[Tangible] Programming at the Computer History Museum

1st Author
1st author's affiliation
1st line of address
2nd line of address
Telephone number, incl. country
1st author's E-mail address

2nd Author
2nd author's affiliation
1st line of address
2nd line of address
Telephone number, incl. country
2nd E-mail

3rd Author
3rd author's affiliation
1st line of address
2nd line of address
Telephone number, incl. country
3rd E-mail

ABSTRACT

In this paper we explore the results of a pilot study of a touch table installation and the Computer History Museum in Mountain View, California designed to engage visitors in visual programming. Analysis includes attitudes towards computer programing of child visitors (ages 8-13) and naturalist observations of a more general visitor population. Based on prior work in museum settings, we also propose a tangible programming interface to replace the graphical interface. Discussion of the tangible implementation includes a discussion of design decisions and prototype considerations.

Author Keywords

Games for learning; interactive surfaces; collaborative learning, informal learning environments; programming; tangible user interface.

Keywords

Computer programming; Computer science education; Museums; Tangible user interface; Tangible programming; Exhibit design; Multi-touch tabletops; Computer supported collaborative learning

INTRODUCTION

Museums and other collection-based institutions have long worked to integrate interactive experiences alongside authentic artifacts that visitors are invited to look at but not touch. The Computer History Museum in Mountain View, California, has begun development on an exhibit called *Make Software, Change the World.* This exhibit is motivated by digital technologies becoming ubiquitous in daily life, and the desire to educate and inform visitors about computers, technology and their relation to programming. *Make Software, Change the World* will include a multi-touch tabletop experience to *"engage visitors in free-form computer programming activities"* [3].

Teaching programming to a diverse audience is more important now than ever. There has been call to integrate computer science education into K-12 systems on a national level, integrate it into STEM curriculum and make it a fundamental part of workforce development [8]. Inclusion within formal learning environments only supports the need for computer science education at informal learning institutions. Previous work by Margolis and Fisher has shown that that informal experiences with computers, in all settings at an early age, greatly impacts a persons propensity towards computer science and related subjects in high school and college [9]. It is imperative that computer literacy to become ubiquitous, as a large portion of future STEM jobs will revolve around computing and programming [8]. Similar to math and reading, literacy of computing concepts should be reinforced through a variety of facets. The reinforcement must not only focus on traditional education, to be sustainable we must also spark interests, establish it was a part of one's identity, and certainly make it enjoyable

In this paper we present a preliminary analysis of the Frog Pond, a tabletop exhibit designed help children learn about programming through a simple block language to control a frog avatar. This pilot study is part of a larger research project aimed at gauging the effectiveness of this exhibit in achieving active prolonged engagement (APE) [7] and creating a positive attitude shift towards programming. Initial analysis will focus on evaluating the effectiveness of our attitude surveys and natural observations. We will look

Figure 1. Frog Pond tangible user interface in action

at how the results may inform future design decisions and we also detail an alternative design, employing a tangible programming interface (TPI).

FROG POND OVERVIEW

The Frog Pond exhibit is a work in progress with Dr. Michael Horn and the Computer History Museum. Frog Pond is a free form sandbox like activity meant to be played on a multi-touch table top device. There are two stations on either side of the table where visitor(s) can engage in 'programming' a frog avatar. With a 'Scratch-like' interface, players can drag frog command blocks (e.g. hop, chirp, turn left, turn right, hatch), control blocks (e.g. if/else, repeat statements) into an active program manager on the screen. Up to twenty blocks can be used to construct a program, which players are able to play/pause/reset at any time.

There are four different types of flies (red, green, blue, yellow), which frogs can eat, and although there are no winning conditions there is a intrinsic goal of collecting all four or as many flies as possible [3].

