

Representar

User manual of the programming environment with drawings

The Representar environment is in its first days of dissemination to the public. The tool is a prototype and may have many aspects to be improved. We encourage you to try it, use it and send us any useful information to improve it. To do this, you can write to us by email at info@represent.ar or you can directly join our email list where other users also write and share ideas and projects. To join, send an email to representarCommunity+subscribe@googlegroups.com. Follow us and contact us both on Instagram and on X by writing to the user @agurafamartinez who is the referent of the project. The main point of our dissemination is the website represent.ar where we will update information and upload new versions of the tool with improvements and corrections.



A memory from 2023 when we held a workshop open to the community where children were able to use this programming tool for the first time. Thanks to them!! There were three days where we really enjoyed and learned from each other. They were the first to help us improve the Representar programming environment.

Index

[1. Installation](#)

[Windows](#)

[Linux](#)

[Mac](#)

[Not yet available on mobile phones](#)

[2. The programming environment](#)

[How to draw a character](#)

[How to use the predefined top bar symbols](#)

[Move the character with the arrows](#)

[push an object](#)

[To pull or pull an object](#)

[That one object cannot pass through another](#)

[Wildcard of any element](#)

[Wild of any symbol](#)

[run freely](#)

[Run towards another object](#)

[Collision rules](#)

[Teleport objects](#)

[Teleport to nothingness: the object disappears](#)

[Move with WASD keys](#)

[Space bar and Enter key](#)

[Categorize elements](#)

[See the result](#)

[3. Features that you can use](#)

[The top bar](#)

[Options on items drawn when on the board](#)

[Options on predefined symbols when on the board](#)

[Inspect execution](#)

[Change the speed of a running object](#)

[Reference a cell adjacent to an object](#)

[Options in the drawing bar](#)

[Options in the top symbol bar](#)

[Keyboard commands](#)

1. Installation

The environment works on computers running Windows, Linux or Mac OS. First you must unzip the Representar.zip file that you downloaded from the website and which contains the full version. Try following the instructions for your type of computer and if it gets complicated, write to us at info@represent.ar so we can help you make it work.

Windows

To open the environment in Windows you must run the file RepresentarWindows.bat. It is likely that some warning signs will appear, try to accept and give permission to run the program to get it running.

Linux

In Linux you must run the RepresentarLinux.sh script. (If your machine is 32-bit maybe the script doesn't work and needs to be fixed). If you have difficulties due to permission problems, use the same commands that we recommend for those who have a Mac. If you have any difficulties, write to us at info@represent.ar

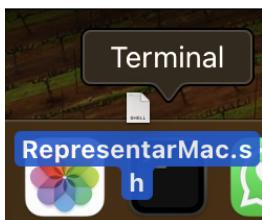
Mac

To open the program you can run the RepresentarMac.sh script. We leave you several ways to do it. Once it opens, allow it to access the folders it requests. If none of these ways work, do not hesitate to write to us at info@represent.ar

- Click the script with the second mouse button and open it with the Terminal:

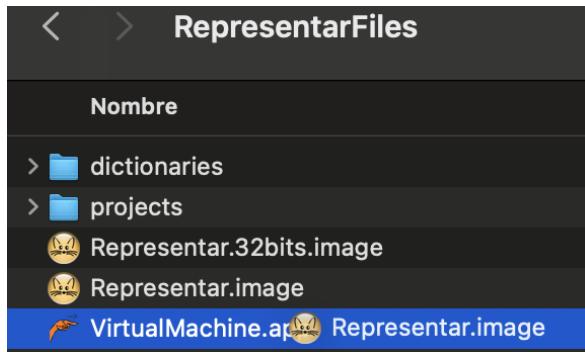


- Drag and drop it into the Terminal if it is in the bar below:



- Open Terminal.app and throw the file RepresentarMac.sh in the open terminal and hit Enter.
- Another way is to enter the RepresentarFiles folder and run VirtualMachine.app, then give 'Permission' to access folders when requested.

- Another way is to go to the RepresentarFiles folder and drag the Representar.image file onto VirtualMachine.app, then give 'Permission' to access folders when prompted.



