

## Sequencing

### Overview

<b>Concept</b>	<b>Sequencing</b> —— A set of ordered steps for accomplishing a task
<b>Explanation</b>	To ensure that Codey Rocky can fulfill the task as expected, we should give instructions in the correct order.  Example: Put the watermelon in the refrigerator.  <b>Bug:</b> Mistakes that cause the failure of programs to run as programmed  <b>Debug:</b> Find bugs and fix them.
<b>Learning Objectives</b>	1. Understand the concept of Sequencing;  2. Design a set of steps for making an animation;  3. Understand the concept of Bug;  4. Find bugs and fix them.
<b>Teaching Preparation</b>	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) for each 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;  4. Hand out a self-review report and a project report for each student.
<b>Time</b>	90-120 min

### Teaching Procedure

#### Step 1: Review —— Event

Review:

- What is Event?
- Can students think of any events in daily lives?
- What events did you use in the last class?

**Event** is an action that causes things to happen. For example, when you press the button, the light is turned on. In this case, pressing the button is the event and the result is that the light is turned on. The events students used in the last class session are: 1) When the program starts up 2) When Button A, B and C pressed.

## **Step 2: Explain New Knowledge —Sequencing**

The new concept for this class session is Sequencing. For instance, if we want to put the watermelon in the refrigerator, then we need to follow these steps:

1. Open the refrigerator door.
2. Put the watermelon in the refrigerator.
3. Close the refrigerator door.

If you follow the wrong sequence, you wouldn't be able to put the watermelon in the refrigerator.

### **Brainstorming**

Ask students if they can think of any cases in which they must follow a set of steps to achieve something. Teachers can give an example here:

*You must first uncap the bottle, pour the water into your mouth and then screw the cap of the bottle.*

If you don't follow the steps, you won't be able to drink the water. (Tip: Each example should only offer one specific order. It means that only when you follow one specific set of steps, can things be done in the example.)

To fulfill a task, you must follow a set of steps. The order the steps are performed is called as **Sequencing**.

## **Step 3: Lead-in Game - I'm a Robot**

The teacher acts like a robot, walking from somewhere in the classroom to the blackboard and drawing a smiley face on it. Students give instructions to the robot and write them on papers.

### **Game Instructions:**

1. Teachers read the students' instructions from top to bottom.
2. Ask students to leave detailed instructions in the correct order.

### **Tips:**

1. If students write instructions from left to right, teachers should still stick to 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, teachers 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, teachers 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.

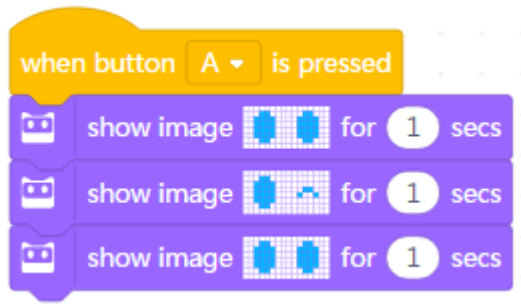
**Summary:** When we are programming, we arrange the blocks in the order from top to bottom to form a set of steps. Machines can follow the steps one by one to perform a task. We refer to the set of steps as **Sequencing**.

#### **Step 4: Student Activity - Animation Design**

Design animations with the block **【show image ( ) for ( ) secs】**. Traditionally, animation designers would make animations in the following way: Put a sheet of static drawing on the table first and unfold a new drawing paper on top of the first paper. Designers will outline the frame and then change the drawing slightly. Then another new 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 make them smooth. Simply speaking, designers animate the drawings.

##### **Task 1: Follow the teachers to design animations**

**Demonstrate:** In this session, teachers will show how to make animations with the block **【show image ( ) for ( ) secs】**. The method is simple: Use the image of the block as the starting picture, 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 first image to be Codey opening eyes;

Duplicate the block **【show image ( ) for ( ) secs】** but edit one eye to be winking;

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.

