Welcome

Thank you for choosing Freenove products!

Getting Started

When reading this, you should have downloaded the ZIP file for this product.
Unzip it and you will get a folder containing tutorials and related files. Please start with this PDF tutorial.

! Unzip the ZIP file instead of opening the files in the ZIP file directly.
! Do not move, delete or rename files in the folder just unzipped.

Get Support

Encounter problems? Don't worry! Refer to “TroubleShooting.pdf” or contact us.

When there are packaging damage, quality problems, questions encountering in use, etc., just send us an email. We will reply to you within one working day and provide a solution.

support@freenove.com

Safety and Precautions

Please follow the following safety precautions when using or storing this product:

- Keep this product out of the reach of children under 6 years old.
- This product should be used only when there is adult supervision present as young children lack necessary judgment regarding safety and the consequences of product misuse.
- This product contains small parts and parts, which are sharp. This product contains electrically conductive parts. Use caution with electrically conductive parts near or around power supplies, batteries and powered (live) circuits.
- When the product is turned ON, activated or tested, some parts will move or rotate. To avoid injuries to hands and fingers, keep them away from any moving parts!
- It is possible that an improperly connected or shorted circuit may cause overheating. Should this happen, immediately disconnect the power supply or remove the batteries and do not touch anything until it cools down! When everything is safe and cool, review the product tutorial to identify the cause.
- Only operate the product in accordance with the instructions and guidelines of this tutorial, otherwise parts may be damaged or you could be injured.
- Store the product in a cool dry place and avoid exposing the product to direct sunlight.
- After use, always turn the power OFF and remove or unplug the batteries before storing.

About Freenove

Freenove provides open source electronic products and services worldwide.

Freenove is committed to assist customers in their education of robotics, programming and electronic circuits so that they may transform their creative ideas into prototypes and new and innovative products. To this end, our services include but are not limited to:

- Educational and Entertaining Project Kits for Robots, Smart Cars and Drones
- Educational Kits to Learn Robotic Software Systems for Arduino, Raspberry Pi and micro: bit
- Electronic Component Assortments, Electronic Modules and Specialized Tools
- **Product Development and Customization Services**

You can find more about Freenove and get our latest news and updates through our website:

http://www.freenove.com

Copyright

Any concerns? ✉ support@freenove.com

Contents

Welcome..1

Contents..1

Prepare..2

Processing Software..2
First Use..5
Communication protocol..7

Chapter 1 Oscilloscope..9

Project 1.1 Oscilloscope..9

Chapter 2 Control 2D and 3D Figures..13

Project 2.1 Ellipse..13
Project 2.2 Box 3D..16

Chapter 3 Snake Game..18

Project 3.1 Snake Game..18
Project 3.2 Snake Game 3D..23

Chapter 4 Pick Apples..25

Project 4.1 Pick Apples..25

Chapter 5 Pong Game..29

Project 5.1 Pong Game..29
Project 5.2 Pong Game 3D..33

Chapter 6 Tetris..35

Project 6.1 Tetris..35

What's Next?..39

Prepare

Processing is an easy-to-use, free and open source software for writing graphical programs to run on a computer.

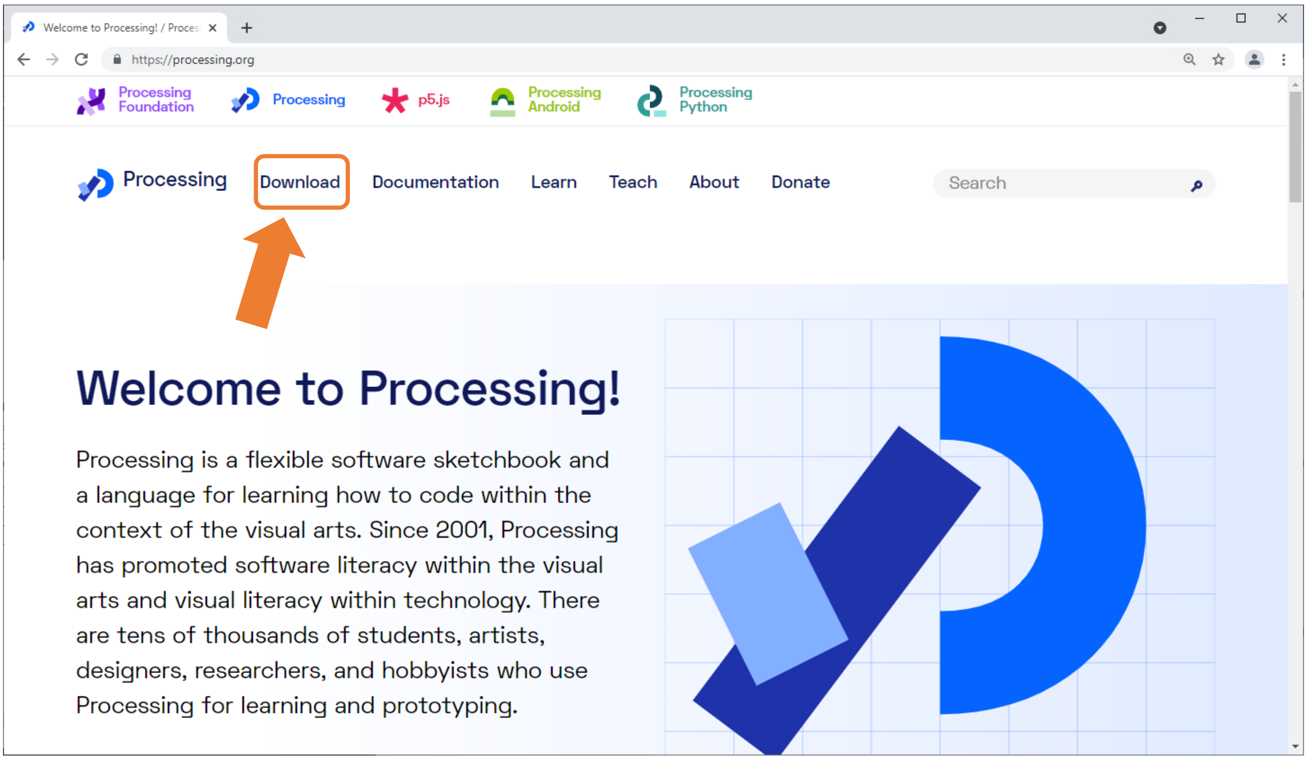
This document will show you how to use Processing to write programs to communicate with the control board. By doing this, we can make virtual instruments, game consoles and other projects.

Processing Software

Processing software / Processing Development Environment (PDE) makes it easy to write programs.

Processing software uses Java programming language by default. Do not worry even if you don't know Java, because we provide complete code. You can learn Java later if you are interested in it.

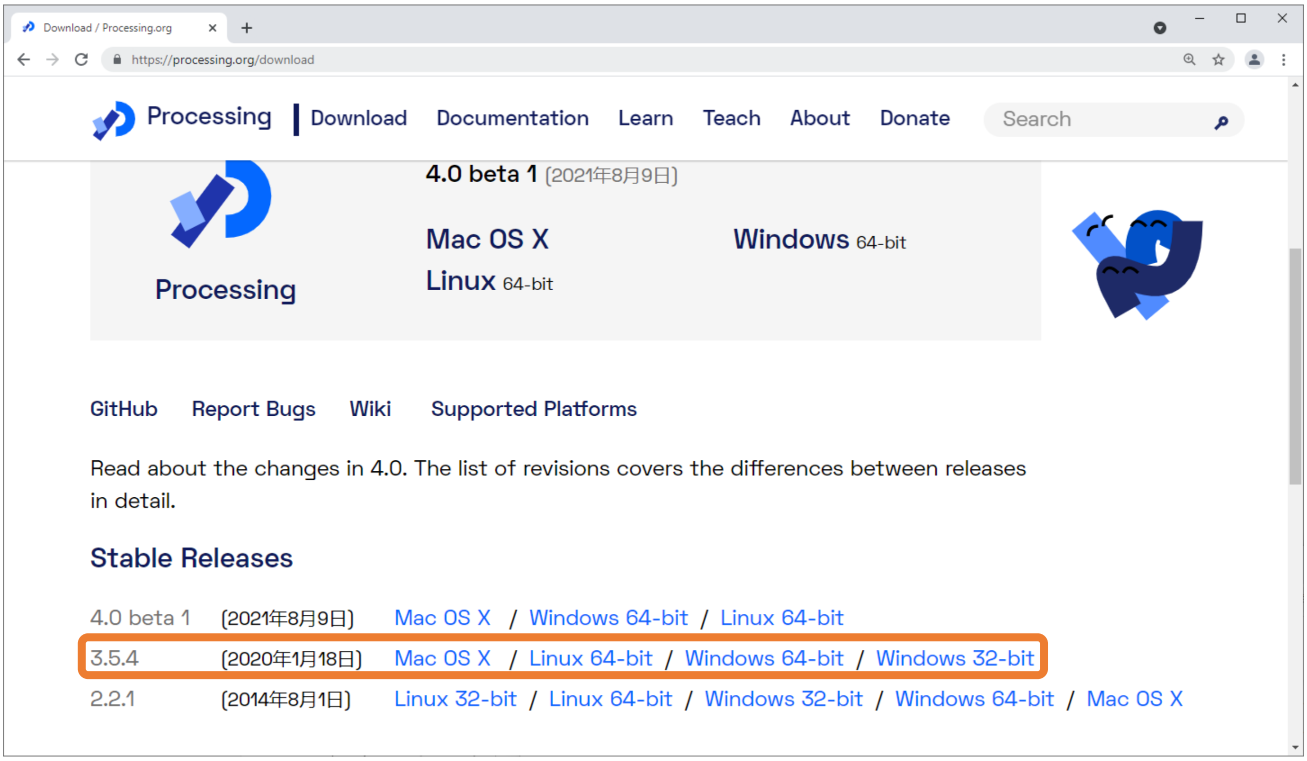
First, install Processing software. Visit https://processing.org/, click "Download" to enter the download page.



Welcome to Processing!

Processing is a flexible software sketchbook and a language for learning how to code within the context of the visual arts. Since 2001, Processing has promoted software literacy within the visual arts and visual literacy within technology. There are tens of thousands of students, artists, designers, researchers, and hobbyists who use Processing for learning and prototyping.

Select the Mac, Windows, or Linux version, depending on what machine you use.
In this tutorial, we use Version 3.5.4. You can also use 4.0 beta 1.



Read about the changes in 4.0. The list of revisions covers the differences between releases in detail.

Stable Releases

4.0 beta 1	(2021年8月9日)	Mac OS X / Windows 64-bit / Linux 64-bit
3.5.4	(2020年1月18日)	Mac OS X / Linux 64-bit / Windows 64-bit / Windows 32-bit
2.2.1	(2014年8月1日)	Linux 32-bit / Linux 64-bit / Windows 32-bit / Windows 64-bit / Mac OS X

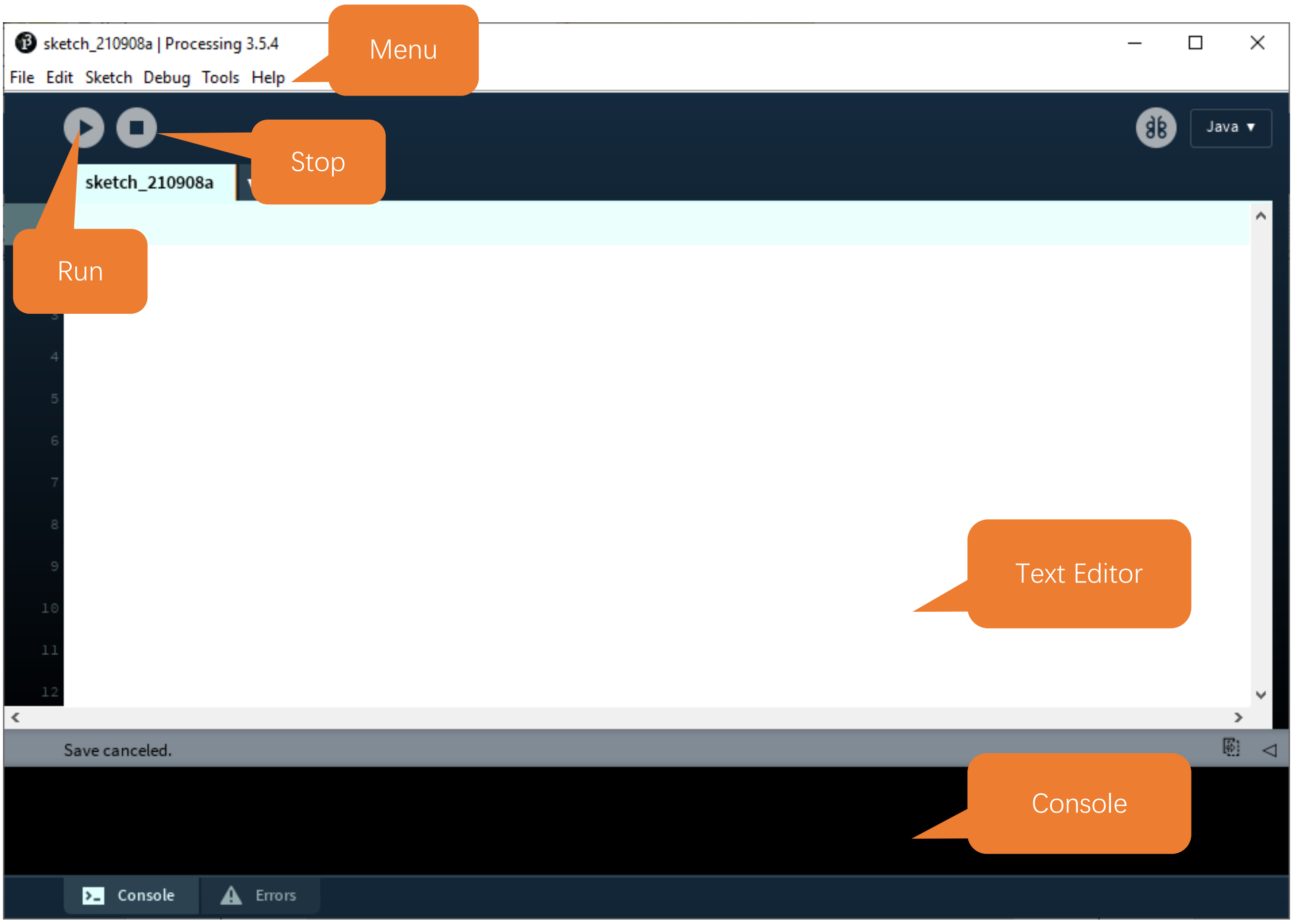
Installation on each machine is straightforward:

- On Windows, you'll have a .zip file. Double-click it, and drag the folder inside to a location on your hard disk. It could be Program Files or simply the desktop, but the important thing is for the processing folder to be pulled out of that .zip file. Then double-click processing.exe to start.

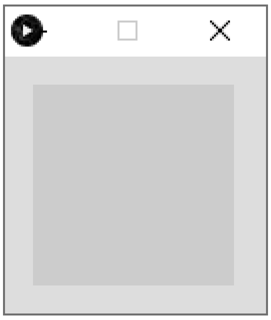
- The Mac OS X version is also a .zip file. Double-click it and drag the Processing icon to the Applications folder. If you're using someone else's machine and can't modify the Applications folder, just drag the application to the desktop. Then double-click the Processing icon to start.

- The Linux version is a .tar.gz file, which should be familiar to most Linux users. Download the file to your home directory, then open a terminal window, and type:
  tar xvfz processing-xxxx.tgz
  (Replace xxxx with the rest of the file's name, which is the version number.) This will create a folder named processing-2.0 or something similar. Then change to that directory:
  cd processing-xxxx
  and run it:
  ./processing

With any luck, the main Processing window will now be visible. Everyone's setup is different, so if the program didn't start, or you're otherwise stuck, visit the troubleshooting page for possible solutions.



Display Window:



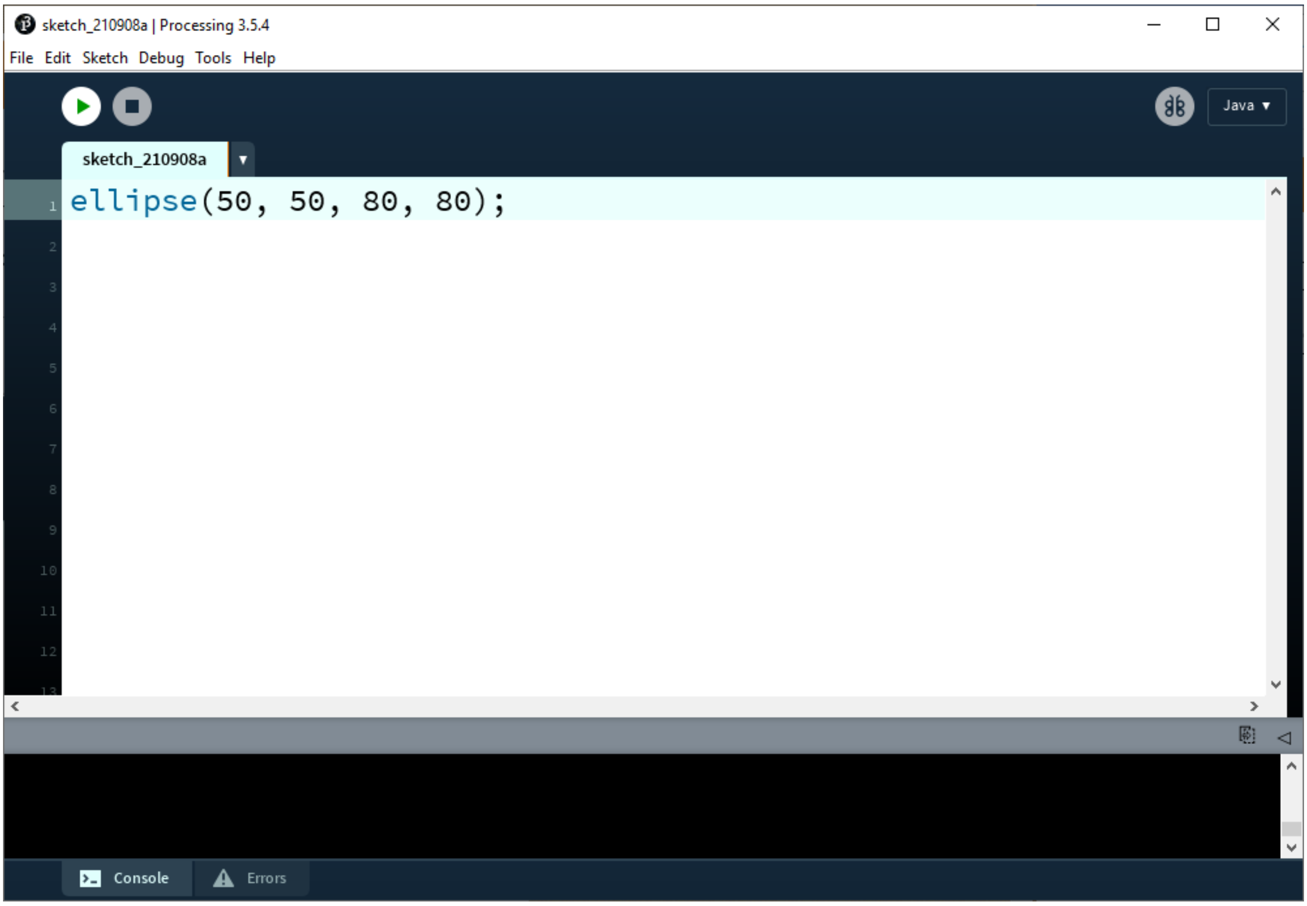
You're now running the Processing Development Environment (or PDE). There's not much to it; the large area is the Text Editor, and there's a row of buttons across the top; this is the toolbar. Below the editor is the Message Area, and below that is the Console. The Message Area is used for one line messages, and the Console is used for more technical details.

