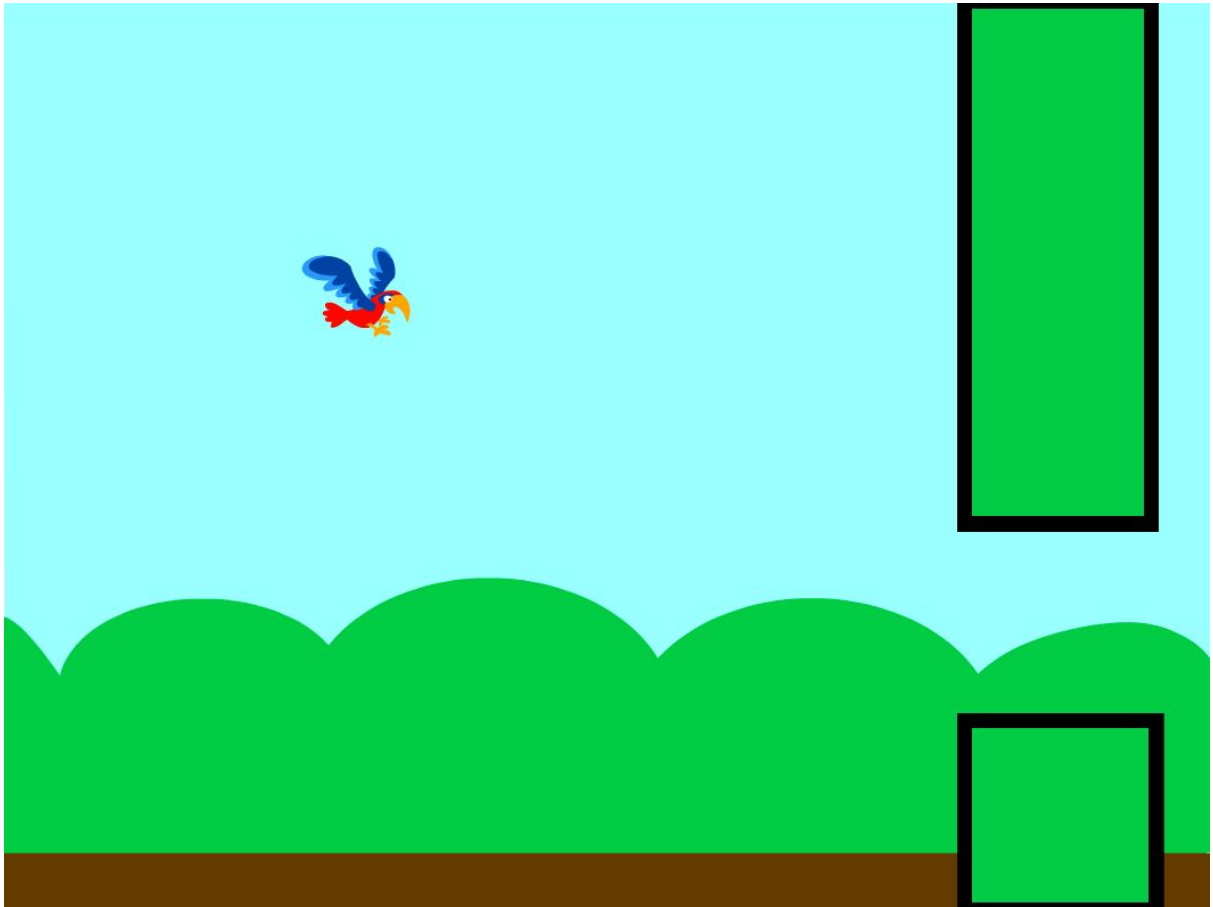


# Coding Flappy Bird with Scratch



By Luke Storry, 15th Nov 2016

Updated by Joe Brown & Aidan Higgins, 13th Feb 2019

# 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.

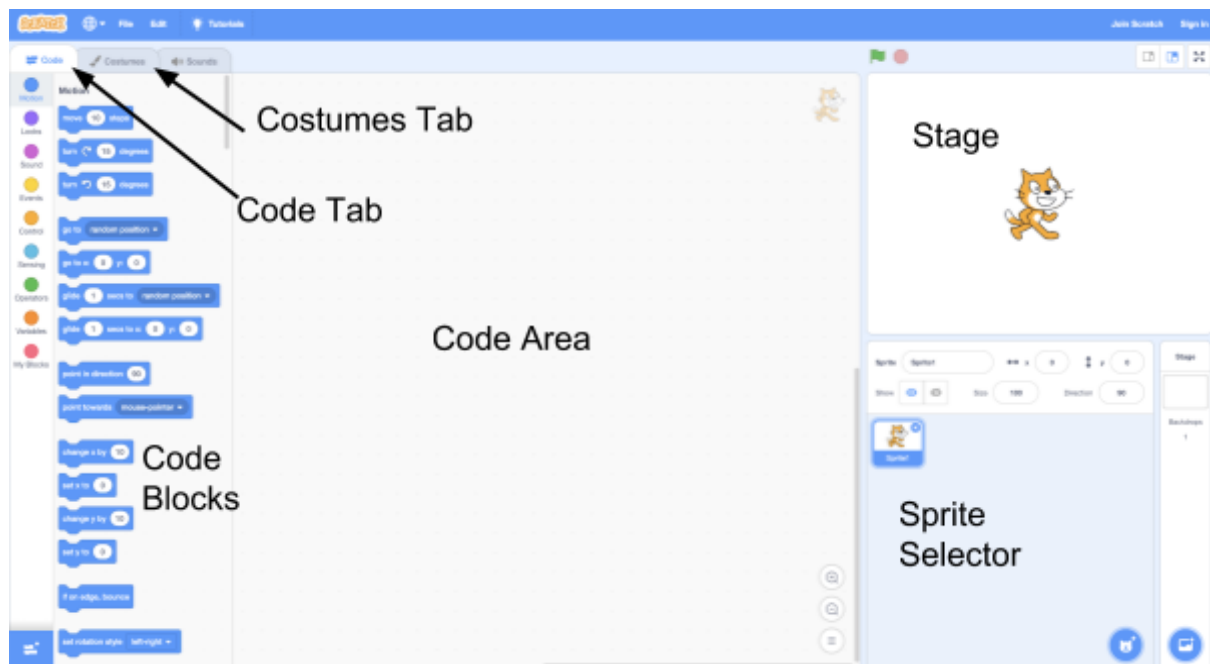
1.1 You should already know:

- some basics of how to use Scratch. (But you'll be able to pick it up if you don't!)
- about coordinates (x & y).

1.2 You will learn to:

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

This is the layout for scratch will all the sections we will be using in this workshop:



## 2 Let's Begin

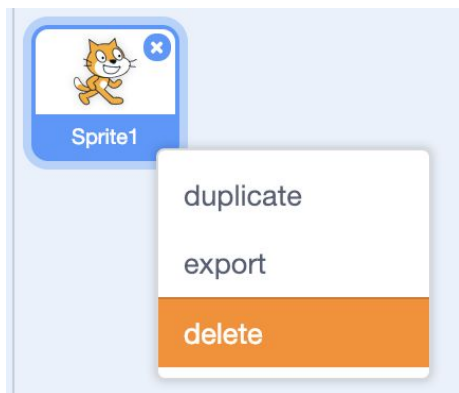
### 2.1 Create a new Project

Go to <https://scratch.mit.edu> and click "Create" in the top bar of the website. This will start a new project.

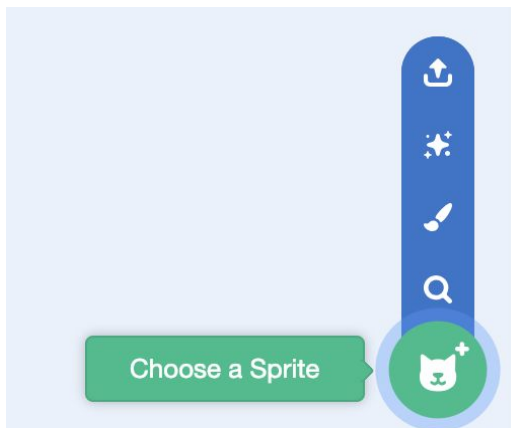


### 2.2 Replace the Main Sprite

I want a flappy bird, not a cat, so let's remove the default cat sprite. The sprite section is in the bottom right of the screen:



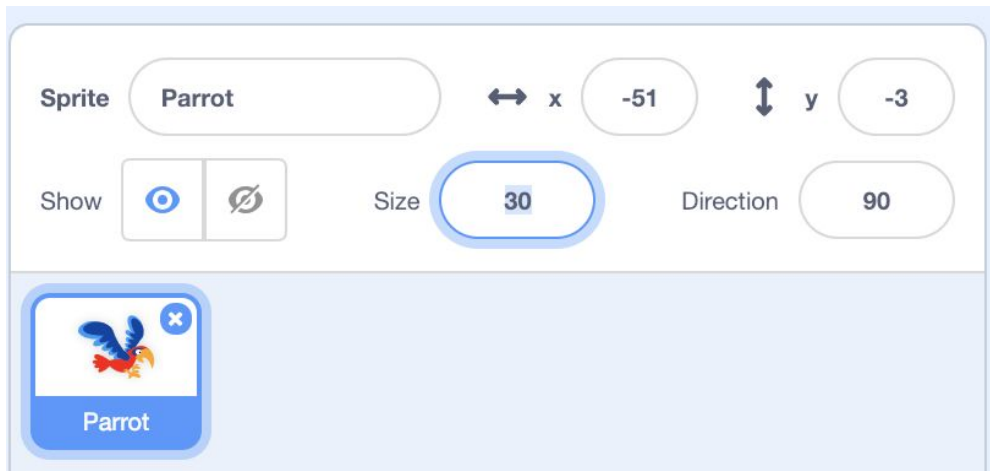
...and replace it with a different sprite



I chose this parrot, feel free to choose another animal if you want. There's an animal button and a search bar to find one (or draw one if you're feeling adventurous!)

