Remix and Robo: sampling, sequencing and real-time control of a tangible robotic construction system

Hayes Raffle and Hiroshi Ishii
Tangible Media Group
MIT Media Lab
77 Massachusetts Ave.
Cambridge, MA 02139, USA
{hayes, ishii} @media.mit.edu

Laura Yip
Massachusetts Institute of Technology
77 Massachusetts Ave.
Cambridge, MA 02139, USA
yipla@mit.edu

Abstract:
We present *Remix* and *Robo*, new composition and performance based tools for robotics control. Remix is a tangible interface used to sample, organize and manipulate gesturally-recorded robotic motions. Robo is a modified game controller used to capture robotic motions, adjust global motion parameters and execute motion recordings in real-time. Children use Remix and Robo to engage in (1) character design and (2) competitive endeavors with Topobo, a constructive assembly system with *kinetic memory*.

Our objective is to provide new entry paths into robotics learning. This paper overviews our design process and reports how users age 7–adult use Remix and Robo to engage in different kinds of performative activities. Whereas robotic design is typically rooted in engineering paradigms, with Remix and Robo users pursue cooperative and competitive social performances. Activities like character design and robot competitions introduce a social context that motivates learners to focus and reflect upon their understanding of the robotic manipulative itself.

Author Keywords
Digital Manipulative, Education, Toy, Learning, Children, Modular Robotics, Controller, Tangible Interface.

ACM Classification Keywords
H.5.2: Information Interfaces and Presentation: User Interfaces.

INTRODUCTION

Walk into a toy store today, and you will find toys that appeal to all ranges of children's motivations. Many appeal to children's desires to perform and act out their ideas through a surrogate object: dolls and puppets stand in for characters and people, remote control vehicles empower children with control of a machine, and video games provide surrogate characters that empower children to navigate through fantasy worlds. These games and toys appeal to children's sense of performance, fantasy and adventure, but the lack means of design and invention that are known to foster creativity and learning [13].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

IDC '07, June 6-8, 2007 Aalborg, Denmark
Copyright 2007 ACM 978-1-59593-747-6/07/0006... $5.00

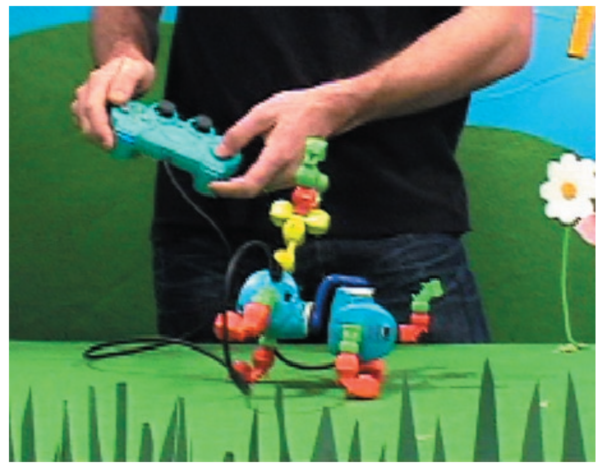

Figure 1. Robo, a modified video game controller, is used to perform a sequence of kinetic recordings with a robotic moose.

Building toys like LEGO are part of a tradition to use manipulatives for hands-on learning [4]. Research into digital manipulatives has sought to combine physical manipulatives and computer programming for children to design creations that have behavior [17]. Educators have found that performative events like robot design competitions motivate children to learn principles of robotic control. But while autonomous control demonstrates a deep understanding of the design of synthetic behavior [13], a building-block approach to control lacks means to reflect children's improvisational performance.

Tangibles have sought to make the programming process more direct—and easier for young children who are not adept with symbolic abstraction—by developing various means for hands-on programming [6, 12, 21]. But while hands-on programming has enabled children to more easily invent objects with behavior, tangibles provide limited tools to control that behavior.

