Module 1

ROBOTICS THROUGH SIGN LANGUAGE: ENSURING ACCESS AND ENGAGEMENT OF STUDENTS WITH DISABILITIES (DEAF OR WITH HEARING IMPAIRMENT) TO THE DIGITAL WORLD OF CODING AND ROBOTICS – ROBOTICS4DEAF

Project no: 2019-1-PL01-KA201-065123

Erasmus+

This project has been funded with support from the European Commission.
This communication reflects the views only of the author,
and the Commission cannot be held responsible for any use
which may be made of the information contained therein.

Version: 01.2020

Disclaimer

This project has been funded by the Erasmus+ Programme of the European Union.

Πίνακας Περιεχομένων

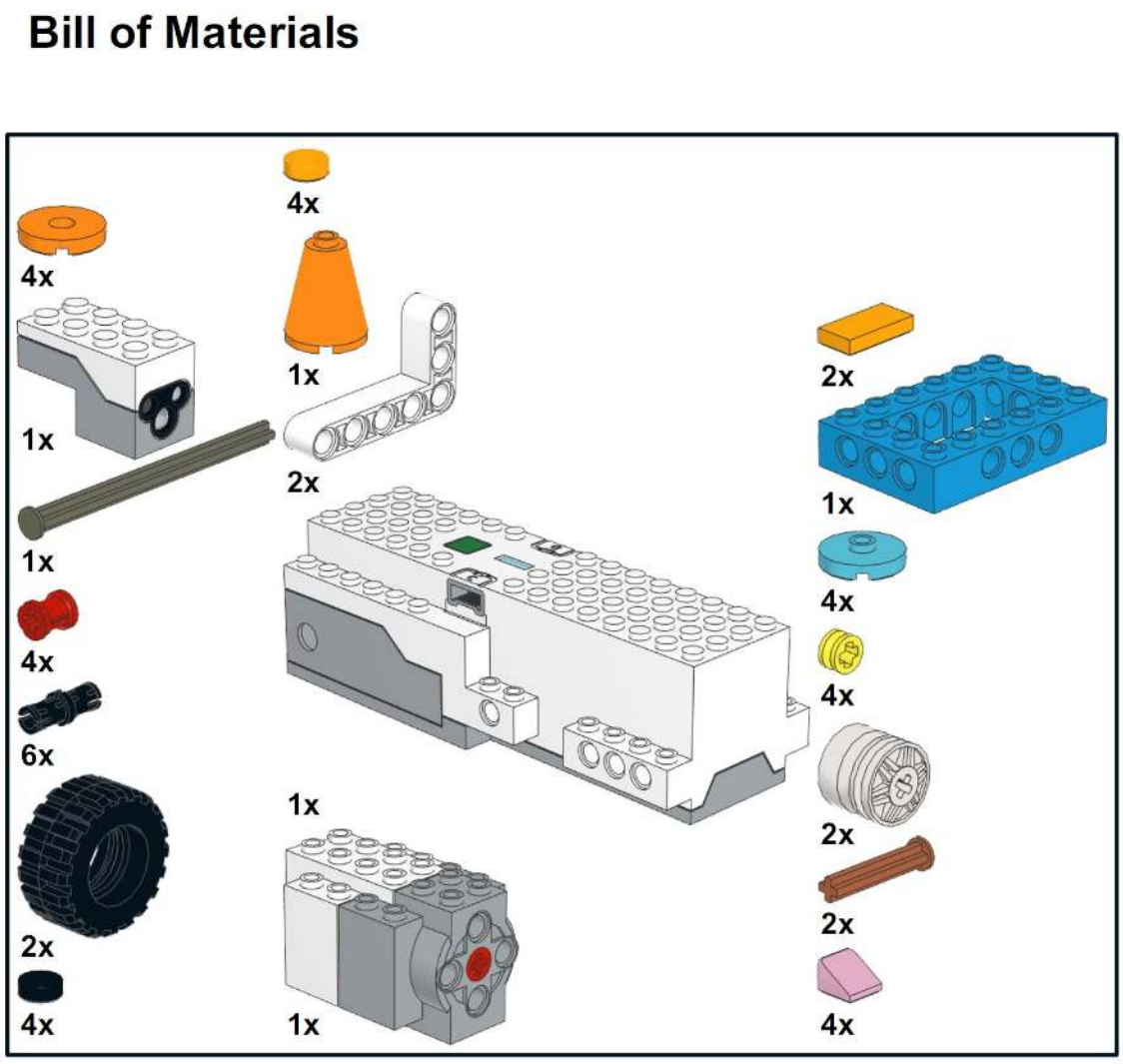
Erasmus+

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

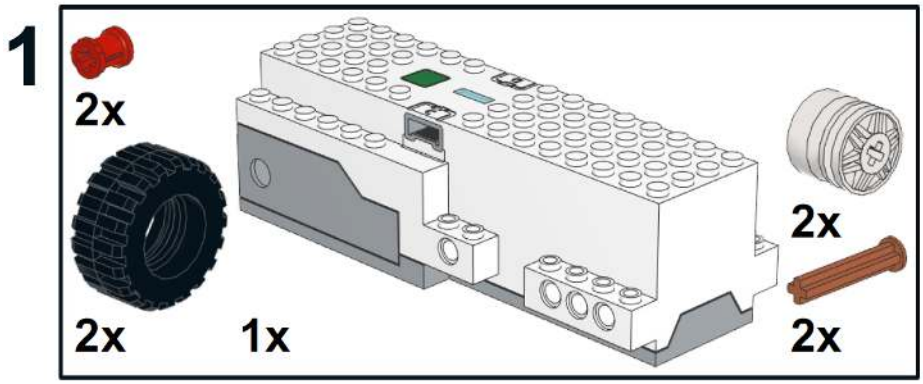
1. SMARC – SMArt Robotic and Coding

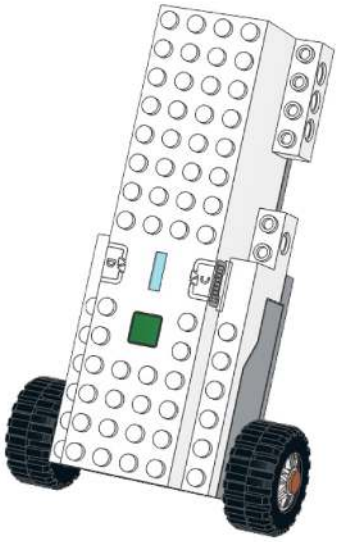
The aim of this section is to introduce students to the basic movements of robots. Students will build the robot SMARC and learn to program it to perform basic movements. They will also learn how to use loops to program it to make repetitive movements.