First Use

In the editor, type the following

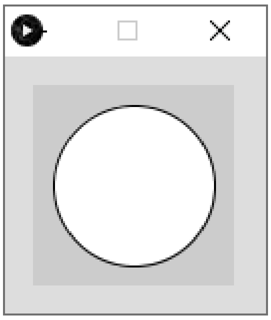
```
ellipse(50, 50, 80, 80);
```

This line of code means "draw an ellipse, with the center 50 pixels over from the left and 50 pixels down from the top, with a width and height of 80 pixels." Click the Run button (the triangle button in the Toolbar).

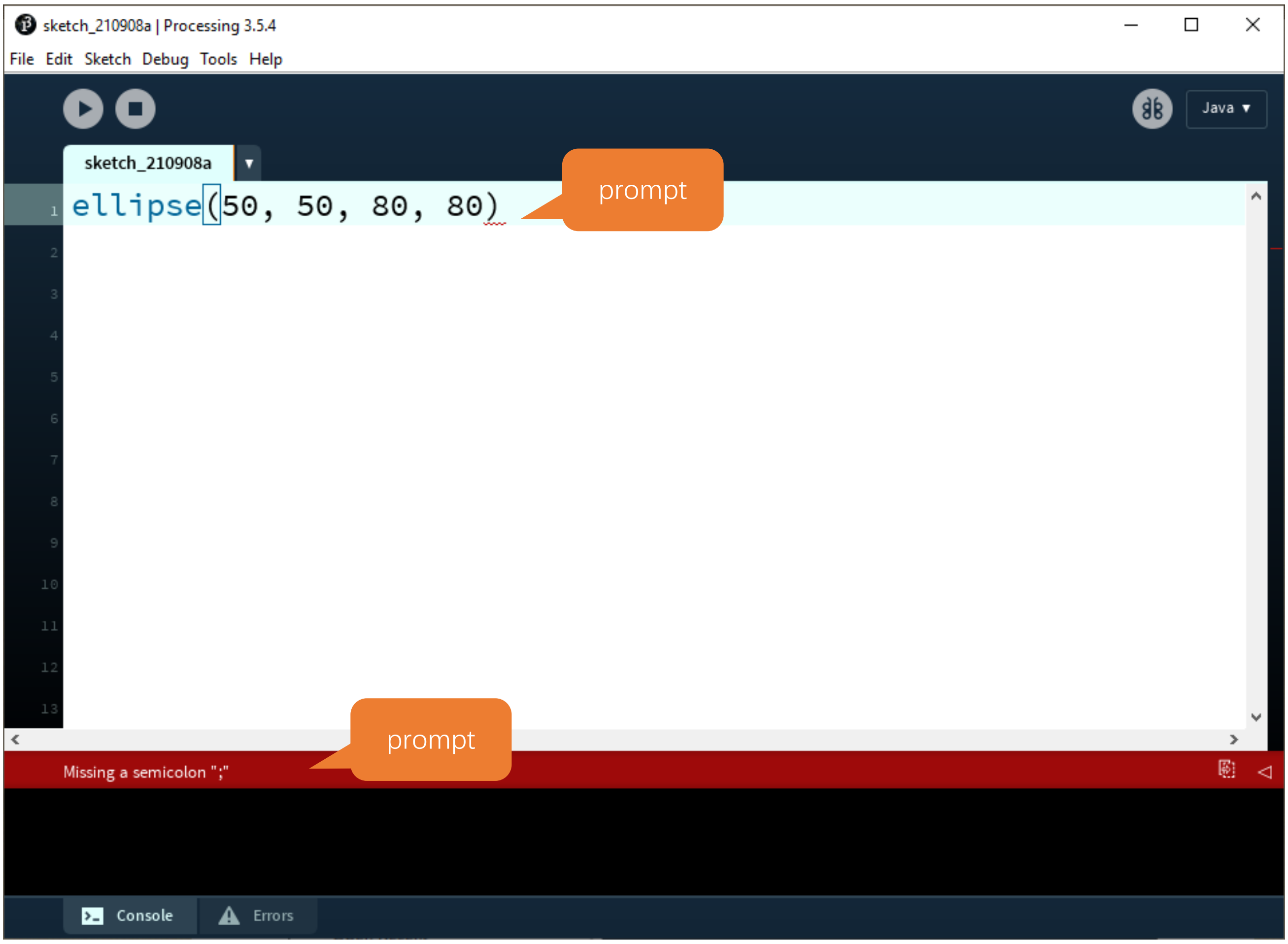

sketch_210908a | Processing 3.5.4
File Edit Sketch Debug Tools Help
Java ▾
sketch_210908a
Console
Errors

```
ellipse(50, 50, 80, 80);
```

If you've typed everything correctly, you'll see a circle on your screen.



If you didn't type it correctly, the Message Area will turn red and complain about an error. If this happens, make sure that you've copied the example code correctly, the numbers should be contained within parentheses and have commas between each of them, and the line should end with a semicolon.

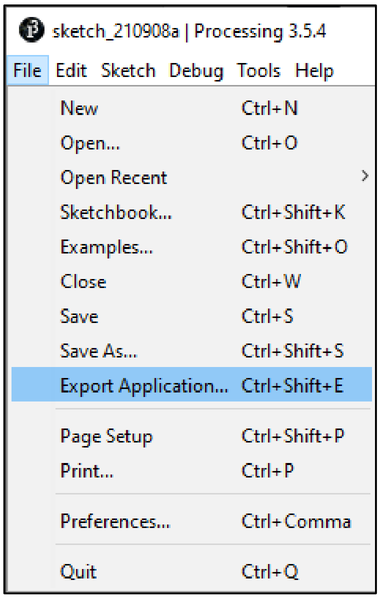

sketch_210908a | Processing 3.5.4
File Edit Sketch Debug Tools Help
Java
sketch_210908a
prompt
prompt
Console
Errors

```
ellipse(50, 50, 80, 80)
```

Missing a semicolon ";"

You can export this sketch to an application to run it directly without opening the Processing.

To export the sketch to the application, you must first save it.



So far, we have completed the first use. I believe you have felt the joy of it.

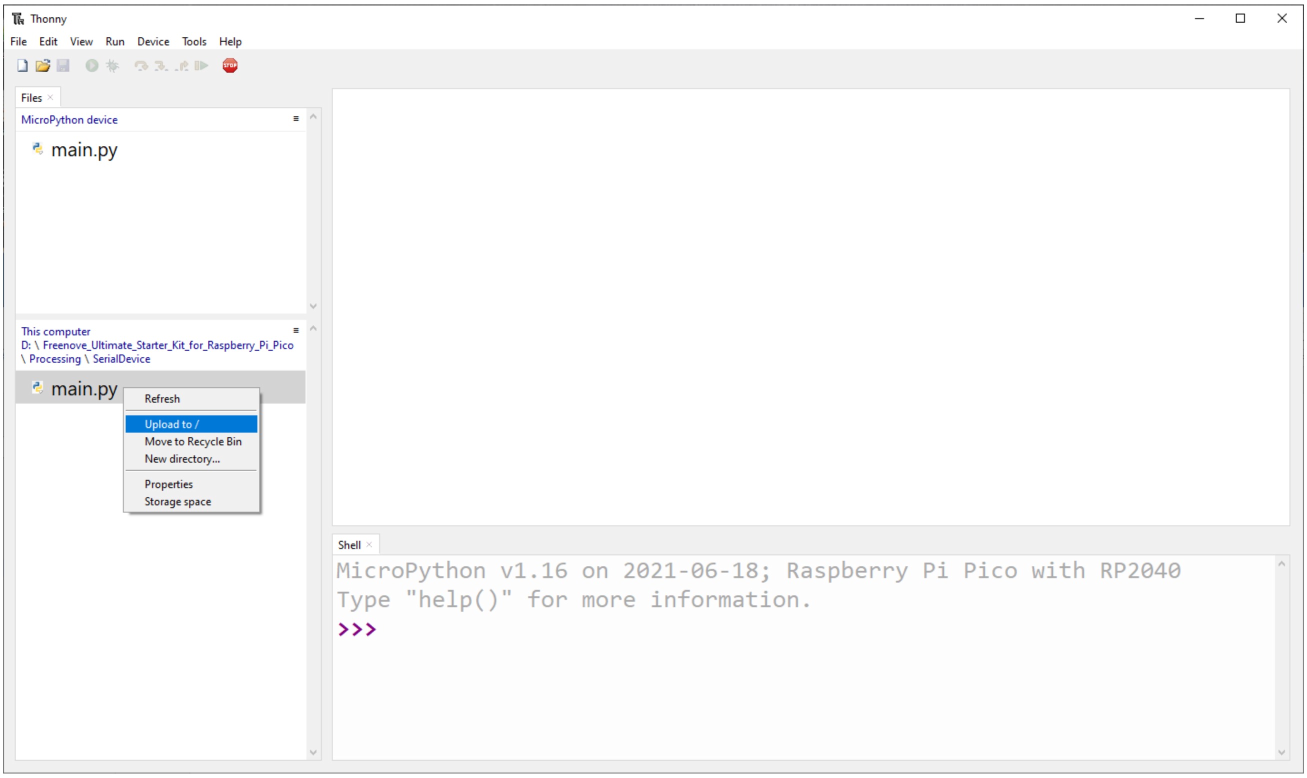
Communication protocol

We need to write code for control board and Processing to complete the interaction project of them, respectively.

To simplify the operation, we've compile SerialDevice class on Processing to communicate with Raspberry Pi Pico. To use this class, we need to upload the below main.py file to Pico.

Freenove_Ultimate_Starter_Kit_for_Raspberry_Pi_Pico\Python\Processing\SerialDevice\main.py

Open Thonny, select the main.py file of the above path, right click the mouse to and select "Upload to /" to upload the file to Pico. Run the main.py file offline, which means you need to disconnect the USB cable from Pico and then reconnect them.



```
MicroPython v1.16 on 2021-06-18; Raspberry Pi Pico with RP2040
Type "help()" for more information.
>>>
```

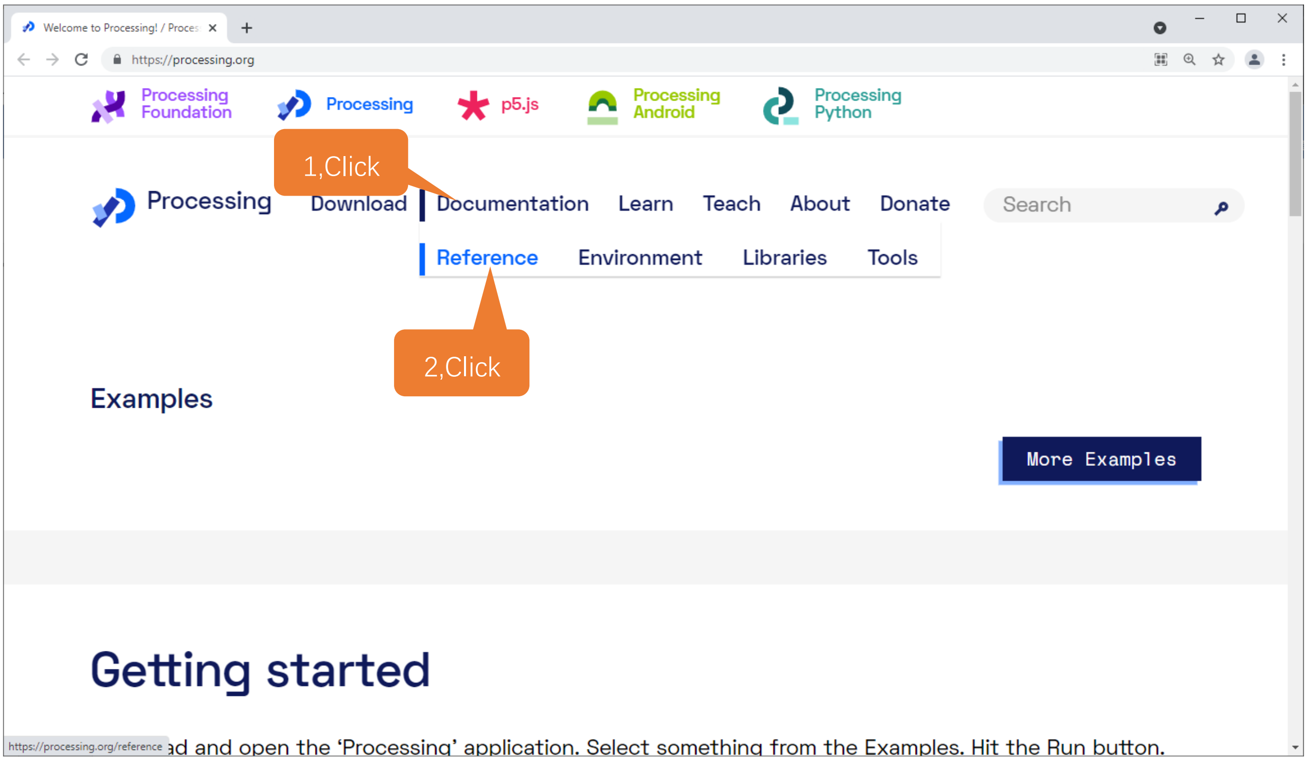
Main.py file only need to be uploaded once. So we won't upload it again for the further projects.

SerialDevice class and main.py defined the communication protocol between them. The features include:

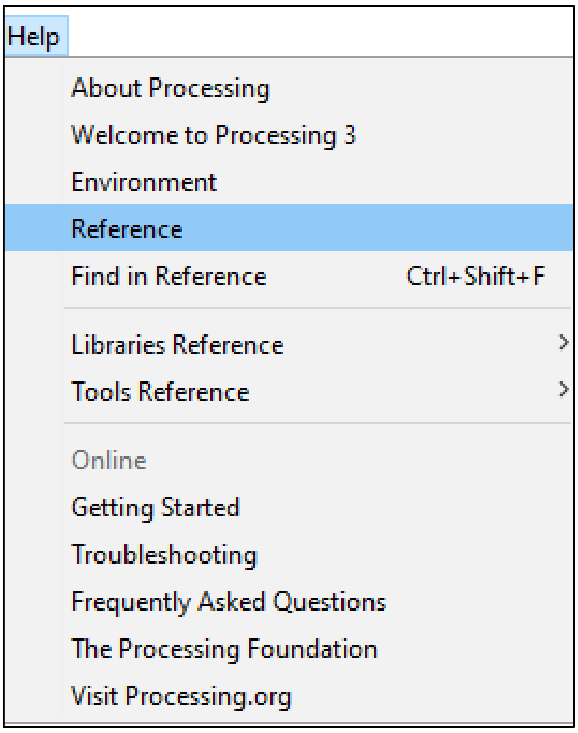
- Recognize the control board that uploads SerialDevice.ino and establish connection with it, automatically. No need to view and set the serial number of the control board connected to the computer, even if there are a number of control boards, it can be connected automatically.
- If the control board that uploads main.py is not connected to computer, the Processing code will not be executed until the connection is done. The Processing sketch does not need to be run again after the connection is done.
- Send data to control board and receive data from it.

Here, Processing sketch code will not be introduced in detail. Interested readers can learn it by themselves.

And as for syntax and standard functions of Processing, you can visit https://processing.org/ and click Reference to view.



Or in the Processing software menu bar, click Help-Reference to view offline documents.



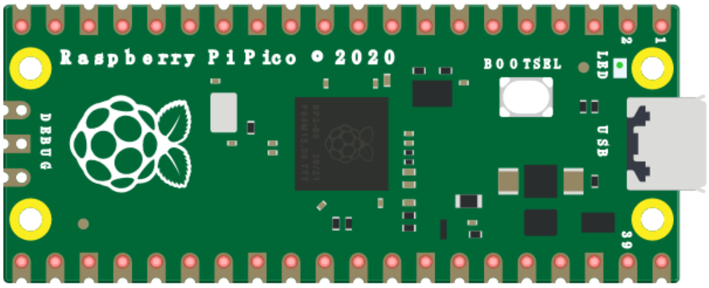
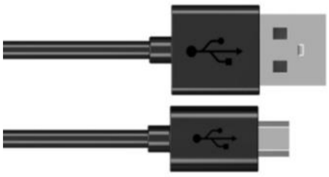
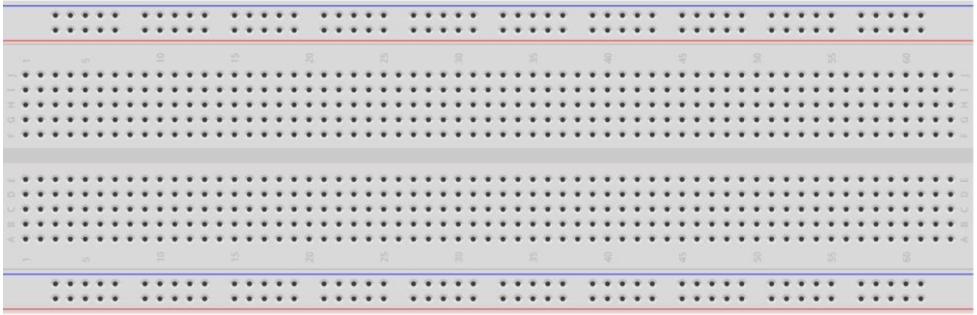
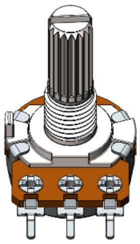
Chapter 1 Oscilloscope

In this chapter, we will make a complex virtual instrument, oscilloscope. Oscilloscope is a widely used electronic test instrument. It can get the electrical signals not directly observed into visible image to facilitate the analysis and study of various electrical signals change process.