Design Principles

The driving design principles for the Frog Pond exhibit are based on the fundamental components for active prolonged engagement (APE). APE was developed by Humphry and Gutwill [7] at the Exploratorium in San Francisco, California as a concept for creating and evaluating exhibits to encourage behavior to make experiences more meaningful and impactful. There are three fundamental components to APE approach:

- *"Support for open-ended exploration with gentle guidance"*
- *"Promotion of self-driven discovery by minimizing the instruction and explanation and by encouraging visitor-initiated observation, speculation, play and construction.*
- *"A shift of the visitor's role from that of recipient (of our instructions and explanations) to that of participant (with the museum and other visitors) in the generation of activities, questions, and explanations related to phenomena"* [7]

These design principles manifests themselves in the design

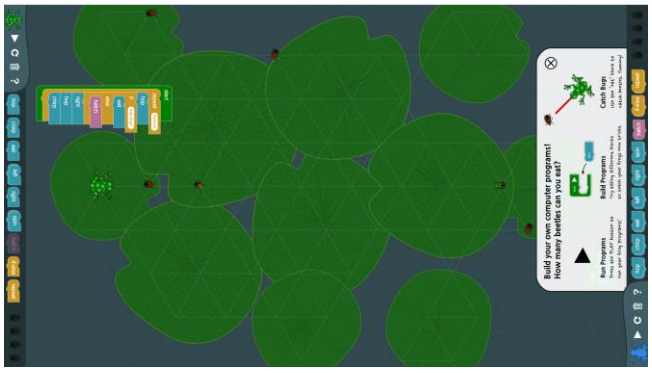

Figure 2. Frog Pond GUI screenshot

of the Frog Pond activity. Support of open-ended exploration embodies itself within the instructions of the game, providing a simple goal, "Catch Bugs". This does not inform the user of how to complete the task, nor what methods are right or wrong. The promotion of self-driven discovery is thoroughly leveraged in the creation of a program. When visitors start the game, they are presented with a program with a single command, 'hop'. They are left with a pile of additional commands and instructions on how they can be used. Creating longer, complex, and 'successful' programs is all left open to the user, whether leveraging trial and error, experimentation, or random testing. Finally the shift of the visitors' role is less defined in the game play but the form factor on which the game is delivered. Placing the game on a large, multi-touch table with two stations at either ends is meant to foster a community around the game. This form factor is meant to invite discussion amongst visitors, acquaintances and strangers, discussion about the goal, methods, mentoring, and other relatable topics.

RELATED WORK

Previous research has investigated many forms of activity around multi-touch tabletops and tangible programming interfaces, including a few focused in informal learning environments. We begin this section with a brief overview of design principles for informal learning environments that guide the ongoing work of this project.

Active Prolonged Engagement

As mentioned in the Frog Pond Overview, APE aims to create museum exhibits that foster more personal, meaningful experiences. It has been becoming popular goal for experimental exhibits within informal environments. The driving concepts of APE exhibits make them approachable to a wide audience of various age and knowledge. As stated by Humphry and Gutwill:

> *"The core of the APE exhibit development process was to shift the role of the visitor from that of recipient of instructions and explanations to that of participant. Our goal was to create exhibits where visitors participated, with the museum and with other visitors, in the generation of activities, questions, and explanations related to engaging phenomena."* [7]

From the users' perspective, Frog Pond aims to be an active experience, an open ended activity with minimal instruction set, giving users the freedom to determine their own goals within a sandbox format. The games style of sandbox interactions encourages prolonged interaction. Particular details such as not setting a time limit or time-based objective make for an experience that is only limited by the users own interest.

Tangible Programming Interfaces

The second aspect of this project focuses on developing an alternative method of programming the frog agent using a

tangible interface instead of the multi-touch GUI. Previous work has explored the use of tangible programming, particularly in the field of robotics. For the Frog Pond exhibit, visitors are looking to program a virtual frog using navigational and sensory/actuator commands (e.g. see-bug, eat). It stands to reason that controlling a virtual avatar should have similar parallels to controlling a robot, although this is an assumption we are interested in exploring through this project. The use of tangible programming interfaces (TPI) within the context of robotics has been a popular subject for some time. Research in the past has ranged from sensory and response relationships [17] to programming robot navigational behavior [5]. These studies have shown that TPI can be more enjoyable then GUI interfaces, offer more opportunity for collaboration and more likely to engage girls over GUI [5,16]. The aim of the Frog Pond TPI design is to leverage these benefits for better APE.

