Leisure Robotics: an African Child's Gateway to Programming

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

Abstract – **This paper describes an alternative programming system which can be constructed with minimal mechanical and electronic knowledge in an environment such as the FabLab. This system, dubbed GameBlocks, makes use of Tangible User Interfaces to control the movements of a motorised toy robot. No prior programming experience is required to use the system and no reading and writing abilities are necessary either. Such a system is possible because the user interface makes use of icons only and consists of three-dimensional objects that are manipulated in 'constructing' the required code sequence for controlling the robot.**

An overview of the system components, the design concepts, and experiences in creating the first prototype in the FabLab is provided. The paper concludes with a short description of our subsequent prototypes, highlighting the features we manipulated to incorporate the learning gained from the first prototype.

I. Introduction

'Hacking' computer code is no longer seen as being 'cool' by the youth. Robots and computer games are. A career in Information Technology is also attractive to the youth because of the perceived associated financial and status benefits. Unfortunately, many who enrol for Information Technology courses at a university experience difficulty in completing their practical computer programming assignments. It can be argued that the difficulty experienced by the students is due to a lack of access to computers at school plus their subsequent lack of the logical thinking experience required when enrolling for these technologically-advanced courses. The transition, from having no programming experience to the compulsory requirement for actual programming, is simply too great for many students. The challenge to educators is to reduce this transition by exposing learners to computer programming at school-level. Considering that few learners have access to computers, it would be wise to investigate alternative mechanisms for them to gain experience in the logical thinking process required for computer programming.

The use of icons in programming is well known in modern on-screen computer environments. However, these environments pose a significant cognitive load for the user. The cognitive load is a result of the abstract representation of the icon. The icon was typically designed by a person that can be described being of Western society orientation/mindset. It can be argued that a Western approach is not always the best, but little has been done to explore more indigenous approaches to computing systems, primarily because there is so little economical incentive for doing so. The challenge faced by African children is in mastering the use of computer technology in addition to mastering the English language in which programmes are commonly written. The aim of our research is to reduce the hurdles a novice programmer has to face by removing these challenges. In doing so, sacrifices are made but we also potentially gain future computer programming experts. What we sacrifice is a comprehensive programming language. The result is a very simplified programming environment, accessible to most children and adults. Another benefit is the creation of a low cost programming environment, one that can potentially be created with limited technologies. A key ingredient in having this objective is having access to a FabLab [1] or other facility with similar infrastructure. A typical FabLab is supplied with standard equipment such as a 35 Watt laser cutter, a sign cutter, a table top milling machine, and an electronics assembly and programming area.

As stated by Noble and Bestley [9 p52], different forms of communication are best suited for different target groups, and some forms of communication are more effective than others when targeting a specific group. This aligns well with our approach. Our approach does not rely on the target group's ability to read or write. Provision for this group has been made by completely eliminating written text in the user programming environment.

Why would we want to make a simple programming environment available to children? In response to this question we draw attention to a well-known construction system called LEGO. This patented construction system was developed in the 1950's and it is estimated that every year children spend about 5 billion hours playing with LEGO [2, p8]. It can be stated without a doubt that these children have an advantage in life over those that don't have access to such a system. Our aim is to make a computing environment available to children, an environment that they can shape and manipulate themselves. This objective is similar to that which the president and CEO of the LEGO Group, K.K. Kristiansen, has for the Group [2 p7].

Technology interaction should be in a context that the user is comfortable with [3 p145], rather than a context that has been designed from an industrial perspective that emphasises aspects such as mass production and ease of after sales support. This industrial perspective does not allow sufficient personalisation of

3rd Robotics & Mechatronics Symposium (ROBMECH 2009), 8-10 November 2009, hosted by CSIR in Pretoria, South Africa.
ISBN: 978-0-620-44721-8.

the dominant programming environment, the desktop computer, to make the novice feel comfortable with the technology.

When contemplating the introduction of new technologies to a developing region, one should not disregard the existing technologies in that region [4 p79]. Take as an example the technologies which the majority of children in Southern Africa and India are used to. These may be called 'regional' technologies and include the radio, cell phone and television. Note that the Personal Computer (PC) does not feature in this list. A small number of children also have access to modern facilities such as the FabLab. In addition to regional technologies, 'local' technologies are artefacts developed by the community itself. Musical instruments and pottery are examples of local technologies.

