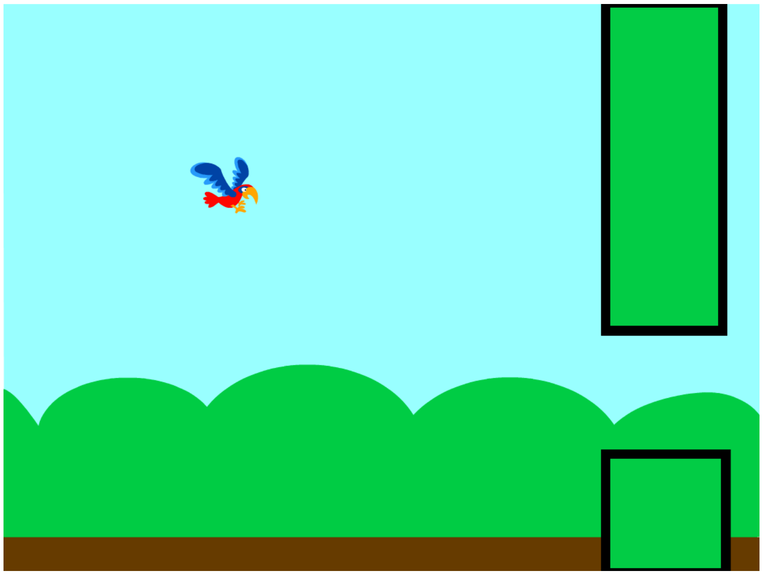
Coding Flappy Bird with Scratch

Luke Storry



Contents

1 Introduction

In this worksheet you will be guided through how to program your own customised version of the popular Flappy Bird game in Scratch. This is a really open-ended project, as there are always more improvements and enhancements you can add into your game.

1.1 You should already know:

- the basics of how to use Scratch.
- about coordinates (x & y).

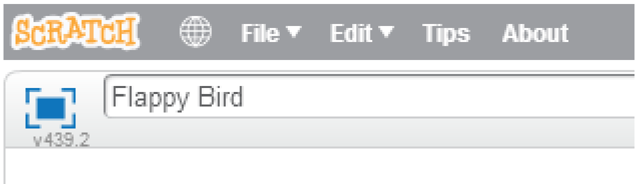
1.2 You will learn to:

- use variables
- use random numbers
- clone objects
- handle events

2 Let's Begin

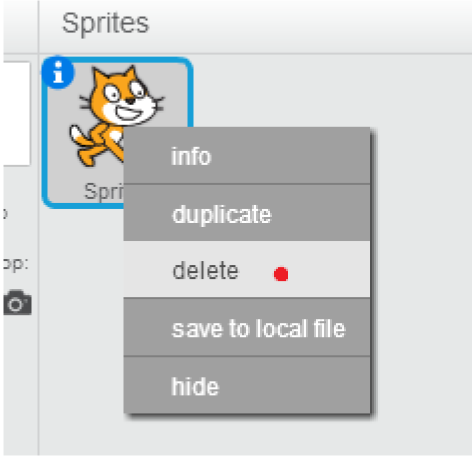
2.1 Create a new Project

Go to `https://scratch.mit.edu` and click "Create" in the top bar. This will start a new project.
Next you need to rename your project from Unititled to something more relevant, like this:

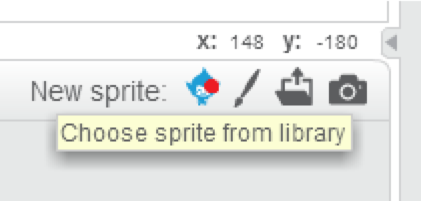


2.2 Replace the Main Sprite

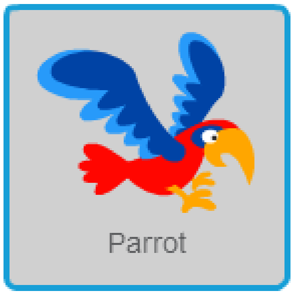
I want a flappy bird, not a cat, so let's remove the default cat sprite:



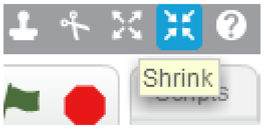
...and replace it with a different sprite.



I chose this parrot, feel free to choose another animal if you want. (or draw one if you're feeling adventurous!)