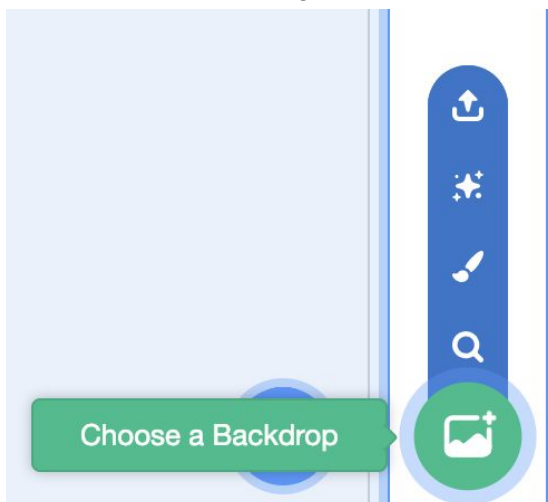


This is a bit big, so make the animal smaller by changing size from 100 to 30:



### 2.3 Replace the Background

Next up is to replace that boring background with something more fun. Click the backdrop button on the bottom right:

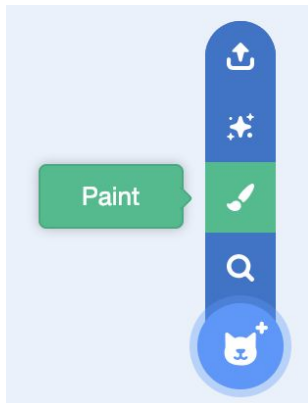


I chose the Blue Sky background as I think it fits with my parrot, but again, feel free to customise and choose your own background.

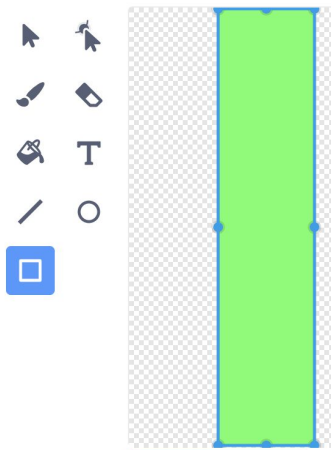


## 2.4 Create the Obstacles

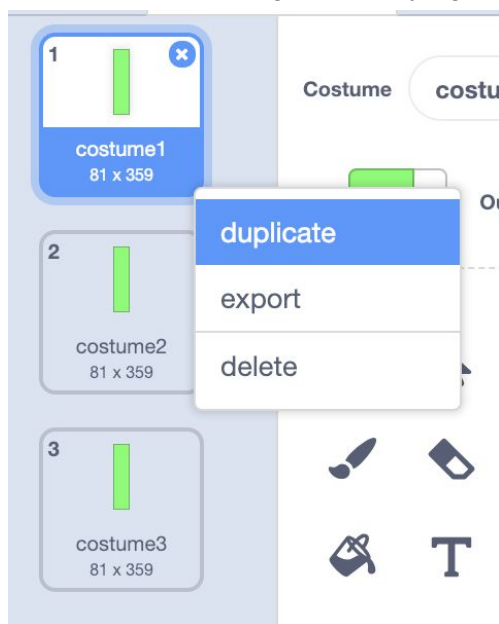
The last thing we need to do before we start scripting is create the sprite for our obstacles. Hover on the sprite button and click the paint button:



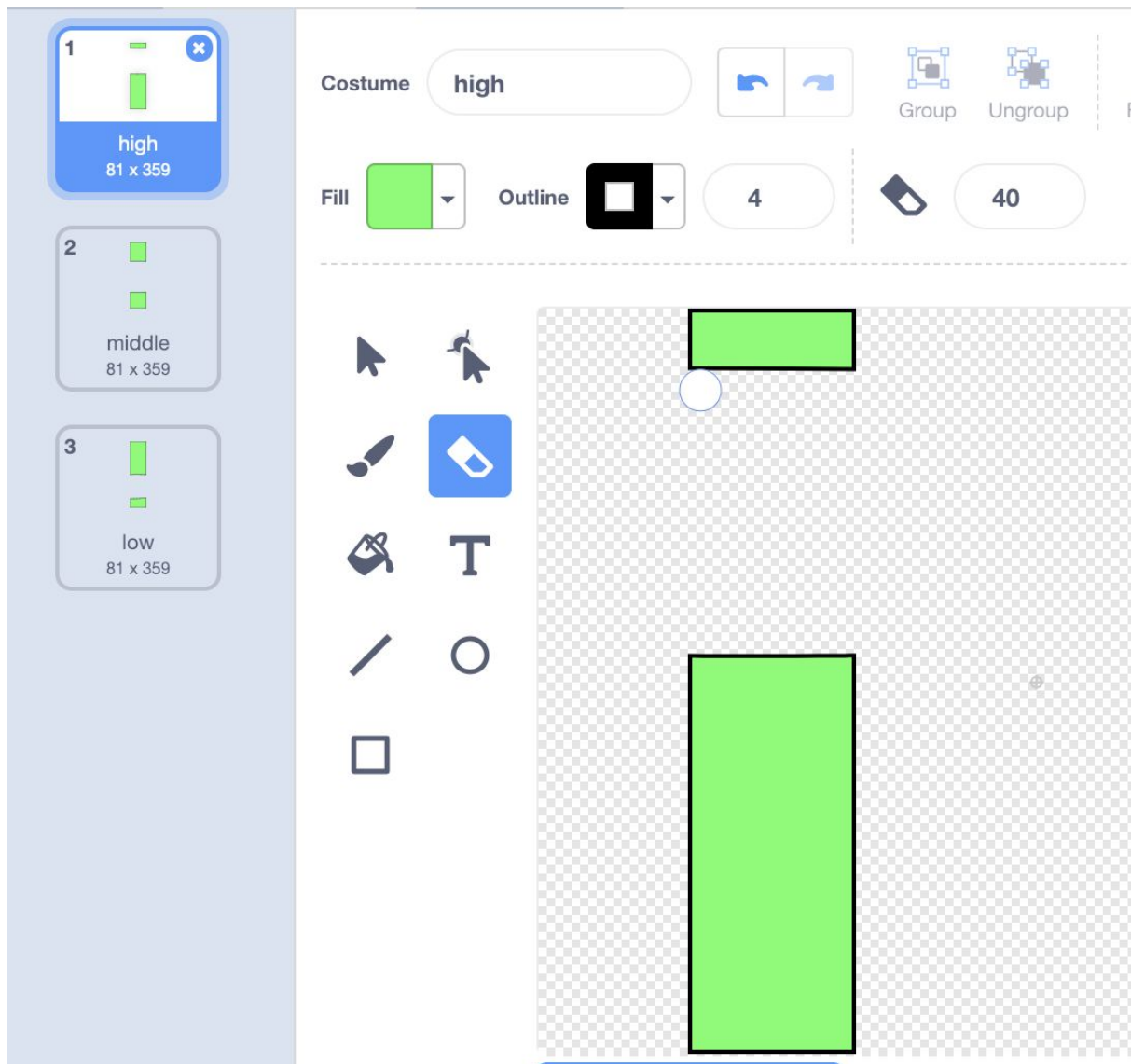
We need a few different costumes for the obstacles pipe, so we can vary between them. Create a tall rectangle for your first obstacle, make it as tall as the draw box:



Duplicate this rectangle twice by right clicking so you have three pipes:



Use the eraser tool to erase a gap in the pipes - one high gap, one middle gap and one low:



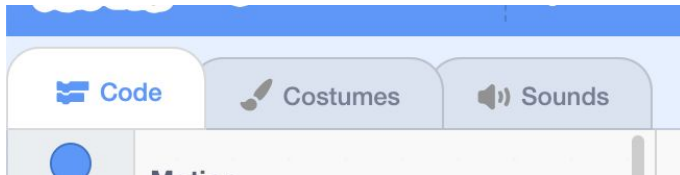
You can set the eraser to a larger number to erase easier.

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

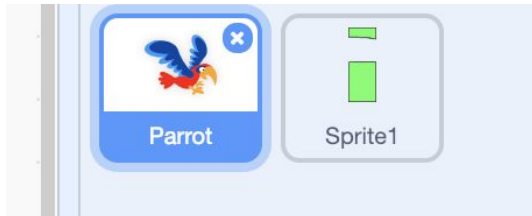
Be creative!

### 3 Programming the Bird

In this section we are going to think about what code we should give our "Bird" sprite. Go back to the code tab on the top left:



and make sure you've selected your animal sprite in the sprite section on the right:



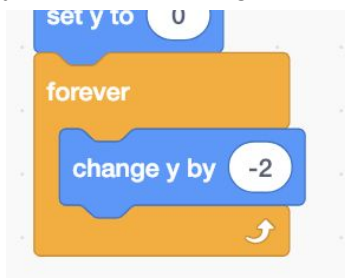
#### 3.1 Initialisation

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:



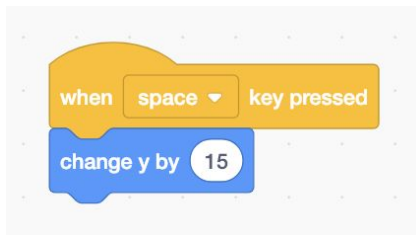
#### 3.2 Gravity

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



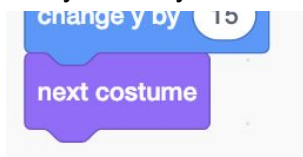
### 3.3 Flight

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



### 3.4 Animation

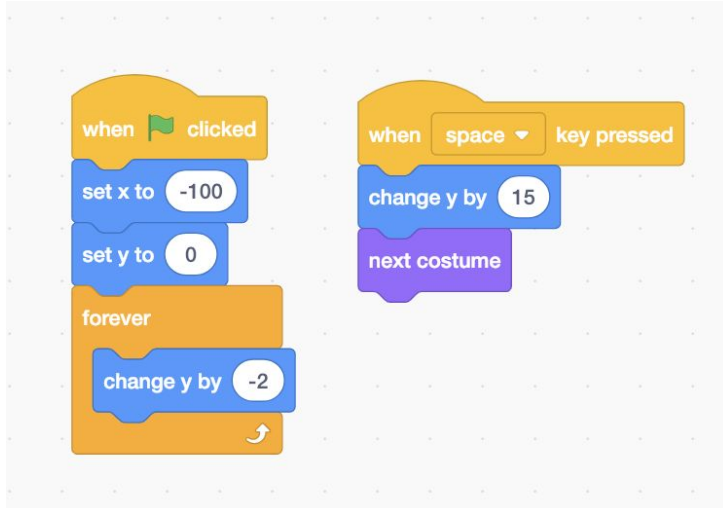
Wouldn't it be cool if the bird's wings could flap when its flying? This can be easily done by switching between different costumes:



The Parrot Sprite that I'm using already 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

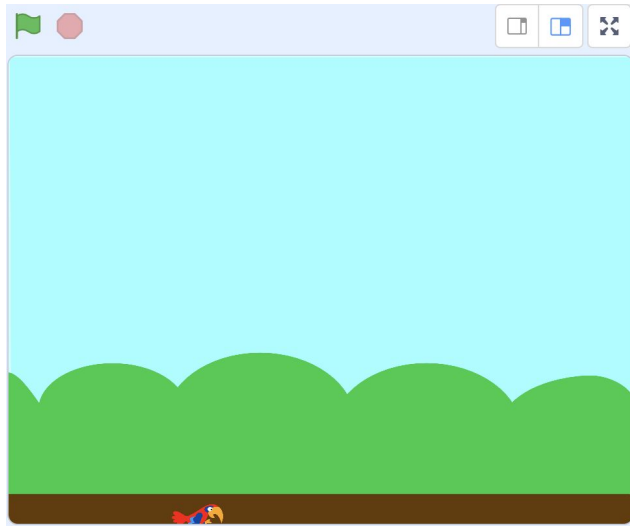
Therefore, the final code for the Bird sprite should look like this:





### Now test your code!

Press the green flag above the stage to try out your code. You can use the expand button on the right to make the stage window larger.



