Boda Blocks: A Collaborative Tool for Exploring Tangible Three-Dimensional Cellular Automata

Leah Buechley and Mike Eisenberg
Department of Computer Science, University of Colorado at Boulder, Boulder, CO USA
buechley@colorado.edu, duck@colorado.edu

Abstract: Construction kits like traditional building blocks provide excellent media for face-to-face collaborative interaction. The complexity and expressive power of these kits are increasingly being augmented with computational elements like controllable lights, motors and sounds. This paper introduces a computationally enhanced set of building blocks, *Boda Blocks*, which allows for collaborative interaction through the construction and programming of tangible three-dimensional cellular automata. We provide a brief introduction to computationally enhanced construction kits, describe the Boda Blocks system and report on the results of a preliminary user study.

Introduction

Construction kits constitute a venerable tradition in the design of educational toys. Kits allow for a relatively unselfconscious medium in which children can work both alone or in collaboration; unlike (on the one hand) explicitly solitary activities such as reading, or (on the other) explicitly group-oriented activities such as board games, construction kits allow for both individual contemplation and group discussion.

Researchers and designers have recently begun to integrate computation and construction kits in various ways. This trend began in the 1970s when architectural researchers (Frazer 1994; Aish 1979) built beautiful kits for architectural modeling. Such kits have entered the educational realm more recently (Eisenberg et al. 2002; Resnick et al. 1996). The Active Cube (Watanabe et al. 2004) and Topobo (Raffle 2004) projects, as well as the work of Wyeth et al. (2002) and Zuckerman et al. (2005) are especially inspiring examples of educational kits.

Boda Blocks

This paper describes a computationally-enhanced construction kit, *Boda Blocks*. The kit, shown in Figure 1, is a set of 16 luminescent cubes that can be arranged in a variety of configurations and programmed, via companion software, with cellular automaton rules to display dynamic three-dimensional patterns of light and color.

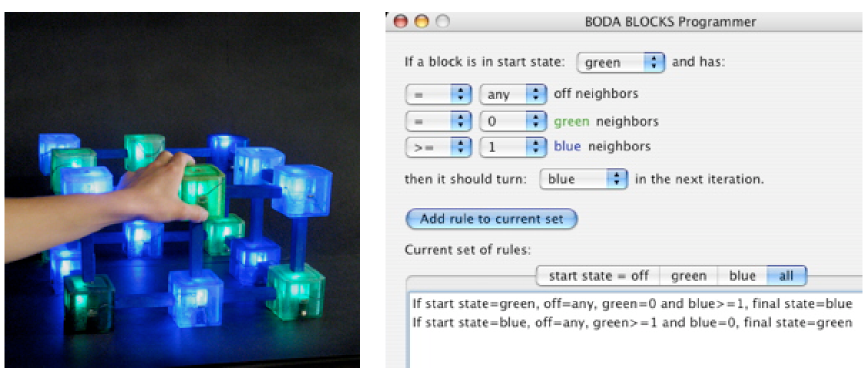

Figure 1. Boda Blocks: the physical kit and the programming interface.

Each block is always in one of three states: blue, green, or off. Boda Block constructions are in one of two modes at any given time. In the *interacting mode*, users can program the blocks with cellular automaton rules and use the switches on individual blocks to set a construction's initial state. (Pressing a switch on a block cycles its state from blue to green to off.) In the *executing mode*, constructions evolve according to the cellular automaton rules with which they have been programmed. Users can change the physical configuration of a construction at any time by adding blocks to it or removing blocks from it. Thus, there are four actions that are undertaken with the kit during its two modes: constructing, programming, initializing and observing.

Cellular automata are mathematical models that explore how local rules, executed in discrete time steps, can result in complex global patterns (Ilachinski 2001). Cellular automata are usually implemented on a computer screen on a grid of colored "cells" (the squares in the grid). Cells can display their "state" through their color, and can communicate with their immediate neighbors (usually the squares surrounding them). As a cellular automaton develops in discrete time steps, each cell computes what its state will be at time t+1 based on its state and the state of its neighbors at time t; the patterns that evolve in a particular cellular automaton are dependent on the initial state of the grid.

Cellular automaton rules for the Boda Blocks are defined using the cellular automaton programming software, shown in Figure 1. This software allows users to experiment with a specific class of cellular automaton rules. This class of rules, termed outer-totalistic (Ilachinkski 2001, p. 45), allows users to specify a block's next state based on its current state and the collective behavior of its neighbors.

Any blocks that are part of a construction will be reprogrammed with the new rule set when it is sent; blocks that are not part of the current construction will not be reprogrammed. We would like to call attention to the fact that this design allows for the interesting possibility of constructions with heterogeneous rules; that is, a single construction may contain blocks that have been programmed with an assortment of rules. During the executing phase, a construction functions as a parallel computer. Each block simultaneously and independently communicates state information to and receives state information from its neighbors (any blocks that are attached to it) and then independently updates its state based on its personal rule.

