Globisens

xploris

CODING

Distance indicator

xploris

CODING

Globisens

DISTANCE INDICATOR

1. Introduction
2. Activity setup
3. Coding
4. Questions
5. Activity summary

1 Introduction

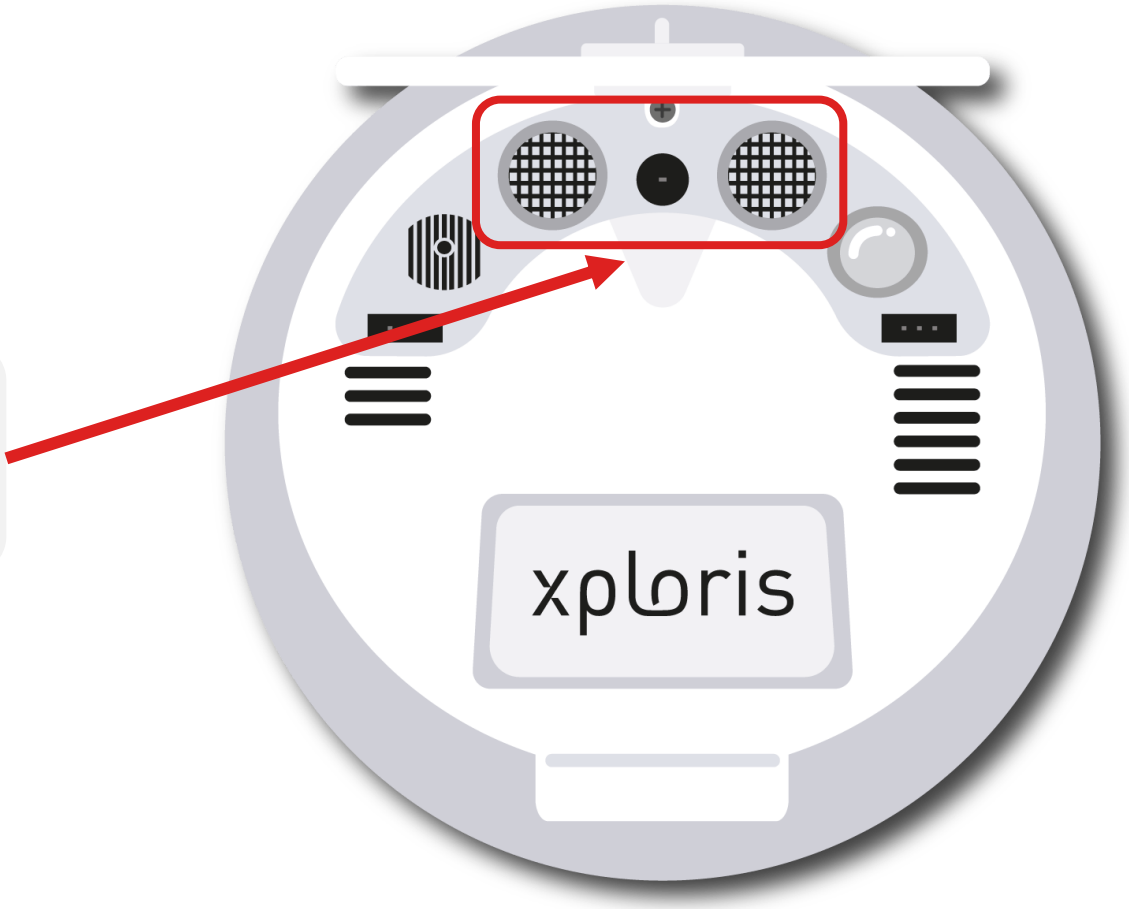
Imagine walking through a mysterious forest shrouded in fog. The path ahead is unclear, but then, you gain a superpower: a special sensor that lets you know how close or far away trees and rocks are. With this power, you can navigate safely and dodge obstacles!

In the real world, many robots and devices use distance sensors to measure how far away objects are. Autonomous cars use them to avoid crashes, robot explorers take them to unknown planets, and even phones use them to know when we bring our hand close to the screen.