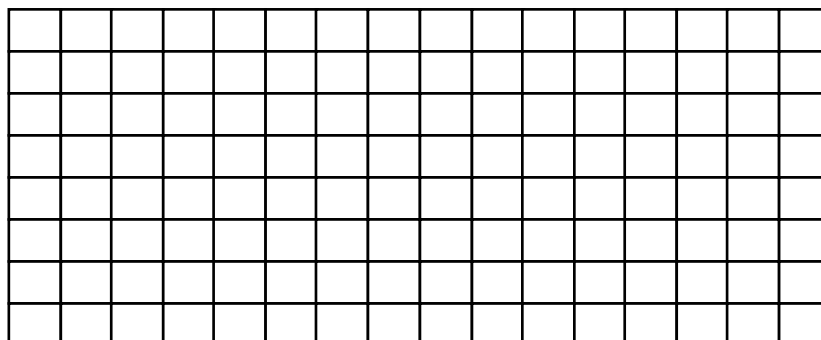
**Student Activity:**

Students work in pairs to make their animations as the teachers did.

**Task 2: Design New Animations**

*At the end of this session, students need to showcase their works and fill in the project reports.*

Students work in pairs and fill the little boxes to form an image first; draw the same image on the block **【show image ( ) for ( ) secs】** ; duplicate the block and change the image slightly; repeat the step until you have a series of blocks that are changing and arranged in sequence.

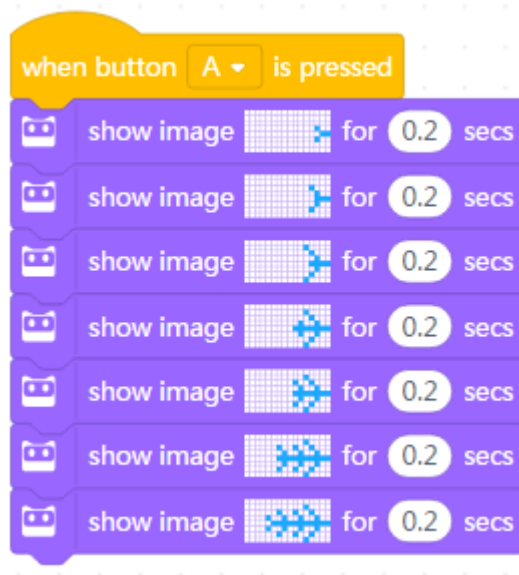


Upload codes to examine the animation effect. Students need to present their works but should complete the project report first. They should display their works by following the questions from the report.

### Tips

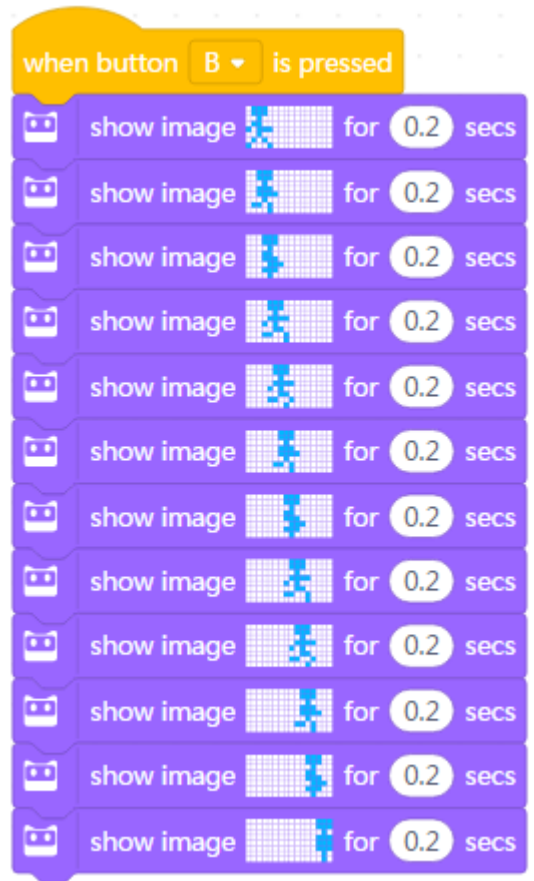
1. If some students accomplish the task ahead of time, tell them to start their animations with multiple events;
2. Or have them change how long they show the image;
3. If some students fail to accomplish the task on time, invite them to share one thing that happened when they were writing programs. It could be something funny, a challenge they confront 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 min

Teachers can show sample projects: *Growing Tree* and *Walking*



### Program Story:

Place the LED matrix display upright. When the button B is pressed, a sapling will be growing.



