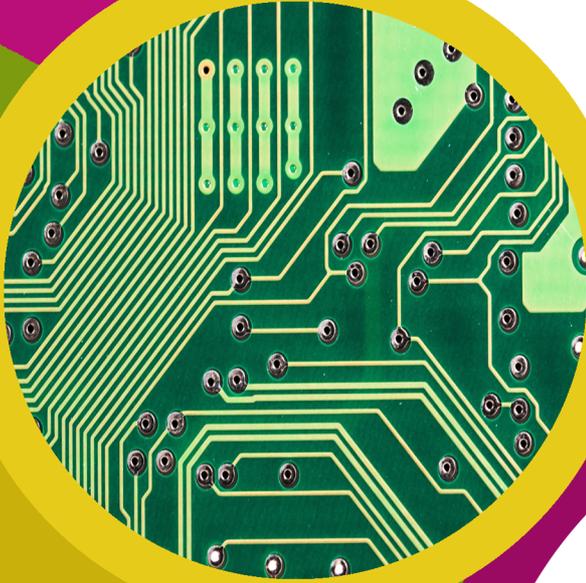


technocamps

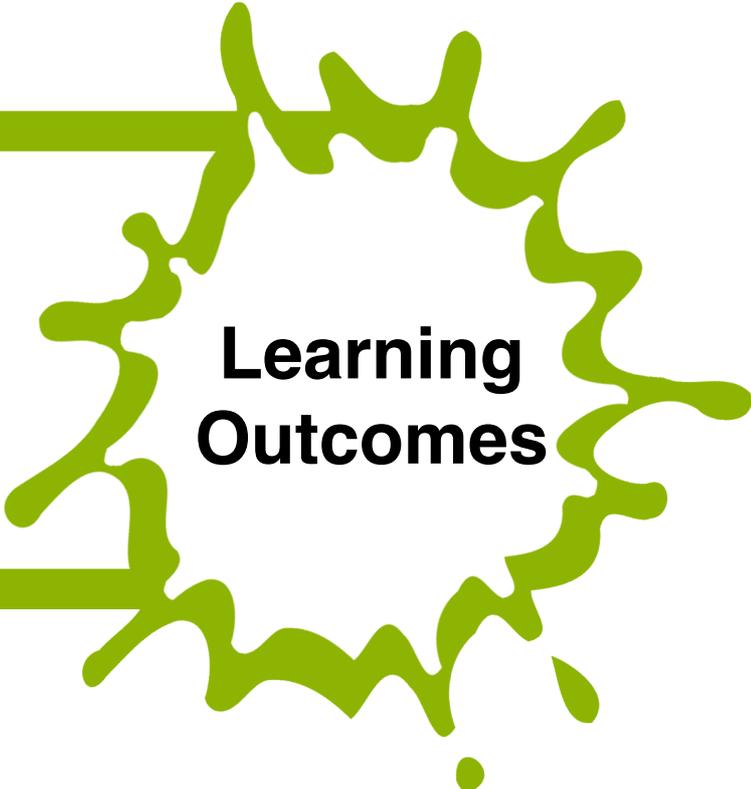
The Weird and Wonderful World Workbook



Overview

In the workshop today we will be looking at wonderful and mind boggling experiments. These experiments are designed to test your ability to think outside the box, problem solving and teamwork. We will be looking at three different experiments, which will test your understanding of the world around you. We will also develop a

1. To learn three weird and wonderful facts about the world around us.
2. To learn the basic functions in Scratch.
3. To model these activities using Scratch.



Learning Outcomes

Attendee Prerequisite

1. No experience required.

Thinking Outside the Box

When we go about our lives and we see things in the world around us, we anticipate what will happen next.

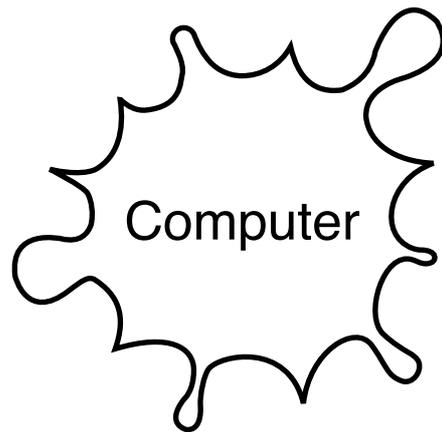
We usually anticipate the obvious e.g. what do we first anticipate will happen when the wind blows and there are leaves on the floor?

