Symbols for Children's Tangible Programming Cubes: an Explorative Study

Andrew Cyrus Smith
CSIR Meraka Institute
PO Box 395, Pretoria
0001, South Africa
+27 12 841 4626

acsmith at csir.co.za

ABSTRACT
In this paper we report on an explorative study done with children with the aim of developing symbols for a tangible programming environment that does not incorporate written text. We describe our methodology, provide results obtained and discuss them. We also provide a short overview of prior physical programming research.

Categories and Subject Descriptors
H.5.2 [**Information Interfaces And Presentation**]: User Interfaces – *haptic I/O.*

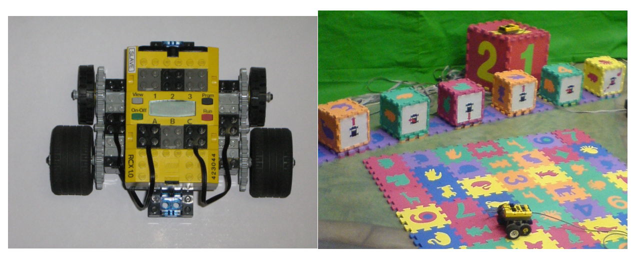
General Terms
Design, Experimentation, Human Factors.

Keywords
Orientation, symbology, evaluation, tangible user interface, text-less programming.

1. INTRODUCTION
Current desktop computer interfaces have been described as being of a *very narrow bandwidth* and *highly constrained* [7]. It can be argued that the communication bandwidth in interacting with the computer approaches zero for people who have had no prior exposure to desktop computers. This is because of the many obstacles a novice computer user faces. Some of these obstacles include: mastery of a written language, the ability to aim the mouse cursor accurately while pressing the mouse button, finding the letters on the small keyboard, and understanding the abstract structure of the operating system's filing sub-system.

Our aim is to "increase" this "bandwidth" for children. We anticipate that this can be done by improving the naturalness of interfaces used to interact with computing devices. This could arguably be achieved by using objects and materials that children are familiar with. Examples include the use of foam cubes and utilising symbols instead of text.

One research project which addresses some of these obstacles is GameBlocks [10]. GameBlocks is a physical programming environment aimed at young children. It makes use of physical syntax elements in the form of cubes to construct a programming sequence. A toy car is controlled (Figure 1a) when the sequence is executed. This is done by placing cubes in a linear sequence on a programming mat and switching on the associated electronic sub-system. Each block is interrogated in sequence and the infrared command, associated with that block, is sent to the toy car for execution (Figure 1b).



(a) **(b)**

Figure 1. (a) The motorised toy car which is controlled by the GameBlocks.

(b) GameBlocks system in use with cubes on the programming mat in the background and the toy car in the foreground.

GameBlocks programme sequences are a simplified Logo-like language [12], using symbols (Figure 2, 3) on the side of cubes instead of text on a computer screen. The results are visible through a physical 3-dimensional object's actions (the toy car), instead of 2-dimensional drawings on a computer screen.

The target group for this programming environment is children with no or limited exposure to computer programming. The added benefit of the system is that no reading or writing skills are required to construct the programme. This is accomplished by manipulating physical objects, each representing a programme instruction. The objects are three-dimensional and can be differentiated from each other by the symbols. The cubes are placed in the sequence (Figure 4) required to execute the task. No electricity is required up to this point. When the child is happy with her programme, she activates the system and the programme is executed by the embedded electronic circuitry. Results are visible through the actuation of the motors in the toy robot car.

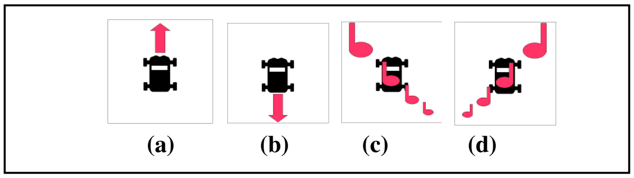


Figure 2. Symbols for a) move forward, b) move back, c) tone with decreasing pitch, d) tone with increasing pitch.

