Globisens

xploris

CODING

Chessboard

xploris

CODING

Globisens

CHESSBOARD

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

1 Introduction

Repetitions, or loops, are like a secret weapon in programming. They let us perform the same task multiple times without needing to write it out repeatedly. It's kind of like making a necklace with colorful beads, where you simply repeat the same pattern over and over until it's complete!

In block languages, repetitions are very useful to build things faster and tidier. Imagine you want to make a path of bricks in a video game. Instead of placing each brick one by one, you can use a loop to have the program place them for you. That way, if you need a longer or shorter path, you just change a number and you're done.

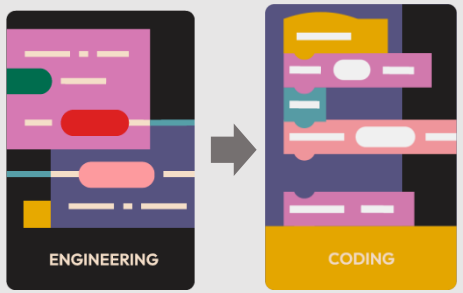
Now, you will become a chess programmer. You will use block language and the Xploris device screen to build your own chessboard.

2 Activity setup

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

Open the XploriLab software on your computer or tablet.

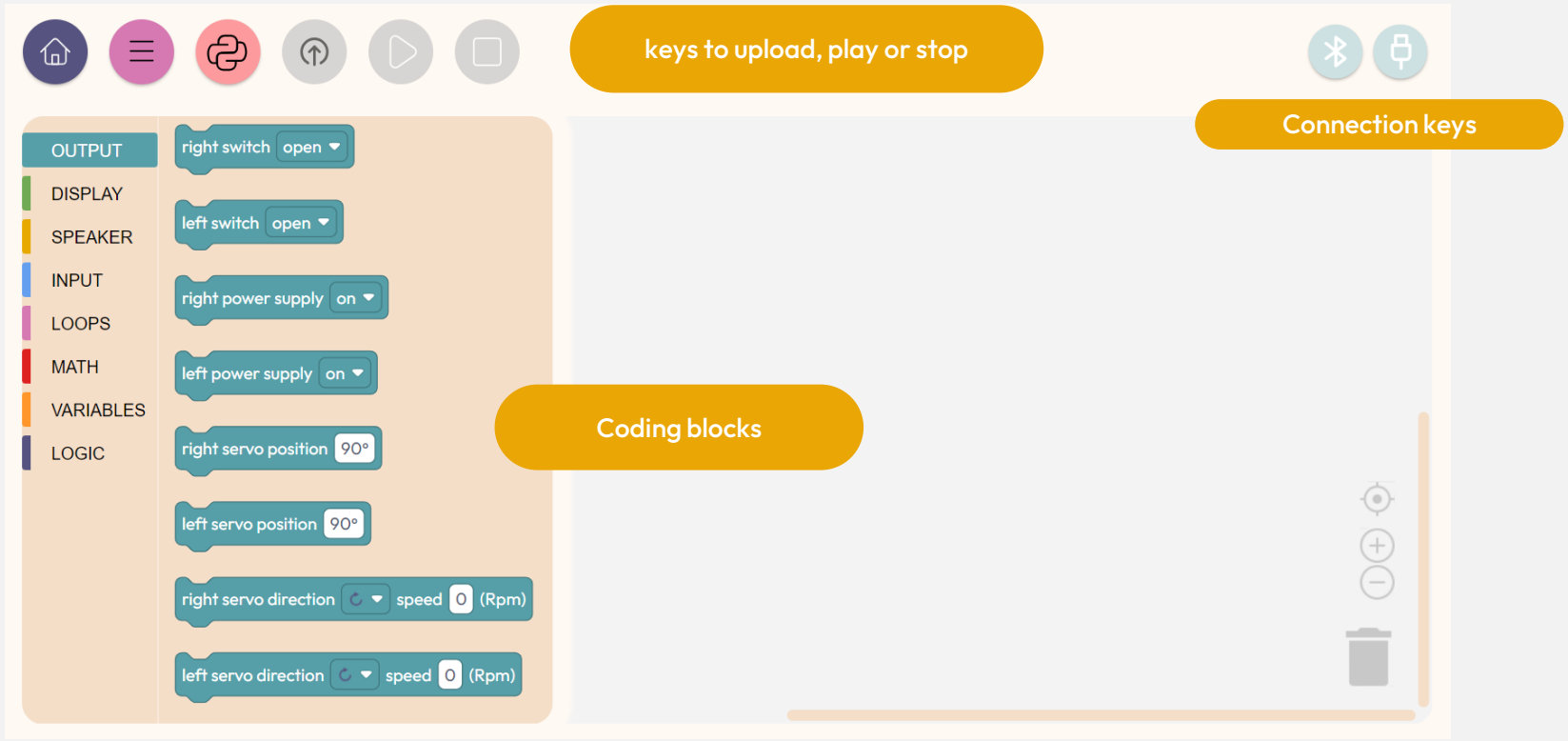
Once inside XploriLab, select the icon to connect the device via USB cable or bluetooth as applicable.



