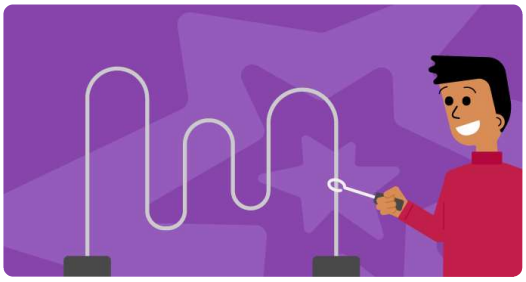
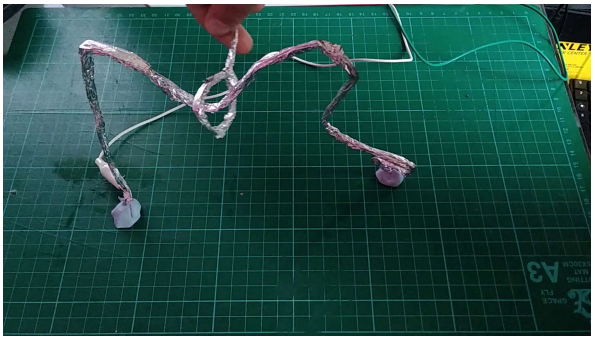
A wire loop game

Make a classic wire-loop game with Scratch and a Raspberry Pi



Step 1 You will make

Make a classic **wire loop game** (https://en.wikipedia.org/wiki/Wire_loop_game), using simple materials and a Raspberry Pi Computer, with Scratch 3.



What you will need

Hardware

- A Raspberry Pi computer
- A buzzer
- 5 × pipe cleaners
- 2 × socket-pin jumper wires
- 2 × socket-socket jumper wires
- 2 × crocodile clip wires
- Aluminium foil
- Plasticine or BluTack

Software

- Scratch 3 Desktop

What you will learn

- How to use a buzzer with a Raspberry Pi computer and Scratch
- How to use a button to sound a buzzer
- How to use a variable to keep a score

Additional information for educators

You can download the completed project **here** (http://rpf.io/p/en/rpi-wire-loop-game-scratch-get).

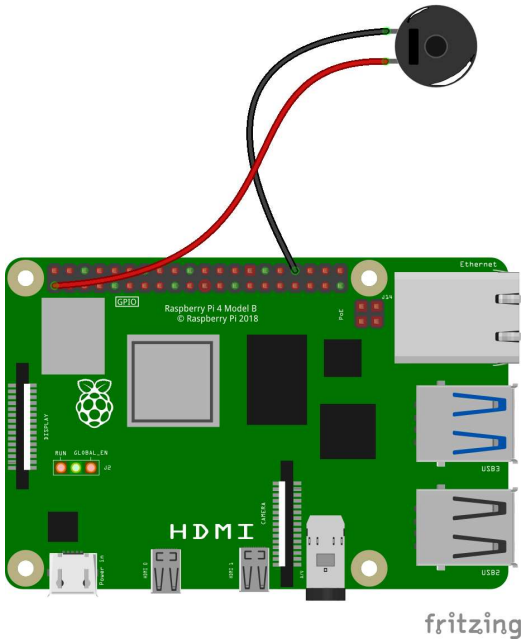
If you need to print this project, please use the **printer-friendly version** (https://projects.raspberrypi.org/en/projects/rpi-wire-loop-game-scratch/print).

Step 2 Wire and test a buzzer

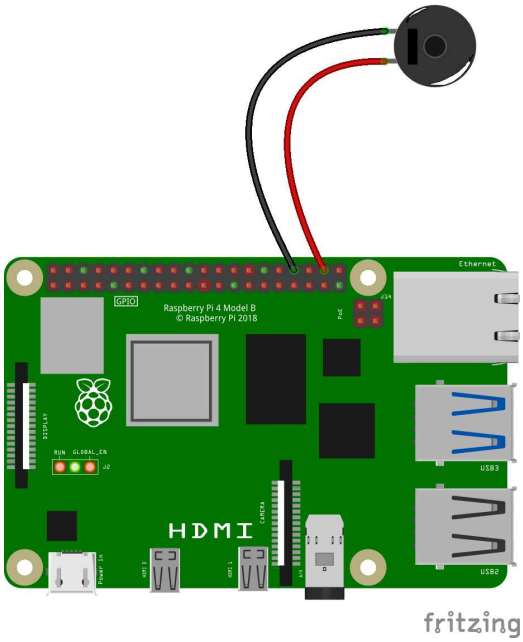
In this step, you will wire up a buzzer and control it with Scratch.

