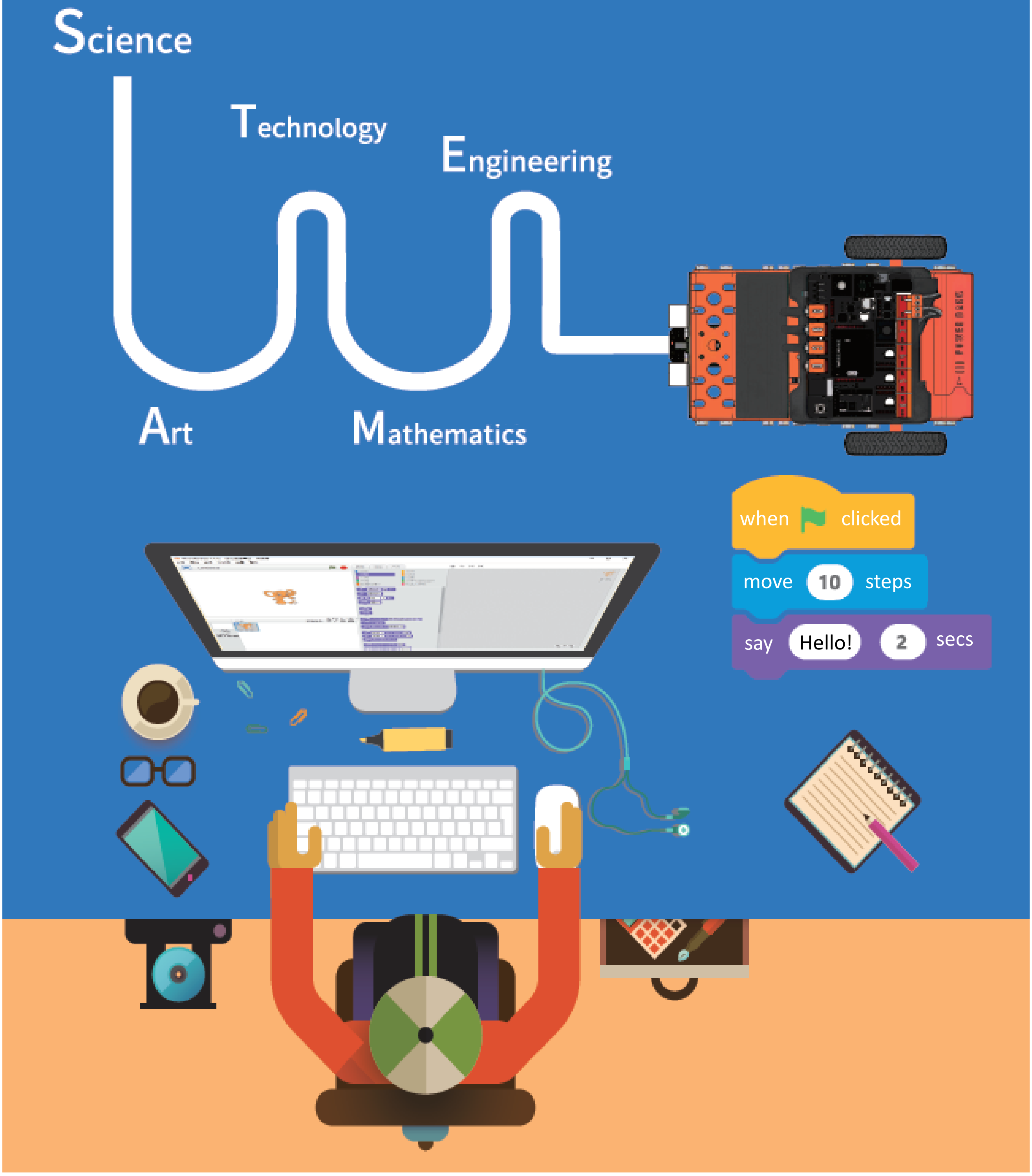
Introduction and Improvement on Scratch Hardware Programming

WeeeBot STEM Robot



CONTENT

Chapter 1 Walk into the World of Robots

Summary

As the first lesson for the student to contact robots, its primary purpose is to let students understand the concept of robots, know the relevant software and hardware knowledge, and experience the fun of learning robots.

Learning Objectives

To know about the common robots in life and understand the concept of robots.
To understand the software and hardware knowledge of WeeeBot robots and experience the default mode of WeeeBot robots.

Course Time

90 minutes.

Pre-class Preparation

WeeeBot robots, computers with WeeeCode software installed, remote controls, and line-following maps.

Teaching Process

I Introduction from life (20min)

1. Students share the robots seen on TV and in their lives and feel that robots can be seen everywhere in our lives.
On TV: Transformers, Baymax, Doraemon, etc.
In life: Robot vacuum cleaner, restaurant service robots, guest guiding robots, etc.

2. Q: Do a robot have to look like humans? What can be called a robot?
Conclusion: A robot is a mechanical device that can automatically execute its work. It not only can accept human commands, but also can run preset programs.

3. Q: what are the responsibilities of a robot?
Conclusion: The robot in English is “robot”, originally “robo”, meaning “slave”, namely human servant, so the responsibility of a robot is to assist human's work.

4. Watch videos and experience the rapid development of robot technology.
By letting students view the relevant videos of the application of robots in life, teachers can let students experience that robots are developed rapidly and are gradually infiltrating into our life.

Ⅱ Introduction of Hardware knowledge (20min)

1. To make students understand the hardware components of robots via analogy.

As we know, a robot not only needs to run the prewritten programs but also to perform a task. It is conceivable that a robot needs a mechanical part to move as well as a software part to give orders. To better understand robots, we take WeeeBot Line-following Knight as an example to find out the software and hardware parts of robots.

The hardware parts of the robots we used is a jeep body robot combined by many parts such as aluminum structures, various sensors, the combination of motor and wheel, vibrant and LED panel module module, mainboard with full functions, and power-supplying battery holder, as shown in FIG. 1-1.

FIG. 1-1

In this robot, each of its components is very similar to the structure of our human body, which plays a crucial role, as shown in FIG. 1-2.

Robot	Human	Role
Mainboard	Brain	Receive and process information from outside and give command to body.
Motors and wheels	Legs and arms	Responsible for body movement.
Battery	Heart	Supply power.
Sensors	Five sense organs	Receive information.
Electronic circuit	Neuron	Link every part of body to form a whole system.

FIG.1-2

2. To understand the composition of the mainboard: Familiarize students with the position of standard port and facilitate the teaching and use of robots at the later stage.

The brain is one of the most critical parts of human beings, which makes people think. Moreover, the mainboard is the most critical part of the robot's structure. It not only can handle the feedback information from the sensor but also can receive our orders to control the movement of the robot. Next, we will know the composition of the robot mainboard, as shown in FIG. 1-3.

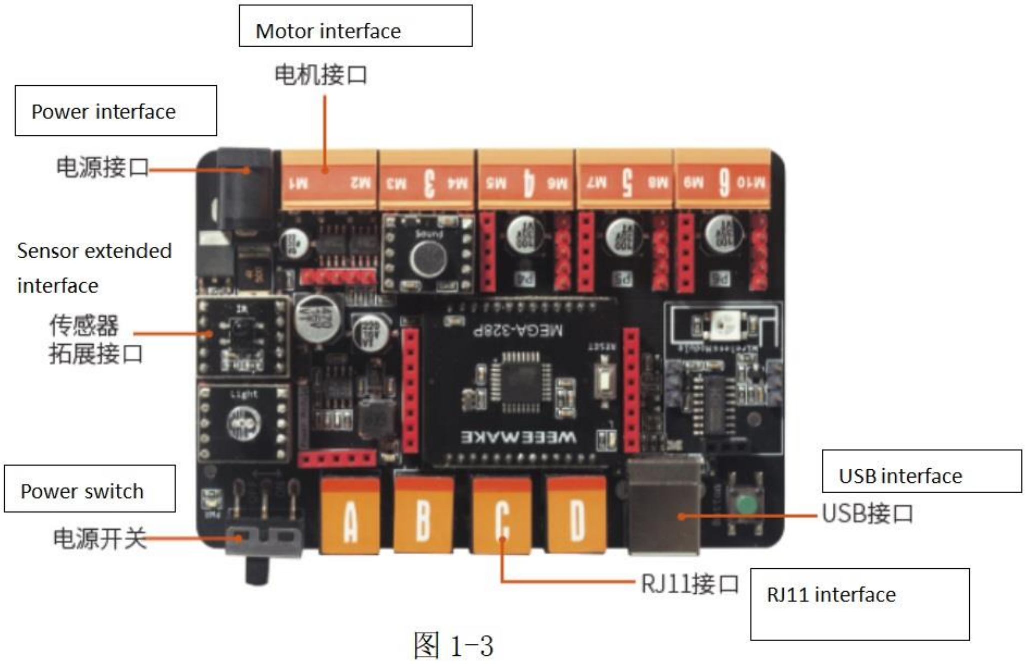


图 1-3

FIG. 1-3

Sensor port: connect external sensors, such as infrared sensors, light sensors, sound sensors and so on.
Power port: a port for the battery to supply power to the mainboard through the power line.
Motor port: a particular port for a motor.
USB port: a port for a computer to connect to the control board through the USB download cable and to upload program.
RJ11 port: a port for common sensors, such as LED panel module, RGB ultrasonic sensor, line-following sensor...
Power switch: to turn on and turn off the power of the robot.

Ⅲ Understanding of Software--- WeeeCode（20min）

1. To understand the programming software port and understand the role of each area in the port.

After knowing the hardware part of the WeeeBot robot, next, we will learn its software. The software of the WeeeBot robot is WeeeCode, which is based on the graphical programming software Scratch 2.0 developed by the Massachusetts Institute of Technology (MIT). It is a robot programming software designed by combining it with the hardware WeeeBot. With this software, we can get various commands to the robot and control it. The software port of WeeeCode is as shown in Figure 1-4.

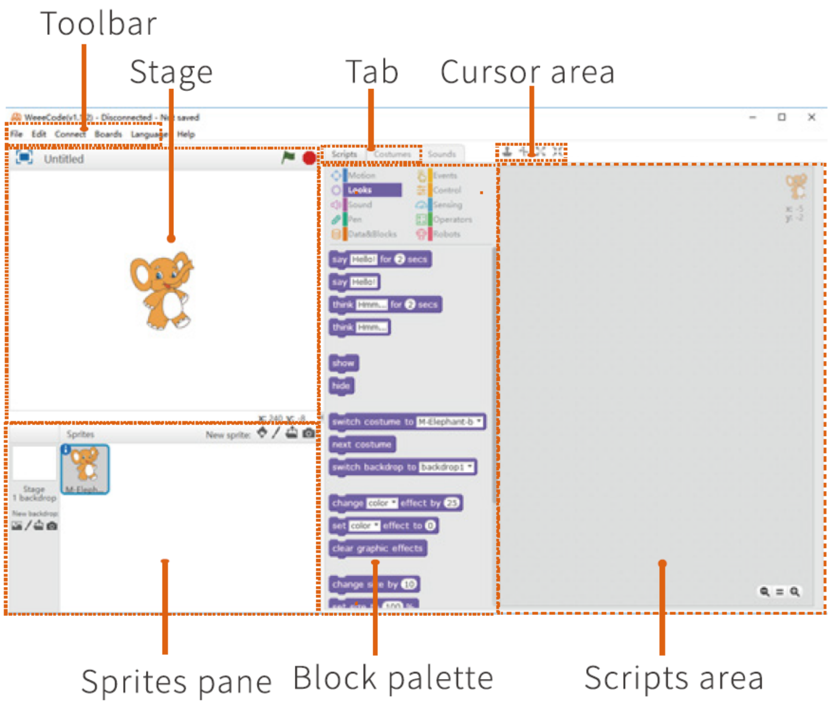


FIG.1-4

Toolbar: An area for operation of project files, software port mode, serial connection and program uploading, mainboard selection, software use language, and software update.

Stage: It is an area for interaction among characters, or between characters and users, and it is also a place to display the running effect of programs.
Control button: green flag button for starting a program and red dot button for stopping the program.

Sprites pane: the area where all the role prototypes are shown, where you can see the name of characters, rotation direction, location, and so on.

Tab: It contains the block palette, custume pane, and the sound pane, to control the action, custome and audio of the sprite.

Block palette: It is here that categories of blocks are color-coded and can be clicked to bring up a new set of blocks that can be dragged into the scripting area to program a sprite or the stage.

Cursor area: It contains copy, delete and zoom out/in buttons, which are used to manipulate the roles on the stage.

Scripts area: Blocks from the Block Palette are dragged into the Scripts Area. These then can be combined with other blocks to form scripts

2. Guide students to write the first program and experience the fun of programming.

Practice:

a. Click the “event” in the block palette and drag the "when the green flag is clicked" block into the script area with the left mouse button.

b. Click the movement block in the block palette, drag the "move 10 steps" block under the "when the green flag is clicked" block in the script area. When the two blocks get close, there will be white highlight hint under the last block, as shown in FIG. 1-5. Then loosen the mouse, and the next will be automatically placed under the last block, as shown in FIG. 1-6.

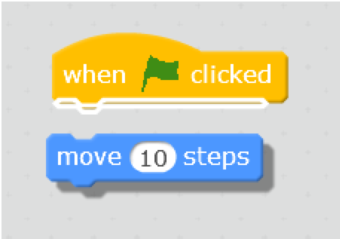


FIG. 1-5

FIG. 1-6

c. Click the “green flag” Green flag or click any block in the block stacks, the program will start to run, and the role will move on the stage.

Ⅳ Experience WeeeBot Line-following Knight (20min)

1. To understand several functions of WeeeBot default mode, cultivate students' interest in learning robots, and provide ideas in programming for students.

After writing the first program, we can find out how exciting and straightforward it is to write a program as it can show the way of thinking. In WeeeBot robot, we have some rules preset. After uploading the factory process of the robot to the robot, we can use the remote control to manipulate the robot, and the function of each key on the remote control is as shown in FIG. 1-7.

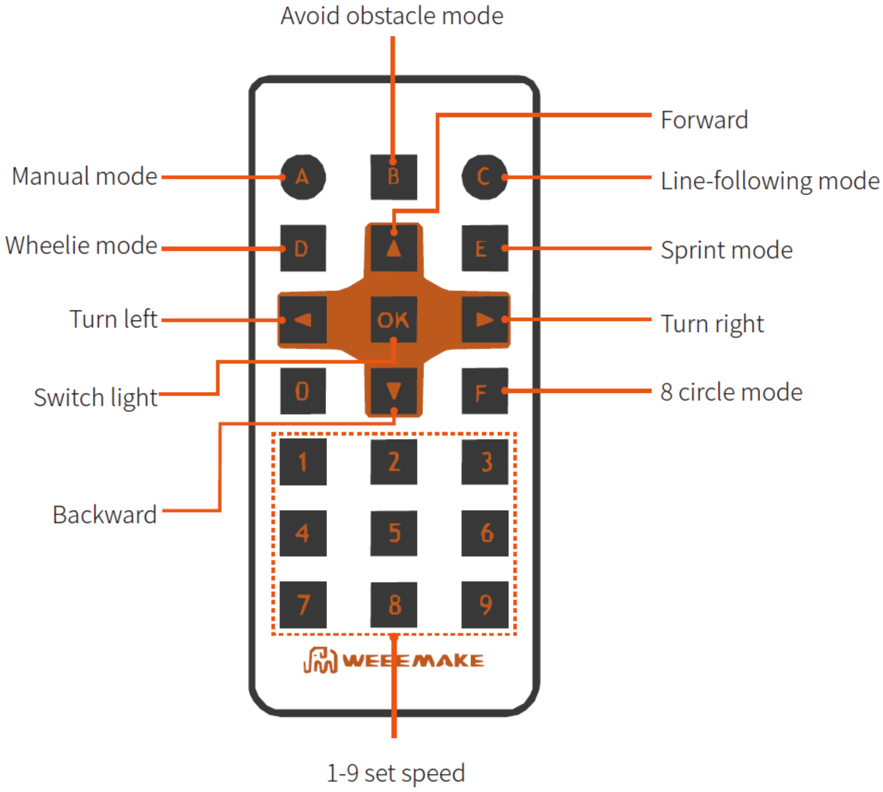


FIG.1-7

Note: when you experience the default mode, please turn on the power switch, and the RGB light and LED panel module will be lit, to ensure that the robot program is the factory default program.

a. Manual mode: Press the "A" key of the remote control, the robot will enter the manual mode. In this mode, press the key of "↑", "↓", "←" and "→", to control the robot to move around accordingly.

b. The obstacle avoidance mode: Press "B" key of the remote control, the robot will enter the obstacle avoidance mode. It will constantly test if there are obstacles in the front in the process of moving forward, and the robot will swerve to avoid any obstacles in the front.

c. Line-following mode: Press "C" key of the remote control, the robot will enter line-following mode. Next open our configured map and put the robot on the black line of the map track, and then the robot can walk along the black line.

d. Wheelie mode: press "D" key of the remote control, the robot will enter Wheelie mode. In this mode, press "↑", "↓", "←" and "→" key on the remote control, the robot will move at high speed. When loosening "↑" or "↓" key after pressing them, the head will turn up. This mode can be used in the sites which have a height difference, so as to turn up the its head for climbing.

e. Sprint mode: Press the "E" key of the remote control, the robot will enter the Sprint mode. In this mode, press the "↑", "↓", "←" and "→" key on the remote control, the robot will move at the fastest speed.

f. Light switch mode: Press "OK" key of the remote control, the robot will enter the light switch mode. When pressing the key for the first time, RGB light color in the ultrasonic sensor began to change. When pressing the key for the second time, the color of the RGB light will be determined.

g. 8-circle mode: Press the "F" key of the remote control, the robot will enter the 8-shape mode. In this mode, the left and right wheels of the robot will rotate by turns and take the 8-shape route.

Exploration: In the remote control, the function of "0~9" key are not explained, try to find out by yourself.

Conclusion: In manual mode, wheelie mode and Sprint mode, press the "1~9" key of the remote control, which can adjust the movement speed of the robot. The larger the number is, the higher the speed will be. The "0" key does not make a sound when pressed, indicating that we haven't set the function of the "0" key.

V Thinking and Conclusion (10min)

1. What have we learned from this chapter?

Have Known about the robot in life, understood the software and hardware knowledge of WeeeBot robot, written the first program, and experienced the default mode of the WeeeBot robot.

2. What are the similarities between the components of WeeeBot robot and the components of our body? What role do they play respectively?

Robot	Human	Role
Mainboard	Brain	Receive and process information from outside and give command to body.
Motors and wheels	Legs and arms	Responsible for body movement.
Battery	Heart	Supply power.
Sensors	Five sense organs	Receive information.
Electronic circuit	Neuron	Link every part of body to form a whole system.

3. What default modes does WeeeBot have?

Manual mode, obstacle avoidance mode, line-following mode, wheelie mode, sprint mode, light switch mode, 8-shape mode.

Chapter 2 Colorful Neon Lights

Summary
After students preliminarily contact and understand the robot in lesson 1, lesson 2 is to let students learn how to program online to control robots, understand the working principle of RGB LEDs and master its control method, and explore the law of the RGB LED color matching.

Learning Objectives
To understand the origin of the three primary colors, understand the working principle of RGB LED and master its control method, master the ratio of the three primary colors and then get the methods of mixing different colors.
To master the method of online programming and debugging of a robot.
To know the structures commonly used for programming---sequence structure and loop structure.

Course Time
90 minutes.

Pre-class Preparation
WeeeBot robots, computers with WeeeCode software installed, drawing software, and USB cables.

Teaching process

I Introduction from life (20min)

1. Students share the scenes of the rainbow seen in their lives and describe the colors of the rainbow.
The place where the rainbow often appears: beside a waterfall, the sky after raining. The appearance of the rainbow is often accompanied by seven colors, and the colors from top to the bottom of the bow are: red, orange, yellow, green, cyan, blue and purple.

Q: why is it so easy to see rainbows after heavy rain or beside waterfalls? How is it formed?
Conclusion: Water and sunlight under certain conditions can form a rainbow. Colorless and transparent sunlight is a mixture of seven kinds of visible light, namely, red, orange, yellow, green, cyan, blue and purple. They all have their color, but when mixed and propagated, it will become colorless and transparent white. When they shine obliquely together along a direction from air to water (or other transparent material), the direction of propagation will have deflection---refraction, but the light deflection

degree of water (other transparent material) to different colors is different. Then the original mixed colors are separated, and so rainbow is formed.

2. To know the origin of the three primary colors of color light through short stories.

The great scientist Isaac Newton, who discovered the famous three laws of mechanics, also discovered the color secret of light. When a beam of white light passes through a triple prism, the white light will be decomposed into seven kinds of colored light, and the seven colors in a sequence are: red, orange, yellow, green, blue, indigo, and purple, and the sequence is fixed. This also is what people often say "seven-color-light."

Then, after discovering the chromatic dispersion secret of light, Newton began to deduce: Since white light can be decomposed and synthesized, can the seven colors be decomposed or synthesized? Through constant experiment and calculation, Newton concluded: In the seven kinds of colors, only red, green and blue three colored light cannot be decomposed, and other four kinds of colors all can be made by mixing the three kinds of color in different proportion. Therefore, red, green and blue are known as "light of three primary colors " or "three primary colors of colored light," as shown in FIG. 2-1.

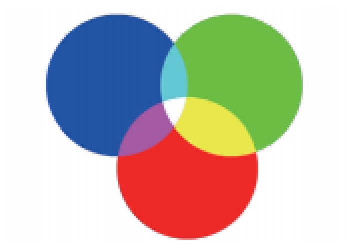


FIG. 2-1

Ⅱ Know about Module and Use (30min)

1. Master the usage of RGB LED.

Do you still remember that there was a light on the mainboard able to be switched into many kinds of colors when we were experiencing WeeeBot robot in our last class? That module is called RGB LED, and the location is as shown in FIG. 2-2. “R” is the abbreviation of English word Red, “G” is the abbreviation of English word Green, “B” is the abbreviation of English word Blue, and the Red, Green, and Blue is what we call the primary colors of colored light. RGB LEDs can be controlled by blocks to make it emit light with different colors.

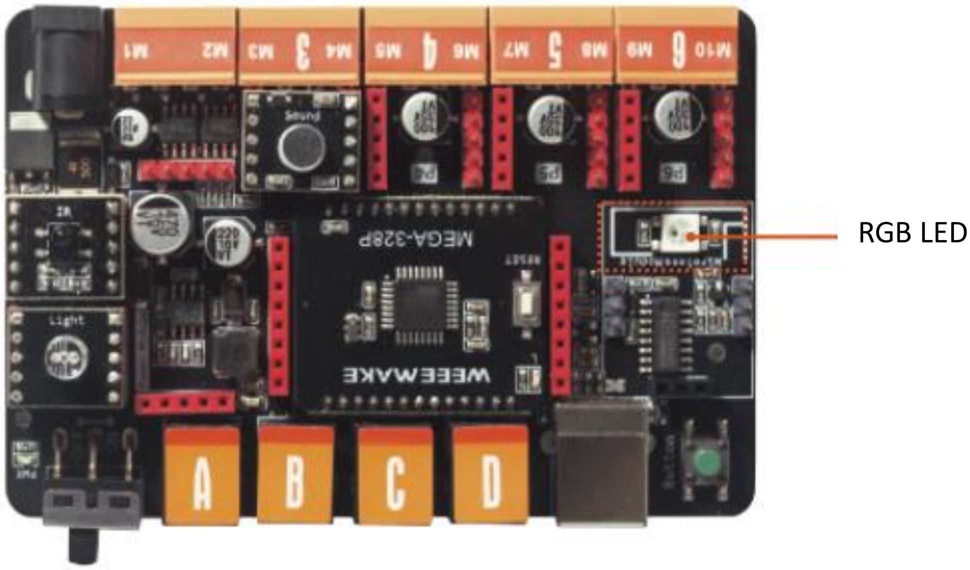


FIG. 2-2

We can find the control block of RGB LED in the “Robot” category, as shown in FIG. 2-3.

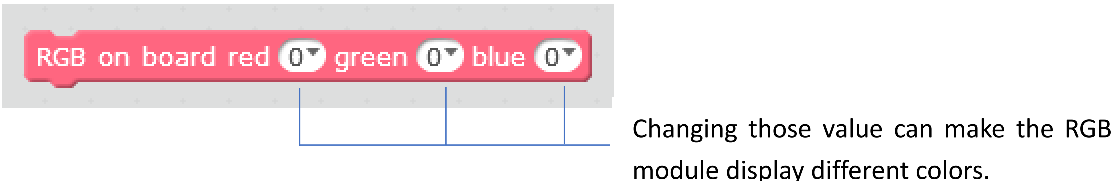


FIG. 2-3

2. Master the method of online programming and debugging.

If the values of RGB blocks are all determined, how to give orders to the robot? Then a USB cable is needed, to upload our program to the control board and build up the communication between the computer and the robot.

Practice:

a. Connect to the USB cable, and select the port of the mainboard connected to the USB cable in the "connect" button of the toolbar, as shown in FIG. 2-4.

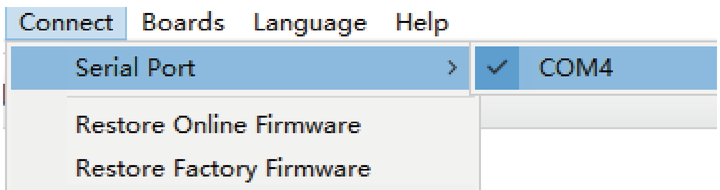


FIG. 2-4

c. Select WeeeBot software platform in the "Boards" button, as shown in FIG. 2-5.

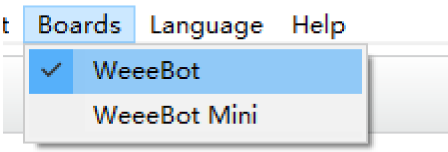


FIG. 2-5

c. Click "restore online firmware" in the "connect" button. If the port of "successful uploading" appears, as shown in FIG. 2-6, the robot will enter the online programming mode.

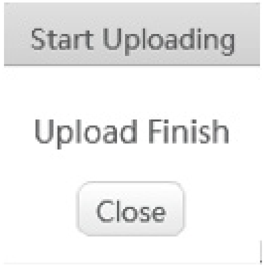


FIG. 2-6

d. Click the “green flag” button or arbitrary position of block stacks at this time, the program will start running, and the RGB LED on the mainboard will emit light corresponding to the RGB values. The process and effect are as shown in FIG. 2-7 and 2-8.

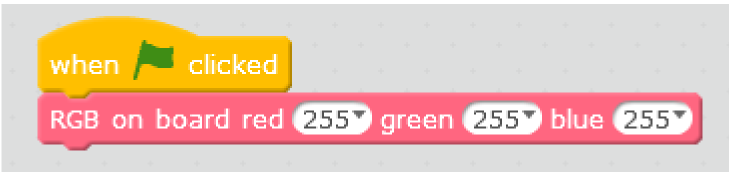


FIG. 2-7

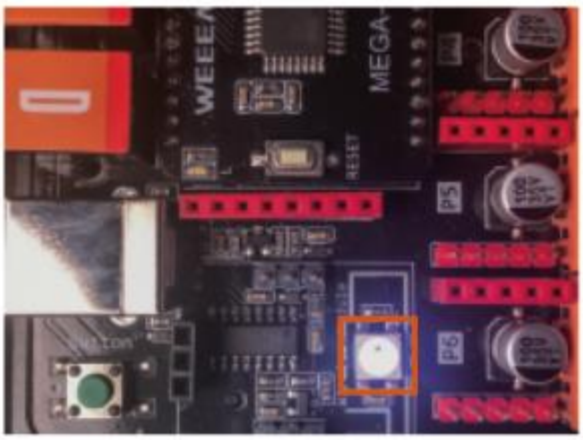


FIG. 2-8

Note:

a) There is no need to use the "when the green flag is clicked" block when programming, clicking the RGB control block can also make the program run and the purpose of adding "when the green flag is clicked" block is to make students keep good programming habits.

b) If you want to let the RGB LEDs to display other colors, please click red dot button first to stop the running program, modify the RGB values, click the green flag button to run the program, and then you can see the color of the RGB LEDs after the values modification. The advantage of the online mode is you can immediately see the effect of program modification.

c) Check the COM serial port: Right click "my computer," select "management," select "device manager," click "port (COM and LPT),", and then check "CH340" port, it can be known that connection port is COM3, as shown in FIG. 2-9.

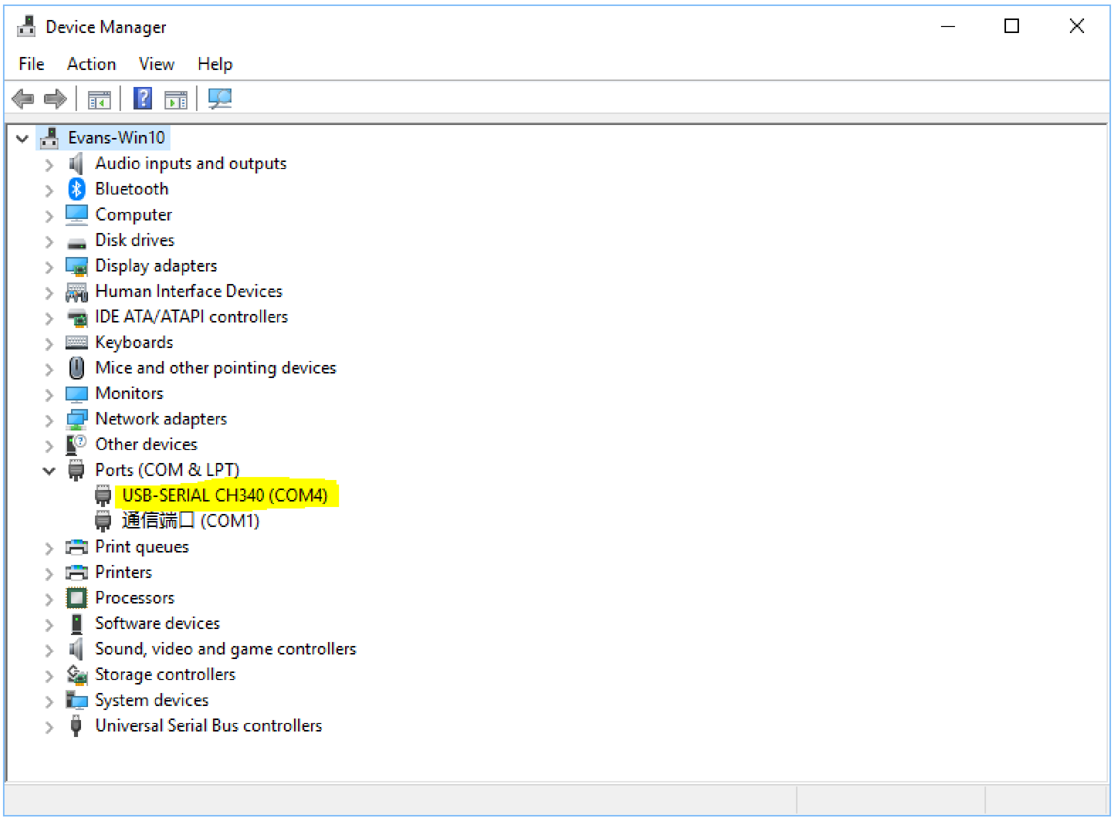


FIG. 2-9

d) When using hardware to interact with computer in online mode, the mainboard must always be connected to the computer. If the USB cable is loose in the process of use, the connection will be disconnected. It only will need to stabilize the USB cable and then to restore the serial port connection.

3. To debug the RGB values of seven colors by analogy with drawing software.

If we want the RGB LED to display rainbow colors, how do we know the RGB value of each color? There is a drawing software in the computer, and we can view the RGB value of each color by drawing software.

a. Press the keyboard “Windows +r” to open the Run box and enter "mspaint. exe", as shown in FIG. 2-10, you can open the drawing software, as shown in FIG.2-11.

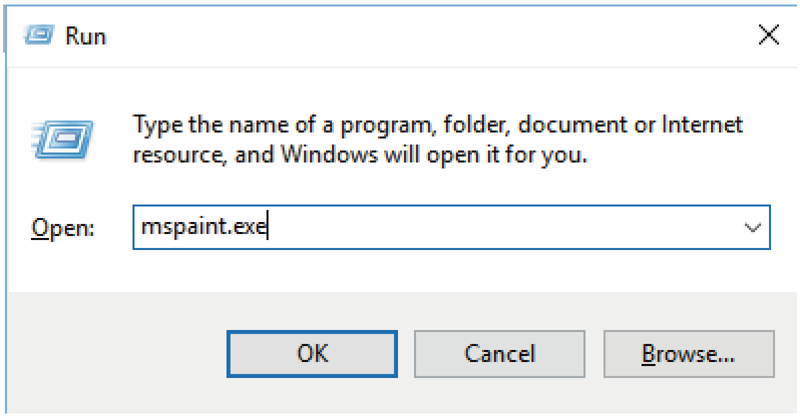


FIG. 2-10

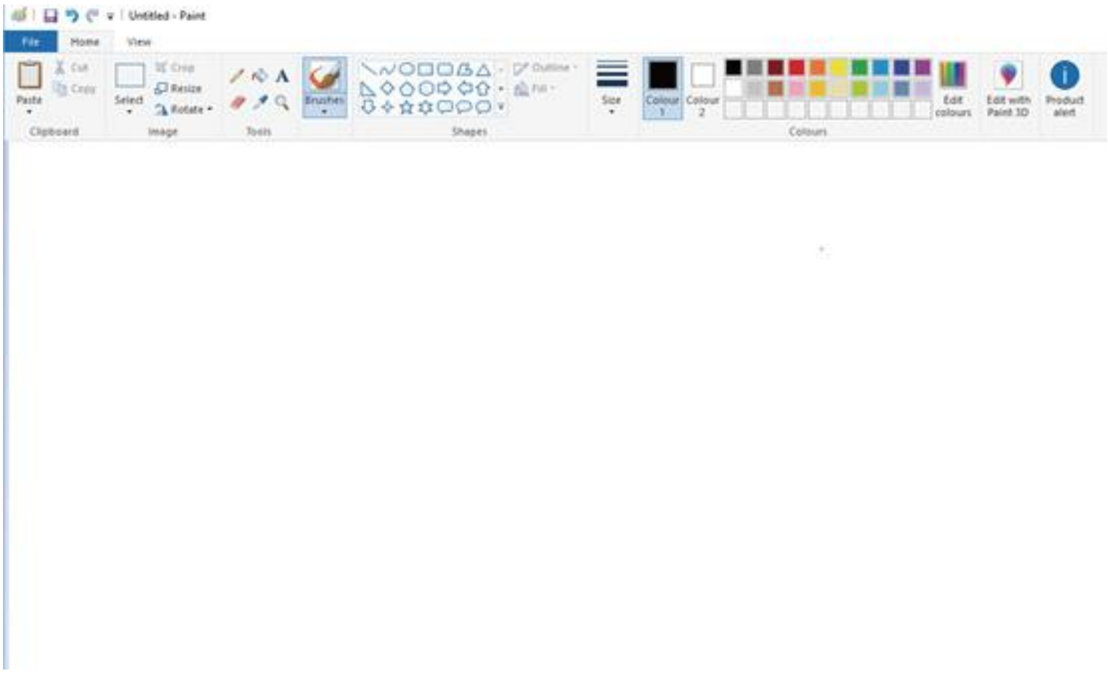


FIG. 2-11

b. Click "Edit color" in the upper-right corner of the paint software, and the interface shown in shown in FIG. 2-12. In the Red, Green, and Blue boxes at the bottom right, we can change the value of each color, of which the range is 0-255. We can also see the color generated by the three colors in the color display box.

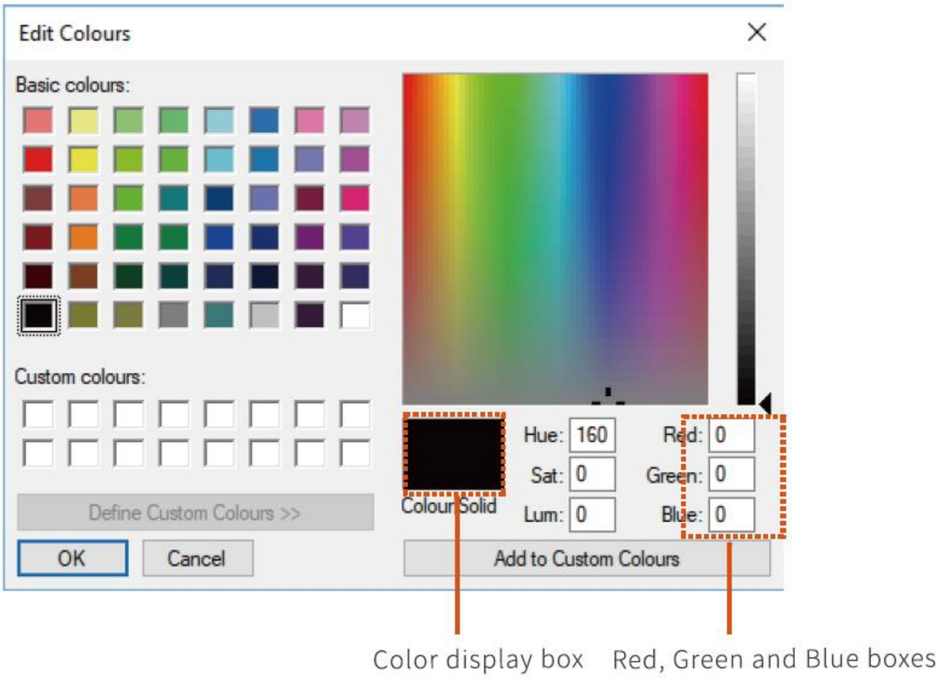


FIG. 2-12

c. We can also select an approximate color directly in the color menu and drag the color bar to adjust the specific color. If the color is determined in the color display box, the RGB value of the color can be obtained in the Red, Green, and Blue boxes, as shown in FIG. 2-13.

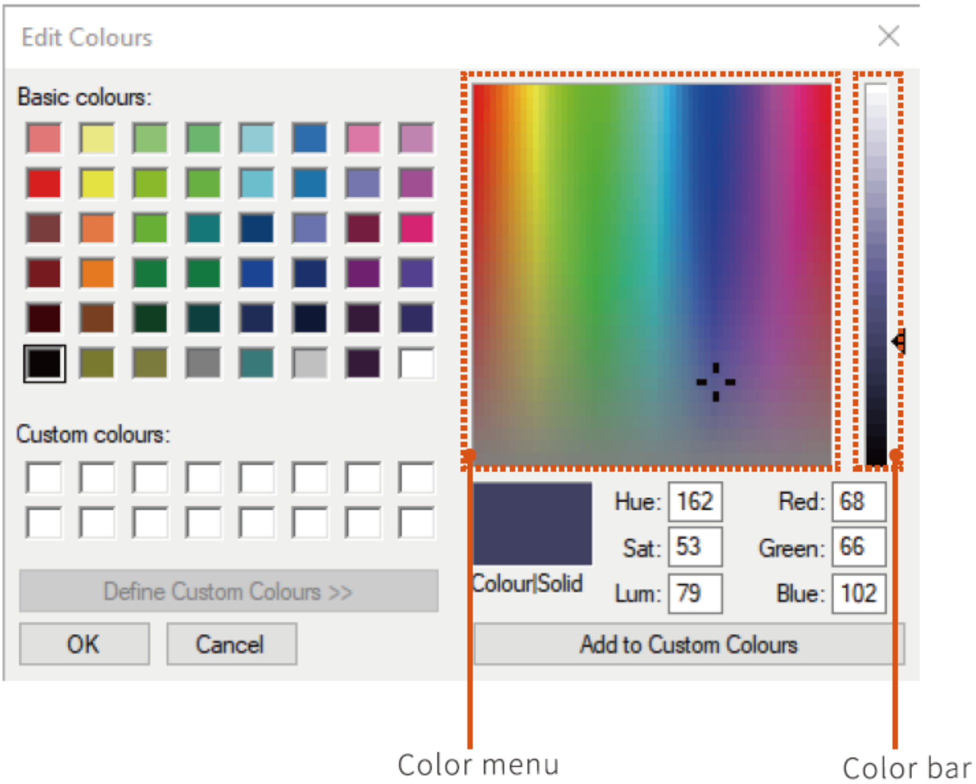


FIG. 2-13

Practice: Use RGB LEDs to display the seven colors of the rainbow respectively and fill in the corresponding RGB values as shown in FIG. 2-14.

Color name	Value for red	Value for green	Value for blue
Red	255	0	0
Orange	255	150	0
Yellow	255	255	0
Green	0	255	0
Cyan	0	150	150
Blue	0	0	255
Purple	255	0	255

FIG. 2-14

Note:

a) This section allows students to generate RGB values and write programs and display seven colors on the robot through the use of drawing software, and the reference program is as shown in FIG. 2-15.

FIG. 2-15

b) The values in the table are just for references only, and similar effects will be fine.

c) The cyan looks like blue, but it is actually the color between green and blue, which is hard to tell by the naked eye, and is commonly known as "Cyan was extracted from blue but darker than blue".

Ⅲ Programming and optimization (30min)

1. Master two common structures in the process of exploration: sequence structure and loop structure.

Practice: How to write a program in order to make the robot show the seven colors of the rainbow in sequence?

Conclusion: You only need to stack the blocks of debugged each color, and the reference program is as shown in FIG. 2-16.

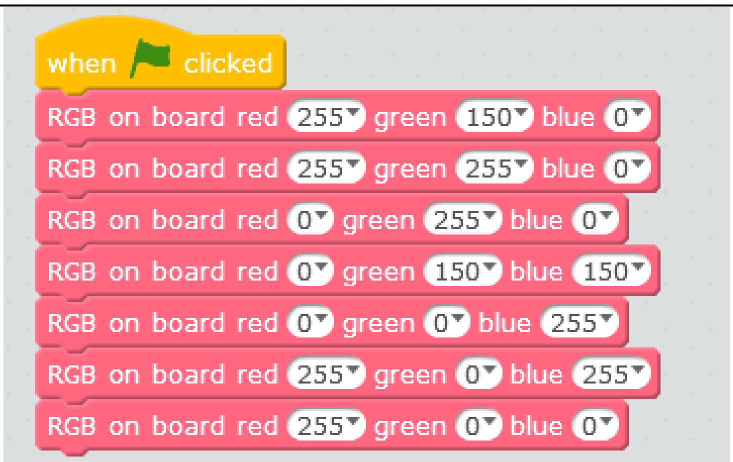


FIG. 2-16

Exploration: In the process of running the above program, we found that the robot can only display rainbow seven color light once. How to write a program if we need the RGB LEDs to keep transforming seven rainbow colors?
Conclusion: In the process of programming, the commonly used program structures include sequence structure and loop structure.

Sequence-structure: The program design of sequence structure is the simplest---to write the statement according to the sequence of solving a problem. The execution sequence is from top to bottom in turn, which is commonly used as the solution of a single, evident problem.

Loop structure: Loop structure is the structure for repeating some statements, which can effectively reduce the writing magnitude of the program. This is a structure that can take full advantages of a computer, which is commonly used in cases that program requires repeating.

To solve the above problems, a loop structure is needed, as shown in FIG. 2-17.

FIG 2-17

2. Observation and discovery: keep improving to optimize program effect.

Exploration: Observe the effect of the program, we found that the color transformation of RGB LED is too fast to see clearly. How to solve this problem?

Conclusion: Select "wait '1' secs" block in the "Control" category. The role of the block is to make the program of sequence structure to run next block after it waits for 1 second in this block. You can also modify the values to change waiting time. Adding this block at the back of every RGB block can delay the color transformation. The optimization program is as shown in FIG. 2-18.

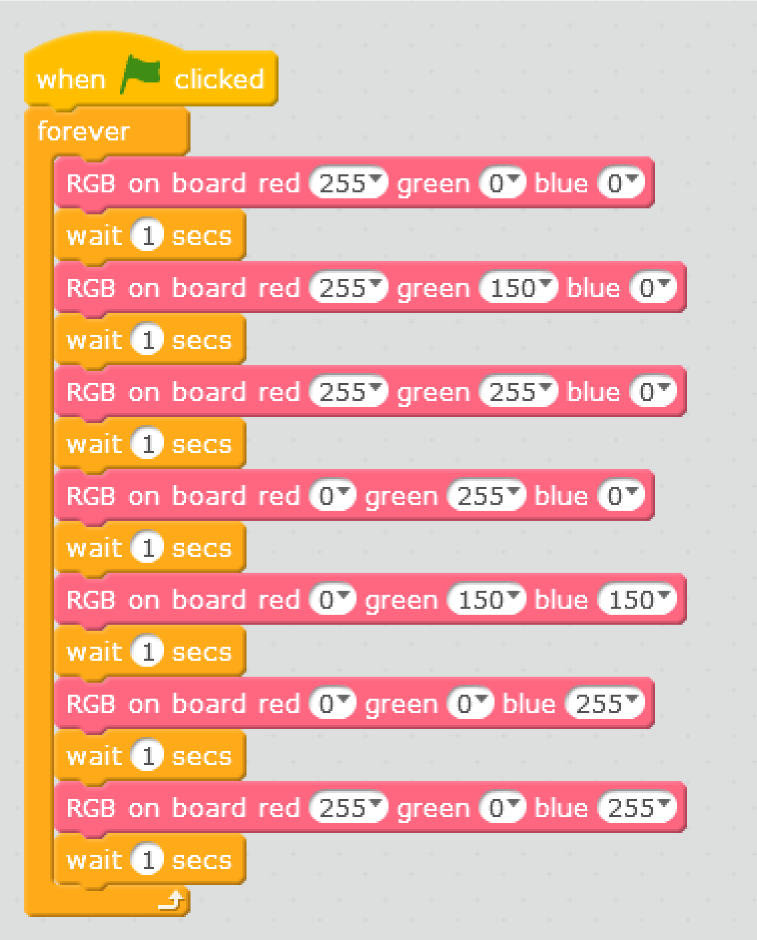


FIG. 2-18

Note: when writing the program, students are easy to ignore the last block--- "wait '1' secs". In the repeatedly executed program, machine will immediately execute the first block after executing the final block, resulting in the red color displayed immediately before the purple can be seen.

Ⅳ Classroom Development and Classroom Evaluation (10min)

1. **Classroom development: The colour modulation law of RGB LED is discovered by modifying RGB values (selected lecture according to class teaching pace).**

Exploration 1: Set the value of RGB LED to 50,50,50, increase the value of a single color, and observe the color change of RGB LED.
Conclusion: As the increase of some color values, finally the displayed color of RGB LED will get closer to that color.

Exploration 2: Set the values of RGB LED to the following groups of values and observe the experimental phenomena.

Serial NO.	Value for red	Value for green	Value for blue
1	50	50	50
2	100	100	100
3	60	30	0
4	120	60	0
5	0	60	30
6	0	120	60

Conclusion:
a. From group 1 and group 2 experiments, it can be concluded that white will be obtained when the three colors are mixed with the same value.
b. From group 3, 4, 5 and 6 experiments, it can be concluded that the color will not change if two of three kinds of colors are mixed in the same proportion. But the color will be brighter if the value of mixed color is larger.
Note: Teachers may choose exploration experiment according to the class teaching pace. The time of exploration one shall be shorter.

2. Classroom evaluation: students' class performance will be evaluated according to the evaluation criteria.

Evaluation criteria:
a. Accurately debug RGB values of seven colors.
b. Able to control the loop transformation of the seven colors.
c. The color transformation of 7 colors is noticeable (able to use delayed blocks).

Chapter 3 Little Performers

Summary

After students visually experienced the colorful transformation of RGB LED in lesson 2, lesson 3 will help students aurally experience the beautiful songs of robots.

Learning Objectives

To understand the working principle of the buzzer and be able to make the buzzer make a different sound skillfully.
To understand music knowledge and be able to write programs for robots to play songs.

Course Time

90 minutes.

Pre-class Preparation

WeeeBot robots, computers with WeeeCode software installed, and USB download cables.

Teaching Process

I Introduction from life (15min)

1. Understand the principle of sound generation by the way of associating the playing method of common musical instrument.

Instrument playing: It will need a certain way to make a sound. For instance, it will need to pluck the string for a guitar, press the key for a piano, and blow in the air for a flute.
Question: Taking guitar for example, what will happen if you put your hand on a string that is playing?
Conclusion: The sounding object vibrates. When the object does not vibrate, the sound will disappear. Musical instrument all make a sound by causing the vibration of objects.

2. **Deepen the understanding of the principle of sound generation through the experience of human voice.**

Exploration:
a. Place your hands lightly on your throat.
b. Open your mouth and continue to whisper "ah" and feel the change of your throat.
c. Continue to yell out "ah" loudly and feel the difference between it and whispered "ah" vibration.
Phenomenon: When phonating, your hand feels "numbed", and actually it is the throat in vibration. When the sound is large, the vibration of the throat will be more frequent and larger.

Conclusion: Sound is a sound wave generated by surrounding air vibration, which is caused by the vibration of an object, thus the sound differs depending on the vibration frequency.

Ⅱ Know about Module and Use It(15min)

1. Master the usage of the buzzer and understand its working principle.

The sound principle of the buzzer is the same as the generation principle of sound, both through vibration. We can control the vibration frequency of the buzzer to control it to make different sounds. The buzzer is as shown in FIG. 3-1.

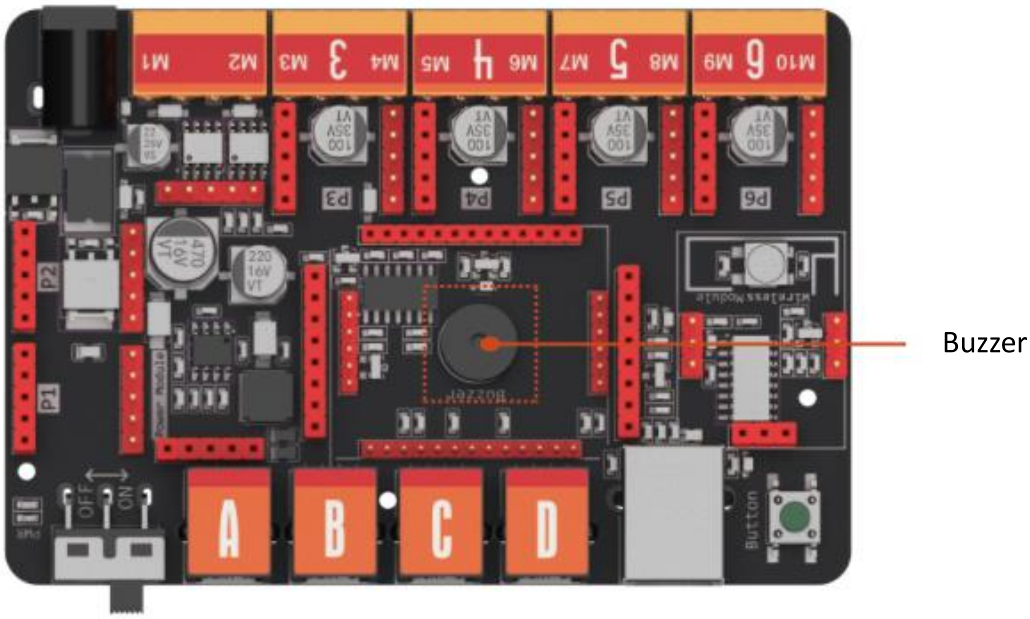


FIG. 3-1

2. Control the buzzer's sound programmatically and explore the role of blocks through experiments.

We can find the control block of the buzzer in the “Robot” category, as shown in FIG. 3-2, and then we can learn the block through experiment exploration.

