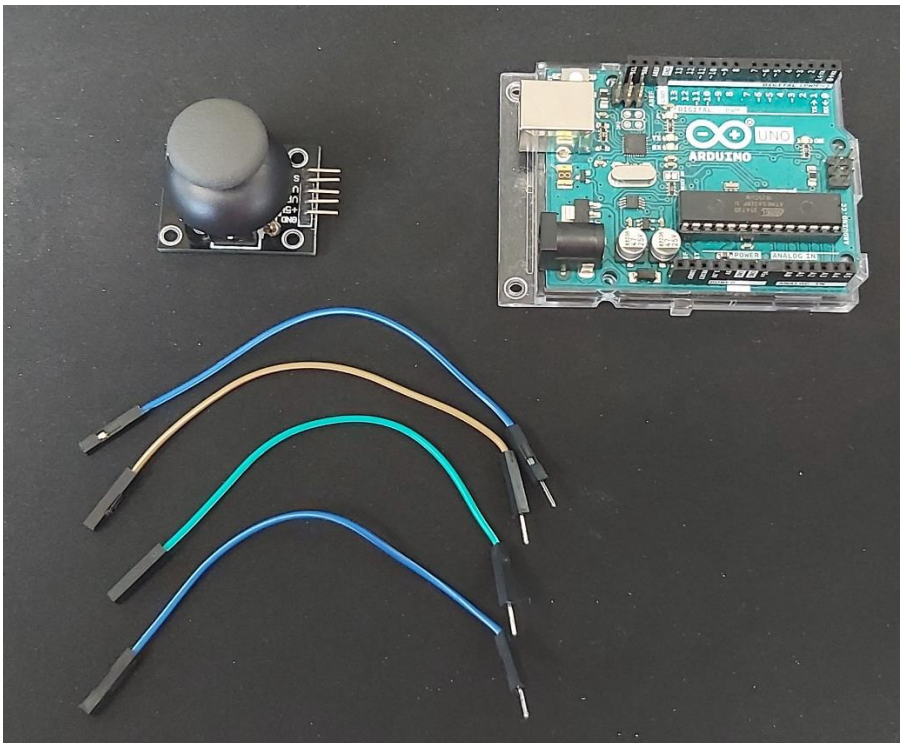
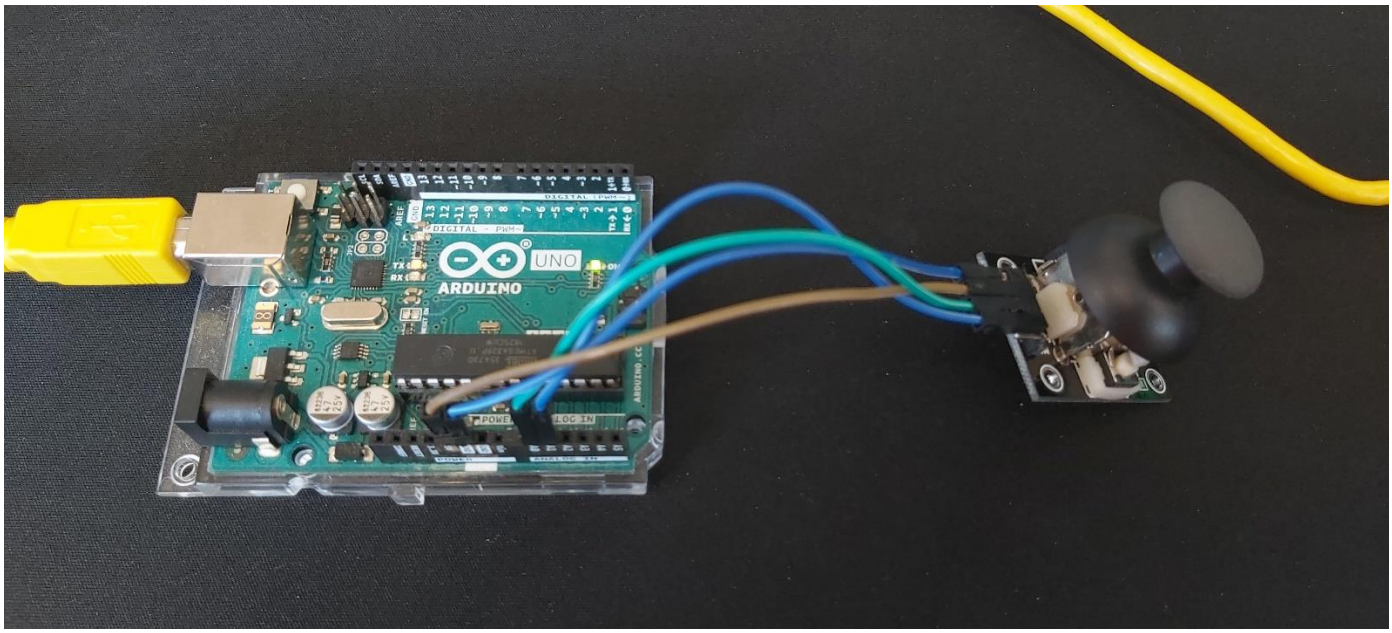
SCENARIO		
Title	CAT vs MICES	
Summary	This activity consists in that the students know the vector axes and acquire skill with the joystick to play the video game. In addition, the student will have to program a video game that will be connected to the Arduino and the Joystick.	
Author/s	AIJU	

