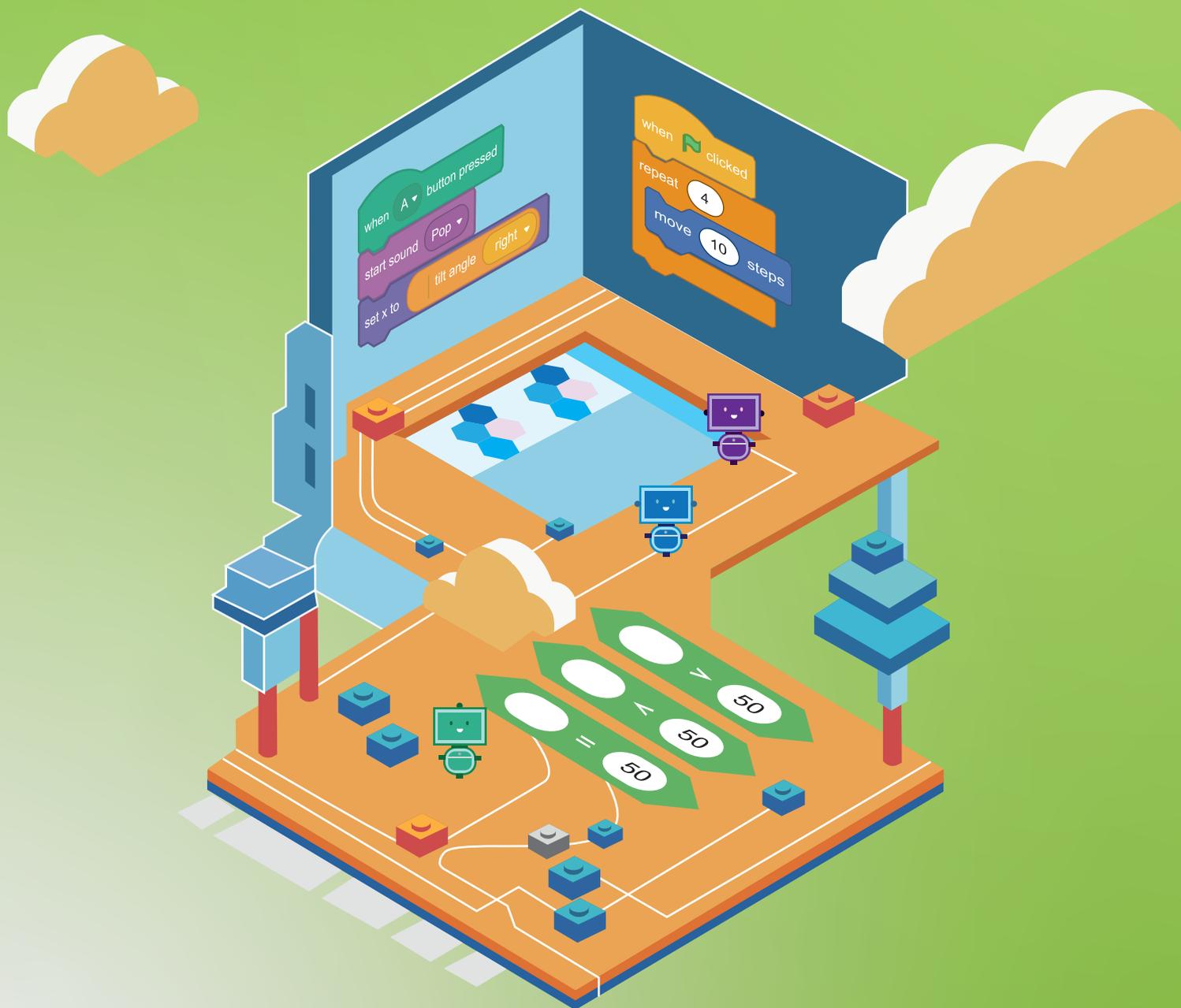


# Enriched Module on Coding Education for Upper Primary Level



Technology Education Section  
Curriculum Support Division  
Education Bureau  
June 2023

Primary  
**Four 4**  
Teacher Version  
Booklet 1

## Preface

The Education Bureau actively promotes innovation and technology (I&T) education for all students. Continuous incorporation of I&T learning elements into both the primary and secondary curricula helps strengthen the cultivation of students' interest in and capability of learning information technology and I&T from an early age, equip students with 21st century skills, and unleash their creativity and potential.

To enhance I&T education, the Education Bureau has launched the “Enriched Module on Coding Education for Upper Primary Level” for schools to adopt. Designed in accordance with the revised “Computational Thinking - Coding Education: Supplement to the Primary Curriculum” published in 2020, the curriculum module helps teachers integrate I&T elements into classroom learning more systematically. Schools should conduct appropriate curriculum planning with reference to the content of the “Enriched Module on Coding Education for Upper Primary Level”, and incorporate 10 to 14 hours of enriched coding education for all upper primary students every year in order to further develop their computational thinking and strengthen their I&T learning.

The “Enriched Module on Coding Education for Upper Primary Level” is adapted from learning and teaching resources of the “CoolThink@JC” project initiated and funded by The Hong Kong Jockey Club Charities Trust and co-created by The Education University of Hong Kong, Massachusetts Institute of Technology, and City University of Hong Kong. The Education Bureau is grateful for the collaboration with The Hong Kong Jockey Club Charities Trust in consolidating and drawing on the experience accumulated by the schools in the project to develop the “Enriched Module on Coding Education for Upper Primary Level” for adoption by all publicly-funded schools in Hong Kong. The Technology Education Section, Curriculum Support Division of the Education Bureau and Department of Mathematics and Information Technology of The Education University of Hong Kong co-developed the curriculum module based on the deliverables produced and experience gained in the project. Views on the content of the curriculum module were collected from the Committee on Technology Education of Curriculum Development Council and their support was sought.

The “Enriched Module on Coding Education for Upper Primary Level” covers basic coding and computational thinking concepts, namely abstraction, algorithm and automation, as well as connection with physical objects, the use of sensors and actuators to interact with the environment, etc., allowing students to develop their computational thinking as well as interest in and ability to learn I&T through the learning of coding.

This Primary 4 curriculum module, the first of three to be developed for upper primary levels (Primary 5 and 6 forthcoming), focuses on establishing a solid foundation for students' in the above basic concepts of coding and computational thinking; through

coding activities, logical thinking and problem solving skills are developed, and computational thinking is cultivated. There are a total of 8 units in the curriculum module, including 6 core units, and 2 optional extension units for schools to provide opportunities for students with a higher ability or strong interest in coding to enrich their learning and deepen their understanding of coding and innovative technology. The curriculum module also includes a project-based component that allows students to apply their computational thinking and creativity, and make good use of programming and innovative technology in different contexts, thereby formulating solutions to everyday problems for the benefit of society.

The recommended lesson time of the curriculum module (excluding the extension units) for each upper primary year level is 14 hours. Please refer to Table 1 and the Appendix for the arrangement of this Primary 4 curriculum module, the recommended lesson time, as well as the pedagogy to be adopted.

**Table 1: Arrangement of the Primary 4 curriculum module and recommended lesson time**

Unit	Unit Title	Core Unit		Extension Unit	
		Recommended Lesson Time (in minutes)	No. of Lessons (35 minutes for each lesson)	Recommended Lesson Time (in minutes)	No. of Lessons (35 minutes for each lesson)
1	Introducing Scratch Programming	70	2		
2	Exploring Under the Sea	70	2		
3	Storytelling	70	2		
4	Space Traveling	105	3		
5	Creating a Maze Game	140	4		
6	Creating a Maze Game with micro:bit			70	2
7	Drawing Shapes in Scratch	105	3		
8	Designing Line Pattern Art			70	2
	Final Project	280	8		
		840 (14 hours)	24	140	4

Views and suggestions on the “Enriched Module on Coding Education for Upper Primary Level” are always welcome. These may be sent to:

Chief Curriculum Development Officer (Technology Education)  
Curriculum Support Division  
Education Bureau  
Room W101, 1/F, West Block  
Kowloon Tong Education Services Centre  
19 Suffolk Road, Kowloon Tong  
Kowloon, Hong Kong

Fax: 2768 8664

E-mail: [teched@edb.gov.hk](mailto:teched@edb.gov.hk)

### Pedagogy

Teachers may make reference to the seven-step guide introduced in the Technological Pedagogical Content Knowledge (TPACK) framework for the teaching of computational thinking (CT). Technological content knowledge (TCK) refers to the knowledge of using block-based programming environments for coding. Content knowledge (CK) refers to the knowledge of CT concepts, practices, and attitudes to be taught. Pedagogical content knowledge (PCK) refers to pedagogies that do not involve the use of programming environments for teaching CK. TPACK refers to the integration of the use of technology and pedagogy to teach CK in context.

Based on the four dimensions of the TPACK framework above, teachers may adopt the seven-step guide in the instruction of each unit with a view to developing students' problem solving skills and digital creativity. The last three steps emphasise applying TCK to exploring the possible use of tools in the programming environments for the cultivation of digital creativity; revisiting and reviewing CK for consolidation; and reflection on PCK to engage in the improvement of teaching practices relevant to CK (Kong, Lai & Sun, 2020; Kong & Lai, 2022; Kong, Lai & Li, 2023).

- Step 1: TCK (Introducing features of the programming environment in a specific context)
- Step 2: CK (Introducing computational thinking concepts, practices and attitudes to be taught)
- Step 3: PCK (Adopting pedagogy such as allowing pre-coding access to games or apps to pave the way for reflection on the design of games or apps; and engaging in unplugged activities to enhance understanding of more difficult coding-related concepts, practices and attitudes)
- Step 4: TPACK (Applying knowledge of using programming environments for teaching CK with appropriate pedagogy in a specific context)
- Step 5: TCK (Encouraging students to suggest applications of relevant features of the programming environment in other contexts, thereby inspiring their digital creativity)
- Step 6: CK (Helping students reflect on CT concepts, practices and attitudes to consolidate their learning)
- Step 7: PCK (Conducting self-reflection on the pedagogy adopted in the unit with a view to improve the next round of teaching)

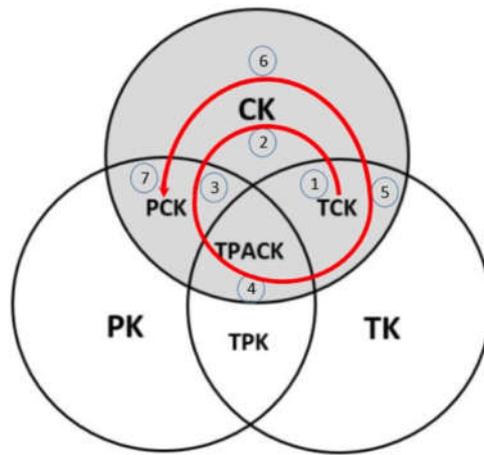


Figure 1 The seven steps in the shaded areas (CK, TCK, PCK, and TPACK) indicate those steps needed for teachers to teach content knowledge of CT. (Kong, Lai & Sun, 2020)

## References

Education Bureau. (2020). *Computational Thinking - Coding Education: Supplement to the Primary Curriculum*. Hong Kong: Author.

Kong, S. C., & Lai, M. (2022). A proposed computational thinking teacher development framework for K-12 guided by the TPACK model. *Journal of Computers in Education*, 9(3), 379-402.

Kong, S. C., Lai, M., & Sun, D. (2020). Teacher development in computational thinking: Design and learning outcomes of programming concepts, practices and pedagogy. *Computers & Education*, 151, 103872.

Kong, S. C., Lai, M., & Li, Y.G. (2023). Scaling up a teacher development programme for sustainable computational thinking education: TPACK surveys, concept tests and primary school visits. *Computers & Education*, 194, 104707.

## **Enriched Module on Coding Education for Upper Primary Level (Primary 4)**

### **Editorial Board Members**

#### **Professor KONG Siu Cheung**

Research Chair Professor of E-Learning & Digital Competency  
Department of Mathematics and Information Technology  
Director, Artificial Intelligence and Digital Competency Education Centre  
Director, Centre for Learning, Teaching and Technology  
The Education University of Hong Kong

#### **Ms SIU Ka Yuk Cora**

Educational Development Manager  
Centre for Learning, Teaching and Technology  
The Education University of Hong Kong

#### **Ms MA Yunsi Tina**

Assistant Educational Development Manager  
Centre for Learning, Teaching and Technology  
The Education University of Hong Kong

#### **Miss TING Wan Yee**

Assistant Educational Development Manager  
Centre for Learning, Teaching and Technology  
The Education University of Hong Kong

#### **Mr LO Hoi Lam Ken**

Educational Development Officer  
Centre for Learning, Teaching and Technology  
The Education University of Hong Kong

# Content for Booklet 1

<b>Unit</b>	<b>Unit Title</b>
1	Introducing Scratch Programming
2	Exploring Under the Sea
3	Storytelling
4	Space Traveling

# Unit 1: Introducing Scratch Programming Teacher Guide

## Content

<b>Teaching Plan</b>	P1
<b>Lesson 1</b>	
To Play	T1-2
To Learn	T1-4
To Code	
(1) Adding / Changing Sprite	T1-5
(2) Adding / Changing Background	T1-6
(3) Event – When  Green Flag Click that trigger actions	T1-7
(4a) Make the Sprite Move	T1-8
(4b) Make the Sprite Change Degree	T1-9
(5) Change the Sprite Costume	T1-10
(6) Adding Sound to the Sprite	T1-11
<b>Lesson 2</b>	
To Create	T1-14
To Reflect	T1-17
Review Questions	T1-18
Revision on Key Concepts & Practices	T1-20
<b>Appendix - Guidelines for Scratch Teacher Account</b>	T1-22

## **Unit 1: Introducing Scratch Programming Teaching Plan**

### **Learning Objectives**

1. Get familiar with the Scratch interface, such as Block Palette, Code Area, Stage and Sprite Pane, Green Flag, Stop Button;
2. Complete tasks in the Scratch programming environment, such as create, save, and share projects to studio;
3. Manipulate a Scratch sprite (e.g. cat) using sequences of code blocks;
4. Know some basics skills of coding with Scratch, e.g. adding / changing sprites, changing backdrop, playing sound in a sprite, movement of a sprite, changing costumes of a sprite;
5. Understand that coding is a creative activity.

### **Learning Elements**

#### **Computational Thinking Concepts and Practices:**

Key Learning Elements	Items
Algorithm	Basic Programming Constructs: Sequence Coding Concept and Practices: Design programs / codes, Testing and Debugging

### **Coding Skills:**

1. Be able to sign in and sign out the Scratch account;
2. Get familiar with the Scratch interface;
3. Manipulate a Scratch sprite (e.g. cat) using sequences of code blocks;

### **Others (including Attitude):**

1. Develop interest in coding;
2. Cultivate an understanding that computing can be a creative activity.

**Lesson Plan:** This unit consists of 2 lessons of 35 minutes.

**Teacher Preparation of computer device and setting for the lesson:** Prepare appropriate computer or mobile device for students to use Scratch. Prepare accounts for students to login Scratch. Prepare a studio for students to share their projects. Please see the “Appendix: Guideline for Scratch Teacher Account” in Teacher Guide for reference.

### Lesson 1

Time	Activity
5 mins	<b>Introduction of Scratch</b> <ol style="list-style-type: none"><li>1. Point out that there are many programming platforms / tools for students to learn coding. In this set of L&amp;T Resources, Scratch is used to teach coding.</li><li>2. Guide students to go to Scratch website <a href="http://scratch.mit.edu">http://scratch.mit.edu</a> and get familiar with the coding platform.<ol style="list-style-type: none"><li>1) Sign into the account</li><li>2) Start a new project by clicking “Create” button.</li><li>3) Sign out</li></ol></li></ol>
10 mins	<b>To Play: Let students Explore Scratch Coding Environment</b> <ol style="list-style-type: none"><li>1. Ask students to explore the Scratch “Tutorials” and start with the “Getting Started” video.</li><li>2. Ask them to try a few different games they like and explore what can be created by using Scratch.</li><li>3. Invite students to jot down the games they like and think about why they like them.</li><li>4. Briefly introduce what Scratch is good for.</li></ol>
5 mins	<b>To Learn</b> <ol style="list-style-type: none"><li>1. Explain the interface of the coding environment in Scratch, which consists of four parts, namely Block Palette, Code Area, Stage and Sprite Pane.</li><li>2. Remind students that it is important to get familiar with the coding environment.</li></ol>
15 mins	<b>To Code</b> <ol style="list-style-type: none"><li>1. Guide students to explore and learn the following basic skills of coding with the Scratch platform with the simple task:<ol style="list-style-type: none"><li>(1) Adding / Changing Sprite</li><li>(2) Adding / Changing Background</li><li>(3) Event – When Green Flag Click that trigger actions</li><li>(4) Doing simple motion: control the sprite to move, changing degree of rotation</li></ol></li></ol>

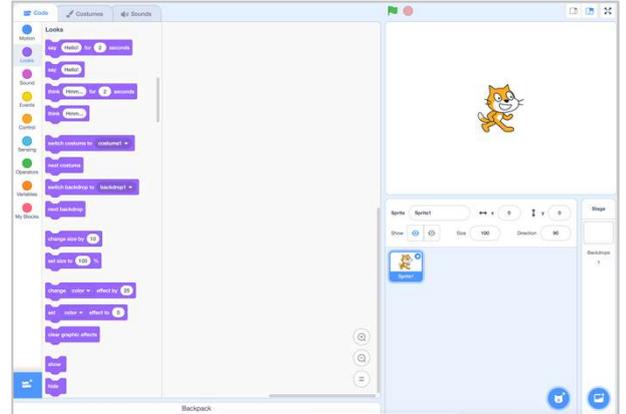
	<p>(5) Changing costumes</p> <p>(6) Adding sound to the sprite</p> <ol style="list-style-type: none"><li>2. Students can further explore / complete the “Make It Spin” and “Add Effects” (or other tutorials) after acquiring the above basics.</li><li>3. Ask them to change the values of the different blocks and explore the differences.</li><li>4. Remind students that they can click the green flag anytime to test if the project works as they expect.</li></ol>
--	--

## Lesson 2

Time	Activity
5 mins	<b>Introduction of the Lesson</b> <ol style="list-style-type: none"><li>1. Guide students to create a new Scratch project to make the cat perform different actions and effects. Teachers should encourage students to make use of the skills they have learnt from the simple tasks earlier.</li><li>2. Tell students they will share their projects to the studio before the end of the lesson.</li></ol>
15 mins	<b>To Create: Task 1</b> <ol style="list-style-type: none"><li>1. Ask students to explore different Scratch blocks. They can experiment it by clicking on each block and see what it does or try snapping blocks together.</li><li>2. Remind students that they can create whatever they want! It is a chance for them to explore different blocks in Scratch.</li><li>3. Student could try to take reference to the Student Guide (optional).</li><li>4. Remind students that they can click the green flag anytime to test if the project works as they expected.</li><li>5. Explain the basic programming construct “Sequence” after student experience their coding task.</li></ol>
15 mins	<b>To Reflect:</b> <b>Share the Projects and Provide Constructive Feedback on Program Design</b> <ol style="list-style-type: none"><li>1. Create a studio and give students the studio URL. Ask students to save and submit their projects to the teacher’s Studio.</li><li>2. Have one or two students share their project with the whole class. Feedback to be given by peer and teachers.</li><li>3. Students should share, think of ways to improve / enhance their program as well as the aesthetic values.</li></ol> <b>Review of Student Learning</b> <ol style="list-style-type: none"><li>1. Review on the features of Scratch, and key concepts and practices learnt in the unit.</li><li>2. Ask students to complete the review questions, appropriate feedback should be given by teachers.</li></ol>

# Introducing Scratch Programming

In this activity, you will learn to sign in and out of the Scratch website and learn how to make the cat move and play music.



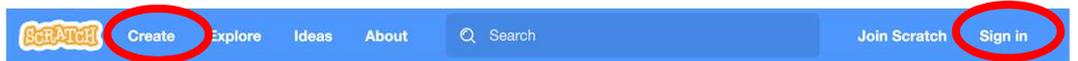
Briefly point out that there are many programming platforms / tools for students to learn coding. In this set of L&T Resources, Scratch is used to teach coding.

After “To Play”, teachers can introduce what Scratch is good for, e.g. students can use it for making games, animation and stories, also adding sounds and images to projects by using coding blocks.

## Start Here

- Go to the Scratch website:

<http://scratch.mit.edu>

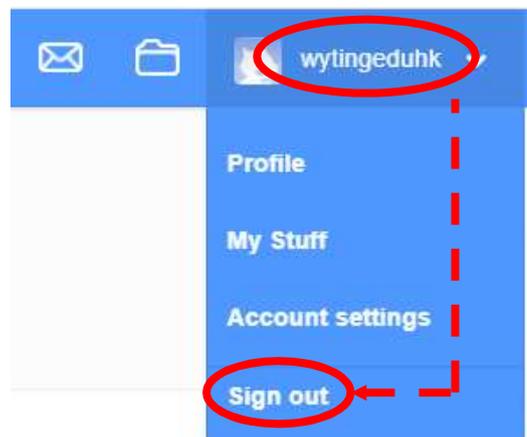


- Sign into your account.



- Click on the **Create** tab located at the top left of the browser to start a new project.

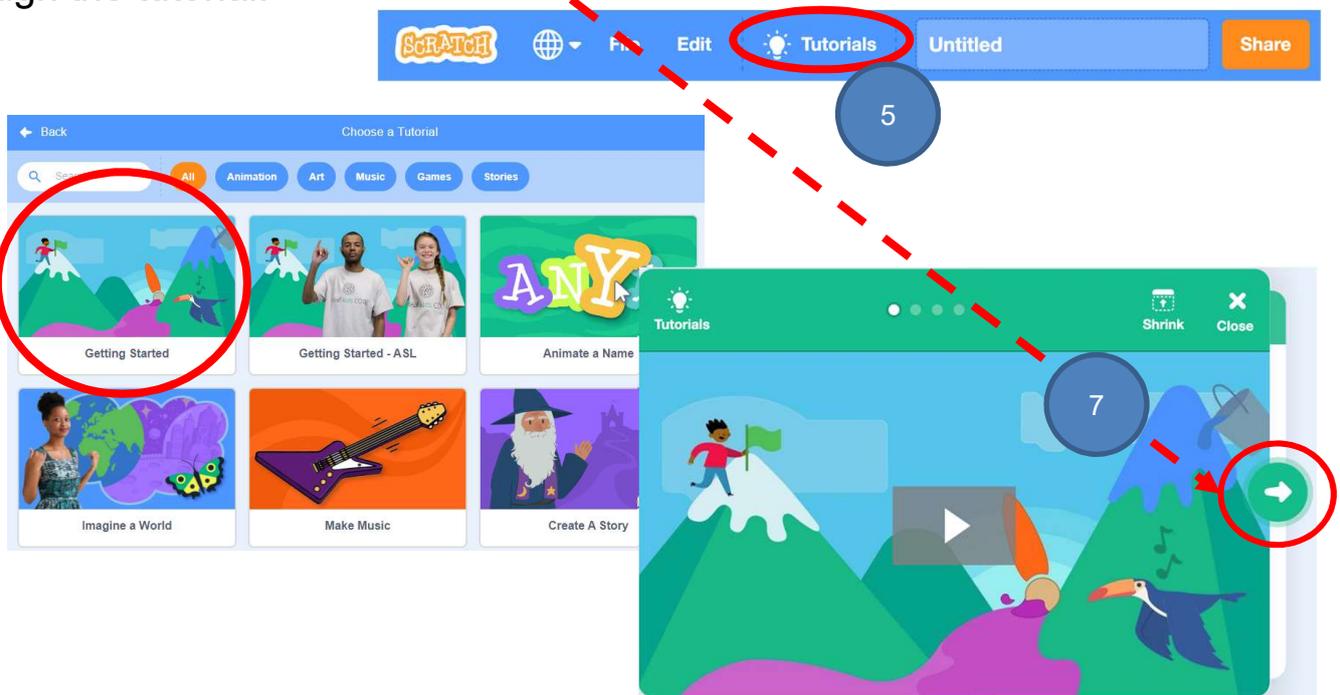
- Sign out by clicking on your name and clicking **Sign out**. Then sign in again!



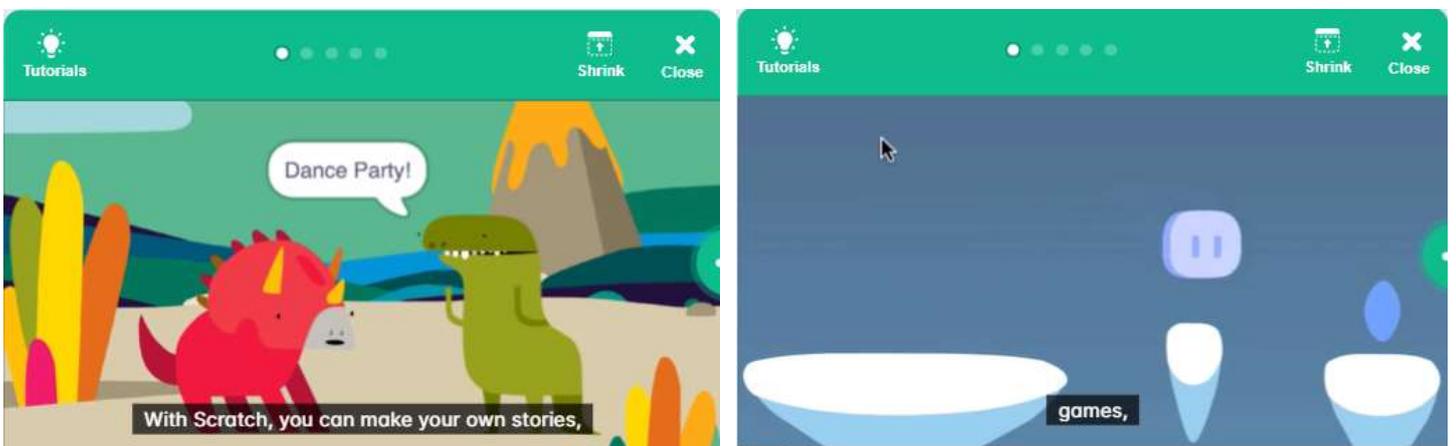
# Introducing Scratch Programming

## To Play

- Time to explore! After clicking “Create”, you will see “Tutorials” on the menu bar. Click on “Tutorials” and choose “Getting Started”. After watching the video, you can click the green arrow on the right side of the window to step through the tutorial.



- With Scratch, you can make your own stories, games and animations!

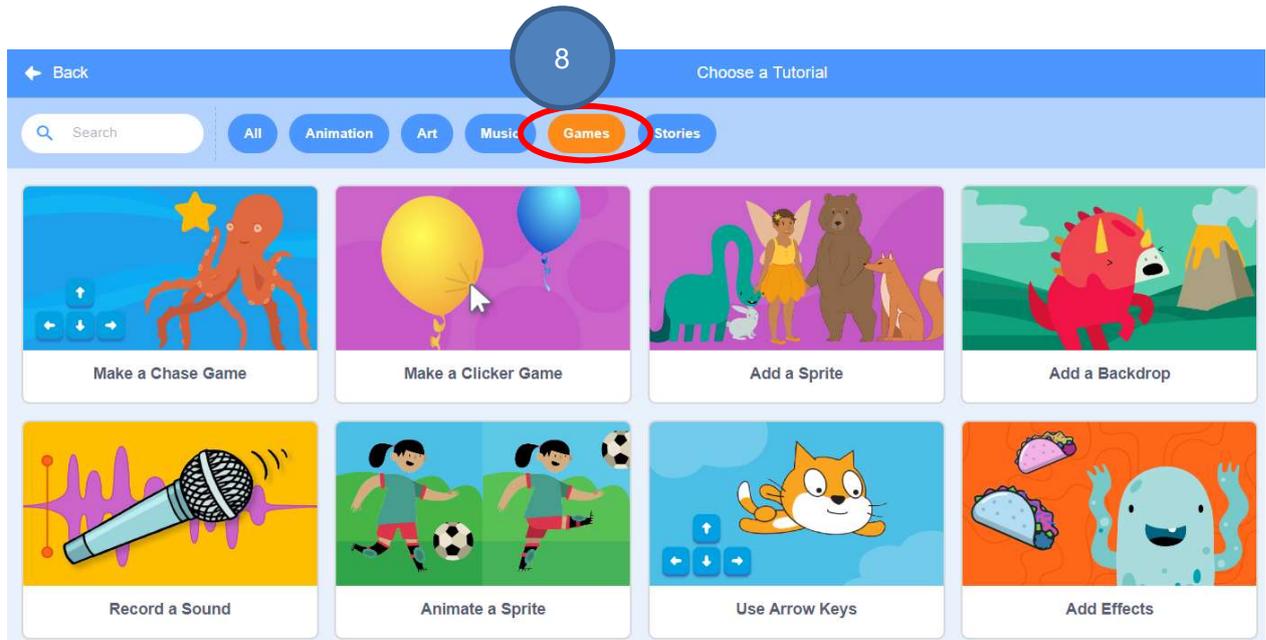


Briefly introduce that Scratch is good for making games, animation and stories by drag and drop some coding blocks, adding sounds and images.

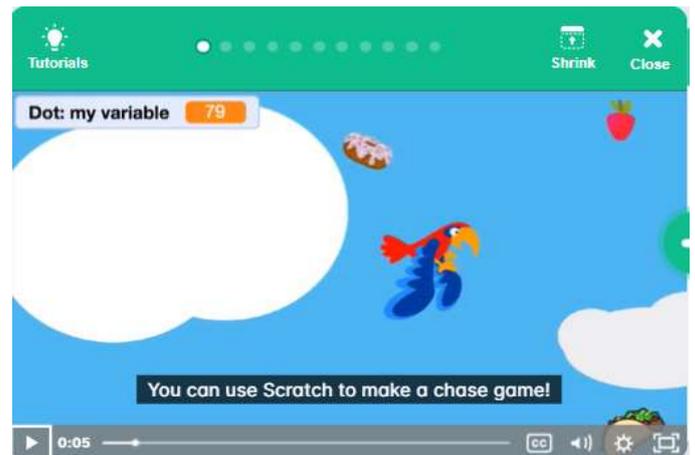
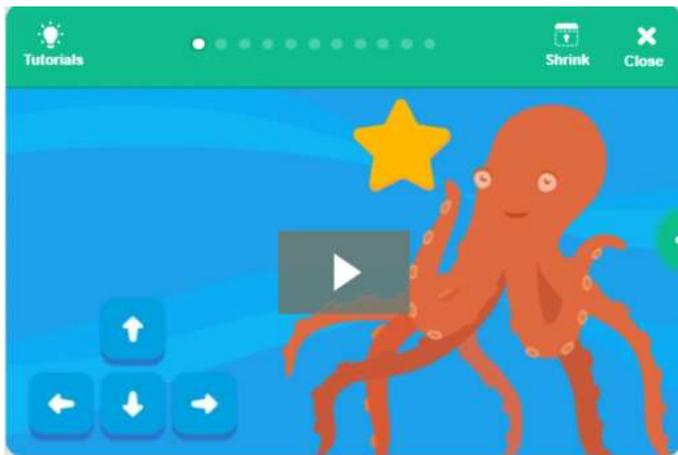
# Introducing Scratch Programming

## To Play

- ❑ Let's try some games! Select "Games" from the category on the top to see a list of games. Choose the games you like and try it!



- ❑ Try different games and see what can be created by using Scratch!



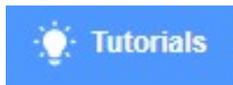
- ❑ Jot down the games you like and think about why you like them.

Ask students to share why they like certain games.

# Introducing Scratch Programming

## To Learn

### Coding Environment in Scratch



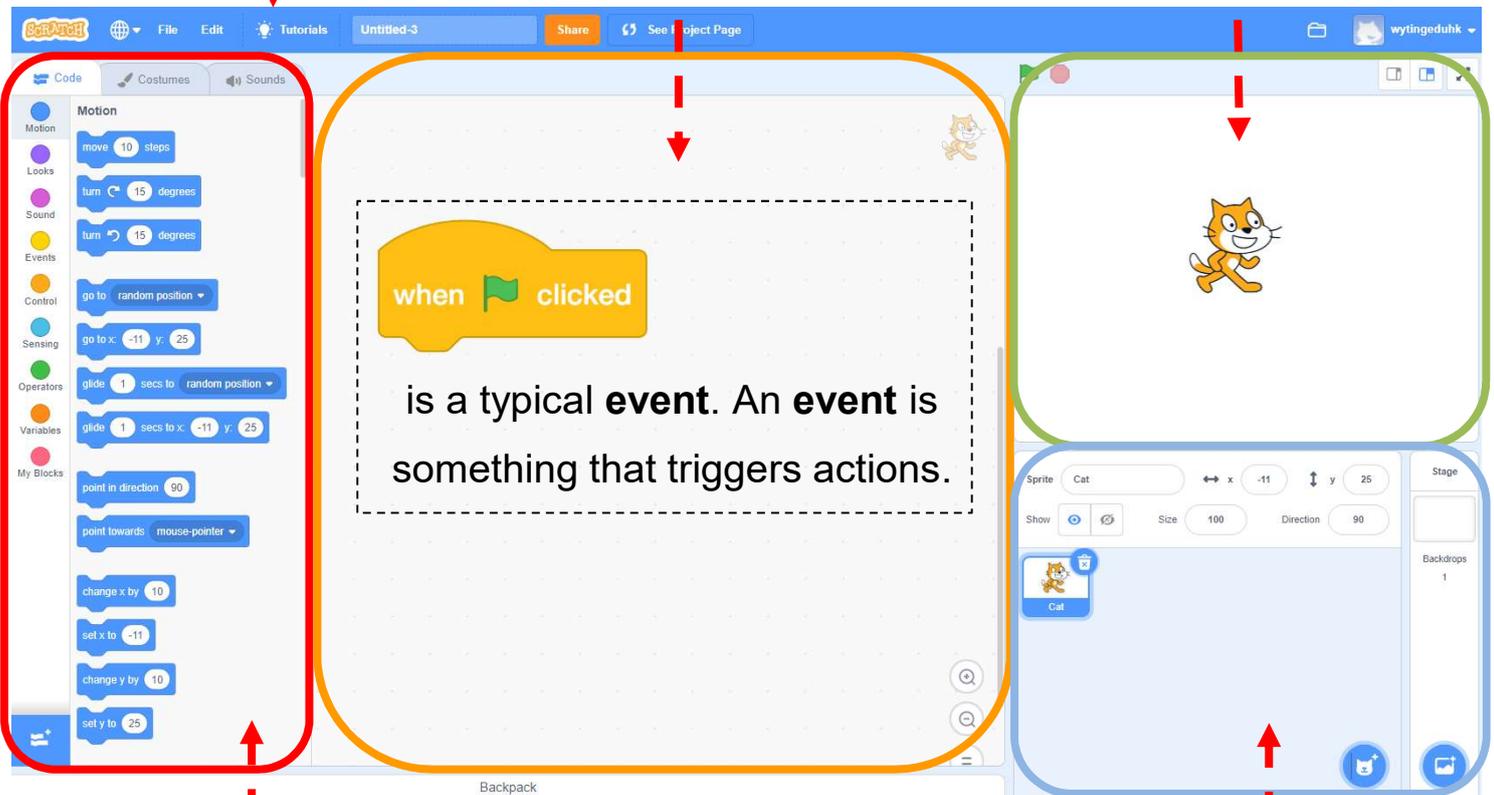
Watch tutorials to learn about Scratch.



An area for storing blocks that run the project.



An area where the sprites are displayed and perform their actions.



**Code:** Drag and drop the coding blocks to the Code Area

**Costumes:** Changing sprite's costumes

**Sounds:** Adding sound



- Adding / Changing Sprite
- Adding / Changing Backdrop

# Introducing Scratch Programming

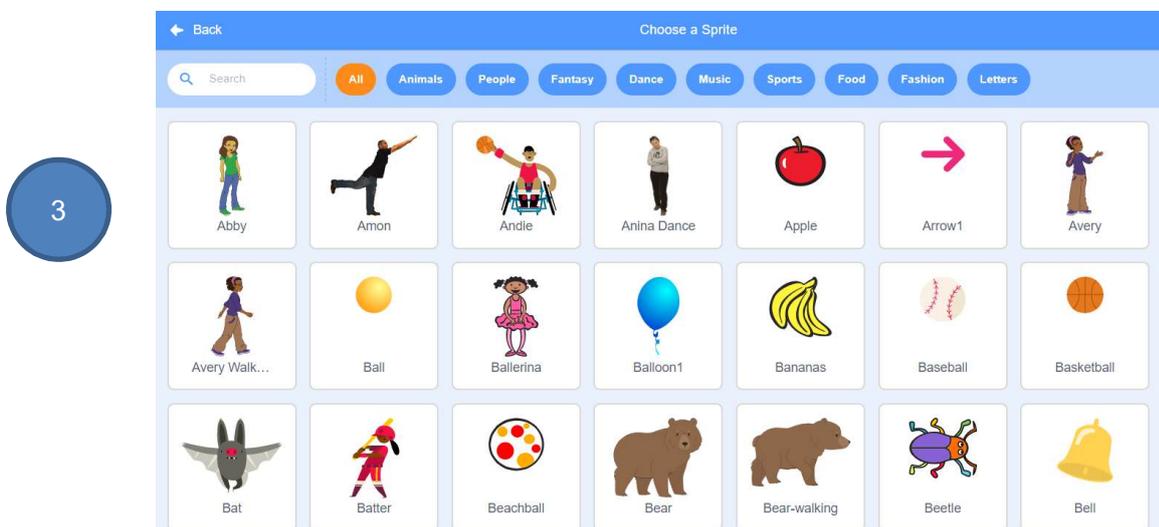
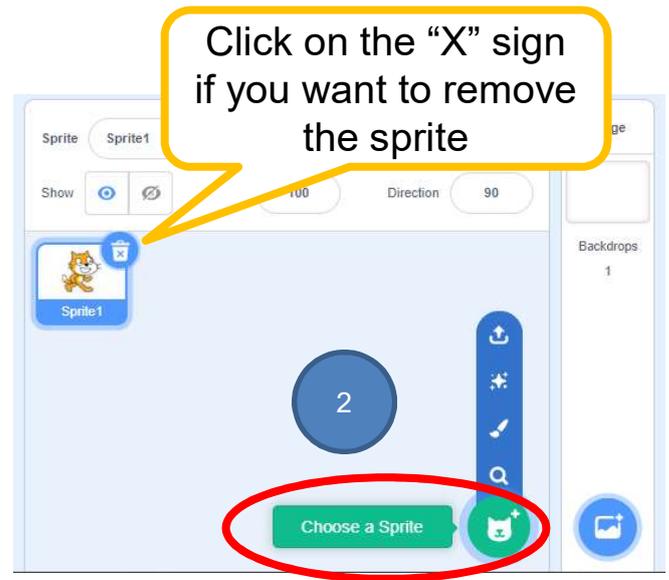
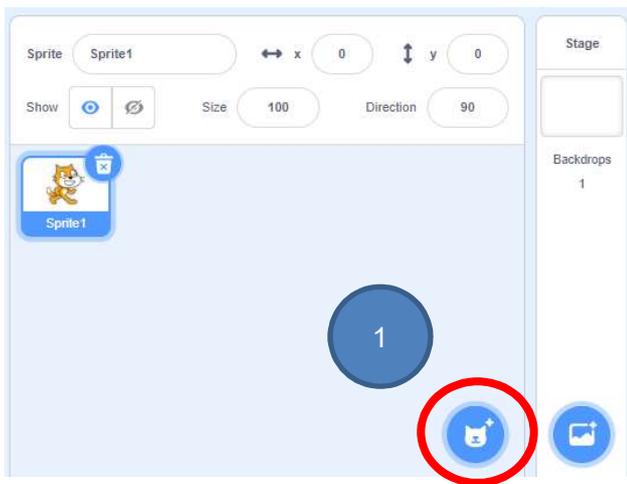
## To Code

Let's try the **simple tasks (1) – (6)** to explore what can be done with Scratch!

### (1) Adding / Changing Sprite

- ❑ In the Sprite Pane, click “Choose a sprite”.
- ❑ A set of sprites will be shown in different categories. Choose your favourite one!

Teachers can provide support to better cater their student need, or let the student with stronger ability to explore Scratch platform on their own.

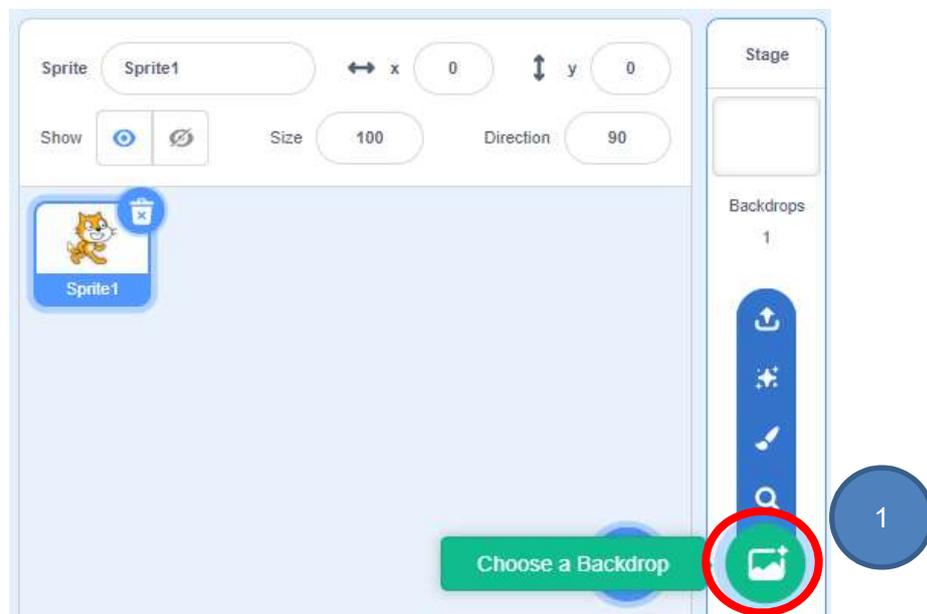


# Introducing Scratch Programming

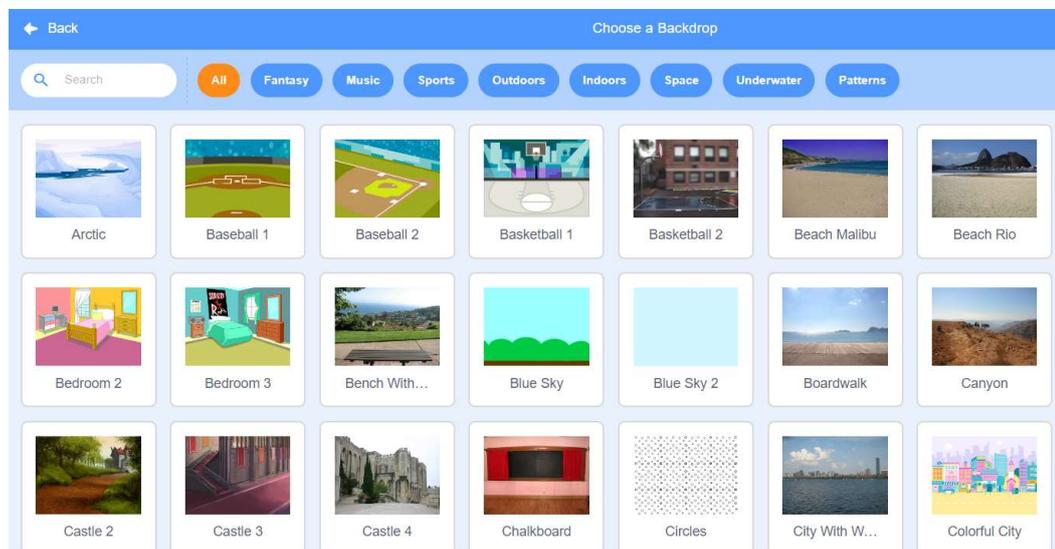
## To Code

### (2) Adding / Changing Background

- ❑ In the Sprite Pane, click “Choose a background”.
- ❑ A set of backgrounds are now shown. Choose a good one for your project!



2

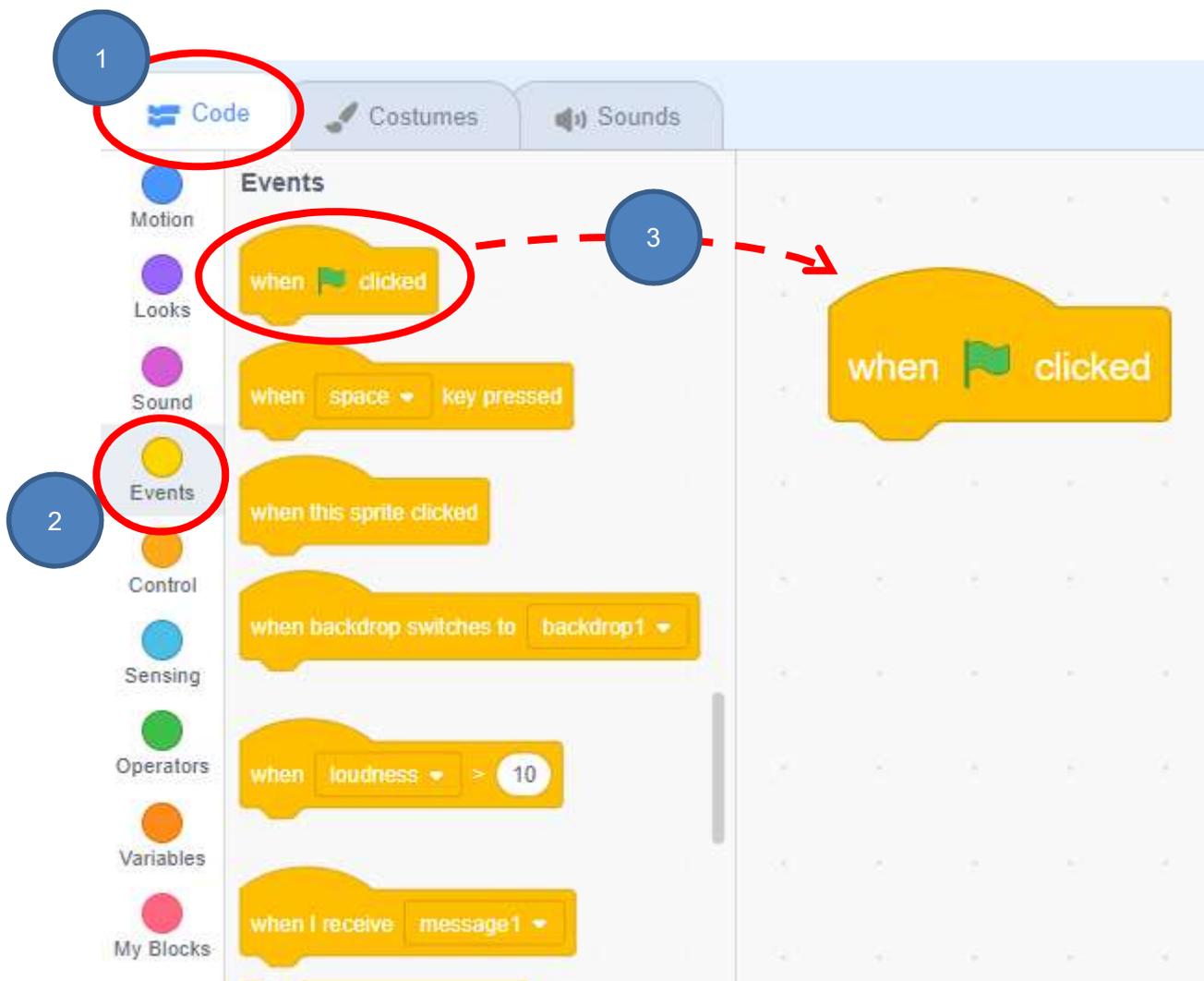


# Introducing Scratch Programming

## To Code

(3) Event – When  Green Flag Click that trigger actions

- ❑ Let's start to code!
- ❑ In the Block Palette, click on “Code” at the top menu.
- ❑ Choose “Events” and drag the “When  clicked” block to the Code area.