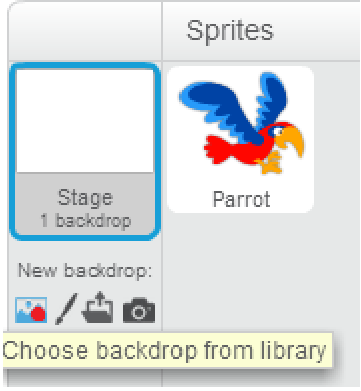


This is a bit big, so select shrink tool and click on the bird 15 times to shrink it to a better size.

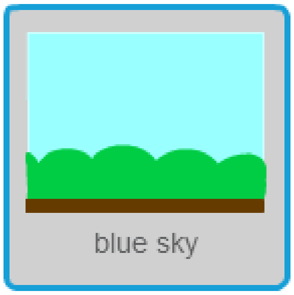


2.3 Replace the Background

Next up is to replace that boring background with something more fun.

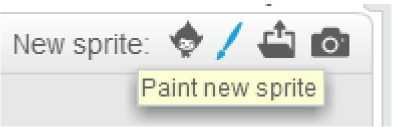


I chose the Blue Sky background as I think it fits with my parrot, but again, feel free to customise and choose your own background.

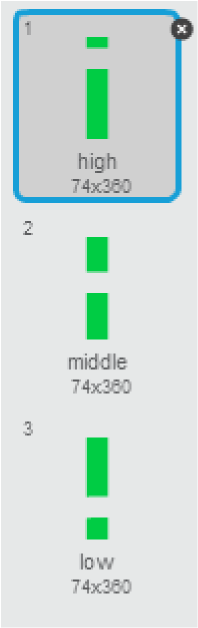


2.4 Create the Obstacles

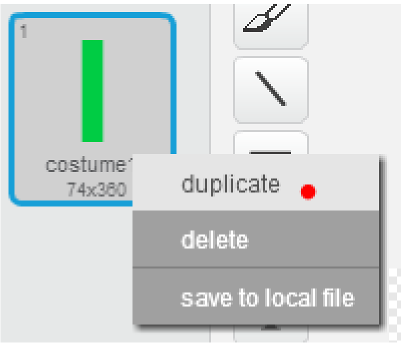
The last thing we need to do before we start scripting is create the sprite for our obstacles.



We need a few different costumes for the obsacles pipe, so we can vary between them.



Hint: if you want your obstacles to look the same, you can use the duplicate tool:



before erasing a gap.

It's your game, change the artwork to be whatever you want, from butterflies dodging flowers to spaceships avoiding planets.

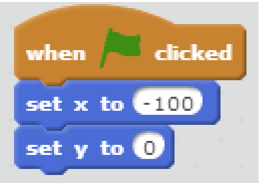
Be creative!

3 Programming the Bird

In this section we are going to think about what code we should give our ”Bird” sprite.

3.1 Initialisation

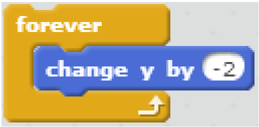
This just means how we want the bird to be set up when the game is started. Placing the bird back in the middle of the screen is a good idea:



```
when ⚑ clicked
set x to -100
set y to 0
```

3.2 Gravity

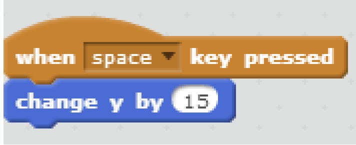
A simple way of giving the bird gravity would be to gradually reduce its y-coordinate, using this block:



```
forever
    change y by -2
```

3.3 Flight

We next need to decide on what we’re going to use as input for this game. I chose the spacebar, so my code block for flight looks like:

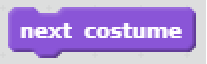


```
when space ▾ key pressed
change y by 15
```

But you can choose any button you like.

3.4 Animation

Wouldn’t it be cool if the bird’s wings could flap when its flying? This can be easily done by switching between different costumes:

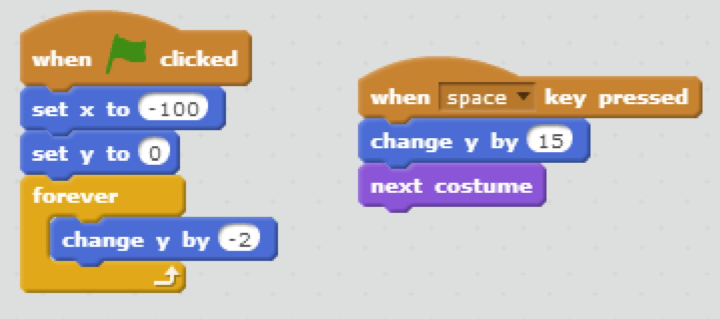


```
next costume
```

The Parrot Sprite that I’m using contains two costumes with the wings in different positions, but you can draw the different costumes if you prefer.