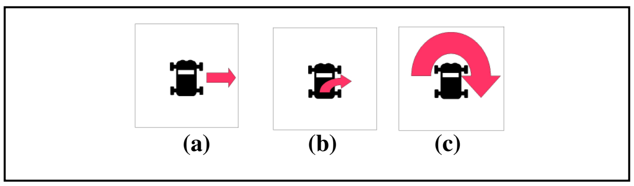


Figure 3. Three variants of "turn right".

The Lego programming environment describes three programming paradigms: iconic, direct manipulation and textual [3]. Our research relates to two of these paradigms. They are: a) the iconic and b) direct manipulation programming paradigms.

We aim to involve children in the symbol designing process to make the symbols on the cubes appropriate and understandable to children. In moving towards this aim, we held a series of workshops with children with the purpose of developing appropriate and understandable symbols for controlling a toy car. This paper reports on the children's designs that emerged from the workshops.

2. DESIGN

Our GameBlocks physical syntax blocks are cubic, allowing for potentially 6 instructions per side. Cubes also have the advantage that they can indicate direction as triangular or circular shapes [8].

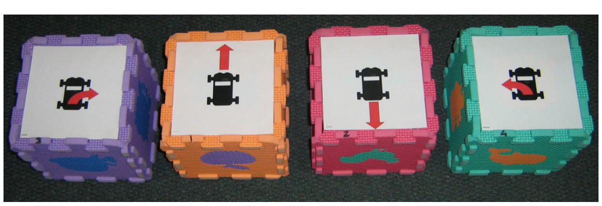


Figure 4. An example of a programme sequence – "turn right", "go forward", go backward", and "turn left". The sequence is interpreted from left to right.

The cubes are constructed using a commercial EVA (Ethylene Vinyl Acetate) - based toy product. Because EVA is closed-cell foam with good vibration absorption qualities, it is well-suited for use by pre-school children.

The symbols on the cubes were designed by the authors, obviously not being of the target age group and with very limited design training. All designs were from a top-view perspective. We never considered that a side view would be valuable. As we report elsewhere in this paper, we were proven wrong by some of the children.

3. METHODOLOGY

3.1 Consent

We required written consent from the parents/guardians for all the children who participated in our study. This was to inform the parents and participants of their rights and inform them of what will be done with the collected data [9]. The sessions were video-recorded and dozens of still-photos taken. The purpose of these was to assist in later evaluation of the sessions.

3.2 Introducing GameBlocks to the Children

We introduced GameBlocks to the children by first explaining the various components (the motorised toy car, the cubes, and the mat). We then explained the various functionalities that each cube represented (forward, left, right, reverse, sounds). This was followed by a demonstration of how the system operates, including a demonstration of how far the toy car moves with each instruction. We also ensured that the children understood how to position the cubes on the programming mat and what the correct orientation of each cubes was.

3.3 Sessions

We presented the workshops on a number of occasions at various venues. Two occasions were regional science events aimed at the youth. Another event was as part of our research with children where five schools are involved, six children per school. Children from four of these schools visit our lab twice a month in two separate groups of 12 children each.

Sessions consisted of groups of between five and 10 children, of mixed gender, ethnic background and ages. However, the ages were mostly in the range seven to 12 years.

3.4 Questionnaire

The questions (Table 1) posed to the participants were aimed at determining their level of prior computer exposure, categorised according to their age and gender [4]. In this paper we do not report on these outcomes. The answers to the questions posed in Section C are discussed in this paper. At the start of each session, the participants completed section A of the questionnaire. Sections B and C were completed at the end of the session with approximately 15 minutes allocated for the answering activity. Participants completed the questions individually.

3.5 Challenge

To make the session interesting we posed a challenge to the children. They had to use the GameBlocks cubes to instruct the motorised toy to reach an objective. The objective was to push a small plastic container which had previously been placed on the mat (Figure 1b, container not shown). It was up to the children to decide which cubes and which sequence of cubes to use to accomplish the objective.

