

Coding basics

focus on problem solving

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

Practicalities



Preparation: 30'



Group size range: 20

Ideal sub-group size: 2



Duration: 2h 30



Workshop made for: children between 12 - 16)



Material needs:



Environment FabLab necessary: NO

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



Educational area:

* Computer science

* Mathematics

- Paper tape
- Scissors
- Access to computers or laptops
(internet)

* 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 (1-3) 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:

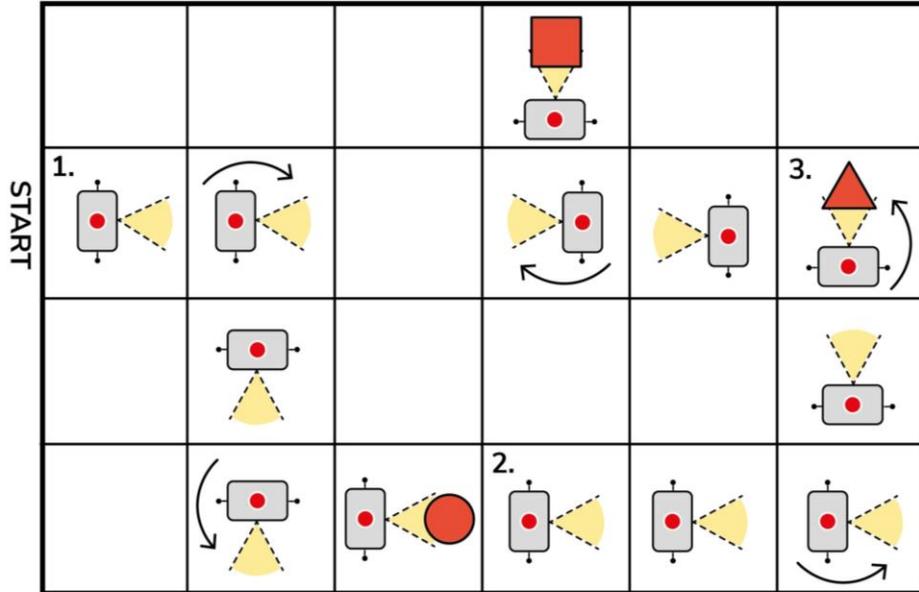
The first few phases are unplugged. Coding unplugged is about all the wonderful activities that reinforce coding concepts and don't involve screens. These unplugged coding activities teach the basic concepts of coding without needing the use of digital devices.

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.

After the unplugged phase the students use their new knowledge in the digital world. With Blocklygames they can try out different things just the same way they have been before. Using steps to tell what the 'robot' should do. But now they also see them in practice by a computer program.

Goals	Activities	Duration
S1, C1	<p>Robot Language Dictionary</p> <p>The basic process is to get all the kids together to start a conversation with some questions:</p> <p><i>What is a robot?</i></p> <p><i>How could we program a robot to make it move?</i></p>	15'

shapes on the map:



- 1.  CIRCLE ↑ ↻ ↑ ↑ ↻ ↑ ⚡

- 2.  TRIANGLE ↑ ↑ ↑ ↻ ↑ ↑ ⚡

- 3.  SQUARE ↻ ↑ ↑ ↻ ↑ ⚡

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.

5'

	<p>Second step</p> <p>Deliver the templates 1-2 to each team.</p>	
--	--	--

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	<p>Template 1</p> <p>What happened to the robot?</p> <p>The robot has lost parts of its body! Try to make it move on the map to retrieve all</p>	30'