- From the terminal, cd command and the directory where the script is RepresentarMac.sh. And then run it like this: ./RepresentarMac.sh
- If you get the message 'Permission denied', you must open the terminal in the script folder (you can throw the 'Representar Desktop' folder in the terminal or else with the terminal open, change the directory for example with: cd /Users/tuUsuario/Desktop/Representar\ Desktop/), then give execution permission with the command chmod -x RepresentarMac.sh and finally execute by typing ./RepresentarMac.sh

Not yet available on mobile phones

For now the tool does not work on mobile phones. The online version that can be accessed by clicking on the represent.ar website is limited, and is particularly difficult to use from a cell phone. If you try, we recommend doing it with the Chrome browser for a better experience.

2. The programming environment



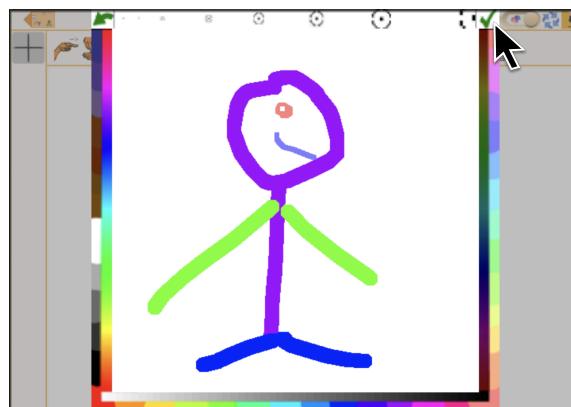
When opening we find this screen. We can create a new project with the button above. Change the language from Spanish to another that is available (if you need another language write to us, it is easy to add more languages).

Then you can enter a gallery of didactic and educational projects. And by clicking on the children's drawing of the mountain you will find some games and challenges.

How to draw a character



When you open the environment you find a button with a cross (top left) that allows you to create your first drawing. Clicking opens the drawing tool.



In the drawing tool you can choose the color by clicking on the color palette at the bottom and on the sides, you can also change the size of the brush by clicking on the top. If you didn't like a brushstroke, you can click on the back arrow to undo it. To erase, use the color white directly. And when you finish the drawing, click on ✓ (top right)

How to use the predefined top bar symbols

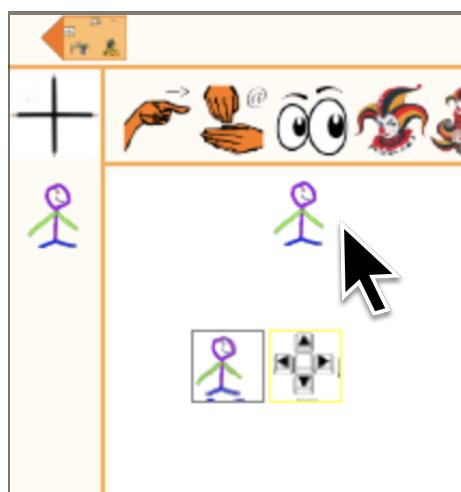
Above you find 16 predefined symbols that offer different possibilities. Let's start with the easiest thing, which is to manage the character with the keyboard and then we will see how to make it interact with other elements that you have drawn.

Move the character with the arrows



On the left you will find your drawings. If you take your drawing and throw it on the central board you will have your protagonist ready to act. What follows is to define a rule to move it.

To talk about your drawing you are going to use the same drawing, another copy, but the second copy is going to be a symbol. To make a drawing a symbol, click with the second mouse button and then click on the black button that appears at the bottom left.



When a box appears in your drawing it means that it is a symbol and you can build rules with it. The first rule is that your drawing moves with the arrows. With that you can move your protagonist.

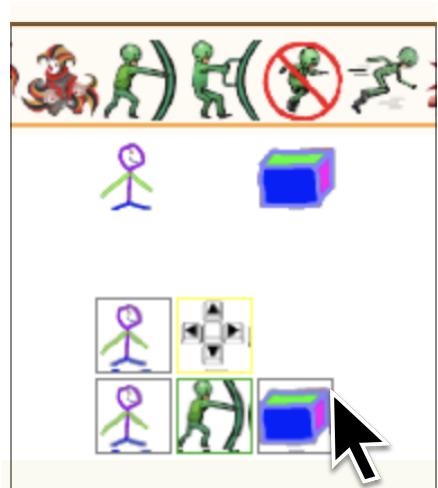
Here is the most important conceptual issue of our tool, a placed item already thrown on the board can be a symbol or an element. If it is a symbol it serves to create rules, if it is an element it participates in the game we are building.

push an object

If we draw another object we can begin to define interaction rules between the character and the other drawing. For example, if we draw a box we can have the character push it. For that we define another rule, which says that "the character pushes the box."

