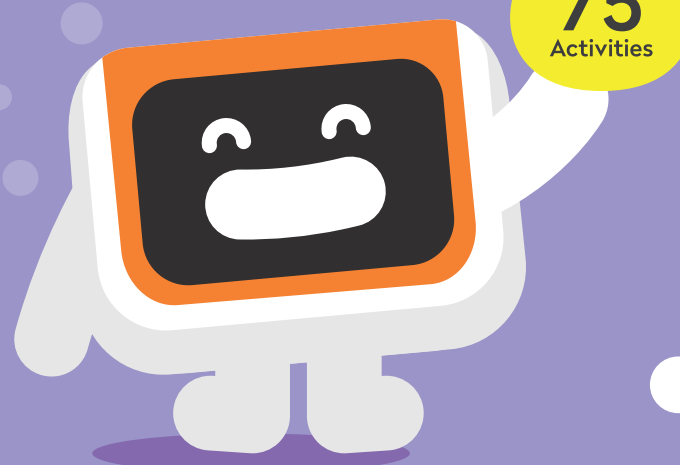
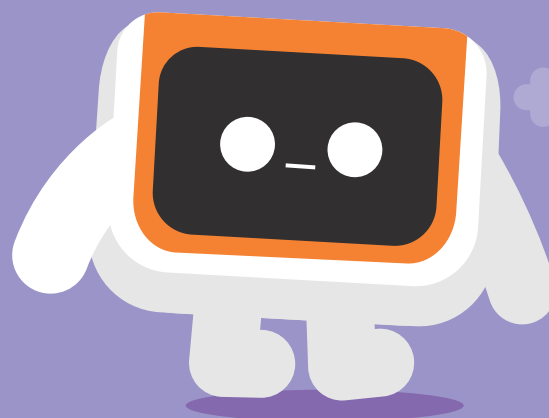


matata
studio

8+
STEAM

Activity Cards

for VinciBot



Core Activities

Graphical programming

A1 Hello, VinciBot!

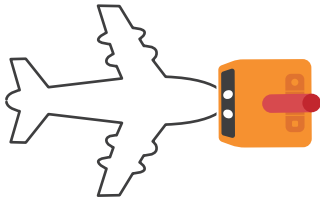


Task : Get familiar with the structure, functions and characteristics of VinciBot by exploring the three preset modes of VinciBot.

- 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.

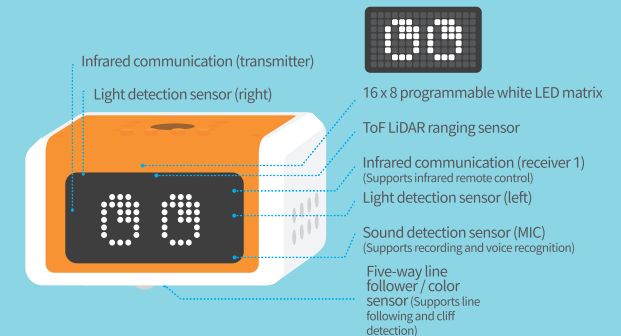


- 2 Explore the three preset modes of VinciBot, and choose its functions or characteristics.

- ☐ 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



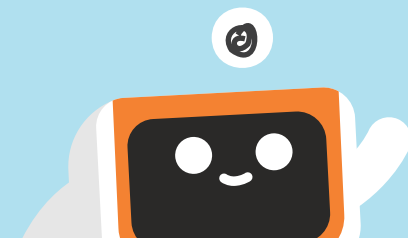
Connect via Bluetooth



- 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.



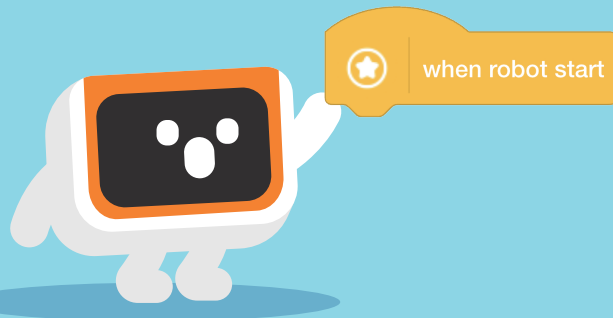
A3 Nice to Meet You

Sequences

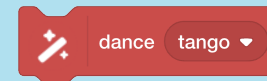
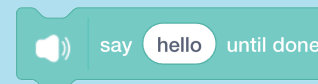
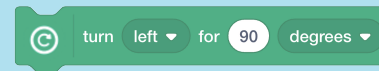
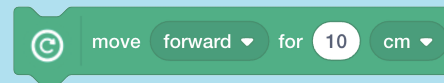


Task :Familiarize with the motion, sound, and effect blocks; program VinciBot to walk up to a toy, say hello to it, and dance.

