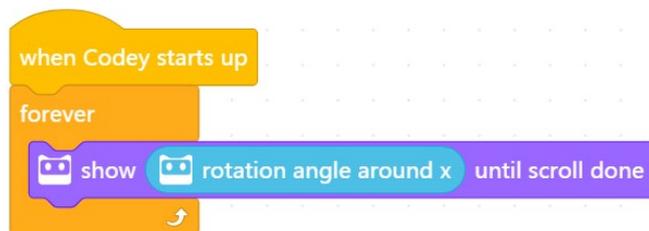
students will learn more about the uses of these coding blocks.



In the **Events** category, we can also find coding blocks related to the gyroscope.



The gyroscope can measure rotations in three dimensions. Upload the following program to Codey to see how the rotation angle around the x-axis changes.



Then reset the axis in the code and try tilting Codey in different directions to test in which situation the rotation angle will change rapidly.

Tips:

- 1) Please calibrate the gyro first to ensure the instrument accuracy. There is a gear icon on top of the "Upload Program" icon. Click the gear icon and follow the instructions to calibrate the gyro.

Applications:

The development of modern technology brings gyroscopes into our everyday life. For instance, gyroscopes are used in smart phones to reduce the jitter generated by the movements

of photographers and used in balance cars to maintain the balance. Similarly, drones are equipped with gyroscopes, which helps keep the drones stable during the flight.

3. Coding Task

After students gain a basic understanding of what a gyroscope is, ask them to complete a task by applying what they've learned. The task is about making a Stand Up alarm. No one will deny that sitting there is comfortable. However, sitting too much could do harm to our body. To enjoy a healthier life, we can use the gyroscope to make ourselves a "Stand Up" alarm.

1) Goal: When button A is pressed, Codey will start counting. And if we sit there too long, Codey will sound an alarm, reminding us to stand up and take a walk.

2) Notes: Typically, we should stand up and move every 60 minutes. However, considering the limited class time, we set the time to 30 seconds. We'll need the timer blocks in the code to determine and reset the time.

4. Take the Challenge

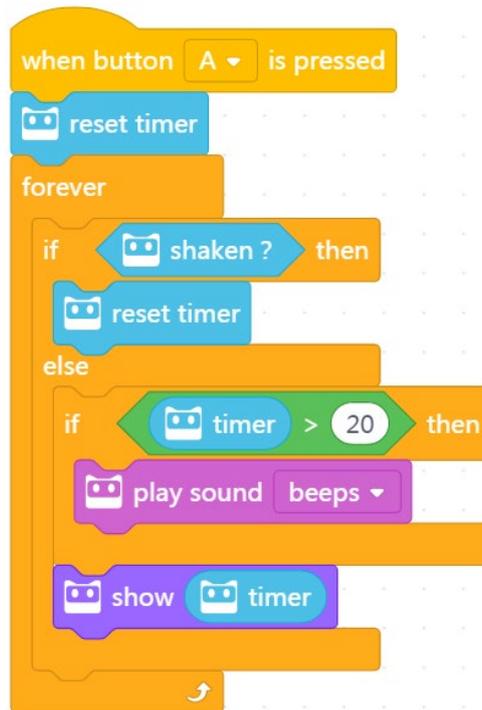
- Have students work on the task.
- When students complete the task, tell them to present their works to the class.

5. Working Principle Analysis

After students complete the task, invite three students to present their code to the class. Tell them to explain how their code work and where to put Codey.

Explain to students: "You now know how to detect movements using Codey. But where to put Codey is an important thing we should consider. Putting Codey on our arm could be a bad choice because the jitters of our arms might lead to inaccuracies. We can tie Codey to our body using the lanyards."

Sample program:



Explain the principles: When button A is pressed, the timer will start counting from 0. The code to check for Codey being shaken will run repeatedly. If Codey is shaken, it means that you are moving; if Codey is not shaken, then Codey will sound an alarm when the timer reaches 20.

Tips: 1) We don't have to put the **show timer until done** block in our code, actually. Displaying the time simply helps us debug the program.

2) Apart from playing sounds, we can also program the LED screen to display images when Codey sounds an alarm.

Extension task: Turn Codey into a wearable that can track sports performances.

6. Wrap up

The gyroscope enables Codey Rocky to detect movements, such as tilt and rotation. It can also measure the rotation angles which we can use in our code to do many things.

Lesson 30 Playing Sports

Lesson Plan

Overview:

Use the gyroscope to accomplish tasks with different levels of difficulty.

Teaching Objectives:

Students will

1. Learn about gyroscopes and their applications;
2. Understand how a gyroscope work;
3. Know how to program the gyroscope to interact with the mBlock stage.

Preparation:

1. Whiteboard and markers
2. One Codey Rocky and a USB cable per student (OR two students share one set);
3. One computer with mBlock installed per student (OR two students share one set).

Prior Knowledge:

This lesson requires students to

1. Know how to use physical buttons;
2. Have basic knowledge of what a gyroscope is and how it works.

Teaching Procedures:

1. Review

Briefly review what's learned in the previous lesson: what a gyroscope can do and its applications in everyday life.

