# Building and predicting what a program will do using a Visual Programming Language (Scratch)

**Experiences and Outcomes** *(following Curriculum for Excellence used in Scotland)*

I understand the instructions of a visual programming language and can predict the outcome of a program written using the language. **TCH 1-14a**

I understand how computers process information. **TCH 1-14b**

---

**Learning Intention –** We are using blocks of code to make our sprite perform actions. (i.e. using a visual programming language to determine certain outcomes happen)

**Success Criteria – I can**

- predict what will happen on my program by looking at the code
- use the if/else block on my scratch application to do two different operations

---

## Introduction

Describe to the class how computers are very literal and logical creations. Computers do exactly what we tell them to do, which can be extremely useful and sometimes very irritating! One example that children will be familiar with is every time they insert an incorrect username and/or password they will not be able to log on to a computer.

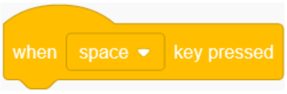
*How do computers process information then?*

Computers take in information from every action you do on it (when you click on something, when you type something etc). Computers are always ready to take in information from you when they are switched on and it is the same on scratch. On scratch the sprite is always waiting blocks of code from you so it can do exactly what you tell it to do!

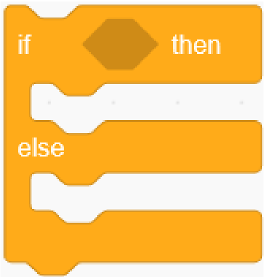
## Main Teaching

Another example of computers doing exactly what you tell it to do is adding a background to your project. *(model this then let the children have a go – give a time limit for choosing a background, they can spend FAR too long on this!)*

First, to get your project to do something, we need to use a block from the EVENTS section to get started. The following block (pictured to the right) is what I would recommend for this lesson for the top of your code.

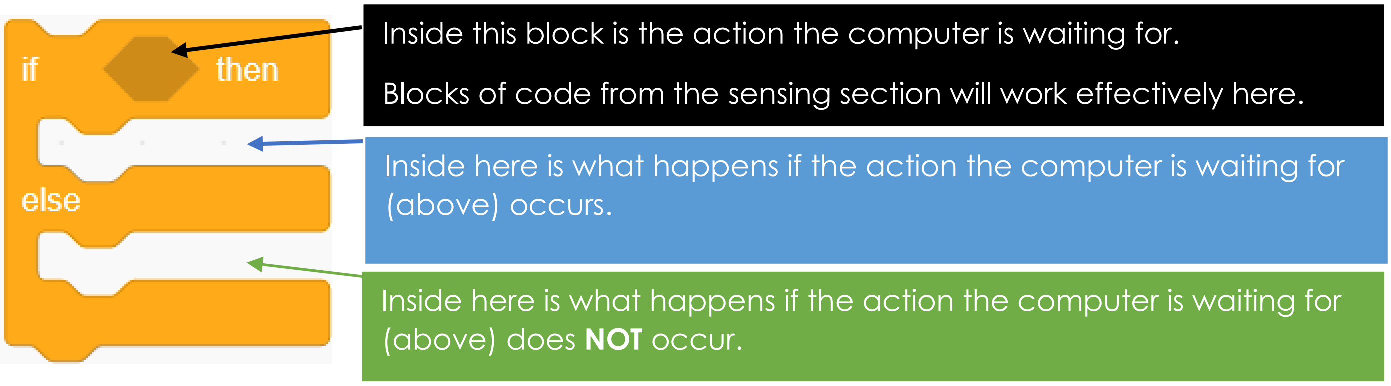


The if else block



This block allows us to have two actions that can happen when a certain event happens. You will find it in the Control section, and you can attach the block by dragging it on to the 'when space key pressed' block of code.

## What does this block do?



Inside this block is the action the computer is waiting for.

Blocks of code from the sensing section will work effectively here.

Inside here is what happens if the action the computer is waiting for (above) occurs.

Inside here is what happens if the action the computer is waiting for (above) does **NOT** occur.

Here is an example of an if else block changing the action the sprite does program changing what action it does according to the if else block.

