

Computer Science

A black laptop is shown from a front-facing perspective, open. The screen displays a learning objective in large, bold, black text on a light blue background. The laptop has a keyboard, a trackpad, and three circular ports on the left side of the bottom bezel. The background of the entire image is a solid blue color.

**LO: I am learning
to create my own
game using video
sensing.**

Creating Plan, make, assess

Creating video sensing game



<https://www.youtube.com/watch?v=8vHEqVdWn08>

Click on the link to see what a video sensing game is. I will show you Mr Viney's game when we are back in class! The programme is a slightly updated version but for now this video will give you an idea.



Creating Plan, make, assess

What do you want your programme to do?

Creating
Plan, make,
assess

Draw and label what your programme will look like (how will the sprites move, what will they say, what sounds will be used.

Tinkering
Try things out

Edit your design with green pen to show how you improved your design.



Creating Plan, make, assess

Example

Creating
Plan, make,
assess

Draw and label what your programme will look like (how will the sprites move, what will they say, what sounds will be used.

The purpose of the game is to keep the ball in the air.
Use motion when there is contact with the sprite

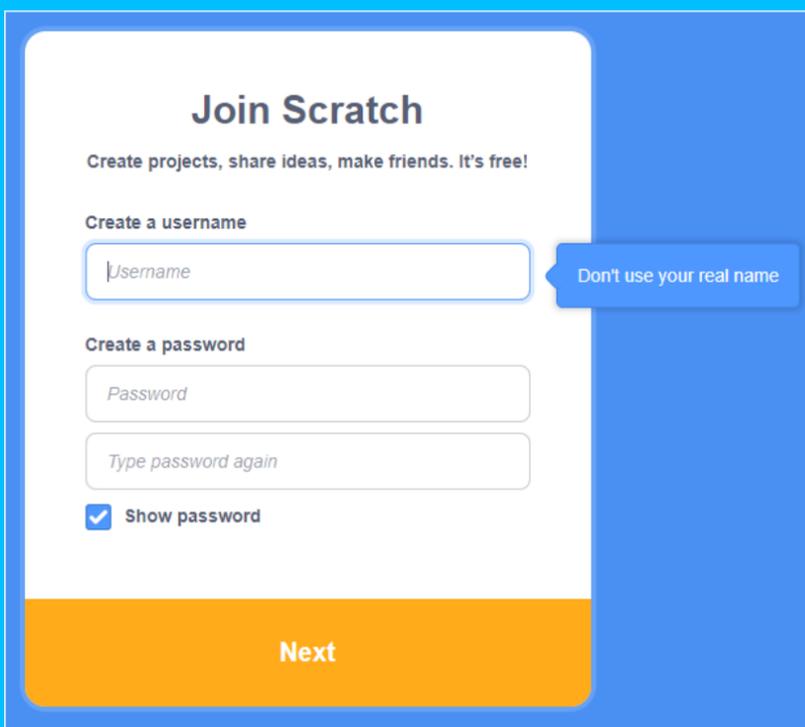
Tinkering
Try things out

Edit your design with green pen to show how you improved your design.

Computer Science

Creating an account and logging in

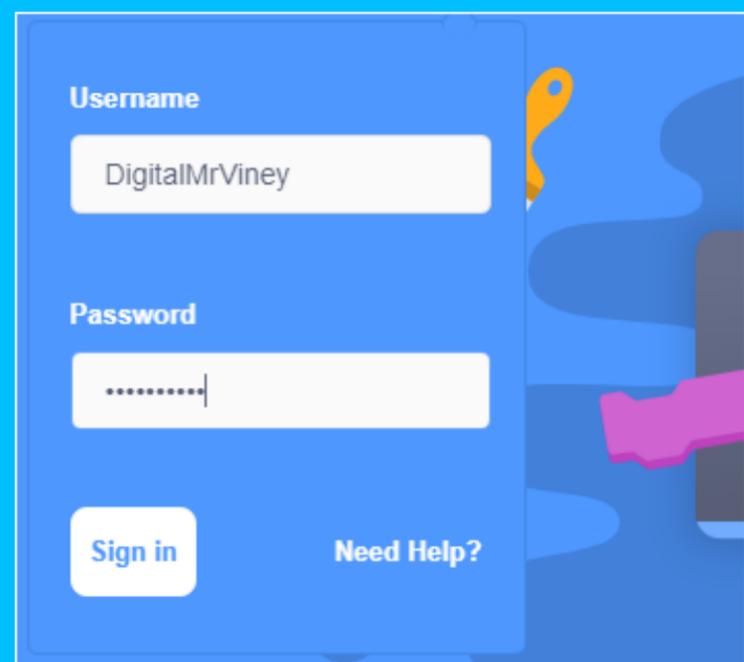
<https://scratch.mit.edu/join>



The screenshot shows the 'Join Scratch' registration page. It features a white form on a blue background. The form includes a title 'Join Scratch', a sub-header 'Create projects, share ideas, make friends. It's free!', and two main sections: 'Create a username' and 'Create a password'. The 'Create a username' section has a text input field with the placeholder 'Username' and a blue callout box that says 'Don't use your real name'. The 'Create a password' section has two text input fields: the first with the placeholder 'Password' and the second with the placeholder 'Type password again'. Below these fields is a checkbox labeled 'Show password' which is checked. At the bottom of the form is a large orange button labeled 'Next'.