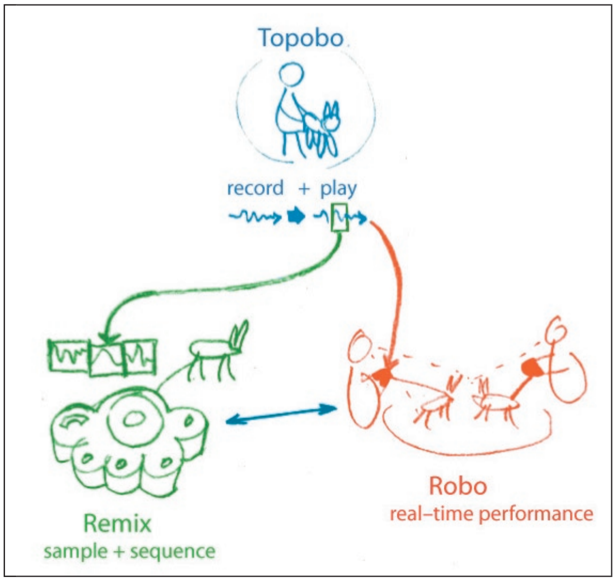

Figure 2. Remix, Robo and Topobo concepts.

Our work aims to provide flexible and accessible tools to control robotic motion created with a tangible interface. We apply an interaction model from the audio domain to the robotic domain: the model *Record, Sample, Sequence,* and *Perform* is used to compose robotic motion, rather than music. While people typically associate sampling and sequencing with music genres like hip-hop, we explore how this interaction model can make robotics design more intuitive, playful and performative for children.

We will discuss design investigations in which we leverage the visual language of performative interfaces like video game controllers and deejay turn tables to help children quickly understand the kinds of play Remix and Robo can support. Through tests with various age users, we will evaluate their usability for applications that are both artistic (e.g. robotic puppet shows) and athletic (e.g. robot competitions).

REMIX & ROBO: NEW TOOLS TO CONTROL A TANGIBLE ROBOTIC CONSTRUCTION KIT

Hypothesis

We hypothesize that providing means for capturing, organizing and controlling movement in real-time will help children analyze, understand, and refine the design of their robotic creations.

Approach

Remix and *Robo* are controllers children use to sample and sequence the movements of a *Topobo* creation. They are designed to support children's narratives and improvisational performances with Topobo (fig. 2).

Topobo (fig. 3) is a 3d constructive assembly system with *kinetic memory*, the ability to record and playback physical motion. Children use Topobo to design and animate playful robotic creations. A child may build a moose with Topobo, twist the moose in her hands to animate the creature, and then watch the moose replay these motions by itself. The same way stacking blocks helps children learn how stone buildings stand up, animating Topobo helps children learn how animals walk [15].

Topobo includes Passive (structural, plastic) and Active (modular robotic) components. Topobo Actives record motions imparted to them, and later replay the motions by driving motors inside the Actives. Actives usually replay the exact motions that are physically imparted to them during recording. But if a child creates a recording with a Queen Active, all connected Actives will mimic the Queen's motions. This enables a form of centralized control.

Remix is a tangible sampler/sequencer to capture, adjust and recompose Topobo motions.

Robo is a modified video game controller that a child will use for real-time performance of his Topobo creation.

Scenario

A child builds a Topobo ant and creates a simple kinetic recording by moving the ant in his hands. The ant replays the child's movements by itself, in this case walking around on a table. The child then uses Remix (fig. 4) to capture a favorite segment of this walking motion for later playback. He attaches Robo (fig. 3) to the ant and adjusts the walking motion he has just captured with Remix, controlling the motion's speed, scale and direction in real-time.

A parallel to other media composition tools

Topobo, Remix to Robo can be compared to video performance tools: in video performance, a camera will be used for pure data capture (Topobo), an editing suite will be used to sample, sequence and organize a library of video clips (Remix), and video-jockey tools will be used to perform video mixing spontaneously (Robo). Such tools are designed to be used interchangeably, have some functional overlap (e.g. one could conceivably video-jockey with raw unedited video data), and are tailored to support different usage patterns.

RELATED WORK

Our design investigation is informed by research in interface design, digital construction kits, and audio/visual sampling and performance equipment.

Tangibles and abstraction

Researchers have invented various means for hands-on "programming." Materials with memory—like brushes children use to paint with kinetic ink and sound [18], and cars that remember the way children have moved them [6]—can give children physical means to author dynamic behavior. Because of tight coupling of control (input) and representation (output), children of even very young ages were able to program compositions without needing to learn traditional programming languages.