An active buzzer works just like an LED. It will make a sound when it is turned on and the sound will stop when the buzzer is turned off.

Use two socket-socket jumper wires to attach your buzzer to your Raspberry Pi device. The long leg of the buzzer must be wired to **3V3** and the short leg of the buzzer should be wired to a **GND** (ground) pin. The buzzer should sound straight away, so you know that it works.



Move the wire from the **3V3** pin to **GP20** as shown in the diagram below, and the buzzer should stop sounding.



Open Scratch Desktop on your Raspberry Pi and add the Raspberry Pi `Simple Electronics` extension.

From the Raspberry Pi `Simple Electronics` extension, find `toggle LED 0` in the menu, and use the drop-down menu to change `0` to `20`.

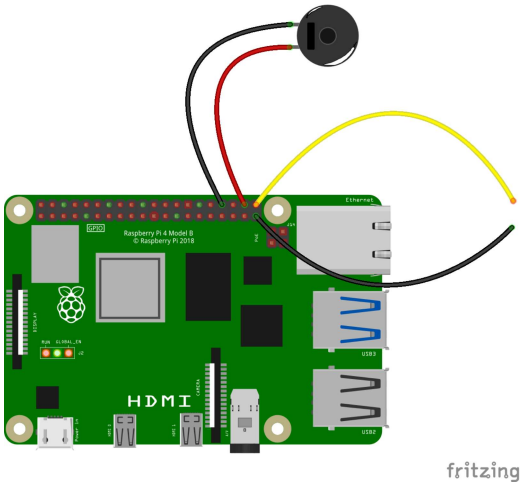
Click on the block and your buzzer should sound. Click again and your buzzer should turn off. It doesn't matter that the block says LED, because buzzers and LEDs work in the same way. They're either on or off.

Save your project

Step 3 Add a button

In this step, you will add a simple button, which will control the buzzer.

Add a Pin-Socket jumper wire to **GP21** and another Pin-Socket wire to a second **GND** pin.



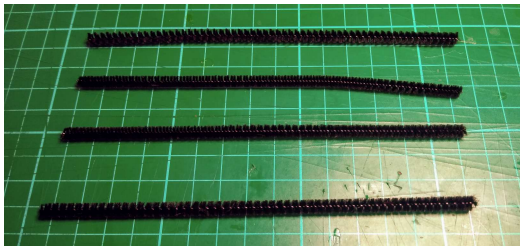
From the Raspberry Pi `Simple Electronics` extension, add the following blocks to control your buzzer.

Run your code and then touch the two M-F jumper wires together. The buzzer should sound. When you take them apart, the buzzer should stop.

Step 4 Build your wire loop game

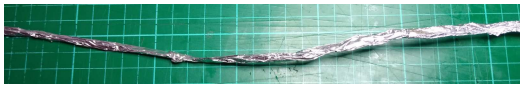
In this step, you will build the physical wire loop game to use with your existing Scratch code.

First, you will need four pipe cleaners.



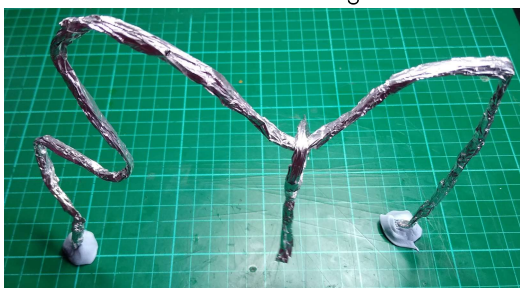
Twist together the ends of the pipe cleaners, so that all four are attached to form one long pipe cleaner.

Wrap the pipe cleaners in a piece of aluminium foil, so that they are covered from end to end.

