Children's Storytelling and Programming with Robotic Characters

Kimiko Ryokai, Michael Jongseon Lee, and Jonathan Micah Breitbart
School of Information
University of California, Berkeley
Berkeley, CA 94720 USA
{kimiko, michael_lee, breitbart}@ischool.berkeley.edu

ABSTRACT

We introduce mixed physical and digital authoring environments for children, which invite them to create stories with enriched drawings that are programmed to control robotic characters. These characters respond to the children's drawings as well as to their touch. Children create their stories by drawing props and programming how the robotic character should respond to those props and to physical touch. By drawing, programming the robotic character's behaviors, and organizing and negotiating the order and meanings of the props, children's story events unfold in creative ways. We present our iterative design process of developing and evaluating our prototypes with children. We discuss the role technology can play in supporting children's everyday creative storytelling.

Author Keywords
Storytelling, children, tangible, programming, robotic toys

ACM Classification Keywords
H.5.2 User Interfaces (D.2.2, H.1.2, I.3.6); Interaction styles (e.g., commands, menus, forms, direct manipulation

General Terms
Design

INTRODUCTION

Storytelling plays an important role in the development of children's creativity and cognition. When children tell stories, they not only practice representing events in narrative forms that can be shared with others [2], but they also explore possible outcomes of different events and negotiate their meanings with others [8]. Storytelling also offers opportunities for children to practice symbolic manipulations, hold multiple abstract concepts in their heads, and create meaning between these ideas [16].

Tangible computational tools have the potential to make the symbolic and abstract manipulations involved in storytelling more concrete and manageable for young children [4]. For example, a toy that allows children to record parts of a story in oral language, can later invite children to physically organize these recorded pieces, thus allowing them to engage in an editing process without relying on handwriting or keyboards - tools which they are not yet fluent with [1, 25]. As young children's storytelling often occurs during interactions with the physical world and other people, it is also important to consider supporting multimodality. The use of tangible props, such as stuffed animals and children's own drawings, makes it easier for children to play and build their ideas [6].

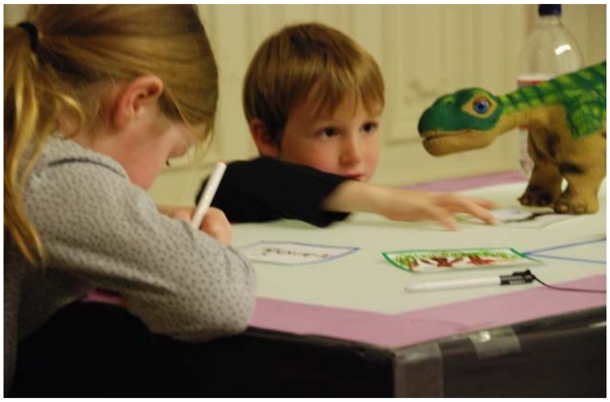


Figure 1. Children drawing on cards and programming how the robotic character should react to the prop.

It is this intersection of concrete and abstract manipulations involved in storytelling that we are interested in facilitating with this research. We introduce a mixed physical and digital authoring environment for children to tell stories with interactive robotic characters. With our system, children tell stories by creating a variety of drawn props the robotic character can recognize and respond to. Children do this by drawing on a set of colored cards and programming what they want the robotic character to say and do when the character sees different cards. On our interactive tabletop, the robotic character comes to life with the behaviors and recordings the children programmed into each of the colored cards. Through the design and evaluation of our system, we explore the role of technology in supporting children's creative multimodal story creation.

C&C'09, October 26–30, 2009, Berkeley, California, USA.
Copyright 2009 ACM 978-1-60558-403-4/09/10...$10.00.

BACKGROUND

Supporting Multiple Styles of Performance

The styles in which children portray their narrative thought may be different depending on their personalities. Howard Gardner and his colleagues found individual differences in children's storytelling, however, some patterns did emerge in their analysis of children's performances. Some children were verbalizers who were less interested in drawing, but produced copious amounts of language. Others were committed visualizers who plunged directly into drawing or building and offered linguistic comments sparingly [9, 10]. It is important not to favor or exclude one medium over another in supporting children's creative development and opportunity to represent their ideas.

With this system, our goal is to support such heterogeneity in children's performance styles by inviting children to create their own stories through drawing, recording voices and sounds, physically touching a robotic character, and programming with a graphical interface - without favoring one mode over another.

Bricolage and "Soft" Programming

We build our work on a constructionist's approach to learning by supporting children's creation of and engagement with personally meaningful projects [15]. Since LOGO programming, many generations of computational tools and environments that can engage children in the construction of personal projects through programming have been developed [2, 13, 24] (see a more detailed discussion in the related work section). Furthermore, in "Epistemological Pluralism," Sherry Turkle and Seymour Papert [26] argued for pluralism in programming by making reference to bricoleurs (with original discussion by Levi-Strauss in the 1960's as a "science of concrete" in contrast to the analytic methodology of Western science). While programmers are traditionally taught to engage in a top-down, divide-and-conquer way, Turkle and Papert argued that there may be a more "soft" approach to programming that resembles bricoleurs who construct meaning by arranging and rearranging and negotiating and renegotiating with a set of familiar materials. Bricolage is a way for a person to organize work by playing with individual elements and moving them around as they would in a collage. In contrast to having a plan, dividing the task, and using modules and sub-procedures to conquer the problem, bricoleurs make meanings through negotiation and association. In the design of programming environments for young children, providing a flexible programming environment that celebrates arranging and rearranging concrete materials may be beneficial.

To this end, our system offers a flexible and familiar programming space where children are free to draw and play at their own pace. They can incrementally add complexity to them by programming what the robotic characters will do when they see the cards. As such, we offer a flexible programming environment that celebrates a pluralistic approach to programming.

Child-Driven Play Space

Robotic toys and animatronics are gaining popularity and becoming more readily available on a consumer level (e.g., by WowWee, Sony, Hasbro, Omron). Ugobe's Pleo [17] is one such consumer level robotic dinosaur toy targeted for children aged 8 and up. Pleo is designed to be a friendly and curious dinosaur that exhibits a variety of life-like behaviors. Out of the box, Pleo responds to touch and gives an impression of learning by reacting to an individual owner in unique ways. However, in reality, Pleo is not equipped with any learning mechanism, but simply runs complex combinations of programmed responses that emulate intelligent behavior.