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

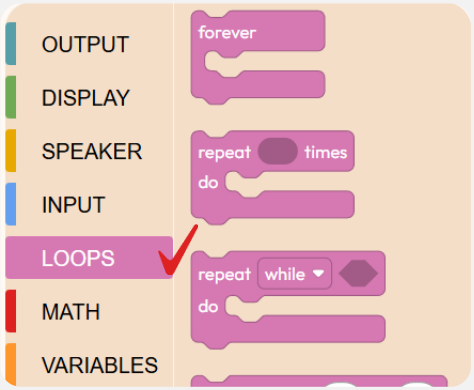
The available tools represent blocks that allow you to perform various actions.

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.

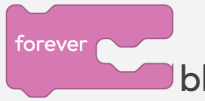
3 Coding

1



In this activity, you will learn how to program a chessboard on the screen of your Xploris device. You will use programming with blocks, which represent programming instructions. The goal is to display a chessboard. To achieve this, you will begin by exploring the functions of LOOPS .

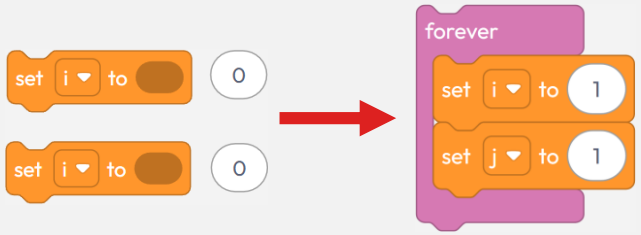
2

From LOOPS , drag the 

block to your workspace to **repeat indefinitely** the instructions you place inside it and see how the program runs on your Xploris device.

3 Coding

3



In the VARIABLES group you will use two set i to blocks, and from MATH, you will need two numeric 0 blocks. Drag them to your workspace. Now you will set two variables to be used: **"i"** and **"j"**, both with value **"1".** Place them inside the forever block!

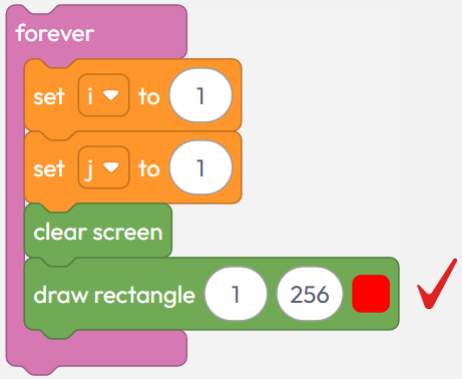
4

Use the clear screen block in the DISPLAY group to clear the screen. You will then have an empty space where you can see the results of your programming without any problems.

Now prepare the background of the chessboard: find the block draw rectangle 1 256 in DISPLAY and add it to your workspace.