I. Build SMARC



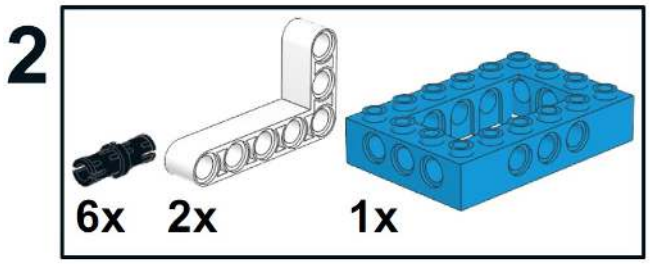
Erasmus+
Co-funded by the Erasmus+ Programme of the European Union

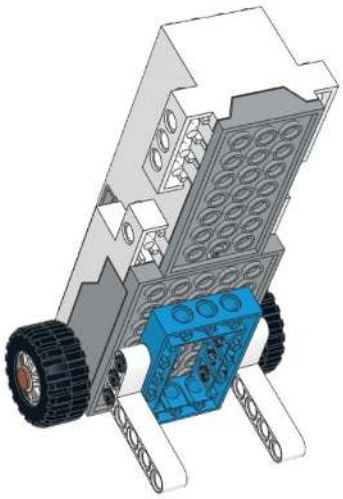




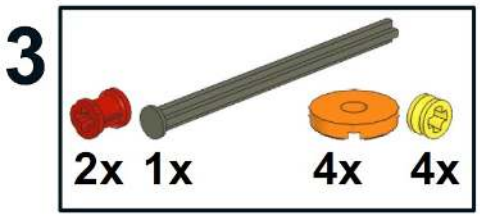
Erasmus+

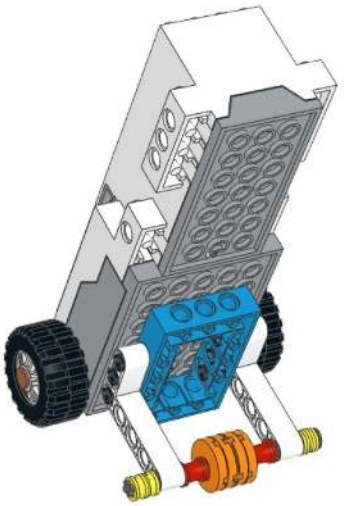
Co-funded by the
Erasmus+ Programme
of the European Union



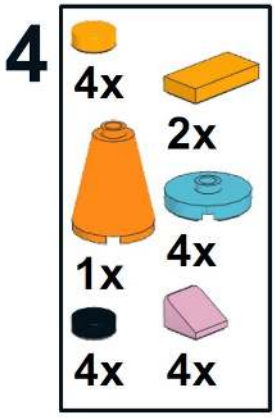


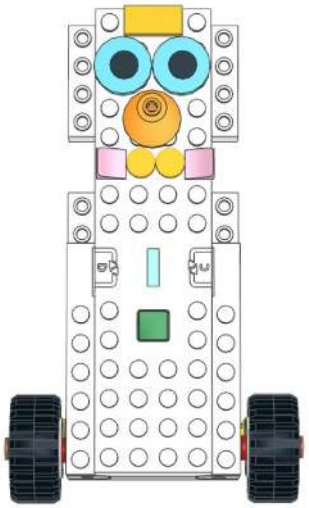
Erasmus+
Co-funded by the Erasmus+ Programme of the European Union



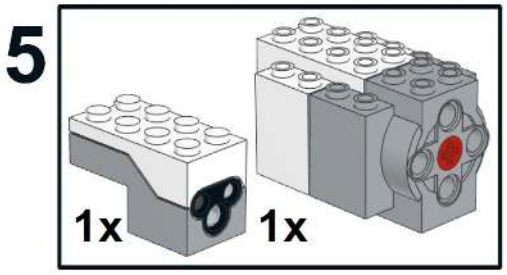


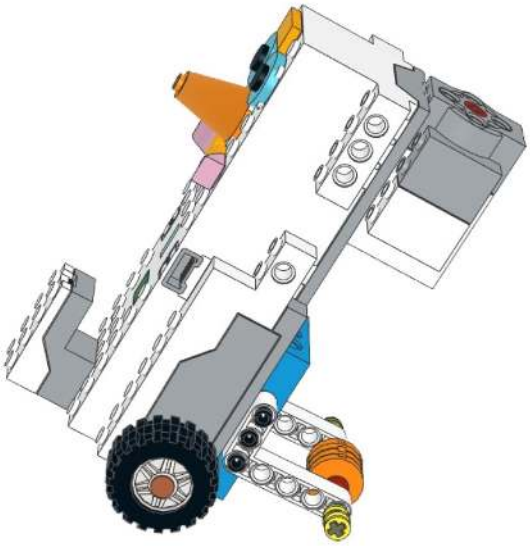
Erasmus+
Co-funded by the Erasmus+ Programme of the European Union





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





II. Learn how to make SMARC move.

The photo below shows the programming environment of the application.

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

It also shows the area where the programming blocks are located. There you will find everything you need to make your own programs

Spend some time to browse the application.

https://www.youtube.com/watch?v=MwRDmAlIGXQ

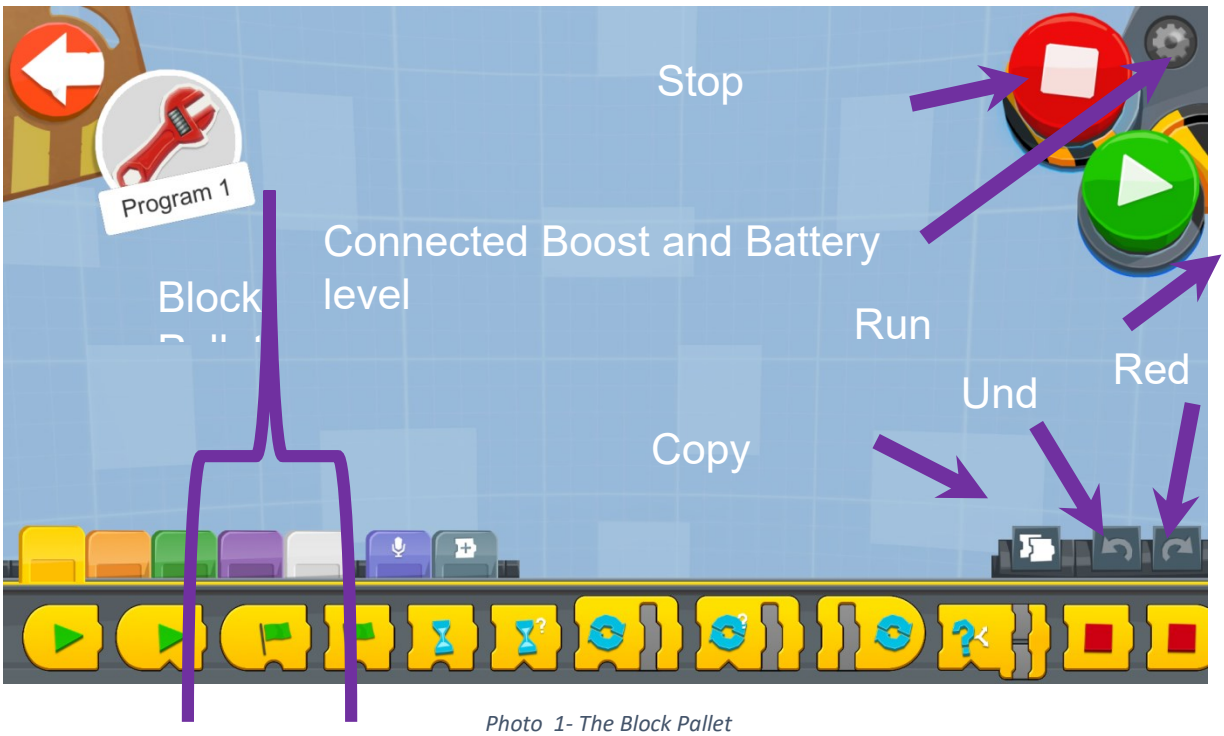


Photo 1- The Block Pallet

A. Grouping the blocks.

In this section you will learn how blocks are grouped according to their color. Knowing the general categories makes it easier to find the right block depending on what we want to do. In the following sections we will examine each category in detail.

