



ROBO SIMULATOR

USER MANUAL

Start

Cofinanciado por el
programa Erasmus+
de la Unión Europea



Secretaria Regional
de Educação, Ciência e Tecnologia
Direção Regional de Educação



TABLE OF CONTENTS

1. Introduction
2. How to play?
3. First level tutorial
4. Robot City Community
5. Create and share your own projects



INTRODUCTION

Robot City is a didactic puzzle game which allows us to solve a total of 16 levels by means of small programming commands in coding language. We will use a small robot in the game and our programming skills to control it and solve the different puzzles in each level. It has also a community where you can share your robot projects and get to know the project from other young people.

At the beginning select the language in which you want to play. You can choose between English, Spanish, Portuguese, Greek and Polish. Please note that the programming commands are in English, regardless of the language you select.

HOW TO PLAY?

To start, click on the Start button.

ROBO SIMULATOR

START

COMMUNITY

PRIVACY POLICY

Next we can see the level selector, we can click on the one we want to play to select it. At the beginning you only see level 1, as you complete the levels you can choose more.

LEVEL



BACK

After selecting the desired level and reading the corresponding level introduction, we can see that by clicking on the upper left icon (house), three buttons are displayed on the left and a command console on the right.



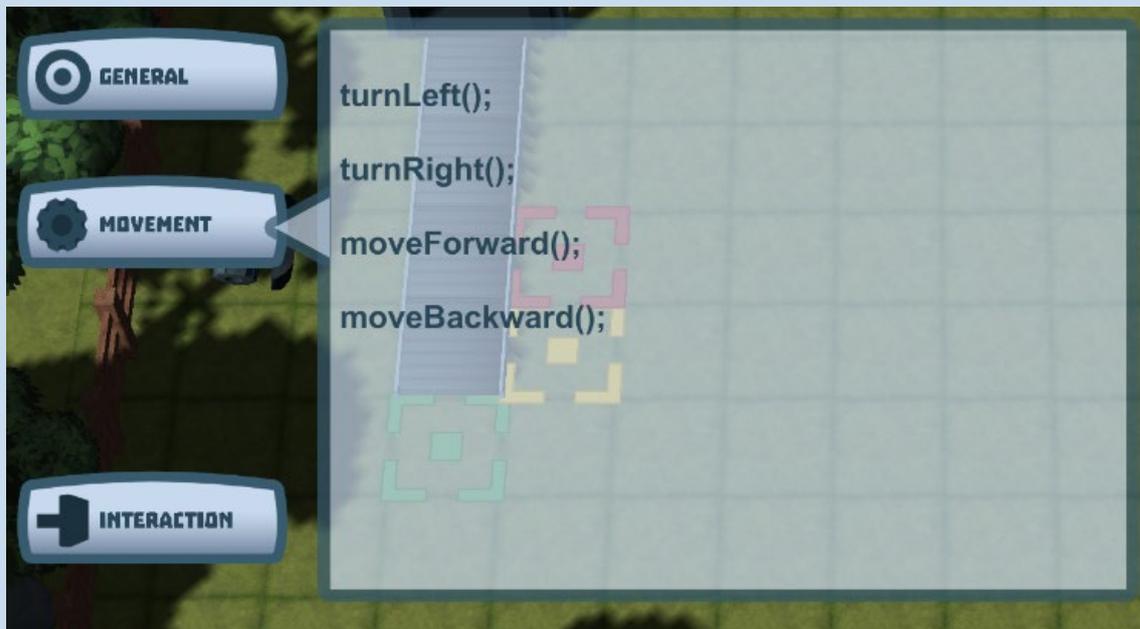
Below we will explain in detail what they are and what they are used for.



The General, Movement and Interaction buttons contain a list of commands that, by combining them, will allow us to control our robot on the map. To do so, we must click on them and they will be added to the command console on the right, where we can see our combinations. If we want all the commands added in this console to be executed, we must click on execute.

- Movement Tab

In this tab we can choose between different commands that will allow us to orientate the robot's view towards a direction and make it move forward or backward.



turnLeft()	<i>Turn the robot to the left</i>
turnRight()	<i>Turn the robot to the right</i>
moveForward()	<i>The robot moves forward one square</i>
moveBackward()	<i>The robot moves back one square</i>

- Interaction Tab

In this tab there is only the push() command;
This will allow the robot to push boxes and other elements in the scene.