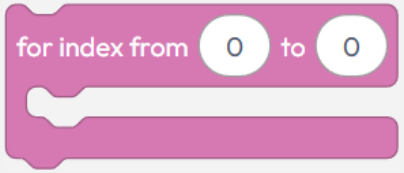
3 Coding

5



Place the blocks in the same order, inside the work inside the  **block** and set the start position of the rectangle to **“1”** , the end position to **“256”** and choose the **color red** so that the background has a good contrast and is easier to see.

6

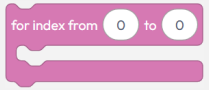


Now, let’s dive into LOOPS ! To really grasp them, think about something you have to do repeatedly, like jumping rope. Instead of saying "jump" over and over, you can give one simple command, and it will repeat itself automatically.

The loops will help you paint and scroll through the Xploris device screen without having to write each step separately, making programming easier and more fun!

3 Coding

7

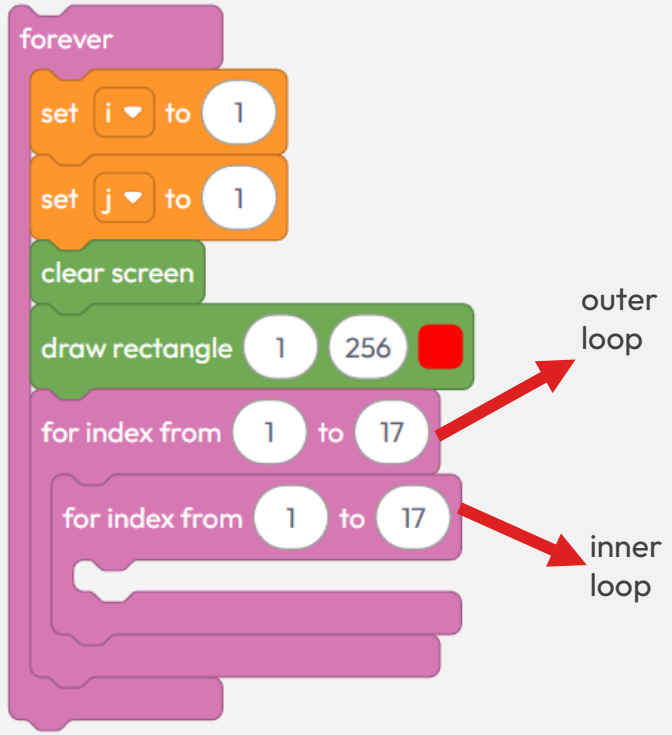
Now you will use two 

loops, which together are called nested loops. This means that one loop will be inside another, like when you put a small box inside a larger one.

- With these loops, you will repeat the instructions inside. Place the first loop (outer loop) underneath the work you did before.

- Then, inside that loop (as if it were a box inside another one), place the second loop (inner loop) and tell them both to artwith **1"** and endwith **"17".**

This way you will be able to repeat actionson all 16 columns and 16 rows of the Xploris display.



3 Coding

8

Now you will create the rulers to draw the chessboard pattern.

- To start: drag the block 

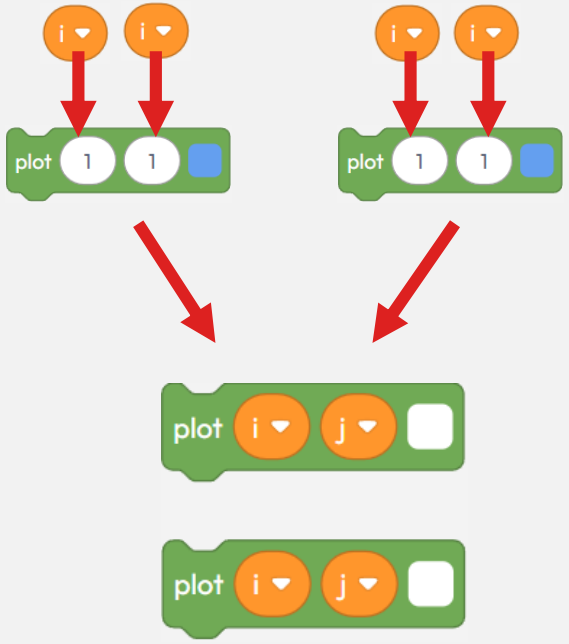
twice from DISPLAY .
- Then, from VARIABLES drag four 

blocks.