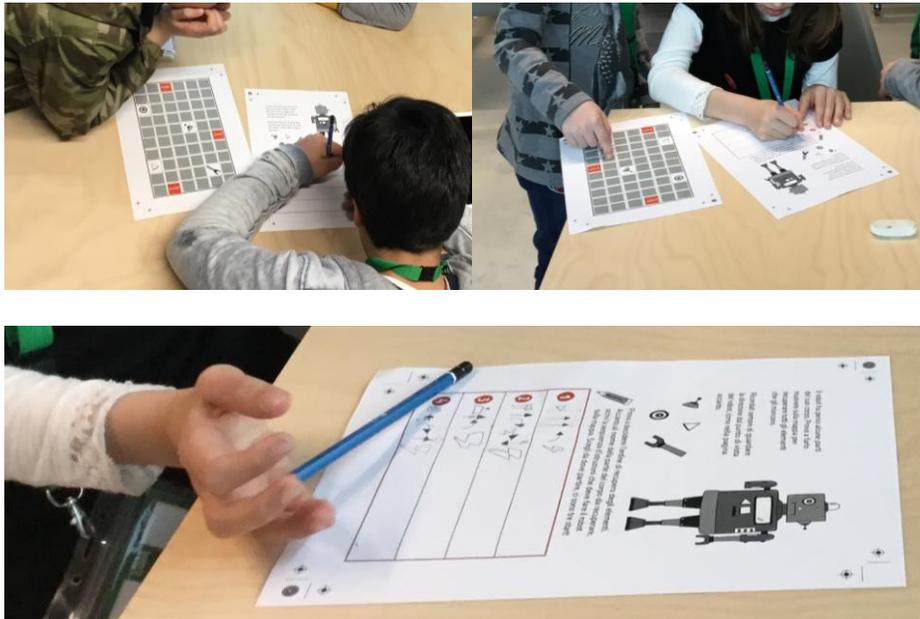
the elements.

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.



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

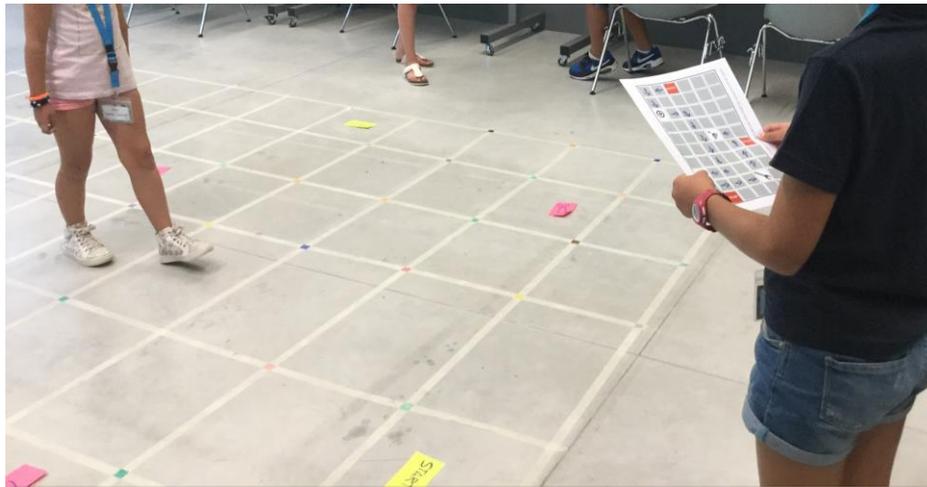
Goals	Activities	Duration
S1, S3, C1, C3	<p>Let's test our code on the chessboard!</p> <p>Option 1</p> <p>The teacher/educator printed all the parts of the robot to be recovered and place them on the chessboard.</p> <p>Option 2</p> <p>Each team draw the parts of the robot in separate sheets and place them on the chessboard when it's their turn.</p> <p>It's time to choose the team roles! Ask to each group:</p>	10'/15'

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.

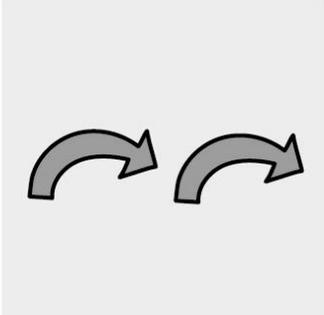


This moment is great for sharing and reflecting together on the first part of the activity.

