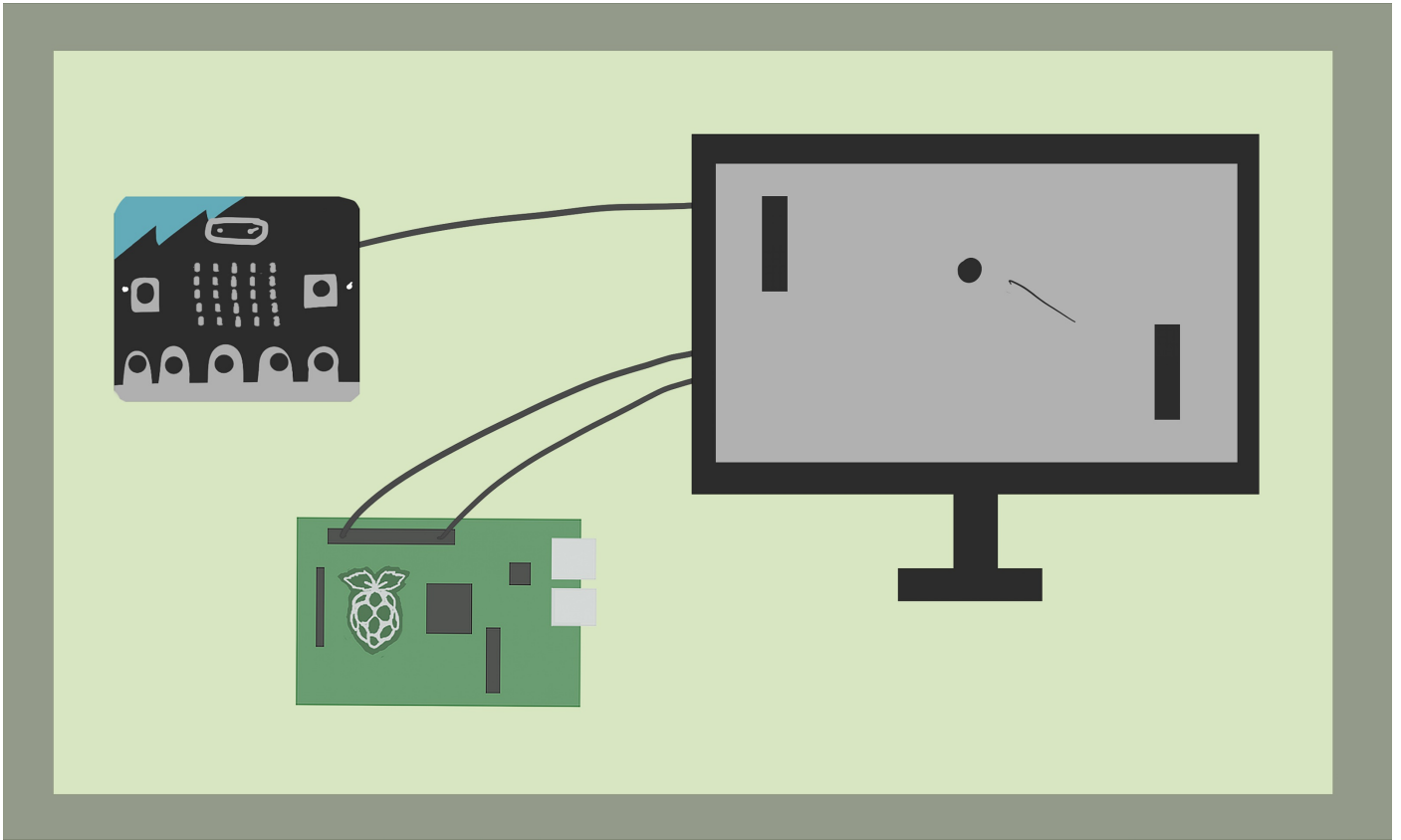


Micro:bit Scratch Controller



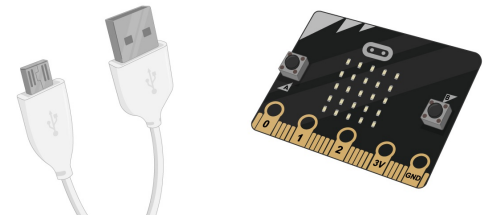
What are we going to do in this worksheet?

