21

TRIGONOMETRY GAME

STEMJAM Teaching Guide

Developing make spaces to promote creativity
around STEM in schools
Acronym: STEMJAM
Project no. 2016-1-ES01-KA201-025470

www.stemjam.eu

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

TRIGONOMETRY GAME

ABSTRACT

The objective is to create a game for students to measure distances and the calculation of angles.

The teacher will draw a geometric figure on a paper board with different color cards, red for the edges and green cards for the vertices.

Students will have to go through these figures with the mBot through requests that the software will ask the user.

DIDACTIC OBJECTIVES

- The student applies the formula Speed = Path / Time.
- Student understands the edge-angle formula of equilateral polygons.
- Student knows the Scalene triangle and their environmental formulas.
- Student knows the area of square and the formulas of the environment.

STEM Subject: Science☒ Technology ☒ Engineering☐ Mathematics☒

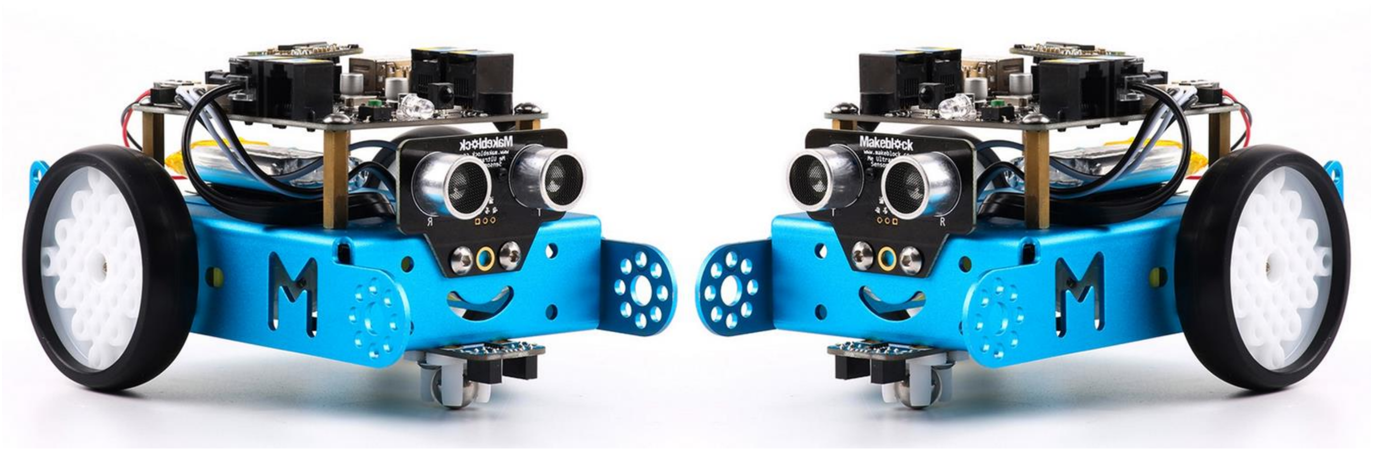
Education Level: 12-14 years☐ 14-16 years☒

PROBLEM STATEMENT

Some students have difficulty calculating angles and measuring distances, so from a game they will perform with the mBots, they will help them carry out these operations.

BOM (Bill of Materials Needed)

- ➢ (x2) mBot => Ref. 90054

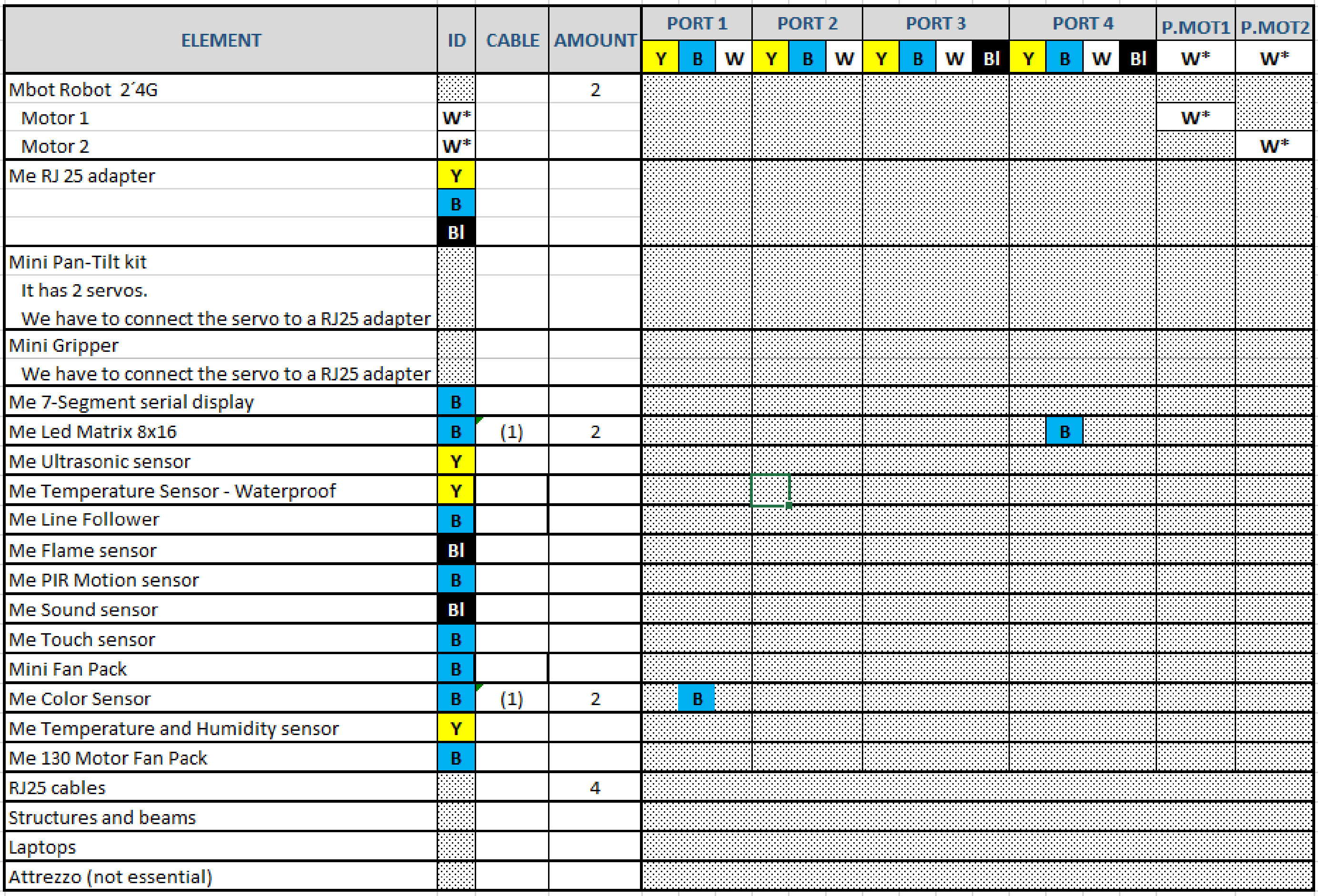


- ❖ mBot Ranger (temporizator)

- ❖ (x2) Colour Sensors for each mBot:

🟨 🟩 🟦 🟥 ⬜ ⬛

⬛ ⬜ 🟥 🟦 🟩 🟨



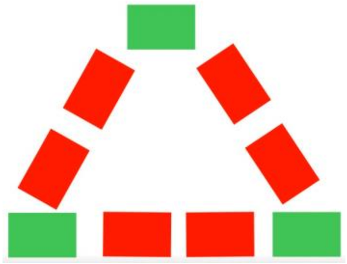
ELEMENT	ID	CABLE	AMOUNT	PORT 1			PORT 2			PORT 3				PORT 4				P.MOT1	P.MOT2
				Y	B	W	Y	B	W	Y	B	W	Bl	Y	B	W	Bl	W*	W*
Mbot Robot 2´4G			2																
Motor 1	W*																	W*	
Motor 2	W*																		W*
Me RJ 25 adapter	Y																		
	B																		
	Bl																		
Mini Pan-Tilt kit																			
It has 2 servos.																			
We have to connect the servo to a RJ25 adapter																			
Mini Gripper																			
We have to connect the servo to a RJ25 adapter																			
Me 7-Segment serial display	B																		
Me Led Matrix 8x16	B	(1)	2												B				
Me Ultrasonic sensor	Y																		
Me Temperature Sensor - Waterproof	Y																		
Me Line Follower	B																		
Me Flame sensor	Bl																		
Me PIR Motion sensor	B																		
Me Sound sensor	Bl																		
Me Touch sensor	B																		
Mini Fan Pack	B																		
Me Color Sensor	B	(1)	2		B														
Me Temperature and Humidity sensor	Y																		
Me 130 Motor Fan Pack	B																		
RJ25 cables			4																
Structures and beams																			
Laptops																			
Attrezzo (not essential)																			

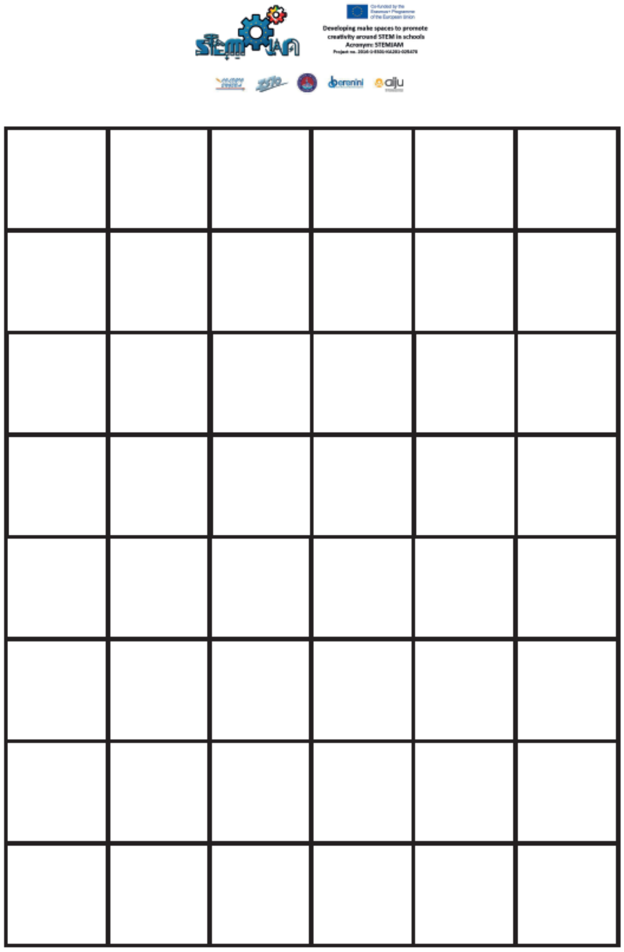
ACTIVITY DESCRIPTION

First version

The activity consists in create a game for students to measure distances and the calculation of angles. The teacher will draw a geometric figure on a paper board with different color cards, red for the edges and green cards for the vertices and checkpoints. Then, students will have to go through these figures with the mBot through requests that the software will ask the user.