- General Tab

In this tab you will find the most complex commands to understand, but that will facilitate and help us greatly to reduce the number of times we repeat a command or a set of them and in turn enhance our creativity in programming.

Wait(); □ Allows us to tell the robot to wait while an action takes place. For example, on level 16, tell it to wait for the conveyor belt to release the boxes.

Loop: for-loop □ Loops are resources widely used in programming, they allow us to repeat a certain number of operations just by writing them once. Example:

```
1 for (int i = 0; i <  i++ ) {  
...   
2 }
```

The first step is to select the number of times we want the loop to repeat, then we enter the corresponding commands we want it to repeat. In this case we would be telling the robot to move forward 3 times.

```
1 for (int i = 0; i <  i++ ) {  
...   
2 }
```



```
1 for (int i = 0; i < 3 i++) {
...
2 }
   turnLeft();
   turnRight();
   moveForward();
   moveBackward();
```

```
1 for (int i = 0; i < 3 i++) {
2   moveForward();
...
3 }
```

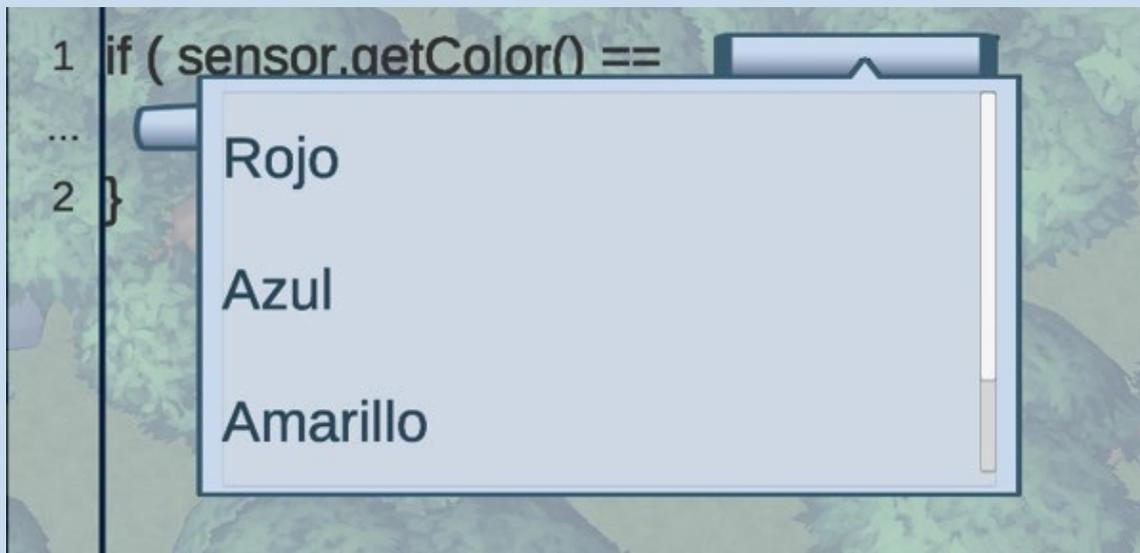
Conditional: if-condition □ Conditionals are very useful to indicate that an action will be carried out only in the circumstances we choose.

```
1 if ( sensor.getColor() == 
...
2 }
```

For example:

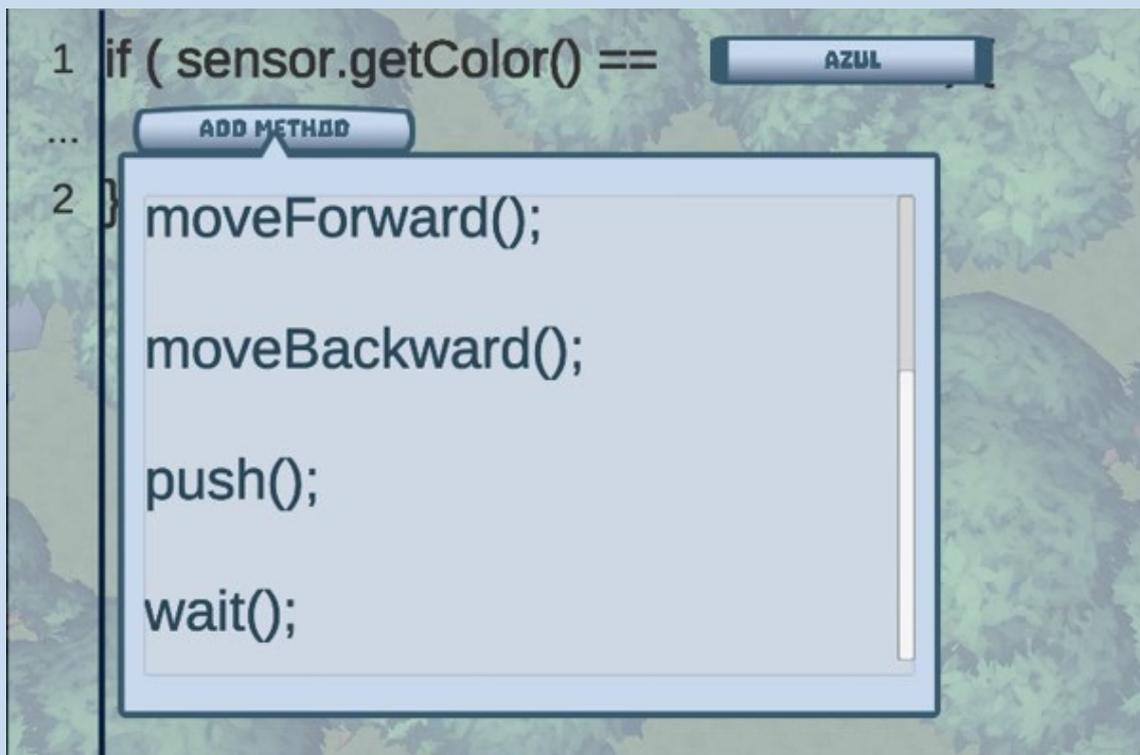
We can instruct the robot that only when it detects a box of the colour **BLUE (AZUL)** it pushes it.

```
1 if ( sensor.getColor() ==  
...  
2 }
```



A screenshot of a code editor showing an if-statement. The code is: `1 if (sensor.getColor() ==` followed by an ellipsis `...` and `2 }`. A dropdown menu is open, showing three options: 'Rojo', 'Azul', and 'Amarillo'.

```
1 if ( sensor.getColor() == AZUL  
...  
2 }  
   ADD METHOD  
   moveForward();  
   moveBackward();  
   push();  
   wait();
```