Project 1.1 Oscilloscope

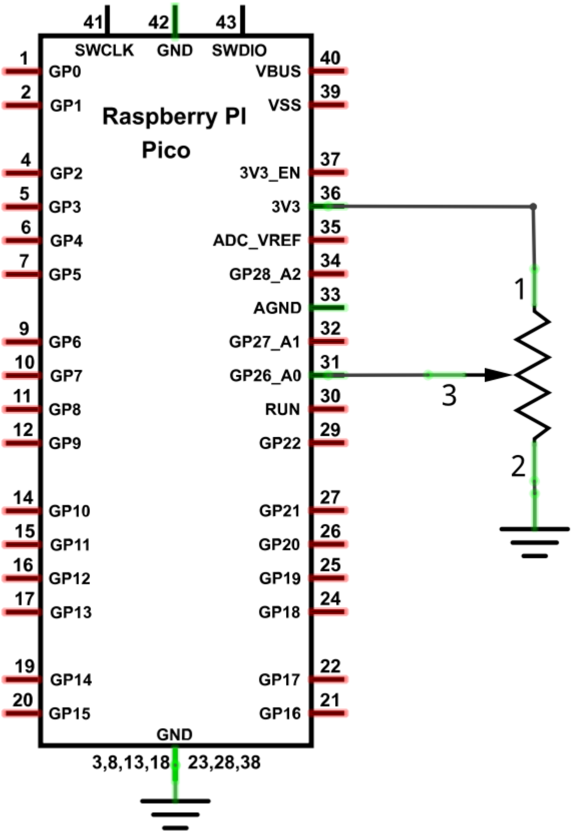
Now, let's use Processing and Raspberry Pi Pico board to create an oscilloscope.

Component List

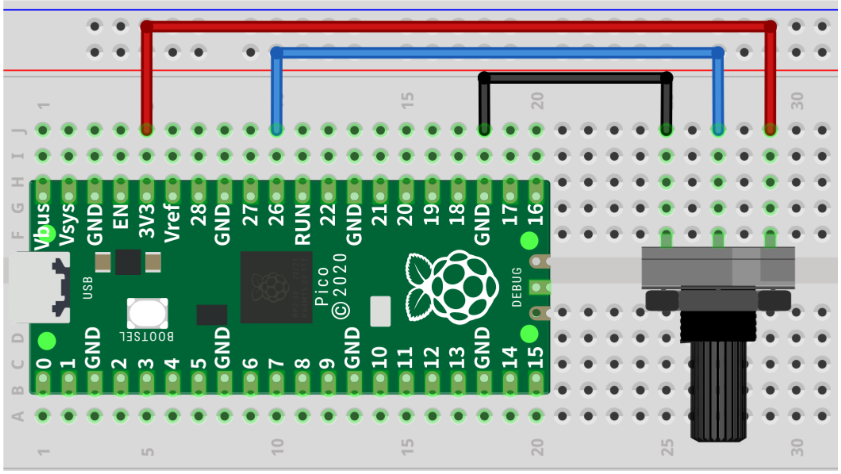
Raspberry Pi Pico x1  	USB cable x1 
Breadboard x1 	
Jumper 	Rotary potentiometer x1 

Circuit

Schematic diagram



Hardware connection. If you need any support, please feel free to contact us via: support@freenove.com



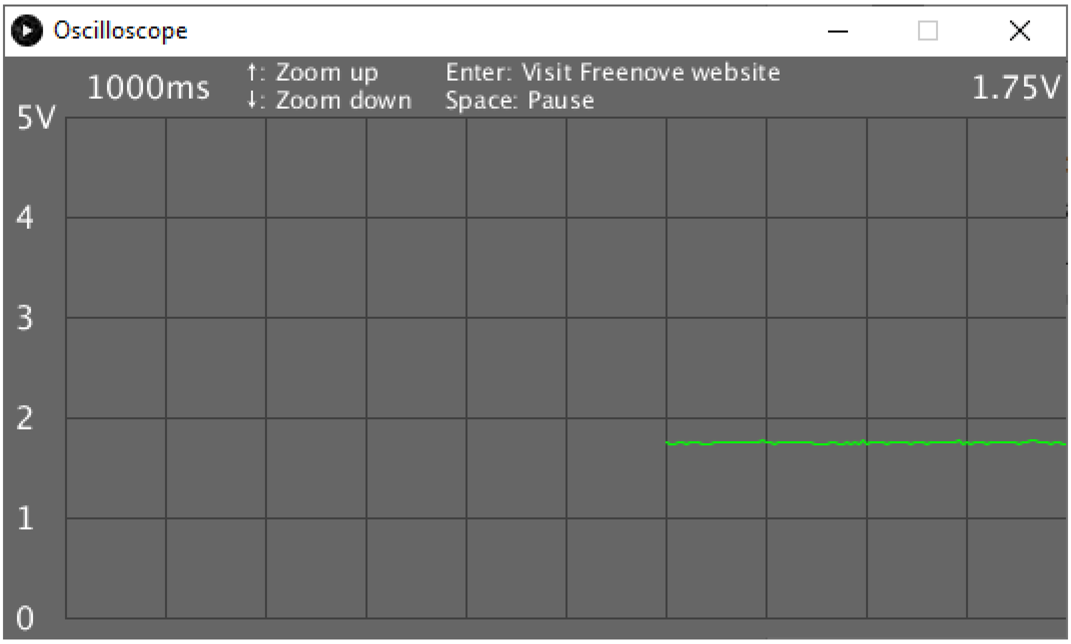
Note: To help users have a better experience when doing the projects, we have made some modifications to Pico's simulation diagram. Please note that there are certain differences between the simulation diagram and the actual board to avoid misunderstanding. Thanks for your support to our product.

Sketch

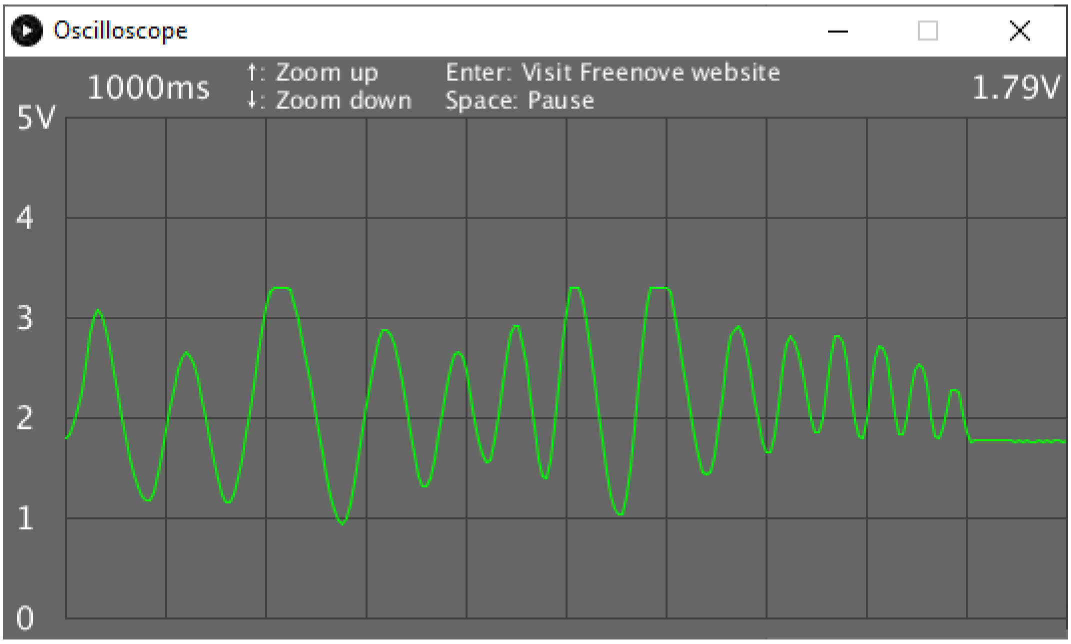
Sketch Oscilloscope

Use Processing to open **Freenove_Ultimate_Starter_Kit_for_Raspberry_Pi_Pico\Python\Processing\Processing\Oscilloscope\Oscilloscope.pde** and click Run.

If the connection succeeds, it will show as follows:



The green line is the waveform acquisited. Rotate the potentiometer, then you can see changes of the waveform:



The left side of the software interface is a voltage scale, which is used to indicate the voltage of the waveform. The "1000ms" on top left corner is the time of a square, and you can press "↑" and "↓" key on keyboard to adjust it.
The "1.79V" on top right corner is the voltage value of current signal.
You can press the space bar on keyboard to pause the display waveform, which is easy to view and analysis.

We believe that with the help of this oscilloscope, you can obtain more intuitive understanding of the actual work of some electronic circuits. It will help you complete the project and eliminate the trouble. You can export this sketch to an application used as a tool.

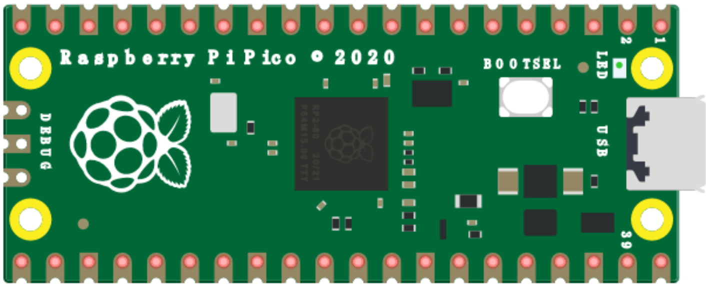
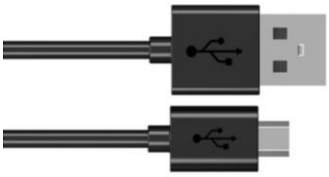
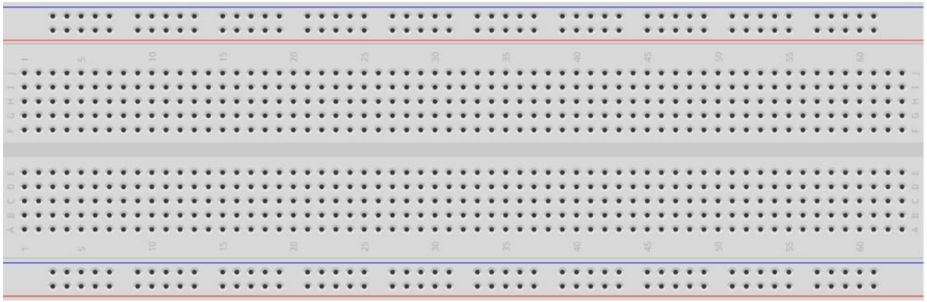
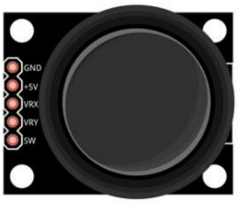
Chapter 2 Control 2D and 3D Figures

In this chapter, we will use Raspberry Pi Pico board to make Processing program control figures to change. And we will control 2D and 3D figures, respectively.

Project 2.1 Ellipse

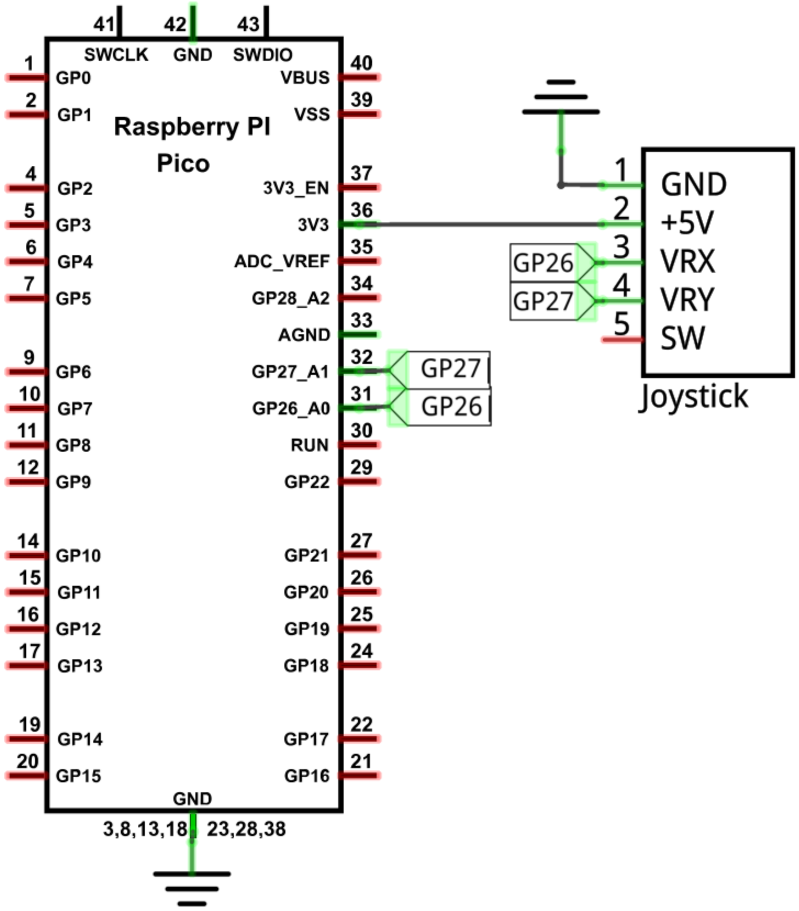
First, Control a 2D figures.

Component List

Raspberry Pi Pico x1 	USB cable x1 
Breadboard x1 	
Joystick x1 	Jumper

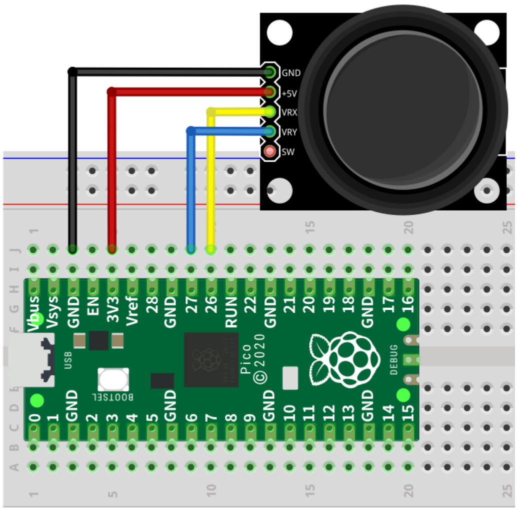
Circuit

Schematic diagram



Note: The ADC of Pico can only collect voltage between 0-3.3V. The joystick here uses 3.3V.

Hardware connection. If you need any support, please feel free to contact us via:support@freenove.com



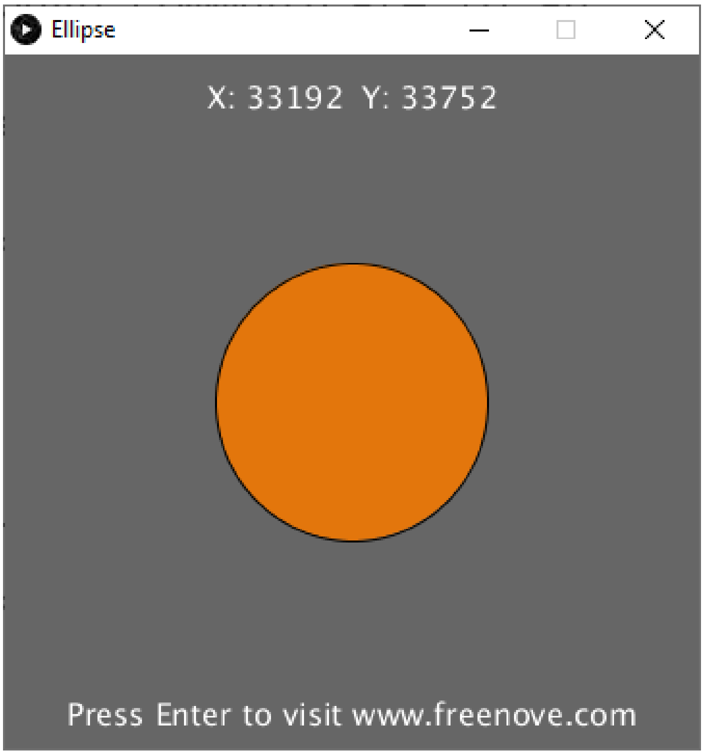
Note: To help users have a better experience when doing the projects, we have made some modifications to Pico's simulation diagram. Please note that there are certain differences between the simulation diagram and the actual board to avoid misunderstanding. Thanks for your support to our product.

Sketch

Sketch Ellipse

Use Processing to open **Freenove_Ultimate_Starter_Kit_for_Raspberry_Pi_Pico\Python\Processing\Processing\Ellipse\Ellipse.pde** and click Run.

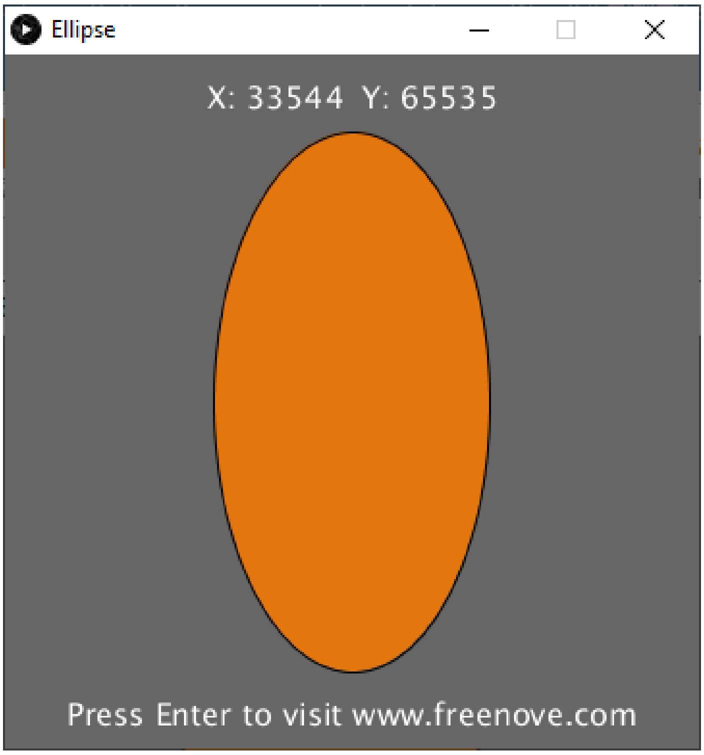
If the connection succeeds, it will show as follows:



X: 33192 Y: 33752

Press Enter to visit www.freenove.com

Then you can change the ellipse shape by shifting the joystick:



X: 33544 Y: 65535

Press Enter to visit www.freenove.com

Project 2.2 Box 3D

Now control a 3D figure.

Component list

The same as last section.