What if the obvious we anticipated does not happen e.g. what if the leaves instead of blowing away stays and flows in the same spot?

How do we process these things in our minds, is there an explanation for this or is it unexplainable?

What Does a Computer Look Like?

Types of Computer



Defying Gravity

What do you think will happen if we place a ping pong ball in a cup and turn it upside down?

Why do you think that is?

What happens if we attach a string to the bottom of a cup, and swing it around while the ball is inside?

Under Pressure

What do you think will happen if we place a piece of paper just below our lips and we blow air over the piece of paper?



Bernoulli's Principle

What do you think will happen when we place a ping-pong ball on the end of a plastic straw and blow through it?



Why do you think that is?

What is Scratch

In this workshop you will be using a software called Scratch, a drag and drop style software you can use to build your own games easily. You can learn fundamental programming principles without having to worry about any spelling or typing errors and enjoy creating and playing your very own game.

Scratch has six main sections. Firstly starting from the top left you have a list of categories to choose from. As you select each one you can see in the box below that different instructions appear (colour coordinated). These can be used to create your game by dragging them across to the right where the scripts can be used for your game.

On the right of the screen you can see the white box with the Scratch character inside, this is your game. As you begin developing your game, you can start and stop it using the green flag and the red circle buttons above.

Motion

Motion blocks deal with the movement of sprites.

Looks

Looks blocks are related to the appearance of sprites and the

Sound

Sound blocks are related to the sounds our game can make.

Events

Event blocks trigger things to happen in our game. For example, when the green flag is clicked, the game will start.

Control

Control blocks organise how the program will be run. This is done through loops (repeating code multiple times) and if

Sensing

Sensing blocks are used to detect when a condition has been met. An example is checking whether a sprite is touching another

Operators

Operator blocks consist of many mathematical functions such as addition and checking if something is equal.

Variables

Variable blocks are used to store information such as the answers to our questions.

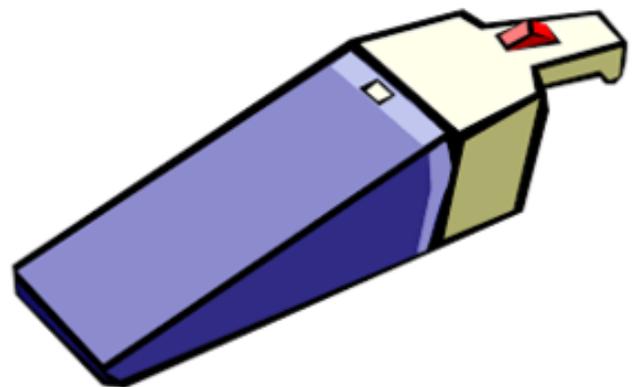
Creating Sprites

Firstly, we need to add our vacuum to the screen. This will be provided to you from the Technocamps website.

Secondly, we need to add another sprite, the ball. This is also located in the given folder.

Hints:

- To add characters from a folder look for the 'import' button.
- Once found, click it and search for the name of the folder.

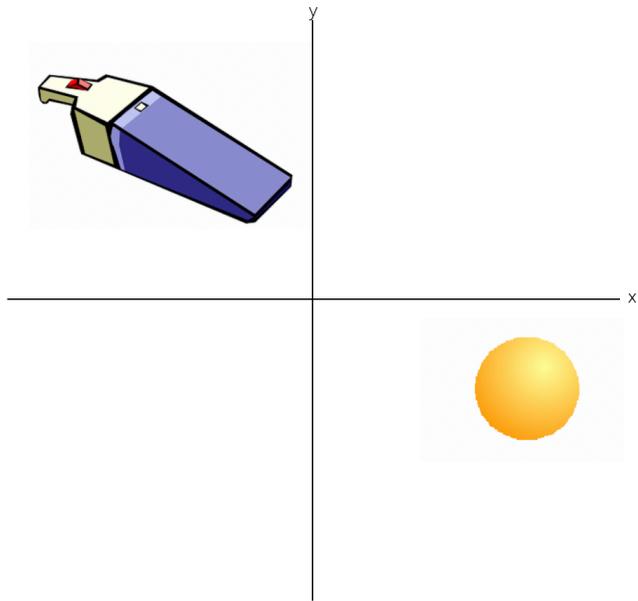


Using the “go to” Block

Firstly, place the vacuum sprite on the top left corner. Next, drag a “go to” block into the sprites’s script and it should automatically enter in the x and y coordinates of where the sprite currently is.

