FIRST CODING
creative coding for kids

SNAKE!

Introduction

Scratch is a visual programming language suitable for children and teens to create their own interactive stories, games and animations. Projects can also be uploaded and shared with the Scratch community via the Scratch website for others to enjoy.

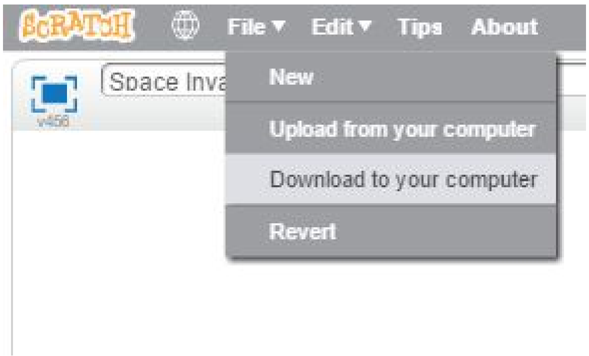
For this lesson we're going to build a simple snake game, moving an ever growing snake around the screen, avoiding double backing on itself whilst staying clear of the screen's edge. We'll award ourselves points as we progress through the game. This game appears simple at first but soon becomes a tricky one to master.

Let's get started by first loading Scratch. You can either install Scratch on your computer or load it via your web browser by navigating to https://scratch.mit.edu/

Top Tip

Don't forget to save your work regularly. It's good practice to get in the habit of saving any progress you've made. So when you're feeling great after solving a coding problem, save it, save it, save it!

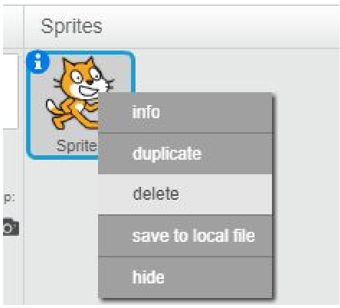
The save options may differ slightly between Scratch versions but head on up to the File option in the top left and you'll find it.



