GUIDE ACTIVITY INTELLECTUAL OUTPUTS

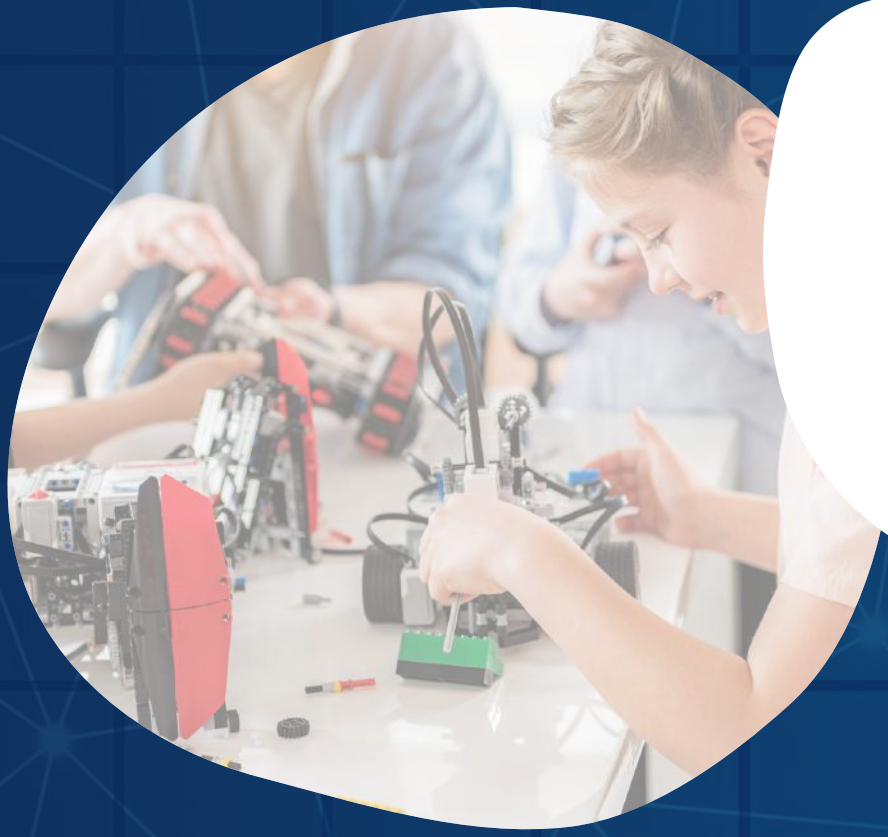
INNOVATIVE SCHOOLS ADAPTED TO THE DIGITAL SOCIETY FOR IMPROVING TECHNOLOGICAL EDUCATIONAL SKILLS

Project no. 2020-1-ES01-KA201-082648

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

OUTPUT 1
EDUCATIONAL ROBOTICS

EDUCATIONAL ROBOTICS IS AN INTERDISCIPLINARY METHOD WHERE SUBJECTS SUCH AS MATHEMATICS, TECHNOLOGY, SCIENCE AND ENGINEERING ARE WORKED ON. THANKS TO IT, CHILDREN CAN DEVELOP LOGICAL THINKING, IMAGINATION AND LINGUISTICS



1

output

ACTIVITY 1

TITLE PYRAMID TREASURE HUNT

ABSTRACT

Inspired by the exploring of the Egypt's 4,500-year-old monument Great Pyramid of Giza's hidden chamber by robots and the necessity of exploring spaces impossible for humans to access, we tried to simulate a similar situation in the class. The activity was a great challenge - 7D grade students came up with a whole list of ideas on how we could organize the tunnel, how to control the robot on the tunnel, how to film the route and how to visualize it. The activity was very engaging and the students' participation was excellent.

The aim of this activity is that we want to teach our students 2 basic things in our project called “Pyramid Treasure Hunt ".
1. Using the Arduino 4WD robot kit,
2. Using the Appinventor software,
We will do these studies at the secondary school level. We will use both text-based and block-based programs as software. We will also produce great works with a 3D printer.

AUTHOR/S

Maria Rosetti Secondary school

DATE 06/12/2022

VERSION 1

DIDACTIC OBJECTIVES

1. The student knows how to use the “Thingiverse” software.
2. In the Student Appinventor software;
2.1 Knows how to create an account
2.2 Knows how to use Appinventor design screen
2.3 Knows how to use Appinventor coding screen.

ACTIVITY 1

3. In the student Arduino Ide software;
3.1Know how to install the program
3.2 Knows how to define and use variables to hold data in the program.
3.3 Arranges the input and output of Arduino pins.
3.4 Creates the program algorithm and writes the code.

4.While the student is creating the disassembled robot;
4.1 Arduino connects to the correct pins on the board
4.2 Establishes the connection between Arduino board and motor driver.
4.3 Establishes the connection between motor driver and moto.
4.4 Establishes the connection between the Bluetooth module and the Android software.

- [x] **SCIENCE**
- [x] **TECHNOLOGY**
- [x] **MATHEMATICS**
- [ ] **GEOGRAPHY/HISTORY**
- [ ] **LANGUAGES**
- [ ] **LITERATURE**
- [ ] **MUSIC**
- [ ] **OTHERS .....**

EDUCATION LEVEL

This activity is prepared to be completed by...

- [x] **12 - 14 YEARS**
- [ ] **14 - 16 YEARS**
- [ ] **OTHERS .....**

TOOLS NEEDED

-Arduino 4WD Kit Contents
- 4 pieces – 65mm diameter wheels.
- 4 pieces – Plastic geared gearmotor + encoder disc.
- 2 Pieces – Top and bottom plexiglass car body.
- Battery bed.
- Mechanical and electronic fittings.

-Tools needed for soldering(Adult help should be sought when soldering.)
-3D Printer
-Software(Arduino Ide , Appinventor, Thingiverse)
-Mobile phone or Camera

ACTIVITY 1

DEVELOP ACTIVITY

First Step: Designing using Thingiverse software

The aim of our project was to search for treasure by making a robot that we can move in the direction we want. For this reason, we have designed objects that we can think of as treasures. Again, as the search process will take place in Egypt as per the project, we can also design objects specific to that area. For example the Pyramid. For this, we need 3D design programs. You can also use Tinkercad if you want. We chose Thingiverse.

Second Step: Mobile Application Development with Appinventor(Design Panel)

We use the buttons to determine the directions. We write "Front", "Left","Rigth","Back" in the text field of the buttons. To stop, we create a "Stop" button. We add two buttons where we can see the connection status. You can customize the design as you wish. Button colors, fonts, some extra labels etc. We chose a simple design. In addition to these, let's add "Bluetooth Client" and “Bluetooth Server” elements. But these are the invisible components.

Third Step: Mobile Application Development with Appinventor(Code Panel)
For each button, we must write the codes that will run when "Touch down” and “Touch Up”. We also determine the numerical values that we will use in the Arduino Ide environment here.
Then, we have to write the codes for connecting to Bluetooth.

Fourth Step: Arduino Ide Software
Arduino Codes
[c]

```
int pwm1 = 10;
int pwm2 = 11;
int way1 = 12;
int way2 = 13;
int received_data = 0;
```

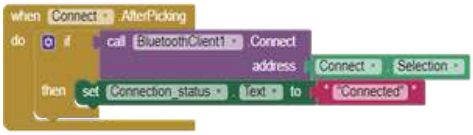
```
void setup()
{
pinMode(pwm1, OUTPUT);
pinMode(pwm1, OUTPUT);
pinMode(yon1, OUTPUT);
pinMode(yon2, OUTPUT);
digitalWrite(pwm1, LOW);
digitalWrite(pwm2, LOW);
digitalWrite(yon1, LOW);
digitalWrite(yon2, LOW);
Serial.begin(9600);
}
void MotorControl(int mway1, int mway2, int pwmInput)
{
digitalWrite(yon1, mway1);
digitalWrite(yon2, mway2);
digitalWrite(pwm1, pwmInput);
digitalWrite(pwm2, pwmInput);
}
void loop()
{
if (Serial.available() > 0)
{
received_data = Serial.read();
if (received_data == 10) //Front
{
MotorControl(LOW, LOW, HIGH);
}
else if (received_data == 20) // Back
{
MotorControl(HIGH, HIGH, HIGH);
}
else if (received_data == 30) // Left
{
MotorControl(HIGH, LOW, HIGH);
}
```

```
else if (received_data == 40) // Right
{
MotorControl(LOW, HIGH, HIGH);
}
else // If Receive another data, stop
{
MotorControl(LOW, LOW, LOW);
}
}
}
[/c]
```

RESOURCES

ACTIVITY 1


when Connect .BeforePicking
do set Connect . Elements to BluetoothClient1 . AddressesAndNames


when Connect .AfterPicking
do if call BluetoothClient1 .Connect
address Connect Selection
then set Connection_status . Text to "Connected"


when Left .TouchUp
do call BluetoothServer1 .Send1ByteNumber
number "50"


when Left .TouchDown
do call BluetoothServer1 .Send1ByteNumber
number "30"


when Front .TouchUp
do call BluetoothServer1 .Send1ByteNumber
number "50"


when Front .TouchDown
do call BluetoothServer1 .Send1ByteNumber
number "10"

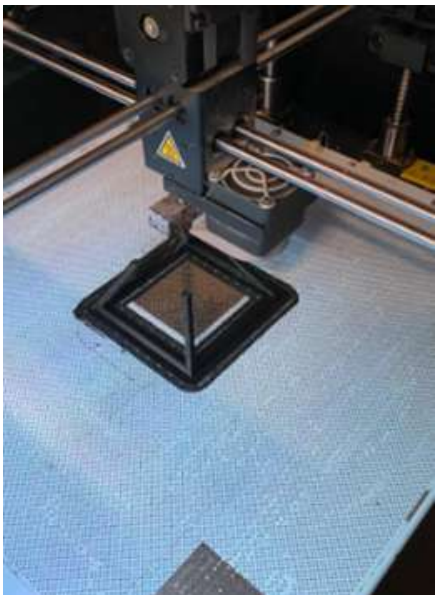

when Right .TouchUp
do call BluetoothServer1 .Send1ByteNumber
number "50"

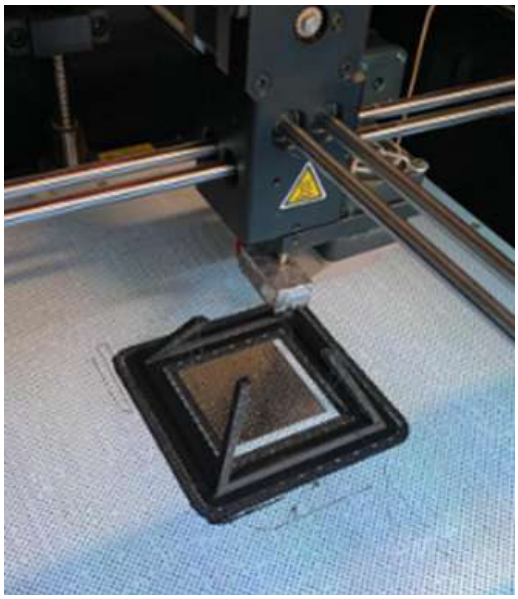

when Right .TouchDown
do call BluetoothServer1 .Send1ByteNumber
number "40"


when Back .TouchUp
do call BluetoothServer1 .Send1ByteNumber
number "50"


when Back .TouchDown
do call BluetoothServer1 .Send1ByteNumber
number "20"

ACTIVITY 1





ACTIVITY 1

STUDENT'S EVALUATION

1. Can the student assemble the disassembled robot correctly?
2. Does the student comply with the safety rules during assembly?
3. Can the student design on the 3D platform?
4. Can the student print from a 3D printer?
5.Can the student write code in the Arduino Ide environment?
Can the student write code in the Appinventor environment?
7. Can the student establish a connection between the robot and the mobile device?
8. Can the student correctly place the camera on the robot?
9. Can the student find the Pyramid and Treasures he is looking for after watching the footage?
10. Can they work in harmony with the student group?

BIBLIOGRAPHY

- https://www.thingiverse.com/
- https://appinventor.mit.edu/
- https://www.arduino.cc/

SCALABILITY

The project is currently done for 4 directions. Intermediate directions can be added in later stages. The study is currently aiming to find treasure by watching the camera recording on the robot.

MORE INFORMATION

This situation can be improved and faster and easier treasure hunting can be done with snapshot transfer.

ACTIVITY 2

TITLE Technological tests

ABSTRACT

Our students had a lot of fun with technology using some objects of robotics:

-they programmed the robot Boost using the specific application, creating the correct movements to cover the itinerary.;
-they assembled correctly the electric circuits of Little Bits in order to make objects move;
-they experienced the three-dimensional glasses choosing games among those proposed by the 3D lens.

AUTHOR/S

IPS Maffeo Pantaleoni

DATE 28/02/2022

VERSION 1

DIDACTIC OBJECTIVES

The objectives to be achieved are the following:
- to promote interdisciplinary learning and teamwork
- to improve interests and enhance motivation
- to stimulate creativity
- to improve problem-solving skills
- to stimulate knowledge of electronics

- [ ] **SCIENCE**
- [x] **TECHNOLOGY**
- [ ] **MATHEMATICS**
- [ ] **GEOGRAPHY/HISTORY**
- [ ] **LANGUAGES**
- [ ] **LITERATURE**
- [ ] **MUSIC**
- [ ] **OTHERS .....**

ACTIVITY 2

EDUCATION LEVEL

This activity is prepared to be completed by...

☐ **12 - 14 YEARS** ☑ **14 - 16 YEARS** ☐ **OTHERS .....**

TOOLS NEEDED

-Lego Boost
-Little Bits
-3D glasses
-Various objects: paper, scissors, plastic, scotch tape, glue, pipe cleaners...
-catalogue of the models to realise

DEVELOP ACTIVITY

The students used the objects in the following ways:
-they arranged the paperboard of the itinerary
-they paced the robot
-they programmed the robot's movements using Lego Boost App.

-they tested the various movements until they found the correct ones which allowed the robot to
reach the end of its path.
-they arranged the electric blocks
-they assembled the blocks in order to hear sounds, to turn on and off the lights, glue the various blocks to the objects to make them move.
-they wore the 3D interactive glasses and performed the movements proposed by the chosen game.

ACTIVITY 2

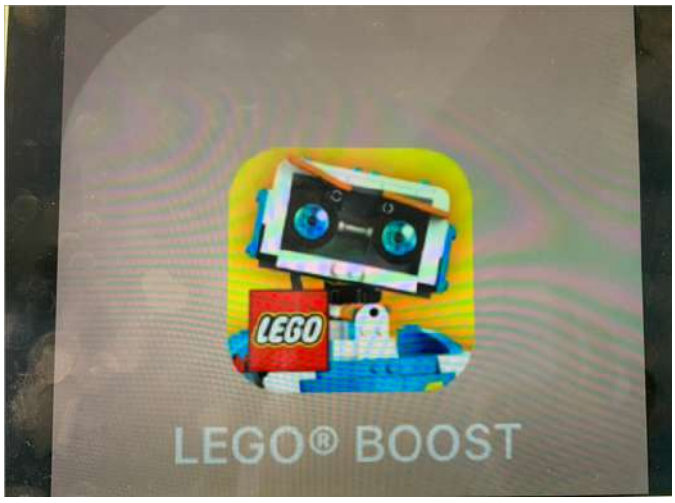
RESOURCES

ACTIVITY 2


2
90
1
90
1
1
360
1
50
1
1
1


50
1
1
1
2
90
1
90
1
2
360
2
0


LEGO
BOOST
BUILD CODE PLAY
LEGO, the LEGO logo, the Minifigure, NINJAGO and the NINJAGO logo are trademarks and/or copyrights of the LEGO Group. © 2018 The LEGO Group. All rights reserved.


LEGO
LEGO® BOOST

ACTIVITY 2

ACTIVITY 2

STUDENT'S EVALUATION

We evaluated the students according to the observation of the achievement of the skills required by the activity.

BIBLIOGRAPHY

Little Bits catalogue

SCALABILITY

In order to acquire the basic coding skills younger pupils could use easier digital and physical blocks.

MORE INFORMATION

Students will be invited to use many complex tools of robotics and physical and digital technological objects such as Arduino.

ACTIVITY 3

TITLE TRANSLATOR ISTEDU

ABSTRACT

Design of an application for mobile phones using App Inventor.
The application incorporates a voice recognition system that captures what is spoken in the form of text. This text can be translated into different languages and, depending on the configuration of the mobile phone, the result of the translation can be heard.
It is very important to note that this application can only be run on mobile phones with an Android system.

AUTHOR/S

IES MEDITERRANEO

DATE 04/04/2022
VERSION 1

DIDACTIC OBJECTIVES

To deepen in the development of programming with pseudocode.
We will work on aspects related to:

- Computational thinking
- Planning a mobile application
- Knowing the basic components of an App
- Designing the screens and the placement of the components of the App
- Programming the components that make up the App

Programming strategy: A strategy based on Computational Thinking will be used. We will divide the global problem, the creation of an application for smartphones that implements a translator, into simpler problems related to each of the components that make it up. Finally, the solutions are integrated into a global project and the possible errors that may arise are checked in order to offer new solutions or improvements to the application.

ACTIVITY 3

The development of the application has two phases:

- Design phase:

The screens and the correct placement of the components that make up the application are designed.

- Programming phase:

The components are programmed and the functionality of the application is tested.

- [x] **SCIENCE**
- [x] **TECHNOLOGY**
- [ ] **MATHEMATICS**
- [ ] **GEOGRAPHY/HISTORY**
- [ ] **LANGUAGES**
- [ ] **LITERATURE**
- [ ] **MUSIC**
- [ ] **OTHERS .....**

EDUCATION LEVEL

This activity is prepared to be completed by...

- [x] **12 - 14 YEARS**
- [x] **14 - 16 YEARS**
- [ ] **OTHERS .....**

TOOLS NEEDED

Necessary materials

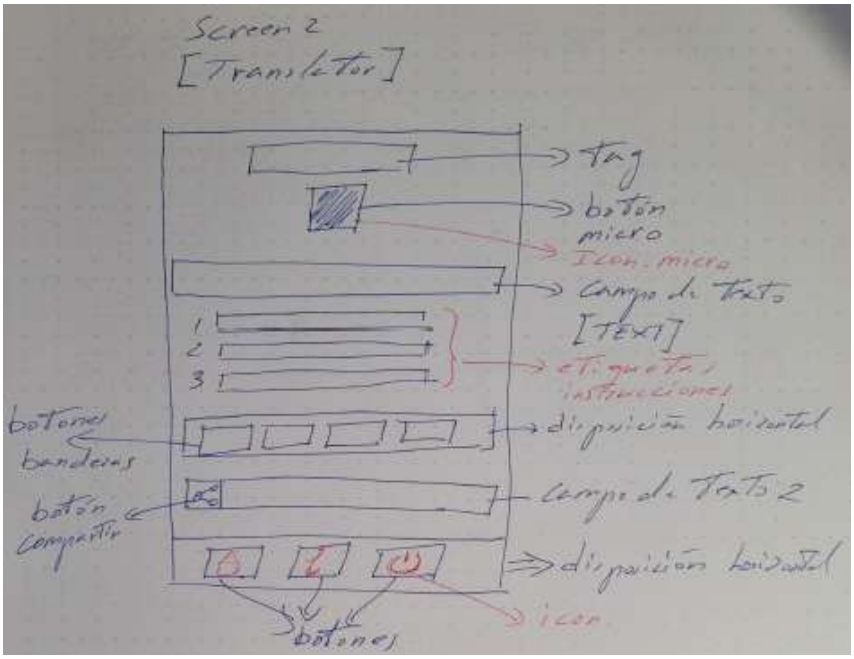
- Notebook and pencils
- Computer
- Smartphone
- Installation of the MIT AI2 Companion application

DEVELOP ACTIVITY

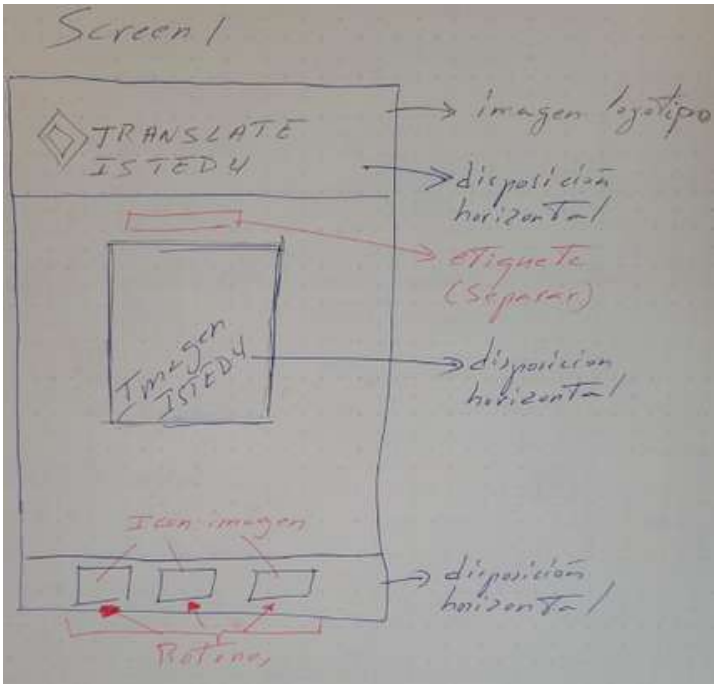
TABLE OF CONTENTS

Step 1. Creating a sketch and planning the project

To create the prototype of the application it is convenient to make a sketch drawing the design and the placement of the components.