Does your animal fall? Does it fly upwards when you press the spacebar? Do the costumes of your sprite swap when you press the spacebar?

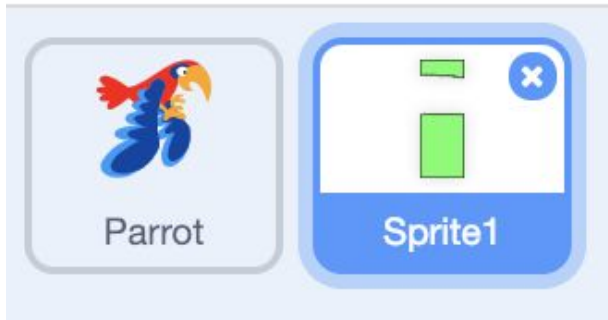
Press the red stop button to stop your code running. Have a look at your code and modify it, if you don't think it behaves the way you think it should.

Remember to stop and restart when you make changes for them to make effect. Ask for help if you're stuck.

## 4 Programming the Obstacles

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

Make sure you've selected your obstacle sprite in the sprite section on the right:



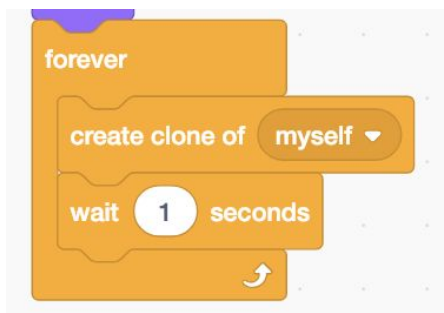
### 4.1 Cloning

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.

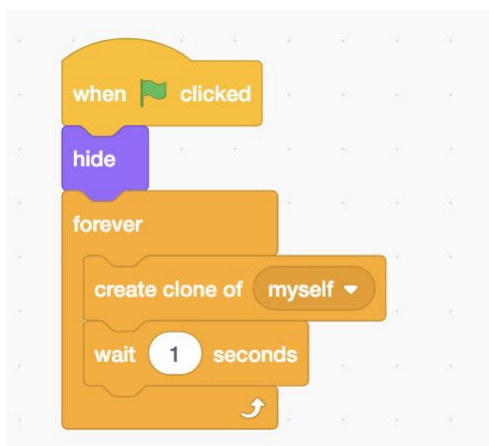
Firstly, we need to make the original invisible:



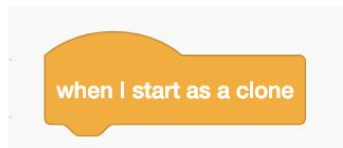
and then repeatedly create a clone:



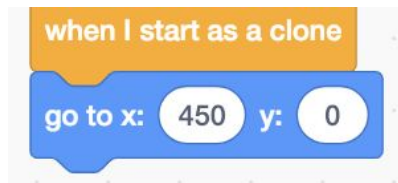
Overall, this looks like:



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

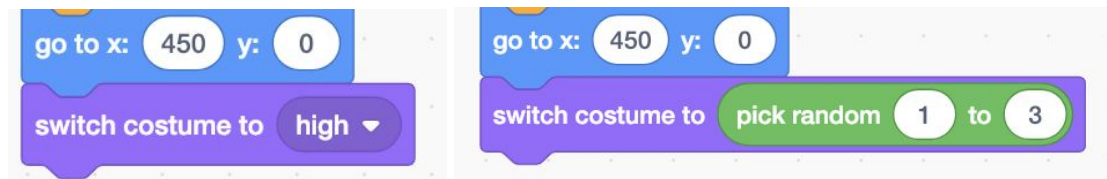


The obstacle needs to start off to the right of the screen, but be centered in the middle, so add in:



#### 4.2 Randomisation

We want to select one of the costumes by random by adding 'switch costume' and 'pick random'

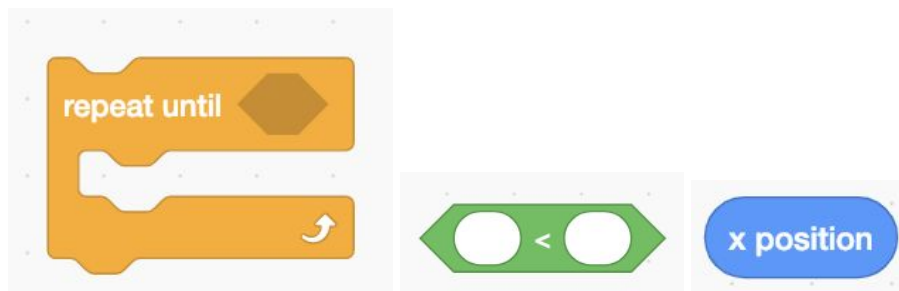


Next we should undo the invisibility, so the clone is visible. Try to find the code piece that does that.

