

# Crashy Bird

(Nivå: Svår)


## Introduktion

Gör en microBit-version av det populära spelet Flappy Birds!!

## Du behöver detta:

- En MicroBit
- Batteripack eller USB-kabel

## Programmet

Gå till [makecode.microbit.org](https://makecode.microbit.org) för att programmera, eller tryck  på iPad.

## Step 1: Add the Bird to the Game

First, we are going to add a sprite for the bird from the Game menu and make it blink.



## Step 2: Make the Bird fly

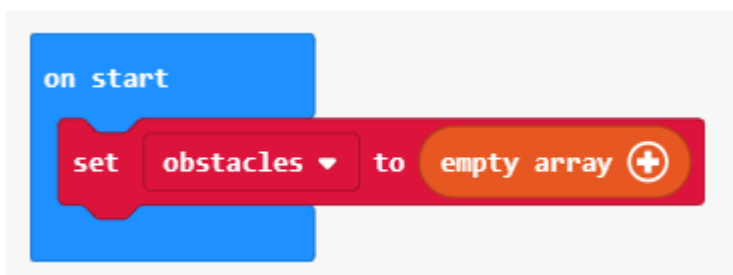
Before creating the code for the game actions, let's first add some controls so that we can move around. We'll control the bird by pressing the A button to go up or the B button to go down.



### Step 3: Generating obstacles

This is where things will start to get interesting. We're going to randomly generate obstacles. We'll keep all obstacles inside the array. All obstacles will have a single hole for the bird to fly through.

First, create an array of `obstacles` which will hold all of the obstacle sprites.



Now generate vertical obstacles consisting of 4 sprites and 1 random hole. Create new variable called `emptyObstacleY`. Using `pick random`, generate a random number from 0 to 4 and store it inside `emptyObstacleY`.

Using `for` loop, iterate from 0 to 4. For every coordinate not equal to `emptyObstacleY` create and add obstacle sprites to the end of the `obstacles` array.

**(Se nästa sida för bild!)**

```
on start
  set emptyObstacleY to pick random 0 to 4
  for index from 0 to 4
  do
    if index ≠ emptyObstacleY then
      obstacles add value create sprite at x: 4 y: index to end
```

Now with every micro:bit restart you should see different autogenerated vertical obstacles.

Before continuing, make sure that obstacles are generated randomly and that the bird is moving up and down.

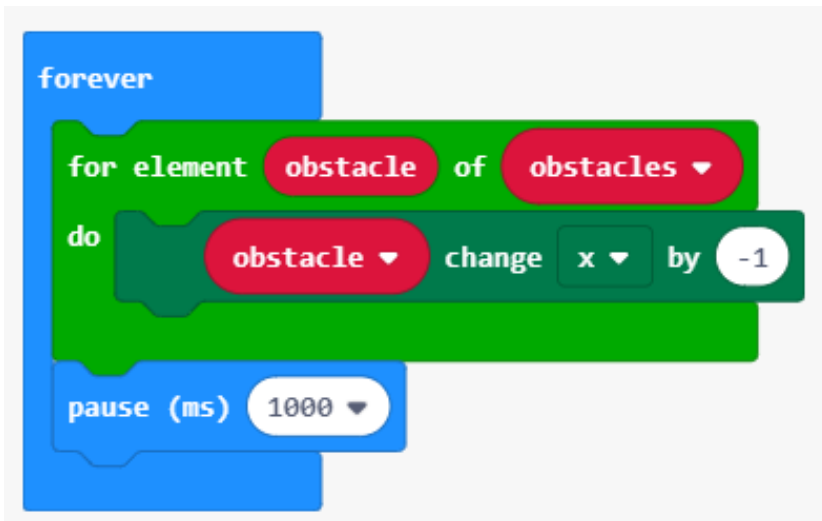
```
on button A pressed
  bird change y by -1

on button B pressed
  bird change y by 1

on start
  set bird to create sprite at x: 0 y: 2
  bird set blink to 300
  set emptyObstacleY to pick random 0 to 4
  for index from 0 to 4
  do
    if index ≠ emptyObstacleY then
      obstacles add value create sprite at x: 4 y: index to end
```