ACTIVITY 3

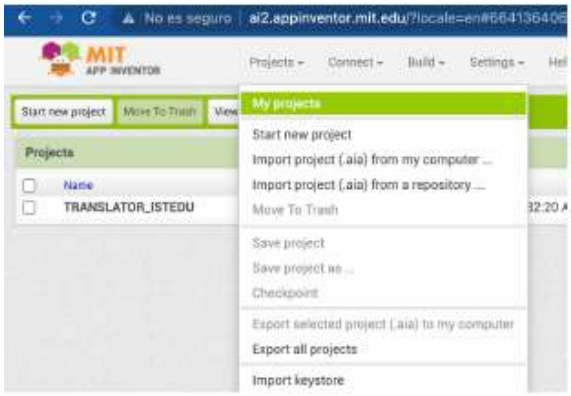


In the design of these sketchesit is convenient to name the components and the function they will perform in the design of the application.

Paso 2. Sign up for App Inventor.

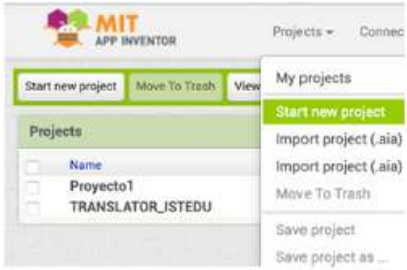
To access the platform, please go to the following link
https://appinventor.mit.edu/
Go to the platform's website and click on "Create Apps".

We will be asked to log in with a Google account and from there we will have direct access to the projects we have worked on. The last project we have worked on will open automatically.
To access our projects, we must deploy the "Projects" tab and access our projects.



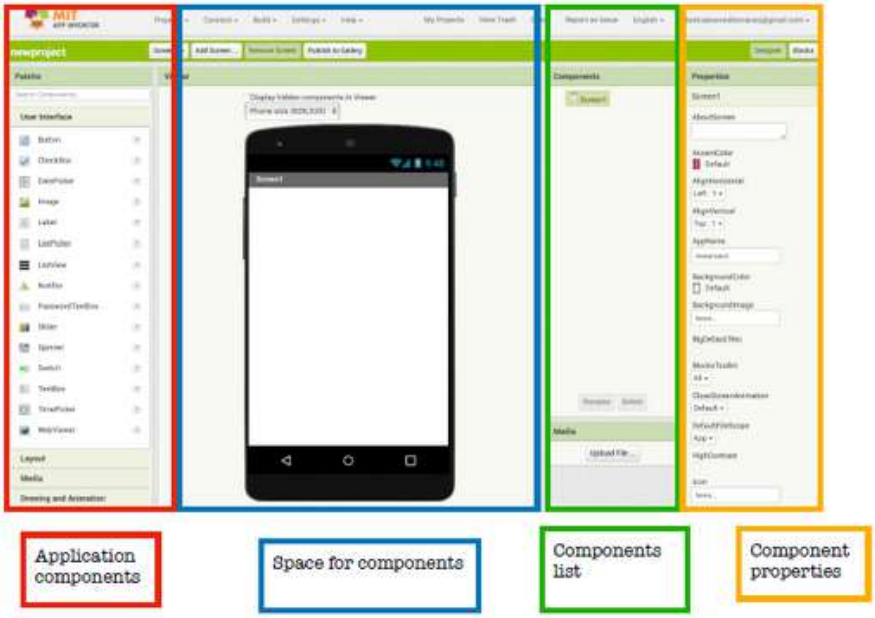
Step 3. Design of the screens that make up the application
The application consists of three screens:
1. Screen1. This is the home screen and where the access buttons to the other screens and all the
information considered relevant to the application will be placed.
2. Translator. This is the screen where the translator will be placed.
3. Information. We use this screen to place information about the project

Screen “Screen1”
To create a new project, simply select this option from the "Projects" tab by selecting "Start a new project".



ACTIVITY 3

We will be asked to name the project and a screen will open with these components



The display shall have the following properties:

In this case it has been decided that the horizontal layout will place the components in the centre.
In the vertical layout the components will be placed starting from the top.
It has been decided not to have any background image.

ACTIVITY 3

From this box we can change the name of the screens.

On this screen we will have the following components

Horizontal arrangement with an image

Horizontal layout with picture

Horizontal arrangement with three buttons.

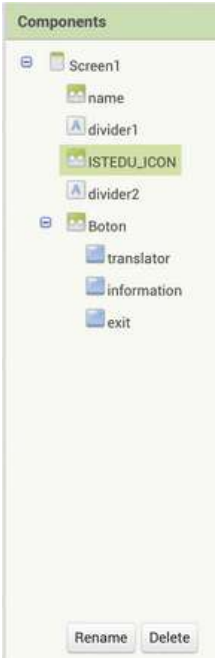
ACTIVITY 3

The layouts act as "boxes" in which the components are placed so that they are arranged in an orderly fashion.

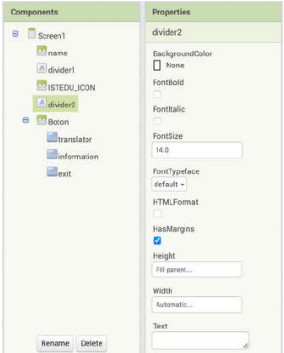
To place a layout we have to open the "Layout" tab from the "Palette" of the components and select "Horizontal Layout". Drag it to the viewer and it will be placed in the centre and at the top.

The properties of this arrangement are as follows

To place an image, simply place the cursor in the image box and click the left mouse button. We can then upload an image from our computer or use an image that has already been uploaded to the project.

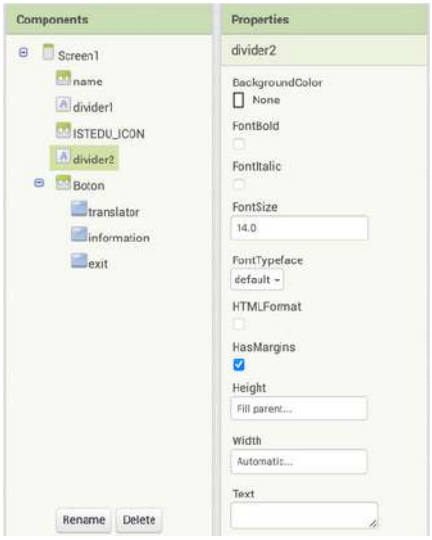


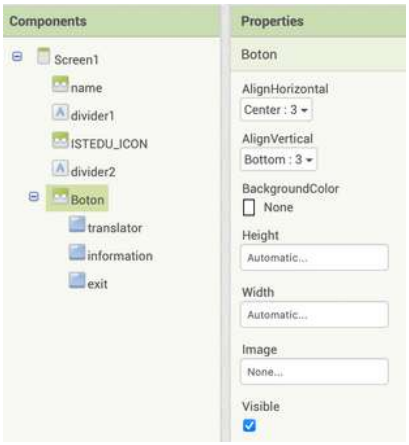
It is important that we rename the components we are going to use. The most useful thing to do is to give them a name that is associated with the function they are going to perform. This is especially important when placing buttons or interactive components.



ACTIVITY 3

In order to adjust the position in the horizontal layout of the screen we can use the "Label" component from the "User Interface" and by making adjustments to its properties and removing the text from the label we can convert it into a divider. In our case, two labels have been used as dividers.

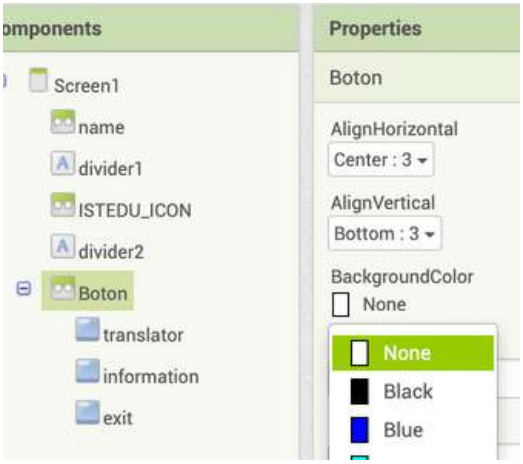




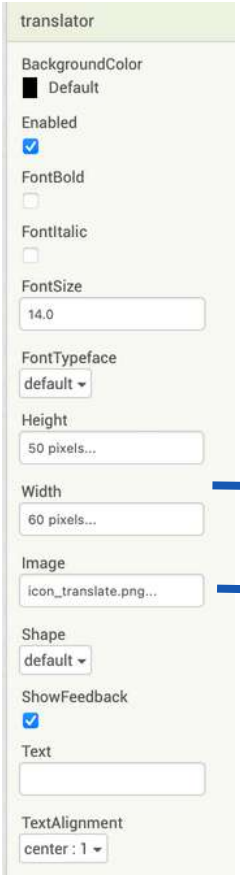
To place the buttons on this screen at the bottom of the screen we will place a horizontal layout and name it "Button" with these properties.

To place a button within a horizontal layout, place the horizontal layout in the centre of the screen and remove the background colour.

Then, from "User interface", we select the buttons and drag them to the "horizontal layout. The buttons will be placed in the appropriate order and can be moved within the horizontal layout.



ACTIVITY 3

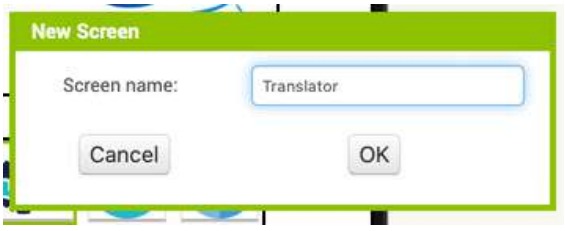


The properties of the buttons are as follows.

Button dimensions are defined.

Select an image to put on the button.

Before designing the new screens we have to create them. To create a window, click on the "Add Screen" button and give a name to the new screen.

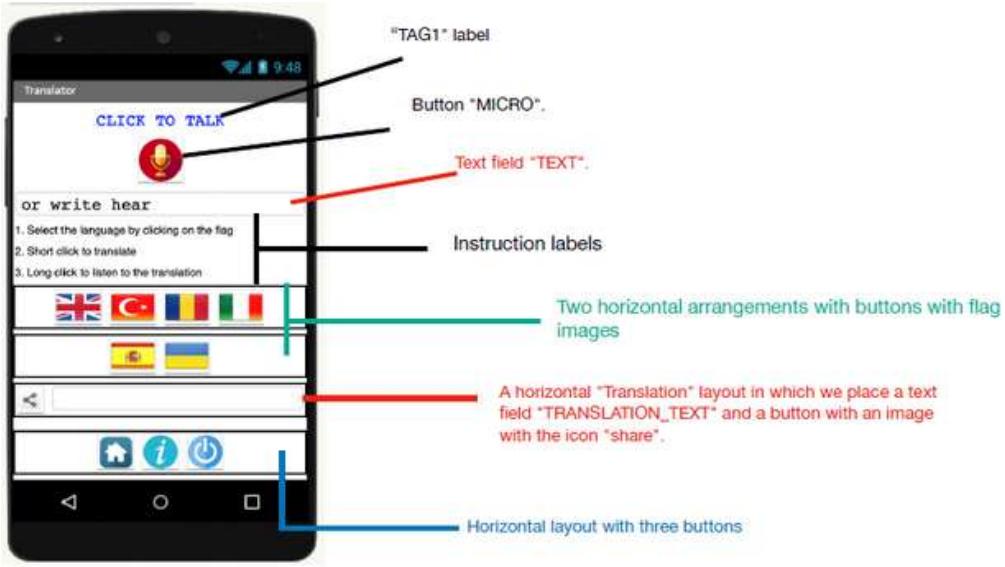


The new screens will be "Translator" where we will program the translator and "Information" where we will place the information about the project.

ACTIVITY 3

Screen “Translator”

The screen components are:



The properties of the two text fields

These two properties are very important

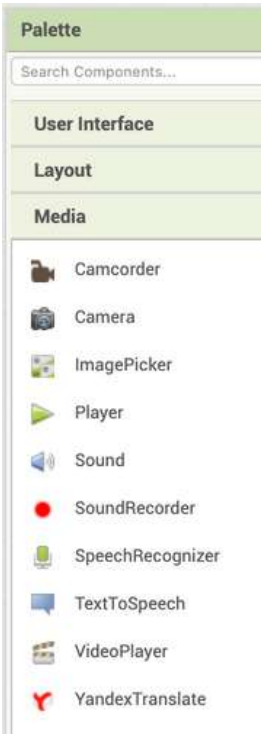
We must check this option "multiline".

This text has been placed to indicate that instead of speaking into the microphone, you can write in the space of the text field.

ACTIVITY 3

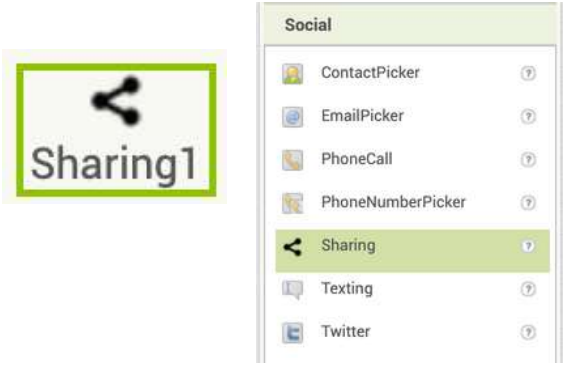
The properties of horizontal arrangements

Non-visible components.



There are a number of components that are selected from the "Media" tab that will be used in programming. These components are selected and dragged onto the screen that simulates the phone. Once we have dragged them, they will be placed outside the screen because they are components that are not shown.
The components we are going to use are:

The Sharing 1 component is located in the "Social" tab.

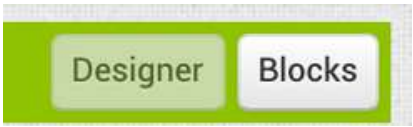


Screen "Information"

On this screen we have simply placed horizontal layouts for the logos of the project partners, a label with information and a horizontal layout with buttons to return to the Home screen, to the translator, or to close the application.

Step 4. Programming the application components

To program the different components of the application, go to the "Blocks" section.



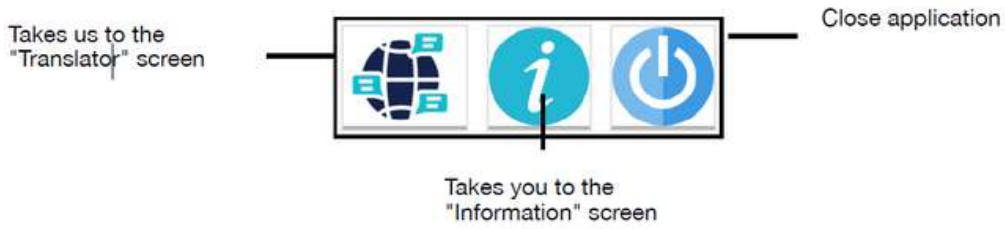
The programming blocks are located on the left-hand side of the platform and are divided into two sections.

- Generic blocks (Integrated) that perform general functions common to all components.

ACTIVITY 3

- Blocks specific to each of the components. They provide component-specific functions, buttons, text fields, labels....

On the first screen "Screen1" we have only three interactive components corresponding to the three buttons



Click on the "Blocks" tab and access the programming section, which will be empty.

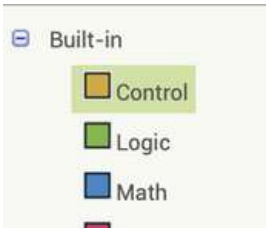
Select the first component in the left panel, which is the "translator" button.

From the Viewer, the programming blocks are displayed.

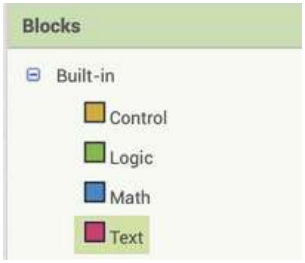
ACTIVITY 3

We select this block and drag it to the programming area. What we are indicating is that clicking on the button will execute a command.

Then, from the common blocks, we deploy the "control" block and select "open another screen...".

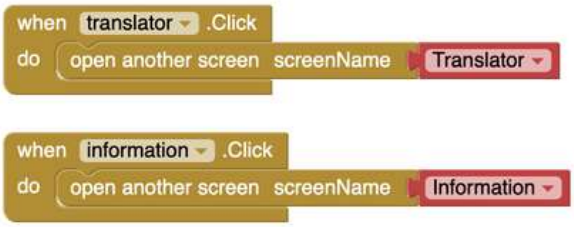


The command that will be executed is that another screen will open. All that remains is to indicate which screen we want to open. For this we select the "Text" block in the common blocks.



Select the block  where we type the name of the screen we want to open.

It is very important that you enter the exact name of the screen you want to open. Finally, all that remains to be done is to fit the programming blocks



This procedure will be performed on the button that opens the "Information" screen.

To program the "Close application" button, select the "Close application" block from "Control".

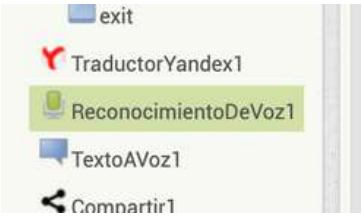
ACTIVITY 3

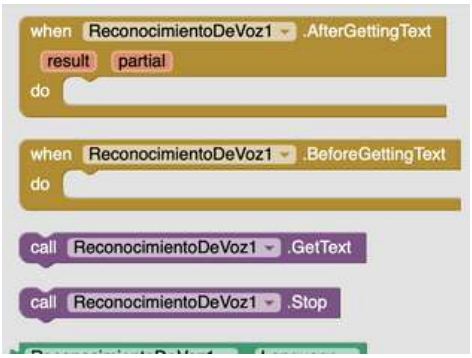
Screen “Translation”

The translator will be programmed from the "Translation" page.

The programming starts by using the "Micro" button. From the Blocks section we select the micro button and use the one-click activation block.

The next step is to select the non-visible component "SpeechRecognition1" and activate speech recognition with this programming block.





This would be the result of linking the blocks. Clicking on the"Micro" button activates speech recognition and will now tell us what to do with the recognised text.

From the speech recognition component, we select this block

Then we deploy the first text block functions "TEXT" and choose this block.

What we are indicating is that once the speech recognition has recognised a text it places it in the Text Field

ACTIVITY 3

From the Speech Recognition block, we drop down the "Result" option and select

The final result will be:

We have already activated the microphone and placed the result of recognising what has been spoken in a text field. Now we activate the translator and tell it to place the result of the translation in text field 2 "TRANSLATION_TEXT".

To activate the translator we select the Yandex translator and from text field 2 we select the text field "TRANSLATION_TEXT". The result will be

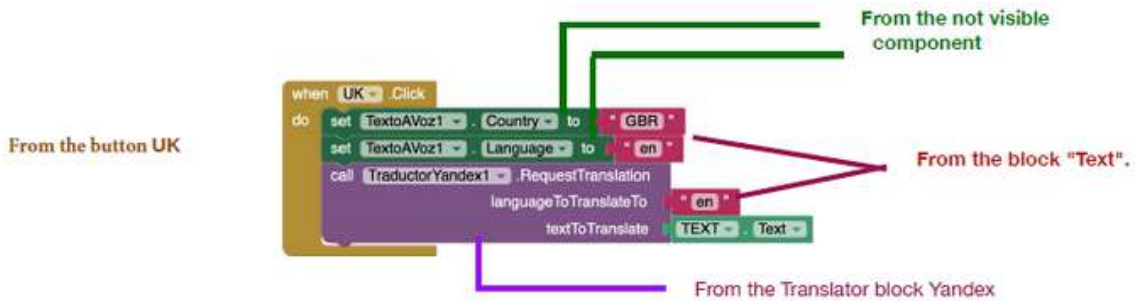
What we are going to do now is to indicate into which language we want the translation to be made. For this we will use the buttons where we have placed the flags of the countries.