Program Story:

When the button B is pressed, a child will be running from one side of the LED matrix display to the other side.

**Expert Tips: Bug and Debug**

Introduce the new concepts Bug and Debug before students work on their tasks.

When we are writing programs, bugs are inevitable. A bug means an insect in its literal sense but refers to a glitch in a computer program here. At the initial stage where computers were invented, unlike the portable computers of today, they were extremely large in size back then. (Picture from Wiki).

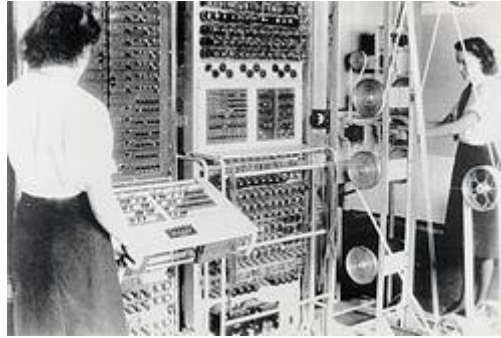


Figure 1 Colossus computers were used to decipher German codes during WW II

It was during this period that a colossus computer failed to work properly. The whole team of programmers was trying so hard to find out 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 log book (see the picture below). From now on, the term bug is common in use when people refer to the mistakes in a computer program. Naturally, Grace Murray Hopper was since then considered as the Mother of Debug. (Picture from Wiki)

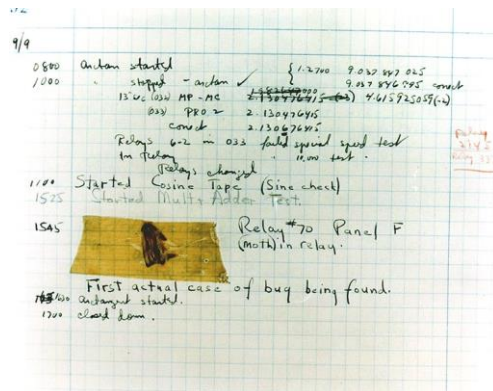
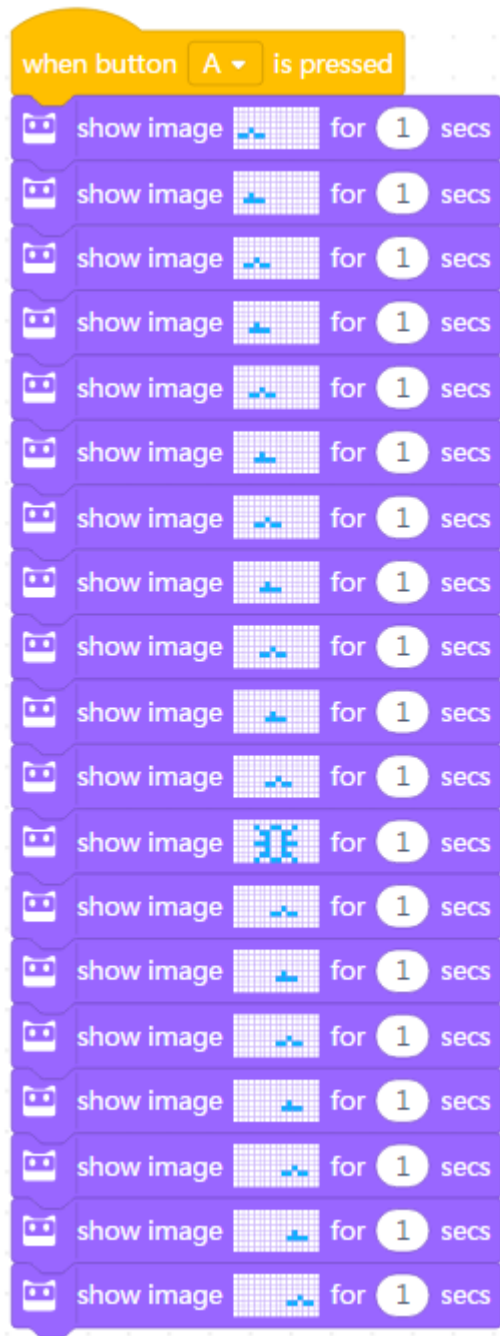


Figure 2 The moth caused the computer mistake and this is the first mistake in a computer program

When writing programs, we need to go through each line of the codes to find bugs and fix them.