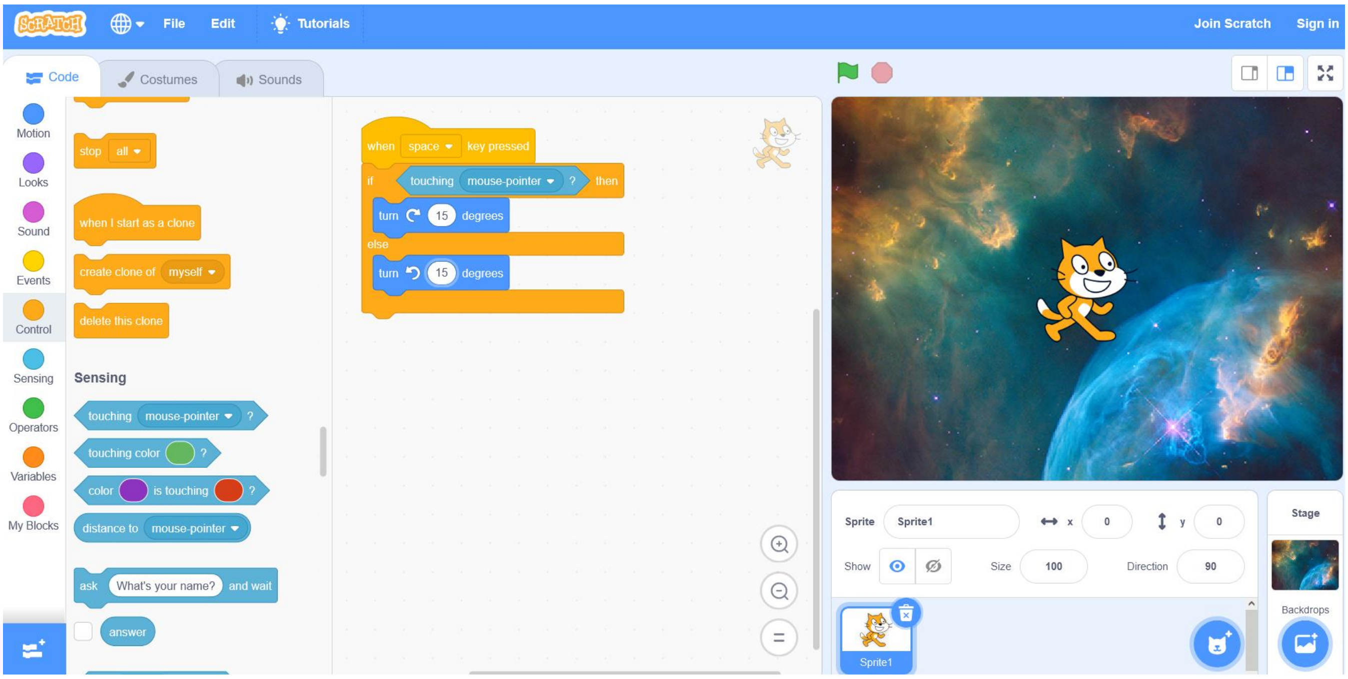
## How this program works

Line 1 - The Scratch program waits for the user to enter the space key

Line 2  - Checks if the **mouse pointer** is touching the cat.