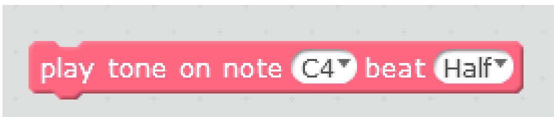


FIG. 3-2

Exploration:

a. Set the robot to online mode, run the program as shown in FIG. 3-3 and listen to the sound.

FIG. 3-3

When the green flag is clicked, the play tone is C4 and beat is 1/2.
Program effect: The sound is very similar to the "do" sound in music.

b. Modify the first parameter C4 in the blocks into D4 and listen to the sound.
Program effect: The sound is very similar to the "re" sound in music.

c. Modify the second parameter half beat in the blocks into double beat and listen to the sound.

Program effect: The playing sound lasts longer.
Conclusion: The first parameter of the buzzer block can control the tone of the sound, and the second parameter can control the duration of the sound.

Ⅲ Know about relevant music knowledge and master the methods of playing songs by programming. (50 min)

1. Know about relevant knowledge of musical tone and scale, and understand the meaning of "C4" in the blocks.

The composition of music has two very key elements -- the tone and beat, the tone determines the kind of the sound, and the beat determines how long the sound will last.

We all know that the common seven music sound (7) is "do", "re", "mi", "fa", "so", "la," "si". Through the experiment just now, we have known that the C, D, E, F, G, A and B in the blocks correspond to the seven sounds. What does "4" in the C4 mean? We know that there are three common musical scales---BASS, MIDDLE, and TREBLE. As "4" represents MIDDLE, therefore, we can know that 3 represents BASS, 5 represents TREBLE and C4 is MIDDLE "do." In the musical expression of numbered musical notation, BASS, MIDDLE, or TREBLE is enough to allow us to play much music.

2. know about a beat and the corresponding delay time and understand the differences between the beat number of numbered musical notation and that of blocks.

To understand the beat, we will need to learn how to read the score. Here we take the numbered musical notation of the song ---"Ode to Joy" for example, to understand the delay time expressed by the beat, as shown in FIG. 3-4.

Ode to Joy

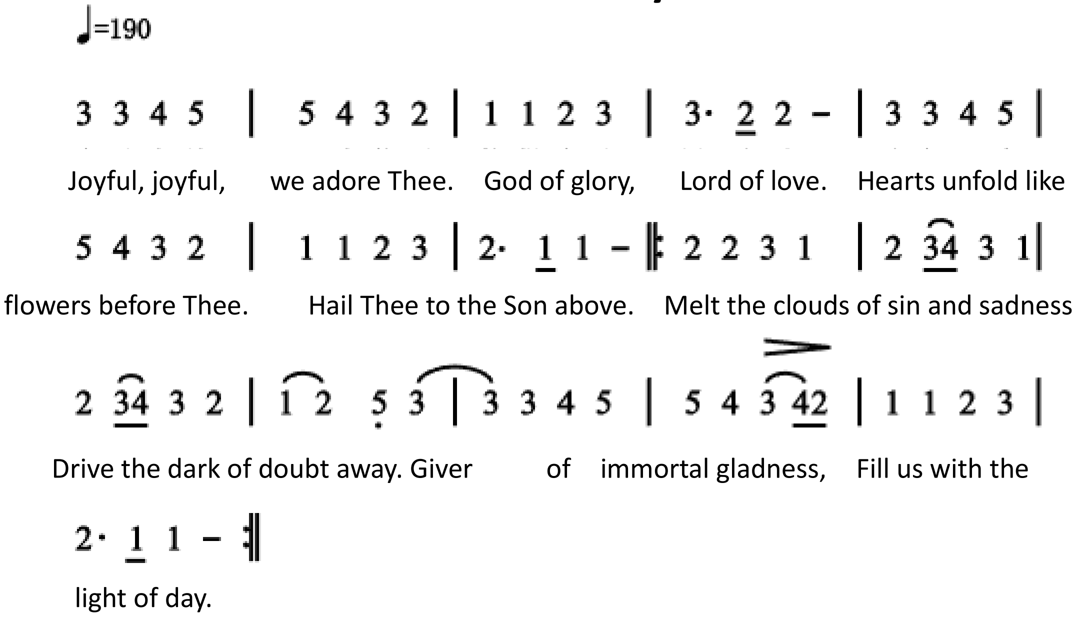


FIG 3-4

The 4/4 beat in the upper-left corner of the numbered musical notation indicates that the music is in four-quarter time, which means that there are 4 beats between each vertical bar. As there are 4 notes in each vertical bar, each ordinary note is 1 beat.

Note: The following sections describe how to calculate the delay time of each beat. If the student has not learned the two-digit division method, the conclusion can be drawn directly.

The delay time of a beat in each song is different. We can see the number”190” in the top left corner of the music score. The number means that you need to play 190 beats in 1 minute. So, we could know that the duration of a beat is “60/190=0.32s=320ms”. However, the delay time of beats in music is different from that of beats in blocks, as shown in FIG. 3-5.

Beats of blocks	Delay time
Eighth beat	125ms
Quarter beat	250ms
Half beat	500ms
Whole beat	1000ms
Double beat	2000ms

FIG. 3-5

Conclusion: The corresponding beat number of blocks can be selected according to FIG. 3-6.

Beat in music score	Beat value in WeeeCode
30	Double beat（2000ms）

60	Whole beat（1000ms）
120	Half beat（500ms）
240	Quarter beat（250ms）
480	eighth beat（125ms）

FIG. 3-6

Taking this song as an example, in the blocks, we can use a half or a quarter beat to represent a beat in the score. In the following programming, we will take a half beat as an example to write the program.

Note:
a. If the requirement in beat number needs to be more accurate, we can use the following method to represent a beat of the music; namely, the value of the beat parameter is as much as the time delay in ms, as shown in FIG. 3-7.
b. If students have difficulty in understanding the content of the beat, we can adjust the length of the beat through the performance of the song.

FIG. 3-7

3. Write program according to the numbered musical notation to make the buzzer to play the song.

Then we can write the program according to the numbered musical notation. At first, we analyze the music in the first four sections of the numbered musical notation, and there are four different notes in the four sections of music:
a. Common notes: The first pitch 3 is a common note, takes a standard beat, in this song a standard beat is a half beat.
b. Underlined notes represent 0.5 beats, which means it is a quarter beat in the music.
c. Some notes are followed by a dot, indicating adding extra 0.5 beats, i.e., one note is 1.5 beats (1+0.5=1.5 beats), which is represented as 1*750 beat in music.
d. Some notes are followed by "1", indicating adding extra one beat, that is, one note is 2 beats (1+1=2 beats), which is represented as the whole beat in music.

The relationship between the note type and the block beat number in this song is as shown in FIG.3-8.

Note type	Block beat number
Common notes	half beat
Underlined notes	quarter beat
Note with dot	1*750 beat
Note with number "1"	Whole beat

FIG. 3-8

Therefore, we can know that the music program of the first four sections is as shown in FIG. 3-9.

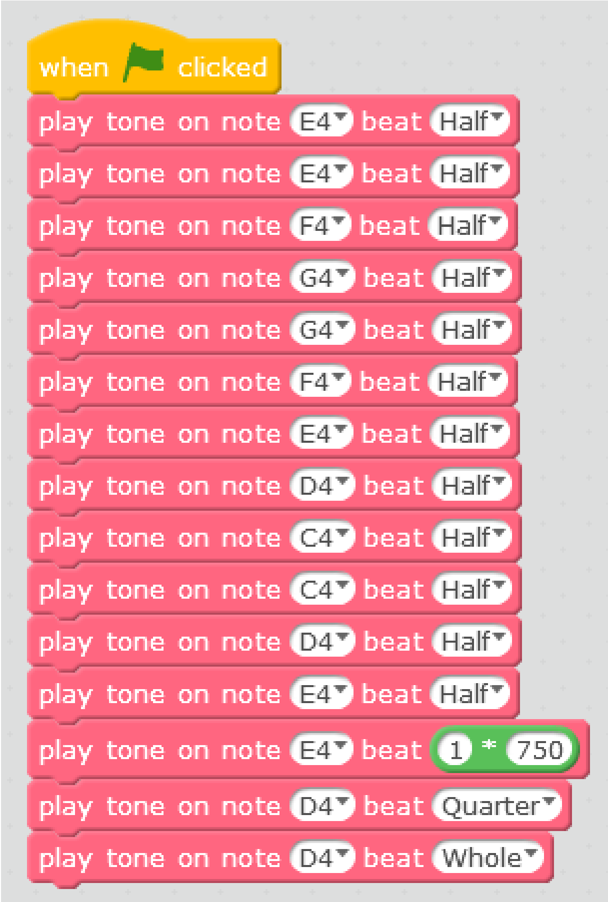


FIG. 3-9

Note:

a. The programming method of music in the latter part is the same as music in the first four sections so that students can write it on their own according to the numbered musical notation.

b. It is easy for students to make mistakes in tone or beat due to carelessness in the programming process. Let students find out and correct the problem through performance, to cultivate their ability to discover and solve problems.

Ⅲ Classroom Development and Classroom Evaluation (10min)

1. Classroom development: To achieve the effect that “tone change, color change” through the combination of color transformation and tone transformation of RGB LED. (selected lecture according to the class teaching pace).

Exploration:

We can find that each musical scale has seven tones, and seven color light transformation of RGB LED also have seven kinds of color. Can me realized the transformation of RGB LED when the buzzer plays “do”, “re”, “mi”, “fa”, “so”, “la” and “si”?

Conclusion: The corresponding color of each tone is as shown in FIG. 3-10, and the program is as shown in FIG. 3-11.

Tone	Color
Do	Red
Re	Orange
Mi	Yellow
Fa	Green
So	Cyan
La	Blue
Si	Purple

FIG. 3-10

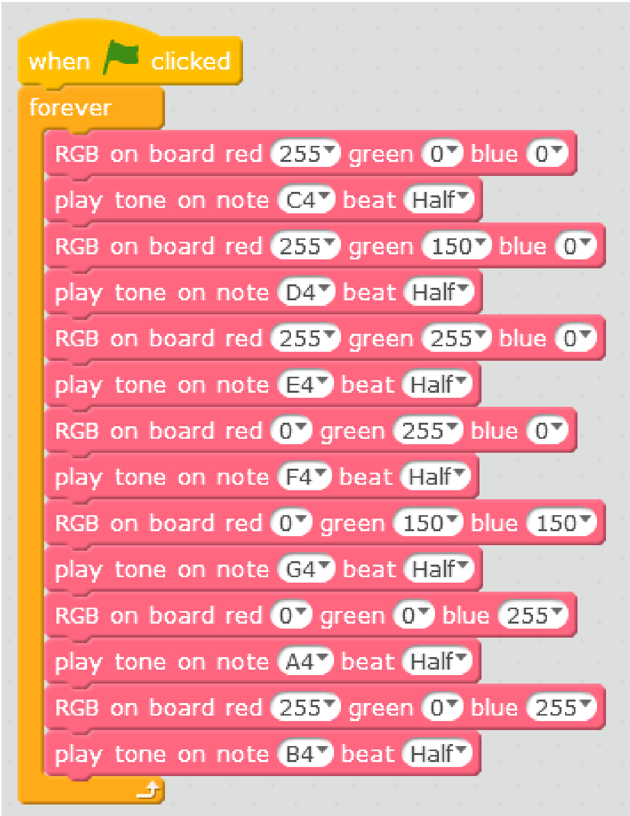


FIG. 3-11

Q: why does the color transformation of RGB LEDs not need to add "wait for 1-second" blocks?

Conclusion: The beat parameters of the buzzer blocks play the role of keeping the sound for a period, which is equivalent to the time delay. As long as there is time delay during the color transformation, each color can be seen clearly.

2. Classroom evaluation: Students' class performance will be evaluated according to the evaluation criteria.

Evaluation criteria:

1. Able to control the sounding of the buzzer.
2. Able to play music without mistakes.

Chapter 4 Robot, run!

Summary

This lesson will let students dynamically experience the fun of programming by learning the control method of the motor to make robots to move in all directions and walk along the preset route.

Learning Objectives

To master the control method of motor and be able to control the robot's movement skillfully.

To master the operation method of off-line robot control.

Able to control the robot to walk along the preset route.

Course Time

90 minutes.

Pre-class Preparation

WeeeBot robots, computers with WeeeCode software installed, and USB download cables.

Teaching Process

I Introduction from Life (15min)

1. Understand the principles of automobile movement in life and understand the power sources of different types of automobiles.

Automobiles can move because the engine provides the power to them. The drive modes of automobiles include three kinds of drives ---forward -wheel drive, rear-wheel drive and four-wheel drive. Forward -wheel drive represents that its drive power source comes from the front two wheels. In the same way, it can be known that the drive power source of rear-wheel drive comes from the rear two wheels and that the drive power source of four-wheel drive comes from four wheels, as shown in FIG. 4-1.

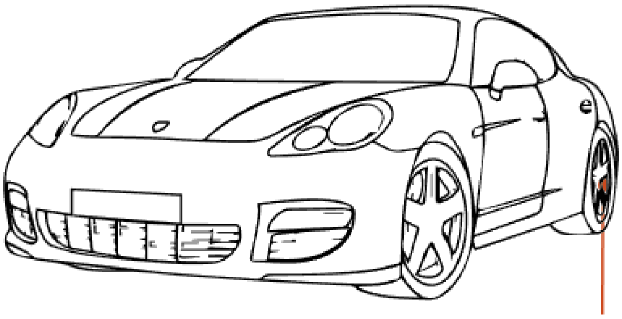


FIG. 4-1

Ⅱ Know about Module and Use it.

1. Master the usage of motor.

We can find the control blocks of the motor in the “Robot” category, as shown in FIG. 4-2 and FIG. 4-3.

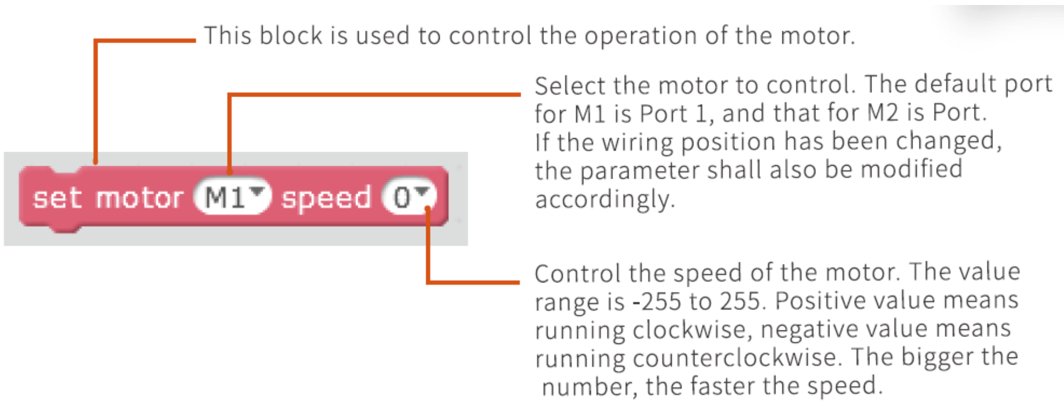


FIG. 4-2

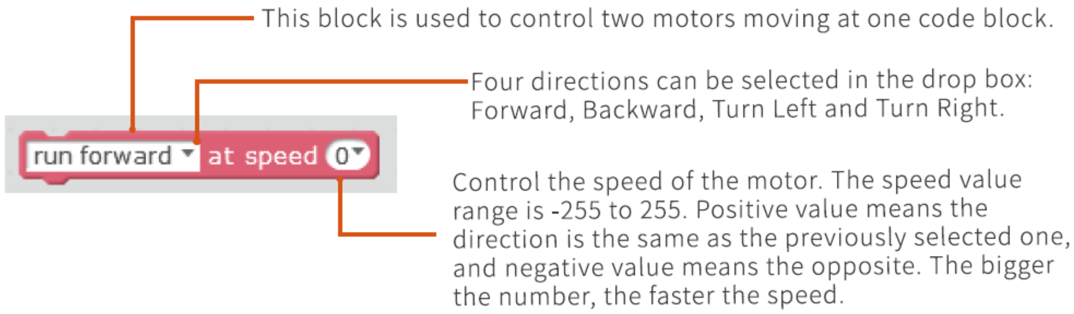


FIG. 4-3

Note:

a. The name setting of the motor is as shown in FIG. 4-4. (Right: M1. Left: M2)

FIG. 4-4

b. Our eyes must be directly facing the motor to tell the running direction of the motor, as shown in FIG. 4-5. For example, to move forward, the right wheel of robot needs to rotate clockwise, and the left wheels need to rotate counterclockwise, as shown in FIG. 4-6.

FIG. 4-5

FIG. 4-6

Practice: Control the robot to move forward in two different ways, and the program is as shown in FIG. 4-7.

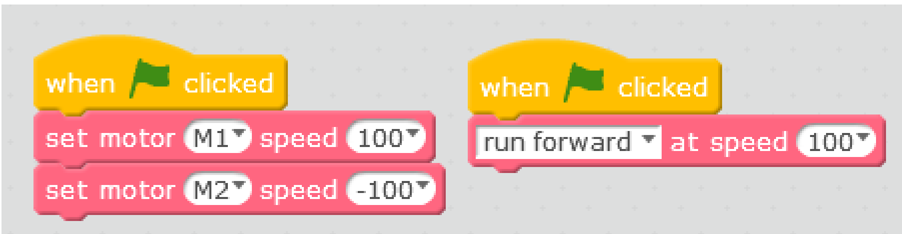


FIG. 4-7

Note:

a. After programming, if it is found that the rotation direction of the wheel is opposite to the original rotation direction, please check two places:

a) Motor wiring position: The right motor shall be connected with M1 port, and the left motor shall be connected to M2 port.

b) Motor wire: The two wires of the motor are one black and one white, and please ensure that the left is black, and the right is white, as shown in FIG. 4-8.

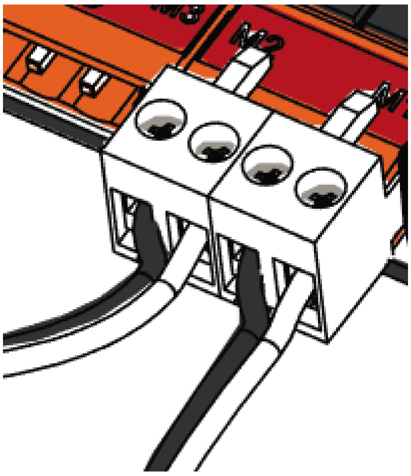


FIG 4-8

b. In the first two lessons, it would be Ok not to turn on the power switch for online control. However, if the power switch is not turned on in the online control of the motor, the motor will not run.

c. In the control of the motor, if the motion state of the motor does not change, as long as the power supply is maintained, it will keep the original motion state.

Ⅲ Master the Offline Control Method of Robots

In the process of practice, we found that it was too troublesome to control the robot with the USB cable, and next, we will learn how to control the robot offline.

Practice:

1. Connect to the USB cable, and select the port of the mainboard connected to the USB cable in the "connect" button of the toolbar, as shown in FIG. 4-9.

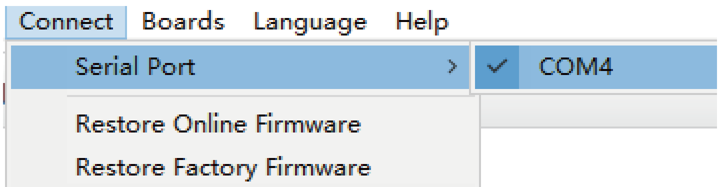


FIG. 4-9

2. Select WeeeBot software platform in the mainboard button, as shown in FIG. 4-10.

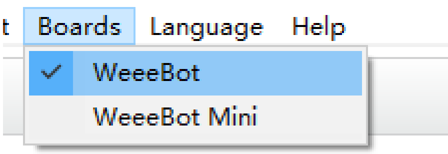


FIG. 4-10

3. Drag the motor control block from the block palette and also drag the "WeeeBot master program" block to the uppermost of the blocks, as shown in FIG. 4-11, which can generate the offline code from the blocks.

FIG. 4-11

4. Click the "Arduino mode" option in the "edit" button of the toolbar or click the blocks, you can see the generated offline code on the right side of the software port, as shown in FIG. 4-12.

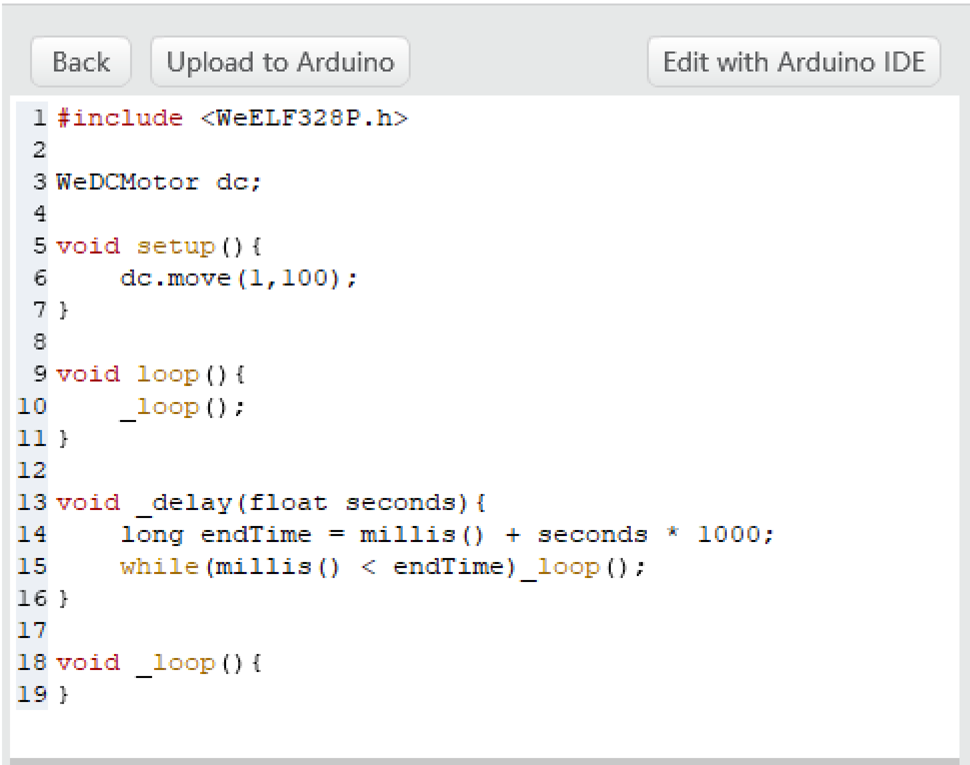


FIG. 4-12

5. Click "Upload to the Arduino" button. The written program has been successfully uploaded to the mainboard if the upload finish interface appears, as shown in FIG. 4-13.

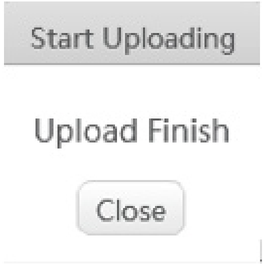


FIG. 4-13

6. Unplug the USB cable and turn on the power switch of the robot, the robot will start to run the offline program.

IV Programming and optimization (40min)

1. Control programmatically and debug the robot to make it walk along a square route.

Exploration: How to write the program if you want to make the robot to walk along the square route?

Conclusion: The movements of robots shall be as follows: move forward→turn left →move forward→turn left→move forward →turn left →move forward→ turn left (reverse will also be OK). The walking route is as shown in FIG. 4-14 and reference program is as shown in FIG. 4-15.

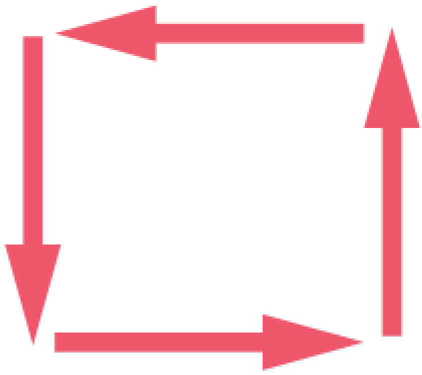


FIG. 4-14

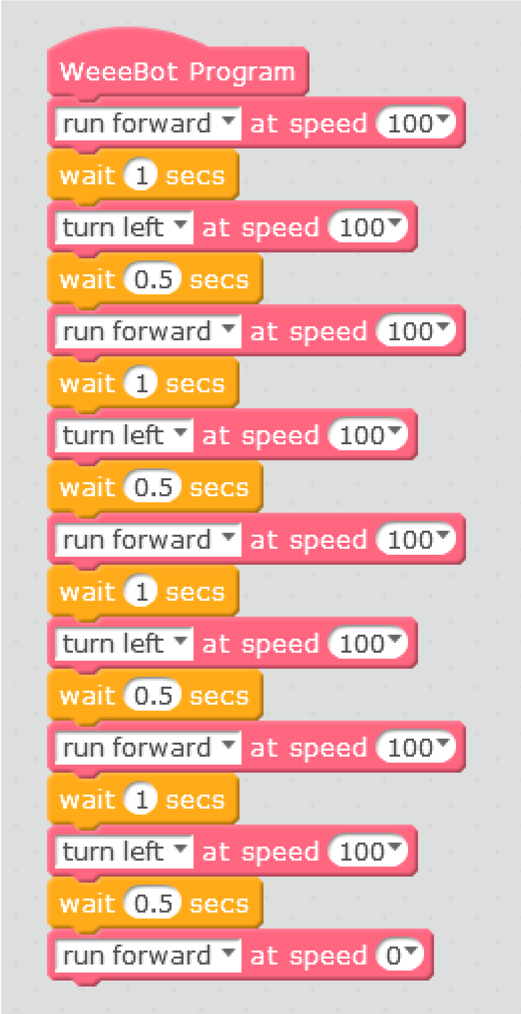


FIG. 4-15

Note:

a. The delay block at the bottom of "forward" block is to control the robot's forward distance, and the delay block at the bottom of the "turn left" block is to control the robot turning angle. When programming, the delay time must be changed according to the actual walking situation of the robot.

b. At the end of the programming, you must stop the running of the motor. Otherwise, the robot will keep the original motion and will always turn left.

2. Observation and discovery: find the operation law of the program and optimize the program.

Q: can the program where the robot walks along the square route be optimized?

Conclusion: robot walking along the square route is controlling wheels to repeat “Move forward and turn left 90 ° ” four times, and the program can be modified as shown in FIG. 4-16.

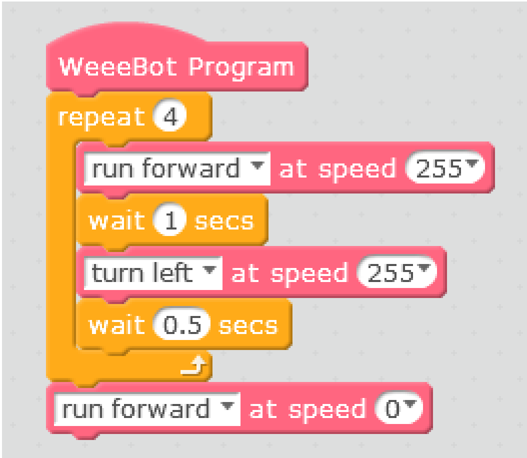


FIG. 4-16

V Classroom Development and Course Evaluation (10min)

1. Classroom development: set the route programming to let the robot walk and train the students' debugging ability (selected lecture according to the class teaching pace).

Teachers arrange simple trajectory according to the class teaching pace and let students write programs according to the requirements, and modify the relevant parameters, as far as possible to coincide with the required trajectory. The reference trajectory is as shown in FIG. 4-17 and reference program is as shown in FIG. 4-18.

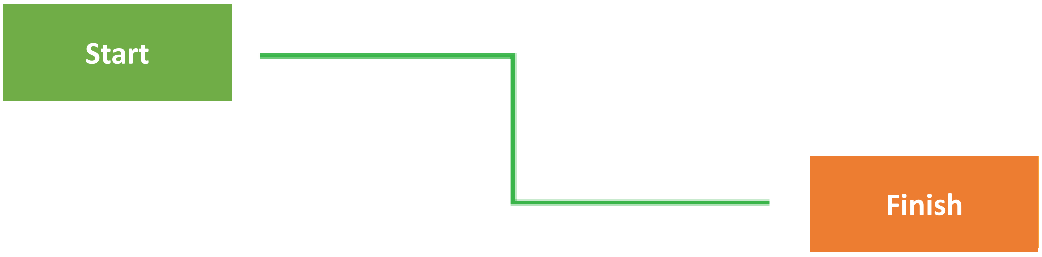


FIG. 4-17

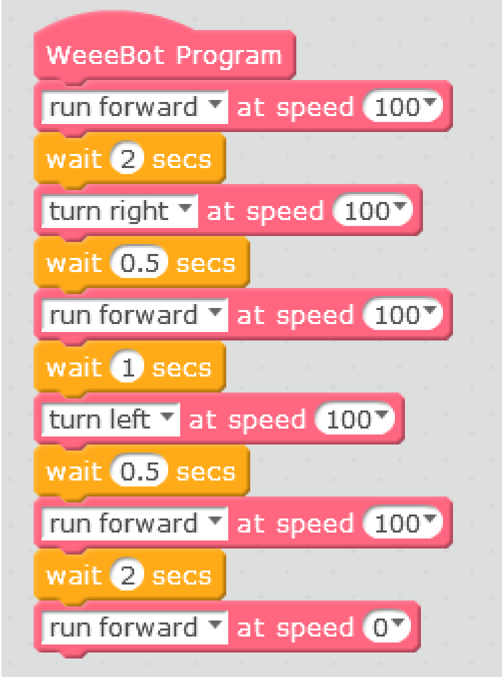


FIG. 4-18

2. Classroom evaluation: Students' class performance will be evaluated according to the evaluation criteria.

Evaluation criteria:

1. Able to write a program to control motor to make the robot move.
2. Able to master the method of offline control.
3. Be able to debug the route of the robot walking in a square shape.

Chapter 5 Obedient Robots

Summary

After students are able to control the robot to walk in lesson 4 programmatically, now we will get started IR remote control module, learn to build a remote-controlled robot accordingly, and cultivate the fun of learning robots.

Learning Objectives

Understand the principle of IR remote control, master the control method of IR remote control module, and be able to write the program of the IR remote-controlled robot.
Understand the flowchart and understand the use of flowchart to clarify the programming ideas.
Know the third structure that is commonly used in programming --- select structure.

Course Time

90 minutes.

WeeeBot robots, computers with WeeeCode software installed, USB download cables and remote controls.

Teaching Process

I Introduction from Life (15min)

1. Associate the application of remote control in life and understand the characteristics of the infrared ray.

Application of remote control in life: TV, air conditioner, remote control toy, and so on. Remote controls usually use an infrared ray to transmit signal, and we can know the two characteristics of infrared ray from the examples in our lives:

1. Invisible
2. Strong penetration and not easy to be disturbed.

2. Understand the Application of Infrared Ray in Life

Infrared ray not only has the two characteristics mentioned above, but also has another characteristic -- the transmission of heat, and due to this characteristic, it makes the application of infrared ray in life very wide.

a. Thermal imager: It uses the received the heat difference of infrared transmission to form an image, and transforms the invisible infrared heat is into a visible thermal image.

b. Bathroom heat lamp: It raises the temperature in the light area through the thermal radiation of the infrared bulb.

c. IR guided missile: infrared guidance is to use the infrared radiation of the target itself to guide the missile to approach the target automatically, to improve the hit rate.

Ⅱ Know about Module and Use It (15min)

1. Know about IR remote control module.

IR remote control module is divided into two parts: infrared emission module and infrared receiver module.

Infrared emission module is mainly used to transmit the infrared signal, namely remote control, and the infrared signal from different buttons is different.

The infrared receiver module is mainly used to receive the infrared signal, as shown in FIG. 5-1.

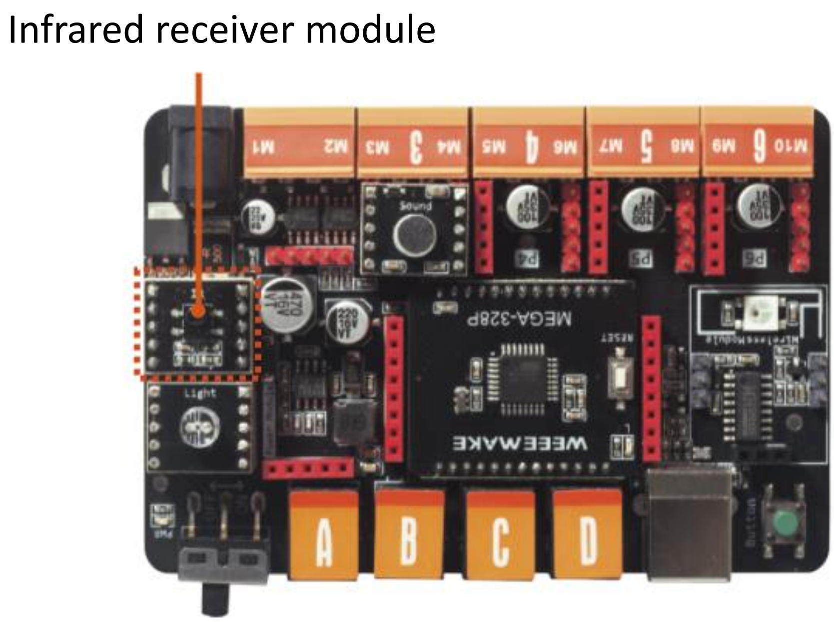


FIG. 5-1

2. Master the usage of IR remote control module.

We can find the control blocks of the IR remote control module in the “Robot” category, as shown in FIG.5-2.

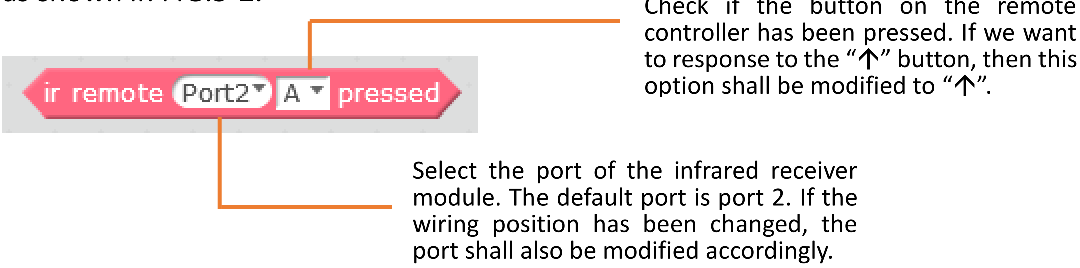


FIG.5-2

Practice:

Select the "M-Elephant" role and write the program as shown in FIG. 5-3 in online mode and run it, then you can use the "→" key of the remote control to control the role to move to the right.

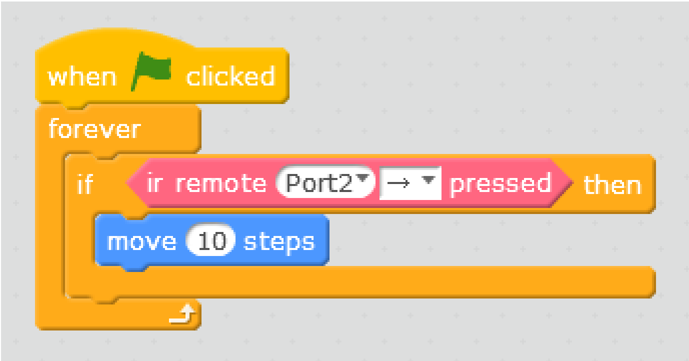


FIG. 5-3

Ⅲ Programming and Optimization (50min)

1. Programming of Remote Controlled Robots

Q: If we are going to make a remote-controlled robot: When different directions of the key are pressed on the remote control, the robot can perform the corresponding action. How to write the program?

Conclusion: Judge by each key pressed and select walking direction accordingly. The program is as shown in FIG. 5-4.

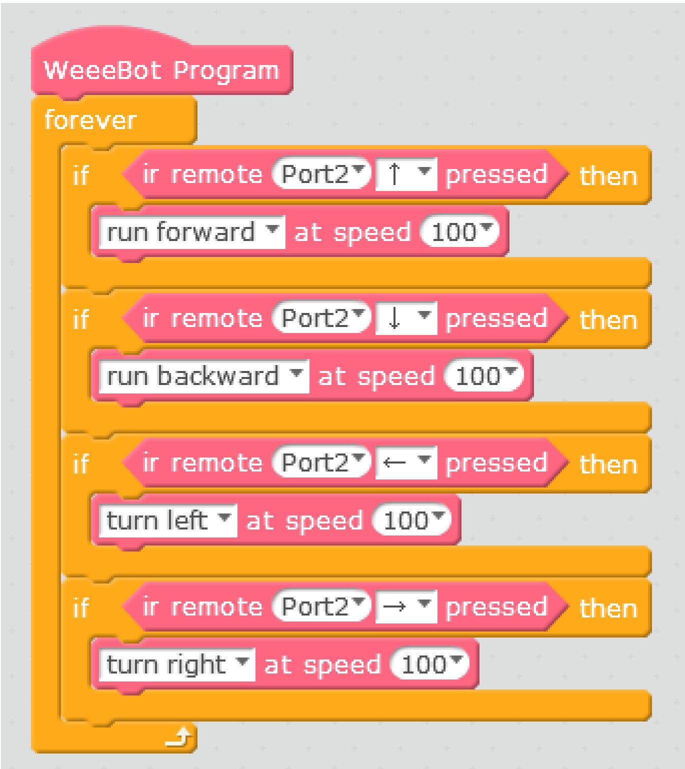


FIG. 5-4

Note:

a. If students can find problems with this programming idea before they write, they can go directly to the next step without writing the program.

b. If the robot does not respond after you press the remote control, please detect two places:
a) Check whether the infrared receiver module is connected to the mainboard port 2.
b) Check whether the infrared emission module can transmit the signal smoothly: press the button of the remote control, turn on the camera function of the phone, and watch the transmitter of remote control through the camera. Press the button, and if the remote control can transmit infrared signals, you can see there is purple-red light in the emission mouth. (a photosensitive device in a camera can receive a broader spectral range than the human eye, and so it can take the photos of the infrared ray.)

c. Because infrared ray is propagated along straight lines, the remote control shall be faced to the robot directly as much as possible in the remote control process.

d. The transmitting distance of the remote control signal is about 10 meters, and try not to remote control under the intense sun because too strong sunlight will cause interference to the infrared remote signal.

e. If there is no preset paring ID between the robot and remote control, the situation that one remote control can control several robots simultaneously will appear. Please pay attention to avoiding the situation of signal interference of remote controls with each other when letting students practice.

Program effect: We can find that there are problems with the program written in this way when running the program: When we do not press the button, the robot will keep the last motion state, and won't stop moving.

2. Know about the third structure commonly used in programming -- select structure.

Judgment structure is not enough to make robot choose the direction of walking intelligently. We need to use the third structure commonly used in program design -- select structure. The select structure represents that there is a branch of the program processing step, which requires the selection of one of the branches to execute in a particular condition. Next, we will know about select structure through an example in our lives, as shown in figure 5-5, and determine whether we need to go to school or not according to the judgment whether it is Sunday or not today through the branch structure.

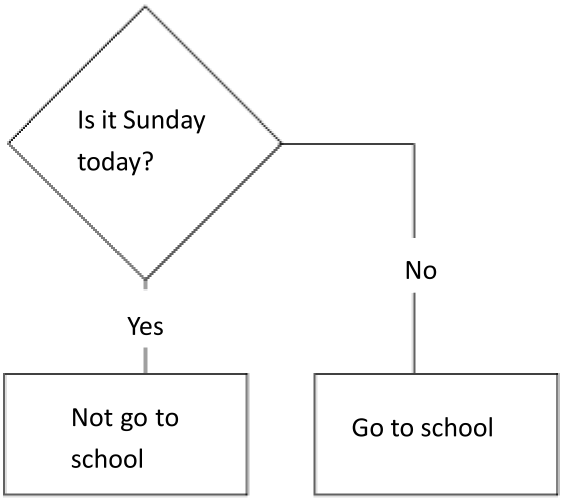


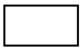
FIG. 5-5

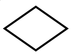
3. Understand the drawing method of the program flowchart and draw the program flow chart of the remote-controlled robot.

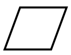
The chart above as shown in FIG. 5-5 is what we call program flowchart, which is a description of the method, idea or algorithm for solving problems. It uses a graphical symbol box to represent the operations of various properties and connects these operations with a flow line. Before the program is designed or problem is solved, it can help us clarify the solution of complex problems through the drawing of a flow chart.

The program flowchart has the following different symbols:

a. : The shape is the starting box, representing the beginning or end of the program. It will need to add this box at the beginning or end of the program.

b.  : This shape is the processing box, representing the processing method. Go to school or not go to school similar to FIG. 5-5 is the processing method of the judgment result.

c.  : This shape is the judgment box, representing the judgment and the branch, which is used to indicate the select structure, and it is often accompanied by Yes and NO two choices to judge the result.

d.  : This shape is an input and output box, representing inputting and outputting commands, which is generally used for variable assignment (This symbol may not be explained).

e.  : This shape is the flow line, representing the execution order of the flowchart, which is used for the connection between two boxes.

Note: The drawing of the flowchart is usually from top to bottom, and from left to right, and the flow line shall avoid crossing.

4. Write an optimized program of the remote-controlled robot according to the flowchart.

We can analyze the program idea according to the requirements: When pressing "↑", "↓", "←" and "→" key, "the robot has corresponding movement direction; When pressing other buttons or not pressing buttons, the robot will not perform any action, and the program requires loops. Therefore, we can make the program flowchart as shown in FIG. 5-6.

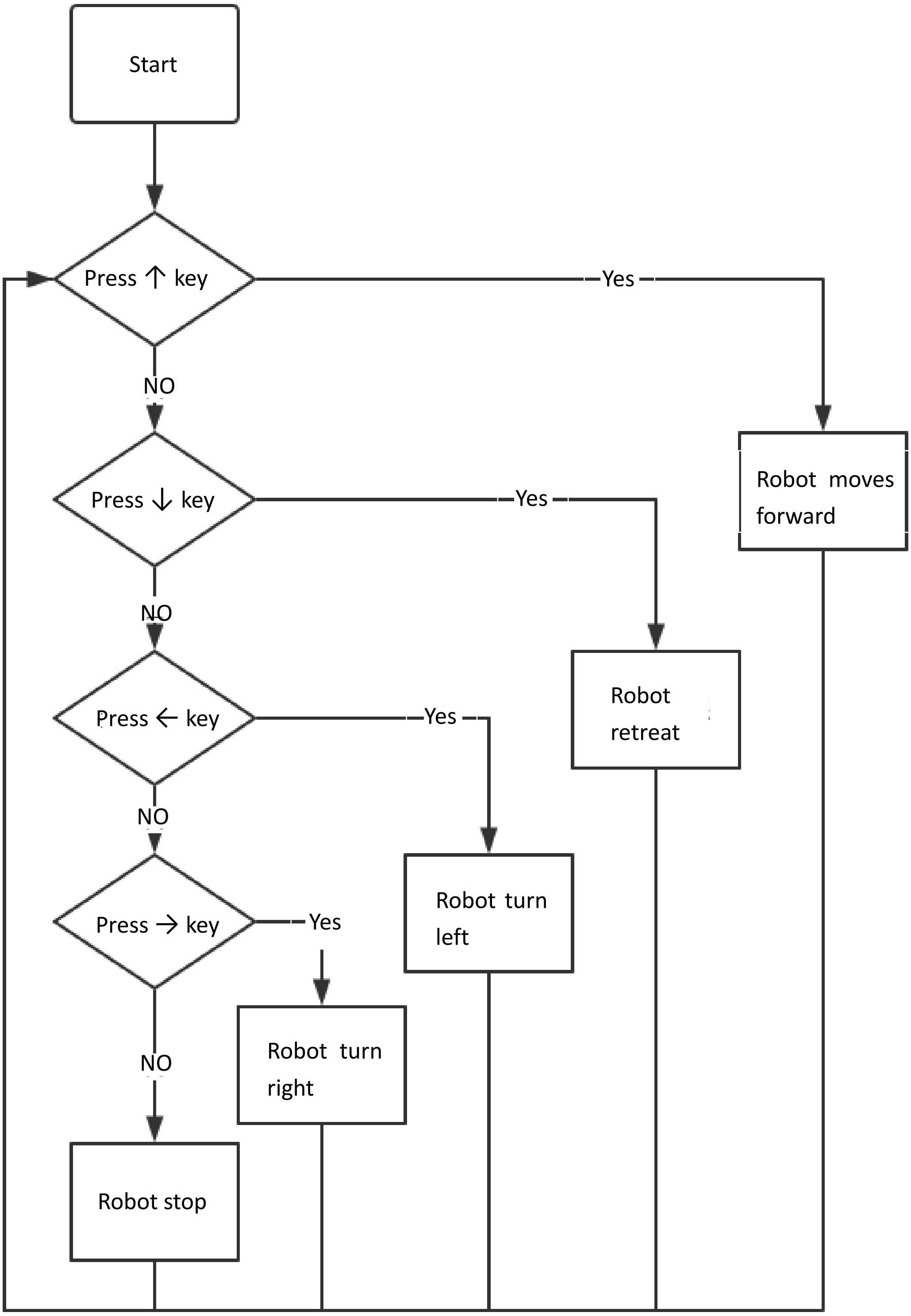


FIG. 5-6

We can write the program according to the flowchart as shown in FIG. 5-7.

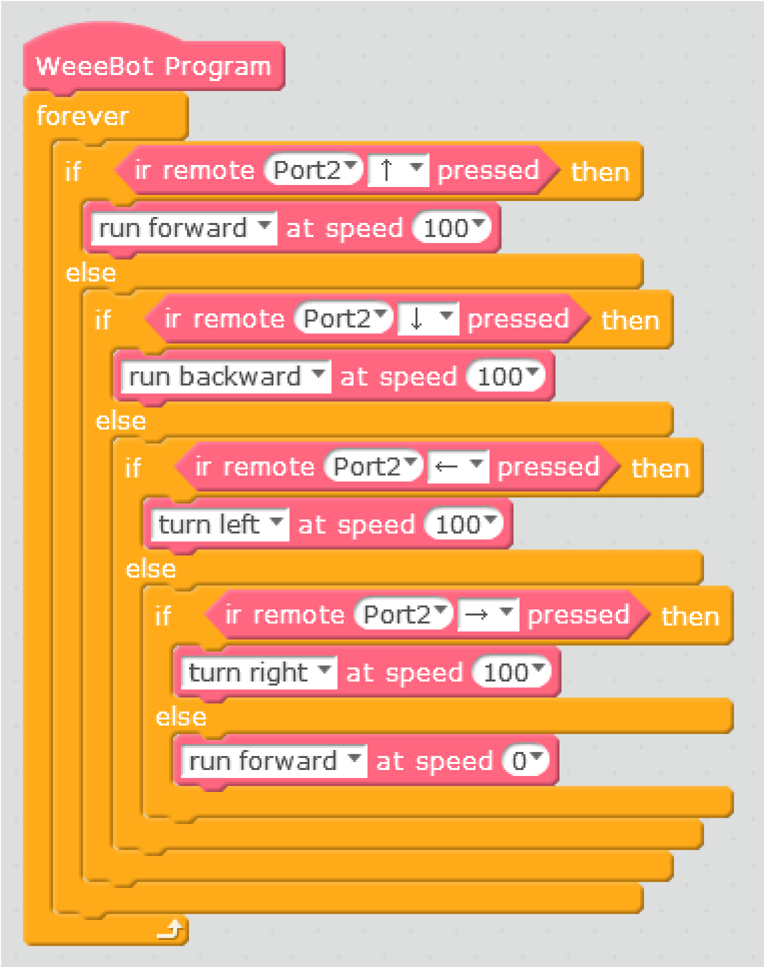


FIG. 5-7

Ⅳ Classroom Development and Classroom Evaluation (10min)

1. Classroom development: To achieve the color transformation function of the remote control RGB LED through the combination of the learned color transformation of RGB LED. (selected lecture according to the class teaching pace).

Write the program to allow the "1~7" button of the remote control to control the different color transformation of RGB LED and its reference program is as shown in FIG. 5-8.