Circuit

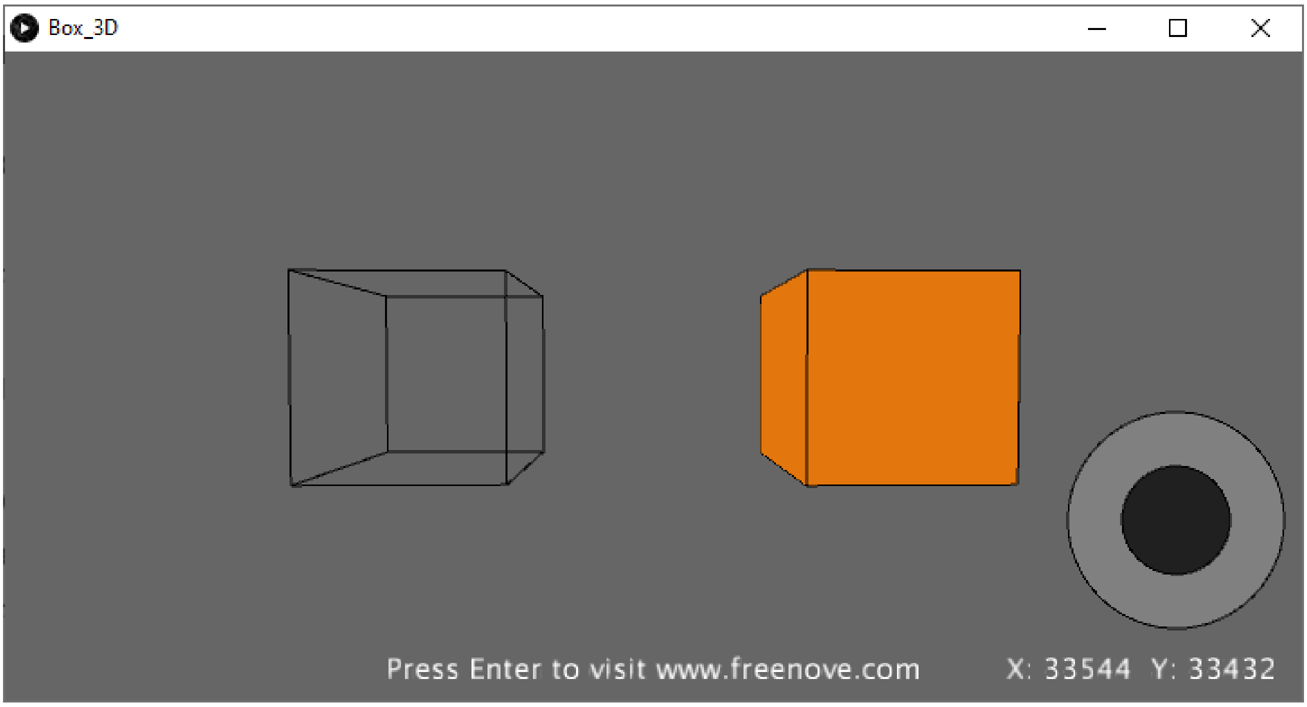
The same as last section.

Sketch

Sketch Box_3D

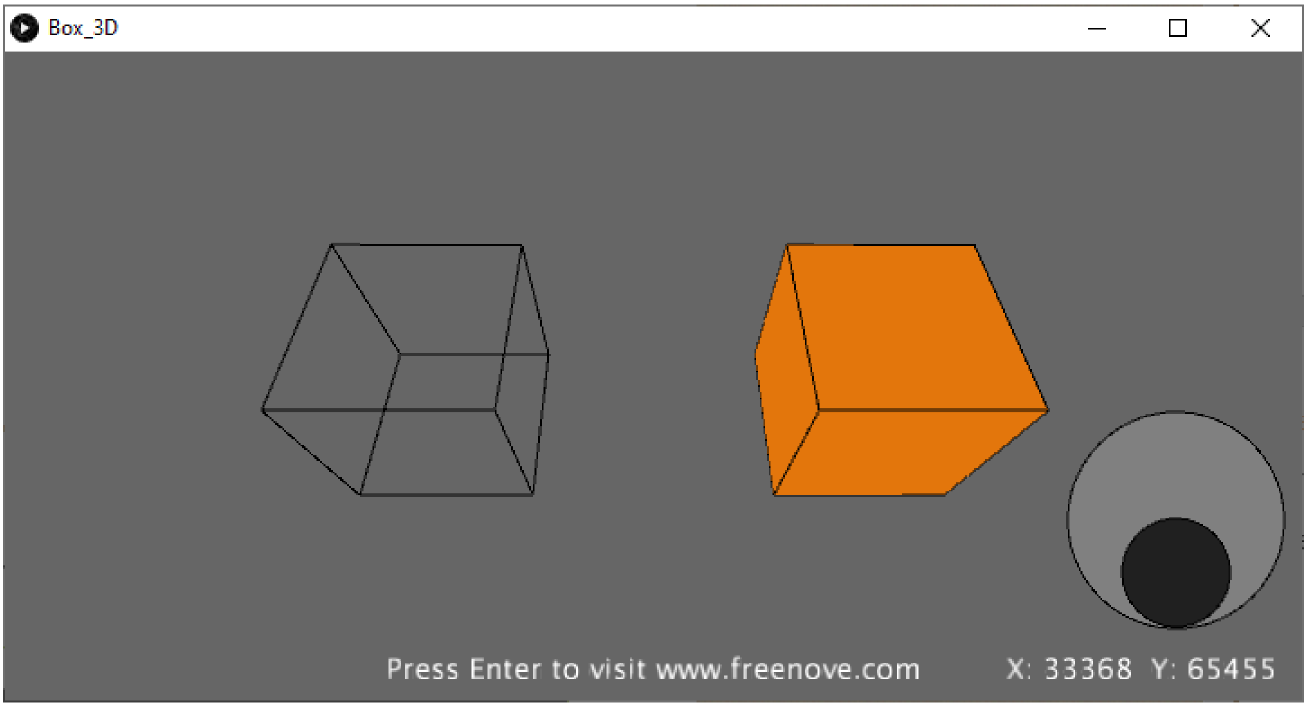
Use Processing to open **Freenove_Ultimate_Starter_Kit_for_Raspberry_Pi_Pico\Python\Processing\Processing\Box_3D\Box_3D.pde** and click Run.

If the connection succeeds, it will show as follows. The left is a 3D box presented by line and the right is a 3D box entity.



```
Press Enter to visit www.freenove.com        X: 33544  Y: 33432
```

Then you can change the space angle of two 3D boxes by shifting the joystick:



Press Enter to visit www.freenove.com

X: 33368 Y: 65455

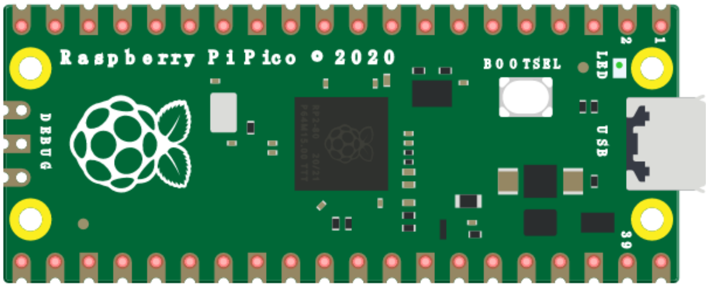
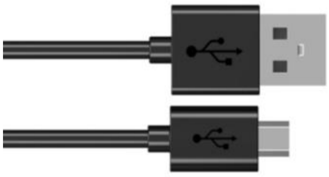
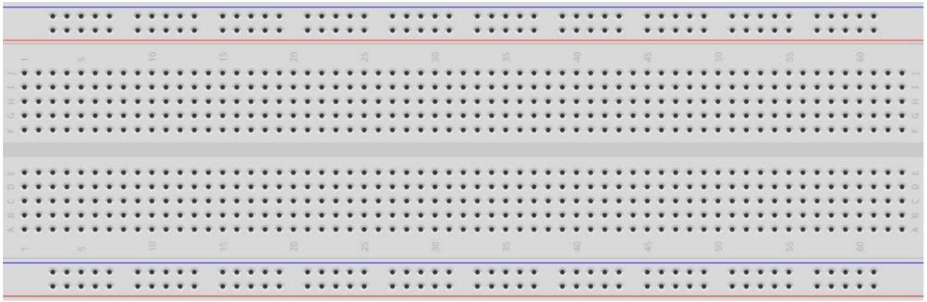
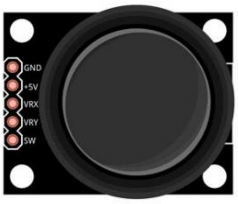
Chapter 3 Snake Game

We have learned how to control 2D and 3D figures. Now, we will use Raspberry Pi Pico to play the classic snake game. There are both 2D and 3D versions.

Project 3.1 Snake Game

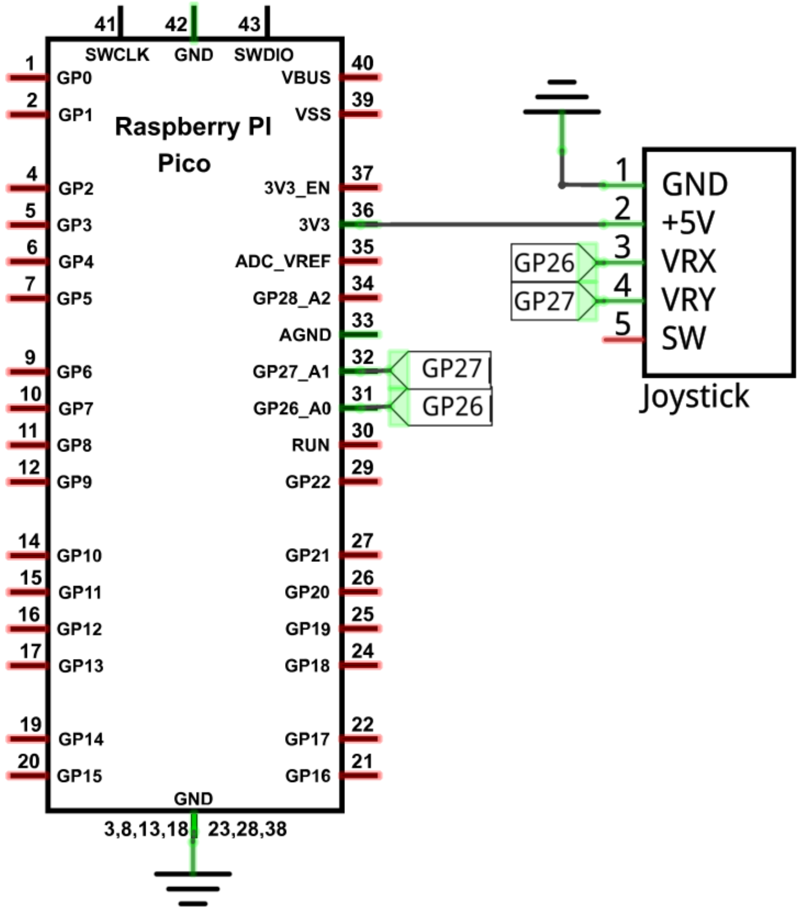
First, let's play a 2D snake game.

Component List

Raspberry Pi Pico x1  	USB cable x1 
Breadboard x1 	
Joystick x1  	Jumper

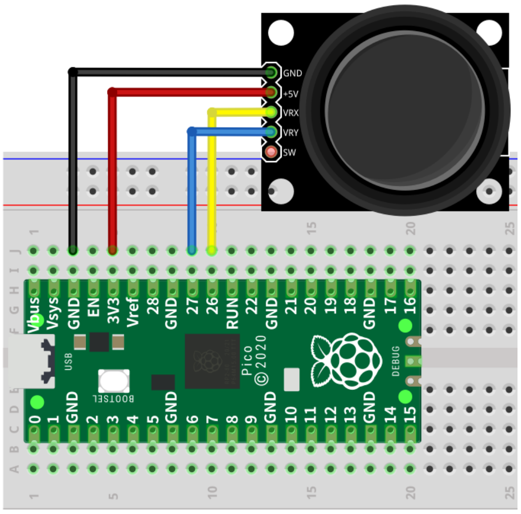
Circuit

Schematic diagram



Note: The ADC of Pico can only collect voltage between 0-3.3V. The joystick here uses 3.3V.

Hardware connection. If you need any support, please feel free to contact us via:support@freenove.com



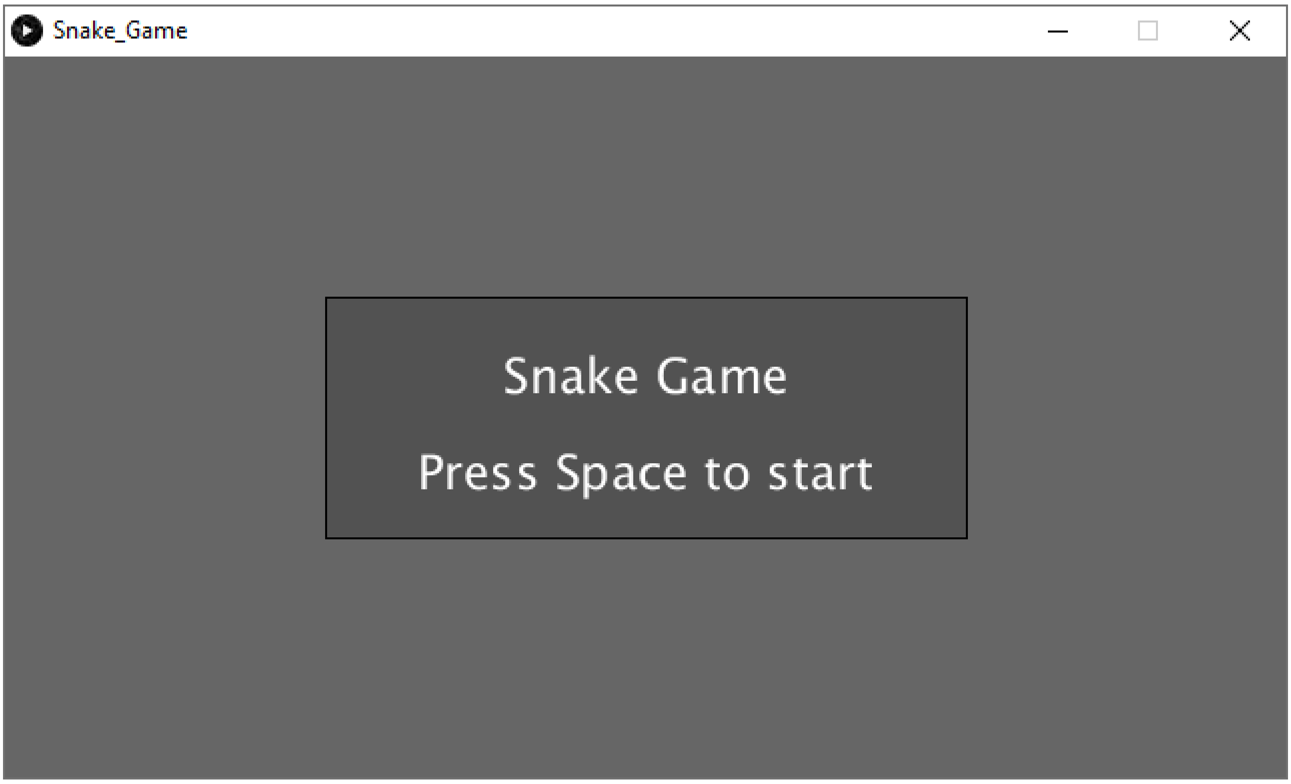
Note: To help users have a better experience when doing the projects, we have made some modifications to Pico's simulation diagram. Please note that there are certain differences between the simulation diagram and the actual board to avoid misunderstanding. Thanks for your support to our product.

Sketch

Sketch Snake_Game

Use Processing to open **Freenove_Ultimate_Starter_Kit_for_Raspberry_Pi_Pico\Python\Processing\Processing\Snake_Game\Snake_Game.pde** and click Run.

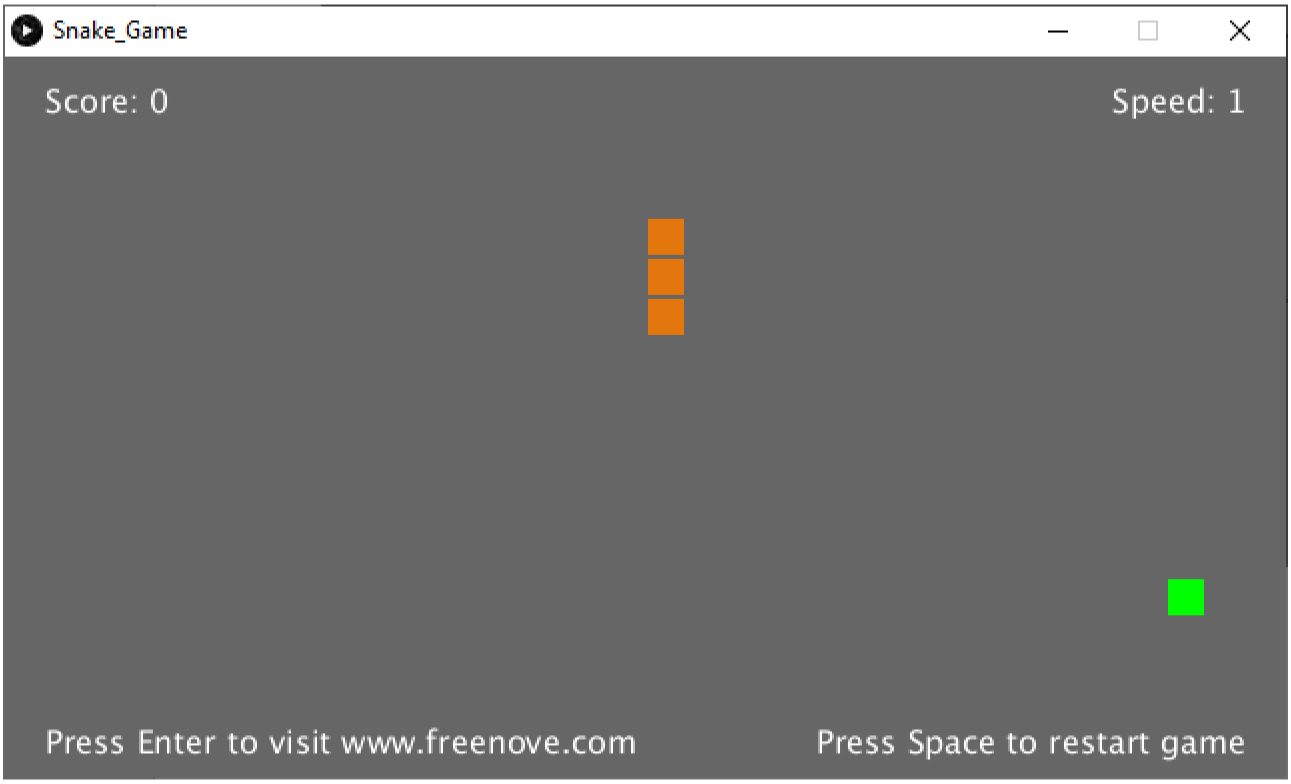
If the connection succeeds, it will show as follows:



Snake Game

Press Space to start

Press the space bar on keyboard to start the game:



Snake_Game

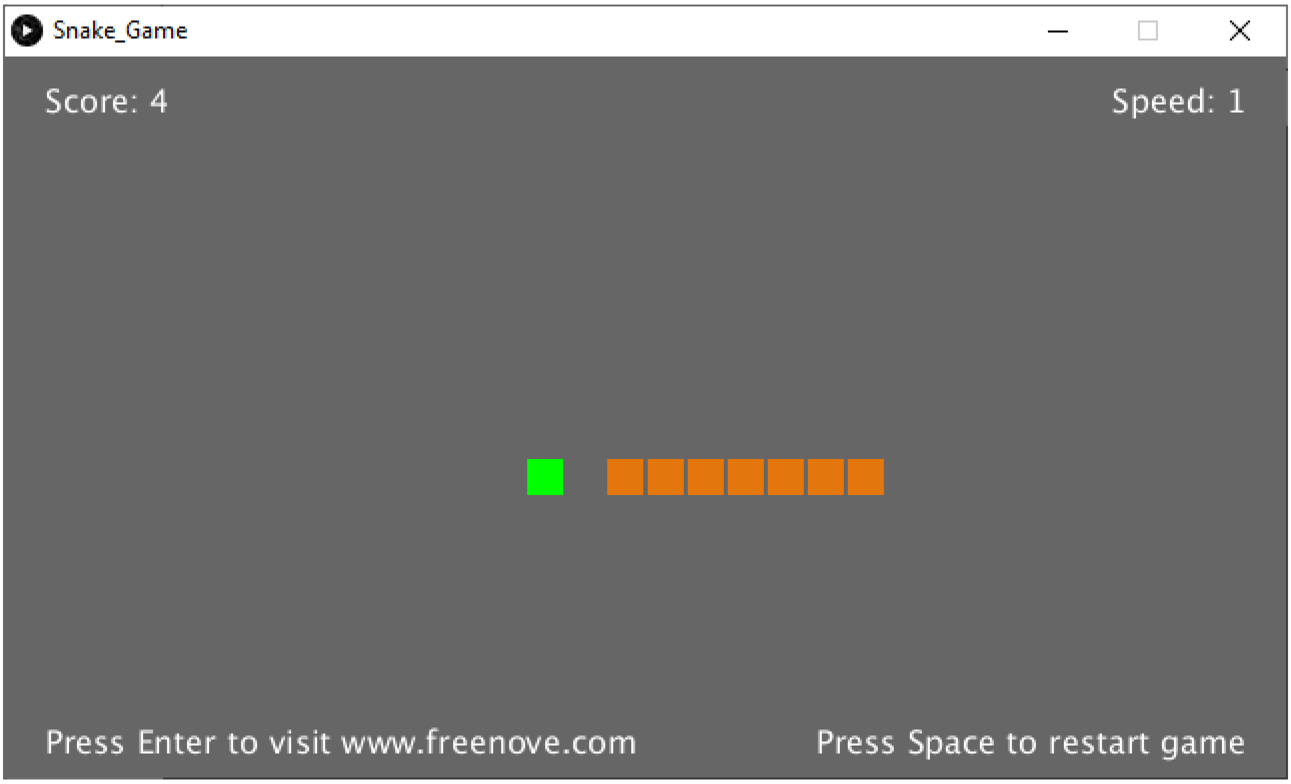
Score: 0

Speed: 1

Press Enter to visit www.freenove.com

Press Space to restart game

Shift the joystick to control the snake's action. The game rules are the same as the classic snake game:



Snake_Game

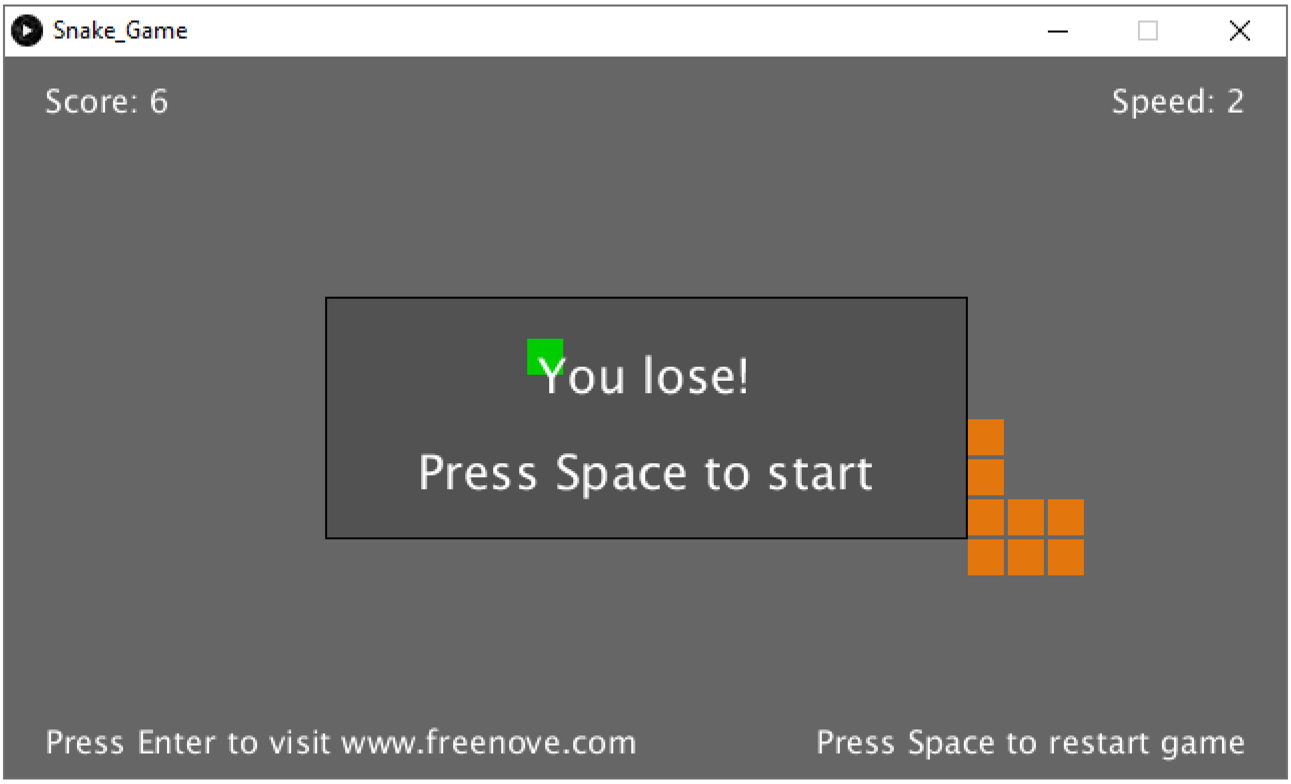
Score: 4

Speed: 1

Press Enter to visit www.freenove.com

Press Space to restart game

When the game fails, press space bar to restart the game:



Snake_Game
Score: 6
Speed: 2
You lose!
Press Space to start
Press Enter to visit www.freenove.com
Press Space to restart game

Additionally, you can restart the game by pressing the space bar at any time.

Project 3.2 Snake Game 3D

Now, let's experience the 3D version game.

Component List

The same as last section.

Circuit

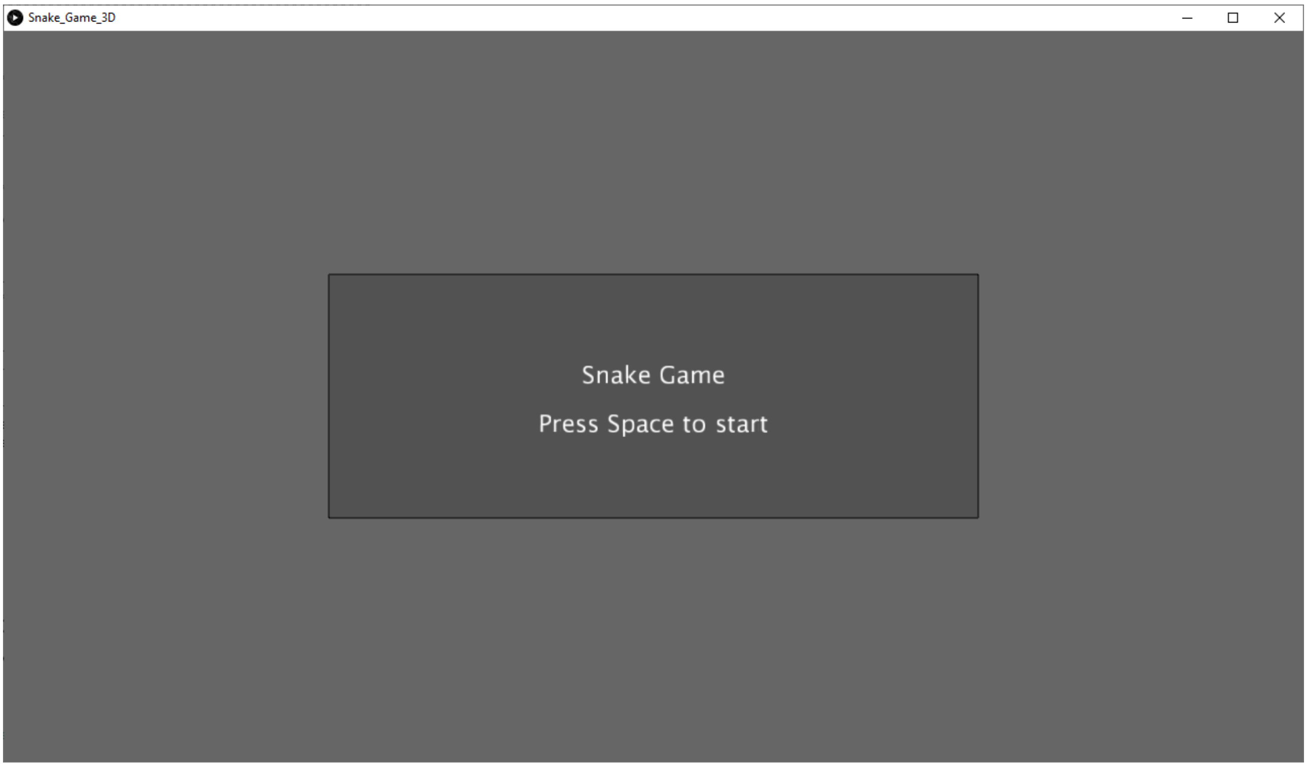
The same as last section.

Sketch

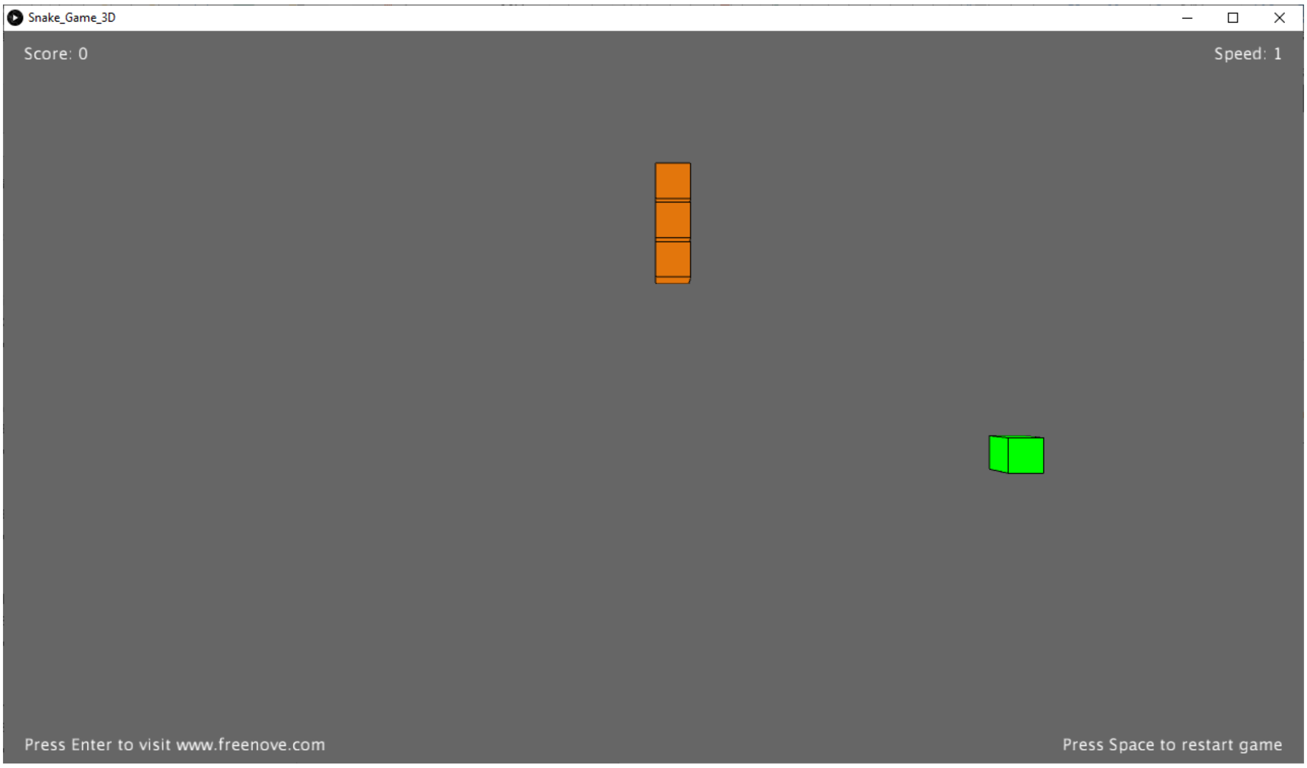
Sketch Snake_Game_3D

Use Processing to open **Freenove_Ultimate_Starter_Kit_for_Raspberry_Pi_Pico\Python\Processing\Processing\Snake_Game_3D\Snake_Game_3D.pde** and click Run.

If the connection succeeds, it will show as follows:

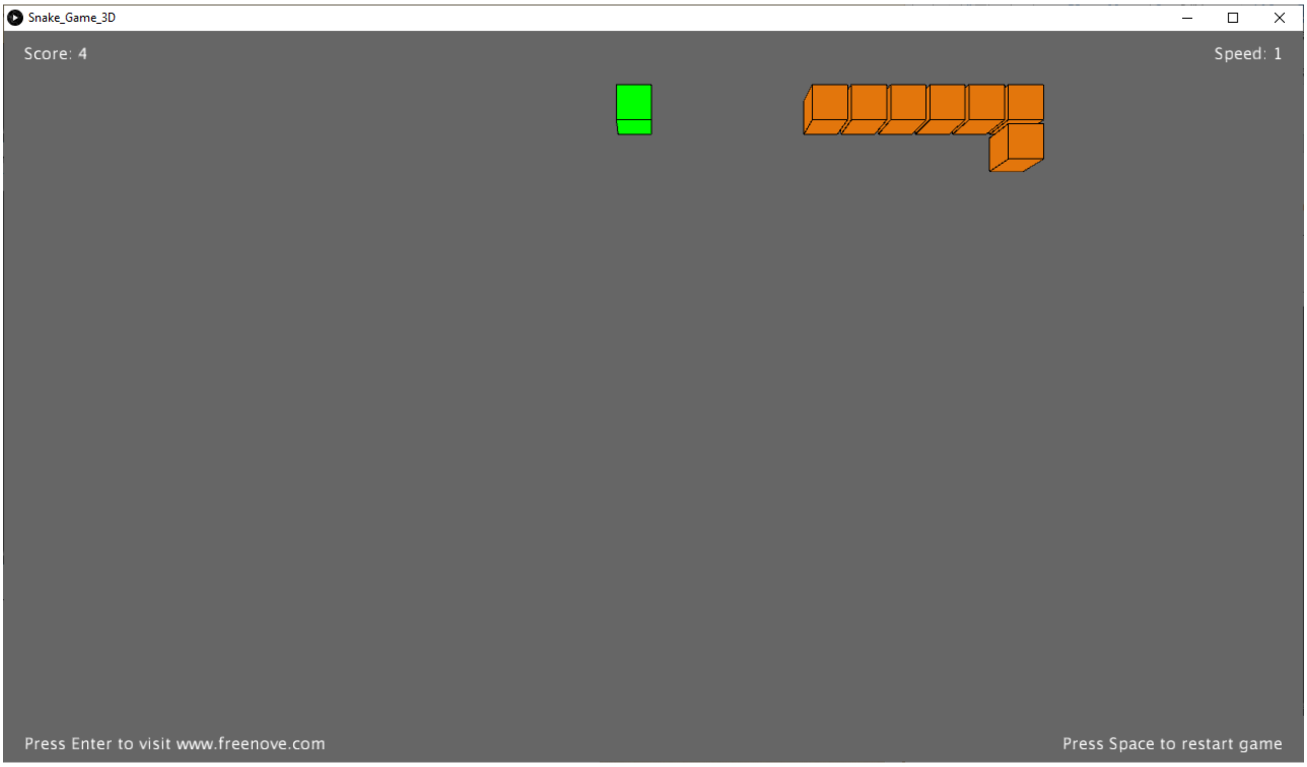


Press the space bar on keyboard to start the game:



Snake_Game_3D
Score: 0
Speed: 1
Press Enter to visit www.freenove.com
Press Space to restart game

Shift the joystick to control the snake's action. The game rules are the same as the classic snake game:



Snake_Game_3D
Score: 4
Speed: 1
Press Enter to visit www.freenove.com
Press Space to restart game

The rest operation is the same as the 2D version.