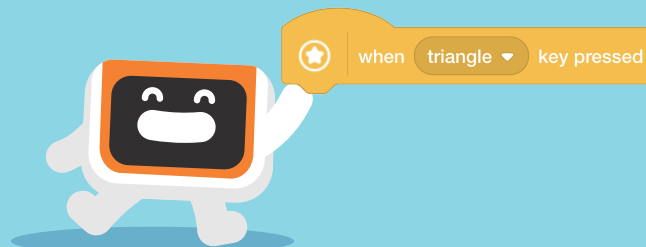
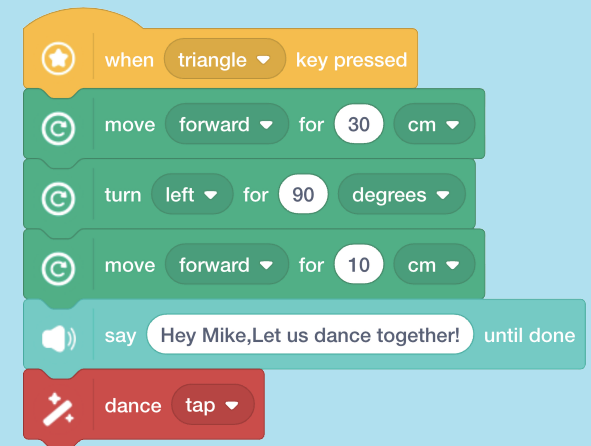
- 1 When writing a program, the first step is to choose an event block that starts the robot.



- 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



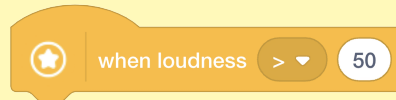
Sound

B1 VinciBot is Pretending to be Sleepy

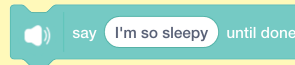


Task : Familiarize with the “new event” coding blocks. When VinciBot hears someone coming (i.e. making a sound), it will pretend to be sleepy, say "so sleepy", and make a "sleepy" sound. After waiting 5 seconds to confirm that the person has walked away, VinciBot will blink and light up to read a book.

- 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.

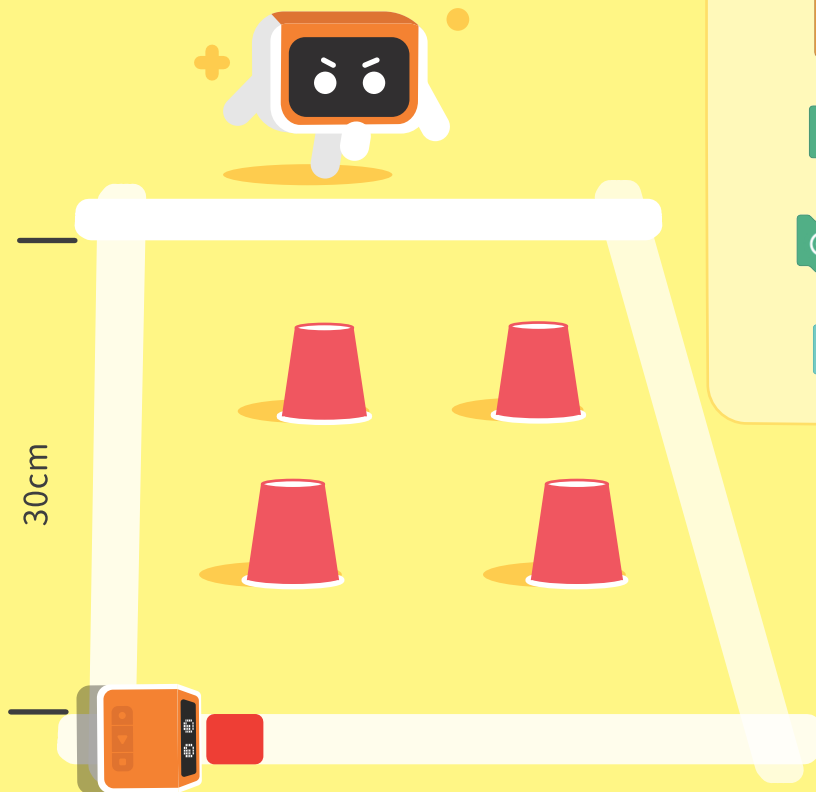


B2 Ready, Go!



