This is an activity that integrates a simple visual programming language (Blockly) with a group of commands or basic instructions that we can combine by fitting pieces which will later allow the red circle on the board to perform certain actions or we can also perform some tasks using geometric shapes.

BLOCKLY - MOVEMENT

The activity consists of making sure that the red circle reaches exactly the position of the white one by creating a program.
It is very useful to get students started in the basic concepts of programming and computational thinking in an intuitive way without having to write code or worrying about if there is any missing character, for example.
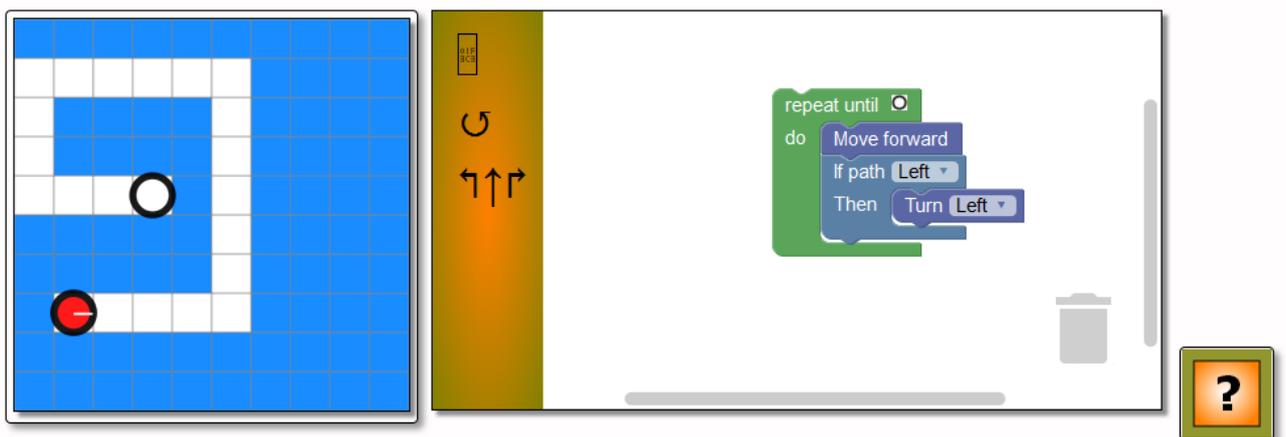
José M, Boo - 2018

TAB 1.- ACTIVITY

a) **Left zone**.

In the left zone we find the following elements:

1.- **A 10 X 10 grid** where the circle will move when the orders are run. The box with the arrow indicates the initial position of the circle that will move when the verification key is hit. The box with the white circle shows the final destination for the red circle. The white boxes indicate the path that the red circle can follow to reach the target. The squares with the "X" are fillers and indicate that the red circle can never move through them (like a wall).

When we start the edition, the 100 boxes are blank (the red circle could move freely throughout). To specify the start, destination or to leave out boxes, you should double-click on the appropriate box and the pop-up window shown in the image on the right will be displayed:
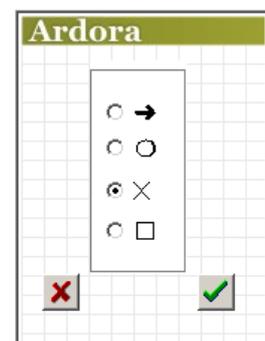
If you wish, you can choose the appropriate option and accept by clicking on the green button. If you had previously established the "start" and/or "destination" boxes in the grid and you choose one of these options again, the original "start" or "destination" boxes will be removed and replaced by the latest option (there can only be one "start" and one "destination").

2.- These buttons with the green arrows allow us to save time; whenever we press them, all the boxes belonging to the respective row or column will be filled with an "X". If there was already an "X" in any of these boxes, it would change to white (path).

3.- With this button we will select the colour of the boxes containing the "X" (wall) once the activity is displayed on the page. Just under this button we can indicate if we want the square that surrounds the grid to have rounded corners or not.

