matata studio

TEACHER'S BOOK

AI Coding and Robotics

with Nous AI Set

01

Hello Nous!

ISSUE： Modular, electronic components, structural parts, robot construction
CORE IDEA： None
BASIC LESSON
Difficulty: ★★ Duration: 60 min
Data: None
Required Items: Nous Hub, Nous Camera Module, 2 Nous Servo Modules, and the parts to build a Nous Robot

Nous is a modular artificial intelligence educational set that supports a variety of artificial intelligence technologies and functions. In this lesson, we will first understand the electronic modules and structural components of Nous, and then use them to build the main form of the Nous Robot. Finally, students will experience the Nous Hub's preset functions.

1 Identify a Problem (5min)

2 Guided Practice (25min)
- Study electronic modules for the Nous
- Study structural components of Nous

3 Independent Practice (20min)
- Build the Nous Robot per the construction manual

4 Wrap-up (10min)
- Explore the Nous Hub's preset functions

Learning outcomes:

TSWBAT: the students will be able to

- Understand the Nous' characteristics.
- Learn the electronic modules for the Nous AI Set: Nous Hub, Nous Camera Module, Nous Servo Module.
- Become familiar with the structural components of the Nous AI Set and be able to build a Nous Robot per the construction manual.
- Explore the Nous' preset functions, such as MatataChat, human face detection, etc., and learn about some basic AI functions.

Identify a problem

5 min

The Nous AI Set is a modular artificial intelligence educational set. In addition to basic robot programming control, it supports a variety of artificial intelligence technologies and functions, including machine learning, neural networks, machine vision, speech recognition, ChatGPT, AIGC, and Autonomous driving capabilities. Additionally, it also supports data collection, AI model training and deployment, as well as graphical programming and Python programming. The robot is designed with a metal structure that is also compatible with building block structures and features an expandable electronic module system. Here, we first watch a Nous case video collection to experience the main functions of Nous.

Nous consists of two parts: hardware and software. The hardware includes the electronic modules, structural components, etc. that we see out of the box, and the software is MatataCode, a programming platform specifically designed to control Nous. In this lesson, we will first understand Nous' hardware part.

As mentioned, the Nous AI Set is a modular set. Various robot forms can be built from these electronic modules and structural components. In this lesson, we will first learn the modules and components, and then use them to build the Nous Robot.

Guided Practice

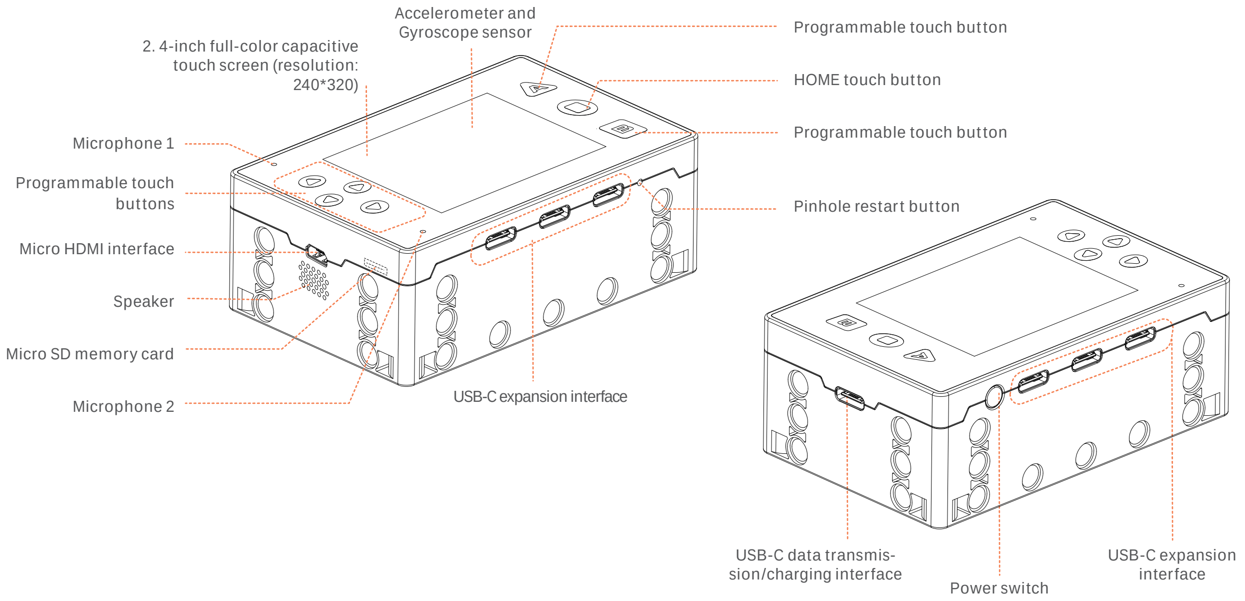
25 min

In this chapter, the teacher will lead students to become familiar with the electronic modules and structural components of Nous.

1. Study electronic modules of Nous

- Nous Hub

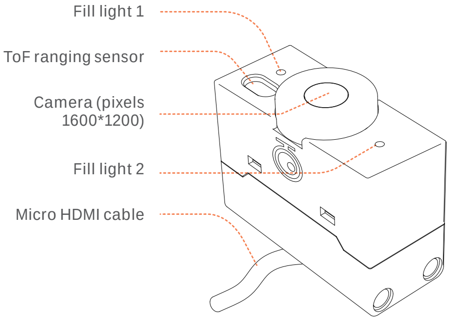
Nous Hub is the core of the Nous and acts as the Nous's brain. As shown in the picture, in the front, there is a 320 x 240,2.4-inch full-color capacitive touch screen, 6 programmable touch buttons on both sides of the touch screen, including "up, down, left, right" and "A, B", as well as a "Home" touch button. In addition, Nous Hub also contains two microphones, a speaker, 6 type-C expansion ports, a data transmission interface to connect Nous Camera to Nous Hub, and a data transmission interface to connect Nous Hub to MatataCode/computer devices (both interfaces can be used to charge Nous Hub). Of course, we can also see the orange switch button in the upper right corner, and the pinhole reset button in the lower right corner of Nous Hub. Additionally, there is a gyroscope inside Nous Hub which can't be seen.



Accelerometer and Gyroscope sensor
2. 4-inch full-color capacitive touch screen (resolution: 240*320)
Programmable touch button
HOME touch button
Programmable touch button
Microphone 1
Programmable touch buttons
Micro HDMI interface
Speaker
Micro SD memory card
Microphone 2
Pinhole restart button
USB-C expansion interface
USB-C data transmission/charging interface
Power switch
USB-C expansion interface

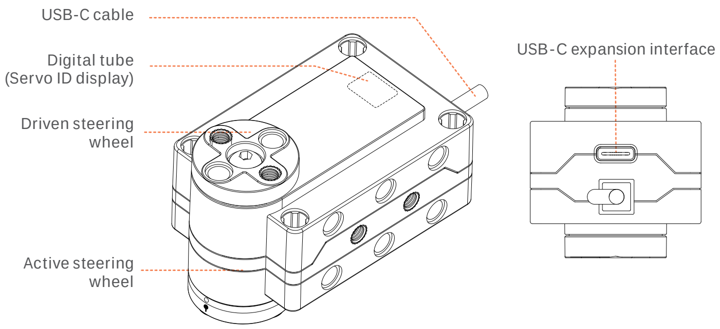
- Nous Camera Module

Nous Camera is the Nous' eyes. As shown in the picture, in addition to the camera, we can also see two camera fill lights and a ToF (Time of Flight) ranging sensor.



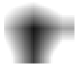
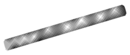
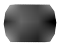
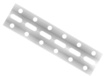
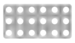
- Nous Servo Module

Nous Servo Module has motor mode and servo mode, which can meet a variety of usage scenarios.



2. Study structural components of Nous

The list of Nous structural parts is as follows. We can see that it mainly includes: double-hole beams, square-hole beams, connecting pieces, brackets, tires, rubber sleeves, screws, etc.

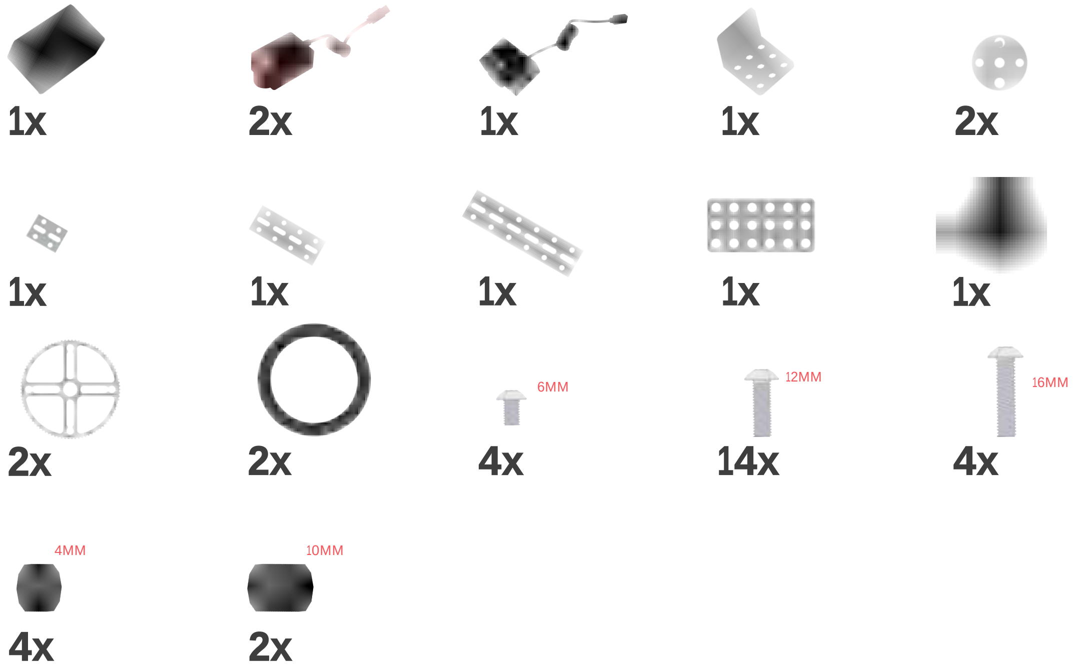
Name	Picture	Qty	Name	Picture	Qty
camera bracket		1	universal wheel		1
3×3 L-shaped bracket		5	transmission fixed plate		2
1×4 square hole beam		2	Ø4-60mm axis		1
2×2 double hole beam		1	Ø7-2mm rubber sleeve		6
2×3 double hole beam		2	Ø7-4mm rubber sleeve		6
2×4 double hole beam		1	Ø7-8mm rubber sleeve		6
2×5 double hole beam		2	Ø7-10mm rubber sleeve		4
2×6 double hole beam		2	M4-6mm screw		18
135° single head connecting piece		2	M4-12mm screw		22
3×6 connecting piece		1	M4-16mm screw		30
Ø68mm tire		2	M4-20mm screw		4
Ø58mm wheel hub		2	M4 nut		6

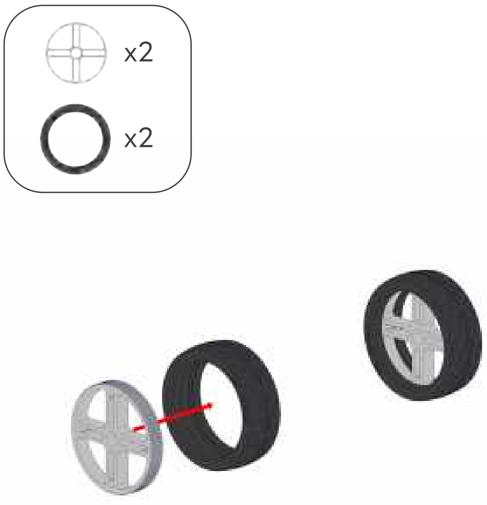
Independent Practice

20 min

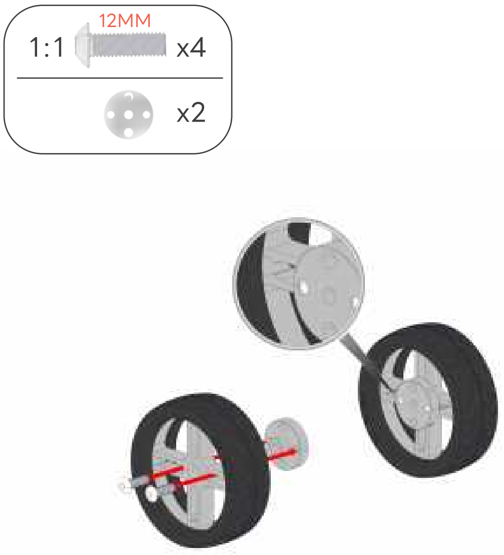
Students independently build Nous Robot, the main form of Nous, according to the construction steps.

