

A cat, a king and a robot: programming with blocks

Alberto BARBERO

Istituto Istruzione Superiore “G.Vallauri”, via San Michele 68, Fossano (CN) - Italy

alberto.barbero@vallauri.edu

Abstract

In Italian secondary schools, the guidelines for the first year include the European Computer Driving License (ECDL) syllabus to ensure a basic common level of digital competencies. The approach to Informatics in Italian secondary schools is being reshaped after the Reform of 2010. The new framework allows the introduction of computer logic and basic programming concepts beginning in the first and second grades. But what is the best way to introduce computer logic and basic programming concepts to teenage students and which tools should be provided to them in order to capture their interest and attention? Help is provided from “Scratch” a free programming language developed by researchers at M.I.T. MediaLab in Boston. Scratch suits the curriculum that teachers sought because it was specifically created for introducing the basic concepts of problem solving and programming to young and inexperienced students. Scratch, in its free extension “S4A” (Scratch for Arduino) developed by Citilab in Barcelona, also allows the introduction of the basic concepts of Robotics through the use of the open board called Arduino. The Arduino board is important because its use, combined with the power of S4A, allows for easy experimentation managing digital and analogical signals and sensors and actuators connected on the board. Another aid in the achievement of our goal is provided by “App Inventor” a Google tool now managed by the same group at the M.I.T. MediaLab. It easily allows the introduction of Android mobile programming to the students using almost the same approach as Scratch.

Introduction

In Italian secondary schools, the guidelines for the first year include the European Computer Driving License (ECDL) syllabus due to very different levels of student familiarity with computers and computer science. Most of the schools find it appropriate to dedicate the entire curriculum to the ECDL certification to ensure a basic common level of digital competencies. The approach to Informatics in Italian secondary schools is being reshaped after the Reform of 2010. The new framework allows the introduction of computer logic and basic programming concepts beginning in the first and second grades. This has proven to be a very good opportunity to finally enhance and give dignity to a discipline such as Informatics without disguising it or transforming it into something else. But what is the best way to introduce computer logic and basic programming concepts to teenage students and which tools should be provided to them in order to capture their interest and attention?

Scratch: a cat and the art of programming

Help is provided from “Scratch” (@ scratch.mit.edu), a free programming language - featured by a cat icon - developed by researchers at M.I.T. MediaLab in Boston. Scratch suits the curriculum that teachers sought because it was specifically created for introducing the basic concepts of problem solving and programming to young and inexperienced students. It combines the simplicity of use with the formalities of artificial languages, all seasoned with a lot of imagination. In Scratch, programmers combining blocks of different colors and shapes, each corresponding to a specific primitive, can easily implement animations while simultaneously executing different processes, making them interact, exchanging messages and managing events.

For example, in the *Build a calculator* program, the student had to write a program managing the basic arithmetic operations, as shown in figure 1. Students discovered the use of simple variables, input/output instructions and events. The “set” command allows the program to execute assignment operations while “ask” and “say” commands manage inputs and outputs. By clicking on one of the

sprites (the icons the programmer puts on the stage - the window where sprites are shown) the corresponding set of block commands are executed doing the operation associated to the event “click”.

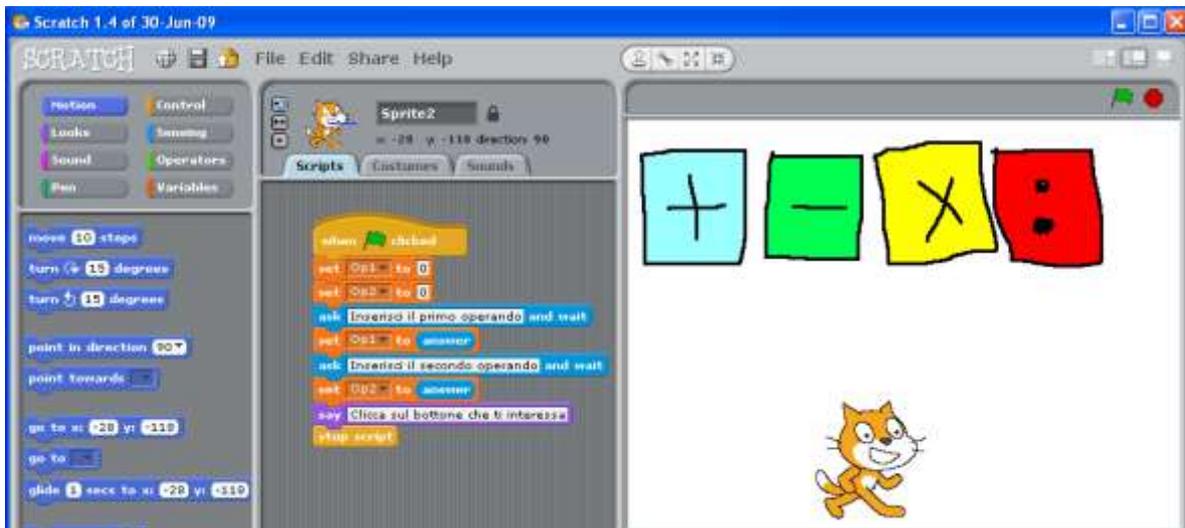


Fig. 1. Build a calculator program.

