



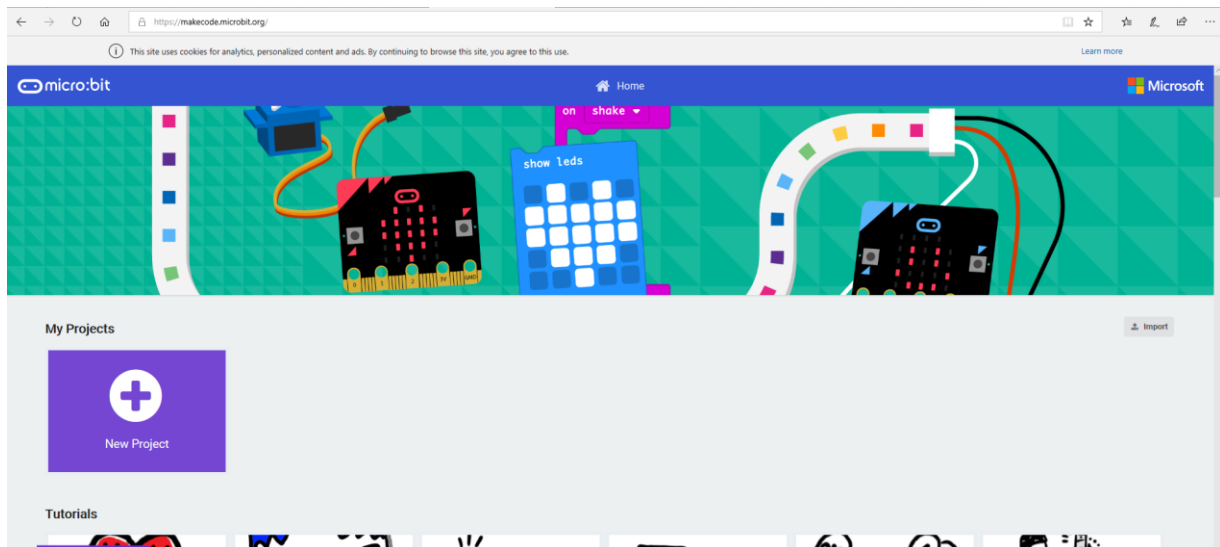
MERCHANT VENTURERS SCHOOL OF ENGINEERING OUTREACH PROGRAMME

MICRO:BIT SNAKE GAME

BY: PAPAKONSTANTOPOULOU ANNA

Setting up the editor environment:

1. Open Google Chrome.
2. Go to: makecode.microbit.org
3. Click the “New Project” button



Connecting the Micro:bit:

1. Plug the Micro:bit into the computer using the USB cable.
2. Ask the demonstrator to give you the .hex file you will need to start your game.

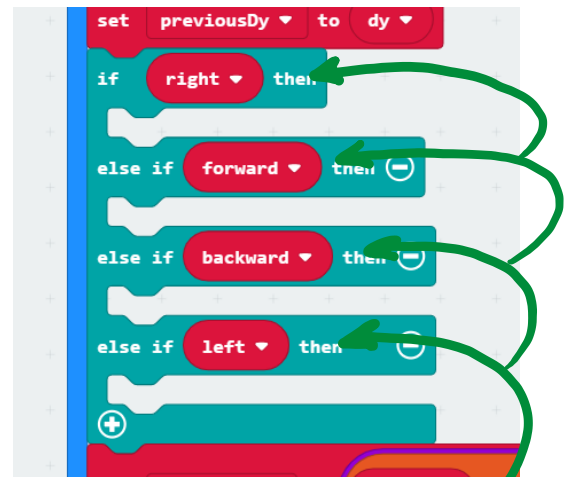
For the completed game to work, all tasks should be completed.

From the code you are given there are 4 sections that seem incomplete – the aim of this workshop is to complete these sections while understanding the principles of direction and snake motion in the game.

1. Defining the direction of movement – coordinates

dx and dy are two variables that define the x and y direction of the snake. dx is along the horizontal axis and dy along the vertical axis.

Task 1: Find the section of the code as pictured and try to set dx and dy to the appropriate values for each case, so that the direction of the snake changes according to the tilting motion of the board.



REMEMBER!!