Combining local and regional technologies with modern facilities could lead to innovative solutions to local challenges [5 p.165].

Introducing computer programming at the same time as introducing a modern technology such as the PC can be overwhelming to the user. In addition, the PC is a piece of technology engineered for low manufacturing costs, with little to attract the novice to explore its potential. The PC has even been likened to a toaster and the software on the PC to toast, at times leaving the user unsatisfied [3 p53].

As a result, an alternative means for introducing programming concepts was investigated; a means that makes use of materials readily available to the user in that region; materials that can be combined and shaped into something personal. Examples of such materials are: discarded plastic bottles, wood, and soft rock such as sandstone. In Fig. 1, community members contribute their respective intellect to construct a personalised toy robot. In this scenario one community member is capable of constructing the controlling electronic circuitry, and the other member constructs the mechanical components. Together they have created something personalised by using local resources.

Soft rock was the first of these materials to be explored. Because the user herself shapes the artefact into which the computing technology is embedded, the soft rock allows the technology to 'become part' of what the user 'is' [6].

Children are more often than not enthusiastic creators, and allowing the creation of handcrafted artefacts as opposed to foreign mass-manufactured toys allows for self-expression and assists in the adoption of the technology.

This approach supports personalization and appropriation of the technology [7 p119], of which both are important technology adoption criteria for children [8 p242].

II. System Overview

The system (Fig. 2) developed consists mostly of low cost technologies. However, a low-cost robotic toy has not yet been developed. An appropriately low cost version is required to make this system truly suitable for developing regions. In our research we have used two distinct types of commercial robotic toys (Fig. 3). Both are available as commercial products. The first robot is humanoid in appearance. The second robot is assembled from a LEGO parts. Both are controlled using infrared signals.

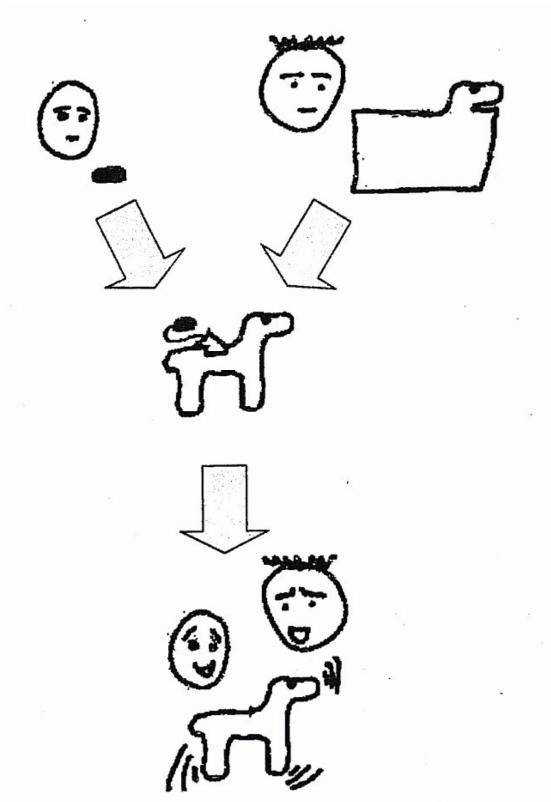

Fig. 1. Community members contribute their respective intellects to construct a personalised toy robot.