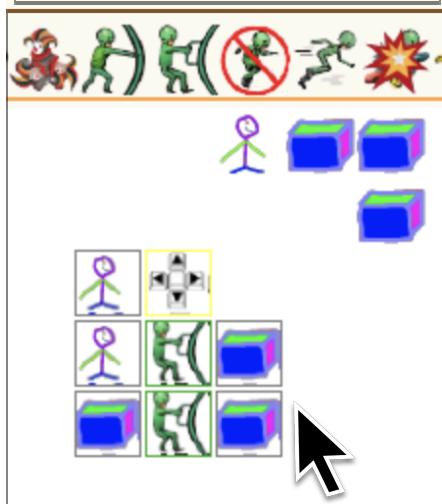
Now when the character encounters the box, he pushes it.

To pull or push an object



If instead of using the push symbol we use the pull symbol, then the character being next to the box will drag it wherever it goes.

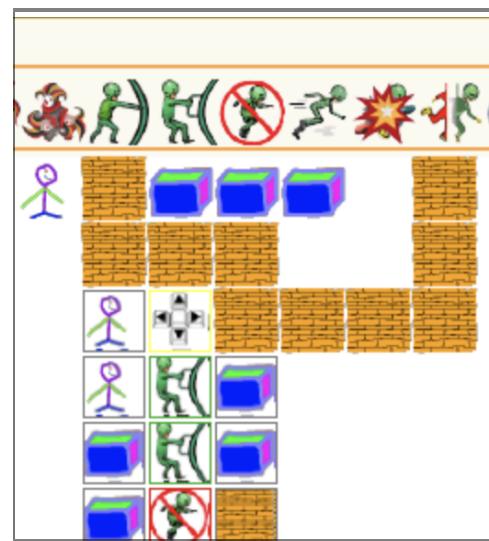
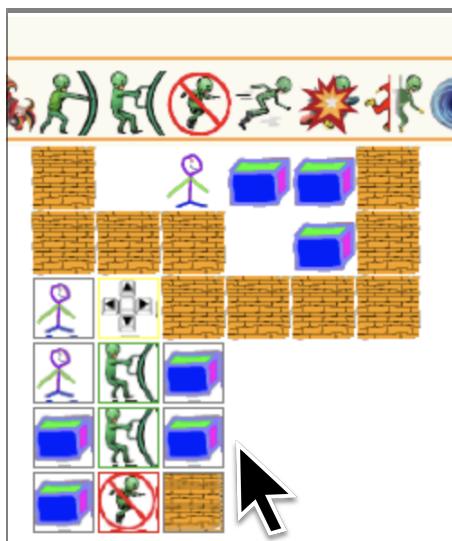
Once the box starts moving attached to the character, it will start to follow him all the time.



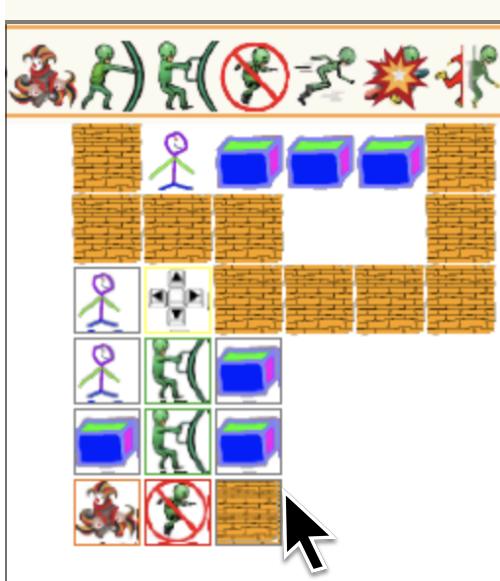
If we create another rule where the boxes also follow the boxes, we will have a little train where our character moves and a row of boxes follow him.

