

{oo/Think @JC >

賽馬會運算思維教育

Inspiring digital creativity 啟發數碼創意

COMPUTATIONAL THINKING EDUCATION



LEVEL

1

Created and Funded by:



The Hong Kong Jockey Club Charities Trust

Co-created by:



香港教育大學
The Education University
of Hong Kong



Massachusetts
Institute of
Technology



香港城市大學
City University of Hong Kong

CoolThink@JC - Computational Thinking Education Programme

Seeking to inspire digital creativity among students and nurture their proactive use of technologies for social good from a young age, CoolThink@JC is a computational thinking education initiative created and funded by The Hong Kong Jockey Club Charities Trust, co-created by The Education University of Hong Kong, Massachusetts Institute of Technology and City University of Hong Kong, and supported by the Education Bureau of Hong Kong. In collaboration with local educators and the world's leading experts, CoolThink@JC empowers teachers with high-quality teaching materials, learning platform and professional development programmes.

CoolThink@JC aims to mainstream computational thinking education in Hong Kong, to prepare students for a fast-changing digital future through a hands-on, minds-on, joyful learning experience and be the creator of technology. Since its launch in 2016, the project has developed a three-year curriculum for upper primary school pupils with 14 class hours per school year. It is on track to support 200 primary schools and more than 600 teachers, and enable over 100,000 upper primary school

students to benefit from the computational thinking education classes. The project has supported more than 368 primary schools in different ways. At the same time, through cooperation with different school-sponsoring bodies, teachers' associations and the Education Bureau, it encourages more than 70% of the public and Direct Subsidy Scheme schools in Hong Kong to jointly promote computational thinking education. An independent evaluation has found that students participated in CoolThink@JC grew twice as much in problem-solving skills when compared with non-participating students.

CoolThink@JC received three international accolades, showing that its overall design, curriculum and pedagogical approach have reached international standards, and have far-reaching impact to Hong Kong's technology education ecosystem. The three accolades include: QS Reimagine Education Awards, International Society for Technology in Education (ISTE) international certification (Seal of Alignment - Student Standard), Education Alliance Finland certification.



Message from The Hong Kong Jockey Club Charities Trust

The Hong Kong Jockey Club is dedicated to acting continuously for the betterment of our society. This is made possible by the Club's unique integrated business model, through which revenue is returned to the community in the form of tax payments and charitable donations. As one of the top ten charity donors in the world, the Club's Charities Trust supports various charitable projects to address different societal needs. With "Children & Youth Development" as one of its focus areas, the Club helps young people give full play to their strengths by developing their knowledge and capabilities. We also want to cultivate positive, compassionate, caring young individuals, who are committed, like the Club, to the betterment of our community.

As digital technology and computational thinking skills are major drivers of today's social and economic development, inspiring the digital creativity of students is among our top priorities. Yet, CoolThink@JC is more than an initiative that "helps everyone code". We believe that empowering students to move beyond technology consumption to problem-solving, creation and innovation requires them to not only acquire coding concepts and skills, but also to explore new perspectives. The aim is to gear them up to address the challenges around them with the application of digital technology as well as advance future opportunities and social good.

The CoolThink@JC project seeks to sustainably develop Hong Kong's computational education ecosystem. It is expected that by the end of 2024, the programme will have nurtured nearly 100,000 local upper primary school students. The project has developed a quality-assured, internationally-recognised curriculum, backed by evidence-based independent evaluation and research. We are delighted to make this quality teaching material accessible to all students citywide. By encouraging both teachers and students to make the best use of this resource, we aim to instill the project's concept of "to play, code, think and reflect" in daily lives.

The Club would like to extend its sincere gratitude to the Hong Kong SAR Government and various sectors of our society for their support of CoolThink@JC. Our heartfelt thanks especially go to its co-creators, The Education University of Hong Kong, Massachusetts Institute of Technology and City University of Hong Kong. We are also grateful to all supporting partners, principals, teachers and parents for their commitment to nurturing our younger generation. We look forward to your continued support of this meaningful project so it can benefit as many schools and students as possible.

Preface from Massachusetts Institute of Technology

More and more, our lives are being shaped by Information Technology. To understand the basic ideas driving this change has become a fundamental skill for everyone, even young children. The sample lessons in this book are an introduction to Computational Thinking for upper level primary school students.

Designed by teams at Massachusetts Institute of Technology (MIT) and The Education University of Hong Kong, the lessons are now being used and tested in 200 primary schools as part of CoolThink@JC, an initiative of The Hong Kong Jockey Club Charities Trust. The goal of CoolThink@JC is not only to teach kids to program, but more importantly to help them be creators in a world transformed by computation, and to empower them to take action to realize their ideas for using computation to improve life for their family, friends and community.

The lessons presented here introduce Computational Thinking in three aspects: concepts, practices and perspectives. Concepts are the pieces from which computer operations are constructed. They are the programming elements that provide sequencing, repetition, responding to events, and other operations. Practices are the habits of thought that make people effective in creating computer artifacts. They are principles like abstraction, testing and incremental design. Perspectives are attitudes by which people come to see themselves as effective participants in an information society, attitudes such as appreciating which kinds of issues can be addressed using computing, or feeling that you are able to express ideas through computing.

Mastering Computational Thinking includes gaining experience with programming. The CoolThink@JC material uses two programming environments: Scratch and MIT App Inventor 2. Scratch was developed at MIT and App Inventor was initially created at Google research and later continued at MIT. Both environments are designed to make it easy for beginners to create programs. Rather than typing programming code character by character, you build programs by working with graphical blocks that you move around on a display screen and piece together like pieces of a jigsaw puzzle. Scratch emphasizes creating interactive graphics and animations that run in the computer browser. App Inventor programs produce mobile applications that you can install on smartphones or tablets and share with others. These applications let people interact with the world around them and draw upon all the power the Internet has to offer.

We're living in a time of wonders, where Information Technology can bring amazing powerful resources within the grasp of even young children. The lessons in this book are mere hints of what's possible and the barest beginning of a great adventure. Please join us on this adventure.

Hal Abelson

Professor of Computer Science and Engineering
MIT

Preface from The Education University of Hong Kong

The Education University of Hong Kong (EdUHK) dedicates to the advancement of professional development for teachers through research. We also focus on leading curriculum development in Hong Kong through new pedagogical design. The Education University of Hong Kong was invited by The Hong Kong Jockey Club Charities Trust in 2016 to launch the CoolThink@JC programme with The Massachusetts Institute of Technology. The two universities jointly develop a set of computational thinking curriculum units suitable for grade four to grade six primary students in Hong Kong. The curriculum aims to encourage schools to provide learning opportunities for upper level primary school students, and to nurture their computational thinking via the programming activities in Scratch and MIT App Inventor environment. The objectives of CoolThink lessons are to boost students' interest in programming, equip students with essential capability to code, and enable students to cultivate their problem-solving skills and digital creativity. It also helps students to succeed in the digital era by laying the foundation for their creative thinking and pioneering spirit.

CoolThink lessons and the associated learning activities are designed and developed base on the student-centered learning approach. It enables students to master Computational Thinking Concepts through direct engagement in programming activities. More importantly, through the process of programming, we introduce important Computational Thinking Practices, broaden students' Computational Thinking Perspectives needed in the digital age, and cultivate students' understanding of how developing games and applications can bring positive impacts to the society.

EdUHK emphasizes the importance of professional development for teachers in computational thinking education. Teacher training includes pedagogy for computational thinking development of students and the follow-up reflection on teaching computational thinking upon classroom practices. The professional development for teacher is a vital part of this programme. Only when teachers have a good understanding of the pedagogy for computational thinking development, and through reflection after teaching practice can they integrate the Concepts, Practices and Perspectives into their classroom teaching and make progress. The EdUHK team modifies the computational thinking pedagogy overtime along with the development of CoolThink@JC. From using the "To play - To think - To code - To reflect" pedagogy initially to promote thinking education, the team further introduces the "seven-step" programming pedagogy to guide teachers to conduct productive computational thinking lessons for inspiring digital creativity. Soon after the commencement of the second phase of CoolThink@JC, students in Hong Kong experienced a long period of online lesson due to suspension of face-to-face teaching under the pandemic. The project team reacted quickly by making a series of animations, which are developed based on CoolThink Scratch and MIT App Inventor curriculum units. This gives raise to the pedagogy of using animations for self-directed learning, as a further enhancement of pedagogical practices for computational thinking development in response to the limited teaching time.

Professor Siu-Cheung Kong

*Professor, Department of Mathematics and Information Technology,
Director, Centre for Learning, Teaching and Technology,*

The Education University of Hong Kong

Table of Contents

CoolThink@JC Framework	8
Unit 1 - Creative Computing with Scratch	9
1. Student Guide Lesson 1	9
2. Student Guide Lesson 2	13
3. Two Stars and a Wish Worksheet	18
Unit 2 - Dancing Cat	19
1. Student Guide Lesson 1	19
2. Student Guide Lesson 2	25
3. Two Stars and a Wish Worksheet	29
Unit 3 - Make a Maze Game with Scratch	30
1. Student Guide Lesson 1	30
2. Unplugged Activity: Variables	34
3. Exploring Forever and If-Then Worksheet	40
4. Questions and Answers	44
5. Score Cards	46
6. Student Guide Lesson 2	48
Unit 4 - Tell a Joke with Scratch	55
1. Student Guide Lesson 1	55
2. Unplugged Activity: Broadcast Skit	60
3. Student Guide Lesson 2	62
Unit 5 - Tell a Story with Scratch	72
1. Three Little Pigs Storyboarding Worksheet	72
2. Student Guide Lesson 2	74
3. Two Stars and a Wish Worksheet	84
Unit 6 - Make Magic	85
1. Student Guide Lesson 1	85
2. Student Guide Lesson 2	95
3. Unplugged Activity: Conditionals	106
4. Unplugged Activity: Guessing Game	107
5. Two Stars and a Wish Worksheet	109
Unit 7-8 - Computational Arts with Scratch	110
1. Student Guide Lesson 1	110
2. Student Guide Lesson 2	118
3. Student Guide Lesson 3	125
4. Student Guide Lesson 4	132
Scratch Final Project	140
1. Scratch Final Project Student Guide	140
2. Scratch Final Project Design Worksheet	146
3. Scratch Final Project Peer Feedback Worksheet	148
4. Two Stars and a Wish Worksheet	153

CoolThink@JC Framework

Computational thinking entails developing fluency with programming **concepts** (key constructs and ideas that are central to most forms of computing). But more fundamental than specific coding details, computational thinking includes the ability to pose problems and seek solutions that use key **practices** (the activities people engage in when creating computational projects) involved in computing and programming. And beyond solving problems with computers, young people should develop **perspectives** like digital empowerment, which enables them to see how challenges in the world around them could be addressed through computing; and computational identity, which instills in them the knowledge that they can have active, positive roles as people who can use computers to enhance their lives and the lives of people around them.

01

CT Concepts



CT Concepts

- < Sequences / > : identify a series of ordered steps for solving a programming task
- < Events / > : one thing causing another thing to happen
- < Conditionals / > : make decisions based on conditions
- < Operators / > : support for mathematical and logical expressions
- < Parallelism / > : make things happen at the same time
- < Repetition / > : run the same sequence multiple times
- < Naming and Variables / > : store information to be referenced and computed in a program
- < Data structures / > : basic ways data are stored, retrieved, and updated
- < Procedures / > : create code blocks to modularize and abstract sequences of commands

02

CT Practices



CT Practices: Problem-solving Skills

- < Testing and debugging / > : make sure things work, otherwise find and solve problems when they arise
- < Being incremental and iterative / > : develop a little bit, then try it out, then develop more
- < Reusing and remixing / > : make something by building on existing code, projects or ideas
- < Abstracting and modularizing / > : explore connections between the whole and the parts
- < Algorithmic thinking / > : articulate a problem's solution in well-defined rules and steps

03

CT Perspectives



CT Perspectives: Identity and Motivation

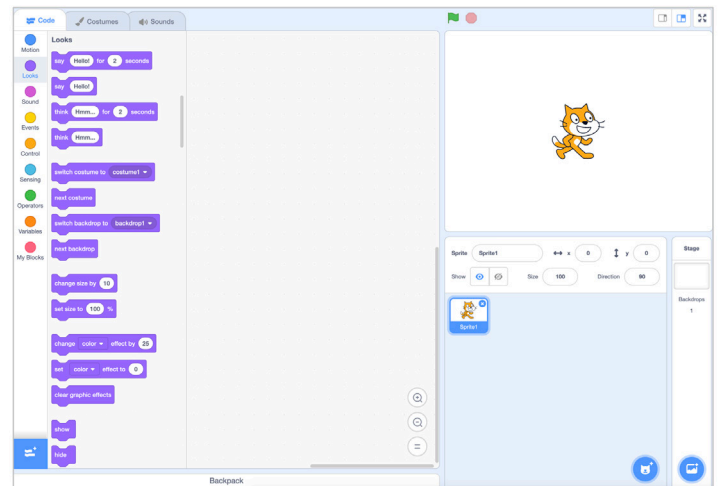
- < Expressing / > : create and express ideas through this new medium
- < Questioning / > : feel empowered to ask questions about and with technology
- < Connecting / > : appreciate that others are engaged with and appreciate one's creations
- < Digital empowerment / > : develop the ability to see problems in the world as solvable through code
- < Computational identity / > : see oneself as being able to enhance the world through coding

Lesson 1

LET'S LEARN SCRATCH!