One challenge has been to create tangible means of assigning and manipulating data. One approach has been to use a physical construction kit to embody and represent a control structure. Tangible programming was explored with AlgoBlocks [19], which provided physical blocks to create tangible, procedural programs on a computer. FlowBlocks allowed children to explore systems dynamics concepts like probability and feedback [23], and children as young as 3 assembled Tangible Programming Blocks [21] to create simple procedural programs. Mediablocks, which inspired the design of the Remix tokens, were wooden blocks that referenced data that resided on a network. Manipulating the blocks could perform various manipulations to the data such as copying and printing a document [20].

Mapping is a general design problem for all of this work: how should an abstract idea be represented and controlled? What is the proper "level of abstraction" to represent? Designers continue to grapple with questions of tight and loose coupling in educational system design [5].

Digital Construction Kits

Much of the work in digital construction kits [9, 12, 16, 17, 19, 23] has focused on science and engineering learning. Digital manipulatives like Mindstorms illustrate how toys can stimulate science and engineering activities through application of engineering based tools (e.g. gears, levers, motors, wires, procedural code). Children often choose such tools if they are already motivated by science and engineering activities [12]. These tools lean on scientific knowledge and interests that kids already have, before they even use the systems. Children's desires to perform and compete have sparked a number of robotic design competitions, e.g. FIRST robotic competition, although most focus on autonomous control and are based on logical (rather than dramatic) styles of learning [7].

Audio samplers and mixers

The emergence of sampler/sequencers and performative mixing devices (like deejay turntables) in the audio domain has inspired us to apply similar techniques to robotics. Where audio is concerned with recording and composition of recorded analog sounds, we envision robotics control as a recording and composition of recorded gestural motion.

DESIGN OVERVIEW

With Remix and Robo Topobo we present a modular system that offers both the benefits of hands-on programming and the flexibility of more abstract controllers. We consider this to be an evolution in tangibles for learning, where a tangible interface's coincident input/output [8] model is extended with the addition of controllers for sampling and sequencing simple programs. This provides a limited form of abstracted control that makes traditional computer interfaces flexible.

ROBO DESIGN AND USE

A child first builds a creation with Topobo. To record a motion, she presses a button on Topobo and moves Topobo in her hands as desired. She presses the button again to stop recording and start a looping playback mode. She can save the recording with Robo (fig. 3), a customized game controller, by pressing Robo's "record" button and then press one of its four "playback" buttons to assign the entire recording to that button.

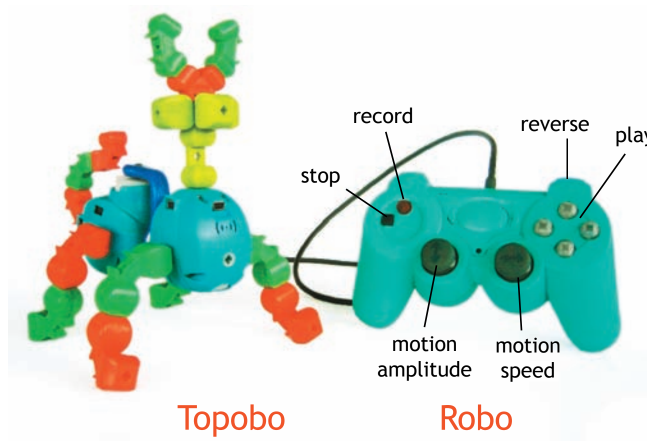

Figure 3. Robo can save motions and control their playback.

When a creation is in "playback" mode, joysticks adjust speed and amplitude of the motions. Continuously depressing a "reverse" button will cause a recorded motions to play backwards. (Reverse may cause a creature that walks forward to walk backwards.) Users of Robo can spontaneously control Topobo motions in real-time to create original sequences of movements.

REMIX DESIGN AND USE

While Robo maps an entire gestural recording to a button for later playback, with Remix a user can sample (record) arbitrary amounts of continuous motion with a wooden token. She can sequence up to four tokens (representing different motion records) for looping playback, while controlling the speed and direction of playback.

A user will first build a creation and set it into looping playback motion. To sample a piece of the motion, she will place a wooden token in Remix's "record" slot and push Remix's "record" button. A red light signifies that Remix is recording. To stop recording, the user will push the record button again or remove the token from the record slot.