Parts List

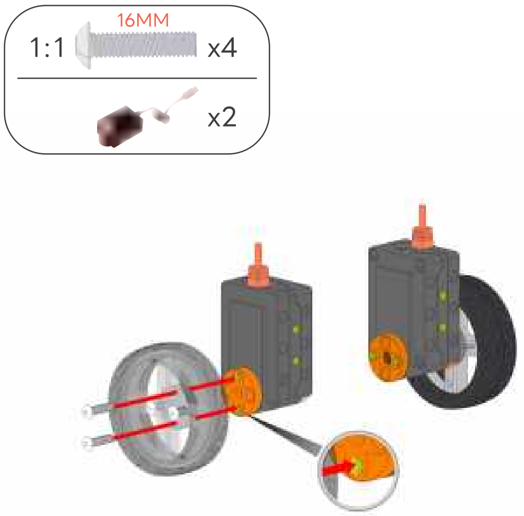




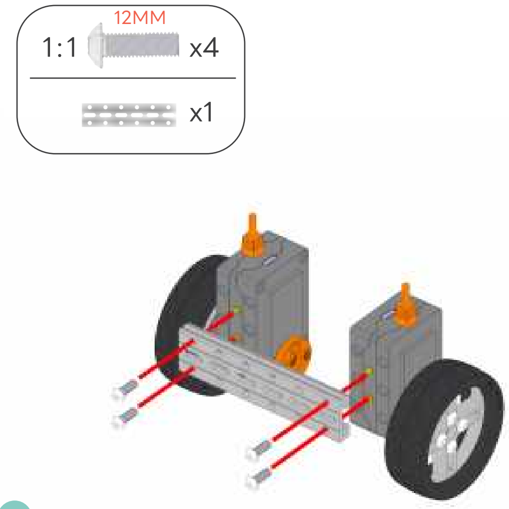
1



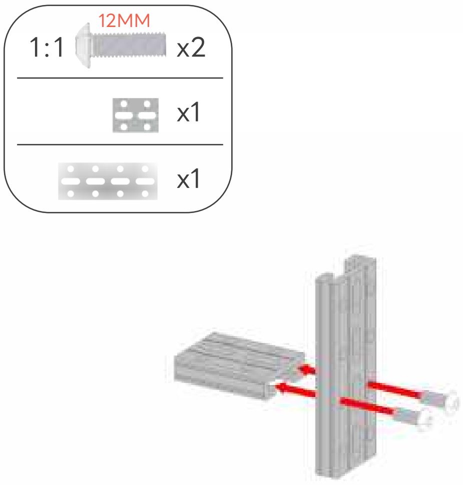
2


1:1
16MM
x4
x2

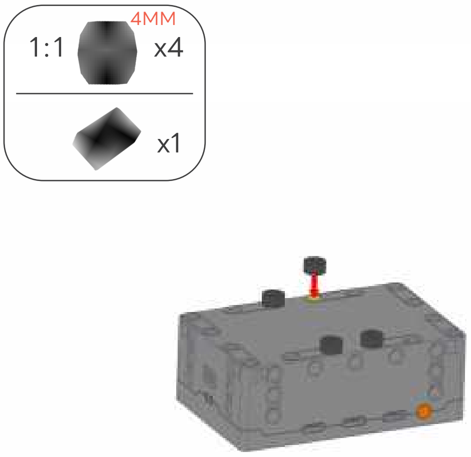
3


1:1
12MM
x4
x1

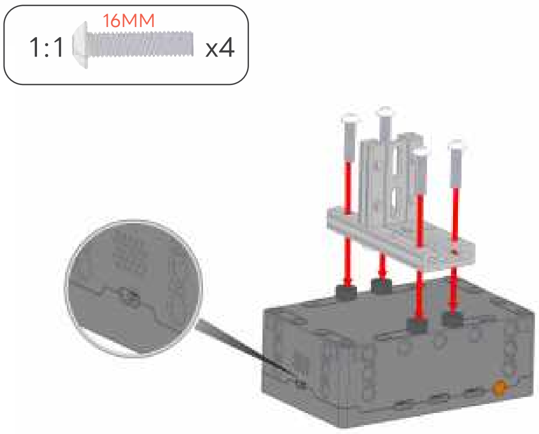
4


1:1
12MM
x2
x1
x1

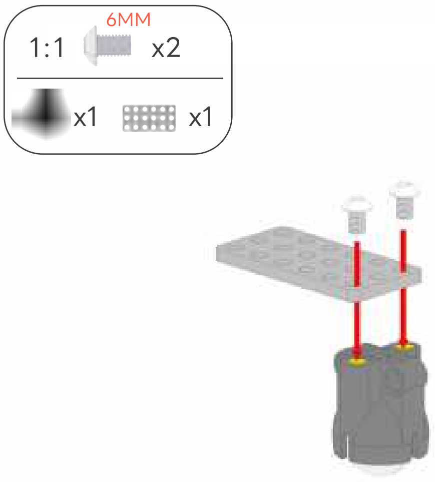
5


1:1
4MM
x4
x1

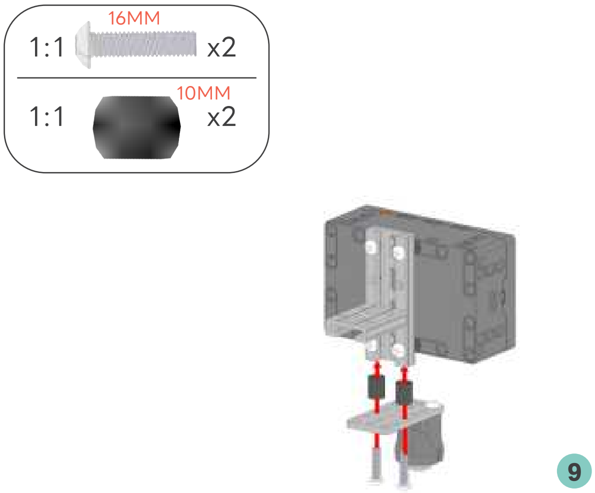
6


1:1
16MM
x4

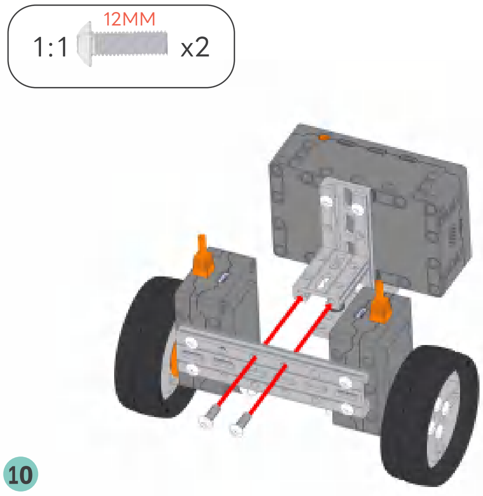
7


1:1
6MM
x2
x1
x1

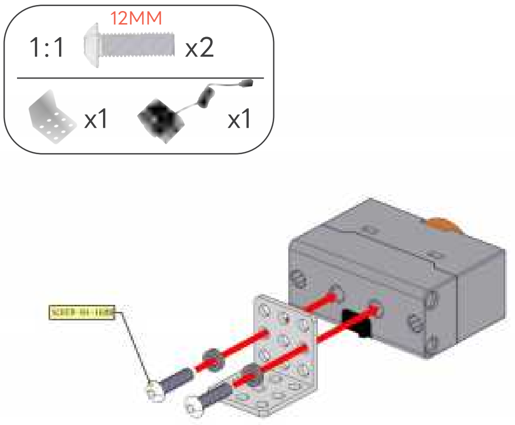
8



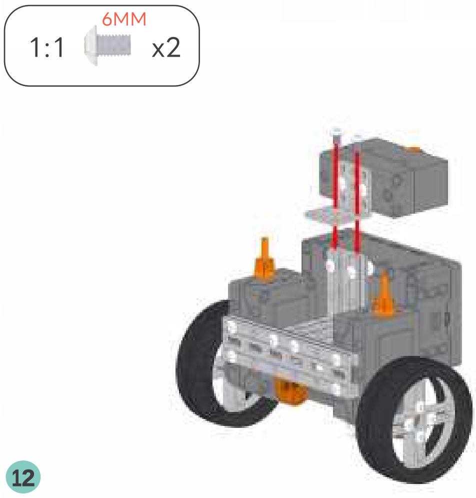
9



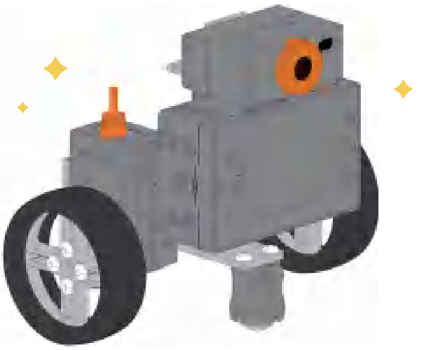
10



11



12



Wrap-up

10 min

Students can freely explore the Nous Hub's preset functions, especially the AI functions such as MatataChat, facial recognition, and AprilTag recognition. Here, facial recognition and AprilTag recognition are just simple recognitions. We can see the positioning box that appears on the screen during recognition. In the next lesson, we will program and experience these functions more intuitively. In the following learning process, we will use these AI functions to complete more complex AI projects such as facial tracking and autonomous driving.

02

Basic Coding for Nous

ISSUE： Coding Platform,MatataCode
CORE IDEA： Perception
BASIC LESSON
Difficulty: ★★ Duration: 60 min
Data: None
Required Items: Nous Robot,PC or Pad,USB cable, AprilTag cards

Nous' programming platform, MatataCode, supports both graphical programming and Python programming. In addition to basic programming, it also supports AI functions such as computer vision, embedded machine learning (Tiny ML), etc. In this lesson, we will first become familiar with the functions of MatataCode, study the firmware upgrade, learn about connection, and program downloads. Then, try to write simple programs for Nous using graphical programming language and run these programs.

1 Identify a problem (5 min)

2 Guided Practice (20 min)

- Overview of programming platform features
- Firmware upgrade
- Connect and download programs
- Become familiar with the types of the programming coding blocks

3 Independent Practice (30 min)

- Write simple programs for Nous Robot

4 Wrap-up (5 min)

Learning outcomes:

TSWBAT: the students will be able to

- Understand the Nous programming platform, master the major functionality and the role of each function.
- Learn how to upgrade the firmware.
- Get familiar with the graphical programming coding blocks, especially visual and screen coding blocks.
- Be able to write simple programs for Nous Robot and run these programs.