That one object cannot pass through another

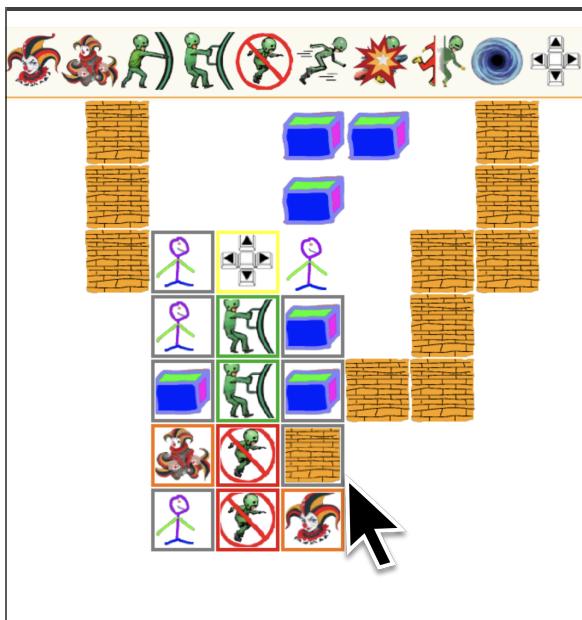
Suppose we now draw the wall and define a rule that says that blue boxes cannot pass through those walls. We added walls to enclose our character and his boxes between them. The character will be able to move freely (because there are no rules that prevent him from doing so) but the boxes will be in the way because they cannot go through the walls.



Wildcard of any element



Wild of any symbol



Let's look at the wildcard that represents any element (not to be confused with the other wildcard that we will see below).

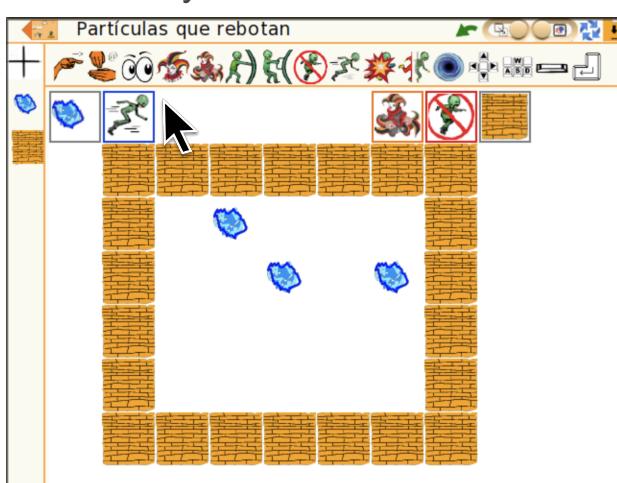
Suppose that now we do not want the boxes or the character to go through walls, and that no other element can do so either. We can create a general rule that says that "nothing and no one can go through walls."

In this way the walls are impossible to pass through any element of the board. Our character and his boxes remain locked.

Characters can move through the symbols we use to define rules. But if we want there to be interaction between the symbols and the characters, we can use the symbol wildcard. For example, if we want the symbols to be a wall for the character, we define the rule below. In this way our character is enclosed by the symbols, in addition to the walls that already enclosed him.

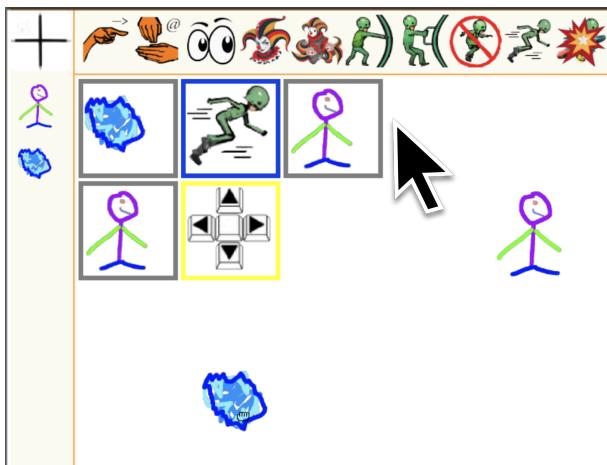
If we allow a character to push or pull the symbols, we can make the character change the program. In this way the program can modify itself, this quality is known as metaprogramming.

Run freely



We can define that an object runs freely on the board and bounces against everything it cannot pass through. For example, we can draw meteorites or particles that remain bouncing between the walls.

