AI Mall

EXPERIMENT

Programming Platform：DobotScartch
Hardware：Magician Lite

DOBOT

Shake Hands With The Future

CATALOGUE

Experiment 1

Identity Authentication

Team Name: Team Member: Date:

Overview

How do you recognize a person? It is, of course, based on his face shape and features of five sense organs. Computers can also recognize faces. The technology of identifying people based on their facial features is called face recognition. The facial features mainly refer to face shape, features of five sense organs, etc. In this experiment, students will learn the principle and process of face recognition from two tasks: how to build a face database and how to use a computer to identify faces and simulate the face recognition-based attendance system.

Objective

1. Understand the face recognition technology.
2. By establishing the database, master the steps to build a face database.
3. Achieve the timing function by instructions of the timer.
4. Build a face recognition-based attendance system.

Requirements

- Take care when using electricity.
- Before the experiment, check whether the experimental equipment is complete and intact. If there is any omission or damage, please report to the teacher.
- Any specific operations during the experiment shall be performed in accordance with the lab manual. Ask your teacher when you have any question.

- During the experiment, the joints will start to work as the robotic arms are powered on. In that case, do not move the joints of the robotic arms hard if you do not press the unlock key.
- Report any device fault during the experiment to your teacher in a timely manner, and do not handle it yourself.
- Arrange all devices after the experiment. You shall not leave the lab before check by the group leader.

Task 1: Create A Face Database

To realize face recognition, build a corresponding face database which contains face images with labels. For example, the label of an identity attribute database for face recognition is the name that corresponds to a face, and that of a gender database for face recognition is gender. Let's look at how the face database is built.

1. Analysis

Look at the face database creation interface in Scratch, analyze the steps to build the database, and then fill in the table below.

Step 1: Locate the face database module on the Scratch software.

Step 2: Collect images.

Step 3: Add __________.

Step 4: _________ model.

Step 5: ________ model, and check the recognition rate of the model.

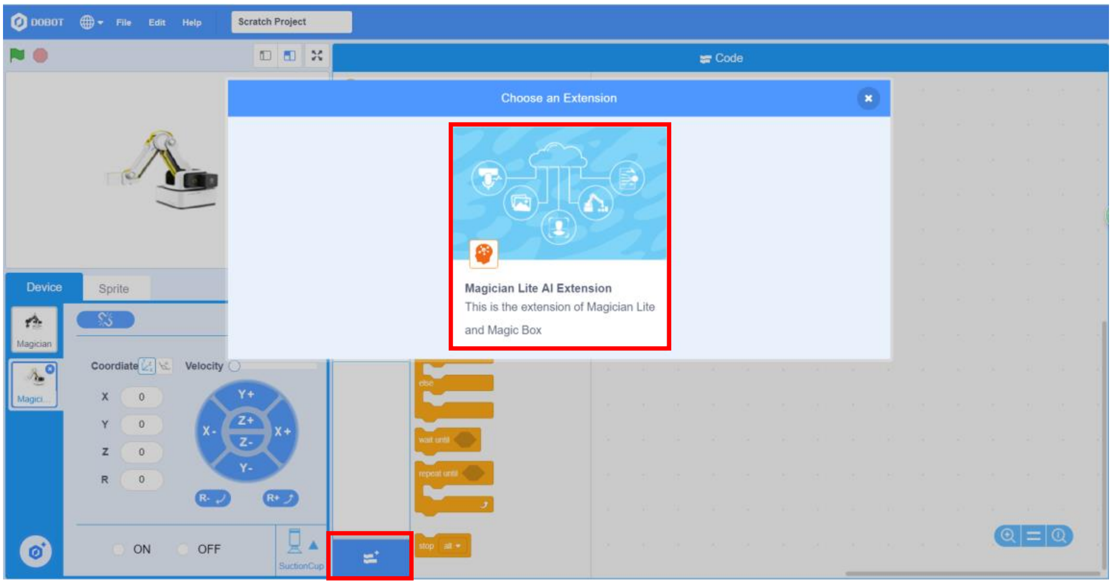
2. Steps

Face recognition is a biometric recognition technology based on key facial features. A series of relevant technologies, which collect images or video streams containing faces with video camera or camera, then detect and track the face in the image automatically, and finally recognize the detected face, are also usually known as portrait recognition or facial recognition.

Step 1: Start the DobotScratch software.

Step 2: Select the **Magician Lite** device.

Step 3: Click the **Add Extension** button to add an AI module, as shown in Figure 1.1.



Choose an Extension

Magician Lite AI Extension

This is the extension of Magician Lite and Magic Box

Figure 1.1 Adding the AI extension module

Step 4: Click the **AI** tab, and select **New face data** under the Face Recognition header to build a new face database, as shown in Figure 1.2.

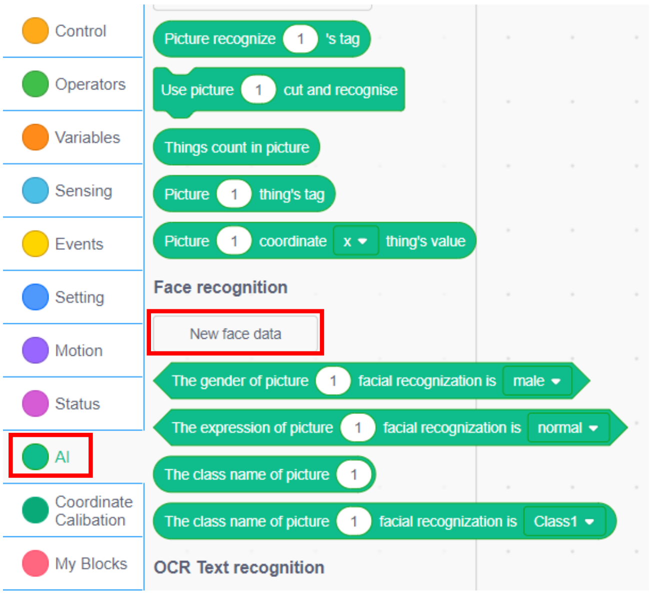

Control
Operators
Variables
Sensing
Events
Setting
Motion
Status
AI
Coordinate Calibation
My Blocks
Picture recognize 1 's tag
Use picture 1 cut and recognise
Things count in picture
Picture 1 thing's tag
Picture 1 coordinate x thing's value
Face recognition
New face data
The gender of picture 1 facial recognization is male
The expression of picture 1 facial recognization is normal
The class name of picture 1
The class name of picture 1 facial recognization is Class1
OCR Text recognition

Figure 1.2 Building the new face data

Step 5: Add facial features and data. Select the PC camera, aim the face to be recognized at the camera, edit the face label, and then click “**+**” to collect images, as shown in Figure 1.3.

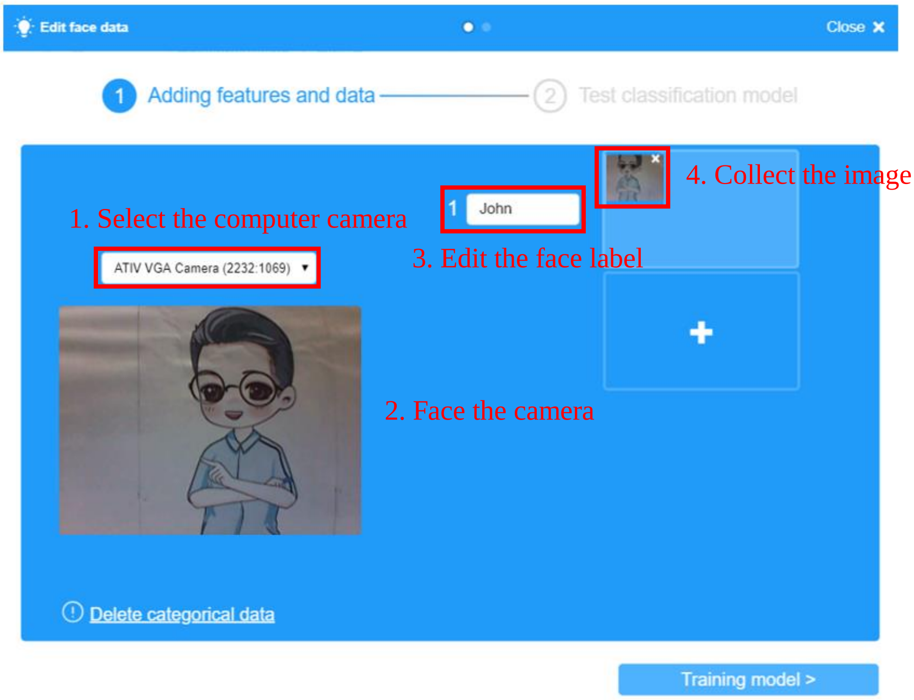


Figure 1.3 Adding the features and data

Step 6: Add the face label. Click "＋" to add a face label group, and collect the facial features and data of Group 2, as shown in Figure 1.4.

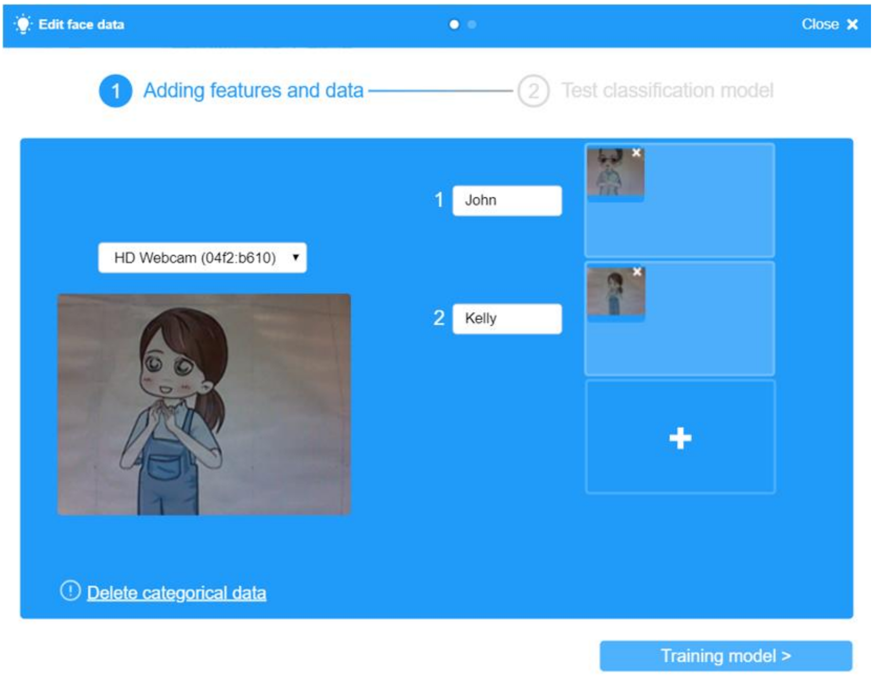


Figure 1.4 Adding the face label

Step 7: Train the model. Click the **Training model** button to train the face classification model, as shown in Figure 1.5.

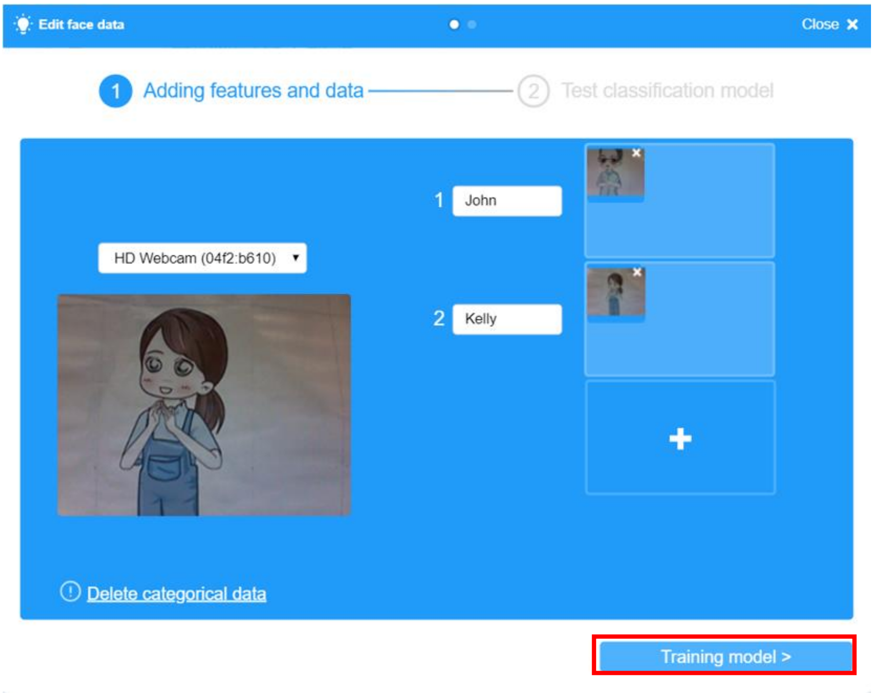


Figure 1.5 Training model

Step 8: Test the model. After completing the model training, shoot a portrait to test the accuracy of the classification model. Click **Test**, as shown in Figure 1.6.

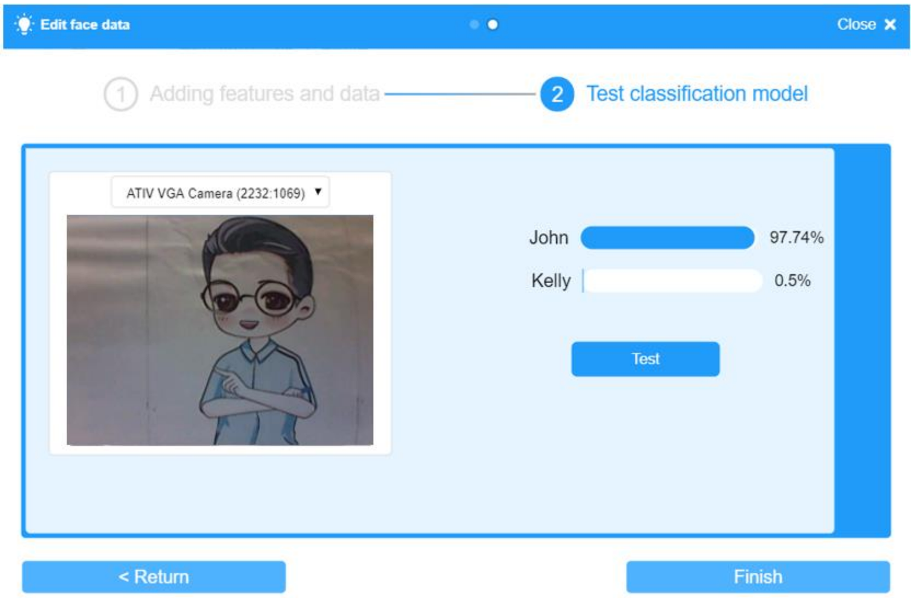

Edit face data
Close ✕
1 Adding features and data
2 Test classification model
ATIV VGA Camera (2232:1069)
John 97.74%
Kelly 0.5%
Test
< Return
Finish

Figure 1.6 Testing the classification model

Accuracy above 95% shows that the classification model performs better and can be used and the face database is built.

3. Summary

(1) Concept of face recognition: ______________________________
____

(2) Process to build a new face database: ______________________
____

(3) Methods for evaluating the performance of face classification model: ______________________________________
____

4. Self-Assessment

Check the completed content in the experiment task. Tick (√) the completed items, and circle (○) the uncompleted items.

Assessment Content	Completion Status
I've known what the face recognition is	
I've known the method for evaluating the performance of face classification model	
I've completed the task of building a new face database	

Task 2: Design A Face Recognition-Based Attendance System

The face recognition can be applied in many scenarios. Let's learn its application in attendance system and understand the face recognition-based attendance system.

1. Analysis

Demonstrate the face recognition-based attendance system, analyze the steps to authenticate identities, and then fill in the steps in the table below.

Step 1: Collect face images.

Step 2: Judge ______.

Step 3: Judge the identity of persons who are late if any, or who attend on time.

Step 4: ______ attendance result.

2. Steps

Step 1: Draw a flow chart by analyzing the identity authentication process, as shown in Figure 1.7.

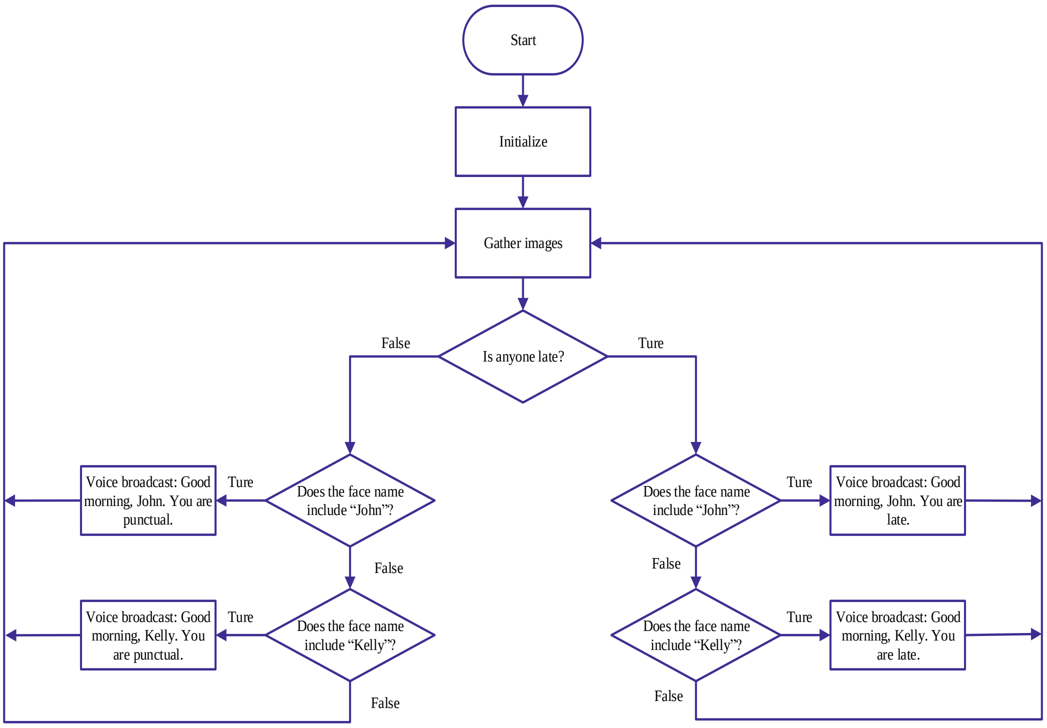

Start
Initialize
Gather images
Is anyone late?
False
Ture
Does the face name include "John"?
Ture
Voice broadcast: Good morning, John. You are punctual.
False
Does the face name include "Kelly"?
Ture
Voice broadcast: Good morning, Kelly. You are punctual.
False
Does the face name include "John"?
Ture
Voice broadcast: Good morning, John. You are late.
False
Does the face name include "Kelly"?
Ture
Voice broadcast: Good morning, Kelly. You are late.
False

Figure 1.7 Flow chart of the face recognition-based attendance system

Why do we use a timer?

Step 2: Initialize the timer. Click the **Sensing** tab, and drag out the "reset timer" building block, as shown in Figure 1.8.

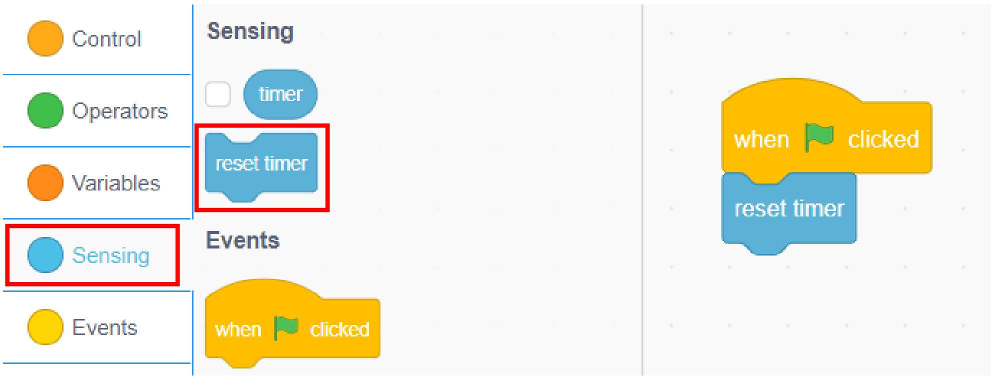


Figure 1.8 Initializing the timer

Step 3: Collect images. Click the **AI** tab, and drag out the "Timeout 3s to take picture" building block, as shown in Figure 1.9.

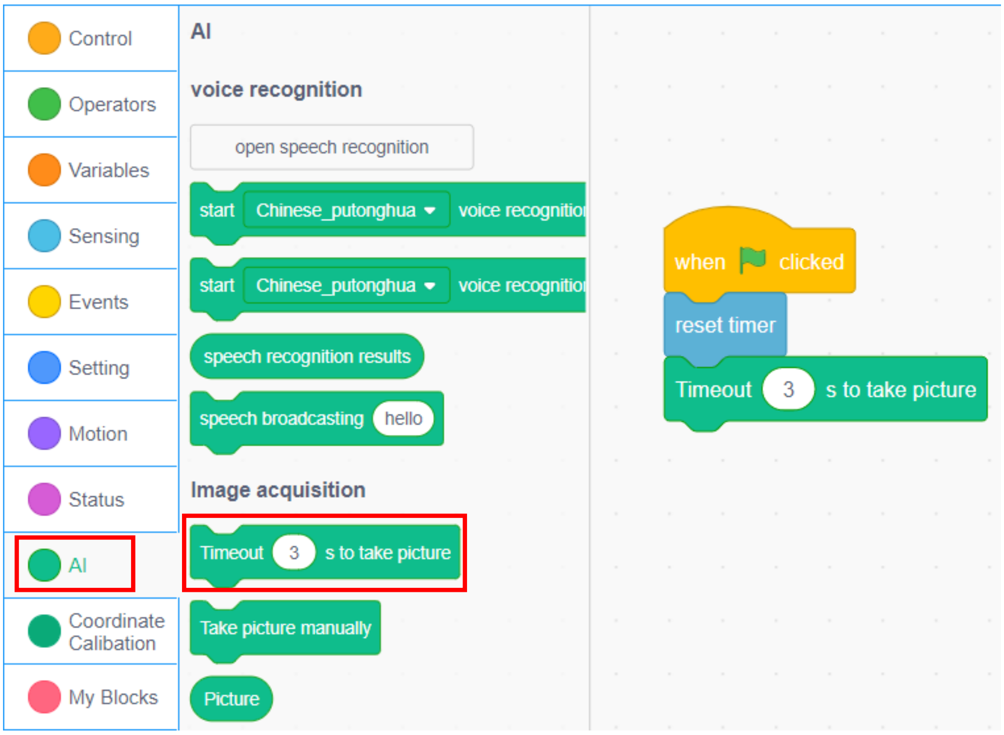


Figure 1.9 Collecting the images

Difference between automatic and manual shooting.

Step 4: Set the judgment criteria for late attendance. If a person scans the face over 20 seconds after the startup of the timer of the face recognition-based attendance system, he or she is judged as "late"; otherwise, he or she will be judged as "on-time". Compare the timer with the number 20, and click the **Operators** tab and drag out the “( )>( )” comparison block. Drag the “timer” block in the **Sensing** tab to the left elliptical blank in the comparison block, and fill in the right elliptical blank with 20, as shown in Figure 1.10.

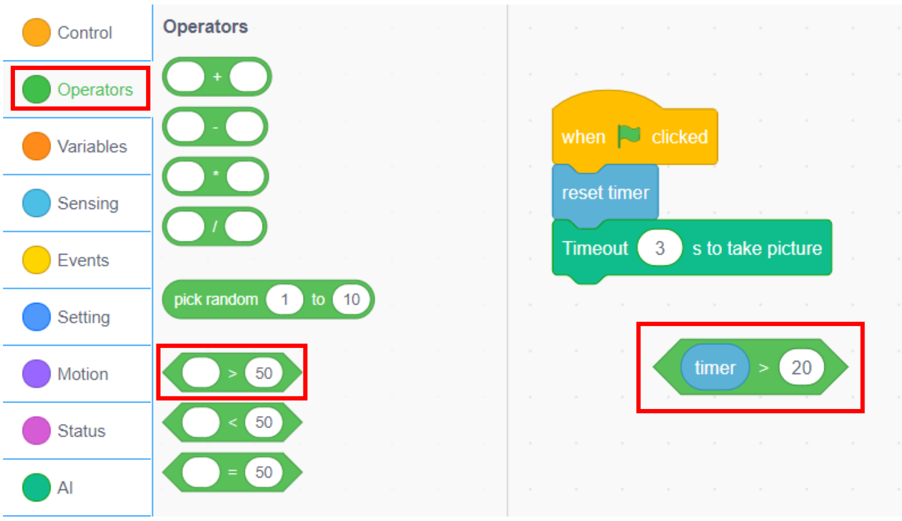


Figure 1.10 Setting judgement criteria for late attendance

☛ Try to develop the judgment criteria for late attendance with the “( )<( )” comparison block.

Step 5: Judge the “late” and “on-time” scenarios logically. Click the **Control** tab and drag out the “If... then else...” block, as shown in Figure 1.11.

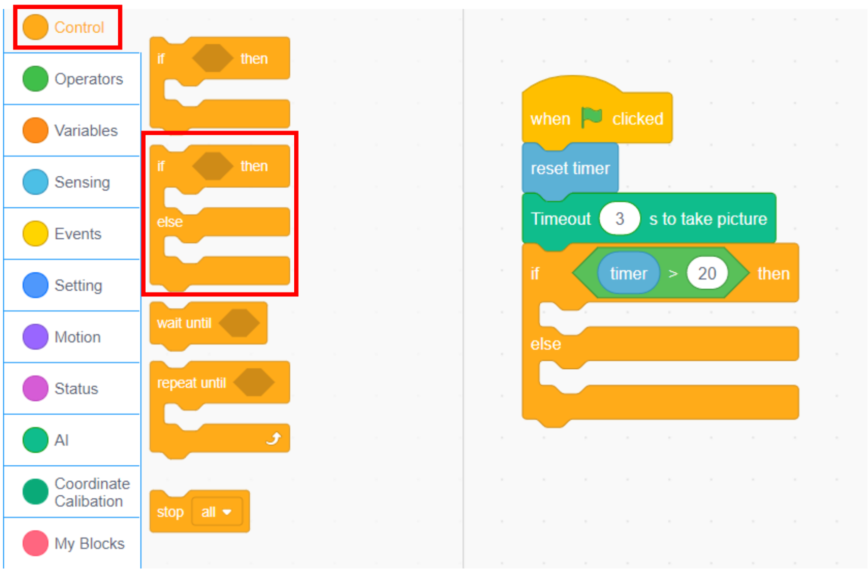
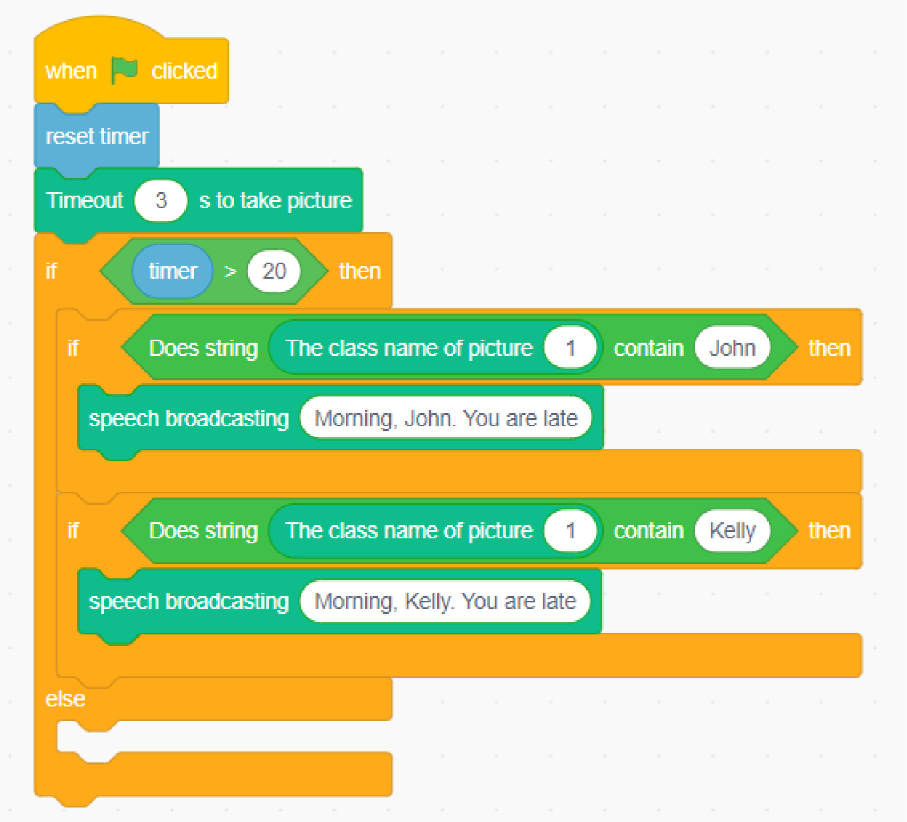


Figure 1.11 Making the logical Judgment

Step 6: Judge late attendance. Identify the late staff. If John is identified, a voice message “Morning, John. You are late” is triggered; if Kelly is identified, “Morning, Kelly. You are late” is triggered, as shown in Figure 1.12.

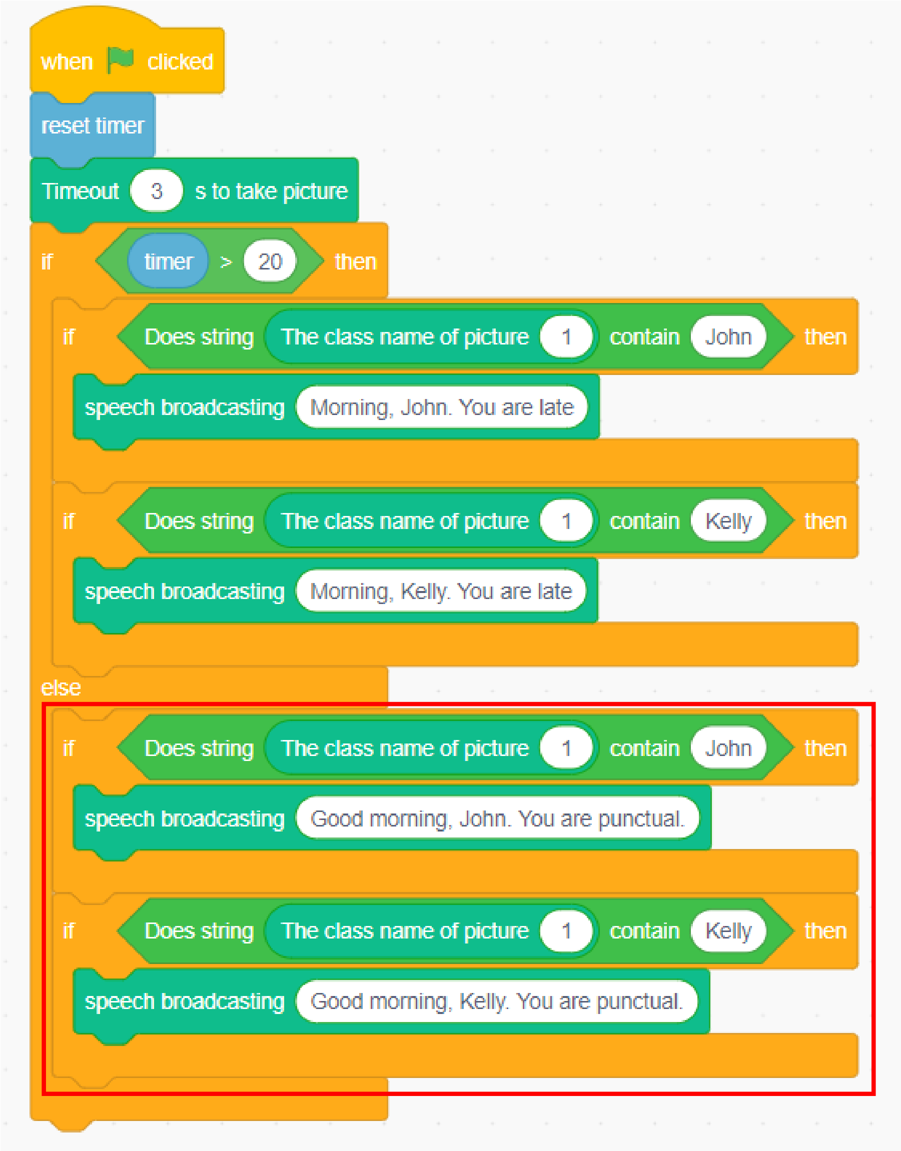


```
when ⚑ clicked
reset timer
Timeout (3) s to take picture
if <(timer) > (20)> then
    if <Does string (The class name of picture (1)) contain (John)> then
        speech broadcasting (Morning, John. You are late)
    if <Does string (The class name of picture (1)) contain (Kelly)> then
        speech broadcasting (Morning, Kelly. You are late)
else
```

Figure 1.12 Identifying late attendance

Step 7: Judge on-time attendance. The programming for on-time attendance is similar to that for late attendance, as shown in Figure 1.13.

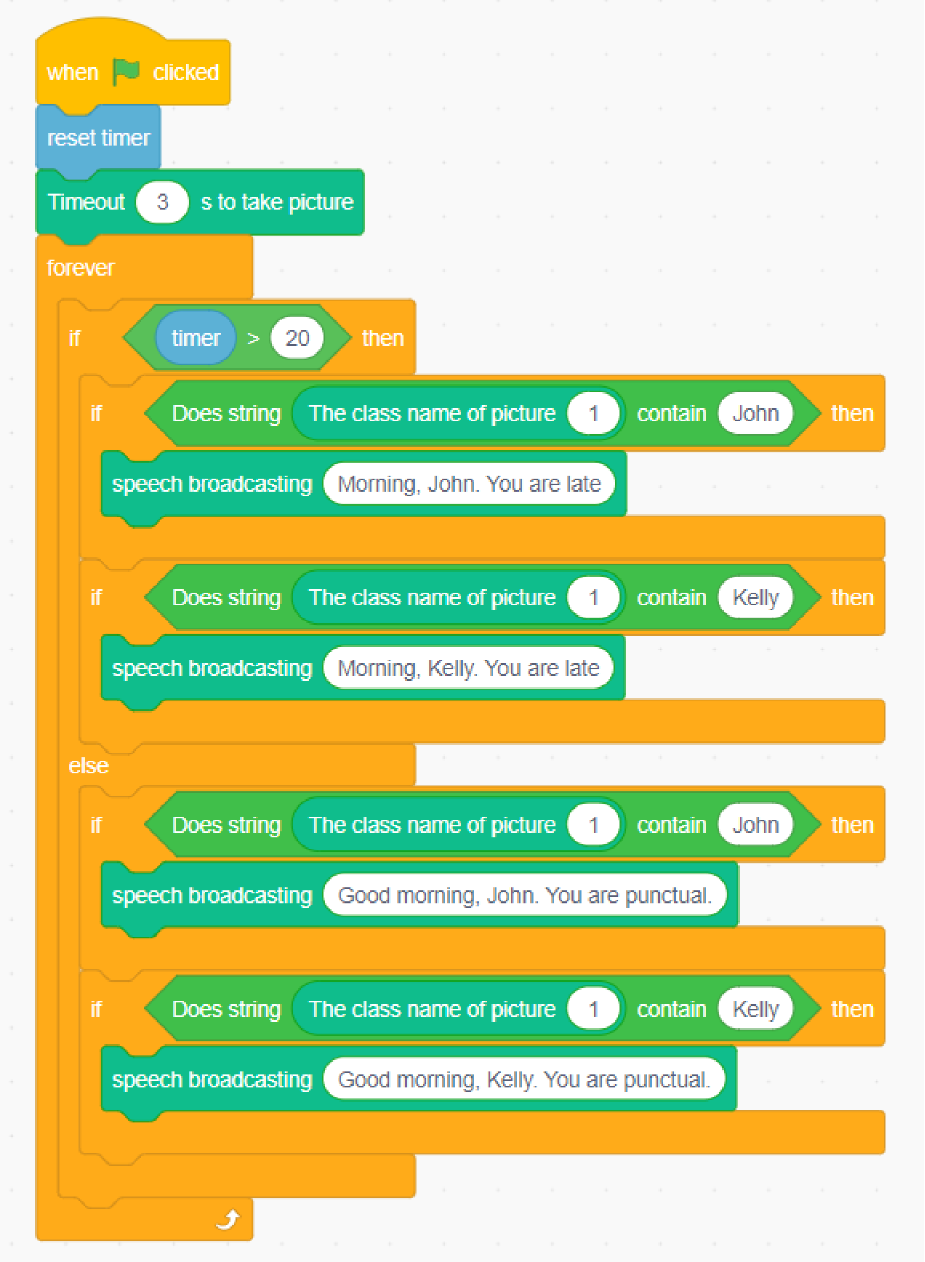
☛ Try to program for on-time attendance.



```
when ⚑ clicked
reset timer
Timeout (3) s to take picture
if <(timer) > (20)> then
    if <Does string (The class name of picture (1)) contain (John)> then
        speech broadcasting (Morning, John. You are late)
    if <Does string (The class name of picture (1)) contain (Kelly)> then
        speech broadcasting (Morning, Kelly. You are late)
else
    if <Does string (The class name of picture (1)) contain (John)> then
        speech broadcasting (Good morning, John. You are punctual.)
    if <Does string (The class name of picture (1)) contain (Kelly)> then
        speech broadcasting (Good morning, Kelly. You are punctual.)
```

Figure 1.13 Identifying on-time attendance

Step 8: Repeat the execution. Repeatedly judge and identify the latecomers, as shown in Figure 1.14.



```
when ⚑ clicked
reset timer
Timeout (3) s to take picture
forever
    if <(timer) > (20)> then
        if <Does string (The class name of picture (1)) contain (John)> then
            speech broadcasting (Morning, John. You are late)
        if <Does string (The class name of picture (1)) contain (Kelly)> then
            speech broadcasting (Morning, Kelly. You are late)
    else
        if <Does string (The class name of picture (1)) contain (John)> then
            speech broadcasting (Good morning, John. You are punctual.)
        if <Does string (The class name of picture (1)) contain (Kelly)> then
            speech broadcasting (Good morning, Kelly. You are punctual.)
```

Figure 1.14 Repeating the execution

3. Summary

(1) Use of “Timer” block: ____

____

(2) Advantages of face recognition-based attendance system:______

____

4. Self-Assessment

Check the completed content in the experiment task. Tick (√) the completed items, and circle (◯) the uncompleted items.

Assessment Content	Completion Status
I’ve used the “Timer” block	
I’ve known the face recognition process	
I’ve completed the programming for face recognition-based attendance system	

Experiment 2

Membership Identification

Team Name: Team Member: Date:

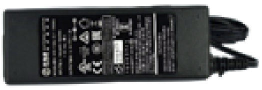
Overview

To improve the shopping experience of customers and precision marketing, many shopping centers provide membership services. During the traditional membership identification, customers need to provide their cellphone numbers or passwords for identification, which is relatively inefficient. This experiment simulates a more efficient method of membership identification, namely, identifying membership through face recognition.

Objective

1. Experience the application of face recognition by completing the task of membership identification.
2. Master the drawing method of stage Backdrops in DobotScratch by creating the member login system interface.
3. Master the drawing method of sprites in DobotScratch by creating the member login system interface.
4. Master the method of communication between sprites by programming the member login system.
5. Understand broadcast instructions by programming the member login system.

Equipment

Equipment Picture	Name	Quantity
	Dobot Magician Lite Robotic arm	1
	Suction cup	1
	Camera	1
	Power adapter	1
	USB cable	1

Requirements

- Take care when using electricity.
- Before the experiment, check whether the experimental device is complete and intact. If there is any omission or damage, please report to the teacher.
- Any specific operations in the experiment shall be performed according to the experiment manual. If you have any questions, please promptly ask the teacher.
- During the experiment, the joints will start to work as the robotic arms are powered on. In that case, do not move the joints of the robotic arms hard if you do not press the unlock key

- ➢ Report any device fault during the experiment to your teacher in a timely manner, and do not handle it yourself.
- ➢ Arrange all devices after the experiment. You shall not leave the lab before check by the group leader.

Task 1: Create the Member Login System Interface

To identify membership, we need to design an interface to interact with users. This interface contains member information and login operations. Now let’s design the member login interface.

1. Observation

Observe the member login interface designed by the teacher. Find what elements are necessary to create the interface, and then fill in the following blanks according to your observation.

The sprites required to create a member login interface include: **VIP Log In** button, interface backdrops, _______, level of membership, _____,_ , ,etc.

2. Steps

Step 1: Create Backdrops of the login interface. Start the DobotScratch software and click **Sprite**. Select **Stage** and then select **Backdrops** to draw the backdrops, as shown in Figure 2.1.

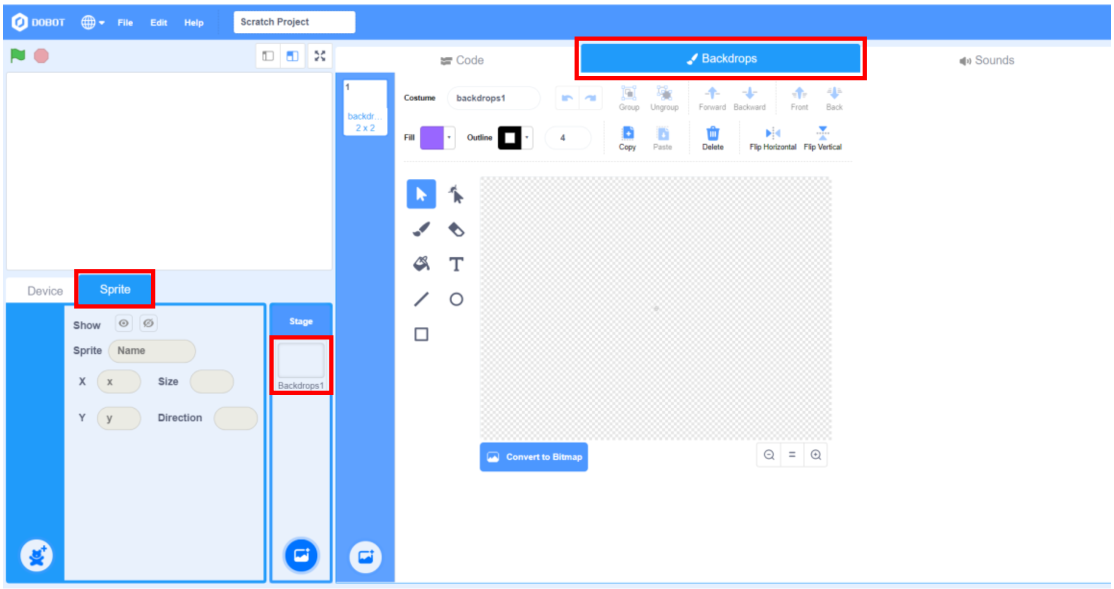


Figure 2.1 Selecting the drawing stage backdrops

Step 2: Click the Text command in the drawing toolbar and enter "Welcome to Smart Mall" at the center of the stage, as shown in Figure 2.2.

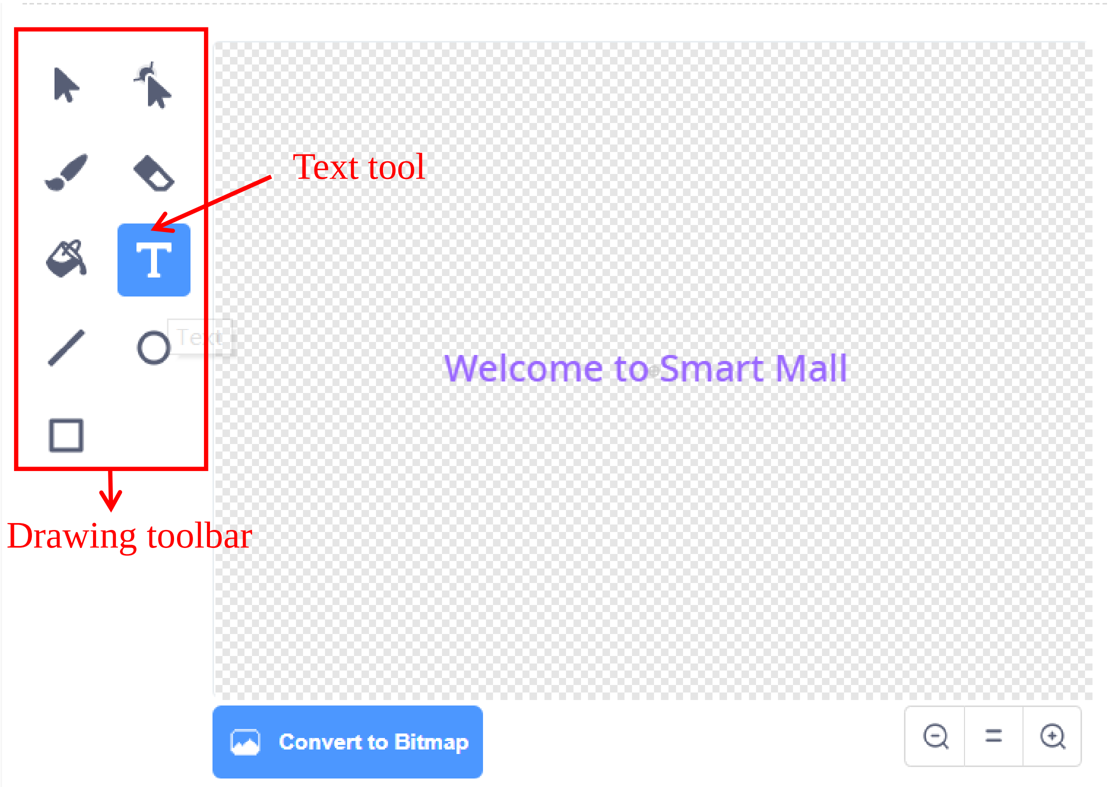


Figure 2.2 Drawing the stage backdrops

Step 3: Draw the login button. Select **Sprite**, click the bear icon, and then select the drawing command to create a new sprite, as shown in Figure 2.3.

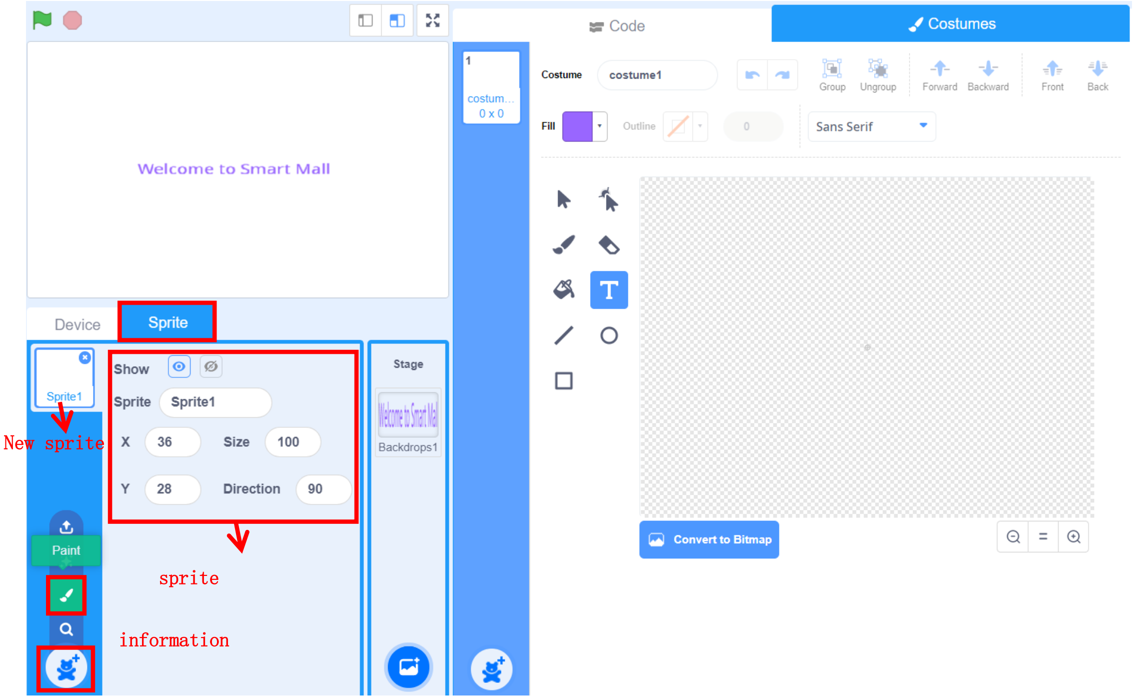


Figure 2.3 Steps to draw a new sprite

Step 4: Select the Circle command in the drawing toolbar to draw an ellipse, as shown in Figure 2.4.

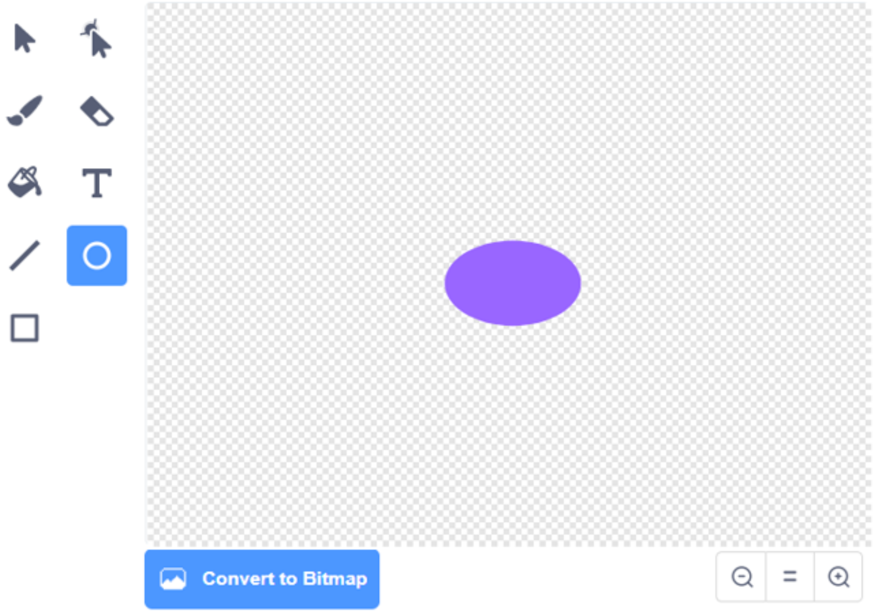


Figure 2.4 Drawing the ellipse

Step 5: Select the Text command, set the **Fill Color** to black, and enter “VIP Log In” at the center of the ellipse, as shown in Figure 2.5.

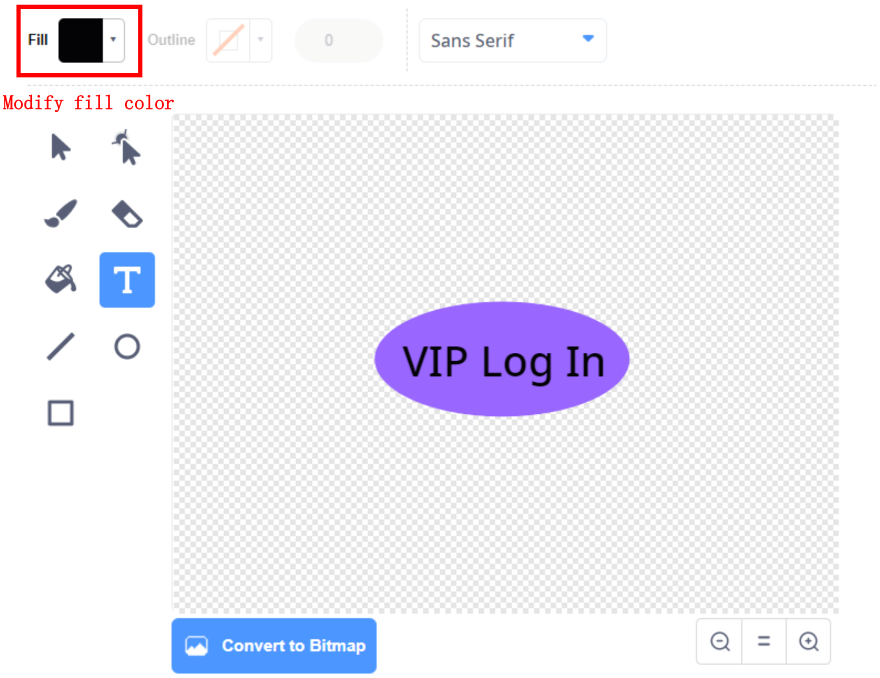


Figure 2.5 Creating the button name

Step 6: Move the button on the stage. Click and hold the button **Sprite** on the stage to drag it to a proper position, as shown in Figure 2.6.

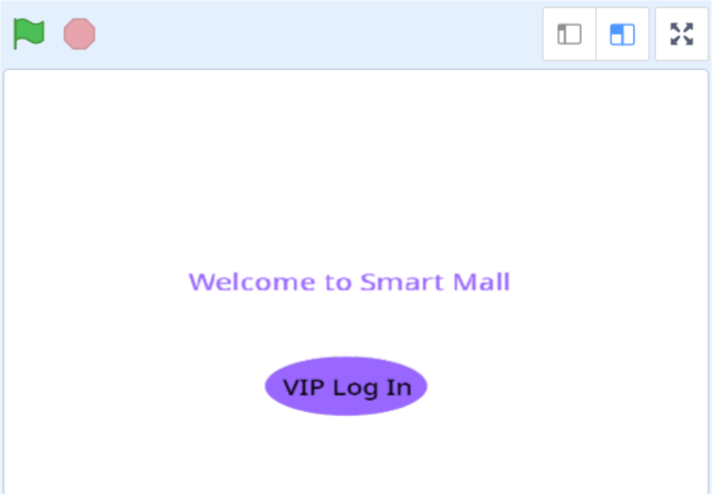
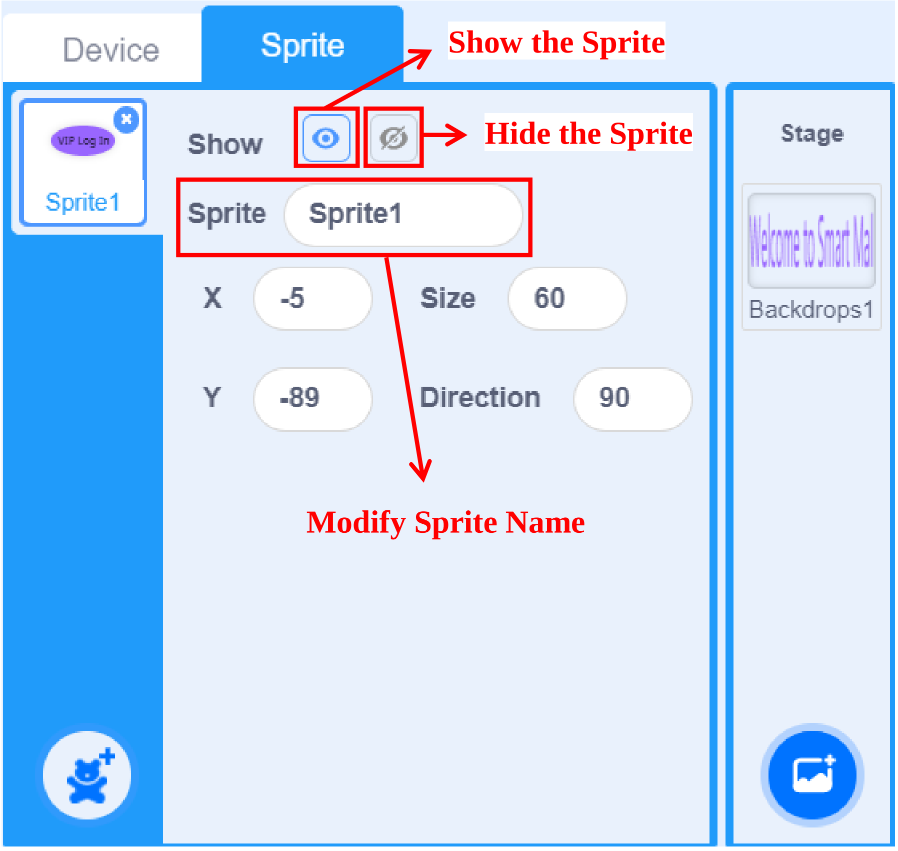


Figure 2.6 Dragging the login button sprite

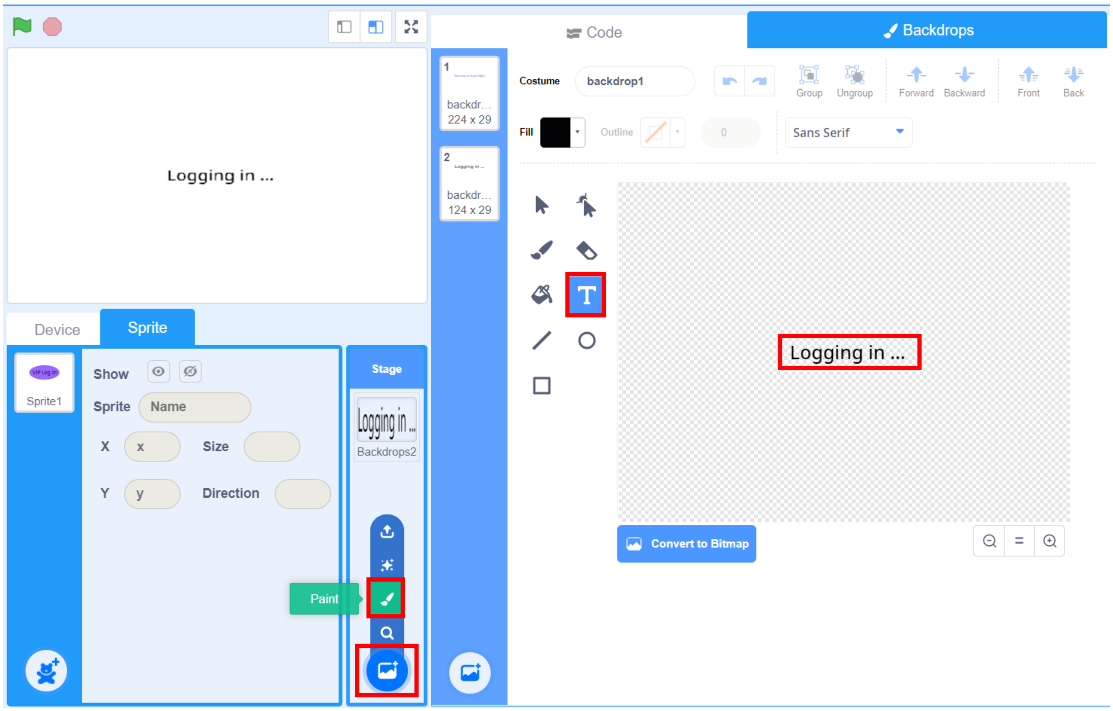
Step 7: Modify the sprite name of the **VIP Log In** button to **Log In**, and hide the sprite to draw other sprites and backdrops, as shown in Figure 2.7.



Show
Sprite Sprite1
X -5 Size 60
Y -89 Direction 90

Figure 2.7 Modifying sprite Information

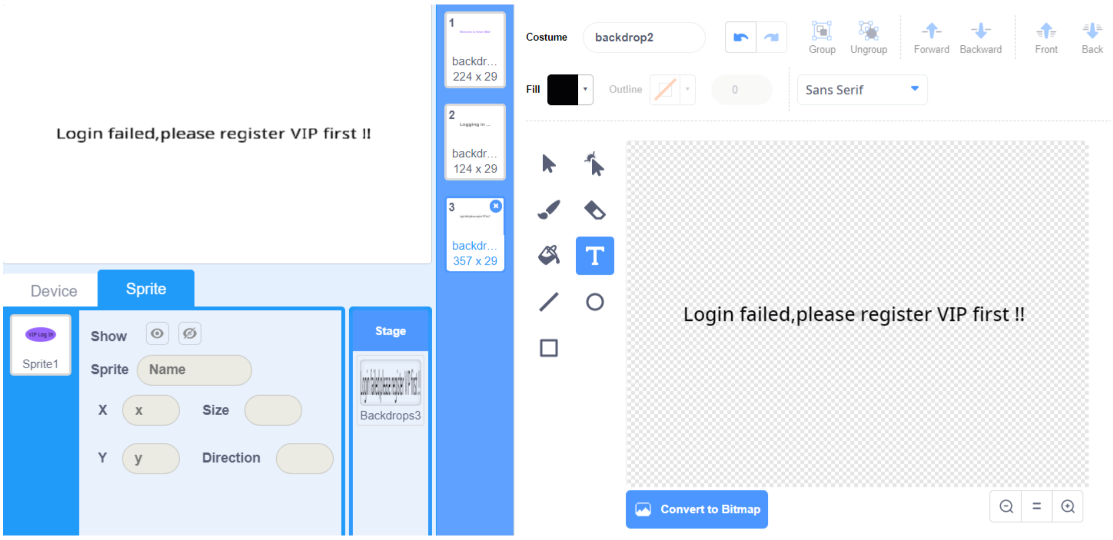
Step 8: Draw the login connection interface. Click **Stage** and select the drawing icon to draw a new backdrops. Draw “Logging in ...” on the new backdrops, as shown in Figure 2.8.



Show
Sprite Name
X x Size
Y y Direction
Costume backdrop1
Fill Outline 0 Sans Serif

Figure 2.8 Drawing the login connection interface

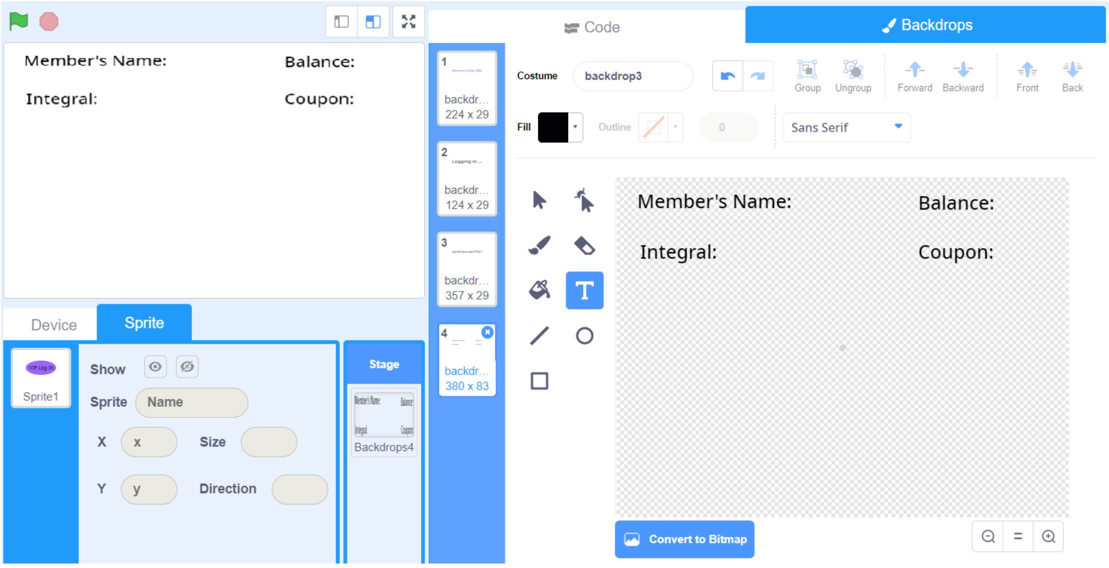
Step 9: Draw the login failure interface in the same way, as shown in Figure 2.9.



Login failed,please register VIP first !!

Figure 2.9 Drawing the login failure interface

Step 10: Draw the member information interface, as shown in Figure 2.10.

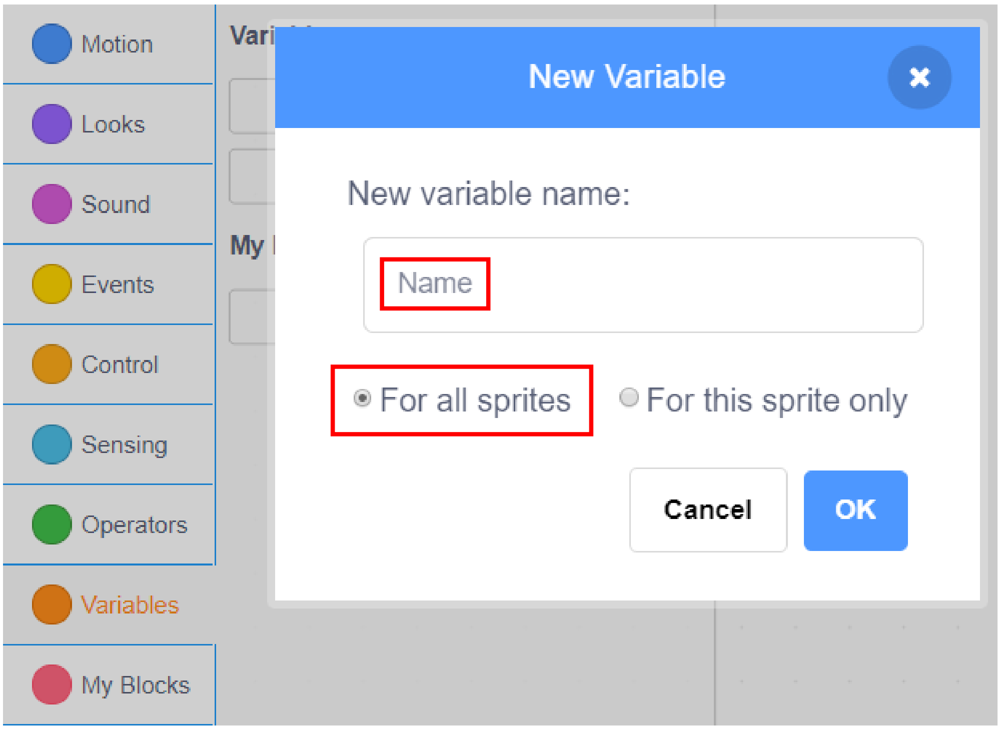


Member's Name: Balance:

Integral: Coupon:

Figure 2.10 Drawing the member Information Interface

Step 11: Define the *name* variable to store the member's name. Select the **Make a variable** option from the **Variables** tab, set the variable name to *name*, and then select **For all sprites**, as shown in Figure 2.11.



New Variable

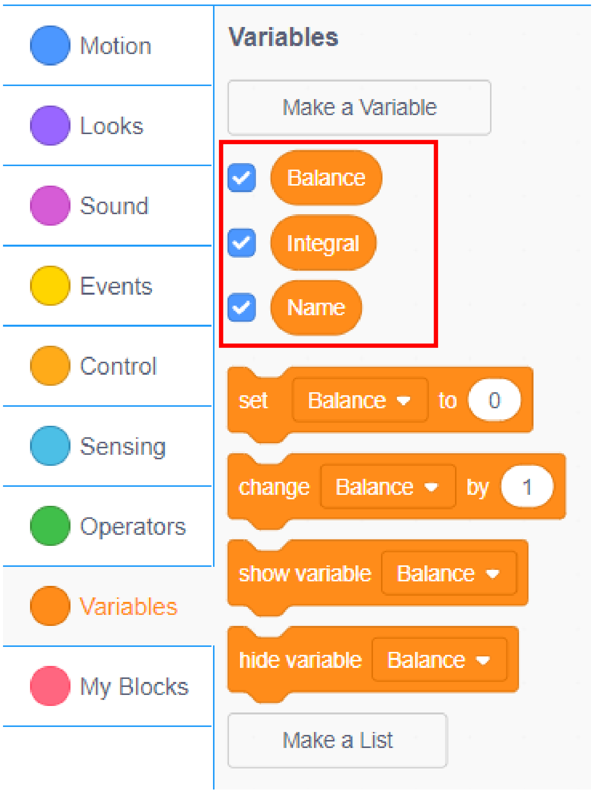
New variable name:

Name

● For all sprites ○ For this sprite only

Figure 2.11 Creating the *Name* variable

Step 12: Create new *balance* and *integral* variables in the same way, as shown in Figure 2.12.



☑ Balance
☑ Integral
☑ Name

set Balance ▾ to 0
change Balance ▾ by 1
show variable Balance ▾
hide variable Balance ▾

Figure 2.12 Creating other variables

Step 13: Hide the name of the variable display box on the stage. Right-click the selected *name* variable tab and select **Large readout**, as shown in Figure 2.13.

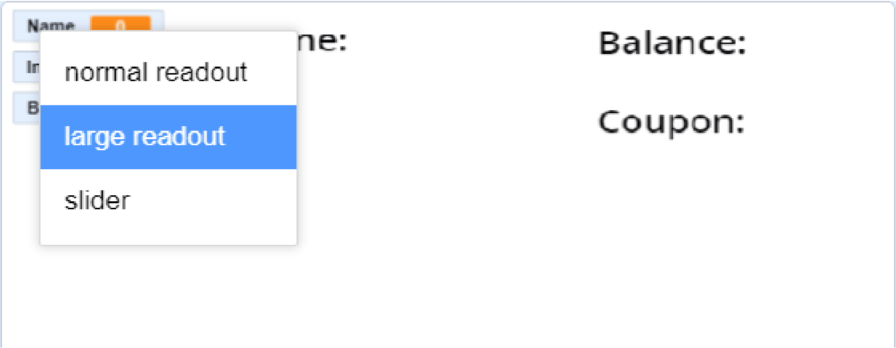


Figure 2.13 Hiding the variable name

Step 14: Move the *name* variable with hidden variable name to the position behind the user name label, as shown in Figure 2.14.

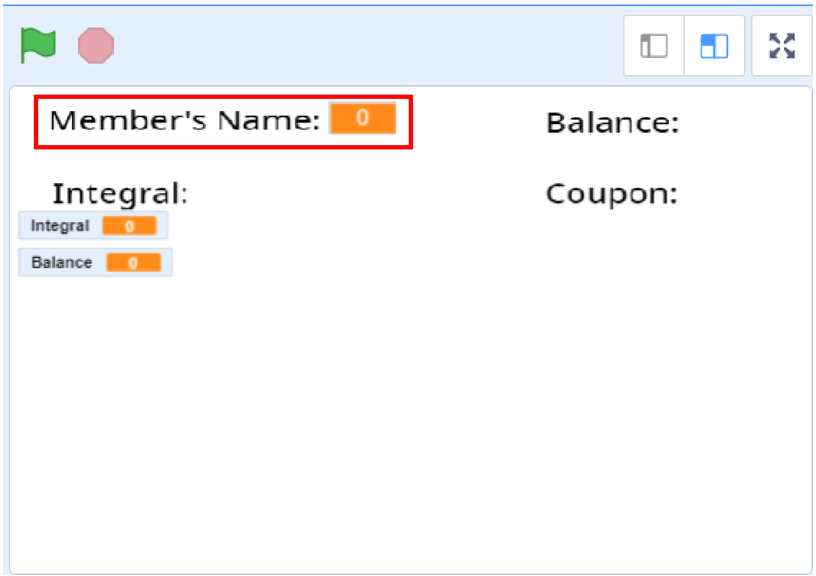
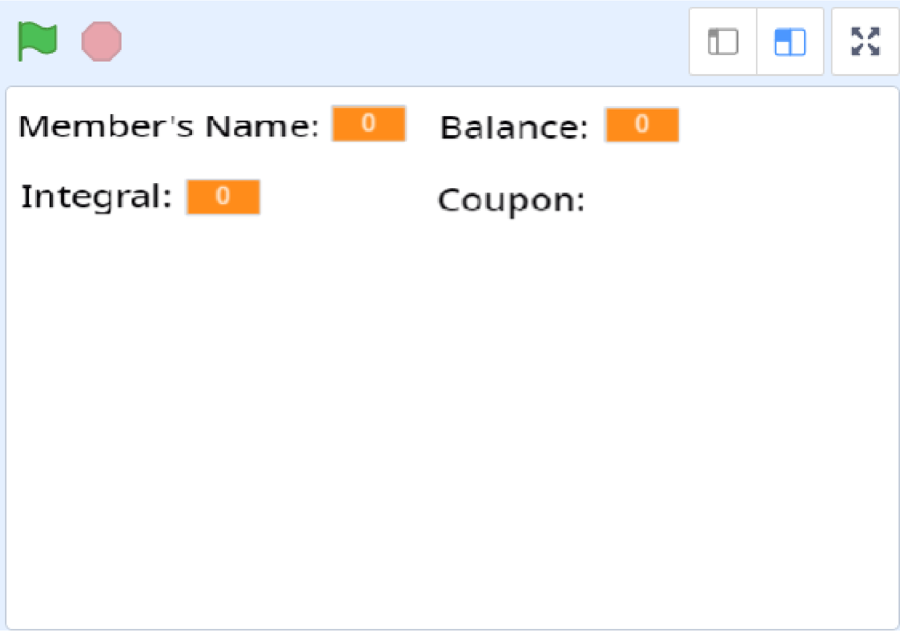


Figure 2.14 Moving the name of variable display box

Step 15: Repeat the above steps to hide names of variable display boxes of *balance* and *integral*, and drag them to corresponding labels, as shown in Figure 2.15.

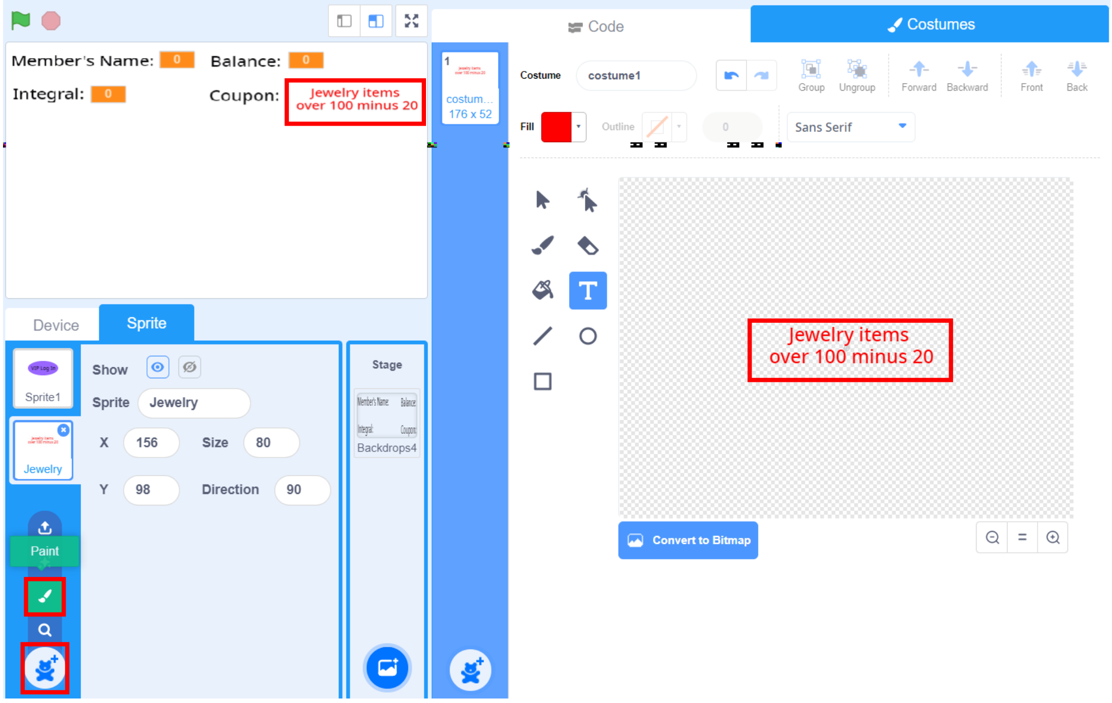


Member's Name: 0 Balance: 0

Integral: 0 Coupon:

Figure 2.15 Moving other variable display boxes

Step 16: Draw the coupon sprite. Set two coupons, separately on jewelry and electric toy, for different VIP customers. Draw the jewelry coupon sprite first. Set the sprite name to **Coupon** and move it to the position behind the coupon label, as shown in Figure 2.16.



Member's Name: 0 Balance: 0

Integral: 0 Coupon: Jewelry items over 100 minus 20

Sprite: Jewelry

X: 156 Size: 80

Y: 98 Direction: 90

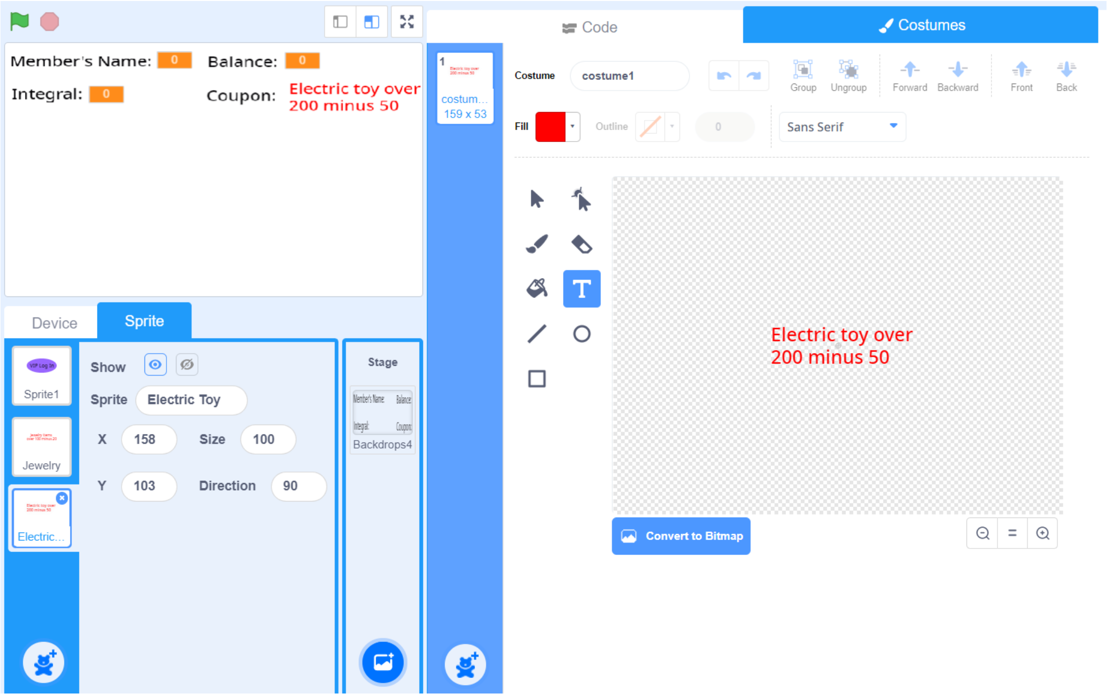
Costume: costume1

Fill Outline 0 Sans Serif

Jewelry items over 100 minus 20

Figure 2.16 Drawing coupon sprite 1

Step17: Hide the jewelry coupon sprite first, and then draw the electric toy coupon sprite and set the sprite name to “Electric Toy”, as shown in Figure 2.17.


