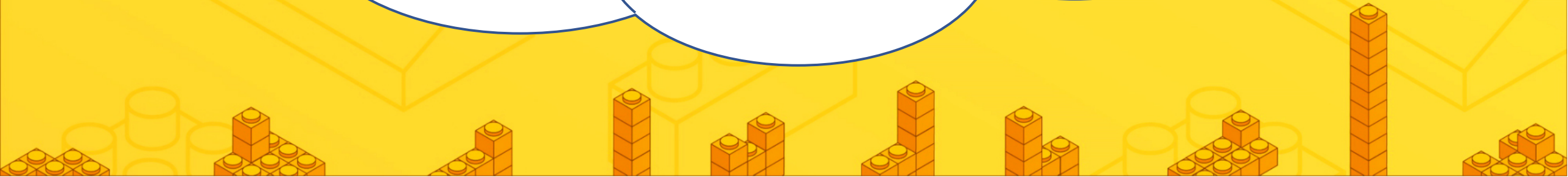




Track Trolley (1)

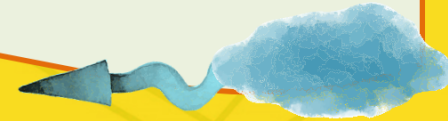


Target

- **Learn to Use a Grayscale Sensor for Basic Line Following**
- **Master the Use of Conditional Structures in Programming**
- **Complete Learning How to Exit Loops**