Our informal observations of children (age 5-10) playing with Pleo showed that they enjoyed cuddling with Pleo, and like with a real pet, wanted to teach Pleo special tricks. In response to this observation, we wanted to provide children with the opportunity to actively create and control Pleo's behavior.

Pleo is an open source platform, allowing technically capable hobbyists to customize and program their own original behaviors beyond the preprogrammed actions (e.g., singing original songs or performing customized dances). However, in order to produce such customized expressions, one needs a relatively high level of technical competency (e.g., knowledge of the C programming language and PAWN scripting). Our goal is to create an environment that allows children to easily program and control Pleo's behaviors.

RELATED WORK

Storytelling Tools for Children

Technology serving as a facilitator for children's design activities has been successful in learning domains beyond math and science. For example, KidPad [7], developed at the University of Maryland, is a drawing program that supports the rich storytelling associated with children's drawings. Zoom-in and zoom-out tools in KidPad allow children to embed and hyperlink their drawings in order to build a complex visual narrative. KidPad offers a whole new lens for children to build and share their visual art. Jabberstamp [22] is a tool for children to embed audio recordings into drawings created on paper. This tool allows children to compose and arrange their recordings on a graphical canvas, but does not provide tools for children to manipulate their recordings. StoryMat [25] recorded children's oral stories and the movements of stuffed animals made on a technologically augmented play mat. When another child played with the mat, the stories were played back as animations – echoes of a previous playmate. Results demonstrated that interacting with peer stories on StoryMat led children to tell more imaginative and structurally advanced stories. Our system combines advances of these prior works.

Programming Environment for Children

The constructionist approach has been applied to a variety of virtual programming environments for children. MOOSE Crossing [2] invited children to construct a virtual environment in which they could interact with each other.

While a fun environment for children to program virtual objects and characters, MOOSE Crossing also served as a forum for children to practice their narrative writing skills. Scratch [13] is a graphical programming environment that allows children to create animated stories and games by snapping together graphical building blocks, each representing a different command or action.

A variety of programming environments for robotic creatures have also inspired our work. Crickets [24] and the commercially available LEGO Mindstorms [12] are systems of physical LEGO blocks, sensors, and actuators, plus a graphical programming environment, that allow children to build their own programmable robotic creations. They invite creators to move between the physical world of model creation with blocks and the virtual world of programming. Topobo [21] is a construction kit with kinetic memory that invites young children to build spatial creatures and program their movements by directly twisting and turning the physical model. Guo and Sharlin presented a system that allows a person to control a robotic character, Aibo, via Nintendo Wii game controllers [11]. Constructionist learning happens when children are actively producing or manipulating personally meaningful artifacts. By combining the physical and the virtual, our approach is to allow children to decide where to focus their actions and easily move between physical interaction and virtual control.

DESIGN PROCESS

We used an iterative design process of studying children's storytelling play with robotic characters, designing and implementing our prototypes with children, and modifying our design.

Combining the Physical and Virtual

During our informal play sessions with an out-of-the-box Pleo, children noticed that Pleo was doing something in response to their physical touch. However, the association between where the child touched the Pleo and the action Pleo performed was not very clear. Therefore, we wanted to give children real-time access to what goes on in the robotic dinosaur's head so that they could better understand the process and possible behaviors they are able to change or control.

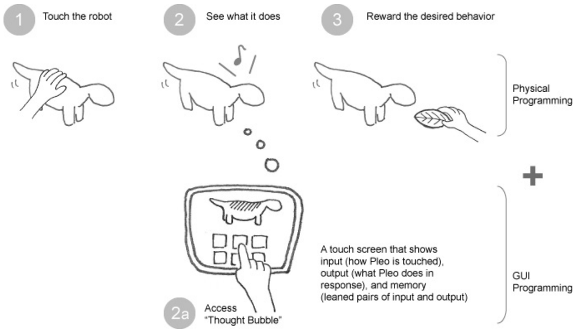

1
Touch the robot
2
See what it does
3
Reward the desired behavior
Physical Programming
+
A touch screen that shows input (how Pleo is touched), output (what Pleo does in response), and memory (leaned pairs of input and output)
2a
Access "Thought Bubble"
GUI Programming

Figure 2. Combining Physical and GUI programming using a "thought bubble" of the robotic character as a metaphor.

Illuminating the Process: "Thought Bubble"

The metaphor we built on is a "thought bubble" of the robotic character, through which children tap into the thought process of the character. For our first prototype, we used a touch screen display in combination with Pleo to show what is happening to Pleo in terms of input (how Pleo is touched), output (what Pleo does in response), and memory (learned pairs of input and output), in real time.

Initial Prototype

Touching different body parts of the robotic Pleo immediately highlights the chosen body part on the screen and shows available Pleo actions tied to that body part (see figure 3). At the same time, Pleo starts to physically perform the possible actions in sequence. . These actions performed by Pleo are also highlighted on the "thought bubble" screen. The desired behavior may be positively reinforced by either feeding the robotic Pleo a provided physical leaf object, or by pressing the virtual leaf icon on the interface. For example, if Pleo's singing behavior while being scratched on its chin is rewarded with the leaf, Pleo associates the chin scratch with the singing behavior. The next time Pleo's chin is touched, Pleo associates that input as a cue to start singing.

"Learned" behaviors are saved in Pleo's "memory bank" (see figure 3). This memory bank serves as a repository showing what tricks Pleo has been taught. The items in the "me" (input) column can be removed by pressing the trashcan icon. Once the input part of the pair is removed, the action is automatically played directly after the preceding pair. For example, figure 3 shows that the topmost pair in the queue would initiate by touching the tail. If the tail is touched, Pleo will start to sing. Immediately after singing, Pleo will wiggle his right leg. After wiggling its right leg, Pleo will wait for the user to touch its back to execute the next pairing.. As such, the memory bank allows basic conditional (procedural) as well as sequential programming.