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

### 1. Earthworm and Insect

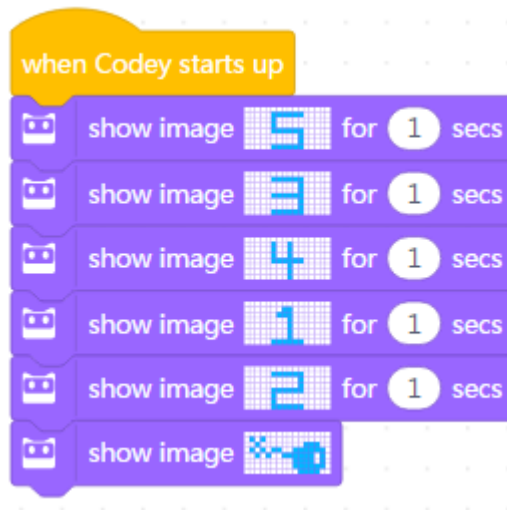


Program Story:

A little earthworm meets a big insect when crawling on the ground. Remove the insect to let the earthworm crawl again (change the insect image to be an image of the earthworm crawling).

**2. The Bomb Can't Count Down!!**





Program Story:

There is a bomb that can't time. Help it!

**3. The car key was stolen!**



Program Story:

You can't start the car because the car key was stolen. You have to find the key right now! (a missing event).

**Step 5: Wrap up**

1. Summarize what the concept of **Sequencing** is. **Sequencing** refers to a set of ordered steps for performing a task.

2. In the mBlock software, the sequencing is **from top to bottom**. If Codey Rocky fails to perform the task as programmed, you can go over each line of codes from top to bottom to find bugs.

3. When you design animations, you can start with a sheet of basic picture, copy the picture and change it slightly every single time. Finally, play the series of pictures in sequence to animate them.

4. Students complete the self-review report.

### Self-review Report

Name:

Age:



- Answer the following questions and record your outcome:

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

Draw a **Sequencing** that happened around you today:

You can paint how you feel about this class session in the upper right corner of the self-review report.

## Project Report

**Name:**

**Group Name:**

- Follow these questions to present your work:

What tasks did you take on? Describe it in one or two sentences.

Did you have any ideas on how to fulfill the tasks?

Write down or draw every single bit of inspiration, being it good or bad. You are doing this to explore more possibilities. Use more papers to record your ideas as you like.

Describe the final effect of your project and why you choose the effect in one or two sentences

Effect

*Example: When the button A is pressed, the earthworm starts crawling ahead.*

Why you choose this (these) effect?

*Example: Because it's funny.*

Did you come across any obstacles? Did you have any solutions? Describe it with one or two sentences. (The solution can be a rough plan)

Difficulties	Solutions
<p><i>Example: Failed to connect Codey Rocky to mBlock; Couldn't upload codes; Failed to achieve consensus...</i></p>	<p><i>Example: Power on Codey Rocky; Rock Paper Scissors...</i></p>

Students can answer the following questions after the sharing session.

Do you like your design? Describe what you like most and least about the project with one or two sentences. And any improvements in the future?

<input type="radio"/> Love it <input type="radio"/> Like it <input type="radio"/> So-so <input type="radio"/> Don't like it <input type="radio"/> Hate it
<p>What I like most</p> <p><i>Example: The animation is vivid; the sound matches the animation.</i></p>
<p>What I like least</p> <p><i>Example: It's challenging for people to tell what the animation is about at first glance.</i></p>
<p>Improve</p> <p><i>Example: Redesign the animation and make it simpler this time; add sounds and lights.</i></p>

### **Instructor Assessment**

1. Cooperation (30%): Evaluate how the group performs regarding labor division, collaboration, and coordination.
2. Completeness (20%) : Evaluate whether the project is complete enough. Of course, the project must stick to the topic first.
3. Innovation (20%) : Evaluate how creative the project is.
4. Functionality (20%): Evaluate whether the work is functional enough?
5. Difficulty (10%) : Evaluate what is the difficulty level of the work?