#### 4.3 Moving

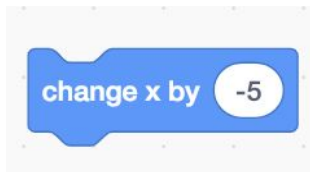
Now our cloning is set up, we need to make them move across the screen. Similarly to the gravity in the "Bird" sprite, we're going to put a movement piece inside a loop.

However, we want our loop to end when the sprite gets to the opposite edge of the screen, so try combining these:

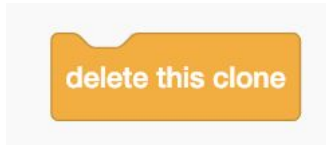


to make the loop stop when the clone reaches the left edge of the screen (where the X-coordinate is -250).

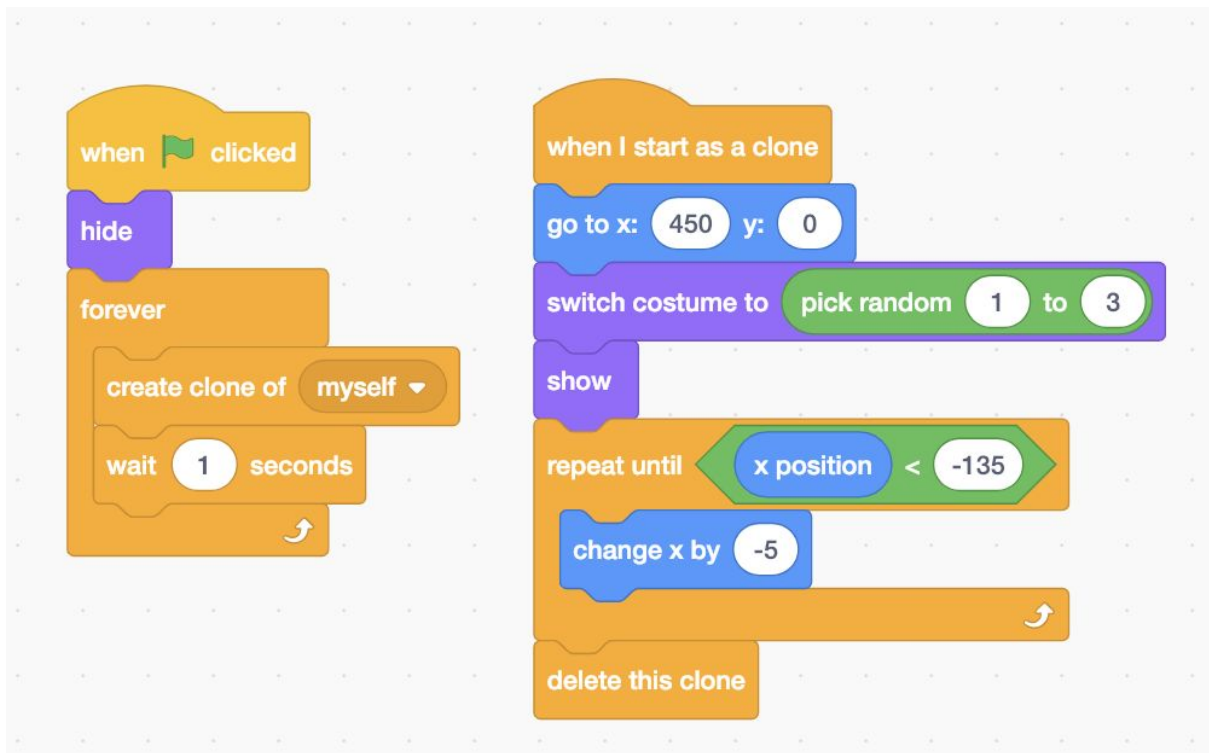
Use change x by to move the obstacle across the screen:



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



Now your code should look something like this:



### Now test your code!

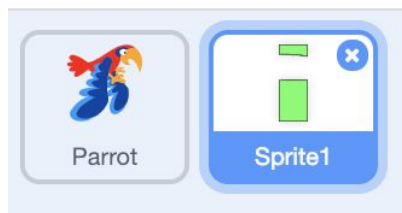
Do the obstacles behave like you expect? Can you tweak their movement and position to behave right? Do they get deleted on the left side of the screen?

Try tweaking the 'go to x' and 'x position' parameters.

Remember to stop and restart when you make changes for them to make effect.