3.5 Finished Code for Bird

Therefore, the final code for the Bird sprite should look like this:



```
when [flag] clicked
set x to -100
set y to 0
forever
    change y by -2
```

```
when space key pressed
change y by 15
next costume
```

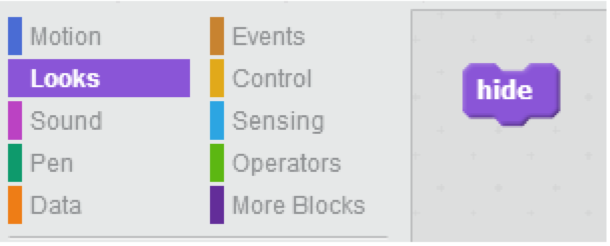
4 Programming the Obstacles

This next section builds up some scripts for the Obstacles sprite.

4.1 Cloning

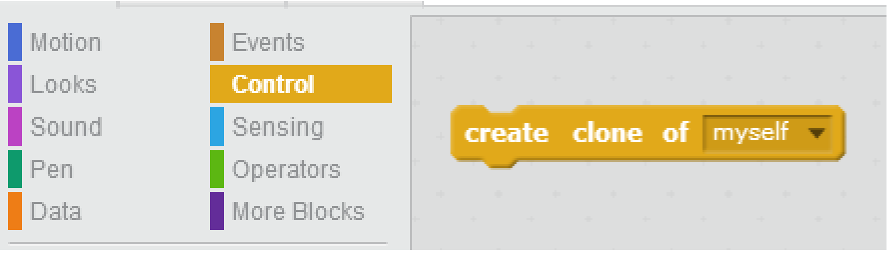
We want the obstacles to keep coming until the game ends, but instead of making loads of different sprites, we can make lots of copies of a single sprite.

Firstly, we need to make the original invisible:



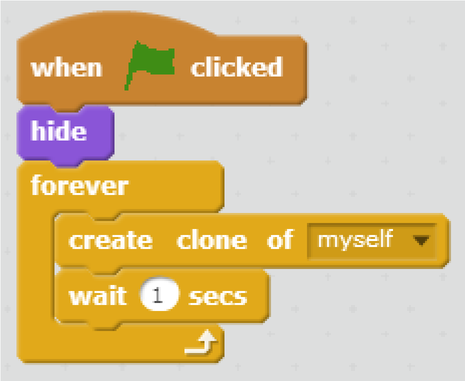
```
hide
```

and then repeatedly create a clone:



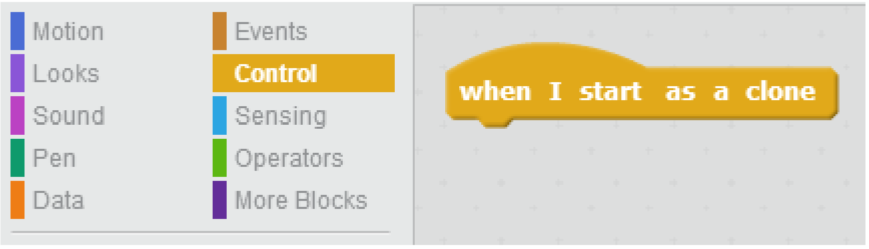
```
create clone of myself ▾
```

Overall, this looks like:



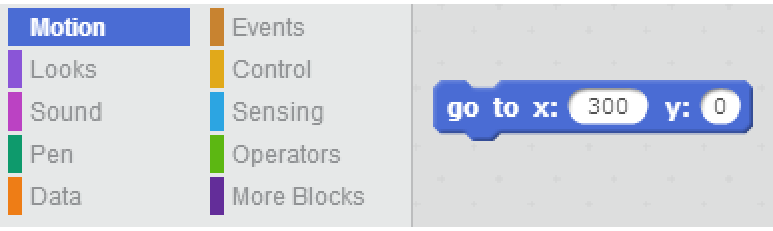
```
when 🏴 clicked
hide
forever
    create clone of myself ▾
    wait 1 secs
```