In this worksheet you'll be learning how to take the values from a Micro:bit and use them in your own game of Pong in Scratch! A Micro:bit is a little computer that has loads of chips on it, one of which is an accelerometer, it can detect acceleration in three different directions, these are called the 3 axis' and they are the x, y and z axis.

What you will need

- Micro:bit
 - MicroUSB cable
-

Running the Micro:bit Script



If you go into the shared resources folder, by going to the two folders at the top-left of your screen, beside the terminal button. On the left hand side you should see all of your folders, click on Shared resources and copy the MicroBitControl.py file to your desktop or home folder by dragging.

Opening Python and running the script



To open the Python 3 shell (IDLE3) we will first need to click the terminal icon on the top bar (highlighted by the box in the picture above). Once terminal has opened type in `sudo idle3` and press enter. Python IDLE should now open.

Once IDLE has opened, find where you move the MicroBitControl.py script to and open it. Once you have your Micro:bit connected to the Pi, you can run it.

Opening and setting up Scratch

Go to the Pi's desktop and open `ScratchGPIO7`, if it isn't there, try and click the `"Install ScratchGPIO"` icon. After a moment `ScratchGPIO7` should appear on the desktop. When you open Scratch GPIO, a pop-up window should say

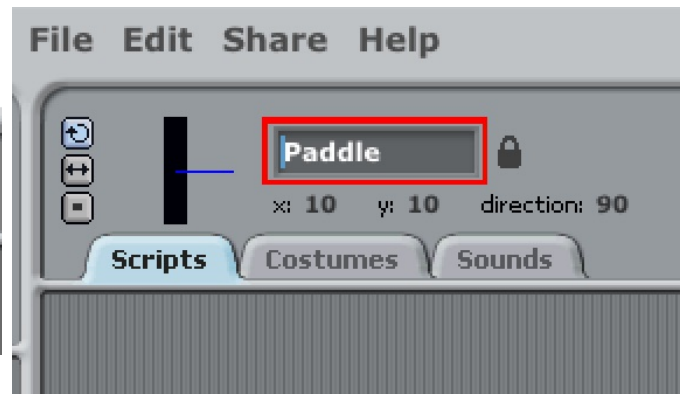


ScratchGPIO 7

`"Remote sensor connections enabled"`. Open a new project, this will be where all our code goes to make our mini-game.

Making the Paddle

We'll start off with making the sprite, we'll do this by pressing the paint new sprite button below the stage. Our paddle will just be a tall rectangle and once you have made the sprite, just it's name to "Paddle" by using the textbox at the top left of the screen.



We will start the program off using the green flag which will call a forever loop which will hold all of our code for the paddle.

We will be controlling the sprite using our Micro:bit's accelerometer values from the Sensing tab. To use the values we take out a

`_ sensor value` block and set it to get the

value of the `x` axis but divide it by one hundred, by using the division block from the operators tab, to make sure the sprite doesn't fling itself across the stage.



Then we will put the blocks together and put them into a `change y by _` block. By the end, you should have something like this,



Now if you run the script with the Micro:bit attached, the paddle should move when you move your Micro:bit!

Making the End Wall

The end wall will be a rectangle like the paddle, but this one will be an obstacle for the player to make sure the ball doesn't hit or else the game will end. You can make it the same way that the paddle was made but a bit longer with a different colour, and this time the name will be "Wall".

Making the ball

The final thing that we will be using is a ball for our paddle to bounce and project from the wall. For this sprite you'll just make another sprite and then make it another different colour. We'll call this "Ball".