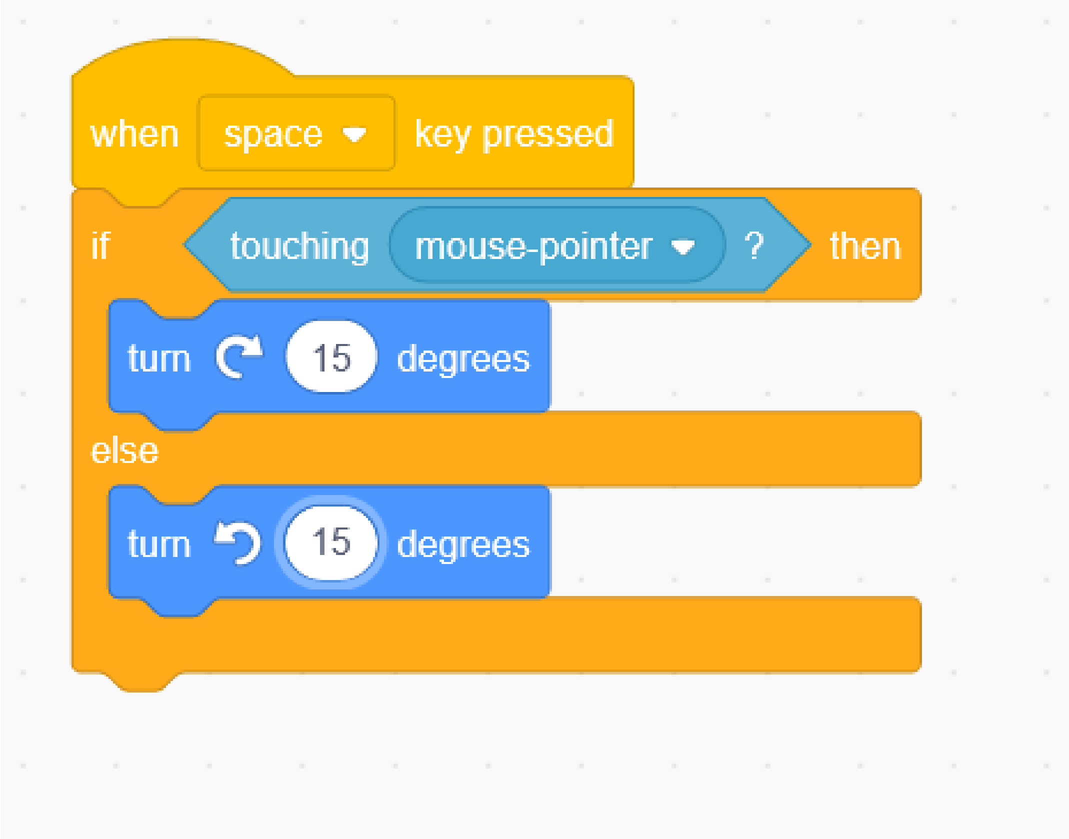
Line 3 - **If** the mouse pointer is on the cat – the cat will rotate clockwise whenever space is entered.

Line 4 - If the pointer is not on the cat – it will rotate anti-clockwise.



```
when space key pressed
if touching mouse-pointer ? then
    turn ↻ 15 degrees
else
    turn ↺ 15 degrees
```

# Tasks



```
when [space ▾] key pressed
if <touching [mouse-pointer ▾] ?> then
    turn ↻ (15) degrees
else
    turn ↺ (15) degrees
```

*This is the code from above example*

Get the children to build this block of code and experiment with it. What happens if they put different motion blocks inside this if else block? (It is much easier for the children to create their own examples from this template)

What happens if they change the degrees?

Get children to use more than one motion block inside an if else block? What can they create?

Can they discuss it to a peer, or build it on the whiteboard and model it to the rest of the class?

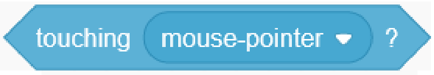
Can you predict what will happen just by looking at someone else's code?

---

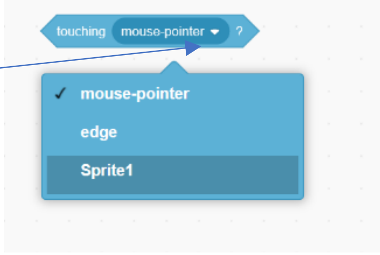
**Support Ideas** – Using the block example above, model using different motion blocks instead of turn 15 degrees. When you use a different block, test the changes so the children can see how the blocks of code lead to different actions being performed by the sprite. This will help children being to see how the if else block allows the sprite to perform different actions.

---

**Extension Tasks** - Create another sprite and get the two sprites to interact with each other. (*With the sensing blocks – the following block of code can be found, which will prove useful*).



```
<touching [mouse-pointer ▾] ?>
```



By clicking the drop-down menu you can select the other sprite (which will be named Sprite1) – this will help the sprites interact with each other.

## **Plenary**

Can the children describe how the if else block works?

Can children share their project to a partner, so that they fully understand what happens – before showing their project in action to prove it?