4.- In this field we will indicate the size of the side of the squares that make up the boxes. Therefore, by increasing or decreasing this value, the total size of the grid will either increase or decrease, occupying more or less space on the web page.

5.- Here we can increase or decrease the thickness of the lines forming the grid.

6.- These option buttons allow you to choose the type of lines the grid will have: continuous, dotted...

7.- With these option buttons we will indicate the direction of the red circle to start its movement. For example, if it were oriented towards a "wall" (box with an " X "), the first order that the user would have to give would be to turn it so that it is pointing to the path (white boxes).

b) **Blockly tab**.

Interactive activity **3**
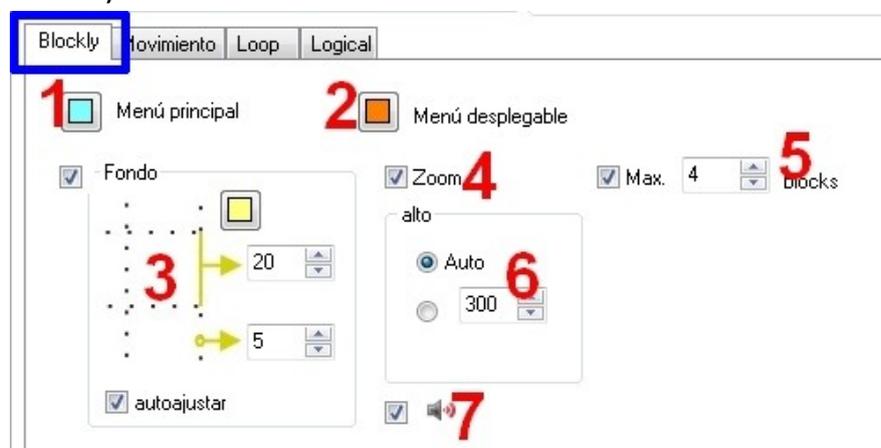
1.- By pressing this button we will choose the colour of the main menu bar of the right panel of the activity (left column with the icons to display the available commands).

2.- This button allows you to choose the background colour of the menu elements (area that appears when you click on one of the icons to display the menu).

3.- If we activate the "Background" box, the main area of the right panel will show a grid with points (small crosses). With the button at the top we can select the colour of these points and with the lower fields the distance between them and the thickness of that point respectively. If we check the "auto-adjust" box, the icons with commands that we drag from the menu will automatically adjust to the grid.

4.- When you check the "Zoom" box, on the right side of the panel above the trash (to which the user can drag any of the blocks to erase them), some buttons will be displayed to increase or decrease the size of the main area including the commands inserted in it. The upper button with the central dot is used to restore the entire set to its original size.
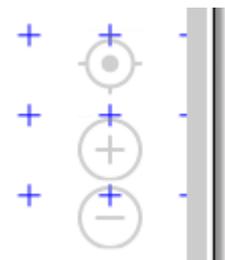
5.- In the world of information technology as well as in the world of mathematics, for example, we are aware that to reach the same result we can follow different paths. Ardora gives us the possibility that, by activating the box "Max." we can restrict the number of commands that the user will be able to drag from the menu to finish the activity in a satisfactory way by typing in that field the maximum number of blocks that the user will be able to use to complete the task correctly.

6.- In the "high" box, if we leave the "Auto" button checked, Ardora will automatically increase or decrease the height of the right panel to match the left panel. By marking the lower button and typing a number in the field, we will make the height of this panel fixed, regardless of the size of the left grid.