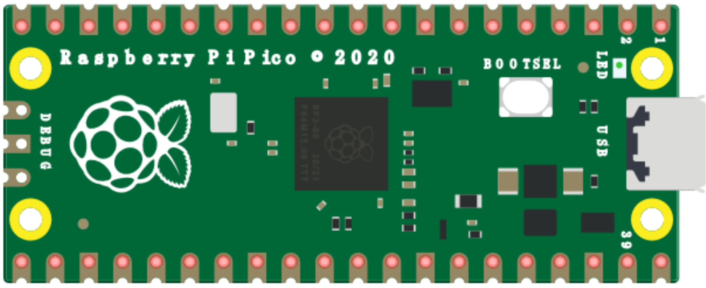
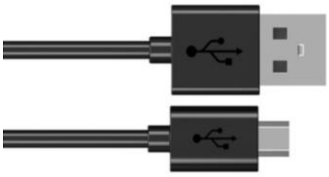
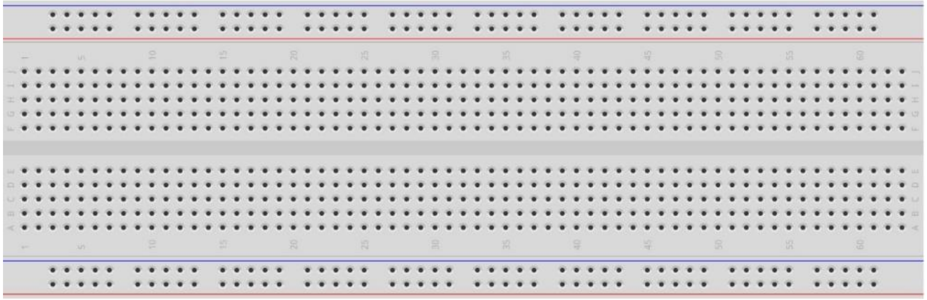
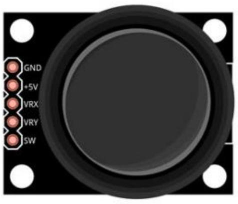
Chapter 4 Pick Apples

In this chapter, we prepare pick apples game. You can use the joystick to pick apples.

Project 4.1 Pick Apples

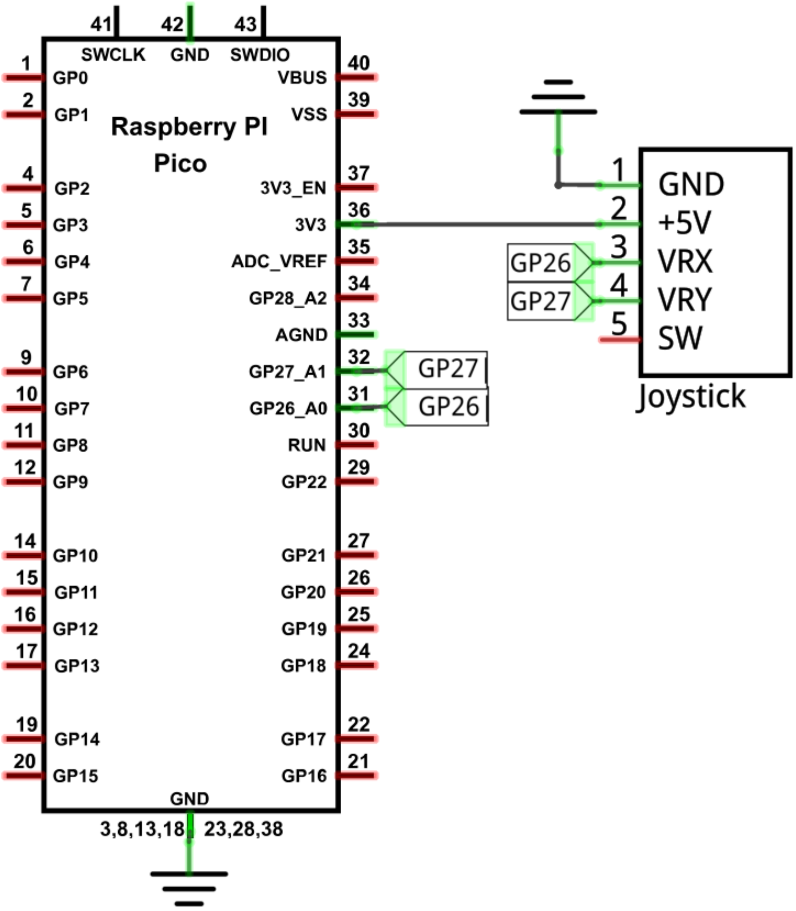
Now, let’s use Processing and Raspberry Pi Pico to achieve the game.

Component List

Raspberry Pi Pico x1  Raspberry Pi Pico © 2020 BOOTSEL LED USB DEBUG 1 2 39	USB cable x1 
Breadboard x1 	
Joystick x1  	Jumper

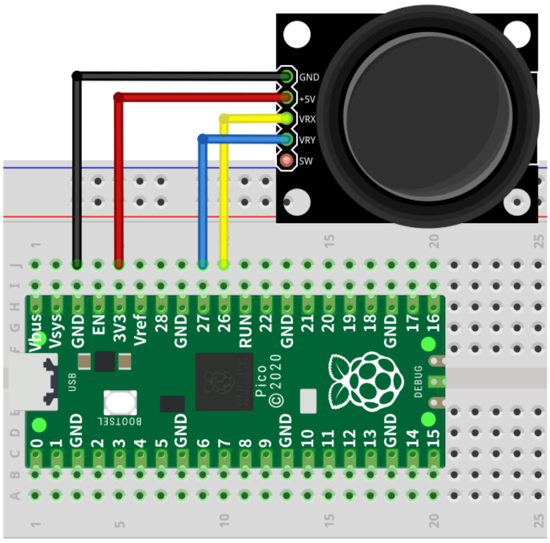
Circuit

Schematic diagram



Note: The ADC of Pico can only collect voltage between 0-3.3V. The joystick here uses 3.3V.

Hardware connection. If you need any support, please feel free to contact us via:support@freenove.com



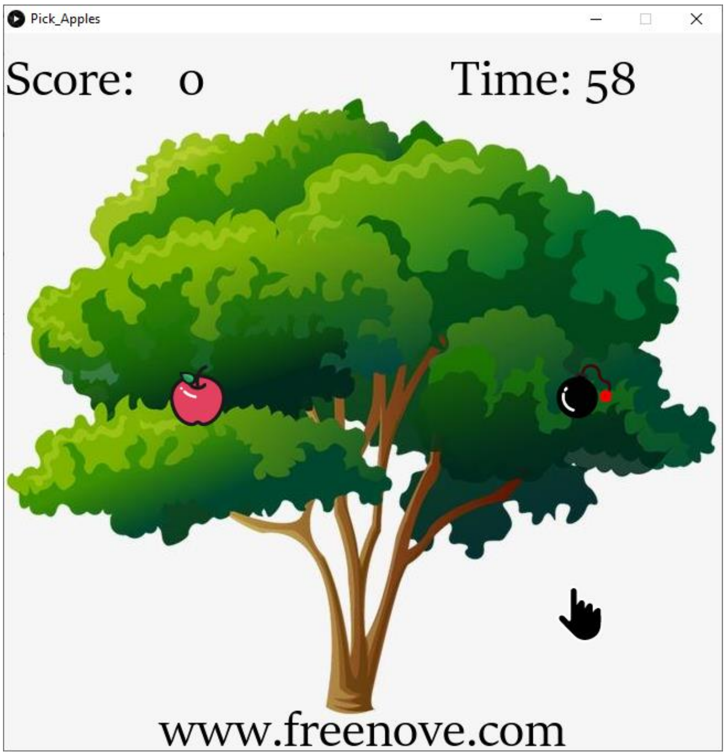
Note: To help users have a better experience when doing the projects, we have made some modifications to Pico's simulation diagram. Please note that there are certain differences between the simulation diagram and the actual board to avoid misunderstanding.

Sketch

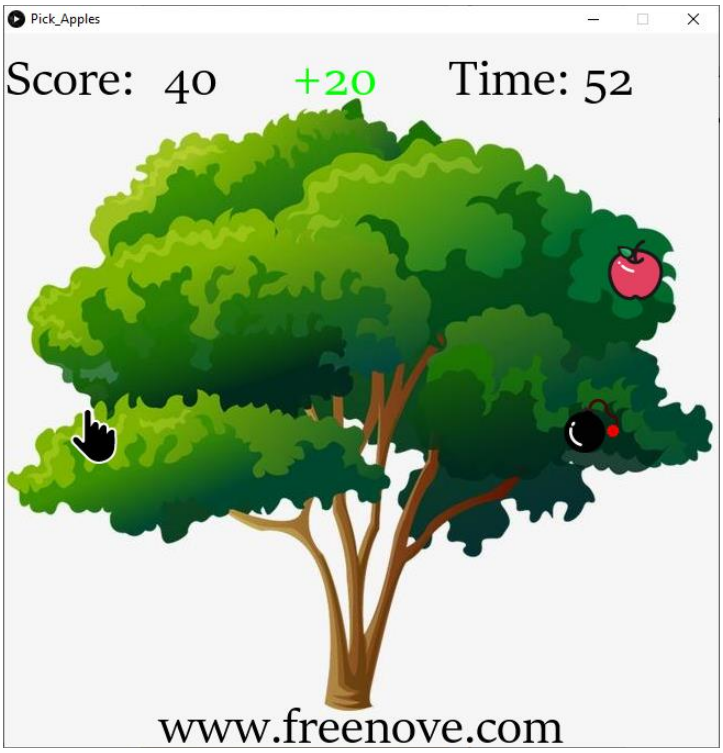
Sketch Pick Apples

Use Processing to open **Freenove_Ultimate_Starter_Kit_for_Raspberry_Pi_Pico\Python\Processing\Processing\Pick_Apples\Pick_Apples.pde** and click Run.

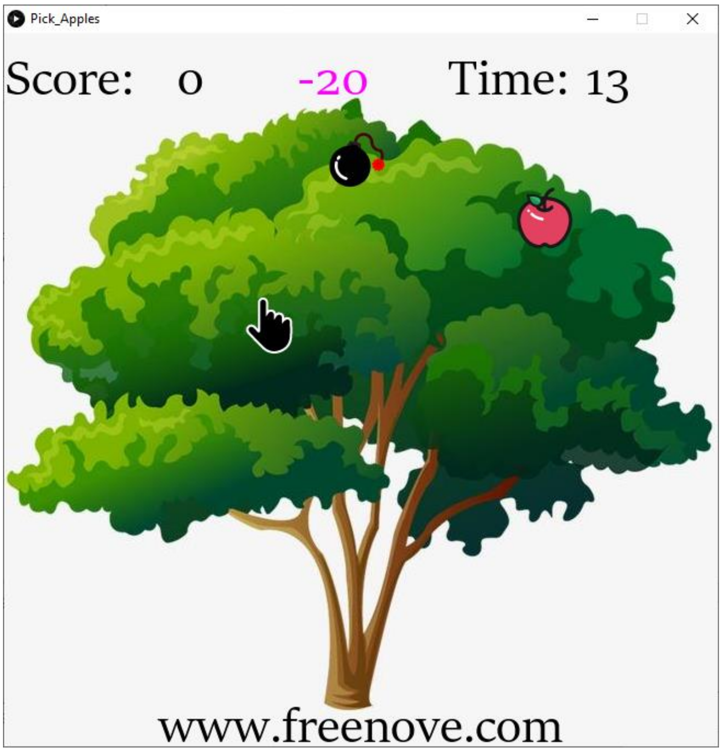
If the connection succeeds, it will show as follows:



You can use the joystick to control the game. When the palm picks the apple, the score increases by 20.

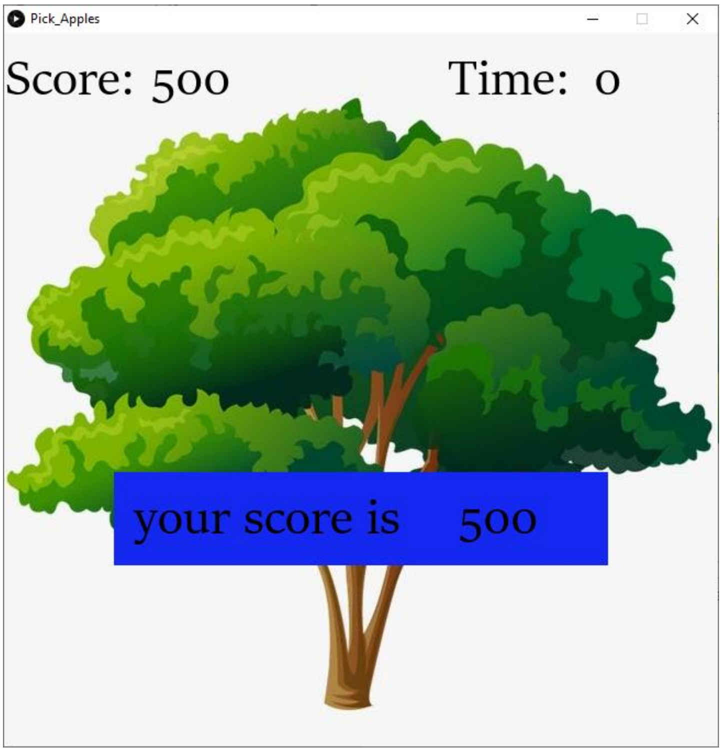


If the palm touches the bomb, the score decreases by 20.



At the same time, you need to pick as many apples as possible before the countdown is over.

When the game fails, press space bar to restart the game:

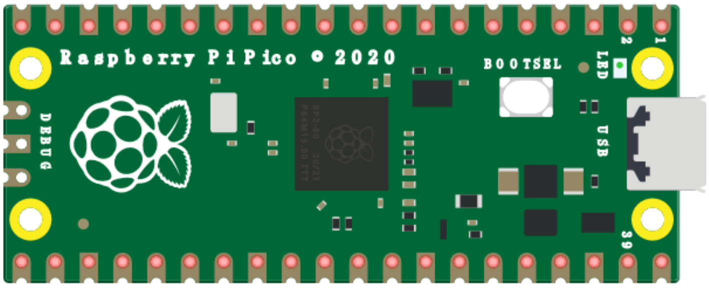
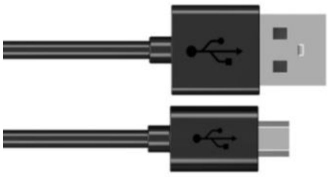
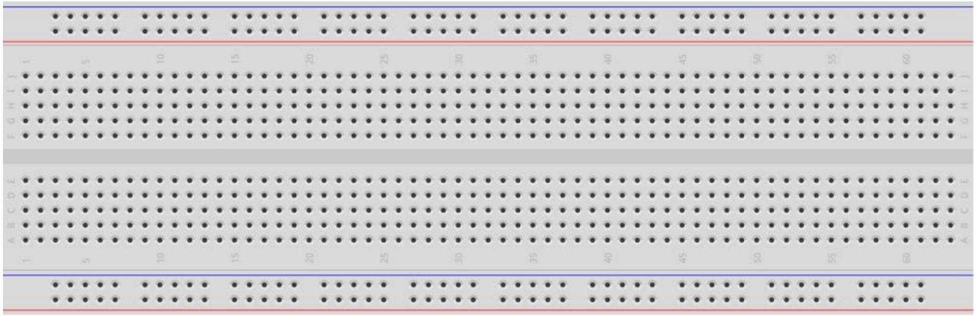
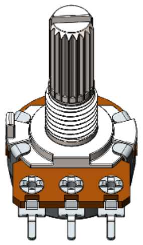


Chapter 5 Pong Game

We have experienced single-player game snake before. Now, let's use Raspberry Pi Pico to play classic two-player pong game. You will experience both 2D and 3D version.

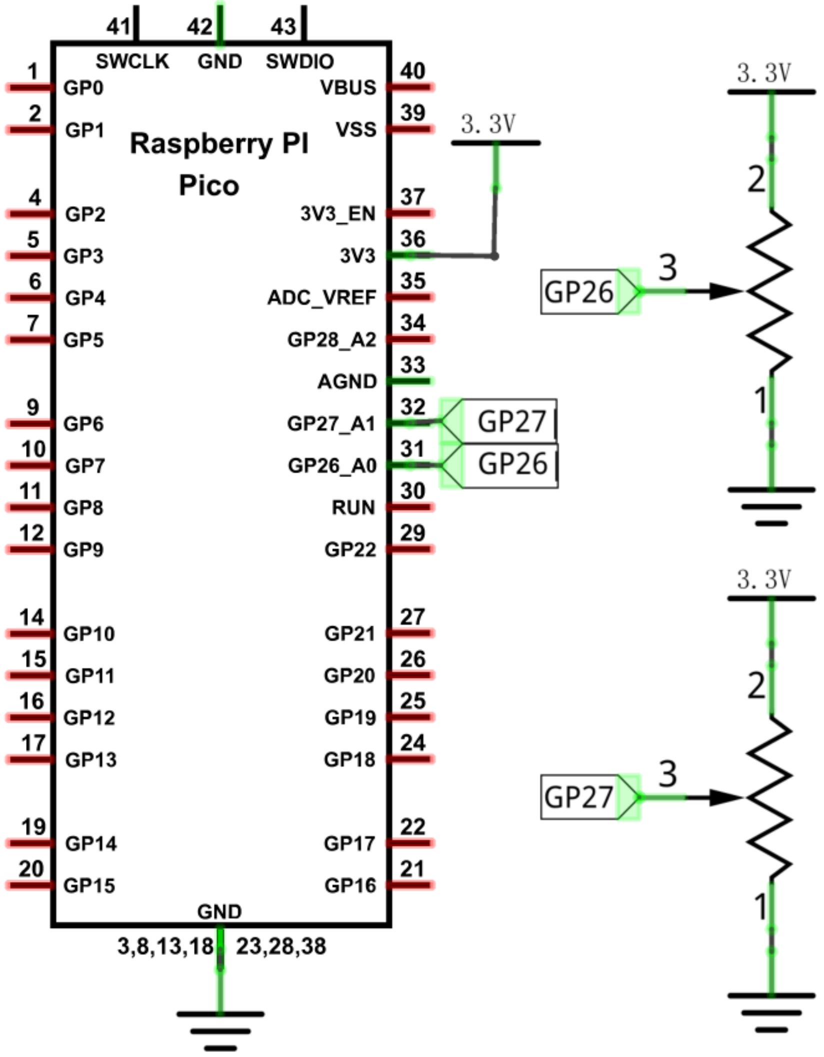
Project 5.1 Pong Game

First, let’s experience the 2D version game.