Member's Name: 0 Balance: 0

Integral: 0 Coupon: Electric toy over 200 minus 50

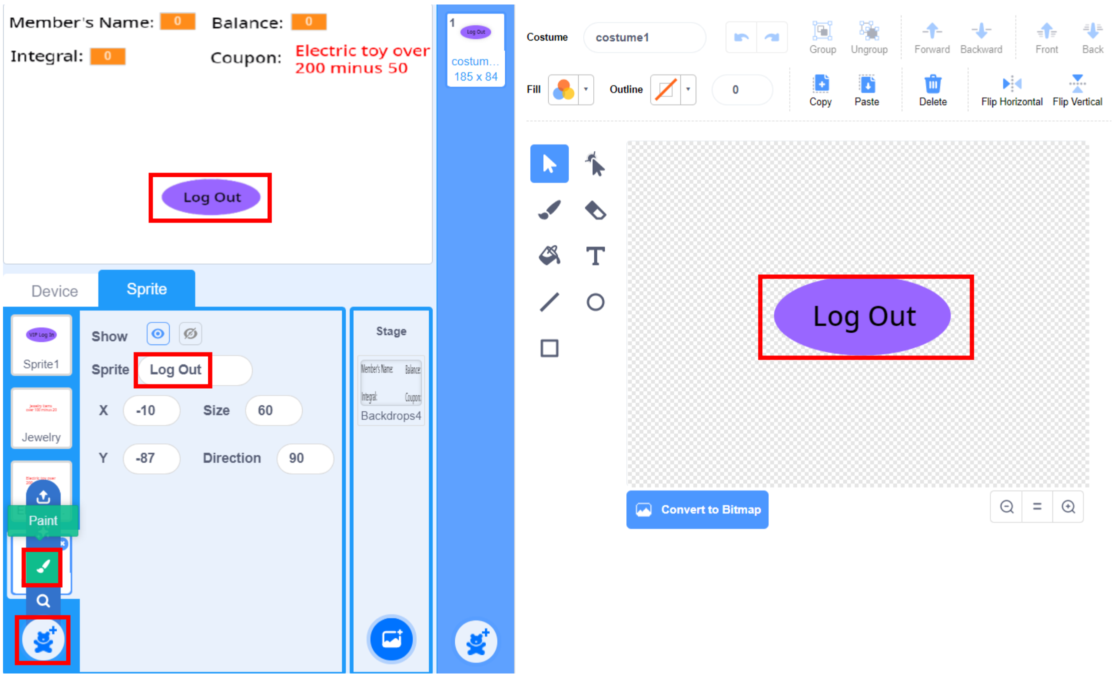
Sprite: Electric Toy X: 158 Size: 100 Y: 103 Direction: 90

Costume: costume1

Electric toy over 200 minus 50

Figure 2.17 Drawing coupon sprite 2

Step 18: Draw the “Log Out” sprite, as shown in Figure 2.18.


Member's Name: 0 Balance: 0

Integral: 0 Coupon: Electric toy over 200 minus 50

Sprite: Log Out X: -10 Size: 60 Y: -87 Direction: 90

Costume: costume1

Log Out

Figure 2.18 Drawing the “Log Out” sprite

Step 19: Save the program as **Membership Identification.sb3**.

3. Conclusion

(1) Steps to draw a new stage backdrops: ______________________
____

(2) Steps to draw a new sprite: ________________________________
____

(3) How to modify the sprite parameters: _______________________
____

4. Self-Assessment

Check the completed content in the experiment task. Tick (√) the completed items, and circle (○) the uncompleted items.

Assessment Content	Completion Status
I've learned how to draw a new Backdrops	
I've learned how to draw a new sprite	
I've created a member login interface	

Task 2: Complete the Face Recognition Program

In the last task, we have designed a member login interface, so how do we log in? Traditionally, we log in with accounts and passwords, which takes more time. To improve login efficiency, we can do so through face recognition. Now let's program for membership identification based on face recognition technology.

1. Observation

Observe how the teacher programs face recognition. Then, fill in the blanks with this process' important steps.

To achieve face recognition technology, step 1 is to collect the number of faces;
step 2 is ;
step 3 is ;
step 4 is .

2. Steps

(1) Prepare Hardware

Step 1: After you have prepared the experimental equipment, power up the robotic arm and connect it to the computer.

Step 2: For the placement of the robotic arm, see task 1. Adjust the position and height of the end of the robotic arm, as well as its camera direction after power-on.

(2) Design Program

Step 1: Draw a flow chart by analyzing the membership identification process, as shown in Figure 2.19.

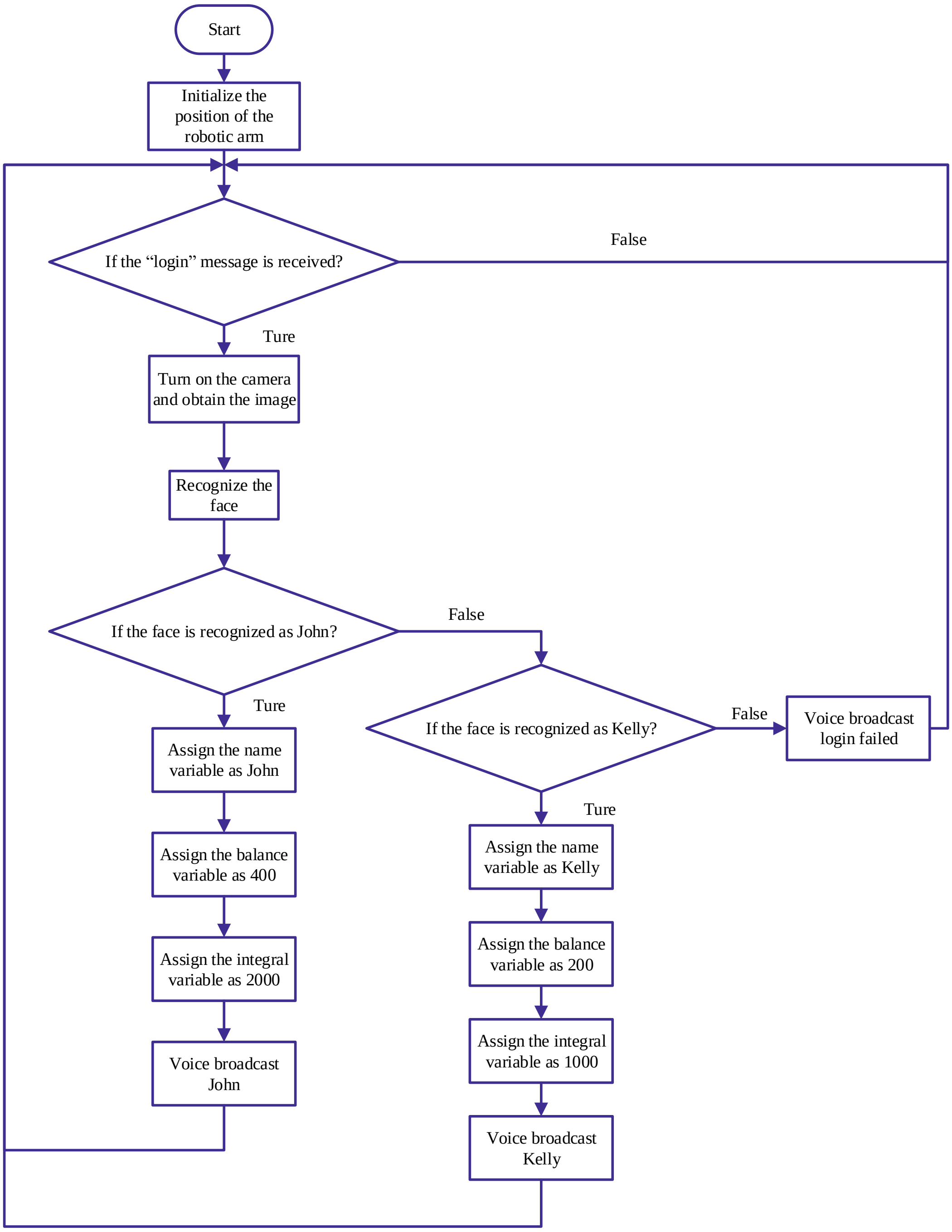
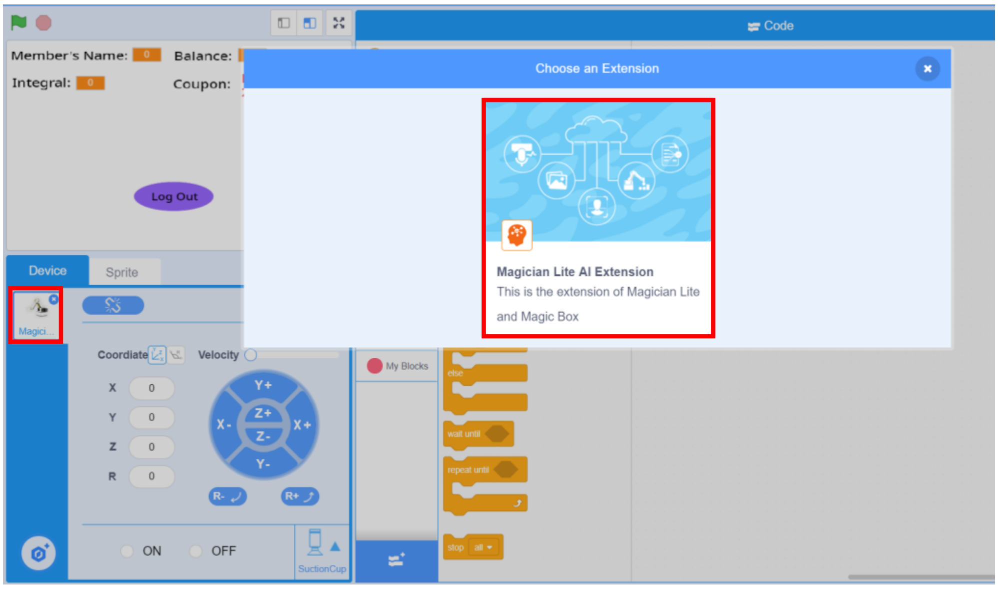

Start
Initialize the position of the robotic arm
If the “login” message is received?
False
Ture
Turn on the camera and obtain the image
Recognize the face
If the face is recognized as John?
False
Ture
Assign the name variable as John
Assign the balance variable as 400
Assign the integral variable as 2000
Voice broadcast John
If the face is recognized as Kelly?
False
Voice broadcast login failed
Ture
Assign the name variable as Kelly
Assign the balance variable as 200
Assign the integral variable as 1000
Voice broadcast Kelly

Figure 2.19 Flow chart of membership identification

Step 2: Enable the DobotScratch software and then the **Membership Identification.sb3** program. Add the Magician Lite device and AI module, as shown in Figure 2.20.



Choose an Extension

Magician Lite AI Extension

This is the extension of Magician Lite and Magic Box

Figure 2.20 Adding the device

Step 3: To identify the membership, scrip the “Log In” sprite before clicking the **Log In** button. Select the “Log In” sprite to start to script, as shown in Figure 2.21.

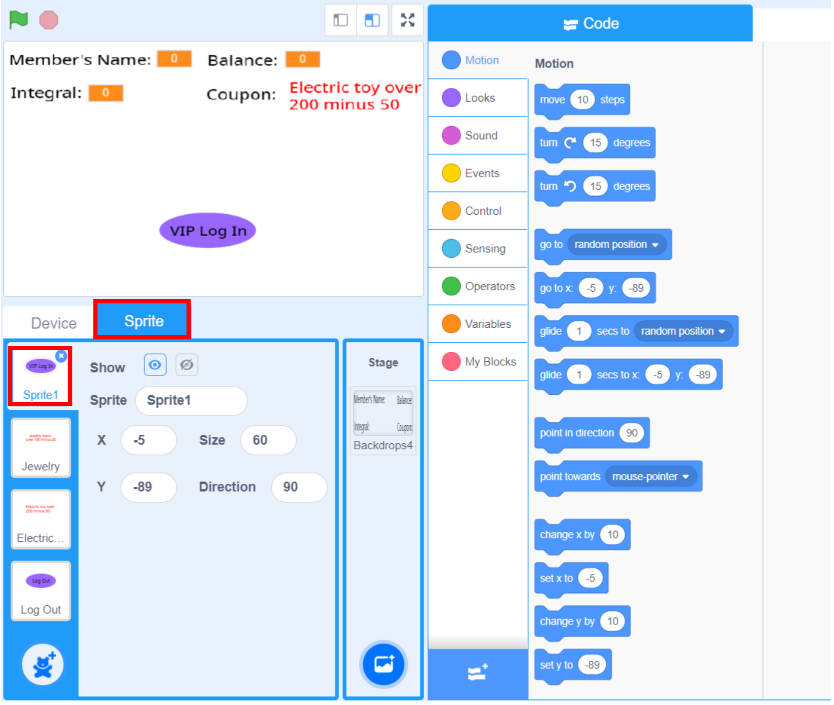


Figure 2.21 Selecting the “Log In” sprite

Step 4: Display the “Log In” sprite when the program starts to run. Start the program by clicking the green flag. Click the **Events** tab and drag the “When ⚑ is clicked” block to the code area. Click the **Looks** tab and drag the “show” block to the code area, as shown in Figure 2.22.

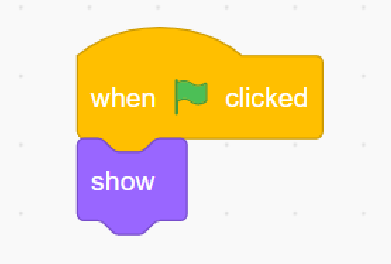


Figure 2.22 Displaying the sprite

Step 5: Broadcast the “Log In” message when the “Log In” sprite is clicked. Click the **Events** tab and drag the “When a sprite is clicked” block to the code area, and then drag the “Broadcast message” block to the code area, as shown in Figure 2.23.

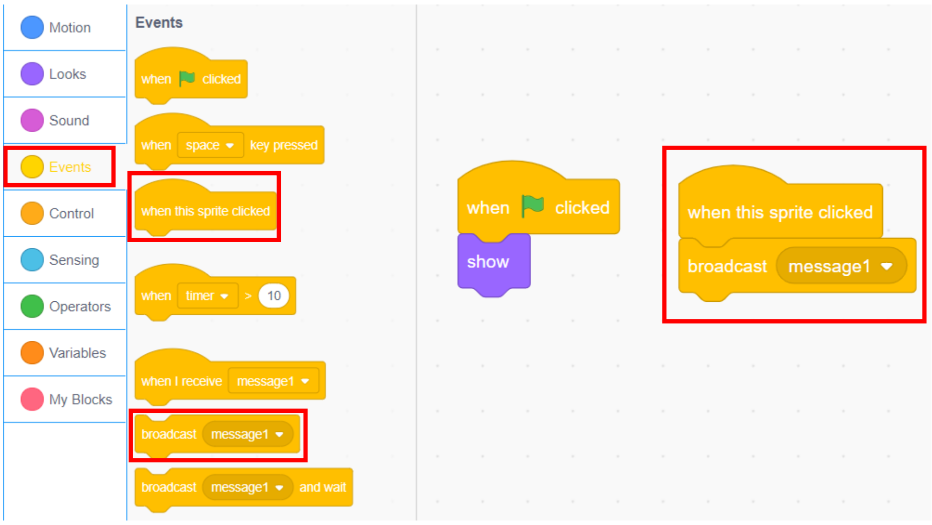
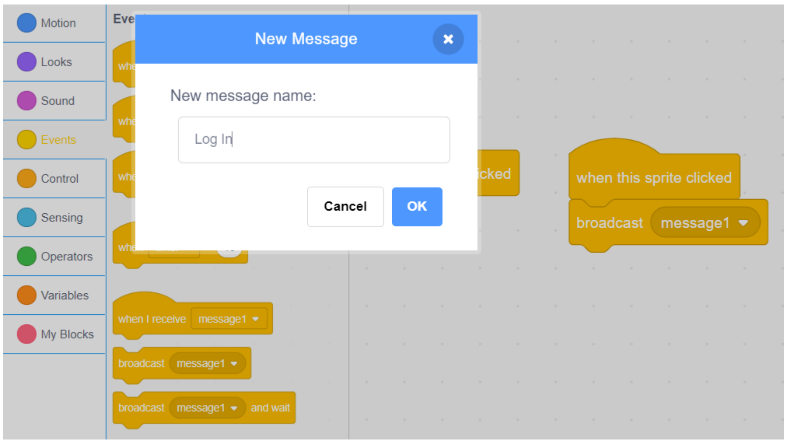


Figure 2.23 Broadcasting the message

Step 6: Modify the message to be broadcast to “Log In”. Click the parameter box of the “Broadcast message” block, select **New Message**, and set the new message name to **Log In**, as shown in Figure 2.24.

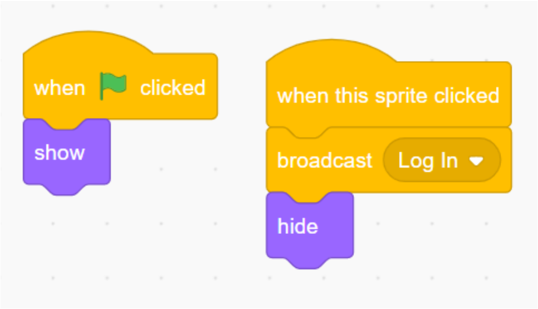

New Message

New message name:

Log In

Figure 2.24 Creating the new message

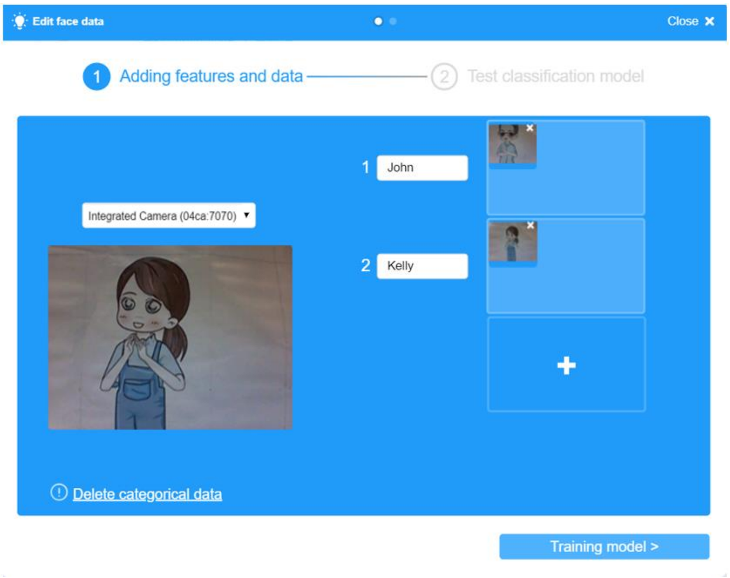
Step 7: Hide “Log In”. After clicking the “Log In” sprite, you will see the login interface appear. Therefore, make sure to hide the sprite. Click the **Looks** tab and drag the “hide” block to the code area, as shown in Figure 2.25.



```
when 🏁 clicked
show

when this sprite clicked
broadcast Log In ▾
hide
```

Figure 2.25 Hiding the sprite

Step 8: After clicking the “Log In” sprite, start membership identification through face recognition. Click the device drop-down list and select **Magician Lite**. To enter a member's face message, click the **AI** tab and create new face data. Enter the face data of John and Kelly, and then train the model, as shown in Figure 2.26.



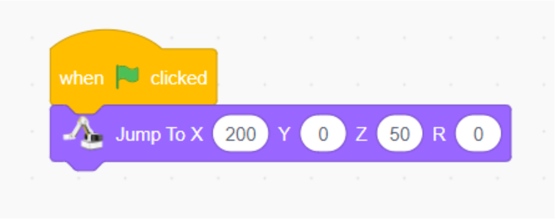
Edit face data
Close ✕
1 Adding features and data
2 Test classification model
Integrated Camera (04ca:7070) ▾
+
Delete categorical data
Training model >

1 John

2 Kelly

Figure 2.26 Creating the face data

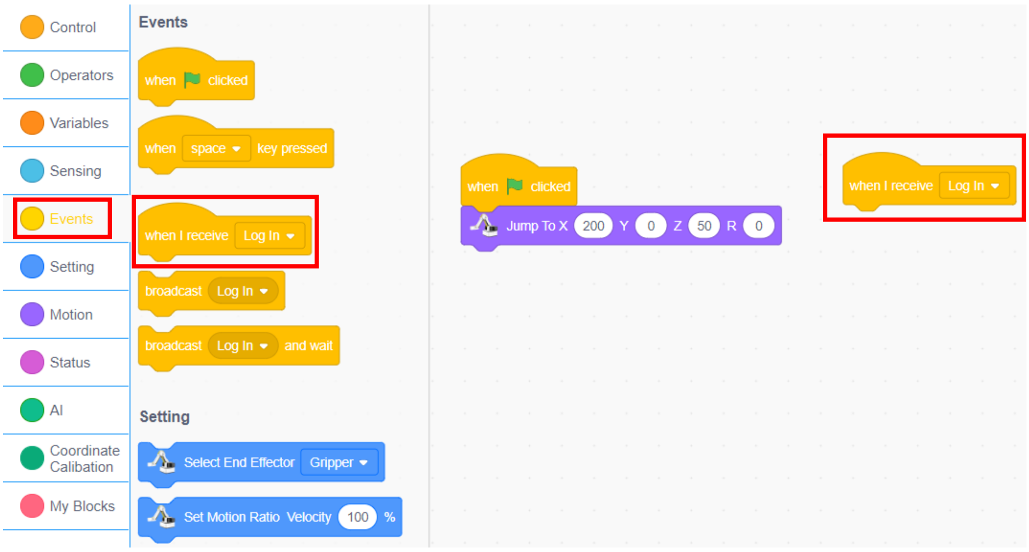
Step 9: Initialize the camera position, as shown in Figure 2.27.



```
when 🏁 clicked
Jump To X 200 Y 0 Z 50 R 0
```

Figure 2.27 Initializing the camera position

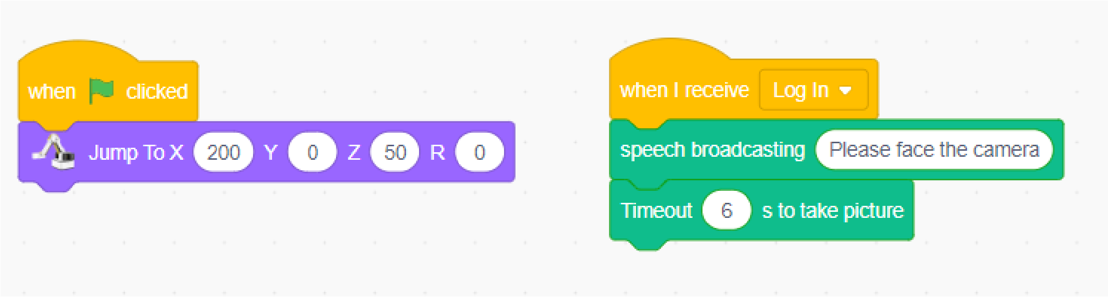
Step 10: Start face recognition starts upon receipt of the “Log In” message by Magician Lite. Click the **Events** tab and drag the “When I receive Log In**”** block to the code area, as shown in Figure 2.28.



```
when clicked
Jump To X 200 Y 0 Z 50 R 0

when I receive Log In
```

Figure 2.28 Receiving the message instruction

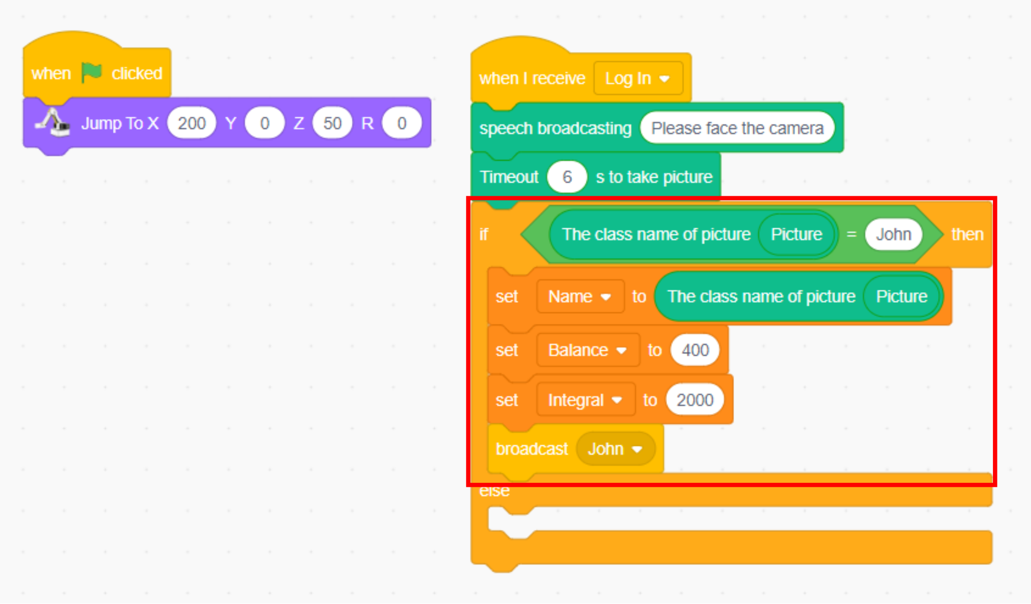
Step 11: Remind you to enable face recognition and turn on the camera, as shown in Figure 2.29.



```
when clicked
Jump To X 200 Y 0 Z 50 R 0

when I receive Log In
speech broadcasting Please face the camera
Timeout 6 s to take picture
```

Figure 2.29 Receiving the message Instruction

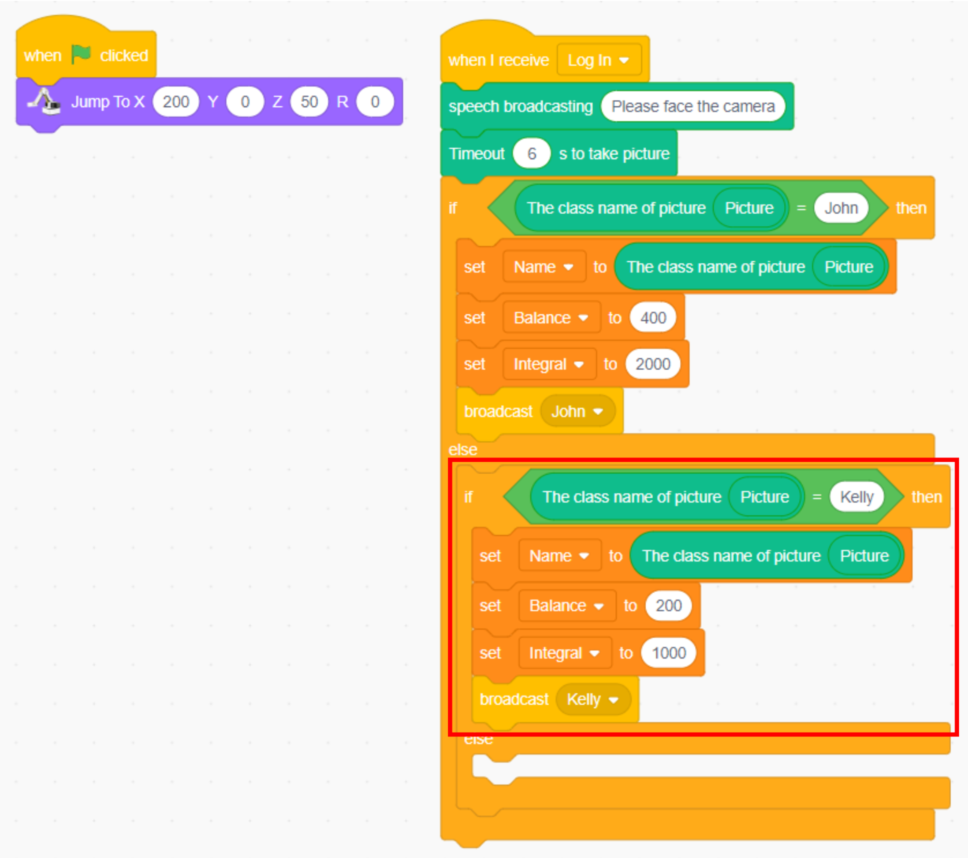
Step 12: Identify and judge whether the person recognized is John. If yes, set his basic information. The system issues a voice message “John", as shown in Figure 2.30.



```
when 🏴 clicked
Jump To X 200 Y 0 Z 50 R 0

when I receive Log In ▾
speech broadcasting Please face the camera
Timeout 6 s to take picture
if The class name of picture Picture = John then
    set Name ▾ to The class name of picture Picture
    set Balance ▾ to 400
    set Integral ▾ to 2000
    broadcast John ▾
else
```

Figure 2.30 Setting the information of John

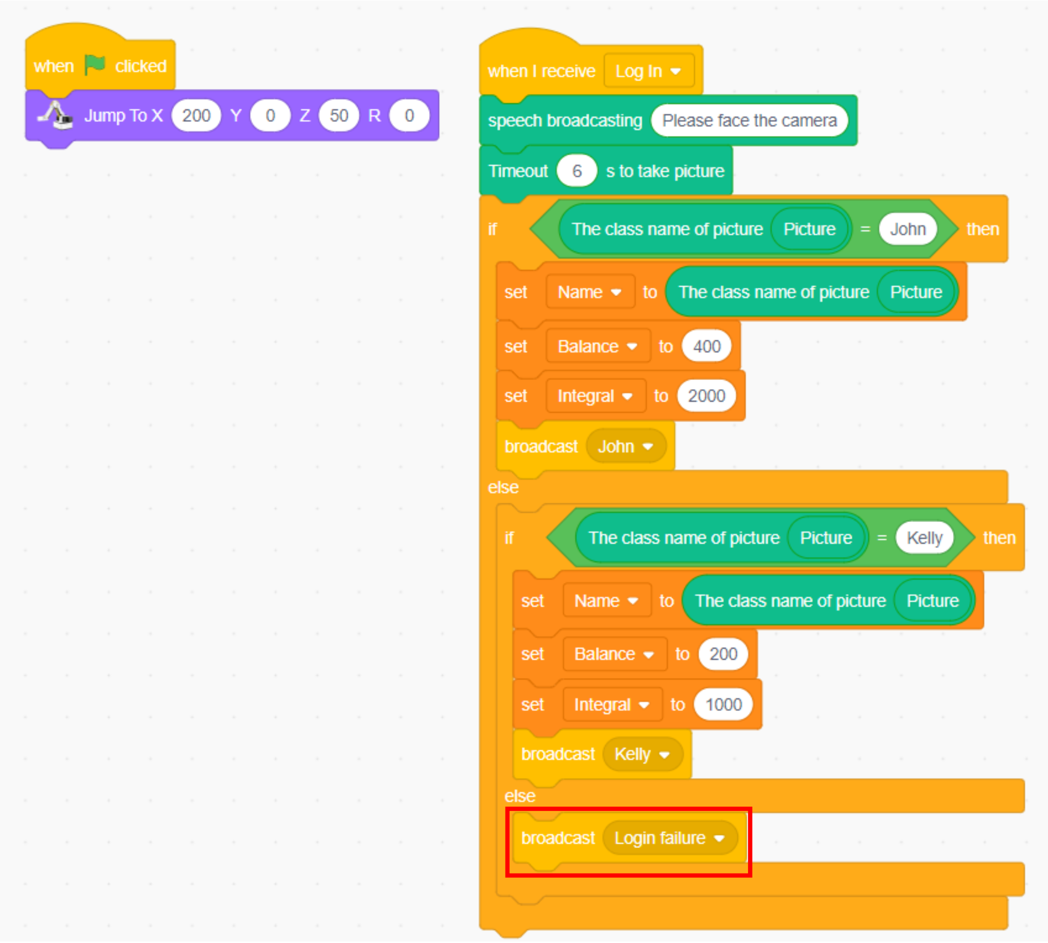
Step 13: If no, judge whether the person recognized is Kelly. If yes, set her basic information, as shown in Figure 2.31.



```
when 🏴 clicked
Jump To X 200 Y 0 Z 50 R 0

when I receive Log In ▾
speech broadcasting Please face the camera
Timeout 6 s to take picture
if The class name of picture Picture = John then
    set Name ▾ to The class name of picture Picture
    set Balance ▾ to 400
    set Integral ▾ to 2000
    broadcast John ▾
else
    if The class name of picture Picture = Kelly then
        set Name ▾ to The class name of picture Picture
        set Balance ▾ to 200
        set Integral ▾ to 1000
        broadcast Kelly ▾
    else
```

Figure 2.31 Setting the information of Kelly

Step 14: If the person is neither Kelly nor John, the system issues a voice message “Login failure”, as shown in Figure 2.32.



```
when ⚑ clicked
Jump To X (200) Y (0) Z (50) R (0)
```

```
when I receive [Log In ▾]
speech broadcasting (Please face the camera)
Timeout (6) s to take picture
if <(The class name of picture (Picture)) = (John)> then
    set [Name ▾] to (The class name of picture (Picture))
    set [Balance ▾] to (400)
    set [Integral ▾] to (2000)
    broadcast [John ▾]
else
    if <(The class name of picture (Picture)) = (Kelly)> then
        set [Name ▾] to (The class name of picture (Picture))
        set [Balance ▾] to (200)
        set [Integral ▾] to (1000)
        broadcast [Kelly ▾]
    else
        broadcast [Login failure ▾]
```

Figure 2.32 Login failure

Step 15: Save the program file.

3. Conclusion

(1) Steps of face recognition: collect face data,______, _______, __

(2) Instructions of face recognition: ______________________________

____

4. Self-Assessment

Check the completed content in the experiment task. Tick (√) the completed items, and circle (◯) the uncompleted items.

Assessment Content	Completion Status
I've known the steps of face recognition	
I've collected face data	
I've trained the face recognition model	
I've tested the face recognition model	
I've completed the face recognition-based login programming	

Task 3: Program the Member Login System

We have designed the member login system interface and simply programmed the system in the previous tasks. Then how can we make the system display the member information after successful login? Can non-members log in to the interface? Now let's program the member login system completely.

1. Observation

Observe how the teacher demonstrates the member login process. Then, fill in the following blanks.

When the face is recognized as a member, the system goes to the member information interface;

If non-member, the system should

2. Steps

Step 1: Enable the DobotScratch software and click **sprite**. First script the stage backdrops, and click **Stage** to code, as shown in Figure 2.33.

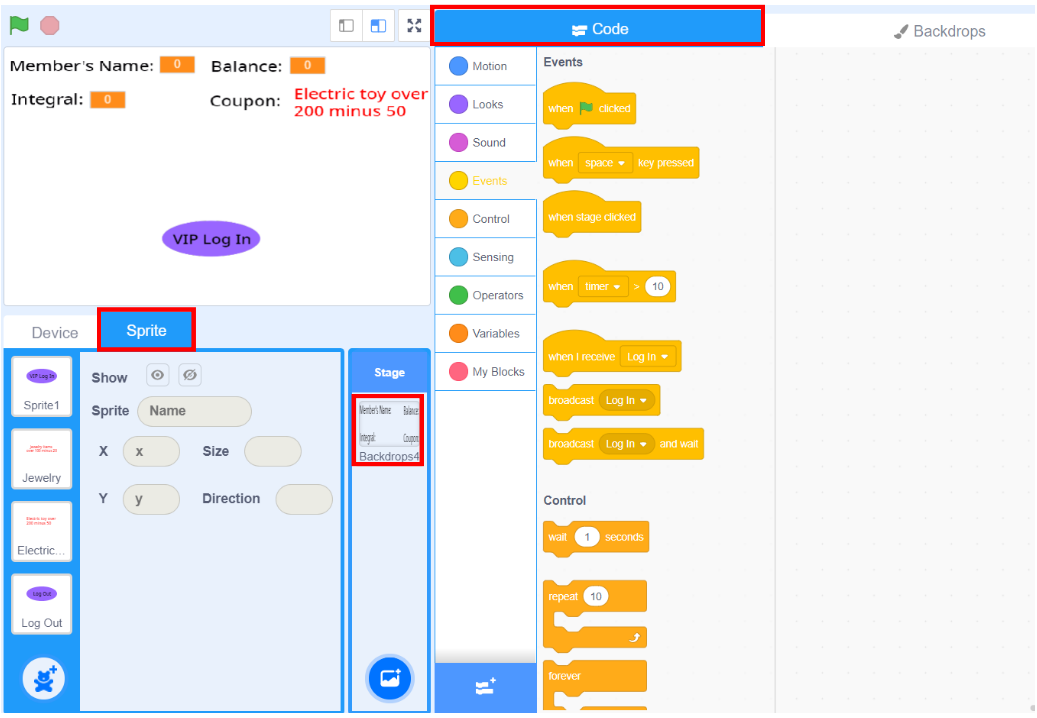


Figure 2.33 Selecting the stage backdrops

Step 2: When the program starts to run, the interface displays “Welcome to Smart Mall”, which is Backdrops 1. Click the **Looks** tab and drag the “Change to Backdrops 1” block to the code area, as shown in Figure 2.34.

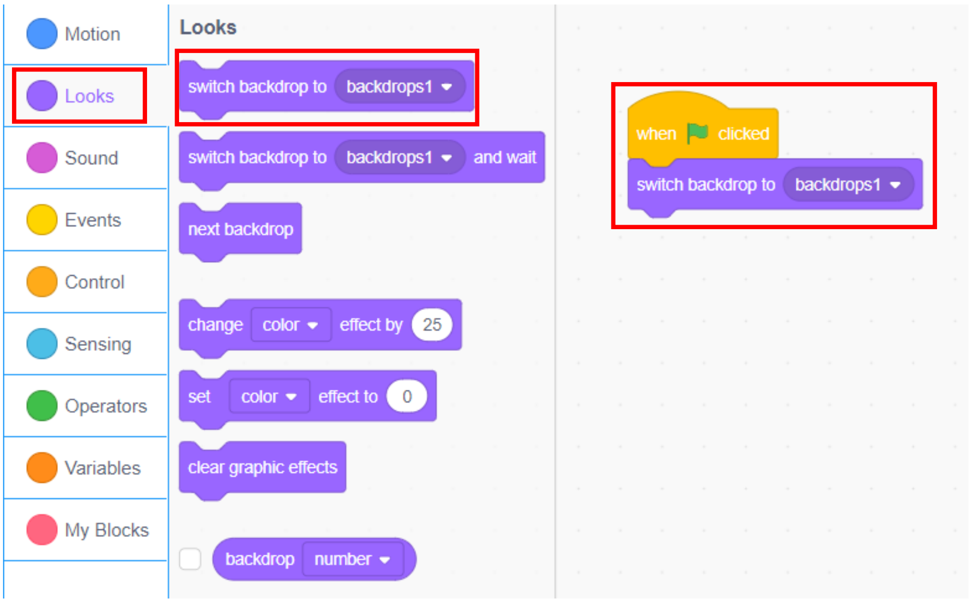
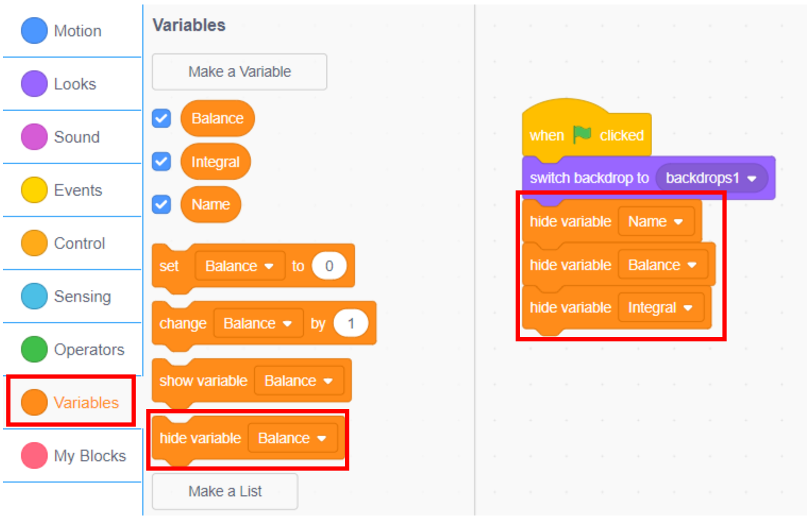


Figure 2.34 Selecting Backdrops 1

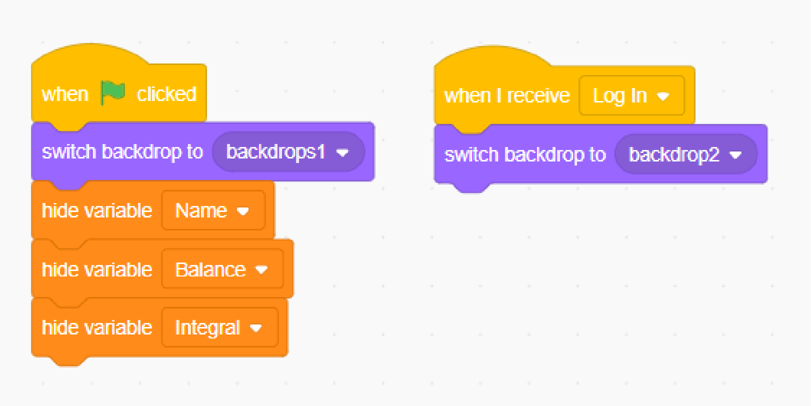
Step 3: Under Backdrops 1, hide the variable display box, as shown in Figure 2.35.



```
when 🏴 clicked
switch backdrop to backdrops1
hide variable Name
hide variable Balance
hide variable Integral
```

Figure 2.35 Hiding the variable display box

Step 4: When “Log In” message is received, the Backdrops switches to "Logging in...", i.e., Backdrops 2, as shown in Figure 2.36.

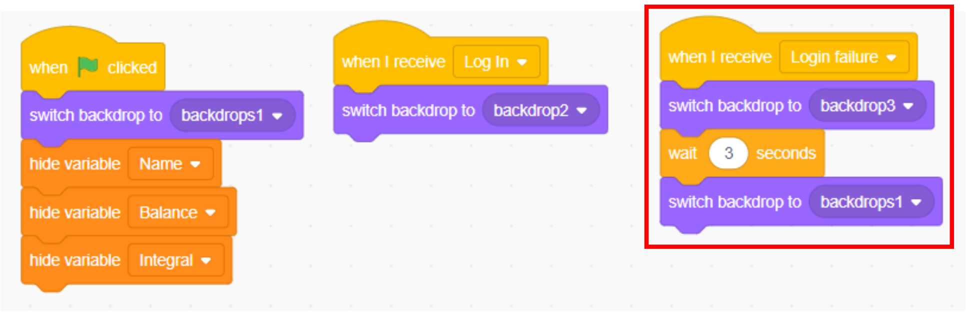


```
when 🏴 clicked
switch backdrop to backdrops1
hide variable Name
hide variable Balance
hide variable Integral
```

```
when I receive Log In
switch backdrop to backdrop2
```

Figure 2.36 Switching to Backdrops 2

Step 5: Upon receipt of “Login failure” message, switch the Backdrops to "Login failure. Register first!!” i.e., Backdrops 3. Three seconds later, the interface switches back to Backdrops 1, as shown in Figure 2.37.



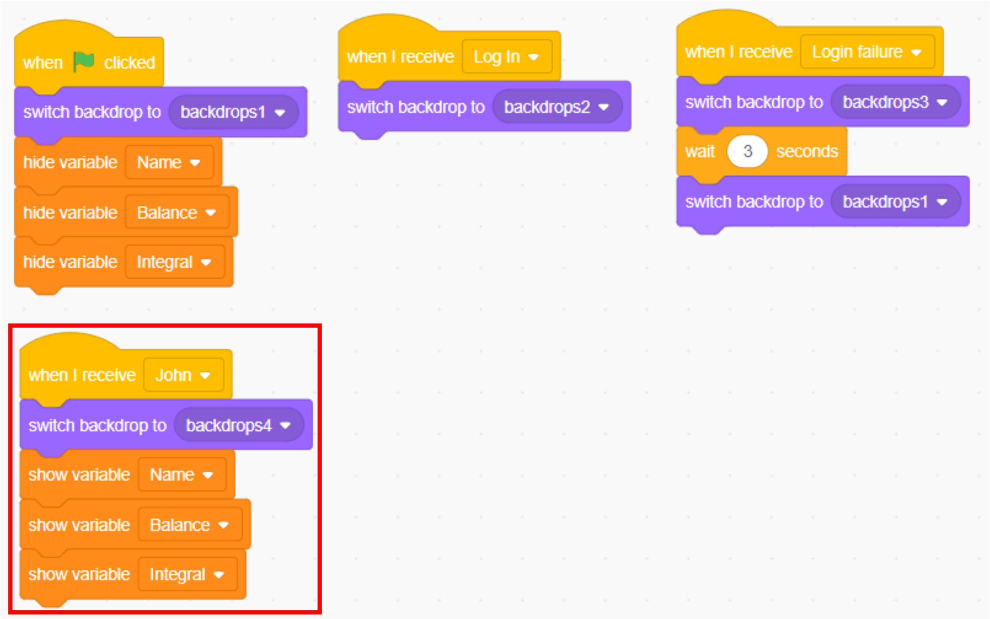
```
when 🏴 clicked
switch backdrop to backdrops1
hide variable Name
hide variable Balance
hide variable Integral
```

```
when I receive Log In
switch backdrop to backdrop2
```

```
when I receive Login failure
switch backdrop to backdrop3
wait 3 seconds
switch backdrop to backdrops1
```

Figure 2.37 Setting the “Login Failure” Backdrops

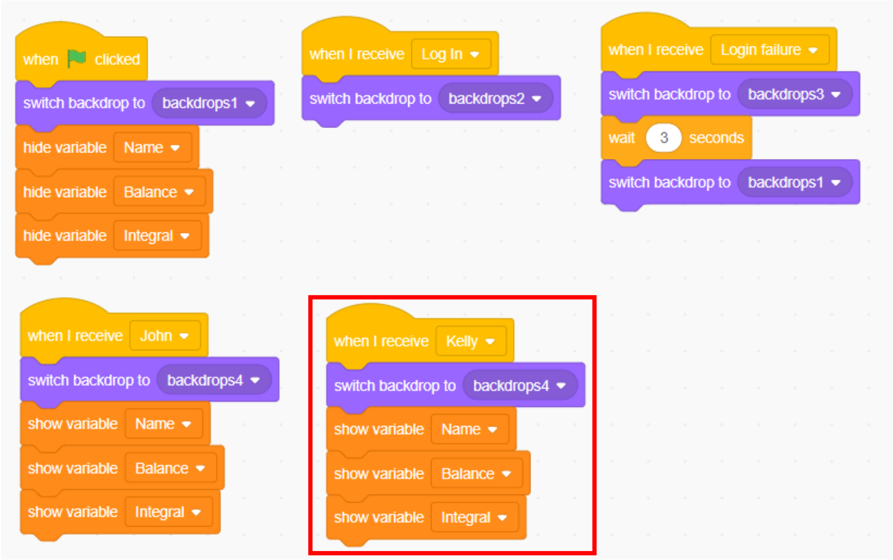
Step 6: When “John” message is received, the Backdrops switches to the member information interface, i.e., Backdrops 4, where the display boxes of each variable are displayed, as shown in Figure 2.38.



```
when ⚑ clicked
switch backdrop to backdrops1 ▾
hide variable Name ▾
hide variable Balance ▾
hide variable Integral ▾

when I receive Log In ▾
switch backdrop to backdrops2 ▾

when I receive Login failure ▾
switch backdrop to backdrops3 ▾
wait 3 seconds
switch backdrop to backdrops1 ▾

when I receive John ▾
switch backdrop to backdrops4 ▾
show variable Name ▾
show variable Balance ▾
show variable Integral ▾
```

Figure 2.38 Receiving the message of John

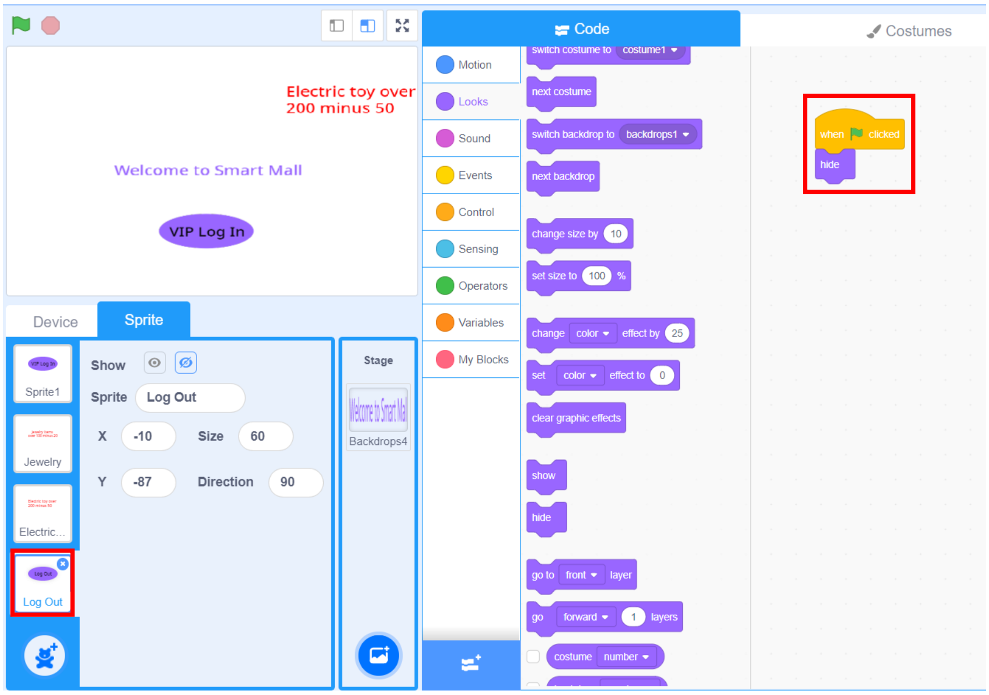
Step 7: When the message of Kelly is received, repeat the above step, as shown in Figure 2.39.



```
when ⚑ clicked
switch backdrop to backdrops1 ▾
hide variable Name ▾
hide variable Balance ▾
hide variable Integral ▾

when I receive Log In ▾
switch backdrop to backdrops2 ▾

when I receive Login failure ▾
switch backdrop to backdrops3 ▾
wait 3 seconds
switch backdrop to backdrops1 ▾

when I receive John ▾
switch backdrop to backdrops4 ▾
show variable Name ▾
show variable Balance ▾
show variable Integral ▾

when I receive Kelly ▾
switch backdrop to backdrops4 ▾
show variable Name ▾
show variable Balance ▾
show variable Integral ▾
```

Figure 2.39 Receiving the message of Kelly

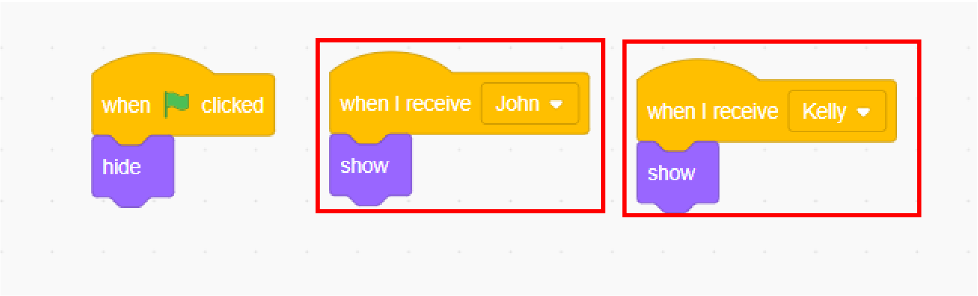
Step 8: Script the “Log Out” sprite. Click the "Log Out" sprite from the sprite list to start scripting. When the program starts to run, the sprite is hidden, as shown in Figure 2.40.



```
when 🏴 clicked
hide
```

Figure 2.40 Hiding the “Log Out” sprite

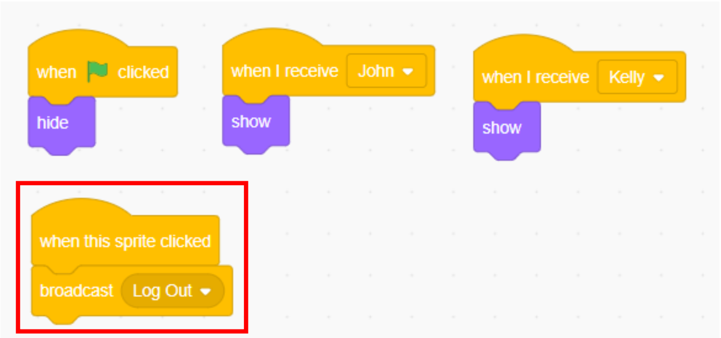
Step 9: When the "Log Out" sprite receives the message of John or Kelly message, the sprite appears, as shown in Figure 2.41.



```
when 🏴 clicked
hide

when I receive John
show

when I receive Kelly
show
```

Figure 2.41 Displaying the “Log Out” sprite

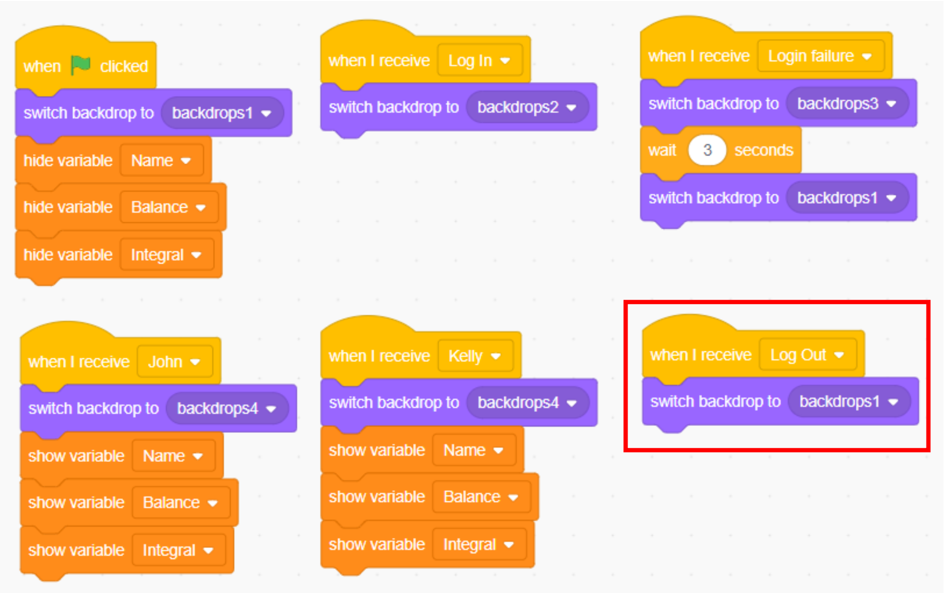
Step 10: Broadcast the “Log Out” message when you click the "Log Out" sprite, as shown in Figure 2.42.



```
when 🏴 clicked
hide

when I receive John
show

when I receive Kelly
show

when this sprite clicked
broadcast Log Out
```

Figure 2.42 Clicking the “Log Out” sprite

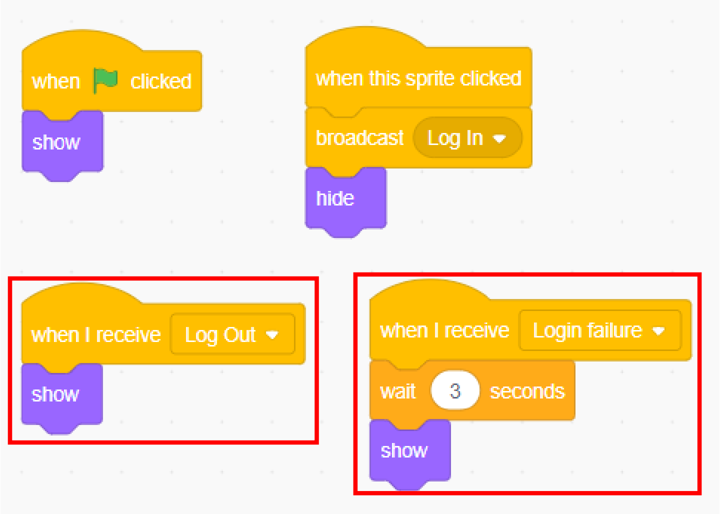
Step 11: When the stage Backdrops receives the “Log Out” message, the Backdrops switches to Backdrops 1, as shown in Figure 2.43.



```
when 🏴 clicked
switch backdrop to backdrops1
hide variable Name
hide variable Balance
hide variable Integral

when I receive Log In
switch backdrop to backdrops2

when I receive Login failure
switch backdrop to backdrops3
wait 3 seconds
switch backdrop to backdrops1

when I receive John
switch backdrop to backdrops4
show variable Name
show variable Balance
show variable Integral

when I receive Kelly
switch backdrop to backdrops4
show variable Name
show variable Balance
show variable Integral

when I receive Log Out
switch backdrop to backdrops1
```

Figure 2.43 Switching backdrops upon receipt of “Log Out” Message

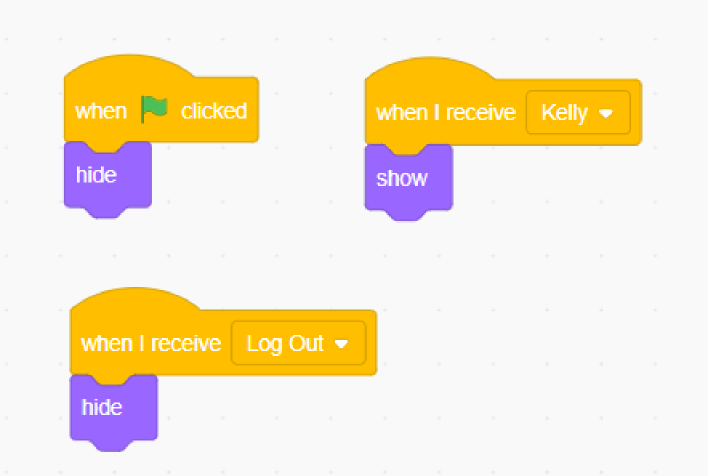
Step 12: When the "Log In" sprite receives the "Log Out" or "Login failure” message, it appears, as shown in Figure 2.44.



```
when 🏴 clicked
show

when this sprite clicked
broadcast Log In
hide

when I receive Log Out
show

when I receive Login failure
wait 3 seconds
show
```

Figure 2.44 Script for displaying the “Log In” sprite

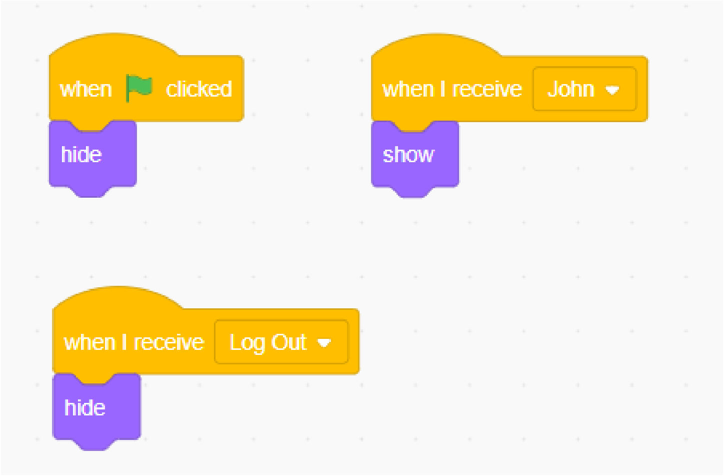
Step 13: Script the “Jewelry” coupon sprite. When the program starts to run, the “Jewelry” sprite is hidden. It appears only upon receiving the “Kelly” message, and is hidden again upon receiving the “Log Out” message, as shown in Figure 2.45.



```
when [flag] clicked
hide

when I receive [Kelly ▾]
show

when I receive [Log Out ▾]
hide
```

Figure 2.45 “Jewelry” sprite script

Step 14: Similarly, you can script the "Electric Toy" sprite. The coupon only appears in the member information of John, so the sprite appears only upon receiving the message of John, as shown in Figure 2.46.



```
when [flag] clicked
hide

when I receive [John ▾]
show

when I receive [Log Out ▾]
hide
```

Figure 2.46 “Electric Toy” sprite script

3. Conclusion

(1) Instructions to hide and display a sprite: ____________________

____

(2) Instructions to communicate between sprites: ________________

____

(3) Instructions to switch stage backdrops: ____________________

____

4. Self-Assessment

Check the completed content in the experiment task. Tick (√) the completed items, and circle (◯) the uncompleted items.

Assessment Content	**Completion Status**
I've mastered the instructions to hide and display a sprite	
I've mastered the instructions to communicate between sprites	
I've mastered the instructions to switch stage Backdrops	
I've completed the member login system programming	

Experiment 3

Intelligent Navigation

Team Name: Team Member: Date:

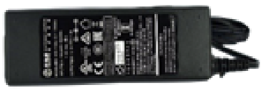
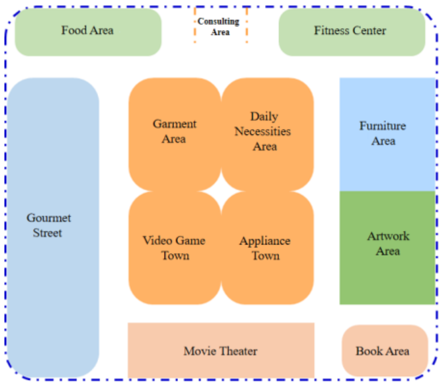
Overview

How to find the right route in a huge Smart Mall? This experiment simulates a speech wakeup robot. When you ask the robot the route to your destination, it draws the route from your current location to the destination. Now, let's complete the task of awaking the robot for intelligent navigation.

Objective

- Understand speech recognition instructions by awaking the robot by voice.
- Understand the sprite of variables and use them skillfully by using the variable related blocks in Scratch.
- Understand the use of branch structure by using the “If-then” block in Scratch.
- Understand the use of loop structure by using the “forever” block in Scratch.
- Use the branch structure and loop structure to program the speech wakeup robot for intelligent navigation.

Equipment

Equipment Picture	Name	Quantity
	Dobot Magician Lite Robotic arm	1
	Pen	1
	Power adapter	1
	USB cable	1
 Food Area Consulting Area Fitness Center Gourmet Street Garment Area Daily Necessities Area Furniture Area Video Game Town Appliance Town Artwork Area Movie Theater Book Area	Mall map	1

Requirements

- ➢ Take care when using electricity.
- ➢ Before the experiment, check whether the experimental device is complete and intact. If there is any omission or damage, please report to the teacher.
- ➢ Any specific operations in the experiment shall be performed according to the experiment manual. If you have any questions, please promptly ask the teacher.

- ➢ During the experiment, the joints will start to work as the robotic arms are powered on. In that case, do not move the joints of the robotic arms hard if you do not press the unlock key.
- ➢ Arrange all devices after the experiment. You shall not leave the lab before check by the group leader.

Task 1: Awaken the Robot

We find in daily life that the hallway light automatically lights up when it "hears" sound. With cellphone, we can open an APP with Voice Assistant to consult the weather, and the like. We can control sound equipment, TV and other devices by voice. Then, can we control the robotic arm by speech? Now let's try to awaken the robot through voice recognition.

1. Analysis

We analyze the main steps to awaken the robot.

(1) In the process of awakening the robot by voice, the first step is _____.

(2) The robot judges whether the speech contains: _____ related content.

(3) If navigation related information is recognized, the system broadcasts by voice that __________.

(4) After voice broadcast, the robotic arm ____________.

2. Steps

(1) Prepare Hardware

Step 1: Power on the Magician Lite.

Step 2: Connect Magician Lite to the computer using a USB cable and turn on the power supply.

(2) Design Program

Step 1: Read the flow chart based on the analysis of the robot wakeup steps, as shown in Figure 3.1.

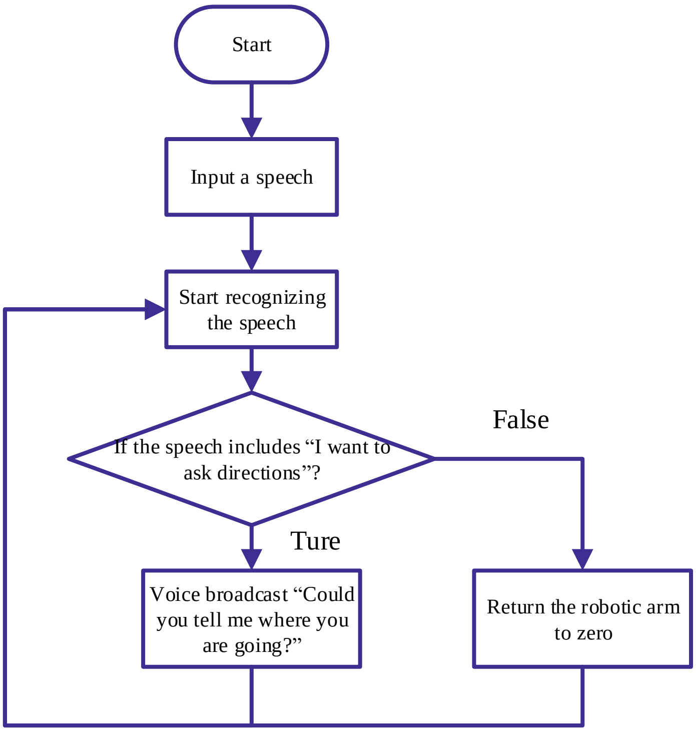

Start
Input a speech
Start recognizing the speech
If the speech includes "I want to ask directions"?
False
Ture
Voice broadcast "Could you tell me where you are going?"
Return the robotic arm to zero

Figure 3.1 Flow chart of robot wakeup

Step 2: Enable the DobotScratch software and select Magician Lite device and connect to it. Add the AI extension module, as shown in Figure 3.2.

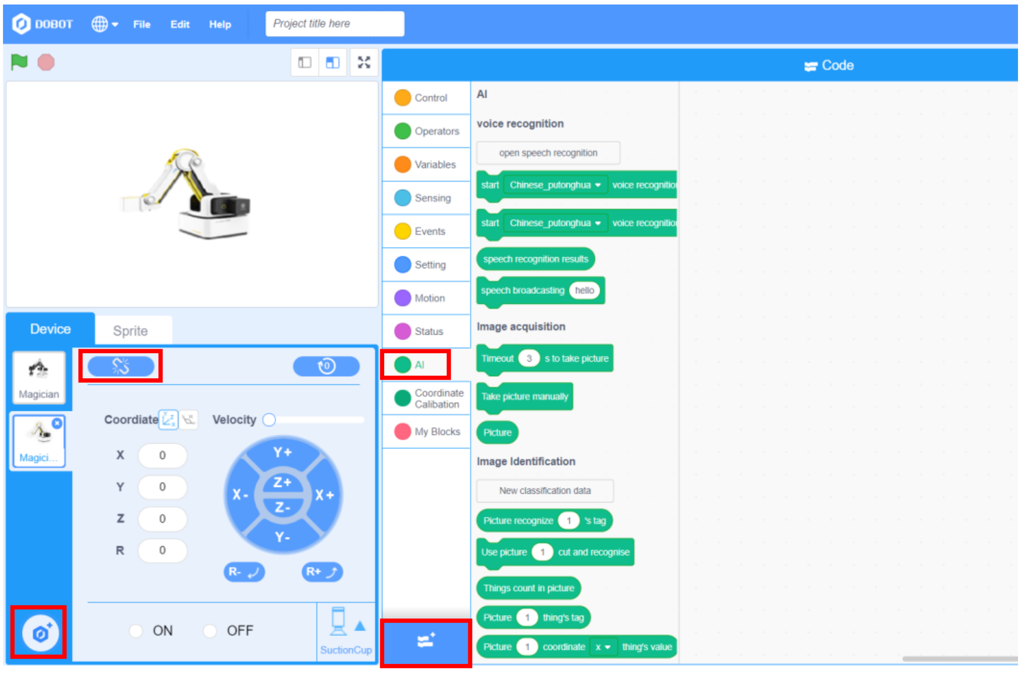


Figure 3.2 Configuring the software

Step 3: Click the **Events** tab, and drag the "When the green flag is clicked" block, as shown in Figure 3.3.

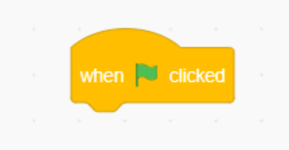


Figure 3.3 “When the green flag is clicked” Block

Step 4: Click the **AI** tab, drag the “Start Chinese_Mandarin (Simplified) voice recognition for 1 second” block to splice with the previous block, and change the duration to 5 seconds, as shown in Figure 3.4.

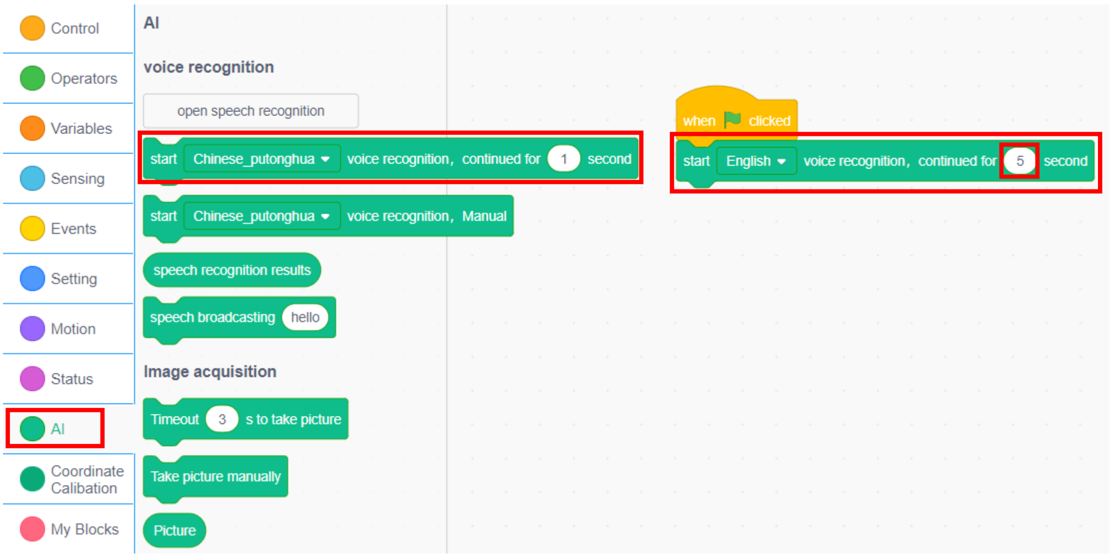

Control
Operators
Variables
Sensing
Events
Setting
Motion
Status
AI
Coordinate Calibation
My Blocks
AI
voice recognition
open speech recognition
start Chinese_putonghua voice recognition, continued for 1 second
start Chinese_putonghua voice recognition, Manual
speech recognition results
speech broadcasting hello
Image acquisition
Timeout 3 s to take picture
Take picture manually
Picture
when clicked
start English voice recognition, continued for 5 second

Figure 3.4 Starting speech recognition

Step 5: Create a variable. Click **Make a Variable** from the **Variables** tab. In the pop-up dialog box, name the new variable *keywords to awaken robot*, and select **For all sprites**. Click **OK** and thus create the variable, as shown in Figure 3.5.

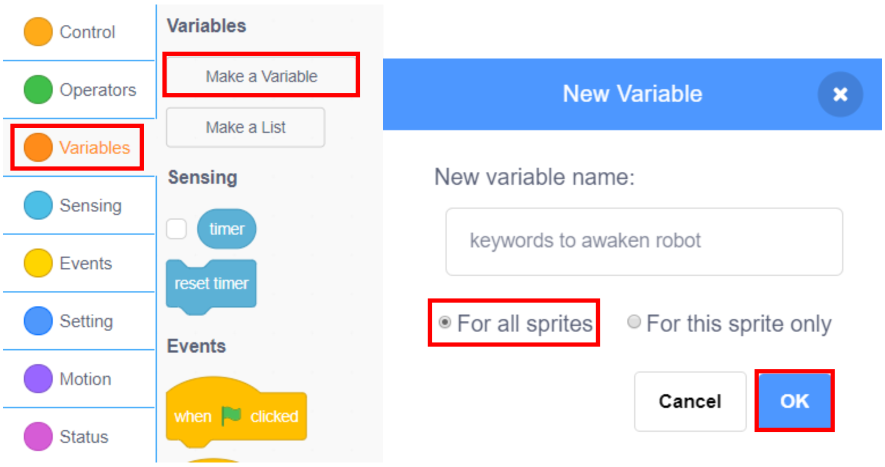
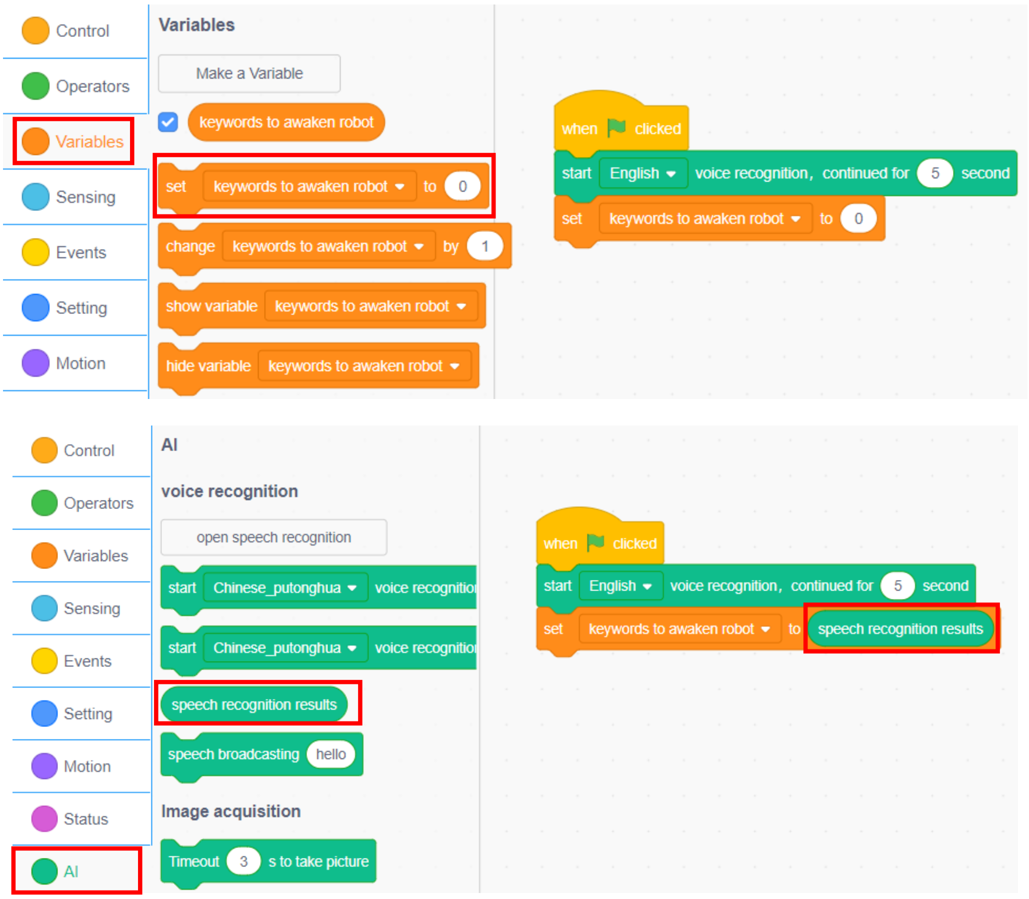


Figure 3.5 Creating the variable

Step 6: Click the **Variables** tab and drag the “Set the ‘keywords to awaken robot’ to (0)” block to splice with the previous block. Click the **AI** tab, drag the “speech recognition results” block and set the *keywords to awaken robot* value to the speech recognition results, as shown in Figure 3.6.

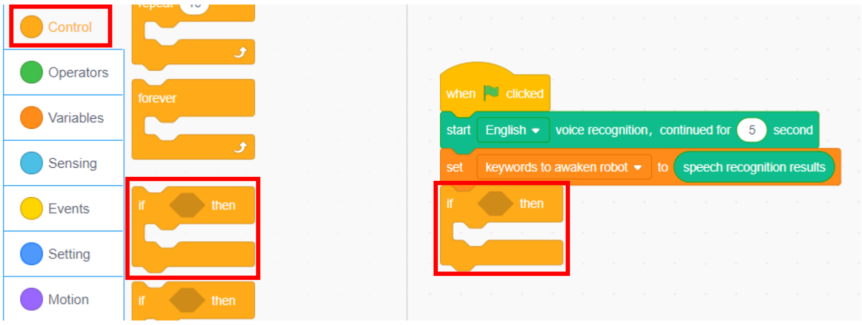

Control
Operators
Variables
Sensing
Events
Setting
Motion
Variables
Make a Variable
keywords to awaken robot
change keywords to awaken robot by 1
show variable keywords to awaken robot
hide variable keywords to awaken robot
Control
Operators
Variables
Sensing
Events
Setting
Motion
Status
AI
AI
voice recognition
open speech recognition
start Chinese_putonghua voice recognitio
start Chinese_putonghua voice recognitio
speech broadcasting hello
Image acquisition
Timeout 3 s to take picture

```
set keywords to awaken robot to 0

when 🏁 clicked
start English voice recognition, continued for 5 second
set keywords to awaken robot to 0

speech recognition results

when 🏁 clicked
start English voice recognition, continued for 5 second
set keywords to awaken robot to speech recognition results
```

Figure 3.6 Setting the k*eywords to awaken robot* variable to speech recognition results

Step 7: Judge if the speech about awakening the robot contains “I would like to ask the route”.

1) Click the **Control** tab and drag the “If... then...” block to splice with the previous block, as shown in Figure 3.7.



```
when 🏁 clicked
start English voice recognition, continued for 5 second
set keywords to awaken robot to speech recognition results
if then
```

Figure 3.7 “If... then...” block

2) Click the **Operators** tab, and drag the "Does string () contain ()" block into the hexagonal box in the "If... then..." block, as shown in Figure 3.8.

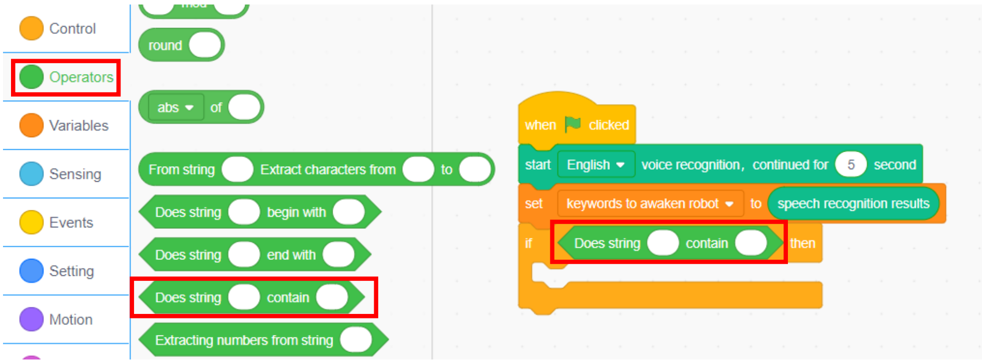

Control
Operators
Variables
Sensing
Events
Setting
Motion
round
abs of
From string Extract characters from to
Does string begin with
Does string end with
Does string contain
Extracting numbers from string
when clicked
start English voice recognition, continued for 5 second
set keywords to awaken robot to speech recognition results
if Does string contain then

Figure 3.8 "Does string () contain ()" block

3) Click the **Variables** tab and drag the "keywords to awaken robot" block to the left elliptical blank in "Does string () contain ()" block. Fill in the right elliptical blank with "I would like to ask the route", as shown in Figure 3.9.