Hints:

- This block should be attached to a “when green flag clicked” block from the Control category
- Below the “go to” block add a “show” block from the looks category.
- Do the exact same for the Ball sprite but of course for a different location, preferably somewhere at a distance from the vacuum.



Co-ordinates

Vacuum

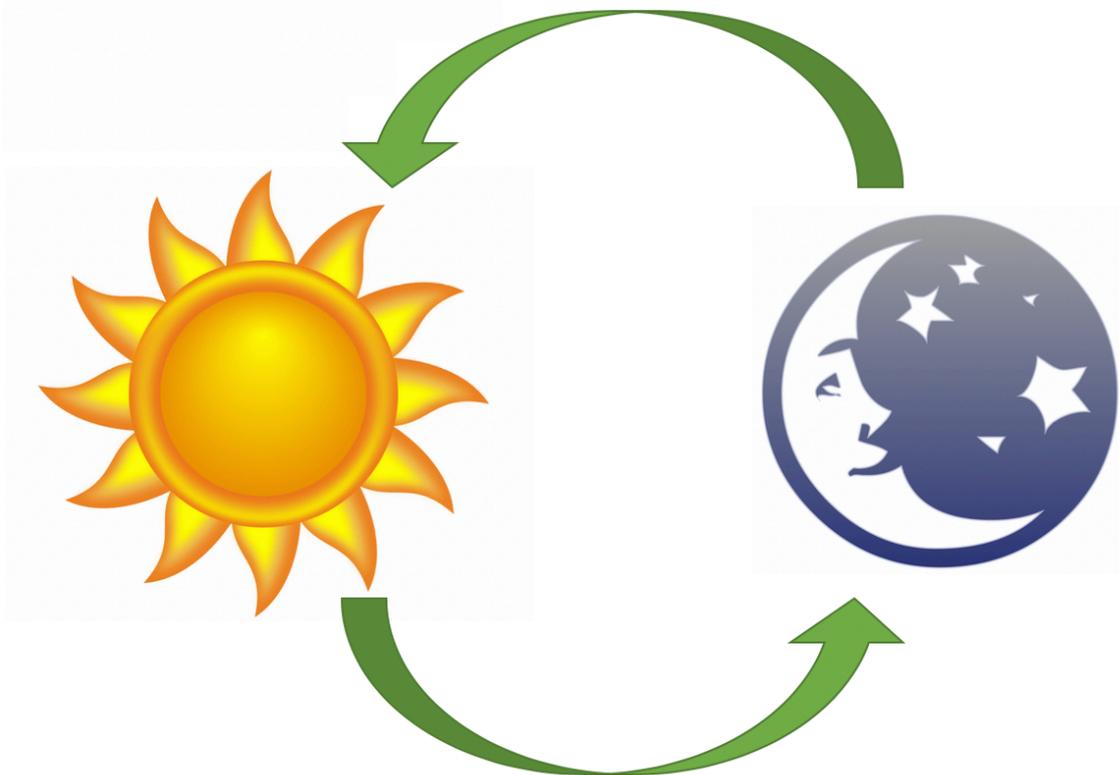
x = _____

y = _____

Forever Point Towards and Move 1

Now we are ready for the vacuum to forever follow and point towards the Ball sprite. To do this we will need three different blocks:

1. The “forever” block found in control.
2. The “point towards” block found in motion.
3. The “move” block found in motion.

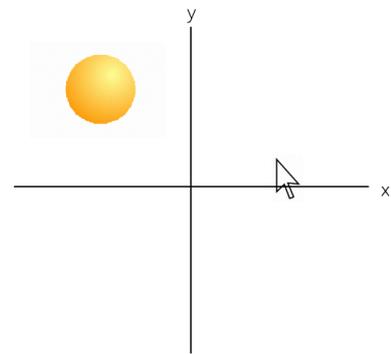


Forever Point Towards and Move 2

Very similar to the previous part, we now need to add code to the scripts in the Ball sprite. To do this, click on the Ball sprite and then go on the “Scripts” tab. Using the same blocks as the previous part, code the Ball so that the sprite will forever follow and point towards the mouse-cursor, also known as “mouse-pointer”.

Again, for this we use three different blocks:

1. The “forever” block found in control.
2. The “point towards” block found in motion.
3. The “move” block found in motion.