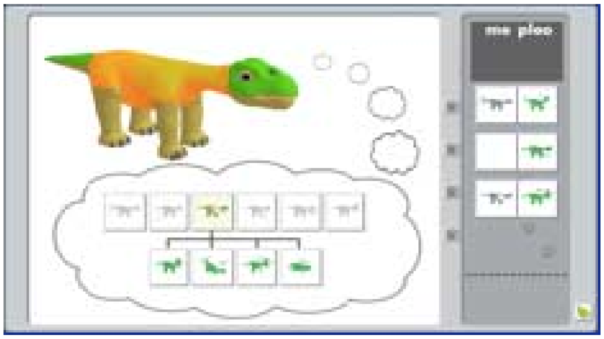


Figure 3. Combining physical and GUI programming using a "thought bubble" of the robotic character as a metaphor. The children understood the idea of the "thought bubble" but did not use it to program a sequence of behaviors.

We wanted to allow multiple entry points to interaction with the robotic toy by providing both physical and virtual interfaces. The child may choose to program 1) with the physical toy only, 2) with the screen interface only, or 3) using a combination of the physical interface and the GUI.

Results from Our Initial Implementation and Reiteration

Nine children (age 5-8) played with our first prototype system. The children understood the relationship between the "thought bubble" screen and Pleo's action with respect to which part was being touched. They also understood that touching appropriate parts of the screen could actively control Pleo's behaviors. The opportunity to teach Pleo behaviors had the children very engaged throughout the process. The children also remembered how to access certain types of behaviors, for example, touching the head to access and activate Pleo's "Moo" sound. The children eagerly showed each other the different tricks Pleo could perform, exclaiming "Look what he can do!" as they touched the thought bubble screen to navigate and activate desired behaviors. Many pairs started by focusing on physically touching Pleo and gradually moved on to interacting with the GUI once they had a better understanding of the system. None of the children completely ignored the GUI screen to focus solely on physical play with Pleo. The children did seem to understand the right hand region of the screen to be the "memory bank." When asked by the investigator to explain what they thought the right hand region represented, children answered, "That's what Pleo knows." However, the children did not use the editing function of the "memory bank" to create sequences of behaviors. They seemed to interpret the "memory bank" as a log or history, and not something to be acted upon. Therefore, no procedural programming was observed. As a result, we concluded that the function of the memory bank needed to be made more explicit and easier for the children to utilize.

Pleo can play back various non-linguistic audio cues, e.g., "grr" "coo" "purr" etc. However, the children wanted to have Pleo say something original. This led us to develop an intuitive way for children to record their own sounds and incorporate them in the programming sequence and this functionality is included in the second iteration of our prototype system.

OUR CURRENT SYSTEM

Our current system builds on our initial prototype and the thought bubble metaphor. Based on the evaluation of the first system, three major improvements were made: 1) Cleaner programming environment that uses a dialog metaphor, 2) Addition of sound recording functionality, and 3) Incorporation of children's own drawings and storytelling. The software is designed to encourage children to tell stories by having Pleo interact with pictures they have drawn. Interaction between Pleo and the children is also encouraged through a turn-based input and performance system. In the following sections, we describe the system and the interaction design.

The Interaction

Children first draw pictures on one of twelve provided 5x4 inch paper cards. After that, the children are shown how to interface the cards with a tablet PC, which displays the graphical programming interface. When a card is placed adjacent to the tablet PC, the screen lights up and the children are prompted by a cartoon representation of Pleo asking what they want Pleo to do. Children can choose to either pick one of the twenty-four behaviors Pleo can perform or record an audio clip that will be played back. Once this is done, the cartoon representation of Pleo pops up asking what "you" (the user) will do (see figure 4). Here, the children can choose to pick a region where they would like Pleo to be touched before he performs his next behavior. However, the children can decide to skip this step

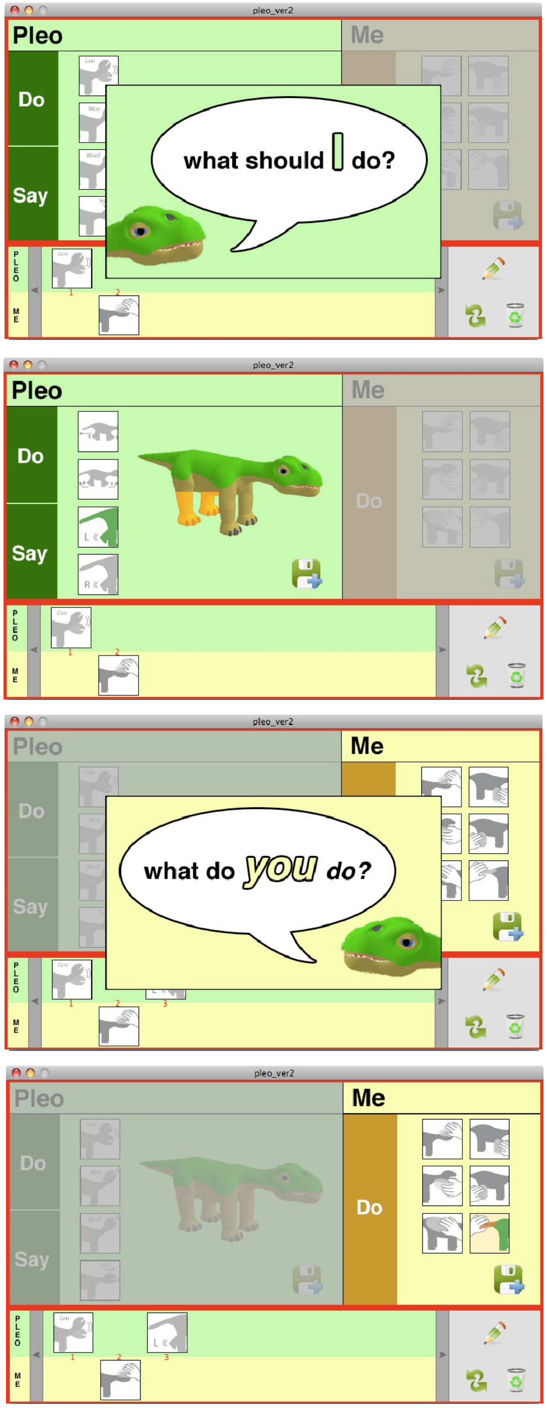


Figure 4. The system walks through the interaction programming in the form of a dialogue.

and choose either to select another behavior or record an audio clip for Pleo to play back. A toolpane is provided so children can modify (edit, swap, or delete) saved commands and recordings. Children can choose to continue adding Pleo and "me" behaviors, or start the process again by drawing on a new card and interfacing it with the tablet PC.

