

KidzCanCode

IGNITE, INSPIRE, INNOVATE

www.kidzcancode.com

Ping Pong

KIDZCANCODE

TREVOR WARREN

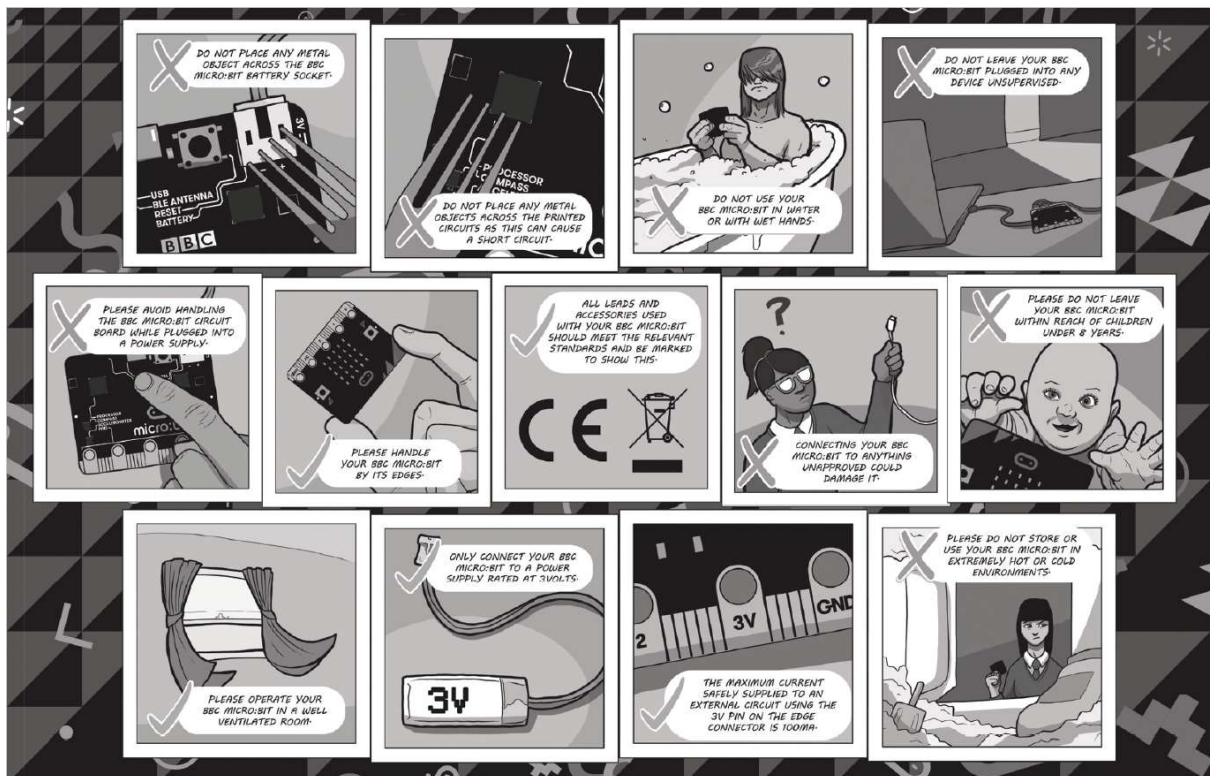
Tutorial 2.1x – Ping Pong

1. Safety Warnings

THE MICRO:BIT IS AN EXPOSED BOARD, TO BE USED WITH CARE PLEASE RETAIN THIS INFORMATION FOR FUTURE REFERENCE. You can read the detailed document at - <http://microbit.org/guide/safety-advice/>

a. General Safety Warnings

Using the BBC micro:bit is easy to use but is designed to have all the electrical parts on display. This does mean there's a small risk that the parts can be damaged and even overheat with a risk of injury but a little bit of care and caution will ensure you and your micro:bit will stay fit and healthy.

