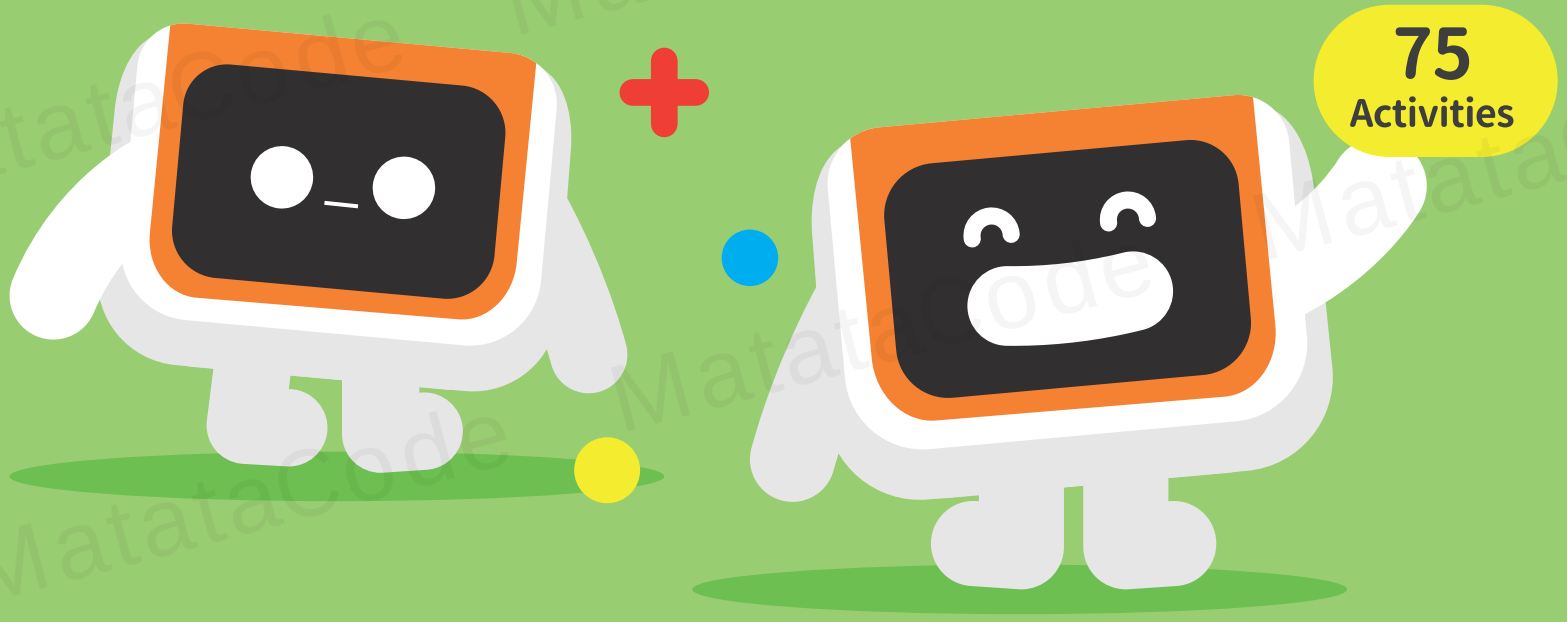
matatalab EDU

Core Activies

Graphical programming

Activity Cards for VinciBot



Watch the Videos

Core Activities Scope & Sequence

Note: Start from Leve A for all beginners no matter how old the students are. Pacing can be adjusted to how quickly your class move through the content.

A

15 activities

Sequences

Loops

- Repeat Forever
- Repeat X

B

15 activities

Events

- Basic Events
- Subroutine

Loops

- Repeat X
- Stacking Loops
- Nested Loops

C

15 activities

Conditionals

- wait until
- repeat until
- If then

Function

- Basic Function

D

15 activities

Conditionals

- If then
- If else

Variables

Function

- multiple function

E

15 activities

Conditionals

- Nested if else

Infrared Communication

Line Following

Number	Concept	Activity name
A-1	Sequence	Hello, VinciBot!
A-2	Sequence	Programming the VinciBot
A-3	Sequence	Nice to Meet You
A-4	Sequence	Information Transfer
A-5	Sequence	Six Facial Expressions
A-6	Sequence	Stone Lover
A-7	Sequence	The Palette
A-8	Sequence	VinciBot Got Lost on the Farm
A-9	Sequence	Hello, Animal Friends!
A-10	Loops (Repeat Forever)	The Rainbow Light
A-11	Loops (Repeat Forever)	The Beating Heart
A-12	Loops (Repeat Forever)	Wake Up, VinciBot!
A-13	Loops (Repeat Forever)	VinciBot is a Musician
A-14	Loops (Repeat Forever)	Guard VinciBot I
A-15	Loops (Repeat Forever)	Guard VinciBot II

A1 Hello, VinciBot!

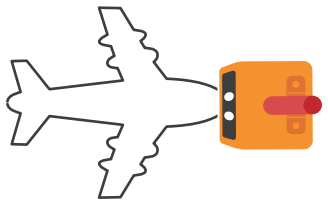
Sequences

Task：Get familiar with the structure, functions and characteristics of VinciBot by exploring the three preset modes of VinciBot.

1 Press to explore VinciBot's three preset modes: IR Remote Control Mode, Line Following Mode, and Drawing Mode.

Drawing Mode

In Drawing Mode, VinciBot draws a picture automatically.



IR Remote Control Mode

An IR remote control is included in the box with VinciBot. It can be used to change the speed and direction of the robot or adjust the volume, etc. Operate the robot on a smooth and flat playground.

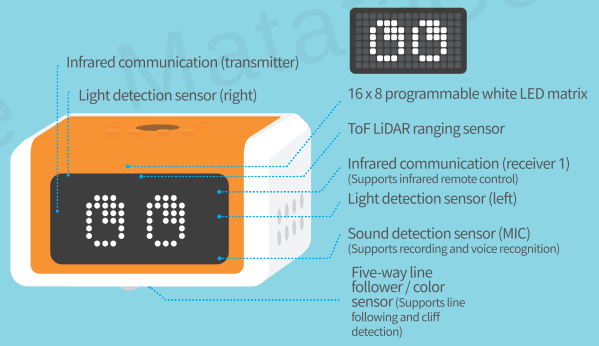
Line Following Mode

In Line Following Mode, VinciBot moves automatically along the black lines.

2 Explore the three preset modes of VinciBot, and choose its functions or characteristics.

- [ ] Sound
- [ ] Music
- [ ] Preset Dances
- [ ] Drawing
- [ ] Line Following
- [ ] LEDs Lights
- [ ] Dot-matrix screen that can display images, numbers and letters

Bonus:Observe the explosion diagram of VinciBot and guess what other functions and usage scenarios it has?



A2 Programming the VinciBot

Sequences

Task:Familiarize with VinciBot's programming platform and how to program VinciBot.

1 Open VinciBot's programming platform.

Website

https://coding.matatalab.com

Android App

https://play.google.com/store/apps/details?id=com.matatalab.vincibot

IOS APP

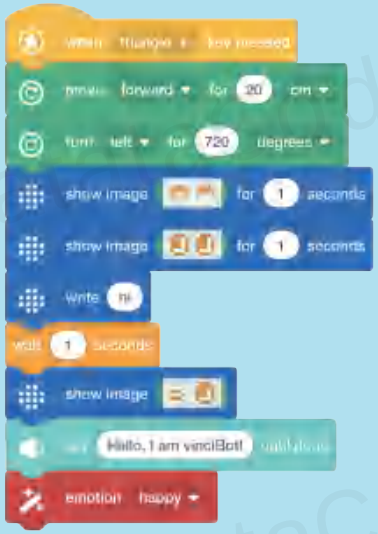
https://apps.apple.com/us/app/matatacode-vincibot/id1661920538

2 How to connect VinciBot and access its programming platform.

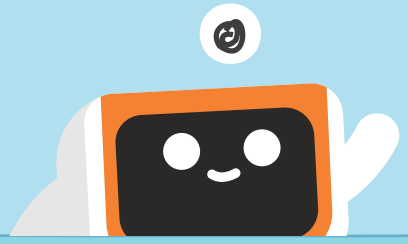
Connect via USB cable

Connect via Bluetooth

3 Referring to the demo program, drag the programming blocks from the list on the left to the programming area to write a program.



4 Run this program to view the results of VinciBot.



A3 Nice to Meet You

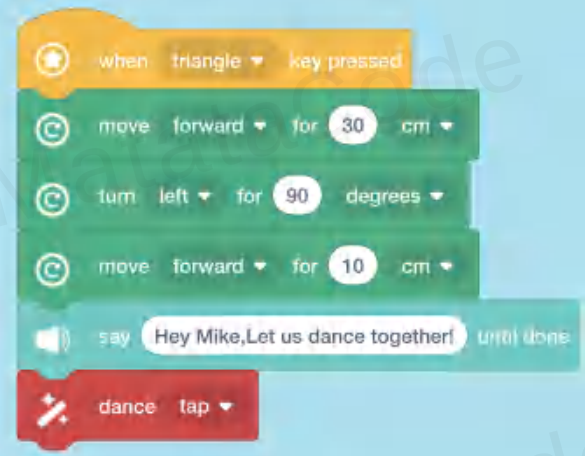
Sequences

Task :Familiarize with the motion, sound, and effect blocks; program VinciBot to walk up to a toy, say hello to it, and dance.

1 When writing a program, the first step is to choose an event block that starts the robot.

2 In order to make VinciBot "walk to the toy", "say hello", and "dance", the following coding blocks must be used.

3 Try to change the input parameters in the blocks and write a new program.



Bonus：Explore more motion, sound, and effect coding blocks, and write more fun programs for VinciBot.

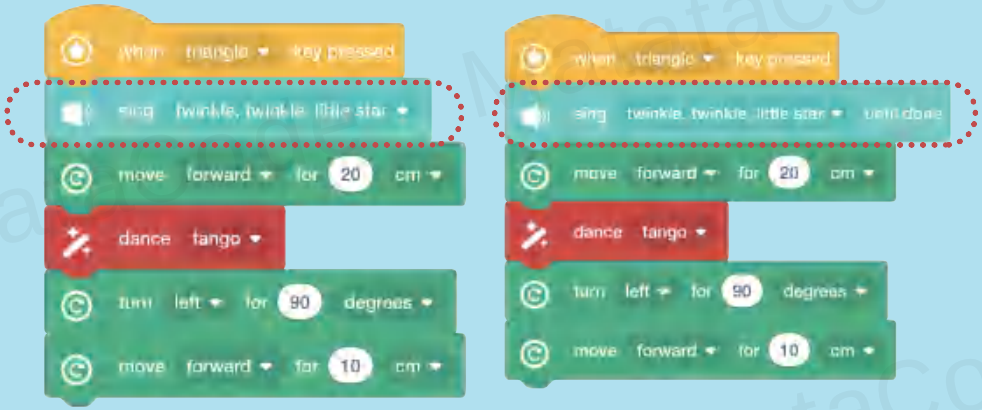
Motion Effect Sound

A4 Information Transfer

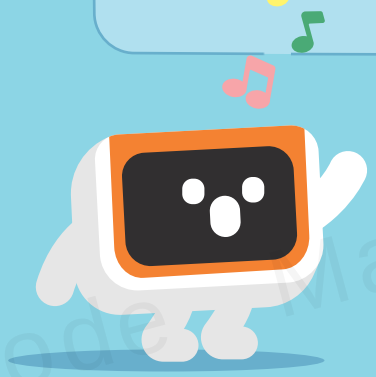
Sequences

Task：Familiarize with motion, sound, and light blocks. Program VinciBot is to walk up to a toy, sing a song, and then display the message "I Love You" on the dot-matrix screen.

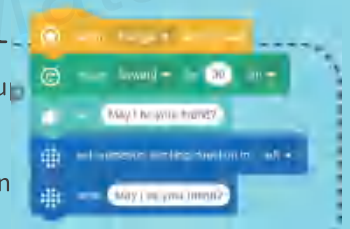
1 Test and consider the following: “What is the difference between these two programs?”



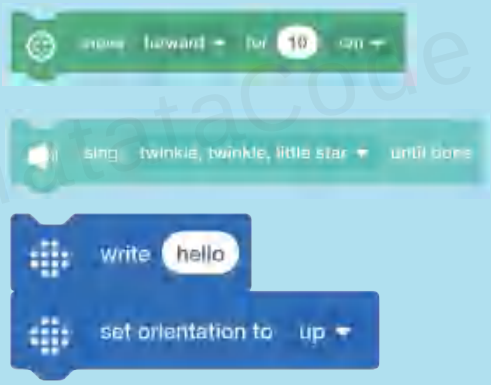
Knowledge points:
There are some similar blocks that come in pairs, the only difference being that there is is one block that uses an "until done" function at the end. This function ("until done") means that the instructions of this block will continue to run until completion before beginning the next set of instructions. When the "until done" function is not utilized, the instructions of this block will be executed at the same time as the next series of instructions. However, if the instructions of this initial block conflict with the next series of instructions, the instructions of the first block will be interrupted.



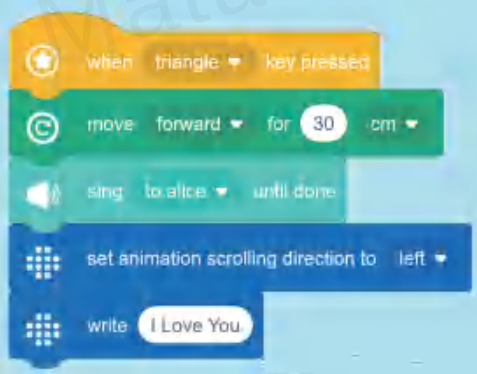
Bonus:Program VinciBot to walk up to a toy and say "May I be your friend?" while displaying that information on the screen.



2 In order to make VinciBot "walk to the toy", "sing a song", and finally display “I Love You” on the dot matrix screen, we need to use the following coding blocks.



3 The demo program.

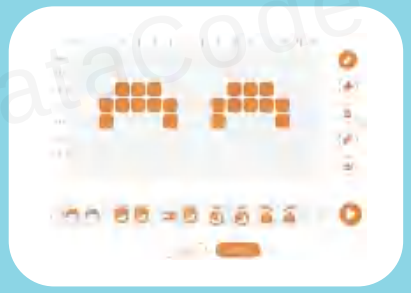


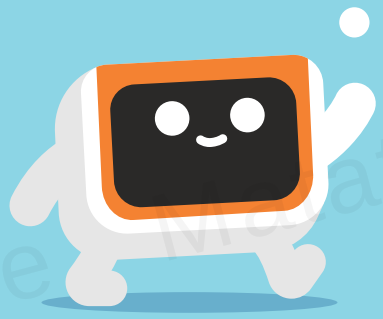
A5 Six Facial Expressions

Sequences

Task:Use the “show image” blocks in the light and the sound effect blocks; program VinciBot to display six distinct expressions.

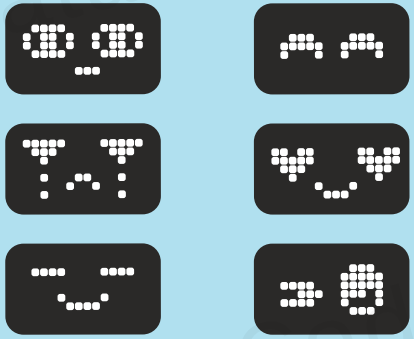
1 Get to know the “show image” coding blocks; explore the preset images and master how to set and store new images.





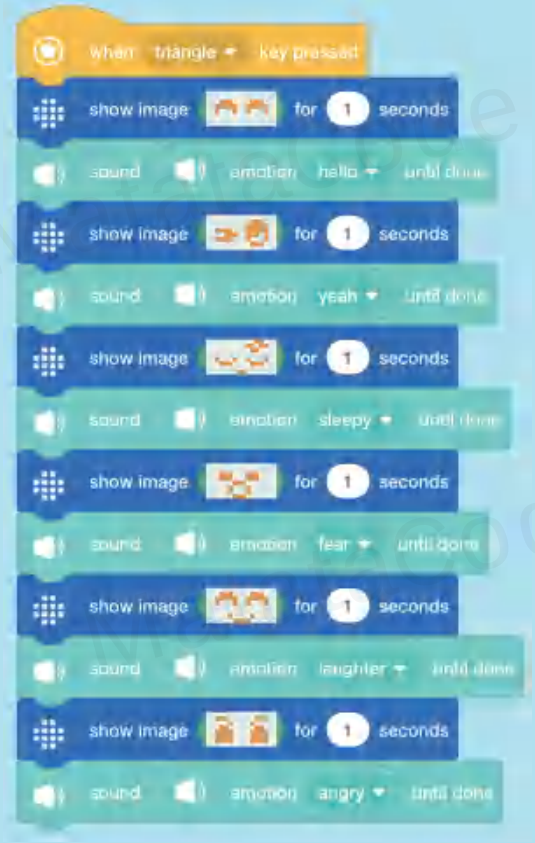
When developing the different expressions, which one of these two coding blocks do we need to choose?

2 Program VinciBot to make six different facial expressions in a row.



3 Add an interesting sound after each expression.

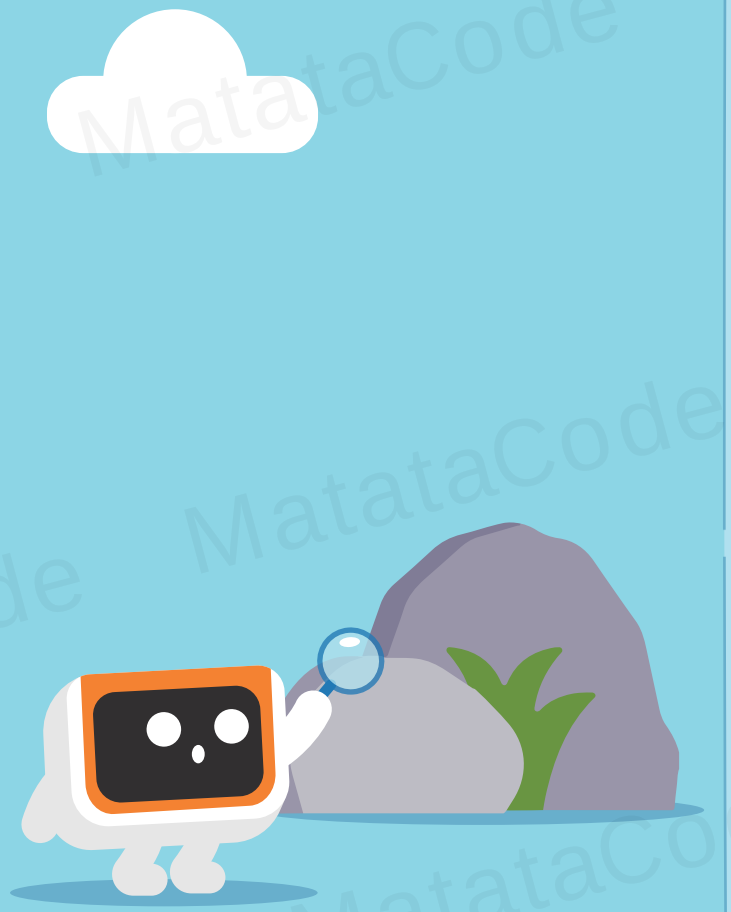
4 The demo program.



A6 Stone Lover

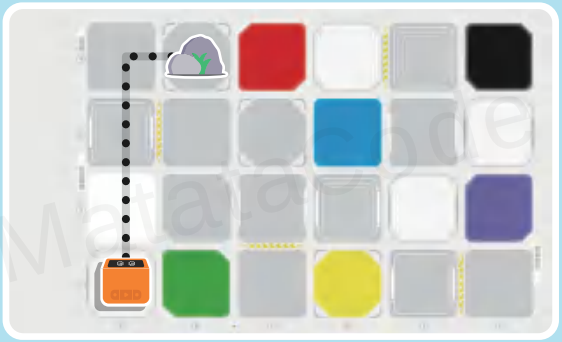
Sequences

Task: Set up the task scene on the map according to the illustrations. Program VinciBot to collect all the stone(s), and make a "score" sound every time it picks up a stone.



1 Set up the stone cards to correspond to the map, as shown below.

2 Program VinciBot to collect all the stone(s), and make a "score" sound every time it picks up a stone.



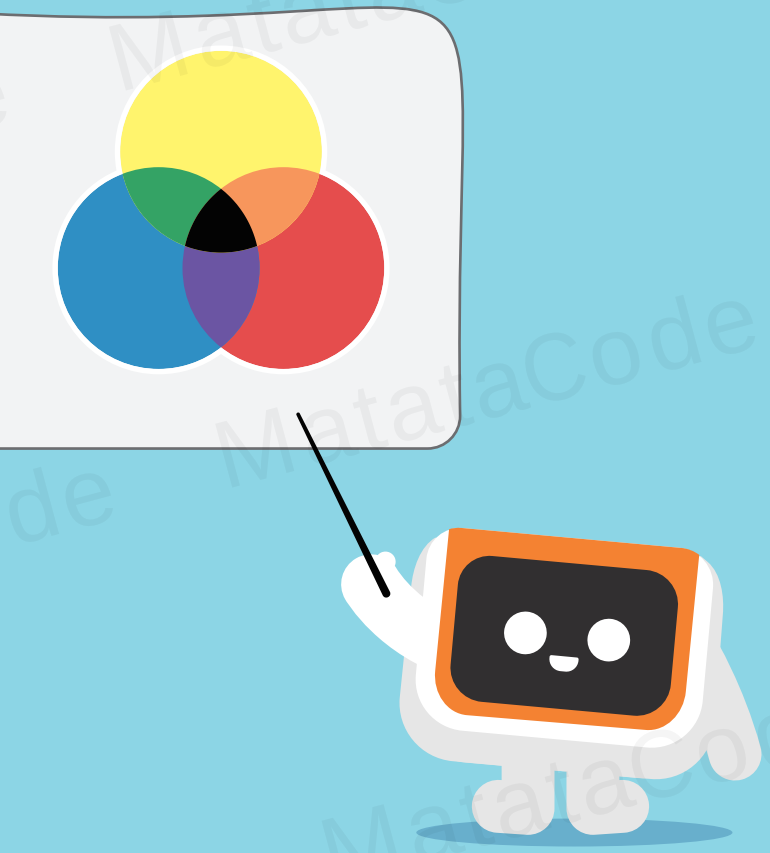
Bonus:Prepare more complex tasks to Vincibot' s functionality.

A7 The Palette

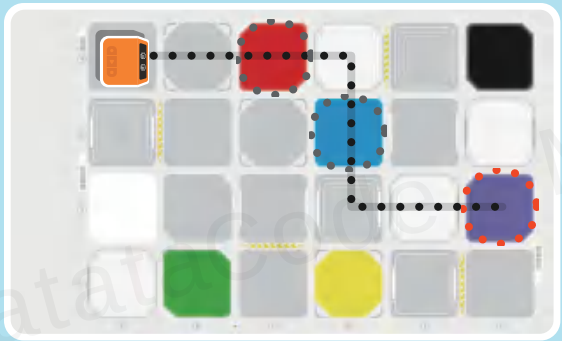
Sequences

Task:Understand the pigments of the three primary colors and related principles, and program VinciBot to "modulate" the color purple, green, and black.

1 Learn about the pigments of the three primary colors (CMYK): red, yellow, blue, and the colors that result when the three interact.



2 Program VinciBot to "modulate" the color purple.



3 In the same way, let VinciBot "modulate" green and black.

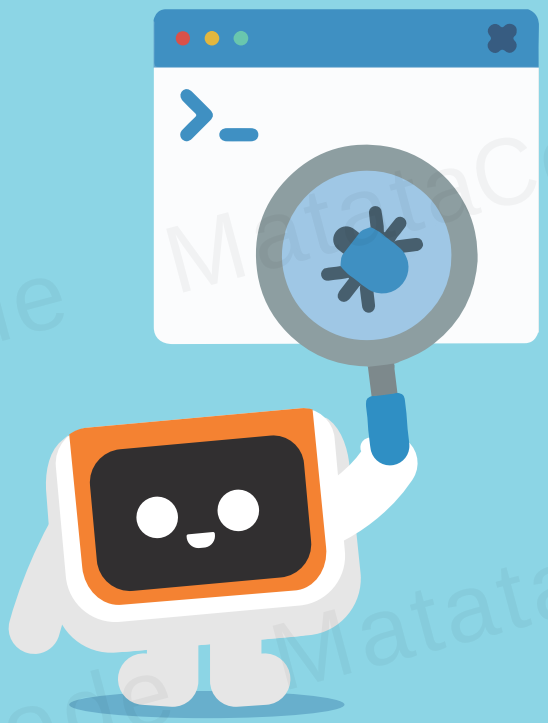
A8 VinciBot Got Lost on the Farm

Task: Understand the concept of “bug” and “debugging.” There are five programs let VinciBot visit the farm animals. Through practical operation, find the bugs in the programs and debug them.

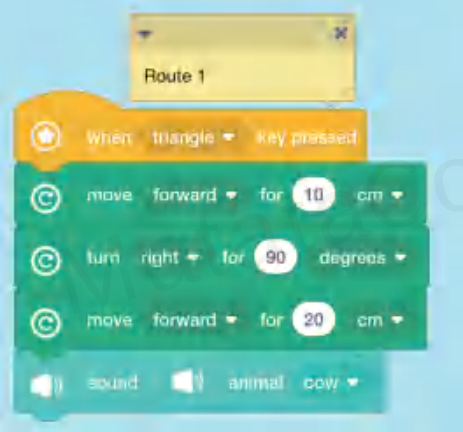
Sequences

1 “Bug” is mostly used to refer to errors in programs. If there are bugs, the program cannot run successfully or achieve the desired effect. The process of program repair is called “debugging.”

BUG=



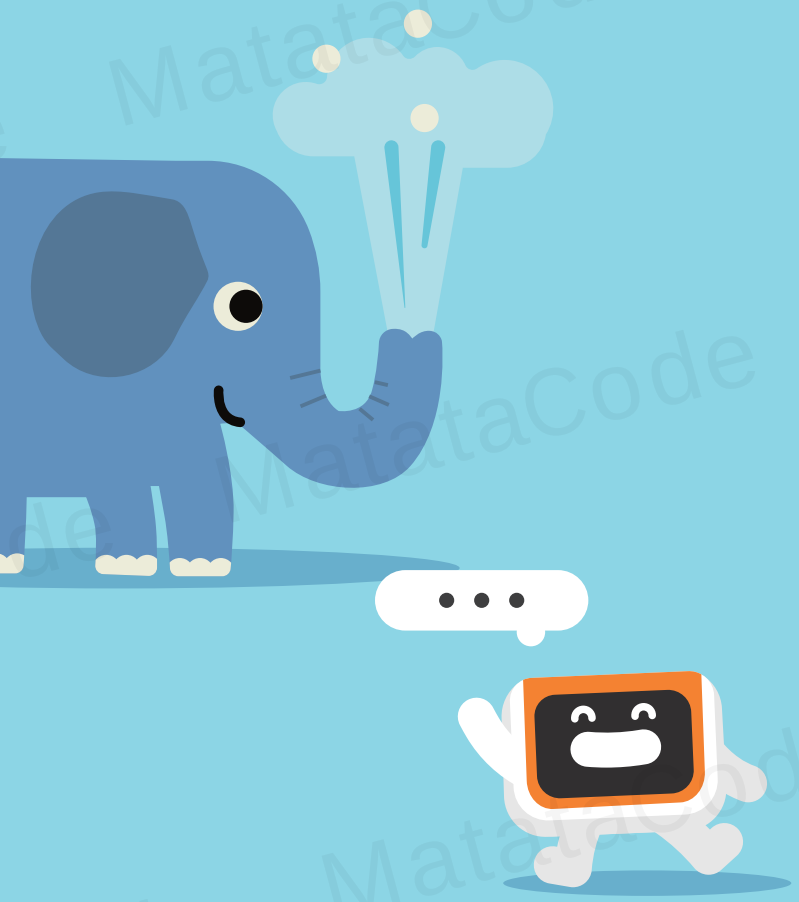
2 VinciBot is visiting the farm. Please observe the routes and the corresponding programs, then identify the bugs in the programs and debug them.



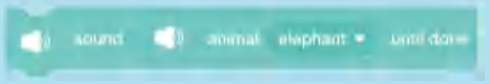
A9 Hello, Animal Friends!

Sequences

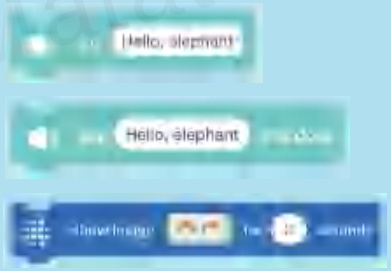
Tsak: VinciBot goes to visit the zoo; every time it walks up to an animal, it will imitate the animal's sound, say hello, and make various funny expressions to make the animal happy.



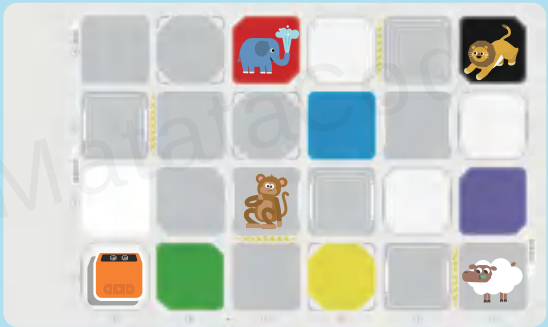
1 VinciBot goes to visit the zoo. Every time VinciBot encounters an animal, it will imitate the animal's sound.



2 After imitating the animal' s sound, VinciBot greets it and makes funny expressions to make the animal happy.



3 Set up the animal cards on the map as shown below. Then program VinciBot to visit all the animals on the map. (An example is shown below.)