In this chapter, you will learn how to program and make use of the distance sensor in Xploris. We will see how to detect if an object is approaching or moving away, and how to respond with sounds and images. With this knowledge, you will be able to create smarter robots that understand the world around them, teaching Xploris to react to the world around it and making you a true programmer.

2 Activity setup

The “distance” sensor is located on the back of the Xploris, make sure it is uncovered as shown in the picture.

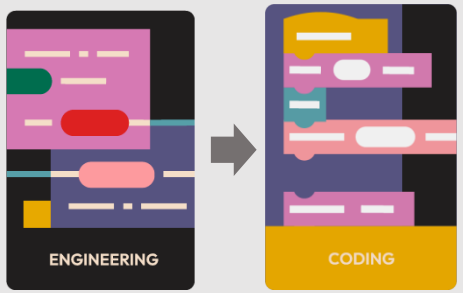


2 Activity Setup

Turn on your Xploris and connect it to your computer or tablet.

Open the XploriLab software on your computer or tablet.

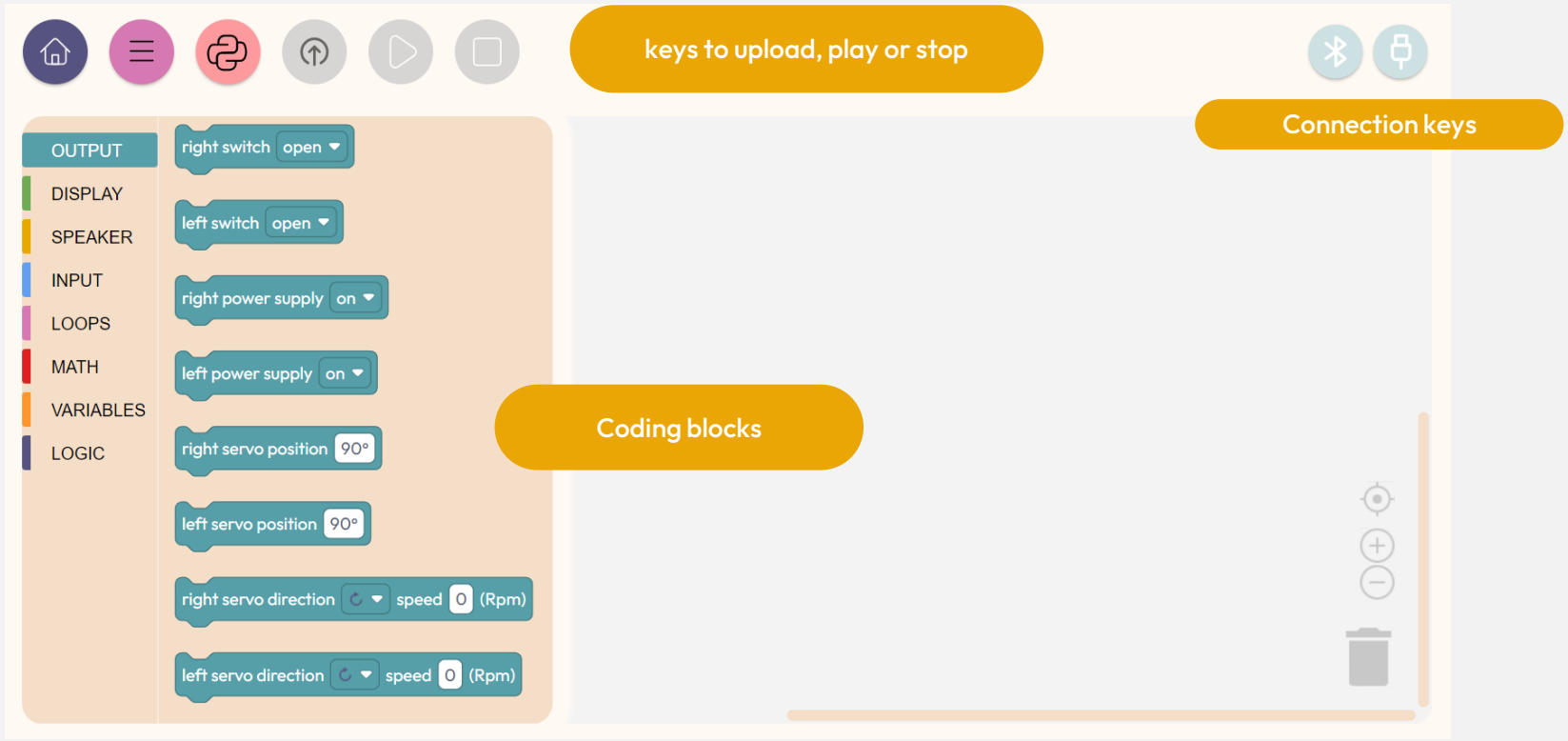
Once inside XploriLab, select the icon to connect the device via cable or bluetooth as applicable.