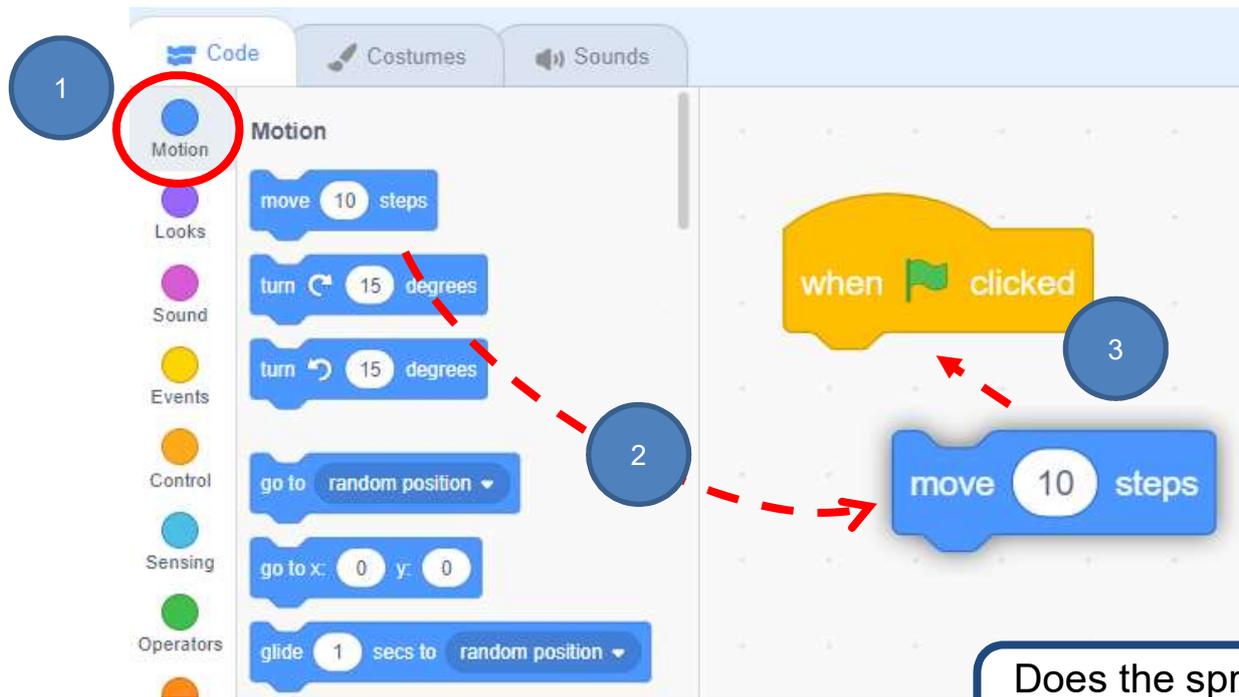


# Introducing Scratch Programming

## To Code

### (4a) Make the Sprite Move

- ❑ Time to make your sprite move! Choose “Motion” and drag the “move 10 steps” block to the Code area. Snap it with “When  clicked” block.



### Testing and Debugging

- ❑ Let's test it! Click the  to test and see if it works!

**Testing** a computer program is the process of checking if it can produce results as designed.

**Debugging** a computer program is the process of finding out ways to revise the program so that the bugs can be removed.



Does the sprite move? How it moves?



What if you put a negative number here? Let's try..

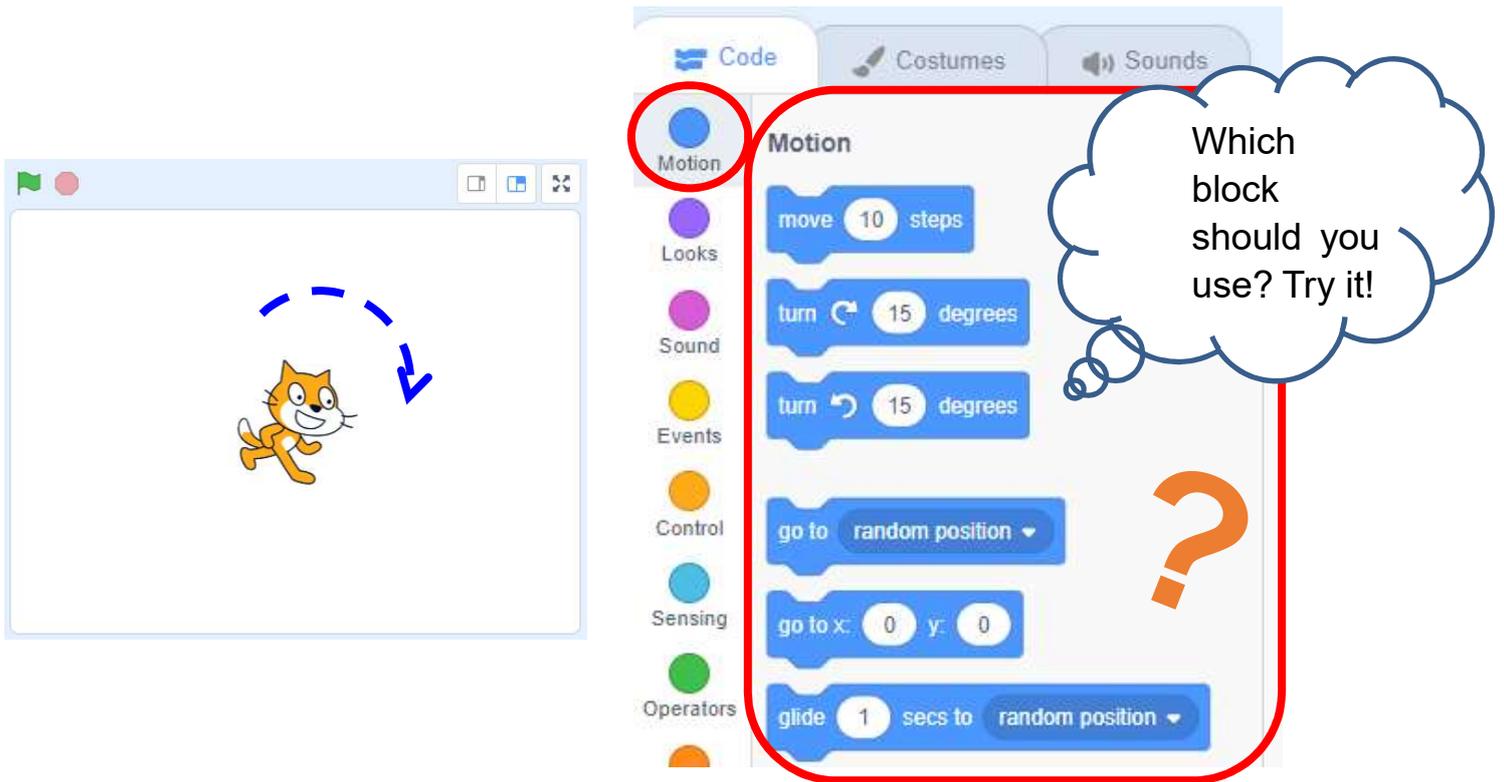


# Introducing Scratch Programming

## To Code

### (4b) Make the Sprite Change Degree

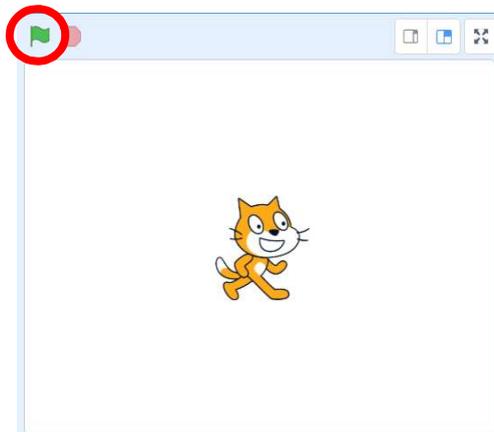
- ❑ Can you try to make the sprite turn around like this?



The image shows the Scratch interface. On the left, a cat sprite is shown with a dashed blue arrow indicating a 180-degree turn. On the right, the code editor is open to the 'Motion' category. A red circle highlights the 'Motion' category icon. A red box highlights the 'Motion' block palette, which includes blocks like 'move 10 steps', 'turn 15 degrees' (clockwise and counter-clockwise), 'go to random position', 'go to x: 0 y: 0', and 'glide 1 secs to random position'. A thought bubble with a question mark asks, 'Which block should you use? Try it!'.

### Testing and Debugging

- ❑ Remember, you can always click the  and see if it works!

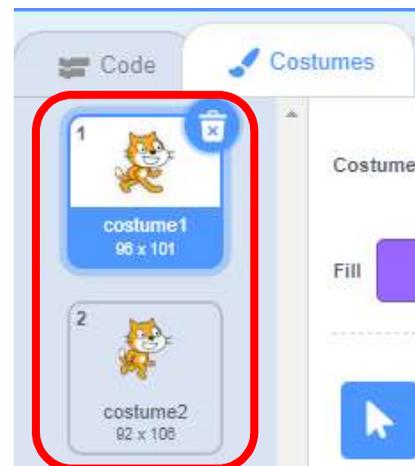
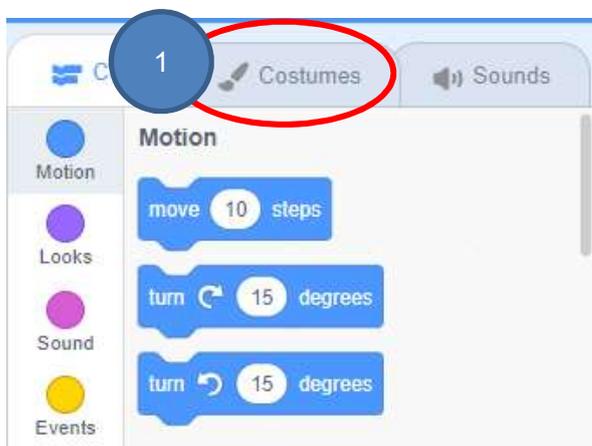


# Introducing Scratch Programming

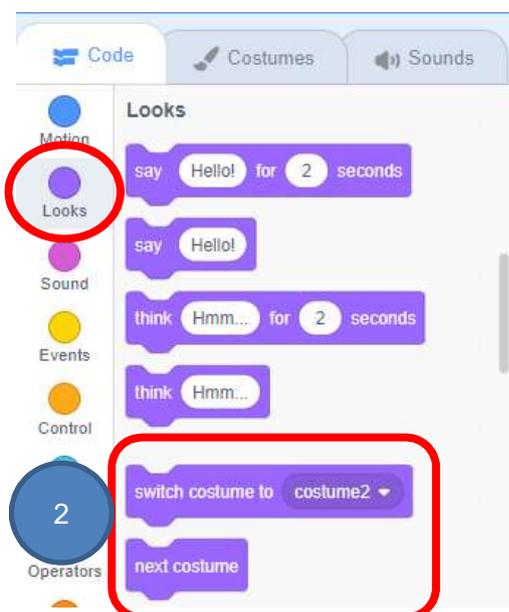
## To Code

### (5) Change the Sprite Costume

- ❑ Now let's try to **change the sprite costume** to make our project more interesting!
- ❑ Click on **"Costumes"**, you will see a set of costumes of the sprite you added.



- ❑ Go back to **"Code"**, choose **"Looks"** and change the sprite's costumes!
- ❑ Drag **"switch costume to \_\_\_"** and **"next costume"**, snap with **"when green flag clicked"** block. Click the  and see what happens.



### Testing and Debugging



The cat should look like running now!  
Can you do it?

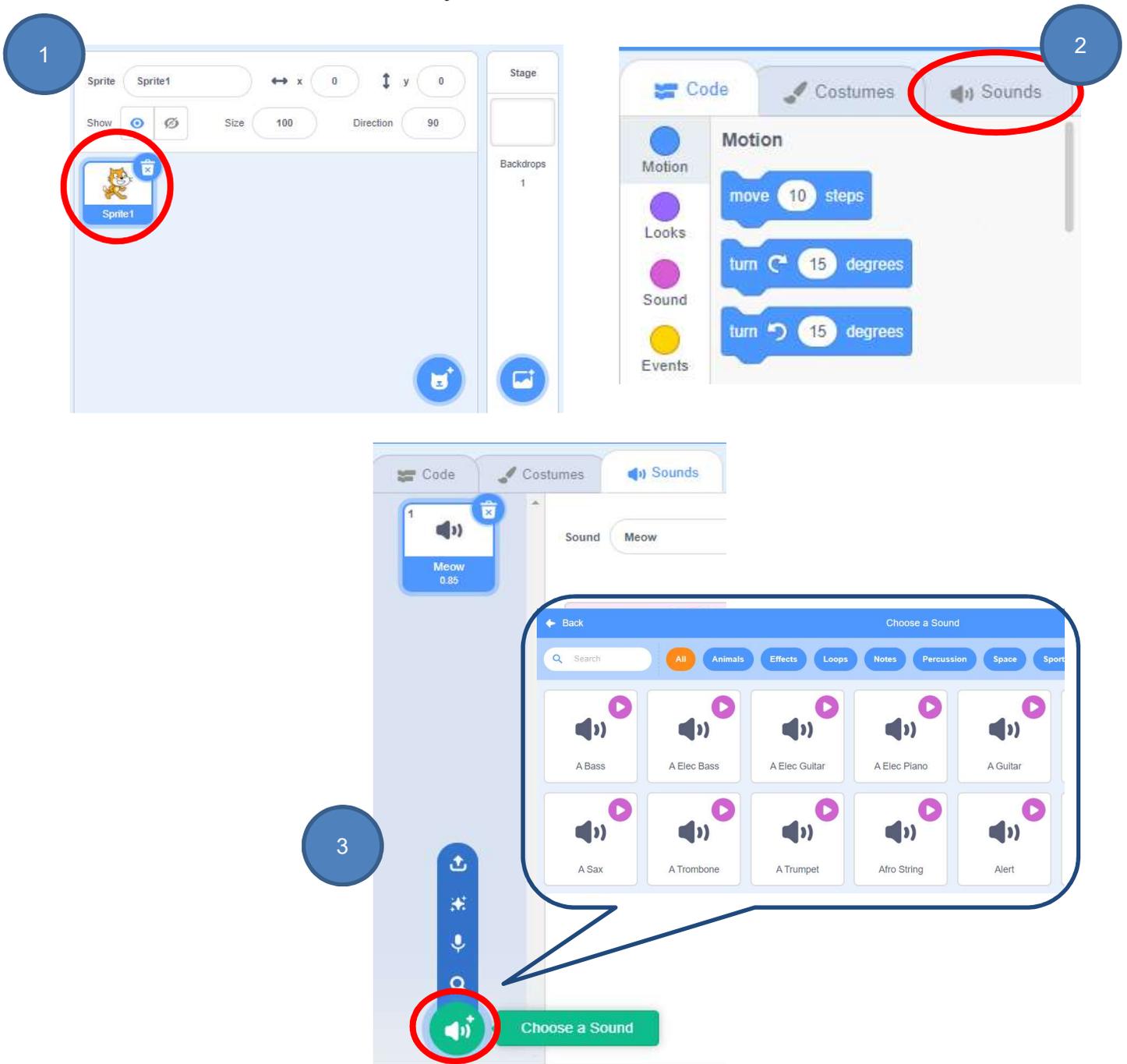


# Introducing Scratch Programming

## To Code

### (6) Adding Sound to the Sprite

- ❑ Click on your Sprite, go to the “Sound” page, click “Choose a Sound”, you will see a lot of sound that you can choose.



# Introducing Scratch Programming

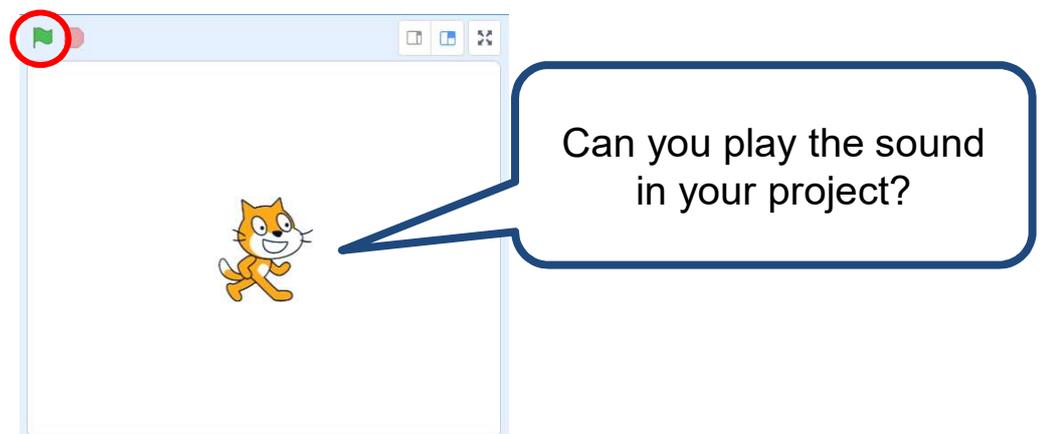
## To Code

### (6) Adding Sound to the Sprite

- ❑ In the “Sound” drawer, drag “play sound\_\_until done” or “start sound \_\_\_”, snap with “when  clicked” block.



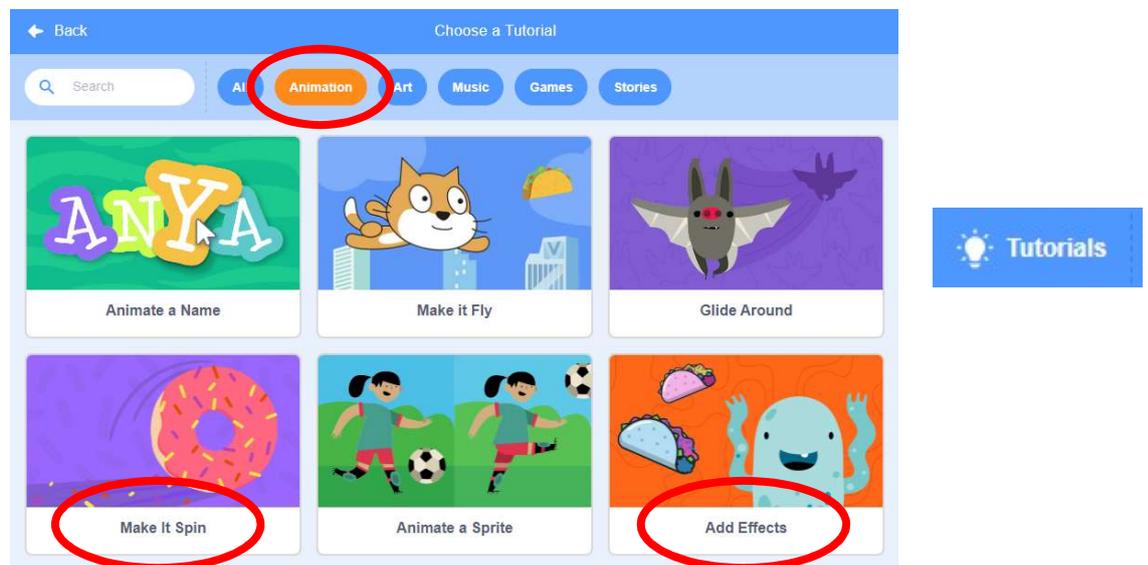
### Testing and Debugging



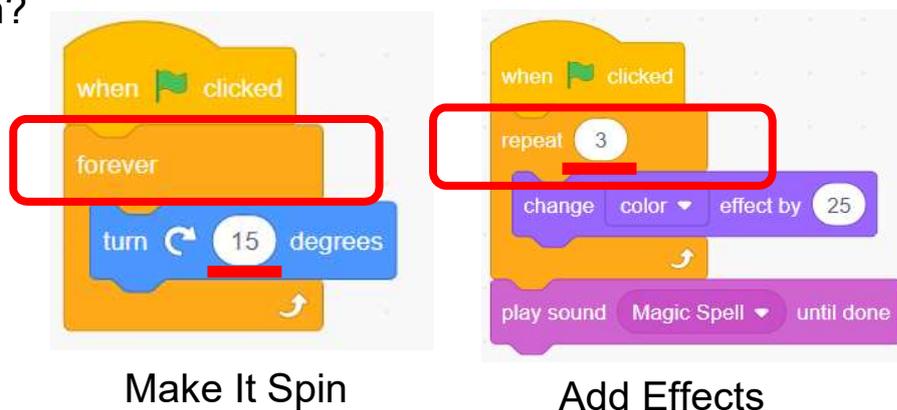
# Introducing Scratch Programming

## To Code

- Now you have got some basic skills for Scratch! Let's explore more.
- Click on "Tutorials" on the menu bar again and then choose the "Animation" category on the "Tutorials" page.
- Complete the "Make It Spin" and "Add Effects" tutorials.



- Change the value of turn degrees or the number of repeat times and see what happen?



When I change the value of turn degrees, I can see... **The Sprite turns differently.**

When I change or the number of repeat times, I can see... **The Sprite changes colours differently.**

# Introducing Scratch Programming

CAN YOU MAKE THE SCRATCH CAT DO SOMETHING INTERESTING?

In this activity, you will create a new project with Scratch and make the cat do different actions with various costumes! What will you create?



## To Create

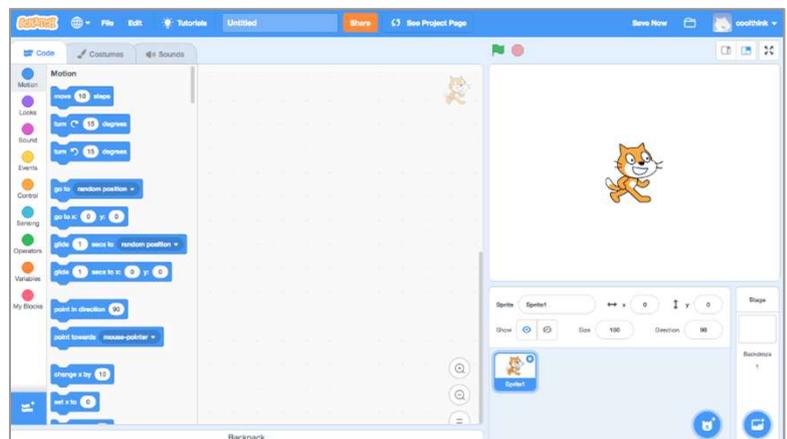
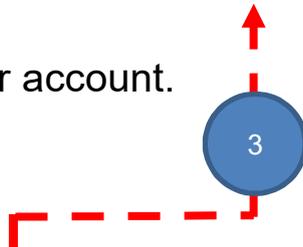
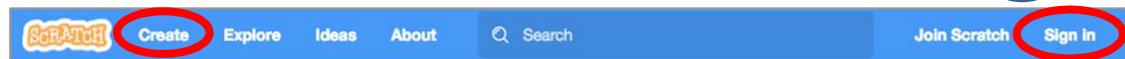
- ❑ Go to the Scratch website:

<http://scratch.mit.edu>

- ❑ Sign into your account.

- ❑ Click on the **Create** tab located at the top left of the browser to start a new project.

- ❑ See if you can use the right blocks to make the cat do what you want.

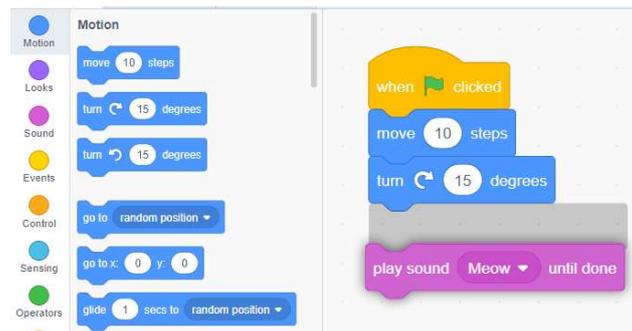


# Introducing Scratch Programming

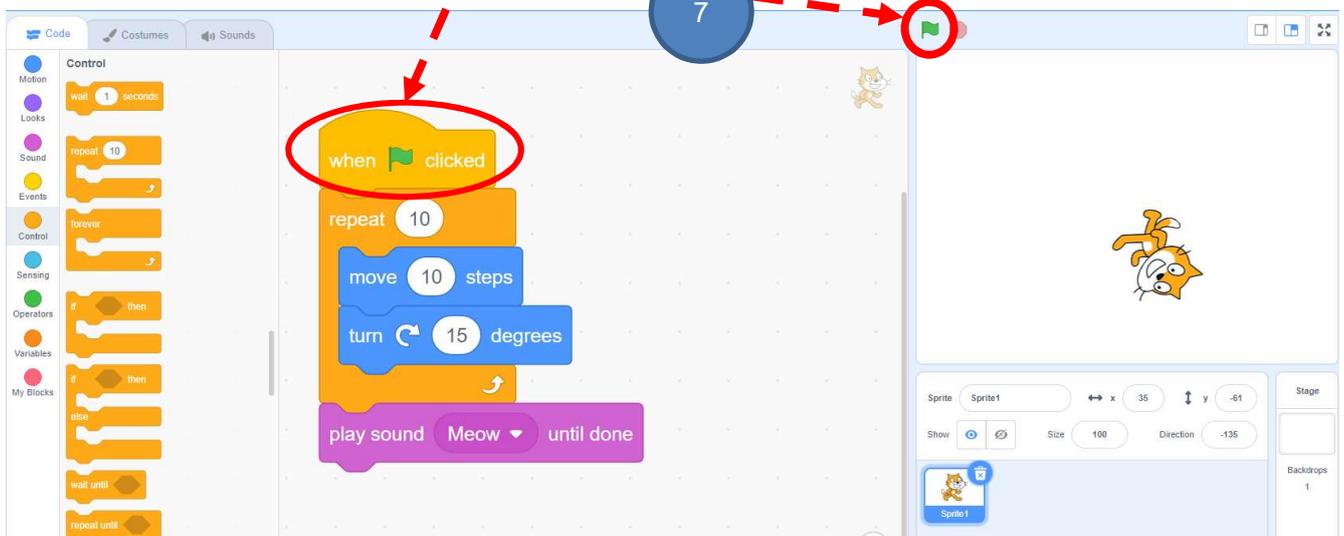
## To Create

Task 1:

- ❑ Drag and drop Scratch blocks into the Code window.
- ❑ Experiment by clicking on each block to see what it does or try snapping blocks together.



- ❑ Don't forget to **Click the Green flag** anytime to test if your sprite works as you expected!



When the Green flag is clicked, does your sprite:

- ✓ Move?
- ✓ Turn around?
- ✓ Make sound?

### To Think:

Do you think there is any **sequences** in the tasks (move, turn and play sound)?



### Knowledge builds up: Sequences

It is a key concept in programming. It is the order in which the programming statements are executed. A wrong order would lead to incorrect programming results.

# Introducing Scratch Programming

## Sharing in Scratch Studio

LEARN HOW TO ADD YOUR PROJECT TO AN ONLINE SCRATCH STUDIO!

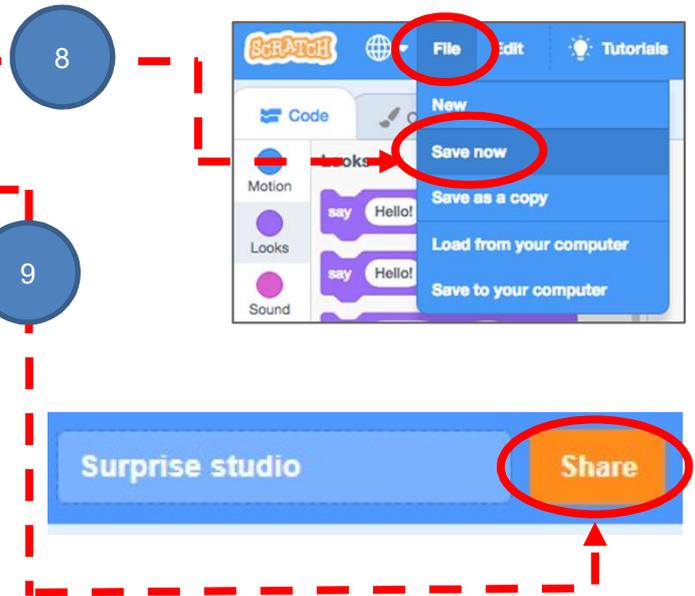
Studios are collections of Scratch projects. Follow the steps below to add your Scratch Surprise program to your class' Surprise Studio on the Scratch website.

### Share to Studio

When you finish, you will add your project to your teacher's Studio:

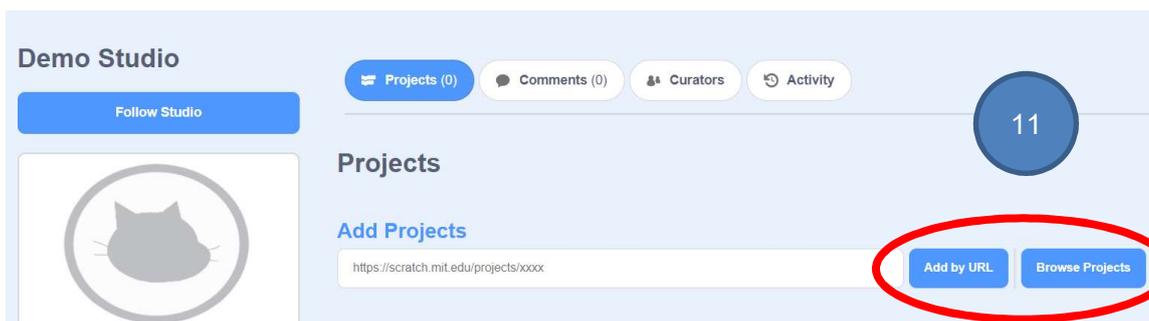
- Save your project by clicking "Save now" under the File menu.
- Click the orange "Share" button.
- Go to your teacher's Studio (Your teacher will give you a URL).
- In the "Add projects" column, you can Add by URL or Browse Projects.

Teacher provides the studio link and name for students to share their projects.



10

11



# Introducing Scratch Programming

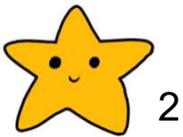
## To Reflect: Two Stars and a Wish Worksheet

Name of Project: \_\_\_\_\_ Name of Creator: \_\_\_\_\_

Please write down two things that you like about this project.



“Two Stars and a Wish” is a reflection strategy designed for student feedback as peer- and self-assessment. Teachers can guide students to give constructive feedback to their peers regarding their Scratch project - two positive (stars) and one hopeful (wish) reflection. Comments can be made on Scratch project’s idea, features and aesthetic aspects etc.



What is one thing you would like to add or change to make this project better?

---

---

---

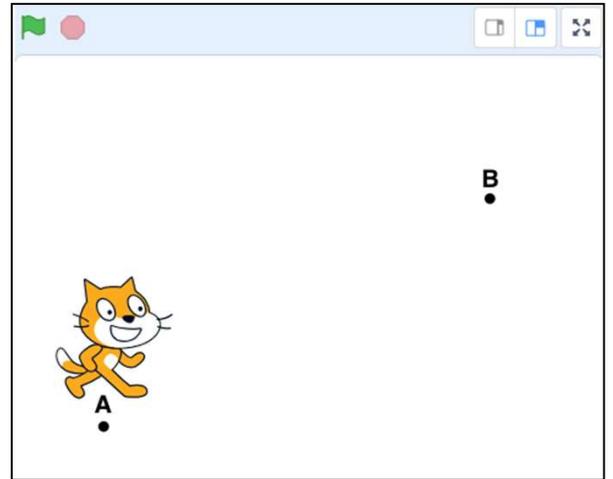


# Introducing Scratch Programming

## Review Questions

1. Which of the following sequences of commands (sets of blocks) will make the Scratch cat move from point A to point B on the stage?

**(Answer: A)**



A.

```
when green flag clicked
  move 320 steps
  say Where am I? for 1 seconds
  turn 90 degrees
  move 200 steps
```

B.

```
when green flag clicked
  move 320 steps
  say Where am I? for 1 seconds
  turn 270 degrees
  move 200 steps
```

C.

```
when green flag clicked
  move 320 steps
  say Hello! for 1 seconds
  turn 90 degrees
  move 200 steps
```

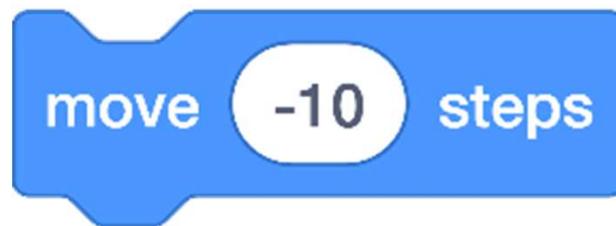
D.

```
when green flag clicked
  move 320 steps
  say Where am I? for 1 seconds
  turn 90 degrees
  move -200 steps
```

# Introducing Scratch Programming

## Review Questions

2. What happens when you put a negative number in the move block?



- A. The cat moves down on the screen.
- B. The cat moves forward (to the right) and backward (to the left).
- C. The cat turns around.
- D. The cat moves backward (to the left) on the screen.

**(Answer: D)**

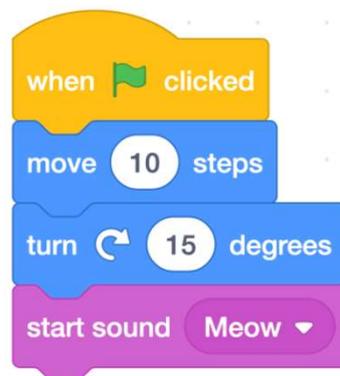
# Introducing Scratch Programming

## Revision on Key Concepts & Practices

**Events:** We use event blocks to trigger Scratch to take actions.



**Sequences:** It is a key concept in programming. It is the order in which the programming statements are executed. A wrong order would lead to incorrect programming results.

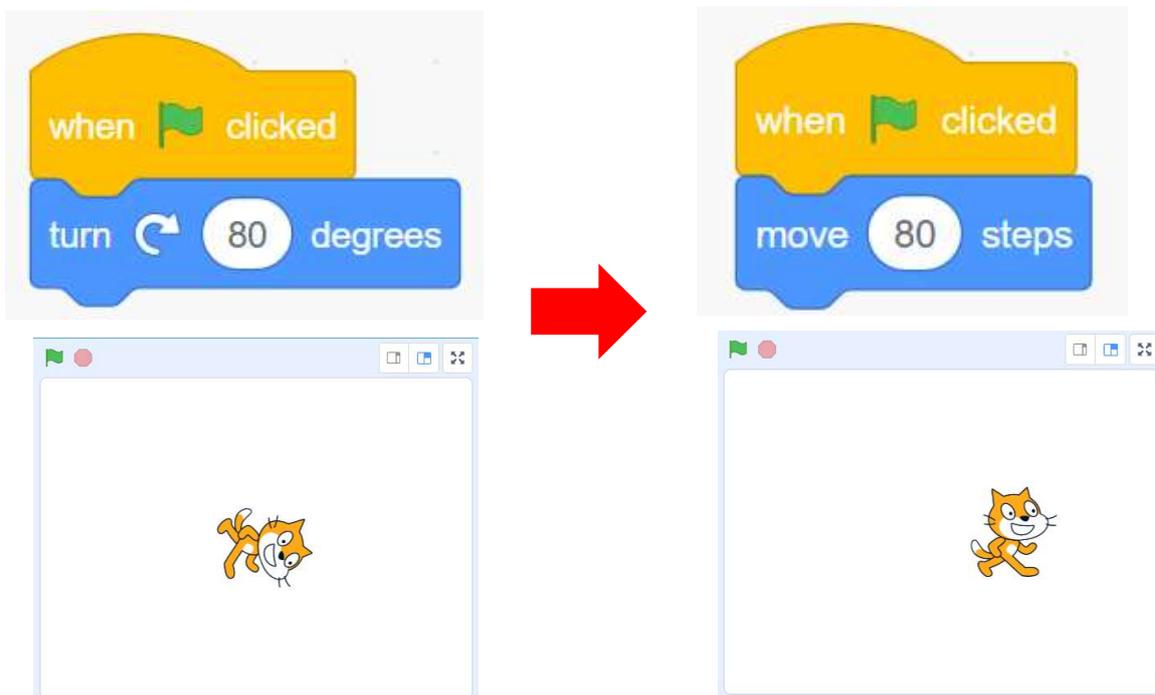


# Introducing Scratch Programming

## Revision on Key Concepts & Practices

**Testing and debugging:** Testing a computer program is the process of checking if it can produce outcomes as designed. Debugging a computer program is the process of finding out ways to revise the program so that the bugs can be removed.

E.g. I want to control the cat moving forward to the right. Test if the program works as designed, debug and remove the bugs.

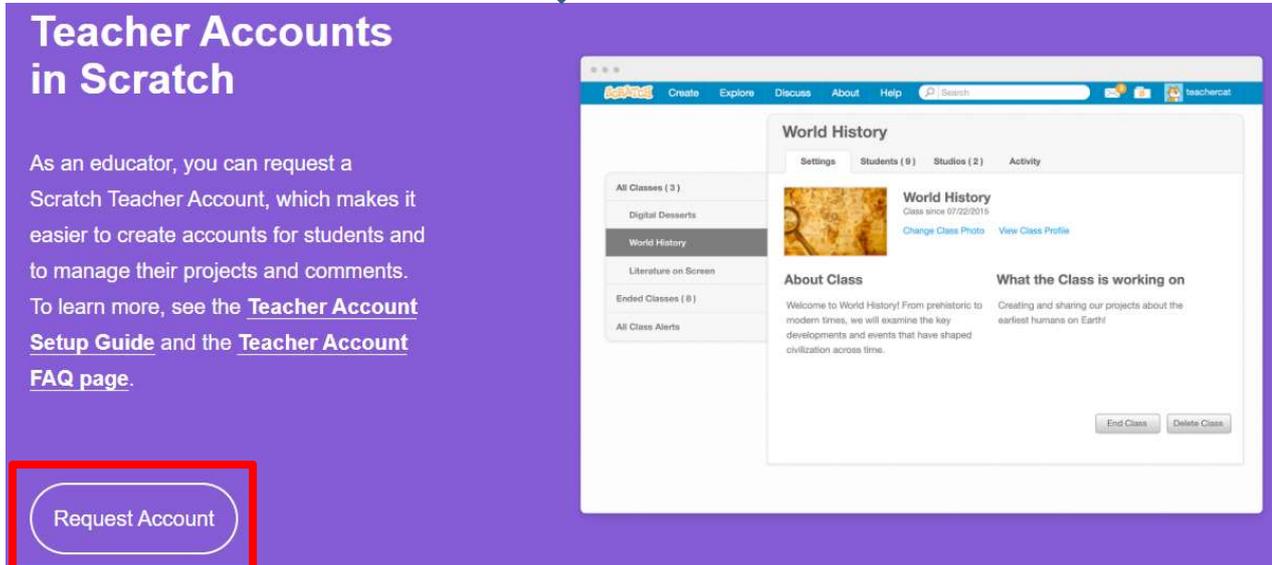
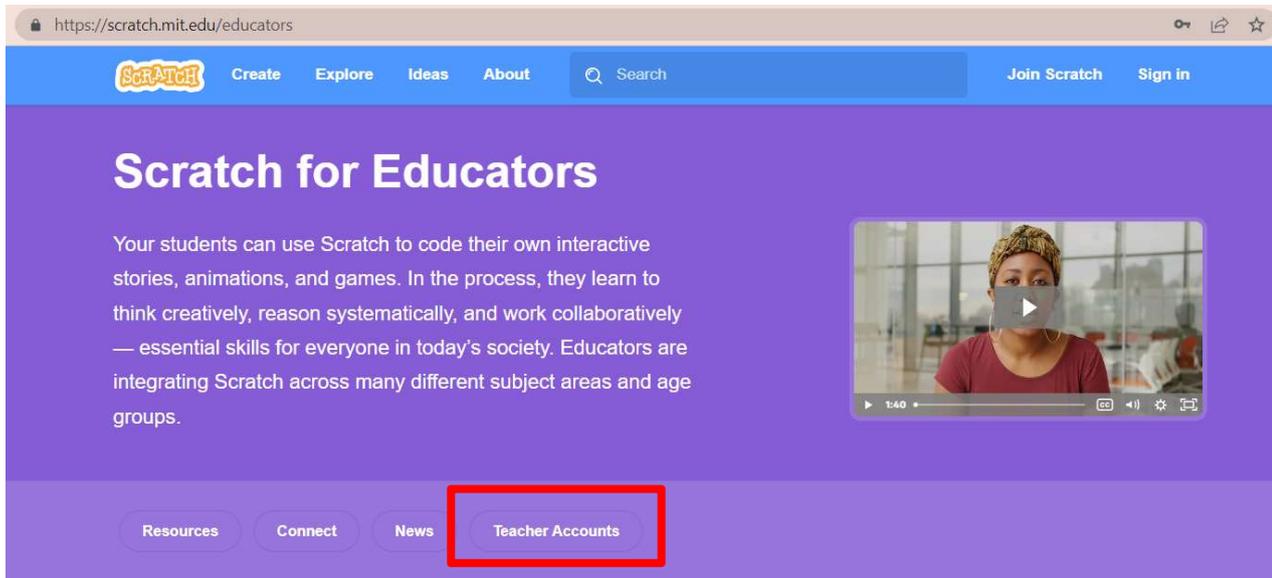


# Appendix

## Guidelines for Scratch Teacher Account

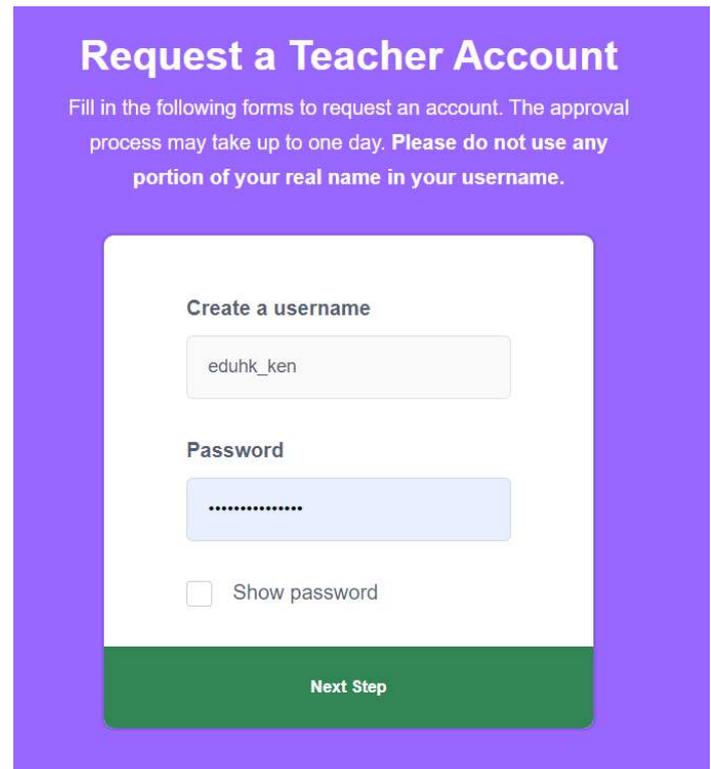
# Create Teacher Account

- Go to <https://scratch.mit.edu/educators>
- Click on **Teacher Account** and then **Request Account**.



# Create Teacher Account

- Type Username and Password. Then Next Step.



**Request a Teacher Account**

Fill in the following forms to request an account. The approval process may take up to one day. **Please do not use any portion of your real name in your username.**

**Create a username**

eduhk\_ken

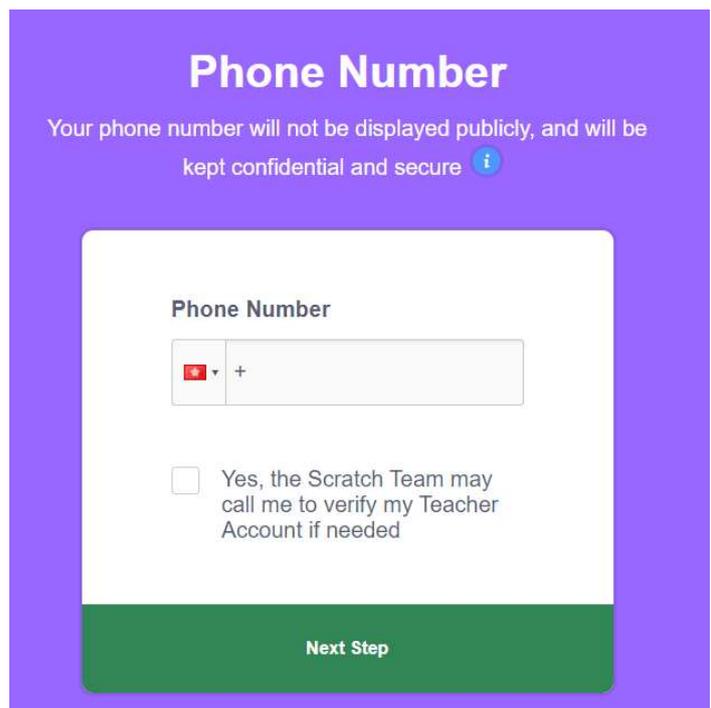
**Password**

.....

Show password

**Next Step**

- Enter Phone Number. You can provide school phone number while mobile phone is not a must.



**Phone Number**

Your phone number will not be displayed publicly, and will be kept confidential and secure [i](#)

**Phone Number**

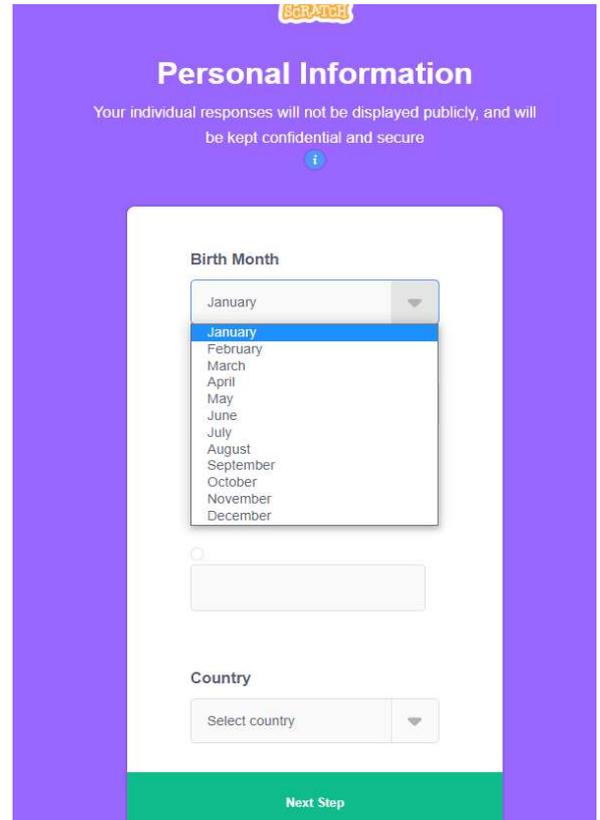
+ [Country Code] [Number]

Yes, the Scratch Team may call me to verify my Teacher Account if needed

**Next Step**

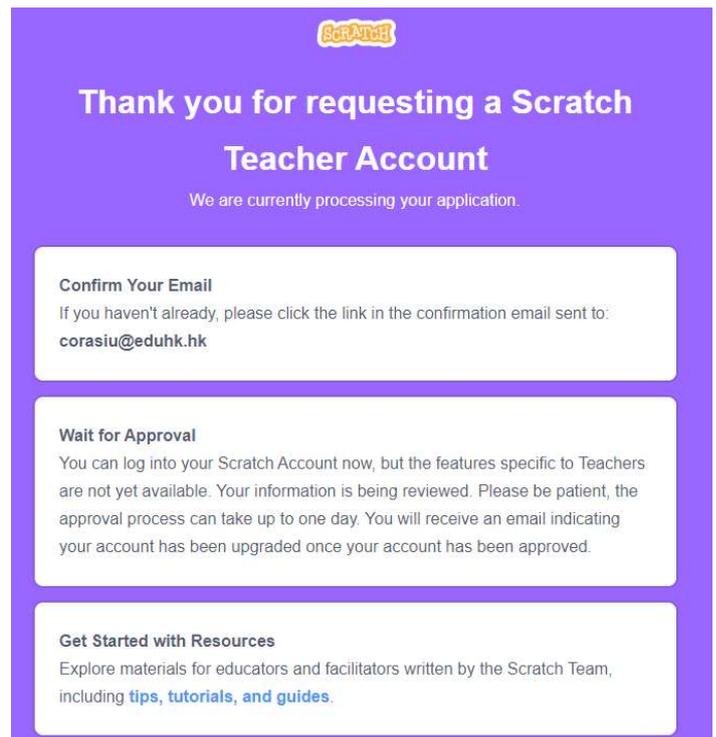
# Create Teacher Account

- Provide all the information required.