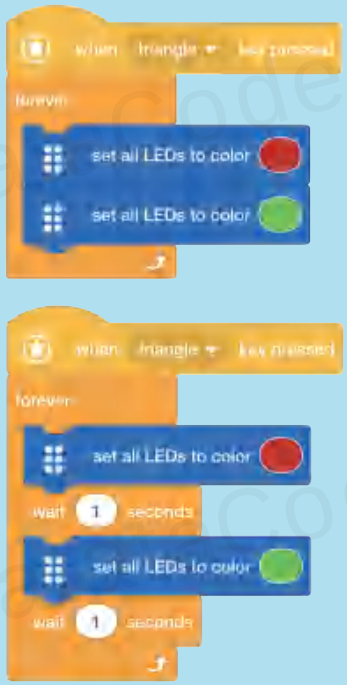


A10 The Rainbow Light

Loops
(Repeat Forever)

Task: Familiarize with the LED light coding blocks; use the LED light coding blocks and repeat coding blocks to create a beautiful rainbow light.

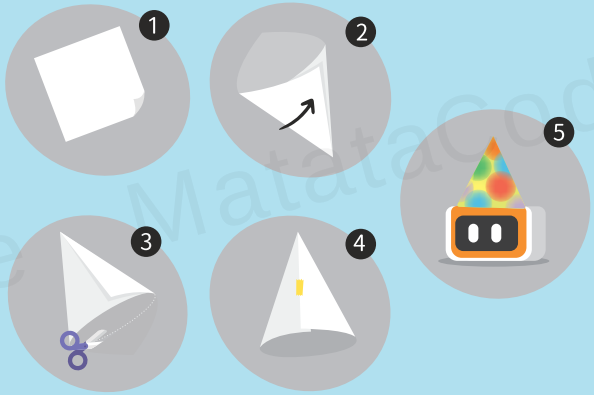
1 What is the difference between these two programs?



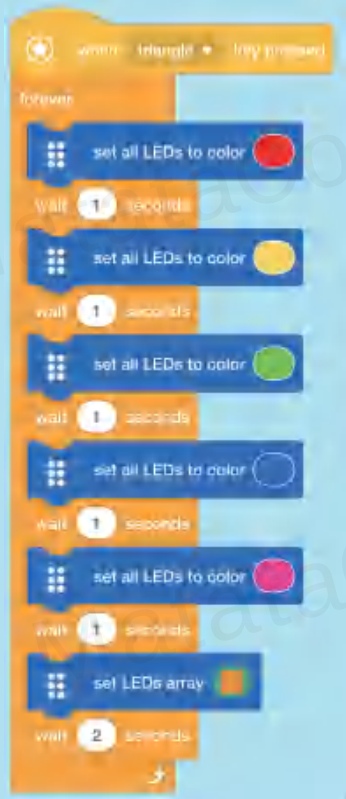
Knowledge points:
When several "state" blocks of the same category are used continuously before and after, the previous state will end instantly, and only the last state will appear. In order to ensure that each state can be displayed, it is necessary to use the wait coding block.

2 Explore the usage of various LED light coding blocks and try editing the color of each LED light.

3 Use paper to make a lampshade for VinciBot and put it over the LED lights.



4 Use repeat coding block to program and transform VinciBot into a rainbow light.

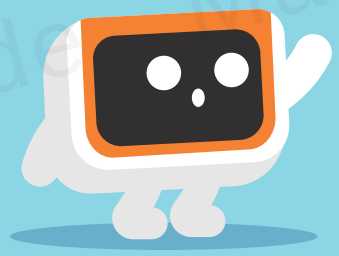


A11 The Beating Heart

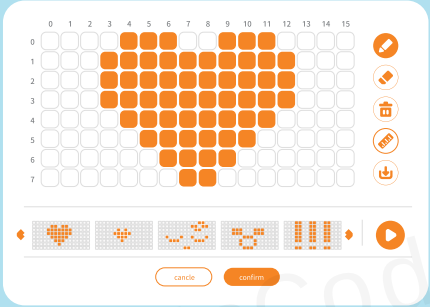
Task: Understand the principle of generating animation. Use the “show image” coding blocks and “repeat” coding block to display the beating heart on VinciBot's dot matrix screen.

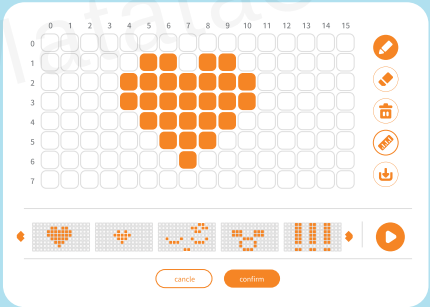
Loops
(Repeat Forever)

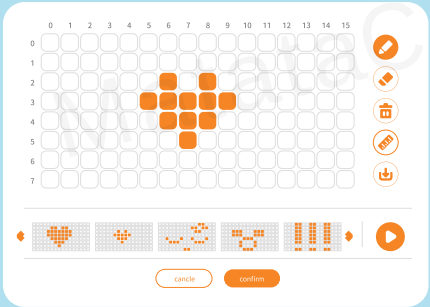
1 Animation is the effect formed by the rapid playback of continuous related images.



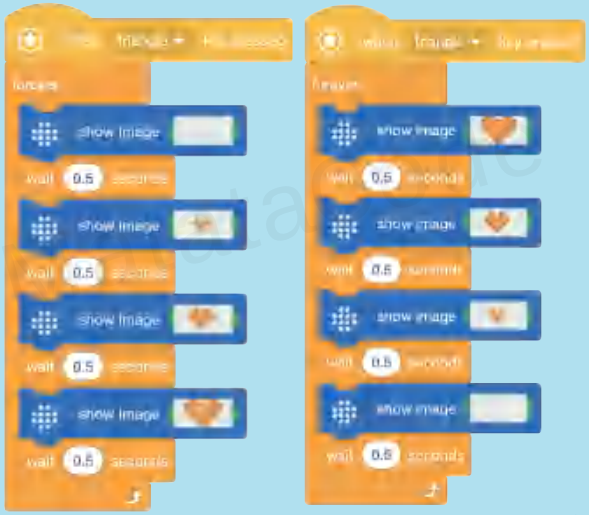
2 Let's make an animation of a beating heart. First, edit three hearts from big to small on the edit page of the show image panel.







3 Add the repeat coding block to make the heart keep beating!



Bonus:Make an animation of beating number 2.

A12 Wake Up, VinciBot!

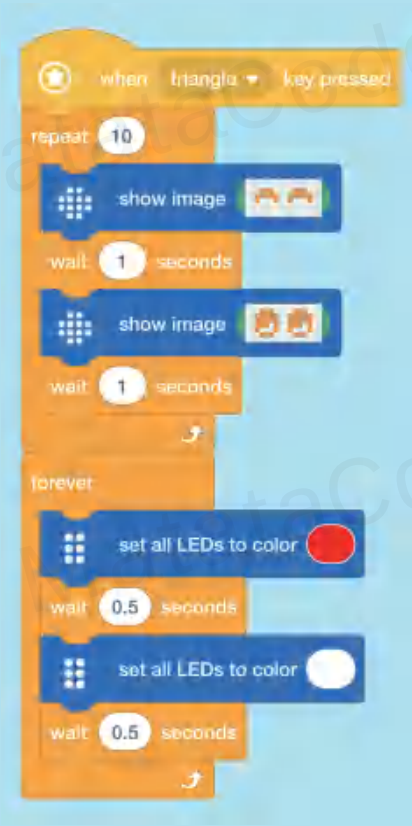
Loops
(Repeat Forever)

Task: The Guard VinciBot is sleepy; its eyes blink continuously. In order to wake himself up, Guard VinciBot turns on its red and white LED lights.

1 The Guard VinciBot is sleepy; its eyes blink continuously. Consider two possible ways to achieve this blinking effect How many times do you want its eyes to blink?

2 In order to wake up, Guard VinciBot lights up the LED lights that flash red and white alternately. How many times do you want its lights to flicker?

3 Combine the two parts and make Guard VinciBot complete both actions simultaneously.



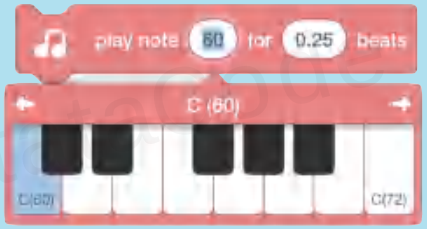
Knowledge points:
When several "state" blocks of the same category are used continuously before and after, the previous state will end instantly, and only the last state will appear. In order to ensure that each state can be displayed, it is necessary to use the wait coding block.

A13 VinciBot is a Musician

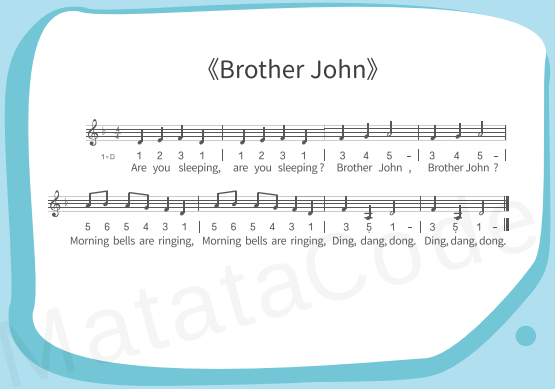
Loops
(Repeat Forever)

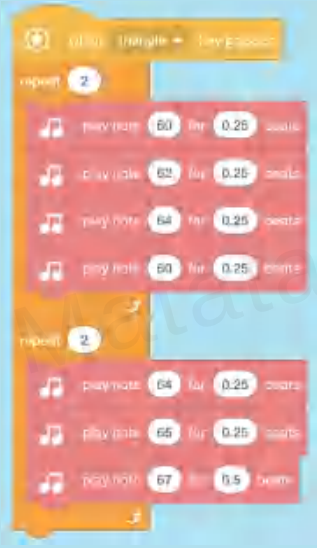
Task: Explore the “music” coding blocks, and program VinciBot to play "Are you Sleepy?" and other songs with different instrument sounds.

1 This music block can be used to set the pitch and duration.

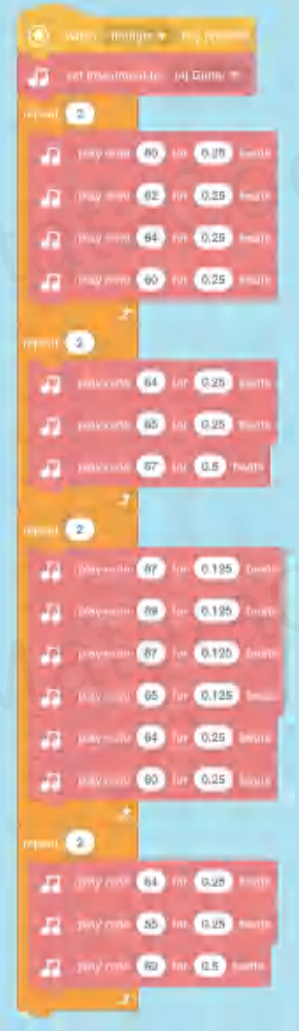


2 Based on the musical score for "Are You Sleepy?", write a music program and attempt to simplify it by using loops.





3 Play "Are You Sleepy?" (You can set up the instrument before playing.)



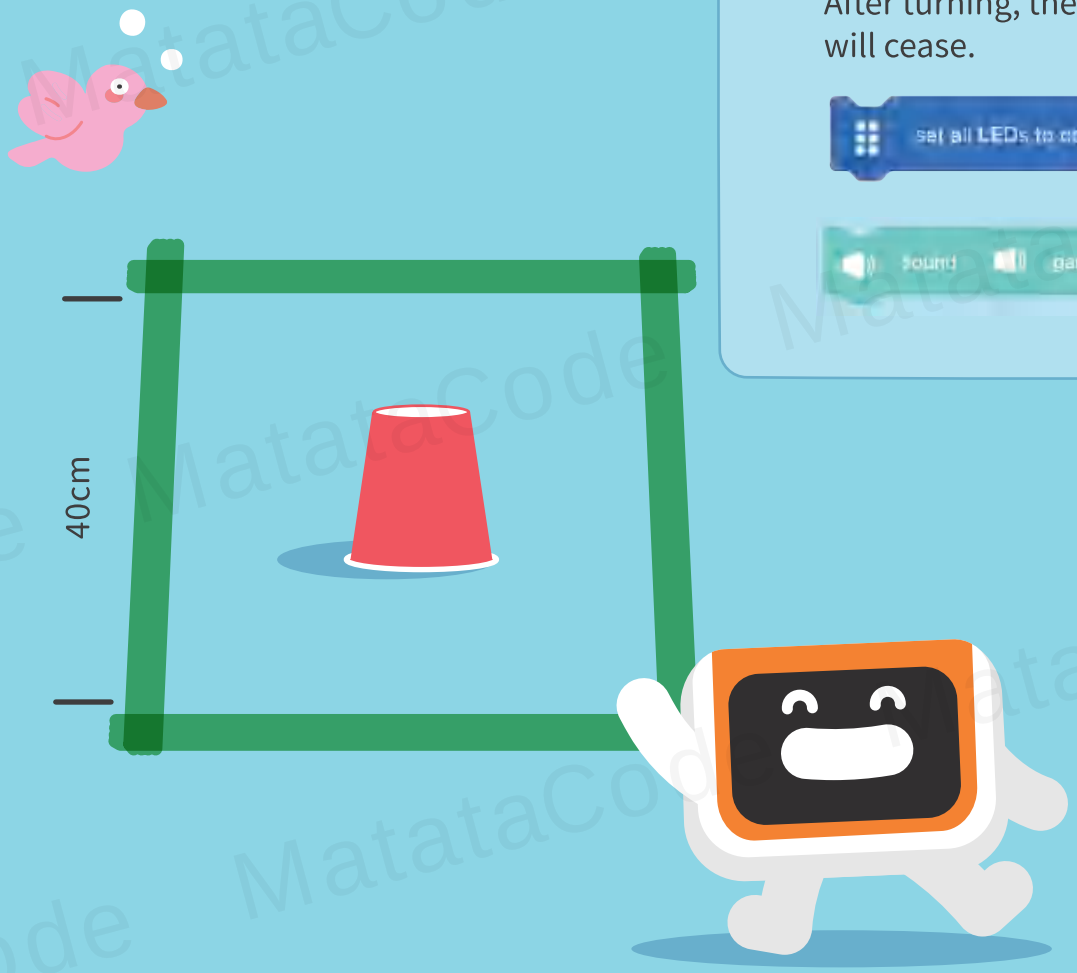
Bonus: Find additional music scores and program VinciBot to play them!

A14 Guard VinciBot I

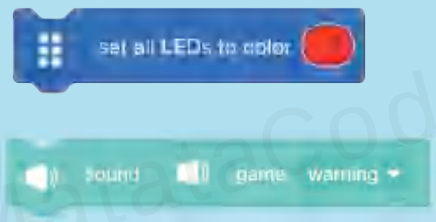
Loops
(Repeat Forever)

Task: The farm employs Guard VinciBot to protect the barn, especially to scare away the birds that steal the rice. Guard VinciBot should patrol around the barn.

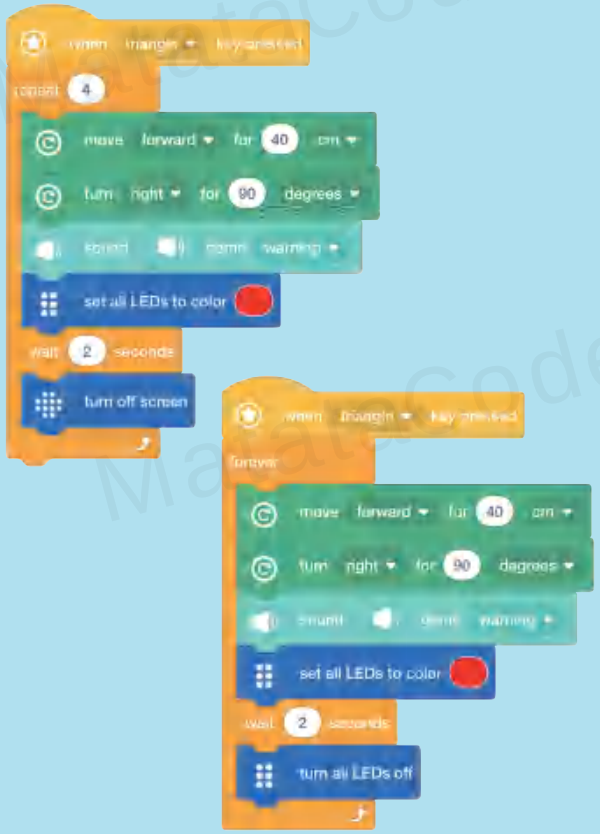
1 Set up the task scene: Put the paper cup (barn) on the ground, and create a 40cm x 40cm patrol route around it.



2 Program Guard VinciBot to run along the patrol route. Every time Guard VinciBot turns, it will turn on its red lights and make an alert sound to scare the birds away. After turning, the lights and sound will cease.



3 What is the loop program that allows Guard VinciBot to finish one round of the patrol route? What is the loop program that allows Guard VinciBot to keep patrolling around the barn?

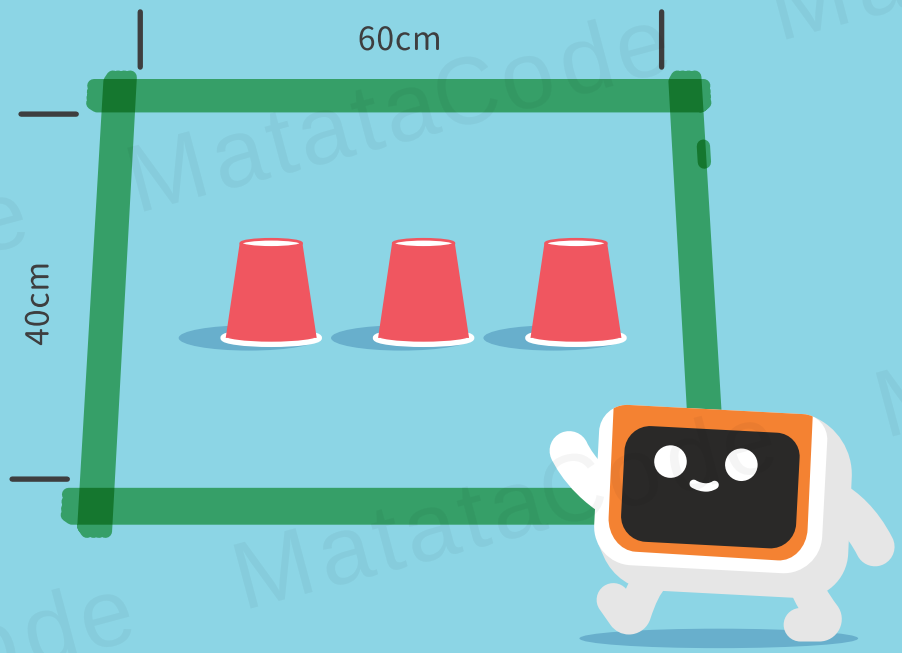


A15 Guard VinciBot II

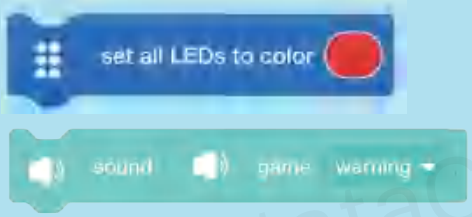
Loops
(Repeat Forever)

Task: The task of Protecting the barn has been completed. Now, Guard VinciBot is invited to continue protecting the rectangular sheepfold.

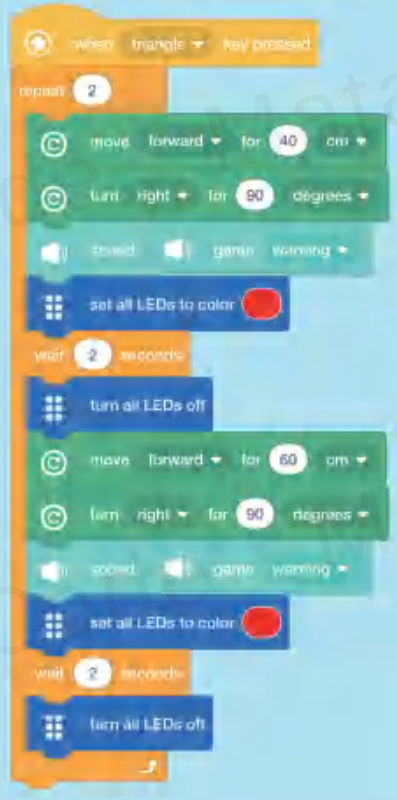
1 Set up the task scene: Put three paper cups (sheep pens) on the flat ground, and stick a 40cm x 60cm patrol route around them with tape.



2 Write a loop program to make Guard VinciBot run along the patrol route. Every time Guard VinciBot turns, it will turn on its red lights and emit an alert sound to scare away the wolves.



3 What program is required for Guard VinciBot to complete a circle around the patrol route? What program is required to keep Guard VinciBot running around the patrol route?



Number	Concept	Activity name
B-1	Events	VinciBot is Pretending to be Sleepy
B-2	Events	Ready, Go!
B-3	Events	Drive the Duck
B-4	Events	Invisible Ruler
B-5	Events	Voice Control Light
B-6	Loops（Repeat X）	VinciBot Loves to Draw I
B-7	Loops（Repeat X）	Candy Collecting
B-8	Loops（Repeat X）	Protect the Marine Environment
B-9	Loops（Stacking Loops）	Sweeping VinciBot
B-10	Loops（Stacking Loops）	ViciBot Loves to Draw II
B-11	Loops（Stacking Loops）	The String Flags I
B-12	Loops（Stacking Loops）	The String Flags II
B-13	Events（Multithreading）	VinciBot’s New Dance
B-14	Events（Multithreading）	VinciBot is a Superstar!
B-15	Events（Multithreading）	Variety Lollipops

B1 VinciBot is Pretending to be Sleepy

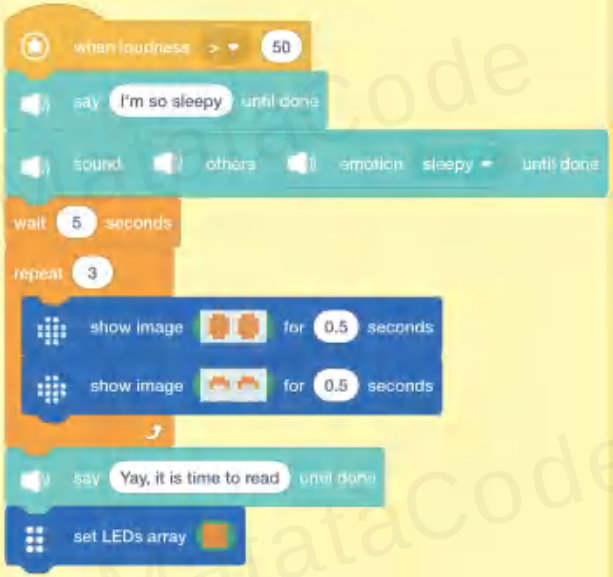
Events

Task：Familiarize with the “new event” coding blocks. When VinciBot hears someone coming (i.e. making a sound), it will pretend to be sleepy, say "so sleepy", and make a "sleepy" sound. After waiting 5 seconds to confirm that the person has walked away, VinciBot will blink and light up to read a book.

1 This event coding block judges the loudness of a sound, and allows VinciBot to start the next action after hearing a sound of a certain loudness.

2 After hearing the sound, VinciBot says "so sleepy" and then makes a “sleepy” sound.

3 After waiting 5 seconds, VinciBot will blink and light up to read a book.



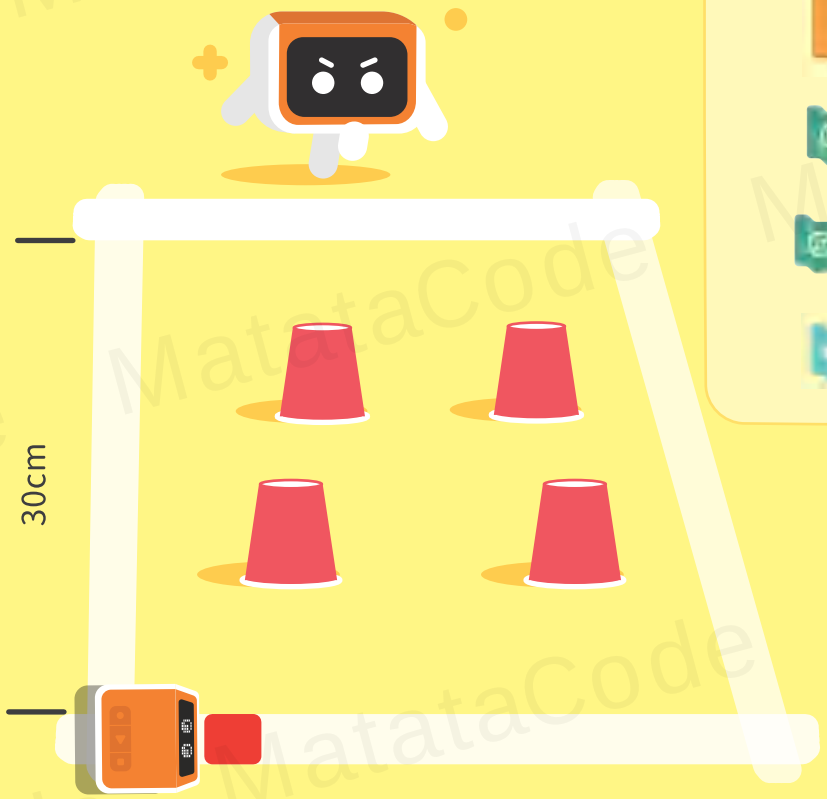
Bonus:Consider what someone does when pretending to be asleep? Program VinciBot to simulate a series of actions consistent with someone pretending to be asleep.

B2 Ready, Go!

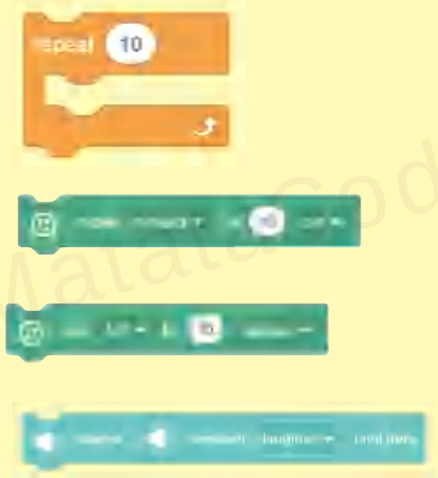
Task : Familiarize with the new event coding blocks. When VinciBot detects the red starting point, it starts to run three laps around the four cups. After the run ends, it laughs happily.

Events

1 Set up the task scene:Place four paper cups on a level surface or table, and create (with tape) or draw (with erasable pens) a square or rectangular running route around them. Then set a red starting point at one corner of the route.



2 Write a loop program that makes VinciBot run three laps around the running route from the starting point, and add a "laugh" sound at the end of the loop program.



3 Add a "color detected" event coding block at the start of the program.

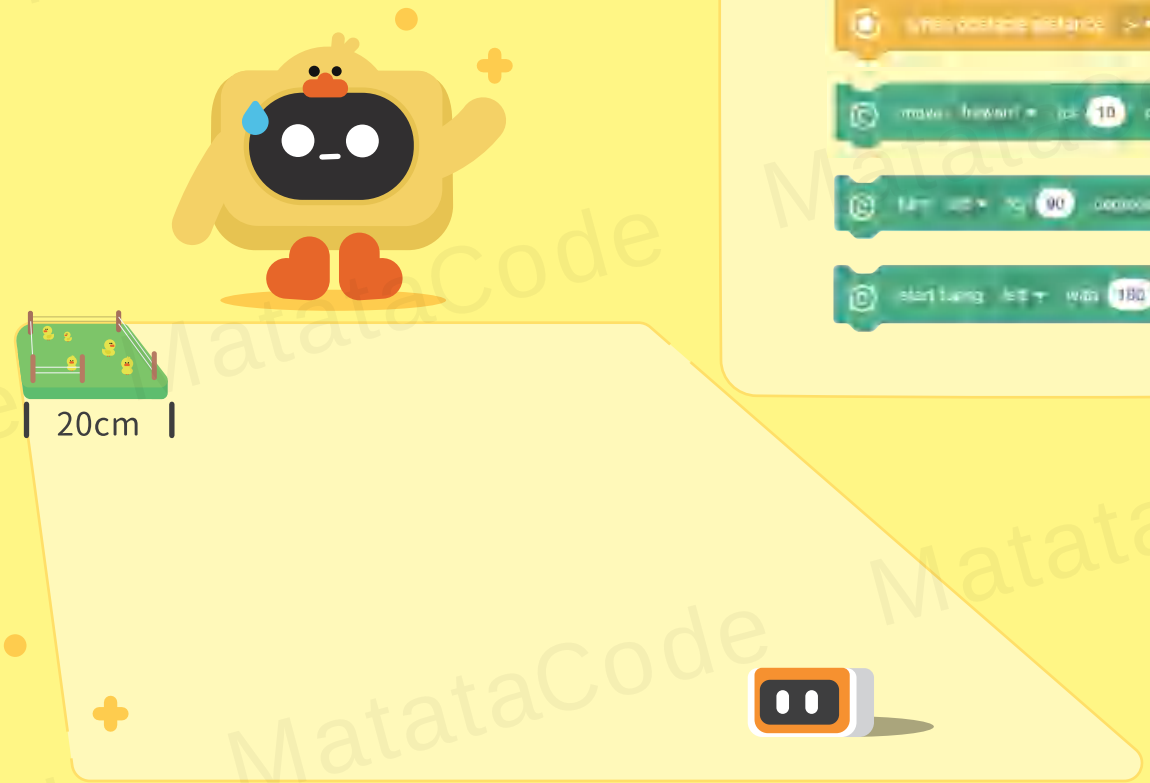
4 After importing the entire program into VinciBot, place VinciBot at the red starting point and observe the results of VinciBot as it runs.