1. Create an account by making up a username and password

2. You will need to create a username and password that you can remember!



The screenshot shows the Scratch login page. It features a white form on a blue background. The form includes two text input fields: 'Username' with the value 'DigitalMrViney' and 'Password' with a masked password represented by dots. Below the password field is a 'Sign in' button and a 'Need Help?' link. The background of the page shows a stylized illustration of a person's hand holding a smartphone.

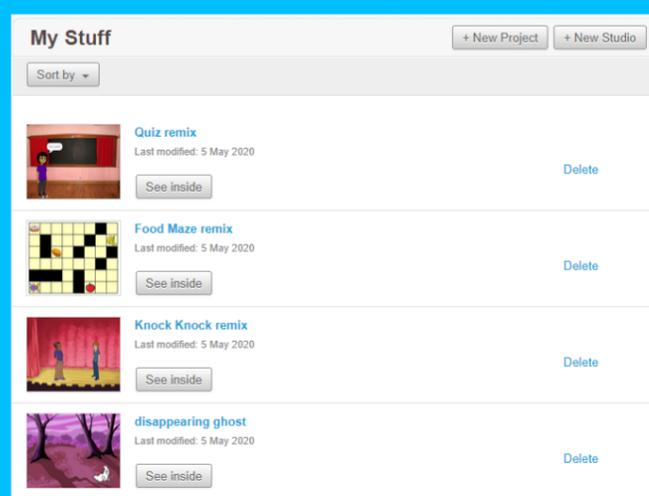
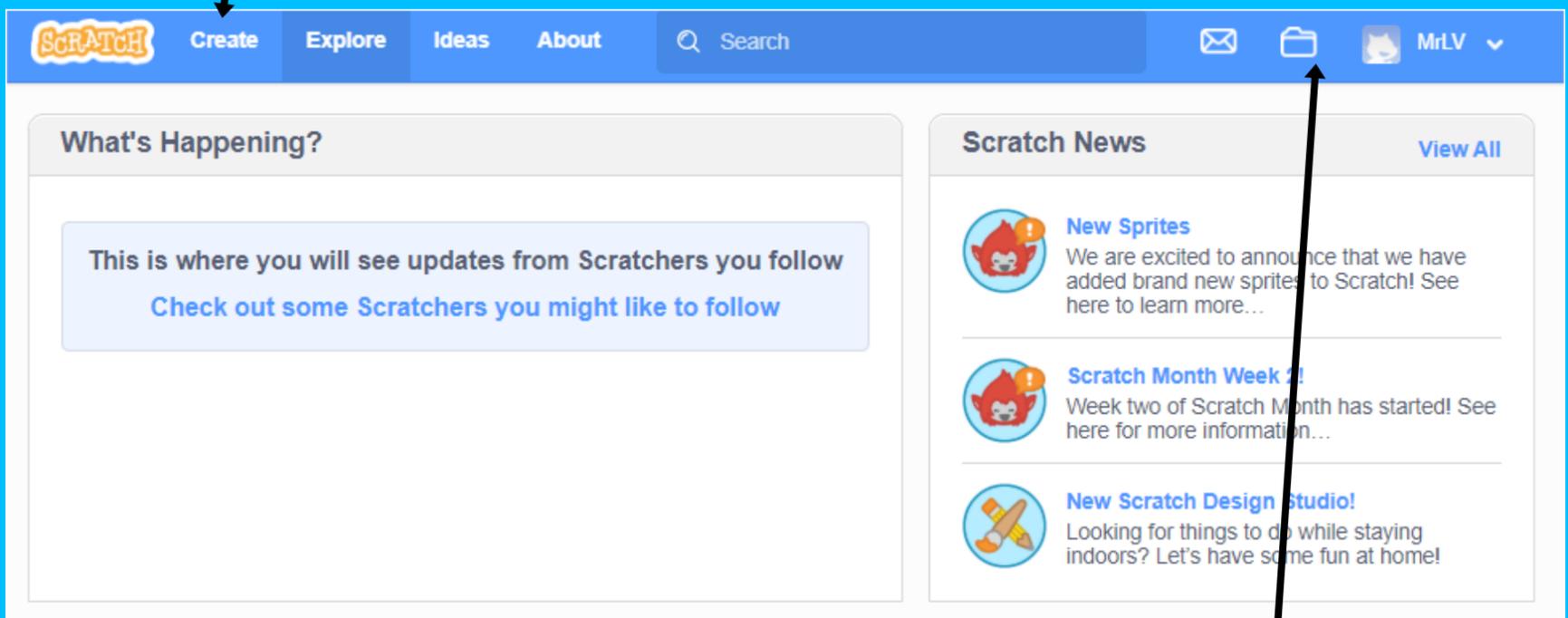
E-Safety: Make sure you use a strong password and do not use your real name.