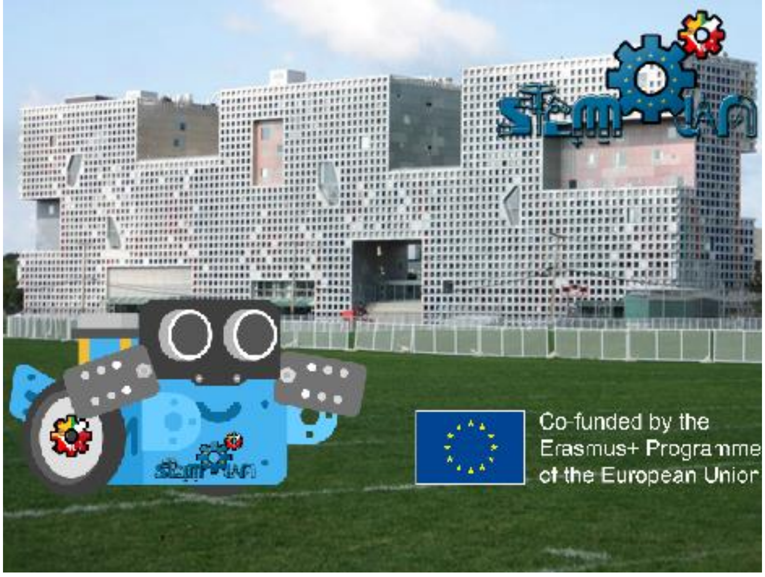
The first step, is design the paper board (you can find in the activity's folder) and the different cards:



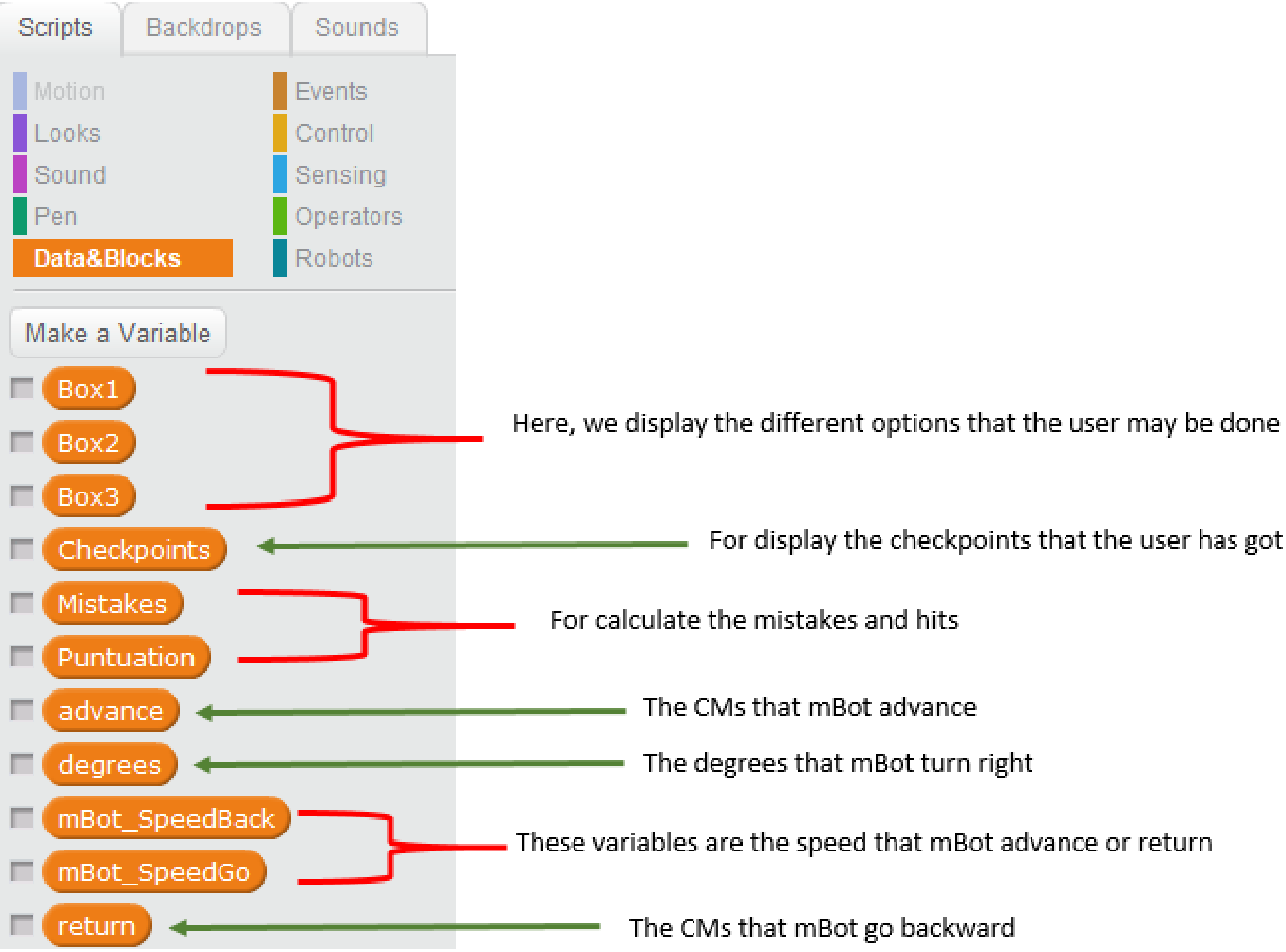


The next step, is create the software for the game:

1. Design the scenes for the game:



2. Start the Programming:
   a. We create the variables that we need in our software:



Here, we display the different options that the user may be done

For display the checkpoints that the user has got

For calculate the mistakes and hits

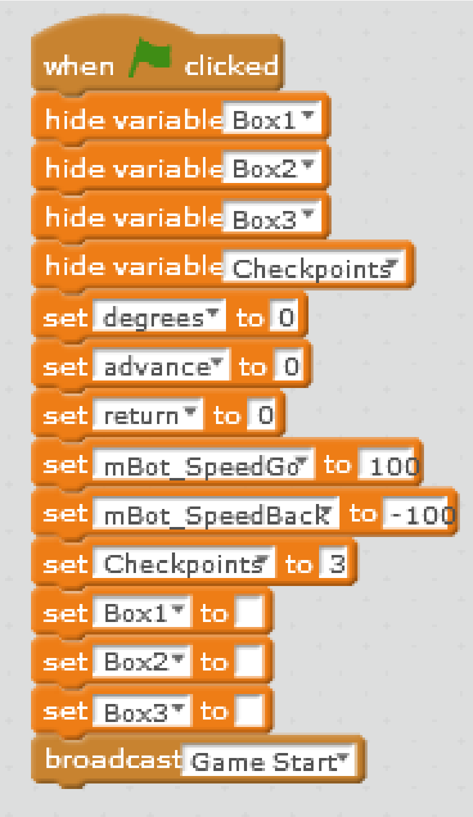
The CMs that mBot advance

The degrees that mBot turn right

These variables are the speed that mBot advance or return

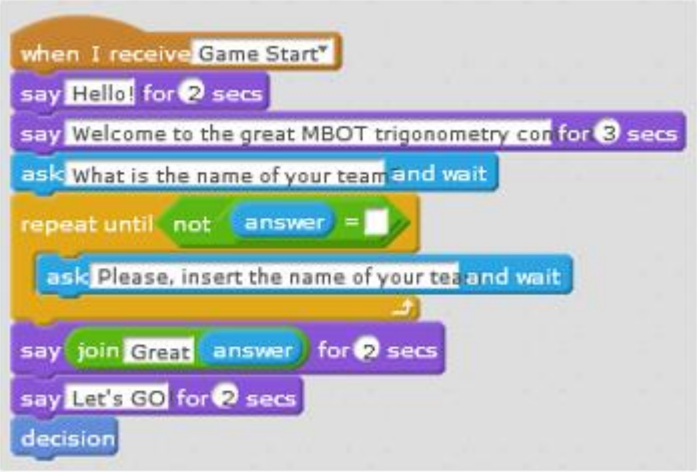
The CMs that mBot go backward

b. The code when the game starts is the following:

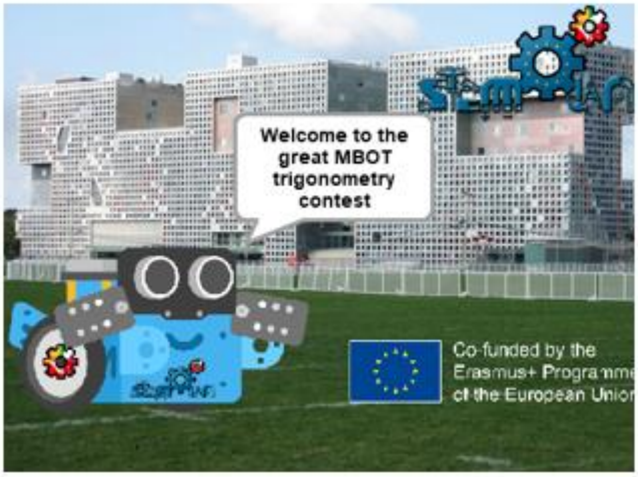


```
when clicked
hide variable Box1
hide variable Box2
hide variable Box3
hide variable Checkpoints
set degrees to 0
set advance to 0
set return to 0
set mBot_SpeedGo to 100
set mBot_SpeedBack to -100
set Checkpoints to 3
set Box1 to
set Box2 to
set Box3 to
broadcast Game Start
```

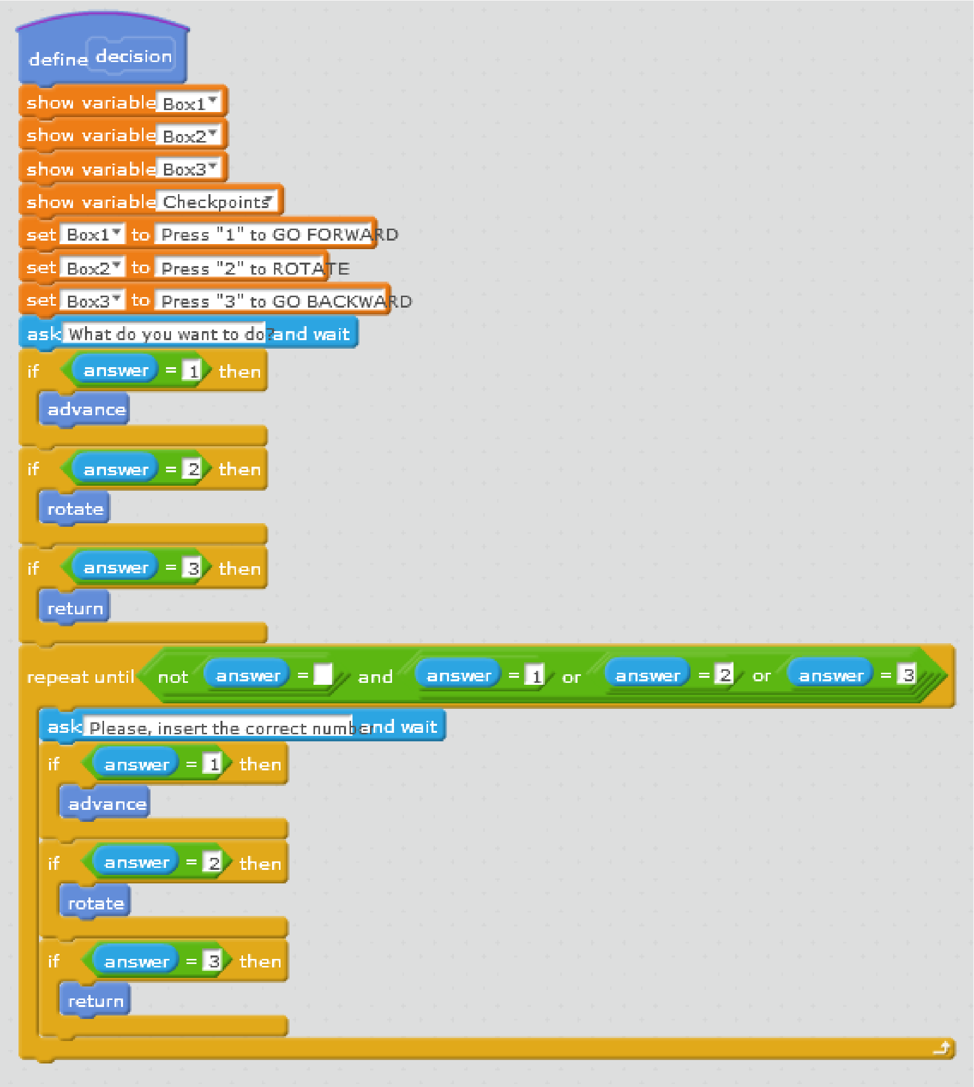
c. The “WELCOME” screen code is:



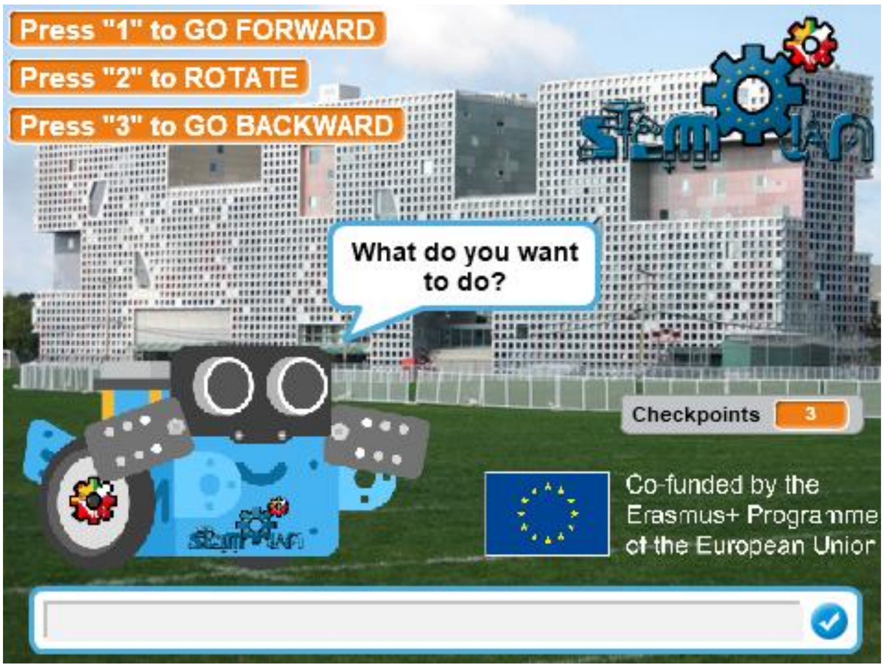
```
when I receive Game Start
say Hello! for 2 secs
say Welcome to the great MBOT trigonometry con for 3 secs
ask What is the name of your team and wait
repeat until not answer =
    ask Please, insert the name of your tea and wait
say join Great answer for 2 secs
say Let's GO for 2 secs
decision
```