To playback this motion, the user will move the wooden token to one of four slots in a donut-shaped "playback arena," and press Remix's "play" button. Green lights beside the token signify that it is mapped to a recording, and a red marquee light advances as the recording plays (fig. 4). The user may turn a green knob on Remix to change the rate of playback, or push a button to change the direction of playback. She can sequence up to four distinct recordings to loop in the playback arena.

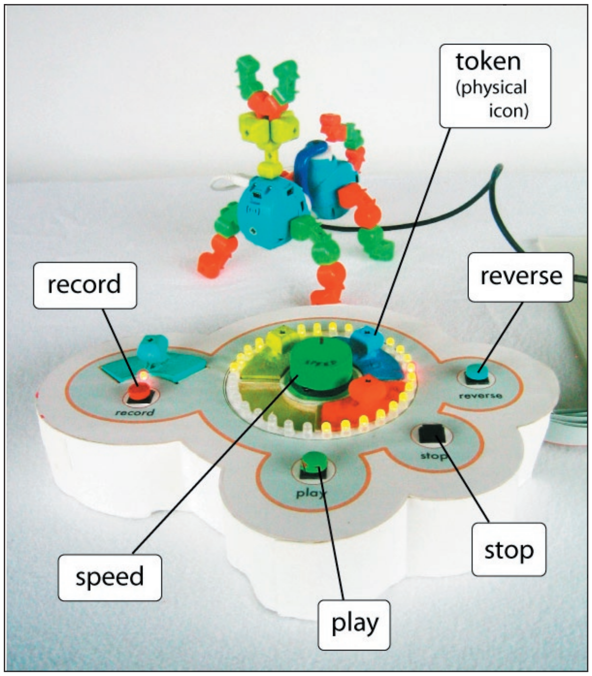

Figure 4. Remix can sample and sequence motion. Wooden tokens represent recordings. Because Remix can simultaneously control motion recording and playback, records can be concatenated, duplicated and altered in different ways.

Manipulating records

Remix records *whatever Topobo is doing when Remix's record light is on*. This enables a number of possibilities.

Partial or multiple loop saves:

A user may gesturally create a very long, changing series of footsteps for a walking creature. On playback, she realizes that a very small section of the recording produces satisfactory walking. She uses Remix to capture only the effective steps. Looping playback of this new recording creates a continuous, repeatable walking movement.

Copying records

A recording is captured with a token and set into playback. A second token is used to record and duplicate the movement.

Saving modulations

Reverses and subtle changes to speed can be saved by recording playback motions that are controlled from Remix. For instance a user first creates a slow gestural recording with Topobo. He will then use Remix to capture the movement, and will use Remix to playback the recording at twice the original speed. This faster playback is again captured with Remix, and then played back at twice its recorded speed (four times the speed of the original gestural record).

Nesting recordings

A user will map a walking motion to the Red token and a dancing motion to the blue token. They are sequenced in the playback arena, and a green token is used to sample (capture) Topobo's performance of both records in series. The green token now references concatenated copies of both the walking and dancing motions.

Improvising

A user may assign several different motions to different tokens. While keeping Remix in "play" mode, he can rapidly place and remove the tokens in the playback arena to force his creation to spontaneously play any single record. The effect is similar to pushing buttons on Robo.

USING REMIX AND ROBO INTERCHANGEABLY

Remix and Robo reference identical nonvolatile memory banks inside Topobo Actives. This allows users to interchangeably use Remix and Robo to control the same creation. For instance, a child may use Remix to accurately sample specific sections of gestural recordings, and then use Robo to perform those motions more spontaneously. Or, a Robo user may discover that a particular sequence of movements creates a desirable effect, and then use Remix to copy that sequence into a single record.

DESIGN PROCESS

Remix and Robo evolved over two years with graphic, industrial and interaction designs refined in response to user feedback.

Remix

Initial Remix designs were conceived on paper and several months were spent writing firmware. We built a GUI prototype to explore questions of mapping and determine how much abstraction was appropriate for the controller. However, the user experience at a GUI was so different than Topobo play that the simulation did not help us to evaluate our basic questions.