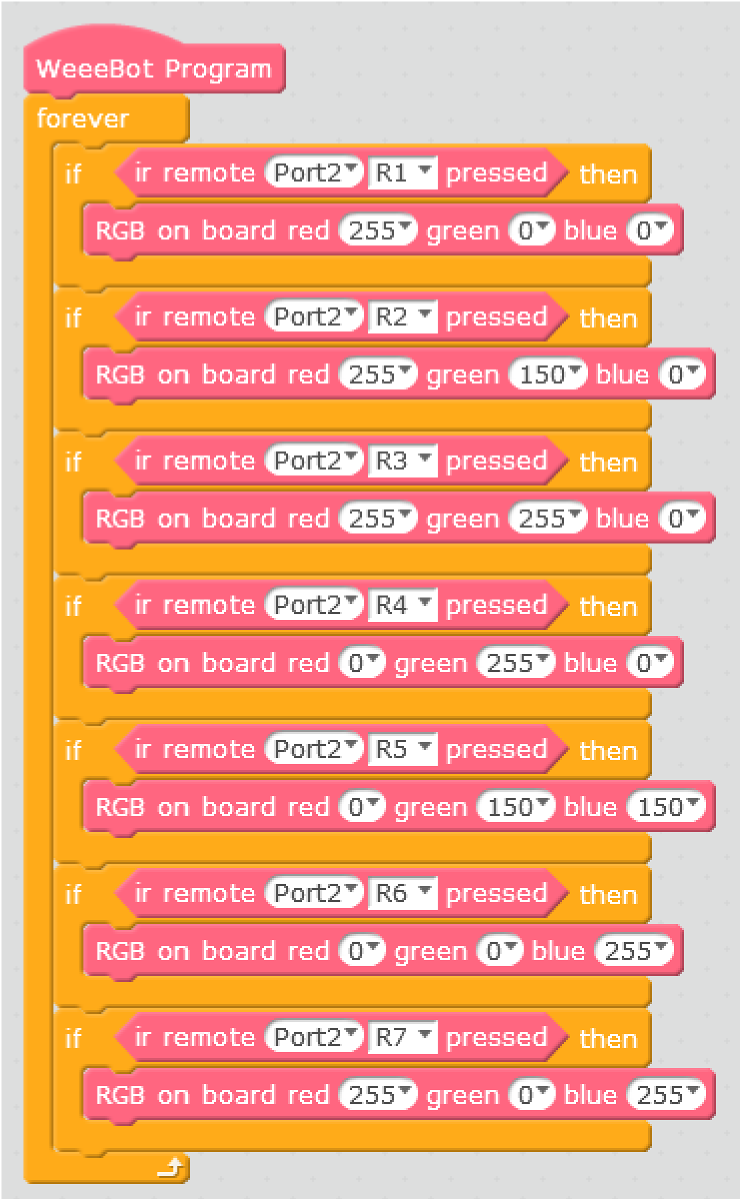


FIG. 5-8

2. Classroom evaluation: students' class performance will be evaluated according to the evaluation criteria.

Evaluation criteria:

a. Proficient in using the control blocks of the IR remote control.

b. Master the drawing method of the flowchart and draw the program flowchart of the remote-controlled robot.

c. Able to write the remote-controlled robot program according to program flowchart.

Chapter 6 Pin-wall Games

Summary

After the students mastered the usage of the IR remote control in the last lesson, we will use the IR remote control to make the pin-wall game through the combination of software and hardware.

Note: Lesson 6 and 8 are both game production courses combined by software and hardware and the course time is 180 minutes. If the student has a weak logical ability, please teach them lesson 6; If students have a strong logical ability, please teach them lesson 8. If it refers to the related knowledge points, such as coordinate system, variable, clone, and so on. It can be explained according to the content of this lesson.

Learning Objectives

Strengthen the understanding in IR remote control and make pin-wall games.

Master the knowledge points, such as clone, variable, coordinate system, repeat until the blocks, and so on. In the WeeeCode software and master its use method.

Course Time

180 minutes.

Pre-class Preparation

WeeeBot robots, computers with WeeeCode software installed, USB cables and remote controls.

Teaching Process

I Course Introduction (15min)

1. Let students know about the gameplay through game sharing, to provide ideas for developing games.

Teachers can let students share and summarize the functions of several game characters, as shown in figure 6-1, through showing the relevant pictures of the pin-wall game or giving the students the experience of playing the brick game.

Pinball	1. Bump against the baffle and wall.
	1. Hit the brick to score.
	2. The game will end if it drops under the baffle.
Baffle	1. Control the movement of the baffle to collide with the pinball.
Brick	2. It will disappear and get score if touching pinball.

FIG. 6-1

II Master the knowledge points related to games production (50min)

Note: this link is to learn the necessary knowledge for students to develop games, and teachers can also explain it in the process of developing games.

1. Understand the coordinate system, master the representation method of the position coordinates of the characters, and know how to determine the coordinate values according to the position.

The coordinate system represents the role position on the stage of the WeeeCode software. Click the "choose backdrop from library" button in the lower-left corner of the software and select the "xy—grid" at the bottom of the gallery, we can see that there is a coordinate system combined by x-axis and y-axis on the stage, as shown in FIG. 6-2. The intersection of the x and y-axes is called the origin, expressed as (x:0,y:0), that is, the x-axis is 0, and the y-axis is 0. The horizontal X-axis is 480 pixels, and its range is -240~240. The vertical Y-axis is 360 pixels, and its range is -180~180.

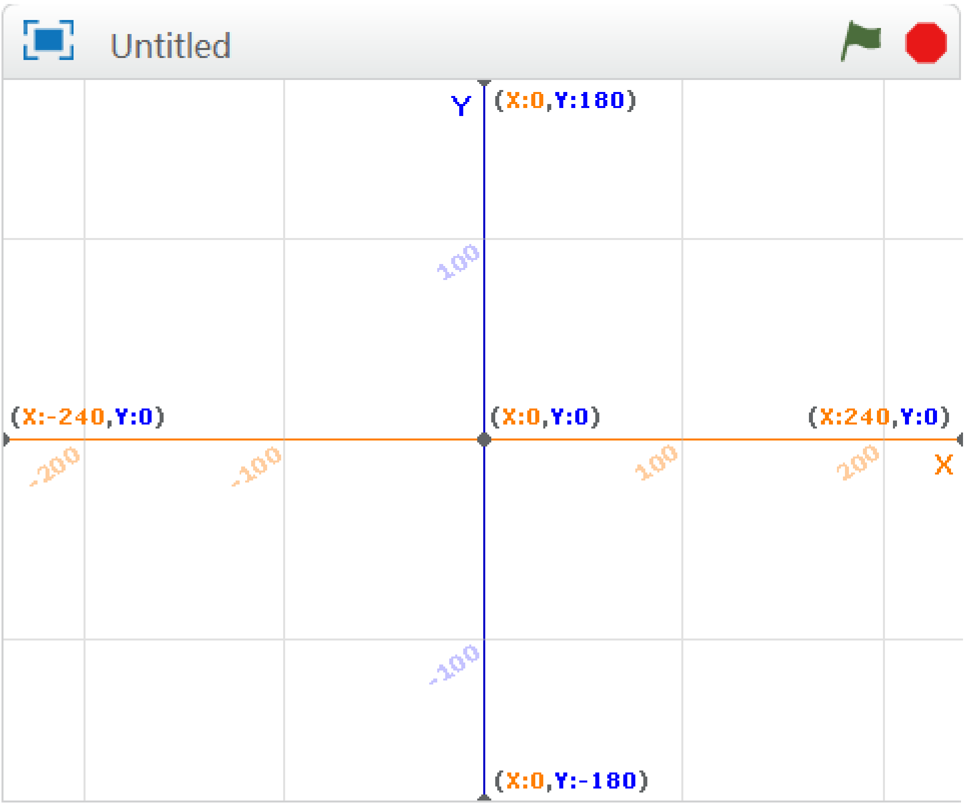


FIG. 6-2

Note: in the sprites pane, we can see the x and y-axiss of the mouse position currently, as shown in FIG. 6-3.

FIG. 6-3

2. Understand the meaning of "variable" and master the use method of "variable" blocks.

In the Data&Blocks, we can see the "Create a variable" option. Click this option, you can name the variable, as shown in FIG. 6-4. After the variable is named (The name can be given by yourself), you can see that there are five blocks in the block palette, as shown in FIG. 6-5.

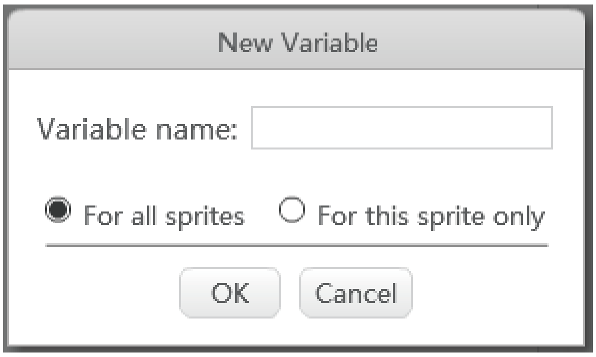


FIG. 6-4

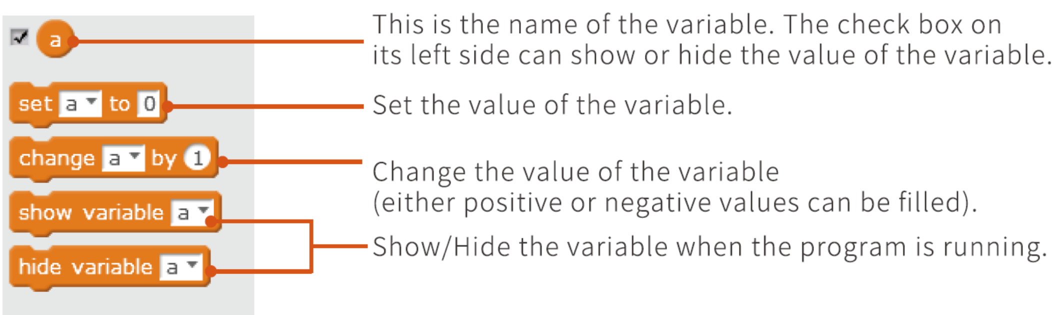


FIG. 6-5

Practice:

Write the program as shown in FIG. 6-6 and run it. Tick in the variable checkbox. On the stage, you can see that the value of variable "a" is increased by 1 every second, as shown in FIG. 6-7.

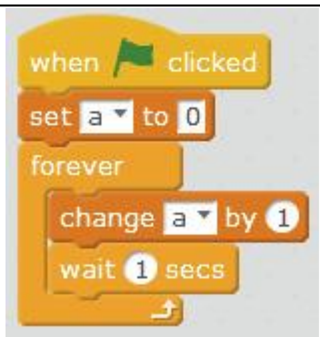


FIG. 6-6

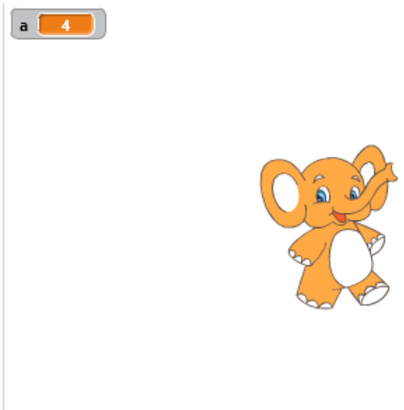


FIG. 6-7

3. Master the usage of "repeat until" blocks.

In previous studies, we have not been exposed to the loop structure with judgment. To represent the loop with judgment, we can use the "repeat until" block, which is in the “Control” category, as shown in FIG. 6-8.

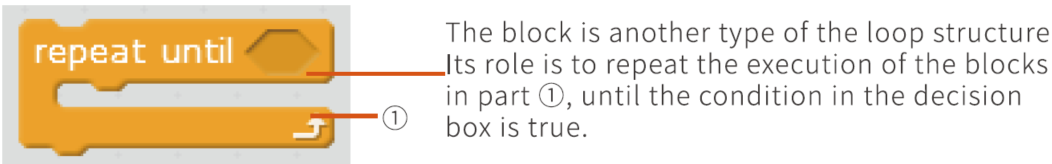


FIG. 6-8

Practice:

Select “M-Elephant” role. Write the program as shown in FIG.6-9 in the role and run it. The little elephant will stop the motion when moving to the right edge of the stage.

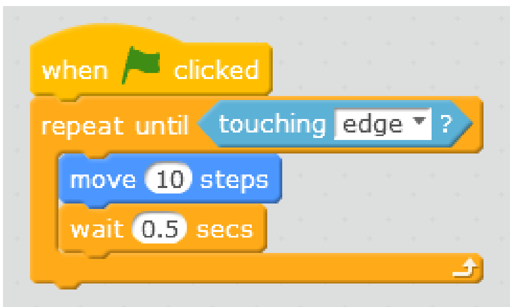


FIG. 6-9

4. Master the usage of "clone" blocks.

In the game, you often need multiple same roles, whose action scrip and shape are identical. What's different is only the position of these roles, such as bullets, fireworks, stars, and so on. It is especially troublesome to import a large number of role operations at this point. In WeeeCode software, there is a "clone" block, the clone is equivalent to copy the role, but it is not a new role, but only a clone. A clone has an entirely same attribute with the role of parents.

The WeeeCode software uses three blocks to realize clone. We can find the following three blocks in the "Control" category:

1. 

: for cloning specified sprite.

2. 

: as a program starting event, when the roles are cloned, it will execute the program in the event.

3. 

: this block is used to delete the clone.

Practice:

Add "Basketball" in the role database and write the program as shown in FIG.6-10 in the role. Observe the experiment effect and deepen your understanding of clone blocks.

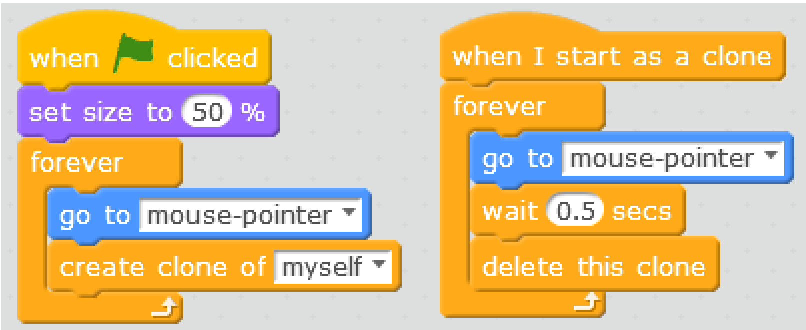


FIG. 6-10

Program effect: Move the mouse after running the program, and the stage effect will be as shown in FIG.6-11. The parent role will always follow the cursor, and continuously produce clones. When the clone is cloned, it will have the same attribute as the parent role---the same size. The clone will always follow the cursor and delete the clone after 0.5 seconds, which will result in a phantom effect.

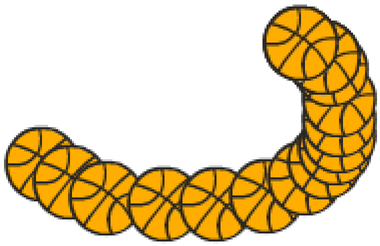


FIG. 6-11

Ⅲ Writing game program (100min)

1. Set stage backdrop and sprite

a. Delete default elephant roles and select “sprkling” as the game background from the background database (Background selection can be different).
b. Select “Basketball” from the role database and rename it as “pinball."
c. Select “Paddle” and rename it as “Baffle."
d. Select “Button3” and rename it as “Brick 1”.
e. Select “Button3” and rename it as “Brick 2”.
After you finish the above procedures, the stage effect will be as shown in FIG. 6-12(The position may be different.) and the sprites pane will be as shown in FIG. 6-13.

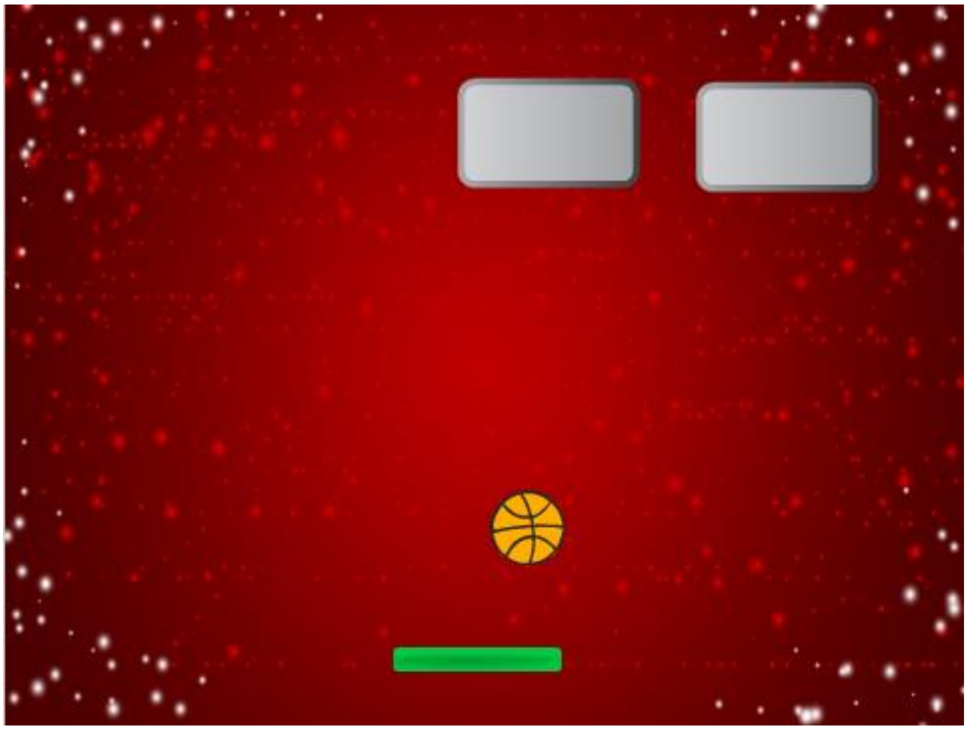


FIG. 6-12

FIG. 6-13

2. Write "Pinball" and "Baffle" Control Program

a. Shrink the "pinball" role and write the mobile animation program of the "pinball" as shown in FIG. 6-14.

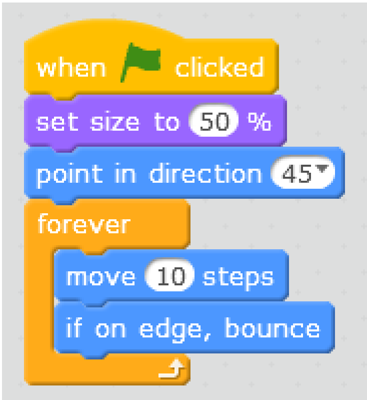


FIG. 6-14

Program effect: The "pinball" is always moving in four directions, 45 ° (upper-right), -45 ° (upper-left), - 135 ° (lower-left) and 135 ° (lower-right), and the basic animation of the "pinball" is finished.

Note:

a) If there is no specified facing direction, you will find "pinball " just moving back and forth in a horizontal direction. This is because the default facing angle is 90 ° to the right.

b) The above program is written in the role of "pinball."

b. Set the initial position of "baffle" to be at the bottom of the stage and control its movement by remote control. Its program is as shown in FIG. 6-15.

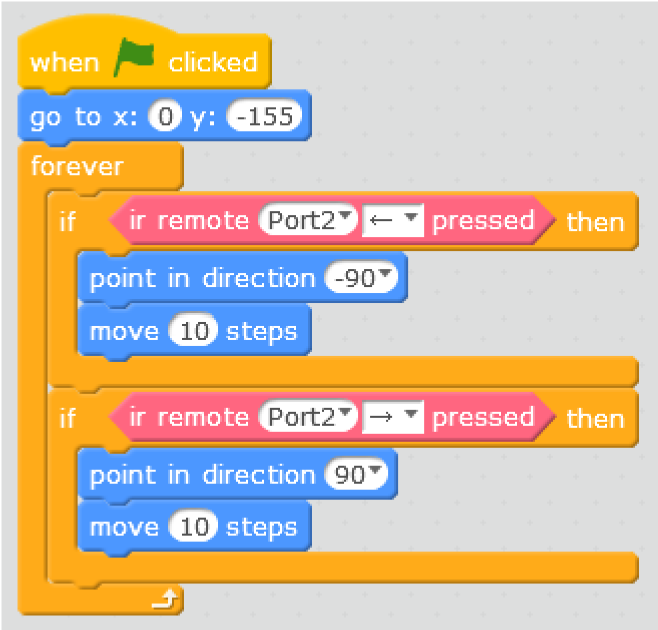


FIG. 6-15

Program effect: The “baffle” can move according to the “←” and “→” key of the remote control. “baffle” program finished.

Note:

a) The moving speed of the “baffle” can be modified according to the difficulties of the games.

b) The program is written in the role of “baffle."

c. Write “pinball” launch program: In the games, the pinball follows baffle to move before the launch. Only when the OK key is pressed, it will be emitted to break the bricks. Write the program in the role of pinball as shown in FIG. 6-16. The pinball program after modification will be as shown in FIG. 6-17.

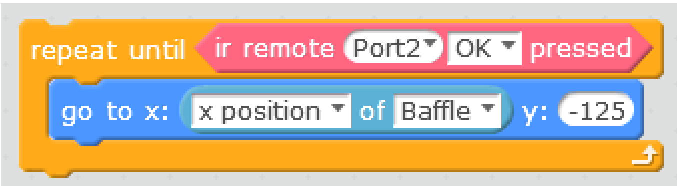


FIG. 6-16

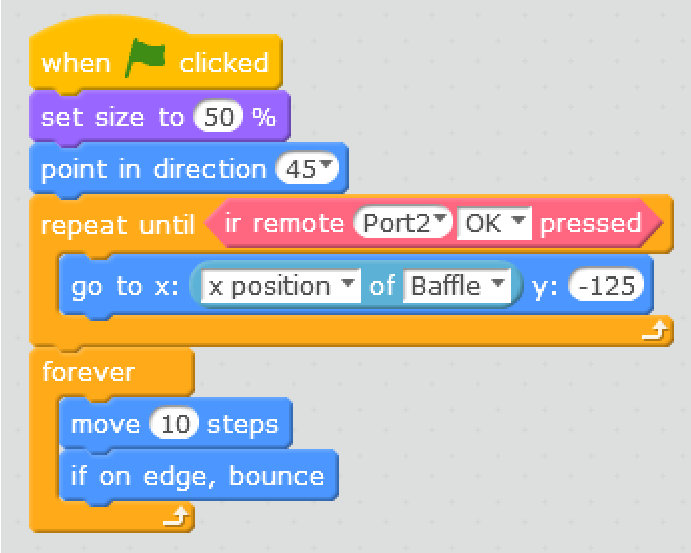


FIG. 6-17

Note: Please set the Y-axis to “125”, put the “pinball” above the “baffle” but not stuck in the middle of the “baffle."

d. Write the rebound program that “pinball” touches the “baffle” and the program is as shown in FIG. 6-18.

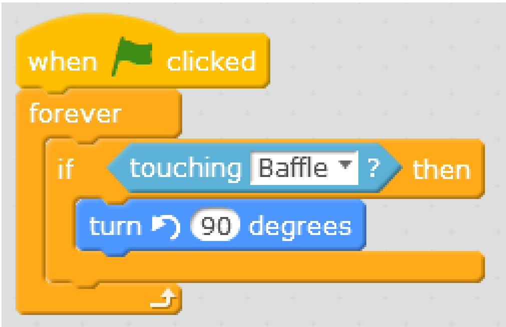


FIG. 6-18

Program effect: (take turn left 90° for example) when the “pinball” moves from the upper left to the lower right and rebound to upper right when touching the “baffle," the animation run smoothly. However, when it moves from upper right to lower left and rebound to upper right when touching the “pinball," it does not run very smoothly. It is because of the 90 ° rotation. However, it does not affect the overall effect, no need to change here.

Note: The reason for setting to 90 ° is to make the "pinball" rotate a certain angle when encountering "baffle," and so it can be rebound at a right angle.

After finishing the above procedures, the overall program of the “pinball” is as shown in FIG. 6-19 and the overall program of the “baffle” is as shown in FIG. 6-15. Now the “pinball” can be rebound after it touched a wall and the remote control can control the “baffle” to move left and right.

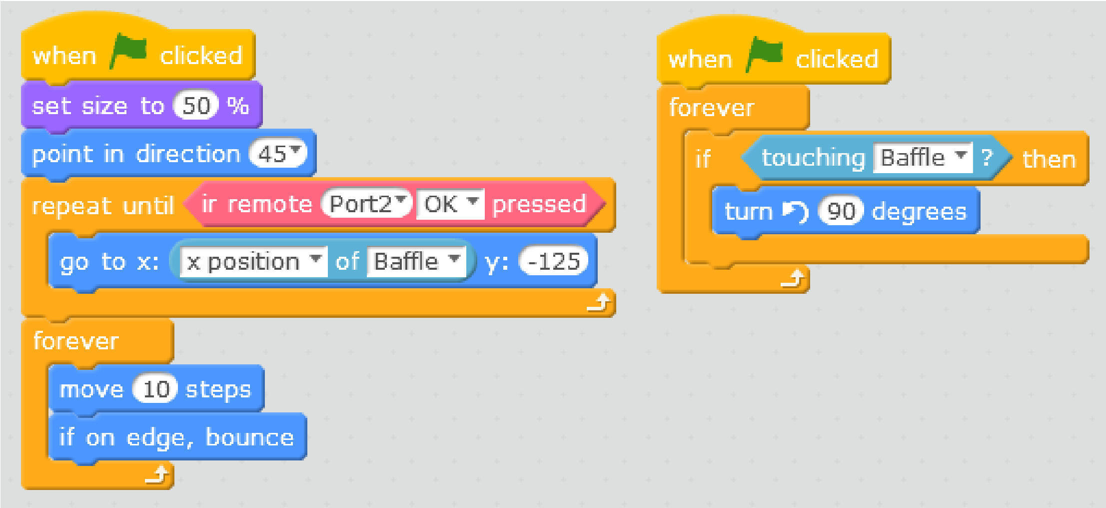


FIG. 6-19

3. Write "Brick 1" Control Program

a. Set the initial position and size of "brick 1". It is not possible to have only a brick in the game, and we are going to use the clone method to clone other bricks automatically. When the game starts, repeat 5 times and clone itself. After each cloning, it will move 60 steps to prepare the next clone. The program is as shown in FIG. 6-20, and the stage effect is shown in FIG. 6-21.

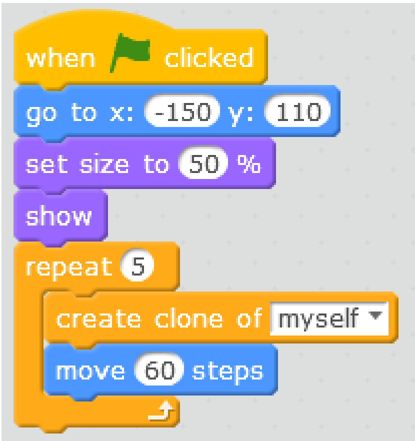


FIG. 6-20

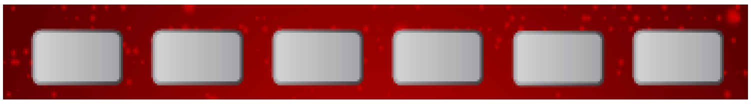


FIG. 6-21

b. Add the animation and scores setting of "Brick 1" as follows: in the game, if a "pinball" encounters "bricks" 1, "bricks 1" will disappear. Moreover, to make the game more interesting, there is usually scores setting, which we use variable to represent. The modification of the "bricks 1" program is as shown in FIG. 6-22.

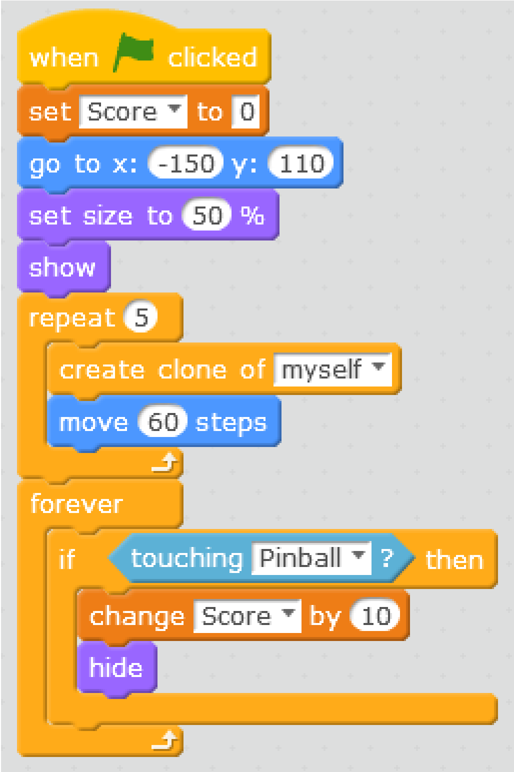


FIG. 6-22

Program effect: Only when the “pinball” touches the "parent" bricks, the score will increase, and the pinball will hide. However, there is no scores increase and hiding for five clones of “Brick 1”. It also needs scores increase and hiding when the clone touches “pinball." The written program is as shown in FIG. 6-23 and the master program of “Brick 1” after modification is as shown in FIG. 6-24.

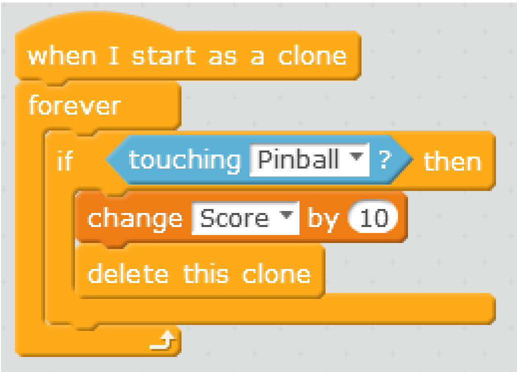


FIG. 6-23

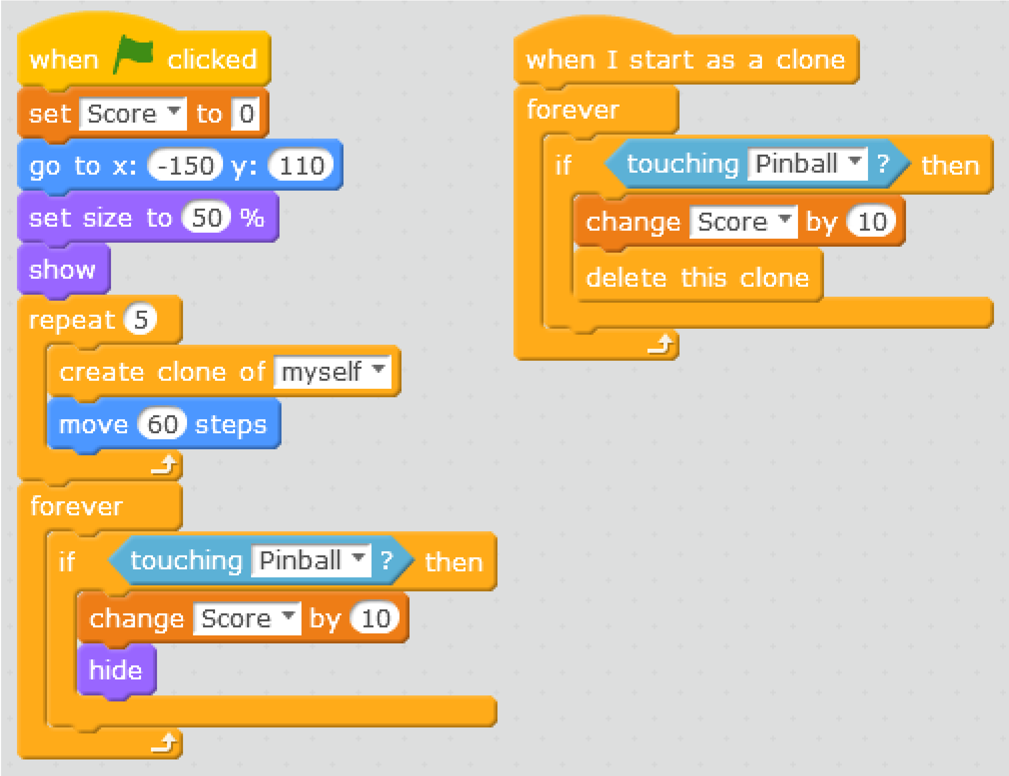


FIG. 6-24

c. Increase the number of bricks: The number of the bricks in the game is too few, and we will make a row of bricks again. The programming of the "Bricks 2" is similarly to "bricks 1". You will only need to change the y-axis of the initial position to 70, and avoid position overlapping of the "Brick 1" and clone. The program is as shown in FIG. 6-25 and the stage effect after running the program is shown in FIG. 6-26.

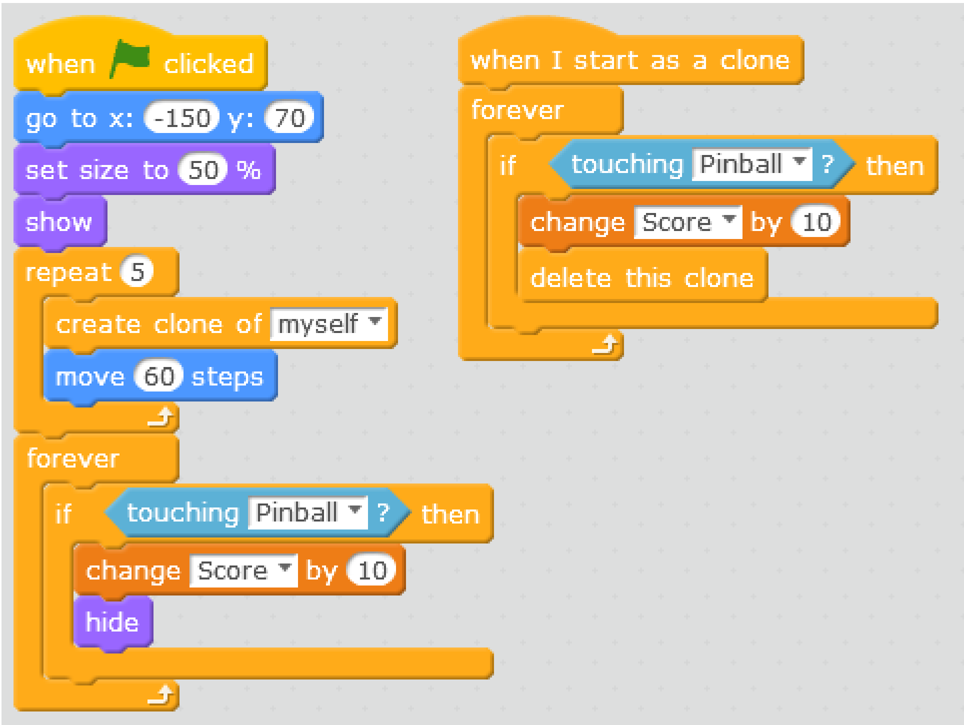


FIG. 6-25

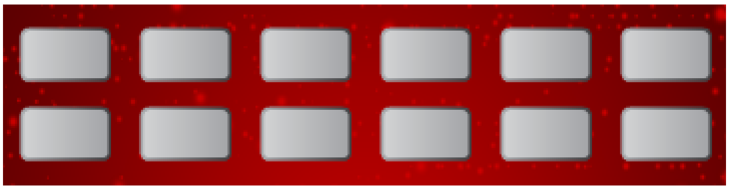


FIG. 6-26

d. Write the program that the "pinball" will be rebound after it touches "Brick 1" and "Brick 2" and write the program in the "pinball" role as shown in FIG. 6-27.

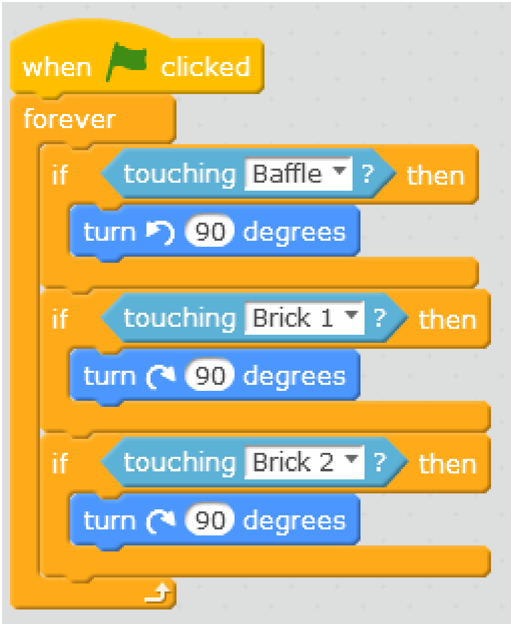


FIG. 6-27

After completing the above steps, the master program of the "Pinball" will be as shown in FIG. 6-28, the master program of the "Baffle" will be as shown in FIG. 6-15, the master program of the "Brick 1" will be as shown in FIG. 6-24, and the master program of the "Brick 2" will be as shown in FIG. 6-25. The total game program has been completed.

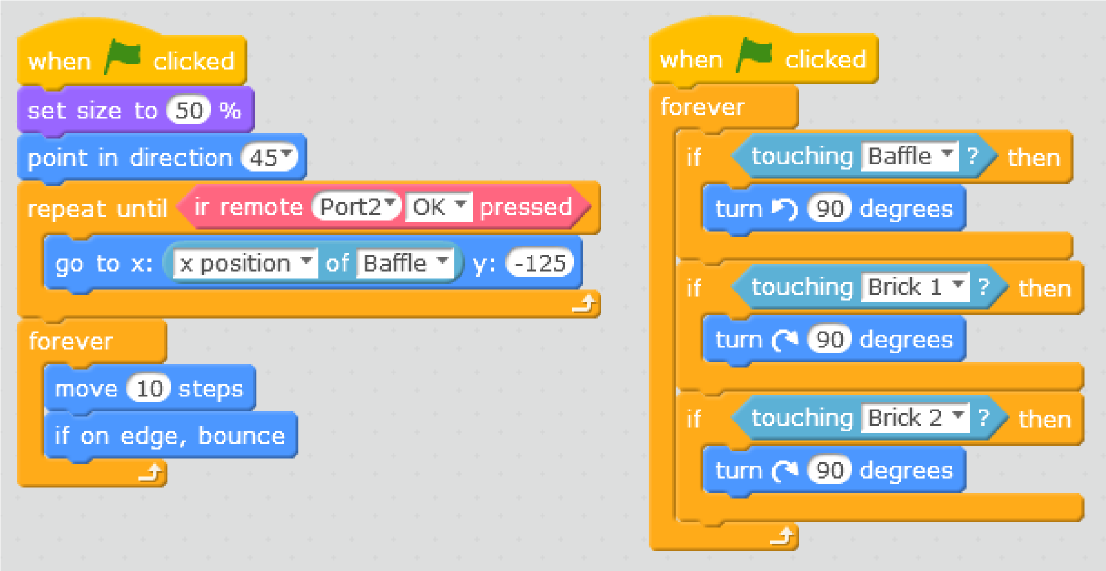


FIG. 6-28

Note: The above program is for reference only. If the student can complete the game in other ways, teachers may give rewards to them.

IV Classroom Development and Classroom Evaluation (10min)

1. Classroom development: Increase the variable of the life value, make the judgment of game victory and failure, and optimize the game experience (selected lecture according to the class teaching pace).

a. Create the "life" variable and show on stage. When the game begins, set the initial value of the "life value" variable as "2". The initializer can be written in any role and the stage effect as shown in FIG. 6-29.

FIG. 6-29

b. Add error judgment: when the "Baffle" does not catch the "Pinball," please use the Y-axis of the "Pinball" to determine whether "Pinball" fell to the bottom of the stage or not. If life decreases and returns to the "Baffle" center, the revised "Pinball" program will be as shown in FIG. 6-30.

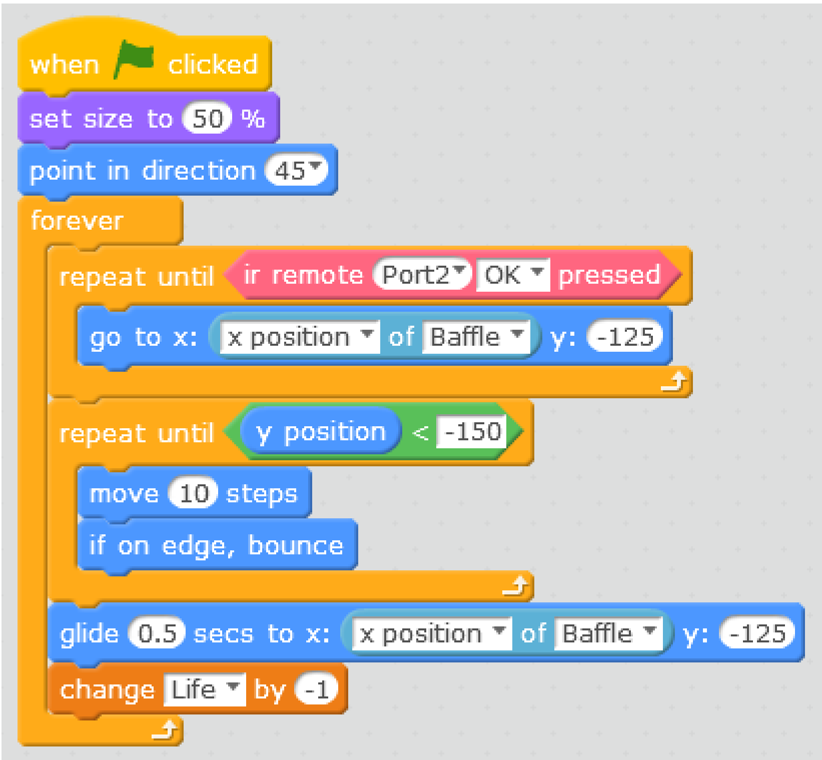


FIG. 6-30

3. Add win and lose judgment: There are 12 bricks in the game, 10 points for each, and the total finished scores are 120 points. If the score variable is 120, you win. The life is 2, and it will be one deducted one point for one fault. If the life is 0, you lose. The program is as shown in FIG. 6-31.

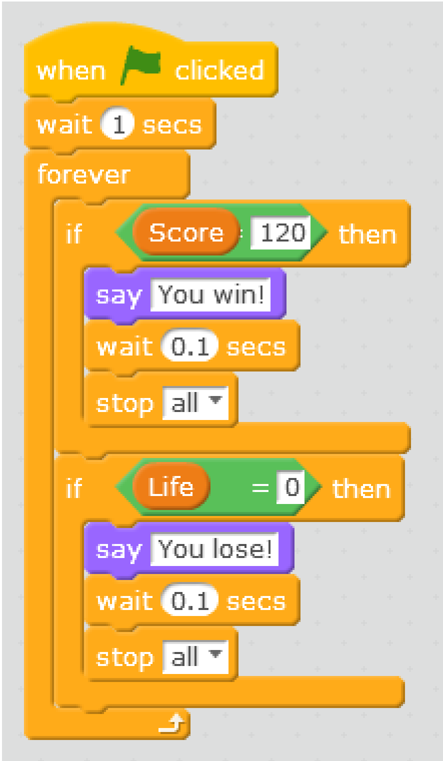


FIG. 6-31

Note: the judgment of game victory or defeat is added to the game. However, if in the last time you are won in the game and score is 120, once started again, it will meet the victory judgment to the scores, and then the game will stop. The defeat judgment will be in the same way. Therefore, we need to add time delay before the judgment.

For the program whose development part only needs to change the "pinball" role, the master program of "pinball" is as shown in FIG. 6-31 and FIG. 6-32.

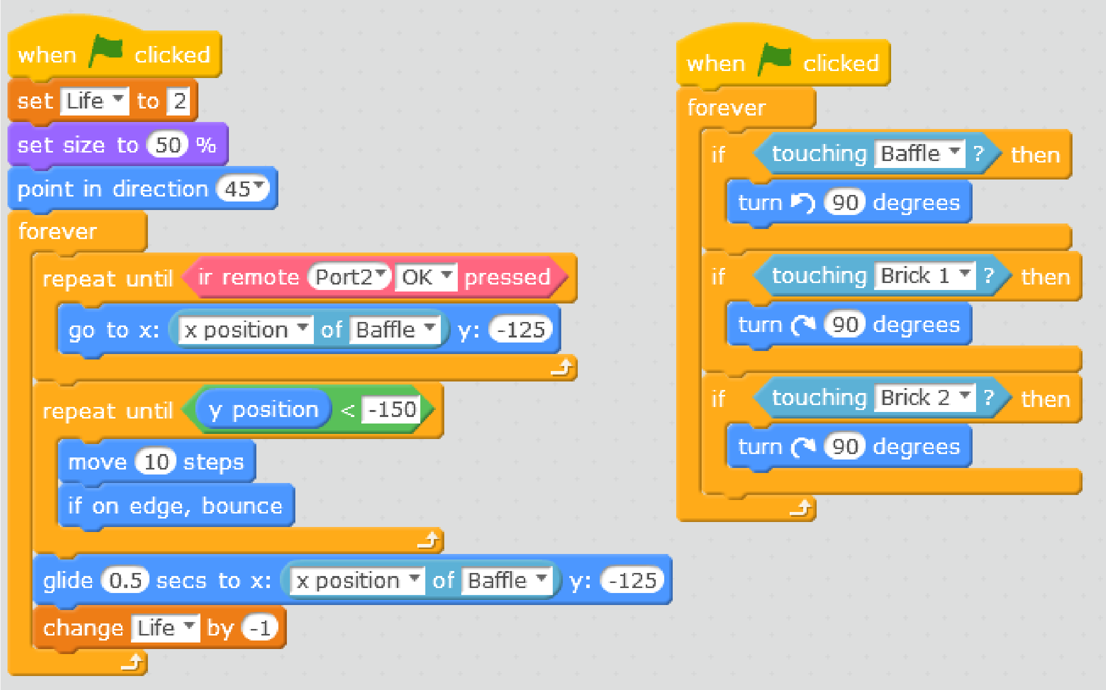


FIG. 6-32

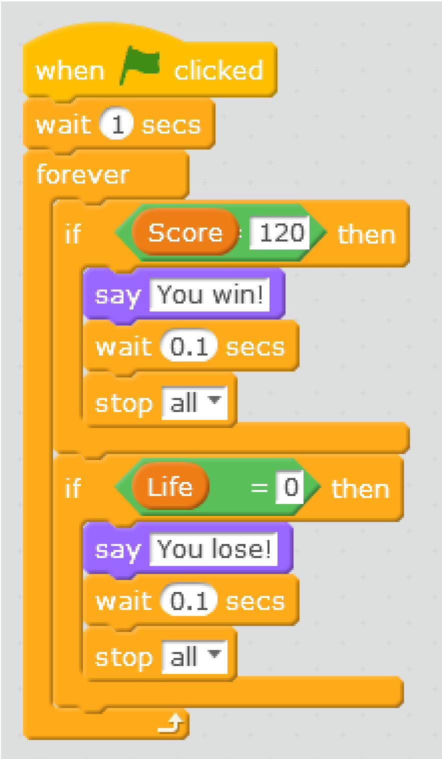


FIG. 6-31 (already existed above, repeat just for references)

2. Classroom evaluation: students' class performance will be evaluated according to the evaluation criteria.

Evaluation criteria:

1. Master and use the variable, clone and “repeat until” blocks.
2. Able to make pin-wall games.

Chapter 7 Dancing Robots

Summary

In this lesson, we begin to contact the RGB ultrasonic sensor and makes the robot's motion more intelligent by learning and using the RGB ultrasonic sensor.

Learning Objectives

To master the control method of RGB ultrasonic sensor and understand its working principle by combining the distance formula.

To understand mathematical logic “and," "or" and "not," and use mathematical logic to solve the judgment problem of a compound condition.

To make dancing robots.

Course Time

90 minutes.

Pre-class preparation

WeeeBot robots, computers with WeeeCode software installed, and USB download cables.

Teaching Process

I Introduction from Life(15min)

1. Know about the characteristics of sound waves through examples from life.

In life, we went to the valley shouting, and we often can hear the echo. However, if we shout to the sky, we could not hear the echo. Thus, we can know the characteristics of sound waves that it can reflect if it meets objects.

2. The application of ultrasonic waves in life and nature.

In nature: Many animals in nature can transmit and receive ultrasonic waves, and the most obvious is in bats and dolphins. Bats can fly in the dark, can emit sound waves, and can tell if there are obstacles in front of them through whether there is the reflection of the sound waves that touches obstacles, that is, they can “see” with their ears. Moreover, it is the same for dolphins. After they emit ultrasonic waves in the water, if there are targets, the ultrasonic waves will reflect, and so they can detect and prey on the fish they love to eat accurately.

In life: The sweeping robot determine whether there are obstacles to avoid hitting the wall by ultrasonic detection; Fishing boats determine whether there are fish in the sea by ultrasonic detection.

Conclusion: the ultrasonic wave is a kind of acoustic wave that can not be heard, and has the characteristic that it will return when touching objects.

Ⅱ Know about Modules and Use It (25min)

1. Know about the RGB ultrasonic sensor and understand its working principle.

The RGB ultrasonic sensor of the robot is as shown in FIG. 7-1, and we can see that it has two "eyes," one to emit sound waves and one to receive sound waves.

FIG. 7-1

In math class, we have learned the formula for distance is The distance = speed * time, which is also the working principle of the RGB ultrasonic sensor. Because sound waves travel through the air at a certain speed, the robot can automatically record the time it takes to send and receive sound waves. Then,

Used time on the road = received time of the sound wave - emitted time of the acoustic time.

The distance of back and forth= propagation speed of the sound wave* used time on the road.

One-way distance = distance of back and forth /2.

2. Master the Usage of RGB ultrasonic sensor.

We can find the control blocks of the RGB ultrasonic sensor in the “Robot” category, as shown in FIG.7-2, and the distance between the robot and the front obstacle can be detected by using the block.

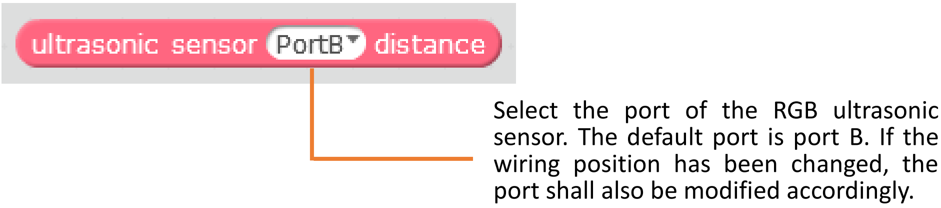


FIG. 7-2

Practice:

Select “M-Elephant” role and write the program as shown in FIG. 7-3 in the online mode. Using an obstacle (or hand) to block in front of the RGB ultrasonic sensor, let the role speak the distance between the RGB ultrasonic sensor and the obstacle.

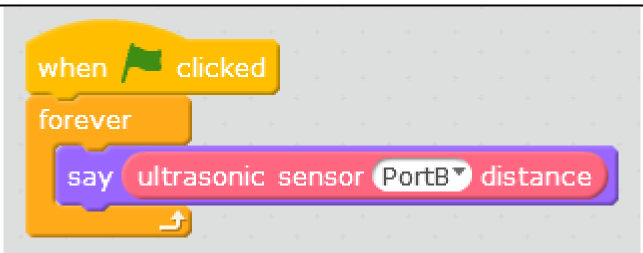


FIG. 7-3

Note:
a) If there is no reading, at first please check whether it enters the online debug mode. If it has entered the online mode, please check whether the RGB ultrasonic sensor has to be connected to port B.
b) The reading unit obtained by the control block of RGB ultrasonic sensor is centimeter.