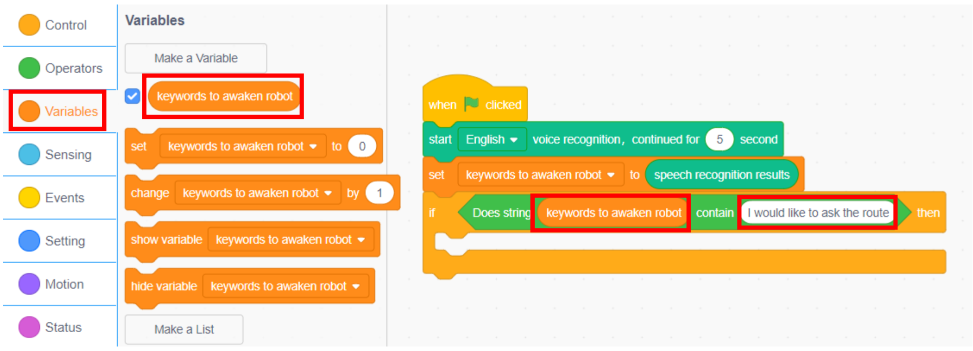

Control
Operators
Variables
Sensing
Events
Setting
Motion
Status
Variables
Make a Variable
keywords to awaken robot
set keywords to awaken robot to 0
change keywords to awaken robot by 1
show variable keywords to awaken robot
hide variable keywords to awaken robot
Make a List
when clicked
start English voice recognition, continued for 5 second
set keywords to awaken robot to speech recognition results
if Does string keywords to awaken robot contain I would like to ask the route then

Figure 3.9 Filling in the blank with judgment condition

Step 8: When the speech about awakening the robot contains the specified keywords, the robot broadcasts "Hello, where do you want to go?". Click the **AI** tab, drag "the robot broadcasts: hello" block into the execution location of "If... then..." block, and change the content "hello" to "where do you want to go". If specified content is not detected, the system returns back to zero, as shown in Figure 3.10.

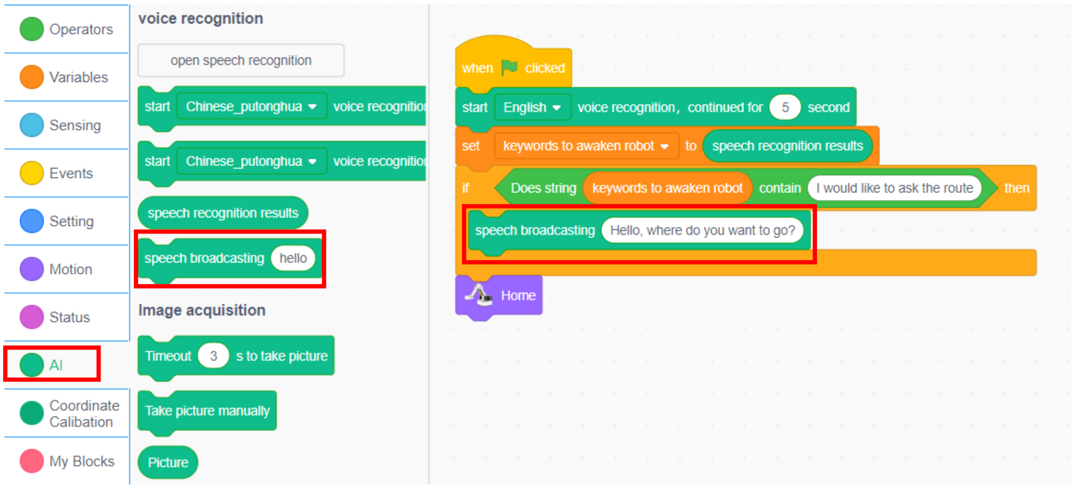

Operators
Variables
Sensing
Events
Setting
Motion
Status
AI
Coordinate Calibation
My Blocks
voice recognition
open speech recognition
start Chinese_putonghua voice recognitio
start Chinese_putonghua voice recognitio
speech recognition results
speech broadcasting hello
Image acquisition
Timeout 3 s to take picture
Take picture manually
Picture
when clicked
start English voice recognition, continued for 5 second
set keywords to awaken robot to speech recognition results
if Does string keywords to awaken robot contain I would like to ask the route then
speech broadcasting Hello, where do you want to go?
Home

Figure 3.10 Speech broadcasting

Step 8: Repeat the execution. Click the **Control** tab, and drag the “forever” block to splice with the “When the green flag is clicked” block, as shown in Figure 3.11.

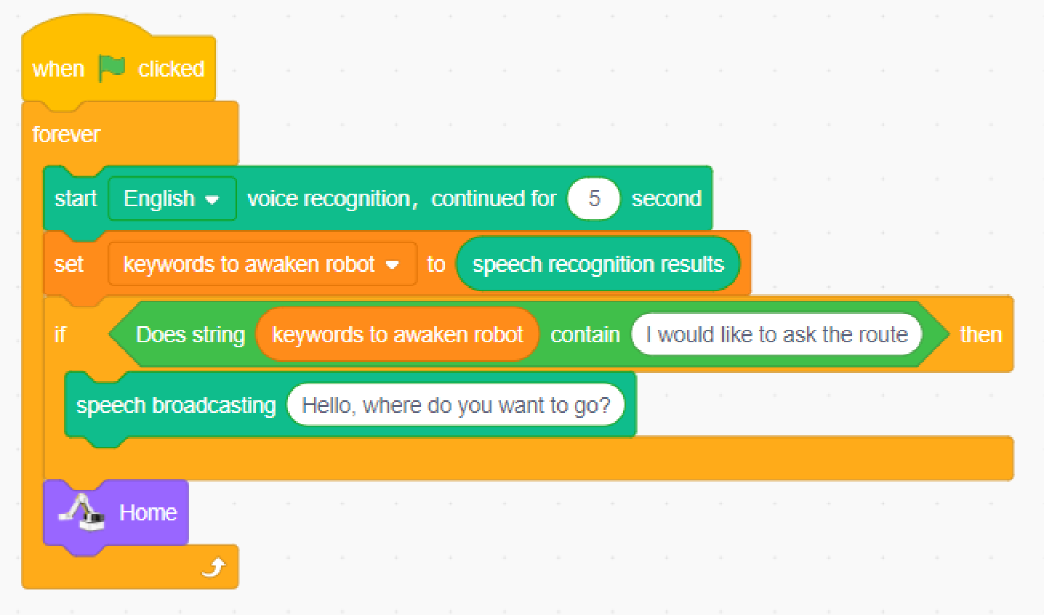

when clicked
forever
start English voice recognition, continued for 5 second
set keywords to awaken robot to speech recognition results
if Does string keywords to awaken robot contain I would like to ask the route then
speech broadcasting Hello, where do you want to go?
Home

Figure 3.11 “forever” block

(1) What happens when you do not use the “forever” block?

(2) What happens when you use blockit?

3. Summary

(1) The creation of a variable requires attention to the application range of the sprite:____

(2) Key steps to awaken the robot:_______________________________

4. Self-Assessment

Check the completed content in the experiment task. Tick (√) the completed items, and circle (◯) the uncompleted items.

Assessment Content	Completion Status
I've skillfully created and used variables	
I've used the instructions about speech input and speech broadcast	
I've used the repeated execution command	
I've known how to use the branch structure	
I've completed the task of speech wakeup robot programming	

Task 2: Program Intelligent Navigation

We have awakened the robot and it can broadcast speech messages. Next, we will program the Scratch to make the robotic arm "hear" our question by speech and then draw the corresponding route based on the Wisdom Mall map and the location asked.

1. Analysis

We analyze the main steps to make the robotic arm give navigation.

(1) Main steps to awaken the robot: speech input, ______________

____

(2) Asking the route to a location in the Wisdom Mall by speech, and recognizing the content in the speech.

(3) Robotic arm ______________________________________

2. Steps

(1) Prepare Hardware

Step 1: After you have prepared the experimental equipment, power up the robotic arm and connect it to the computer.

Step 2: Place the equipment in the corresponding position in the map, as shown in Figure 3.12. Replace the end of the robotic arm with the pen end. Adjust the robotic arm and turn it on.

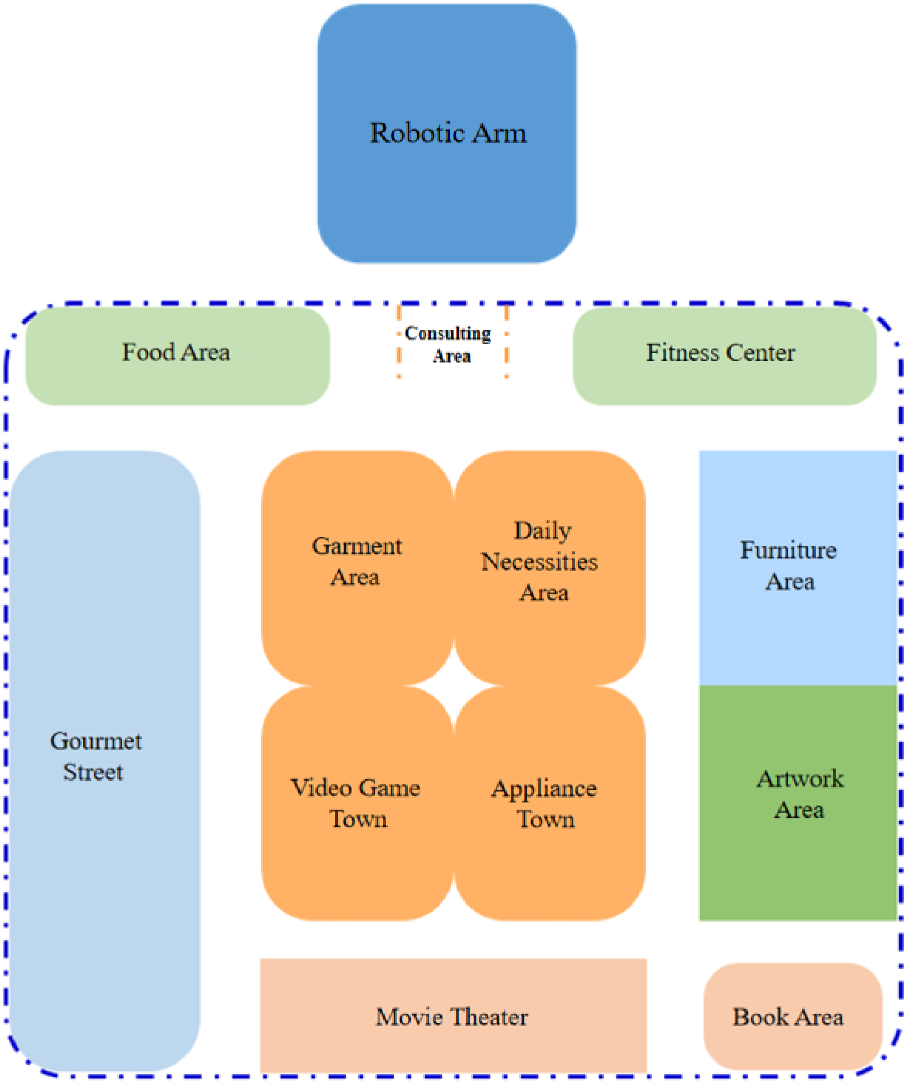


Figure 3.12 Placement

(2) Design Program

Step 1: Understand the flow chart by analyzing the intelligent navigation implementation process, as shown in Figure 3.13.

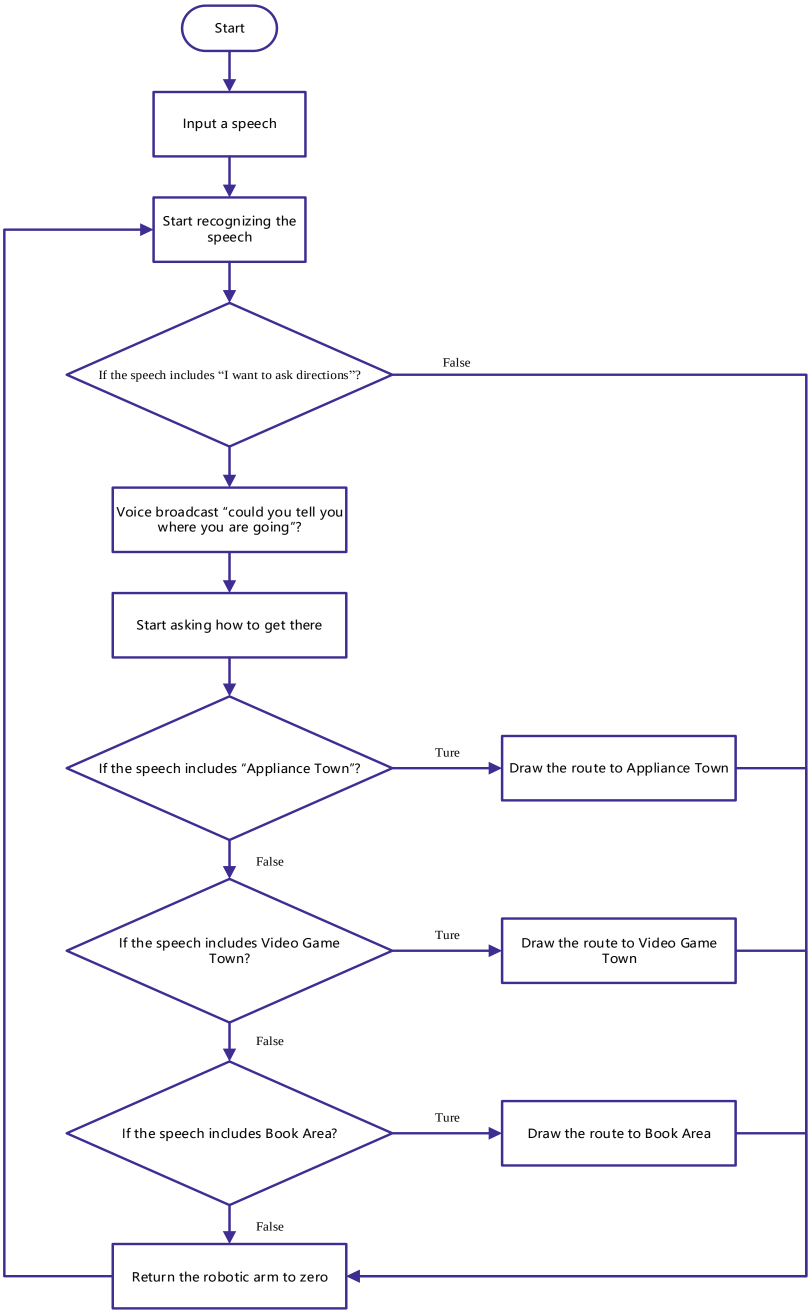

Start
Input a speech
Start recognizing the speech
If the speech includes “I want to ask directions”?
False
Voice broadcast “could you tell you where you are going”?
Start asking how to get there
If the speech includes “Appliance Town”?
Ture
Draw the route to Appliance Town
False
If the speech includes Video Game Town?
Ture
Draw the route to Video Game Town
False
If the speech includes Book Area?
Ture
Draw the route to Book Area
False
Return the robotic arm to zero

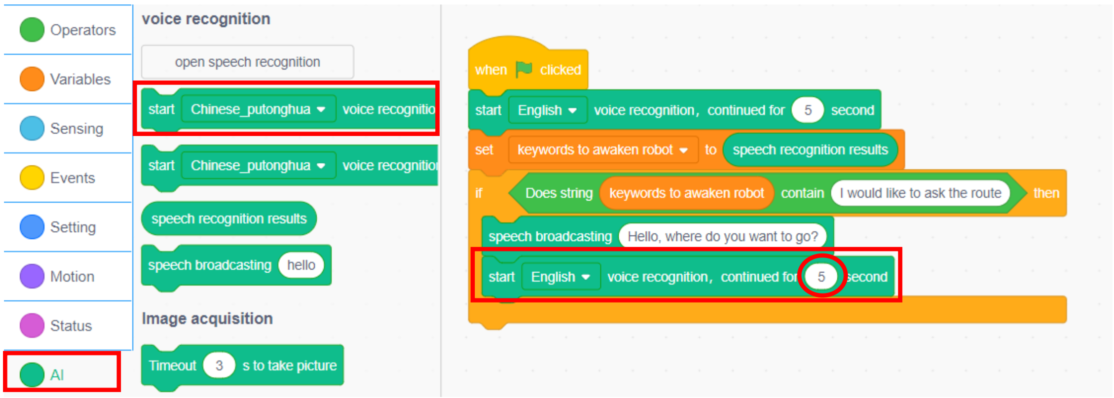
Figure 3.13 Flow chart of intelligent navigation

Step 2: Awaken the robot.

Review the steps to awaken the robot in Task 1.

In Scratch, program the robot and awaken it once.

Step 3: Ask the route. Click the **AI** tab, drag the “Start Chinese_Mandarin (Simplified) voice recognition for 1 second” block to splice with “the robot broadcasts” block, and change the broadcast duration to 5 seconds, as shown in Figure 3.14.


Operators
Variables
Sensing
Events
Setting
Motion
Status
AI
voice recognition
open speech recognition
start Chinese_putonghua voice recognitio
start Chinese_putonghua voice recognitio
speech recognition results
speech broadcasting hello
Image acquisition
Timeout 3 s to take picture

```
when 🏁 clicked
start English ▾ voice recognition, continued for 5 second
set keywords to awaken robot ▾ to speech recognition results
if Does string keywords to awaken robot contain I would like to ask the route then
    speech broadcasting Hello, where do you want to go?
    start English ▾ voice recognition, continued for 5 second
```

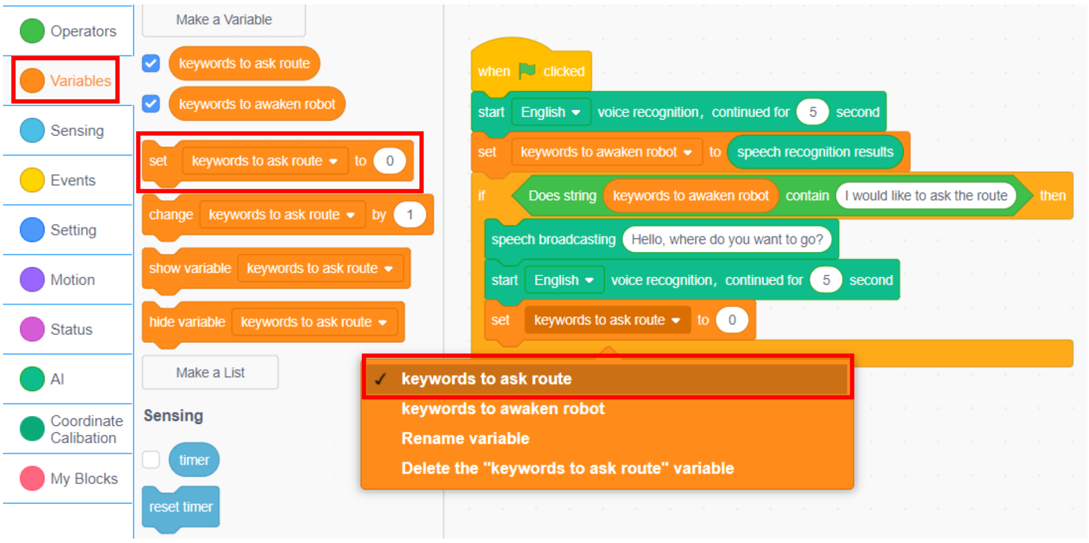
Figure 3.14 Starting to ask the route

Step 4: Create the *keywords to ask route* variable.

Review steps to create the *keywords to awaken robot* variable in Task 1.

In Scratch, try to create a variable and name it *keywords to ask route*.

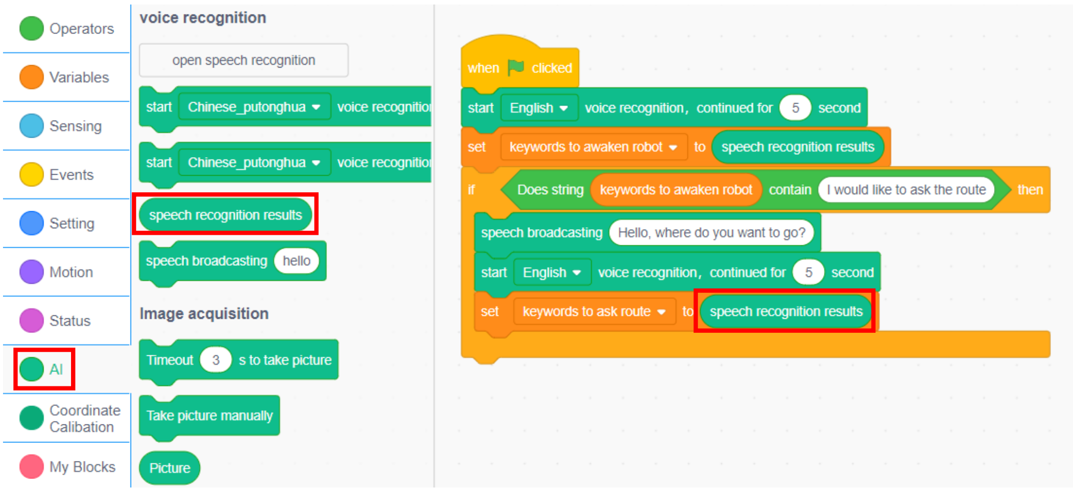
Step 5: Click the **Variables** tab, and drag the “Set the ‘keys to awaken robot’ to (0)” block to splice under the “start English voice recognition, continued for 1 second” block. Click the inverted triangle symbol in the block and select “**keywords to ask route**” from the drop-down menu, as shown in Figure 3.15.



```
when 🏁 clicked
start English voice recognition, continued for 5 second
set keywords to awaken robot to speech recognition results
if Does string keywords to awaken robot contain I would like to ask the route then
    speech broadcasting Hello, where do you want to go?
    start English voice recognition, continued for 5 second
    set keywords to ask route to 0
```

Figure 3.15 Setting the *keywords to ask route* variable

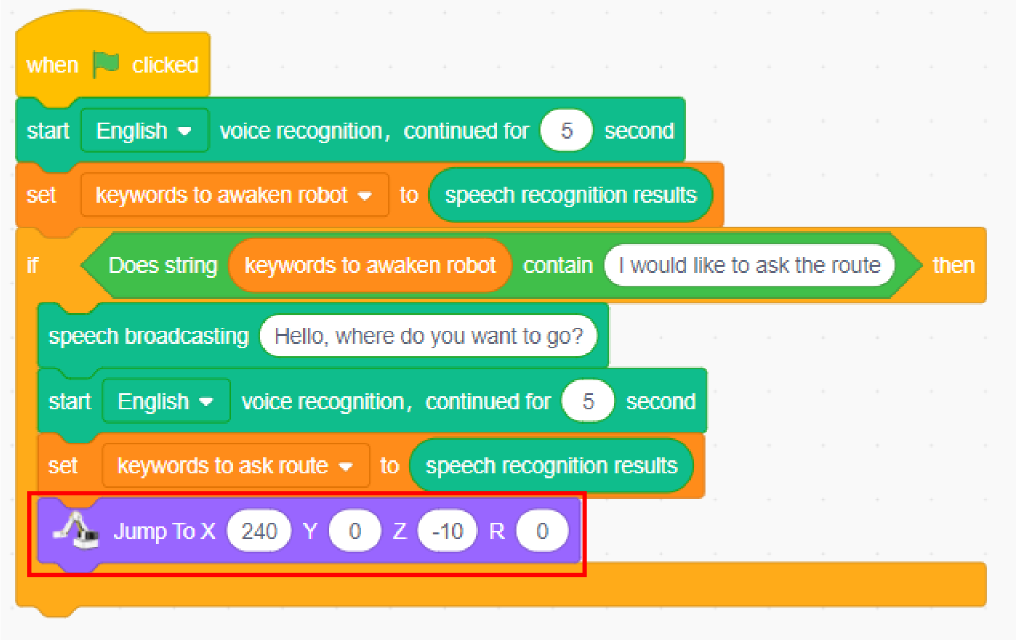
Step 6: Click the **AI** tab, drag the “speech recognition results” block, and set the *keywords to ask route* variable value to the speech recognition results, as shown in Figure 3.16.



```
when 🏁 clicked
start English voice recognition, continued for 5 second
set keywords to awaken robot to speech recognition results
if Does string keywords to awaken robot contain I would like to ask the route then
    speech broadcasting Hello, where do you want to go?
    start English voice recognition, continued for 5 second
    set keywords to ask route to speech recognition results
```

Figure 3.16 Setting the *keywords to ask route* value to speech recognition results

Step 7: Move the robotic arm to the Consulting Area. Click the **Motion** tab, drag the “Jump To” block, and change the coordinates to (240, 0, -10) (the coordinate values can be changed based on actual situation), as shown in Figure 3.17.



```
when [flag] clicked
start English voice recognition, continued for 5 second
set keywords to awaken robot to speech recognition results
if Does string keywords to awaken robot contain I would like to ask the route then
    speech broadcasting Hello, where do you want to go?
    start English voice recognition, continued for 5 second
    set keywords to ask route to speech recognition results
    Jump To X 240 Y 0 Z -10 R 0
```

Figure 3.17 Jump To Consulting Area

Step 8: Move from Consulting Area to Appliance Town.

1) When drawing the route, make the robotic arm move ahead, then move rightward a distance, and finally move ahead again. The shortest route from Consulting Area to Appliance Town is shown in Figure 3.18.

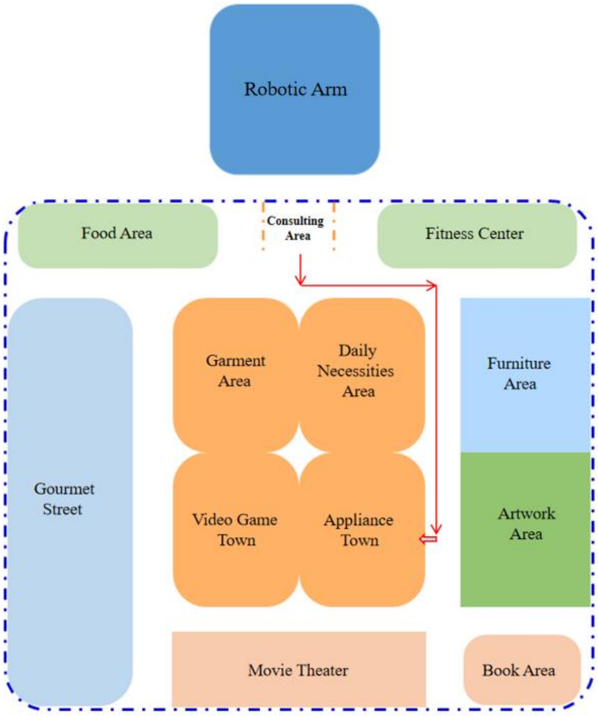
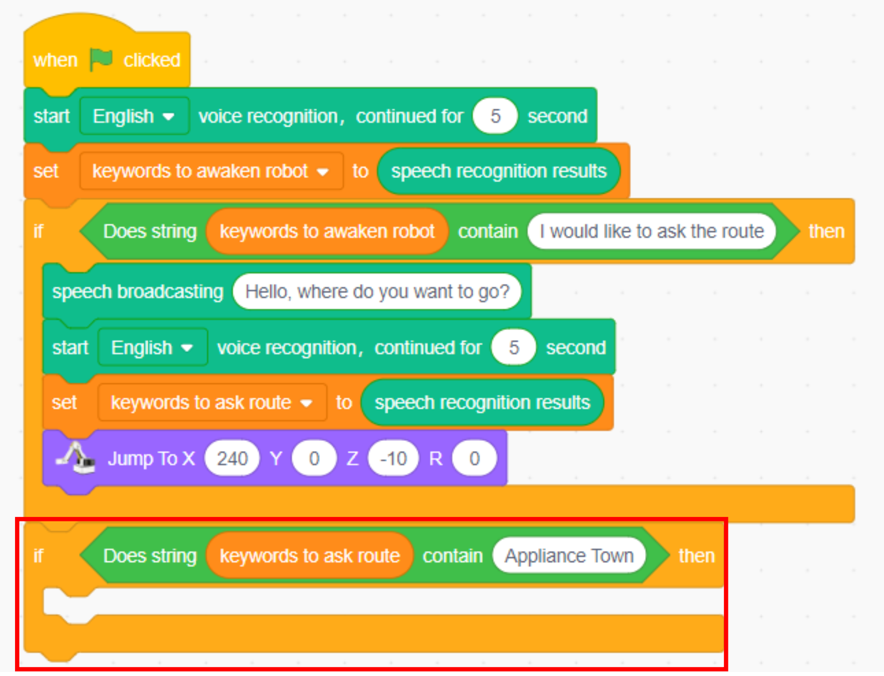


Figure 3.18 Route to Appliance Town

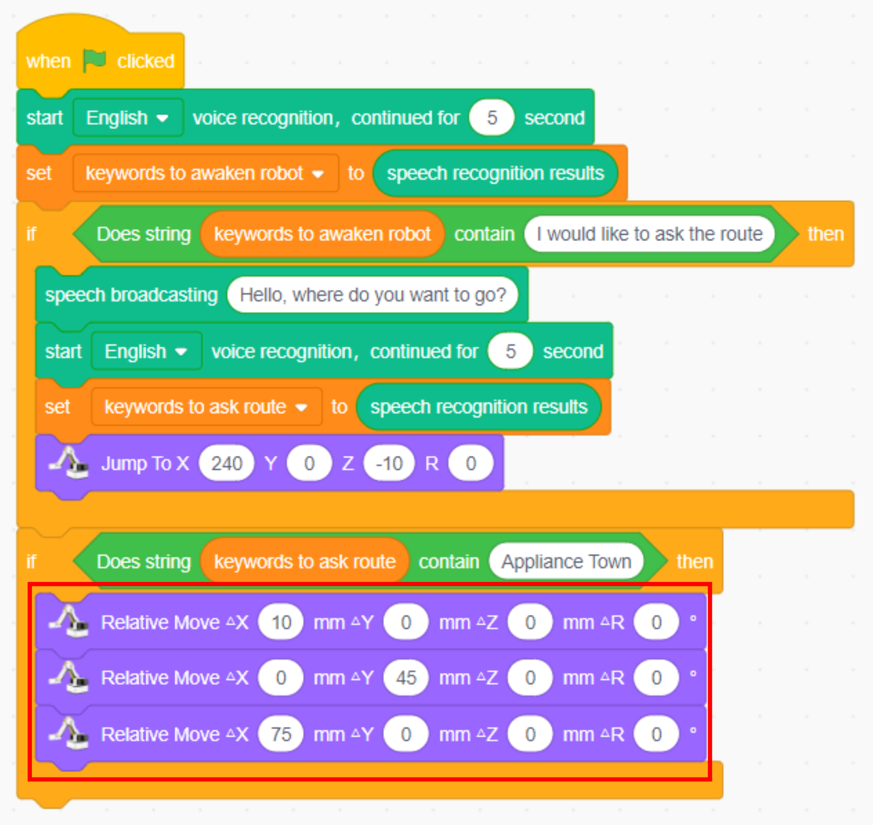
2) Judge if the speech about asking the route contains the keywords “Appliance Town”. Click the **Control** tab and drag the “If... then...” block to splice with the previous block. Click the **Operators** tab, and drag "Does string () contain ()" block into the hexagonal box in the “If... then... ”block. Click the **Variables** tab, and drag the “keywords to ask route” block to the left elliptical blank in “Does string () contain ()” block. Fill in the right elliptical blank with “Appliance Town”, as shown in Figure 3.19.



```
when 🏁 clicked
start English ▾ voice recognition, continued for (5) second
set keywords to awaken robot ▾ to (speech recognition results)
if <Does string (keywords to awaken robot) contain (I would like to ask the route)> then
    speech broadcasting (Hello, where do you want to go?)
    start English ▾ voice recognition, continued for (5) second
    set keywords to ask route ▾ to (speech recognition results)
    Jump To X (240) Y (0) Z (-10) R (0)

if <Does string (keywords to ask route) contain (Appliance Town)> then

```

Figure 3.19 Judging the keywords in speech about asking route

3) From the Consulting Area, the robotic arm moves ahead 10 mm, then moves rightward 45 mm, finally turns left and moves backward 75mm. The route of each motion can be modified based on actual situation. Click the **Motion** tab, drag the “Relative Move” block three times to the execution location of the “If... then...” block, and modify the corresponding value, as shown in Figure 3.20.



```
when ⚑ clicked
start English ▾ voice recognition，continued for 5 second
set keywords to awaken robot ▾ to speech recognition results
if Does string keywords to awaken robot contain I would like to ask the route then
    speech broadcasting Hello, where do you want to go?
    start English ▾ voice recognition，continued for 5 second
    set keywords to ask route ▾ to speech recognition results
    Jump To X 240 Y 0 Z -10 R 0
if Does string keywords to ask route contain Appliance Town then
    Relative Move ΔX 10 mm ΔY 0 mm ΔZ 0 mm ΔR 0 °
    Relative Move ΔX 0 mm ΔY 45 mm ΔZ 0 mm ΔR 0 °
    Relative Move ΔX 75 mm ΔY 0 mm ΔZ 0 mm ΔR 0 °
```

Figure 3.20 Drawing the route by the robotic arm

In addition to the “Relative Move” block, which motion blocks can the robotic arm use when it is drawing the route?

Step 9: Move from Consulting Area to Video Game Town. When drawing the route, the robotic arm moves ahead, then turns left and moves a distance, and finally moves ahead again. The shortest route from Consulting Area to Video Game Town is shown in Figure 3.21.

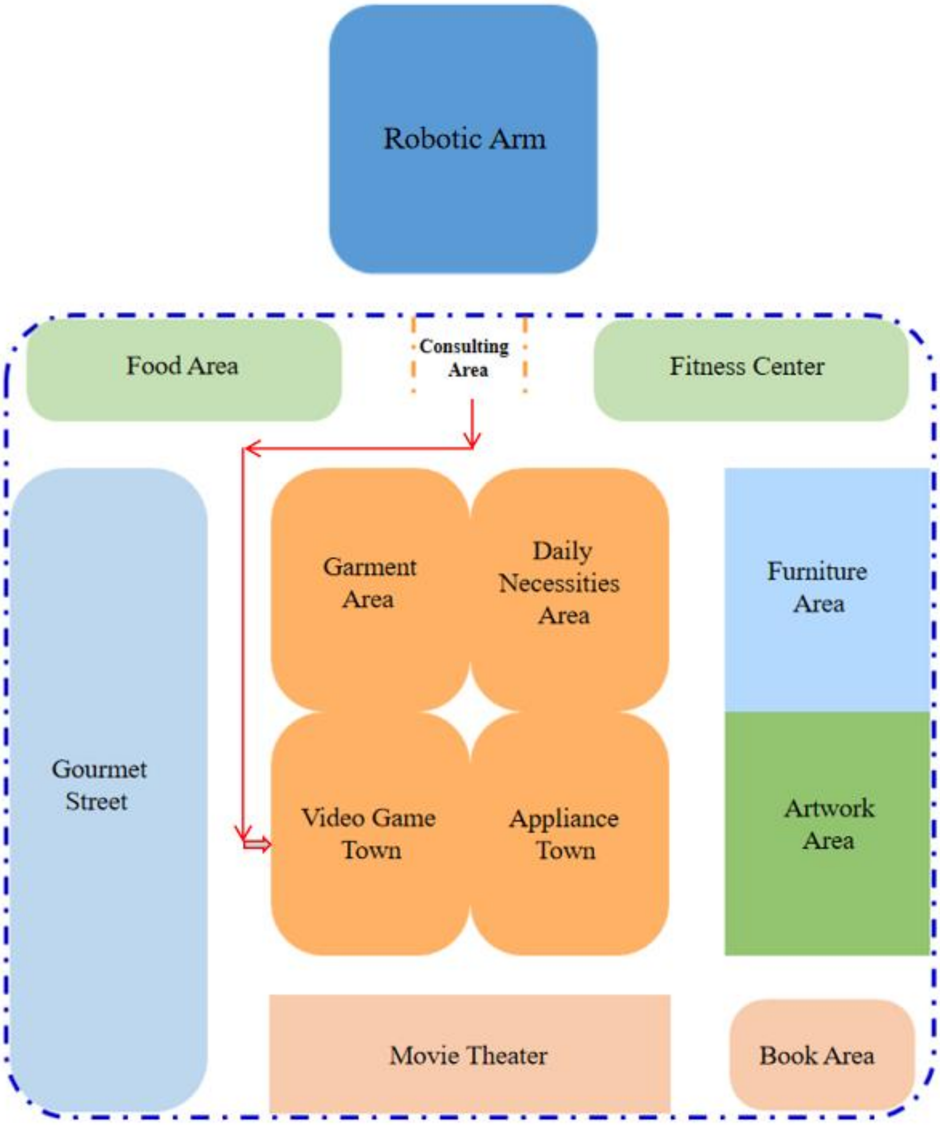


Figure 3.21 Route to Video Game Town

Review the planning of the route to the Video Game Town and its programming, and familiarize yourself with the route to the Video Game Town.

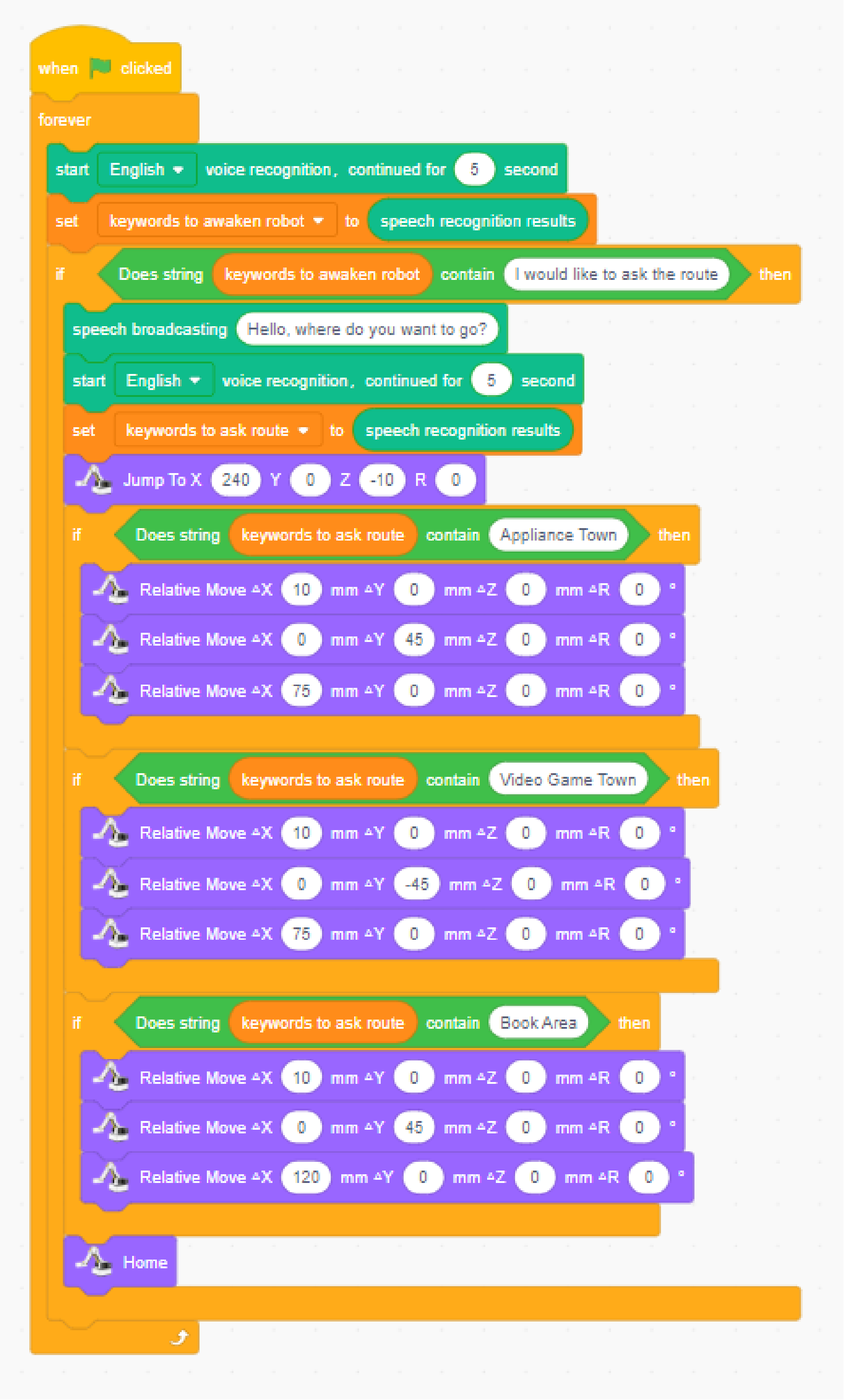
By programming in Scratch, make the robotic arm draw the shortest route from Consulting Area to Video Game Town when the speech about asking the route contains the keywords “Video Game

Step 10: Move from Consulting Area to Book Area.

Analyze the shortest route from Consulting Area to Book Area.

By programming in Scratch, make the robotic arm draw the shortest route from Consulting Area to Book Area.

Step 11: Repeat the execution. Click the **Control** tab, and drag the “forever” block to splice with the “When the green flag is clicked” block, as shown in Figure 3.22.



```
when 🏴 clicked
forever
    start English ▾ voice recognition, continued for (5) second
    set keywords to awaken robot ▾ to (speech recognition results)
    if <Does string (keywords to awaken robot) contain (I would like to ask the route)> then
        speech broadcasting (Hello, where do you want to go?)
        start English ▾ voice recognition, continued for (5) second
        set keywords to ask route ▾ to (speech recognition results)
        Jump To X (240) Y (0) Z (-10) R (0)
        if <Does string (keywords to ask route) contain (Appliance Town)> then
            Relative Move ΔX (10) mm ΔY (0) mm ΔZ (0) mm ΔR (0) °
            Relative Move ΔX (0) mm ΔY (45) mm ΔZ (0) mm ΔR (0) °
            Relative Move ΔX (75) mm ΔY (0) mm ΔZ (0) mm ΔR (0) °
        if <Does string (keywords to ask route) contain (Video Game Town)> then
            Relative Move ΔX (10) mm ΔY (0) mm ΔZ (0) mm ΔR (0) °
            Relative Move ΔX (0) mm ΔY (-45) mm ΔZ (0) mm ΔR (0) °
            Relative Move ΔX (75) mm ΔY (0) mm ΔZ (0) mm ΔR (0) °
        if <Does string (keywords to ask route) contain (Book Area)> then
            Relative Move ΔX (10) mm ΔY (0) mm ΔZ (0) mm ΔR (0) °
            Relative Move ΔX (0) mm ΔY (45) mm ΔZ (0) mm ΔR (0) °
            Relative Move ΔX (120) mm ΔY (0) mm ΔZ (0) mm ΔR (0) °
        Home
```

Figure 3.22 Repeating the execution

3. Summary

Main steps to make the robotic arm give navigation:

4. Self-Assessment

Check the completed content in the experiment task. Tick (√) the completed items, and circle (◯) the uncompleted items.

Assessment Content	Completion Status
I've skillfully used the instructions about speech input and broadcast	
I've used the “If... then... ”block	
I've used the repeated execution instruction	
I've known how to use the branch structure and loop structure	
I've completed the task of speech wakeup robot programming	

Experiment 4

Making Hamburgers

Team Name: Team Member: Date:

Overview

Do you like hamburgers? If yes, you can make hamburgers in a simple way at home! This experiment is to enable the robotic arm to automatically produce delicious hamburgers. The experiment will use Tinkercad software and UltimakerCura software for 3D modeling and slicing. The ingredients for making hamburgers, such as buns, vegetables and meat slices, are produced by 3D printing. After preparing these ingredients, we will use DobotScratch to program for hamburger making by the robotic arm.

Objective

1. Understand the main interface of Tinkercad software by designing the bun model.
2. Master the basic knowledge of UltimakerCura slicing software by designing the bun model.
3. Master how Tinkercad builds the basic geometry and adjusts geometric shape by designing the bun model.
4. By designing the bun model, understand how to print a model with the printer.
5. By designing the vegetable model, learn how Tinkercad draws the geometry.
6. Understand the image recognition technology and master the use of image recognition instructions by programming the robotic arm to make hamburgers.

7. By programming the robotic arm to make hamburgers, understand how to simplify the program and master the function of self-made blocks.

Equipment

Equipment Picture	Name	Quantity
	Dobot Magician Lite Robotic arm	1
	Suction cup kit	1
	Camera	1
	Type-C cable	1
	Power adapter	1
	MOOZ-2	1
	PLA	4

Requirements

- Take care when using electricity.
- Before the experiment, check whether the experimental equipment is complete and intact. If there is any omission or damage, please report to the teacher.
- Any specific operations in the experiment shall be performed according to the experiment manual. If you have any questions, please promptly ask the teacher.
- During the experiment, the joints will start to work as the robotic arms are powered on. In that case, do not move the joints of the robotic arms hard if you do not press the unlock key.
- Report any device fault during the experiment to your teacher in a timely manner, and do not handle it yourself.
- After the experiment, the equipment shall be arranged well. You are allowed to leave the laboratory only after the inspection by the group leader.

Task 1: Design the Bun Model

Design the bun model using Tinkercad and print the bun with a 3D printer.

1. Analysis

Analyze the steps to design the bun model, and follow the prompts to fill in the table below.

Step 1: Start Tinkercad.

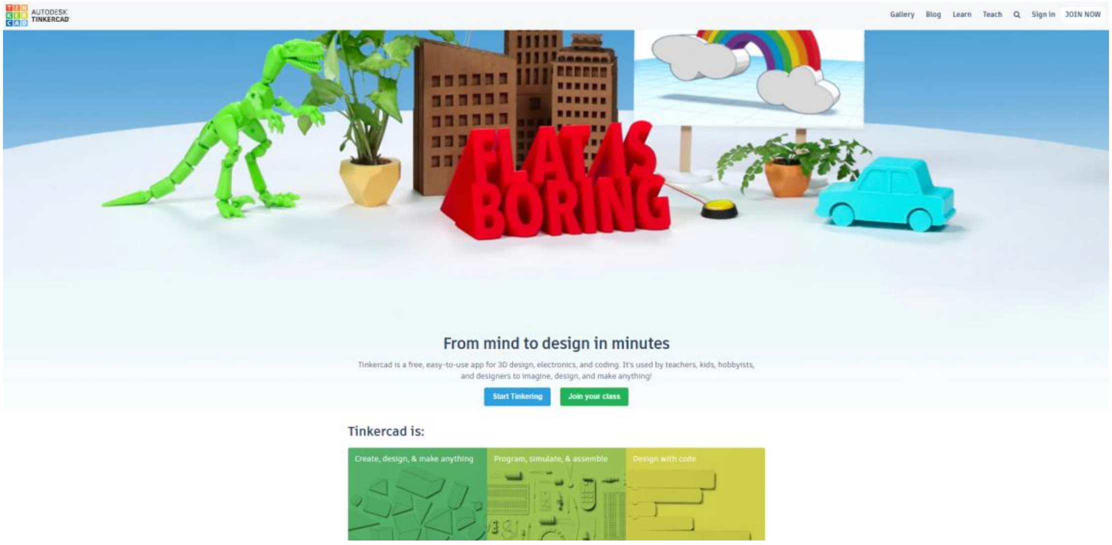
Step 2: Create a project.

Step 3: Design the bun model for the hamburger: Select the hemispheroid from Basic Shapes → ______________________

(Complete the remaining steps as prompted)

2. Steps

Step 1: Visit the website https://www.tinkercad.com/, as shown in Figure 4.1.



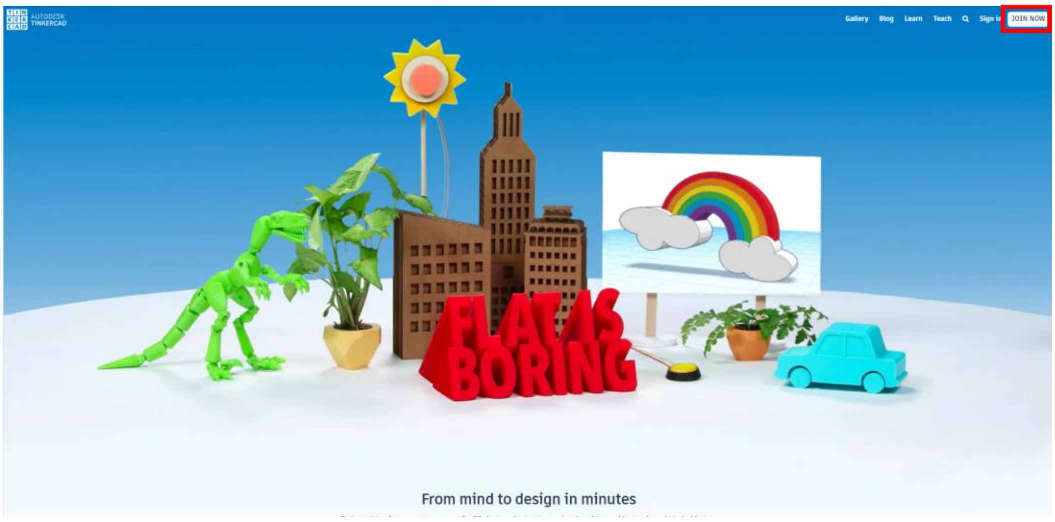
From mind to design in minutes

Tinkercad is a free, easy-to-use app for 3D design, electronics, and coding. It's used by teachers, kids, hobbyists, and designers to imagine, design, and make anything!

Tinkercad is:

Figure 4.1 Visiting the Tinkercad website

Step 2: Create a Tinkercad account and click **Join now** in the upper right corner of the page, as shown in Figure 4.2, to go to the interface as shown in Figure 4.3. Select **Create a personal account**, and then select a mode from the list to register your account, as shown in Figure 4.4.



From mind to design in minutes

Figure 4.2 Registering your Tinkercad account

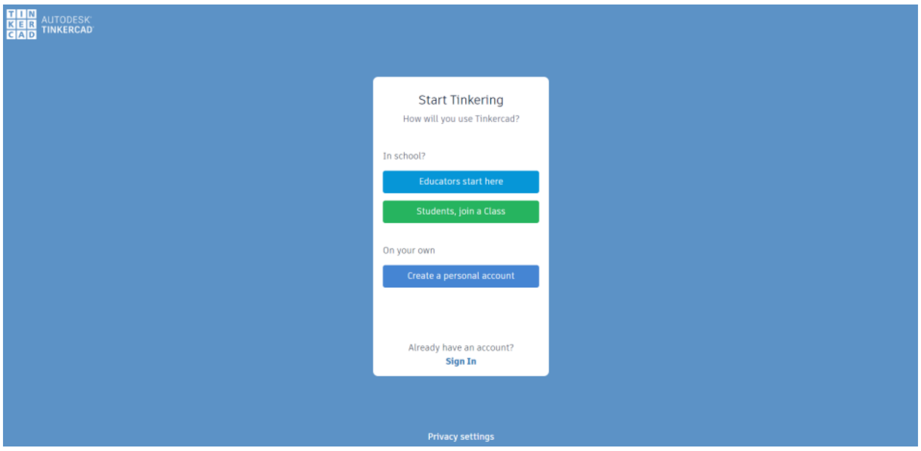


Figure 4.3 **Creating a personal account”**

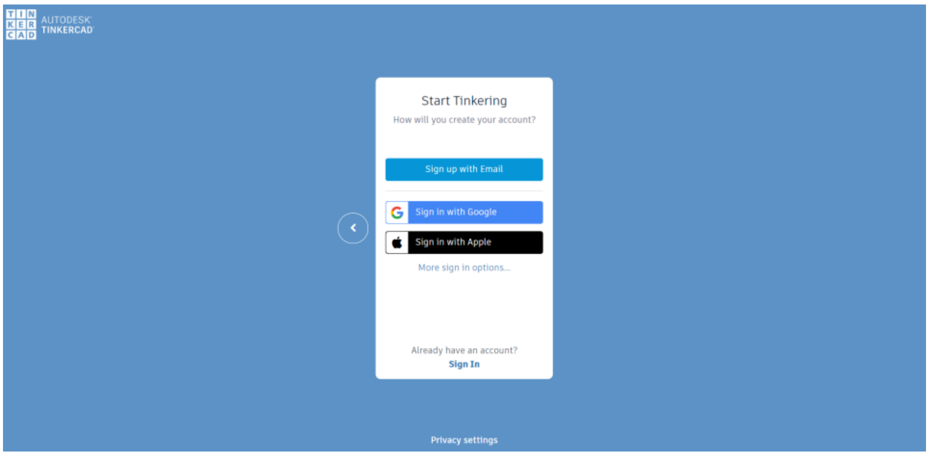


Figure 4.4 Selecting a mode of account registration

Step 3: Create a project. After logging in to your personal account, you can start to create a project, as shown in Figure 4.5. Add a description, as shown in Figure 4.6. Click the drop-down menu to the right of **Create** and select **3D Design**, as shown in Figure 4.7.

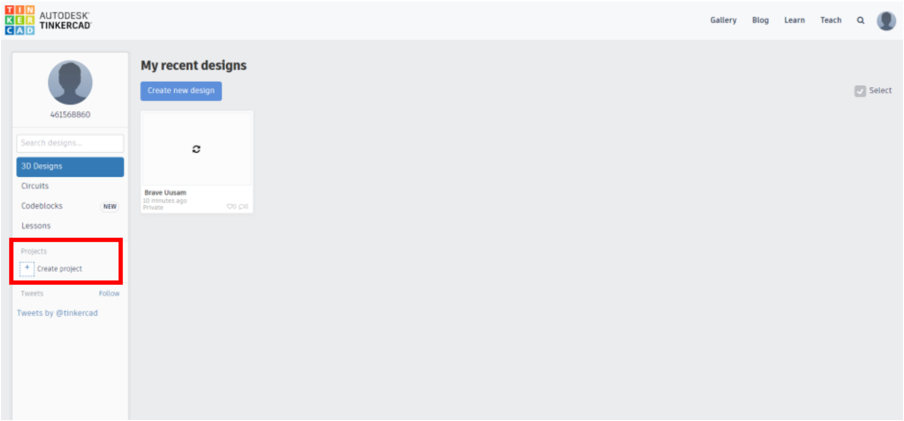


Figure 4.5 Creating the project

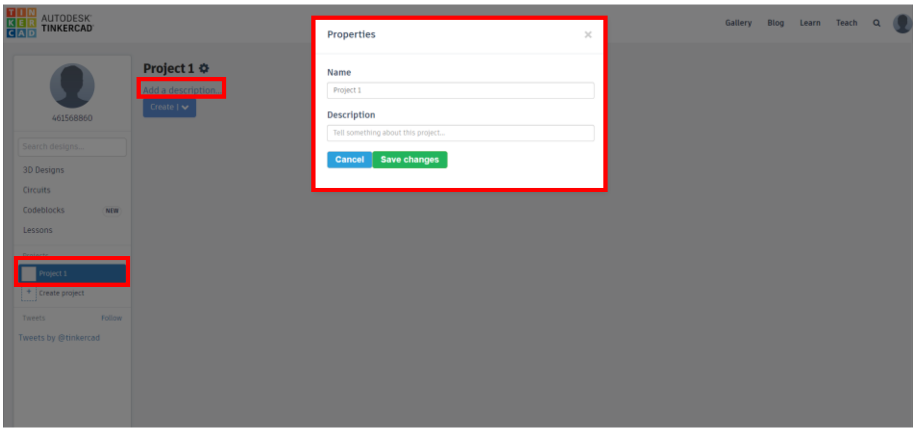


Figure 4.6 Adding a description

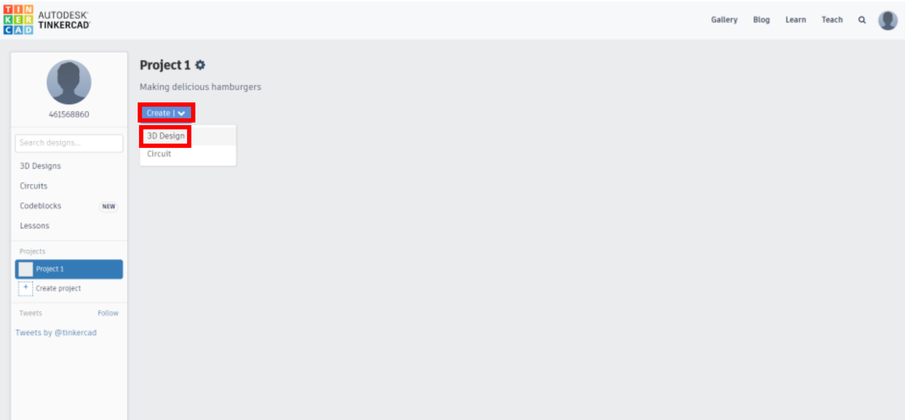


Figure 4.7 Selecting the 3D Design

Main interface of the software

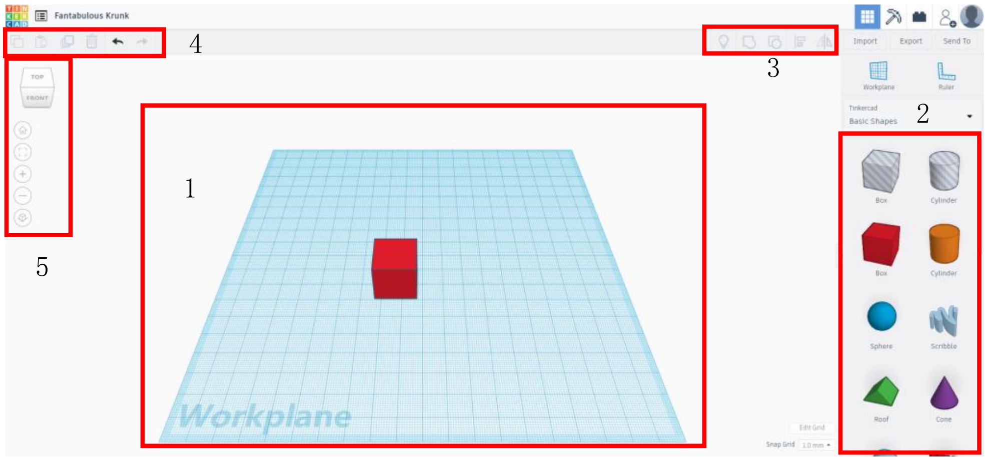
The blue area (labeled as 1) at the center of the page: This area is called workplane, where we conduct all design work.

The geometrical panel on the right of the page (labeled as 2): Here are basic blocks of anything to be designed by us.

The menus on the top (separately labeled as 3 and 4): We use these menu items to perform operations such as alignment and merging.

The toolbar on the left (labeled as 5): Here we can switch views of the model among top view, bottom view, left view, right view, front view, back view and axonometric view; here are also functions such as default viewing angle, viewing angle adjustment, and zoom in and out.

Right-click and drag the cursor to change the 3D perspective of the Workplane. Scroll the mouse up and down to zoom in and out.



Main interface of Tinkercad

Step 4: Open a new project page through the previous steps, as shown in Figure 4.8.

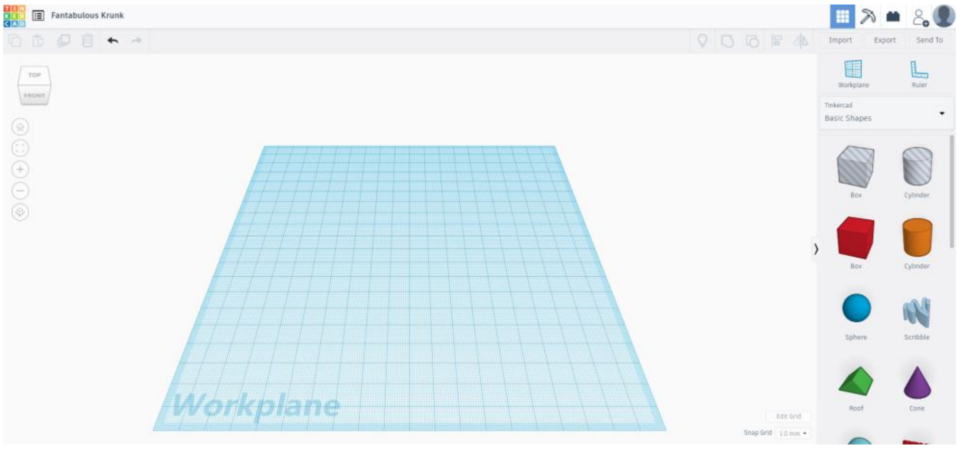


Figure 4.8 Project page

Step 5: Understand Tinkercad's main interface. Then, start to design the bun model. Observe the real hamburger bun, as shown in Figure 4.10, and design a bun model using the image as a template.

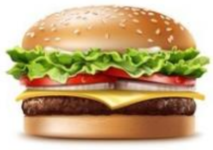


Figure 4.10 Hamburger

Step 6: Observe that the upper bud of the hamburger is like a half sphere. Select **Half Sphere** from **Basic Shapes** and drag it to the Workplane, as shown in Figure 4.11.

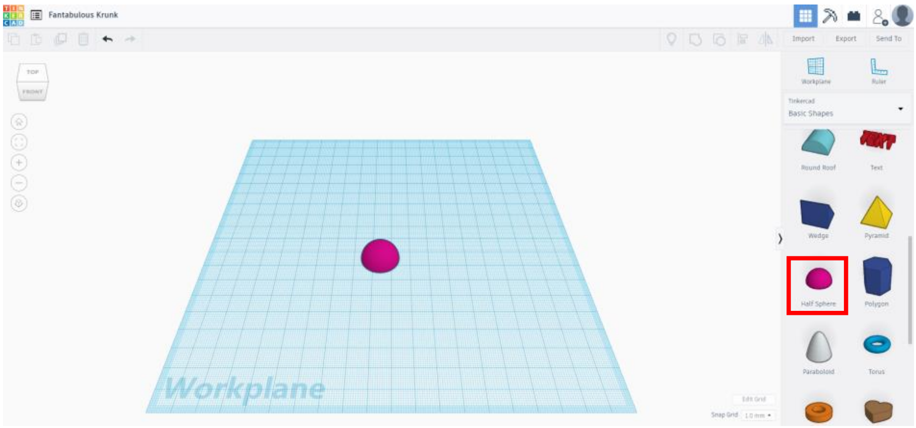


Figure 4.11 Selecting the half sphere

Step 7: Select the Half Sphere on the Workplane. You will find that the half sphere has five white square handles, three of which are over the half sphere, and two of which below, as shown in Figure 4.12.

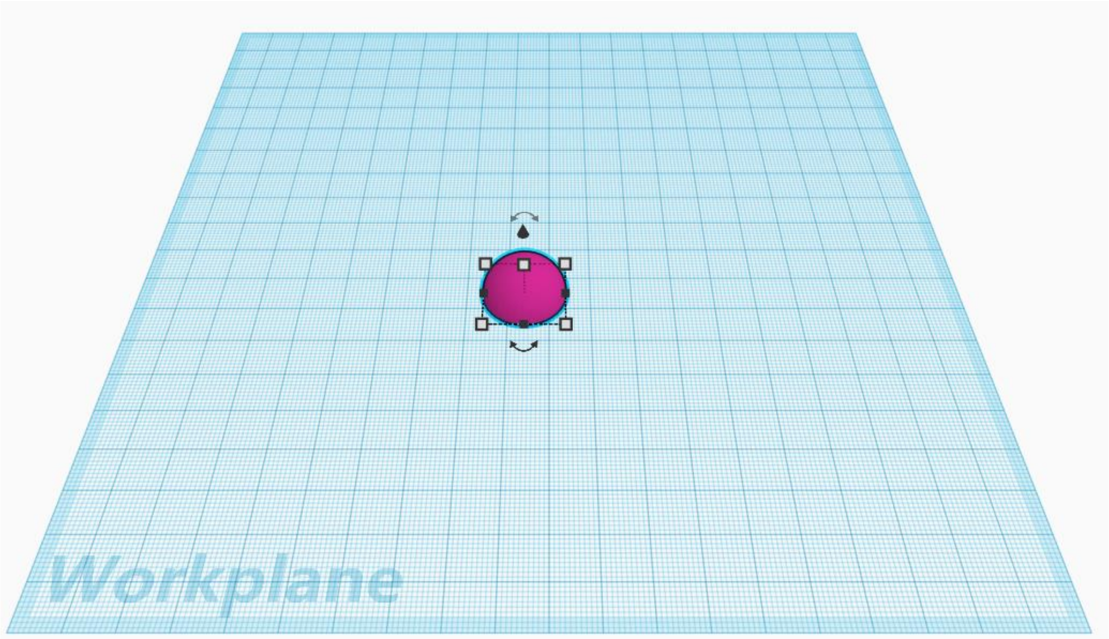


Figure 4.12 White square controls

Step 8: Allow you to change the half sphere’s size through the white square controls. As shown in Figure 4.13, rely on the white square handles within orange boxes to adjust the half sphere’s length and width. Set the diameter of the half sphere bottom to 85 mm, and drag a white square handle until both the length and width of the half sphere become 85 mm, as shown in Figure 4.14. Use the white square control within the orange box to adjust the half sphere’s height. Set the half sphere’s height to 10 mm. Click and drag the white square handle down until the height becomes 10 mm, as shown in Figure 4.15.

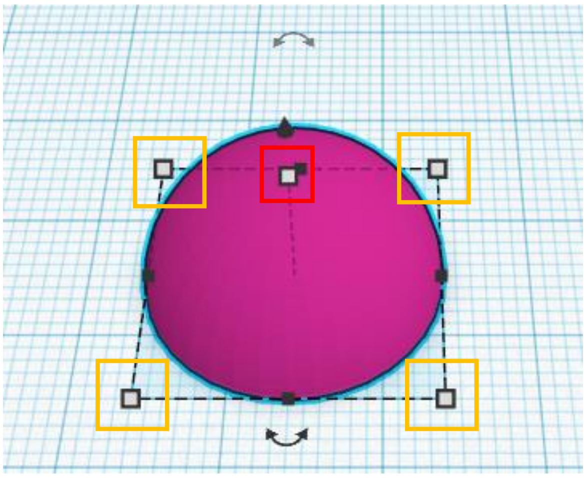


Figure 4.13 Changing the size of Tinkercad shape

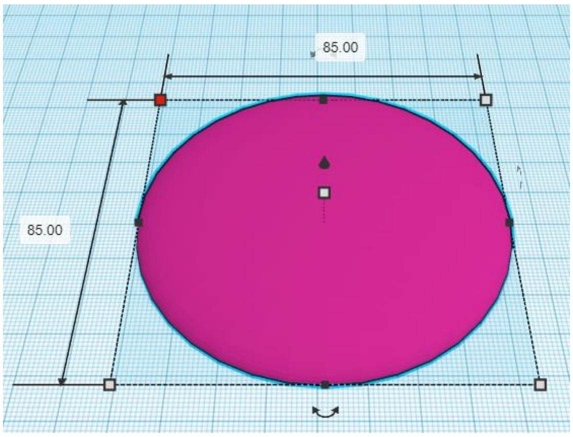


Figure 4.14 Adjusting the length and width to 85 mm

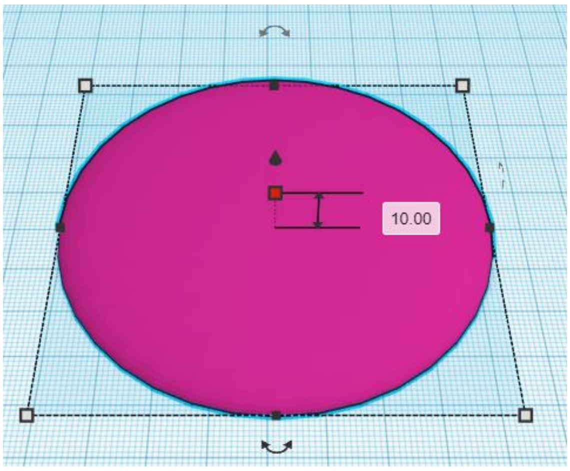


Figure 4.15 Adjusting the height to 10 mm

Step 9: Click the bun model and change the model material to Hole, as shown in Figure 4.16.

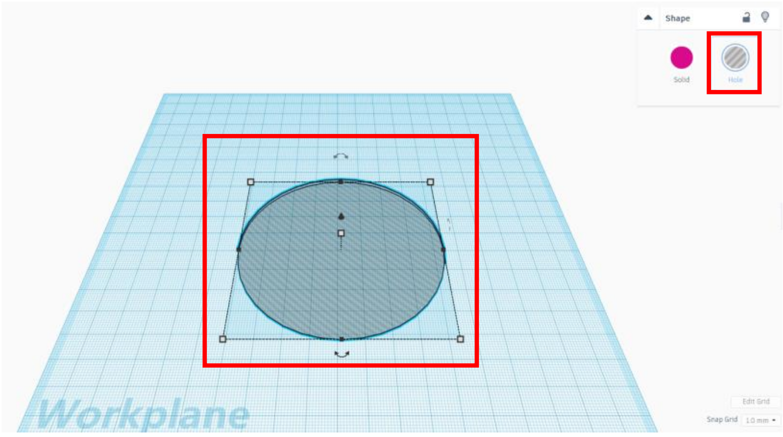


Figure 4.16 Changing the material to Hole

Step 10: Export the designed bun model, as shown in Figure 4.17. Select the **.STL** format, as shown in Figure 4.18.

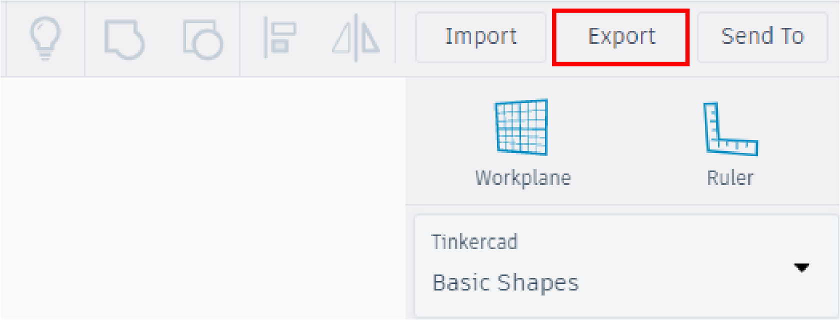
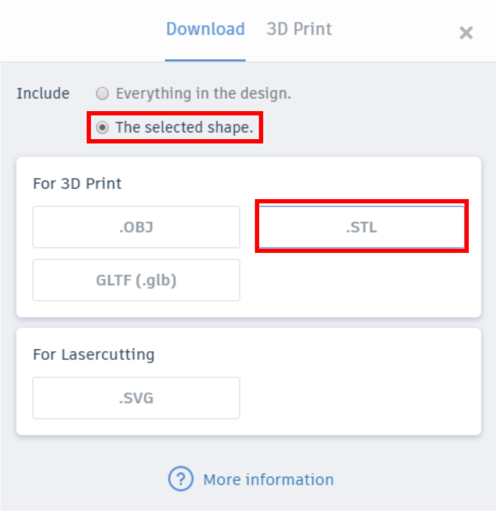


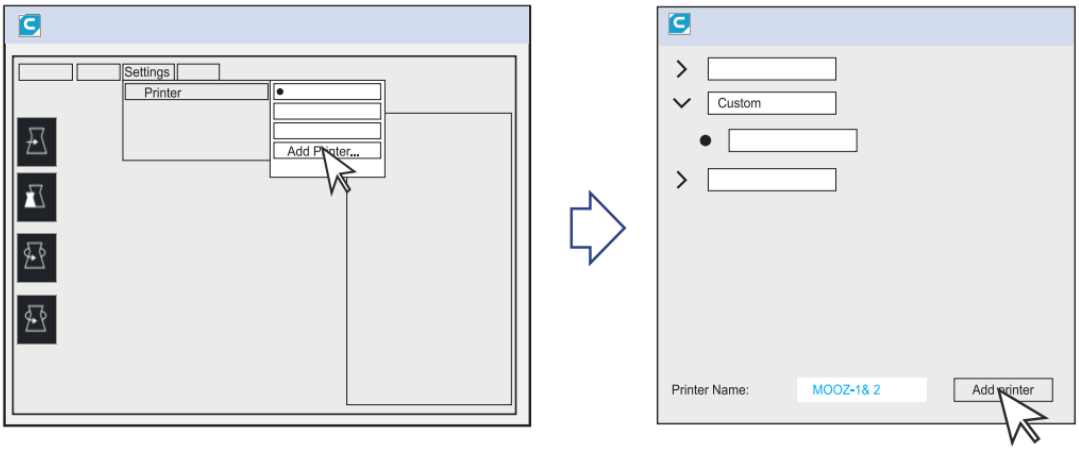
Figure 4.17 Exporting the model



Include ○ Everything in the design.
● The selected shape.

Figure 4.18 “.STL” format

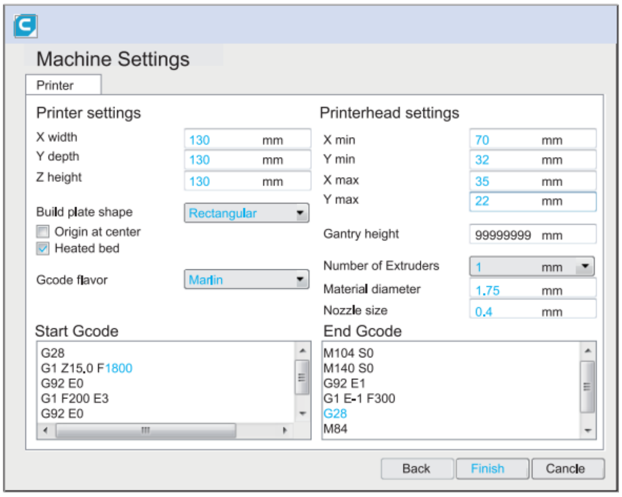
Step 11: Design the bun model, and slice it. Run Cura3.1.0, choose **Settings**>**Printer**>**Add printer**>**Custom**, and then enter **MOOZ-1&2** in the **Printer Name** field, as shown in Figure 4.19.



Printer Name: MOOZ-1& 2

Figure 4.19 Adding the printer

Step 12: Click **Add printer**. The printer setting interface pops up. Set the printer parameters, as shown in figure 4.20.



Machine Settings

Printer

Printer settings

X width 130 mm

Y depth 130 mm

Z height 130 mm

Build plate shape Rectangular

□ Origin at center

☑ Heated bed

Gcode flavor Marlin

Printerhead settings

X min 70 mm

Y min 32 mm

X max 35 mm

Y max 22 mm

Gantry height 99999999 mm

Number of Extruders 1 mm

Material diameter 1.75 mm

Nozzle size 0.4 mm

Start Gcode

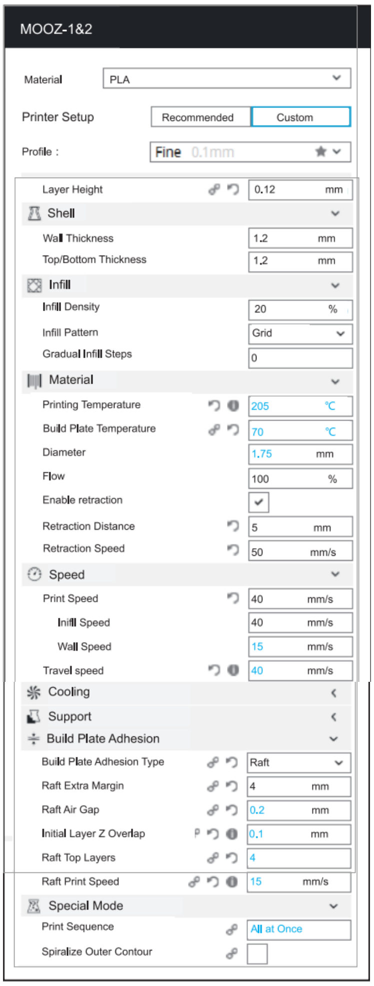
```
G28
G1 Z15.0 F1800
G92 E0
G1 F200 E3
G92 E0
```

End Gcode

```
M104 S0
M140 S0
G92 E1
G1 E-1 F300
G28
M84
```

Figure 4.20 Setting the printer parameters

Step 13: Set key printer parameters, as shown in Figure 4.21.



MOOZ-1&2

Material PLA

Printer Setup: Recommended | Custom

Profile : Fine 0.1mm

Layer Height 0.12 mm

Shell

Wall Thickness 1.2 mm

Top/Bottom Thickness 1.2 mm

Infill

Infill Density 20 %

Infill Pattern Grid

Gradual Infill Steps 0

Material

Printing Temperature 205 ℃

Build Plate Temperature 70 ℃

Diameter 1.75 mm

Flow 100 %

Enable retraction ☑

Retraction Distance 5 mm

Retraction Speed 50 mm/s

Speed

Print Speed 40 mm/s

Infill Speed 40 mm/s

Wall Speed 15 mm/s

Travel speed 40 mm/s

Cooling

Support

Build Plate Adhesion

Build Plate Adhesion Type Raft

Raft Extra Margin 4 mm

Raft Air Gap 0.2 mm

Initial Layer Z Overlap 0.1 mm

Raft Top Layers 4

Raft Print Speed 15 mm/s

Special Mode

Print Sequence All at Once

Spiralize Outer Contour □

Figure 4.21 Description of key printer parameters

Step 14: Click **File (F)**>**Open Files(s) (O)** to select the model to be sliced. Cura supports the model in STL / OBJ / AMF format, as shown in Figure 4.22.

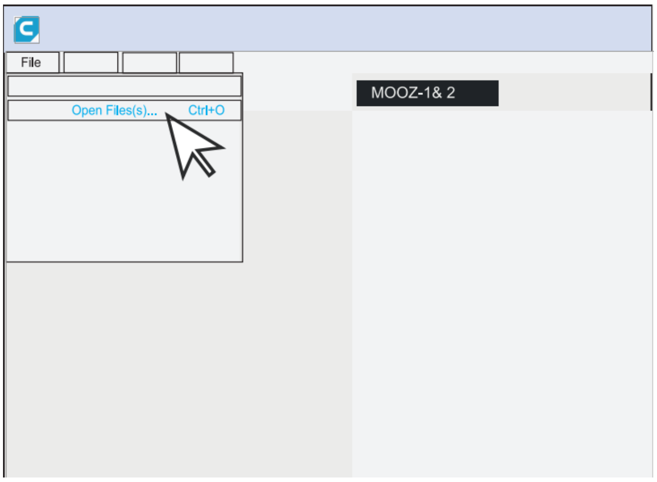


Figure 4.22 Opening the model to be sliced

Step 15: Adjust the model. Click the model, and five options appear on the left side of the interface - **Move**, **Scale**, **Rotate**, **Mirror**, and **Per Model Settings. Among them, Per Model Settings** adjusts key parameters of the printer and is not used usually. Adjust these parameters as needed and move the model to the center of the print platform, as shown in Figure 4.23.