https://www.youtube.com/watch?v=mByDBP3O4mk

- With **Yellow** blocks you can control the flow of the program

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

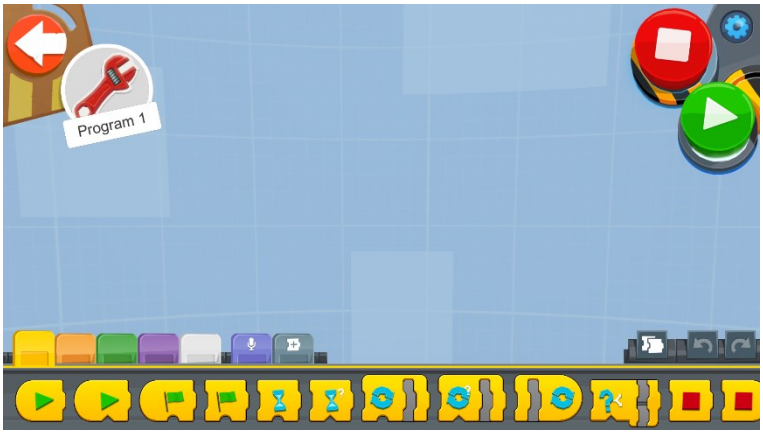
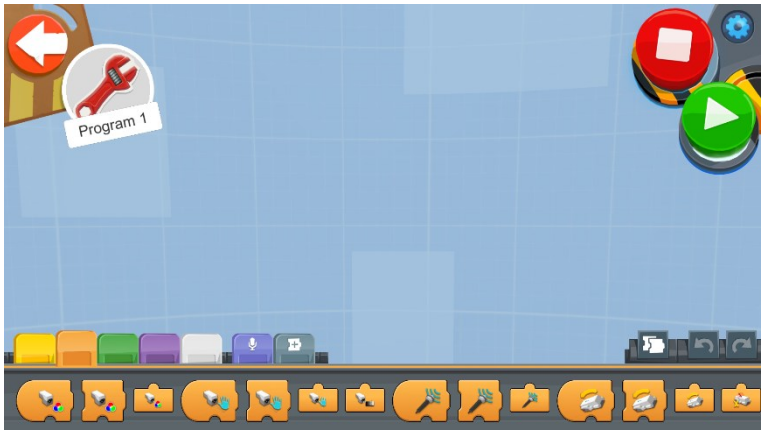


Photo 2-Yellow Blocks

You can use them to start a program, stop a program, pause a program or even loop a group of commands

- The **Orange** blocks work with the Color and Distance Sensor, microphone and the Move Hub's Tilt Sensor



When one of the above sensors is activated, it will also activate those blocks.

Erasmus+

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

- **Purple** blocks are used for communication of the robot with the outside world.

They can play sounds through the user's device speakers. They can also change the color of the lights on the Move Hub & Color and Distance Sensor.

- **White** blocks let you do some complex programming using 'variables' and 'constants'

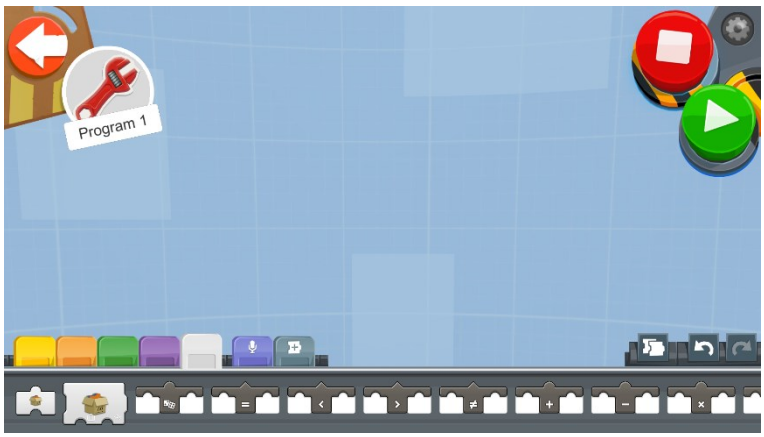


Photo -White Blocks

They can also be used when performing various mathematical calculations, creating logic expressions and generating random numbers

Erasmus+

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

- **Light Purple** blocks let you record a sound for the robot to repeat.

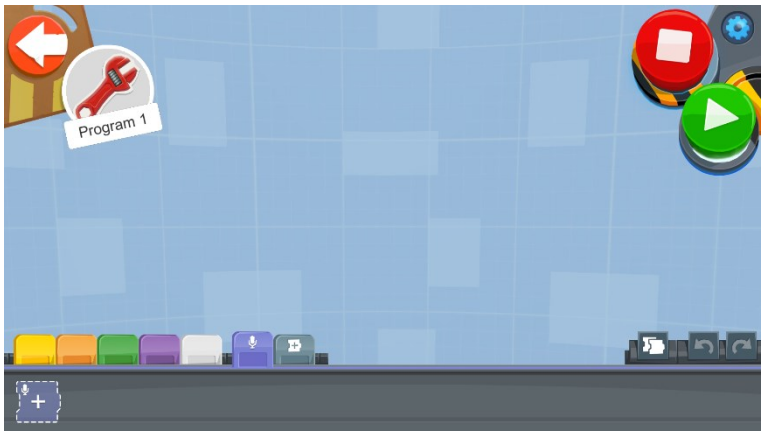


Photo -Light Purple Blocks

- **Group Actions** block let you group as one block button several actions

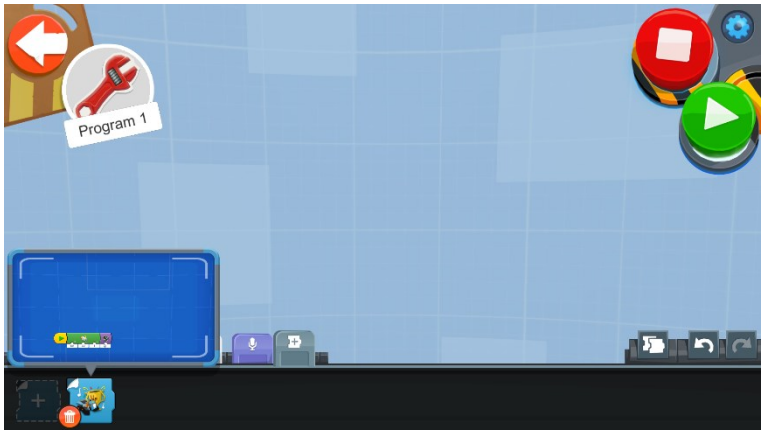


Photo - Group Action Blocks

This is the equivalent of creating sub-programs

Erasmus+

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

B. How to make SMARC move

In order to begin coding SMARC the first thing we need is the Start Block located in the Yellow Pallet

https://www.youtube.com/watch?v=Sdv190liSJc

There is a number of blocks, which control the movement of the Move Hub. Those blocks can be found in the Green Section of the Pallet. Here we present the most common ones.

https://www.youtube.com/watch?v=UjusigZPIr0

Move Base for Duration - Motor one speed (from -100 to 100) and Motor two speed (from -100 to 100) for duration (in seconds)

Move Base Tank for Distance - Motor one speed (from -100 to 100) and Motor two speed (from -100 to 100) for distance (in degrees)