The translation will be activated in two different ways.

- With a short click, the result of the translation will be written in Text field 2.
- With a long click you can listen to the result of the translation in the selected language. It is important to note that this possibility will depend on the language settings of each mobile phone.

To translate a text into English, follow these steps:

ACTIVITY 3

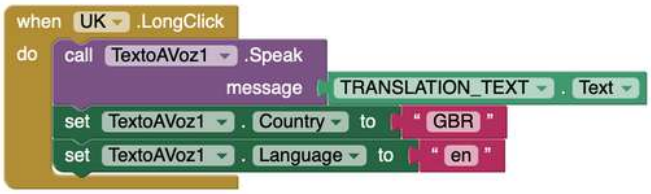


For abbreviations of languages and countries, the ISO codes are used.

Link for ISO languajes

Link to ISO country code

To activate the translation to be heard in the selected language we place these programming blocks



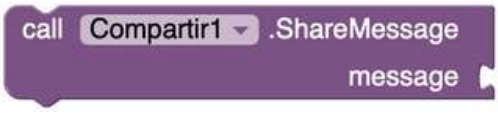
Once we have programmed the UK button, we can copy the codes and simply change the name of the button and the ISO codes of the countries and languages. We will do this operation for as many languages as we have placed buttons with flags.

All that remains is to set the "Share" button. This button allows you to copy the translated text and paste it into new social networks, a word processor or an email.

ACTIVITY 3

From the "Share" button we select the option to Click.

From the not visible component "Share1" we activate the share function



Finally, we tell it which text we want to share.

The buttons in the lower section are programmed in the same way as they were programmed on the "Screen1" screen.

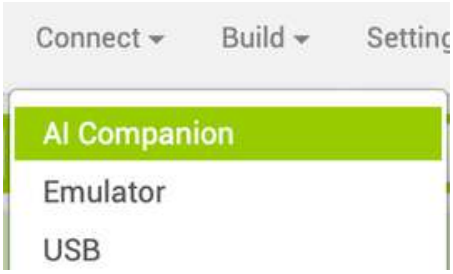
Screen "Information"

This screen requires no programming other than the buttons that allow you to switch screens or close the application.

Step 5. Check results and install the application on a mobile phone

In order to be able to check each step in the development of the application, it is necessary to have the application installed on your mobile phone.

From the "Connect" tab, select the AI Companion option. It is important that the computer and the mobile phone are connected to the same WIFI network.

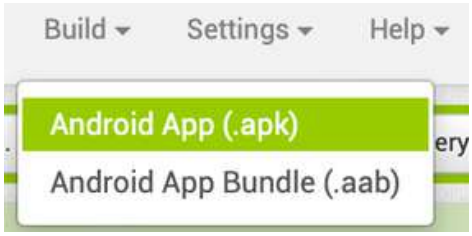


ACTIVITY 3

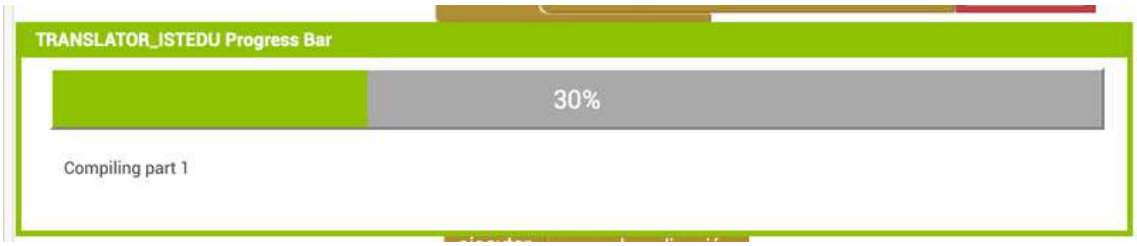
When you make your selection, a page with a QR code will open.

Now from the mobile we will open the MIT AI2 Companion application and select "scan QR code" and we will scan the QR code.

A window will open indicating the level of progress. Once it is finished, the application will be operational on the mobile phone. It must be taken into account that no application has been installed on the phone, we are only performing an emulation of the App.
The "Close the application" option button does not work in emulation mode.
To install the application on a smartphone, from the Generate tab, select Android App (.apk).

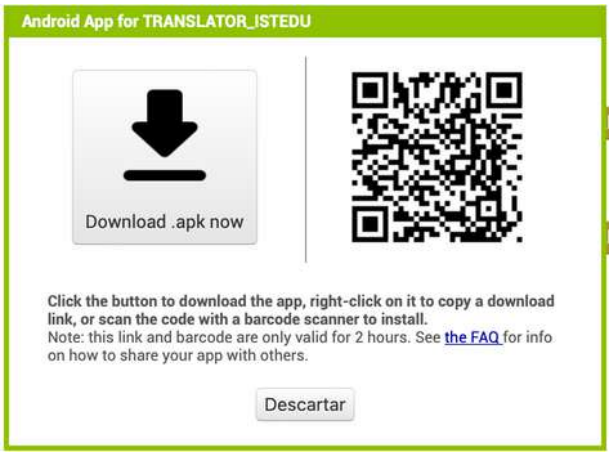


A window will appear indicating the progress of the compilation of the App.



ACTIVITY 3

When finished, an apk file will be generated, which can be downloaded and sent to the smartphone.



RESOURCES

ACTIVITY 3

STUDENT'S EVALUATION

To be assessed:

- Designing an overall project strategy
  - To specify well the function of the App
  - Designing a sketch with the screens and components of the App
  - Use of the basic user interface elements
  - Correct use of the Layout elements
- Breaking down the project into simpler tasks
- Correct and functional design of the App screens
- Programming the App components
- Reviewing versions of the project and correcting possible errors
- Functionality of the project
- Possibility of extending the App with the incorporation of new functionalities

BIBLIOGRAPHY

- MIT App Inventor: http://appinventor.mit.edu/
- Guía de iniciación a App Inventor: http://codeweek.eu/resources/spain/guia-iniciacion-appinventor.pdf
- Tutorial MIT App Inventor: https://rominirani.com/tutorial-mit-app-inventorfirebase-4be95051c325
- App Inventor 2 (en español): http://kio4.com/appinventor/
- Curso App Inventor: https://www.youtube.com/watch?v=sQ2EmGNp2U4

ACTIVITY 3

SCALABILITY

The development of applications with App Inventor makes it possible to create applications that connect
with Arduino cards and act as controllers via wifi or bluetooth for robots or home automation
environments.
It is also possible to create applications that incorporate AI

MORE INFORMATION

Link to the Google Drive folder in which the App is uploaded 

Link to ISO language table 

Link to obtain the ISO code of the countries 

ACTIVITY 4

TITLE Football game

ABSTRACT

STEM: Science, Technology, Engineering and Math. That is a lot of complex material to cram into one acronym. Fortunately for young learners, robotics makes STEM simple and fun to learn.
The passion for STEM and FOR ROBOTS can be realized through games that are, through a pleasant and attractive form, increasing the students' motivation to learn and practice and to put into practice numerous activities and projects.
With so many options for what kids can learn with robotics, it is easy to see the skill-based benefits of using it as a hobby. In addition, there is more good news: Learning the ins and outs of robotics means building real world, career-based skills.
So why not take the plunge today and take these steps?
The most important educational benefits of robotics for children and teenagers are due to the interactive, tactile nature of this field, plus the availability of school clubs, simple robot toys for home and online robotics courses.
Robotics offers excellent opportunities for teamwork and collaboration to design and build a robot to take on opponents in a series of challenges.
That's why we made a short soccer match between two teams using 1 robot for each group of students

AUTHOR/S
SCOALA GIMNAZIALA MARAI ROSETTI

DATE 30/11/2022
VERSION 1

ACTIVITY 4

DIDACTIC OBJECTIVES

Students learn:
- To control a robot
- Scrap programming concepts
- Progressive transition from scratch to Arduino by comparison being a very useful way to learn
-What is mbot, how does it work
- Recognizes Mblock program
- How to use keyes BT car, Tank car apk and keyes 4wd.apk

- [ ] **SCIENCE**
- [x] **TECHNOLOGY**
- [ ] **MATHEMATICS**
- [ ] **GEOGRAPHY/HISTORY**
- [ ] **LANGUAGES**
- [ ] **LITERATURE**
- [ ] **MUSIC**
- [ ] **OTHERS .....**

EDUCATION LEVEL

This activity is prepared to be completed by...

- [x] **12 - 14 YEARS**
- [ ] **14 - 16 YEARS**
- [ ] **OTHERS .....**

TOOLS NEEDED

Mbot(Bluetooth)
Mblock Program
Mbot Remote
Polystyrene to delimit the playing field
2 Robots
2 telephones
ANDROID and IPHONE software

ACTIVITY 4

DEVELOP ACTIVITY

The project setup is given below step by step

1. The polystyrene pieces were measured and cut to be fixed on the floor. They represent the robot's movement limit for the floor. It is preferable to fix the polystyrene pieces well, otherwise the robots will move them during the game.
2. The gates were made of A4 paper
3. The ball used was a tennis ball, but any small ball can be used but with some weight to be handled more easily.

We controlled 3 different robots in the project.

- The first and easiest one is the Mbot controlapplication, which we downloaded from the Play Store. After establishing a bluetooth connection between the mbot and the mobile device, you can easily manage the robot.

- The second is to code Mbot with the Mblock program. It can work with mblok 3.0 and above versions. There are code blocks on Mblock that allow the robot to turn right-left directly. These have been used. We preferred this because we work in a wide area; but those working in a narrower area can also provide rotation by differentiating the engine right and left engine speeds. For example, when M1 engine speed is 100 and M2 engine speed is 255, Mbot will turn in one direction. At this stage, we used the keys of the computer. (wasd and space)

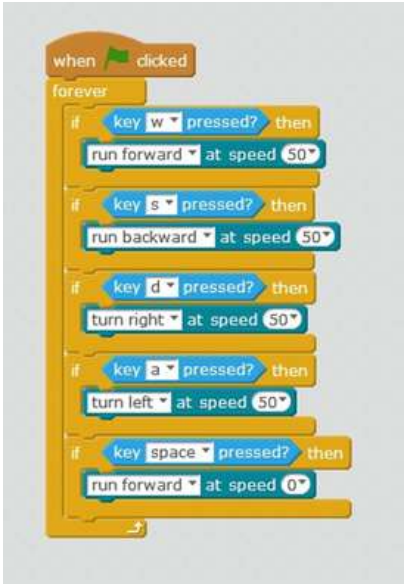
- The third one is a little more difficult but more instructive for our students. We write our own application, assemble the robot ourselves, establish the connection between the robot and the mobile application ourselves.

ACTIVITY 4

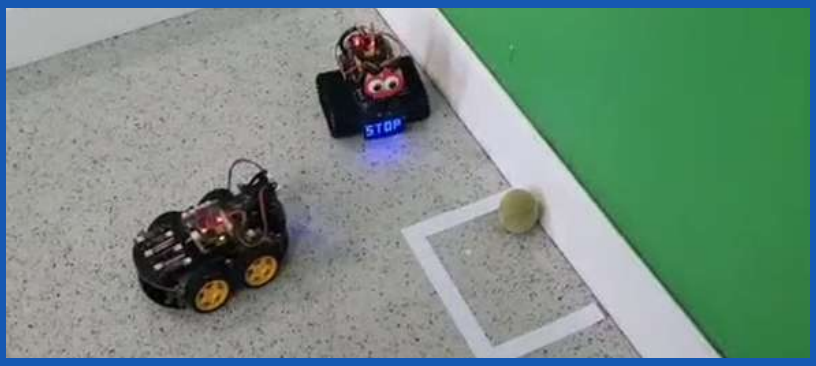
RESOURCES

ACTIVITY 4


ISTEDV


when clicked
forever
if key w pressed? then
run forward at speed 50
if key s pressed? then
run backward at speed 50
if key d pressed? then
turn right at speed 50
if key a pressed? then
turn left at speed 50
if key space pressed? then
run forward at speed 0

ACTIVITY 4



STUDENT'S EVALUATION

Areas in which our students will evaluate themselves:

1. Can I use Mblock software for Mbot?
2. Can I model with Tinkercad?
3. Can I use the project materials correctly and create the appropriate ground?

BIBLIOGRAPHY

https://ide.mblock.cc/
https://appinventor.mit.edu/

SCALABILITY

This project can be considered beginner-intermediate for fifth and sixth graders.

When Mbot sees an obstacle it stops by the distance sensor and is then directed by the phone to be able to hit the ball

ACTIVITY 5

TITLE Road signs with Makeblock

ABSTRACT

The use of technology is becoming an increasingly important aspect in everyday life. New professional figures and new work tools are designed to carry out innovative tasks.
The school's task is to update itself to this reality, but above all to provide pupils with the first bases and prepare them for the future. Today we will therefore see the importance of coding at school.
Being in step with the times is the key to being able to capture the interest of students and make them participate in the learning process.
Change and innovation are therefore a real responsibility for schools as well.
The first goal of coding is to teach children to develop an elastic mind and projected to effective solutions of simple problems

The students build a path and program the robot that follows the path in compliance with the road signs it encounters.

AUTHOR/S

IPS Maffeo Pantaleoni

DATE 14/09/2022
VERSION 1

DIDACTIC OBJECTIVES

- Knowing how to solve problems
- Knowing how to make decisions
- Creativity
- Critical sense
- Self-awareness
- Interpersonal skills
- Effective communication
- Emotion management

ACTIVITY 5

- Stress management
- Empathy
- Experience team work
- Encourage the integration of disabled students;
- Encourage the integration of foreign students;
- Foster a collaborative spirit;
- Stimulate creative thinking;
- Increase decision-making skills, a sense of responsibility and self-esteem;
- Develop the ability to analyze and solve problems;
- Improving a programming language.

- [ ] **SCIENCE**
- [x] **TECHNOLOGY**
- [ ] **MATHEMATICS**
- [ ] **GEOGRAPHY/HISTORY**
- [x] **LANGUAGES**
- [ ] **LITERATURE**
- [ ] **MUSIC**
- [x] **OTHERS: LAW AND CIVIC EDUCATION**

EDUCATION LEVEL

This activity is prepared to be completed by...

- [ ] **12 - 14 YEARS**
- [ ] **14 - 16 YEARS**
- [x] **OTHERS: 16 - 18 YEARS**

TOOLS NEEDED

- Makeblock robot
- Cardboard and markers
- Computer
- Road signs
- Software Mblocks

ACTIVITY 5

DEVELOP ACTIVITY

- Learning by doing
- Laboratory methodology
- Cooperative learning

PHASES OF THE PROJECT:

Phase 1 of familiarization / discussion they know the robot, observing, exploring, touching it, manipulating it.

Phase 2 of action-test / discussion they discover, in an autonomous way, the functionality of the robot, through trial and error.

Phase 3 of play / learning / discussion they program in blocks after being shown how it works, with practical examples. (Learning by doing)

Phase 4 of the didactic work through the planned of the road signs, they build the road signs perform specific procedures with blocks they write the procedures to make the robot execute a certain path.

Discussion phase

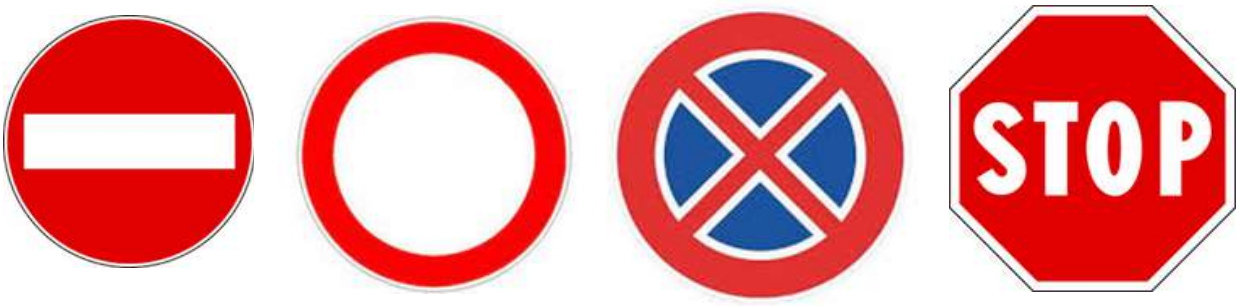
Sharing doubts and seeking solutions.

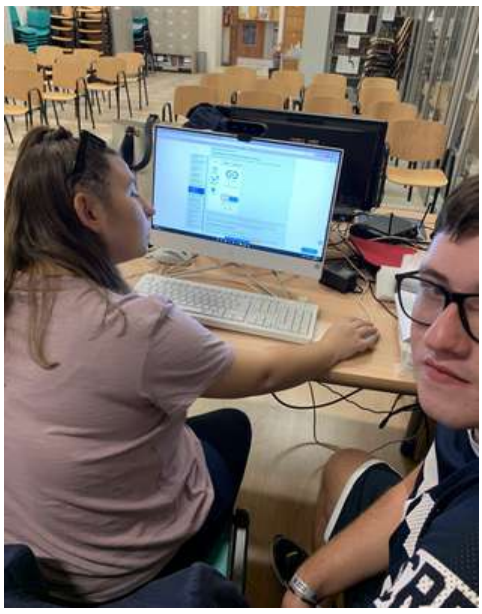
Active teaching

ACTIVITY 5

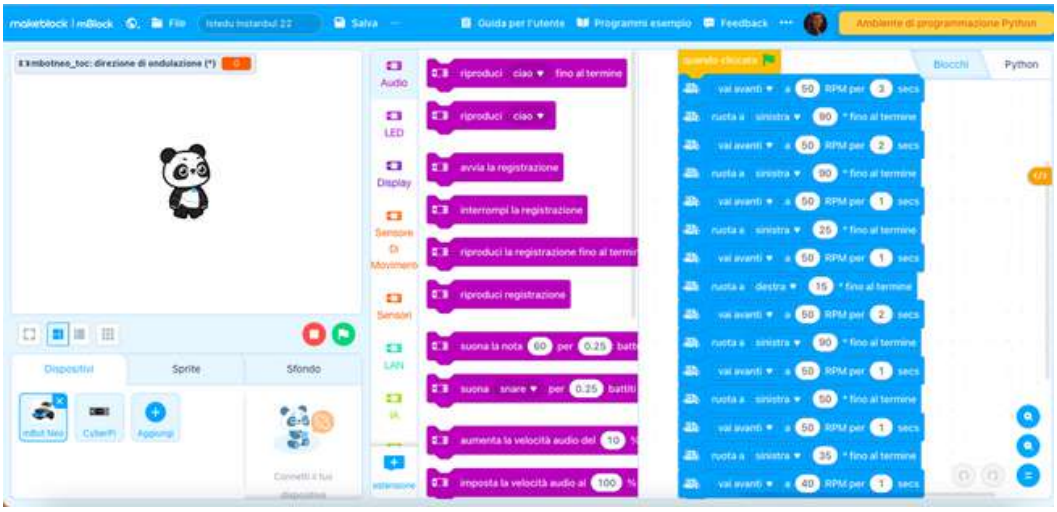
RESOURCES

Road signs: no entry, no transit, stop, no parking

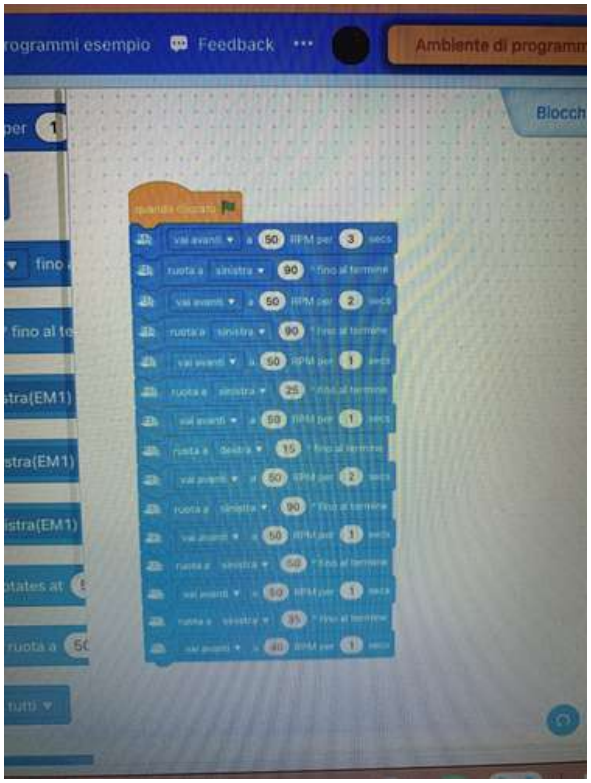




ACTIVITY 5


makeblock | mBlock
File
Salva
Guida per l'utente
Programmi esempio
Feedback
Ambiente di programmazione Python
mbotneo_toc: direzione di ondulazione (*)
Audio
LED
Display
Sensore Di Movimento
Sensori
LAN
IA
estensione
riproduci ciao fino al termine
riproduci ciao
avvia la registrazione
interrompi la registrazione
riproduci la registrazione fino al termine
riproduci registrazione
suona la nota 60 per 0.25 battiti
suona snare per 0.25 battiti
aumenta la velocità audio del 10 %
imposta la velocità audio al 100 %
Blocchi
Python
vai avanti a 50 RPM per 2 secs
ruota a sinistra 90 * fino al termine
vai avanti a 50 RPM per 2 secs
ruota a sinistra 90 * fino al termine
vai avanti a 50 RPM per 1 secs
ruota a sinistra 25 * fino al termine
vai avanti a 50 RPM per 1 secs
ruota a destra 15 * fino al termine
vai avanti a 50 RPM per 2 secs
ruota a sinistra 90 * fino al termine
vai avanti a 50 RPM per 1 secs
ruota a sinistra 50 * fino al termine
vai avanti a 50 RPM per 1 secs
ruota a sinistra 35 * fino al termine
vai avanti a 40 RPM per 1 secs
Dispositivi
Sprite
Sfondo
mBot Neo
CyberPi
Aggiungi

ACTIVITY 5



STUDENT'S EVALUATION

Assembly and installation of the various parts of the robot;
Help and collaboration between companions;
Programming of the different robot movements with blocks: straight line, left and right curve, stop, respect road signs.