B3 Drive the Duck

Events

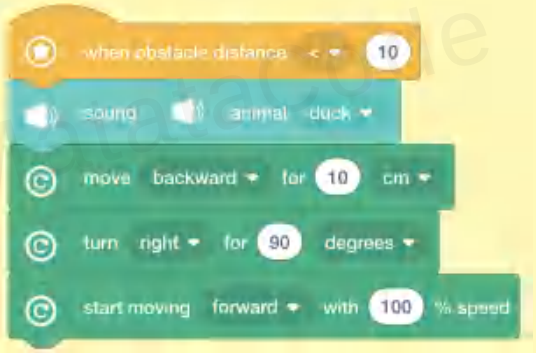
Task : VinciBot acts as a duckling that quacks and turns to the right and continues walking whenever it detects an obstacle ahead. Use the event block that detects the distance of obstacles to drive it to the duck house.

1. Set up the task scene: Lay out a large flat surface or desktop, and draw a 20cmx20cm duck house in the lower right corner of this area.



2. Write a program using the event coding block that detect the distance of obstacles. Every time the VinciBot duckling detects an obstacle ahead, it will quack, turn right, and then continue to move forward.

3. Run the program and try to drive the VinciBot duckling into the duck house by hand.



B4 Invisible Ruler

Task: Apply VinciBot's ToF ranging sensor to measure the length or height of an item; display the measured distance on the dot matrix screen and program VinciBot to verbalize the distance.

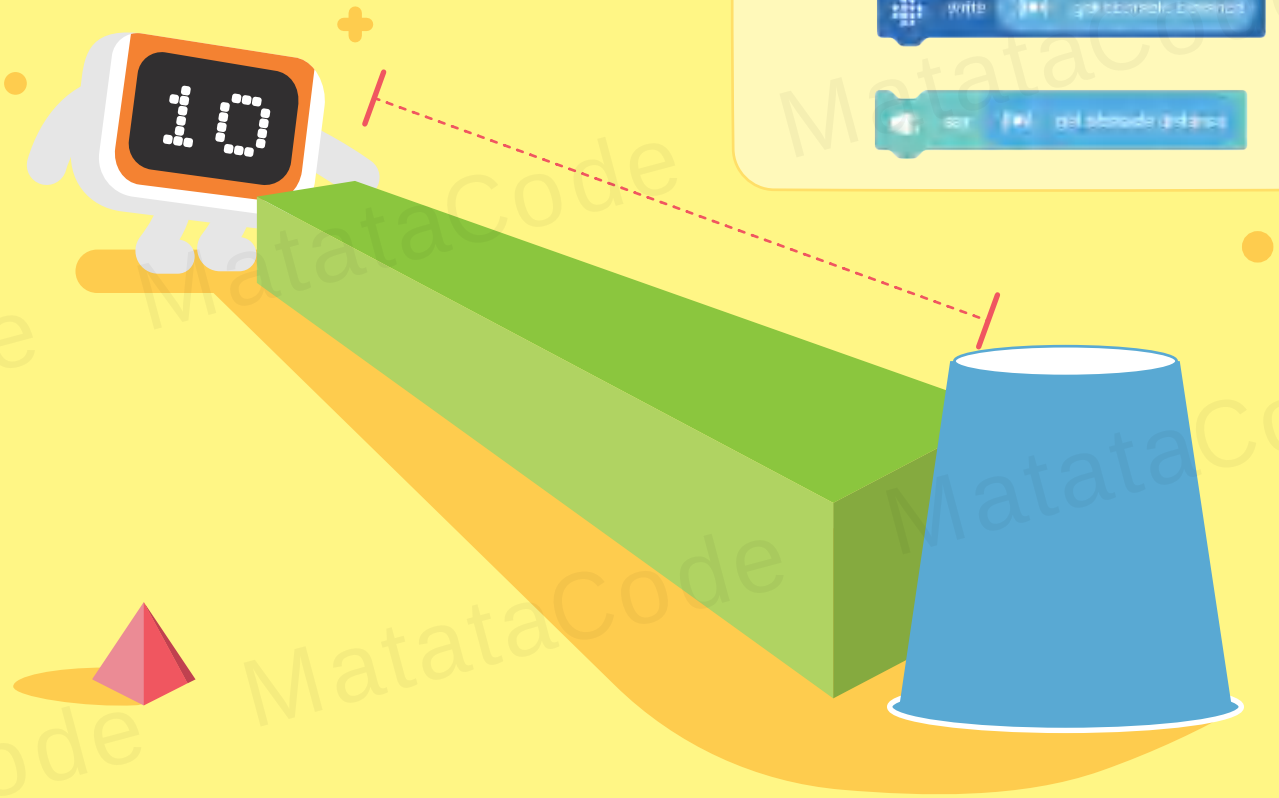
Events

1. Choose an item to measure, such as the length of a box or the height of a table. After ensuring that there is an obstacle at the end of the span to be measured, place Vincibot at the starting point.

2. The following coding block will be used to measure the distance.

3. Write a program to make VinciBot display and verbalize the measured distance.

4. When running the program, point the VinciBot towards the end point, and start measuring according to the event block that has been selected.

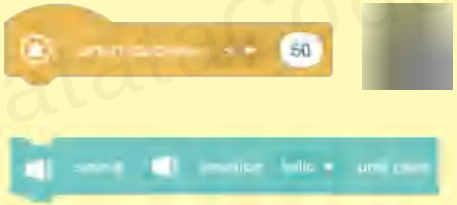


B5 Voice Control Light

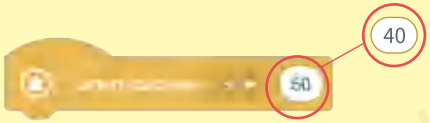
Task :Apply the sound sensor to make VinciBot turn on its light and say "hi" when it hears a loud sound, and to automatically turn off after a period of time and say "bye".

事件 Events

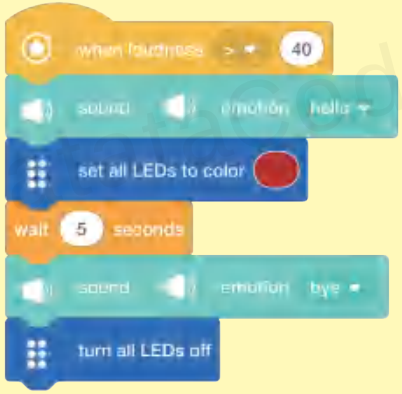
1 Use the event coding block that detects sound intensity, the LED light blocks, etc. to program VinciBot to light up and say "hi" after hearing a sound.

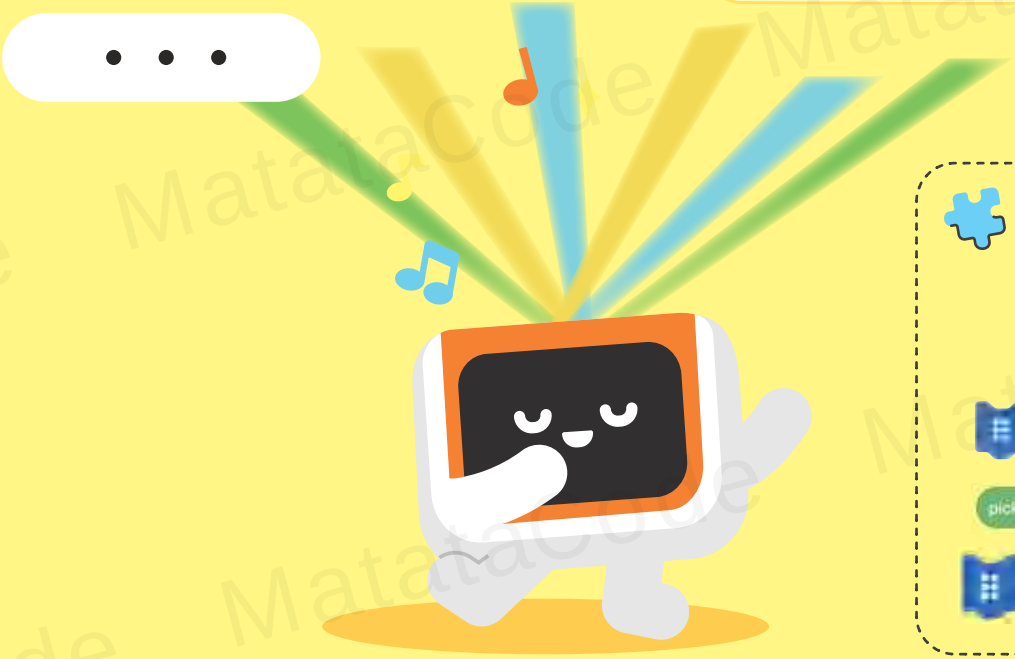


2 Modify the volume parameters to ensure that a clapping sound is sufficiently loud to wake up VinciBot (and turn on its LED lights), while ensuring sounds at low volumes will not accidentally turn on the lights.

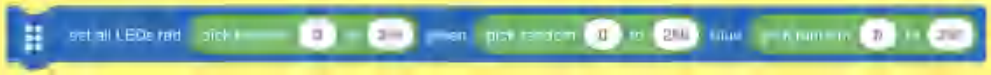


3 Set a wait time for VinciBot. After the wait time, program VinciBot to automatically turn off its lights and say "bye".





Bonus:How can LED lights of different colors be randomly chosen to light up every time a sound is heard? Use the blue blocks below to set the RGB (red, green, and blue primary colors of light) values; the value of each primary color is between 0 and 255. The three primary colors can be randomly combined into all colors. Use the green blocks below to achieve random values in the range of 0-255.

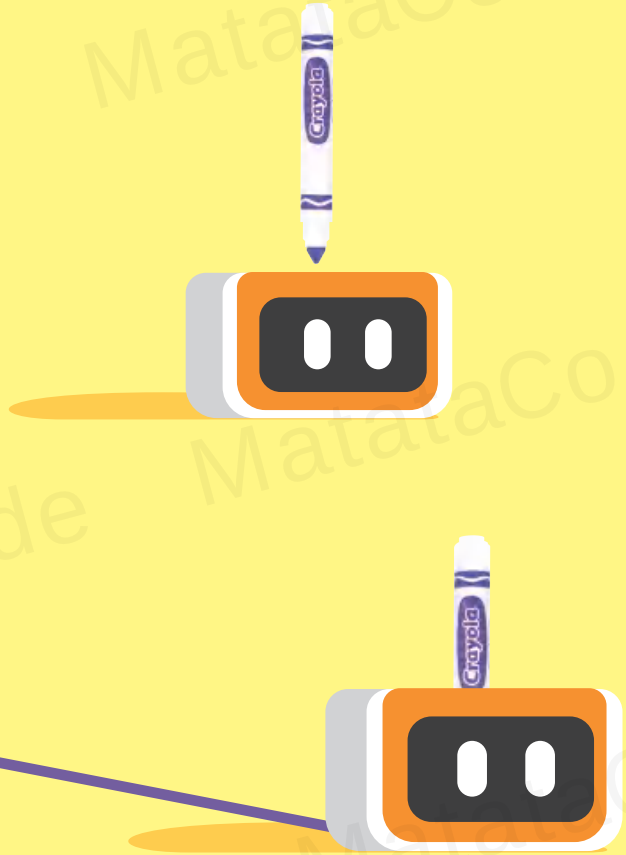


B6 VinciBot Loves to Draw I

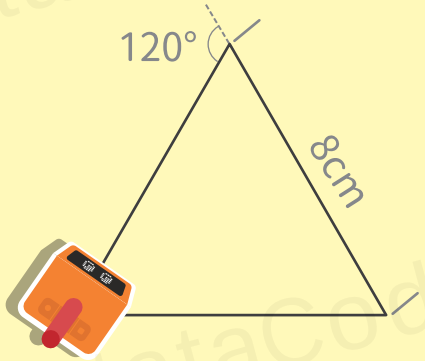
Loops (Repeat X)

Task: Learn how to draw with VinciBot, and program VinciBot to draw simple shapes.Insert the washable marker into the hole in the middle of VinciBot, and use the motion blocks to draw simple shapes.

1 Learn how to draw with VinciBot, and program VinciBot to draw simple shapes.Insert the washable marker into the hole in the middle of VinciBot, and use the motion blocks to draw simple shapes.



2 Let's take a look at the program to draw a triangle.



3 Try to draw more shapes below.

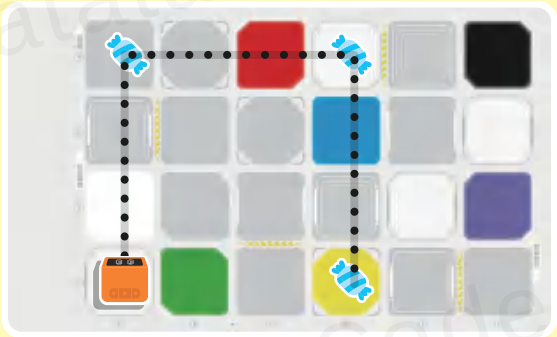
Bonus: Every time VinciBot finishes drawing a shape, it will say "I had a great time today" and blink twice.

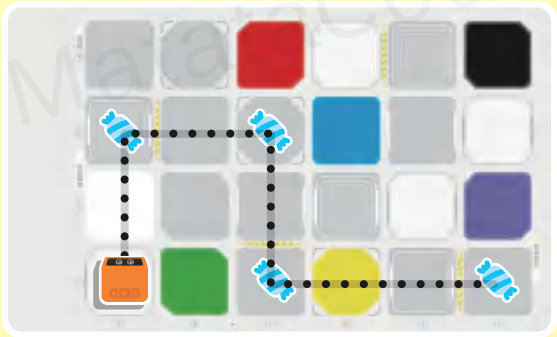
B7 Candy Collecting

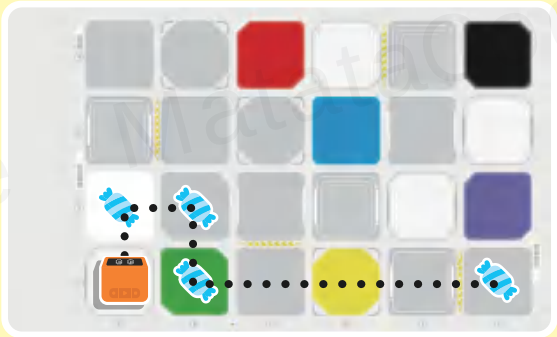
Task: Set up the task scene on the map according to the illustrations. Write a loop program to have VinciBot begin at the starting point to collect all candies on the map, and each time VinciBot reached a candy, the "score" sound will be played.

Loops
(Repeat X)

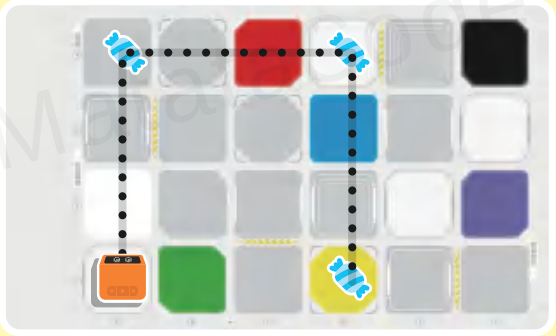
1 Use the candies/candy cards to set up the map as shown below; use a pencil to draw the route.

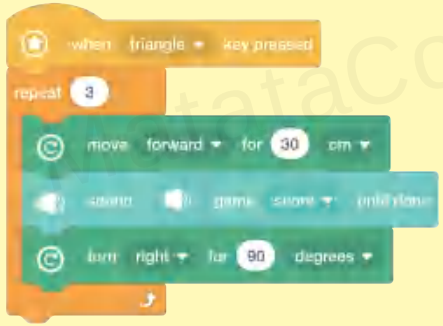






2 Observe the task map, and determine how to write a loop program to make VinciBot collect all the candies; a "score" sound will play each time a candy is collected (One example is shown below).





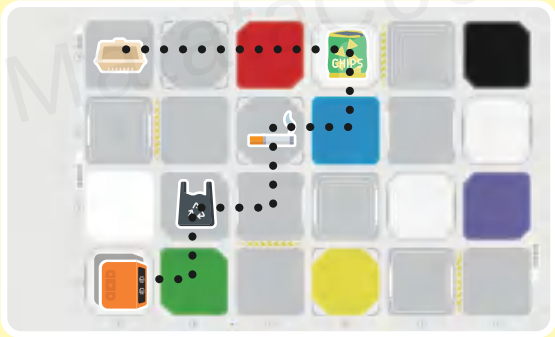
Bonus: What do all the maps and routes in today's programming exercise have in common?

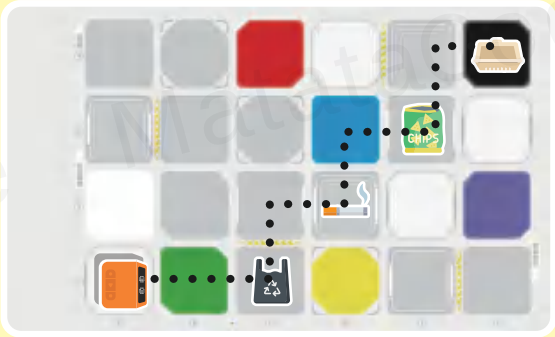
B8 Protect the Marine Environment

Loops
(Repeat X)

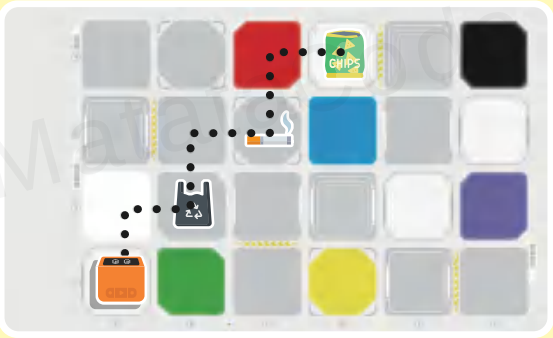
Task :Set up the task scene on the map according to the illustrations. Write a loop program to have VinciBot begin at the starting point to collect all marine litter on the map; each time VinciBot reaches litter, an "alert" sound will be played.

1 Use the marine litter cards to set up the map as shown below; use a pencil to draw the route.





2 Observe the task map, and determine how to write a loop program to make VinciBot collect all the litter; an "alert" sound will play each time it collects litter (One example is shown below).



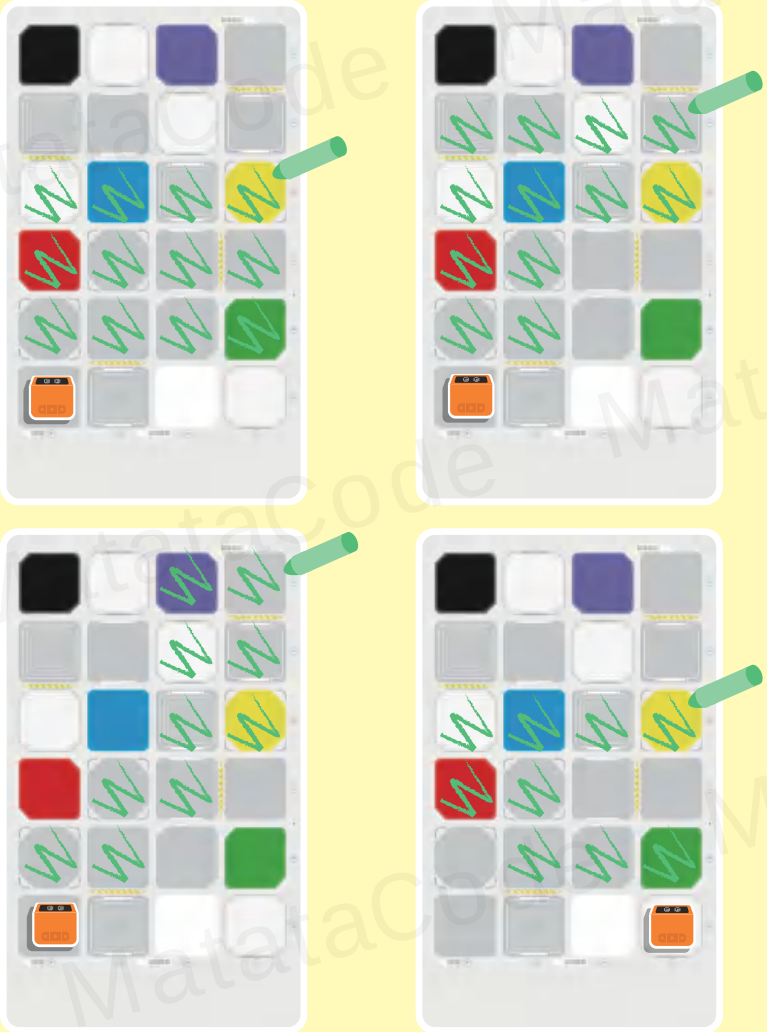
Bonus：What do all the maps and routes in today's programming exercise have in common?

B9 Sweeping VinciBot

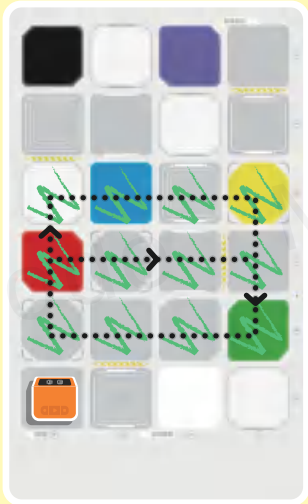
Loops
(Nested Loops)

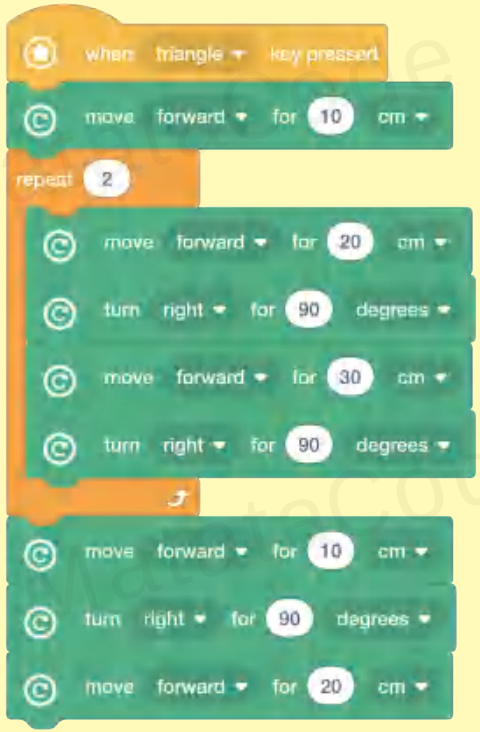
Task: According to the illustrations, use a pencil to mark the range that is to be cleaned by Sweeping VinciBot on the map. Then plan the route and write the loop program to have Sweeping Vincibot complete the cleaning tasks.

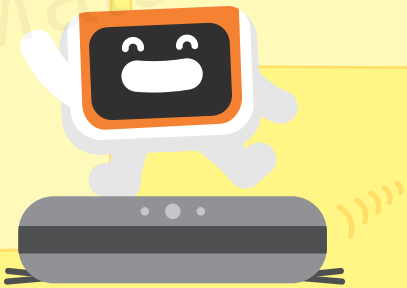
1 Use a pencil to mark the area to be cleaned by Sweeping VinciBot on the map as shown below. Note: Sweeping VinciBot can only operate within the cleaning range.



2 Consider how to plan the route, and then write a loop program to make Sweeping VinciBot clean every corner of the cleaning range (An example is shown below).







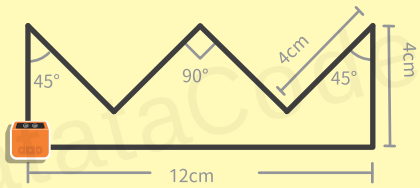
Bonus: Design and complete more challenge tasks!

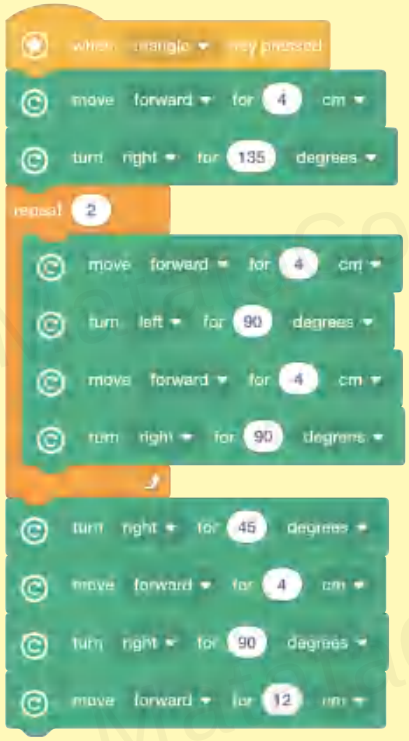
B10 ViciBot Loves to Draw II

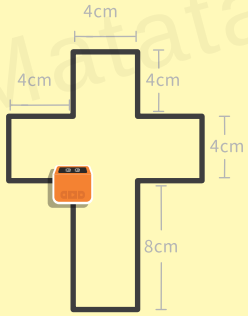
Task: Program VinciBot to draw more complex shapes, such as a cross, arrow, etc., and then allow for artistic creations based on the shapes.

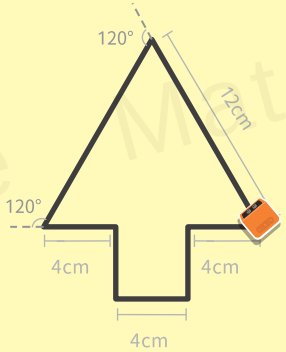
Loops (Nested Loops)

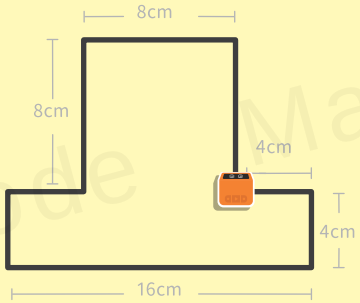
1 VinciBot can draw more complex shapes, as well as analyze and disassemble them.



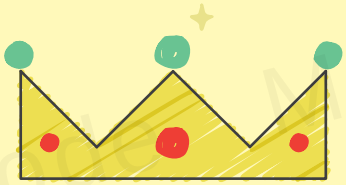


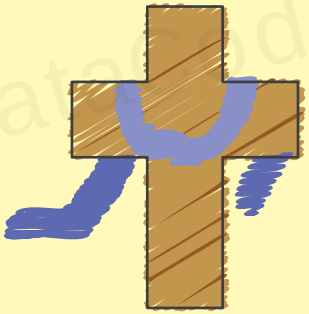


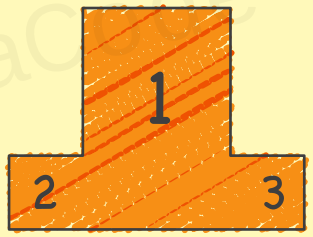




2 Program VinciBot to draw shapes and then add more details (i.e. by filling them with crayons) to make artistic creations.





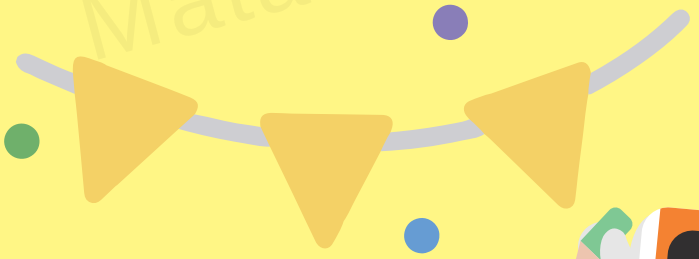


B11 The String Flags I

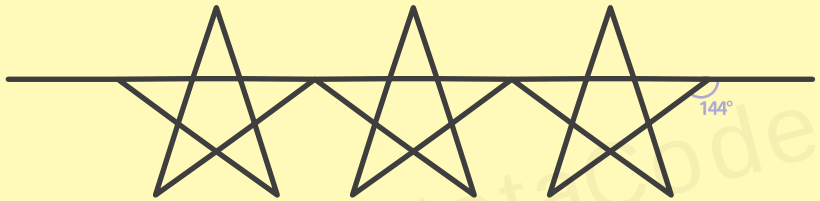
Task : Write a nested loop program to have VinciBot draw a string of identical shapes, such a string of triangles, a string of pentagons, etc.

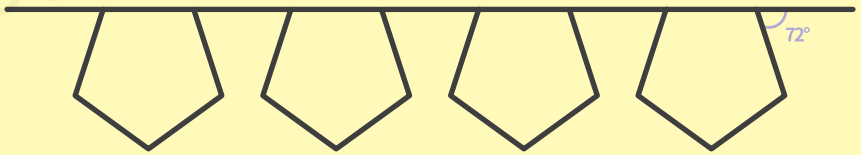
Loops
(Nested Loops)

1 Observe and analyze the program that draws the pennant below.



2 Try to draw more string flags below.





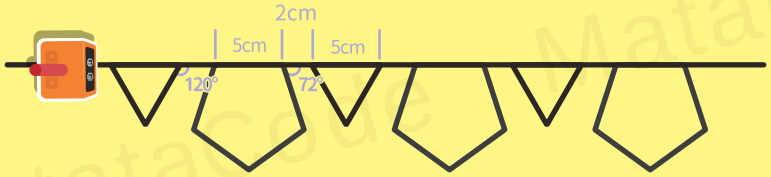
Bonus: What other shape’ s string flags could you draw?

B12 The String Flags II

Task :Write a nested loop program to have VinciBot draw a string of identical shapes, such as a string of squares and triangles.

Loops
(Nested Loops)

1 Observe and analyze the program that draws the string flags below.



2 Try to draw more string flags below.

Bonus：Design a unique string flag.

B13 VinciBot's New Dance

Events
(Multithreading)

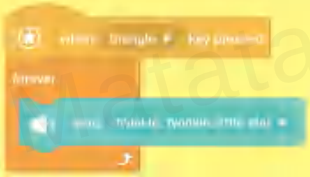
Task: Learn the multithreading, and program VinciBot to dance while singing.

