ROBOT GAMES (LEVEL 1)

Ages 4 to 7 (Level 1)

Description:	Learners will the basic concepts of programming such as sequencing and repetition. They will then apply the concepts to solve challenges on a maze.
Leading question:	How can you code a robot to perform activities and win puzzles?
Age group:	4 to 7 years
Subjects:	Computer Science, Literacy, Numeracy
Total time required:	1 hour and 15 mins a day for 5 days
Self-guided / Supervised activity:	Supervised by parents / guardians
Resources required:	Paper, pen, pencils, small toys

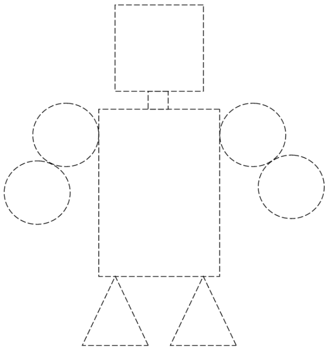
Day	Time	Activity and Description
1	15 minutes	**Day 1: Learning about sequencing** Learning about steps and creating instructions Learners will be asked if they know how computers and phones work. Explain to the learner that most electronics and programs work by following **code** – a series of exact instructions, that are followed step-by-step. Use walking to explain code: in order to walk forward, ask them what are the steps? *Explain that in order to walk, they need to follow the following steps:* • *Lift one foot,* • *Move that foot forward* • *Place their foot down* • *Lift their other foot,* • *Move it forward,* • *Place the other foot down* Ask the learner what happens when they miss a step? They get stuck or fall – the same thing happens with code! Ask the learner to try break down another regular task: picking up something, turning a page, etc. *Try to break down the steps to another task, such as a popular dance routine; for example, the Hokey Pokey needs the following steps:* • *Put right foot in,* • *Pull right foot back,* • *Put right foot in,* • *Shake right foot around,*

- *Put right foot out,*
- *Do the hokey pokey*
- *Turn around in a circle.*

Explain that they will be learning to program their own robot and using it to solve puzzles!

30 minutes

Ask the learner if they've seen any robots recently – ask them to list some of the features of a robot they know of.



Next, get the learner to design and draw their own robot – they can decorate their robot as they like.

Numeracy activity: As a first draft of their robot, have the learn sketch out a design using shapes that they already know of – shown on the left.

Tip: The robots they design can be as simple as a box that they decorate,
cubes stacked on top of each other or a 2D cutout (see Appendix 2 for sample robots).

Learners will now create a grid of 5 by 5 squares, an example is attached below – (a sample is attached under Appendix 1). The grid has to be large enough to accommodate the robot to stand in a single space

Additionally, cut out 2 or 3 other squares and mark them differently (using either different colored paper or coloring them)

Pick one square to be a start square, draw a green arrow on it
Pick another square to be the end and draw a red circle on it
Place an object or a toy as a "prize" on any square or use a colored square

Tip: You could also use any existing space: tiles on the floor or mark out a space in sand/mud and mark the start/end square with a different colored paper/toy

Here is an example of what a set up could look like this:

⬇				
		Prize		
				●

	10 minutes	Explain to the learner that they have to list all of the commands to: 1. Move from the start 2. Pick up the object 3. Move to the finish For their first try, help the learner “code” by making a list of the commands. *For the sample set up, the code will could be:* • *Start* • *Move forward* • *Move forward* • *Turn right* • *Move forward* • *Move forward* • *Pick up toy* • *Move forward* • *Move forward* • *Turn right* • *Move forward* • *End* Have the learner try different versions of the code – if their robot falls out of the grid, they’ve “broken” the code
	20 minutes	Move the prize around and have the learner write code to reach the end point to reach and pick up the toy
2		**Day 2: Counting** Today the learner will be taught to shorten code by learning repeating functions
	15 minutes	**Literacy activity:** Have the learner think about the different tasks that are already done by robots and how they make their lives easier – what tasks would the learner like to have a robot do for them?
	10 minutes	Ask the learner what they realized about the coding activity yesterday Ask if they realized that they had to repeat a number steps several times? Explain the concept of “**counting**” in code: that a code can be repeated for a set number of times. Example: move forward for 2 steps
	20 minutes	Use the example set up (from the previous day) and explain that counting could accomplish the same goal **Original code:** • *Start* **New Code:** • *Start*

