

# microbit / Meteors!

Intermediate level microbit - 1hr

Original game & code by Emily & Claire!

In Meteors! we create a game that you control using the A & B buttons moving your spaceship to avoid meteors coming towards you.

## You will need..

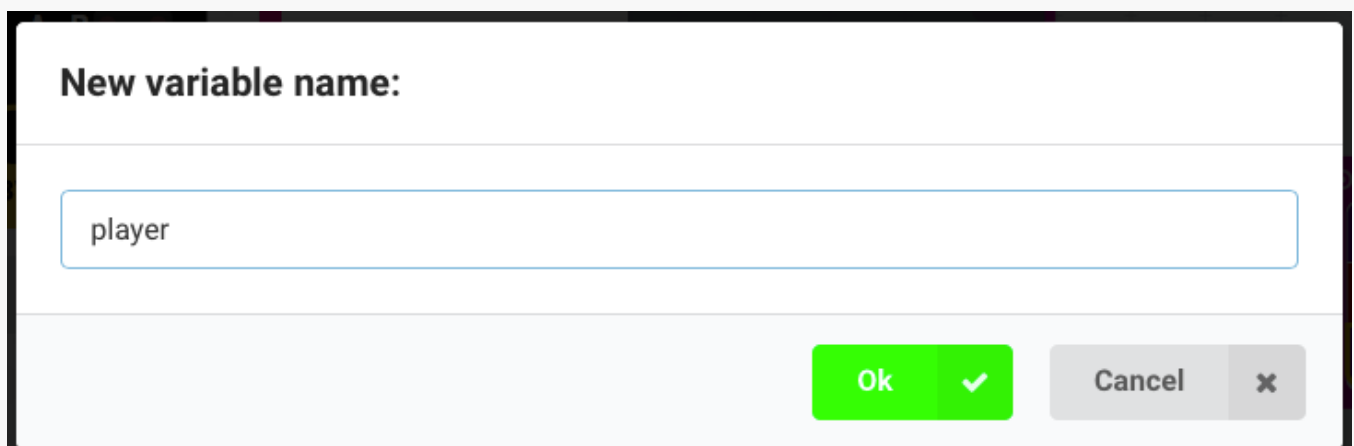
- A BBC microbit

## Step Zero: Open up the block editor

[Open up the Microbit PXT editor](#) and then plug into your microbit into your computer.

## Step One: Controlling the player LED

We will use the A & B buttons to move left and right. To keep track of where the user is, we need to use a variable: make a new one called `player`.



**New variable name:**

Ok ✓ Cancel ✕

Now, use this code to move the player LED left and right.

```

on button A pressed
  unplot x: player y: 4
  change player by -1
  plot x: player y: 4

on button B pressed
  unplot x: player y: 4
  change player by 1
  plot x: player y: 4

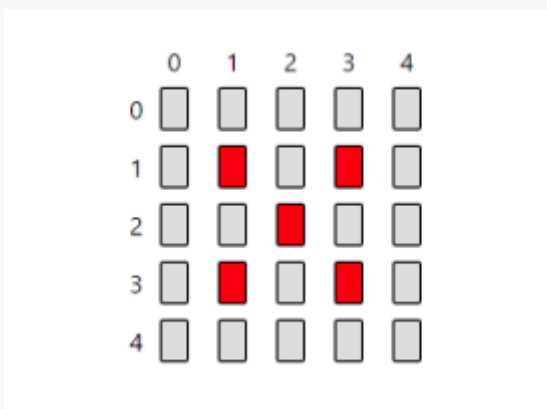
```

Upload the code to the microbit. What happens when you click A when the LED is all the way to the left?

## Step Two: Fix the player controls

The LED for the player disappear off left and right when you go too far!

The LEDs on the microbit have values that they correspond to, you can see them here...



You can the player dropping off the edges by stopping the `player` variable from getting too big (more than 4 going right) or too small (less than 0 going left).

