

## Variables

### Overview

<b>Concept</b>	<b>Variable</b> —— a container storing information that can be changed.
<b>Explanations</b>	Examples: Candy jar, Scoreboard, Wallet.
<b>Learning Objectives</b>	1. Understand the concept of <b>Variable</b> ; 2. Create a variable and use the variable in your programs.
<b>Teaching Preparation</b>	1. A box of nuts and lots of variables paper strips (step 3- Lead-in game); 2. A whiteboard and a whiteboard marker (or you can use a blackboard and chalks); 3. One Codey Rocky and a Bluetooth dongle (or a USB cable) per student but it's fine if 2 or 3 students share one set; 4. A computer per student, but it's fine if 2 or 3 students share a computer. And install mBlock 5 on each computer; 5. A self-review report form and project report form per student.
<b>Time</b>	120-240min

### Teaching Procedure

#### Step 1: Review——Functions

Review:

- How were functions used in a program in the previous lesson? (walking squarely and dancing)
- How did functions help make programs more efficient? (A function can be called multiple times, which makes programs more concise).
- What will the programs be like if we don't use functions? (The programs will be lengthy because some pieces of code might be repeated multiple times. And the logic of the programs might be unclear.)
- What's the difference between a function and a loop? (A loop is repeated in one certain place in a program while a function can be called over and over again in different places in a program.)

A function refers to a group of coding instructions that can be called over and over again. In the mBlock 5, we need to make a block under the category of **My Blocks**. Then, we can call the function block at any time.

## Step 2: Explain New Knowledge —— Variables

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. A scoreboard works in the same way, serving as a container that stores the match scores. Before the match ends, you can change the scores on the scoreboard when necessary. Specifically speaking, a variable is a storage location in a computer, and it is paired with a symbolic name. When a variable is created, the computer will leave a memory location to store the variable and give the location a symbolic name. Through the variable name, programs can read and modify the data 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 has a three-point shot, the number on the scoreboard will change accordingly. A new value takes the place of the old value.

How a variable is used can be explained from three aspects:

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 **Revising 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 **Comparison**. You can read the current value and compare it with another value. Through comparison, you can tell that when a variable is greater than a certain value, then something will happen. For instance, if the scores of the red team exceed the scores of the blue team, then the red team wins.

There are plenty of examples of variables in real life. The **Health** of characters in games is an example. Similarly, the health points of characters are changing over time in games. The amount of

money in your wallet and the number of candies are both examples of variables. We can read the values and change the amounts of money and candies by adding or subtracting them. Moreover, we can compare the values with other values. For instance, if the amount of money is greater than the price of a certain product, then we can afford to purchase the product; if the amount of candies in the jar exceeds a certain number, then we can share them with friends.

### Step 3: Lead-in Game —— Squirrel's Nuts Box

#### Story and Teaching Preparation:

A little squirrel has a nuts box. Every day, it puts nuts into the box and takes nuts out of the box, so the amount of nuts keeps changing all the time.

It's another day. In the morning, the squirrel gets up and opens its nuts box, finding out there are 10 nuts left.

*Teaching Preparation:* Teachers prepare a box that's shadow. Put a white paper on the box and writes down a number 10 on the paper. Open the box and show the number to students.

Let's have a look at what happens in the nuts box?

*Teaching Preparation:* Teachers prepare some paper strips. Each paper strip describes a situation of how the number of nuts changes (as shown below). 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. Following the instructions on the paper strips, students should count the current number of the nuts, erase the previous number and write down a new number with pencils.

In the morning, the little squirrel eats 2 nuts for breakfast.
If the maximum temperature today exceeds 5°C, the squirrel will pick 10 nuts outdoors.
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 little 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 with 7 nuts.

If the amount of the nuts in the box is less than 5, then the squirrel will pick 10 nuts outdoors.

If the amount of the nuts in the box exceeds 12, then the little squirrel will turn in a circle happily.

Ask students:

1. What's the variable in the game? What name will you give it?
2. What's the initial value of the variable?
3. What operations were applied in the game?
4. How did you compare the variable with other values in the game?

**Tips:**

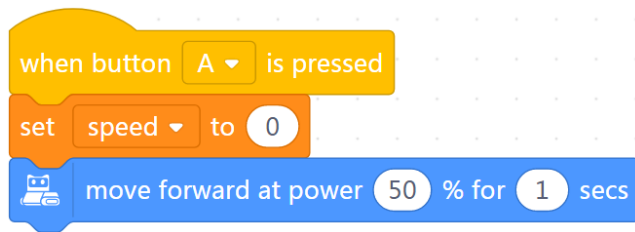
The order of how the paper strips are picked affects the calculated results. In some cases, students might get a negative number after they calculate the nuts. If students have already learned about the concept of a negative number, tell them to replace the previous value with a negative number; if they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

**Summary:** Imagine that a variable is a box and you can store a value that can be changed in the box. When you need to use the variable in a program, the box will open and the value will be taken out. Next, you can add, subtract, multiply or divide the value, or compare the value with another value. By doing this, you get a new value and can replace the original value with a new one.