Some tips

Ask to the whole class: *Did you have to repeat the same command several times?*

Like these two:

	<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="background-color: #e0e0e0; padding: 20px; border: 1px solid #ccc;">  </div> <div style="background-color: #e0e0e0; padding: 20px; border: 1px solid #ccc;">  </div> </div> <p style="text-align: center; margin: 20px 0;"><i>We may find a way to make it short:</i></p> <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="background-color: #e0e0e0; padding: 20px; border: 1px solid #ccc;">  </div> <div style="background-color: #e0e0e0; padding: 20px; border: 1px solid #ccc;">  </div> </div> <p style="text-align: center; margin-top: 20px;">Ask to the whole class if they have any suggestions.</p>	
--	---	--

Phase 4: Let's race it out, who is the better programmer



Material needs:

Essential:

- Paper tape
- *template 3
- Template 4
- Chronometer
- Whitboard or blackboard to note score

*(see page 'TEMPLATE 3 and 4 below)

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, S2	<p>The children now know all the basics needed to make improvements themselves and aim for the lowest possible score.</p> <p>Each group gets a bundle of layouts on the chessboard with increasing difficulty. Within an allotted time limit they have to finish as many as possible.</p> <p>Each pattern they finish can earn them points. The group at the end with the most points wins a (chosen by the teacher) prize.</p> <p>How to score groups? -> each action input like move or turn is one action. After completion of the course count all the actions they made and that is their score for that pattern. When the time is up an groups haven't finished all the courses, each course counts as bonus points. (10 points per unfinished pattern). The group with the least points wins.</p> <p>The score is noted per group on the board.</p> <p>Each exercise has a blanco for the student to use and a solved for the teacher to use. There are multiple ways to solve these puzzles zo when they have a different route, it doesn't mean they are wrong.</p> <p>With the collected pieces they can complete the robot on template 4</p> <p>Roles assignment</p> <p>When groups want to test-show their code they put up their finger. When 2 groups are ready, 1 person of a group is the programmer and 1 person of the other group is the robot. (AA and BB form AB BA).</p> <p>The robot has no interest in cheating because he would be helping another group.</p>	30'

Phase 5: Blockly



Material needs:

Essential:

- ≠ Pencil, and any template
- ≠ Access to a computer per group
- ≠ Access to an internet connectio



Goals:

Skill Goals (**Blue**)

Goals	Activities	Duration
	<p>Phase one: learning loops</p> <p>When using the program called blocklygames they have to know how to code with loops. In this first phase the teacher explains how loops work and gives several examples.</p> <p>First up is giving the inputs used in blockly on paper.</p> <p>The first three are self-explanatory, with the last 2 you should explain it in depth and maybe give a short example.</p> <div style="text-align: right;">  </div> <p>Example (X= object to collect)</p>	30'

The following code wil make it so that the robot will follow the outer lines and collect all the objects.

	X		
			x
x			
->			

Phase two: using blockly

URL to website: <https://blockly.games/?lang=en>
then go to MAZE for the exercises.

After the students have practiced some loops, they can move on to using the program. In the program they will also get a short tutorial how to use the blocks and how to place them.

As teacher you can check if they have successfully completed a level and not skipped one if the circle corresponding to the level is grayed out.



The first two levels are completed, but the third is not completed and skipped over.

Phase 6: 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, S4,	<p>Reflection</p> <p>Ask each participant to reflect on their own about the whole process.</p> <p>Provide them with some guiding questions like:</p> <ul style="list-style-type: none"> ● 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? <p>Sharing</p> <p>Let students share their reflection.</p>	10'