Move Tank and Steer for Duration - Both motors speed (from -100 to 100) and steering direction (from -100 to 100) for duration (in seconds)

Move Tank and Steer for Distance - Both motors speed (from -100 to 100) and steering direction (from -100 to 100) for distance (in degrees)

Move Tank - Motor one speed (from -100 to 100) and Motor two speed (from -100 to 100)

Erasmus+

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

Move Tank and Steer for Duration - Both motors speed (from -100 to 100) and steering direction (from -100 to 100)

C. Sample Programs

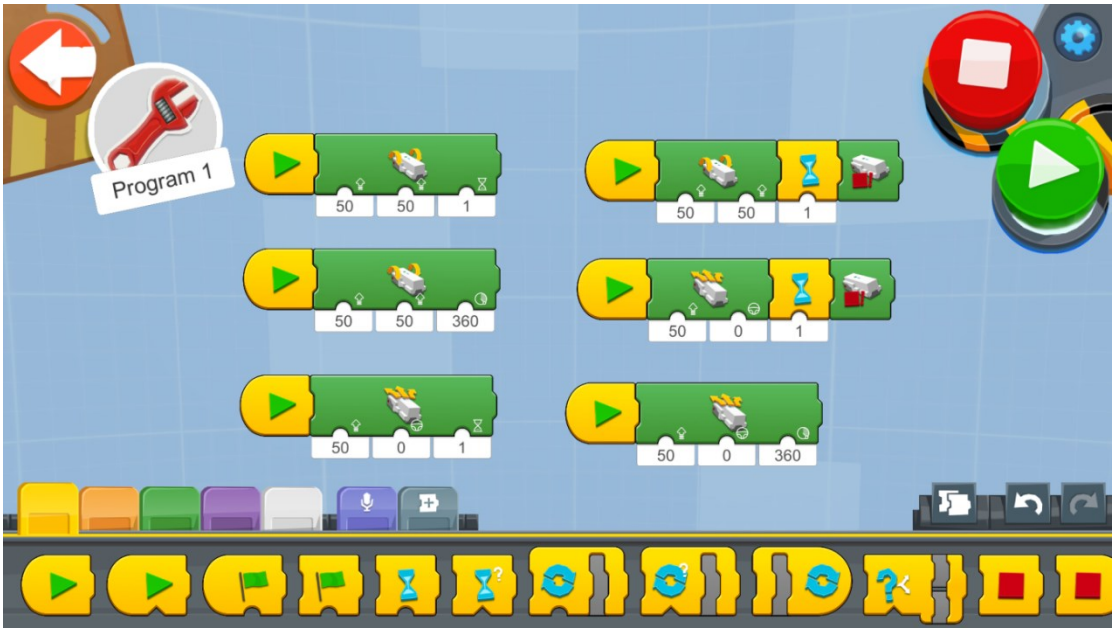


Photo 8 – Make SMARC move

Try using the different blocks for moving like the example program above.

Video with the following text and Smarc executing the programs.

Create a new project on the Creative Canva and run each block of code and notice the differences and similarities.

Erasmus+

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

Note that the moving blocks which do not set the time interval or the amount of degrees (rotations) of the movement will continue to turn the motors intefinetly unless a wait block is added which sets the time interval for the motors to work followed by the stop motors block.

Learn how to make Smarc move. - https://youtu.be/-Jor9F0y4fA

Note: if the learner has completed the first activity with the simple car oe the first of Vernies series, can use a more simplified version of the movement blocks.

Photo 9-The Main Lobby Scene of Lego Boost App

Step 1: Go to Simple Car Icon on the Main Lobby Scene and click it

Erasmus+

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

Photo 10-The Simple Car Activity 1

Step 2: Click on the First Activity

HINT for the teacher: You can unlock the programs by pressing the activity you want to unlock for five seconds.

Erasmus+

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

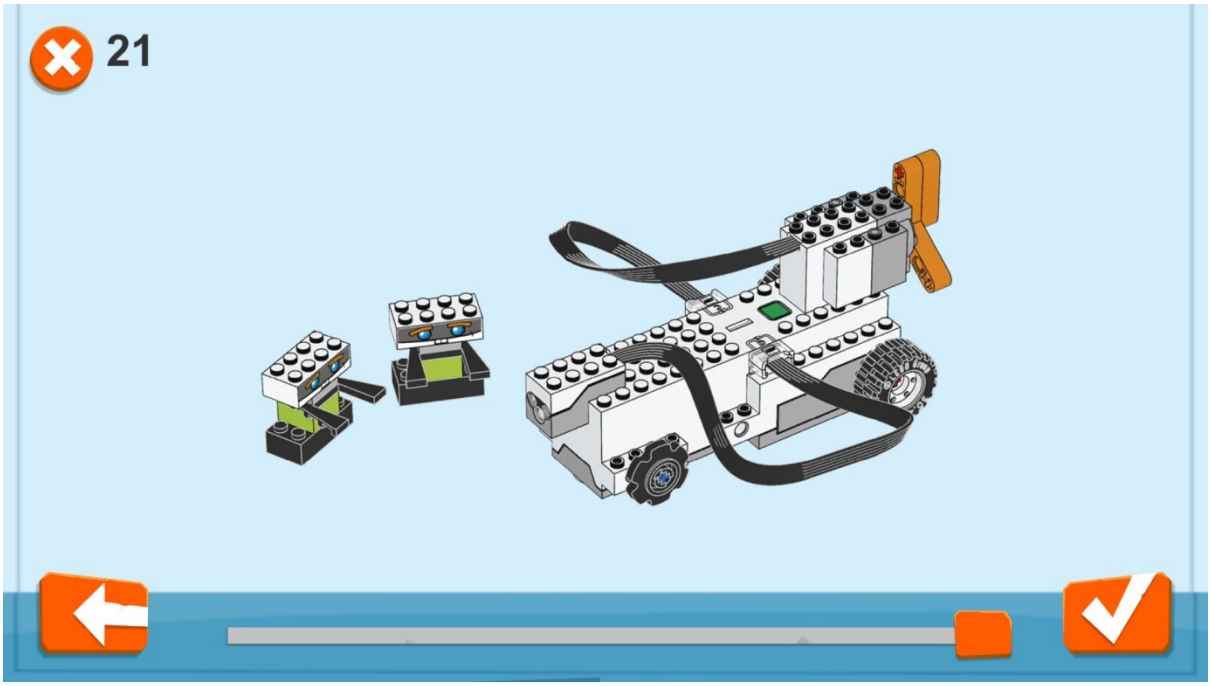


Photo 11-Building the Simple car

Step 3: If it is the first time opening this specific activity the step by step guide for building the car showed. You can always just scroll to the last page and click on the tick icon.

Photo 12-The Basic blocks for simple movements

Step 4: Use the simple move blocks to move Smarc. How far does he move when one straight arrow block is used? Does it move in the expected direction?

Erasmus+

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

Note for the teacher: The driving motors for Smarc are in the front of the robot, on the Simple Drive Car they are situated in the back of the robot.

III. The loop coding concept

Loops allow Smarc to easily repeat blocks of code without having to put blocks of code again and again. There are different types of loops in coding.

