Using Magnets in Physical Blocks That Behave As Programming Objects

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

ABSTRACT
In this paper we describe the implementation of GameBlocks, a novel digital manipulative system for coding simple programme sequences to control a toy robot. A contact-less, magnetic field-based mechanism for transferring information about the blocks is described. The mechanical and electronic system components are described. We position this implementation in relation to prior related work. Problems encountered are given, with suggestions for future work.

Author Keywords
TUI, Digital manipulatives, Montessori-inspired manipulatives, illiterate programmer.

ACM Classification Keywords
H5.2. Information interfaces and presentation (e.g., HCI): User Interfaces.

INTRODUCTION
GameBlocks can be classified as a digital MiM ("Montessori-inspired Manipulative")[13], consisting of blocks that are physically manipulated by placing them on trays to form a sequence in a specific order. See Figure 1. The relative positioning of the blocks is significant and represents an underlying logical structure. Six robot movements are controlled using the blocks [8]. As for [10], the blocks support direct manipulation of objects and learning-by-doing. No PC (personal computer) is required to "write" a short robot-controlling sequence. To do this, manual dexterity is practiced as opposed to using a computer keyboard. In recent years, children's social interaction has changed from intimate to solitary, similar to what is reported for board games versus computerised games in front of computer monitors [2]. It has been reported that children often struggle more with programming syntax and not so much with the concepts. Children in the age group 4-7 years sometimes have difficulty in using the keyboard and interpreting error messages [1].

Our system aims to provide a mechanism for constructing a programme without the coder being literate. It can be used by children and adults alike to learn simple sequential computing principles. None of the mechanisms described in the section on related work given below is used in our system. The blocks only contain inexpensive magnets. When placed on a tray, magnets close magnetic switches.

A number of design guidelines have previously been proposed [13], and we have embodied a number of them; generic structures versus real-world objects (the toy humanoid robot can just as easily be replaced by a toy tank), level of abstraction, semantic association (done with the use of differing coloured blocks and icons on the blocks), coincide i/o (no other interface needs to be manipulated), and synchronous i/o (the effect of manipulation is immediately evident).

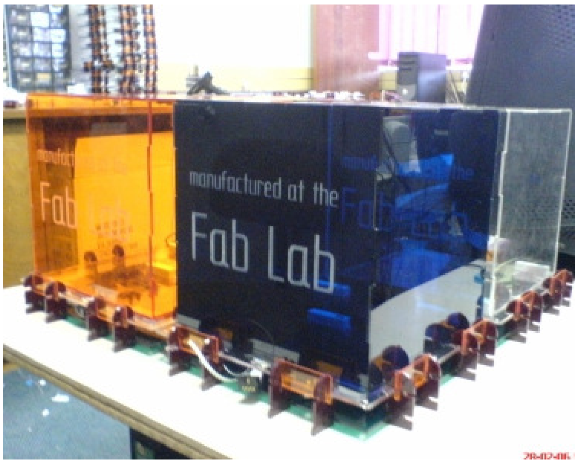


Figure 1. Blocks placed on trays, ready for interpretation by the associated electronic circuitry.

RELATED WORK
Several researchers have explored intelligent objects to enhance children's learning experiences.

"Electronic Blocks" [12] embeds electronic circuitry inside commercially available plastic building blocks. Electrical connections allow for messages to be passed between the blocks. This implementation has input (sensor), output and logic blocks. Others [4][7][9][11] also use embedded electronics. These methods result in a higher costing solution because of replication of electronic circuits. Another implementation uses a low-cost system using optical images to transfer the tangible code to a text file [6]. The text file is then used as an input to the compiler. Another communication mechanism uses optical fibres to send a message from one block to the next, until it reaches the controlling block. The complete message string is then sent to a synthesiser to pronounce the word. This system allows the construction of words by joining physical blocks in sequence [5, p23].

Our blocks are differentiated in that they do not have costly and sophisticated technology embedded in them. Instead we use magnets in the blocks and a centralised electronic circuit. This reduces the cost of the system significantly.