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



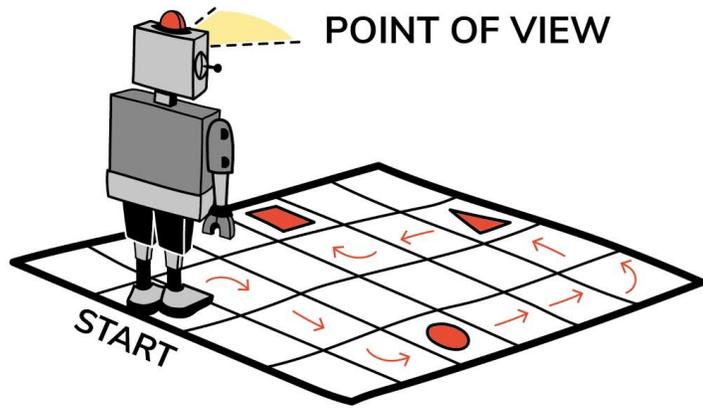
Content links

Some interesting links that can inspire you:

- helloruby.com/play

- [how to train your robot](#)
- [csunplugged.org](#)

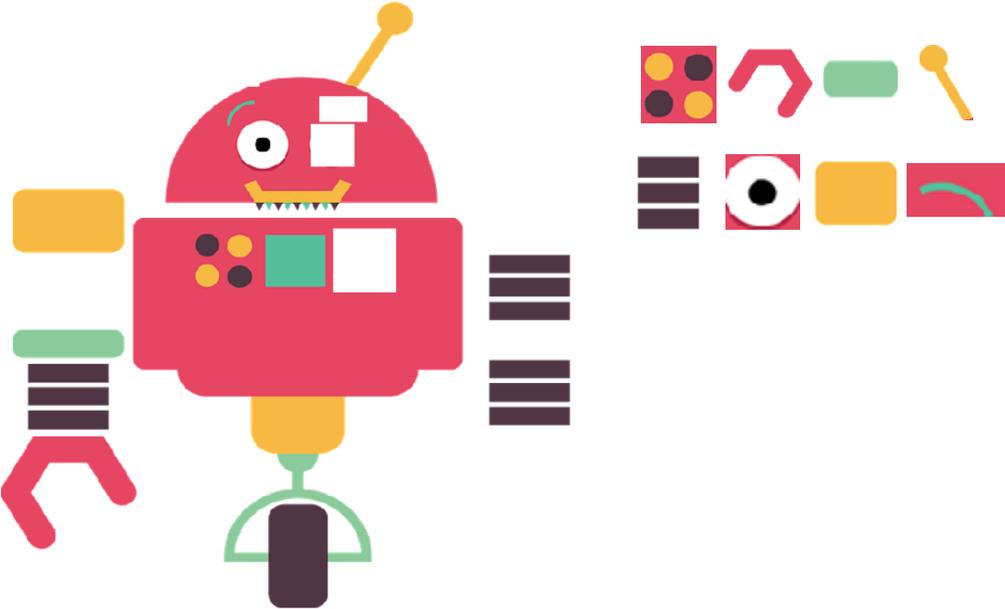
INSTRUCTIONS



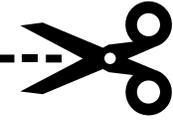
START	1.				3.
			2.		