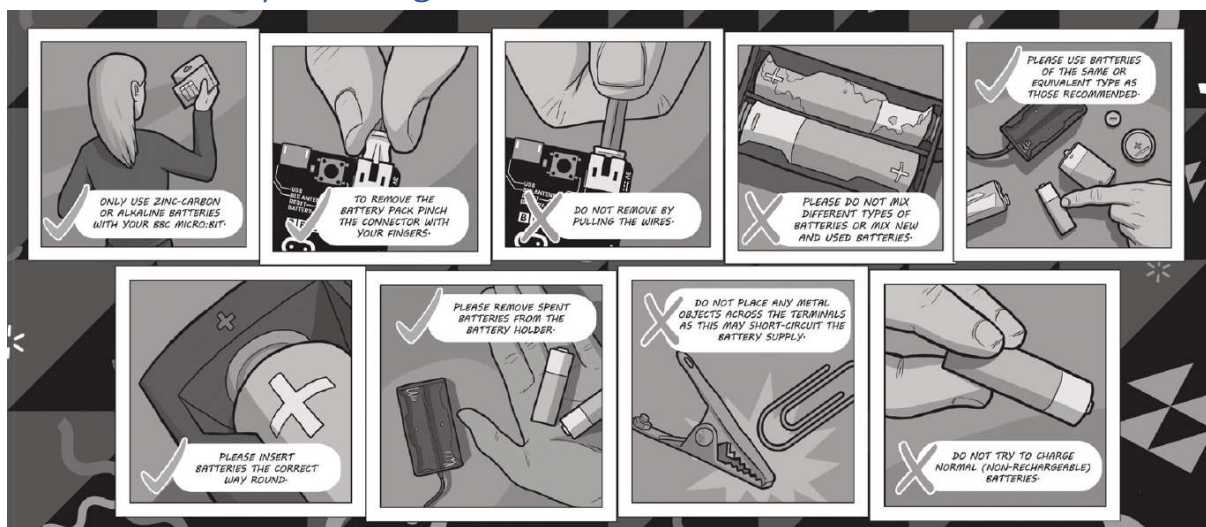


1. Always keep your BBC micro:bit in the anti-static bag when not in use. It's good practice for students to earth themselves before handling it.
2. Please handle your BBC micro:bit by its edges. This minimises the risk of damage through an electrostatic discharge.
3. Please use the battery pack and the USB lead provided to power your micro:bit. Do not use portable battery chargers or USB charging ports (often marked with a lightning bolt or 'SS'), to power your micro:bit. Using these may damage your micro:bit and stop it working properly.
4. Please avoid handling the BBC micro:bit circuit board while plugged into a power supply.
5. All peripherals (for example: USB cable, battery holder, sensors) used with your BBC micro:bit should comply with the relevant standards and should be marked accordingly.
6. Connecting your BBC micro:bit to any unapproved peripherals could damage your BBC micro:bit
7. Please do not attempt to keep using faulty micro:bits. If a school-issued micro:bit develops a fault, contact the vendor immediately.

Tutorial 2.1x – Ping Pong

8. The maximum current safely supplied to an external circuit using the 3V pin on the edge connector is 100mA. Please make sure this limit is not exceeded.
9. Please do not store or use your BBC micro:bit in extremely hot or cold environments.
10. Do not place any metal objects across the printed circuits on the board as this can cause a short circuit damaging your BBC micro:bit. This can cause risk of burn or fire.
11. Do not use your BBC micro:bit in water or with wet hands.
12. Do not leave your BBC micro:bit plugged into a computer or any other device unsupervised.
13. Please do not leave your BBC micro:bit within reach of children under 8 years of age.
14. Please operate your BBC micro:bit in a well ventilated room To remove the battery pack, pinch the connector with your fingers. Do not remove by pulling the wires.

b. Battery Warnings



1. Do not try to charge normal (non-rechargeable) batteries
2. Please do not mix different types of batteries or mix new and used batteries.
3. Please use batteries of the same or equivalent type as those recommended.
4. Please insert batteries the correct way round (with the correct polarity).
5. Please remove spent batteries from the battery holder.
6. Do not short-circuit the battery supply terminals, for example by placing a metal object across the terminals.
7. Only use Zinc or Alkaline batteries with your BBC micro:bit.
8. Please do not use rechargeable batteries

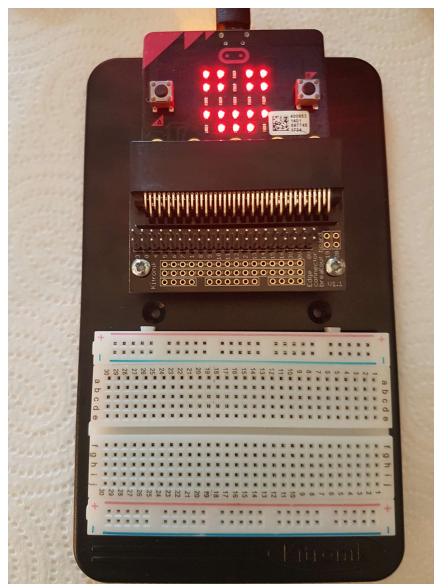
Tutorial 2.1x – Ping Pong

2. Pre-requisites

If you have questions with the assembly of the micro:bit, edge connector breakout board, mounting board and the breadboard please drop us a note at help@kidzcancode.com. The edge connector board, mounting board and the breadboard are part of the Kitronix Inventors kit which needs to be purchased separately.

To be able to perform this tutorial you will need the following components –

1. Parts required –
 - a. 1 x BBC Micro:bit
 - b. 1 x Mounting Plate
 - c. 1 x Edge connector breakout board
 - d. 1 x Bread board
2. Assembly required –
 - a. Bread board mounted on top of the mounting plate
 - b. BBC Micro:bit inserted into the Edge Connector breakout board



Before proceeding please check your setup and confirm that all the required parts are configured as demonstrated in the above picture.

Tutorial 2.1x – Ping Pong

3. Learning Objectives

The objectives of this tutorial are to introduce the student to the following concepts –

- Understanding the use of custom functions for purposes of programming
- Declaring variables and assigning values to variables at different times of execution in the program
- Coding for use of the buttons on the micro:bit, perform a given action when a button is pressed.
- Understanding nested loops
- Calling functions in a recursive manner without using the forever code block
- Working through the strategy for design of a game
- Working through the different components of the games, the functions required including interaction of the different functions with each other

The BBC micro:bit is a powerful little computer. Through programming these games kids explore more advanced computer science concepts. Along the way kids are encouraged to share, create and extend the games using their own imagination and creativity.

