IST-Africa 2009 Conference Proceedings
Paul Cunningham and Miriam Cunningham (Eds)
IIMC International Information Management Corporation, 2009
ISBN: 978-1-905824-11-3

Hand-Crafted Programming Objects and Visual Perception

Andrew Cyrus SMITH
CSIR Meraka Institute, Building 43, PO Box 395, Pretoria , 0001, South Africa
Tel: +27 12 8414626, Fax: +27 12 8414720, Email: acsmith@csir.co.za

Abstract: Novice programmers on the fringe of mainstream computer programming face a number of obstacles such as computer literacy and mastering the programming language syntax. Two programming environments currently available are briefly discussed as examples of attempts at assisting the novice programmer. These environments address the obstacle of the programming syntax, but not of computer literacy. However, a tangible programming environment can help the novice programmer overcome the computer literacy obstacle. Good visual perceptual skills is one of the requirements to be a successful tangible programmer. We present a quantitative test developed for measuring certain aspects of visual perception skills as appropriate to tangible programming. The results of a pilot test are given.

Keywords: children, physical programming environment, tangible learning technologies, visual perception.

1. Introduction

The electronic computer has its roots in machines developed for war, specifically World War II [1]. It was designed by scientists for use by scientists, to accomplish very specific objectives. More than 60 years have elapsed and the Personal Computer (PC) is still an object that does not serve mankind without him having spent significant time mastering it. The PC is still a far way off from being as user-friendly as the modern automobile. Granted, an automobile is a complicated piece of mechanical and electronic machinery requiring a specialist for servicing, but the modern automobile is very simple to operate.

The novice programmer faces three challenges when programming: mastering the computer, mastering the programming process and mastering the programming syntax. What the novice programmer needs is a machine that is simple to operate. As is the case with a modern automobile, the complex underlying technology should remain hidden during normal operation. At IST-Africa 2008 it was reported that some tertiary-level learners experience difficulty in completing their computer programming assignments. University-level computer science students will experience less difficulty in completing their assignments if their exposure to the world of computers has been a gradual transition from concrete- to abstract thinking. To address this, our research aims to provide an alternative introduction to computer programming, specifically for young children.

Previous research [2] has highlighted the lack of sensitivity to local problems when aid is given to developing countries. The result is often compromised aid and at times the aid has no value. Our research approach is one of being sensitive to the local conditions in developing countries by allowing flexibility in the implementation of the tangible programming environment. The flexibility is possible because of the modular approach of the environment: the user herself decides on the appropriate representation of the programming objects.

In general, our research addresses two specific heuristics of Nielsen [3]. These are: (1) having a “match between system and the real world” by allowing a user to develop her own representations of the underlying concepts and (2) relying on “recognition rather than recall” because an obvious relationship exists between the object created by the user and the function it represents.

It is well known that children excel at playing. We therefore explore a tangible environment as an alternative introduction to programming for children. Children also excel at building their own toys when faced with a lack of funding. Also, not all children have access to the latest technology that is available in the developed world. For these two reasons we are also researching the integration of African craftsmanship with low-cost digital technology. It is important to note that the Physical Programming Environment that we discuss in this paper is not meant as a substitute for the conventional textual- and visual environments used by professional programmers. Rather, it is intended as an alternative introduction for novices to the field of computer programming.

2. Objectives

Our research is aimed at developing alternative interfaces to ICT-based equipment in general, and specifically for programming. Such interfaces have the potential to provide a larger group of people access to modern technology. Consider the small controls such as keyboard keys found on a cell phone and computer keyboard. These work well with people with no physical- or mental disabilities, but fail illetterate (not able to read or write) groups and those with reduced muscle control, sight or touch sensation.

Tangible interfaces provide a larger population group access to elementary programming because some of the obstacles found in current interfaces are removed. For example, tangible interfaces are large compared to modern interfaces and therefore usable by shaky and fumbling fingers.

