ISBN: 978-0-620-45997-6 TEDC 2010 Proceedings - Technology for Innovation and Education in Developing Countries.

Dialando: Tangible Programming for the Novice with Scratch, Processing and Arduino

Andrew Cyrus Smith
CSIR Meraka Institute, PO Box 395, Pretoria, 0001, South Africa
acsmith @ csir.co.za

Abstract

This paper reports on a tangible programming system designed for the novice user in developing regions. The system integrates three open source tools with low cost hand-made hardware incorporating recycled material. The result is a simplistic system that a novice can use to code a five-step sequence using an instruction set of four commands. Using a single PC, and multiple instances of the hardware described here, numerous novice programmers may potentially make use of the same PC simultaneously.

1. Introduction

The personal computer (PC) is a technological marvel, so much so that it has contributed to the digital divide [1]. The PC is not simple to use, it has been blessed by engineers and software developers with wide ranging capabilities in the form of software and hardware. In some societies children are exposed to this complex machine from a young age. These children have no problem in adjusting to this technology and making full use of the capabilities encased in the dull box which is the PC. However, a substantial number of children in developing regions do not have the luxury of having access to a PC from a young age. When first exposed to this technology, these prospective users are overwhelmed by the techno-speak and abstract mental models required to comprehend this machine. It is no wonder that the current design of both the machine, and the applications that execute on it, contribute to the digital divide. This "Swiss Army Knife", which we call the PC has been designed to do many things. But does it have real use if the novice is repelled by the mental load required to do so?

Perhaps what the novice computer programmer requires is a dedicated "appliance" [2] that reduces this complex technology to something which only addresses the current need of the user. For the purpose of this paper, that need is for a novice to gain exposure to the logical thinking required in computer programming (Figure 1).

Figure 1. The Dialando programming system in use at a conference demonstration session. The unit at the bottom contains the Arduino controlling circuitry which is connected to the laptop computer using a USB cable.

2. Our Approach

Our research aims to provide the novice user a gentle introduction to the abstract programming abilities of the PC. In order to achieve this, our approach is to first introduce programming concepts by making use of both concrete inputs and outputs. With such an approach the user can physically manipulate the inputs to a programme and observe the results as a physical manifestation in the real world. Through the initial introduction to programming using concrete

inputs and outputs (concrete-concrete interaction), the novice can progress to more abstract concepts [3, p89] such as required for using on-screen input and output (abstract-abstract interaction) in conventional programming environments. An optional concrete-abstract stage can be introduced between these concrete-concrete and abstract-abstract interaction approaches. Such a three-step approach supports a novice to gracefully become a computer programmer [4, p70].

In this paper we report on a simplistic concrete-abstract system which consists of a combination of recycled materials and electronic circuitry which interfaces to a PC. Using this system, the novice programmer manipulates tangible inputs and observes the results on the PC screen (Figure 2).

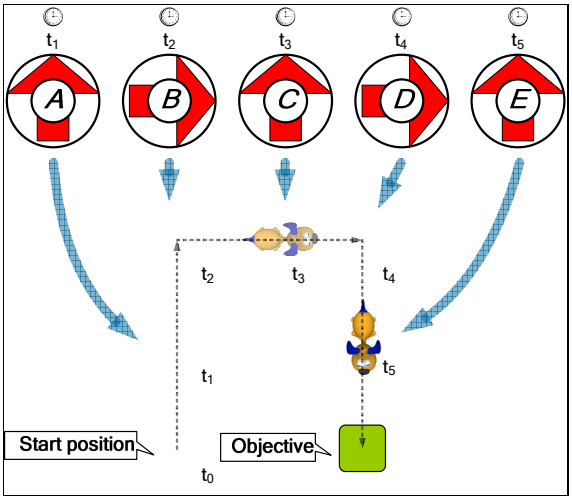

Figure 2. The mapping between the tangible instruction objects and the on-screen execution is shown in this example programme. Each instruction object [*A*…*E*] is interpreted sequentially at times [t_1…t_5]. At these instances in time, the on-screen avatar executes corresponding movements.