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.

- 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.

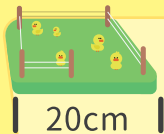


B3 Drive the Duck

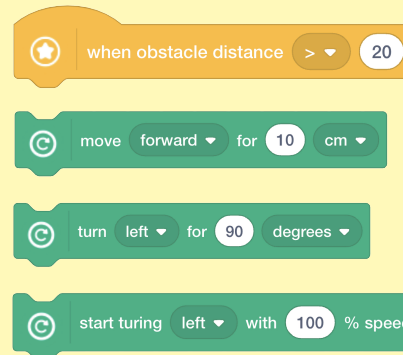


Task : VinciBot acts as a duckling that quacks and turns to the right and continues walking whenever it detects an obstacle ahead. Use the event block that detects the distance of obstacles to drive it to the duck house.

- 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.



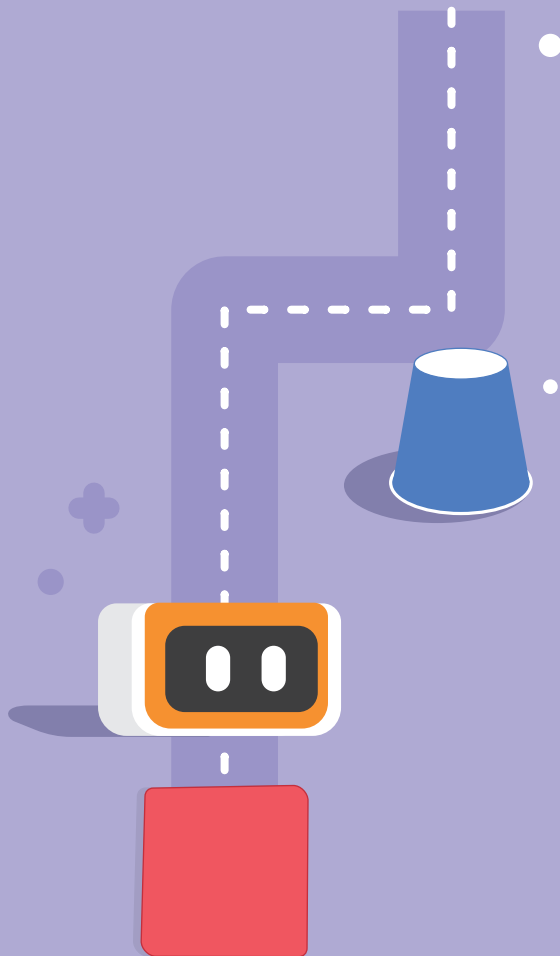
C1 Autopilot I

Conditionals
(wait until)

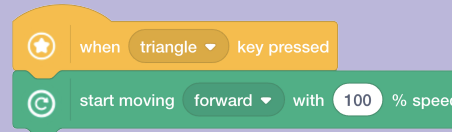


Task :Learn how to utilize the “wait until” statement in the conditional statement to allow VinciBot to automatically bypass obstacle(s) while running forward, and to stop when the red end point is detected.

- 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.



wait until

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?