Paper models of tangible controllers allowed us to quickly test for usability, size, and aesthetics for Remix. For our final Remix design, we connected a foam-core and paper prototype with embedded LEDs and switches to breadboarded electronics.

Robo

Robo designs began on paper with storyboard prototyping of interactions and play-acting of its conceived function with novice Topobo users. For our final implementation we modified a standard game controller by removing many functions and creating Topobo-compatible embedded circuitry with backlit buttons.

Robo evolved from experiments with Remix. Some users found that viewing Remix's intricacies and lights distracted them from viewing the movements of their creations, and one objective with Robo was to create a performance–based controller whose operation required only a user's kinesthetic sense. When children can quickly learn to operate the device through touch alone, the child's eyes and ears are free to focus on the Topobo creations themselves, or on other children who are participating in the activity.

Technical approach

Topobo is a distributed system comprised of individual robotic elements each with their own internal parameters (e.g. speed) that define their behavior. Topobo Actives have embedded motors and electronics to manage power distribution, motor control, and a custom distributed peer-to-peer network. Robo and Remix allow for centralized global control of Topobo so that all Topobo Actives share a common set of parameters (fig. 5). All computation is embedded and distributed among the toys, and external power is supplied to a single element for distribution to all others in a creation.

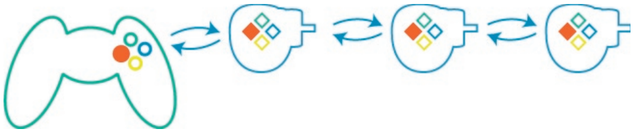

Figure 5. Robo and Remix allow for globalized, central control of a decentralized system.

USER EVALUATIONS

Our qualitative evaluation is designed to address how controllers can support children to analyze and refine their robotic designs. Developing proficiency with Topobo takes all users a minimum of one or two hours of play. Creating quirky and fun Topobo creatures is easy, but understanding the dynamics of Topobo behavior is extremely difficult.

We assumed that a deep understanding of all tools would require several further hours of practice with them. Therefore, we conducted our study with a wide range of users to capture their usability for people at different levels of development and expertise.

We worked with 16 users from age 4–adult to evaluate the design, usability and function of the interfaces. We sought to understand how users would integrate Remix and Robo into their design and problem solving strategies, and how Remix and Robo might support or interfere with iterative design strategies that successful Topobo users applied in previous studies [15]. The evaluations revealed a variety of styles of performative play.

Methodology

Groups of 2-5 users worked with Topobo simultaneously, in a playful lab environment. A researcher explained the Topobo system, showing how parts could be assembled, gesturally programmed to move, and adjusted to achieve different kinds of behavior. Walking creatures and a video of Topobo locomotion were quickly demonstrated. A researcher then explained how Robo and Remix worked. Users were asked to explore the Topobo system and design a character.

All users elected to work for a minimum of three hours. Typically, a user spent one hour building various creations with Topobo, exploring different kinds of movement and trying to understand how to make a robot walk. Following explorations incorporated the controllers in various ways. If a user seemed to be confused a researcher may have offered suggestions. Some users returned on multiple sessions. One eleven year old reportedly discussed his work with his mother for five continuous days in between play sessions. All sessions were video taped and later analyzed and coded by a researcher for analysis.

Competitive endeavors among users age 7 to Adult

Desires to perform and compete can motivate children to play with our system. Young boys, in particular, love to get together and act out battles with their action figures and other toys. This inspired us to organize "Battle Bots" competitions. The competitions posed a steep challenge: people often discover that their Topobo creations may "walk," but creating Topobo creatures that walk predictably—and can be controlled—is extremely difficult. We hoped that Battle Bots may provide a socially and emotionally motivating reason for boys to develop mastery with Topobo locomotion.

Jonathan, 7 years old, plays battle bots

"THEY'RE GREAT! GREAT! This is better than action figures... better than video games. Why? It's just funner, I don't know....can we do a little more fighting?"

Jonathan and his friend have been playing with Topobo for three hours, spending the first 90 minutes with free play and experimentation. Both boys are paired with an adult (parent or researcher) because they can discover controllable locomotion much more quickly with the support of an older peer or adult [22].