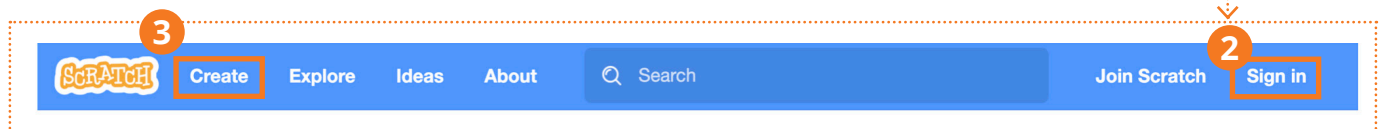


In this activity, you will learn to sign in and out of the Scratch website and learn how to make the cat move and play music.

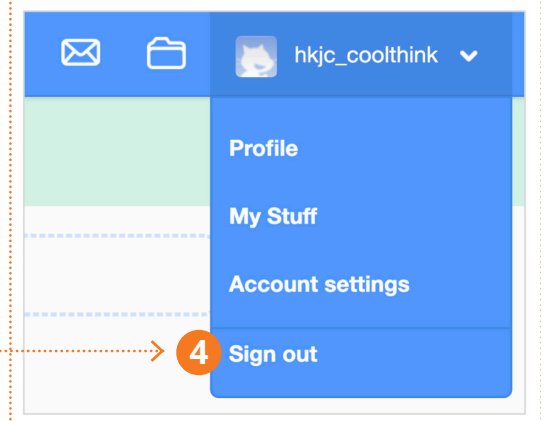


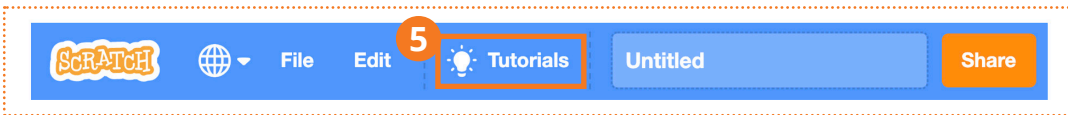
START HERE

- ▶ 1 Go to the Scratch website: <https://scratch.mit.edu>
- ▶ 2 Sign into your account.

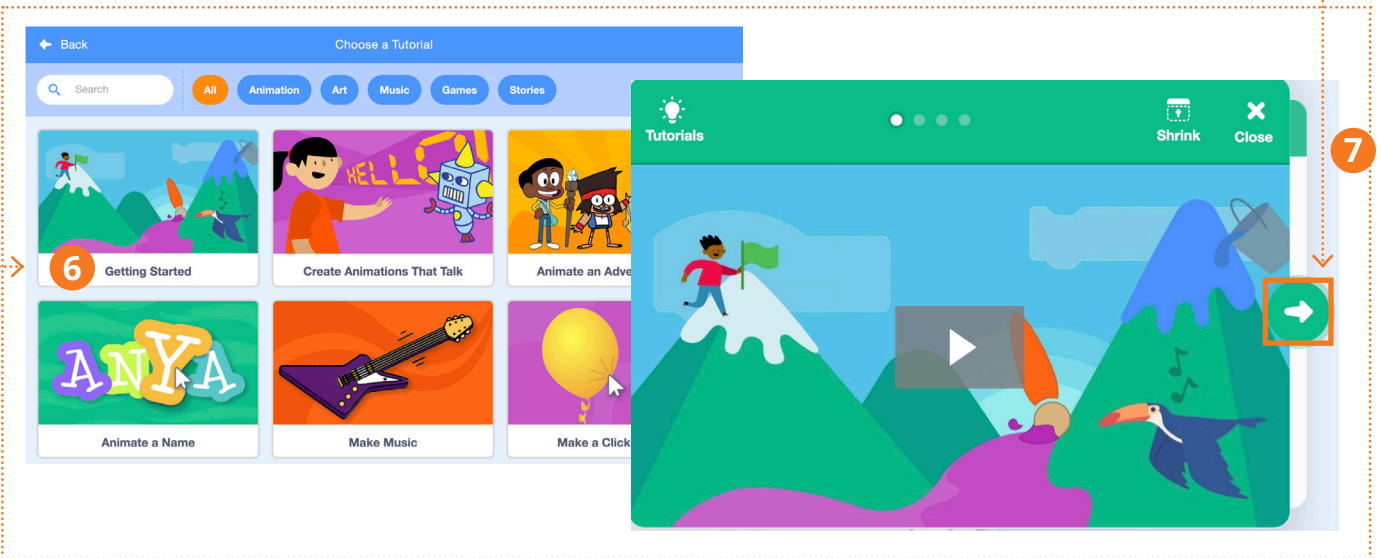


- ▶ 3 Click on the **Create** tab located at the top left of the browser to start a new project.
- ▶ 4 Sign out by clicking on your name and clicking **Sign out**. Then sign in again!

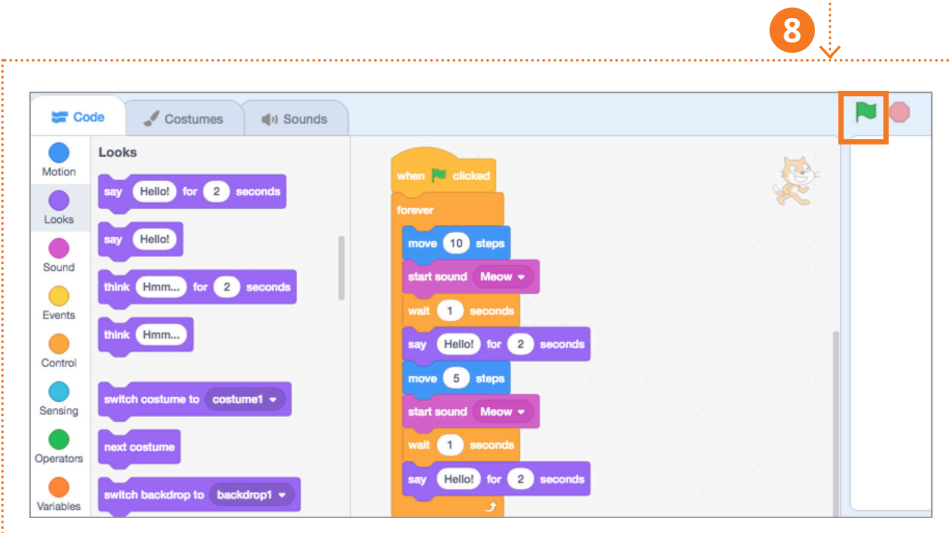




▶ **5 6 7** Time to explore! Click on **Tutorials** on the menu bar and then choose **Getting Started** on the “Tutorials” page. After the video, you can click the **green arrow** on the right side of the window to step through the tutorial.



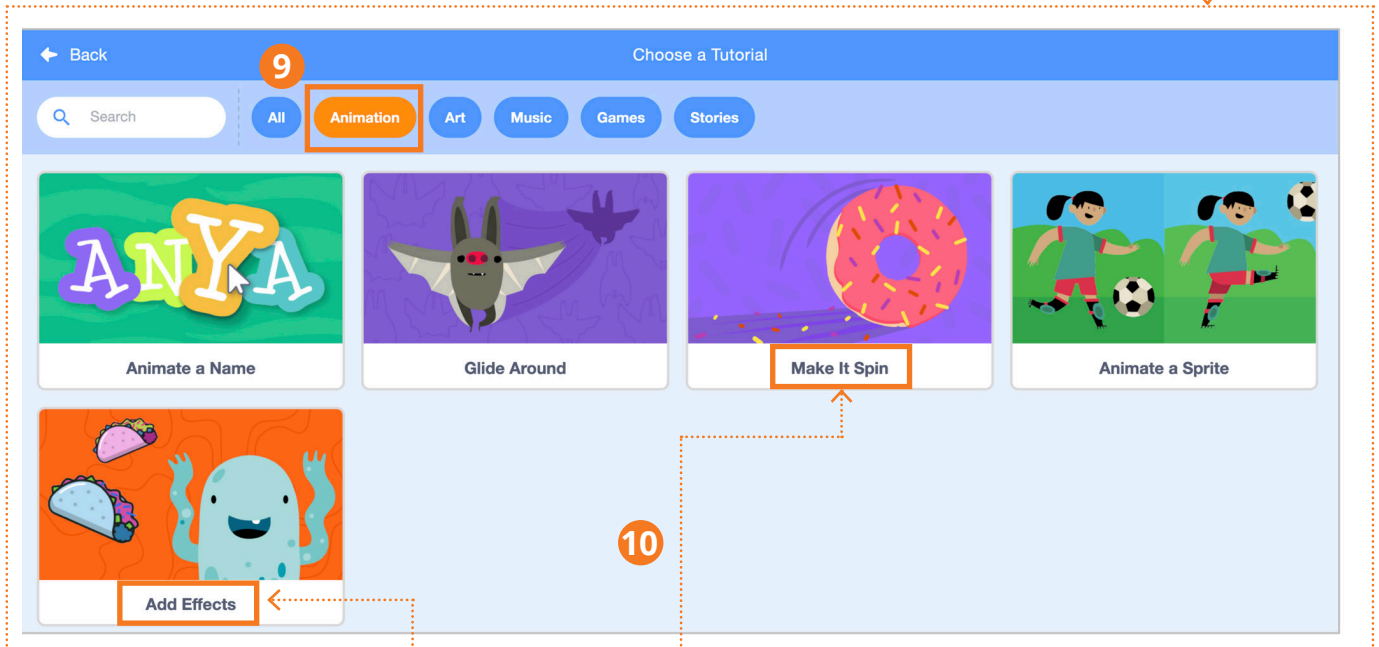
▶ **8** Now that you have created some code blocks, can you answer this question? What happens to the sprite when you click the green flag?



CT Tips

when green flag clicked is a typical **event**. An **event** is something that triggers actions.

- **9** Click on **“Tutorials”** on the menu bar again and then choose the **Animation** category on the “Tutorials” page.



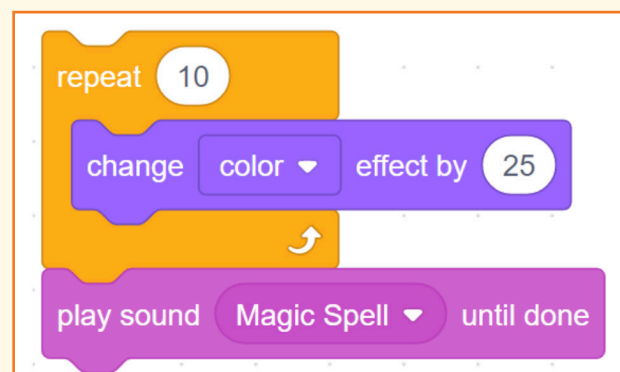
- **10** Complete the **Add Effects** and **Make It Spin** tutorials.



Now that you have created some code blocks from the two tutorials, can you answer this question? What is the difference between the use of forever and repeat blocks?



Make It Spin

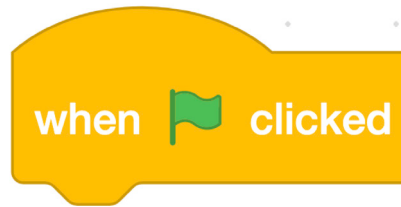


Add Effects

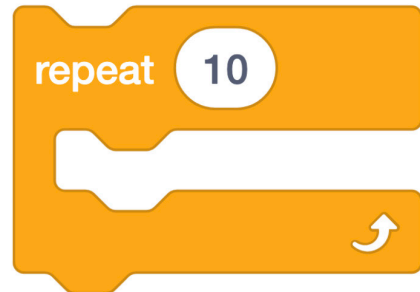
COMPUTATIONAL THINKING CONCEPTS

The following are the computational thinking concepts learnt in Lesson 1.

1 Events



2 Repetition

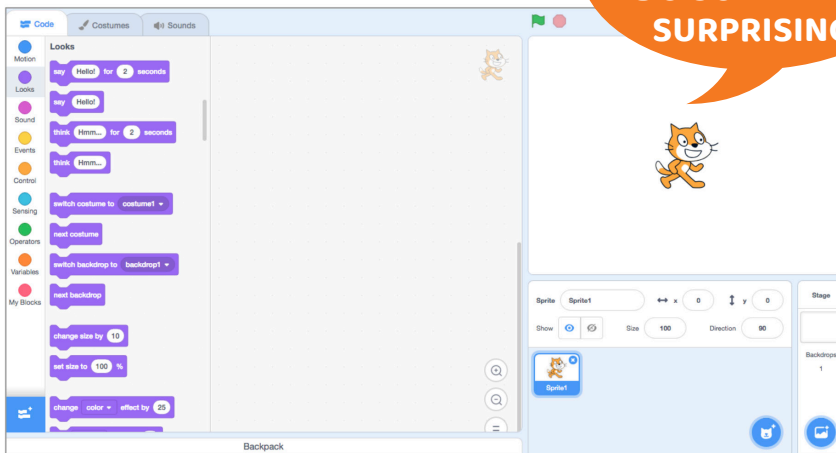


Lesson 2

SCRATCH SURPRISE



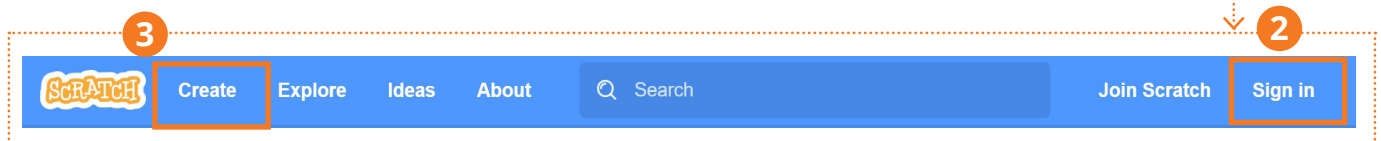
CAN YOU MAKE
THE SCRATCH CAT
DO SOMETHING
SURPRISING?