Welcome to the great MBOT trigonometry contest
Co-funded by the Erasmus+ Programme of the European Union

d. The software ask to user the operation that he or she wants to do:

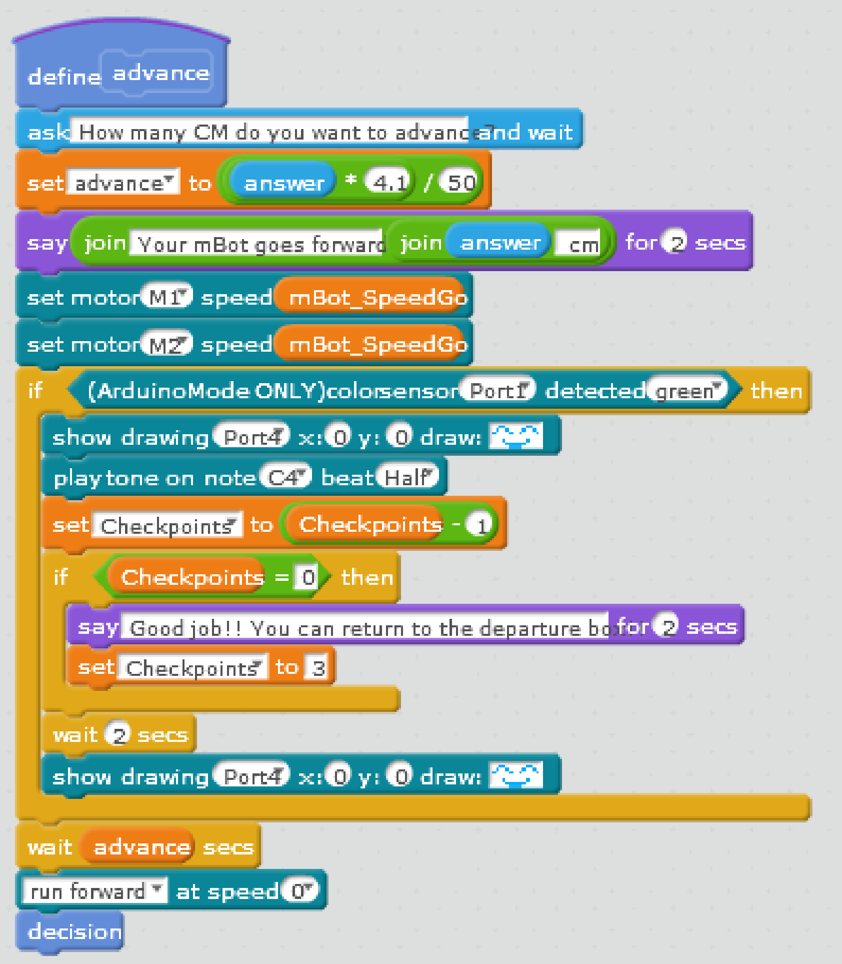


```
define decision
show variable Box1
show variable Box2
show variable Box3
show variable Checkpoints
set Box1 to Press "1" to GO FORWARD
set Box2 to Press "2" to ROTATE
set Box3 to Press "3" to GO BACKWARD
ask What do you want to do? and wait
if answer = 1 then
    advance
if answer = 2 then
    rotate
if answer = 3 then
    return
repeat until not answer = and answer = 1 or answer = 2 or answer = 3
    ask Please, insert the correct numb and wait
    if answer = 1 then
        advance
    if answer = 2 then
        rotate
    if answer = 3 then
        return
```

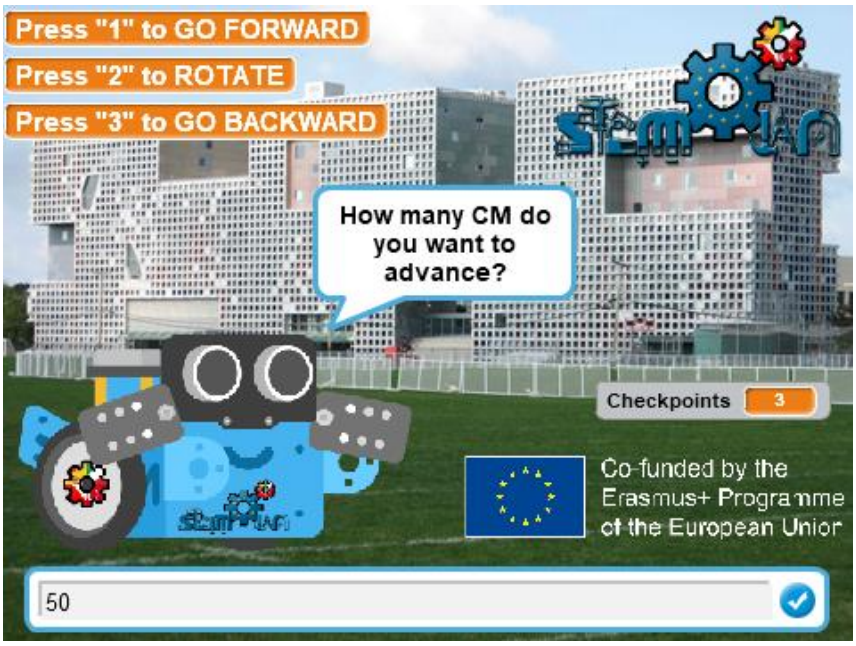


The “checkpoints” indicates the missing checkpoints to pass.

e. If you select the first option, “go forward”, the codes is the following:

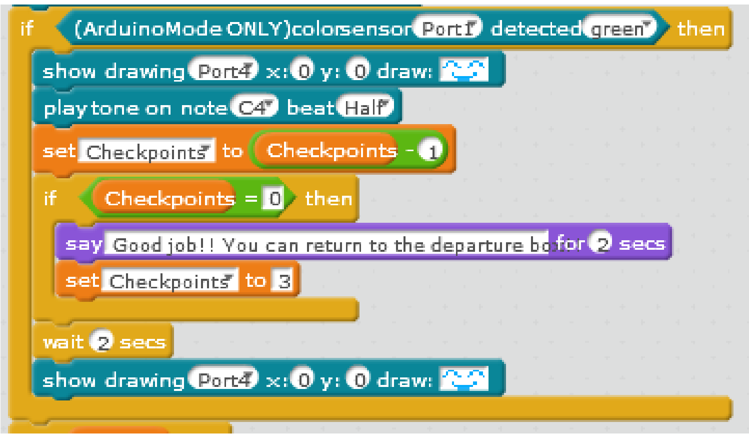


```
define advance
ask How many CM do you want to advanc and wait
set advance to answer * 4.1 / 50
say join Your mBot goes forward join answer cm for 2 secs
set motor M1 speed mBot_SpeedGo
set motor M2 speed mBot_SpeedGo
if (ArduinoMode ONLY)colorsensor Port1 detected green then
    show drawing Port4 x: 0 y: 0 draw:
    play tone on note C4 beat Half
    set Checkpoints to Checkpoints - 1
    if Checkpoints = 0 then
        say Good job!! You can return to the departure bo for 2 secs
        set Checkpoints to 3
    wait 2 secs
    show drawing Port4 x: 0 y: 0 draw:
wait advance secs
run forward at speed 0
decision
```



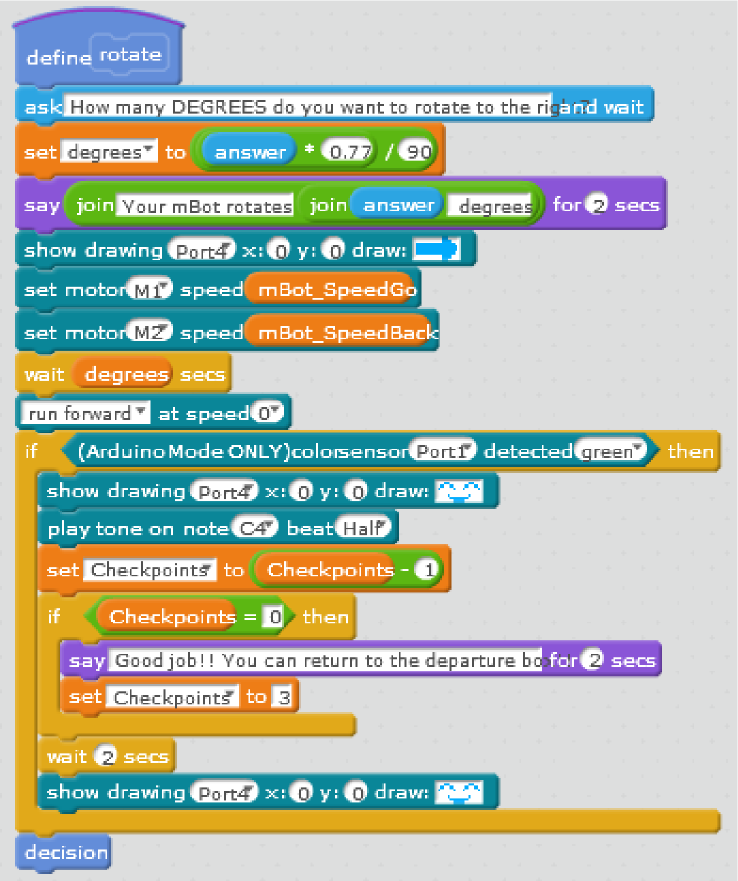
Press "1" to GO FORWARD
Press "2" to ROTATE
Press "3" to GO BACKWARD
How many CM do you want to advance?
Checkpoints 3
Co-funded by the Erasmus+ Programme of the European Union
50

When the mBot passes over a green card, the color sensor will detect it and show a smile.

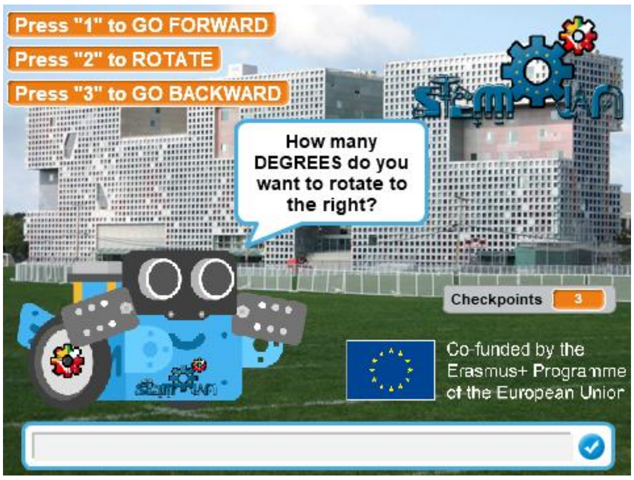


```
if (ArduinoMode ONLY)colorsensor Port1 detected green then
    show drawing Port4 x: 0 y: 0 draw:
    play tone on note C4 beat Half
    set Checkpoints to Checkpoints - 1
    if Checkpoints = 0 then
        say Good job!! You can return to the departure bo for 2 secs
        set Checkpoints to 3
    wait 2 secs
    show drawing Port4 x: 0 y: 0 draw:
```

f. And if the user select the second option, “rotate”:



```
define rotate
ask How many DEGREES do you want to rotate to the rig and wait
set degrees to answer * 0.77 / 90
say join Your mBot rotates join answer degrees for 2 secs
show drawing Port4 x: 0 y: 0 draw:
set motor M1 speed mBot_SpeedGo
set motor M2 speed mBot_SpeedBack
wait degrees secs
run forward at speed 0
if (ArduinoMode ONLY)colorsensor Port1 detected green then
    show drawing Port4 x: 0 y: 0 draw:
    play tone on note C4 beat Half
    set Checkpoints to Checkpoints - 1
    if Checkpoints = 0 then
        say Good job!! You can return to the departure bo for 2 secs
        set Checkpoints to 3
    wait 2 secs
    show drawing Port4 x: 0 y: 0 draw:
decision
```