BIBLIOGRAPHY

https://youtu.be/Lp_n77kuWy8
https://www.weturtle.org/dettaglio-tutorial/42/tutorial-motori-makeblock-mbot.html
https://ide.mblock.cc/
https://www.tinkercad.com/
https://education.makeblock.com/help/mbot2-start/

ACTIVITY 5

SCALABILITY

The robot can also be used with lower secondary school pupils (age 11-13)

ACTIVITY 6

TITLE Designing a version of the Space Invaders video game using Mblock

ABSTRACT

The game consists of moving a sprite (ship) in the horizontal plane and eliminating a series of objects (invaders) that move in formation from left to right and downwards. The ship sprite must avoid shots fired from a sprite moving in the upper horizontal plane. This sprite can be eliminated if it receives three hits. It must also avoid coming into contact with a third sprite that has a random movement and will eventually try to hit the ship sprite.

The game has object-defences that allow you to hide the ship sprite. These sprites will be "destroyed" when they are hit by a shot or come into contact with invading objects.

The game ends when all invading sprites are eliminated or when the ship controlled by the player is eliminated. This ship sprite is eliminated when it receives three hits or when the invading sprites reach a certain position on the Y-axis.

AUTHOR/S

IES Mediterraneo

DATE 26/03/2022

VERSION 1

DIDACTIC OBJECTIVES

To deepen in the development of programming with pseudo-code.
We will work on aspects related to:

- Uploading sprites
- Movements
- Use of variables
- Use of clones
- Use of sounds
- Dialogue creation

ACTIVITY 6

Programming strategy: A strategy based on Computational Thinking will be used. We will divide the global problem, the creation of a "Space Invaders" type video game, into smaller problems and a solution will be given to each of these problems to finally obtain a solution to the general problem.

The project can be extended by installing devices controlled by an Arduino board.

General problem: Creation of a "Space Invaders" type videogame with an aesthetic based on the Star Wars saga.

Partial problems:
1º Creation of backgrounds
2º Access to the game
4º Creation of main ship (X Wing)
5º Creation of attacking ship (Death Star)
6º Creation of secondary ship (Starfighter TIE Fighter)
7º Interaction of sprites
8º Incorporate sound
9º End of game

- [x] **SCIENCE**
- [x] **TECHNOLOGY**
- [ ] **MATHEMATICS**
- [ ] **GEOGRAPHY/HISTORY**
- [ ] **LANGUAGES**
- [ ] **LITERATURE**
- [ ] **MUSIC**
- [ ] **OTHERS .....**

EDUCATION LEVEL

This activity is prepared to be completed by...

- [ ] **12 - 14 YEARS**
- [x] **14 - 16 YEARS**
- [ ] **OTHERS .....**

ACTIVITY 6

TOOLS NEEDED

Necessary equipment

- Computer
- Mblock version 3.11

Material needed to extend the project

- Arduino board
- Protoboard board
- LEDs
- Connection cables
- Resistors
- Yoystick

DEVELOP ACTIVITY

TABLE OF CONTENTS

Step 1: Creation of backgrounds

Before we start programming the video game we need to clean it of objects.

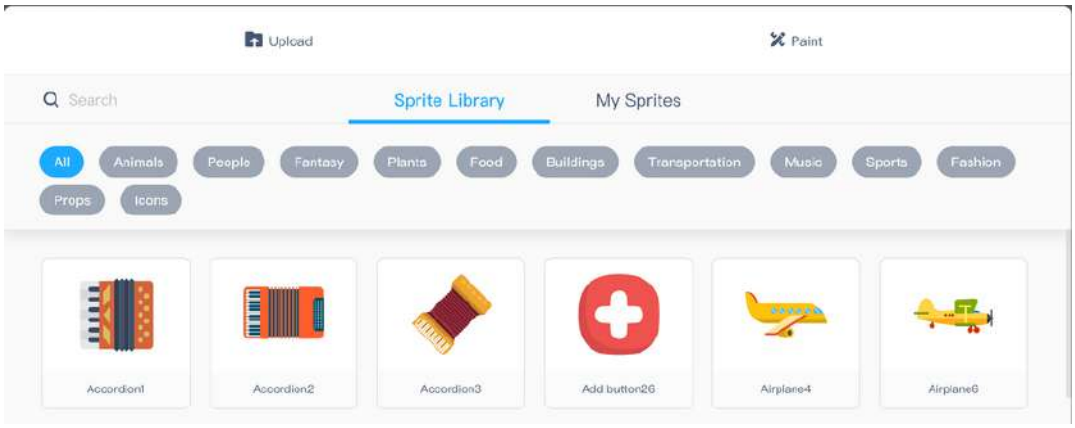
The game requires four backgrounds:

1. Presentation background
2. Introduction background
3. Game development background
4. Background of losing the game (Game Over)
5. Winning the game background ("Victory")

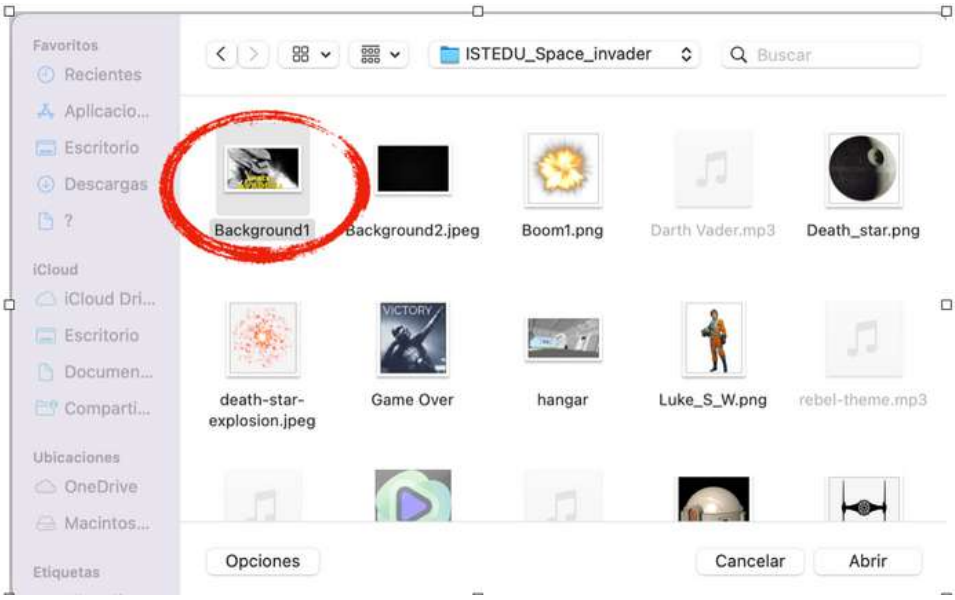
ACTIVITY 6

We will start by uploading the presentation background of the game. The rest of the backgrounds are made following the same procedure.

To upload the presentation background we will need an image that we have saved on our computer.
(In the supplementary material a url address will be added to a shared folder where the images and objects of the project are incorporated).
In the "Background" tab, click on "Costumes" and click on "Add a background".



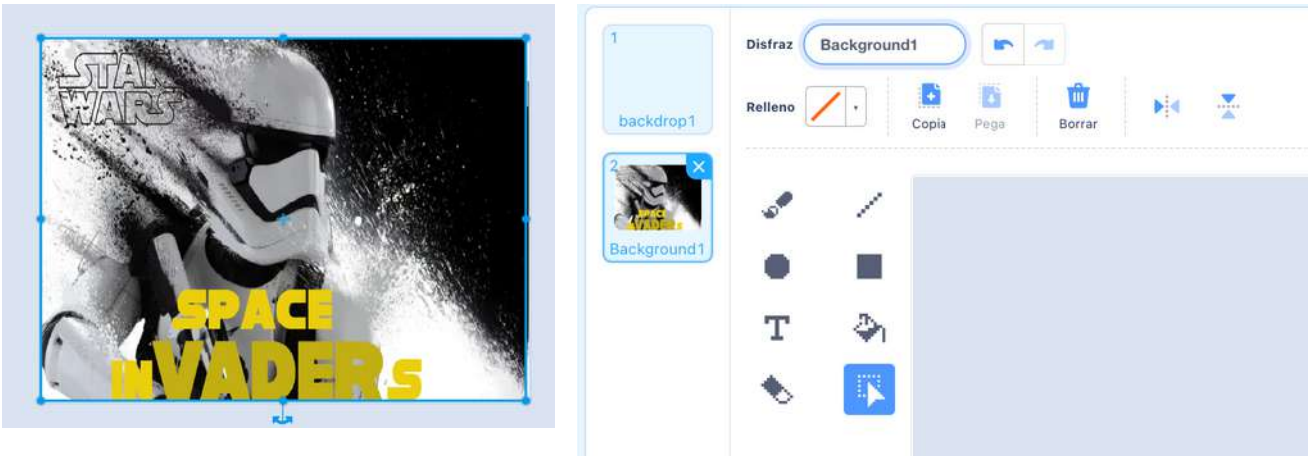
On the screen that opens, click on "Upload" and select the image from your computer.



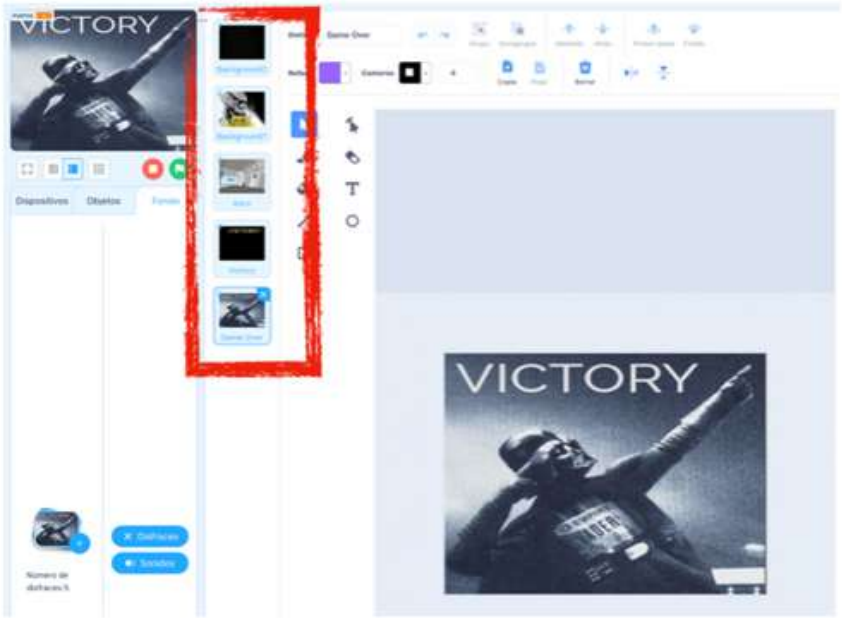
Select the image and click on "Open" and " Accept".

ACTIVITY 6

To occupy all the space of the background, we must select the image and select the whole image to enlarge it so that it occupies the whole surface. Finally we will give it the name "Background1" and we will eliminate the background that appears by default "backdrop1".



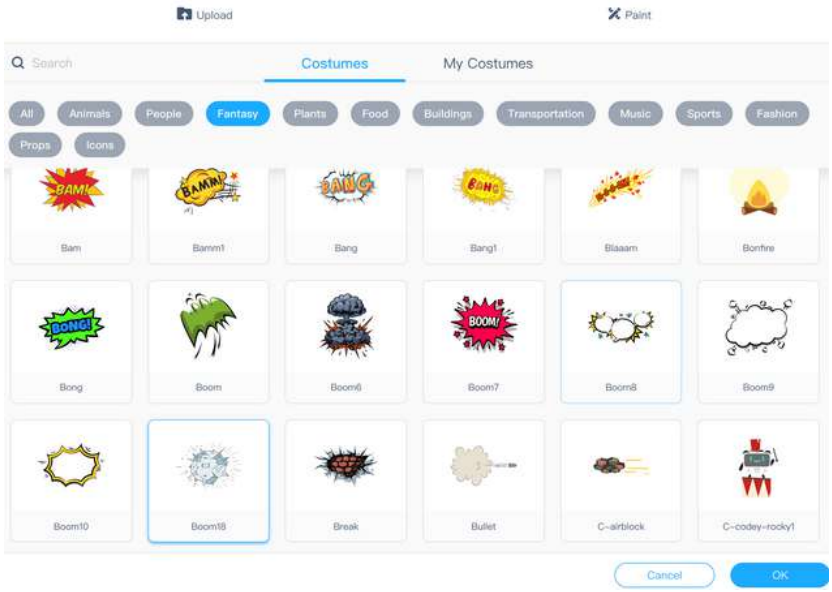
The rest of the backgrounds are made in the same way as above. It is very important to save the backgrounds with names that make programming easier. As we can see, we have 5 backgrounds that we will use in the programming of the video game.



Step 2. Creation of main ship (X-Wing) and access to the game.

To start programming the game we will have to upload the Bacground1 and the X-Wing sprite.
To upload the X-Wing sprite, which will be the ship we will control, we just have to go to the "Sprites" section and click on "Add".

We will upload the sprite and we will also add a disguise by selecting an explosion from the repertoire of sprites offered by M-Block. To do this, click on "Add disguise" and from the repertoire select the one that best fits your needs.

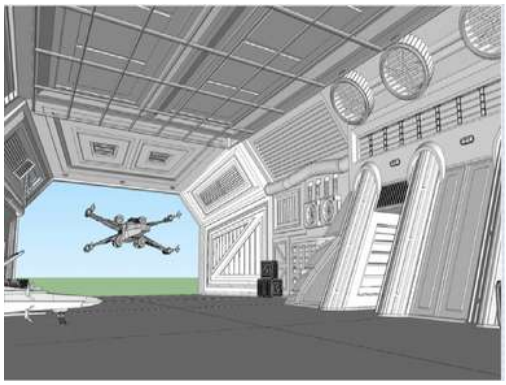


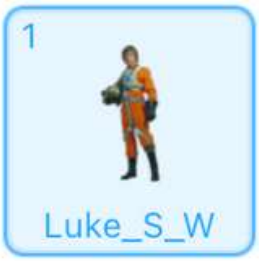
From the programming area in the X-Wing sprite we will start the game with this simple program. What we indicate is that when pressing "Green flag" the game starts with the background "Bacground1" and the sprite is hidden

ACTIVITY 6

Next we will switch to the "Background" tab and place this program so that after 4.5 seconds it will change to the "Enter" background.

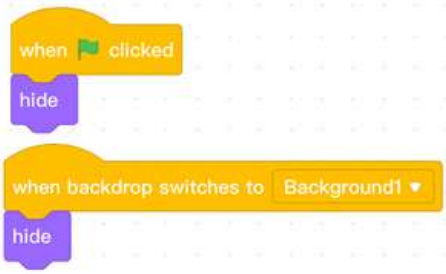
Background Intro





The next step is to upload the “Luke_S_W” sprite.

The first program to upload with each sprite is to indicate when we want it to be hidden or visible. In this case we will indicate that when pressing "green flag" and in the background "Background1" it will be hidden.

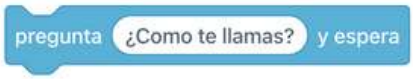


We will then indicate that when the background changes to "Enter" the sprite
- Will is visible
- We will set a size
- We will give it some coordinates in the X and Y axis to place it in the place we want in the background.

ACTIVITY 6

The next step is to enter a dialogue with the player to start the game. From the "Looks" section we will include an introductory sentence

From the section "Sensing" we will include a question



To continue with a new question in which we can include the player's answer we will need to join four blocks found in the "Sensing" and "Operators" section.

The aim is to take the answer with the player's name and ask a question directly to the player. For this we will link the blocks as follows.

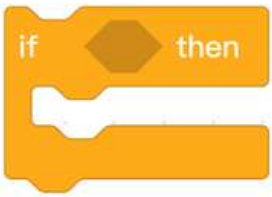
1st

2nd

3th

We will now link these blocks with the previous ones

ACTIVITY 6



Finally, all that remains is to join the player's new answer. If the answer is "yes" the game will start and if the answer is "no" the game will return to the beginning. For this we only have to use two conditionals from "Control".

These new blocks will be joined to the previous ones.

The end result should look something like this

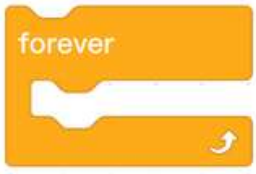
Place for response

The answer is taken with the player's name, and you are asked if you want to continue the game. If the answer is "yes" you will be welcomed and prompted to press the "space" key in order to play. This is very important because passing the "space" key is the command that allows the game to start.

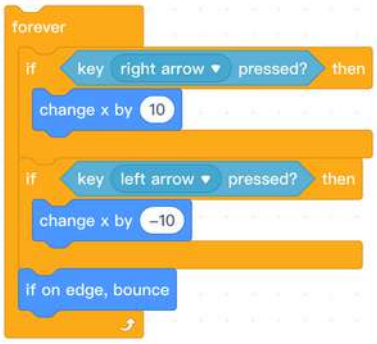
ACTIVITY 6

Finally, all that's left to do is to give the X-Wing a moving X-Wing.
The first thing we will do is call the object when the background changes to "Background2" and tell it to show up in the X-Wing costume, if we don't do this the object may show up in the "explosion" costume.

We call the object, tell it to be displayed in the corresponding costume, set a size and give it X and Y coordinates.



To give movement to the ship, use the left and right arrow keys. It is important to place conditionals in a "forever" structure.



By changing X by 10 units we are giving travel speed, so we can adapt the speed by modifying the value of X

This sprite will only be shown in the background where the game takes place, which is "Background2", so it will be necessary to indicate that in the rest of the backgrounds the object will be hidden.

Step 3. Creating clones.

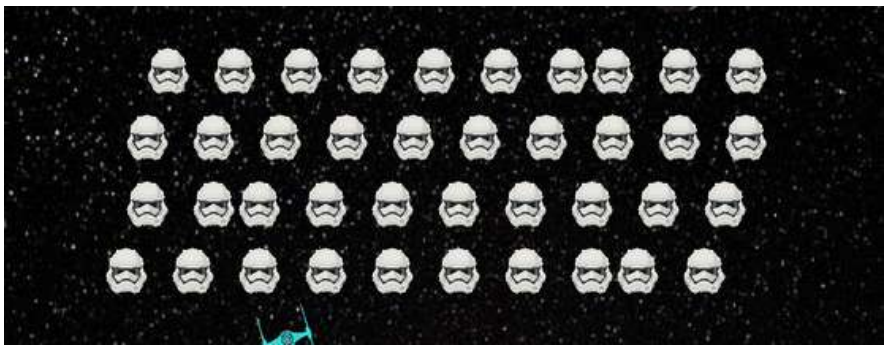
To create the "invaders" we need to upload the sprite "storm_troopers".

This sprite must be hidden in all backgrounds except the one in which the game "Background2" takes place.

The function of this sprite will be to defend the "Deatd_star" object and attack the defences and the X-Wing ship. To program this sprite and give it movement, clones of the sprite will be created. The clone creation process should start when the "space" key is pressed. By pressing this key we switch to the game background "Background2", set the size of the sprite and give it a position on the X and Y axes.