The screenshot shows the 'Personal Information' form on the Scratch website. At the top, the Scratch logo is visible. Below it, the title 'Personal Information' is centered, followed by a privacy notice: 'Your individual responses will not be displayed publicly, and will be kept confidential and secure'. An information icon is below the notice. The form itself is a white box with a light blue border. It contains a 'Birth Month' section with a dropdown menu currently showing 'January' and a list of months from January to December. Below this is an empty text input field. The 'Country' section has a dropdown menu with 'Select country' and a downward arrow. At the bottom of the form is a green button labeled 'Next Step'.

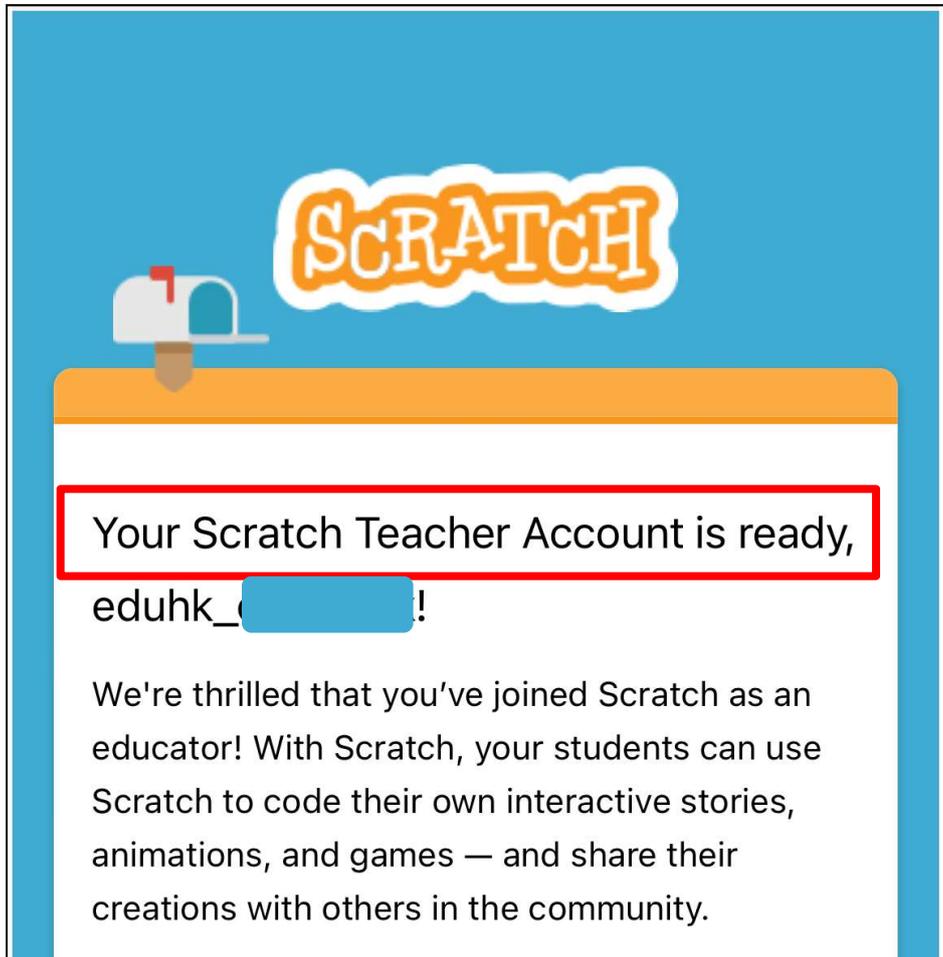
- Click Next Step until you reach the confirmation page.



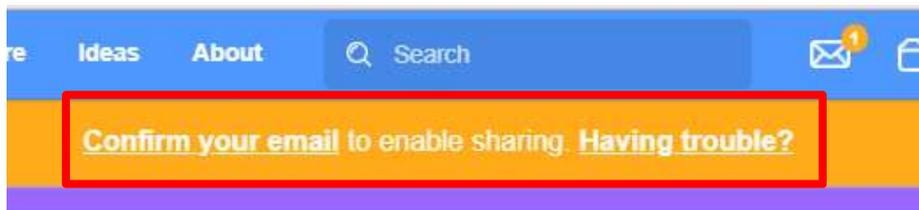
The screenshot shows the confirmation page on the Scratch website. At the top, the Scratch logo is visible. Below it, the title 'Thank you for requesting a Scratch Teacher Account' is centered. Underneath, a message states: 'We are currently processing your application.' The page is divided into three white boxes with light blue borders. The first box is titled 'Confirm Your Email' and contains the text: 'If you haven't already, please click the link in the confirmation email sent to: [corasiu@eduhk.hk](mailto:corasiu@eduhk.hk)'. The second box is titled 'Wait for Approval' and contains the text: 'You can log into your Scratch Account now, but the features specific to Teachers are not yet available. Your information is being reviewed. Please be patient, the approval process can take up to one day. You will receive an email indicating your account has been upgraded once your account has been approved.' The third box is titled 'Get Started with Resources' and contains the text: 'Explore materials for educators and facilitators written by the Scratch Team, including [tips](#), [tutorials](#), and [guides](#).'

# Create Teacher Account

- Check email (may go to SPAM) and activate.



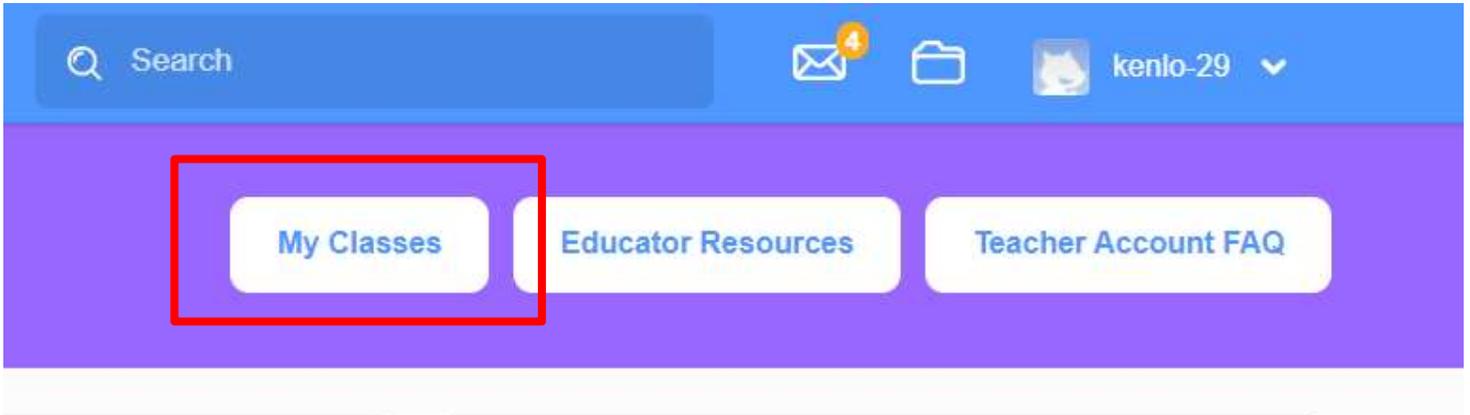
- Click Confirm your email if you did not get the email.



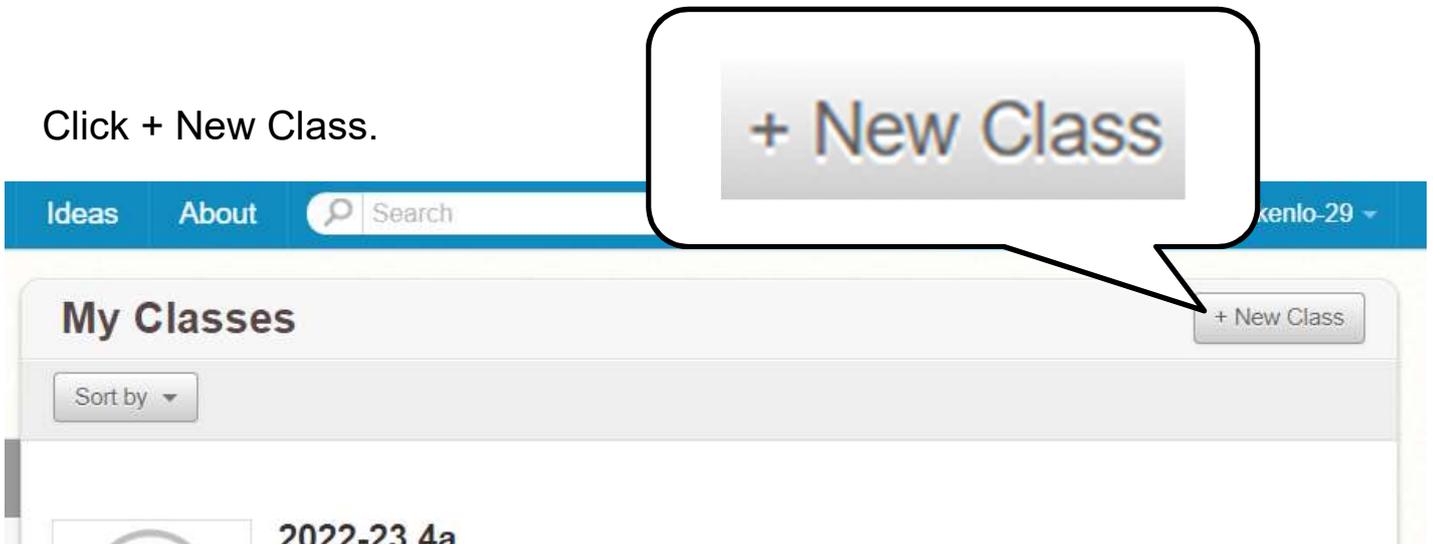
Teacher Account Created

# Create Classes

- Log on, Click My Classes.



- Click + New Class.



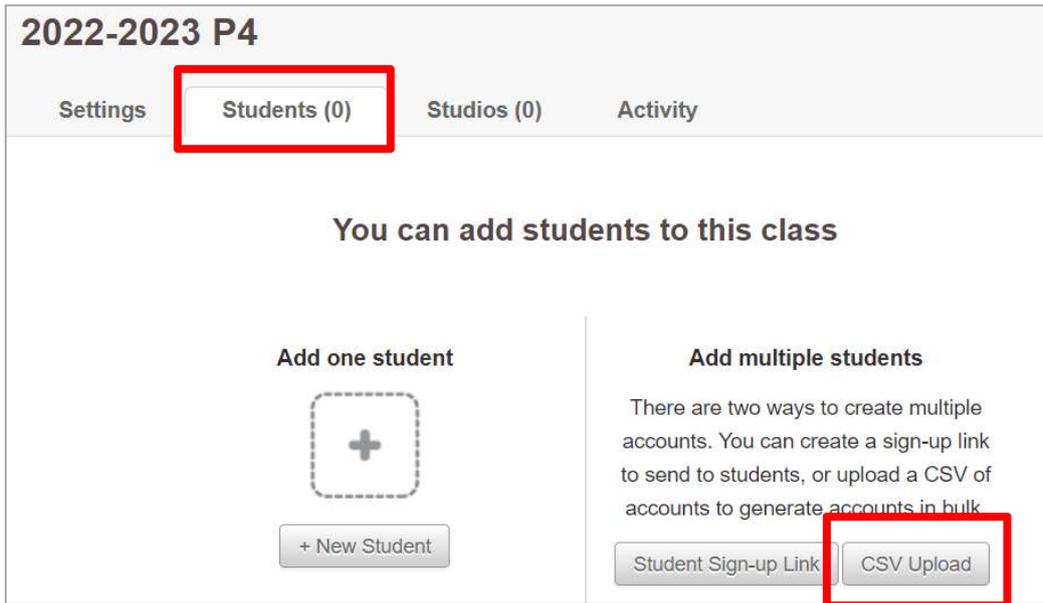
- Fill in Class Name and Class Description, then Add Classes.

A screenshot of a modal window titled 'Add a New Class' with a close button (X) in the top right corner. The form contains two input fields: 'Class Name' with the text '2022-2023 P4' and 'Class Description' with the text 'Primary 4 in 2022-2023'. At the bottom of the form are two buttons: 'Add Class' (highlighted with a red border) and 'Cancel'.

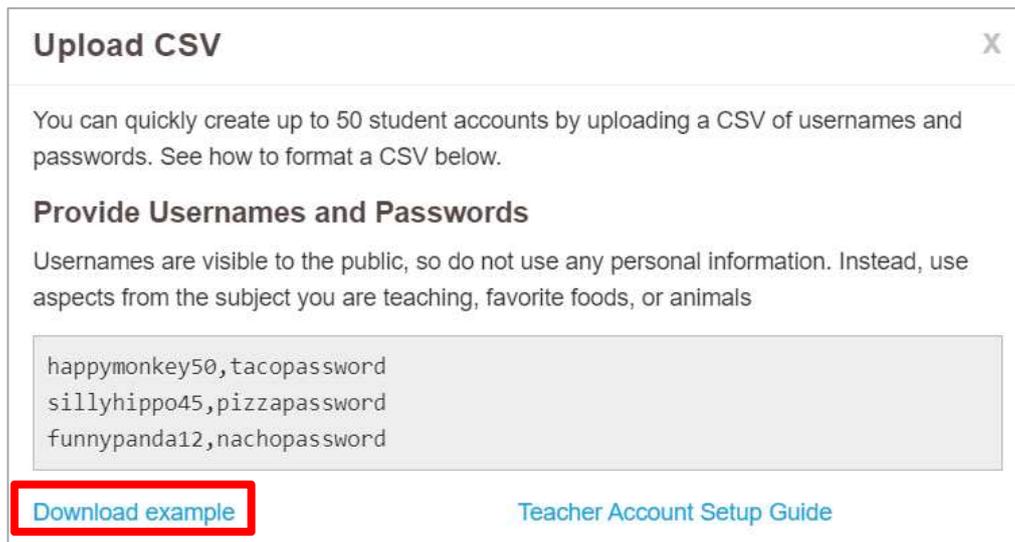
  
Class Created

# Create Students

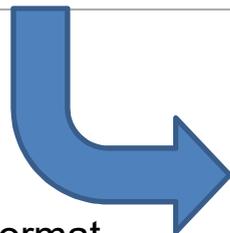
- Go to Students tap. Press **CSV Upload** button.



- Download example before creating students data: Usernames & Passwords



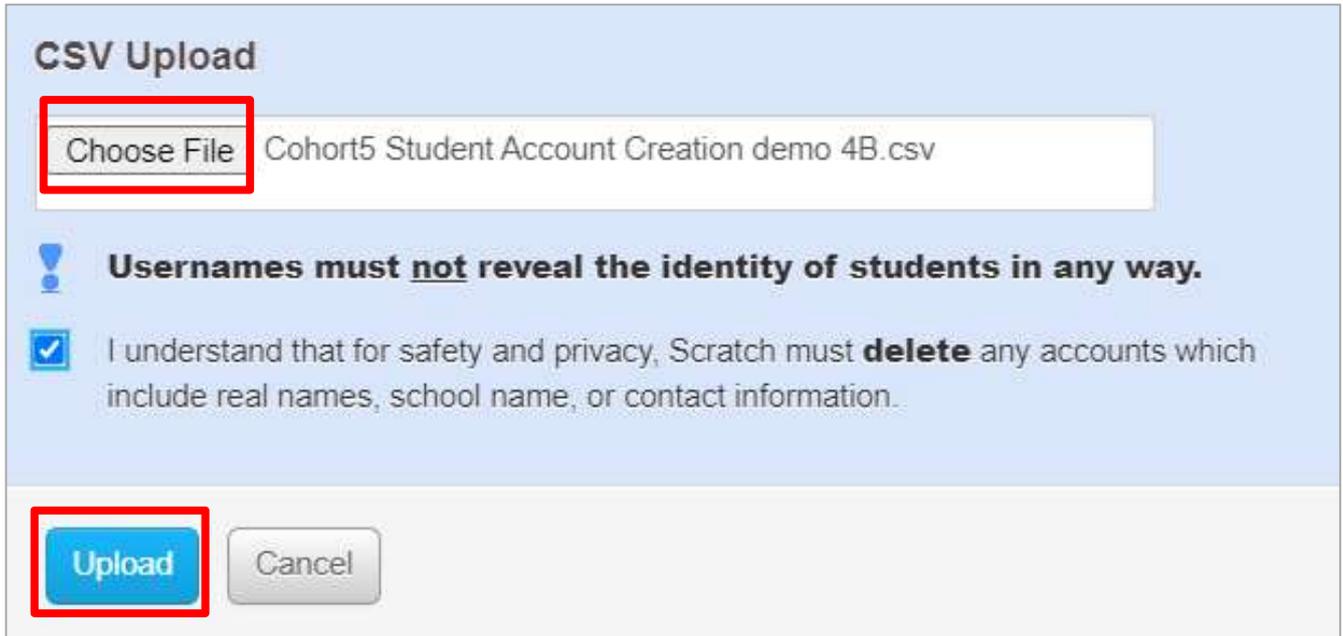
- Save file as .csv format.
- Max=50 students for one file.



	A	B
1	EDUHK_4B_001	4B01Pass116
2	EDUHK_4B_002	4B02Pass17
3	EDUHK_4B_003	4B03Pass420
4	EDUHK_4B_004	4B04Pass29

# Create Students

- Choose File and then Upload.



**CSV Upload**

Choose File Cohort5 Student Account Creation demo 4B.csv

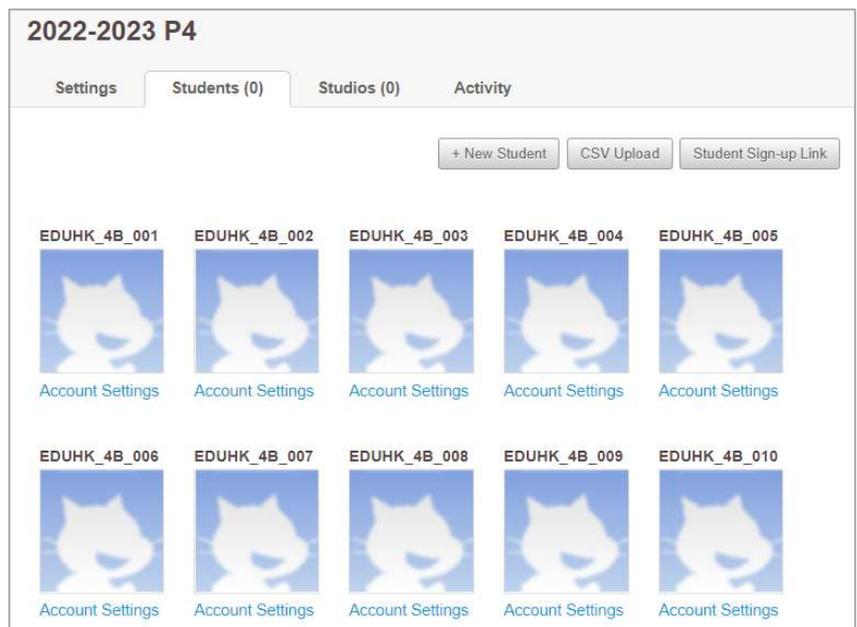
**!** Usernames must not reveal the identity of students in any way.

I understand that for safety and privacy, Scratch must **delete** any accounts which include real names, school name, or contact information.

Upload Cancel

- Repeat the steps until all classes / students created.

- Limitation:  
There is no rollover  
for Scratch Classes.



2022-2023 P4

Settings Students (0) Studios (0) Activity

+ New Student CSV Upload Student Sign-up Link

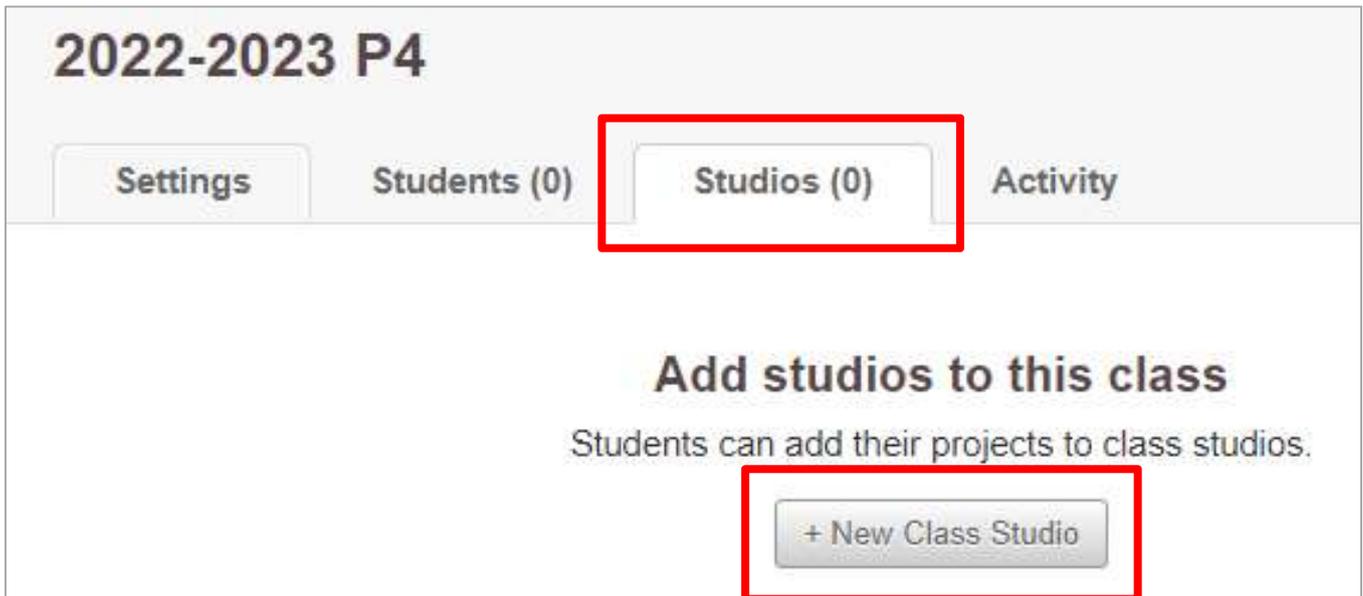
EDUHK_4B_001	EDUHK_4B_002	EDUHK_4B_003	EDUHK_4B_004	EDUHK_4B_005
				
Account Settings	Account Settings	Account Settings	Account Settings	Account Settings
EDUHK_4B_006	EDUHK_4B_007	EDUHK_4B_008	EDUHK_4B_009	EDUHK_4B_010
				
Account Settings	Account Settings	Account Settings	Account Settings	Account Settings



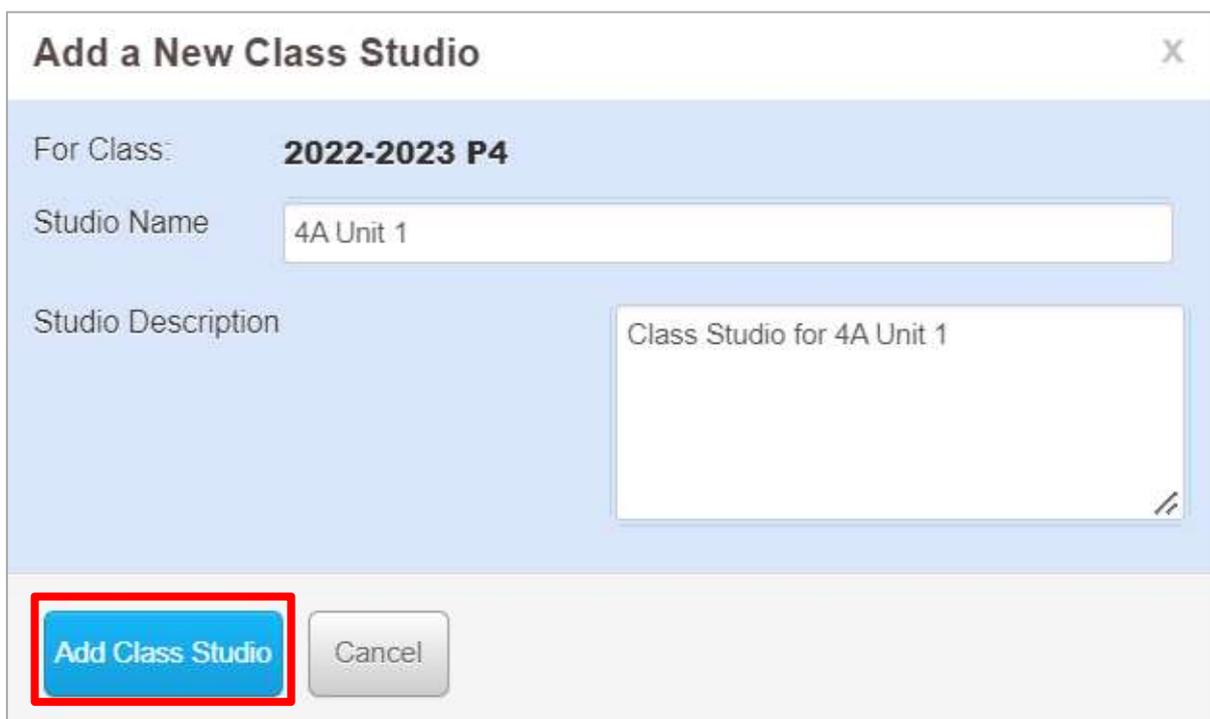
Students Created

# Create Studio

- Go to **Studios** tap, Click **+ New Class Studio**.



- Fill in Studio information, then **Add Class Studio**
- Remark: You may discuss within your school for the Studios structure. E.g. By units, classes or grades?



The screenshot shows the 'Add a New Class Studio' form. The title bar includes the text 'Add a New Class Studio' and a close button 'X'. The form is set for the class '2022-2023 P4'. It contains the following fields:

- For Class:** 2022-2023 P4
- Studio Name:** 4A Unit 1
- Studio Description:** Class Studio for 4A Unit 1

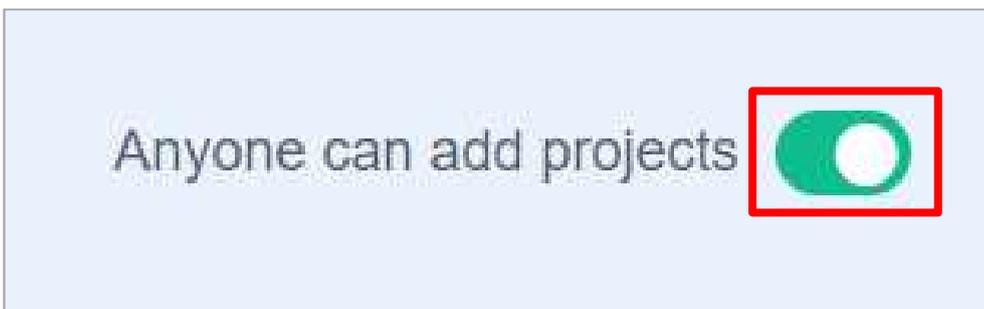
At the bottom of the form, there are two buttons: 'Add Class Studio' (highlighted with a red box) and 'Cancel'.

# Create Studio

- Click the Studio name.



- Switch on Anyone can add projects.



Studio Created

# For Students Sign-in

- Students sign in with the information teachers provided.

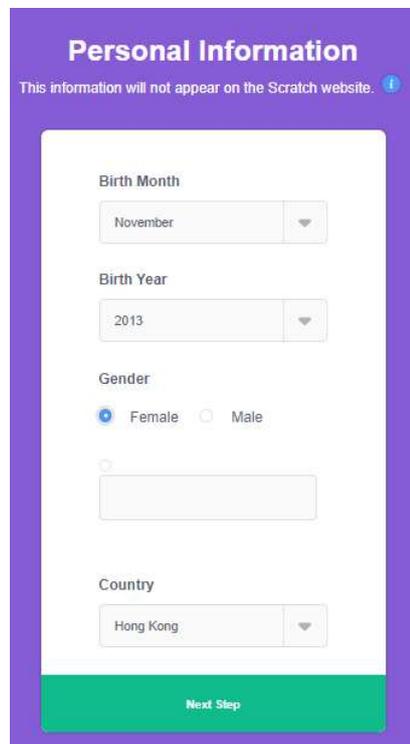


The image shows the Scratch sign-in interface. At the top, there are two buttons: "Join Scratch" and "Sign in". The "Sign in" button is highlighted with a red box. Below these buttons, there are two input fields: "Username" and "Password". The "Username" field contains the text "EDUHK\_4B\_001". The "Password" field contains a series of dots. Below the input fields, there is a "Sign in" button, also highlighted with a red box, and a link that says "Need Help?".

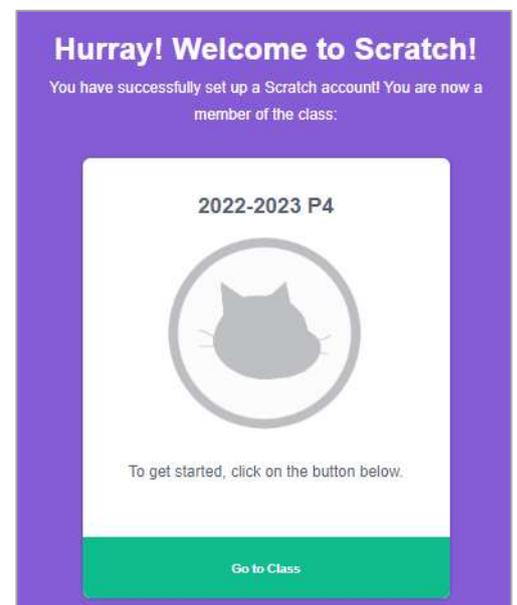
- (For the first logon only) Click **Get Started**, fill in related information. **Next Step** until done.



The image shows the Scratch invitation screen. At the top, there is the Scratch logo and the text "EDUHK\_4B\_001". Below this, it says "you have been invited to join the class:". In the center, there is a Scratch cat icon and the text "2022-2023 P4". Below the icon, it says "invited by CTE\_corasiu". At the bottom, there is a green button that says "Get Started". At the very bottom, there is a link that says "Not you? Log in as another user".



The image shows the Scratch personal information form. At the top, there is the text "Personal Information" and a note that says "This information will not appear on the Scratch website.". Below this, there are several input fields: "Birth Month" (set to November), "Birth Year" (set to 2013), "Gender" (with radio buttons for Female and Male, where Female is selected), a text input field, and "Country" (set to Hong Kong). At the bottom, there is a green button that says "Next Step".



The image shows the Scratch welcome screen. At the top, there is the text "Hurray! Welcome to Scratch!" and a message that says "You have successfully set up a Scratch account! You are now a member of the class:". In the center, there is a Scratch cat icon and the text "2022-2023 P4". Below the icon, it says "To get started, click on the button below.". At the bottom, there is a green button that says "Go to Class".



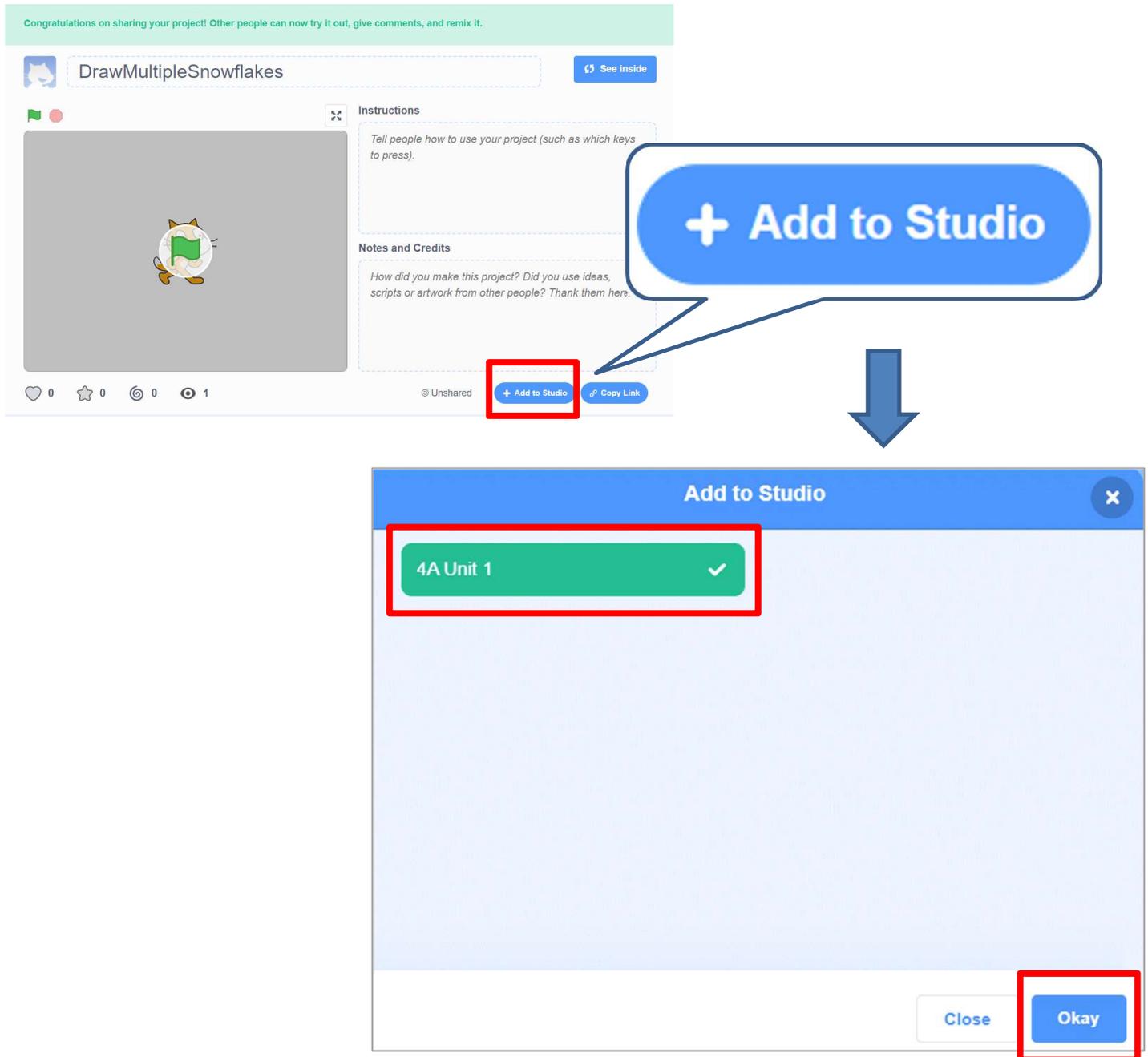
Students Signed in

# Sharing Projects

- Rename project name (e.g. Add “4B\_01\_”) and click **Share** button.



- Click **+ Add to Studio** button, select the Studio, then **Okay**.



# Sharing Projects

- Now, teacher and classmates can see the students' projects shown on the Studio.

The screenshot shows a Scratch Studio interface for a class named "4A Unit 1". On the left sidebar, there is a "Follow Studio" button, a circular thumbnail of the Scratch cat, and an "Edit Thumbnail" link. Below that is a section for "Class Studio for 4A Unit 1". The main area is titled "Projects" and includes a toggle for "Anyone can add projects" which is turned on. Below the title is an "Add Projects" section with a text input field containing the URL "https://scratch.mit.edu/projects/xxxx", an "Add by URL" button, and a "Browse Projects" button. A project thumbnail featuring the Scratch cat is highlighted with a red border. The project title is "4B\_001\_MyFirstProgram" and the creator's name is "EDUHK\_4B\_001". At the top of the main area, there are navigation buttons for "Projects (1)", "Comments (0)", "Curators", and "Activity".



Project Shared

# Unit 2: Exploring Under the Sea

## Teacher Guide

### Content

<b>Teaching Plan</b>	P2
<b>Pre-lesson Worksheet</b>	T2-1
<b>Lesson 1</b>	
To Play	T2-2
To Think	T2-4
To Code	
(A) Add Music to the Backdrop	T2-6
(B) Change Looks of the Sprite	T2-8
(C) Make Sprite Swim	T2-10
Understanding Sequence	T2-11
<b>Lesson 2</b>	
To Create	T2-12
To Reflect	T2-15
Review Questions	T2-17
Revision on Key Features	T2-19
Revision on Key Concepts & Practices	T2-20
<b>Appendix - Operation Manual</b>	T2-23
<b>Program Codes</b>	T2-30

## Unit 2: Exploring Under the Sea Teaching Plan

### Prior Knowledge

Students understand the basics of code and run a simple program in the Scratch programming environment.

Students know the skills of adding sound to a sprite, adding simple motion of a sprite, and changing costumes for a sprite.

### Learning Objectives

1. Learn features in Scratch, including playing sound in a backdrop / sprite, movement of a sprite, changing costumes of a sprite;
2. Understand computational thinking concepts and practices of sequences and iteration, parallelism, reuse and remix programs/codes, testing and debugging;
3. Develop interest in programming, show perseverance and positivity in testing and debugging, and show motivation to be creative;
4. Foster students' creativity by staging their coding artefacts – Exploring Under the Sea, and sharing with their classmates, teachers and parents.

### Key Learning Elements

#### Computational Thinking Concepts and Practices:

Key Learning Elements	Items
Algorithm	Basic Programming Constructs: Sequence, Iteration Coding Concepts and Practices: Design, Reuse and Remix programs / codes, Testing and Debugging

### Coding Skills:

1. Play sound in a backdrop / sprite, add motions of a sprite, change costumes of a sprite;
2. Use iteration - forever and repeat blocks;
3. Reuse and remix programs / codes;
4. Apply testing and debugging in completing each task of this unit.

### Others (including Attitude):

1. Develop interest in programming;
2. Show perseverance and positivity in testing and debugging;
3. Inspire students to be creative and innovative to enhance their projects.

**Lesson Plan:** This unit consists of 2 lessons of 35 minutes.

**Pre-task for Students:** Review the coding environment. Explore adding a backdrop and adding sound to the backdrop.

### Lesson 1

Time	Activity
5 mins	<p><b>To Play:</b> Help students gain an initial understanding of the program</p> <ol style="list-style-type: none"><li>1. Ask students what they see in the demo project, including sprite, sound and backdrop.</li><li>2. Play the Scratch project, Under the Sea: <a href="https://scratch.mit.edu/projects/722781437">https://scratch.mit.edu/projects/722781437</a>.</li><li>3. Review with students about the demo project, guide students to use the mind map to observe in detail, including adding sound to a sprite, changing costumes for a sprite, movement of a sprite, and adding sound to a backdrop.</li></ol>
10 mins	<p><b>To Think</b></p> <ol style="list-style-type: none"><li>1. Guide students to think about what things they need to do to complete this project. (Use flow diagram for problem solving, use mind map to show and design the algorism.)</li><li>2. Support students on developing problem solving skills through decomposing the programming problems into sub-tasks.<ol style="list-style-type: none"><li>1) Add Music to Backdrop</li><li>2) Change Looks of the Sprite</li><li>3) Make Sprite Swim</li></ol></li></ol>
15 mins	<p><b>To Code (Under the Sea project)</b></p> <ol style="list-style-type: none"><li>1. Ask students to follow the instructions in the Student Guide and work independently on their own project.</li><li>2. Remind students that they need to test and debug when they complete each sub-task.</li><li>3. Support students to address learner diversity.</li><li>4. Students will be invited to share their “Under the Sea” project in Lesson 2.</li></ol>
5 mins	<p><b>Discussion</b></p> <ol style="list-style-type: none"><li>1. Explain that there is no specific sequence for the 3 sub-tasks in this project.</li><li>2. Support students to complete the coding process with their own idea. (For example, they can add sound effect at the end but not at the beginning.)</li></ol>

## Lesson 2

Time	Activity
5 mins	<p><b>Introduction of Task 2</b></p> <ol style="list-style-type: none"><li>1. Today’s agenda: remixing code and explaining parallelism with three sprites.</li><li>2. Have students pair up and show their peers their “Under the Sea” project created in Lesson 1.</li><li>3. Encourage conversation on what looks cool and discussion on how specific movements were coded in the project.</li><li>4. Ask students to remix something cool that they saw in someone else’s project into their own projects.</li></ol>
15 mins	<p><b>To Create: Task 2</b></p> <ol style="list-style-type: none"><li>1. Review the Scratch features in Lesson 1: playing sound in a backdrop / sprite, movement of a sprite, changing costumes of a sprite.</li><li>2. Add two more sprites.</li><li>3. Reuse and remix the code in Lesson 1 to make these two new sprites swim.</li><li>4. Add more features to enhance for fun, e.g. sound effect, changing backdrop.</li><li>5. Students can use the mind map to plan their designs.</li><li>6. (Optional) Teachers can invite students to use a mind map to design a new project, e.g. a bird flies in the sky.</li></ol>
10 mins	<p><b>To Reflect:</b></p> <p><b>Share the Projects and Provide Constructive Feedback on Program Design</b></p> <ol style="list-style-type: none"><li>1. Create a studio and give students the studio URL. Ask students to save and submit their projects to the teacher’s Studio.</li><li>2. Have one or two students share their project with the whole class. Feedback to be given by peer and teachers.</li><li>3. Students should share, think of ways to improve / enhance their program as well as the aesthetic values.</li></ol> <p><b>Review of Student Learning</b></p> <ol style="list-style-type: none"><li>1. Review on the features of Scratch, and key concepts and practices learnt in the unit.</li><li>2. Ask students to complete the review questions, appropriate feedback should be given by teachers.</li></ol>

# Exploring Under the Sea

❑ Let's review the coding environment in Scratch by putting the letters A, B, C, D and E in the correct boxes.

<b>A.</b> Block Palette		<b>B.</b> Adding / Changing costumes
<b>C.</b> Adding sound	<b>D.</b> Adding / Changing Sprite	<b>E.</b> Adding / Changing Backdrop

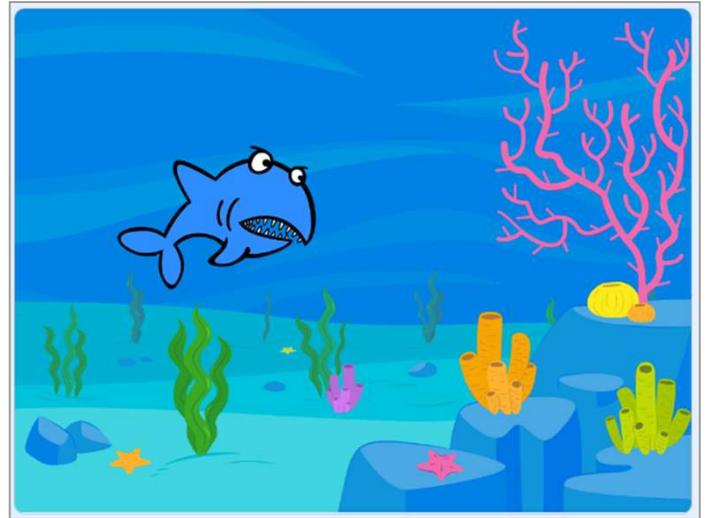
The diagram shows the Scratch interface with several components highlighted and labeled:

- Code Area:** An orange dashed box labeled "Code Area" with the description "An area for storing blocks that run the project." It encompasses the central workspace where a "when clicked" block is visible.
- Stage:** A green dashed box labeled "Stage" with the description "An area where the sprites are displayed and perform their actions." It encompasses the area where the Cat sprite is currently displayed.
- Block Palette (A):** A red box labeled "A" points to the left sidebar containing various coding blocks.
- Costumes (B):** A blue box labeled "B" points to the "Costumes" tab in the top toolbar.
- Sounds (C):** A blue box labeled "C" points to the "Sounds" tab in the top toolbar.
- Sprite (D):** A blue box labeled "D" points to the "Sprite" panel at the bottom, which shows the "Cat" sprite selected.
- Backdrop (E):** A blue box labeled "E" points to the "Backdrops" panel at the bottom, which shows a single backdrop.

# Exploring Under the Sea

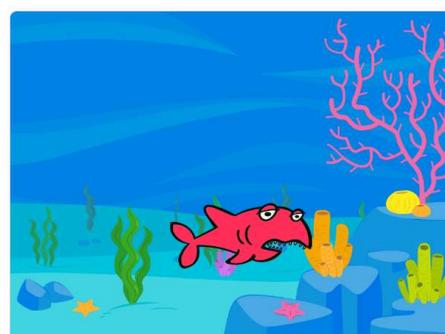
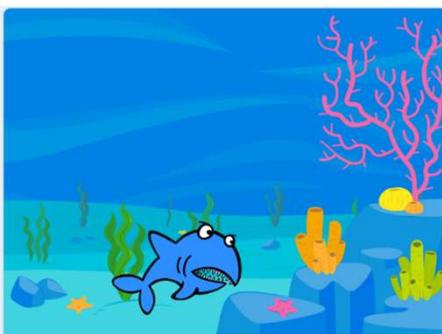
Let's explore under the sea!

Let's take what we've learnt so far with Scratch and make our sprites move! You will explore under the sea where you will add music, sound and movements.



## To Play

- ❑ **Play** the Scratch project, Under the Sea: <https://scratch.mit.edu/projects/722781437>
- ❑ **Click the green flag** and see what will happen.
  - How many **sprites** do you see?
  - How many **looks** of the sprite?
  - Where does the **shark swim** to?
  - Do you hear any **sound effect**? Is it from the shark or from the sea?



# Exploring Under the Sea

What did you observe when playing? Try to tick (✓) the correct box as follows:  
(You may choose more than one answer.)