In this activity, you will create a new project with Scratch and use your homework ideas to make the cat do something surprising! What will you create?

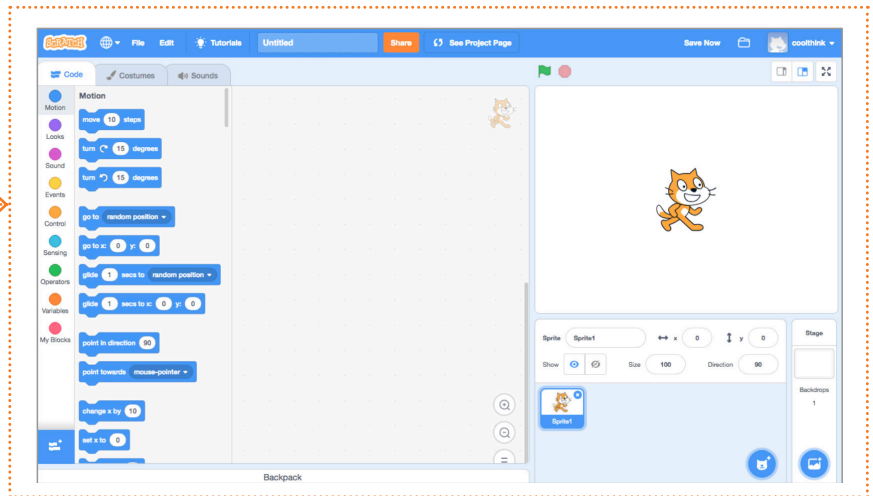
START HERE

- ▶ 1 Go to the Scratch website: <https://scratch.mit.edu>
- ▶ 2 Sign into your account.



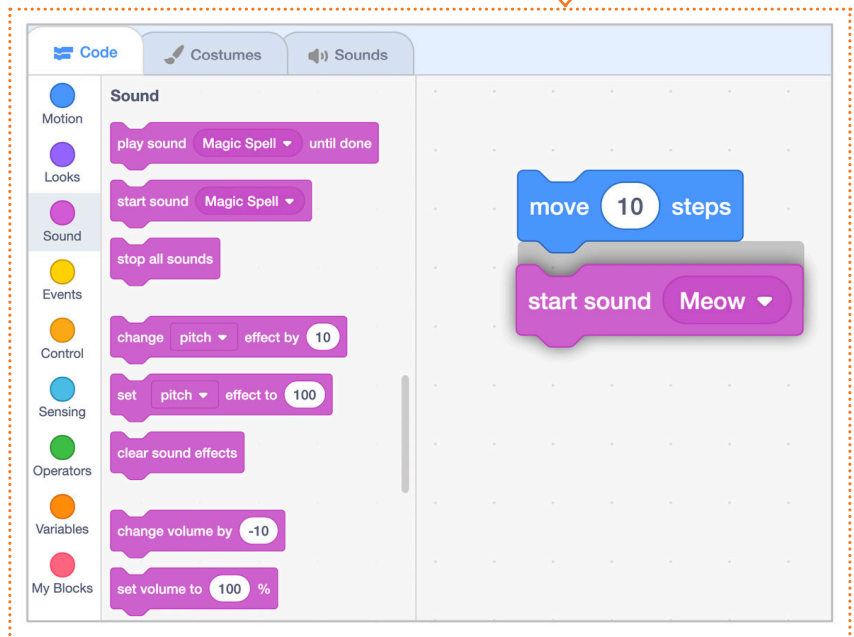
- ▶ 3 Click on the **Create** tab located at the top left of the browser to start a new project.

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



4

- ▶ **5** Explore different Scratch blocks too! 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.



5

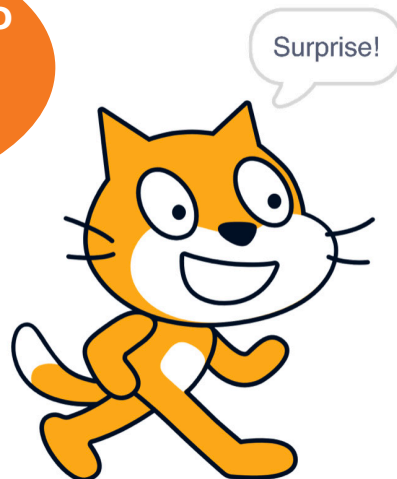


A **sequence** is an ordered set of steps that can be executed by a computer, in this case, the cat.

Lesson 2

SCRATCH SURPRISE

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

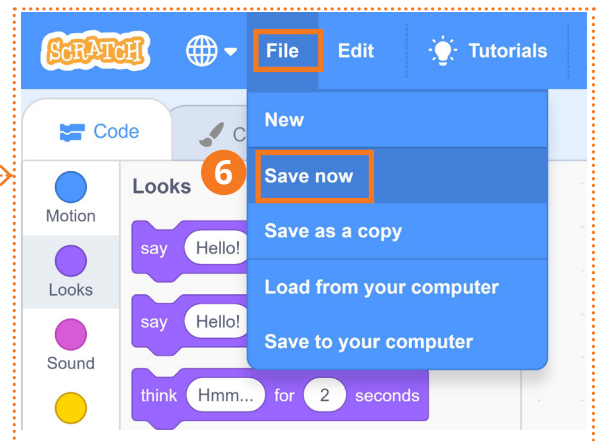
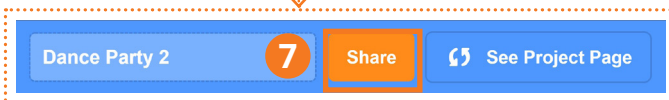


Studios are collections of Scratch projects. Follow the steps below to add your Scratch Surprise program to your class' Surprise Studio on the Scratch website.

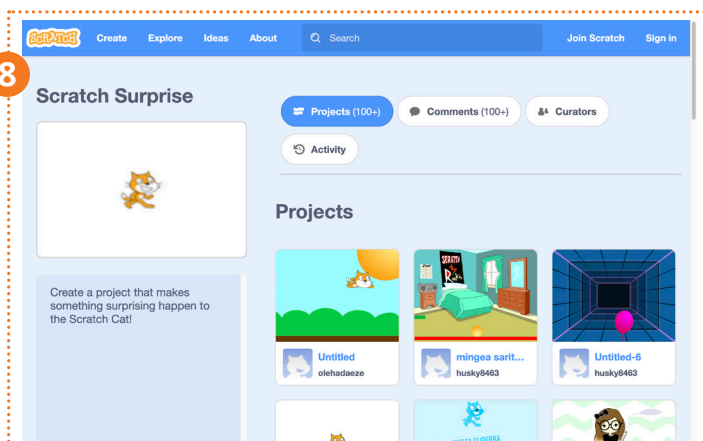
START HERE

- ▶ **6** Save your project by clicking **"Save now"** under the File menu.

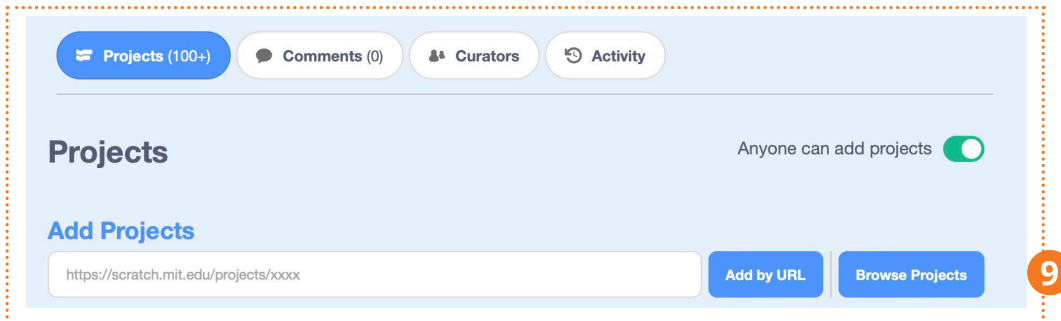
- ▶ **7** Click the orange **"Share"** button.



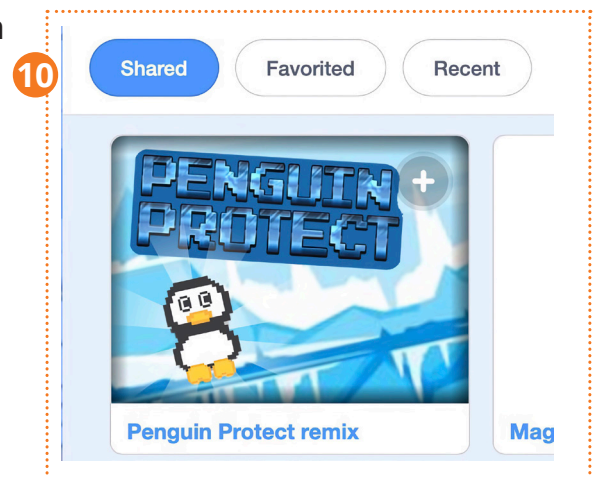
- ▶ **8** Go to the Scratch Surprise Studio using the link your teacher provides to you.



- ▶ **9** Click on the **“Browse Projects”** button at the top of the page to show your projects, favourite projects and recently viewed projects.



- ▶ **10** Find your Scratch Surprise project and then click **“+”** to add your project to the Studio.

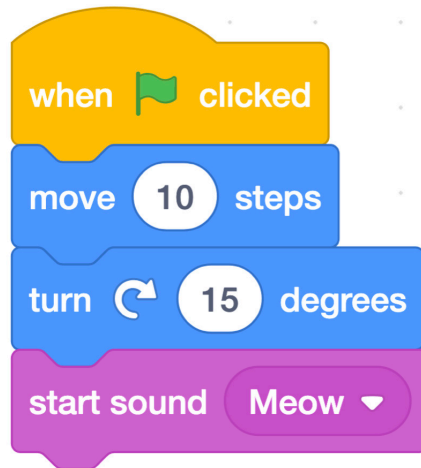


This work is adapted from Pp.10 and 12 of the Creative Computing Learner Workbook created by the ScratchEd team at the Harvard Graduate School of Education, and is licensed under a Creative Commons Attribution-sharealike 4.0 International License.

COMPUTATIONAL THINKING CONCEPTS

The following is the computational thinking concept learnt in Lesson 2.

1 Sequences



COMPUTATIONAL THINKING PRACTICES

The following are the computational thinking practices used in this unit.

1 Being incremental and iterative:

- a) Add blocks
- b) Run your project to see what the blocks do

2 Testing and debugging:

- a) Run the blocks to see what they do

3 Algorithmic thinking:

- a) Create a series of steps to make the cat do something

Two Stars and a Wish Worksheet

Name of Project: _____

Name of Creator: _____

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



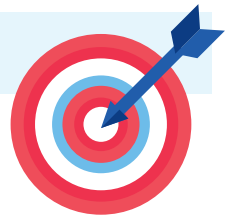


What would you like to add or change to make this project better?



Lesson 1

SCRATCH DANCE PARTY



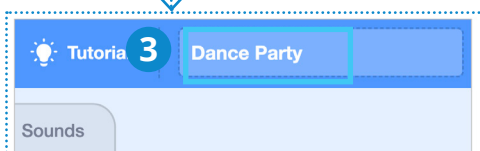
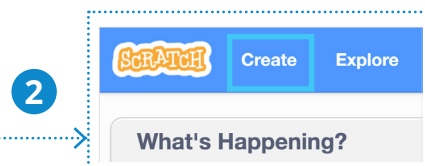
Let's take what we've learnt so far with Scratch and really make our sprites move! You will make a dance animation where you will add music, sound and dance movements for a fun dance party!

IT'S TIME TO DO SOME DANCING!