Introduction
to the platform

Computer Science

Where to go to create your own
programmes

Your profile and log out



Where your work is saved

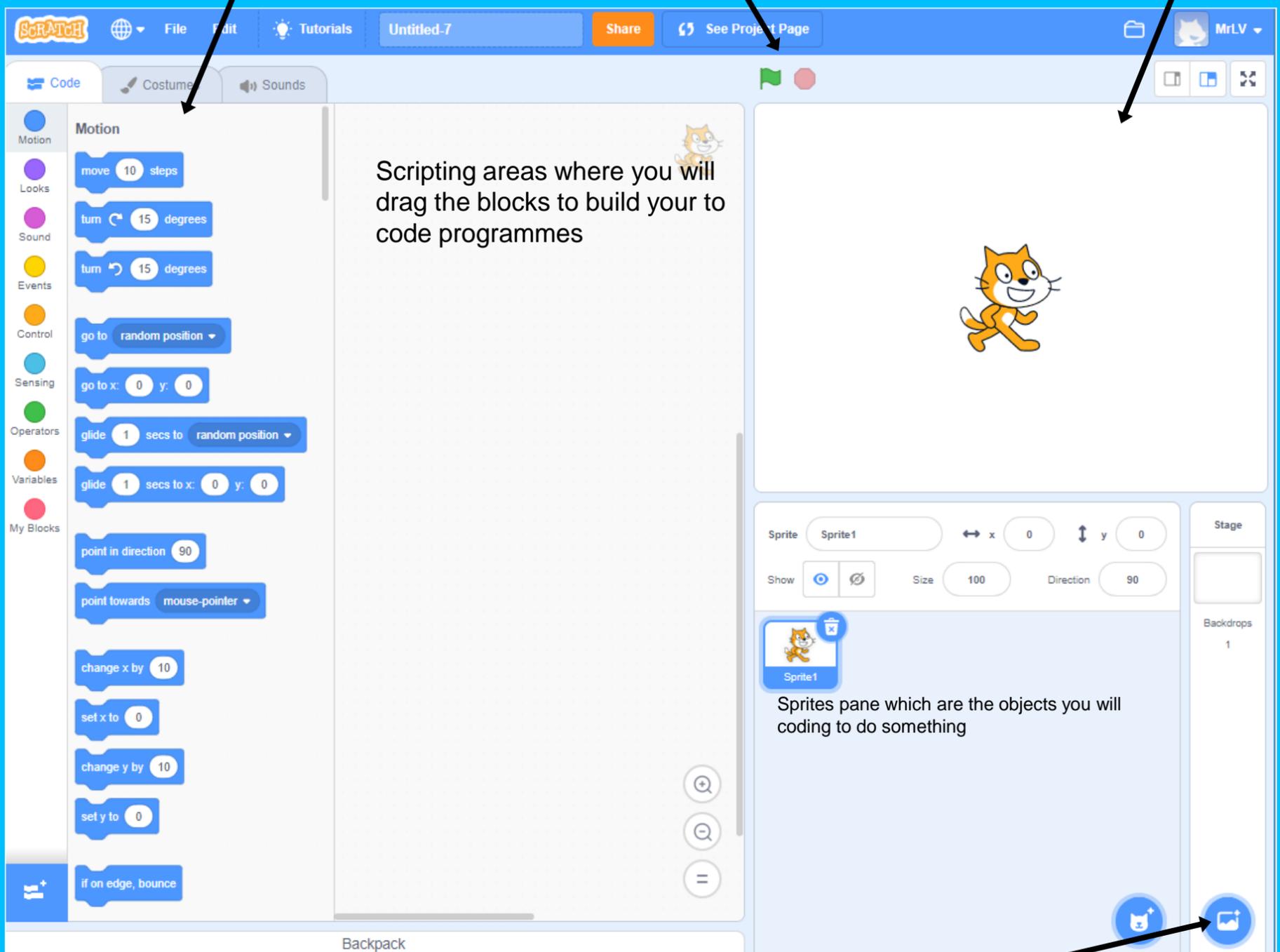
Create

Computer Science

These are command BLOCKS which are grouped to make them easy to find.

How to start and stop your code

Staging area where your coding will come to life!