Press "1" to GO FORWARD

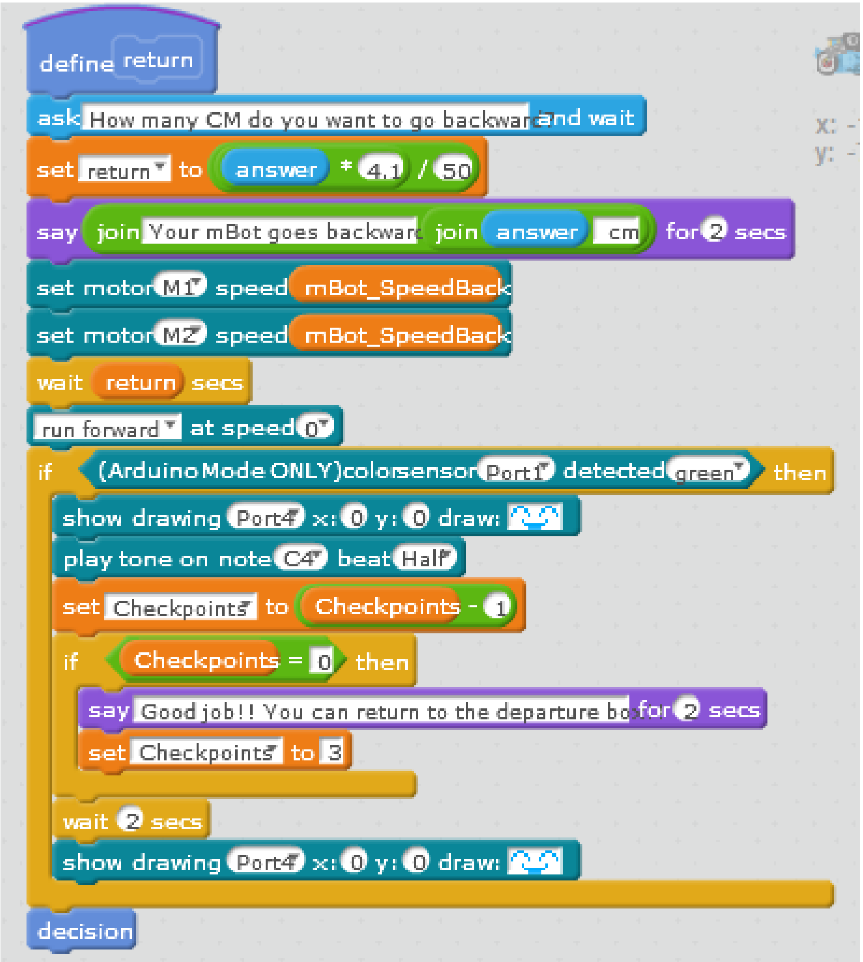
Press "2" to ROTATE

Press "3" to GO BACKWARD

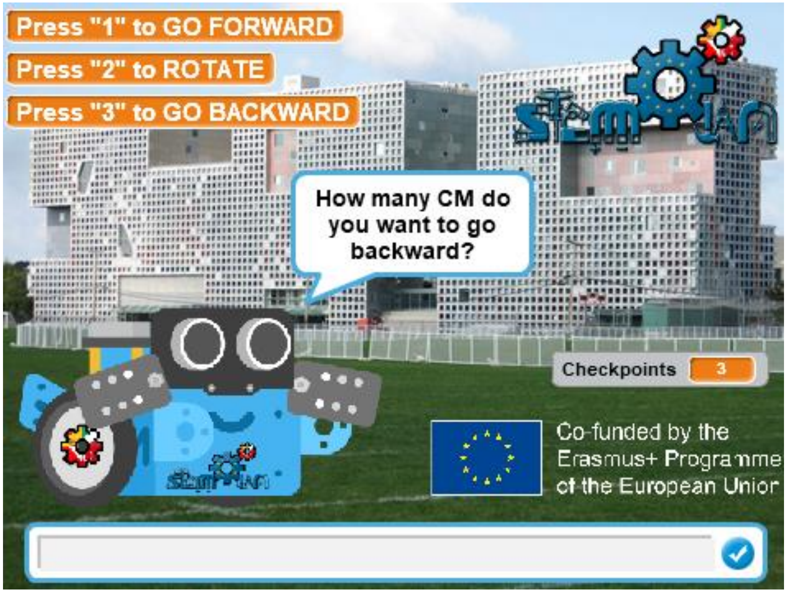
How many DEGREES do you want to rotate to the right?

Checkpoints 3

g. For return, the code is very similar than the first option:



```
define return
ask How many CM do you want to go backwar and wait
set return to answer * 4.1 / 50
say join Your mBot goes backwar join answer cm for 2 secs
set motor M1 speed mBot_SpeedBack
set motor M2 speed mBot_SpeedBack
wait return secs
run forward at speed 0
if (Arduino Mode ONLY)colorsensor Port1 detected green then
    show drawing Port4 x: 0 y: 0 draw:
    play tone on note C4 beat Half
    set Checkpoints to Checkpoints - 1
    if Checkpoints = 0 then
        say Good job!! You can return to the departure bo for 2 secs
        set Checkpoints to 3
    wait 2 secs
    show drawing Port4 x: 0 y: 0 draw:
decision
```



Press "1" to GO FORWARD

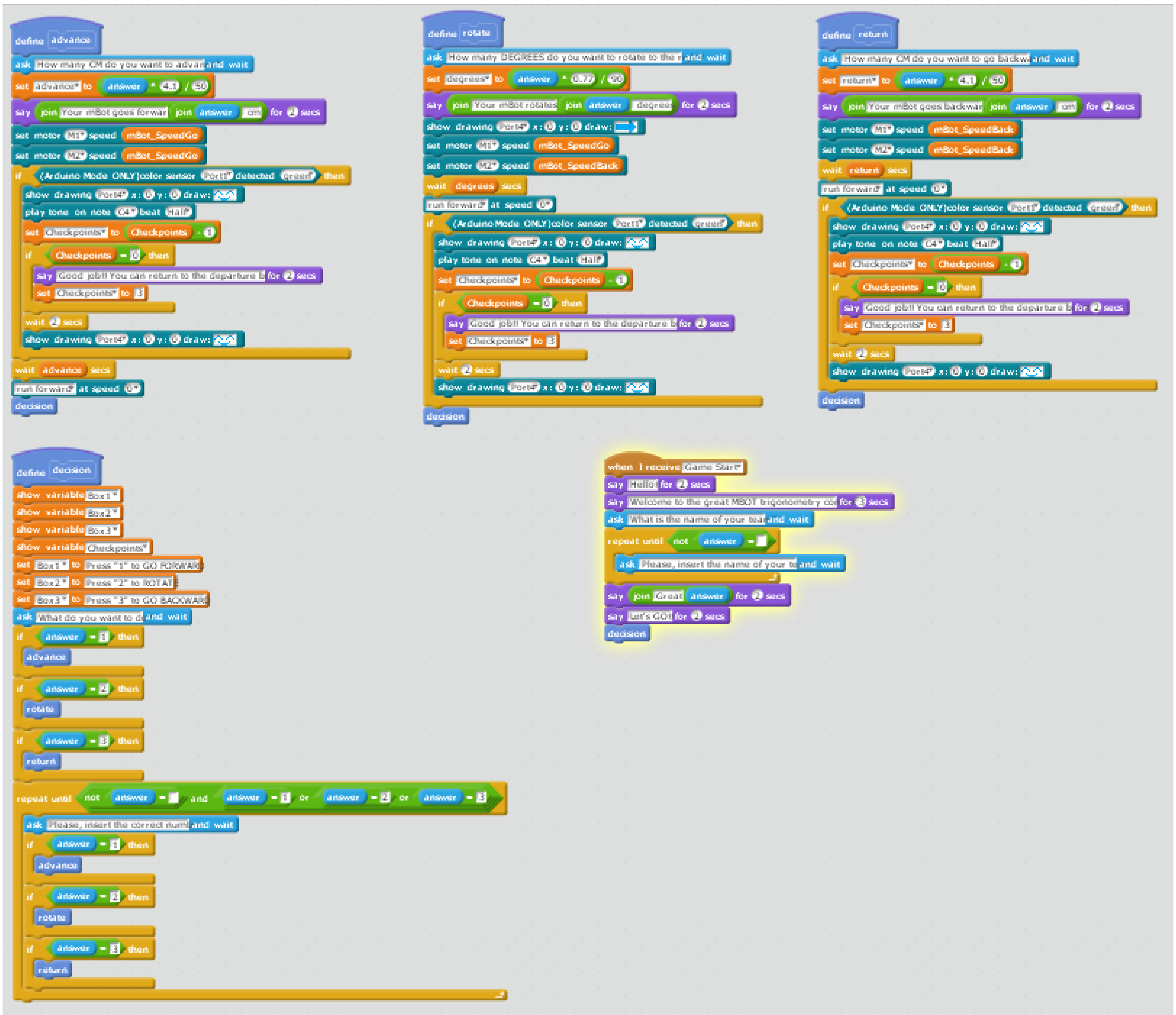
Press "2" to ROTATE

Press "3" to GO BACKWARD

How many CM do you want to go backward?

Checkpoints 3

3. <u>The resulting code would be</u>:



```
define advance
ask How many CM do you want to advan and wait
set advance to ((answer * 4.1) / 50)
say (join Your mBot goes forwar (join answer cm)) for 2 secs
set motor M1 speed mBot_SpeedGo
set motor M2 speed mBot_SpeedGo
if <(Arduino Mode ONLY)color sensor Port1 detected green> then
    show drawing Port4 x: 0 y: 0 draw:
    play tone on note C4 beat Half
    set Checkpoints to (Checkpoints - 1)
    if <Checkpoints = 0> then
        say Good job!! You can return to the departure b for 2 secs
        set Checkpoints to 3
    wait 2 secs
    show drawing Port4 x: 0 y: 0 draw:
wait advance secs
run forward at speed 0
decision
```

```
define rotate
ask How many DEGREES do you want to rotate to the r and wait
set degrees to ((answer * 0.77) / 90)
say (join Your mBot rotates (join answer degrees)) for 2 secs
show drawing Port4 x: 0 y: 0 draw:
set motor M1 speed mBot_SpeedGo
set motor M2 speed mBot_SpeedBack
wait degrees secs
run forward at speed 0
if <(Arduino Mode ONLY)color sensor Port1 detected green> then
    show drawing Port4 x: 0 y: 0 draw:
    play tone on note C4 beat Half
    set Checkpoints to (Checkpoints - 1)
    if <Checkpoints = 0> then
        say Good job!! You can return to the departure b for 2 secs
        set Checkpoints to 3
    wait 2 secs
    show drawing Port4 x: 0 y: 0 draw:
decision
```

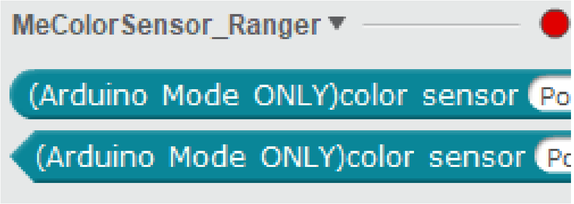
```
define return
ask How many CM do you want to go backw and wait
set return to ((answer * 4.1) / 50)
say (join Your mBot goes backwar (join answer cm)) for 2 secs
set motor M1 speed mBot_SpeedBack
set motor M2 speed mBot_SpeedBack
wait return secs
run forward at speed 0
if <(Arduino Mode ONLY)color sensor Port1 detected green> then
    show drawing Port4 x: 0 y: 0 draw:
    play tone on note C4 beat Half
    set Checkpoints to (Checkpoints - 1)
    if <Checkpoints = 0> then
        say Good job!! You can return to the departure b for 2 secs
        set Checkpoints to 3
    wait 2 secs
    show drawing Port4 x: 0 y: 0 draw:
decision
```

```
define decision
show variable Box1
show variable Box2
show variable Box3
show variable Checkpoints
set Box1 to Press "1" to GO FORWARD
set Box2 to Press "2" to ROTATE
set Box3 to Press "3" to GO BACKWARD
ask What do you want to d and wait
if <answer = 1> then
    advance
if <answer = 2> then
    rotate
if <answer = 3> then
    return
repeat until <not <answer = []> and <<answer = 1> or <<answer = 2> or <answer = 3>>>>
    ask Please, insert the correct numb and wait
    if <answer = 1> then
        advance
    if <answer = 2> then
        rotate
    if <answer = 3> then
        return
```

