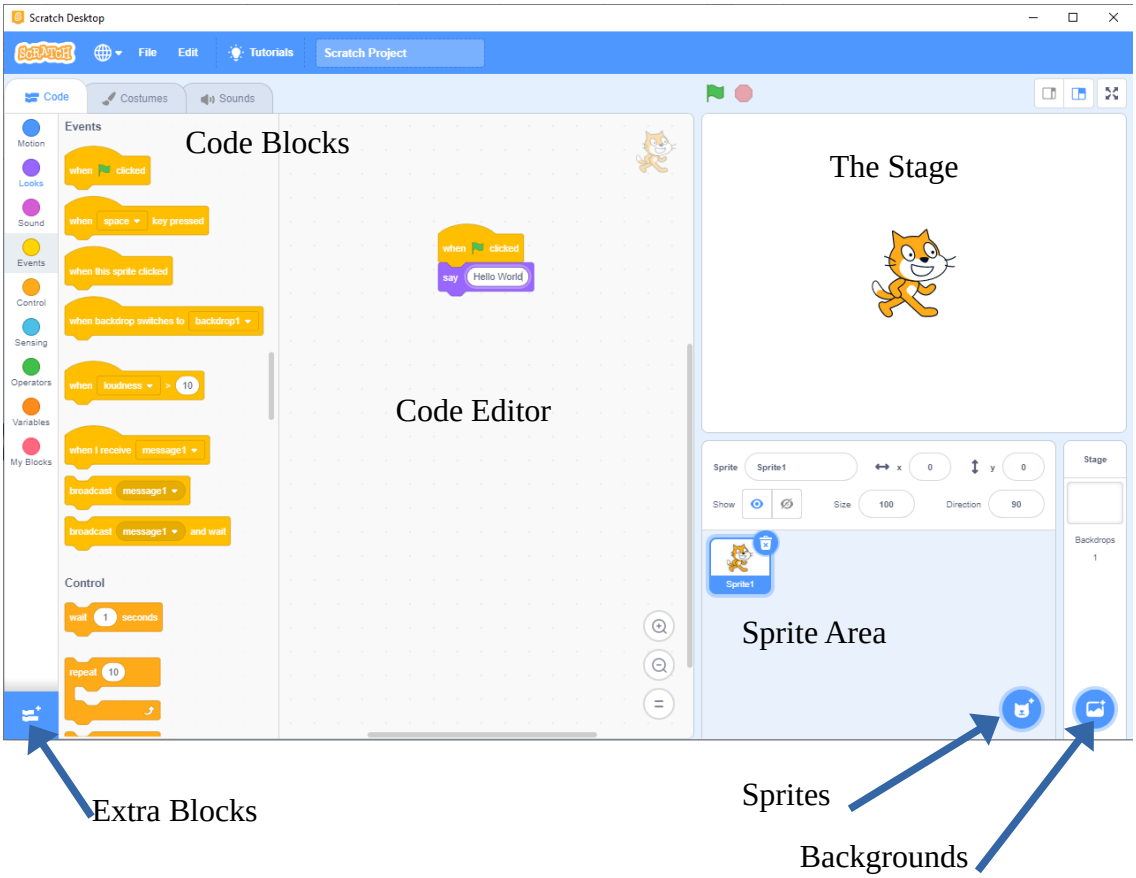
Siavonga Code Club Scratch Workbook

Date Created September 2020

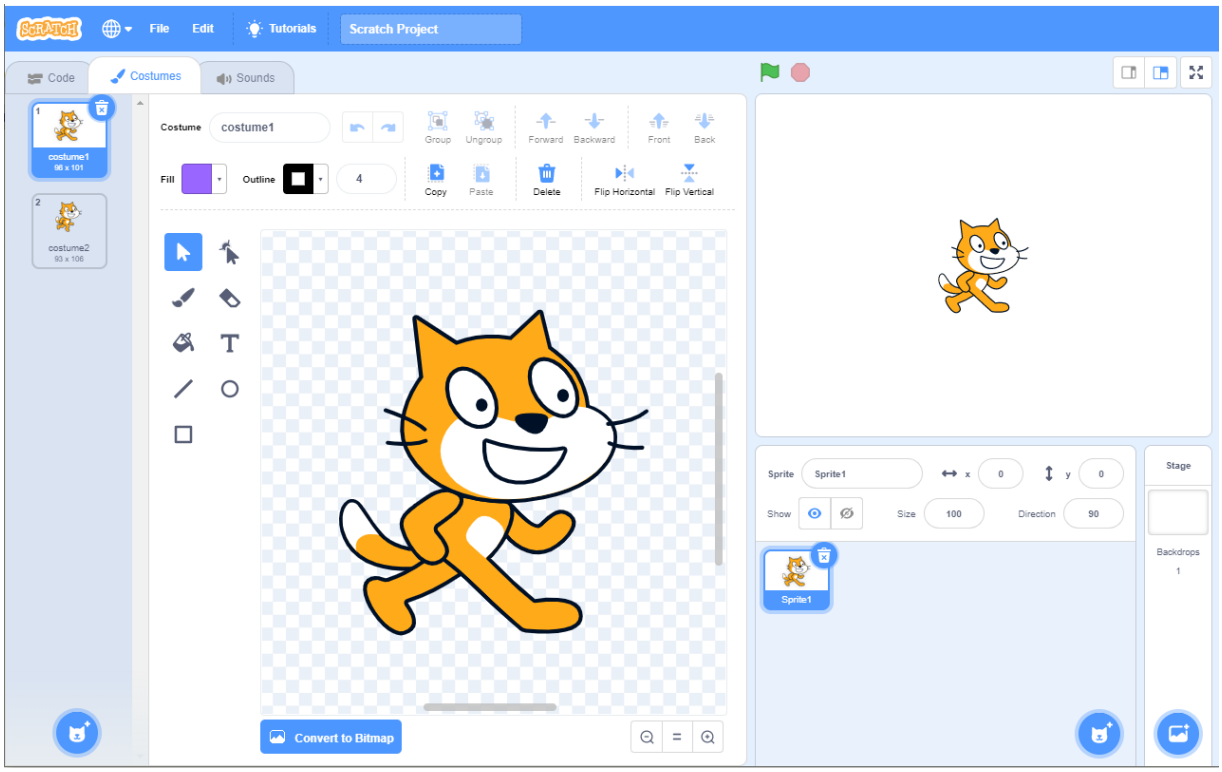
Reviewed

www.giakonda.org.uk info@giakonda.org.uk

The Scratch screens



The Sprite Editor



Movement and Loops

You will learn how to build simple scripts to make a sprite move around the stage.

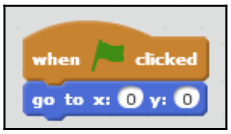
Moving

Program Name: Movement

Click *File/New* to create a new project and enter a project name of *Square*.

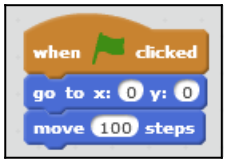
1 From the ***Events*** menu, drag a ***when green flag clicked*** block to the scripts area. Your scripts area should look like this:

From the ***Motion*** menu, drag a ***goto x: 0 y: 0*** block to the scripts area and snap it to the bottom of the



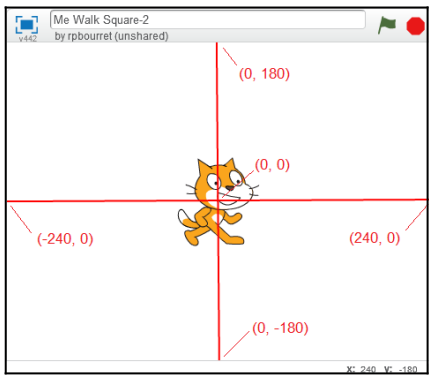
when green flag clicked block. Your script should look like this:

Add a ***move 10 steps block*** to the bottom of your script and change the 10 to 100.



Click the green flag at the top right corner of the stage to run your program. Let's look at what happened:

- The ***when green flag clicked*** block tells Scratch that the other blocks in the script are to be executed at the start of the program — that is, when the green flag is clicked.
- The ***go to x: 0 y: 0*** block tells Scratch to move the sprite — that's the cat, which is what you're writing instructions for — to the coordinates (0, 0). The stage has a coordinate system with (0, 0) in the centre, *x* values ranging from –240 to +240, and *y* values ranging from –180 to +180.



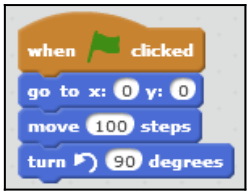
- The ***move 100 steps*** block tells Scratch to move the sprite 100 units in the direction it is facing.

Turning and Waiting

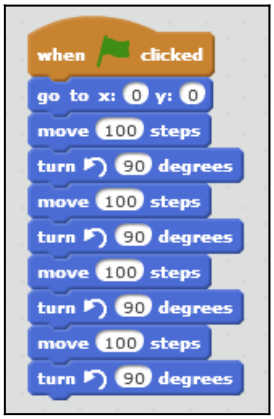
Program Name: Square

Now it's time to make your sprite move in a square.