TPI has not been limited to the field of robotics and robotics education. TPI have been developed for programming story-telling [12,13] virtual navigation [18] and simple pseudo programming [2]. All these studies provide creative design solutions for teaching programming concepts outside of the realm of robotics and classroom material. Additionally both the Quetzal Language [1], the Digital Dream Lab [13] leverage puzzle piece style interfaces to support their physical representation of computer language. The benefit with the Frog Pond is that the GUI can provide baseline for a comparison with the TUI. The findings in this paper pilot evaluation of the Frog pond GUI experience. Previous has shown that physical interfaces can provide easily accessible interfaces and encourage repeat engagement [19] but does not necessarily increase the learning outcomes when comparing a TUI to a GUI [10,11]. These studies have shown that TUIs can increase engagagemnt time with programming activies for informal learning environments compared to GUI. These GUIs used in comparision are typically mouse and keyboard. In the case of Frog Pond we suspect the use of a multi-touch tabltop will be a engaging and attractive format then a mouse. Here we present the results of a pilot study that indicates that this might be the case. We also propose a TUI design and the groundwork for comparing this to the enhanced GUI of a multi-touch tabletop proramming interface.

METHODS

This study concerning the pure GUI elements of the frog pond took place at the Computer History Museum located in Mountain View, California. Data was gathered on the floor over the course of 8 days spread across four weeks (primarily Saturday and Sunday, the busy days at the museum). The exhibit was placed at the entryway to the main portion of the museum, such that a majority of visitors walked by the table as they entered and exited the exhibits. Data was gathered under two separate phases: observations and surveys (consisting of two conditions: before engagement and after engagement).

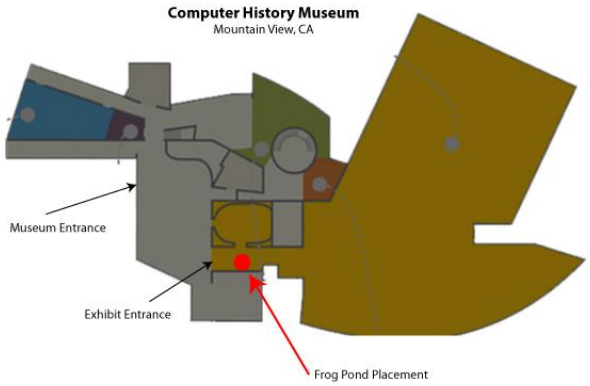

Figure 3. Frog Pond Placement in CHM

Participants

A total of 318 groups (consisting of 777 individuals) participated in this study. Recruited groups for the survey phase consisted of at least one adult and one child between the ages of 8 and 13. The demographics of the recruited dyads provided some insight into the visitor population of the Computer History Museum and it is most likely not a representation of general public. Of the 27 dyads in the study, 14 of those reported speaking another language other than English at home (including, Spanish, Chinese, Tamil, Telugu, Danish, German, Finnish, Croatian, Hindi, Russian and Cantonese). Ten dyads identified themselves as Asian American; thirteen as White; three as other; and one as Black. Twenty of the 27 parents had at least a four-year college degree with 13 of those having a master's degree of higher. This sample suggests that visitors to the Computer History Museum are both very diverse and very well educated.

Observation Condition

As part of the naturalistic observations phase we took notes on 291 groups (totaling 722 individuals) and a gender distribution of 68.8% male and 31.2% female. Average observed (i.e. engaged at the table) group size was 2.5 persons (SD = 1.3 persons). Engagement times ranged from a few seconds to over 60 minutes.

To gather observations researchers positioned themselves at a nearby table (six-eight feet away from the multi-touch table placement) Observations tracked engagement time, group size, approximate age, general behavior and insightful quotes. In general researchers did not engage with visitors but happily entertained any inquiries and feedback from visitors.