1 Write a a single thread to have VinciBot repeat a unique "dance."

2 If you want to program VinciBot to dance while singing, an additional thread that makes VinciBot sing will need to be added.

knowledge points:
Multithreading is a model of program execution that allows for multiple threads to be created within a process, executing independently but concurrently sharing process resources.

Bonus: Test and consider potential problems in the program below. You can refer to the knowledge points in Activity A04 ("Information Transmission") and A10 ("The Rainbow Lamp").



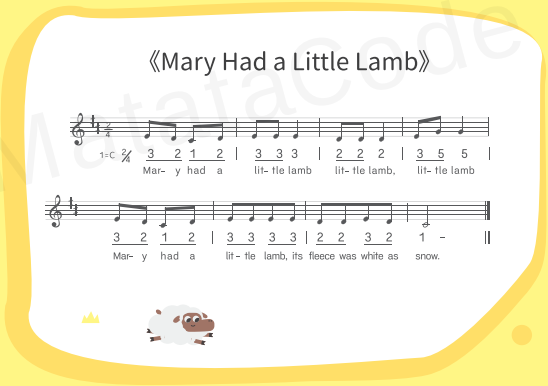
3 Import the multithreading into VinciBot, and run VinciBot to observe the effect.

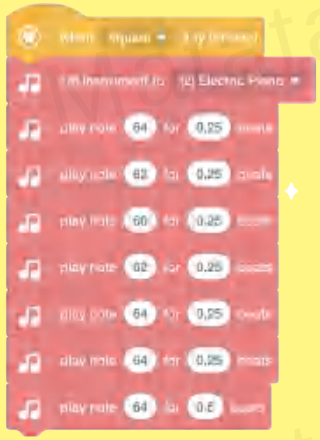
B14 VinciBot is a Superstar!

Events
(Multithreading)

Task: Run the multithreading to make VinciBot sing while blinking its LED lights.

1 Write a thread to have VinciBot sing a song, such as "Mary Had a Little Lamb."

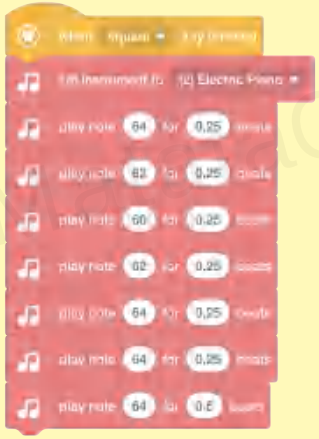


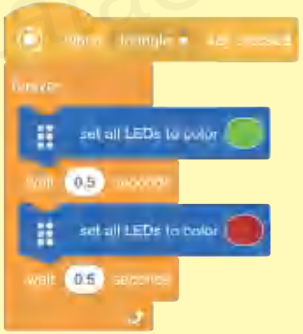


2 To achieve the effect of making VinciBot sing while blinking its LED lights, a thread that makes VinciBot blink its LED lights will need to be added.

3 Familiarize with the “stop script” coding block, and consider how to make VinciBot turn off its LED lights after finishing a song.

4 Import the multithreading into VinciBot and run VinciBot to observe the effect.





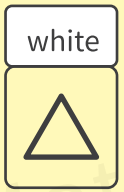
B15 Variety Lollipops

Task: VinciBot sells lollipops at the carnival. Write a multithreading so that VinciBot draws different lollipops that correspond to a detected color.

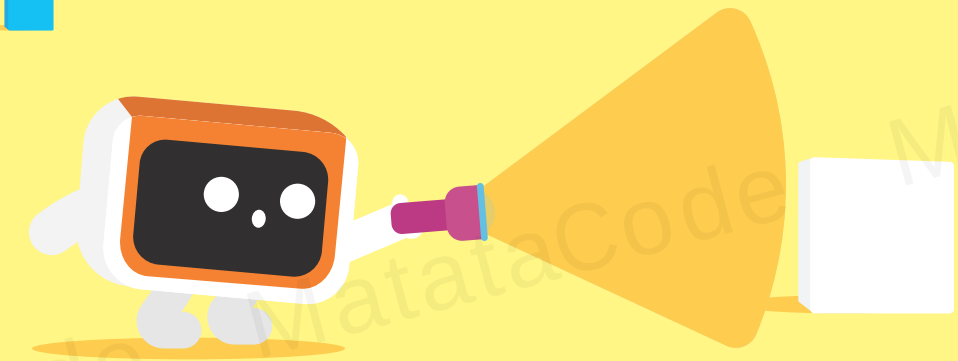
Events
(Multithreading)

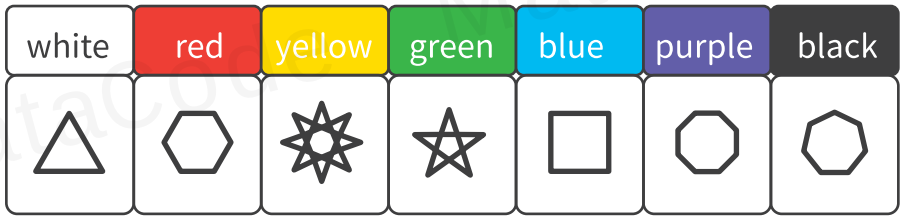
1 How many colors can be detected by VinciBot?

2 Write senven threads to program VinciBot to draw a lollipop with a shape that corresponds to a particular detected color. For example, when white is detected, a pentagram lollipop will be drawn.



3 Import the multithreading into VinciBot. Let your friends choose a preferred shape and draw a lollipop in this shape.





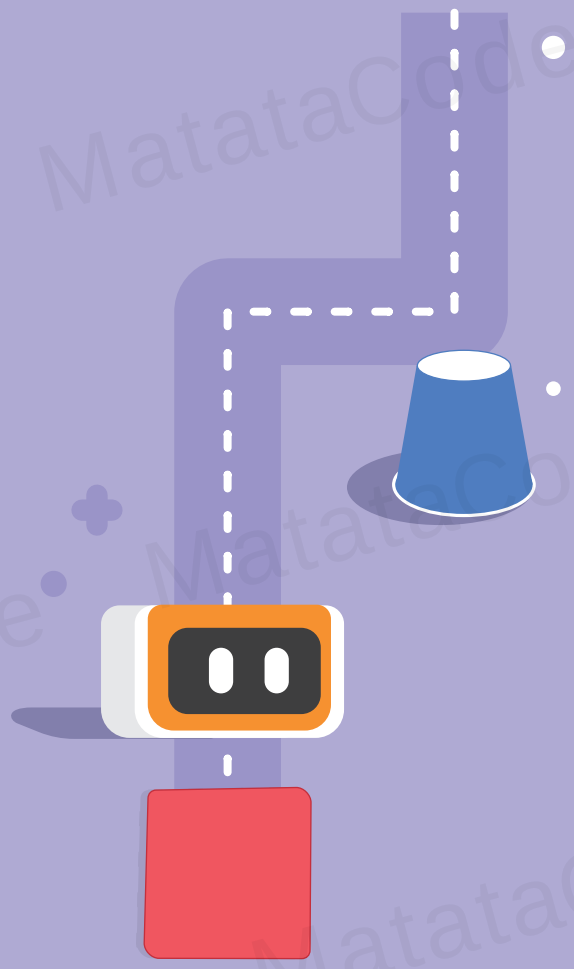
Number	Concept	Activity name
C-1	Conditionals (wait until)	Autopilot I
C-2	Conditionals (wait until)	Escape from the Chamber of Secrets
C-3	Conditionals (repeat until)	The Parade Float
C-4	Conditionals (repeat until)	VinciBot Fire Engine
C-5	Conditionals (If……then)	Light-On Reminder
C-6	Conditionals (If……then)	Eye Guard
C-7	Conditionals (If……then)	Cliff Detected
C-8	Conditionals (If……then)	Storytelling with the Pictures
C-9	Conditionals (If……then)	The Traffic Light
C-10	Function	Litter Stars
C-11	Function	Ode to Joy
C-12	Function	Puppy VinciBot
C-13	Function	VinciBot Warrior
C-14	Function	Autopilot II
C-15	Function	The VinciBot Train

C1 Autopilot I

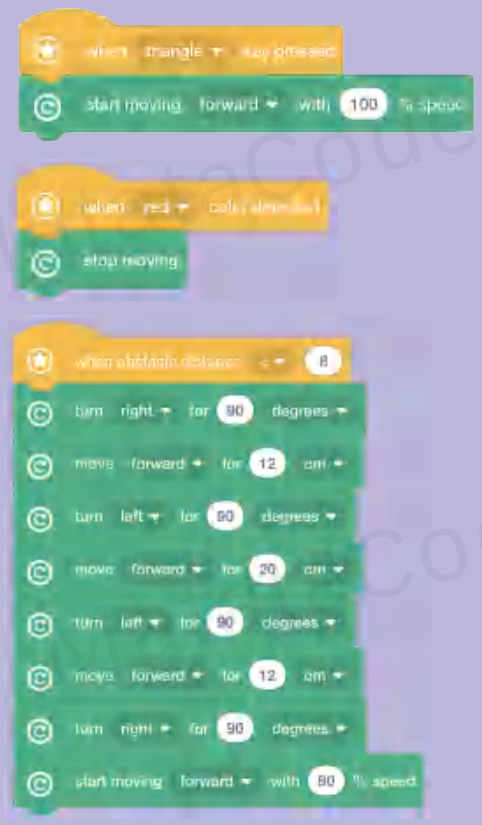
Conditionals
(wait until)

Task :Learn how to utilize the “wait until” statement in the conditional statement to allow VinciBot to automatically bypass obstacle(s) while running forward, and to stop when the red end point is detected.

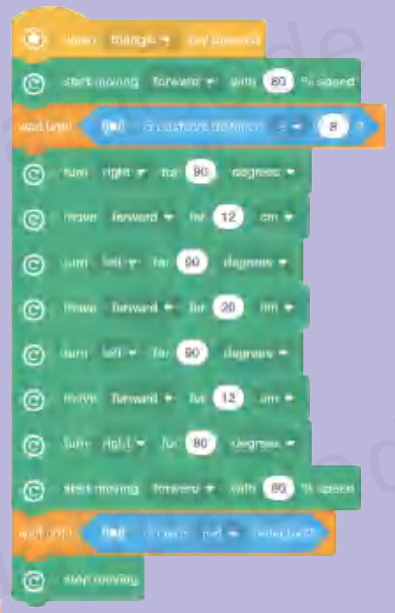
1 Set up the task scene:Set up an obstacle (paper cup) and a red end point.



2 Three subroutines can be written to have VinciBot move forward (subroutine 1); to automatically bypass obstacles and continue to move forward (subroutine 2); and to stop when the end point (red) is detected (subroutine 3).



3 Learn how to utilize the “wait until” statement in conditional statements; try to write a program that uses the “wait until” statement in order to allow VinciBot to achieve the same effect.



The ” wait until” coding block will allow VinciBot to keep checking the conditions in this block; until the condition is met, the next instruction will not begin.

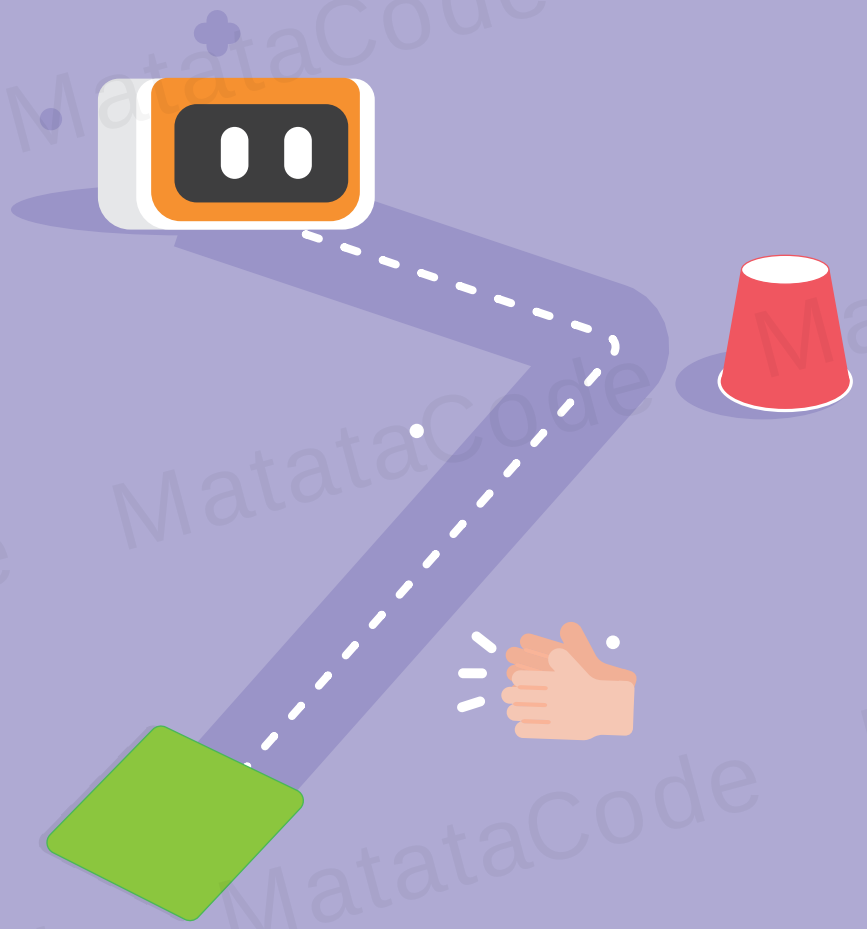
Bonus：If two or three obstacles are set before the red end point, how does the program need to be adjusted?

C2 Escape from the Chamber of Secrets

Conditionals
(wait until)

Task :Apply the “wait until” statement to program VinciBot to walk out of the Chamber of Secrets as directed.

1 Set up the task scene.



2 VinciBot needs to escape from the Chamber of Secrets according to the following guidelines:

VinciBot begins at the starting point and walks forward slowly. When VinciBot encounters an obstacle, it needs to turn right and continue to move forward slowly.

When a sound is detected, VinciBot should speed up.

When VinciBot reaches the green safe zone, it will stop and make a "yeah" sound to celebrate its escape.

Bonus：Design a new Chamber of Secrets task scene and attempt to escape from this secret room with repeated test attempts.

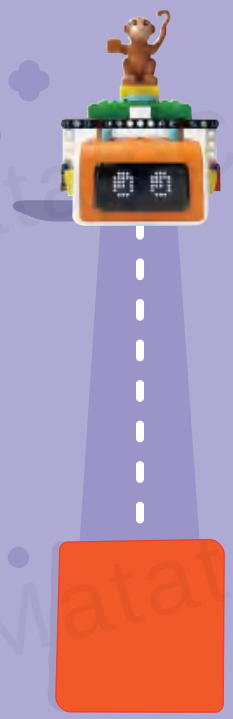
C3 The Parade Float

Conditionals
(repeat until))

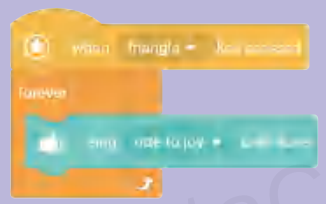
Task : Learn the “repeat until” statement, and use the “repeat until” coding block programming so that the VinciBot Parade Float makes facial expressions and sings while moving forward, and stops all actions when it reaches the red end point.

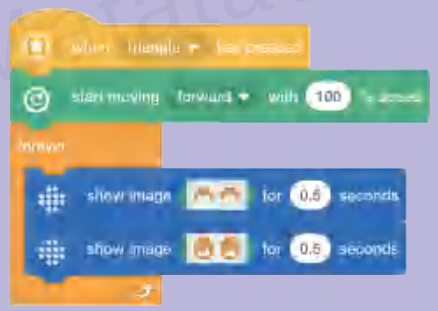
1 Dress up VinciBot as a parade float.

2 Set up the task scene:Place a red card on the side of a level table or surface as an end marker.



3 Program the VinciBot Parade Float to make facial expressions and sing while moving forward. Because the music and expressions are displayed at the same time, two subroutines are required.





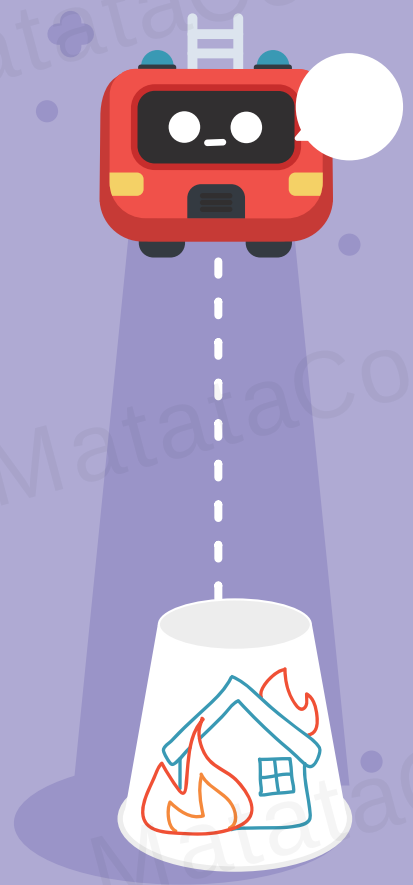
4 When the VinciBot Parade Float reaches the red end point, the movement, expressions, and music should all be stopped. Consider which repeat coding blocks should be replayed by the repeat until coding block in the two subroutines. Where should the “stop all” script blocks be placed?

C4 VinciBot Fire Engine

Task: Apply the “repeat until” statement to the program. When VinciBot Fire Engine detects an alert sound, it will rush to the fire point (obstacle), and continue displaying the warning sign. When VinciBot Fire Engine reaches the fire point (detects the obstacle), it will stop to put out the fire (make a “sprinkler” sound).

Conditionals
(repeat until)

1 Set up the task scene:Place a "burning" house on a level table or surface (build a house out of LEGO blocks or draw a burning house on a paper cup).



2 When VinciBot Fire Engine detects an alert sound, it will rush to the fire point (obstacle), and continue displaying the warning sign. When VinciBot Fire Engine reaches the fire point (detects the obstacle), it will stop to put out the fire (make a “sprinkler” sound).

3 After the fire is extinguished, the VinciBot Fire Engine will turn to face everyone with a “laughter” sound and a happy expression.

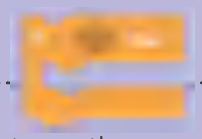
4 The demo program.

C5 Light-On Reminder

Task : Learn the “If...then” statement, and apply the light sensor so that VinciBot prompts the owner to turn on the light via voice and flashes lights when it detects that the surrounding environment is too dark.

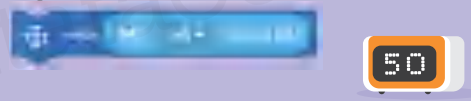
Conditionals
(If·····then)

1 Ambient light affects the eyes a lot. Consider how to transform Vinci-Bot into a device that prompts its owner to turn on lights when it detects that an environment is too dark. The “If...then” statement is required here.

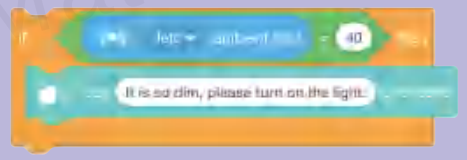


The biggest difference between the “If...then,” “wait until,” and “repeat until” statements is that the “If...then” coding block does not require VinciBot to check whether the conditions in the coding block are met constantly; rather, it only checks whether the conditions inside the building block are met once. To achieve continuous detections, this block is often used in conjunction with the “forever” block.

2 How can the current ambient light value be detected?



3 We can set:When the detected ambient light intensity is found to be less than 30, it means that the ambient light is too dark, and VinciBot will issue a prompt tone to remind the owner to increase (turn on) the light.



4 Add flashing LED lights to make reminders more visible.

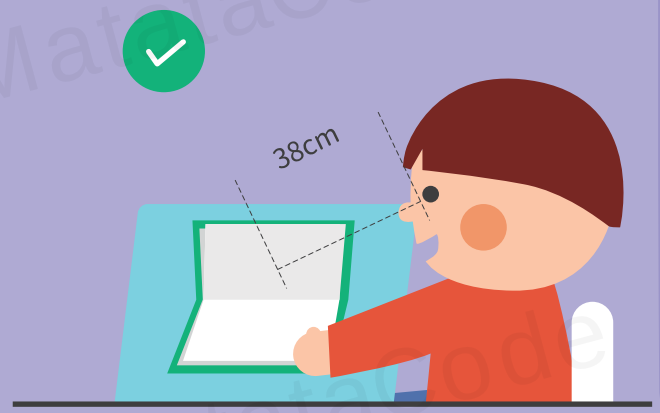
5 The demo program.

C6 Eye Guard

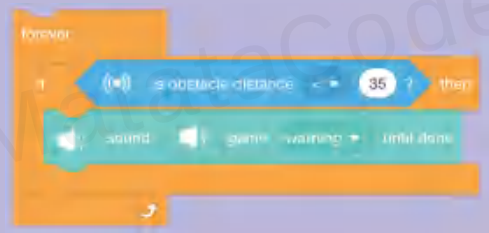
Task: Transform VinciBot into an Eye Guard with the ToF sensor: When it detects that the human eye is too close to the desktop, the VinciBot Eye Guard will utilize sound, expressions, and LED lights to remind the owner to maintain a good seated posture .

Conditionals
(If……then)

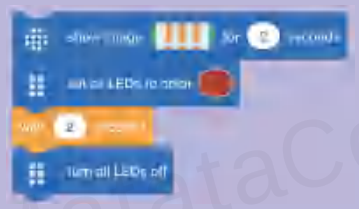
1 When reading, the best distance between one' s eyes and a book is 35-40 cm. If the distance is less than 35 cm, it can easily cause nearsightedness.



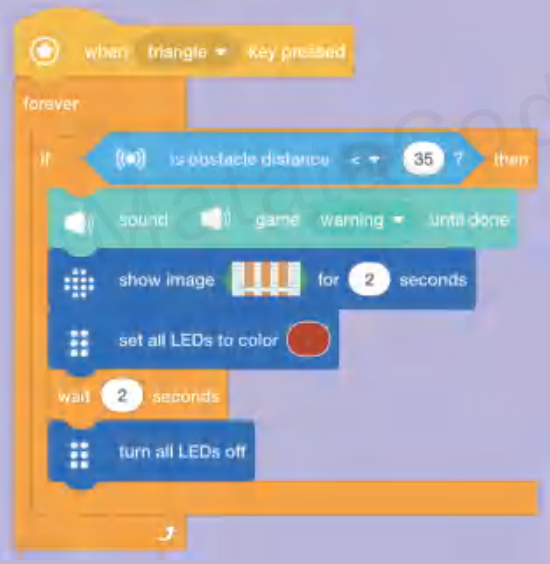
2 Write a program that allows VinciBot Eye Guard to continuously detect the distance between the human eye and a book, and if the distance is less than 35 cm, VinciBot Eye Guard will sound an alarm. Note: VinciBot should be placed next to the book, facing the eyes at a 45-degree upward angle.



3 Add expressions and LED lights to make alerts more visible.



4 The demo program.

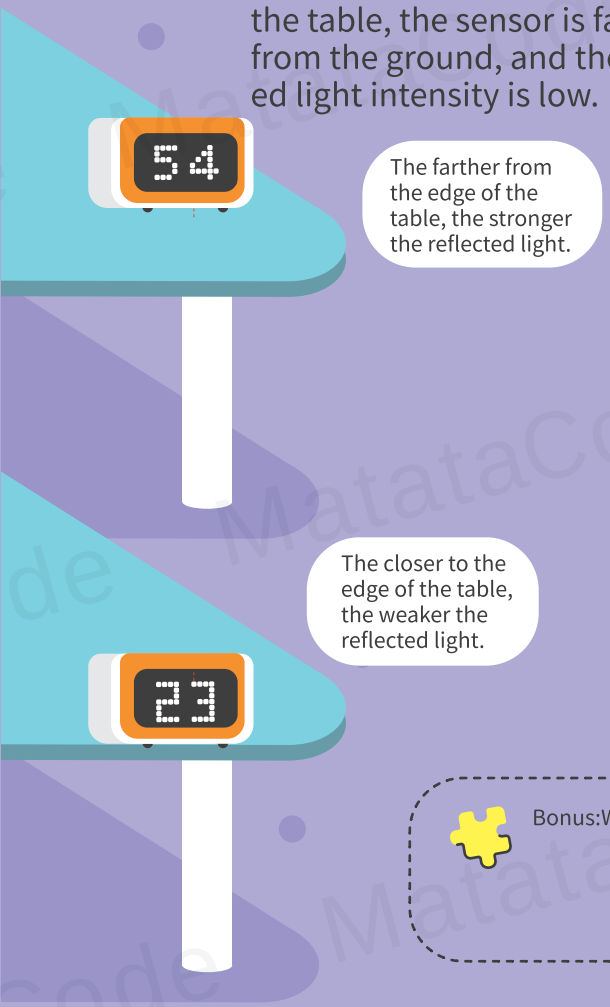


C7 Cliff Detected

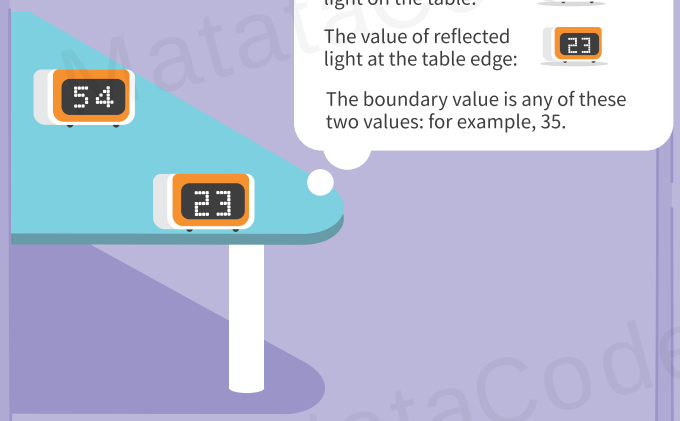
Task: Apply the function for detecting the intensity of reflected light with the line tracking sensor to detect a cliff; that is, to identify a point so that danger is averted: VinciBot is programmed to move forward, and when it detects that the reflected light on the edge of the table is weak, it stops moving forward and backs up to a safe position.

Conditionals
(If……then)

1 When VinciBot is on a table, the line tracking sensor at the bottom is close to the table, and the reflected light intensity is high; when VinciBot is on the edge of the table, the sensor is farther from the ground, and the reflected light intensity is low.



2 Write a program to test the reflected light value of VinciBot on the desktop and the edge of a table, and determine a boundary value.



3 Write a program that causes VinciBot to keep moving forward. When it reaches the edge of the desktop and detects that the reflected light value is less than the boundary value, it stops moving, lights up the red warning light, and moves back a certain distance.

Bonus:Write a program that causes VinciBot to move forward, but stops moving and switches its expression to “feared” when it detects that it has reached the edge of the desktop; it will then turn on the red warning light, step back 10cm to reach a safe position, switch the expression and light (to green), and turn right 90° to continue moving forward.

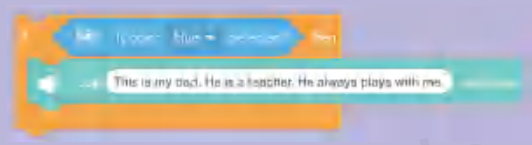
C8 Storytelling with the Pictures

Task: Apply the “If...then” statement so that VinciBot tells a story about a character or item represented by that color whenever it detects said color.

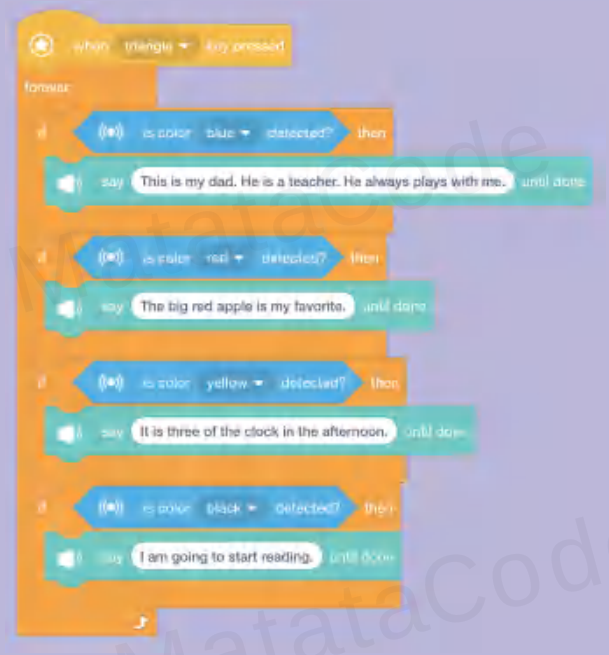
Conditionals
(If……then)

1 Draw a colorful painting with one dominant color for each character and object, such as “Dad in a blue shirt.”

2 Apply the “If...then” statement so that VinciBot recites a paragraph about the character or item represented by that color for each color it detects.