Now we can start creating the clones. First we will tell it to create four rows of sprites to the left of the background, and then we will tell it to create 10 clones in each of the rows by changing the X axis by 35 units to create them to the right and the Y axis by -35 to create them downwards to complete the four rows.

Everything should be put together in a single programming block. The result should look like this



A variant of the game would allow four independent rows of clones to be created and programmed so that their movement is better coordinated, but this complicates the game. This variant is included in the accompanying files.

Once the clones have been created, they will be given movement.

1st move to the right and wait
2nd move to the left and wait
3rd scroll down

The system is better ordered if instead of one object, four sprites are uploaded and the clones are created in four rows, but this requires programming each sprite separately. That will be the variant of the game that will be incorporated in the files.

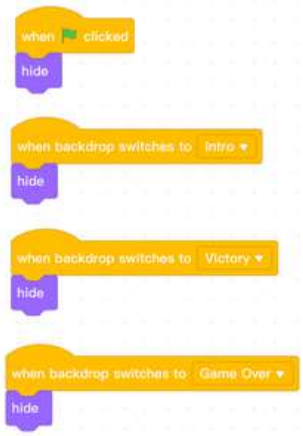
Paso 4. Creación de las defensas.

The creation of the defences requires uploading the sprite "Tatooiine_ Building" and adding 6 disguises to it. First we will upload the object and from the "Costumes" option we will make a copy of the sprite duplicating it. Each copy will have part of the drawing erased with the costume editor to simulate different phases of destruction.

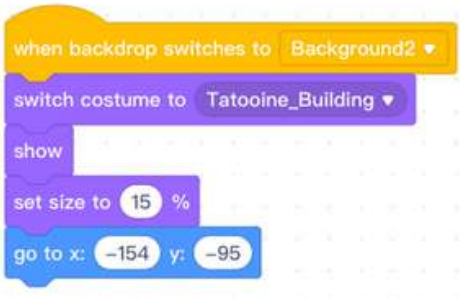
ACTIVITY 6

It is very important that the last costume of the sprite is a totally empty image.

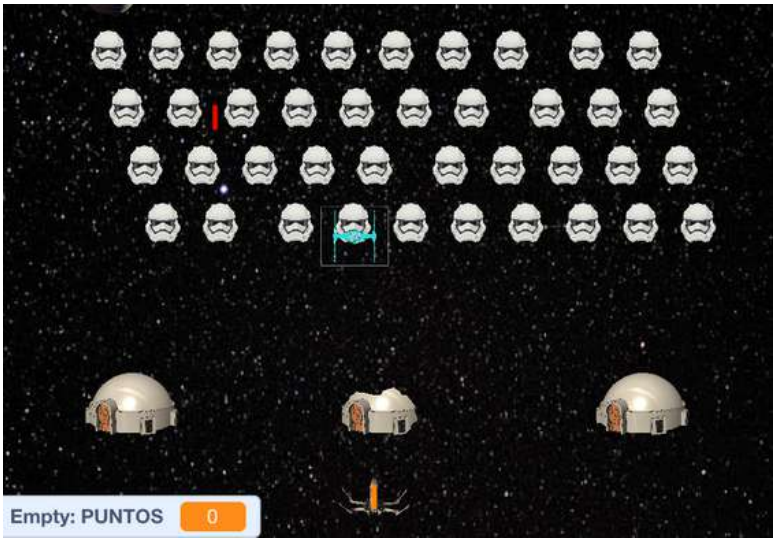
This operation must be repeated three times or as many times as the number of defenders we want to include in the game.



To program the defenses, start by indicating that they should not be displayed on backgrounds other than "Background2".
Finally, all that remains is to indicate that they appear in the game background and give them a location.

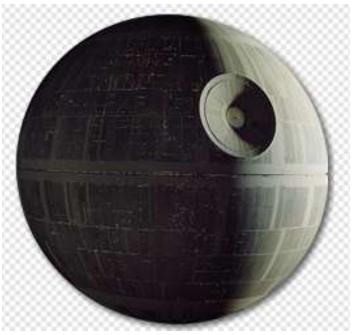


It is very important to indicate that at the beginning you switch to the first costume where the complete drawing is shown. Then set a size and coordinates at the bottom of the screen leaving the necessary space for the X-Win object behind the fenders. This program is dragged to each of the fenders changing only the value of the X-axis so that they are separated from each other at the same distance. In a next phase of the program we will see what happens when a laser hits the defenses. The result will be this.

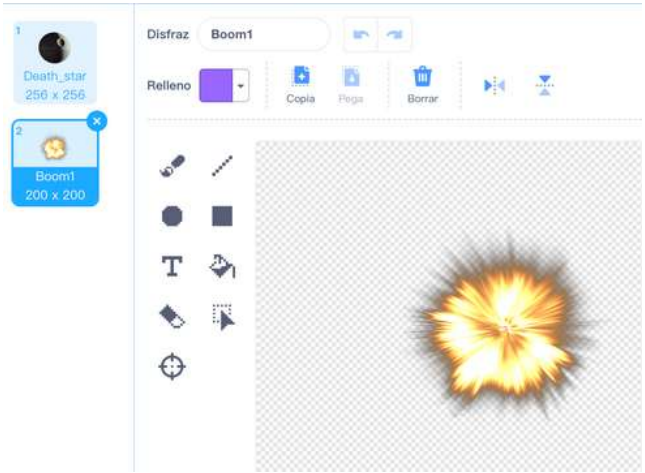


Step 5. Creation of the attacking ship (Death Star).

To create the attacking ship we will upload the Death_star sprite



To this sprite we will add a costume which will be an explosion, which can be taken from the repertoire of sprites offered by Mblock or upload our own explosion sprite.



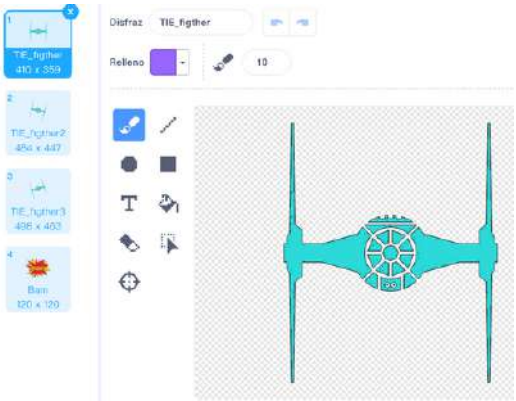
Again, we will indicate that the sprite is to be hidden in all backgrounds except in the "Background2" game background and we will give it movement. The function of this sprite is to move from left to right and from right to left by firing a laser.

It is very important to indicate that the costume in which it is displayed is "Death_star" to avoid it appearing as an explosion.
We set a size and coordinates on the X and Y axes. Finally, within a "forever" structure, we indicate the movement to the right and left by sliding the sprite.

Step 6. Creating the secondary ship (TIE-figther)

We must upload the sprite "TIE_figter".

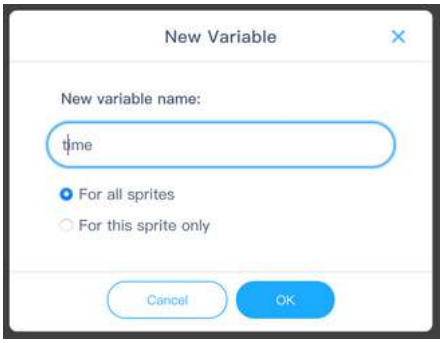
Three more costumes have been added to this sprite. Two of them are the left and right leaning sprites and one more explosion costume.



This sprite will come out of the Death Star and will
have random movements and finally aim at the X-Wing ship and crash into it. The objective of the game will be to shoot this sprite down or avoid hitting it.

ACTIVITY 6

We will start by saying that the sprite is hidden in all backgrounds and that it is presented in the costume “TIE-figther”.



For the sprite to appear in the game it is necessary to create a variable that we will call "time". From the "Variables" section, click on "Create a variable", name it "time" and OK.

Pressing the "green flag" will set the time to 0 and create a timer when the space key is pressed.

To create the timer, within a "forever" structure we will indicate to wait for one second and to change the value of the time variable to 1

Step 7. Interaction of sprites

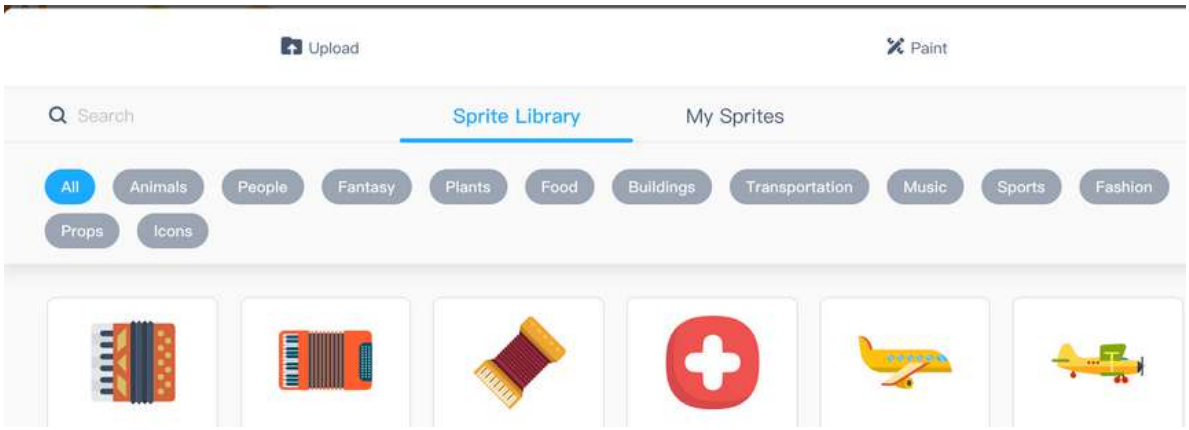
In this section we are going to see how the sprites interact for the development of the game.

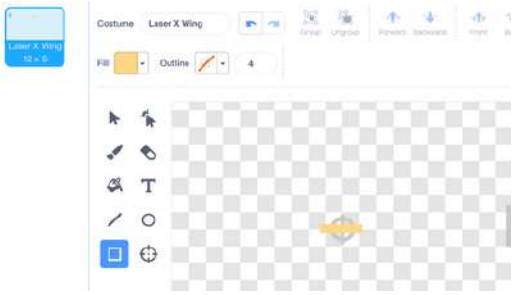
We need to create two sprites that will act as lasers, one for the attacking ship (Death-stat) and one for the X-Wing ship.

Create lasers for X-Wing and for Death_star

From the Objects section, click on "Add" and then on "Paint".

ACTIVITY 6





From this screen we will draw a rectangle with a colour that allows it to stand out on a black background. It is important to place the object in the centre. We will name it "Laser X Wing.
We will perform the same operation to create a second sprite of a different colour which we will call "Laser Death Star".

The two sprites must be hidden in all backgrounds of the game and will be shown when pressing the "space" key.

Laser X-Wing.

This laser is fired from the X-Wing ship and will allow you to eliminate the clones (Storm-troopers), the Death Star ship, the auxiliary TIE_Fighter ship and the defences. It will also allow you to score points for
each Storm_troopers clone eliminated.

It is necessary to create a variable that we will call "PUNTOS" that we will set to 0 and a clone of this object.

ACTIVITY 6

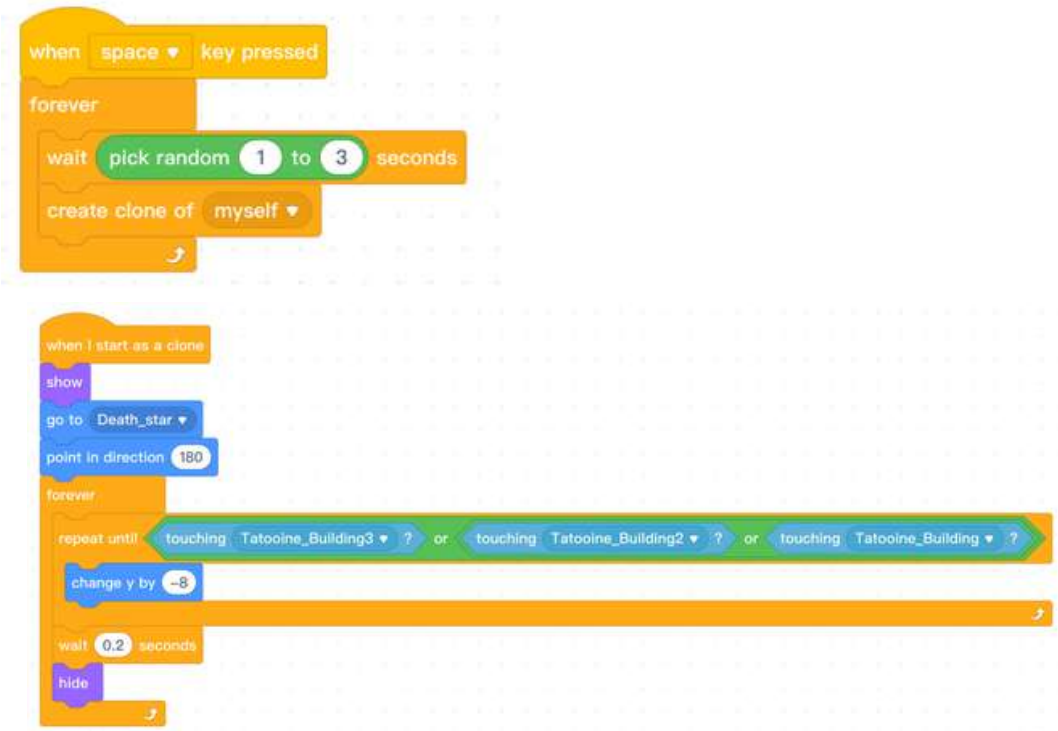
To give the sprite movement we will attach it to the X-Wing ship so that the shot always comes from the ship when pressing the 'a' key. The movement is created by changing the Y-axis by 12 units and is repeated until the object touches an edge.

To interact with the defences we will use the programming block "send a message". The strategy is that when it touches a defence it sends the message “stop" (detener) and when it receives the message it hides. It is necessary to programme what happens when it touches each of the defences.

Laser Death_star

At the beginning of the programming we will indicate that the sprite will be hidden in all the backgrounds of the game. This sprite is fired from the Deadth_star ship, so we will have to fix the sprite to the ship. In addition, to make the game more interesting and to ensure that the firing frequency is not always the same, we will indicate that the firing is random between a random number between 1 and 3. This frequency can be modified by extending the range between the numbers

We will add a speed by modifying the Y-axis by -8 units and this downward movement is repeated until it touches one of the defences. When that happens it hides



We indicate that the object should be attached to the "Death Star" object so that shots are always fired from the ship. Here we must also indicate what happens when the object hits the defences. The object is moving with a Y-axis value of -8 until it hits the defences and hides.

Clones Storm_troopers

These sprites move sideways and downwards. Their function is to defend the Death_star ship, break through the defences and if they reach a certain position, end the game by defeating the player. Defeat also occurs if any of the clones touch the X-Wing ship.

When one of these sprites touches the X-Wing laser it will add a point to the player. It is important that the laser sprite stops when a clone is hit to prevent it from continuing its upward trajectory and eliminating more clones.

ACTIVITY 6

When it hits the Laser X Wing object it adds up to one point

When it reaches a position on the Y-axis (-152) the background will change to Game Over and the game will stop.

Defenses Tattoine_Building

We will program one defence and then copy the programme to the other two.
At the beginning we will say that when we press the green flag and in the backgrounds Enter, Game Over and Victory it will be hidden.
The defences are destroyed when they touch any of the "Laser" objects. Each time a laser touches the defence it changes its disguise until it reaches the last one, which is an empty image

If the defence touches a Storm_troopers clone, the defence is directly hidden..

X-Wing ship

The X-Wing ship will be eliminated and the game will be over when:
A. Receive three laser hits from the Deatd star ship's laser.
B. Touches any of the Storm Troopers clones.
C. Hits the TIE-Fighter ship..

ACTIVITY 6

Each of these options must be programmed separately. To program option A) we must create a variable in which the impacts received will be stored. This variable will be called "X Wing" and its value will be set to 3

The Death Star's laser impacts will be programmed from the "Laser Death Star" sprite.

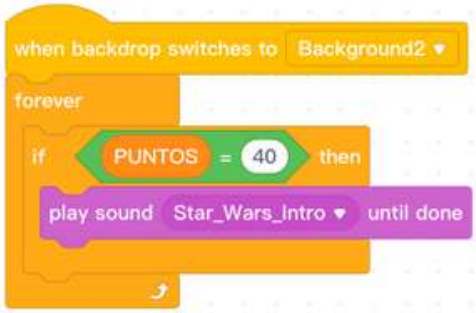
What we indicate is that if the laser hits the X Wing it will change the variable to -1. To avoid that once the sprite continues to subtract points as it moves, we say to wait 0.25 seconds. This clone will hide when it touches an edge.

What remains now is to indicate what happens when the X Wing variable has a value of 0 points..

We indicate that if the value of the variable = 0 to change the explosion disguise and to change the background to Game Over where the game ends.
The change in the size of the disguise is necessary to adapt it to the game.

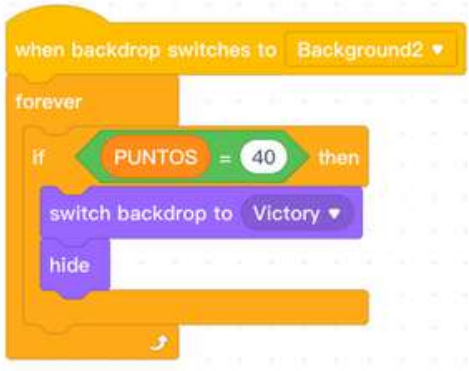
ACTIVITY 6

Option B) is to be programmed in parallel to the end of the game. The game ends when a score of 40 points has been reached, which is the number of clones that appear at the start of the game.



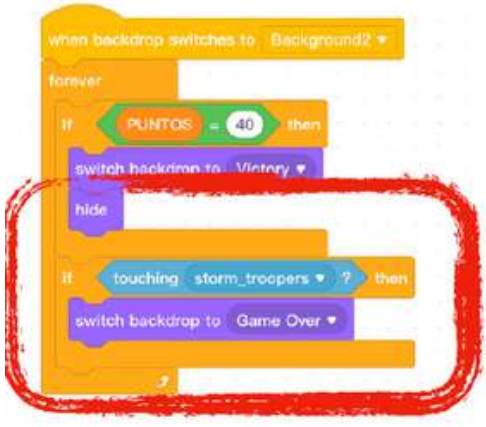
You have to upload a Star Wars Intro theme sound to play when you reach 40 points. If, on the other hand, the X Wing ship is hit by a clone, the background will change to Game Over and the game will end.

This second part is important to place in parallel because we want the two programmes to run at the same time and not one after the other.

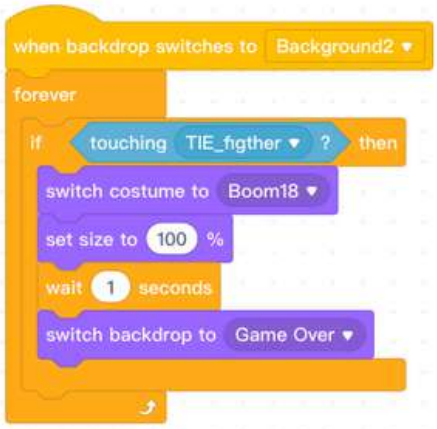


What we say in this part of the programme is that when 40 points are reached, change the background to "Victory" and hide the ship.

We can use these two programmes to indicate what happens in option B), when a Storm Troopers clone hits the X Wing ship and thus save programmes. The situation would finally look like this:



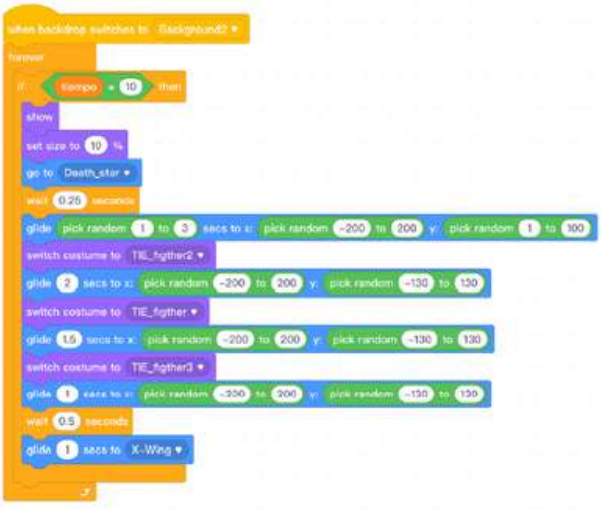
In option C) the programme is similar



We only indicate that the game ends when the TIE Fighter hits with X Wing.