Survey Condition

In the baseline condition attitudes and knowledge surveys were conducted before engagement with the frog pond. In the exhibit study condition exhibit play attitudes and knowledge surveys were conducted after engagement. A total of 27 guardian – child dyads were recruited for the baseline and exhibit studies, (27 guardians and 28 children) with the average child age of 10.6 years (SD = 1.4 years) and a gender distribution of 59.2% male, 40.8% female.

Of the 27 dyads recruited 13 pairs participated in the baseline condition, with an average child age of 11.2 years (SD = 1.6 years) and gender distribution of 61.5% male and 38.5% female. Participants in this group were asked to do surveys before interacting with the frog pond exhibit. Parents filled out a demographic survey while children were asked a series of attitude questions followed by a proctored programming knowledge activity.

The other 14 of the 27 dyads participated in the exhibit condition, where they first engaged with the frog pond exhibit for eight minutes before being asked to fill out the surveys and participate in the programming activity. Child ages for this group averaged to 10.1 years (SD = 1.1 years) and a gender distribution of 57.1% male and 42.9% female.

Procedure

As mentioned earlier, baseline dyads that agreed to partake in the study filled out consent forms. They were then asked to participate in an attitude and demographic survey to help us improve the quality of the *Make Software* future exhibits. Before the survey, children were asked what brought them to the museum and what they were most excited to see or do. Then the children were verbally administered an attitude survey consisting of a 16 questions based on a 5-point Likert-scale. The 16 questions focused on four concepts: 'Enjoyment' of programing, 'Future Interests' in programming, 'Identity' as a programmer, and perceived 'Competence' in programming. Listed below is a sample questions for each category in the survey:

Enjoyment:

- "I think computer programming can be fun"

Future Interests:

- "I am interested in learning more about programming"

Identity:

- "I would be embarrassed to be known as a good programmer"

Competence:

- "I could get good grades in a programming class"

Experience:

- "I have spent a lot of time programming"

Meanwhile the guardians were provided with a demographic survey and questions asking their estimation of the child's programming experience. Guardians were asked to the child's experience in the following manner, "How much do you think your child knows about computer programming? (circle one) Not Much/Some/A Lot". Note that these three response types were then converted to a 5-point Likert-scale with a conversion to 1/3/5, respectively. When parents circled two adjacent options, these were recorded as 2 or 4. The responses from this demographic survey and the child attitude survey were averaged to compute the experience level for each participant.

After the child finished the attitude survey they were asked to participate in a programming exercise. Here the researcher presented the child with a print out of pseudo code for a NetLogo simulation along with a glossary of terms used within the code (see Item 1 at the end of the paper). The code consisted of a three functions *setup*, *go* and *fly* which defined a flocking behavior for three hundred 'bird' agents. Children were asked three open ended questions to gauge their understanding of the code, agent behavior and the system. First they were asked to explain to the researcher what the code provided does. If not initially provided, the researcher would ask the child what sort of behavior the 'birds' would exhibit once the program ran. Once the child responded, their hypothesis was tested by executing the program. At this point the second open question was asked, "Can you explain to me what is happening? Is the behavior you expected?" The final question was to make another behavior hypothesis for when we removed a line from the pseudo code and evaluate the result, similar to the second question. Responses were recorded with audio and video. After the survey and programming activity, dyads were allowed to engage with the Frog Pond exhibit for as long as desired.