7.- If we tick this box, each time the user joins a block with another, a sound will be heard (small click) to know that it is correctly coupled.

c) **Movement tab**.
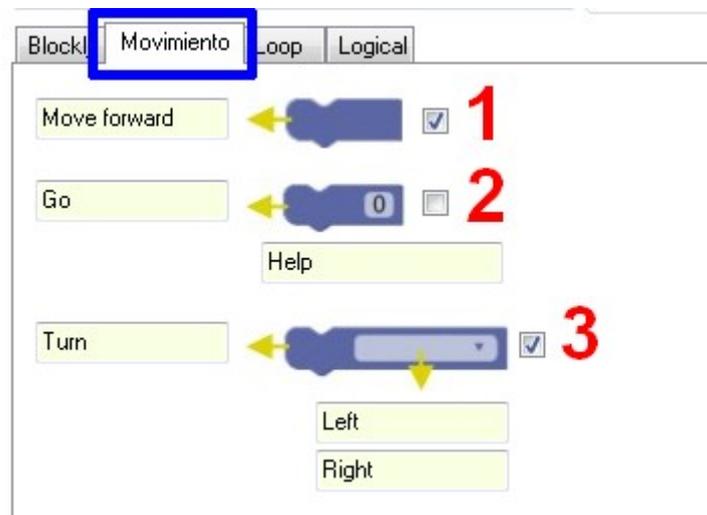
In this tab, and also in the following two (Loop and Logical), we can include or not some of these commands in the icons of the main menu (panel on the right, left side). In each of the fields of each of the three tabs you can find the commands in English for each of the commands, we can change the names of these fields if we want to display the blocks with the texts we wish, but it would be convenient to keep these texts in English, as this is how they would have to be written in any programming language.

1.- **Move forward >** Move forward indefinitely in the direction indicated by the tiny white line set in the red circle.

2.- **Go >** The same as the previous one, it allows you to move forward but this time only the number of squares that the user indicates in the small field embedded in the block. In the lower field (Help) we can write a short text that will be displayed in a pop-up label generated by the browser whenever the user places the mouse pointer over the block.

3.- **Turn >** Turn left or right (the user will have a drop-down list within the block) once the red circle has reached a given position.

d) **Loop tab**.

1.- **Repeat - times - do >** Repeat the action the number of times we decide in that field and then "do" or execute every order inside the input field after the word "do", where the user can insert several blocks.
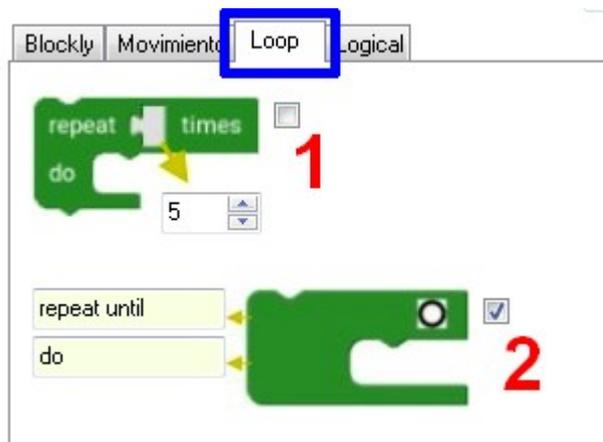
2.- **Repeat - until - do >** Repeat the commands that the user defines in the input field after the word "do" until the final white circle is reached, where the execution of orders will be immediately interrupted.

e) **Logical tab**.

1.- **If path - then>** If there is a path … (the user will have to display the list and choose between "ahead", "left" and "right"); then the orders inside the block "Then" will be executed.

2.- **If path - then -else >** just like the previous block, but this time adding one or more instructions. If there is a path... (the user will have to display the list and choose between "ahead", "left" and "right"); then the command of the block or blocks inserted into the field "then" will be executed. Once these commands are executed, the execution of the commands that the user includes in the second field "else" will be done.
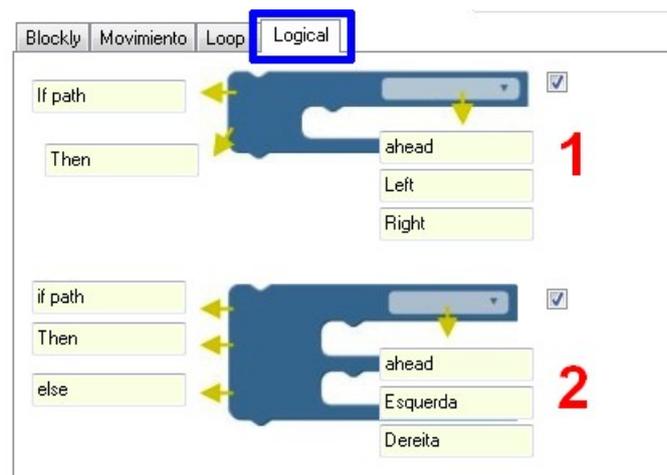
BLOCKLY - GEOMETRY