This tutorial builds upon concepts introduced in previous tutorials so please make sure you have covered the previous tutorials before you dive into this one. So overall this tutorial intends to build upon concepts learnt in previous tutorials while exploring new concepts.

In future tutorials we will continue to build upon the concepts learned here and will build more complex interactive games using the functionality provided by the micro:bit.

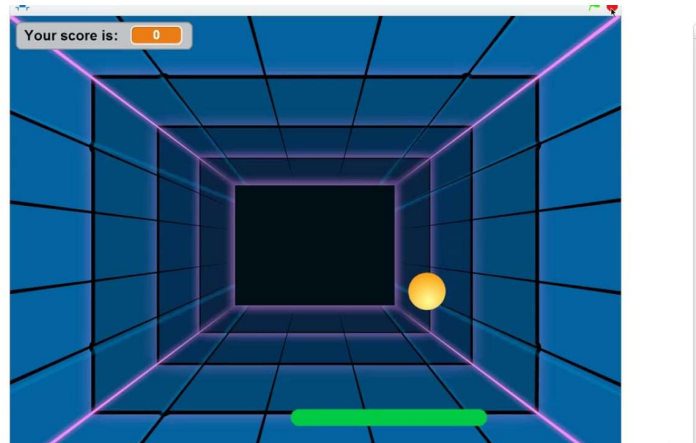


Tutorial 2.1x – Ping Pong

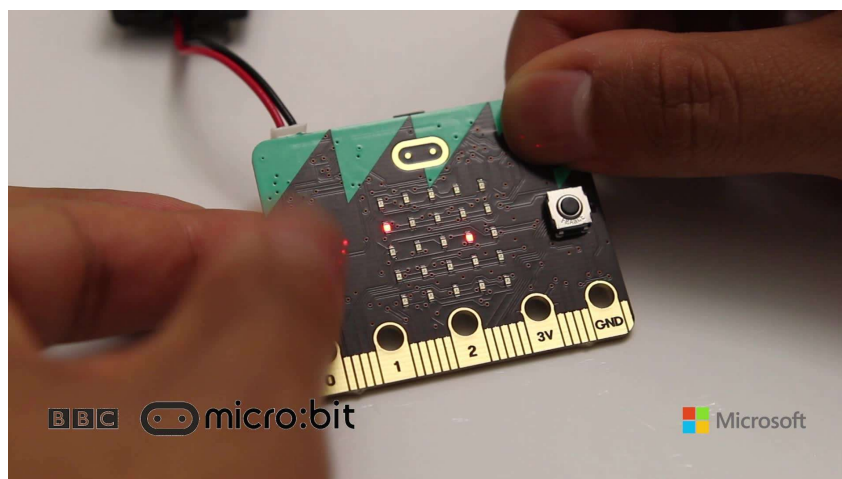
4. Activity

a. Activity

If you have played with the pong game before you know where we are heading with this one. The pong game we are going to design for the micro:bit is very similar to the pong game we have designed and played with scratch. In-fact with scratch we've challenged ourselves and turned the pong game into a multiplayer game.



With the micro:bit we will attempt to put together a simple single player pong game. This tutorial is a challenging one requiring you to think through the approach for game design, the different functionality you will need to implement, your choice of the code blocks/functions you will create in the game and how these code blocks/functions interact with each other.



The activity is designed to have additional challenges which allow the developer to keep pushing the boundaries. The main challenges in this activity involve –

- You've got to code for a the pong ball to be moving across the screen like we did in scratch
- If the pong ball hits either of the sides of the screen it's deflected based on a random angle selected. Use your creativity (and math function) to choose the approach to randomly deflect the ball

Tutorial 2.1x – Ping Pong

- A bat is provided to the user which is to be used to prevent the pong ball from hitting the bottom of the screen
- The aim of the game is to make sure that the pong ball doesn't hit the bottom of the screen
- A counter is maintained with regards to how many times the user hits the ball before the pong ball hits the bottom of the screen.
- When the ball hits the bottom of the screen the game ends and the users points are displayed on the screen

This is an interesting game but also does get a lot more challenging in term of game design and programming logic. The pong game on micro:bit takes our learning to a whole new level by encouraging use of the micro:bit to develop an interactive game that can be played on a battery operated handheld device. You've come a long way starting out with pong on scratch, continuing your learning journey by coding the pong game on the micro:bit.

Best of luck and happy hacking!!!!