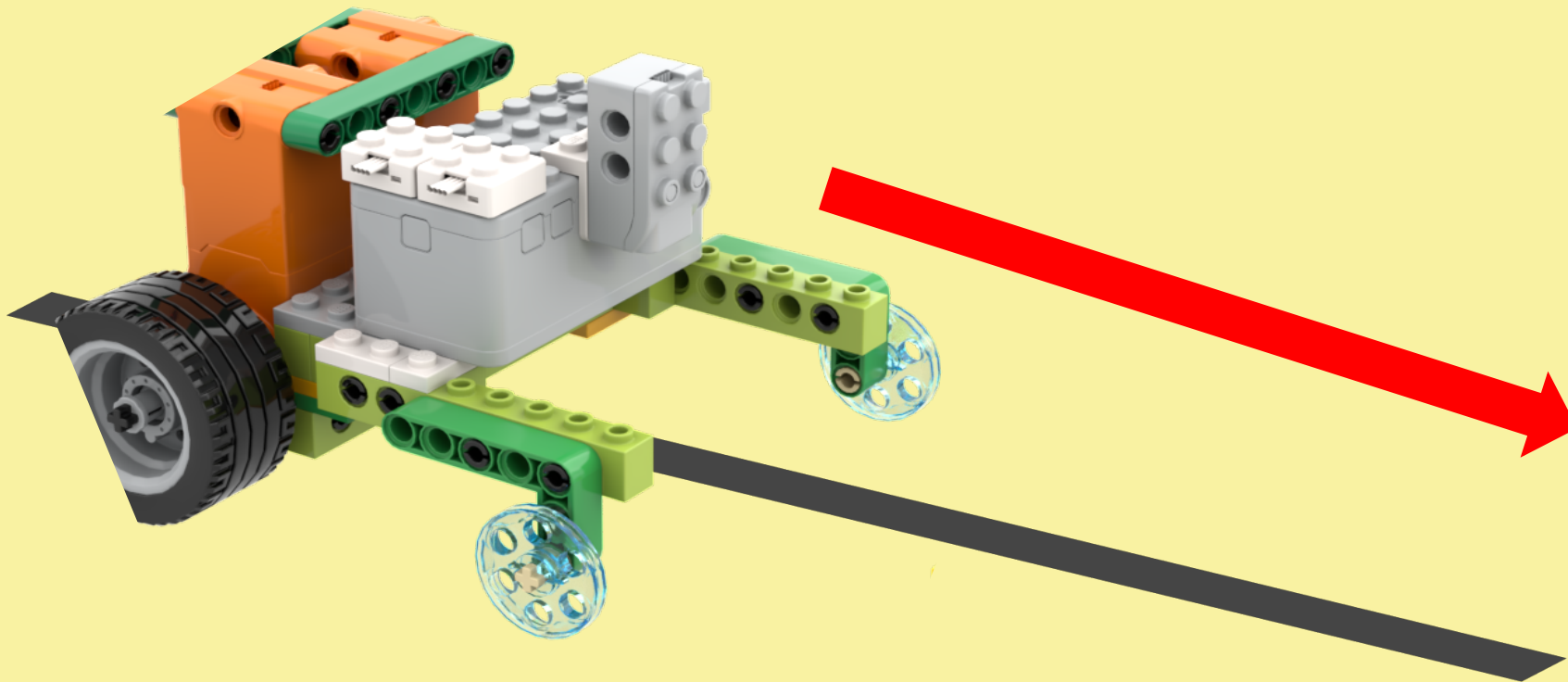
01 Task





Task

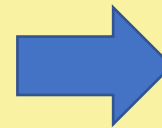
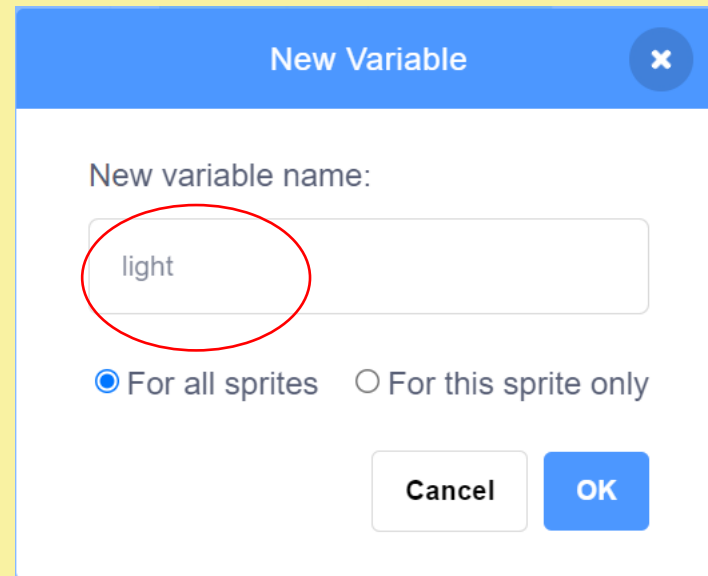
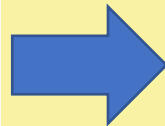
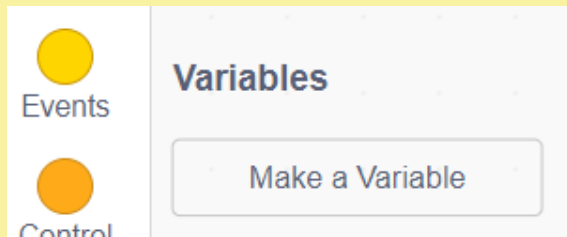
- **Task 1: Use the Grayscale Sensor to Follow the Black Line**



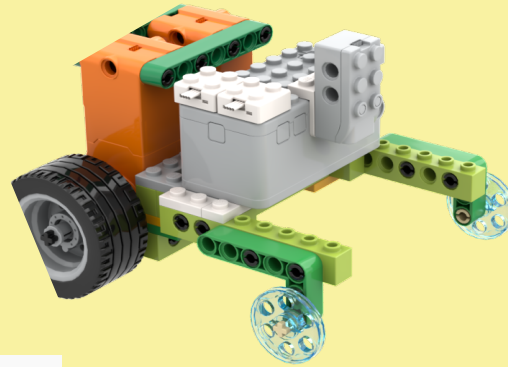
Coding Technique 1

Using Variables and Viewing Sensor Values

Note: Variable names should use only English letters.