Now, we need to tell the Obstacles Sprite what to do when it is cloned:



```
when I start as a clone
```

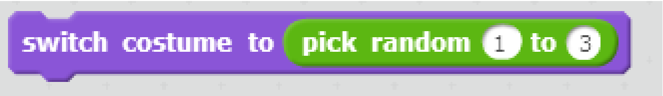
The obstacle needs to start off to the right of the screen, but be centered in the middle, so add in:



```
go to x: 300 y: 0
```

4.2 Randomisation

We want to select one of the costumes by random:



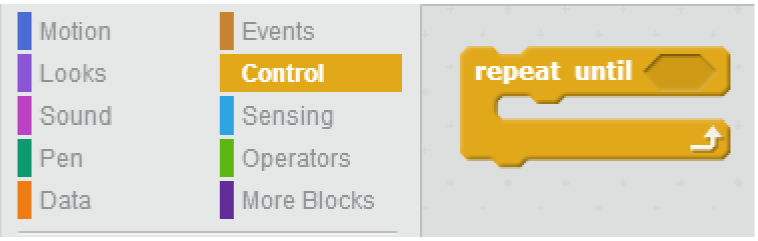
```
switch costume to pick random 1 to 3
```

Next we should undo the invisibility, so the clone is visible. Try to find the code piece that does that.

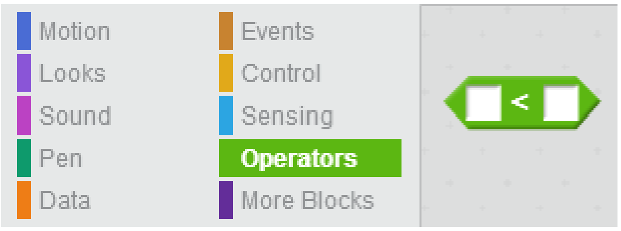
4.3 Moving

Now our cloning is set up, we need to make them move across the screen.

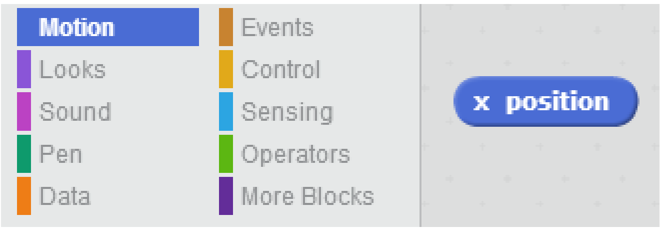
Similarly to the gravity in the "Bird" sprite, we're going to put a movement piece inside a loop.
However, we want our loop to end when the sprite gets to the opposite edge of the screen, so try combining:



```
repeat until
```