Now, you will use the instruction to draw (or "plot") different points on the screen. For that, you must take into account the location of **"i"** and **"j"** and the **color**. In both 

: fill in the spaces with the four variables 

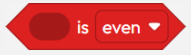
. In the first space of both blocks, use the variable **"i"** and in the second space, use the variable **"j"** and choose the **color white**.

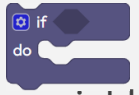
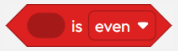
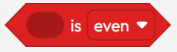


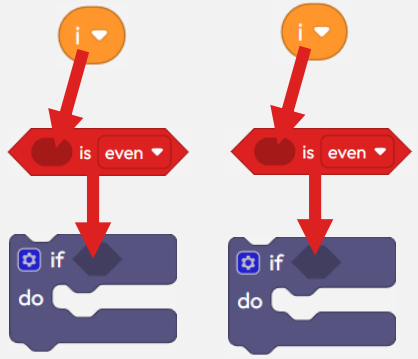
3 Coding

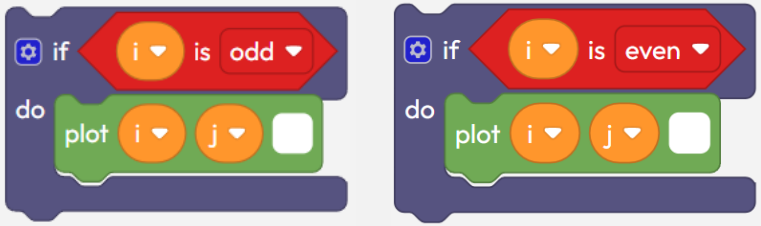
9

Keep building the board following a special rule that **“recognize if a number is odd or even ”**:

Drag two  blocks from LOGIC . Then, from MATH , get two  blocks. You will also need two  variables.

Together, the  and  blocks will search for an odd or even variable: fill this last block with the variable  and on the right side of the first one select the option “odd” and in the second  block the option “even” next to the variable “i”. Now it is time to join it to the work done previously: drag both  blocks into  .





3 Coding

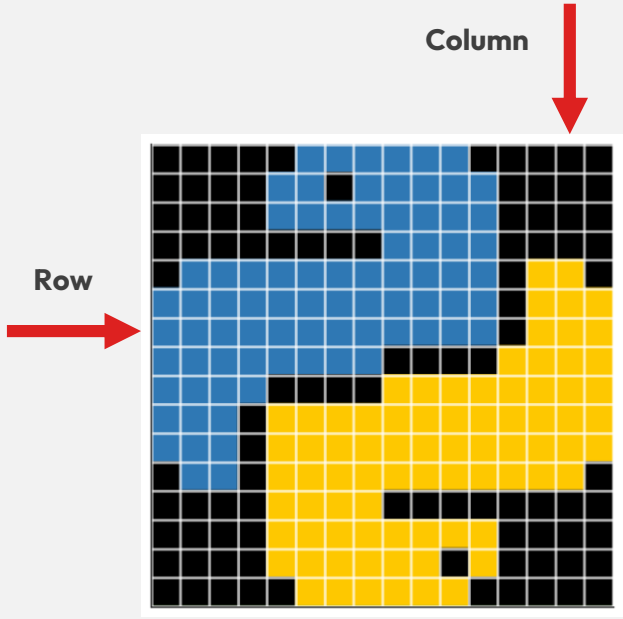
10

Imagine that your board is drawn one column row at a time. But also Thus, to make it complete, you need to move to the next row and keep drawing.

So far, your program can already go through each column, checking whether the pixels are odd or even with the variable i .

Now it is time to do the same with the variable j !

This will help you move on to the next rows and combine all the work so that the board looks perfect.