In this option, the user will need to add the necessary commands to build geometric forms that fulfill the conditions previously established or draw forms at will:

José M. Boo - 2018

TAB 1.- ACTIVITY

a) **Left panel**. in the left area we can find the following items:

1.- **Type of activity >** If we choose the option "Free", the lower box "Conditions" will be removed and the user will have the option to build the polygon he wants without any restriction. However, it would be advisable to leave the "Max." box unchecked inside the "Blockly" tab on the right side of the screen, so that the user has the option to create shapes without any limitation. If we choose the option "Conditioned", the box "Conditions" will be displayed at the bottom. It's there where we can set the parameters we are interested in to manage the activity according to our own standards.

2.- **Conditions >** In this box we will indicate, from top to bottom, the number of polygon sides that the user will need to create; in the drop-down list the kind of form (it doesn't need to be a regular polygon), the perimeter of that form and finally the area if we consider it appropriate. Read the attached notice for further information.

3.- **Left panel >** as in the "movement" option, in the upper field we will indicate the size of the side of the squares forming the boxes. In the next box we can increase or decrease the thickness of the grid points and with the buttons on the right we can select the colour of the grid points and the colour of the lines which will make up the polygon whenever the user clicks on the check button. In the lower field we can adjust the distance (horizontal and vertical) between the grid points.

4.- **Initial position >** as in the "movement" version, with these option buttons we will specify where the red starting circle will be facing to begin its movement and where to draw the first line of the polygon. With the coordinates shown in the parenthesis at the bottom, we can set the initial position of the red circle within the grid; for example, if we change the values of the parenthesis to (1,1), the circle will be displayed at the top left-hand side of the grid.

5.- **Left panel box >** With these buttons we can specify whether or not we want the box surrounding the grid to have rounded corners.

b) **Blockly tab**.

1.- By pressing this button we can choose the colour of the main menu bar of the right panel of the activity (left column with icons to display the available commands).
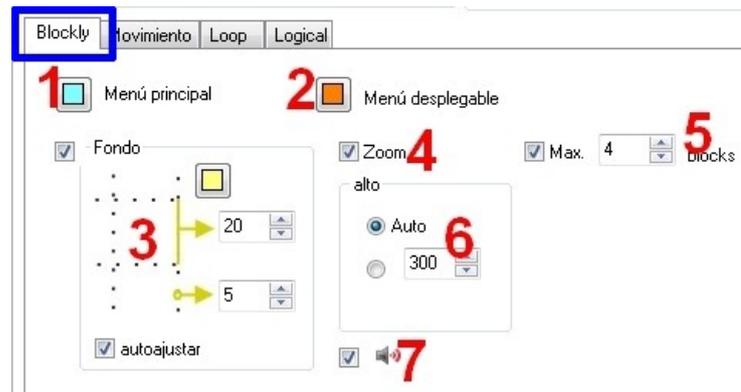
2.- This button allows you to choose the background colour of the menu elements (area shown when you click on one of the icons to display the menu).

3.- If we tick the "Background" box, the main area of the right panel will show a dotted grid (small crosses). With the button at the top we can choose the colour of these points and with the lower fields the distance between them and the thickness of that point respectively. If we tick the "auto-adjust" box, the icons with the commands that we drag from the menu will automatically adjust to the grid.