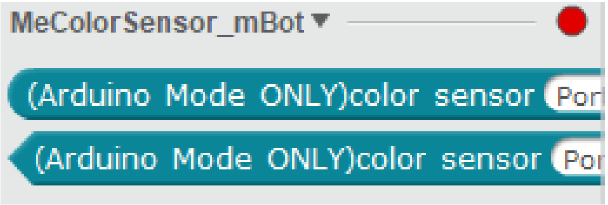
```
when I receive Game Start
say Hello! for 2 secs
say Welcome to the great MBOT trigonometry co for 3 secs
ask What is the name of your tea and wait
repeat until <not <answer = []>>
    ask Please, insert the name of your te and wait
say (join Great answer) for 2 secs
say Let's GO! for 2 secs
decision
```

For <u>install the library on mBlock</u>:

➢ We need to install the library on mBlock:



```
MeColorSensor_Ranger ▼
(Arduino Mode ONLY)color sensor Po
(Arduino Mode ONLY)color sensor Po
```

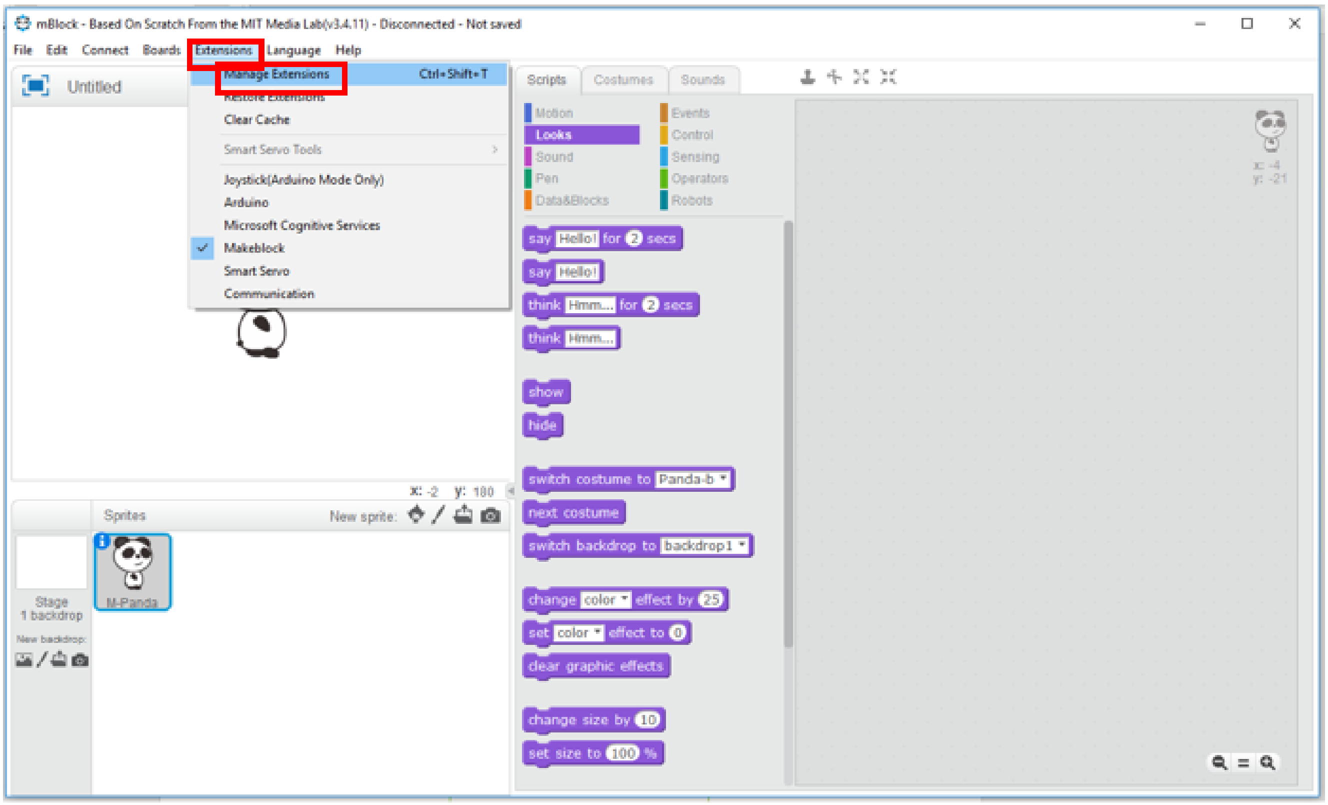


```
MeColorSensor_mBot ▼
(Arduino Mode ONLY)color sensor Por
(Arduino Mode ONLY)color sensor Por
```

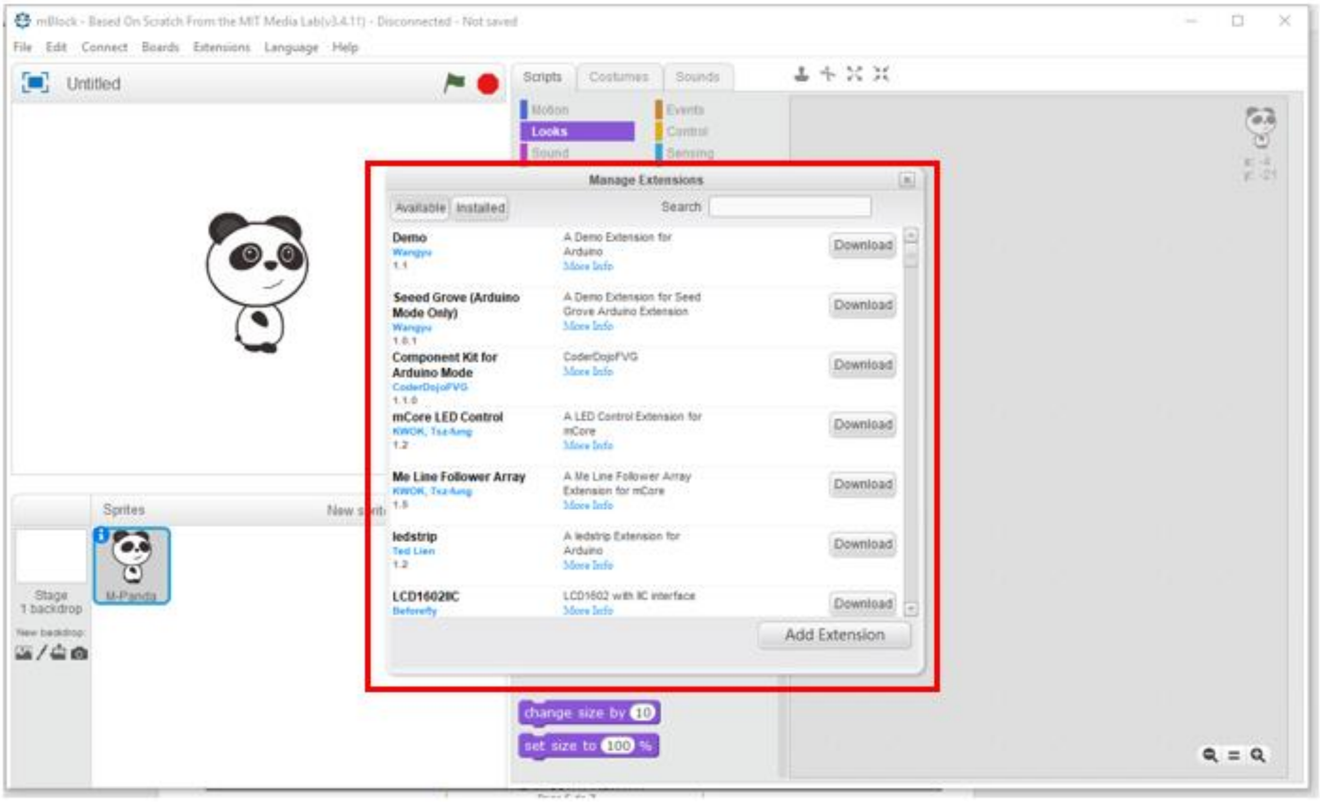
➢ **Nowadays, it only works if we upload the code on the board.**

For install the colour sensor library, we follow the next steps:

1. Start the mBlock software and go to “Extensions” => “Manage Extensions”:

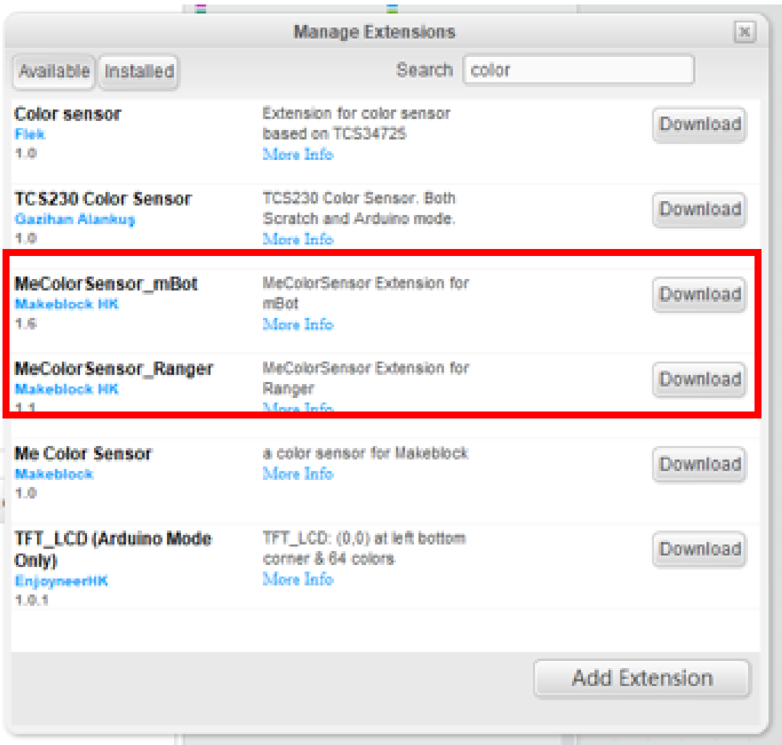


2. A new window will appear:



Demo Wangyu 1.1	A Demo Extension for Arduino More Info	Download
Seeed Grove (Arduino Mode Only) Wangyu 1.0.1	A Demo Extension for Seed Grove Arduino Extension More Info	Download
Component Kit for Arduino Mode CoderDojoFVG 1.1.0	CoderDojoFVG More Info	Download
mCore LED Control KWOK, Tsz-Aung 1.2	A LED Control Extension for mCore More Info	Download
Me Line Follower Array KWOK, Tsz-Aung 1.5	A Me Line Follower Array Extension for mCore More Info	Download
ledstrip Ted Lien 1.2	A ledstrip Extension for Arduino More Info	Download
LCD1602IIC Beforefly	LCD1602 with IIC interface More Info	Download