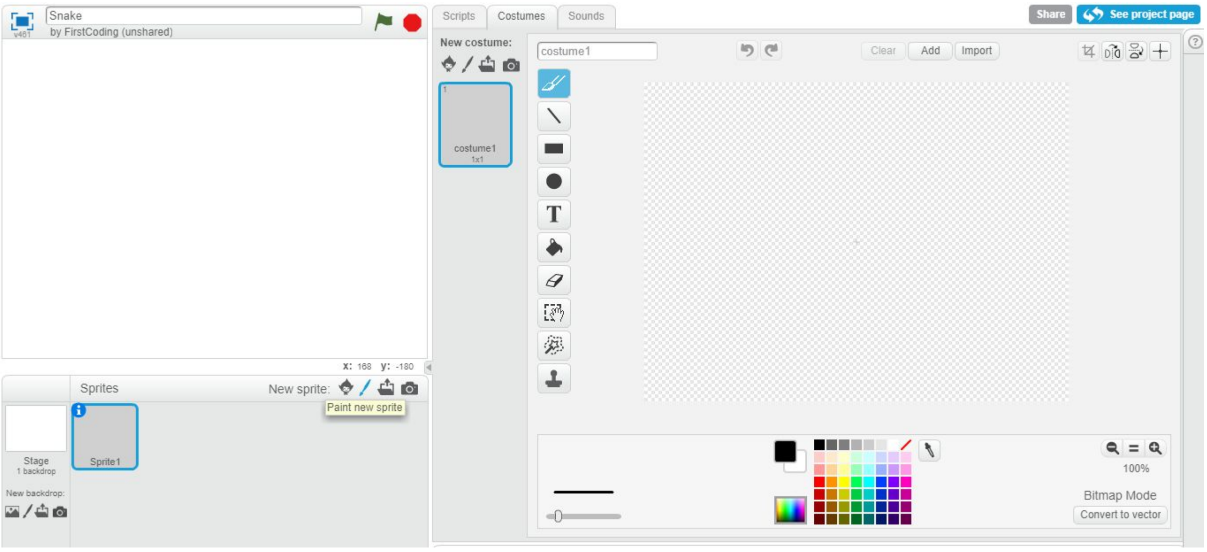
Let's get started

Creating the snake sprite

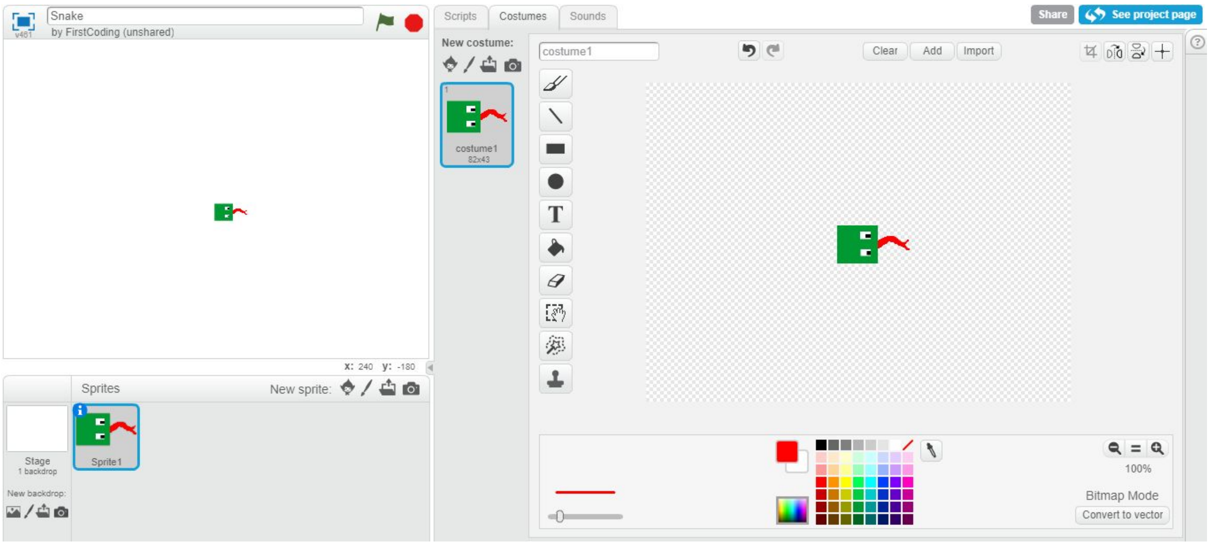
The first thing we need to do is delete the cat sprite that gets created by default. To do this just right click on the sprite in the bottom left corner and select DELETE. This will give us a good clean starting point for our snake game.



Select the PAINT NEW SPRITE option as shown will allow you to create the snake in the editor on the right.

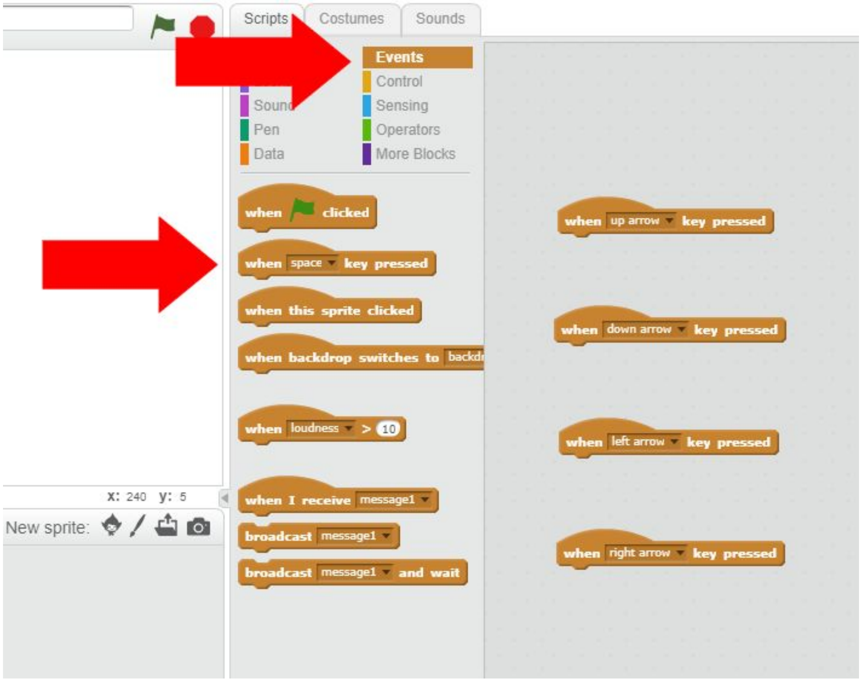


We only need the head of the snake creating at this point. Make sure the snake’s head is small in size and most importantly has a red tongue sticking out. We will be using the red tongue for colour detection later on.