2. Task 1 Jumping Codey Rocky 2.0

Briefly describe the rules to students. This can help those who have no patience for written texts better understand the rules. Of course, students can follow the instructions on the Challenge Card to complete the task step by step. After they finish, ask students to present their works to the class. For this challenge, have students work in pairs.

Jumping Codey Rocky 2.0

Use the gyroscope sensor to help Codey Rocky avoid obstacles in the way.

Device-Codey Rocky	Stage Sprite—Codey	Stage Sprite-Obstacle FIRE
<p>1. Use the gyroscope to control Codey's jumps.</p> <p>2. The game begins when button A is pressed.</p> <p>3. When the game is over, the LED screen shows a sad face.</p>	<p>1. When the game begins, the sprite appears in the middle of the stage.</p> <p>2. When the sprite meets an obstacle, it has to jump over the obstacle; the game is over once the sprite touches an obstacle.</p>	<p>1. When the game begins, the sprite keeps moving from the right side of the stage to the left side.</p>

Connect Codey Rocky to mBlock 5

Connect Codey Rocky via Bluetooth or a USB cable.

Add/Delete Stage Sprites

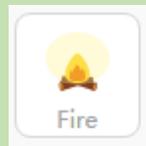
1. Add the sprite C-codey-rocky1-a

- Rename it as Codey
- Adjust it to the appropriate size



2. Add the sprite bonfire

- Rename it as Fire
- Adjust it to the appropriate size



3. Delete the sprite Panda

Write Programs

Device: When button A is pressed, Codey Rocky will show a smiley face and broadcast "start".

Sprite - Codey: When the game begins, the sprite appears on the middle left hand side of the stage.
When the sprite Codey fails to avoid the sprite Fire, a message "game over" will be broadcast. Then, the sprite will end up with a loser gesture and all the scripts will be stopped.

```
go to x: -90 y: -40
```

Stage sprite-Fire: When receiving the message "start", the sprite Fire will move to the right-hand side of the stage. In the next 3.5 seconds, the sprite, moving from the right-hand side to the left side, will end up disappearing.

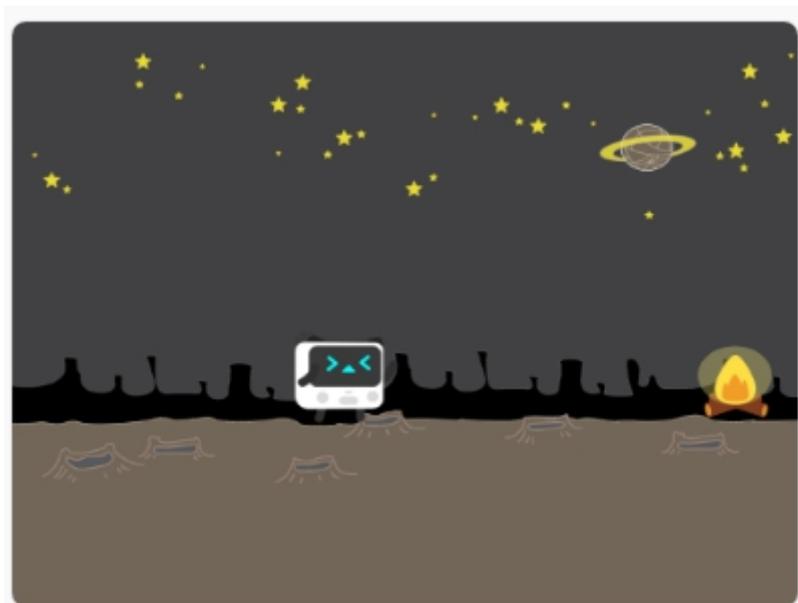
Device: When Codey Rocky is powered on, it starts sensing objects; when Codey is shaken, a message "jump" will be sent.

Stage sprite-Codey: When receiving the message "jump", Codey will move up to the preset position within 0.4 seconds and fall within 0.8 seconds, which is a simulation of the jumping process.

```
glide 0.4 secs to x: -90 y: 110  
glide 0.8 secs to x: -90 y: -40
```

Device: When receiving the message "game over", the LED screen will display a sad face.

Referential Stage Design



3. Task 2 Penalty Kick

Penalty Kick

Using the gyroscope sensor, you can turn the sprite Codey into a goalkeeper.

Device-Codey Rocky	Stage Sprite-Codey	Stage Sprite-football	Background Goal
1. Use the gyroscope to control the stage sprite. 2. Display facial expressions.	1. Move (left/right) in sync with the gyroscope.	1. Press the space key to shoot the ball. 2. Check for the soccer ball going into the net, the goal being missed or the ball being blocked.	No code required for the background

Connect Codey Rocky to mBlock 5

Connect Codey Rocky via Bluetooth or a USB cable.

Add Sprites

Pick the sprite soccer ball from the Sprite Library.

Click the sprite soccer ball to add it to the stage;

Adjust its size to make it fit the stage.