Drag a ***turn counter-clockwise 15 degrees*** block to the bottom of the script and change the 15 to 90:



Add more blocks until you have four move/turn pairs in your script:



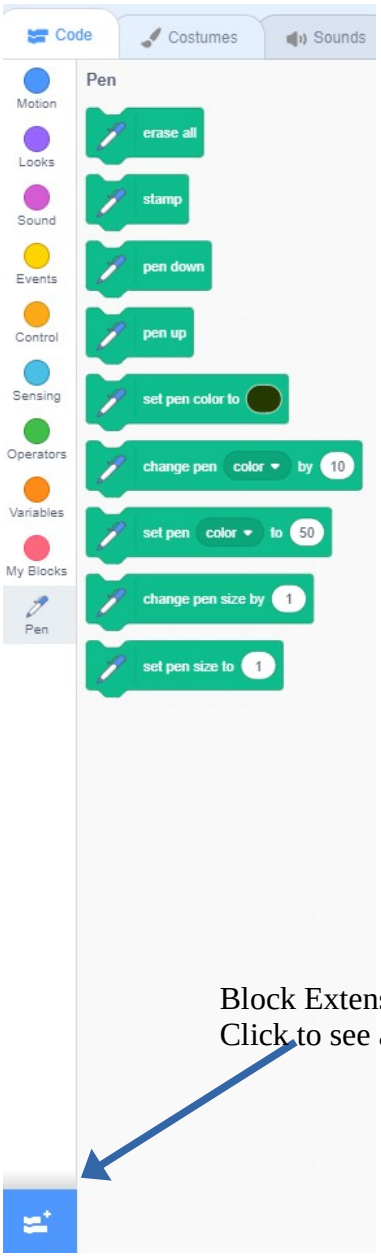
Run your program. What happened? Nothing? Actually, the sprite moved in a square, but so quickly that you couldn't see it.
You can fix that by adding ***wait 1 secs*** blocks from the ***Control*** menu to the stack:

Run your program again. Notice the 1 second pause. If your sprite is too slow, you can always change the wait time, say to 2 seconds.
We still can't see a square so we'll fix that in the next script

The Pen blocks.

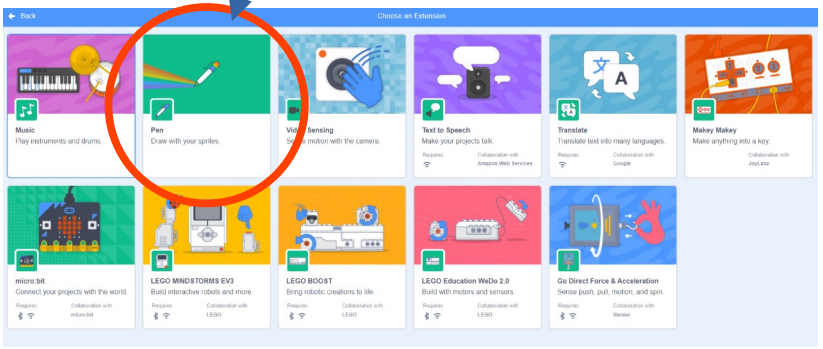
These are a set of blocks that are very useful if you want a record of where the sprite travelled.

To see the blocks you will have to add them to the Block menu. You do this from the Add Extension tool at the bottom left of the block list



Block Extension Tool
Click to see available blocks

The Pen Blocks



Scratch Task: Add the Pen Down block directly below the green flag block and run this script 4 Marks

Repeat Loops

Program Name: Square loop

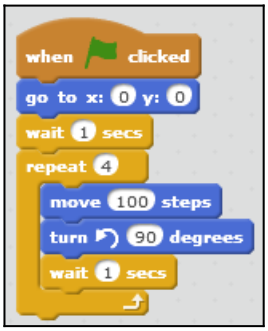
Look at your script. One thing that should strike you is that three blocks — move, turn, wait — are repeated four times. Imagine if you wanted your cat to walk in a square 100 times. You'd need 1,200 blocks! That's insane. (Not to mention a *lot* of work.)

Fortunately, there is an easy way to fix this: the *repeat loop*.

Drag the four sets of move/turn/wait blocks away from the bottom of the script.

Drag a ***repeat 10*** block to the bottom of your script and change the 10 to a 4.

Drag one set of move/turn/wait blocks inside your repeat block. The result should look like this:



Now run your program. Your sprite still moves in a square, but with far fewer commands.

> Repeat loops are an incredibly important programming tool. They allow you to repeat actions without having to add extra blocks, thus saving lots of work. **Use repeat loops!**

More Repeat Loops

Program Name: Square (continued)

You may have noticed that the move block makes your sprite jump from one place to another. Sometimes, this is what you want. Other times, you might want your sprite to move smoothly across the screen. One way to do this is to make it move multiple, shorter distances.

Replace the ***move 100 steps*** block with a repeat loop that moves the sprite 10 times, moving 10 steps each time.

Remove the wait blocks so that the sprite doesn't pause. (Remember, we added the wait blocks only because the sprite moved too fast to see.)

Run your program and watch the sprite move more smoothly in a square.

Forever Loops

Program Name: Square (continued)

What if you want to make your cat really, really dizzy? Or just want to do something forever? There is a special kind of repeat loop for this called a *forever loop*, which you can find in the ***Control*** menu.