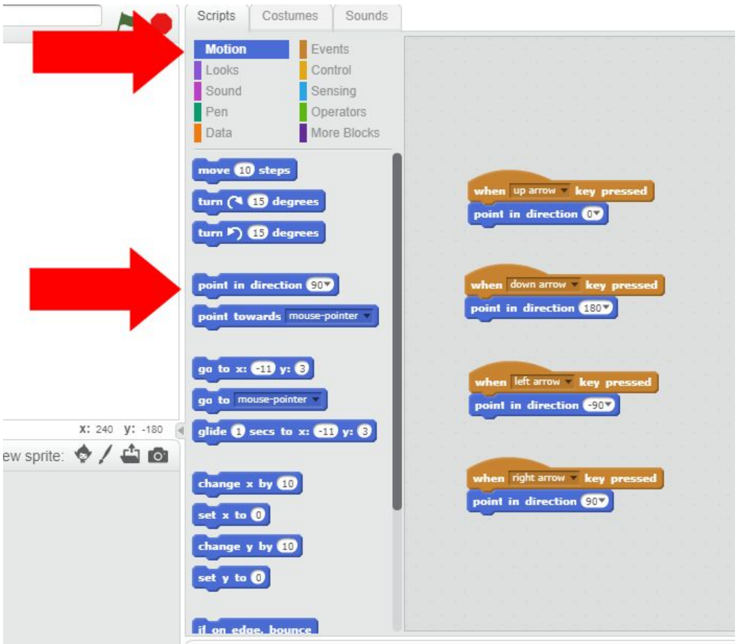
Controlling the snake

We're going to be controlling our snake using the arrow keys. In order to detect what keys are pressed, use the WHEN [KEY] IS PRESSED block found in the EVENTS section as shown. Don't forget we'll need four of these blocks to detect when either the UP, DOWN, LEFT or RIGHT arrow keys are pressed.



```
when up arrow key pressed

when down arrow key pressed

when left arrow key pressed

when right arrow key pressed
```

Once an arrow key is pressed, the snake will need to move in that direction. As the snake will be moving on its own, we'll just be changing the direction it'll be heading in. This is done using the POINT IN DIRECTION block found in the MOTION section. Don't forget to use the dropdown in this block to set the direction of travel. Once completed, it should look something like this...

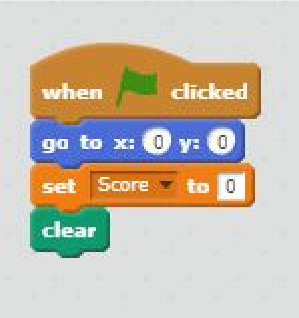


```
when up arrow key pressed
point in direction 0

when down arrow key pressed
point in direction 180

when left arrow key pressed
point in direction -90

when right arrow key pressed
point in direction 90
```

Let’s get things moving

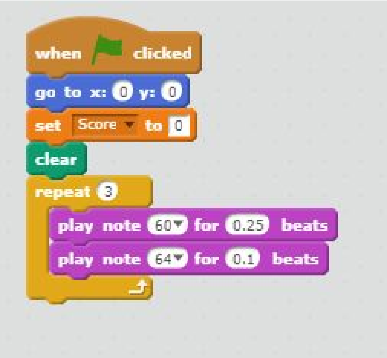
In order to get the snake moving, we’ll slowly build up the logic needed for the game. The first stage is to locate the GREEN FLAG block in EVENTS as a press of this will tell the computer that we want to start the game.

Once a new game starts we need to ensure the snake starts from the correct position every time and that any previous scores have been reset to zero. We do this by building the following blocks.



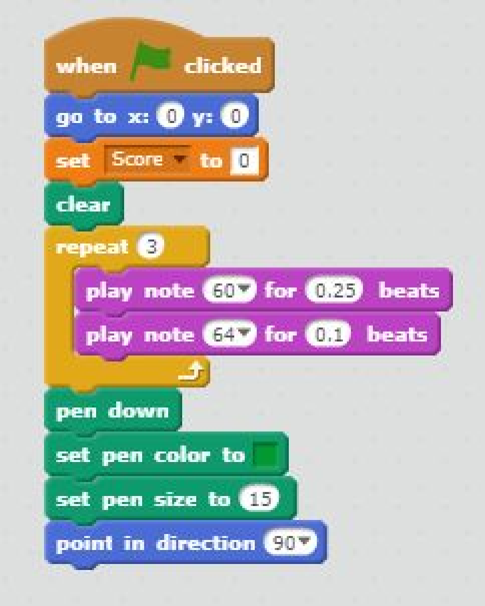
```
when 🏁 clicked
go to x: 0 y: 0
set Score ▾ to 0
clear
```

Next we’ll play a little introductory music to indicate that a new game is about to start. A simple tune can be achieved by repeating a couple on notes. Add the following notes and REPEAT block as shown.



```
when 🏁 clicked
go to x: 0 y: 0
set Score ▾ to 0
clear
repeat 3
    play note 60▾ for 0.25 beats
    play note 64▾ for 0.1 beats
```