Go to the ENGINEERING section and then to CODING.

3 Coding

In the Coding window, you will find the tools you need to create code using blocks.



3 Coding

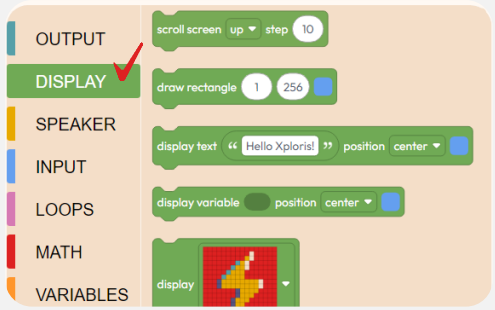
The available tools represent blocks that allow you to perform various actions.

Functions

OUTPUT	Output blocks to activate Xploris switches, power source and/or servos.
DISPLAY	Blocks to control the Xploris screen.
SPEAKER	Blocks to control the Xploris speaker: play sound tracks, notes and control the speaker volume.
INPUT	Blocks that enables you to use all Xploris keys and sensors, such as temperature, light, distance, sound and voltages.
LOOPS	Loop blocks to perform an action continuously or as long as their conditions are met.
MATH	Mathematical blocks, such as +,- and many other math functions.
VARIABLES	Blocks for creating variables, assign and replace their values.
LOGIC	Logic operators that will allow decisions to be made based on the state of the data.

3 Coding

1



In this activity, you'll learn how to program Xploris to detect whether an object is getting closer or moving away. You'll use programming blocks, which work like LEGO bricks: each block has a specific function, and by combining them, you'll create the instructions that tell Xploris what to do.

The goal is to make Xploris display an arrow on your screen and emit an audible alert when an object approaches, moves away or stays still. You will start by exploring the DISPLAY function.

2

From DISPLAY bring the clear screen block to your workspace to clear the device screen, ensuring that you have a clean space to display the results of your programming.

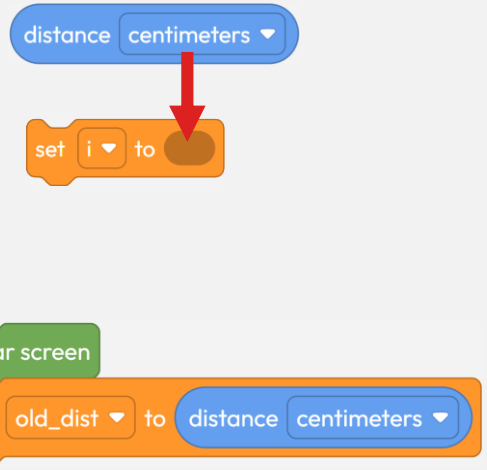
3 Coding

3

Go to the VARIABLES group and click on Create variable. There you can create a new variable that we will use later. Name it “old_dist”, which stands for “old distance”.