Visual perception is an important ability when using tangible interfaces and we demonstrate how a tangible interface itself can be applied in quantifying this ability. This we do by reporting on a pilot study using hand-crafted interfaces for measuring certain aspects of visual perception amongst pre-school children.

3. Methodology

Our pilot design [4] has three parts (Figure 1). The first part consists of four tests to measure the child’s ability in making the mental mapping between a 3D object and its 2D representation. This we call “3D/2D Matching”. The second part measures the child’s ability to align 3D objects with 2D arrows. This we call “Orientation Matching”. The third part of the design measures the child’s ability to orientate and correctly place 3D objects with the objective of controlling a toy robot car. This is done by placing the 3D objects in a prescribed sequence. We call this “Tangible Programming”.

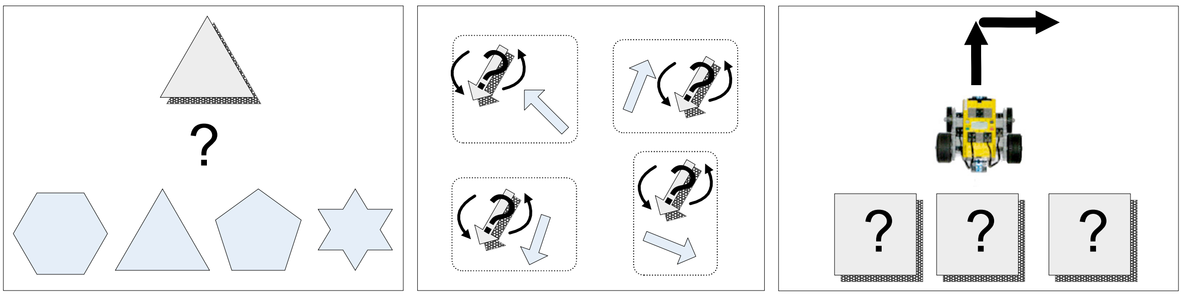

Figure 1: Diagrammatic representations of (left) a 3D/2D Matching test, (middle) Orientation Matching, and (right) Tangible Programming.

The tests were conducted with the assistance of children from two schools in South Africa. The schools serve two communities from differing socio-economic backgrounds. School A is based to the east of the city of Pretoria and School B is based south-east of the city of Johannesburg. The tests were conducted at the relevant school during normal school hours. Testee ages ranged between five and six years. All children from these two schools who were five or six years of age were offered the opportunity to participate in the tests.

The team of researchers consisted of two males and one female. The medium of teaching at both schools is English, and test instructions were given in English. When required, instructions were repeated in a language more familiar to the testee, such as Sepedi. The mix of testee gender at the two schools was not equal and we did not factor that into our test design.

Prior to each test, the group of testees were collectively instructed about the objective of the test and how it should be executed. The tests were executed with only one testee and three researchers present in the test room.

4. Related Work

Allowing users to construct their own intelligent interfaces is no longer novel and has been explored in various contexts. One system [12] incorporates position sensing elements in clay. The clay can then be moulded into various shapes. The sensors communicate their positions to an application run on a PC. A virtual 3D model is then automatically constructed in real-time and rendered on a monitor. Another design [13] attaches sensors on the edges of mechanically connected cardboard sheets. As the sheets are bent along the joint, the attached sensors transmit data representing the angles to an application. A model of the physical cardboard shape is shown on the computer display. A system that could be especially appealing to girls integrates electronic circuitry with textiles [14]. It provides a means for the user to computationally enhance clothes, adding actuators such as lights and speakers that respond to inputs from sensors, for example embedded switches.

5. Technology Description

In this section we consider examples of two popular programming interface styles and elaborate on a third style which is within the domain of our research. This interface third style was used in the execution of the tests. In each of the examples a short coding sequence is given. Two popular programming styles are the so-called (1) textual and (2) graphical interfaces. The third interface style is what we call "tangible programming". Figure 3 relates these interfaces to each other regarding the cognitive effort required by the programmer and the intuitiveness of the styles.