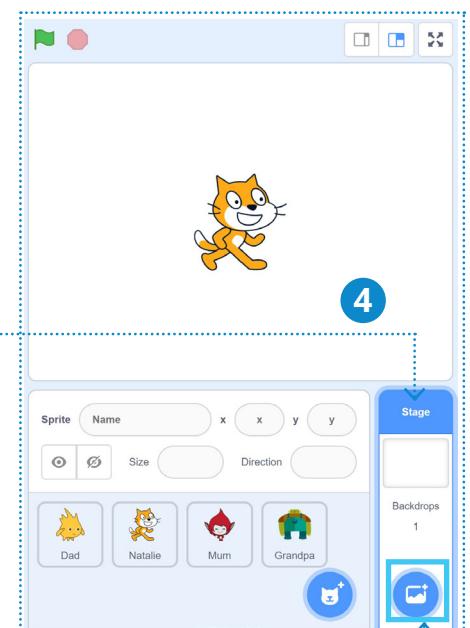


START HERE

- ▶ 1 Sign into your account at scratch.mit.edu
- ▶ 2 Go to **"Create"** to start a new project
- ▶ 3 Name it **"Dance Party"**

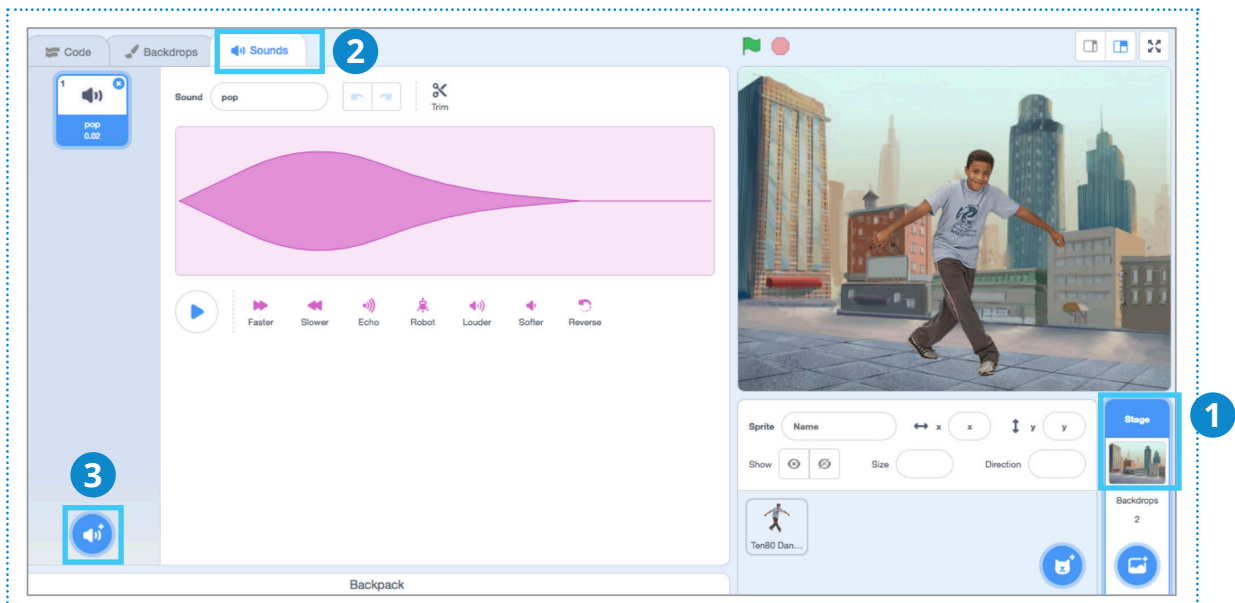


- ▶ Choose a good dance backdrop for your stage:
 - 4 Click on the **"Stage"**.
 - 5 Click on the **"Choose a Backdrop"** icon.



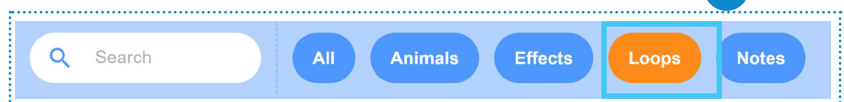
5

ADDING MUSIC



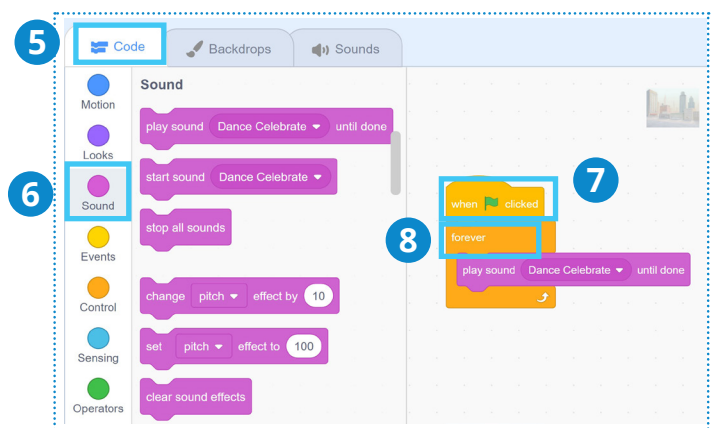
► Choose some background music for your dance.

- 1 Click on the Stage.
- 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 for the dance.



► Now add your music to the stage.

- 5 Click on Code.
- 6 Play sound until done!
- 7 Add a **“when clicked block”** event block.
- 8 Add the **forever** block.



DANCING SPRITE

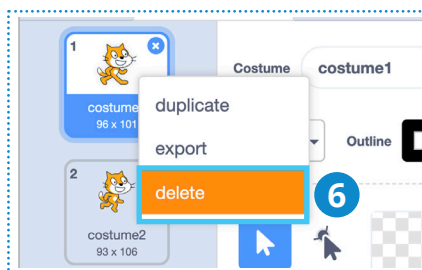
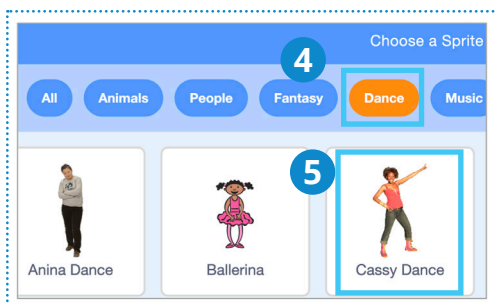
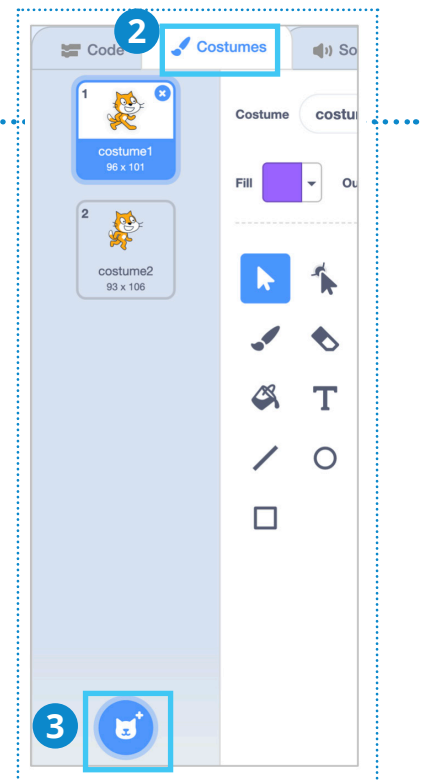
► Choose a good sprite for dancing!

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 **“Dance”** category on the top to see a large set of dance costumes.

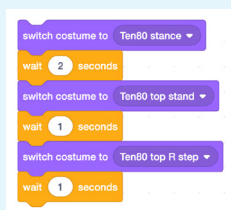
5 Add multiple costumes for your sprite or person.

6 Delete the original Scratch cat costumes.



CT Tips

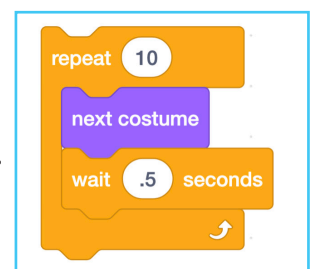
“switch costume and then wait” is an example of a **sequence**.



► Remember how to switch costumes?

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

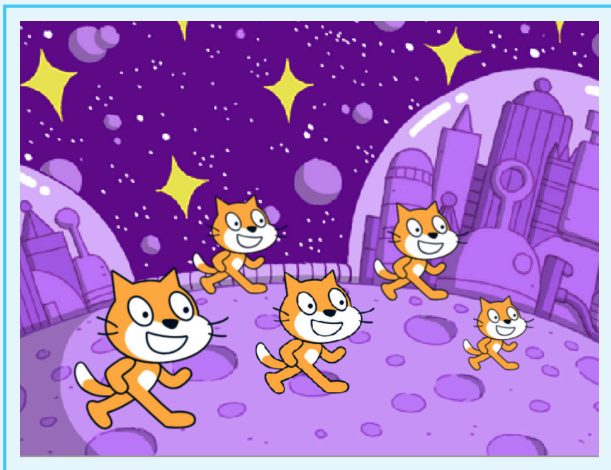
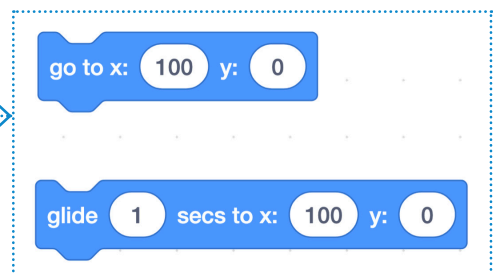
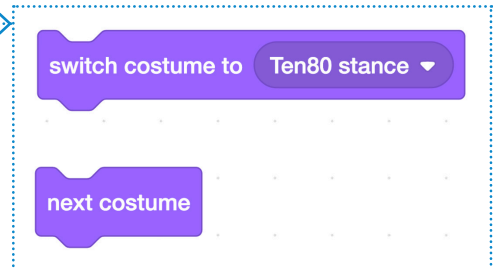
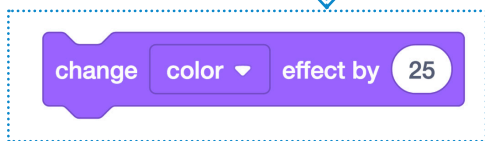
► Don't forget about the **“repeat”** block!



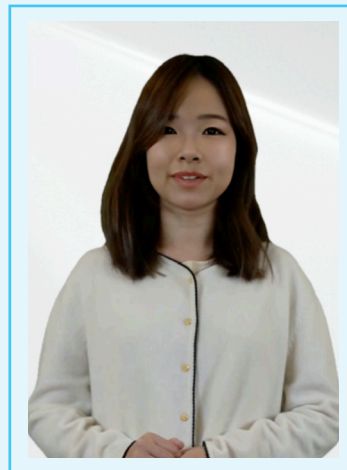
FANCY MOVES

▶ Now, let's make your sprite do some fancy moves. Remember how to...

- ▶ Change costumes
- ▶ Move the sprite around the stage
- ▶ Change colours!



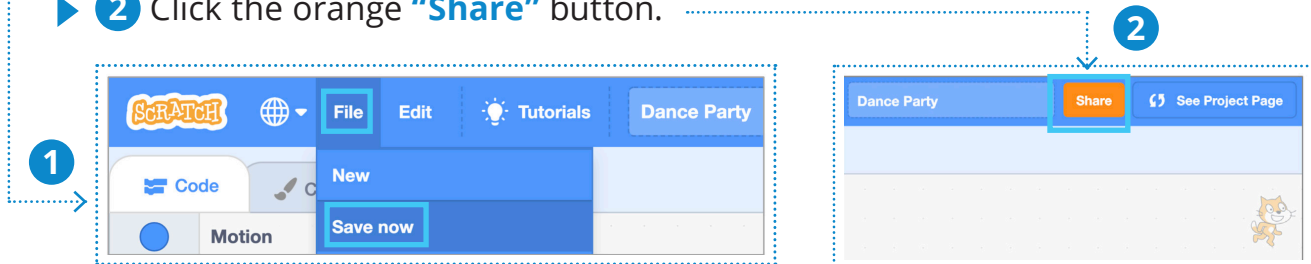
Do you think you can make your sprite do a Gangnam Style Dance?



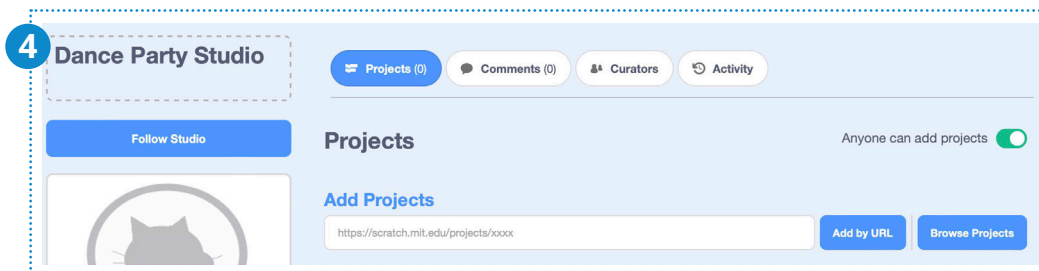
What about one of the video dances?

DANCE, DANCE, DANCE!

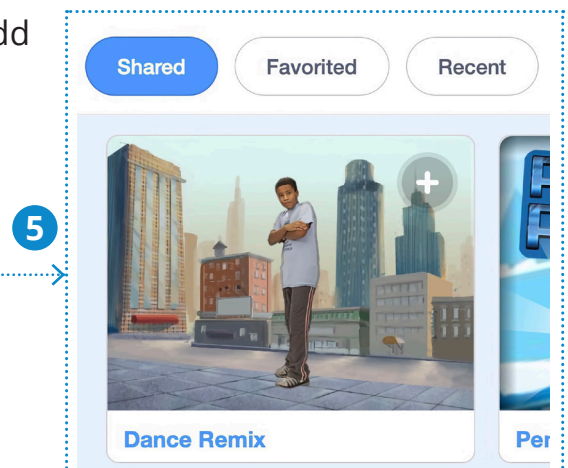
- ▶ When you finish, you will add your project to your teacher's Studio.
- ▶ **1** Save your project by clicking **"Save now"** under the File menu.
- ▶ **2** Click the orange **"Share"** button.