4.- When you tick the "Zoom" box, on the right side of the panel over the bin, some buttons will be displayed in order to increase or decrease the size of the main area including the commands inserted in it. The upper button is used to restore the whole to its original size.
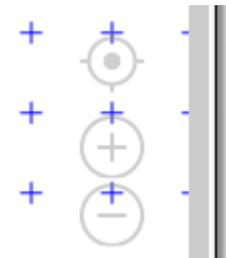

5.- In the world of computer science as well as in the world of mathematics, for example, we are aware that in order to reach the same result we can follow different paths. If we check the "max." box, Ardora will offer us the possibility of restricting the number of commands that the user can drag from the main menu to finish the activity in a satisfactory way by typing in this field the maximum number of blocks that the user can take to successfully complete the task.

6.- If you leave the "Auto" button marked in the "height" box, Ardora will automatically increase or decrease the height of the right panel to match the left panel. By marking the lower button and typing a number in the field, we will fix the height of this panel, regardless of the size of the left grid.

7.- If we tick this box, each time the user joins a block with another one, a sound will be heard (small click) when it is correctly attached.

c) **Movement tab**.

In this tab, and in the following two (Loop and Logical), we can add or not some of these commands to the icons of the main menu (right panel, left side). In all the fields of each of the three tabs there are commands in English corresponding to each of the commands; we can change the names of these fields if we want the blocks to be displayed with the texts we wish, but it would be advisable to maintain these texts in English, as this is how they would have to be written in any programming language.
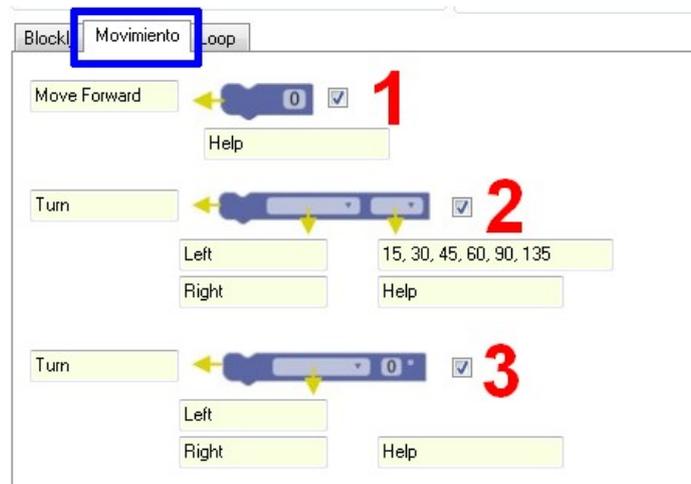
In the "Help" boxes we can write a short text that will be displayed in a pop-up label generated by the browser whenever the user places the mouse pointer over the corresponding block.

1.- **Move forward >** Move forward in the direction indicated by the small white line set in the red circle as many boxes as the user decides to enter in the small field included in the block.

2.- **Turn >** With this block, the users will have at their disposal 2 drop-down lists: the first one to turn left or right and the second one to specify the number of rotation degrees. A series of default angles are shown in the field, but we can modify this list by removing or including the angles we want in relation to the polygon that we would like to build. Just write the angles followed by a comma and separated by a space from the next angle, as shown in the default list.

3.- **Turn >** This block is similar to the previous one, with the only difference that instead of the second drop-down list, the user will find a text field in order to write the desired rotation angle.

d) **Loop tab**.

Repeat - times - do > Repeat the action the number of times we write in the field and then "do" or execute everything included in the entry after the word "do" where the user can insert more blocks of commands.
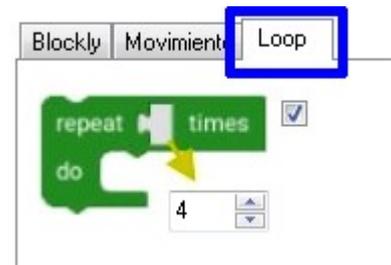
TABS 2 Y 3 - EXECUTION OPTIONS AND WEB PAGE

These tabs are used to set the different general parameters for messages, time, attempts, statement, web site background, tab title, etc. These common features can be consulted on the Ardora website, help page, document: points, time, messages and statement.