Once the children are satisfied with their cards and corresponding programmed behaviors, they can place a card in front of Pleo. When Pleo recognizes a card, Pleo will follow the recorded behaviors step-by-step. In the case where there is a "you" command, Pleo will wait for the appropriate physical input before continuing on to the next command. While Pleo is looking at a card, the tablet PC also shows Pleo's "thought process" and highlights the current command being executed or the physical input Pleo is waiting for. Once Pleo completes the tasks for a particular card, the GUI goes blank and Pleo remains stationary until shown another card. In other words, Pleo's behavior is driven by the children's instructions.

The Interface

The GUI is split into two major regions. The top half of the screen contains commands children can select and the bottom half of the screen shows the list of chosen behaviors (see the second and forth images of figure 4). The top half, or "command section," is broken down further into two regions with the left side showing Pleo behaviors, and the right side showing "me" behaviors. Using the Pleo command side, users can instruct Pleo to "do" a command by clicking on a region of Pleo's body and clicking on the desired behavior. Additionally, users can tell Pleo to "say" something by clicking on the record button, saying something into the microphone, and saving the recording. Using the "me" command side, users can tell Pleo to wait for a physical input from the user by choosing the corresponding body part icon.

The bottom half, or "queue section," is also broken down into two regions. The main part of the bottom region consists of the list of behaviors that children have saved. A smaller region is dedicated for a "toolpane" consisting of an edit button, a swap button, and a delete button. Users can use these tools to modify the behaviors in the queue sequence.

Physical Setup

The play space we designed consists of a 46.5 x 30.75 x 19.25 inch (118 x 78 x 49 cm) table with a glass top (see figure 5). For the graphical programming interface, we use a 12-inch tablet PC that lays flat on the table. Children use the tablet pen to interact with the programming interface. The tablet PC is positioned at the edge of the table so the children can easily interact with the interface. Pleo sits on the tabletop and is connected to the computer via a USB cable. For the audio recording portion of the system, we attached an external microphone to the tablet PC and placed it next to the computer. For audio playback, we placed two external speakers on the floor to the right and left of the table.

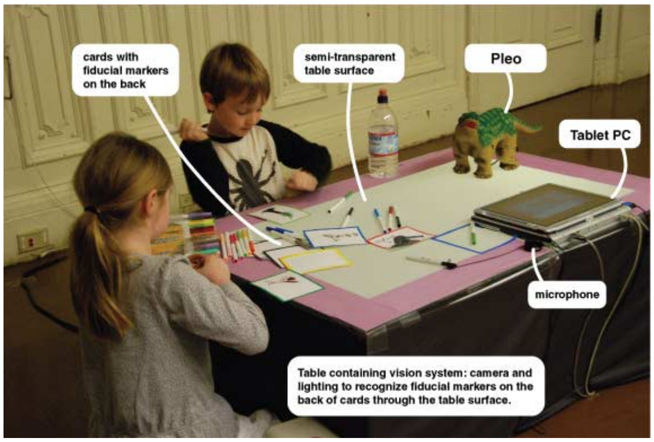

cards with fiducial markers on the back
semi-transparent table surface
Pleo
Tablet PC
microphone
Table containing vision system: camera and lighting to recognize fiducial markers on the back of cards through the table surface.

Figure 5. System setup.

Technical Implementation

The system was developed using the Processing [20] programming language. Vision and fiducial tracking is implemented using the reacTIVision tangible user interface library (TUIO) [23]. Communications with Pleo are achieved by sending serial commands over the USB port using built-in JAVA libraries.

Edit Mode

To allow the system to distinguish between cards, each one has a uniquely colored border on top, and a unique fiducial marker printed on the bottom. When a card is placed adjacent to the tablet PC, the vision tracking software identifies the card using the fiducial marker. The tablet PC also has a fiducial marker that tracks where it is on the play space. In this iteration, the tablet PC is stationary, so its position is constant, but we originally envisioned a portable touch screen monitor that could be moved around. The touch screen was planned to activate when placed in proximity to a card using the vision software.

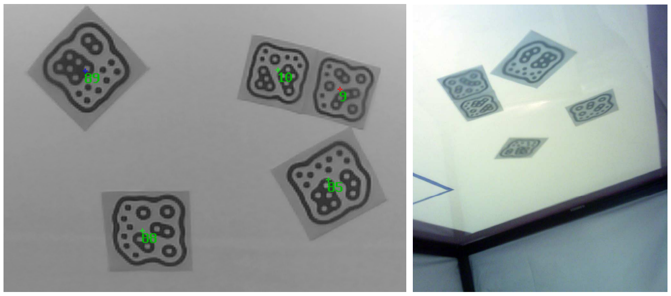


Figure 6. Fiducial markers seen from under the table.

The Behaviors

For this prototype, Pleo is able to perform four distinct behaviors for each of his six stimulus points (tail, body, back legs, front legs, top of head, and chin) for a total of twenty-four possible behaviors. Behaviors include a combination of movement and sounds, and were chosen to be loosely associated with the stimulus point, memorable, and brief (between four and ten seconds long). Behaviors were reviewed and identified using a combination of software applications: Dino-MITE [5] and MySkit [14].

Dino-MITE is a third-party tool that interfaces with Pleo's USB port and is used to debug, monitor, and send commands to Pleo. We used it to obtain a list of Pleo's built-in behaviors and corresponding commands. MySkit is a third-party application that allows users to create "skits" or additional behaviors using a three-dimensional model of Pleo. The twenty-four behaviors we chose for Pleo included built-in behaviors identified through Dino-MITE, example behaviors from the MySkit library, and custom designed behaviors using MySkit. The additional behaviors from MySkit were loaded into Pleo using an SD card. Commands are sent to Pleo by our software via a serial USB connection.