- 1. CIRCLE
- 2. TRIANGLE
- 3. SQUARE

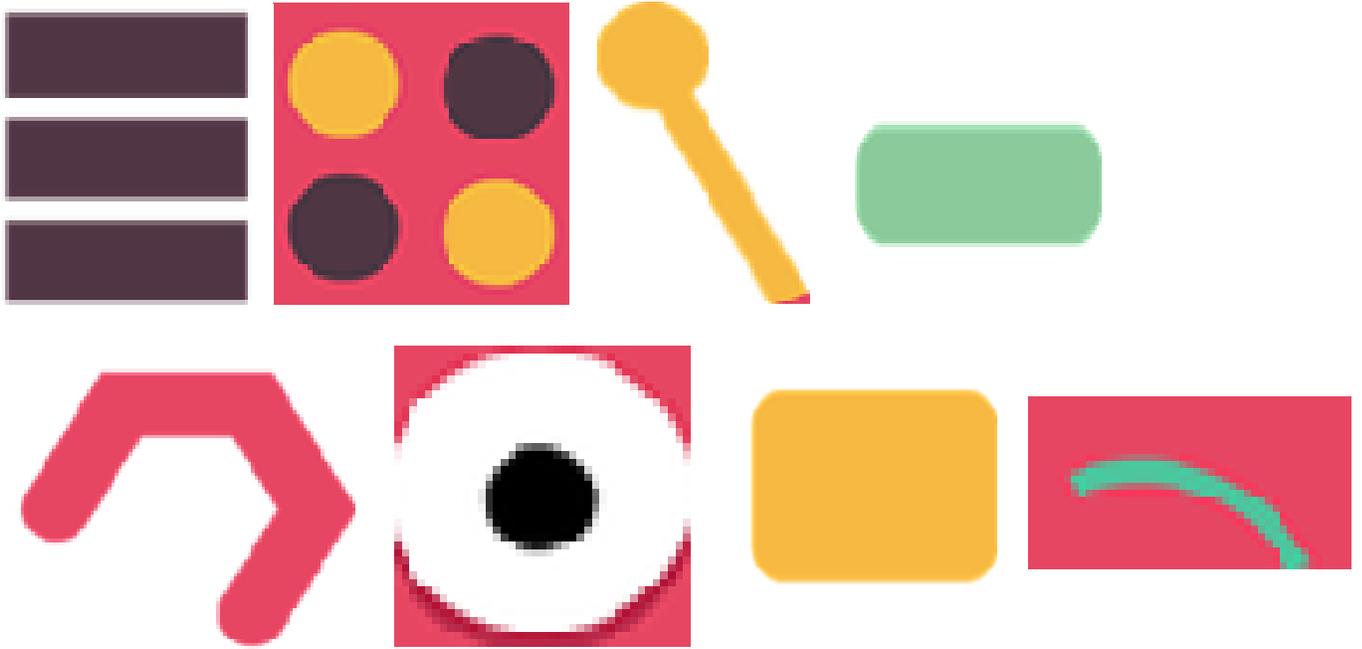
Template 1



1	
2	
3	
4	
5	
6	
7	
8	

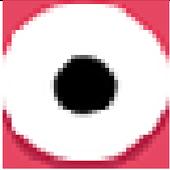


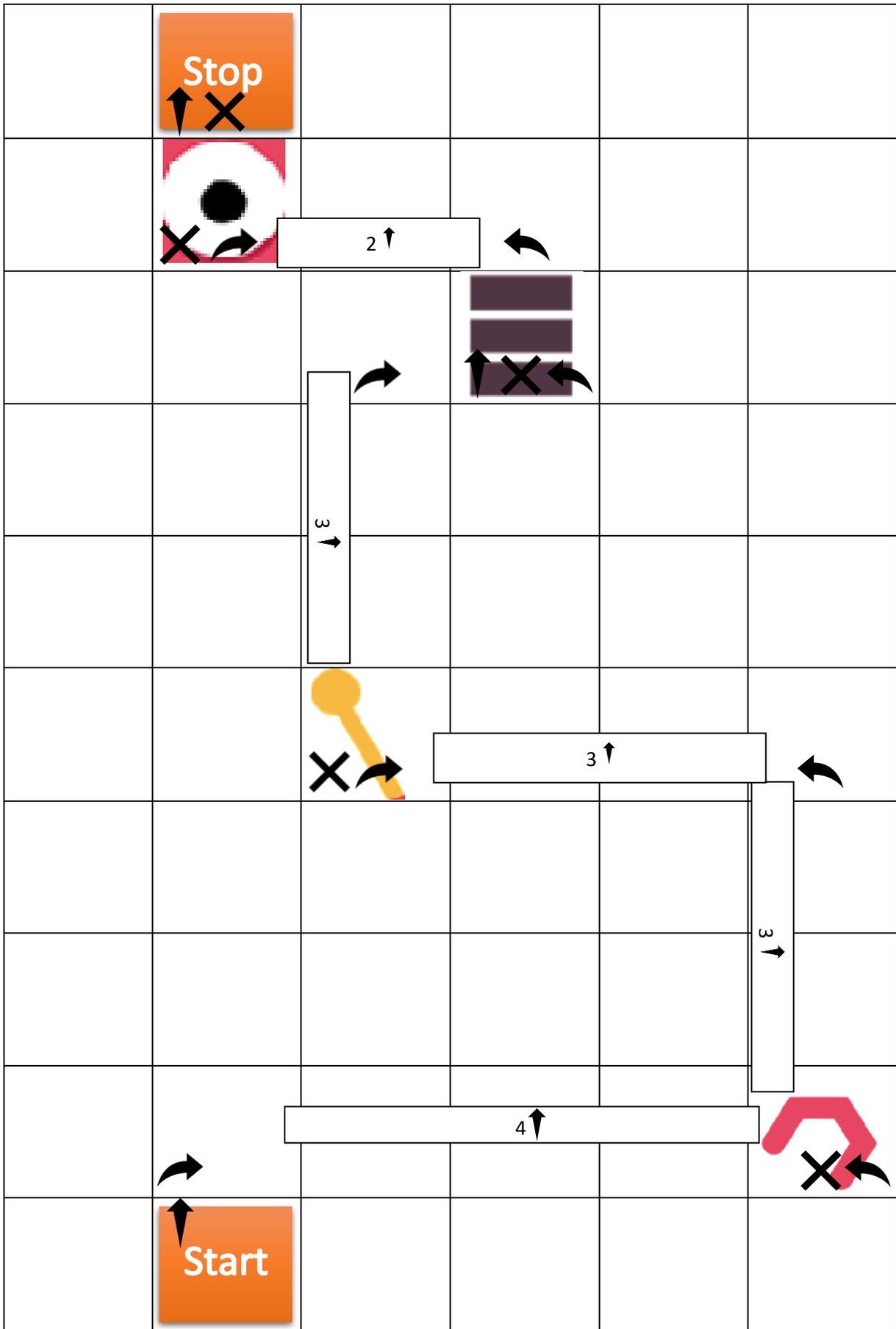
Template 2



Template 3

	Go forward.
	Go left.
	Go right.
	Stop / Robot Piece
	Repeat

	Stop				
					
					
					
					
	Start				
   			Min: Max:		
Possible code:					