**Step 4: Tasks**

Guided by teachers, students learn how to create, set and use a variable. **Tell them to tick the square box when they finish the step (The Tasks Cards are included at the end of this document. Print them).**

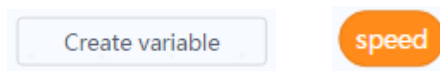
**Guided by Teachers:** Demonstrate to students how to write the following programs:



### Instruction:

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

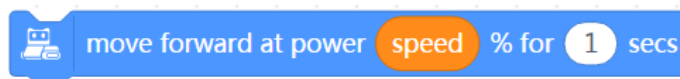
Name the variable:



2. Set a value:



3. Use the variable:



### Task 1: Assign a value to the variable

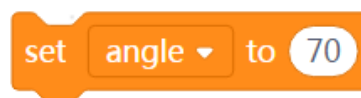
Have students work in pairs or work on their own to complete the tasks.

#### Task 1- Assign a value to the variable

##### Codey Rocky is standing on the stage to greet the audience.

- ☐ Codey Rocky turns left 70 degrees to face towards the left and then turns right 140 degrees to face towards the right. Finally, it turns left 70 degrees to return to the starting place. To make Codey Rocky turn specific degrees, we have to create a variable named “**angle**”.

- ☐ Set the value of the variable to be 70.



- ☐ Codey Rocky turns left by the preset degrees.



- ☐ Assign a new value to the variable “**angle**” to make Codey Rocky turn right 140 degrees and then turn left 70 degrees.

- ☐ Add expressions, sounds and lights to each rotation of Codey Rocky.

### Tips:

1. After we create a variable, new blocks will show up in the Scripts area:

Variable block: 

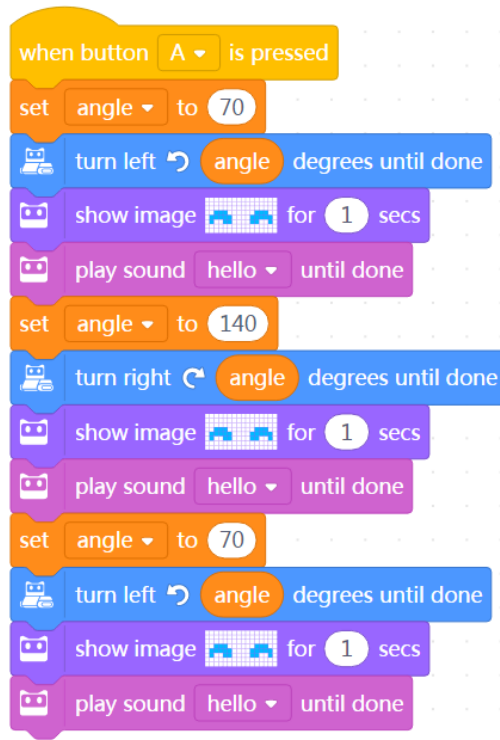
Block for setting the value: 

Block for changing 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 has three different settings and is used three times.

### Sample Program:



### Discussion:

1. 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?

### Task 2: Add and Subtract

#### Task 2-Add and Subtract

**Codey Rocky have learned the basics of operations and knew how to add 1 to a value and subtract 1 from a value.**

- ☐ Codey Rocky would like to start with the number 0, so we can set the value of the variable “**number**” to be 0.



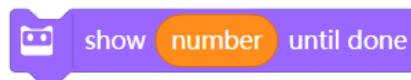
- ☐ When the button A is pressed, the number increases by 1.



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



- ☐ If you would like to show each calculated result on the LED display, you can add the block “**show ( ) until done**” under each event block.



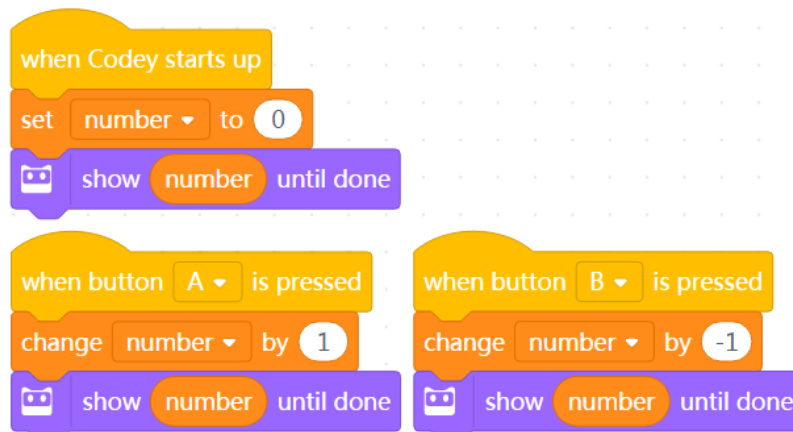
Tips:

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 Programs:**

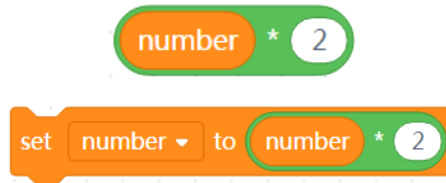


### Task 3: Multiply and Divide

#### Task 3-Multiply and Divide

Codey Rocky also learned how to multiply and divide.