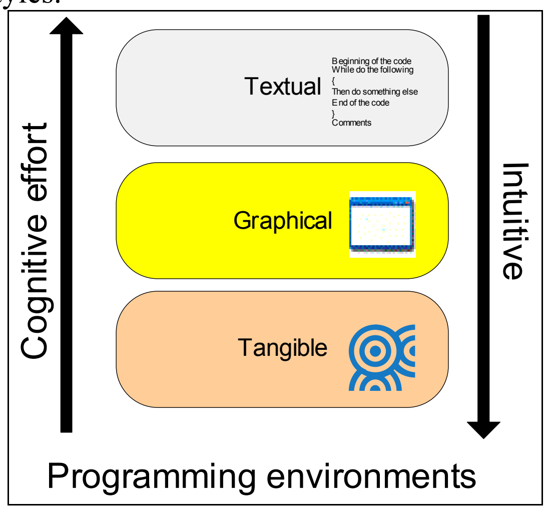

Figure 3: The cognitive effort to master a programming environment varies. A tangible environment requires less effort and is more intuitive than a textual programming environment.

5.1 - Textual Programming Environment

The first interface style is that of text-on-screen. This style is well-known to the professional programmer but to a lesser extent to the modern computer novice. The textual programming environment requires significant time to master due to its reliance on the programmer's ability to recall, from memory, the elements which constitute the language and the sequence in which these elements have to be placed. Figure 4 is an example of a text-based programming interface.

```
# Draw a square
paintWhite()
repeat(4)
{
   forward (2)
   right()
}
stopPainting()
```

Figure 4: A textual programme written in the RoboMind [5] environment to make an object on the screen move the outline of a square. The environment is completely dependant on textual input from the user.

Figure 5: A graphical program written in the Scratch [6] programming environment to make an object move the outline of a square. Although mostly a graphical interface, some parameters need to be adjusted using a textual interface.

5.2 - Graphical Programming Environment

The second screen-based interface is less demanding on the user and incorporates some graphical elements to make the environment visually attractive. It also provides clues to its use through the use of various geometric shapes, drop-down lists and the auto-completion of instructions. Figure 5 is an example of a graphical programming interface.

5.3 - Tangible Programming Environment

The third interface we consider is tangible and contains graphical elements. The nature of this interface is at the core of this paper. In a tangible environment, programmes are constructed by placing physical elements in a sequence that corresponds to the execution sequence. These physical elements are called Tangible User Interfaces (TUI's). Figure 6 is an example of a tangible programme construction comprising of seven GameBlocks [7] elements.

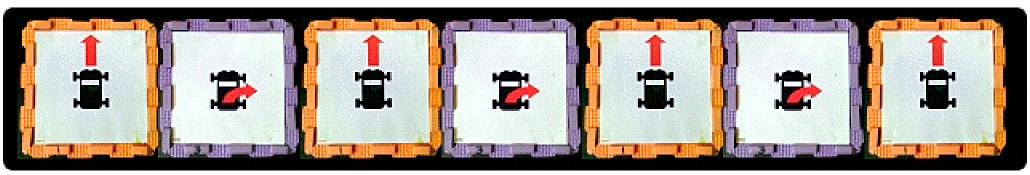

Figure 6: A tangible programme constructed using the GameBlocks programming environment. When executed, an object in the real-world follows the outline of a square. This environment does not have any parameters and is strictly graphical.

5.4 - Hand-crafted Tangible Programming Environment

An extension of the tangible programming environment is the use of hand-crafted artifacts. Using this approach, the end-user can shape the tangible objects herself. An example of this interface is RockBlocks [8]. RockBlocks consists of a number of arrows crafted from soft rock using simple hand tools (Figure 7) Once the arrow has been sculpted, it is mounted on a wooden block (Figure 8). The wooden block has magnets embedded in a pre-determined configuration. This assembly is then placed on top of a "reading' tray (Figure 9) which is in turn connected to an electronic circuit [7]. The circuit interrogates and interprets the configuration of arranged arrows. It then sends instructions to a motorised toy for execution.