Hints:

- Always click on the sprite in which you want to code. Clicking the sprite first will make sure you are adding scripts or code into that sprite only.
- The mouse-cursor or “mouse-pointer” is the little arrow you see when you use a computer. The “point towards” block by default has a “mouse-pointer” option in its drop down bar.
- The Vacuum sprite should be held the speed of the Ball sprite

Completing the Vacuum

Now that we have our starting positions - where every time we click the green flag our sprites start at a distance from each other, we have the Vacuum sprite following the Ball sprite and the Ball sprite following the mouse-cursor, we have a chain.

We need to code what happens when the Vacuum catches the Ball. To do this we will need an “if then” block and a “broadcast” block found in Controls. Lastly, a “touching” block found in Sensing and “hide” block found in Looks.

Challenge:

Can you code the Vacuum sprite so that when the Ball sprite is touched by the Vacuum sprite, a broadcast called “Ball Caught” would be sent. Afterwards hide the Vacuum sprite.

Hints:

- The “touching” block attaches to the “if then” block.
- When you click on the drop down bar of the “broadcast” block, you must write a new broadcast and call it something like “Ball Caught”.
- The “broadcast” block does not display a message on the screen,

Completing the Ball

Not as grand as it seems but lastly, to complete the Ball sprite, we also need to make it disappear when the Vacuum sprite catches it. We also need to stop the game.

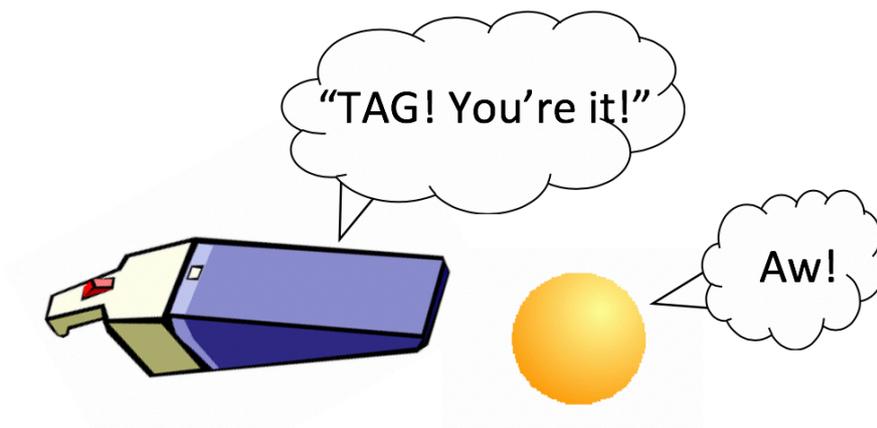
To do this we need a “when I receive” and a “stop all” block found in Controls and a “hide” block found in Looks.

This is where the “broadcast” block found in the Vacuum sprite is used. Similar to playing tag with friends, when the Vacuum sprite “tags” or touches the Ball sprite, the Vacuum shouts “Tag!” or “Tag! You’re it!”

The Ball sprite now certainly knows and then can do something about it.

Hints:

- The
I

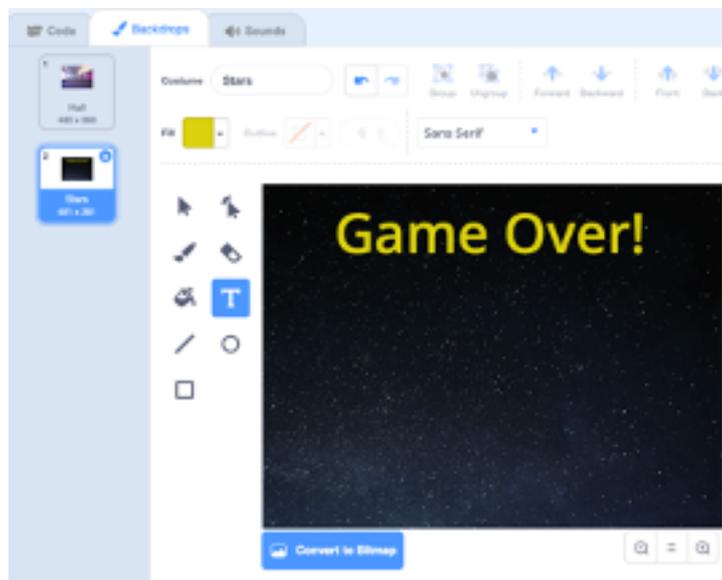


“when

Cosmetic: Backdrops - Game Over

To make it look pretty and professional why not add a backdrop or background. We can also add transitions where the backdrop changes into a game over display when the Ball is caught.

To do this first we need to click on the “backdrop” icon and then on the “backdrops” tab and import a background you wish to use during game play. Next, add another background and create over display.