AI VOCA

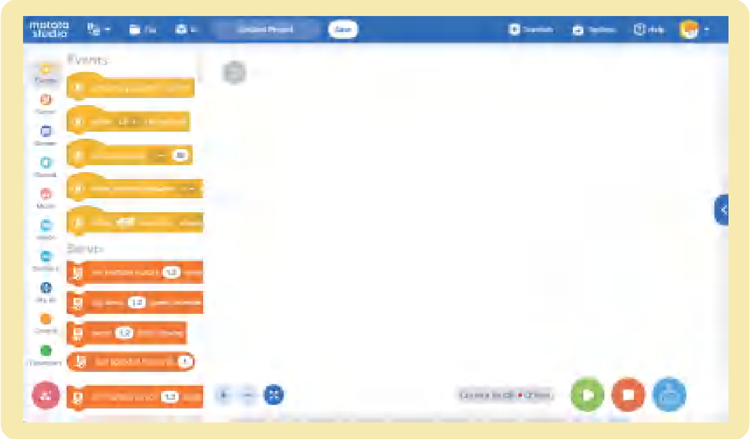
- Perception: Nous can "see" image information.
- Natural Interaction: Nous Robot can realize facial recognition, AprilTag recognition, and other AI functions to interact with people.

Identify a problem

5 min

In the last class, we learned about the hardware part of Nous and built Nous Robot. In this lesson, we'll get familiar with the programming platform of Nous, MatataCode. Master the connection method between Nous and MatataCode, try to use graphical programming language to write simple programs for Nous, and run these programs.

MatataCode App

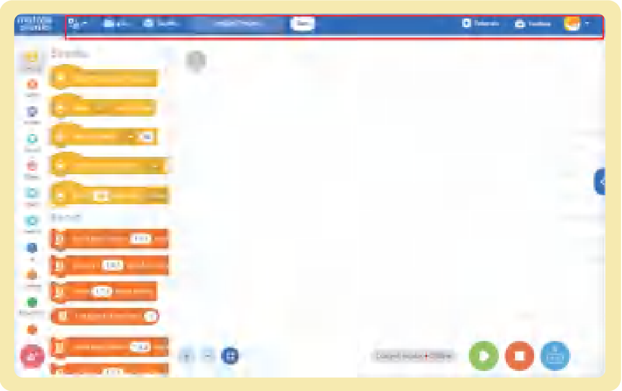


Guided Practice

20 min

In this chapter, the teacher will lead students to become familiar with the Nous' programming platform, MatataCode, and master how to use MatataCode to program Nous. Nous' programming platform, MatataCode, has a website version and APP version. The URL of the web version is https://nous.matatastudio.com/. After entering the website, we can download the APP version from the application download in the "Toolbox".

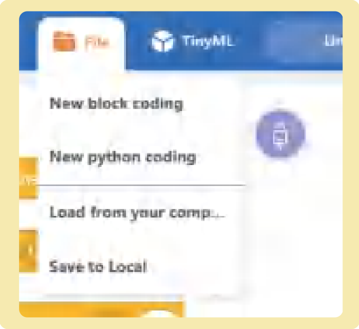
1. Overview of MatataCode's functions.

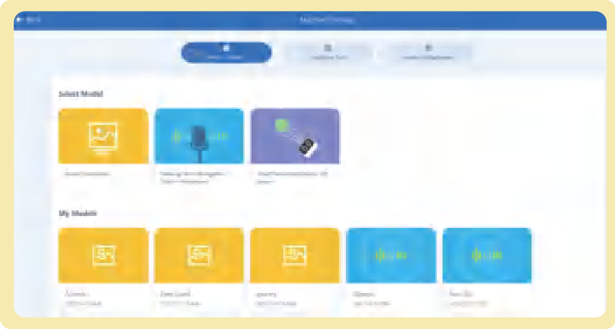


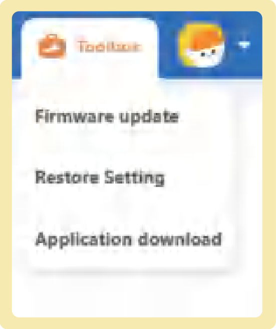
When we open the Nous programming platform, we can see a page as shown below. The menu bar above the page contains "language", "file", "Tiny ML", a place to change the file name, "save", "tutorials", "toolbox" and "my account".

Among them, you need to pay attention to the following toolbars: "File" contains four functions: new block coding, new Python coding, load from your computer(mc format program), and save to the local (mc format program); Press "AI" and you will see the drop-down menu which includes "Tiny ML model". It is the entrance to create embedded machine learning models, which we will explain later; "Tutorials" is the entrance to cases; "Toolbox" has three functions: firmware upgrade, restore setting, and application download.

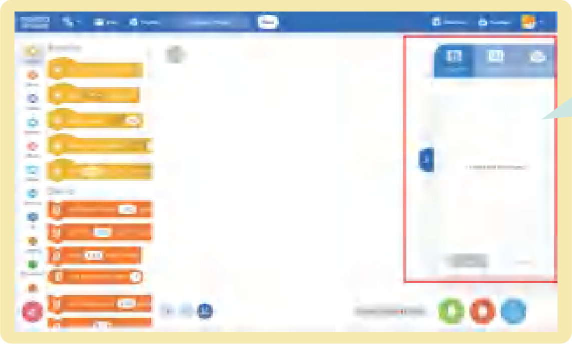
In addition, after registering an account in the "My Account", we can save personally written programs into our own online account for easy recall anytime and anywhere. When registering an account, to ensure network security, we need to set an appropriate account name and password.

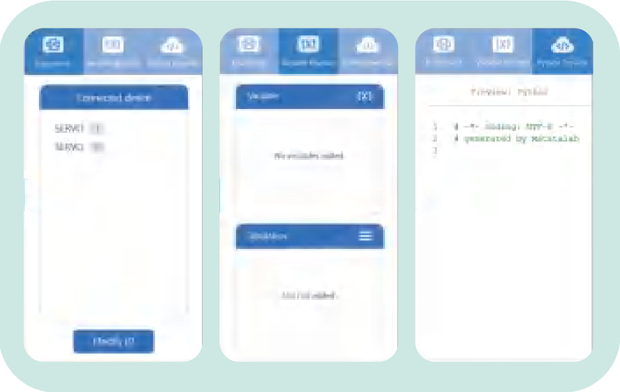






On the right side of the page, there are mainly three functions: equipment, variable monitor, and Python preview. The equipment is mainly used to modify the ID number of the servo. In the variable monitor, you can mainly see the variables or lists used in programming. The Python preview function can help us become familiar with the corresponding Python text program that will be used after completing the graphical programming.

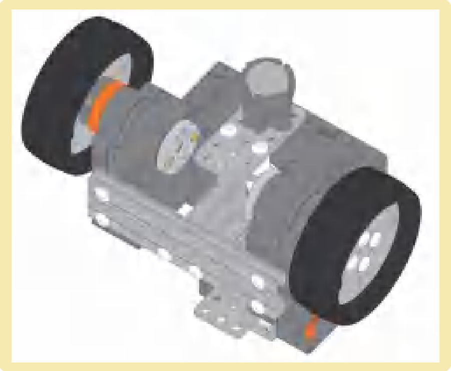




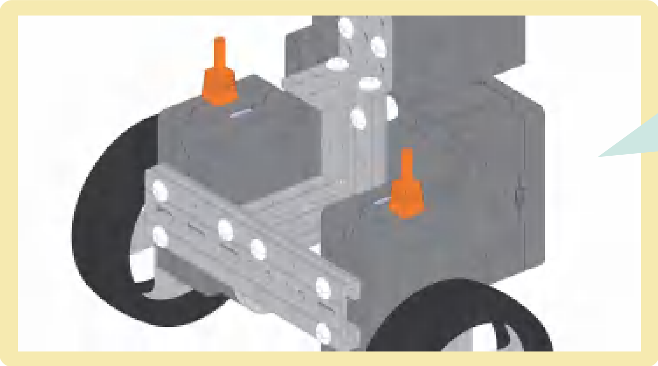
```
1  # -*- coding: UTF-8 -*-
2  # generated by mCotalab
3
```

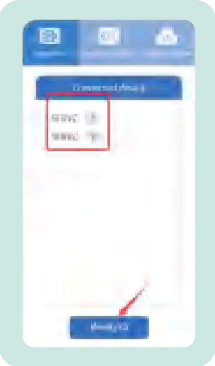
Here you need to pay special attention to how to modify the servo ID.

Nous Robot uses the rotation of two Nous servos to rotate the wheels, and then cooperates with a universal wheel to control the Nous Robot's forward, backward, left and right turns. Nous servo should have its own ID. During programming, the rotation speed and direction of the servo and tires can be controlled by the corresponding ID, allowing the robot to move freely in different directions.



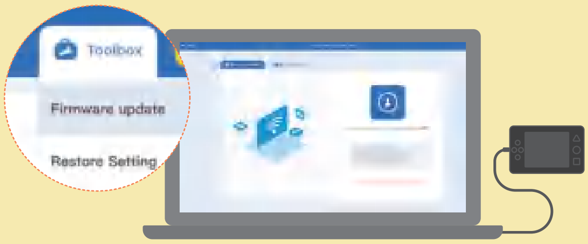
After turning on the Nous Robot, you can see the corresponding ID on the inside of each Nous servo. The IDs of the two servos maybe the same or they maybe different. Note that when two servos have the same ID, only one ID will appear in "equipment". Now, you first need to disconnect one Nous servo, change the ID of the connected Nous servo to a different ID, and then connect the other servo for modifications.



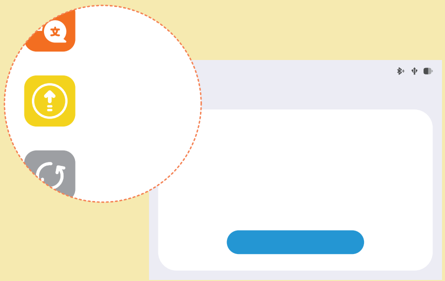


2. Learn how to upgrade the firmware.

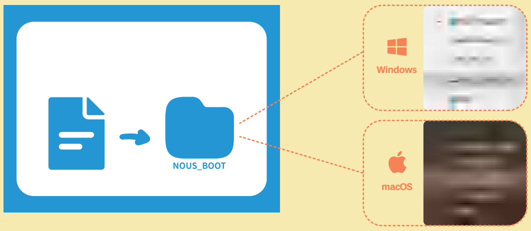
Log in to nous.matatastudio.com and connect Nous Hub to MatataCode. Select Toolbox-Firmware Upgrade to check if Nous Hub has latest version firmware. If not, download latest version firmware. connection between Nous Hub and computer device. If fails, log in to Help Center at nous.matatastudio.com for detailed guidance.



Enter the Nous Hub system operation interface, select Settings-Firmware Upgrade, enter upgrade mode.



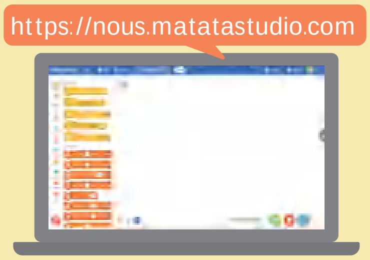
After connecting Nous Hub to my computer via USB-C cable, 'NOUS_BOOT'disk appears. Drag the latest version downloaded firmware into disk, and Nous Hub will automatically update firmware and restart.



During upgrade process, don't interrupt connection between Nous Hub and computer device. If fails, log in to Help Center at nous.matatastudio.com for detailed guidance.

3. Learn how to connect Nous to MatataCode, and how to run the program.

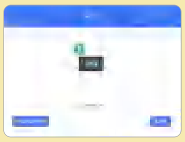
Log in to the web programming platform:
https://nous.matatastudio.com



Select the USB data cable (USB-C) or Bluetooth connection method to connect the Nous Hub to your device.