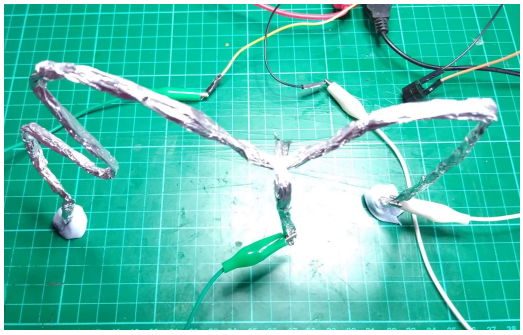


Take another pipe cleaner, wrap it in foil, and then bend it into a loop.

Take the long, wrapped set of pipe cleaners and bend them into an interesting shape. Thread the wire loop you made onto the long wire.
Then, use Plasticine or BluTack to attach the ends of the long wire to a suitable base.



Use a crocodile clip lead to attach one end of the long wire to one of the Pin-Socket jumper wires, and another crocodile clip lead to attach the handle of the wire loop to the other Pin-Socket jumper wire.

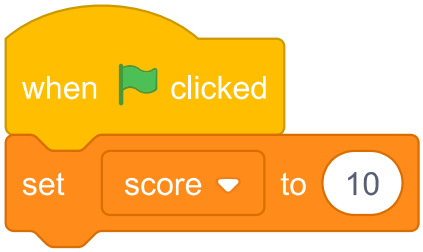


Run your Scratch program, and then try to move the wire loop along the long wire without them touching. Each time they touch, the buzzer should sound.

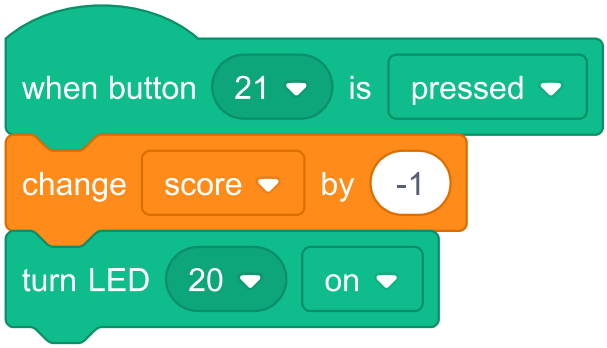
Step 5 Add a score

Now that you have a working game, you can add some code to keep a score of how the player is doing.

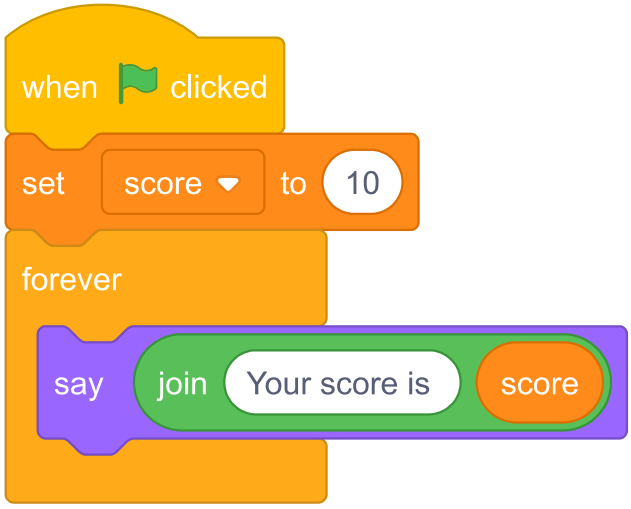
Create a new variable and call it `score`. When you run the program, the score should start at `10`.



Each time the loop touches the wire, the score should decrease by `1`. Alter the section of your code that starts with `when button 21 is pressed` so that it looks like this:

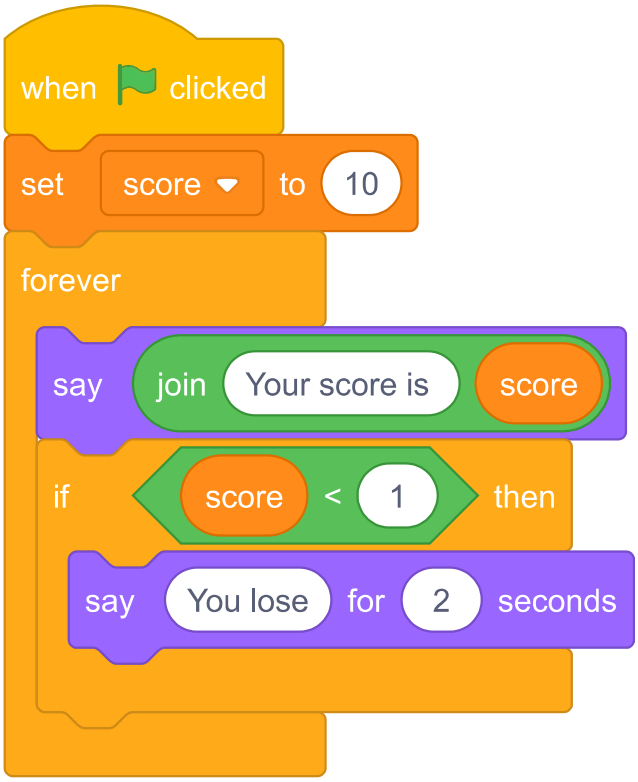


Now you can have **Sprite1** report the score by using a `forever` loop:



Run your program and **Sprite1** should tell you the score, and each time you touch the wire it should decrease.

Add an `if ... then` block to your code, so that you can tell the player that they have lost when their score is less than `1`.



Run your program and then play your wire loop game. Every time you touch the wire, your score should drop.

Save your project

Upgrade your project

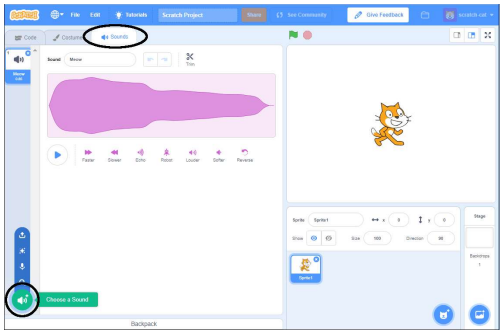
Now, you could add sound effects to your wire loop game to play when the score drops to a certain level, to warn the player they are about to lose.

First, choose a sound from the sounds library in Scratch to act as a warning sound and add it to Sprite1.

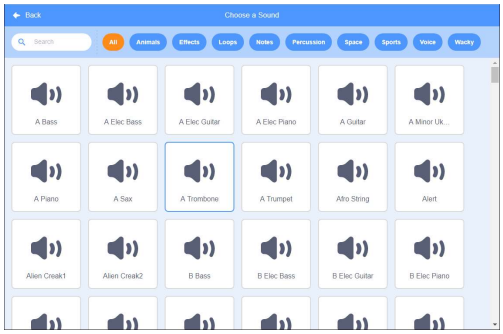
Adding a sound from the library

- Select the sprite you want to add the sound to.

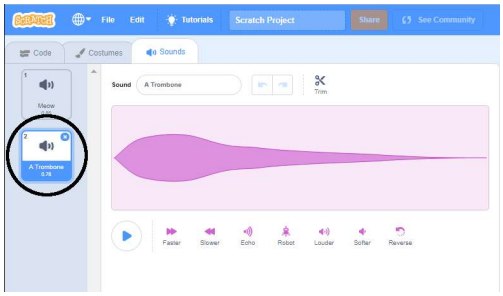
- Click the **Sounds** tab, and click **Choose a Sound**:



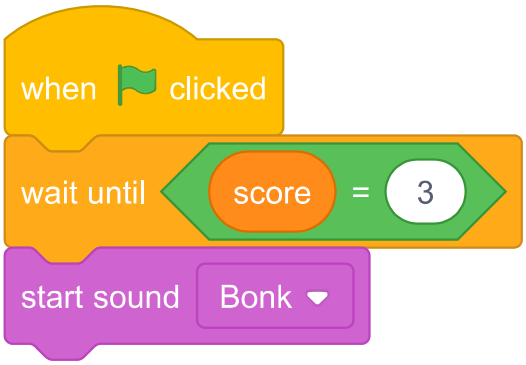
- Sounds are organised by category, and you can hover over the icon to hear a sound. Choose a suitable sound.



- You should then see that your sprite has your chosen sound.