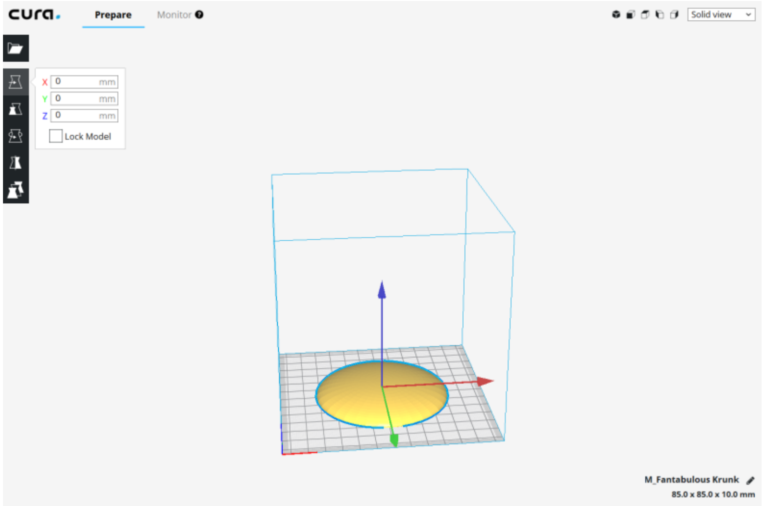
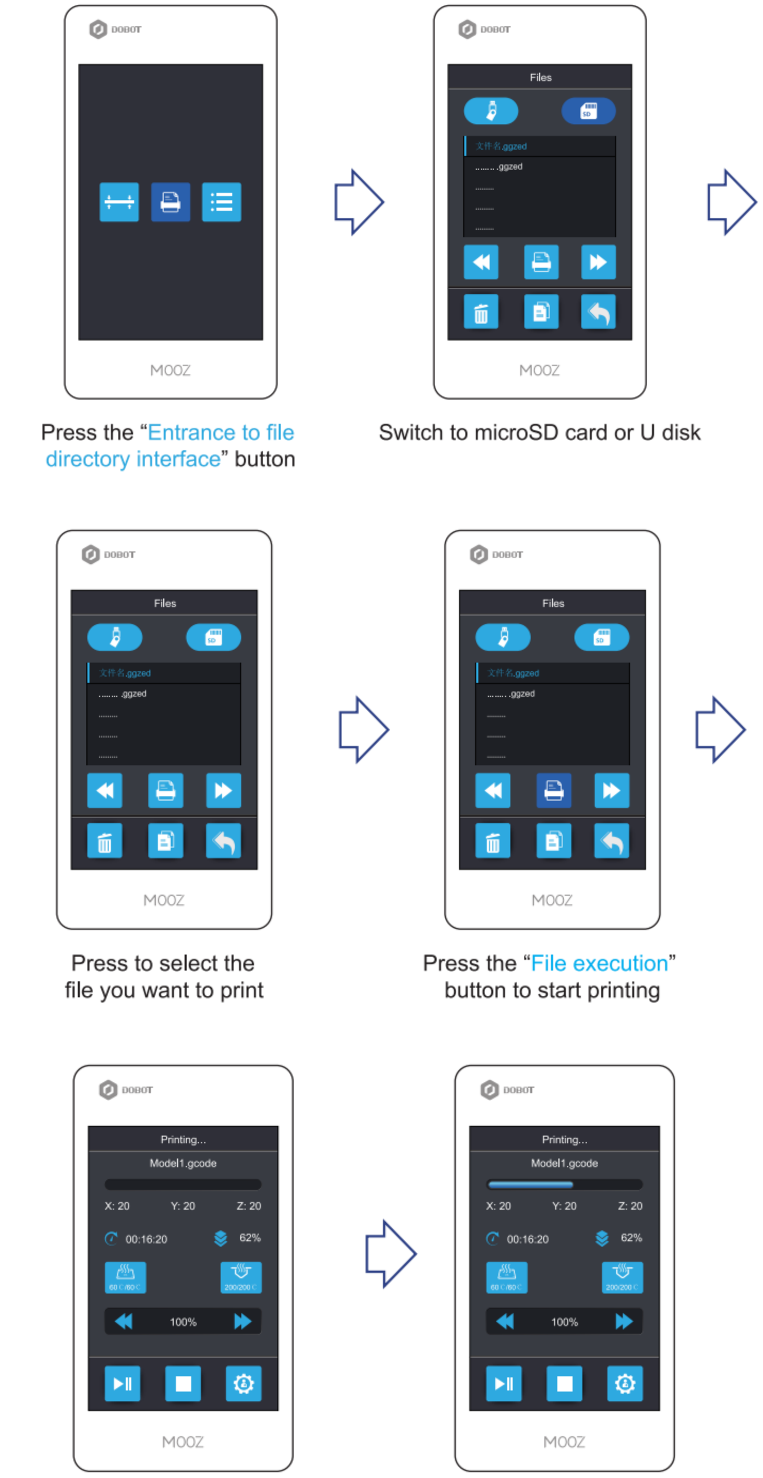


Figure 4.23 Adjusting the model

Step 16: Export the “g-code” file to the USB flash drive.

Step 17: Print the gcode model file in the USB flash drive in the following order using the printer, as shown in Figure 4.24.



Press the “Entrance to file directory interface” button

Switch to microSD card or U disk

Press to select the file you want to print

Press the “File execution” button to start printing

Wait for the heated bed and nozzle to be heated to the target temperature

Start printing automatically

Figure 4.24 Printing the file on the printer

☛ Refer to the design method of the hamburger's upper bun, and design the lower bun model in Tinkercad.

3. Summary

(1) Steps to design the hamburger's upper bun: ________________
____
____

(2) Method of modifying geometry parameters: ________________
____
____

(3) Steps to slice with UltimakerCura software: ________________
____
____

4. Self-Assessment

Check the completed content in the experiment task. Tick (√) the completed items, and circle (○) the uncompleted items.

Assessment Content	Completion Status
I've created a project	
I've modified geometry parameters	
I've designed the hamburger's upper bun model	
I've designed the hamburger's lower bun model	
I've completed the task of slicing	

Task 2: Design Models of Vegetables and Meat Slices

Design models of vegetables and meat slices using Tinkercad and print them with a 3D printer.

1. Analysis

Analyze the steps to design models of vegetables and meat slices. Then, fill in the table below according to prompts and your analysis.

Step 1: Refer to sketches of vegetables and meat slices

Step 2: Select **Scribble** from **Basic Shapes**

Step 3:

2. Steps

Step 1: Log in to your personal account and create a new project.

Step 2: Before designing the models of vegetables and meat slices, find some sketches of hamburgers, vegetables and meat slices, as shown in Figure 4.25.

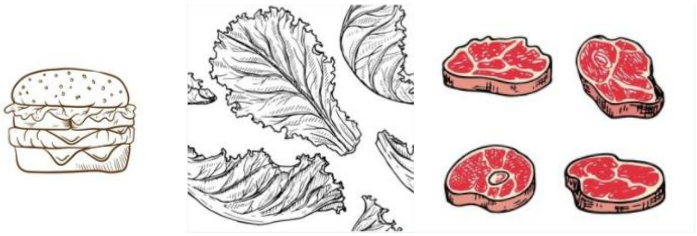


Figure 4.25 Sketches of vegetables and meat slices

Step 3: Start to design models of vegetables and meat slices when you know their general shapes. Select **Scribble** from **Basic Shapes** and drag it to the Workplane, as shown in Figure 4.26.

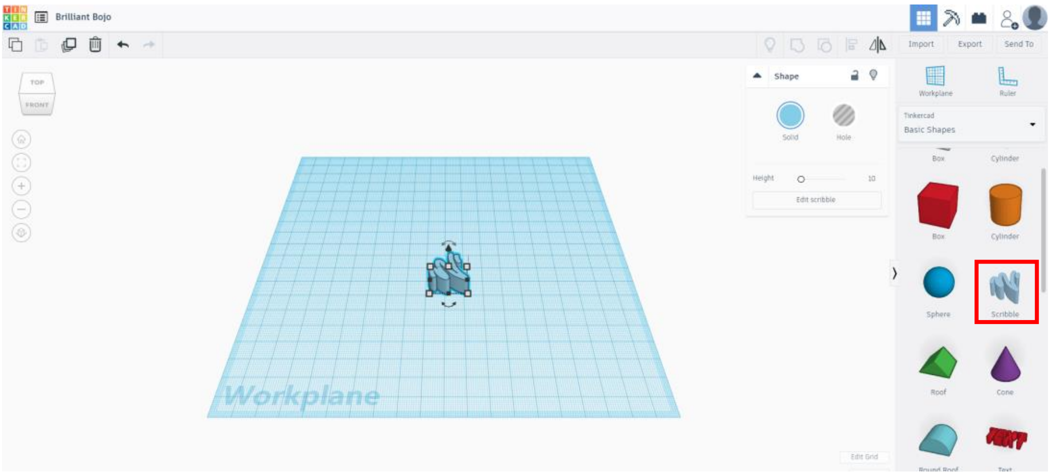


Figure 4.26 Selecting “Scribble**”** from “Basic Shapes”

Step 4: Drag **Scribble** onto the Workplane. A drawing window pops up, as shown in Figure 4.27.

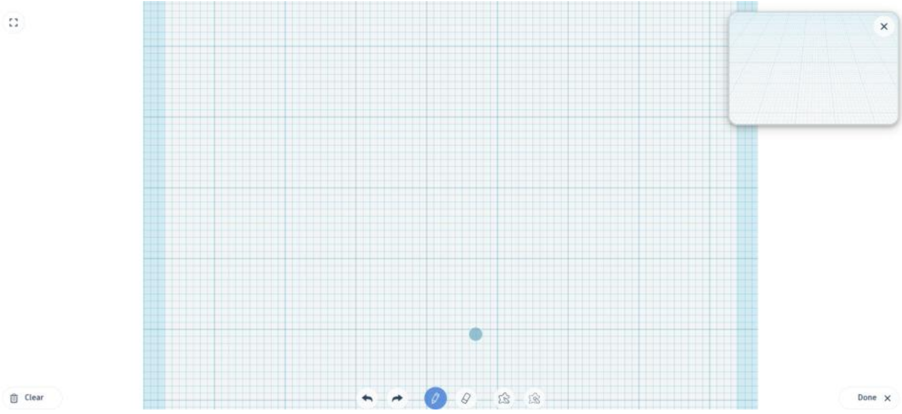


Figure 4.27 Drawing window

Step 5: Find six function buttons at the bottom of the drawing window: **Undo**, **Redo**, **Draw**, **Erase**, **Draw Shape** and **Erase with Shape**. Have **Clear** button in the lower left corner of the window and **Done** in the lower right corner, as shown in Figure 4.28.

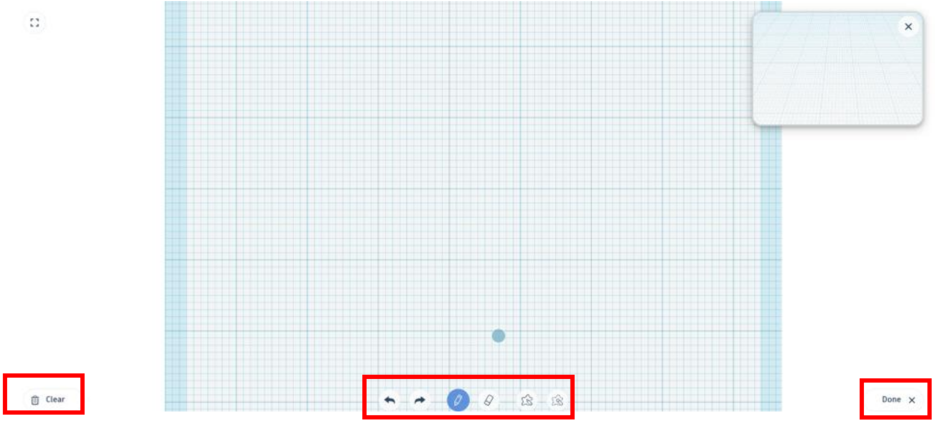


Figure 4.28 Function buttons in the drawing window

Step 6: Click the **Draw** button to outline the vegetable, as shown in Figure 4.29.

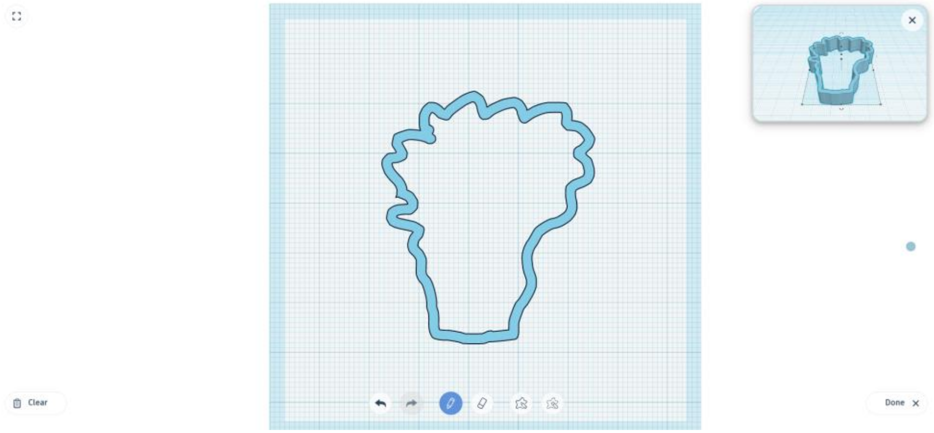


Figure 4.29 Drawing the vegetable

Step 7: Click the **Draw** button to fill the outline, as shown in Figure 4.30.

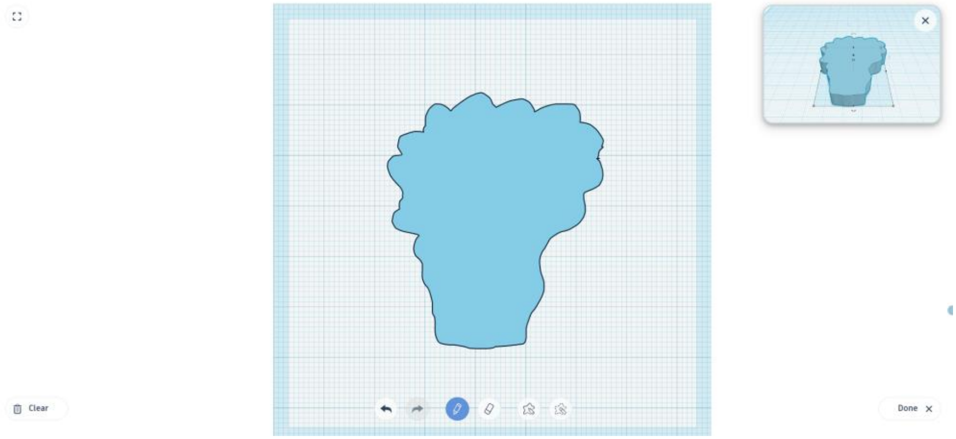


Figure 4.30 Filling the vegetable outline

Step 8: After drawing the vegetable, click **Finish** and fill it with green. The vegetable model is shown in Figure 4.31.

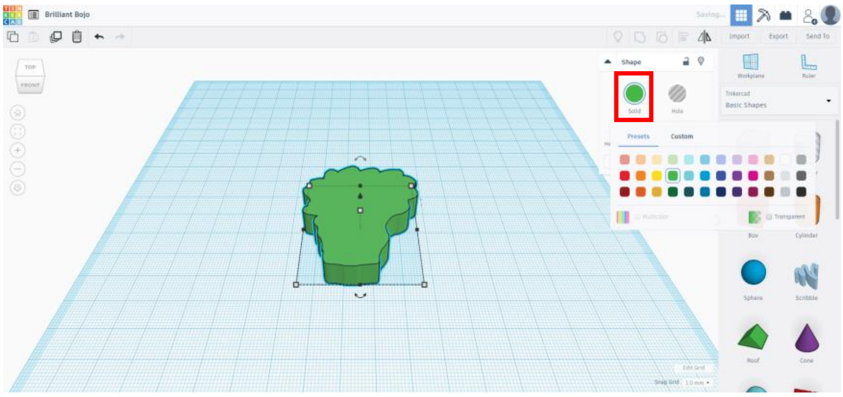


Figure 4.31 Vegetable model

Step 9: Modify the length to 85 mm, the width to 80 mm, and the height to 8 mm, as shown in Figure 4.32.

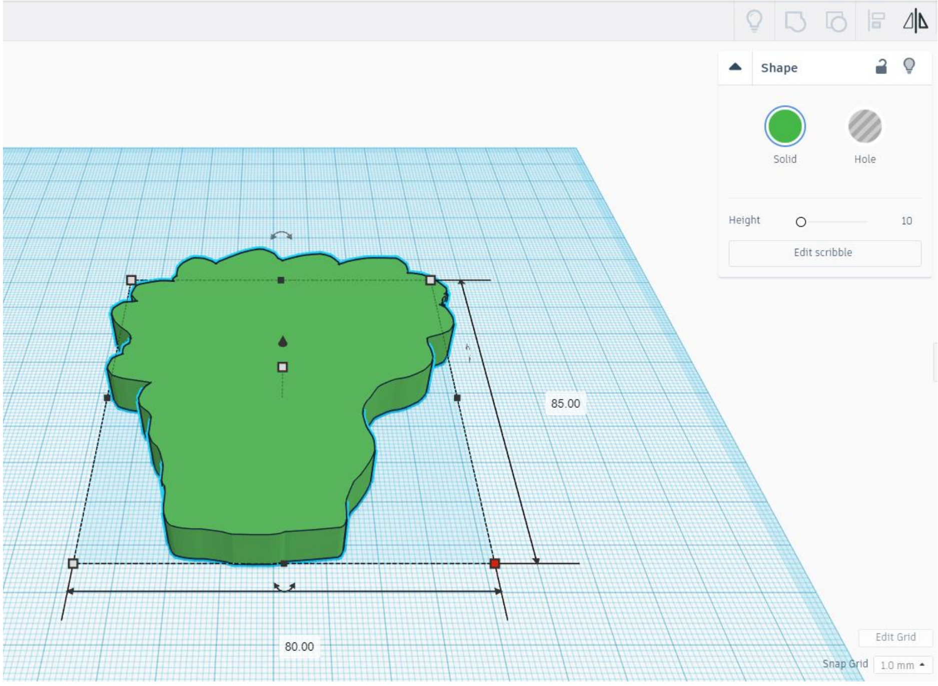


Figure 4.32 Modifying the model size

Step 10: Export the vegetable model and save it in .STL format.

Step 11: Perform slicing. Run Cura3.1.0 and select the printer MOOZ1&2. Refer to Task 1 for detailed steps.

☛ Refer to the meat slice sketches, and design models of chicken and beef slices in Tinkercad.

3. Summary

(1) Method of drawing the geometry:______________________________

____

(2) Steps to design the vegetable model: __________________________

____

____

4. Self-Assessment

Check the completed content in the experiment task. Tick (√) the completed items, and circle (◯) the uncompleted items.

Assessment Content	Completion Status
I’ve created a new project	
I’ve designed the vegetable model	
I’ve designed the chicken slice model	
I’ve designed the beef slice model	

Task 3: Enable the Robotic Arm to Make Delicious Hamburgers

We have prepared the bun, vegetable and meat slice models through Task 1 and Task 2. In Task 3, we will program the robotic arm to make hamburgers automatically.

1. Analysis

Analyze the steps to enable the robotic arm to make hamburgers automatically. Then, fill in the table below according to the prompts and your analysis. .

Step 1: Prepare hardware

Step 2: Design Program: Initialization → Voice acquisition and recognition → If “chicken hamburger” is recognized in the speed → Making a chicken hamburger → ______

(Complete the remaining steps as prompted)

Step 3: Steps to make the hamburger by the robotic arm:

2. Steps

(1) Prepare Hardware

Step 1: Build hardware, as shown in Figure 4.33.

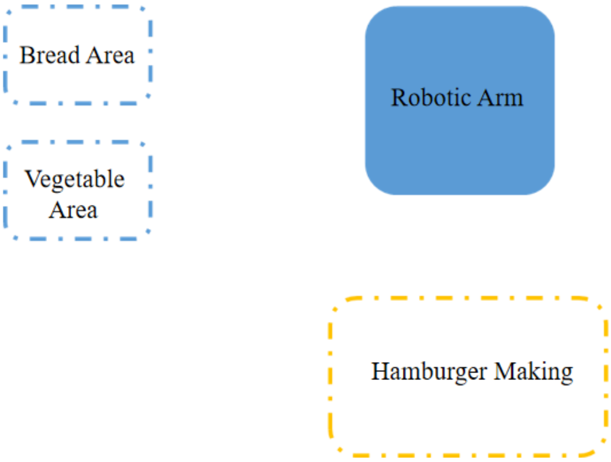


Figure 4.33 building hardware

(2) Design Program

Step 1: Analyze the steps to automatically make a hamburger by the robotic arm, and draw the flow chart, as shown in Figure 4.34.

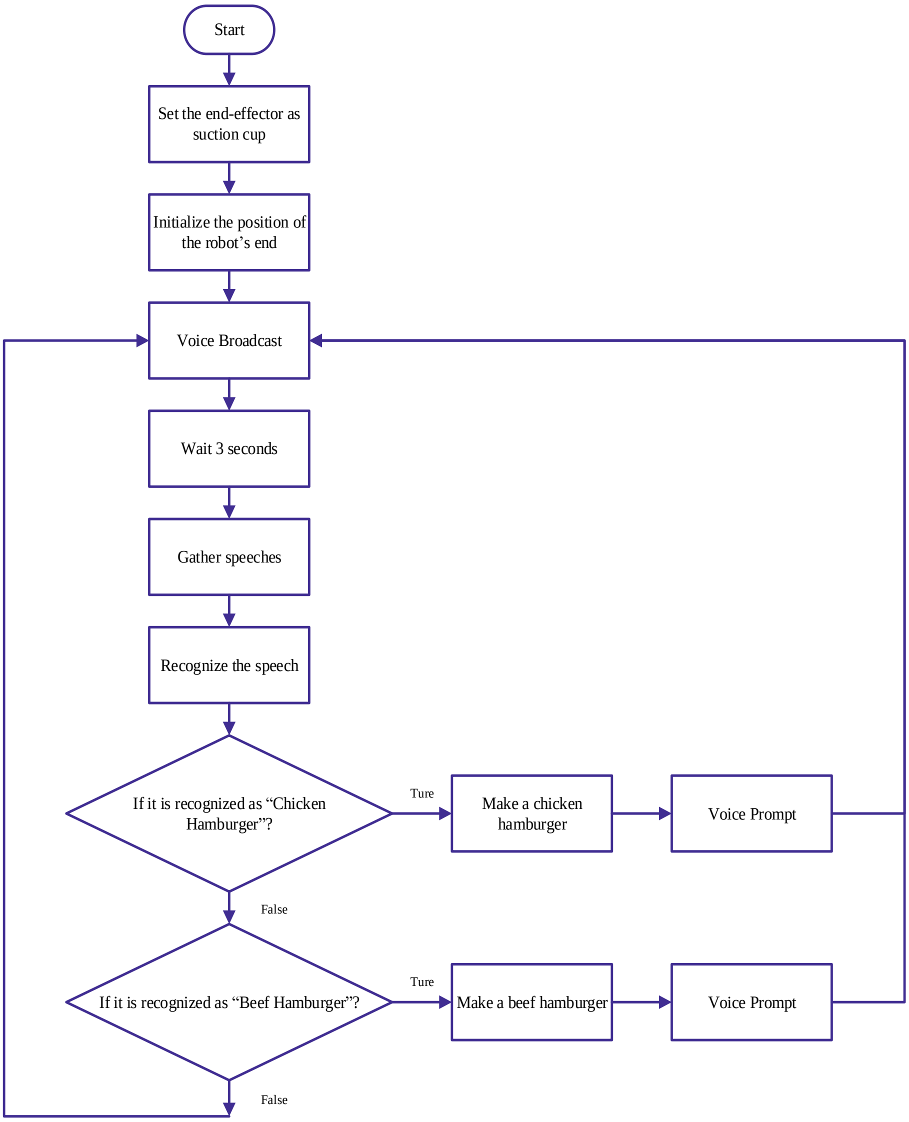
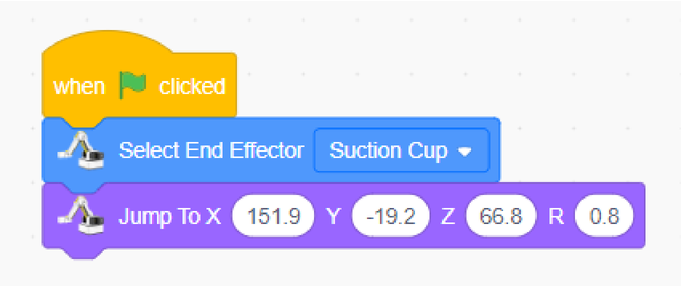

Start
Set the end-effector as suction cup
Initialize the position of the robot's end
Voice Broadcast
Wait 3 seconds
Gather speeches
Recognize the speech
If it is recognized as "Chicken Hamburger"?
Ture
Make a chicken hamburger
Voice Prompt
False
If it is recognized as "Beef Hamburger"?
Ture
Make a beef hamburger
Voice Prompt
False

Figure 4.34 Flow chart

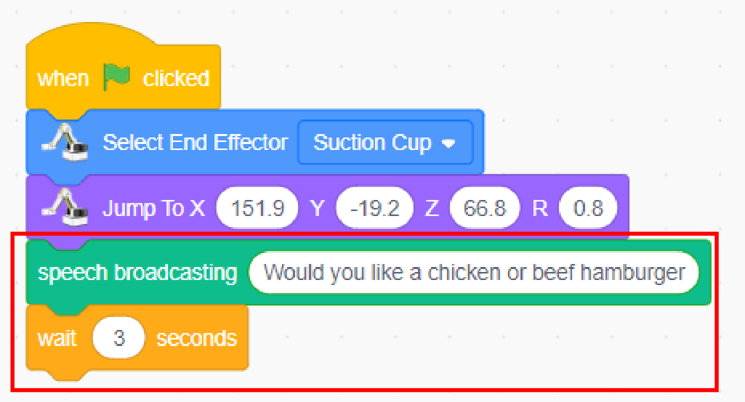
Step 2: Set the end-of-arm tool "Suction Cup", and initialize the end position of the arm, as shown in Figure 4.35.



```
when 🏴 clicked
Select End Effector Suction Cup ▾
Jump To X 151.9 Y -19.2 Z 66.8 R 0.8
```

Figure 4.35 Initializing the end-of-arm tool and position

Step 3: Broadcast the speech message "Would you like a chicken or beef hamburger" and this ends after 3 seconds, as shown in Figure 4.36.

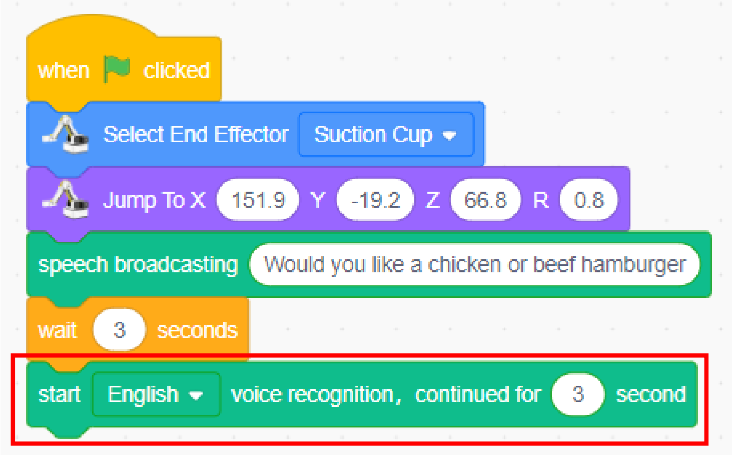


```
when [flag] clicked
Select End Effector [Suction Cup ▾]
Jump To X (151.9) Y (-19.2) Z (66.8) R (0.8)
speech broadcasting (Would you like a chicken or beef hamburger)
wait (3) seconds
```

Figure 4.36 Broadcasting the speech message

Why is “Wait 3 seconds” required?

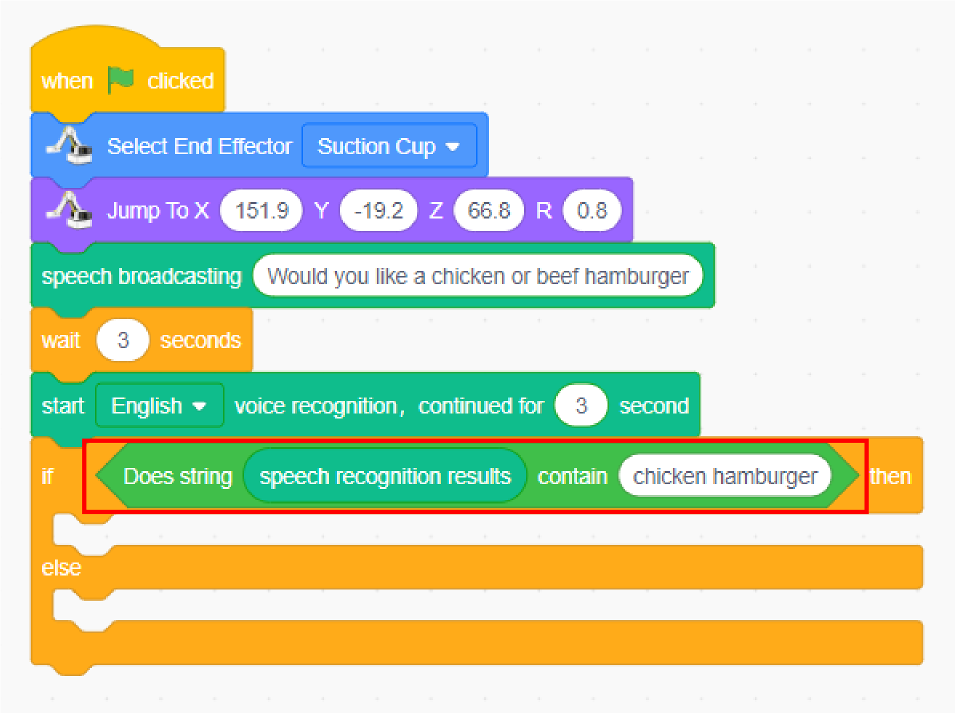
Step 4: Click the **AI** tab, drag the “start English voice recognition, continued for 1 second” block, and change the duration to 3 seconds to collect and recognize speech, as shown in Figure 4.37.



```
when [flag] clicked
Select End Effector [Suction Cup ▾]
Jump To X (151.9) Y (-19.2) Z (66.8) R (0.8)
speech broadcasting (Would you like a chicken or beef hamburger)
wait (3) seconds
start [English ▾] voice recognition, continued for (3) second
```

Figure 4.37 Speech recognition

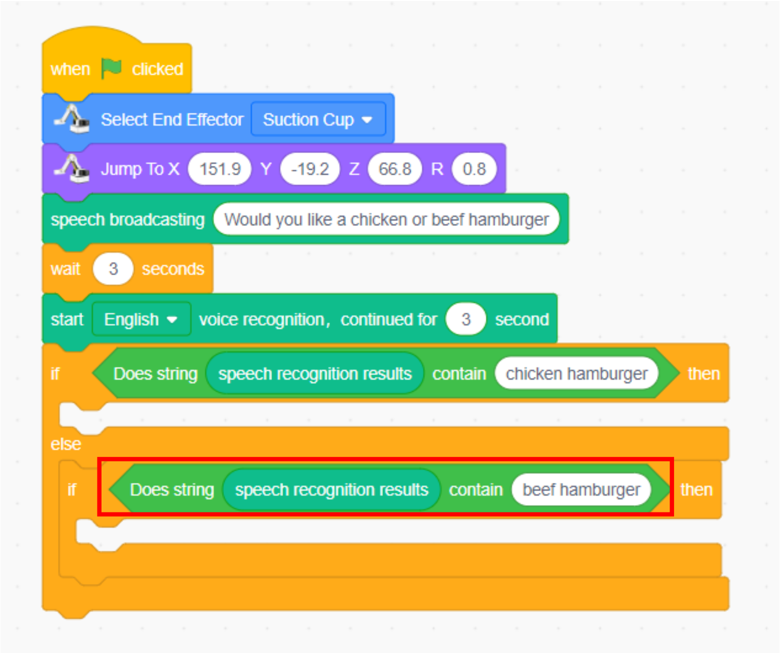
Step 5: Set the judgment condition. Judge if the speech recognition result contains the keywords “chicken hamburger”, as shown in Figure 4.38.



```
when ⚑ clicked
Select End Effector  Suction Cup ▾
Jump To X 151.9 Y -19.2 Z 66.8 R 0.8
speech broadcasting  Would you like a chicken or beef hamburger
wait 3 seconds
start English ▾ voice recognition, continued for 3 second
if  Does string  speech recognition results  contain  chicken hamburger  then

else

```

Figure 4.38 Setting the judgment condition

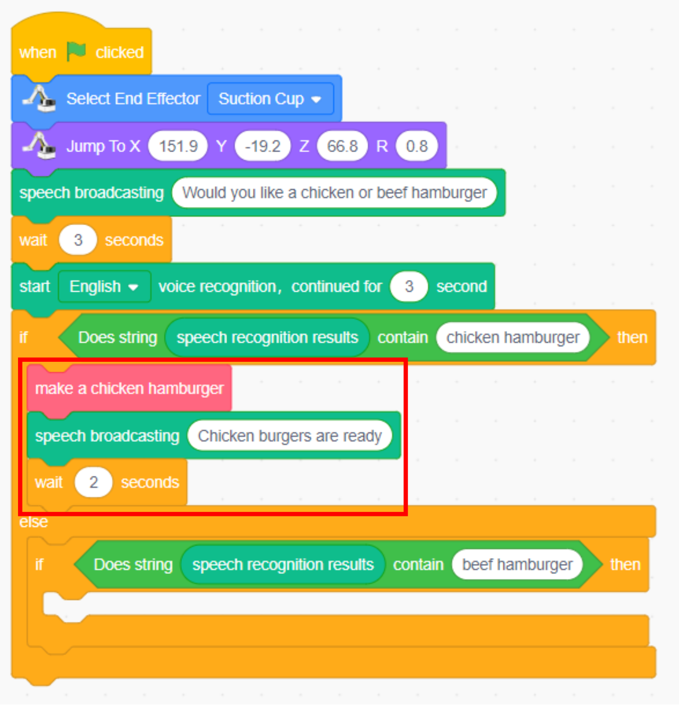
Step 6: Set the judgment condition for the second case. Judge if the speech recognition result contains the keywords “beef hamburger”, as shown in Figure 4.39.



```
when ⚑ clicked
Select End Effector  Suction Cup ▾
Jump To X 151.9 Y -19.2 Z 66.8 R 0.8
speech broadcasting  Would you like a chicken or beef hamburger
wait 3 seconds
start English ▾ voice recognition, continued for 3 second
if  Does string  speech recognition results  contain  chicken hamburger  then

else
    if  Does string  speech recognition results  contain  beef hamburger  then

```

Figure 4.39 Setting the judgment condition for the second case

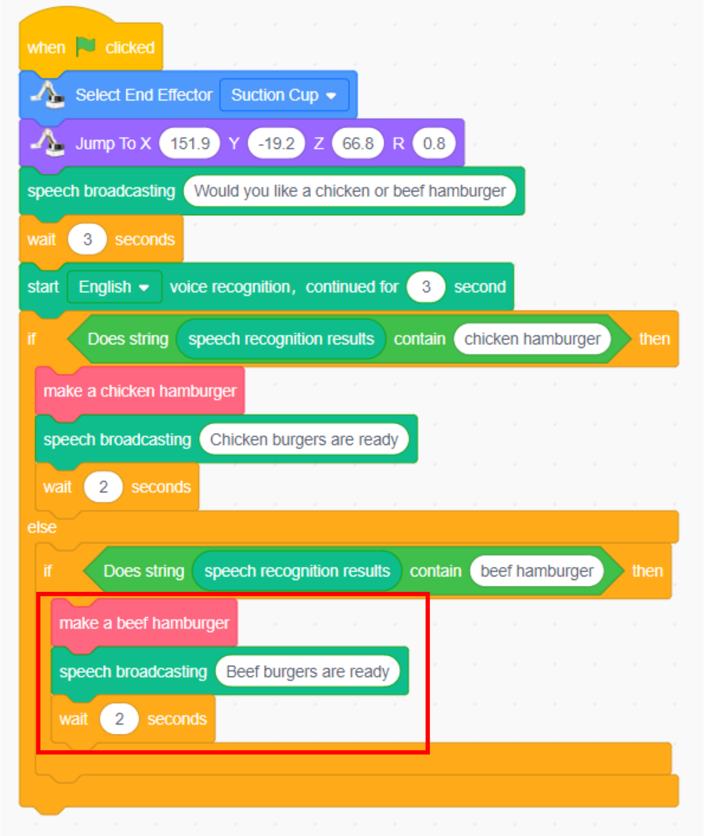
Step 7: Start to “make a chicken hamburger” by the robotic arm once the speech recognition result contains the keywords “chicken hamburger”, as shown in Figure 4.40.



```
when 🏳 clicked
Select End Effector [Suction Cup ▾]
Jump To X (151.9) Y (-19.2) Z (66.8) R (0.8)
speech broadcasting (Would you like a chicken or beef hamburger)
wait (3) seconds
start [English ▾] voice recognition, continued for (3) second
if <Does string (speech recognition results) contain (chicken hamburger)> then
    make a chicken hamburger
    speech broadcasting (Chicken burgers are ready)
    wait (2) seconds
else
    if <Does string (speech recognition results) contain (beef hamburger)> then
```

Figure 4.40 Starting to make a chicken hamburger by the robotic arm

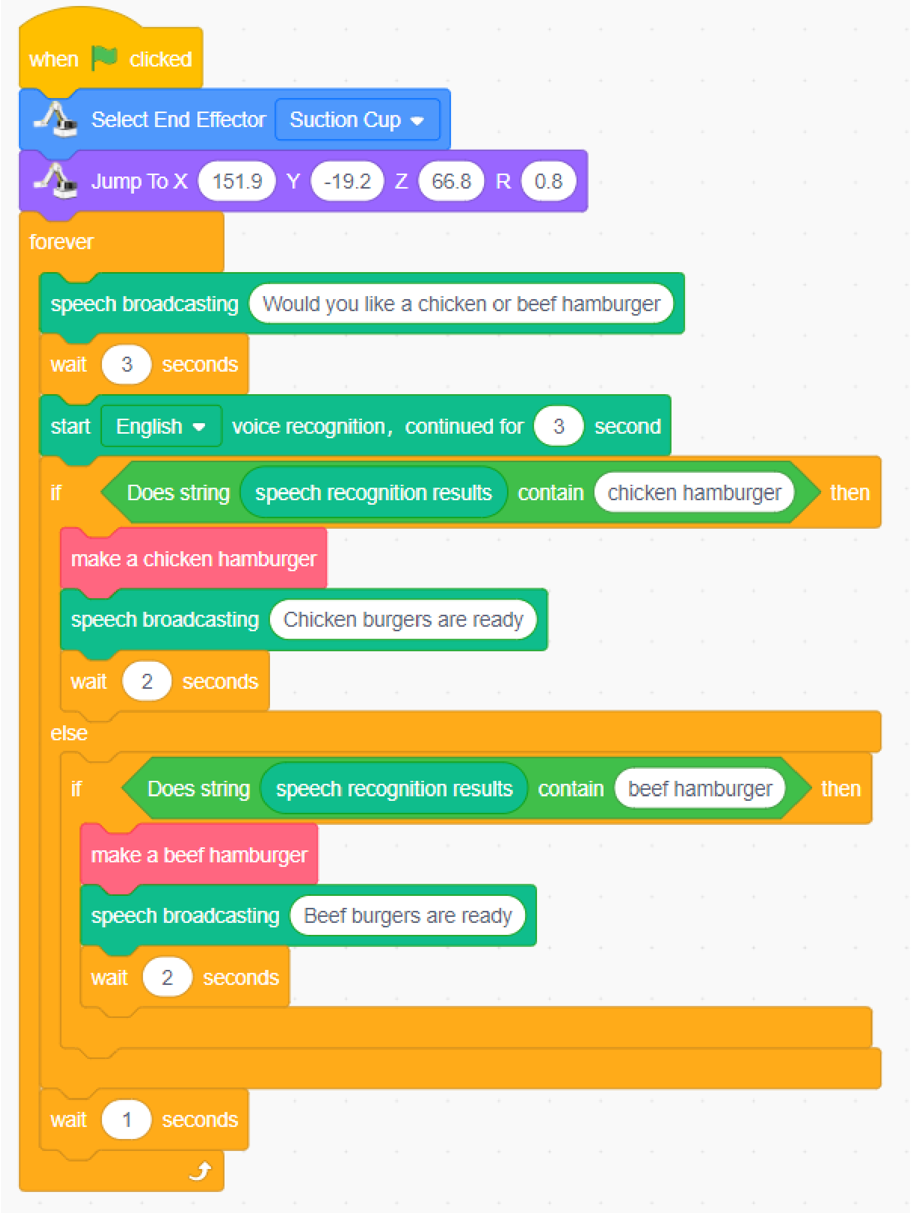
Step 8: Start to “make a beef hamburger” by the robotic arm once the speech recognition result contains the keywords “beef hamburger”, as shown in Figure 4.41.



```
when 🏳 clicked
Select End Effector [Suction Cup ▾]
Jump To X (151.9) Y (-19.2) Z (66.8) R (0.8)
speech broadcasting (Would you like a chicken or beef hamburger)
wait (3) seconds
start [English ▾] voice recognition, continued for (3) second
if <Does string (speech recognition results) contain (chicken hamburger)> then
    make a chicken hamburger
    speech broadcasting (Chicken burgers are ready)
    wait (2) seconds
else
    if <Does string (speech recognition results) contain (beef hamburger)> then
        make a beef hamburger
        speech broadcasting (Beef burgers are ready)
        wait (2) seconds
```

Figure 4.41 Starting to make a beef hamburger by the robotic arm

Step 9: Add the “forever” block, as shown in Figure 4.42



```
when ⚑ clicked
Select End Effector [Suction Cup ▾]
Jump To X (151.9) Y (-19.2) Z (66.8) R (0.8)
forever
    speech broadcasting (Would you like a chicken or beef hamburger)
    wait (3) seconds
    start [English ▾] voice recognition, continued for (3) second
    if <Does string (speech recognition results) contain (chicken hamburger)> then
        make a chicken hamburger
        speech broadcasting (Chicken burgers are ready)
        wait (2) seconds
    else
        if <Does string (speech recognition results) contain (beef hamburger)> then
            make a beef hamburger
            speech broadcasting (Beef burgers are ready)
            wait (2) seconds
    wait (1) seconds
```

Figure 4.42 Repeating the execution

The program uses two self-made blocks, namely "make a chicken hamburger" and "make a beef hamburger". By analyzing the robotic arm's operations for making different hamburgers, try to define the two blocks, and program the robotic arm to make chicken and beef hamburgers.

3. Summary

(1) Summarize the use of self-made blocks:____________________

____

(2) Summarize the use of the judgment structure: _______________

____

____

____

4. Self-Assessment

Check the completed content in the experiment task. Tick (√) the completed items, and circle (◯) the uncompleted items.

Assessment Content	Completion Status
I've built the hardware	
I've completed the task of speech acquisition and recognition	
I've made the new "make a chicken hamburger" block	
I've made the new "make a beef hamburger" block	
I've used the hamburger making program with the nested branch structure	

Experiment 5

AI Waiter

Team Name: Team member: Date:

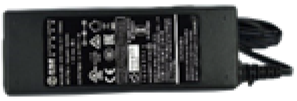
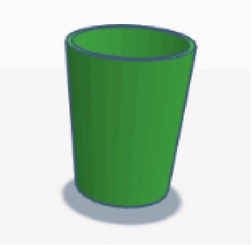
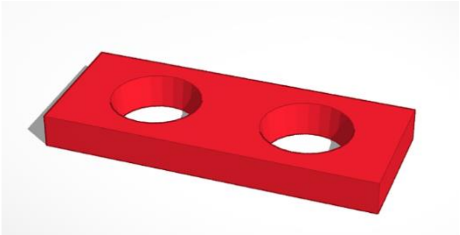
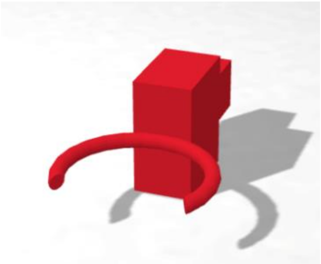
Overview

Smart speakers, handset smart assistants and so on have become part of our lives. But what is the speech recognition technology? How do we use it? In this experiment, through simulating the building of a smart voice bar counter, we aim to help students: experience graphical programming-based modeling like cup modeling and grasp the Tinkercad-based modeling flow; build the saucer model through conventional means and programming, and explore the relations between modeling and graphical programming; design the end of the cup clamp and explore diverse ends of the robotic arm; learn and use the speech recognition technology, and explore its application.

Objective

1. Understand Tinkercad's shortcut key by copying and pasting a model.
2. Master Tinkercad's alignment function by designing the saucer model.
3. Skillfully build the model through Tinkercad.
4. Understand the use of graphical programming by building the modeling.
5. By building the saucer model and conventional modes and programming, understand the differences and similarities between the two modeling modes.
6. Explore diverse ends of the robotic arms by designing the end of the cup clamp.
7. Understand the graphical programming-based modeling by programming the building of the cup clamp model.
8. Understand the speech recognition technology by designing the bar counter assistant.

Equipment

Equipment Picture	Name	Quantity
	Dobot Magician Lite Robotic arm	1
	USB Type-C interface cable	1
	Power adapter	1
	Camera	1
	Creatively designed cup	2
	Saucer	1
	Cup clamp	1

Equipment Picture	Name	Quantity
	MOOZ-2 Plus	1
	PLA	4

Requirements

- Take care when using electricity.
- Before the experiment, check whether the experimental equipment is complete and intact. If there is any omission or damage, please report to the teacher.
- Any specific operations in the experiment shall be performed according to the experiment manual. If you have any questions, please promptly ask the teacher.
- During the experiment, the joints will start to work as the robotic arms are powered on. In that case, do not move the joints of the robotic arms hard if you do not press the unlock key.

Task 1: Design the Cup

To build a smart bar counter, prepare: a drink cup, and a saucer. This experiment is to design a cup model. Using Tinkercad, you can creatively design a personalized cup model.

1. Analysis

Design a cup model through Tinkercad, and print the model through 3D printing technology. Observe the model, analyze the model design steps, and complete the realization steps in the table below.

Step 1: Look for a proper cup model for reference, and analyze the shape of the reference model.

Step 2: Measure the size of the reference model, and take them as the dimension reference data of the model.

Step 3: Log in to the Tinkercad account, and create a work.

Step 4: Design the cup model as per the reference model.

Step 5: Print the model.

2. Steps

Step 1: Refer to a model. Look for a proper cup as a reference model, as shown in Figure 5.1.

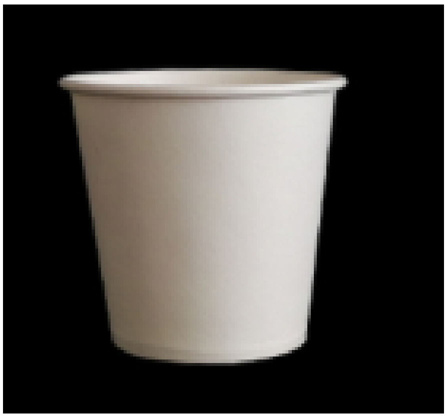


Figure 5.1 Paper cup as a reference model

Step 2: Measure the size. Measure the size of the paper cup, as shown in Figure 5.2.

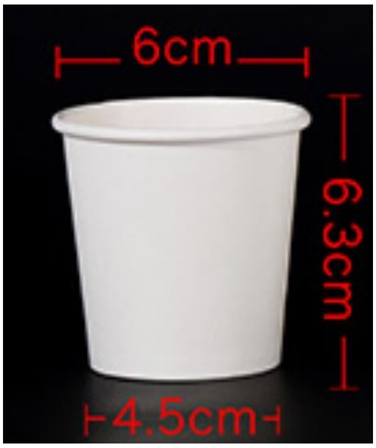


Figure 5.2 Size of the paper cup

Why do we measure the reference model?

Step 3: Log in to the Tinkercad account, create a project, and name it cup, as shown in Figure 5.3.

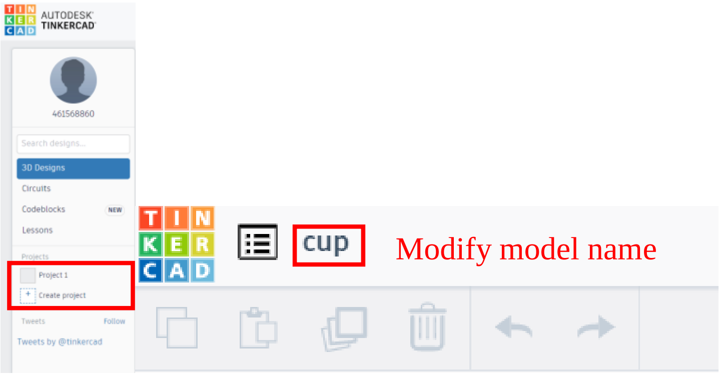
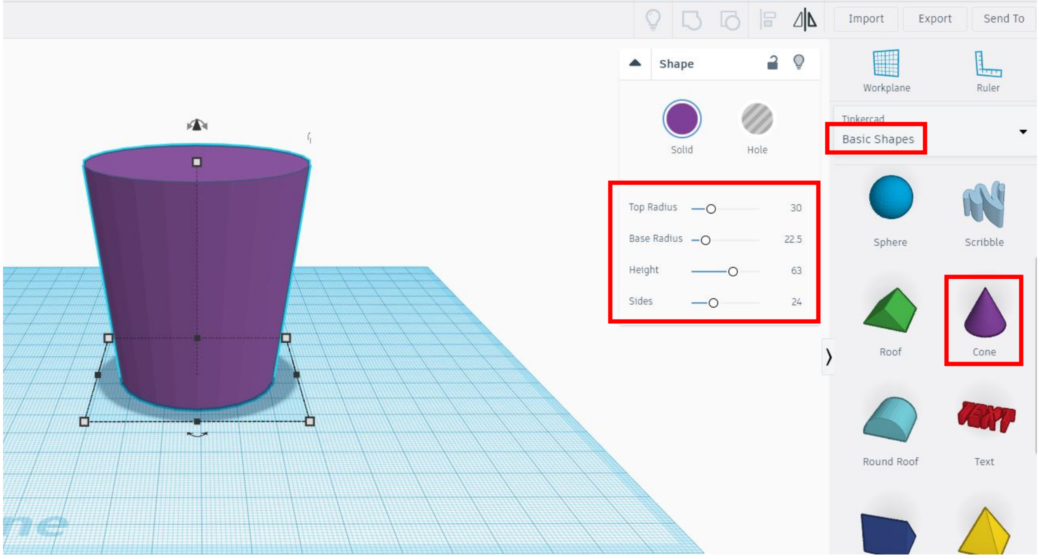


Figure 5.3 Creating the cup model project

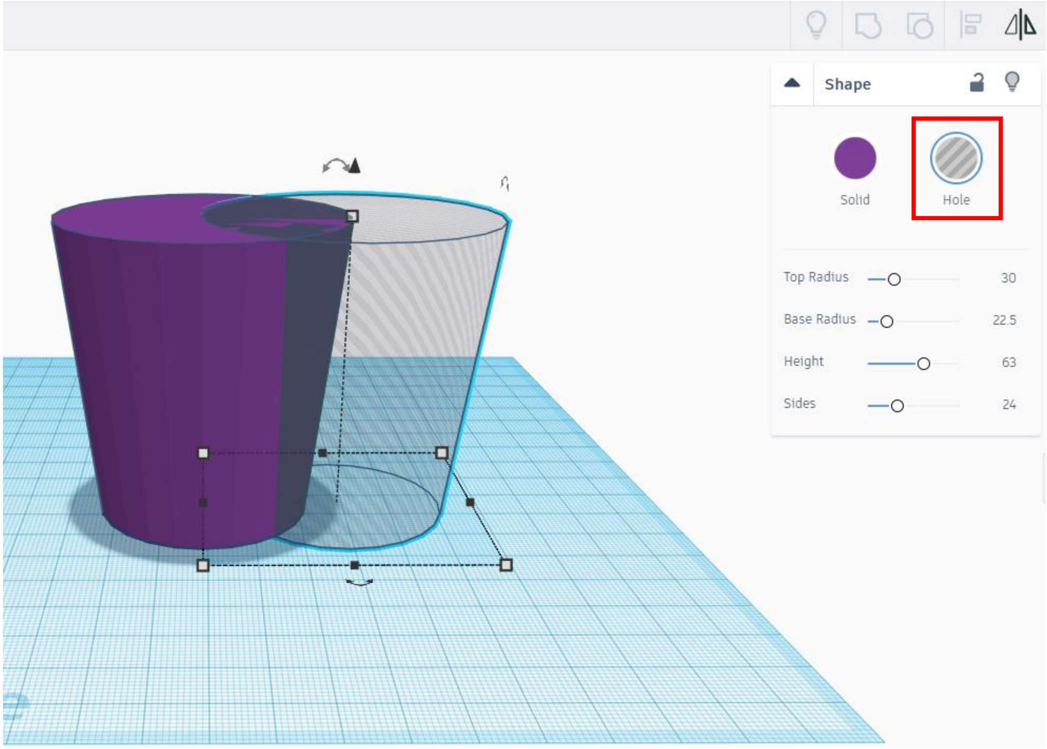
Step 4: Observe the cup reference model, and we find that we can design a paper cup to have the shape of a truncated cone with a large top and a small bottom. Based on the measured size, set the truncated cone to 30 mm in top radius, 22.5 mm in base radius and 63 mm in height, as shown in Figure 5.4.



Top Radius 30
Base Radius 22.5
Height 63
Sides 24

Figure 5.4 Truncated cone

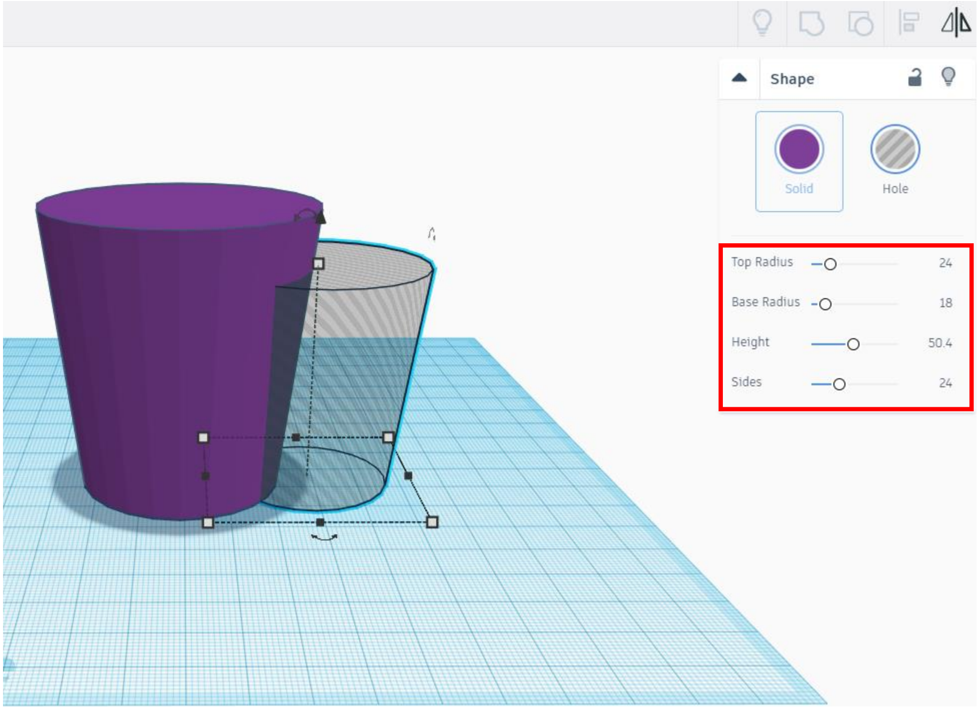
Step 5: Hollow the truncated cone because a cup is hollow. Use the shortcut keys **ctrl+c** and **ctrl+v** to copy and paste the original truncated cone respectively. Set the material of the copied and pasted truncate cone to hole as that of the hollowed truncated cone, as shown in Figure 5.5.



Top Radius 30
Base Radius 22.5
Height 63
Sides 24

Figure 5.5 Copying and pasting the truncated cone

Step 6: Reduce the copied truncated cone by 0.8 times. As a result, the top radius is 24mm, the base radius is 18 mm and the height is 50.4 mm, as shown in Figure 5.6.



Top Radius 24
Base Radius 18
Height 50.4
Sides 24

Figure 5.6 Reducing the truncated cone by 0.8 times

Step 7: Align. Align the truncated cone as the hole with the original truncated cone, press the **shift** key to select the two truncated cones,

alignment icon

click the alignment icon, and select three points to align, as shown in Figure 5.7.

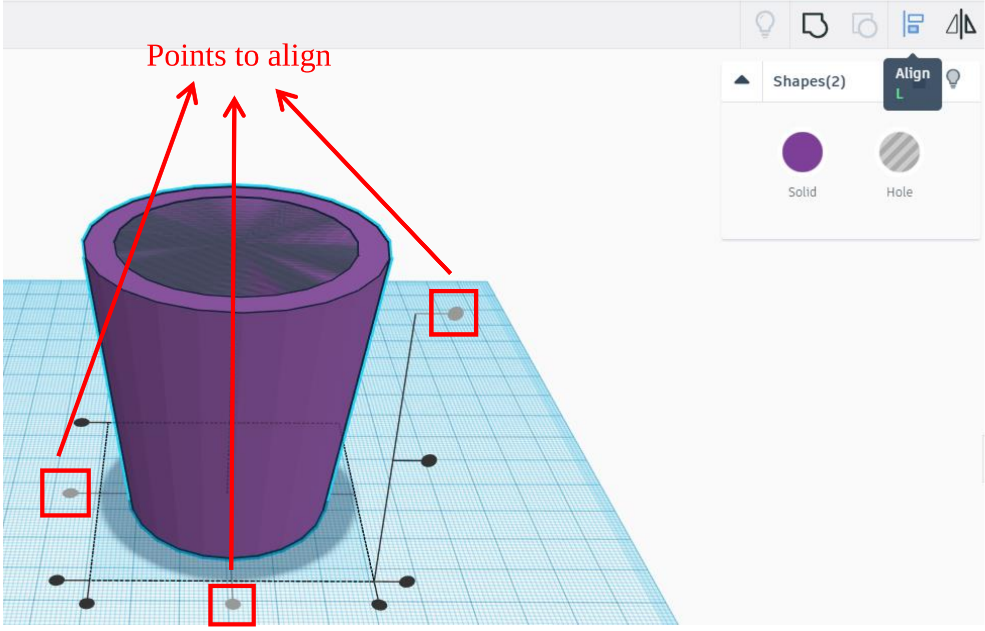
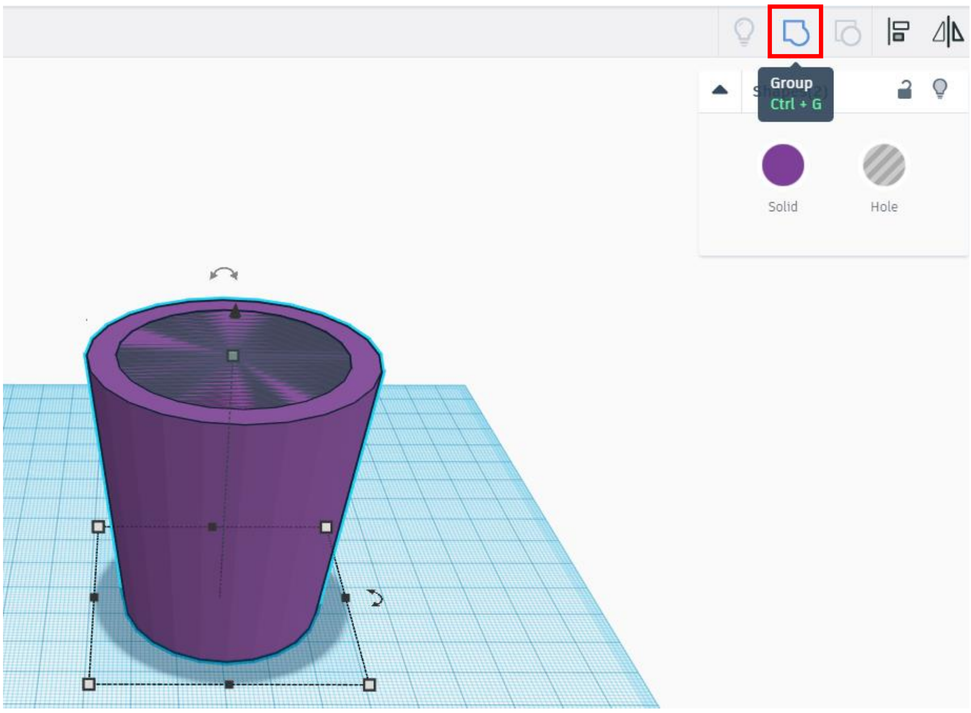


Figure 5.7 Alignment

Step 8: Combine the objects. Combine the original truncated cone with the one as the hole, and hollow the water cup, as shown in Figure 5.8.



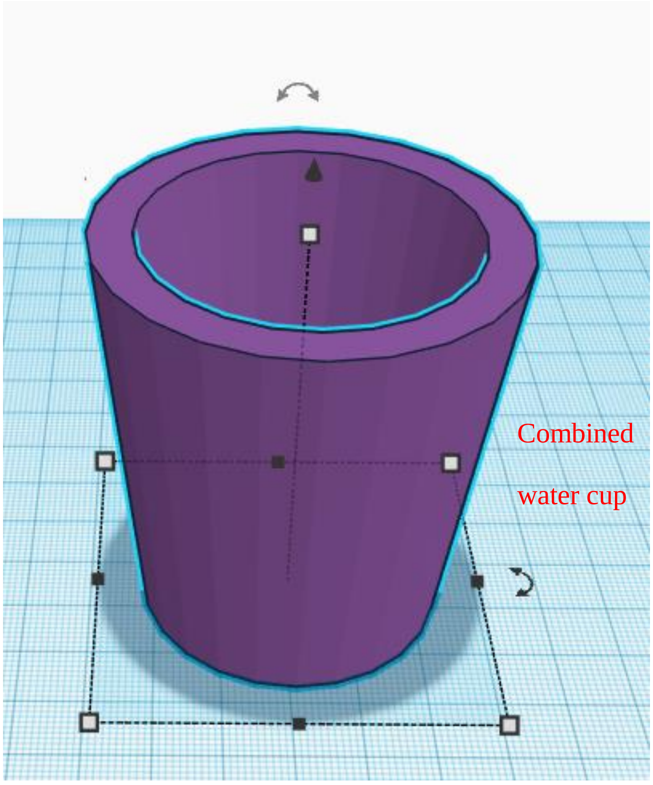


Figure 5.8 Combination

Analyze the steps to build the cup model in the flow chart way.

Step 9: Export the model. Export the cup model in the "stl" format.

Step 10: Slice the cup. Do so with Cura software, convert the "stl" file into the "g-code" file, and copy the latter to the printer for printing.

3. Summary

(1) The method of setting the model parameter: ________________
____

(2) The uses of copying and pasting: ________________________
____

(3) The method of aligning the model: _______________________
____

(4) The steps to build the cup model: ________________________
____

4. Self-Assessment

Check the completed content in the experiment task. Tick (√) the completed items, and circle (◯) the uncompleted items.

Assessment Content	Completion Status
I've known the shape of the cup	
I've known the cone and the truncated cone	
I've have designed the cup model	

Task 2: Design the Saucer Model

It is not easy to fix the cup on the bar counter. We easily knock it over, spilling its drink. We can design a saucer model to determine and fix the cup position, making it hard to knock over.

1. Analysis

Use Tinkercad to design a saucer model and print it through 3D printing technology. Observe the model, analyze the modeling design steps, and complete the realization steps in the blank in the table below.

Step 1: Analyze the shape and size of a saucer, for example, a saucer enough for two cups.

Step 2: Provide a cup model, and set the model to material.

Step 3: Amplify the cup model.

Step 4: Take____ as the saucer body.

Step 5: ____ the cup and the saucer.

Step 6: Group the cup and the saucer, and allow the cup to punch the saucer.

Step 7: Save and print the model.

2. Steps

Step 1: As per the cup size, analyze the saucer size, as shown in Figure 5.9.

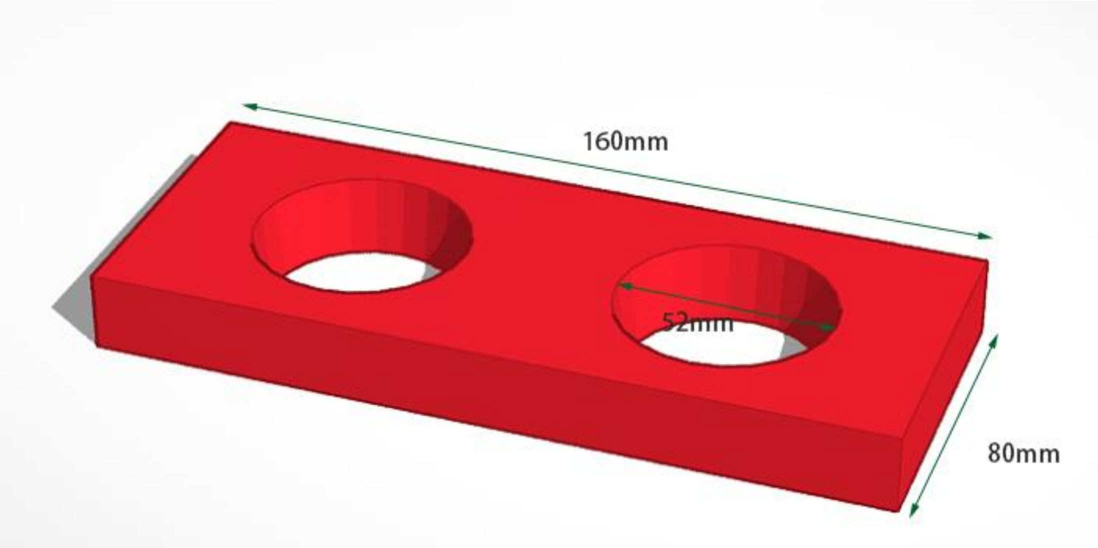
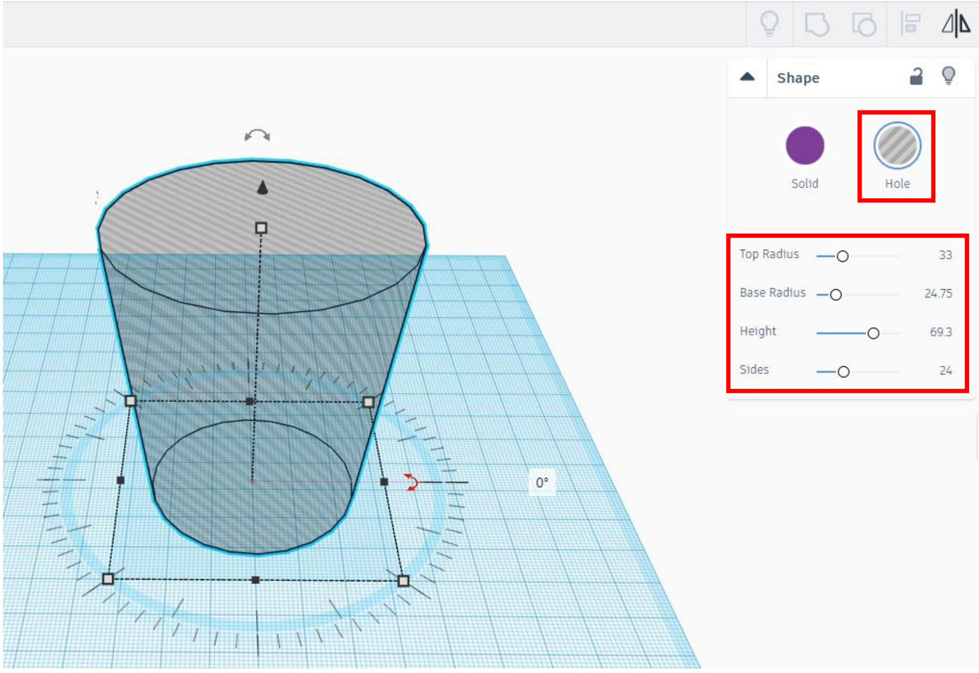


Figure 5.9 Saucer size

💡 A saucer is to bear a cup, so its hole size must be greater than a cup size. We can amplify the cup by 1.1 times as the hole of the saucer. In that case, the cup must be solid.

Step 2: Punch the truncated cone with the saucer, select the truncated cone, which is 1.1 times as large as the above cup model. The truncated cone is 33 mm in top radius, 24.75 mm in base radius and 69.3 mm in height, as shown in Figure 5.10.



Shape

Solid Hole

Top Radius 33

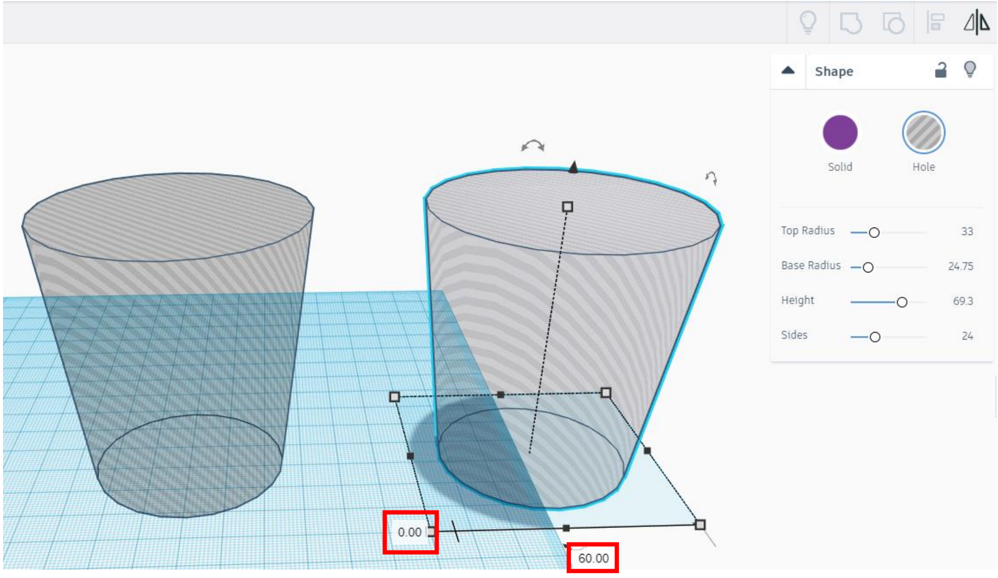
Base Radius 24.75

Height 69.3

Sides 24

Figure 5.10 Punching the truncated cone

Step 3: Use the shortcut keys **ctrl+c** and **ctrl+v** to copy and paste the truncated cone respectively. The horizontal distance between the copied truncated cone and the original truncated cone is 60 mm, as shown in Figure 5.11.



Shape

Solid Hole

Top Radius 33

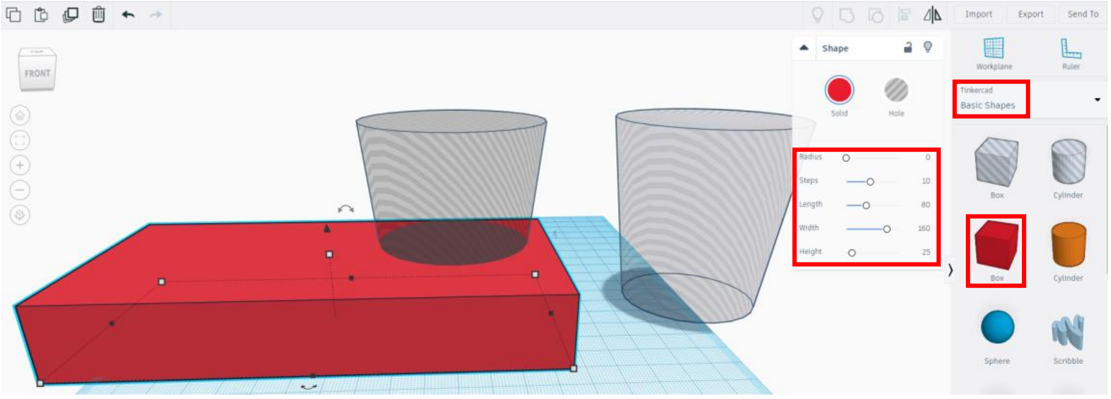
Base Radius 24.75

Height 69.3

Sides 24

Figure 5.11 Copying and pasting the truncated cone

Step 4: Drag out a block with 80 mm in length, 160 mm in width and 25 mm in height, as shown in Figure 5.12.



Radius 0
Steps 10
Length 80
Width 160
Height 25

Figure 5.12 Block

Step 5: Group the truncated cones. Press the **shift** key to select the two truncated cones, and click the group icon to become a group, as shown in Figure 5.13.

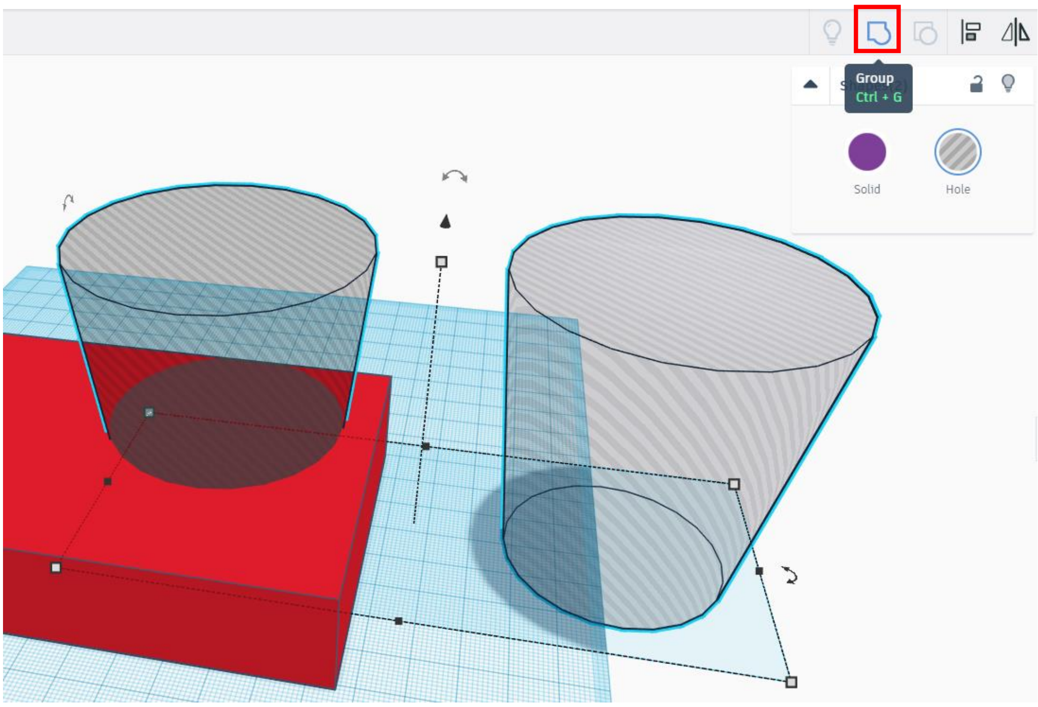


Figure 5.13 Grouping the truncated cones

Step 6: Align the related points. Put the block between the two truncated cones, press the **shift** key to select the block and the two cones, click the alignment icon, and select the points to align as shown in Figure 5.14.

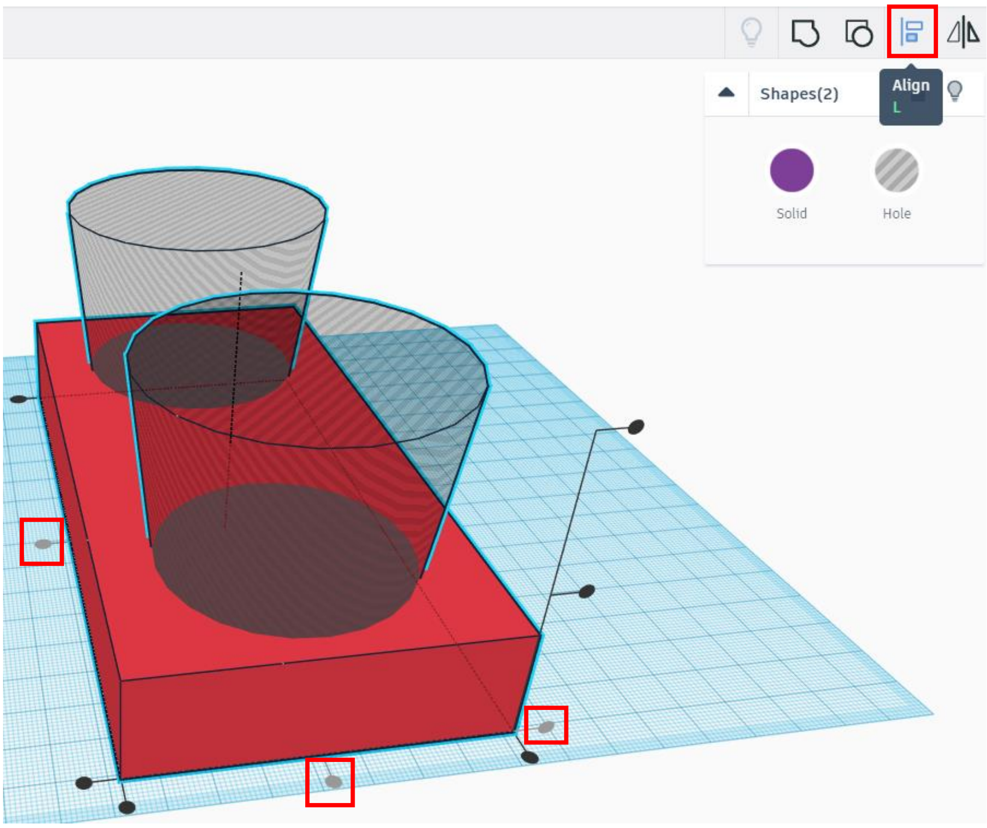
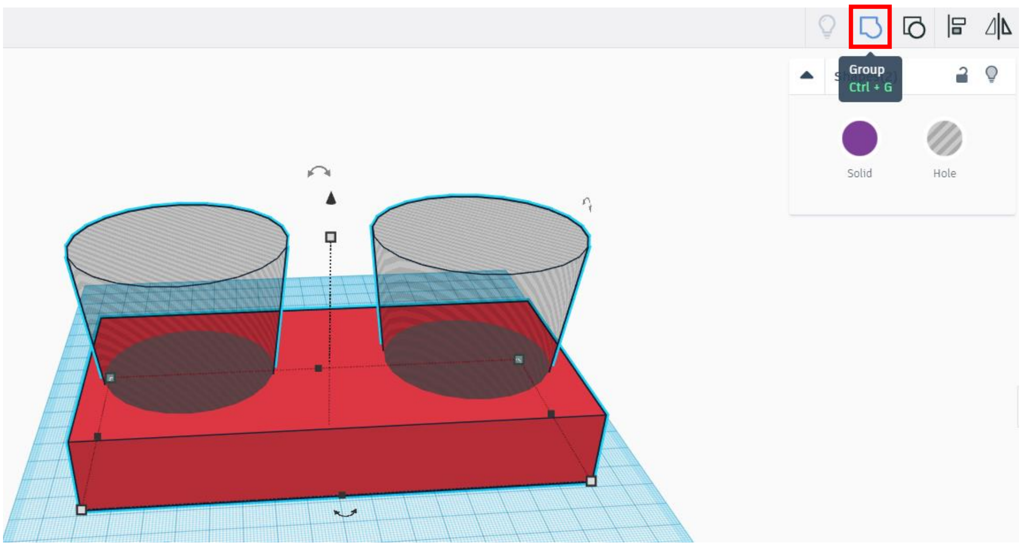


Figure 5.14 Alignment

Step 7: Punch the block, and combine the block with the two truncated cones, as shown in Figure 5.15.