A second example is *High-low guess a number* program, which involves a cat and a dog sprites. The dog has to guess the number the cat is thinking of in a given range. The aim is to try to minimize the number of guesses the dog makes. Obviously the best strategy would be to implement a binary search type algorithm.



Fig. 2. High-low guess a number program.

This program, partly shown in figure 2, introduces the iteration method with the “repeat until” command, the possibility of communications by messages (broadcast command) between sprite processes and the binary search algorithm.

S4A: a king to role the board

Scratch, in its free extension “S4A” (Scratch for Arduino @ seaside.citilab.eu/scratch/arduino) developed by Citilab in Barcelona, also allows the introduction of the basic concepts of Robotics. This is done through the use of the open board called Arduino and all of its available sensors and actuators. The Arduino board, conceived in the Piedmont region (Arduino was a King of Italy), is important because its use, combined with the power of S4A, allows for easy experimentation with concrete programming and the ability to see the immediate effects of the code on something

“touchable”. S4A extends the Scratch IDE with new commands for programming digital and analogical signals and for managing sensors and actuators which are connected on the board through firmware dedicated to communication. These are some of the projects developed: a LED blinks with a variable delay; a LED turns on and off by pushing a button; a decimal number is converted in binary notation corresponding to LEDs on and off; the amount of light is detected using a light sensor; a sprite is moved on the stage by a joystick; the "Space Invaders" videogame is played using a slider and a button.

In order to easily manage the Arduino board, is possible to use the “Tinkerkit” (see figure 3), a shield that can be hooked up to the Arduino board to simplify the electronic connections to sensors and actuators.

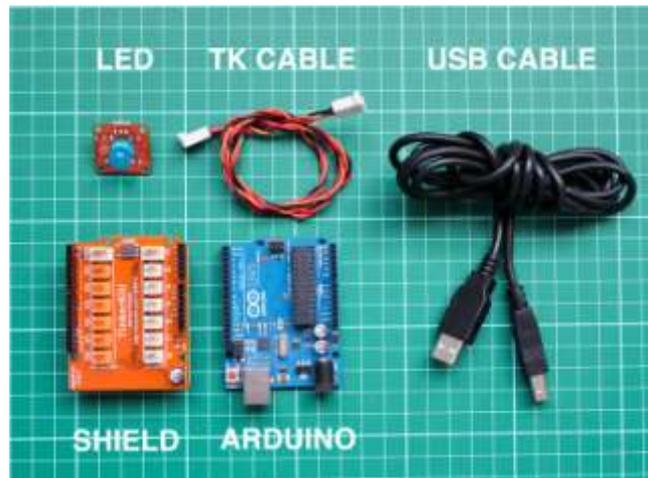


Fig. 3. A Tinkerkit shield with various other components (picture from www.tinkerkit.com).

The *LED on and off* S4A program, for example, shows how to make a blink effect by turning a LED on and off every half a second. The LED is connected to the Arduino board through the Tinkerkit shield. As you can see in figure 4, the S4A IDE offers in the block area new commands for programming the analogic/digital signals. Figure 5 shows the easy connection of a LED to the Tinkerkit shield using the proper cable.



Fig. 4. The S4A code for switching on and off a LED.

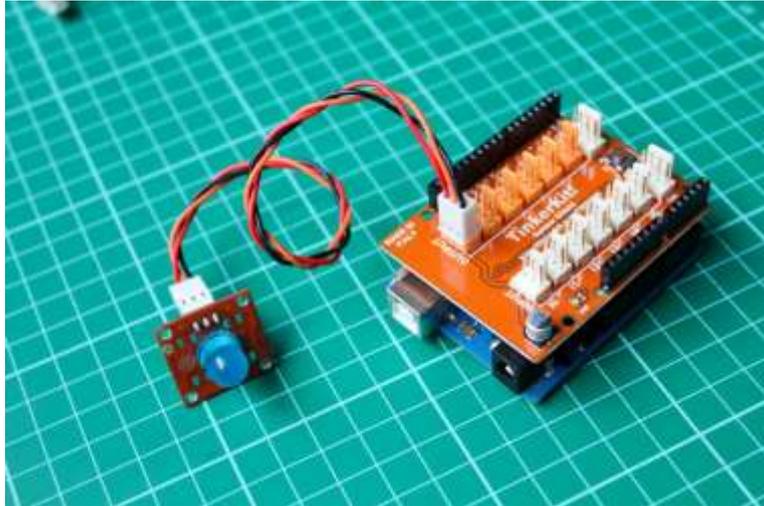


Fig. 5. Connection between a LED and the Tinkerkit (picture from www.tinkerkit.com).