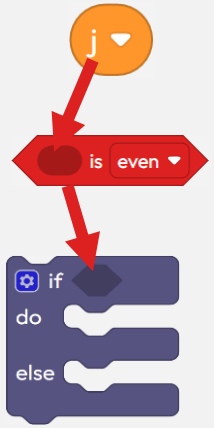
3 Coding

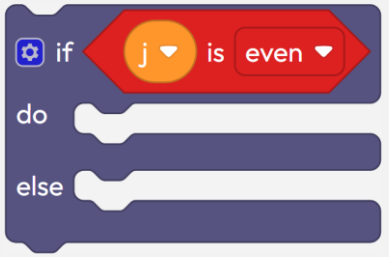
11

Time to go through the j rows! Thus please , drag the if / do / else block from LOGIC group.

This block is similar to the others, but if the condition isn't met, it performs a different action, giving you control over what happens in each case!

Now, move the is even block and a numeric j block to your workspace. Check that it is the variable “j” that you are adding to the is even block and select the **“even”** option. Then, join it with the if / do / else block next to **“if”.**





3 Coding

11

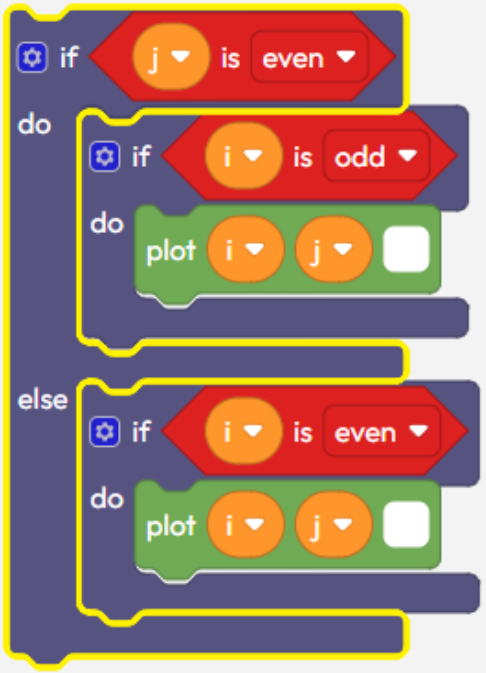
Now, you are going to join the previously made structures to form a nested conditional structure.

- First, drag the first conditional block inside 

, next to “do”.

- Then add the second conditional block next to **“else”.**

In this way, your program will decide when and where to draw dots on the screen, using the variables **“i” and “j”.**