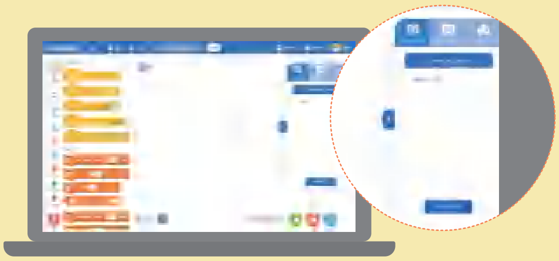
② Choose how you want to connect

It is recommended to use USB data cable connection to ensure efficient and stable data transmission



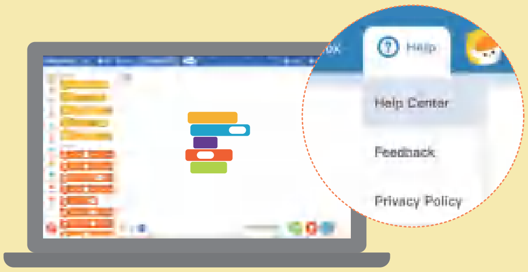
③ After the connection is successful, your device will automatically pop up a prompt.

When the USB-C expansion interface of the Nous Hub is connected to the servo module, you can view and modify the servo's ID address in the device management.



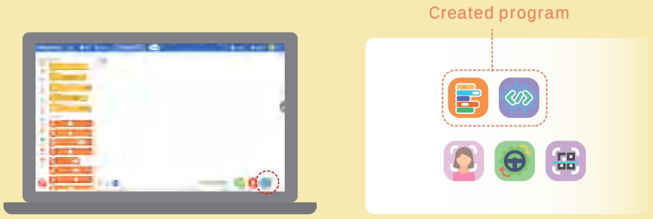
*Note: The factory default ID addresses of the two servos configured by Nous have been set to 1&2.

Create your own program.Use building blocks to create your own program code.



*Note: For more instructions on how to use Nous,please click the Help Center in the navigation bar.

Download the program to your Nous Hub. Click the download program icon to download the created program to the Nous Hub. At this time, you can see the program you just created in the Nous Hub operation interface. Click to run.



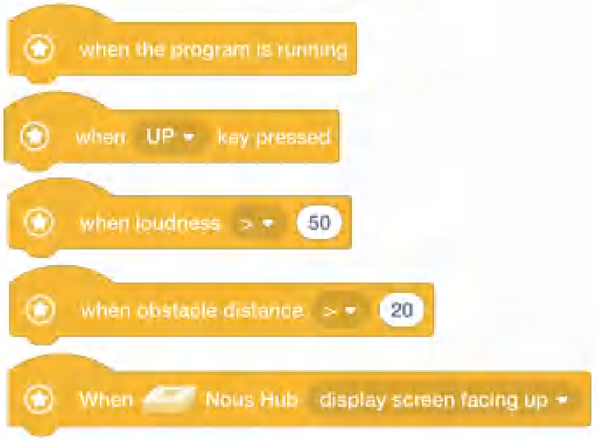
Quickly switch program & delete program

●Nous supports storing multiple programs. When you want to switch from the currently running program to another program, you can long press the Home button to exit and select the corresponding program;

●When you want to delete a personally created program, you can long press the corresponding program icon, and the system will prompt you to confirm whether you want to delete it. Just select to confirm the deletion.

4. Get familiar with the graphical programming coding blocks, especially visual and screen coding blocks.

Events



when the program is running

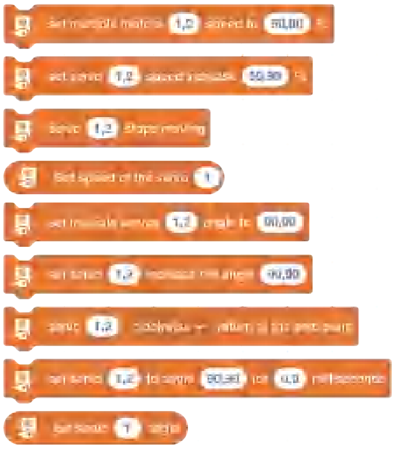
when UP ▾ key pressed

when loudness > ▾ 50

when obstacle distance > ▾ 20

When Nous Hub display screen facing up ▾

Servo



set multiple motors 1,2 speed to 50,00 %

set servo 1,2 speed increase 50,50 %

Servo 1,2 Stops moving

Get speed of the servo 1

set multiple servos 1,2 angle to 90,90

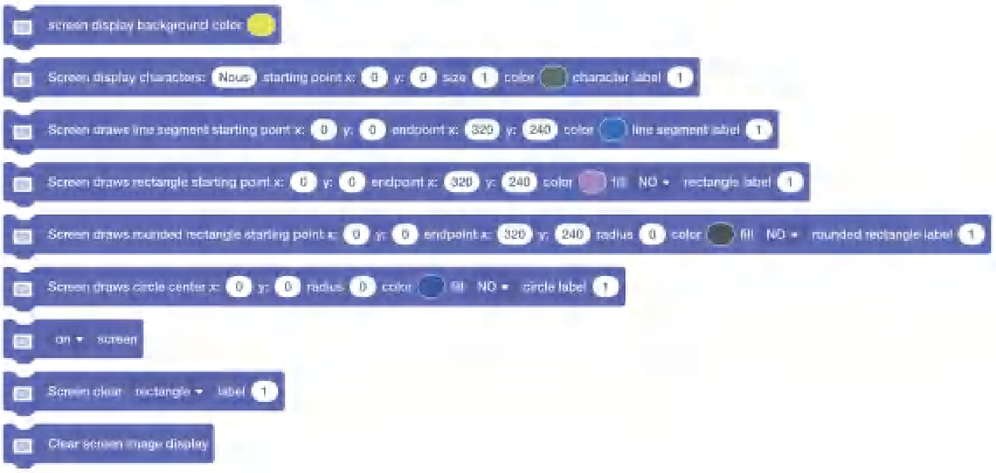
set servo 1,2 increase the angle 90,90

servo 1,2 clockwise ▾ return to the zero point

set servo 1,2 to angle 90,90 for 0,0 milliseconds

Get servo 1 angle

Screen



screen display background color ●

Screen display characters: Nous starting point x: 0 y: 0 size 1 color ● character label 1

Screen draws line segment starting point x: 0 y: 0 endpoint x: 320 y: 240 color ● line segment label 1

Screen draws rectangle starting point x: 0 y: 0 endpoint x: 320 y: 240 color ● fill NO ▾ rectangle label 1

Screen draws rounded rectangle starting point x: 0 y: 0 endpoint x: 320 y: 240 radius 0 color ● fill NO ▾ rounded rectangle label 1

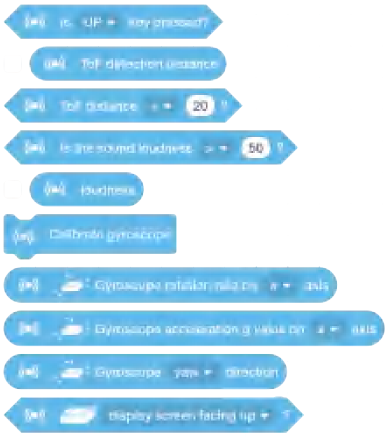
Screen draws circle center x: 0 y: 0 radius 0 color ● fill NO ▾ circle label 1

on ▾ screen

Screen clear rectangle ▾ label 1

Clear screen image display

Sensors



is UP ▾ key pressed?

ToF detection distance

ToF distance > ▾ 20 ?

Is the sound loudness > ▾ 50 ?

loudness

Calibrate gyroscope

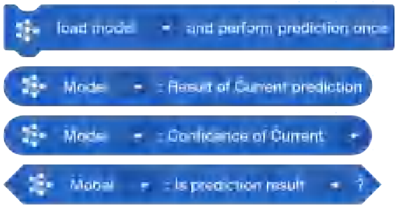
Gyroscope rotation rate on x ▾ axis

Gyroscope acceleration g value on x ▾ axis

Gyroscope yaw ▾ direction

display screen facing up ▾ ?

My AI



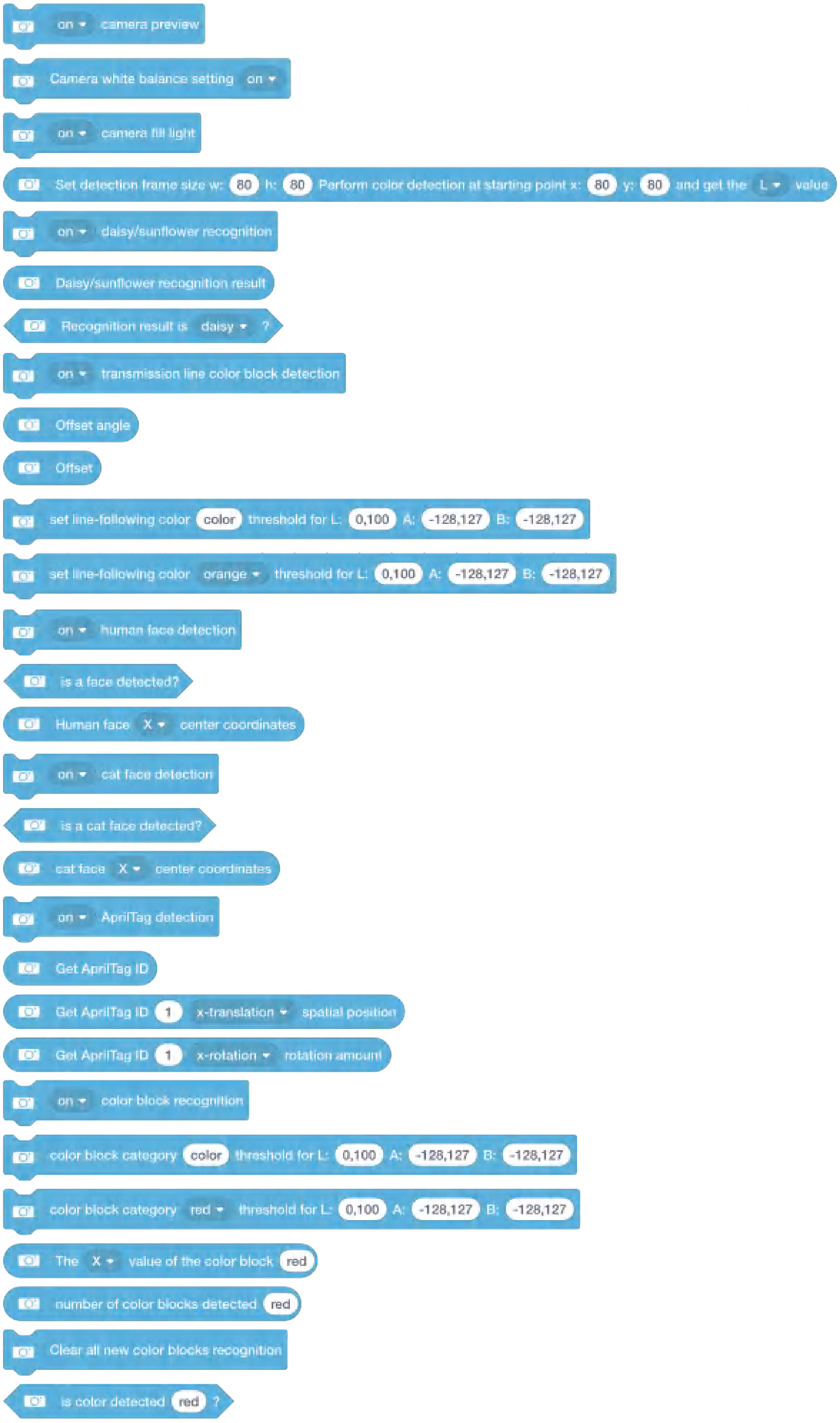
load model ▾ and perform prediction once

Model ▾ : Result of Current prediction

Model ▾ : Confidence of Current ▾

Model ▾ : Is prediction result ▾ ?