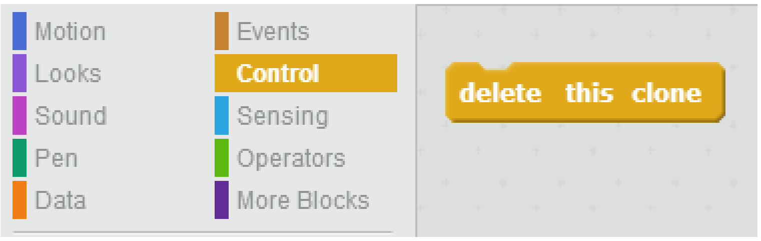


and

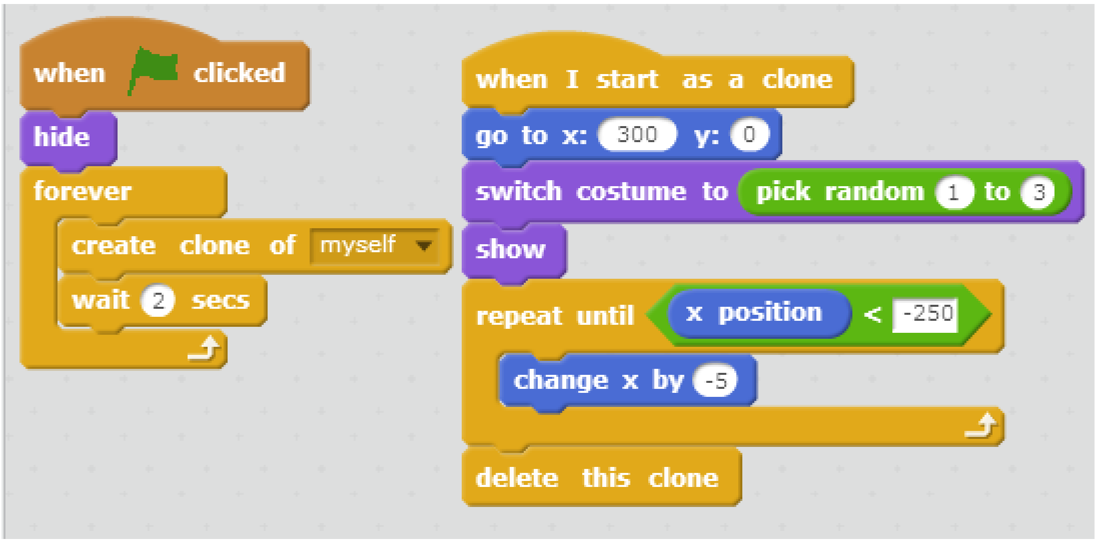


to make the loop stop when the clone reaches the left edge of the screen (where the X-coordinate is -250).

Once we don't need a clone anymore, it is really important to delete it:



Now your code should look something like this:



```
when ⚑ clicked
hide
forever
    create clone of myself
    wait 2 secs

when I start as a clone
go to x: 300 y: 0
switch costume to pick random 1 to 3
show
repeat until x position < -250
    change x by -5
delete this clone
```

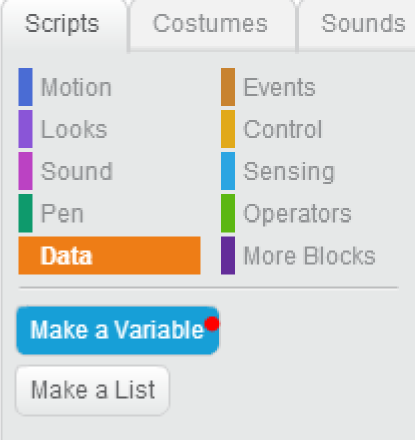
You might want to have a fiddle with the "wait" and "move" numbers here and in the bird sprite, so that the game is as easy or as hard as you wish.

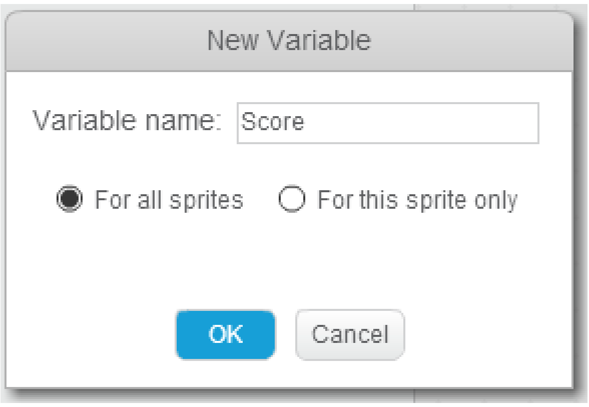
5 Scoring

In this section, we will go through how to keep track of our Score in the game. To do this, we are going to use something called a "Variable", which is just a way to store information.

5.1 Make a new Variable

Go to to your obstacle sprite, and in the Data section, select "Make a Variable":

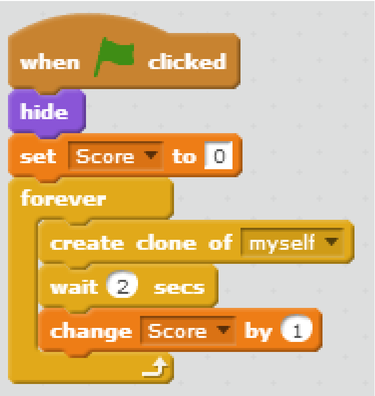




Variable name: Score

● For all sprites ○ For this sprite only

We want our Score to start at 0 each game, and then it to be increased by one each time a new obstacle is cloned, so add the orange data blocks to your code:



```
when ⚑ clicked
hide
set Score ▾ to 0
forever
    create clone of myself ▾
    wait 2 secs
    change Score ▾ by 1
```