Exploration:
a) Run the program of FIG.7-3, use obstacles to block the RGB ultrasonic sensor and observe the changes in the reading.
b) Run the program of FIG.7-3, face the robot to the place without obstacles and read the arisen max. Reading.
c) Run the program of FIG.7-3, use an obstacle to block in front of the robot and in the position of reading 4, move the obstacle to get close to the RGB ultrasonic sensor and observe the changes of reading.

Conclusion:
a. When obstacles block the RGB ultrasonic sensor entirely or when the RGB ultrasonic sensor faces the distant place, the sound wave sent from the RGB ultrasonic sensor all cannot be received. Therefore, the distance between the RGB ultrasonic sensor and the obstacle cannot be measured, and the maximum distance it can detect will be 500 cm.
b. When the distance between the obstacle and the RGB ultrasonic sensor is less than 4cm, the distance error detected by the RGB ultrasonic sensor is relatively more significant. Therefore, this situation shall be avoided as far as possible.

The RGB ultrasonic sensor also has the function of RGB, and the control block is as shown in FIG. 7-4.

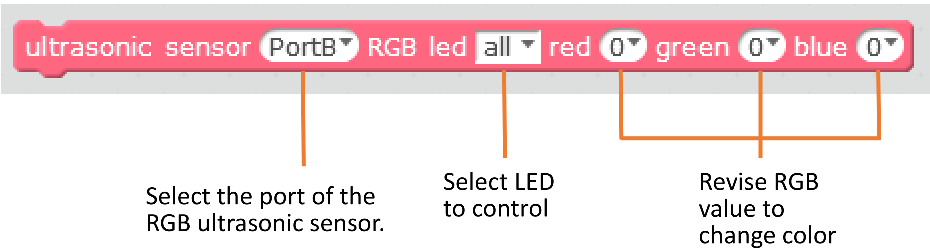


FIG. 7-4

Ⅲ Programming and Optimization (40min)

1. Write the program for dancing robots and make the robots follow your hands to move.

Q: Tango is a couple dance, and the dance step is that one person gets close and another moves away. How to write the program if you are going to let the robot dance with our hands and realize that it moves back when your hands get close, and it moves forward when your hands are far away from it?

Conclusion: To make the robot realize this effect, it is necessary to determine the motion mode of the robot by judging the distance value measured by the RGB ultrasonic sensor. The distance value of the RGB ultrasonic sensor gets smaller, indicating that the hands are getting closer and then the robot needs to step back when the value is small to a specific value. The distance value of the RGB ultrasonic sensor gets larger, indicating that the hands are getting farther and then the robot needs to step forward when the value is large to a specific value. We set the middle value to 20cm. Therefore, we can write the program as shown in FIG. 7-5.

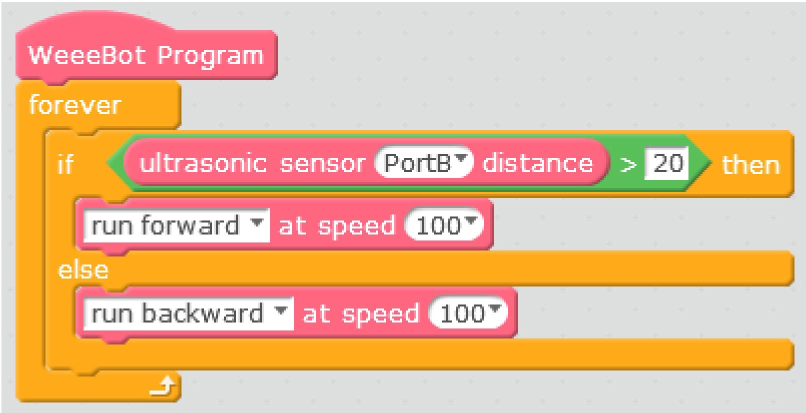


FIG. 7-5

Program effect： When the hands are far away, the robot step forward. When the hands get close, the robot will step back. However, when the hands are still, the robot will step back and forth repeatedly. This is because we have not set a range for the robot to stop the motion.

2. Understand mathematical logic “and," "or" and "not," and use mathematical logic to solve the judgment problem of a compound condition.

From the above situation, we can know that the program function needs to be modified as shown in FIG. 7-6.

The value measured by ultrasonic sensor	Motion of the robot
Larger than 20cm	Move forward
15~20cm	Still
Smaller than 15cm	Move backward

FIG. 7-6

Now the only problem is how to express the range of 15~20cm with the program. In the programming, we usually use the following blocks to express the judgment of the compound conditions.

a. 

: Only when both two conditions in the box are true, the result will be true. If one of the two conditions are false, the result will be false, as shown in FIG.7-7.

And	
compound conditions	Result
True+True	True
True+False	False
False+True	False
False+False	False

FIG. 7-7

b. 

: Only when both two conditions in the box are false, the result will be false. If one of the two conditions are true, the result will be true, as shown in FIG.7-8.

Or	
compound conditions	Result
True+True	True
True+False	True
False+True	True
False+False	False

FIG. 7-8

c. 

: Take the contrary result. The contrary result of true is false and the contrary result of false if true, as shown in FIG. 7-9.

Not	
compound conditions	Result
True	False
False	True

FIG. 7-9

Therefore, we can use the program as shown in the FIG. 7-10 to express the number of 15~20.

FIG. 7-10

3. Write the optimized program of the dancing robot

Combined with our previous analysis: The measured distance range of the RGB ultrasonic sensor is 4cm~500cm, and the error of values beyond that range will be huge, which will not be adopted by us. Therefore, we can write the program as shown in FIG. 7-11.

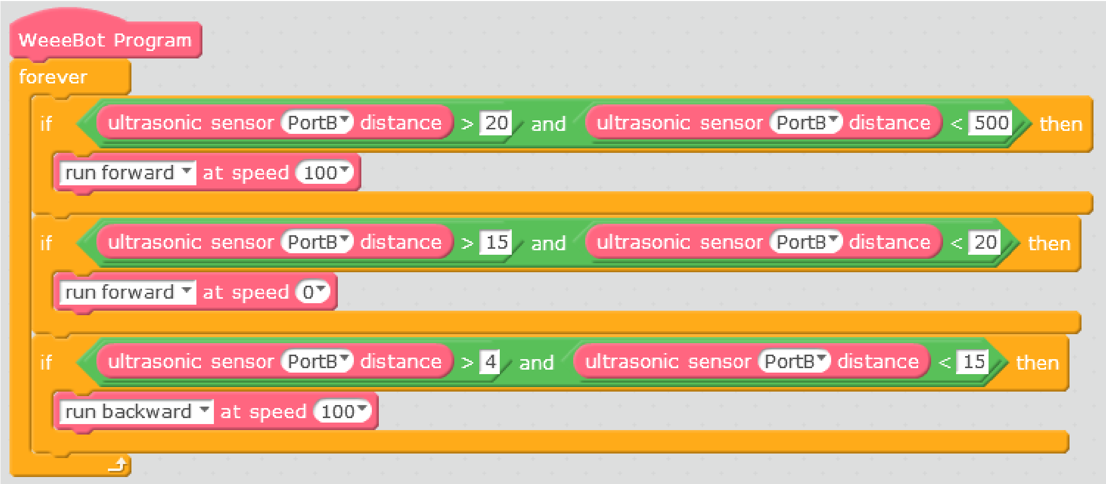


FIG. 7-11

IV Classroom Development and Classroom Evaluation (10min)

1. **Classroom development: make obstacle avoidance robots**

a. Simple obstacle avoidance robots: In the proceeding process, the car can detect whether there is an obstacle in the front through ultrasonic waves.

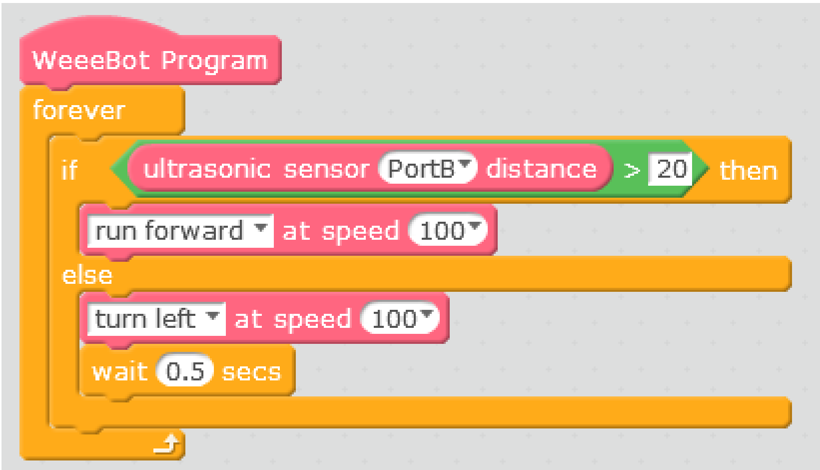


FIG. 7-12

b. Complex obstacle avoidance robots: In the proceeding process, the car can detect whether there is an obstacle in the front through ultrasonic waves. If there are obstacles, the RGB light of the RGB ultrasonic sensor will flash twice continuously, and the car will swerve. If there is no obstacle, the RGB white light of the RGB ultrasonic sensor will be on, and the car will step forward. The reference program is as shown in FIG. 7-13.

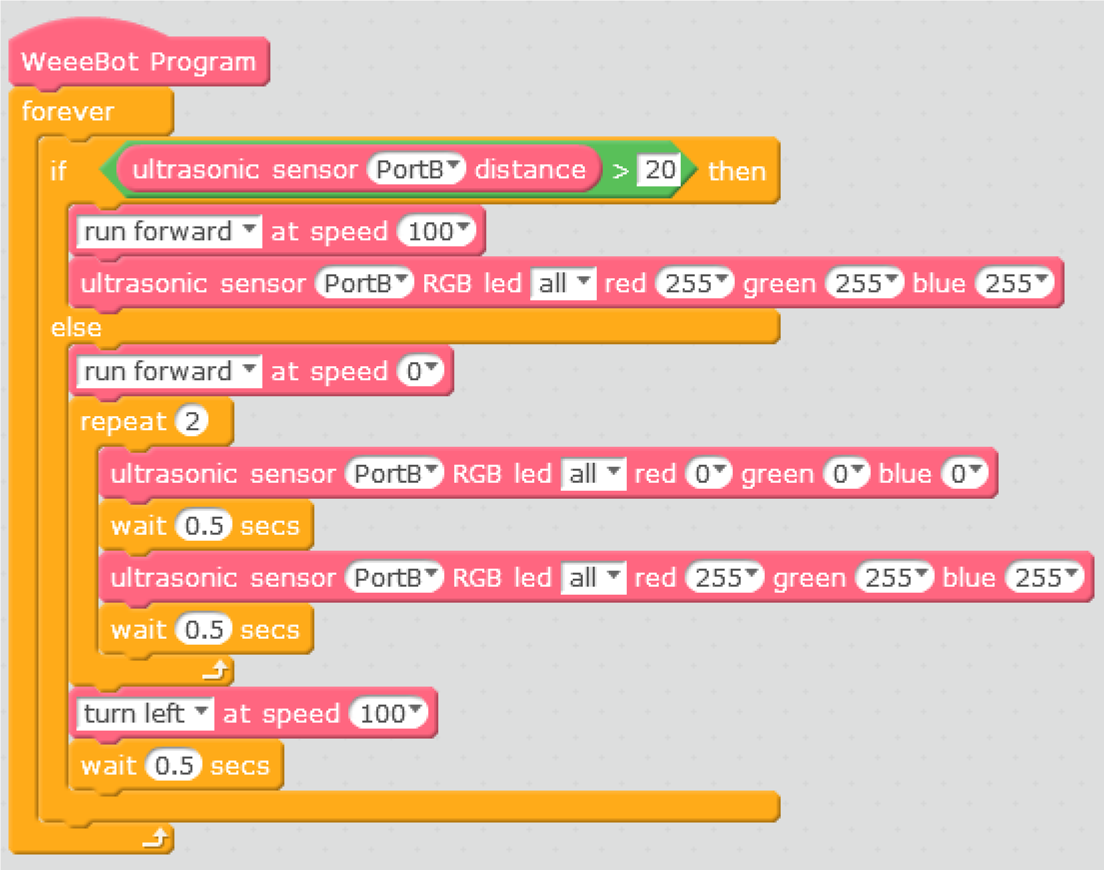


FIG. 7-13

2. Classroom evaluation: students' performance will be evaluated according to the evaluation criteria.

Evaluation criteria:

1. Proficient in using the measuring and RGB function of the RGB ultrasonic sensor.
2. Able to use mathematical logic to judge compound conditions.
3. Able to write a program and make dancing robots.

Chapter 8 Flying Elephant Games

Summary

After we mastered the usage of distance measuring of RGB ultrasonic sensors in the last lesson, in this lesson, we will use the RGB ultrasonic sensor to make a flying elephant game to add the fun of programming through the combination of software and hardware.

Learning Objectives

Strengthened in the mastering the usage of variable and distance measuring of RGB ultrasonic sensors.

Understand broadcast, modeling transformation, random number, and " Wait until" blocks in the WeeeCode software and master its usage.

Able to write programs, discover and solve problems continuously during this process.

Course Time

180 minutes.

Pre-class preparation

WeeeBot robots, computers with WeeeCode software installed, and USB cables.

Teaching Process

I Course Introduction (20min)

1. Let students know about the gameplay through game sharing, as so to provide ideas for developing games.

Teachers can ask students to share and summarize the functions of several roles of the game by showing the related pictures of the flappy bird game or by letting them experience the flappy bird games, as shown in FIG. 8-1.

Button	1. It will become larger when touched by the cursor.
	2. The game will begin once it is clicked.
Water pipe	Appear on the rightmost of the screen and keep moving.
Little bird	1. It can be controlled to move up and down to avoid obstacles.
	2. It will get scores for avoiding the water pipe and will die when it touches the water pipe.

FIG. 8-1

Ⅱ Master the relevant knowledge points of making games.

Note: this link is to learn the necessary knowledge for students to develop games, and teachers can also explain it in the process of developing games.

1. Master the modeling switch function of the roles to make the animation of the roles more vivid.

In the game of flappy birds, birds are required to flap their wings during flight. This animation effect is a motion animation achieved through several different movements of the bird. The different actions of a role in WeeeCode software are called costume, and there are two blocks in the "Looks" category that can be used to control the costume switch, as shown in FIG. 8-2. Through the costume switch, it can make the role move, which makes the sprite movement more vivid.

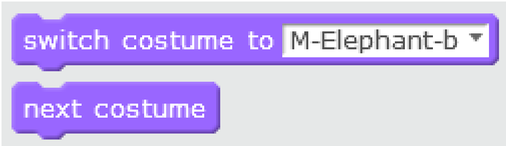


FIG. 8-2

: Each costume has specified name, which can be checked in the costume tab. This block can let the sprite switch to other costume.

: Sprite can switch the costume according to the order in the costume tabs.

Practice:

a. Select "Boy3 Walking" role, as shown in FIG.8-3. Select the modeling tabs, and you can see that the roles have five different modelings, as shown in FIG. 8-4.

FIG. 8-3

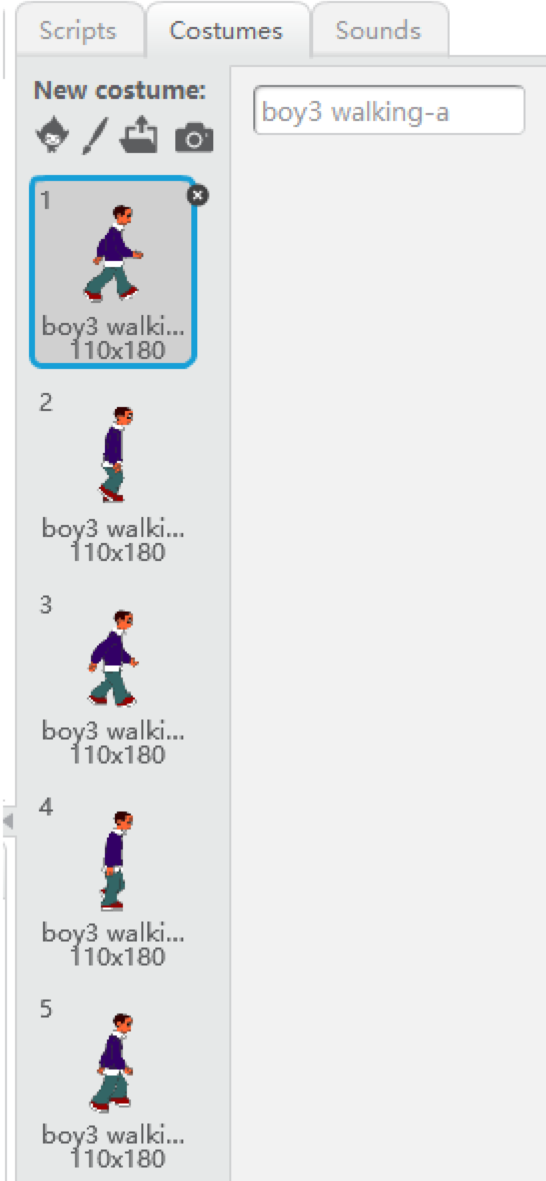


FIG. 8-4

b. Write the program in the role, as shown in FIG.8-5 and view the program effect.

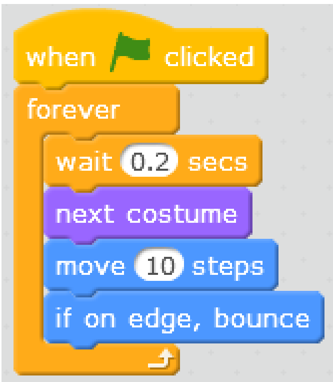


FIG. 8-5

Program effect: The role walks on the stage but will be upside down if touching the wall. This is because the rotation mode of this role is “Arbitrary rotation” mode. In FIG. 8-6, the rotation mode has been changed to “Horizontal rotation” mode. You can see the role moving around on the stage.

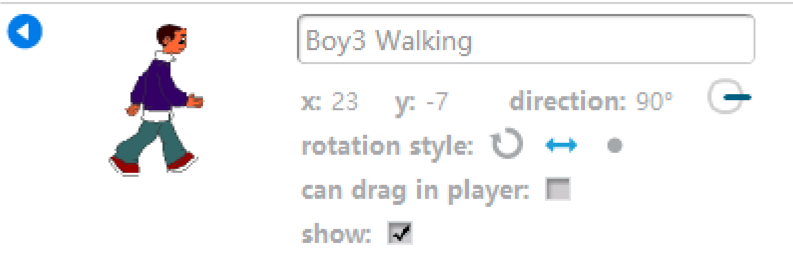


FIG. 8-6

2. Master the usage of broadcast block and understand how to use the broadcast function to control the linkage among sprites.

In WeeeCode software, the role of broadcast block is used for the mutual control among sprites. There are mainly three types of application forms:

a. "point-to-point" broadcast, that is, “sprite 1” control “sprite 2” through broadcast.

b. "point-to-surface" broadcast, that is, “sprite 1” can control multiple sprites simultaneously.

c. "Surface-to-point" broadcast, that is, there are multiple control terminals to transmit the broadcast, while the controlled side realizes corresponding response actions.

The use of "broadcast" plays a vital role in the cooperative relationship among multiple roles. In the “Events” category, we can see the three blocks of broadcast function:

a. 

: Broadcast a piece of news to other sprites, trigger them to do something and wait for them to complete and then execute the script behind the block.

b. 

: Broadcast a piece of news to other roles, and then execute the script behind the block.

c. 

: Linked with the broadcast transmission function and act as the starting event for controlling the running of the program after it received information.

a. Add spirtes “Calvrett” and “Breakdancer1”, as shown in FIG. 8-7.

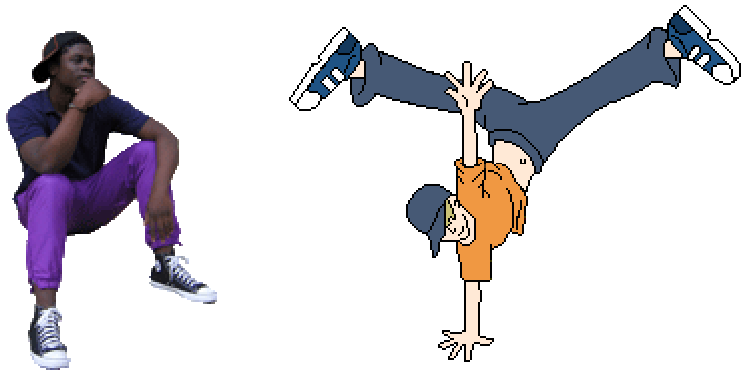


FIG. 8-7

b. Write program for "Calvrett," as shown in FIG. 8-8.

FIG. 8-8

c. Write program for “Breakdancer1”, as shown in FIG. 8-9.

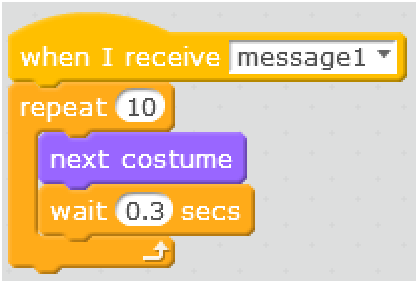


FIG. 8-9

Program effect: After running the program, sprite "Calvrett" broadcast a message and wait. The "Breakdancer1" will begin the costume switch once receives the message. When all the costumes of "Breakdancer1" are switched, the "Calvrett" will start to switch again.

Exploration: What will happen if the “broadcast and wait” block is changed to the “broadcast” block in the sprite “Calvrett”?
Program effect: The costumes of the two roles will be switched simultaneously.

Note: You can create broadcast and modify the message name in the drop-down option of the "broadcast" block and "broadcast and wait" block. In the process, the linkage function can be achieved if the message name of the "broadcast" block and "broadcast and wait" block is corresponding with that of the "when receiving the broadcast" block.

3. Understand the meaning of random number block and grasp the usage of pick random block.

In mathematics and logic operation module, we can see the block as shown in FIG. 8-10. The block has two parameters that can be modified. The value of the two parameters forms a range, in which a number will be generated randomly.

FIG. 8-10

Practice:

Select sprite "M-Elephant" and write the program in the role, as shown in FIG. 8-11, and then observe the program effect.

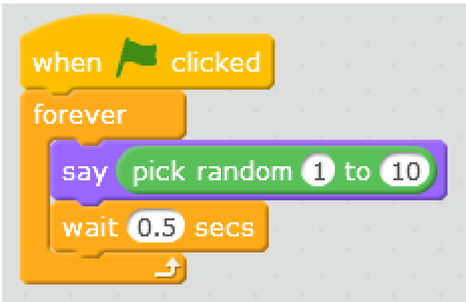


FIG. 8-11

Program effect: The program will generate a number from 1 to 10 continually (including 1 and 10).

4. Master the usage of "wait until" block.

"Wait until" block is in the "Control" category, as shown in FIG. 8-12. The function of this block: always judge the conditions of the box until when the condition in the box is true, the program will go down; otherwise the program has remained in the position, which is often used in the case that needs to repeat judging the conditions in the acyclic structure.

FIG. 8-12

Practice:

Select "M-Elephant" role and write the program in the online mode, as shown in FIG. 8-13, and observe the program effect.

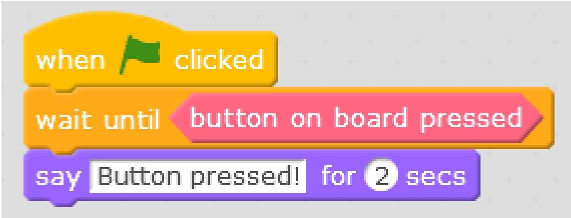


FIG. 8-13

Program effect: Run the program, press the button on the mainboard as shown in FIG.8-14, and then the stage will have the effect as shown in FIG. 8-15.

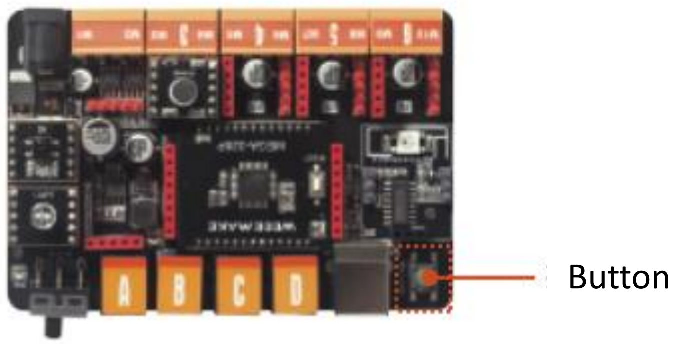


FIG. 8-14

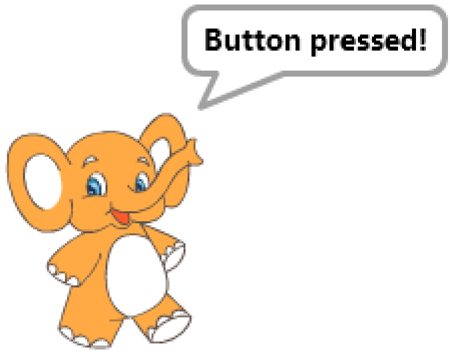


FIG. 8-15

Ⅲ Write a game program(100min)

1. Set stage backdrop and sprites

a. Select “blue sky” from the backdrop library as the stage background.
b. Rename “M-Elephant” like “Elephant”
c. Select “Waterpipe” from the sprites library and rename as “pipe."
d. Select “Button2” from the sprites library and rename as “start button."

After completing the above procedure, the stage will be as shown in FIG. 8-16 (the position of the role may be different).

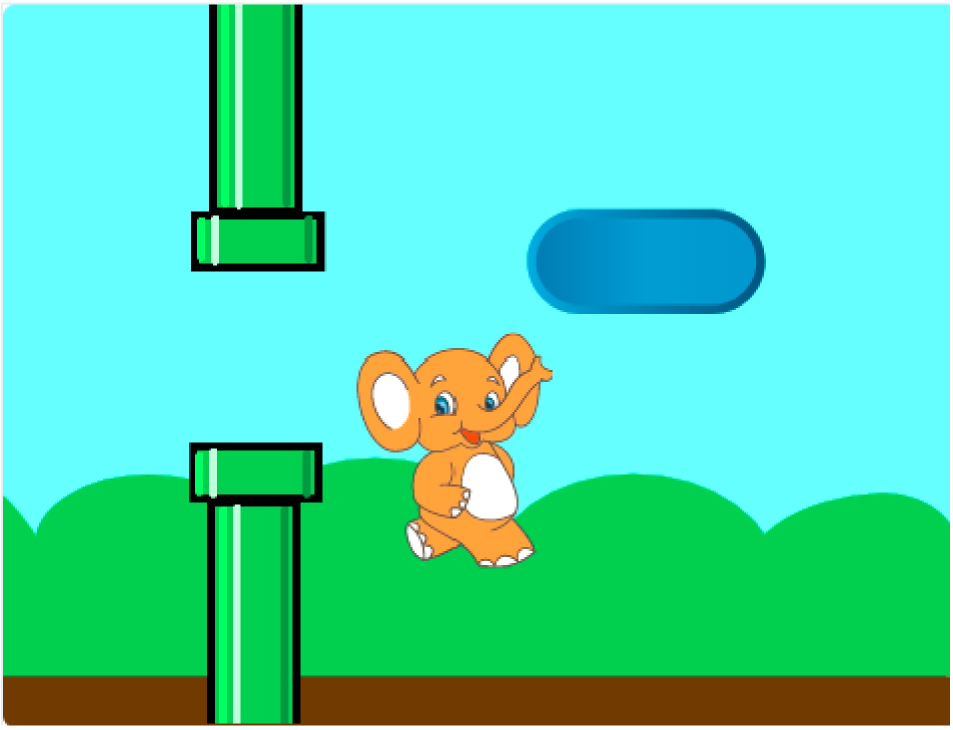


FIG. 8-16

2. Write the animation and control program of the "start button."

a. Set the initial position of the "start button" and make the animation effect: zoom in when the cursor touches the role and get back when the cursor moves away. The program is as shown in FIG. 8-17.

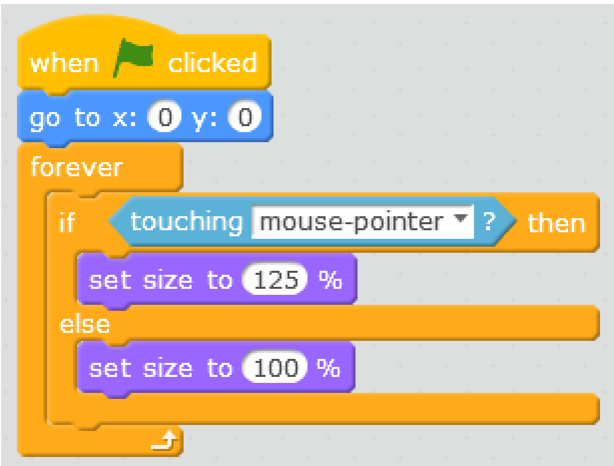


FIG. 8-17

b. Set the beginning of the games when the "start button" is clicked. Other roles can be linked through full function, and the program is as shown in FIG. 8-18.

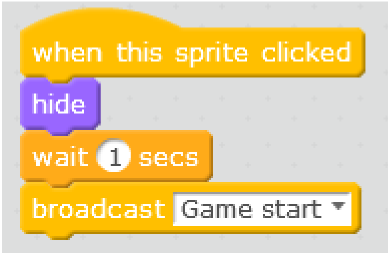


FIG. 8-18

Program effect: Run the program. When the “start button” is clicked, it will hide. Moreover, it tells other sprites the beginning of the games through the broadcast. However, we found that when pressing the "start button” to run the program later again, the "start button” disappears. This is because when running the program, we have not display the "start button" after we hide the role. The optimized "start button” program is as shown in FIG. 8-19.

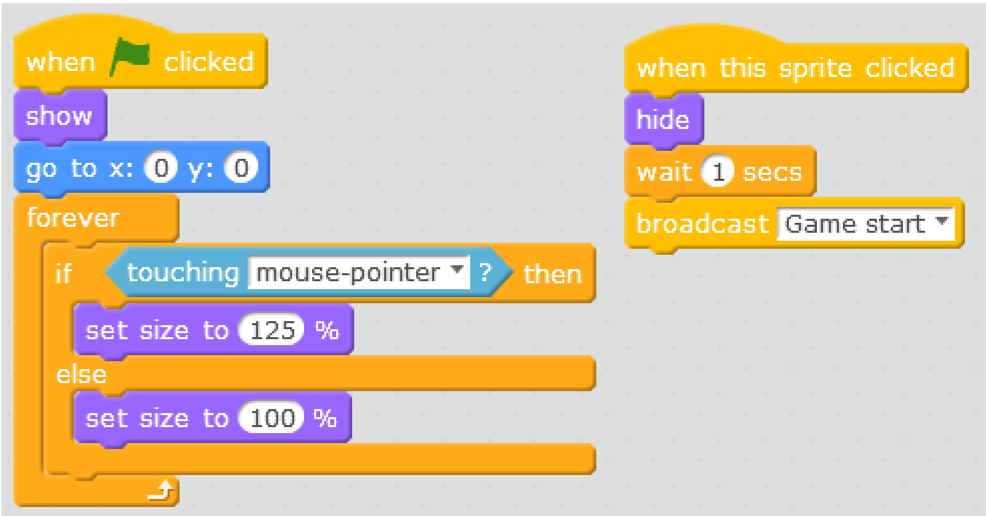


FIG. 8-19

3. Write the animation and control program of the “Elephant."

a. Set the size and initial position of the “Elephant," write the control program of the “Elephant," as shown in FIG. 8-20.

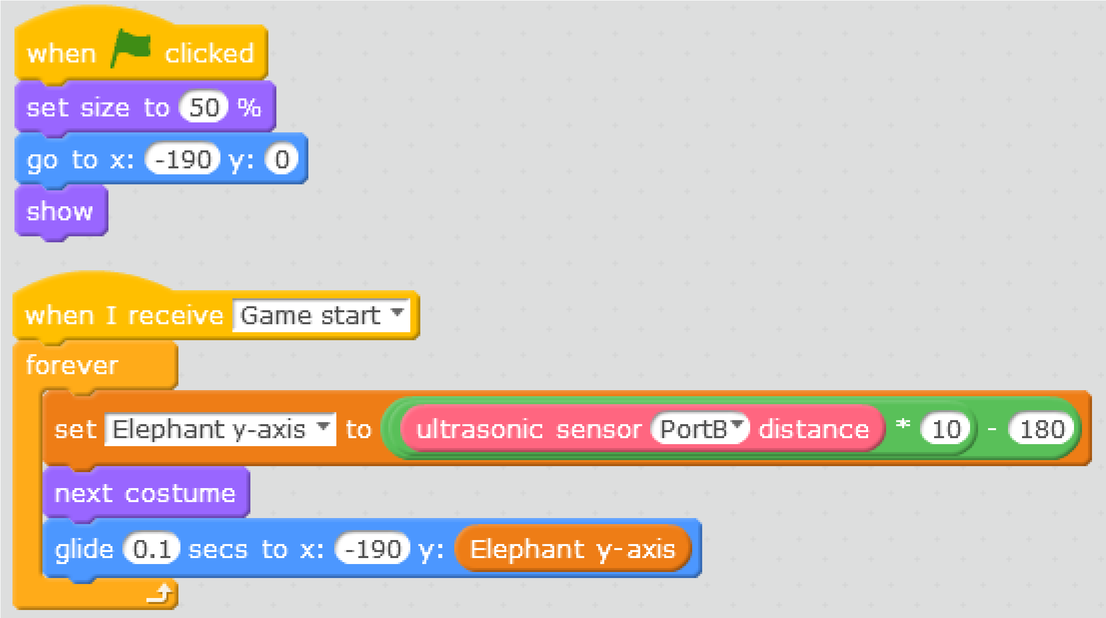


FIG. 8-20

Note:
a) The size of the "elephant" shall be set to 50%, and if it is too large, it will be difficult to escape the "pipe."
b) The operation of the "elephant y-axis" is first multiplied by 10, and the unit of RGB ultrasonic sensor detection value shall be changed to the millimeter, to enlarge the fluctuation range of the detection value of RGB ultrasonic sensor. Then minus 180 and you can change the detection value to the corresponding coordinate value on the stage, that is, if the detection value is 0, then it shall be at the bottom of the screen, and the y-axis will be -180.

Program effect: After running the program, the "elephant" will move to the leftmost position of the screen. You can control the "elephant" to move up and down by controlling the distance between the RGB ultrasonic sensor and the obstacle (wall or hands). The longer the distance is, the higher the "elephant" flies. However, when the detection value of the RGB ultrasonic sensor is too large, the "elephant" will be beyond the top of the stage. Therefore, the maximum value of the "elephant y-axis" variable needs to be limited. The optimization program is as shown in FIG. 8-21.

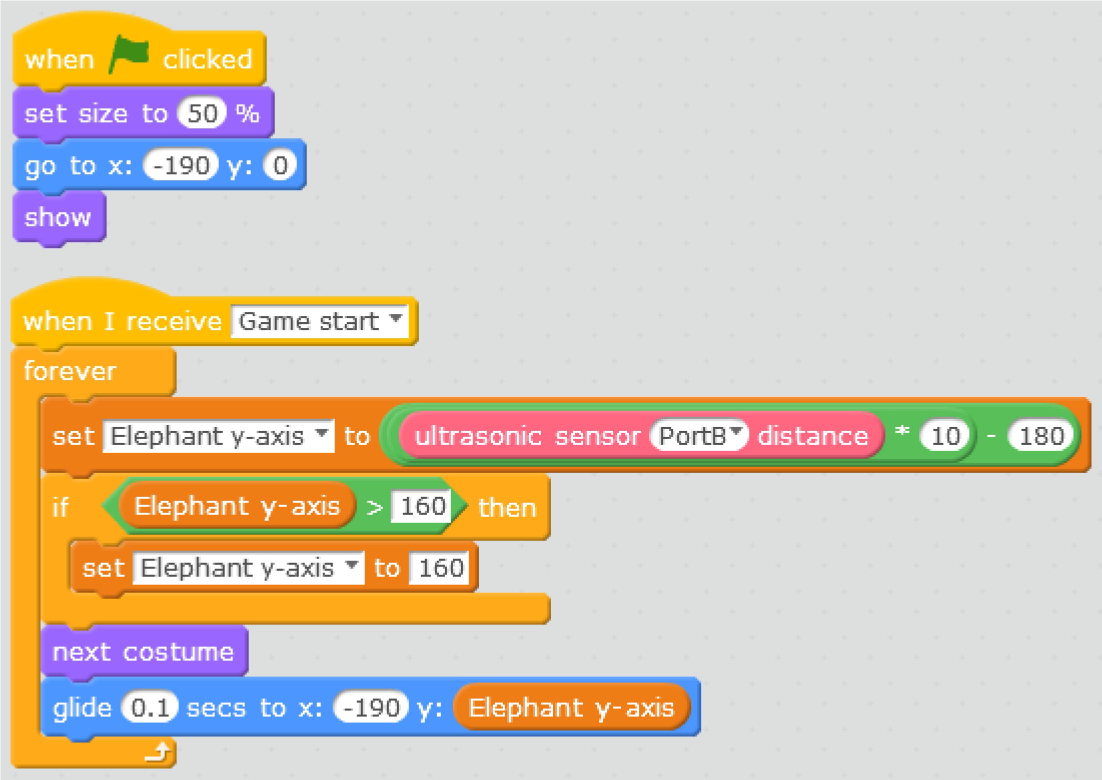


FIG. 8-21

4. Write the animation and motion program of the "pipe."

a. Initialize the setting of the variable such as game scores, motion speed of the "water pipe," and the fail state of the "elephant," and set the initial position and size of the "water pipe." The program is as shown in FIG. 8-22.

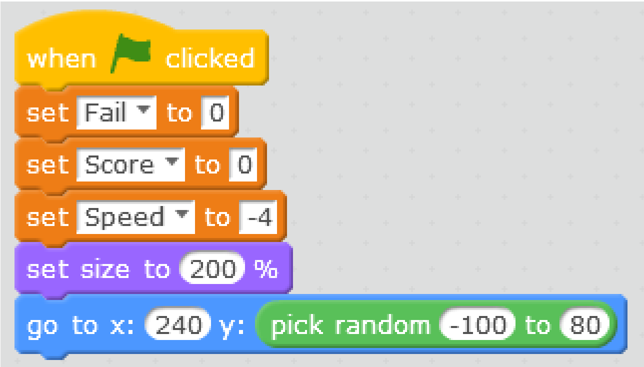


FIG. 8-22

Note:

a) The initialized setting of the variable can be written in any sprite, and here it is written in "pipe."

b) Hide the variable of the "fail," "speed" and the "elephant y-axis" and only remain the scores variable on the stage.

c) Because the "pipe" moves from the right to the left, that is, the value of x-axis decreases, then the value of speed variable will be negative.

d) The variable of "fail" is used to express whether the elephant has touched and

failed the game. If the value is 0, it means go on play. If the value is 1, it means failed.
e) The range " -100~80" of the y-axis of the "pipe" is the lowest position and the highest position of the "pipe."

b. Write the motion program of the "pipe": The "pipe" will begin to move when receiving the broadcast. Before the fail of the "elephant," the pipe will move to the left at the speed of "-4". When it moves to the leftmost side, it will appear in the rightmost side and move again, and the scores will increase. The program is as shown in FIG. 8-23.

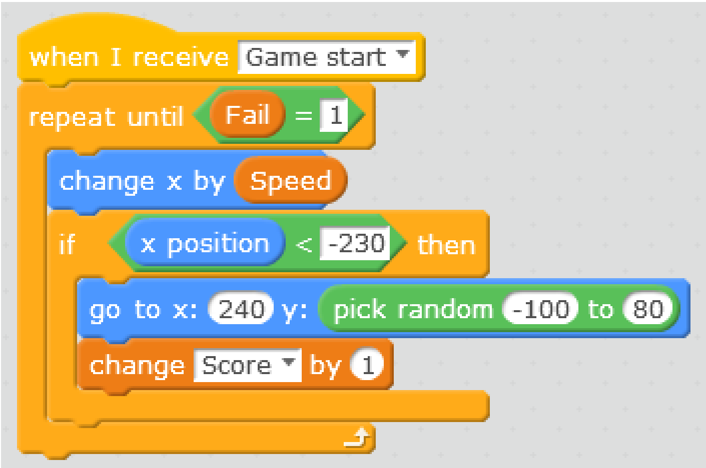


FIG. 8-23

Note: The x-axis cannot be "-240", as the coordinate of the sprite is in the center of the role and the "pipe" has a width. If the edge of the sprite touches the edge of the stage, the sprite will stop moving and it can never move to "-240".

c. Write the fail animation of the "elephant": The "elephant" shall stop moving when touching the "pipe" and all the program shall stop. Meanwhile, only when the variable value of "fail" is 0, the "elephant" can move with the detection value of the RGB ultrasonic sensor. The modified "elephant" program is as shown in FIG. 8-24.

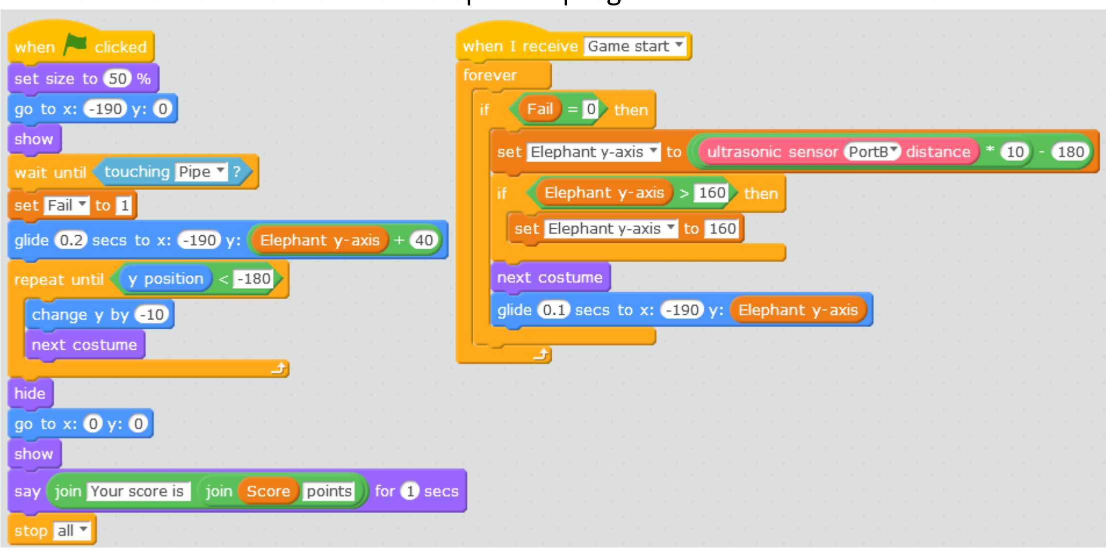


FIG. 8-24

Program effect: Run the program, the "elephant" will move up and down with the detection value of the RGB ultrasonic sensor, and the score will increase by 1 every time it escapes a pipe. After touching the pipe, the "elephant" will stop moving and hide, the " pipe" will stop moving and the program will stop.

After completing the above procedures, the game has been completed. The total program of the "elephant" is as shown in FIG. 8-24. The total program of the "start button" is as shown in FIG. 8-19. The total program of the " pipe" is as shown in FIG. 8-25.

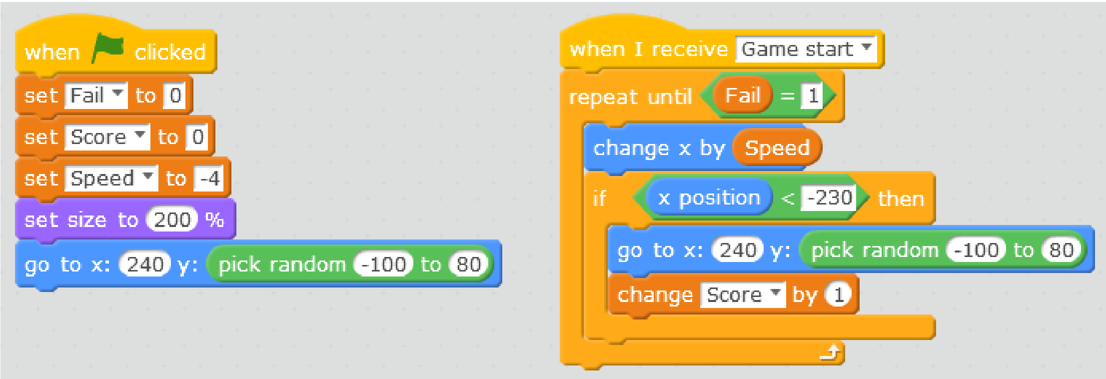


FIG. 8-25

Note: The above program is only for reference. If students can finish the game by other means, teachers can give an award to them.

IV Classroom Development and Classroom Evaluation (10min)

1. Classroom development: Make the animation effect of fail, add acceleration function of the "pipe" and enhance the game experience (selected part according to the time of class).

a. Make the fail animation of the "elephant": After the "elephant" touches the "water pipe," the "water pipe" will stop moving. Moreover, the "elephant" will move upwards first and then move downwards until it drops to the bottom and hides. Then it will be displayed in the center of the stage and say the scores. The modified "elephant" program is as shown in FIG. 8-26.

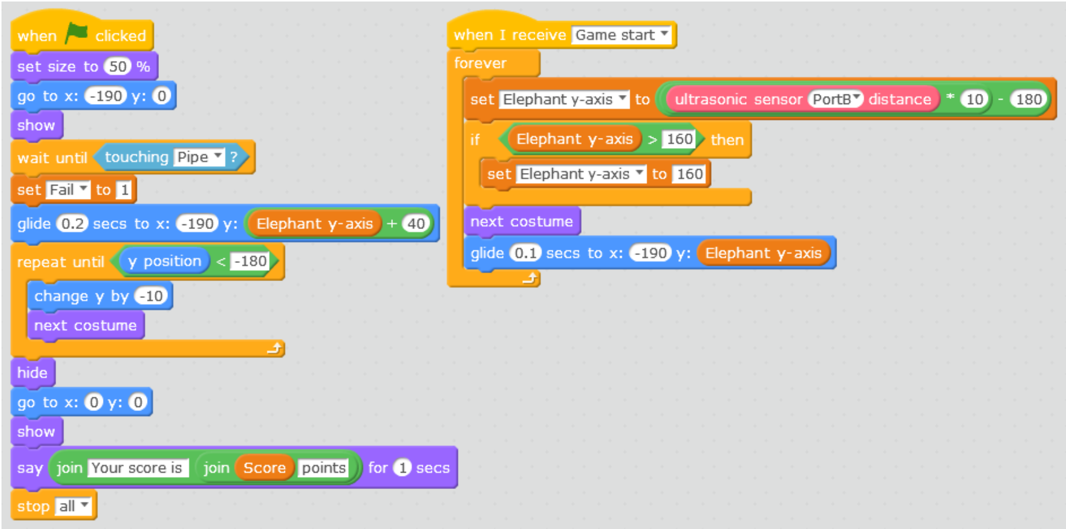


FIG. 8-26

Note: join block is in the operator module.

b. Make the acceleration effect of the "pipe": Make the motion speed of the "pipe" be higher and higher with the higher scores in the game. Add the difficulty of the games and the modified "pipe" program is as shown in FIG. 8-27.

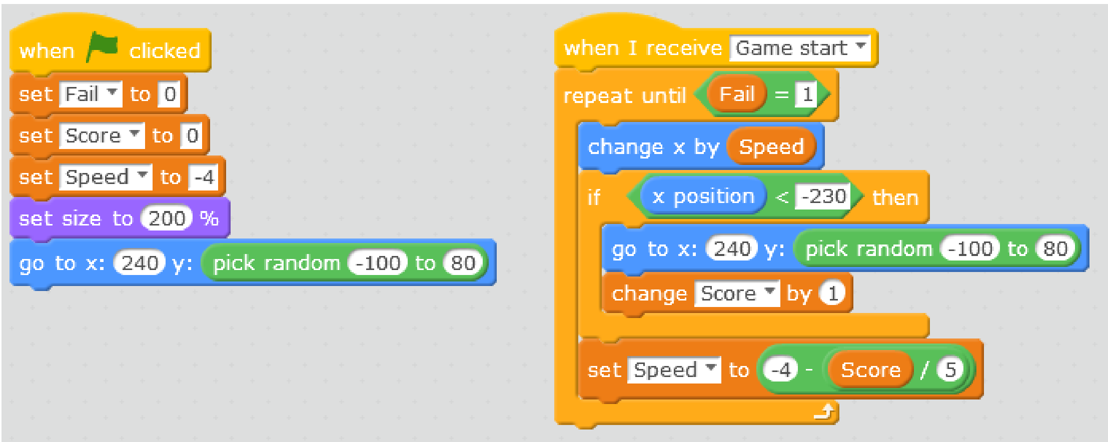


FIG. 8-27

Note: The setting of the "speed" variable value will only be added at the end of the loop structure in the "pipe" program, that is, every 5 points score will increase the speed by 1.

2. Classroom evaluation: Students' classroom performance will be evaluated according to the evaluation criteria.

Evaluation criteria:

a. Able to skillfully use the broadcast, modeling transformation, random number, and "wait until" blocks.

b. Able to make flying elephant games.

Chapter 9 Little Pianist

Summary

As the review and conclusion of the above knowledge points, this lesson will teach students to make ultrasonic piano through the combination of the RGB ultrasonic sensor, buzzer and RGB LED and let them experience the beauty of the combination of programming and arts.

Learning Objectives

Review the knowledge points learned above, strengthen the mastery of the old knowledge points, and make an ultrasonic piano.

Master the usage of the ruler and intuitively understand the length of distance.

Cultivate students' teamwork ability through the form of division of work.

Course Time

90 minutes.

Pre-class Preparation

WeeeBot robots, computers with WeeeCode software installed, USB cable, ruler, color pens, and papers.

Teaching Process

I Knowledge review (15mim)

1. Review the control blocks in learned modules and strengthen the mastery degree of module control.

Q: How many modules control method have we mastered in the previous course of learning? What are they respectively? Which blocks are they controlled by? What should we pay attention to when using?

Conclusion: as shown in FIG. 9-1.

Hardware name	Control blocks and annotations (Numbers represent the number of parameters in the block)	Attention when using
RGB LED	RGB on board red 0 green 0 blue 0 : ①~③ adjust the value of each color of the RGB.	① The RGB value can be known in "MSPaint" software. ② There shall be time delay among the color changes.