Vision



```
on ▾ camera preview
Camera white balance setting on ▾
on ▾ camera fill light
Set detection frame size w: 80 h: 80 Perform color detection at starting point x: 80 y: 80 and get the L ▾ value
on ▾ daisy/sunflower recognition
Daisy/sunflower recognition result
Recognition result is daisy ▾ ?
on ▾ transmission line color block detection
Offset angle
Offset
set line-following color color threshold for L: 0,100 A: -128,127 B: -128,127
set line-following color orange ▾ threshold for L: 0,100 A: -128,127 B: -128,127
on ▾ human face detection
is a face detected?
Human face X ▾ center coordinates
on ▾ cat face detection
is a cat face detected?
cat face X ▾ center coordinates
on ▾ AprilTag detection
Get AprilTag ID
Get AprilTag ID 1 x-translation ▾ spatial position
Get AprilTag ID 1 x-rotation ▾ rotation amount
on ▾ color block recognition
color block category color threshold for L: 0,100 A: -128,127 B: -128,127
color block category red ▾ threshold for L: 0,100 A: -128,127 B: -128,127
The X ▾ value of the color block red
number of color blocks detected red
Clear all new color blocks recognition
is color detected red ?
```

Independent Practice

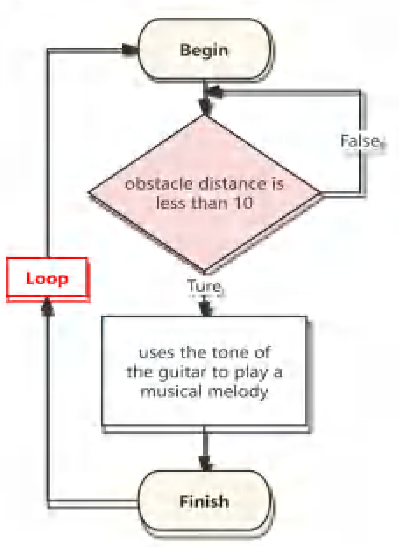
30 min

Now that we are familiar with the Nous programming platform, let's program the Nous Robot.

There are 4 programming tasks lised on the worksheet. Students write programs for Nous and run these programs according to the task requirements.

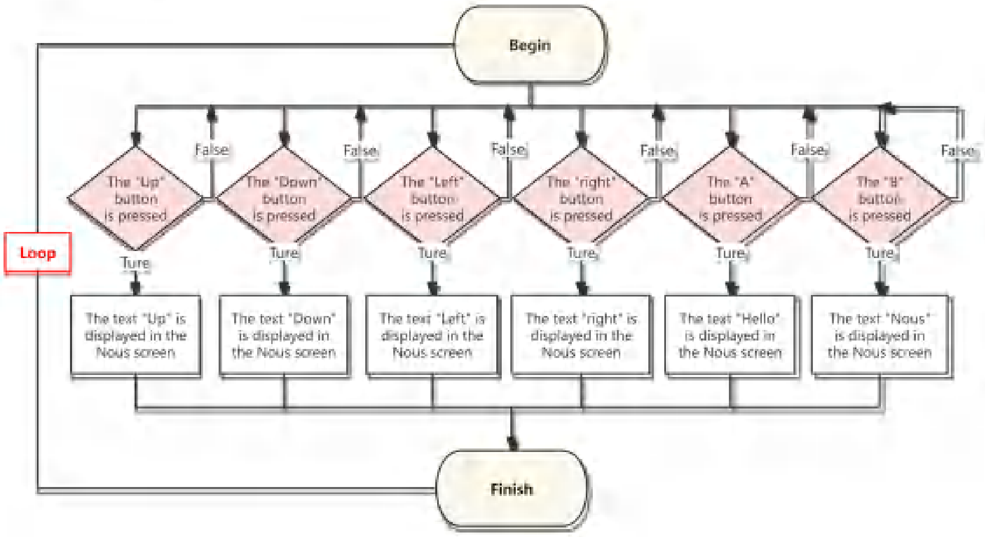
Task 1 Nous is a musician

When the obstacle distance is less than 10, Nous uses the tone of the guitar to play a musical melody, such as "Are You Sleeping?".


Begin
False
obstacle distance is less than 10
Loop
Ture
uses the tone of the guitar to play a musical melody
Finish

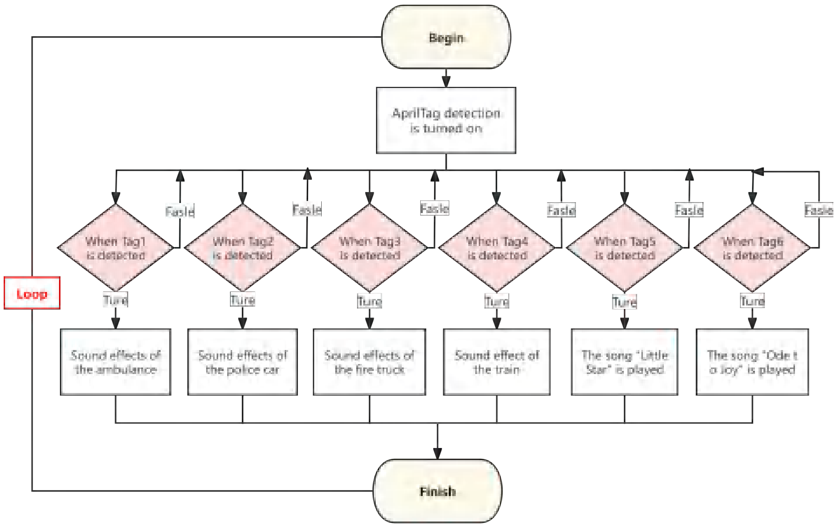
Task 2 Nous can show text

When the "Up" button is pressed, the text "Up" is displayed on the Nous' screen.
When the "Down" button is pressed, the text "Down" is displayed on the Nous' screen.
When the "Left" button is pressed, the text "Left" is displayed on the Nous' screen.
When the "right" button is pressed, the text "Right" is displayed on the Nous' screen.
When button A is pressed, the text "Hello" is displayed on the Nous' screen.
When button B is pressed, the text "Nous" is displayed on the Nous' screen.


Begin
Loop
The "Up" button is pressed
False
Ture
The text "Up" is displayed in the Nous screen
The "Down" button is pressed
False
Ture
The text "Down" is displayed in the Nous screen
The "Left" button is pressed
False
Ture
The text "Left" is displayed in the Nous screen
The "right" button is pressed
False
Ture
The text "right" is displayed in the Nous screen
The "A" button is pressed
False
Ture
The text "Hello" is displayed in the Nous screen
The "B" button is pressed
False
Ture
The text "Nous" is displayed in the Nous screen
Finish

Task 3 Nous detects AprilTags

When the program is running, AprilTag detection is turned on. When Tag1 is detected, Nous plays the sound effects of an ambulance until done; when Tag2 is detected, Nous plays the sound effects of a police car until done; when Tag3 is detected, Nous plays the sound effects of a fire truck until done; When Tag4 is detected, Nous plays the sound effect of a train until done; when Tag5 is detected, the song "Little Star" is played until done; when Tag6 is detected, the song "Ode to Joy" is played until done.

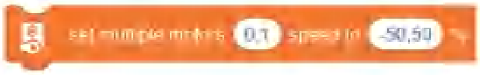

Begin
AprilTag detection is turned on
When Tag1 is detected
When Tag2 is detected
When Tag3 is detected
When Tag4 is detected
When Tag5 is detected
When Tag6 is detected
Fasle
Ture
Loop
Sound effects of the ambulance
Sound effects of the police car
Sound effects of the fire truck
Sound effect of the train
The song "Little Star" is played
The song "Ode t o Joy" is played
Finish

Task 4 Hello, I am Nous!

When the program is running, turn on human facial detection. If a face is detected, Nous says "Hello, I am Nous!", Nous displays "Hello" on the screen, Nous sings "Little Star", and Nous sings "Little Star" at 50% speed. Walk forward for one second, and Nous walks back for one second at 50% speed.

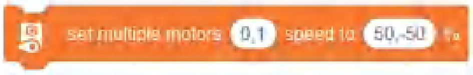
Note: When the ID of the servo connected to the left wheel of Nous Robot is 0, and the ID of the servo connected to the right wheel is 1. If the servo 0 is set to a negative number and the servo 1 is set to a positive number, the Nous Robot will move forward; otherwise, the Nous Robot will move backward.

Move Forward

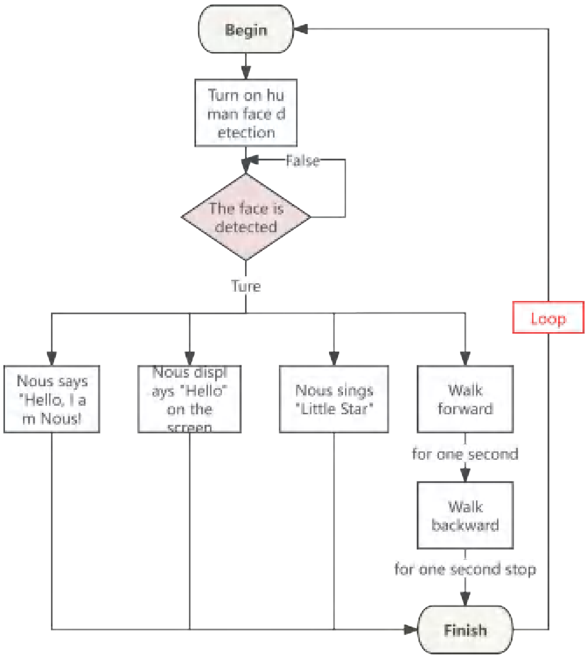


```
set multiple motors (0,1) speed to (-50,50) %
```

Move backwrd



```
set multiple motors (0,1) speed to (50,-50) %
```

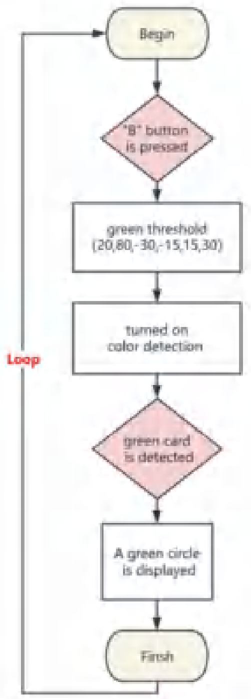


Task 5 Nous recognizes colors

When the "A" button is pressed, the color red threshold is set to (20,80,32,60,16,45), and color detection is turned on. When a red card is detected, a red circle is displayed in the center of the screen.

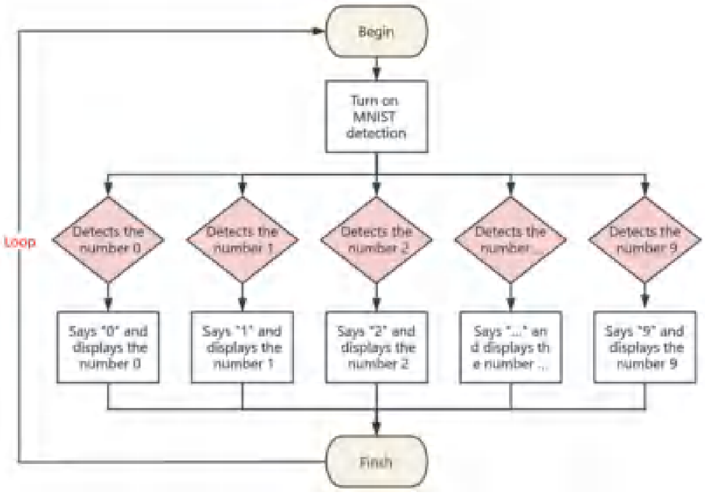
When the "B" button is pressed, the color green threshold is set to （20,80,-30,-15,15,30）, and color detection is turned on. When a green card is detected, a green circle is displayed in the center of the screen.

When the "Up" button is pressed, the color yellow threshold is set to （20,90,-15,2,30,56）, and color detection is turned on. When a yellow card is detected, a yellow circle is displayed in the center of the screen.


Begin
"B" button is pressed
green threshold (20,80,-30,-15,15,30)
turned on color detection
Loop
green card is detected
A green circle is displayed
Finsh

Task 6 Nous recognizes numbers

When the program is running, turn on the MNIST (hand-written digital 0-9) detection. When Nous detects the number 0, it says "0" and displays the number 0 on the screen; when Nous detects the number 1, it says "1" and displays the number 1 on the screen...