Playback Mode

The vision software recognizes and starts playback of the cards when in proximity to Pleo. Communication with Pleo is established using a serial USB connection.

EVALUATION

Participants

Eleven children (5 girls and 6 boys) between the ages of 5 and 9 participated in our study. Ten children played with our system in pairs. One child played with the system by himself as his partner was absent.

Methodology

The children were invited to come in to the room where the system was set up. The parents were asked to wait outside during the children's play session. The investigator first briefly introduced Pleo to the children and invited them to draw pictures on one of twelve provided cards. After the children drew their pictures, they were shown how to interface the cards with the tablet PC. Once each child had gone through the first practice run with the experimenter (approximately 5 minutes), the children were free to play on their own as long as they wanted to, for up to 60 minutes. During their free play, the children were encouraged to talk out loud and share what was happening. At the end of the session, the children and the experimenter invited the parents to come into the room and watch what the children had created.

At the time of the experiment, Pleo's serial USB connection was not designed to send joint and/or touch sensor states in real-time. Due to this limitation, we used a "wizard-of-oz" approach for the portions of our project that required recognizing when Pleo was being touched.

RESULTS

Each group played with our system during the entire duration of the session (from 45 to 60 minutes). The children created complex stories involving their own drawings, various behaviors they programmed into Pleo, reactions they provided to Pleo, and their narrating voices that connected these elements. Figure 8 is an example story by 8-year-olds, Mina and Heather. It illustrates the complexity of their creation in terms of what they programmed and recorded, and how they interacted with their creation as they narrated.

In the following sections, we discuss in detail the children's experiences in terms of their programming and storytelling activities.

Programming

The children found the GUI programming interface easy to work with. Once the experimenter had gone through an example sequence with the children, they were able to engage in programming on their own.

Building Interactions between the Child and Pleo

In contrast to our first prototype system where children created a chain of behaviors Pleo could perform by itself, our current system encouraged children to explore and program "interactions" between Pleo and themselves. In other words, the children understood that they were programming both Pleo's "output" behaviors (e.g., wagging its tail, mooing, etc.) and "input" that Pleo was expecting (e.g., touch its head, body, etc.). The dialogue prompts shown during each step of the programming phase ("What should I do?" and "What do you do?") seemed to work well as scaffolds. The children also understood that they did not have to follow this suggested turn-taking between Pleo and themselves, as they created both Pleo-only sequences (e.g., Pleo sniffs, walks, moos, and then waits for the child to pet its back) and Pleo-and-Me sequences (e.g., Pleo sniffs, the child pets his head, Pleo walks, the child pets his back, etc).

During playback mode, the children paid attention to the behavior sequences they programmed, and responded to Pleo accordingly when their input was expected. In playback mode, the screen not only displays the sequence of behaviors programmed for the given card, but also highlights the current step in the behavior queue [see figure 9]. This made it clear to the children where they were in the story and what was supposed to happen next. For example, most children were able to anticipate what Pleo would do next by monitoring where they were in the queue (e.g., a child may narrate, "and then he sniffs" before Pleo actually performs sniffing behavior). If Pleo was waiting for an input and a child forgot to touch Pleo accordingly, the partner child would remind him/her that Pleo was waiting

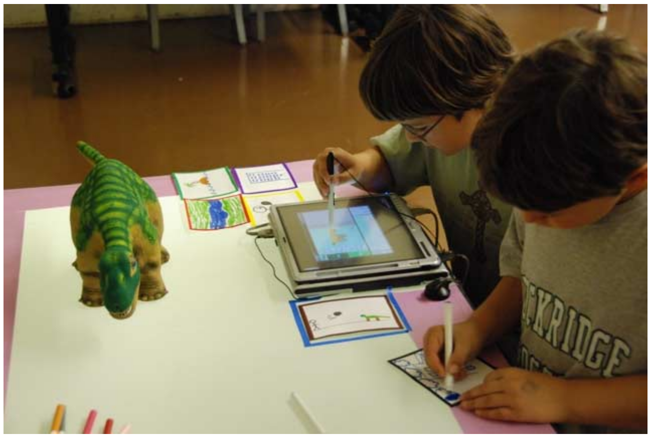


Figure 7. Children drawing and programming their cards.

to be touched so that it could move on to the following behaviors.

Editing and Rebuilding

Once the children got the idea of how to work with the system, their subsequent drawings seemed to be influenced by what they learned Pleo could do. For example, after seeing Pleo's ability to make the chomping sound, one child drew a leaf on his new card and programmed Pleo to make chomping sounds when he saw the card.

The children moved fluidly between drawing on the cards, programming the cards, playing back the completed cards, adding new cards, and modifying the existing cards with

Heather: Do you want to do it this way? This one this one?

Mina: that one and then that one.

Heather: So first, Pleo goes into the park. Then he wants some apples. So he eats some apples. Then he decides to chase a bird, and I come in and pet him.

Mina: And then he goes to the garden. And he sees flowers.

[Pleo sniffs]

He sniffs them, then I pet him on his head.

[Pleo purrs]

He is purring and then

Heather: He sniffs.

[Pleo sniffs]

Mina: He sniffs.

Heather: And then, for this one [holding the card], he sees fireworks in the air. He is wondering what they are. And he thinks it's very amazing. And then he sees people and dogs he had never seen them.

[Heather shows Pleo the card]

[Audio playback] "What are these? They are popping in the air! Oh look, there are people and dogs everywhere!"

Heather: And then I pet him on his back.

[this triggers Pleo to move forward and look around]

And then he walks over to check things out

[Pleo makes happy sound]

And then I pet his tail [to see] and then he says [playback starts]

[Audio playback] "Oh, look! Hi Girls! How are you doing?"

Heather: And then he says "Hi" to the girls.

[Mina puts down the car picture]

Mina: And then he sees the car.

[Pleo starts to shake and grr]

Mina: and then he starts to

Heather: starts to growl

Mina: growl.

