Erasmus+

2Play 2Learn

Reviving hands-on educational play for learning skills of tomorrow
PROJECT N° 2019-1-UK01-KA201-061466

MODULE 2

Scratch 2.0

DEVELOPED BY IDEC & PLATON

CIVIC idec ΕΚΠΑΙΔΕΥΤΗΡΙΑ ΠΛΑΤΩΝ dlearn EUROPEAN DIGITAL LEARNING NETWORK scholé Emphasys CENTRE CCS Digital Education

MODULE DESCRIPTION

Scratch is a visual programming tool which allows the user to create animations and games with a drag-and-drop interface. It allows you to create your own computer games, interactive stories, and animations using some programming techniques without actually having to write code. It’s a great way to get started programming on the Raspberry Pi with young people.

The version of Scratch included with the Raspberry Pi has a number of unique features; one of the most useful is its ability to communicate with the GPIO pins (General Purpose Input Output). These pins allow you to connect your Raspberry Pi to a range of devices, from lights and motors to buttons and sensors.

Co-funded by the
Erasmus+ Programme
of the European Union

LEARNING OUTCOMES

By completing this resource you will learn:

- How to program in Scratch 2.0
- How to control the GPIO pins using Scratch

Co-funded by the
Erasmus+ Programme
of the European Union

TOPICS

- Getting started with Scratch 2.0
- Physical Computing With Scratch

Co-funded by the
Erasmus+ Programme
of the European Union

MODULE 2 – Scratch 2.0

# Getting started with Scratch

LEARNING OUTCOMES

- Getting started
- Start moving
- Add a sound
- Start a dance
- Again and again
- Say something
- Green flag
- Change color

Co-funded by the
Erasmus+ Programme
of the European Union

LEARNING OUTCOMES

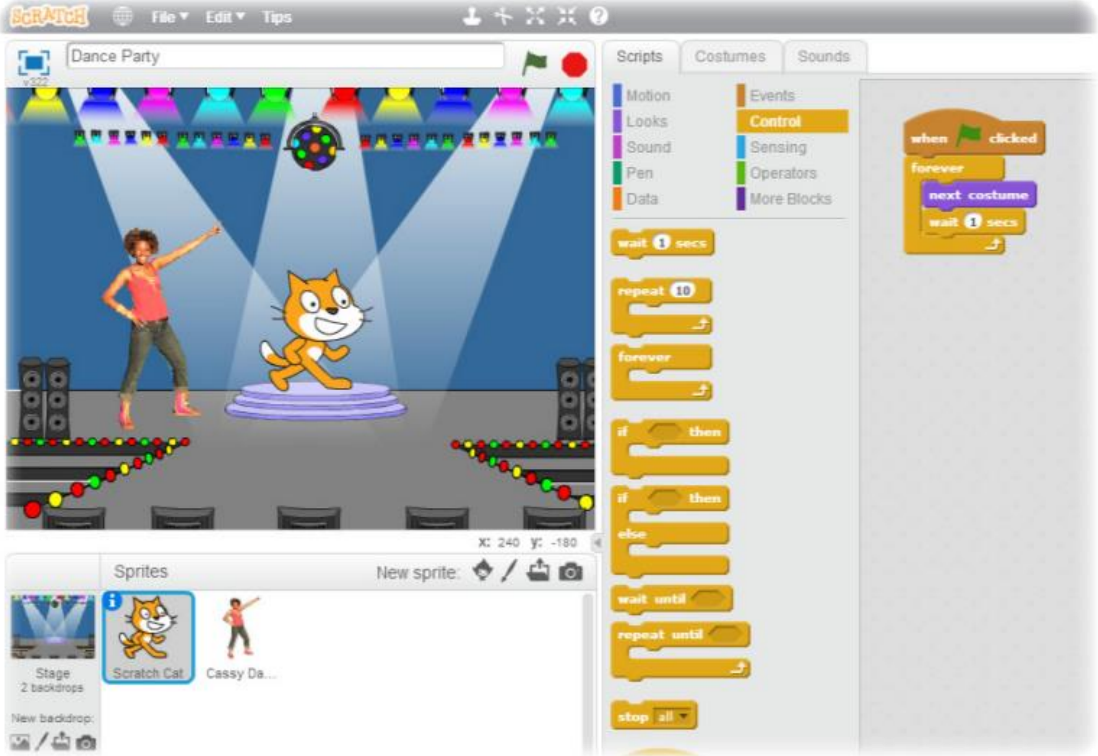
- Key press
- Add a backdrop
- Add a sprite
- Explore
- Tips
- Save and share

Co-funded by the
Erasmus+ Programme
of the European Union

MODULE 2 – Scratch 2.0

Getting started with Scratch – Getting started

SCRATCH is a programming language that lets you create your own interactive stories, animations, games, music, and art.



This guide shows you how to make a project in SCRATCH.

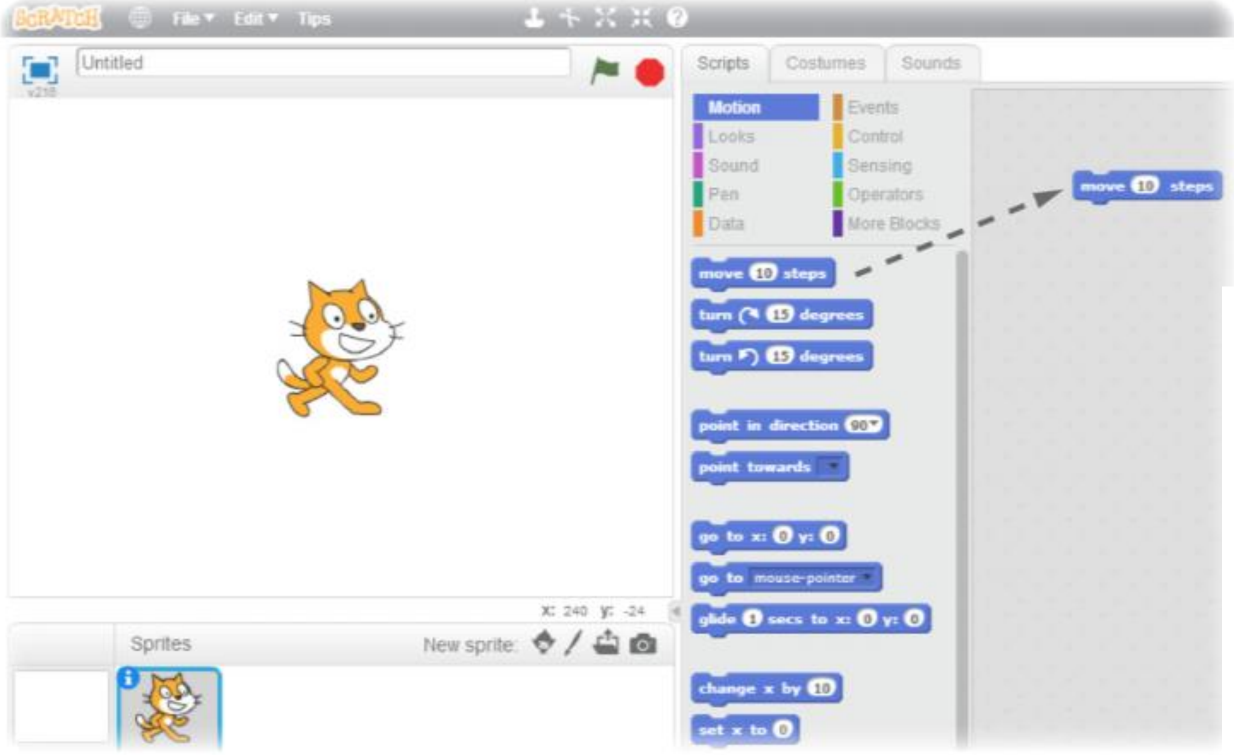
To start a new project, go to SCRATCH and click **Create**.

If you have a SCRATCH account, sign in so your project saves.

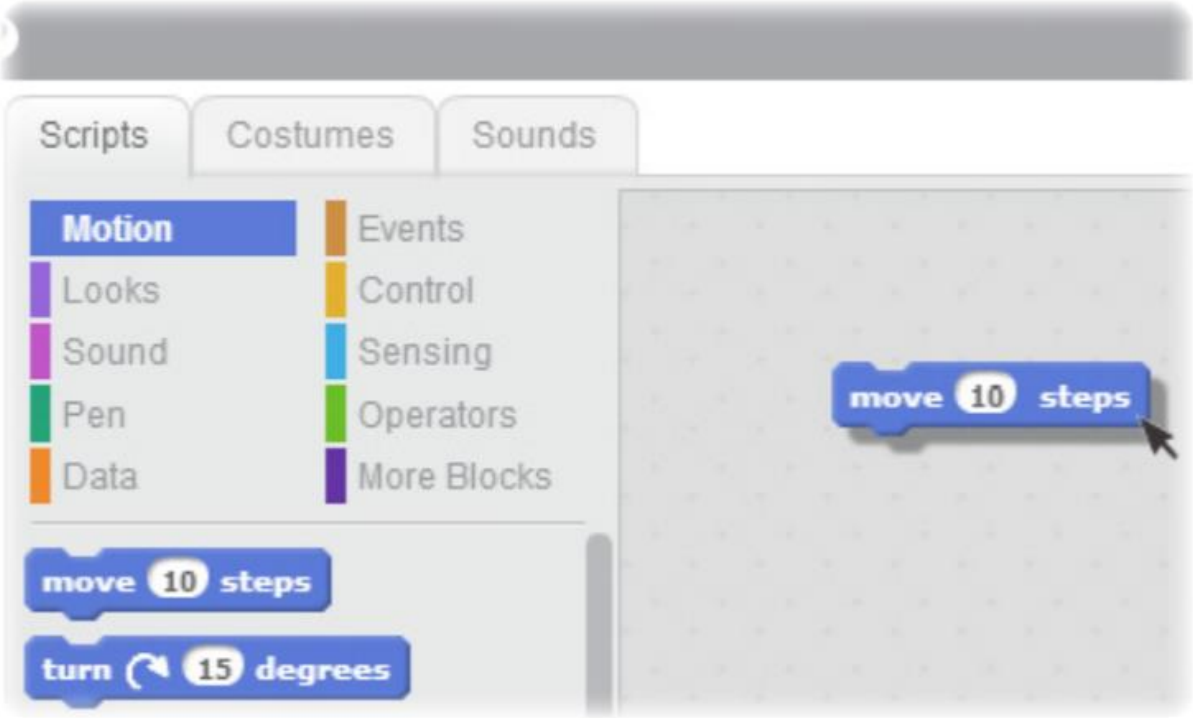
Co-funded by the
Erasmus+ Programme
of the European Union

MODULE 2 – Scratch 2.0

Getting started with Scratch – Start moving



Drag a **MOVE** block into the Scripts area.

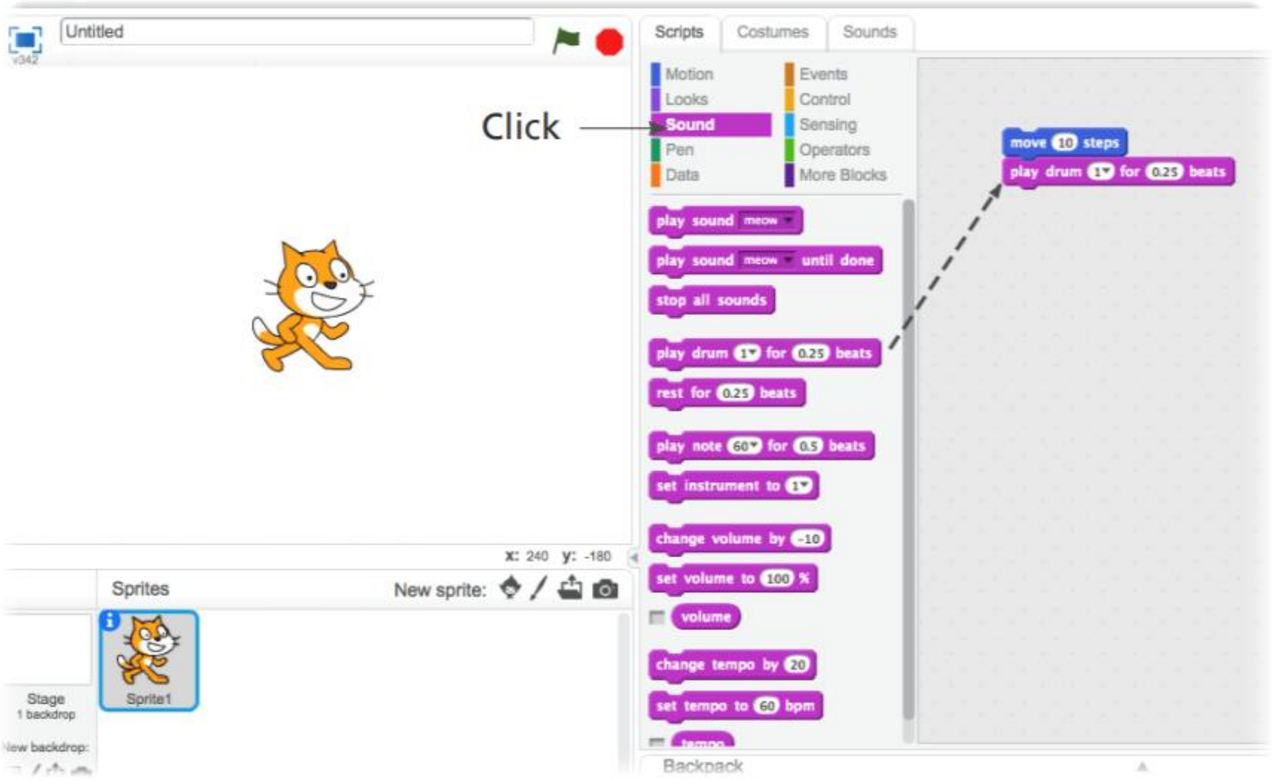


Click on the block to make the cat move.

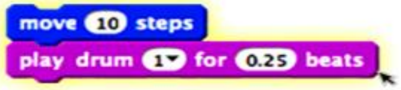
Co-funded by the
Erasmus+ Programme
of the European Union

MODULE 2 – Scratch 2.0

Getting started with Scratch – Add a sound

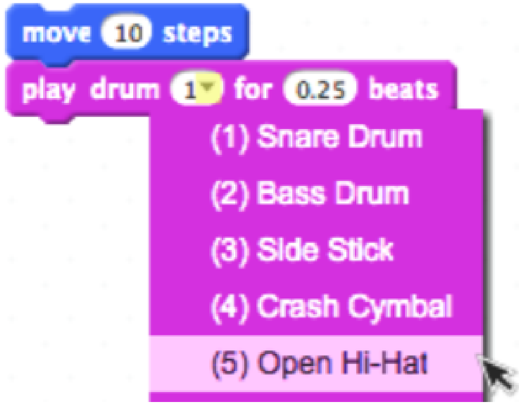


Drag out a PLAY DRUM and snap it onto the MOVE block.



Click and listen.

If you can't hear it, check that the sound on your computer is on.

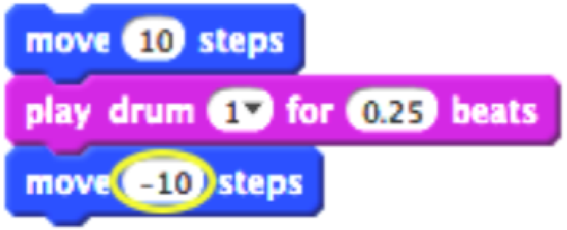


You can choose different drums from the pull-down menu.

Co-funded by the Erasmus+ Programme of the European Union

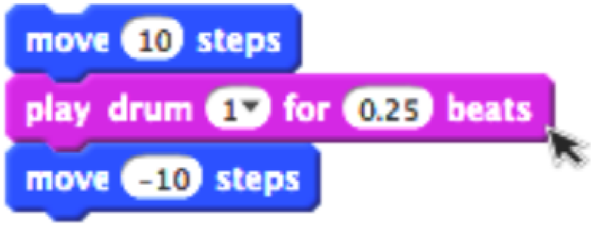
MODULE 2 – Scratch 2.0

Getting started with Scratch – Start a dance



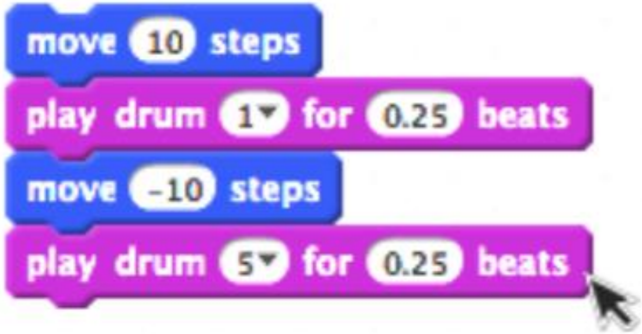
```
move (10) steps
play drum (1 ▾) for (0.25) beats
move (-10) steps
```

Add another **MOVE** block. Click inside the block and type in a minus sign.



```
move (10) steps
play drum (1 ▾) for (0.25) beats
move (-10) steps
```

Click on any of the blocks to run the stack.



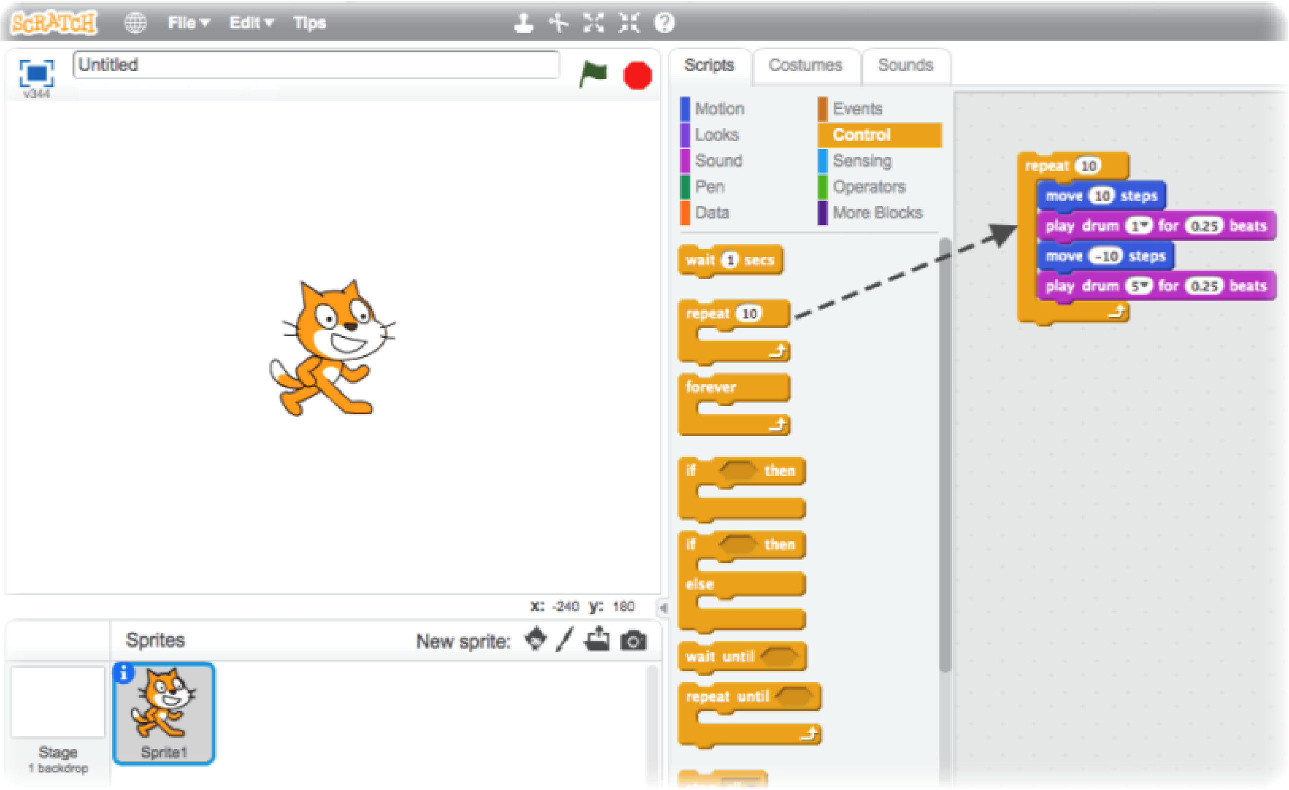
```
move (10) steps
play drum (1 ▾) for (0.25) beats
move (-10) steps
play drum (5 ▾) for (0.25) beats
```

Add another **PLAY DRUM** block, then choose a drum from the menu. Click to run.

Co-funded by the
Erasmus+ Programme
of the European Union

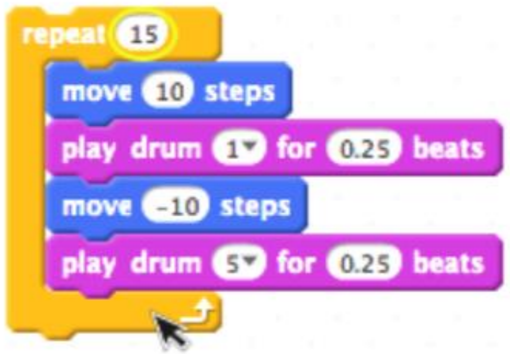
MODULE 2 – Scratch 2.0

Getting started with Scratch – Again and again



Drag out a **REPEAT** block and drop it on top of the stack.
You want the mouth of the **REPEAT** to wrap around the other blocks.

To drag a stack, pick it up from the top block.



You can change how many times it repeats.

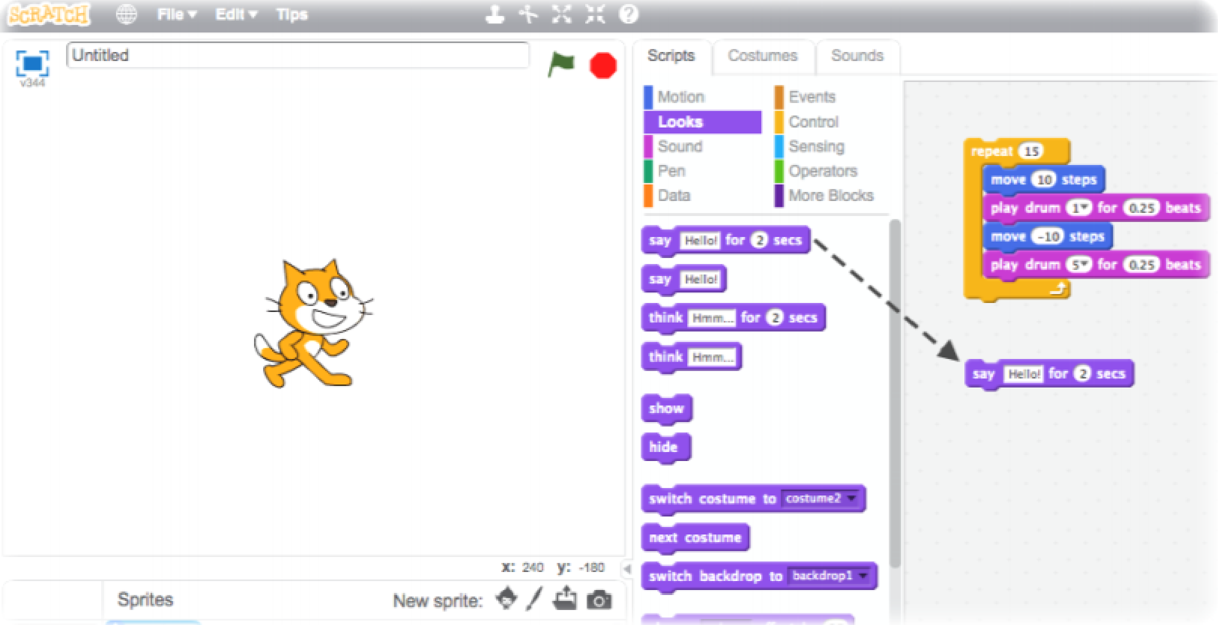
Click to run.

You can click on any block to run a stack.

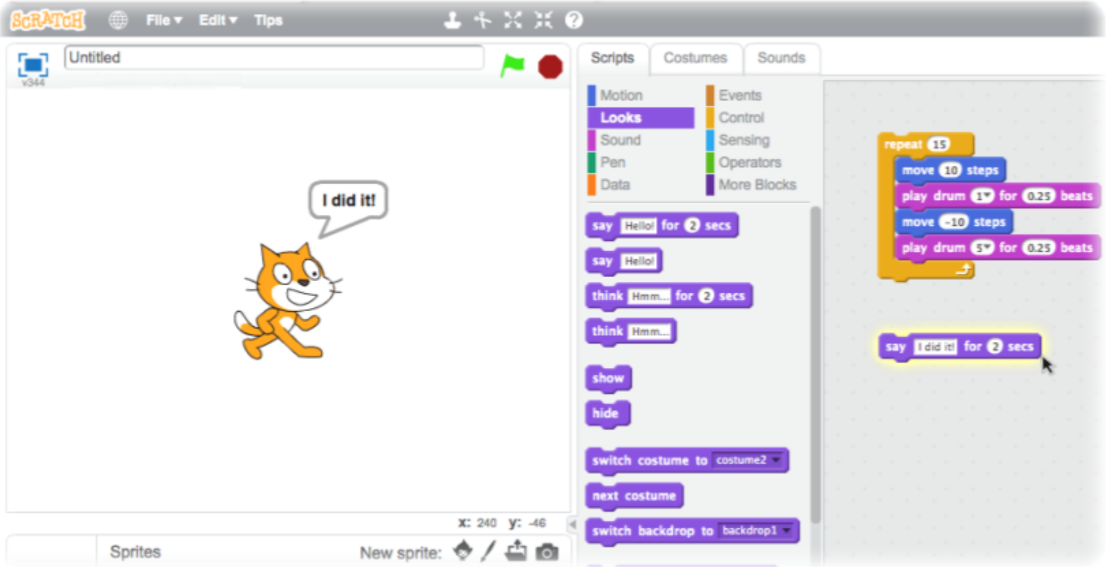
Co-funded by the
Erasmus+ Programme
of the European Union

MODULE 2 – Scratch 2.0

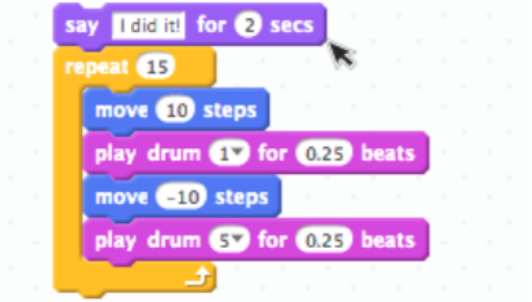
Getting started with Scratch – Say something



Click the LOOKS category and drag out a SAY block.



Click inside the SAY block and type to change the words. Click to try it.

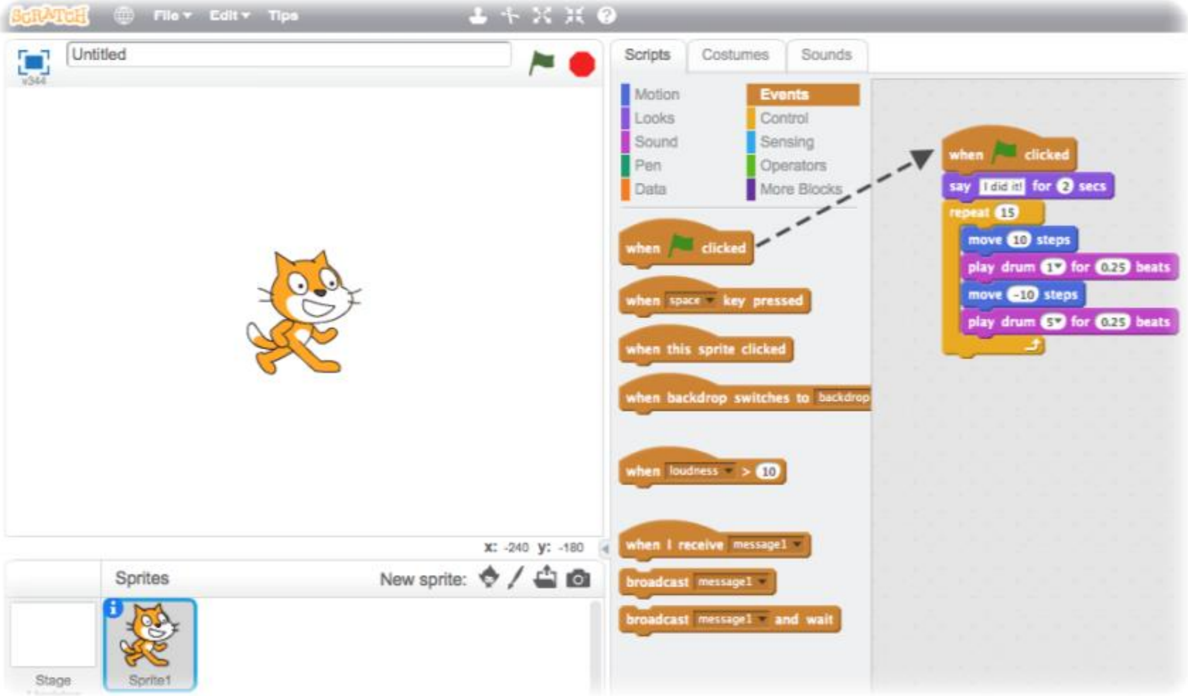


```
say I did it! for 2 secs
repeat 15
    move 10 steps
    play drum 1 for 0.25 beats
    move -10 steps
    play drum 5 for 0.25 beats
```

Then snap the SAY block on the top.

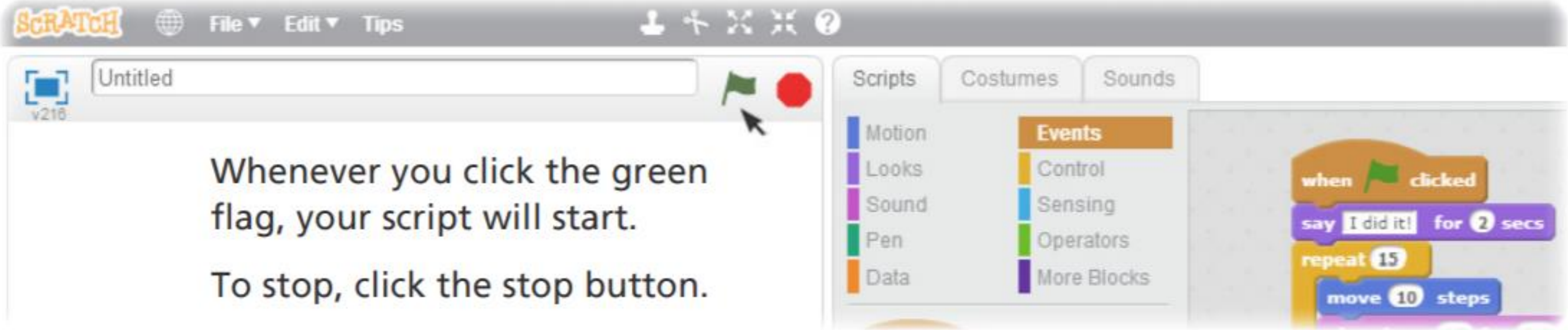
Co-funded by the
Erasmus+ Programme
of the European Union

Getting started with Scratch – Green Flag



```
when ⚑ clicked
say I did it! for 2 secs
repeat 15
    move 10 steps
    play drum 1 for 0.25 beats
    move -10 steps
    play drum 5 for 0.25 beats
```

Drag out a **when ⚑ clicked** block and snap it on top.



Whenever you click the green flag, your script will start.

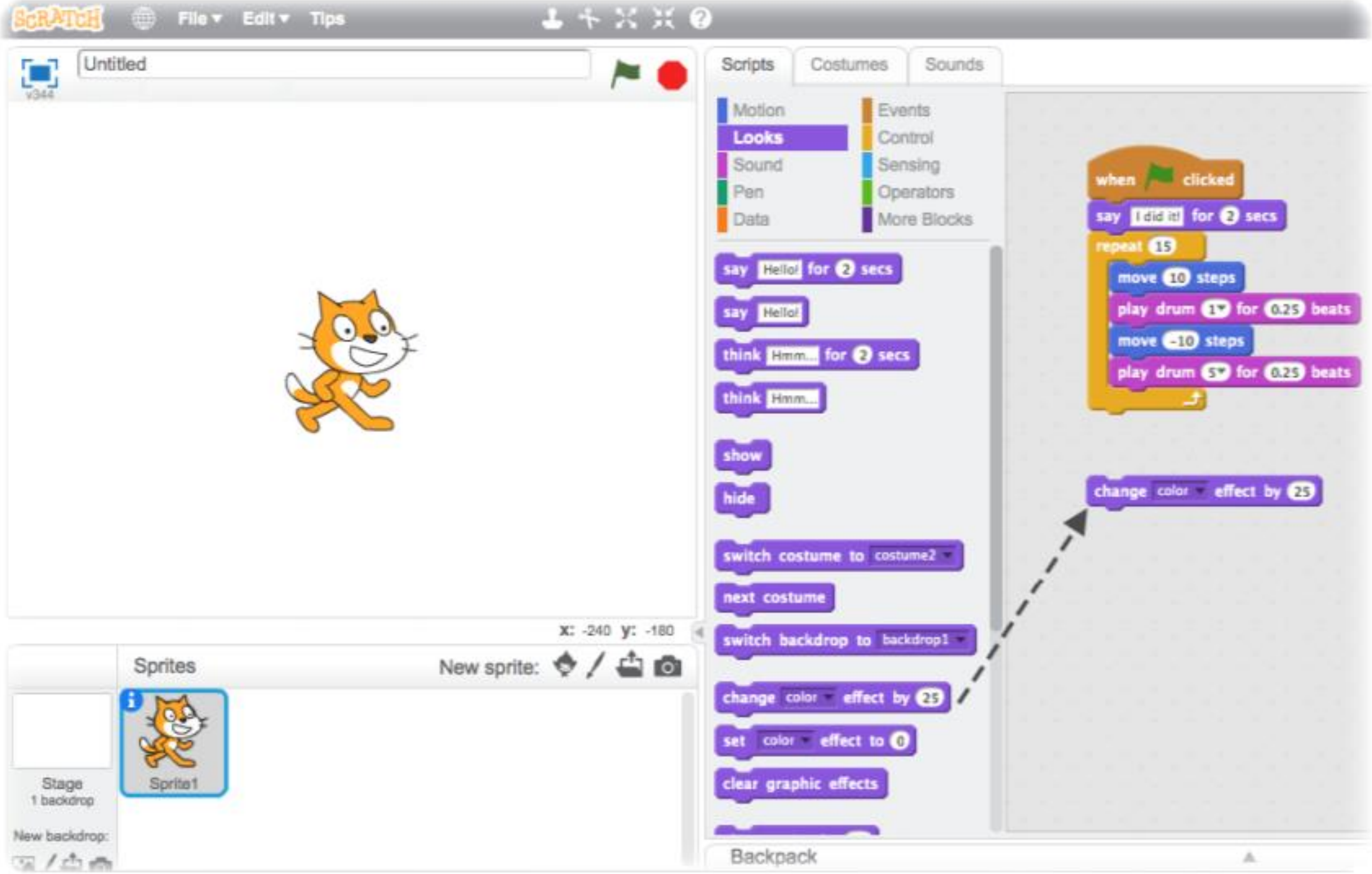
To stop, click the stop button.

```
when ⚑ clicked
say I did it! for 2 secs
repeat 15
    move 10 steps
```

Co-funded by the Erasmus+ Programme of the European Union

MODULE 2 – Scratch 2.0

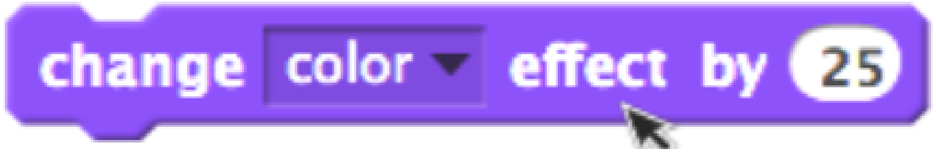
Getting started with Scratch – Change color



```
when [flag] clicked
say I did it! for 2 secs
repeat 15
    move 10 steps
    play drum 1 for 0.25 beats
    move -10 steps
    play drum 5 for 0.25 beats

change color effect by 25
```

Drag out a **CHANGE EFFECT** block.

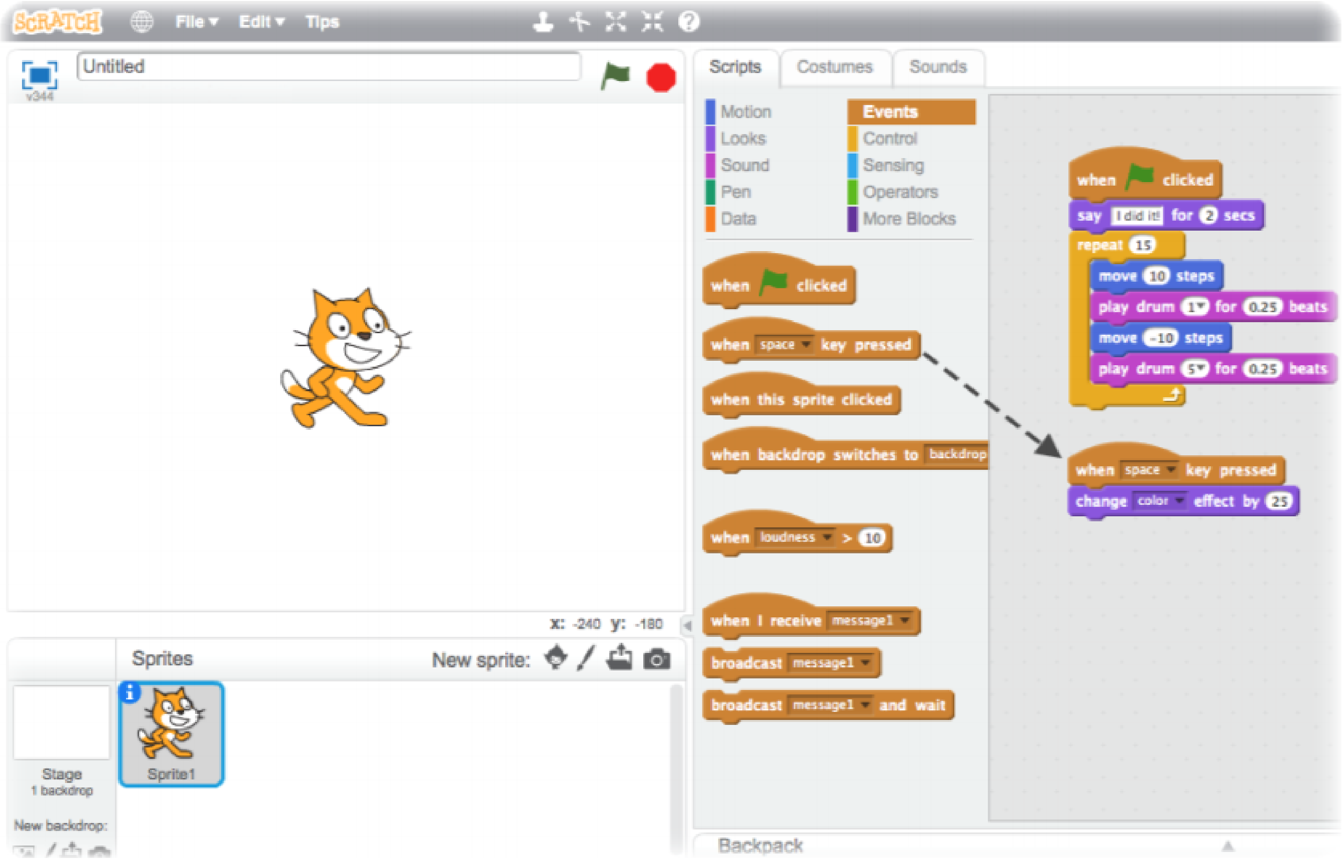


Click to see what it does.

Co-funded by the
Erasmus+ Programme
of the European Union

MODULE 2 – Scratch 2.0

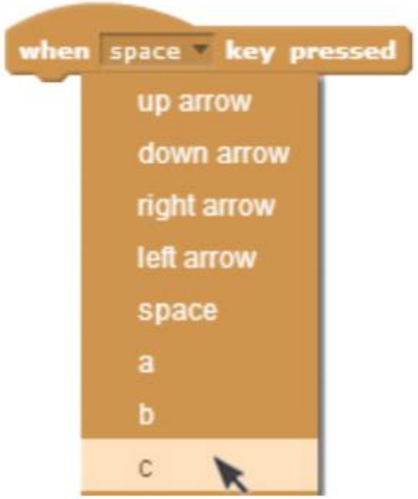
Getting started with Scratch – Key press



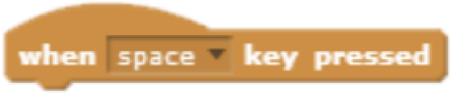
```
when [flag] clicked
say [I did it!] for (2) secs
repeat (15)
    move (10) steps
    play drum (1▾) for (0.25) beats
    move (-10) steps
    play drum (5▾) for (0.25) beats

when [space ▾] key pressed
change [color ▾] effect by (25)
```

Now press the space bar on your keyboard.



You can choose a different key from the pull-down menu.

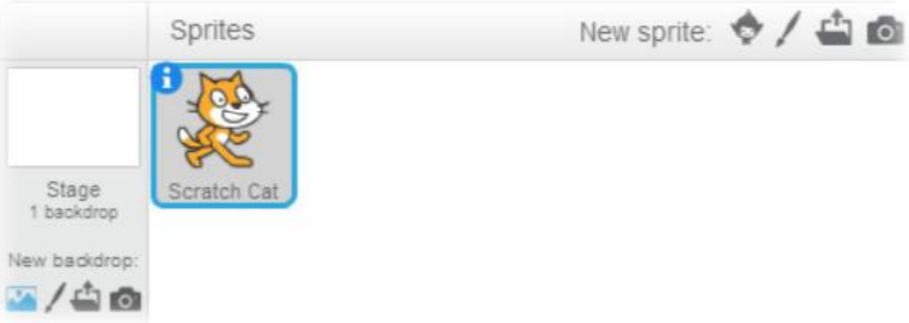
Snap on a 

Co-funded by the Erasmus+ Programme of the European Union

MODULE 2 – Scratch 2.0

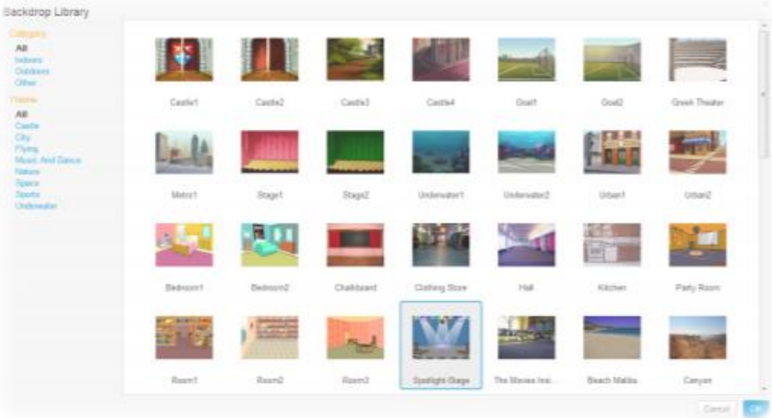
Getting started with Scratch – Add a backdrop

You can add a backdrop to the Stage.



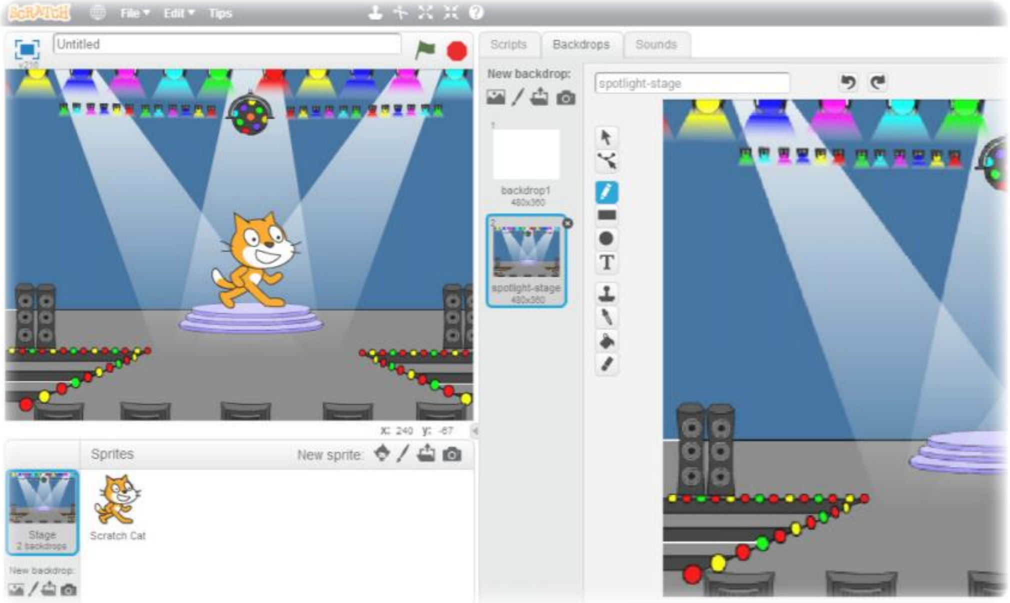
Click 🖼 to choose a new backdrop.

Choose a backdrop from the library (such as "Spotlight-Stage").



Click OK.

The new backdrop now appears on the Stage.

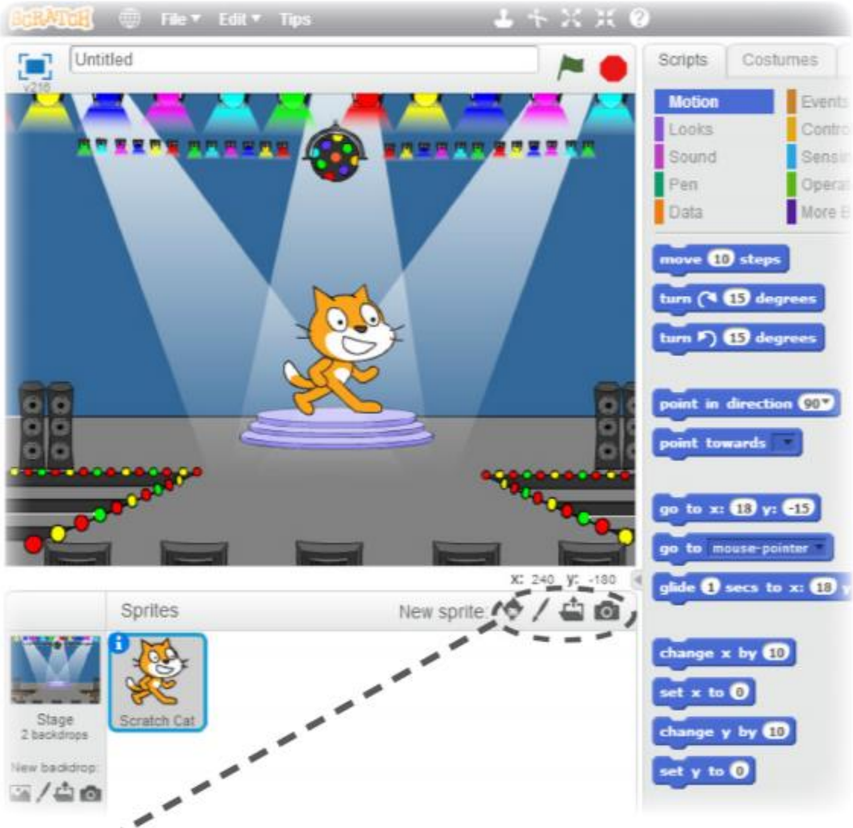


Co-funded by the
Erasmus+ Programme
of the European Union

MODULE 2 – Scratch 2.0

Getting started with Scratch – Add a sprite

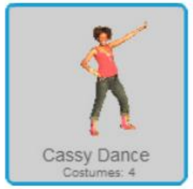
Each object in Scratch is called a sprite.



To add a new sprite, click one of these buttons.

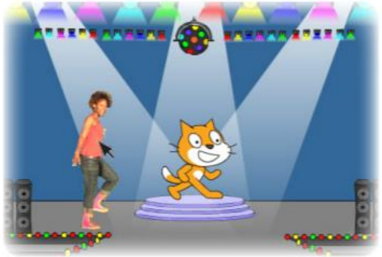
NEW SPRITE BUTTONS:

- Choose from the library
- Paint your own sprite
- Upload your own image or sprite
- Take a picture (from a webcam)



To add this sprite, click then click **People** and select "Cassy Dance."

You can drag the characters to where you want them.

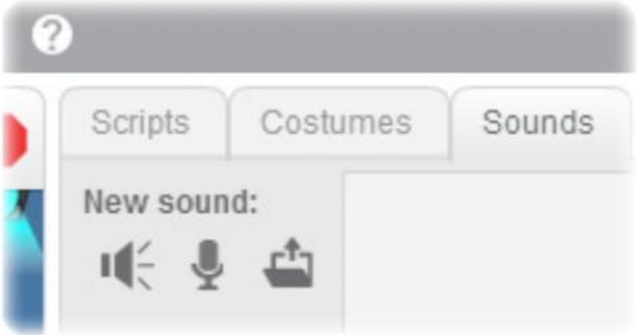


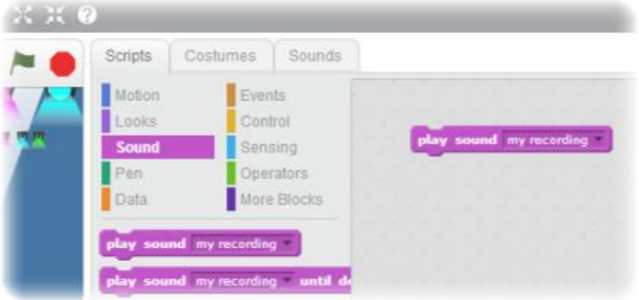
Co-funded by the
Erasmus+ Programme
of the European Union

MODULE 2 – Scratch 2.0

Getting started with Scratch – Explore

Now you can tell the sprite what to do. Try the following, or explore on your own.





```
play sound my recording
```

ADD SOUND

Click the **SOUNDS** tab.

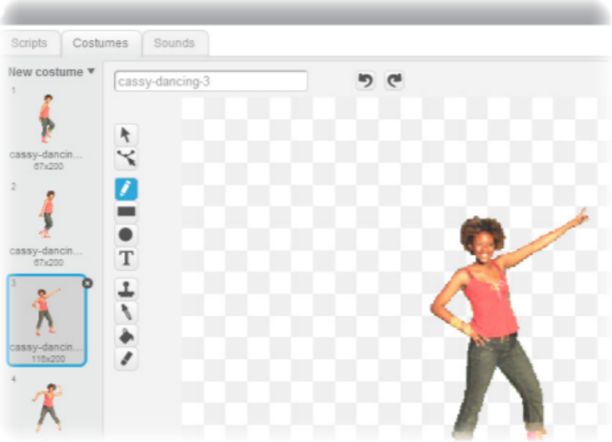
You can **Choose** 🔈 a sound

Record 🎤 your own sound

Or **Import** 📤 a sound file.
(MP3, AIF, or WAV format)

Then, click the **SCRIPTS** tab,
and drag in a PLAY SOUND block.

Choose your sound from
the pull-down menu.

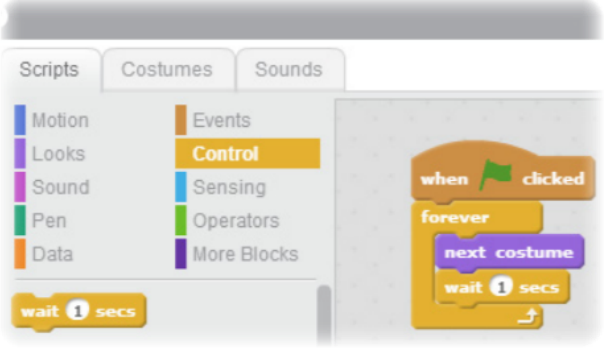


CHANGE COSTUMES

Each sprite can have more than one costume.

To change the current costume, click the **COSTUMES** tab.

Then click on a different costume for the sprite.



```
when 🏁 clicked
forever
    next costume
    wait 1 secs
```

ANIMATE

You can animate a sprite by switching between costumes.

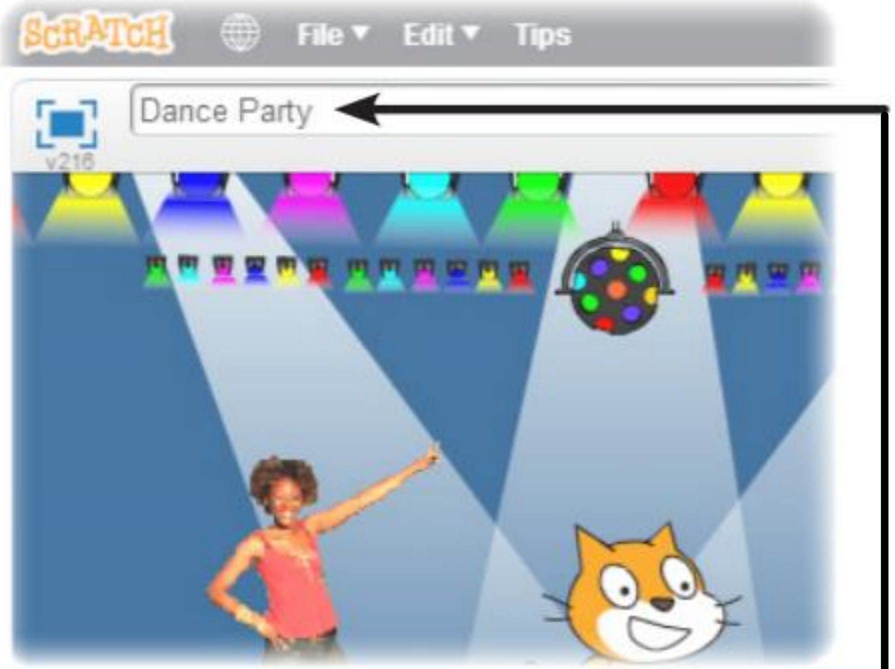
Click the **SCRIPTS** tab.

Create a script that switches between costumes.

Co-funded by the
Erasmus+ Programme
of the European Union

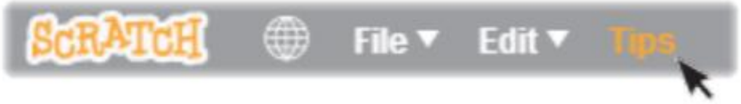
MODULE 2 – Scratch 2.0

Getting started with Scratch – Tips



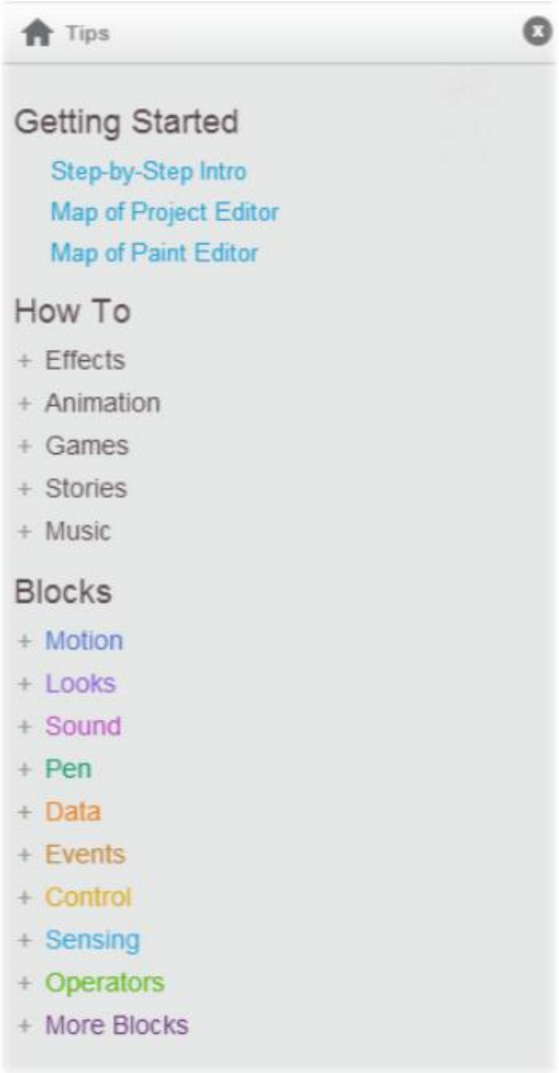
Type a title for your project.

For more ideas, click **Tips**:



The **Tips Window** shows example scripts you can use in your project.

It also explains what each of the blocks in SCRATCH does.



Tips

Getting Started

Step-by-Step Intro

Map of Project Editor

Map of Paint Editor

How To

- + Effects
- + Animation
- + Games
- + Stories
- + Music

Blocks

- + Motion
- + Looks
- + Sound
- + Pen
- + Data
- + Events
- + Control
- + Sensing
- + Operators
- + More Blocks

Co-funded by the
Erasmus+ Programme
of the European Union

MODULE 2 – Scratch 2.0

Getting started with Scratch – Save and share

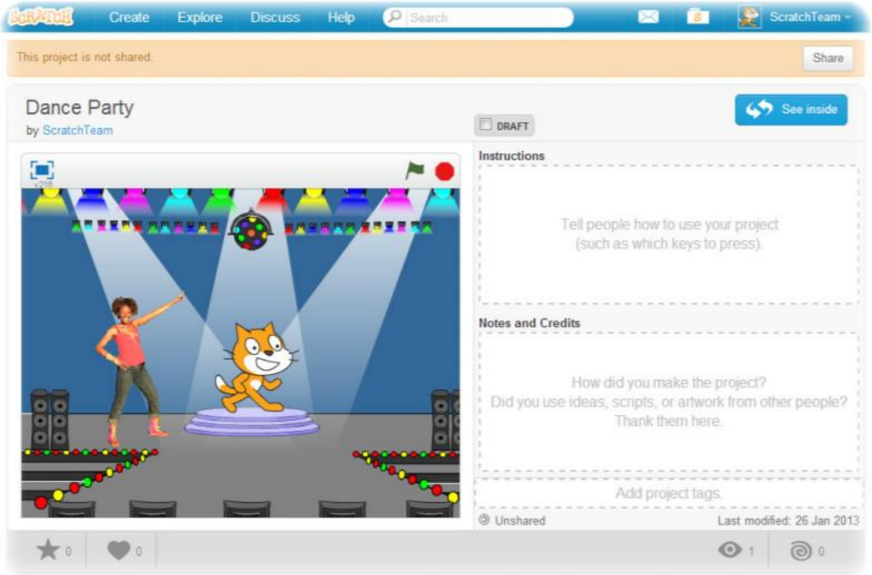
To save your project online, make sure to sign in. Sign in ▾

(If you want to save the file to your computer drive, click the **File** menu and choose "Download to your computer.")

When you are ready, click See project page

Project Page

Click ▣ for full screen viewing.



This project is not shared.

Dance Party
by ScratchTeam

Tell people how to use your project (such as which keys to press).

How did you make the project? Did you use ideas, scripts, or artwork from other people? Thank them here.

Add project tags.

Last modified: 26 Jan 2013

Click Share for others to see and play with your project.

Type in notes about your project.

Co-funded by the Erasmus+ Programme of the European Union

Getting started with Scratch – Save and share

When you share, others can visit and interact with your project.

Now what? You can Create a new project or Explore for ideas.

To find out more, click Help or go to http://scratch.mit.edu/help

Co-funded by the
Erasmus+ Programme
of the European Union

LEARNING OUTCOMES

- Learn about GPIO pins
- Lighting a LED
- Using a switchable pin
- Constructing a Scratch program
- Connecting a button
- Configuring a button
- Responding to a button press
- Controlling a LED with a button push
- Using buzzers

Co-funded by the
Erasmus+ Programme
of the European Union

Physical Computing With Scratch - Learn about GPIO pins

One powerful feature of the Raspberry Pi is the row of GPIO pins along the top edge of the board. GPIO stands for General-Purpose Input / Output. These pins are a physical interface between the Raspberry Pi and the outside world.

The GPIO pins allow the Raspberry Pi to control and monitor the outside world by being connected to electronic circuits. The Pi is able to control LEDs, turning them on or off, run motors, and many other things. It’s also able to detect whether a switch has been pressed, the temperature, and light. We refer to this as physical computing.

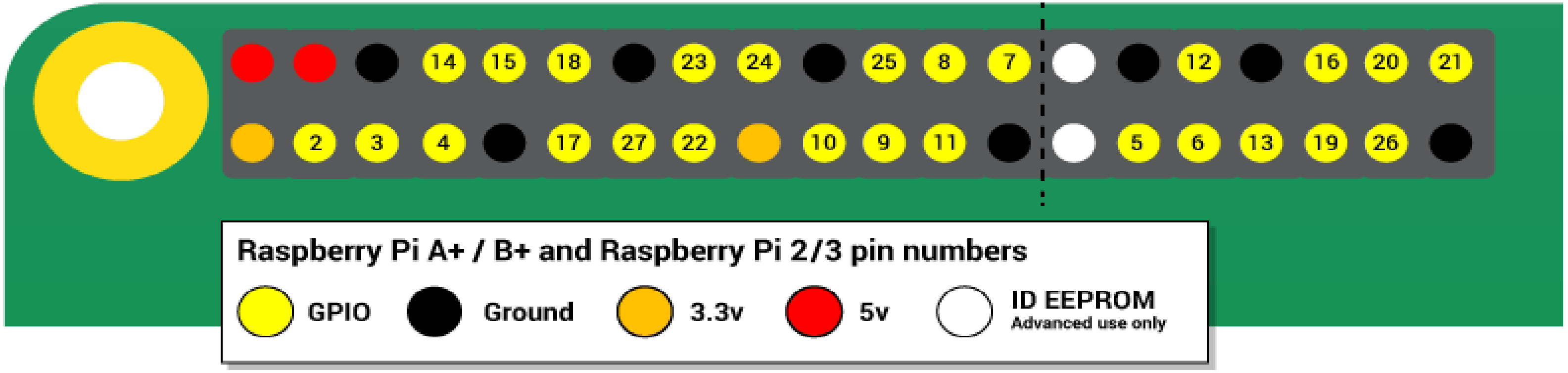
There are 40 pins on the Raspberry Pi, and they provide various different functions.

Co-funded by the
Erasmus+ Programme
of the European Union

MODULE 2 – Scratch 2.0

Physical Computing With Scratch - Learn about GPIO pins

If you don’t have a pin label, then this guide can help you to identify the pin numbers:

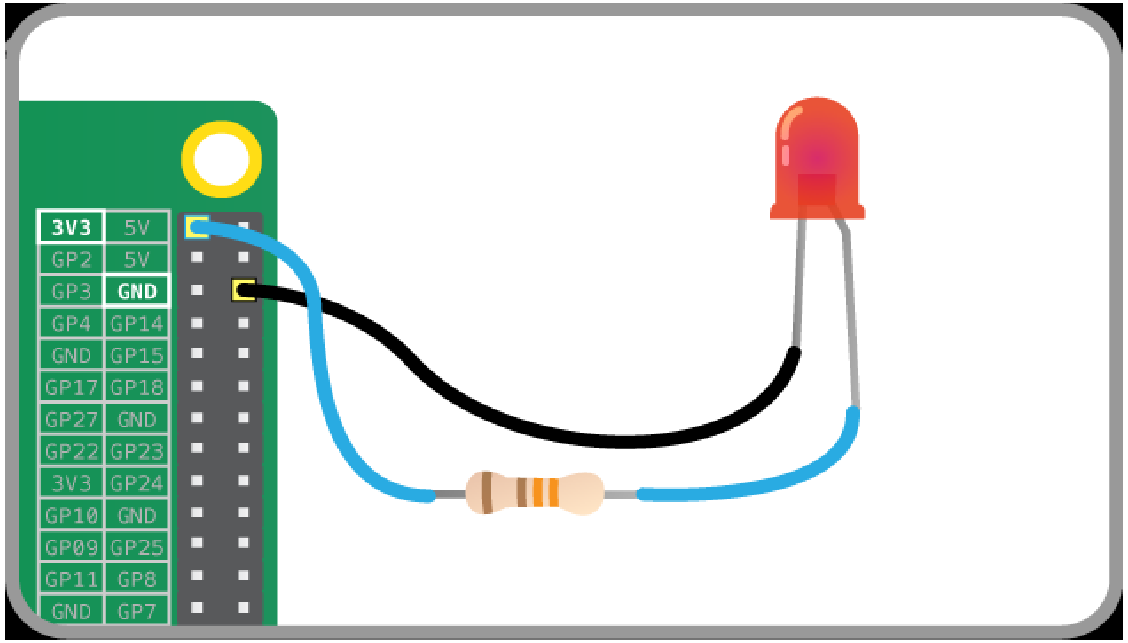


3V3	3.3 volts	Anything connected to these pins will always get 3.3V of power
5V	5 volts	Anything connected to these pins will always get 5V of power
GND	ground	Zero volts, used to complete a circuit
GP2	GPIO pin 2	These pins are for general-purpose use and can be configured as input or output pins
ID_SC/ID_SD/DNC	Special purpose pins	

Co-funded by the
Erasmus+ Programme
of the European Union

You can test whether your GPIO pins and LEDs are working by building the circuit below. You can use any resistor over about 50Ω.

- The LED is connected directly to the GND pin and the 3V3 pin via the 330 Ohm resistor, and should light up.
- Be sure to connect your LED the correct way round; the longer leg should be connected to the 3V3 pin:

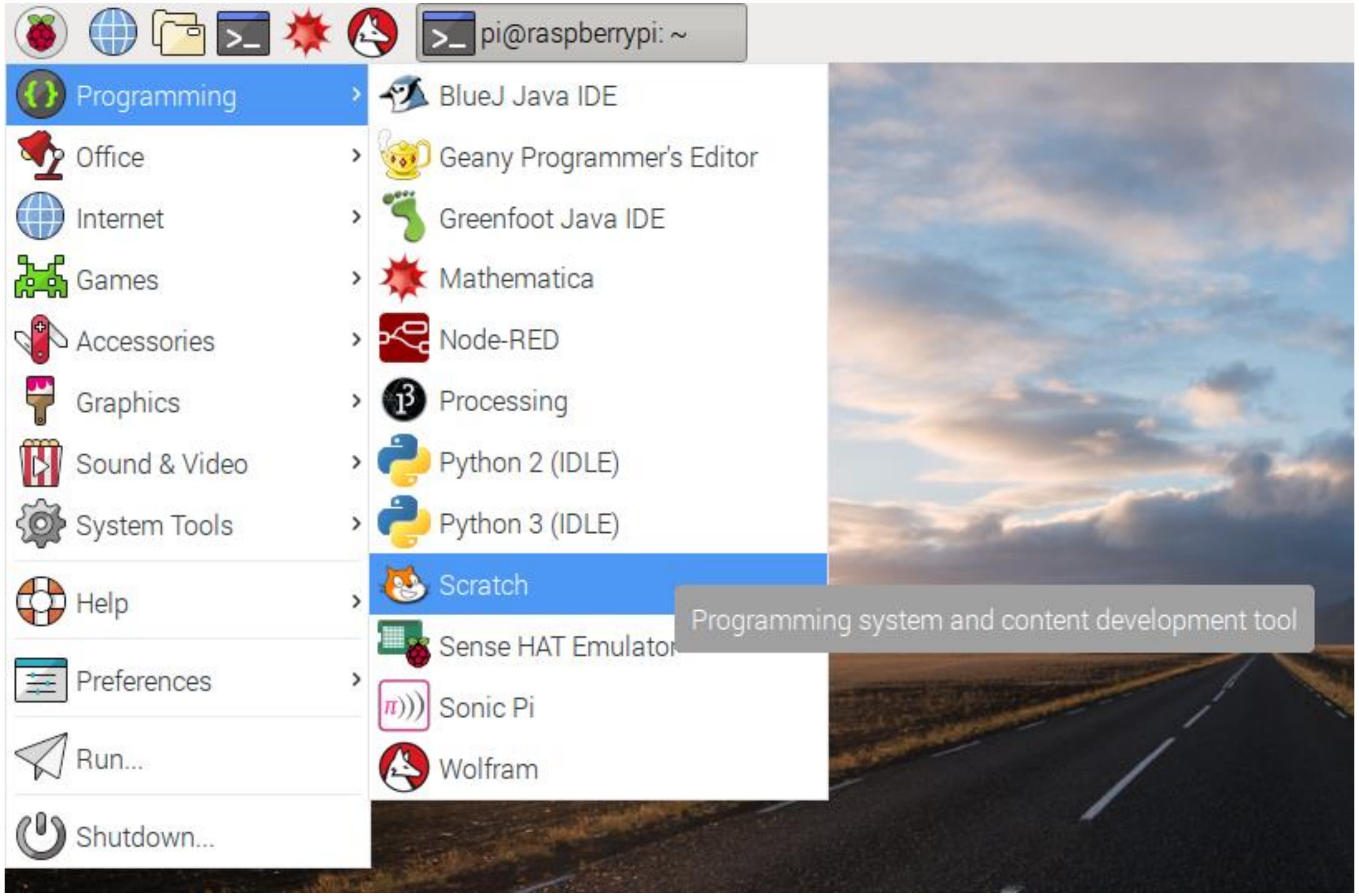


Co-funded by the Erasmus+ Programme of the European Union

MODULE 2 – Scratch 2.0

# Physical Computing With Scratch - Constructing a Scratch program

- Locate the Scratch program by clicking on Menu followed by Programming, and selecting Scratch.



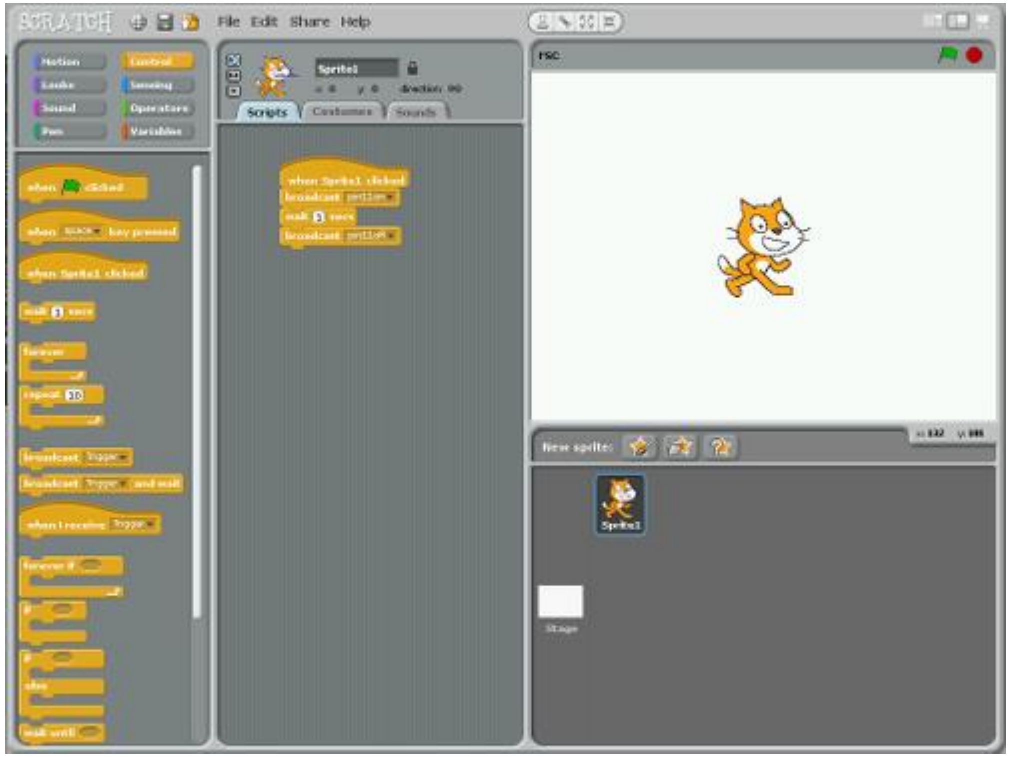
Programming system and content development tool

Co-funded by the
Erasmus+ Programme
of the European Union

MODULE 2 – Scratch 2.0

# Physical Computing With Scratch - Constructing a Scratch program

- The familiar Scratch interface will then load:



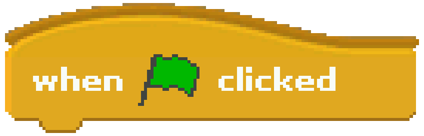
```
when Sprite1 clicked
broadcast [illegible]
wait [illegible] secs
broadcast [illegible]
```

Co-funded by the
Erasmus+ Programme
of the European Union

MODULE 2 – Scratch 2.0

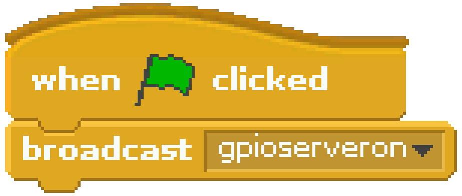
# Physical Computing With Scratch - Constructing a Scratch program

- Click on **Control** in the top-left display. Drag the **when GreenFlag clicked** block onto the scripts area:



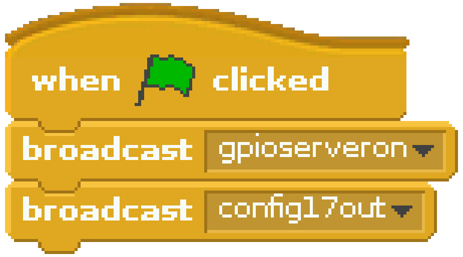
```
when ⚑ clicked
```

- Scratch uses **broadcast** blocks to communicate with the GPIO pins; the first broadcast you need is **gpioserveron** which activates the GPIO functionality:



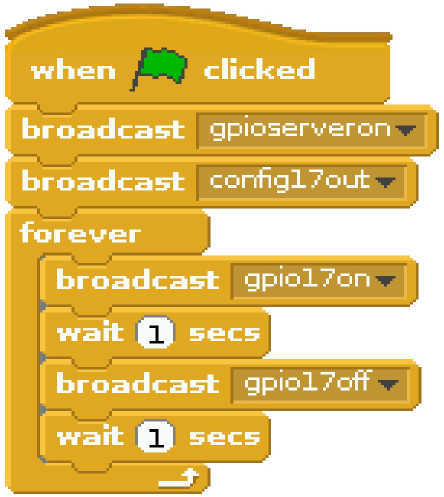
```
when ⚑ clicked
broadcast gpioserveron ▾
```

- As your GPIO pin can be used as either input or output, you'll need to specify in which mode your pin is being used with the **config17out** broadcast:



```
when ⚑ clicked
broadcast gpioserveron ▾
broadcast config17out ▾
```

- From this point on, you can control your LED using two broadcasts: **gpio17high** to turn it on and **gpio17low** to turn it off. Using these two messages and some pauses, you can make an LED flash continuously:

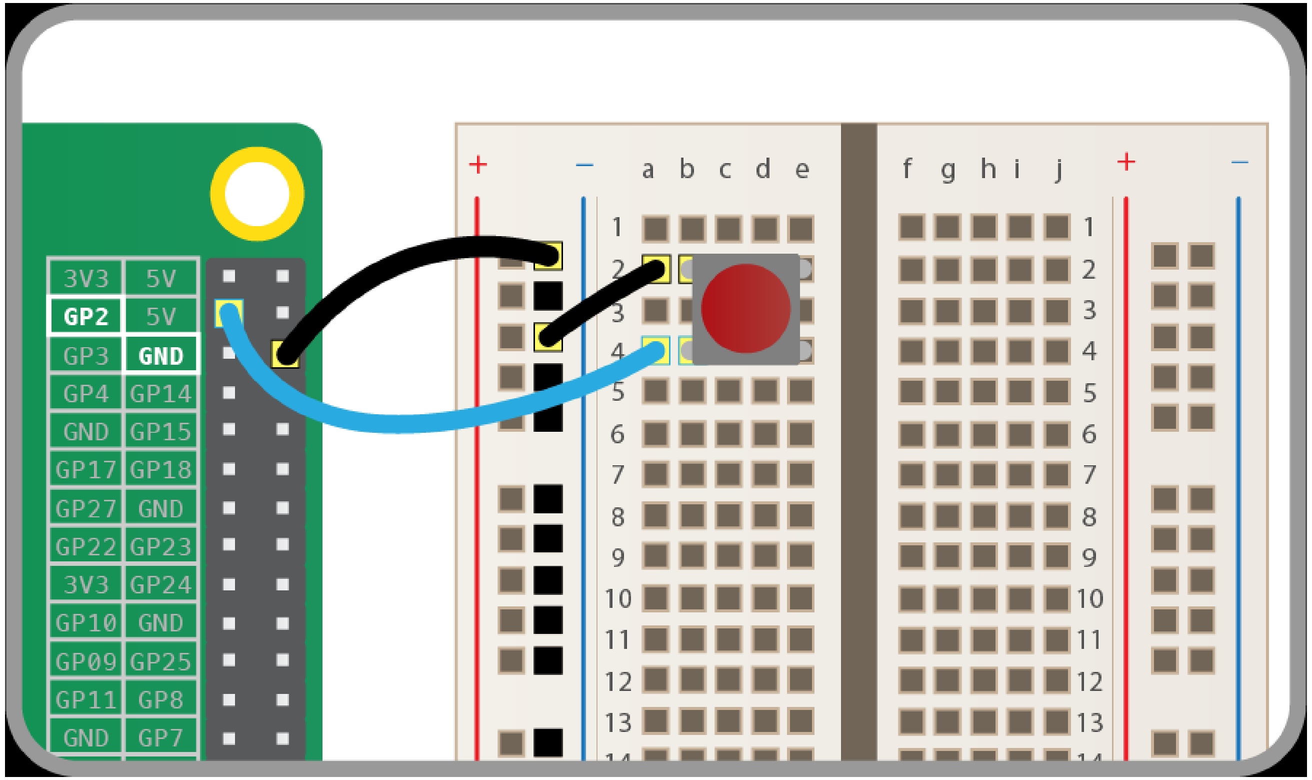


```
when ⚑ clicked
broadcast gpioserveron ▾
broadcast config17out ▾
forever
    broadcast gpio17on ▾
    wait 1 secs
    broadcast gpio17off ▾
    wait 1 secs
```

Co-funded by the
Erasmus+ Programme
of the European Union

# Physical Computing With Scratch - Connecting a button

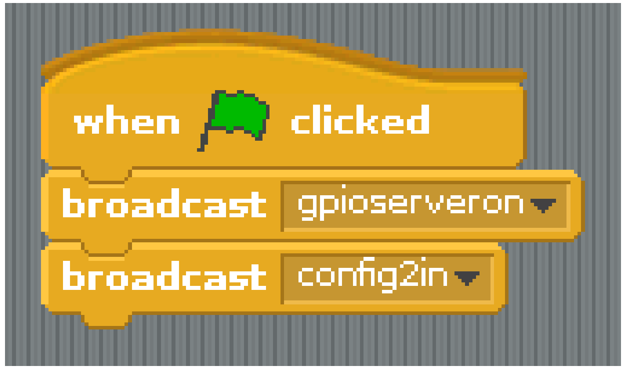
- As well as controlling the physical world, you can react to it using an input device such as a button.
- Connect your button to a breadboard, then connect one pin to a ground pin and the other to a numbered GPIO pin. In this example pin 2 has been used:



Co-funded by the
Erasmus+ Programme
of the European Union

# Physical Computing With Scratch – Configuring your button

- Before Scratch can react to your button, it needs to be told which pin is configured as an input pin.
- Assuming you have started a new Scratch file, you'll also need to start the GPIO server. The following code will configure pin 4 as an input:



```
when [green flag] clicked
broadcast gpioserveron
broadcast config2in
```

- Once you have built the code above, you need to click the green flag in order for it to run and for your pin to be set up.

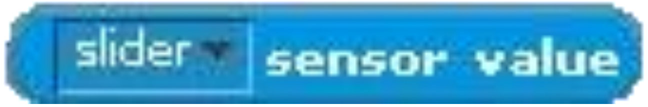
Co-funded by the
Erasmus+ Programme
of the European Union

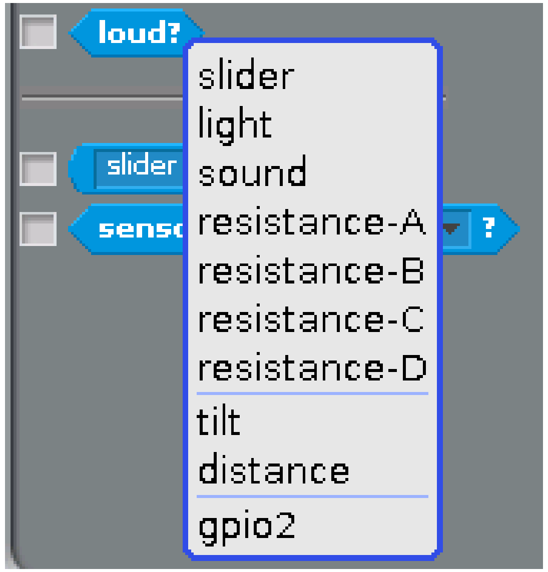
MODULE 2 – Scratch 2.0

# Physical Computing With Scratch – Configuring your button

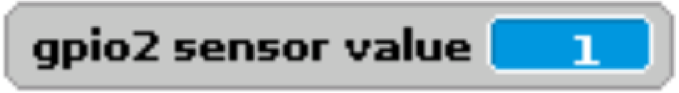
- Next, you need to go to the **Sensing** menu in Scratch:

- From here you need to find the **slider sensor value** block and click the triangle to reveal a menu.
  Select **gpio2** from the menu and click the tickbox to the left:





- You should now see the current state of the pin in the stage area:

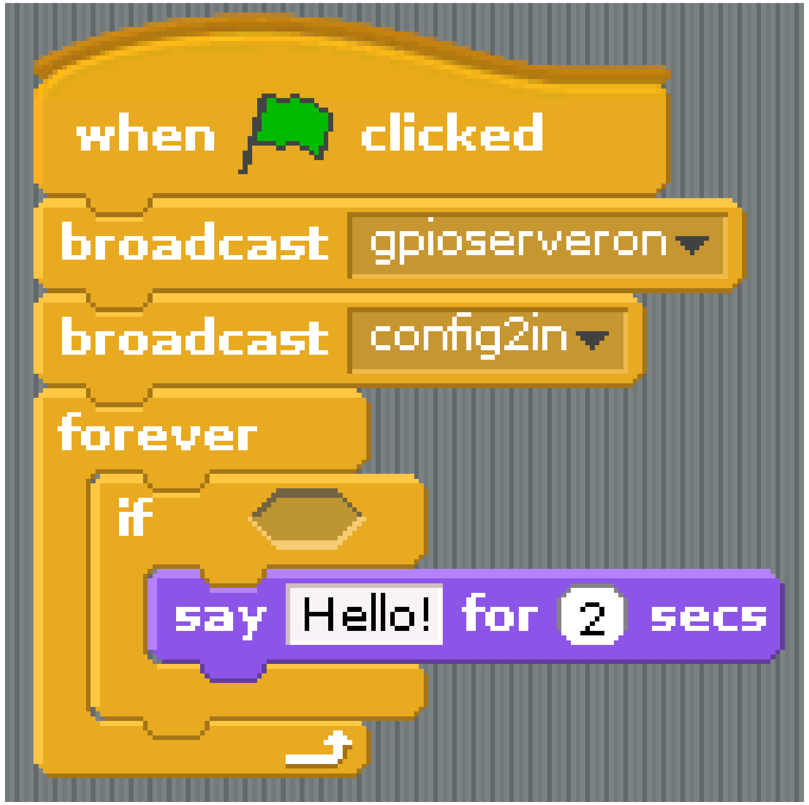


- Now when you press your button, the state should change from 1 to 0.

Co-funded by the
Erasmus+ Programme
of the European Union

- Now that your button is all set up and working, you can make it do something. You can start off by making it control a sprite.

- Begin with a **forever** loop with an **if** block inside it. This will continually check the **if** condition and perform some action if the condition is met. The action in the example below will make the current sprite say “Hello!”:

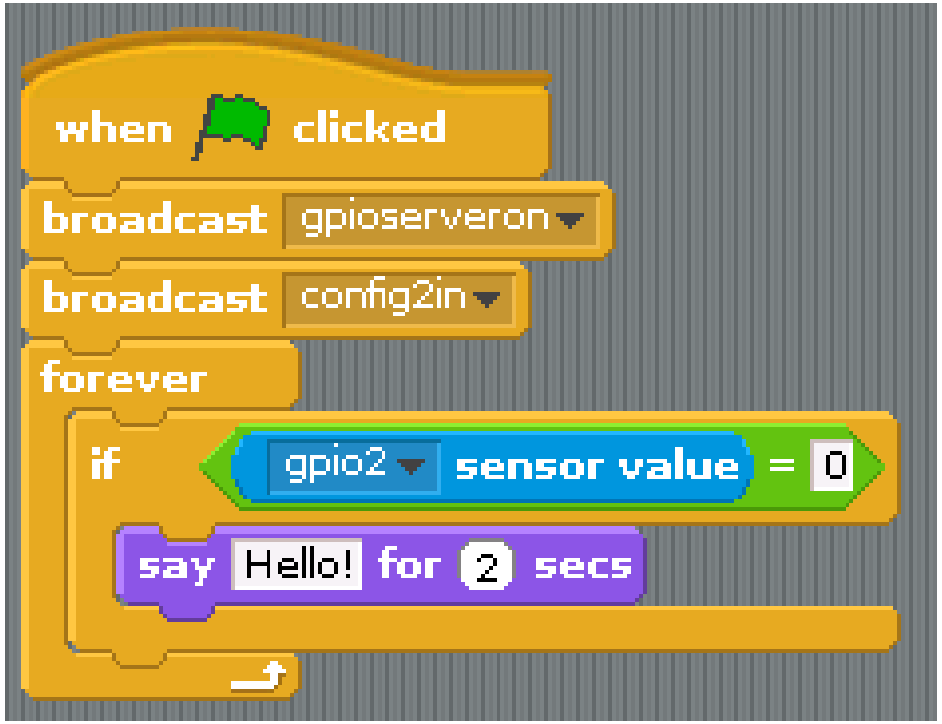


```
when ⚑ clicked
broadcast gpioserveron ▾
broadcast config2in ▾
forever
    if < >
        say Hello! for 2 secs
```

Co-funded by the
Erasmus+ Programme
of the European Union

Physical Computing With Scratch – Responding to a button press

- Finally, to make this work you need to add the condition, which is that we want the sprite to speak when the **button value = 0:**



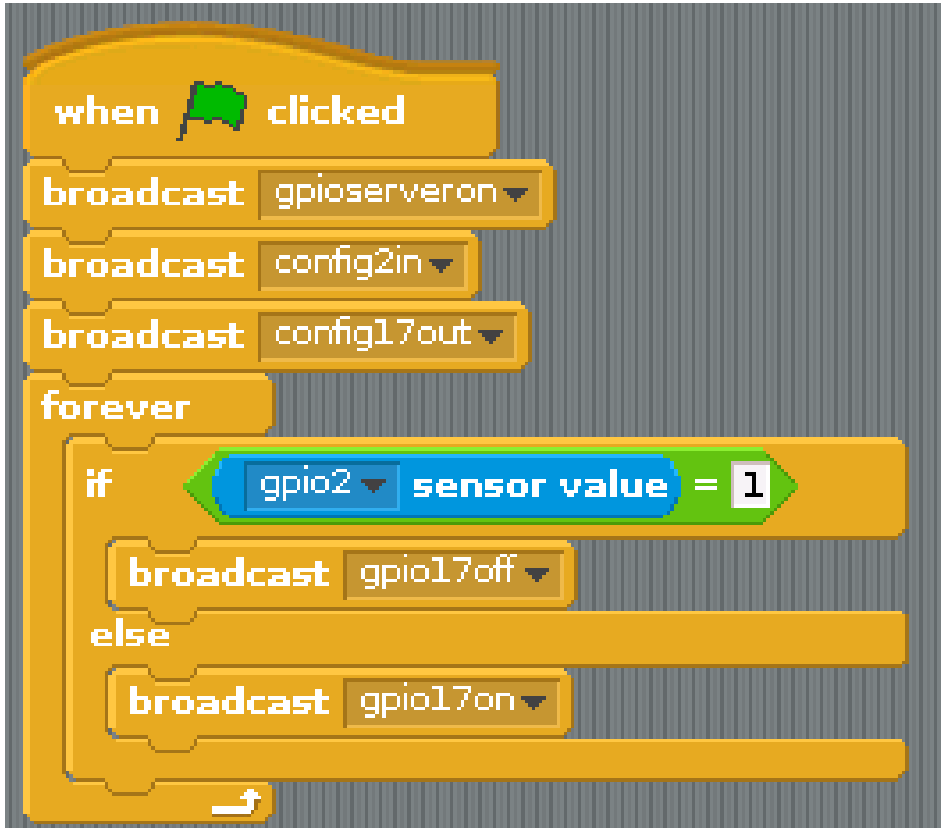
```
when [flag] clicked
broadcast gpioserveron
broadcast config2in
forever
    if gpio2 sensor value = 0
        say Hello! for 2 secs
```

- If everything is correct, your button should make the sprite speak.