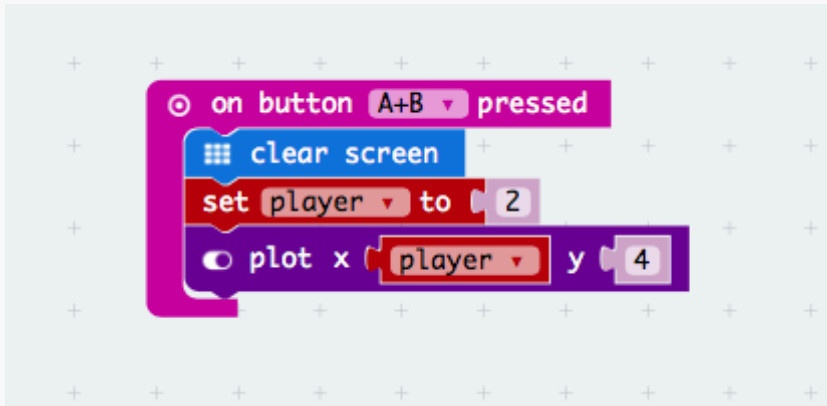
```

on button A pressed
  unplot x: player y: 4
  change player by -1
  if (player < 0)
    then set player to 0
  plot x: player y: 4

on button B pressed
  unplot x: player y: 4
  change player by 1
  if (player > 4)
    then set player to 4
  plot x: player y: 4

```

We also want to start with the player in the middle, so we can do this by pressing both A+B buttons at the same time which will reset and start the game.



```
on button A+B pressed
  clear screen
  set player to 2
  plot x player y 4
```

Upload the code to the microbit. Test that the player LED doesn't fall off the edges!