- ▶ **3** Go to your teacher's Dance Party Studio (they will give you a URL).
- ▶ **4** Click the **"Browse Projects"** button.



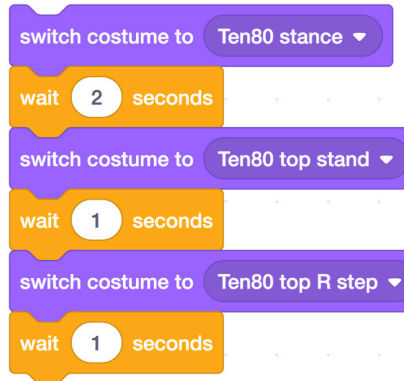
- ▶ **5** Select your project and then click **"+"** to add it to the Studio.



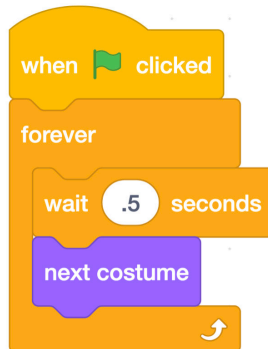
COMPUTATIONAL THINKING PRACTICES

The following are the computational thinking concepts learnt in Lesson 1.

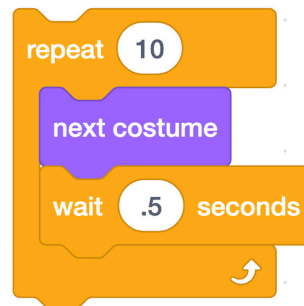
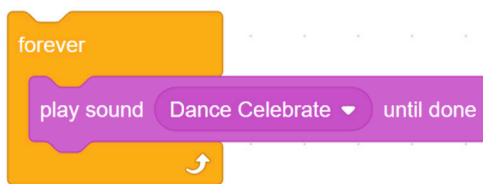
1 Sequences



2 Events



3 Repetition



Lesson 2

SCRATCH DANCE PARTY



LET'S
MAKE THIS
A PARTY!

In this lesson, you will add more dancers to your dance project to make it more fun and interesting!

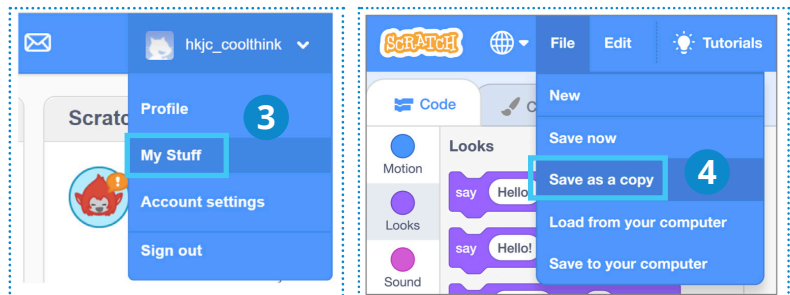


START HERE

- ▶ Pair up with the person to your left.
- ▶ Show each other your dance party projects.
- ▶ Pick the best part of your partner's dance and tell them why you like it.
- ▶ Share how you coded your dance with your partner.



- ▶ **2** Sign into your account at scratch.mit.edu
- ▶ **3** Click on My Stuff.
- ▶ **4** Open your Dance Party project
- ▶ **5** Save as a copy.
- ▶ **6** Name it **Dance Party 2**.



5

Dance Party 2

Share

See Project Page

LET'S ALL DANCE!

- ▶ **7** Add another Sprite to your dance party.
- ▶ **8** See if you can make your sprites dance together. Or take turns!

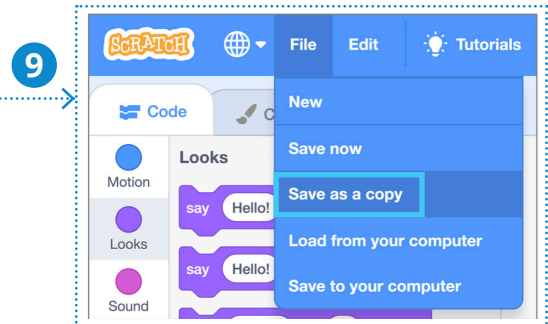


Parallelism: Multiple sprites can dance together in parallel.

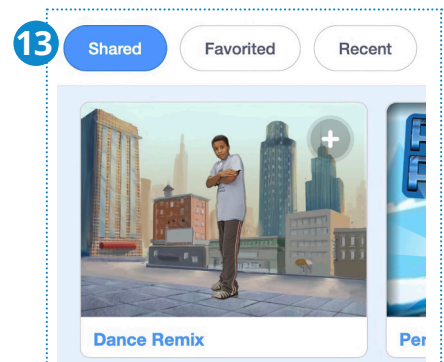
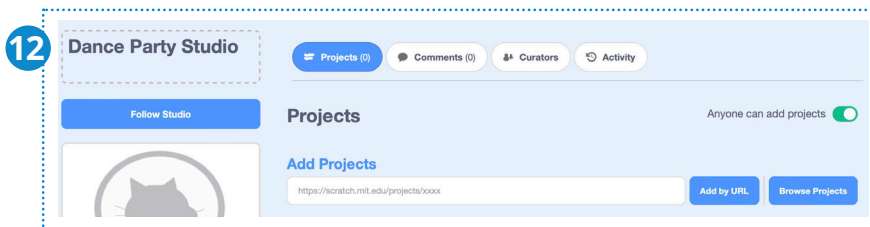


When you finish, you will add your project to your teacher's Studio

- ▶ **9** Save your project by clicking **"Save now"** under the File menu.
- ▶ **10** Click the orange **"Share"** button.



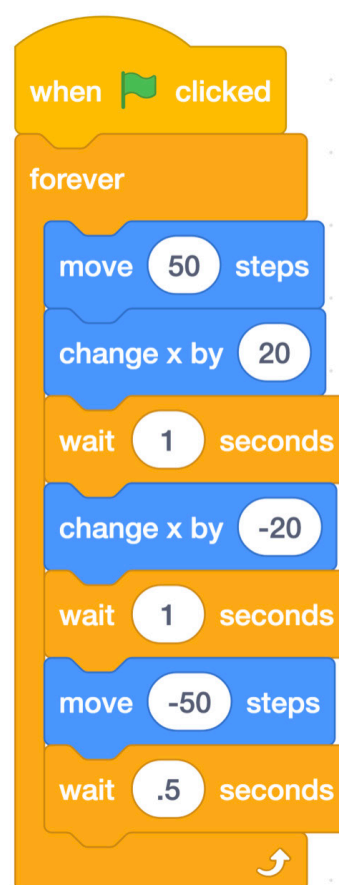
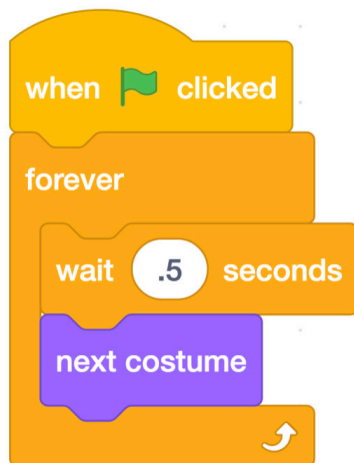
- ▶ **11** Go to your teacher's Studio (they will give you a URL).
- ▶ **12** Click the **"Browse Projects"** button.
- ▶ **13** Select your project and then click **"+"** to add it to the Studio.



COMPUTATIONAL THINKING CONCEPTS

The following is the computational thinking concept learnt in Lesson 2.

1 Parallelism



COMPUTATIONAL THINKING PRACTICES

The following are the computational thinking practices used in this unit.

1 Reusing and remixing:

- a) Watch classmates' projects and remix dance ideas

2 Being incremental and iterative:

- a) Make a sprite dance
- b) Add a second sprite and have them dance together

3 Testing and debugging:

- a) Test that the sprite dances as expected using code blocks
- b) Test that the two sprites dance together as expected using code blocks

4 Algorithmic thinking:

- a) Use code blocks and sequences correctly to get the sprites to dance the way you want them to

Two Stars and a Wish Worksheet

Name of Project: _____

Name of Creator: _____

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





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



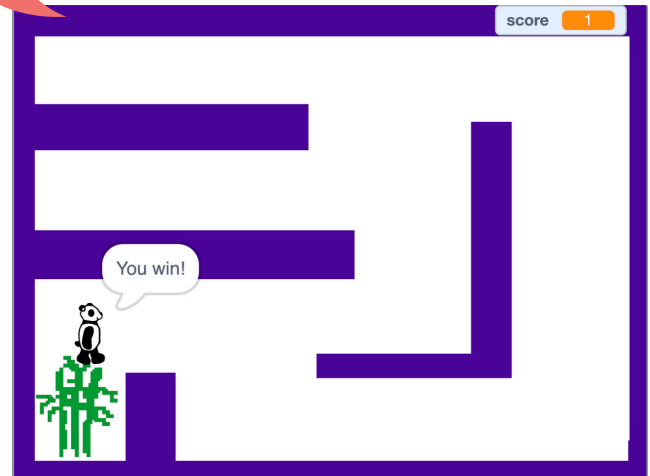
Lesson 1

SCRATCH MAZE GAME

LET'S LEARN TO
MAKE A GAME WITH
SCRATCH!

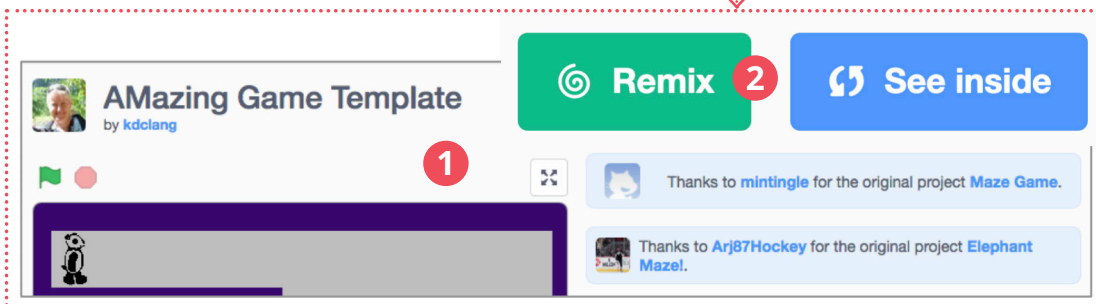
In this game, you will guide the panda to its bamboo forest with your keyboard, and prevent it from hitting the wall of the maze. You will earn 1 point if you reach the goal and lose 1 point if you hit the wall.

In Part 1, you will learn how to control the panda's movements with arrow keys on your keyboard.

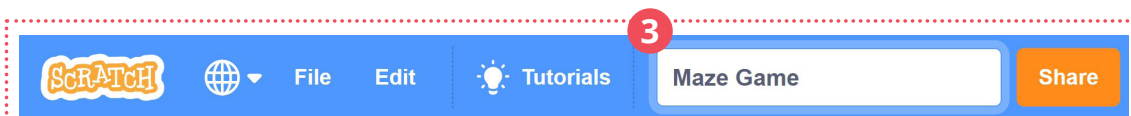


START HERE

- ▶ 1 Sign into your account at scratch.mit.edu
- ▶ 2 Open the Maze Game Template project: <https://scratch.mit.edu/projects/222153103> and click on the **"Remix"** button.

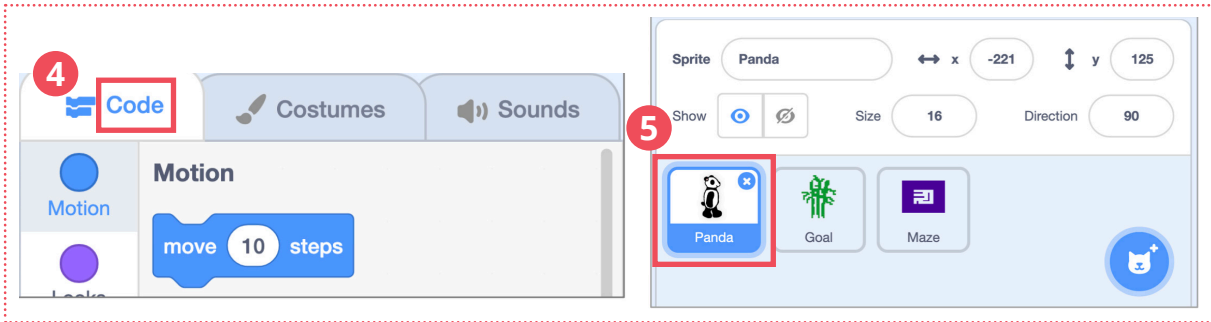


- ▶ 3 Rename the project as **"Maze Game"**.