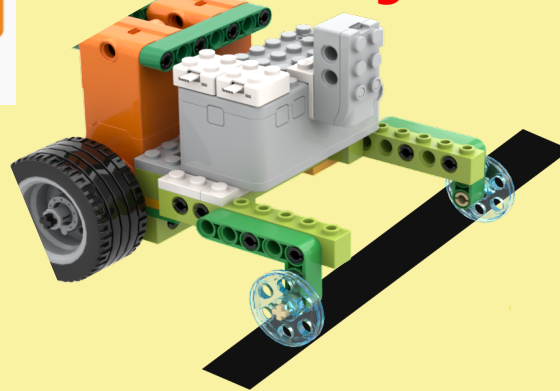


Coding Technique 1



light 18

Pay close attention to the difference between detecting the black line and not detecting it.

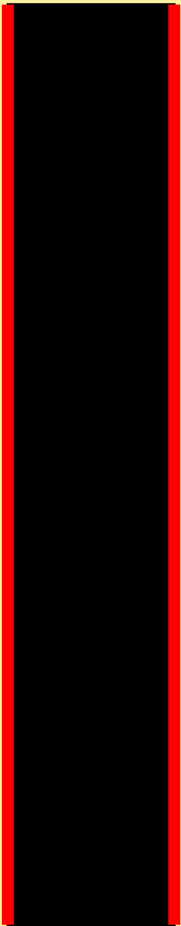


light 29

```
when clicked
  forever
    set light to 1# single channel line tracker's value
```



Coding Technique 1



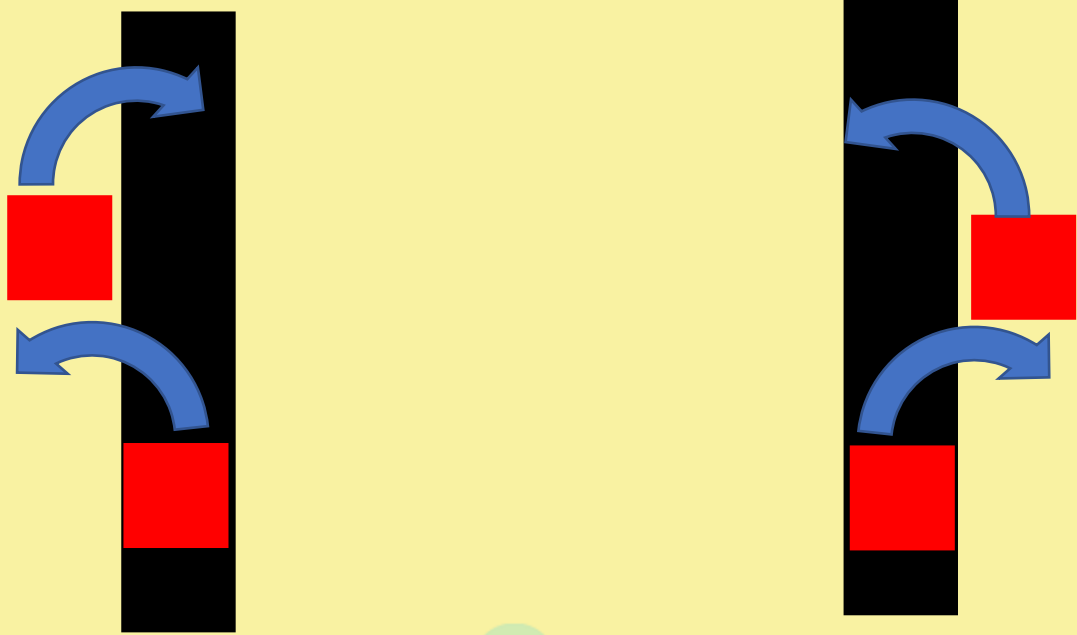
- A robot with a single grayscale sensor does not follow the black line itself.
- Instead, it follows the line where the black line meets the white area (the red line in the left image).
- Since there are two lines where the black line meets the white area, which one should the robot follow?