To start off this program, we'll use the green flag as well and it will also go into a forever loop like our paddle. Before our forever loop we're going to put in a **turn _ degrees** block to make sure the ball goes off at an angle.

The first bit of code to go into our loop is a **move _ steps** block that is set to 5 steps. You can find this block in the Motion tab.



then we'll use an **if on edge, bounce** block to make it automatically bounce when it hits an edge.



You can run the code now and it the ball should move and then bounce from one end to another!

Once you know that the ball moves and bounces around the stage, we can get the other two bits of code done. The first bit it the code that will end the game when we hit to wall. We're going to program it to say "Game over" and move to the center of the screen.



To check **if** we have hit the wall sprite, we can use an if block with the condition being if it's touching the wall, which we can check using the **touching _** block, and we'll set it to wall and put it into our if loop.

To say "Game over" we'll use the **say _** block from the Looks tab. Then all you need to do is change the text to "Game Over".

Then to make it go the center of the screen we can use the **got to x:_ y:_** block. But to make it so that the ball stays in the center of the screen, we'll put it into a forever loop.



Now put this all together into the **if** block that we created before for the **if touching wall**.

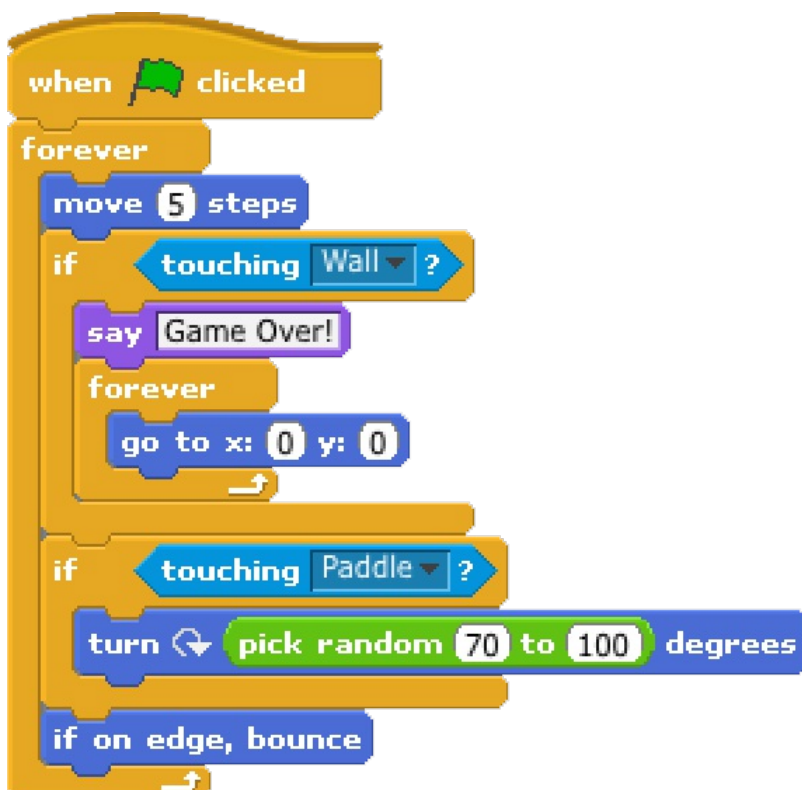
Now we need to use another **if** block and to check if the ball is touching the paddle. If it is, then we will make the ball turn randomly.



To make the ball turn randomly, we'll use the **turn _ degrees** block and then a new block called **pick random _ to _**. The two boxes are for putting the smallest and the biggest value that the random number can be. It can be found in the Operators tab and we'll make the smallest 70, and the biggest 100



Now put it all together and you should have something like this...



Now plug in your Micro:bit and run!

Now try: Changing the random values and see what happens!

What next?

- Why not reset the Sprite's position when you press both the A and B button on the Micro:bit
- Make a space shooter by changing the Sprite's costume and the stage background and coding targets
- Make the Micro:bit control a car (Maybe from the "A day at the races" worksheet?)