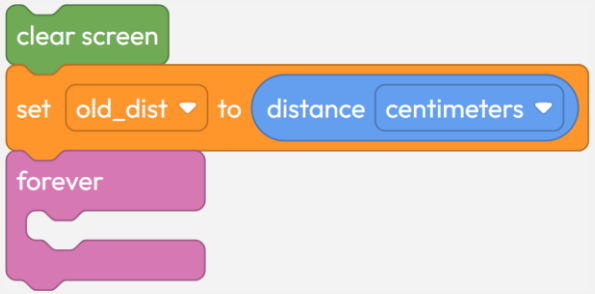
Then, find the set i to block and drag it to your workspace. Next, go to the INPUT group and find the distance centimeters block.

Now, in the set i to block, change the variable “i” to “old_dist”. To the right of this block, place the distance centimeters block, and place it next to the work you have done.



3 Coding

4



In the LOOPS group, use the 

block below your current work. This will let you repeat the instructions inside it endlessly.

5

Now, from the LOGIC group you will use a **conditional block** that will help your program to **make decisions** according to certain conditions, without the need to use many blocks. Imagine that you are creating a game where a character changes its speed according to the amount of life it has:

- If it has 100 or more life, it runs very fast.
- If it has between 50 and 99 life, it slows down slightly.
- If it has less than 50 life, it walks slowly.

This block allows the program to decide what to do according to the character's life, without having to program each case separately.

3 Coding

6

The time has come to create the different cases and conditions and indicate what will be done in each of them.

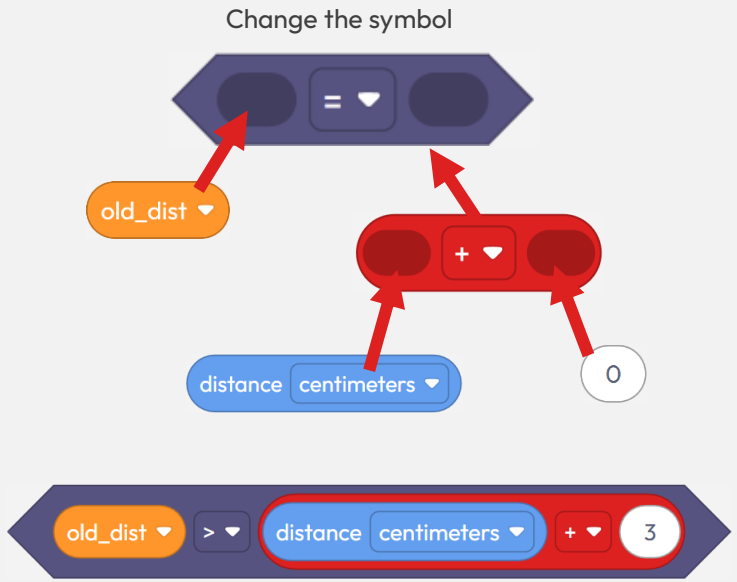
- In the first case, **you'll be able to detect when an object comes within 3 centimeters** of the distance initially measured by your Xploris.
- In a second case, **you'll be able to detect if an object moves more than 3 centimeters** away from the device.
- And in a third case, **you'll be able to detect if an object has not moved more than 3 centimetersin any direction f**rom the device.

3 Coding

7

For the first case, you'll use the distance measured by the device and the value stored in **"old_dist."** To determine if an object is approaching. Here's how you can do it:

From the LOGIC group, drag the = block, and from VARIABLES drag the old_dist variable and place it to the left of the previous block. Choose the "greater than" operator. Next, go to MATH and drag the + block, placing a numeric block 0 to the right. Assign the value "3" to this block, then add the sum operator. Finally, drag the distance centimeters block from INPUT, and place it to the left of the incrementer block, then attach this block to the right side of the comparator block.

