INTRODUCING PROGRAMMING LANGUAGES VIA ROBOT GAME (OFFLINE)

DIGITAL CONTENT CREATION > 3.4 PROGRAMMING

TARGET GROUP	AGE GROUP	PROFICIENCY LEVEL	FORMAT	COPYRIGHT	LANGUAGE
School drop outs, Students (primary school), Students (secondary school)	Children, Teenagers	Level 1	Activity sheet	Creative Commons (BY-SA)	English, French

This workshop develops notions of programming language and is an extension of a previous robot game-based activity.

General Objective	Knowledge acquisition
Preparation time for facilitator	less than 1 hour
Competence area	3 - Digital content creation
Time needed to complete activity (for learner)	0 - 1 hour
Name of author	Anissa Bouchareb - BSF Belgique
Support material needed for training	Materials to draw a grid on the ground (rope, ribbon, chalk, etc.) Materials to place objects on the grid (real objects, chalk to draw, etc.)
Resource originally created in	French

WORKSHOP DIRECTIONS

1 Introduction

This activity is designed to introduce, in an offline way, notions of programming languages. It is an extension of 'Introduction to Programming via Robot Game (Offline)', and is meant for children or parent-children groups who do not have experience in programming.

2 Preparation

Familiarise yourself with the ideas below, as well as the activity 'Introduction to Programming via Robot Game (Offline)'. It is best to have already followed this workshop in advance (as much for yourself as for the group). Print the instructions that will serve as the basic language to be used by participants, for example from images of arrows or of Scratch blocks. It is possible to refer to material often used in robotics workshops (Dash, Thymio).

3 Some theory

Computers cannot understand human language. They understand only **binary** (a way of presenting information in 1s and 0s using electricity). Humans can 'speak' binary only with difficulty and impracticality. We have therefore invented **programming languages**, easily manipulated by humans, which are translated to binary to be understood by computers. A programming language is therefore a vehicle for **linking** between the programmer (person who writes programs) and the computer.

Through a programming language, orders – **instructions** – can be given to the computer to tell it what to do (play a video, open a website, etc.) and how to react (to a mouse click, to a keypress, etc.). Human languages have evolved through practice, i.e. in the creation of new words, the disappearance of obsolete words, the mixing of languages, etc. As for programming languages, they have evolved through technological **progress**.

Different IT languages have **different** results: managing internet sites, making smartphone apps, controlling robots, etc. In general, programming languages use English words to express simple precise

things. The closer a programming language is to binary, the more it is known as '**low level**'.
The further away from binary a language is on the other hand, the more it is known as '**high level**'. In Scratch, we can choose the human language in which we will program, therefore we use high level language when using Scratch.

4 Some examples of languages

Human languages are grouped by 'families': Germanic languages, Romance languages, etc. Their categorisation depends on their origins and their common characteristics. Programming languages also have 'families' based on **paradigms**: their methodologies allowing them to program using different styles. Scratch: a **block-based language** designed to introduce the audience to programming.
It is a simple and practical way to start with programming.

HTML : a **tag-based language** used to create webpages via hypertext.

Javascript : an **object-oriented language** which can render dynamic webpages (with animations, games, applications, etc.)
Python: also object-oriented, it is very versatile and popular, including in the domain of education.
C : a **procedural** low-level language for manipulating electronic components. It is used in many devices from dishwashers to satellites to cars, robots and aircraft.
C++ : the object-oriented evolution of C, like Python, it can do many things and is used everywhere from business management to video games.

5 Robot game extension

Host a discussion with guided questions, or another method of your choice, focusing on the **notions** above.
The most important thing to get across is that a programming language is a way of addressing a computer without needing to 'speak' binary, and that there are different types.
Explain that now, the group will be able to use a simple language to program the robot game. Keep the map you used for the robot game.
Add the printed instructions and review them with the group to make sure everyone understands the arrows/blocks/movements in the same way (keep in mind: 'pivot' and 'turn' don't mean the same thing).
As with the base game, divide participants into pairs. One will play the role of 'programmer' and the

other the robot who executes the program.

The different robots should start at different departure points. Give the pretext and the robot's objective depending on the assigned route and the objects/drawings set in place (e.g. go to the chest and take the treasure). The 'programmers' will to use the **instruction** sheet, without speaking, to communicate their commands to the robots in a clear way and in the chosen programming language.

For the first round, programmers will show **one sign at a time** and the robots will execute their orders immediately. When everyone understands the game, repeat it, this time using groups of three: two programmers and one robot. The programmers discuss which instructions to choose and place them on the ground in order. When they think their **program** is ready, they read the instructions to the robot who will execute them in order. If an error occurs, the robot goes back to the start and the programmers will need to correct their program.