[Heather looks at the monitor]

Heather: [to Mina] he sniffs

Mina; And then he sniffs

[Pleo sniffs]

Mina: At the car. And then I pet him the bottom of his head [pets his chin]

Mina: and then he wags his tail

[pleo wags his tail]

Mina: and then I pet him on his legs and he wags his tail.

[pleo wags his tail]

Heather: and that's our story!

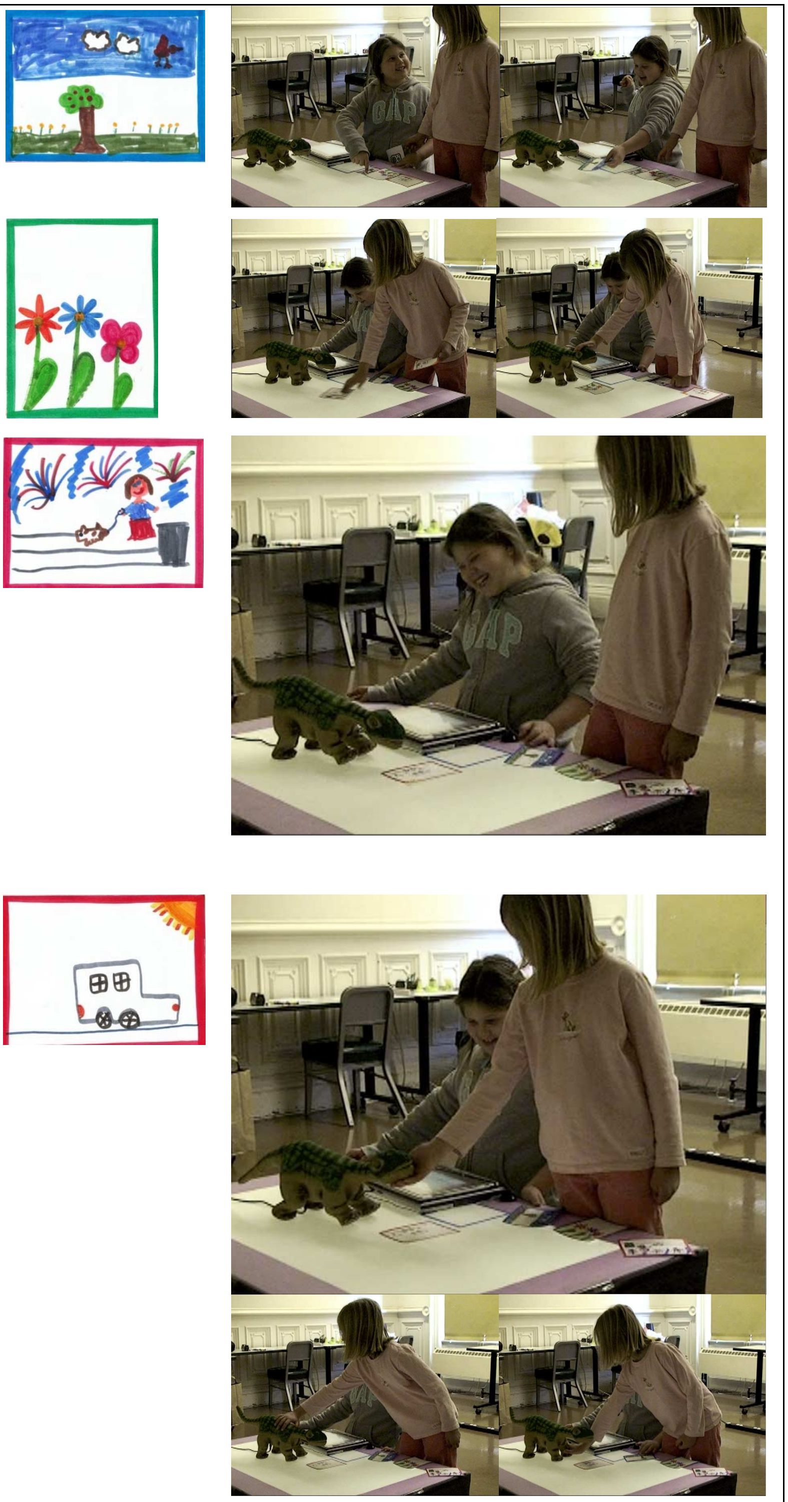


Figure 8. Mina and Heather's story.

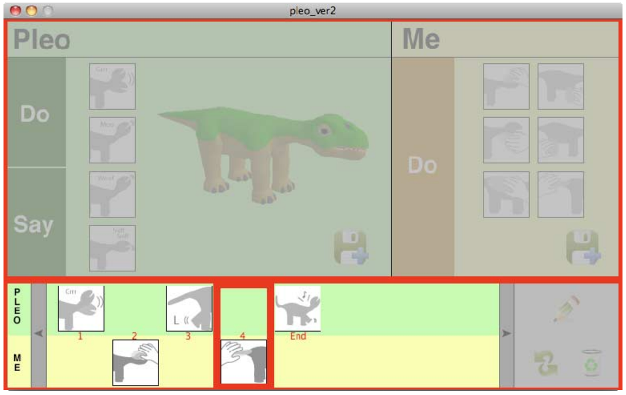
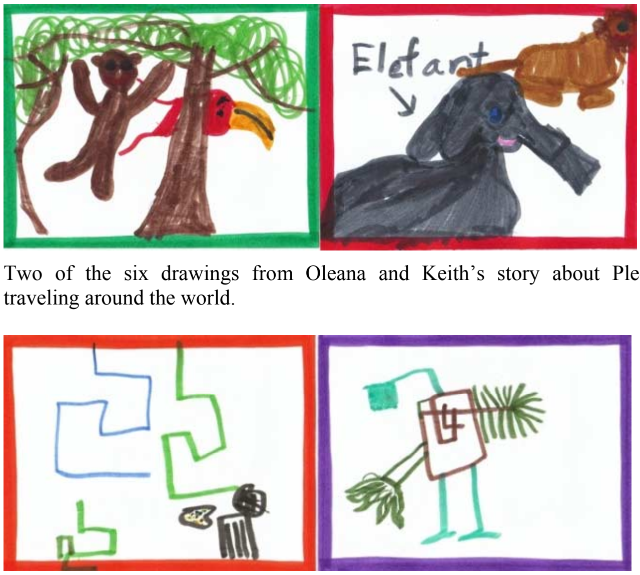