Begin
Turn on MNIST detection
Loop
Detects the number 0
Detects the number 1
Detects the number 2
Detects the number...
Detects the number 9
Says "0" and displays the number 0
Says "1" and displays the number 1
Says "2" and displays the number 2
Says "..." an d displays th e number ...
Says "9" and displays the number 9
Finsh

Wrap-up

5 min

Students show and run the programs they wrote for Nous, and briefly describe which functions of Nous are used in each program.

03

Nous Loves to Draw

ISSUE: Nous Screen, Geometry, Spatial Reasoning, Art Design
CORE IDEA: Perception
BASIC LESSON
Difficulty: ★★★ Duration: 60 min
Data: None
Required Items: Nous Hub/ Nous Robot, PC or Pad, AprilTag cards

There is a 320x240, 2.4-inch touch screen on Nous. The screen doesn't just display numbers, text, variable information, etc., but also displays images. In this lesson, students will design patterns based on the resolution ratio and coordinate information for the Nous screen, and then display these patterns on the screen through programming. For example, concentric circles, Olympic rings, a truck, etc. Finally, program Nous so that when it sees a certain AprilTag card, it displays the corresponding pattern on its screen.

1 Identify a problem (5 min)

2 Guided Practice (25 min)

- Understand the concept of resolution ratio, and know that the resolution ratio of the Nous Hub' screen is 320×240.
- Analyze the method for displaying concentric circles on the Nous screen.
- Analyze the method for displaying Olympic rings on the Nous screen.
- Achieve animation effects.

3 Independent Practice (25 min)

- Practice programming to display different patterns on the Nous screen.
- Use AprilTag recognition to upgrade the program.

4 Wrap-up (5 min)

Learning outcomes:

TSWBAT: the students will be able to

- Understand the concept of resolution ratio, and know that the resolution ratio of the Nous Hub' screen is 320×240.
- Master the method of displaying patterns on the Nous Hub's screen.
- Be able to program to display concentric circles, Olympic rings, a truck, and Mickey Mouse on the Nous' screen.
- Further understanding of computer vision, practice programming to display the corresponding pattern when Nous sees certain AprilTag cards.

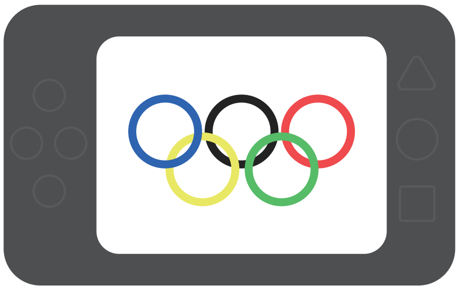
AI VOCA

- Perception: Nous Robot can "see" our commands.
- Natural Interaction: Nous Pet can understand the image signs given by humans and take corresponding actions, thereby realizing interaction with humans.

Identify a problem

5 min

There is a 320×240, 2.4-inch touch screen on Nous. The screen doesn't just display numbers, text, variable information, etc., but also displays images. In this lesson, students will design patterns based on the resolution ratio and coordinate information for the Nous screen, and then display these patterns on the screen through programming. For example, concentric circles, Olympic rings, a truck, etc. Finally, students also need to train a speech recognition model, and program Nous to display the corresponding pattern when Nous hears a certain wake-up word.



Guided Practice

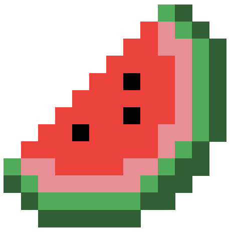
25 min

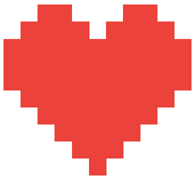
1. Understand the concept of resolution ratio, and know that the resolution ratio of the Nous Hub' screen is 320×240.

Before displaying something on the screen, we need to first understand how the screen displays patterns. Here, we must mention two important concepts: pixels and resolution.

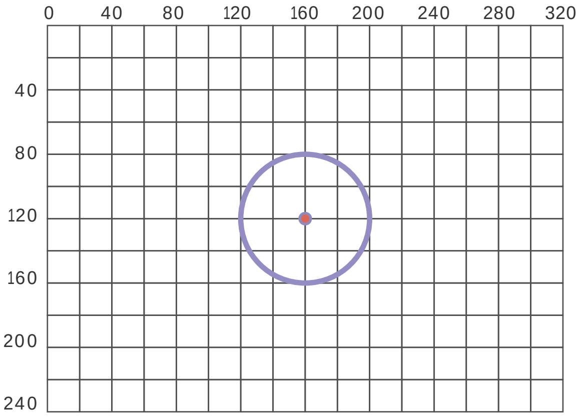
Pixel: simply stated means a point, many pixels can be combined to form the image we see.

Resolution ratio: Resolution ratio is used to indicate how many pixels there are horizontally and vertically.

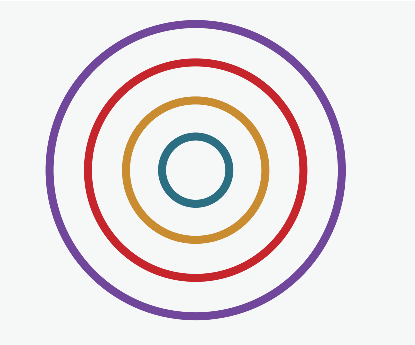


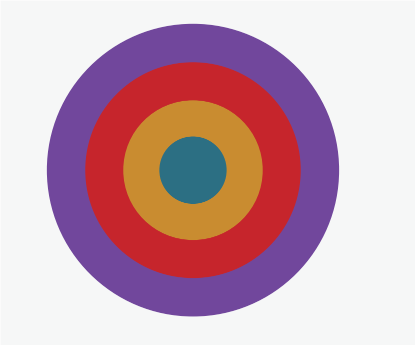


The resolution ratio of the Nous screen is 320x240. Based on this information and the knowledge of coordinates, we can determine how to display some basic patterns on the screen. For example, use the center point of the screen as the circle point to draw a circle with a radius of 40. Then, the coordinate information of this center point is: X=160, Y=120.

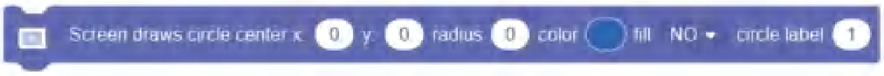


2. Analyze the method of displaying concentric circles on the Nous screen.


Screenshot 4-1-1


Screenshot 4-1-2

To draw the two types of concentric circles in the pictures above, use the program block in the screenshot below.

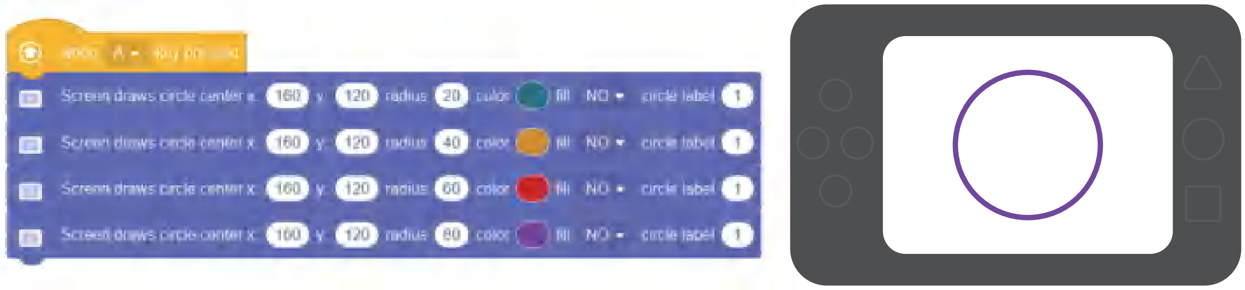


```
Screen draws circle center x (0) y (0) radius (0) color (●) fill NO ▾ circle label (1)
```

In this program block, "X" and "Y" are the coordinate information of the circle center; "radius" refers to the radius of the circle (unit is pixels); "fill" can either be "No" or "Yes", which refers to whether the circle is filled or not. The circle label here is very important. Usually if we need to draw multiple circles on the screen, we need to fill in different labels for these circles.

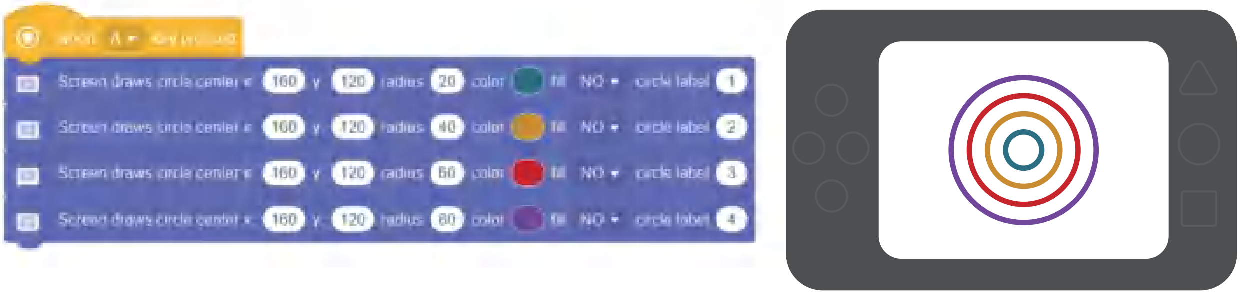
Since the resolution of the Nous screen is 320x240, we can determine that the coordinate information of the center point is: X=160, Y=120. We use the center point of the screen as the circle's center for the concentric circles, and set the radii of the circles in the concentric circles to 20, 40, 60 and 80, respectively.

First, we display concentric circles with no fill color. Here, pay special attention to changing the circle label. Compare the two sets of programs in the figures below: When the circle labels of the four circles are all set to 1, only the last circle in the program is displayed on the screen; when the circle labels of the four circles are set to 1, 2, 3, and 4 then the concentric circles will be displayed on the screen.



```
when A ▾ key pressed
Screen draws circle center x (160) y (120) radius (20) color (●) fill NO ▾ circle label (1)
Screen draws circle center x (160) y (120) radius (40) color (●) fill NO ▾ circle label (1)
Screen draws circle center x (160) y (120) radius (60) color (●) fill NO ▾ circle label (1)
Screen draws circle center x (160) y (120) radius (80) color (●) fill NO ▾ circle label (1)
```

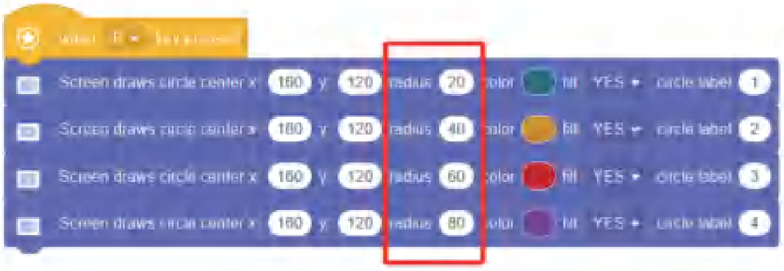
Screenshot 4-2-1



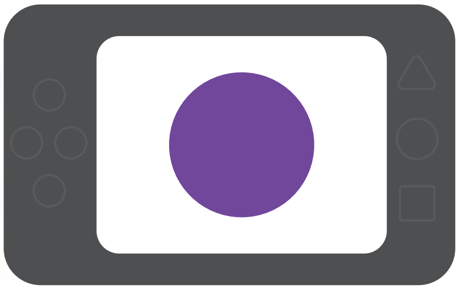
```
when A ▾ key pressed
Screen draws circle center x (160) y (120) radius (20) color (●) fill NO ▾ circle label (1)
Screen draws circle center x (160) y (120) radius (40) color (●) fill NO ▾ circle label (2)
Screen draws circle center x (160) y (120) radius (60) color (●) fill NO ▾ circle label (3)
Screen draws circle center x (160) y (120) radius (80) color (●) fill NO ▾ circle label (4)
```

Screenshot 4-2-2

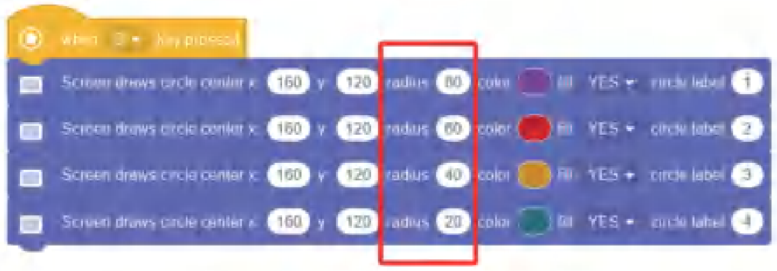
Next, we display the concentric circles with color fill. Pay special attention to the order of these circles. Compare the two programs in the figure below: When the order of the four color-filled circles goes from small to large, the last color-filled circle in the program is displayed on the screen, and it which covers over the previous three circles (See Screenshot 4-3-1). When the four filled-color circles are ordered from largest to smallest, then only the concentric circles filled with colors will be displayed on the screen (See Screenshot 4-3-2).