Add the sprite C-codey-rocky.

Rename the sprite "Codey";

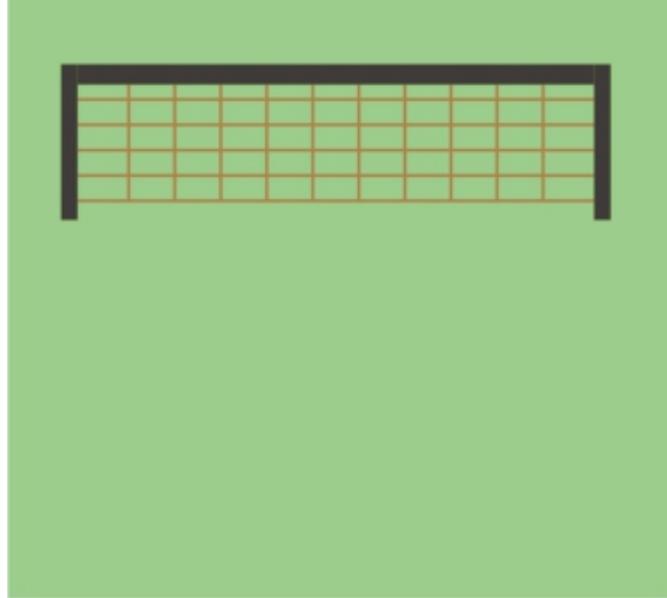
Adjust its size to make it fit the stage.



Add the background – Gate.

Click Background – Costumes- Add Backgrounds- My Backdrops- upload file gate.png

Adjust its size to cover the whole stage.



Delete the sprite-Panda.

Write Programs

Device: Create a communication variable MOVE to store the y-axis value of the gyroscope. When Codey starts up, it will assign the value of the gyroscope to the variable MOVE repeatedly.



Stage sprite-Codey: When the green flag is clicked, the code to make the sprite Codey move in sync with the variable will run repeatedly.

Note: Multiplying the variable with 1.5 can make the sprite move faster on the stage.)



Stage sprite-soccer ball: When the key space is pressed, the soccer ball will move to the initial position.



Wait 1 second and point in random direction between -60 and 60.

xt, the code to make the soccer ball move forward 30 steps will run repeatedly until one of the following situations happens: ① The ball hits the edge of the stage; ② The ball hits Codey; ③ The ball hits the net; ④ The ball hits the posts or the crossbars.



Stage sprite-soccer ball: If the soccer ball hits the sprite Codey, the posts or the crossbars, then the ball will move -100 steps, simulating the process of bouncing back. If the ball hits the net, then it will move forward 30 steps and broadcast "Goal".

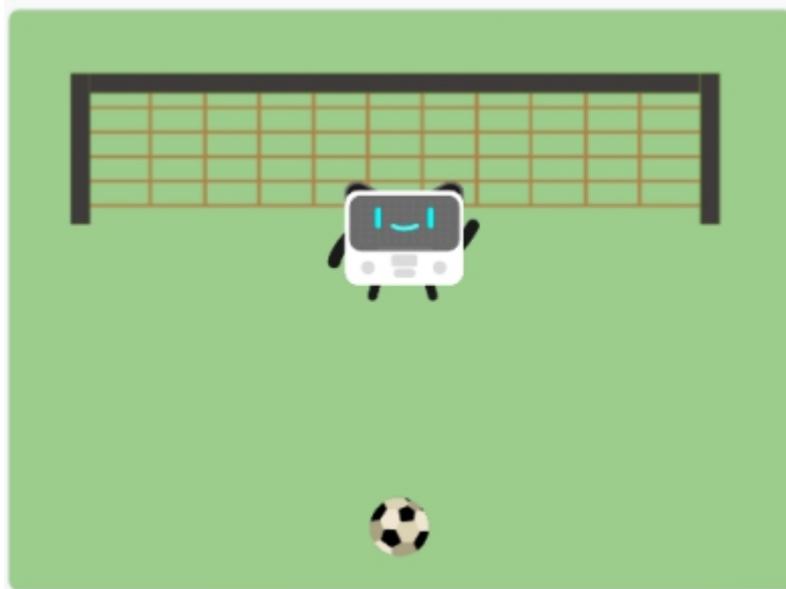
Device: When receiving the message "Goal", Codey Rocky will show a sad face for 1 second.

Device: When button C is pressed, the rotation angle around the y-axis will be reset.



Note: You can use the block above to reset the value of the gyro.

Referential Stage Design



3. Independent Practice

- ① Students work in pairs to complete the task.
- ② Raise hands when the task is done.

4. Presentation

5. Wrap up

We can program the gyroscope to detect movements and measure rotations. Using the measurements and the coordinate system of the stage, you can easily interact with the stage sprites, for instance, controlling them to move as you would like.

Lesson 31 The Greeter Codey

Lesson Plan

Overview:

Learn about ultrasonic sensors and have a basic understanding of how they work.