SYSTEM DESCRIPTION

The system described here consists of Instruction Blocks, Programming Trays, electronic circuitry, and a remote controlled toy (Figure 2). The user need not be literate in order to construct a programme that can control the humanoid toy robot. By simply inspecting the symbols on each block, the user chooses the appropriate one and places it on a tray. This is equivalent to writing one line of code in the traditional computer programming environment.

The system can operate in either immediate or batch mode. In immediate mode the object under control responds immediately when an instruction block is placed on the programming tray. This mode has been described as "coincident and synchronous" [13]. In the batch mode, the user (a child) gives an indication to the system when interpretation can commence. The user does not interact further until the execution comes to a halt. This paper only describes the immediate mode of operation.

Functions

Our current implementation has six instruction blocks; forward, backward, body left, body right, head left, and head right [8]. Figure 3 shows a sequence example of blocks required to programme the robot to trace the outline of a square when it moves. A "pause" function of fixed duration is inferred when a tray has no block placed on it. This is also known as the no-operation (NOP) function. The current implementation of our design concept allows for linear placement of 20 blocks onto 24 trays. The linear configuration can be changed. Examples are a single row, a single column, a combination of rows and columns (such as rows in a text book). See Figure 4 for an example using two rows. The optimal configuration has yet to be determined through experimentation. We encode the instruction block functions by superimposing a virtual 3x3 grid on the trays, and bottom of each instruction block. In each square thus defined by the grid, we can position either a magnet (instruction block) or reed switch (tray).

The current implementation utilises 5 virtual square positions. This allows for 2^5-1, that is, 31 functions. The

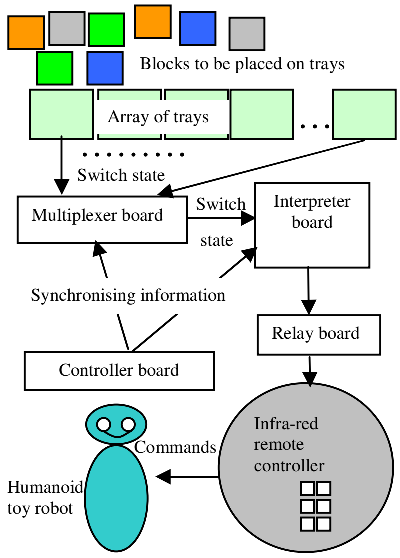


Figure 2. System components.

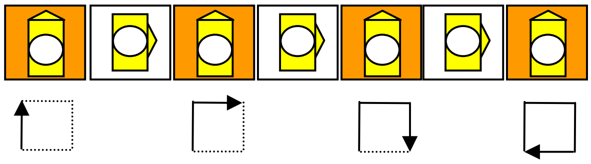


Figure 3. Physical sequencing example of blocks and the resultant movement of the robot.

Figure 4. Blocks placed on two rows of trays during a workshop session.

NOP function could be regarded as the 32nd one. It would be feasible to use all 9 squares to encode 2^9-1, that is, 511 functions. Being a concept demonstrator, we opted to limit this implementation to only 6 functions. A simple-state machine in the controller board loops through the sampling process, sequentially sending the state information of each tray to another state-machine on the interpreter board. This is done at a rate sufficiently slow for the toy to have finished execution of the current instruction before the next one is received. There is no feedback path from the toy that can influence execution of either state-machine.

Mechanical design

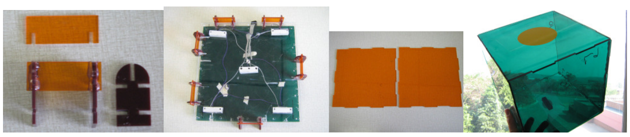
A fundamental design criterion for early childhood-development and learning is allowing children to experience active manipulation of real materials [12, p94]. For this reason we have opted for a design using large blocks that can best be used on the classroom floor. The blocks measure 247x247x247mm each. The tray dimensions are 275x275x35mm.

We arbitrarily used grey, green, orange, blue, white and clear blocks. Although the toy robot has many available functions, we have chosen to only implement six of them in this system. The mapping of the colours to the functions were arbitrarily made and is given in Figure 7.