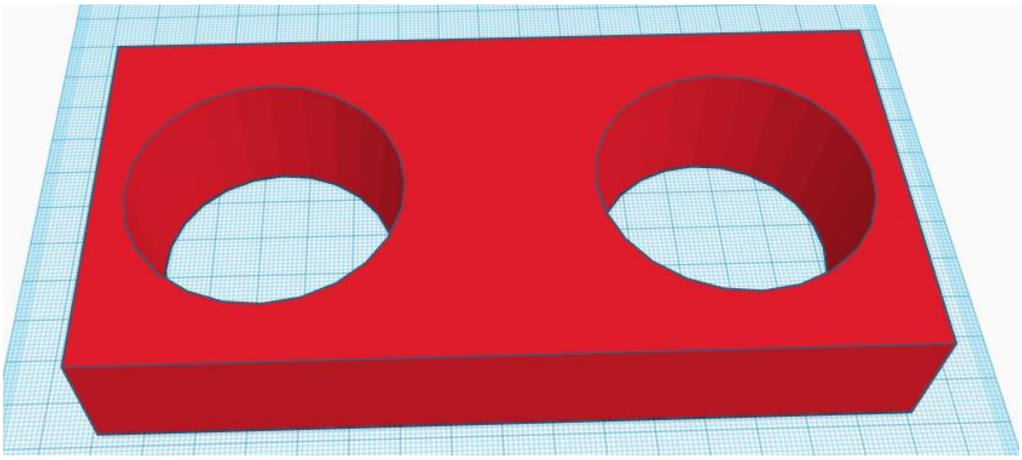


Figure 5.15 Punching the block

☛ Try summarizing the saucer modeling process through a flow chart.

3. Summary

(1) The reason for amplifying the cup model: ____________________

____

(2) The saucer modeling process: ______________________________

____

4. Self-Assessment

Check the completed content in the experiment task. Tick (√) the completed items, and circle (○) the uncompleted items.

Assessment Content	Completion Status
I've used the alignment function	
I've used the grouping function	
I've known the process of the saucer model design	
I've designed the saucer model	

Task 3: Program the Saucer Model

Can 3D modeling work with programming? Can we design a model through programming? This experiment gives a positive answer. We can do so. Let us try designing the saucer model in task 2 with the programming method.

1. Analysis

As per the steps to design the saucer model in task 2, analyze the steps to program the saucer model.

Step 1:____

Step 2:____

Step 3: ____

Step 4: ____

Step 5: ____

Step 6: ____

Step 7: ____

2. Steps

Step 1: As per the experiment analysis result, read the flow chart of programming-based saucer modeling design, as shown in Figure 5.16.

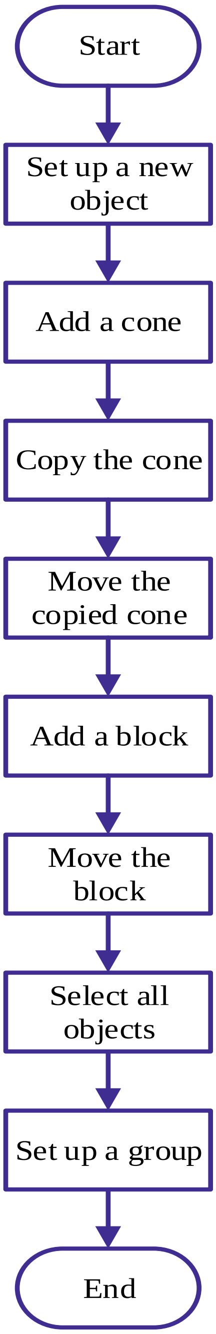


Figure 5.16 Flow chart of saucer modeling

Step 2: Switch to the programming mode, select **Codeblocks**, and click **Create new Codeblock**, as shown in Figure 5.17.

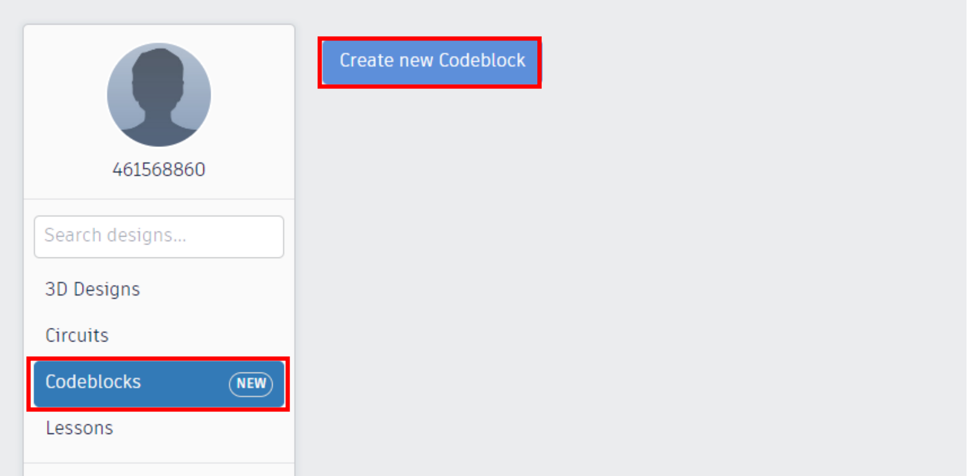


Figure 5.17 Switching to the programming mode

Step 3: Create a new object. Click the **Modify** tab to drag out the "Create New Object0” block, as shown in Figure 5.18.

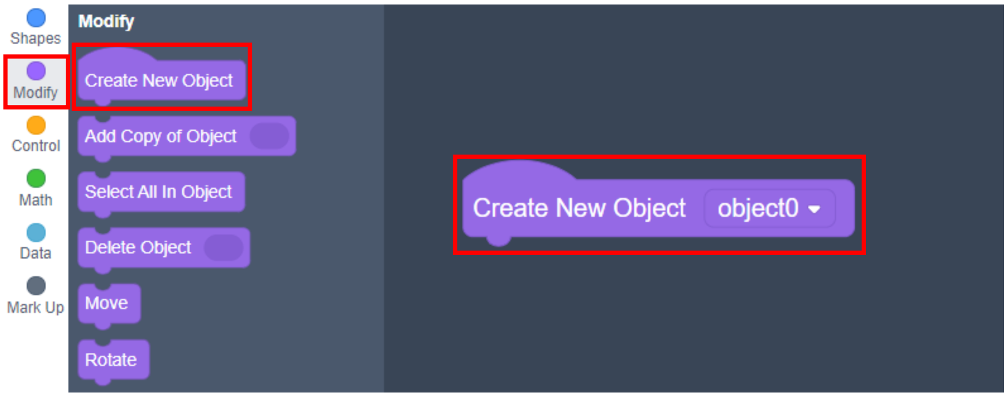


Figure 5.18 Creating the new object

Step 4: Add the cone. Click the “Shapes" tab, and drag out the "Cone" block, whose material is hole. Fill in the top radius 33 mm, the bottom radius 24.75 mm, and the height 69.3 mm, as shown in Figure 5.19.

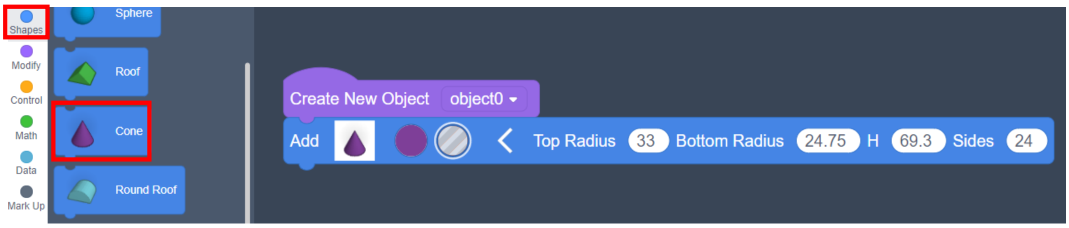
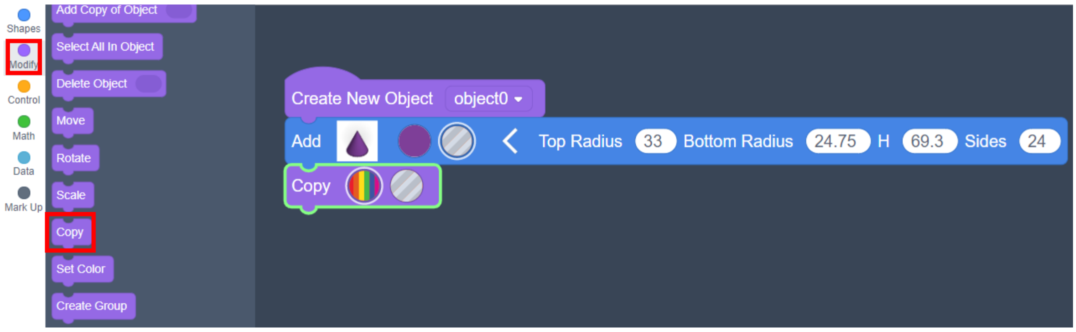


Figure 5.19 Adding the cone

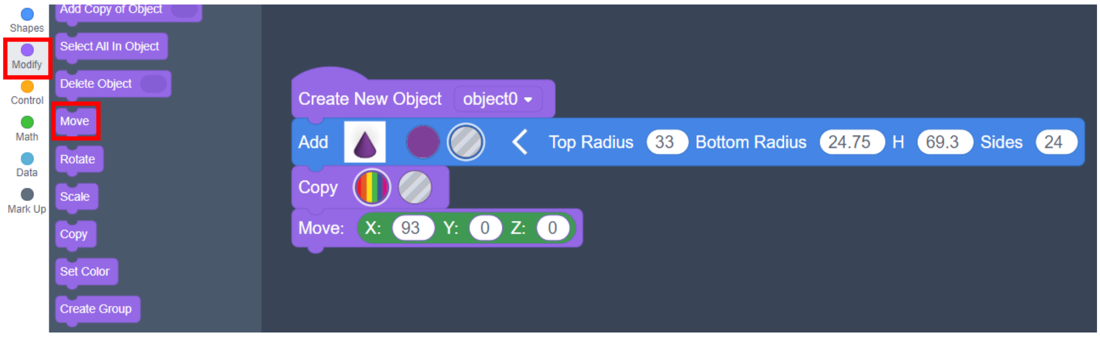
Step 5: Copy the truncated cone. Click the **Modify** tab, drag out the "Copy" block, and copy the truncated cone, as shown in Figure 5.20.



Create New Object object0
Add Top Radius 33 Bottom Radius 24.75 H 69.3 Sides 24
Copy

Figure 5.20 Copying the truncated cone

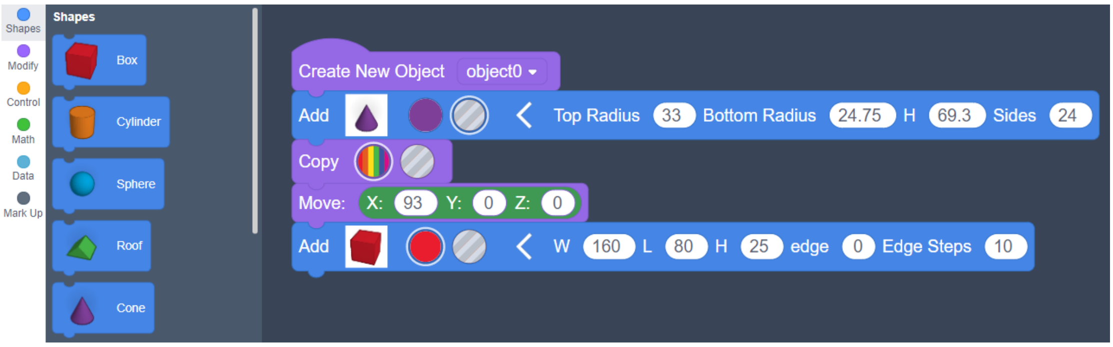
Step 6: Move the copied object. Task 2 indicates that the distance between the two truncated cones is 60 mm and their top radius is 33 mm. So the circle center is 93 mm away from the side, and move the copied object 93 mm along the X axis, as shown in Figure 5.21.



Create New Object object0
Add Top Radius 33 Bottom Radius 24.75 H 69.3 Sides 24
Copy
Move: X: 93 Y: 0 Z: 0

Figure 5.21 Moving the copied object

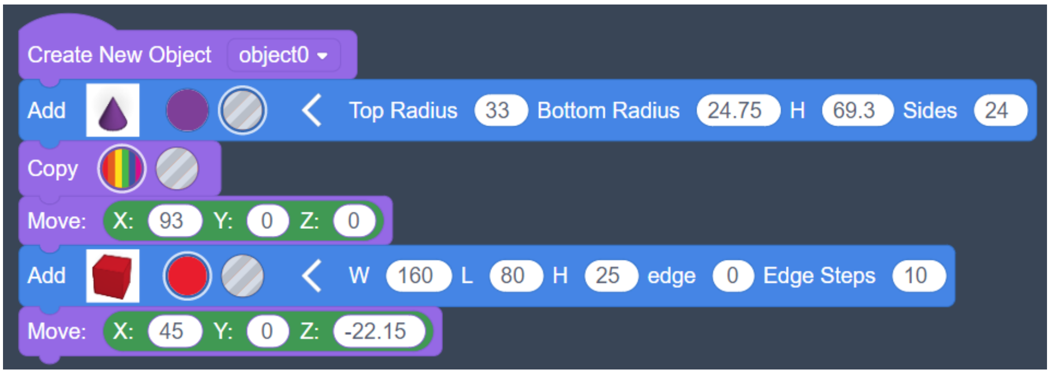
Step 7: Add a block. Click the **Shapes** tab, drag the “Box” block, and modify the parameters to 80 mm in length, 160 mm in width and 25 mm in height, as shown in Figure 5.22.



Create New Object object0
Add Top Radius 33 Bottom Radius 24.75 H 69.3 Sides 24
Copy
Move: X: 93 Y: 0 Z: 0
Add W 160 L 80 H 25 edge 0 Edge Steps 10

Figure 5.22 Adding the block

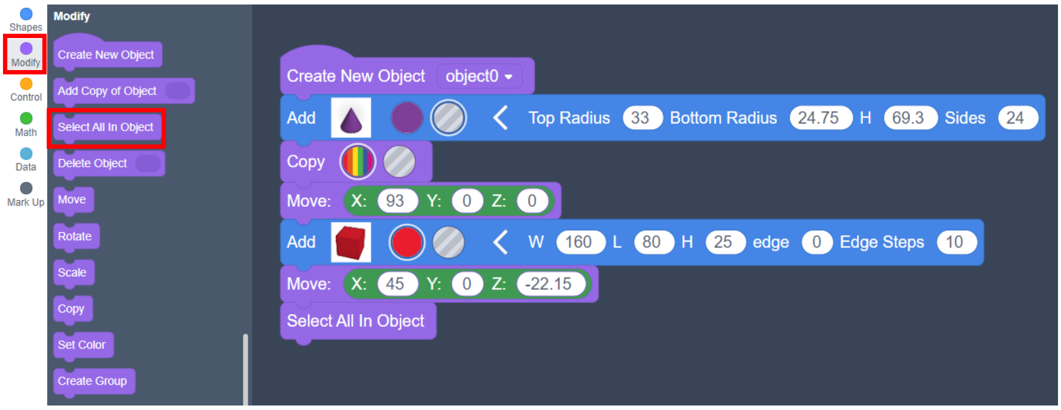
Step 8: Move the block. If the block and the central point of the first truncated cone are located at one point, the block needs to be moved 45 mm along the positive direction of the X axis and 22.15 mm along the negative direction of the Z axis, as shown in Figure 5.23.



```
Create New Object  object0
Add  <  Top Radius 33  Bottom Radius 24.75  H 69.3  Sides 24
Copy
Move:  X: 93  Y: 0  Z: 0
Add  <  W 160  L 80  H 25  edge 0  Edge Steps 10
Move:  X: 45  Y: 0  Z: -22.15
```

Figure 5.23 Moving the block

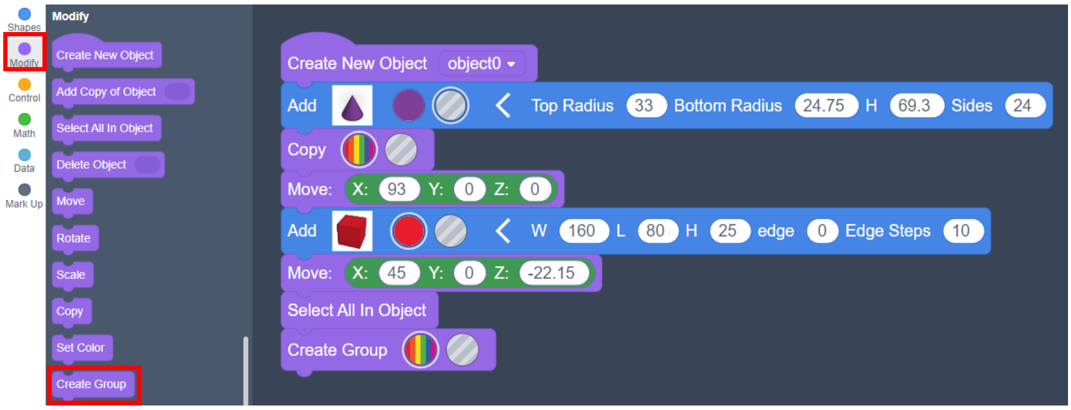
Step 9: Select All in Object. Click the **Modify** tab, and drag out the "Select All In Object" block, as shown in Figure 5.24.



```
Create New Object  object0
Add  <  Top Radius 33  Bottom Radius 24.75  H 69.3  Sides 24
Copy
Move:  X: 93  Y: 0  Z: 0
Add  <  W 160  L 80  H 25  edge 0  Edge Steps 10
Move:  X: 45  Y: 0  Z: -22.15
Select All In Object
```

Figure 5.24 Selecting All In Object

Step 10: Create a group. Click the **Modify** tab, and drag out the "Create Group" block, as shown in Figure 5.25.



```
Create New Object  object0
Add  <  Top Radius 33  Bottom Radius 24.75  H 69.3  Sides 24
Copy
Move:  X: 93  Y: 0  Z: 0
Add  <  W 160  L 80  H 25  edge 0  Edge Steps 10
Move:  X: 45  Y: 0  Z: -22.15
Select All In Object
Create Group
```

Figure 5.25 Creating the group

Step 11: Click the "run the program" to generate a saucer model, as shown in Figure 5.26.

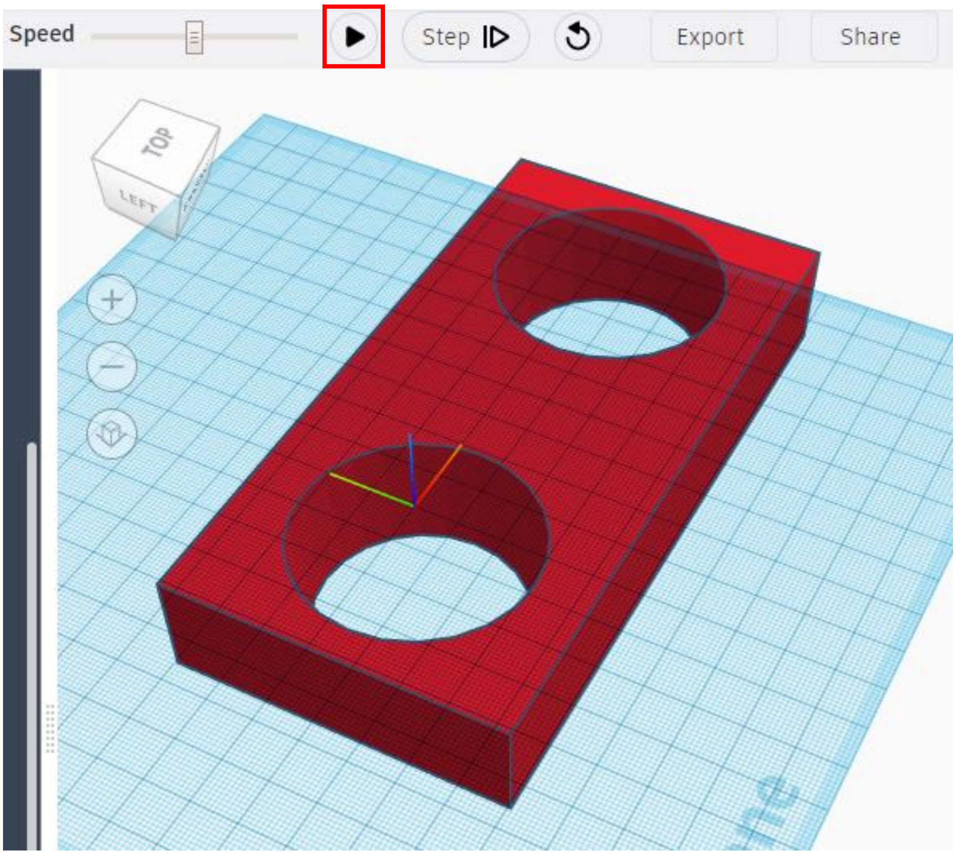


Figure 5.26 Running the program to generate the model

Think about the differences and similarities between the conventional modeling and the programming-based modeling.

3. Summary

(1) The advantages of programming-based modeling: ____________

____

(2) The steps for programming-based saucer modeling: ___________

____

4. Self-Assessment

Check the completed content in the experiment task. Tick (√) the completed items, and circle (◯) the uncompleted items.

Assessment Content	Completion Status
I’ve designed the saucer model through programming	
I’ve known the saucer modeling flow chart	
I’ve completed the saucer model design through programming	

Task 4: Program the Cup Clamp Model

The end of the robotic arm is replaceable. But if no available clamps meet requirements, what shall we do? This experiment tells us that the end of the robotic arm can be designed as needed. Let’s look at how to design a cup clamp.

1. Analysis

Use Tinkercad to design a saucer model and print it through 3D printing technology. Observe the model, analyze the modeling steps, and complete the realization steps in the blanks in the table below.

Step 1: Analyze the model of the cup clamp, and measure its ____.

Step 2: Design to plug the module of the end of the robotic arm.

Step 3: Design the support for the cup clamp.

Step 4: Design the cup handle.

Step 5: ______model.

Step 6: Maintain and print the model.

2. Steps

Step 1: As per the experiment analysis, read the flow chart of cup clamp programming, as shown in Figure 5.27.

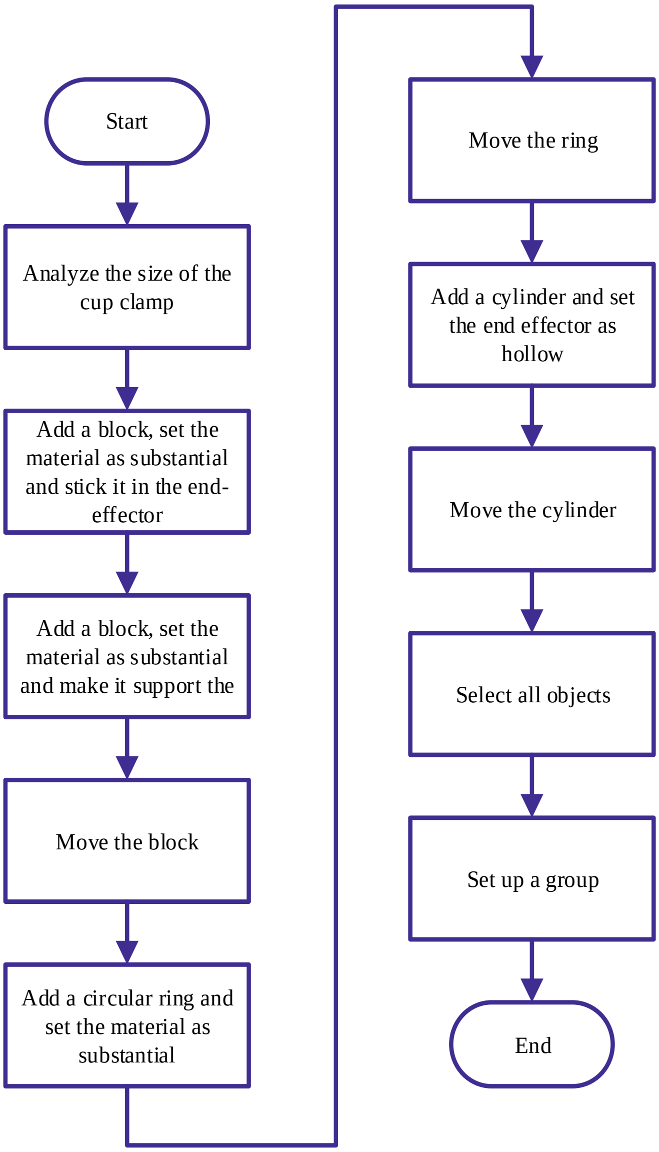

Start
Analyze the size of the cup clamp
Add a block, set the material as substantial and stick it in the end-effector
Add a block, set the material as substantial and make it support the
Move the block
Add a circular ring and set the material as substantial
Move the ring
Add a cylinder and set the end effector as hollow
Move the cylinder
Select all objects
Set up a group
End

Figure 5.27 Flow chart of cup clamp modeling

Step 2: Analyze the size of the cup clamp, as shown in Figure 5.28.

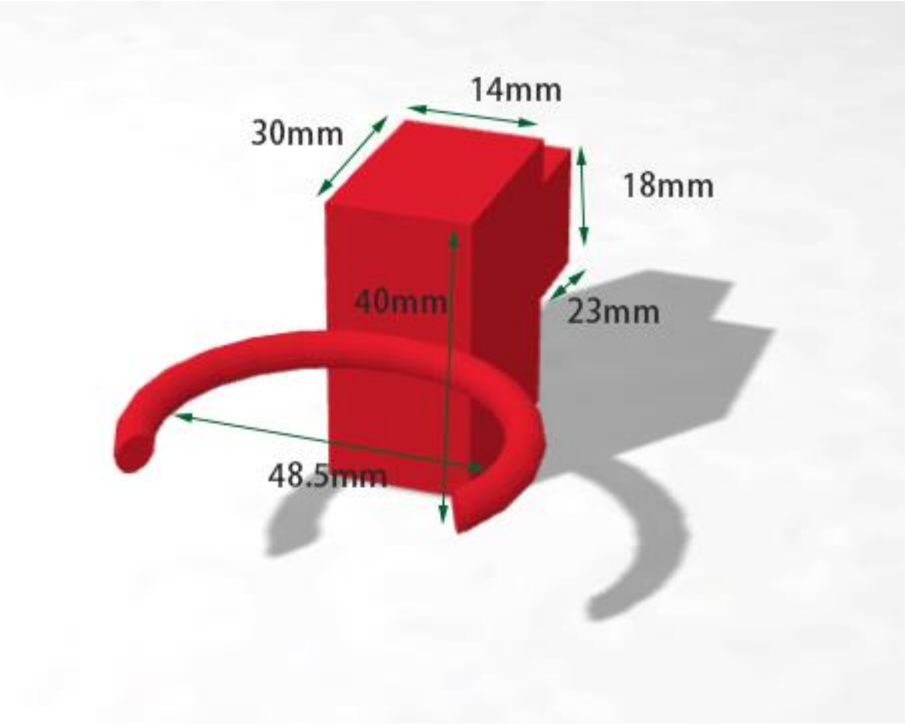
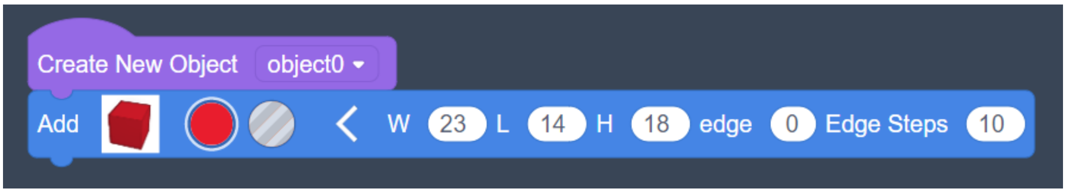


Figure 5.28 Size of the cup clamp

Step 3: Add a Box block. Set its parameters to 23 mm in width, 14 mm in length and 18 mm in height, as shown in Figure 5.29.

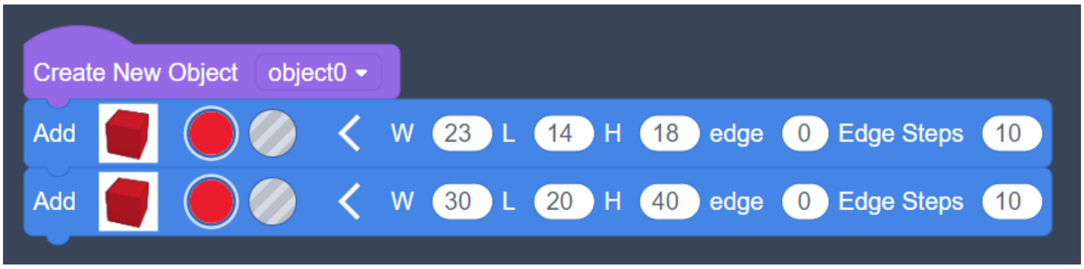


Create New Object object0 ▾

Add W 23 L 14 H 18 edge 0 Edge Steps 10

Figure 5.29 Adding the block to the end of the robotic arm

Step 4: Add a solid block. Set its parameters to 30 mm in width, 20 mm in length and 40 mm in height to support a ring, as shown in Figure 5.30.



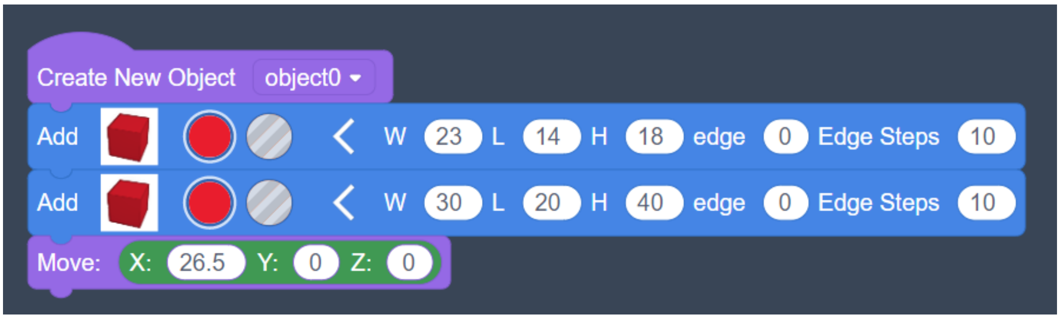
Create New Object object0 ▾

Add W 23 L 14 H 18 edge 0 Edge Steps 10

Add W 30 L 20 H 40 edge 0 Edge Steps 10

Figure 5.30 Adding the block as a ring support

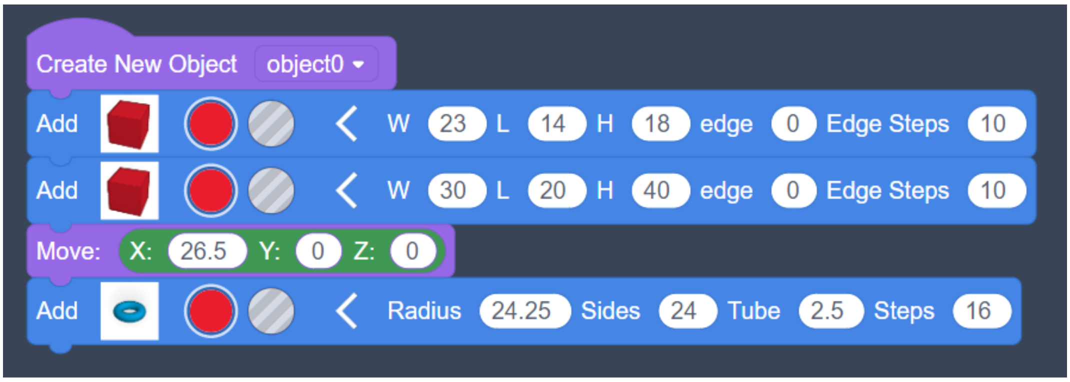
Step 5: Move the block. In the above step the two blocks have the same central point. Divide them, and move them 11.5 +15 = 26.5 mm along the positive direction of the X axis, as shown in Figure 5.31.



```
Create New Object  object0
Add  W 23  L 14  H 18  edge 0  Edge Steps 10
Add  W 30  L 20  H 40  edge 0  Edge Steps 10
Move:  X: 26.5  Y: 0  Z: 0
```

Figure 5.31 Moving the blocks

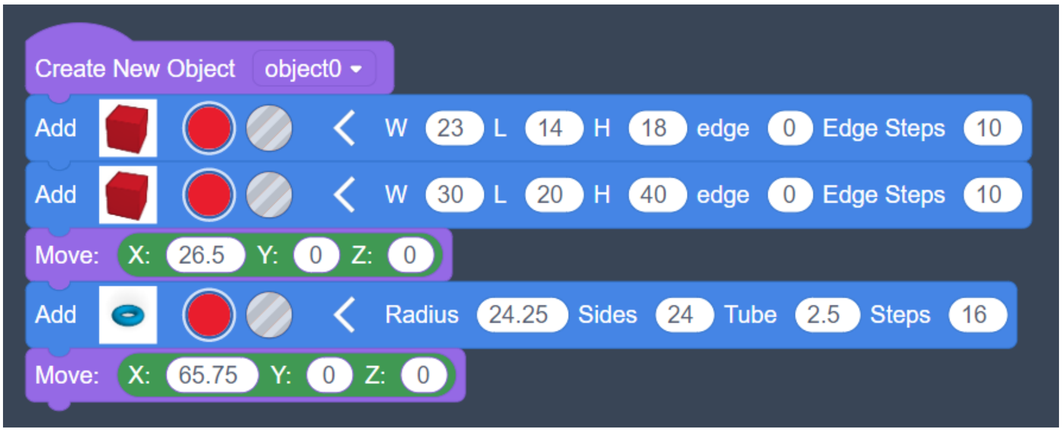
Step 6: Add the ring. The ring is solid and red. Modify its parameter to 24.5 mm in radius, and the parameter of the suction tube to 2.5 mm in radius, as shown in Figure 5.32.



```
Create New Object  object0
Add  W 23  L 14  H 18  edge 0  Edge Steps 10
Add  W 30  L 20  H 40  edge 0  Edge Steps 10
Move:  X: 26.5  Y: 0  Z: 0
Add  Radius 24.25  Sides 24  Tube 2.5  Steps 16
```

Figure 5.32 Adding the ring

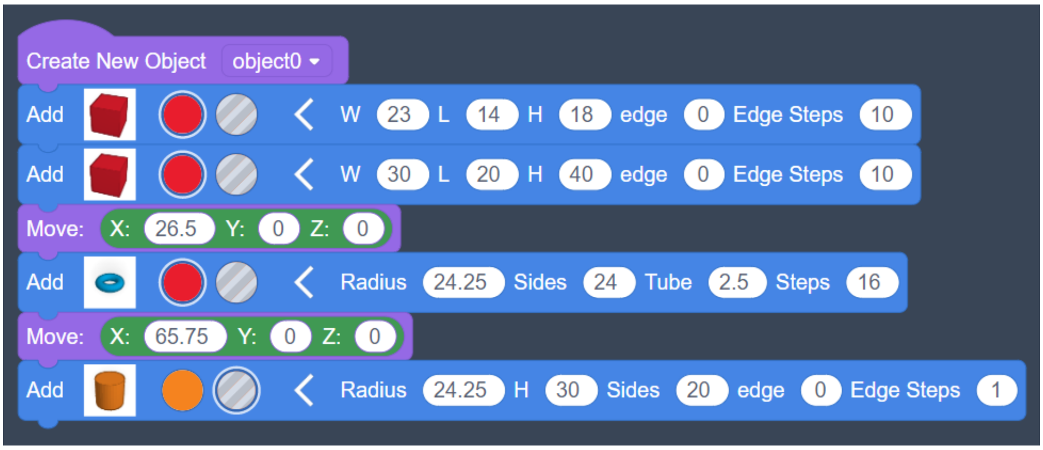
Step 7: Move the ring. In the above step the ring and the first block have the same central point. Move the ring 65.75 mm along the positive direction of the X axis, as shown in Figure 5.33.



```
Create New Object  object0
Add  W 23  L 14  H 18  edge 0  Edge Steps 10
Add  W 30  L 20  H 40  edge 0  Edge Steps 10
Move:  X: 26.5  Y: 0  Z: 0
Add  Radius 24.25  Sides 24  Tube 2.5  Steps 16
Move:  X: 65.75  Y: 0  Z: 0
```

Figure 5.33 Moving the ring

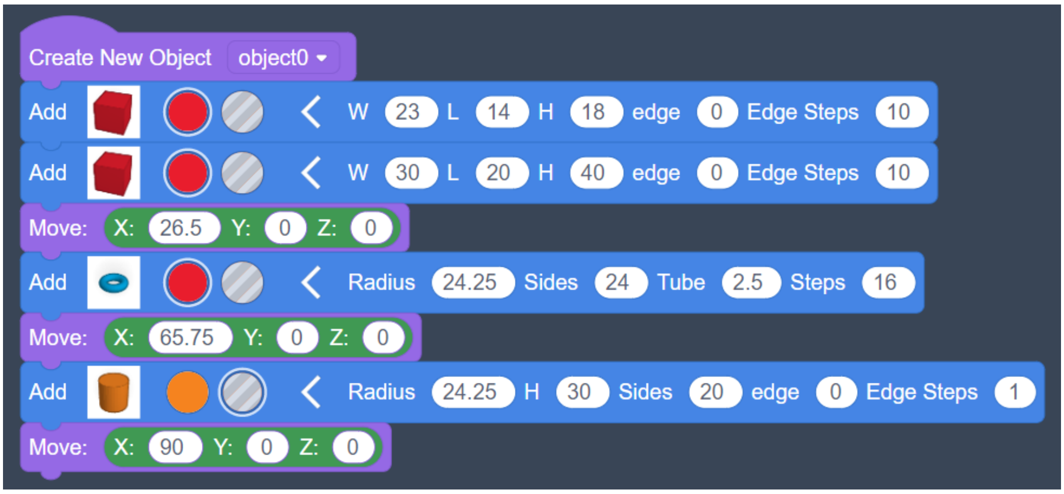
Step 8: Add a cylinder with the material being the hole. Set its parameters to 24.5 in radius and 30 mm in height, as shown in Figure 5.34.



```
Create New Object  object0 ▾
Add  W 23 L 14 H 18 edge 0 Edge Steps 10
Add  W 30 L 20 H 40 edge 0 Edge Steps 10
Move: X: 26.5 Y: 0 Z: 0
Add  Radius 24.25 Sides 24 Tube 2.5 Steps 16
Move: X: 65.75 Y: 0 Z: 0
Add  Radius 24.25 H 30 Sides 20 edge 0 Edge Steps 1
```

Figure 5.34 Adding the cylinder

Step 9: Move the cylinder. Move it 90 mm along the positive direction of the X axis, as shown in Figure 5.35.

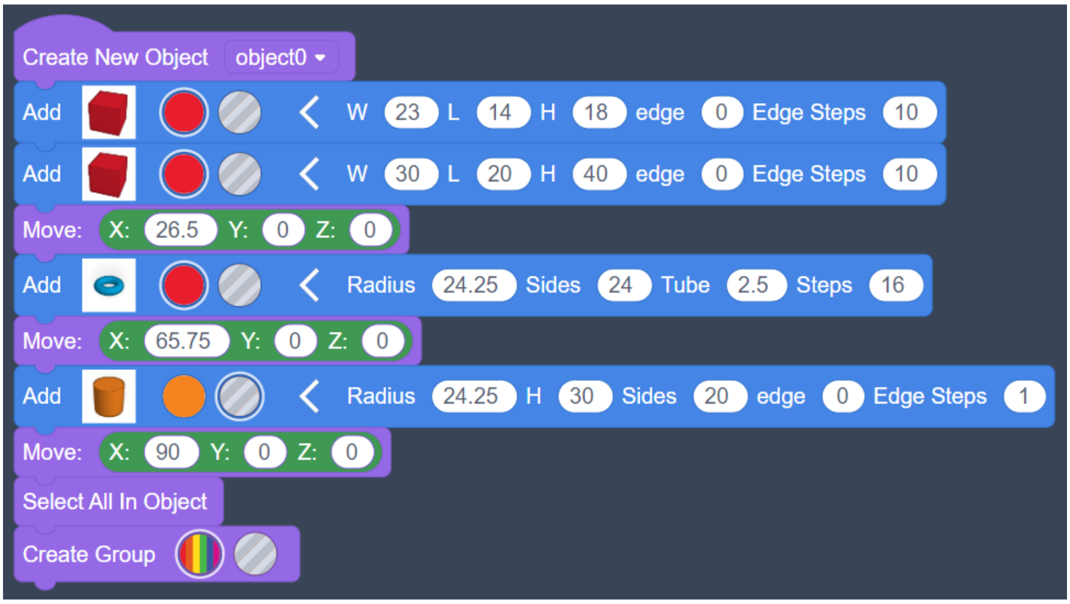


```
Create New Object  object0 ▾
Add  W 23 L 14 H 18 edge 0 Edge Steps 10
Add  W 30 L 20 H 40 edge 0 Edge Steps 10
Move: X: 26.5 Y: 0 Z: 0
Add  Radius 24.25 Sides 24 Tube 2.5 Steps 16
Move: X: 65.75 Y: 0 Z: 0
Add  Radius 24.25 H 30 Sides 20 edge 0 Edge Steps 1
Move: X: 90 Y: 0 Z: 0
```

Figure 5.35 Moving the cylinder

For the calculation skill of movement distance of each object, the center distance between the first block and the second one is: Half of the width of the first block + Half of the width of the second block = 11.5 + 15 = 16 mm. The center distance between the ring and the first block is: Half of the width of the first block + Width of the second block + Radius of the ring = 11.5 + 30 + 24.25 = 65.75 mm.

Step 10: Select **All In Object** and create a group, as shown in Figure 5.36.



```
Create New Object  object0 ▾
Add  ●  W 23  L 14  H 18  edge 0  Edge Steps 10
Add  ●  W 30  L 20  H 40  edge 0  Edge Steps 10
Move:  X: 26.5  Y: 0  Z: 0
Add  ●  Radius 24.25  Sides 24  Tube 2.5  Steps 16
Move:  X: 65.75  Y: 0  Z: 0
Add  ●  Radius 24.25  H 30  Sides 20  edge 0  Edge Steps 1
Move:  X: 90  Y: 0  Z: 0
Select All In Object
Create Group
```

Figure 5.36 Select All In Object and create group

Step 11: Click the "run the program" to generate a cup clamp model, as shown in Figure 5.37.

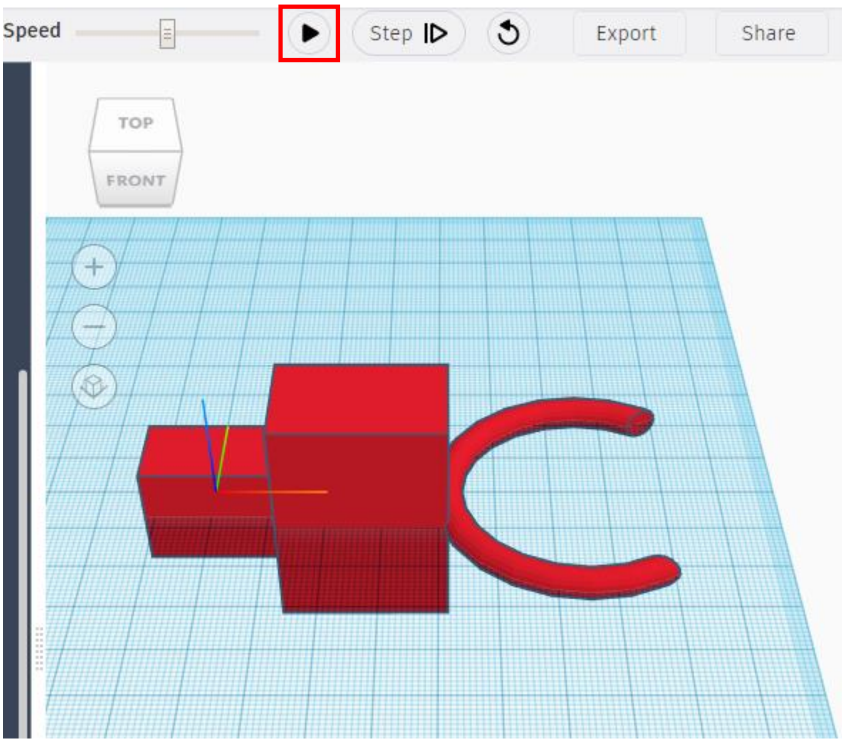


Figure 5.37 Running the program to generate the model

3. Summary

(1) The center distance between objects is calculated below: __________

____

(2) The steps to program the modeling of the cup clamp: ______________

____

4. Self-Assessment

Check the completed content in the experiment task. Tick (√) the completed items, and circle (○) the uncompleted items.

Assessment Content	Completion Status
I've known how to calculate the center distance between objects	
I've tried designing the end of the robotic arm as needed	
I've designed the cup clamp model through programming	

Task 5: Design the Bar Counter

We have completed the preparations. Let's rely on the speech recognition technology to grab a certain drink on hearing the speech instruction, and simulate the method of designing a bar counter assistant.

1. Analysis

The teacher demonstrates how to design a bar counter assistant. As per the movement condition of the robotic arm, analyze the steps to program the bar counter assistant, and complete the steps in the blanks in the table below.

Step 1: Set a robotic arm, a saucer and a cup.

Step 2: Initialize the placement position of the robotic arms.

Step 3: _______ .

Step 4: Judge _______.

Step 5: If the keywords include _____, grab coffee.

Step 6: If the keywords include _____, grab water.

Step 7: Repeat steps 3, 4, 5 and 6.

2. Steps

(1) Prepare Hardware

Step 1: Set the robotic arm, the saucer and the cup, as shown in Figure 5.38.

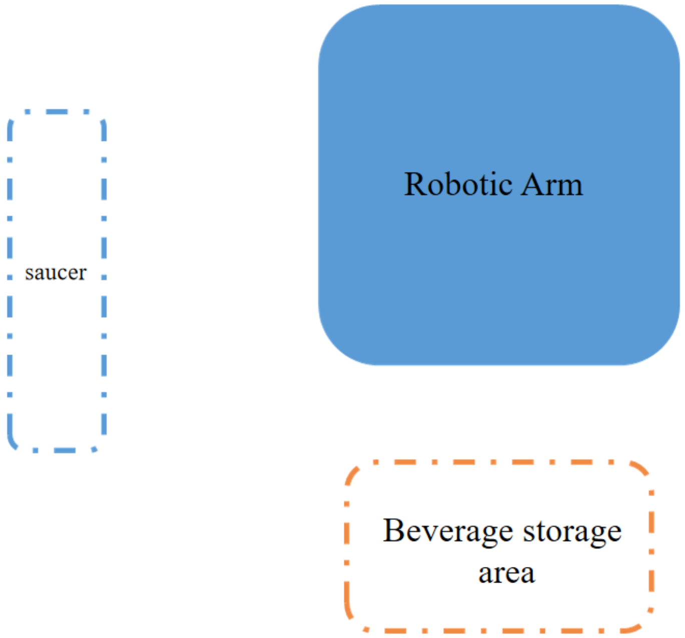


Figure 5.38 Setting the materials

(2) Design Program

Step 1: As per the experiment analysis, read the design flow chart of bar counter assistant, as shown in Figure 5.39.

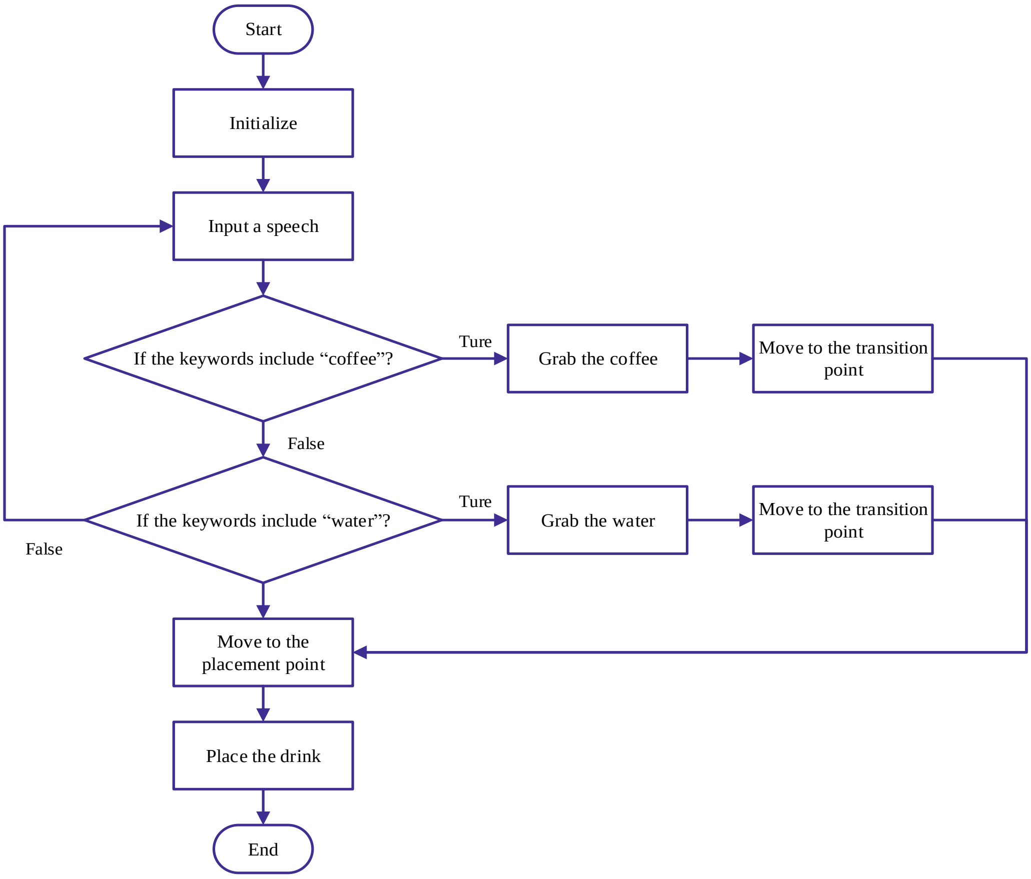
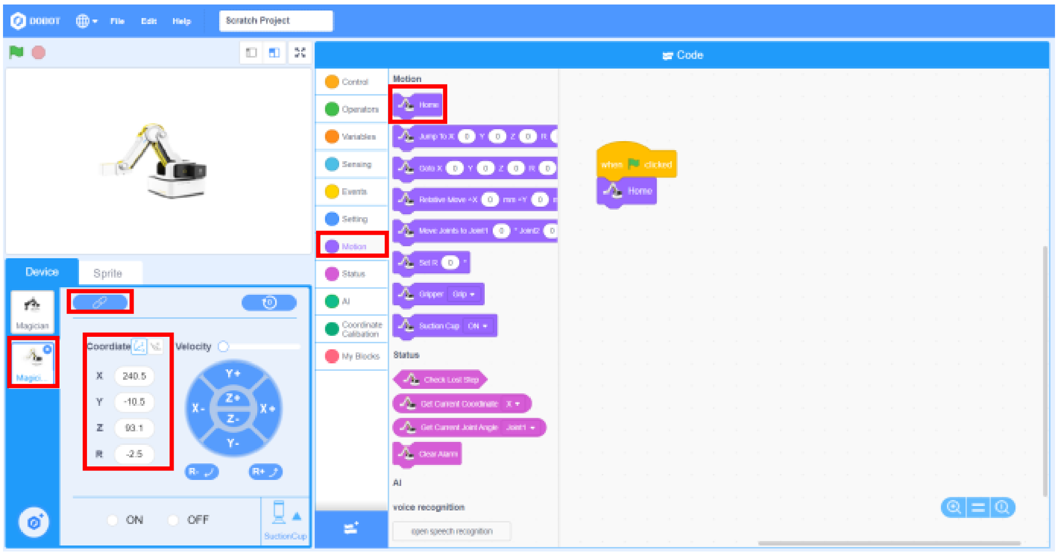

Start
Initialize
Input a speech
If the keywords include “coffee”?
Ture
Grab the coffee
Move to the transition point
False
If the keywords include “water”?
Ture
Grab the water
Move to the transition point
False
Move to the placement point
Place the drink
End

Figure 5.39 Designing the flowchart of a smart bar counter

Step 2: Initialize the equipment. Enable Dobot Scratch software, click the Add Device "Magician Lite", Connect Device, Add Extension "AI", Robotic Arm Home, and Set the End Tool icons, as shown in Figure 5.40.



X 240.5
Y -10.5
Z 93.1
R -2.5

Figure 5.40 Initialization

Step 3: Input a speech. Click the **AI** tab, drag out the "start Chinese_putonghua voice recognition, continued for 1 second" block, and set its parameter to 5, as shown in Figure 5.41.

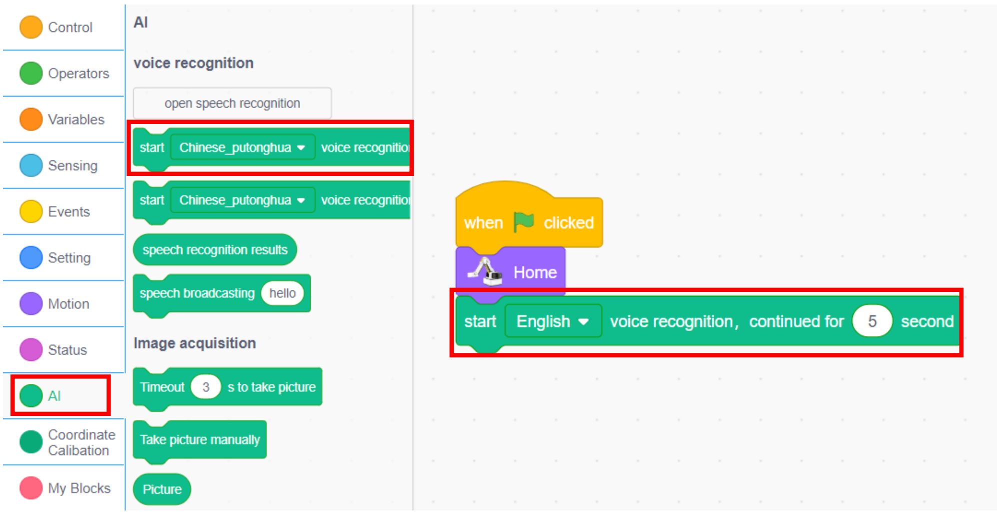

Control
Operators
Variables
Sensing
Events
Setting
Motion
Status
AI
Coordinate Calibation
My Blocks
AI
voice recognition
open speech recognition
start Chinese_putonghua voice recognitio
start Chinese_putonghua voice recognitio
speech recognition results
speech broadcasting hello
Image acquisition
Timeout 3 s to take picture
Take picture manually
Picture
when clicked
Home
start English voice recognition, continued for 5 second

Figure 5.41 Inputing the speech

Step 4: Judge whether the keywords of the speech recognition result include "coffee", as shown in Figure 5.42.

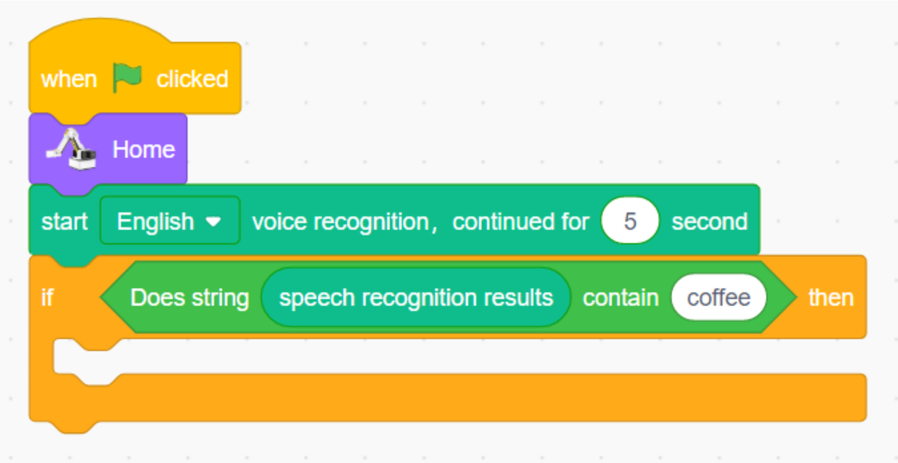
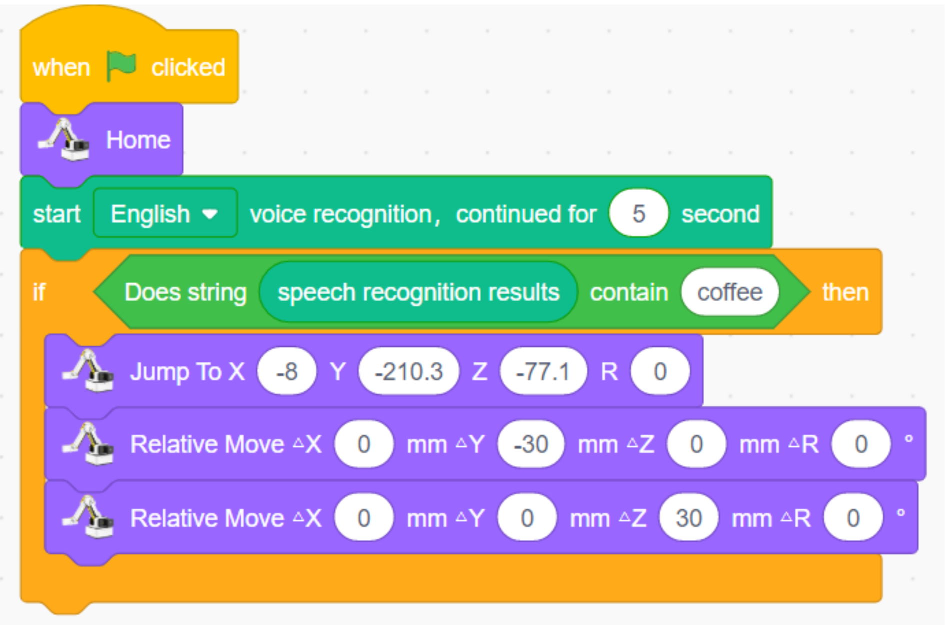

when clicked
Home
start English voice recognition, continued for 5 second
if Does string speech recognition results contain coffee then

Figure 5.42 Judging whether the speech recognition result includes coffee

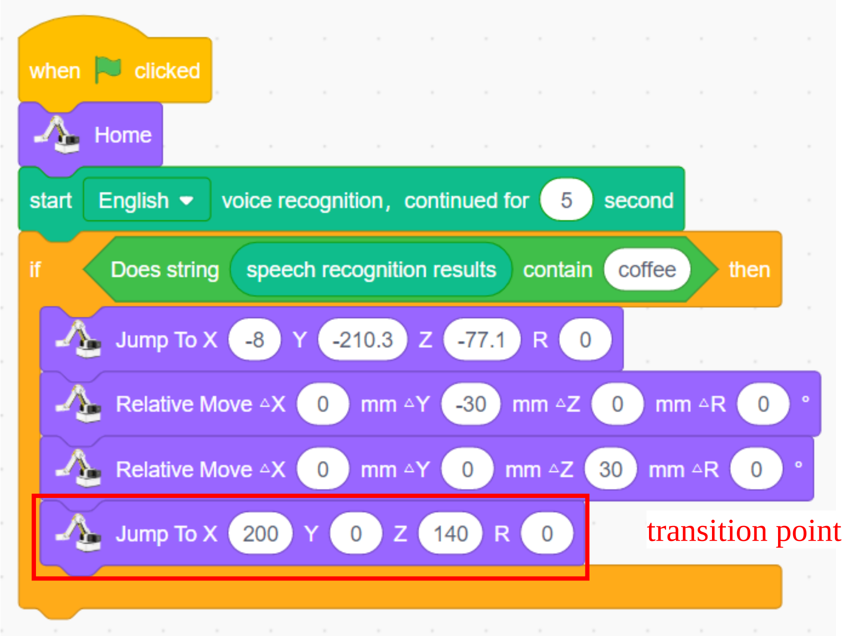
Step 5: Grab coffee. Move to the position in front of the coffee cup, and grab coffee, as shown in Figure 5.43.



```
when 🏁 clicked
Home
start English ▾ voice recognition, continued for 5 second
if Does string speech recognition results contain coffee then
    Jump To X -8 Y -210.3 Z -77.1 R 0
    Relative Move ΔX 0 mm ΔY -30 mm ΔZ 0 mm ΔR 0 °
    Relative Move ΔX 0 mm ΔY 0 mm ΔZ 30 mm ΔR 0 °
```

Figure 5.43 Grabbing coffee

Step 6: Moving the robotic arm to the transition point (200, 0, 140, and 0), as shown in Figure 5.44.

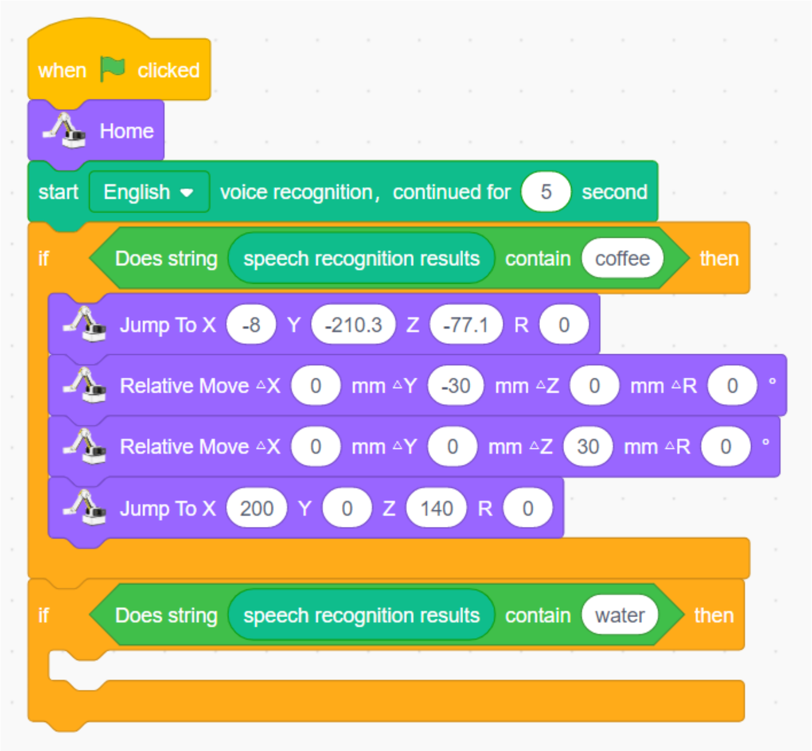


```
when 🏁 clicked
Home
start English ▾ voice recognition, continued for 5 second
if Does string speech recognition results contain coffee then
    Jump To X -8 Y -210.3 Z -77.1 R 0
    Relative Move ΔX 0 mm ΔY -30 mm ΔZ 0 mm ΔR 0 °
    Relative Move ΔX 0 mm ΔY 0 mm ΔZ 30 mm ΔR 0 °
    Jump To X 200 Y 0 Z 140 R 0
```

transition point

Figure 5.44 Moving to the transition point

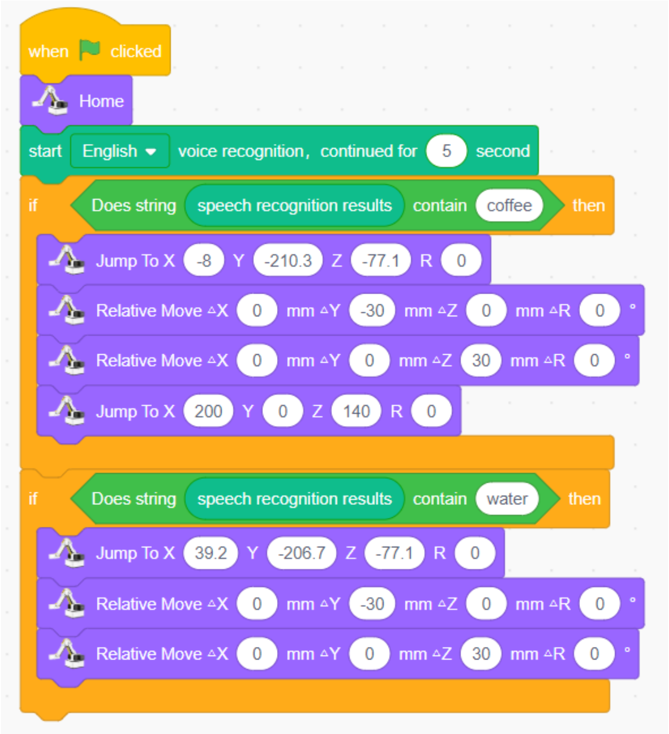
Step 7: Judge whether the keywords of the speech recognition result include "water", as shown in Figure 5.45.



```
when 🏁 clicked
Home
start English ▾ voice recognition，continued for 5 second
if Does string speech recognition results contain coffee then
    Jump To X -8 Y -210.3 Z -77.1 R 0
    Relative Move △X 0 mm △Y -30 mm △Z 0 mm △R 0 °
    Relative Move △X 0 mm △Y 0 mm △Z 30 mm △R 0 °
    Jump To X 200 Y 0 Z 140 R 0
if Does string speech recognition results contain water then
```

Figure 5.45 Judging whether the keywords of the speech recognition result include "water"

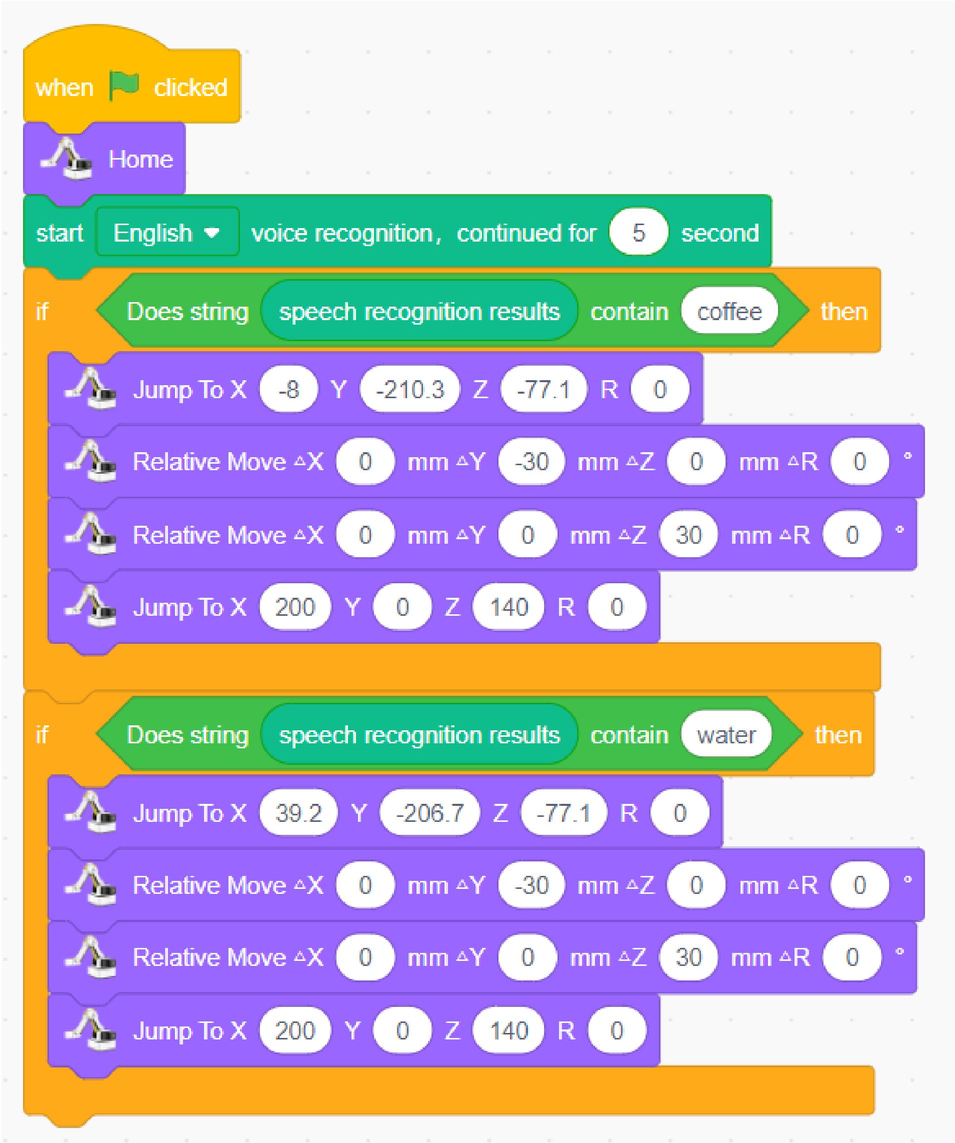
Step 8: Grab the water cup. Move to the position in front of the cup, and grab it, as shown in Figure 5.46.



```
when 🏁 clicked
Home
start English ▾ voice recognition，continued for 5 second
if Does string speech recognition results contain coffee then
    Jump To X -8 Y -210.3 Z -77.1 R 0
    Relative Move △X 0 mm △Y -30 mm △Z 0 mm △R 0 °
    Relative Move △X 0 mm △Y 0 mm △Z 30 mm △R 0 °
    Jump To X 200 Y 0 Z 140 R 0
if Does string speech recognition results contain water then
    Jump To X 39.2 Y -206.7 Z -77.1 R 0
    Relative Move △X 0 mm △Y -30 mm △Z 0 mm △R 0 °
    Relative Move △X 0 mm △Y 0 mm △Z 30 mm △R 0 °
```

Figure 5.46 Grabbing the cup

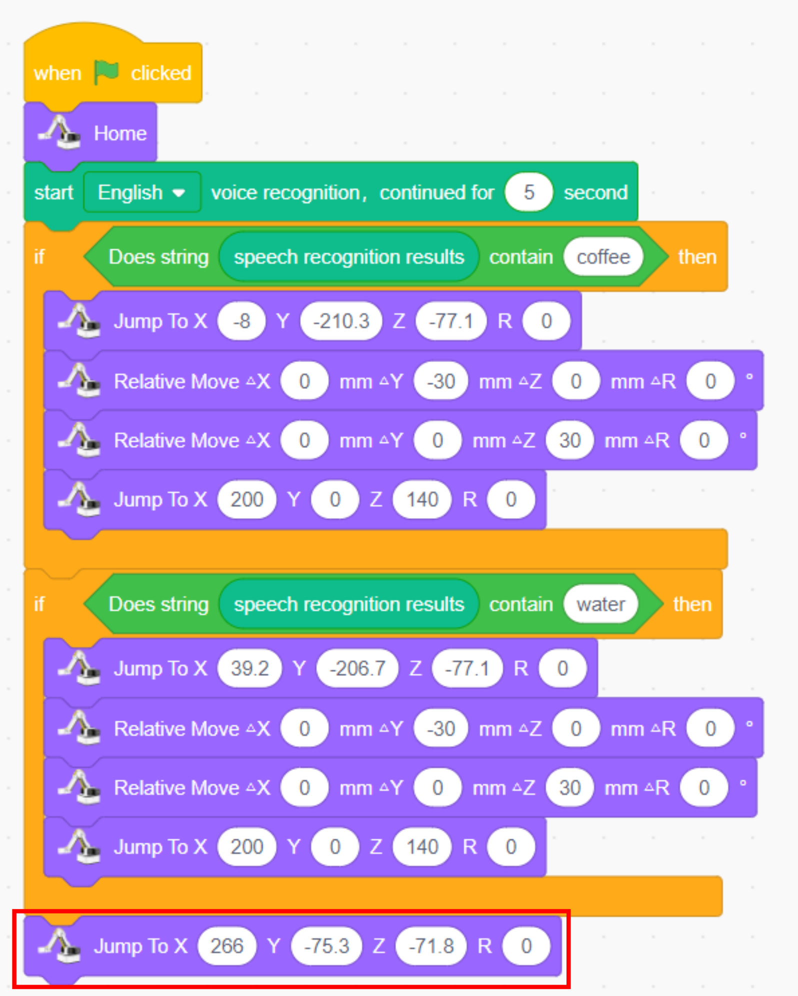
Step 9: Move the robotic arm to the transition point (200, 0, 140 and 0), as shown in Figure 5.47.



```
when ⚑ clicked
Home
start English ▾ voice recognition, continued for 5 second
if Does string speech recognition results contain coffee then
    Jump To X -8 Y -210.3 Z -77.1 R 0
    Relative Move ΔX 0 mm ΔY -30 mm ΔZ 0 mm ΔR 0 °
    Relative Move ΔX 0 mm ΔY 0 mm ΔZ 30 mm ΔR 0 °
    Jump To X 200 Y 0 Z 140 R 0
if Does string speech recognition results contain water then
    Jump To X 39.2 Y -206.7 Z -77.1 R 0
    Relative Move ΔX 0 mm ΔY -30 mm ΔZ 0 mm ΔR 0 °
    Relative Move ΔX 0 mm ΔY 0 mm ΔZ 30 mm ΔR 0 °
    Jump To X 200 Y 0 Z 140 R 0
```

Figure 5.47 Moving to the transition point

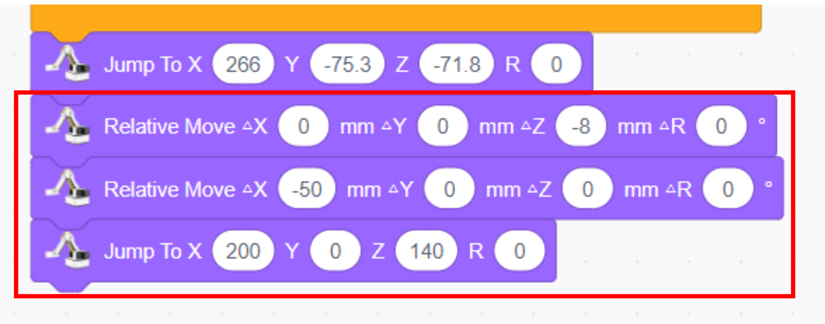
Step 10: Move the robotic arm to the placement point (266, -75.3, and -71.8), as shown in Figure 5.48.



```
when 🏳 clicked
Home
start English ▾ voice recognition, continued for 5 second
if Does string speech recognition results contain coffee then
    Jump To X -8 Y -210.3 Z -77.1 R 0
    Relative Move ΔX 0 mm ΔY -30 mm ΔZ 0 mm ΔR 0 °
    Relative Move ΔX 0 mm ΔY 0 mm ΔZ 30 mm ΔR 0 °
    Jump To X 200 Y 0 Z 140 R 0
if Does string speech recognition results contain water then
    Jump To X 39.2 Y -206.7 Z -77.1 R 0
    Relative Move ΔX 0 mm ΔY -30 mm ΔZ 0 mm ΔR 0 °
    Relative Move ΔX 0 mm ΔY 0 mm ΔZ 30 mm ΔR 0 °
    Jump To X 200 Y 0 Z 140 R 0
Jump To X 266 Y -75.3 Z -71.8 R 0
```

Figure 5.48 Moving to the placement point

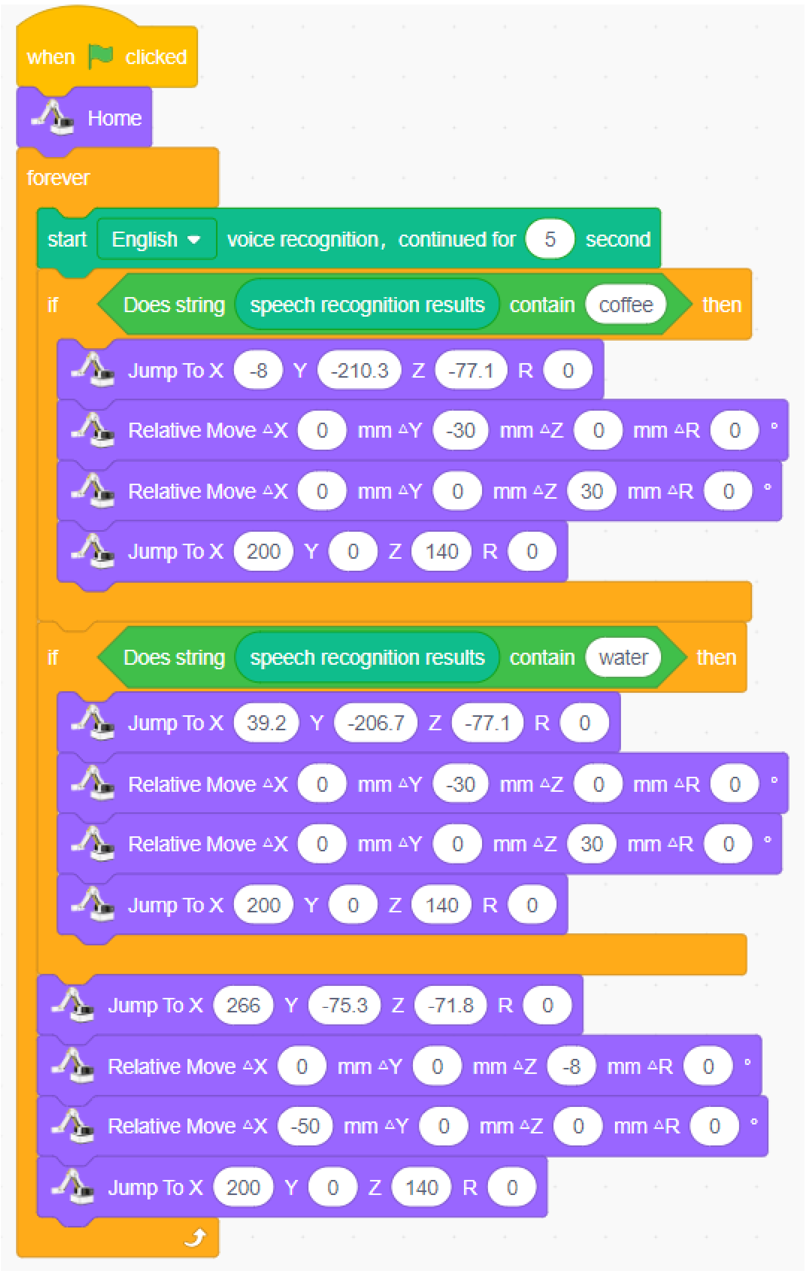
Step 11: Place drink. Design how the robotic arms put drink, as shown in Figure 5.49.



```
Jump To X 266 Y -75.3 Z -71.8 R 0
Relative Move ΔX 0 mm ΔY 0 mm ΔZ -8 mm ΔR 0 °
Relative Move ΔX -50 mm ΔY 0 mm ΔZ 0 mm ΔR 0 °
Jump To X 200 Y 0 Z 140 R 0
```

Figure 5.49 Placing the drink

Step 12: Repeat the execution and integrate the program, as shown in Figure 5.50.