3. Insert the “color” word in the searcher:



Manage Extensions

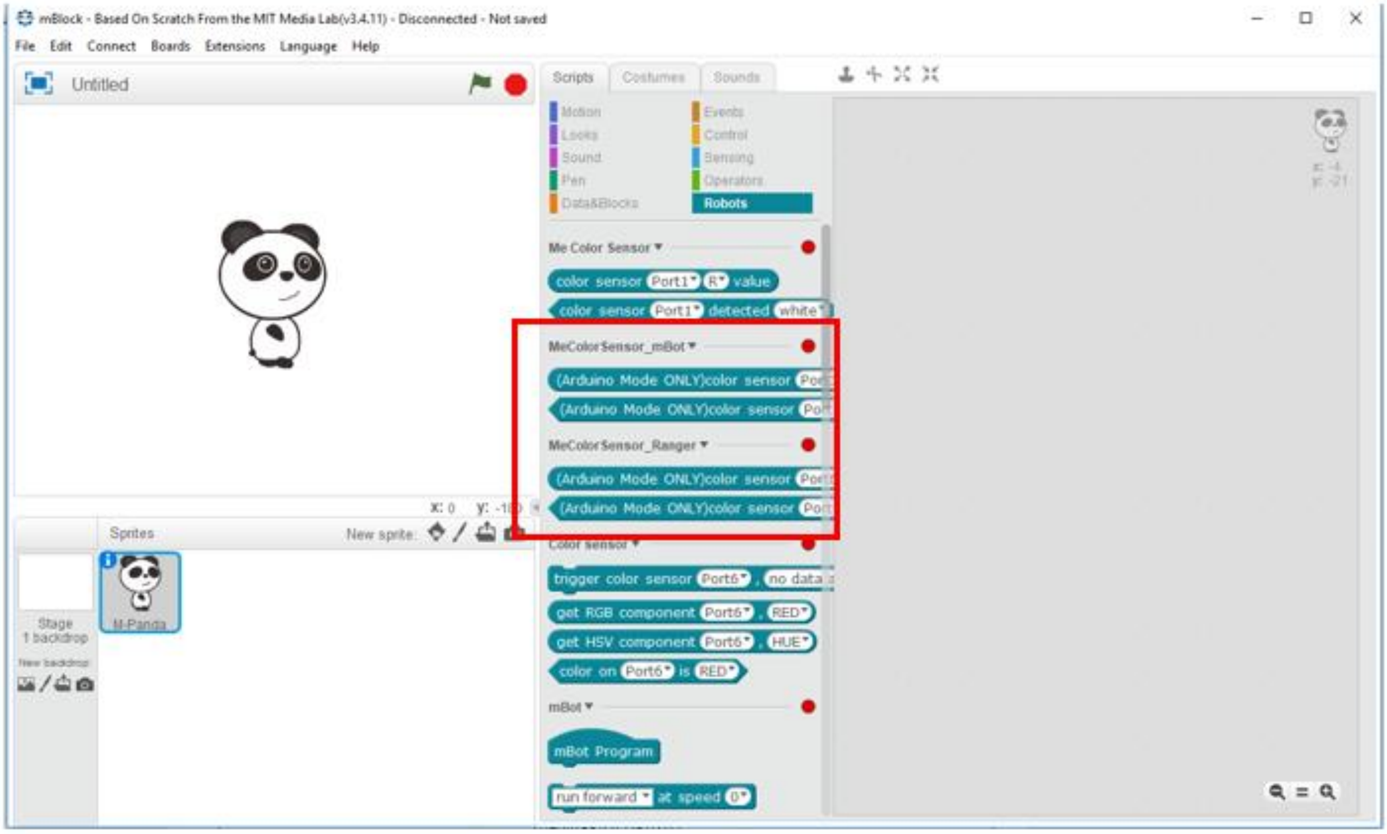
Available | Installed | Search: color

Color sensor Flek 1.0	Extension for color sensor based on TCS34725 More Info	Download
TCS230 Color Sensor Gazihan Alankuş 1.0	TCS230 Color Sensor. Both Scratch and Arduino mode. More Info	Download
MeColorSensor_mBot Makeblock HK 1.6	MeColorSensor Extension for mBot More Info	Download
MeColorSensor_Ranger Makeblock HK 1.1	MeColorSensor Extension for Ranger More Info	Download
Me Color Sensor Makeblock 1.0	a color sensor for Makeblock More Info	Download
TFT_LCD (Arduino Mode Only) EnjoyneerHK 1.0.1	TFT_LCD: (0,0) at left bottom corner & 64 colors More Info	Download

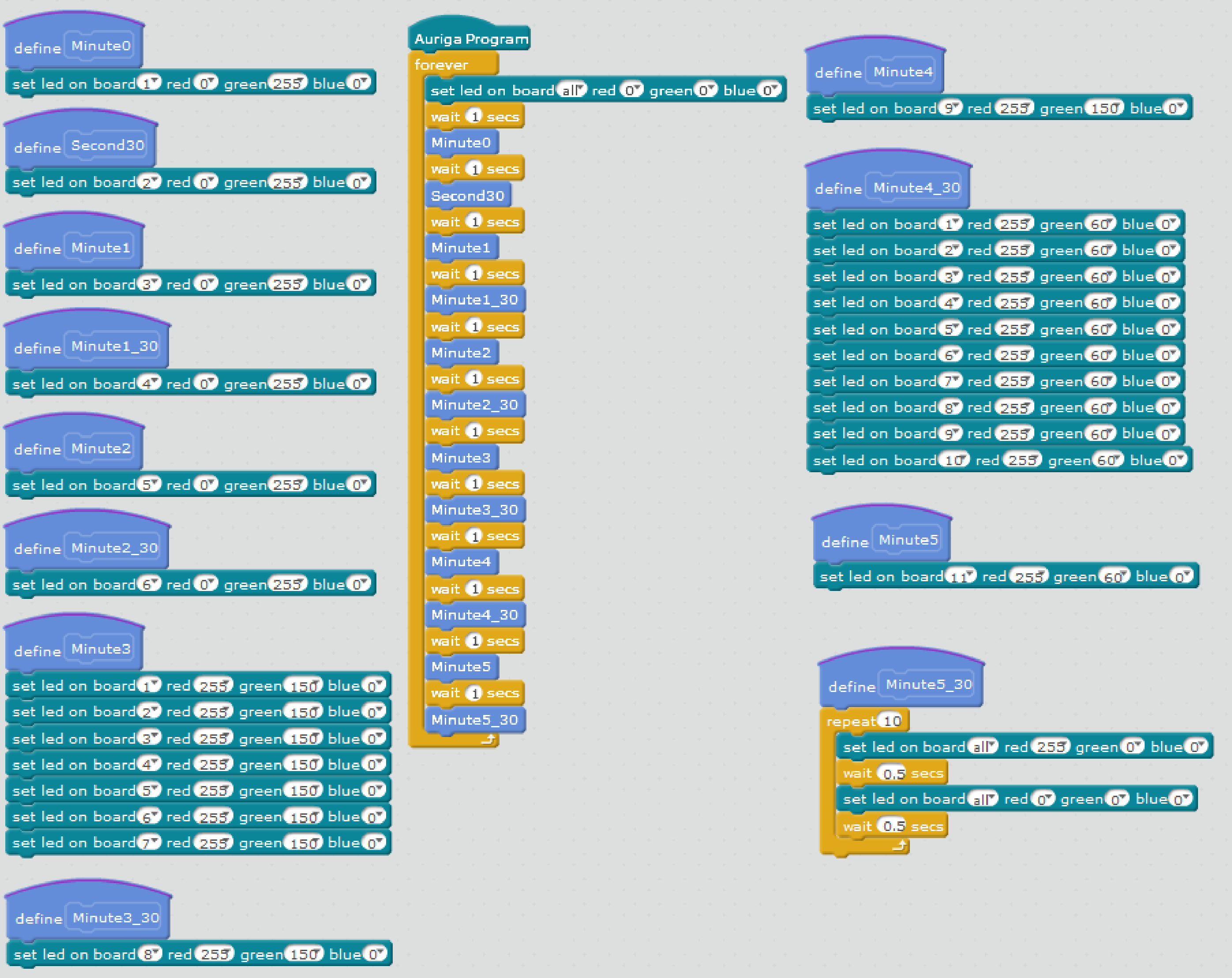
Add Extension

4. The “MeColorSensor_mBot” and the “MeColorSensor_Ranger” will appear. Now, click on “download”.

5. If everything went well, the installed libraries will appear in the "robots" section.


mBlock - Based On Scratch From the MIT Media Lab(v3.4.11) - Disconnected - Not saved
File
Edit
Connect
Boards
Extensions
Language
Help
Untitled
Scripts
Costumes
Sounds
Motion
Looks
Sound
Pen
Data&Blocks
Events
Control
Sensing
Operators
Robots
Me Color Sensor
color sensor Port1 R value
color sensor Port1 detected white
MeColorSensor_mBot
(Arduino Mode ONLY)color sensor Port
(Arduino Mode ONLY)color sensor Port
MeColorSensor_Ranger
(Arduino Mode ONLY)color sensor Port
(Arduino Mode ONLY)color sensor Port
Color sensor
trigger color sensor Port6 , no data
get RGB component Port6 , RED
get HSV component Port6 , HUE
color on Port6 is RED
mBot
mBot Program
run forward at speed 0
Sprites
New sprite:
Stage
1 backdrop
New backdrop:
M-Panda
x: 0 y: -1

Now, we develop the temporizator code with mBot Ranger:

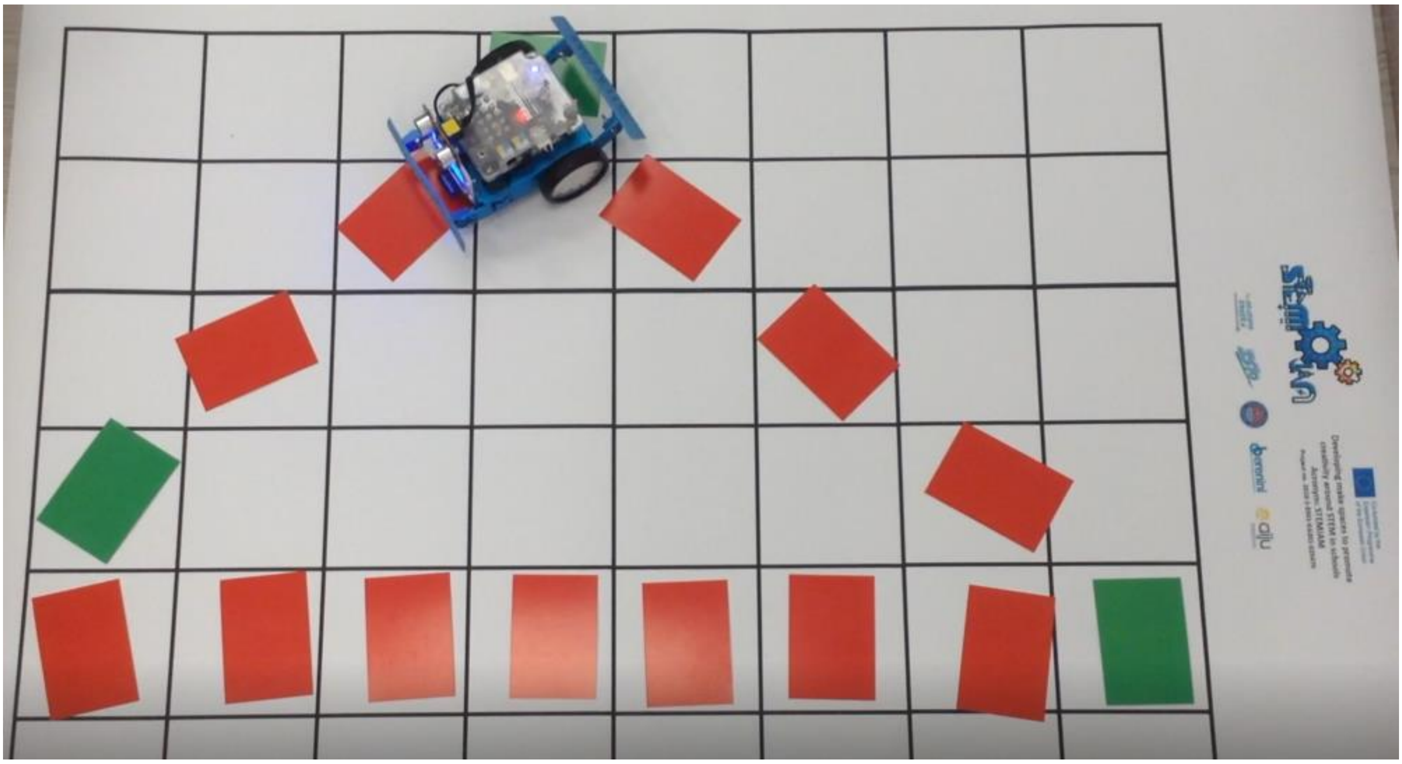


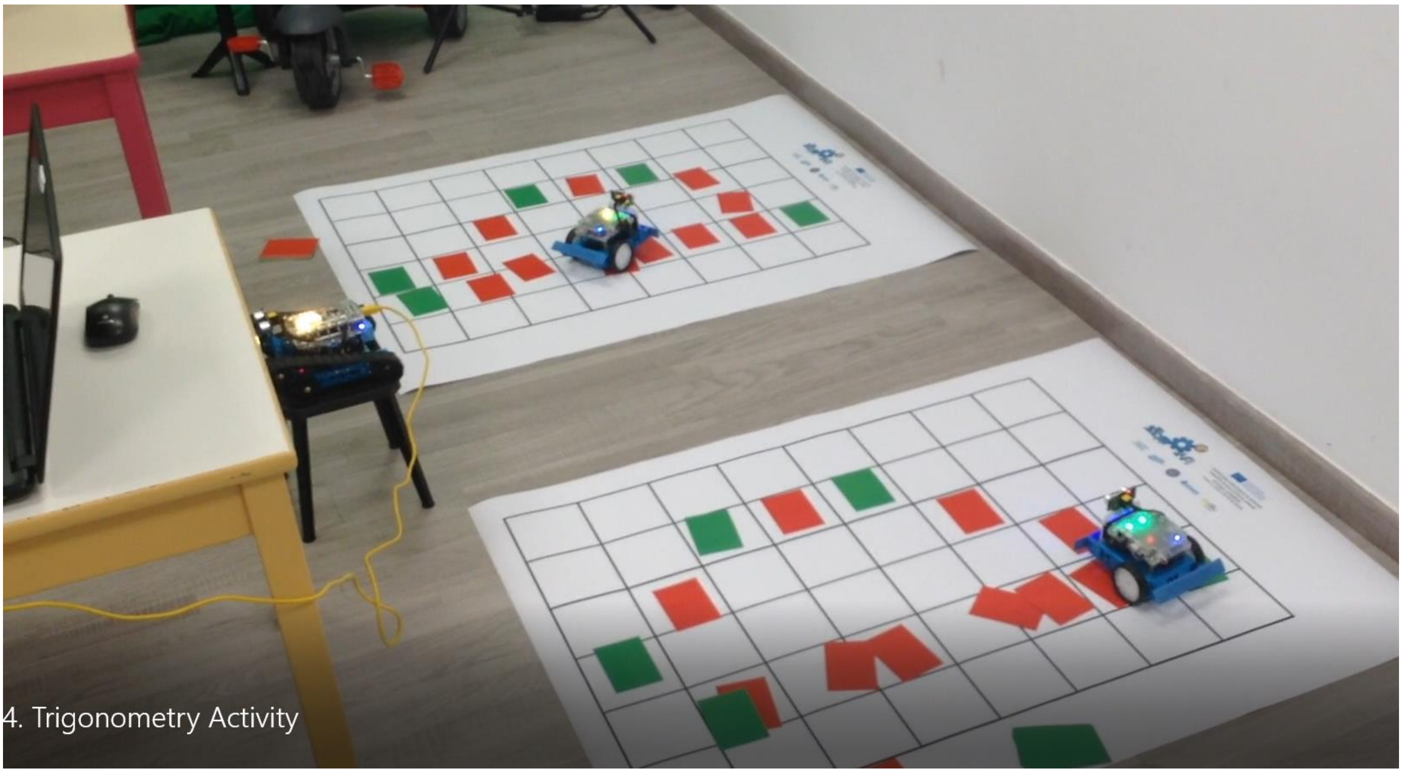
```
define Minute0
set led on board 1 red 0 green 255 blue 0

define Second30
set led on board 2 red 0 green 255 blue 0

define Minute1
set led on board 3 red 0 green 255 blue 0

define Minute1_30
set led on board 4 red 0 green 255 blue 0

define Minute2
set led on board 5 red 0 green 255 blue 0

define Minute2_30
set led on board 6 red 0 green 255 blue 0

define Minute3
set led on board 1 red 255 green 150 blue 0
set led on board 2 red 255 green 150 blue 0
set led on board 3 red 255 green 150 blue 0
set led on board 4 red 255 green 150 blue 0
set led on board 5 red 255 green 150 blue 0
set led on board 6 red 255 green 150 blue 0
set led on board 7 red 255 green 150 blue 0

define Minute3_30
set led on board 8 red 255 green 150 blue 0
```

```
Auriga Program
forever
    set led on board all red 0 green 0 blue 0
    wait 1 secs
    Minute0
    wait 1 secs
    Second30
    wait 1 secs
    Minute1
    wait 1 secs
    Minute1_30
    wait 1 secs
    Minute2
    wait 1 secs
    Minute2_30
    wait 1 secs
    Minute3
    wait 1 secs
    Minute3_30
    wait 1 secs
    Minute4
    wait 1 secs
    Minute4_30
    wait 1 secs
    Minute5
    wait 1 secs
    Minute5_30
```

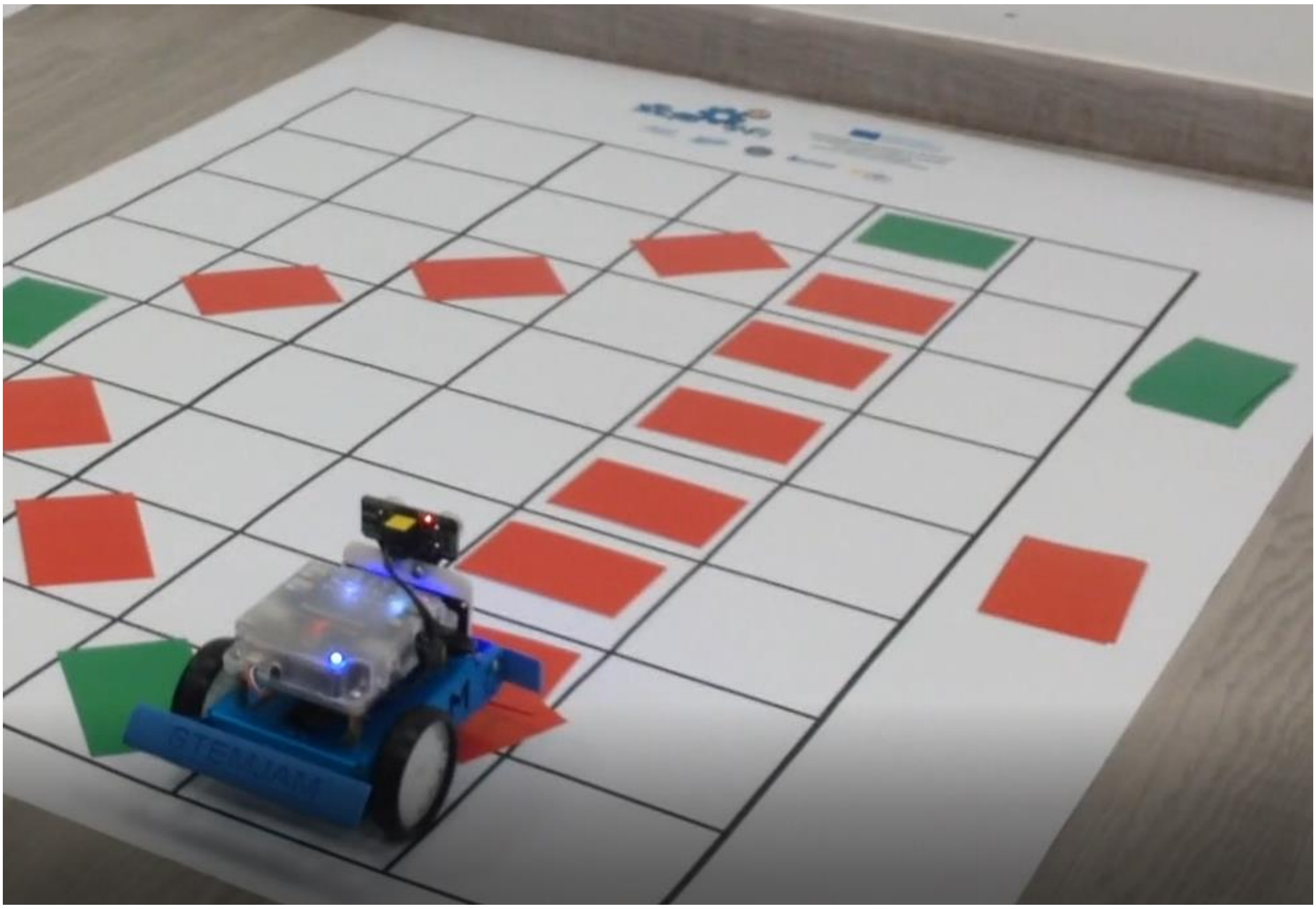
```
define Minute4
set led on board 9 red 255 green 150 blue 0

define Minute4_30
set led on board 1 red 255 green 60 blue 0
set led on board 2 red 255 green 60 blue 0
set led on board 3 red 255 green 60 blue 0
set led on board 4 red 255 green 60 blue 0
set led on board 5 red 255 green 60 blue 0
set led on board 6 red 255 green 60 blue 0
set led on board 7 red 255 green 60 blue 0
set led on board 8 red 255 green 60 blue 0
set led on board 9 red 255 green 60 blue 0
set led on board 10 red 255 green 60 blue 0

define Minute5
set led on board 11 red 255 green 60 blue 0

define Minute5_30
repeat 10
    set led on board all red 255 green 0 blue 0
    wait 0.5 secs
    set led on board all red 0 green 0 blue 0
    wait 0.5 secs
```

For finally the activity description, we show some images for the activity:



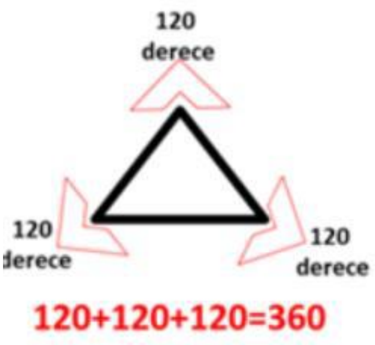


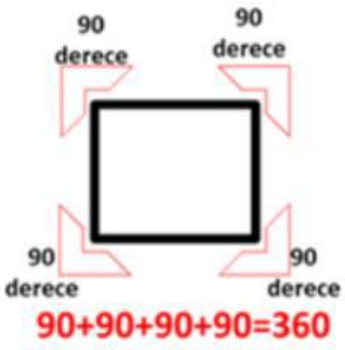
4. Trigonometry Activity

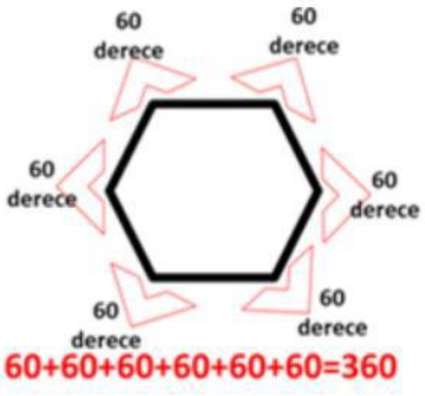


Second Version

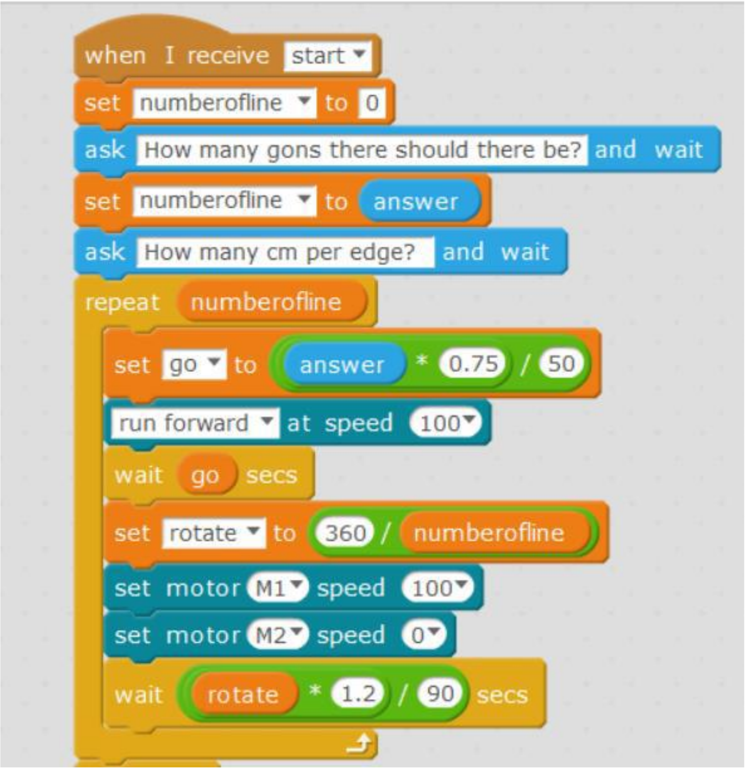
Step 1: We calculated the distance for mBot. We accepted the speed option as 100.