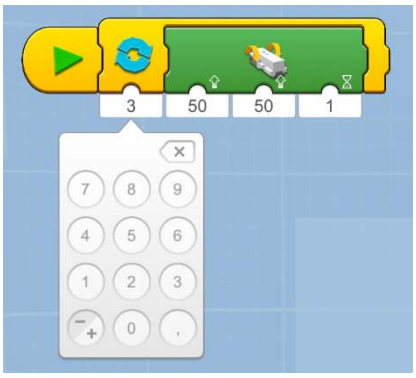
"For Loops" let programmers repeat code a specific number of times.

"While Loops" require a condition to be true in order to repeat code.

"Forever Loops" repeat code forever and should be avoided when not necessary because they can consume all the resources of our computer and run out the batteries of our robot.

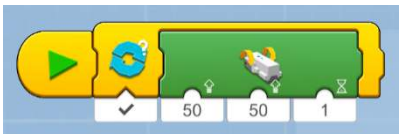
https://www.youtube.com/watch?v=kAUcAlgB-XA

- "For Loops".
  The Loop For Count: Loops the enclosed blocks of code for a specific number of times which can be set by the user

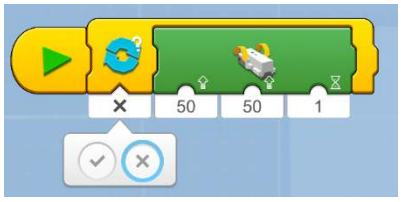


Video

- "While Loops"
  The Loop While True: Loops the enclosed blocks of code while contition is true.

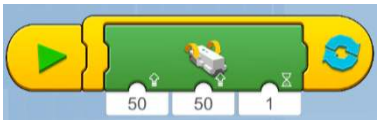


  The user can set if the loop will be executed if the condition is true or false.



Video

- "Forever Loops"
  The Loop Forever: Loops the enclosed blocks of code



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

loop coding concept - https://youtu.be/hPyylPF2QZM

Sample Programs of Loop Blocks

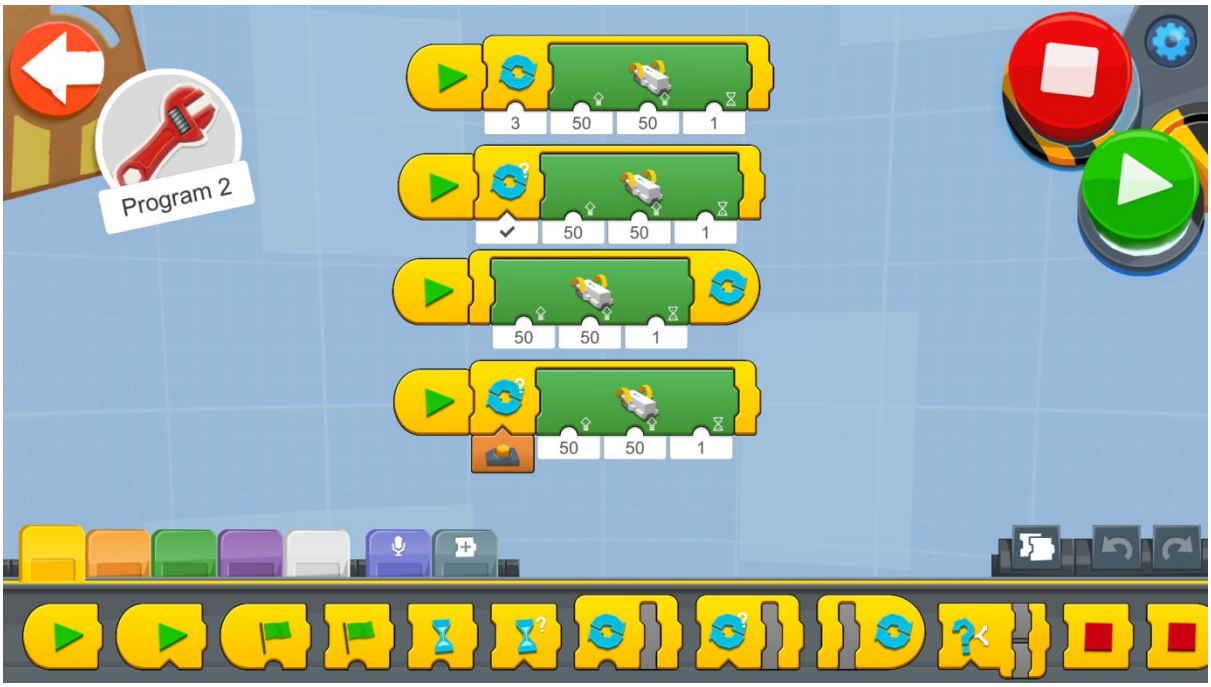


Photo 13-Program - Loop blocks

Try using the different blocks for Looping commands like the example program above.

Create a new project on the Creative Canva and run each block of code and notice the differences and similarities.

Notice that the last block of code is not working properly. Try and make it work so when a button is pressed the Loop is activated (the loop condition becomes true).

loop sample programs - https://youtu.be/PcGk9zD2M5k

Erasmus+

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

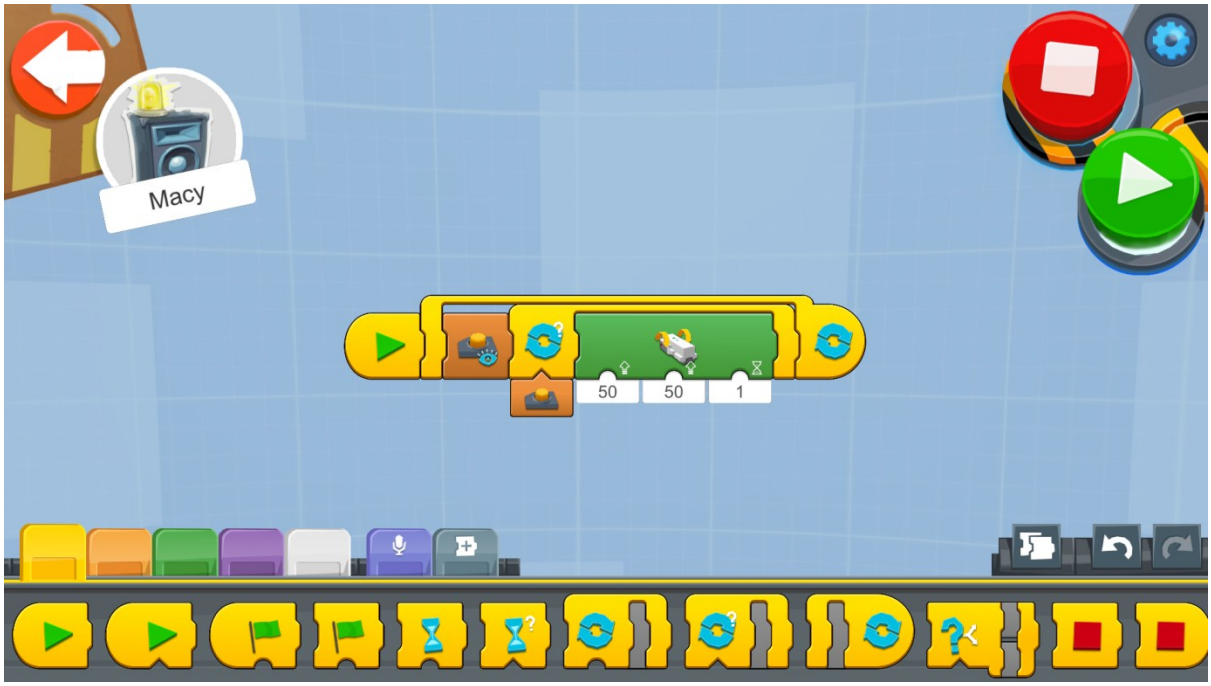


Photo 14-Program - Nested Loop

The solution is to insert another Loop Forever loop and add the button appearance block outside the Loop While True block. This is called a nested loop (a loop inside a loop).