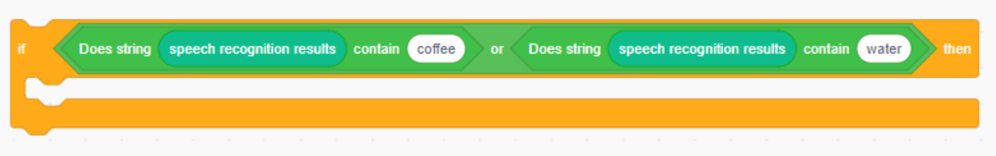


```
when 🏳 clicked
Home
forever
    start English ▾ voice recognition, continued for 5 second
    if Does string speech recognition results contain coffee then
        Jump To X -8 Y -210.3 Z -77.1 R 0
        Relative Move ΔX 0 mm ΔY -30 mm ΔZ 0 mm ΔR 0 °
        Relative Move ΔX 0 mm ΔY 0 mm ΔZ 30 mm ΔR 0 °
        Jump To X 200 Y 0 Z 140 R 0
    if Does string speech recognition results contain water then
        Jump To X 39.2 Y -206.7 Z -77.1 R 0
        Relative Move ΔX 0 mm ΔY -30 mm ΔZ 0 mm ΔR 0 °
        Relative Move ΔX 0 mm ΔY 0 mm ΔZ 30 mm ΔR 0 °
        Jump To X 200 Y 0 Z 140 R 0
    Jump To X 266 Y -75.3 Z -71.8 R 0
    Relative Move ΔX 0 mm ΔY 0 mm ΔZ -8 mm ΔR 0 °
    Relative Move ΔX -50 mm ΔY 0 mm ΔZ 0 mm ΔR 0 °
    Jump To X 200 Y 0 Z 140 R 0
```

Figure 5.50 Smart bar counter program

After the program is run, which problems do you find?

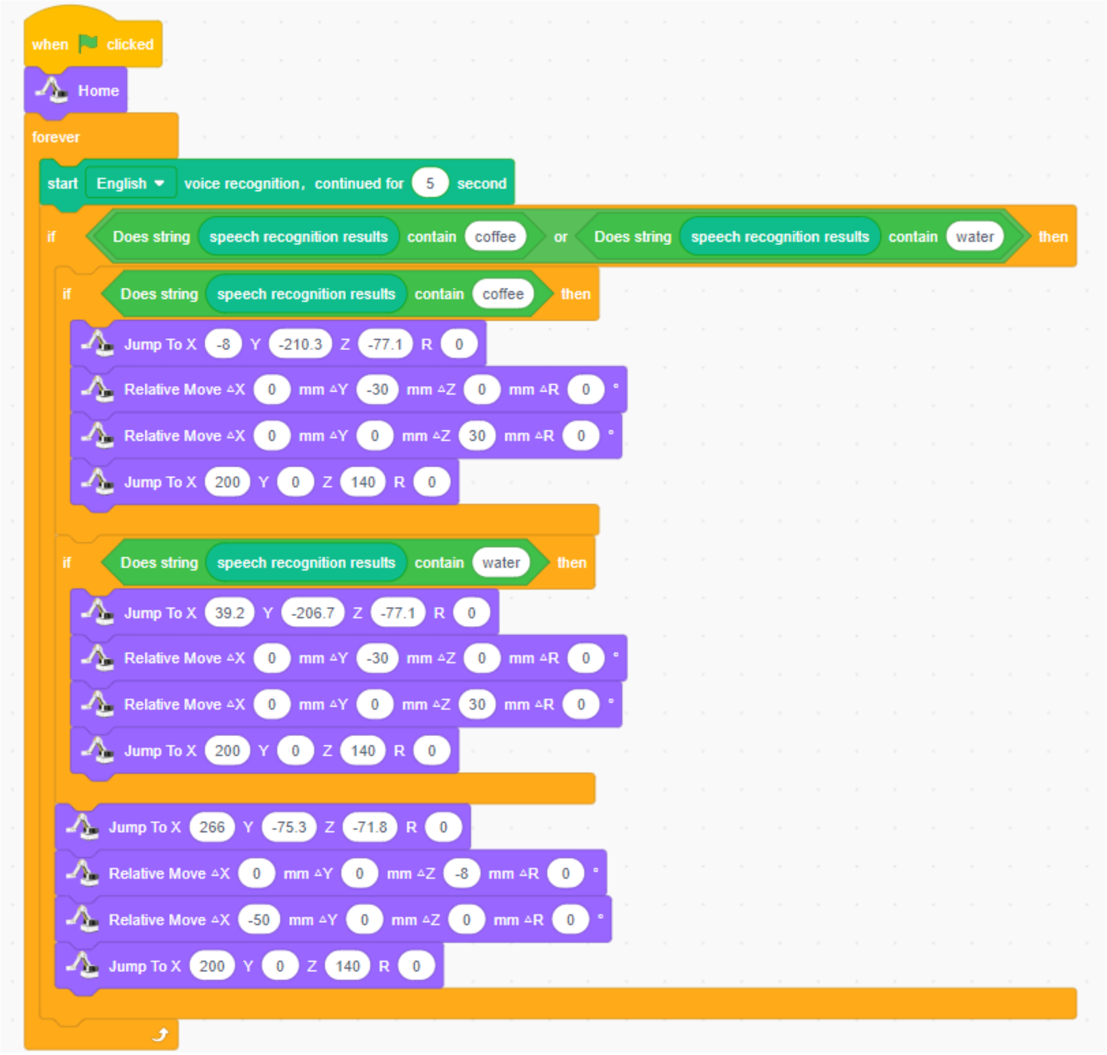
Step 13: Prevent the robotic arm from repetitively placing drink. Place drink only when getting the speech input of drink. Take all information as a condition for judging whether to place drink, as shown in Figure 5.51.



```
if < Does string (speech recognition results) contain (coffee) > or < Does string (speech recognition results) contain (water) > then

```

Figure 5.51 Judgment condition of all information

Step 14: Integrate the program, as shown in Figure 5.52.



```
when [flag] clicked
Home
forever
    start English ▾ voice recognition, continued for (5) second
    if < Does string (speech recognition results) contain (coffee) > or < Does string (speech recognition results) contain (water) > then
        if < Does string (speech recognition results) contain (coffee) > then
            Jump To X (-8) Y (-210.3) Z (-77.1) R (0)
            Relative Move ΔX (0) mm ΔY (-30) mm ΔZ (0) mm ΔR (0) °
            Relative Move ΔX (0) mm ΔY (0) mm ΔZ (30) mm ΔR (0) °
            Jump To X (200) Y (0) Z (140) R (0)
        if < Does string (speech recognition results) contain (water) > then
            Jump To X (39.2) Y (-206.7) Z (-77.1) R (0)
            Relative Move ΔX (0) mm ΔY (-30) mm ΔZ (0) mm ΔR (0) °
            Relative Move ΔX (0) mm ΔY (0) mm ΔZ (30) mm ΔR (0) °
            Jump To X (200) Y (0) Z (140) R (0)
        Jump To X (266) Y (-75.3) Z (-71.8) R (0)
        Relative Move ΔX (0) mm ΔY (0) mm ΔZ (-8) mm ΔR (0) °
        Relative Move ΔX (-50) mm ΔY (0) mm ΔZ (0) mm ΔR (0) °
        Jump To X (200) Y (0) Z (140) R (0)
```

Figure 5.52 Smart bar counter modified

3. Summary

(1) The applications of the speech recognition technology in this experiment: ____

____

(2) The steps to design the smart bar counter: ____________________

____

4. Self-Assessment

Check the completed content in the experiment task. Tick (√) the completed items, and circle (◯) the uncompleted items.

Assessment Content	Completion Status
I’ve known the speech recognition technology	
I’ve programmed the bar counter assistant design	

Experiment 6

Auto Order Pickup and Purchase in Supermarket

Team Name: Team member: Date:

Overview

What do you think a smart supermarket is like? How do you check, select and purchase commodities in it? In this experiment, we conduct some simple design, involving two tasks. Task 1 is about "query of commodity information", which is to store commodity information in the list to help its query and modification. Task 2 is about "auto commodity pickup", which is to allow the robotic arm to automatically select commodities through the character recognition technology.

Objective

1. Master the basic operations of the sprite in the interface by completing task 1 "query of commodity information"..
2. By completing task 1 "query of commodity information", understand how to use the list.
3. Master the optical character recognition (OCR) instruction by designing the "auto commodity pickup" system.
4. By designing the "auto commodity pickup" system, understand how to use the nested structure.

Equipment

Equipment Picture	Name	Quantity
	Dobot Magician Lite Robotic arm	1
	Suction cup kit	1
	Camera	1
	Type-C cable	1
	Power adapter	1
CHOC OLATE BISCUIT CHOC OLATE MILK CHOC OLATE ALMOND CHOC OLATE HAZELNUT	Commodity	A certain number
	Card	A certain number

Requirements

- ➢ Take care when using electricity.

- Before the experiment, check whether the experimental equipment is complete and intact. If there is any omission or damage, please report to the teacher.
- Any specific operations in the experiment should be performed according to the experiment manual. If you have any questions, please promptly ask the teacher.
- During the experiment, the joints will start to work as the robotic arms are powered on. In that case, do not move the joints of the robotic arms hard if you do not press the unlock key.
- Report any device fault during the experiment to your teacher in a timely manner, and do not handle it yourself.
- Arrange all devices after the experiment. You shall not leave the lab before check by the group leader.

Task 1: Design the Query of Commodity

This task is to store the information of supermarket commodities in the list to help staff query and update it.

1. Analysis

Analyze the steps to store commodity information in the list. Then, fill in the blanks in the table below according to the prompts and your analysis.

Step 1: Enable DobotScratch, and switch to the **Sprite** interface from the **Device** interface.

Step 2: Delete the default sprite on the stage.

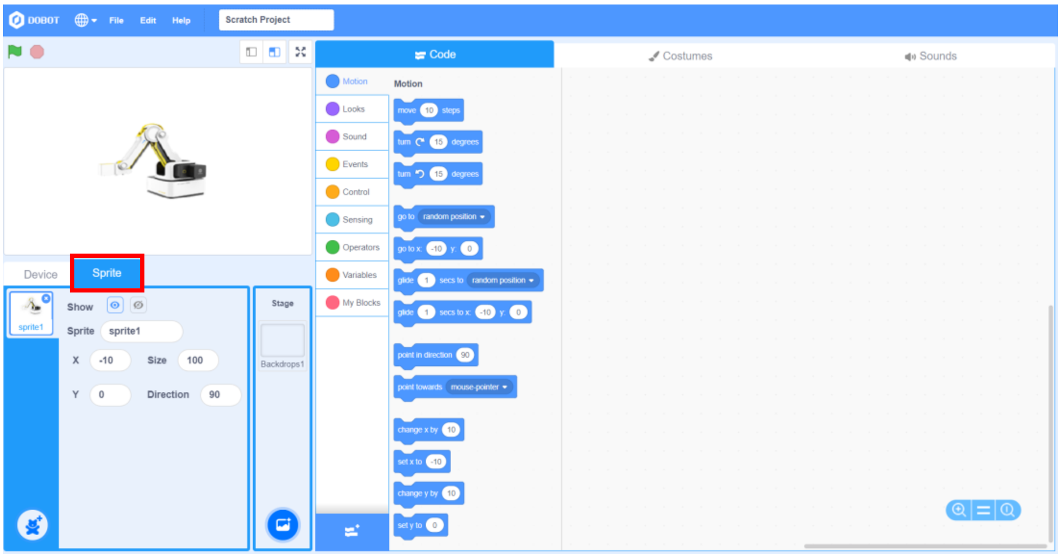
Step 3: Create lists, and name them commodities.

Step 4:

(As per the prompts, fill in the remaining steps.)

2. Steps

Step 1: Enable DobotScratch software, and switch to the **Sprite** interface from the **Device** interface, as shown in Figure 6.1.



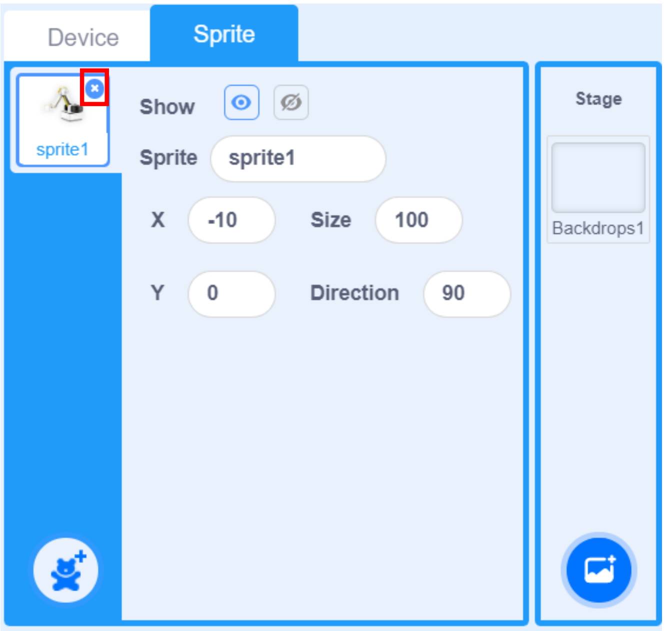
Sprite sprite1

X -10 Size 100

Y 0 Direction 90

Figure 6.1 **Sprite** interface

Step 2: Delete the default sprite "Magician Lite", as shown in Figure 6.2.



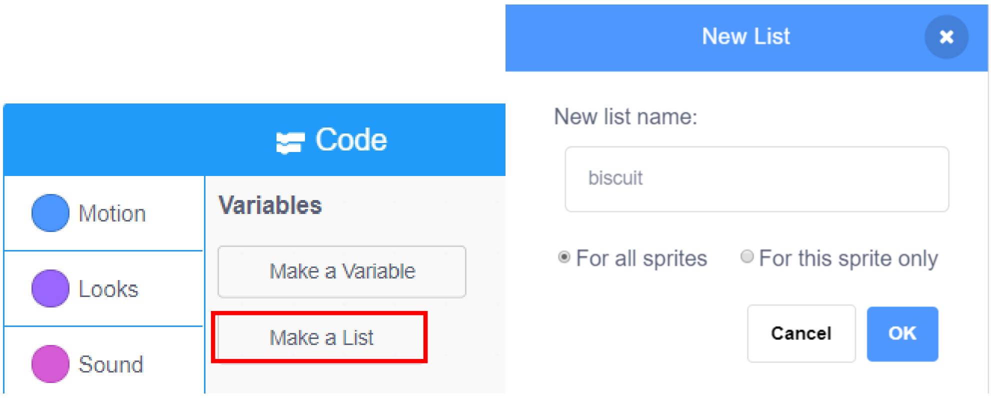
Sprite sprite1

X -10 Size 100

Y 0 Direction 90

Figure 6.2 Selecting the "City Bus" sprite

Step 3: Click the **Variables** tab, and select the "Make a List", as shown in Figure 6.3.



New list name: biscuit

● For all sprites ○ For this sprite only

Figure 6.3 Creating a list

List: It can be understood as a set of variables. The list, equivalent to a queue, is usually used in the same class of variable group.

Step 4: Name the first list *biscuit*. For changes in the stage area and the block area after the list creation, as shown in Figure 6.4.

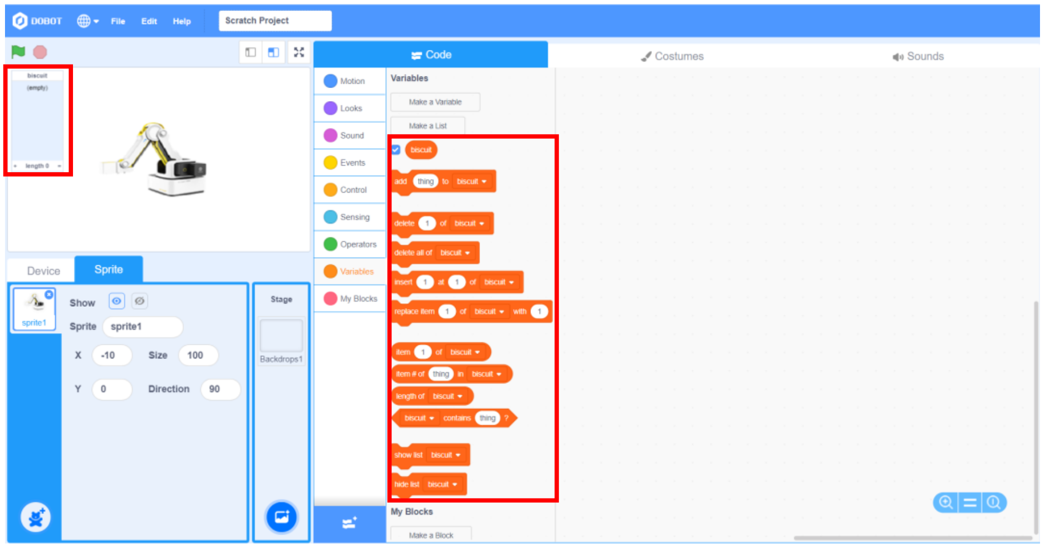
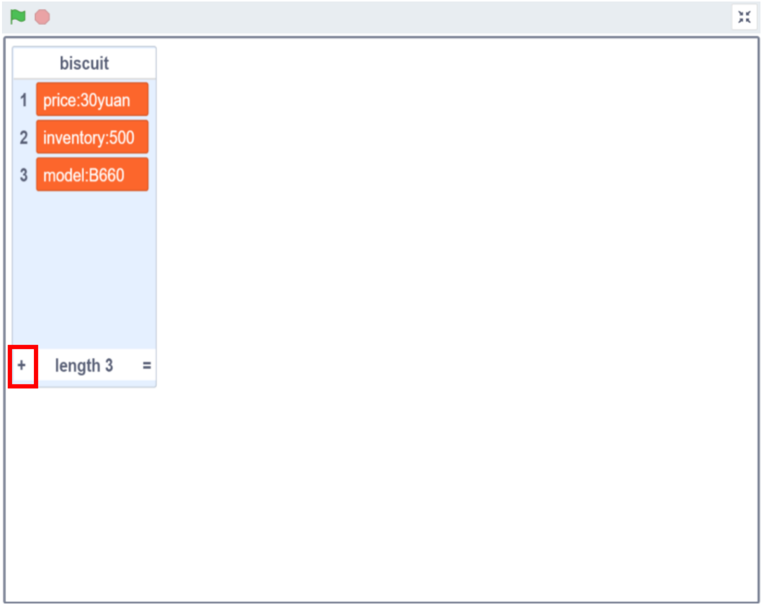


Figure 6.4 Changes in the stage area and the block area after the list creation

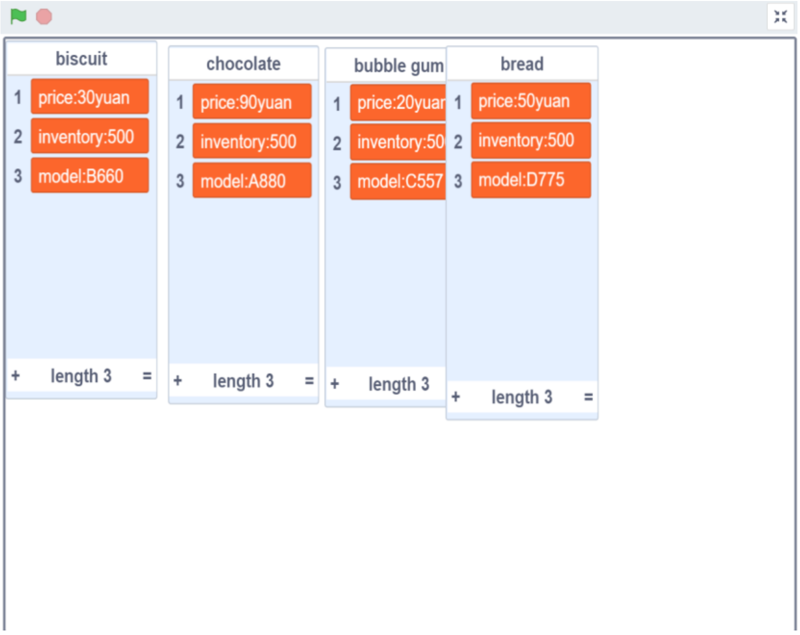
Step 5: Click the plus sign in the lower left corner of the channel list to add the commodity information, including the price, the inventory and the model, as shown in Figure 6.5.



	biscuit
1	price:30yuan
2	inventory:500
3	model:B660

Figure 6.5 Adding the commodity information

Step 6: Create another three lists in the same way, name them *chocolate*, *bubble gum* and *bread* respectively, and add the commodity information, as shown in Figure 6.6.



	biscuit	chocolate	bubble gum	bread
1	price:30yuan	price:90yuan	price:20yuar	price:50yuan
2	inventory:500	inventory:500	inventory:50	inventory:500
3	model:B660	model:A880	model:C557	model:D775

Figure 6.6 Creating three lists and adding the commodity information

Step 7: Hide the list on the stage. Click “☑" button, as shown in Figure 7.

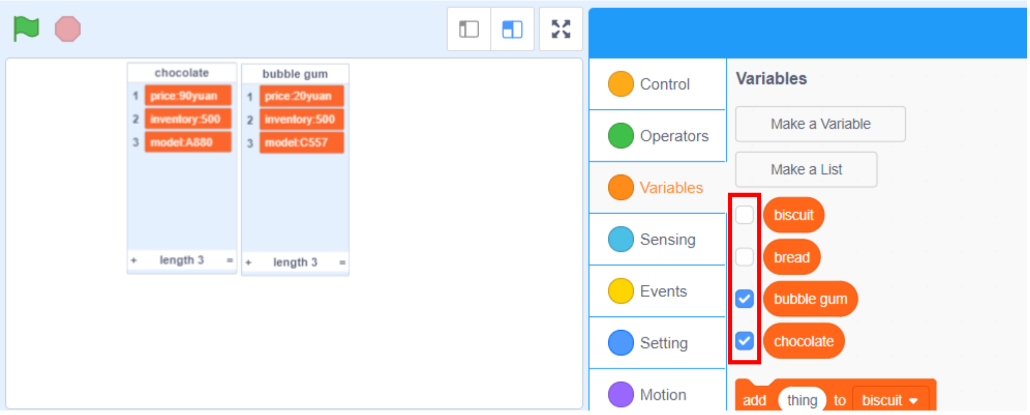


Figure 6.7 Hiding the list on the stage

Step 8: Draw the new sprite. Click the "paint" icon, as shown in Figure 8. In the drawing sprite area, click the Text icon to edit the "query of commodity information", and click the character color, as shown in Figure 6.9.

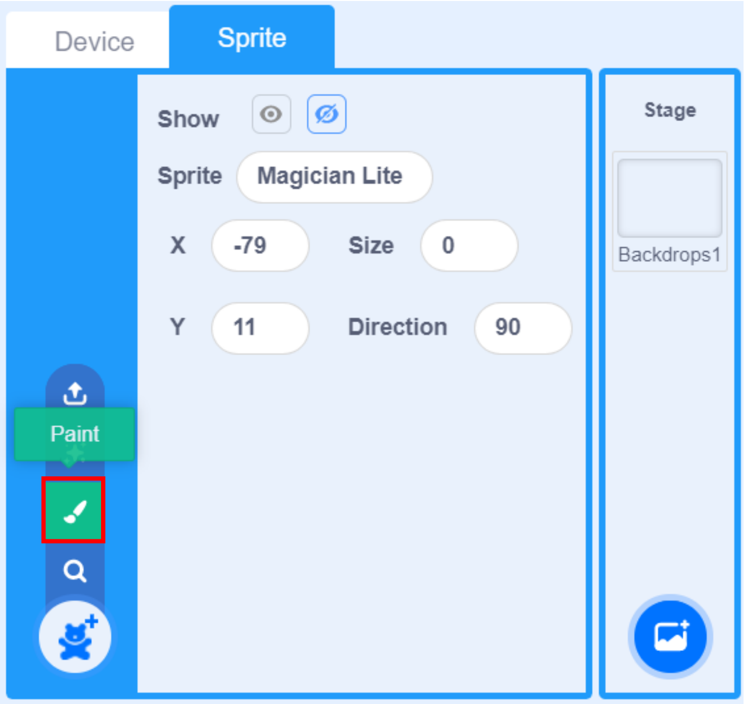


Figure 6.8 Drawing

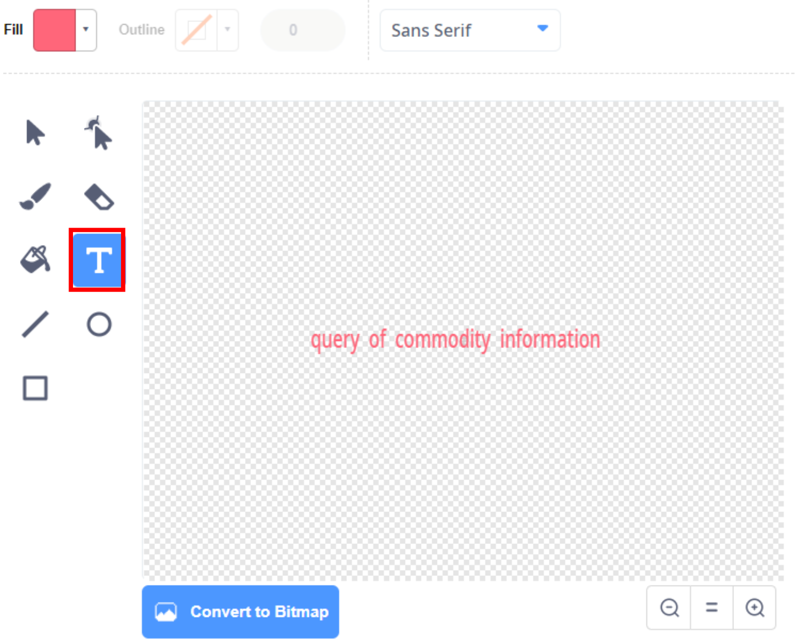


Figure 6.9 Drawing the sprite

Step 9: Set the size of "query of commodity information", as shown in Figure 6.10.

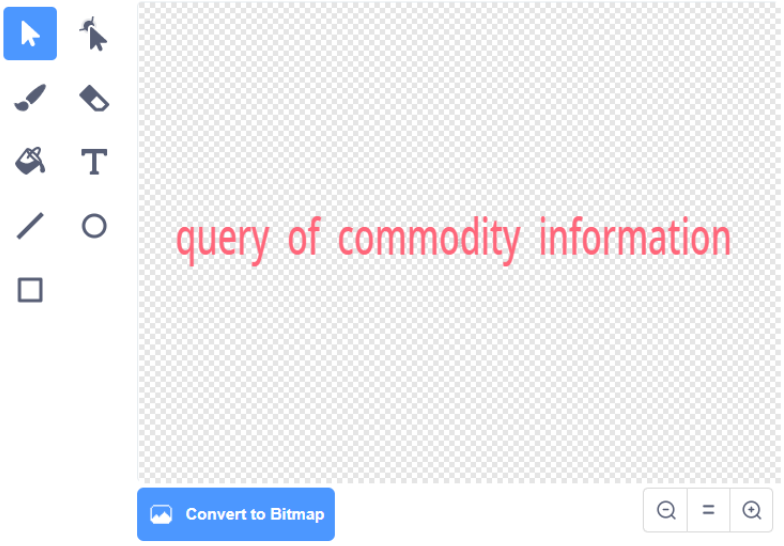
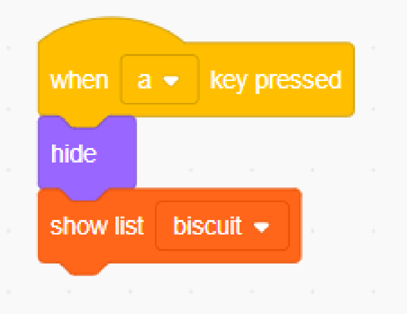


Figure 6.10 Setting the size of the sprite

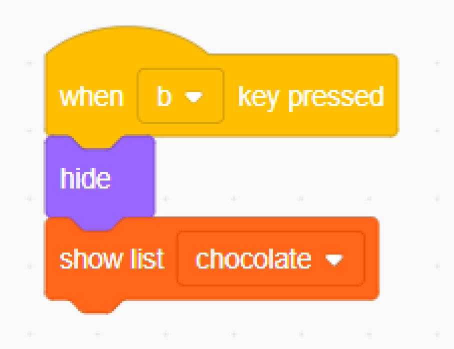
Step 10: Return to the code area, and start to write the program. As you press the **a** key, the interface hides the sprite "query of commodity information" and displays the list *biscuit*. In that case, you can query and change the biscuit information. Figure 6.11 shows the piecing mode.



```
when a key pressed
hide
show list biscuit
```

Figure 6.11 Querying the biscuit information

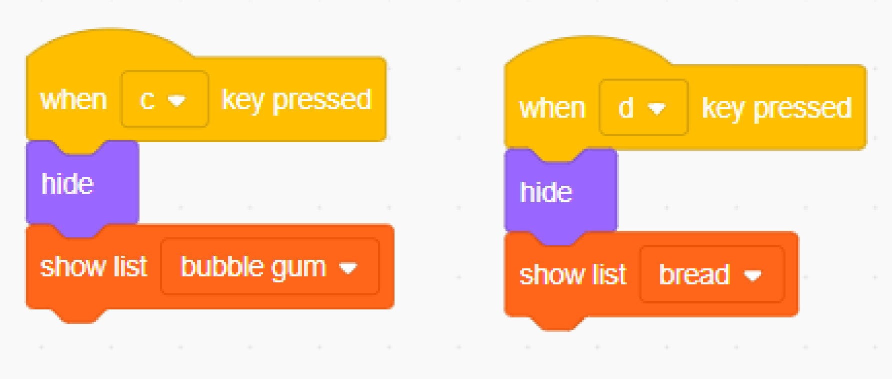
Step 11: Press the **b** key. Then, the interface hides the sprite "query of commodity information" and displays the list *chocolate*. In that case, you can query and change the chocolate information. Figure 6.12 shows the piecing mode.



```
when b key pressed
hide
show list chocolate
```

Figure 6.12 Querying the chocolate information

Step 12: In the same writing way, query and change the information on bubble gum and bread. Figure 6.13 shows the piecing mode.

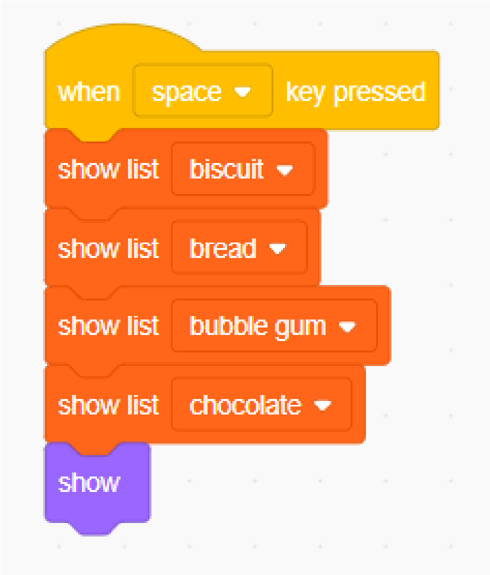


```
when c key pressed
hide
show list bubble gum
```

```
when d key pressed
hide
show list bread
```

Figure 6.13 Querying the information on bubble gum and bread

Step 13: Press the space key. Then, the interface hides all commodity information, and shows the sprite "query of commodity information". Figure 6.14 shows the piecing mode.



```
when space ▾ key pressed
show list biscuit ▾
show list bread ▾
show list bubble gum ▾
show list chocolate ▾
show
```

Figure 6.14 Showing the sprite "query of commodity information"

3. Summary

(Summarize the knowledge learned from this task)

4. Self-Assessment

Check the completed content in the experiment task. Tick (√) the completed items, and circle (◯) the uncompleted items.

Assessment Content	**Completion Status**
I’ve created a list	
I’ve created four lists	
I’ve added the commodity information	
I’ve programmed the "query of commodity information" system	

Task 2: Design the Auto Commodity Pickup

Auto commodity pickup based on the OCR technology allows the robotic arm to select the right commodities by recognizing the characters on the card.

1. Analysis

Analyze the steps to design the auto commodity pickup by the robotic arm based on the OCR technology. Then, fill in the blanks in the table below according to the prompts and your analysis.

Step 1: Prepare Hardware.

Step 2: Program: Initialize → Gather images and recognize characters → ____

____

____

______________________(As per the prompts, fill in the remaining steps.)

2. Steps

(1) Prepare Hardware

Step 1: Prepare the experiment equipment, power on the robotic arm, and connect it to the computer.

Step 2: Set the equipment to its proper position, as shown in Figure 6.17.

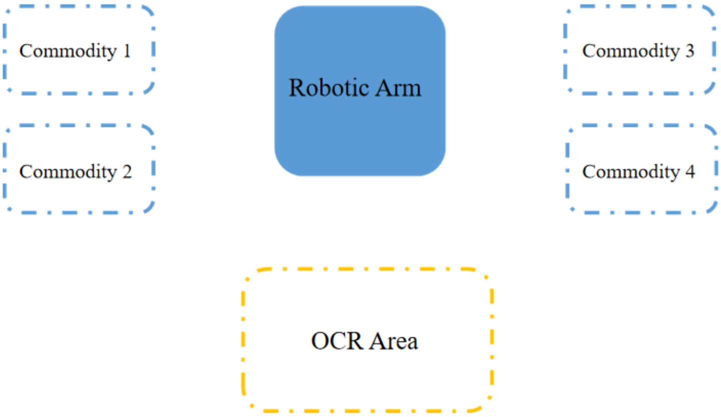
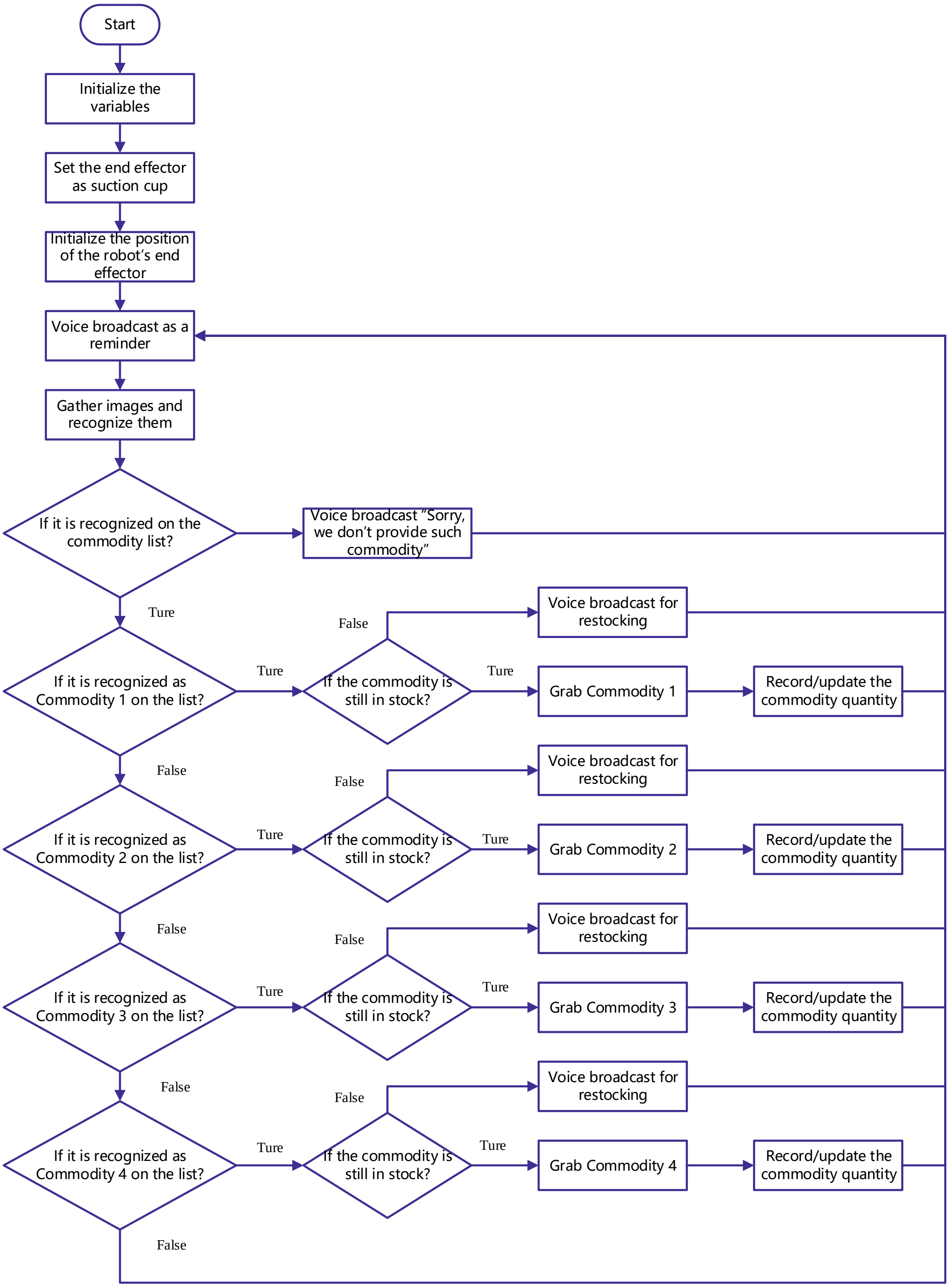


Figure 6.17 Placement position

(2) Design Program

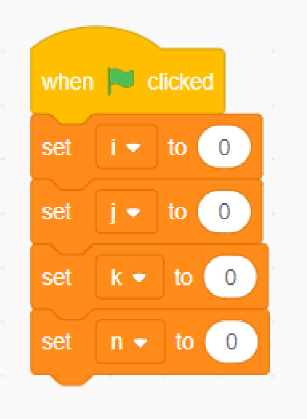
Step 1: Analyze the steps to allow the robotic arm to automatically select commodities through the OCR technology. Then, draw the program flow chart, as shown in Figure 6.18.



Start
Initialize the variables
Set the end effector as suction cup
Initialize the position of the robot's end effector
Voice broadcast as a reminder
Gather images and recognize them
If it is recognized on the commodity list?
Voice broadcast "Sorry, we don't provide such commodity"
Ture
If it is recognized as Commodity 1 on the list?
Ture
If the commodity is still in stock?
False
Voice broadcast for restocking
Ture
Grab Commodity 1
Record/update the commodity quantity
False
If it is recognized as Commodity 2 on the list?
Ture
If the commodity is still in stock?
False
Voice broadcast for restocking
Ture
Grab Commodity 2
Record/update the commodity quantity
False
If it is recognized as Commodity 3 on the list?
Ture
If the commodity is still in stock?
False
Voice broadcast for restocking
Ture
Grab Commodity 3
Record/update the commodity quantity
False
If it is recognized as Commodity 4 on the list?
Ture
If the commodity is still in stock?
False
Voice broadcast for restocking
Ture
Grab Commodity 4
Record/update the commodity quantity
False

Figure 6.18 Realization flow

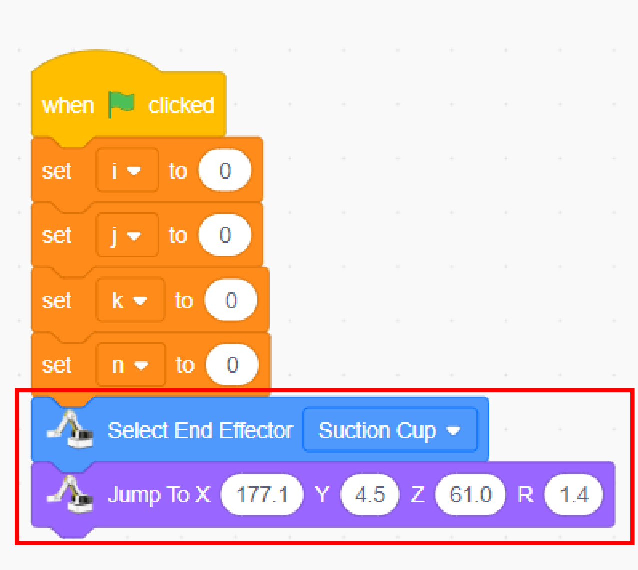
Step 2: Create four variables to represent the quantities of four kinds of commodities sold, and set each of the four variables to *0*, as shown in Figure 6.19.



```
when ⚑ clicked
set i ▾ to 0
set j ▾ to 0
set k ▾ to 0
set n ▾ to 0
```

Figure 6.19 Setting the initial values of the four variables to *0*

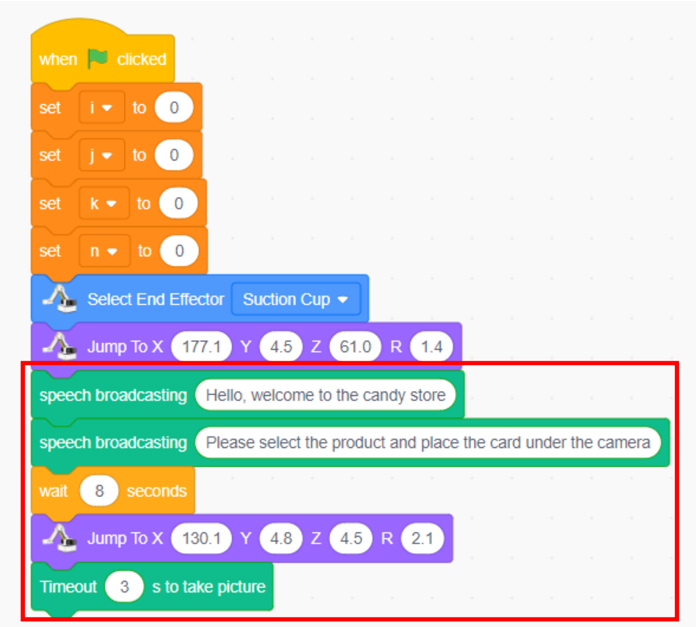
Step 3: Set the end effector as "suction cup", and initialize the position of the end of the robotic arm, as shown in Figure 6.20.



```
when ⚑ clicked
set i ▾ to 0
set j ▾ to 0
set k ▾ to 0
set n ▾ to 0
Select End Effector Suction Cup ▾
Jump To X 177.1 Y 4.5 Z 61.0 R 1.4
```

Figure 6.20 Setting the end effector and the position of the end of the robotic arm

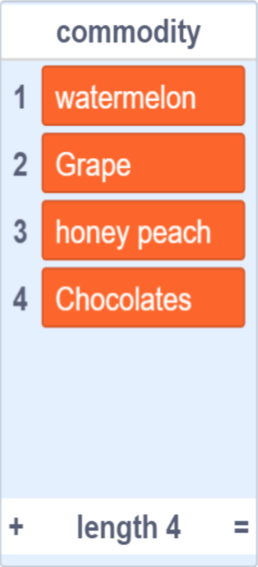
Step 4: In this experiment, allow the robotic arm to automatically select commodities through the OCR technology. Therefore, surely collect the images with characters, broadcast the speech prompt, set the photographing position and take pictures, as shown in Figure 6.21.



```
when 🏴 clicked
set i ▾ to 0
set j ▾ to 0
set k ▾ to 0
set n ▾ to 0
Select End Effector Suction Cup ▾
Jump To X 177.1 Y 4.5 Z 61.0 R 1.4
speech broadcasting Hello, welcome to the candy store
speech broadcasting Please select the product and place the card under the camera
wait 8 seconds
Jump To X 130.1 Y 4.8 Z 4.5 R 2.1
Timeout 3 s to take picture
```

Figure 6.21 Broadcasting the speech prompt, setting the photographing position and taking pictures

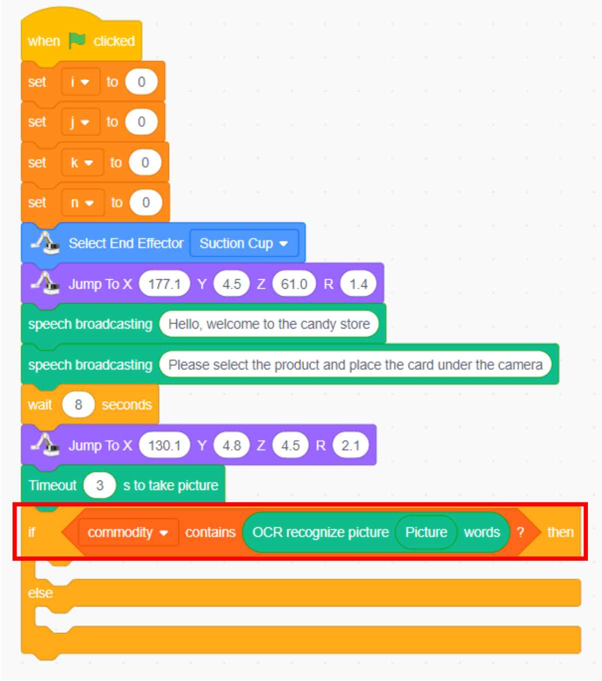
Step 5: Create a list, set the list name to *commodity*, and store the commodity name in this list, as shown in Figure 6.22.



commodity	
1	watermelon
2	Grape
3	honey peach
4	Chocolates

Figure 6.22 "Commodity" list

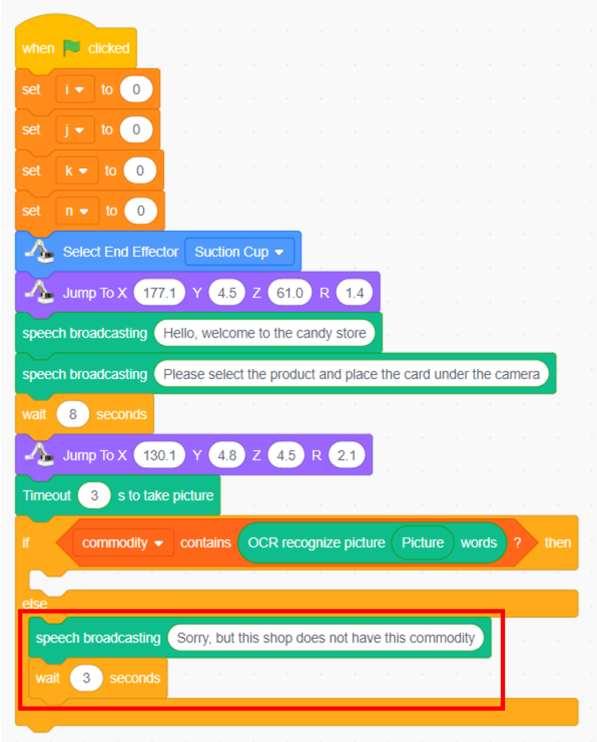
Step 6: Recognize the characters on the image, and judge whether the commodity list contains the recognized commodities, as shown in Figure 6.23.



```
when 🏳 clicked
set i ▾ to 0
set j ▾ to 0
set k ▾ to 0
set n ▾ to 0
Select End Effector Suction Cup ▾
Jump To X 177.1 Y 4.5 Z 61.0 R 1.4
speech broadcasting Hello, welcome to the candy store
speech broadcasting Please select the product and place the card under the camera
wait 8 seconds
Jump To X 130.1 Y 4.8 Z 4.5 R 2.1
Timeout 3 s to take picture
if commodity ▾ contains OCR recognize picture Picture words ? then
else
```

Figure 6.23 Setting the judgment condition

Step 7: Find the name of a recognized commodity in the list. This indicates that the commodity is available for sale in the candy store. In that case, you can sell it by taking it by the robotic arm. Otherwise, it is unavailable in the store, and the system broadcasts the speech message "Sorry, but this shop does not have this commodity". Figure 6.24 shows the programming mode.

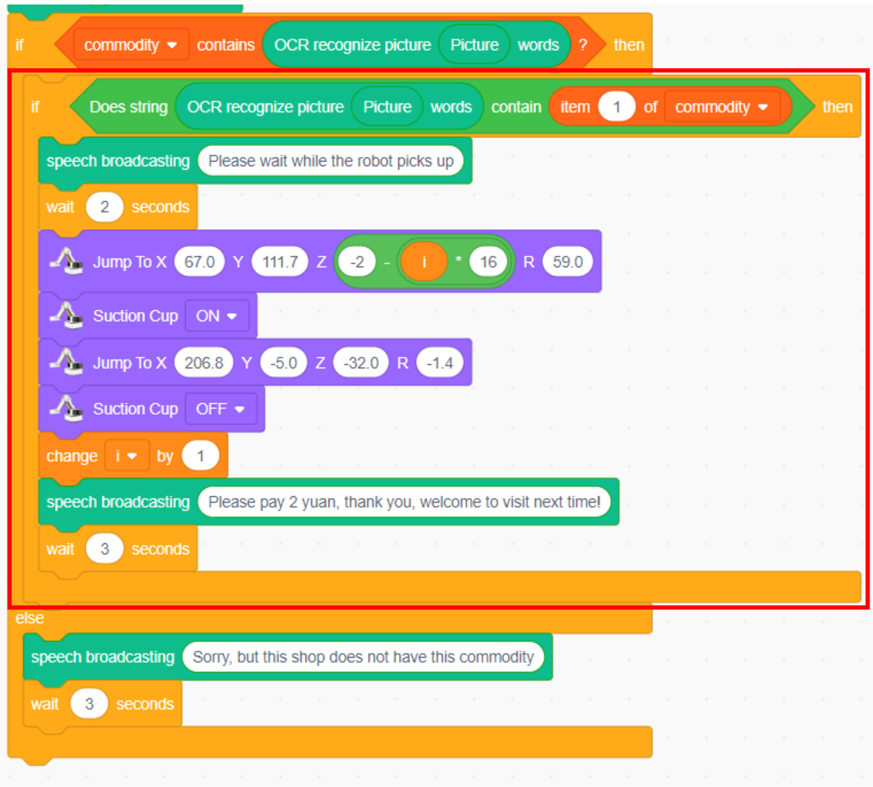


```
when 🏳 clicked
set i ▾ to 0
set j ▾ to 0
set k ▾ to 0
set n ▾ to 0
Select End Effector Suction Cup ▾
Jump To X 177.1 Y 4.5 Z 61.0 R 1.4
speech broadcasting Hello, welcome to the candy store
speech broadcasting Please select the product and place the card under the camera
wait 8 seconds
Jump To X 130.1 Y 4.8 Z 4.5 R 2.1
Timeout 3 s to take picture
if commodity ▾ contains OCR recognize picture Picture words ? then
else
    speech broadcasting Sorry, but this shop does not have this commodity
    wait 3 seconds
```

Figure 6.24 Broadcasting the speech message "Sorry, this shop does not have this

commodity"

Step 8: Have four cases if the list contains the recognized commodity. Case 1: If the recognized result involves commodity 1 in the list, the robotic arm grabs it, as shown in Figure 6.25.

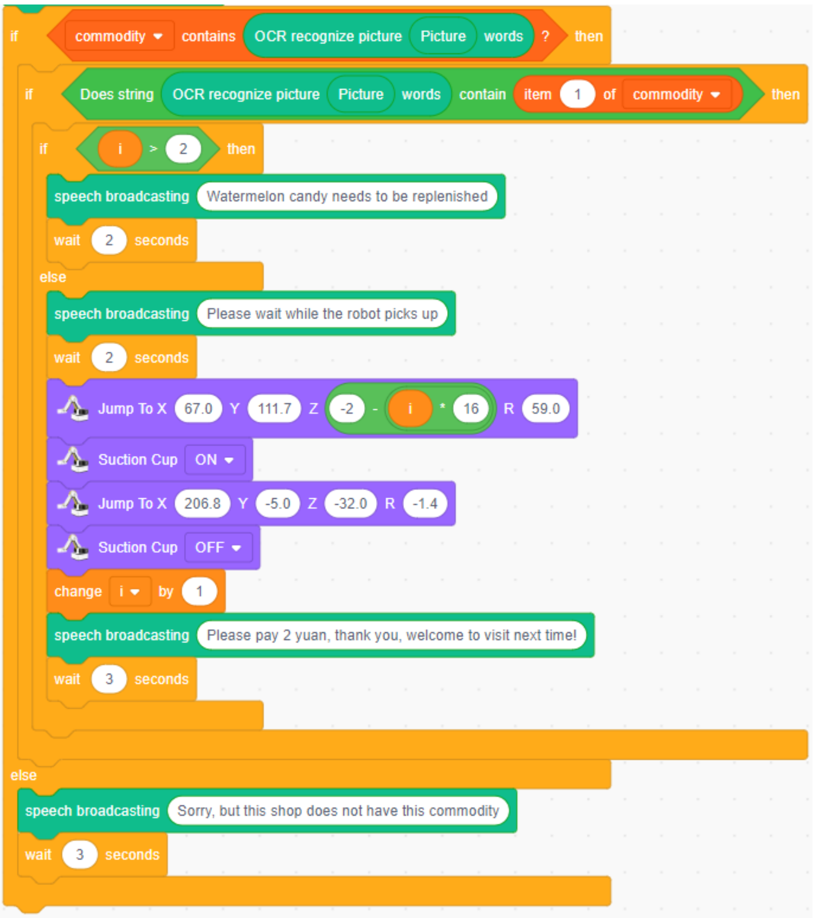


```
if commodity contains OCR recognize picture Picture words ? then
    if Does string OCR recognize picture Picture words contain item 1 of commodity then
        speech broadcasting Please wait while the robot picks up
        wait 2 seconds
        Jump To X 67.0 Y 111.7 Z -2 - i * 16 R 59.0
        Suction Cup ON
        Jump To X 206.8 Y -5.0 Z -32.0 R -1.4
        Suction Cup OFF
        change i by 1
        speech broadcasting Please pay 2 yuan, thank you, welcome to visit next time!
        wait 3 seconds
else
    speech broadcasting Sorry, but this shop does not have this commodity
    wait 3 seconds
```

Figure 6.25 Case 1

Here, insert "if…then" into the "if…then…else" block. This is a nested branch structure. Do you have any questions about the variable *i*? For example, how does "-2-i x16" appear? Why do we "increase i by 1"? Next we will answer these questions.

Step 9: Observe Figure 6.26. Add the "if…then…else" block to judge whether the variable *i* is greater than 2 (supposing that here are only three pieces of commodity 1). If yes, by default, the commodity is sold up, with a speech message being "Watermelon candy needs to be replenished". Otherwise, commodity 1 (watermelon candy) is still available, and can continue to be sold.

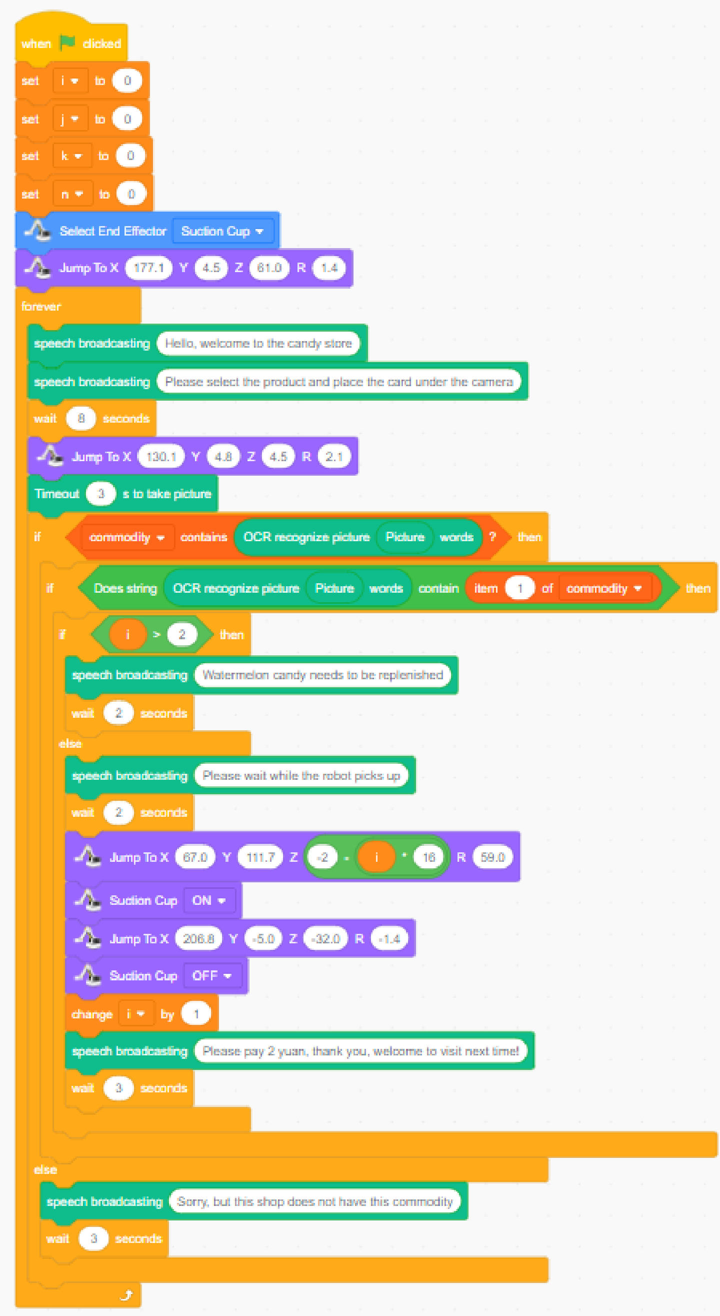


```
if commodity contains OCR recognize picture Picture words ? then
    if Does string OCR recognize picture Picture words contain item 1 of commodity then
        if i > 2 then
            speech broadcasting Watermelon candy needs to be replenished
            wait 2 seconds
        else
            speech broadcasting Please wait while the robot picks up
            wait 2 seconds
            Jump To X 67.0 Y 111.7 Z -2 - i * 16 R 59.0
            Suction Cup ON
            Jump To X 206.8 Y -5.0 Z -32.0 R -1.4
            Suction Cup OFF
            change i by 1
            speech broadcasting Please pay 2 yuan, thank you, welcome to visit next time!
            wait 3 seconds
else
    speech broadcasting Sorry, but this shop does not have this commodity
    wait 3 seconds
```

Figure 6.26 Processing method for the OCR recognition result being commodity 1

From step 9, we know the variable *i* denotes the quantity of commodity 1 sold. If there are a total of three pieces of commodity 1 and the initial value of *i* is 0, "-2- i × 16" becomes the calculation formula of the Z coordinate of commodity 1(wherein 16 denotes the height of the commodity). With the sale of a piece, *i* increases by 1. As *i* is 2, this calculation result corresponds to the final piece; as *i* increases to 3, the system broadcasts th speech message that the shop needs to replenish goods.

Step 10: Add the repetitively executed blocks, and Figure 6.27 shows the programming mode.



```
when 🏴 clicked
set i ▾ to 0
set j ▾ to 0
set k ▾ to 0
set n ▾ to 0
Select End Effector Suction Cup ▾
Jump To X 177.1 Y 4.5 Z 61.0 R 1.4
forever
    speech broadcasting Hello, welcome to the candy store
    speech broadcasting Please select the product and place the card under the camera
    wait 8 seconds
    Jump To X 130.1 Y 4.8 Z 4.5 R 2.1
    Timeout 3 s to take picture
    if commodity ▾ contains OCR recognize picture Picture words ? then
        if Does string OCR recognize picture Picture words contain item 1 of commodity ▾ then
            if i > 2 then
                speech broadcasting Watermelon candy needs to be replenished
                wait 2 seconds
            else
                speech broadcasting Please wait while the robot picks up
                wait 2 seconds
                Jump To X 67.0 Y 111.7 Z -2 - i * 16 R 59.0
                Suction Cup ON ▾
                Jump To X 206.8 Y -5.0 Z -32.0 R -1.4
                Suction Cup OFF ▾
                change i ▾ by 1
                speech broadcasting Please pay 2 yuan, thank you, welcome to visit next time!
                wait 3 seconds
    else
        speech broadcasting Sorry, but this shop does not have this commodity
        wait 3 seconds
```

Figure 6.27 Repeating the execution

☛ The above program is to solve the case where the recognized result involves commodity 1. For other three cases, you can independently program them in the same way. Note that different commodities have different X, Y and Z coordinates. So we need to fix the point once again and change the calculation formula of Z coordinate "--2-i x 16".

3. Summary

(Summarize the knowledge learned from this task)

4. Self-Assessment

Check the completed content in the experiment task. Tick (√) the completed items, and circle (◯) the uncompleted items.

Assessment Content	Completion Status
I’ve initialized the "auto commodity pickup” system	
I’ve designed the speech broadcasting message of the "auto commodity pickup" system through programming	
I’ve collected and recognized the image in the "auto commodity pickup" system through programming	
I’ve judged the OCR recognition result	
I’ve programmed the entire "auto commodity pickup" system	

Experiment 7

Auto-Medicine Vending Robot

Team Name: Team member: Date:

Overview

In this experiment we simulate the auto-medicine vending robot system in smart mall. Through this system, users can choose and purchase medicines themselves (users have four choices in this experiment: cold medicines, fever medicines, headache medicines and anti-inflammafory medicines) from the computer screen. Then, the robot picks up medicines, and users pay through face recognition payment function. From this experiment, students will experience the comprehensive application of face recognition, human-computer interaction and communication technologies.

Objective

1． Understand the method of drawing the sprite by making the operation interface of auto-medicine vending robot.

2． By making the operation interface of auto-medicine vending robot, understand how to make the stage backdrops.

3． Know the "broadcast message" instruction well by programming auto-medicine vending robot.

4． By programming auto-medicine vending robot, experience the communication technology between the robot and the computer.

5． By programming auto-medicine vending robot, experience the application of the face recognition technology.

Equipment

Equipment Picture	Name	Quantity
	Dobot Magician Lite Robotic arm	1
	USB Type-C interface cable	1
	Suction cup	1
	Power adapter	1
	Camera	1
	Medicine box	1

Requirements

- Take care when using electricity.
- Before the experiment, check whether the experimental equipment is complete and intact. If there is any omission or damage, please report to the teacher.

- Any specific operations in the experiment shall be performed according to the experiment manual. If you have any questions, please promptly ask the teacher.
- During the experiment, the joints will start to work as the robotic arms are powered on. In that case, do not move the joints of the robotic arms hard if you do not press the unlock key.
- Report any equipment fault during the experiment to your teacher in time. Do not handle it by yourself.
- Arrange all devices after the experiment. You shall not leave the lab before check by the group leader.

Task 1: Make the User Operation Interface

If customers want to purchase medicines from a store with auto-medicine vending robot, the store will provide an operation platform. Thus, customers can select medicines and pay. Next, let’s make the user operation interface.

5. Observation

Observe the user operation interface demonstrated by the teacher, and analyze which Sprites and elements you need in making such interface:

The Sprites and elements of the user operator interface include: medicine picture, medicine name, commodity selection button, and so on.

6. Steps

Step 1: Enable DobotScratch software, click **Sprite**, and delete the Magician Lite Sprite on the stage. Click **Stage**, and select **Backdrops** to draw the backdrops, as shown in Figure 7.1.

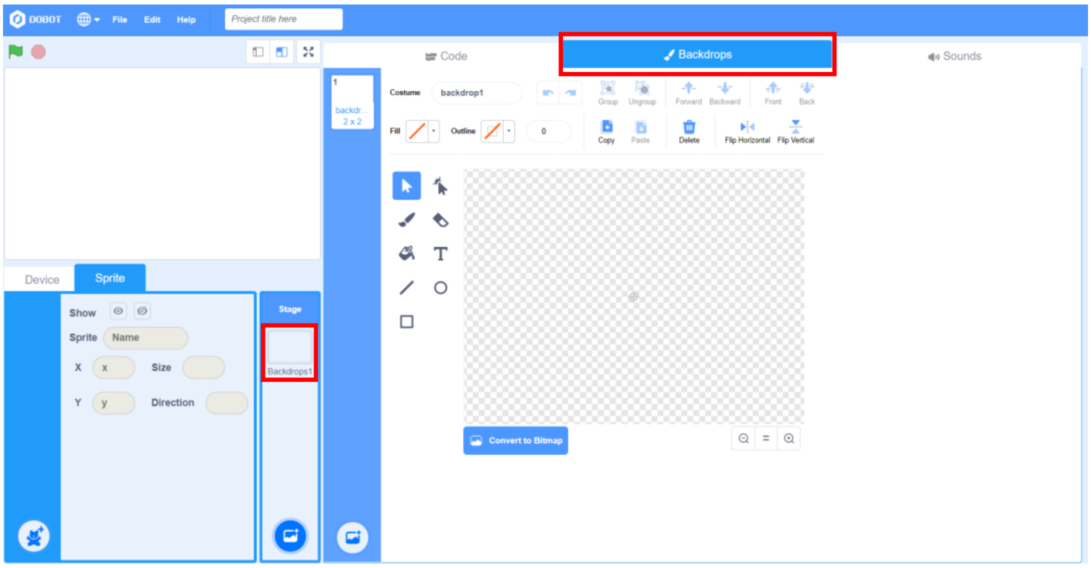


Figure 7.1 Selecting the stage backdrops

Step 2: Draw the medicine label in the stage backdrops. Click the text icon in the toolbar, and enter "cold medicine" on the stage, as shown in Figure 7.2.

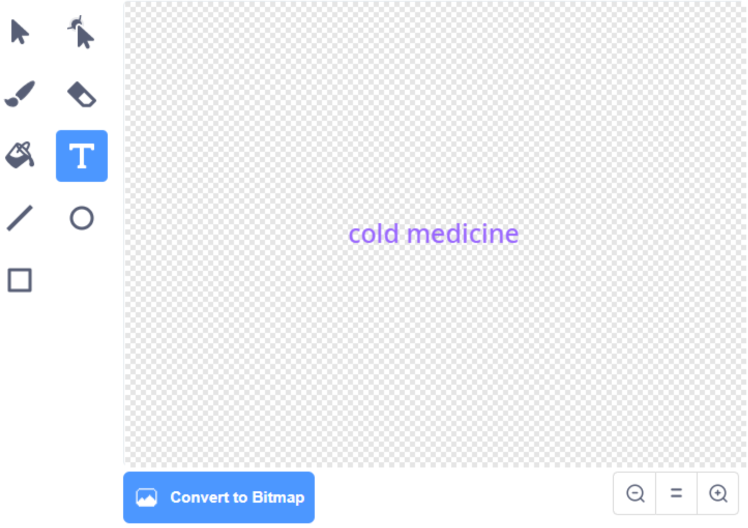


Figure 7.2 Drawing the medicine label

Step 3: Repeat the same step to draw "fever medicine", "headache medicine" and "anti-inflammatory medicine" labels, as shown in Figure 7.3.

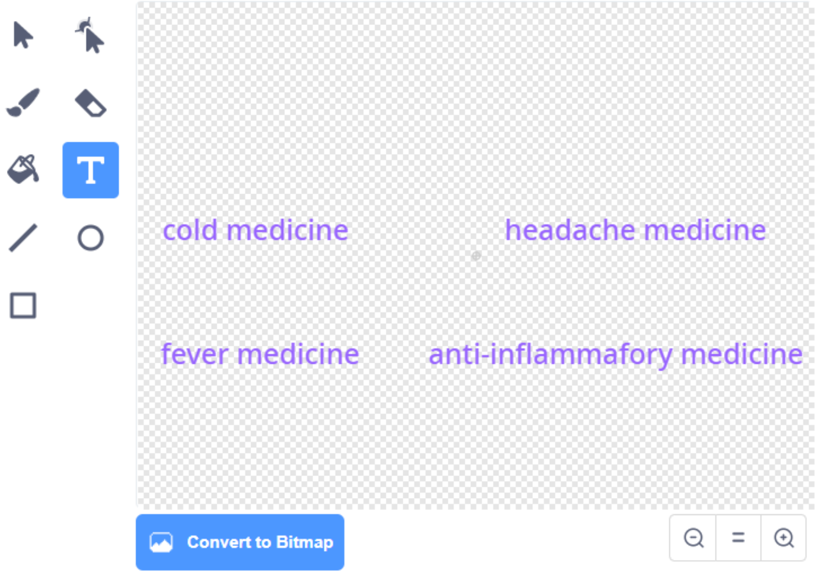


Figure 7.3 Drawing other medicine labels

Step 4: Click the selection icon in the toolbar, and then click and hold the medicine label to drag it to a proper position. Note that sufficient space needs to be left for placing a picture under each label, as shown in Figure 7.4.

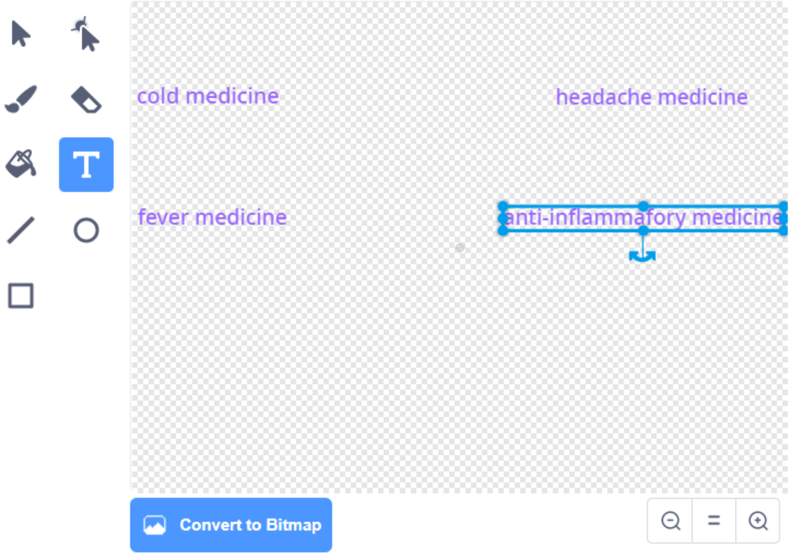


Figure 7.4 Dragging the medicine label

Step 5: Draw the label that displays the total cost of the medicines purchased, as shown in Figure 7.5.

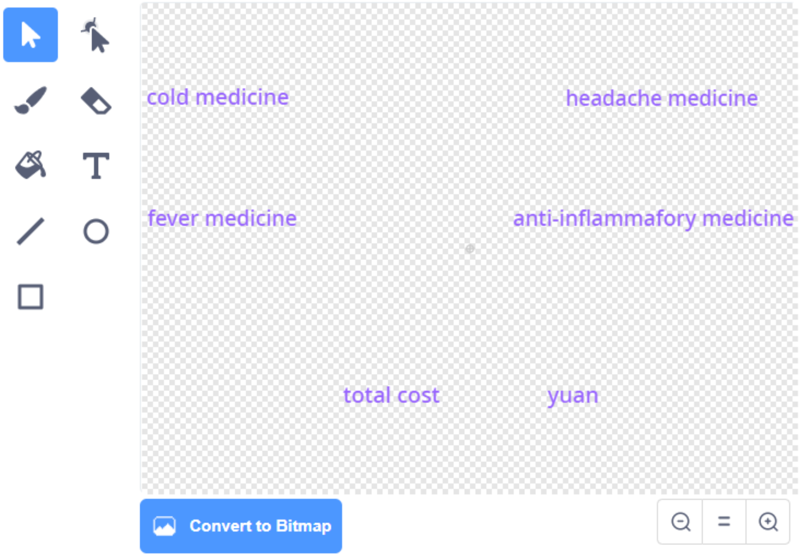
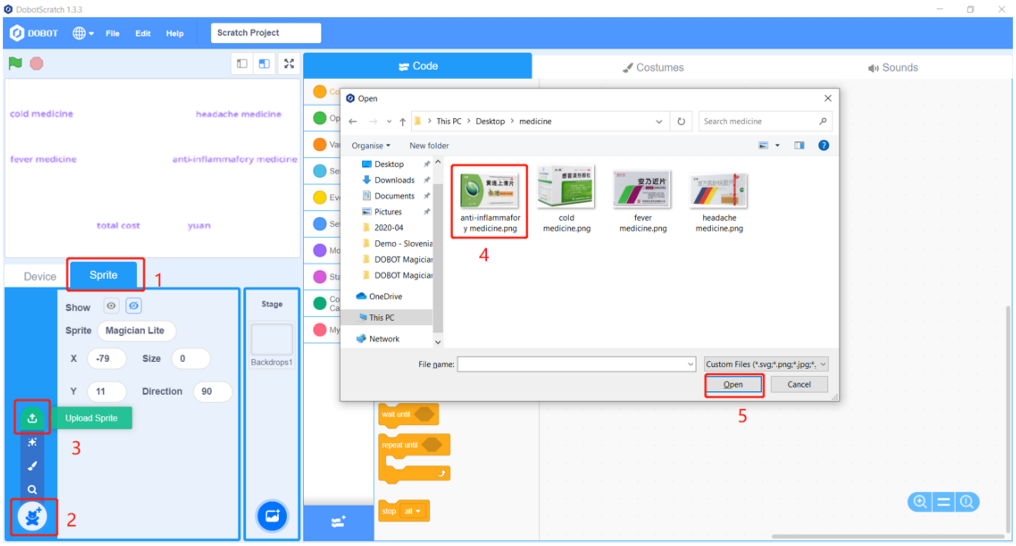


Figure 7.5 Drawing the label of total cost

Step 6: Download the pictures of four medicines from Internet, and put them in a folder, as shown in Figure 7.6.

Figure 7.6 Downloading the medicine pictures

Step 7: Upload the medicine pictures. Click **Sprite**, move the cursor to the little bear icon in the lower left corner, and choose **Upload Sprite**. Find the folder for storage of medicine pictures, select and open a picture inside. Then, upload it, as shown in Figure 7.7.



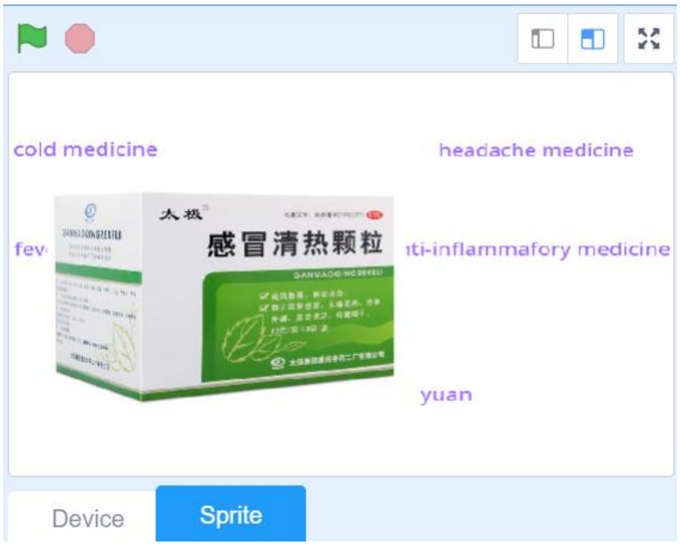
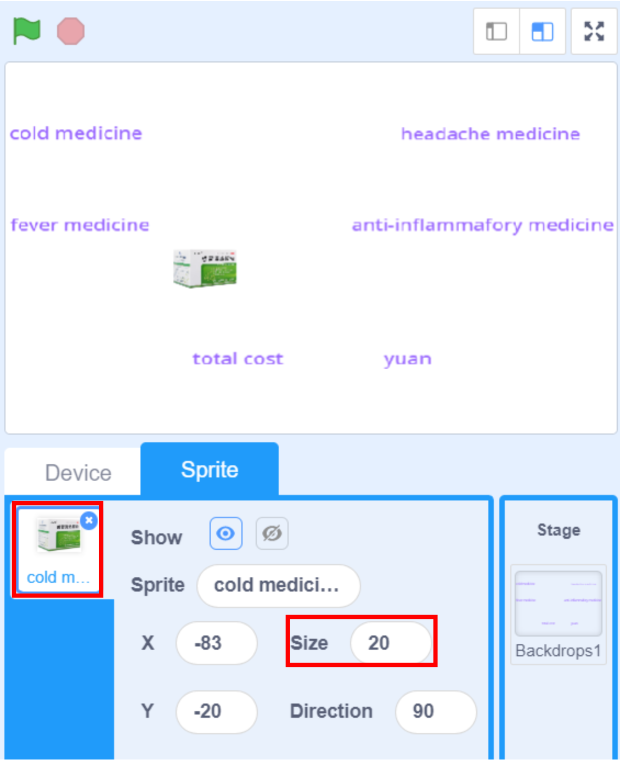


Figure 7.7 Uploading the medicine pictures

Step 8: Set the size of the sprite. Select the cold medicine from the sprite list, and set the size to **15**, as shown in Figure 7.8.



Show
Sprite cold medici...
X -83
Size 20
Y -20
Direction 90

Figure 7.8 Setting the size of the sprite

Step 9: Drag the sprite on the stage. Click and hold the cold medicine Sprite on the stage to drag it to a position under the cold medicine label, as shown in Figure 7.9.