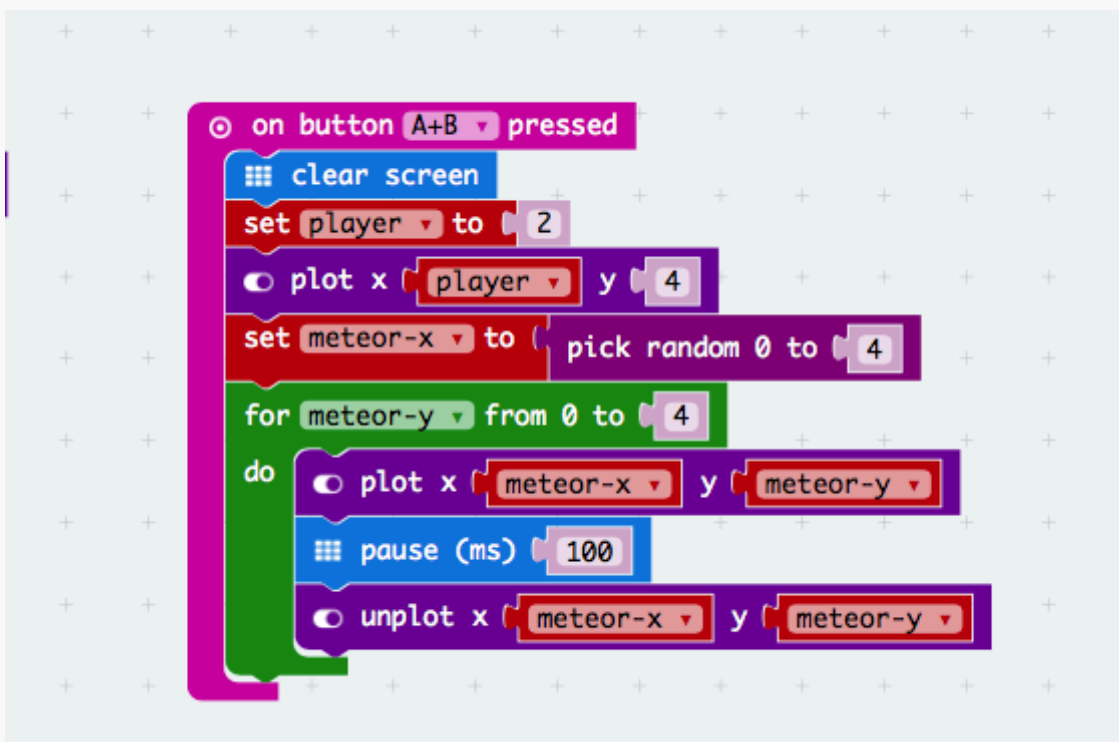
## Step Three: Falling meteors

Now we've got the player working, we can move onto the meteors.

These start at a random position on the top row of the grid and gradually fall down. So we need two variables to control their position, `meteor-x` and `meteor-y`.

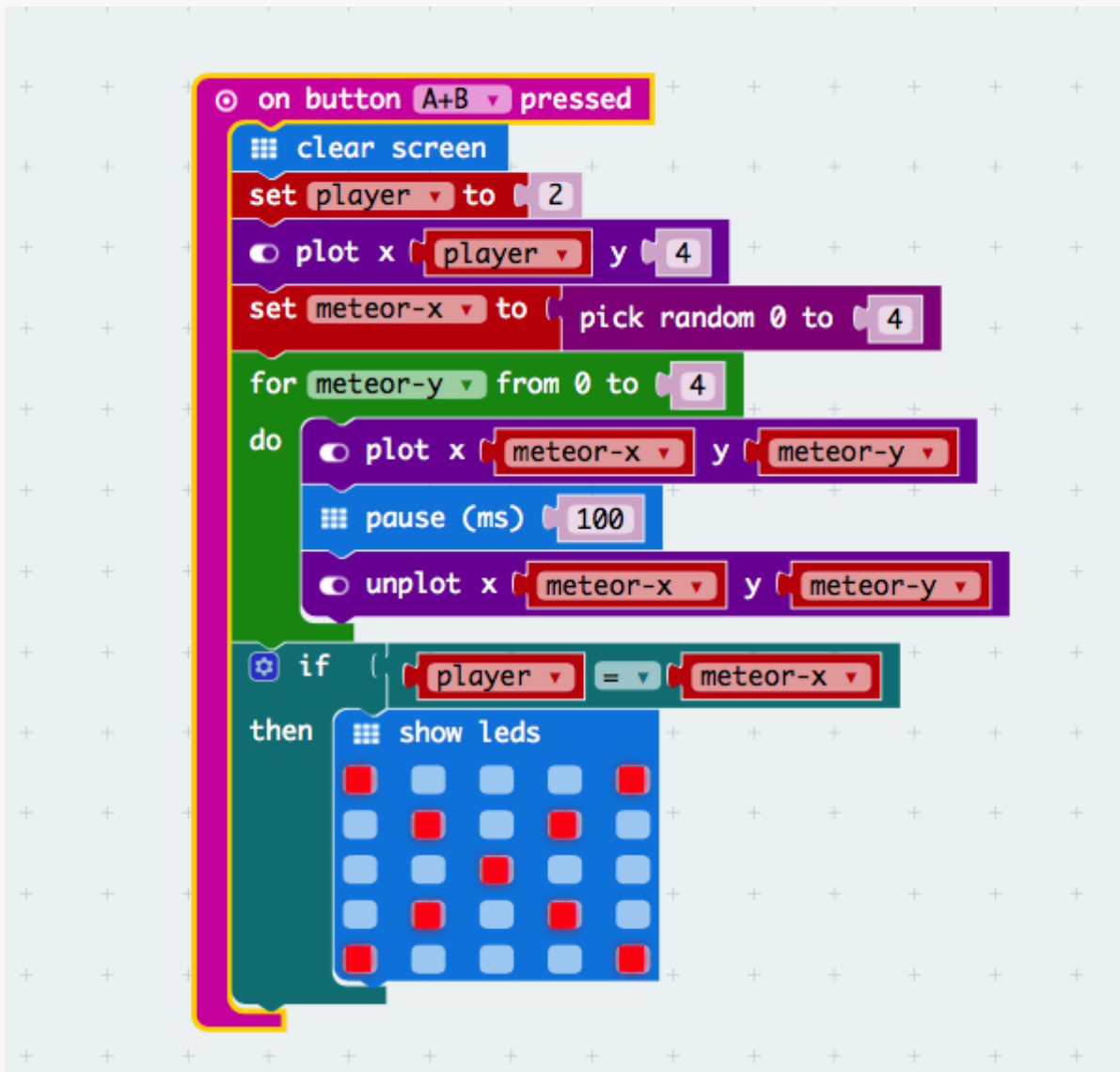
`meteor-x` is set to a random value between 0 and 4, and `meteor-y` is controlled by a `for` loop which sets `meteor-y` from 0 to 4 in stages.