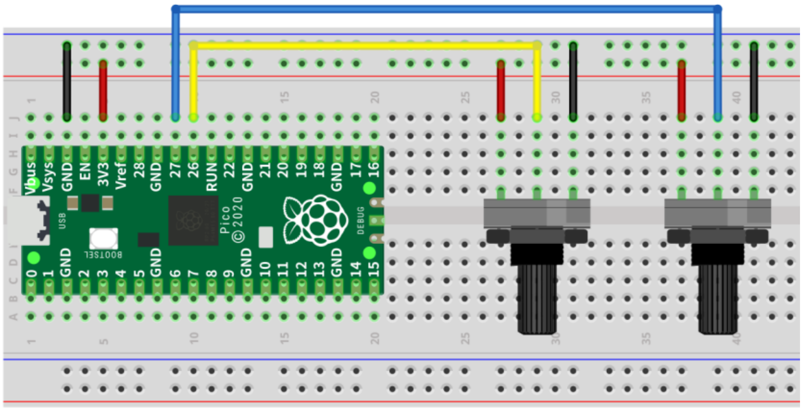
Raspberry Pi Pico x1 	USB cable x1 
Breadboard x1 	
Jumper 	Rotary potentiometer x2 

Circuit

Schematic diagram



Hardware connection. If you need any support, please feel free to contact us via: support@freenove.com

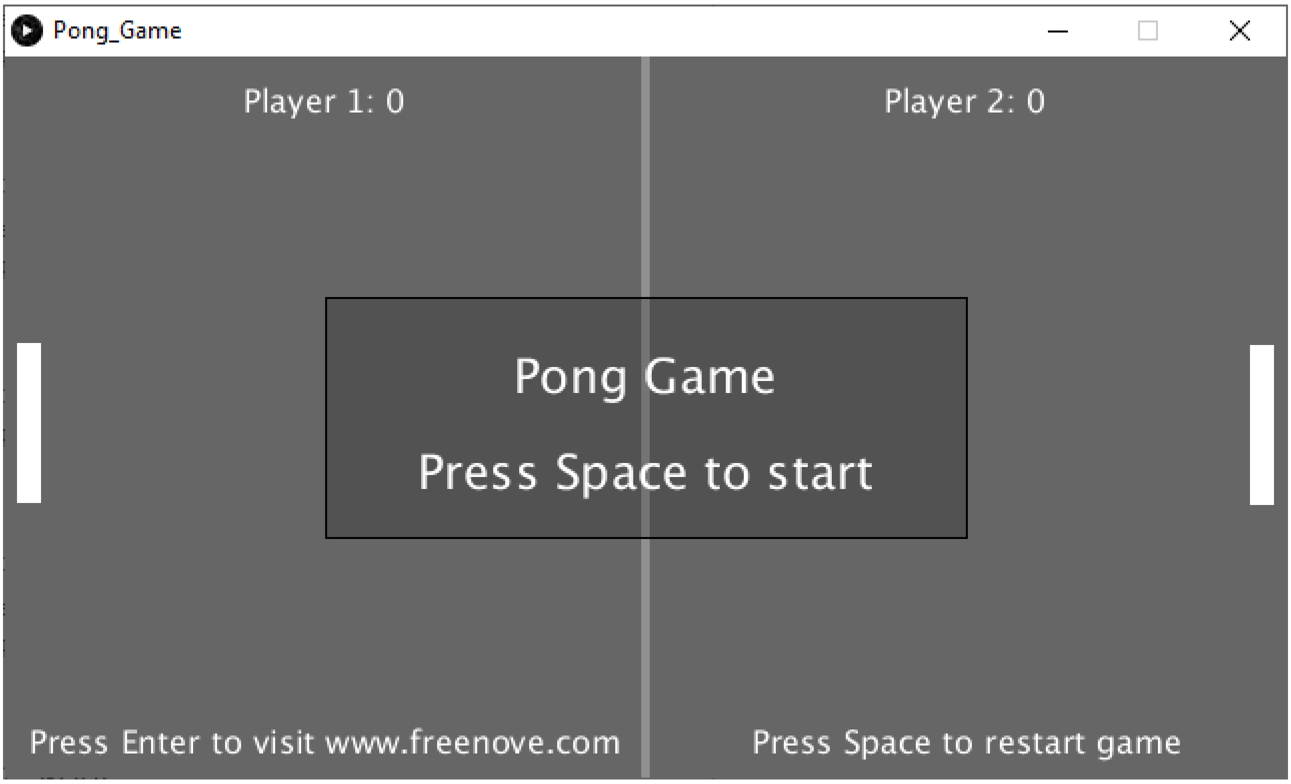


Note: To help users have a better experience when doing the projects, we have made some modifications to Pico’s simulation diagram. Please note that there are certain differences between the simulation diagram and the actual board to avoid misunderstanding.

Sketch

Sketch Pong_Game

Use Processing to open Pong_Game and click Run. If the connection succeeds, it will show as follows:



Pong_Game

Player 1: 0

Player 2: 0

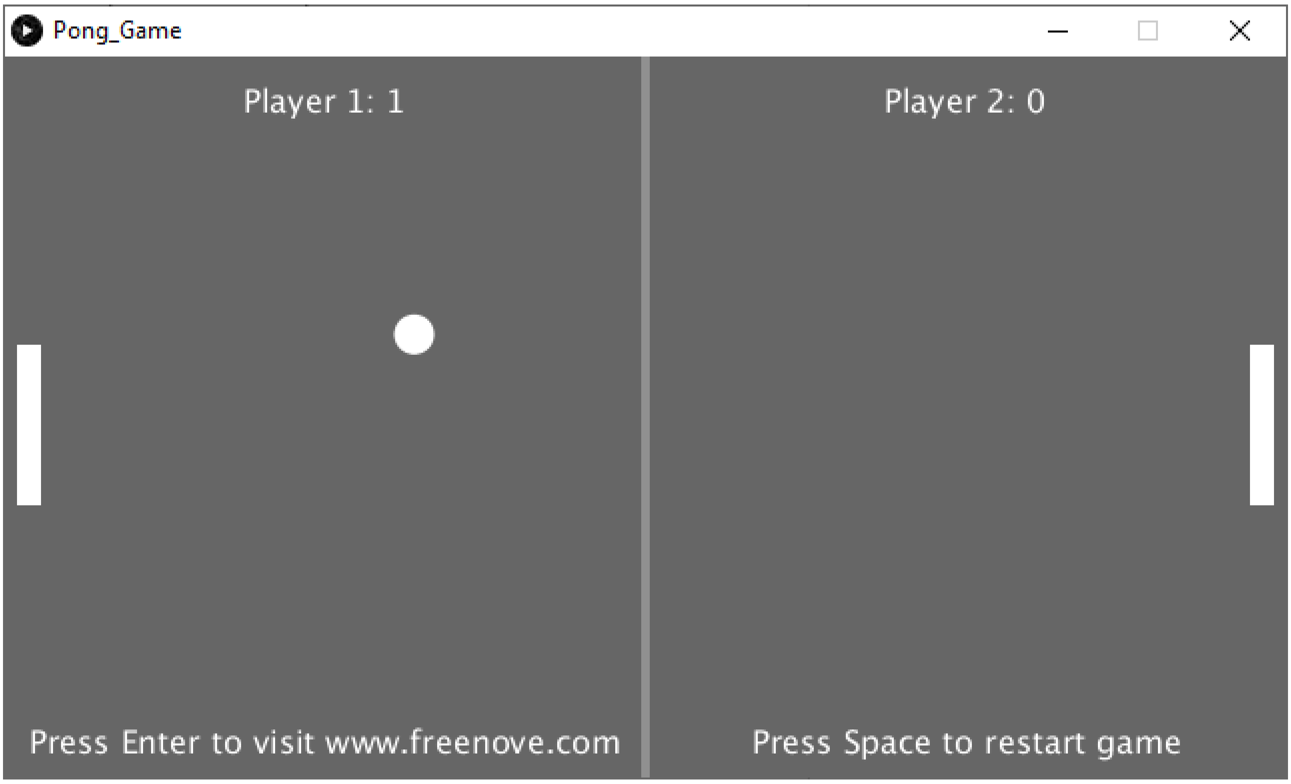
Pong Game

Press Space to start

Press Enter to visit www.freenove.com

Press Space to restart game

Now you can try to rotate the potentiometer to control the movement of paddle without ball. Press space bar to start the game:



Pong_Game

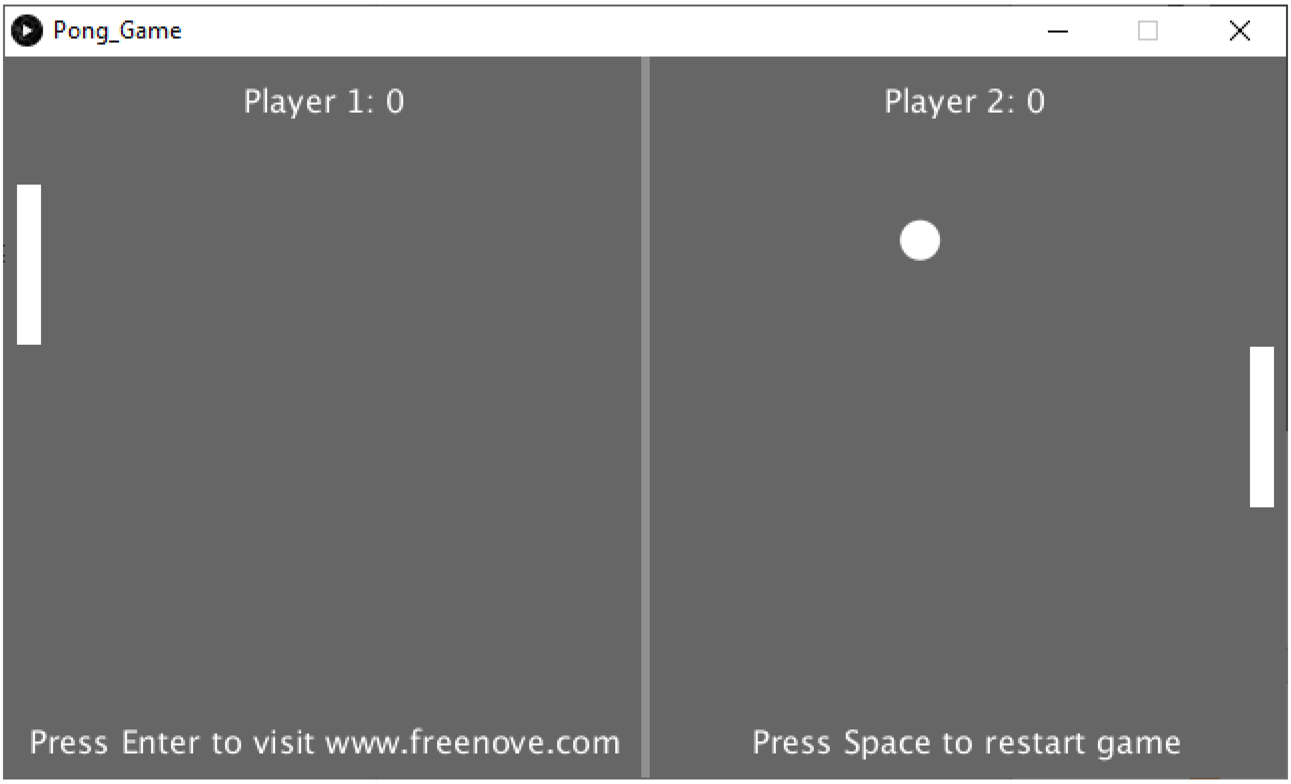
Player 1: 1

Player 2: 0

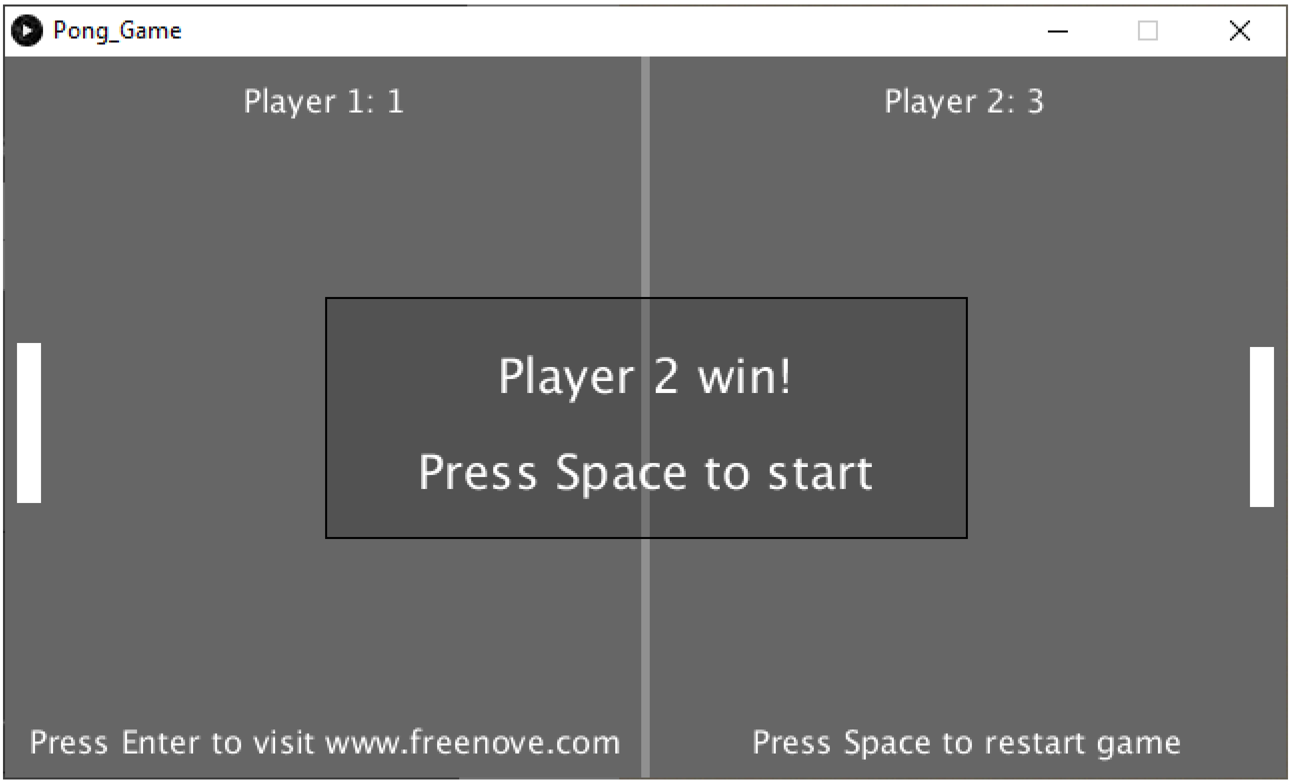
Press Enter to visit www.freenove.com

Press Space to restart game

Use potentiometer to control the movement of paddle to block the ball back. The game rules are the same as classic pong game:


Pong_Game
Player 1: 0
Player 2: 0
Press Enter to visit www.freenove.com
Press Space to restart game

The game will be over when one side reaches three points. Pressing the space bar can restart the game:


Pong_Game
Player 1: 1
Player 2: 3
Player 2 win!
Press Space to start
Press Enter to visit www.freenove.com
Press Space to restart game

Additionally, you can restart the game by pressing the space bar at any time.

Project 5.2 Pong Game 3D

Now, let’s experience the 3D version game.

Component List

The same as last section.

Circuit

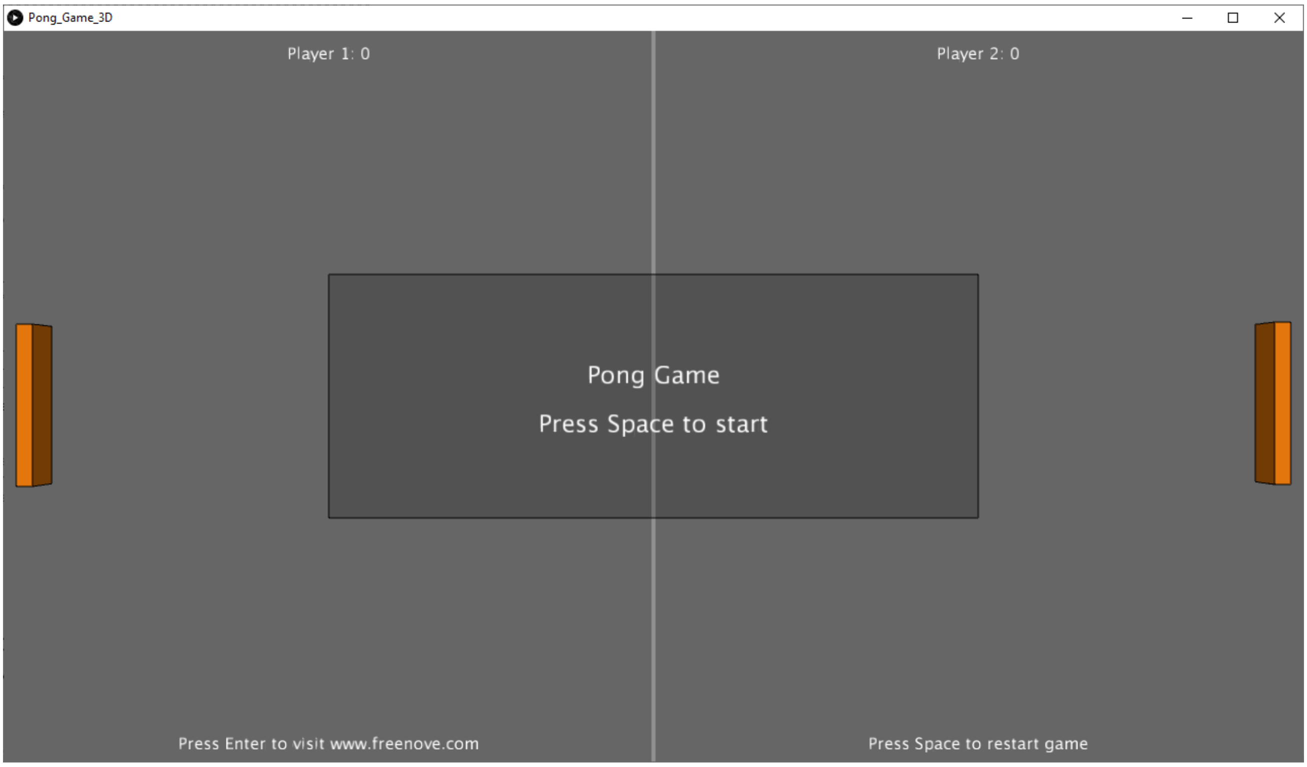
The same as last section.

Sketch

Sketch Pong_Game_3D

Use Processing to open **Freenove_Ultimate_Starter_Kit_for_Raspberry_Pi_Pico\Python\Processing\Processing\Pong_Game_3D\Pong_Game_3D.pde** and click Run.