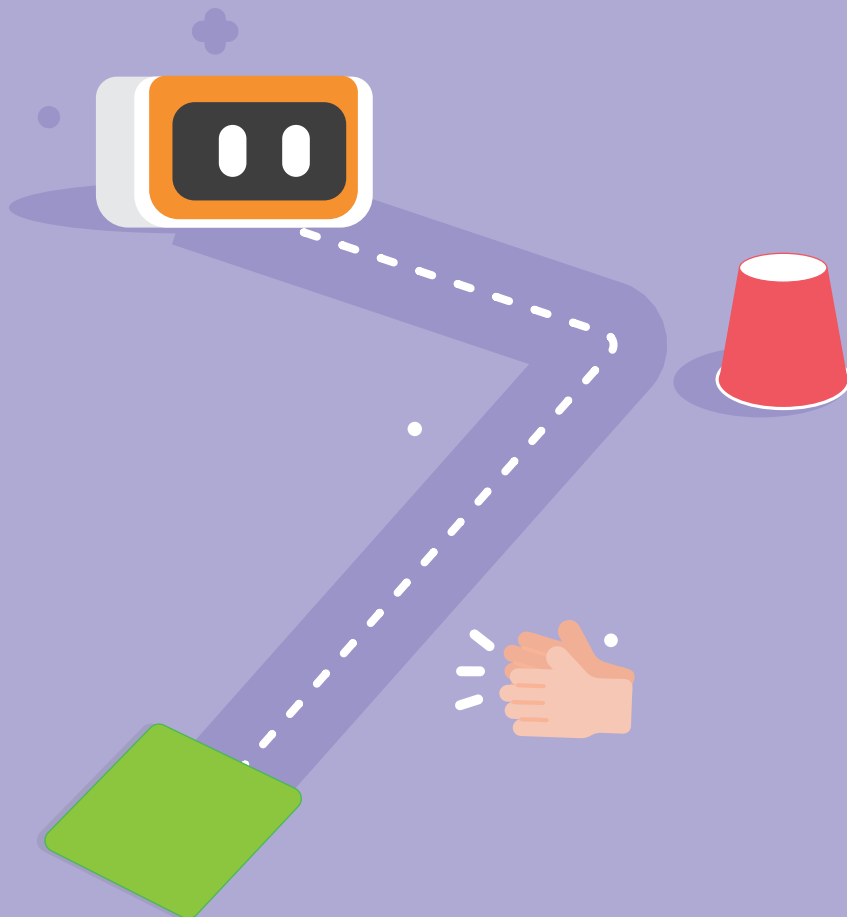
C2 Escape from the Chamber of Secrets

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.



```
start moving forward with 60 % speed
wait until (is obstacle distance < 10 ?)
turn right for 90 degrees
start moving forward with 60 % speed
```

When a sound is detected, VinciBot should speed up.



```
wait until (loudness > 50)
set all LEDs to color red
start moving forward with 100 % speed
```

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



```
wait until (is color green detected?)
stop moving
sound emotion yeah until done
```



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)

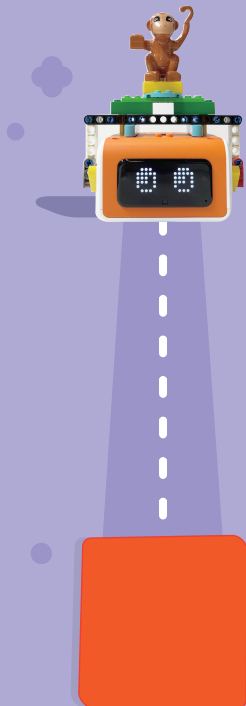


Task : Learn the “repeat until” statement, and use the “repeat until” coding block programming so that the VinciBot Parade Float makes facial expressions and sings while moving forward, and stops all actions when it reaches the red end point.

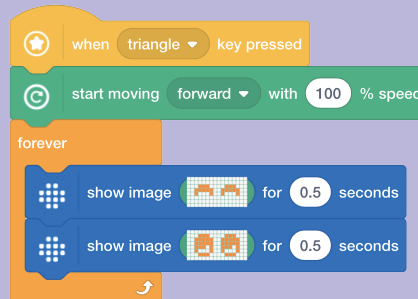
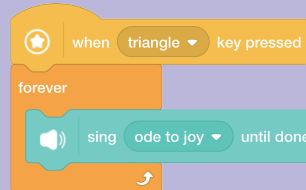
- 1 Dress up VinciBot as a parade float.



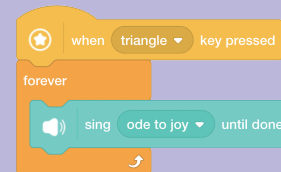
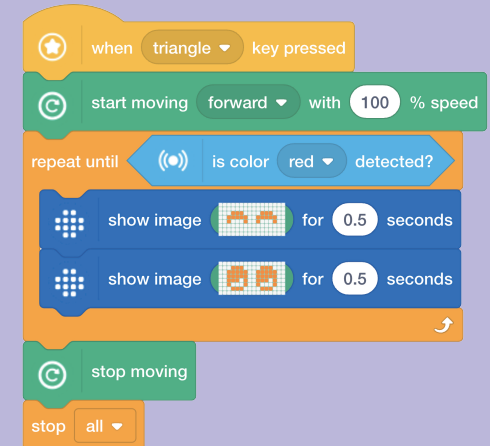
- 2 Set up the task scene: Place a red card on the side of a level table or surface as an end marker.