## Step 4: Make obstacles move

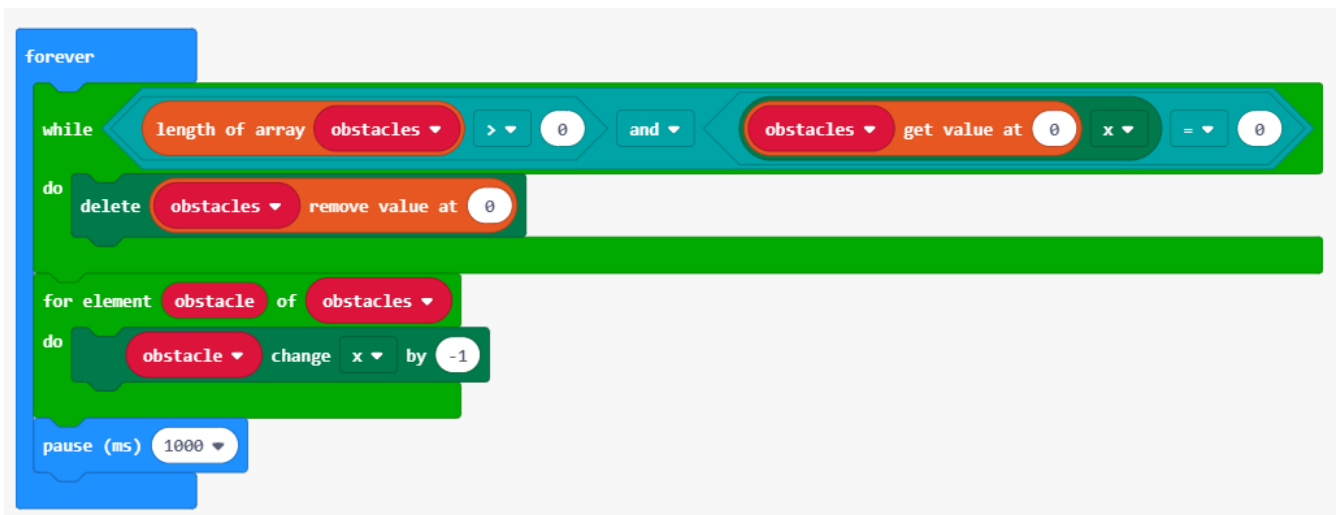
Access each obstacle using a **for element** loop (*iterate* over the `obstacles` array) and decrease the `obstacle` `x` coordinate by 1. Right click on the **value** block and rename it to **obstacle**; then drag that **obstacle** block on top of **sprite** in the **change** block.



Obstacles should move towards left every second.

## Step 5: Make obstacles disappear

Make obstacles disappear after reaching leftmost corner. Iterate over all obstacles, delete the obstacle sprites where the `x` coordinate equals 0, and remove them from the `obstacles` array.



## Step 6: Generate more obstacles

At the moment, our code generates just one vertical obstacle. We need to put obstacle generation code into the `forever` loop so that it keeps generating more and more obstacles.

```
forever
  while (length of array obstacles > 0) and (obstacles get value at 0 x = 0)
  do
    delete obstacles remove value at 0
  end
  for element obstacle of obstacles
  do
    obstacle change x by -1
  end
  set emptyObstacleY to pick random 0 to 4
  for index from 0 to 4
  do
    if (index ≠ emptyObstacleY) then
      obstacles add value create sprite at x: 4 y: index to end
    end
  end
  pause (ms) 1000
```

Now our screen is full of moving obstacles. Create some spaces between generated obstacles. Let's introduce a `ticks` variable to count how many iterations the `forever` loop has done and execute obstacle creation only if `ticks` is divisible by 3.

```

forever
  while (length of array obstacles > 0) and (obstacles get value at 0 x = 0)
  do
    delete obstacles remove value at 0
  for element obstacle of obstacles
  do
    obstacle change x by -1
  if (remainder of ticks ÷ 3 = 0) then
    set emptyObstacleY to pick random 0 to 4
    for index from 0 to 4
    do
      if (index ≠ emptyObstacleY) then
        obstacles add value create sprite at x: 4 y: index to end
    change ticks by 1
  pause (ms) 1000

```

## Step 7: Game Over

Right now nothing happens when the bird is hit by obstacle. Fix this by iterating over the `obstacles` array and checking if any obstacle sprite coordinate equals the bird coordinate.

```
forever
  while (length of array obstacles > 0 and obstacles get value at 0 x = 0)
  do
    delete obstacles remove value at 0

  for element obstacle of obstacles
  do
    obstacle change x by -1

  if remainder of ticks ÷ 3 = 0 then
    set emptyObstacleY to pick random 0 to 4
    for index from 0 to 4
    do
      if index ≠ emptyObstacleY then
        obstacles add value create sprite at x: 4 y: index to end

  for element obstacle of obstacles
  do
    if obstacle x = bird x and obstacle y = bird y then
      game over

  change ticks by 1
  pause (ms) 1000
```

Komplett kod:

```

on button A pressed
  bird change y by -1

on button B pressed
  bird change y by 1

forever
  while length of array obstacles > 0 and obstacles get value at 0 x = 0
  do
    delete obstacles remove value at 0

  for element obstacle2 of obstacles
  do
    obstacle2 change x by -1

  if remainder of ticks ÷ 3 = 0 then
    set emptyObstacleY to pick random 0 to 4
    for index2 from 0 to 4
    do
      if index2 ≠ emptyObstacleY then
        obstacles add value create sprite at x: 4 y: index2 to end

  for element obstacle3 of obstacles
  do
    if obstacle3 x = bird x and obstacle3 y = bird y then
      game over

  change ticks by 1
  pause (ms) 1000

on start
  set index to 0
  set obstacles to empty array
  set bird to create sprite at x: 0 y: 2
  bird set blink to 300

```

## Utmaningar!

Försök lägga till dessa funktioner till spelet:



1. Räkna och visa poäng.
2. Gör så att spelet går snabbare och snabbare varje gång ett hinder passeras.

Spelet är nu klart! Testa i simulatorm, eller ladda ner spelet till micro:Bit via genom att trycka på

 Ladda ned

Nu kan du testa spelet med en kompis som gjort ett likadant, eller ladda över det till en annan microbit!

Om det krånglar på iPad, följ anvisningen "Parkoppla micro:Bit med iPad", som finns som separat instruktion.