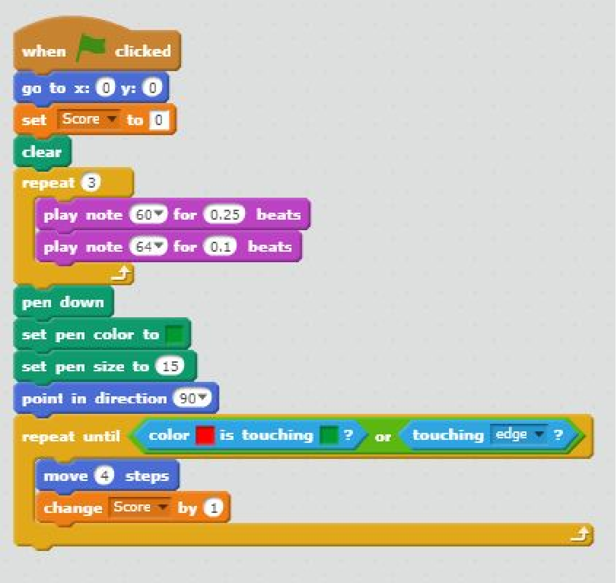
Before the snake moves we need to set the pen’s size and colour as well ensure that it’s been placed down on the canvas.



```
when 🏁 clicked
go to x: 0 y: 0
set Score ▾ to 0
clear
repeat 3
    play note 60▾ for 0.25 beats
    play note 64▾ for 0.1 beats
pen down
set pen color to 🟩
set pen size to 15
point in direction 90▾
```

We are going to continuously move the snake until the head either touches part of its green body or touches the edge of the screen as both these scenarios will mean it's game over.

In order for us to achieve this we first need a REPEAT UNTIL block found in the CONTROL section. The blocks needed for the two situations we'll be testing for can be found in the SENSING section. Incidentally, we're checking to see if the colour red is touching green - this is basically when the snake's red tongue touches its green body.

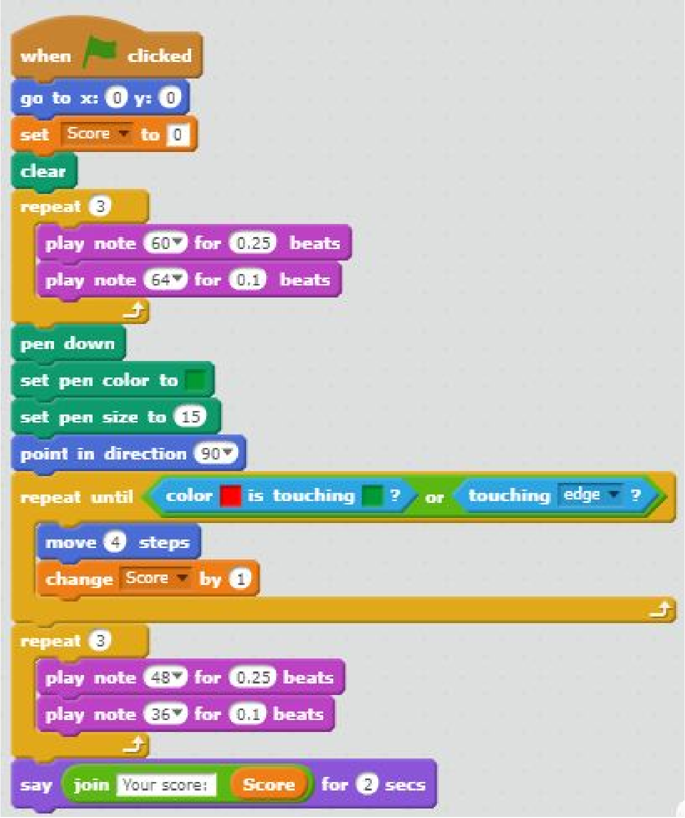


```
when 🏁 clicked
go to x: 0 y: 0
set Score to 0
clear
repeat 3
    play note 60 for 0.25 beats
    play note 64 for 0.1 beats
pen down
set pen color to 🟩
set pen size to 15
point in direction 90
repeat until < color 🟥 is touching 🟩 ? or touching edge ? >
    move 4 steps
    change Score by 1
```

We'll also award ourselves a point for every fours steps moved, so add the CHANGE [VARIABLE] BY block located in the DATA section.

Game Over

Finally, we need to consider what happens when it's game over. The following REPEAT loop will play a simple game over tune but this will only get played once the REPEAT UNTIL loop we built previously has been exited.



```
when 🏁 clicked
go to x: 0 y: 0
set Score to 0
clear
repeat 3
    play note 60 for 0.25 beats
    play note 64 for 0.1 beats
pen down
set pen color to 🟩
set pen size to 15
point in direction 90
repeat until < color 🟥 is touching 🟩 ? or touching edge ? >
    move 4 steps
    change Score by 1
repeat 3
    play note 48 for 0.25 beats
    play note 36 for 0.1 beats
say join Your score: Score for 2 secs
```