Scratch Task: Replace the ***repeat 4*** block with a ***forever*** block.	3 Marks

Run your program and watch your cat go round and round and round and round and ...

Clean-up and Save

Scratch Task: You probably have a lot of unused blocks laying around the script area.

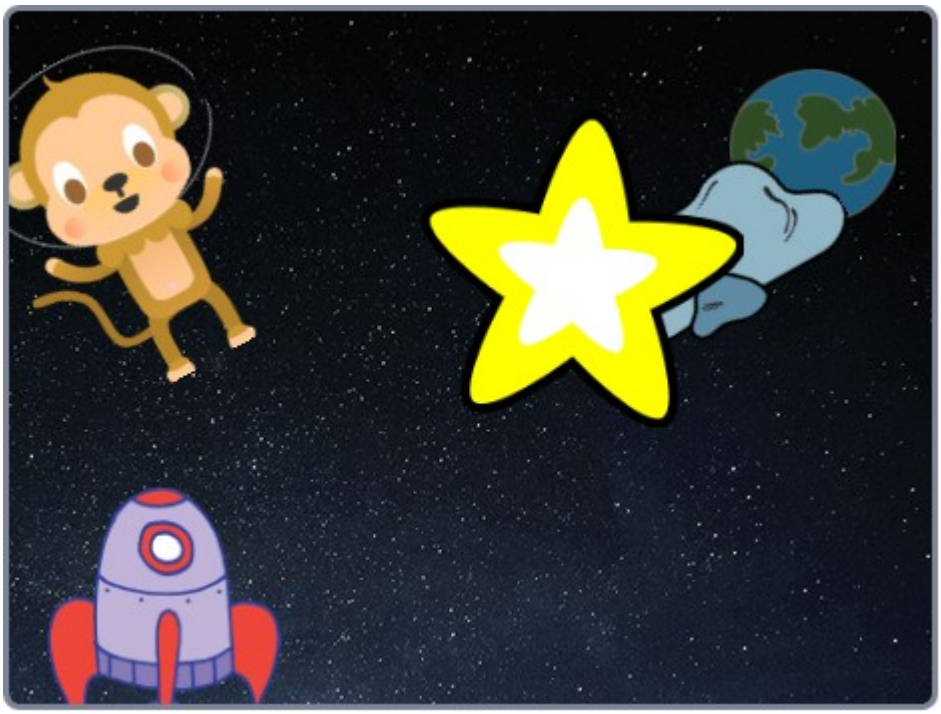
Drag the unused blocks to the blocks palette to delete them.

Click *File/Save* to save your project. 8 Marks

Lost in Space

You are going to learn how to program your own animation!

What you will make



- Use a repeat loop to animate a sprite in Scratch
- Use a forever loop to repeat an animation indefinitely
- Understand that loops can be nested within each other

Note: The work for this is in a separate document called **lost in space**

Creating a game.

Defining the game

The object of this task is to create a game that involves catching a falling apple in a basket.

Requirements:

1. A suitable background.

2. A falling apple, preferably more than one

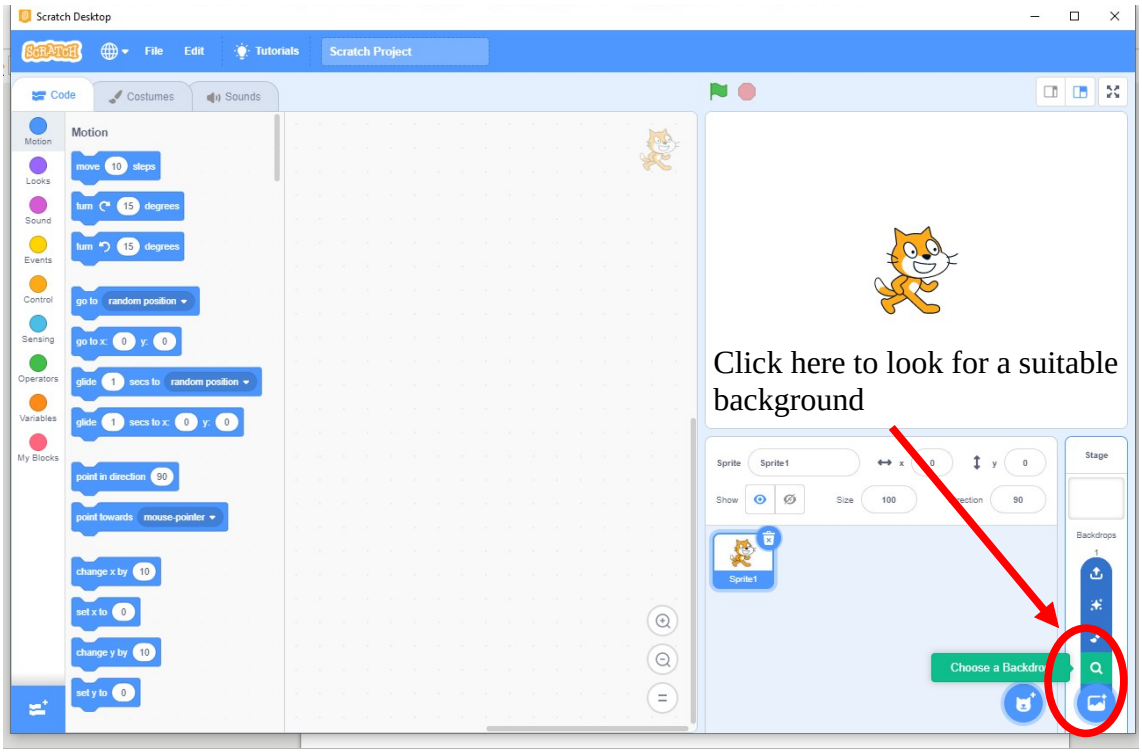
3. Something to catch the apple(s)

