Coding Unplugged

focus on problem solving

Stimulate problem solving	**Gold**	**Stimulate entrepreneurship**	**Bronze**
Stimulate creativity	**Gold**	**Informal learning enviro.**	**Gold**
Stimulate critical thinking	**Gold**	**Technology use**	**Bronze**
Stimulate group work	**Platinum**		

Practicalities

Preparation: 30’

Duration: 1h / 1h 30’

Material needs:

- Copy paper or pieces of white paper
- Multi-Colored Paper
- Pencils
- Rubber
- Markers
- Paper tape
- Scissors

Group size range: 20
Ideal sub-group size: 2

Workshop made for: up to 12 (-12)
Easily transferable to workshops for ages between: 12 - 16

Environment FabLab necessary: NO

Educational area:

* Computer science
* Mathematics
* Engineering

Precognition

> *Programs are made up of sequences of instructions.*
> *A sequence is just steps of instructions in a certain order. This set of instructions is called an algorithm that is created to solve problems, like searching for and sorting information. Some algorithms are very effective, some of them fast, some of them accurate. When trying to find a route between two places, sometimes you want the fastest route, sometimes you want the fewest crossroads, and sometimes you want to stop and smell the flowers. Different algorithms can help you do each.*
>
> Liukas, L. (2015). *Hello Ruby.* Crawfordsville, Indiana: R.R. Donnelley & Sons Company

Students apply logic and sequencing skills to write instructions for a student to complete a simple task acting as a robot.

(see box 'content links' below)

Preparation

Be sure to have the following materials available:

- * Templates (1-2) to each team (it's better to print more copies)
- Pencils
- Markers
- Paper tape

*(see templates below)

Environment

You need a large environment where build a sort of big chessboard on the floor where kids can move (when the subgroup is bigger than 10, it's better to provide more chessboards).

I suggest to create the chessboard with paper tape as the grid illustrated in the template 2, with the same number of squares (6X10). The square size should be adapted according to the kids´foot.

Make sure you have some tables where the groups can work on the paper templates before moving on the chessboard.

Workshop Guidelines

Phase 1: Orientation and Instruction Phase

Material needs:

Essential:

- *Picture of instructions
- *Template 1 and 2

*(see page 'INSTRUCTIONS' below)

*(see page 'TEMPLATE 1 and 2' below)

Goals:
Skill Goals (**Blue**)

(S1) Awareness of the topic
(S2) Teamwork

Content Goals (**Green**)