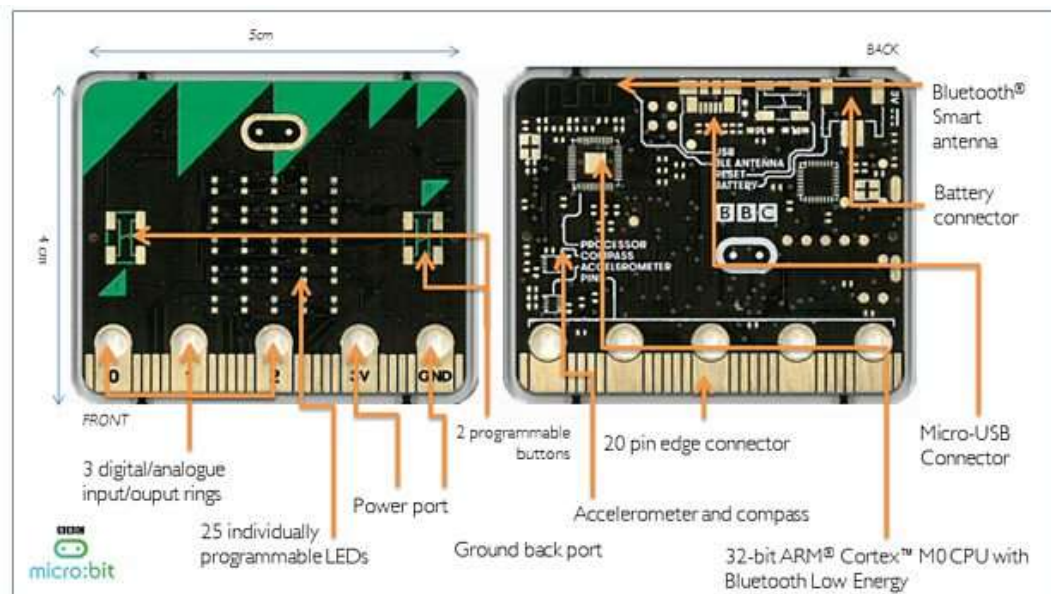
Tutorial 2.1x – Ping Pong

b. How Does It Work

This section talks about the BBC micro:bit and what it's made up of. If you have already read through this section then feel free to skip directly to the next section. The BBC micro:bit is a powerful little board and has various types of sensors on board. Here's what makes up the BBC micro:bit.

1. Size: approx. 5cm x 4cm.
2. Weight: 8g.
3. Processor: 32-bit ARM Cortex M0 CPU.
4. Bluetooth Low Energy.
5. Digital Compass.
6. Accelerometer.
7. Micro-USB controller.
8. 5x5 LED matrix with 25 red LEDs.
9. Pins for connecting external sensors, LED's, etc.

Here's what the micro:bit looks like –



Front of the board (left hand side)

1. Button A (left button with edge connector at the bottom) – labelled A on the board
2. Button B (right button with edge connector at the bottom) – labelled B on the board
3. P0 (left large pin (crocodile clip port) with edge connector at the bottom) - labelled 0 on the board
4. P1 (middle large pin (crocodile clip port) with edge connector at the bottom) - labelled 1 on the board
5. P2 (right large pin (crocodile clip port) with edge connector at the bottom) - labelled 2 on the board
6. +3V - labelled 3V on the board. This is 3V PWR OUT
7. GND
8. P3 – P22 pins from left to right with edge connector at the bottom. Referred to as Pins when referencing that part of the board. Text will talk about 'pins' when referring to

Tutorial 2.1x – Ping Pong

individual connections or the general way of connecting to the board – not labelled on the front of the board

9. LED matrix referred as the 'screen' - not labelled on the board
10. LED coordinates starting at 0,0 top left corner and ending at 4,4 at the bottom corner - not labelled on the board

The order of the large pins as follows: P0 P1 P2 3V GND labelled 0, 1, 2, 3V GND on the board

Rear of the board (Right hand side)

1. 1. USB Plug (Micro-USB plug) – labelled USB on the board
2. Button R (reset button) – labelled Reset on the board
3. Status LED – not labelled on the board
4. Battery socket – labelled Battery on the board

Other components on the board include

1. Accelerometer
2. Compass
3. Bluetooth Smart Technology Antenna
4. AAA Battery Holder - not labelled on the board
5. Processor (Cortex M0)

The BBC micro:bit is programmable in a few different languages. You can write code for the micro:bit using the Makecode block coding interface, Javascript, Python or even in C. Most of our tutorials will cover the use of the Makecode block coding interface built by Microsoft for the micro:bit.