1. How many sprites appear in the game?

- 1
- 2
- 3

2. Which color(s) did the shark's costume change?

- Red
- Yellow
- Black
- White
- Purple

3. Any sound effects in the game?

- Ripples sound
- Rain sound
- Bite sound of shark sprite
- Cheers
- No sound effect

4. Which background(s) appeared in the project?

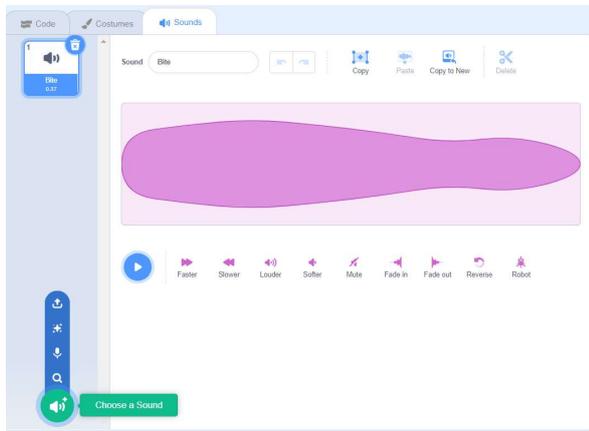
- Forest
- Under the Sea
- Swimming Pool
- Playground

# Exploring Under the Sea

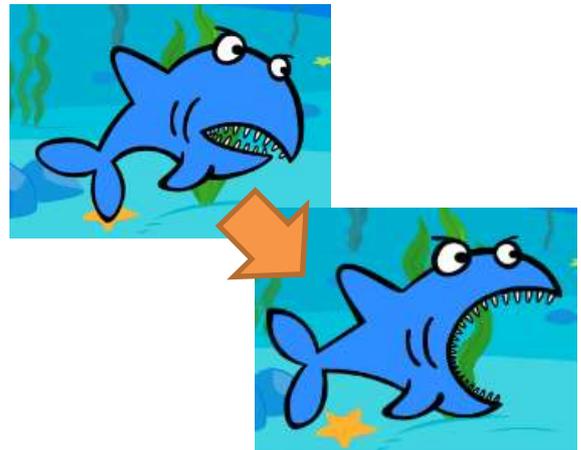
## To Think

- After playing the Scratch project, we need to think how to code.
- Let's think about which action you should take first? Write down your sequence in the boxes:

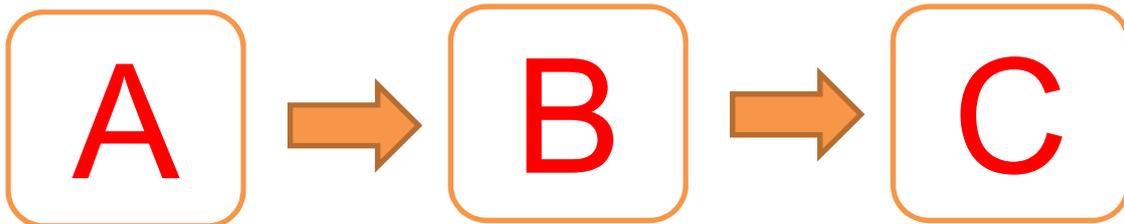
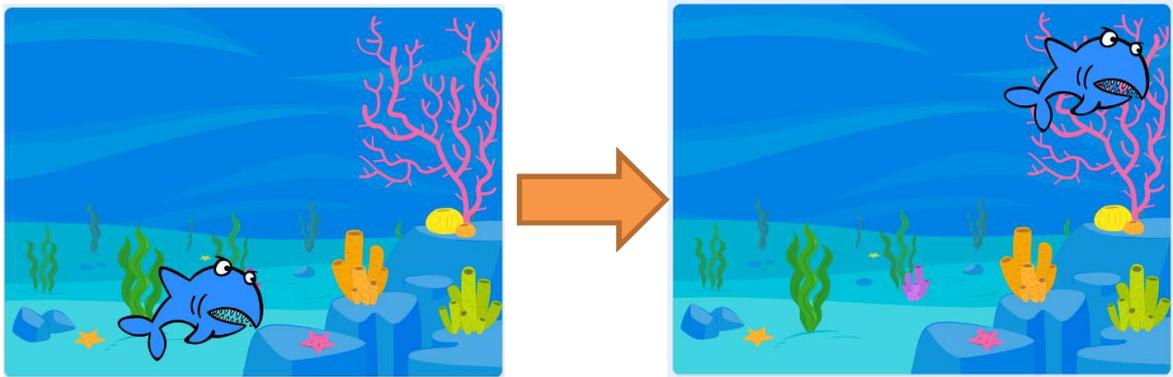
A. Add Music to Backdrop



B. Change Looks of the Sprite



C. Make Sprite Swim



Suggested Answer

Teacher may follow the instructions of the programming sequences (A>B>C) with students. In the next lesson, they can design their own projects and think about the programming sequences themselves.

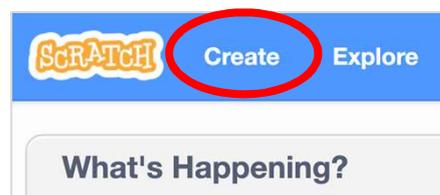
# Exploring Under the Sea

## Start Here

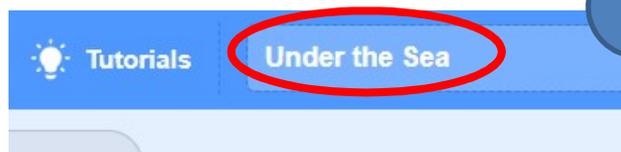
1. Sign into your account at <https://scratch.mit.edu/>.



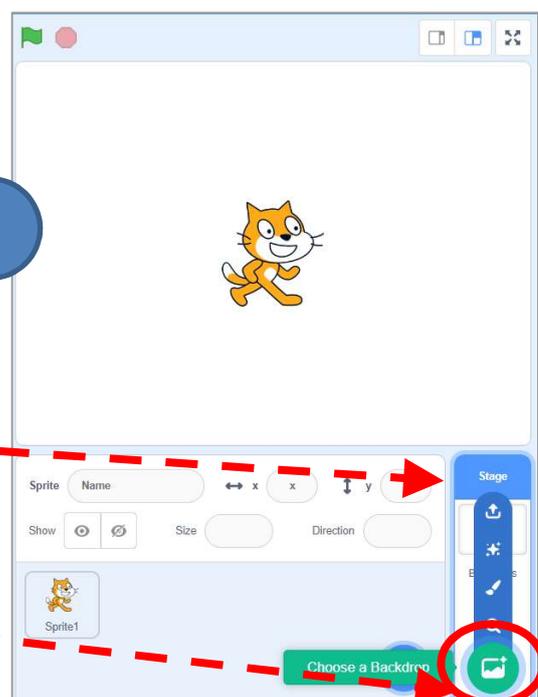
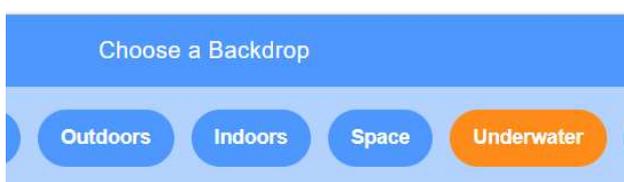
2. Go to **“Create”** to start a new project.



3. Name it **Under the Sea**.



4. Choose an underwater **backdrop** for your stage.



Click on the Stage

Click on the “Choose a Backdrop” icon

# Exploring Under the Sea

## To Code: (A) Add Music to the Backdrop

To Think:

You have learnt to add sound to a sprite, but how to add to the backdrop instead of a sprite?

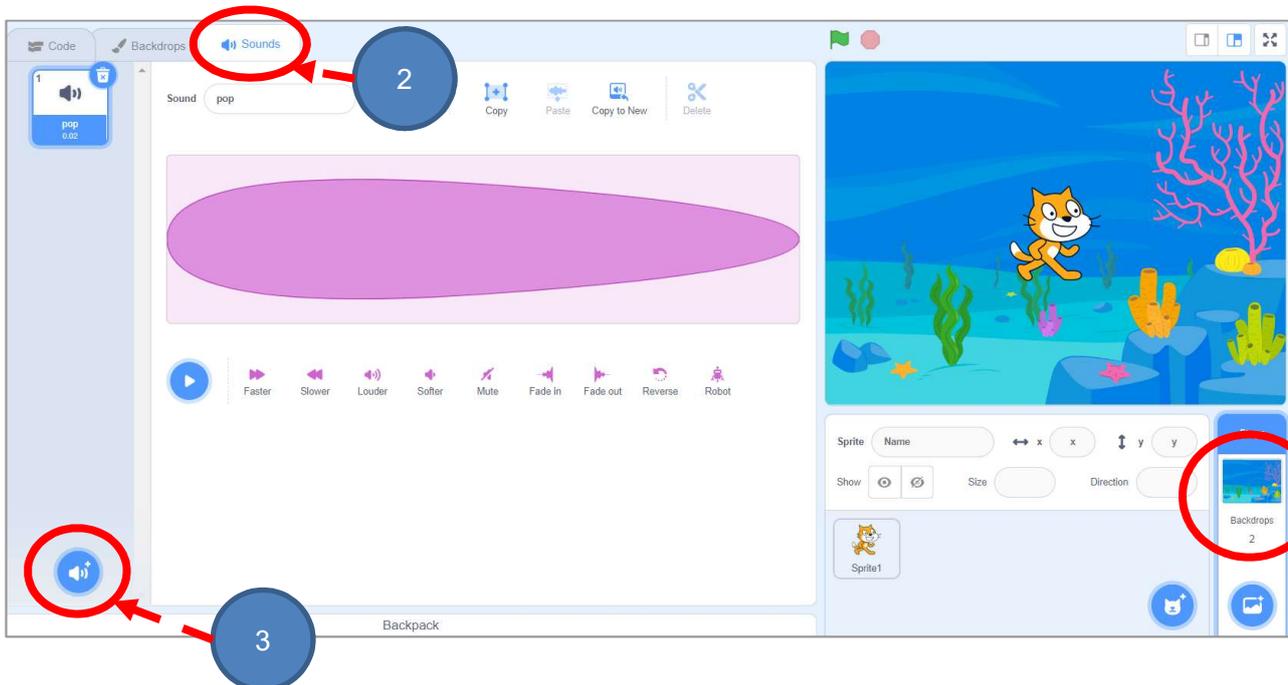
Let's try and code on your own. You may refer to below hints if needed.

- Choose some background music under the sea.

1. Click on the "Stage".

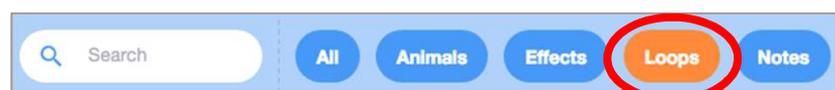
1

2. Click on the "Sounds" tab.



3. Click on the "Choose a Sound" icon at the bottom left to select a sound.

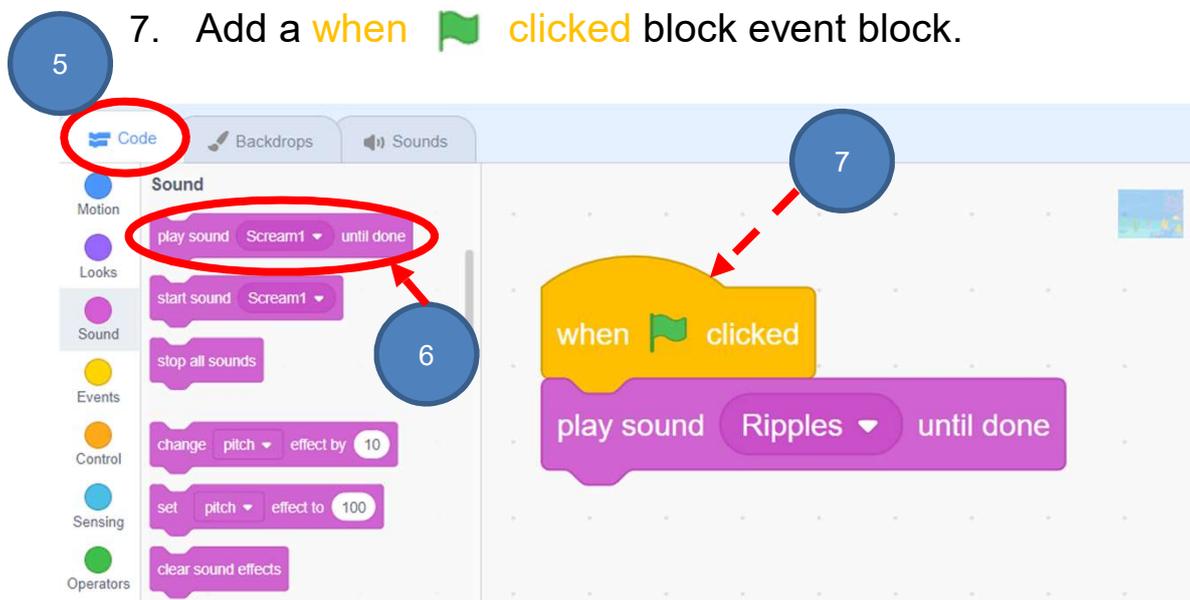
4. Choose a music sound (under "Loops" category is recommended) to play.



# Exploring Under the Sea

## To Code: (A) Add Music to the Backdrop

- ❑ Now add your music to the stage.
  5. Click on **Code** tab.
  6. Click **Sound** drawer, drag out the relevant block to play sound until done.
  7. Add a **when**  **clicked** block event block.



### Testing and Debugging

Click the  above the Stage to see what happens.

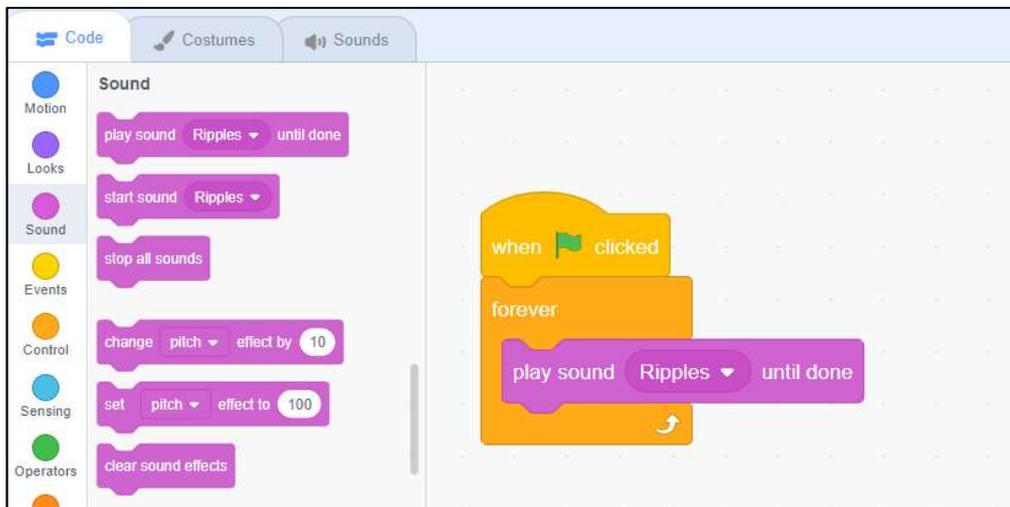
Can you hear any sound? Is the music playing non-stop until the “Stop” button is clicked?

Try adding the “forever” block in “Control” to keep the music playing when the green flag is clicked.



# Exploring Under the Sea

Teacher can demonstrate the forever block after students' testing and debugging.



## To Code: (B) Change Looks of the Sprite

1. Use what you learnt in Unit 1, now choose a good **sprite** for swimming under the sea! Add multiple **costumes** for your sprite. [\*You can delete the original Scratch Cat costumes.]
2. Switch costume to certain look or **next costume**. You can also switch costumes to make the fish looks like opening and closing its mouth.
3. Remember **which block** should we add to make sure we can see each costume when the green flag is clicked?



4. How many times is the **costume changed**? Do you observe any pattern?  
**3 times. The shark will change a costume of different colour every 1 second and move to the right a bit.**

Let's try and code.

See Appendix  
P.24

# Exploring Under the Sea

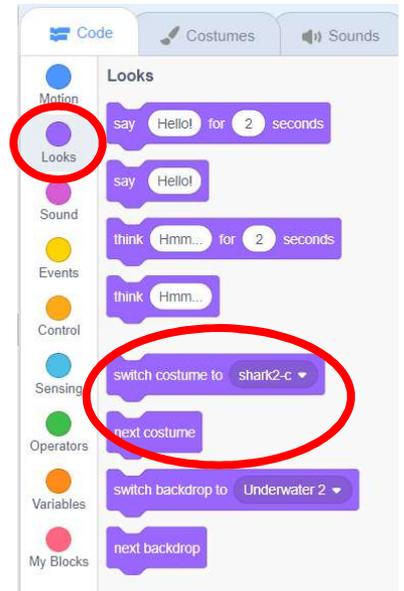
## To Code: (B) Change Looks of the Sprite

Remember how to switch costumes?

1. Click **Looks** Drawer, you can **switch costume** to certain look or **next costume**.

```

switch costume to shark2-a
wait 1 seconds
switch costume to shark2-b
wait 1 seconds
switch costume to shark2-c
wait 1 seconds
    
```



2. Time how long the sprite waits between switching costumes with the **“wait”** block.

**1 second.** Teachers can ask students to input different waiting time, e.g., 5 seconds, 10 seconds, to see whether the movement of the shark is natural.

Which codes are repeated? Can you identify the pattern?  
 Can you use Repeat block to switch costumes?  
 Please refer to the codes:  
 1) next costume, 2) change color 3) wait 4) move

```

when green flag clicked
go to x: -141 y: -66
repeat 3
  next costume
  change color effect by 25
  wait 1 seconds
  move 50 steps
    
```



### Testing and Debugging

Time to test! Simply click the green flag to see the changes you made.



Add Music to the Backdrop



Change Looks of the Sprite



### Knowledge builds up: Iteration – repeat

Iteration is repeating a process in order to produce a sequence of outcomes. Other than “forever” block, “repeat” block can also trigger iteration in Scratch.

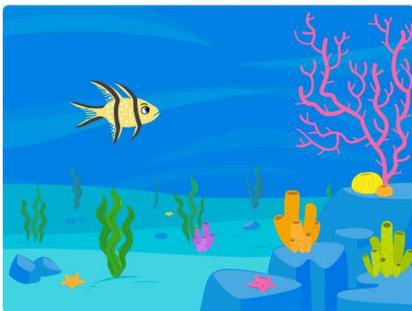
# Exploring Under the Sea

See Appendix  
P.25

## To Code: (C) Make Sprite Swim

How does the sprite move? Now add blocks to make the sprite move (swim). You can explore [move](#), [turn](#), [go to](#) and [glide](#) blocks.

Let's try and code.



### Testing and Debugging

When you complete, click the green flag to test it and see if everything works fine.



### Knowledge builds up: Sequence

Put the following morning routines into a reasonable sequence by filling 1, 2, 3...

- |   |                           |
|---|---------------------------|
| ( 4 ) Put on your pants / skirt.                | ( 7 ) Grab your backpack. |
| ( 1 ) Get out of bed.                           | ( 6 ) Put on your shoes.  |
| ( 5 ) Eat breakfast.                            | ( 2 ) Brush your teeth.   |
| ( 8 ) Get on bus/get in car to drive to school. | ( 3 ) Put on your shirt.  |

\*The first step must be get out of bed and the last step must be get on bus / get in car. Other steps can be put in different order as appropriate.

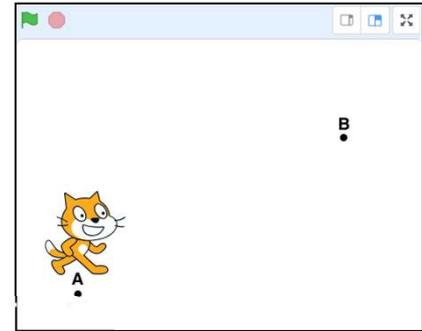
# Exploring Under the Sea

## To Code: Understanding Sequence

Do you think sequence is important?

Let's try and explore.

Which of the following sequences of commands (sets of blocks) will make the Scratch cat move from point A to point B on the stage? Please tick.



```

when green flag clicked
  move 150 steps
  turn 90 degrees
  move 50 steps
    
```



```

when green flag clicked
  turn 90 degrees
  move 50 steps
  move 150 steps
    
```



### Knowledge builds up: Sequence

Sequence is a key concept in programming. It is the order in which the programming statements are executed. A wrong order would lead to incorrect programming results. For example, there is a specific sequence of dialogues for a meaningful conversation to go on.

### Being incremental and iterative



Add Music to the Backdrop



Change Looks of the Sprite



Make Sprite Swim



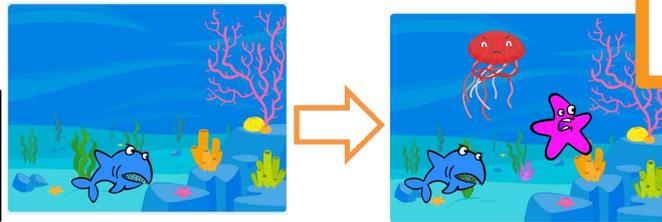
### Knowledge builds up: Being incremental and iterative

It is a programming approach. It is iterative because it plans for the work of one iteration (Add music) and then another iteration (Change looks). It is incremental because each iteration will be built on the previous one with improvement (completed the first two steps, then enhance the program to make sprite swim) until the programming task is completed. (Examples can be found in Revision on Key Concepts & Practices on P.22)

# Exploring Under the Sea

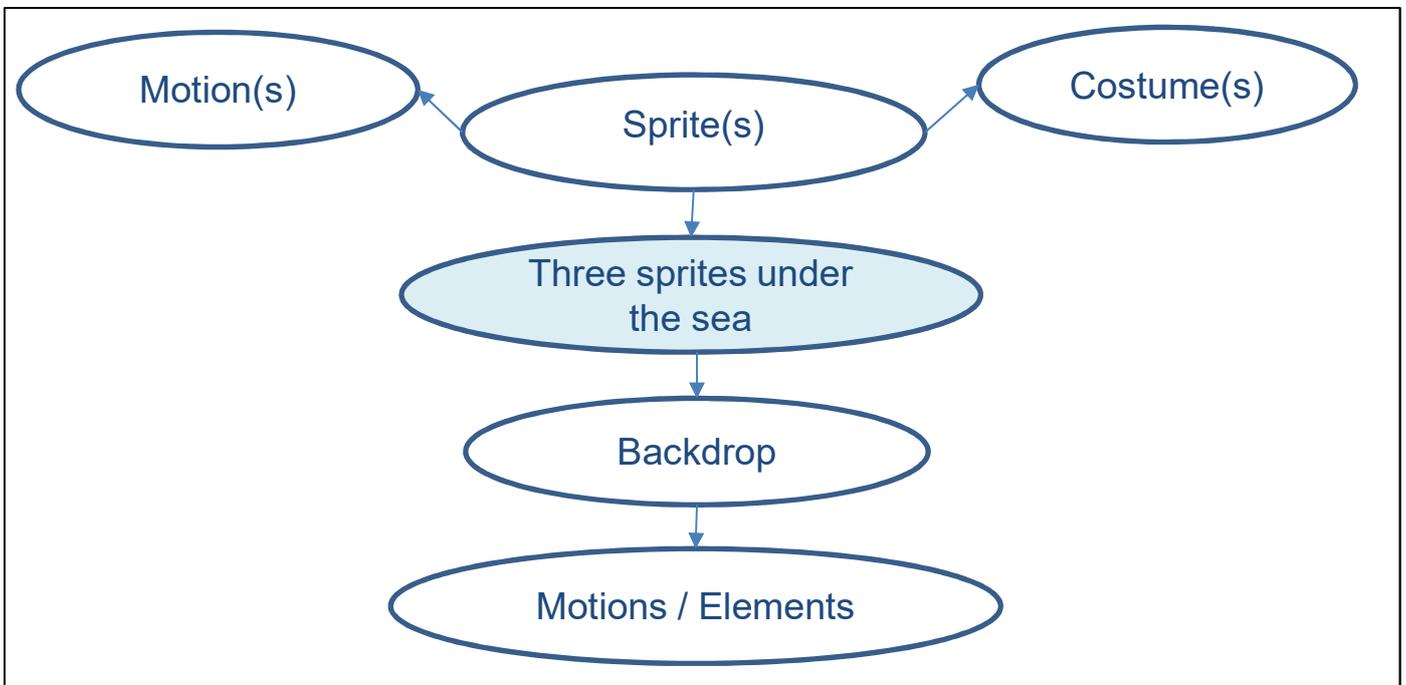
## To Create

Let's make more friends under the sea!



See Appendix P.26-27

- Task 2: Make your design, e.g. add two more sprites, and more different effects! Take a look on the following mindmap, think about what you want to add (e.g. motions and costumes of sprites etc.)



- How will you plan the programming sequences for your own design? Please fill in the boxes with A, B or C.

A. Add Sprites	B. Change Costumes	C. Add Sound
----------------	--------------------	--------------



Teachers should remind students that they must “Add Sprites” before going for the other programming. There is no specific sequences for task B and C.

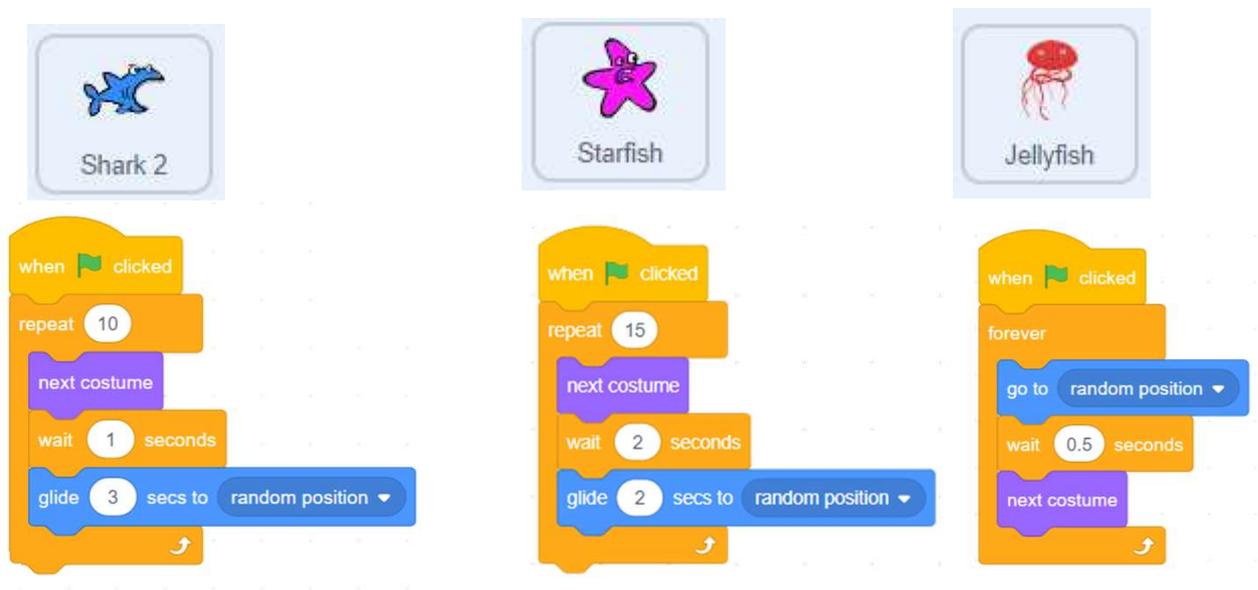
# Exploring Under the Sea

## To Create

See Appendix  
P.28-29

In Task 2, we have three sprites, swimming and having fun under the sea.

When we click the green flag, all the sprites start to move. This is called **parallelism** in programming.



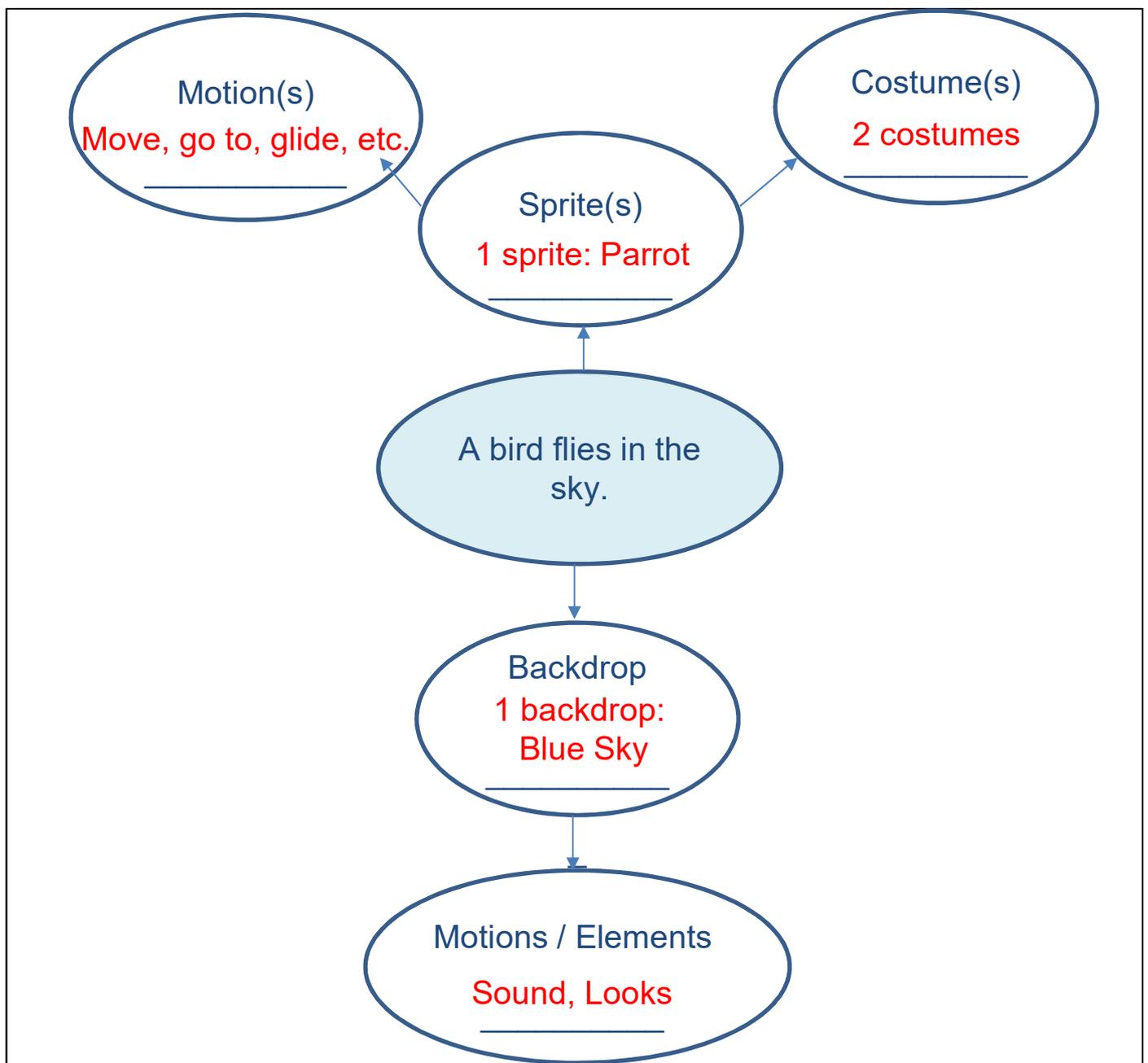
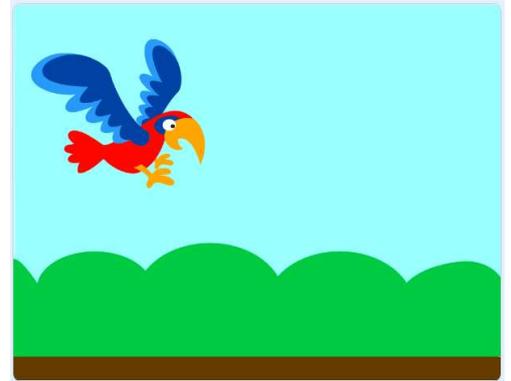
### **Knowledge builds up: Parallelism**

Scratch uses parallelism to allow more than one events to take place at the same time. For example, Scratch allows more than one character to perform an action at the same time in a Scratch program.

(Based on the above code blocks, when green flag clicked, the above three sprites will perform actions together.)

# Exploring Under the Sea To Create

- Can you make use of what you learnt in this unit?
- Let's draw something new.  
e.g. A bird flies in the sky.
- You can use a mind map to plan your design.



# Exploring Under the Sea

## To Reflect: Two Stars and a Wish Worksheet

Name of Project: \_\_\_\_\_ Name of Creator: \_\_\_\_\_

Please write down two things that you like about this project.



“Two Stars and a Wish” is a reflection strategy designed for student feedback as peer- and self-assessment. Teachers can guide students to give constructive feedback to their peers regarding their Scratch project - two positive (stars) and one hopeful (wish) reflection. Comments can be made on Scratch project’s idea, features and aesthetic aspects etc.



What is one thing you would like to add or change to make this project better?

---

---

---

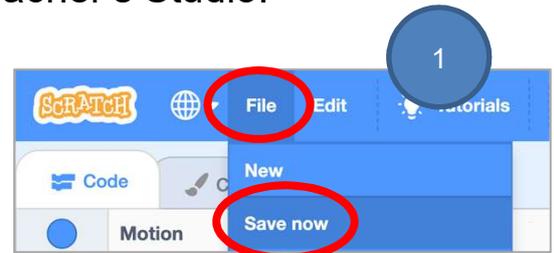


# Exploring Under the Sea

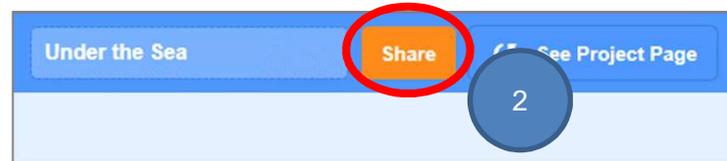
## Sharing to Studio and Provide Constructive Feedback on Program Design

When you finish, you will add your project to your teacher's Studio.

1. Save your project by clicking "Save now" under the File menu.



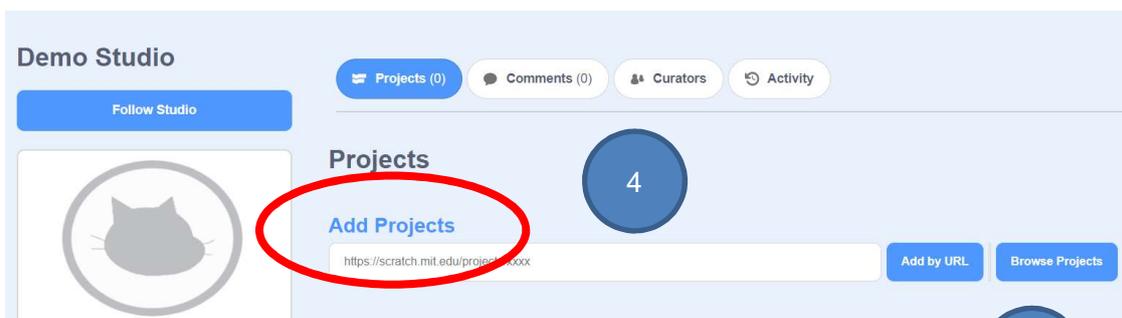
2. Click the orange "Share" button.



3. Go to your teacher's Studio (they will give you a URL).



4. In the "Add projects" column, you can Add by URL or Browse Projects.



5. If you choose Browse Project, then you will see all your shared projects.

Find the right project and click the "+" to add it to studio.

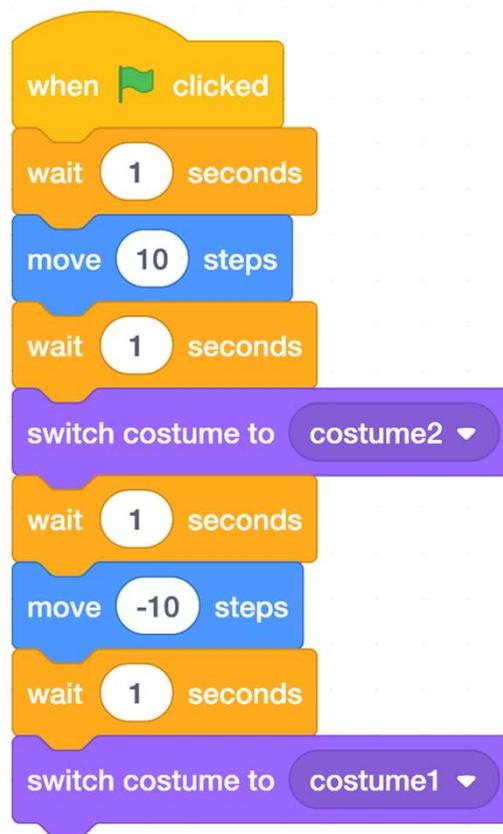


Teacher may use feedback form e.g. "1 Star – 2 Wishes", to guide students having peer / self-evaluation and to think of ways to enhance their program, including the aesthetic aspect.

# Exploring Under the Sea

## Review Questions

1. Assuming that the sprite is initially wearing costume1 and facing right, what will its state be 3 seconds after the green flag is clicked?



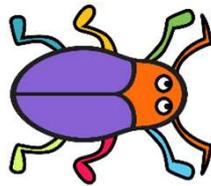
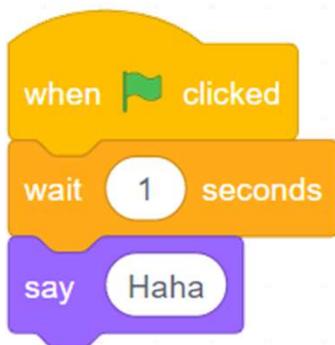
- A. The sprite will wear costume1 and be in its original position.
- B. The sprite will wear costume1 and has moved 10 steps to the right.
- C. The sprite will wear costume2 and be in its original position.
- D. The sprite will wear costume2 and has moved 10 steps to the right.

**(Answer: C)**

# Exploring Under the Sea

## Review Questions

2. When green flag clicked, who will make sound first?



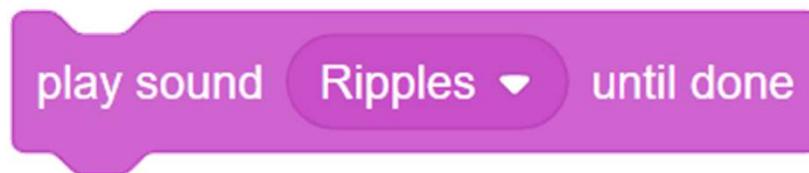
- A. Duck
- B. Beetle
- C. Frog
- D. All sprites will make sound at the same time.

**(Answer: B)**

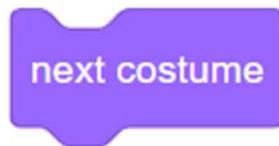
# Exploring Under the Sea

## Revision on Key Features

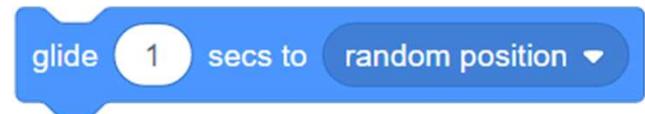
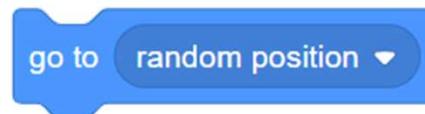
### Music:



### Looks:



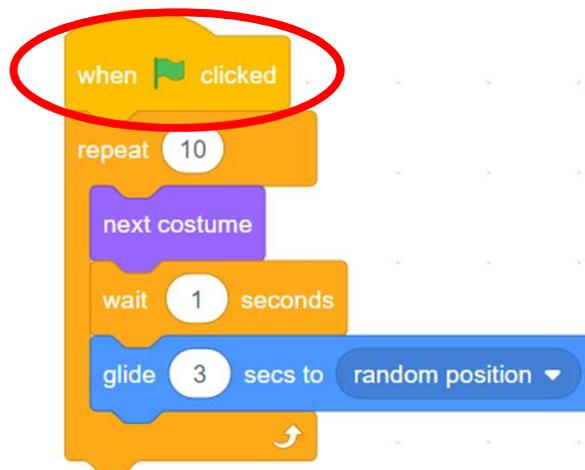
### Motion:



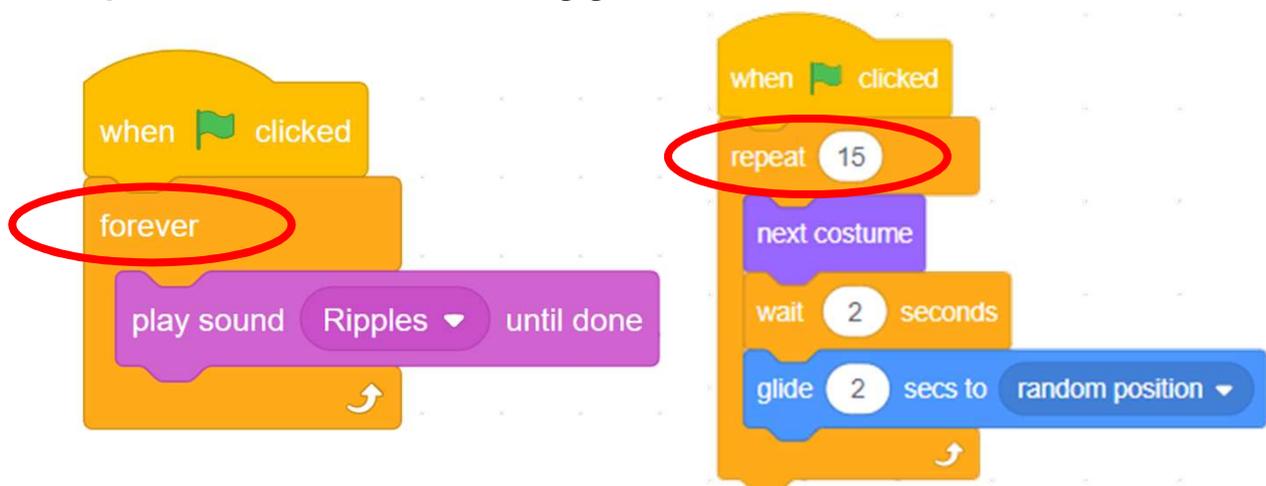
# Exploring Under the Sea

## Revision on Key Concepts & Practices

**Events:** We use event blocks to trigger Scratch to take actions. For example, when green flag clicked is a typical event.



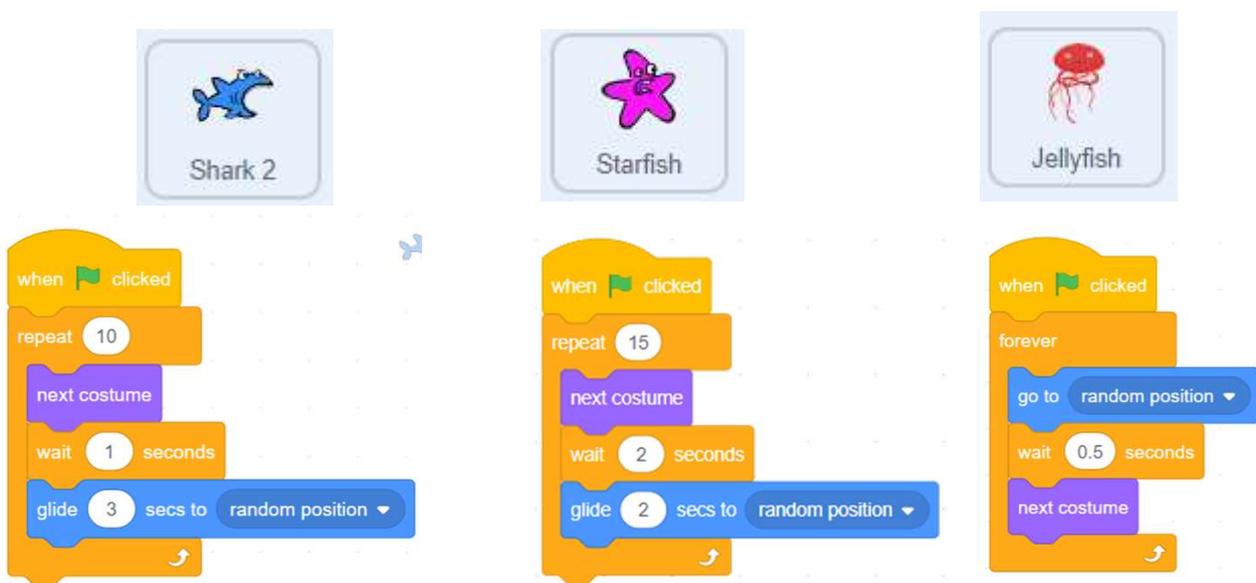
**Iteration - repeat:** Iteration is repeating a process in order to produce a sequence of outcomes. “Forever” and “repeat” blocks can trigger iteration in Scratch.



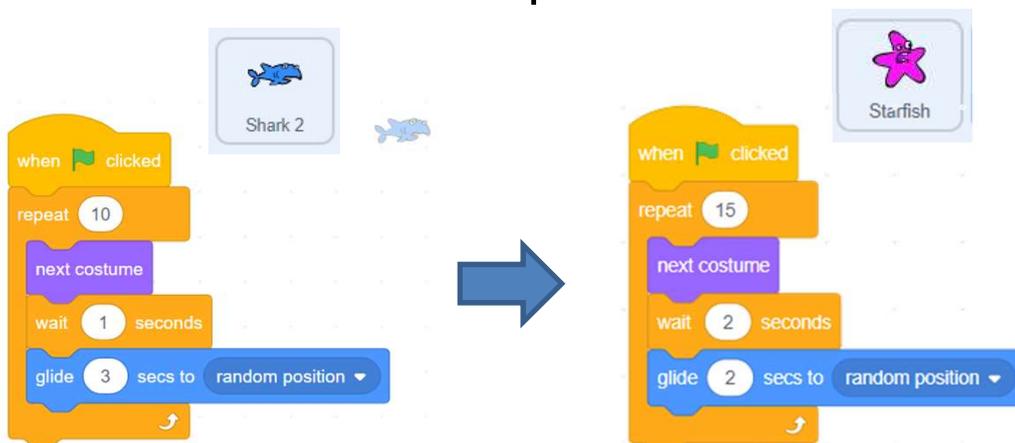
# Exploring Under the Sea

## Revision on Key Concepts & Practices

**Parallelism:** Parallelism is a series of codes running at the same time. Scratch supports parallelism across multiple sprites.



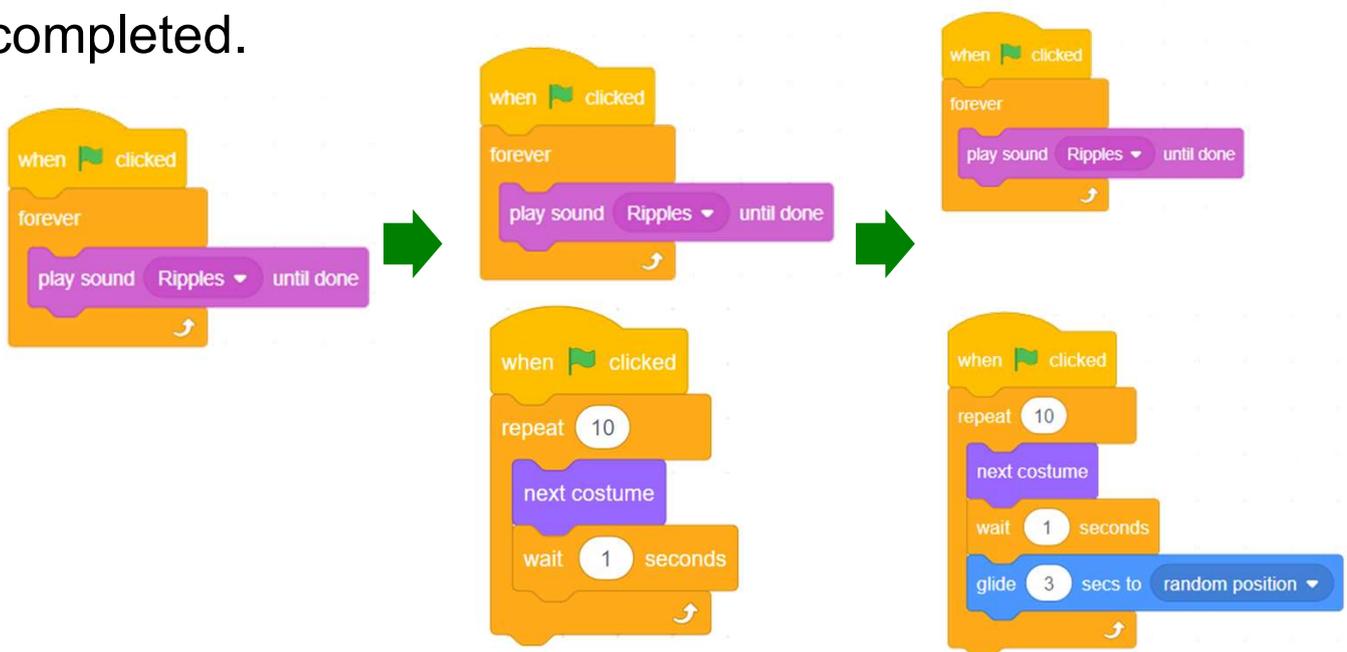
**Reuse and remix programs / codes:** In Task 2, we reuse and remix the codes of the shark sprite and use them for the second and third sprites.



# Exploring Under the Sea

## Revision on Key Concepts & Practices

**Being incremental and iterative:** To work out a sub-task as an iteration, try it out, then work out another sub-task in another iteration until the whole programming task is completed.



**Testing and debugging:** Testing a computer program is the process of checking if it can produce outcomes as designed. Debugging a computer program is the process of finding out ways to revise the program so that the bugs can be removed.