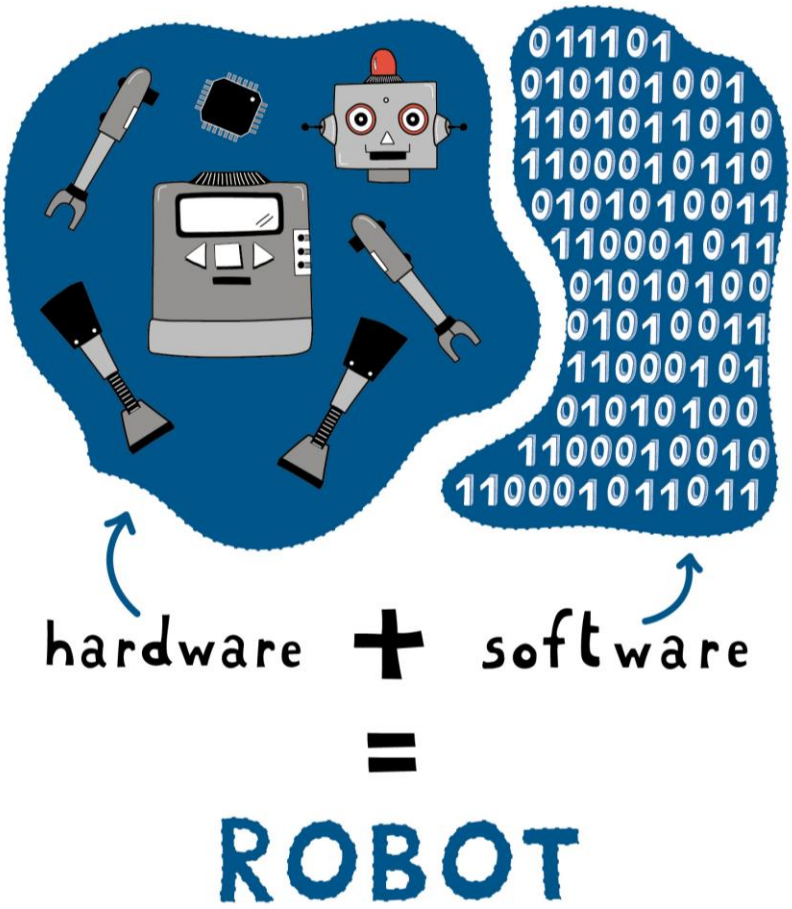
(C1) Understand algorithm concept

Background story:
Coding unplugged is about all the wonderful activities that reinforce coding concepts and don't involve screens. These unplugged coding activities teach the basic concepts of coding without needing the use of digital devices.

Unplugged coding activities have an interesting approach which involves grids or maps drawn on a sheet of paper or on the floor. In this way kids are involved in role-playing games: the role of the programmer and the role of the robot that has to perform the instructions. The programmer uses a symbolic programming language, for example four directional arrow and writes the program on a small grid. The robot listens to the instructions and performs them walking on the grid.

Goals	Activities	Duration
S1, C1	**Robot Language Dictionary** The basic process is to get all the kids together and starting a conversation with some questions: *What is a robot?* *How could we program a robot to make it move?* *How does a robot know what to do in order to complete a task?* It's not easy to define what robots are, and it's not easy to categorize them either. There are many different types of robots: Industrial, Medical, Drones, Military and Security, Self-Driving cars, Humanoids, Entertainment, ... Each robot has its own unique features, and as a whole robots vary hugely in size, shape, and capabilities. They were created to perform one or more specific actions in a certain environment, with or without the human being.	15'

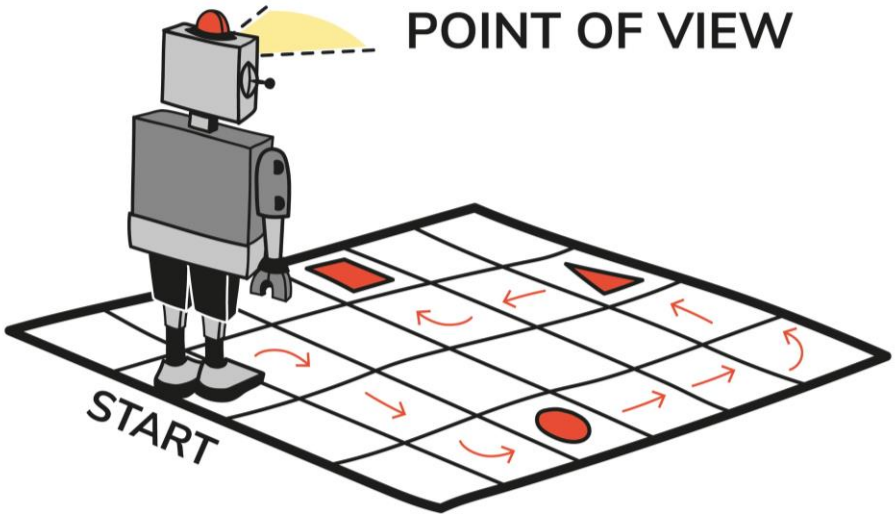
A robot is composed by the electrical or mechanical parts which are called ***hardware*** and the instructions and the programs that run inside a computer which are called the ***software***.



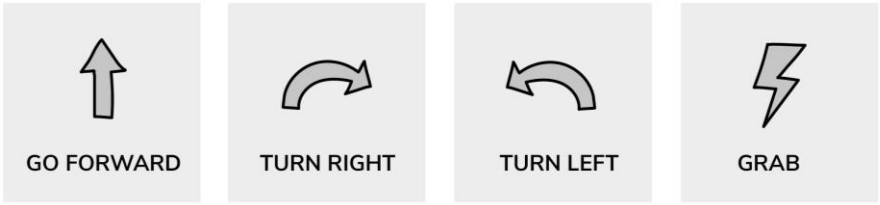
After a quick chat with the class it's time **to start the challenge**.