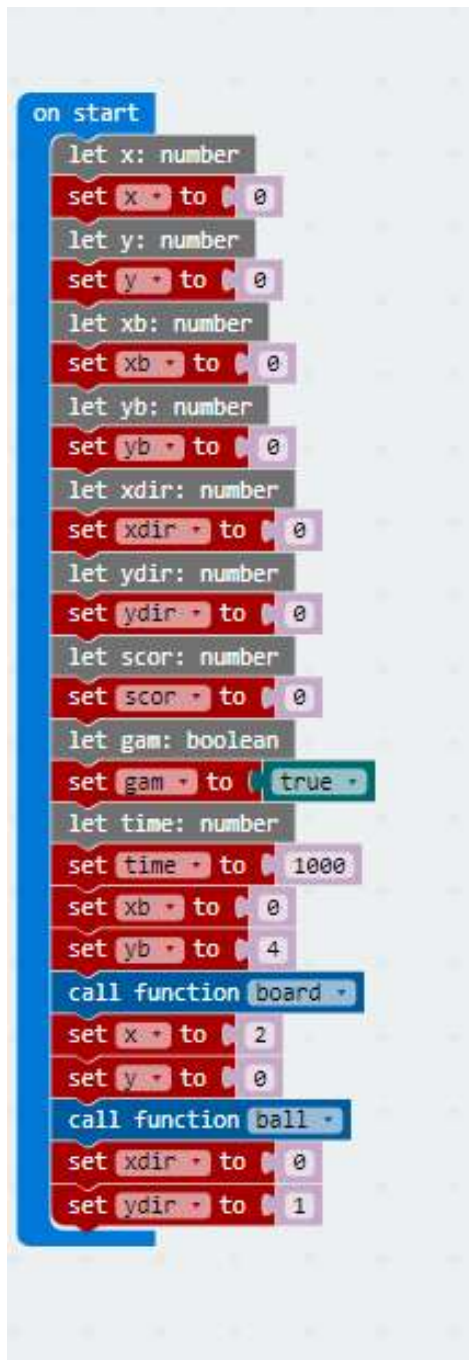


Tutorial 2.1x – Ping Pong

5. Let's write some code – Block Based

It's time to write some code and get going with coding our game. So let's head over to the micro:bit block code editor page (<https://makecode.microbit.org/>) and get coding!!! We have also provided the Javascript version of this game, so if you are interested you can start coding in javascript as well.

A downloadable hex file is provided at the end of this tutorial. So if you are keen to see the game in action before you can start coding then just head over to the end of this tutorial and use the hex file provided. We use the on-start function to initialize all the relevant variables and set them up for use across rest of the functions in the game.