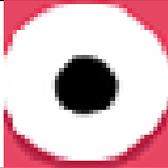


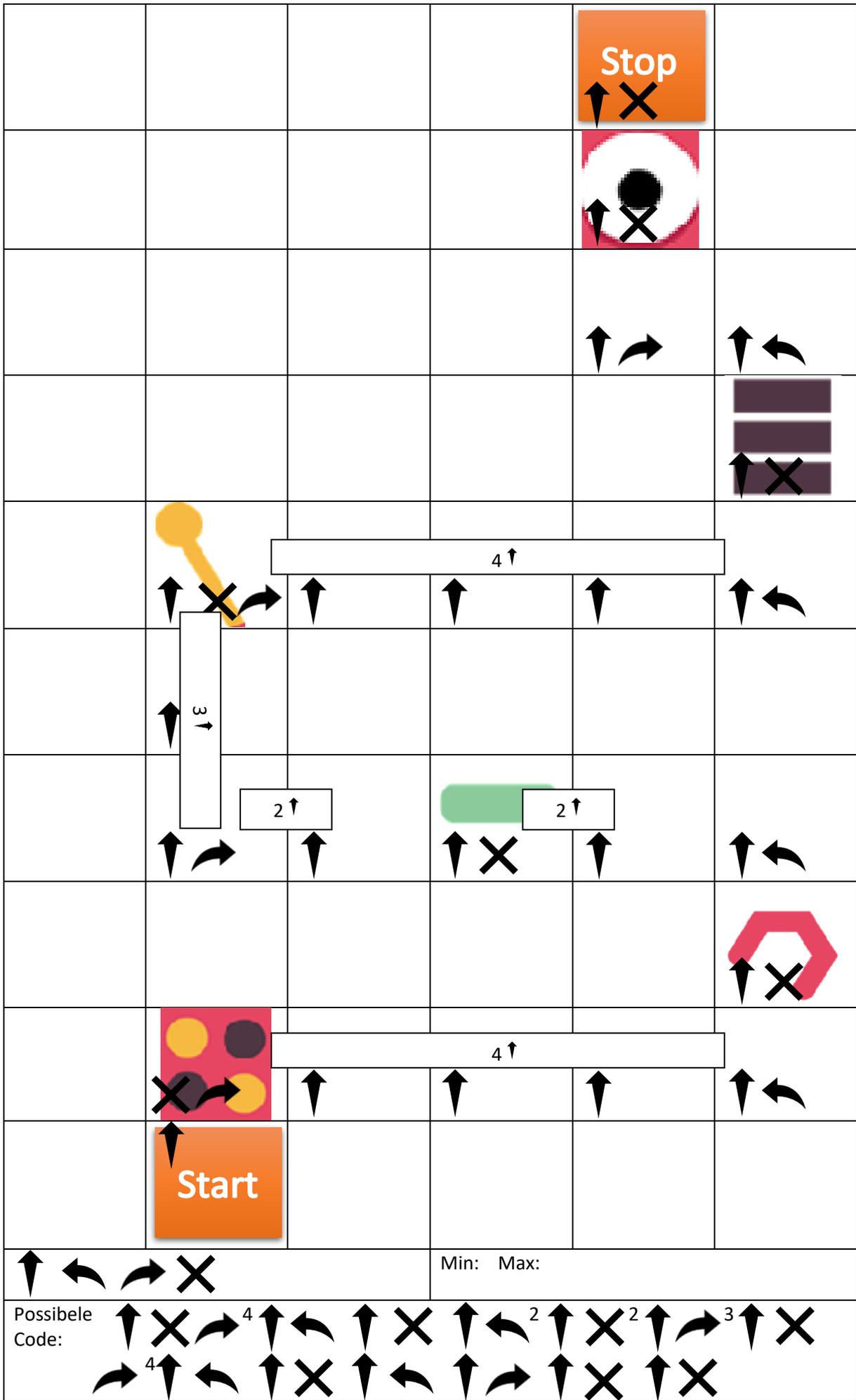
↑ ← → X

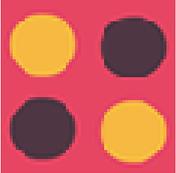
Min: Max:

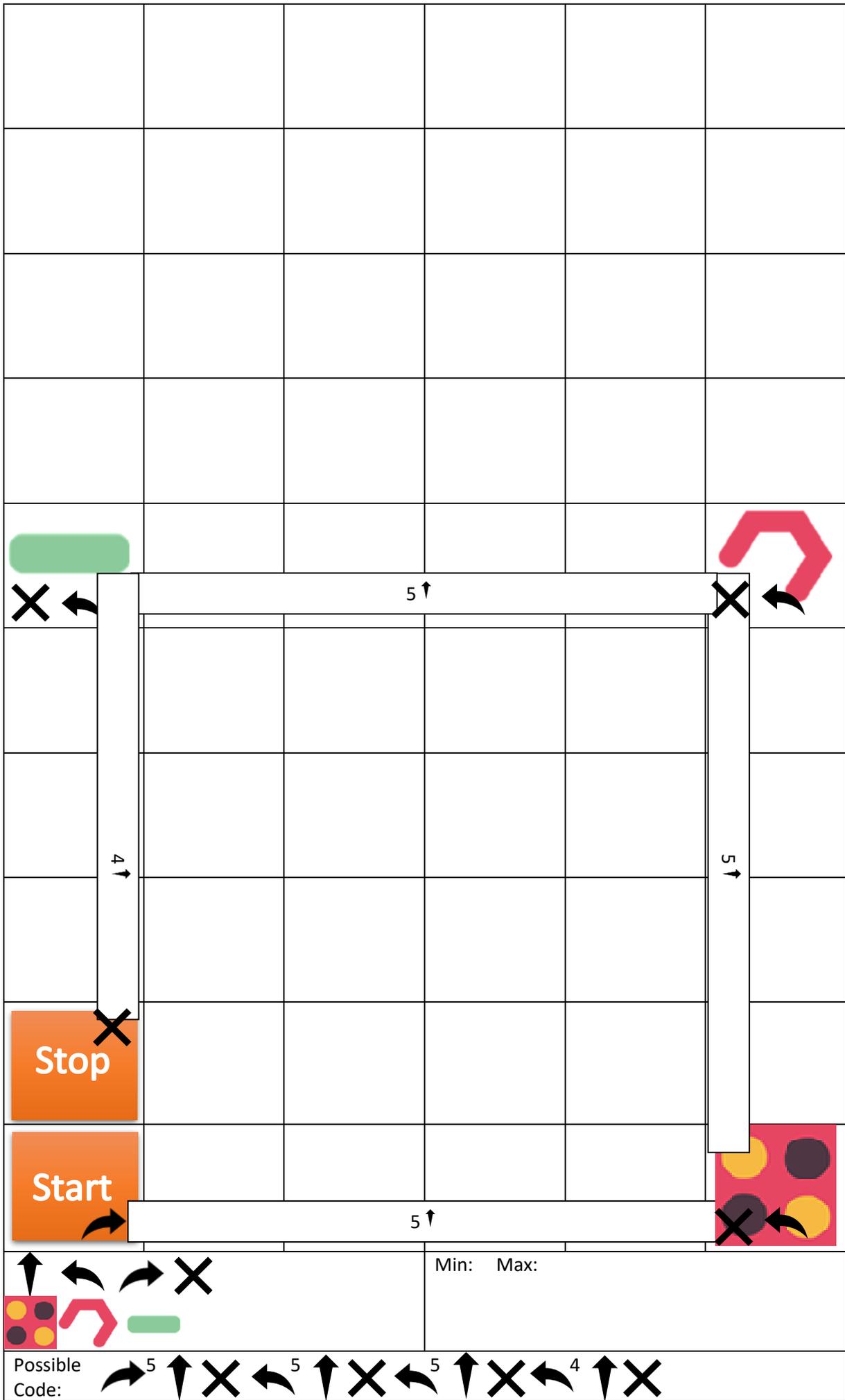
Possible code:

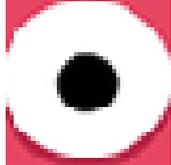
↑ →⁴ ↑ X ←³ ↑ ←³ ↑ X →³ ↑ → ↑ X ← ↑
 ←² ↑ X → ↑ X

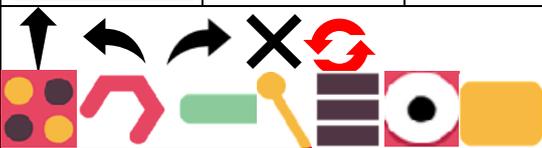
				Stop	
					
					
					
					
					
					
	Start				
   			Min: Max:		
Possible Code:					



					
Stop					
Start 					
      	Min: Max:				
Possible Code:					

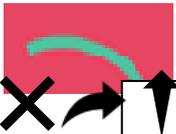
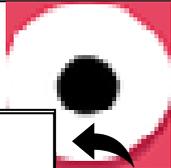
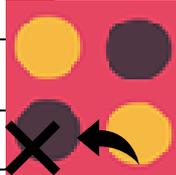


Stop					
					
					
					
Start					



Min: Max:

Possible
Code:

																								
																								
																								
																								
																								
																								
																								
																								
																								
																								
					Min: Max:																			
																								
Possible Code:		2		(5				2				5				2							