Each participant was given paper sheets with copies of the cube symbols. The number of sheets corresponded with the number of GameBlocks cubes. The children were allowed time to design their program using these sheets.

During the evaluation, teach participant was asked to point at each block as it was "executed". This was to a) confirm that the underlying electronic circuitry was executing correctly (we have previously had problems in the implementation reliability) and b) to let the child follow the logical flow of the programme.

Table 1. Sections A, B, and C of the questionnaire.

Section A	Section B
1. Have you ever used a computer? 2. Have you ever written a computer programme? 3. Is there a computer in your home?	1. Did you like GameBlocks or did you find it boring/difficult to use? 2. Do you think you can use it on your own, or would you like someone to help you next time? 3. Would you like to play with GameBlocks again? 4. What would you like to change about GameBlocks?
Section C We also asked the testees to draw pictures of the following car actions, without using text: 1. Move forward, keep going, 2. Move forward and stop, 3. Turn right and stop, 4. Turn right and keep going.	

After each participant has had a turn using the system, we asked the children to draw pictures (without adding text) to illustrate the actions of the toy car. The actions they had to describe were for the toy car to: a) move forward without stopping, b) move forward and stop, c) reverse without stopping, d) reverse and stop, e) turn left, and f) turn right.

The reason for not using text is twofold. First, our research is aimed at people unable to read or write, such as pre-school children and the elderly in developing countries. Second, we want to develop a symbolic "language" that could be used internationally, by people from varying backgrounds and cultures, and having knowledge of diverse languages.

4. RESULTS AND DISCUSSION

The data presented here was collected at a series of workshops held with children. Some sessions were held at our lab in Pretoria. The other sessions were held at a regional science event in Grahamstown, specifically targeted at children. Most of the children have had no prior computer exposure.

In our analysis all drawings were considered, even those of children who where not able to complete the programming task on their own.

Some children preferred a side-view of the vehicle (Figure 5). However, most children depicted the car by means of a top-view (Figures 6 and 7).

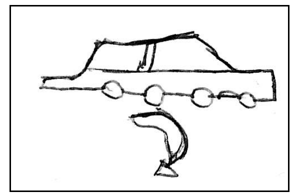


Figure 5. Very few of the drawings are a side view of the object.

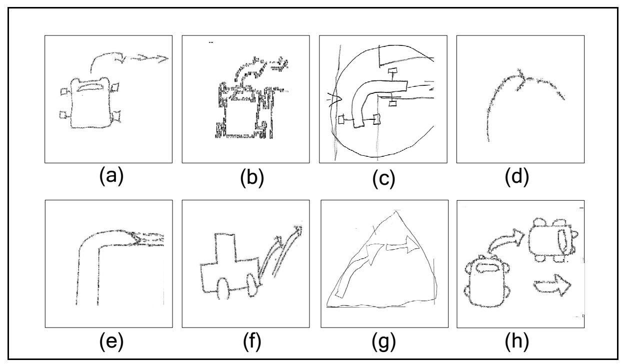


Figure 6. Drawings depicting 90 degrees right turn *without stopping*.

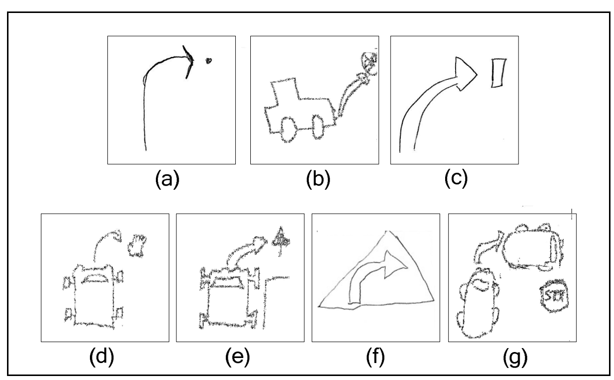


Figure 7. Drawings depicting various 90 degrees right turn actions *followed by a stopping action*.