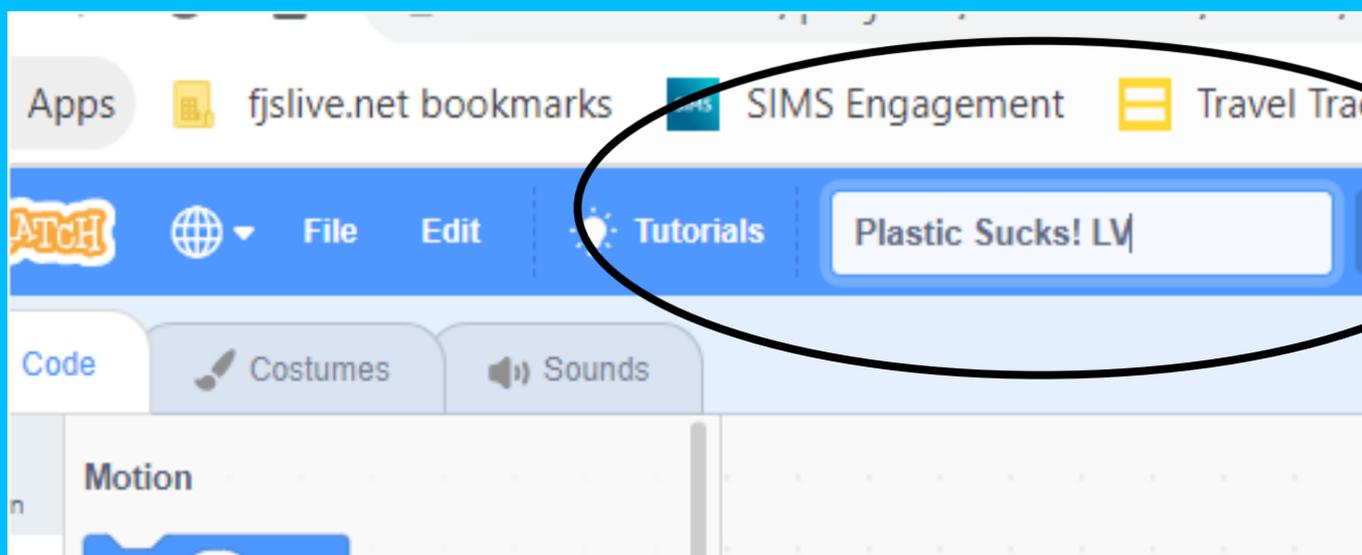


Make your code look great with a background (backdrop)

Name your  creation

Computer Science

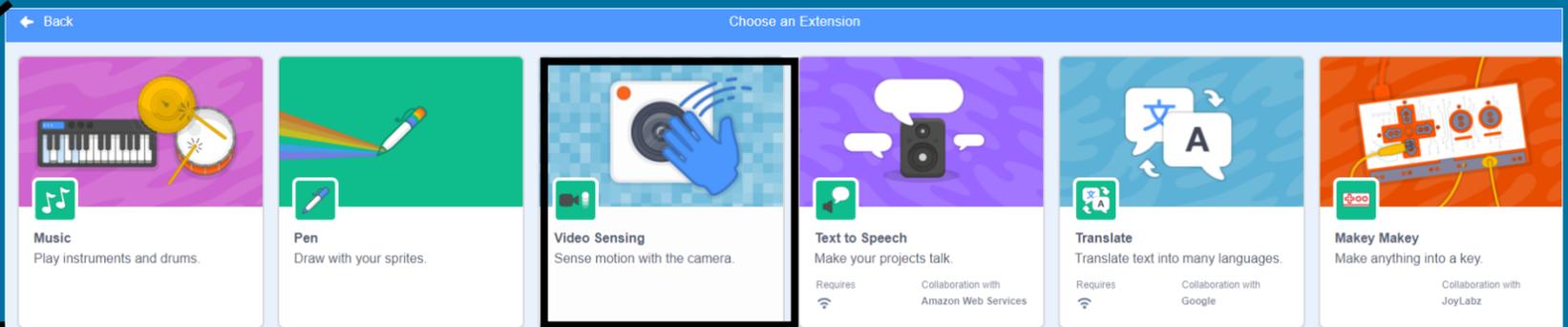
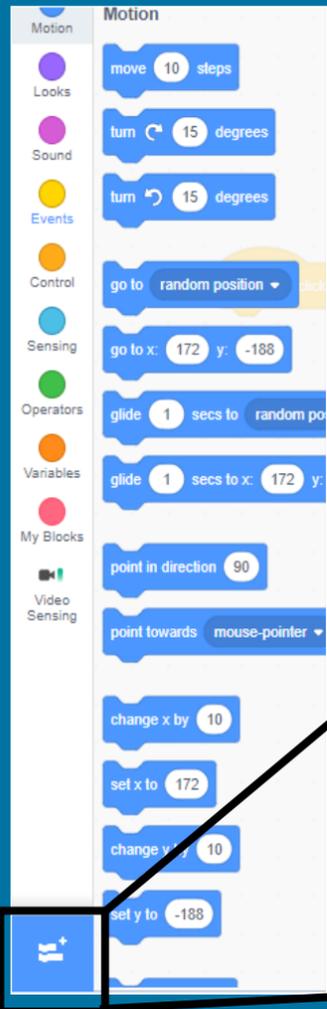
Change the name of the project