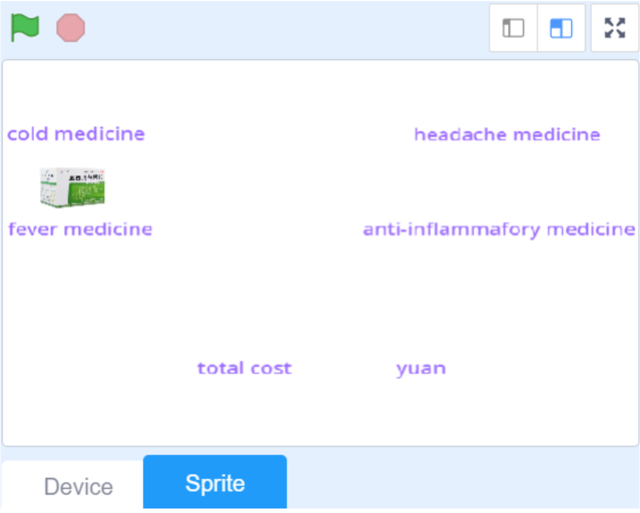
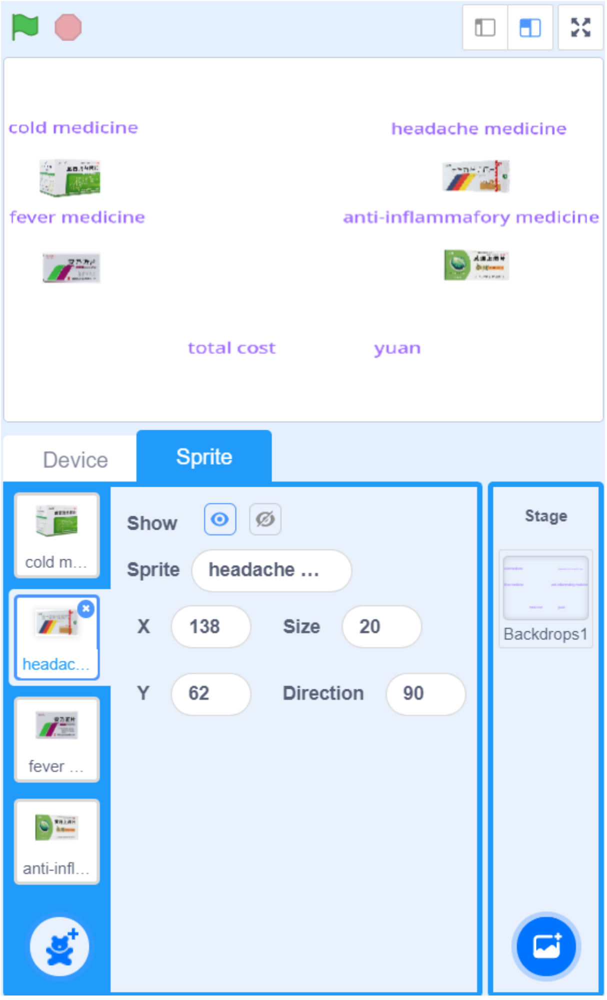


Figure 7.9 Dragging the cold medicine sprite

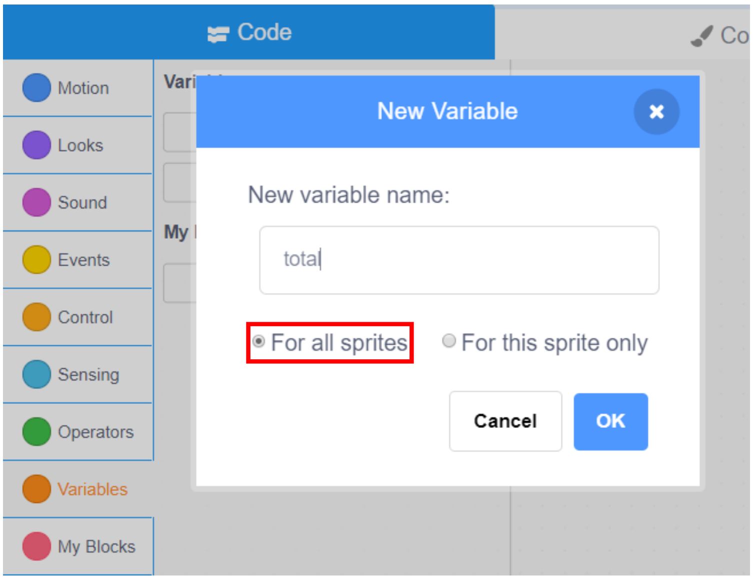
Step 10: Repeat the above steps to upload the pictures of the other three medicines and drag them to positions under their respective medicine labels, as shown in Figure 7.10.



Show
Sprite: headache ...
X: 138
Size: 20
Y: 62
Direction: 90

Figure 7.10 Uploading the pictures of other medicines

Step 11: Define the variable *total* to represent the total cost of the medicines purchased, and apply it to all sprites, as shown in Figure 7.11.



New Variable

New variable name: total

● For all sprites ○ For this sprite only

Cancel OK

Figure 7.11 Defining the variable *total*

Step 12: Hide the name of the variable on the stage. Right-click the variable *total* display box on the stage, and select the large readout option, as shown in Figure 7.12.

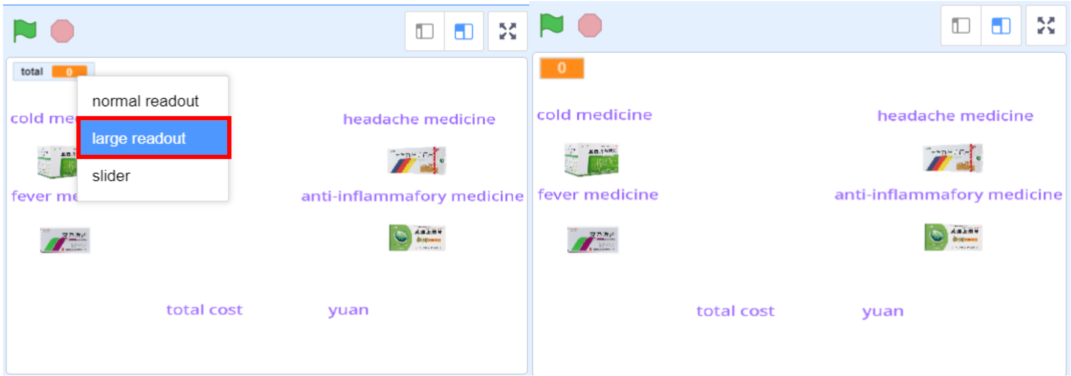


Figure 7.12 Hiding the name of the variable

Step 13: Move the variable display box to the cost label, as shown in Figure 7.13.

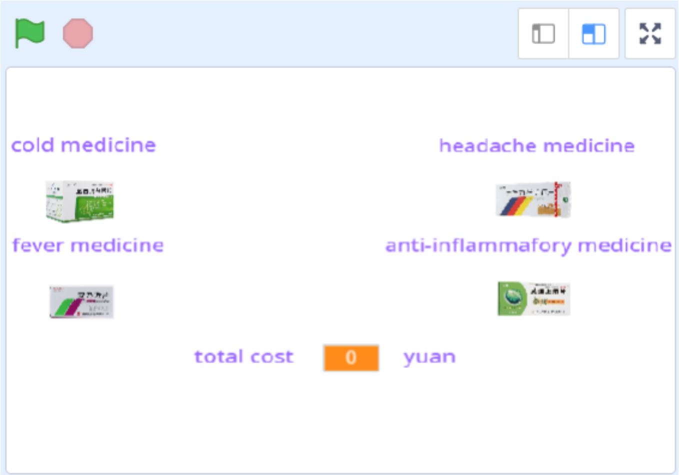


Figure 7.13 Moving the variable display box

Step 14: Make the **Select Commodity** functional button. Click the little bear icon in the lower left corner, and select the drawing icon, as shown in Figure 7.14.

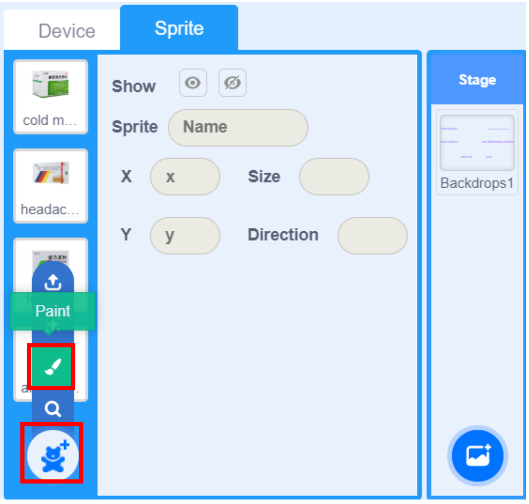


Figure 7.14 Drawing the new sprite

Step 15: Click the Circle option in the toolbar to draw an ellipse button, as shown in Figure 7.15.

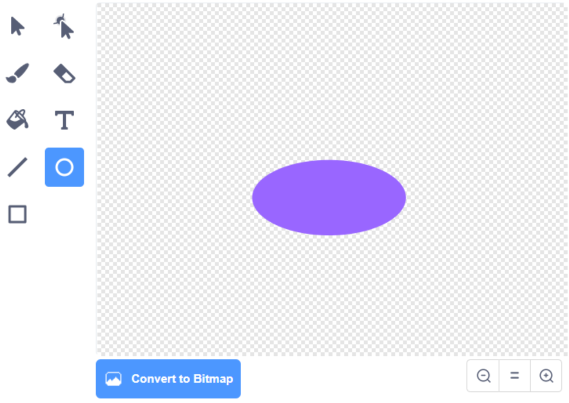


Figure 7.15 Drawing the ellipse button

Step 16: Click the text option, enter "select commodity" on the ellipse button, and fill the font with black, as shown in Figure 7.16.

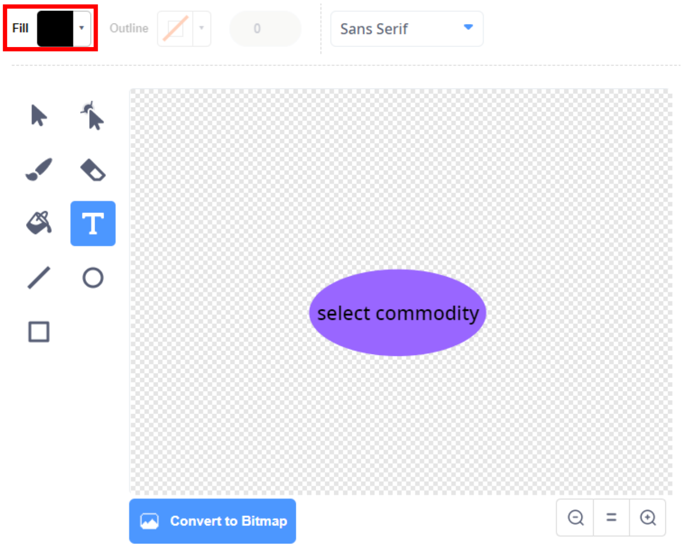
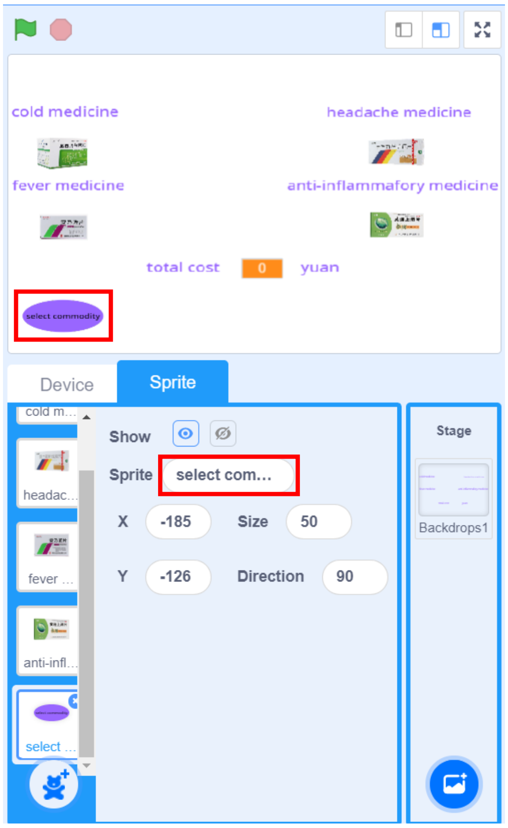


Figure 7.16 Drawing the "select commodity"

Step 17: Click and drag the "select commodity" button on the stage to a proper position, and change the sprite name of the button to "select commodity", as shown in Figure 7.17.



Show

Sprite select com...

X -185 Size 50

Y -126 Direction 90

Figure 7.17 Drawing the button name

Step 18: Repeat the same steps to make a **Checkout** functional button, as shown in Figure 7.18.

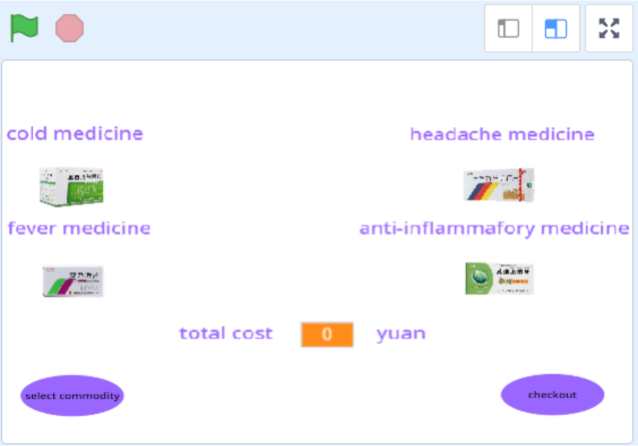


Figure 7.18 Drawing the "checkout" button

Step 19: Save the program, and name it "Auto-Medicine Vending Robot.sb3".

7. Conclusion

(1) The steps to draw the stage backdrops: ____________________

____

(2) The steps to draw the new sprite: _________________________

____

(3) How do we modify the position and size of the sprite on the stage:

____

8. Self-Assessment

Check the completed content in the experiment task. Tick (√) the completed items, and circle (◯) the uncompleted items.

Assessment Content	Completion Status
I’ve known how to modify the position and size of the sprite	
I’ve made the user operation interface of the Auto-Medicine Vending Robot system.	

Task 2: Compile the Script of Each Sprite

After designing the user operation interface, we continue to consider how each sprite is logically related to each element in the interface. Next, let’s program each sprite and element in the operation interface.

1. Observation

Analyze the steps to program each sprite as per the functional requirements of the Auto-Medicine Vending Robot system.

When the "select commodity" Sprite is clicked, the Auto-Medicine Vending Robot system should: ______________________________

When the "cold medicine" Sprite is clicked, the Auto-Medicine Vending Robot system should: ______________________________

When the "checkout" Sprite is clicked, the Auto-Medicine Vending Robot system should: ______________________________

2. Steps

Step 1: Compile the script of the "select commodity" sprite. Compile the flow chart, as shown in Figure 7.19.

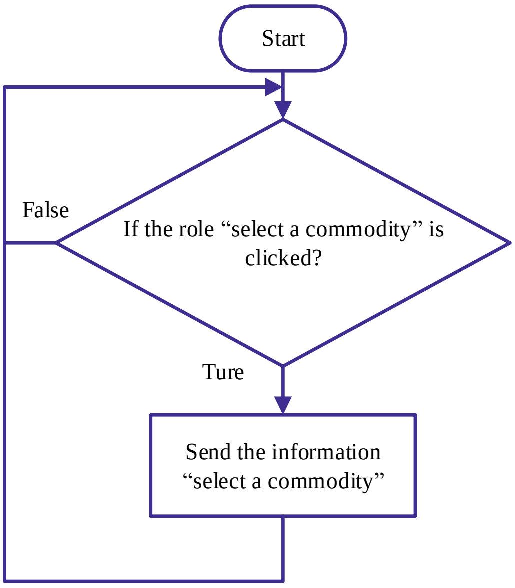
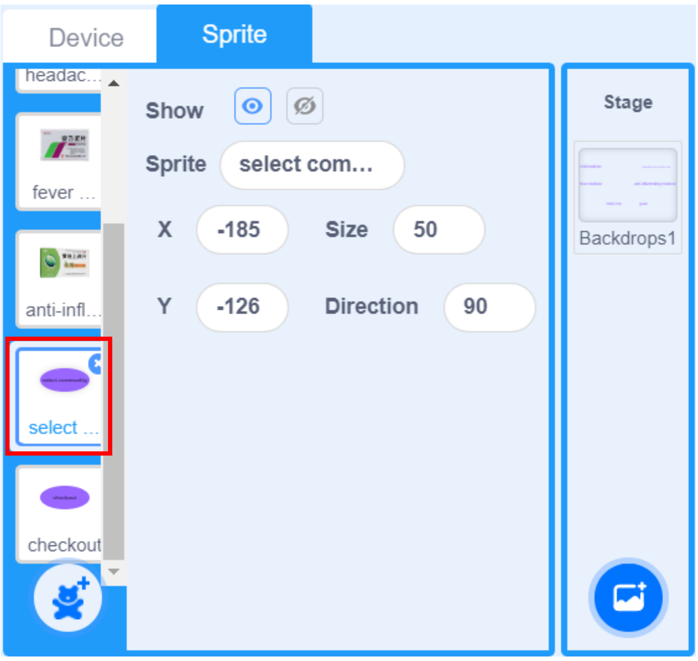

Start
False
If the role “select a commodity” is clicked?
Ture
Send the information “select a commodity”

Figure 7.19 Script flow of the "select commodity" sprite

Step 2: Open the "Auto-Medicine Vending Robot.sb3" file, and click the "select commodity" sprite in the sprite list to compile the script, as shown in Figure 7.20.



Show

Sprite select com...

X -185 Size 50

Y -126 Direction 90

Figure 7.20 Selecting the "select commodity" sprite

Step 3: Judge whether the "select commodity" is clicked. Click the **Events** tab, drag the "when this sprite clicked" block to the code area, as shown in Figure 7.21.

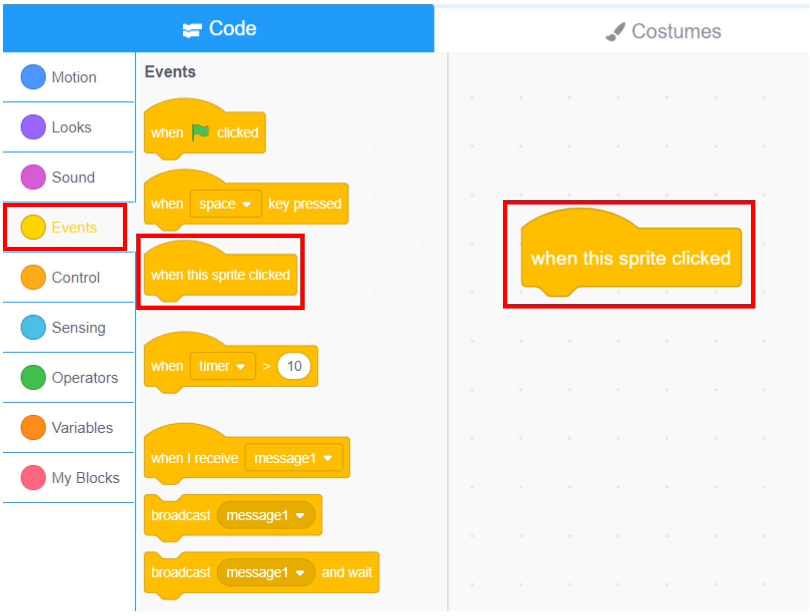


Figure 7.21 Judging whether the sprite is clicked

Step 4: Click the **Events** tab, and draw "broadcast message" to the code area, as shown in Figure 7.22.

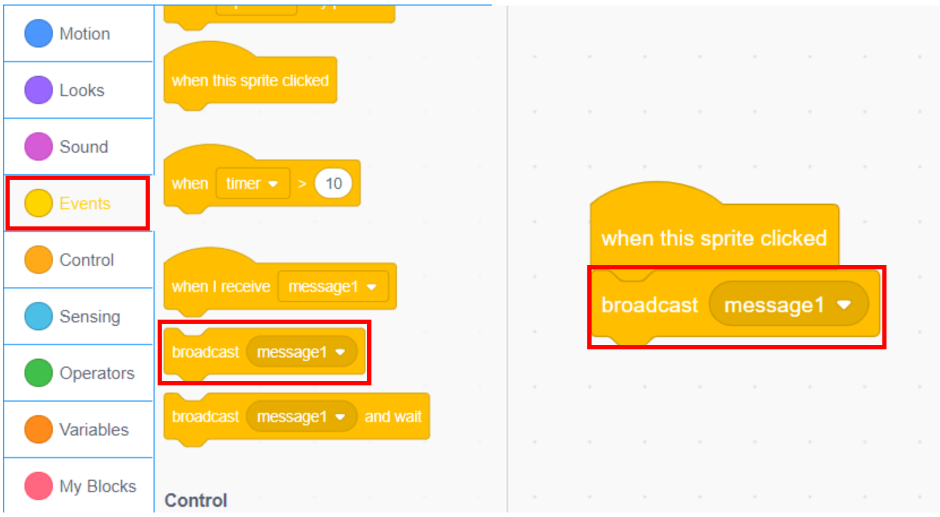


Figure 7.22 Broadcast message instruction

Step 5: Compile the message to be broadcast as the "select commodity" sprite is clicked. Click the option box in the "broadcast message" block, select the new message, and write its name as "select commodity", as shown in Figure 7.23.

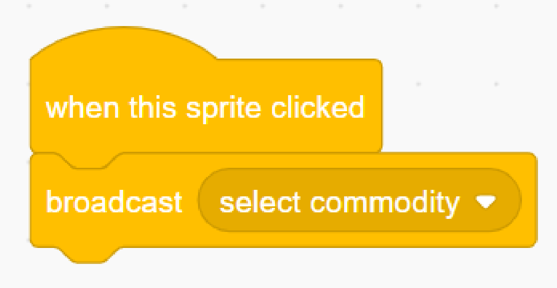


Figure 7.23 Creating the broadcast message

Step 6: Broadcast the "checkout" information by the "checkout" sprite when you click the "checkout" button, as shown in Figure 7.24.

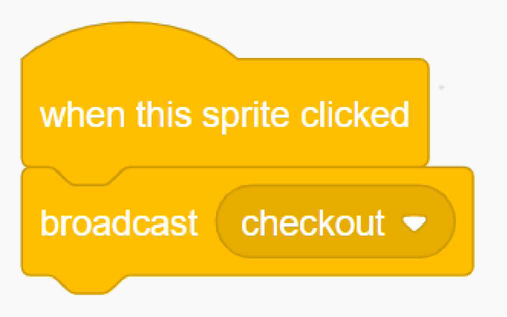


Figure 7.24 Script of the "checkout" sprite

Step 7: Compile the script of the "cold medicine" sprite. Only when receiving the "select commodity" message, users can select the cold medicine sprite. Select the "cold medicine" sprite, click the **Events** tab, drag the "when I receive..." block to the code area, and set the received message to "select commodity", as shown in Figure 7.25.

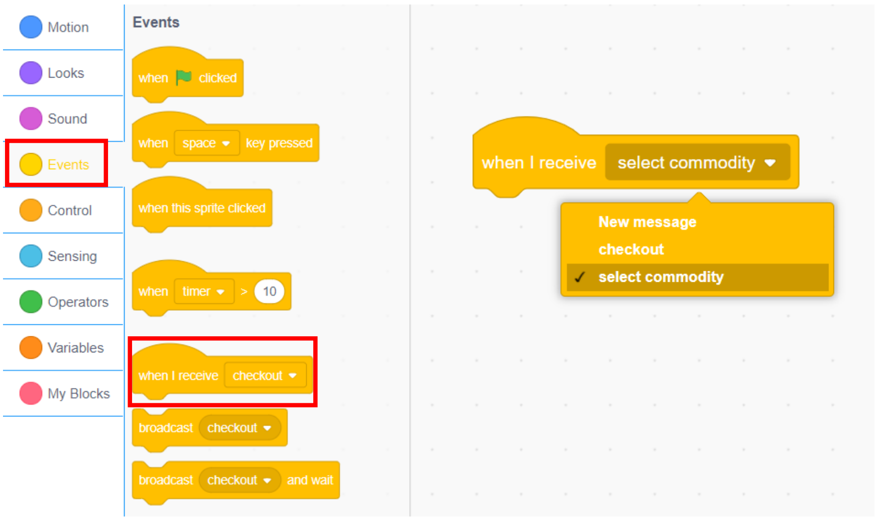
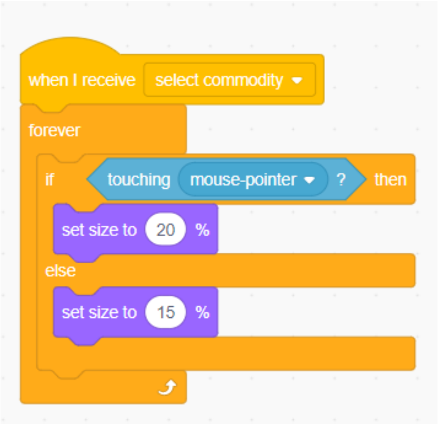


Figure 7.25 Receiving the "select commodity" message

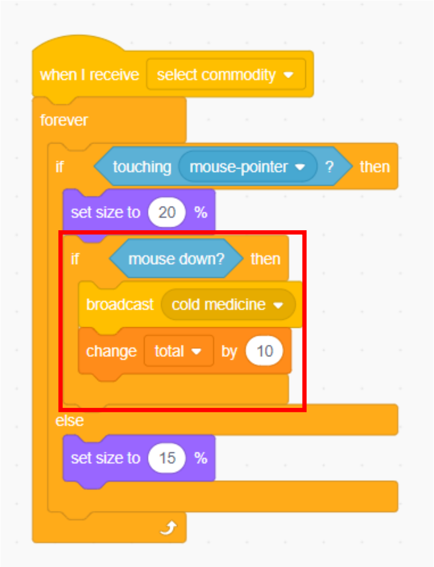
Step 8: Design the amplification function of the medicine picture. After a medicine is selected, its picture will become large if the picture touches the mouse pointer in the detection; otherwise, the picture remains the same size. Click the **Sensing** tab, and find the "touching mouse-pointer?" block. This block can judge whether the sprite touches the mouse pointer. Click the **Appearance** tab, find the "set size to..." block, and modify the parameter of this block to set the size of the sprite, as shown in Figure 7.26.



```
when I receive select commodity
forever
    if touching mouse-pointer ? then
        set size to 20 %
    else
        set size to 15 %
```

Figure 7.26 Picture amplification function

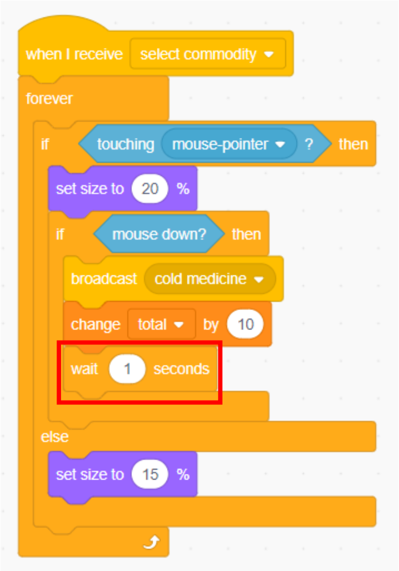
Step 9: Judge whether the mouse has clicked the "cold medicine" sprite. If yes, the system broadcasts the message of "cold medicine". In that case, the total cost increases by 10. Click the **Sensing** tab, and find the "mouse down?" block, which can monitor whether the mouse is pressed, as shown in Figure 7.27.



```
when I receive [select commodity ▾]
forever
    if <touching [mouse-pointer ▾] ?> then
        set size to (20) %
        if <mouse down?> then
            broadcast [cold medicine ▾]
            change [total ▾] by (10)
        end
    else
        set size to (15) %
    end
end
```

Figure 7.27 Monitoring whether the mouse is pressed

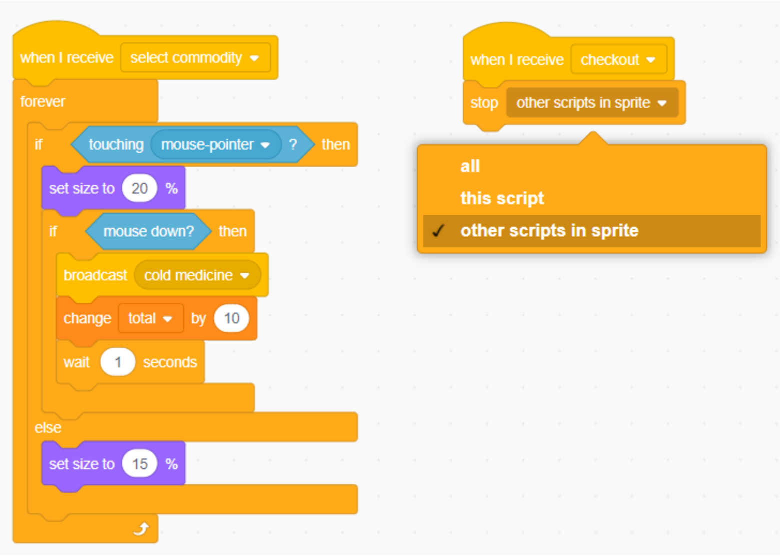
Step 10: One second after every mouse operation, detect whether the user presses the mouse down once again. This is to prevent the program from repetitively detecting the mouse operation, as shown in Figure 7.28.



```
when I receive [select commodity ▾]
forever
    if <touching [mouse-pointer ▾] ?> then
        set size to (20) %
        if <mouse down?> then
            broadcast [cold medicine ▾]
            change [total ▾] by (10)
            wait (1) seconds
        end
    else
        set size to (15) %
    end
end
```

Figure 7.28 Setting the detection of the time for pressing the mouse

Step 11: Prohibit the "cold medicine" sprite from being selected by the user when the sprite receives the "checkout" message. Drag out the "when I receive..." block, and set the message to "checkout". Then, click the **Control** tab, drag the "stop all" block to connect the "when I receive 'checkout'" block, and switch to "stop other scripts in sprite" from "stop all", as shown in Figure 7.29.



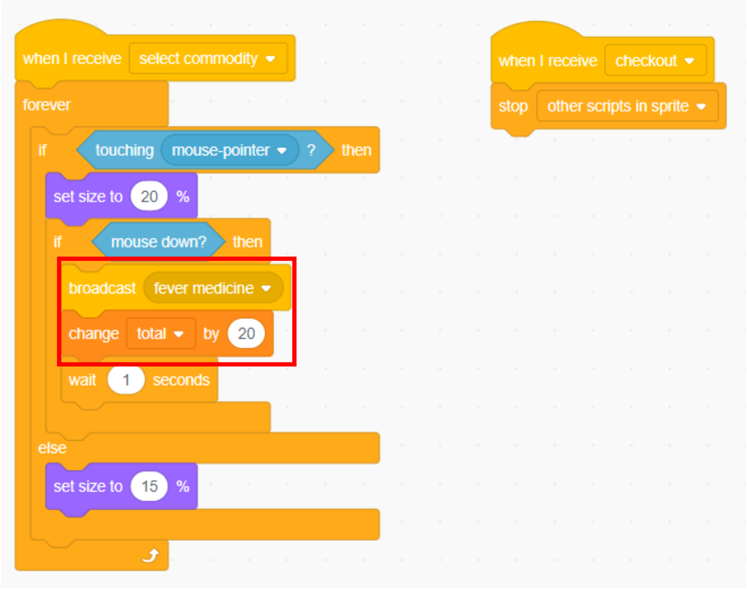
```
when I receive [select commodity ▾]
forever
    if <touching [mouse-pointer ▾] ?> then
        set size to (20) %
        if <mouse down?> then
            broadcast [cold medicine ▾]
            change [total ▾] by (10)
            wait (1) seconds
    else
        set size to (15) %
```

```
when I receive [checkout ▾]
stop [other scripts in sprite ▾]
```

Figure 7.29 Program banning the sprite from being selected

The "stop all scripts" block offers three modes: "stop all scripts", "stop this script", and "stop other scripts of this sprite". The "stop all scripts" block is to terminate all scripts of all sprites; "stop this script" is to terminate the script of this block; "stop other scripts of this sprite" is to terminate the scripts other than the script of this block.

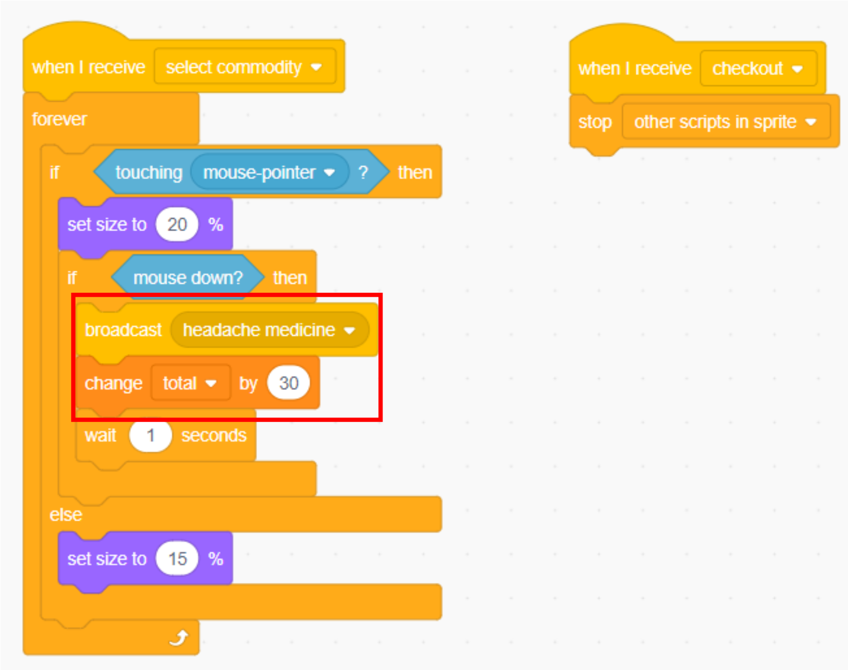
Step 12: Compile the scripts of other three medicine sprites, and set the prices of the three scripts to 20, 30 and 40 respectively, as shown in Figure 7.30.



```
when I receive [select commodity ▾]
forever
    if <touching [mouse-pointer ▾] ?> then
        set size to (20) %
        if <mouse down?> then
            broadcast [fever medicine ▾]
            change [total ▾] by (20)
            wait (1) seconds
    else
        set size to (15) %
```

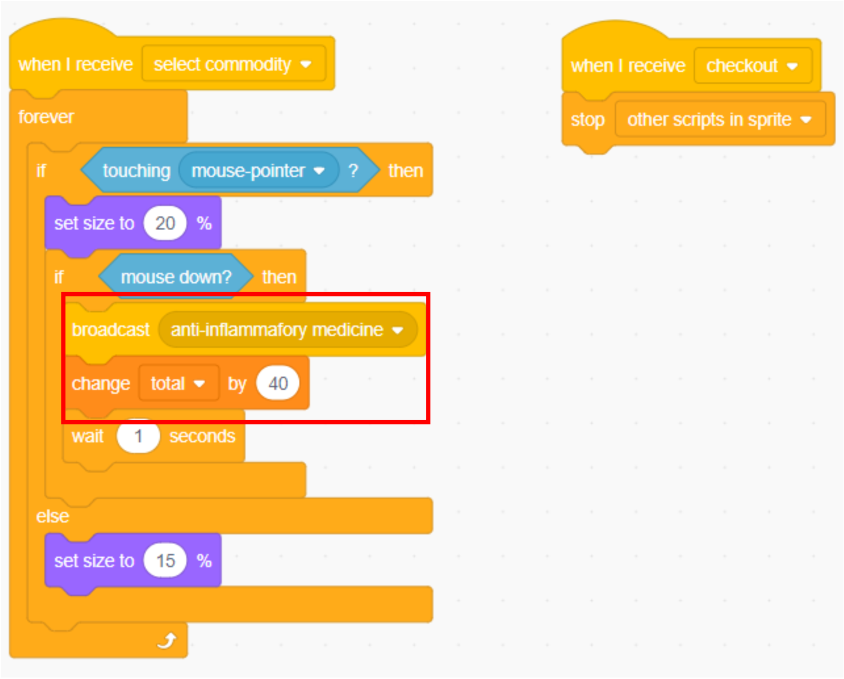
```
when I receive [checkout ▾]
stop [other scripts in sprite ▾]
```

Script of the "fever medicine" sprite



```
when I receive [select commodity ▾]
forever
    if <touching (mouse-pointer ▾) ?> then
        set size to (20) %
        if <mouse down?> then
            broadcast (headache medicine ▾)
            change [total ▾] by (30)
            wait (1) seconds
    else
        set size to (15) %

when I receive [checkout ▾]
stop [other scripts in sprite ▾]
```

Script of the "headache medicine" sprite



```
when I receive [select commodity ▾]
forever
    if <touching (mouse-pointer ▾) ?> then
        set size to (20) %
        if <mouse down?> then
            broadcast (anti-inflammafory medicine ▾)
            change [total ▾] by (40)
            wait (1) seconds
    else
        set size to (15) %

when I receive [checkout ▾]
stop [other scripts in sprite ▾]
```

Script of the "anti-inflammatory medicine" sprite

Figure 7.30 Scripts of other medicine sprites

Step 13: Save the program.

3. Conclusion

(1) The broadcast message instruction for the sprite include: "when I receive the message" instruction, ______________________________

____

(2) The mouse detection instruction is: ____________________________ .

(3) The instruction of setting the size of the sprite is: ______________ .

4. Self-Assessment

Check the completed content in the experiment task. Tick (√) the completed items, and circle (◯) the uncompleted items.

Assessment Content	Completion Status
I've learned the broadcast message instruction	
I've learned the mouse detection instruction	
I've learned the instruction of setting the size of the sprite	
I've compiled the program script of each element and sprite in the user operation interface.	

Task 3: Take Medicines and Pay

We have made the operation interface from which customers select medicines. Next, we will complete medicine pickup and payment. In this task, after a user selects a medicine, we will rely on the robot to give the medicine to him. The user will pay through face recognition technology. Then, let's complete such two processes.

1. Observation

Observe the entire medicine purchase process demonstrated by the teacher, watch and analyze the medicine pickup and payment, and record their important steps.

When the "cold medicine" Sprite is clicked, the robot should: __________

____

When the "fever medicine" Sprite is clicked, the robot should: _________

____

When the "purchase" Sprite is clicked, the robot should: _____________

____

2. Steps

Step 1: Design the experiment map, as shown in Figure 7.31.

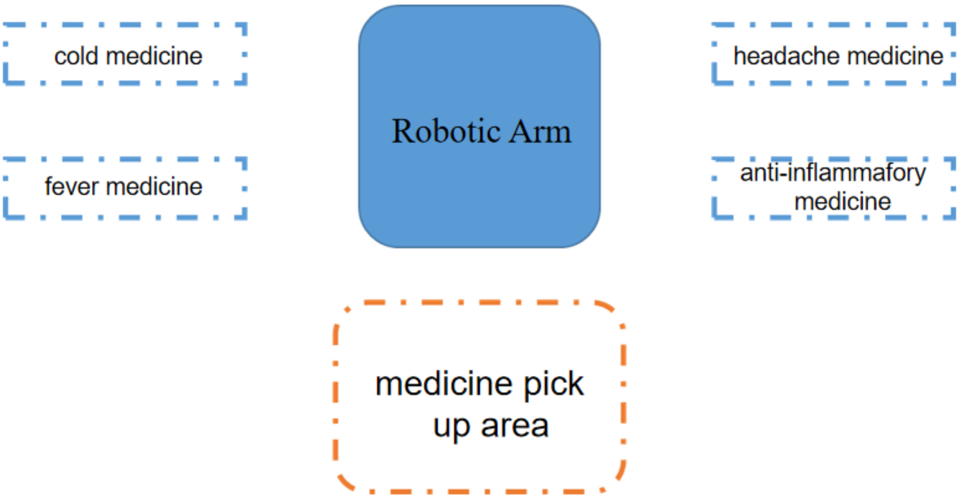


Figure 7.31 Scripts of other medicine sprites

Step 2: Compile the flow chart of medicine pickup and payment by Magician Lite robotic arm, as shown in Figure 7.32.

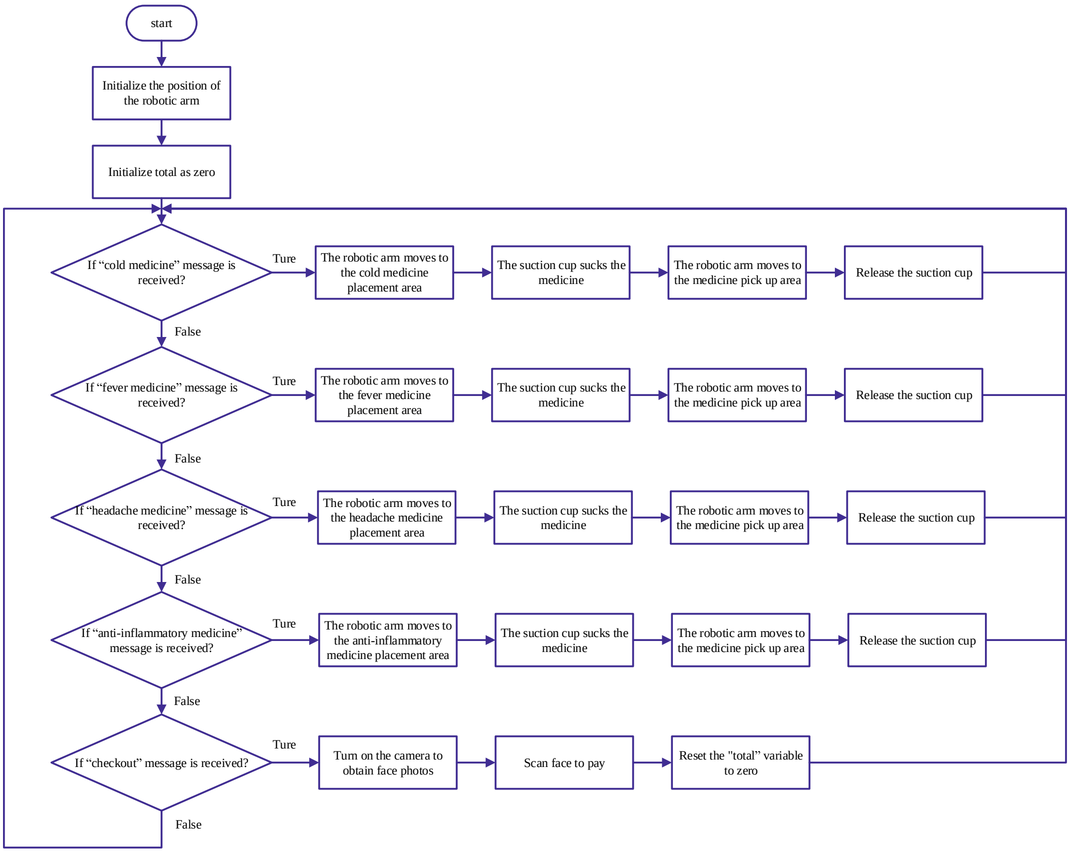
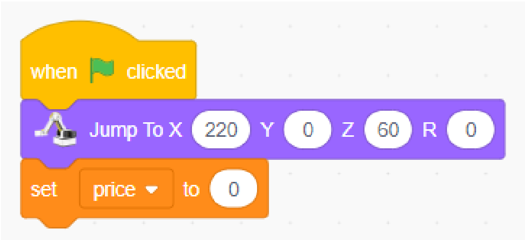

start
Initialize the position of the robotic arm
Initialize total as zero
If "cold medicine" message is received?
Ture
The robotic arm moves to the cold medicine placement area
The suction cup sucks the medicine
The robotic arm moves to the medicine pick up area
Release the suction cup
False
If "fever medicine" message is received?
Ture
The robotic arm moves to the fever medicine placement area
The suction cup sucks the medicine
The robotic arm moves to the medicine pick up area
Release the suction cup
False
If "headache medicine" message is received?
Ture
The robotic arm moves to the headache medicine placement area
The suction cup sucks the medicine
The robotic arm moves to the medicine pick up area
Release the suction cup
False
If "anti-inflammatory medicine" message is received?
Ture
The robotic arm moves to the anti-inflammatory medicine placement area
The suction cup sucks the medicine
The robotic arm moves to the medicine pick up area
Release the suction cup
False
If "checkout" message is received?
Ture
Turn on the camera to obtain face photos
Scan face to pay
Reset the "total" variable to zero
False

Figure 7.32 Flow chart of the robotic arm program

Step 3: Enable DobotScrach, open "Auto-Medicine Vending Robot.sb3", click **Device**, and add Magician Lite robotic arm.

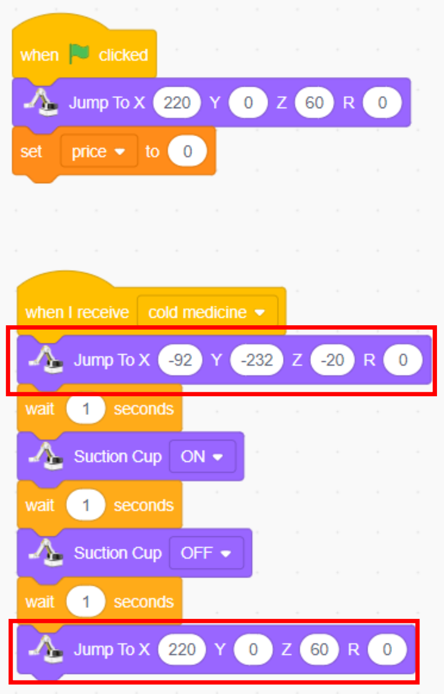
Step 4: Compile the script of Magician Lite robotic arm. Initialize the position of the robotic arm, and the variable *total* to 0, as shown in Figure 7.33.



```
when 🏁 clicked
Jump To X 220 Y 0 Z 60 R 0
set price ▾ to 0
```

Figure 7.33 Initializing the script

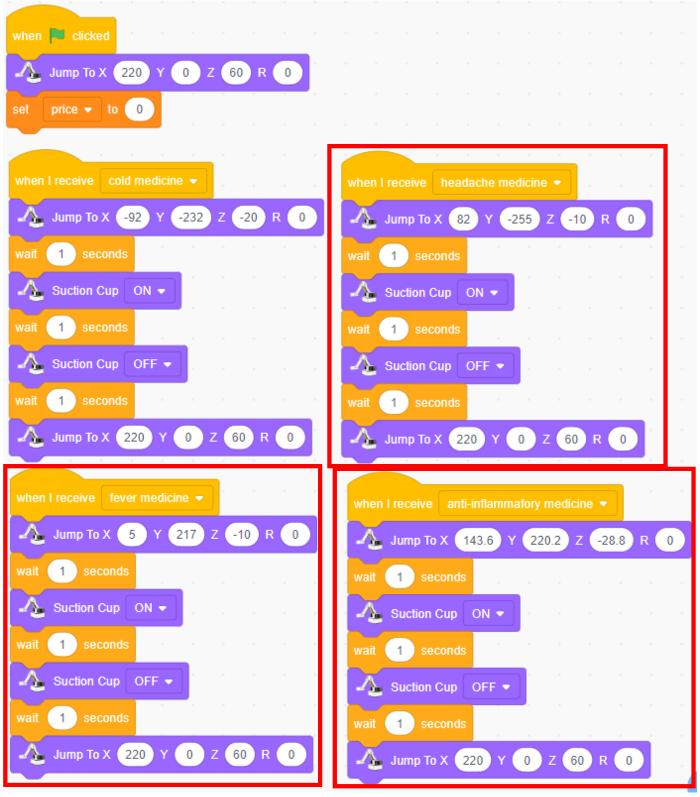
Step 5: Judge whether the robotic arm has received the "cold medicine" message; if yes, control the robotic arm to grab a cold medicine, as shown in Figure 7.34.



```
when [flag] clicked
Jump To X (220) Y (0) Z (60) R (0)
set [price ▾] to (0)

when I receive [cold medicine ▾]
Jump To X (-92) Y (-232) Z (-20) R (0)
wait (1) seconds
Suction Cup [ON ▾]
wait (1) seconds
Suction Cup [OFF ▾]
wait (1) seconds
Jump To X (220) Y (0) Z (60) R (0)
```

Figure 7.34 Script of cold medicine pickup

Step 6: Judge whether the robotic arm has received the message of other three medicines; if yes, the robotic arm grabs the proper cold medicine to the medicine pickup area, as shown in Figure 7.35.



```
when [flag] clicked
Jump To X (220) Y (0) Z (60) R (0)
set [price ▾] to (0)

when I receive [cold medicine ▾]
Jump To X (-92) Y (-232) Z (-20) R (0)
wait (1) seconds
Suction Cup [ON ▾]
wait (1) seconds
Suction Cup [OFF ▾]
wait (1) seconds
Jump To X (220) Y (0) Z (60) R (0)

when I receive [fever medicine ▾]
Jump To X (5) Y (217) Z (-10) R (0)
wait (1) seconds
Suction Cup [ON ▾]
wait (1) seconds
Suction Cup [OFF ▾]
wait (1) seconds
Jump To X (220) Y (0) Z (60) R (0)

when I receive [headache medicine ▾]
Jump To X (82) Y (-255) Z (-10) R (0)
wait (1) seconds
Suction Cup [ON ▾]
wait (1) seconds
Suction Cup [OFF ▾]
wait (1) seconds
Jump To X (220) Y (0) Z (60) R (0)

when I receive [anti-inflammafory medicine ▾]
Jump To X (143.6) Y (220.2) Z (-28.8) R (0)
wait (1) seconds
Suction Cup [ON ▾]
wait (1) seconds
Suction Cup [OFF ▾]
wait (1) seconds
Jump To X (220) Y (0) Z (60) R (0)
```

Figure 7.35 Script of pickup of other medicines

Step 7: Enable the face recognition payment function upon receipt of the "checkout" message by the robotic arm. Edit the face data, and enter the face data of customers. Add the AI module, click the **AI** tab, select the "New face data", enter face data and compile the face label, as shown in Figure 7.36.

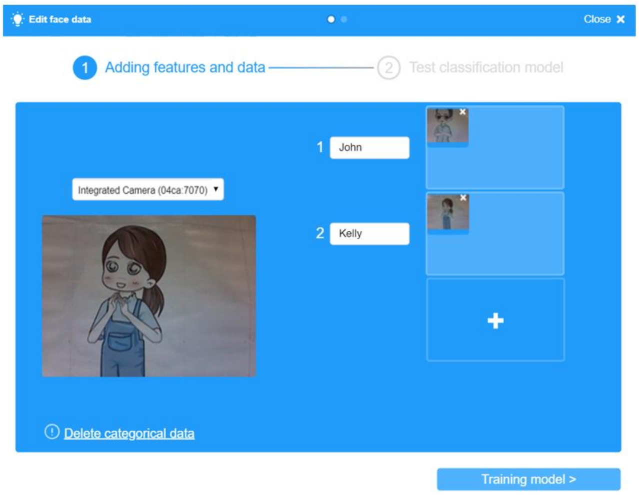
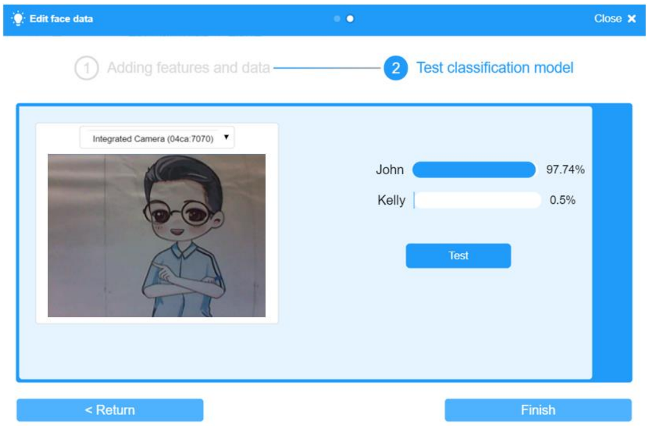

Edit face data
Close ✕
1 Adding features and data
2 Test classification model
Integrated Camera (04ca:7070) ▾
1 John
2 Kelly
+
Delete categorical data
Training model >

Figure 7.36 Entering the face data

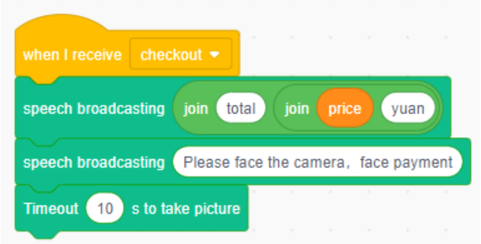
Step 8: Train the face recognition model, as shown in Figure 7.37.


Edit face data
Close ✕
1 Adding features and data
2 Test classification model
Integrated Camera (04ca:7070) ▾
Test
< Return
Finish

John	97.74%
Kelly	0.5%

Figure 7.37 Training the face recognition model

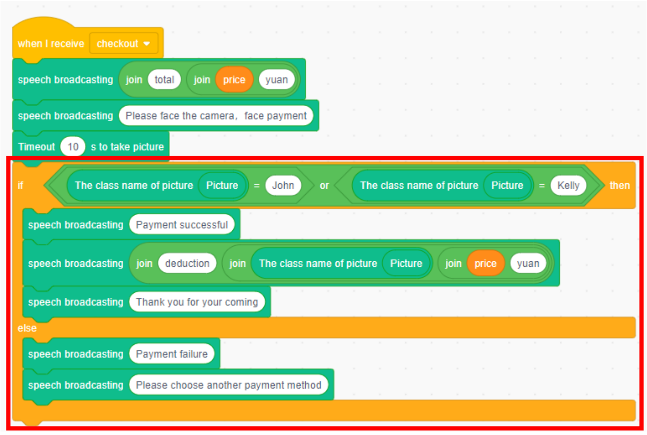
Step 9: Compile the face payment script. Receive the speech message of the total commodity cost, and then get the face picture of the customer, as shown in Figure 7.38.



```
when I receive checkout ▾
speech broadcasting join total join price yuan
speech broadcasting Please face the camera, face payment
Timeout 10 s to take picture
```

Figure 7.38 Getting the face picture

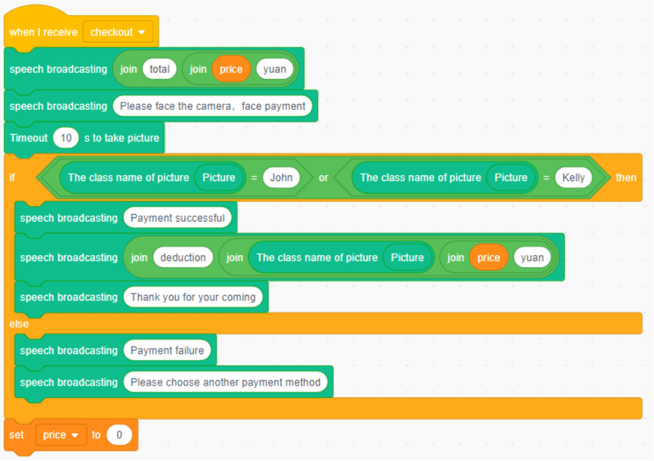
Step 10: Judge whether the face has been input into the data set of the face payment system. If yes, payment succeeds; otherwise, a message "payment failure" appears. In that case, use other payment methods, as shown in Figure 7.39.



```
when I receive checkout
speech broadcasting join total join price yuan
speech broadcasting Please face the camera，face payment
Timeout 10 s to take picture
if The class name of picture Picture = John or The class name of picture Picture = Kelly then
    speech broadcasting Payment successful
    speech broadcasting join deduction join The class name of picture Picture join price yuan
    speech broadcasting Thank you for your coming
else
    speech broadcasting Payment failure
    speech broadcasting Please choose another payment method
```

Figure 7.39 Face payment script

Step 11: Clear the variable *total* upon payment, as shown in Figure 7.40.



```
when I receive checkout
speech broadcasting join total join price yuan
speech broadcasting Please face the camera，face payment
Timeout 10 s to take picture
if The class name of picture Picture = John or The class name of picture Picture = Kelly then
    speech broadcasting Payment successful
    speech broadcasting join deduction join The class name of picture Picture join price yuan
    speech broadcasting Thank you for your coming
else
    speech broadcasting Payment failure
    speech broadcasting Please choose another payment method
set price to 0
```

Figure 7.40 Clearing the variable *total*

3. Conclusion

(1) The instructions about receiving and sending messages by the robotic arm: "when I receive the message" instruction, _________

(2) The steps to edit face data: collect face data, _______________ , test the face model.

(3) The face recognition instruction: __________________________ .

4. Self-Assessment

Check the completed content in the experiment task. Tick (√) the completed items, and circle (◯) the uncompleted items.

Assessment Content	Completion Status
I’ve programmed the medicine pickup of the robot	
I’ve programmed the face recognition payment	

Experiment 8

Smart Checkout

Team Name: Team member: Date:

Overview

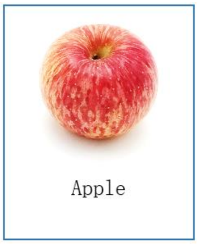
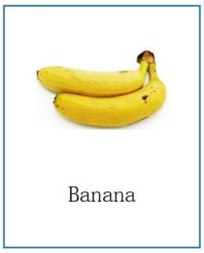
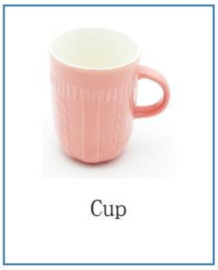
Before leaving smart mall, we must pay for commodities purchased. Then, how can we do this faster and more conveniently? In this experiment, we quickly pay by scanning commodities, using the system to automatically calculate prices and finally deducting fees through face recognition technology. Next, we will complete automatic pricing and automatic payment.

Objective

- Understand the speech recognition technology and skillfully use the speech recognition instruction by entering and recognizing the speech and performing other speech operations.
- Use the character recognition instruction skillfully by recognizing the characters on the picture and learning the OCR technology.
- Understand how to apply the face recognition instruction by collecting and training face data.
- Use the variable skillfully and grasp its basic logic operations by calculating commodity prices and total purchase prices.
- Understand how to program automatic payment by using the branch structure comprehensively and the loop structure to automatically calculate prices.

Equipment

Equipment Picture	Name	Quantity

	Dobot Magician Lite Robotic arm	1
	Camera	1
	Power adapter	1
	USB cable	1
	Apple card	A certain number
	Banana card	A certain number
	Cup card	A certain number

Requirements

- ➢ Take care when using electricity.

- Before the experiment, check whether the experimental equipment is complete and intact. If there is any omission or damage, please report to the teacher.
- Any specific operations in the experiment shall be performed according to the experiment manual. If you have any questions, please promptly ask the teacher.
- During the experiment, the joints will start to work as the robotic arms are powered on. In that case, do not move the joints of the robotic arms hard if you do not press the unlock key.
- Arrange all devices after the experiment. You shall not leave the lab before check by the group leader.

Task 1: Design Automatic Pricing

After we purchase something in shops or supermarkets, the system will automatically calculate the total price of our purchased commodities as the QR codes on the commodity package are scanned. Then, can we allow the robotic arm to automatically calculate the price after scanning commodities?

5. Observation

The main steps to analyze automatic pricing:

(1) Enter the speech before arousing the robotic arm.

(2) Through speech recognition, judge whether the speech recognition result includes:________________________________ .

(3) When finding "I will scan commodities" through speech recognition, the robotic arm starts to ________________________ .

(4) Detect whether the speech is entered. And recognize the entered speech content.

(5) When finding through speech recognition, the robotic arm starts to broadcast the total price of this purchase; when not recognizing relevant information, the robotic arm continues to scan other commodities.

6. Steps

(1) Prepare Hardware

Step 1: Prepare the experiment equipment, power on the robotic arm, and connect it to the computer.

Step 2: Set the equipment to the corresponding position, as shown in Figure 8.1. After powering on the robotic arm, adjust the position and height of the end of the robotic arm to help scan commodities.

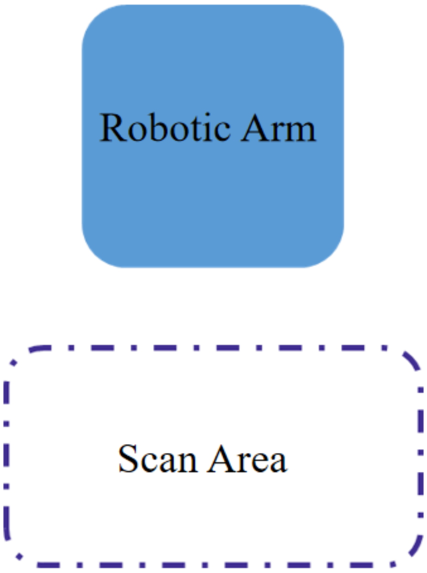


Figure 8.1 Placement position

(2) Design Program

Step 1: Analyze the automatic pricing process, and read the flow chart, as shown in Figure 8.2.

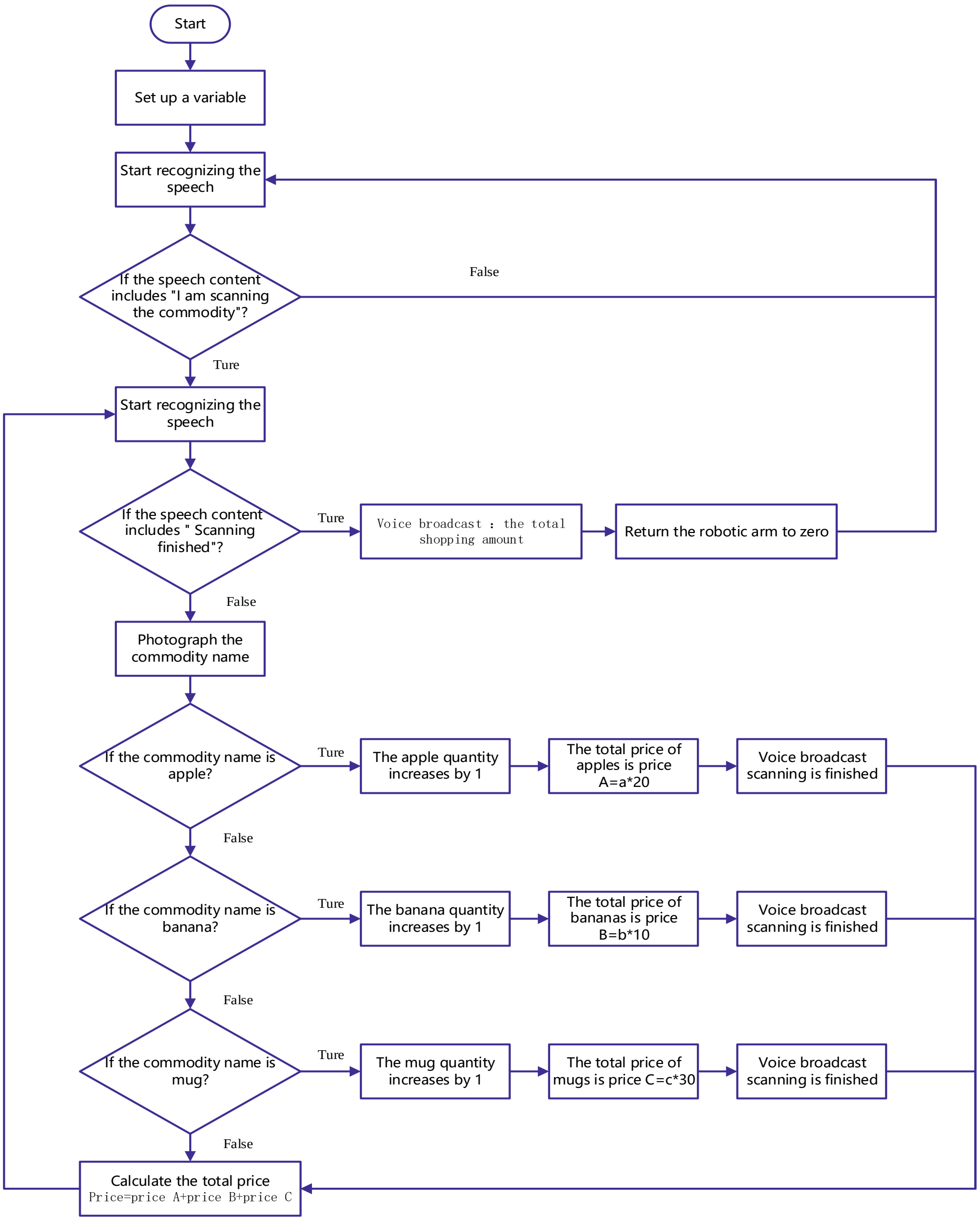

Start
Set up a variable
Start recognizing the speech
If the speech content includes "I am scanning the commodity"?
False
Ture
Start recognizing the speech
If the speech content includes " Scanning finished"?
Ture
Voice broadcast : the total shopping amount
Return the robotic arm to zero
False
Photograph the commodity name
If the commodity name is apple?
Ture
The apple quantity increases by 1
The total price of apples is price A=a*20
Voice broadcast scanning is finished
False
If the commodity name is banana?
Ture
The banana quantity increases by 1
The total price of bananas is price B=b*10
Voice broadcast scanning is finished
False
If the commodity name is mug?
Ture
The mug quantity increases by 1
The total price of mugs is price C=c*30
Voice broadcast scanning is finished
False
Calculate the total price Price=price A+price B+price C

Figure 8.2 Realization flow chart

Step 2: Enable DobotScratch software, select and connect the device Magician Lite; and add the AI Extension module.

Step 3: Click the **Events** tab, and drag the "when the green flag is clicked" block to the code area.

Step 4: Create a variable. Create the "scan payment" variable to record the speech recognition result; create another variable to record the quantity and total price of each kind of commodity, and the total price of all commodities.

☛ In Scratch, create a variable to meet the operation needs of automatic pricing.

(1) To create a variable, refer to the mode as shown in Figure 8.3.

(2) If you do not set the initial value of each created variable, the initial value is, by default, *0*.

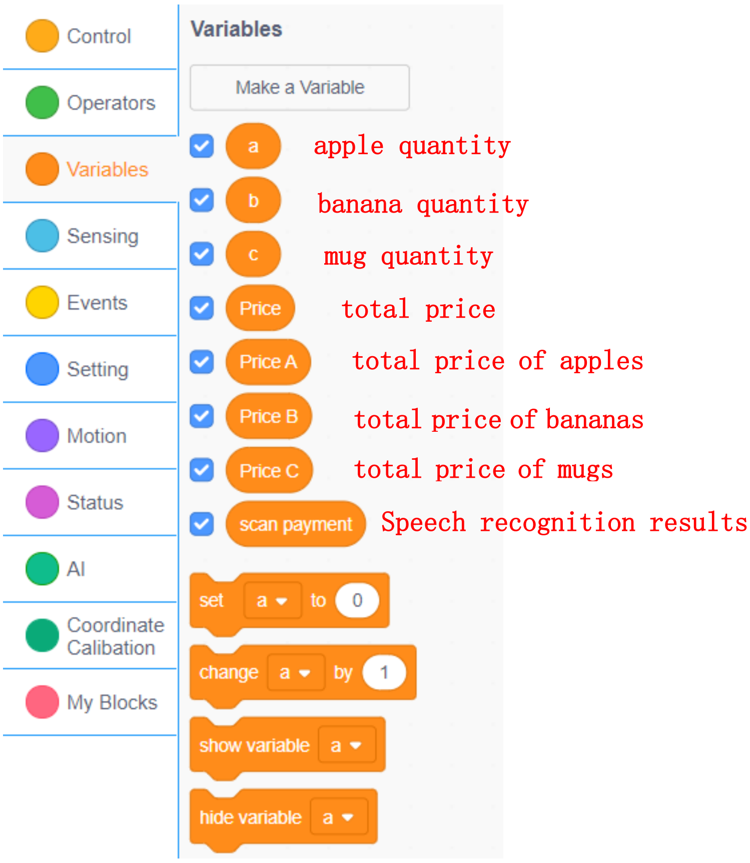


Figure 8.3 Creating the variable

Step 5: Click the **AI** tab, drag the "start English voice recognition, continued for 1 second" block to connect the preceding block, and modify the duration to 5 seconds, as shown in Figure 8.4.

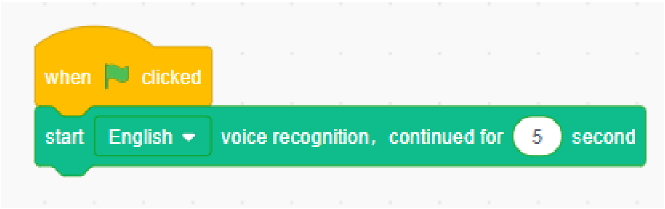
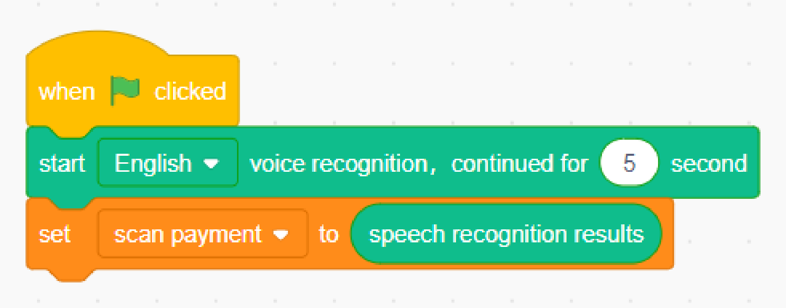


Figure 8.4 Starting the speech recognition

Step 6: Click the **Variables** tab, and drag the "set the scan payment to (0)" block to connect the preceding block; click the **AI** tab, drag the "speech recognition results" block, and set the "scan payment" value as the speech recognition result, as shown in Figure 8.5.



```
when 🏁 clicked
start English ▾ voice recognition, continued for 5 second
set scan payment ▾ to speech recognition results
```

Figure 8.5 Setting the "scan payment" value as the speech recognition results

Step 7: Judge whether the speech recognition result includes "I will scan commodities".

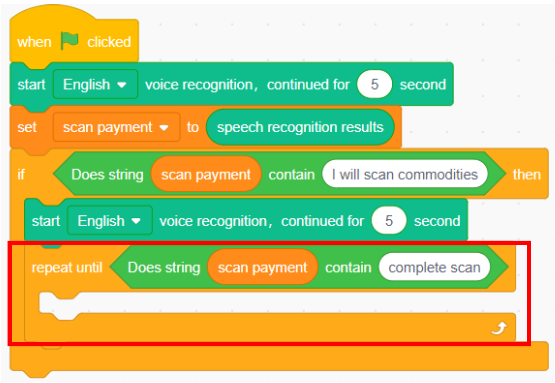
☛ In Scratch, write the judgment program, and detect whether relevant contents are recognized.

Step 8: Restart the speech recognition.

☛ In Scratch, drag the "open speech recognition" block, and restart the speech recognition.

Step 9: If the speech recognition result includes "complete scan", the robotic arm broadcasts the speech message "The total price of this purchase is…"; if not, the robotic arm repeats the scan.

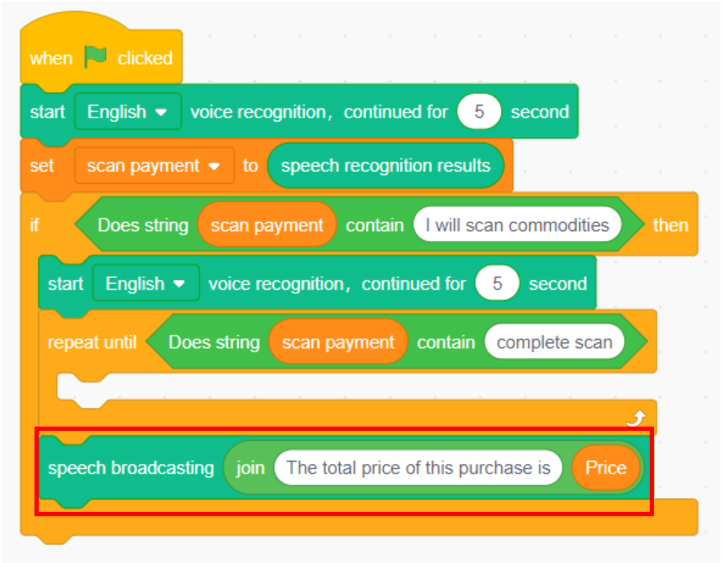
Click the **Control** tab, and drag the "repeat until…" block to the position under the second speech recognition block; click the **Operators** tab, drag the "does string () contain ()" block to the position of the hexagonal decision box; click the **Variables** tab, drag the variable "scan checkout" block to the left ellipse position of the "does string () contain ()", and fill in "complete scan" on the right side, as shown in Figure 8.6.



```
when 🏴 clicked
start English ▾ voice recognition, continued for 5 second
set scan payment ▾ to speech recognition results
if Does string scan payment contain I will scan commodities then
    start English ▾ voice recognition, continued for 5 second
    repeat until Does string scan payment contain complete scan
        
```

Figure 8.6 Judging whether the speech recognition result includes "complete scan"

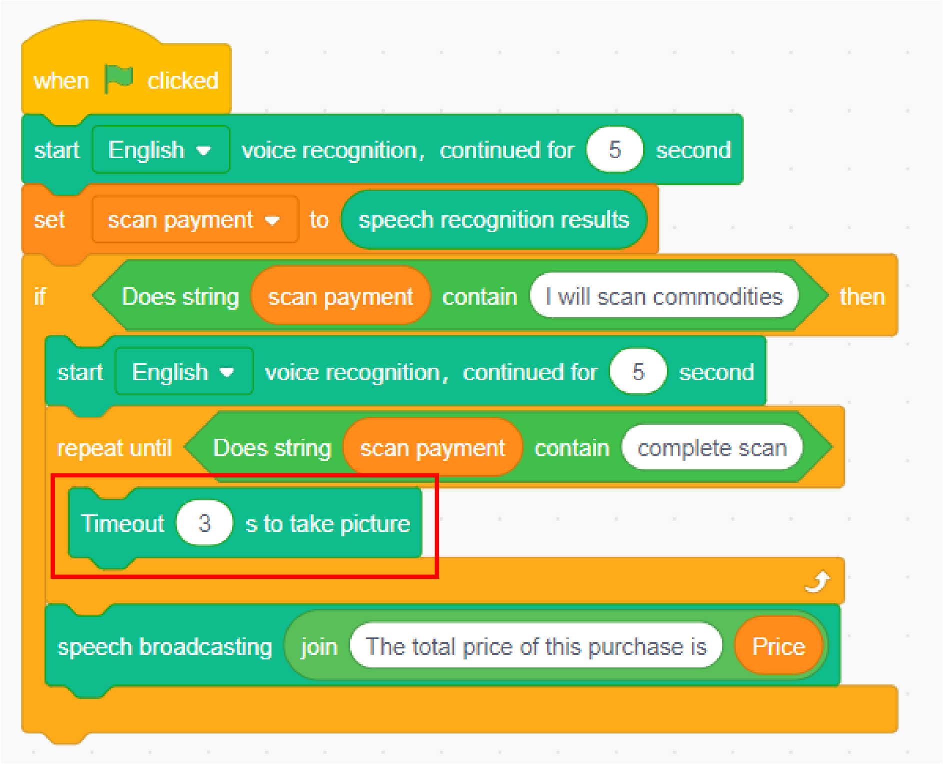
Step 10: If the speech recognition result includes "complete scan", the robotic arm broadcasts the speech message "The total price of this purchase is…". Click the **AI** tab, and drag the "speech broadcasting hello" block to connect the position under the "repeat until…" block; click the **Operators** tab, and drag the "join () and ()" block to the position of the speech broadcast content; fill in the "the total price of this purchase is…" on the left side of the block, connect the variable "Price" block on the right, and put the variable "Price" block in the **Variables** tab, as shown in Figure 8.7.



```
when 🏴 clicked
start English ▾ voice recognition, continued for 5 second
set scan payment ▾ to speech recognition results
if Does string scan payment contain I will scan commodities then
    start English ▾ voice recognition, continued for 5 second
    repeat until Does string scan payment contain complete scan
        
    speech broadcasting join The total price of this purchase is Price
```

Figure 8.7 Setting the speech broadcasting

Step 11: When the speech recognition result excludes "complete scan", the robotic arm starts to repetitively scan the commodity, but first automatically photograph the commodity. Click the **AI** tab, and drag the "Timeout 3s to take picture" block to the position of the "repeat until…", as shown in Figure 8.8.

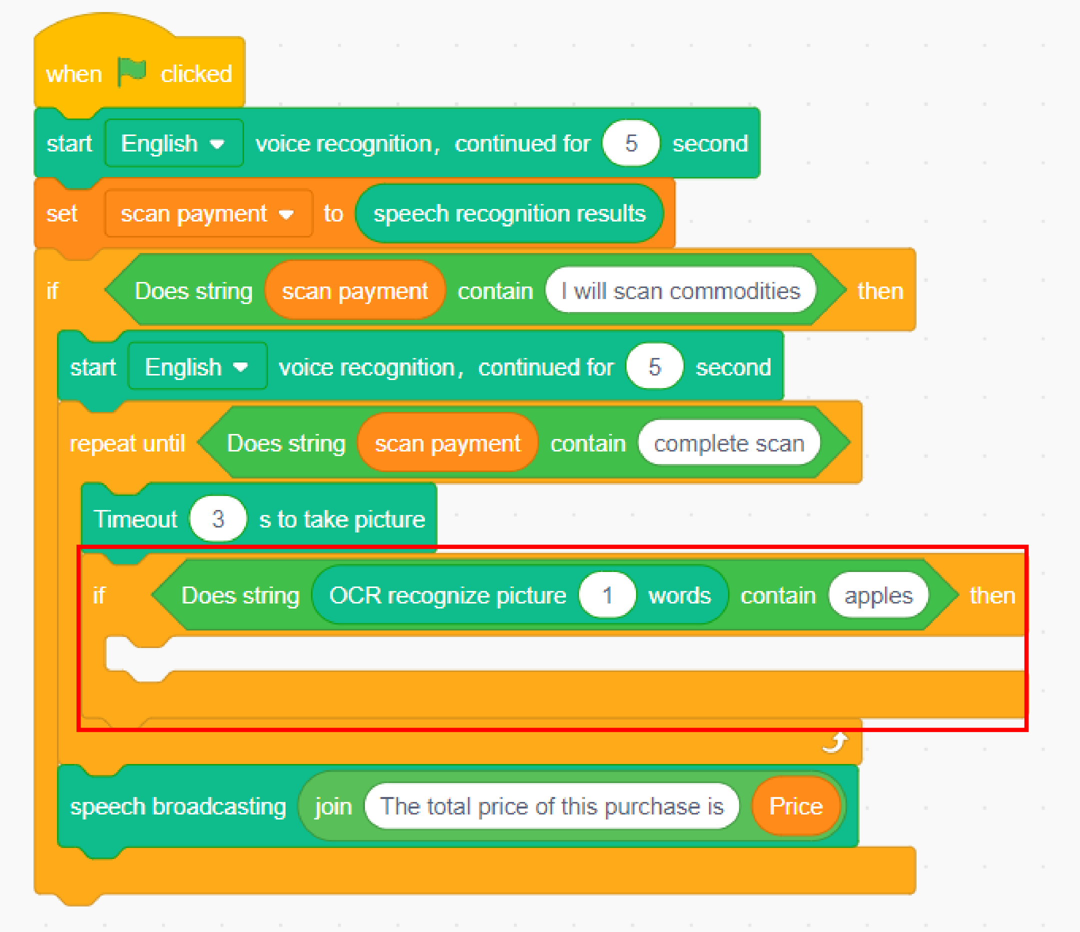


```
when 🏴 clicked
start English ▾ voice recognition, continued for (5) second
set scan payment ▾ to (speech recognition results)
if <Does string (scan payment) contain (I will scan commodities)> then
    start English ▾ voice recognition, continued for (5) second
    repeat until <Does string (scan payment) contain (complete scan)>
        Timeout (3) s to take picture
    ↺
    speech broadcasting (join (The total price of this purchase is) (Price))
```

Figure 8.8 Automatic photography block

Step 12: Judge whether the commodity is apple. If yes, execute the apple-related operations; otherwise judge whether the commodity is other products.

Click the **Control** tab, drag the "if…then…" block to connect the position under the "Timeout 3s to take picture" block; click the "operators" tab, and drag the "Does string () contain ()" block to the hexagonal decision box of the "if…then…" block; click the **AI** tab, drag the " OCR recognize picture(1) words" block to the left ellipse position of the "Does string () contain ()", and fill in the keyword "Apple" on the right side, as shown in Figure 8.9.