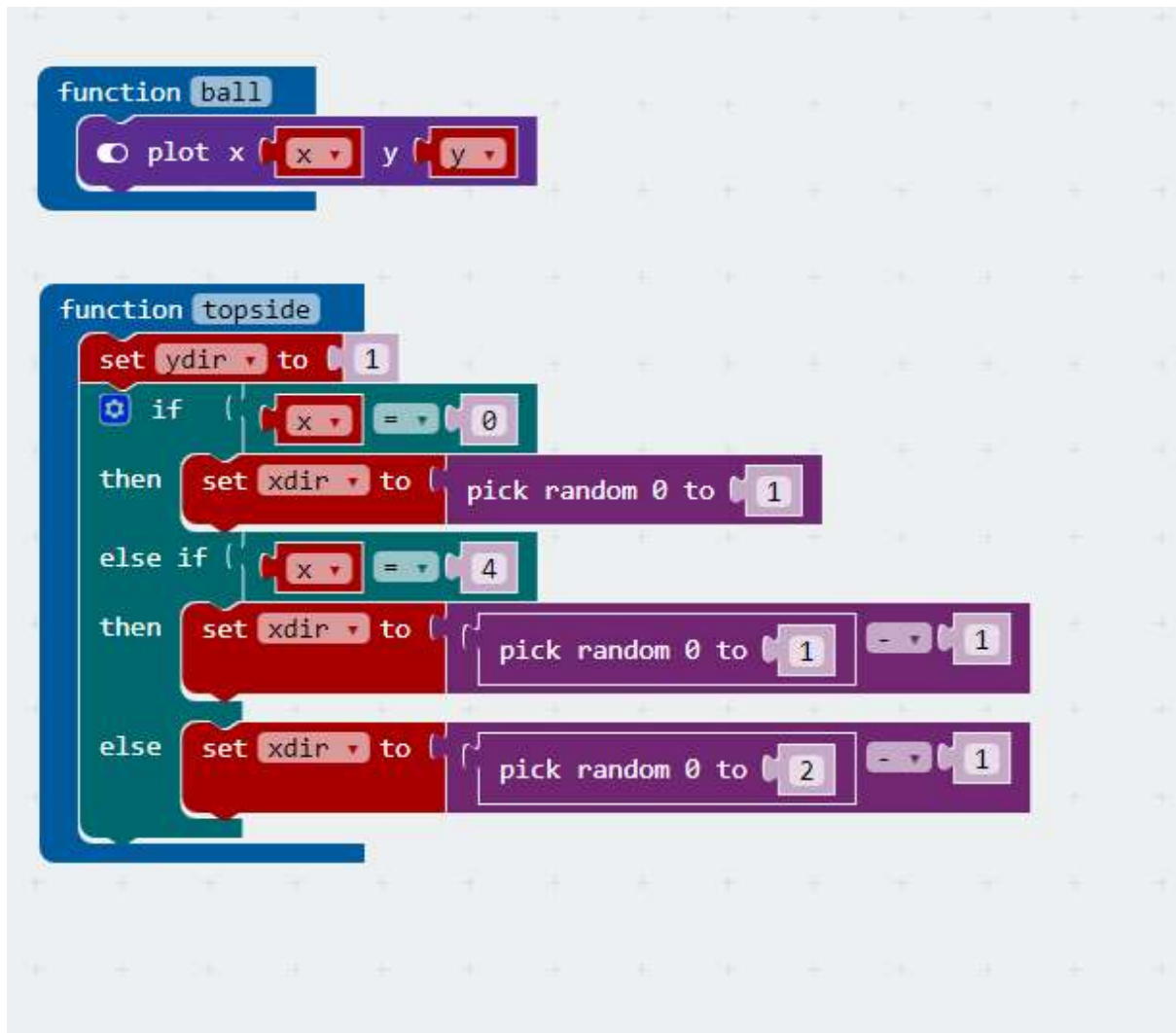


Tutorial 2.1x – Ping Pong

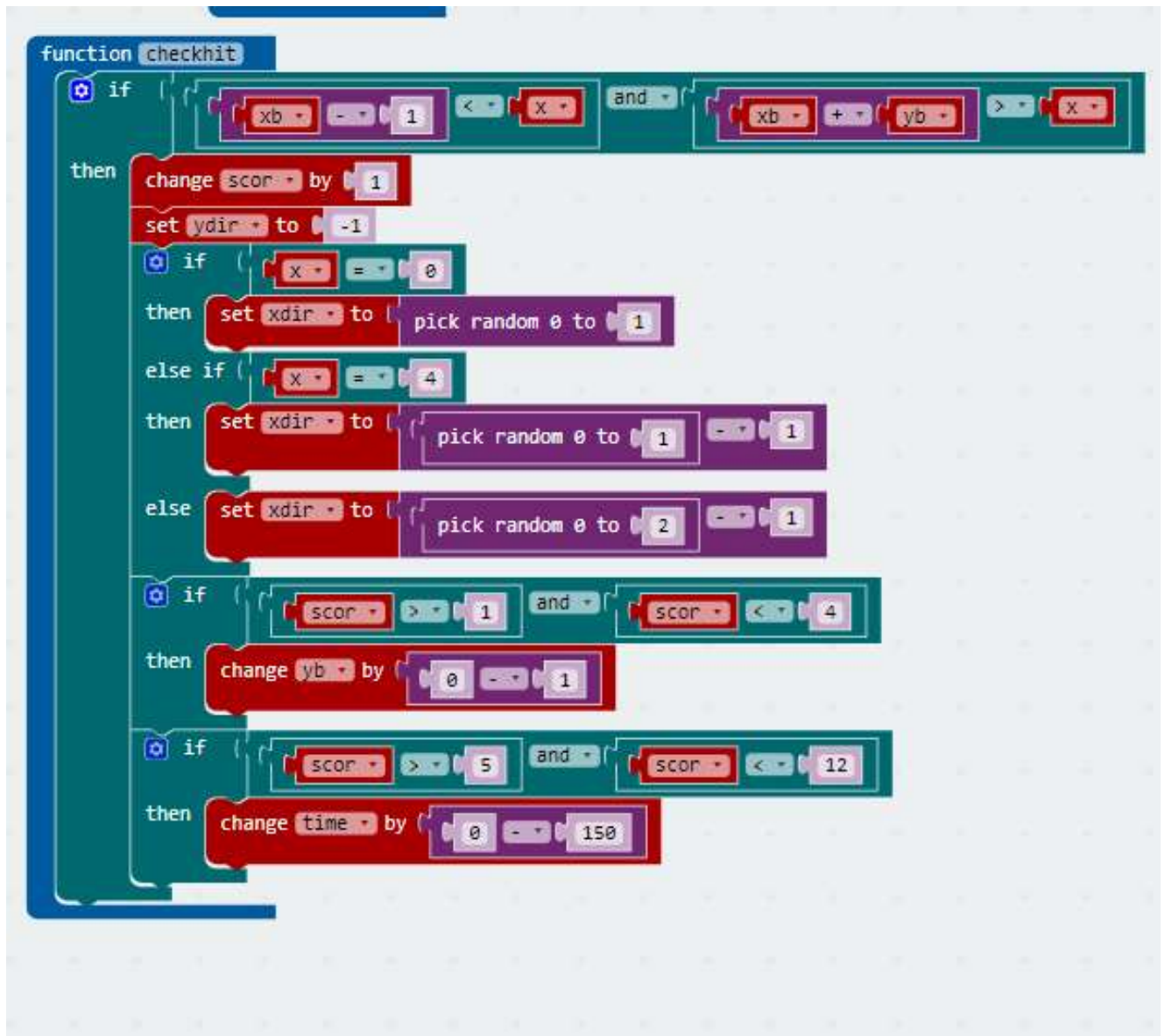
The image displays a collection of Scratch code blocks for a Ping Pong game, organized into several functional groups:

- Main Game Loop:** A `forever` loop containing a `while` loop that checks if `gam` is `true`. Inside the `while` loop, there is a `pause (ms)` block with `time` and a `call function` block for `moveball`.
- Board Function:** A function named `board` that uses a `for` loop: `for (let i = xb; i < (xb + yb); i++) { led.plot(i, 4) }`.
- Left Movement Function:** A function named `left` that changes `xb` by `-1`, clears the screen, and calls the `board` and `ball` functions.
- Right Movement Function:** A function named `right` that changes `xb` by `1`, clears the screen, and calls the `board` and `ball` functions.
- Left Side Initialization:** A function named `leftside` that sets `xdir` to `1`.
- Right Side Initialization:** A function named `rightside` that sets `xdir` to `-1`.
- Button Press Events:** Two event blocks: `on button B pressed` calling `right`, and `on button A pressed` calling `left`.
- Initialization Function:** A function named `corners` that sets `xdir` to `0` and `ydir` to `0`.