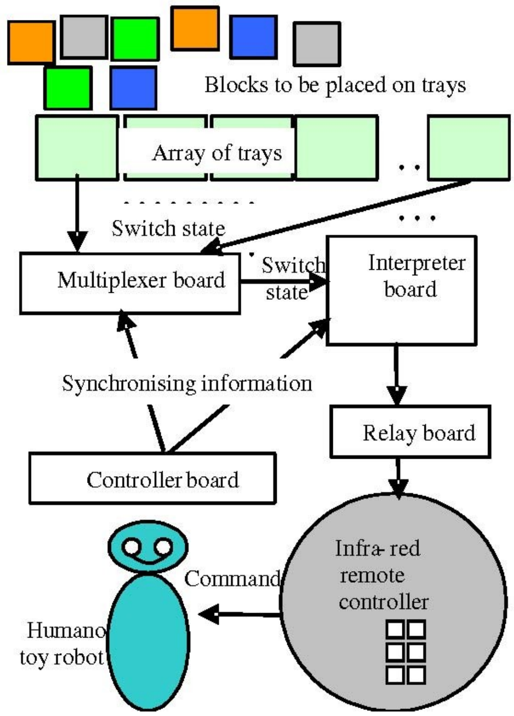

Fig. 2. GameBlocks system overview.

3rd Robotics & Mechatronics Symposium (ROBMECH 2009), 8-10 November 2009, hosted by CSIR in Pretoria, South Africa.
ISBN: 978-0-620-44721-8.

A programme is ‘constructed’ by placing physical programming objects on a number of programming ‘trays’. The programming objects are embedded with magnets that interact with the trays. Each tray represents a single line of programming code in a ‘traditional’ computer programme.

The user then activates the system by pressing a button. Each tray is sequentially ‘read’ by a custom-designed electronic circuit. As each tray is ‘interpreted’, an appropriate infrared signal is sent to the robot for execution.

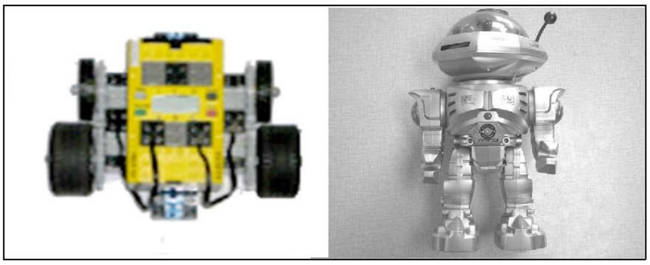

Fig. 3. Two toy robots used in experiments at science workshops with children.

III. Prototypes

A number of prototype programming systems have been developed, each an improvement on the previous, and designed explicitly to explore a different design approach.

The initial acrylic cube prototype was followed by a closed cell foam version. Next to be explored was the use of natural materials, at the same time changing the concept of a three-dimensional cube to one that functions in two dimensions only. The final prototype reported on here also functions in two-dimensions only. A significant change in this last prototype is the form in which the results are made visible. Instead of producing results in the physical world using toy robots, the results are made visible on a desktop computer screen.

The following is an overview of the four prototypes.

A. Acrylic cube prototype

The large acrylic cubes were laser cut at a local FabLab (Fig. 4). The design allowed for easy assembly, theoretically not requiring any glue due to the tongue-and-groove design. The same design principle was followed in the design and manufacture of the trays. Unfortunately this approach was not successful in practice and hot glue was used to keep the various pieces in place.

Our initial electronic circuit was designed and implemented to sense 24 trays. Because of the large number of wires required when attaching the trays to the electronic circuit, it was decided to reduce the number to eight in the next design iteration.

The magnets are placed in the corners and centre of the bottom cubes bottom surface. Every tray has five low-cost magnetic sensors attached at positions that correspond to the five possible magnet positions in the cubes. Provision for up to five magnets was made, allowing for a maximum of $2^5 = 32$ magnet combinations. The zero-magnet condition was reserved for indicating that no cube had been placed on a tray. This implies that a total of 31 possible instructions can be encoded by using five sensors.

Fig. 4. Transparent acrylic cubes and programming trays.

Transparent acrylic was utilised for all laser-cut parts so as to show the simplicity of the system to the various workshop attendees. The aim was to entice children to construct their own tangible programming environment.

Design errors soon surfaced. Here are a few. The trays were fragile: the small distances between the slots and the sharp angles would cause the spacers to break along the corners. Designing for friction-fit assembly, as in this design, is risky unless the 90 degree inner angle is increased. However this possible solution has yet to be tested.