Run towards another object



We can also define that the particulars run towards the character. They will move all the time to the place where the character is. If the character moves they will move wherever he goes.

Collision rules



Through the collision symbol and the pointing hand, we can define actions that occur when two objects touch each other. In this case the particle is still, when the character moved by arrows touches the particle the defined collision rule is applied.

The rule reads: "if the character collides with the particle, then the particle begins to move freely."

Teleport objects



The teleport symbol can be used in a collision rule. In this case if the character touches the particle, then the particle teleports to the green mountain.

Teleport to nothingness: the object disappears



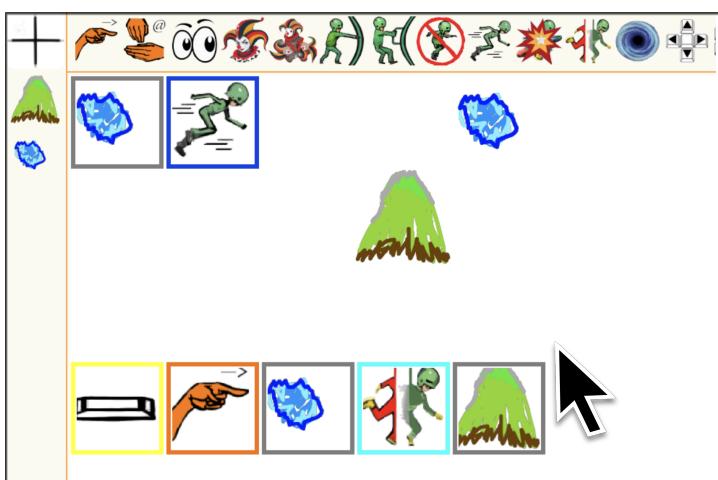
In this case the destination is not another object but the black hole. This black hole represents nothingness, the no-place. If something teleports to the black hole, it disappears. It is a way to delete an element through the program itself.

Move with WASD keys



If we want to move two characters differently, we can move one with arrows and move the other with the WASD keys on the keyboard.

Space bar and Enter key



We can generate actions that are performed when we touch the space bar (or Enter).

In this case, by pressing the keyboard bar, a new particle is teleported to the mountain. It is a way of adding objects through rules.

In this case, if we press the bar many times, the board will be filled with particles that run freely. They are all going to emerge on the mountain.

Categorize elements

The symbol with two hands expresses that one symbol is placed inside another. Creating categories allows us to create more general rules.

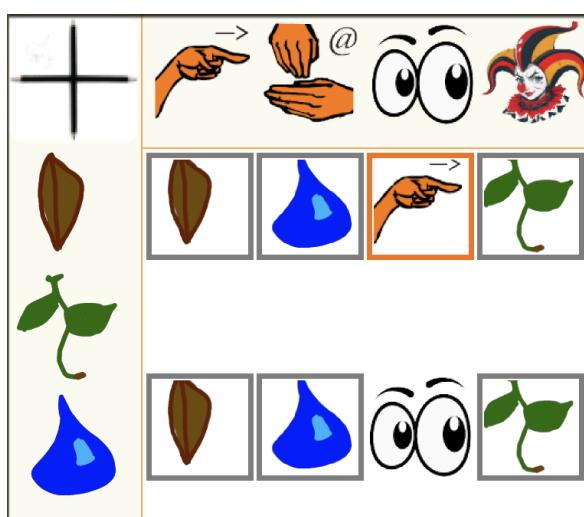
If we have two types of particles, red ones and blue ones. And we create a symbol that will represent the idea of a particle. By placing the red and blue particles inside this idea, we can

create rules about that idea directly: for example that every particle runs freely. The abstract idea of particle is an example of a category based classification.

Another way to categorize is through exemplars. For example, if a new type of wall appears with bars, we can say that this new type of wall is categorized by the original wall. The rules that applied to the wall now also apply to the wall with bars. This is an example of a category based on a specific item that already existed. That is, a category based on prototypes.



See the result



The pointing hand allows the creation of substitution relationships between symbols. For example 'seed and water' is replaced by 'sprout'. Then with my eyes I can see the result of evaluating the expression 'seed and water'. When the expression is evaluated, the defined substitution is applied, and the result is an outbreak.

The eyes take the sequence of symbols on their left and print the result on their right. In computing jargon this functionality is known as REPL (read-eval-print-loop).