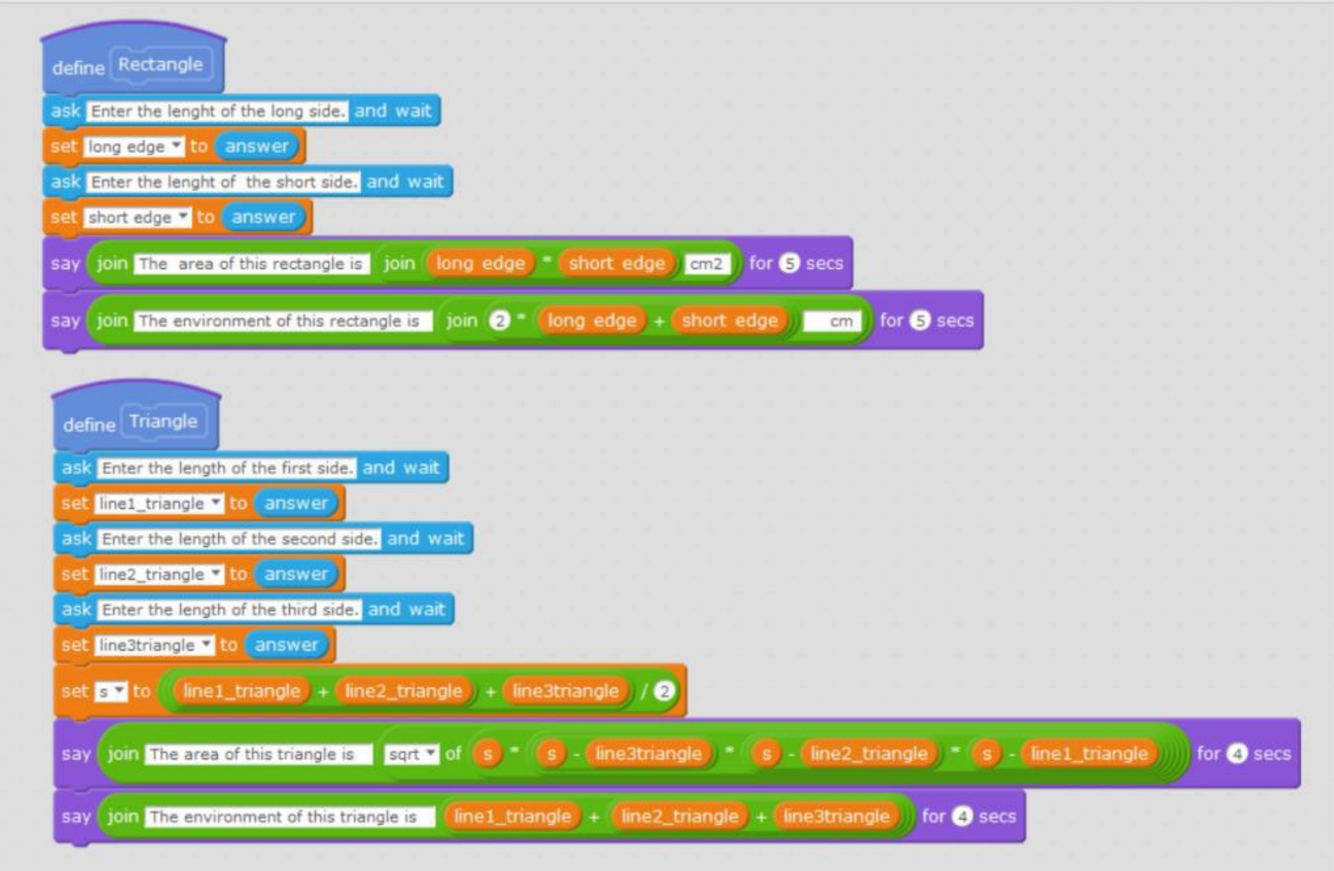


Step 2: Angle-to-edge coding, improvement and turning coding are performed on the MBot.



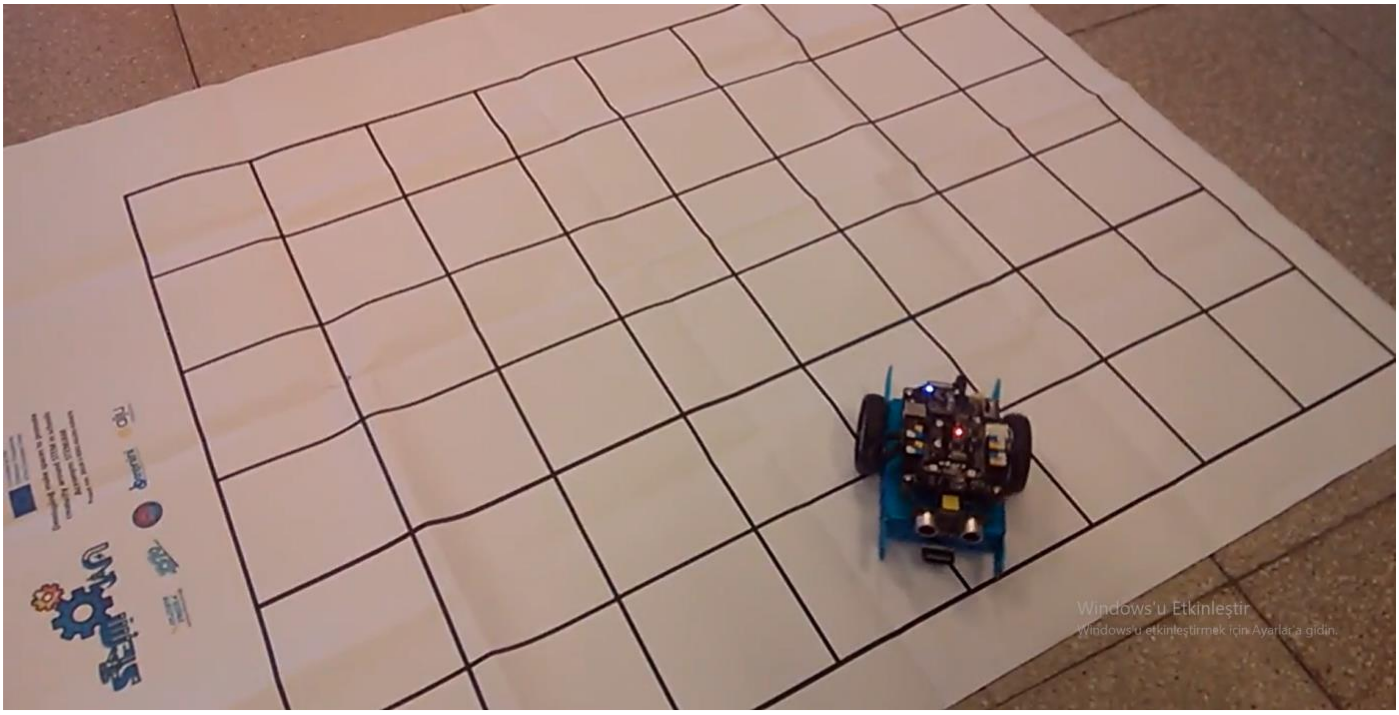
```
when I receive start
set numberofline to 0
ask How many gons there should there be? and wait
set numberofline to answer
ask How many cm per edge? and wait
repeat numberofline
    set go to answer * 0.75 / 50
    run forward at speed 100
    wait go secs
    set rotate to 360 / numberofline
    set motor M1 speed 100
    set motor M2 speed 0
    wait rotate * 1.2 / 90 secs
```

Step 3: We wrote area and perimeter calculation formulas on mBot.

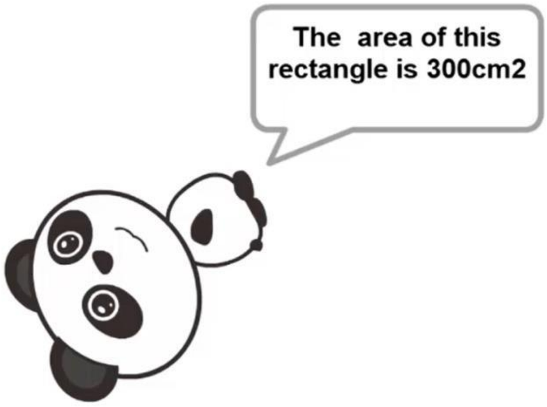


```
define Rectangle
ask [Enter the lenght of the long side.] and wait
set [long edge ▾] to (answer)
ask [Enter the lenght of  the short side.] and wait
set [short edge ▾] to (answer)
say (join [The  area of this rectangle is] (join ((long edge) * (short edge)) [cm2])) for (5) secs
say (join [The environment of this rectangle is] (join ((2) * ((long edge) + (short edge))) [   cm])) for (5) secs

define Triangle
ask [Enter the length of the first side.] and wait
set [line1_triangle ▾] to (answer)
ask [Enter the length of the second side.] and wait
set [line2_triangle ▾] to (answer)
ask [Enter the length of the third side.] and wait
set [line3triangle ▾] to (answer)
set [s ▾] to ((((line1_triangle) + (line2_triangle)) + (line3triangle)) / (2))
say (join [The area of this triangle is] ([sqrt ▾] of ((s) * (((s) - (line3triangle)) * (((s) - (line2_triangle)) * ((s) - (line1_triangle))))))) for (4) secs
say (join [The environment of this triangle is] (((line1_triangle) + (line2_triangle)) + (line3triangle))) for (4) secs
```

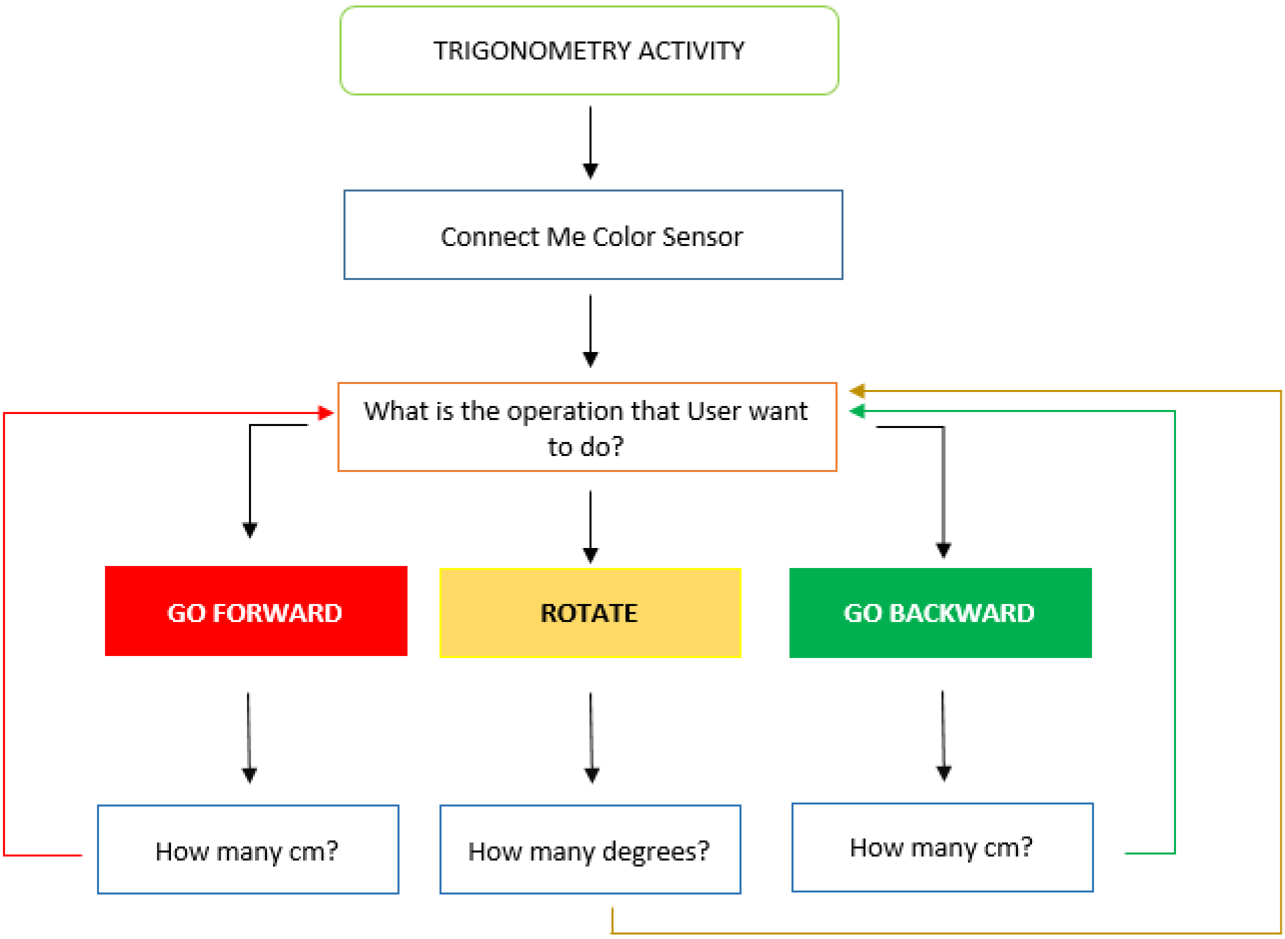
Step 4: We used geometric motions by rulers.



Step 5: We have calculated the polygon area and the perimeter on the screen.



FLOW CHART



STUDENT'S EVALUATION

All students improved themselves about Geometrical shapes and calculating area of polygons.

BIBLIOGRAPHY

https://www.makeblock.es/productos/sensor_color/

MORE INFORMATION

<u>First Version</u>:

1) V = x / t and time and distance with mBot. Distance is adjusted with keyboard, time/distance were formuled and mBot moved or made a turning. In this activity, it is assumed that the MBot takes an average of 12 cm in one second but depends on battery power.

2) We put the colour sensor on the mBot, mBot sense green colour as corner and red colour as edge.

<u>Second Version</u>:

1) We worked on geometry in this activity. We calculated the outside angle of an equilateral polygon. From the user, it is necessary to know how many edges and equilateral polygons should precede the MBot; and then how many centimetres each side should be. The motion is again v = x / t formula.

2) In this activity, we calculated the area and circumference of the geometric bodies. We have defined the formula for each object separately.