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

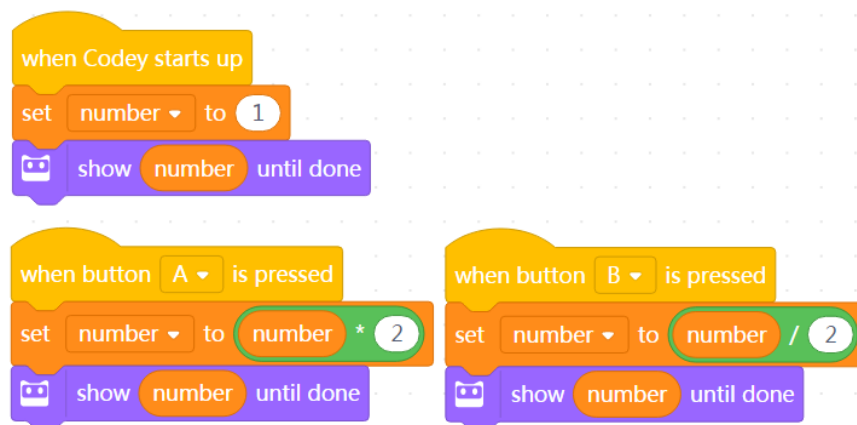


- ☐ When the button B is pressed, the number is divided by 2.
- ☐ Show each calculated result on the LED display. It is worth pointing out that the LED display can only show a value ranging from -999 to 9999.
- ☐ **Challenge:** Change an initial number and a factor (replace the number 2 with another value). Observe the result.

#### Tips:

1. We need to use the Operators blocks to get calculated results. We replace the previous value with a new value by reassigning the calculated 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 Programs:





### Discussion:

1. If we have a closer look at the values on the LED display, we will notice one thing: if there is a time that the calculated result includes a decimal, then the following calculated results will continue to show up in the form of decimals.

### Task 4: Speed up

#### Task 4-Speed up

**Classes are over. Codey Rocky is speeding up and rushing home at its top speed.**

- ☐ The initial speed of Codey Rocky is set to be 0, which means that Codey Rocky stays there not moving.
- ☐ When the button A is pressed, the speed will keep rising by 20 at a time. Repeating 5 times, the speed will finally go up to 100.



- ☐ If the speed exceeds 70, the RGB LED will light on with color red to give a warning. In this case, we need to use the **comparison operator block** to tell whether the **conditional statement** is true or not.

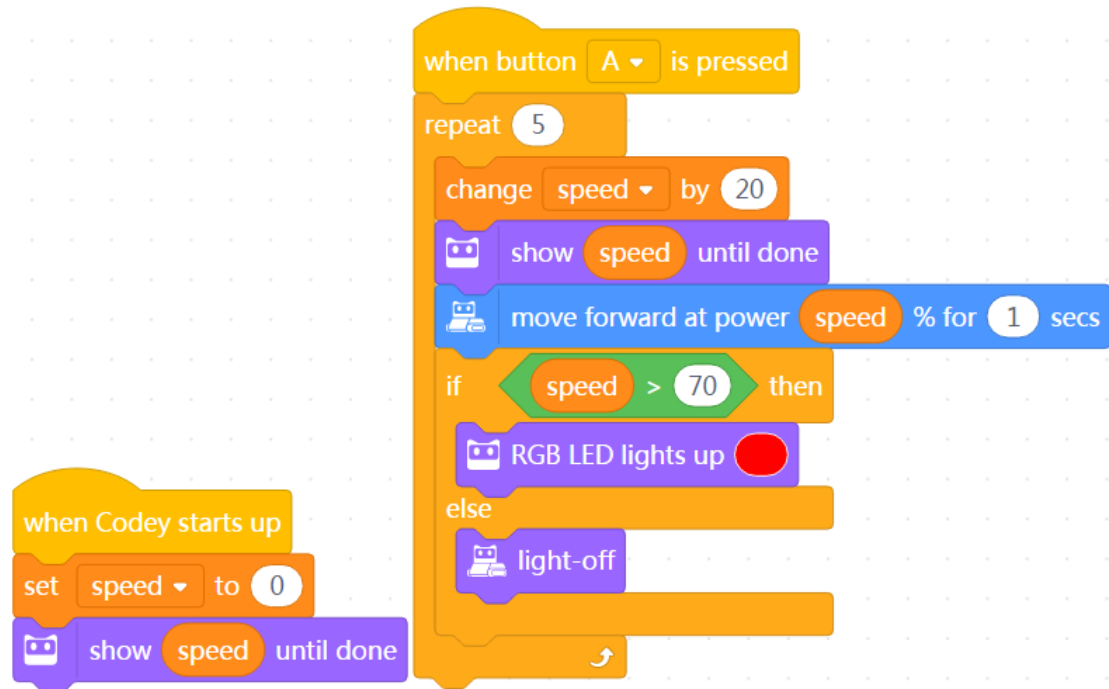


- ☐ Remember to display the value on the LED screen in real time.
- ☐ **Challenge:** Set the speed to be 100 and then keep slowing the speed until making Codey Rocky stop.

### Tips:

1. The top power of Codey Rocky is 100% . If a power value exceeds 100%, Codey Rocky will continue to move forward at power 100%.
2. The minimum power of Codey Rocky is 0%. If a power value is a negative number, then Codey Rocky will move in an opposite direction. In other words, it will move backward.
3. Make the LED display show the value promptly the moment the variable changes.

### Sample Programs:



### Discussion:

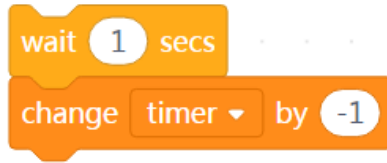
1. Set the initial speed of Codey Rocky to be 100 and keep lowering the speed until it stops. How to revise the programs?
2. Set the power to be a negative number. Observe how Codey Rocky will move?