Add this code to the current A+B button press event...



```
on button A+B pressed
  clear screen
  set player to 2
  plot x player y 4
  set meteor-x to (pick random 0 to 4)
  for meteor-y from 0 to 4
  do
    plot x meteor-x y meteor-y
    pause (ms) 100
    unplot x meteor-x y meteor-y
```

When the meteor reaches the bottom row, we need to see if the meteor has hit the player. We do this by comparing the `meteor-x` and `player` variables. If they are the same, then the meteor has hit the player :(




```
on button A+B pressed
  clear screen
  set player to 2
  plot x player y 4
  set meteor-x to pick random 0 to 4
  for meteor-y from 0 to 4
    do
      plot x meteor-x y meteor-y
      pause (ms) 100
      unplot x meteor-x y meteor-y
  if (player = meteor-x) then
    show leds
```

The code block shows a sequence of actions triggered by a button press. It starts with clearing the screen and setting the player's x-coordinate to 2. A meteor's x-coordinate is chosen randomly between 0 and 4. A loop then moves the meteor from the top row (y=0) to the bottom row (y=4), plotting and unplotting its position at each step with a 100ms delay. Finally, an if-statement checks if the meteor's x-coordinate matches the player's x-coordinate. If they match, the 'show leds' block is executed, displaying a 5x5 grid of LEDs with several lit up.

Almost done! We now want to make sure that we keep playing the game until a meteor hits the player. We can do this using a `alive` variable and a `while` loop

```

on button A+B pressed
  set alive to true
  clear screen
  set player to 2
  plot x player y 4
  while alive
    do
      set meteor-x to pick random 0 to 4
      for meteor-y from 0 to 4
        do
          plot x meteor-x y meteor-y
          pause (ms) 100
          unplot x meteor-x y meteor-y
      if player = meteor-x
        then
          set alive to false
          show leds
  
```



Finally... change the player controls so that the player can't move if the game is over:

```

on button A pressed
  if alive
    then
      unplot x player y 4
      change player by -1
      if player < 0
        then set player to 0
      plot x player y 4

on button B pressed
  if alive
    then
      unplot x player y 4
      change player by 1
      if player > 4
        then set player to 4
      plot x player y 4
  
```

Upload the code to the microbit. Play the game!

Challenge I: Keep a score (beginners, 5 mins)

- Add a `score` variable and add 1 to it every time a meteor is avoided.
- Show this variable instead of the giant X at the end of the game.

## Challenge II: Make the game harder as your score increases (intermediate, 15 mins)

- Now you've got a `score` variable, use it to change how long the delay is in the meteor dropping code.
- The meteor shower starts slow and gets quicker as the `score` variable increases.

## Full code

[Full code including the challenges is seen here](#)

---

This microbit worksheet 'Meteors!' from CoderDojo York is licensed under a



[Creative Commons Attribution-NonCommercial 4.0 International License](#).

No commercial use of our material is permitted