3. Features that you can use

The top bar



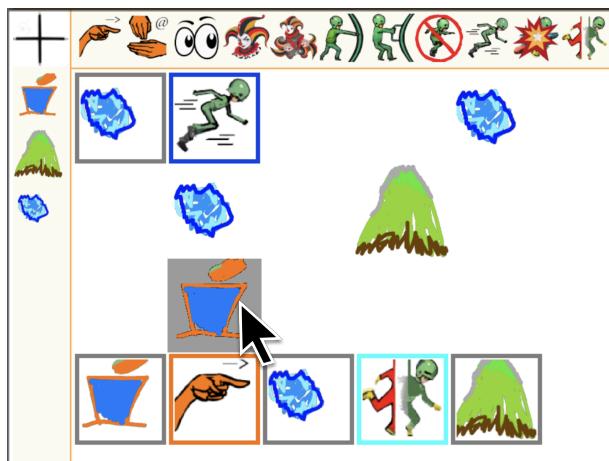
From left to right these are the functions in the top bar.

- Button to return to the previous screen.
- Space to put the name of your project.
- Button with a green arrow that allows you to undo the movements that have occurred on the board.
- Switch that supports two ways of placing elements. Standard mode allows you to grab an item from the symbol bars and throw it on the board. The other way works like a stamp: you choose a drawing and stamp as many elements as you want from that drawing as if it were a stamp.
- Switch that admits two states of the drawings to be placed. One where every drawing is an element of the board and another where everything that is placed is always a symbol. Any placed element can then be inverted from symbol to element and vice versa. This mode is like when we are working purely with symbols, for example when representing natural numbers.
- Button to reuse another project within this project. It allows us to choose another project and have all its drawings available. The rules from that other project are valid for this one even if they are not visible on the board.
- Button to save the project to file. To share your program you can save your project to a file and send it. It is saved with the name you have given it without spaces, a version number and the extension represent.ar (e.g. Nombredelproyecto.001.represent.ar). It will never delete a previous file, it always creates one with a new version number.

Options on items drawn when on the board



When you click with the second mouse button on an element you have drawn, 4 options appear. The red button with the x allows you to delete it. The green button with the eye allows you to make it invisible. The black/white button at the bottom left transforms it from a symbol to an element and vice versa.



The green button at the bottom left transforms the symbol into a button, when the button is pressed the symbol is evaluated. For example, it is possible to create a button that, when pressed, evaluates a substitution that teleports new particles to the mountain. In this case the button works the same as an Enter key or Spacebar button, the difference is that it is pressed with the mouse.



When a drawing is set as an element (without the symbol box) it is possible to enlarge it.

Options on predefined symbols when on the board

The predefined symbols have particular options detailed below.

Inspect execution

The little eyes, which allow you to see the result, have the option of seeing how the execution went. What is known in programming as debugging. By clicking on the green button below we see how the execution was. To see it, all the substitutions applied are listed (in this case only one) to arrive at the result.

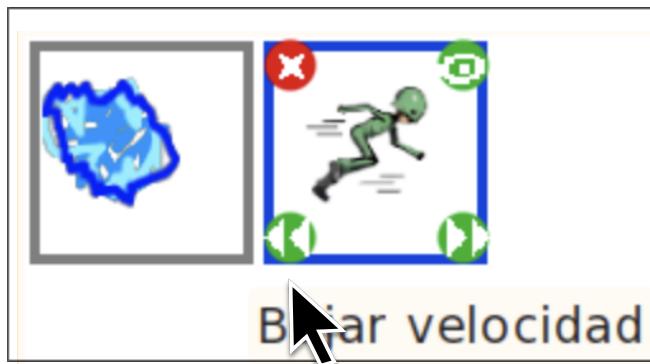


Another way to inspect the execution is to set what is known as a “breakpoint”. In this case, when the substitution that has the interruption indicated is applied, then the panel with the executions opens. The substitution that interrupted the execution is indicated in green.



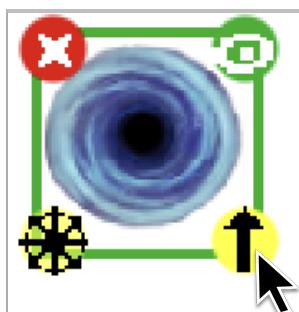
Any relationship where the pointing hand allows us to define a breakpoint. This helps you see how the program works.