Buzzer	: ①Control the tone of the sound. ②Control the beat of the sound.	①C~B corresponds with the do~si tone, and 3~5 co-responds with the BASS, MIDDLE, or TREBLE three musical scales. ②The relationship between rhythm number of numbered musical notation and the corresponding block beat number of a beat in the numbered musical notation: 30- double beat, 60- a whole beat, 120- 1/2 beats, 240- 1/4 beats, 480- 1/8 beats.
DC motor	1. : ① To select the port of the motor. ② To control the running speed of the motor and the range is -255~255. If the value is positive, it will run clockwise. If the value is negative, it will run anti-clockwise. The larger is the number is, the higher the speed will be. 2. : ①To control to move in four directions. ②To control the running speed of the motor and the range is -255~255. If the value is positive, it will run clockwise. If the value is negative, it will run anti-clockwise. The larger is the number is, the higher the speed will be.	① Port: Right M1 and left M2; Motor wires: Left black and right white. ② It will need to turn on the power supply switch if in the online mode. ③ the motor will keep the previous motion state before changing the motion state.
IR remote control	:	The telephone camera can be used to test

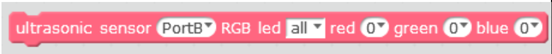
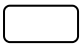
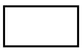
	① To select the wiring port of the infrared receiver module. ② To select the key pressed on the remote control.	whether the remote can emit infrared signals or not.
RGB ultrasonic sensor	1.  ①To select the port of the RGB ultrasonic sensor. The default port is B. The distance value between ultrasonic wave and obstacles can be got in this block, and its unit is cm. 2.  ①Same as above。 ② To select the light to be controlled. ③~⑤ Adjust the value of each color of the RGB.	The distance measurement range of the RGB ultrasonic sensor is 4~500cm.

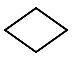
FIG. 9-1

2. Review the drawing methods of the flowchart.

Program flowchart is a description of the method, idea or algorithm for solving problems. It uses a graphical symbol box to represent the operations of various properties and connects these operations with a flow line. Before the program is designed or problem is solved, it can help us clarify the solution of complex issues through drawing a flowchart.

a.  : The starting box, represents the beginning or end of the program.

b.  : The processing box, represents the processing method.

c.  : The judgment box, represents the judgment and the branch.

d.  : The flow line, represents the execution order of the flowchart.

Ⅱ Assignment and work division of the roles (15min)

1. Assignment

Task requirements:

a. It requires students to use the buzzer to play different songs according to the different sounds made through the different distance between the obstacle (or hand)

and RGB ultrasonic sensor. Each tone corresponds with a color of the RGB light and when the sound changes, the color will change accordingly.
b. Draw the flowchart to clarify the programming ideas and write the program.
c. Use a ruler to draw the piano key area diagram on the paper: the tones and colors in different key areas will be different.
d. Write programs and play songs according to the selected numbered musical notation.

2. Discuss in the group according to the task requirements, divide the work combined with the situation of the members, and arrange the time assignment of each link reasonably.
Work assignment among members: Students will be assigned with work according to their advantages and please fill the form as shown in FIG. 9-2 for reference.

Members	Responsible content	Reason	Estimated time	Actual time
Abby	Draw flowchart	Logical and careful.	10min	
Brian	Write programs	High mastery in hardware module, and strong logic ability.	20min	
Carrie	Play songs	Strong sense of rhythm.	20min	
David	Draw key area diagram	Solid art foundation.	20min	

FIG. 9-2

Note:
a. Suggest 4~5 students as a group.
b. Besides guiding students to divide works in the activity process, it also needs to guide students to arrange time reasonably. If possible, please draw a flowchart and write the program. After the program debugging, exercise the songs playing and draw the key area diagram.
c. Actual time will be filled in after the task is completed.

Ⅲ Design scheme and prototype creation (50min)

1. Draw the flowchart of the ultrasonic piano program
You can draw the program flowchart as shown in FIG. 9-3 according to the task requirements.

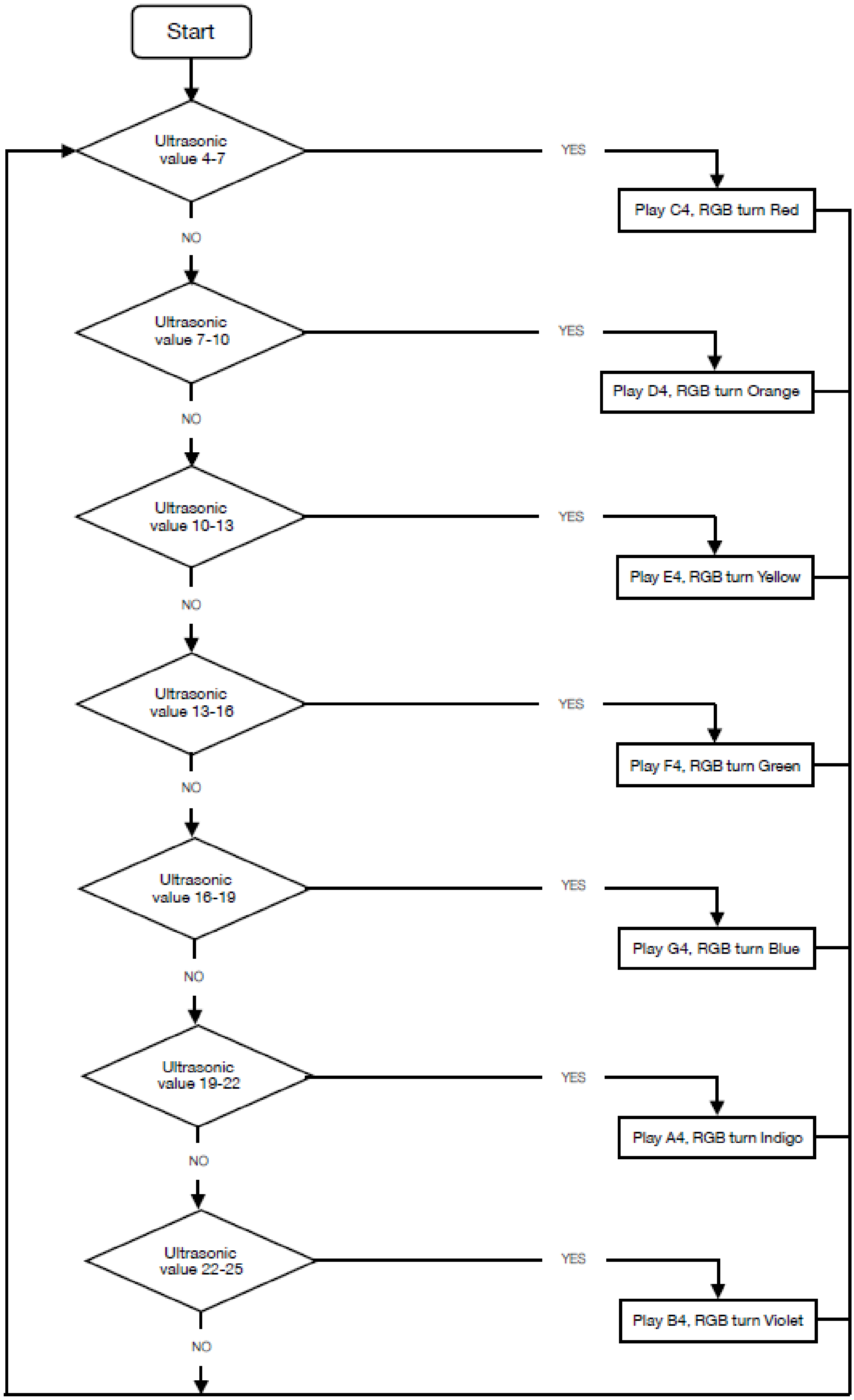

Start
Ultrasonic value 4-7
YES
Play C4, RGB turn Red
NO
Ultrasonic value 7-10
YES
Play D4, RGB turn Orange
NO
Ultrasonic value 10-13
YES
Play E4, RGB turn Yellow
NO
Ultrasonic value 13-16
YES
Play F4, RGB turn Green
NO
Ultrasonic value 16-19
YES
Play G4, RGB turn Blue
NO
Ultrasonic value 19-22
YES
Play A4, RGB turn Indigo
NO
Ultrasonic value 22-25
YES
Play B4, RGB turn Violet
NO

FIG. 9-3

Note:

a. The distance value interval of this program flowchart is 3 and students can define the interval range according to their requirements.
b. You can modify the corresponding sound and color of each interval.
c. There is only seven key area drawn in this program flowchart. If the music scale range of the played songs is too broad, the range can be enlarged by yourselves.

2. **Write an ultrasonic piano program**

You can write the program as shown in FIG. 9-4 according to the program flowchart.

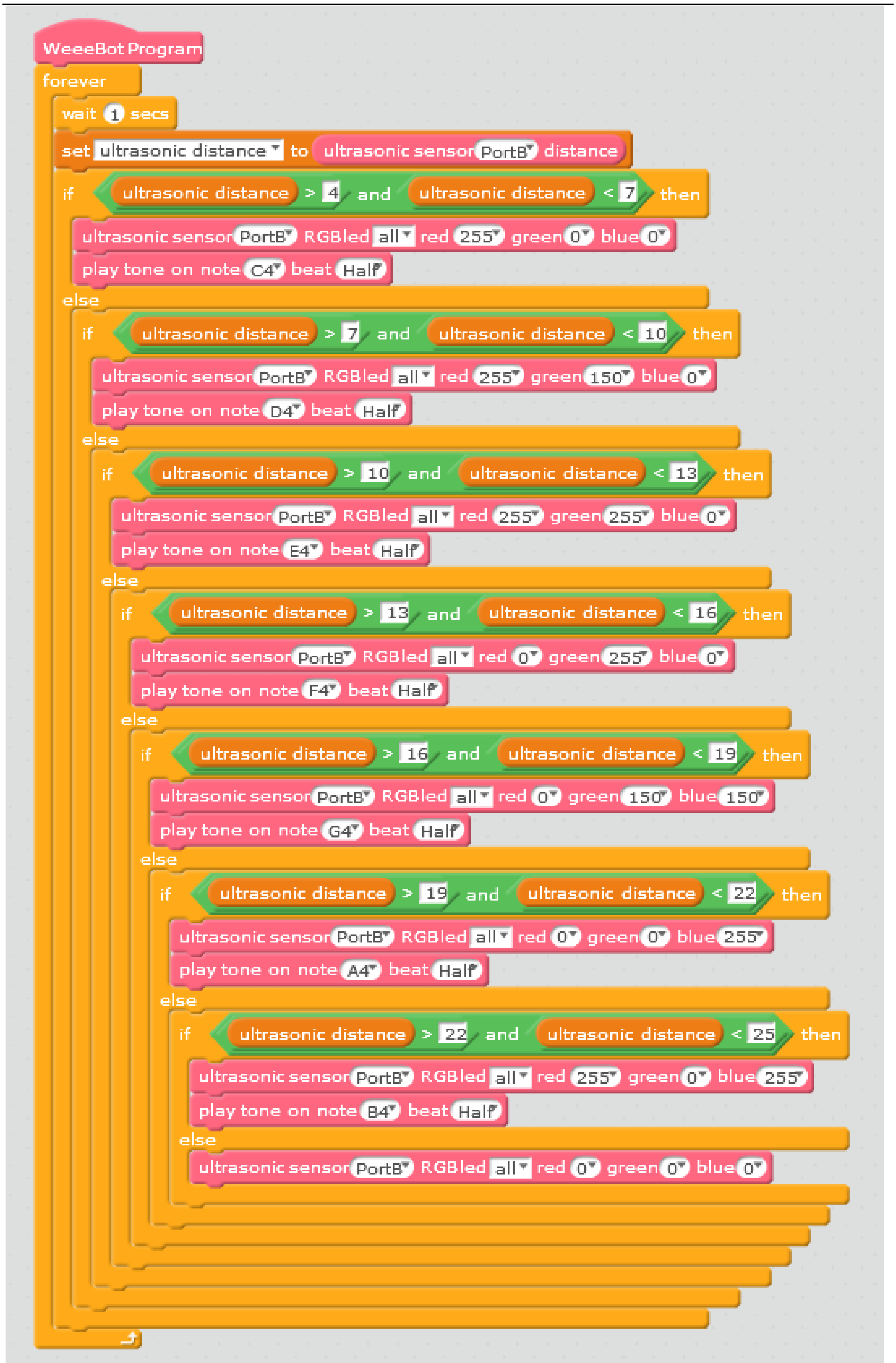

WeeeBot Program
forever
wait 1 secs
set ultrasonic distance to ultrasonic sensor PortB distance
if ultrasonic distance > 4 and ultrasonic distance < 7 then
ultrasonic sensor PortB RGBled all red 255 green 0 blue 0
play tone on note C4 beat Half
else
if ultrasonic distance > 7 and ultrasonic distance < 10 then
ultrasonic sensor PortB RGBled all red 255 green 150 blue 0
play tone on note D4 beat Half
else
if ultrasonic distance > 10 and ultrasonic distance < 13 then
ultrasonic sensor PortB RGBled all red 255 green 255 blue 0
play tone on note E4 beat Half
else
if ultrasonic distance > 13 and ultrasonic distance < 16 then
ultrasonic sensor PortB RGBled all red 0 green 255 blue 0
play tone on note F4 beat Half
else
if ultrasonic distance > 16 and ultrasonic distance < 19 then
ultrasonic sensor PortB RGBled all red 0 green 150 blue 150
play tone on note G4 beat Half
else
if ultrasonic distance > 19 and ultrasonic distance < 22 then
ultrasonic sensor PortB RGBled all red 0 green 0 blue 255
play tone on note A4 beat Half
else
if ultrasonic distance > 22 and ultrasonic distance < 25 then
ultrasonic sensor PortB RGBled all red 255 green 0 blue 255
play tone on note B4 beat Half
else
ultrasonic sensor PortB RGBled all red 0 green 0 blue 0

FIG. 9-4

Note:

a. The beat parameter value of the buzzer control block is not very important in this program. It will be ok only if the sound is delayed for a period.
b. At the beginning of the program, it is necessary to add the "wait 1 secs" block, which is to judge the distance between the RGB ultrasonic sensor and the hand every 1 second. If there is no such building block, the RGB ultrasonic sensor will continuously detect and continuously trigger judgment, which may result in incomplete pronunciation.

3. Draw key area diagram

Draw key area diagram as shown in FIG. 9-5 according to the program.

FIG. 9-5

Note:
a. FIG. 9-5 is only for reference. Students can design by themselves, but there must be snap line and distance range interval.
b. The measure line is the alignment line between the RGB ultrasonic sensor and the drawing, and its placement is as shown in figure 9-6.

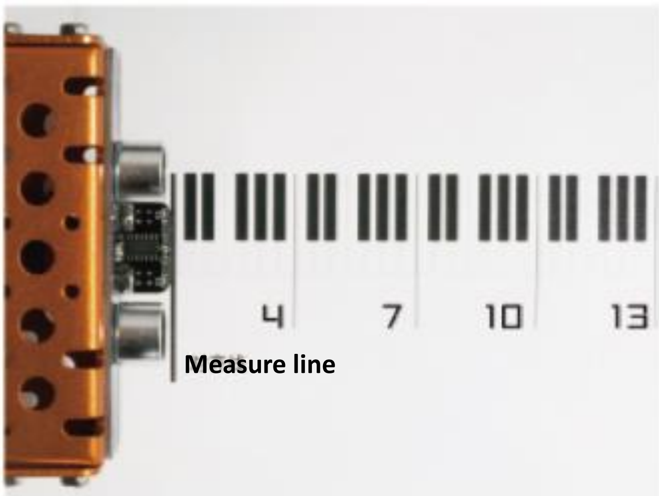


FIG. 9-6

4. Song performance practice

Students can practice according to the selected songs. During the practice, you need to put an object on the corresponding key area of the first tone, as shown in FIG.9-7. After the buzzer makes a sound, please lift the object and put it on the corresponding key area of the next tone for detection.

Note: Teachers can select a few simple songs in advance for students to choose from. The involved musical scale range of selected songs shall be as small as possible.

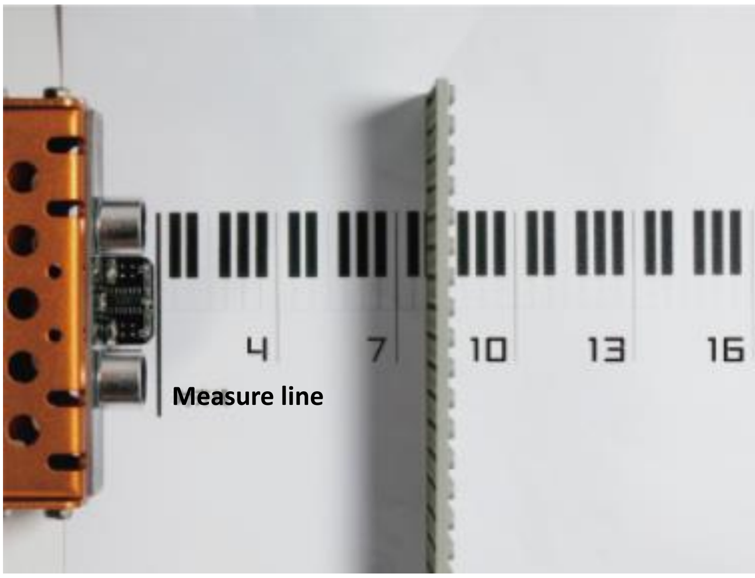


FIG. 9-7

IV Classroom Presentation and Classroom Evaluation (15min)

1. Classroom Presentation

a. The groups play songs to each other and score the performance effects of the songs.

b. Teachers select the group randomly, and the selected group shares the work division of members and time allocation within the group and plays the songs. Teachers may award them appropriately if the group performs well.

2. Classroom evaluation: Students' classroom performance will be evaluated according to the evaluation criteria.

Evaluation criteria:

Routine	Evaluation regulations
Flowchart	1. There are no mistakes in the shape of basic flowchart types. 2. The flowchart can clearly show the logic of the program.
Program effect	1. Able to play songs with RGB ultrasonic sensors. 2. The color changes of RGB lights accompany the performance of the song.
Key area diagram	1. The design of the vital area diagram is elegant, which can show the critical area clearly. 2. The design of the vital area diagram has unique ideas and exquisite appearance, and it is original.

Work division of members and time allocation	The work division of members has a reasonable plan, which ensures the task requirement can be finished in the estimated time.
Playing effect	The performance of the song is relatively more coherent and skilled (Teachers can score points properly according to the difficulty of the song).

Chapter 10 The secret of Sound and Light

Summary

In this lesson, we will make the robot obtain two new way to sense the environment through learning sound sensor and light sensor and let students feel the intelligence of controlling programmatically through making intelligent corridor lamps.

Learning Objectives

To master the usage of the sound sensor and light sensor.

To understand the control principle of intelligent corridor lamp and make it.

Course Time

90 minutes.

Pre-class Preparation

WeeeBot robots, computers with WeeeCode software installed, and USB cable.

Teaching Process

I Introduction from Life (20min)

1. Share the work conditions of corridor lamps in life and conclude its work principles.

The work conditions of corridor lamps are as shown in FIG. 10-1.

Light	Sound	The conditions of lights
Strong	Low	Closed
Strong	Loud	Closed
Weak	Low	Closed
Weak	Loud	Open

FIG. 10-1

The working principles of the corridor lamp: The acoustic control and optical “Control” category are installed in the corridor lamp with the automatic delay switch. The light is bright during the day, and the light will be off. At night, the light is weak, and when the sound sensor detects a big value, the light will be on.

2. The application of sound sensor and the light sensor in life

The sound sensor applied in life plays a role of a switch, such as sound control washing machine---control the running and stopping of the washing machine by acoustic control; acoustic control camera---control the shutter time of the camera by acoustic control.

The light sensor plays a role of saving energy in life, such as smart keyboard---the keyboard light will be on in the daytime and will be off at night; smart screen---adjust the screen luminance intelligently according to the light intensity.

Ⅱ Know about modules and use it

1. Master the usage of sound sensor

The sound sensor is located in the position of the mainboard as shown in FIG. 10-2. We can find the control block of the sound sensor in the “Robot” category, as shown in FIG. 10-3. The sound value in robot's environment can be concluded by using the block.

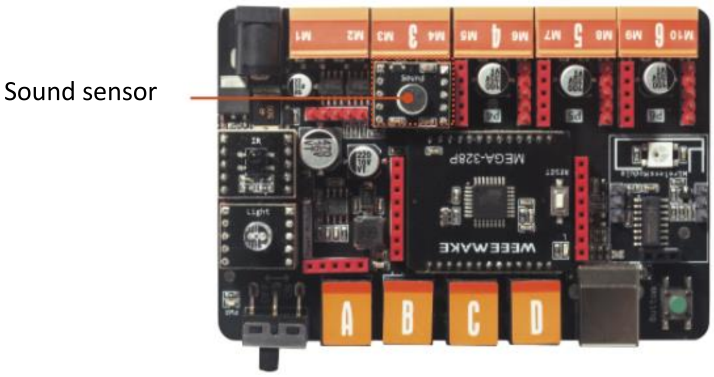


FIG. 10-2

FIG. 10-3

Practice:

Select “M-Elephant” role and write the program in the role as shown in FIG. 10-4. Then run it in the online mode and the role will say the sound value of the environment.

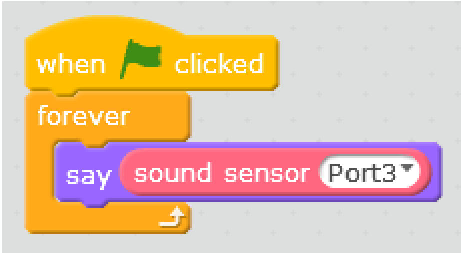


FIG. 10-4

2. Master the usage of the light sensor

The light sensor is located in the position of the mainboard as shown in FIG. 10-4. We can find the control block of the light sensor in the "Robot" category, as shown in FIG. 10-5. The light value in robot's environment can be concluded by using the block.

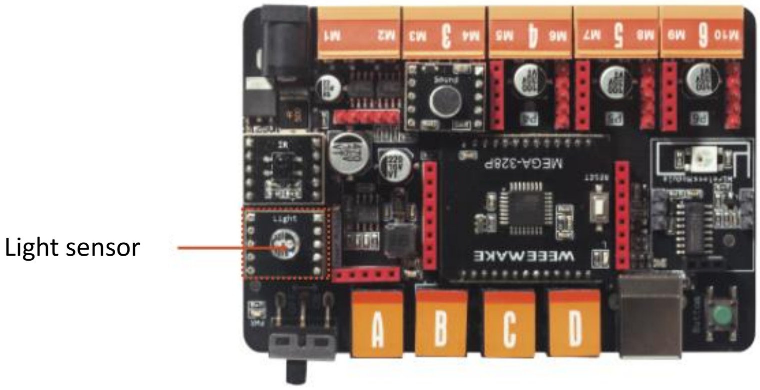


FIG. 10-5

Select the port of the light sensor. The default port is port 1. If the wiring position has been changed, the port shall also be modified accordingly.

FIG. 10-

Practice:

Select "M-Elephant" role and write the program in the role as shown in FIG. 10-7. Then run it in the online mode and the role will say the light value of the environment.

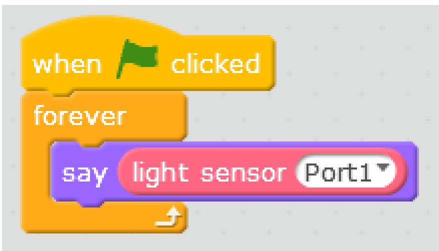


FIG. 10-7

Ⅲ Programming (30min)

1. Analyze programming ideas.

Q: According to the form as shown in FIG. 10-1, if we want to make a smart corridor lamp, how to write the program?

Conclusion: It can be known from the figure that the intensity of the light is the first condition that determines the opening and closing of the light, followed by the level of the sound, which can be concluded that the judgment of the light sensor should be preferable to the judgment of the sound sensor.

2. Make smart corridor lamp

The small value of the light sensor represent night; otherwise, it represents daytime. The small reading of the sound sensor represents no person; otherwise, it represents the presence of the person. The reference program is as shown in FIG. 10-8.

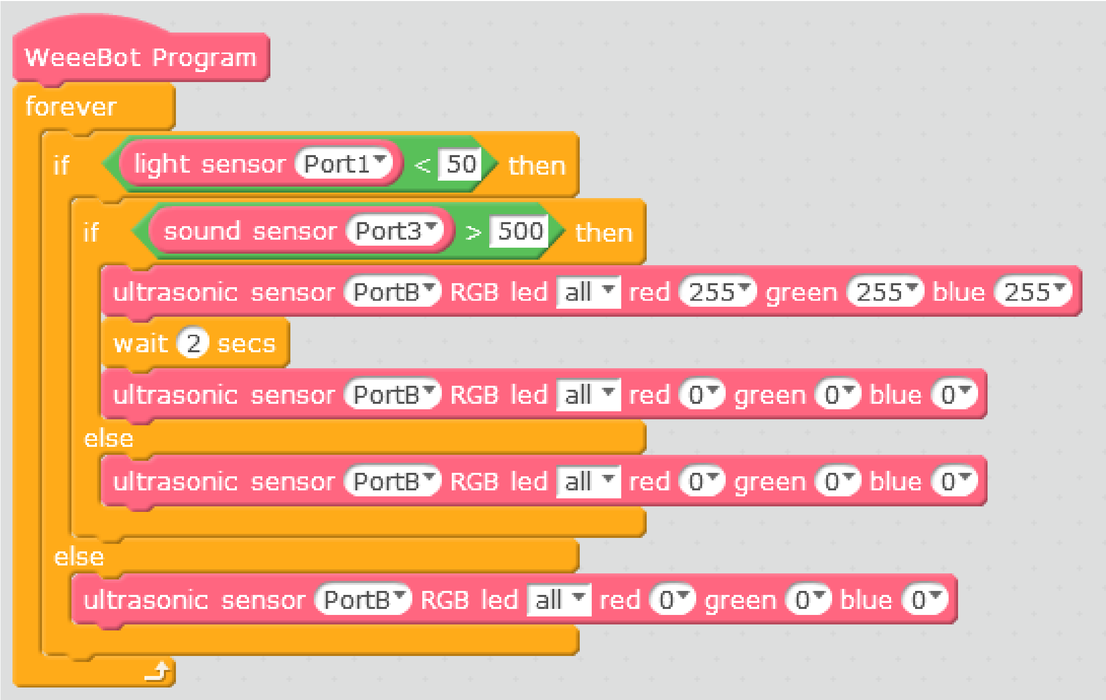


FIG. 10-8

Note: The value of the judgment condition shall be set according to the environment. If the light value in the daytime is 800 and the light value at night is 20, the judgment value of the light sensor can be set to 50-700. If the sound value in the typical environment is 300 and the sound value of clapping is 700, the judgment value of the sound sensor can be set to 400-600.

Ⅳ Classroom Development and Classroom Evaluation (10min)

1. Classroom development: Use the light sensor or sound sensor to control the robots (selected part according to the class time).

a. Acoustic control robot: When the sound is loud, the robot advances in the distance and the sound is low, the robot will be at rest. The program is as shown in FIG. 10-9.

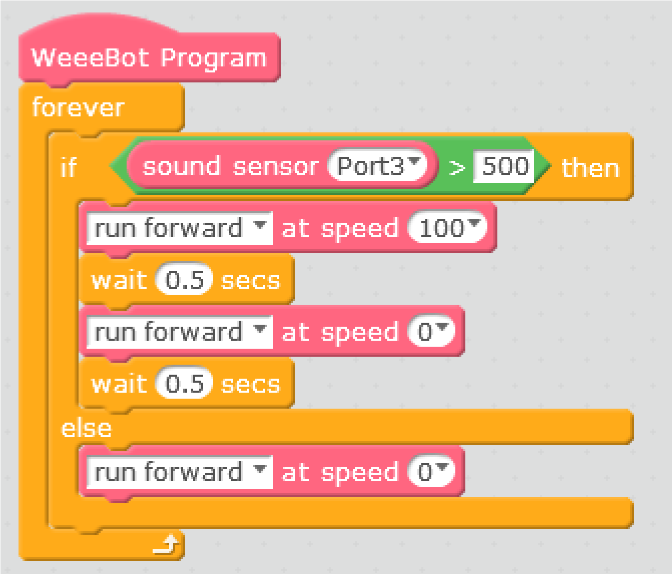


FIG. 10-9

Note: Wait for a period after stilling, and then judge the detection value of the sound sensor, to eliminate the interference of the sound when walking.

b. Light control robot: The robot will move forward when the light is weak, and the robot will be still when the light is bright. The program is as shown in FIG. 10-10.

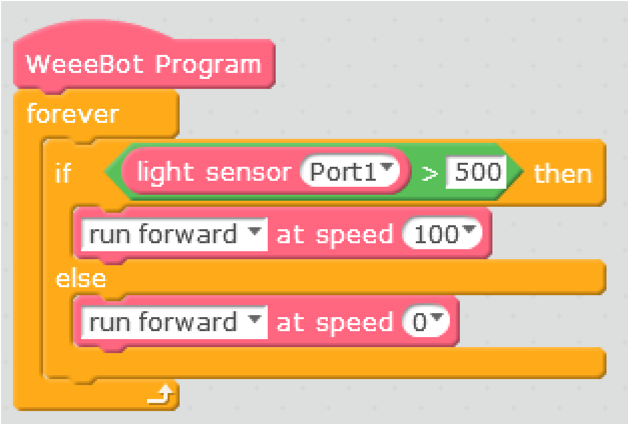


FIG. 10-10

2. Classroom evaluation: students' class performance will be evaluated according to the evaluation criteria.

Evaluation criteria:

a. Proficient in using the control blocks of sound sensors and light sensors.

b. Able to make smart corridor lamps with sound sensors and light sensors.

Chapter 11 Intelligent Robot

Summary
After we mastered the basic control methods of the sound sensors and light sensors in the last lesson, in this lesson, we will combine the two modules with the other modules we learned before to make an intelligent robot and let students feel the charm of programming.

Learning Objectives
Able to combine hardware knowledge learned previously to write programs to make intelligent robots.
Understand mind maps, draw mind maps of intelligent robots, and train students' logical ability through multilevel logical judgment.

Course Time
90 minutes.

Pre-class Preparation
WeeeBot robots, computers with WeeeCode software installed, USB cable and remote control.

Teaching Process

I Introduction from Life (15min)

1. Understand the functions of current high-tech cars.
a. Semi-automatic driving: You only need to set the walking speed, and the distance between your car and the car in front of you, and your car will drive automatically according to the set drive mode. In certain conditions, the system can manipulate the car's steering and braking autonomously.
b. Automatic parking: When the car reaches the parking space, it will start the automatic parking function, and the car will drive into the parking space automatically.
c. Automatic cruise: At a certain speed, the car will automatically move at a fixed speed without stepping on the gas pedal.
d. Anti-collision radar system: Real-time monitor objects in front of vehicles. When targets appear, the radar system will alert the driver so that the driver can respond in time.
e. Alcohol-test car lock system: It will detect if you have drunk alcohol through your palms sweat. If it detects that the alcohol concentration in your sweat is too high, the gearbox will be locked automatically, and there will be voice prompt, which can prevent the driver from drunk driving.
Note: The car uses much high-tech technology, and teachers can select the automation

control technology as the main lecture.

2. Imagine the function of future cars.

Automatic drive, wireless charging, sea and air walking, automatic emergency stop, stealth, capable of firing shells, and so on.

Ⅱ Design scheme and prototype creation (65min)

1. Understand mind maps.

A mind map is a reified method of radioactive thinking. We know that radioactive thinking is a natural way of thinking of the human brain. Each kind of information into the brain, whether it is a feeling, memory, or idea, all can become a thinking center, which radiates outward thousands of nodes, and each node represents a link with a central theme, and each link can be another central theme, and then emanate outward thousands of nodes. Take FIG. 11-1 as an example to guide students to think radioactively and draw mind maps.

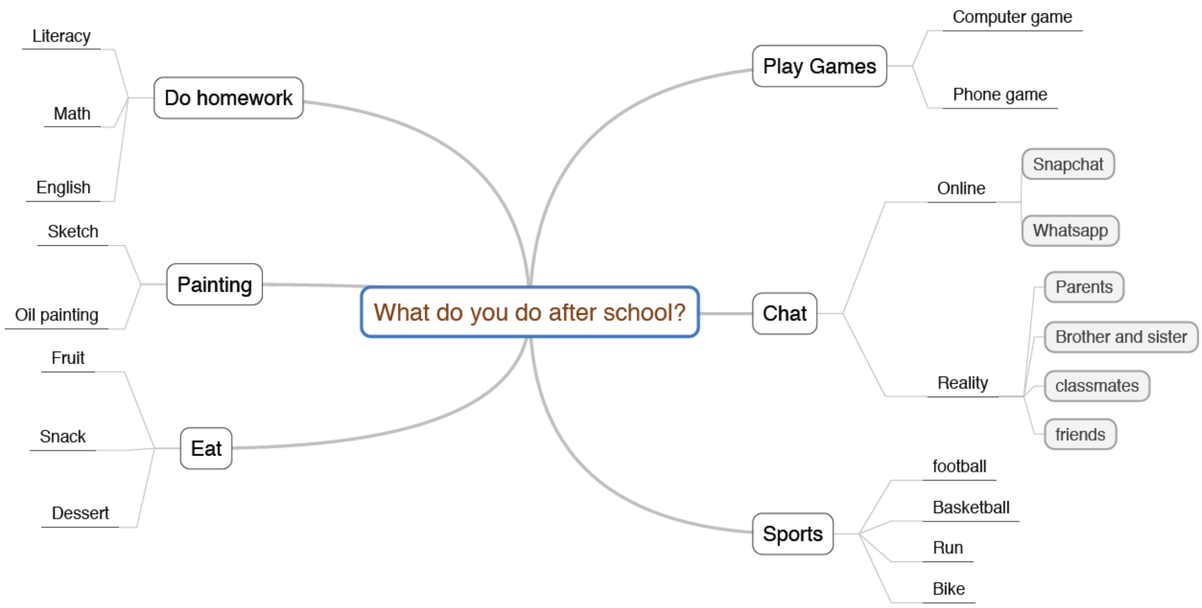


FIG. 11-1

2. Combine the hardware we learned previously to draw the mind maps of the intelligent robots.

What kind of intelligent robot can we make combined with the robot hardware control we learned currently? Take FIG. 11-2 as an example to draw the mind maps of the intelligent robot.

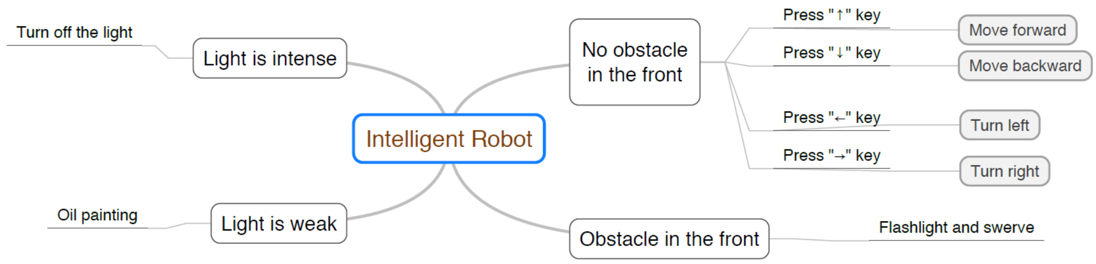


FIG. 11-2

a. The robot can turn on/off the light in the daytime and at night automatically.
b. The robot will detect whether there are obstacles in the front continuously in the proceeding process. If no obstacle, it will walk definitely according to the remote control. If there are obstacles, the robot will twinkle lights and make a turn.

Note:

a) The mind map is only for reference. If the students have their ideas, freedom of ideas may be given to them. However, it must be combined with the hardware we learned previously as much as possible.

b) The flowchart can be understood as the node time flowchart that must be completed for work; A mind map can be understood as a work procedure diagram, which is a flowchart of work guidance.

3. Write intelligent robot program according to the mind map.

a. Write the first-layer frame program of the intelligent robot mind map: Execute different motions according to the measured distance value, and light intensity and the program is as shown in FIG. 11-3.

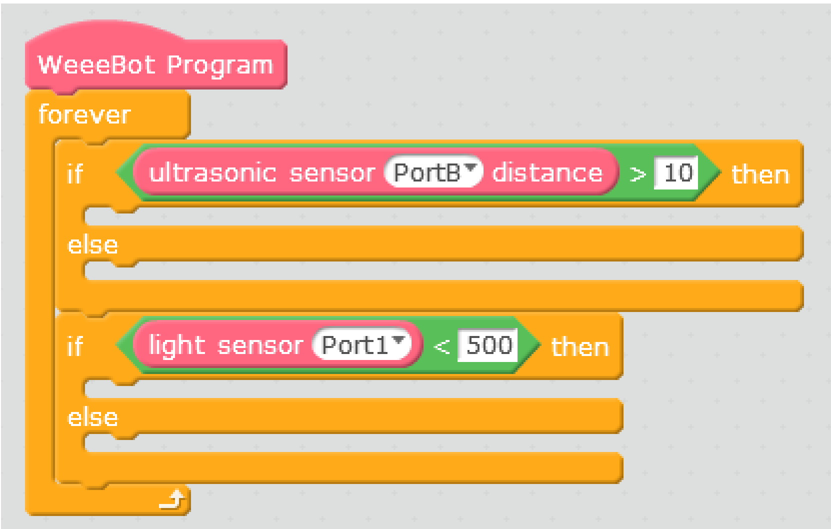


FIG. 11-3

b. Write the judgment program of light intensity: If the light is intense, the light will be off; otherwise, the light will be on. The program is as shown in FIG. 11-4.

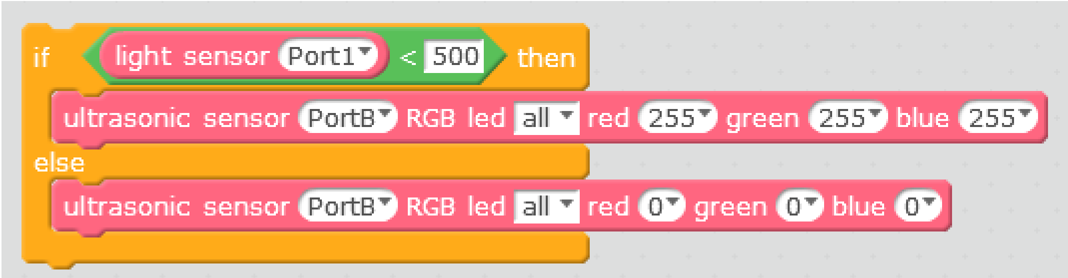


FIG. 11-4

c. Write the remote control program of robots when there is no obstacle in the font: Press "↑" key, the robot will move forward; Press "↓" key, the robot will move back; Press "←" key, the robot will turn left, Press "→" key, the robot will turn right, If no key is pressed, the robot will be still. The program is as shown in FIG. 11-5.

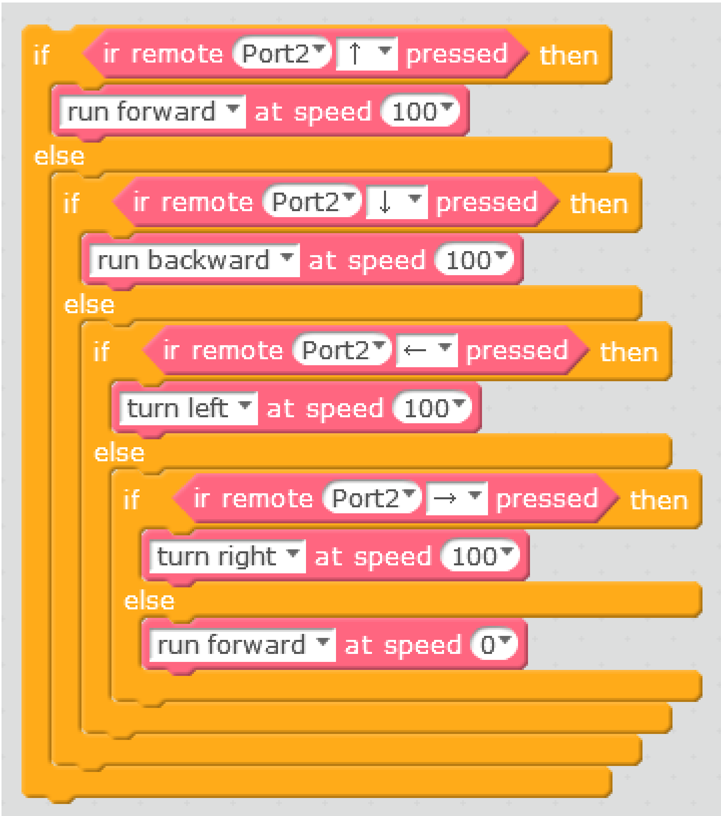


FIG. 11-5

d. Write the control program of the robot when there are obstacles in the front: At first, stop the running of the motor, and then flashlight and swerve. The program is as shown in FIG. 11-6.

FIG. 11-6

After the above procedures are completed, the total program will be as shown in FIG. 11-7. The programming is completed.

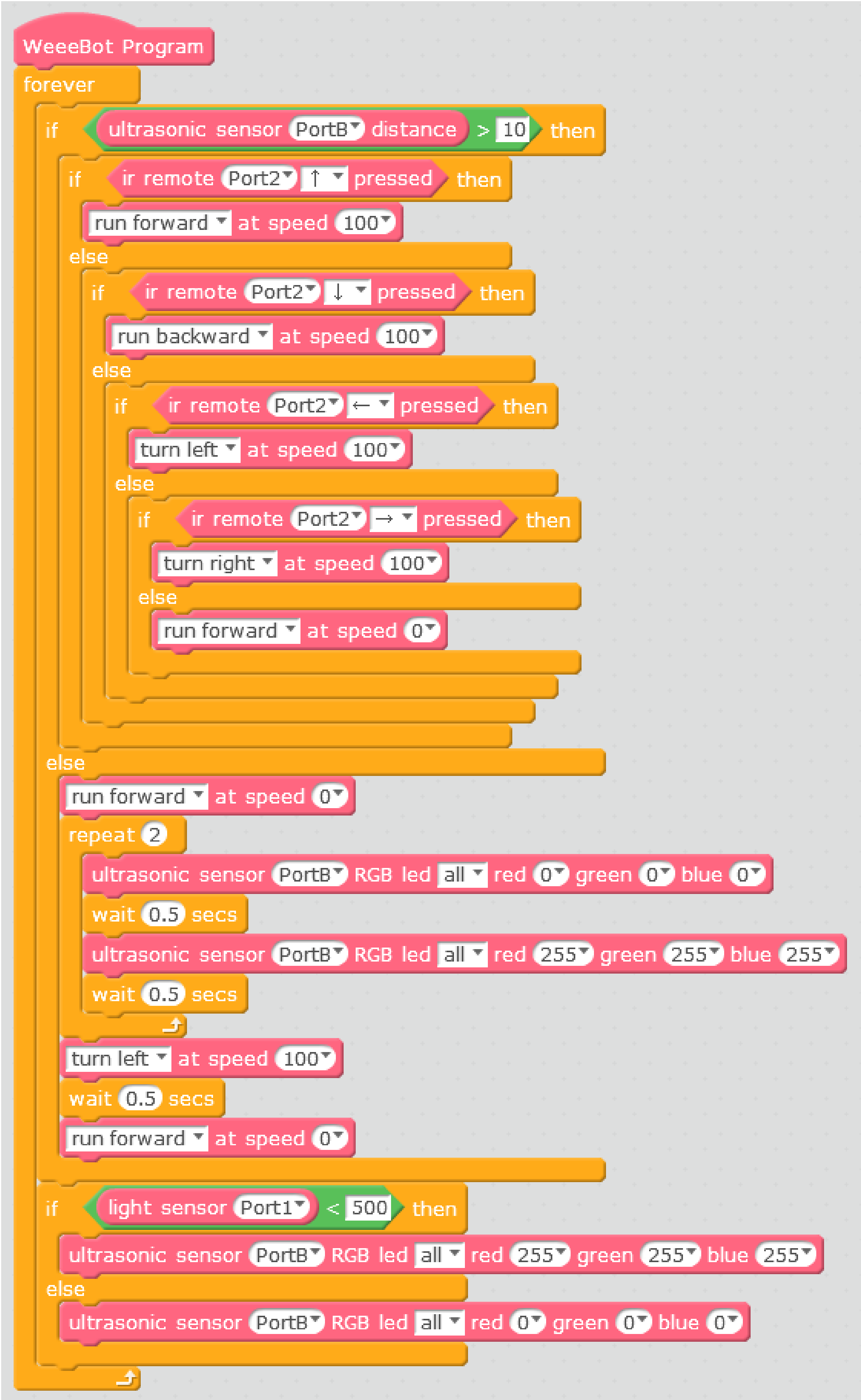

WeeeBot Program
forever
if ultrasonic sensor PortB distance > 10 then
if ir remote Port2 ↑ pressed then
run forward at speed 100
else
if ir remote Port2 ↓ pressed then
run backward at speed 100
else
if ir remote Port2 ← pressed then
turn left at speed 100
else
if ir remote Port2 → pressed then
turn right at speed 100
else
run forward at speed 0
else
run forward at speed 0
repeat 2
ultrasonic sensor PortB RGB led all red 0 green 0 blue 0
wait 0.5 secs
ultrasonic sensor PortB RGB led all red 255 green 255 blue 255
wait 0.5 secs
turn left at speed 100
wait 0.5 secs
run forward at speed 0
if light sensor Port1 < 500 then
ultrasonic sensor PortB RGB led all red 255 green 255 blue 255
else
ultrasonic sensor PortB RGB led all red 0 green 0 blue 0

FIG. 11-7

Ⅲ Classroom Development and Classroom Evaluation (10min)

1. Classroom development: volume walking robot (selected lecture according to the classroom teaching pace).

Volume moving robot: The sound triggers the robot to move. The walking distance will be determined by the size of the volume in the triggered condition. The louder the sound is, the farther the robot will move. The program is as shown in FIG.11-8.

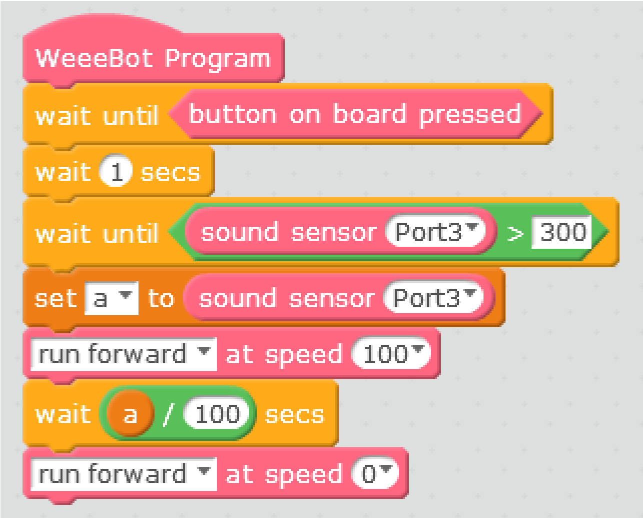


FIG. 11-8

Note:

a. There are two challenges in the program:

a) To eliminate the misjudgment through whether the machine talent starts to move when the sound sensor detection value is higher than a specified value.

b) Store the detection value of the sound sensor in the variable “a” and connect the value of “a” with the proceeding time of the robot. If the “a” value is 400, the robot will stop moving after proceeding 4 seconds.

b. The robot can be used for the competition of students, and teachers can use it as a classroom game.

2. Classroom evaluation: students' class performance will be evaluated according to the evaluation criteria.

Evaluation criteria:

a. Able to understand mind maps and draw the mind map for the intelligent robot.

b. Able to make an intelligent robot with the hardware knowledge learned previously.

Chapter 12 WeeeBot Line-following Robot

Summary

In this lesson, we will learn the use of the line-following sensor, figure out the line-following algorithm in the exploration, and programme to make the robot walk along the line.

Learning Objectives

To understand the working principle of the line-following sensor and master its usage.
To understand the line-following algorithms and be able to select and use them flexibly in different situations.
Able to make line-following robot according to the line-following algorithm.

Course Time

90 minutes.

Pre-class Preparation

WeeeBot robots, computers with WeeeCode software installed, USB cable and electrical tape.

Teaching Process

Ⅰ Introduction from Life (10min)

1. Understand the working principles of a food-delivery robot.

In this hi-tech age, even the dining room is equipped with robots. The robot delivers the food to the customer, slides through the dining room, stops at the table, and announces "your order, please enjoy it" in Mandarin.

The food-delivery robot walks along a fixed route through a magnetic strip sensing technology. When the robot reaches the designated table, it will stop until the customer takes the food away. In the process of walking, if the ultrasonic sensor detects someone in the way, the robot will also remind: "I am delivering the meal, please kindly avoid, thanks!"

Ⅱ Know about the Module and Use It (35min)

1. Know about the working principle of the line-following sensor.

The working principle of the line-following sensor is the same as that of the magnetic strip sensor. We can also make the robot walk along the line by the use of the line-following sensor. In this lesson, we will learn the usage of the line-following sensor.

The line-following sensor of the robot is as shown in FIG. 12-1, which consists of two line-following sensors--- s1 and s2.

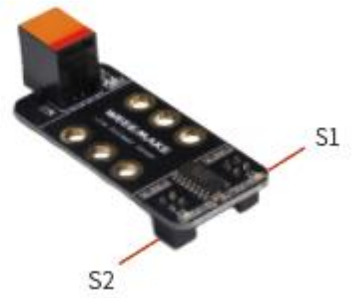


FIG. 12-1

A line-following sensor is an infrared sensor. We can see that the s1 and s2 sensor have two small lights respectively, one to transmit infrared ray and the other to receive infrared ray. So, the position of the sensor can be determined by the received quantity of the reflected infrared ray of the reflective surface.

Note: The infrared detection of the line-following sensor is different from that of IR remote control. The former can only be used to detect the quantity of the infrared ray, but the latter can be used to detect infrared signals.

According to the nature of the line-following sensor, most of the line-following sensors patrol along the black line. It is because that the black in colors can absorb a lot of infrared rays. If the line-following sensor is on the black line, it will receive less infrared rays; If the line-following sensor is not on the black line, it will receive more infrared rays. Branch structure can be used to judge the situation and make the robot walk along the line.