DIDACTIC OBJECTIVES				
- Know the vector axes. - Programming a Videogame connected to Arduino and the Joystick.				
Physics **X**	Mathematics **X**	Information Technology ☐	Robotics ☐	Programming **X**
Education Level:	10-12 years ☐	12-14 years **X**		

PROBLEM STATEMENT
At first, programming and the Arduino world can be difficult, but if students are motivated with the incentive to create their own video game and even their own console, the interest and desire to learn is awakened.

BOM (Bill Of Materials needed)
- Arduino Device - (x4) Cables - Joystick

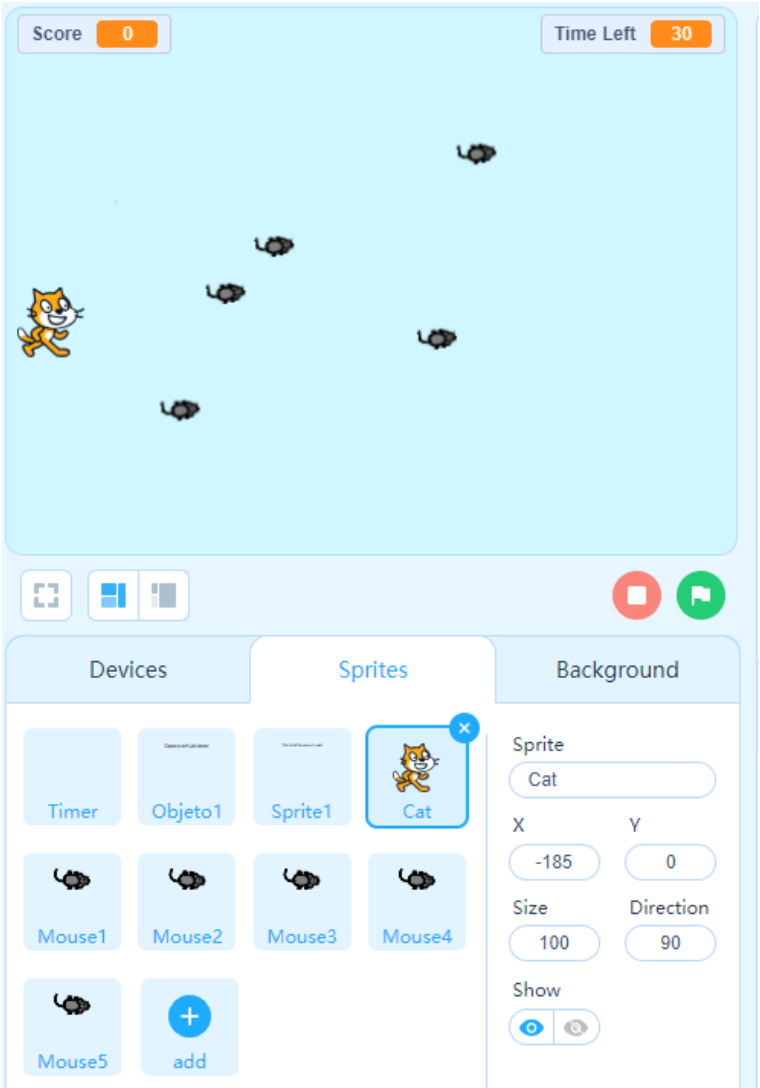




ACTIVITY DESCRIPTION