Figure 7: hand tools and the unprocessed rock.

Figure 8: The completed tangible artefact shaped like an arrow and mounted on a painted wooden block with the aid of a metal bolt with a ring at one end.

Figure 9: A testee constructing a programming sequence using the hand-crafted Tangible User Interface.

6. Results

The collected data (Figure 10) indicate that children from School A had an initial advantage over those from School B. In the first test, which measured the children's ability to match 3D objects to 2D drawings, the children from School A had significantly better results than those from School B. The results from the second test, where the orientation of 3D objects are matched to 2D drawings, indicate that the difference in performance between the two schools is much less. Results from the third test indicate that the difference between the two groups is almost non-existent.

There was no evidence that the children from either school experienced difficulty in handling the TUI's. The large metal ring that is part of the TUI assembly presented itself as an intuitive handle for manipulation.

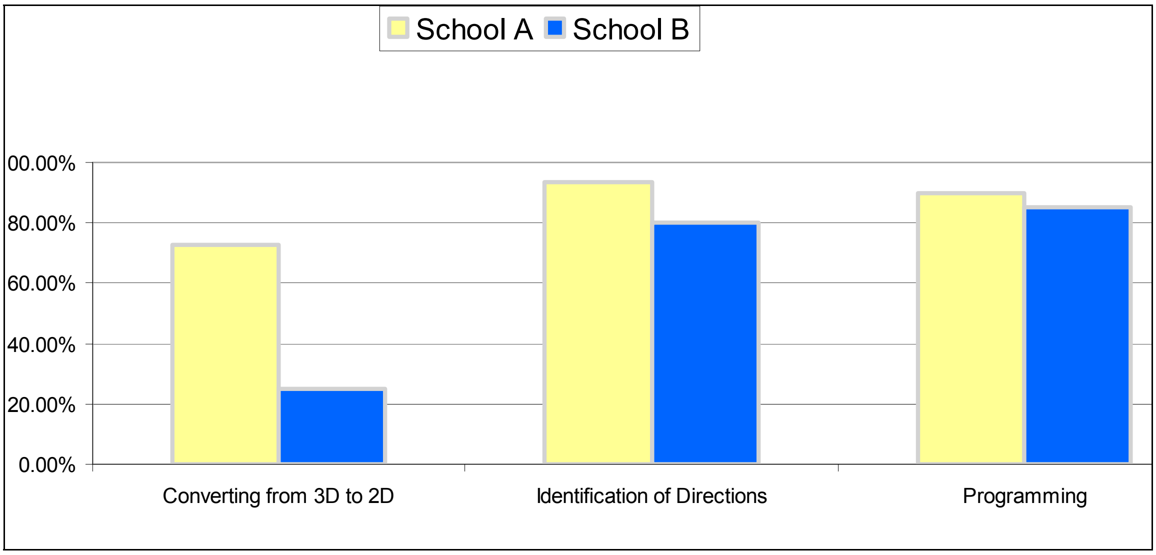

Figure 10: Test results of the two schools. School A seems to have had an initial advantage over School B. This advantage was significantly reduced by the time the third test was done.

7. Recommendations

The results given in this paper are from a pilot test. Further testing using a larger sample from additional diverse locations is required to validate the results. Although the rocks used in the tests have been crafted by hand, it was done by an adult. It would be interesting to explore the children's own ability to shape soft rocks, taking note of the tools they prefer and how their designs evolve as they become familiar with the properties of the rocks. Replacing soft rocks with paper mache, wood, or clay would provide an additional worthwhile research dimension.

8. Conclusions

Africa does not have the long tradition of digital technology integrated into daily activities as is the case in developed countries [9]. But that should be no reason why the application of logical thinking may not be used by the digital novice for controlling events and objects in the sub-Saharan context.