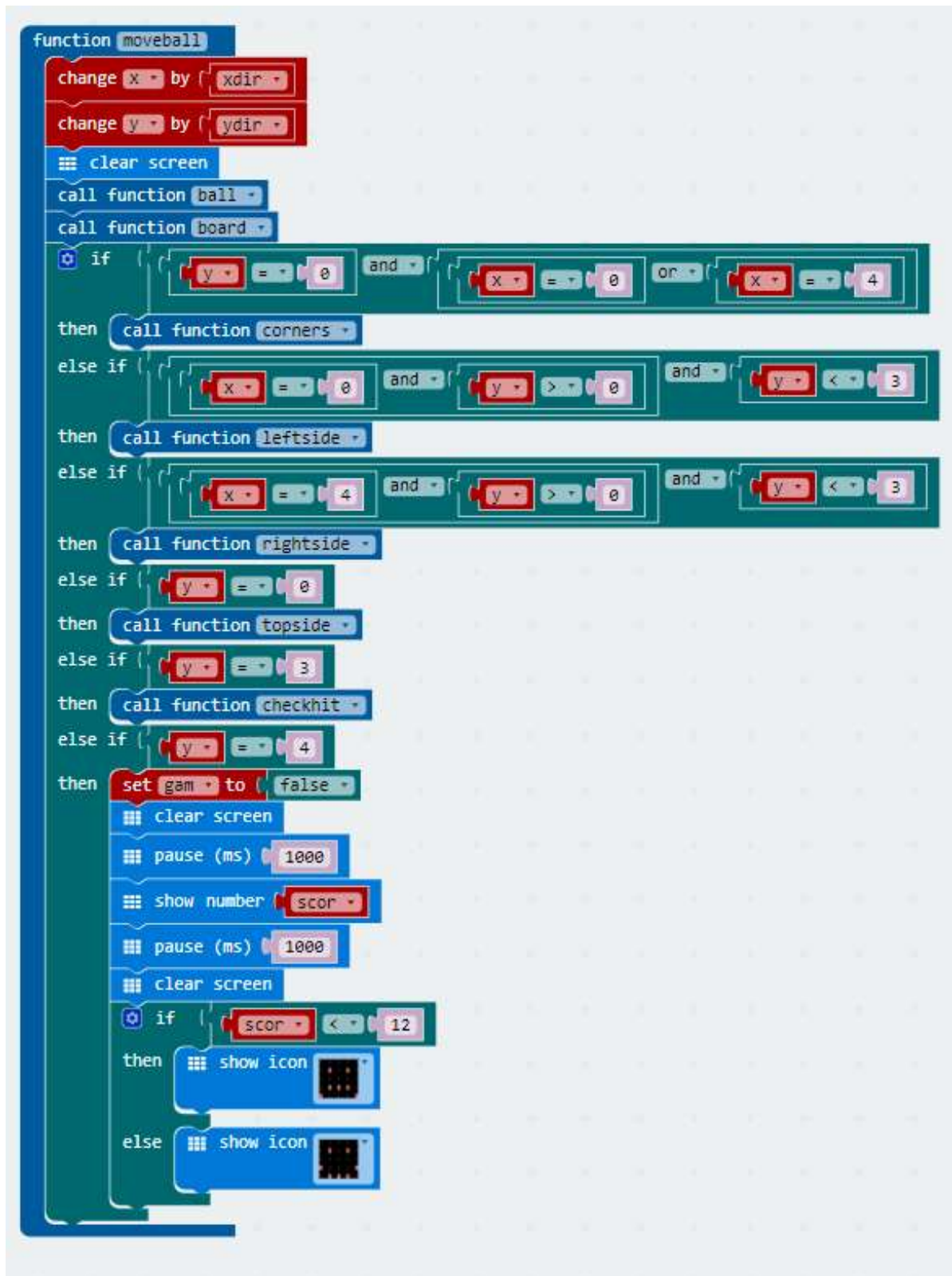
Tutorial 2.1x – Ping Pong



Tutorial 2.1x – Ping Pong



Tutorial 2.1x – Ping Pong



You are encouraged to make changes, improvise and customize the game using your own ideas.

Tutorial 2.1x – Ping Pong

6. Let's write some code – Javascript Based

If you are a fan of Javascript and keen to code the pong game in microbit then refer to the code below. The Javascript code below will deliver the same logic and in-fact the same set of blocks as we have used in the previous section to design the pong game.

So let's head over to the micro:bit block code editor page (<https://makecode.microbit.org/>) and get coding!!! Here's a view of the code in javascript form –

```
#####  
# Code Starts Here  
#####  
function rightside() {  
    xdir = -1  
}  
function left() {  
    xb += 0 - 1  
    basic.clearScreen()  
    board()  
    ball()  
}  
input.onButtonPressed(Button.B, () => {  
    right()  
})  
function topside() {  
    ydir = 1  
    if (x == 0) {  
        xdir = Math.random(2)  
    } else if (x == 4) {  
        xdir = Math.random(2) - 1  
    } else {  
        xdir = Math.random(3) - 1  
    }  
}  
function board() {  
    for (let i = xb; i < (xb + yb); i++) {  
        led.plot(i, 4)  
    }  
}  
function right() {  
    xb += 1
```