TIE Fighter ship

To program this ship, a variable called “tiempo" (time) was created to determine when this ship would start to act. In this case we have assigned a value of 10 seconds. The ship will lock onto the "Death Star" sprite and display itself, wait 0.25 seconds and start its movement. What we want is for the ship to display a series of random movements before heading towards the X Wing ship to hit it. The aim of the game will be to avoid impact or shoot it down with a laser shot.



These are the random movements. This way, every time the game starts, the ship will make unpredictable movements. When it has finished making these movements, it will head towards the X Wing ship.

ACTIVITY 6

In this section we will indicate that if the ship touches an X Wing laser it will change its disguise to an explosion and hide.

In the event that we avoid the impact and do not shoot down the ship, it will hide 20 seconds after the movement starts, thus preventing the Neve from staying in the game.

It is important to ensure that the ship remains hidden on all backgrounds in the game and is only shown when instructed to do so.

Death Star Ship

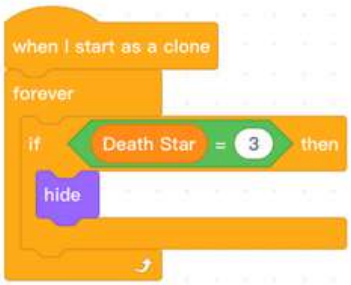
When we uploaded the object "Death Star" we already indicated its movement, what we will do now is to program its elimination from the game. What we will do is create a variable called "Death Star" whose function will be to store the hits it receives. It will be programmed so that the ship is eliminated when it receives three hits.

It is important that we set the value of the variable to 0 when we create the ship.

ACTIVITY 6

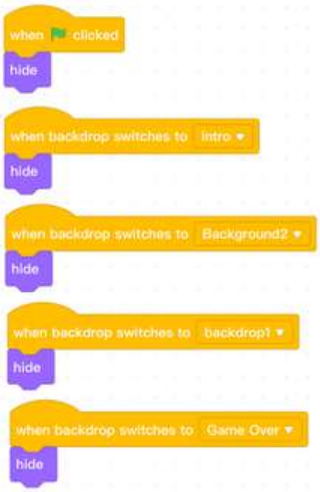
We indicate that if touching laser changes the value of the variable by 1

We indicate that when the variable has a value of 3 it changes its costume to explosion, waits a second and hides.

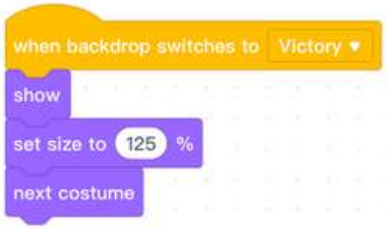


It is very important to go back to the programming of the "Death Star Laser" object and indicate that when the ship is eliminated the laser stops firing.

Sprite "Victory"



This is an optional item that is placed at the end of the game when you have reached 40 points. It is a gif item with 33 costumes. The object has been collected from the web and as it is an optional object it can be avoided. Its programming is as follows:
We hide it on all backgrounds in the game and only when the background changes to the "Victory" background is it shown and prompted to change costume.



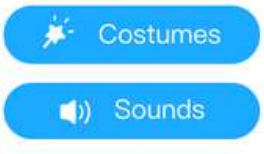
ACTIVITY 6

Very important note:

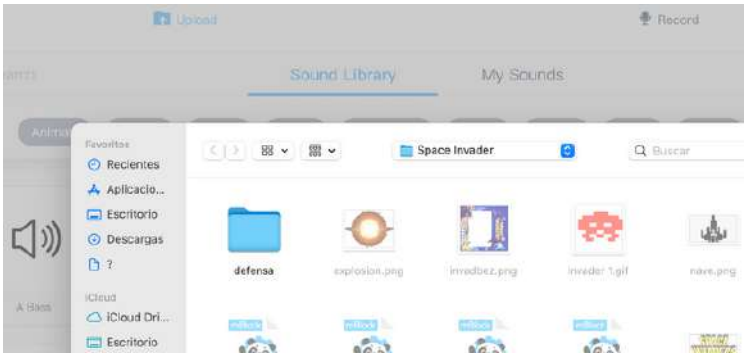
All variables that have been included in the game must be hidden except the one that indicates the points we are reaching. We can select the variables to be shown but they take away the visibility of the game.

Step 8. Incorporate sounds

To add sounds to the video game, simply select "Sounds" from the "Sprites" tab

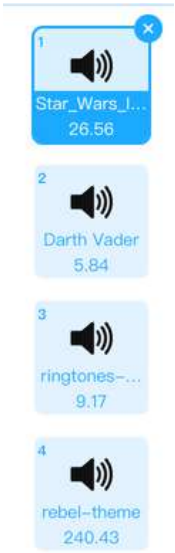


Then select "Add sound".



And click on "Upload". We access our folders on the computer and select the sounds we want to incorporate into the game.

ACTIVITY 6



In our game these are the sounds that are incorporated

From the "X Wing" sprite we can incorporate sounds

ACTIVITY 6

RESOURCES

STUDENT'S EVALUATION

To be assessed:

- Basic use of Mblock
  - Uploading sprites
  - Creating backgrounds
  - Creating sprites with costumes
  - Showing and hiding sprites in the appropriate sequence
  - Basic movements with sprites
  - Using conditional structures
- Designing an overall project strategy
- Break the project into simpler tasks
- Reviewing project versions and correcting possible errors
- Create clones of sprites
- Working with variable

BIBLIOGRAPHY

Marjil, Made (2014). Learn to Program with Scratch: A Visual Introduction to Programming with Games, Art, Science, and Math

Morris, Mike (2019). Scratch Programming: An In-depth Tutorial on Scratch Programming for Beginners: 1

Sweigart, Al (2021). Scratch 3 Programming Playground: Learn to Program by Making Cool Games

Yuqiang, Liao & Tongzheng; Zhao.

Mblock Kid maker. Rocks with the robots (PDF)

Tutorials

SCALABILITY

The video game allows to replace the movement of the X Wing ship by means of keys with a Joystick that can also be used as a trigger.
LEDs can be added to indicate the hits the X Wing ship has received.
A button can be included to fire the X Wing's lasers.

MORE INFORMATION

Link to the folders with the objects to design the software and the games

ACTIVITY 7

TITLE Mbot Racing

ABSTRACT

Competitions prove success. We reward success with these competitions.

In school life, competitions have a special importance for students. Because instructors can use competitions to increase the permanence of their lessons.

In this project, we wanted to have fun and coding our Mbots.

Our project is based on the fact that the students answer the questions that were previously arranged with the codes on their mobiles, and the Mbot uses for each correct answer makes a little progress. The first who reaches the finish line by Mbots win the competition.

AUTHOR/S

Sultantepe Prof. Dr. Cemil Taşçıoğlu Ortaokulu

DATE 24/02/2022

VERSION 1

DIDACTIC OBJECTIVES

With this study, the student;

- Knows how to write code in Mblock program.
- Knows how to use Kahoot software.
- Develops hand-motor muscles.
- Knows how to use Bluetooth controlled mbot.
- Learns by having fun.
- Develops General Culture.

ACTIVITY 7

- [ ] SCIENCE
- [x] TECHNOLOGY
- [ ] MATHEMATICS
- [ ] GEOGRAPHY/HISTORY
- [ ] LANGUAGES
- [ ] LITERATURE
- [ ] MUSIC
- [ ] OTHERS .....

EDUCATION LEVEL

This activity is prepared to be completed by...

- [x] 12 - 14 YEARS
- [ ] 14 - 16 YEARS
- [ ] OTHERS .....

TOOLS NEEDED

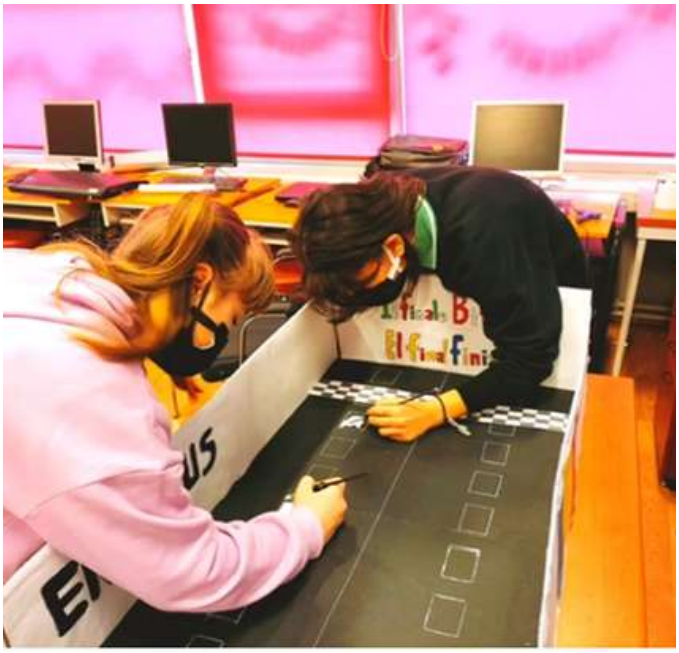
Required material list

- Computer
- Mblock online version
- Kahoot program
- Mobile devices
- Bluetooth Mbot
- Glue, cardboard, scissors
- Paint, Brush

ACTIVITY 7

DEVELOP ACTIVITY

Step 1: Creating the Competition Platform



Step 2: Preparing to compete with Kahoot

In this project, our aim is not only to study robotics, but also to reach the subject gains. For this reason, our students prepare for the competition by making pre-tests on Kahoot for the questions to be asked.

ACTIVITY 7

Step 3. Questions are prepared in the Mblock program.

RESOURCES

ACTIVITY 7

STUDENT'S EVALUATION

They will be evaluated:
- Student's Kahoot performance.
- Student response time to Mblock questions
- Number of correct answers given by the student to Mblock questions
- Student's time to end point
- The robot can follow a straight line, a right-hand curve or a left-hand curve.

BIBLIOGRAPHY

https://www.makeblock.es/productos/robot_educativo_mbot/?
https://kahoot.com/

ACTIVITY 7

SCALABILITY

Since the study was carried out within the scope of the project, a short platform was prepared. Longer platform can be created.

More enjoyable competitions can be made with more Mbots.

MORE INFORMATION

No sensors other than Bluetooth were used in the project. Those who want to contribute to the project

1. By using the LEDs on the Mbot, it can be ensured that the Mbot responds to correct and incorrect answers with green/red colors.
2. Mbot will move more straight if it is supported with line tracking codes and sensor
3. In the event that the Mbot reaches the end point, it can be ensured to play the music determined by the codes.
4. By placing an ultrasonic distance sensor in front of the Mbot, it can be prevented from hitting the wall/cardboard when it reaches the end point.

ACTIVITY 8

TITLE Construction and programming of a line-following robot

ABSTRACT

Technology is becoming increasingly necessary in our lives, and it is to be expected that it will become even more so in the future. Therefore, it is necessary that our students begin to learn about it from an early age.

The aim of this activity is for students to build and program a line-following robot, to build their own circuit and finally, to personalise the robot.

AUTHOR/S

IES MEDITERRANEO

DATE 10/02/2022

VERSION 1

DIDACTIC OBJECTIVES

The following objectives are to be achieved:

- To become familiar with 3-D design and all the possibilities it entails.
- To design 3-D objects.
- To print 3-D objects.
- To promote interdisciplinary learning and teamwork.
- To improve the motivation and interest of the pupils.
- Improve spatial vision.
- To increase creativity.
- To plan a project from the beginning.
- Improve problem-solving skills.
- Installation and programming of electronic components.

ACTIVITY 8

- [ ] SCIENCE
- [x] TECHNOLOGY
- [ ] MATHEMATICS
- [ ] GEOGRAPHY/HISTORY
- [ ] LANGUAGES
- [ ] LITERATURE
- [ ] MUSIC
- [ ] OTHERS .....

EDUCATION LEVEL

This activity is prepared to be completed by...

- [ ] 12 - 14 YEARS
- [x] 14 - 16 YEARS
- [ ] OTHERS .....

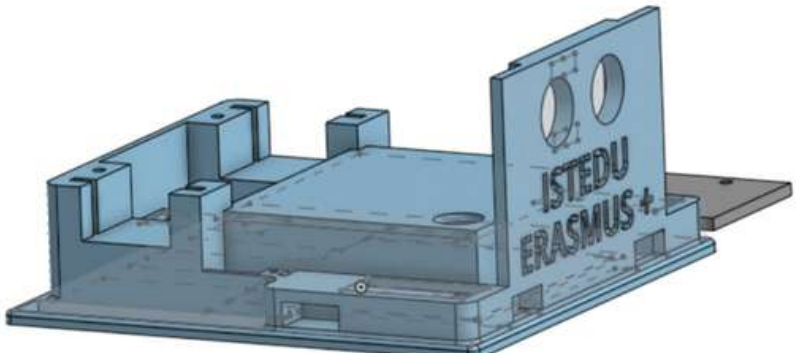
TOOLS NEEDED

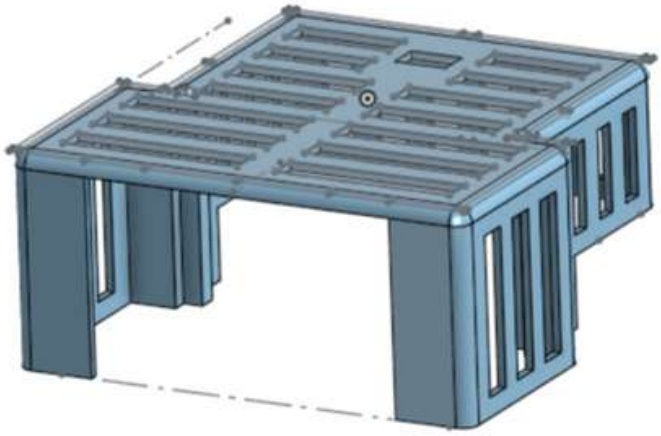
The following materials are needed

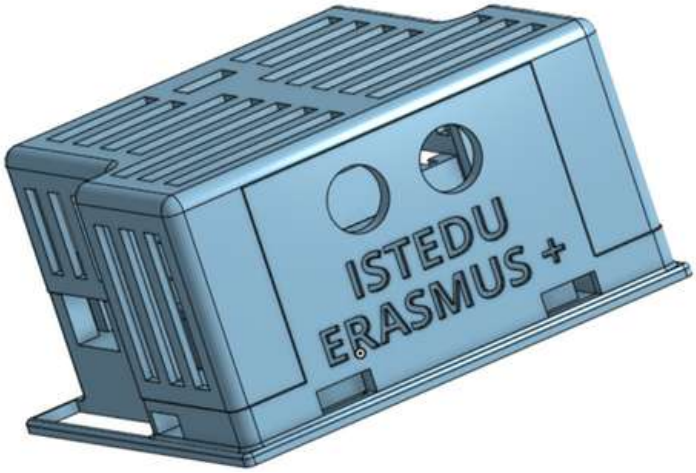
- Computer.
- Mblock version 3.11
- 3 D design software
- 3 D printer
- Arduino board
- Shield for Arduino board
- Servo 360 sg90
- HC-SR04 distance sensor
- TCRT5000 line sensor
- Wheels
- F-F cables
- Battery holder

DEVELOP ACTIVITY

Step 1: 3D printing the parts







Step 2. Connect all components, following the diagram below:

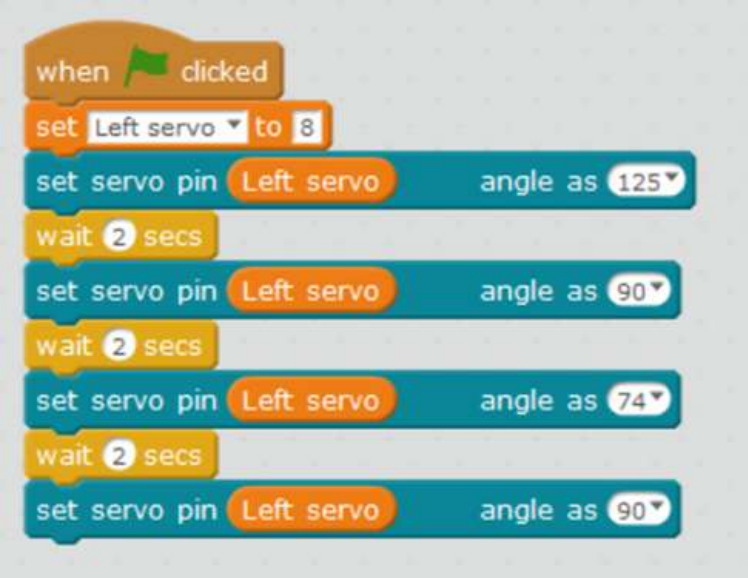
- Right motor to digital pin number 9
- Left motor to digital pin number 8
- Left infrared sensor to digital pin number 7
- Right infrared sensor to digital pin number 6
- Ultrasonic sensor or distance sensor: Trig to digital pin number 13 and Echo to digital pin number 12.

Step 3. Once all components are connected, check one by one that they are working properly.

- Infrared sensors:
  - We open MBlock and introduce the following program in Mblock mode.

  - We place the sensor, which we are going to test, on a blank sheet of paper with black lines. For it to work correctly, it must give 0 when it is on white and 1 when it is on black.
  - We do the same for the other sensor.

- Servos:
  - We open Mblock and upload the following programme:



- The left engine must rotate in one direction for 2 seconds, stop for another 2 seconds, rotate in the opposite direction and finally stop.
- The same must be done with the other engine.

Step 4. Print out the circuit sheets and stick them on the floor or on a rigid surface.

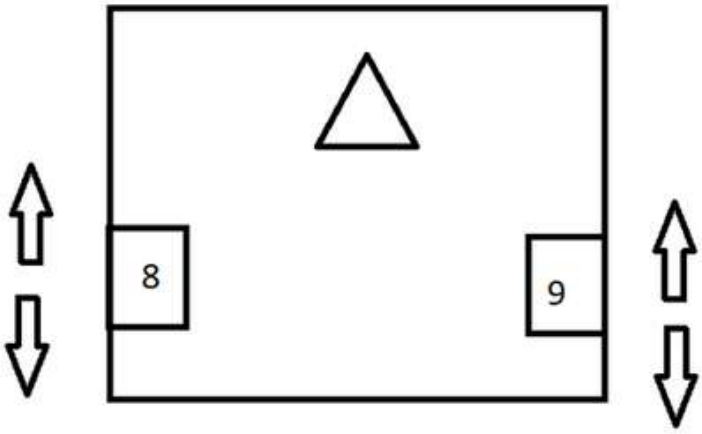
Step 5 Program the robot to follow the line.

This is not a difficult task, but if you try to approach it without splitting it into different parts it can be difficult.

1. Programming movements

- Program it to walk straight. To do this, the rotation angles of the two servos must be adjusted. Using a diagram similar to the following, indicate the angle of rotation so that it moves in each direction. It is convenient to adjust the angles for different speeds.

ACTIVITY 8



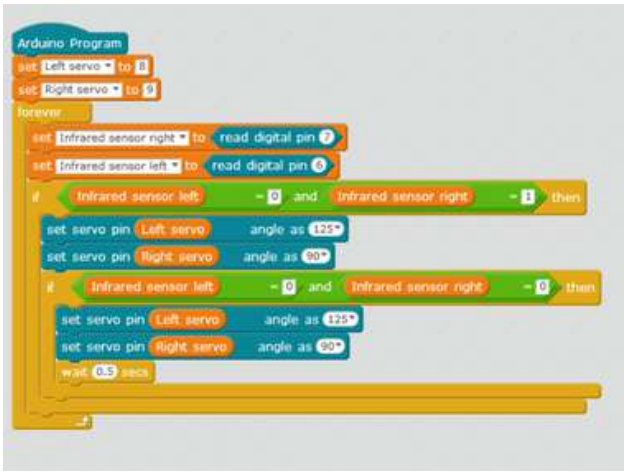
- Program it to turn right, you have to adjust the angles of the two servos, it would be convenient to try different speeds.
- Program it to turn to the left, the angles of the two servos need to be adjusted, it would be convenient to try different speeds.

2. Follow part of the line

Now we must programme the infrared sensors and the robot's movements together.