3 The demo program.

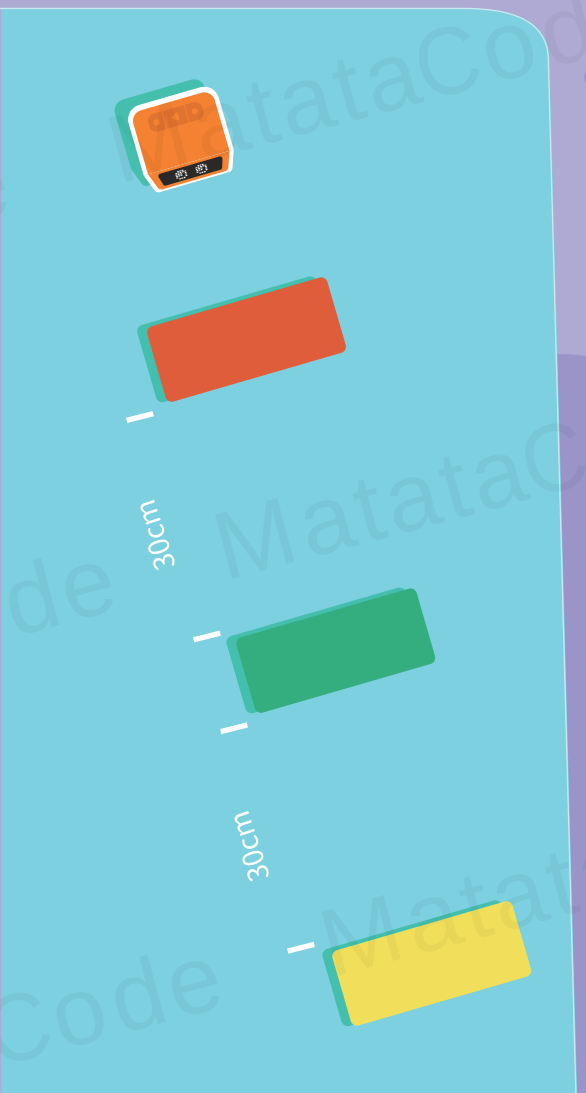


C9 The Traffic Light

Task: Apply the color sensor to simulate the situation in which a car encounters traffic lights; when the triangle button is pressed, VinciBot starts moving forward, but it will react differently when encountering red, yellow, and green lights; and when the square button is pressed, VinciBot stops moving.

Conditionals
(If……then)

1 Set up the task scene:Set up the red, green, and yellow cards in order and place them in a straight line.



2 Program VinciBot keep moving forward. When it encounters a red light, it will wait for five seconds, and then continue moving forward; when encountering a green light, it will pass directly through; when encountering a yellow light, it will wait for one second and beep before continuing to move forward.

3 Because each card has a width, it takes VinciBot a certain amount of time to pass. Therefore, a waiting time must be added after each instance of moving forward, otherwise, VinciBot will continuously detect that color and may be unable to pass the card successfully.

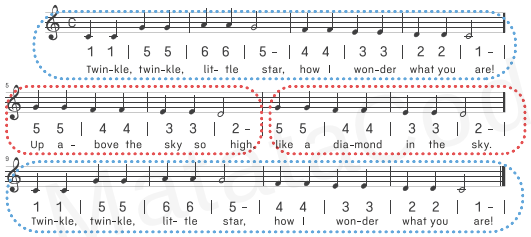
C10 Twinkle Twinkle Little Star

Function

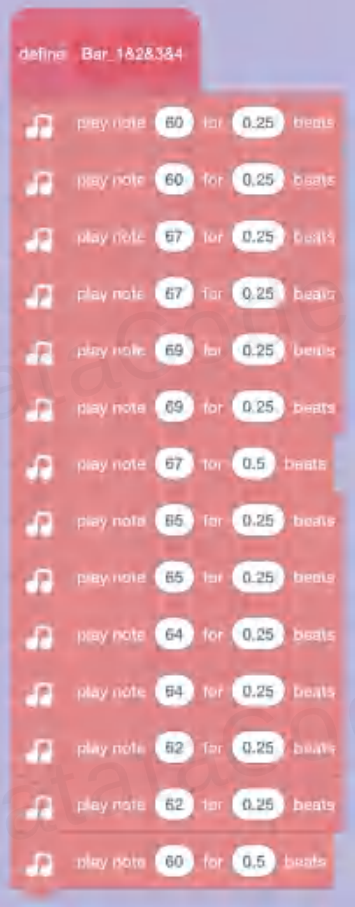
Task: Learn to utilize basic functional statements: when writing music programs, the same bars of music often appear. Allocate the repeated section into a new block so that it is able to complete the writing of the music program more efficiently.

1 Look at the music score for "Twinkle Twinkle Little Star" and identify the same bars in the score.

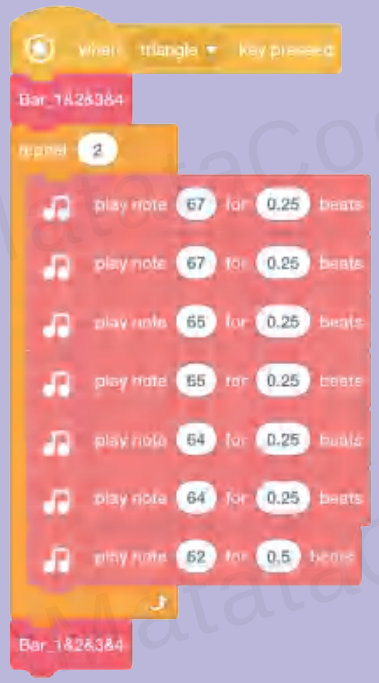
《Twinkle Twinkle Little Star》



2 Define the repeated but discontinuous sections as a new block.



3 Invoke the new block to finish writing the music program of "Twinkle Twinkle Little Star" .



In the function statement, a group of instructions that appear multiple times can be defined in a new block according to specific requirements; this new block can then be invoked several times, effectively simplifying the program.

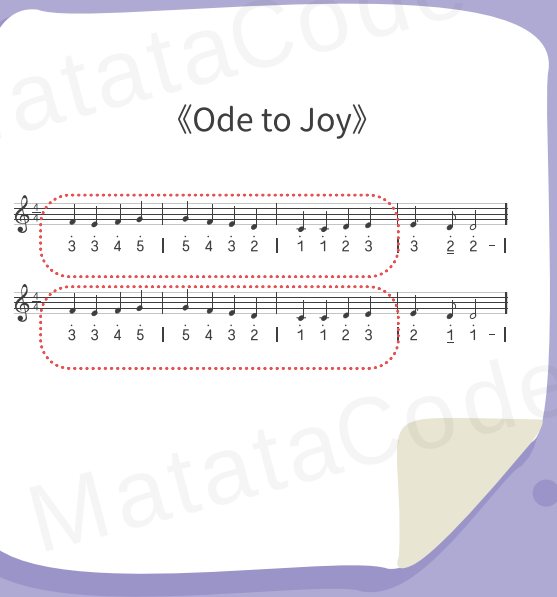
Bonus: Write a subroutine to make VinciBot sing while blinking and flashing the rainbow LED lights.

C11《Ode to Joy》

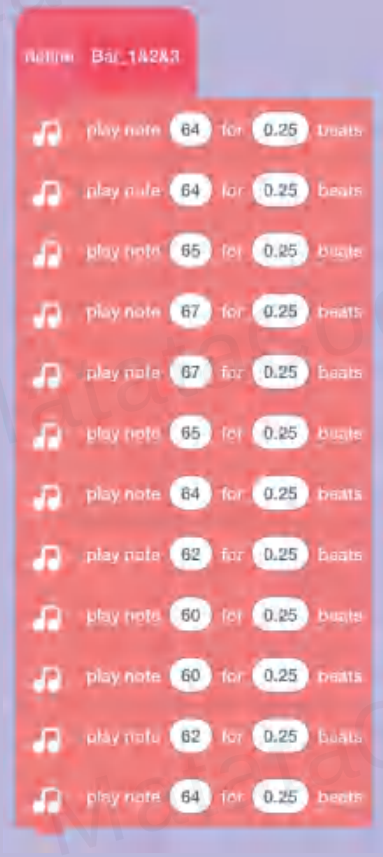
Function

Task: Familiarize with basic function statements, and apply these functions to independently write the music program "Ode to Joy."

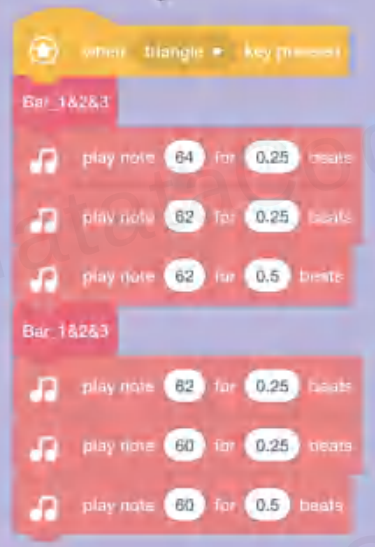
1 Observe the music score for "Ode to Joy" and identify the repeated bars in the score.



2 Define the repeated section as a new block, "Bar1&2&3".



3 Invoke the new block to finish writing the music program of "Ode to Joy."



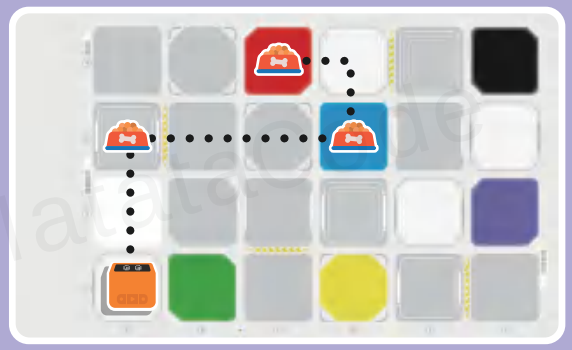
Bonus: Write a subroutine to make VinciBot sing while blinking and flashing the rainbow LED lights.

C12 Puppy VinciBot

Task: Make a "happy" block, which allows VinciBot to imitate a dog barking happily, as well as blinking and turning around when it eats a treat. Program Puppy VinciBot to eat all the treats on the map, and each time it eats a treat, it will be very "happy."

Function

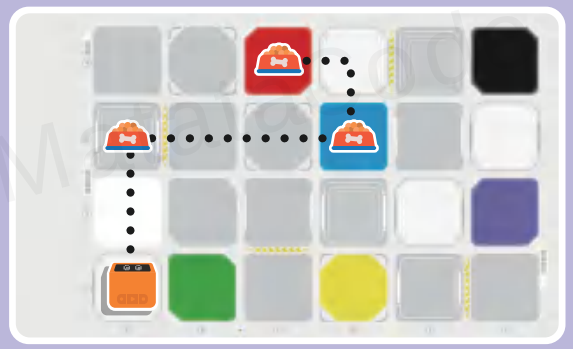
1 Set up the treat cards on the map as shown below.

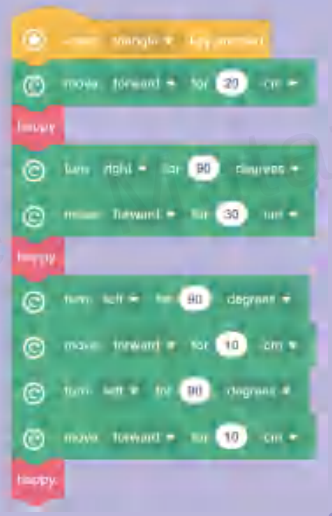


2 Design a group of puppy actions with motion, lights, and a puppy bark, and combine this group of actions into a new block called "Happy".

Bonus:Plan the route and write the shortest program to make Puppy VinciBot eat all the treats on the map.

3 Plan the route and program Puppy VinciBot to eat all the treats, and each time it eats a treat, it will be very "happy" (An example is shown below.).



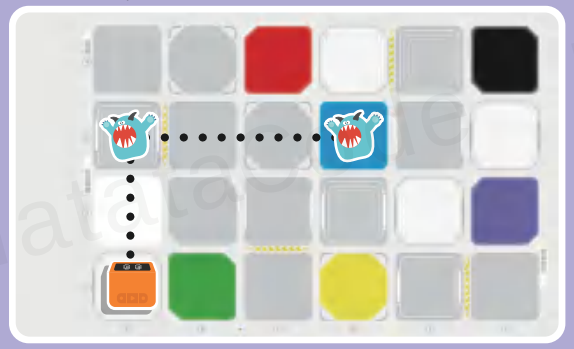


C13 VinciBot Warrior

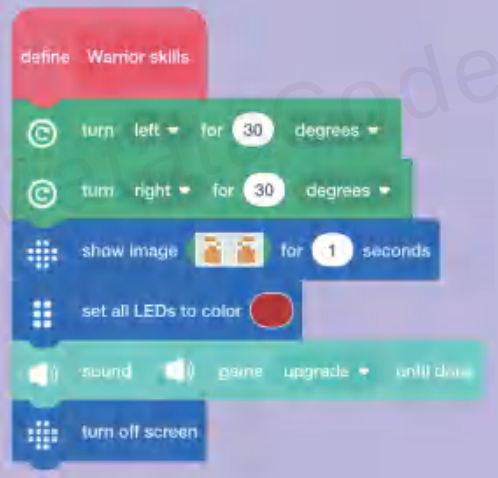
Task: Make a "Warrior Skill" block, so that VinciBot Warrior can call upon this skill to defeat the monsters in the task scene every time VinciBot Warrior encounters one.

Function

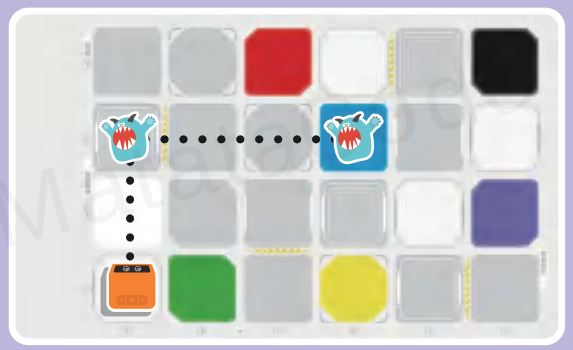
1 Set up the monster cards on the map as shown below.

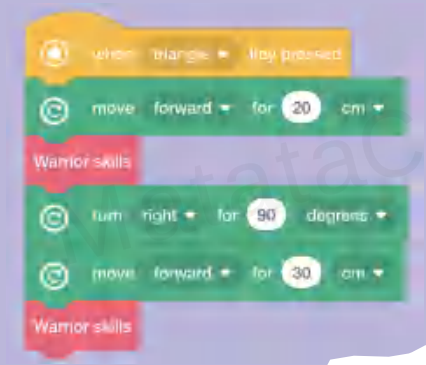


2 Make a "Warrior Skill" block so that VinciBot Warrior can defeat the monsters by using motion, light, and sound blocks.

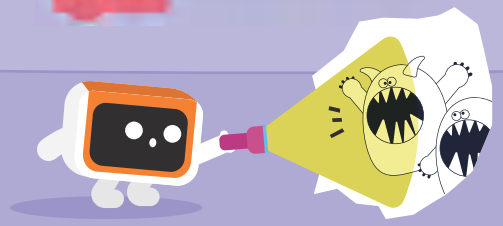


3 Plan the route and program VinciBot Warrior to defeat all the monsters using the "Warrior Skill" block. (An example is shown below.)





Bonus:Plan the route and use the shortest program to make VinciBot Warrior defeat all the monsters on the map.

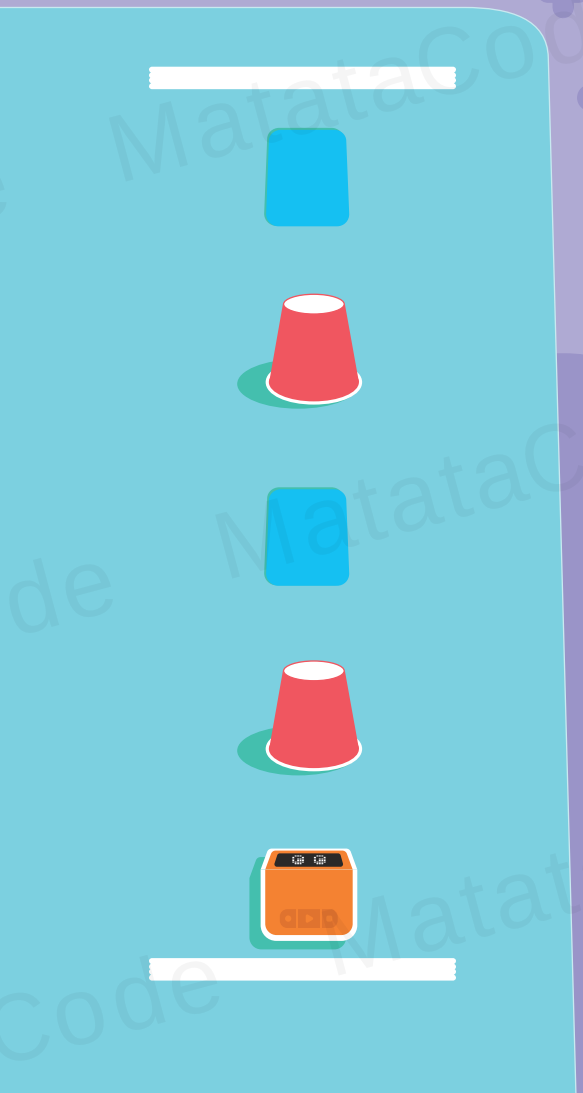


C14 Autopilot II

Function

Task: The VinciBot Autopilot needs to automatically avoid obstacles on the road and automatically refuel every time it encounters a blue gas station.

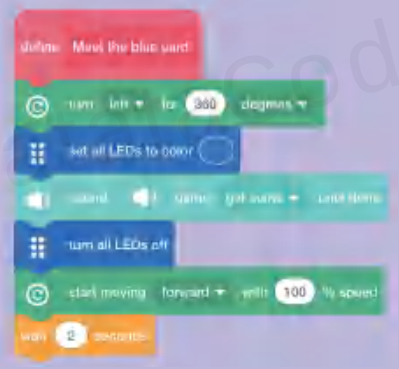
1 Set up the task scene: place two paper cups (obstacles) and two blue cards (gas stations) on a straight road.



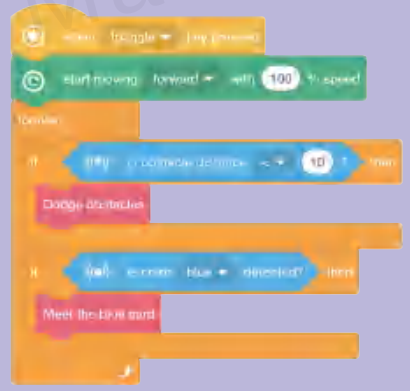
2 Make a new "obstacle avoidance" block, and program VinciBot to automatically bypass each obstacle, making a “score” sound each time.

Bonus:Try to make different blocks for avoiding obstacles on the road and refueling at the gas station.

3 Make a new "Meet the Blue Card" block; program VinciBot to turn around, turn on the blue light, and make a “get coin” sound each time it encounters a blue gas station.



4 Write a program that allows VinciBot to begin at the starting point, invoke each new block twice, and successfully reach the end point.

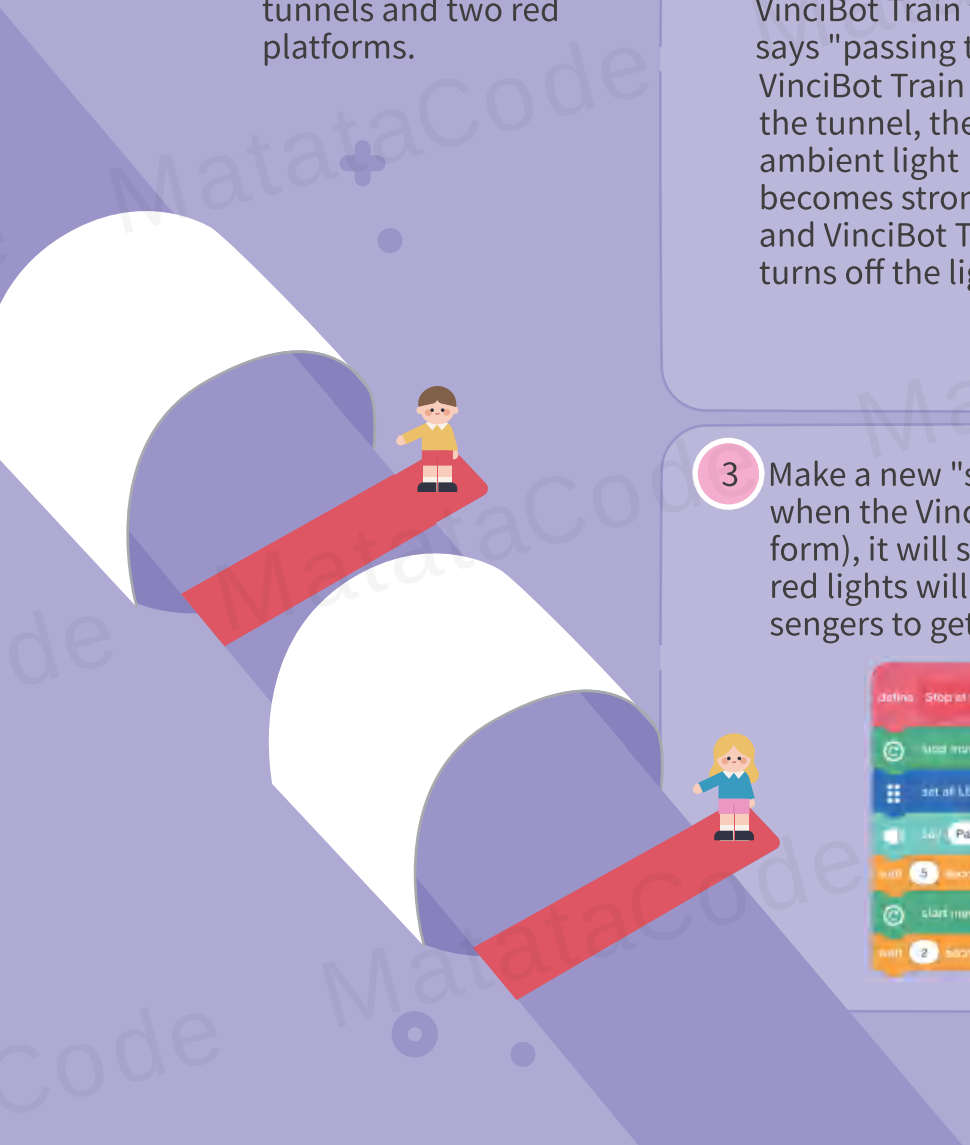


C15 The VinciBot Train

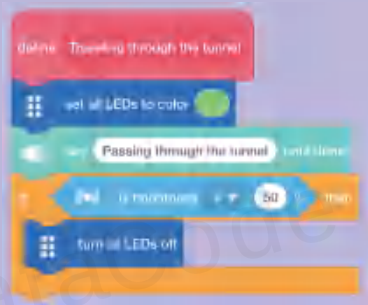
Function

Task:The VinciBot Train is going to pass through the tunnel; please program it to successfully pass through multiple tunnels as required and automatically stop each time it reaches a red platform.

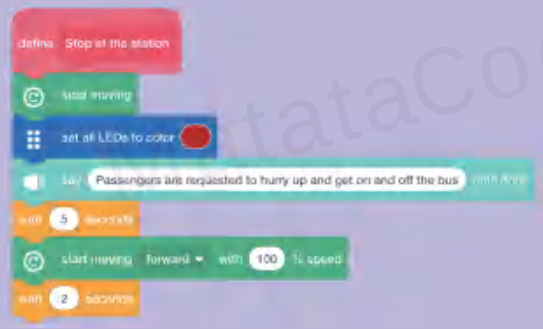
1 Set up the task scene:A train track has two LEGO or cardboard tunnels and two red platforms.



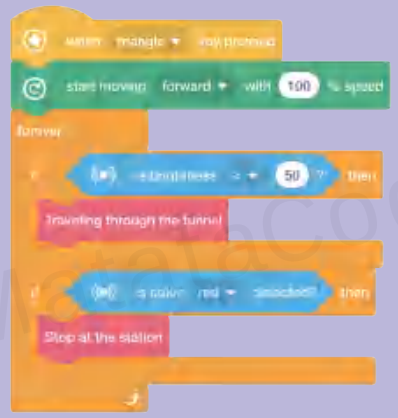
2 Make a new "traveling through the tunnel" block: when the VinciBot Train enters the tunnel, the ambient light becomes weak, VinciBot Train turns on the green light and says "passing through the tunnel"; when the VinciBot Train exits the tunnel, the ambient light becomes stronger, and VinciBot Train turns off the light.



3 Make a new "stop at the station" block; when the VinciBot Train detects red (platform), it will stop for five seconds, and the red lights will be turnedon to remind passengers to get on and off the train.



4 The demo program.



Bonus:Program VinciBot Train to slow down each time it enters a tunnel and accelerate after exiting a tunnel.

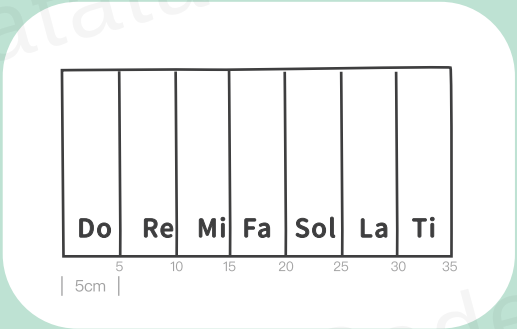
Number	Concept	Activity name
D-1	Conditionals (if then)	The Magic Air Piano
D-2	Conditionals (if else)	Coward VinciBot
D-3	Conditionals (if else)	Close Friends
D-4	Conditionals (if else)	Light Chaser 1.0
D-5	Variables	Spiral Graphics
D-6	Variables	Marathon
D-7	Variables	Charging Station
D-8	Variables	The Reward and Punishment Machine
D-9	Variables	Pleasant Music
D-10	Variables	Catch 3!
D-11	Variables	The Counter
D-12	Variables	Stopwatch
D-13	Function (multiple function)	Speed Change by Color
D-14	Function (multiple function)	Regular Polygons
D-15	Function (multiple function)	Dancing VinciBot

D1 The Magic Air Piano

Task : Learn the meaning and usage of the "AND, OR, and NOT" coding blocks. Then use the new blocks, the ToF ranging sensor, and the music blocks to make an "air piano."

Conditionals
(if then)

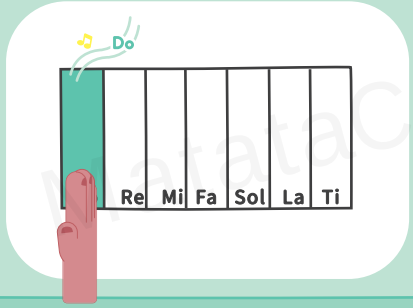
1 Draw seven equal distances on the white paper (Recommended distance is 5 cm; However, the distance can be adjusted according to the actual playing habits). Write Do, Re, Mi, Fa, Sol, La, and Ti on each space.



2 Learn the meaning and usage of the "AND, OR, and NOT"coding blocks.

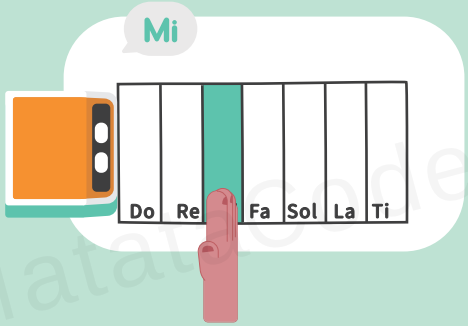
The operations of "AND, OR, and NOT", respectively, indicate when the conditions on both sides are satisfied at the same time ("AND"); when one of the conditions is satisfied ("OR"); when these conditions are not satisfied ("NOT"), execute the next command.

3 When the ToF ranging sensor measures different distance ranges, different notes are played.

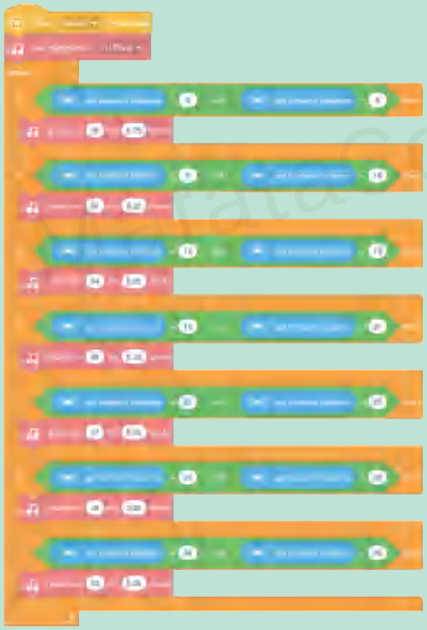


Bonus:When playing the air piano, the corresponding note should be displayed on the dot matrix screen.

4 Put your hands on the "keys" and play beautiful piano songs.



5 The demo program.

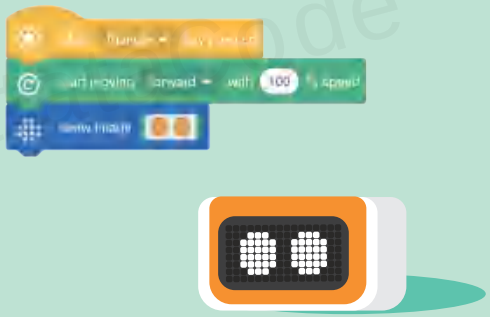


D2 Coward VinciBot

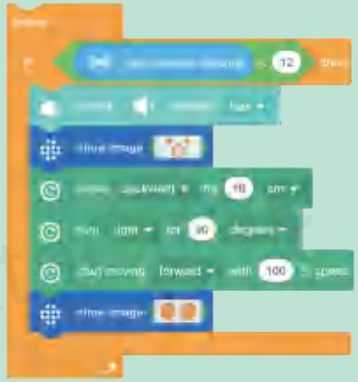
Task: Learn how to utilize the “if else” statement in conditional statements and the program flow diagrams. The Coward VinciBot has been walking forward with wide eyes. Whenever it encounters an unknown obstacle, it will move back out of “fear,” turn right, and then keep moving forward with wide eyes.