# Appendix

## Operation Manual

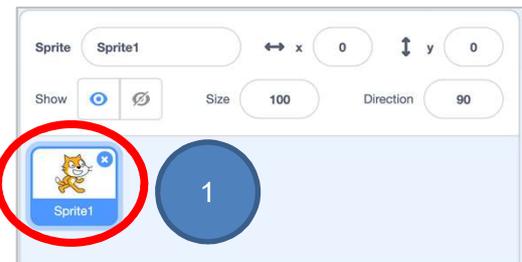
# Exploring Under the Sea

See Teacher  
Guide P.8

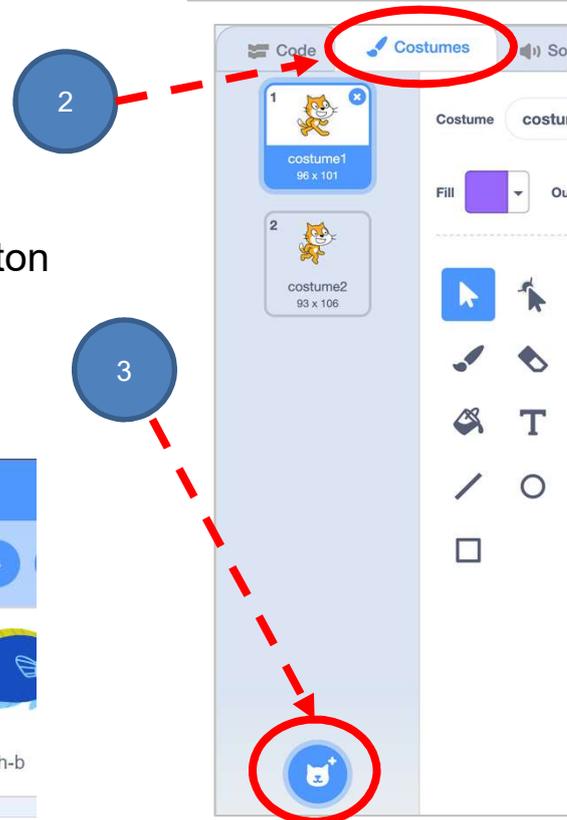
## To Code: (B) Change Looks of the Sprite

Choose a good sprite for swimming under the sea!

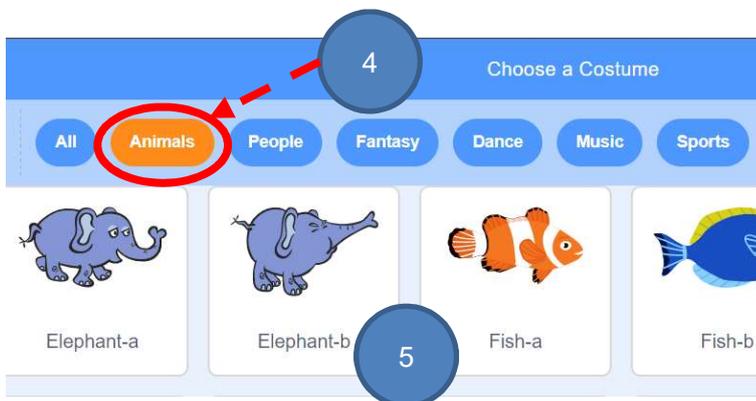
1. Click on the Scratch cat.



2. Click on the **Costumes** tab.



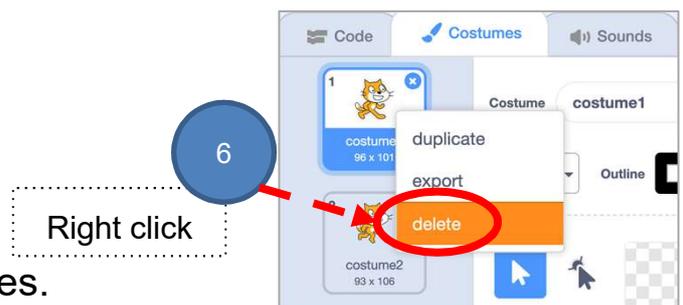
3. Click on the “**Choose a Costume**” button at the bottom left to select a costume.



4. Click on the “**Animals**” category on the top to see a large set of animal costumes.

5. Add *multiple* costumes for your sprite.

6. Delete the original Scratch cat costumes.



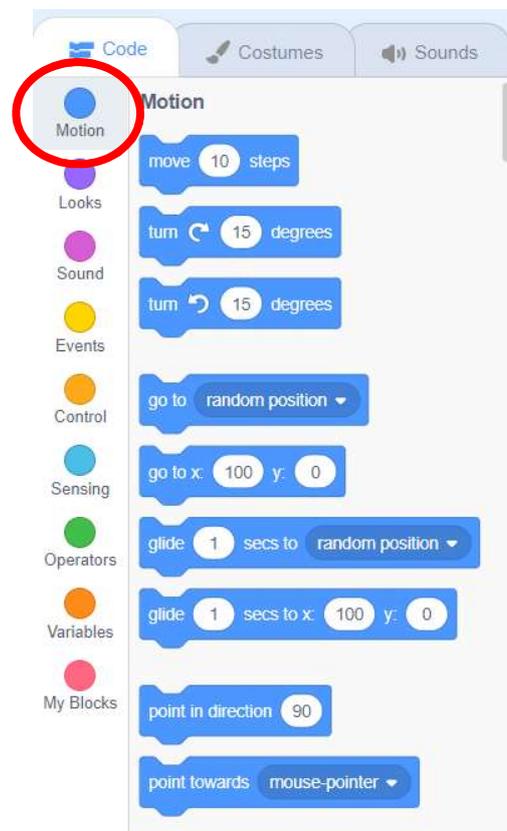
# Exploring Under the Sea

## To Code: (C) Make Sprite Swim

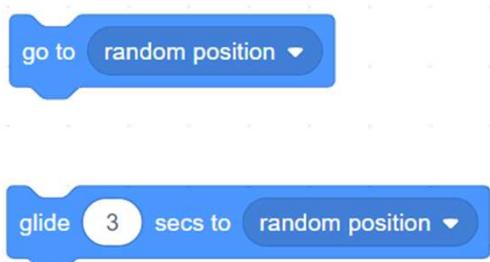
See Teacher Guide P.10

Now, let's make your sprite swim under the sea.

1. Click **Motion** drawer, you can move the sprite to certain point.



2. Or let it swim and explore to random position with different motion.



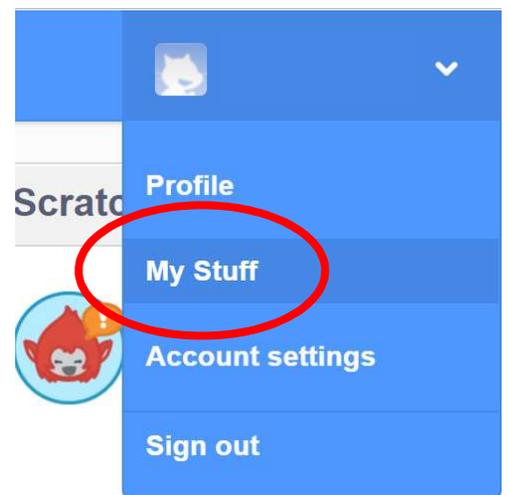
Test to see the difference between **go** and **glide**.

# Exploring Under the Sea

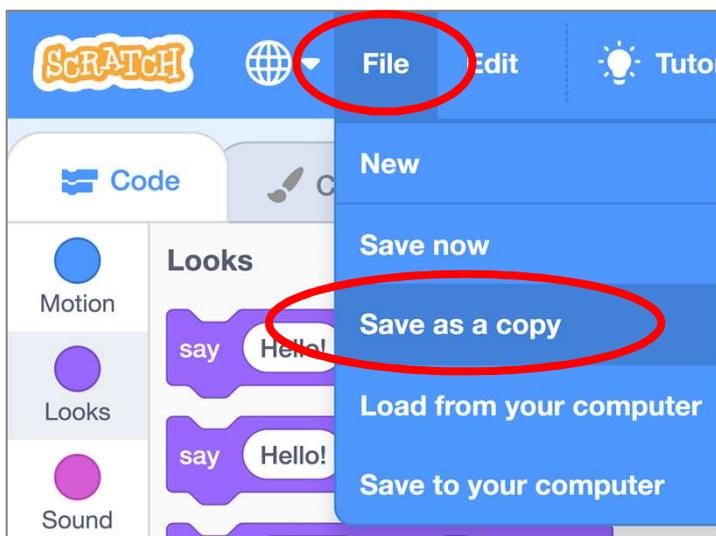
See Teacher  
Guide P.12

## Save Project as Copy

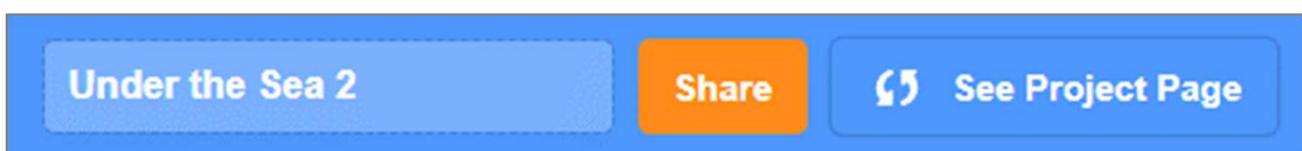
1. Sign into your account at <https://scratch.mit.edu/> Click on “My Stuff” then open your “Under the Sea” Project.



2. Save as a copy.



3. Name it **Under the Sea 2**.

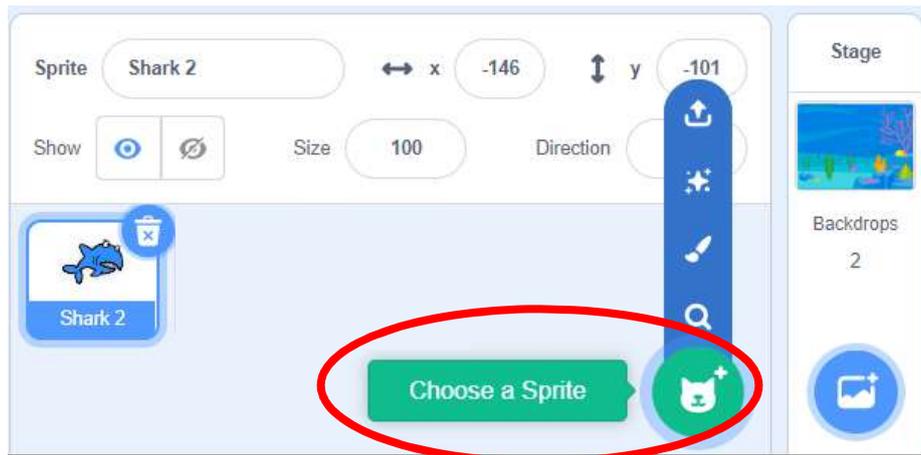


# Exploring Under the Sea

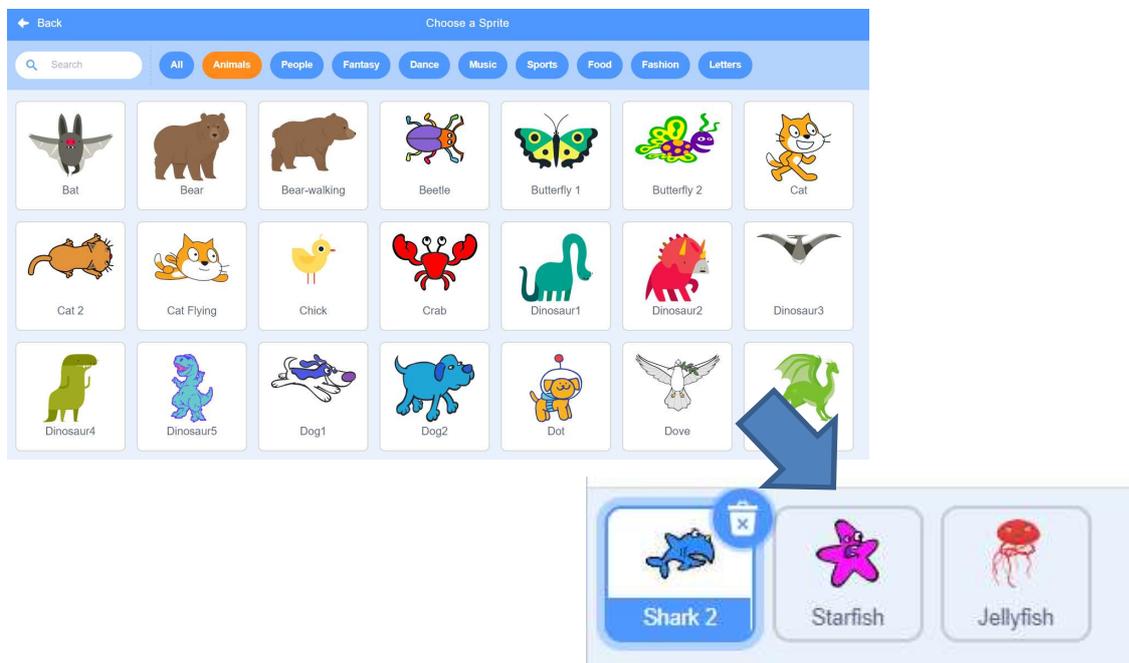
See Teacher  
Guide P.12

## To Code: Add Sprites

1. Let's make friends under the sea. Click **Choose a Sprite**.



2. Add another sprite (and even more) to the sea.



# Exploring Under the Sea

## Reuse and Remix Code Blocks of Sprites

1. Click the sprites we created in lesson 1.

See Teacher Guide P.13

2. Drag the whole set of code blocks to other sprites you want.
3. Now you will see two sprites have the same code blocks.

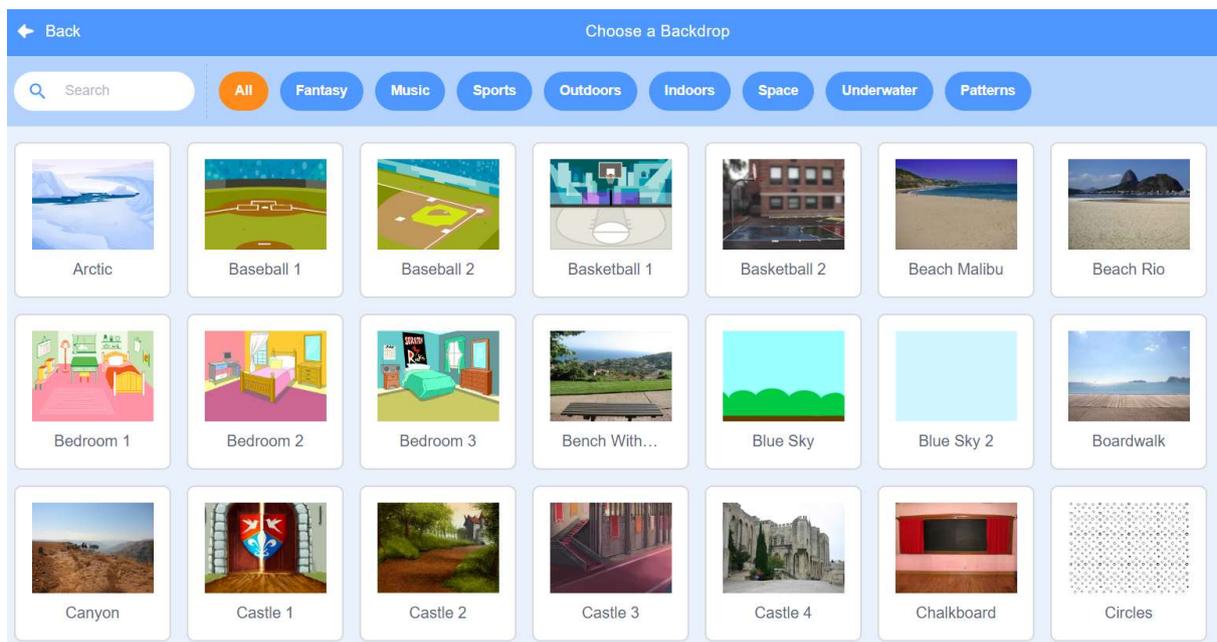
4. Simply remix the code e.g. repeat **15**, wait **2** seconds and more...to make them act in different ways.

# Exploring Under the Sea

## More Effects: Add Backdrop and Sound

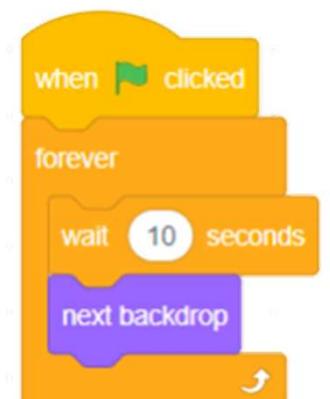
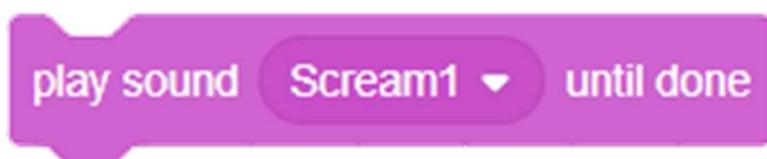
See Teacher Guide P.13

1. You can add more backdrops (e.g. underwater, ocean or sea etc.)!



2. Here it's an example to keep changing the backdrop.

3. You may also add sound effect:



# Exploring Under the Sea

## Program Codes (Task 2)



Sprite (Shark 2)

```
when green flag clicked
repeat 10
  next costume
  play sound Bite until done
  wait 1 seconds
  glide 3 secs to random position
```

# Exploring Under the Sea

## Program Codes (Task 2)



Sprite (Starfish)

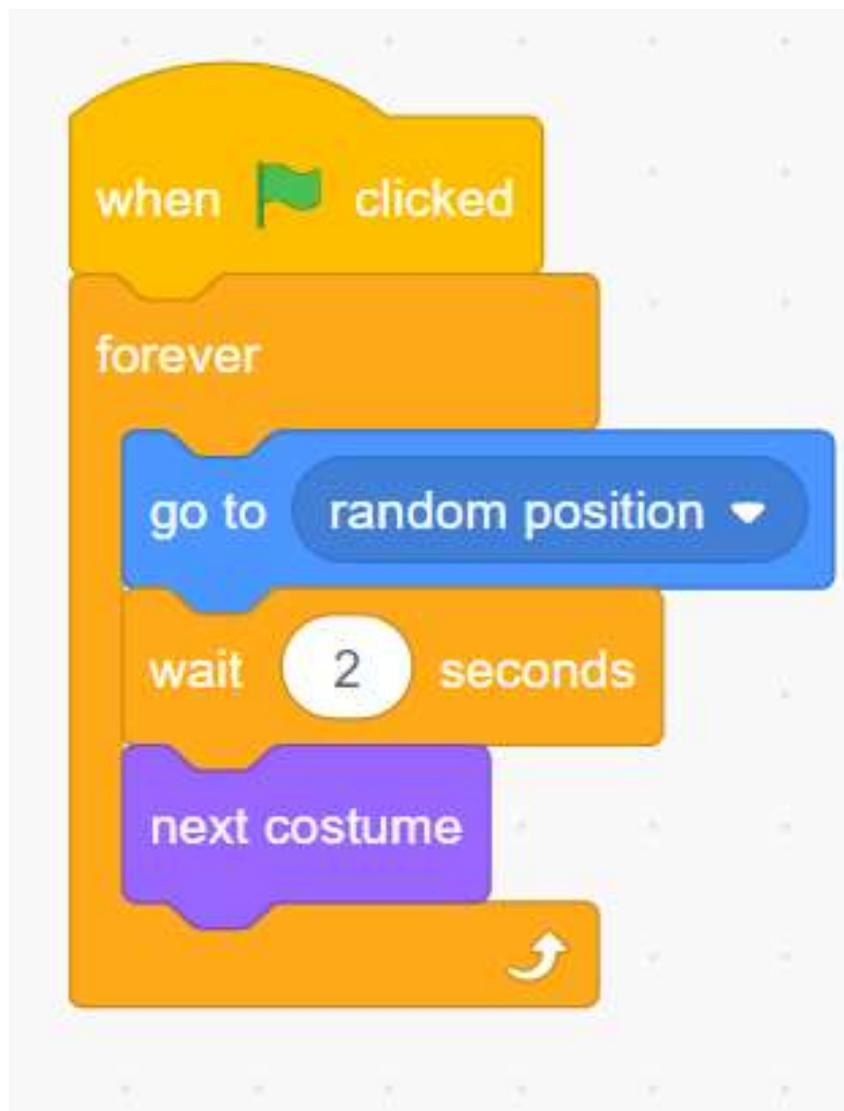
```
when green flag clicked
repeat (15)
  next costume
  wait (2) seconds
  glide (2) secs to random position
```

# Exploring Under the Sea

## Program Codes (Task 2)



Sprite (Jellyfish)

A screenshot of Scratch code blocks. The code starts with a yellow "when green flag clicked" block. Below it is an orange "forever" loop block. Inside the loop, there are three blocks: a blue "go to random position" block, an orange "wait 2 seconds" block, and a purple "next costume" block. The "forever" loop block has a white arrow at the bottom right, indicating it repeats.

# Exploring Under the Sea

## Program Codes (Task 2)



Backdrops

```
when clicked clicked
forever
  wait 10 seconds
  next backdrop
```

```
when clicked clicked
forever
  play sound Ripples until done
```

# Unit 3: Storytelling Teacher Guide

## Content

<b>Teaching Plan</b>	P3
<b>Lesson 1</b>	
To Play	T3-1
To Think	T3-3
To Code	
Remove "Wait" Block	T3-9
Replace with "broadcast" and "when I receive"	T3-11
Complete the Second Dialogue of Scene 1	T3-13
Complete the Dialogue of Scene 2	T3-16
<b>Lesson 2</b>	
To Create	T3-19
To Reflect	T3-24
Review Questions	T3-25
Revision on Key Features	T3-28
Revision on Key Concepts & Practices	T3-29
<b>Program Codes</b>	T3-33

## Unit 3: Storytelling Teaching Plan

### Prior Knowledge

Students are expected to know the skills of adding sound to a sprite, adding sound to a backdrop, and changing costumes for the sprite in Scratch.

### Learning Objectives

1. Demonstrate an understanding of “broadcast” and “when I receive” blocks and switching backdrops in a project;
2. Apply the computational thinking concepts and practices of sequences in the production of a Scratch project; reusing and remixing through a starter project with provided assets;
3. Foster students’ creativity by creating their own story and sharing with their classmates, teachers and parents.

### Learning Elements

#### Computational Thinking Concepts and Practices:

Key Learning Elements	Items
Abstraction	Decompose tasks into sub-tasks
Algorithm	Problem Solving Procedures: Problem identification, Problem analysis, Algorithm design, Programming Basic Programming Constructs: Sequence Coding Concepts and Practices: Design, Reuse and Remix programs / codes, Testing and Debugging

#### Coding Skills:

1. Make use of “broadcast” and “when I receive” commands;
2. Understand the importance of sequence in programming through storyboarding and coding;
3. Apply Testing and Debugging in completing each task of this unit.

#### Others (including Attitude):

1. Develop interest in programming;
2. Show perseverance and positivity in testing and debugging;
3. Inspire students to be creative and innovative to tell their own story;
4. Develop positive values and attitude through story telling activity e.g. concern for the environmental protection issues.

**Lesson Plan:** This unit consists of 2 lessons of 35 minutes.

**Teacher Preparation for the Lesson:** Prepare the studio for students to share their projects.

**Pre-task for Students:** Pre-task for Lesson 2. Ask students to think about what story they would like to create with Scratch during the lesson.

### Lesson 1

Time	Activity
5 mins	<b>To Play: Help students gain an initial understanding of the program</b> <ol style="list-style-type: none"><li>1. Play the story (Demo) <a href="https://scratch.mit.edu/projects/753681874/">https://scratch.mit.edu/projects/753681874/</a>.</li><li>2. Ask students what they can observe in the demo project e.g. sprites, backdrop and conversation etc.</li></ol>
20 mins	<b>To Think</b> <ol style="list-style-type: none"><li>1. Ask students to fill in the boxes with correct letters to understand the sequence of the whole story.</li><li>2. Read the code blocks to understand how the wait blocks make the conversation go.</li><li>3. Ask students try to adjust the seconds of wait time in Scene 1 and click the green flag to see what happens.</li><li>4. Guide them to think about:<ol style="list-style-type: none"><li>1) What is the problem with “wait” block?</li><li>2) Why would that happen?</li><li>3) What did they do to solve this problem?</li></ol></li><li>5. Explain to students that it is difficult to remember how long to wait for the other sprite to speak, bringing out the need to use “broadcast” and “when I receive”.</li><li>6. Run the “Unplugged Activity: Role Play on Broadcast” with students and let them understand more about the use of “broadcast” and “when I receive” before coding.</li></ol>
10 mins	<b>To Code</b> <ol style="list-style-type: none"><li>1. Remove the wait blocks and replace it by broadcast / when I receive code blocks.</li><li>2. Ask students to complete scene1 and scene2 with broadcast blocks.</li><li>3. Testing and Debugging: Students can click the green flag to test to see how the story goes and debug if needed.</li></ol>

## Lesson 2

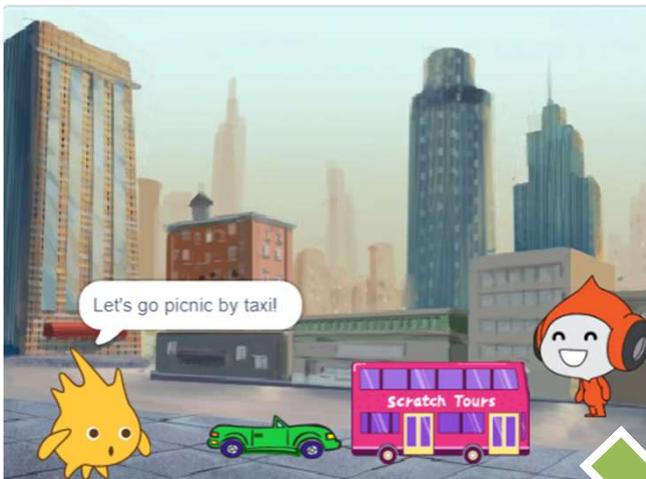
Time	Activity
10 mins	<p><b>To Create: Task 3 (Tell Your Own Story)</b></p> <ol style="list-style-type: none"><li>1. Ask students to design their own story by drawing on the storyboard in the student guides, add numbering to show the sequence.</li><li>2. Tell students that they do not have to design a perfect story but they should have something completed on the paper because they can edit it anytime.</li><li>3. Remind students that they should spare time for coding on Scratch rather than drawing on the paper for the whole lesson.</li><li>4. Complete the mind map according to their storyboard before they start to code.</li></ol>
15 mins	<p><b>To Code: Task 3 (Tell Your Own Story)</b></p> <p>Ask students to code on Scratch with the storyboard and mind map they create.</p>
10 mins	<p><b>To Reflect:</b></p> <p><b>Sharing to Studio and Provide Constructive Feedback on Program Design</b></p> <ol style="list-style-type: none"><li>1. Create a studio and give students the studio URL. Ask students to save and submit their projects to the teacher's Studio.</li><li>2. Have one or two students share their animations with the whole class.</li><li>3. Students should share, think of ways to improve / enhance their program, including the aesthetic aspect.</li></ol> <p><b>Review of Student Learning</b></p> <ol style="list-style-type: none"><li>1. Review on the features of Scratch, and key concepts and practices learnt in the unit.</li><li>2. Ask students to complete the review questions, appropriate feedback should be given by teachers</li></ol>

# Storytelling

## To Play

- Play the story (Demo)  
<https://scratch.mit.edu/projects/753681874/>
- How many sprites are there?
- What do they say?
- Why can they speak like having a conversation? How to do that?
- Does the conversation run too slow or too fast? Can you adjust the speed?

Teachers can adjust the guiding questions to better suit their student need.



Scene1

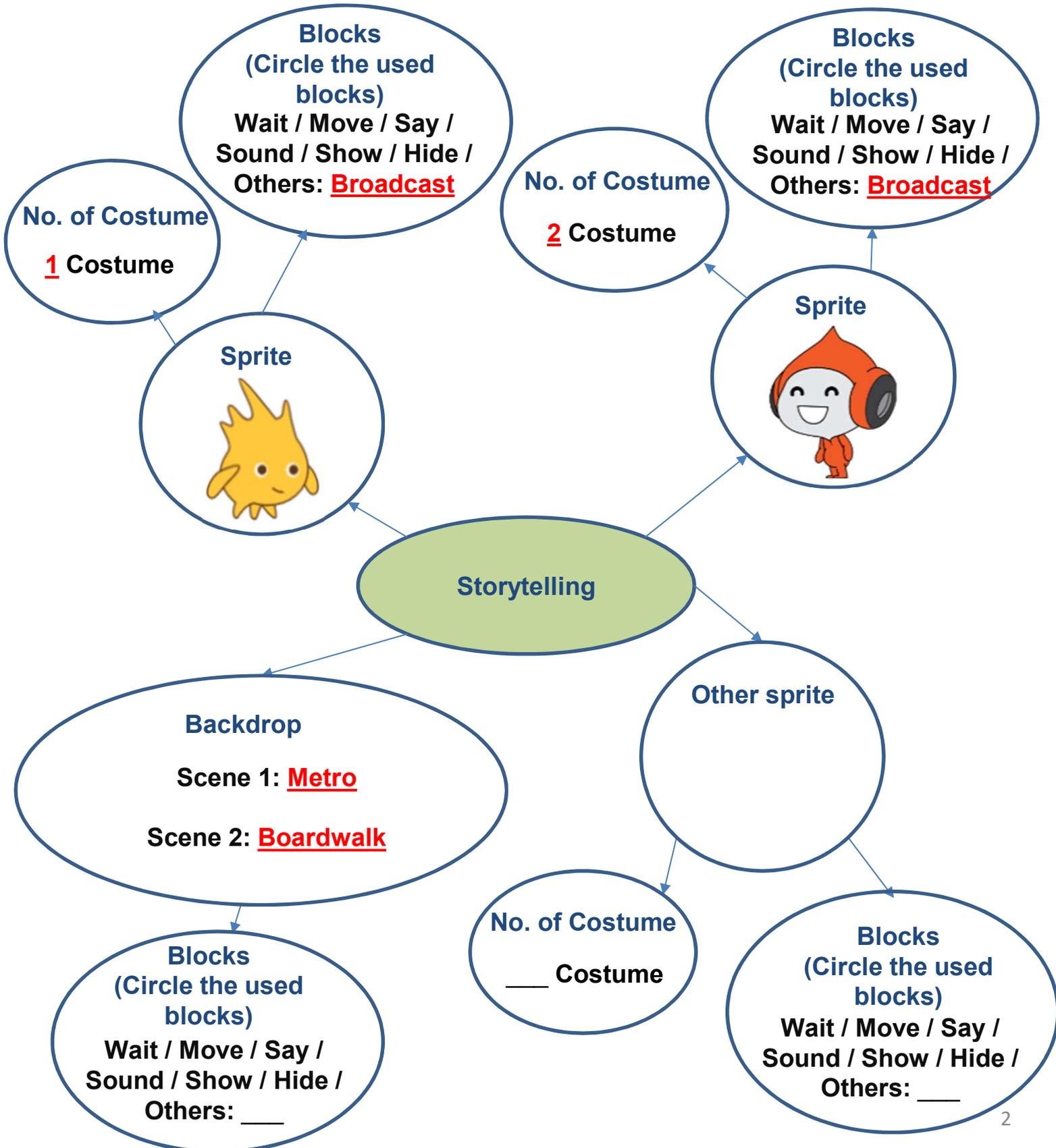


Scene2

# Storytelling

- ❑ Complete the **mind map** below when playing to observe.

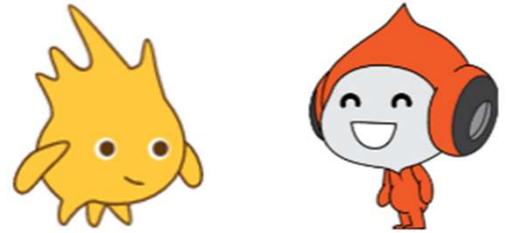
Students are invited to find out the main elements of the demo project.



# Storytelling

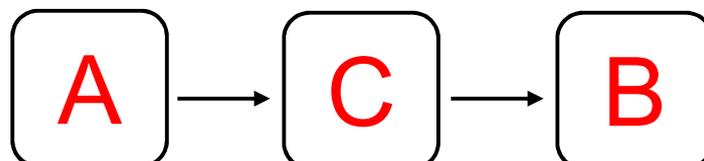
## To Think

☐ Check that you understand the sequence of steps of the storyboard in Scene 1 at Metro.



☐ Fill the empty boxes with the correct letters.

### Scene 1 at Metro

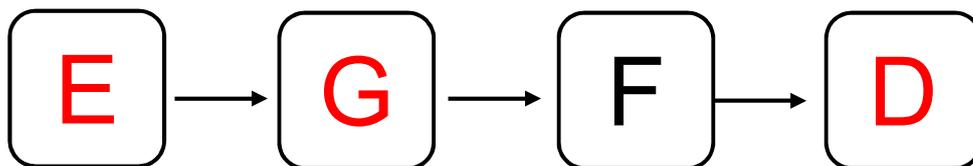
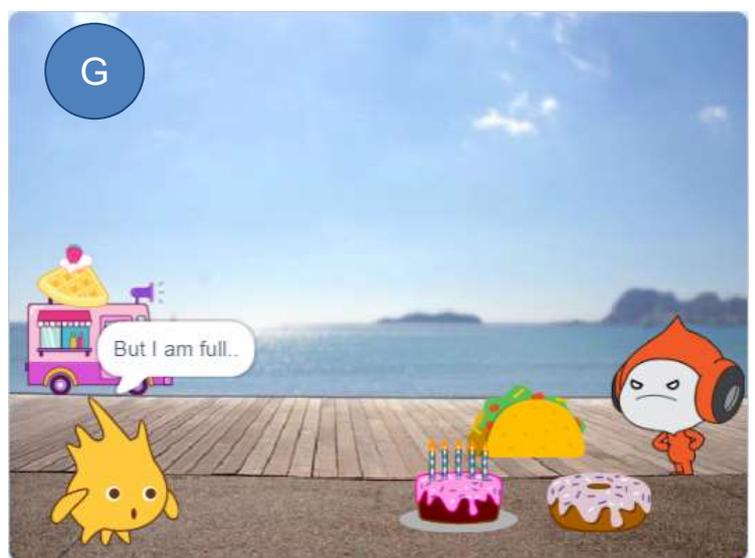
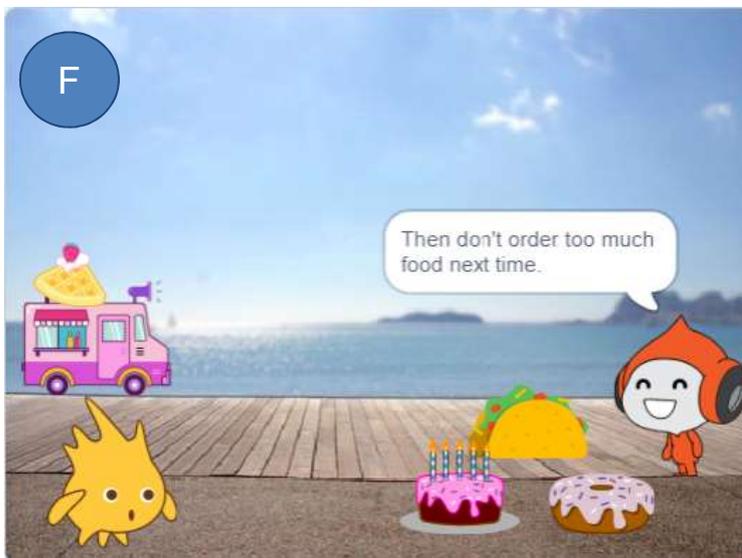


# Storytelling

## To Think

- How about the sequence of steps of the storyboard in Scene 2 at the Boardwalk?
- Fill the empty boxes with the correct letters “D” to “G”.

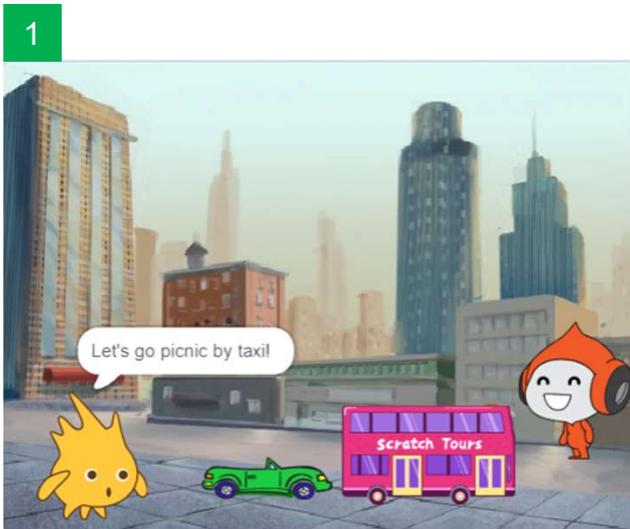
### Scene 2 at Boardwalk



# Storytelling

## To Think

- Review the sequences on Scene 1 at Metro to follow the current blocks that tell the story.
- By using the “wait” block, it seems that they are talking.



Gobo:

```
when clicked
say Let's go picnic by taxi! for 3 seconds
wait 6 seconds
say Ok, let's go! for 3 seconds
```



Pico:

```
when clicked
switch costume to pico-b
wait 3 seconds
say Let's take the bus. for 3 seconds
say It is more environmental friendly. for 3 seconds
```

- Now, you try to add “**Shall we?**” after Gobo’s “Let’s go picnic by taxi”.
- Then click the  and see what happens!



Gobo:

```
when clicked
say Let's go picnic by taxi! for 3 seconds
say Shall we?
wait 6 seconds
say Ok, let's go! for 3 seconds
```

After updating the script, did they speak to each other in order?  
Did they speak at the same time?  
Why would this happen?

## To Think:



- What is the problem with the “wait” block?

The sprite does not respond according to the time when the other finish his script, sometimes the two sprites speak at the same time.

- Why would that happen?

It is difficult to find out how long to wait for the other sprite to speak.

- What did you do to solve this problem?

Try to write down the wait time for each sprite and do testing for a few times.

## Unplugged Activity: Role Play on Broadcast

This role play activity will ask two students to play the part of Gobo and Pico, which try to show students how Gobo gives a signal to Pico after finishing his words. It aims to let students to understand the use of “broadcast” and “when I receive”.

Student 1 (Gobo) 	Student 2 (Pico) 
<b>When  is clicked</b> Say: Let's go picnic by taxi! <b>(Raise Card No.1)</b>	
	<b>WHEN I RECEIVE Message 1</b> Say: Let's take the bus instead. It is more environmental friendly. <b>(Raise Card No.2)</b>
<b>WHEN I RECEIVE Message 2</b> Say: Ok, let's go! <b>(Raise Card No.3)</b>	
	<b>WHEN I RECEIVE Message 3</b> Say: Why are there so much leftover food? It is not environmental friendly. <b>(Raise Card No.4)</b>
<b>WHEN I RECEIVE Message 4</b> Say: But I am full... <b>(Raise Card No.5)</b>	
	<b>WHEN I RECEIVE Message 5</b> Say: Then don't order too much food next time. <b>(Raise Card No.6)</b>
<b>WHEN I RECEIVE Message 6</b> Say: We are all eco-warrior!	<b>WHEN I RECEIVE Message 6</b> Say: We are all eco-warrior!

# Storytelling

In this activity, you will modify a Scratch project to use “broadcast” and “when I receive” blocks instead of the “wait” block.



## To Code

1. Sign into your account at [scratch.mit.edu](https://scratch.mit.edu).

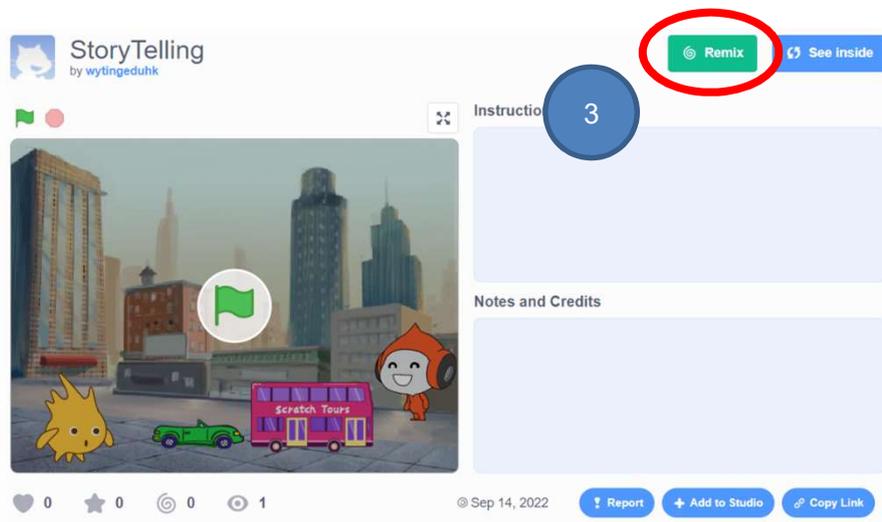
1

2. Go to the Storytelling project at:

<https://scratch.mit.edu/projects/753681874/>

2

3. We are going to use the code from this project, so we need to “remix” the project first. Click the “Remix” button.



4. You can now use this original code and save it as your own project! Rename the project to “**Storytelling with Broadcast**” and save it.



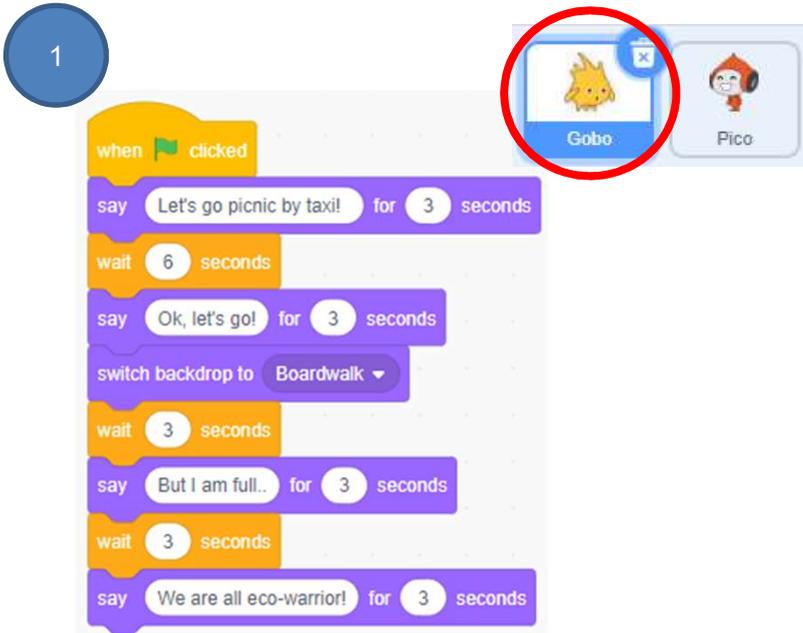
4

# Storytelling

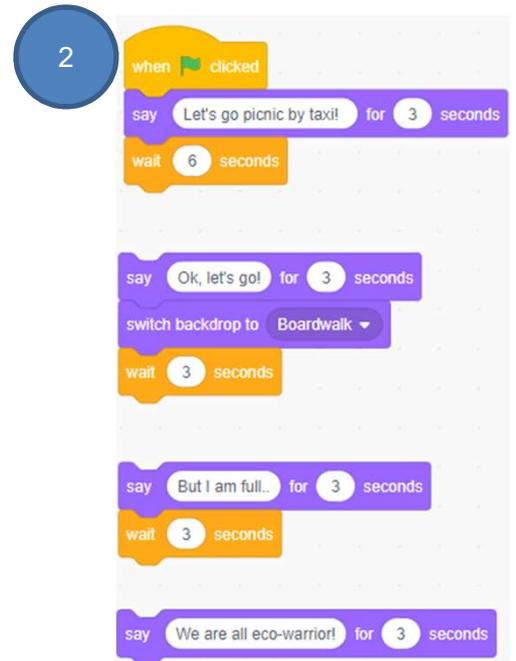
## To Code: Remove “Wait” Block

Now we are going to use “Broadcast” and “When I receive” to replace “wait” block. Break the code out for Gobo and Pico by each part of the storyboard, and then **remove the “wait” blocks**.

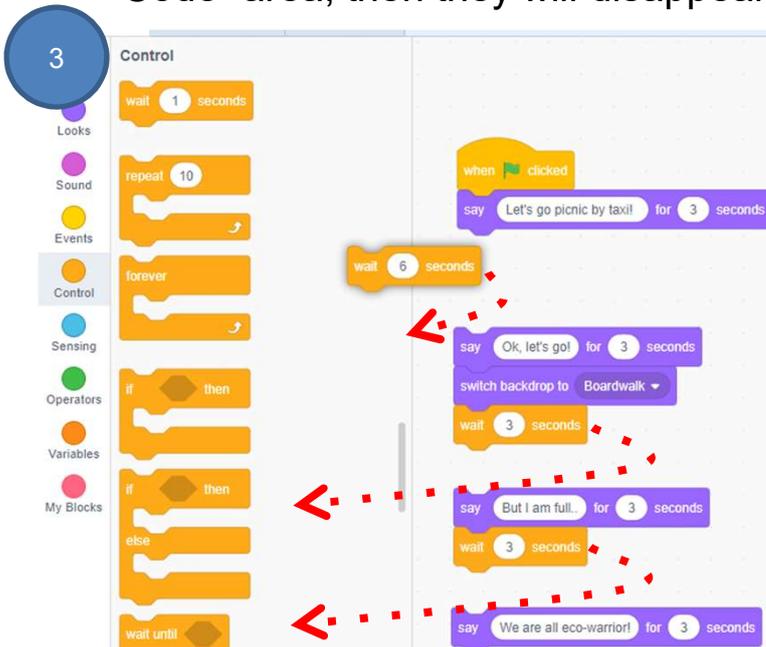
1. Click on Gobo under “Sprite” to open its script.



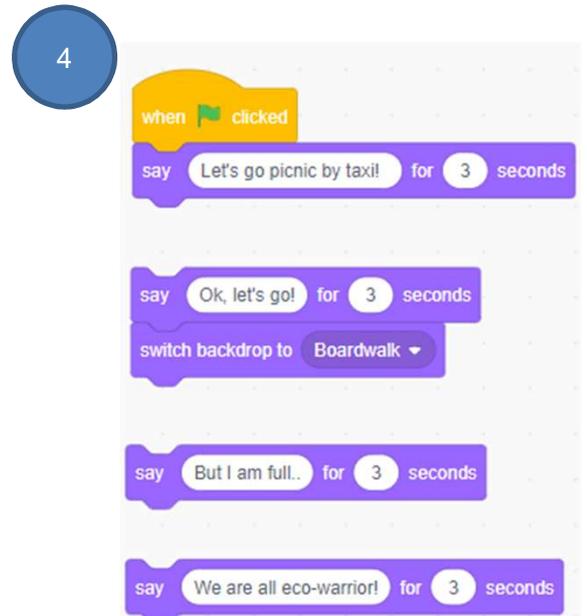
2. Break the code out.



3. Drag the “wait” blocks out of the “Code” area, then they will disappear.



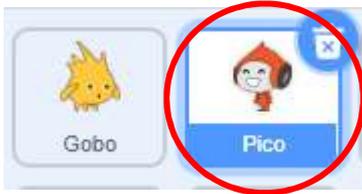
4. Done!



## To Code: Remove “Wait” Block

- Repeat the steps we just completed for Gobo, remove the “wait” block for Pico as below.