When the robot is on the circuit, at least three situations can occur:

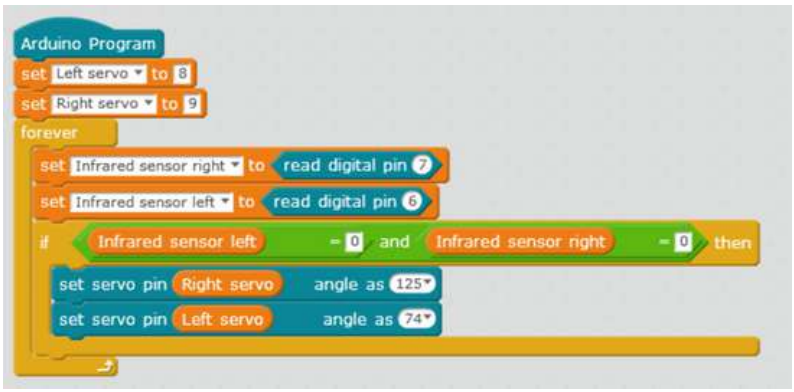
The left sensor is on the black line and the right sensor is on the white line, the robot must turn to the right..

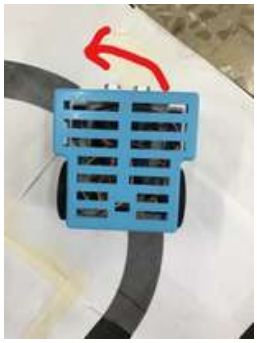


It is possible for you to go ahead and continue to the inside of the curve, to return to the inside you must keep turning right.

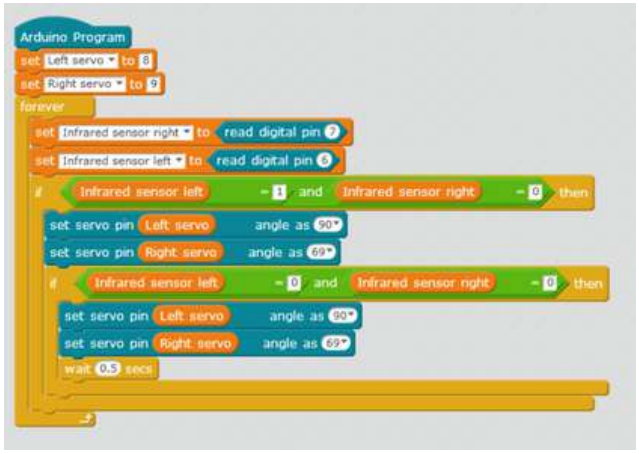
ACTIVITY 8

The two sensors do not detect the black line, so it must move in a straight line.





The right sensor is on the black line and the left sensor is on the white line, then the robot must turn to the left.

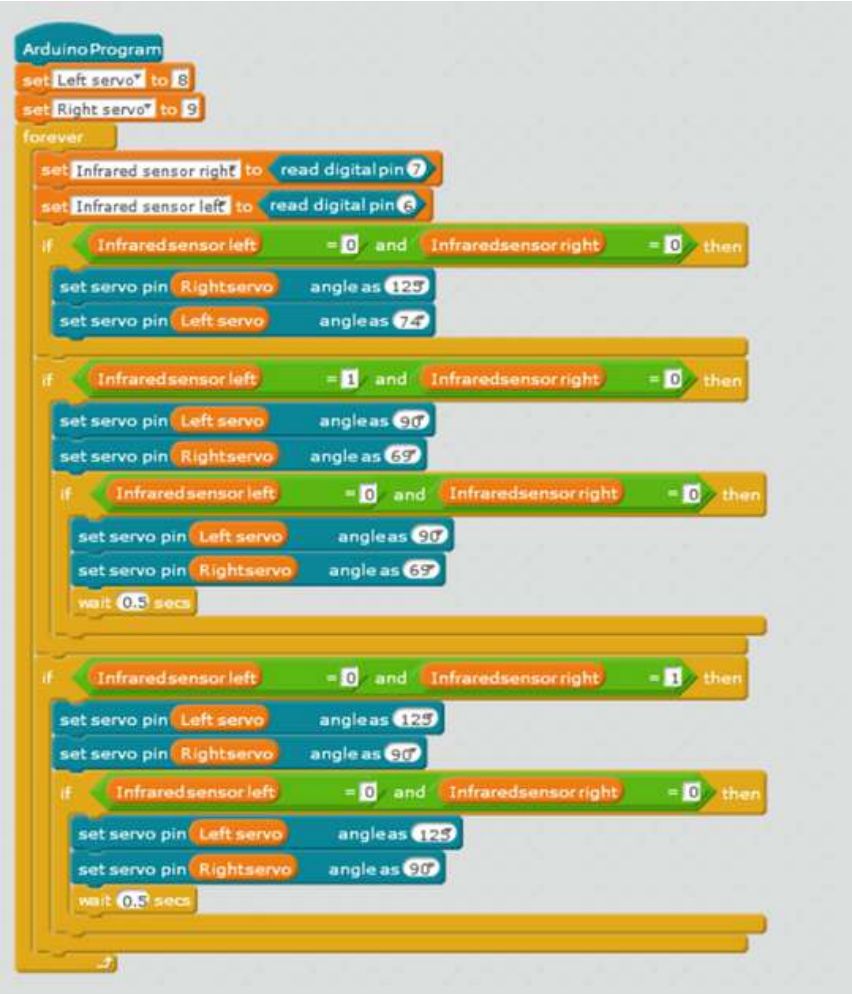


It is possible for it to go ahead and continue to the outside of the curve, and to return to the inside it must continue to turn left.

It is advisable to test each of these situations separately, in order to know if each of them has been programmed correctly.Seguir todo el circuito.

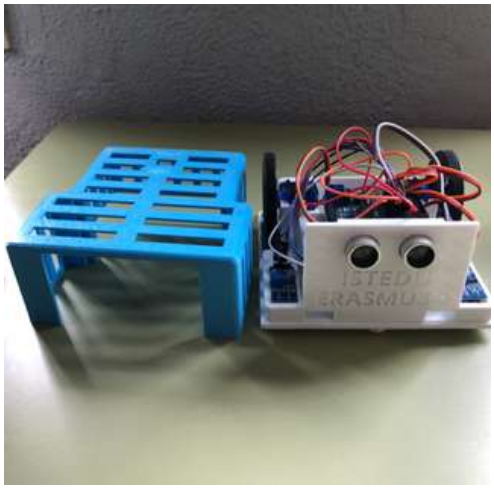
3. In this case, we must programme the situations of the previous section together, trying to do the circuit slowly and increasing the speed so that it is done as quickly as possible.

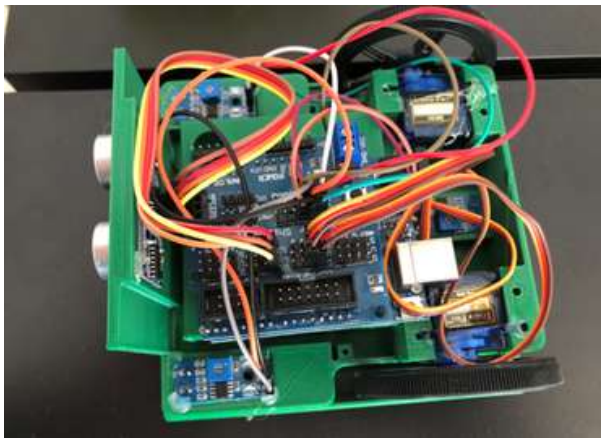
ACTIVITY 8

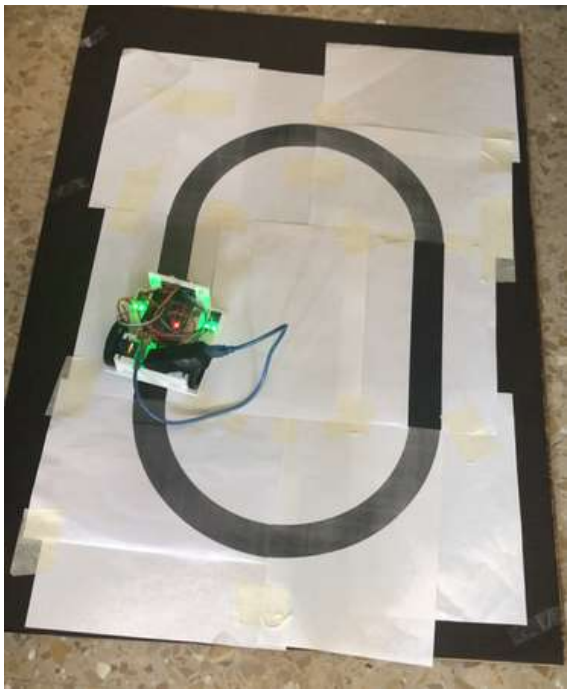


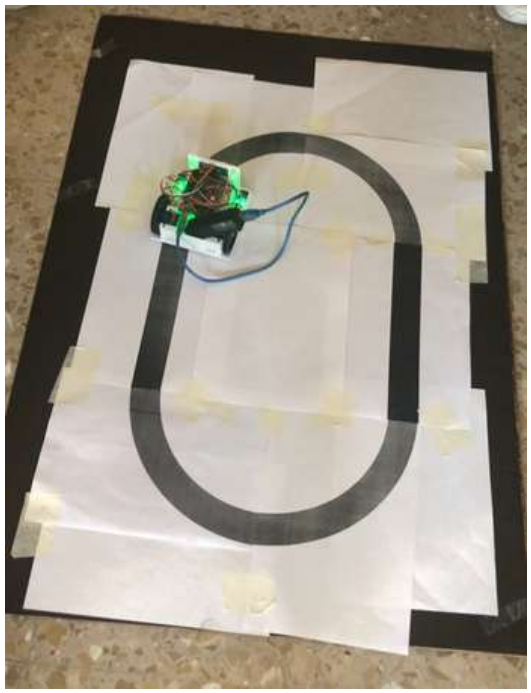
RESOURCES

ACTIVITY 8









STUDENT'S EVALUATION

It will be evaluated:

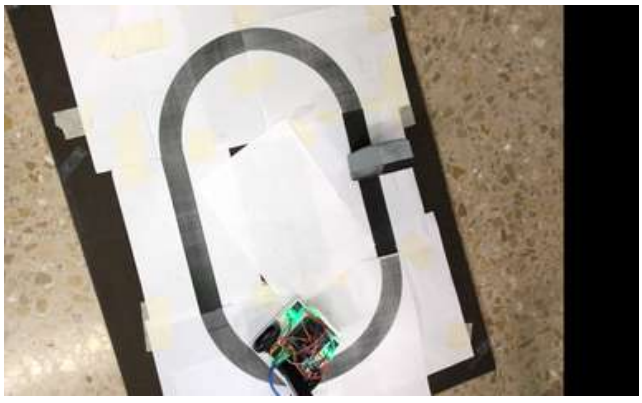
- Printing of the different parts of the robot.
- Installation of the electronic components.
- Correct operation of all the electronic components by means of the individual programming of each one.
- Programming of the different movements of the robot.
- The robot can follow a straight line, a right-hand curve or a left-hand curve.
- The robot can make the whole circuit without going off.
- The robot can make the circuit as fast as possible.

BIBLIOGRAPHY

https://www.arduino.cc/

SCALABILITY

Obstacles can be added along the way for you to avoid them and return to the line.



It can be programmed so that it does not leave the circuit, i.e. when it detects the black line, it turns inwards. Another option is to make a sumo, with glasses inside the circuit and it must lead out of the circuit.

ACTIVITY 9

TITLE Construction and programming of a light following robot

ABSTRACT

Technology is becoming increasingly necessary in our lives, and it is to be expected that it will become even more so in the future. Therefore, it is necessary that our students begin to learn about it from an early age.

The aim of this activity is for students to build and program a line-following robot, to build their own circuit and finally, to personalise the robot.

AUTHOR/S

IES MEDITERRANEO

DATE 10/02/2022

VERSION 1

DIDACTIC OBJECTIVES

The following objectives are to be achieved:

- To become familiar with 3-D design and all the possibilities it entails.
- To design 3-D objects.
- To print 3-D objects.
- To promote interdisciplinary learning and teamwork.
- To improve the motivation and interest of the pupils.
- Improve spatial vision.
- To increase creativity.
- To plan a project from the beginning.
- Improve problem-solving skills.
- Installation and programming of electronic components.

ACTIVITY 9

- [ ] SCIENCE
- [x] TECHNOLOGY
- [ ] MATHEMATICS
- [ ] GEOGRAPHY/HISTORY
- [ ] LANGUAGES
- [ ] LITERATURE
- [ ] MUSIC
- [ ] OTHERS .....

EDUCATION LEVEL

This activity is prepared to be completed by...

- [x] 12 - 14 YEARS
- [x] 14 - 16 YEARS
- [ ] OTHERS .....

TOOLS NEEDED

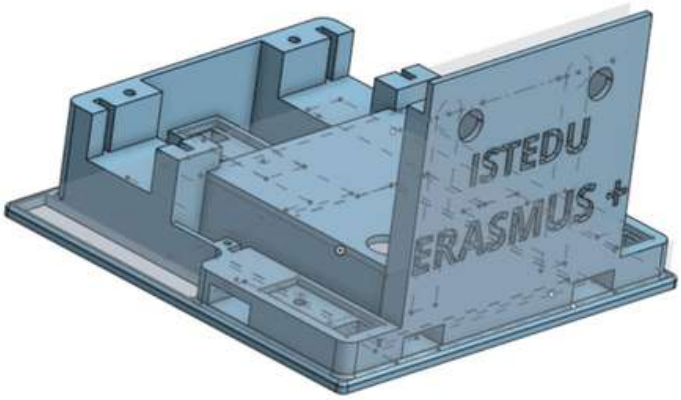
The following materials are needed

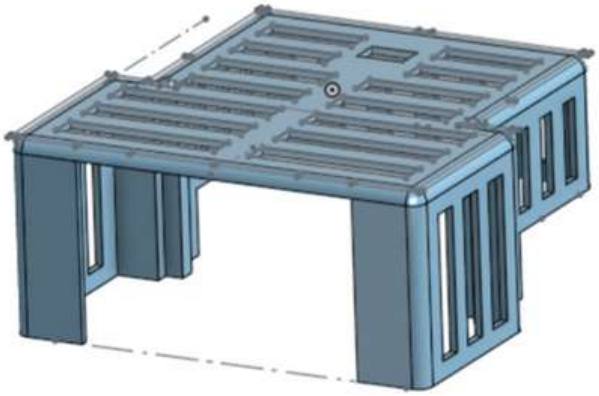
- Computer.
- Mblock version 3.11
- 3 D design software
- 3 D printer
- Arduino board
- Shield for Arduino board
- Servo 360 sg90
- Light sensor LDR
- Wheels
- Resistors
- F-F cables
- Battery holder

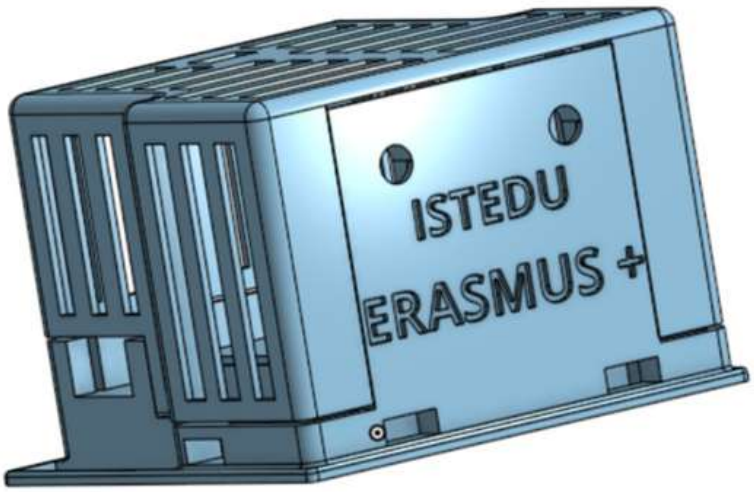
ACTIVITY 9

DEVELOP ACTIVITY

Step 1: 3D printing the parts





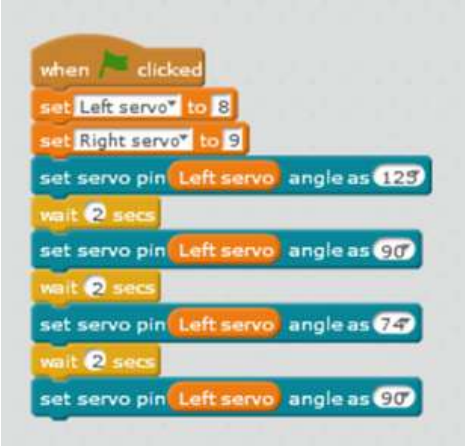


Step 2. Connect all components, following the diagram below:

- Right motor to digital pin number 9
- Left motor to digital pin number 8
- Connect the voltage dividers on the left to A0 and on the right to A1.

Step 3. Once all components are connected, check one by one that they are working properly:

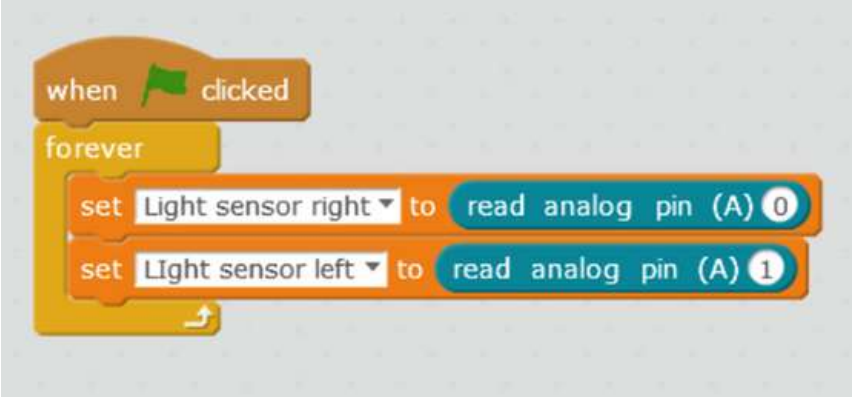
- Servos:
  - Open Mblock and upload the following program:



The left engine must rotate in one direction for 2 seconds, stop for another 2 seconds, rotate in the opposite direction and finally stop.

  - The same must be done with the other engine.

- Light Sensor LDR:
  - Open Mblock and upload the following program :



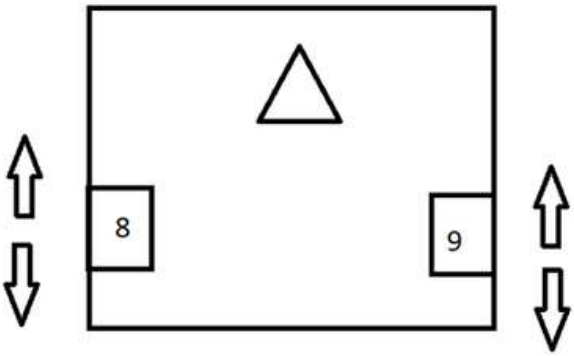
If they work correctly they should give a value between 0 and 1023, if we put our hand in front of the sensors or light them up the values should change.

Step 4. Program the robot to follow the light.

It is not a difficult task, but if you try to approach it without dividing it into different parts, it can be difficult. We will divide it into 3 parts.

1. Program movements

- Program it to walk straight. To do this, the rotation angles of the two servos must be adjusted. Using a diagram similar to the following, indicate the angle of rotation so that it moves in each direction



It is convenient to adjust the angles for different speeds.

- Program it to turn to the right, the angles of the two servos must be adjusted, it would be advisable to try different speeds.
- Program it to turn left, the angles of the two servos need to be adjusted, it would be advisable to try different speeds.

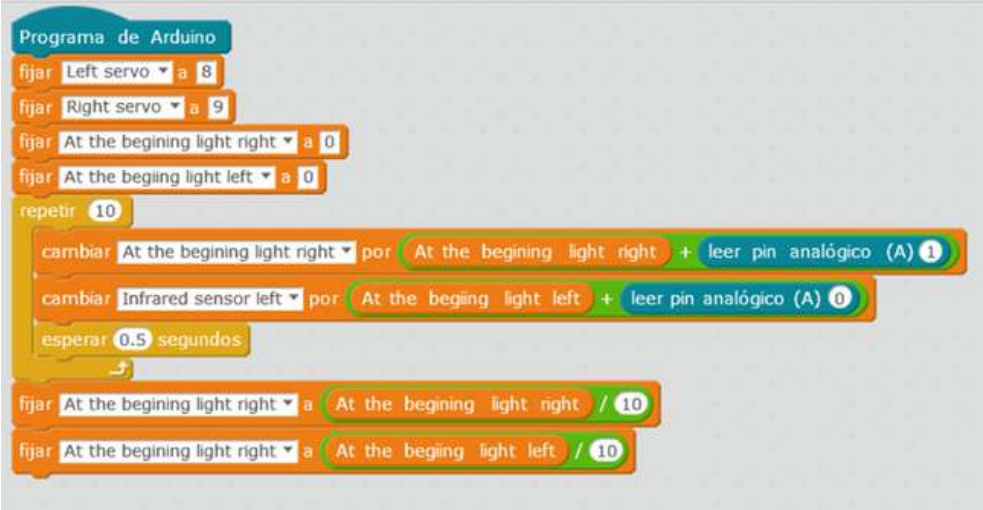
2. Follower light

Now we must program the light sensors and the robot's movements together.