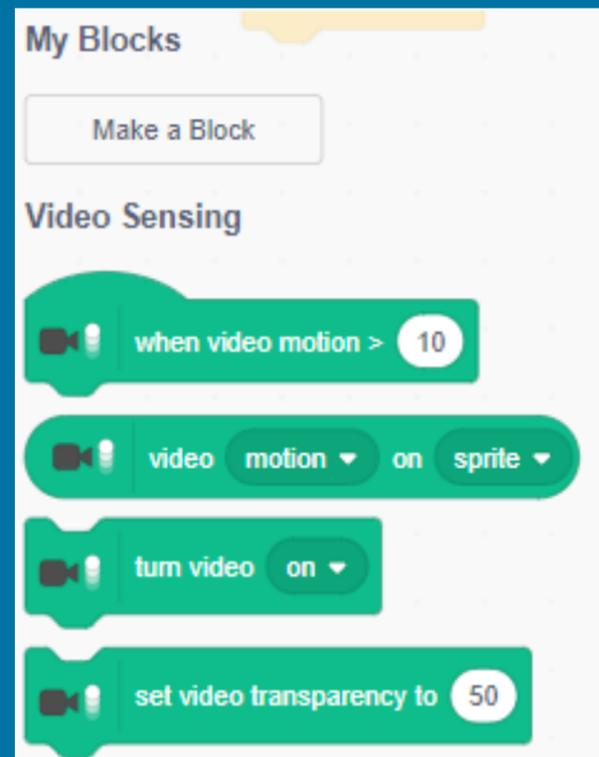
E-Safety: Make sure you do not use your whole name, just initials please.

Creating Plan, make, assess

1. Select the "video sensing" extension.

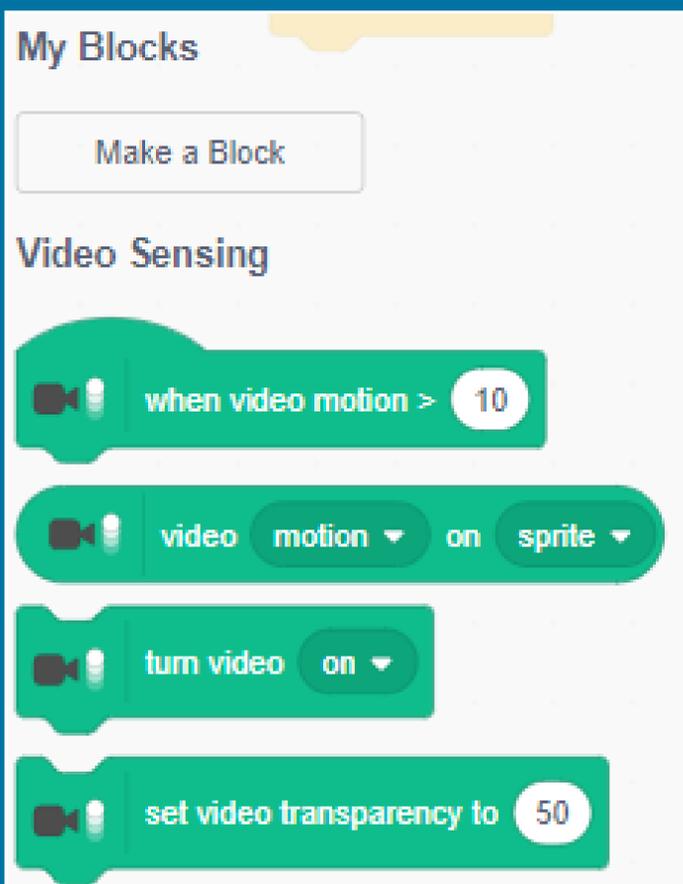


2. "video sensing" is added to the bottom of the command blocks



Creating Plan, make, assess

Video motion: This sets how sensitive the movements are. It goes from 0 where there is no movement on camera to 100 where the video senses a lot of movement.