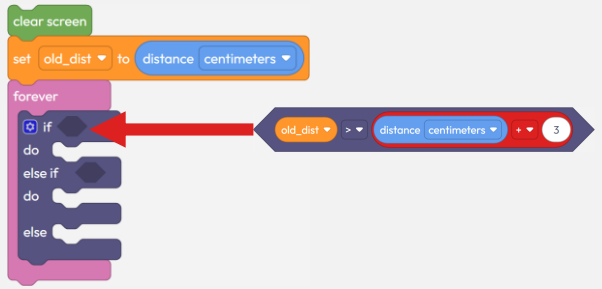


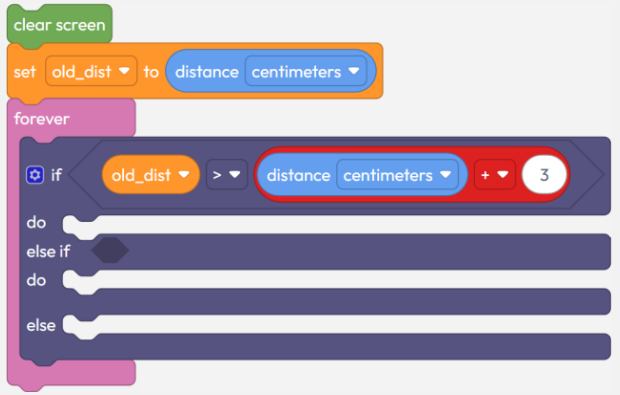
3 Coding

8

Take the comparator block you made earlier and drag it into the conditional block, just to the right of **“if”.**

Now, it’s time to tell your program what to do when the condition is met - that is, when the previously saved distance is greater than the current distance plus 3 centimeters. **This means the object was farther away before and is now closer**





3 Coding

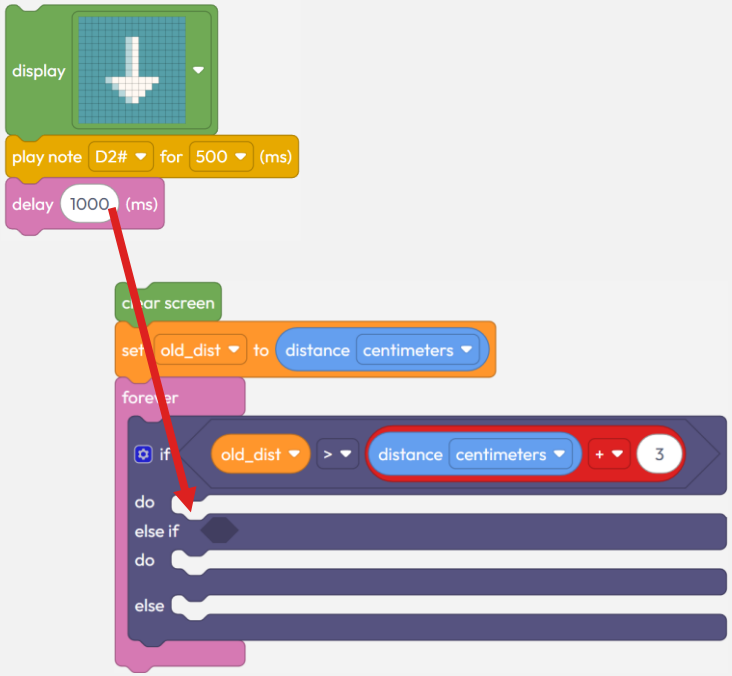
9

You're starting to create something great! To continue, drag the following to your workspace:

- First, from DISPLAY , the 

block and select the downward pointing arrow.
- Then, from SPEAKER , the play note C for 125 (ms) block and select the “D2#” note and make sure it lasts “500 ms” or milliseconds.
- Finally, from LOOPS use the delay 100 (ms) block with a value of “1000 ms. Place it inside the conditional block, in its first “do” function andafter he Play note block .

The downward arrow will indicate that the object is getting closer!