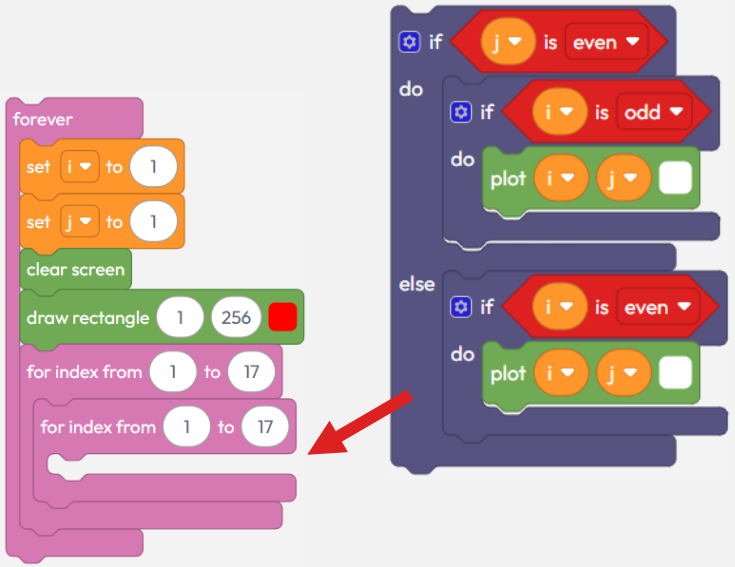
3 Coding

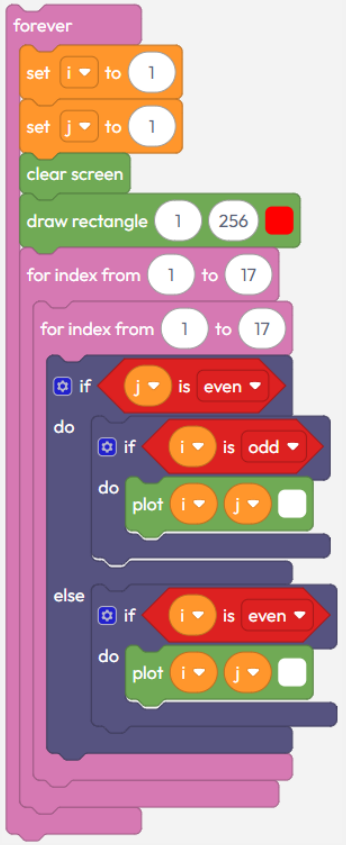
12

You are one step closer to finishing your board!

Now, drag the nested conditional structure you have just created and place it inside the nested loops you made at the beginning.

With this step, your board will take shape and will be ready very soon!



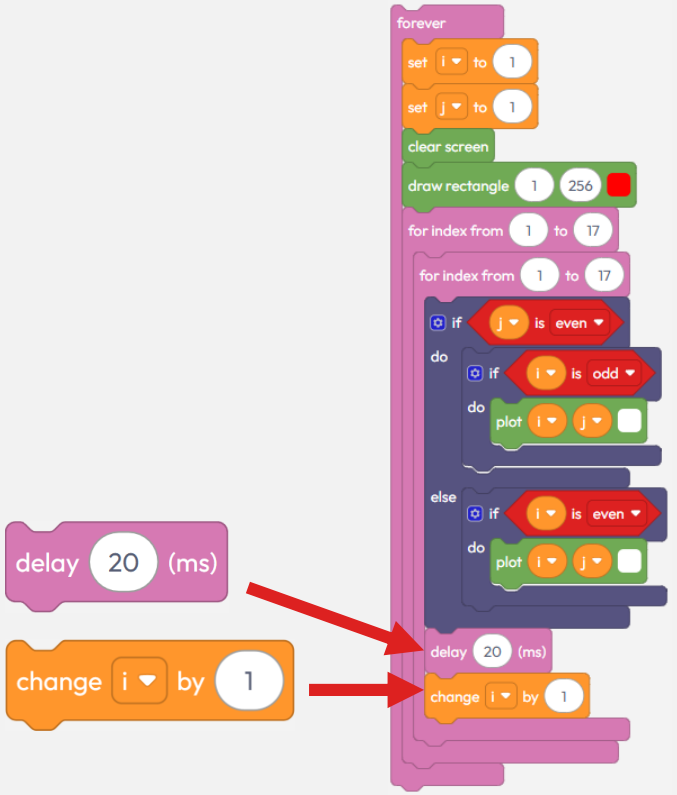


3 Coding

12

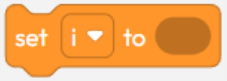
You're almost done with your board! To see how your board is built step by step, add a waiting time between each action, called a **"delay“.** You also need to add an **incrementer** to the variables "i" and "j" so that the loops can advance correctly from the start (1) to the end (17).

At the end of the **inner loop**, add the delay 100 (ms) block from the LOOPS group and set it to **“20” (ms).** Then drag the change i by block incrementer from VARIABLES and place a numeric 0 block inside it. Select the variable **“i”** and enter the value **“1”.** This will make the **“i”** column advance until it reaches 17.