The operation will be as follows:

- Al principio debe detectar la luz ambiente y guardarla en una variable.

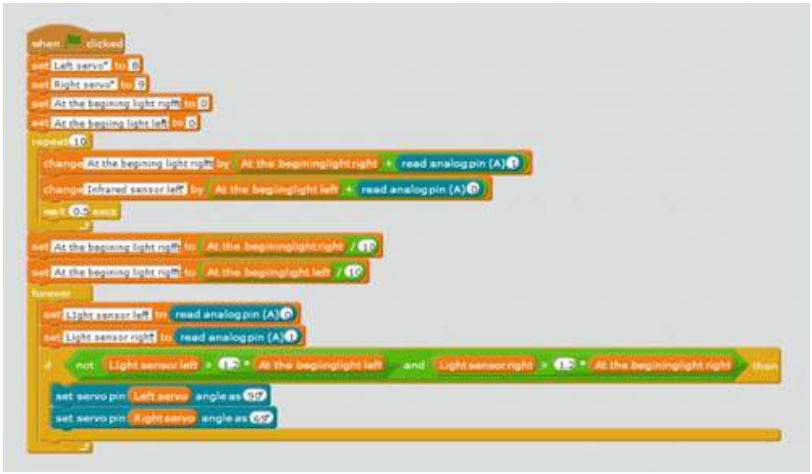
ACTIVITY 9



- It will then compare the value of the ambient light with the light received by each sensor, when the light received is greater than the ambient light, that wheel will move forward.
- There are at least three possible situations:Posteriormente comparará el valor de la luz ambiente con la luz recibida por cada sensor, cuando la luz recibida sea mayor que la luz ambiente , esa rueda caminará hacia delante.

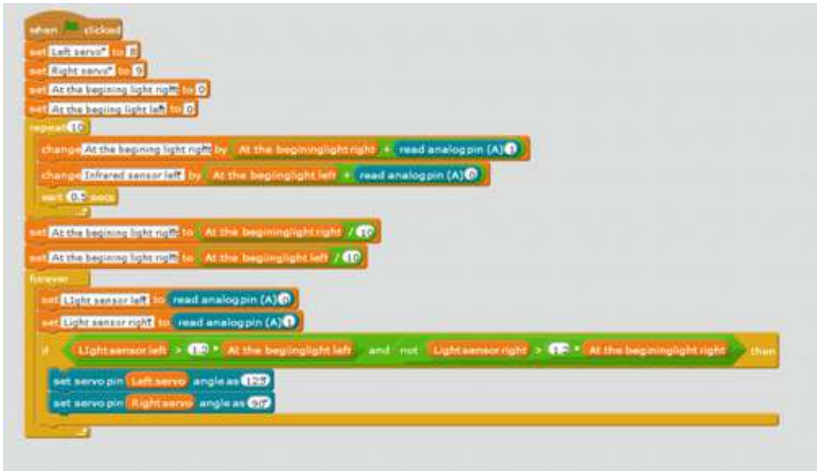
At least three situations are possible:

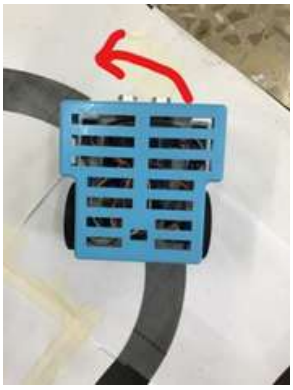
The left sensor is on the black line and the right sensor is on the white line, the robot must turn to the right..



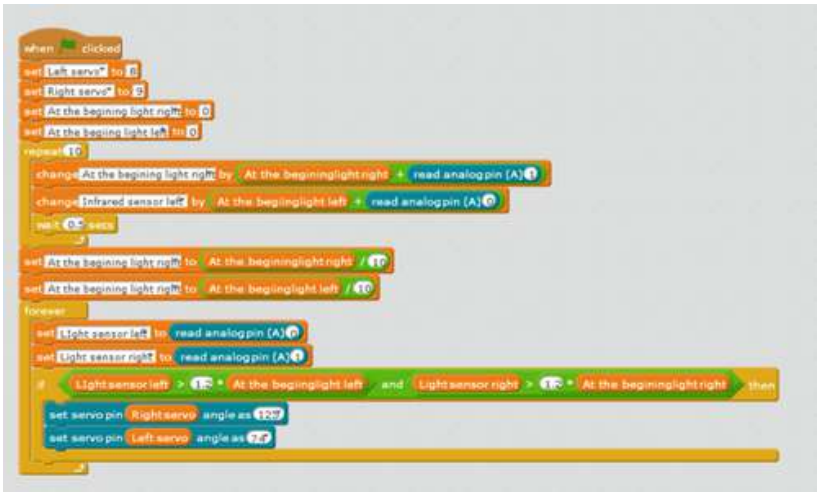
ACTIVITY 9

The two sensors detect approximately the same light, then go straight ahead.



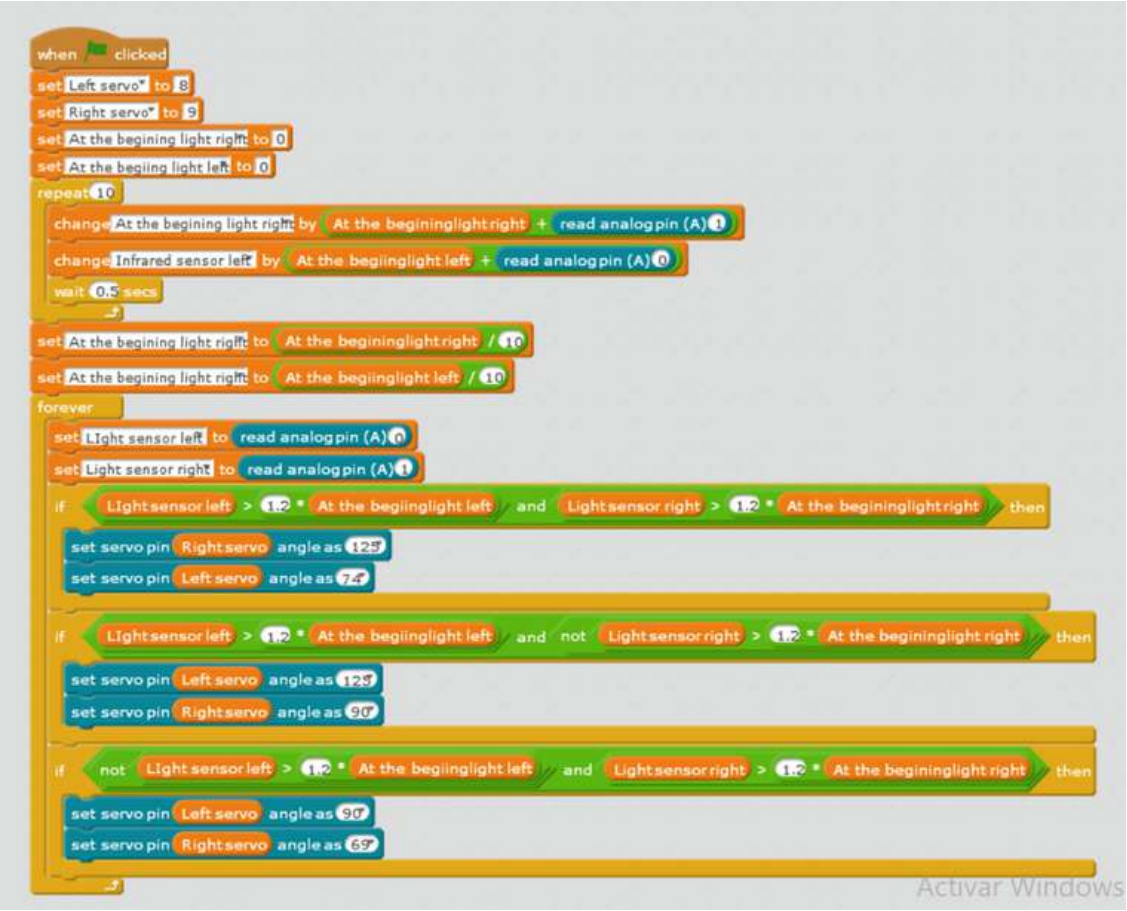


The sensor on the right side detects more light than the left side, then it must turn to the left.



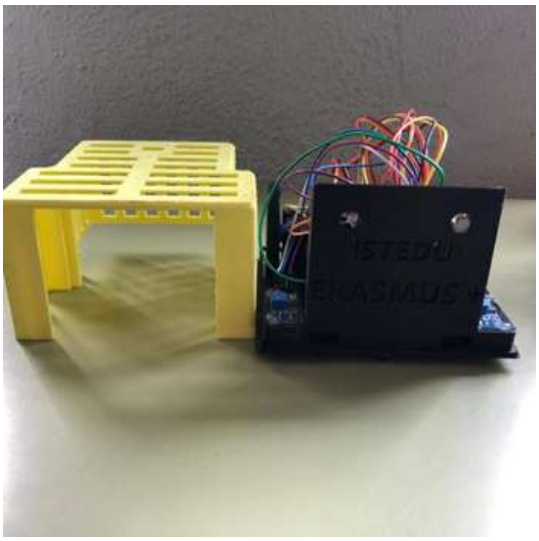
It is advisable to test each of these situations separately, in order to know if each of them has been programmed correctly.

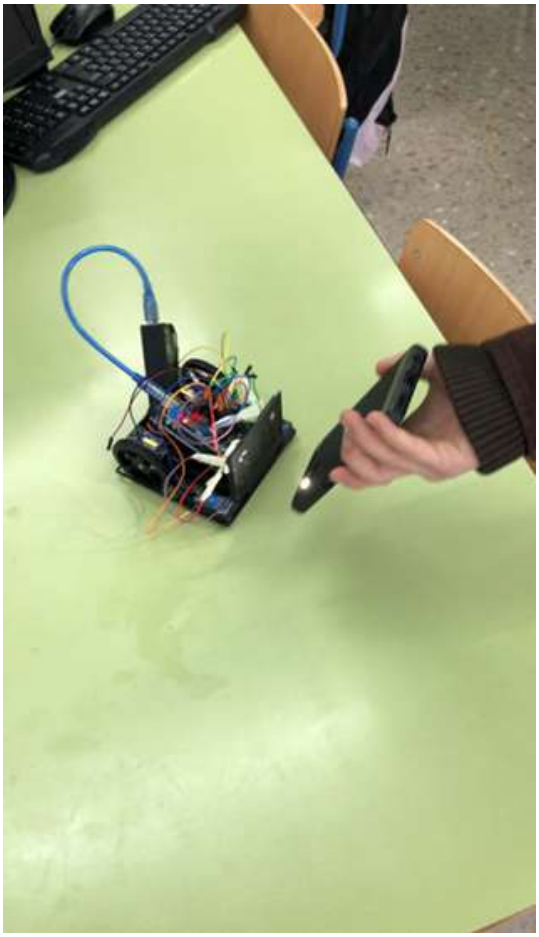
3. The final programme will be

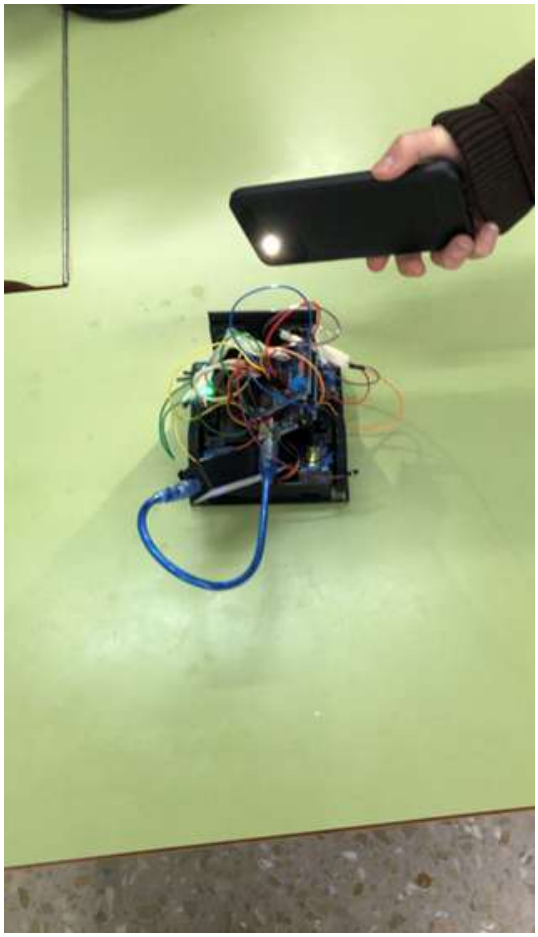


RESOURCES

ACTIVITY 9







STUDENT'S EVALUATION

It will be evaluated:

- Printing of the different parts of the robot.
- Installation of the electronic components.
- Correct operation of all the electronic components by means of the individual programming of each one.
- Programming of the different movements of the robot.
- The robot can follow a straight line, a right-hand curve or a left-hand curve.

BIBLIOGRAPHY

https://www.arduino.cc/

SCALABILITY

It is possible to create a circuit with obstacles and have the robot avoid them. On the other hand, the speed of the robot can vary depending on the light received compared to the ambient light.

ACTIVITY 10

TITLE Snow Plow

ABSTRACT

The rapid developments in technology have been at a dizzying level in recent years. While many objects and technologies mentioned in science fiction novels and movies in the 1970s and 1980s were considered as products of imagination, they are now in our lives. Robots, which were considered as servants when the term robot was first introduced, are used in many fields from industry to military today. Although it is not very common yet, education has started to be an area where robots are used. Robots in education; It is used as a teacher, teaching material, peer tutor and assistant teacher. There have been studies on the use of robots in many areas of education. Completed studies show that the use of robots in education is effective and increases motivation. With the increase in research in this field and the decrease in production costs, it will be possible to see robots in many fields of education in the near future.
In this project, we designed Mbot as a snow plow. We looked at the snowplows, which are of vital importance in winter days, from the eyes of our students.

AUTHOR/S

Sultantepe Prof. Dr. Cemil Taşçıoğlu Ortaokulu

DATE 26/01/2022

VERSION 1

ACTIVITY 10

DIDACTIC OBJECTIVES

Beginner student:
-What is mbot, how does it work
-Recognizes Mblock program
-Recognizes Tinkercad software.

Knows what a 3D printer is.
- Hand-motor muscles develop in the use of materials

Intermediate student;

How to write code, program interface, language preferences, connection in Mblock program.
-Knows how to output in .stl format on Tinkercad platform.
-Knows how to use slice program for 3D printer.
- Hand-motor muscles develop in the use of materials

- ☐ SCIENCE
- ☑ TECHNOLOGY
- ☐ MATHEMATICS
- ☐ GEOGRAPHY/HISTORY
- ☐ LANGUAGES
- ☐ LITERATURE
- ☐ MUSIC
- ☐ OTHERS .....

EDUCATION LEVEL

This activity is prepared to be completed by...

- ☑ 12 - 14 YEARS
- ☐ 14 - 16 YEARS
- ☐ OTHERS .....

ACTIVITY 10

TOOLS NEEDED

Mbot(Bluetooth)
Mblock Program
Mbot Remote
3d printer
Cardboard
Tape, Glue, Scissors, Cotton, Hot Silicone
Model house, model trees
Computer (For Modeling with Tinkercad)

DEVELOP ACTIVITY

The project setup is given below step by step.

1. The cardboards considered for the floor are cut in appropriate sizes. It is preferable to use different color cardboards to indicate that the ground and the road are separated from each other.

2. Model houses and trees are glued on cardboard.

3. Cottons representing snowfall are distributed on the cardboard floor in such a way that it also closes the roads.

4. According to the dimensions of the front part of the Mbot, the snow plow apparatus is designed in Tinkercad software.

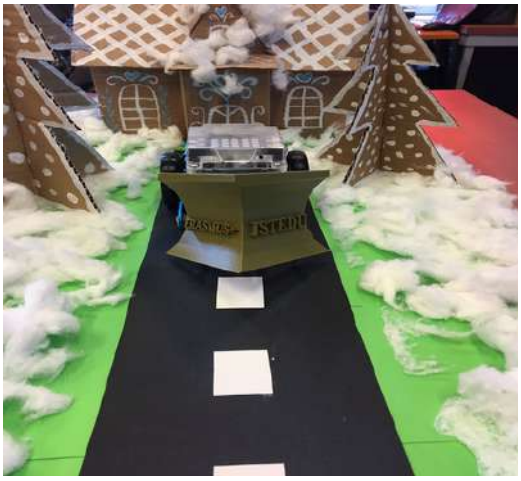
5. The design is printed using a 3D printer, slicing program and PLA filament. (Hand contact should be avoided while taking 3D printing. The melted filament will be 200 0C to the environment.) The printed print is mounted on the Mbot.

6. In the Mblock program, a program is written so that Mbot can be controlled with a remote.

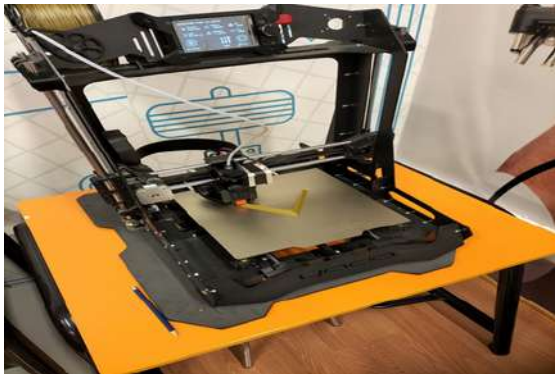
7. Mbot is allowed to clean the road from the snow on the cardboard.

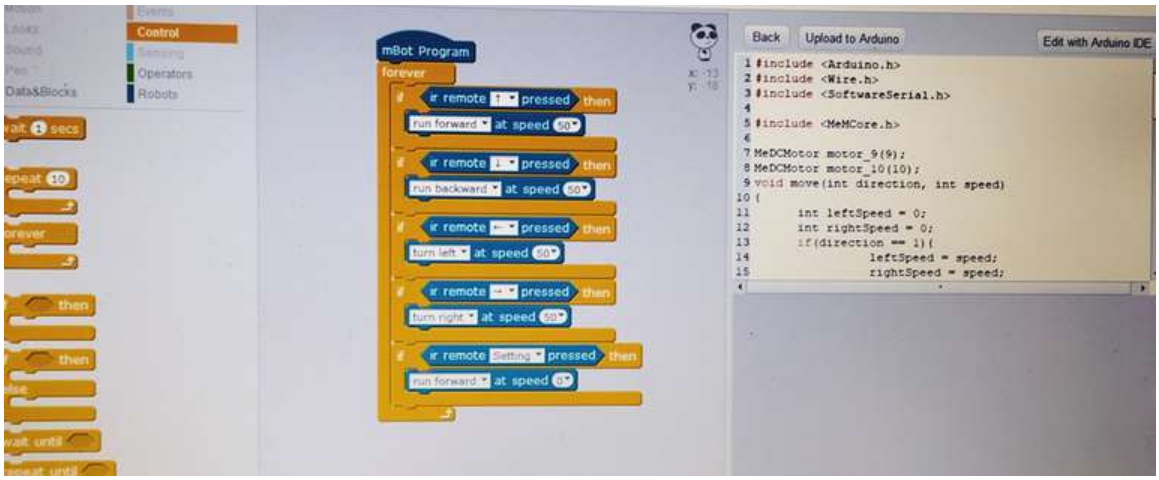
ACTIVITY 10

RESOURCES



ACTIVITY 10





STUDENT'S EVALUATION

Areas in which our students will evaluate themselves:

1. Can I use Mblock software for Mbot?
2. Can I model with Tinkercad?
3. Can I use the project materials correctly and create the appropriate ground?

can be listed as.

BIBLIOGRAPHY

https://ide.mblock.cc/
https://www.tinkercad.com/

SCALABILITY

This project can be considered as beginner-intermediate level for fifth and sixth grade students. According to the theme, this project;

Evade when Mbot sees an Obstacle using the distance sensor It can be developed to prevent it from going out of the road by using a line tracking sensor, it can be made autonomous. In addition, a trailer can be attached to the rear of the snowplow, a servo motor connection can be made to the 3D printing part, and the snow collected can be transferred to the trailer at regular intervals. This also removes roadside snowhills.