- 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?



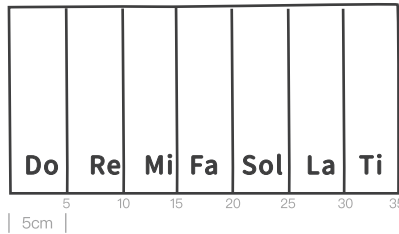
D1 The Magic Air Piano



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.

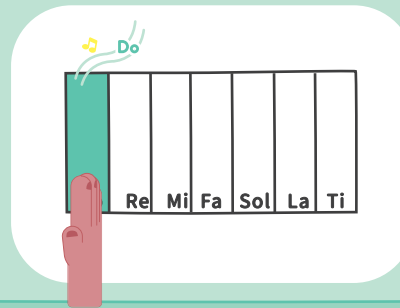
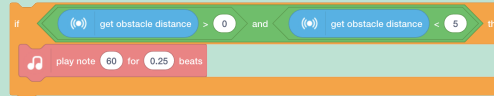


- 2 Learn the meaning and usage of the "AND, OR, and NOT" coding blocks.



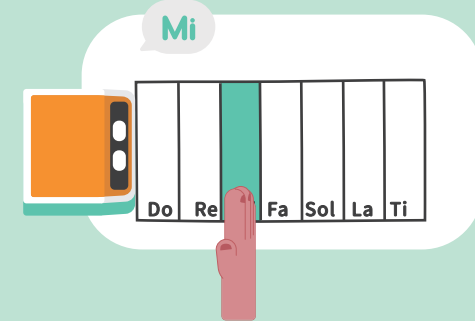
The operations of "AND, OR, and NOT", respectively, indicate when the conditions on both sides are satisfied at the same time ("AND"); when one of the conditions is satisfied ("OR"); when these conditions are not satisfied ("NOT"), execute the next command.

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

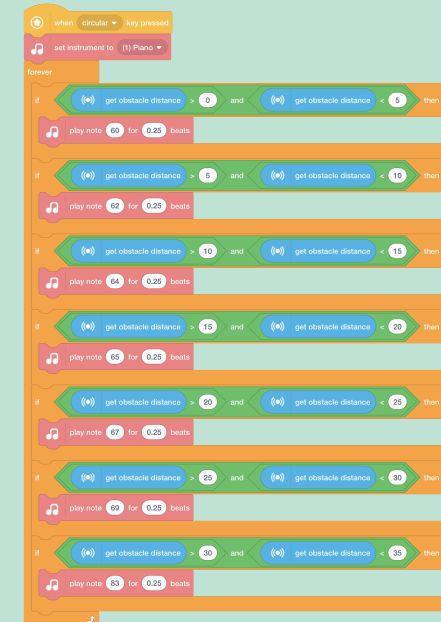


Bonus: When playing the air piano, the corresponding note should be displayed on the dot matrix screen.

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



- 5 The demo program.



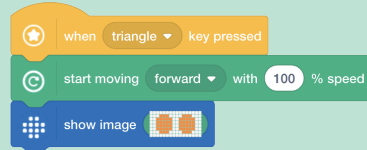
D2 Coward VinciBot

Conditionals (if else)

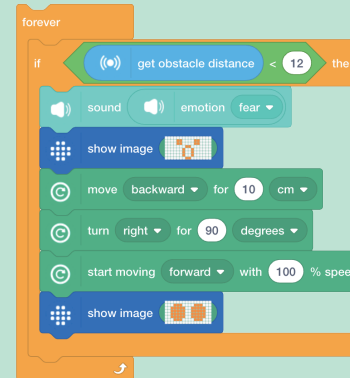


Task: Learn how to utilize the “if else” statement in conditional statements and the program flow diagrams. The Coward VinciBot has been walking forward with wide eyes. Whenever it encounters an unknown obstacle, it will move back out of “fear,” turn right, and then keep moving forward with wide eyes.

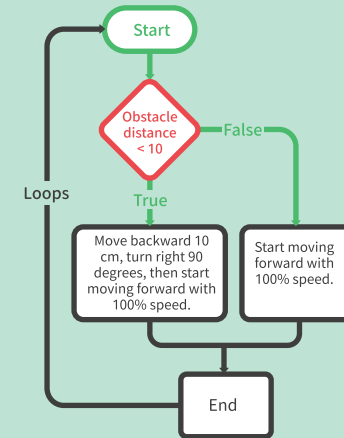
- 1 The VinciBot keeps moving forward, with a wide-eyed expression displayed on the dot matrix screen.