Coding Technique 1

- Which side to follow depends on your actions on the black line. Observe the diagram below.
- Currently, only two states are detected, so there can only be two corresponding actions.



Task

Referrable Program

Don't forget to upload the robot's program. Otherwise, it might not work successfully.

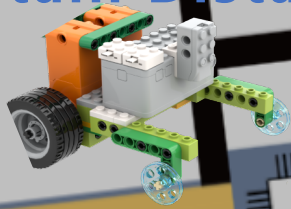
$$\text{Threshold} = \frac{(\text{black line value} + \text{white value})}{2}$$

It is recommended to switch to the power mode.

```
when clicked
  set 1# ext servo's origin
  set 2# ext servo's origin
  wait 0.3 seconds
  forever
    if 1# single channel line tracker's value > 15 then
      set 1# ext servo to keep running at 0 (-100~100)% power on anticlockwise
      set 2# ext servo to keep running at 30 (-100~100)% power on clockwise
    else
      set 1# ext servo to keep running at 30 (-100~100)% power on anticlockwise
      set 2# ext servo to keep running at 0 (-100~100)% power on clockwise
```

Task

- **Task 2: Use the Grayscale Sensor to Follow the Black Line for a Certain Distance and Stop Near Point D.**



2



Task

Referrable Program

First, move straight to exit Starting Point 1.

Use the "Repeat Until" command to stop the loop when the relative angle of the clockwise-rotating motor exceeds 2600 degrees.

The image shows a Scratch script for controlling two servo motors. The script starts with a 'when clicked' event, followed by two 'set' blocks for 'ext servo's origin' for motors 1# and 2#. A 'wait 0.3 seconds' block follows. Then, two 'set' blocks for 'ext servo to keep running at' are shown, both set to 30% speed. Motor 1# is set to 'anticlockwise' and motor 2# to 'clockwise'. A 'wait 1 seconds' block follows. A 'repeat until' loop is used to rotate motor 2# until its 'counted degrees' exceed 2600. Inside the loop, an 'if' statement checks if the 'single channel line tracker's value' for motor 1# is greater than 15. If true, motor 1# is set to 0% power 'anticlockwise' and motor 2# is set to 30% power 'clockwise'. If false, motor 1# is set to 30% power 'anticlockwise' and motor 2# is set to 0% power 'clockwise'. The script ends with a 'stop all ext motor(s)' block.

```
when clicked clicked
set 1# ext servo's origin
set 2# ext servo's origin
wait 0.3 seconds
set 1# ext servo to keep running at 30 (-100~100)% speed on anticlockwise
set 2# ext servo to keep running at 30 (-100~100)% speed on clockwise
wait 1 seconds
repeat until 2# ext servo's counted degrees > 2600
if 1# single channel line tracker's value > 15 then
set 1# ext servo to keep running at 0 (-100~100)% power on anticlockwise
set 2# ext servo to keep running at 30 (-100~100)% power on clockwise
else
set 1# ext servo to keep running at 30 (-100~100)% power on anticlockwise
set 2# ext servo to keep running at 0 (-100~100)% power on clockwise
stop all ext motor(s)
```