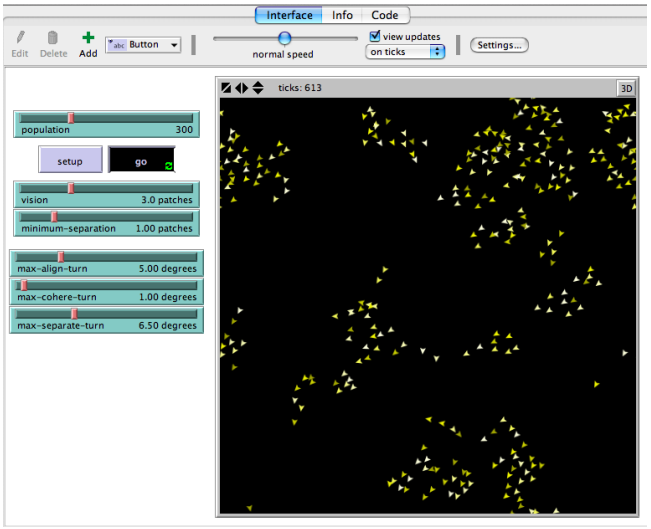

Figure 4. NetLogo Flocking Simulation Screenshot

Dyads in the exhibit condition who consented to participate in the research study were asked to engage with the Frog Pond activity for eight minutes. After that time, the researcher would ask the dyad to end engagement and then asked to complete the surveys in the same manner as the baseline condition, including demographics surveys for the adult and attitude and programming activities for the child. The entire session, game play, surveys and programming activities were actively recorded with audio and video devices.

Data Analysis

As mentioned earlier, the questions were grouped into four themes: *Future Interests* (Cronbach's α = 0.833), *Enjoyment of Programming* (Cronbach's α = 0.667), *Competence in Programming* (Cronbach's α = 0.417), and *Identity as a Programmer* (Cronbach's α = 0.310). These themes are to act as our dependent variables. To evaluate the attitudes of the sample from the pilot study we focus on two conditions, gender and reported experience level. We are not focusing on baseline and exhibit conditions as we had an age bias in the sampling for female participants (9-10 year olds in our baseline and 12-13 in the exhibit). We are using experience level (rather than age) as a main factor in our analysis because we believe that it provides a more useful indication of participant attitudes. There was no correction found between the reported experience level and subjects age.

Open ended questions will be coded and reviewed in future studies.

RESULTS

In this section of the paper we are going to discuss the results of the quantitative portions of the study, including a breakdown of the survey data and general conclusions from the naturalist observations. Qualitative assessment of the audio and video data, including the game play and programming activity will come at a later date.

Holding Time

Taken from the naturalist observations, holding time was recorded starting at the moment the visitor directed attention to the exhibit, including active engagement or attentively watching others, and ended when they were no longer watching or engaged with the exhibit. The average holding time for 723 individuals was 3.9 minutes (SD = 6.3 minutes) and the distribution can be seen in Figure 5. The break down of the holding times over age and gender can be seen in Table 1.

Clearly this break down shows that children were engaged in the exhibit for longer periods than adults, particularly girls, with an averaged engaged time of 6.3 minutes. The variability amongst children was much higher compared to adults as well.

	Males 18 years or younger	Females 18 years or younger	Males over 18 years	Females over 18 years
Average Time(minutes)	5.0	6.3	2.9	3.0
S.D. (minutes)	7.0	10.7	4.4	5.0
Total Count	211	70	286	157

Table 1. Holding times for non-recruited participants broken down by age and gender

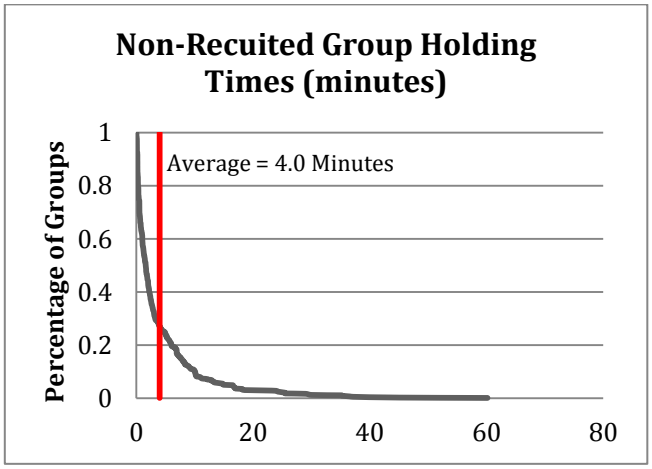

Figure 5. Holding times for non-recruited participants

Attitude Surveys

Looking at the survey results by using the four survey categories as dependent variables against gender and experience level we see a couple of strong correlations. In regard to the reported experience level there was a significant difference ($F(1,23) = 5.013$, $p = 0.035$) in the reported identity, with those with more experience having a strong attitudes towards programming being part of their identity (mean 4.44) against those with less experience (mean = 4.07). With respect to gender, there was moderate significance ($F(1,23) = 3.649$, $p = 0.069$) in the difference in reporting future interests in programming. Males tended to express a more positive attitude towards future interests in programming (mean = 4.45) then females (mean = 4.00). We did not see a significant difference within the other two themes competence and enjoyment.

