

# TOOL - SCRATCH: FREQUENT ERRORS

DIGITAL CONTENT CREATION > 3.4 PROGRAMMING

TARGET GROUP	AGE GROUP	PROFICIENCY LEVEL	FORMAT	COPYRIGHT	LANGUAGE
Facilitators	N/A	Level 2	Preparatory guide	Creative Commons (BY-SA)	English, French

This document contains background information for facilitators before they run the workshop with participants. It gives some tips on how to avoid some common mistakes when programming with Scratch.

**General Objective** Skillset building

**Preparation time for facilitator** less than 1 hour

**Competence area** 3 - Digital content creation

**Resource originally created in** French

## WORKSHOP DIRECTIONS

### 1 Lack of looping

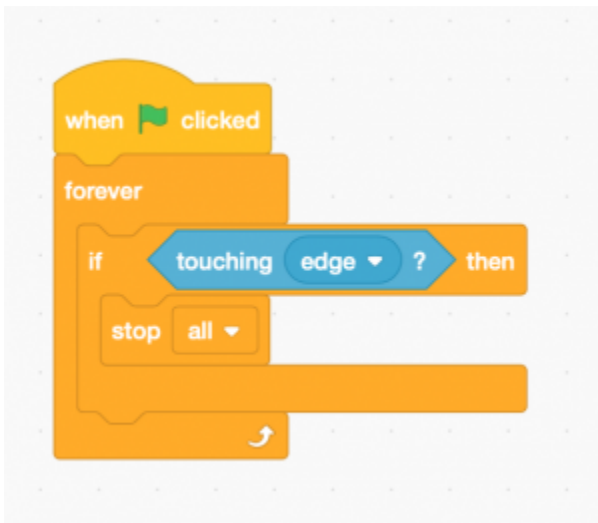
**Symptoms:** Some elements don't work, such as collision detection, code with conditions, etc.

**Cause:**



Here, the game will not work since the code only verifies once (at the point of execution) if a condition is true, at which point the code stops.

**Resolution:**



By adding a loop, the computer will constantly check if a condition is true.

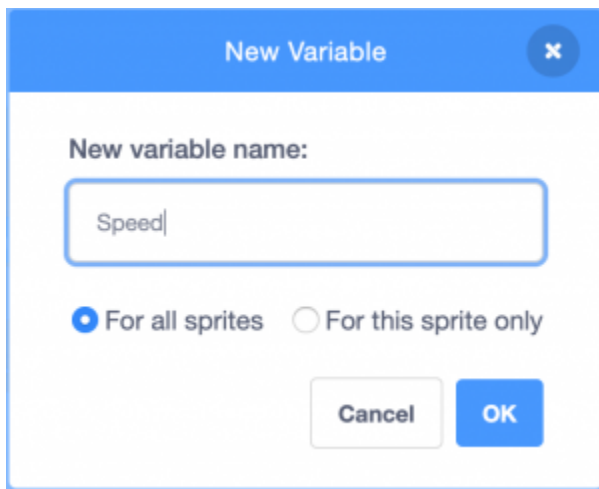
## 2 The global variable

### Symptoms:

A variable does not appear or does not work.

### Cause :

While creating a variable, there are two choices: apply to to all sprites (global variable) or only for one sprite (local variable). Generally, an issue may arise when we use a local variable.



**Resolution:** create a new variable

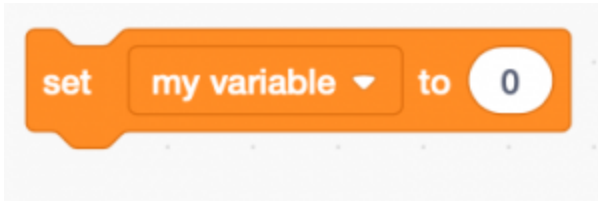
## 3 Setting and adding to variables

### Symptoms:

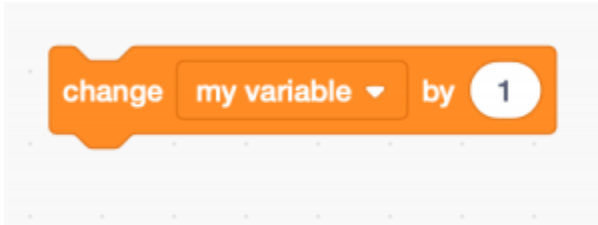
The value of a variable does not behave the way we want it – it resets itself to 0 at the beginning.

### Cause:

There are two different blocks:



The above block sets the variable to 0. This can be used for example to reset the scores to 0 at the beginning of the game.



This adds 1 to the current value. This can be used to increase a score for example.