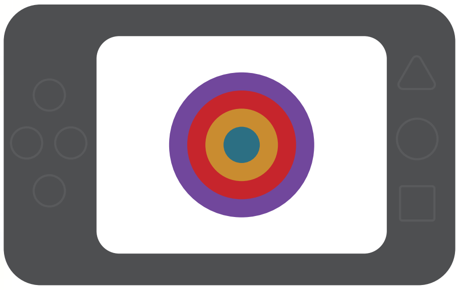
```
when [illegible] key pressed
Screen draws circle center x 160 y 120 radius 20 color 🟩 fill YES circle label 1
Screen draws circle center x 160 y 120 radius 40 color 🟧 fill YES circle label 2
Screen draws circle center x 160 y 120 radius 60 color 🟥 fill YES circle label 3
Screen draws circle center x 160 y 120 radius 80 color 🟪 fill YES circle label 4
```



Screenshot 4-3-1

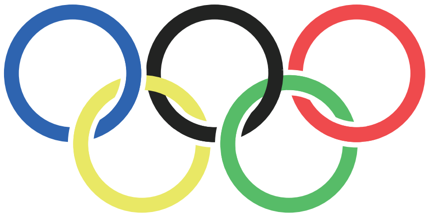


```
when [illegible] key pressed
Screen draws circle center x 160 y 120 radius 80 color 🟪 fill YES circle label 1
Screen draws circle center x 160 y 120 radius 60 color 🟥 fill YES circle label 2
Screen draws circle center x 160 y 120 radius 40 color 🟧 fill YES circle label 3
Screen draws circle center x 160 y 120 radius 20 color 🟩 fill YES circle label 4
```

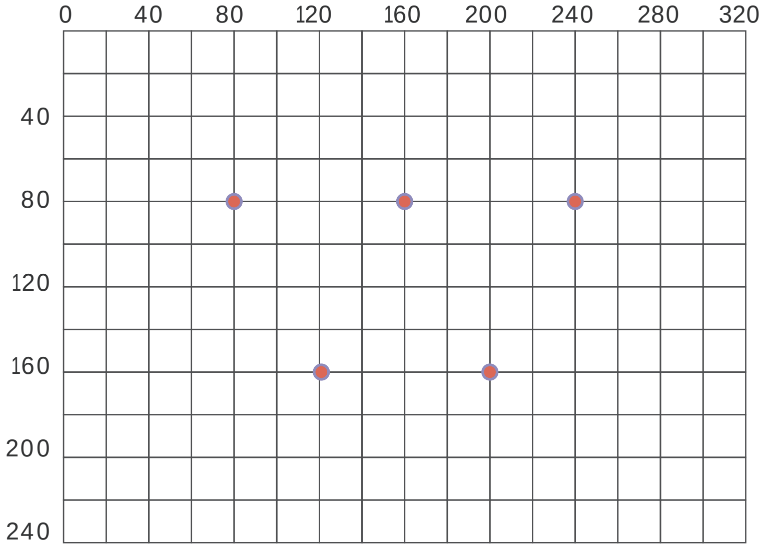


Screenshot 4-3-2

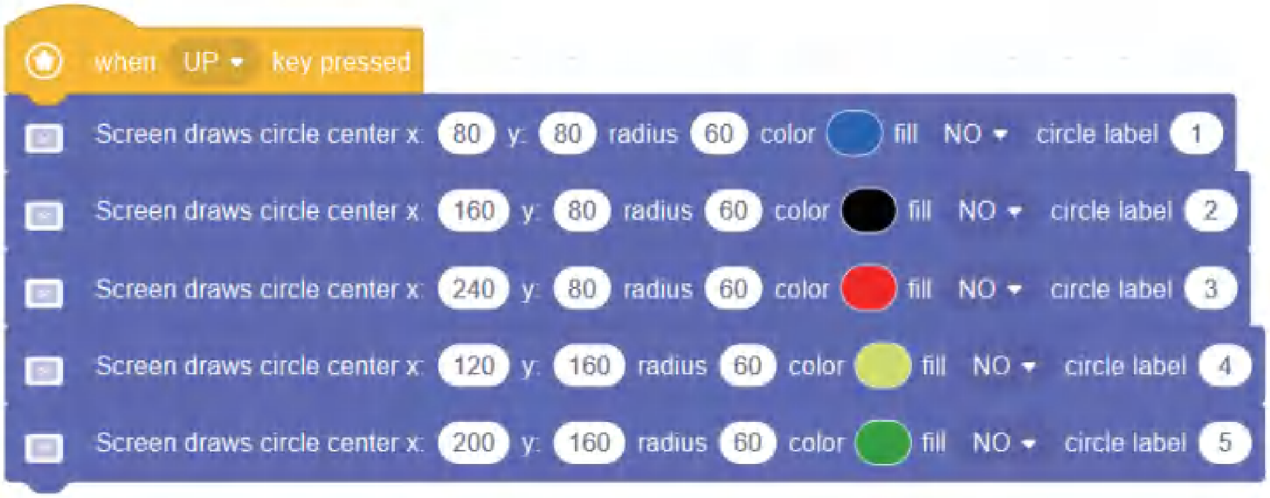
3. Analyze the method of displaying the Olympic rings on the Nous screen.



Now, let's analyze the method for displaying the Olympic rings on the screen. The first step is to locate the coordinates for the center of each Olympic ring; the second step is to set the appropriate radius length according to the screen size and the position of each circle's center; the third step is to set the color of each circle.



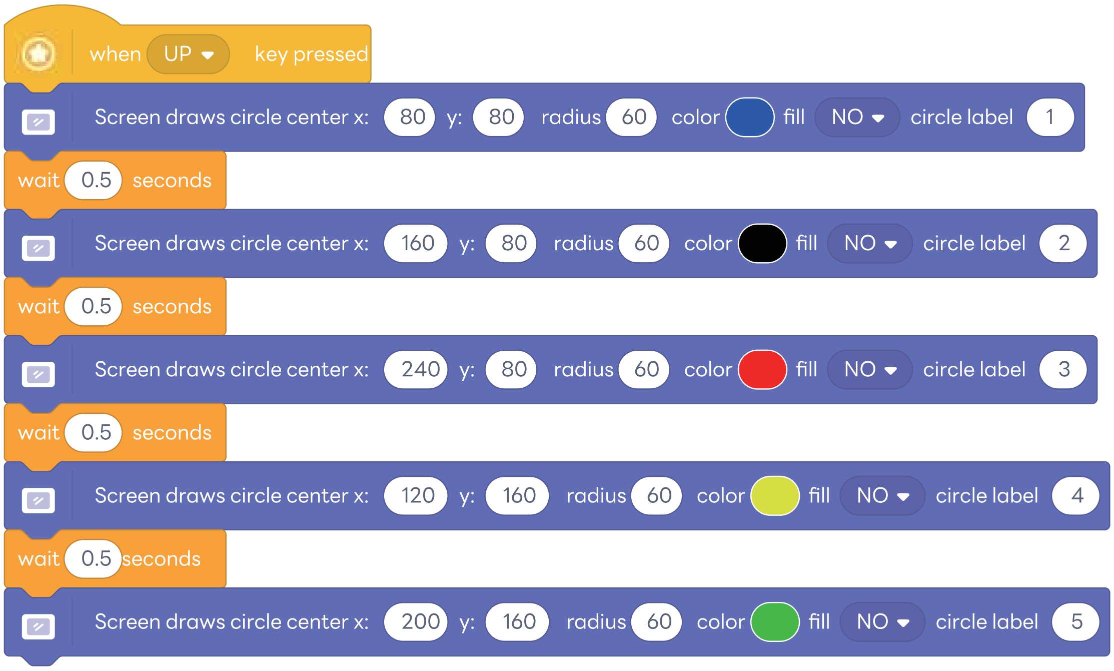
Demo Program:



```
when UP ▾ key pressed
Screen draws circle center x: 80 y: 80 radius 60 color 🔵 fill NO ▾ circle label 1
Screen draws circle center x: 160 y: 80 radius 60 color ⚫ fill NO ▾ circle label 2
Screen draws circle center x: 240 y: 80 radius 60 color 🔴 fill NO ▾ circle label 3
Screen draws circle center x: 120 y: 160 radius 60 color 🟡 fill NO ▾ circle label 4
Screen draws circle center x: 200 y: 160 radius 60 color 🟢 fill NO ▾ circle label 5
```

4. Achieve animation effects.

If we want to achieve animation effects when displaying these patterns, we only need to add appropriate waiting time between each geometric figure. For example,an animation effect program for the Olympic rings is shown below.

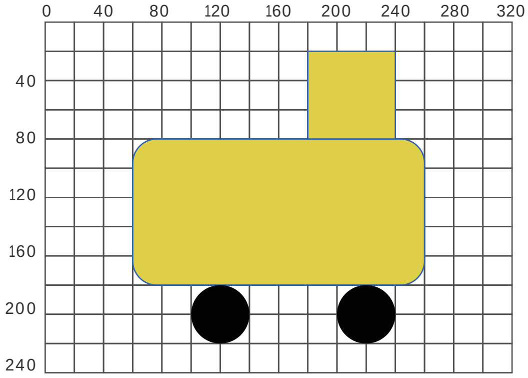


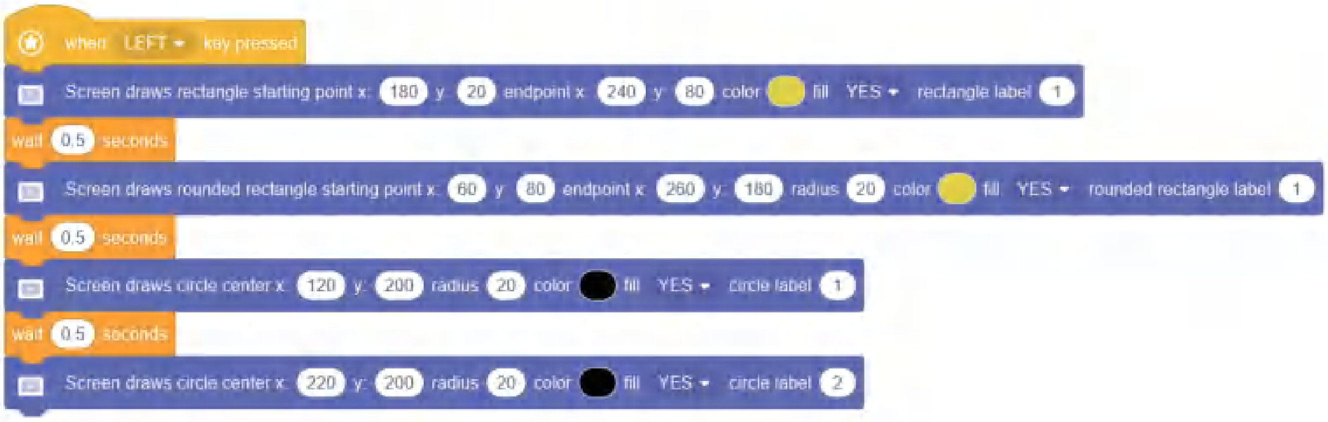
```
when UP ▾ key pressed
Screen draws circle center x: 80 y: 80 radius 60 color 🔵 fill NO ▾ circle label 1
wait 0.5 seconds
Screen draws circle center x: 160 y: 80 radius 60 color ⚫ fill NO ▾ circle label 2
wait 0.5 seconds
Screen draws circle center x: 240 y: 80 radius 60 color 🔴 fill NO ▾ circle label 3
wait 0.5 seconds
Screen draws circle center x: 120 y: 160 radius 60 color 🟡 fill NO ▾ circle label 4
wait 0.5 seconds
Screen draws circle center x: 200 y: 160 radius 60 color 🟢 fill NO ▾ circle label 5
```