2. Master the usage of the line-following sensor.

The control block pf the line-following sensor can be found in the "Robot" category, as shown in FIG. 12-2.

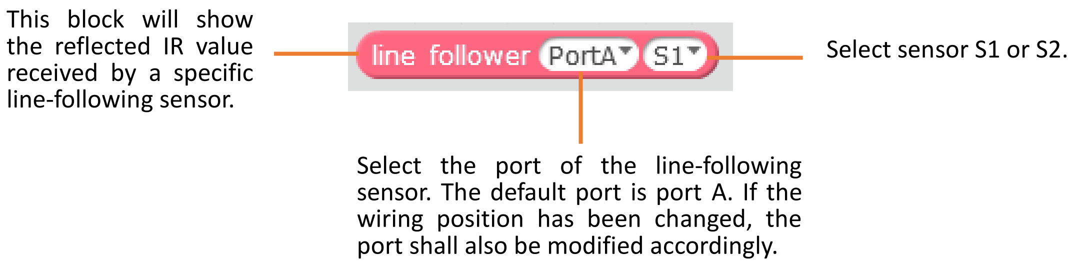


FIG. 12-2

Practice:

Select "M-Elephant" role, write the program in the role as shown in FIG. 12-3 and run it in the online mode. Observe the value changes of the line-following sensor on the different reflective surface.

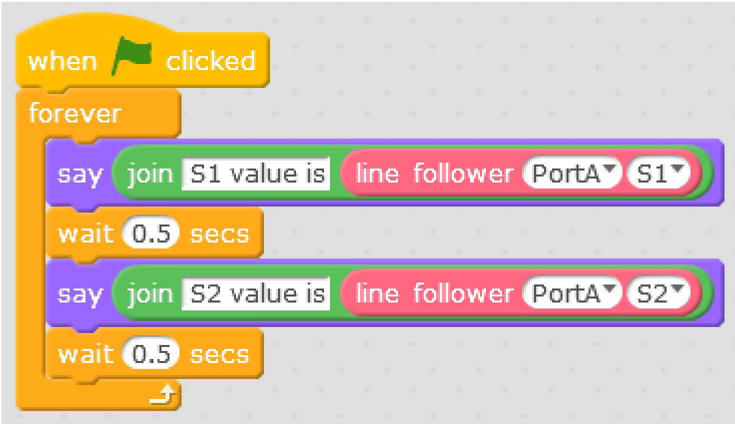


FIG. 12-3

Note:

a) If there is no reading, at first, please check whether it has entered online mode. If yes, please check whether the line-following sensor has to be connected to port A.

b) You can use a camera to observe whether the line-following sensor can work. Open the camera and view the line-following sensor. If you see that the little light under the line-following sensor emits purple-red light, the line-following sensor can work.

Exploration: Run the program as shown in FIG. 12-3, face the line-following sensor to the black line (electrical tape) and the white line (ordinary reflective surface), as shown in FIG. 12-4 and FIG. 12-5, and observe the changes in the reading.

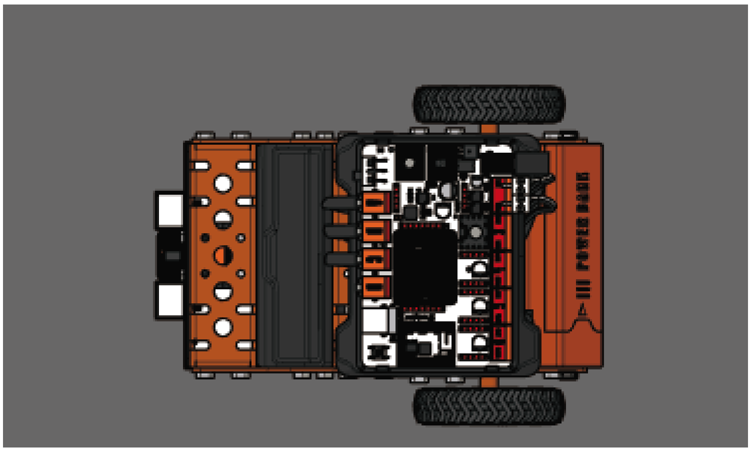


FIG. 12-4

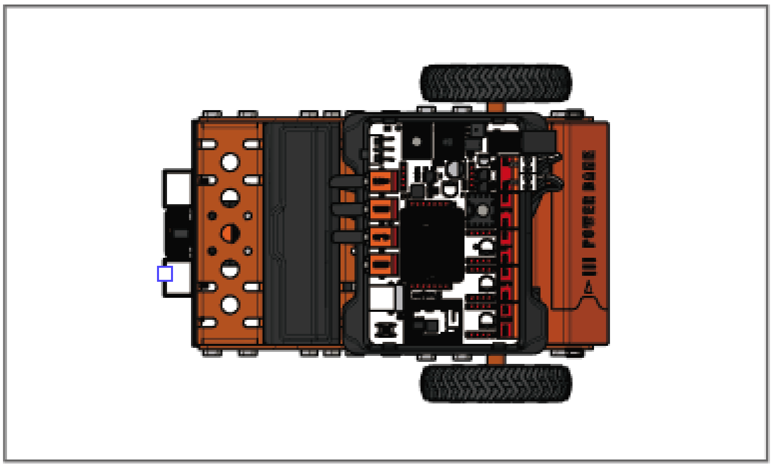


FIG. 12-5

Program effect: The reading on the black line is 70. The closer to the black line, the smaller the reading will be. The reading on the white line is 700. The closer to the white line, the larger the reading will be.

Conclusion: The more IR received by the line patrol sensor, the larger the value will be; Otherwise, the value will be smaller. Namely, the value on the white line is larger than the value on the black line.

Note: The above reading may fluctuate in a specific range based on different environment.

Ⅲ Programming (35min)

1. Master two kinds of line-following algorithms in the exploration and write the program according to the line-following algorithms.

There are two line-following methods in the line-following sensor:

a) The black line clamps the line-following sensor, as shown in FIG. 12-6.

b) The line-following clamps the black line, as shown in FIG. 12-7.

We will take the first line-following method as an example to explore the line-following algorithm.

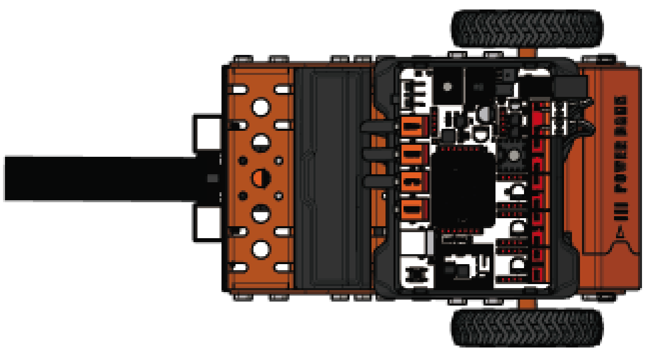


FIG. 12-6

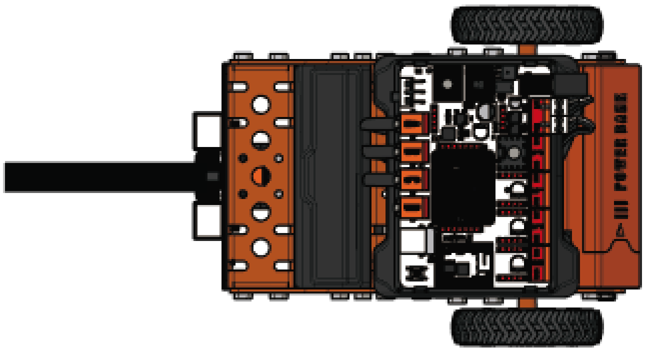


FIG. 12-7

Exploration: Observe the five cases which may occur in the line-following process as shown in FIG. 12-8. Think and fill in the form as shown in FIG. 12-9.

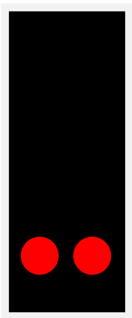
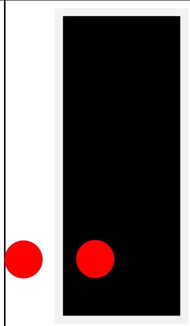
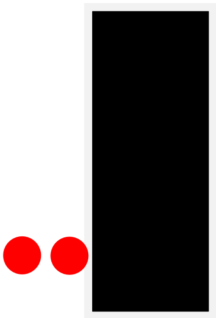
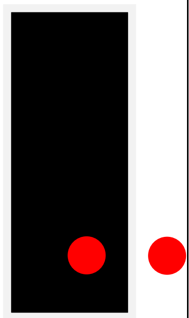
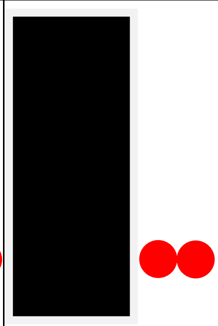
Serial NO.	1	2	3	4	5
Case					
The relations between the line-following sensor and the position of the black line	both S1 and S2 are on the black line.	S2 is on the black line but S1 not.	Neither S1 nor S2 is on the black line	S1 is on the black line but S2 not.	Neither S1 nor S2 is on the black line.

FIG. 12-8

Note:  represents line-following sensor and  represents the patrolled line when the robot car is walking.

	Is S1 on the black line?	Is S2 on the black line?	How shall the car move to make the two sensors both are on the black line?
1	Yes	Yes	Move forward
2	Yes	No	Turn left

3	No	Yes	Turn right
4	No	No	Still

FIG. 12-9

According to the laws explored in FIG. 12-9, we can store the detected value of the S1 and S2 sensor in the variable a and b, and then judge the size of the a and b value. As the value of the line-following sensor is 700 on the white line and 70 on the black line, we can select values within this range to judge (300 is selected in this example for judgment). If the value is less than 300, the line-following sensor will be on the black line. If larger than 300, the line-following sensor will not be on the black line. The program is as shown in FIG. 12-10.

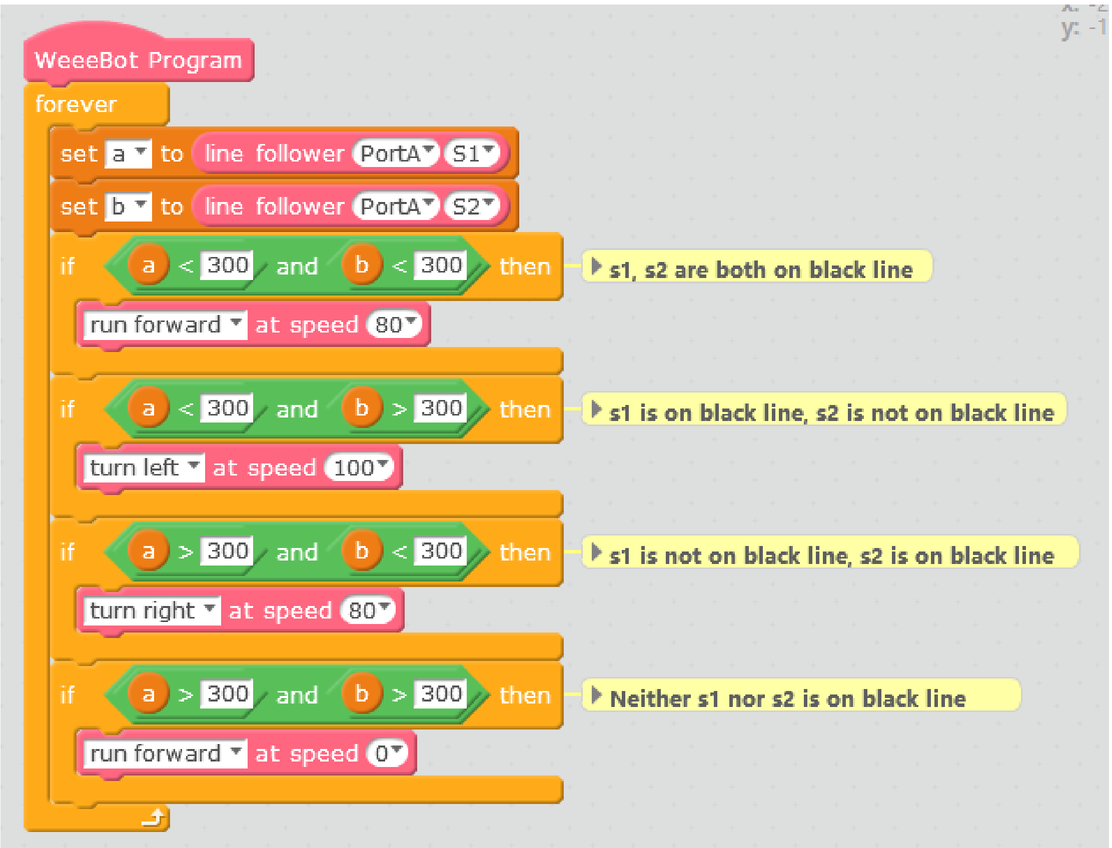


FIG. 12-10

Note: We need to adjust the speed of moving forward and turning according to the actual line-following situations.

Above is the line-following algorithm of "black line clamps line-following sensor." Next, we will explore the line-following algorithm of "line-following sensor clamps black line."

Exploration: Observe the five cases which may occur in the line-following process as shown in FIG. 12-11. Think and fill in the form as shown in FIG. 12-12.

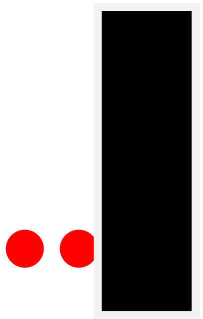
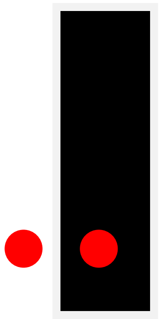
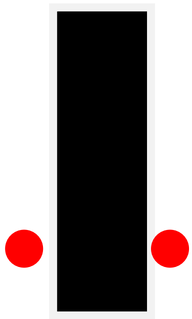
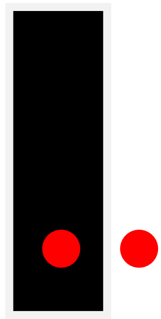
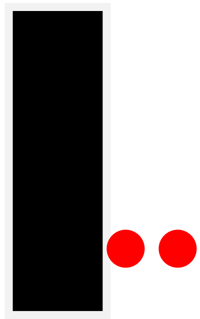
Serial NO.	1	2	3	4	5
Cases					
The relations between the line-following sensor and the position of the black line	Neither S1 nor S2 is on the black line	S2 is on the black line but S1 not.	Neither S1 nor S2 is on the black line	S1 is on the black line but S2 not.	Neither S1 nor S2 is on the black line.

FIG. 12-11

	Is S1 on the black line?	Is S2 on the black line?	How shall the car move to make the two sensors both are on the black line?
1	Yes	Yes	Still
2	Yes	No	Turn left
3	No	Yes	Turn right
4	No	No	Move forward

FIG. 12-12

We can write the program as shown in FIG. 12-13 according to the laws concluded in FIG. 12-12.

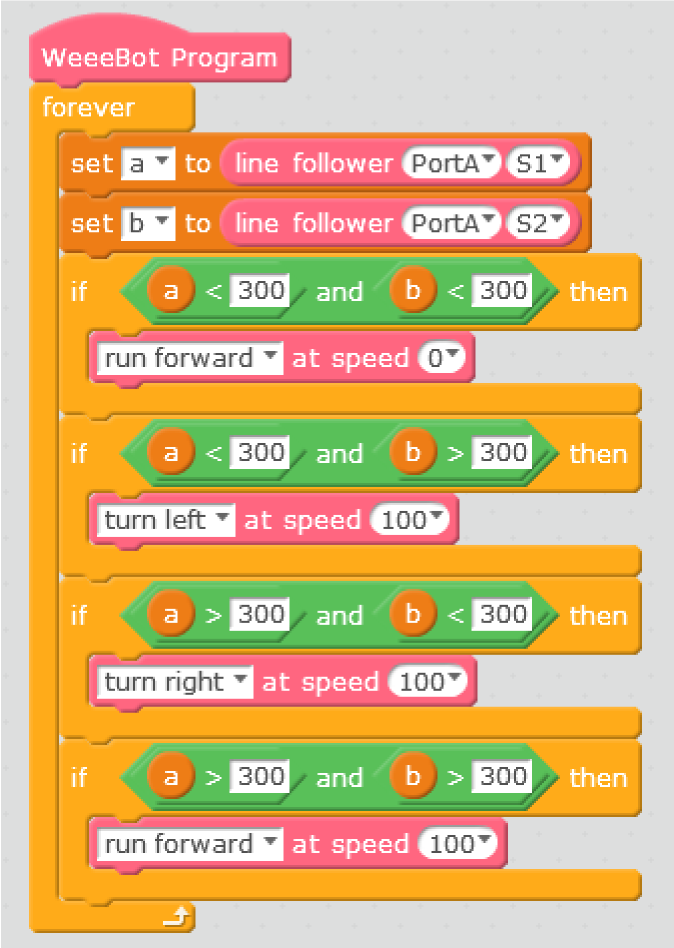


FIG. 12-13

Note:

a) The line-following algorithm of the double-way line-following sensor cannot be used to pass 90° intersection, such as L shape, T shape, or cross intersection.

When in mode 1: two sensors both leave the black line simultaneously; or mode 2: two sensors both enter the black line simultaneously, the robot does not know which direction it shall turn to.

b) If it is not 90° intersection, when the swerving angle is bigger, the proceeding speed shall be slower. Otherwise, the robot will rush out of the black line easily.

c) In the process of program test, students can stick the electrical tapes to the desktop to design the walking route at will, observe whether the robot can walk along the route, and clear up the desk after the class.

Ⅲ Classroom Development and Classroom Evaluation (10min)

1. Classroom development (selected lecture according to the classroom teaching pace).

Note: Development 1 “Stop on the precipice” is elementary and interesting, and teachers can give lectures according to the classroom teaching pace. The line-following of Development 2 map is very difficult, only for teachers’ learning or for the reference of students who have much interest in it.

a. Stop on the precipice: Let the robot walk on the desk. When it walks to the edge of the desk, it will turn around automatically.

a) Store the detected value of the two line-following sensors in the variable a and b, and then judge the value of a and b. If the robot is on the desk, the IR will reflect, which is equivalent to two line-following sensors on the white line, and the robot will move forward. The written program is as shown in FIG. 12-14.

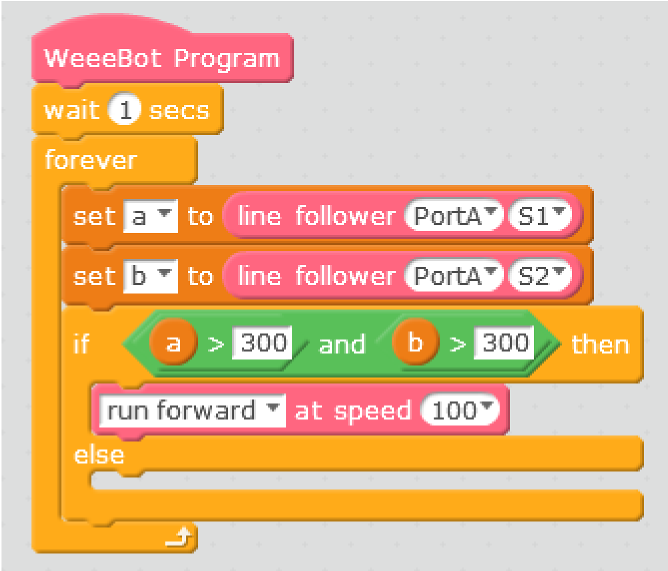


FIG. 12-14

Note:

①The proceeding speed of the robot shall not be very fast; otherwise, it will rush out of the desk quickly.

②Time delay is added at the beginning of the program, to avoid the misjudgment of the sensor when the robot is just turned on and avoid the execution of false motion.

b) If the two line-following sensors leave the edge of the desk simultaneously when walking, the IR cannot be reflected, which is equivalent to two line-following sensors both on the black line, and so the robot will need to move back and swerve. The written program is as shown in FIG.12-15.

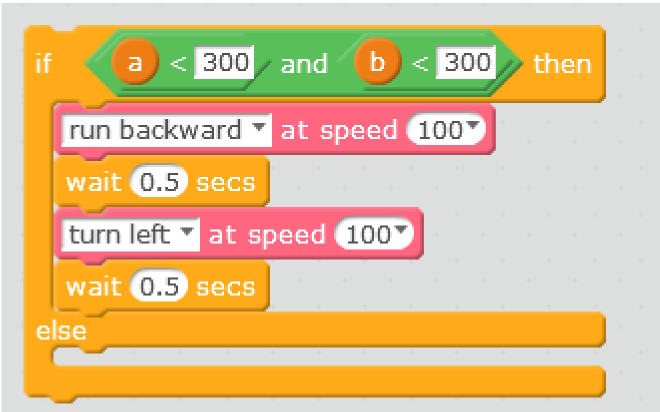


FIG. 12-15

c) If the left part of the robot leaves the edge of the desk first when walking, the IR of S1 cannot be reflected, but that of S2 can be reflected. Then the robot will need to move back and turn right to make the left part of the robot come back to the desk. In the same way, if the right part of the robot leaves the edge of the desk first when walking, the IR of S2 cannot be reflected, but that of S1 can be reflected. Then the robot will need to move back and turn left to make the left part of the robot come back to the desk. The written program is as shown in FIG. 12-16.

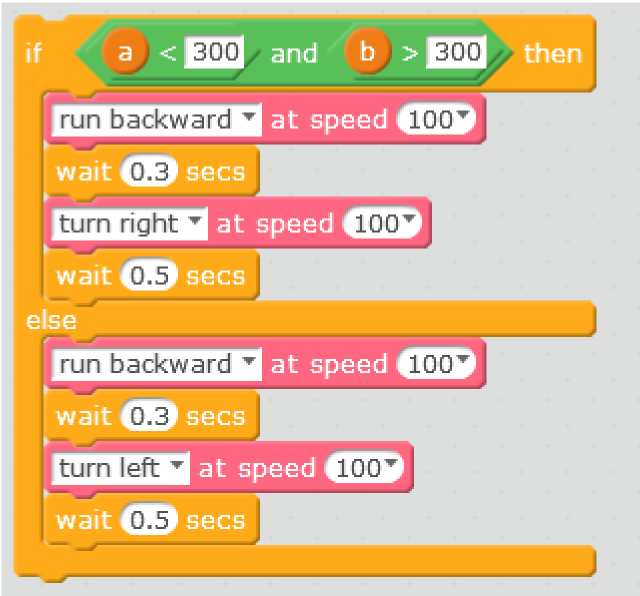


FIG. 12-16

Summarize the above programs, and we can get the total program as shown in FIG. 12-17.

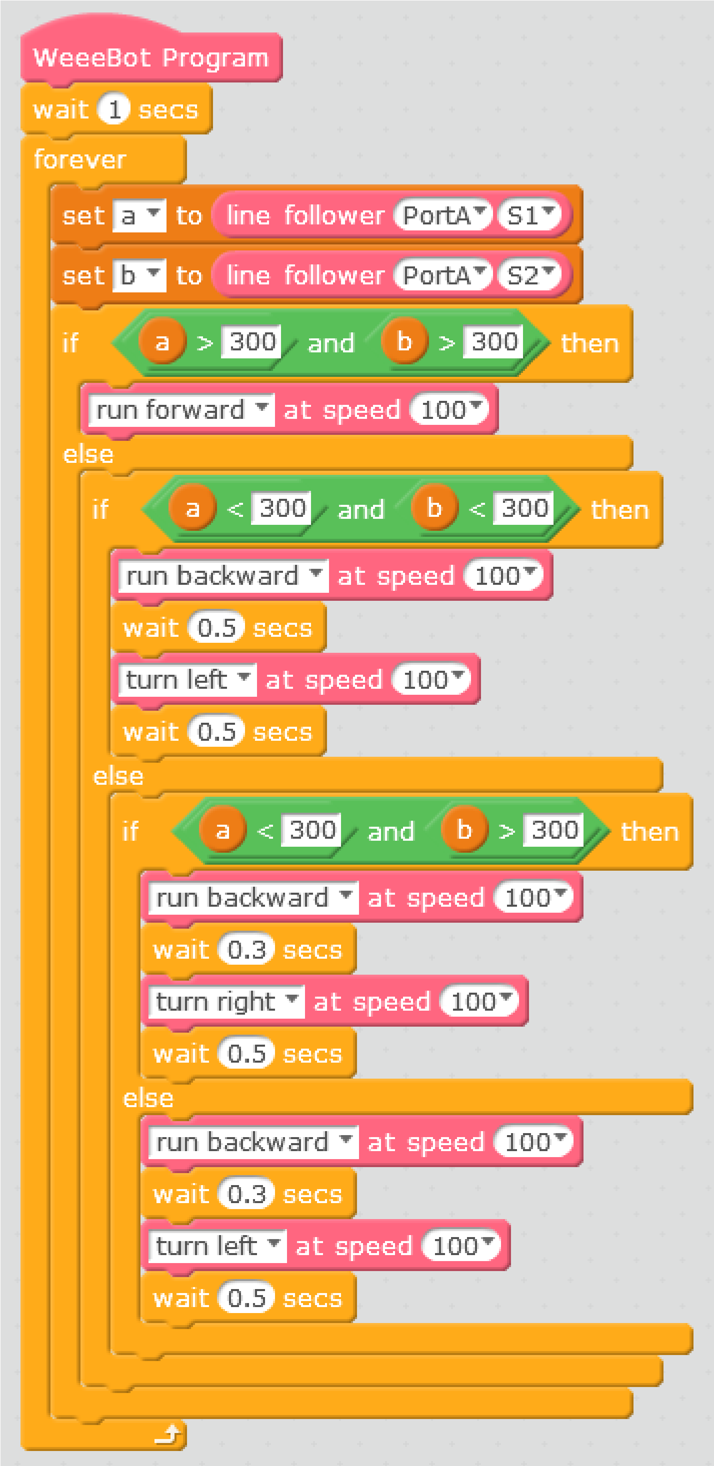


FIG. 12-17

Note:

① In the walking process of the robot, please take care to prevent the misjudgment of the sensor resulting in the falling of the robot from a high place.

② It will need to modify the swerving angles of the robot according to the desk radian. The larger the radian is, the longer the swerving time shall be.

b. Map line-following: Let the robot walk along the configured maps and pass

through the 90° intersection with round corners through the use of advanced patrol algorithm.

a) Store the detected value of the two line-following sensors in the variable a and b, and then judge the value of a and b. If the two line-following sensors are both on the black line, the robot will move forward. The written program is as shown in FIG. 12-18.

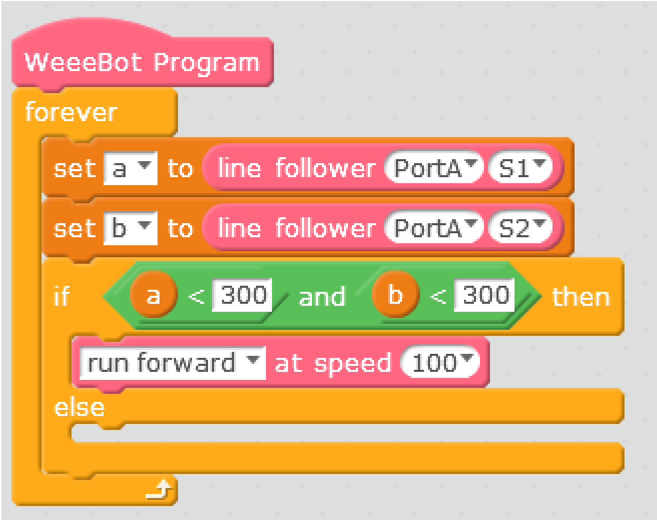


FIG. 12-18

b) If there is one line-following sensor on the black line, the robot will continue to go straight and record the state of the two line-following sensors: If S1 is out of the black line and S1 is in the black line, the state will be stored in variable c and be set to 1. Otherwise, be set to 2. The program is shown in FIG. 12-19.

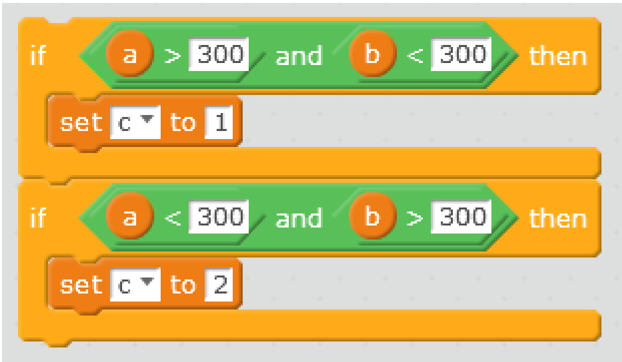


FIG. 12-19

Note: The algorithm can prevent the robot from keeping adjusting left and right directions during the walking.

c. If the robot keeps moving straight in the status of S1 beyond the black line and S2 in the black line, the robot will go out of the black line. When going out of the black line, the direction of the robot must be inclined to the left, as shown in FIG. 12-20. Then the robot will stop moving and move forward on the left wheel to keep the robot in the right direction, and vice versa. After running the program, there will be one or more line-following sensors on the black line. The robot will continue to move forward, as shown in FIG. 12-21.

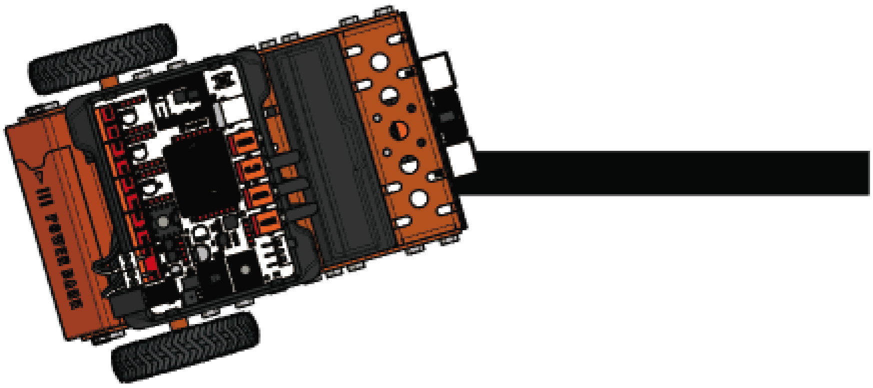


FIG. 12-20

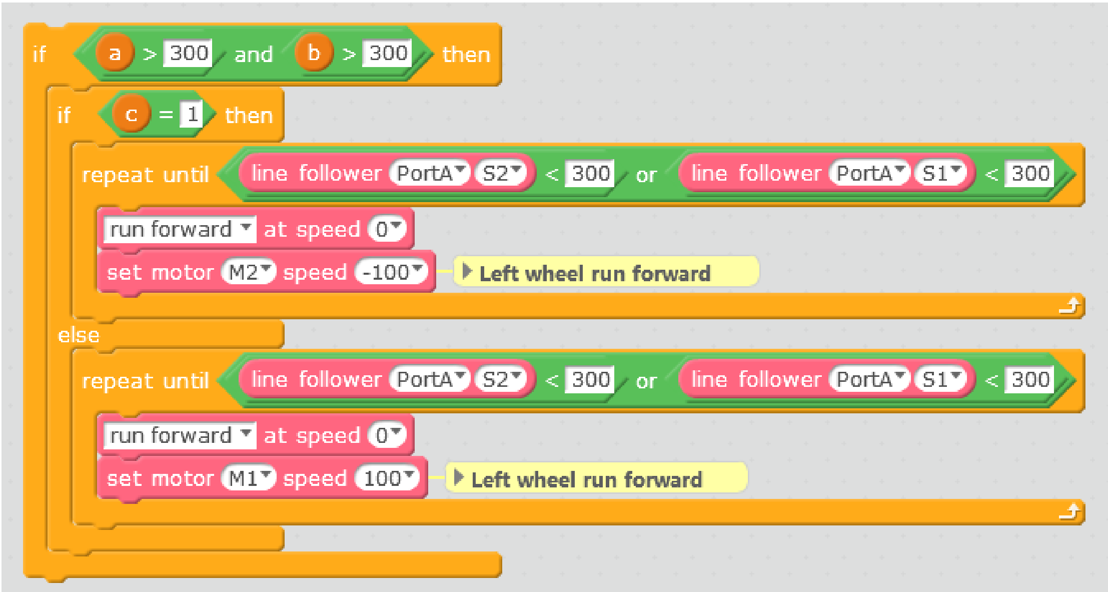


FIG. 12-21

Note: In this control mode, when the robot encounters a 90° intersection with round corners, it will move forward with a single wheel to make itself swerve. The two wheels will not move forward until there is one or more than one line-following sensor on the black line, to avoid the case that it rushes out of the black line.

Summarize the above program and the program is as shown in FIG. 12-22.

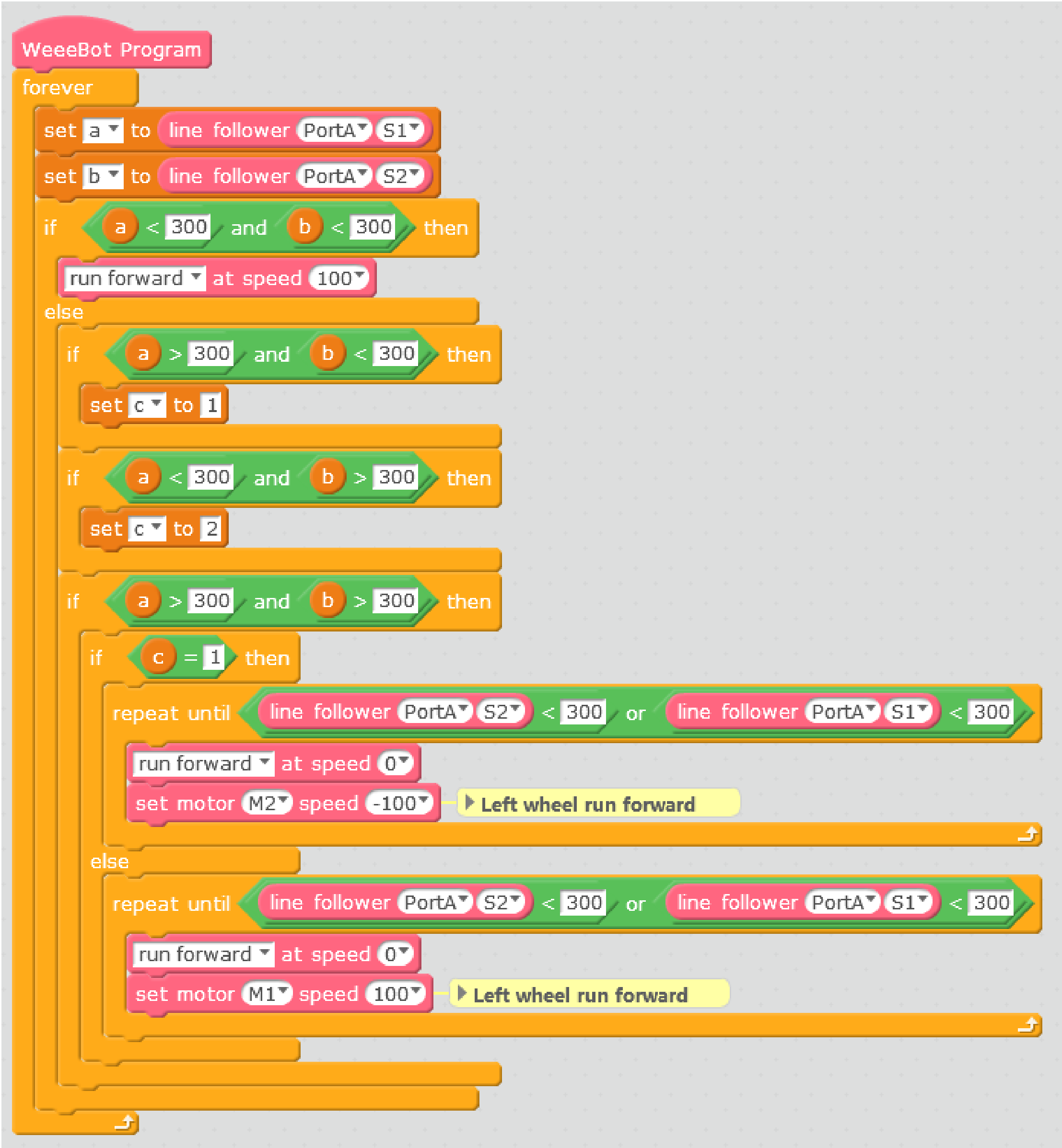


FIG. 12-22

2. Classroom evaluation: Students' class performance will be evaluated according to the evaluation criteria.

Evaluation criteria:

a. Understand the working principle of the line-following sensor and use it skillfully.

b. Proficient in two kinds of line-following algorithms.

c. Capable of making a line-following robot.

Chapter 13 My learning Partner

Summary
In this lesson, we will master the control method of LED panel module through exploratory learning and make learning partners---smart calculators.

Learning Objectives
To understand the working principle of the LED panel module and master the control method of the LED panel module.
Proficient in using the LED panel module to make calculators.

Course Time
90 minutes.

Pre-class Preparation
WeeeBot robots, computers with WeeeCode software installed, and USB download cables.

Teaching Process

I Introduction from Life (15min)

1. Share the application of screen in life, understand the types of screen and the coloring principle.
The applications of screens are far-reaching in life. Classified by the displayed color, screens can be divided into the single primary color display (including pseudo color display), double primary colors display, and full color (3 primary colors) display.

A single primary color display can only display one color, consisting of single color light, such as a calculator, showing black and white. Such as the electronic clock, this is called pseudo-color, which can only display a single color.

A double primary color display can display three colors, consisting of red and green lights, while the two together can constitute yellow, such as billboards.

The three primary color display can display any color, which is made up of red, green and blue lights, such as TV, mobile phone, and so on.

Ⅱ Know about the Module and Use It (45min)

1. Know about the LED panel module and understand its working principle.

No matter what color display, it is composed of many LED lights. Observe the LED panel module in WeeeBot, it is formed of 7 lines and 21 columns of LED lights, as shown in FIG. 13-1. The information can be displayed by controlling the on/off of the LED light in the LED panel module.

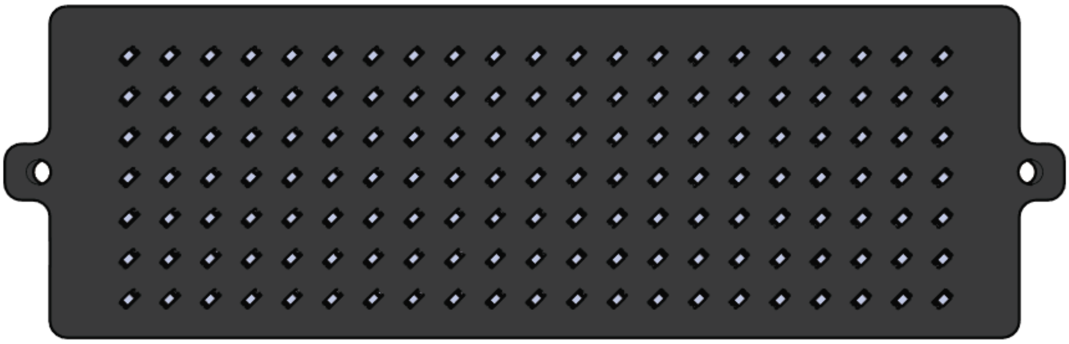


FIG. 13-1

2. Master the usage of the LED panel module.

We can find the control block of the LED panel module in the "Robot" category, as shown in FIG. 13-2 ~ FIG. 13-10.

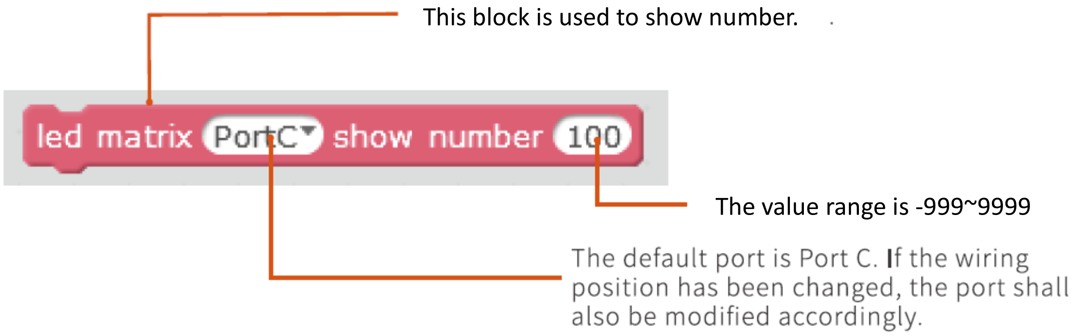


FIG. 13-2

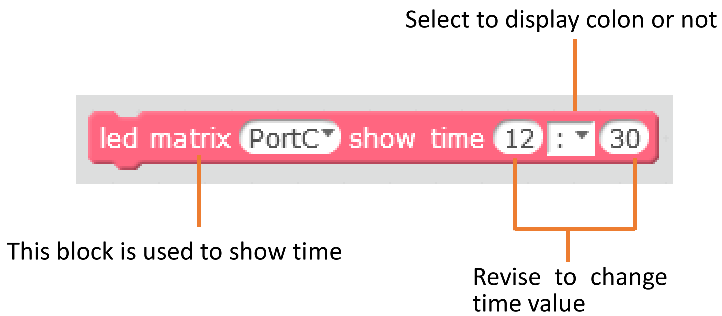


FIG. 13-3

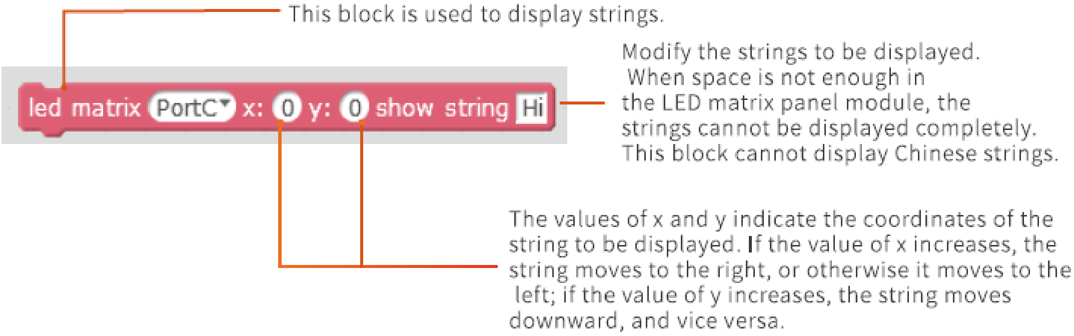


FIG.13-4

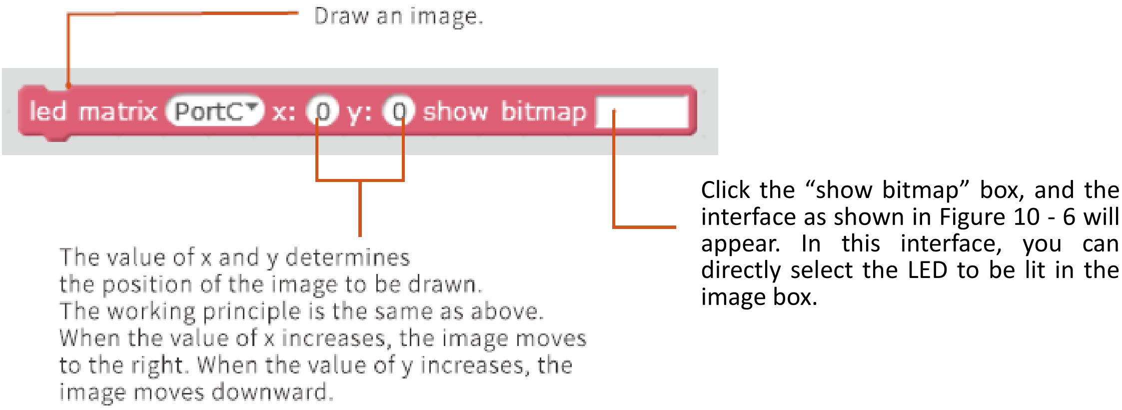


FIG. 13-5

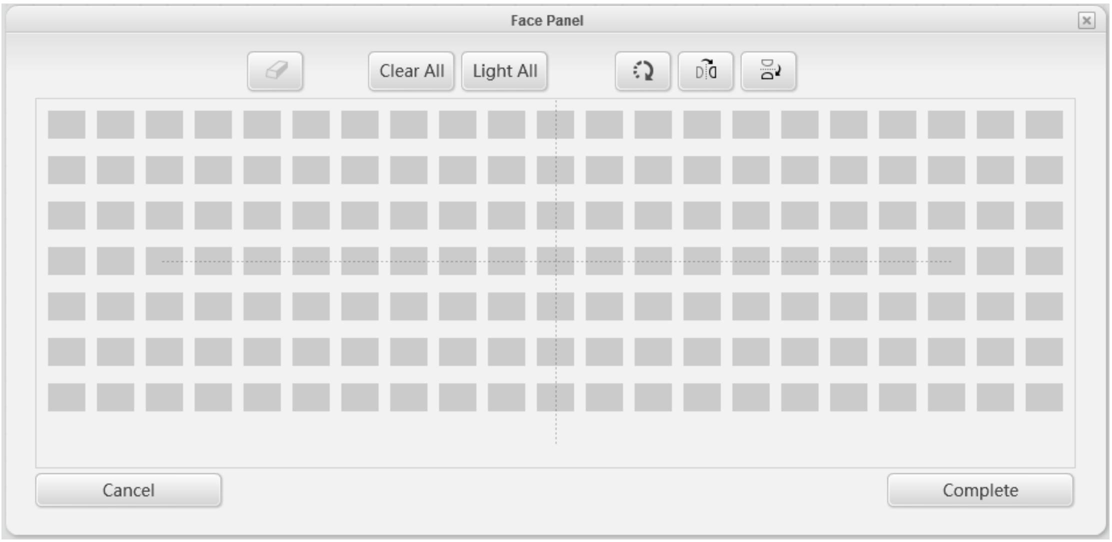


FIG.13-6

This block is used to turn on the LED light at a certain position on the LED matrix panel module. If the LED matrix panel module is seen as a coordinate system (as shown in Figure 10-8), then each LED light has a definite position in such coordinate system. For example, the coordinates of the LED light at the upper left corner and the one at the upper right corner are respectively (0, 0) and (20, 6). If you want to turn on any LED light, you only need to enter the coordinates of such LED light into the corresponding parameter field in this block.

FIG. 13-7

FIG. 13-8

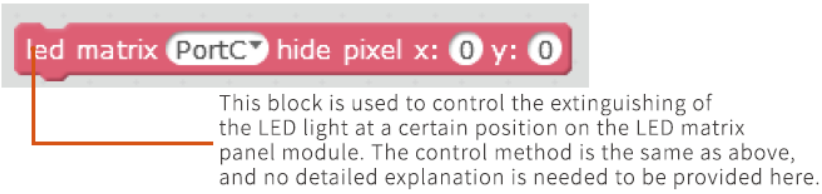


FIG. 13-9

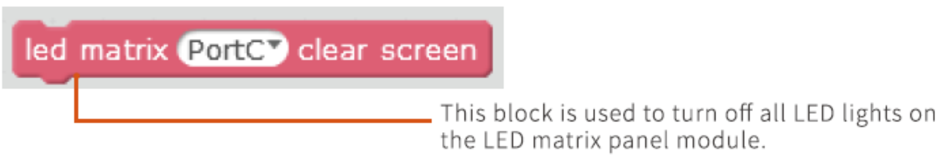


FIG. 13-10

Note: Teachers can combine the practice link to explain, and deepen the students' understanding in the control blocks of the LED panel module.

Practice 1: Write the program in the online mode as shown in FIG. 13-11 and run it. The LED panel module will display value “100”.

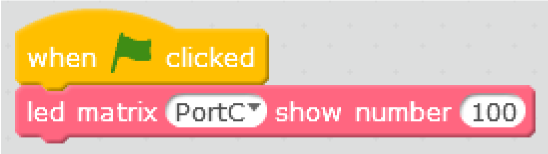


FIG. 13-11

Practice 2: Write the program in the online mode as shown in FIG. 13-12 and run it. The LED panel module will display the current time.

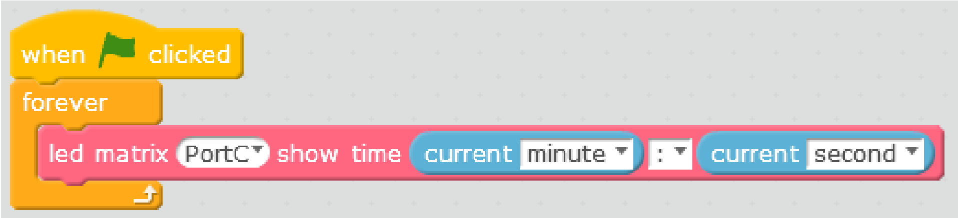


FIG. 13-12

Note: The “Current minute” block and the “Current second” block are in the Sensing category.

Practice 3: Write the program in the online mode as shown in FIG. 13-13 and run it. The LED panel module will display the character “HI."

FIG. 13-13

Exploration 1: Run the program as shown in FIG. 13-13. Change the X-axis from 0 to 1 and observe the effect.
Program effect: “HI” move to the right on the LED panel module.

Exploration 2: Run the program as shown in FIG. 13-13. Change the Y-axis from 0 to 1 and observe the effect.
Program effect: “HI” move downwards on the LED panel module.

Exploration 3: Run the program as shown in FIG. 13-13. Change the character to HAHA and observe the effect.
Program effect: “HAHA” is displayed on the LED panel module.

Conclusion: If x increases, the character will move to the right; if y increases, the characters will move downwards. It can be seen from exploration three that the space of each letter is five columns, and the spacing between letters is 1 column. When the space of the LED panel module is not enough, the displayed characters will be incomplete.

Practice 4: Write the program in the online mode as shown in FIG. 13-14 and run it. The LED panel module will display a smiling face.

FIG. 13-14

Practice 5: Write the program in the online mode as shown in FIG. 13-15 and run it. The LED light of the first line and the first column in the LED panel module will flash every 1 second.