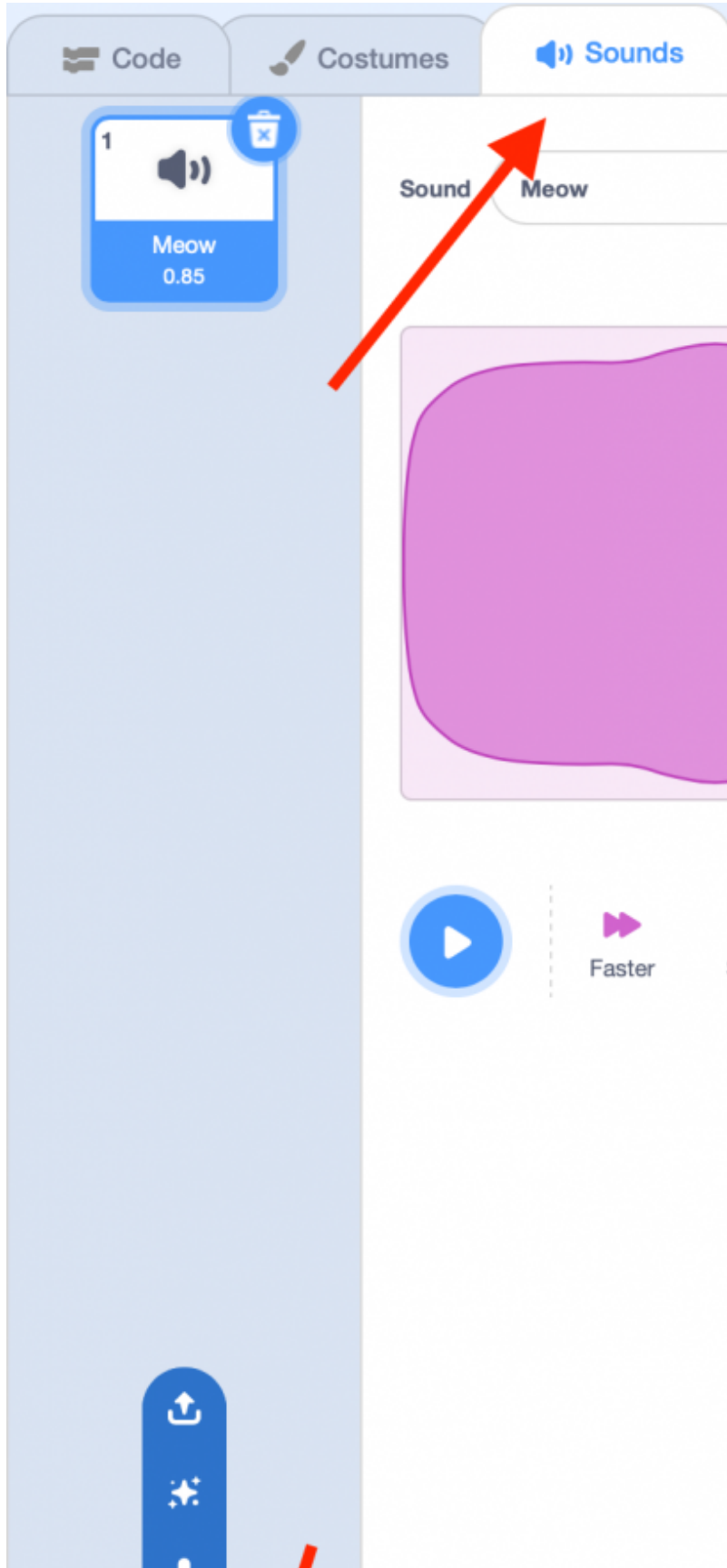
**Resolution:** Choose the correct block

## 4 Importing and playing sounds

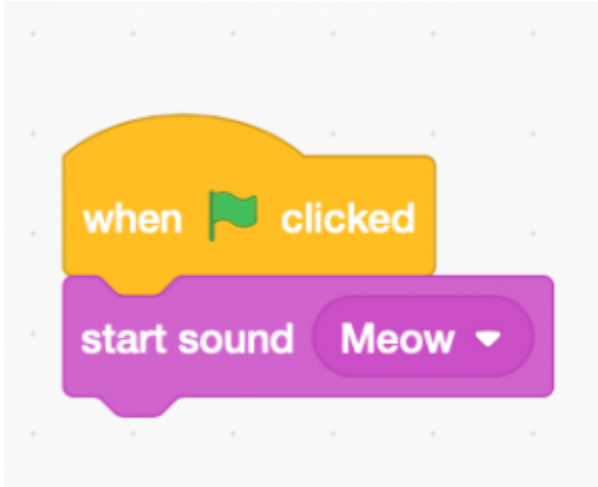
**Symptoms:** A sound is not played.

**Cause**

: To play a sound, first go to the tab entitled 'Sound' and import (or record) a sound.



Once this is stored in memory, we need to program when the sound will be played.

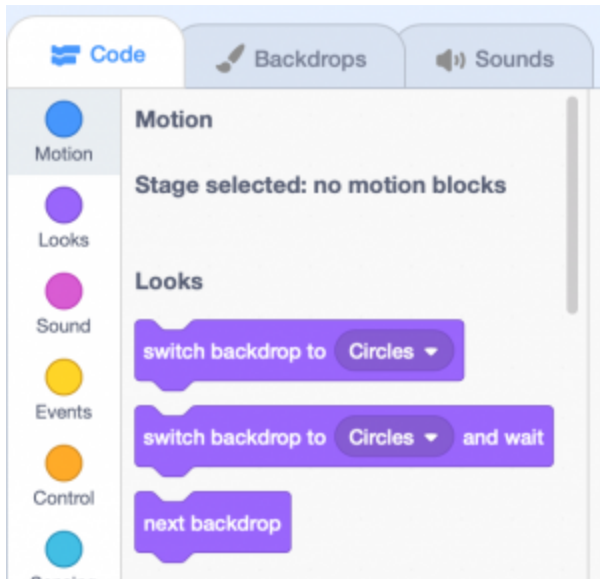


**Resolution:** (see below)

## 5 Scripting backdrops and sprites

**Symptoms:** There are blocks missing from the library.

**Cause:** You are currently in the backdrop window, and since backdrops don't move, there are no movement blocks.



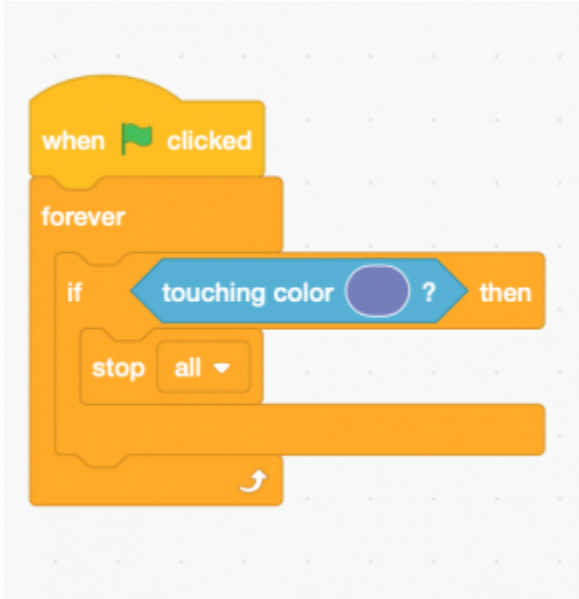
**Resolution:** Click on your sprite.

## 6 Collision detection problems

**Symptoms:** You have placed colour detection blocks, though your sprite does not detect the colour or does so poorly.

**Cause**

: With a block of this type, if the colour appears at all in the backdrop, the condition will be met.



**Resolution:** Avoid colour detections as much as possible or choose colours which do not appear in your background. (For example, fluorescent green hardly appears in nature).

## 7 Some general advice for problem-solving

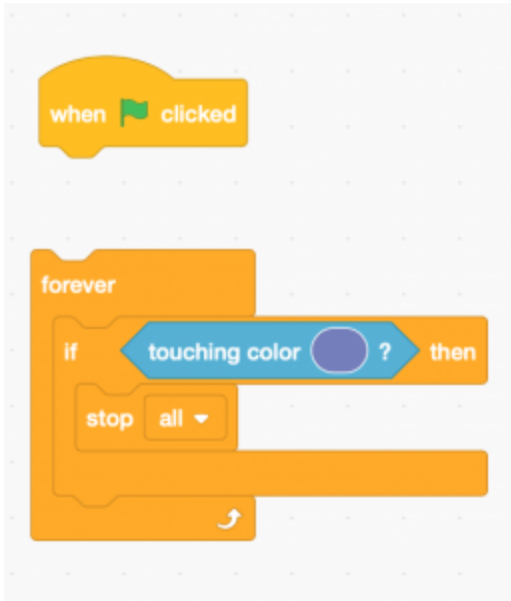
- **Know clearly what you want to do:**

If you don't know exactly what you want to program, there is little chance the computer will understand you.

- **Identify clearly what does what:**

which code is responsible for which part of the game, which objects manage detections etc.  
Feel free to 'take apart' the code:





By moving the blocks around like this, the code will no longer be active. This is often useful when we want to identify where an error is coming from.

- Try writing your code/algorithm on a **piece of paper**: Sometimes errors can become clearer when we think outside the screen for a while.