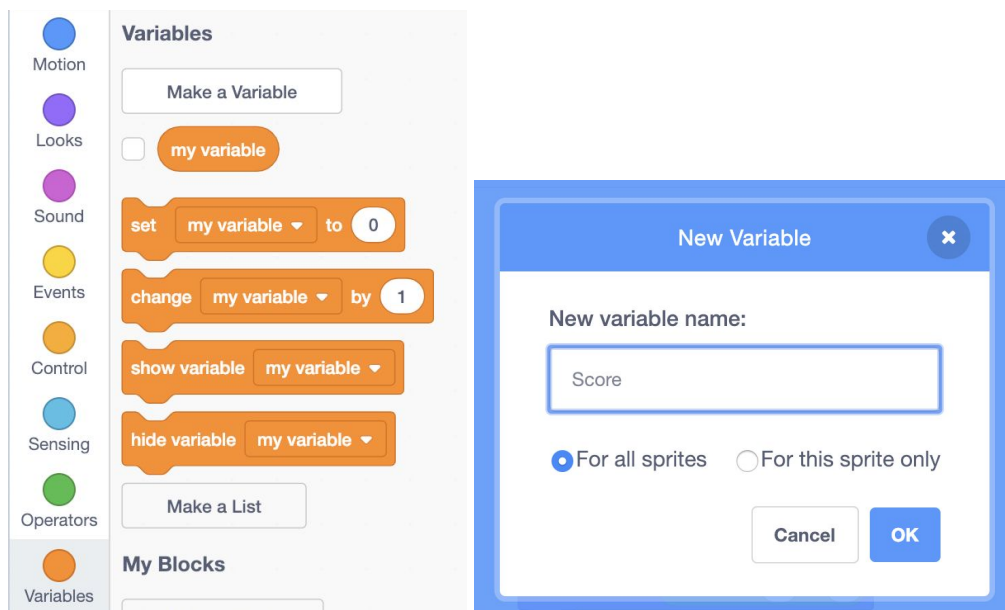
## 5 Scoring

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. Select your obstacle sprite:

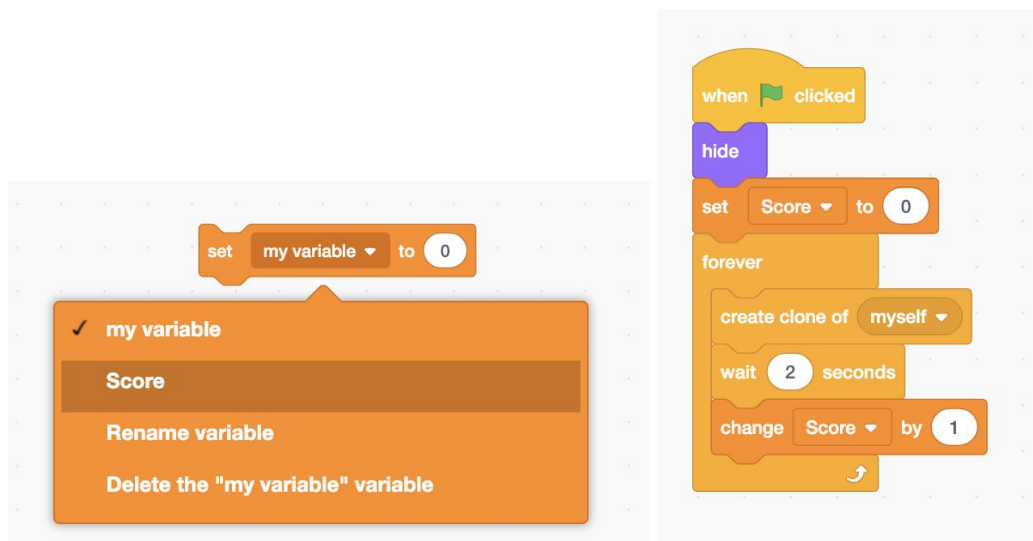


### 5.1 Make a new Variable

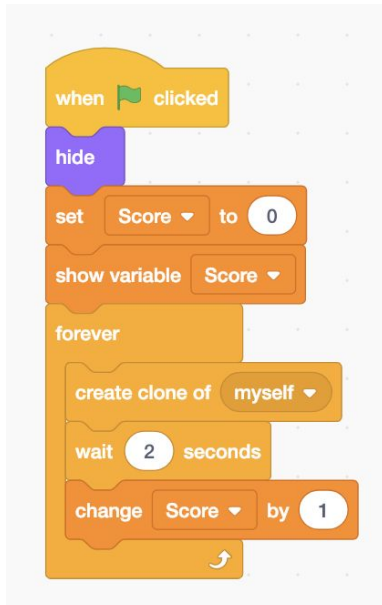
In the variables section, select "Make a Variable" and call it "Score":



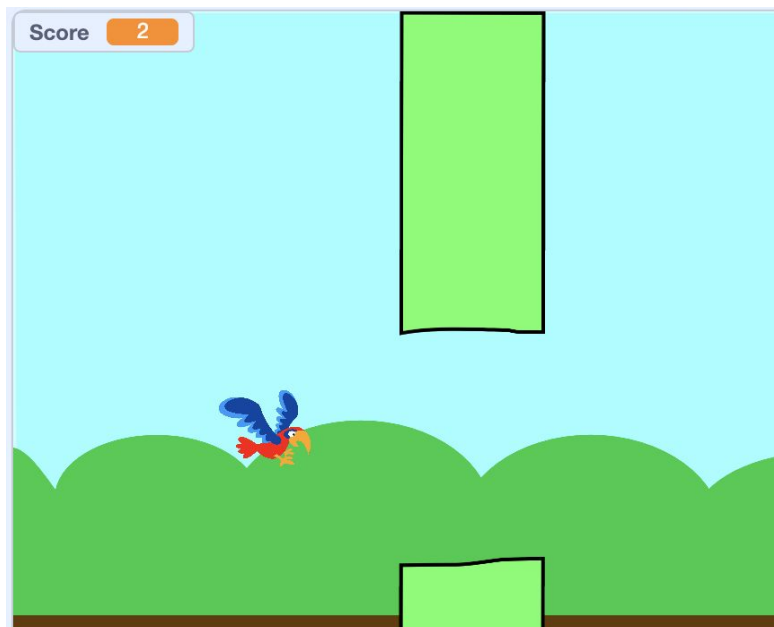
We want our Score to start at 0 each game, and then it to be increased by one each time a new obstacle is cloned, so add the orange data blocks to your code for "set" and "change":



To make sure you can see your score on the stage, add a “show variable” from the “variables” section and make sure it’s set to “Score” and add it after the “set: Score to 0”:



You should now see it on the stage. So now every time you pass an obstacle, your score should increase by 1!