Conditionals
(if else)

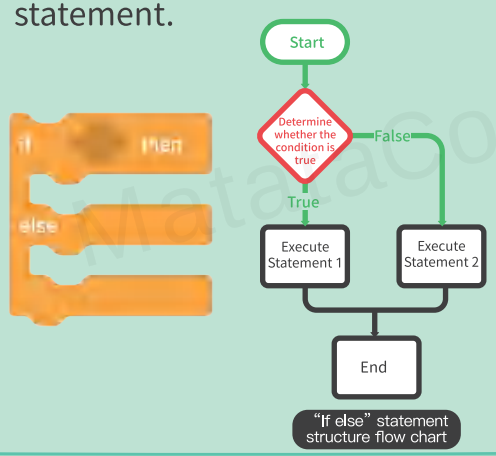
1 The VinciBot keeps moving forward, with a wide-eyed expression displayed on the dot matrix screen.



2 First, use the "if then” statement to program. If VinciBot encounters an obstacle, it will move backward for 10 cm while making a "fear" sound and a frightened expression; then it will turn right and continue to move forward with wide eyes .

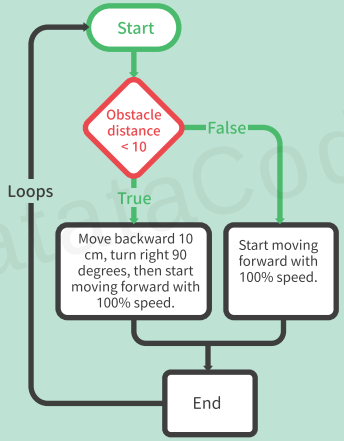


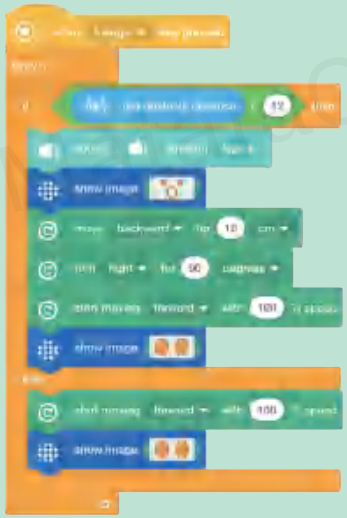
3 Learn to utilize the “if else” statement.

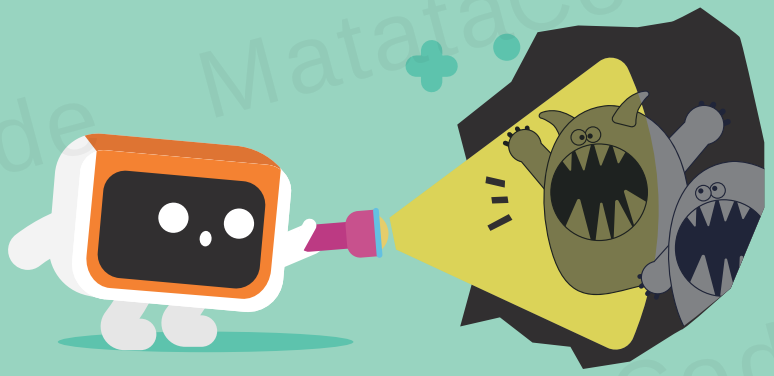


“If else” statement structure flow chart

4 Rewrite the program by using the “if else” statement.







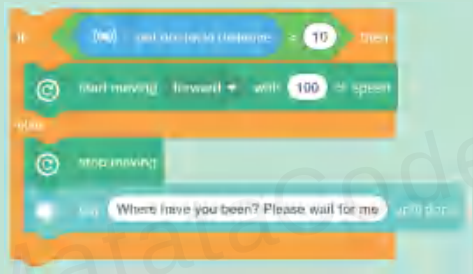
D3 Close Friends

Task: Use the ToF ranging sensor to make VinciBot follow the little bear. When the little bear is suddenly picked up, VinciBot stops moving and asks "Where have you been?" Then when the little bear comes back, VinciBot continues to follow it.

Conditionals
(if else)

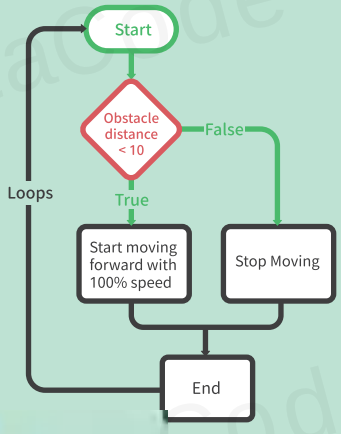
1 Prepare a little bear (or other toy) and place it very close to VinciBot.

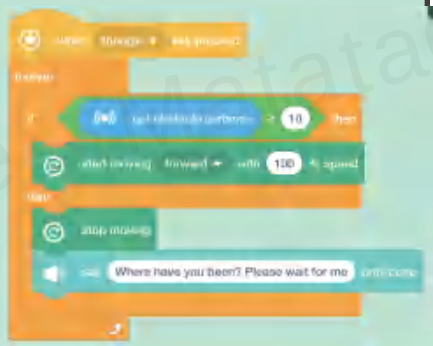
2 When VinciBot is within a certain distance of the bear, make it approach the bear, moving forward slowly. VinciBot will stop and say "Where have you been? Please wait for me".



Note: Move the bear forward slowly by hand.

3 Every time VinciBot says “Please wait for me” , the bear will be moved back towards VinciBot. VinciBot will move towards the bear again. Thus, the “forever” coding block will be used.





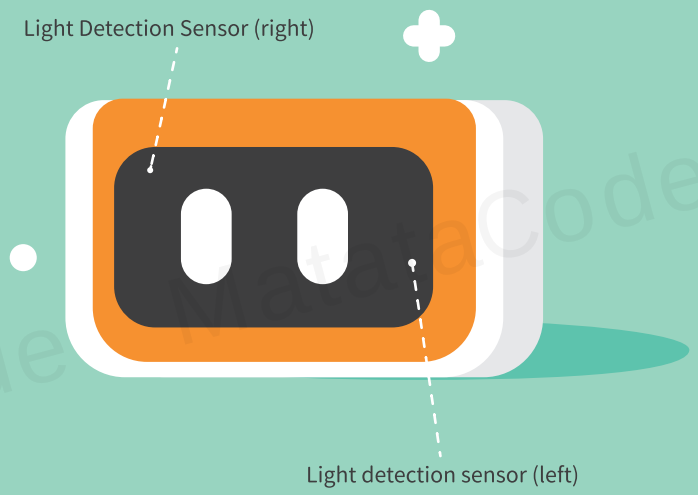
Bonus:Write a new program. When VinciBot is following the bear, the distance between VinciBot and the bear will be displayed in real time on the screen.

D4 Light Chaser 1.0

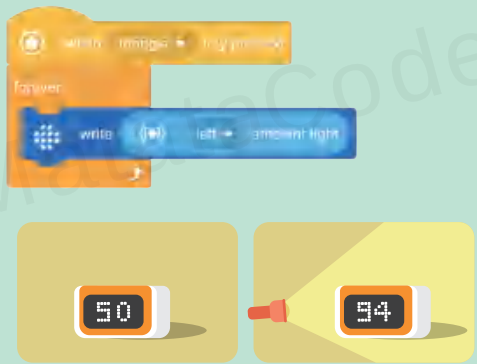
Task: Use the light detection sensor to program and control VinciBot to following a strong light source. When the strong light source disappears, VinciBot will stop moving.

Conditionals
(if else)

1 Behind the dot matrix screen, there are two light detection sensors (on both the left and right sides) which can detect changes in ambient light in front of VinciBot. The value of the ambient light is between 0 and 100.



2 Write a program to test the value of ambient light on one side under normal ambient light. Then, prepare a strong light source, such as a flashlight. Turn on the flashlight, point it directly to the front of VinciBot, and test the ambient light value again. (Note: This activity should be performed with normal ambient light values between 40-60 as much as possible.)

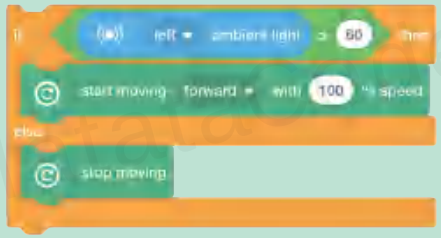


3 Determine a boundary value (an intermediate value between the two ambient light values) based on the ambient light values in normal light and strong light.

50 60 94

Boundary Value

4 Write a program:When the ambient light is greater than a certain value, VinciBot keeps following the strong light; otherwise, it stops.



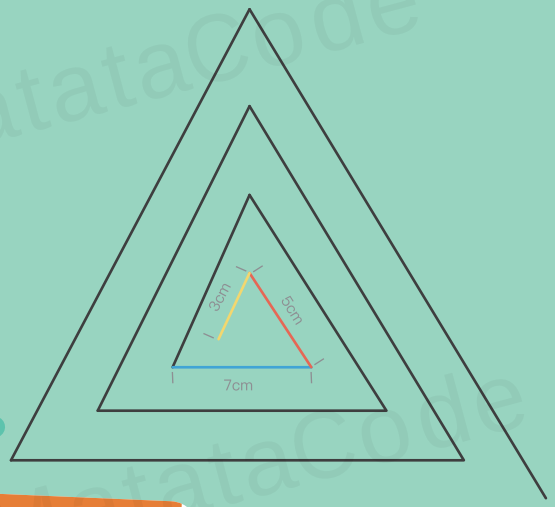
5 The Demo Program.

D5 Spiral Graphics

Variables

Task: Study the structure and characteristics of spiral graphics with VinciBot; study the variables, and make VinciBot draw the spiral graphics by using variables via programming.

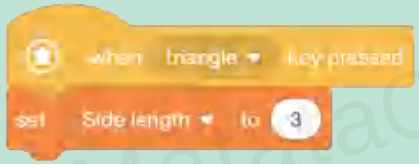
1 A characteristic of the spiral graph is that its side length will change continuously during the drawing process.



2 Set the variable to represent the side length of the graphic.

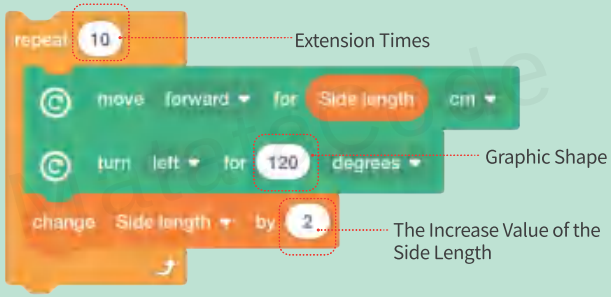
The value of the variable block will change as the program runs.

3 Set the initial side length of the graphics.



Bonus:Draw a square spiral graphic. Why are nested loops required to draw a square spiral graphic?

4 Set the extension times of the spiral graphic, the graphic shape, and the increase value of the side length.



5 Draw triangle and pentagon spiral graphics.

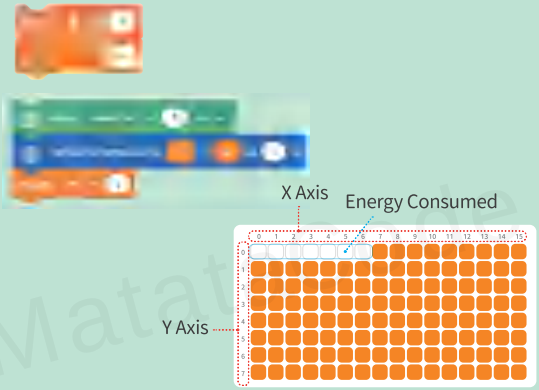
D6 Marathon

Variables

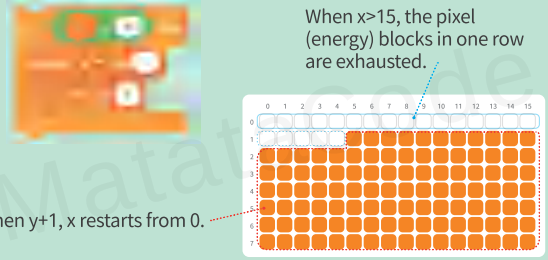
Task：VinciBot participates in a marathon competition. Every time it moves forward for 5 cm, one pixel (energy) block will be turned off. When the dot matrix screen turns black, have VinciBot say "I'm too tired", turn on its red LED lights, and turn them off after three seconds.

1 Before the competition, program VinciBot to have all of its pixel blocks lit up and its green LED lights turned on.

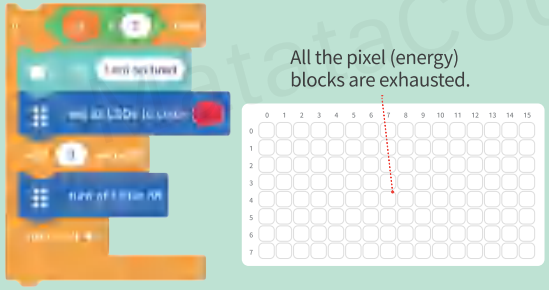
2 Every time VinciBot moves forward 5 cm, one pixel (energy) block will be turned off: set two variables (x, y), representing the coordinates of each pixel block.



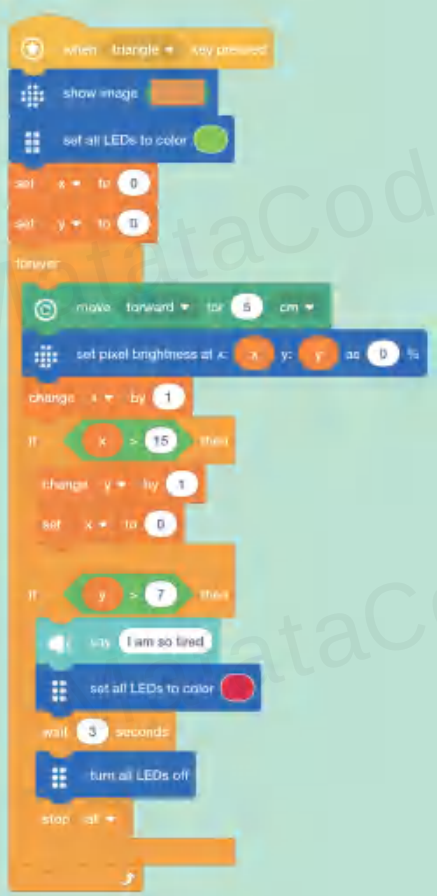
3 When the pixel (energy) blocks in one row are exhausted, the pixel (energy) blocks in the next row will also be consumed.



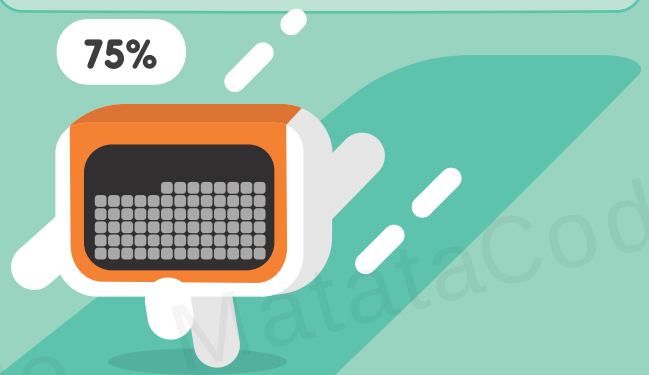
4 When all the pixel (energy) blocks are exhausted, VinciBot says "I'm so tired", turns on its red LED lights, and turns off them after three seconds.



5 The demo program.



Bonus：When there are only two rows of pixel (energy) blocks left, turn on the yellow LED lights to indicate low energy.

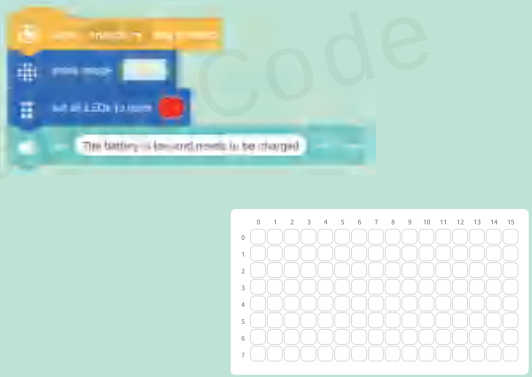


D7 Charging Station

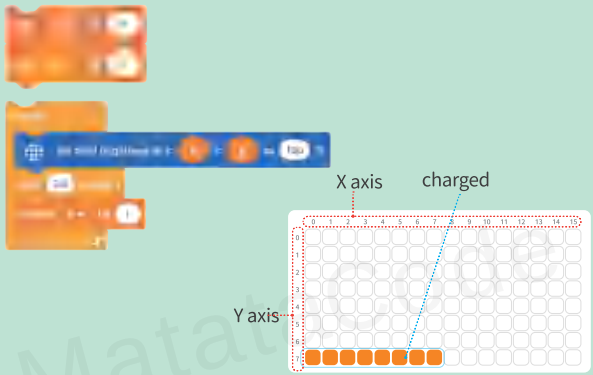
Task: VinciBot ran out of energy during the marathon and needs emergency charging; after pressing the triangle button, VinciBot says "low battery, start charging", and the pixel (energy) blocks on the dot matrix screen gradually light up; when the dot matrix screen is fully lit, charging will stop and VinciBot should say "the battery is fully charged".

Variables

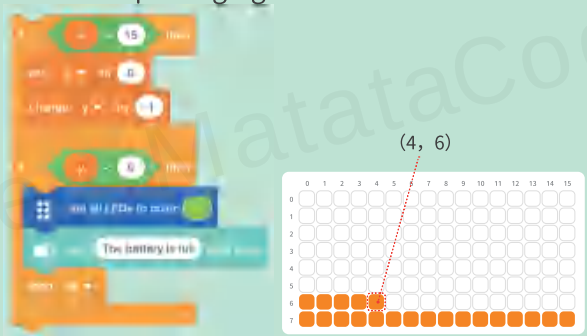
1 VinciBot ran out of energy during the marathon and needs to be charged.



2 One pixel (energy) block will light up every 0.2 seconds. Two variables (x, y) should be set to represent the coordinates of each pixel (energy) block.



3 When one row of pixel (energy) blocks are fully lit up (x>15), it is necessary to start from a new row; that is, x is reset to 0, and y is reduced by 1. When all pixel (energy) blocks are lit up, i.e. y<0, stop charging.



4 The demo program.

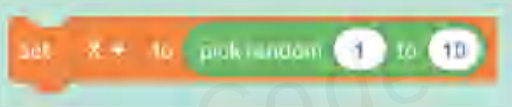
Bonus:When the first six rows' pixel (energy) blocks are lit up, the charging speed is reduced by half. Meanwhile, the blue LED lights are turned on until all pixel (energy) blocks are lit up. The blue LED lights will then change to green.

D8 The Reward and Punishment Machine

Variables

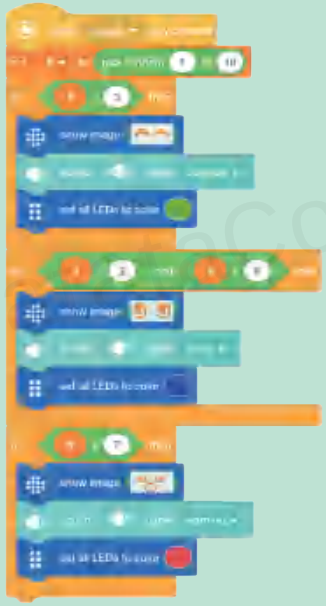
Task: Transform VinciBot into a party tool - a reward and punishment machine! Set three modes: "Pass", "Reward", and "Punish" for VinciBot. A random situation occurs when the button is pressed.

1 Set a variable to take a random value between 1-10.



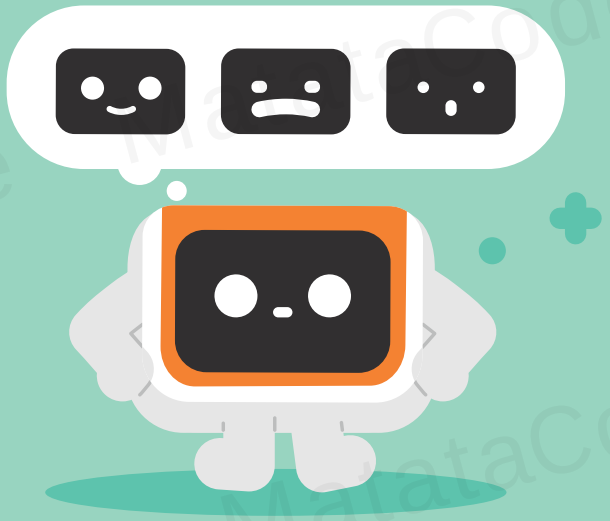
2 Set the probability of the three modes ("Pass", "Reward" and "Punish"). Use expressions, LED lights, sound effects, and other coding blocks to design the effects of the three modes.

3 The demo program.



4 Press the button to start the game, and compare who has better luck! For example: "Pass" means safety, "Reward" means you can assign one person to perform a show, and "Punish" means you need to complete Truth or Dare.

5 Change the probability of the three modes, and compare the results of the statistical game over several plays!



D9 Pleasant Music

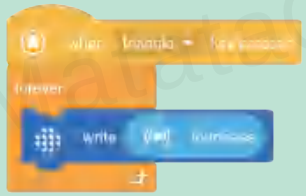
Variables

Task :VinciBot is listening to music. Program VinciBot so that the pixel blocks on the dot matrix screen rise and fall with the music.

1 Three different heights of audio diagrams are designed by displaying pixel blocks, corresponding to different sound volumes.

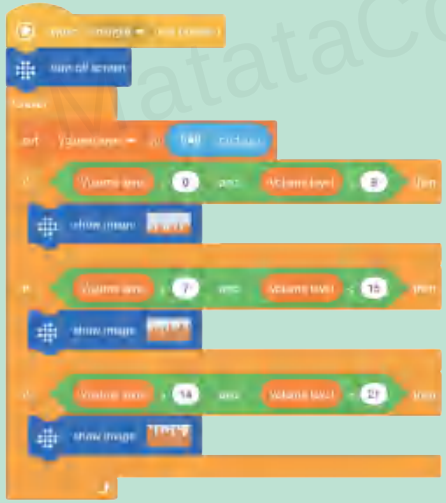
2 Set the variable "Volume Level" to indicate the detected sound level.

3 First detect the approximate range of the loudness of the music to be played.



4 Set three volume intervals, and make VinciBot show different audio diagrams corresponding to different sound volumes.

5 The demo program



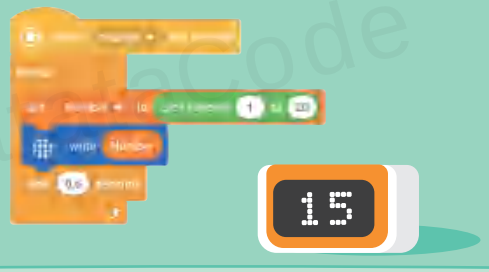
Bonus：Based on the program above, use “LED light” coding blocks to program so that the LED lights change colors according to different volumes.

D10 Catch 3!

Variables

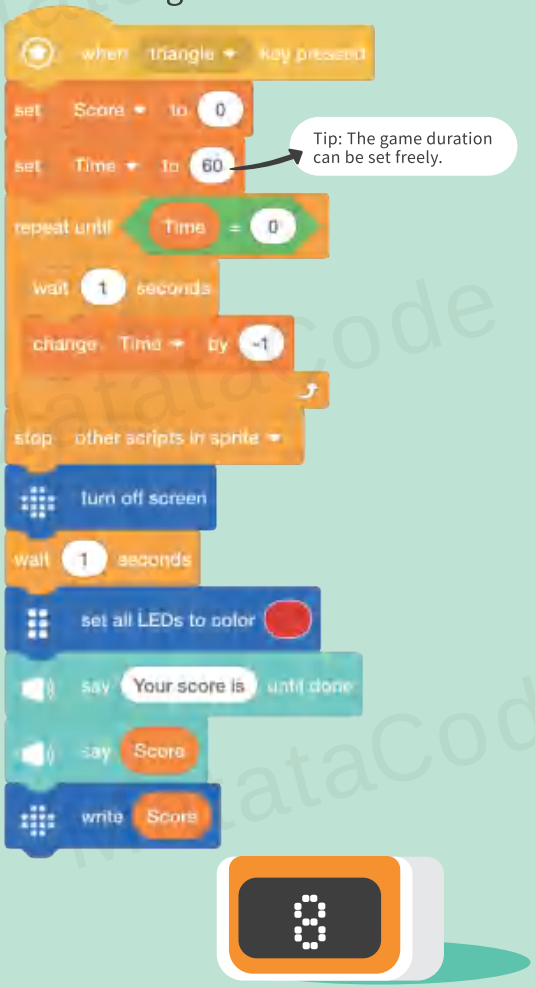
Task: The numbers 1-20 appear randomly on the dot matrix screen of VinciBot within a set time. Please observe the random numbers and "grab" the number 3 or a multiple of 3 by pressing the button. Finally, observe how many 3s you have caught.

1 Game starts! VinciBot randomly displays numbers from 1 to 20, and sets the interval time between the numbers.

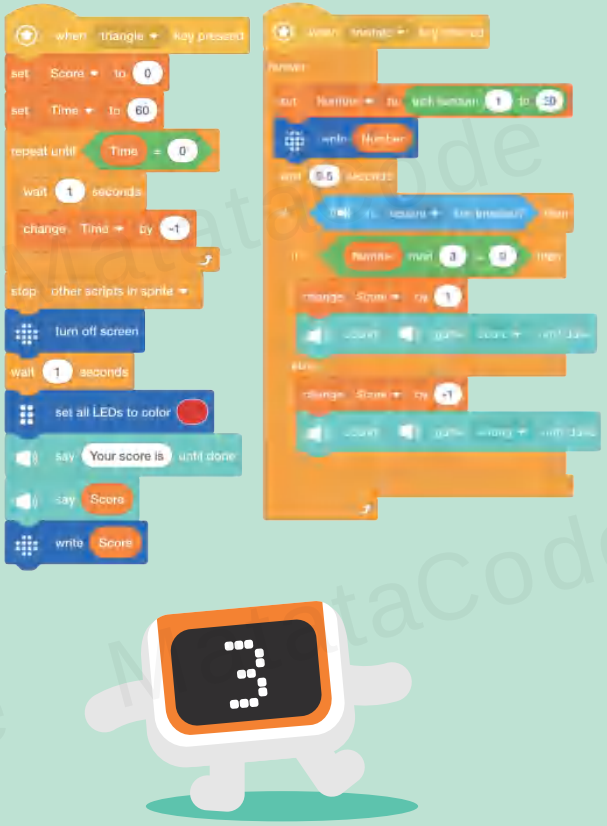


2 Set the scoring mechanism: when the displayed number is 3 or a multiple of 3, quickly press the square button. When a 3 is successfully caught in time, one point is earned. Otherwise, one point deducted.

3 Set the game duration; display and read out the final score at the end of the game.



4 The demo program.



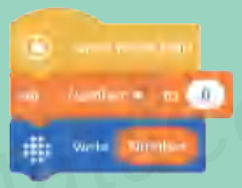
Bonus 1:Try to grab other numbers.
Bonus 2:Change the scoring method from pressing the button to clapping.

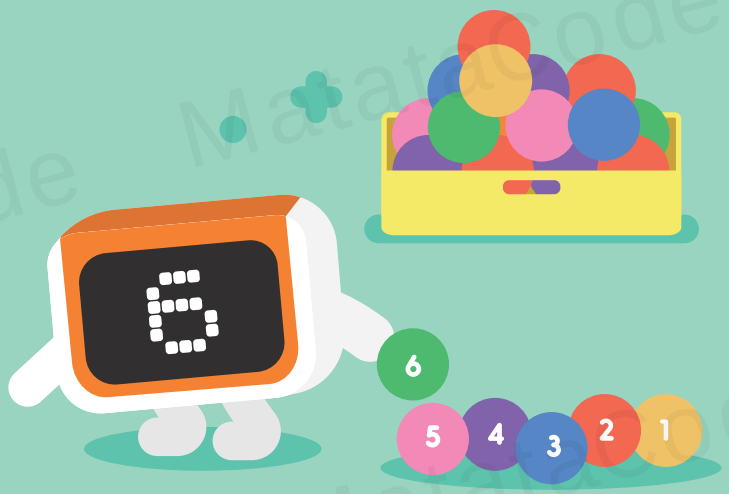
D11 The Counter

Variables

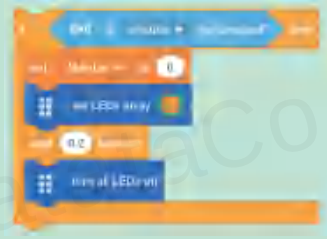
Task: VinciBot turns into a counter: press different buttons to increase, decrease, or reset the number.

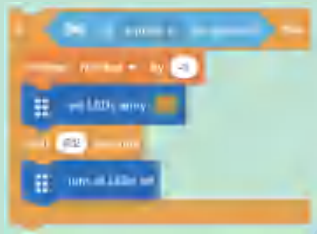
1 Create a new variable "number" as the number of counts.

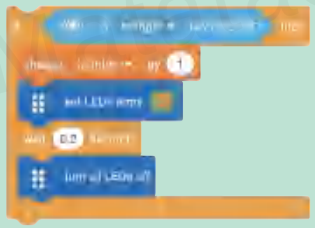




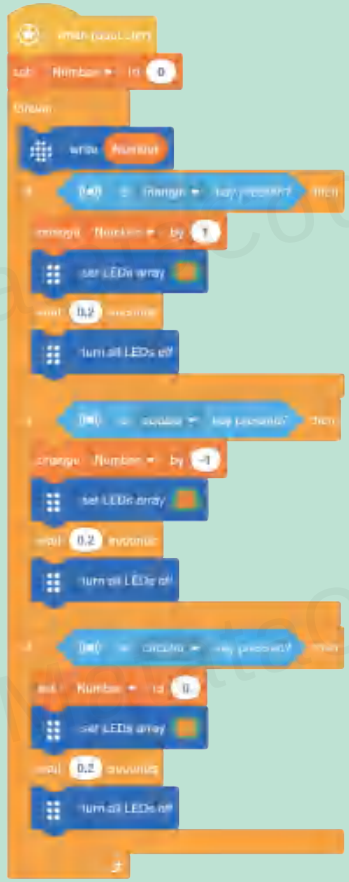
2 Set the conditions for triggering a change in quantity:Press different 3 keys to increase, decrease, and reset the number and display the number on the dot matrix screen. Also, add LED light effects when each key is pressed.