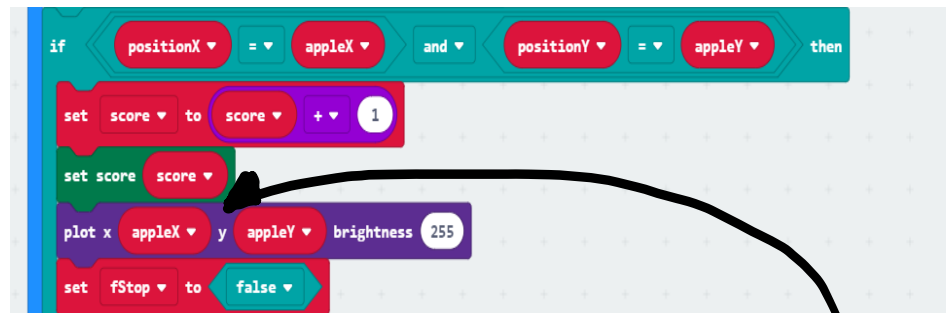
Setting the direction does not mean setting the position – hence you only need to define it by using '1's and '0's.

In these spaces, write your code by using the "set__ to__" block, which can be found under the "Variables" tab.

Variable is a thing (for example position or direction or coordinate) that can be imagined as an empty box, waiting for us to set a value to it (for example 5). Inside loops, we can change the values of variables multiple times by iteration. The most used way that we are changing the value of variables in the snake game, is by tilting the board or when the snake eats the apple!

2. Creating the tail of the snake - arrays

Task 2: Find the section of the code as pictured and try to extend the length of the snake tail after the game score (green block) is set to the value of the variable score (red block).



Insert the new blocks between the "set score" and the "plot" blocks.

Hint: Use the "insert at _ value _" block found under the "Arrays" tab.

Also, the snake's X and Y coordinates have been previously defined as arrays and hence to extend the tail, it means to extend the length of these arrays.



* It is good to use the "length of array _" function which can be also found under the "Arrays" tab.

We use that as we want to extend the length of the array snake by the coordinates of the apple.

Arrays

We can imagine the snake being made by boxes – starting with one "box" for the head and adding up as many apples as it eats for the tail. Every "box" has an x (horizontal) and a y (vertical) coordinate, and all the "boxes" together, are called an array. As the snake eats the apple, a new "box" is added, and hence the length of the existing array is being extended both vertically and horizontally (as each "box" has both coordinates).

3. Where's the apple? – Making LEDs light up

Task 3: Find the section of the code as pictured and set the “apple” to appear in random places within the LED matrix.



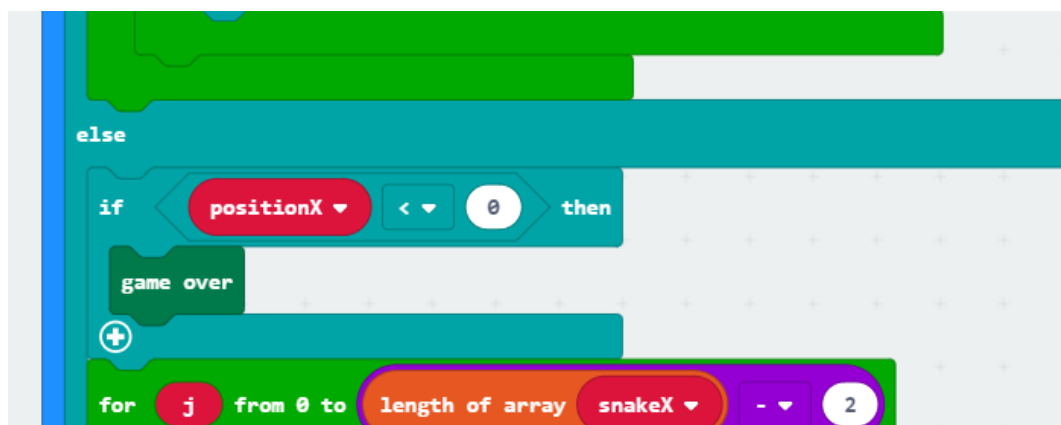
Hint: The apple has two coordinates which in the code are defined by the variables appleX and appleY.

Insert the block of code here.

! Since we want the apple to appear in random positions within the LED matrix ~ check the “Math” tab. !

4. Game Over? - Conditional statements

Task 4: Find the section of the code as pictured below. The condition for the game to be over when the snake touches the wall is incomplete. Find the full condition and replace the one that is already there.



Hint: Think about the dimensions of the LED matrix and hence the coordinate boundaries for both “positionX” and “positionY” variables.

* Try to find all 4 possible inequalities that are the conditions of the snake hitting the “walls” of the LED matrix.

(two for the X component and two for the Y component of the snake array)