**Pico's code:**



```
when clicked
  switch costume to pico-b

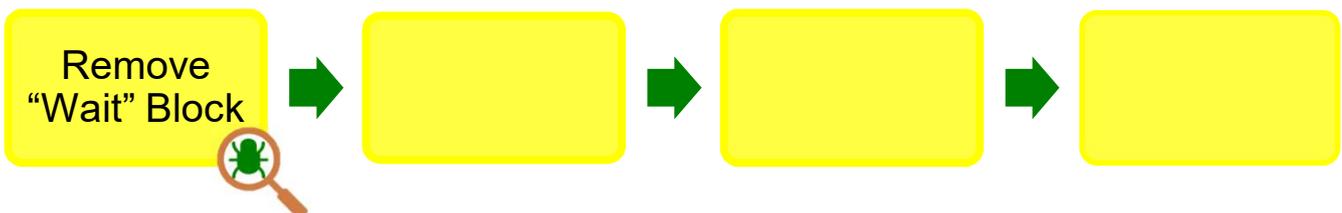
say Let's take the bus. for 3 seconds
say It is more environmental friendly. for 3 seconds

when backdrop switches to Boardwalk
  switch costume to Pico-d
  say Why are there so much leftover food? for 3 seconds
  say It is not environmental friendly! for 3 seconds

switch costume to pico-b
say Then don't order too much food next time. for 3 seconds
say We are all eco-warrior! for 3 seconds
start sound Cheer
```

### Testing and Debugging

Let's test it! After removing all “wait” block, what happened when you click the green flag?



# Storytelling

## To Code: Replace with “broadcast” and “when I receive”

Use the “broadcast” and “when I receive” blocks to replace all of the “wait” blocks for the conversation of the sprites. Remember that both blocks are in the Events drawer.

- Start with Gobo’s first statement in Scene 1 at Metro. Add a “broadcast message1” block to the end. The message number follows the storyboard sequences.

1

Let's go picnic by taxi!

Scratch Tours

when green flag clicked

Gobo

say Let's go picnic by taxi! for 3 seconds

broadcast message1

1

Events

- when clicked
- when space key pressed
- when this sprite clicked
- when backdrop switches to backdrop1
- when loudness > 10
- when I receive message1
- broadcast message1
- broadcast message1 and wait

- Switch to Pico’s script and add a “when I receive message1” block to the top of her first statement.

2

Let's take the bus instead.

Scratch Tours

when I receive message1

Pico

2

say Let's take the bus instead. for 3 seconds

say It is more environmental friendly. for 3 seconds

# Storytelling

## To Code: Replace with “broadcast” and “when I receive”

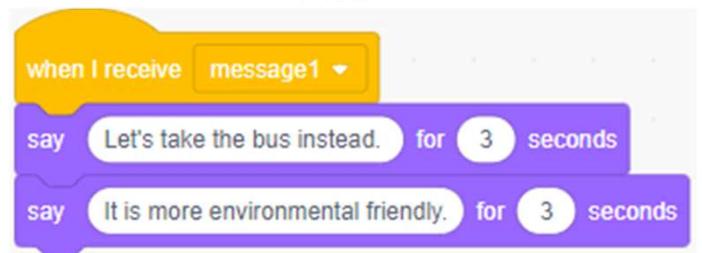
Now you should have the following blocks after adding “when I receive” and “broadcast” block.



Gobo



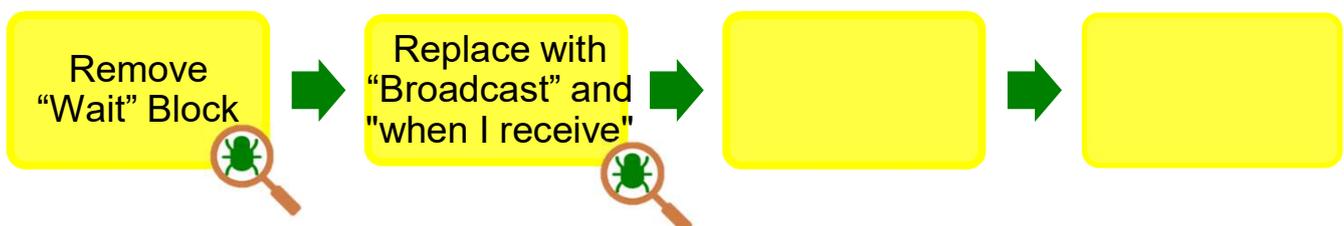
Pico



Remember to include a “when I receive” block before a sprite speaks and a “broadcast” block after the sprite speaks.

### Testing and Debugging

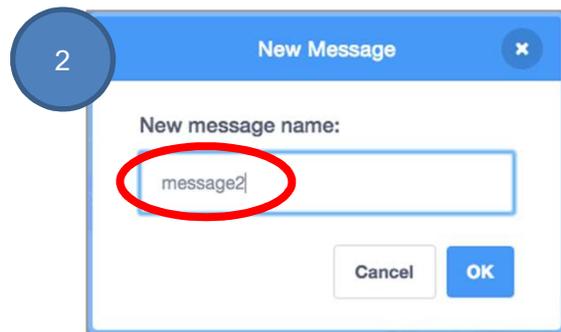
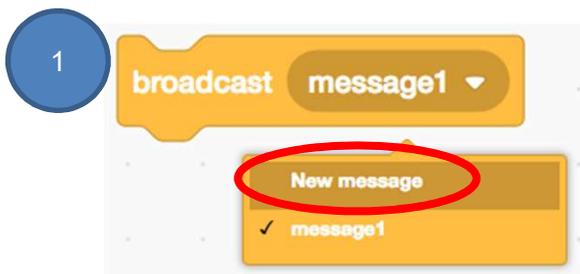
Let's test it! When you click the green flag button, what happens?  
Did the sprites talk to each other?



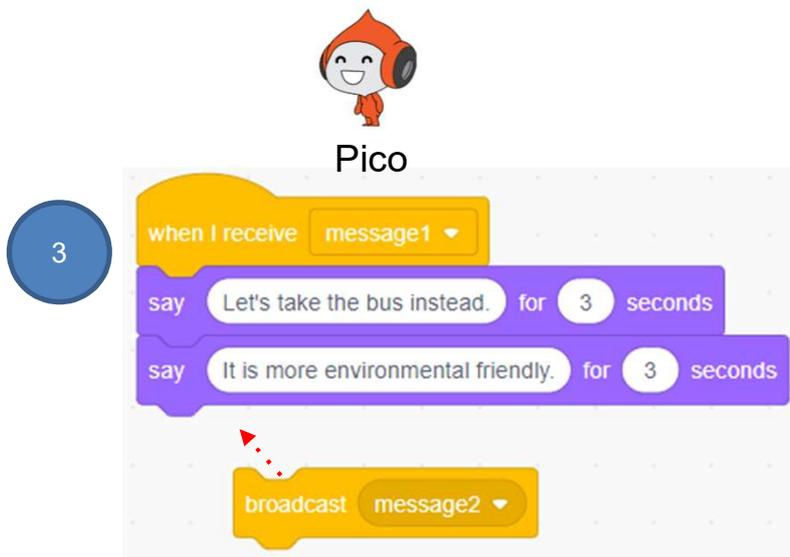
## To Code: Complete the Second Dialogue of Scene 1

Let's add "Broadcast" Block to all statements in Scene 1. Now Gobo needs to broadcast back to Pico that she's responded, and it's Gobo's turn to speak.

1. Drag out a new "broadcast" block, and click on message1, and select New message.
2. Name it message2.



3. Snap the broadcast block to the end of Pico's speaking block.

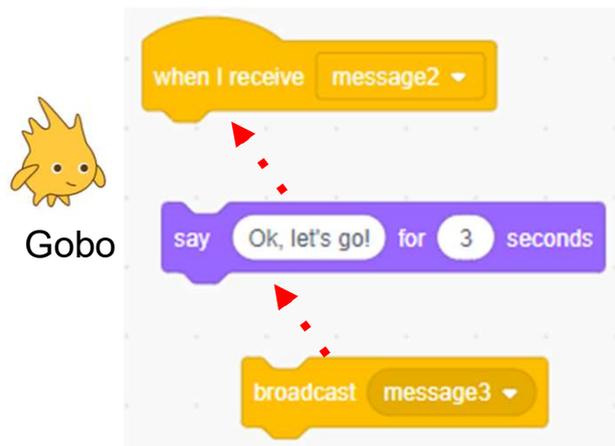


# Storytelling

## To Code: Complete the Second Dialogue of Scene 1

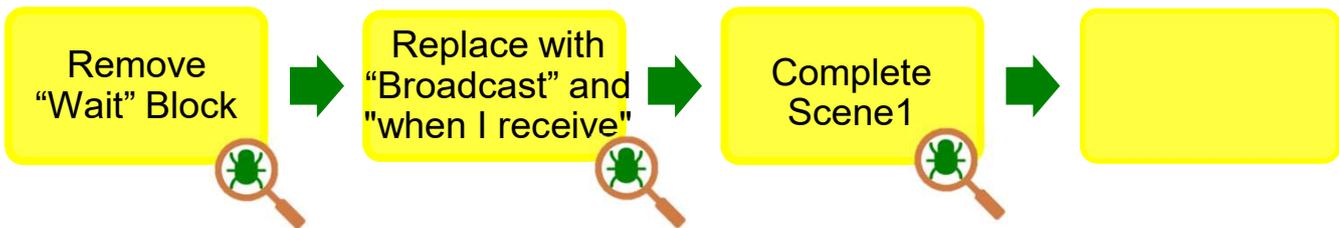
The next step (3) in the storyboard is Gobo's next statement.

- ❑ Repeat the step you completed earlier. Add "when I receive" and "broadcast" blocks to Gobo's script. Name the new message as message3.



### Testing and Debugging

Let's test it again! Did the sprites talk to each other same as you expected?



- ❑ Still remember how to add new backdrop you have learnt in the previous unit?
- ❑ We are going to switch to another backdrop (Scene 2 in Boardwalk) in the next step! We will use "When backdrop switches to \_\_\_" to trigger all actions in Scene 2.



"When backdrop switches to \_\_\_" is another Event. Event is something that triggers actions.

# Storytelling

Before going to Scene 2, let's take a look at how Scene 1 should look like now.

## Scene 1

Screen	 Gobo	 Pico
1 A cityscape backdrop with a green taxi and a pink bus. Gobo is on the left, and Pico is on the right. A speech bubble from Gobo says "Let's go picnic by taxi!".	<pre>when clicked say Let's go picnic by taxi! for 3 seconds broadcast message1</pre>	
2 The same cityscape backdrop. Pico is on the right, and Gobo is on the left. A speech bubble from Pico says "Let's take the bus instead.". The taxi and bus are still present.		<pre>when I receive message1 say Let's take the bus. for 3 seconds say It is more environmental friendly. for 3 seconds broadcast message2</pre>
3 The same cityscape backdrop. Gobo is on the left, and Pico is on the right. A speech bubble from Gobo says "Ok, let's go!". The taxi and bus are still present.	<pre>when I receive message2 say Ok, let's go! for 3 seconds broadcast message3 switch backdrop to Boardwalk</pre>	

## To Code: Complete the Dialogue of Scene 2

Now, let's move on the Scene 2 at Boardwalk to have their picnic!

- ❑ The next step (4) in the storyboard is **Pico** next statement.
- ❑ Add **“when I receive”** and **“broadcast”** blocks to **Pico**'s script. Name the New message as message4.



Pico

Change **Pico**'s costume to show his expression



4



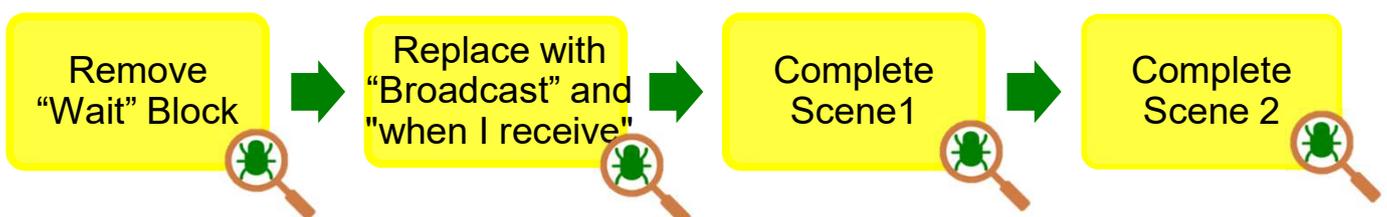
## To Code: Complete the Dialogue of Scene 2

- The next step (5) is **Gobo**'s next statement. Can you complete it (5) and also the rest of them (6) (7)?
- Try to add some **sound** at the end to make the story more interesting!



### Testing and Debugging

Good job! Click the green flag to see if the story works.



# Storytelling

Now Scene 2 should look like ...

## Scene 2

Screen



Gobo



Pico



```

when I receive message3
  switch costume to Pico-d
  say Why are there so much leftover food? for 3 seconds
  say It is not environmental friendly! for 3 seconds
  broadcast message4
  
```



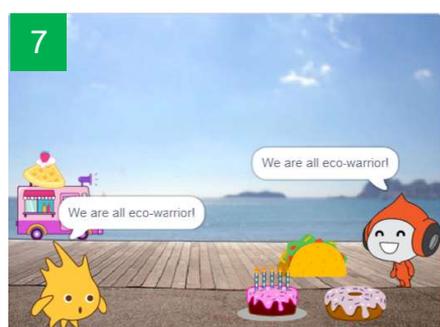
```

when I receive message4
  say But I am full.. for 3 seconds
  broadcast message5
  
```



```

when I receive message5
  switch costume to pico-b
  say Then don't order too much food next time. for 3 seconds
  broadcast message6
  
```



```

when I receive message6
  say We are all eco-warrior! for 3 seconds
  start sound Cheer
  
```

```

when I receive message6
  say We are all eco-warrior! for 3 seconds
  start sound Cheer
  
```

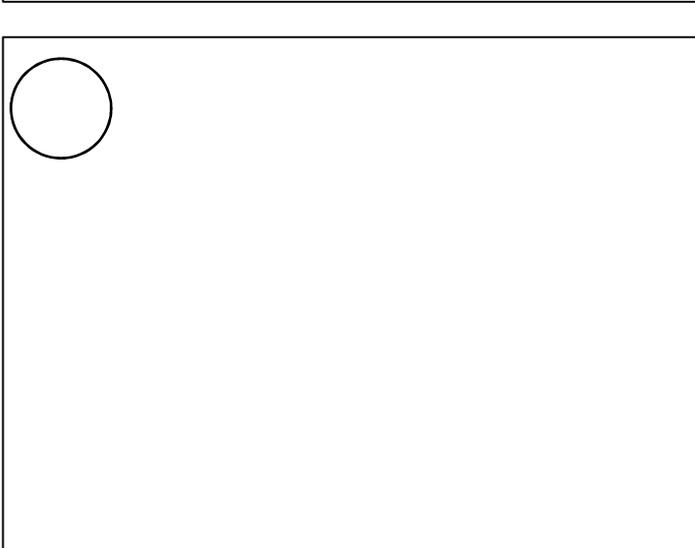
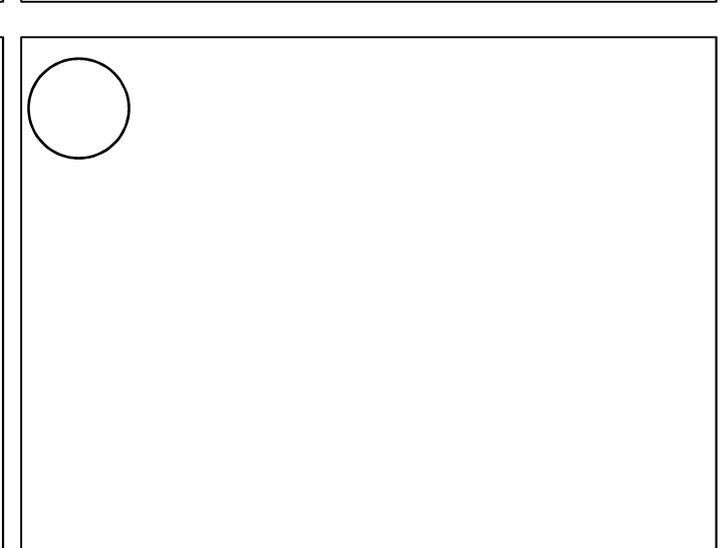
# Storytelling

## To Create: Storyboarding

### Task 3:

Time to design your own story with your own theme! (e.g. helping to improve the living of the people in need, recycling for protecting our environment etc.)

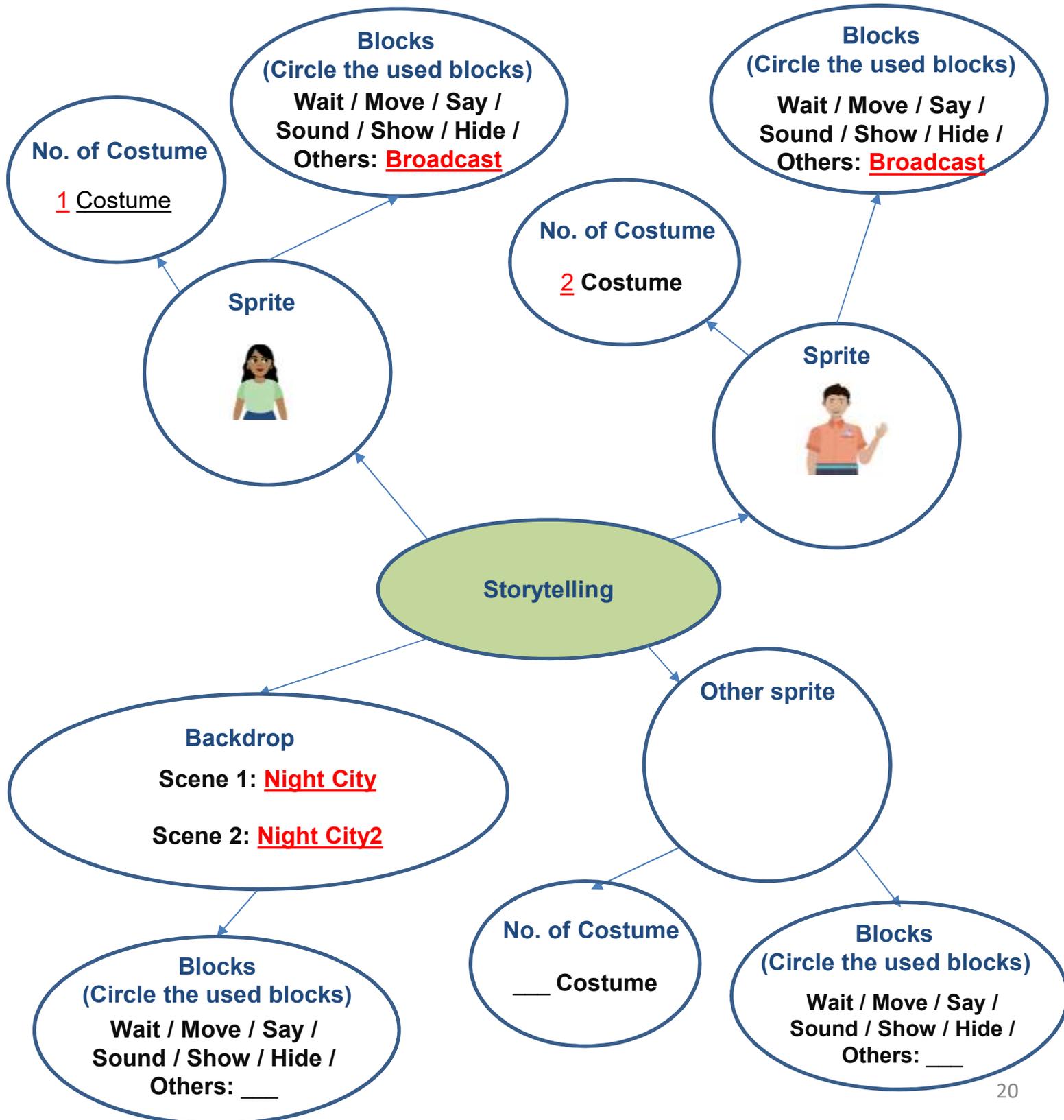
- ✓ Number the dialogues in the order they happen (1, 2, 3, 4...)
- ✓ Try to think about the **aesthetic aspects** (For example, the colour and position of your sprites, how they can form the colour harmony with the background...)
- ✓ Try to think about the theme and write down your ideas below:
  1. Describe the costumes/ motion of your sprites in the story.
  2. Introduce the design of using different backdrops

# Storytelling

❑ Complete the **mind map** below you start coding.

To cater for the students' ability, teachers can flexibly make use of P.20-P.22 to help students plan their story coding.



# Storytelling

Suggested answer only. Teachers should encourage students to create their own story by using the new features learnt from this unit.

Go to: <https://scratch.mit.edu/projects/768307650/>

Screen



Sasha



Character 2



```

when clicked
  switch costume to Sasha-c
  say Look! The night view of our city is so beautiful! for 3 seconds
  broadcast message1
  
```



```

when I receive message1
  say Yes! But remember we need to protect our environment! Save our energy! for 3 seconds
  broadcast message2
  
```



```

when I receive message2
  say How to do that? for 3 seconds
  broadcast message3
  
```

# Storytelling

Suggested answer only. Teachers should encourage students to create their own story by using the new features learnt from this unit.

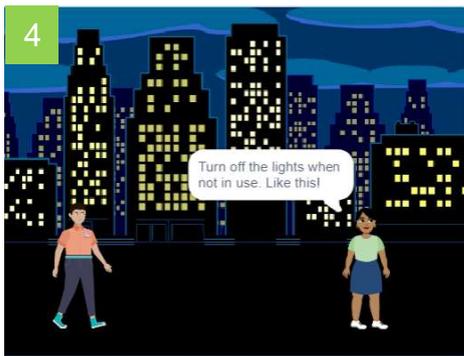
Screen



Sasha



Character 2



```

when I receive message3
say Turn off the lights when not in use. Like this! for 3 seconds
start sound Coin
switch backdrop to Night City2
wait 2 seconds
broadcast message4
    
```



```

when I receive message4
switch costume to Sasha-b
say Got it! for 2 seconds
    
```

# Storytelling

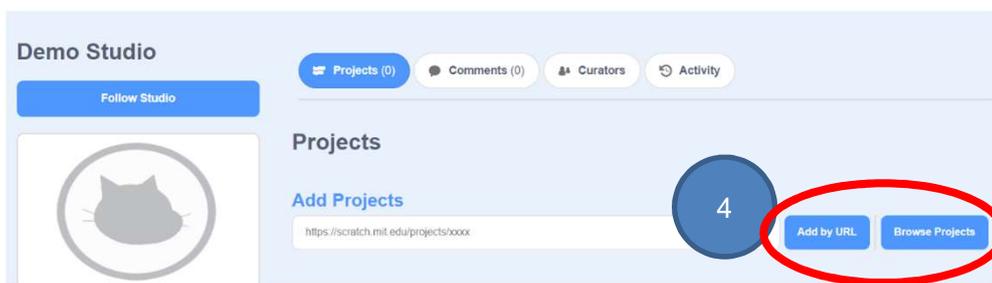
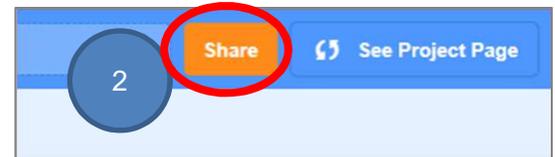
## To Create: Code and Tell Your Story

- ❑ Sign into your account at [scratch.mit.edu](https://scratch.mit.edu).
- ❑ Create a new project named Storytelling.
- ❑ Based on the storyboarding you made, start to create your own story with Scratch.
- ❑ Make use of “Broadcast” & “When I receive block” to complete the conversation.

## Sharing to Studio and Provide Constructive Feedback on Program Design

When you finish, you will add your project to your teacher’s Studio.

1. Save your project by clicking “Save now” under the File menu.
2. Click the orange “Share” button.
3. Go to your teacher’s Studio (they will give you a URL).
4. In the “Add projects” column, you can Add by URL or Browse Projects.



5. If you choose Browse Project, then you will see all your shared projects. Find the right one and click the “+” to add it to studio.



Teacher may use feedback form e.g. “Two Star and A Wish”, to guide students having peer / self-evaluation and to think of ways to enhance their program, including the aesthetic aspect.

# Storytelling

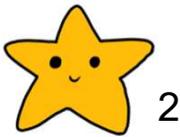
## To Reflect: Two Stars and a Wish Worksheet

Name of Project: \_\_\_\_\_ Name of Creator: \_\_\_\_\_

Please write down two things that you like about this project.



“Two Stars and a Wish” is a reflection strategy designed for student feedback as peer- and self-assessment. Teachers can guide students to give constructive feedback to their peers regarding their Scratch project - two positive (stars) and one hopeful (wish) reflection. Comments can be made on Scratch project’s idea, features and aesthetic aspects etc.

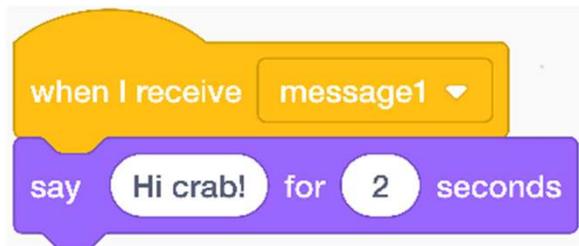
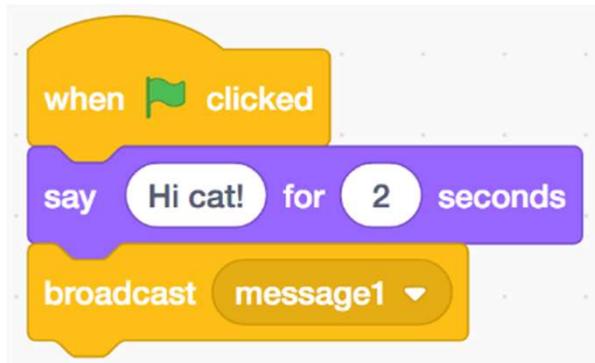


What is one thing you would like to add or change to make this project better?



## Review Questions

1. For the blocks below, what happens when you click the green flag?

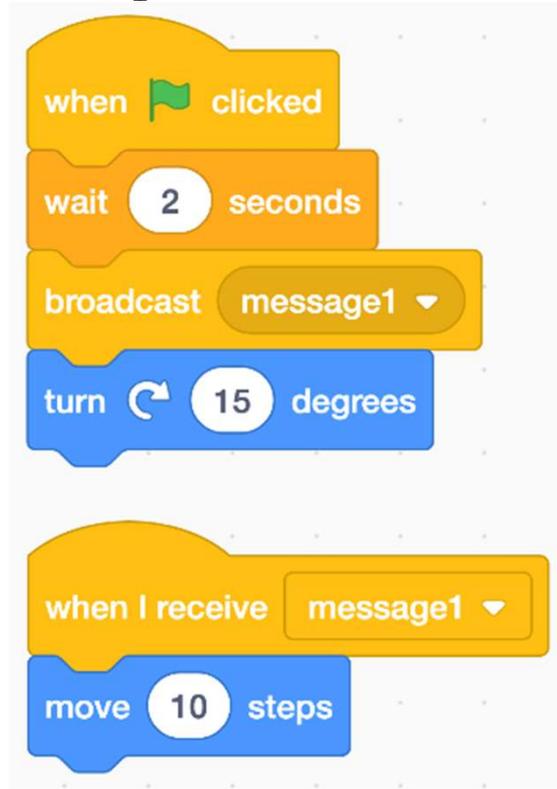


- A. The crab says “Hi cat!” for 2 seconds. The cat does nothing.
- B. The cat says “Hi crab!” for 2 seconds. The crab does nothing.
- C. The crab says “Hi cat!” and the cat says “Hi crab!” at the same time for 2 seconds.
- D. The crab says “Hi cat!” for 2 seconds, and then the cat says “Hi crab!” for 2 seconds.

**(Answer: D)**

## Review Questions

2. What does the crab do when the green flag is clicked?

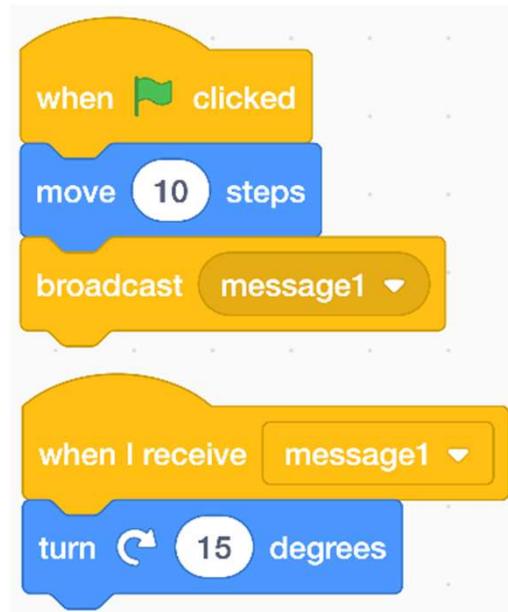
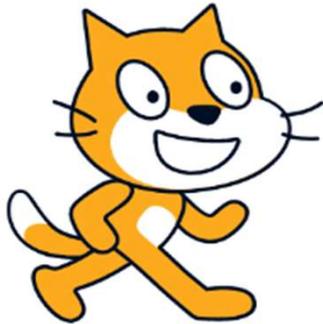


- A. After a 2-second delay, the crab moves 10 steps and turns 15 degrees.
- B. After a 2-second delay, the crab moves 10 steps.
- C. The crab moves 10 steps, waits 2 seconds, and then turns 15 degrees and moves another 10 steps.
- D. The crab moves 10 steps, waits 2 seconds, and moves another 10 steps.

**(Answer: A)**

## Review Questions

3. What does the cat do when the green flag is clicked?

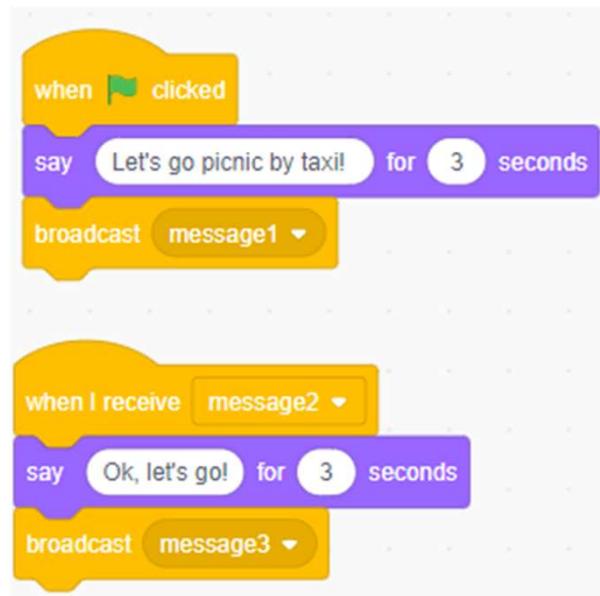
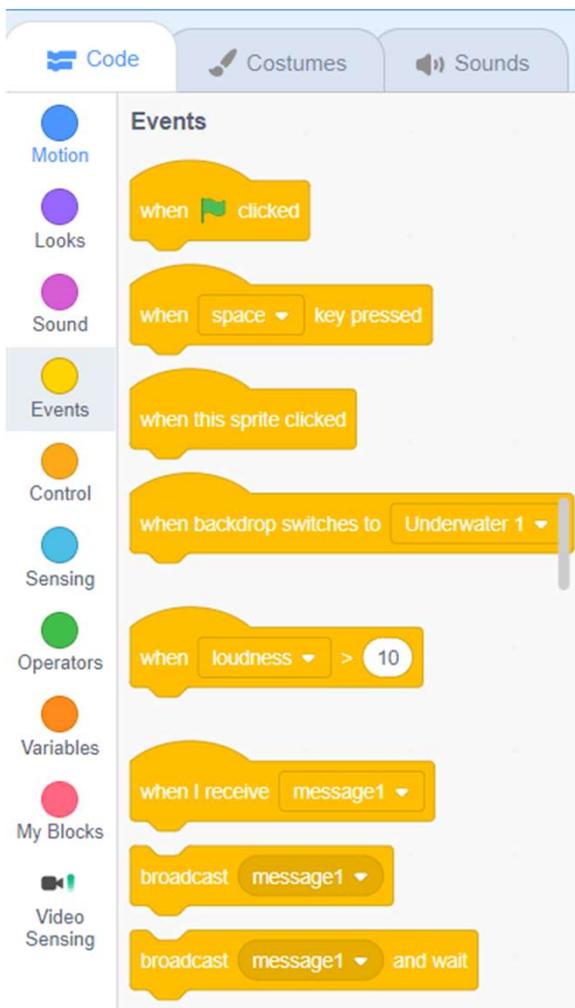


- A. The cat moves 10 steps and turns 15 degrees, waits 2 seconds, and then turns 15 degrees more.
- B. The cat moves 10 steps and turns 15 degrees, waits 2 seconds, and then moves another 10 steps and turns another 15 degrees.
- C. The cat moves 10 steps and turns 15 degrees.
- D. The cat turns 15 degrees.

**(Answer: C)**

## Revision on Key Features

- **Broadcast (message)**
- **When I receive (message)**



## Revision on Key Concepts & Practices

**Sequences:** It is a key concept in programming. It is the order in which the programming statements are executed. A wrong order would lead to incorrect programming results. For example, there is a specific sequence of dialogues for a meaningful conversation to go on.

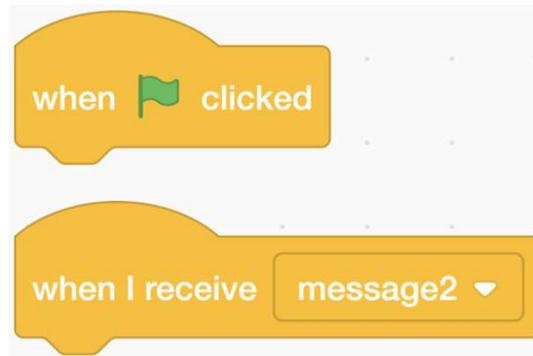


```
when clicked
say Let's go picnic by taxi! for 3 seconds
wait 6 seconds
say Ok, let's go! for 3 seconds
```

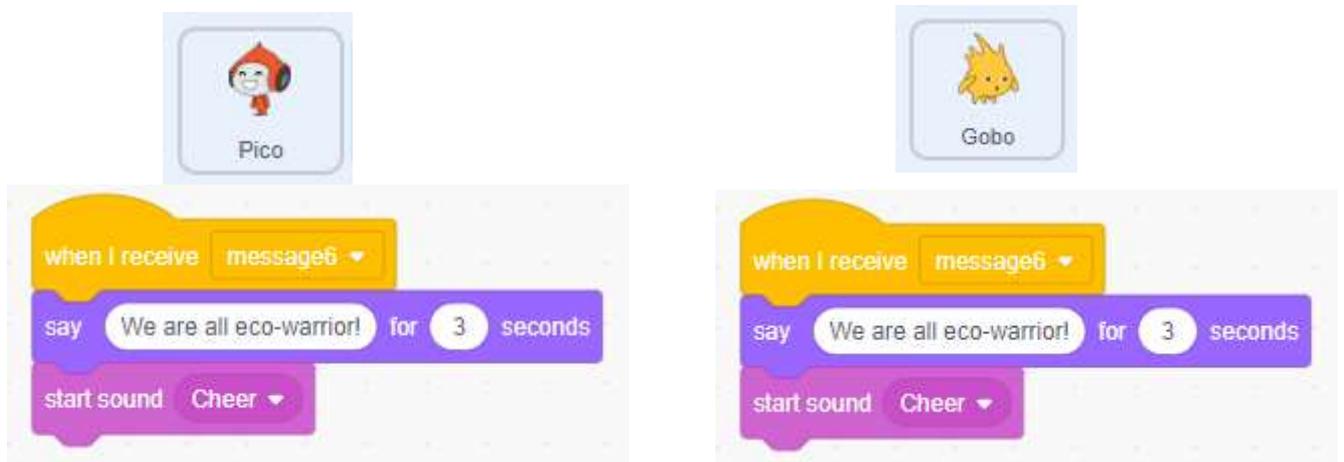
```
when clicked
switch costume to pico-b
wait 3 seconds
say Let's take the bus. for 3 seconds
say It is more environmental friendly. for 3 seconds
```

## Revision on Key Concepts & Practices

**Events:** We use event blocks to trigger Scratch to take actions.

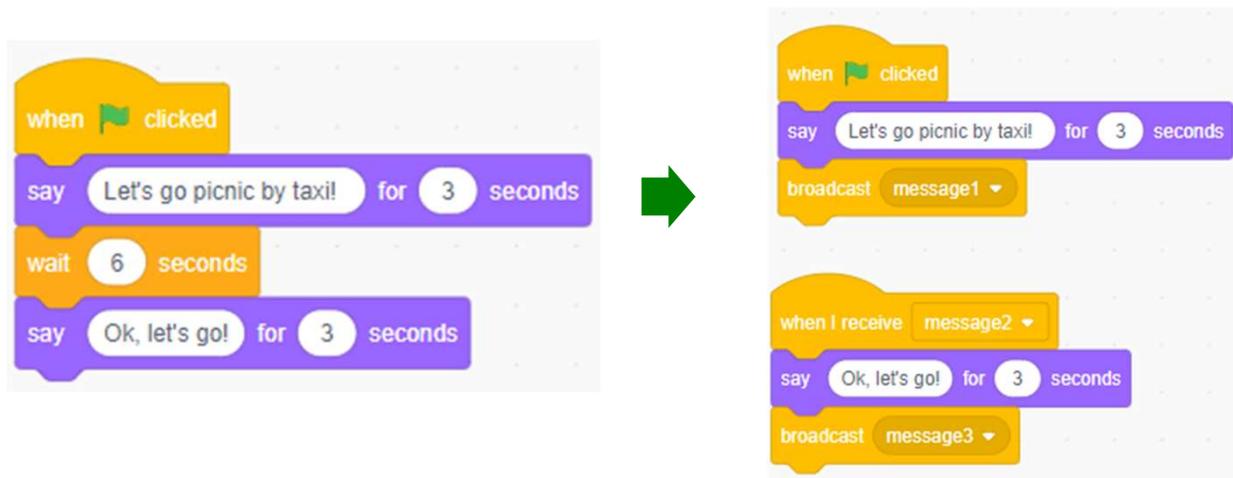


**Parallelism:** Scratch uses parallelism to allow more than one events to take place at the same time. For example, Scratch allows more than one character to perform an action at the same time in a Scratch program.

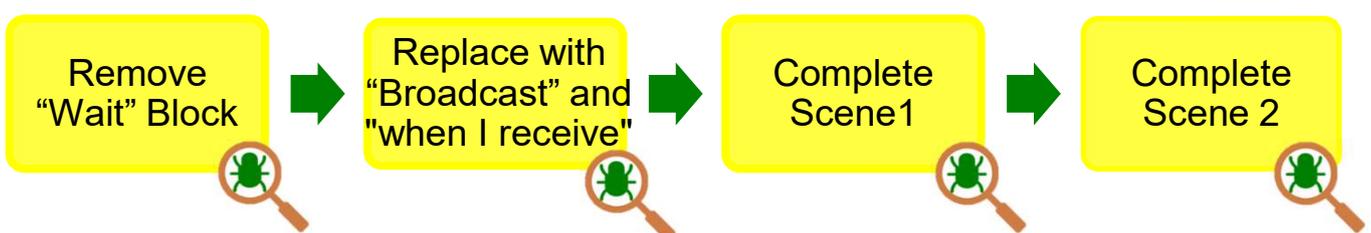


## Revision on Key Concepts & Practices

**Being incremental and iterative:** to work out a sub-task as an iteration, try it out, then work out another sub-task based on the codes of the previous sub-task in one more iteration until the whole programming task is completed.

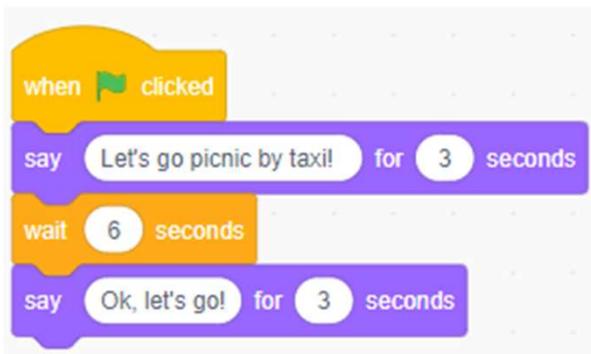


**Testing and debugging:** Testing a computer program is the process of checking if it can produce outcomes as designed. Debugging a computer program is the process of finding out ways to revise the program so that the bugs can be removed.

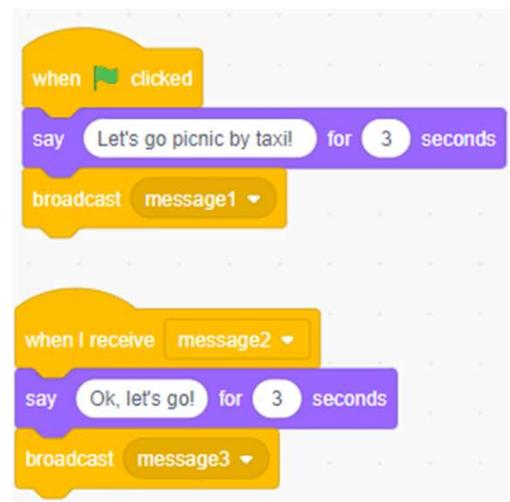


## Revision on Key Concepts & Practices

**Reuse and Remix programs/codes:** The reuse and remix of the works of other programmers are crucial in the online communities of Scratch. For example, we can reuse and remix the codes of a sprite such as the shark in Unit 2 and use them for the second and third sprites.



```
when clicked
say Let's go picnic by taxi! for 3 seconds
wait 6 seconds
say Ok, let's go! for 3 seconds
```



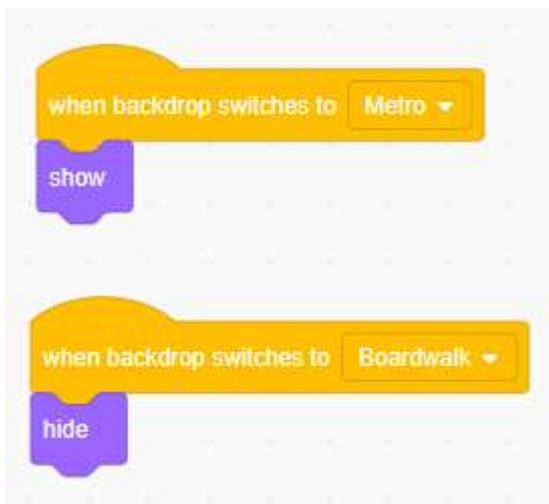
```
when clicked
say Let's go picnic by taxi! for 3 seconds
broadcast message1

when I receive message2
say Ok, let's go! for 3 seconds
broadcast message3
```

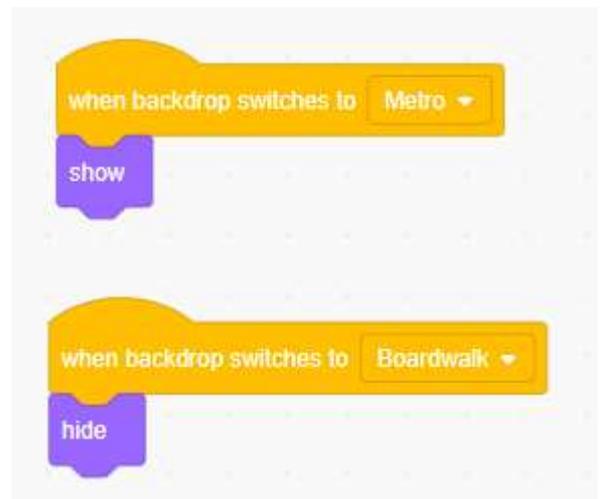
## Program Codes (Given in the template):



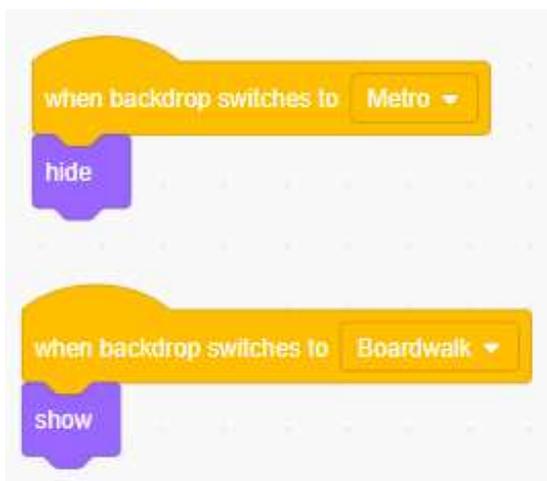
Sprite (City Bus)



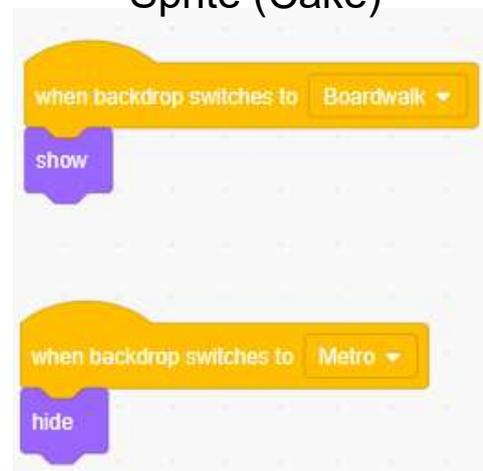
Sprite (Convertible)



Sprite (Donut)



Sprite (Cake)



# Storytelling

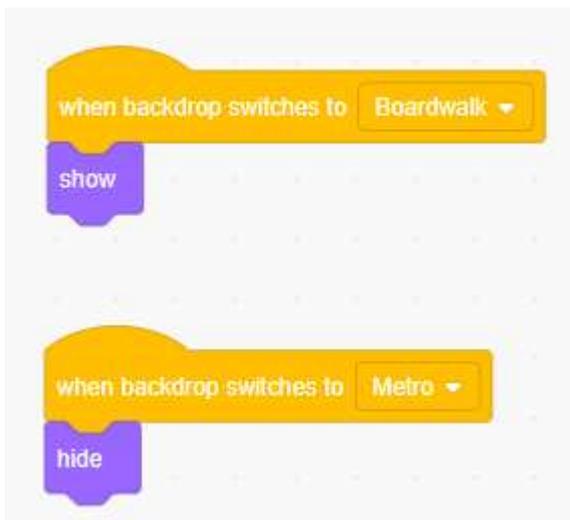
## Program Codes (Given in the template):



Sprite (Taco)



Sprite (Food Truck)



# Storytelling

## Program Codes:



Sprite (Gobo)

```
when clicked
  switch backdrop to Metro

when I receive message4
  say But I am full.. for 3 seconds
  broadcast message5

when clicked
  say Let's go picnic by taxi! for 3 seconds
  broadcast message1

when I receive message6
  say We are all eco-warrior! for 3 seconds
  start sound Cheer

when I receive message2
  say Ok, let's go! for 3 seconds
  broadcast message3
  switch backdrop to Boardwalk
```

\*Student only need to complete the code “Broadcast” and “When I receive”.

# Storytelling

## Program Codes:



Sprite (Pico)

```
when clicked
  switch costume to pico-b

when I receive message1
  say Let's take the bus. for 3 seconds
  say It is more environmental friendly. for 3 seconds
  broadcast message2

when I receive message6
  say We are all eco-warrior! for 3 seconds
  start sound Cheer

when I receive message3
  switch costume to Pico-d
  say Why are there so much leftover food? for 3 seconds
  say It is not environmental friendly! for 3 seconds
  broadcast message4

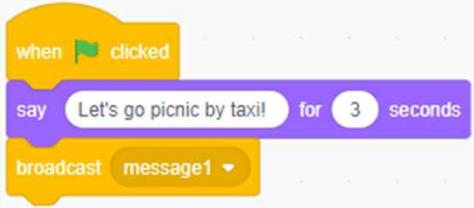
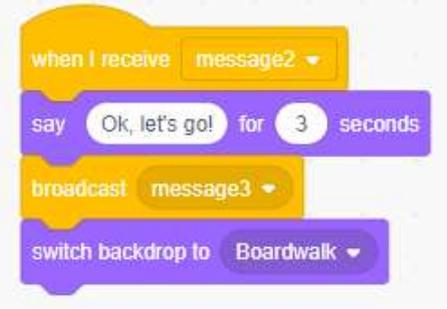
when I receive message5
  switch costume to pico-b
  say Then don't order too much food next time. for 3 seconds
  broadcast message6
```

\*Student only need to complete the code “Broadcast” and “When I receive”.