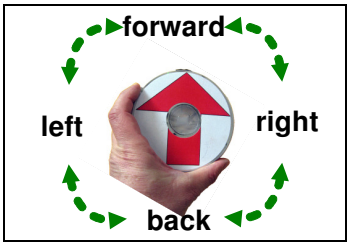

Figure 3. Four discrete directions are selected by simply rotating the programming disk while it is in position on the programming tray.

3. Prior Work

Other low-cost systems, which help the novice user in developing regions understand fundamental programming principles, have previously been reported. These include GameBlocks (both acrylic-based [5] and foam-based [6] versions) and RockBlocks [7].

4. Design Considerations

Dialando is a derivative of the words “dial”, “and”, “do” which means to dial the action required and the avatar does it – subsequently “dial-and-do” came to be known as Dialando.

“Dialling” an action is accomplished by rotating the disk to align with one of the four discrete directions imprinted on the programming surface (Figure 3). These directions correspond with the four actions (move forward, move backward, turn left, turn right) which the avatar is able to execute. The action of “dialling” an action increases the richness of the user’s interaction with the system [8, p11].

Dialando is also the name of the avatar (a puppy) which executes the instructions on the screen.

To the novice programmer the Dialando system seems to consist of only three component types. The first of these are the five tangible input disks which the user rotates. The second type is the set of five programming surfaces with the embedded sensors. Included in this second component type is the encased controller which houses an open source Arduino board [9]. The user need not be aware of the contents of this box which is stringed at the end of the five programming surfaces. To the user this box simply seems like a termination of the other five. The third component as it appears to the user is the PC.

Once the PC has been started and the five programming surfaces connected to it via a single USB connection, the required application is executed on the PC. These activities are executed by a computer literate person. Now the system can be used by the novice programmer whose only interaction with the PC is in using the tangible programming disks.

This interface to the application executing on the PC removes the novice from the many frustrating error messages and other problems [10, p28,30] a programmer can expect when making use of an abstract programming environment using a keyboard and mouse to enter programming code.

From our experience in using GameBlocks and RockBlocks at numerous science workshops with children, it became clear that aligning the object with

the reading surface is a recurring problem. Our first GameBlocks implementation made use of mechanical alignment mechanisms and eliminated this problem. However, as we tested new designs with the foam version of GameBlocks and later with RockBlocks, we neglected this important aspect of the design.

5. Implementation

We subsequently sought a reliable means of aligning the programming object with the programming tray. We still aimed for a method that does not require electrical contacts or the use of expensive sensing circuitry.

Similar to our previous systems, the solution we implemented retained the use of reed switches that sense the presence of a magnet. The combination of magnets in close proximity to the reed switched would uniquely represent the programme instruction required.

In addition to the encoding magnet/reed switch pairs, we added strong rare earth magnets in the centre of both the programming trays and the programming disks. This provided a means of aligning the disk with the tray without effort on the part of the user (Figure 4). With this configuration the user was free to simply rotate the disk in the required direction without having to re-align the disk to the tray.

Two magnets are embedded in each of the disks in such a way that four unique rotation angles can be sensed with the accompanying trays. Each tray has three reed switched embedded that coincide with the outer rim of a disk when placed onto the tray.

Each programming disk is comprised of two discarded compact disks (CD) and a short length of discarded thick electrical cable to encase the magnets. The electrical cable is simply used for aesthetic reasons to round off the outer rim of the programming disk assembly.

The programming trays are constructed from discarded plastic DVD cases. All the components in both the programming trays and programming disks are held together with hot glue.

The Dialando interface design is based on the rotation of a disk. The disk which has been placed on top of a sensing surface represents the intended direction the on-screen avatar is to move on the screen. The sensing surface contains three magnetic sensors. The sensors are part of low-cost intrusion detection circuitry commonly found in Western homes. Resistors are connected across each reed switch, resulting in changing resistance as the switches are opened and closed as a magnet approaches it. The switches are physically configured to detect four valid orientations the programming disk can take. These orientations are Forward, Backward, Left, and Right.

The Forward and Backward instructions result in the avatar covering a fixed linear distance on the computer screen. The Right and Left instructions result in 90 degrees avatar rotations on the screen in the respective direction. The avatar moves according to its own reference system, just as would the on-screen turtle of Papert's Logo. We found that this makes programming a non-trivial task as the programmer has to keep re-orientation herself as the avatar executed the instructions.