An adult programs a robot to walk and shows Jonathan how to control the walking with Robo. Jonathan immediately wants to battle his friend, who is not ready. Jonathan then sets himself to learn to use Robo, and gesturally records and captures his own recordings, improving on the adult's design.

Battles ensue. For an hour, the boys compete, redesign, and compete again. A researcher asks: "Was [Robo] confusing at

Figure 6. Jonathan reaches in to a battle to redesign his robot.

first?" Jonathan: "Yeah, but then it's easy now. You needed to get how to control it. It would have been hard to figure out if no one was teaching me."

Jonathan loves the idea of Battle Bots. "When I want to protect myself I want to do the kicking move [acts out kung-fu moves with his body]." But Robo became motivating for Jonathan only when he could successfully control a creature someone else had designed. For his age and skill level, Jonathan needs more time to develop controllable locomotion himself.

Topobo Battle Bots may be too difficult a task for a young child to engage in alone, unless he is provided with specific examples that allow him to feel successful very quickly. Older children may succeed more easily. This feeling of immediate success seems necessary to motivate a child to develop mastery, but Topobo play typically leads to quirky robots with amusing motions, not vehicles with highly controllable locomotion. However, Jonathan's overwhelming excitement at the idea of battle bots suggests that researchers should establish techniques to support dramatic play with a digital manipulative. The challenge remains to remove the "speed bumps" associated with learning how to transform simple, playful designs into understandable and controllable ones.

Character Design for storytelling

Jasper (age 11) demonstrates a flying "phoenix" that can flap its wings in different ways. He animates the wings and then practices flying it in the air by waving its entire body around. He then hangs it from the ceiling and experiments with recreating his earlier gestural motions with Topobo.

An adult suggests that Jasper picture his Phoenix in a movie, flying over a moving background. Jasper immediately imagines his bird diving for a mouse, and uses Robo to capture a diving posture he invents. Then, Jasper proceeds to create and capture several different recordings. Some are static postures—akin to an animator's keyframes—and others are dynamic recordings. Jasper demonstrates his Phoenix's range of movements in anticipation for a story.

Jasper is most excited to animate his creation when he actively imagines an animated background behind it. This indicates that while Robo provided tools for performance, his activity was lacking a context.

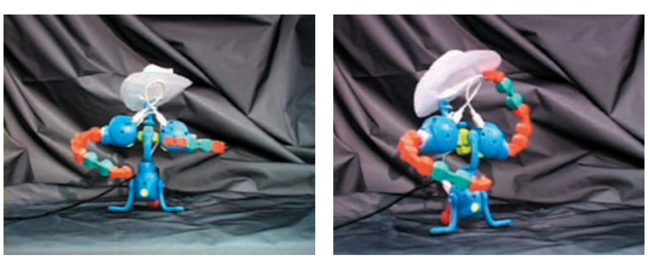

Figure 7. Bob uses Robo to direct his cowboy to show anger (left) and respect (right).

Robotic Puppeteering

This is walking. This is anger. And this is respect... With a few moves, you have enough expression to do a whole movie.

Bob, an experienced adult animator, uses paper to decorate his cowboy creature (fig. 7) and experiments both with continuous animation and with "keyframe" recording using Robo, by recording still gestures. "[Keyframes are] a little more 'real time.' I was constantly pushing buttons to do everything, which was satisfying."

Robo and Remix allow Bob to create characters with a wide range of expressive range. "You couldn't do character animation without these controllers. This is a different problem than getting something to walk.... It would be interesting to work from a script, because I bet we could get something rapidly across." Bob suggests applying the interface to expert puppeteering [3].

DISCUSSION

Our evaluation confirmed our hypothesis that providing tools to control robotic behavior supported children to analyze and refine their designs. All users related to the system first as a building toy and secondly as a robotic vehicle, a character, or a puppet for narrative performance. The introduction of Robo and Remix did not alter the basic character or play pattern with Topobo, evidenced by all users' intense interactions with Topobo prior to employing the controllers. A user explains, "why didn't we use the controllers in the beginning? We needed a creature first!"