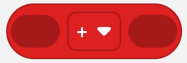
3 Coding

10

It is time to go on and indicate what you want the conditional block to do next in case the first condition has not been met. For this, you will follow very similar steps to the first condition, with some variations:

- Drag a 

block and the 

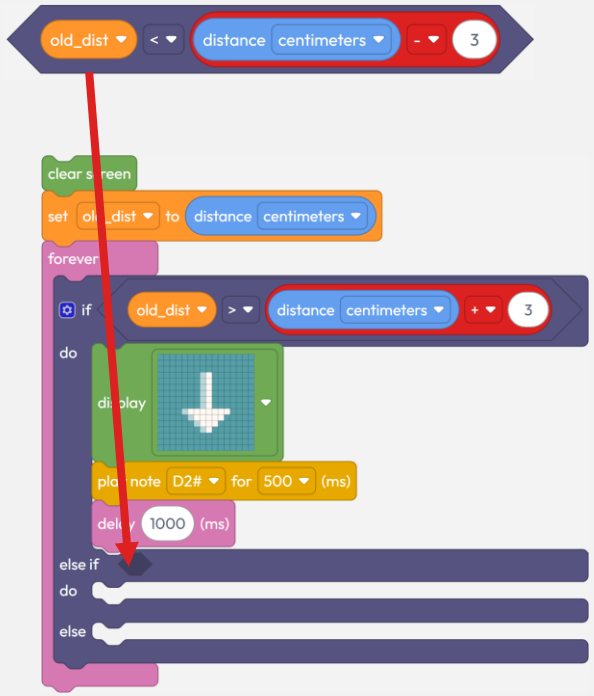
variable block and place it to the left of the previous block and this time, **choose the "less than" symbol**.
- Drag a 

block and place to its right a numeric 

block, assigning it the value "3" and the **choose subtraction operator**.
- Then, drag a 

block and place it to the left of the incrementer block. This last block, drag it to the right of the comparator block.

Enter it in the conditional block, next to "else if".



3 Coding

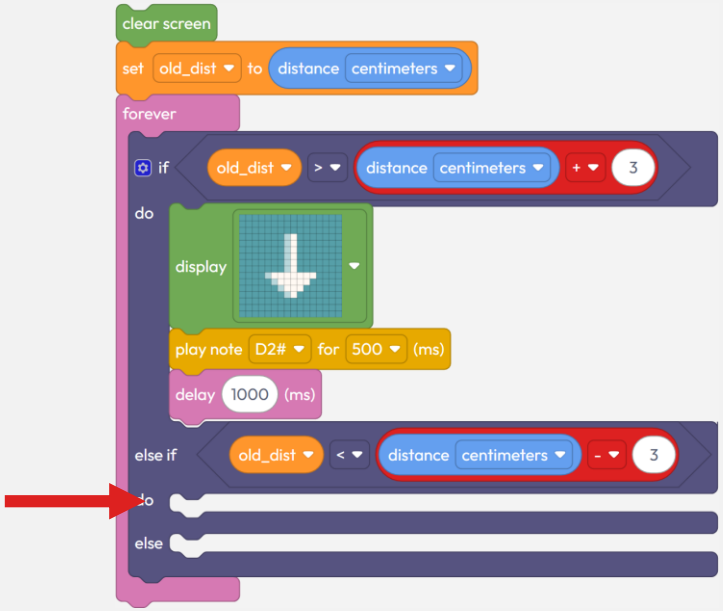
11

Perfect! Now you are going to tell the program what to do if:

- The first **condition is not met**: that is, the object does not come close enough!
- The second **condition is met**: that is, the object is moved away by more than 3 centimeters!

You will use blocks similar to those of the past, but with a twist:

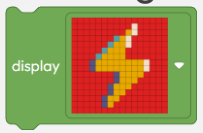
- The arrow on the screen will point upwards: So you know the object is moving away!
- The musical note will be different: To make it sound different and more fun!



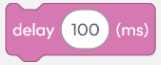
3 Coding

12

Tell he program what to do if the second **condition is met**, by dragging the following into your workspace:

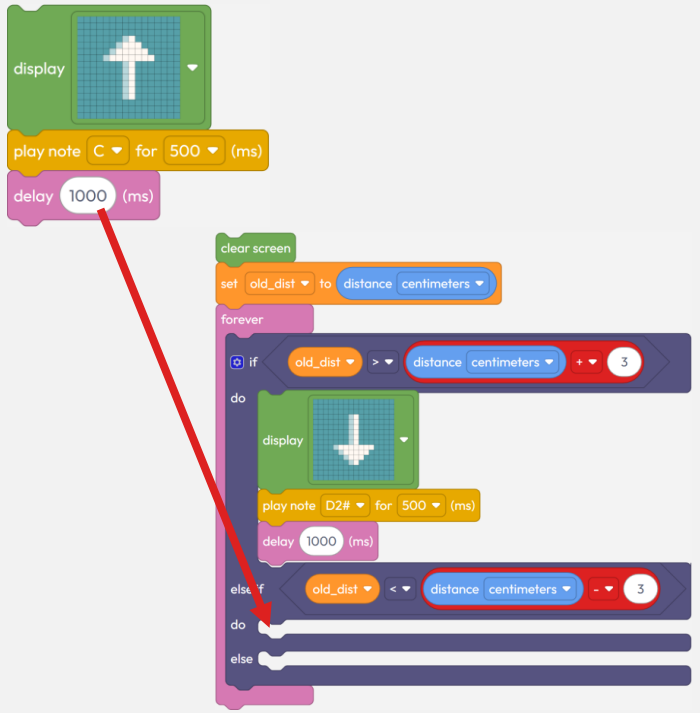
- First, the 

block and **choose the arrow pointing up.**
- Then, the 

block and **select the note "C"** and make sure it lasts "500 ms".
- Finally, the 

block and assign it "1000 ms".

Then, place these blocks inside the conditional block, in its second "do" function, following the same order. You can also join them before and move them all together by dragging the first block, which will make it easier!

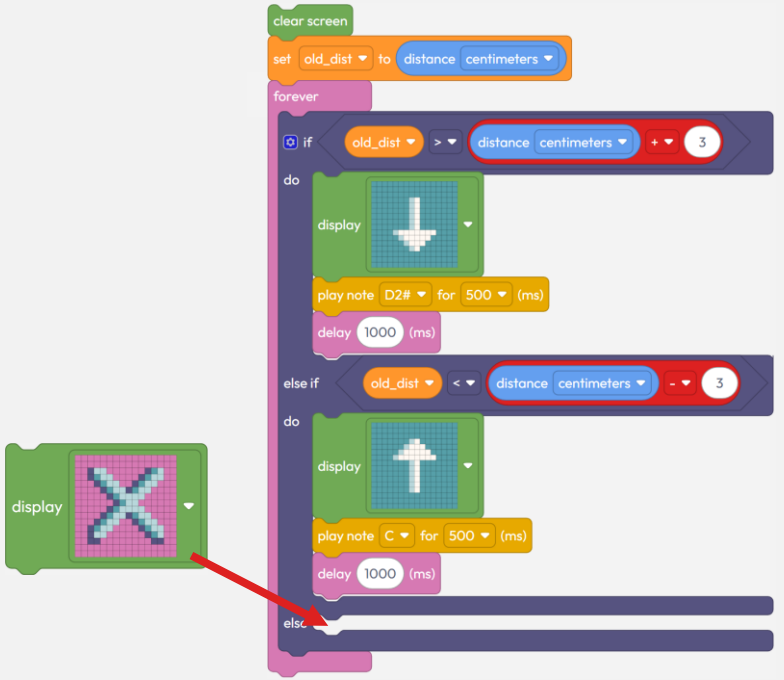


3 Coding

13

You're almost there! You've already programmed two out of the three actions that Xploris will perform.

- The third action will detect if the movement of an **object is less than 3 centimeters**.
- To do this, go to the DISPLAY group and drag the  block and choose the X image.
- With this X, you will indicate that there is no significant movement, without emitting sound. Place this block at the end of the conditional block, just to the right of “else”.