Programming Trays

The trays each consists of two acrylic sheets. The bottom sheet is a translucent green, and the top sheet is clear. The clear sheet is meant to allow children participating in the workshops to see the underlying construction of the trays, including the positioning of the low-cost reed switches typically used in home intruder alarm systems. The magnets used in the blocks have an appearance and dimensions similar to the reed switches.

The two square acrylic sheets are separated at a spacing of 19mm and parallel to each other by six spacers per tray. See Figure 5 (a). The spacers consist of two pillars and a cross-beam. These components are also laser-cut out of acrylic material. The total height and width of the spacers are 54mm and 64mm respectively.



(a) (b) (c) (d)

Figure 5. (a) Spacers for the trays. (b) Assembled tray. (c) Two of the six block sides, showing where they interlock. (d) Fully assembled block.

Spacers are positioned on the four sides, two per side (for two of the sides) and one per side for the other two. This configuration allows for interlocking the trays when setting up at science shows. It also allows for experimenting with various lay-outs. Figure 5 (b) shows one of the assembled trays.

Five reed switches are glued to the bottom sheet, in the space between the two sheets. All trays have five reed switches mounted in positions C1, A1, B2, C3, and A3 respectively. See Figure 5 (b) and 6 (left).

Instruction Blocks

As a research project that presented workshops at various science shows, the design of the blocks had to allow for easy assembly and collapse. A simple tongue-and-groove design that allows for this is shown in Figure 5 (c). Sides of each block are simply pressed into position and held there by friction. Figure 5 (d) shows a fully assembled block. The positioning of the magnets in the blocks and the reed switches in the trays were determined experimentally. We had to determine what spacing between adjacent switch and magnet pairs is required so that they do not interfere with each other. The positioning as reported on in this paper provides more that adequate spacing and could be reduced significantly. With closer spaced magnets/reed switches, a more compact block/tray is possible. Alternatively, more functions can be encoded in the existing space by using a larger number of magnets per block and a correspondingly number of reed switches in the trays.

Electronic circuitry

Two low-cost microprocessors (PIC16F628) implement two simple state-machines, one for the controller board and another for the interpreter board. The controller board keeps the multiplexer and interpreter boards synchronized. A synchronizing signal is sent to both boards at a constant rate. With each occurrence, the multiplexer circuit samples the next tray in sequence, and the controller board latches the output from the multiplexer circuit. This value represents the function of the block that has been placed on the sampled tray. After the value has been interpreted, a corresponding contact on the infra-red remote controller is closed using the relay board. This sends a command to the toy robot for immediate execution.

Encoding of the Functions

To position the reed switches in the tray and the magnets in the block, we superimposed a virtual grid over each. Each possible position is referred to by its co-ordinate, for example co-ordinate B2 refers to the central position. Co-ordinate C1 refers to the bottom-left position. See Figure 6. In the current implementation we only make use of five positions, these are at co-ordinates C1, A1, B2, C3 and A3. Each tray has a reed switch in a corresponding position.

Figure 6. (left) Location (marked yellow) of reed switches on the trays. (right) Allowed magnet positions (marked red) in the block.

A block has between one and five magnets glued to its base. The position and number of magnets encode that particulars block's function. In Figure 7, a red square represents a magnet in that position. Other possible magnet positions, but not used in this research, are represented by clear squares. The icon which is visible on the top of each block is also given in Figure 7 and is a representation of the instruction.

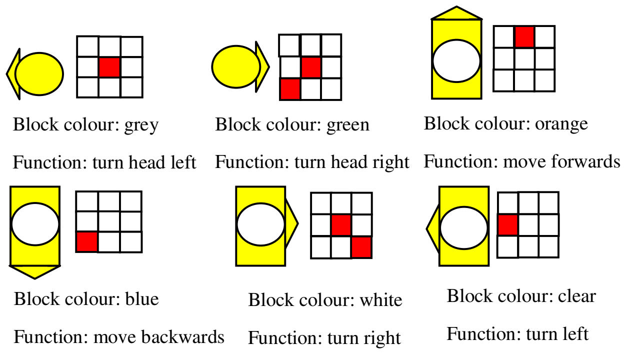


Figure 7. Icons on the block, magnet positions in the grid, colour of the block, and the functions they represent.