4. Something to keep score of the apples dropped and the number caught.

Lets look at these components separately.

1. A suitable background.

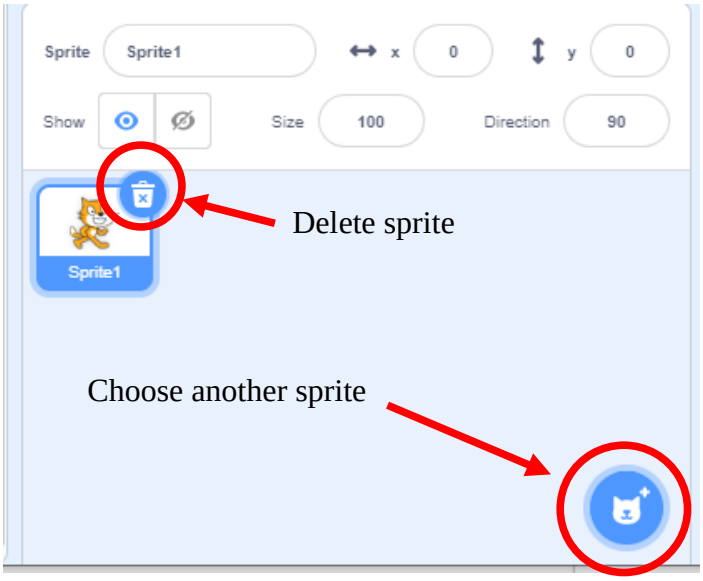
Any background with a tree might be useful. Maybe an orchard? We will need to search the backgrounds in scratch and come up with something useful. Alternatively if you have a flare for drawing you might consider drawing your own background.



As you will see there are many backgrounds to choose from. Some which might be suitable are Farm, Forest, Jungle, Pathway, Tree and Woods.

I chose tree

Of course we don't need the cat sprite so we can get rid of that simply by clicking the waste bin at the top right of the sprite. We can then choose another sprite



But that comes later. For now we have completed the first of our objectives.

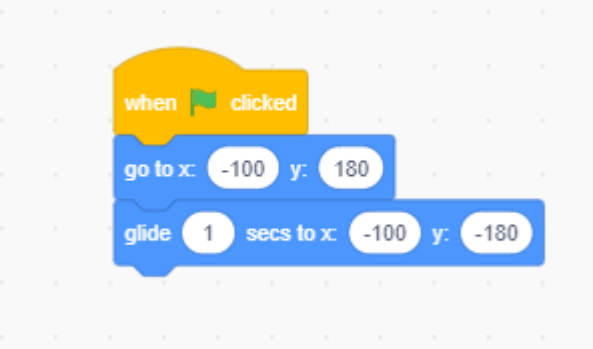
2. A falling apple, preferably more than one

There are several ways of doing this part. We could choose multiple apple sprites but that would be difficult to manage. Instead I am going to select one apple sprite and make it fall from the top of the stage to the bottom several times from random positions.