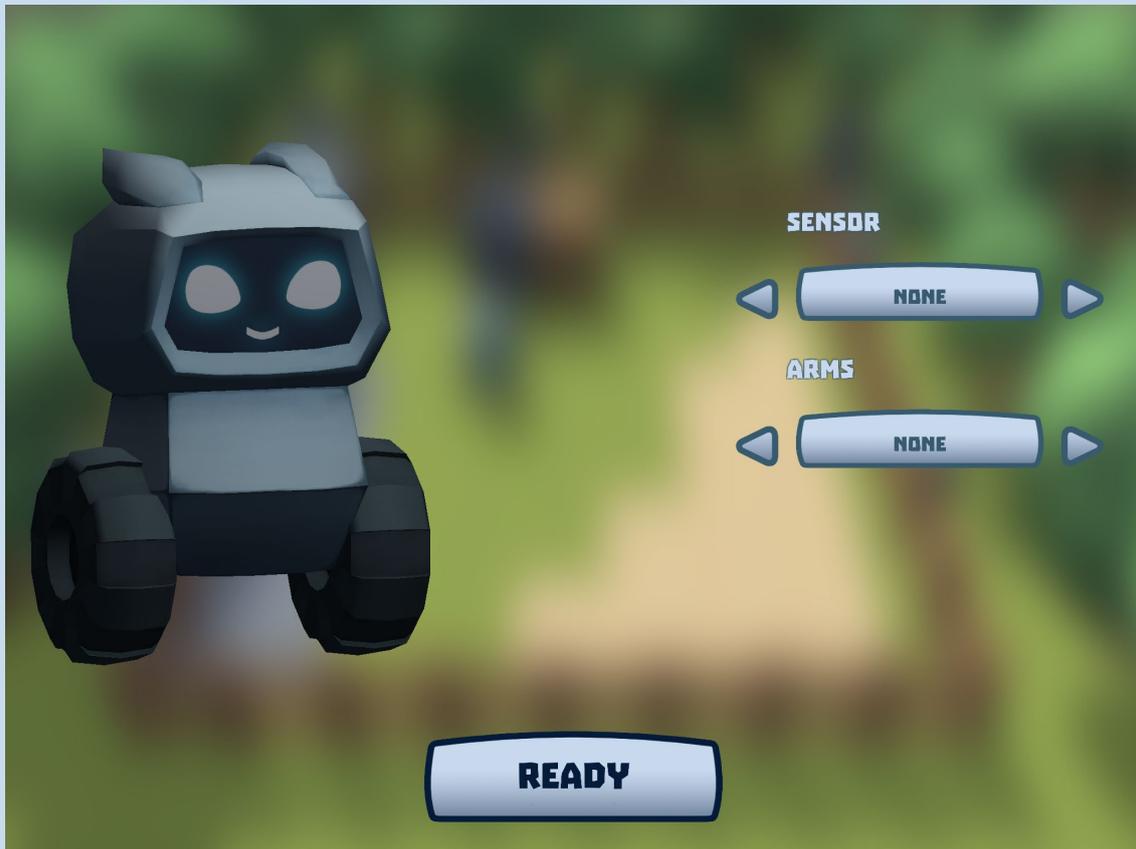


A screenshot of a code editor showing an if-statement. The code is: `1 if (sensor.getColor() ==` followed by a dropdown menu with 'AZUL' selected, an ellipsis `...`, and `2 }`. Below the if-statement, there is a block of code: `ADD METHOD`, `moveForward();`, `moveBackward();`, `push();`, and `wait();`.

```
1 if ( sensor.getColor() == AZUL
2   push();
... ADD METHOD
3 }
```

FIRST LEVEL TUTORIAL

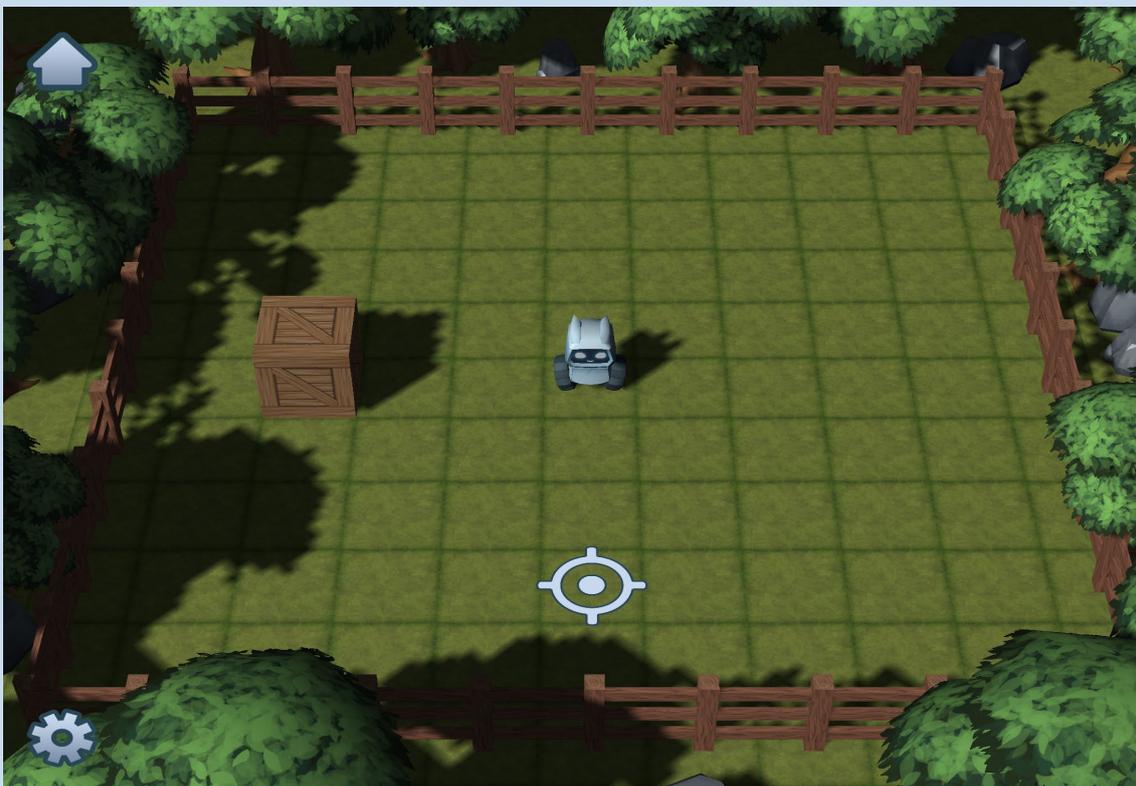
After selecting the first level we will see a customisation screen for the robot:



After pressing READY, the level begins. The doctor will give us a short introduction and welcome.



The level consists of moving the robot to the position marked on the map.



To carry out this process, click on the arrow in the top left-hand corner to access the menu of commands. Once this menu is open, click on movement to see the movement commands, as we want the robot to advance three squares to the indicated position.



We click three times on `moveForward()` to move forward three boxes, and once they are added to the command console on the right, we execute the code by clicking on the corresponding button.



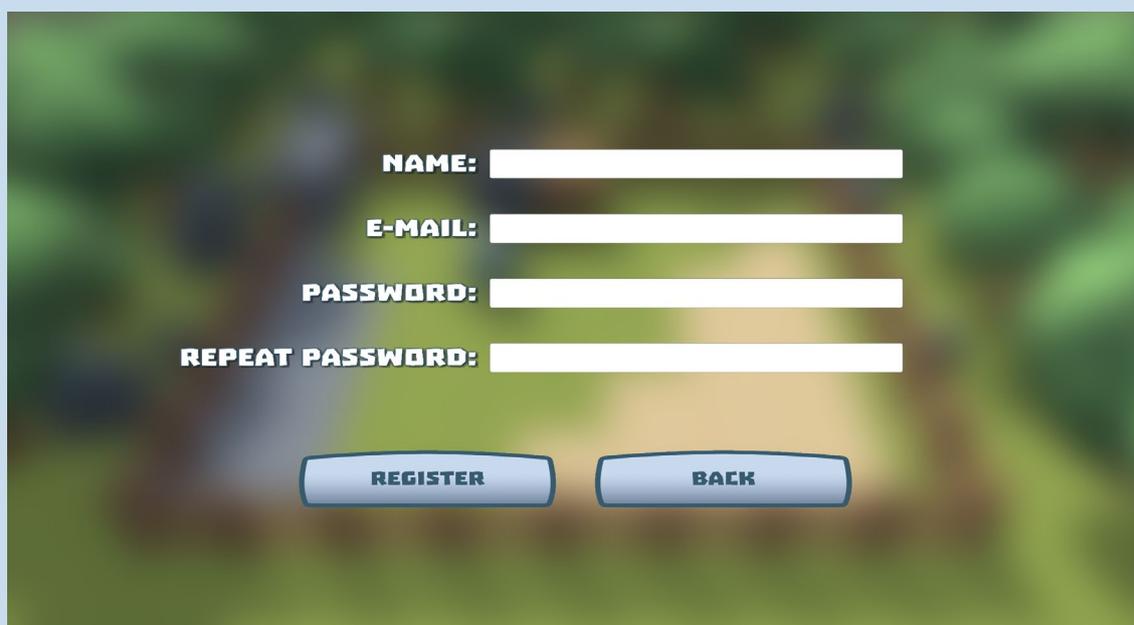
As we can see, the robot moves forward three squares and stops in the correct position.



Congratulations! You have successfully completed your first level in Robot City, we invite you to keep playing and see how much fun programming is.
Keep up the good work!

ROBOT CITY COMMUNITY

Welcome to the Robot City community, if you are a new user you can easily create an account. Simply add your name, email and twice the password to start using this great and connected community.



Registration form interface with a blurred green background. It contains four input fields: NAME, E-MAIL, PASSWORD, and REPEAT PASSWORD. Below the fields are two buttons: REGISTER and BACK.