Preliminary User Testing

We recently held the first user test of the Boda Blocks system. The test took place in our lab with a group of four children ages 11-14, two females and two males. The questions we were most interested in answering at this preliminary stage were: "Is the kit usable?" and "Is the kit engaging; is it capable of maintaining sustained user interest?" Our principal means of assessment for these questions was observation. We took several photographs, and noted down interesting quotes, but otherwise did not impose on our users. The remainder of this section will report on this initial study, highlighting issues surrounding collaboration.

The four children spent approximately one hour interacting with the Boda Blocks system. A pile of blocks and connectors was placed on a table, and users could sit around three sides of it to interact with the system. A laptop was provided for the block programming activity. To begin the Boda Blocks session, a workshop leader (the first author) explained the Boda Blocks phases and activities: constructing, programming, initializing and executing. The participants then quickly began building constructions with their pile of blocks and connectors. They experimented with a few different configurations, but after building the tower form seen in Figure 2, did not return to the construction activity. The rest of the session was spent experimenting with different rules and initial configurations set on this form.

Figure 2. Users interacting with Boda Blocks.

Once the tower was built and connected to the computer, each child was given the opportunity to program the construction in turn. While one child manipulated the programming interface, the other children either worked on setting the initial state of the blocks, observed executing behavior, or assisted the programmer with her task.

Figure 2 shows images of typical interaction: in the left image, two children interact with the kit simultaneously while one observes; in the right image, two children collaborate on the programming activity while a third works on setting the blocks' initial configuration.

The interaction was immediately highly collaborative and remained so throughout the session. Participants kept up an ongoing and lively discussion about the rules, initial states and dynamic patterns. Interaction between the programmers and block manipulators was coordinated and productive. Initially, the participants seemed to somewhat randomly experiment with rules and behavior, but as they observed interesting patterns—one that died out for example—they attempted to modify the rules and starting state to obtain intentional results—a pattern that did not die out, or one that oscillated.

The users also became immediately engaged in predicting the behavior of their constructions. Given an initial state and a rule, they would almost always embark on a period of collective speculation and initial state modification before executing their construction. Often the observed behavior would confound their expectations, usually because they had neglected to fully think about all three dimensions. For example, expecting an oscillating pattern on a horizontal plane, users were surprised when their construction evolved into a seemingly random flashing pattern. After first protesting that "the blocks are broken!" the users then reset the initial configuration and realized that they had neglected to take vertical interactions into account. This pattern of experience was repeated many times throughout the session, with users periodically experiencing three-dimensional "aha" moments.

At the end of the session we felt that we had achieved a good informal indication that our kit was indeed useable and engaging. Since our assessment was based entirely on observation we cannot be certain that each of the participants completely understood the system and felt comfortable with it, but our initial findings were positive. As has been detailed, the users built their own constructions successfully and were quite capable of interacting with the system with very little intervention. All of the users seemed to quickly understand the relationship between programs, the blocks' starting states and the patterns that they observed in the blocks.

We also found that the kit sustained the users' interest throughout the one hour session. Somewhat to our surprise, the children remained independently engaged and actively explored different rules and behaviors without outside prompting. As was stated above, we did periodically intervene to make sure that each child had a turn at the computer to program the blocks, but other than that we did not structure the experience. Participants seemed to genuinely enjoy working with the kit. Indeed, the participants were reluctant to leave the workshop at the end of the session, as they were immersed in attempting to get their construction to exhibit a desired behavior.

References

Aish, R. (1979). 3D Input for CAAD Systems. *Computer-Aided Design,* 11:2, 66-70.

Eisenberg, M., Eisenberg, A., Gross, M., Kaowthumrong, K., Lee, N., and Lovett, W. (2002). Computationally-Enhanced Construction Kits for Children: Prototypes and Principles. In *Proceedings of the Fifth International Conference of the Learning Sciences*, 23-26.

Frazer, J. (1994). *An Evolutionary Architecture*. Architectural Association: London.

Ilachinski, A. (2001). *Cellular automata: a discrete universe*. World Scientific. River Edge, NJ.

Raffle, H (2004). Topobo: A Constructive Assembly System with Kinetic Memory. In *Proceedings of CHI 2004*.

Resnick, M., Bruckman, A., and Martin, F. (1996). Pianos Not Stereos: Creating Computational Construction Kits. *Interactions*, 3:6.

Watanabe et al. (2004). The Soul of ActiveCube – Implementing a Flexible, Multimodal, Three-Dimensional Spatial Tangible Interface. *ACM Computers in Entertainment*, 2:4.

Wyeth, P. and Wyeth, G. (2002). Electronic Blocks: Tangible Programming Elements for Preschoolers. In *Proceedings of CHI 2002*.

Zuckerman O., Arida S., Resnick M. (2005). Extending Tangible Interfaces for Education: Digital Montessori-Inspired Manipulatives . In *Proceedings of CHI 2005*.

Acknowledgments

We would like to thank the University of Colorado's Science Discovery program, Nwanua Elumeze, Yingdan Huang, Jaime Catchem, Mark Gross, Clayton Lewis, and all of our workshop participants. This work was funded in part by the National Science Foundation award no. 0326054.