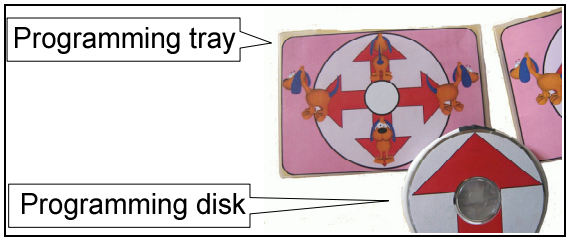

Figure 4. The Dialando programming system consists of five programming disks that are placed on top of five programming trays.

6. Conclusions and future work

Our research in new technologies for developing regions relies on the testing of novel concepts. We often have to integrate custom built hardware and software. Because our target beneficiaries are often from poor communities, this should be done easily and at low cost. The open source tools (Arduino, Processing [11], and Scratch [12]) we used in this particular research project made the integration between hardware and software almost a trivial task (Figure 5).

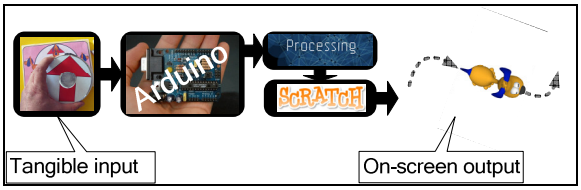

Figure 5. System overview, showing the open source components used.

Although the USB interface between hardware and software holds the promise of simply adding another hardware interface as required, we have yet to confirm that. Our aim is to confirm that multiple instances of the Dialando programming hardware subsystem can be interfaced to a single PC. If this can be achieved then multiple groups of programming novices may share the same PC simultaneously. This would expose more

children to programming without an excessive increase in the related education budget.

7. Acknowledgements

This research was funded by the South African Department of Science and Technology. Chris Krause provided much inspiration to this research and many technical suggestions.

8. References

[1] Baskaran, A. and Muchie, M., Bridging the digital divide: innovation systems for ICT in Brazil, China, India, Thailand and Southern Africa, Adonis & Abbey Publishers Ltd, 2006.

[2] Norman, D.A., The Invisible Computer: Why Good Products Can Fail, the Personal Computer is so Complex, and Information Appliances are the Solution, MIT Press, 1999.

[3] Vygotsky, L.S., Mind in Society: The Development of Higher Psychological Processes, Harvard University Press, London, 1978.

[4] Shneiderman, B., Leonardo's Laptop: Human Needs and the New Computing Technologies, MIT Press, Cambridge, 2003.

[5] Smith, A. C., Tangible cubes as programming objects. In Proc. ICAT 2006, IEEE Computer Society, 2006.

[6] Smith, A.C., In Proc Conference of the Southern African Computer Lecturers' Association, Eastern Cape, 2009.

[7] A.C. Smith, Handcrafted Physical Syntax Elements for Illetterate Children: Initial Concepts, In Proc IDC '08, ACM Press, 2008.

[8] Overbeeke, K., Djajadiningrat, T., Hummels, C., Wensveen, S., and Frens, J., Let's Make Things Engaging, In Funology: From Usability to Enjoyment, (Eds) Blythe, M.A., Overbeek, K., Monk, A.F., and Wright, P.C., Kluwer Academic Publishers, Dordrecht, 2003.

[9] Mellis, D., Banzi, M., Cuartielles, D., and Igoe, T. Arduino: An Open Electronic Prototyping Platform. Presented at alt. CHI 2007.

[10] Jones, M., and Marsden, G., Mobile Interaction Design, John Wiley and Sons, England, 2006.

[11] Reas, C., Fry, B., Processing: A Programming Handbook for Visual Designers and Artists, MIT Press, 2007.

[12] Maloney, J., Burd, L., Kafai, Y., Rusk, N., Silverman, B., and Resnick, M., Scratch: A Sneak Preview, In Proc International Conference on Creating, Connecting and Collaborating through Computing, IEEE Computer Society, Washington DC, 2004.