NAME:

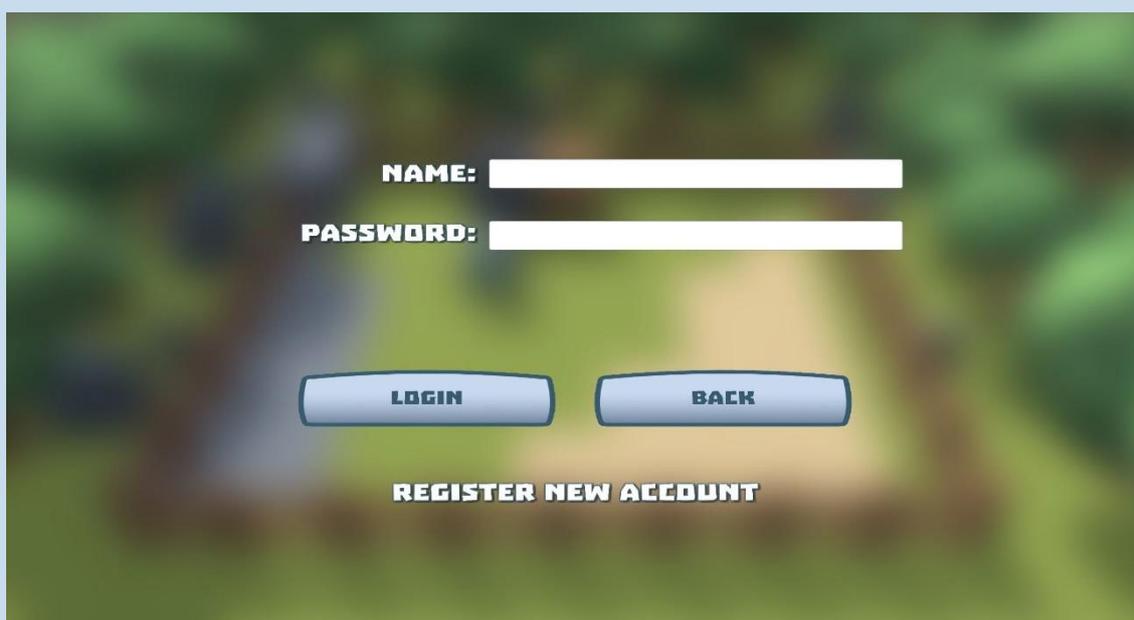
E-MAIL:

PASSWORD:

REPEAT PASSWORD:

REGISTER **BACK**

To log in once registration is complete, simply enter the username and password you registered with and click the login button.



Login form interface with a blurred green background. It contains two input fields: NAME and PASSWORD. Below the fields are two buttons: LOGIN and BACK. At the bottom, there is a link to REGISTER NEW ACCOUNT.

NAME:

PASSWORD:

LOGIN **BACK**

REGISTER NEW ACCOUNT

CREATE AND SHARE YOUR OWN PROJECTS

After accessing the community we can see a map with properties.

Some of them have owners and others can be claimed for free. Once claimed, the properties can be customised to your liking.

Here you can share your own projects. You can write your own robotics project or you can also share a link to a project you have created on another website, such as www.instructables.com or your school website.

Robot city is a multilingual game, with an international community, so you can create your projects in Spanish, Polish, German, Greek, Portuguese or English. A flag will appear on your property depending on the language you use. We advise you to write it in English to reach more users.