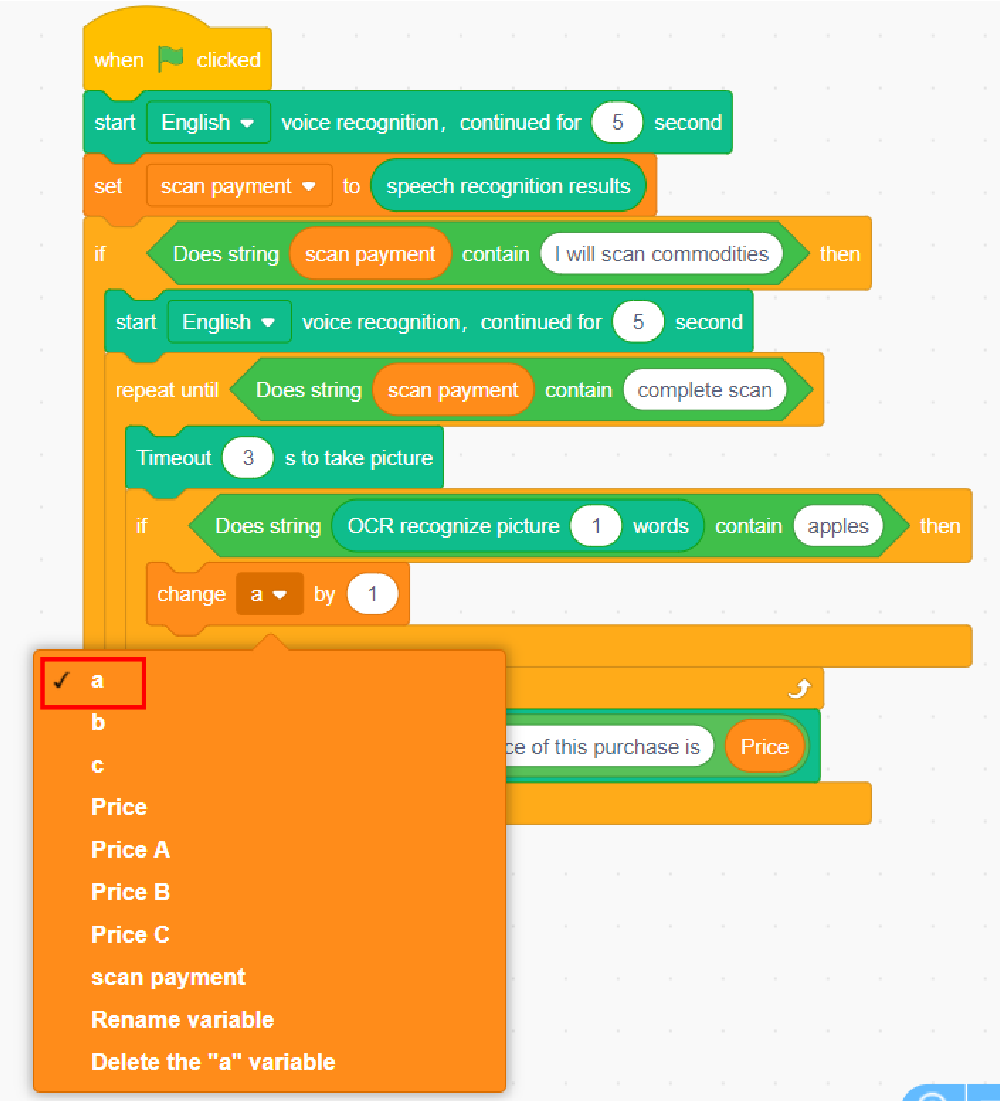


```
when ⚑ clicked
start English ▾ voice recognition, continued for 5 second
set scan payment ▾ to speech recognition results
if Does string scan payment contain I will scan commodities then
    start English ▾ voice recognition, continued for 5 second
    repeat until Does string scan payment contain complete scan
        Timeout 3 s to take picture
        if Does string OCR recognize picture 1 words contain apples then

    speech broadcasting join The total price of this purchase is Price
```

Figure 8.9 Judging the commodity as apple

Step 13: For the variable *a* that records the quantity of apples, increase it by 1 when the commodity is apple. The total price of apples is calculated as Price A=a*5 (5 indicates the unit price of apples, that is, 5 yuan/piece).

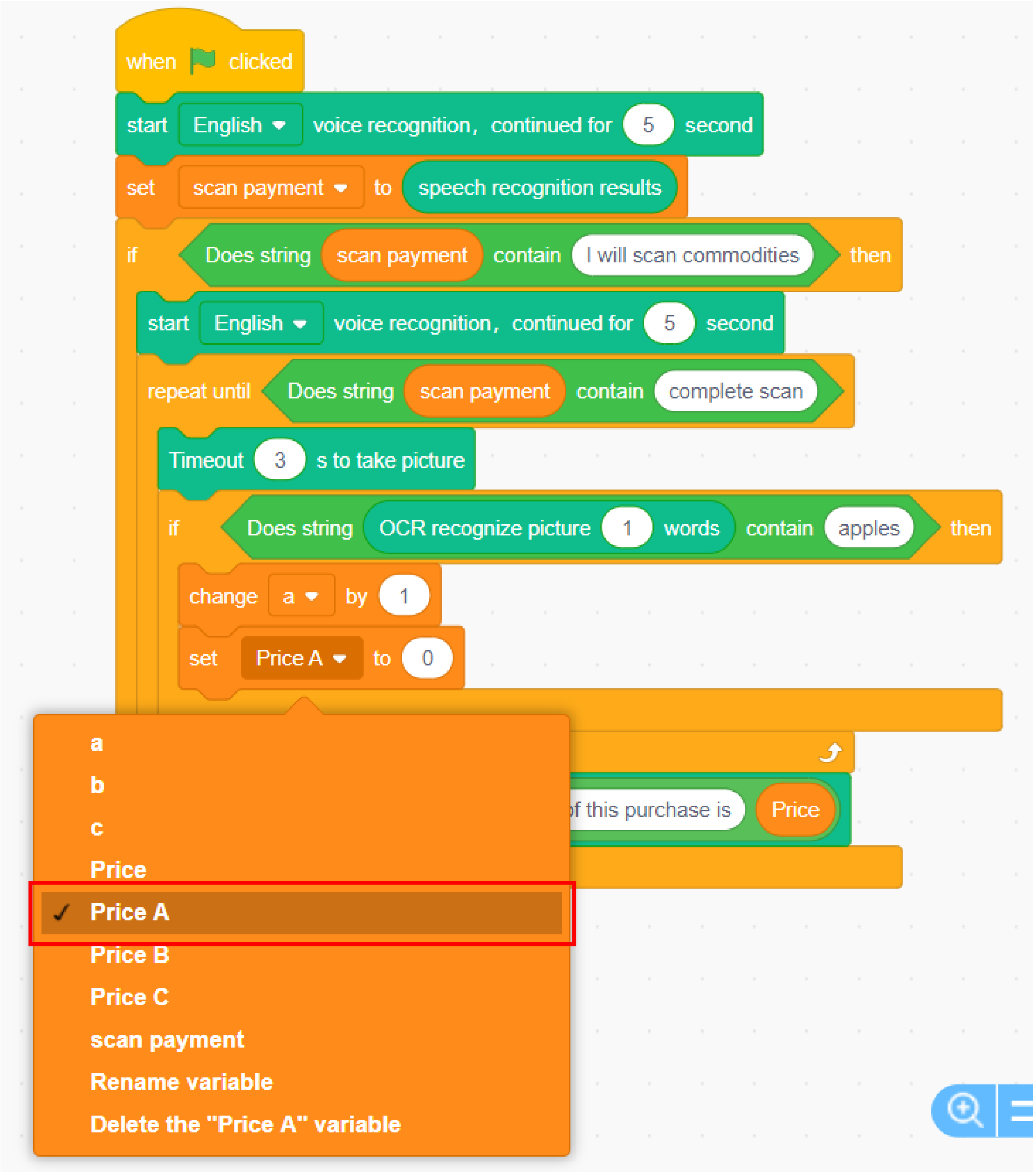
1) Click the **Variables** tab, and drag the "increase the scan payment by 1" block to the position of the "if…then…" execution block; click the inverted triangular sign of the "increase the scan payment by 1", and select the variable *a* from the drop-down list, as shown in Figure 8.10.



```
when 🏁 clicked
start English ▾ voice recognition, continued for 5 second
set scan payment ▾ to speech recognition results
if Does string scan payment contain I will scan commodities then
  start English ▾ voice recognition, continued for 5 second
  repeat until Does string scan payment contain complete scan
    Timeout 3 s to take picture
    if Does string OCR recognize picture 1 words contain apples then
      change a ▾ by 1
ce of this purchase is Price
```

Figure 8.10 Increasing the variable *a* by 1

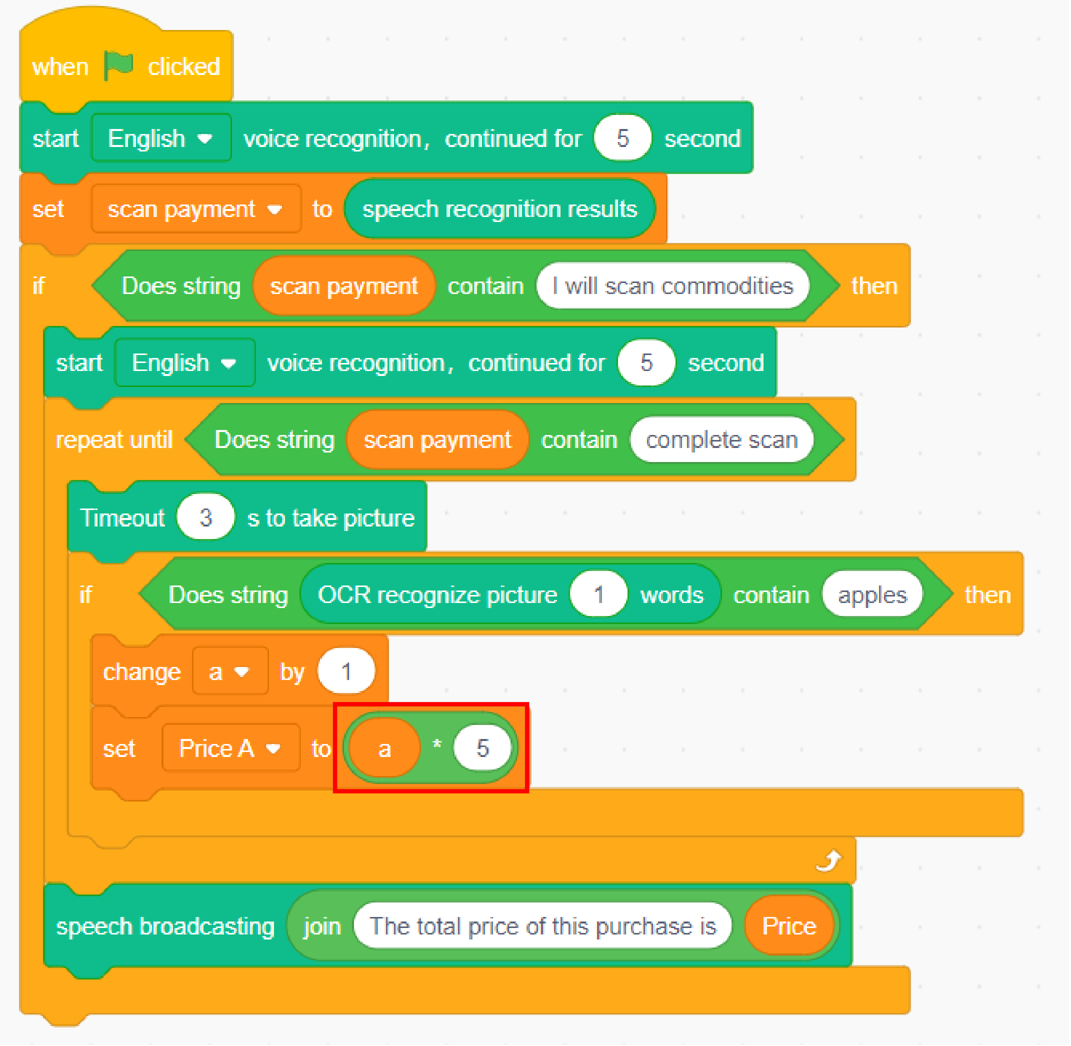
2) Click the **Variables** tab, and drag the "set the scan payment to 0" block to connect the position of the "increase *a* by 1" block; click the inverted triangular sign of the "set the scan payment to 0", and select the variable *Price A* from the drop-down list, as shown in Figure 8.11.



```
when ⚑ clicked
start English voice recognition, continued for 5 second
set scan payment to speech recognition results
if Does string scan payment contain I will scan commodities then
    start English voice recognition, continued for 5 second
    repeat until Does string scan payment contain complete scan
        Timeout 3 s to take picture
        if Does string OCR recognize picture 1 words contain apples then
            change a by 1
            set Price A to 0
```

Figure 8.11 Variable block

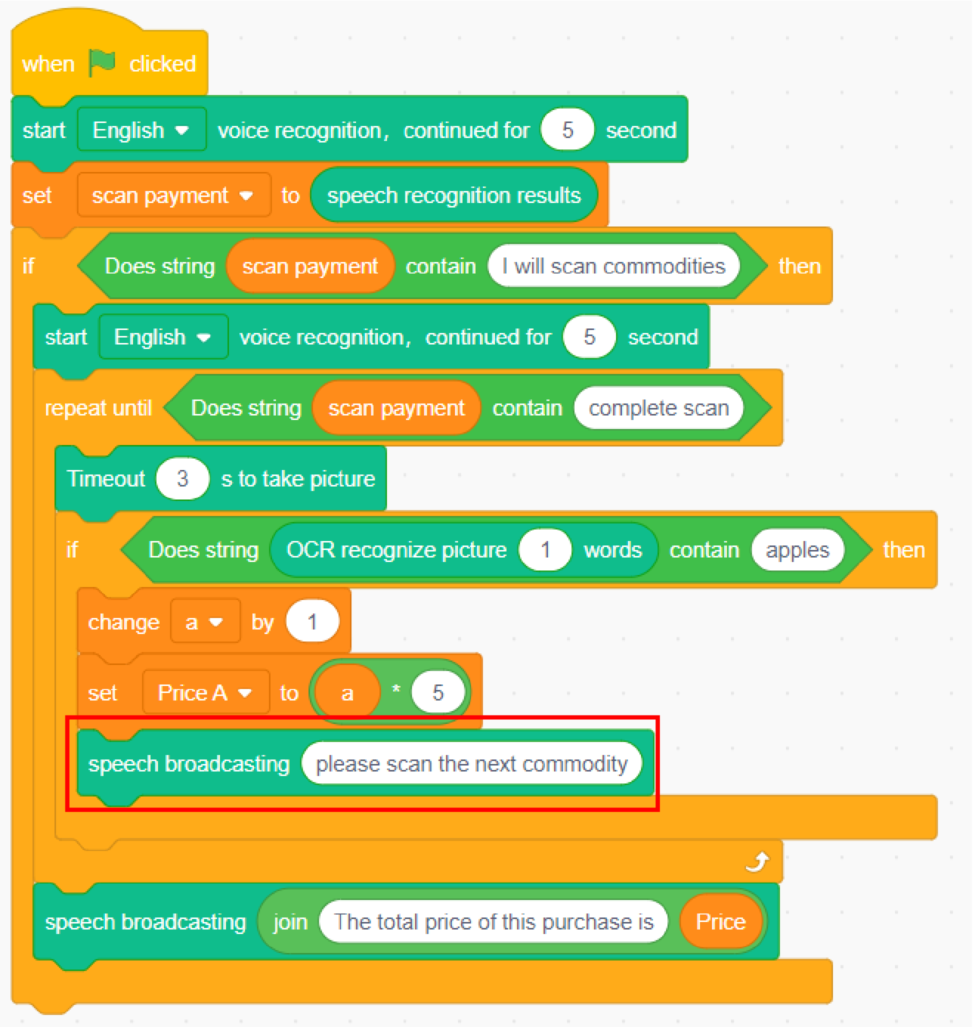
3) Click the **Operators** tab, and drag the "multiple sign" block to the position of the right value 0 of "set Price A to 0"; click the **Variables** tab, drag the variable "a" block to the left ellipse position of the "multiple sign" block, and fill in the value 5 on the right side, as shown in Figure 8.12.



```
when ⚑ clicked
start English ▾ voice recognition，continued for 5 second
set scan payment ▾ to speech recognition results
if Does string scan payment contain I will scan commodities then
    start English ▾ voice recognition，continued for 5 second
    repeat until Does string scan payment contain complete scan
        Timeout 3 s to take picture
        if Does string OCR recognize picture 1 words contain apples then
            change a ▾ by 1
            set Price A ▾ to a * 5
    speech broadcasting join The total price of this purchase is Price
```

Figure 8.12 Setting the variable *Price A*

Step 14: Broadcast the speech message "please scan the next commodity", and Figure 8.13 shows the programming mode.



```
when ⚑ clicked
start English ▾ voice recognition, continued for (5) second
set scan payment ▾ to speech recognition results
if Does string scan payment contain I will scan commodities then
    start English ▾ voice recognition, continued for (5) second
    repeat until Does string scan payment contain complete scan
        Timeout (3) s to take picture
        if Does string OCR recognize picture (1) words contain apples then
            change a ▾ by (1)
            set Price A ▾ to (a * 5)
            speech broadcasting please scan the next commodity
        
    ↺
    speech broadcasting join The total price of this purchase is Price
```

Figure 8.13 Broadcasting the message "please scan the next commodity"

Step 15: Judge whether the commodity is banana. If yes, execute the banana-related operations; otherwise, judge whether the commodity is other products.

Review: If the commodity is judged as apple, provide the programming steps.

In Scratch, the programming realizes: When the commodity is banana, the variable *b* that records the quantity of bananas increases by 1, and the total price of bananas is Price A=b*2 (2 indicates the unit price of bananas, that is, 2 yuan/piece).

Step 16: Judge whether the commodity is mug. If yes, execute the mug-related operations; otherwise, judge whether the commodity is other products.

Review: If the commodity name is judged as apple or banana, provide the respective programming steps.

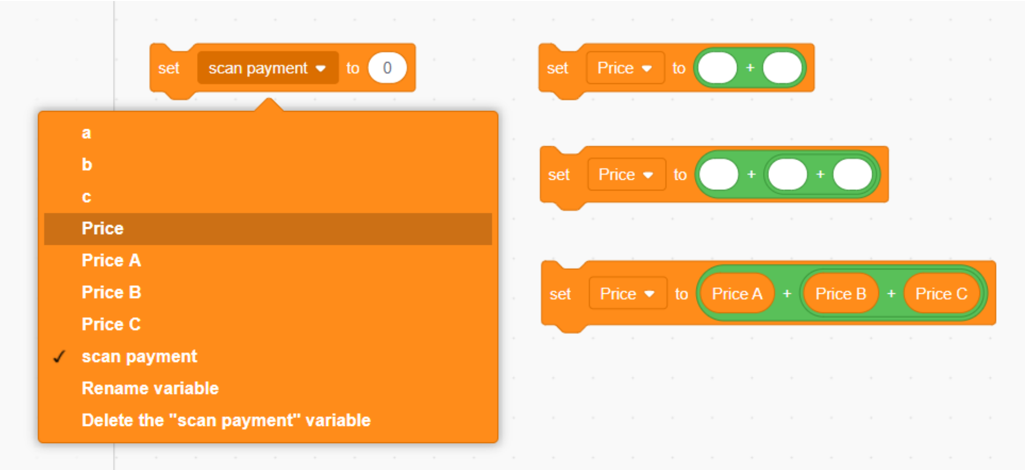
In Scratch, the programming realizes: When the commodity name is cup, the variable *c* that records the quantity of bananas increases by 1, and the total price of cups is Price C=c*30 (30 indicates the unit price of cups, that is, 30 yuan/piece).

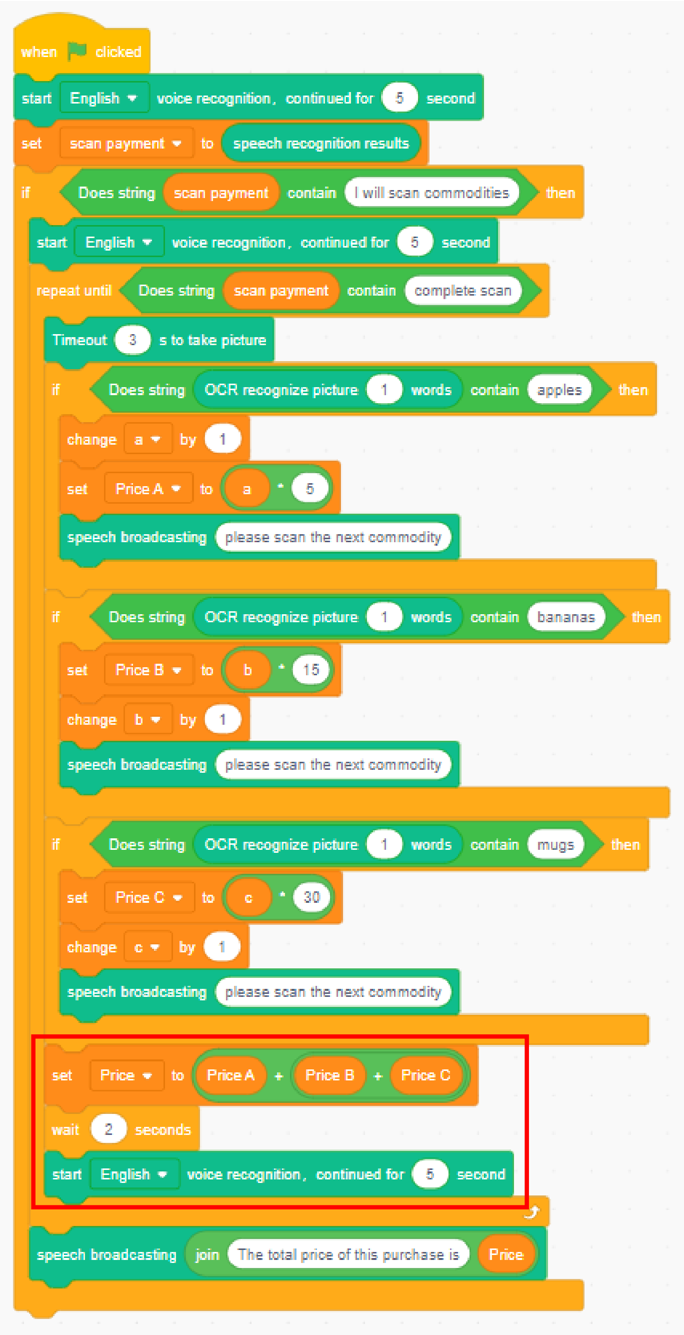
Step 17: After scanning the commodity and calculating its quantity and price, calculate its total price in real time as follows: Price = Price A+ Price B+ Price C.

1) Click the **Variables** tab, drag the "set the scan payment to 0" block to the code area, click the inverted triangular sign of the "set the scan payment to 0" block, and select the variable *Price* from the drop-down list.

2) Click the **Operators** tab, and drag the "plus sign" block to the right ellipse position of "set Price to 0"; drag the "plus sign" block again to the right ellipse position of the original "plus sign" block.

3) Click the **Variables** tab, and drag the variables *Price A*, *Price B* and *Price C* to the ellipse positions corresponding to the "plus sign" block; 2 seconds later, restart the speech recognition, as shown in Figure 8.14.

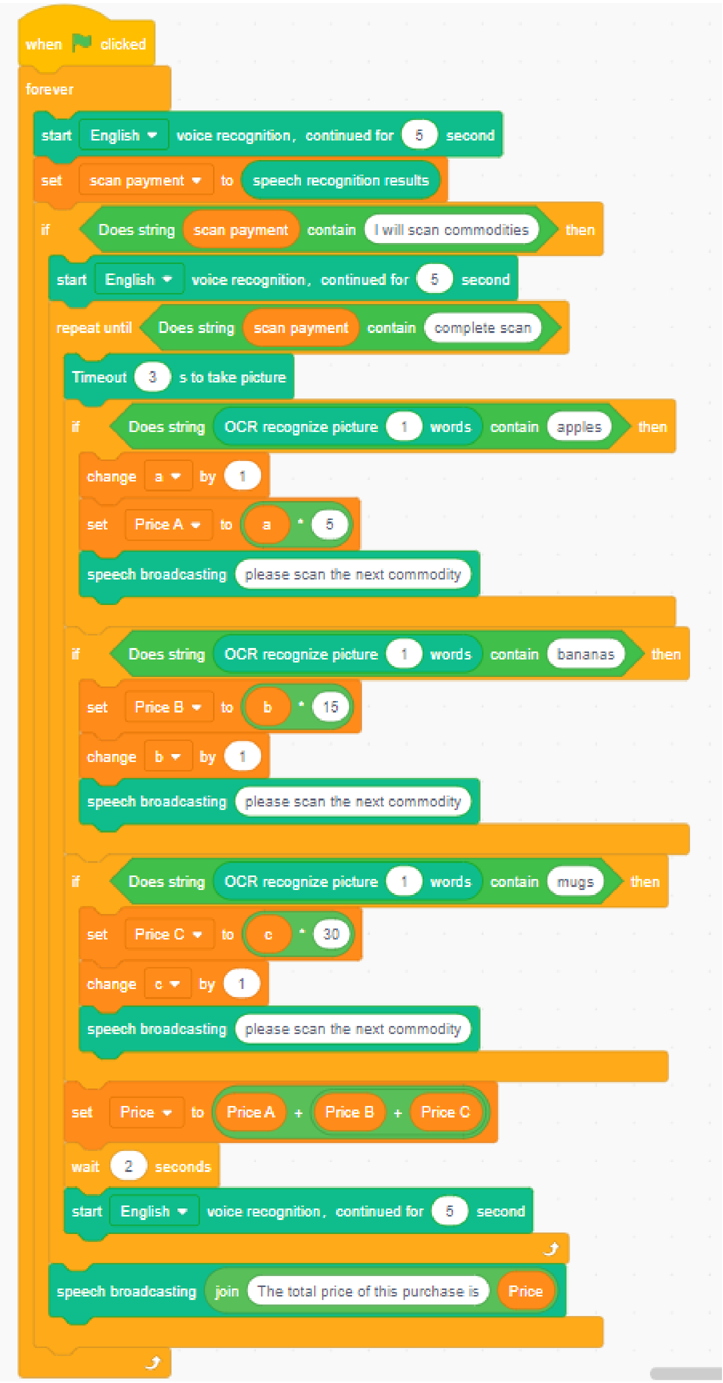




```
when ⚑ clicked
start English ▾ voice recognition, continued for 5 second
set scan payment ▾ to speech recognition results
if Does string scan payment contain I will scan commodities then
    start English ▾ voice recognition, continued for 5 second
    repeat until Does string scan payment contain complete scan
        Timeout 3 s to take picture
        if Does string OCR recognize picture 1 words contain apples then
            change a ▾ by 1
            set Price A ▾ to a * 5
            speech broadcasting please scan the next commodity
        if Does string OCR recognize picture 1 words contain bananas then
            set Price B ▾ to b * 15
            change b ▾ by 1
            speech broadcasting please scan the next commodity
        if Does string OCR recognize picture 1 words contain mugs then
            set Price C ▾ to c * 30
            change c ▾ by 1
            speech broadcasting please scan the next commodity
        set Price ▾ to Price A + Price B + Price C
        wait 2 seconds
        start English ▾ voice recognition, continued for 5 second
    speech broadcasting join The total price of this purchase is Price
```

Figure 8.14 Calculating the total price

Step 18: Repeat the above execution. Automatic pricing process takes place not just once. Therefore, the robotic arm must repeat the speech recognition, commodity scanning, commodity name recognition, calculation of quantity and price of commodities, and total price broadcasting of this purchase, as shown in Figure 8.15.



```
when ⚑ clicked
forever
  start English ▾ voice recognition, continued for 5 second
  set scan payment ▾ to speech recognition results
  if Does string scan payment contain I will scan commodities then
    start English ▾ voice recognition, continued for 5 second
    repeat until Does string scan payment contain complete scan
      Timeout 3 s to take picture
      if Does string OCR recognize picture 1 words contain apples then
        change a ▾ by 1
        set Price A ▾ to a * 5
        speech broadcasting please scan the next commodity
      if Does string OCR recognize picture 1 words contain bananas then
        set Price B ▾ to b * 15
        change b ▾ by 1
        speech broadcasting please scan the next commodity
      if Does string OCR recognize picture 1 words contain mugs then
        set Price C ▾ to c * 30
        change c ▾ by 1
        speech broadcasting please scan the next commodity
      set Price ▾ to Price A + Price B + Price C
      wait 2 seconds
      start English ▾ voice recognition, continued for 5 second
    speech broadcasting join The total price of this purchase is Price
```

Figure 8.15 Repeating the execution

7. Summary

The key steps to design the automatic pricing:

8. Self-Assessment

Check the completed content in the experiment task. Tick (√) the completed items, and circle (○) the uncompleted items.

Assessment Content	Completion Status
I’ve skillfully used the speech recognition instruction	
I’ve skillfully used the OCR instruction	
In the programming process I’ve created and used the variable	
I’ve known the pricing method and grasped the related logic operations	
I’ve used the branch structure and the loop structure	
I’ve programmed the automatic pricing	

Task 2: Design Automatic Payment

Through programming we can automatically calculate the price of commodities purchased. Next, we will try automatic payment: send the speech message to remind the robot to collect the identity information of the payer; after broadcasting the speech message on the total price of the purchase, make automatic payment as soon as confirming the correct information of the payer.

1. Observation

The key steps to analyze automatic payment are:

(1) Use the camera at the end of the robotic arm to collect face data, and ______________________________ .

(2) Broadcast the speech message to inform the shopper of his total purchase price.

(3) Before deducting money, reconfirm the matching of the face information, and give a speech message: ________________ .

(4) When the face information is matched, deduct the ___________ of the correspond person.

(5) When the face information is not matched, give a speech message.   .

2. Steps

(1) Prepare Hardware

Step 1: Prepare the experiment equipment, power on the robotic arm, and connect it to the computer.

Step 2: Set the position of the robotic arm in the same way as in task 1. Power on the equipment, adjust the position and height of the end of the robotic arm, and set the direction of the camera direction on the arm.

(2) Design Program

Step 1: Analyze the automatic payment process, read the flow chart, as shown in Figure 8.16.

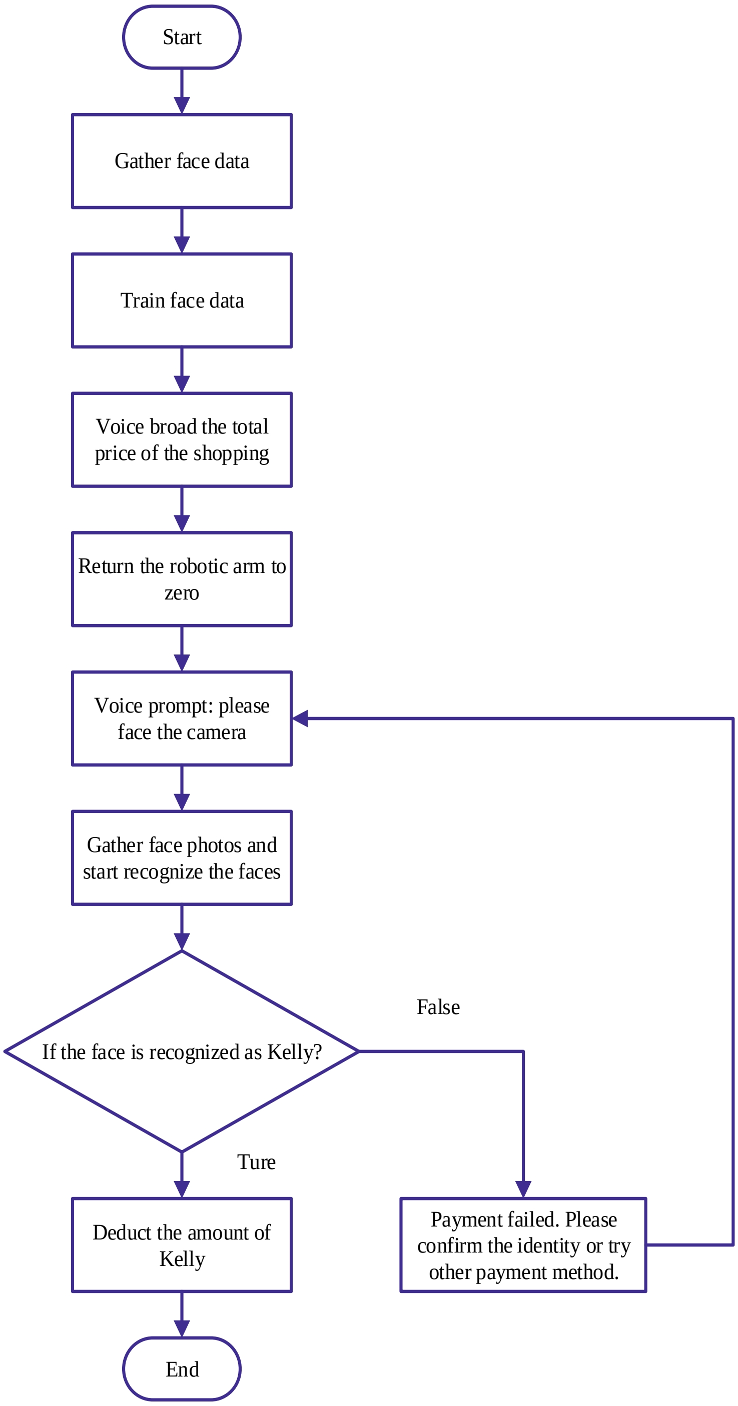

Start
Gather face data
Train face data
Voice broad the total price of the shopping
Return the robotic arm to zero
Voice prompt: please face the camera
Gather face photos and start recognize the faces
If the face is recognized as Kelly?
False
Ture
Deduct the amount of Kelly
Payment failed. Please confirm the identity or try other payment method.
End

Figure 8.16 Automatic payment flow chart

Step 2: Enable DobotScratch software, select and connect the device Magician Lite; and add the AI Extension module.

Step 3: Collect and train face data.

Review: Which steps do collection and training of face data involve?

In Scratch, the programming can: collect the face data of John and Kelly, and accurately recognize their faces. Note that you need to adjust the camera direction.

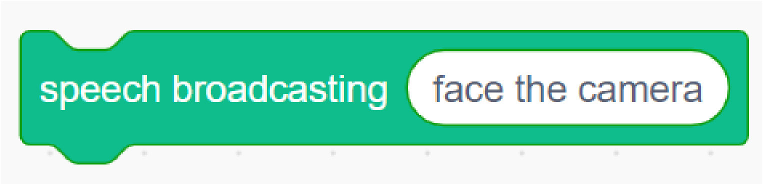
Step 4: Click the **Events** tab, and drag the "when space key pressed" block to the code area.

Step 5: Broadcast the speech message of the total price of this purchase.

In Scratch, the programming can: broadcast the speech message of the total purchase price.

Step 6: Set the robotic arm to home after the equipment broadcasts the speech message of the total purchase price. Click the **Motion** tab, and drag the "Home" block to connect the preceding block.

Step 7: Give a speech message. Click the **AI** tab, drag the "speech broadcasting hello" block, and set the speech broadcast content as "face the camera". See Figure 8.17 for the editing method.



speech broadcasting face the camera

Figure 8.17 Speech broadcasting block

Step 8: Use face scan payment.

As per the learned knowledge, summarize the steps to make face scan payment, and write related steps.

In Scratch, the programming can: collect face photos before recognizing the faces, judge whether the face recognition result involves Kelly. If yes, automatically deduct the amount of Kelly; otherwise, recognize the faces of other persons again or choose other payment methods.

3. Summary

The steps to collect face data:

4. Self-Assessment

Check the completed content in the experiment task. Tick (√) the completed items, and circle (○) the uncompleted items.

Assessment Content	Completion Status
I’ve skillfully used the speech recognition instruction	
I’ve skillfully used the face recognition instruction	
I’ve used the branch structure and the loop structure	
I’ve programmed automatic payment	

Experiment 9

Present Gifts

Team Name: Team member: Date:

Overview

In order to thank consumers for their support, stores often give gifts when they buy more goods. In this experiment, we will design a small gift presentation system with the robotic arm. There are two small tasks. Task 1 is to give personalized gifts to different consumers through face recognition. Task 2 is to give different gifts to males and females by gender recognition from facial images.

Objective

- Master face recognition instructions by programming for gift presentation.
- By programming for gift presentation, understand how to use control, operation, and variable instructions.
- Understand the application of face attributes by programming for gender recognition and gift presentation by gender.

Equipment

Equipment Picture	Name	Quantity
	Dobot Magician Lite Robotic arm	1

Equipment Picture	Name	Quantity
	Suction cup kit	1
	Camera	1
	Type-C cable	1
	Power adapter	1
	Blocks with different labels	3
	Three character cards	A group
	Small table (optional)	3

Requirements

➢ Take care when using electricity.

- Before the experiment, check whether the experimental equipment is complete and intact. If there is any omission or damage, please report to the teacher.
- Any specific operations in the experiment shall be performed according to the experiment manual. If you have any questions, please promptly ask the teacher.
- During the experiment, the joints will start to work as the robotic arms are powered on. In that case, do not move the joints of the robotic arm hard if you do not press the unlock key.
- Report any device fault during the experiment to your teacher in a timely manner, and handle it with guidance from your teacher.
- Arrange all devices after the experiment. You shall not leave the lab before check by the group leader..

Task 1: Present Gifts

Suppose there are three people and each can get a personalized gift. The present for John is a gift parcel; that for Kelly is a small cake; and that for the teacher is a bunch of flowers. Task 1 is to deliver personalized gifts to them based on the face recognition technology.

1. Analysis

Analyze the steps to give gifts to the three people based on the face recognition technology, and fill in the blanks in the table below according to the prompts and your analysis.

Step 1: Prepare Hardware

Step 2: Programming: Initialization Face image acquisition and → recognition → ____

____

____

______________________(Complete the remaining steps as prompted)

2. Steps

(1) Prepare Hardware

Step 1: After you have prepared the experimental instruments, power up the robotic arm and connect it to the computer.

Step 2: Build scenarios. The placement diagram is shown in Figure 9.1. If you do not choose to print character images on the tables, you can stick the character cards to them.

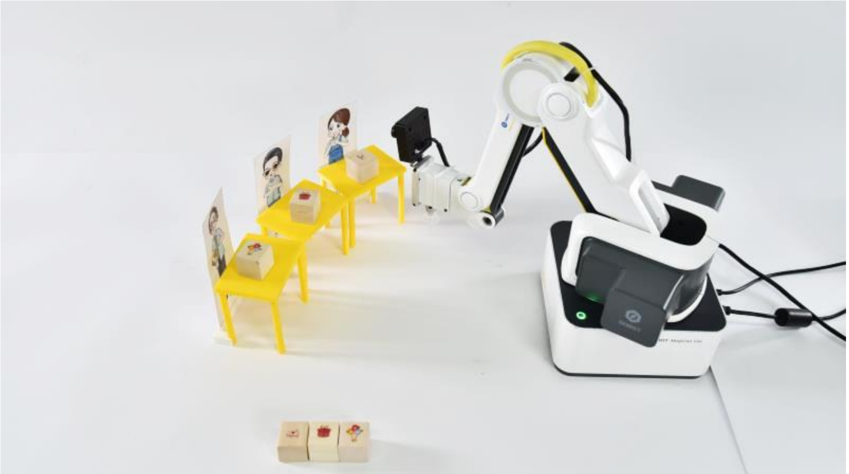


Figure 9.1 Placement diagram

(2) Design Program

Step 1: Draw a flow chart by analyzing the gift presentation programming process, as shown in Figure 9.2.

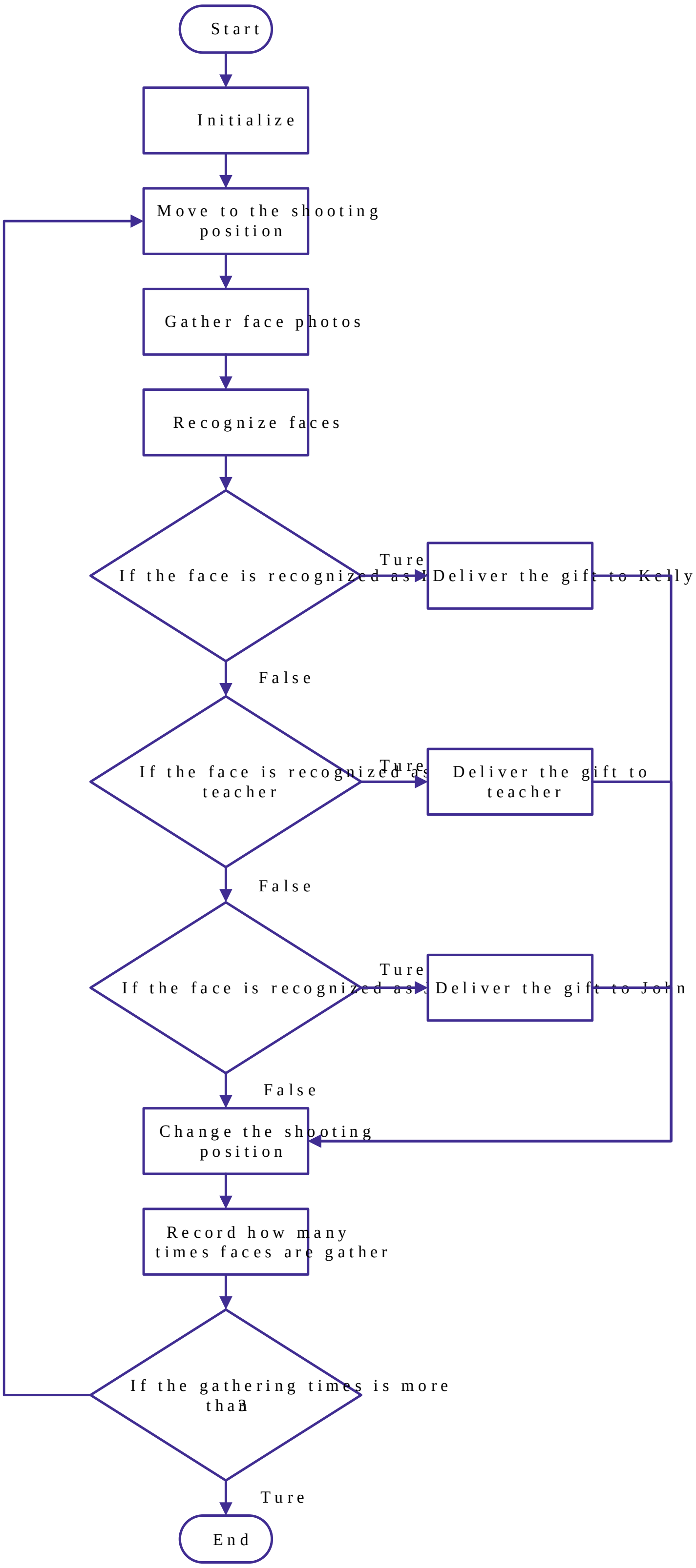

Start
Initialize
Move to the shooting position
Gather face photos
Recognize faces
If the face is recognized as
Ture
Deliver the gift to Kelly
False
If the face is recognized as teacher
Ture
Deliver the gift to teacher
False
If the face is recognized as
Ture
Deliver the gift to John
False
Change the shooting position
Record how many times faces are gather
If the gathering times is more than 3
Ture
End

Figure 9.2 Flow Chart

Step 2: Create new face data. Click the **AI** tab, and click **New face data** under the Face Recognition header, as shown in Figure 9.3.

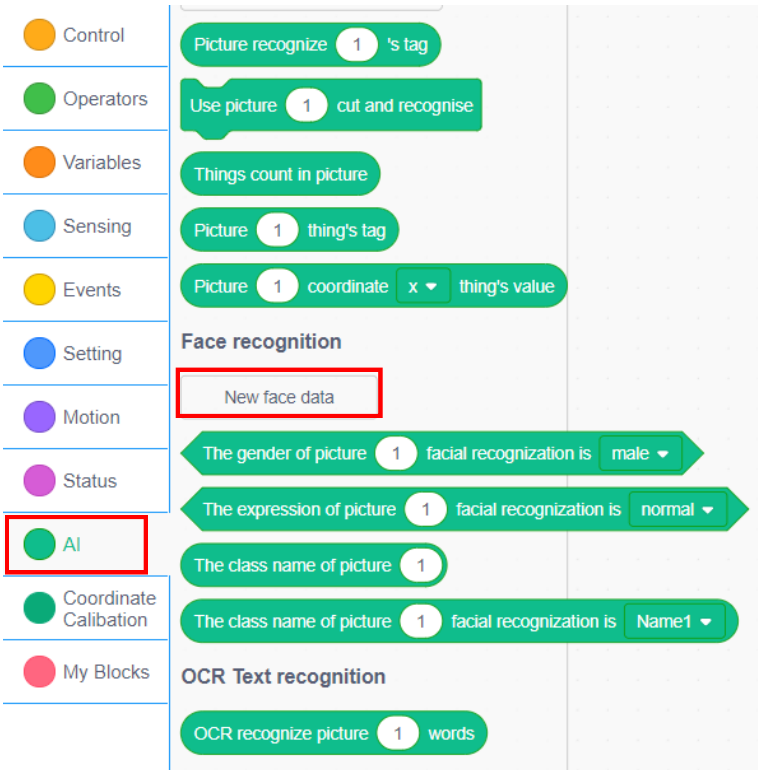
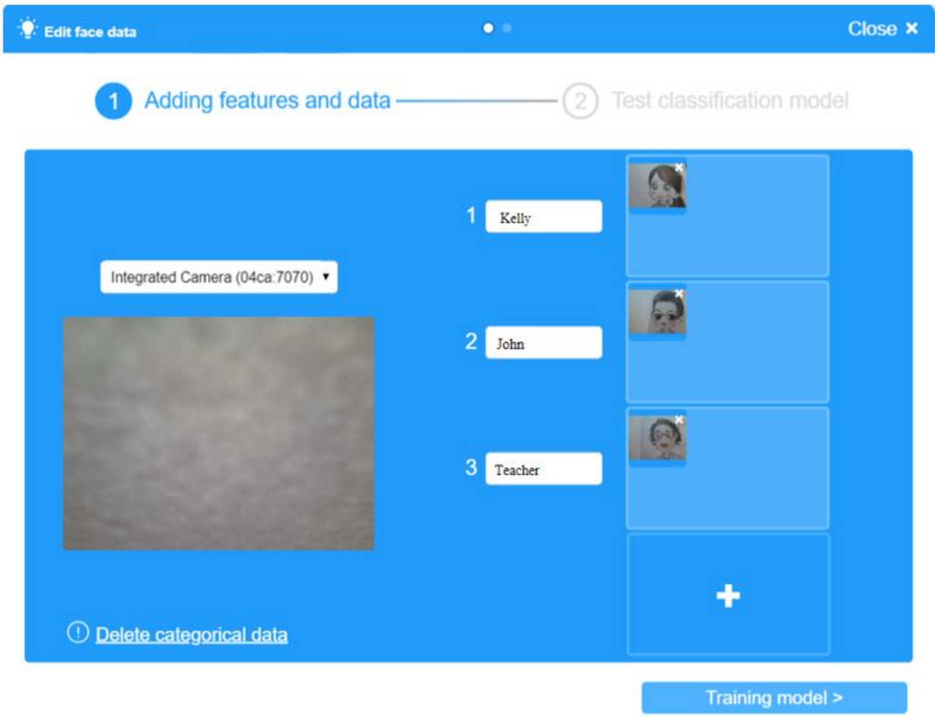

Control
Operators
Variables
Sensing
Events
Setting
Motion
Status
AI
Coordinate Calibration
My Blocks
Picture recognize 1 's tag
Use picture 1 cut and recognise
Things count in picture
Picture 1 thing's tag
Picture 1 coordinate x thing's value
Face recognition
New face data
The gender of picture 1 facial recognization is male
The expression of picture 1 facial recognization is normal
The class name of picture 1
The class name of picture 1 facial recognization is Name1
OCR Text recognition
OCR recognize picture 1 words

Figure 9.3 Creating new face data

Step 3: Add features and data, that is, face data. Take pictures for the faces to be recognized, store them and set labels, as shown in Figure 9.4.



Integrated Camera (04ca:7070)

1 Kelly

2 John

3 Teacher

Figure 9.4 Storing face data

Step 4: Train the model. Store the face data, and start training, that is, a learning process, as shown in Figure 9.5.

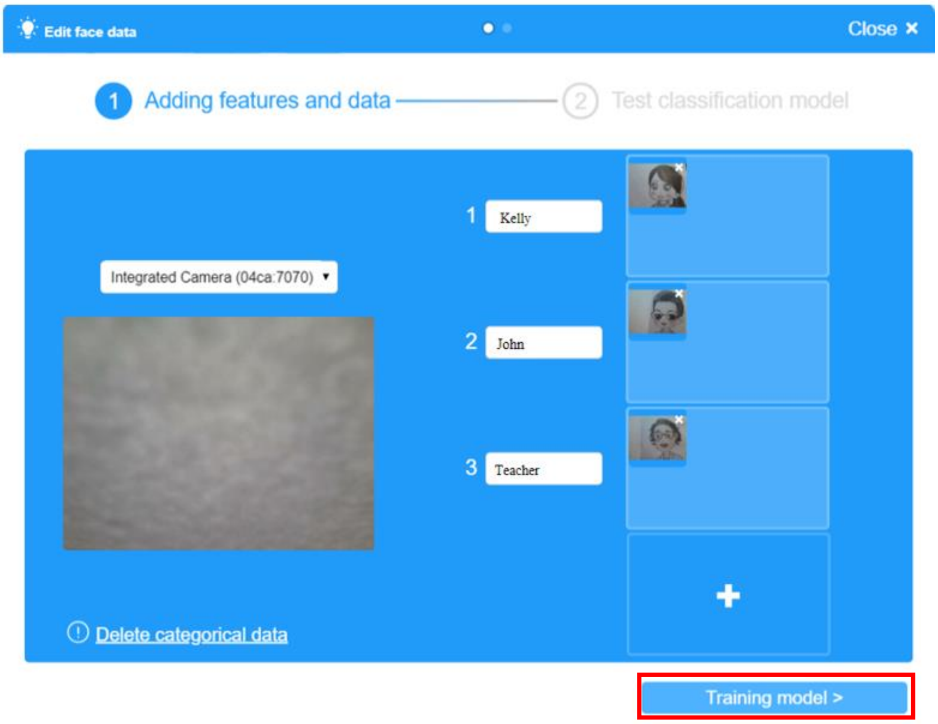
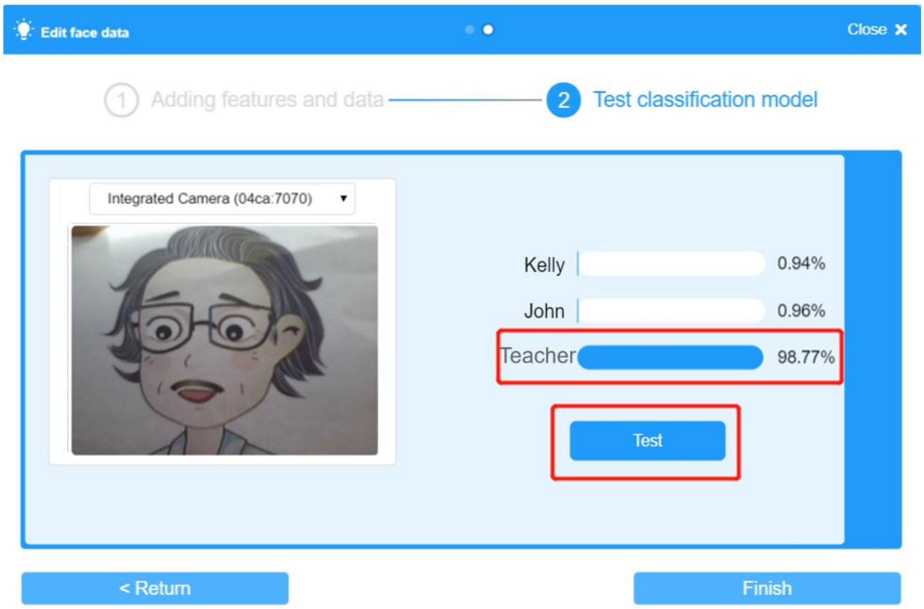

Edit face data
Close ✕
1 Adding features and data
2 Test classification model
Integrated Camera (04ca:7070)
1 Kelly
2 John
3 Teacher
Delete categorical data
Training model >

Figure 9.5 Training the model

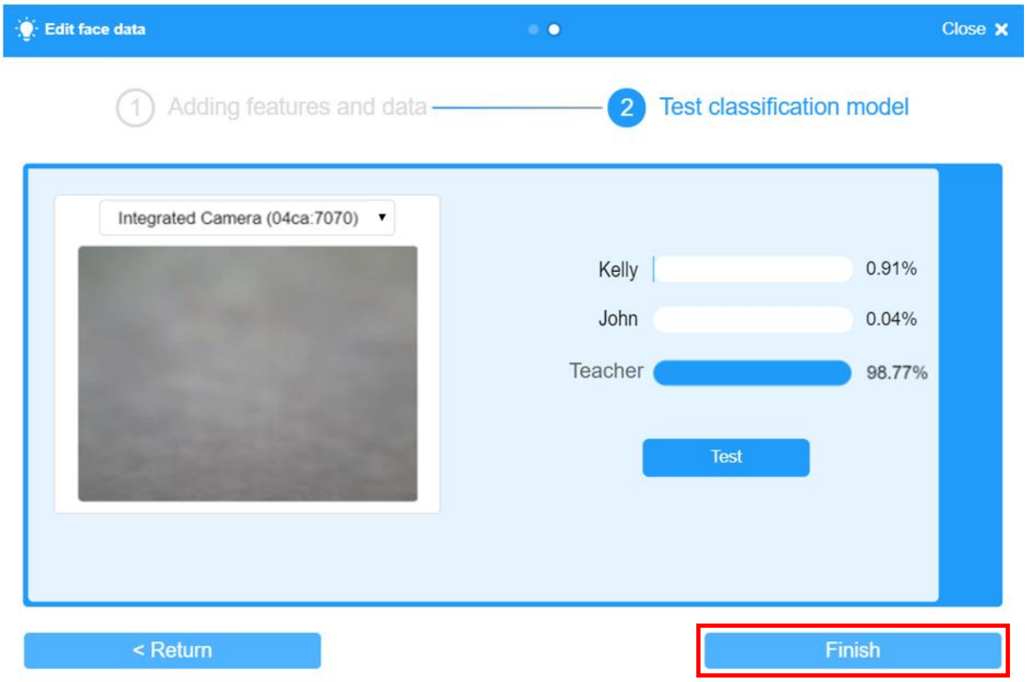
Step 5: Test the accuracy of face recognition. Randomly select a face photo to aim it at the camera, and then click **Test** to see if the face recognition technology can correctly Kelly, the teacher and John, as shown in Figure 9.6.


Edit face data
Close ✕
1 Adding features and data
2 Test classification model
Integrated Camera (04ca:7070)
Test
< Return
Finish

Kelly	0.94%
John	0.96%
Teacher	98.77%

Figure 9.6 Testing Accuracy of Face Recognition

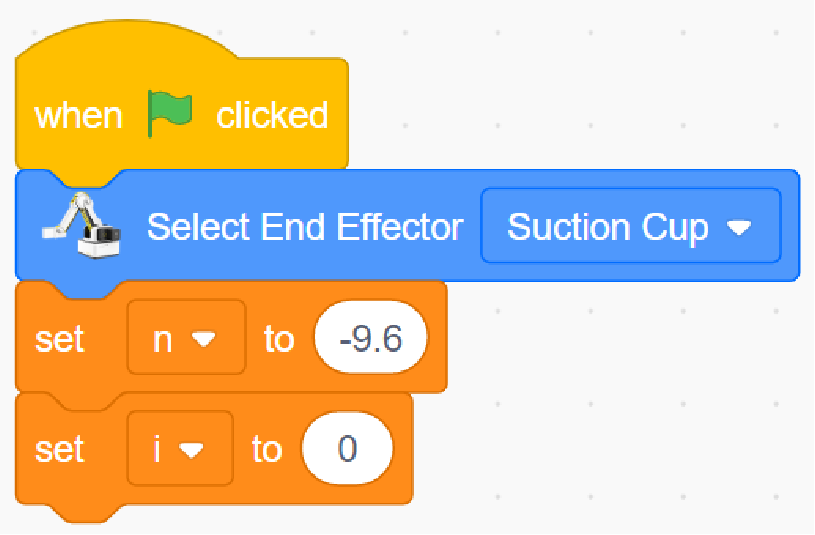
Step 6: Click **Finish** to start programming, as shown in Figure 9.7.


Edit face data
Close ✕
1 Adding features and data
2 Test classification model
Integrated Camera (04ca:7070) ▾
Test
< Return
Finish

Kelly 0.91%
John 0.04%
Teacher 98.77%

Figure 9.7 Finishing creating face data

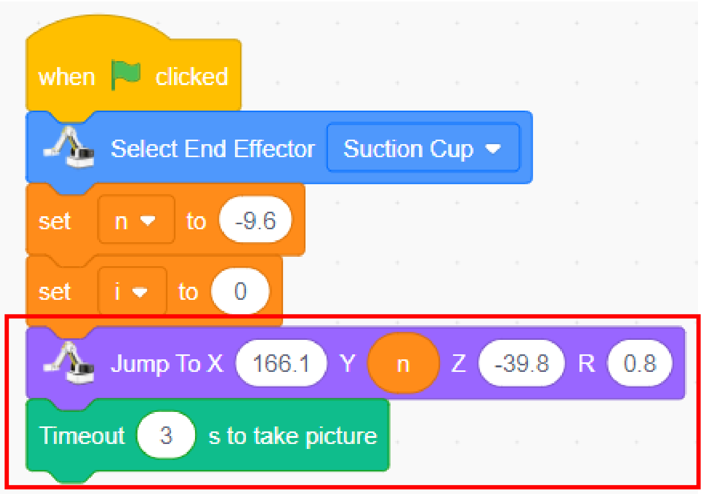
Step 7: Create *n* and *i* variables. Initialize the *n* variable to -9.6 and *i* to 0. *n* is the Y-coordinate of the photographing position, as shown in Figure 9.8.



```
when 🏁 clicked
Select End Effector  Suction Cup ▾
set n ▾ to -9.6
set i ▾ to 0
```

Figure 9.8 Initializing variables

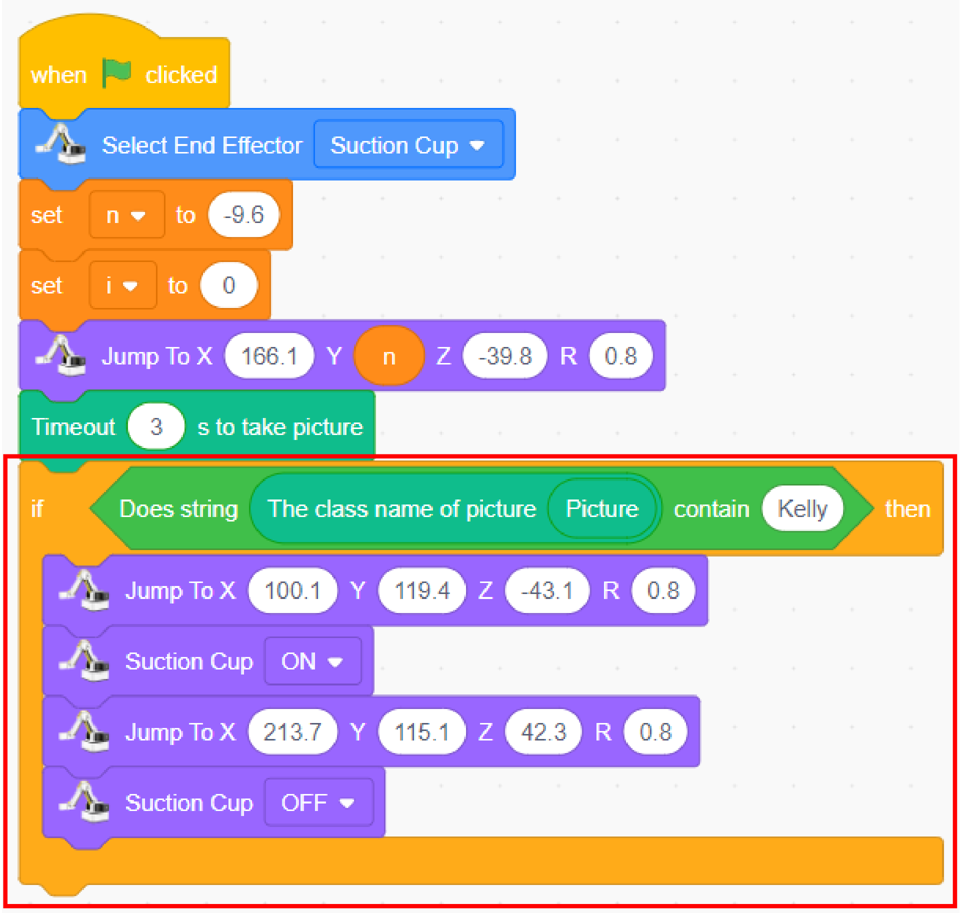
Step 8: Move to the photographing position, which varies with the position of a face image card. Set the auto-photographing time to 3 seconds, as shown in Figure 9.9.



```
when 🏴 clicked
Select End Effector [Suction Cup ▾]
set [n ▾] to (-9.6)
set [i ▾] to (0)
Jump To X (166.1) Y (n) Z (-39.8) R (0.8)
Timeout (3) s to take picture
```

Figure 9.9 Moving to the position for auto-photographing

Step 9: Recognize the face and match the recognition result with a person's name to judge if the identified person is Kelly. If yes, the robotic arm performs the action of presenting the gift, as shown in Figure 9.10.



```
when 🏴 clicked
Select End Effector [Suction Cup ▾]
set [n ▾] to (-9.6)
set [i ▾] to (0)
Jump To X (166.1) Y (n) Z (-39.8) R (0.8)
Timeout (3) s to take picture
if <Does string (The class name of picture (Picture)) contain (Kelly)> then
    Jump To X (100.1) Y (119.4) Z (-43.1) R (0.8)
    Suction Cup [ON ▾]
    Jump To X (213.7) Y (115.1) Z (42.3) R (0.8)
    Suction Cup [OFF ▾]
```

Figure 9.10 Judging if the identified person is Kelly

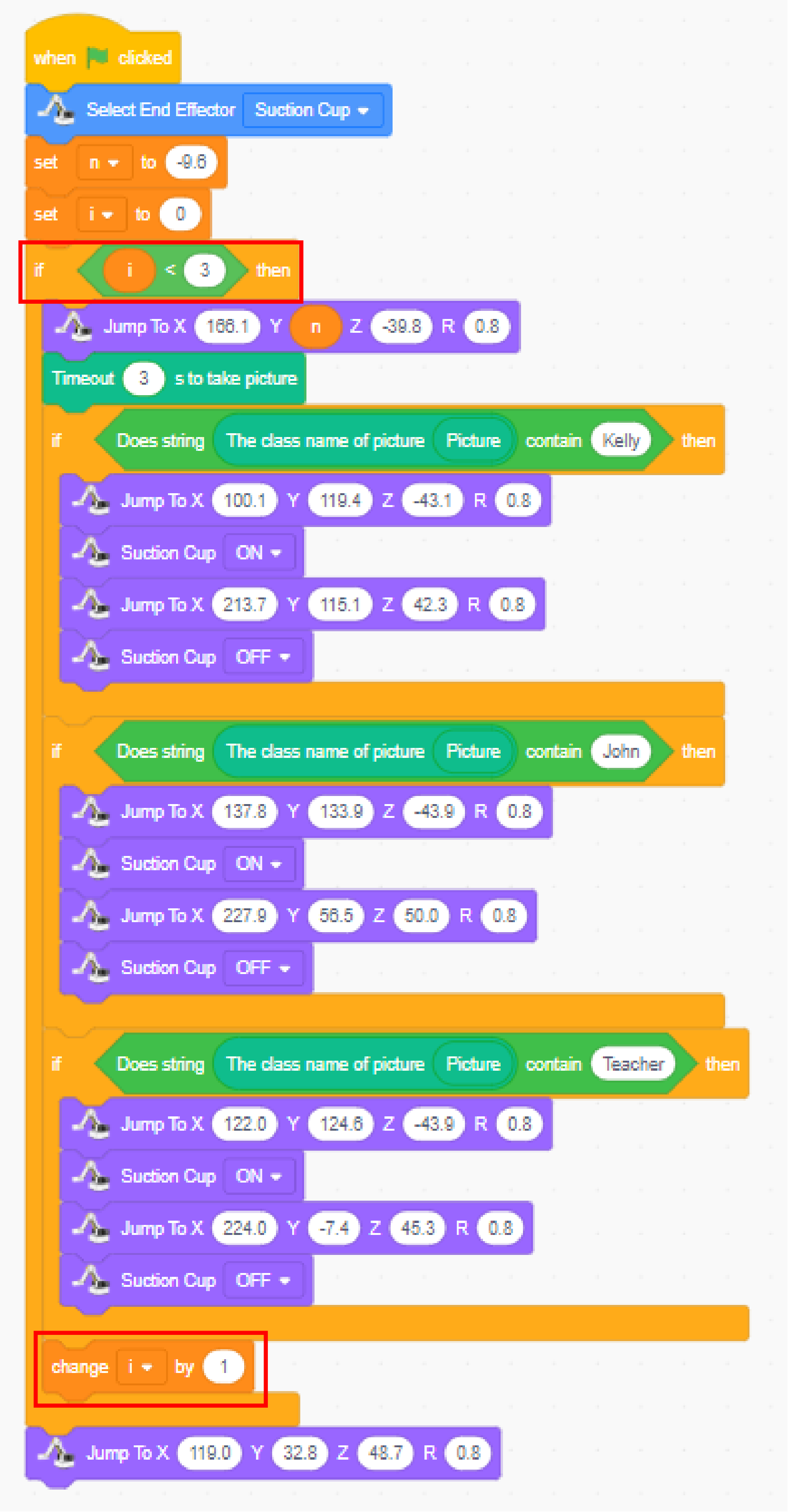
Step 10: Judge if the face recognition result involves John.

☛ In Scratch, program to judge if the face recognition result involves John. If yes, the robotic arm presents a gift.

Step 11: Judge if the face recognition result involves the teacher.

☛ In Scratch, program to judge if the face recognition result involves the teacher. If yes, the robotic arm presents a gift.

Step 12: In this experiment, present gifts to three people, that is, need to make judgment three times, as shown in Figure 9.11.



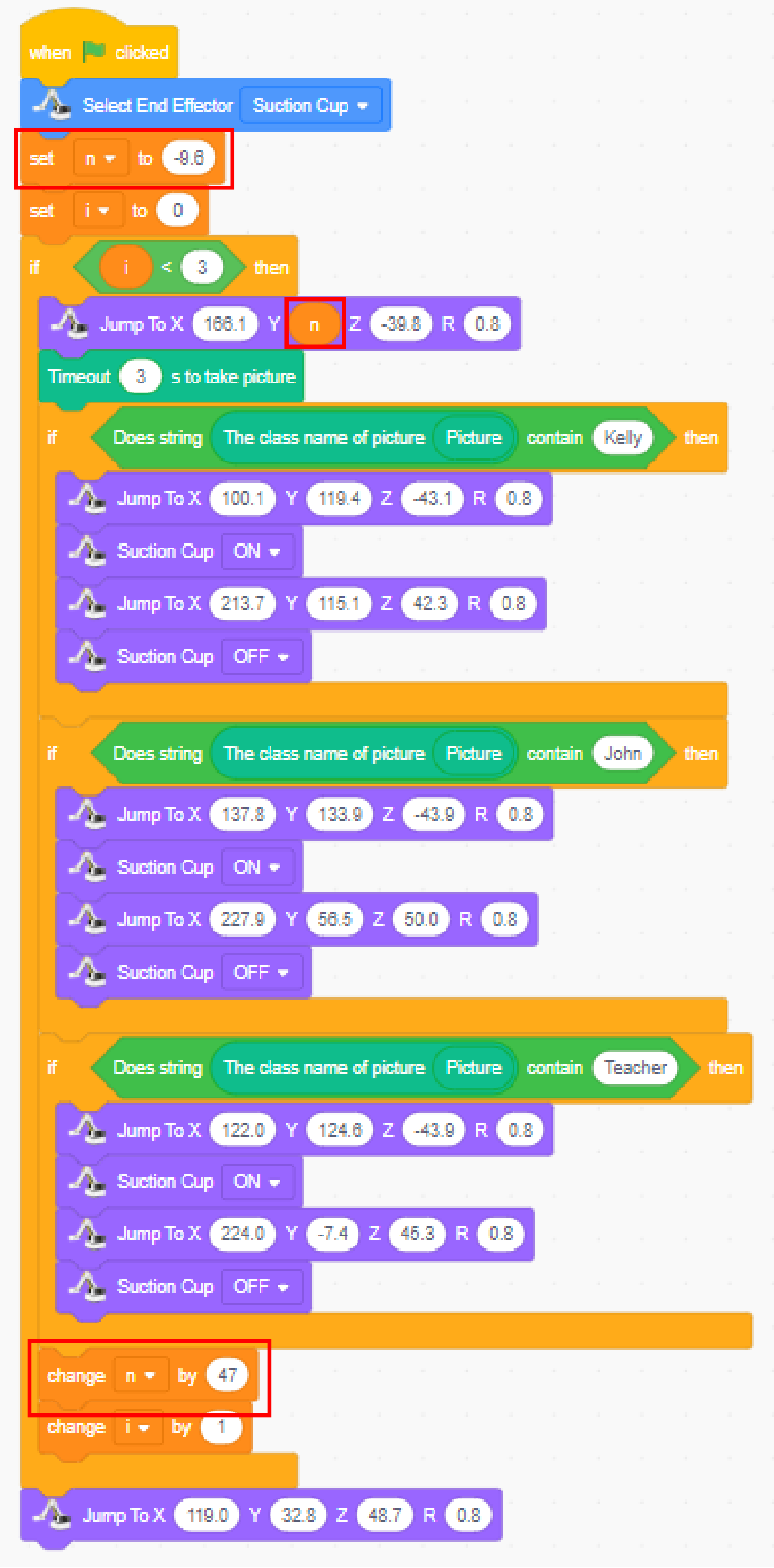
```
when 🏁 clicked
Select End Effector [Suction Cup ▾]
set [n ▾] to (-9.6)
set [i ▾] to (0)
if <(i) < (3)> then
    Jump To X (166.1) Y (n) Z (-39.8) R (0.8)
    Timeout (3) s to take picture
    if <Does string (The class name of picture (Picture)) contain (Kelly)> then
        Jump To X (100.1) Y (119.4) Z (-43.1) R (0.8)
        Suction Cup [ON ▾]
        Jump To X (213.7) Y (115.1) Z (42.3) R (0.8)
        Suction Cup [OFF ▾]
    if <Does string (The class name of picture (Picture)) contain (John)> then
        Jump To X (137.8) Y (133.9) Z (-43.9) R (0.8)
        Suction Cup [ON ▾]
        Jump To X (227.9) Y (56.5) Z (50.0) R (0.8)
        Suction Cup [OFF ▾]
    if <Does string (The class name of picture (Picture)) contain (Teacher)> then
        Jump To X (122.0) Y (124.6) Z (-43.9) R (0.8)
        Suction Cup [ON ▾]
        Jump To X (224.0) Y (-7.4) Z (45.3) R (0.8)
        Suction Cup [OFF ▾]
    change [i ▾] by (1)
Jump To X (119.0) Y (32.8) Z (48.7) R (0.8)
```

Figure 9.11 Repeating the

execution

The variable *i* records the times of repeated execution. *i* = 0 indicates the first program execution; *i* = 1 indicates the second execution; *i* = 2 indicates the third execution; and *i* = 3 (> 2) indicates that the system jumps out of the cycle.

Step 13: Arrange the placement position, as shown in Figure 9.1. This indicates that the X and Z coordinates of the photographing position change little, so we do not modify them. However, the Y coordinate changes a lot. According to the placement, we conclude that the change to Y coordinate is about 47 (we can change this value based on actual situation). Figure 9.12 shows the programming condition.



```
when 🏴 clicked
Select End Effector [Suction Cup ▾]
set [n ▾] to (-9.6)
set [i ▾] to (0)
if <(i) < (3)> then
    Jump To X (166.1) Y (n) Z (-39.8) R (0.8)
    Timeout (3) s to take picture
    if <Does string (The class name of picture (Picture)) contain (Kelly)> then
        Jump To X (100.1) Y (119.4) Z (-43.1) R (0.8)
        Suction Cup [ON ▾]
        Jump To X (213.7) Y (115.1) Z (42.3) R (0.8)
        Suction Cup [OFF ▾]
    if <Does string (The class name of picture (Picture)) contain (John)> then
        Jump To X (137.8) Y (133.9) Z (-43.9) R (0.8)
        Suction Cup [ON ▾]
        Jump To X (227.9) Y (56.5) Z (50.0) R (0.8)
        Suction Cup [OFF ▾]
    if <Does string (The class name of picture (Picture)) contain (Teacher)> then
        Jump To X (122.0) Y (124.6) Z (-43.9) R (0.8)
        Suction Cup [ON ▾]
        Jump To X (224.0) Y (-7.4) Z (45.3) R (0.8)
        Suction Cup [OFF ▾]
    change [n ▾] by (47)
    change [i ▾] by (1)
Jump To X (119.0) Y (32.8) Z (48.7) R (0.8)
```

Figure 9.12 Setting the photographing position

How do we modify the program to make the frequency of repeated execution 10?

3. Summary

Summarize the steps to create new face data:

4. Self-Assessment

Check the completed content in the experiment task. Tick (√) the completed items, and circle (◯) the uncompleted items.

Assessment Content	Completion Status
I’ve created new face data	
I’ve collected and recognized face images three times	
I’ve judged face images three times	
I’ve presented personalized gifts to the three people	

Task 2: Apply Face Attributes

By gender recognition from face images, present different gifts to males and females, such as Gift A for males and Gift B for females.

1. Analysis

Analyze the steps to present gifts based on gender recognition, and fill in the blanks in the table below according to the prompts and your analysis. .

Step 1: Building hardware

Step 2: Programming:____

____

____

________________________(Complete the remaining steps as prompted)

2. Steps

Complete this task independently. See Figure 9.13 for the program flow chart.

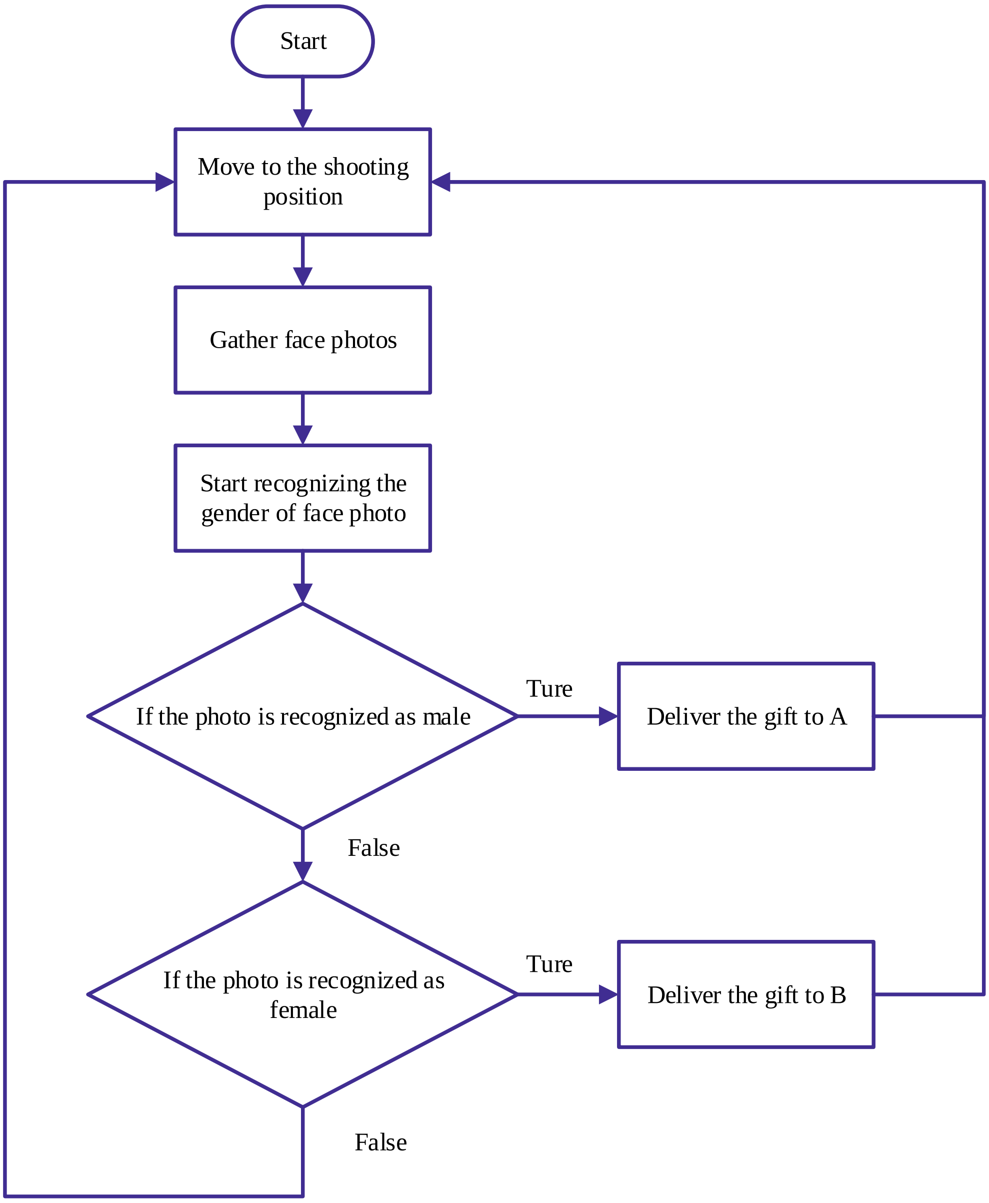

Start
Move to the shooting position
Gather face photos
Start recognizing the gender of face photo
If the photo is recognized as male
Ture
Deliver the gift to A
False
If the photo is recognized as female
Ture
Deliver the gift to B
False

Figure 9.13 Flow Chart

☛ In task 2, you can add the function of speech message in addition to face attributes. Please have a try.

3. Summary

(Summarize what you have learned in this task)

4. Self-Assessment

Check the completed content in the experiment task. Tick (√) the completed items, and circle (◯) the uncompleted items.

Assessment Content	Completion Status
I’ve built the hardware	
I’ve collected face images and recognized face attributes	
I’ve programmed for presenting Gift A to males based on face recognition	
I’ve programmed for presenting Gift B to females based on face recognition	