2. Using Sensors with Smarc

IV. Sensors

In this section, we will see how we can incorporate a sensor to Smarc and how we can utilise this sensor in order to detect obstacles, various colors, the intensity of light as well as the distance from an object.

A. The Lego Boost Color and Distance Sensor

A sensor is a device, which detects events or changes in the surrounding environment and sends the information to a computing device in order for it to process it and take actions. There are many examples of sensor usage in our everyday life. From a motion sensor, which detects movement and turns on the lights or opens a sliding door to temperature sensors, which detect the temperature and switch on heaters and air conditions. Sensors are also used in most types of remote-controlled devices such as Televisions and Air Conditions.

https://www.youtube.com/watch?v=Wc6kaqE3wow

Erasmus+

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

The LEGO BOOST Color and Distance sensor as shown in Figure is used for:

- Sensing distance - How far an object-obstacle is situated from the sensor.
- Sensing color - it can detect specific colors (Black, Blue, Green, Yellow, Red, White) as well as the absence of a color.
- Detecting movement - it can be used as a motion detector
- Detecting light - It can detect the light level reflected on the sensor measuring from 0 - darkest to 10 - lightest.

Additionally, the Color and Distance sensor can emit different colors (Red, Green, Blue, and a combination of the three).

B. Detecting Objects

The Detecting Objects Blocks

Trigger on Distance Block - Triggers when the distance determined by the sensor is less than the distance indicated by the value below it. When it is triggered, it executes the following code sequence. Values from 0 to 10 can be taken.

Sensor Distance Reporter - Displays the current distance determined by the sensor in real time. It must be connected to the bottom of the other blocks in order to be included in the software. Values from 0 to 10 can be taken.

Erasmus+

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

Wait for Distance - Waits for the distance determined by the sensor to be less than the distance indicated by the value below it. If the object is not closer than the stated value, the program will remain paused and the program will begin to follow the following sequence of instructions until the condition is met. Values from 0 to 10 can be taken.

Sample Programs of Detecting Obstacles

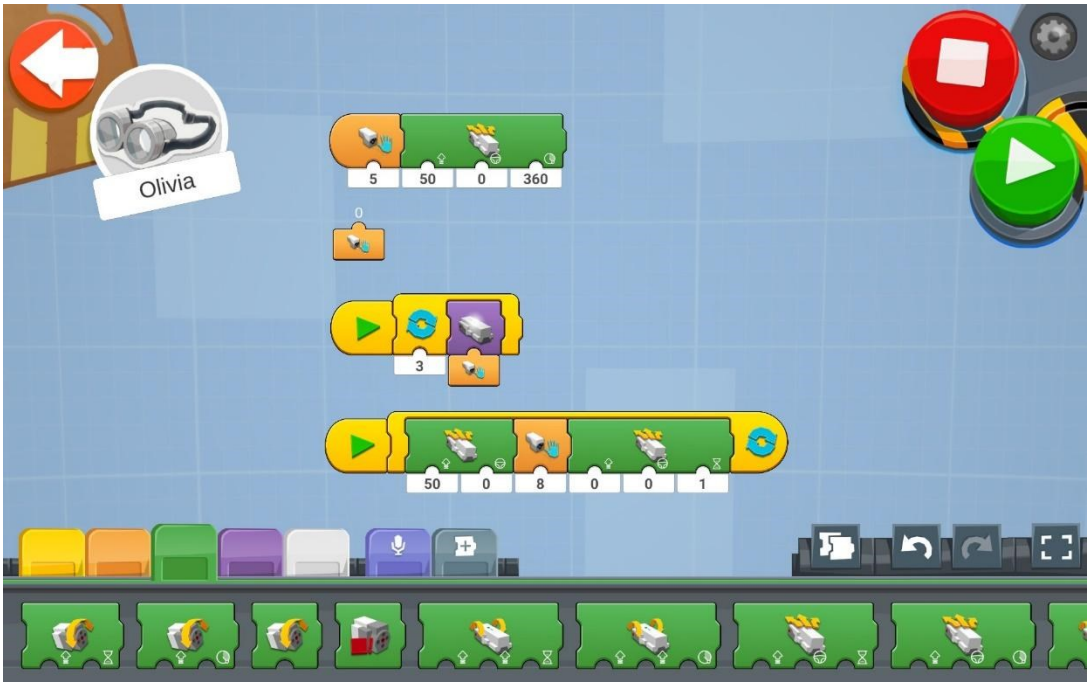


Photo 15-Detecting Obstacles

1. The first program activates when an object is less than distance 5 in front of Smarc. Smarc will then move 360 degrees forward at speed 50.
2. The second block just indicates in real time the distance from an object in front of the sensor.
3. The third program takes the distance and uses it as an input for selecting and showing colors on the movehub lamp. Notice that when the distance from an object changes the color also changes.

In the fourth program Smarc starts moving at speed 5 forward and when it detects an object in distance less than 8 it stops for a second, then tries to move again.

detecting obstacles - https://youtu.be/CAXnOmMiUDQ

Erasmus+

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

C. Detecting Colors

The LEGO BOOST sensor can detect six colors (Black, Blue, Green, Yellow, Red, White, and the absence of color - no color, which means no object is being detected by the sensor).

https://www.youtube.com/watch?v=f2nJZYSGHdo

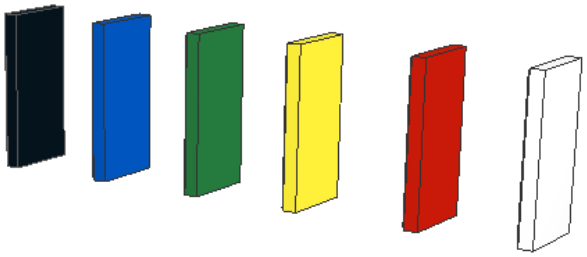


Photo 16-The Colors, which the Sensor can see: Black, Blue, Green, Yellow, Red, White

Notes:

If an object is not in the proposed range of the sensor, the color might not be detected correctly. For example, a green object might be detected as blue.

Also, if the color of the object is not one of the six detectable colors the application will present it as the closest color possible. For example, orange will be presented as red.

https://www.youtube.com/watch?v=_8Q9QR87nI0

The Detecting Colors Blocks:

Trigger on Color - Triggers when the color measured by the sensor is equal to the color shown by the value below it. When activated, the following code sequence is executed. Seven values can be used: No color, Black, Blue , Green , Yellow, Red and White.

Wait for Color - Waits for the color determined by the sensor to be equal to the color shown by the value below it. If the color detected is not equal to the value specified, the program will remain paused and the following sequence of instructions will begin until the condition is met. Seven values can be used: No color, Black, Blue , Green , Yellow, Red and White.

Erasmus+

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

Sensor Color Reporter - Shows the current color measured by the sensor in real time. In order to be used in a program, it must be located on the bottom of the other bricks. You will see seven values: No color, Black, Blue , Green, Purple, Red and White.

Sample Programs of Detecting Colors

We will first need to install the interactive motor before we see the detecting colors programs.