Teaching Objectives:

Students will

1. Learn about ultrasonic sensors and their applications;
2. Understand what ultrasonic sensors can do and complete tasks using the Ultrasonic Sensor block;
3. Know how to read values of the Ultrasonic Sensor block and understand how the sensor works.

Preparation:

1. Whiteboard and markers
2. One Codey Rocky per student (OR two students share one set), a USB cable and an Ultrasonic Sensor block;
3. One computer with mBlock installed per student (OR two student share one set).

Prior Knowledge:

This lesson requires students to

1. Master the basics of ultrasonic sensors;
2. Know how to program the ultrasonic sensor block using mBlock 5;
3. Know how to read the values of the gyroscope and compare variables.

Teaching Procedures:

1. Situated Learning

Ask students: When you enter a shopping mall, you need to walk through a door. In most cases, the door will automatically open once you come close. Do you know why?

Possible answer: Because the door is electric. It can sense people when someone comes close.

Ask again: But how does an automatic door sense people?

Possible answer: It's because of the IR sensors or ultrasonic sensors.

Introduce to students: For automatic doors, there are many ways to sense people. Today, we are going to learn about one of those sensors that is applied in automatic door control system, **Ultrasonic Sensor**. Does anyone know what "ultrasonic" means?

Have students brainstorm about the question.

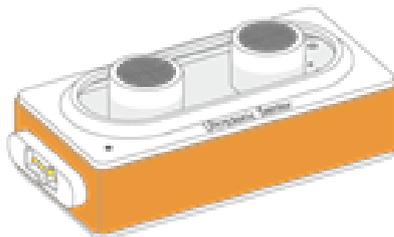
The frequency range of human hearing falls between 20Hz and 20000Hz. Those frequencies greater than 20000Hz are called **ultrasounds**. Ultrasonic sensors are often used in robots to help them avoid obstacles and measure distance.

The following picture shows the voice frequencies and audible frequencies of some animals:

Voice and Hearing Frequency Ranges f/Hz		
	Voice Frequency (Hz)	Hearing Frequency (Hz)
Human	85~1100	20~20000
Dogs	452~1800	15~50000
Cats	760~1500	60~65000
Bats	10000~120000	1000~120000
Dolphins	7000~120000	150~150000

2. Explain New Concept

Codey Rocky has no built-in ultrasonic sensor. Luckily, it can borrow an Ultrasonic Sensor block from its friend, Neuron.



An ultrasonic sensor comprises a transmitter and a receiver. Through the transmission and reception of ultrasounds, ultrasonic sensors can detect objects in its view and determine the distance to an object. The measurement range of the Neuron Ultrasonic Sensor block is 3cm~300cm.

Remind students: "We will need these Neuron-related purple blocks if we want to

program the Ultrasonic Sensor to do things."

 ultrasonic sensor 1 distance (cm)

3. Game Task The Greeter Codey

Tell students that they are going to play a game called **The Greeter Codey**. In this game, students will learn to

Demonstrate:

- 1) Connect Codey to the Neuron Ultrasonic Sensor block;
- 2) When button A is pressed: The LED screen displays 2 hearts and the indicator turns red.
- 3) Detect obstacles: When your hand gets close to the Ultrasonic Sensor block, the LED screen displays the word "Hello" and the indicator turns green;

Explain Game Rules

- Upload programs to Codey (Remind students that they should not see the programs at this stage/OR the teacher can help upload programs for students)
- Press button A to start the game.

Independent Practice

- Upload programs to Codey.
- Work on the challenge.
- Raise hands when the task is done. And present the works to the class.

4. Working Principle Analysis

Ask your students: " How does Codey detect obstacles in front of it? If the obstacle is too close, what will you see on the LED screen? Tell us what our idea is."

Possible answer: " The measurement range of the Neuron Ultrasonic Sensor is 3cm~300cm. Codey will blink its eyes when it senses an object within 10cm. If an object is too close, the LED screen will just display 2 hearts."

Give explanations: " Yes. In the code, we set a distance limit for the Ultrasonic Sensor, which is 10cm. Only when Codey detects an object within 10cm will it say hello to you."

This is because the measurement range of the Neuron Ultrasonic Sensor is 3cm~300cm.

Therefore, an object which is too close to the sensor can not be detected. And the max distance is 300cm. The LED screen will display 2 hearts when an object is too far (exceed 300cm).

Main Program:

when button A is pressed

forever

if ultrasonic sensor 1 distance (cm) < 10 then

- RGB LED lights up (green)
- play sound hello
- show image (winking eyes) for 0.4 secs
- show image (winking eyes) for 0.4 secs

else

- show image (red eyes)
- RGB LED lights up (red)

Set the detection range to "<10".

The indicator turns green, Codey plays a sound and winks when an object is detected.

The indicator turns red and the LED screen shows hearts when no object is detected.

5. Task Car Reversing Aid System

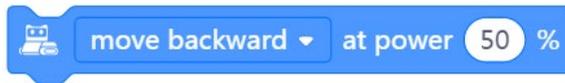
Navigate students: "Ultrasonic sensors can be used to detect objects and determine the distance to an object. Now, you can apply what you just learned to make yourself a car reversing aid system."