3 Implement counting function.



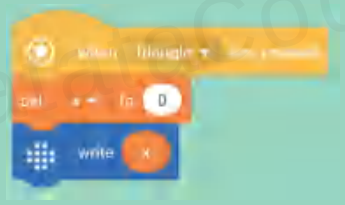
Bonus:Consider different counting scenarios and set up new counting trigger conditions, such as color counting, light counting, sound counting, etc.

D12 Stopwatch

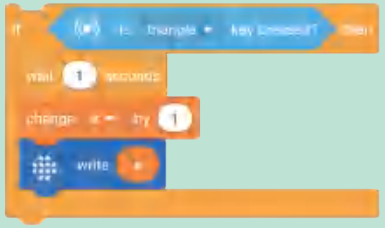
Task: VinciBot turns into a stopwatch timer:When condition A is triggered, the timing starts, and the timing ends when condition B is triggered.When triggering condition C, the time returns to 0.

Variables

1 Think about the function of a stopwatch, and create a new variable "x" to represent the timing period.

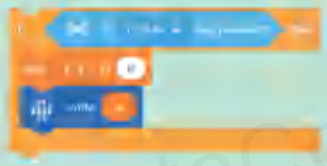


2 Set condition A to trigger the timing start (when the triangle button is pressed), and the interval time for the timing display.

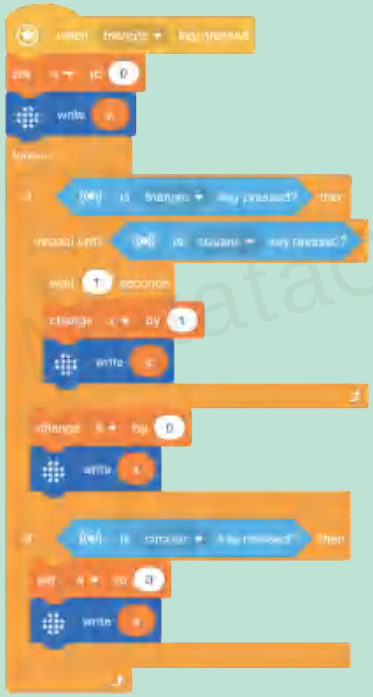


3 Apply the “repeat until” statement to the program, and set the timing to stop when another condition B (when the square button is pressed) is triggered.

4 Finally, set the trigger condition C (when the round button is pressed) for the timer to reset to 0.



5 The demo program.



D13 Speed Change by Color

Function
(multiple function)

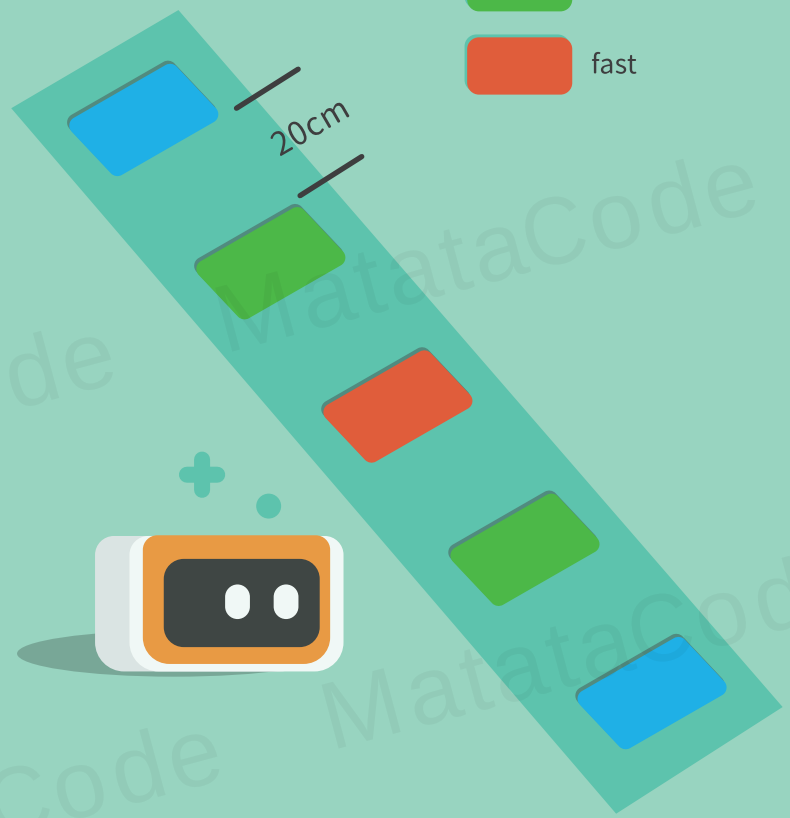
Task : Make a new block “Speed” , and find a way to add a parameter ” x” as an input to the new block. The parameter ” x” represents the change in speed. Program VinciBot to change speed when it detects different colors.

1 The blue, red, and green cards are placed at intervals within a row, and these color cards represent the shifting zones on the road.

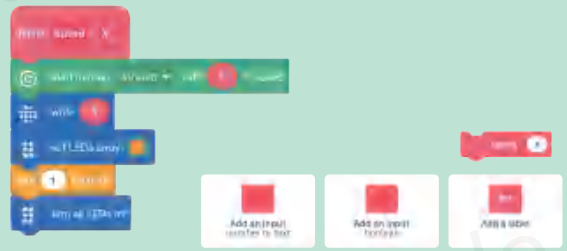
slow

medium speed

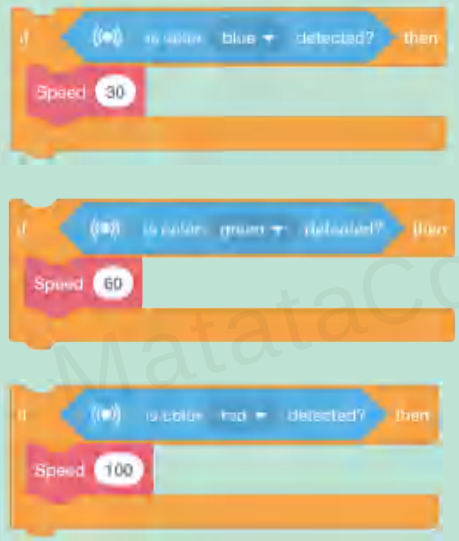
fast



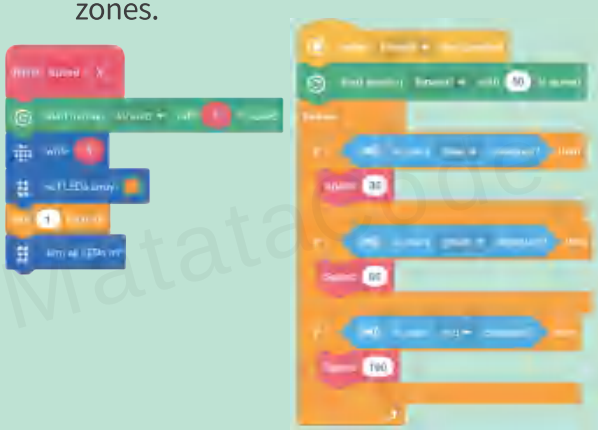
2 Make a new block "Speed", and set the parameters of "Speed". When the speed changes, flash the LED lights and display the current speed on screen.



3 Set the speed of VinciBot when passing through different colored areas.



4 Write a program that makes VinciBot pass through the road with the shifting zones.



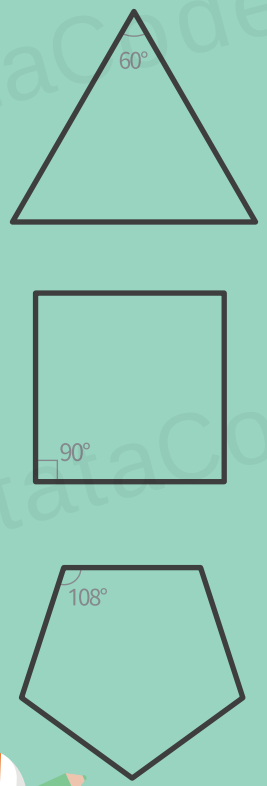
Bonus: Incorporate sound effects, LED lights, and other effects to VinciBot for when it passes through different shifting zones.

D14 Regular Polygons

Task: Add two parameters to the new block; the two parameters represent the "number of sides" and "angle". By modifying these two parameters, the VinciBot can draw various regular polygons.

Function
(multiple function)

1. In a regular polygon, each side has the same length and each interior angle has the same angle.



2. When VinciBot draws a regular polygon, the “number of sides” equals the number of repetitions, while the “angle” equals 180 degrees minus the angle of the interior angle. Make a new block, and add two parameters representing the "number of sides" and "angle".

3. Modify different parameters to help VinciBot draw more regular polygons.

Bonus:Can this method be used to draw a shape that it is not a regular polygon? Why or why not?

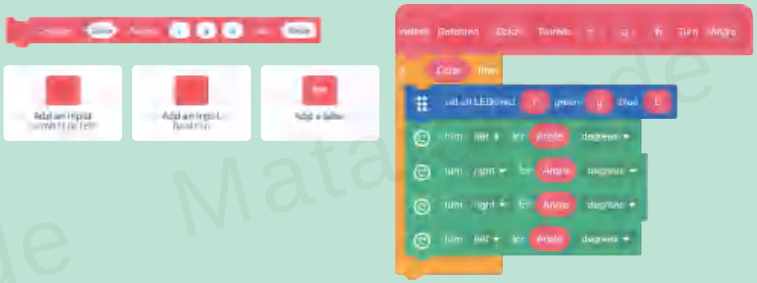
D15 Dancing VinciBot

Bonus:Add RGB parameters and judgment conditions to the new block, and program the VinciBot to make swinging movements of different amplitudes while displaying LED lights corresponding to different colors as they are detected.

Function
(multiple function)

1 Program VinciBot to perform different dances (swinging movements of different amplitudes) while displaying LED lights corresponding to different colors as they are detected.

2 Analyze the new blocks and identify how to modify the parameters so that VinciBot can perform different dances (swinging movements of different amplitudes) and display the corresponding LED lights when different colors are detected. Particularly,  focus on how the color of the LED lights are changed by setting the RGB parameter values.



3 The demo program.

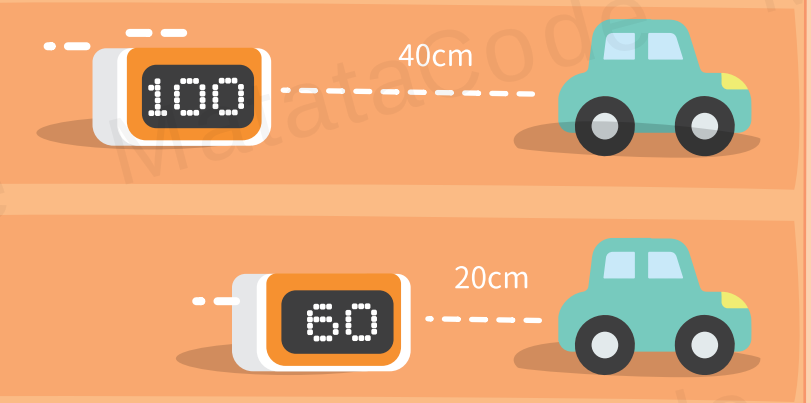
Number	Concept	Activity name
E-1	Conditional (Nested if else)	Smart Cruise
E-2	Conditional (Nested if else)	VinciBot Radar
E-3	Conditional (Nested if else)	Light Chaser 2.0
E-4	Conditional (Nested if else)	Light Chaser 3.0
E-5	Conditional (Nested if else)	Spirometer
E-6	Infrared communication	Traffic Statistics
E-7	Infrared communication	Power Supply
E-8	Infrared communication	Heart to Heart
E-9	Infrared communication	Dance for Two
E-10	line following	Line Following I (Part A)
E-11	line following	Line Following I (Part B)
E-12	line following	Line Following II (Part A)
E-13	line following	Line Following II (Part B)
E-14	line following	Line Following III (Part A)
E-15	line following	Line Following III (Part B)

E1 Smart Cruise

Task :Learn to utilize the “nested if else” statement, and program VinciBot to detect the distance from a vehicle (obstacle) in front, causing it to automatically change its running speed.

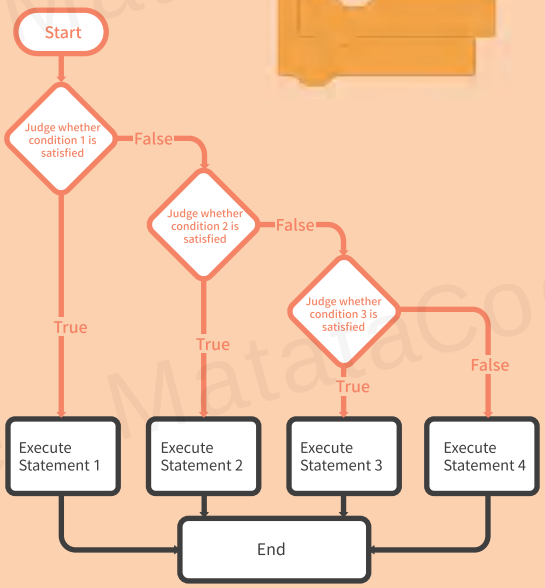
Conditional
(Nested if else)

1 VinciBot detects the distance from the vehicle in front (obstacle); the closer it is to the vehicle (obstacle) in front, the slower the speed, and vice versa.

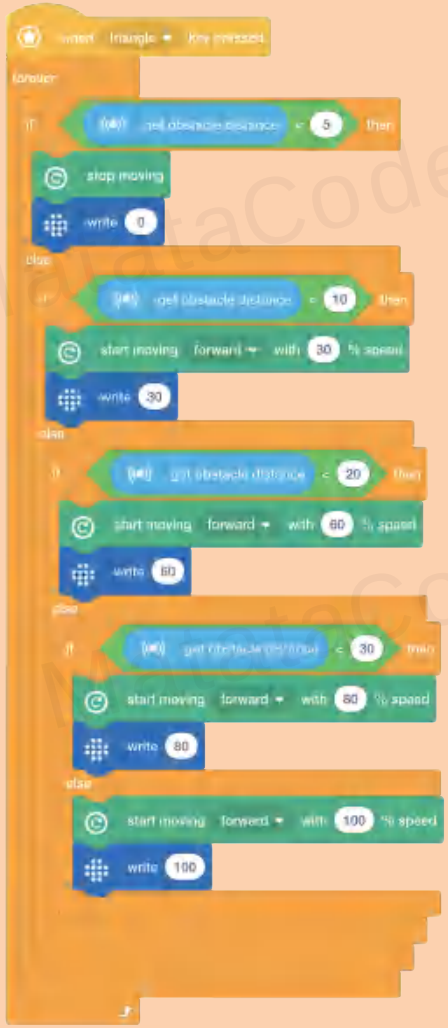


2 Set the movement speed of VinciBot at different distances from the vehicle (obstacle) in front, and display it on the dot matrix screen.

3 In order to change speed in real time, VinciBot needs to constantly detect the distance to the obstacle. To do this, not only does "forever" need to be used, but the “nested if else” statement is also required to program.



4 The demo program.

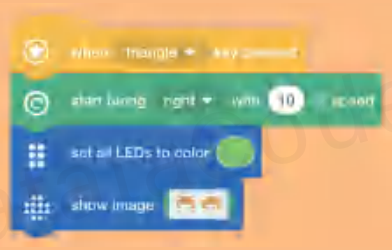


E2 VinciBot Radar

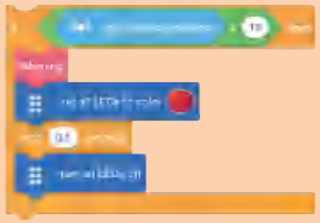
Task: VinciBot simulates radar to conduct a 360° rotating patrol. When an unknown object (obstacle) is found, it will issue different alarms based on its distance from the object.

Conditional
(Nested if else)

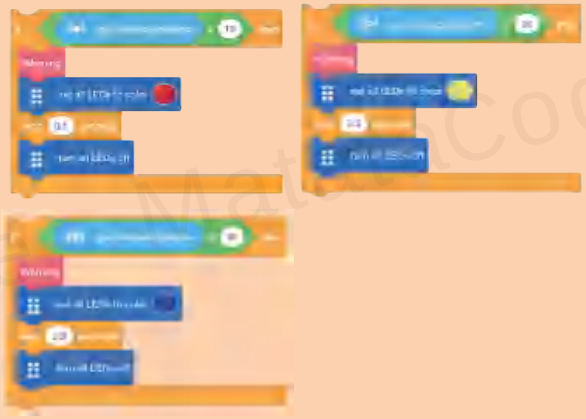
1 Set the rotation speed of VinciBot when it patrols.



2 Make a new “alarm” block to define the alarm state of VinciBot when it detects an unknown object.



3 Set the frequency of the alarm sound and light flashing when VinciBot finds unknown objects (obstacles) at different distances: the closer the distance is, the higher the frequency, and vice versa.



4 Write a program using “nested if else” statements to allow VinciBot to simulate radar patrols.

E3 Light Chaser 2.0

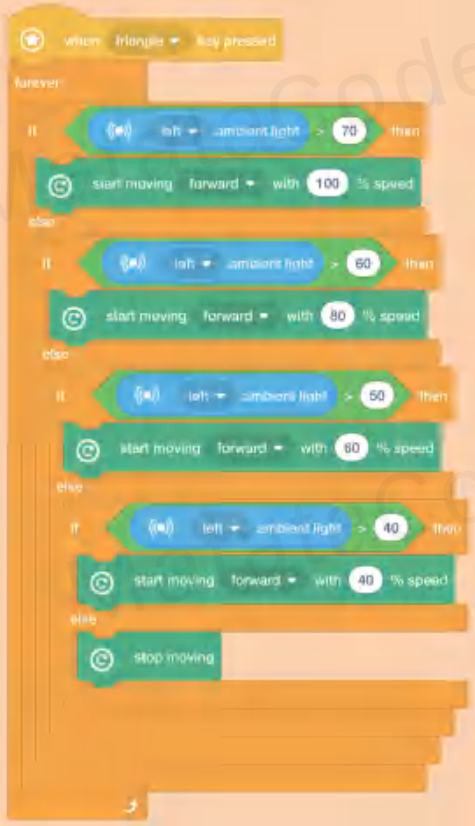
Task : VinciBot will change its forward speed to correspond to changes in ambient light intensity; the stronger the light, the faster the speed, and vice versa.

Conditional
(Nested if else)

1 VinciBot detects ambient light of different intensities: when the light source is closer to VinciBot, the ambient light is stronger; when the light source is more distant, the ambient light is weaker.

2 Set the forward speed of VinciBot under different ambient light intensity values.

3 Write a program using “nested if else” statements so that VinciBot moves forward at real-time speed based on ambient light intensity.



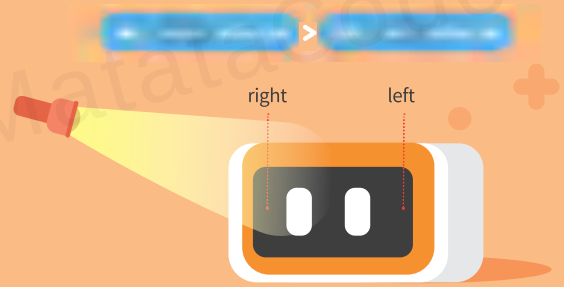
Bonus: Added a function to make VinciBot move backward according to the ambient light intensity.

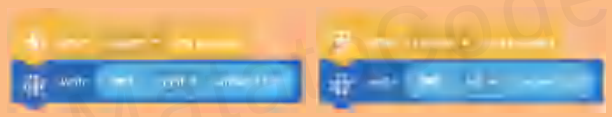
E4 Light Chaser 3.0

Task : Program VinciBot to turn left or right or move forward according to a light source when it detects changes in ambient light to the left or right.

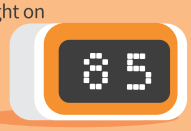
Conditional
(Nested if else)

1 When a light source is to the left side of VinciBot, the ambient light intensity on the left is greater than the ambient light intensity on the right, and the opposite is true when the light source is on the right.





The value of ambient light on the left



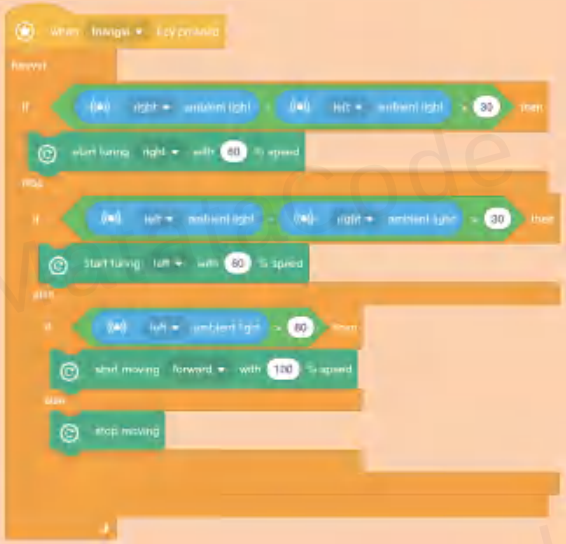
The value of ambient light on the right

The difference:85-50=35

2 Check the difference between the left and right ambient light values when the light source is on the left and right side (Because it is impossible for the left and right ambient light values to be completely equal in reality, a difference can be set so that VinciBot will turn left or right).

3 Write a program so that when the light source is on the left, VinciBot keeps turning left to follow the light source; when the light source is on the right, VinciBot keeps turning right and to follow the light source.

4 When the light source is in front of VinciBot, it runs after the light.

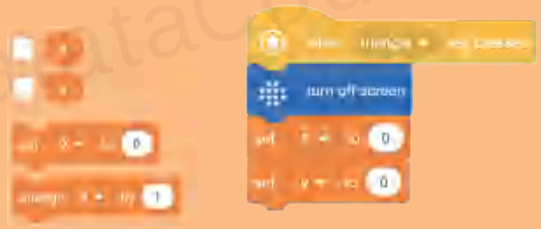


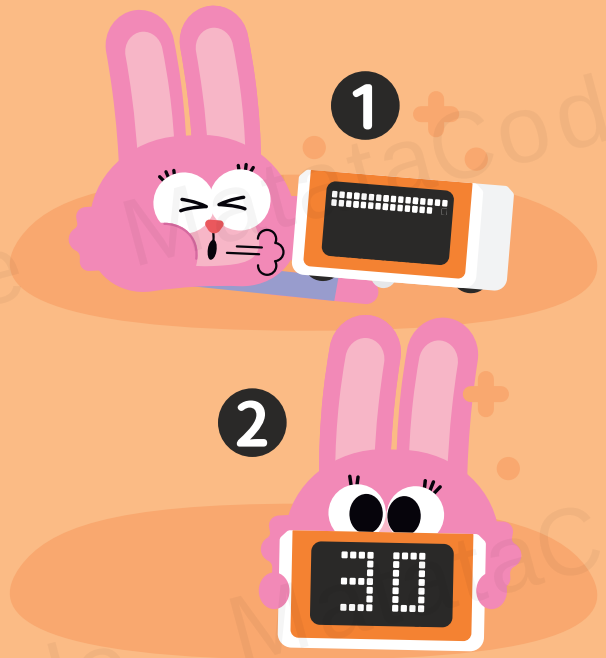
E5 Spirometer

Task :VinciBot is a spirometric tester: the longer you blow on VinciBot, the more pixel blocks will light up on the dot matrix screen.

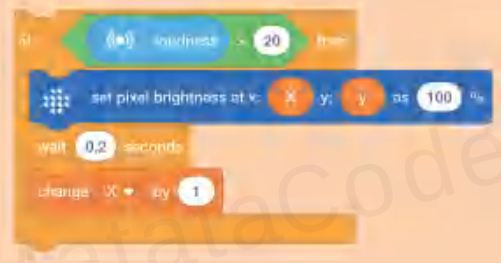
Conditional
(Nested if else)

1 Set two variables (x and y) to represent the coordinates of the pixel block on the 8*16 dot matrix screen. Prior to the test starting, the initial value is set to x= 0, y=0, and no pixel blocks are lit up.

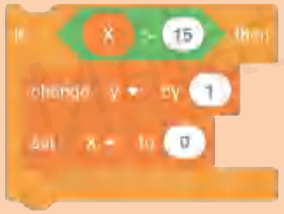




2 Set the conditions for triggering spirometry detection: when the intensity of blowing on VinciBot is greater than 20, the detection is triggered, and a pixel block lights up every 0.2 seconds.

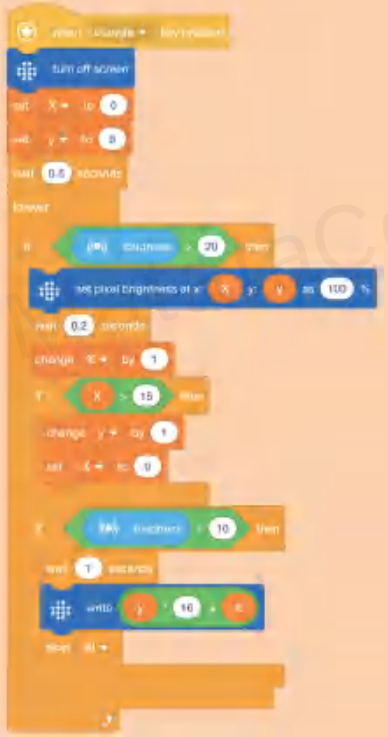


3 When the detection duration is long enough and the pixel blocks in the first row are all lit up, that is, x>15, a new row will begin, that is, x is reset to 0, and y is increased by 1.



4 When it is detected that the blowing intensity is less than 10, the detection is over, and the value of vital capacity is displayed on the dot matrix screen.

5 The demo program.

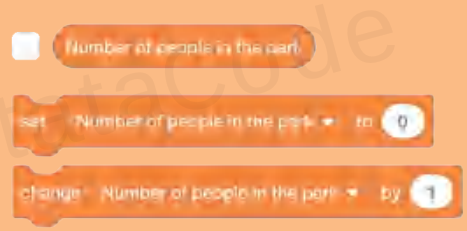


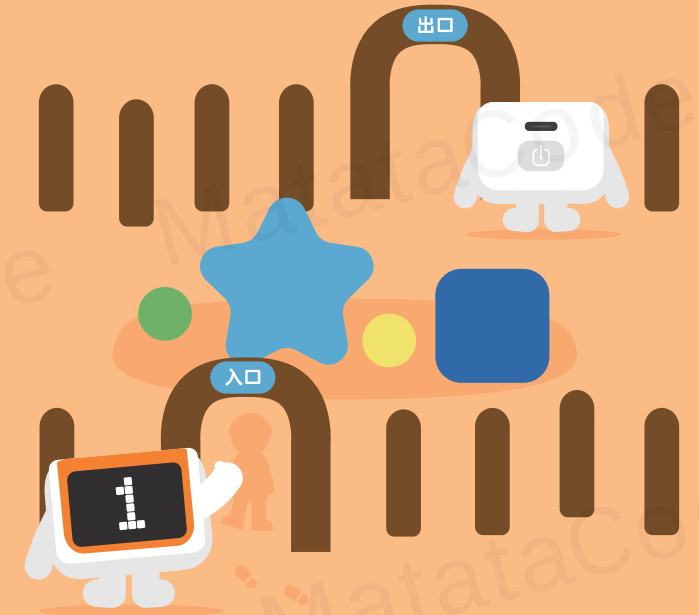
E6 Traffic Statistics

Task: Two VinciBots cooperate to detect the number of people in the park. VinciBot A detects the number of people entering the park at the entrance, and VinciBot B detects the number of people exiting the park at the exit. Apply the infrared communication to count the real-time number of people in the park.

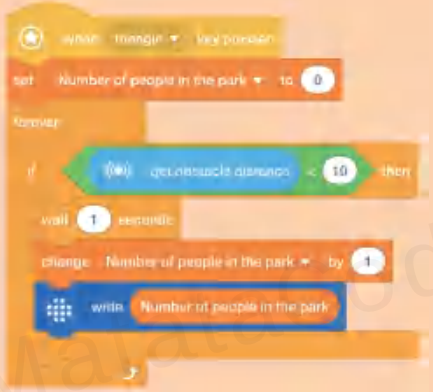
Infrared Communication

1 Create a new variable "number of people in the park" to represent the real-time number of people in the park.

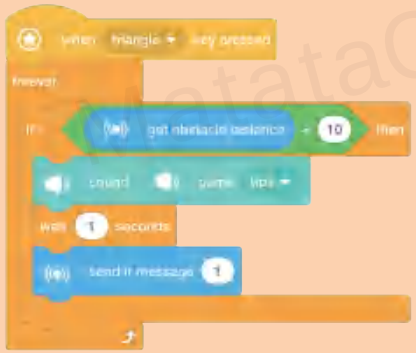




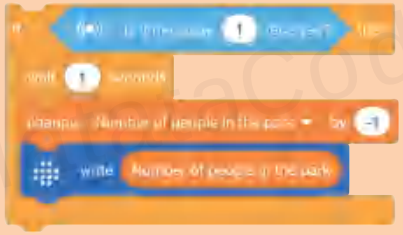
2 When VinciBot A (at the entrance) detects that someone has entered, the total number of people in the park will be counted +1.