Limited design effort was applied to the design of the symbols that indicate various robot movements. The resultant abstract symbols were cut using the vinyl cutter available at the local FabLab. These were then added to the cube surfaces. The effort required in explaining the function of the symbols during workshops was an indication that the design of symbols is not a trivial exercise. An attempt to address this problem was made in the next prototype design.

B. Closed cell foam prototype

The first prototype had some limitations as discussed in the previous paragraph and therefore an alternative implementation for the second prototype was subsequently tested. In this prototype commercial closed cell foam blocks were used.

With this design iteration the original 24 trays to eight. Eight ‘lines of programme’ code proved to be sufficiently challenging to the novice users who participated in our workshops.

More design effort went into the design of the symbols for this prototype. The result are more intuitive symbols that those used in the first acrylic cube prototype.

The softer, and more durable, closed cell foam cubes eliminated the risk of breakage completely (Fig. 5). This design retained the property of the first in that it can be assembled and disassembled easily without the use of either chemical or mechanical aids.

The only design problem that surfaced in this iteration was the need for careful alignment of the cube with the tray. Incorrect magnet combinations would be registered if the alignment was not done properly.

3rd Robotics & Mechatronics Symposium (ROBMECH 2009), 8-10 November 2009, hosted by CSIR in Pretoria, South Africa.
ISBN: 978-0-620-44721-8.

Fig. 5. Closed cell foam cubes and a programme consisting of seven steps.

C. Natural rock prototype

Having experimented with artificial raw products in the first and second prototypes, the use of natural materials was explored next.

This design takes into consideration the materials available to the user [9 p55], with the design being adjusted to accommodate this material.

Soft, natural rock is readily available in the geographical area of Southern Africa. The natural rock prototype is in essence a two-dimensional interface in comparison to the three dimensional prototypes of the acrylic cubes and the closed cell foam cubes.

The objective was to experiment with materials that can be crafted by the end-user and would also show wear and tear over time. It has been said that 'meaning' does not reside in the object itself, but rather that 'meaning' is brought to the object by its user [the French philosopher Roland Barthes as referenced by [9 p 99]]. In the same way, the rocks allow the use to change the shape of the artefact originally materialized by the manufacturer, be it a mass-producer or a village craftsman. Whatever the source, the user is empowered to modify the artefact in such a way as to provide personalized value. In addition, this interface approach allows modification by the user, even if produced elsewhere, perhaps in another developing country with differing abstraction ideas.

It can be argued that this approach places the user in an advantaged position relative to pears in developed regions. In these regions creativity has arguably been stifled through the ready access to entertainment technologies such as television programmes and video gaming consoles.

The concept of hand crafting the artefacts, using hand-tools only (Fig. 6), was subsequently evaluated informally. The results were promising and an additional eight artefacts were produced, this time with the aid of power tools. This was done for the simple reason of reducing the manufacturing time.

The rocks were shaped into arrows and mounted onto wooden bases. These in turn had magnets embedded into them.

As with the previous iterations, physical alignment also proved to be a problem with this design.

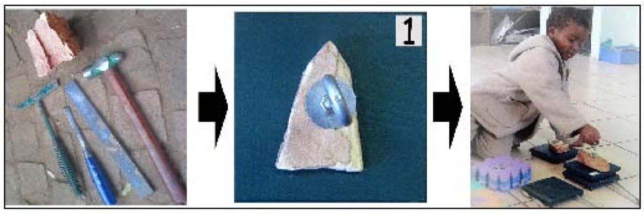

Fig. 6. Natural rock is shaped using hand tools and used as a tangible programming object

D. Rotary dial prototype

This prototype explores the progression of a novice user from having both concrete input and output modes, to only having a concrete input mechanism with a virtual output mode available. In this prototype the on-screen character provides the virtual output mode.

The rotary dial prototype is called the Dialando system. The programmer uses a series of rotation movements to instruct the on-screen character (Fig. 7). The instructions are similar to those in the natural rock prototype.

Dialando differs from the other prototypes in that a tight integration with the personal computer and PC-based programming has been created. Here the actions of an animated on-screen character is controlled. As for the other prototypes, the motions are simple. These consist of forward, back, left and right movements. The movements are limited to fixed distances in the case of forward and back movements, and fixed angles (90 degrees) in the case of rotations. The user is set a challenge of instructing the animated character to move over a fixed position on the screen.