Challenge — Car Reversing Aid System

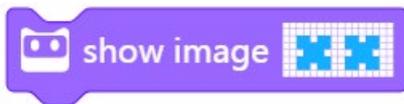
Combine Codey Rocky with the Neuron Ultrasonic Sensor and you can get a superb car reversing aid system.

Connect the ultrasonic sensor to the rear of Codey Rocky, helping alert Codey Rocky of obstacles.

Codey Rocky shows "start" on its LED screen when it starts up. The indicator turns red and after 1 second, the car starts driving in reverse.



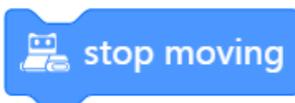
When detecting an object within 10cm at the back, the car stops and the LED screen shows a surprised face;



Wait 2 seconds. If no object is detected, the car keeps driving in reverse until it detects an object;



Wait 2 seconds. If an object is detected, the indicator turns black and the car stops.



6. Independent Practice

- ① Have students work in pairs to complete the task.
- ② Ask students to raise hands when the task is finished.

7. Presentation

8. Wrap up

Ultrasonic sensors detect objects and determine the distance to an object using the principle of reflections of ultrasounds. An ultrasonic sensor will send sound pulses at intervals. When these pulses hit objects, they will be bounced back. Then the ultrasonic sensor will measure the distance to the object based on the time that it takes the echo to return to the sensor. The Neuron Ultrasonic Sensor has a measurement range spanning from 3cm~300cm, which means it can only detect an object within this range.

Lesson 32 Who has the fastest hand?

Lesson Plan

Overview:

Learn how to complete game tasks by using the Neuron Ultrasonic Sensor and the mBlock stage.

Teaching Objectives:

Students will

1. Learn about ultrasonic sensors and their applications;
2. Understand what ultrasonic sensors can do and complete game tasks.

Preparation:

1. Whiteboard and markers;
2. One Codey Rocky per student(OR two students share one set), a USB cable and a Neuron Ultrasonic Sensor block;
3. One computer with mBlock installed per student (OR two students share one set).

Prior Knowledge:

This lesson requires students to

1. Master the basics of ultrasonic sensors;
2. Know how to program the Neuron Ultrasonic Sensor block in mBlock.

Teaching Procedures:

1. Review

Briefly review what's learned in the previous lesson: features and applications of ultrasonic sensors.

2. Task Who has the fastest hand?

Briefly describe the rules to students. This can help those who have no patience for written texts better understand the rules. Of course, students can follow the instructions on the Challenge Card to complete the task step by step. When they finish, students are expected to present their works to the class.

Who has the fastest hand?

Use the ultrasonic sensor to challenge your friend. Get your hand near the Neuron ultrasonic sensor and each time the sensor detects your hand, the number on Codey's LED screen will increase by 1. Ask your friend to do the same thing. Limit the time to 30 seconds. Compete with your friend to see who has the fastest hand?

Device-Codey Rocky	Stage Sprite-Bulb	Backdrop-Spotlight-stage3
<p>1. The LED screen display a smiley face when Codey Rocky starts up;</p> <p>2. Using the Funny Touch, you can control Codey to send messages to the stage;</p> <p>3. The game starts when button A is pressed;</p> <p>4. The LED screen shows "Game over" when the game comes to the end.</p>	<p>1. When the game starts, the sprite appears in the middle of the stage;</p> <p>2. The sprite will keep changing its brightness as a way to tell us whether there is an obstacle in front of the ultrasonic sensor. It will illuminate when the ultrasonic sensor detects an object;</p> <p>3. When the game is over, the sprite will end up in the final look.</p>	<p>1. When the game begins, the sprite starts changing its color.</p>

Connect Codey Rocky to mBlock 5

Connect Codey Rocky via Bluetooth or a USB cable.

Add/Delete Stage Sprites

Add the sprite Bulb.

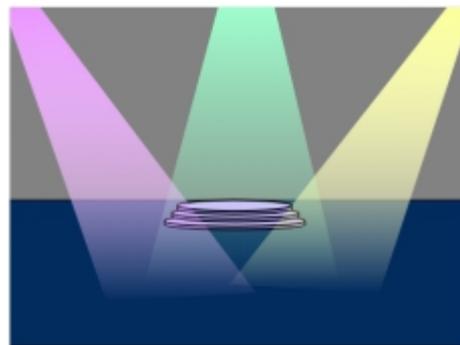
Adjust its size to make it fit the stage.



Delete the sprite Panda.

Select the backdrop: Spotlight-stage3

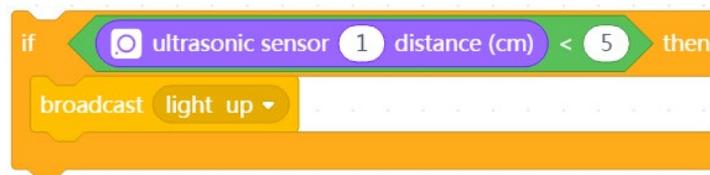
Define the color of each spotlight.