Controllers supported users' individual interests

Users who had developed successful characters employed the controllers in various ways—competition, performance, global controls for investigating physics dynamics—depending on users' personal interests. Some people used Remix and Robo to refine their gestural designs, for instance to create more successful locomotion. Others used the controllers to apply their work to a secondary application domain, such as narrative performance. For most users, the controllers played into people's existing hands-on design process, allowing people to adjust and understand abstract variables for motion, and to reflect on their own design and thinking.

One challenge with Topobo is to predict how a gestural recording will make a creature behave once it is set on a table, reacting to friction and gravity rather than to the movements of one's hands. For several users, controllers were a convenient way to debug motions, since variations to movement could be observed while they were being created.

From direct to remote control

Departure from Topobo's tightly coupled input/output model is a necessary compromise because tight i/o coincidence is very limited. To accommodate increasingly skilled users, an interface must reflect users' thinking at multiple levels of abstraction. Our goal with Robo and Remix is to help children climb a mountain of ideas about dynamic physics, helping them understand how and why moving structures like animals behave the ways they do.

The original Topobo system includes "Queens," special orange Actives that instruct all connected Actives to mimic the motion recorded with the Queen. Some people use the Queens as remote controllers to program the behavior of a creature, observing the movement of the creature as they are programming it. In comparison to Queens, Robo and Remix facilitate less direct interactions with Topobo. Their benefit is greater flexibility and a higher degree of control.

We observed that users ages 7–adult found Remix and Robo to be an important part of their mastery of new ideas. According to one adult who rapidly learned how to achieve his goals with Topobo, "Robo and Remix show that the system does actually develop with you. Even as you get smarter, you can still learn something with Topobo. [Remix and Robo] are something you use in different ways as you get better at it."

Expressive and exploratory learning

Work in developmental psychology suggests that effective learning should involve both *expressive* activity, where the tangible represents or embodies the learner's behavior (physically or digitally), and *exploratory* activity, where the learner explores the model embodied in the tangible interface [1, 2, 10]. The challenge is to engage the learner in an immersive and exploratory activity, and then help him to think about and understand what he has done.

When working with manipulatives, we believe that controllers may facilitate this process: they encourage a physical "stepping-back" and observing of one's work (fig. 8), and by carefully mapping controls to concepts that underlie a system's behavior, they can make important concepts manipulable and salient for users.

Physical controllers versus GUI controllers:

For hands-on learning the road to abstraction may not lead to the GUI. We asked users over the age of 10 if they would have preferred a graphical interface to Robo and Remix. *All* of the users said no. While one user suggested that a GUI could enable people to precisely represent and control the motor movements in their Topobo creations, she thought it might be distracting.

People said that specialized controllers "fit the Topobo system" better, that they liked the Remix tokens and enjoyed moving them around, and that the controllers were easy to use. Several users liked that they didn't need to "use a computer" to play with the system. One user commented that the controllers seemed more similar to the basic Topobo system because what the user did with his hands seemed to be more directly related to what Topobo was doing. "They're somewhere in between a tangible interface and a graphical interface."

CONCLUSION

Remix and Robo are specially designed controllers that enable sampling, sequencing and real-time modulation of gesturally-recorded robotic motion. The controllers motivate and support users to learn about dynamic physics concepts like center of mass and dynamic balance through focused play with Topobo. Some users employ the controllers as part of an iterative design process, where global control of variables allows users to better understand why their creations behave as they do. Other users focus on learning how to make Topobo perform predictable and controllable behaviors specifically to participate in new applications like competition or storytelling. Remix and Robo appeal to users who are (1) interested in model making with Topobo and (2) have an interest in dramatic social interactions. Remix and Robo support basic playful learning with Topobo and provide valuable tools to both novice and expert users.

Figure 8. Controllers encourage people to step back and reflect on their experiences.

A major problem in introducing computing (and embedded computing in particular) to kids stems from the disconnect between the physical and computational realms, or the "layers of abstraction" that separate them. This paper presents a system that has eliminated the distance between computation and the "real" world" while providing possibilities for sophisticated activities - intellectual, playful and physical.

We pursue new approaches to constructivist education, or learning by actively experimenting with ideas in the world [14]. In hands-on education, a child may build something, and that thing enters the child's social context [22]. In some situations, a child may wish to *design* or *control* his creation's behavior in that context. Specialized controllers are one tool children may use to design behavior for their creations, in a way that captures the spontaneity and improvisational spirit that radiates from a child's experimentation and play.