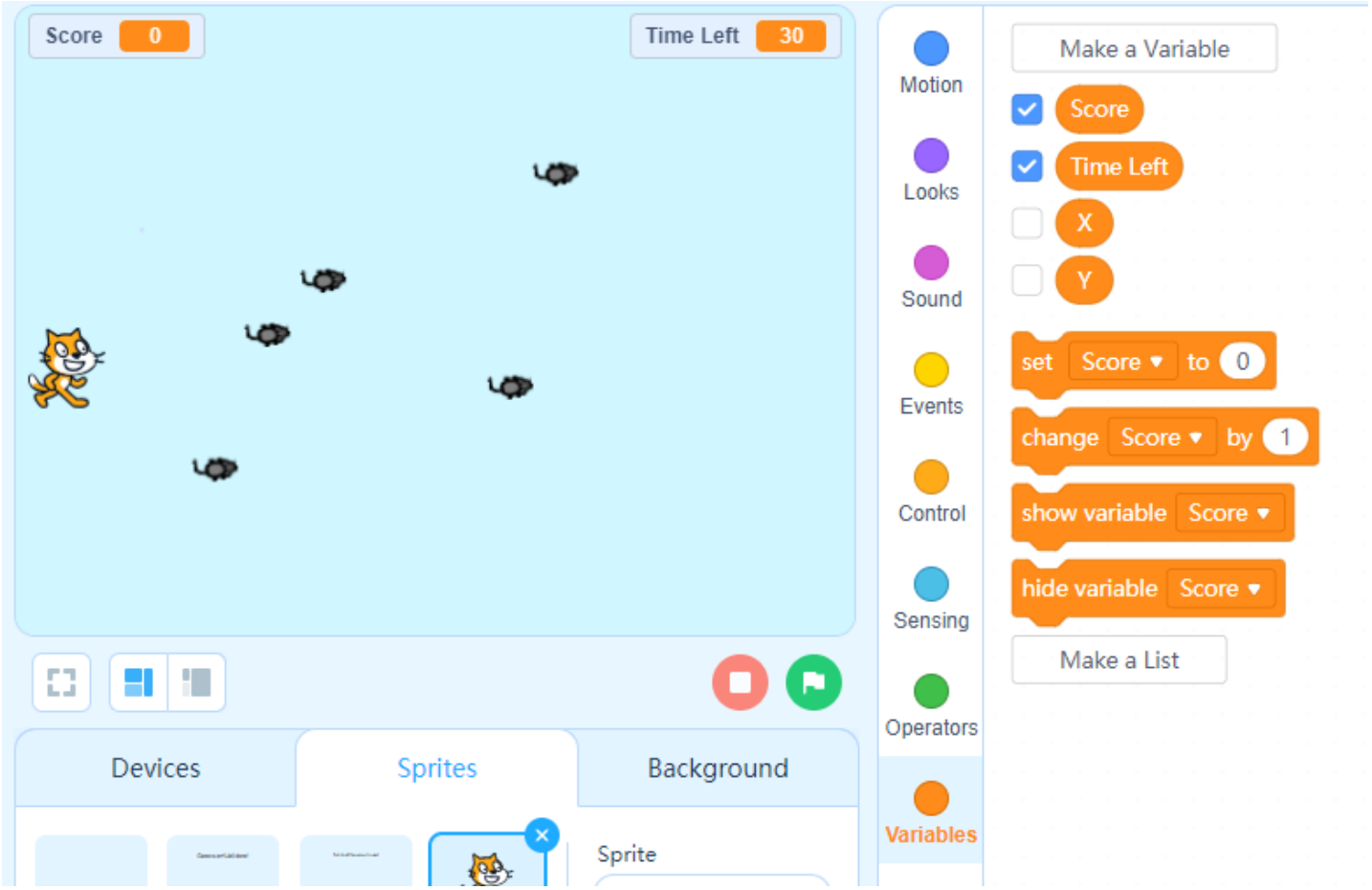
For the development of the activity, we will use software that allows us to unify the game developed in Scratch with the use of the Arduino board. In this case, we have used the mBlock software: (https://mblock.makeblock.com/en-us/)

First of all, we will make the graphic composition of the activity:



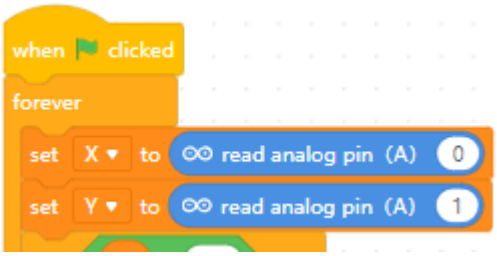
We add the "Score" and "Time Left" variables, for the calculation of the score and the remaining time.

In addition, we will create the X and Y variables that will be responsible for collecting the information generated by the Arduino board.