A robot is able to move if it receives specific instructions, otherwise it doesn't move.

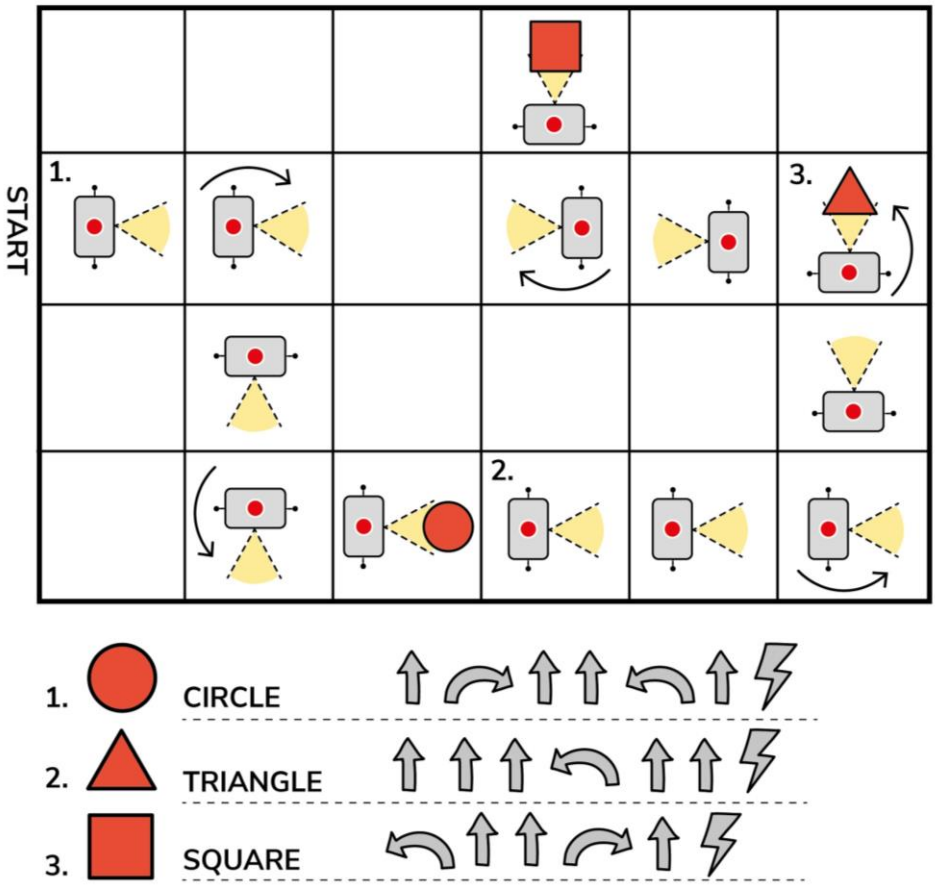
Below an example of how a robot could move on a map to retrieve some geometric shapes:



Instructions:



Below the code written by a programmer to help the robot recover all the geometric shapes on the map.

	 Show all the pictures to the whole class! The instructions could be written on a board or projected on a screen. **(see page 'INSTRUCTIONS' below)**	
S2	**First step** Divide the whole class into pairs or groups of three. **Second step** Deliver the templates 1-2 to each team.	5'

Phase 2: Design Phase

Material needs:

Essential:

- *Template 1-2
- Pencils
- Markers
- Rubbers
- Scissors

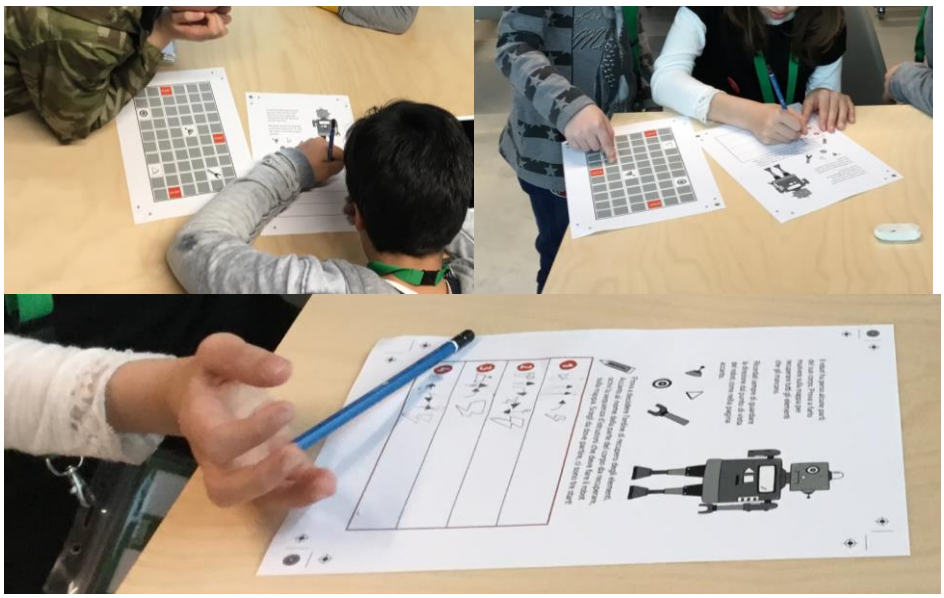
*(see page 'TEMPLATE 1 and 2' below)

Goals:

Skill Goals (**Blue**)
(S1) Teamwork
(S2) Problem solving
(S3) Reach goals

Content Goals (**Green**)
(C1) Decomposition
(C2) Sequence
(C3) Pattern recognition
(C4) Debugging

Goals	Activities	Duration
S1, S2, S3, C1, C2, C3, C4	**Template 1** What happened to the robot? The robot has lost parts of its body! Try to make it move on the map to retrieve all the elements. Remember to look at the direction from the robot's point of view. ● Decide on the recovery order of the elements. ● There are three **START** on the map**,** each team can choose where to start. ● Write the right sequence of the instruction next to each element. **Template 2** The map with four elements of the robot to be recovered. Students can add some variations on the map, like: ● The least amount of steps necessary (optimisation-process) ● Hurdles on the way ● ...	30'



Phase 3: Testing Phase

Material needs:

Essential:

- Paper tape
- Copy paper or pieces of white paper
- Multi-Colored Paper

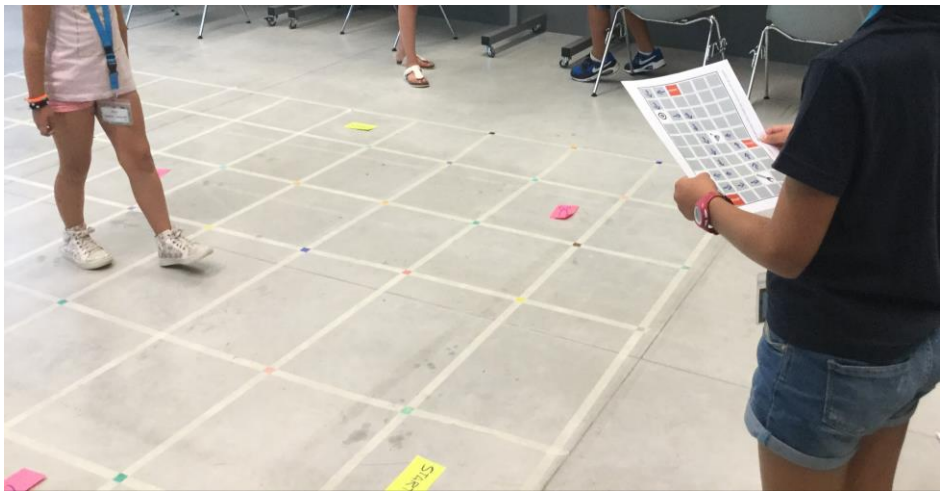
Optional:
*Prints all the parts of the robot to be recovered and put them on the chessboard