# Storytelling

## Program Codes (By Scene):

### Scene 1

Screen	 Gobo	 Pico
<p>1</p> 		
<p>2</p> 		
<p>3</p> 		

# Storytelling

## Scene 2

Screen



Gobo



Pico

4



```

when I receive message3
  switch costume to Pico-d
  say Why are there so much leftover food? for 3 seconds
  say It is not environmental friendly! for 3 seconds
  broadcast message4
  
```

5



```

when I receive message4
  say But I am full.. for 3 seconds
  broadcast message5
  
```

6



```

when I receive message5
  switch costume to pico-b
  say Then don't order too much food next time. for 3 seconds
  broadcast message6
  
```

7



```

when I receive message6
  say We are all eco-warrior! for 3 seconds
  start sound Cheer
  
```

```

when I receive message6
  say We are all eco-warrior! for 3 seconds
  start sound Cheer
  
```

# Unit 4: Space Travelling Teacher Guide

## Content

<b>Teaching Plan</b>	P4
<b>Lesson 1</b>	
To Play	T4-1
To Think	T4-2
To Code	
Adding the Backdrop and Sprite	T4-4
Add Video Sensing	T4-5
Change Costumes	T4-7
Glide in the Space - glide to random position	T4-11
<b>Lesson 2</b>	
To Learn	
Unplugged Activity - Branching / Selection / Conditionals	T4-12
To Code	
Change Backdrop	T4-15
Iteration	T4-19
Add Wait Block	T4-20
Add Sound Effect (Sprite)	T4-21
Add Sound Effect (Backdrop)	T4-21
To Reflect	T4-22

### **Lesson 3**

To Create T4-23

To Reflect T4-25

Review Questions T4-26

Revision on Key Features T4-29

Revision on Key Concepts & Practices T4-30

**Appendix - Operation Manual** T4-34

**Program Codes** T4-40

## Unit 4: Space Traveling Teaching Plan

### Prior Knowledge

Students know how to add sprites, change the sprite's costumes and add sounds. They should also know how to use some coding blocks, including “wait”, “repeat” and “forever” blocks.

### Learning Objectives

1. Create an interactive Scratch project using video sensing;
2. Apply the knowledge and skills on coding and computational thinking previously learnt to design a Scratch project;
3. Use a conditional correctly within a project to trigger an action;
4. Design a project that can interact with users through the Scratch programming environment and interface;
5. Foster students' creativity by creating their own coding artefacts and understand that coding can be a fun and social experience through sharing their Space Traveling projects with others.

### Learning Elements

#### Computational Thinking Concepts and Practices:

Key Learning Elements	Items
Abstraction	Express the algorithm
Algorithm	Problem Solving Procedures: Problem identification, Problem analysis, Algorithm design, Programming Basic Programming Constructs: Sequence, Branching / Selection, Iteration Coding Concept and Practices: Design, Reuse and Remix programs / codes, Testing and Debugging

#### Coding Skills:

1. Make use of video sensing commands;
2. Understand the Basic Programming Constructs: Branching / Selection and Iteration;
3. Understand “>”, “<” and “=” operators through coding;
4. Apply Testing and Debugging in completing each task of this unit.

#### Others (including Attitude):

1. Develop interest in programming;
2. Show perseverance and positivity in testing and debugging;
3. Inspire students to be creative and innovative to enhance their Space Traveling projects.

**Lesson Plan:** This unit consists of 3 lessons of 35 minutes.

**Teacher Preparation for the Lesson:** Prepare appropriate computer or mobile device with Camera for students to detect movement in Scratch. Prepare accounts for students to login Scratch. Prepare a studio for students to share their projects.

### Lesson 1

Time	Activity
5 mins	<b>To Play: Help students gain an initial understanding of the program</b> <ol style="list-style-type: none"><li>1. Play the Scratch project “Space Travelling”: <a href="http://scratch.mit.edu/projects/727401089">http://scratch.mit.edu/projects/727401089</a>.</li><li>2. Ask students to turn on the computer webcam.</li><li>3. Wave their hands in front of the webcam and observe what happens.</li><li>4. Ask them questions e.g. “What happens to the sprite and backdrops?” or “Do you hear any sound effect?”</li></ol>
10 mins	<b>To Think</b> <ol style="list-style-type: none"><li>1. Ask student to complete the mind map before they start to code.</li></ol>
20 mins	<b>To Code: Add Video Sensing and Change Costumes</b> <ol style="list-style-type: none"><li>1. Introduce the new features used in this unit – Video Sensing.</li><li>2. Guide the students to explore the video sensing feature and code to change sprite’s costumes.</li><li>3. Adjust the video motion value if students want after they test, write down their observations.</li><li>4. Encourage the students to find the relationship between video motion value on webcam, teachers can then explain.</li><li>5. Complete the rest of coding, e.g. make the sprite glide to random position.</li></ol>

## Lesson 2

Time	Activity
15 mins	<p><b>To Learn</b></p> <ol style="list-style-type: none"><li>1. Run the “Unplugged Activity: Guessing the Right Number”, which aims to let the students learn conditional expression and operators.</li><li>2. Explain to students the concept of Branching/Selection and the conditional expressions using operators such as the greater than (&gt;), less than (&lt;) and equal (=).</li></ol>
20 mins	<p><b>To Code: Change backdrop</b></p> <ol style="list-style-type: none"><li>1. Ask student to complete the flowchart before they start to code.</li><li>2. Continue to code with the help of the student guide.</li><li>3. Change Backdrop:<ol style="list-style-type: none"><li>1) <u>Video motion value</u>: Encourage students to try a different value to change the backdrop and do the testing. Tell students that they will use a larger value than the one they used in changing sprite’s costumes.</li><li>2) <u>Forever block (Iteration)</u>: After testing, they will find out the blocks would only work once and they should add a “forever” block to keep the program running.</li><li>3) <u>Wait block</u>: During testing, they will realise the backdrop could be changing too fast. So, they will go for debugging and add a “wait” block that they have learnt it in previous unit.</li></ol></li><li>4. Add sound effect for the sprite and backdrop to make this space travel trip interesting.</li></ol>

### Lesson 3

Time	Activity
25 mins	<p><b>To Create: Make your own project</b></p> <ol style="list-style-type: none"><li>1. Ask students to create their own project. Browse Scratch library (e.g. Sports category) to get more ideas on what they want to create.</li><li>2. Draw the new project ideas and the algorithm design/ mind map on the student guide. They are encouraged to complete the coding task on their own.</li></ol>
10 mins	<p><b>To Reflect:</b></p> <p><b>Share the Projects and Provide Constructive Feedback on Program Design</b></p> <ol style="list-style-type: none"><li>1. Create a studio and give students the studio URL. Ask students to save and submit their projects to the teacher's Studio.</li><li>2. Have one or two students share their project with the whole class. Feedback to be given by peer and teachers.</li><li>3. Students should share, think of ways to improve / enhance their program as well as the aesthetic values.</li></ol> <p><b>Review of Student Learning</b></p> <ol style="list-style-type: none"><li>1. Review on the features of Scratch, and key concepts and practices learnt in the unit.</li><li>2. Ask students to complete the review questions, appropriate feedback should be given by teachers.</li></ol>

# Space Travelling

Let's learn how to travel in space with Scratch!

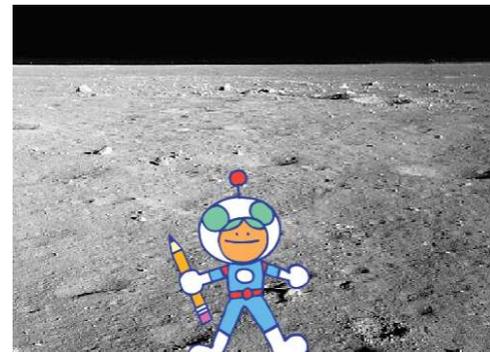
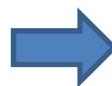
Through this trip, you will learn how to change the costumes of the sprite using the video sensing feature in Scratch.



## To Play

Play the Space Travelling (Demo): <https://scratch.mit.edu/projects/727401089>.

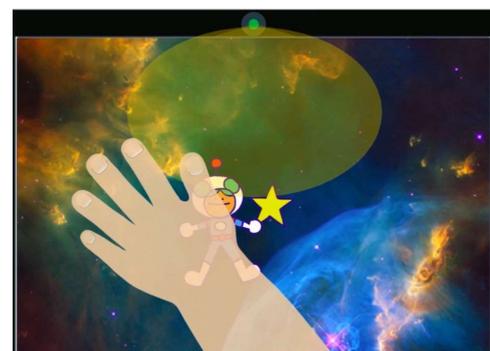
Try to **move your hand** in front of the webcam of your computer.



What happened to **the sprite and backdrop**?

How fast did you **wave your hand**?

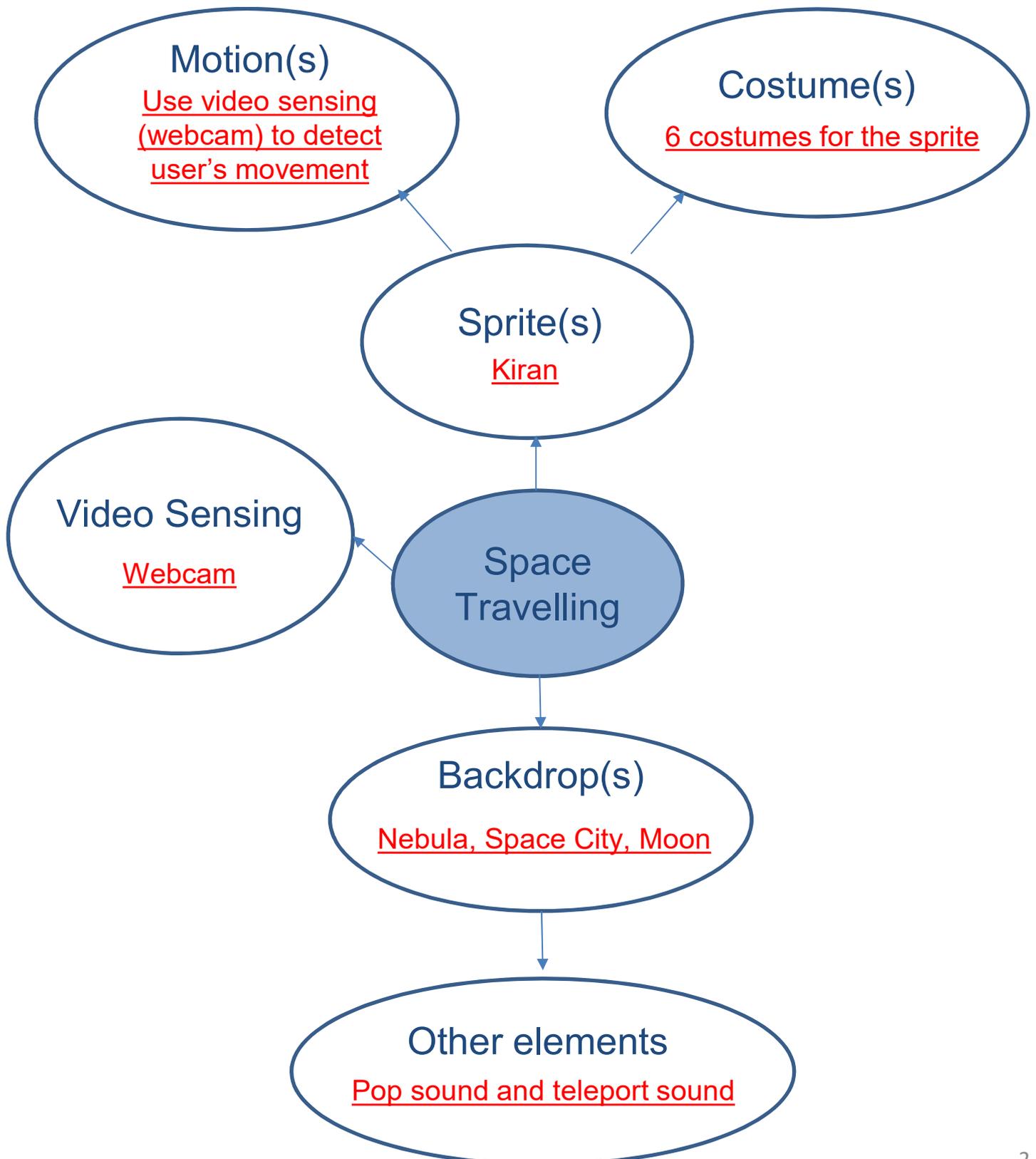
Did you hear any **sound** effect?



# Space Travelling

## To Think

Complete the mind map below before your start to code.

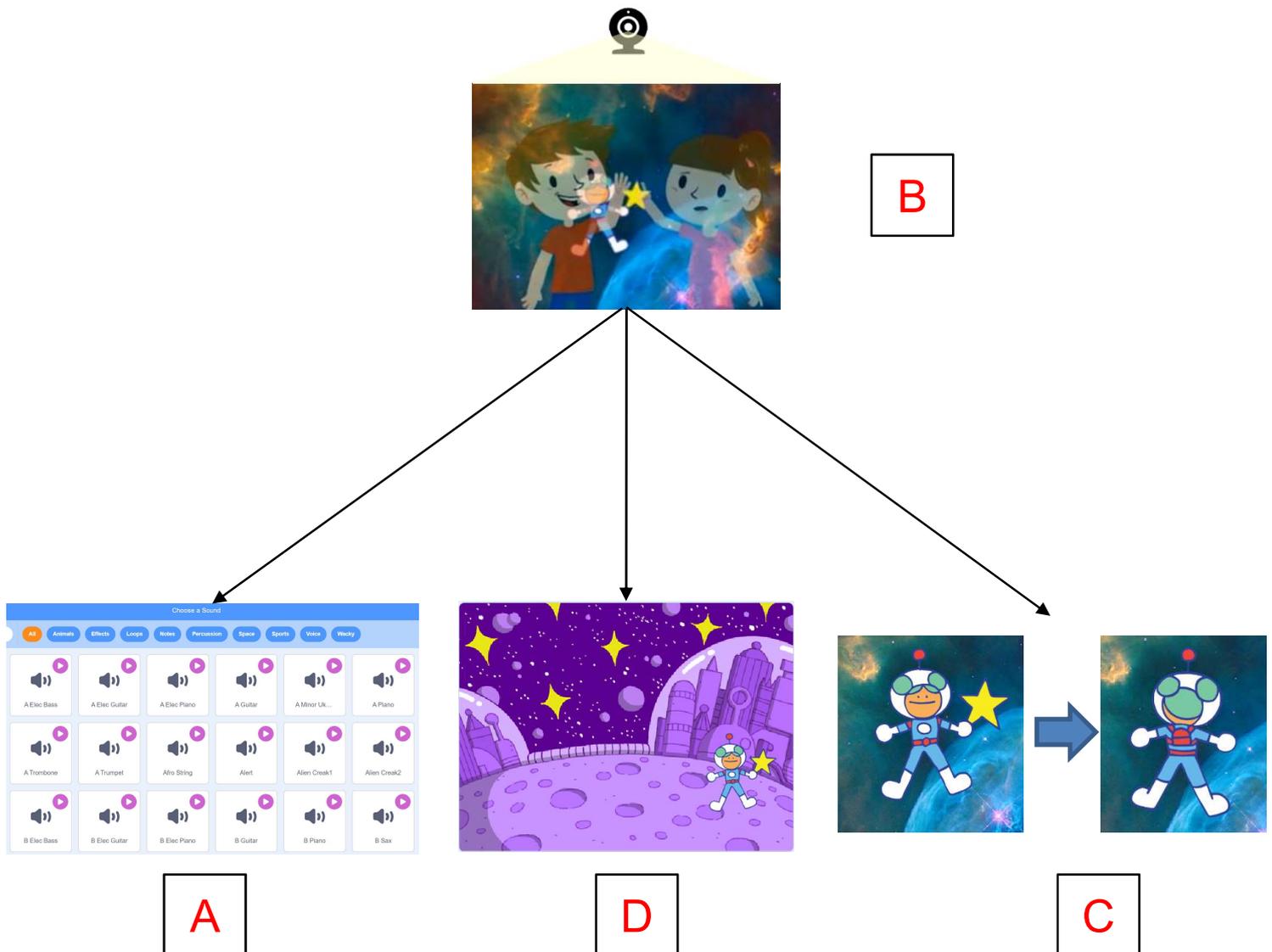


# Space Travelling

## To Think

What happened when you wave your hand in front of the camera?  
Can you fill in the actions triggered below?

(A) Play Sound Effect	(B) Webcam detecting your movement
(C) Change Costumes	(D) Change Backdrop



# Space Travelling

## To Code: Adding the Backdrop and Sprite

1. Turn on the webcam of your computer.

See Appendix  
P.35-36



2. Sign in and create a new project. Name it **Space Travelling**.



3. Choose a good backdrop.



4. Add a sprite and some different costumes for it.



Change its  
size?

# Space Travelling

## To Code: Add Video Sensing

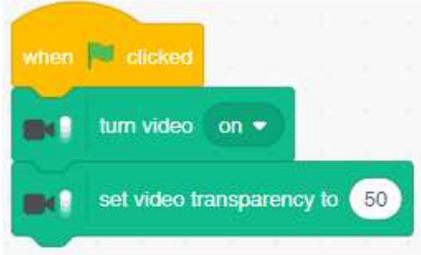
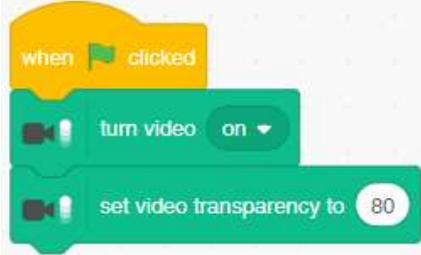
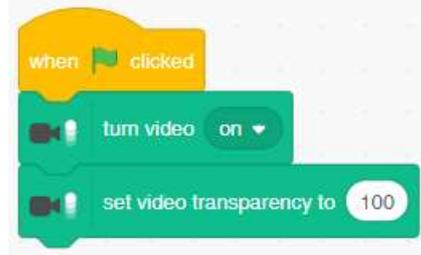
Let's find the video sensing blocks for space travelling!

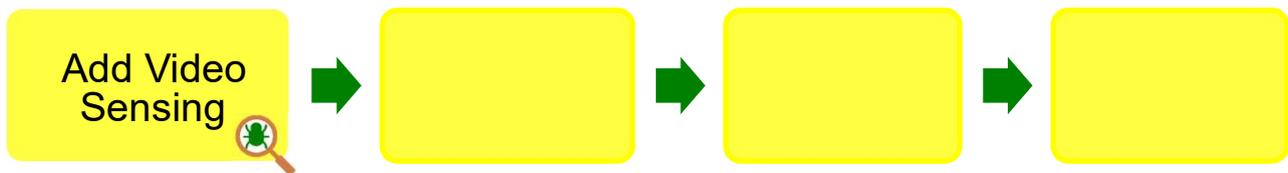
1. Click on the "Add Extension" icon at the bottom left of the page.
2. Choose the "Video Sensing" feature.
3. You will see a list of Video Sensing blocks in green.

The image illustrates the process of adding the Video Sensing extension in Scratch. It shows the Scratch interface with the 'Add Extension' icon circled in red (1). The 'Choose an Extension' dialog box is shown, with the 'Video Sensing' option circled in red (2). The 'Video Sensing' block palette is shown, with the 'Video Sensing' category icon circled in red (3). A 'when video motion > 10' block is selected, and the 'turn video on' block is circled in red (4). The 'turn video on' block is being dragged into a script area, where it is placed below a 'when clicked' block (5).

## To Code: Add Video Sensing

Let's try these three events and see what will happen?

1	2	3
		



### Testing and Debugging

Let's test it! Click the green flag.  
Try to **adjust the video transparency for your own project!**



For Teachers' reference:

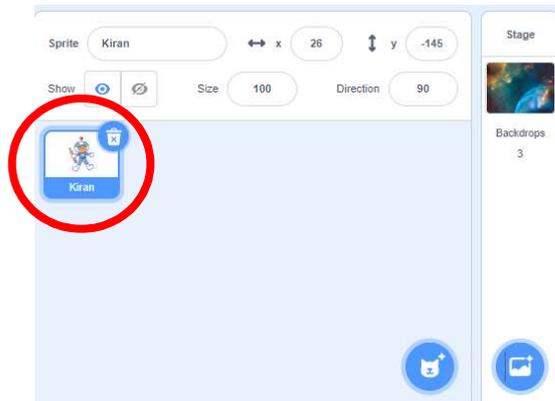
Try to enter different values in the box. Every time you change the value, click on the "when green flag clicked" block to see what happens.

- Set the transparency to 100 to hide the current video output.
- Set the transparency to a smaller value to make the current video be part of the background of your project.
- Set the transparency to 0 to make the camera video fully visible.

# Space Travelling

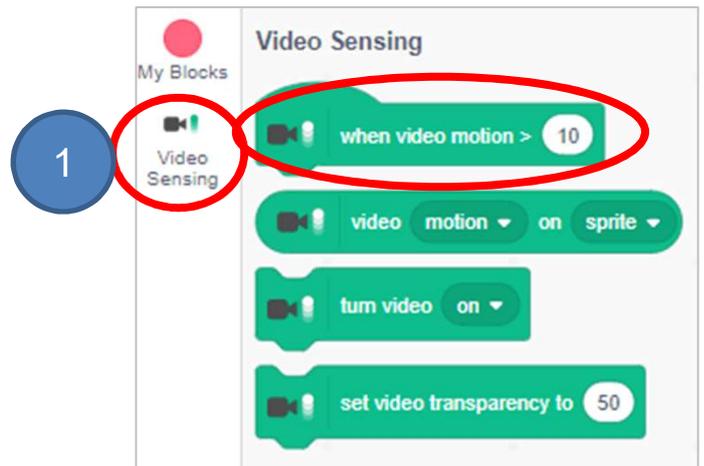
## To Code: Change Costumes

Click the “**Sprite**”, start to code for the sprite!

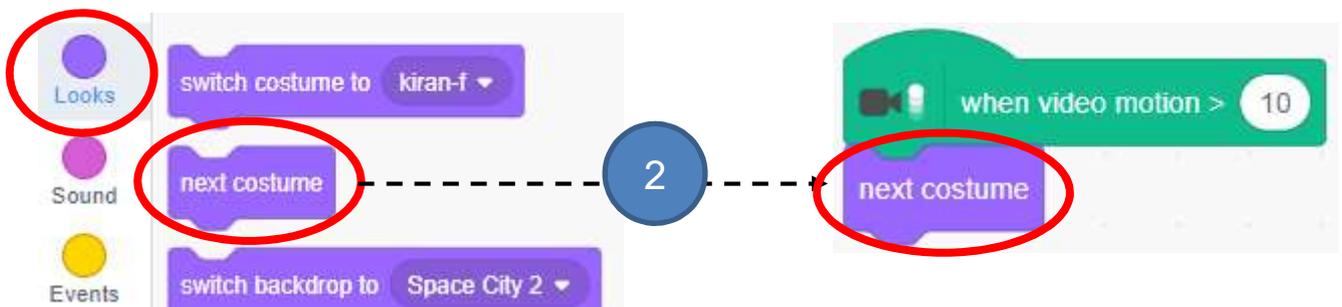


We want the sprite to do the action when the video motion meets the threshold.

1. Drag out a “when video motion > 10” block.

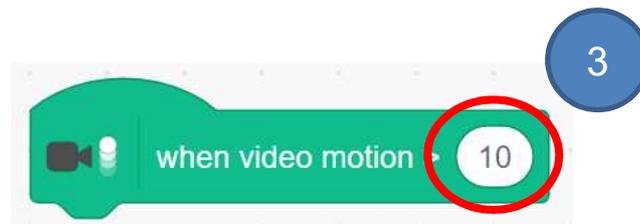


2. To change the costumes of your sprites.



## To Code: Change Costumes

3. Try to change the value of video motion level (e.g. 10, 30, 100 or other values). Wave your hand in front of the camera to see the difference. (Hint: 1 will start with very little movement, 100 requires a lot of movement.)



Please circle your observation below:

Value of Video Motion	e.g. >60	>30	>100	
What if you move your hand slower?	Costume changes / <u>does not change</u>	Costume changes / does not change	Costume changes / does not change	
What if you move your hand quicker?	<u>Costume changes</u> / does not change	Costume changes / does not change	Costume changes / does not change	

The Video Sensing feature in Scratch, detects the change of each pixel in the camera. The value of “video motion” means how many pixels are changing. Actually, our range and speed (even near and far) of movement mean different values. Students are encouraged to try different values to explore the differences by themselves. Teachers may only elaborate if necessary.

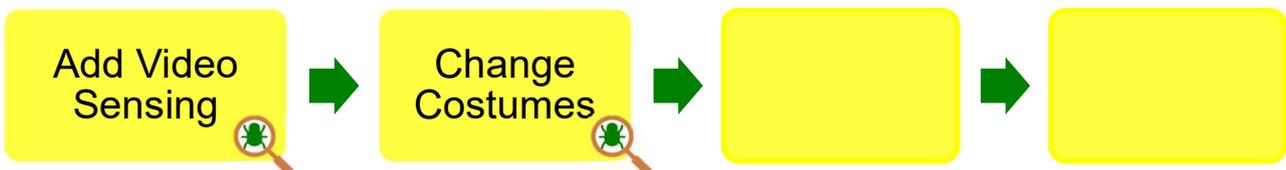
## To Code: Change Costumes



### Testing and Debugging

Let's test it!

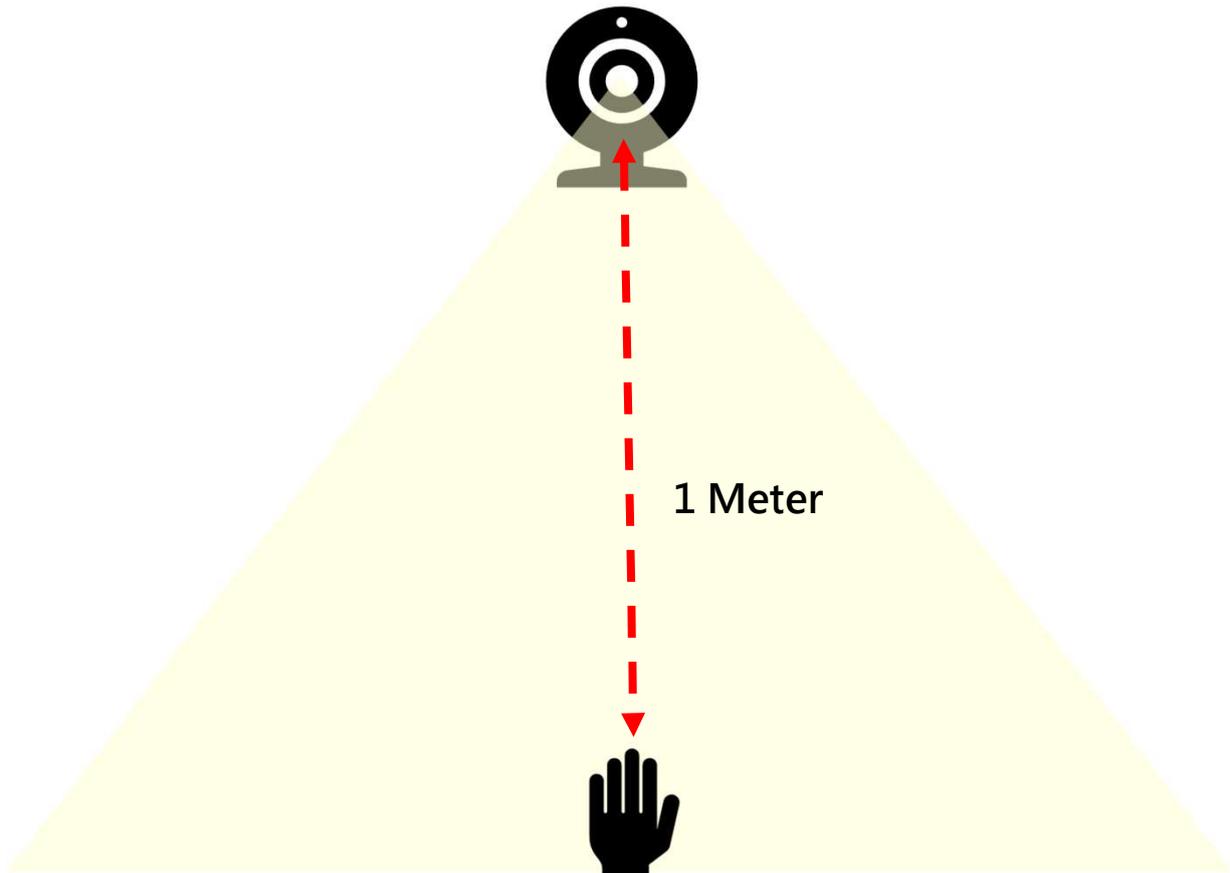
If you enter a smaller value for the sprite, try to move your hand slowly.  
Did you change it to the next costume?



Tell students that they can move hands quicker or slower to the webcam to see what happens.

## Relationship between Video Sensing and Video Motion

Place your hand 1 meter away from the webcam as follows:



Move hands quicker or slower in front of the webcam to see what happens.

**Conclusion:** The quicker you move hands, the bigger value of video sensing. Teachers can tell students to test and find the suitable speed.

# Space Travelling

## To Code: Glide in the Space

Try to make your sprite **glide** to **random position** when the video motion is triggered!

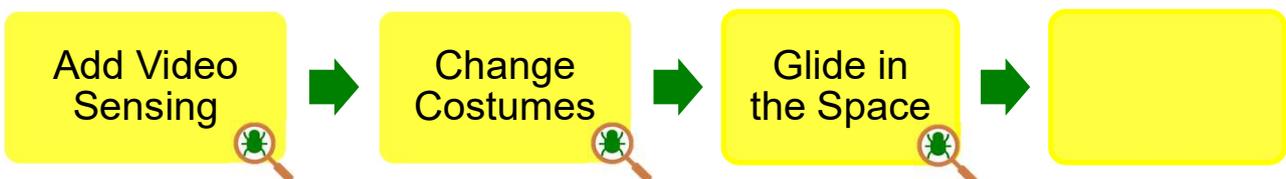
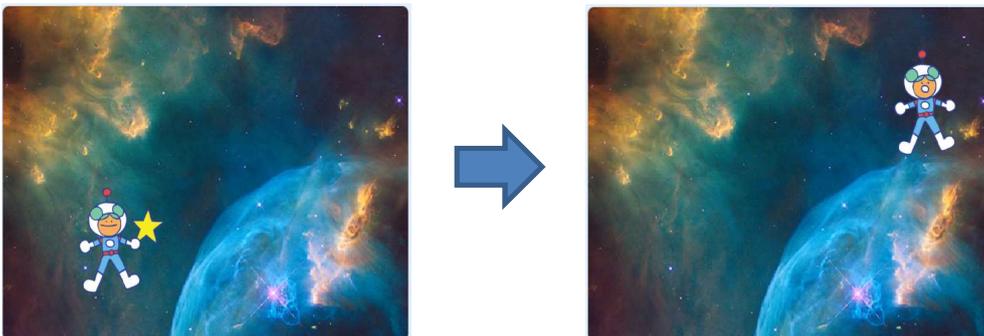
Which one should be used?

The image shows a Scratch code editor interface. On the left, a red dashed box highlights four code blocks from the Motion and Looks categories: 'go to random position', 'go to x: 26 y: -145', 'glide 1 secs to random position', and 'glide 1 secs to x: 26 y: -145'. An arrow points from this box to a separate code block on the right. This block is a green 'when video motion > 30' block with a purple 'next costume' block attached to it. Below these is a blue block with a red question mark, indicating a choice of what to do next.



### Testing and Debugging

Wave your hand to test the project, see if the sprite **changes its costume** and **glides to random position**.



## To Learn

### Unplugged Activity: Guessing the Right Number

How to play?

A teacher thinks of a number between 1-25. Have some students try to guess the number by asking a series of questions. The teacher reacts according to students' questions.

Example:

Teacher selects (20).

Student: If the number is greater than ( $>$ ) 13, please raise your hand.

Teacher: (Raises his/her hand)

Student: If the number is greater than ( $>$ ) 19, please raise your hand.

Teacher: (Raises his/her hand)

Student: If the number is greater than ( $>$ ) 22, please raise your hand.

Teacher: (No action)

Student: If the number is greater than ( $>$ ) 20, please raise your hand.

Teacher: (No action)

Student: If the number is less than ( $<$ ) 20, please raise your hand.

Teacher: (No action)

Student: If the number is equal to ( $=$ ) 20, please raise your hand.

Teacher: (Raise his/her hand).

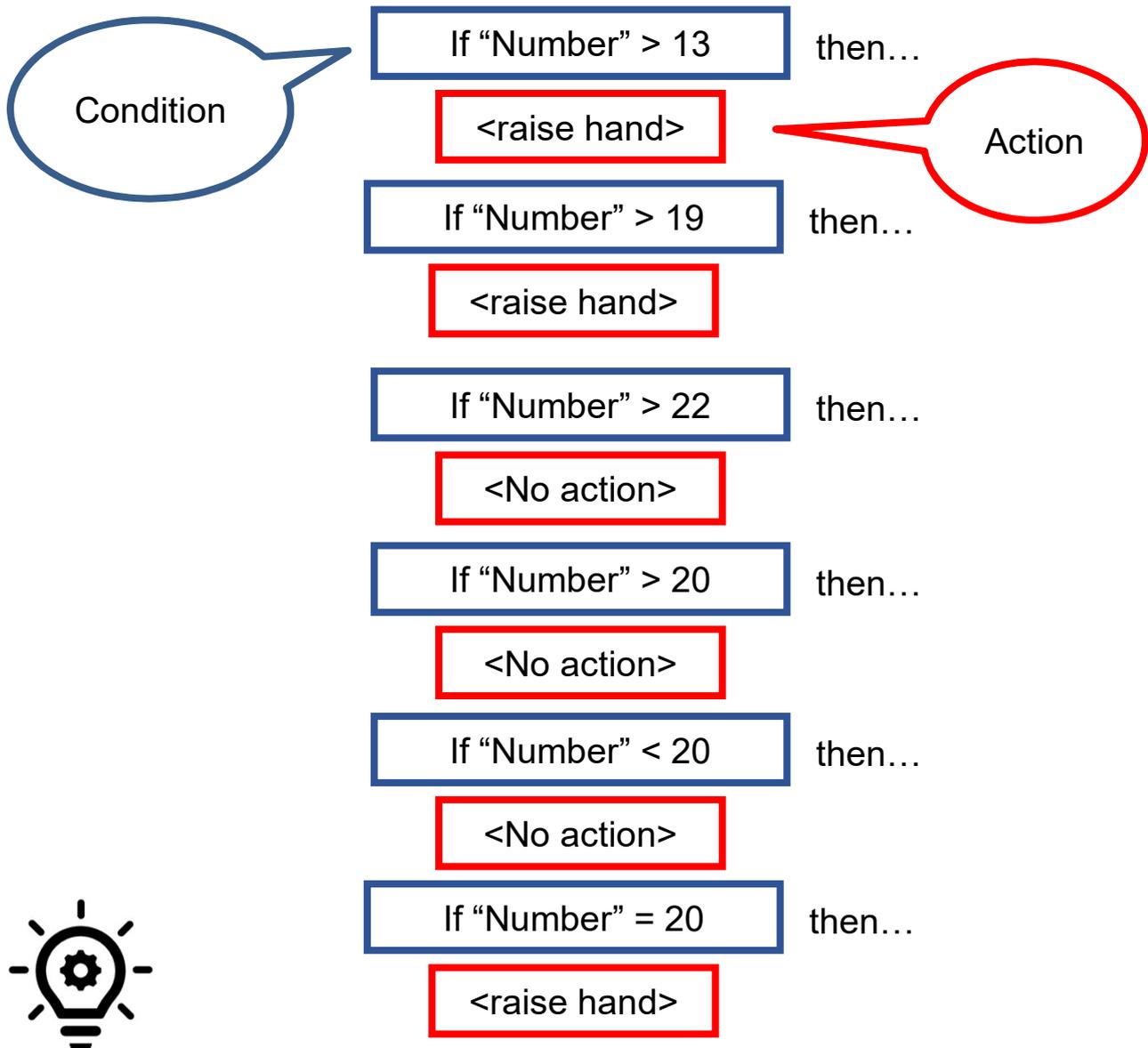
This game aims to let students learn to check their guess with a conditional expression. Students are expected to have an understanding of the concepts of conditional expressions using operators.

## To Learn

### Unplugged Activity: Reflection

In the game, you learn to check your guess with a conditional expression and the teacher **reacts** based on the **condition** you provided in the question.

Do you remember the conditional expressions made?



#### **Knowledge builds up: Branching / Selection / Conditionals**

We use conditional statements in programming to enable computers to make decisions. Conditionals always have an "if" part, which tells the program in the "then" part what to do when the condition is true.

## To Learn

Conditional Operators	Meaning
$>$	Is greater than
$<$	Is less than
$=$	Is equal to



### Knowledge builds up: Conditional Operators

Conditional Operators: We use operators to evaluate whether a condition is true or false. Conditional expressions always use operators such as greater than ( $>$ ), less than ( $<$ ) or equal ( $=$ ).

Question:

There is a height requirement for roller coasters at a theme park for safety reason. You need to be taller than 112cm in order to play those roller coasters.

Which **conditional operator** can be used to express this height requirement?

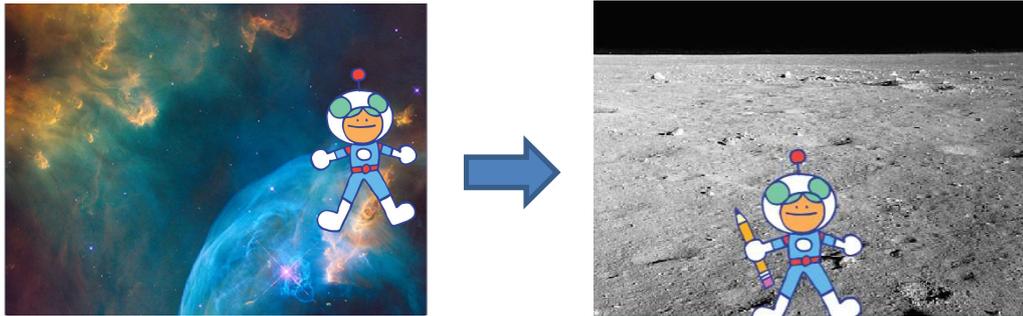
A. Height  $>$  112 cm

B. Height  $<$  112 cm

C. Height  $=$  112 cm

# Space Travelling

Let's continue to travel in space with Scratch!  
We will travel to other planets when we move our hand fast.



## To Code: Change Backdrop

Try to add a total of 3 backdrops for your space travelling!

See Appendix  
P.37



# Space Travelling

## To Code: Change Backdrop

Now, you can see a set of backdrop in [backdrops](#).

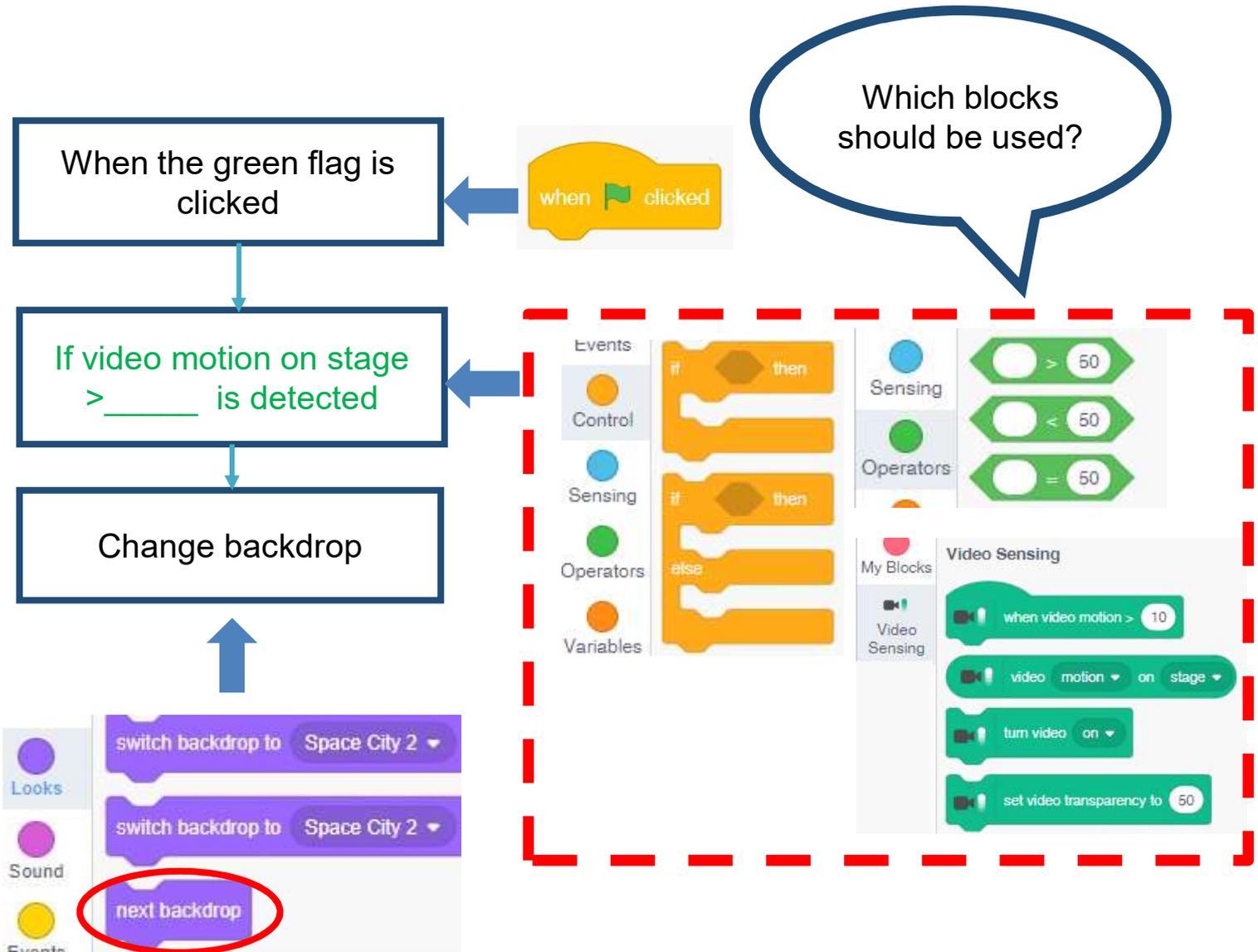
What is the sequence about changing Backdrop of this project? Please fill in the Flowchart.



# Space Travelling

## To Code: Change Backdrop

Let's have a look at the codes! Which blocks should be used to trigger the actions we want?



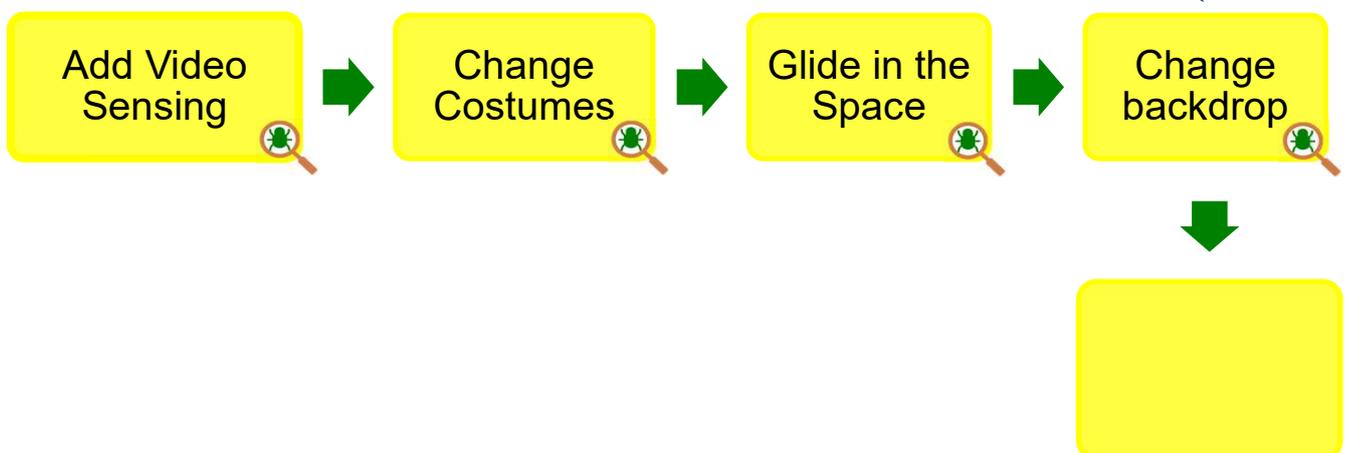
# Space Travelling

Try different values and see what happens! Try to enter 100? 200? Or more?



## Testing and Debugging

Let's do the testing. What happened after you clicked the green flag?



# Space Travelling

## To Code: Iteration

Control

wait 1 seconds

repeat 10

forever

when clicked

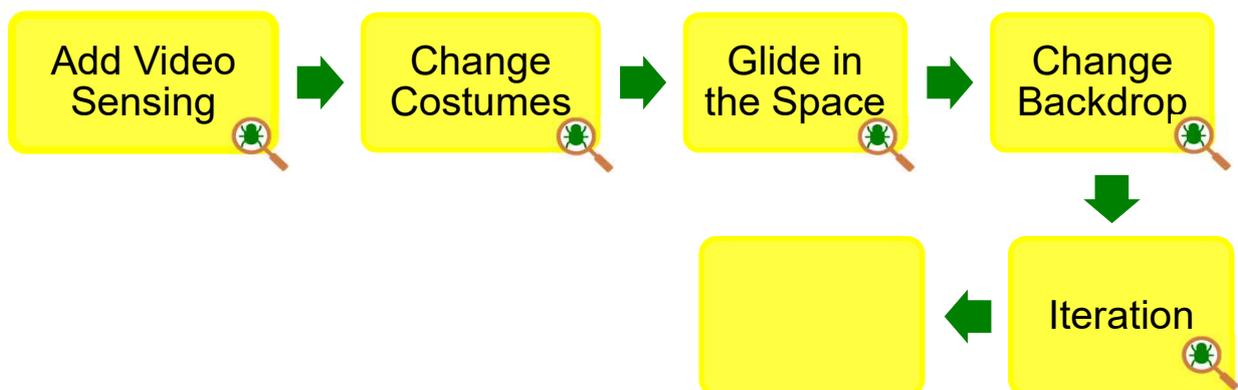
?  
if video motion on stage > 350 then  
next backdrop

Which block should be added to continuously check for the condition?



### Testing and Debugging

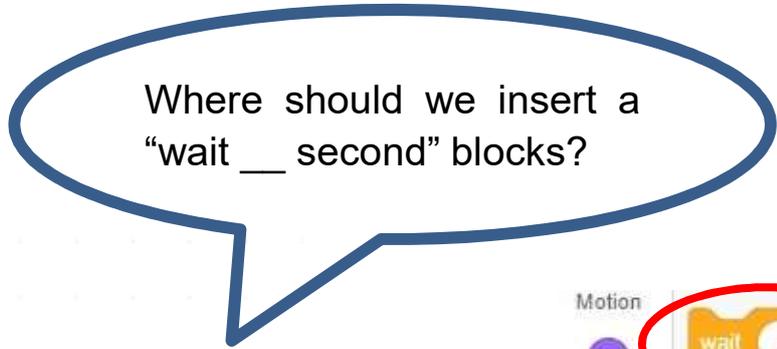
Test it again. Is the changing speed too fast or slow?  
How to solve this issue?



# Space Travelling

## To Code: Add Wait Block

Where should we insert a “wait \_\_ second” blocks?