The top-view abstract image we used to represent the toy car seemed successful. Approximately 95% of all participants successfully associated our images (Figures 2 and 3) of the toy car with the toy car itself. This is because participants correctly associated the white rectangle in the picture as the windscreen of a car, thereby identifying the front of the car.

The results given here are for the "turn right" situation, arguably the most complicated situation to draw without using any text to supplement the drawing.

Some children used text to describe the actions (Figure 7g), even though it was made clear to them at the start of the evaluations that only pictures should be used. Perhaps those children felt that their own designs were not sufficiently clear for others to understand and added the text to compensate.

From the results it is clear that there are common elements in the symbols drawn by the participants. We found that the common elements can be classified into two categories. The first category includes the use of a dot or rectangle to indicate a stopping action (Figure 7 a,c), and multiple arrows, dotted lines or multiple parallel lines to indicate continuation without stopping (Figure 6 a,b,d,e,h).

The second category includes the use of road signs (found either on the side of the road or on the road surface) or hand signals to indicate that stopping is required (Figure 7 b,d,e,f,g).

These two categories constituted the majority of the symbols.

Very few participants depicted a car as if it is made of a plastic-like material, allowing it to deform as it goes through a 90 degrees turn (Figure 6c).

5. PRIOR WORK IN THE FIELD

The following are real-world physical objects that children can manipulate to achieve pre-determined outcomes.

Story telling (Figure 8a) cubes uses pictures on cube faces for directing a story line [13].

Block jam [8] uses simple, electronically controlled, dynamic, abstract icons that do not directly associate with any known functionality.

ActiveCubes does not make use of any symbols or text [5]. All cubes in this system are identical. Only the relative positions of the cubes have any significance.

Navigation blocks (Figure 8b), with the text "Who What Where When", are manipulated to construct database queries [2].

Media Cubes (Figure 8c) combines logos and text on cube surfaces to implement a home automation system [1].

AlgoBlock (Figure 8d) utilizes a combination of symbols and text [11].

Electronic Blocks (Figure 8e) makes use of symbols on the sides of plastic blocks [14].

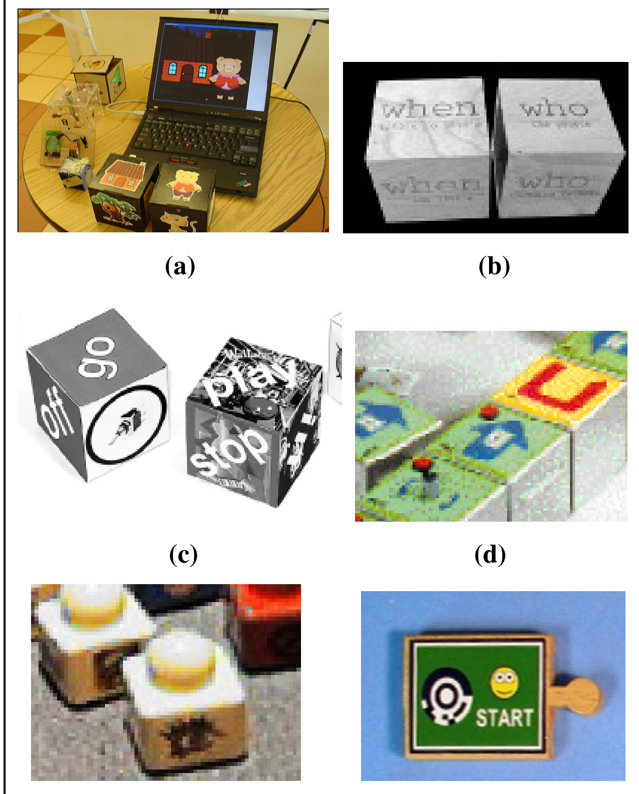
Tern (Figure 8f) makes use of text on 2-dimentional wooden blocks and image processing to recognize the relative positions of the blocks to each other [6].