Write Programs

Device: When button A is pressed, the LED light will start flashing. Codey will show a smiley face, start counting down from 3 and show the word "Go".

Device: When Codey Rocky starts up, the ultrasonic sensor will turn on its detection mode. The number on the LED screen increases by 1 each time the ultrasonic sensor detects an obstacle.



Stage sprite-Bulb: The sprite appears in the upper middle of the stage, changing its costume to "bulb-1" when receiving the message "light up"; otherwise, it will change the costume to "bulb-2".

When the time runs out, the game will be over and all the scripts in this sprite stop.



Referential Stage Design



Extension Task

Use variables in your code to achieve this effect: The background changes each time the number of obstacles detected by the ultrasonic sensor reaches 30.

An ultrasonic sensor comprises a transmitter and a receiver. Through the transmission and reception of ultrasounds, ultrasonic sensors can detect objects in its view and determine the distance to an object. The measurement range of the Neuron Ultrasonic Sensor block is 3cm~300cm.

Remind students: "We will need these Neuron-related purple blocks if we want to program the Ultrasonic Sensor in mBlock."



3. Independent Practice

- ① Have students work in pairs to complete the task.
- ② Raise hands when the task is finished.

4. Presentation

5. Wrap up

In this lesson, students learned to program the Neuron Ultrasonic Sensor block to detect

obstacles. When an obstacle is detected, the stage sprite will follow specific commands.

Lesson 33 Speedy Colors

Lesson Plan

Overview:

Learn about LED strips and have a basic understanding of how they work.

Teaching Objectives:

Students will

1. Learn about LED strips and their applications;
2. Understand what the Neuron LED strip can do and use it to complete task.

Preparation:

1. Whiteboard and markers
2. One Codey Rocky a USB cable, a Neuron Ultrasonic Sensor block, an LED strip and an LED strip driver per student (OR two students share one set),;
3. One computer with mBlock installed per student (OR two students share one set).

Difficulties & Key Points:

1. Learn about LED strips.
2. Master how to program the Neuron LED strip in mBlock.

Teaching Procedures:

1. Situated Learning

Ask students: "Have you ever watched those quiz contests shows? Based on the answers, players score different points. When the player scores 5 points, a red light in front of him or her illuminates; a green light for 3 points; a yellow light for 0 point. At the end of the contest, we can determine the scores of each player based on the status of the lights. Do you know why?"

Possible answer: "Different colors refer to different points. At the end of the contest, we can determine the final scores of players based on the status of the lights."

Ask students: "Why does the light illuminate different colors then?"

Possible answer: "This is because the lights are controlled by someone. The lights illuminate different colors based on the commands they receive."

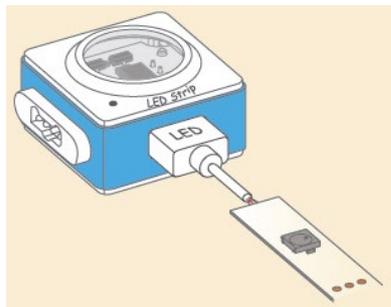
Introduce to students: "That's right! There are many ways to control the lights to

illuminate different colors. We use different colors to represent different scores. In today's lesson, we are going to control the Neuron LED strip to illuminate different colors using Codey Rocky and other Neuron sensors. Do you have any idea of how to make it happen?"

Have students brainstorm about the question.

2. Explain New Concept

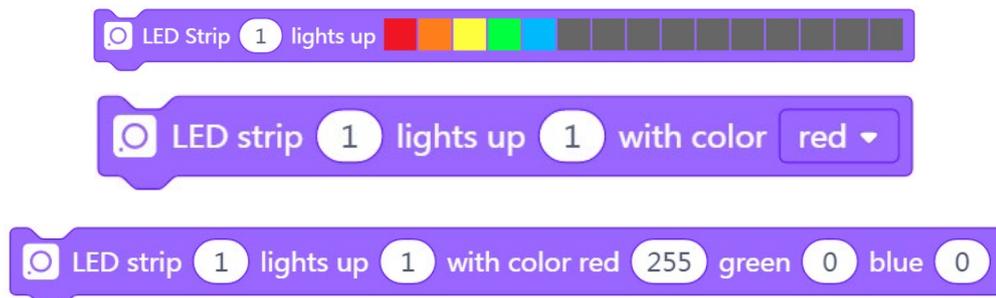
Codey Rocky has no built-in RGB LED indicator and LED strip. But it can borrow an LED strip from Neuron.



LED strip driver + LED strip

The Neuron LED strip contains 15 RGB LEDs that let it illuminate different colors. By programming the Neuron LED strip, we can change the color and brightness of each LED however you like. The Neuron LED strip must work with the LED strip driver if it needs connecting to other devices.

Remind students: "When programming the LED strip, we need to use these purple Neuron-related coding blocks."



3. Challenge Time

Tell students that they are going to play a game called Speedy Colors. This game is an upgraded version of the game Who has the fastest hand in the previous lesson.