First lets get an apple to fall

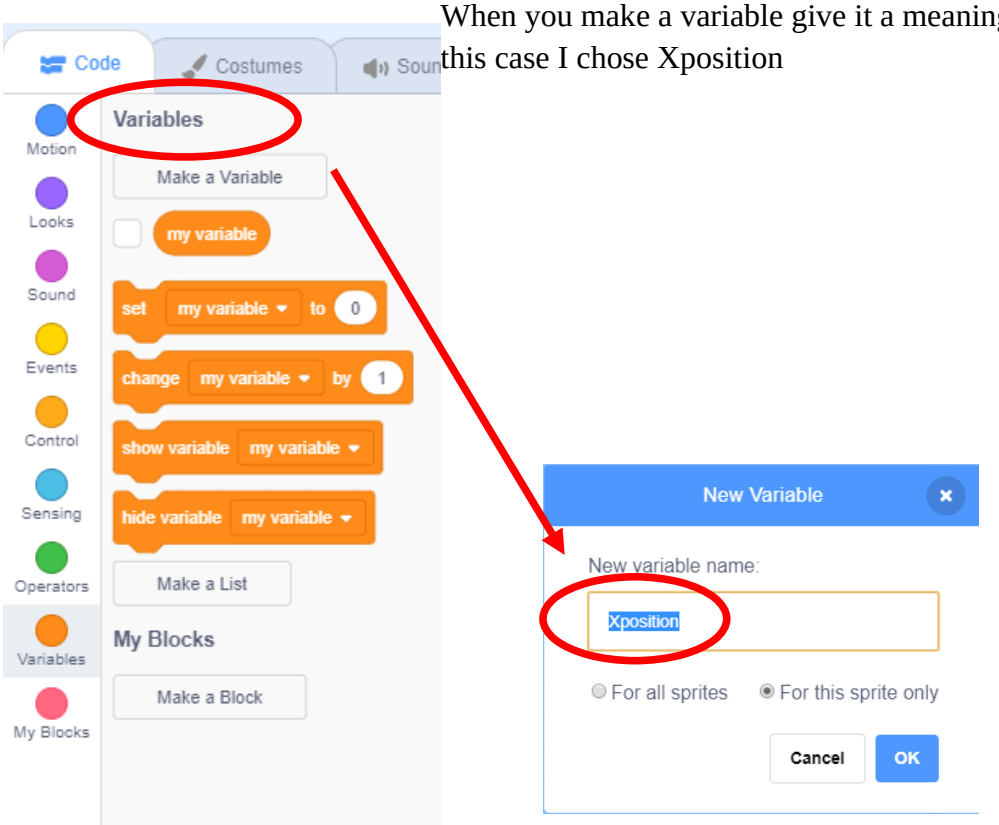
Task: Select the apple sprite from the

This is the code to do that



This is OK but the apple will fall from the same position each time it falls. Fortunately there is a Random function in scratch and we can use this to change the X coordinate each time the apple falls.

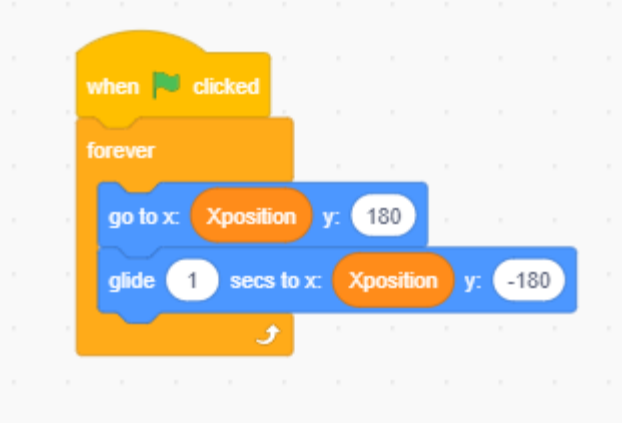
To do this we must first create a variable, that I will call Xposition.



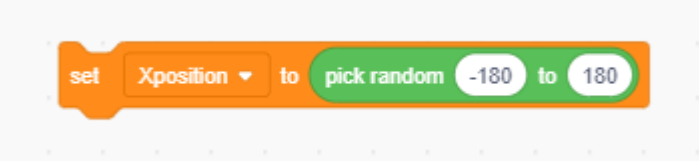
When you make a variable give it a meaningful name. In this case I chose Xposition

We can use the variable Xposition to place the apple at some position along the X axis but its still not random. This piece of code will cause the apple to fall but we haven't given Xposition a value so it takes 0 and falls from the centre of the stage each time.