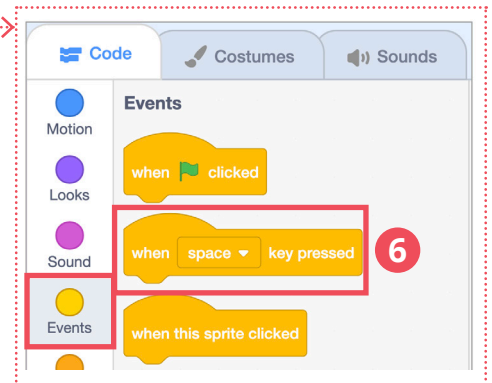
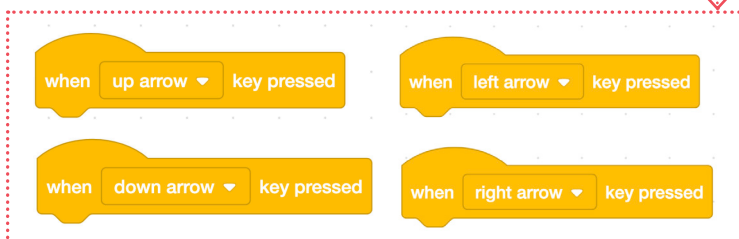
MOVING THE PANDA

- ▶ **4 5** We are now going to make the sprite move with keyboard presses. Go back to the **"Code"** tab and select the Panda sprite.



- ▶ **6** Go to the **"Events"** drawer under **Code** tab and drag four **"when space key pressed"** blocks onto the coding area.

- ▶ **7** For each of the four blocks, change the **space** key entry to **up arrow**, **down arrow**, **left arrow** and **right arrow**.



- ▶ **8** To make the sprite move when the keys are pressed, use blocks from the **Motion** drawer. You will need to use the **change x by #** and **change y by #** blocks.

- ▶ Try to figure out which block and number (10 or -10) goes under which key event.

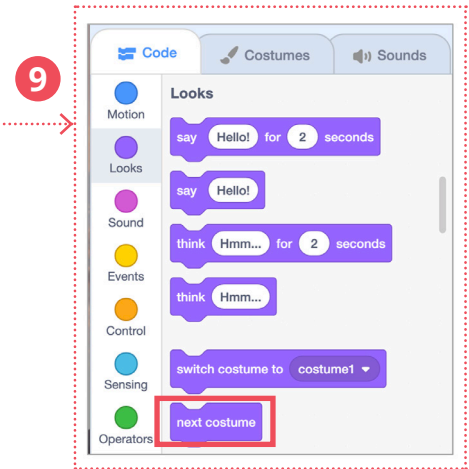


CT Tips

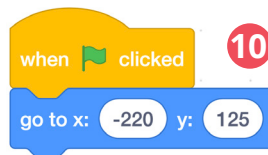
Pressing the keys are **events** that control where the sprite moves.

ALMOST DONE!

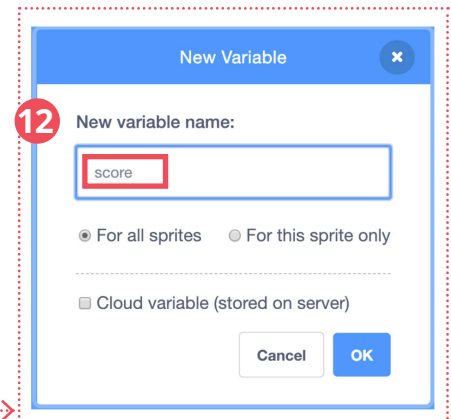
▶ **9** How would you change the sprite’s costume to make it look like it is walking? Hint: Think about using the **next costume** block.



▶ Make your sprite go to your desired starting coordinates when the green flag is clicked.

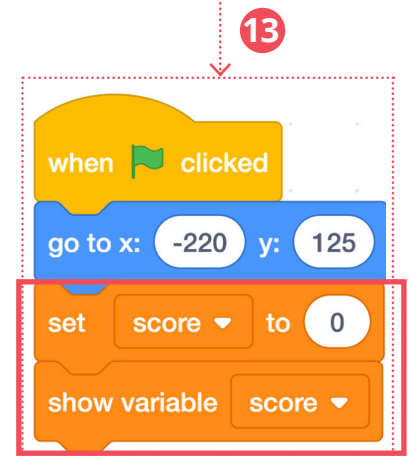


▶ **11** To add a variable for recording the score, go to the **Variables** drawer and click **Make a Variable**.



▶ **12** Name the new variable as **score**.

▶ **13** Drag **set score to 0** and **show variable score** blocks to **when the green flag is clicked** blocks.



▶ **14** Play your game. What works? What doesn't? Discuss with your teacher and your classmates.

▶ **15** Save your project by clicking **Save now** under the **File** menu

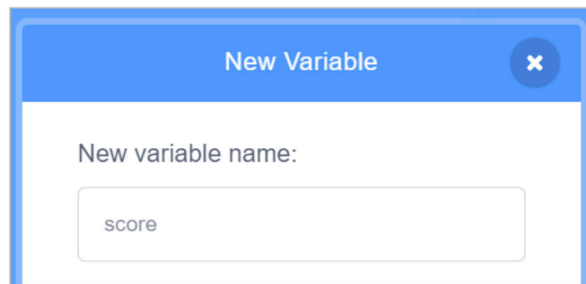
COMPUTATIONAL THINKING CONCEPTS

The following are the computational thinking concepts learnt in Lesson 1.

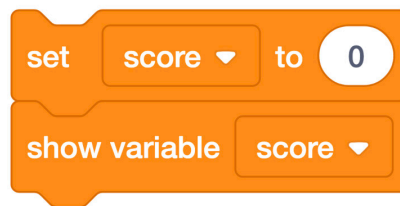
1 Events



2 Naming



3 Manipulation of data and elementary data structures:

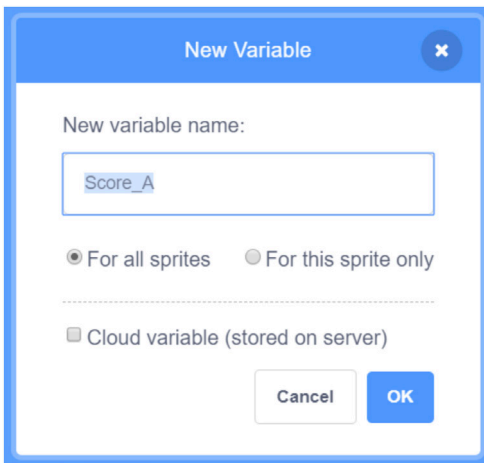


Unplugged Activity: Variables

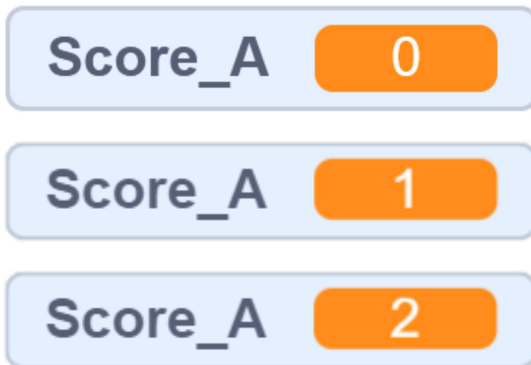
What are variables?

Variables are used to store values. Variables have the following properties.

☆ Variables have names.



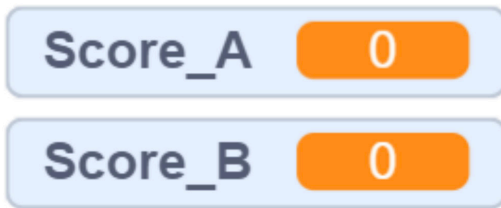
☆ A variable can only store one value at a time.



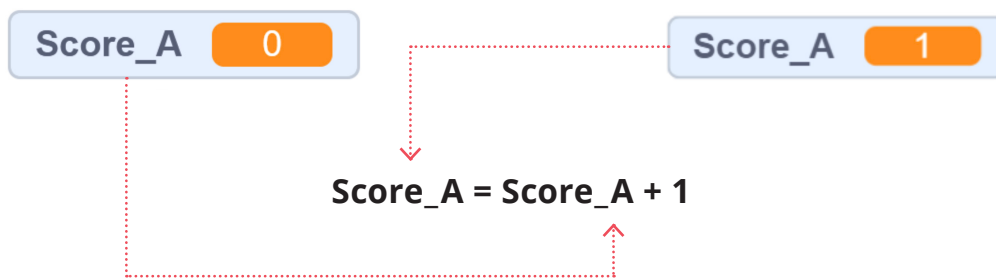
Unplugged Activity: Variables

☆ **The variable should have an initial value.**

When the game starts, the value of Score_A, and Score_B are reset to 0.



☆ **The value of the variable can be updated.**



☆ **The value of the variable can be a string, or a number.**

Score_A = 3
Score_B = 1
Winner = "A"



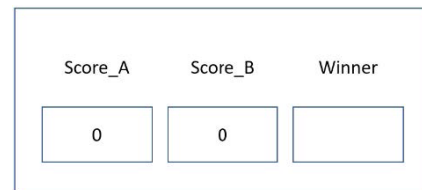
Unplugged Activity: Variables

Who will be involved?

Two students will play a game and one student helps them keep track of the game using variables (**Score_A**, **Score_B**, and **Winner**).

What do we need?


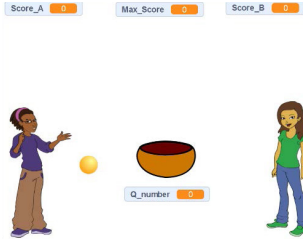
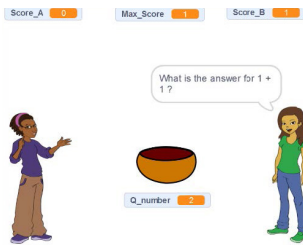
- Set of 6 question cards with answers on the back.
- Set of index cards to use for variable values.
- Whiteboard for writing down current variable values.



Using by	Tools
Student A	Student A & B will take turns in the game below: Student A picks a number card from the Question box, then goes to the table to pick the card on the table based on the number they drew from the Question box.
Student B	Student A speaks out the question printed on the card loudly and Student B needs to give the answer. Student A will tell Student C if Student B's answer is correct. The process is reversed on the next round. Student B picks the card, asks the question, and Student A answers.
Judge	<u>Correct answer</u> If Student A answers a question correctly, add 1 to the existing variable " Score_A " and add 1 to the existing score of " Score_B " if Student B answers a question correctly on their turn. <u>Incorrect answer</u> If Student A answers the question incorrectly, subtract 1 from his/her existing score (Score_A), unless the score is 0. Score_A should never become negative. If Student B answers their question incorrectly, subtract 1 from his/her existing score (Score_B), except if their score currently is 0. Once the game finishes, the letter for the student with the higher score will be written under the Winner label on the whiteboard.

Unplugged Activity: Variables

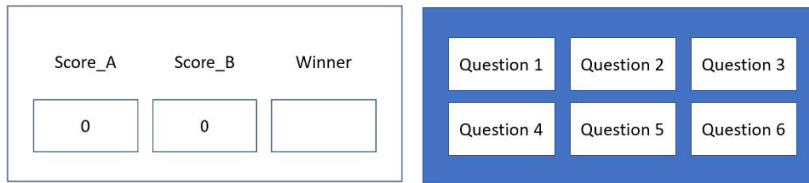
How does it run? <https://scratch.mit.edu/projects/363328697/>

	Score_A	Score_B
<p>At the beginning of the game, the two scores are reset to zero.</p> 	0	0
<p>Student takes a turn to ask a question. The other student will need to answer the question. Student A asks the first question. Correct answer will score 1 point.</p> 	0	<p>If Student B answers correctly, judge will update Score_B = Score_B + 1. If Student B answer incorrectly, judge will update Score_B = Score_B - 1. Or remain 0 if already 0.</p>
<p>After answering the question, Student B will ask a question and Student A needs to answer the question. The score calculation will be the same. The game ends until there are no more questions.</p> 	<p>If Student A answers correctly, judge will update Score_A = Score_A + 1. If Student A answer incorrectly, judge will update Score_A = Score_A - 1. Or remain 0 if already 0.</p>	1

Unplugged Activity: Variables

Procedure

- (1) Judge: (a) Write down three labels, Score_A, Score_B and Winner, on the whiteboard; (b) Give an initial value of zero (0) to Score_A and Score_B, and remove anything written under the Winner label; (c) put 6 question cards on the table.



- (2) Student A: Randomly picks a card from the table and speaks out the question loudly to Student B.
- (3) Student B: Answers the question asked by Student A.
- (4) Student A: Acknowledges to the Judge if Student B's answer is correct.
- (5) Judge: If Student B's answer is correct, calculate the new Score_B by adding 1, erase the number written under the score_B label, and write the new number under the **Score_B** label on the whiteboard. If the answer is incorrect, and if **Score_B** is greater than zero, the Judge calculates the new **Score_B** by subtracting 1, erasing and marking the new **Score_B** under the **Score_B** label on the whiteboard.
- (6) Student B: Randomly picks a **Question** card from the table and speaks out the question loudly to Student A.
- (7) Student A: Answers the question asked by Student B.
- (8) Student B: Acknowledges to the Judge if Student A's answer is correct.
- (9) Judge: If Student A's answer is correct, adds 1 point to **Score_A**. If the answer is incorrect, and if **Score_A** is greater than zero, the Judge subtracts 1 from **Score_A**.

