Improving the Thymio Visual Programming Language Experience through Augmented Reality

Stéphane Magnenat (ETH Zürich) and Francesco Mondada (EPFL)

March 24, 2014

Abstract

This document is a roadmap describing two directions for improving the user experience of the Thymio robot and its visual programming language using augmented reality techniques.

Recent progress in visual programming language (VPL) design [2] has shown that children can understand elements of a core computer science concept — event handling — even in the course of a 1.5 hour workshop while using a VPL together with the Thymio mobile robot [1]. However, that work shows that while students had a good comprehension of, and were able to implement, robotic tasks that were directly accessible and reactive, they struggled with tasks involving reasoning about different event-action sets and their interaction at different moments in the task. We believe that the reason is the difficulty of tracing the execution of the program while writing it, as mentioned in the literature [3]. However, to become a good programmer, one has to become fluent in this process, to the point of doing it unconsciously while programming. While the VPL for Thymio solves the problem of writing a program in the first place, it does not solve the problem of learning to imagine what this program will do beyond simple actions in isolation, when the different actions are executed at various times due to different environmental conditions. Indeed, VPL programs are composed of event-actions sets, for which all actions associated to an event are executed when the event fires, which happens asynchronously with respect to other events. This makes it very easy to create a simple behaviour in which the robot reacts to an object placed in front of it for instance, but quickly makes the creation of more complex behaviours involving timer, state and multiple elements of the world very difficult, as the programmer has to reason about the firing order of the different events.

In this document, we outline a roadmap to address this problem. It is articulated into two complementary approaches, that should be tested in parallel: making the behaviour of the robot live observable while creating the program, and visualizing the execution of the program to allow children to build a proper notional machine of the program-robot-environment system.

1 Making the behaviour live observable

This approach proposes to program the Thymio robot interactively. The current version of VPL has minimal support for this paradigm, for instance the background of the colour action block displays live the selected colour and the motor action block shows a small simulated animation of the robot corresponding to the selected wheel speeds. However, there is currently no way to see what the program will do before executing it on the real robot.

We propose to address this limitation by showing a simulated behaviour of the full robot in a virtual world while the student edits the VPL program. In this virtual world, the execution of the behaviour will run faster than real time, allowing the student to quickly have an idea of the effect of her program. Moreover, visual cues such as a transparent "shadow" left in the course of the robot path could help the student to track the behaviour of the robot. This idea is inspired by previous work from interaction designers [5, min. 15]. Moreover, an icon of the event-actions set could be left in the virtual world at the location the latter was executed.

The virtual world could be made from scratch in a computer graphics software, but it could also be acquired through augmented reality. This would be particularly interesting because it would allow the student to have a preview of the execution of the program in the same environment the real robot will execute that program, and allow the student to manipulate the environment while developing the program. This is important, because the execution of the program and the behaviour of the robot are highly dependant of the environment.

The technical solutions we envision for implementing the augmented reality are to use 3-D mapping devices such as structured-light sensors (Kinect or Google's project Tango) or to track known images and markers. The augmented environment will be displayed on a tablet which will localise itself in the real environment, allowing the student to concentrate her observation on an area of interest.

2 Visualizing the program execution

This approach proposes to first program the robot and then run the program, but allows the student to visualize (trace) the execution of the program through augmented reality. Using a tablet, the student will be able to see which event-actions sets are executed when and where in the environment. The visualization application will allow real-time inspection of the executed sets but also a post-mortem visualization with a scratchable time-line. Thanks to augmented reality, and by moving her tablet to locations of interests, the student will be able to concentrate on what happened at these specific locations. Seconded by a proper curriculum and teaching assistants, this system should enable students to engage with their program up to the level *Presenting-Own content* of the 2DET engagement taxonomy [4, Sect. 4.2].

This proposed visualization application is the robotics and augmented reality counterpart of a program visualizer for text programs. The technical solutions for implementing augmented reality are similar to the ones described in the previous section.

References

[1] Stéphane Magnenat, Jiwon Shin, Fanny Riedo, Roland Siegwart, and Morderchai Ben-Ari. Teaching a core cs concept through robotics. In *19th Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE)*, 2014. To be published.

[2] Jiwon Shin, Roland Siegwart, and Stéphane Magnenat. Visual programming language for Thymio II robot. In *Interaction Design and Children*, 2014. To be published.

[3] Juha Sorva. Notional machines and introductory programming education. *Trans. Comput. Educ.*, 13(2):1–31, Jul 2013.

[4] Juha Sorva, Ville Karavirta, and Lauri Malmi. A review of generic program visualization systems for introductory programming education. *Trans. Comput. Educ.*, 13(4):1–64, Nov 2013.

[5] Bret Victor. Inventing on principle. `http://vimeo.com/36579366`.