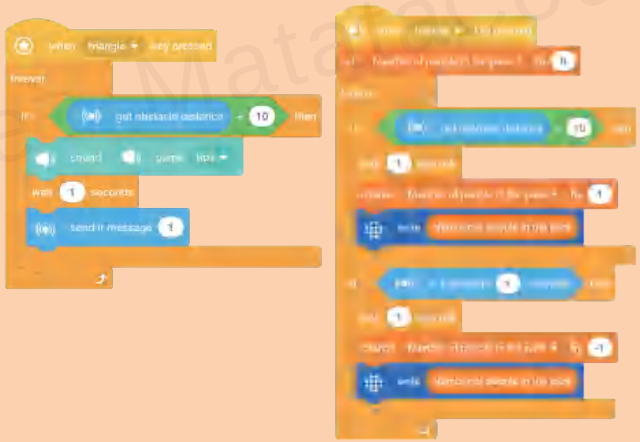
3 When VinciBot B (at the exit) detects that someone has exited, it will make a sound and send an infrared message to inform VinciBot A at the entrance.



4 After VinciBot A, at the entrance, receives the infrared message, the total number of people in the park will decrease (-1).



5 The demo program.

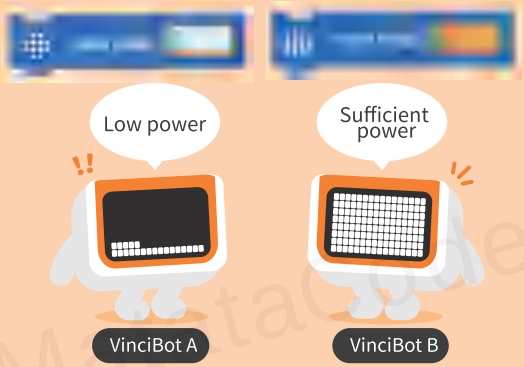


E7 Power Supply

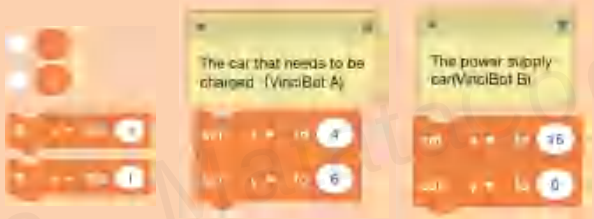
Infrared Communication

Task: There are two VinciBots. One represents a car that needs to be charged, and the other represents the power supply car; transmit messages through infrared, and display the power changes of the two VinciBots in real time on the dot matrix screen to simulate the process of power replenishment.

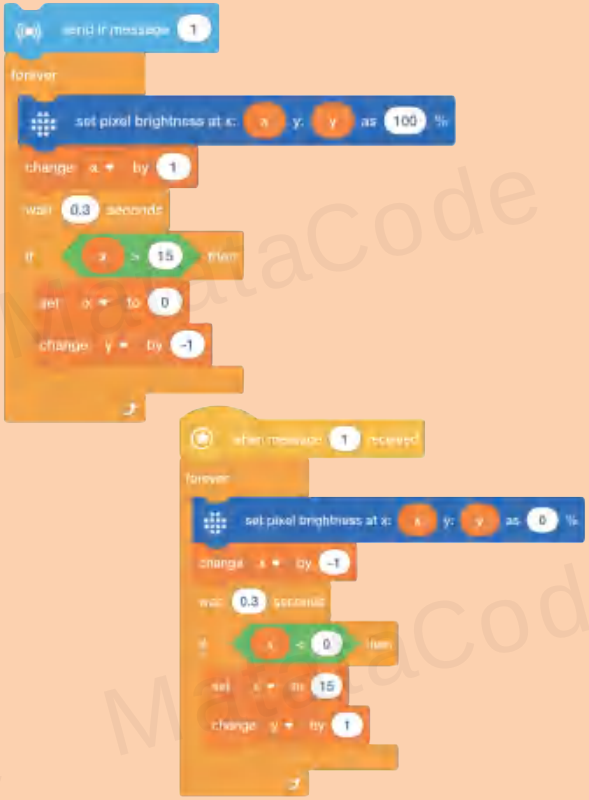
1 VinciBot A represents the car that needs to be charged, and the pixel blocks on the dot matrix screen show that power is low; VinciBot B represents the power supply car, and the pixel blocks on the dot matrix screen show that the power is sufficient (The illustration is shown below).



2 Create variables (x, y) to represent the pixel blocks' coordinates of the power value, and set the initial values.



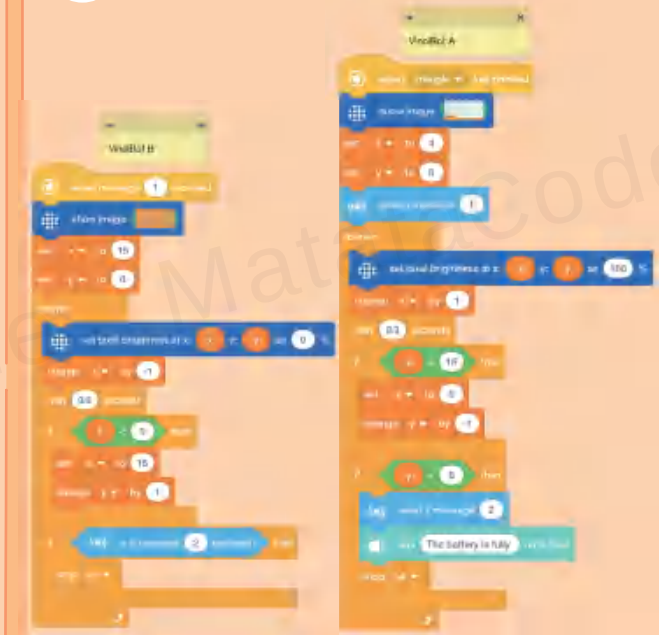
3 When VinciBot A starts charging, it sends an infrared message to VinciBot B, and VinciBot B receives the message and starts to transmit power.



Note: When applying infrared transmission, the two VinciBots need to be facing each other or placed in front of each other.

4 When VinciBot A is fully charged, it stops charging and sends an infrared message to tell VinciBot B. VinciBot B stops sending the power supply after receiving the message.

5 The demo program.

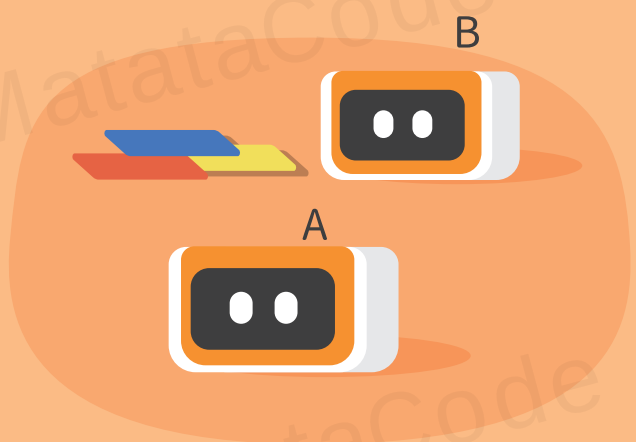


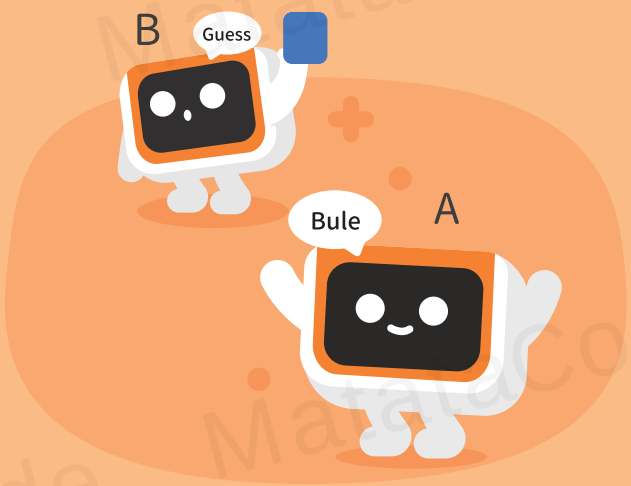
E8 Heart to Heart

Task: Using infrared communication to transmit messages, allow two VinciBots to complete a color-guessing game.

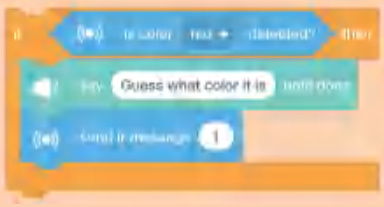
Infrared Communication

1 VinciBot A is placed in front of VinciBot B with its back is facing towards VinciBot B. Some color cards are prepared for VinciBot B to detect.

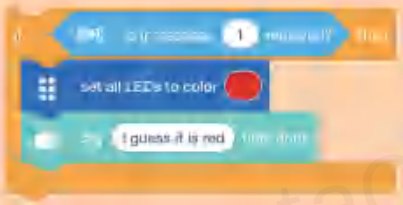




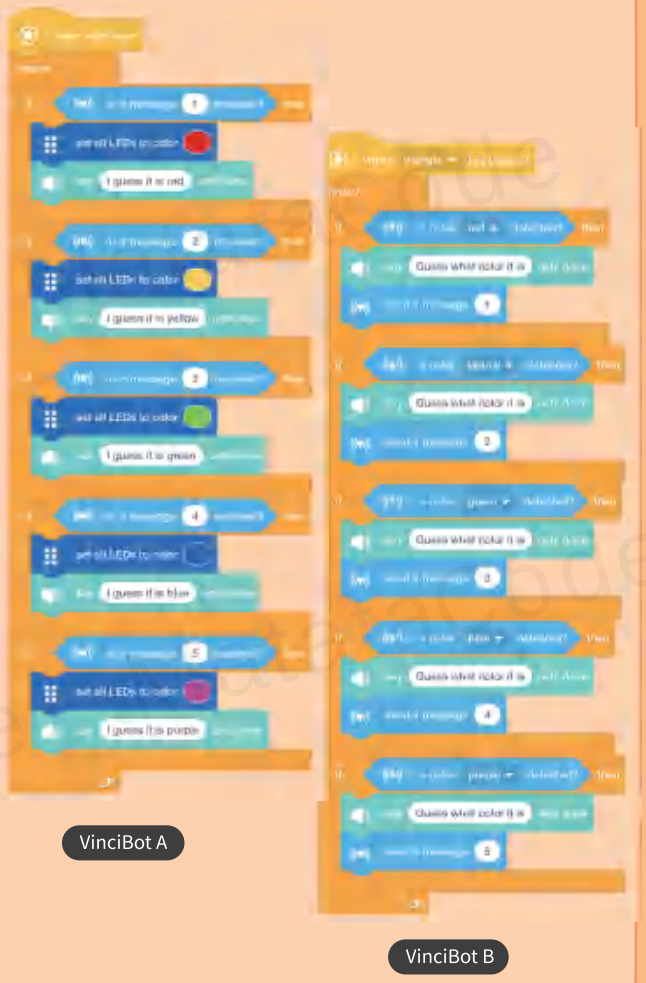
2 VinciBot B detects the color and asks “Guess, what is the color?” while sending infrared information to VinciBot A.



3 When VinciBot A receives the infrared message, it will answer with the color detected by VinciBot B and turn on the corresponding LED lights.



4 The demo program.

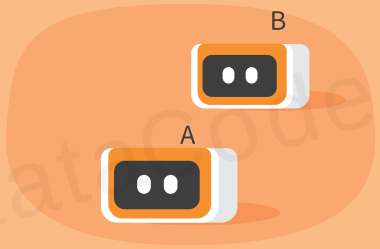


E9 Dance for Two

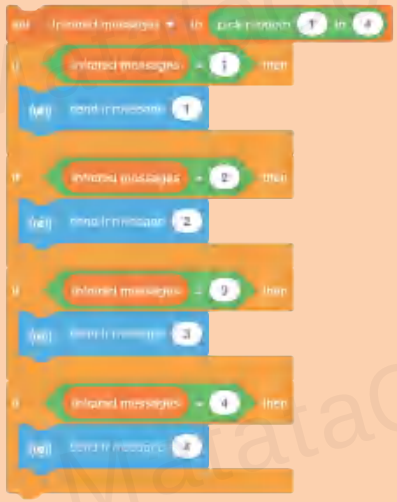
Task :Using infrared communication to transmit messages, allow two VinciBots to perform synchronized dances on the stage.

Infrared Communication

1 Set up VinciBot A and VinciBot B in a straight line, one after the other.



2 Create a variable "infrared message", set the variable to take a random value between 1 and 4, and send different infrared messages through different variable values (different infrared messages may be sent randomly).

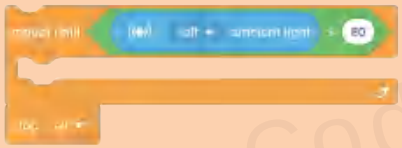


3 When VinciBot B sends an infrared message and starts dancing, VinciBot A receives the corresponding infrared message and performs the same dance simultaneously.

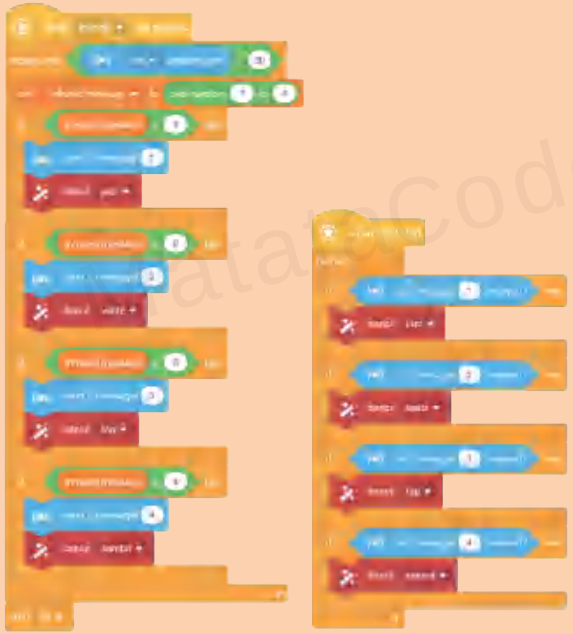
VinciBot A

VinciBot B

4 Set the condition for VinciBot B to stop dancing: when the ambient light intensity is greater than 80.



5 Apply loops to make the two VinciBots repeat synchronized dance performances.



VinciBot B

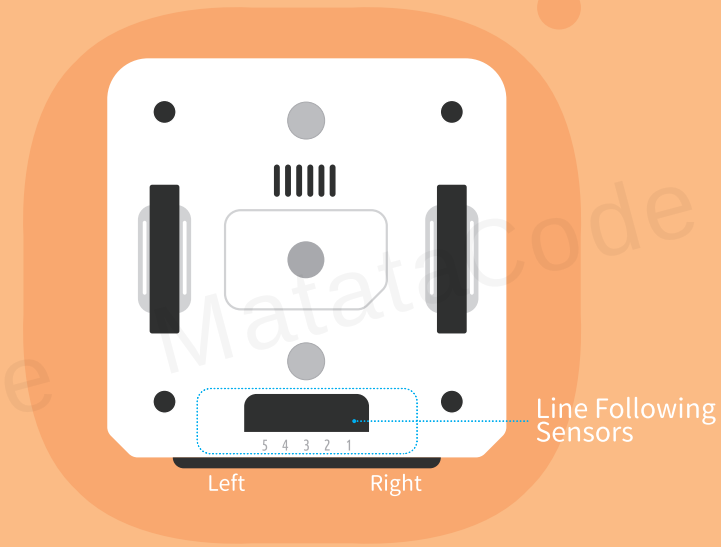
VinciBot A

E10 Line Following I (Part A)

Task: Learn how to test the value of the reflected light of the line following sensor, and how to apply the motor coding blocks. Prepare to write a line following program using the No. 1 and No.5 line following sensors.

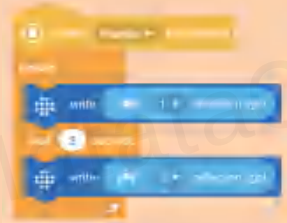
Line Following

1 There are five line following sensors underneath VinciBot, of which 1, 2, 4, and 5 are grayscale sensors, while 3 is a color sensor. They can all detect the reflected light intensity in black and white colors. When moving along wider lines, No.1 and No.5 line following sensors can be used to achieve line following.



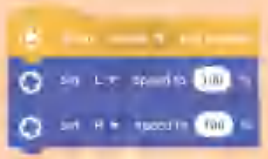
2 There are four situations in which VinciBot may patrol along a line using No. 1 and No.5 line follower sensors: completely on the line, completely off the line, or inclined (to the left or right of the line).

3 Write a program to test the values of reflected light corresponding to the No.1 and No.5 line following sensors in these four situations, and record them.



According to the principle of light reflection, black will absorb a light source, therefore the value of reflected light is relatively low; conversely, the value of white reflected light is relatively high.

4 Explore and try to set the parameters of the motor coding blocks ; run the program to observe how to make VinciBot move forward, backward, and turn left and right, and record the parameters.



L	R	State
-100	100	Move Forward
100	-100	Move Backward
-20	80	Turn Left
-80	20	Turn Right

E11 Line Following I (Part B)

Task: Write a line following program using the No. 1 and No.5 line following sensors.

Line Following

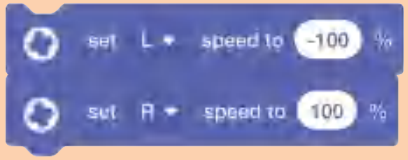
1 After testing, a rule for the left and right reflected light values in the four situations can be determined.

	Completely On the Line	Completely Off the Line	Inclined Left	Inclined Right
Left Reflected Light	<40	>80	Left > Right	Right > Left
Right Reflected Light	<40	>80		

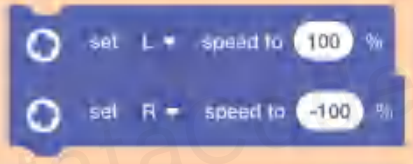
2 These four situations can be represented by four conditions.

3 For the four possible line following situations , VinciBot performs four actions, respectively.

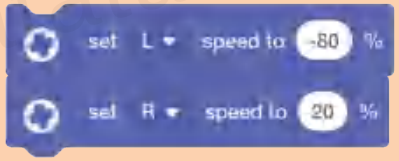
Move Forward



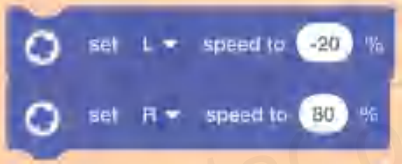
Move Backward



Turn Right



Turn Left



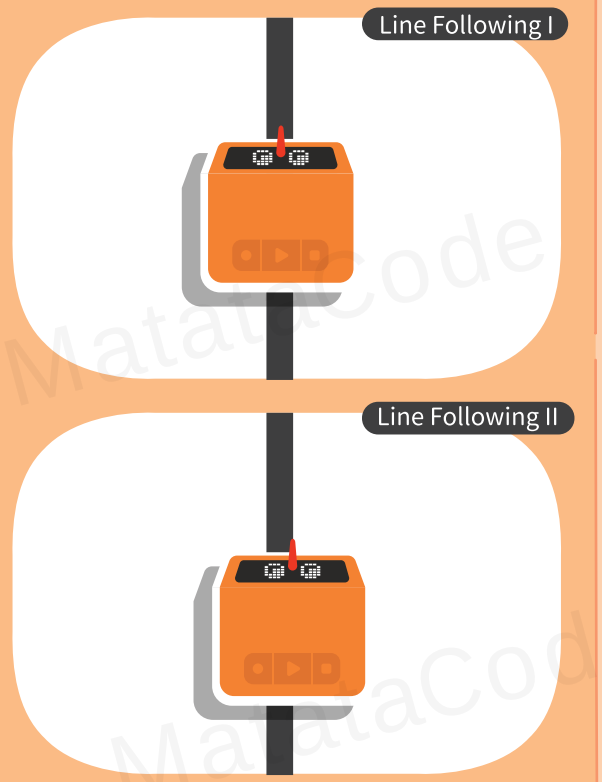
4 The demo program.

E12 Line Following II (Part A)

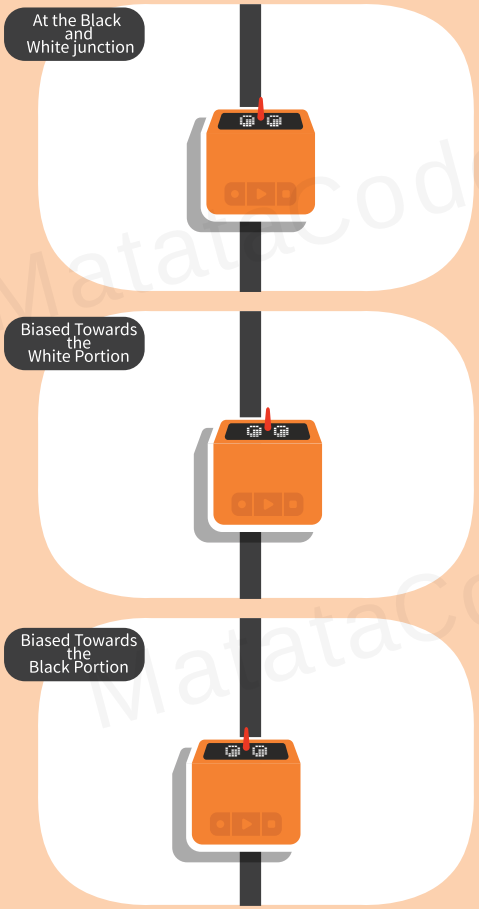
Task: Test the reflected light values of the No.3 line following sensor/color sensor under the three conditions on the patrol map.

Line Following

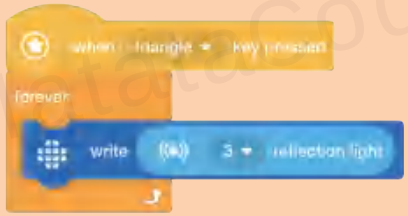
1. The No. 3 line following sensor is a color sensor. Unlike the line following program that uses the No. 1 and No.5 line follower sensors (that make VinciBot patrol along the middle of the line), when using the color sensor to patrol the line, VinciBot runs along the junction of black and white.



2. There are three situations for VinciBot to patrol along the junction line: one is at the junction; one is biased towards the white portion; the other is biased towards the black portion.



3. Write a program allowing the color sensor to measure several reflected light values in the three situations: when VinciBot is on black; when it is on white; and when it is at the junction.



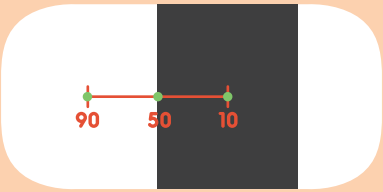
4. Test and record the values.

E13 Line Following II (Part B)

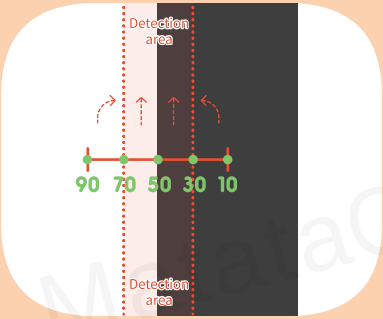
Line Following

Task: Apply the color sensor to write a program so that VinciBot follows along the line at the junction of black and white.

1 After testing, it may be determined that the reflected light value on the black line is about 10, while the value on the white line is about 90; therefore, it can be calculated that the value at the junction of black and white is about (10+90)/2=50.

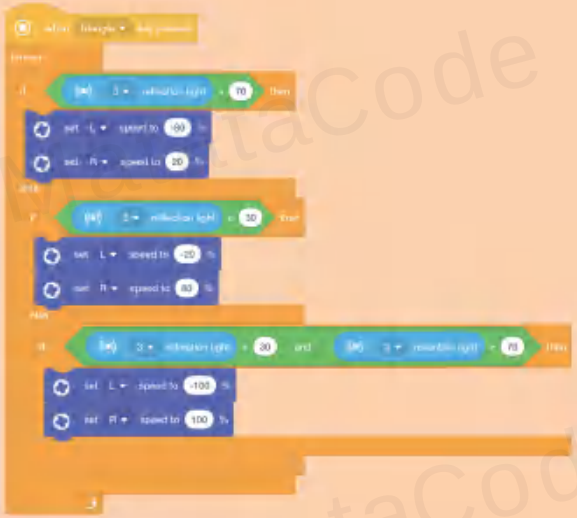


2 The black line can be divided into four parts (as shown below). VinciBot patrols the line in the area shown in the figure below; if it is outside the area, it should turn left or right.



3 When VinciBot deviates to the white part, that is, when the reflected light value is greater than 70, it needs to turn right. When VinciBot deviates to the black part, that is, when the reflected light value is less than 30, it needs to turn left. When VinciBot is moving in the two middle parts, that is, when the reflected light value is greater than 30 and less than 70, it will go straight.

4 The demo program.

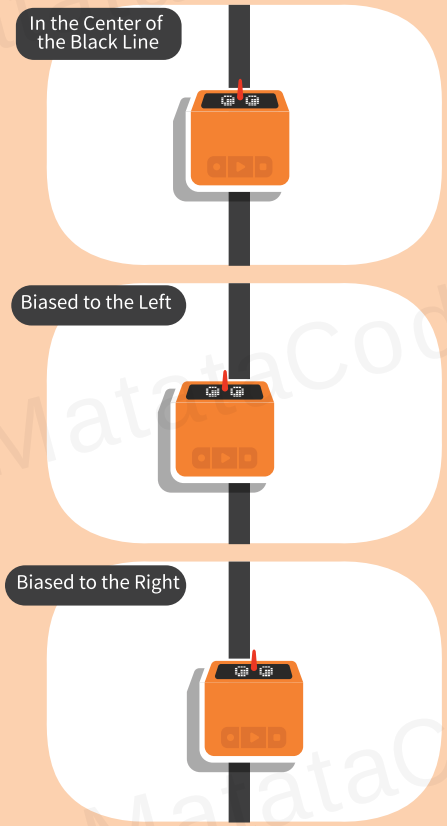


E14 Line Following III (Part A)

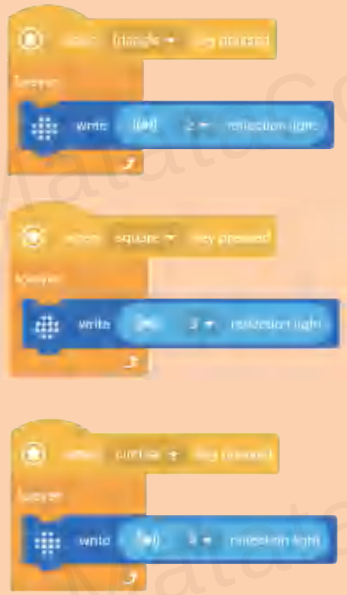
Task: Test the value of reflected light corresponding to the No. 2, 3, and 4 line following sensors in the three typical situations.

Line Following

1 There are also three situations in which VinciBot patrols the line using the No.2, No.3, and No.4 liner following sensors: one is in the center of the black line; one is biased to the left; the other is biased to the right.



2 Write a program to test the values of reflected light corresponding to the No.1 and No.5 line following sensors in these 3 situations: when the VinciBot is in the middle of the black line; when it is at the junction of black and white on the left; and when it is at the junction of black and white on the right.



3 Test record the values.

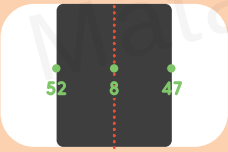
	Number 2 Line Follower Sensor	Number 3 Line Follower Sensor	Number 4 Line Follower Sensor
In the Middle of the Black line			
At the Junction of Black and White on the left			
At the Junction of Black and White on the Right			

E15 Line Following III (Part B)

Task: Write a line following program using the No. 2, No.3, and No.4 line following sensors.

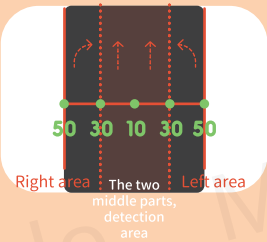
Line Following

1 After testing, it may determined that the values of the three sensors are very close. Take the No. 3 sensor as an example: the reflected light value is about 10 when VinciBot is in the middle of the black line, and the reflected light value is about 50 when the black and white junctions are on the left and right sides.



	In the Middle of the Black Line	At the Junction of Black and White on the Left	At the Junction of Black and White on the Right
No.3 line following sensor	8	47	52

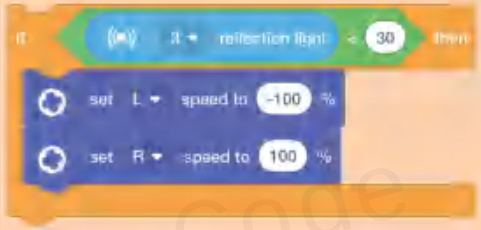
2 The black line can be divided into four parts as shown below. VinciBot patrols the line in the two middle areas; if it deviates to the left or right area, it will turn right or left to straighten: the reflected light value of the rounded calculation area is about 10-30. When the reflected light of the No.3 sensor is less than 30, VinciBot moves forward. When the No.2 sensor is less than 30, VinciBot deviates to left, and it needs to turn right. When the No.4 sensor is less than 30, VinciBot deviates to the right, and it needs to turn left.



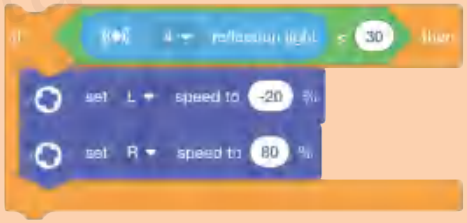
Note: The range of the detection area can be adjusted according to the actual situation, and the value may also fluctuate accordingly. In this case, only the average value is used.

3 According to the analysis, write the three programs that make VinciBot move forward, turn left, and turn right on the line.

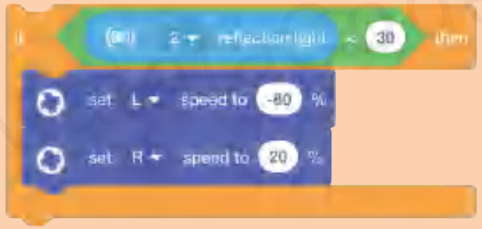
Move forward



Turn left



Turn right



4 The demo program.