3 Coding

13

At the end of the outer loop, add the 

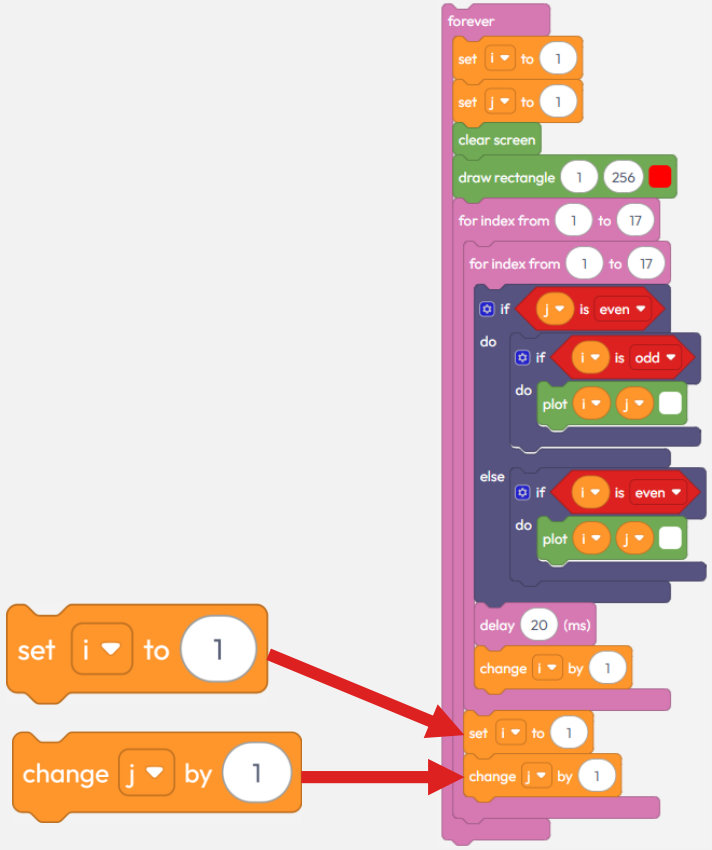
block and set the variable to “i” with the value 1, so that once the inner loop has been executed 17 times, it will execute the inner loop again after changing the value of “i” to “1” and going through the columns again.

Also, add another 

incrementer for the variable “j” together with a 

numerical block with the value “1” so that it advances from row to row, until it reaches 17.

With these last steps, your board is ready !



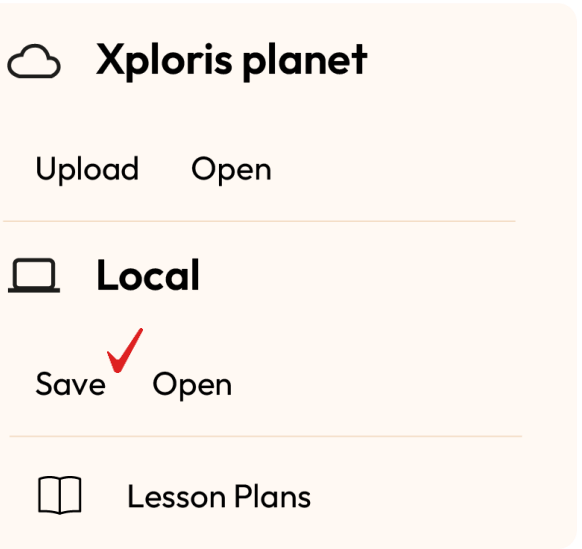
3 Coding

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. Enjoy the result of your efforts: an incredible chessboard. Keep creating and having fun with technology!



4 Questions

1

Sciences

What would happen if this program were run on a screen of different sizes? How could we make it adapt to different resolutions?

2

Art

If you wanted to create a digital work of art with this code, what colors would you change to make it more visually interesting?

3

Let's keep experimenting!

What would happen if we reduced the range in milliseconds (ms) of the “delay” instruction? And if you changed the “odd” instruction to “even” instruction in the blocks? how would the output on the Xploris screen will change?

5 Activity summary

- We used the Xploris screen to paint its pixels and draw a pattern.
- We use block programming making use of loops, conditions and incrementers to create the pattern of a chessboard.
- We created a code using the XploriLab app, explored the changes made to the code, and then loaded the program to test it with Xploris.

Globisens

xploris

CODING

Chessboard