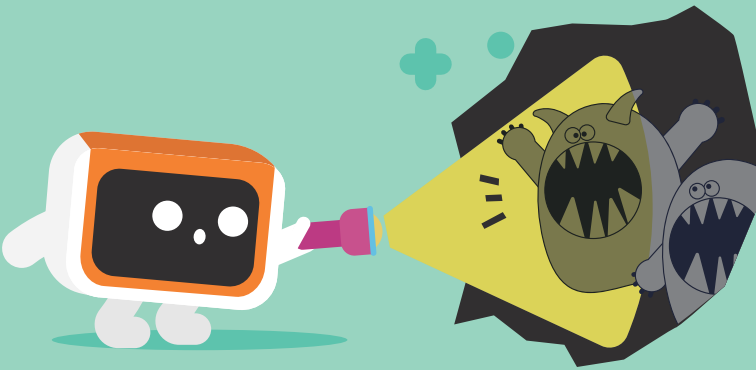
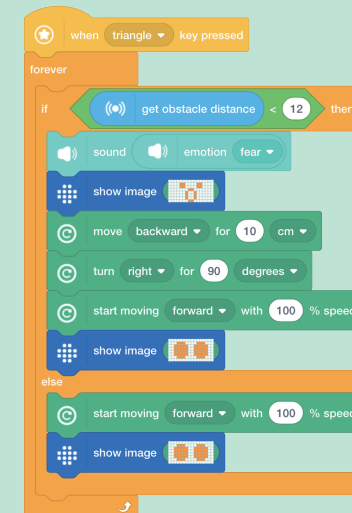
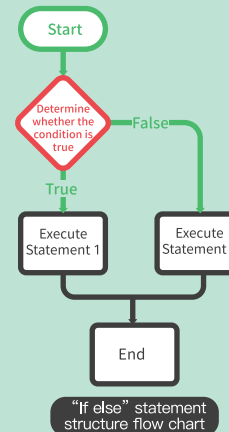
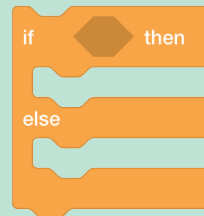
- 2 First, use the “if then” statement to program. If VinciBot encounters an obstacle, it will move backward for 10 cm while making a “fear” sound and a frightened expression; then it will turn right and continue to move forward with wide eyes.



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



- 3 Learn to utilize the “if else” statement.



D3 Close Friends



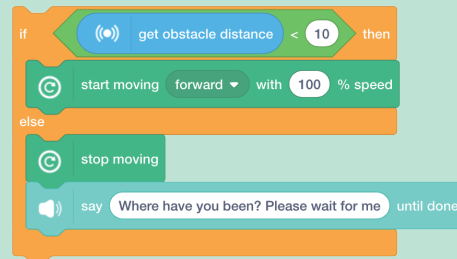
Task: Use the ToF ranging sensor to make VinciBot follow the little bear. When the little bear is suddenly picked up, VinciBot stops moving and asks "Where have you been?" Then when the little bear comes back, VinciBot continues to follow it.

Conditionals
(if else)

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

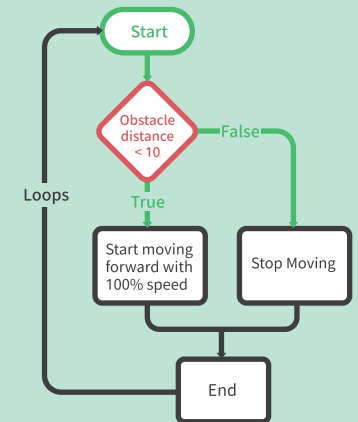


- 2 When VinciBot is within a certain distance of the bear, make it approach the bear, moving forward slowly. VinciBot will stop and say "Where have you been? Please wait for me".



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.

E1 Smart Cruise

Conditional (Nested if else)

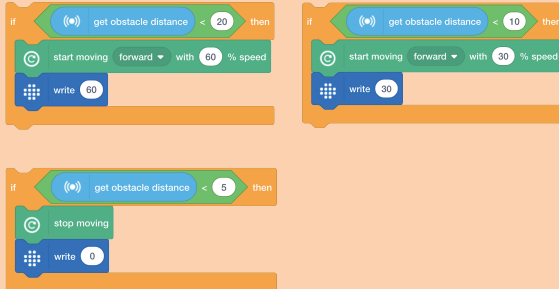


Task :Learn to utilize the “nested if else” statement, and program VinciBot to detect the distance from a vehicle (obstacle) in front, causing it to automatically change its running speed.