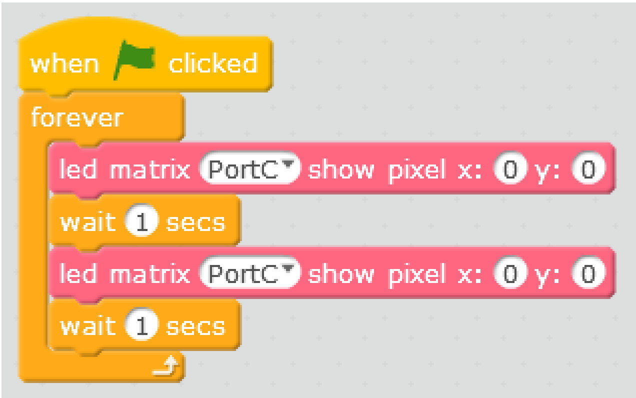


FIG. 13-15

Ⅲ Programming (20min)

1. Make Learning Partners---Smart Calculator.

We know that the WeeeCode software has its computing power, and we can write programs to display the results on the LED panel module.

a. Create four variables “a," “b," “c” and “d”: “a” and “b” is used to store the value to calculate, “c” is used to store the calculated result and “d” is used to store the algorithm to initialize the variable. The program is as shown in FIG.13-16.

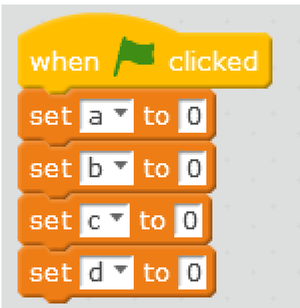


FIG. 13-16

Note:

① To calculate, please change the value of “a," “b” and “d."

② The value of “d” represents different algorithms. “1” is “plus”, “2” is “minus”, “3” is “multiply” and “4” is “divide”.

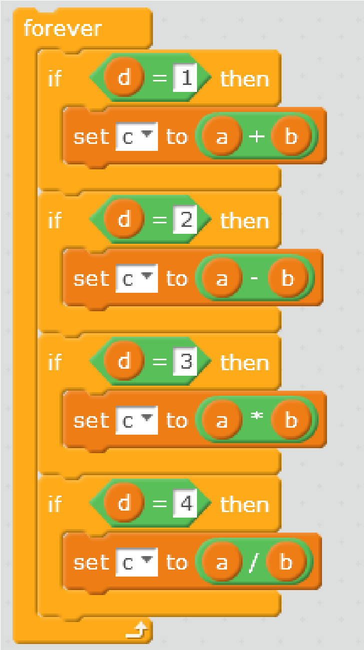


FIG. 13-17

c. Display the calculated result “c” on the LED matrix and the total program is as shown in FIG. 13-18.

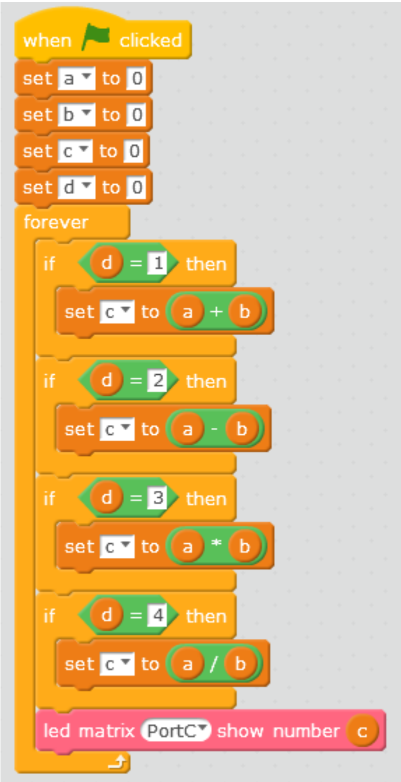


FIG. 13-18

Ⅳ Classroom Development and Classroom Evaluation (10min)

1. Classroom development: make emotional partners -- creative expression packages that cultivate students' creativity and imagination (selected lecture according to class teaching pace).

The “display image” block of the LED panel module can be used to create an expression, and the following two development parts are selective to give lectures by teachers.

a. Control the “blink eyes” programmatically in the LED panel module. The reference program is as shown in FIG. 13-19.

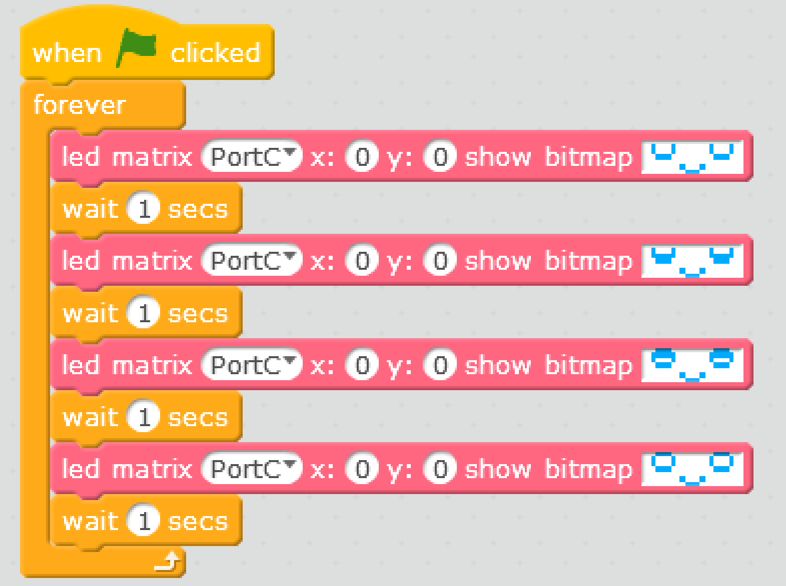


FIG. 13-19

b. Control the LED panel module programmatically to display different expressions, and the reference program is as shown in FIG. 13-20.

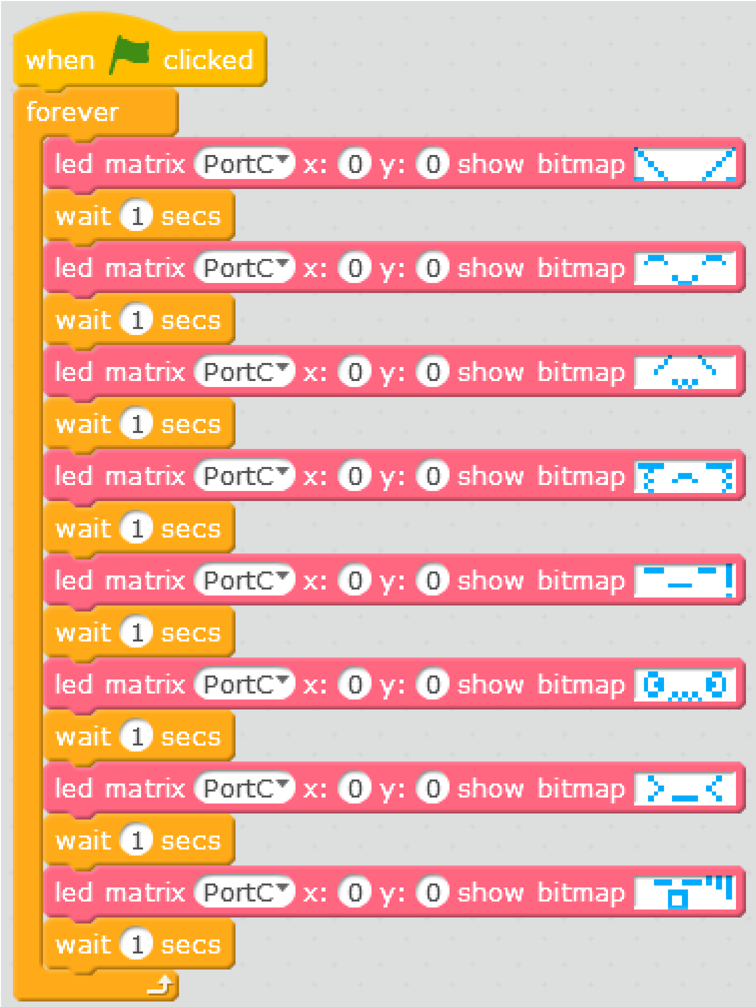


FIG. 13-20

Note: This development can be creative and can enrich the control mode of the robot by combining motor movement or tone change.

2. Classroom evaluation: students' class performance will be evaluated according to the evaluation criteria.

Evaluation criteria:

a. Master the control blocks of the LED panel module.

b. Able to use the LED panel module to make smart calculators.

Chapter 14 Cool LED matrix

Summary

On the basis of the mastery of LED panel module control blocks in the last lesson, in this lesson, we will cultivate students' logical ability in interesting programming by finding expression animation running laws and by programming to transform the displayed content of the LED panel module.

Learning Objectives

To master the animation operation rules of the LED panel module and clarify the programming ideas in the complicated logic.
To make a LED rolling animation and the bouncing animation when the lights touch the wall.

Course Time

180 minutes.

Pre-class Preparation

WeeeBot robots, computers with WeeeCode software installed, and USB cables.

Teaching Process

Ⅰ Introduction from Life (10min)

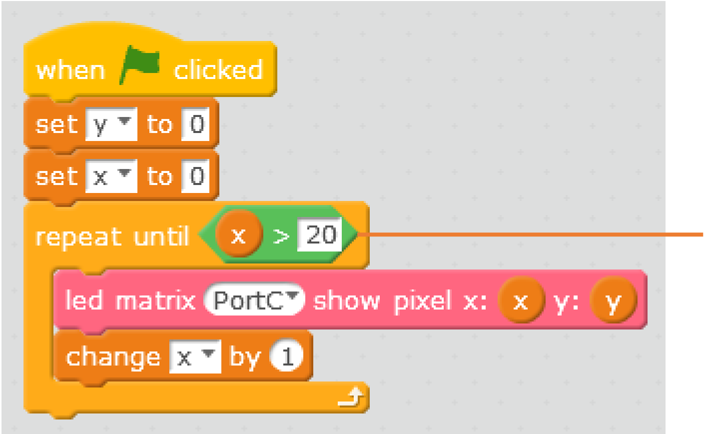
1. Share the common screen animation in life.

In our lives, we not only let the screen display the information, but also often make it play in a particular animation, such as the rolling of the slogan, the rolling of the lights, the dynamic movement of the lights, and so on.

Ⅱ Programming (160min)

1. Program a LED rolling animation: Turn on the light of the LED panel module from left to right and from up to down. After all the lights of the screen are lit, turn off the lights one by one.

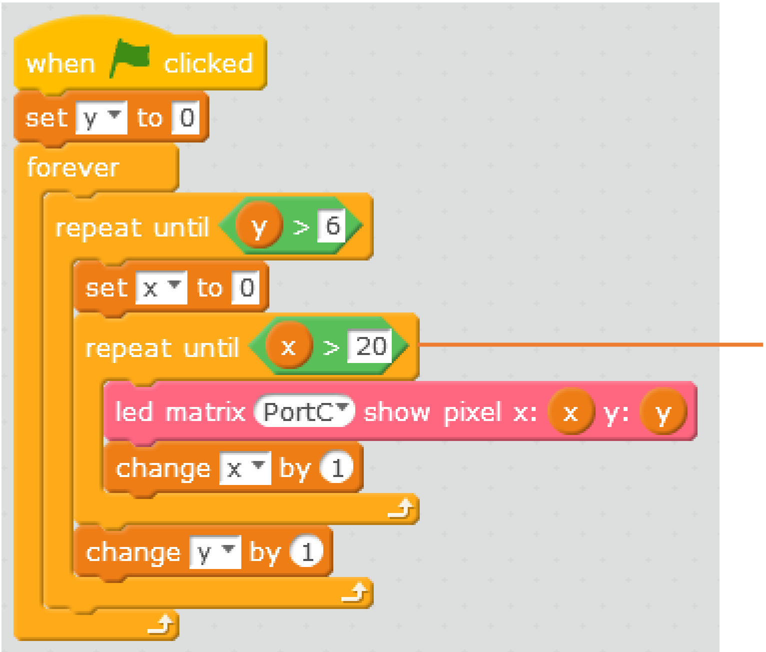
a. Create the coordinate position of X and Y variable to store the light, Write the loop program and lighten the lights in the first line of the LED panel module. The program is as shown in FIG. 14-1.



After lit the LEDs of the coordinate (x: 19,y:0), the value of "x" will increase by 1 and become 20.
It does not meet the judgment of 20>20; the program will continue.
After lightening the light of the coordinate (x: 20,y:0), the value of "x" will increase by 1 and become 21.
It meets the judgment of 21>20; the program will jump out of the loop.

FIG. 14-1

b. After lightening the lights in the first line, if you want to lighten the lights in the second line, you will need to increase the value of "y" by 1 and reset the value of "x," that is, you will need to increase the line number and initialize the column number. The modified program is as shown in FIG. 14-2.



After lit the light of the coordinate (x: 20,y:5), the value of "y" will increase by 1 and become 6.
It does not meet the judgment of 6>6; the program will continue.
After lightening the light of the coordinate (x: 20,y:6), the value of "y" will increase by 1 and become 7.
It meets the judgment of 7>6; the program will jump out of the loop and stop.

FIG. 14-2

c. In the same way, the light extinguishing program can be concluded as shown in FIG 14-3. The total program is as shown in FIG. 14-4.

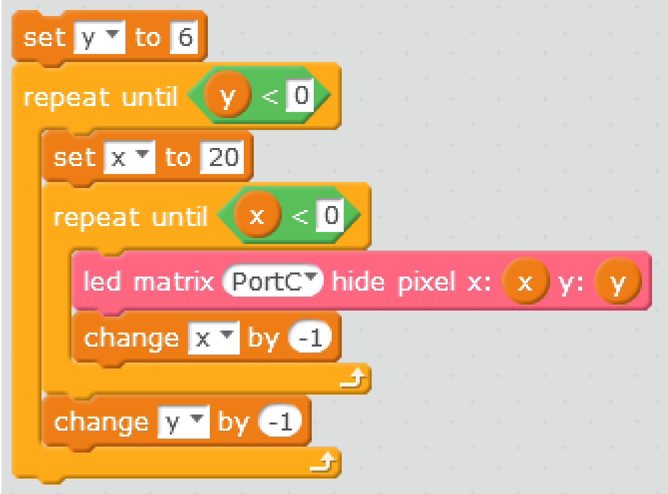


FIG. 14-3

Note: Before extinguishing the lights, you will need to set the value of variable "y" to "6". This is because after running the program shown in FIG. 14-2, the value of the variable "y" is 7, which is beyond the line scope of the LED panel module.

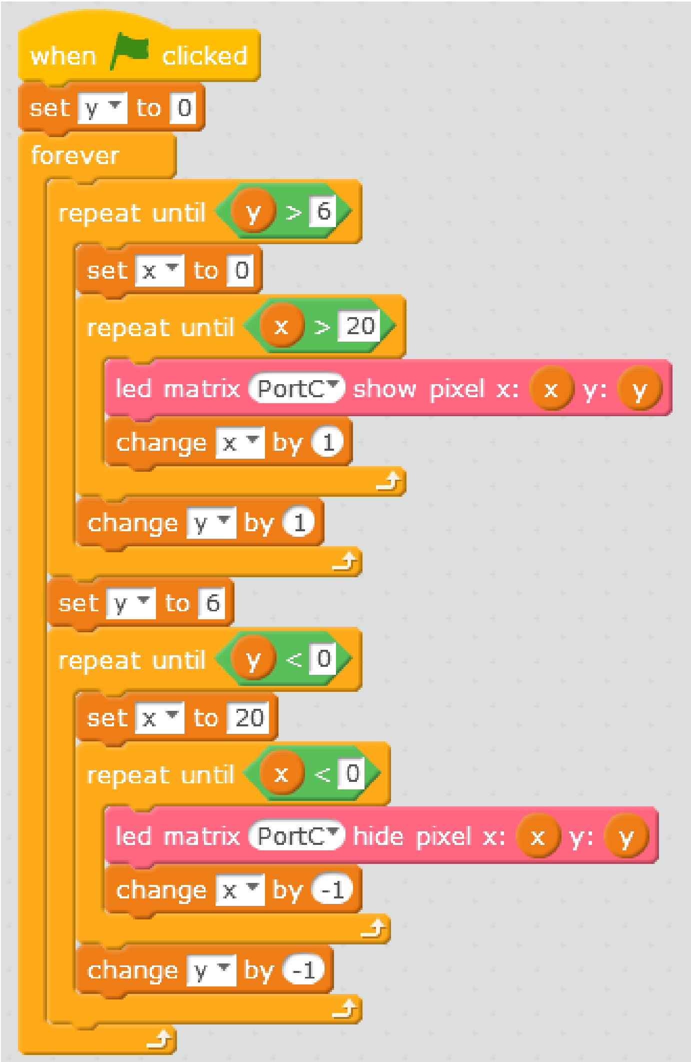


FIG. 14-4

2. Write the bouncing animation when the lights touch the wall: The light will start from the upper-left corner of the LED panel module and move to the below direction of 45° angle. When touching the bottom, it will bounce to the upper direction of 45° angle. It will repeat the motion and will eliminate the motion trajectory in the movement process.

a. Create two variables “x” and “y." “x” and “y” are the coordinates of the light position. When the light starts to move downward, the values of “x” and “y” shall increase, and the written program is as shown in FIG. 14-5.

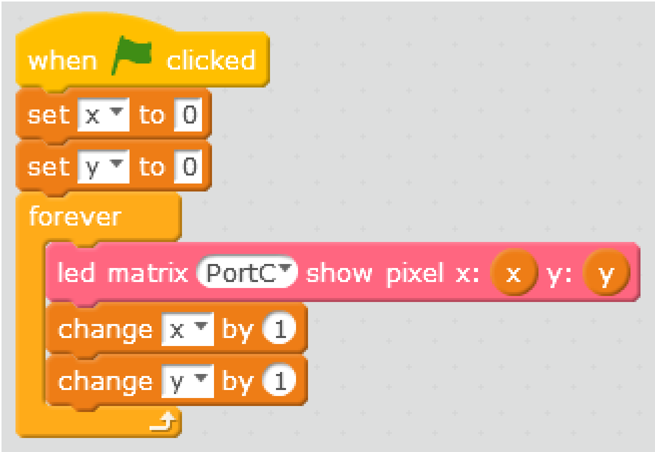


FIG. 14-5

b. In the motion process of the light, you will need to turn off the light in the last position, and then create two variable “previous x” and “previous y ”, which are used to store the former value before the coordinate of the xy changes, so as to extinguish the lights in the position of “previous x” and “previous y ”. The written program is as shown in FIG. 14-6.

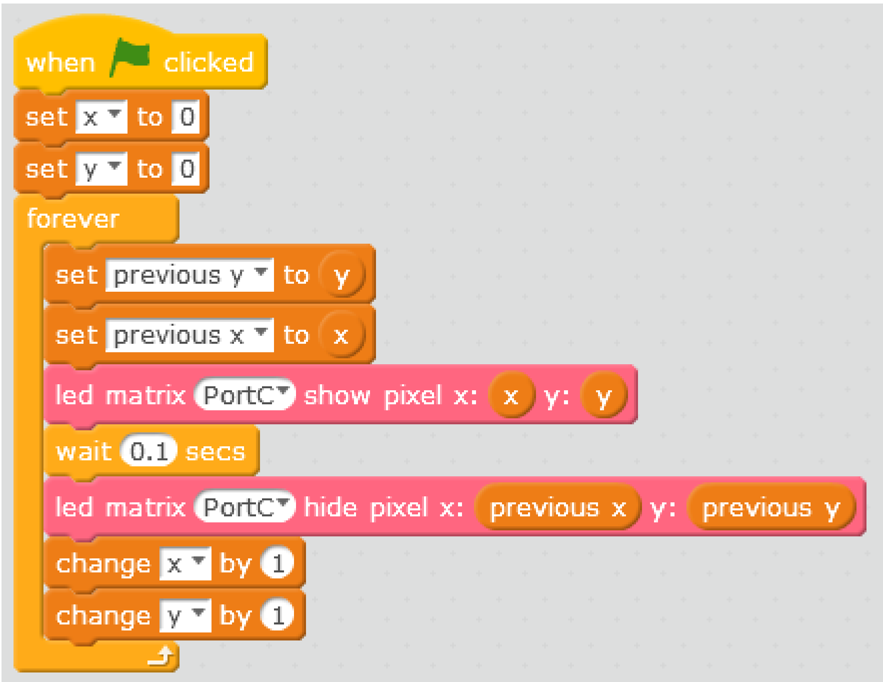


FIG. 14-6

Note: In this program, you will need to set the variable values of "previous x" and "previous y " before the changes of x and y value.

c. It can be know from the task requirements that when the program is run to this step, the movement direction of the light shall change from downwards 45° to upwards at 45°. The x value will increase, and the y value will decrease. It is not hard to find that when y increases to 6 or decreases to 0, the y value needs to change from an increase to a decrease or from a decrease to an increase. The same goes when the x value increases to 20 or decreases to 0. To simplify the program, we create two new variables, x change value and y change value, to modify the value of x and y, and initialize it to 1. The modified program is as shown in FIG. 14-7.

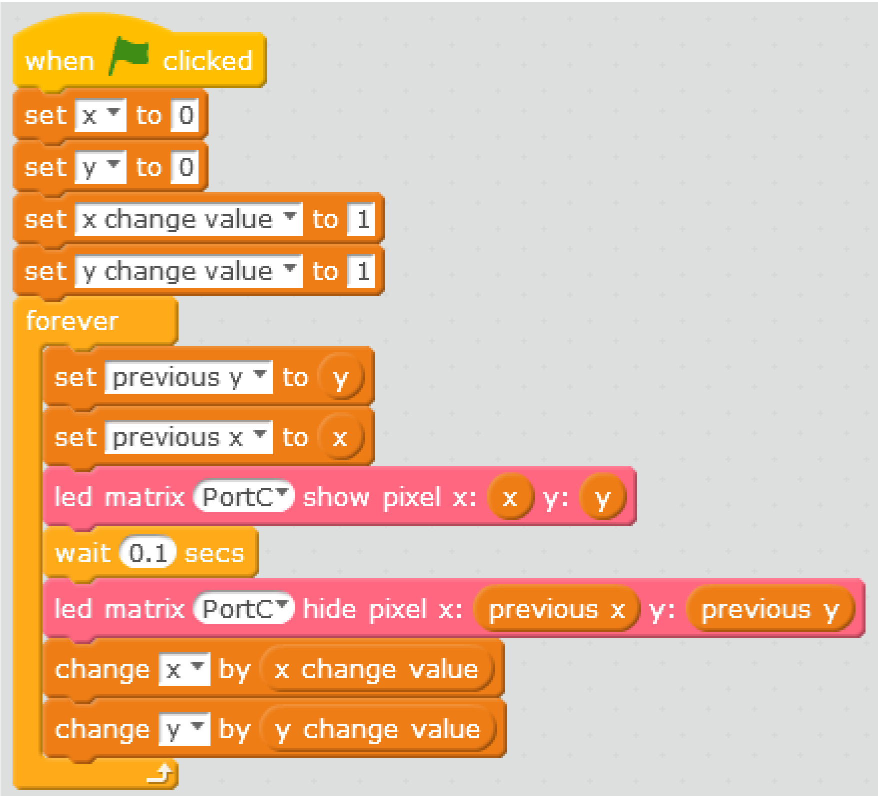


FIG. 14-7

The program running effect will be the same as above. Then increase the judgment to x and y value. If the light reached the edge of the screen, it would change direction, and the state will change from an increase to a decrease or from a decrease to an increase through (0 - change value). The written program is as shown in FIG. 14-8 and the modified total program is as shown in FIG. 14-9.

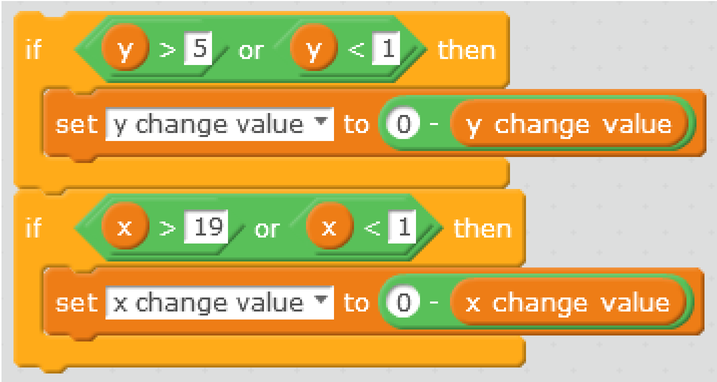


FIG. 14-8

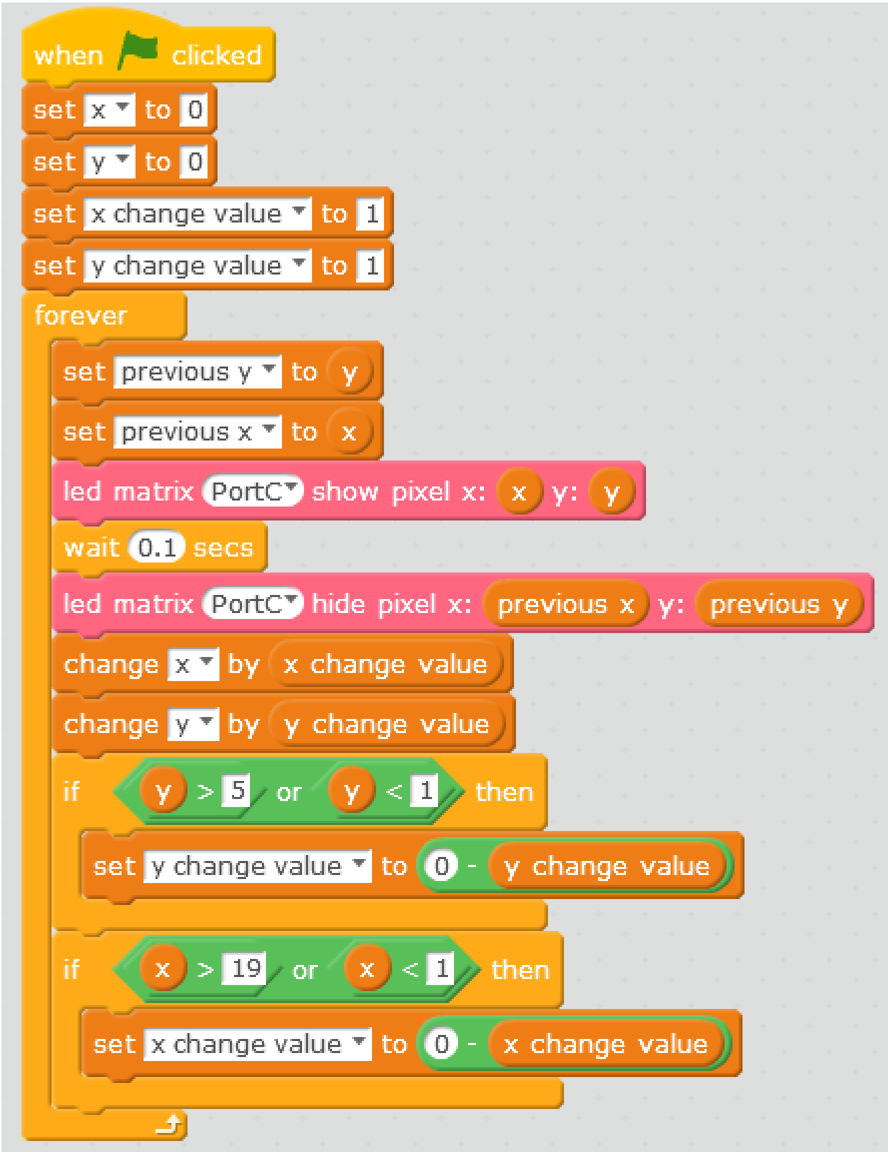


FIG. 14-9

Program effect: Run the program, the light can move and will bounce when touching the edge of the screen.

①The judgment of the x and y values must be after the changes in x and y values. Otherwise, the light will remain in the initial position.
②It is very difficult to understand the stacking sequence of the program blocks and the limited range of the values. Teachers can use more actual values to help students to understand in the teaching process.

Ⅳ Classroom Development and Classroom Evaluation (10min)

1. Classroom development: make a LED matrix rolling display program (selected part according to the class time).

a. Create a variable x as the x-axis of the character. Realize the right shift of the character by increasing the x-axis. The program is as shown in FIG. 14-10.

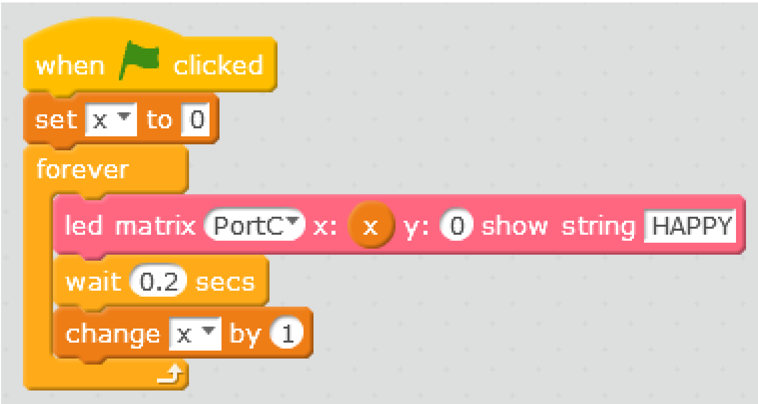


FIG. 14-10

Program effect: Run the program, it can be discovered that the character appears directly in the center of the screen, the letter "Y" has already exceeded the display scope of the LED panel module, and the character’s scrolling to the right is not circular.

a. Create a variable “number of characters," representing the numbers in character. For example, there are five letters in “HAPPY," and the number of characters is 5. In the previous exploration, we know that the horizontal range of the LED panel module is 0~20, and each letter occupies five columns, and the spacing of each letter is 1, and thus each character occupies 6 Spaces. We set the initial value of x to 0, and use “x- (number of characters *6)” as the initial position of the scrolling screen. If the number of characters is 5, it will start scrolling from the x-axis of -30, as shown in FIG. 14-11. It will scroll until “x - (number of characters * 6) > 20," and then the scrolling is completed, namely the coordinate values of the first letter in character is broader than the display scope of the LED panel module, as shown in FIG. 14-12. At this time reset x to zero, and the modified program is as shown in FIG. 14-13.

FIG. 14-11

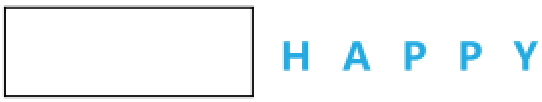


FIG. 14-12

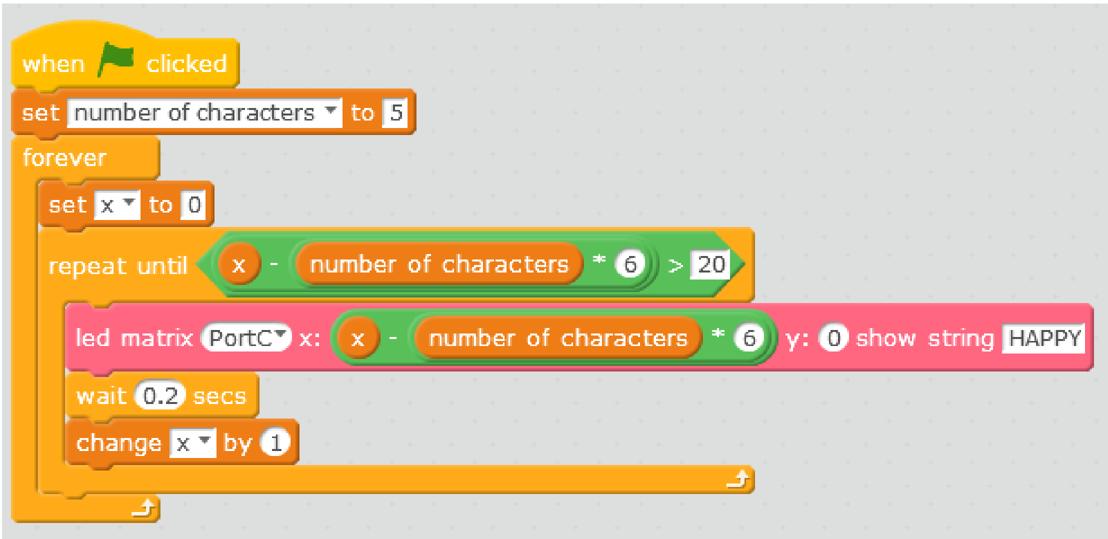


FIG. 14-13

Note: If you want to change the characters of the scrolling screen, please modify the variable values of the character and the “display character” parameters of the control block in the LED panel module.

2. Classroom evaluation: students' class performance will be evaluated according to the evaluation criteria.

Evaluation criteria:

a. Able to make a LED rolling animation on LED panel module.

b. Able to make a bounce animation when light touches the wall.

Chapter 15 Gluttonous Snake

Summary

In this lesson, we will combine the remote control and LED panel module to make the gluttonous snake game and cultivate students' logical thinking ability in the complicated and exciting logical relations.

Learning Objectives

To understand the concept of the list and master the use of list function.
To clarify the complex logic relations and make the gluttonous snake game.

Course Time

180 minutes.

Pre-class Preparation

WeeeBot robots, computers with WeeeCode software installed, USB cable and remote controls.

Teaching Process

Ⅰ Course Introduction (20min)

1. Let the students know about the game through game sharing, as so to provide ideas for developing games.

Teachers can let students share and summarize the functions of several roles of the game by showing the relevant pictures of the gluttonous snake game or by letting students experience the gluttonous snake game, as shown in FIG. 15-1.

Snake	1. Control the snake to move in each direction through the key.
	2. It will get longer after it eats food.
	3. The game will be over once it touches itself and the wall.
Food	Appear in the random position of the screen.

FIG. 15-1

Ⅱ Master the knowledge points related to game production (40min)

Note: This link is the learning of primary knowledge points for students to write the game, and teachers can also explain it in the process of writing the game.

1. Understand the concept of the list and master its usage.

A list is a container to store many variables. You can store or obtain each value of a variable in the container. It is like a dressing table with many drawers, and each drawer is stored with items, as shown in FIG. 15-2.

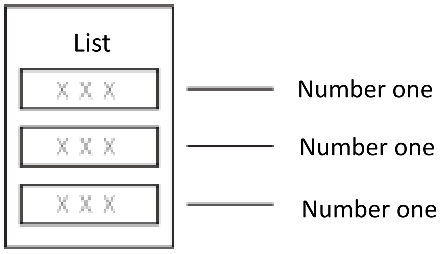


FIG. 15-2

In the “Data&Blocks” category, we can see the "Make a List" block. Click this option and enter the list name (“a” in this case), as shown in FIG. 15-3. After naming the list, you can see there are ten blocks in the block palette, as shown in FIG. 15-4.

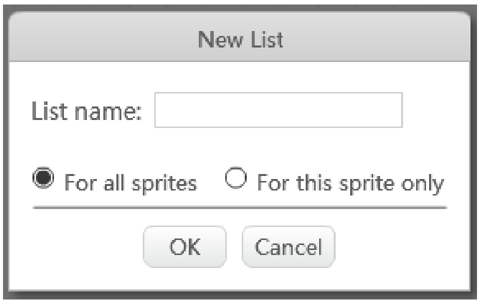


FIG. 15-3

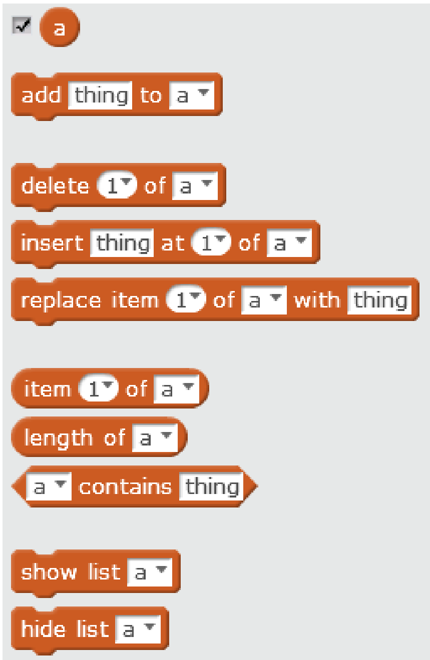


FIG. 15-4

a. 

: The list name, clicking the checkbox can display or hide the list value display on the stage.

b. 

: Add new variable at the end of the list.

c. 

: Delete the variable in the designated position of the list.

d. 

: Insert variable in the designated position.

e. 

: Replace the variable in the designated position to another variable.

f. 

: Obtain the variable value in the designated position.

g. 

: The quantity of variable that returns to the list.

h. 

: Determine whether there is specific value in the list.

i. 

: Display the list on the stage.

j. 

: Hide the list on the stage.

After creating the list, the list will be displayed on the stage by default, as shown in FIG. 15-5. The list is initially empty, and so the list length is 0.

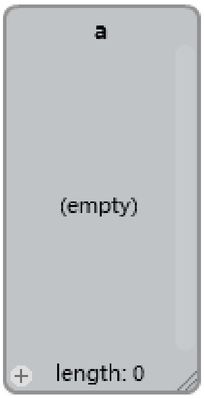


FIG. 15-5

Practice 1: Write the program as shown in FIG. 15-6. Understand the control blocks related to the list through the program effect.

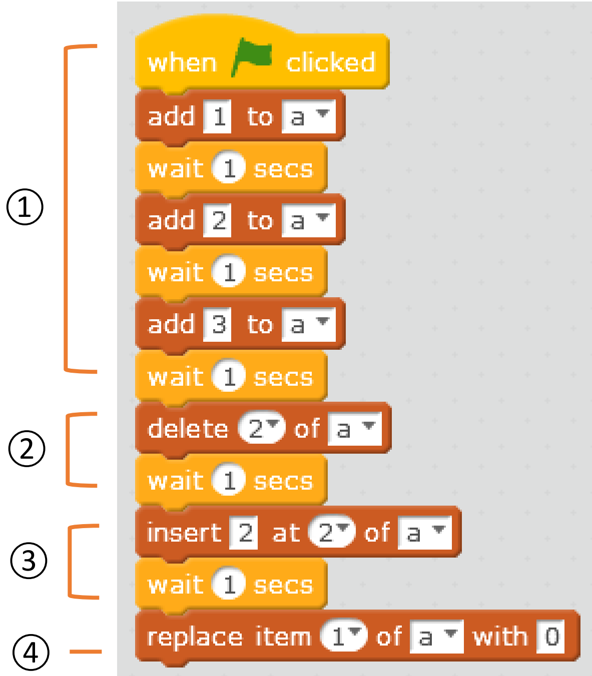


FIG. 15-6

Program effect:

① Add value to the list every 1 second and finally generate the list “a." There are three variables in the list “a," as shown in FIG. 15-7.

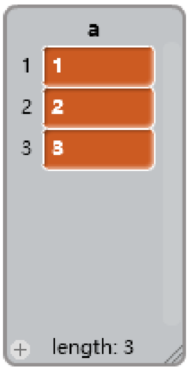


FIG. 15-7

② Delete the variable “2” in item 2, and there are only two variables “1” and “3” in the list, as shown in FIG. 15-8.

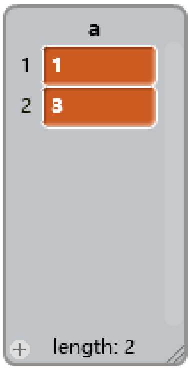


FIG. 15-8

③Insert the variable "2" in item 2, and there are three variables ---1, 2 and 3 in the list "a," as shown in FIG. 15-9.

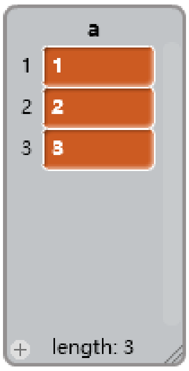


FIG. 15-9

④Replace the value of item 1 in the list with 0. There are three variable---0, 2 and 3 in the list "a," as shown in FIG. 15-10.

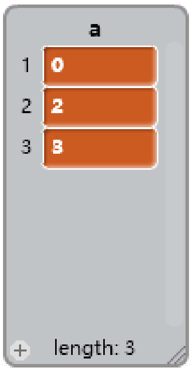


FIG. 15-10

Note:

a. If the parameter of the third part block position of the program is modified to 5, as shown in FIG. 15-11, the list will not change. That is because 5 exceeds the length "3" of the list. If the parameter of the third part block position of the program is modified to 4, it will have the same function with the "Add …to [list]" block, because position 4 is close to the end of position 3.

FIG. 15-11

b. If the parameter of the second part block of the program is modified to 5, as shown in FIG. 15-12, the list will not change, as there are three variables in the list only.

FIG. 15-12

Practice 2: Write the program as shown in FIG. 15-13 by the generated list as shown in FIG. 15-10. Understand the control blocks related to the list through the program effect.

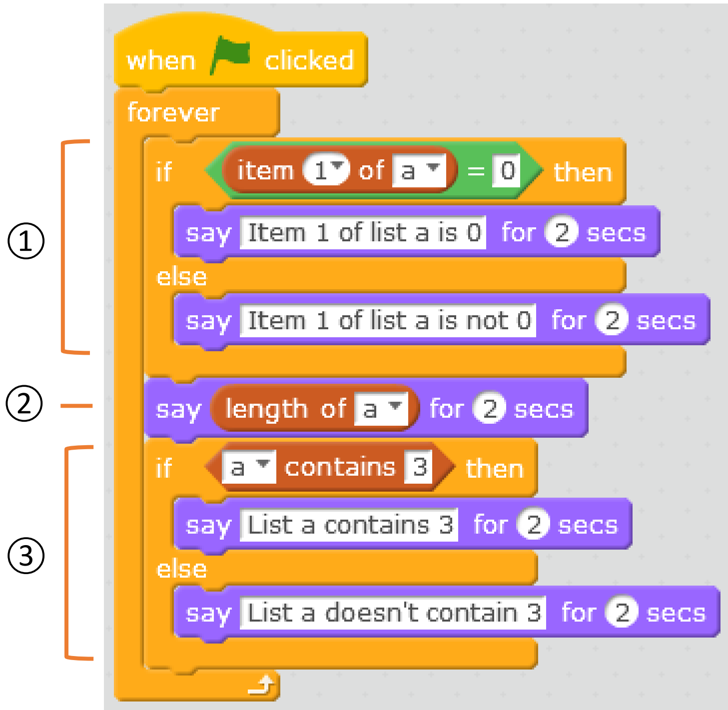


FIG. 15-13

Program effect:

① Judge the variable value included in the list, and the program effect is as shown in FIG. 15-14.

FIG. 15-14

② Obtain the number of the variable in the list and say through the role. The program effect is as shown in FIG. 15-15.

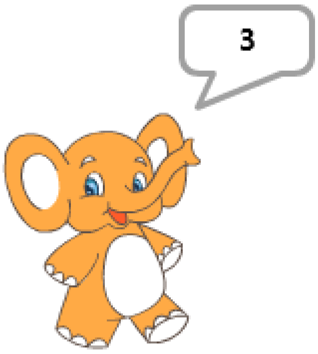


FIG. 15-15

③ Judge whether the list "a" includes variable 3 or not. The program effect is as shown in FIG. 15-16.

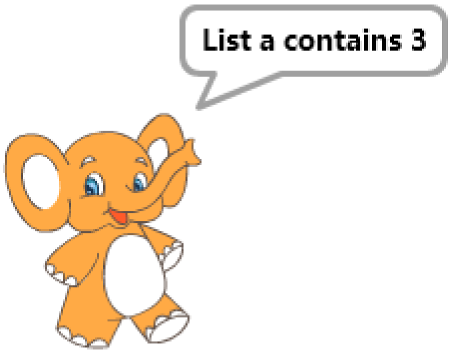


FIG. 15-16

Ⅲ Write Game Programs (110min)

1. Set the initial length and direction of the snake and make it move.

a. Create a variable and initialize them.

a) The initial value of the "body length" is 3, representing the body length of the snake.

b) The initial values of x and y both are 0, used to represent the coordinate position of the snakehead, that is, the snake starts to move from the upper-left corner of the screen.

c) The initial x and y change values are 1 and 0 respectively, used to represent the moving direction of the snake, that is, the snake moves to the right at the beginning. When it moves to the right, its x-axis value will increase, and its y-axis value will remain the same. The principle is as shown in FIG. 15-17 and the initialization program is as shown in FIG. 15-18.

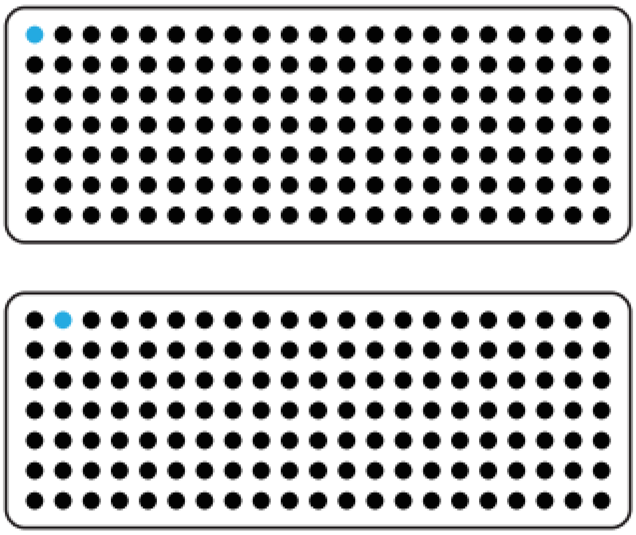


FIG. 15-17

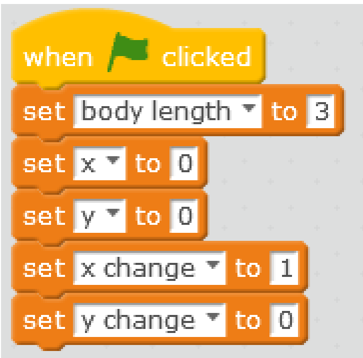


FIG. 15-18

b. Set the snake movement animation: If the snake is to move, the snake-head coordinate x, y needs to change, and the lights in the position of the x and y-axiss need to be displayed on the LED panel module. The program is as shown in FIG. 15-19.

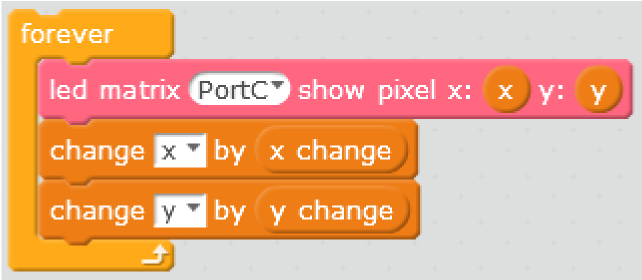


FIG. 15-19

After the above procedures are completed, the total program is as shown in FIG. 15-20.

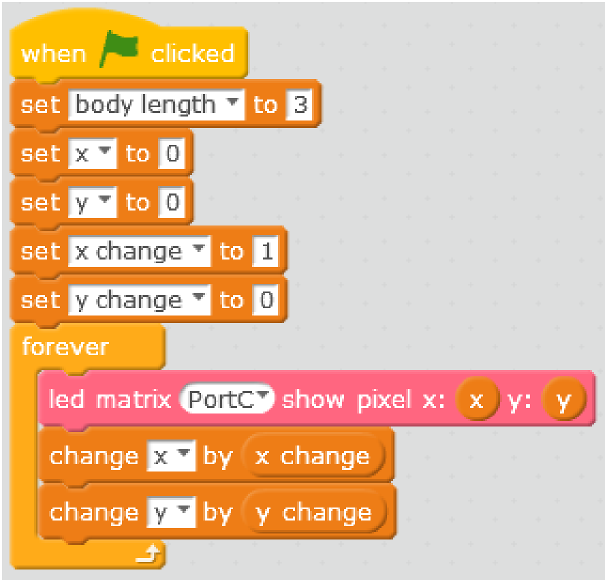


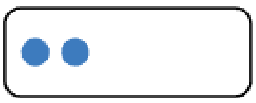
FIG. 15-20

Program effect: The snake begins to appear from the upper left of the LED panel module and moves to the right until it reaches the right side of the LED panel module. Moreover, during the time, the snake's body grows longer continuously.

2. Control the body length of the snake by controlling the length of the list.

a. Create "x-axis " and "y-axis " of the list, to store the coordinate position where snakehead passes through. The program is as shown in FIG. 15-21. After running the program three times, the relationships between the effect of the LED panel module and list are as shown in FIG. 15-22, namely to ensure that the coordinates of the item 1 values in the two list represent the coordinates of the snakehead.

FIG. 15-21



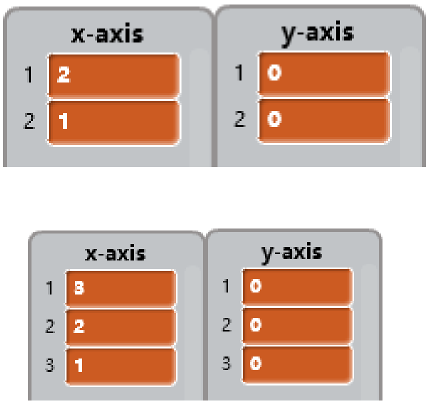


FIG. 15-22

b. (Challenge) Create a new variable “i” and set the value of “i” to the length of the list (in this case, take x-axis list for example), as shown in FIG. 15-23. Judge the “i” value. If the “i” value is larger than the value of the body length, extinguish the light of item “i” of the list x-axis and y-axis, that is, the light in the last item of the x and y position. Then delete the variable in the last item of the list x-axis and y-axis and keep the length of the two list are consistent with the body length. Moreover, the values of the last item in the two lists shall both be the coordinate values of the snake tail. The program is as shown in FIG. 15-24.

FIG. 15-23

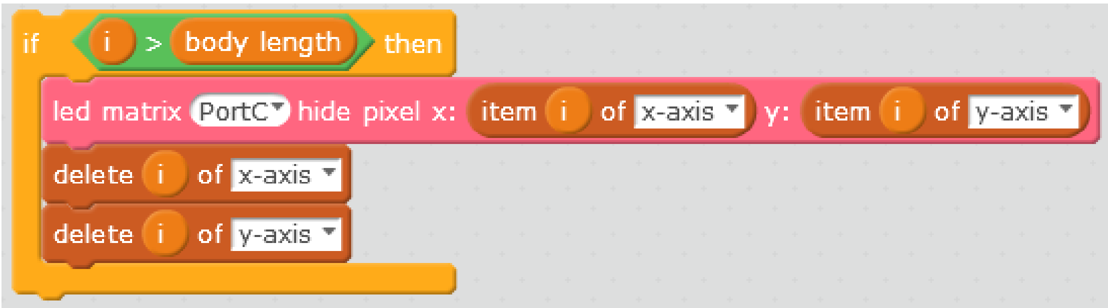


FIG. 15-24

When the program is running for the third time, the value of “i” is 3, which does not meet the judgment that shall be larger than the body length. The displayed conditions of the LED panel module are as shown in FIG. 15-25. When the program is run for the fourth time, the contained values in the lists x-axis and y-axis are as shown in FIG. 15-26. At the moment, the value of “i” is 4, which meets the judgment that shall be larger than the body length. Then extinguish the light of the coordinate (0, 0), which represents the fourth value of the list "x-axis" and "y-axis," that is, extinguish the light of the snake tail. Then lighten the light in the position after the snakehead moves. The change process is as shown in FIG. 15-27. At the moment, the values at the end of list "x-axis" and "y-axis" still are the coordinates of the snake tail lights, as shown in FIG.