6. CONCLUSION

In this explorative study we have found that many children based their graphical designs on the existing design as already incorporated on the GameBlocks cubes. Perhaps our expectations were too high, or perhaps the children's creativity had already been limited by the educational system and thereby reducing their creative thinking. An alternative explanation could be that the children did not understand the instruction to create their own symbols, and tried to imitate those on the GameBlocks cubes.

Some design results were similar across all the workshop sessions. However, the groups participating in the workshops were independent of each other with no known prior contact, eliminating the possibility of collaboration as being the reason for the similar results across groups.

We have not yet begun investigating the various cultures when designing symbols for use in Tangible User Interfaces (TUI's). Much literature is available on this topic; although the authors doubt that anything significant is available that specifically supports programming concepts. It will be a worthwhile contribution to the field of TUI's to develop such symbols.



(a) (b)

(c) (d)

(e) (f)

Figure 8. (a) Story telling cubes (c) Media Cubes (d) AlgoBlocks (e) Electronic Blocks (f) Navigation blocks (f) Tern blocks.

We plan to investigate the use of materials commonly found around the home, especially homes in developing countries, in the making of tangible user interfaces. These include artefacts made from recyclable materials such as soft-drink containers, wood, and natural stone.

It would be interesting to conduct a second series of workshops with children, only this time using the symbols reported on in this paper instead of the symbols designed by the authors.

7. ACKNOWLEDGMENTS

We wish to thank Mpho Nkuna for the implementation of the electronic circuits used in this research. This research was funded by the South African Department of Science and Technology.

8. REFERENCES

[1] Blackwell, A.F. and Hague, R. 2001. Designing a programming language for home automation. Proc. 13th Annual Workshop of the Psychology of Programming Interest Group, 85-103.

[2] Camarata. K, Yi-Luen Do, E., Johnson, B. R., and Gross, M. D. 2002. Navigational blocks: navigating information space with tangible media, Proc. 7th international conference on intelligent user interfaces.

[3] Cockburn, A. and Bryant, A. 1997. Leogo: An Equal Opportunity User Interface for Programming. Journal of Visual Languages and Computing 8(5-6):601-619.

[4] GameBlocks Evaluation Form. http://playpen.icomtek.csir.co.za/~acdc/education/ScienceUnlimited2007/questionnaire%20v1'3.odt . Accessed 23 January 2009.

[5] Kitamura, Y., Itoh, Y. and Kishino, F. 2001. Real-time 3D interaction with ActiveCube, CHI '01 extended abstracts on Human factors in computing systems.

[6] Horn, M., and Jacob, R. J. K. 2007. Tangible programming in the classroom with Tern, Proc CHI '07 extended abstracts on Human factors in computing systems, 1965-1970 ACM Press.

[7] Jacob, R. J. K. 2003. Computers in human-computer interaction, In The human-computer interaction handbook: fundamentals, evolving technologies and emerging applications. 147-149. Jacko J. A., Sears, A. (eds). Lawrence Erlbaum Associates, Publishers.

[8] Newton-Dunn, H., Nakano, H. and Gibson, J. 2003. Block jam: a tangible interface for interactive music, Proc. New interfaces for musical expression, 170-177.

[9] Participant instructions and informed consent. http://playpen.icomtek.csir.co.za/~acdc/education/ScienceUnlimited2007/consent%20form%20v1'1.odt . Accessed 23 January 2009.

[10] Smith, A. C. 2007. Using magnets in physical blocks that behave as programming objects, Proc. Tangible and embedded interaction, 147-150, ACM Press.

[11] Suzuki, H., and Kato, H. 1995. Interaction-level support for collaborative learning: AlgoBlock—an open programming language, Proc. Computer support for collaborative learning, 349-355, Lawrence Erlbaum Associates, Inc.

[12] Papert, S. 1980. Mindstorms. Sussex, The Harvester Press.

[13] Tang, T. S.: Storytelling Cube: A tangible interface for playing a story .

[14] Wyeth, P. and Wyeth, G. 2001. Electronic Blocks: Tangible Programming Elements for Preschoolers. Proc. Conference on Human-Computer Interaction.