### Task 5: Bomb Countdown

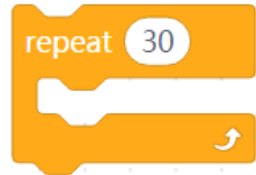
#### Task 5- Bomb Countdown

**Rules:** 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 blasts.

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



The same step repeats 30 times.



Add sounds to make the countdown more intense and display how much time is left.



- ☐ 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 “timer” becomes 0, the countdown ends and the bomb blasts. At this time, the RGB LED light on with the color red.
- ☐ **Challenge:** Reset the initial value of the variable “**timer**” to make the game time longer or shorter.

### Sample Programs:



## Task 6: The Explosive Number

### Task 6-The Explosive Number

**Rules:** When the button is pressed, Codey will generate a number randomly. Have two students play the game rock-paper-scissors. Anyone who loses the game has to press the button. And the number on the LED display will decrease by 1. Repeat the steps until the display shows the number that Codey randomly generates. Boom! The bomb blasts.

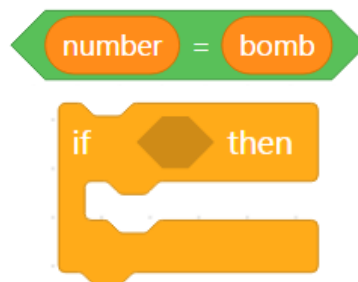
- ☐ We need to create two variables: **bomb** and **number**. The variable “**bomb**” represents the randomly generated number. The variable “**number**” indicates the changing numbers (starting from 0). The bomb blasts when the two variables have the same values.
- ☐ When Codey Rocky starts up, the 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 that falls within the range of 1-20.



Meanwhile, Codey puts on his sunglasses and plays the sound “**ready**”.

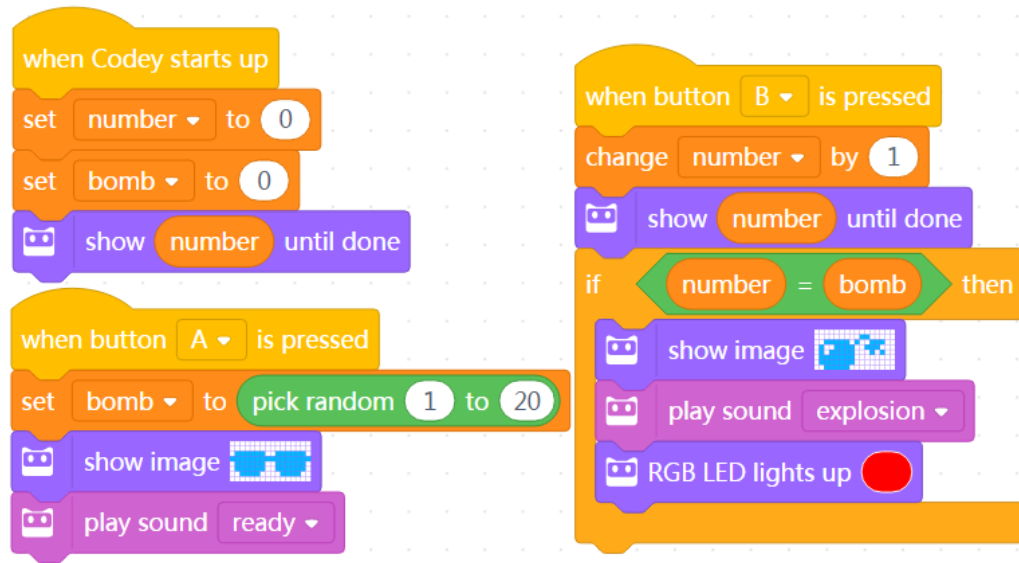
Two students play the game rock-paper-scissors. Anyone who loses the game has to press the button B. When the button is pressed, the value on the LED display increases by 1 and the new value appears on the LED display.

- ☐ If the value of the variable “**number**” is the same one as the “**bomb**” value, an image bomb will appear on the LED display and then comes the sound. You will have to use the operator blocks of comparison and conditional blocks to assess whether the two variables are identical in values.



- ☐ **Challenge:** Revise the value range of the **bomb**.

### Sample Programs:



### Task 7: Rock-Paper-Scissors

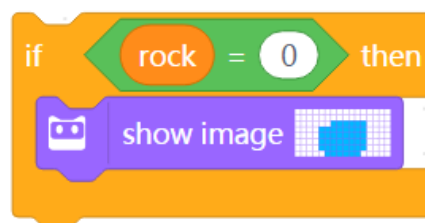
#### Task 7-Rock-Paper-Scissors

Use Codey Rocky to play the game rock-paper-scissors with friends.

- ☐ When Codey Rocky starts up, the initial values of all the variables are set to be 0.
- ☐ Codey Rocky forms one of the shapes randomly. It uses the number 0, 1 and 2 to represent rock, paper and scissors respectively. When Codey is shaken, the value of the variable “shape” will be set to one of the three numbers randomly.



If the value of the variable “shape” is set to be 0, then an image of a **fist** will appear on the LED display.



If the value of the variable “shape” is set to be 1, then an image of **scissors** will appear on the LED display.

If the value of the variable “shape” is set to be 2, then an image of **paper** will appear on the LED display.