ACKNOWLEDGMENTS

Thanks to Mike Fleder, Benazeer Noorani, Mitchel Resnick, and the members of the Tangible Media Group.. This work has been supported by the LEGO Group, the Microsoft Corporation, and the MIT Media Lab's Things That Think consortium.

REFERENCES

1. Ackermann, E. "Enactive Representations in Learning: Pretense, Models, and Machines." In Bliss, J., Light, P. and Saljo, R. eds. *Learning Sites: Social and technological contexts for learning*, Elsevier, 1999, 144-154.
2. Ackermann, E. Perspective-taking and object construction: two keys to learning, in Kafai, Y. and Resnick, M. eds. *Constructionism in practice: designing, thinking, and learning in a digital world*, Lawrence Erlbaum, Mahwah, NJ, 1996, 25-35.
3. Bell, John. *Strings, Hands, Shadows: A Modern Puppet History.* Detroit Institute of Arts (2000).
4. Brosterman, N. *Inventing Kindergarten*. New York, Harry N. Adams, Inc, 1997.
5. Fernaeus, Y. and Tholander, J. Finding Design Qualities in a Tangible Programming Space. CHI 2006.
6. Frei, P. curlybot: *Designing a New Class of Computational Toys*. Master's Thesis, Massachusetts Institute of Technology. 2000.
7. Gardner, H. (1983). *Frames of mind: The theory of multiple intelligences*. New York: Basic Books.
8. Ishii, H. and Ullmer, B. Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms. *Proceedings of CHI 1997*, ACM Press, (1997), 234-241.
9. LEGO Mindstorms. http://mindstorms.lego.com.
10. Marshall, P., Price, S., and Rogers, Y. Conceptualising tangibles to support learning. *Proceedings of Interaction Design and Children*, Preston, England, July 1-3, pages 101-110. 2003
11. Montessori, M. *The Montessori Method*. Translated from 1912 original by Anne George. New York: Schocken Books (1964).
12. O'Malley, C. Fraser, D. "Literature Review in Learning with Tangible Technologies." NESTA Futurelab series. Report 12, 2005. http://www.nestafuturelab.org/research/reviews/reviews_11_and12/12_01.htm
13. Papert, S. *Mindstorms: Children Computers and Powerful Ideas*. Cambridge, Massachusetts: Perseus Publishing, 1980.
14. Piaget, Jean. *The Grasp of Consciousness*. Cambridge: Harvard University Press, 1976.
15. Raffle, H. Parkes, A. Ishii, H. Topobo: A Constructive Assembly System with Kinetic Memory. *Proceedings of CHI 04*. ACM Press, (2004), 869-877.
16. Raffle, H, Parkes, A. Ishii, H. "Beyond Record and Play. Backpacks: Tangible Modulators for kinetic Behavior." *Proceedings of CHI 2006*, ACM Press, (2006) 427-436.
17. Resnick, Martin, Berg, et al. Digital Manipulatives: New Toys to Think With. Paper Session, *Proceedings of CHI 1998*, ACM Press, (1998) 281-287.
18. Ryokai, K., Marti, S., Ishii, H. I/O Brush: Drawing with Everyday Objects as Ink, in *Proceedings of CHI 04*.
19. Suzuki, H., and Kato, H. "AlgoBlock: A Tangible Programming Language." In Proceedings of the 4th European Logo Conference, August 1993. pp. 297-303.
20. Ullmer, B., and Ishii, H. (1999) MediaBlocks: Tangible interfaces for online media (video). In *Extended Abstracts of Conference on Human Factors in Computing Systems* (CHI1999). 31-32. ACM Press.
21. Wyeth, P. and Purchase, H. Tangible Programming Elements for Young Children. *Extended abstracts of CHI 2002*, ACM Press, (2002) 774-775.
22. Vygotsky, L.S. (1978). *Mind in Society.* Cambridge: Harvard University Press.
23. Zuckerman O., Arida, S., and Resnick M. (2005). Extending Tangible Interfaces for Education: Digital Montessori-inspired Manipulatives. *Proceedings of CHI 2005*.