Demonstration:

- 1) Connect the Nuero Ultrasonic Sensor block and the LED strip to Codey Rocky;
- 2) When button A is pressed: The LED screen will start counting down;
- 3) Detect obstacles: Place one hand in front of the Ultrasonic Sensor block. The number on the LED screen increases by 1 each time the Ultrasonic Sensor detects an item in its view;
- 4) LED strip turned on: As the number on the LED screen keeps growing, the RGB LEDs are turned on one by one;
- 5) When the countdown comes to an end, the game is over.



Picture from the Internet

Explain Game Rules

- Upload programs to Codey (Remind students that they should not see the programs at this stage/OR the teacher can help upload programs for students);
- Press button A to start the game;
- When the time runs out, the game is over.

Independent Practice

- Upload programs to Codey Rocky.
- Work in pairs to complete the task.
- Raise hands when the task is done.

4. Working Principle Analysis

Ask students: "Why do each LED strip have different amounts of LEDs turned on?
And how does each LED determine whether to turn on or not?"

Possible answer: "How many LEDs are turned on depends on the hand speed. Someone might be faster and someone might be slow. And as required by the program, only when a

specific amount of obstacles are detected will one LED be turned on. This means that those who have faster hand speed can turn on more LEDs."

Explain to students: "Yes, programs are written to turn on the LED under certain circumstances, like when a specific amount of obstacles are detected by the Ultrasonic Sensor. Therefore, those who have faster hand speed can turn on more LEDs. That's why your LED strips look different."

Here is the code for "Speedy Colors":

```
when button A is pressed
  set hand speed to 0
  set countdown timer to 30
  LED Strip 1 lights up
  broadcast start
  repeat until countdown timer = 0
    change countdown timer by -1
    wait 1 seconds
```

Make two variables "hand speed" and "countdown". Set the "countdown" to 30 seconds and define the initial status of the LED strip.

```
when I receive start
  repeat until countdown timer = 0
    if ultrasonic sensor 1 distance (cm) < 5 then
      change hand speed by 1
      wait 0.1 seconds
  show hand speed
  if hand speed > 10 then
    LED strip 1 lights up 手速 / 10 with color red pick random 1 to 255 green pick random 1 to 255 blue pick random 1 to 255
```

We write this piece of code to measure the distance to an obstacle. If the distance is less than 5 cm, the variable "hand speed" increases by 1.

One LED lights up with random color each time the variable "hand speed" is above 10.

5. Coding Task

Navigate students: Each LED strip has 15 RGB LEDs. And each of the LED is adjustable. Now, we need to use the LED strip to complete a task. That could be interesting. In the previous session, we used the LED strip to show how fast our hands can be. But you can use the strip to do many more things! Try rewriting the code to make the LED strip

interact with the stage!"

6. Wrap up

The Neuron LED strip includes 15 RGB LEDs so it can illuminate different colors. By programming, we can change the color and brightness of each RGB LED. And there's one more thing that needs attention. The LED strip needs to work with the LED strip driver to gain the ability to connect to other devices.

Lesson 34 Glowing Light

Lesson Plan

Overview:

Use the LED strip and the mBlock stage to complete tasks.

Teaching objectives:

Students will

1. Learn about LED strips and their applications;
2. Understand what the Neuron LED strip is used for and complete tasks. \

Preparation:

1. Whiteboard and markers;
2. One Codey Rocky, a USB cable, an LED strip and an LED strip driver per student

(OR two students share one set);

3. One computer with mBlock installed per student (OR two students share one set).

Prior Knowledge:

This lesson requires students to

1. Have basic knowledge of LED strips;
2. Master how to code the Neuron LED strip using mBlock.

Teaching Procedures:

1. Review

Briefly review the following sections:

- Functions and applications of LED strips
- Functions and applications of light sensors
- Functions and applications of IR sensor

2. Task: Glowing Light – Jump Up

Briefly describe the rules to students. This can help those who have no patience for written texts better understand the rules. Of course, students can follow the instructions on the Challenge Card to complete the task step by step. When they finish, students are expected to present their works to the class.

Glowing Light—Jump up

You will use the light sensor and the IR sensor in this task. When it gets dark and someone gets close, the LED strip starts turning on the lights; when all the lights are turned on, the sprites on the stage starts dancing.

Device- Codey Rocky	Stage Sprite —Dorian	Stage Sprite— Jamie	Stage Sprite— Casey	Stage Sprite— Dot	Stage Background— Spotlight-stage1
<p>1.Codey Rocky shows a smiley face when it starts up;</p> <p>2. Use the light sensor and to detect the changes of the light intensity and the IR sensor to tell whether someone is coming close.</p> <p>3. When it gets dark and someone is getting close, the LED strip turns on the RGB LEDs; when a specific number of</p>	<p>1. The sprite Dorian appears on the left side of the stage;</p> <p>2. The sprite begins dancing when it receives the message.</p>	<p>1. The sprite Jamie appears on the front right side of the stage when the game begins;</p> <p>2. The sprite begins dancing when it receives the message.</p>	<p>1. The sprite Casey appears in the middle of the stage when the game begins;</p> <p>2. The sprite begins dancing when it receives the message.</p>	<p>1.The sprite Dot appears on the front left side of the stage when the game begins;</p> <p>2.The sprite begins dancing when it receives the message.</p>	<p>1.The background appears when the game begins;</p> <p>2. The light effects on the stage starts changing when it receives the message.</p>

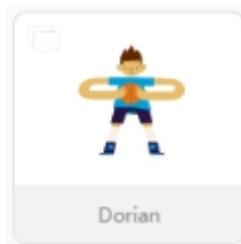
LEDs light up, a message will be sent to the stage.