Naturalist Observations

To supplement the attitude surveys, there were qualitative notes taken by researchers to get a broader understanding of user interactions. Similar to the survey demographics, the naturalistic observations saw a significant number of visitors with a large pool of prior knowledge, particularly concerning programming.

	Low Experience	High Experience	Males	Females
Enjoyment	Mean: 4.462 SD: 0.5189	4.786 0.3231	4.688 0.4425	4.545 0.4719
Future Interest	Mean: 4.1731 SD: 0.6876	4.3571 0.5162	4.4531 0.5494	4.000 0.5916
Identity	Mean: 3.808 SD:0.9023	4.464 0.6344	4.281 0.7296	3.955 0.9606
Competence	Mean: 3.4038 SD: 0.7468	3.5357 0.63441	3.5626 0.7932	3.341 0.4722

Table 2. Survey Results

*"Can you just make a fork bomb? *laughs*" M26*

"If you make it run out of memory here, I'm going to be truly amazed! F25

"First thing (name-redacted) does is a fork bomb."M23

Clearly this group of visitors is familiar with programming, using computer science jargon "fork bomb", a type of denial of service attack that tries to crash programs and systems by overwhelming system resources. This prior experience also shows up in parent child interaction:

"You have to program it. Put a loop in there. Repeat. Go!" M45 to F16 daughter

This prior knowledge wasn't limited to adults either. Kids were often excited to see an interface they were familiar with, and related it to prior programming experiences:

"Oh, I know what this is. It's Scratch" M9

And:

"It's called Blockly….It's called code.org" M6

Introductory programming experiences such as Scratch [15], Blockly [20], and Alice [14] were all referenced by users while researchers observed. This certainly suggests that non-negligible portion of the visitors to the Computer History Museum, younger and older, have at least some prior introduction to basic programming concepts, in particularly visual programming constructs.

Visitors also created their own goals within the sandbox gameplay. Typically people aimed at simple tasks, such as getting the frog to move in a desired direction. For those with longer engagement times, the goals sometimes shifted:

"I am trying to eat bugs" M16

And:

"I'm trying to beat them in the bug eating contest" M13

In one instance a user built a program to cross the pond and hatch frogs around the other player, preventing them from moving or eating.

DISCUSSION

As a pilot study there were a number of successful take away from the study and will help inform future research. Unlike typical institutions that have a broad array of topics (e.g. science, art, zoology) the Computer History Museum is more targeted with the content they is trying to convey. The history of computing, though a topic with many facets, is a topic that will likely attract a selective audience than a traditional institution such as zoo or natural history museum. The attitude and demographic surveys helped verify that bias in visitors to the Computer History Museum. A majority had at least one group member with some prior interest and/or experience related to computer science. Therefore traditional metrics such as age and gender may not be as informative. In our case, age metric was less informative then an experience metric within the ages of 8-13. Naturalistic observations support this idea as well, younger visitors were eager to bring prior knowledge of visual programming languages. Improvements to the surveys could include more in-depth surveying of prior experience, granting a higher resolution within that variable. This may allow greater insight into the value of the exhibit, and help to distinguish what experience level users can take away, (e.g. Do less experienced users take away a more comfortable attitude towards programming? Do more experienced users learn something about emergent behavior of agents?). Finding the takeaways of the exhibit can help focus which features and characteristics to enhance. Understanding how prior knowledge shapes the experience with interactive exhibits can help inform the design of more catered learning experiences.

DESIGN IMPLICATIONS

In this brief study there was a strong connection with the visual style of the programming interface as it illicit recalls to prior experience in programming, even if it was to something dissimilar to Scratch (i.e. Alice). This recall certainly supports the design decision to mimic an established format and lowers the barrier for some parties. In moving forward with the tangible interface, it assures us that visual form is a proven format.