15-28.

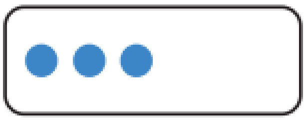


FIG. 15-25

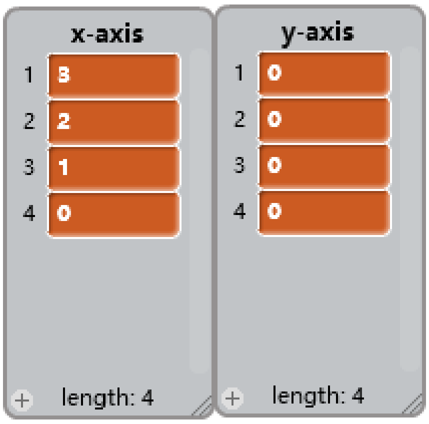


FIG. 15-26

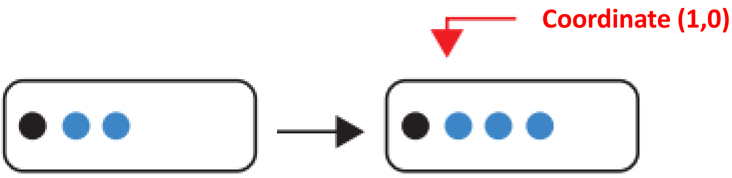


FIG. 15-27

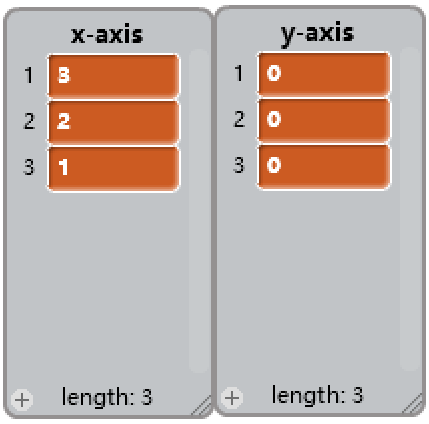


FIG. 15-28

Note: The program is challenging. Teachers can explain the program to students through a number pick and combine the change process of the snake body.

After completing the above procedures, the total program will be as shown in FIG. 15-29.

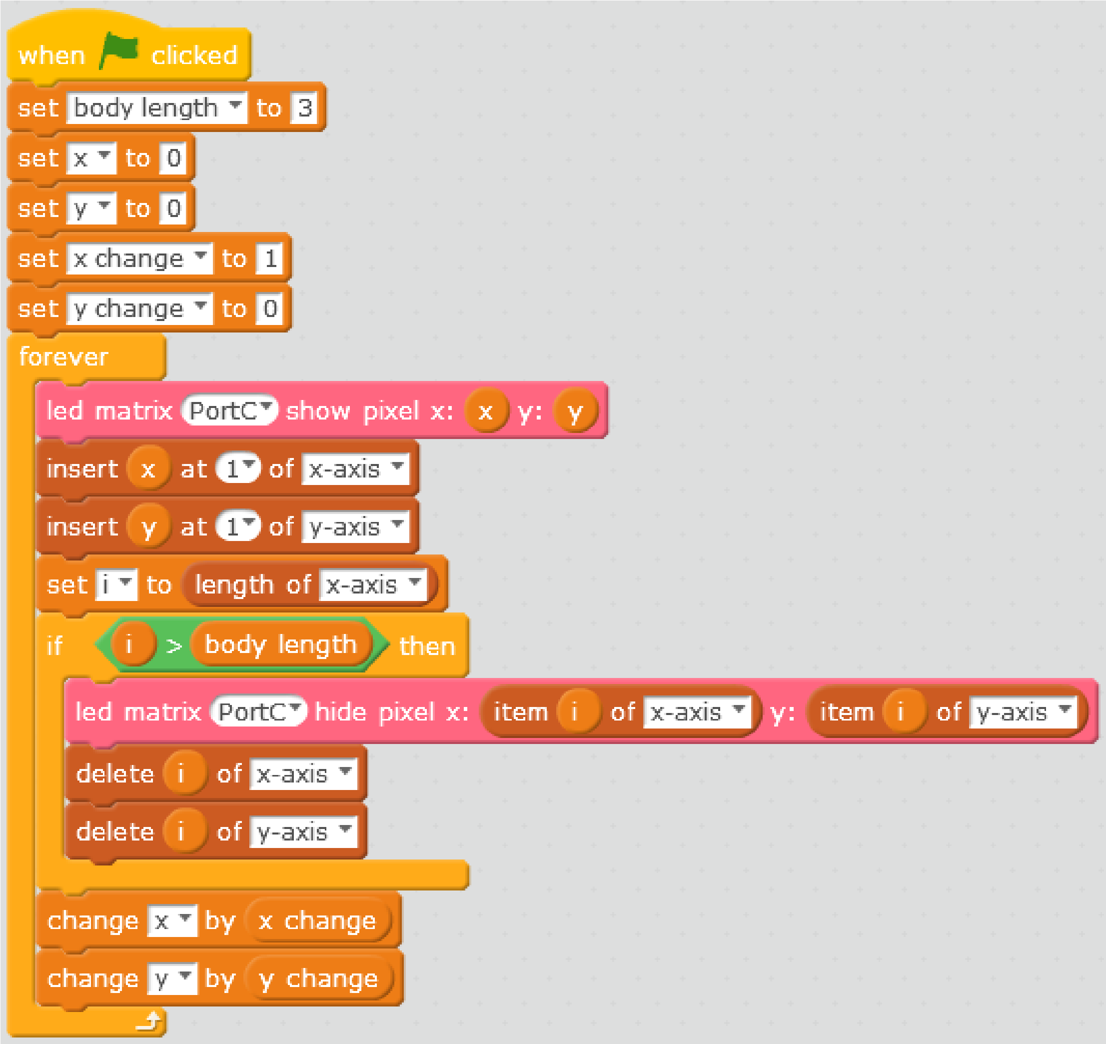


FIG. 15-29

Program effect: The snake moves to the right and maintains the body length---3. However, repeat running the program, it can be found that the initial length of the snake changes. It is because we have not initialized the list, namely to reset the list contents. Its program is as shown in FIG. 15-30 and the modified total program is as shown in FIG. 15-31.

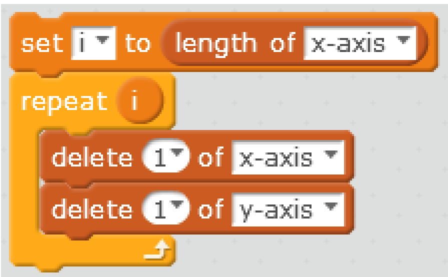


FIG. 15-30

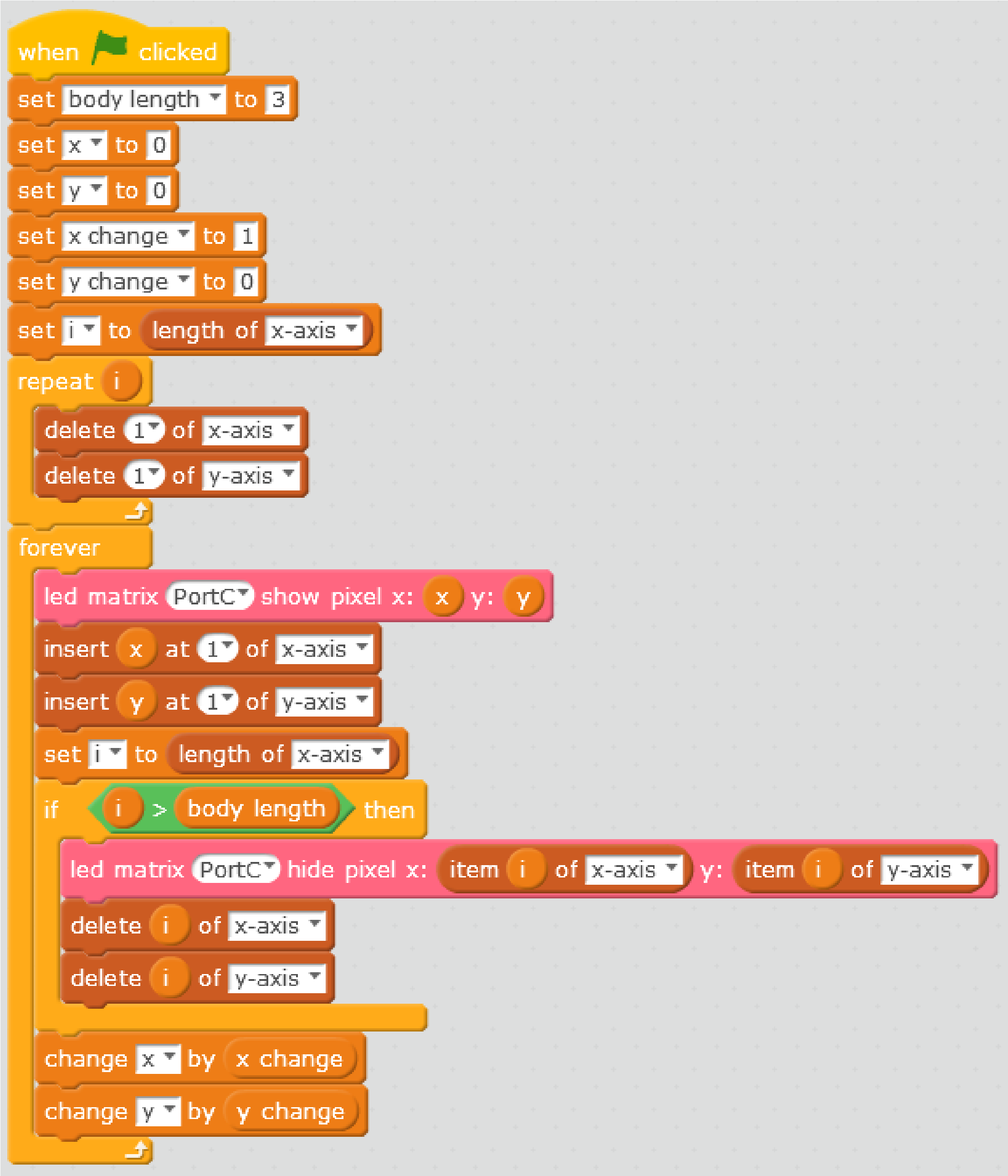


FIG. 15-31

3. Write the Remote-Control Program

In the gluttonous snake game, we will change the movement direction of the snake through remote control. From the above program effect, we can know that when the snake moves to the right, the x-axis will increase, and the y-axis will remain the same. In the same way, it can be concluded that when the snake moves to the left, the x-axis will decrease, and the y-axis will remain the same. When the snake moves downwards, the x-axis will remain the same, and the y-axis will increase. When the snake moves upwards, the x-axis will remain the same, and the y-axis will decrease. The program is as shown in FIG. 15-32.

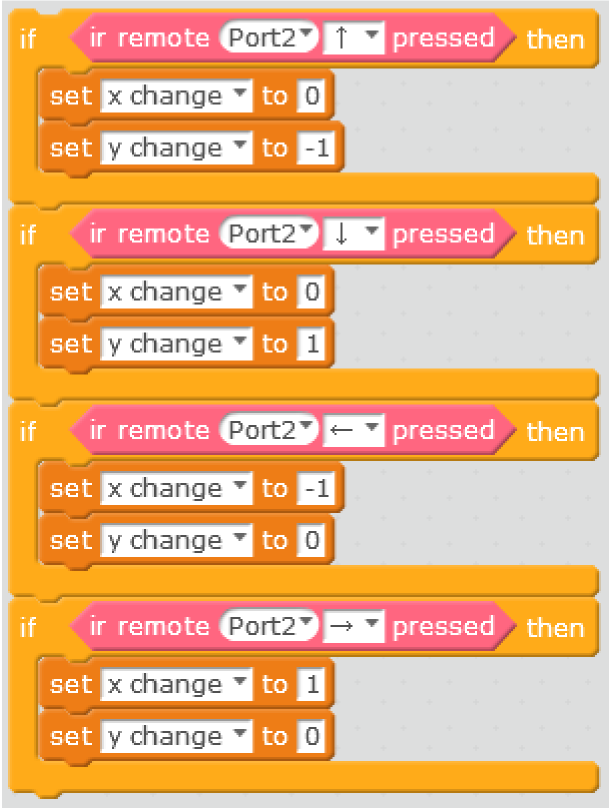


FIG. 15-32

Note: There will be some problem to write the program like this, namely press the "↑" key when the snake is moving downwards, the snake will change the direction. However, it is not allowed to operate like this in the actual games. The solutions will be explained in the classroom development. The logic is very complicated and will not be explained here.

After the above procedures are completed, the total program will be as shown in FIG. 15-33. At this moment, the remote control can be used to control the movement of the snake in each direction.

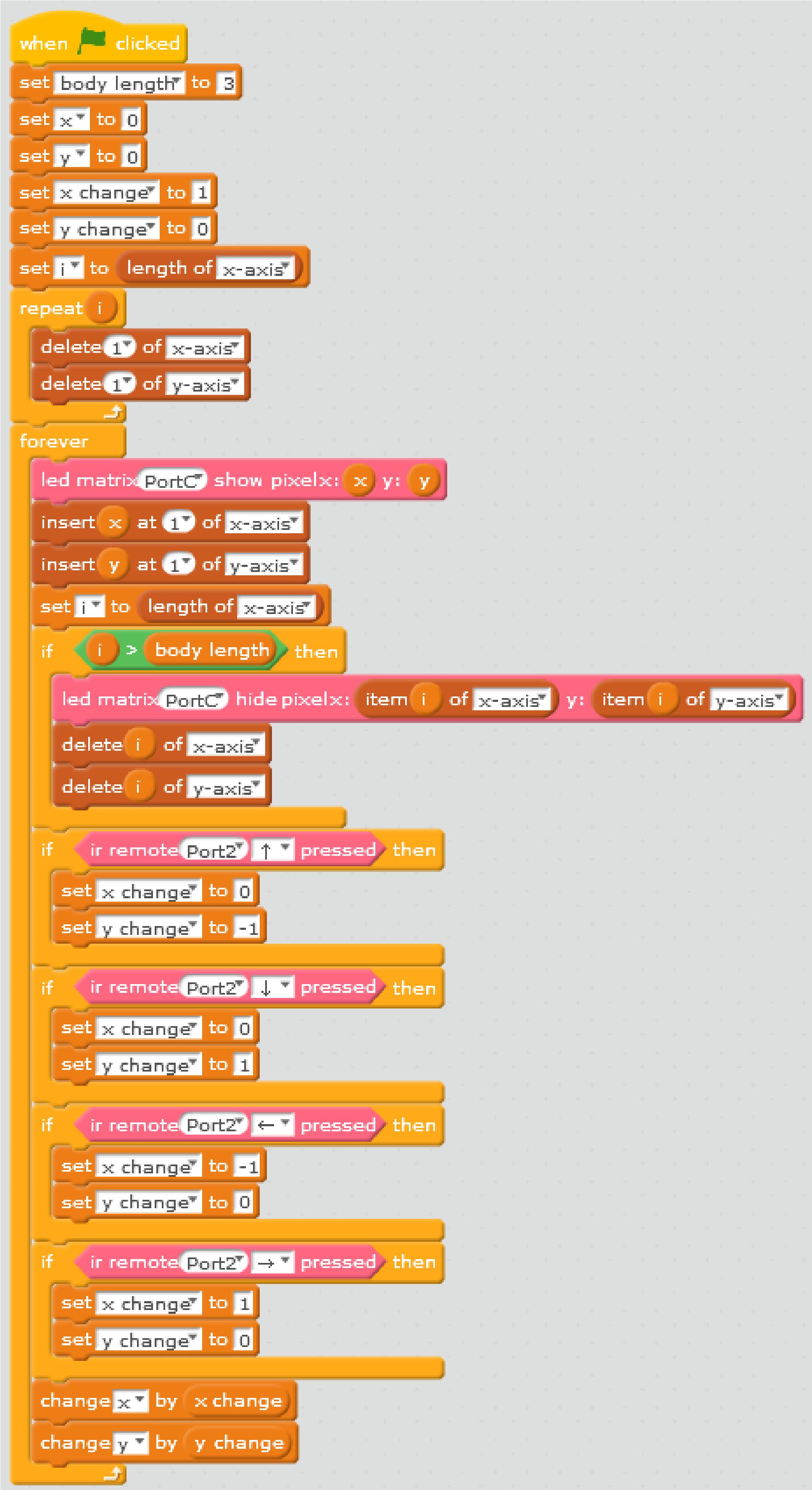

when clicked
set body length to 3
set x to 0
set y to 0
set x change to 1
set y change to 0
set i to length of x-axis
repeat i
delete 1 of x-axis
delete 1 of y-axis
forever
led matrix PortC show pixelx: x y: y
insert x at 1 of x-axis
insert y at 1 of y-axis
set i to length of x-axis
if i > body length then
led matrix PortC hide pixelx: item i of x-axis y: item i of y-axis
delete i of x-axis
delete i of y-axis
if ir remote Port2 ↑ pressed then
set x change to 0
set y change to -1
if ir remote Port2 ↓ pressed then
set x change to 0
set y change to 1
if ir remote Port2 ← pressed then
set x change to -1
set y change to 0
if ir remote Port2 → pressed then
set x change to 1
set y change to 0
change x by x change
change y by y change

FIG. 15-33

Note: the blocks of changing the x and y value need to be placed after the judgment of the remote control; otherwise the operation will be delayed.

4. Set the food position and the changes of the snake after eating.

a. Set variable "food x" and "food y" to represent the x and y-axis value of the food appearing and combine the random number blocks to generate the first food in the random position of the LED panel module. The program is as shown in FIG. 15-34.

FIG. 15-34

Note: The start position of the food cannot overlap with the snake, so the range of the variable "food x " is 3 ~ 20 but not 0 ~ 20. In the same way, the range of the variable "food y" is 1 ~ 6 but not 0 ~ 6.

b. Through the judgment, whether the x and y-axiss of the snakehead are equal to the x and y-axiss of the food at the same time; if equal, it means that the snake has eaten the food, the body length gets longer, and the chain table length is longer. The program is shown in FIG. 15-35.

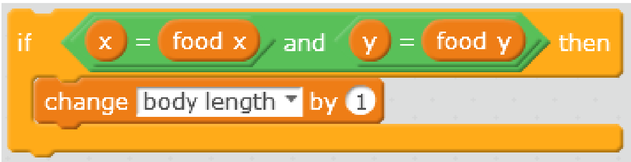


FIG. 15-35

c. (Challenge) If the snake eats the first food, the next food shall appear randomly on the screen, but not in the body of the snake. If it appears on the body of the snake, the new food will be regenerated, as shown in FIG. 15-36.

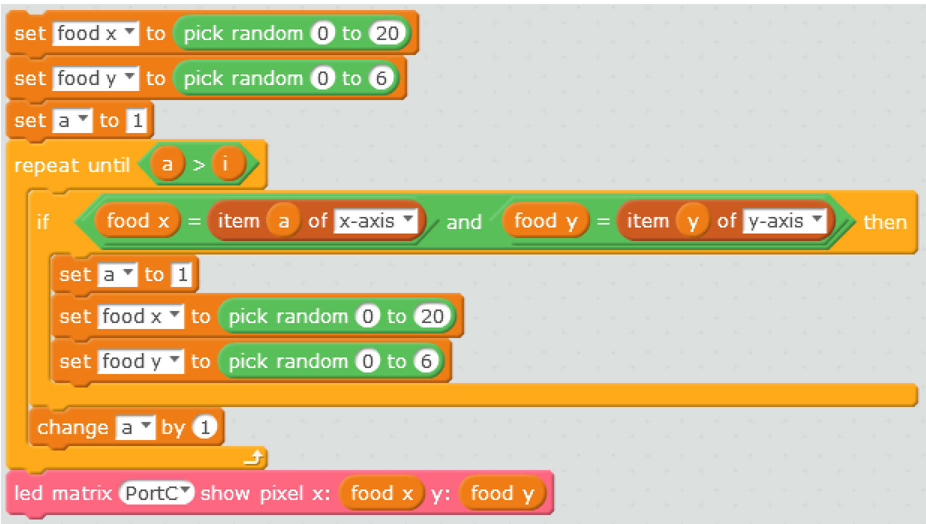


FIG. 15-36

From the game demand, we can know when the coordinate values of a new food are generated, it will need to be compared with all the values of the “x-axis” and “y-axis” of the list, to determine whether the food is on the snake. If The x and y value in a certain item in two lists are equal to the value of “food x” and “food y” at the same time, it proves that the generated food is on the snake and new food need to be generated. Here we will compare the x and y-axis of the food with the value of item 1 in the two list by creating variable “a” and initialize the variable to “1”. If unequal, increase the variable “a” and then compare it with the value of next item of the two list, which will be repeated until the value of variable “a” is larger than the list length. If there is an equal case in the comparison process, it will generate new food coordinate and initialize the value of “a," to be compared with the value of the two list again.

Note: The block “[list] contains [thing]” cannot be used here. That’s because even if the two lists contain the values equal to the variable “food x” and variable “food y”, it can not prove the food is on the snake.

5. Add the fail judgment of the snake touching the wall.

By judging the value of the x-axis and the y-axis of the snakehead to determine whether the snake encounters the wall: if the x-axis of the snakehead is less than 0 or greater than 20, the snakehead exceeds the screen scope, and the game ends. Similarly, when the y-axis of the snakehead is less than 0 or greater than 6, the game will end. The program is as shown in FIG. 15-37.

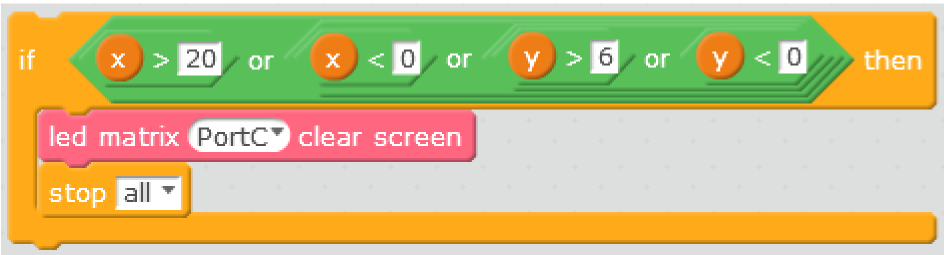


FIG. 15-37

After the above procedures are completed, the total program is as shown in FIG. 15-38-1 and 15-38-2. The game program of the gluttonous snake ends basically.

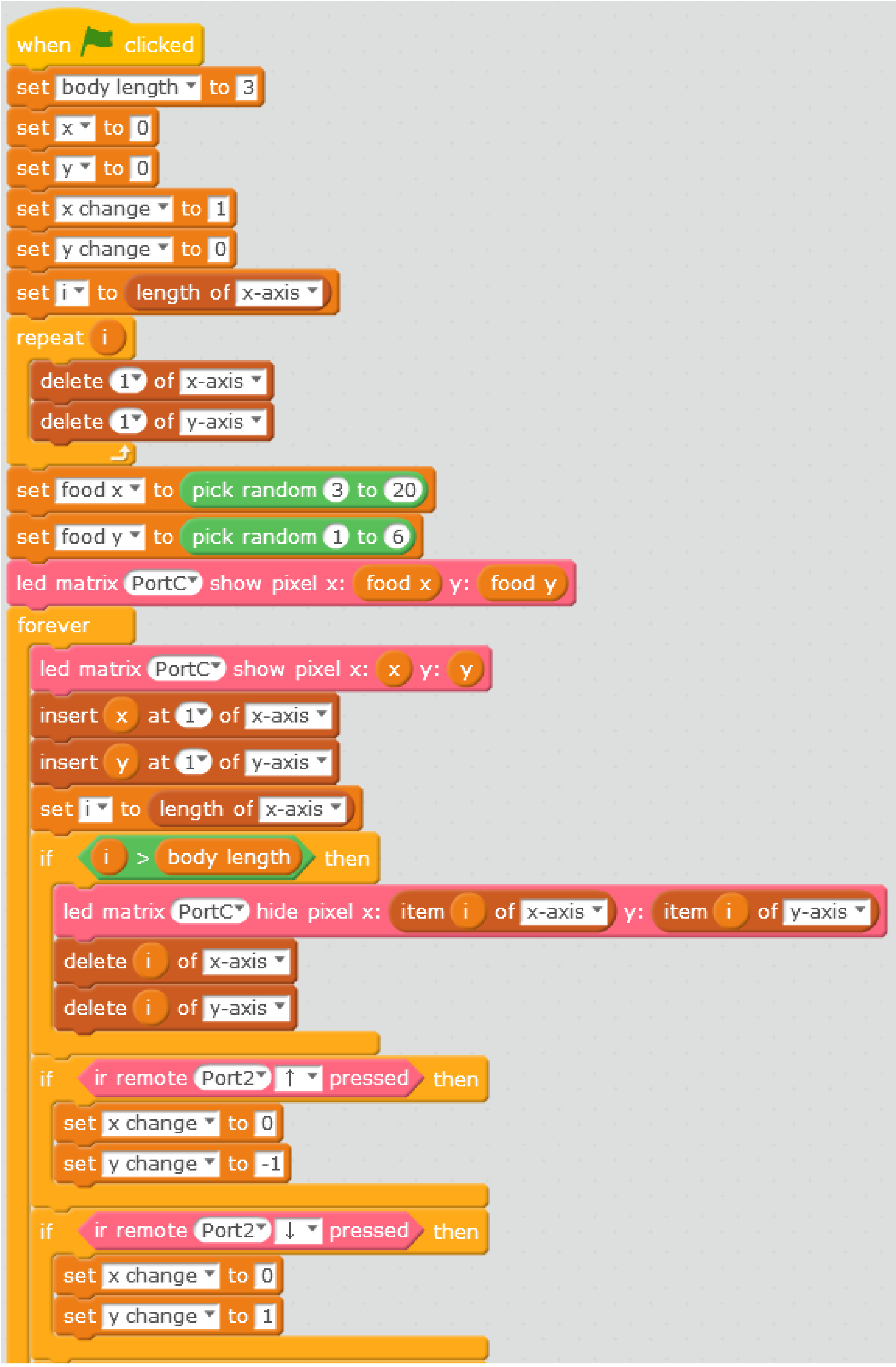

when clicked
set body length to 3
set x to 0
set y to 0
set x change to 1
set y change to 0
set i to length of x-axis
repeat i
delete 1 of x-axis
delete 1 of y-axis
set food x to pick random 3 to 20
set food y to pick random 1 to 6
led matrix PortC show pixel x: food x y: food y
forever
led matrix PortC show pixel x: x y: y
insert x at 1 of x-axis
insert y at 1 of y-axis
set i to length of x-axis
if i > body length then
led matrix PortC hide pixel x: item i of x-axis y: item i of y-axis
delete i of x-axis
delete i of y-axis
if ir remote Port2 ↑ pressed then
set x change to 0
set y change to -1
if ir remote Port2 ↓ pressed then
set x change to 0
set y change to 1

FIG. 15-38-1

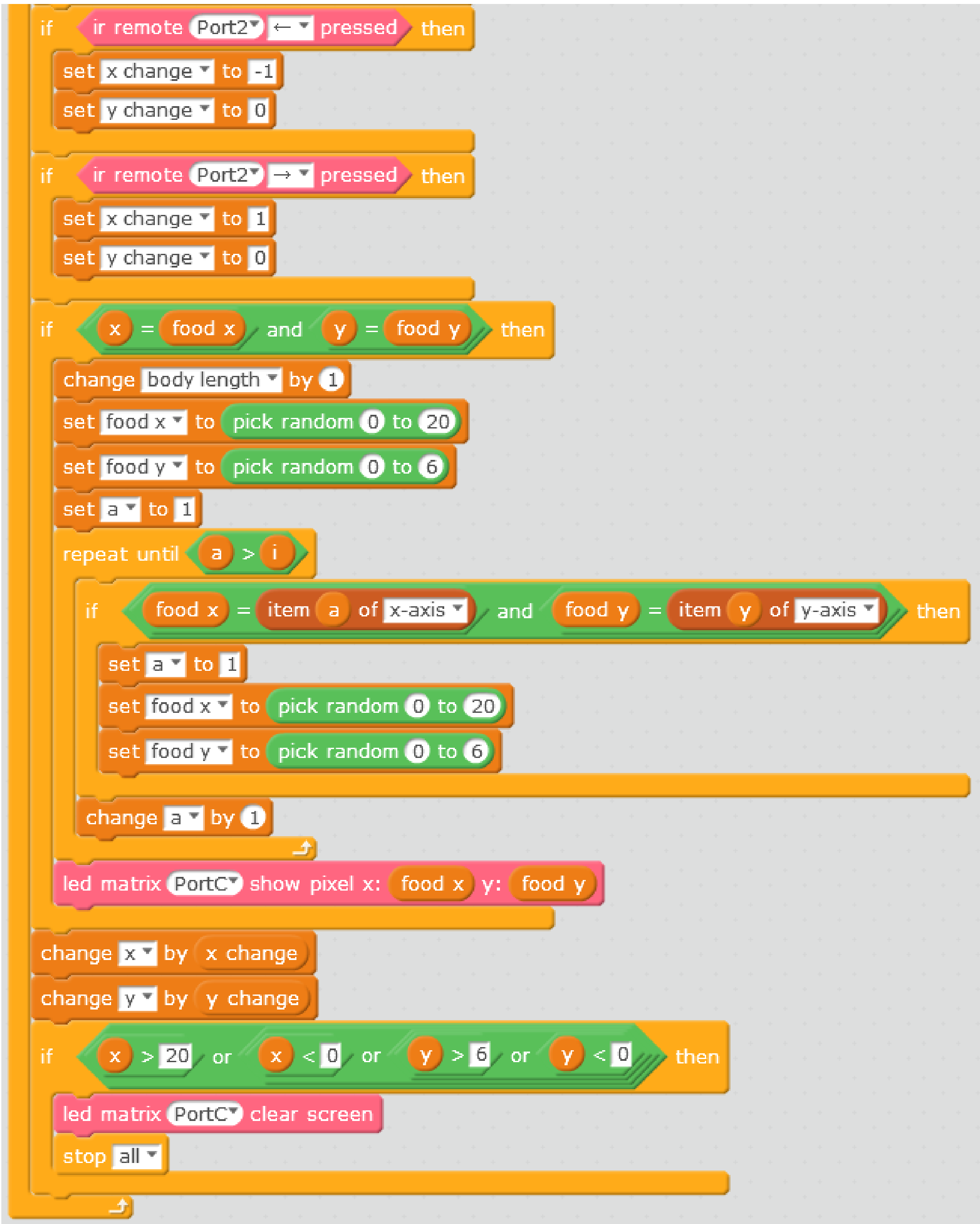


FIG. 15-38-2

Note:

a. The program game will not end if the snakehead encounters snake body. Students can develop the function by themselves by referring to the judgment methods of eaten food.

b. This program is online codes. To use offline programming, students should use block “WeeeBot Program” instead of the block “When green flag clicked” as shown in FIG. 15-39 and add block “wait 0.2 secs” in the end of program. As program is running faster in offline mode than in online mode, the speed of snake would be too fast to play game without a delay.

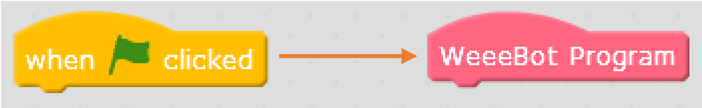


FIG. 15-39

Ⅳ Classroom Development and Classroom Evaluation (10min)

1. Classroom development: add the judgment of remote control key limit (selected part according to class time).

Run the program shown in FIG. 15-38, we can know when the snakes move downwards, press the "↑" key, the snake will immediately change the direction, which does not conform to the actual game control. We will need to add the judgment when changing the direction: If the "↑" key is pressed and the snake does not move downwards (“y change” is not 1), the movement direction can be changed to upwards. The control method of other direction is in the same way. The modified program of the remote-control part is as shown in FIG. 15-40.

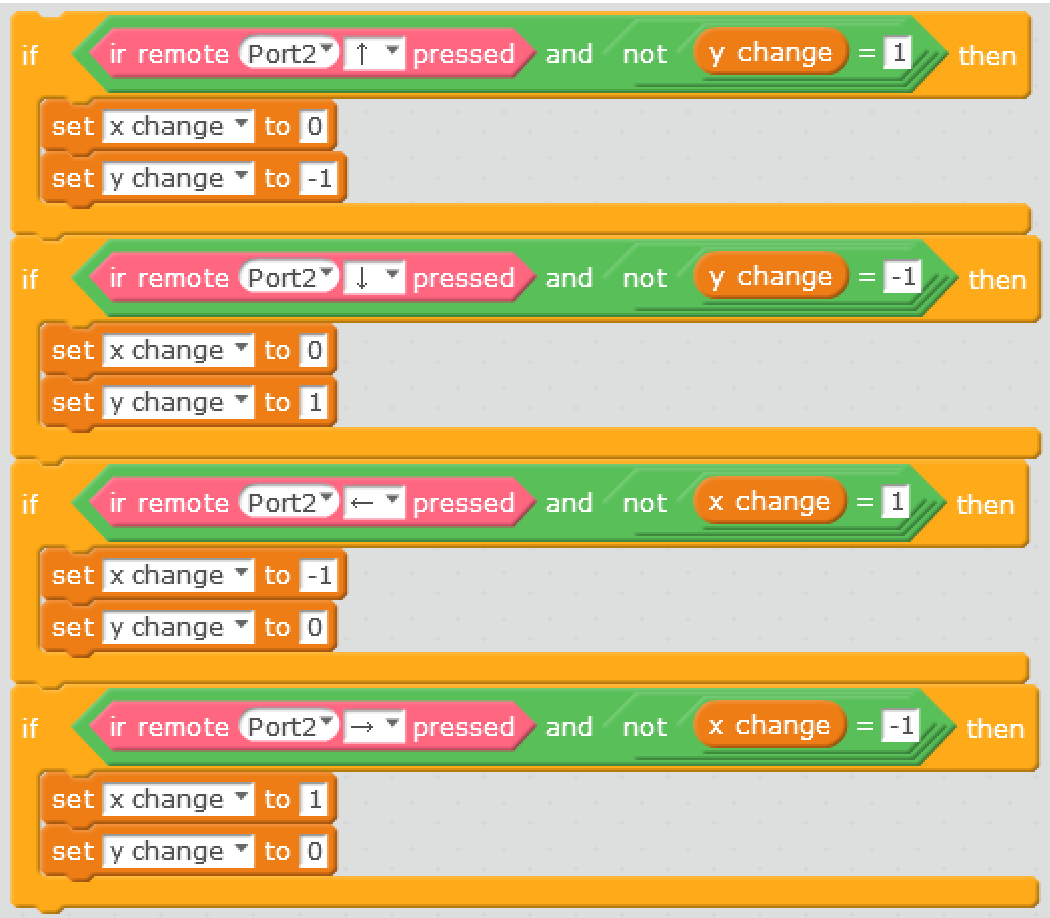


FIG. 15-40

2. Classroom evaluation: students' class performance will be evaluated according to the evaluation criteria.

Evaluation criteria:

a. Able to understand the concept of lists and be proficient in the use of list functions.

b. Able to make gluttonous snake games.

Chapter 16 WeeeBot Robot Challenges

Summary

As the last lesson of the robot course, this lesson will combine the hardware knowledge points we learned previously to control the robot programmatically to challenge the game, inspect students' learning outcomes in the funny competitions, and enhance students' learning interest in robots.

Learning Objectives

Review the knowledge points learned previously, strengthen the mastery of the old knowledge points, and use them to produce robots to complete the challenges.
Cultivate students' team spirit and teamwork ability during the competition, and build up the students' practical ability, program debugging ability and on-spot problem-solving ability.

Course Time

90 minutes.

Pre-class Preparation

WeeeBot robots, computers with WeeeCode software installed, USB cable, remote controls, and competition maps.

Teaching Process

Ⅰ Knowledge review (10min)

1. Review the control blocks of hardware modules learned in this semester and strengthen the mastery of hardware control.

Briefly review the control blocks of the hardware modules learned previously, as shown in FIG. 16-1, and deepen the students' impression on the control blocks of hardware modules.

Hardware name	Control blocks	The role of blocks
RGB LED	RGB on board red 0 green 0 blue 0	Control the RGB LED to display different colors.
Buzzer	play tone on note C4 beat Half	Control the buzzer to make the tone and beat of the sound.
Motor	set motor M1 speed 0	Control the movement and

	run forward ▾ at speed 0▾	stopping of the motor.
IR remote control	ir remote Port2▾ A ▾ pressed	Judge the detected IR signals of the IR receiver module.
RGB ultrasonic sensor	ultrasonic sensor PortB▾ distance ultrasonic sensor PortB▾ RGB led all ▾ red 0▾ green 0▾ blue 0▾	Send back the measured distance value of the RGB ultrasonic sensor and control the displayed colors of the RGB ultrasonic sensor RGB light
Sound sensor	sound sensor Port3▾	Send back the sound value of the sound sensor in the environment.
Light sensor	light sensor Port1▾	Send back the light value of the light sensor in the environment.
Line-following sensor	line follower PortA▾ S1▾	Send back the reflected IR value received by a specific line-following sensor in the line-following sensor.
LED Panel Module	led matrix PortC▾ show number 100 led matrix PortC▾ x: 0 y: 0 show string Hi led matrix PortC▾ show time 12 : ▾ 30 led matrix PortC▾ show pixel x: 0 y: 0 led matrix PortC▾ hide pixel x: 0 y: 0 led matrix PortC▾ clear screen	Control the display of the lights on the LED matrix and display specific information.

FIG. 16-1

Ⅱ Challenge Competition (60min)

1. Competition Rules

a. Competition maps: There are three passes in the competition. The first is remote control competition, in which students can use the remote controls to manipulate the robots. The second is line-following competition, in which the robots will need to walk out the area along the trajectory on their own. The third is obstacle avoiding competition, in which the robots will need to avoid the obstacles and walk out of the site on their own depending on the RGB ultrasonic sensor.

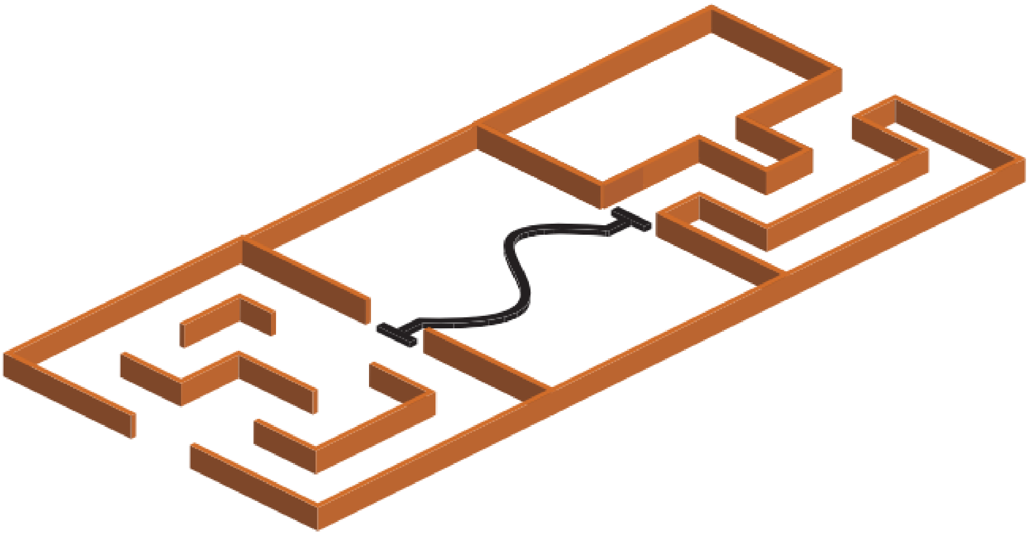


FIG. 16-2

Note:

① The map is only for reference. Teachers can set the map difficulty by themselves according to the students' mastery of the knowledge.

②electrical tapes stuck to the floor can be used to represent the wall in the remote control area, and wooden panel or books can be used to represent the wall in the obstacle avoiding the area.

b. Competition mode:

①The challenge competition is a team competition, members in each team is three people, a robot per person, everyone is responsible for the robot programming of a pass.

②The competition adopts the relay race forms: In the official game, the first player will put its robot on the entrance to challenge through remote control. When the first robot moves to the start of the second pass, the first player will need to pick up the robot and the second player shall place his robot on the entrance of the second pass to challenge. When the second robot patrol to the start of the third pass, the second player needs to pick up the robot and the third player shall place his robot on the start of the third pass began to challenge. The game ends when the third robot avoids the obstacle and reaches the destination.

c. Competition rules:

①Before the competition starts, the referee will take all the robots back and place them in the designated position.

②At the beginning of the competition, each team will take back their robots and begin to programme, modify and debug programs on the spot for 30 minutes. When programming, teams are not allowed to communicate and discuss with each other. Members of the team can communicate with each other. After 30 minutes, the robot will be taken back and put in the designated position.

③Teachers determine the order of the competition. The team will get their robots in order and begin the competition. They need to state their team name and the name of members and finish the game in the specified 3 minutes. The referee will record the scores.

d. Game scoring

①The task competition is divided into three passes. The first pass includes 30 points. In the first pass, the robot will be deducted 5 points every time it walks on the line in the walking process by remote control, and the first pass task of the robot fails if negative scores appear. The second pass includes 30 points. In the second pass, the robot is required to track the line all the time. In the line-following process, if the robot body leaves the black line for more than 2 seconds, the first pass task fails. The first pass includes 40 points. The robot will be deducted 5 points every time it touches the wall in the walking process, and it will be deducted 10 points if it fails to reach the destination. The third pass task of the robot fails if negative scores appear.

② In the competition, if students find the robots cannot finish the task of the corresponding pass, they can give up the challenge and get no points in this pass.

③ The three tasks must be completed in 3 minutes. If the tasks are not completed in 3 minutes, it will be determined as task failure. If the tasks are completed in 3 minutes, the corresponding time and scores will be recorded.

④ The competition results will be ranked according to the scores from higher scores to lower scores. If two teams have the same scores, the team that spends shorter time will be ranked higher.

⑤It is not allowed to interfere with the other team's robot to complete the task in the competition process. Once found, the scores of the team will be canceled.

⑥After started, the robot must complete the task by itself in the game. No human help is allowed. Otherwise, no score is included.

⑦If there are teams borrowing robots from each other, their team scores will be canceled.

e. Referee:

①There will be two referees in each game to execute referee work. One is technology referee, responsible for recording robot challenge scores. The other is a timing referee, responsible for recording robot challenge times. Moreover, the technology referee will also be responsible for signing up the form as shown in FIG.16-3. The referees can be selected from the students.

Registration form of competition results						
Team Name	Team Member	Scores in the first pass	Scores in the second pass	Scores in the third pass	Total scores	Competition time

FIG.16-3

②Teachers are the chief referees in the competition and are responsible for:

a) Ensuring that the team and the robot competitors will correspond.

a. Supervising all fouls and record them.

C. Maintaining the discipline at the competition site to ensure the smooth conduct of the competition.

2. The programming of Challenge Robot

Note: Teachers can help students to sort out the difficulties to pay attention to in each pass before the competition.

a. The reference program of the remote-controlled robot in the first pass is as shown in FIG. 16-4. The program difficulties are as follows:

① The use of branch structure.

② The robot needs to stop moving when the key of the remote control is not pressed.

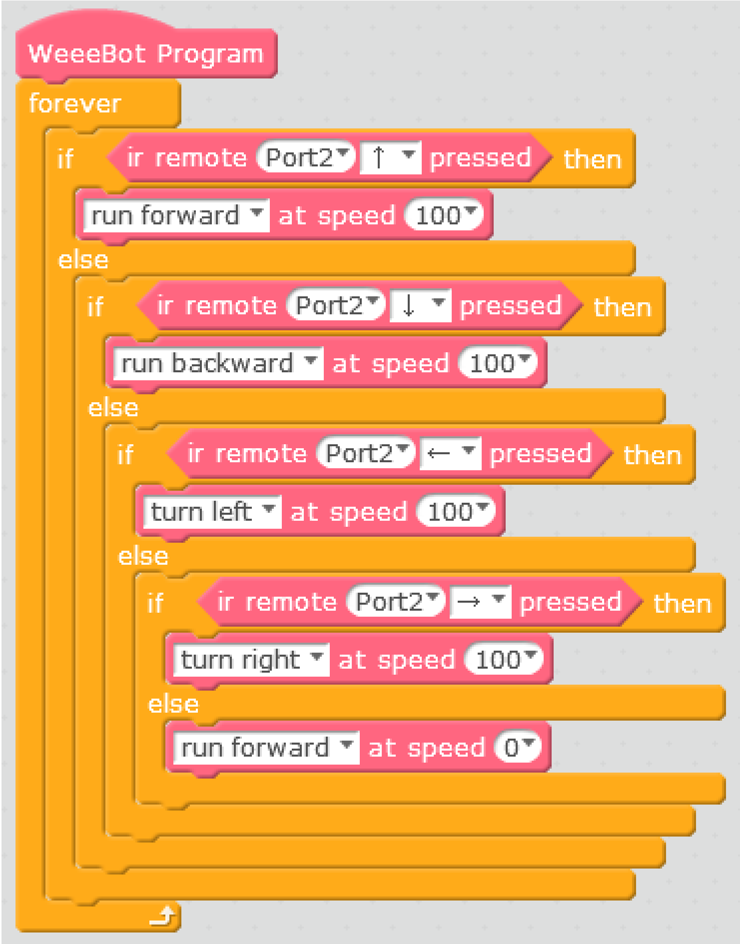


FIG. 16-4

b. The line-following reference program of the second pass is as shown in FIG. 16-5(The example program is the "black line clamps line-following sensor." The program difficulties are as follows:

① The judgment method of compound conditions.

②The selection and use of the line-following algorithm.

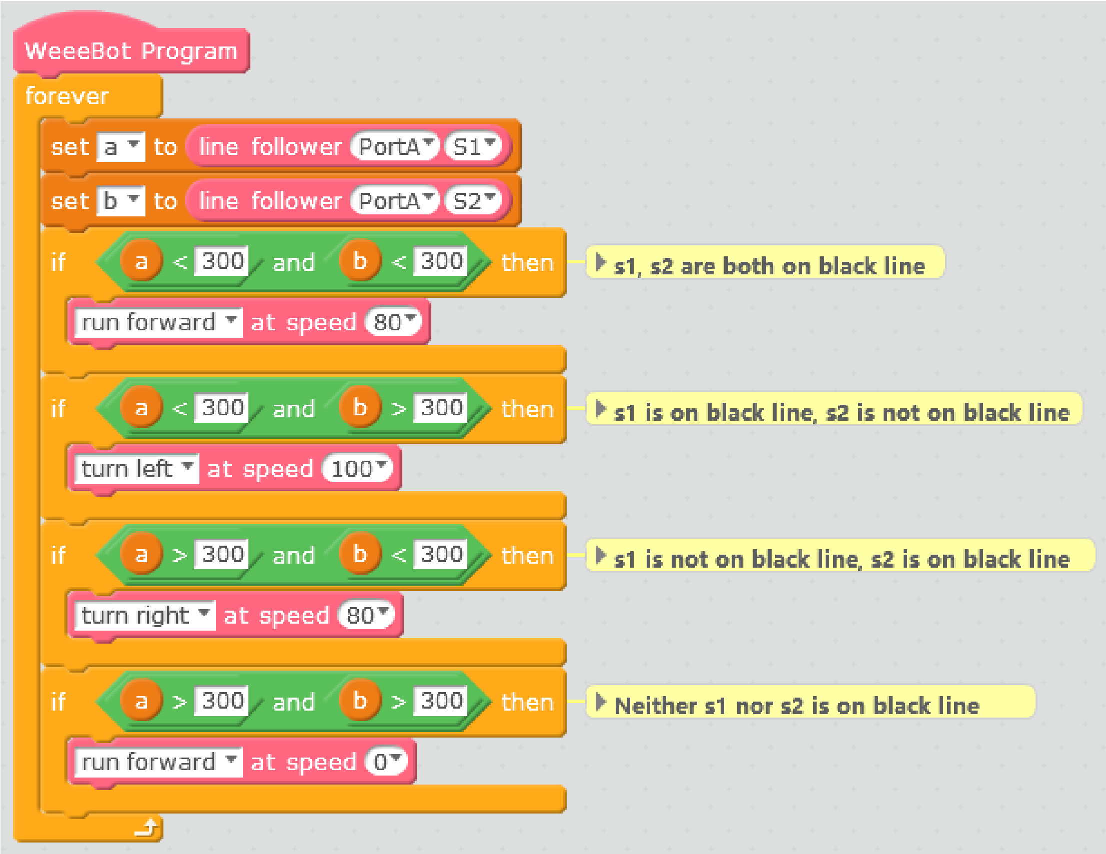


FIG. 16-5

c. The reference program of the obstacle avoiding robot in the third pass is as shown in FIG. 16-6. The program difficulties are as follows:

① Debugging of the swerving time: If you cannot adjust the swerving Angle to 90 °, the robot to bump into the wall easily.

② The use of "Repeat until" block: As the swerving angle of the four corners in the route is different, the robot can not just turn left and turn right. It must go straight to the first turn (the ultrasonic detection value is less than 10cm.) and turn right. Then go straight until the second turn and turn left. Then go straight until the third turn and turn left. Then go straight until the fourth turn and turn right.

FIG. 16-6

Ⅲ Competition award (10min)

Teachers are responsible for maintaining the game order on the spot in the challenge process of the students, and the referees are responsible for recording the game scores. After the completion of the game, teachers will rank the students according to the game results and give corresponding awards to the team with outstanding achievements.

Ⅳ Evaluation and summary (10min)

Teachers can sum the classroom evaluation scores of each student in this semester, Comment on each student properly, award the excellent performers, and at the same time encourage all students to continue learning robots after class to cultivate their logical thinking ability.