Fig. 7. Dialando is a physical programming system using five rotational inputs to control an on-screen avatar.

3rd Robotics & Mechatronics Symposium (ROBMECH 2009), 8-10 November 2009, hosted by CSIR in Pretoria, South Africa.
ISBN: 978-0-620-44721-8.

IV. Prior Work

Pro-bot [12] is a commercial product aimed at young children. It is programmed by using the keys on the device and a liquid crystal display (LCD) provides visual feedback. An optional pen can be inserted for leaving a permanent output on paper when the programme executes (Fig. 8).

Bee-bot is a simplified version of Pro-bot, having a greatly reduced number of push buttons and it also does not have an LCD for reviewing the programme.

Fig. 8. Bee-bot and Pro-bot.

V. Future Work

Four input modalities for controlling a toy robot have been briefly explored in this paper. What is still lacking is an output device that can also be readily made in developing regions. The basic technologies required are not new and have been exploited for centuries to such an extent that these technologies have mostly been forgotten in developed regions where the focus has moved away from mechanical mechanisms to information technology such as the electronic computer. Examples are the mechanical movements which played such an important role in the industrial age [10]. Mechanical devices for the novice constructor have recently received renewed interest from the public through the collectable devices available as ready-made artefacts or in the form of construction plans [11]. These devices and plans bring robotics into the grasp of the novice.

This vision of the user constructing her own mechanical robot is not so far-fetched as one would imagine. Examples of constructing solutions to local challenges by using local resources are abundant [5 p165]. Using a FabLab to address local problems with simple technology is also possible. An example is the use of a low-cost 8 pin microprocessor and the facilities offered in a FabLab to determine the dilution and fat content of milk, a real problem when it was suspected that a local supplier was diluting milk.

We plan to extend our research to find a suitable mechanical output device design concept for the children of Africa. Such a device should ideally be manufactured by members of the local community using low cost materials and local facilities.

Acknowledgement

We would like to thank the many adults and children, together with their parents and teachers, who have participated in the numerous interactive workshops and provided valuable insights used to inform our designs. This work was sponsored by the South African Department of Science and Technology.

References

[1] FabLab, http://fab.cba.mit.edu/, accessed 1 September 2009.
[2] D. Pickering, N. Turpin, C. Jenner (eds), "The ultimate LEGO book", Dorling Kindersley, 1999.
[3] N. Gershenfeld, "When things start to think", An Owl Book, 2000.
[4] Hoadley, "Social impacts of mobile technologies for children: keystone or invasive species?", Mobile technology for children (ed. A. Druin), Morgan Kaufmann Publishers, 2009.
[5] N. Gershenfeld, "FAB: the coming revolution on your desktop – from personal computers to personal fabrication", Basic Books, 2005.
[6] C. B. Merkel et al., "Sustaining computer use and learning in community computing contexts: making technology part of 'who they are and what they do'. Journal of community informatics, 1(2), 158-174, 2005.
[7] Jones, "Mobile interaction design matters.", Mobile technology for children (ed. A. Druin), Morgan Kaufmann Publishers, 2009.
[8] Hourcade, "Early OLPC experiences in a rural Uruguayan school", Mobile technology for children (ed. A. Druin), Morgan Kaufmann Publishers, 2009.
[9] I. Noble and R. Bestley, "Visual research: an introduction to research methodologies in graphic design", AVA Publishing SA, 2005.
[10] Brown, H. T., 507 "Mechanical movements: mechanisms and devices", Dover Publications, 2005. Originally published by Brown and Seward, 1868.
[11] A.L. Onn, G. Alexander, "Cabaret mechanical movement: Understanding movement and making automata", Cabaret Mechanical Theatre, 2006.
[12] Pro-bot, http://www.terrapinlogo.com/bee-bot.php, accessed 1 September 2009.

3rd Robotics & Mechatronics Symposium (ROBMECH 2009), 8-10 November 2009, hosted by CSIR in Pretoria, South Africa.
ISBN: 978-0-620-44721-8.