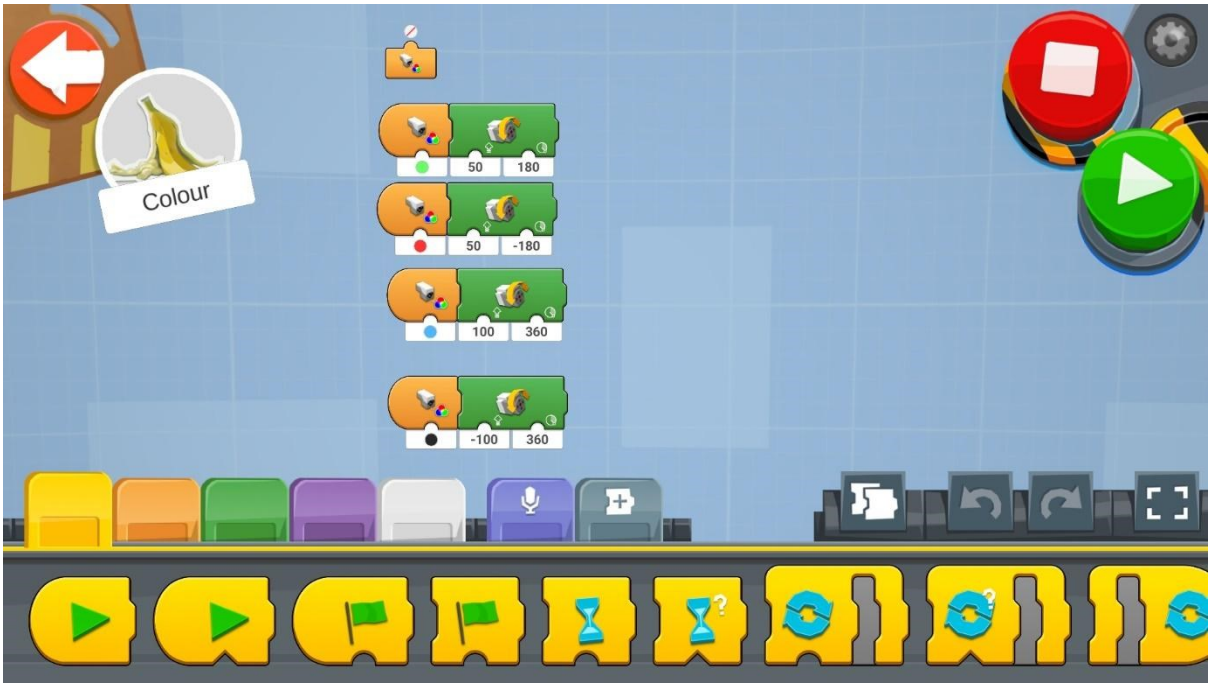


Photo 17- Program - Detecting Colors

1. The first block just indicates in real time the color of the object in front of the sensor.
2. The second program activates when an object of green color is in front of Smarc's sensor. Smarc will then move the propeler 180 degrees to the right at speed 50.
3. The third program activates when an object of red color is in front of Smarc's sensor. Smarc will then move the back propeler 180 degrees to the left at speed 50.
4. The fourth program activates when an object of blue color is in front of Smarc's sensor. Smarc will then move the back propeler 360 degrees to the right at speed 100.
5. The fifth program activates when an object of green black is in front of Smarc's sensor. Smarc will then move the back propeler 360 degrees to the left at speed 100.

detecting colors - https://youtu.be/onDfjAmNHio

Erasmus+

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

D. The If/Else Blocks:

The If/Else statement allows Smarc to easily make decisions based on the inputs from the sensor. If a condition is met (meaning it is TRUE), Smarc will execute a specific block of code. Else if the condition is not met (meaning it is FALSE), Smarc will execute another block of code.

https://www.youtube.com/watch?v=NB4jivbIhqc

If/Else - If a condition is True then execute the top sequence, else if it is False, execute the bottom sequence.

Equal Operator - Returns True when an input from a sensor (color/distance/ambient light) is equal to a value.

Less Than Operator - Returns True when an input from a sensor (color/distance/ambient light) is less than a value.

Greater Than Operator - Returns True when an input from a sensor (color/distance/ambient light) is greater than a value.

Not Equal Operator - Returns True when an input from a sensor (color/distance/ambient light) is not equal to a value.

Measuring the Amount of Light Reflection: The Measuring Light Reflection Blocks:

Sensor Light Level Reporter – Displays in real time the current ambient light level measured by the sensor. In order to be used in a program it needs to be attached to the bottom of other blocks. It can indicate values from one (1) to ten (10). One being the darkest and ten being the brightest.

Erasmus+

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

V. Detecting Sound with Smarc

A. React on Sound

A sound-activated robot

In this section we will see how we can control Smarc with sounds such as clapping our hands, and by using different sound intensities, with the use of the Sound Sensor blocks.

In the case of Smarc, the sound sensor is actually a built in microphone, which is not located in the Move Hub. Instead, it relies on the microphone, which is located inside of the device, which Smarc is connected to such as the tablet, laptop or a smartphone.

The Lego Boost App does recognize how loud a sound is. Hence, we can program it to react differently depending on the sound intensity it receives.

https://www.youtube.com/watch?v=ybygsyWfOh4&t=1s

B. The Sound Sensor Blocks:

Trigger on Sound Level - Triggers when the sound level measured by the sensor is higher than the sound level shown by the value below it. When it is triggered, it executes the code sequence that follows. It will take 11 values from 0 to 10.

Wait for Sound Level - Waits for the sound intensity measured by the sensor to be higher than the sound level shown by the value below. If the sound level registered is not greater than the value specified, the program remains paused and the following series of instructions begins until the criterion is met. You will take eleven values from 0 to 10.

Sound Level Reporter - Displays the latest sound frequency calculated by a sensor in real time. To be used in a program, it must be connected to the bottom of the other bricks. It will represent values from 0 to 10, including one decimal number, e.g. 7,8.

Sample Programs for Detecting Sound

Erasmus+

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

Program 1

Photo 24-Program 1- Detect Sound

Create a new project on the Creative Canvas and run the block of code:

1. The code will execute when the Trigger on Sound Level block triggers. This will happen when a noise of level greater than 5 is detected.
2. If the sound is detected, Smarc moves forward using the Drivebase Move Steering block at speed 50.
3. The Wait for Time block is set for 1 second. The reason for this block is for Smarc not to be confused with the same sound used for activating the program to move forward and to trigger the rest of the code.
4. The Wait for Sound Level block waits for a noise of level greater than 5 in order to trigger the execution of the rest of the code.
5. Finally, when the Wait for Sound Level block is triggered, the Drivebase Stop activates and stops Smarc from moving.

Detecting sound 1 - https://youtu.be/3xrI2SGrlh8

Erasmus+

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

Program 2

Photo 25- Program 2 - Detect Sound

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

Create a new project on the Creative Canvas and run the block of code:

1. An Infinite Loop will run the code repeatedly so that Smarc checks the reading from the sensor all the time and act accordingly.
2. Inside the Loop, there is an If/Else block which if TRUE, Smarc will move forward and if FALSE, Smarc will come to stop.
3. The Compare Greater Than block sets the condition for the Switch block. The Compare Greater Than block compares the output of the Sensor Sound Level Reporter block to the value 2. It will return true, when the sensor detects sound levels greater than 2 and false, when it is equal or less than 2.
4. If the output of the If/Else block is true then the Drivebase Move Steering block will move Smarc at speed 55, at an angle of 0 degrees and will keep on doing this for 0.2 seconds. If it is false, the Drivebase Stop block will stop Smarc from moving.

https://youtu.be/oMq_oIyajE0

VI. Navigating Smarc with a Remote Controller

A. Remote Control

In this section, we will see how Smarc can move with the use of a remote control.