Co-funded by the
Erasmus+ Programme
of the European Union

To finish off, you can combine your two programs so that the button can turn the LED on and off.

- Adapt your script and use an **If Else** block so that it looks like the example below:



```
when [flag] clicked
broadcast gpioserveron
broadcast config2in
broadcast config17out
forever
    if <(gpio2 sensor value) = 1>
        broadcast gpio17off
    else
        broadcast gpio17on
```

- Now when you push the button, the LED should light up.

Co-funded by the Erasmus+ Programme of the European Union

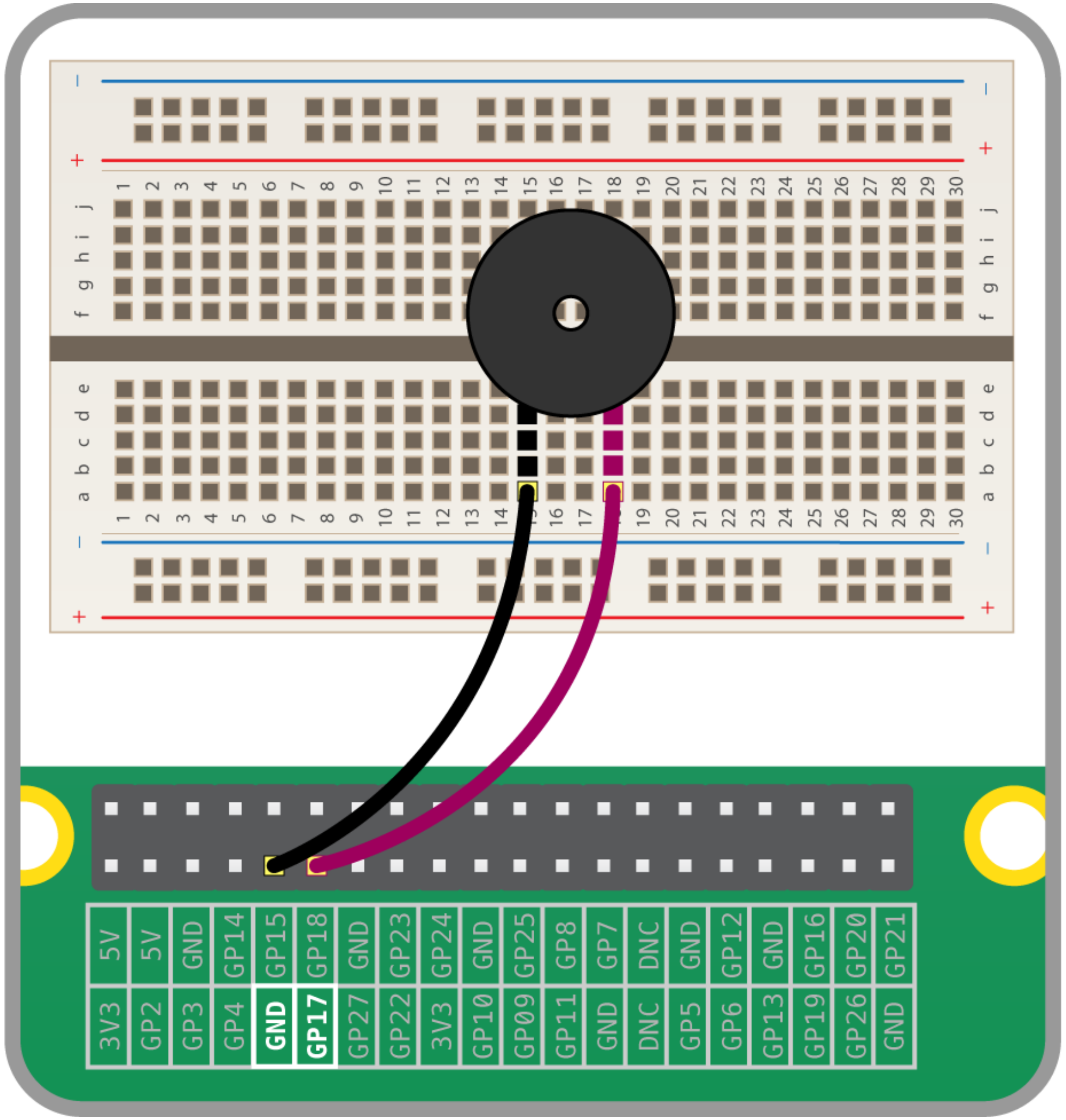
- There are two main types of buzzer: **active** and **passive**.

- A passive buzzer emits a tone when a voltage is applied across it. It also requires a specific signal to generate a variety of tones. The active buzzers are a lot simpler to use, so these are covered here.

Co-funded by the
Erasmus+ Programme
of the European Union

Connecting a buzzer

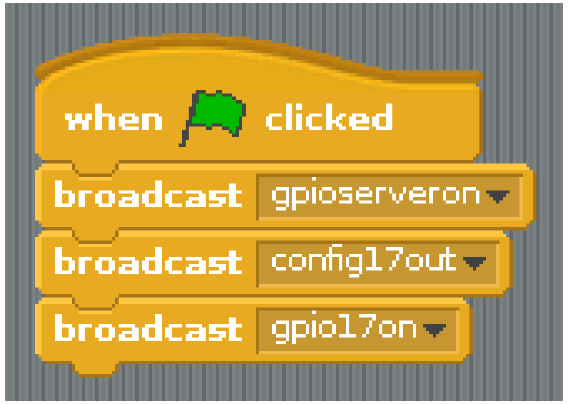
- An active buzzer can be connected just like an LED, but as they are a little more robust, you won't be needing a resistor to protect them.
- Set up the circuit as shown below:



Co-funded by the
Erasmus+ Programme
of the European Union

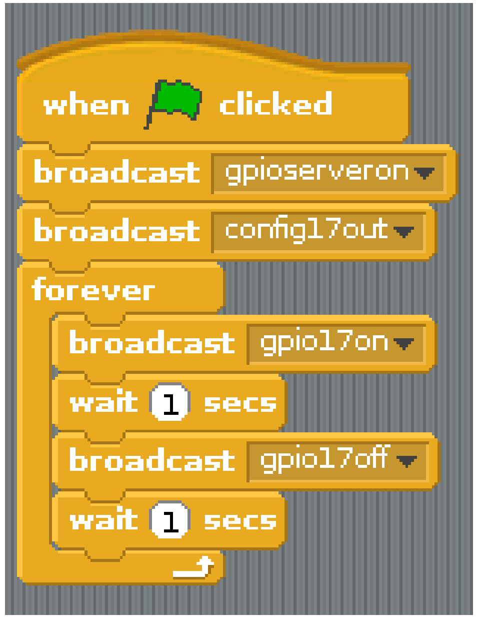
Coding a buzzer

- Now you can sound the buzzer by using the code below:



```
when [flag] clicked
broadcast gpioserveron
broadcast config17out
broadcast gpio17on
```

- Or even make the buzzer beep:



```
when [flag] clicked
broadcast gpioserveron
broadcast config17out
forever
    broadcast gpio17on
    wait 1 secs
    broadcast gpio17off
    wait 1 secs
```

Co-funded by the
Erasmus+ Programme
of the European Union

CONCLUSION

If you followed this resource, you are expected to:

- To program using the basic commands in Scratch
- Have learned about GPIO pins
- Be able to light a LED
- Use a switchable pin
- Construct a Scratch program
- Connect a button
- Configure a button
- Respond to a button press
- Control a LED with a button push
- Use buzzers

Co-funded by the
Erasmus+ Programme
of the European Union

REFERENCES

- https://cdn.scratch.mit.edu/scratchr2/static/__709da8e5f3d72129538a4ccdbcbf5f2a__/pdfs/help/Getting-Started-Guide-Scratch2.pdf
- https://projects.raspberrypi.org/en/projects/physical-computing-with-scratch/10

Co-funded by the
Erasmus+ Programme
of the European Union

EXTRA RESOURCES

- https://projects.raspberrypi.org/en/projects/create-your-own-world-scratch2/1
- https://projects.raspberrypi.org/en/projects/lost-in-space-scratch2/1
- https://projects.raspberrypi.org/en/projects/ghostbusters-scratch2
- https://projects.raspberrypi.org/en/projects/chatbot-scratch2

Co-funded by the
Erasmus+ Programme
of the European Union

MODULE 2 – Scratch 2.0

GLOSSARY

Term	Description
Interface design	When we design a program, we turn our imagination into a creation that can be shared with others. We create the flow of the program, how the user interacts with the project, and the actions each sprite takes to tell our story.
Loops (iteration)	A loop repeats (iterates) through a list of programming commands (also known as blocks in Scratch). Often, we'll use conditional statements to control when and how often a loop runs.
Boolean logic	A Boolean command evaluates a given statement as true or false. In Scratch, a Boolean command can check whether a specified condition is true (for example, is the color blue?), or we can compare values with and, or, and not operators. For example, if 4 > 0 and 4 < 2.
Variables	Variables store text or numbers for reuse in the program. For example, if x > 0 creates a conditional statement that evaluates whether the number assigned to x is greater than 0.

Co-funded by the
Erasmus+ Programme
of the European Union

GLOSSARY

Term	Description
Arrays (Lists)	Arrays are similar to variables in that they store information that may or may not change. However, a list stores multiple values in the same way a grocery list stores a group of items.
Events	Scratch provides an entire group of event blocks that allows us to tell our program what to do when that event happens. For example, events include when flag is clicked or when a space key is pressed.
Synchronization and coordination	Programming a sprite to receive a broadcast message from another sprite coordinates a cause and effect. Broadcasting a message and waiting for all the other sprites to act on the broadcast synchronizes the action. Throughout the book, broadcasts are a technique we will use often, and they provide the fundamental communication between the sprites in the project.
Concurrency	Creating two scripts to run on the same control enables parallel execution. For example, programming four different sprites to pixelate when the green flag is clicked creates four concurrent actions.

Co-funde
Erasmus+ Programme
of the European Union

GLOSSARY

Term	Description
Random numbers	This concept picks a random number from a specified range.
Cloud data	Scratch 2.0 introduces cloud variables that enable projects to store data on the Scratch web servers so that the data is available to other Scratch users. For example, the use of cloud data might include keeping a high score or tracking the survey results.
Procedures	Procedures can also be called as functions or methods in other programming languages. Scratch 2.0 adds the ability to create custom blocks that allows you to create a stack of blocks under a single name. When you use a custom block, you can pass an argument, such as a sprite number, into the procedure.
Vector and Bitmap graphics	Scratch includes a built-in image editor that enables you to create graphics and sprites for your projects. Vector graphics is a new feature of Scratch 2.0.
Cloning	New to Scratch 2.0, cloning allows a sprite to duplicate itself while the program is running. Clones inherit the parent sprite's costumes and scripts. For example, many people create games that need to shoot something, such as asteroids. Cloning in Scratch 2.0 allows us to shoot multiple times.

Co-funded by the
Erasmus+ Programme
of the European Union

Erasmus+

2Play2Learn

Reviving hands-on educational play for learning skills of tomorrow

PROJECT N° 2019-1-UK01-KA201-061466

CIVIC idec ΕΚΠΑΙΔΕΥΤΗΡΙΑ ΠΛΑΤΩΝ dlearn EUROPEAN DIGITAL LEARNING NETWORK scholé Emphasys CENTRE CCS Digital Education