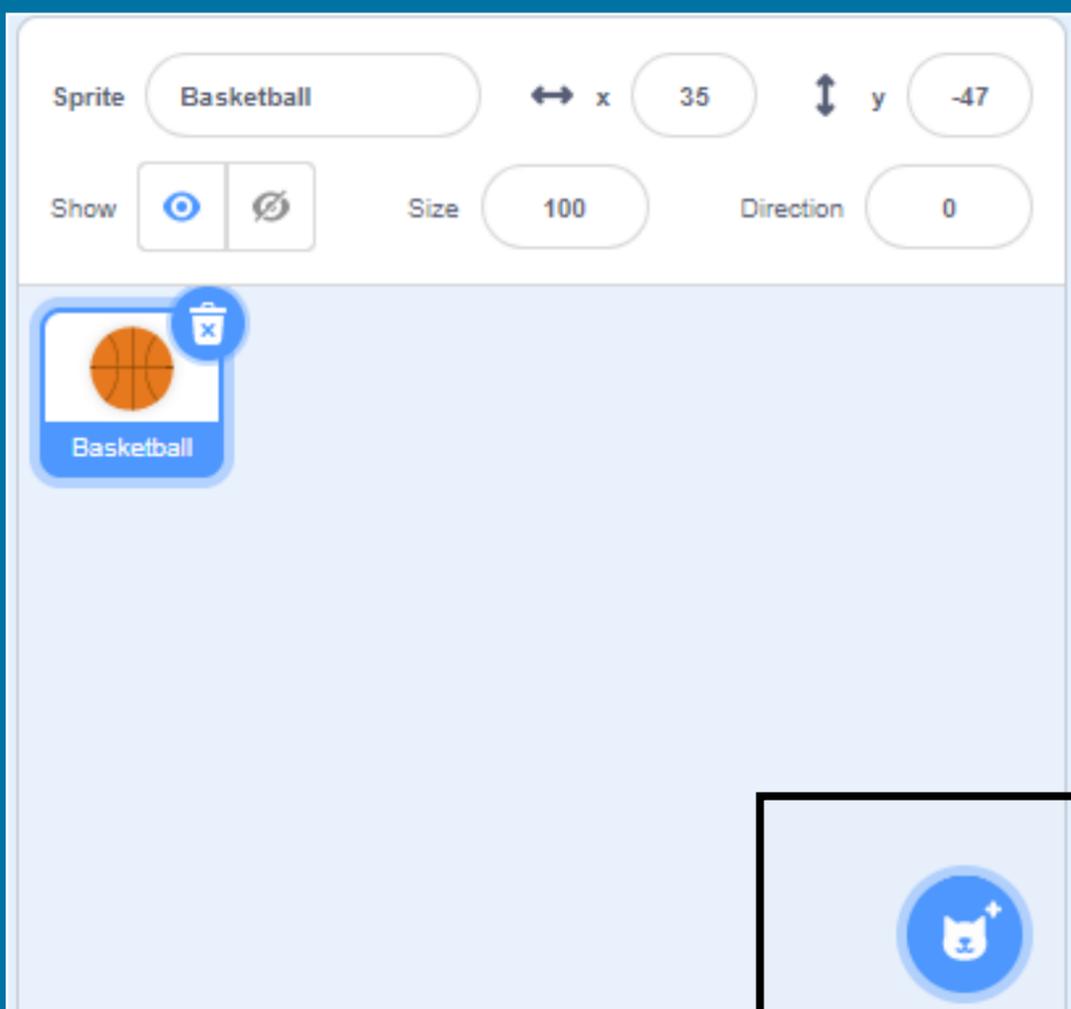
Video motion on sprite: Sense the motion on the sprite so when the sprite is touched it moves.

Turn video on/ off: Command to turn the webcam on/off

Transparency: The backdrop is set to white (0 = 100% transparent, 100 = white backdrop)

Creating Plan, make, assess

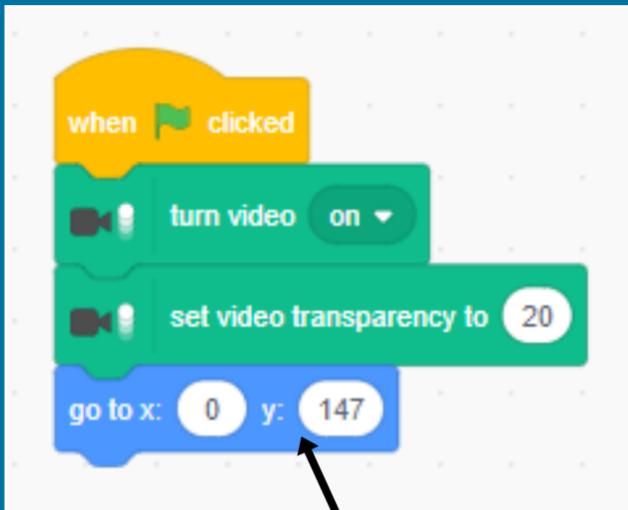
1. Add the sprites you need (you may want to create your own better sprites!). Place them on the backdrop.



Use the sprite search to select a basketball

Turn video sensing on

Creating Plan, make, assess



1). Use the video sensing blocks to turn the video on.

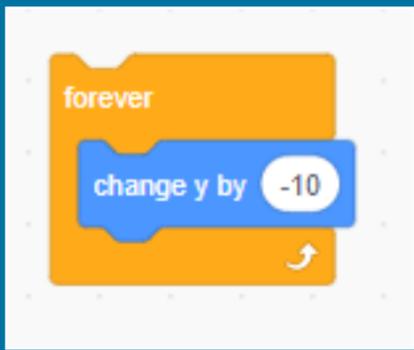
2). Set the video transparency to 20 (tinker with this to see how it changes the video view).