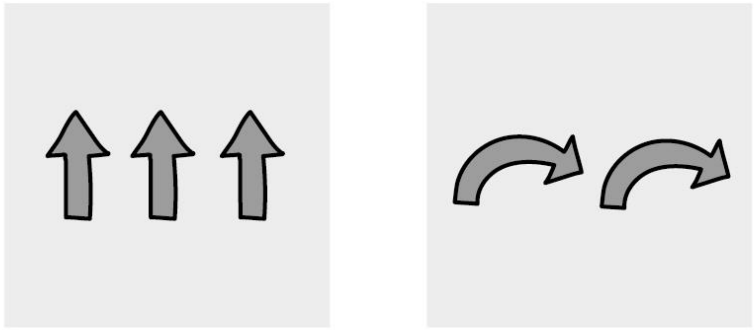
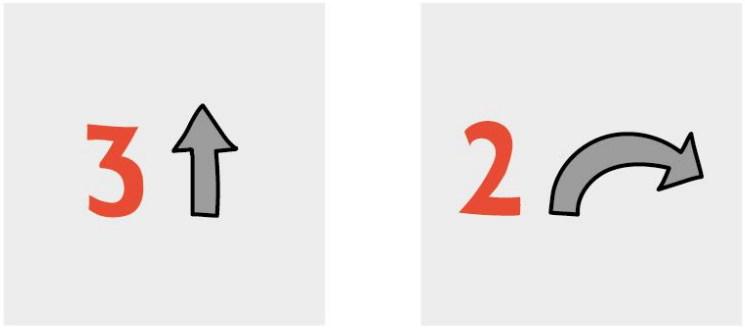
*(see pictures below)

Goals:

Skill Goals (**Blue**)
(S1) Teamwork
(S2) Problem solving
(S3) Identify yourself in a role
(S4) Reach goals

Content Goals (**Green**)
(C1) Sequence
(C2) Debugging
(C3) Orientate yourself in a physical space

Goals	Activities	Duration
S1, S3, C1, C3	Let’s test our code on the chessboard! **Option 1** The teacher/educator printed all the parts of the robot to be recovered and place them on the chessboard. **Option 2** Each team draw the parts of the robot in separate sheets and place them on the chessboard when it's their turn. **It's time to choose the team roles!** Who is the *robot?* Who is the *programmer?* The *robot* has to complete the code that the *programmer* says. The teacher/educator pick a team that will test their instructions on the chessboard as an example to the rest of the class.  This moment is great for sharing and reflecting together on the first part of the activity. Also it could be the right moment to talk about the *debug* definition (discovering and solving mistakes in computer programs). *The word bug originally came from a moth found in a computer in 1947 by Admiral Grace Hopper. There are generally two types of bugs computer programmers face. First, the syntax errors, like when a programmer mistypes a word or forgets a semicolon. Second, the logic errors, where the code doesn’t do the right thing.* Probably students had to do the same with their codes after saw the robot in action.	10’/15’

	Some tips Ask to the whole class: *Did you have to repeat the same command several times?* Like these two:  *We may find a way to make it short:*  Ask to the whole class if they have any suggestions.	

Goals	Activities	Duration
S1, S2, S4, C1, C2, C3	**First turn** At least three groups can test their code at the same time on the chessboard.	15’/20’

	 In the meantime the rest of the class watch the groups waiting for their turn outside the chessboard. **Let's start the second turn! (If the time is not over)** Team roles are reversed and the groups repeat another time the instructions on the chessboard.	

Phase 4: Evaluation Phase

Material needs:

Essential:

- Pencils
- Paper or post-it

Goals:

Skill Goals (**Blue**)
(S1) Reflecting on the project
(S2) Communication

Goals	Activities	Duration
S1, S2	**Reflection** Ask each participant to reflect on their own about the whole process. Provide them with some guiding questions like: ● Was it hard to write the instructions to recover all the parts of the robot? ● What did work well? Why? ● What was the most difficult thing to achieve as a team? Why?	10'

	Sharing Let students share their reflection. If students want to continue the experience at home, you can show to the whole class some software and app about the way people tell a computer what to do using instructions that the computer understands. Here some examples: Lightbot.com Runmarco.allcancode.com *Bee-Bot app*	

Pedagogical tips

Learning how to program is going to be the most useful new skill we can teach today.