Change the speed of a running object



For any object that runs it is possible to lower and increase the speed by modifying the running symbol.

Reference a cell adjacent to an object



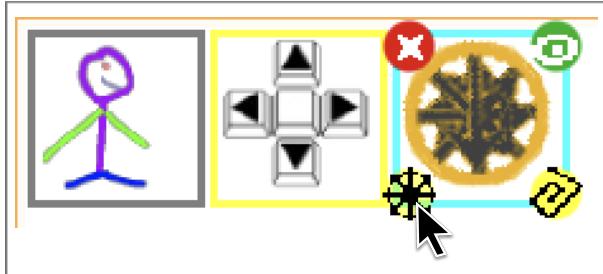
The black hole symbol that represents non-place can be transformed into a symbol that refers to places such as a position higher, lower, etc.



It is possible to use it to point in different directions and run in a fixed direction.

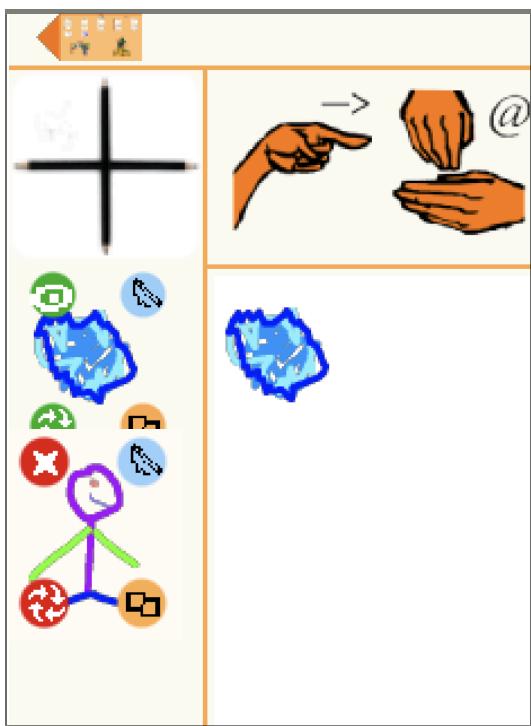


It is also possible to use this symbol to teleport an object from where it is to the next cell in any direction that has been configured.



This combination allows the character to move with the arrows, but instead of moving cell by cell, they keep moving until they encounter an obstacle. That is, with a single touch on the arrow, you move in that direction as far as you can.

Options in the drawing bar



By clicking with the second mouse button on a drawing in the left sidebar, 4 options appear. The eye allows all elements of this type to be made invisible (if they have a frame, they remain visible, because they are symbols). In the event that there is no element or symbol of that type on the board, instead of the eye appearing, a red button appears that allows you to delete the drawing (a drawing can only be deleted from the bar if it is not being used in the board).

Then a light blue button with a pencil that allows you to edit the drawing (when it is modified all copies of the board are updated).

The beige button at the bottom right allows you to create a new drawing from this one. Instead of starting with the white canvas, we start with this drawing as a starting point.

At the bottom right a green button that configures whether this drawing is reusable when the project is reused by another project. When the drawing is “hidden” from other projects the button displays red.

Options in the top symbol bar



If there are hidden elements and symbols and we want them all to become visible, we have an option to do so on the wildcard that represents all the items.



If we want to hide all the symbols on the board we have the option in the wildcard that represents the symbols.

It serves to make the code invisible. When you click on the green eye, all the rules of the program are no longer visible and only the game elements are visible. If this button is red, what it will do is the opposite, making all the hidden symbols visible.

Keyboard commands

There are several keyboard commands available. They are activated using the Control key and at the same time some letter on the keyboard. On Mac the key is Command and on other operating systems the key may be Alt.

Control-Z	Undo. The last move of the board goes back.
Control+	Shows board items larger
Control-	Show the board items smaller
Control-W	Allows you to go back, to return to the project gallery
Control-A	Allows you to select all the items on the board

By clicking the right mouse button on the board and releasing it in another position, we can select the items enclosed in those positions. With them we can use the following commands:

Control-C	Copy selected items
Control-X	Cut selected items
Control-V	Get copied or cut items