There are many different ways to communicate with a robot. Robots are most often controlled using either a wired or a wireless controller or they can run autonomously by getting instructions from their control program (which is what we did with Smarc in the previous chapters).

Erasmus+

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

Remote controlled robots can have various scientific uses including hazardous environments such as extreme heat or cold and radioactive locations, working in the deep ocean to place pipes and cables, as well as exploring the outer space and planets.

https://www.youtube.com/watch?v=ZFEJJx9Zlr0

B. The Remote Control Blocks:

Joystick Widget Show - Presents the Joystick Widget in the Lego BOOST App when this block is activated.

Joystick Widget Hide - Hides the Joystick Widget from the Lego BOOST App when this block is activated.

Joystick Widget Speed Reporter - Displays in real time the current Joystick Widget speed (-100..100). In order to be used in a program it needs to be attached to the bottom of other blocks. In most cases it is used as the steering input to a Drivebase Move Steering block. It can show values from -100 to 100 including two decimal values, for example 78,89.

Joystick Widget Steering Reporter - Displays in real time the current Joystick Widget steering (-100..100). In order to be used in a program it needs to be attached to the bottom of other blocks. In most cases it is used as the speed input to a Drivebase Move Steering block. It can show values from -100 to 100 including two decimal values, for example 78,89.

Sample Programs for Remote Control
Program 1

Erasmus+

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

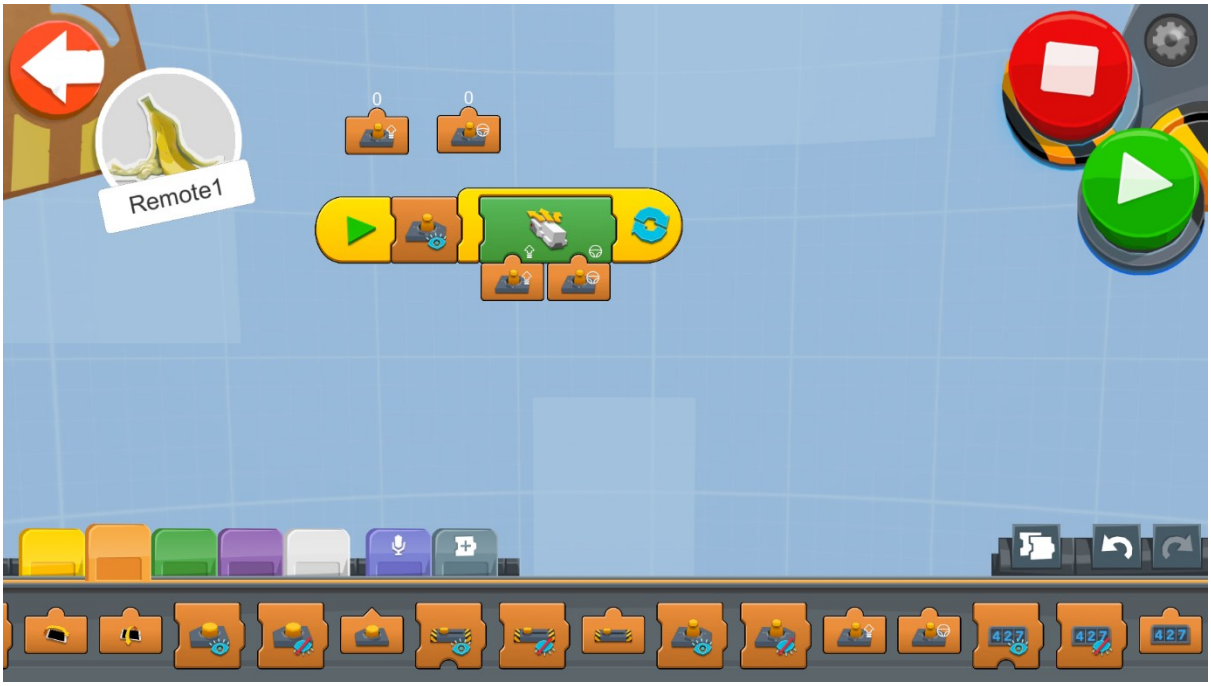


Photo 26-Program - Remote Control

Create a new project on the Creative Canvas and run the block of code:

1. The Joystick Widget Show will show the joystick controller on the app (See Figure).
2. An Infinite Loop will run the code repeatedly so that Smarc checks all the time for the reading from the joystick and act accordingly.
3. Inside the Loop there is a Drivebase Move Steering block which will take as speed input the Joystick Widget Speed Reporter block reading and as direction input the Joystick Widget Steering Reporter block reading.

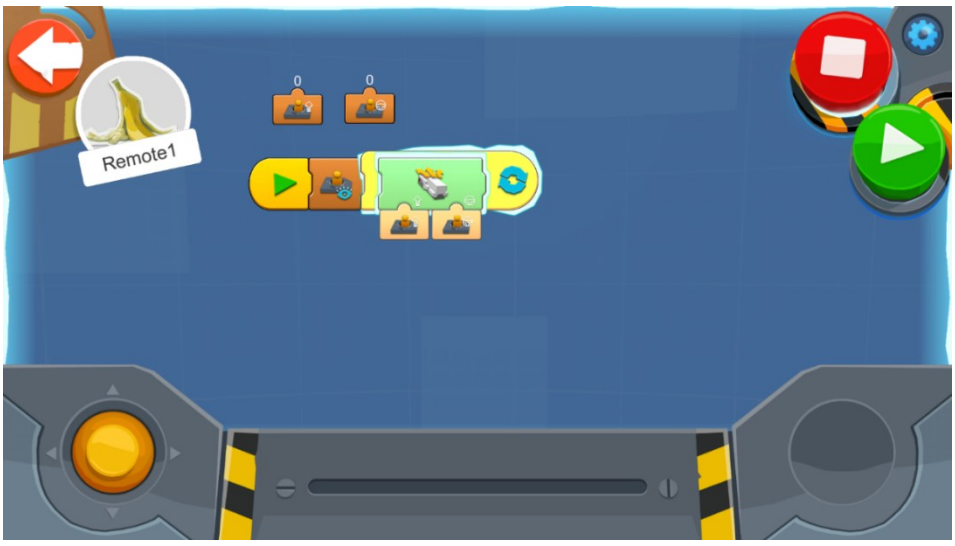


Photo 27-Program - Joystick Controller

Erasmus+

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

Program 2

Photo 28- Program 2 - Remote Control

Sometimes we need more accurate control of Smarc and the input speed is too fast for this. For this we can do a trick in order to reduce the input speed which is sent by the Joystick Widget Speed Reporter block. A Division Operator is used for the speed input of the Drivebase Move block. The input of the Joystick Widget Speed Reporter block is first divided by 4 and then the value is sent as the speed value. For example, if the value reading from the joystick is 100 the speed will be:

$$100 \div 4 = 25$$

So, the real speed input would be 25.

remote control - https://youtu.be/Tswdypjm7bk

Erasmus+

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

www.robotics4deaf.eu