Repeat steps (2) to (9) to continue the game until no more cards can be drawn from **Questions** on the table.

If Student A has the higher score, then the Judge write "A" under the **Winner** label on the whiteboard. The Judge then announces, "Student A is the winner and they have a score of [Score_A]."

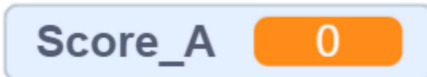
If Student B has the higher score, then the Judge write "B" under the **Winner** label on the whiteboard. The Judge then announces, "Student B is the winner and they have a score of [Score_B]."


If the scores are the same, the Judge write "None" under the **Winner** label on the whiteboard and announces, "The game is a tie".

Unplugged Activity: Variables

Summary of What We Learned

☆ Variables have names.

Score_A 

Score_B 

☆ Variables have values.

Score_A = 1

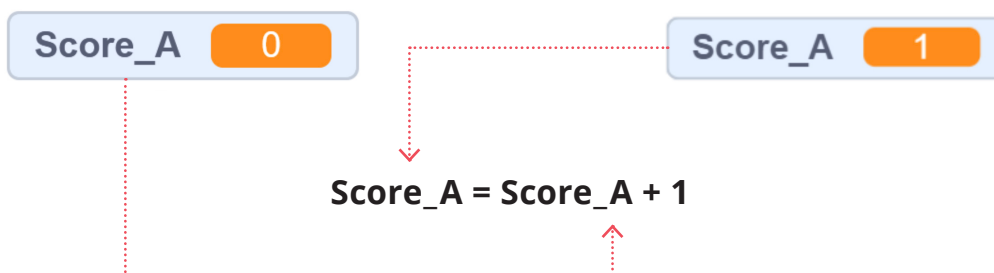
☆ The variable should have an initial value.

When the game starts, the value of Score_A, and Score_B are reset to 0.





☆ The value of the variable can be updated.



☆ The value of the variable can be a string, or a number.

Score_A = 3
Score_B = 1
Winner = "A"



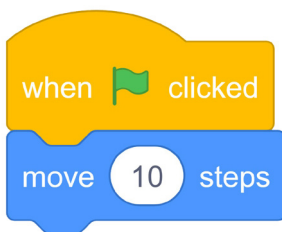
Exploring Forever and If-Then Worksheet



In this activity, we will explore the **forever** and **if-then** blocks. Please follow the instructions and answer the questions in the boxes with words or drawings of blocks.

We will attempt to make the Scratch cat run back-and-forth across the screen forever as shown in class when we click the green flag.

1. Go to Scratch and create a new project.
2. Make the block below.

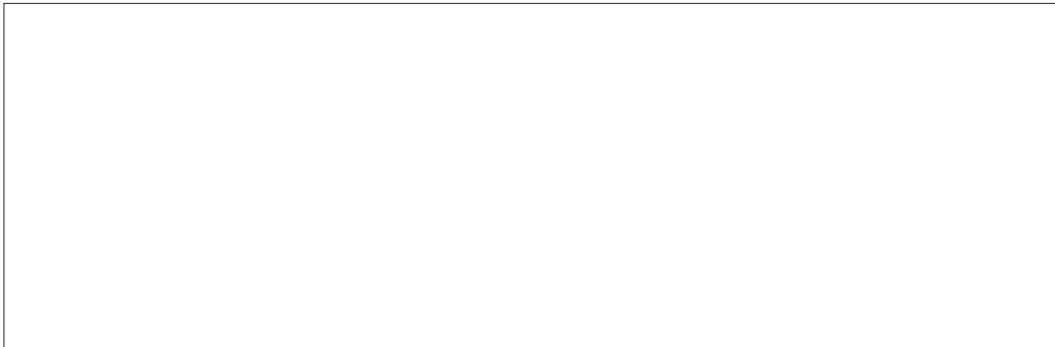


3. Try clicking the green flag a few times. Note what happens.
4. We want the cat to continuously move to the right and eventually come back, but clicking so many times is tiring... Try to make the block below. (Hint: The **forever** block is in the **Control** drawer)

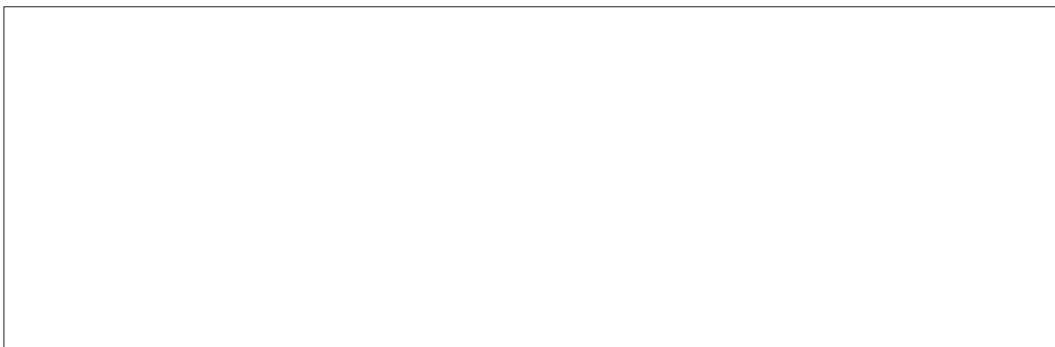


Exploring Forever and If-Then Worksheet

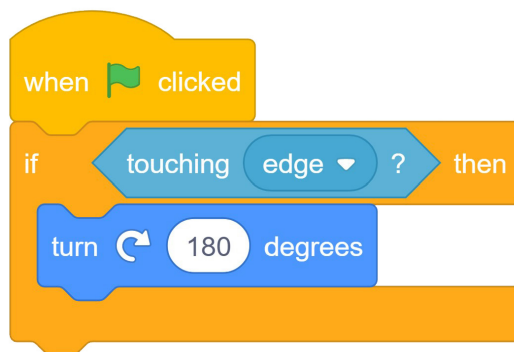
5. What happens when you click the green flag?
What happens when you click and drag the cat away from the edge of the screen?



6. Based on what you saw just now, what do you think the **forever** block does?



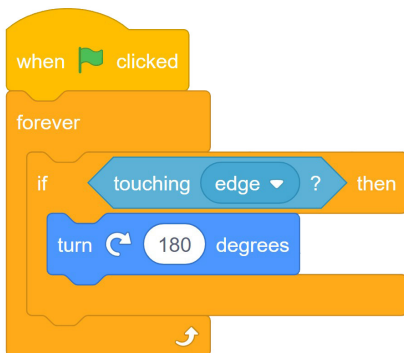
7. We want the cat to turn when it arrives at the edge. Try to make the block below.



Exploring Forever and If-Then Worksheet

8. Drag the cat to the middle of the screen and then click the green flag. Note what happens.
9. Drag the cat to the edge of the screen and click the green flag again. Note what happens.
10. Based on what you saw just now, what do you think the **if-then** block does?

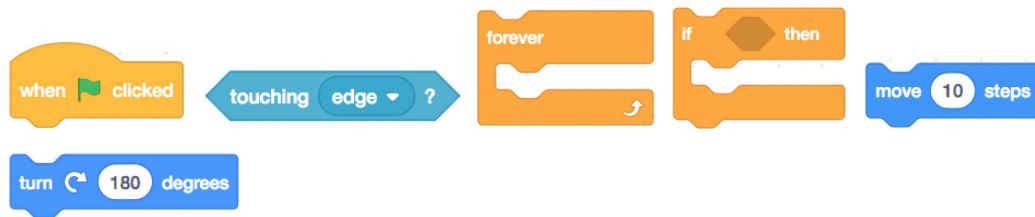
11. We want the cat to turn so that it faces the other way every time it reaches the edge because this allows the cat to move away from the edge by itself. Try adding the **forever** block to make the block below.



12. Try clicking the green flag when the cat is in the middle of the screen and at the edge of the screen. Note what happens.

Exploring Forever and If-Then Worksheet

13. We can almost reach our goal of making the cat run back-and-forth across the screen! Use a combination of the blocks we just learnt to make it happen (listed below). Draw your code in the box below.



Questions and Answers

Rules for scoring the maze game:

When the game starts, the score is 0. When the ball collides with any wall, 1 point will be deducted from the score. When the bamboo sprite is reached, 1 is added to the score. There is no further reduction once the score reaches 0. Cut out the questions/answers on each row and then fold into half.

Question 1

John starts the game, what will the score be?

Question 2

Mary plays the game and reaches the bamboo sprite twice, and then the ball collides with one of the walls. What is her score?

Question 3

Peter plays the game and reaches the bamboo sprite 5 times without hitting any walls. What is his score?

Question 4

Amy plays the game and hits the walls 5 times, then she finally touches the bamboo sprite. What is her score?

Question 5

John plays the game and reaches the bamboo sprite 5 times, then the ball collides with 4 walls before hitting the bamboo sprite again. What is his score?

Question 6

June plays the game and hits the wall once, then reaches the bamboo sprite. Then she hits the wall twice and reaches the bamboo sprite a second time. What is her score?

Questions and Answers

Answer 1

0. When the game starts, the score will be reset to 0.

Answer 2

1. Mary scores 2. Colliding with the wall will deduct 1 from the score. $2-1=1$

Answer 3

5. Every time you reach the bamboo sprite, the score will increase by 1.

Answer 4

1. Score starts at 0 and there is no deduction for the 5 hits on walls. Then she gains 1 at the end of the game.

Answer 5

2. $5 - 4 + 1 = 2$.

Answer 6

1. Her score goes back to 0 when she hits the wall, so her score is 1 for the second touching of the bamboo.

Score Cards

0

1

2

3

A

None

Score Cards

0

1

2

3

B

COMPUTATIONAL THINKING PRACTICES

The following are the computational thinking practices used in this unit.

1 Reusing and remixing:

- a) Use the template project and remix it

2 Being incremental and iterative:

- a) Add blocks to control the sprite movement
- b) Make a score variable
- c) Add if-then to the project
- d) Add the “forever” block

3 Testing and debugging:

- a) Test and debug collision
- b) Test that the score is updated when the sprite wins or loses

4 Algorithmic thinking:

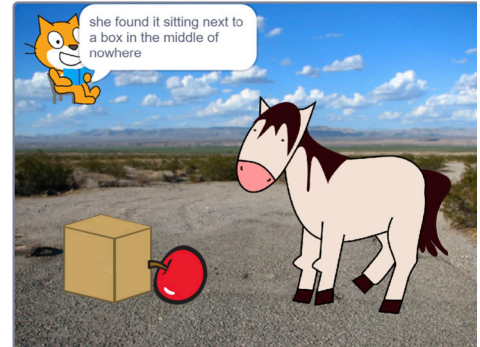
- a) Learn to make the sprite move based on key press events

SCRATCH FINAL PROJECT



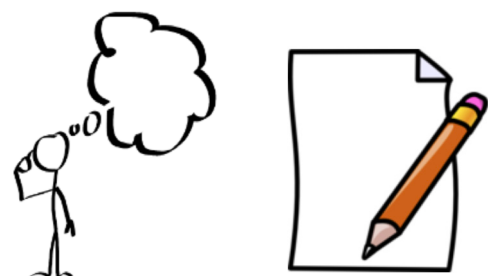
HERE IS YOUR
CHANCE TO SHOW
OFF YOUR CODING
SKILLS!

In this project, you will design and build a Scratch project that can be of two general types - a story or a greeting card. It is up to you what kind of project you will make.



START HERE

- ▶ Think about the kind of project you and your partner would like to make.
- ▶ What was your favourite Scratch project so far?
- ▶ Would you like to make a story or a greeting card?
- ▶ Would you like to take one of your previous Scratch projects and build something based on that?
- ▶ With your partner, fill out the Scratch Final Project Design Worksheet to plan out your project.

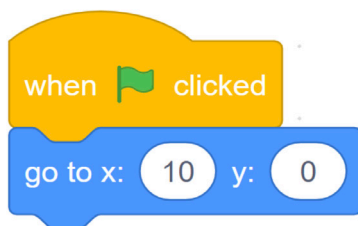


CODING CONCEPTS

As you plan your project, think about the blocks you have used before. For your project, try to use as many of the Scratch blocks you've learned about so far.

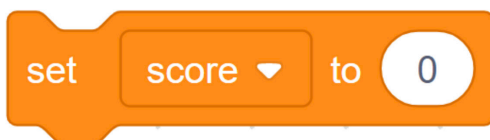
EVENTS

Control sprites based on an event like a key press.



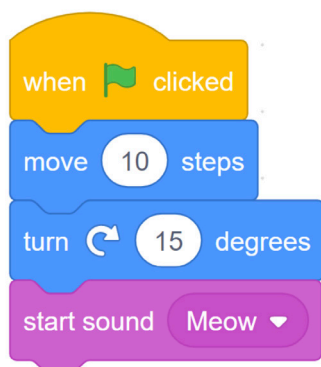
VARIABLES

Variables can help you save information and update it in your Scratch project.



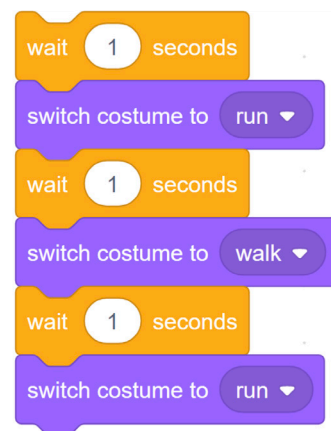
SEQUENCES

Multiple instructions put together form a sequence that happens in order.