3). Position your sprite on the back drop.

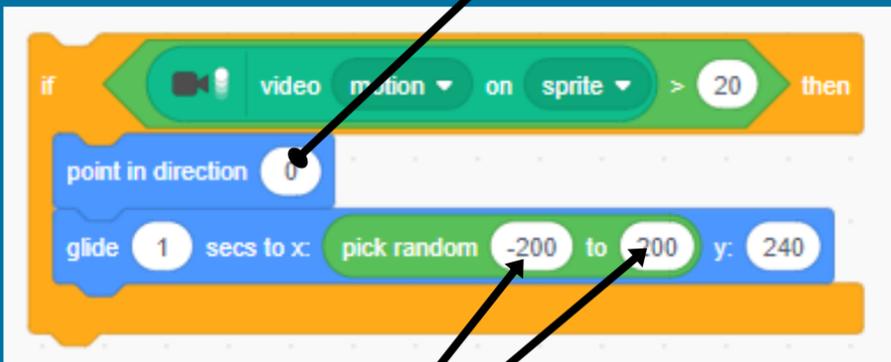


Creating Plan, make, assess

Describe what the code is doing.



Makes the sprite fall 10 steps (y axis) continually (for ever)



When some motion (level 20) is detected on the sprite it will move in direction 0 (upwards).

Glide to the top (y axis) to a random place on the x axis (horizontal).

The x and y axis goes from -240 to 240



Creating Plan, make, assess

Put the code together like this...

```
when clicked
  turn video on
  set video transparency to 20
  go to x: 0 y: 147
  forever
    change y by -10
    if video motion on sprite > 20 then
      point in direction 0
      glide 1 secs to x: pick random -200 to 200 y: 240
```

The image shows a Scratch script on a white grid background. The script starts with a yellow 'when clicked' block. This is followed by two green video blocks: 'turn video on' and 'set video transparency to 20'. Next is a blue 'go to x: 0 y: 147' block. An orange 'forever' loop contains three blocks: a blue 'change y by -10' block, a green 'if video motion on sprite > 20 then' block, and a blue 'glide 1 secs to x: pick random -200 to 200 y: 240' block. The 'if' block contains a blue 'point in direction 0' block. The 'forever' loop ends with a white arrow icon.

Play your keepy uppy game!

Tinkering

Try things out

Let's review our code.
What do you think?

```
when clicked
  turn video on
  set video transparency to 20
  go to x: 0 y: 147
  forever
    change y by -10
    if video motion on sprite > 20 then
      point in direction 0
      glide 1 secs to x: pick random -200 to 200 y: 240
```

What can you change to see how the
game could be improved?



Tinkering

Try things out

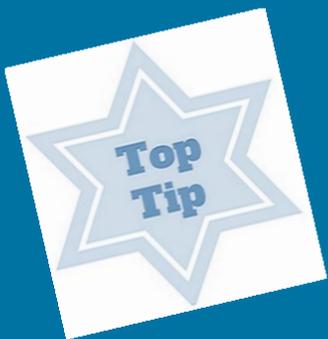
How can you improve the game?



Add a bounce sound
Change colour



Add more sprites using the same code



Make sure you test your code EVERY TIME you make a change!
Use the stop button to reset the programme



LEVEL
UP!

Tinkering

Try things out



Add a bounce sound
Change colour

```
when clicked
  turn video on
  set video transparency to 20
  go to x: 0 y: 147
  forever
    change y by -10
    if video motion on sprite > 20 then
      start sound basketball bounce
      change color effect by 25
      point in direction 0
      glide 1 secs to x: pick random -200 to 200 y: 240
```

Add the start sound

Change colour of sprite

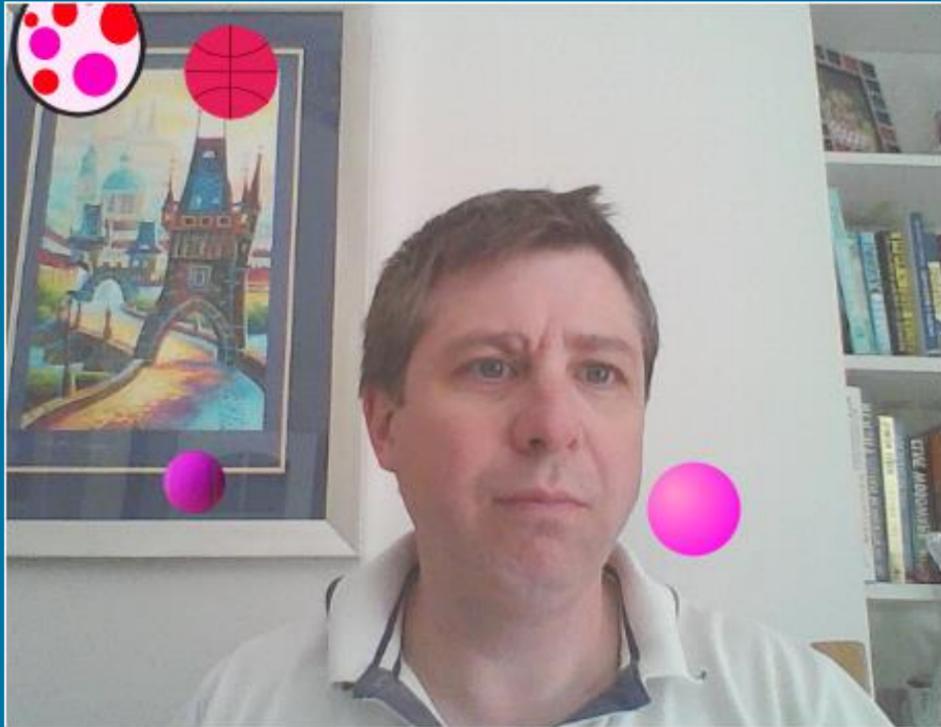
LEVEL
UP!