		• *Move forward* • *Move forward* • *Turn right* • *Move forward* • *Move forward* • *Pick up toy* • *Move forward* • *Move forward* • *Turn right* • *Move forward* • *End* • *Repeat 2 times: move forward* • *Turn right* • *Repeat 2 times: move forward* • *Pick up toy* • *Repeat 2 times: move forward* • *Turn right* • *Move forward* • *End*
	30 minutes	Set up the grid with the prize in any chosen square and have the learner go from start to finish while picking up the prize.
3		**Day 3: Make your own functions** Today the learner will be encouraged to improve their robot by creating their own functions.
	15 Minutes	Explain that a function is a group of code, so for example: Walk: • *Lift one foot,* • *Move that foot forward* • *Place their foot down* • *Lift their other foot,* • *Move it forward,* • *Place the other foot down* So now, every time you need the robot to move, you can use the Walk function, instead of writing 7 lines of code – which makes it easier to code!
	15 minutes	Ask the learner to come up with their own functions for their robot navigate the grid *Encourage them to think of functions that would help them navigate the grid faster, such as "Jump 2" or "Turn Around" – let them get creative!*
	30 minutes	Set up the grid again and have the learners use their own functions to pick up the prize and navigate the grid
	15 minutes	**Literacy activity:** Have the learner make a list of the activities they perform in their daily lives. Can they convert those activities into functions?
4		**Day 4: New Obstacles**
	10 minutes	Introduce new obstacles to the grid such as walls or holes or cracks that would affect the movement of robot around the grid

		Tip: The grid can be expanded in size to accommodate more obstacles.
	15 minutes	Work with the learner to decide on what new functions they would need to navigate and help them create new functions
	35 minutes	Reset the grid with the new obstacles and the prize and have the learner navigate the code, make it an obstacle course they have to solve. For every time the learner gets the code right, either increase the number of prizes or obstacles to make the levels harder
5		**Day 5: Mission: Robot** Today the learner will use their new coding skills to create a story and a mission for their robot to complete
	10 minutes	Together with the learner, create a story around the robot having to solve a mission. *For example:* • *The grid contains items causing pollution, such as trash, plastic bags, etc which need to be picked up to clean up the space.* • *The learner creates a mission to find a gem/key – they need to find the object (which can be hidden under an obstacle) in order to open/unlock another obstacle and get to the end, etc*
	5 minutes	Help the learner design and set up their own obstacles/prizes/boosters *These can include:* • *small boxes to hide prizes under* • *A booster could special function that helps them overcome the grid (such as "break rocks", "melt obstacle", etc.*
	45 Minutes	Without the learner looking, set up the grid including the new obstacles, prizes and boosters Have the learner write the code to solve the mission and keep track of the different attempts *This might take several tries* Once they have successfully solved the mission, have them build a story around the mission, including how they had to restart the mission, what new boosters they found on the way, etc.

Assessment	Short and clear code is the main assessment. Once the learner has understood functions, assess how "efficient", their code is – the more functions they develop, the shorter their code (in length) should be in order to achieve their goal.
Learning outcomes:	- Logical thinking and problem solving - Problem solving

Required previous learning:	None
Modifications for simplification	The learners can convert the whole project into a game where they are the robot themselves – making it one where physical activity is encouraged. They can then play the grid, where they jump over obstacles.

APPENDIX 1 – SAMPLE GRID

EAA welcomes feedback on its projects in order to improve, please use this link: https://forms.gle/LGAP9k17fMyJrKJN7

APPENDIX 2 – SAMPLE ROBOTS