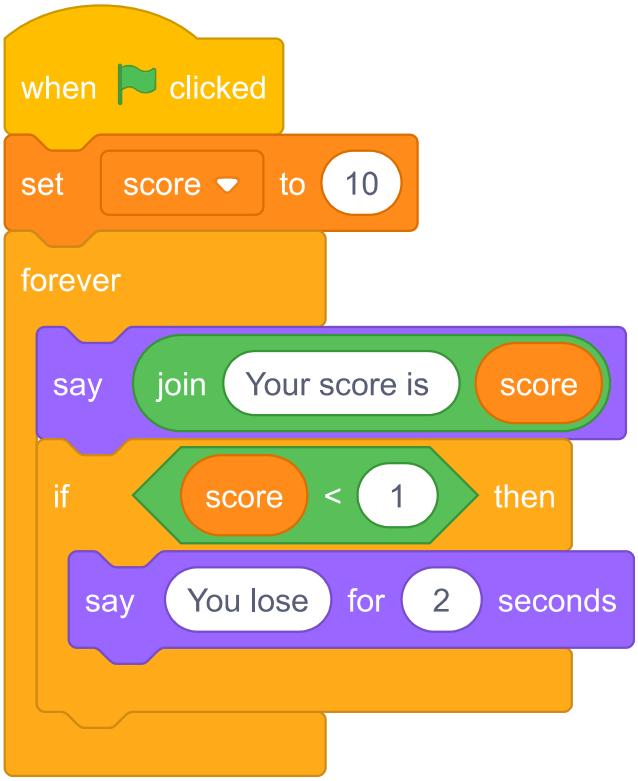


On Sprite1, create a new script that looks like this:

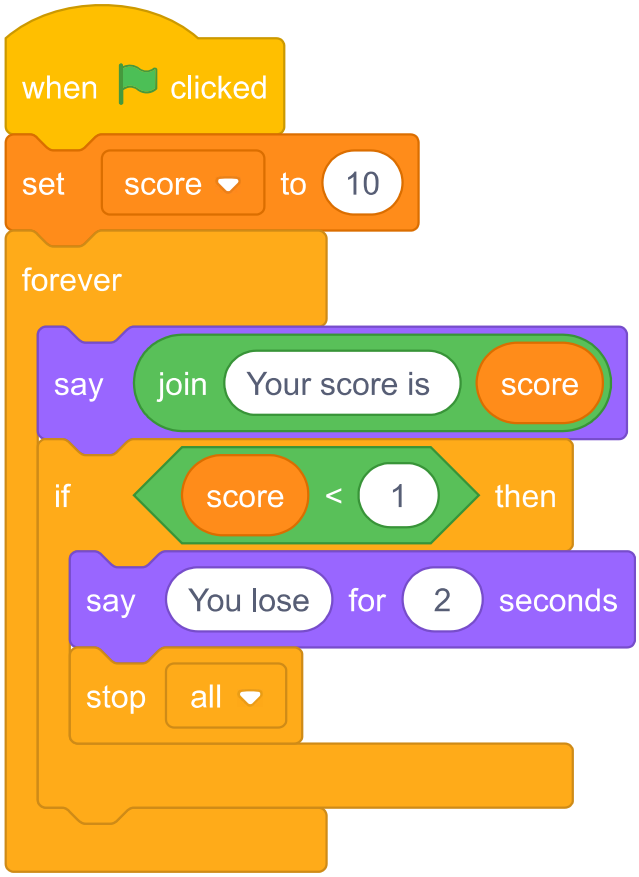


You can also add a 'lose state' which will end the game if your score drops to 0.

On Sprite1, find the code that controls the score:



At the very end of the script, simply add a `stop all` block to the end:



This will stop the game from playing once the score reaches 0.

If you wanted to further upgrade your wire loop game, you could get some more robust materials to create it from!

Upgraded Wire Loop Game:

This example has used a lasercut box for the enclosure, while the shaped wire and loop are both made from a metal coat hanger bent into shape using pliers. The handle of the loop comes from a recycled paint roller, while the sound is made by a small USB speaker powered by the Raspberry Pi.

What next?

If you are following the **Physical computing with Scratch and the Raspberry Pi path** (https://projects.raspberrypi.org/en/pathways/physical-computing-with-scratch-and-the-raspberry-pi) pathway, you can move on to the **3D science display project** (https://projects.raspberrypi.org/en/projects/scratch-3d-science) project. In this project, you will make a 3D LED science display.

If you want to have more fun exploring Scratch, then you could try out any of **these projects** (https://projects.raspberrypi.org/en/projects?software%5B%5D=scratch&curriculum%5B%5D=%201).