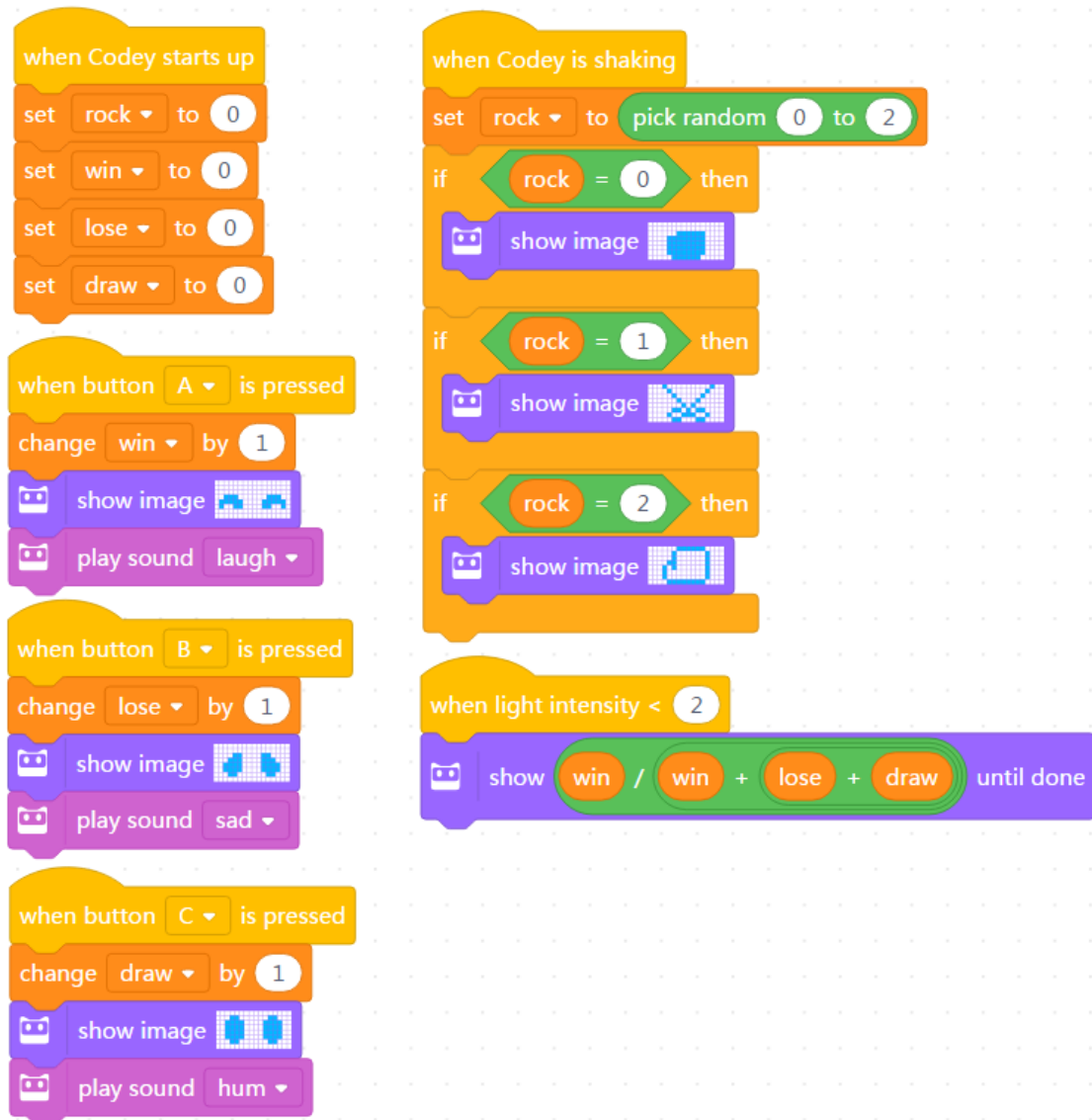
- If Codey wins the game, then press the button A. The variable “**win**” will increase its value by 1. Meanwhile, Codey shows a smiley face and laughs out.
- If Codey loses the game, then press the button B. The variable “**lose**” will increase its value by 1. Meanwhile, Codey shows a sad face and cries.
- If it is a draw, then press the button C. The variable “draw” will increase its value by 1. Meanwhile, Codey looks the same and hums.
- In some cases, Codey might peep at its winning probability. When the light intensity exceeds 2, the winning probability will appear on the LED display in the form of decimals.



#### Tips:

1. When we write programs for Codey Rocky, we can use six events at most.

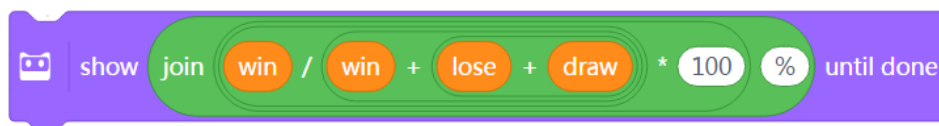
#### Sample Programs:



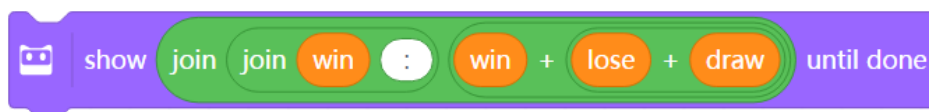
#### Discussion:

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 answer:**



#### Task 8: Infrared Battle

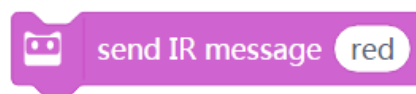
### Task 8- Infrared Battle

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

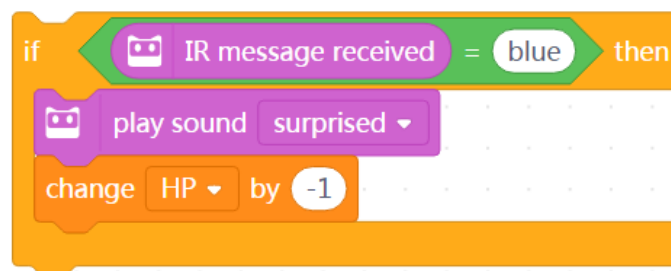
- The two ears of the Codey are preset with an IR receiver and an IR transmitter, which facilitates the wireless communication between two Codeys.