TANGIBLE DESIGN

The overall design of the Frog Pond experience is focused on encouraging APE and fostering a collaborative environment between friends, family, and strangers. This work expands the prior work by building a complementary TUI for the Frog Pond exhibit. As mentioned in the Related Work, TUIs have been shown to increase engagement time and produce positive attitudes towards the activity. These are beneficial characteristics for fostering APE experience.

Design Principles

The tangible program interface aims to replace the GUI drag and drop of virtual blocks with physical representations. We believe there are a number of

affordances of physical representations that support their inclusion in learning activities. For example, TUIs have been impactful in increasing engagement times compared to over their mouse and keyboard counterparts [19].

Mirroring the design principles that informed the GUI interface, the TUI aims to follow the fundamental components to APE. TUIs can also offer more affordances to the design with respect to these APE components. The APE concept of shifting the visitor's role away from the recipient of instructions is embodied in this project via the form factor. Creating a TUI has a large impact on this form factor, with the intention of providing a more accessible platform for more social engagement. In this case we are using wooden puzzle pieces as programming blocks (mirroring their digital counterpart). The expectation is that an assortment of blocks will further enlarge the circle of those who can participate. In the GUI it is only feasible for one person to drag blocks into the program stack, where as with a tangible block, a number of people can pick up blocks and place them in order, allowing more people to be engaged thus creating more opportunities for discussion and dialog. Supporting collaborative engagement with the Frog Pond activity could enhance the learning experience by prompting discussion between users and maintain attention.

The use of puzzle pieces is an informed design decision. As stated by Horn [6] certain objects in a scenario can evoke cultural form which may be leveraged in a tangible interactions . We hope to leverage the cultural forms of puzzles and wooden blocks that offer a variety of affordances for combining pieces and building chains. In our case, the wooden programming blocks are shaped as puzzle pieces, are meant to evoke the cultural form a jigsaw puzzles. A user seeing these jigsaw puzzle pieces will hopefully leverage their prior knowledge that these pieces interlock in a variety of ways in order to complete a goal. Evoking the idea of jigsaw puzzles and providing physical pieces reduces the dependency on screen based interactions and could separate the association to video games (evoked by a digital display). Previous work as shown that adults can have negative perceptions of video games in informal learning environments [4] which could discourage engagement.

Figure 6. Photo of TUI at CHM

Aesthetics and Kinesthetic Design

The design of the tangible programming interface TPI was directly informed by the graphical user interface (GUI) used in the Frog Pond. Visual cues such as block color and spatial indentation for control loops follow the same format in both interfaces. This design decision was made to support furture work that will directly compare the use of TPIs as an alternative to GUIs.

The physical aspect of this TPI can be described in two parts, the programming blocks and the workspace. Each block represents a programming command that can be strung together in a chain to form a program. Secondly, the box serves as a workspace, in which the aforementioned blocks must placed onto properly in ensure that the program executes.

We used a laser cutter to shape the blocks out of wood, approximately 3.00"x1.50"x0.25" in dimensions. Wood was chosen for its more traditional role in the cultural form of toy blocks. The intention is to leverage the cultural form ideas of classic wooden blocks to make the interface more toy-like and thus more inviting. Following the GUI representation, the physical version has the same color palate, with standard command blocks are blue (e.g. 'hop', 'chirp', 'left', 'right', 'eat'), control blocks (e.g. repeat, if/else) are yellow, and the agent creation block (e.g. 'hatch') is purple. Coloring was done with a food coloring/alcohol stain to maintain a wood grain patterns, as opposed to an acrylic based paint, which would override the woods natural grain resulting in a smooth more plastic like touch.

Blocks interlock with each other in a puzzle piece style format, allowing them to form linear chains. Tolerance between interlocking pieces was adjusted to make it easy to remove from one another, but tight enough to limit the extraneous movement side to side. Special control blocks are used to represent control statements such as for loops and if/else statements. The control blocks are shaped in such a way that they make a visual indentation line of blocks, physically shifting the nested blocks, consistent with the Frog Ponds GUI.