Tutorial 2.1x – Ping Pong

```
    basic.clearScreen()
    board()
    ball()
}
function ball() {
    led.plot(x, y)
}
function leftside() {
    xdir = 1
}
function corners() {
    xdir = 0 - xdir
    ydir = 0 - ydir
}
input.onButtonPressed(Button.A, () => {
    left()
})
function checkhit() {
    if (xb - 1 < x && xb + yb > x) {
        scor += 1
        ydir = -1
        if (x == 0) {
            xdir = Math.random(2)
        } else if (x == 4) {
            xdir = Math.random(2) - 1
        } else {
            xdir = Math.random(3) - 1
        }
    }
    if (scor > 1 && scor < 4) {
        yb += 0 - 1
    }
    if (scor > 5 && scor < 12) {
        time += 0 - 150
    }
}
}
function moveball() {
    x += xdir
```



Tutorial 2.1x – Ping Pong

```
y += ydir
basic.clearScreen()
ball()
board()
if (y == 0 && (x == 0 || x == 4)) {
    corners()
} else if (x == 0 && y > 0 && y < 3) {
    leftside()
} else if (x == 4 && y > 0 && y < 3) {
    rightside()
} else if (y == 0) {
    topside()
} else if (y == 3) {
    checkhit()
} else if (y == 4) {
    gam = false
    basic.clearScreen()
    basic.pause(1000)
    basic.showNumber(scor)
    basic.pause(1000)
    basic.clearScreen()
    if (scor < 12) {
        basic.showIcon(IconNames.Sad)
    } else {
        basic.showIcon(IconNames.Happy)
    }
}
}

let x: number
x = 0
let y: number
y = 0
let xb: number
xb = 0
let yb: number
yb = 0
let xdir: number
xdir = 0
```



Tutorial 2.1x – Ping Pong

```
let ydir: number
ydir = 0
let scor: number
scor = 0
let gam: boolean
gam = true
let time: number
time = 1000
xb = 0
yb = 4
board()
x = 2
y = 0
ball()
xdir = 0
ydir = 1
basic.forever(() => {
    while (gam == true) {
        basic.pause(time)
        moveball()
    }
})

#####
# Code Ends Here
#####
```



Tutorial 2.1x – Ping Pong

7. Downloadable Copy Of The Project

If you are keen to try out the code on your micro:bit and see the game in action before you start writing your code then go on and use the attachment provided below.

Start off by coping the attachment, saving it to your local hard disk and then just drag the file (hex) onto the micro:bit drive (which should show up on your local machine just like a new storage drive).

Please ensure that you have connected the micro:bit to your machine and identified the drive that the micro:bit shows up as using windows explorer. To re-start the game just hit the reset key on the rear of the micro:bit and your game will re-start.



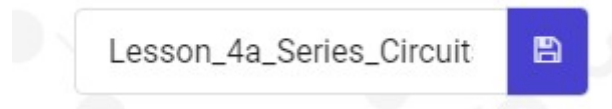
microbit-pong-lite.hex



Tutorial 2.1x – Ping Pong

8. Downloading Your Code To The micro:bit

Once you have completed the program, enter a name for your program using the option provided below i.e. in the text box adjacent to the small save button.



Now that you have given your program a name and saved it you can download it your micro:bit. But before we do that let's confirm what drive your micro:bit shows up as. On most machines the micro:bit will show up as an additional USB drive. So head over into windows explorer and confirm what drive name (D:, E:, F:, G:, etc.) the micro:bit shows up as. You need to absolutely be sure what drive the micro:bit shows up as. Once you've confirmed what drive the micro:bit shows up as on your machine you can select the right drive when downloading the code to the micro:bit. If in doubt please ask the volunteer/mentor/session facilitator helping out.



To download the newly written code to the micro:bit, hit the download button shown above. You should now see a dialog box open up and you will be asked to save the file somewhere on your machine. Please choose the drive your micro:bit shows up as i.e. D: or E: or F: or whatever it shows up as on your machine. A sign of success is when you see the lights on the rear (orange) lighting up in quick succession suggesting that the code is being written to the micro:bit. On completion the micro:bit reboots and you should now see the code in action on the micro:bit.

If hitting the download button shown above does not open up a dialog box asking you to save to the micro:bit please save the file (you will have a <filename.hex> file) to your desktop. Then open up windows explorer and drag that file onto the drive which is your micro:bit. A sign of success is when you see the lights on the rear (orange) lighting up in quick succession suggesting that the code is being written to the micro:bit. On completion the micro:bit reboots and you should now see the code in action on the micro:bit.

Please feel free to customize the code blocks, have a play. Add your own custom code and re-download the code to the micro:bit. Give yourself a tap on the back, you've just completed your first circuit!!!!

Tutorial 2.1x – Ping Pong

9. Challenges

Well done for completing the tutorial. There's a lot of ground we have covered in this tutorial so please feel free to make notes, come back to the tutorial at some point down the line and ask your learning facilitator any questions or doubts you might have on the concepts covered this far.

Let us now stretch it a bit further -

1. Implement code such that the speed of the ball increases as the points increase i.e. gradually increase speed every 10 points, etc.
2. Implement code such that the user can tilt the board and get the paddle to move
3. Are you able to think of new ideas or approaches to changing the pong game? We would be keen for you to share those ideas of your with rest of the class and show us a working demo using a micro:bit.