Through coding and without digital devices it's possible to get basic programming concepts developing computational thinking.
Programming helps to achieve:

- Logical skills
- Problem solving in a creative way
- Management of complex tasks
- Planning
- Sequencing
- Debugging (the process of identifying and removing errors from a code)

The debugging process is really important in this workshop.
Facilitating this activity require you to support each team to test the code and not be afraid to make mistakes.
The code can be written several times and improved, just like real programmers do!

Invite groups to take a look on different solutions and what's happening around during the activity.

How to transfer to non-Fablab environment

Transfer to non-fablab environment is very feasible, as long there is enough space on the floor and a few tables, one for each group.

Evaluation of achievements

At the end of the workshop you can give the different groups achievements.
For example for:

- The most collaborative team or the team that helped more the others.
- The best debugging process to achieve a good code
- The best performance/role-play on the chessboard

Content links

Some interesting links that can inspire you:

- helloruby.com/play
- how to train your robot
- Csunplugged.org
- Olaf de Robot

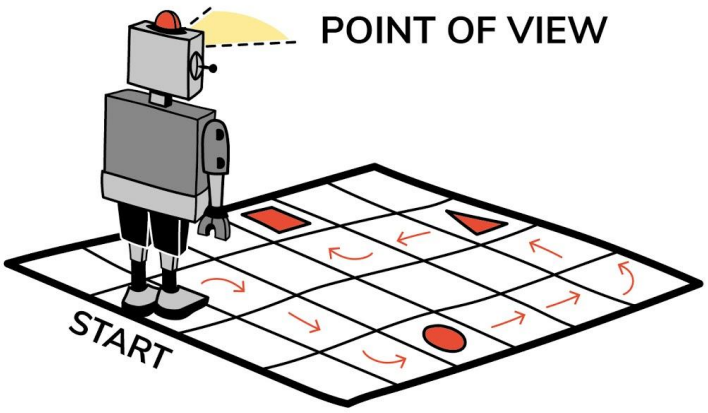
INSTRUCTIONS

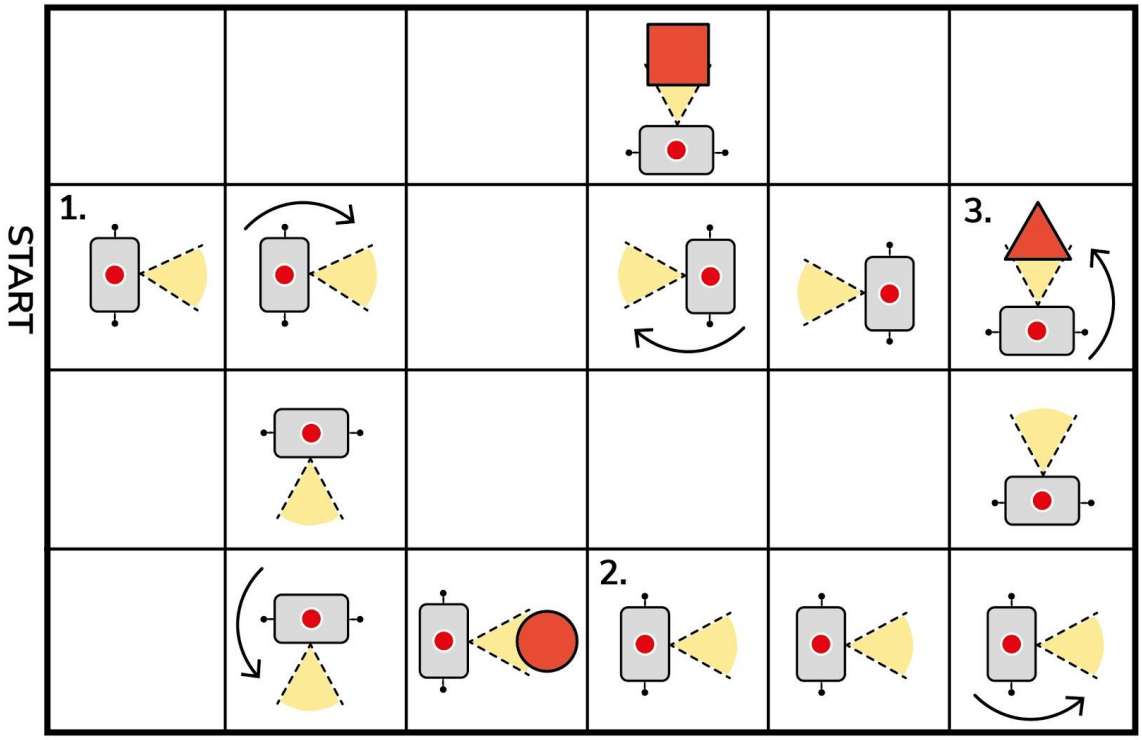
GO FORWARD

TURN RIGHT

TURN LEFT

GRAB





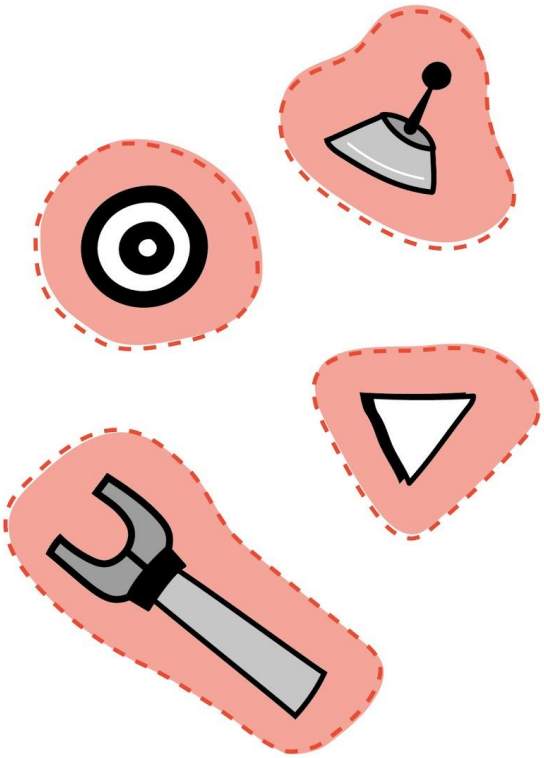
1. CIRCLE

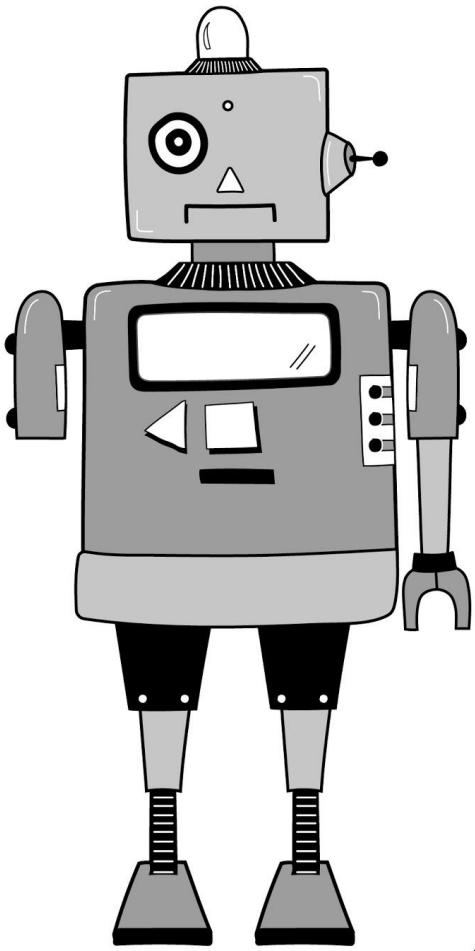
2. TRIANGLE

3. SQUARE

ILLUSTRATION BY VALENTINA ERCOLANI

TEMPLATE 1





1.