Once we have the graphic composition and the variables created, we will start with the programming:

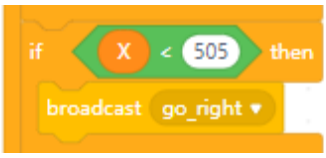
1. In the first place we will program the Arduino board, where we will save in the variables X (movement in the X axis) and Y (movement in the Y axis) the data that when moving the Joystick is generated, so that this data is sent to the game character at through the two variables:



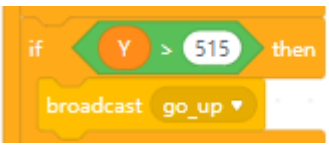
2. When the joystick is in the centered position, the values for variable X are usually between 505 and 510, and for variable Y, it is usually between 510 and 515, so we will have to program 4 movements:

   - If we want to move to the left, we will tell the system that when X is greater than 510 send "go_left" to the programming of the "Cat" object:

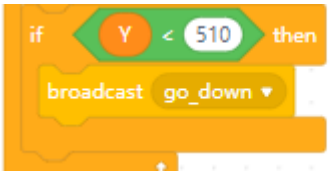
   - If we want to move to the right, when X is less than 505 the object will move to the right, through the call "go_right":



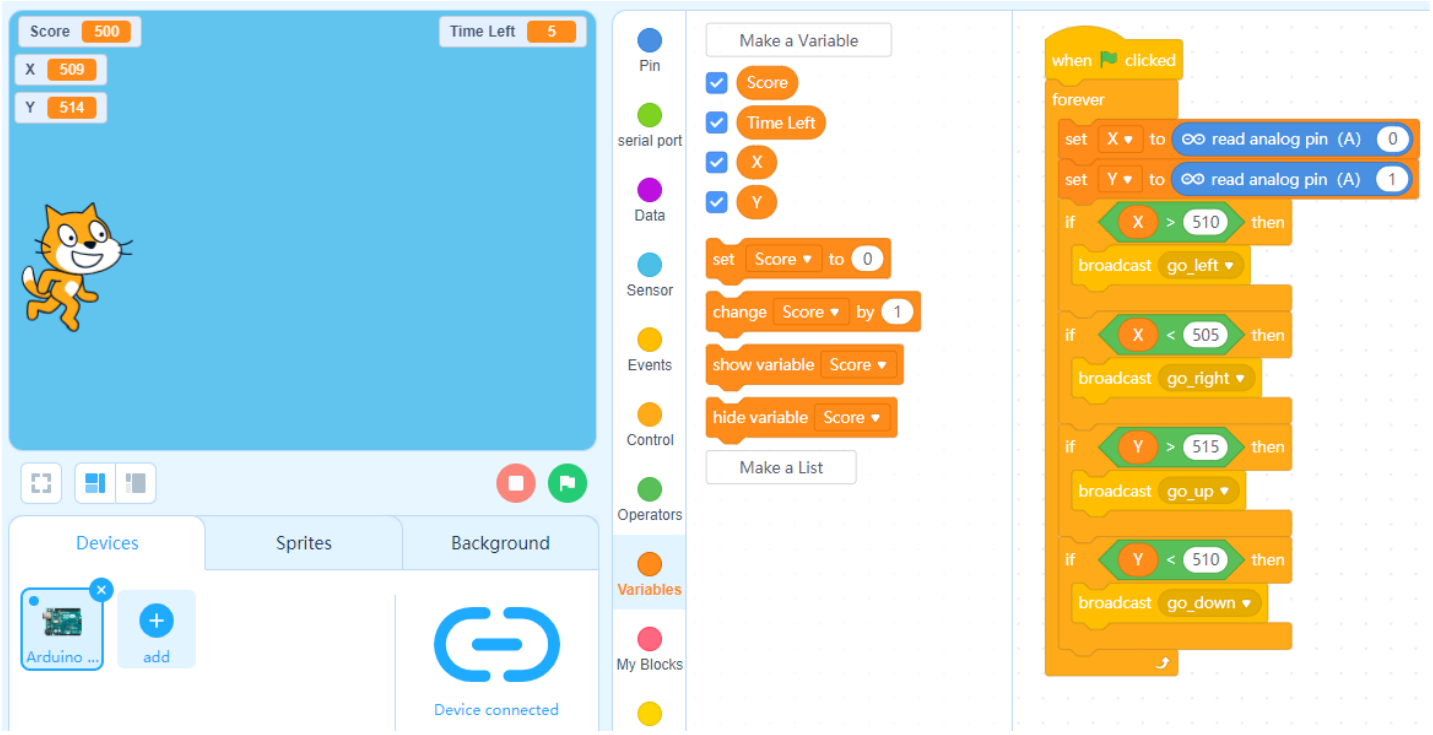
   - If we want the object to move up, through the "go_up" call, when Y is greater than 515 this movement will occur:



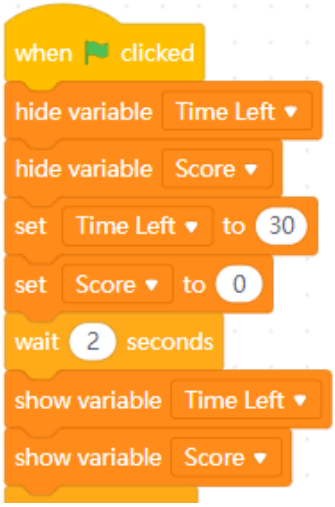
   - And finally, when Y is less than 510, the object will scroll down through the "go_down" call:



3. Therefore, in this way the programming referring to the Arduino board would be:

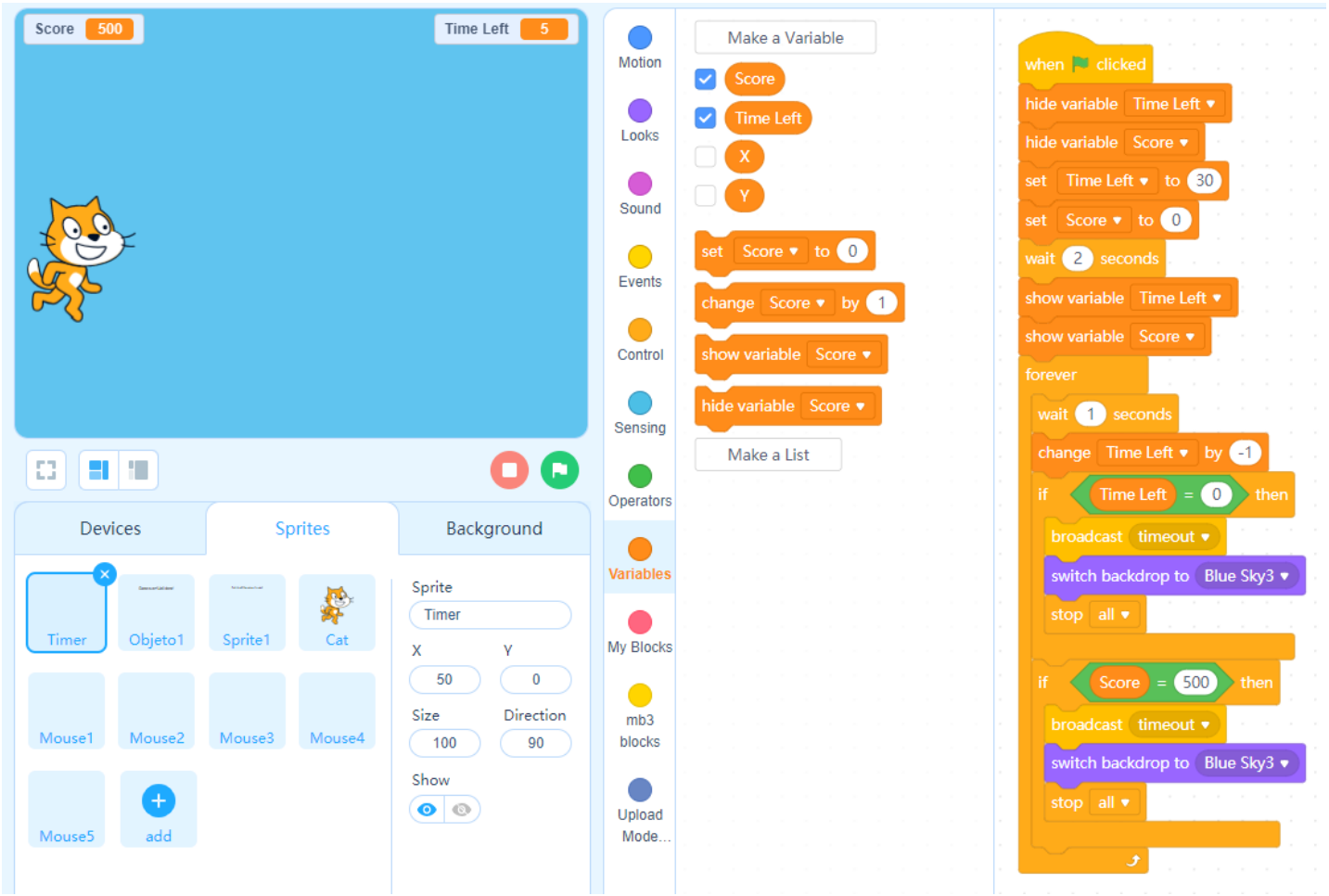


4. Next, we proceed to program the chronometer, which will count down. Really, the important thing is to start the variable "Time Left" in the seconds that we want the chronometer to have. In this case, it will be 30 seconds. Also, we indicate that the variables "Score" and "Time Left" are shown in the game so that the user knows his score and the time he has left to finish capturing the mice:

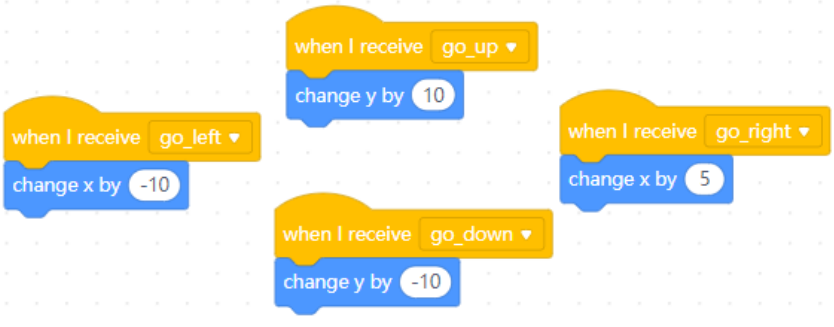


5. In the loop, we will indicate that every second deducts one on the chronometer and if the time reaches 0 or if the player captures all 5 mices, the game stops and the game background is changed:

6. The programming for this section would be as follows:

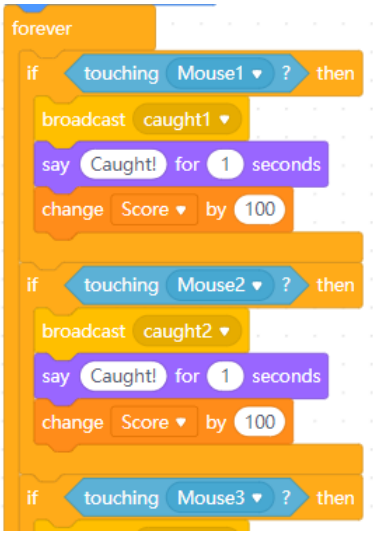


7. Now we will proceed with the programming of the "Cat". When the game starts, the cat will wait to receive the orders sent by the Arduino board through the previously established messages:

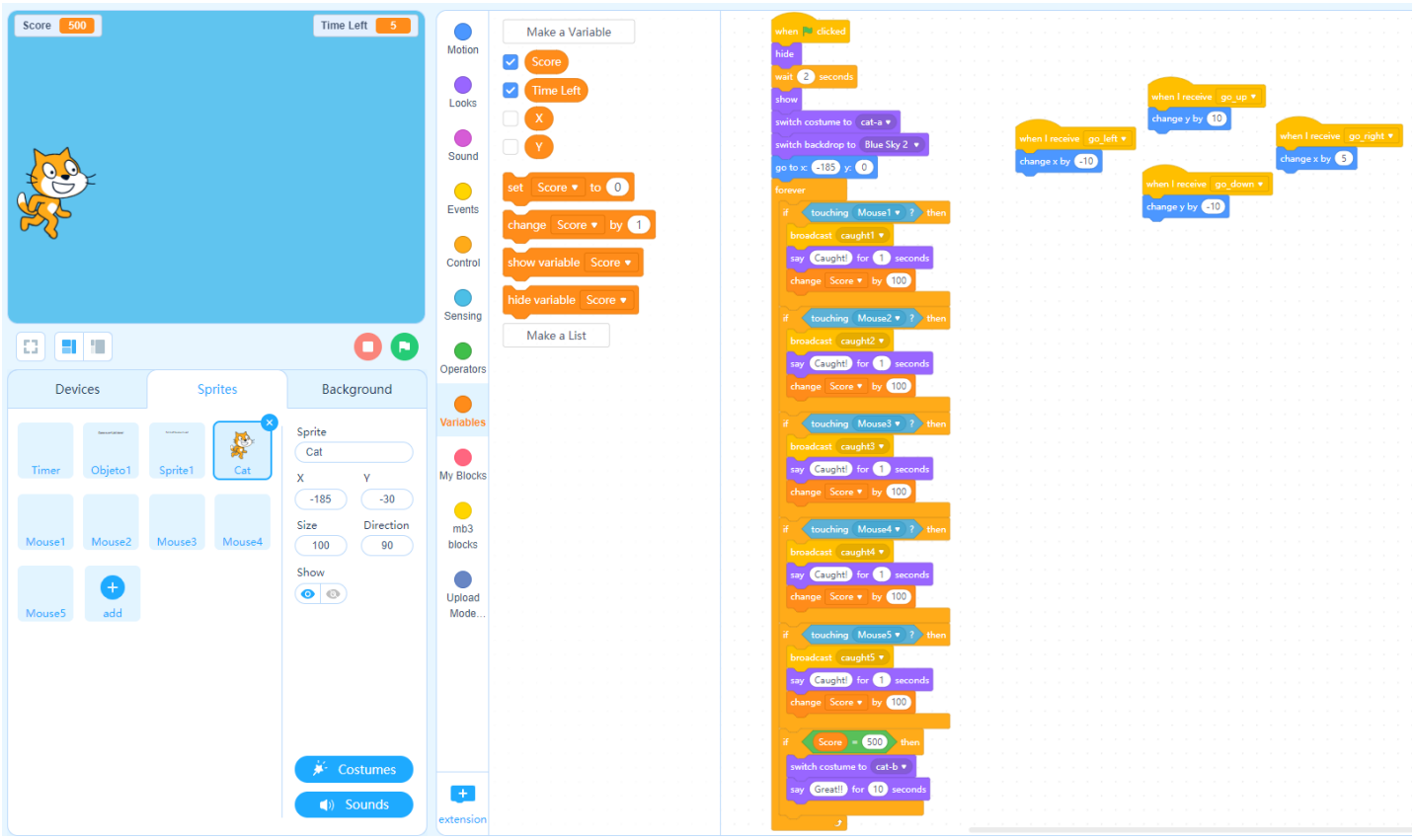


Values 10 or -10 indicate movement speed.

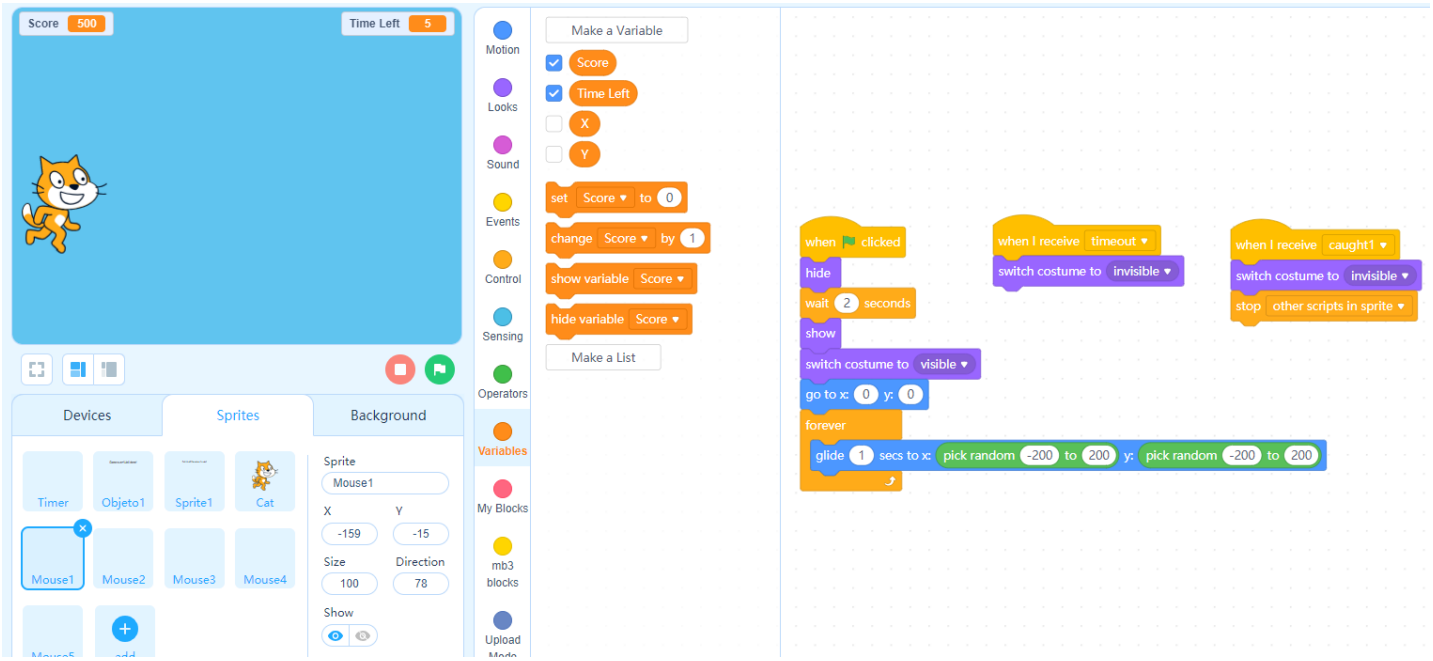
8. Then we will create a loop where we will program the cat's reaction when capturing a mouse: "Caught!":