Tinkering

Try things out

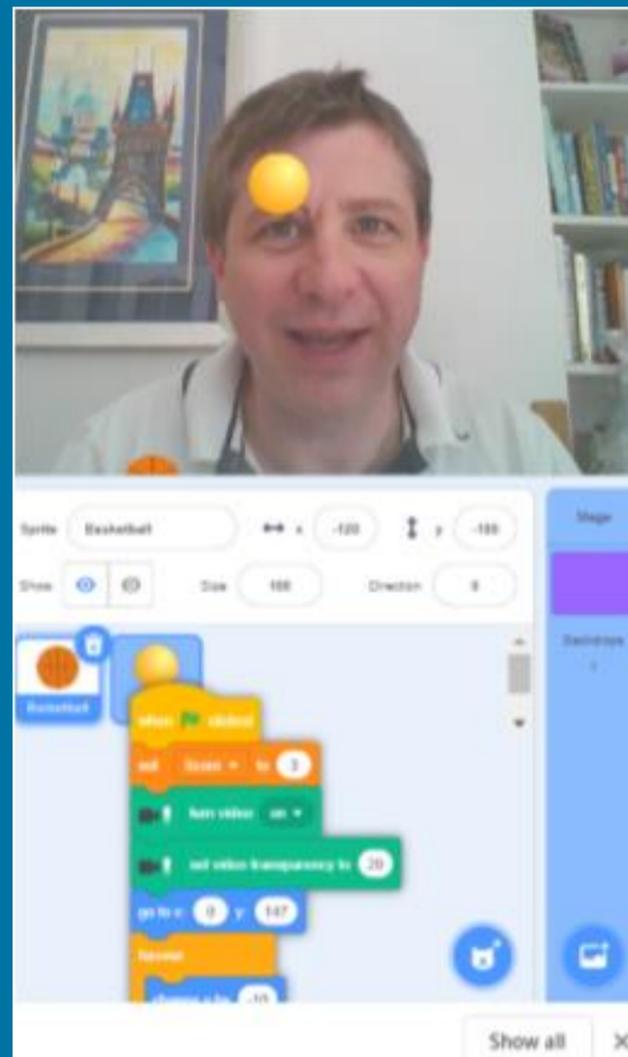


Add more sprites using the same code



1). Add new sprites

2). Drag and drop the code on to the sprite in the sprite area





Tinkering

Try things out

Use green pen to annotate your design with the new ideas for your programme.

Creating
Plan, make
assess

Draw and label what your programme will look like (how will the sprites move, what will they say, what sounds will be used).

The purpose of the game is to keep the ball in the air.
Use motion when there is contact with the sprite.
Use a score variable.
Use sound and change the colour of the sprite to make the game more interesting.

Tinkering
Try things out

Edit your design with green pen to show how you improved your design.

De-bugging

Identify and correct mistakes, evaluate

What have you had to de-bug today?

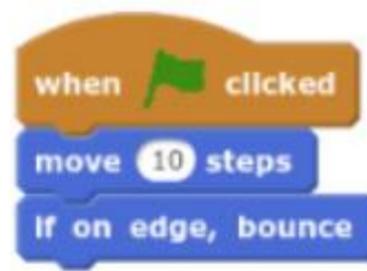
What skills do you need to "de-bug"?

De-bugging

Identify and correct mistakes, evaluate

What's wrong with this code and how would you correct it?

Here is some code for a ball sprite:



Why will this code not result in the ball constantly bouncing from one side of the screen to the other?



- The 'if on edge, bounce' block is not needed.
- The code needs to be within a repeat block.
- The code needs to be within a forever block.
- It needs another 'move 10 steps' block after the 'if on edge bounce' block.

The code needs a forever block because the ball would only move 10 steps then stop.

Saving your project

Computer Science

You do not need to save your project but if you are able to take a screen shot of your game, I would like to see it!