If the connection succeeds, it will show as follows:



Pong_Game_3D

Player 1: 0

Player 2: 0

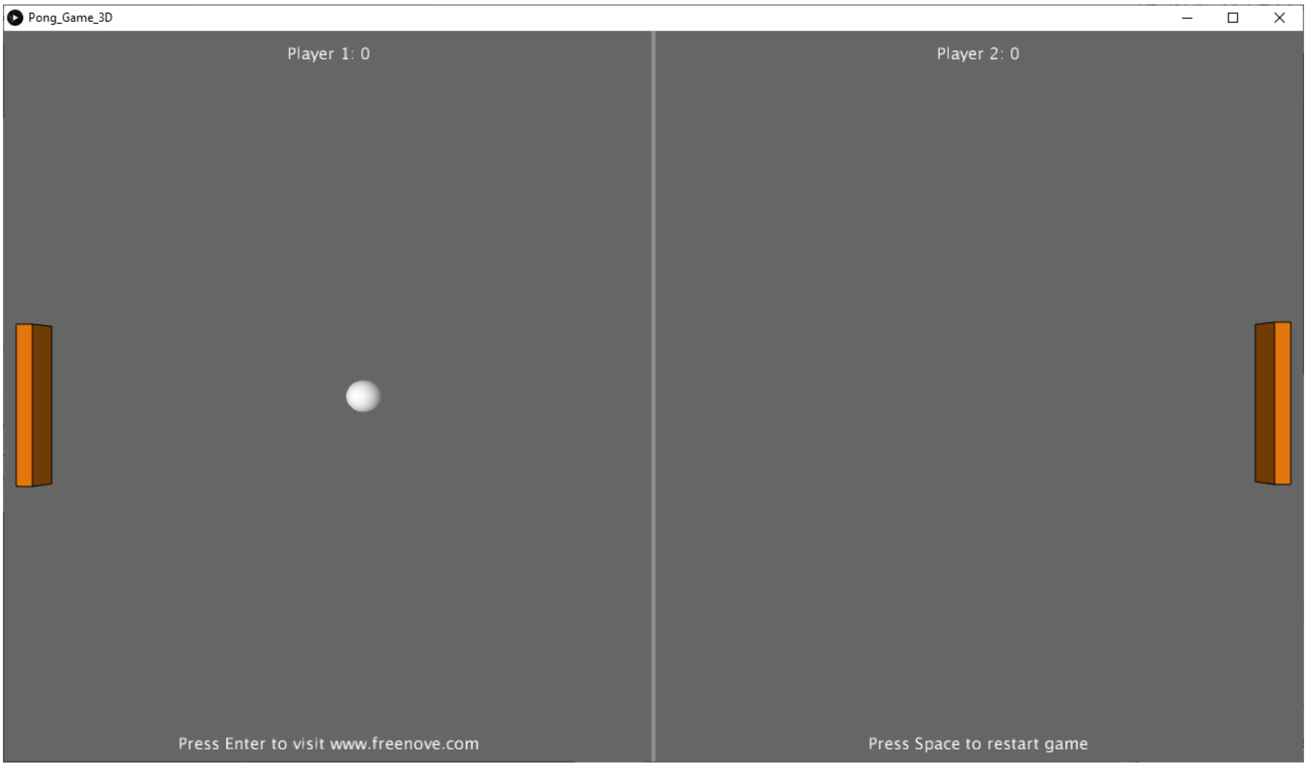
Pong Game

Press Space to start

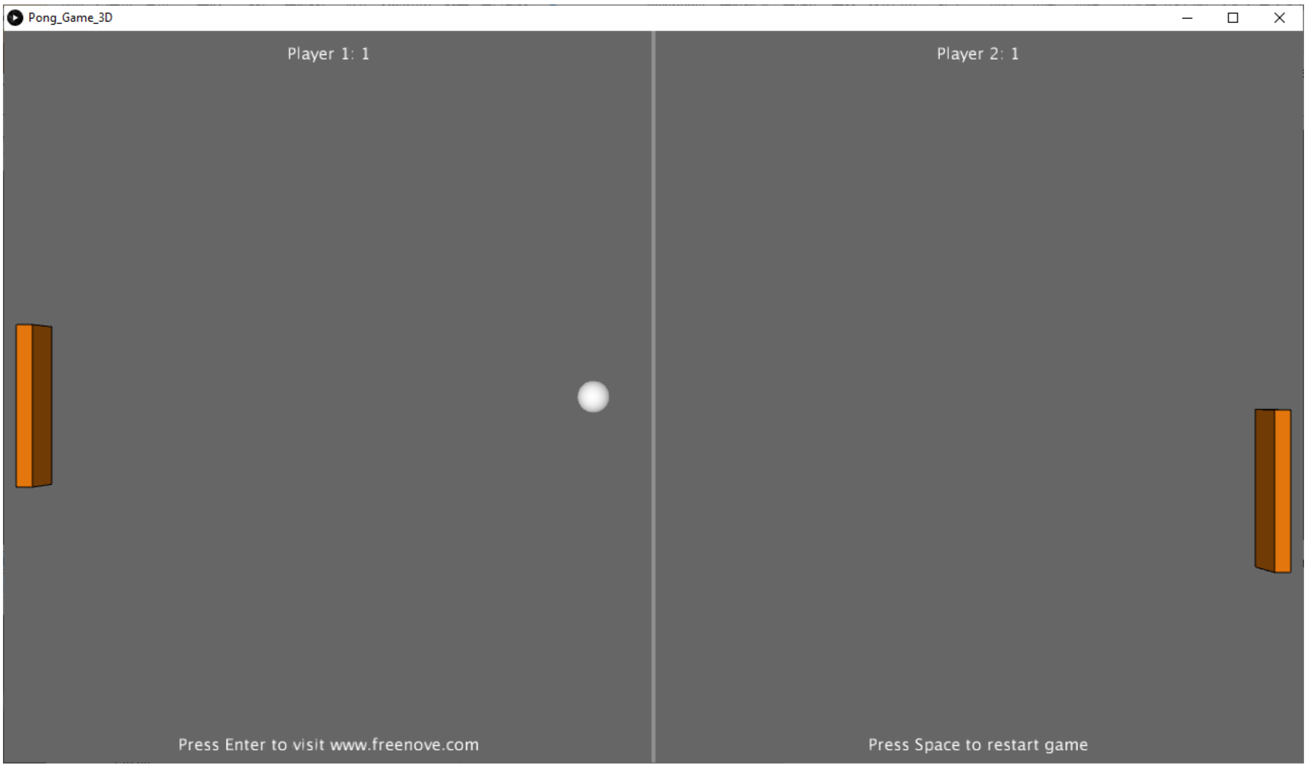
Press Enter to visit www.freenove.com

Press Space to restart game

Now you can try to rotate the potentiometer to control the movement of paddle without ball. Press space bar to start the game:


Pong_Game_3D
Player 1: 0
Player 2: 0
Press Enter to visit www.freenove.com
Press Space to restart game

Use potentiometer to control the movement of paddle to block the ball back. The game rules are the same as classic pong game:


Pong_Game_3D
Player 1: 1
Player 2: 1
Press Enter to visit www.freenove.com
Press Space to restart game

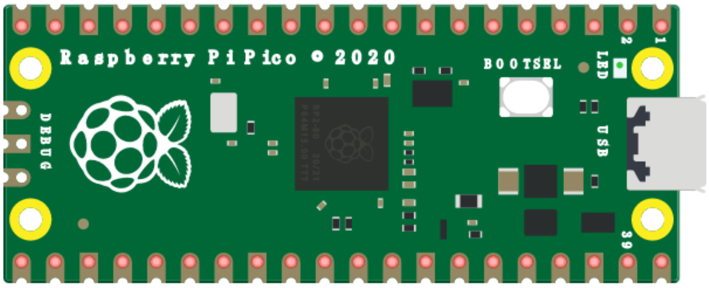
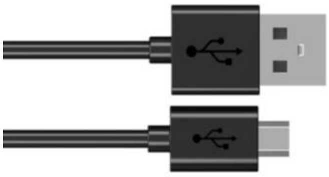
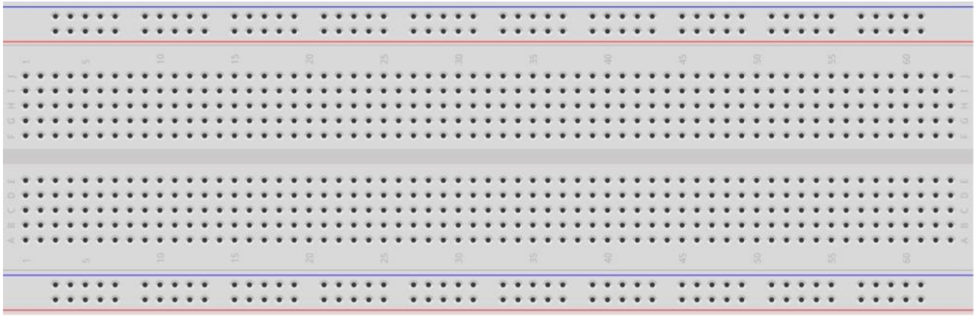
The rest operation is the same as the 2D version.

Chapter 6 Tetris

In this chapter, we prepare a tetris game for you. You can play the game by pressing the buttons.

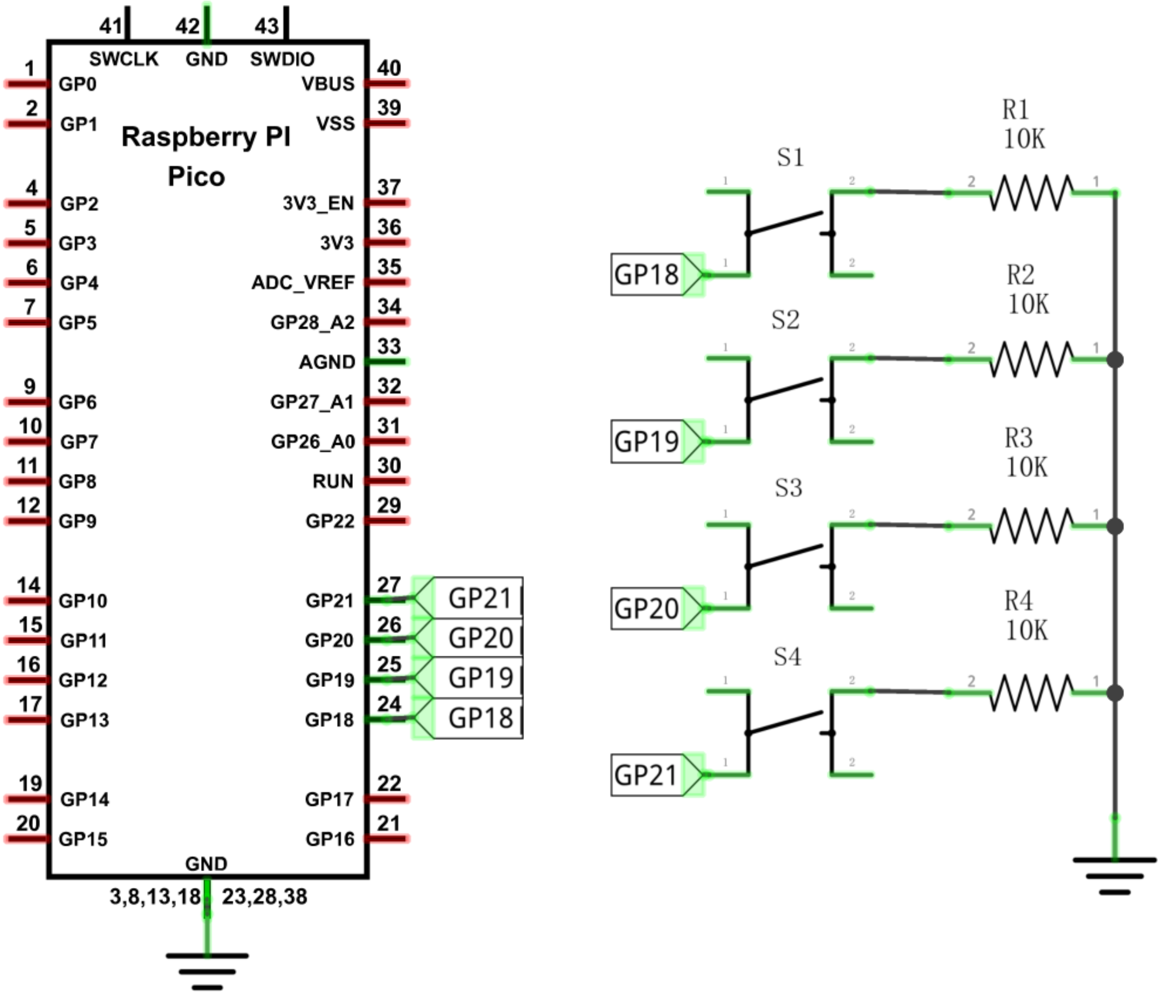
Project 6.1 Tetris

Now, let's use Processing and Raspberry Pi Pico to achieve the tetris game.

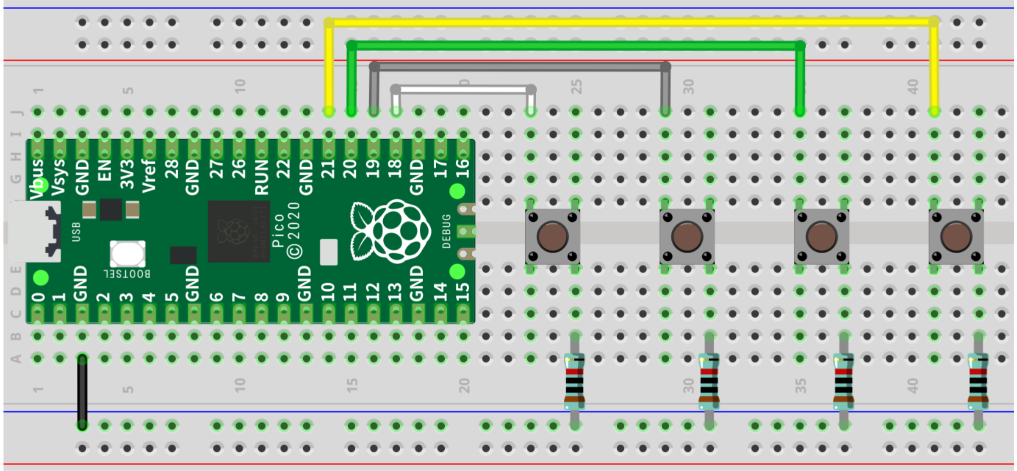
<table>
<tr><td colspan="2">Raspberry Pi Pico x1
 </td><td colspan="2">USB cable x1
</td></tr>
<tr><td colspan="4">Breadboard x1
</td></tr>
<tr><td>Jumper
 </td><td colspan="2">Resistor 10kΩ x4
 </td><td>Push button x4
 </td></tr>
</table>

Circuit

Schematic diagram



Hardware connection. If you need any support, please feel free to contact us via: support@freenove.com

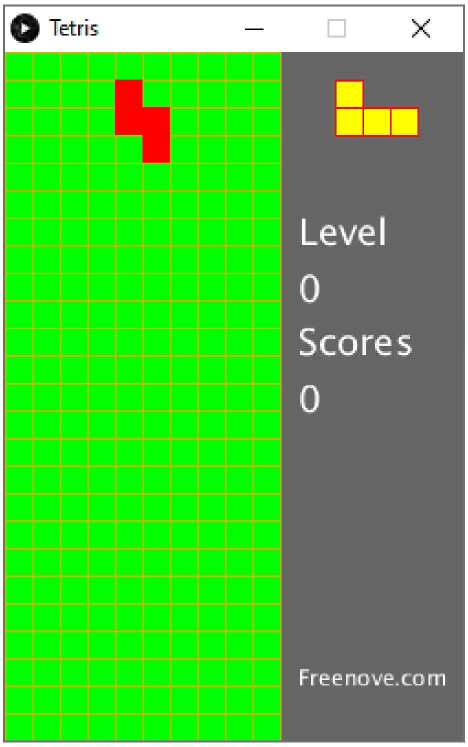


Note: To help users have a better experience when doing the projects, we have made some modifications to Pico's simulation diagram. Please note that there are certain differences between the simulation diagram and the actual board to avoid misunderstanding.

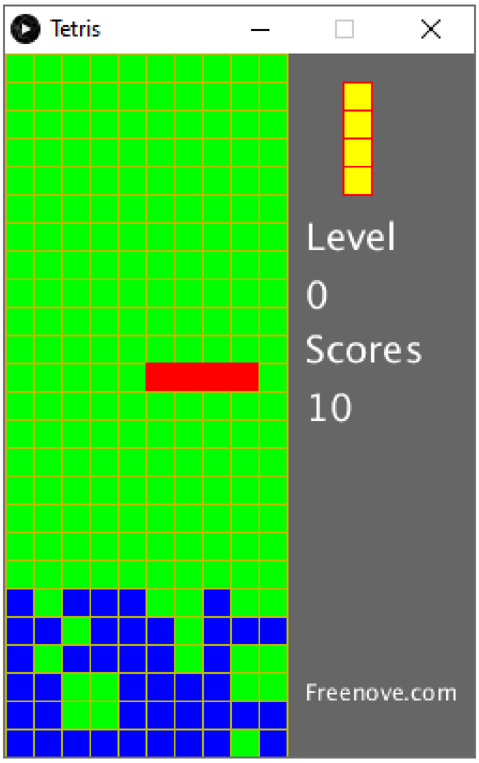
Sketch

Sketch Tetris

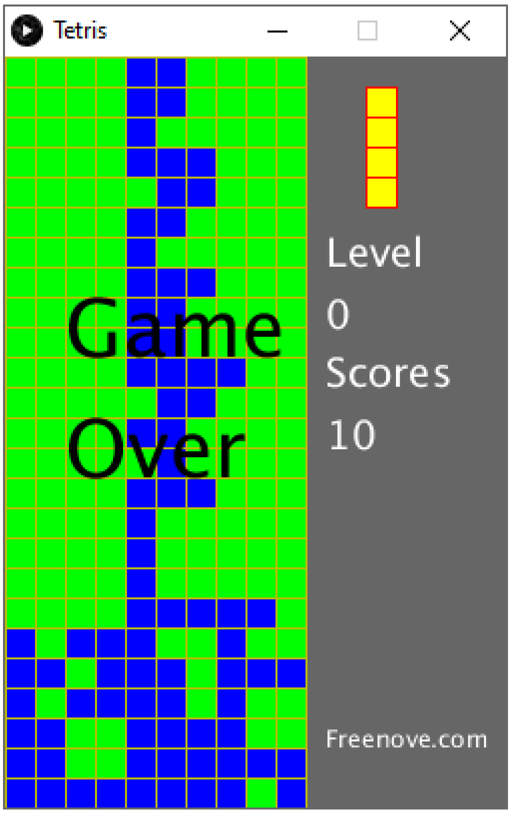
Use Processing to open Tetris.pde and click Run. If the connection succeeds, it will show as follows:



Now you can try using the buttons to control the falling block. Press the space bar to pause the game.



When the game is over, you can press the space bar to restart the game. Press ESC to exit the game.



What's Next?

THANK YOU for participating in this learning experience!

We have reached the end of this Tutorial. If you find errors, omissions or you have suggestions and/or questions about the Tutorial or component contents of this Kit, please feel free to contact us: **support@freenove.com**
We will make every effort to make changes and correct errors as soon as feasibly possible and publish a revised version.

If you want to learn more about Arduino, Raspberry Pi, Smart Cars, Robotics and other interesting products in science and technology, please continue to visit our website. We will continue to launch fun, cost-effective, innovative and exciting products.
http://www.freenove.com/

Thank you again for choosing Freenove products.