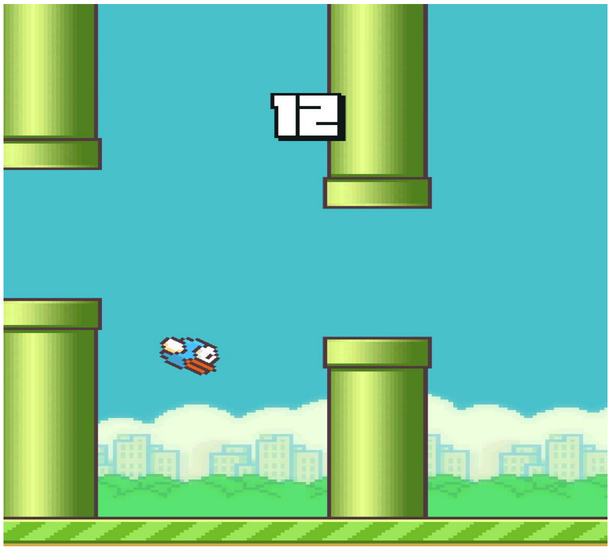
Flappy Bird Micro:Bit Project

Flappy Bird is a side-scrolling mobile game featuring 2D retro style graphics. The objective was to direct a flying bird that moves continuously to the right, between sets of Mario-like pipes. If the player touches the pipes, they lose.



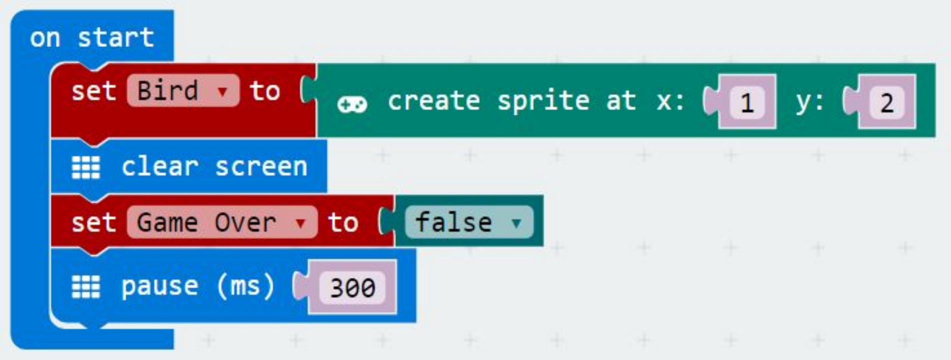
Our version of the game will be a little more simplistic (due to the fact that we have only 25 LEDs). Instead of pipes, we will simply have single sprites moving across the LED's that our bird must avoid.

Variables Needed in Project

- **Bird** - sprite variable that will act as the bird in the game.
- **Obstacle** - sprite variable that will act as the obstacle in the game.
- **Score** - stores the score that the player has obtained.
- **Start Timer** - stores the time that the bird started touching the edge
- **End Timer** - stores the time that the bird finished touching the edge
- **Index** - stores the index of a for loop block.
- **Game Over** - stores a boolean used to end the game.

Create Bird Sprite

First you will need to set up your "*Bird*" sprite by spawning it in a particular location when the program is started.



```
on start
    set Bird ▾ to  create sprite at x: 1 y: 2
    clear screen
    set Game Over ▾ to  false ▾
    pause (ms) 300
```

<u>Tasks</u>

1. Copy the code blocks above.

Obstacle Movement

The next step is to move the “*Obstacle*” sprite across the screen and test whether or not it has collided with the “*Bird*”.

```
forever:
    while “Game Over” is equal to false:
        pause for 2000 ms
        create “Obstacle” sprite at x: 4 , y: random number between 0 and 4
        pause for 300 ms

        for “Index” from 0 to 4:
            change “x” of “Obstacle” by “-1”
            pause for 300 ms

            if “Bird” is touching “Obstacle”, then:
                set “Game Over” equal to true
                show “Score”
                pause for 1000 ms
                set game over
        change “Score” by 1
        delete “Obstacle”
```

Tasks

1. Convert the pseudocode to working code blocks.

Bird Collision with Edge

The next step will be to test whether or not the “*Bird*” is touching the top or bottom edge and end the game if it is doing so for more than a second.

```
forever:
    while “Game Over” is equal to false:
        pause for 400 ms

        if “Bird” is touching edge, then:
            set “Start Timer” equal to running time (ms)
            while “Bird” is touching edge:
                set “End Timer” equal to running time (ms)

                if “End Timer” - “Start Timer” >= 1000 ms, then:
                    set “Game Over” equal to true
                    show “score
                    pause for 1000 ms
                    set game over
```

Tasks

1. Convert the pseudocode into working code blocks.

Bird Movement

The final step is to assemble the code blocks that deal with the movement of the “*Bird*” sprite. It is important to know that the “*Bird*” should always be moving down towards the bottom of the screen, unless the A button is pressed, in which case it should move upwards..

Tasks

1. Write code to make the “Bird” go **up** when the **A button** is pressed.
2. Write code to make the “Bird” go **down** when the **A button** is **not** pressed.
3. Download the code.

Hint: If the “*Bird*” is moving up and down the screen too fast after the A button is pressed, add in a pause of about 100 ms to slow it down after you change it’s “*y*” position.