9. The complete “Cat” programming would be the following:



10. Finally, we will program the movements of the mice:

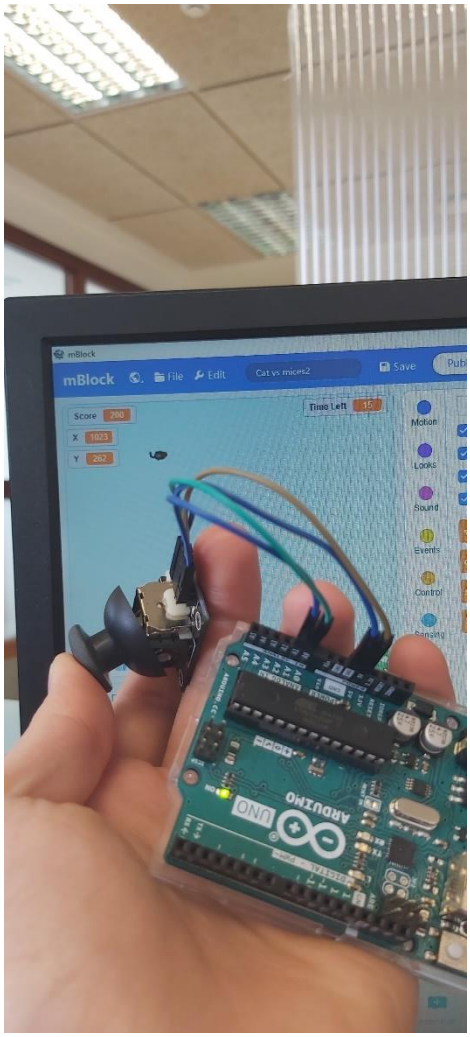


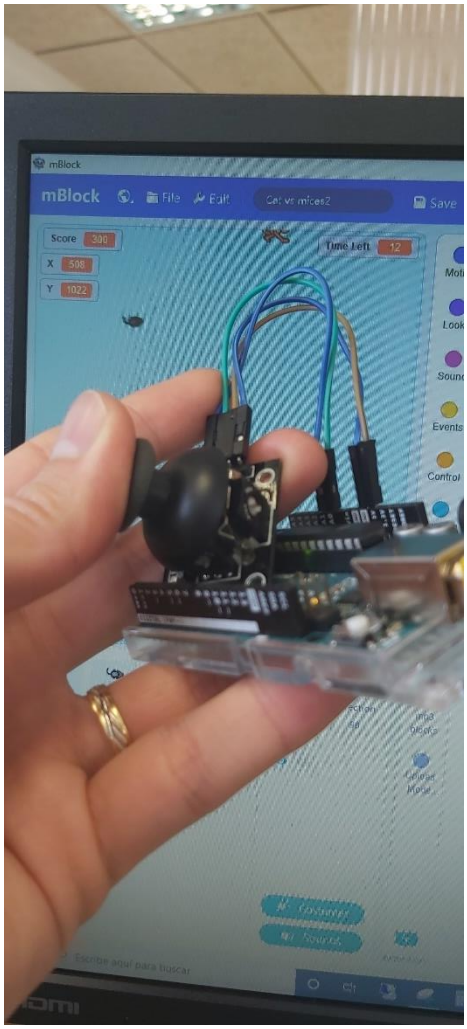
The most interesting block would be the one that makes the mouse move randomly across the game screen:

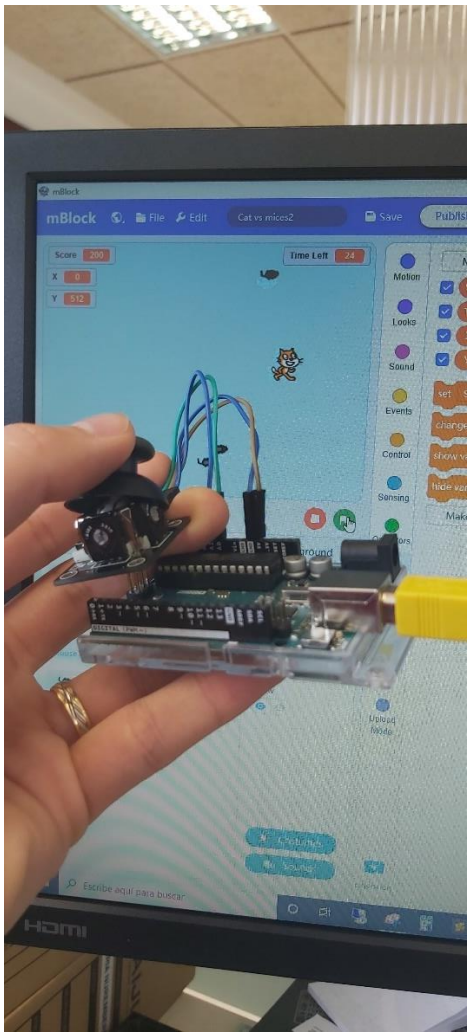
SCALABILITY

Regarding the concept of scalability, the complexity could be increased by adding more objects in the game or even buttons on the Arduino board that allow some special function for the video game character.

RESOURCES


mBlock
File
Edit
Cat vs mices2
Save
Score
Time Left
Motion
Looks
Sound
Events
Control
Sensing
ARDUINO
UNO


mBlock
File
Edit
Cat vs mices2
Save
Score
Time Left


mBlock
File
Edit
Cat vs mices2
Save
Score
Time Left
Motion
Looks
Sound
Events
Control
Sensing
HDMI