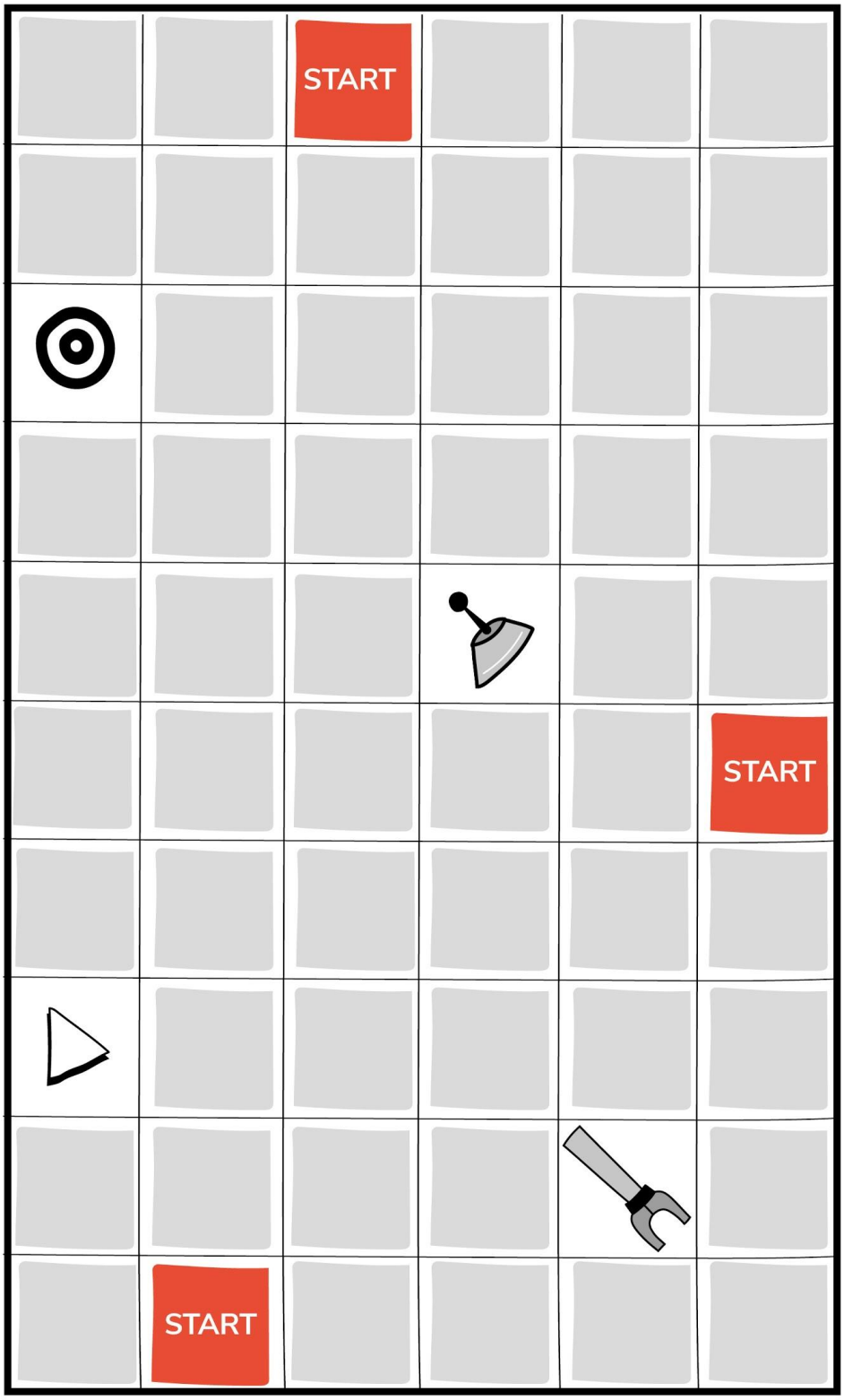
2.

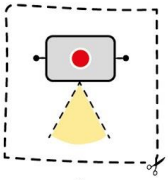
3.

4.

ILLUSTRATION BY VALENTINA ERCOLANI

TEMPLATE 2





Cut this small square and use it as a guide. It will help you to recover all the elements on the grid!

ILLUSTRATION BY VALENTINA ERCOLANI