In this paper we have discussed a tool to measure the differences in, what we consider to be a requirement for successful tangible programming, certain visual perception skills between pre-school children from differing socio-economic backgrounds. We measured the ability of the children to map 3D objects to 2D drawings, rotate 3D objects to align with 2D drawings, and use 3D objects for programming.

The results from a pilot to quantify the differences indicate that the differences previously reported [10] are already evident at the ages of five and six years. The diminishing differences between the results of the two groups can possibly be explained to be as a result of the children from School B's increased familiarity with the tangible programming technology as the tests progressed.

We are of the opinion that having given the instructions in English did not affect the results as it was evident from our interactions with the children that they were proficient in that language.

The combination of wood, foam padding and rock used in the construction of the programming objects is durable as it has withstood the handling of many children. The very same objects have also been used at interactive science workshops held in the Eastern Cape Province, South Africa.

We anticipate that tangible programming environments will prove less intimidating to the novice programmer in developing countries than current textual and graphical systems.

Similar to Druin's [11] comments on introducing the novice to robotics, our intention is not to make expert programmers out of novices. Rather, our aim is to expose them to the opportunities offered through programming, be it textual, graphic, or tangible.

Acknowledgements

We are grateful to Dr Foko and Mrs van Deventer who assisted in the conceptualisation, design and execution of the experiments as well as the subsequent analysis of the collected data. Chris Krause provided invaluable inputs to this paper. The authors wish to express gratitude to the Morning Star school and the Zama care centre. This research was funded by the South African Department of Science and Technology.

9. References

[1] S. Papert, The children's machine: rethinking school in the age of the computer, Basic Books, 1993, p 157.
[2] M. Vesisenaho, Developing university-level introductory ICT education in Tanzania: a contextualized approach. Dissertation, University of Joensuu, Finland, ISBN 978-952-219-010-9, 2007.
[3] J. Nielsen, Ten Usability Heuristics, Available online from: www.useit.com/papers/heuristic , Accessed 12 November 2008.
[4] A.C. Smith, T. Foko, A. van Deventer, Quantifying the Visual Perception Skills of Pre-school Testees Using a Novel Tangible Electronic Test Instrument. Science real and relevant: 2nd CSIR Biennial Conference, CSIR International Convention Centre Pretoria, November 2008.
[5] RoboMind, Available online: http://www.robomind.net , Accessed 12 November 2008.
[6] Scratch, Available online: http://scratch.mit.edu/ , Accessed 12 November 2008.
[7] A.C. Smith, Tangible Cubes as Programming Objects; Artificial Reality and Telexistence: 16th international conference. California: IEEE Computer Society, 2006.
[8] A.C. Smith, Handcrafted Physical Syntax Elements for Illetterate Children: Initial Concepts, Proceedings of IDC '08, ACM Press, June 2008.
[9] H. H. Lund, Robots at Play, ISBN 978-87-992302-0-4, HTOdense, Denmark, 2007.
[10] T. Foko, The Role of Computer Games and Social Constructivism in Skills Development of Learners from the Disadvantaged Backgrounds. PhD Dissertation. University of KwaZulu Natal, 2006.
[11] A. Druin and J Hendler (Eds.), Robots for Kids: Exploring new technologies for learning, Morgan Kaufmann Publishers, ISBN1-55860-597-5, 2000, p 163.
[12] M. Reed, Prototyping Digital Clay as an Active Material, Proceedings of TEI '09, ACM Press, February 2009.
[13] Y. Haung, M. D. Gross, E. Y. Do and M. Eisenberg, Easigami: A reconfigurable folded-sheet TUI, Proceedings of TEI '09, ACM Press, February 2009.
[14] L. Buechley, N. Elumeze and M. Eisenberg, Electronic/Computational Textiles and Children's Crafts, IDC '06, ACM Press, June 2006.