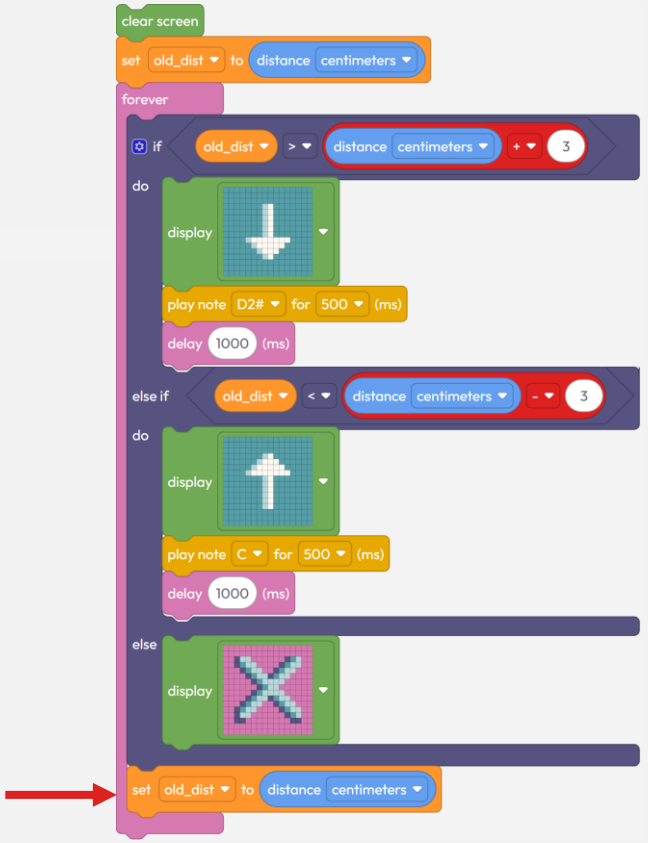


3 Coding

14

After the great work you have done so far, it is time to tell the program that, after making all the checks, it **must re-measure the distance** and saveit current value. To do so: drag the set i to block, and the distance centimeters block to your workspace. Select the **“old_dist”** variable in the drop-down menu and to the right, place the distance centimeters block.Place the Set lock just after the conditional block.

Thanks to your hard work, your new detector is all set and ready to go!



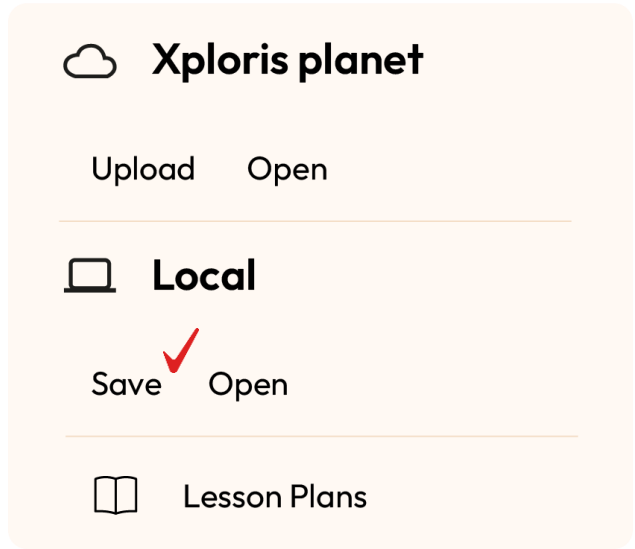
3 Coding

To make sure that the program works correctly, we will follow these final steps:

Press the three-bar icon at the top and select the “Save” option. Then, assign a name and save our program.

Press the “Upload” button in the Xplorilab interface. This will transfer the program to the Xploris device.

Once the program is loaded, press the “Play” button on Xplorilab software. it's time to play and experiment with your new detector!



4 Questions

1

Sciences

How do you think the detector knows if an object is closer or farther away?

2

Art

What kind of sound do you think would be helpful to alert you when an object is approaching? What fun sounds could you create to make the detection even more exciting?

3

Let's keep experimenting!

How can you make the detector more accurate and what would happen if you put the Xploris device in front of a mirror?

5 Activity summary

- We used the Xploris display and its “distance” sensor to detect the movement of objects.
- We used block programming with loops, conditions and variables to detect differences in the distance measured by Xploris.
- We programmed using the XploriLab application, explored changes made in the code and loaded the created program to our Xploris.

Globisens

xploris

CODING

Distance indicator