### Now test your code!

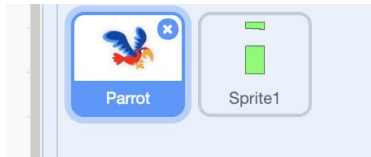
Do you see your score go up as you pass an obstacle?

Remember to stop and restart when you make changes for them to make effect.

## 6 Ending the Game

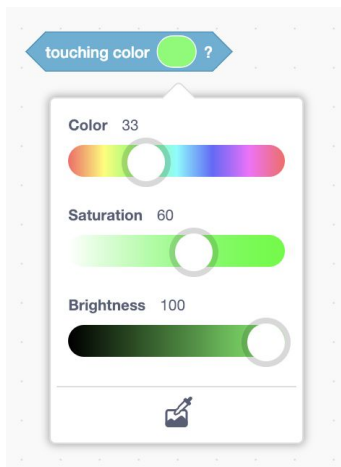
Currently you can't lose the game, you can just fly through obstacles. That's not very fun to play! In this section we're going to add "collision detection". This is when our animal hits an obstacle we end the game

Select the animal sprite:



### 6.1 Touching Colour

We can use "touching color" to detect when the animal touches the obstacle. Select the "touching color" code block from the "Sensing" section:

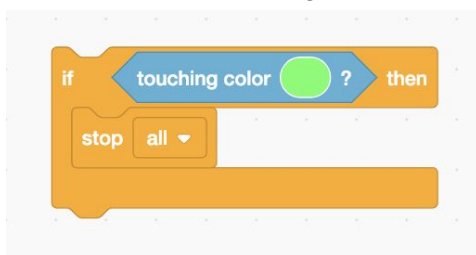


Use the colour picker tool at the bottom of the colour box to select the colour of your obstacle. If you can't see the obstacle, hit the green flag first and then pick the colour of your obstacle from the game.

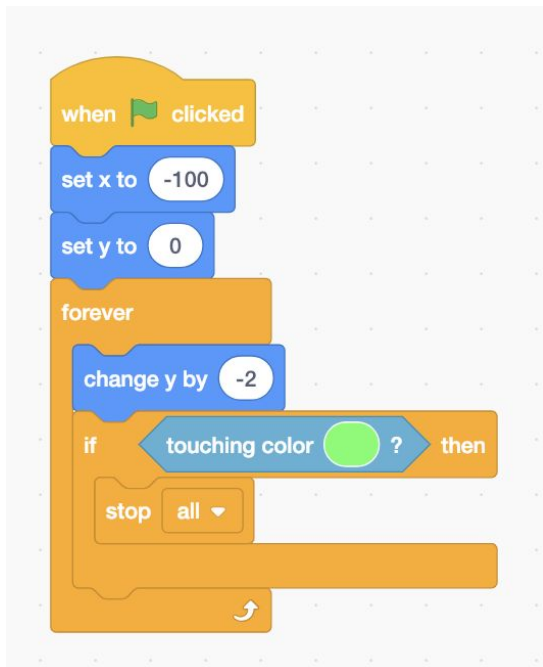
Add the "touching color" to an "if" block from the "Control" section.

Now add "Stop all" block from the "control section"

It should look something like this:



Add this within to your “forever” loop:



So now, if your animal touches the colour of your obstacle, the game should end!

### Now test your code!

Does the game end when you hit an obstacle?

Remember to stop and restart when you make changes for them to make effect.

### Finished

Congratulations you’ve made Scratch Flappy Bird! You have a playable game with a score and a losing condition.

If you have extra time left in the workshop, **on the next page there are extensions** to make the game more like the original flappy.



## Extensions

Can you add these features to fill out the game to be more like the original flappy bird?

Easy:

**Sound Effects:** See “Sound” section of code blocks.

Can you trigger sounds for the game starting, the bird flapping, and hitting an obstacle?

You can add more sounds on the sound tab at the top left.

**Lose by hitting the ground:** Can you make the game end if you hit the ground OR hitting an obstacle? You’ll need a code block from the “Operators” section

Medium:

**High Score:** If your score is better than any previous score, you get the new high score!

Steps to make this:

- Create a variable called “high score”
- Create an “If” for “current score > high score” where “>” means greater than
- Set the high score to current score.
- Use “show variable” (just before “stop: all”) to show the high score after you’ve lost and “hide variable” (at the beginning) if you want to hide it for the rest of the time

Hard:

**Velocity:** Instead of changing the y position of the bird there is a variable called “velocity”

This gets increasingly big as you fall down the y-axis and then set to a positive number when spacebar is pressed - this is exactly how the original flappy bird works!

The steps to make this:

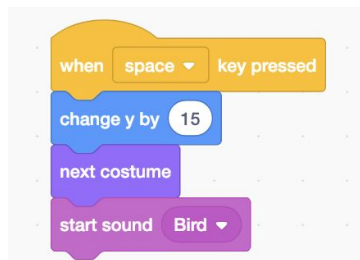
- Create a variable called “velocity”
- Set “velocity” to 5 when you hit the spacebar.
- Change “velocity” by -0.5 in the “forever” loop
- Instead of changing y by -2, change it by “velocity”

So you start falling faster and faster until you hit the ground or hit the spacebar!

**Solutions to these are on the next page.** But have a go before looking!

## Extensions Solutions

Sound example:



Lose by hitting ground solution:



Highscore solution:



Velocity solution:

