salesianos
EL PILAR - SOTO DEL REAL

TECNOLOGÍA 1ª ESO

EJERCICIOS DE RECUPERACIÓN VERANO

Practical 1.

STARTING OUT WITH SCRATCH. PRACTISING WITH TUTORIALS AND EDITING SPRITES AND STAGES

When we open the online version of Scratch, a window pops up in the scripts area, inviting us to watch the first video tutorial. But, first of all, we will change the language of the application to English: in the menu bar, click on the language icon (the globe) and select **English**.

Watch the **Getting Started** tutorial (Figure 1). If you cannot see this tutorial in the scripts area, click on the **Tutorials** icon (Figure 2) on the menu bar. After watching the video, click on the **arrow** on the right (Figure 3) to see how we add the first instructions.

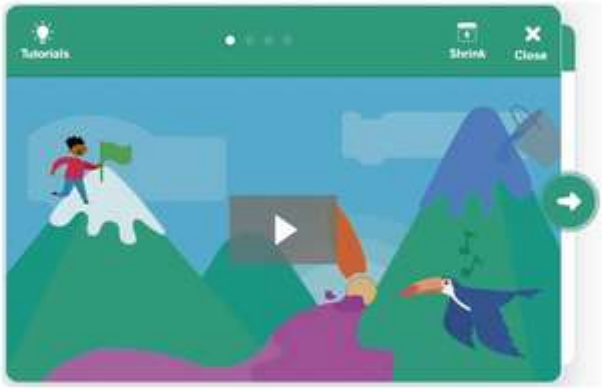
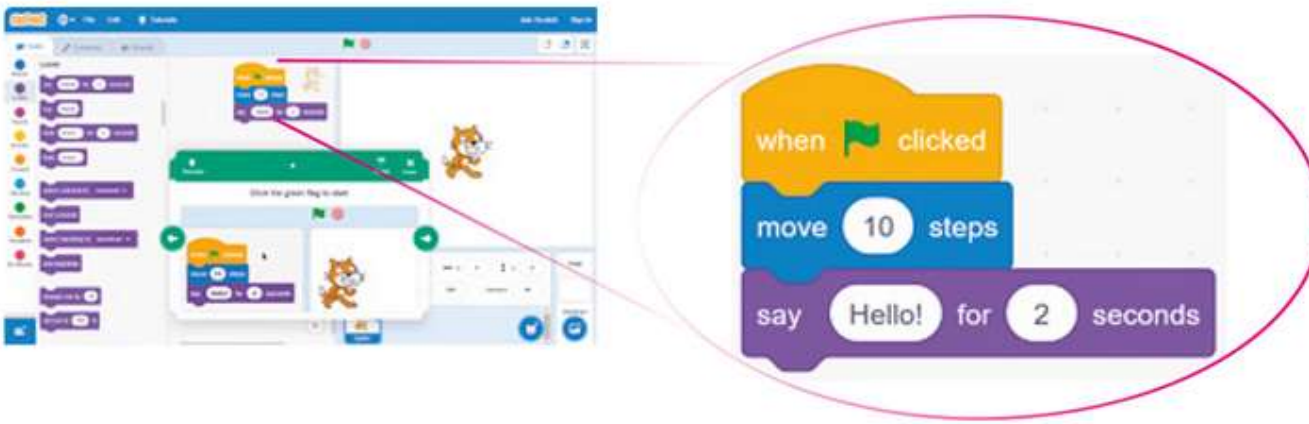


Fig. 1

Fig. 2

Fig. 3

Without closing the tutorial window, drag the blocks of instructions that you see in the animation (Figure 5) to the scripts area (Figure 4). Now click on the green flag to run your first program.



As you have seen, we only drag the blocks of instructions from the relevant category. You can identify each category

(**Motion**, **Looks**, etc.) with a colour to help you find the instruction you need.

EDITING SPRITES AND STAGES

We will put all the sprites of the program in the window in the bottom right-hand corner: stages, characters, etc. In the **Choose a backdrop** section (in the **Stage** window), click on the magnifying glass to add a new background from the library and select **Theater**, for example (Figure 6)

Fig. 6

For the character, **Objeto1** (**Cat**) is displayed by default. To add new characters, click on **Choose a sprite**. Select **Dog2**, for example (Figure 7).

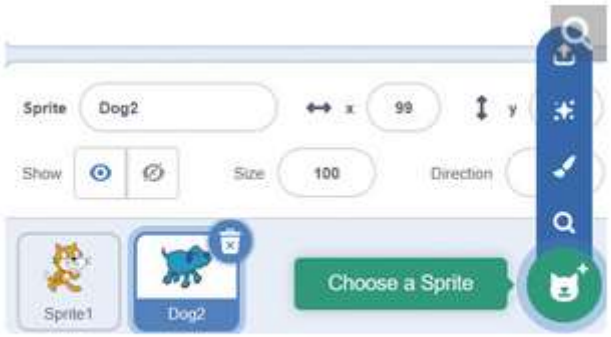


Fig. 7

We can easily edit sprites: first select the sprite, in this case **Objeto1** (**Cat**), and then go to the **Costumes** tab (in the top left). There are two costumes for this sprite. Select **disfraz1** (Figure 8).

Fig. 8

The tools for editing the sprite are on the left of the working window (Figure 9). Click on **Select** and then hover your cursor over the sprite. You will see all the shapes that form the sprite (click on **Ungroup** in the top menu to break down the sprite into smaller parts). Select a shape, then use the **Fill** and **Outline** options to change the colours of the sprite.

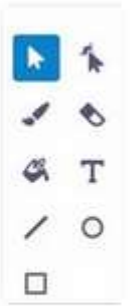


Fig. 9

Now select **Reshape** (the second button). Click on the image. Some points will appear on it. If you drag these, you can change the shape of the character. Drag the points to make the cat's tail and ears bigger (see Figure 10). To do this more precisely, use the zoom buttons in the bottom right corner.

Fig. 10

We are going to add a new character now.

Search the internet for the image of a FROG

Importing a character

We can Import a character that we created before or that we downloaded from the Internet. These characters may have an Image format (their extension will be **.jpg**, **.jpeg**, **.png** or **.gif**) or they could be characters that we saved from other Scratch projects (these have the extension **.sprite2** or **.sprite3**).

Fig. 12

Fig. 13

Sprites have sounds associated with them. Select the **Dog2** sprite, go to the **Sounds** tab and press play; you will hear a bark. Find out which is the cat's.

Fig. 14

You have to end the program by making the following changes

Modify the program:

1. Draw a new character with two costumes. Add colours and a sound.

Save the program File name:

ApellidoNombre-TPR1EB-practical 1

Practical 2.

Blocks in the Events and Looks menus. Processing

Creating the interface

Delete the default character by clicking on the trash icon that is displayed on it.

Go to **Choose a backdrop** (Figure 15) and select **Jurassic**.

Fig. 15

Now select the characters. Go to **Choose a sprite** (Figure 16) and select **Dinosaur1**.

Fig. 16

Starting a program. Changing looks

In Scratch, we can start programs in different ways. You are going to make two programs with different starts for your character.

Characters can have different costumes; in other words, they can appear in different positions. Select the **Costumes** tab and check that your character has four different costumes. This lets us create more interesting programs.

Select the **when green flag clicked** block in the **Events** category and drag it to the scripts area.

Now select the **say ... for ... seconds** block from the **Looks** category and include the text "Hello!" for **2** seconds (you can change the text and the number of seconds). Stack **switch costume to ...** from the same category with this block. In the dropdown menu, select **dinosaur1-c** (Figure 17).

Fig. 17

Now add the **change ... effect by ...** block from the **Looks** category, select **color** from the dropdown menu and enter **50**. Check that your program looks like the one in Figure 18.

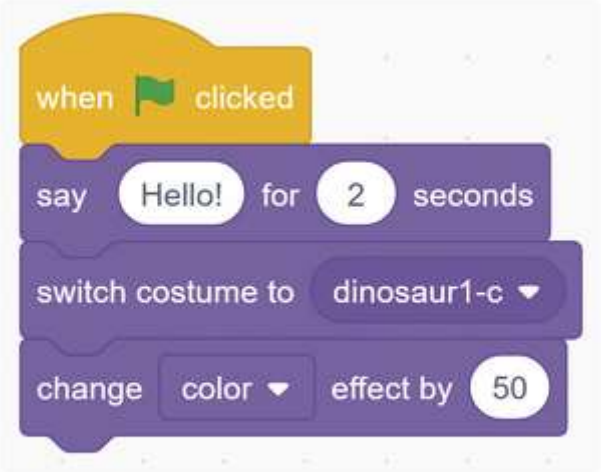


Fig. 18

Test the program by clicking on the green flag. Your sprite should change colour and costume.

Now add the **hide** block from the **Looks** category. Test the program.

Your character has now disappeared and you do not know how to show him again. This is why it is always important to create another set of instructions to return sprites to their original status.

To show the sprite again, delete the **hide** block and replace it with the **show** block from the **Looks** category. Click on the green flag.

Let's restore the sprite to how it was originally. To do this, select the **when this sprite clicked** block from the **Events** category.

Now include the **clear graphic effects** block from the **Looks** category. Then stack the **switch costume to ...** and **think ... for ... seconds** blocks from this category.

Starting a program

Practically all of the blocks in the **Events** category have a special form: blocks can only be stacked under them. These are the only blocks that we can use to start a program in Scratch.

In the switch costume to ... block, select **dinosaur1-a**. In the think block, include the text "Here I am" for **2** seconds (Figure 19).

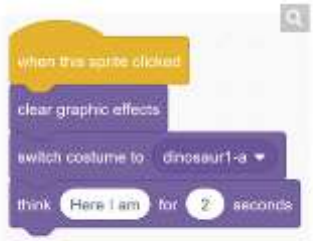


Fig. 19

The "show" and "hide" blocks

We can use the **show** and hide blocks in the **Looks** category to create fun games. To use the **hide** block, you must create an auxiliary program to show the sprite again when it disappears.

Simulating movement

To simulate movement, start the program by dragging the **when space key pressed** block, from the **Events** category, to the scripts area.

Add the **switch costume to ...** block, from the **Looks** category, under the last block, and select **dinosaur1-b** from the dropdown menu.

Then stack the **wait ... seconds** block, from the **Control** category, and set a time of **0.5** seconds.

Repeat steps 2 and 3 three times. You should have nine blocks stacked together.

Modify the **switch costume to ...** instructions by selecting the different costumes of your sprite in this order: **dinosaur1-b**, **dinosaur1-c**, **dinosaur1-d** and **dinosaur1-a** (Figure 20).

Fig. 20

"Walking" sprites

As you have seen in this practical, we can use sprite costumes and the **wait** instruction to simulate movement on screen.

salesianos
EL PILAR - SOTO DEL REAL

You have to end the program by making the following changes

Modify the program:

Modify the program using the blocks in the **Events** menu. Activate the program when the backdrop changes.

1.- Choose a sprite with at least three different costumes and simulate its movement in a program.

2.- imports from the internet a character of a **DoG and a CAT**

These characters have to be an image format (their extension will be **.jpg**, **.jpeg**, **.png** or **.gif**).

3.- Make a conversation between the two sprites.

Save the program File:

ApellidoNombre-TPR1EC-practical 2

salesianos
EL PILAR - SOTO DEL REAL

Practical 3.

Blocks in the Events and Looks menus. Processing

Creating the interface

We are going to use the **Xy-grid** backdrop.

This is very useful when we need to see the coordinates so that we can move our characters along them.

To do this, go to **Choose a backdrop** and select the **Xy-grid** backdrop (Figure 21).

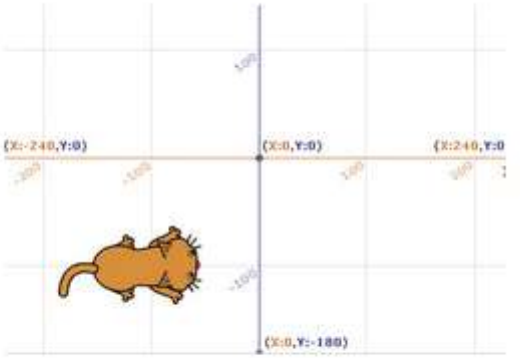


Fig. 21

To select the character, go to **Choose a sprite** and select **Cat2**.

Knowing the position and direction of a sprite

- Look at the indications underneath the backdrop on the screen.

You should see something similar to what is shown in Figure 22.

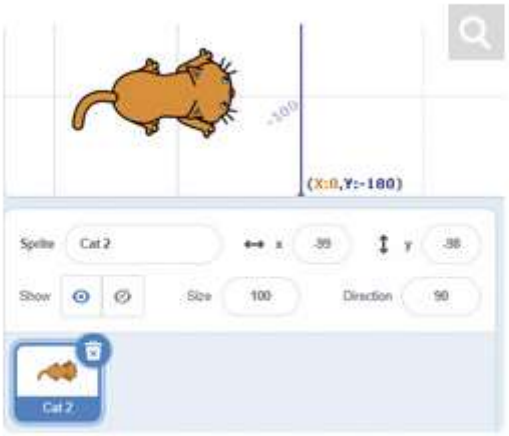


Fig. 22

The sections *x* and *y* indicate the position of the cat

(in the figure, $x = -99$ and $y = -98$).

In Scratch, the values of the *x* axis go from –240 to +240 and the values of the *y* axis go from –180 to +180.

- To determine the direction of your sprite, look at the **Direction** indicator. In our case, it is 90.

  Click on **Direction** and watch how the sprite changes direction when you modify the value.

- Now change the direction of the cat back to 90.

Moving and tracing a square

- The cat will trace a square with sides of 200 and tell you when it reaches its destination.

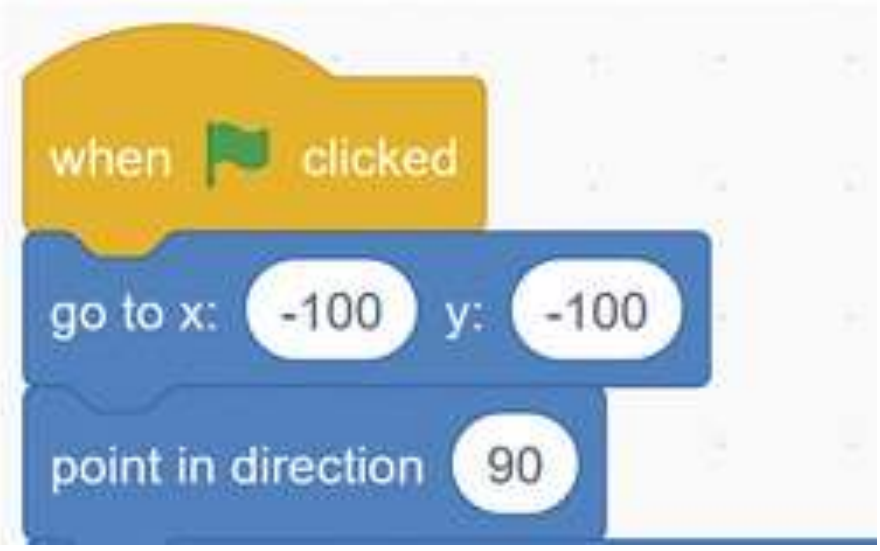
  It will move from the coordinates (–100, –100) and return to the same spot.

  The program starts with the **when green flag clicked** block, from the **Events** category.

- To position the cat at the coordinates (–100, –100).

- Turn it to the right, use the **go to x: –100 y: –100** block of the **Motion** category.

- Then select **90** in the **point in direction ...** block of the **Motion** category.



- In the **glide ...** block of the **Motion** category, enter **4** seconds and the coordinates x = 100 and y = –100.

- The cat will now be at the first corner of the square.

- To continue tracing the square, select the **point in direction ...** block and enter **0** (Figure 23).

- Now the cat will be facing up and we can continue drawing the figure.

- Look at Figure 24. It contains the square that the cat must fit inside

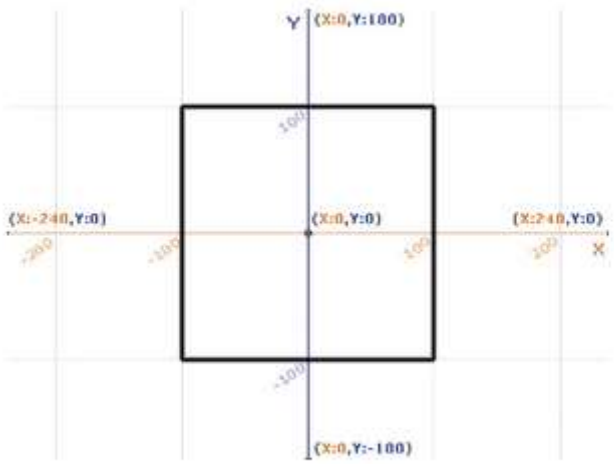


Fig. 24

- To draw, first we click on **Add extension** and add the **Pen** extension (Figure 38).

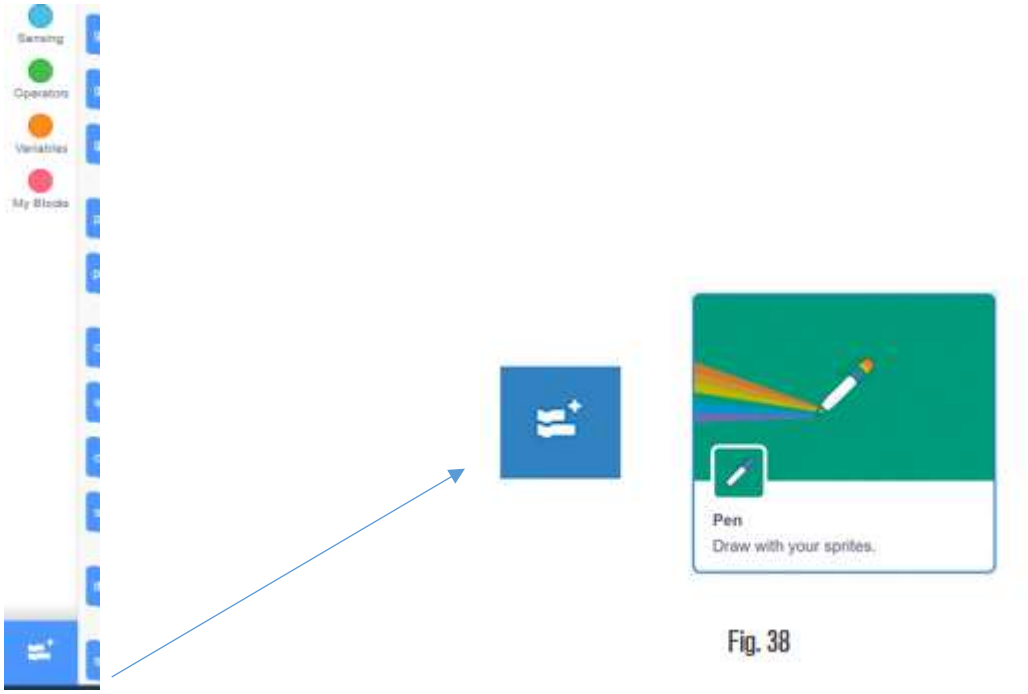
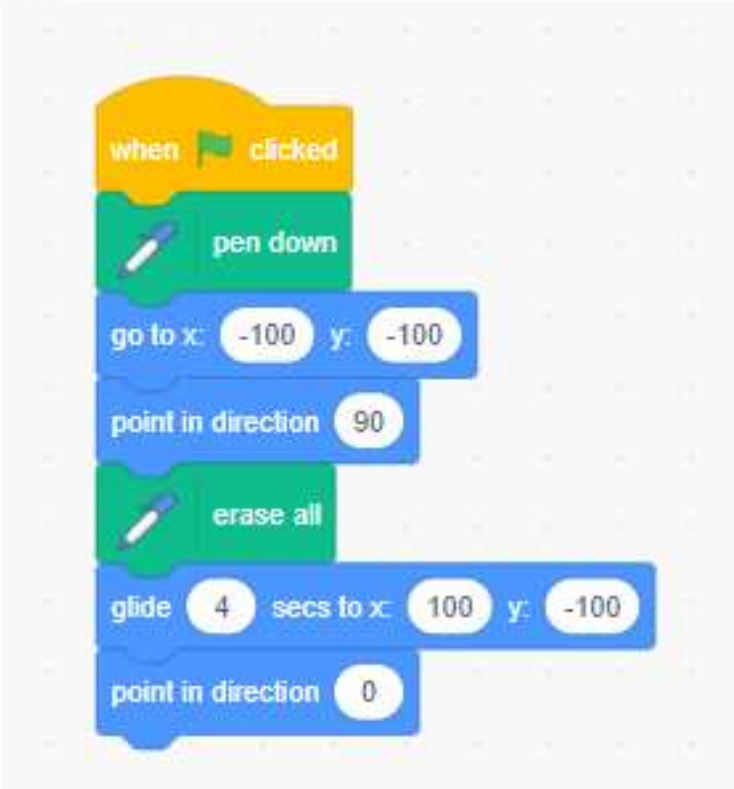


Fig. 38

- The instruction blocks panel contains all the instructions for drawing and changing the colour and size of our strokes, etc.

- Then **pen down** TO DRAW
- We should always clear the screen before we start drawing, so first select the **erase all** block from the **Pen** category. It is better when you start your square



- The table gives the coordinates of the vertices of this square and the direction of the sprite at each point.

- Use the block GLIDE and wait 4 seconds between each point. Select the **point in direction** the line

- Make sure that you understand which positions these coordinates refer to.

salesianos
EL PILAR - SOTO DEL REAL

	Grid reference	Direction
Starting position	X= -100, Y= -100	90
Vertex 1	X= 100, Y= -100	0
Vertex 2	X= 100, Y= 100	-90
Vertex 3	X= -100, Y= 100	180
Vertex 4	X= -100, Y= -100	90

- Complete the square with the data from the table

The "point" and "glide" blocks

The **point** and **glide** instructions in the **Motion** category work differently. The first one, **point**, only indicates the direction that the sprite will point in; it does not change the position of the sprite. The second one, **glide**, changes the position of the sprite by moving it along a straight line from its initial position to where we move it.

You have to end the program by making the following changes

Modify the program:

Modify the program by changing the color and thickness of the line.

salesianos
EL PILAR - SOTO DEL REAL

Change the character so that your character draws a **hexagon.**

Save the program File name:

ApellidoNombre-TPR1EB-practical 3

Practical 4.

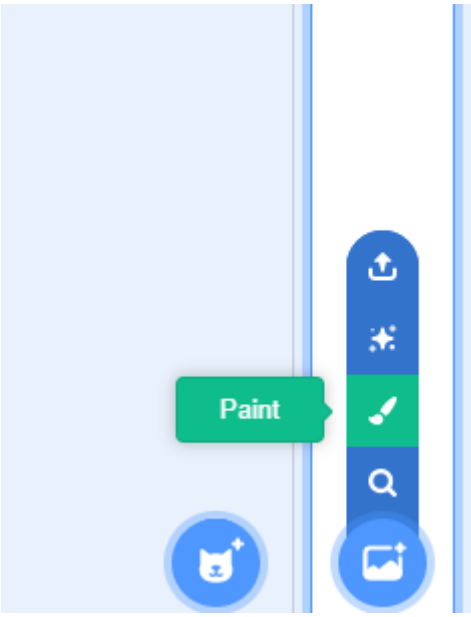
Blocks in the Sensing menu.

The ballerina and the knight

Creating the interface

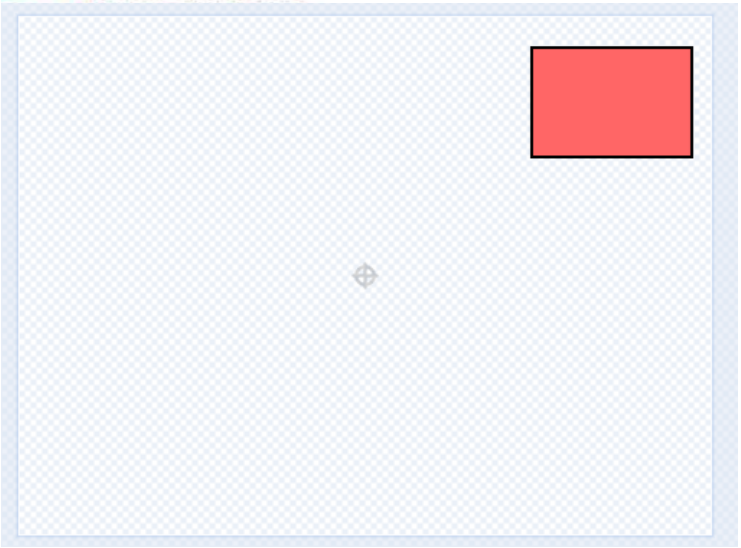
We are going to create the backdrop.

Go to **Choose a backdrop** and select the **Paint** option.



Use the **Rectangle** tool to draw a rectangle.

Select red from the **Fill** dropdown menu and colour in the rectangle.



Go to **Choose a sprite** and select **Ballerina**.

Interacting with the user

When the user presses the space key, the ballerina will move and then follow the mouse pointer. Finally, she will fall over when she is hit by a sprite in the backdrop (in this case, the red rectangle that you coloured in before).

Start the program with the when space key pressed block from the EVENTS category.

Now we will position the sprite and select its starting costume.

Choose **switch costume to ...** in the LOOKS category and select *ballerina-a.*

Choose **point in direction** ... in the MOTION category and write *90*.

This way, every time we press the space key, the sprite will be in the vertical position and be wearing the ballerina-a costume (Figure 29).

To tell the sprite to follow the mouse pointer, select the **go to** ... block in the MOTION category and select *mouse pointer*.

Now go to the CONTROL category to include the **if ... then block.**

Then, in the SENSING category, choose the **color ... is touching ...?** block and slide it inside the **if ... then** block in the space between the two words (Figure 29).

Click on one of the colours in the **color ... is touching ...?** block.

Select the eyedropper and click on the character's dress to select the colour of her dress.

Do the same with the other colour in the block. This time, select the same red colour as the rectangle in the backdrop.

In the LOOKS block, stack the **switch costume to** ... block (from the LOOKS category) and select ballerina-c. Now include the point in direction *180* and **think That hurts! for 1 seconds** blocks .

Interaction between sprites

We will now add a new sprite to the last program. Go to Choose a sprite and select Knight.

Duplicate the previous program.

Change the first block to **when the a is clicked** block, in the EVENTS category.

Change the **color ... is touching ...?** block to in the SENSING category, select the **touching ...?** block and then, in the dropdown menu, select **Knight** inside the control block**.**

Now delete the point in direction 180 block; change switch costume to ... to ballerina-b and replace the think **That hurts!** for 1 seconds block with say Hi! for 2 seconds .

You have to end the program by making the following changes

salesianos
EL PILAR - SOTO DEL REAL

Modify the program:

1. Modify the program and include a third character that the ballerina and the knight say "hello" to.
2. Import the third character that you have to download **from the Internet**. This character has to be an image format (their extension will be **.jpg**, **.jpeg**, **.png** or **.gif**).
3. Then, Modify again the program o that the ballerina only falls if the colour of her dress touches the colour of the third character.

Save the program File name:

ApellidoNombre-TPR1EB-practical 4

Practical 5.

Blocks in the Control Menu.

ESCAPING FROM GHOST

Creating the interface

BACKDROP

We are going to create the backdrop.

Go to **Choose a backdrop** and select **Castle 3** from the **Fantasy** category.

Add the **Castle 2** backdrop too because we will use it later.

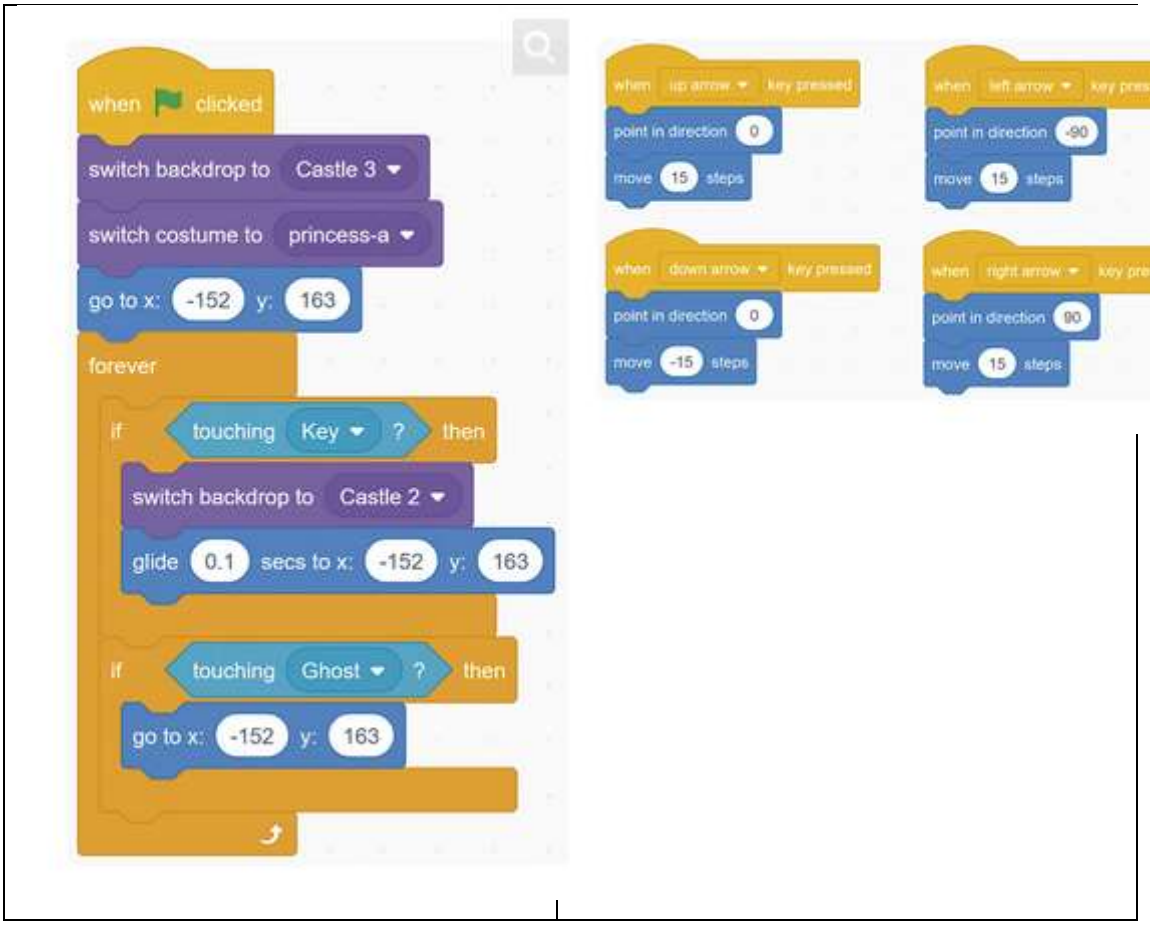
SPRITE

Go to **Choose a sprite** and select:

- Princess
- Ghost
- Key .

Control Blocks

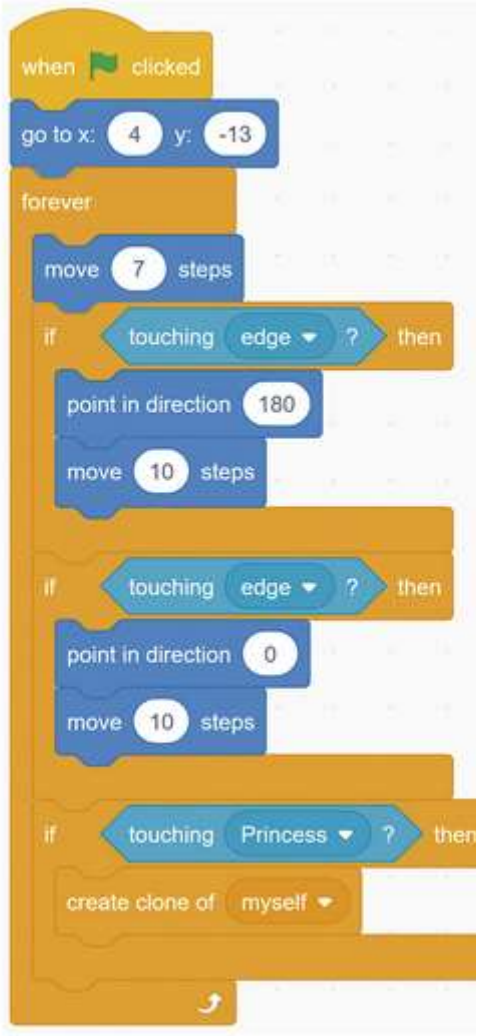
Select the **PRINCESS** sprite and write the next program