Figure 9. Shows sequence and highlight where they are in the queue.

new behaviors. Having the physical cards with their drawing as representations of programmed interactions between Pleo and themselves seemed to make the manipulation easier. For example, one child, Nick, was telling a portion of his story where Pleo was walking into a building, but noticed that in reality, Pleo was physically walking away from his card. Nick took the card and placed it next to the tablet PC to edit the sequence. Nick, then, not only made Pleo walk towards the building, but added a few additional behaviors so that the story flowed as follows: "Pleo walks up the stairs, I pet his leg because he was getting tired. And then Pleo walks out of the building." The children used the "thought bubble" interface as the window to tap into Pleo's mind. Compared to our earlier prototype, our current system was significantly more successful at inviting children to construct, reflect, and produce creative narratives.

Storytelling



Two of the six drawings from Oleana and Keith's story about Pleo traveling around the world.

Two of the nine drawings Jake made. Stories about Pleo meeting an alien with a garbage can (left) and a cleaning machine (right).

Figure 10. Example drawings from the children's stories.

Our current system was also more successful at engaging children in storytelling. The children's stories varied from Pleo traveling around the world with drawings of various places such as Africa, Asia, and the Eiffel Tower, to Pleo taking a walk from his home, meeting aliens, and even going to the bathroom. Drawings could be a part of a larger story or individual, unrelated events.

Improvisation

Although Pleo could produce life-like movements, for the children's storytelling, Pleo's life-like movements needed not be truthfully matched with what they were describing in their stories. For example, Pleo's walking movements could be Pleo "swimming" or Pleo "chasing the bird/". Pleo shaking its head could be narrated as "Pleo was looking around" or "Pleo was happy." The transcript below is a story told by Nick and Tom. Nick and Tom creatively constructed their narrative by connecting different elements of their stories and Pleo's actions with narrative phrases, even when they thought Pleo's behaviors didn't fit (e.g., "… no idea why he mooed," "He was just hungry so he mooed.")

Tom: Can we do it from the top? [pointing at their 6 cards laid out on the table]

Nick: Sure

Nick: [takes out the first card] When he was sleeping, I sort of imagined him laying down because he was trying to get to the ocean and he is walking along the river. And then, he was walking forwards, walking down the river

Tom: And then he was walking, he was walking across to get to the ocean

Nick: And I have no idea why he mooed.

Tom: Yeah [laughs] He was just hungry, so he just mooed.

Nick: Okay. So now,

[audio playback] Let's go swimming [Nick's voice]

Nick: He finally got to the ocean, when he was walking [Pleo walks] he was swimming, he fell asleep on the island

[audio playback] "Wow, it's a guy with the ball!" [Nick's voice]

[Pleo makes cooing sounds, shakes his head]

[audio playback] "After I see your house, we have to play catch" [Tom's voice]

Tom: Well, that was, he was so excited, he was doing all these amazing things, and then he was walking at the very end. That was when he was going into his house.

[Pleo moos]

[audio playback] "Look it's a blue building!" [Nick's voice]

Nick: he was walking in and then he

Tom: Well, he was walking up.

Nick: Yeah, and then he was

I think he was playing with the person inside. And then he was walking outside, so that he can play catch with the person.

So the catch [reaching for the card and showing it to Pleo] [audio playback] "Yay, I like playing catch with the person with the ball"

[Pleo makes the happy sound.]

[audio playback] "Look out, it's a thunderstorm!" [Tom's voice]

Tom: Well the guy was, he was, um, I was petting him on the back, starting to tell him there is a thunderstorm, and then he fell asleep, so I was like: "There's a thunderstorm, there's a thunderstorm!" And then he woke up again and then I pet his tale and he woke up again and then he fell asleep again. The end.

Creatively recording their voices

One of the elements that made the children's stories richer was the voice recordings they made for Pleo to say during story playback. During the programming stage, the children carefully distinguished whether or not they wanted Pleo to "do something" (i.e., actions such as walking, tail waging, etc.) or "say something." When they wanted Pleo to "say something," the children planned and recorded their own speech using the recording interface by themselves.

The number of recordings for each card varied between none and up to two recordings (average 0.7 recordings per card). Most children made voice recordings from Pleo's perspective (e.g., "Hi girls!" "Look out, it's a thunderstorm!" as seen in the example transcripts). A few children however, used the "Pleo Says" button to record sound effects. Jasper, for example, drew a leaf on a new card, and then recorded the opening and closing of the Velcro on his pants pocket. With the card, Jasper told a story about Pleo finding a leaf and eating his leaf, with the Velcro sound effect in the background mimicking Pleo's chomping sound. Even though the system prompted what Pleo should say, the children also used the tool to come up with creative sound effects.

Creative use of cards

In the study sessions, each child drew and programmed between 2 and 9 cards (average 3.8). Typically, each card consisted of 4-10 programmed behaviors. Some children expanded their story by adding new cards. Other children expanded by either modifying or adding new behaviors to existing cards. In either case, having the physical cards with their own drawings seemed to make it easier for the children to think about the flow and the sequence of their stories.

Using the cards, the children also talked about and negotiated the sequence of events. The following is what Mina and Heather responded when they were asked by the experimenter to explain how their story went:

Mina: He (Pleo) is in the park and then he goes to a garden. And then he starts to walk and sees the girl and the dog, and the fireworks, so then he sees the car. Cause he is on the sidewalk with them. So then he sees the car. So it kind of **fits in**.

Experimenter: Fits in?

Heather: Yeah

Mina: Yeah, like from the park to the garden, and then sidewalk. Since he is on the sidewalk, he can see the car.

Heather: So Yeah, it's like, it's **a good flow** with the orders.

The children engaged in meta-narrative discussion about the flow, order, and connection of their cards and stories. At the end of the study, the children's parents were invited into the room and the children were eager to show their parents what they created.