Additional Task

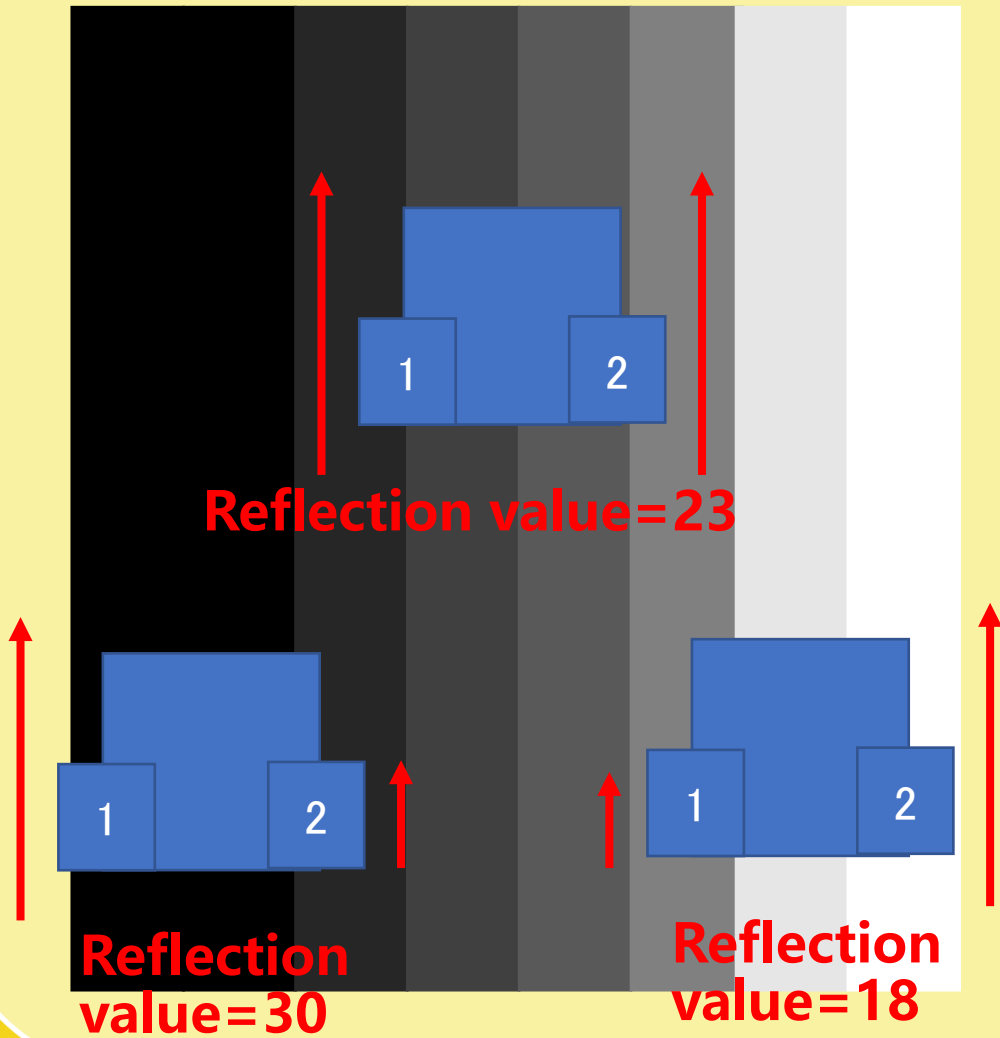
- **Note: This task is not mandatory for learning. You only need to know which parameters to adjust.**

- Additional following



perform line-

Coding Technique 2



- The boundary between black and white is a gradient pattern.
- The arrows represent the power levels.
- Have you noticed the proportional relationship between the reflection value and the motor power?



Coding Technique 2

Original Power
(20~30)

$$\text{Threshold} = \frac{(\text{black line value} + \text{white value})}{2}$$

The image shows a Scratch script with the following components:

- when clicked** trigger block.
- forever** loop block containing two **set servo to keep running at** blocks.
- The first servo block has a value of $25 - 2 \cdot 10$. The numbers 25, 2, and 10 are highlighted with red boxes, and red arrows point from the text "Original Power (20~30)" to the 25, and from the equation above to the 2 and 10.
- The second servo block has a value of $25 + 2 \cdot 10$. A red arrow points from the text below to the 2.
- Both servo blocks are connected to a **single channel line tracker's value** sensor, with the power set to **on** and the direction set to **anticlockwise** for the first and **clockwise** for the second.

The larger the proportional value, the more intense the oscillation.