Connect Codey Rocky to mBlock 5

1. Connect Codey Rocky via Bluetooth or a USB cable.

Add/Delete Stage Sprites

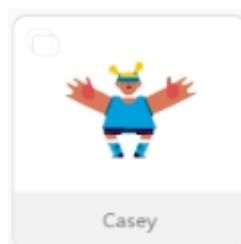
1. Add the sprite Dorian.
 - Adjust its size to make it fit the stage.



2. Add the sprite Jamie.
 - Adjust the size of the sprite.



3. Add the sprite Casey.
 - Adjust the size of the sprite.



4. Add the sprite Dot.
 - Adjust the size of the sprite.

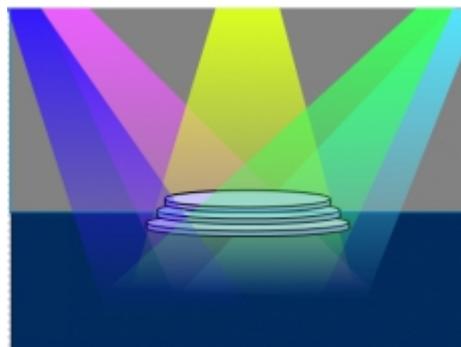


5. Delete the sprite Panda.



6. Set the backdrop: Spotlight-stage1

- Define the light effect. You can add spotlights if you need and you can change the color of each spotlight.



□ Write Programs

1. Device: Codey Rocky turns on the LED light and shows "welcome" when it starts up.
1. Device: When button A is pressed, Codey Rocky will use its light sensor and IR sensor to measure the light intensity and detect whether someone is nearby;
2. LED Strip: when the light intensity is lower than 25, the LED strip will light up;

```

forever
  show ambient light intensity
  if ambient light intensity < 25 and color sensor reflected light intensity > 5 then
    LED strip 1 lights up 30 / ambient light intensity with color red 150 green 150 blue 150
  
```

3. Device: when all the lights are turned on, a message "start dancing" will be sent to the stage;

4. Stage sprite -Dorian: The sprite Dorian will appear on the upper side of the stage. When the sprite receives the message "start dancing", it will start dancing, repeat the motion and slide to a specific position as programmed;

```

repeat (5)
  switch costume to dorian-a
  wait (0.2) seconds
  switch costume to dorian-b
  wait (0.2) seconds
  switch costume to dorian-c
  wait (0.2) seconds
  switch costume to dorian-d
point in direction (90)
glide (1) secs to x: (180) y: (-60)
  
```

5. Stage sprite -Jamie: Jamie will appear on the front right side of the stage. When Jamie receives the message "start dancing", it will start dancing and repeat the code 2 times. Then it will slide to the left side of the stage;

```

point in direction (90)
change x by (-10)
if on edge, bounce
  
```

6. Stage sprite-Casey: The sprite appears in the middle of the stage. When it receives the message "start dancing", it will say "let's dance!" and start dancing;

```

when I receive start dancing
say Let's dance! for (2) seconds
  
```

7. Stage sprite-Dot: The sprite appears on the front left side of the stage. When the sprite receives the message "start dancing", it will move to the right-hand side and bounce back when touching the edge;

```

if direction > 50 then
  change x by 10
  if on edge, bounce
  set rotation style left-right
if direction < -30 then
  change x by -10
  if on edge, bounce
  set rotation style left-right

```

8. Stage backdrop- Spotlight-stage1: When the stage receives the message, it will start changing the light effect;
9. Device: when button B is pressed, the LED strip, the indicator and the screen all turn black. And a message "stop" will be broadcast.

Referential Stage Design



Extension task

Codey Rocky and Neuron have many other sensors. You can use these sensors to make your game more appealing. For instance, you can add background music and change the light color effects along with the music.

The Light Sensor block is in the left upper part of Codey and the IR Sensor is located below Rocky. The darker the environment is, the lower the light intensity value will be; the closer the obstacle is, the higher the reflected infrared light intensity will be.

3. Independent Practice

- ① Have students work in pairs to work on the task;
- ② Ask students to raise their hands when the task is completed.

4. Presentation

5. Wrap up

In this lesson, we learned to program the Light Sensor and the IR Sensor to detect environmental changes. When the light intensity is below 4, a message "start dancing" will be broadcast to the stage, and then the sprites on the stage will start acting as programmed.