Challenge:

Using the “when green flag clicked” and “when I receive” blocks, can you code the game so that when the green flag is clicked, the game play backdrop is displayed and when the Vacuum catches the Ball, the game over backdrop is displayed.

Power Ups!

How about adding power ups to your Ball sprite to give him that extra edge to get away from the wicked Vacuum?

Maybe, when you press the “space” bar, the Ball sprite gains a temporary speed boost? A different button might even give the sprite the power to shrink or temporary immunity perhaps?

More Enemies!

If you manage to add power ups to the Ball sprite and you are finding it way too easy now, what about adding more enemies to dodge? With different shapes and sizes!

You may duplicate the Vacuum and use it as a template.

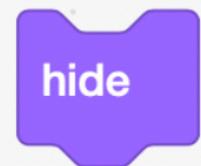
Score System!

If you have made it this far, well done! You have a completed game but you can still make it better. The last challenge is to add a score system to the game? The longer you stay alive the more points you get.

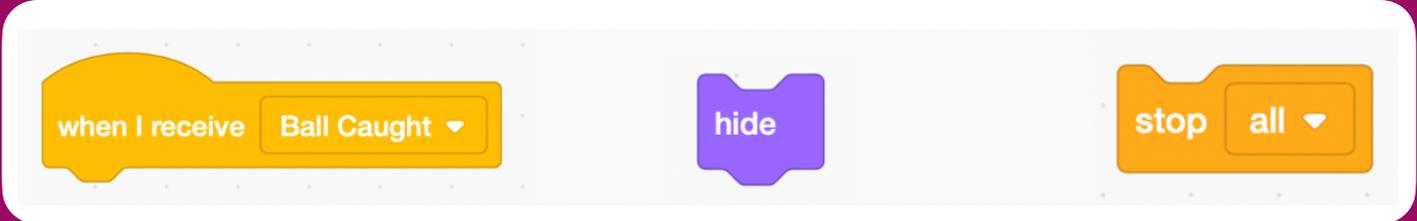
Using the “go to” Block



Completing the Vacuum



Completing the Ball



Keywords

Thinking outside the box
Problem solving
Team building

Sprites
Loops
if Statements

Thinking outside the box - or thinking beyond the box is a phrase that means to think differently, unconventionally, or from a new perspective. This phrase is often referred to as creative thinking.

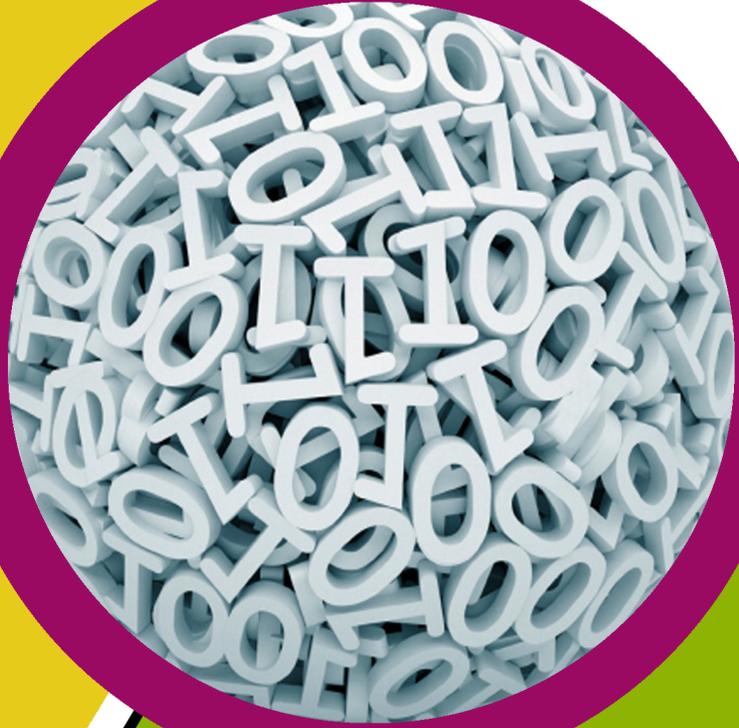
Problem solving - is the process of working through the problems details step by step to reach a solution.

Team building - is the ability to be able to adapt to situations with your teams, and to motivate and help each other through each task.

Sprites - the images in our game that contain blocks of code in order to interact with one another.

Loops - these allow sections of blocks to be repeated.

if Statements - these check if a condition is met and executes



technocamps



@Technocamps



Find us on
Facebook