Independence Practice

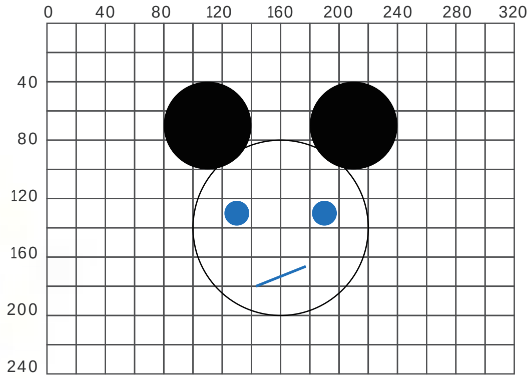
25 min

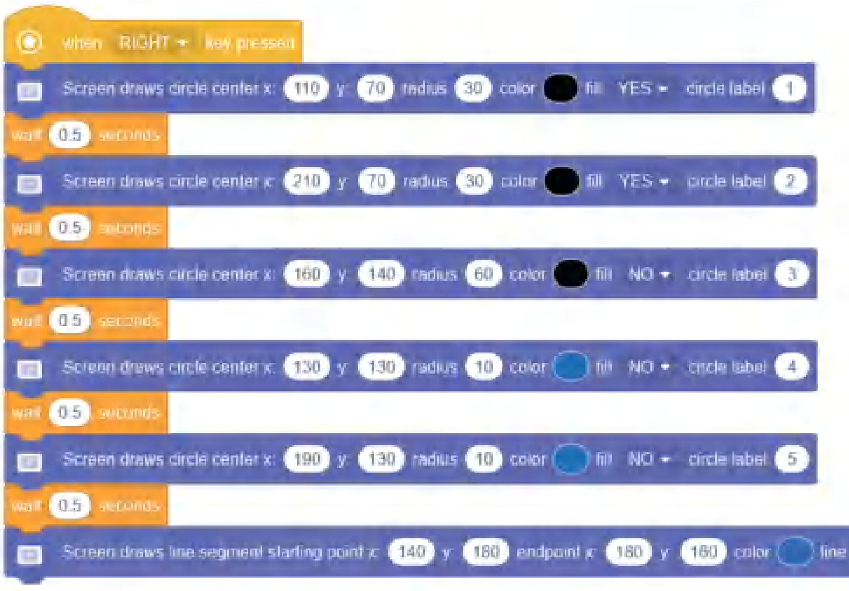
1. Students follow the prompts on the worksheet, and program to display concentric circles, Olympic rings, a truck and Mickey Mouse on the Nous screen.





```
when LEFT ▾ key pressed
Screen draws rectangle starting point x: 180 y: 20 endpoint x: 240 y: 80 color ● fill YES ▾ rectangle label 1
wait 0.5 seconds
Screen draws rounded rectangle starting point x: 60 y: 80 endpoint x: 260 y: 180 radius 20 color ● fill YES ▾ rounded rectangle label 1
wait 0.5 seconds
Screen draws circle center x: 120 y: 200 radius 20 color ● fill YES ▾ circle label 1
wait 0.5 seconds
Screen draws circle center x: 220 y: 200 radius 20 color ● fill YES ▾ circle label 2
```



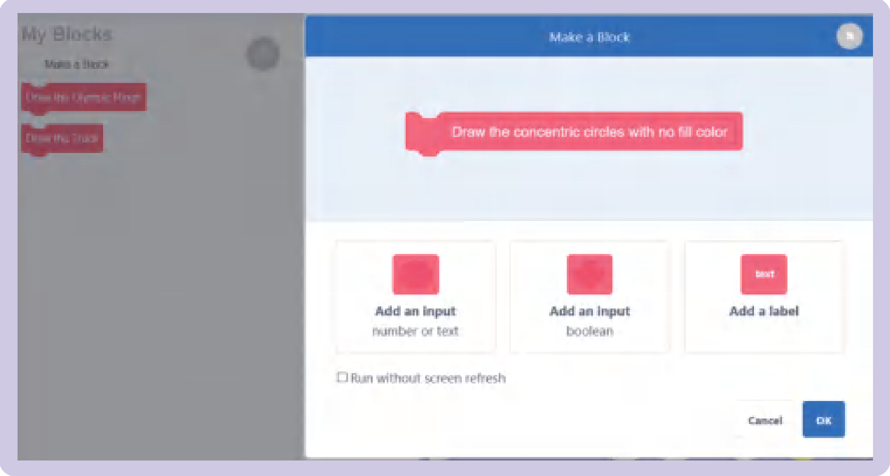


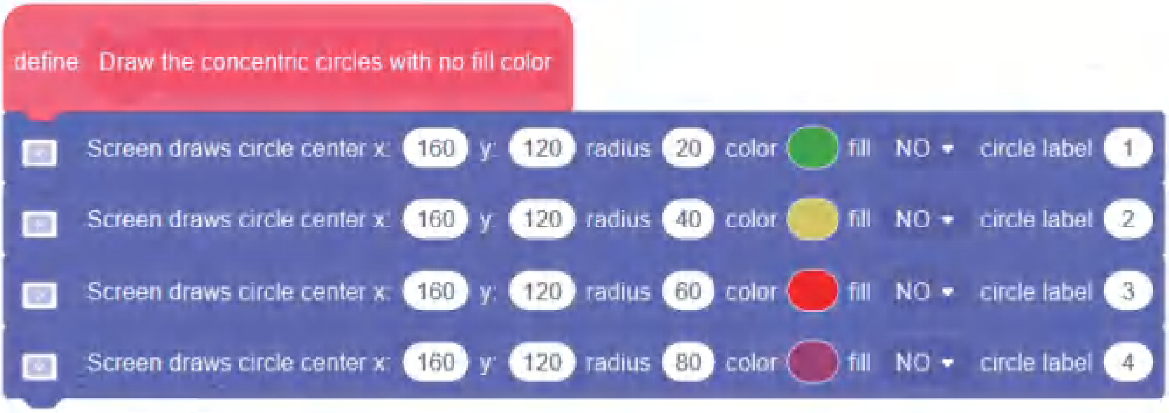
```
when RIGHT ▾ key pressed
Screen draws circle center x: 110 y: 70 radius 30 color ● fill YES ▾ circle label 1
wait 0.5 seconds
Screen draws circle center x: 210 y: 70 radius 30 color ● fill YES ▾ circle label 2
wait 0.5 seconds
Screen draws circle center x: 160 y: 140 radius 60 color ● fill NO ▾ circle label 3
wait 0.5 seconds
Screen draws circle center x: 130 y: 130 radius 10 color ● fill NO ▾ circle label 4
wait 0.5 seconds
Screen draws circle center x: 190 y: 130 radius 10 color ● fill NO ▾ circle label 5
wait 0.5 seconds
Screen draws line segment starting point x: 140 y: 180 endpoint x: 180 y: 160 color ● line segment label 1
```

2. Program so that when Nous sees a certain AprilTag card, it displays the corresponding pattern on its screen.

AprilTag 1	The concentric circles with no fill color
AprilTag 2	The concentric circles with filled color
AprilTag 3	The Olympic Rings
AprilTag 4	The truck
AprilTag 5	The Mickey Mouse

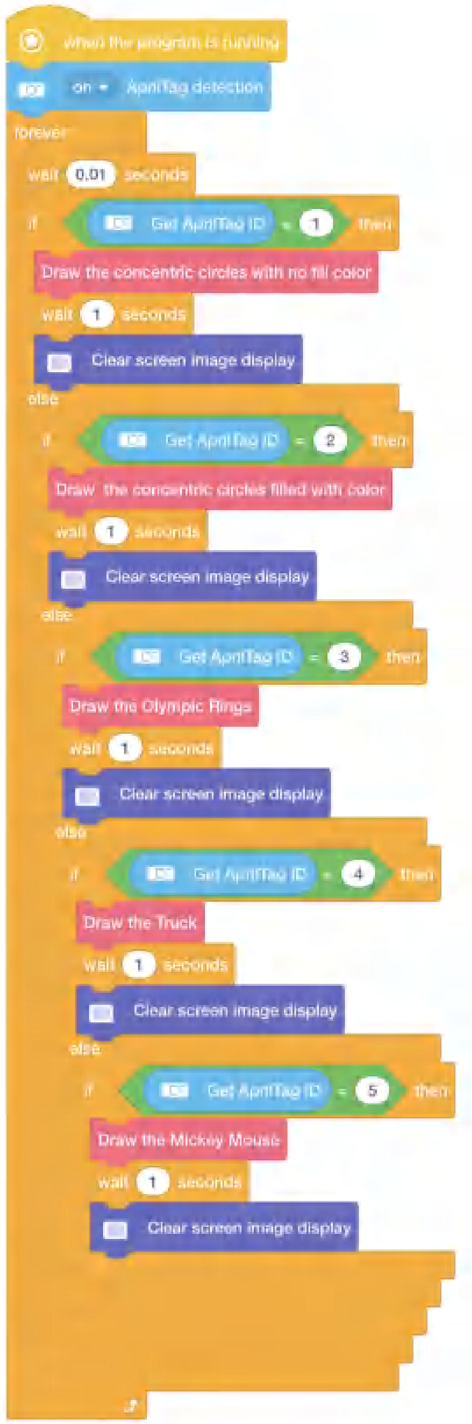
To facilitate program writing, we can first use the "Make blocks" coding block to make the programs for drawing the above five patterns into five new blocks. Example is shown below.





```
define Draw the concentric circles with no fill color
Screen draws circle center x: 160 y: 120 radius 20 color 🟢 fill NO ▾ circle label 1
Screen draws circle center x: 160 y: 120 radius 40 color 🟡 fill NO ▾ circle label 2
Screen draws circle center x: 160 y: 120 radius 60 color 🔴 fill NO ▾ circle label 3
Screen draws circle center x: 160 y: 120 radius 80 color 🟣 fill NO ▾ circle label 4
```

Demo Program:



```
when the program is running
on ▾ AprilTag detection
forever
  wait 0.01 seconds
  if Get AprilTag ID = 1 then
    Draw the concentric circles with no fill color
    wait 1 seconds
    Clear screen image display
  else
    if Get AprilTag ID = 2 then
      Draw  the concentric circles filled with color
      wait 1 seconds
      Clear screen image display
    else
      if Get AprilTag ID = 3 then
        Draw the Olympic Rings
        wait 1 seconds
        Clear screen image display
      else
        if Get AprilTag ID = 4 then
          Draw the Truck
          wait 1 seconds
          Clear screen image display
        else
          if Get AprilTag ID = 5 then
            Draw the Mickey Mouse
            wait 1 seconds
            Clear screen image display
```

Wrap-up

5 min

In this lesson, we learned the concepts of pixels and resolution. After getting familiar with the coordinate information on the Nous screen, we programmed it to display patterns such as concentric circles, Olympic rings,a truck and Mickey Mouse on the screen. Finally, the program was upgraded by combining the AprilTag recognition to further experience computer vision.