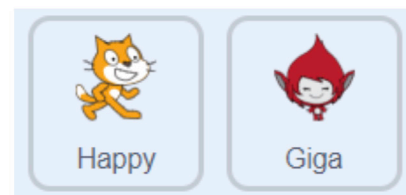
COSTUME CHANGES

Show movement by changing a sprite's costume!



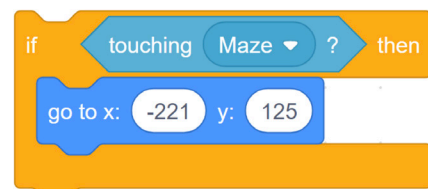
NAMING

Give an good name to the sprites is also important.



CONDITIONALS

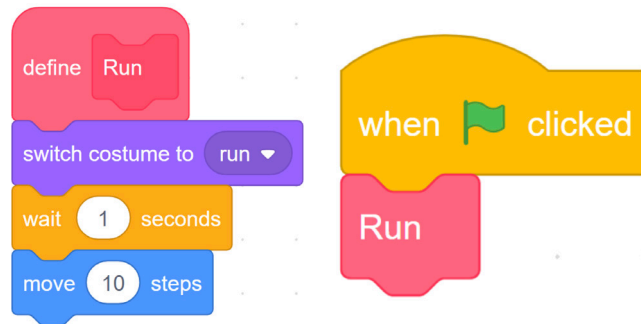
Sometimes you need to make decisions. Trigger sprite action based on **if-then** blocks.



MORE CODING CONCEPTS

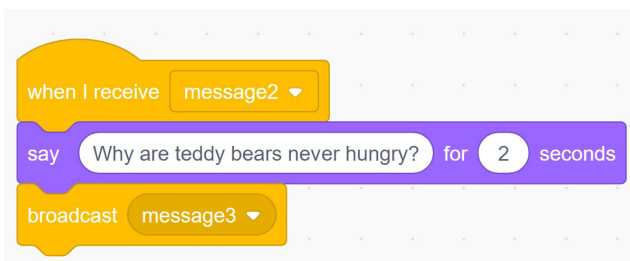
MY BLOCKS

Use custom blocks (My Blocks) to break complex tasks into smaller blocks of code that can be reused.



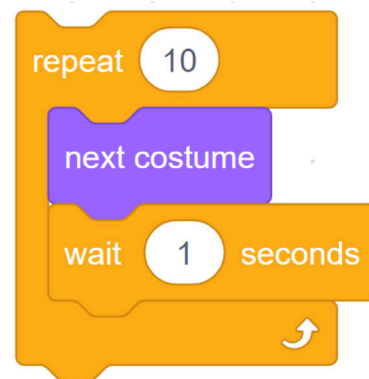
BROADCASTS

Broadcast and **when I receive** blocks let sprites talk to each other to trigger actions.



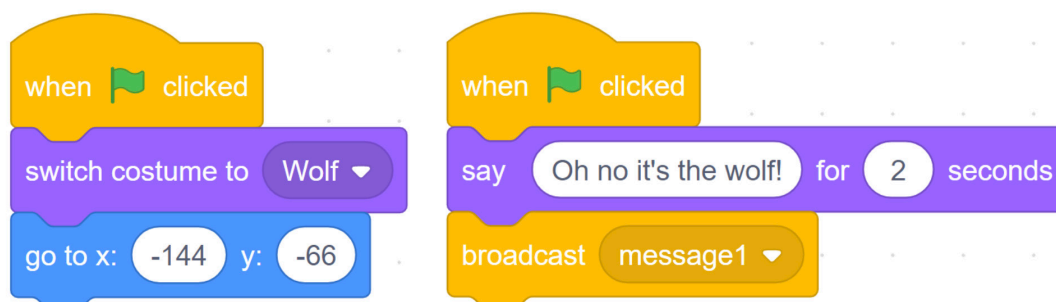
REPETITION

Repeat blocks let your sprite easily do things over and over.



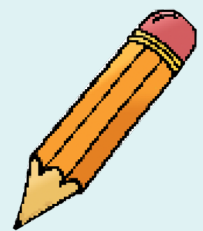
PARALLELISM

You can have sprites do things at the same time with the help of events or broadcasts.



TIME FOR FEEDBACK

- ▶ When your teacher tells you, your team should pair up with another team.
- ▶ Take turns showing each other your Scratch Final Project Design Worksheet and explaining your project.
- ▶ Listen carefully to your classmates when they explain the project. Ask questions if anything is not clear.
- ▶ Pick two things you like about the project and tell them.
- ▶ Now tell them one wish you have for their project.
- ▶ When your classmates ask questions or give suggestions about your project, write what they say on the last page of the Scratch Final Project Design Worksheet. Write everything down, even if you don't agree!
- ▶ When you go back to finish your project, read your classmates' suggestions. If you want to update your project design based on their comments, make those changes now.



ARE YOU READY TO CODE?

- ▶ Sign into your account on scratch.mit.edu and create a new Scratch project.
- ▶ Give it an appropriate name.
- ▶ Using your Scratch Final Project Design Worksheet, add your sprites to the project.
- ▶ Add a backdrop to your stage for your story or card.
- ▶ Start Coding!

Remember Rules for Pair Programming



1. Be respectful
2. Talk to one another about the work
3. Explain what you are doing
4. Think ahead and make suggestions
5. Switch roles often



1. Be a bossy navigator
2. Grab the driver's mouse or keyboard

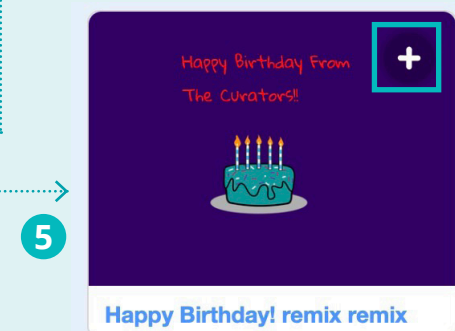
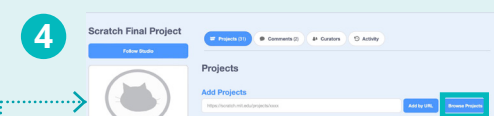
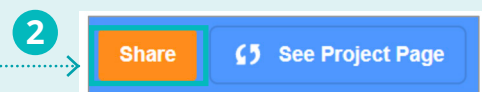
DESIGN REVISION

- ▶ Now you should have completed part of your project.
- ▶ When your teacher tells you, your team should pair up with another team in your class.
- ▶ Take turns trying out each other's project.
- ▶ Ask the other group to answer the following questions about your project:
 1. What did you like about the project?
 2. Was there any part of the project that did not work well or was unclear?
 3. Is this project similar to your own project? How?
 4. What could be added to make the project more interesting or fun?
- ▶ Based on their answers, complete the Scratch Final Project Peer Feedback Worksheet.
- ▶ Update your design and the remaining tasks in your Design Worksheet, based on the feedback from your peers.
- ▶ There should be at least one improvement or added feature integrated into your design.

FINISHING UP

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

- 1 Save your project by clicking the **Save** button at the top left.
- 2 Click **Share** button at the top centre.
- 3 Go to your teacher's Scratch Final Project Studio (they will give you a URL).
- 4 Click the "**Browse Projects**" button.
- 5 Select your project and then click **"+"** to add it to the Studio.



PRESENTATION

Following your teacher’s instructions, you may present orally or assess each other’s work using the Scratch Studio and Two Stars and a Wish Worksheet.

- 1** If you present orally, you should explain:
 - ▶ The theme/goal of the project (to tell a story, or a greeting card);
 - ▶ What you are most proud of accomplishing with the project;
 - ▶ Difficulties you overcame with the project.

- 2** If you review other groups’ projects through the Scratch studio, you should :
 - ▶ Open the teacher’s Scratch Final Project Studio;
 - ▶ Choose two projects to assess;
 - ▶ Fill out the Two Stars and a Wish Worksheet.

REFLECTION

Think about your project ... some questions to ponder.

- 1** Did you meet your own goals for your project?
- 2** If you could redo your project, would you change anything about how you designed or coded the project?



Scratch Final Project Design Worksheet

Name: _____

Class: _____ Date: _____

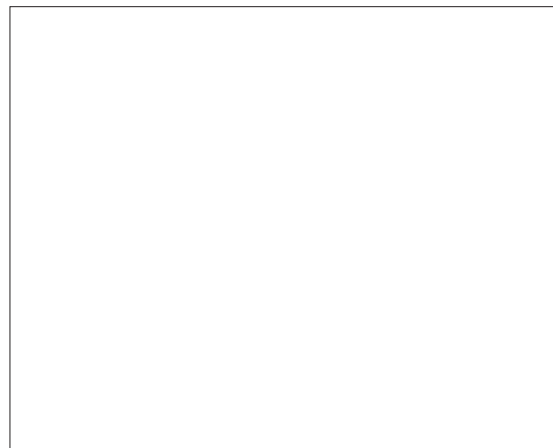
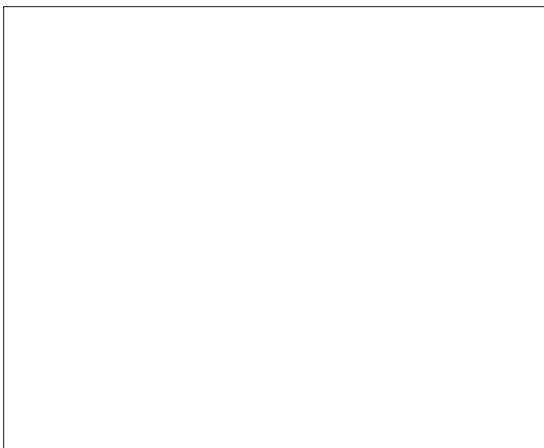
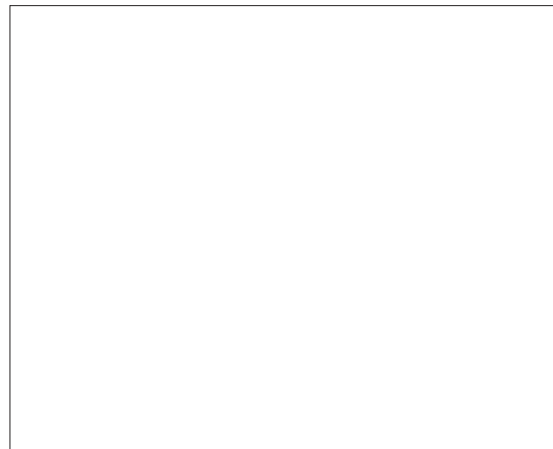
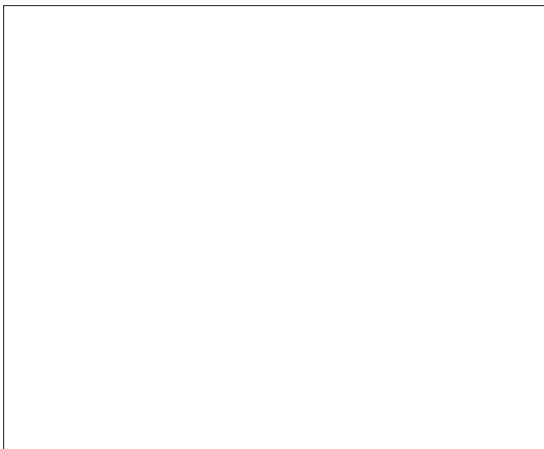
You can choose to make a Scratch story or a Scratch greeting card for your final project. Think about what kind of project you would like to make. Remember that you will have 8 lessons for this project.

Describe the project you want to create.

List the steps needed to create your project.

Scratch Final Project Design Worksheet

Storyboard: Draw pictures to represent the sequence of your story or animated card in the boxes below.



Copyright © The Hong Kong Jockey Club Charities Trust 2022

All rights reserved. The intellectual property subsisting in the material published is owned by The Hong Kong Jockey Club Charities Trust. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior written permission of the publisher.

Publisher: The Hong Kong Jockey Club Charities Trust

July 2022 Edition

ISBN: 978-988-74553-3-2

Address: 1 Sports Road, Happy Valley, Hong Kong

Website: <https://www.coolthink.hk>

Email: coolthink@hkjc.org.hk

Created and Funded by



The Hong Kong Jockey Club Charities Trust

Content Co-created by



香港教育大學

The Education University
of Hong Kong



Massachusetts
Institute of
Technology

Design and Produced by



MIT Hong Kong
Innovation Node

Disclaimer

All content in this book is intended for educational use and non-profit marking purpose only. The publisher shall not be held liable for any claims or losses arising from any use of the content. Websites and hyperlinks in the references are published or operated by independent organizations including but not limited to MIT Scratch, MIT App Inventor and YouTube. We make no warranties whatsoever and we shall not be deemed to have endorsed, recommended or approved any information, products or services contained in or offered through such sites.

{oo/Think @JC >
賽馬會運算思維教育
Inspiring digital creativity 啟發數碼創意

ISBN 978-9-88-745533-2



9 789887 455332 >