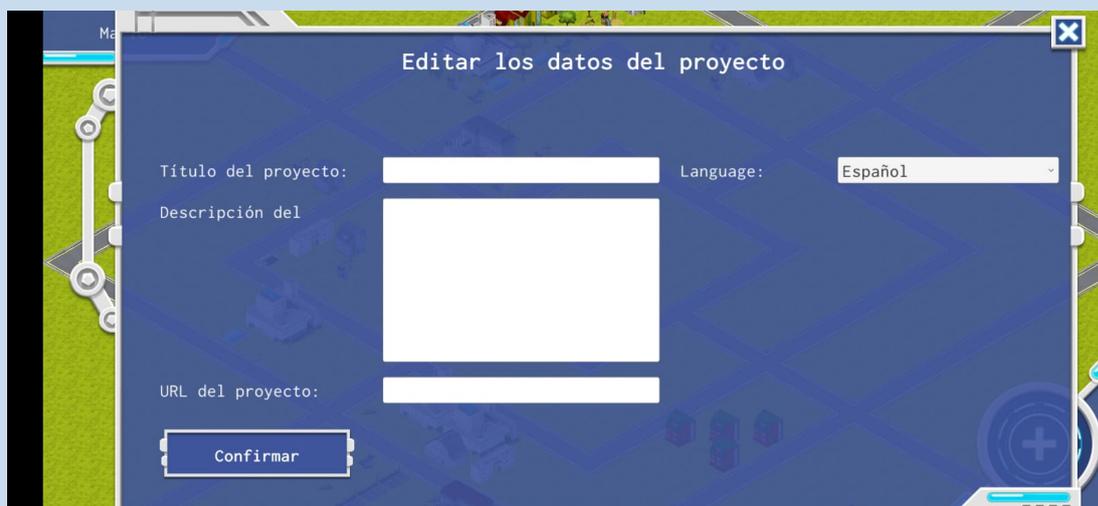


These properties are projects created by the users of the community, we can also create our own project and share it. You can appreciate the projects of other users by clicking "like" on the selected project. We encourage you to visit the projects of other users and click  on the ones you like the most.

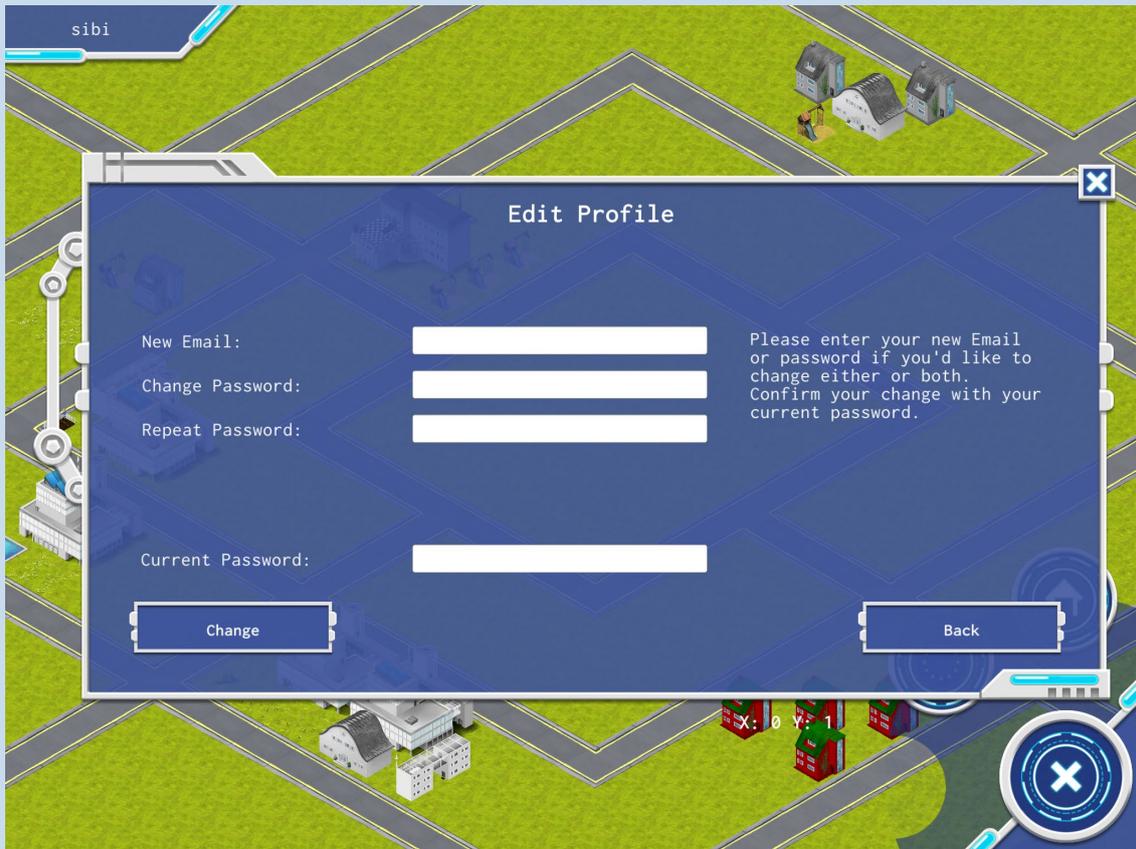
Project created by another user:



Own project:



We can also change our user information:



We encourage you to use the community to publish your robotics projects and also to learn about other people's projects around the world and to grow your holdings with new projects.