FEEDBACK FROM USERS

On average, groups of 20 children were exposed to the GameBlocks during sessions lasting 45 minutes. The children reported mixed feelings after having been exposed to the GameBlocks [3, p9]. They provided verbal comments that will influence future developments: they could not always hear what the robot said, the blocks are too large, they would prefer the robot to move in the direction indicated on the block (that is, in world co-ordinates and not according to its own co-ordinates), and the movements can be made more visible (exaggerated). Of particular interest is the comment on the size of the blocks. Our research interest is aimed at the use of large tangibles,which is directly opposed to the suggestion of reducing the size of the blocks.

FUTURE WORK

The use of a physical key to open the blocks and insert instruction items needs to be researched. The block will then inherit the inserted item's functionality. This idea is based on the special key reported on in [1] for simplifying the editing process in an icon-based programming environment. The materials from which educational artefacts such as the GameBlocks are made need to be re-considered [5, p26].The current design reported on in this paper is not ideal; the plastic used is hard and brittle, the edges and corners (where three surfaces intersect) are sharp, the surfaces are too smooth for small hands to grasp. Smith reports [8] on the problems experienced during workshops, which includes magnets becoming dislodged. This can be overcome by using appropriate glue for fixing the magnet to the plastic.

CONCLUSION

We have described the design and implementation of a novel and low cost mechanism to remotely control a toy robot using physical blocks. A number of shortcomings of the current implantation have been given and possible solutions to some of them suggested.

ACKNOWLEDGMENTS

We wish to thank Mpho Nkuna, Reinett Mafentile and the people at the FabLab for implementing the design. Corne Kruger designed and executed the usability tests. Feedback received from the conference reviewers is greatly appreciated. This research was funded by the South African Department of Science and Technology.

REFERENCES

1. Alexandrov, K., Soprunov, S., Yakovleva, E., Logo for the illiterate programmers, *Eurologo97*, 20-23.
2. Anderen, T.L., Kristensen, S., Nielsen, B.W., Grønbæk, K., Designing an augmented reality board game with children: the battleboard 3D experience, In *Proc. IDC 2004*, ACM Press.
3. Bekker, A., Kruger C., Usability and educational review on GameBlocks and Body Ping Pong. Test report, Test and Data Services, Available online http://playpen.icomtek.csir.co.za/~acdc/education/ScienceUnlimited2006/reports/ICT in Education Report 1.0.doc , 26 April 2006.
4. Blackwell, A.F., and Hague, R., Autohan: An architecture for programming in the home. In *Proc. IEEE Symposia on Human-Centric Computing Languages and Environments 2001*, 150-157.
5. Eisenberg, M., Tangible ideas for children: material science as the future of educational technology, In *Proc. IDC 2004*, ACM Press.
6. Horn, M.S., Jacob, R.J.K., Tangible programming in the classroom: a practical approach. In *Proc. CHI 2006,* ACM Press.
7. McNerney, T.S., From turtles to Tangible Programming Bricks: explorations in physical language design. *Personal Ubiquitous Computing*, 8(5), Springer-Verlag (2004), 326–337.
8. Smith, A.C., Tangible cubes as programming objects. In *Proc. ICAT 2006*, ISBN-13: 978-0-7695-2754-3, IEEE Computer Society (2006) Order Number P2754, 157-161.
9. Suzuki, H. and Kato, H. Interaction-level support for collaborative learning: Algoblock–an open programming language. In *Proc. CSCL 1995*, Lawrence Erlbaum (1995).
10. Valente, A., Explorations in theoretical computer science for kids (using paper toys), In *Proc. IDC 2004*, ACM Press.
11. Wyeth, P., Purchase, H.C., Tangible programming elements for young children. *Ext. Abstracts CHI 2002*, ACM Press (2002), 774–775.
12. Wyeth, Peta., Purchase, H.C., Using developmental theories to inform the design of technology for children, In *Proc. IDC 2003*, ACM Press.
13. Zuckerman, O., Arida, S., Resnick, M., Extending Tangible Interfaces for Education: Digital Montessori-inspired Manipulatives. In *Proc. CHI 2005*, ACM Press