App Inventor: a robot in your mobile

Another aid in the achievement of our goal is provided by “App Inventor” (@ appinventor.mit.edu), a Google tool now managed by the same group at the M.I.T. MediaLab. It easily allows the introduction of Android (the o.s. with the robot...) mobile programming to the students using almost the same approach as Scratch. App Inventor is a free programming language that utilizes the fitting together of blocks of different colors and shapes, like in a puzzle, and lets the student design how the app looks and behaves. App Inventor can be used to develop apps using the main smartphone’s and tablet’s sensors such as the touch function, the accelerometer, the location and orientation sensor, the speech recognizer, the text-to-speech function, the texting function, etc., combined with serious programming concepts. A live connection between computer and phone allows the app to appear on the phone. These are some of the projects developed: a paint-like app; videogames (mole mash, ladybug chase, find the pair, puzzle, tic-tac-toe, labyrinth); apps using Google maps service; a quiz app on the phone; apps using the texting function.

The *Tic tac toe* app shows the use of the touch function by managing the “touched” event provided by App Inventor. The first step in the App Inventor project is to design the mobile interface in the Viewer by selecting the components from the Palette and by setting the corresponding Properties (see Figure 6).



Fig. 6. The App Inventor Designer IDE.

The next step is to arrange the command blocks in the Blocks Editor to manage the events that occur to the components. Figure 7 shows the code associated to the “Touched” event on one of the 9 sprite boxes. The final step consists of launching the smartphone Emulator and seeing the execution of the app and finally uploading the program on the mobile device.

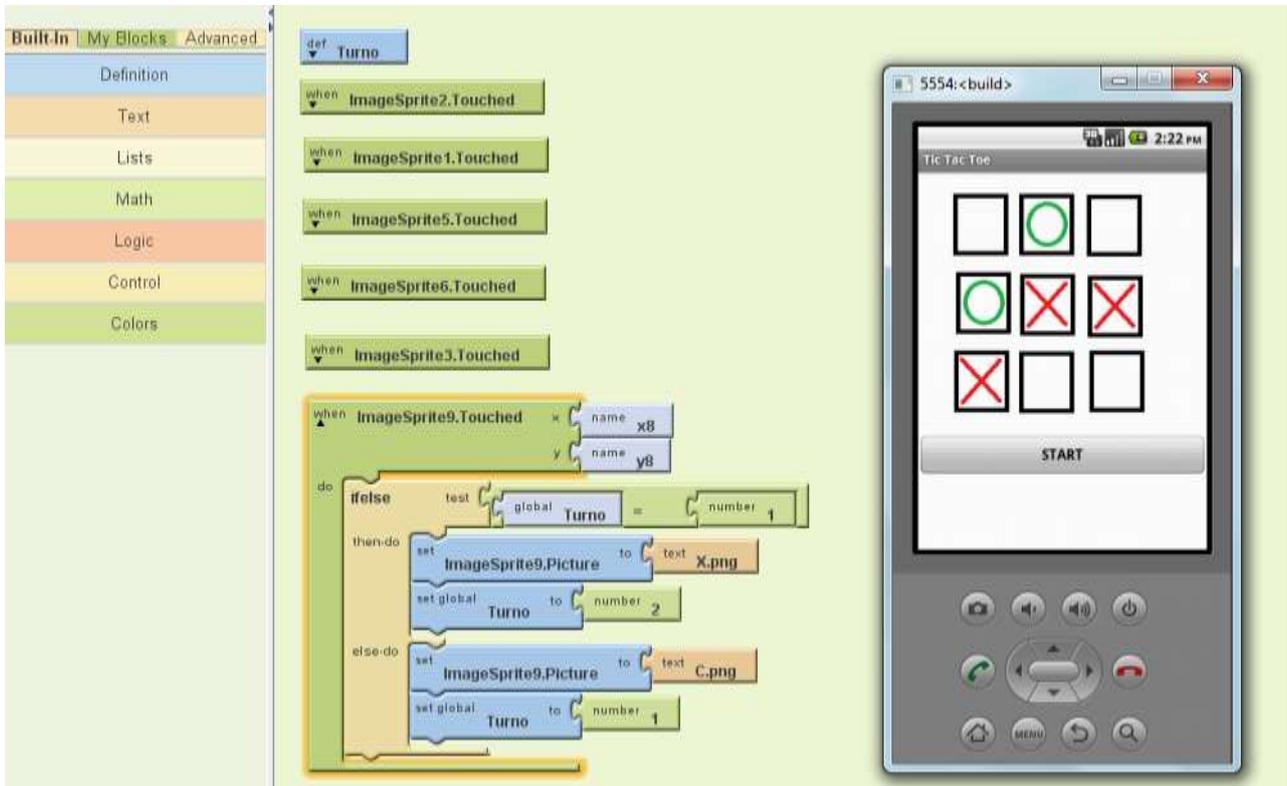


Fig. 7. The Block Editor and the Emulator

Acknowledgements

I would like to thank Professor Barbara Demo of the Department of Computer Science at the University of Torino for her valuable guidance, encouragement and help. I would also like to thank Professor Francesco Vaschetto for his help in the development of this curricula and Dott. Charles J. DeVirgilio for his translation to English assistance.