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



- When one Codey receives the message from another Codey, then we consider the first Codey gets shot and its health points reduce by 1. We use the conditional block here to assess whether the Codey receives an IR message (gets shot). If Codey gets shot, it will make a scary sound and the health points will fall by 1.



- When Codey starts up, the initial value of the health point should be set to 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 “scary”. The game is over when the health points fall to 0 and Codey looks so sad.



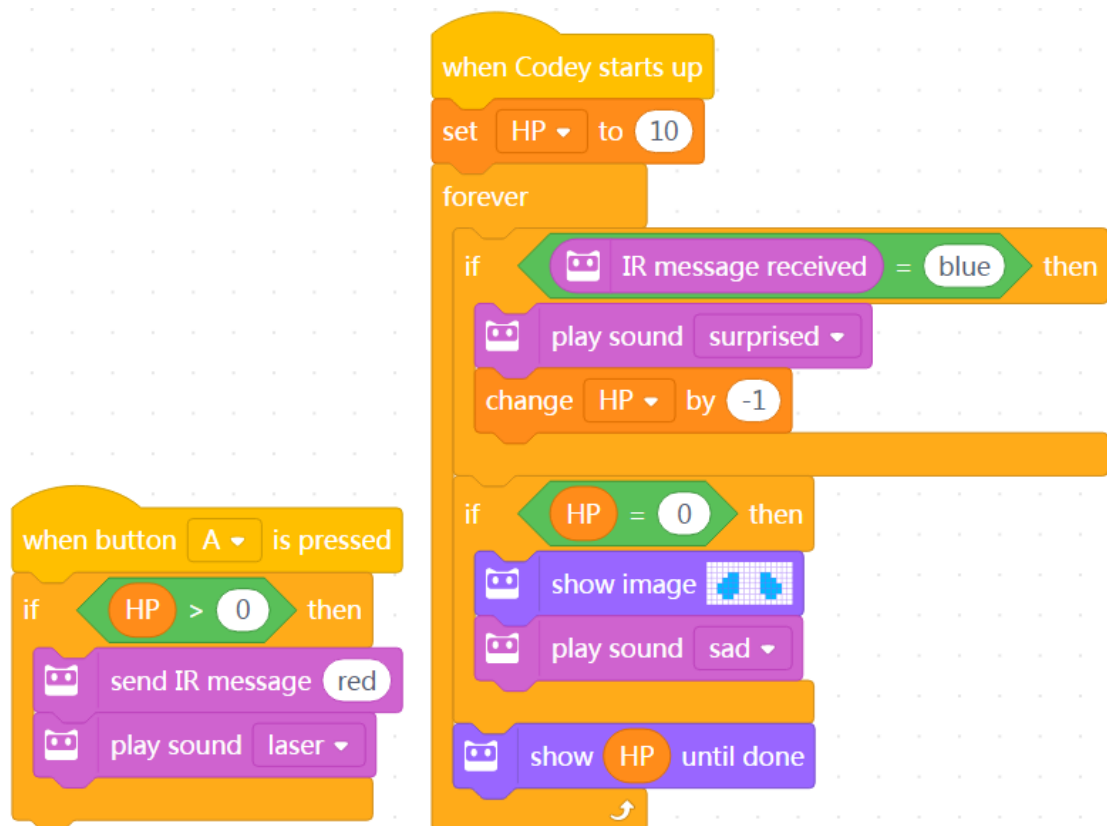
☐ When the button A is pressed, if Codey still has health points (the value is greater than 0), then it can send IR messages and play the sound “laser”.

☐ **Challenge:** If it is a battle of three, then how to revise the programs?

**Tips:**

1. When we write programs for Codey Rocky, we can use six events at most.

**Sample Programs:**



**Discussion:**

1. If there are three teams, then how should we revise the programs?

### Self-review Report

Name:

Age:

- Answer the following questions to record your learning outcomes:

Describe what you've learned with one or two sentences.

Describe what you like most and least about this class session briefly.