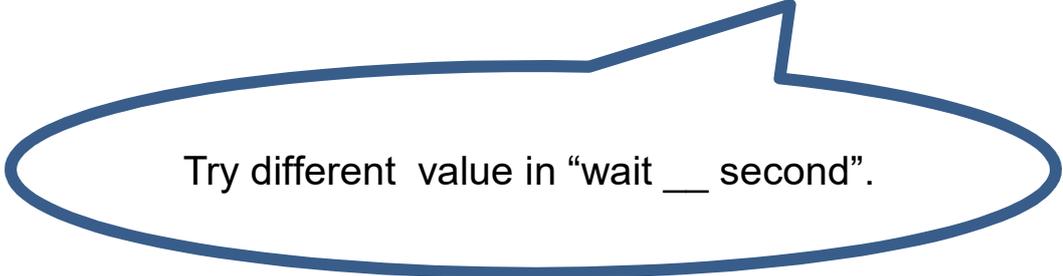
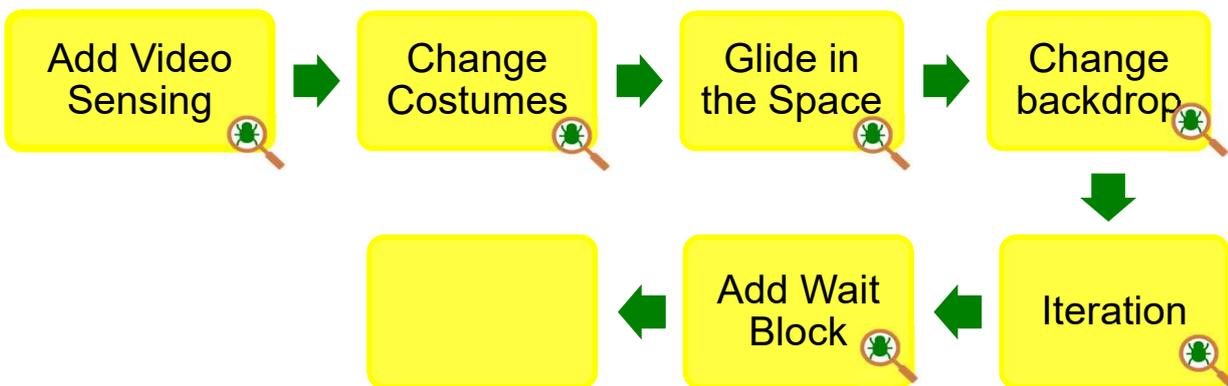


The image shows a Scratch code editor. At the top, there is a 'when clicked' event block. Below it is a 'forever' loop block. Inside the loop, there is an 'if' block with the condition 'video motion on stage > 350'. If true, it executes a 'next backdrop' block. To the right, the 'Motion' category is expanded, and a 'wait 1 second' block is circled in red. A blue arrow points from this block to a question mark in the code area, indicating where it should be inserted.



### Testing and Debugging

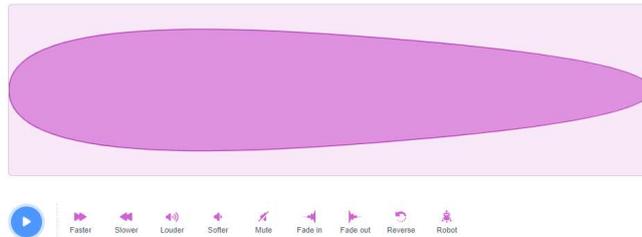
Can't wait to see the changes? Let's test it now!



## To Code: Add Sound Effect (Sprite)

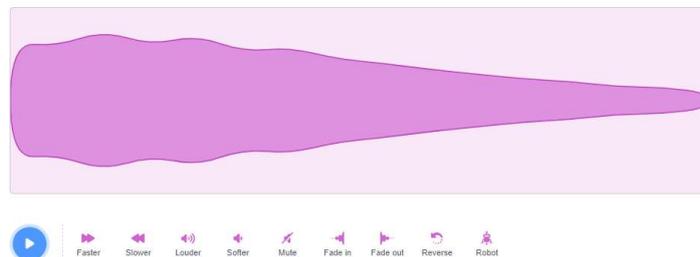
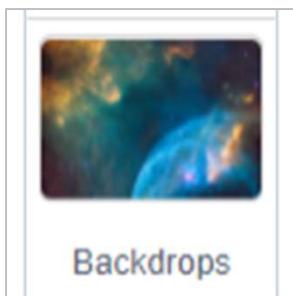
Remember how to add the sound? Try to make the **sprite** play a sound based on the video motion.

See Appendix  
P.38-39



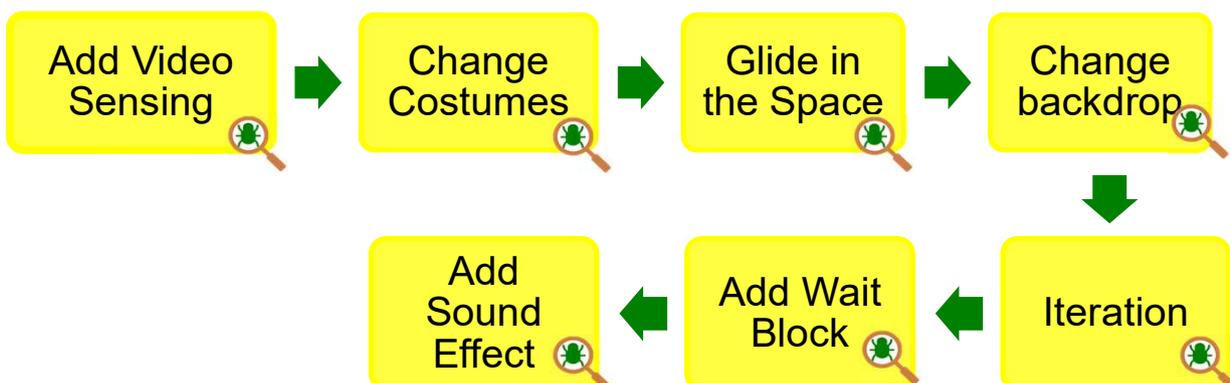
## To Code: Add Sound Effect (Backdrop)

Repeat the previous step to make the **backdrop** play a sound based on the video motion.



### Testing and Debugging

You can test anytime! Wave your hand and see if you hear the sound effect you just added.



# Space Travelling

## To Reflect

Which one has the same effect with this event?

```
when clicked
  forever
    if video motion on stage > 350 then
      next backdrop
      start sound Teleport2
      wait 3 seconds
```

Video Sensing

- when video motion > 10
- video motion on stage
- turn video on
- set video transparency to 50

In this lesson, you will create your own story by using Video Sensing feature! What story would you like to create?

## To Create: Make your own project

Task:

Go to “Create” to start a new project.

Let's start to create your own project with different **Theme**, **Sprite** and **Backdrop**.

### Theme

Jungle

Sports

Sea

Space

### Sprite



Dinosaur1



Dinosaur2



Dove



Dragon



Basketball



Batter

### Backdrop



Arctic



Baseball 1



Bedroom 2



Bedroom 3



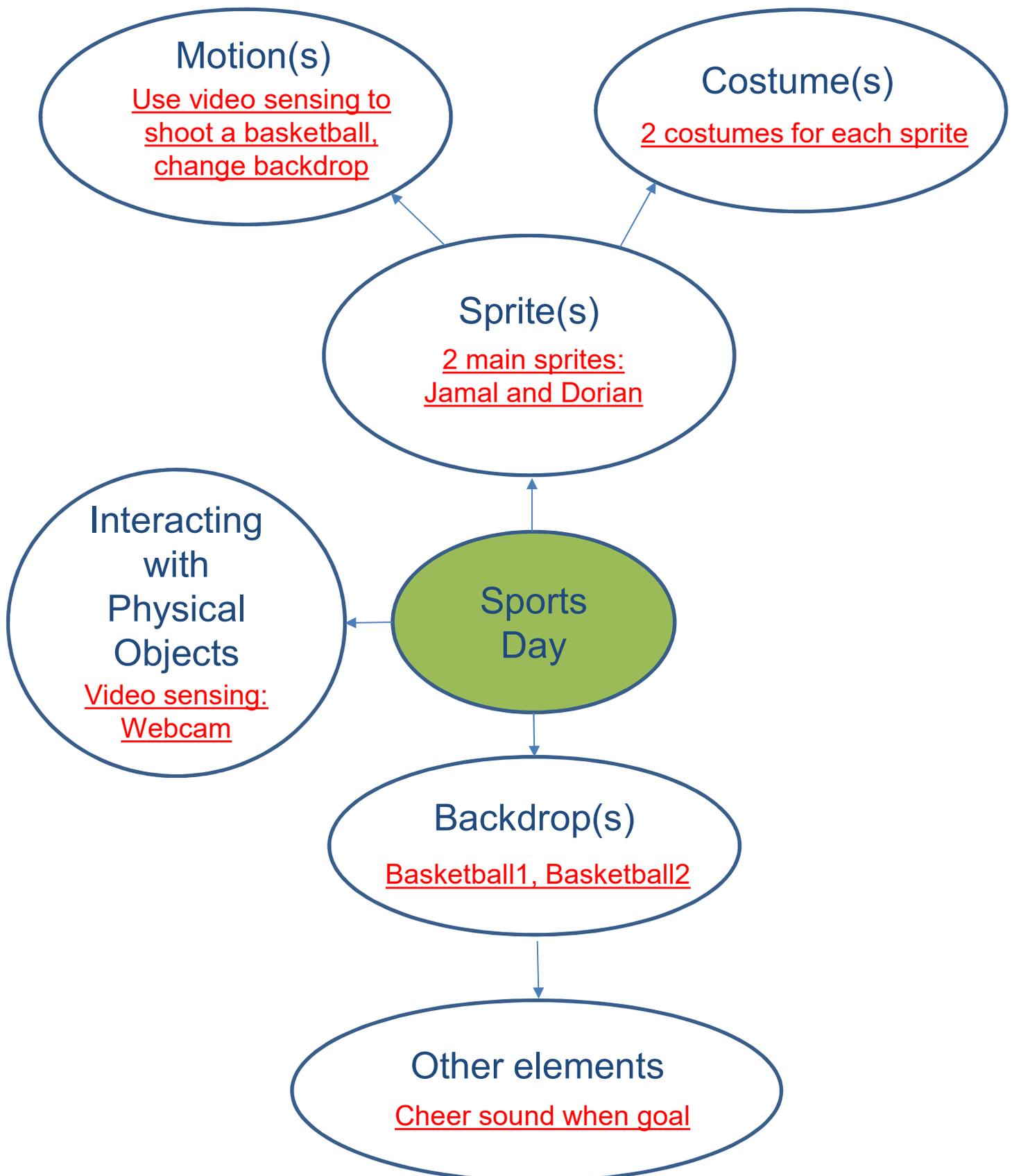
Castle 2



Castle 3

# Space Travelling

Complete the mind map below before your start to code!



## To Reflect: Two Stars and a Wish Worksheet

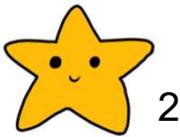
Name of Project: \_\_\_\_\_ Name of Creator: \_\_\_\_\_

Please write down two things that you like about this project.



“Two Stars and a Wish” is a reflection strategy designed for student feedback as peer- and self-assessment.

Teachers can guide students to give constructive feedback to their peers regarding their Scratch project - two positive (stars) and one hopeful (wish) reflection. Comments can be made on Scratch project’s idea, features and aesthetic aspects etc.



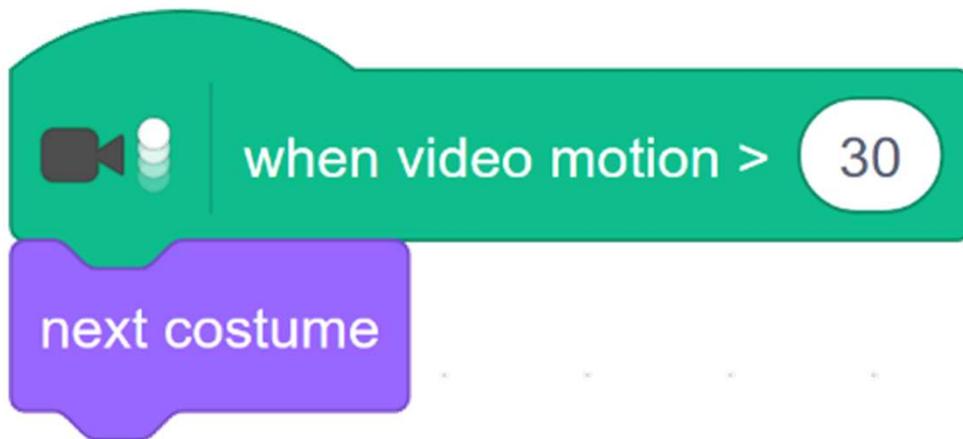
What is one thing you would like to add or change to make this project better?



# Space Travelling

## Review Questions

1. The following blocks are **NOT** an example of:

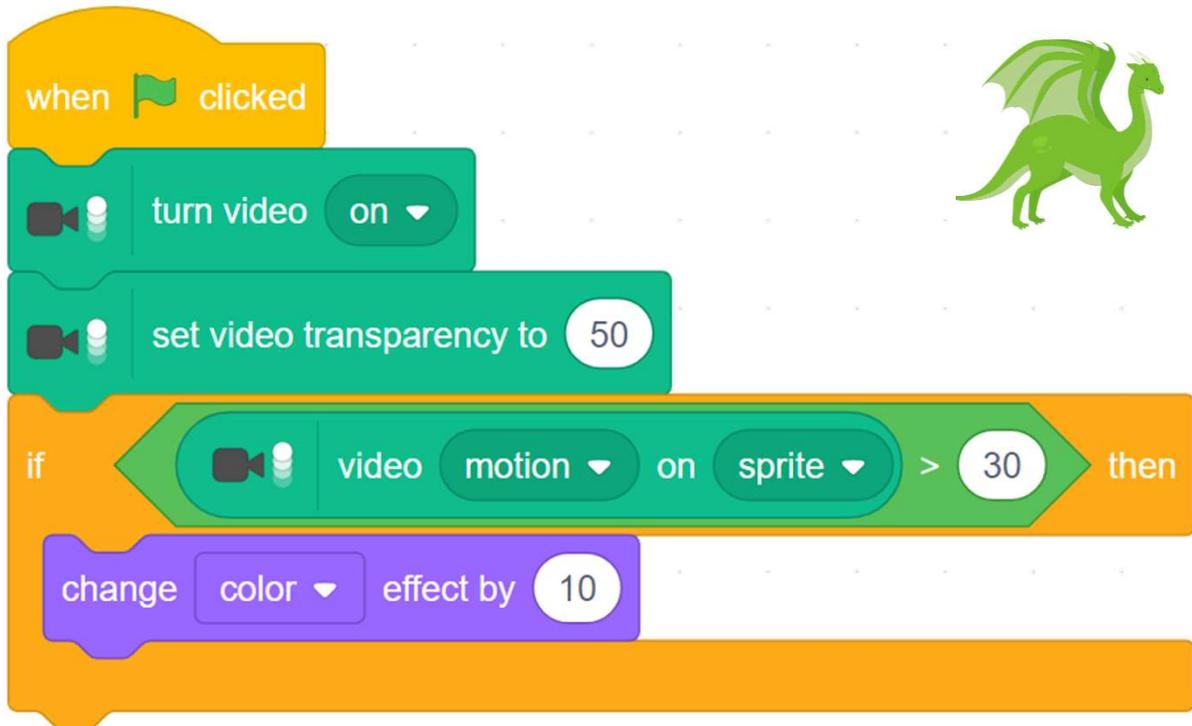


- A. Events
- B. Naming
- C. Conditionals
- D. Operators

**(Answer: B)**

## Review Questions

2. A student writes the following code and tests it by moving his hand in front of the webcam on his computer. He expects the dragon to change colour, but it only changes once and stops. What would you tell him to help him fix his problem?

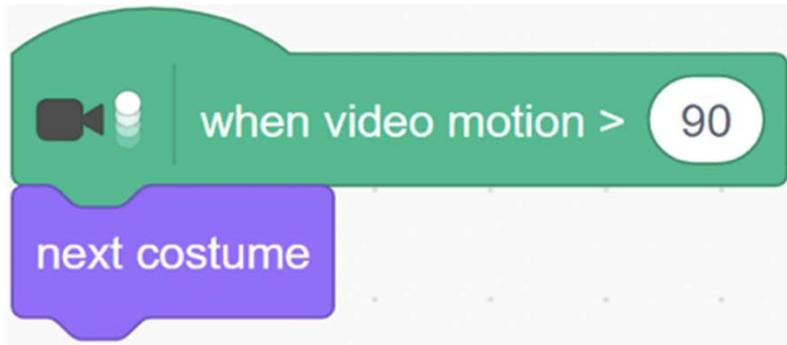
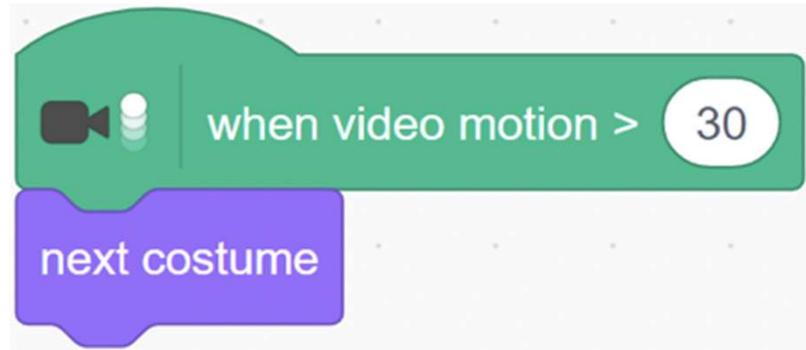


- A. Change 30 to a higher number in the “if-then” block.
- B. Change 10 in the “change colour effect by” block to a higher number.
- C. Put the “if-then” block inside a “forever” block.
- D. Use a “next costume” block instead of “change colour effect by” block.

**(Answer: C)**

## Review Questions

3. When the video sensing is on, if a user waves his hand slowly, Clothes (Dress) and Face (Harper), which one will be easier to change?

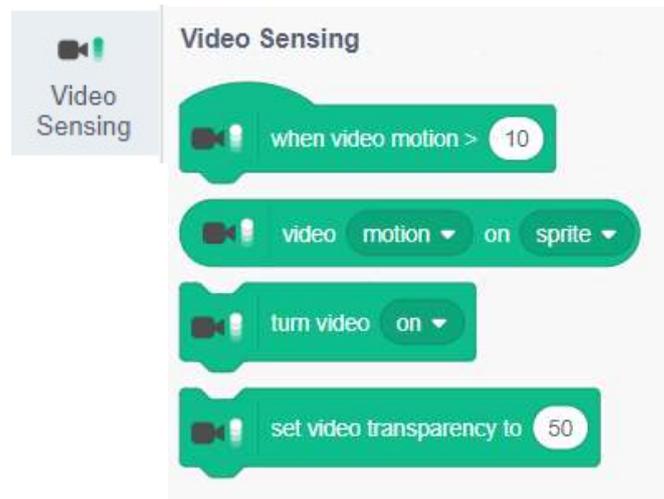
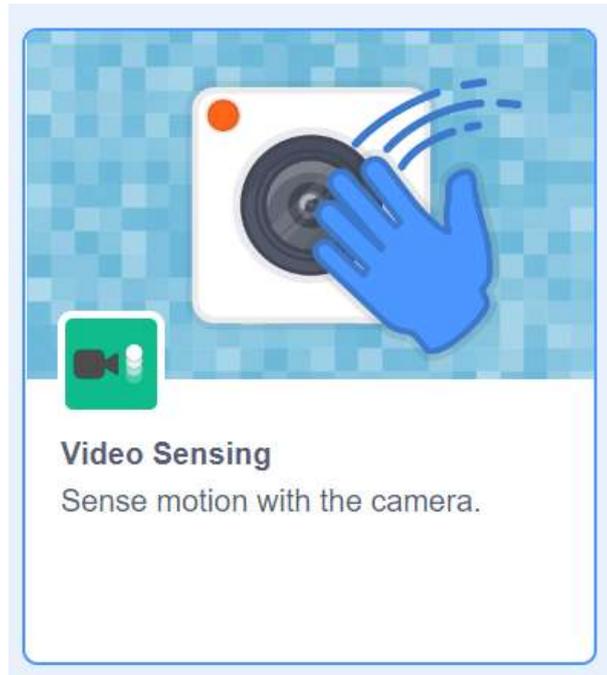


- A. Dress.
- B. Harper.
- C. None will change.
- D. No difference between them.

**(Answer: A)**

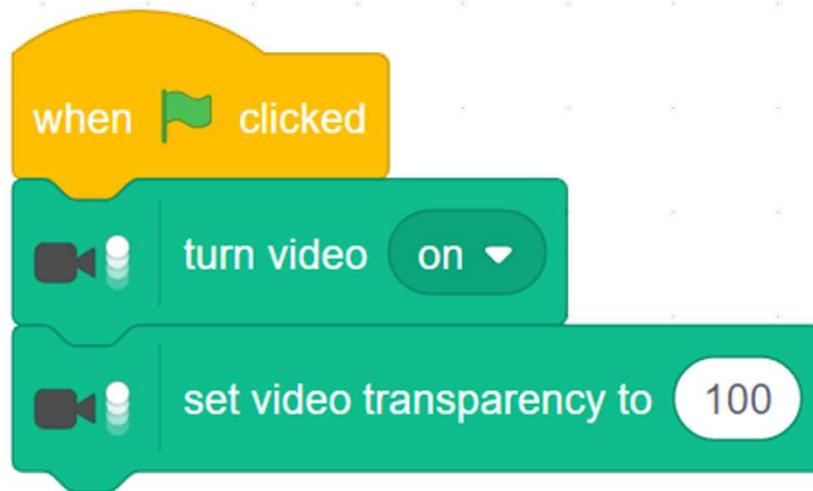
## Revision on Key Features

### Video Sensing:

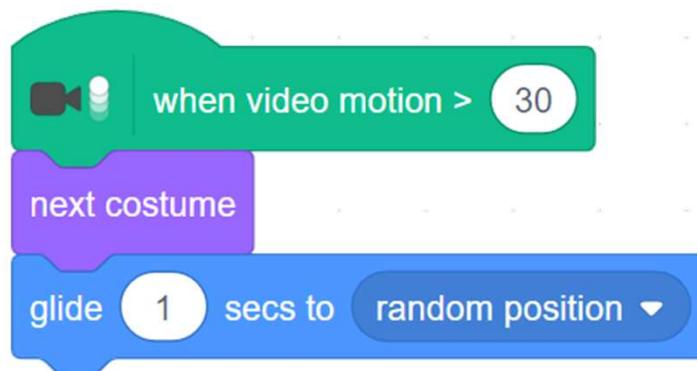


## Revision on Key Concepts & Practices

**Sequences:** It is the order in which the programming statements are executed. A wrong order would lead to incorrect programming results.

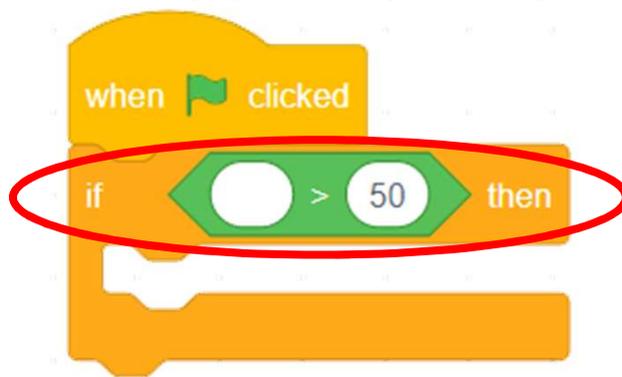


**Events:** We use event blocks to trigger Scratch to take actions.

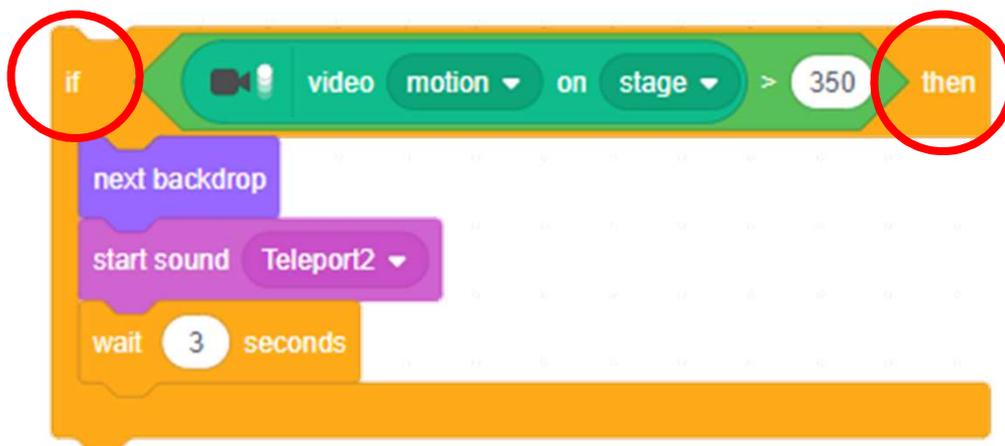


## Revision on Key Concepts & Practices

**Operators:** We use conditional operators to evaluate whether a condition is true or false. Conditional expressions always use operators such as greater than ( $>$ ), less than ( $<$ ) or equal ( $=$ ).

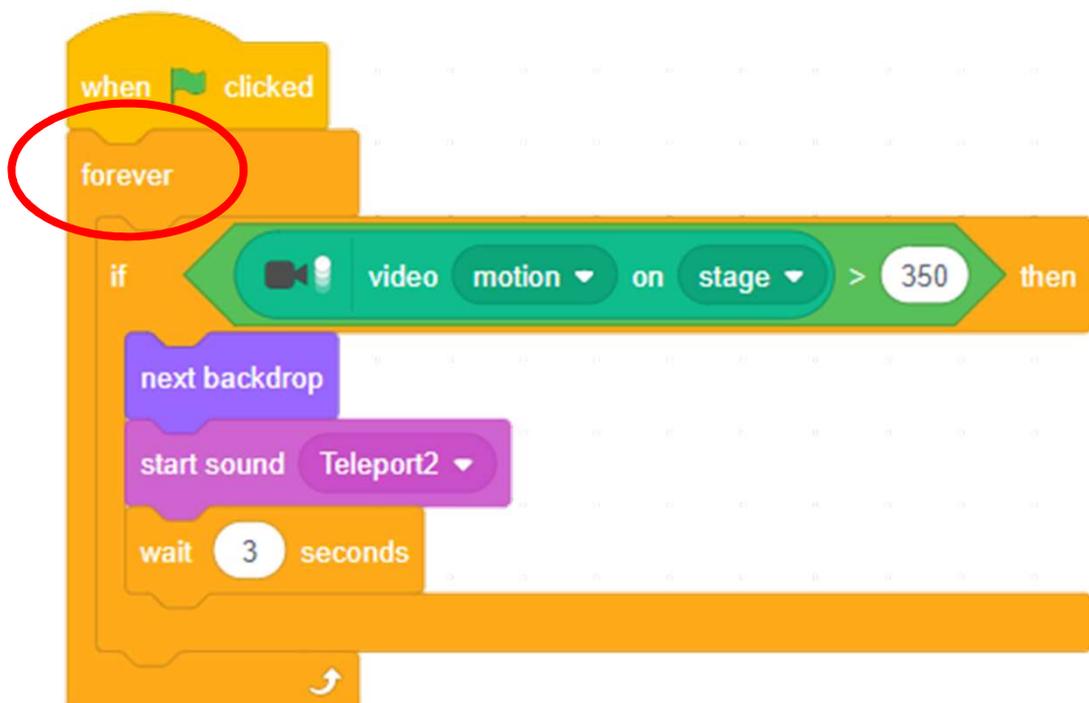


**Branching/Selection:** We use conditional statements in programming to enable computers to make decisions. Conditionals always have an “if” part, which tells the program in the “then” part what to do when the condition is true.



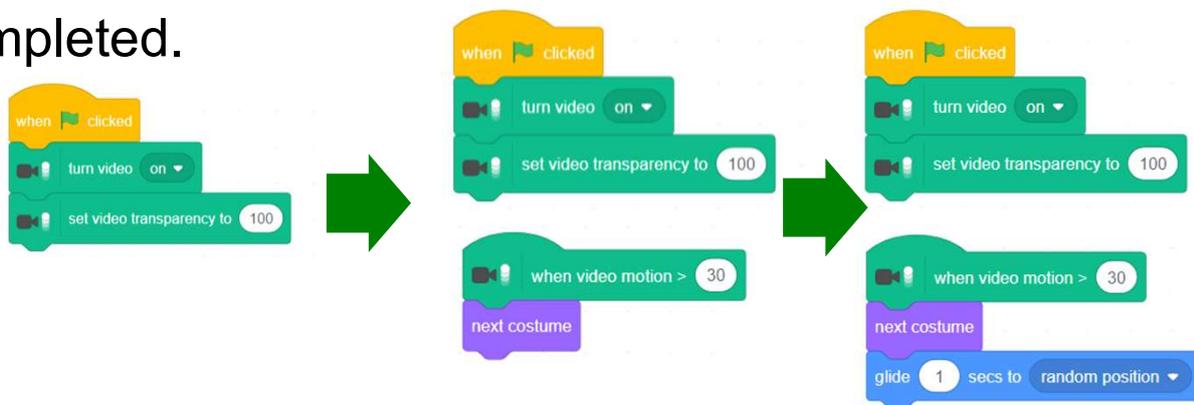
## Revision on Key Concepts & Practices

**Iteration - Forever:** Iteration is repeating a process in order to produce a sequence of outcomes. Forever and repeat blocks can trigger iteration in Scratch.

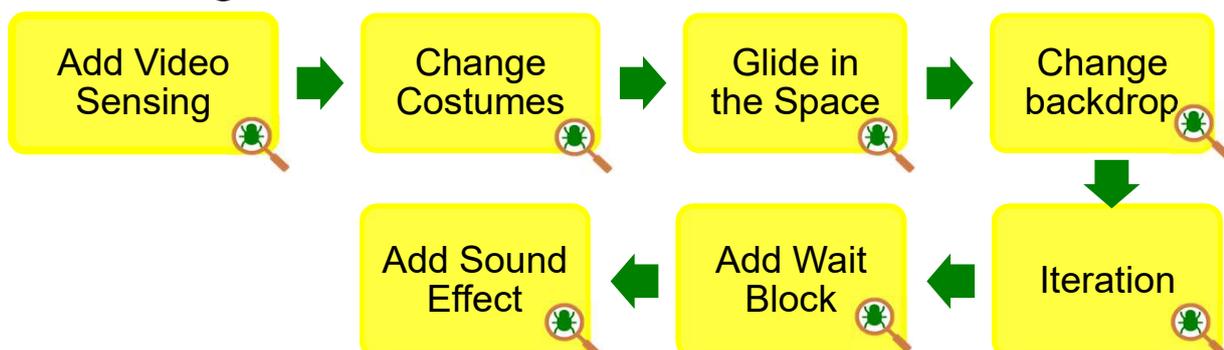


## Revision on Key Concepts & Practices

**Being incremental and iterative:** to work out a sub-task as an iteration, try it out, then work out another sub-task based on the codes of the previous sub-task in one more iteration until the whole programming task is completed.



**Testing and debugging:** Testing a computer program is the process of checking if it can produce results as designed. Debugging a computer program is the process of finding out ways to revise the program so that the bugs can be removed.



# Appendix

Operation Manual

# Space Travelling

## To Code: Adding the Backdrop and Sprite

1. Turn on the webcam of your computer.



See Teacher Guide P.4

2. Sign into your account at scratch.mit.edu.



3. Go to “**Create**” to start a new project.



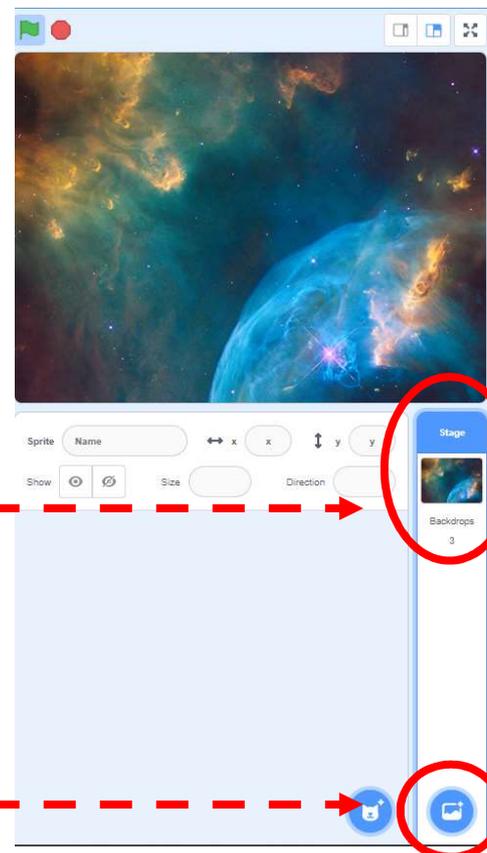
4. Name it “**Space Travelling**”.



5. Choose a good backdrop for the space.



Click on the Stage



6

Click on the “Choose a Backdrop” icon

## To Code: Adding the Backdrop and Sprite

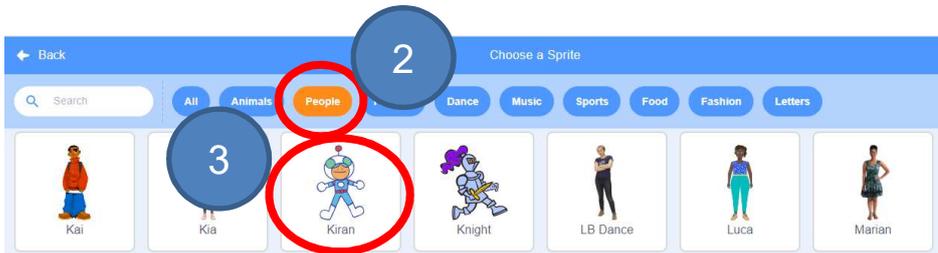
### Choosing Sprites and Resizing

Delete the original Scratch cat sprite by clicking the “X” in the upper right corner of its image.

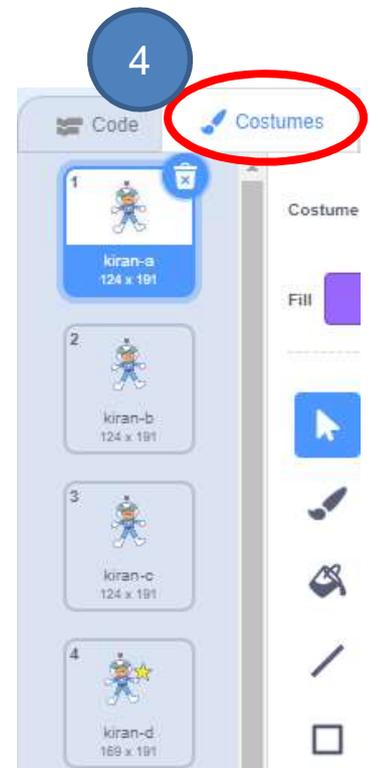
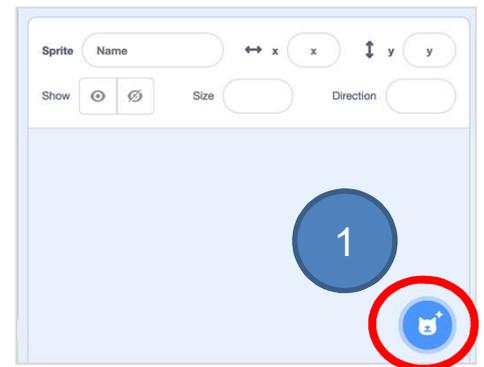
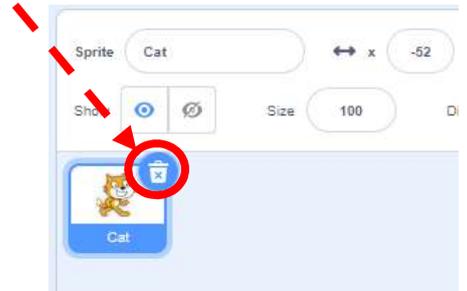
See Teacher Guide P.4

Let's choose the new sprites for your project.

1. Click on the “**Choose a Sprite**” icon.
2. Click on the “**People**” category on the top to see a large set of sprites.
3. Add “**Kiran**” sprite to the project.



4. Now, you can see a set of costumes for Kiran.
5. You can also resize and rename “**Kiran**” sprite.



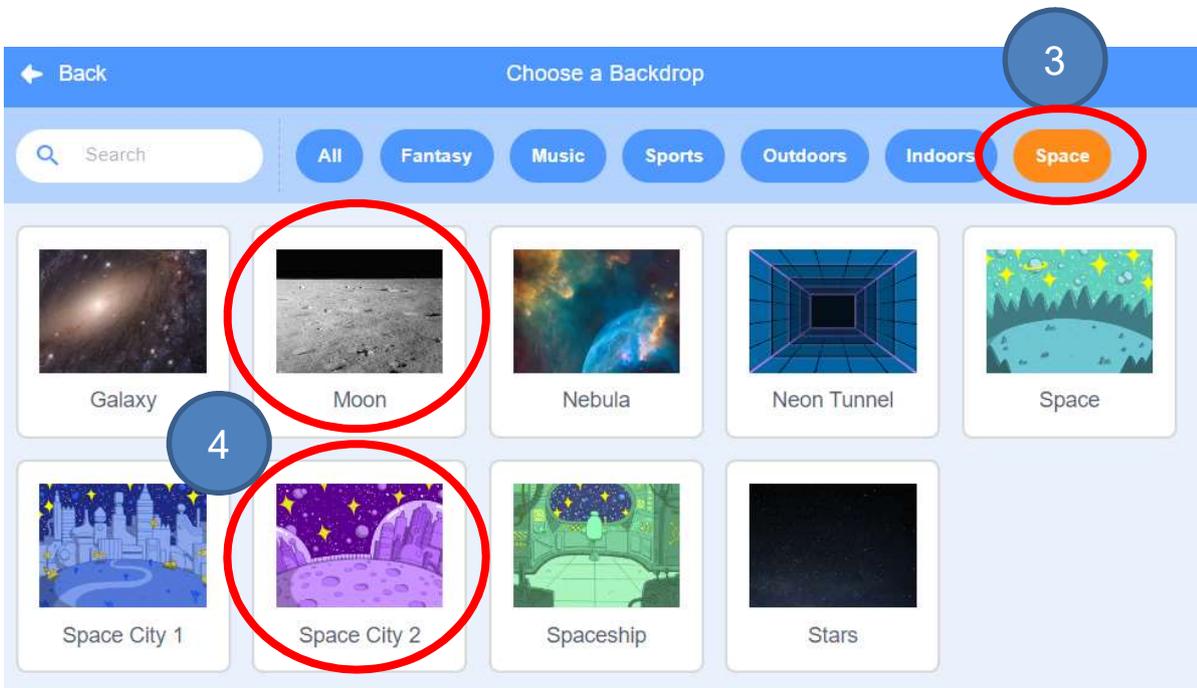
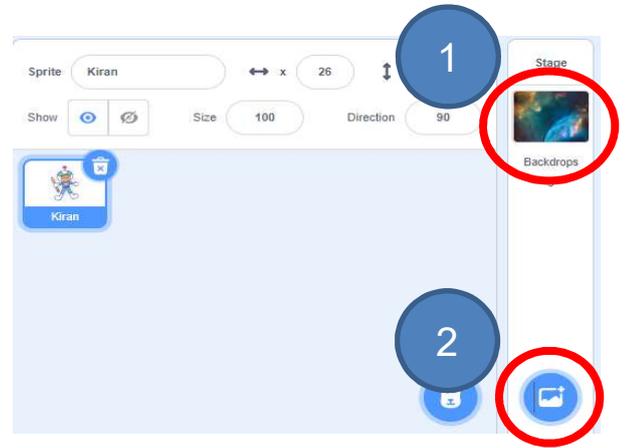
# Space Travelling

## To Code: Change Backdrop

See Teacher  
Guide P.15

Try to add a total of 3 backdrops for your space travelling!

1. Click the “**Backdrop**” in **Stage**.
2. Click on the “**Choose a Backdrop**” icon.
3. Click “**Space**” category.
4. Choose two more backdrops.



# Space Travelling

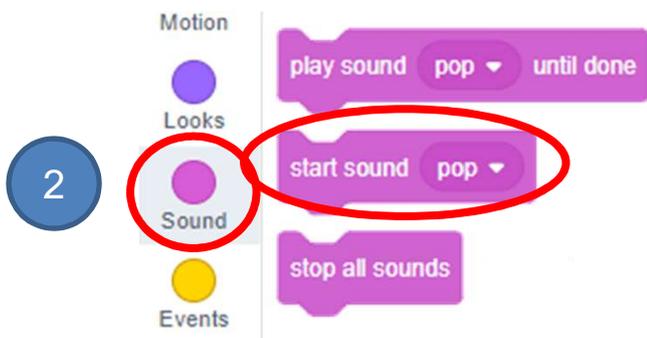
## To Code: Add Sound Effect (Sprite)

See Teacher  
Guide P.21

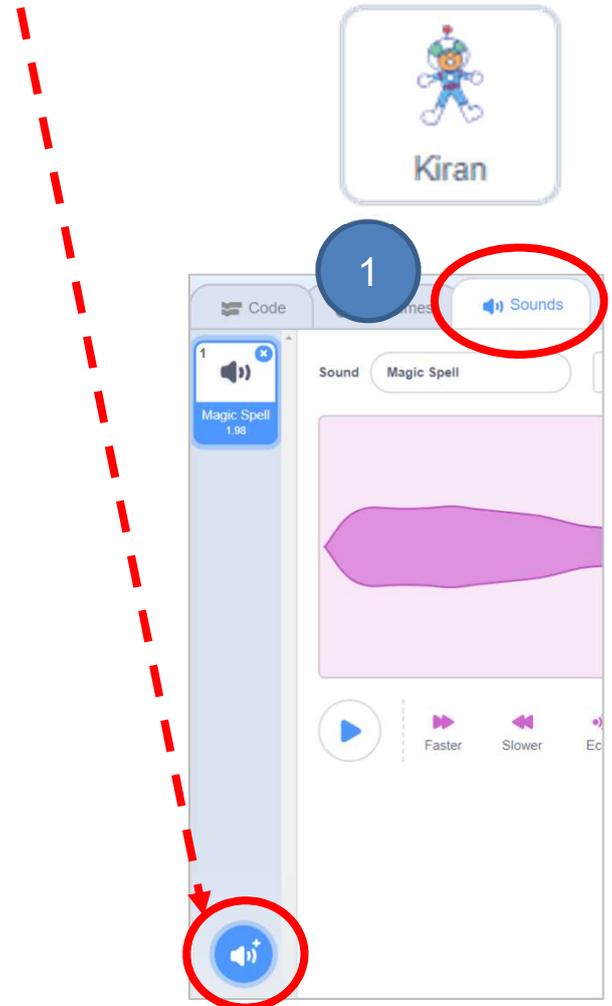
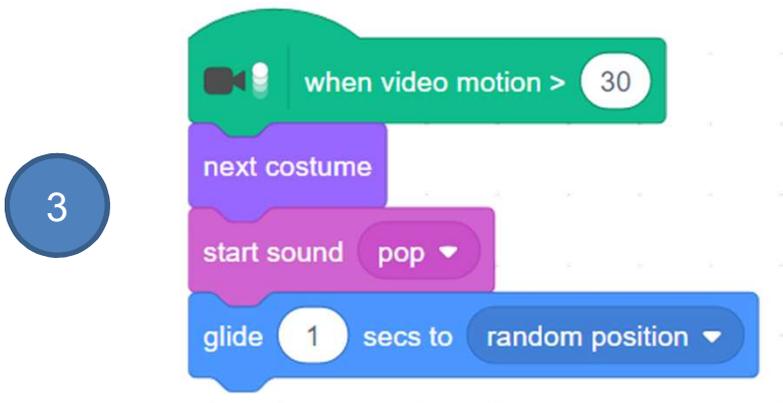
Make the sprite play a sound based on the video motion.

1. Go to the “**Sounds**” tab. Click on the “**Choose a Sound**” icon at the bottom left to select music from the library.

2. Drag out the “**start sound**” block from the “**Sound**” drawer.



3. Snap the “**start sound**” block after “**next costume**” block.

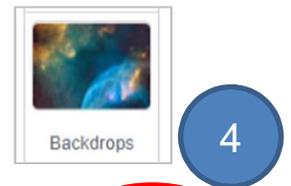


## To Code: Add Sound Effect (Backdrop)

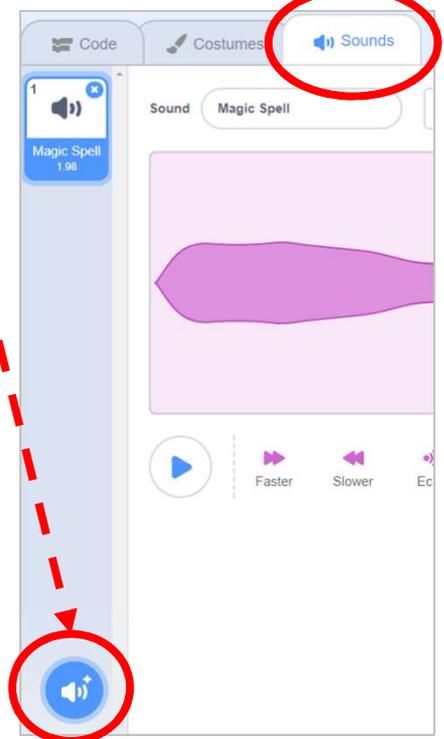
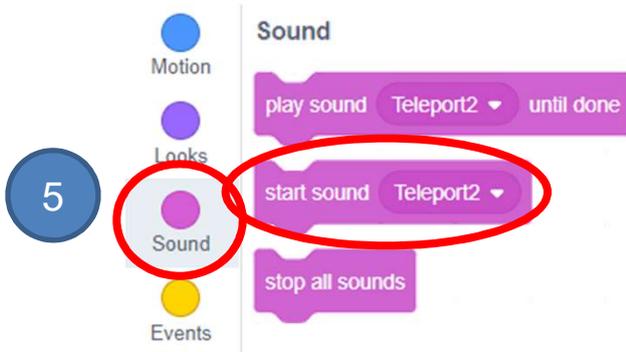
Repeat the previous step to make the backdrop play a sound based on the video motion.

See Teacher Guide P.21

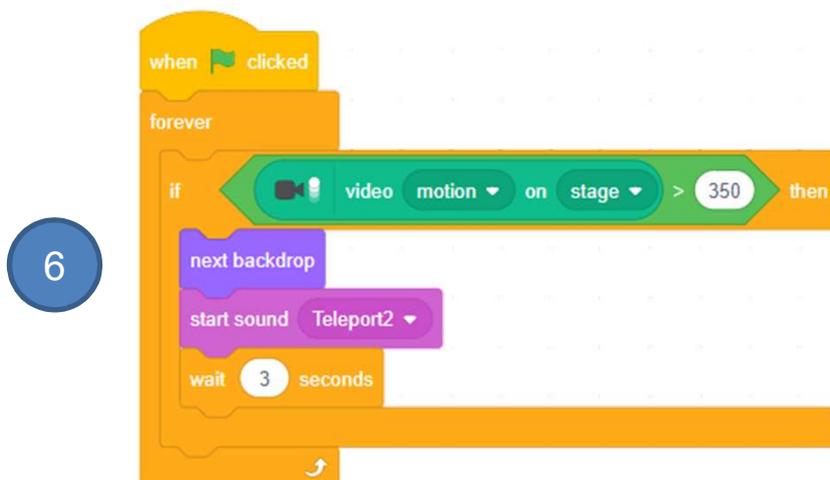
- Go to the **"Sounds"** tab. Click on the **"Choose a Sound"** icon at the bottom left to select music from the library.



- Drag out the **"start sound"** block from the **"Sound"** drawer.



- Snap the **"start sound"** block inside the **"if-then"** block.



## Program Codes



Sprite (Kiran)

```
when clicked
  turn video on
  set video transparency to 100
  when video motion > 30
    next costume
    glide 1 secs to random position
    start sound pop
```

```
when clicked
  forever
    if video motion on stage > 350 then
      next backdrop
      start sound Teleport2
      wait 3 seconds
```



Backdrop

