BBC micro:bit
Bit:Commander - Evasion Game

Introduction

This is a simple game using the joystick and buzzer of the Bit:Commander. The player controls the position of the bright dot by moving the joystick. The dot is moved according to the absolute position of the joystick. The dimmer dot is the spaceship that chases the player, moving towards them every now and then. The player can move more quickly than the spaceship. A beep is sounded each time the ship moves. If the ship catches the player, the game is over. The ship moves a teeny bit quicker each time.

Programming

The player position is updated more frequently than the ship. This is achieved by using running_time() instead of sleep to control the timing of the game.

The ship movement is calculated each time based on the player position. The player's dot is only redrawn if the position changes. This stops the display flickering that you can get if you update the screen too often.

```
from microbit import *
import music

def joy2grid():
    x = pin1.read_analog()
    y = 1023 - pin2.read_analog()
    x = int((x / 1023) * 4 + 0.5)
    y = int((y / 1023) * 4 + 0.5)
    return x,y

def shipmove(a,b):
    if a>b: return -1
    if a<b: return 1
    return 0

def play_game():
    px = 2
    py = 2
    lastx = 2
    lasty = 2
    shipx = 0
    shipy = 0
    t = running_time()
    shiptime = 1000
    display.set_pixel(px,py,9)
    display.set_pixel(shipx,shipy,5)
    playing = True
    while playing:
        px,py = joy2grid()
        # move player
        if lastx!=px or lasty!=py:
            display.set_pixel(lastx,lasty,0)
```

BBC Microbit

Collapse All Expand All

+ Block Editor - The Basics
+ Block Editor - Components
+ Kodu - micro:bit Worlds
+ JavaScript Blocks
+ JavaScript Blocks - Exercises
+ Blocks - Bit:Bot
+ Blocks - Bit:Commander
+ MicroPython - Starting Off
+ MicroPython - Examples
+ MicroPython - Components
+ MicroPython - Breakout Boards
+ MicroPython - Exercises
+ MicroPython - Pi Accessories
+ MicroPython - Bit:Bot
- MicroPython - Bit:Commander
  - Bit:Commander
  - The Joystick
  - The Neopixels
  - The Potentiometer
  - The Pushbuttons
  - The Buzzer
  - Evasion Game
  - Light's Out Game
  - Simon Game
  - Bit:Bot/Robot Controller
  - Text Entry
  - Unicorn Commander
+ MicroPython - Projects
+ MicroPython - Visual Basic
+ Other - Odds & Ends

```
            display.set_pixel(px,py,9)
        if running_time()-t>shiptime:
            t = running_time()
            shiptime -= 1
            display.set_pixel(shipx,shipy,0)
            music.pitch(440,100,wait=False)
            shipx += shipmove(shipx,px)
            shipy += shipmove(shipy,py)
            display.set_pixel(shipx,shipy,5)
        if px == shipx and py == shipy:
            playing = False
            music.play(music.WAWAWAWAA)
            display.show(str((1000-shiptime)))
        # set followers
        lastx=px
        lasty=py
        sleep(5)

play_game()
```

The game is quite playable. The small changes in game speed (shiptime - larger=slower) could be adjusted to make the game more interesing. After a period of time, the game could start speeding up by a larger factor or you could have the speed reset and add a ship.