What I like most

What I like least

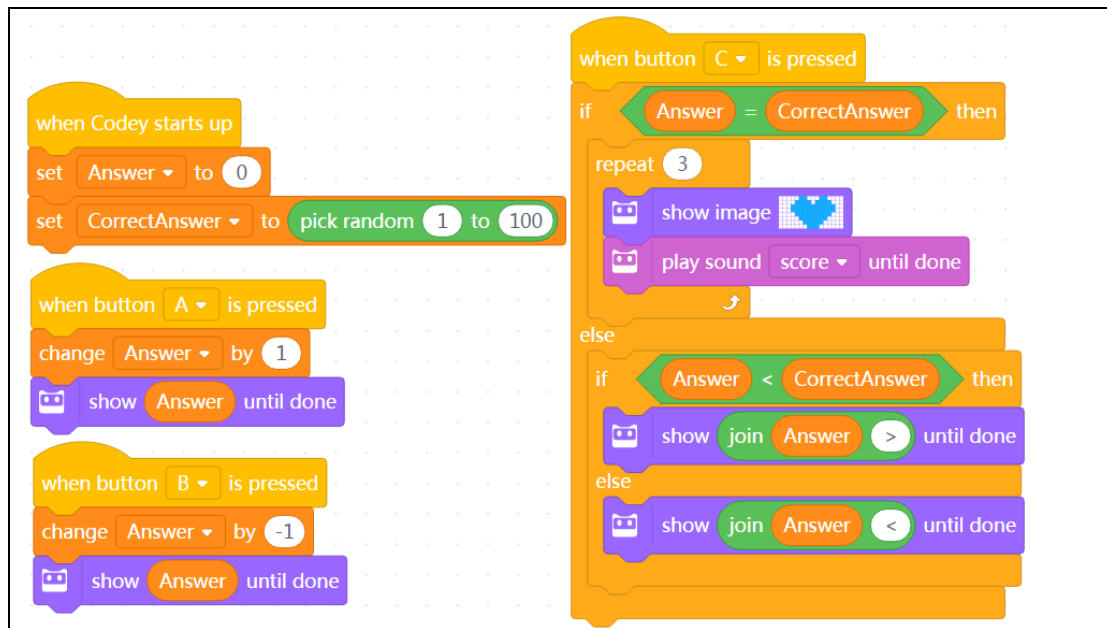
Use one or two sentences to describe variables.

Why do we say a variable is a container?

What does the variable store as a container in this lesson?

How do you use variables in this lesson?

Guess what a game it is? (Answer: \_\_\_\_\_)



You can draw out how you feel about this class session at the upper right corner of the self-review report.

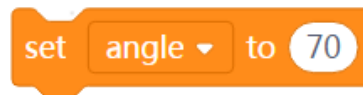
In the morning, the little squirrel eats 2 nuts for breakfast.
If the maximum temperature today exceeds 5°C, the squirrel will pick 10 nuts outdoors.
At noon, the squirrel eats 3 nuts for lunch.
In the afternoon, a little bird pays a visit and gives the squirrel 5 nuts.
Today is the monkey's birthday. The little 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 with 7 nuts.
If the amount of the nuts in the box is less than 5, then the squirrel will pick 10 nuts outdoors.
If the amount of the nuts in the box exceeds 12, then the little squirrel will turn in a circle happily.

### Task 1- Assign a value to the variable

#### Codey Rocky is standing on the stage to greet the audience.

☐ Codey Rocky turns left 70 degrees to face towards the left and then turns right 140 degrees to face towards the right. Finally, it turns left 70 degrees to return to the starting place. To make Codey Rocky turn specific degrees, we have to create a variable named “**angle**”.

☐ Set the value of the variable to be 70.



☐ Codey Rocky turns left by the preset degrees.



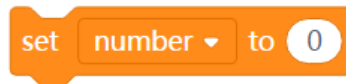
☐ Assign a new value to the variable “**angle**” to make Codey Rocky turn right 140 degrees and then turn left 70 degrees.

☐ Add expressions, sounds and lights to each rotation of Codey Rocky.

### Task 2-Add and Subtract

**Codey Rocky have learned the basics of operations and knew how to add 1 to a value and subtract 1 from a value.**

- ☐ Codey Rocky would like to start with the number 0, so we can set the value of the variable “**number**” to be 0.



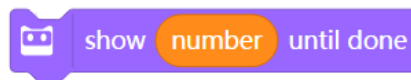
- ☐ When the button A is pressed, the number increases by 1.



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



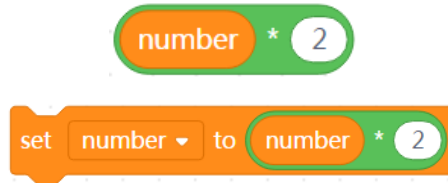
- ☐ If you would like to show each calculated result on the LED display, you can add the block “**show ( ) until done**” under each event block.



### Task 3-Multiply and Divide

**Codey Rocky also learned how to multiply and divide.**

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



- ☐ When the button B is pressed, the number is divided by 2.
- ☐ Show each calculated result on the LED display. It is worth pointing out that the LED display can only show a value ranging from -999 to 9999.
- ☐ **Challenge:** Change an initial number and a factor (replace the number 2 with another value). Observe the result.

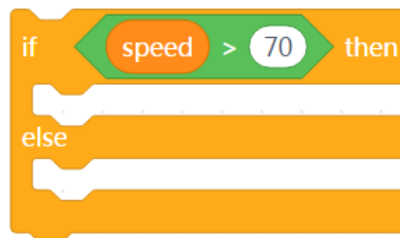
#### Task 4-Speed up

**Classes are over. Codey Rocky is speeding up and rushing home at its top speed.**

- ☐ The initial speed of Codey Rocky is set to be 0, which means that Codey Rocky stays there not moving.
- ☐ When the button A is pressed, the speed will keep rising by 20 at a time. Repeating 5 times, the speed will finally go up to 100.



- ☐ If the speed exceeds 70, the RGB LED will light on with color red to give a warning. In this case, we need to use the **comparison operator block** to tell whether the **conditional statement** is true or not.