LIMITATIONS & FUTURE WORK

While the system ran smoothly without any crashes during the study, there were some technical limitations that need to be addressed for a future system. First, the tracking of the fiducial markers needs to be more robust. The current study required assistance from the "wizard" to fill in the detection of the fiducial markers when they were near Pleo, as real time tracking of Pleo's position and orientation was difficult. When the children played with the cards away from the table surface, or when the cards accidentally overlapped with other cards, the vision system became confused. We plan to solve these issues by incorporating a more robust video camera, lighting, and diffuser to be installed inside the table, or perhaps implementing RFID tracking of system components.

The 60 minutes passed quite quickly for each session. We plan to conduct a longitudinal study to see how children's creative processes evolve over time (e.g., over several days) as they become more fluent with the tool. As we prepare for such long-term interactions, we will need to develop intuitive ways to store and give children access to their stories that evolve over multiple play sessions. For instance, having a storybook format where each child could access a collection of stories in progress may be helpful.

The current system has one robotic character. We will also consider how having multiple robotic characters and thus multiple perspectives may influence the design of our system.

CONCLUSION

Using our system, the children drew pictures, planned their stories, programmed a character's behaviors, ran their programs, and edited their programs. Through these activities, the children created complex stories that involved interaction between their own drawings, a physical robotic character, sound recordings, and responses to the children's physical input. Our system supported children's verbal, visual, and kinetic expressions, through which the children found ways to creatively develop their personal stories.

Like bricoleurs who construct meaning by arranging and rearranging and by negotiating and renegotiating with a set of familiar materials, the children played with their physical drawings as representations of programmed interactions between Pleo and themselves. By physically moving the cards (with behaviors encoded in them) like a collage, they manipulated complex elements in their stories.

Most of all, our system gave children the control to drive their own interactive characters and imaginative stories. As they constructed, listened to, and manipulated their own story elements, they advanced their stories in creative ways. With this ongoing work, we contribute to the design of multimodal tools for children's creative storytelling creation.

ACKNOWLEDGMENTS

We thank the children and their parents, Deirdre Mulligan, Caleb Chung, John Sosoka, and Barbara Barza from UGOBE, and Matt Bauer for his help with Dino-MITE.

REFERENCES

1. Ananny, M., & Cassell, J. (2001, April). Telling tales: A new toy for encouraging written literacy through oral storytelling. Society for Research in Child Development, Minneapolis, MN.
2. Bruckman, A. (1997). MOOSE Crossing: Construction, Community and Learning in a Networked Virtual World for Kids. PhD Thesis. Massachusetts Institute of Technology, Cambridge, MA; 1997.
3. Bruner, J. (1986). Actual minds, possible worlds. Cambridge, Mass. : Harvard University Press, 1986.
4. Cassell, J (2004) "Towards a Model of Technology and Literacy Development: Story Listening Systems" Journal of Applied Developmental Psychology 25 (1): 75-105.
5. Dino-MITE www.bauerindependents.com/SUBMAIN/dinomite.htm
6. Druin, A., Montemayor, J., Hendler, J., McAlister, B., Boltman, A., Fiterman, E., Plaisant, A., Kruskal, A., Olsen, H., Revett, I., Schwenn, T. P., Sumida, L., and Wagner, R. 1999. Designing PETS: a personal electronic teller of stories. In Proceedings of CHI '99. ACM.
7. Druin, A., Stewart, J., Proft, D., Bederson, B., Hollan, J. (1997). KidPad: a design collaboration between children, technologists, and educators. Proceedings of CHI'97, ACM Press, (1997).
8. Dyson, A.H. (1993). A sociocultural perspective on symbolic development in primary grade classrooms. In Daiute (Ed.) The development of literacy through social interaction. San Francisco: Jossey-Bass Publishers.
9. Gardner, H. (1982). Art, mind, and brain: a cognitive approach to creativity. New York : Basic Books, c1982.
10. Gardner, H. (1983). Frames of mind: The theory of multiple intelligences. New York: Basic Books.
11. Guo, C. and Sharlin, E. (2008) Exploring the use of tangible user interfaces for human-robot interaction: a comparative study. In Proceeding of CHI '08. ACM Press.
12. LEGO MindStorms http://www.mindstorms.lego.com/
13. Maloney, J., Burd, L., Kafai, Y., Rusk, N., Silverman, B. and Resnick, M. (2004). Scratch: A Sneak Preview. International Conference on Creating, Connecting, and Collaborating through Computing, Kyoto, Japan.
14. MySkit http://www.dogsbodynet.com/myskit/index.html
15. Papert, S. (1980). Mindstorms. New York: Basic Books. 1980.
16. Pellegrini, A. D., & Galda, L. (1991). Longitudinal relations among preschoolers symbolic play, metalinguistic verbs, and emergent literacy. In J. Christie (Ed.), Play and early literacy development (pp. 47-67). Albany: State University of New York Press.
17. Pleo. http://www.pleoworld.com/
18. pygame game development library. http://www.pygame.org/news.html
19. pyserial http://pyserial.sourceforge.net/
20. Processing. http://www.processing.org/
21. Raffle, H., Parkes, A., Ishii, H. (2004) Topobo: A Constructive Assembly System with Kinetic Memory, in Proceedings of CHI '04. ACM Press.
22. Raffle, H., Vaucelle, C., Wang, R., Ishii, H. Jabberstamp: embedding sound and voice in traditional drawings. To appear in the Proceedings of Interaction Design and Children, 2007. Aalborg, Denmark. June 6-8, 2007. ACM Press.
23. ReacTIVision. http://reactivision.sourceforge.net/
24. Resnick, M., Martin, F., Sargent, R. and Silverman, B. (1996). Programmable Bricks: Toys to Think With. IBM Systems Journal, vol. 35, no. 3-4, pp. 443-452.
25. Ryokai, K. and Cassell, J., (1999) "Computer Support for Children's Collaborative Fantasy Play and Storytelling", In Proceedings of CSCL '99.
26. Turkle, S., & Papert, S. (1990). Epistemological Pluralism. Signs, vol. 16, no. 1.