Task: Try it out for yourself



The piece of code that fixes this is



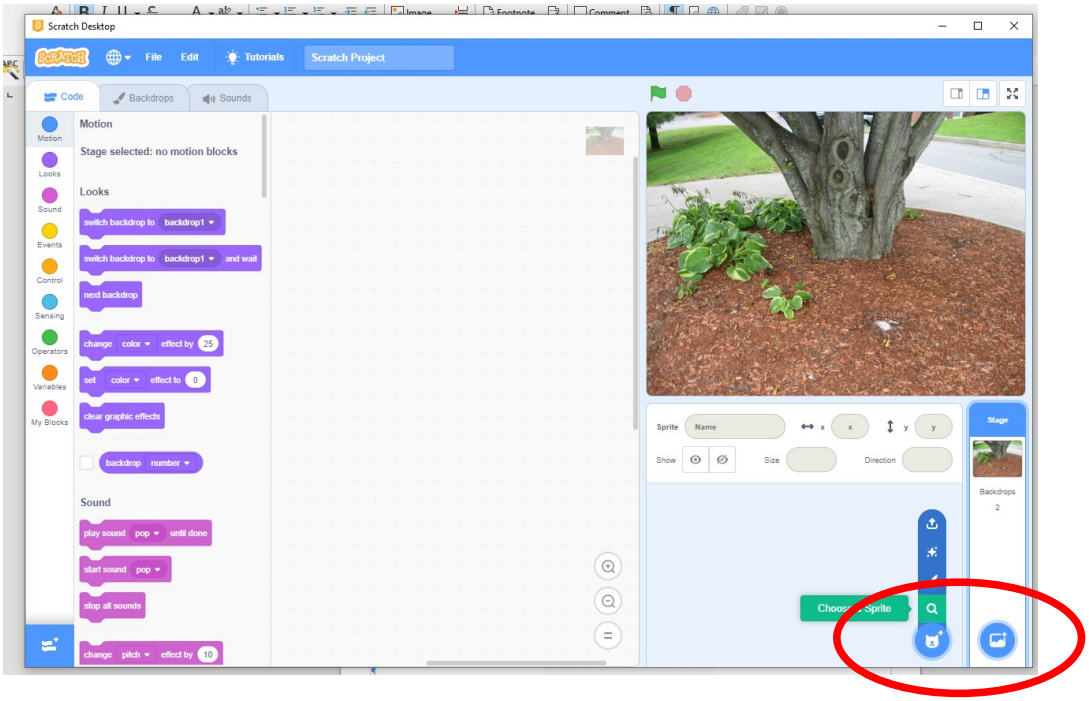
pick random block is in the **Operators** block list.

Task: Put that last block into the code in the right place.

That's a big chunk of the programming done, but not the hardest.
Now it's time to move on to the Basket

3. Something to catch the apple(s)

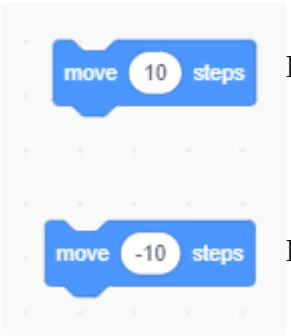
We need a suitable sprite so we use the select a sprite tool



There are no baskets to choose from but there is a bowl

Once we have selected the sprite We can add some code to Place it in the bottom centre of the stage and make it move back and forth i.e. to the left or to the right. We can control the movement with any keys we choose but the most obvious would be the left and right arrow keys. If the game is to be played on a device without the arrow keys could you could choose S and D or any other keyboard letters.

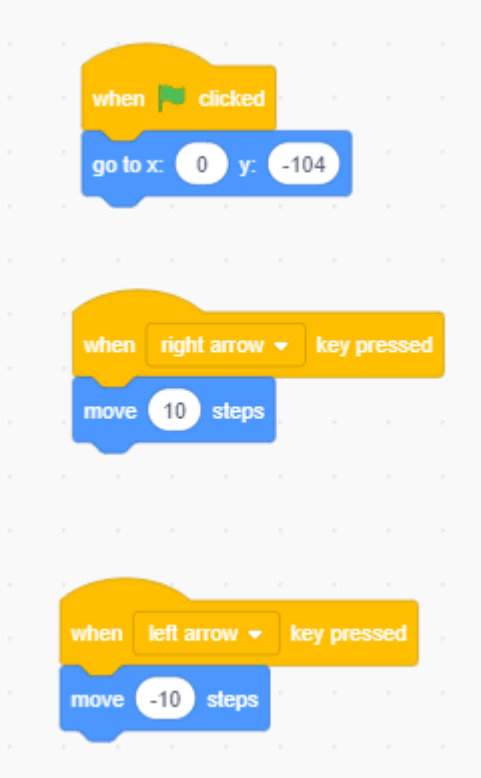
Whichever keys you choose moving to the right can be achieved with the move block if it has a positive number in it. Similarly a move to the right would need a negative number in the move block.



Moves to the right

Moves to the left

So putting the code together



Places the sprite in the bottom centre of the stage

Moves the sprite to the right

Moves the sprite to the left

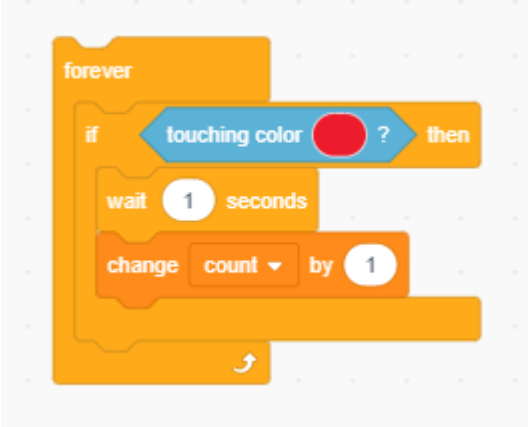
Task: Enter this code into scratch but make sure you select the basket sprite first

How do we know when an apple has hit the basket and how can we count this?

Hitting the basket

We can use a block, **touching colour,** from the Sensing list along with an **If block** from the control list. To recognise when an apple hits the basket.

