

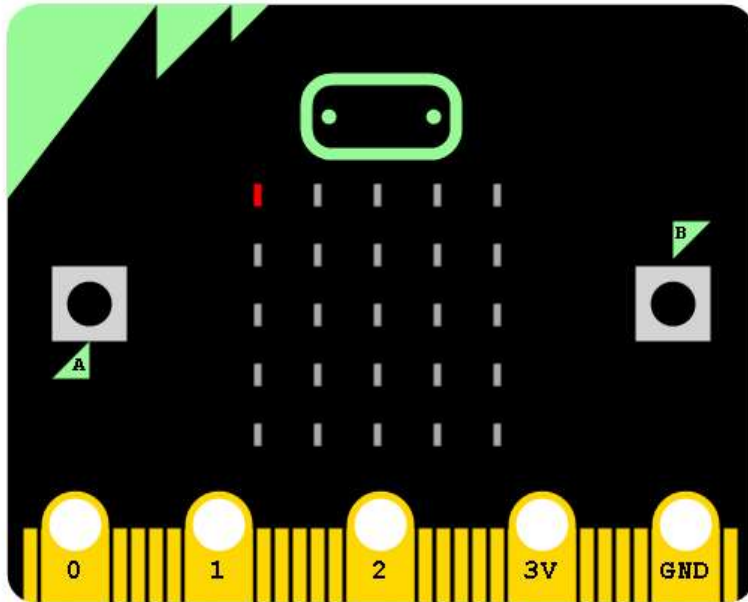
# MultiWingSpan

[Home](#) [Programming](#) [Web Design](#) [Computer Science](#) [Twisting Puzzles](#) [Arduino](#) [BBC micro:bit](#)

## BBC micro:bit Concentration Game

### Introduction

This example is a simple concentration/reaction game. You have to press the buttons when all of the lights are on. You get a short time in which to do this. If you manage it, the speed of the game increases. When you fail, and you will, you are told your score.



```
from microbit import *

score = 0
start = 100
paws = start
while True:
    x = 0
    y = 0
    while y < 5:
        display.set_pixel(x,y,9)
        if button_a.was_pressed():
            sleep(1000)
            break
        x = x + 1
        if x == 5:
            x = 0
            y = y + 1
            # a little bit of help
            if x == 4 and y == 4:
                sleep(paws*2)
            else:
                sleep(paws)
    if x == 4 and y == 4:
        # hit
        paws = paws - 4
        display.show(Image.YES)
        score = score + 1
        sleep(1000)
    else:
        # miss
        display.show(Image.NO)
        sleep(1000)
        display.show(str(score))
        paws = start
        score = 0
        sleep(1000)
        a = button_a.was_pressed()
    display.clear()
    sleep(500)
```

### Challenges

1. The speed of the drawing is something that you can experiment a lot with. Notice the way that the delay is doubled for the last pixel. You might do this for different pixels too to make the game easier. Another effect you might explore is to have the speed increase as the animation progresses.
2. The game could do with some sort of title/menu base so that things are not so rushed at the start. You might also flash some of the images that inform the player what is happening.

### BBC Microbit



+ <b>Block Editor - The Basics</b>
+ <b>Block Editor - Components</b>
+ <b>Kodu - micro:bit Worlds</b>
+ <b>JavaScript Blocks</b>
+ <b>JavaScript Blocks - Exercises</b>
+ <b>Blocks - Bit:Bot</b>
+ <b>Blocks - Bit:Commander</b>
+ <b>MicroPython - Starting Off</b>
- <b>MicroPython - Examples</b>
✱ <b>Dice Rolling</b>
✱ <b>Shut The Matrix</b>
✱ <b>Encoding Morse Code</b>
✱ <b>Encoding Ciphers</b>
✱ <b>Drawing A Maze</b>
✱ <b>Scrolling Race Track</b>
✱ <b>Vertical Scroller</b>
✱ <b>Concentration Game</b>
✱ <b>Text Entry - Accelerometer</b>
✱ <b>Charlie's Python Game</b>
✱ <b>Lights Out Game</b>
✱ <b>Sam's French Number Game</b>
✱ <b>Bryn's Concentrated Clocks</b>
✱ <b>Lattice Paths</b>
✱ <b>Knight Moves</b>
+ <b>MicroPython - Components</b>
+ <b>MicroPython - Breakout Boards</b>
+ <b>MicroPython - Exercises</b>
+ <b>MicroPython - Pi Accessories</b>
+ <b>MicroPython - Bit:Bot</b>
+ <b>MicroPython - Bit:Commander</b>
+ <b>MicroPython - Projects</b>
+ <b>MicroPython - Visual Basic</b>
+ <b>Other - Odds &amp; Ends</b>



3. A buzzer and some nice noises would liven up the process.
4. Adapt the game so that the animation becomes the timer for some other action that needs to be performed, like a sequence of accelerometer gestures.