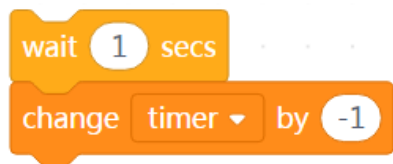


- ☐ Remember to display the value on the LED screen in real time.
- ☐ **Challenge:** Set the speed to be 100 and then keep slowing the speed until making Codey Rocky stop.

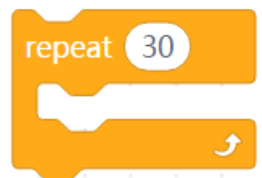
### Task 5- Bomb Countdown

**Rules:** 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 blasts.

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



The same step repeats 30 times.



Add sounds to make the countdown more intense and display how much time is left.



- ☐ 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 “timer” becomes 0, the countdown ends and the bomb blasts. At this time, the RGB LED light on with the color red.
- ☐ **Challenge:** Reset the initial value of the variable “**timer**” to make the game time longer or shorter.



### Task 6-The Explosive Number

**Rules:** When the button is pressed, Codey will generate a number randomly. Have two students play the game rock-paper-scissors. Anyone who loses the game has to press the button. And the number on the LED display will decrease by 1. Repeat the steps until the display shows the number that Codey randomly generates. Boom! The bomb blasts.

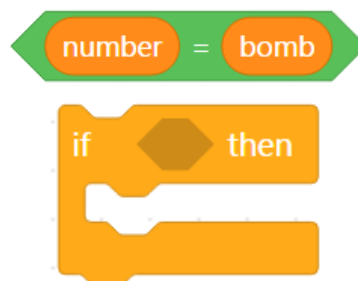
- ☐ We need to create two variables: **bomb** and **number**. The variable “**bomb**” represents the randomly generated number. The variable “**number**” indicates the changing numbers (starting from 0). The bomb blasts when the two variables have the same values.
- ☐ When Codey Rocky starts up, the 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 that falls within the range of 1-20.



Meanwhile, Codey puts on his sunglasses and plays the sound “**ready**”.

Two students play the game rock-paper-scissors. Anyone who loses the game has to press the button B. When the button is pressed, the value on the LED display increases by 1 and the new value appears on the LED display.

- ☐ If the value of the variable “**number**” is the same one as the “**bomb**” value, an image bomb will appear on the LED display and then comes the sound. You will have to use the operator blocks of comparison and conditional blocks to assess whether the two variables are identical in values.



- ☐ **Challenge:** Revise the value range of the **bomb**.

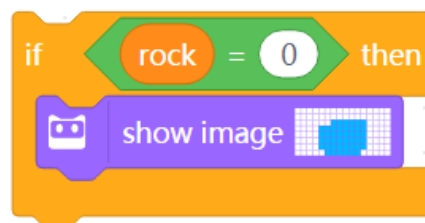
## Task 7-Rock-Paper-Scissors

Use Codey Rocky to play the game rock-paper-scissors with friends.

- ☐ When Codey Rocky starts up, the initial values of all the variables are set to be 0.
- ☐ Codey Rocky forms one of the shapes randomly. It uses the number 0, 1 and 2 to represent rock, paper and scissors respectively. When Codey is shaken, the value of the variable “shape” will be set to one of the three numbers randomly.



If the value of the variable “shape” is set to be 0, then an image of a **fist** will appear on the LED display.



If the value of the variable “shape” is set to be 1, then an image of **scissors** will appear on the LED display.

If the value of the variable “shape” is set to be 2, then an image of **paper** will appear on the LED display.

- ☐ If Codey wins the game, then press the button A. The variable “**win**” will increase its value by 1. Meanwhile, Codey shows a smiley face and laughs out.
- ☐ If Codey loses the game, then press the button B. The variable “**lose**” will increase its value by 1. Meanwhile, Codey shows a sad face and cries.
- ☐ If it is a draw, then press the button C. The variable “draw” will increase its value by 1. Meanwhile, Codey looks the same and hums.
- ☐ In some cases, Codey might peep at its winning probability. When the light intensity exceeds 2, the winning probability will appear on the LED display in the form of decimals.



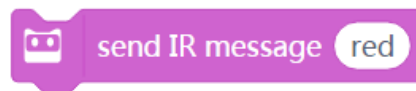
### Task 8- Infrared Battle

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

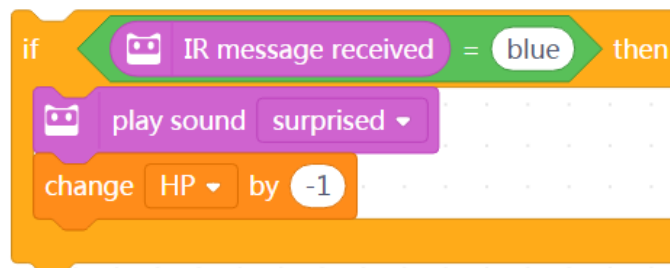
- The two ears of the Codey are preset with an IR receiver and an IR transmitter, which facilitates the wireless communication between two Codeys.



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



- When one Codey receives the message from another Codey, then we consider the first Codey gets shot and its health points reduce by 1. We use the conditional block here to assess whether the Codey receives an IR message (gets shot). If Codey gets shot, it will make a scary sound and the health points will fall by 1.



- When Codey starts up, the initial value of the health point should be set to 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 “scary”. 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 (the value is greater than 0), then it can send IR messages and play the sound “laser”.
- **Challenge: If it is a battle of three, then how to revise the programs?**