Counting the hits
We must create a variable with a sensible name, say count and add 1 to this count each time the basket is hit.

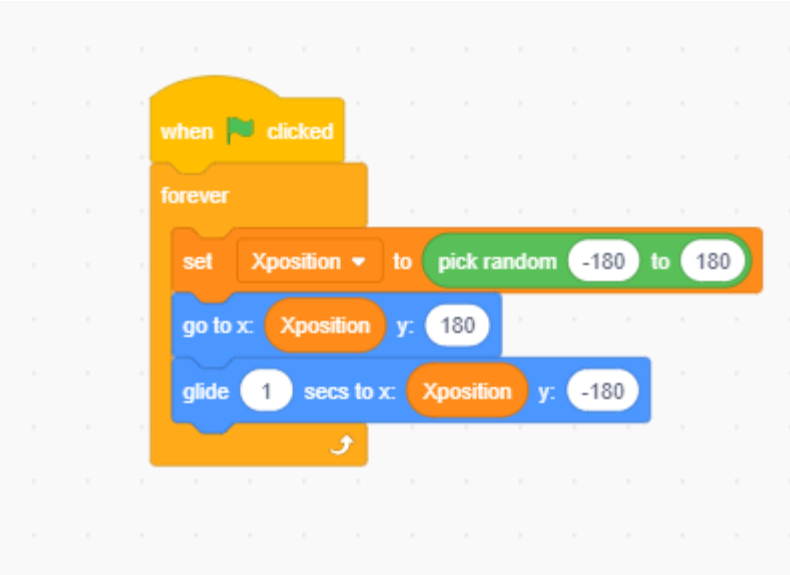


Task: Why is the wait block there?

Task: How can we make sure the count is zero at the start?

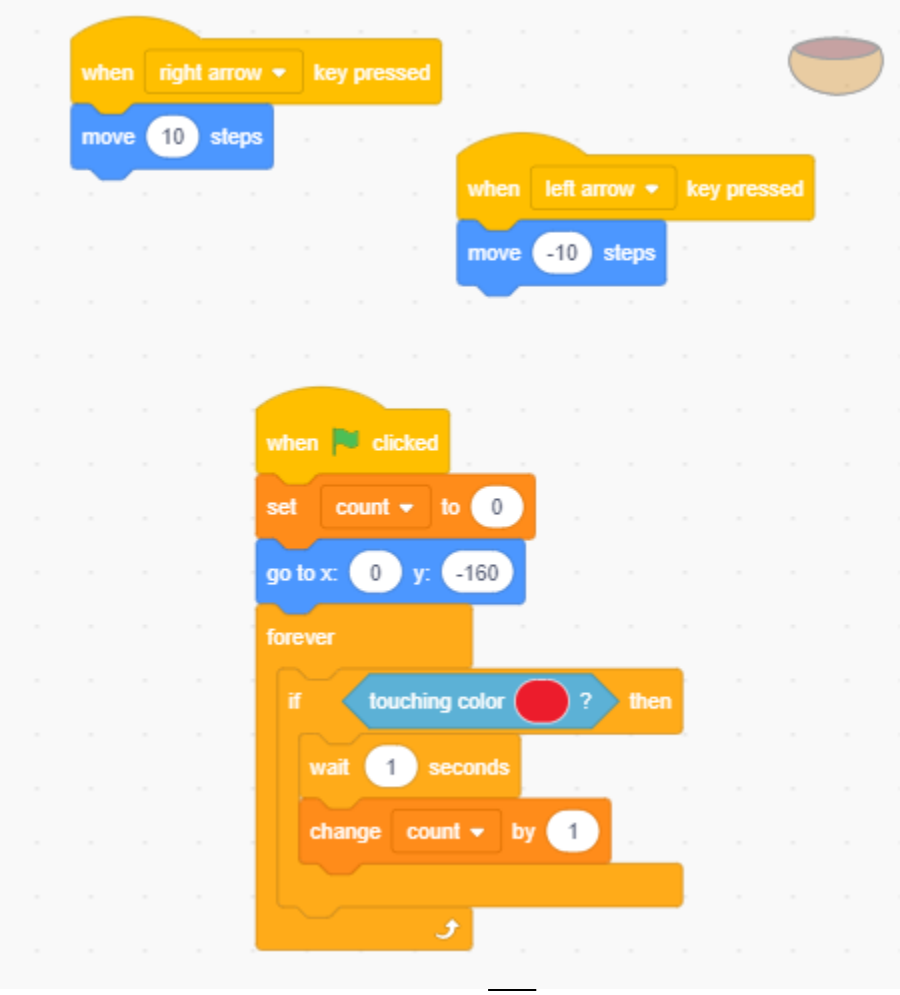
Solutions

Apple drop code



Task: Change this code to count the apples dropped. Hint create a variable dropped

Basket Code



Apple dropped code with apple counter.