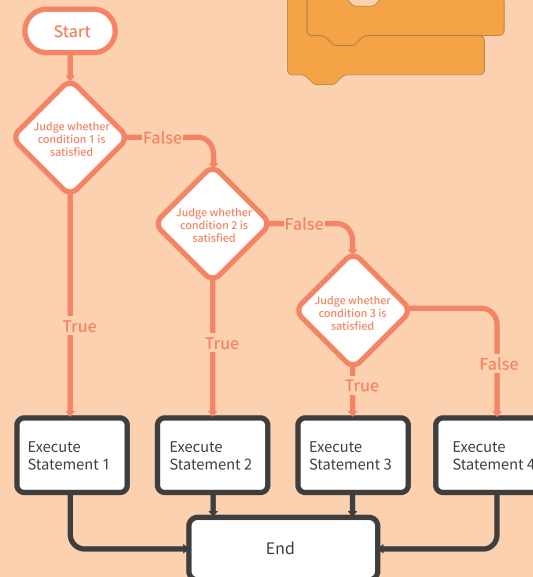
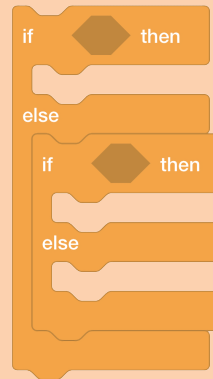
- 1 VinciBot detects the distance from the vehicle in front (obstacle); the closer it is to the vehicle (obstacle) in front, the slower the speed, and vice versa.



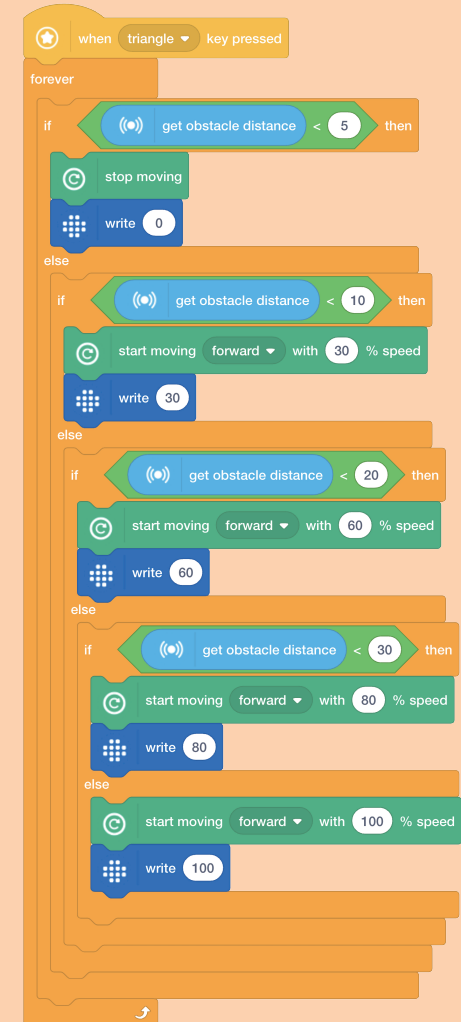
- 2 Set the movement speed of VinciBot at different distances from the vehicle (obstacle) in front, and display it on the dot matrix screen.



- 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.



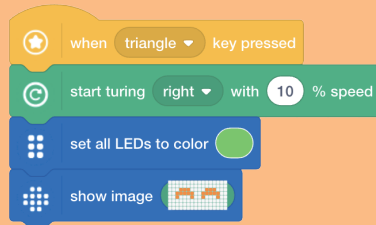
E2 VinciBot Radar



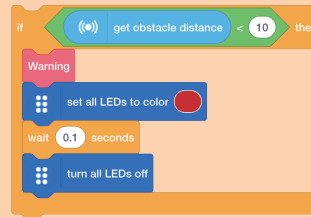
Task: VinciBot simulates radar to conduct a 360° rotating patrol. When an unknown object (obstacle) is found, it will issue different alarms based on its distance from the object.

Conditional
(Nested if else)

- 1 Set the rotation speed of VinciBot when it patrols.