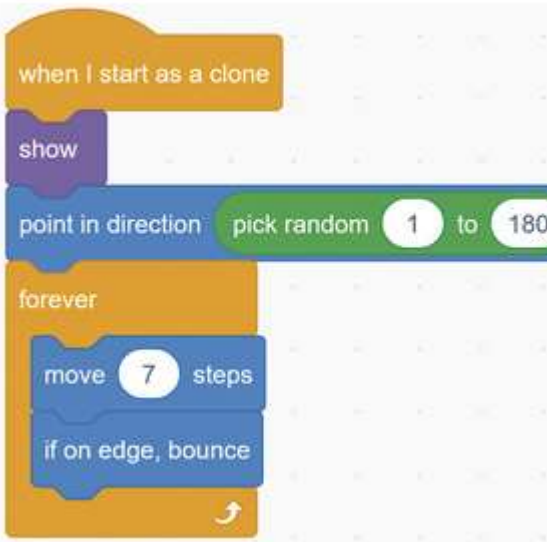


Use the arrow keys to get the princess across the screen until she reaches the magic key.

If the princess reaches the key, the backdrop will change.

Select the **GHOST** sprite and write the next program





A ghost will try to stop THE PRINCESS and, every time it reaches her, it will send her back to the start and the ghost will replicate.

You have to end the program by making the following changes

Modify the program:

Modify the program to create another level of action in the second backdrop (castle 2) using similar instructions.

Save the program File name:

ApellidoNombre-TPR1EB-practical 5

Practical 6.

Blocks in the Control Menu.

THE VIRTUAL DICE

Creating the interface

BACKDROP

We are going to create the backdrop.

Go to **Choose a backdrop** and select **Light**.

SPRITE

Go to **Choose a sprite** and select:

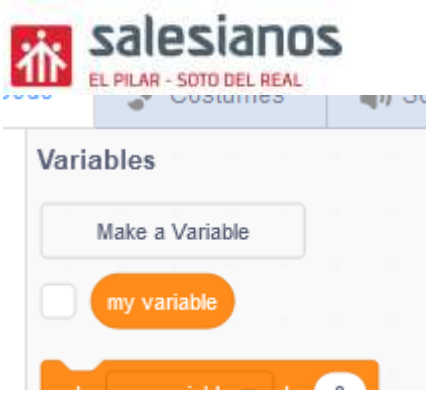
- Star.

Position the STAR in the centre.

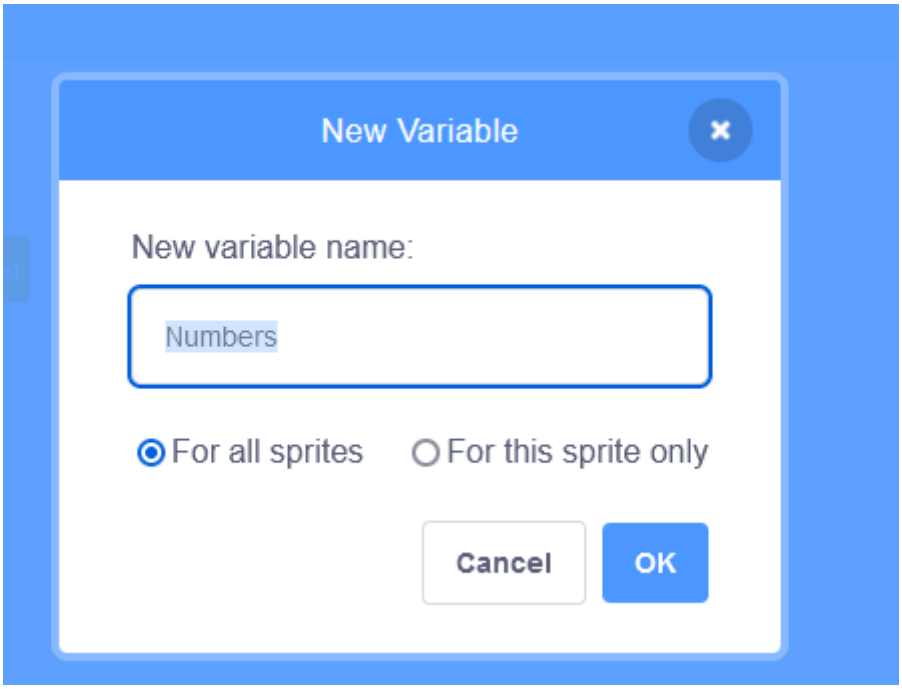
NUMBERS

To create the Numbers box, we have to create a variable.

To do this, select the Variables category and then click on Make a variable.



In the pop-up window, type “Numbers” as the name of the variable



Select For all sprites and click on OK

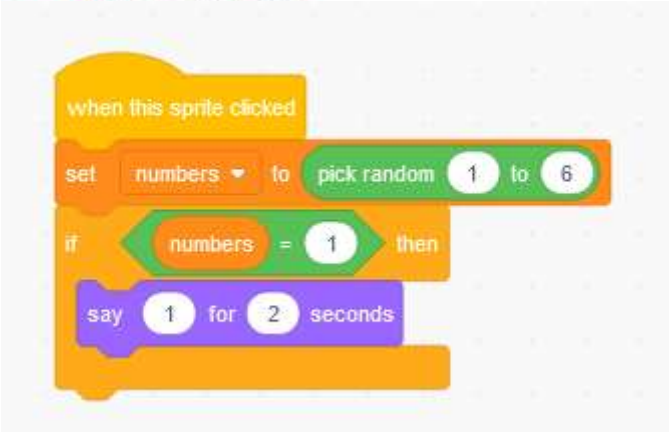
You will see how it appears on the stage and in the Variables category

Creating the program

When we click on the star, values from 1 to 6 will appear at random.

Now we want a message to appear showing the number we obtained. For this, create the entire program shown in the figure, with the six **if ... then** blocks.

salesianos
EL PILAR - SOTO DEL REAL



Continue the program putting the blocks for the numbers from 2 to 6.

You have to end the program by making the following changes

Modify the program:

Modify the program and create a program in which instead of putting a number on each face of the die, you will put a drawing that you like on each face

Save the program File name:

ApellidoNombre-TPR1EB-practical 6

Practical 7.

Blocks in the Operators Menu.

BUILDING CALCULATOR.

Creating the interface

SPRITE

Search the internet for the image of a **teacher** to make the program in that character

Using the **Upload sprite** option in the **Choose a sprite** button,

Upload the files you have in Google Classroom:

- Search the internet for the image (or similar) of 
- Search the internet for the image (or similar) of 
- Search the internet for the image (or similar) of 
- Search the internet for the image (or similar) of 

PROGRAM

We are going to do the **program for the PROFESOR**

When the program starts, we want it to ask “What is your name?” and store the answer in the answer block (Figure 46).

Fig. 46

To make the program answer “Hello” followed by our name:

- use the join operator from the Operators category to join “Hello” with the answer to our question

.

the program will be

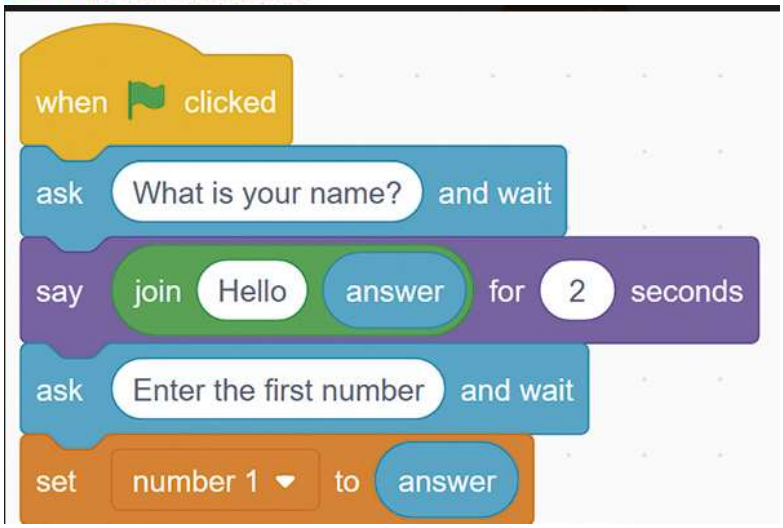
VARIABLES

We are going to ask for the two numbers that we will use to perform the operations.

- First we must create a variable to store the value of each number.
  - number 1
  - number 2
- In the Variables category, click on Make a variable and create number 1 and number 2 (Figure 48).

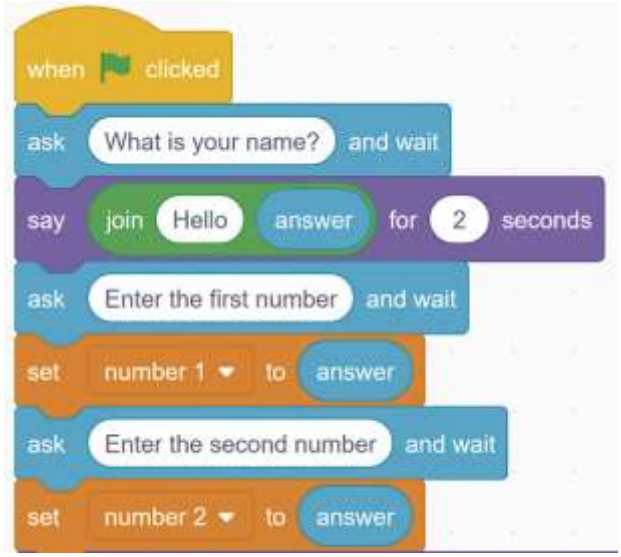
Fig. 48

- Now we will ask for the first number and, using the set number 1 to answer block,



we will store the result in the variable number 1.

- Now we will ask for the first number and, using the set number 2 to answer block, we will store the result in the variable number 2.



we will store the result in the variable number 2.

BUILDING CALCULATOR

- Select each sprite and write the code so that it tells us the result of the operation when we click on the sprite.

- To do this, use the add, subtract, multiply and divide join blocks from the Operations category

- PROGRAM FOR THE SPRITE ADD

  To do this, use the ADD and divide join blocks from the Operations category

- PROGRAM FOR THE SPRITE SUBTRACT

  To do this, use the SUBTRACT and divide join blocks from the Operations category

- PROGRAM FOR THE SPRITE DIVIDE

  To do this, use the DIVIDE and divide join blocks from the Operations category

  DO IT YOURSELF

- PROGRAM FOR THE SPRITE MULTIPLY

  To do this, use the MULTIPLY and divide join blocks from the Operations category

  DO IT YOURSELF

salesianos
EL PILAR - SOTO DEL REAL

Check that the program works correctly

You have to end the program by making the following changes

Modify the program:

Modify the program to display the result: “multiplying one number (number 1) by the second number (number 2) equals (result)”

Save the program File name:

ApellidoNombre-TPR1EB-practical 7