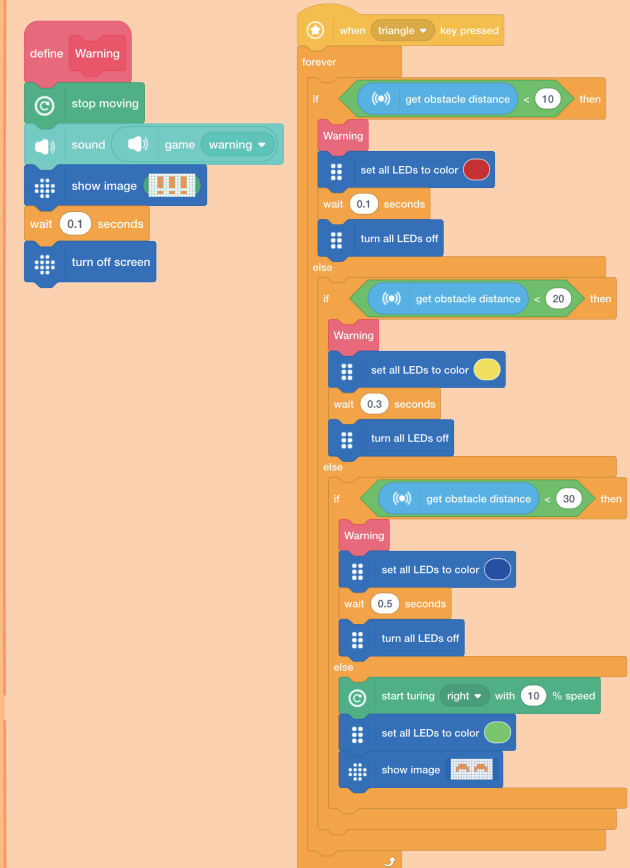
- 2 Make a new “alarm” block to define the alarm state of VinciBot when it detects an unknown object.



- 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

Conditional
(Nested if else)

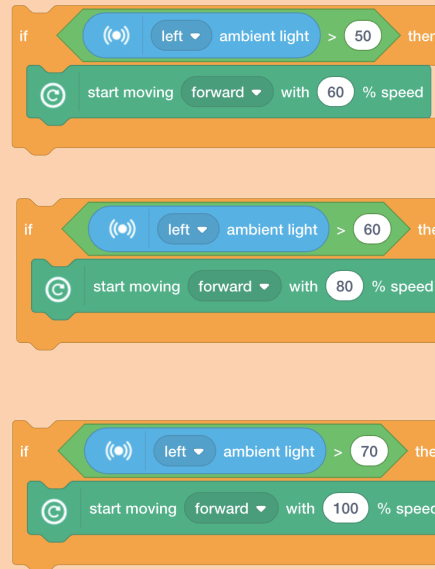


Task : VinciBot will change its forward speed to correspond to changes in ambient light intensity; the stronger the light, the faster the speed, and vice versa.

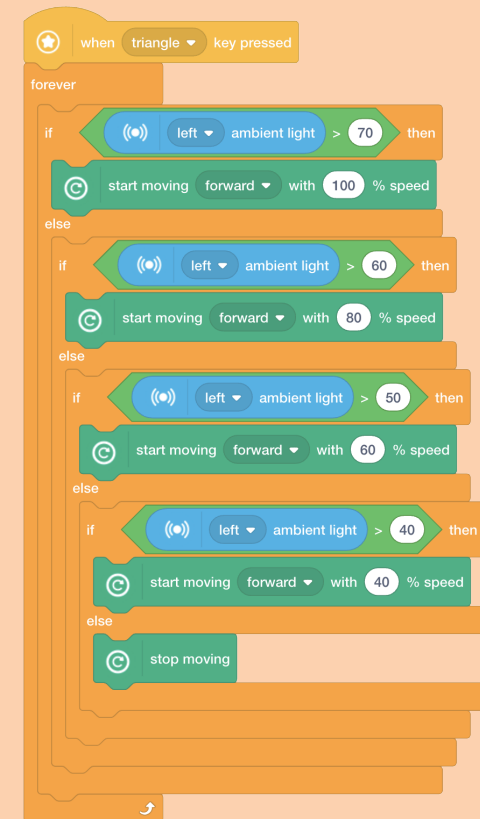
- 1 VinciBot detects ambient light of different intensities: when the light source is closer to VinciBot, the ambient light is stronger; when the light source is more distant, the ambient light is weaker.



- 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.

Designed by MATATALAB in China
MATATALAB CO., LTD.