The workspace is a simple long box with dimensions 24.0"x6.0"x3.0". Similarly to the blocks, the workspace has been constructed from balsa wood with the aim of providing an air of familiarity to traditional wooden toys. On the top of the box there is a green start block fixed in position on the left side to indicate where visitors can begin assembling their program blocks.

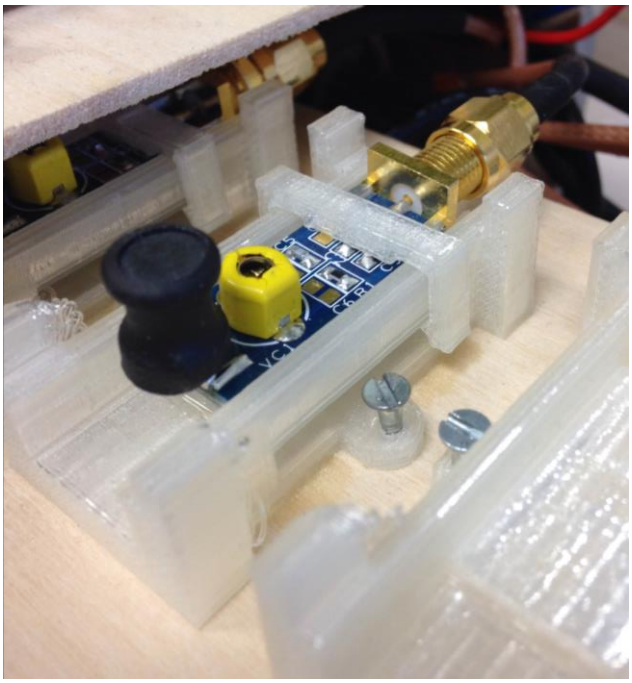

Figure 7. Close up of RFID Antenna Clip System

Technical Implementation

At a high level, the reading of a "block program" is done via a system of RFID antennas that read RFID tags embedded in the blocks. There are 16 RFID antennas that are spaced out in the base of the workspace in such a way that when the blocks are interlocked in a chain, the RFID tags align with each antenna. There are two readers in the workspace base that run in parallel, each multiplexing through eight antennas while reading the blocks RFID unique identification number (UID). The two RFID readers are controlled by a Raspberry Pi, running a python script. The onboard Raspberry Pi monitors a simple push button, which on press (to play the program) commands the RFID readers to read UID tags for blocks on the workspace and broadcasts a list of UIDs to the USB port via serial communication protocol.

All of the aforementioned technology is contained within the workspace housing effectively making the workspace a simple USB/Serial device. Currently the Frog Pond exhibit is built off of Google Dart web frame working, making the game a simple web application. In order to interface the USB/Serial device to a web application requires an intermediate step. In this scenario we used a python script that parses incoming serial data and hosts a web socket server, which the Frog Pond application connects to as a client. Once the UIDs have been passed to the web application, they are parsed through a hash table of known blocks and translated to their block type (e.g. 'hop', 'chirp', 'repeat', 'if','endif', 'hatch' etc).

Prototype Design Considerations

As a working prototype there were a few design decisions made to accommodate the common turmoil of hardware development. The internals of the workspace box are all mounted on an internal frame, which can be slid out from the external housing, allowing access to all the components without having to disconnect a myriad of wires. Additionally the RFID antennas are affixed to the internal frame using a custom 3-D printed housing and clip system, allowing for simple replacement when issues arose with particular antennas, see Figure 7. Finally the introduction of the Raspberry Pi into the system allowed for a more streamlined architecture, offloading all the hardware specific drivers for the RFID readers from the host computer (most often an embedded PC in a multi-touch table) onto an embedded component, making testing of various displays a much easier task.

Initial Impressions

When the prototype was tested at the CHM, it was well received, even in a prototype state with minimal block support (no control blocks, repeat/if-else). Compared to the GUI counterpart, the TUI was a much slower paced interaction, as physically manipulating the blocks requires more work than a simple drag and drop interface. Although there were signs pointing to more thought-out program because of the effort required a more in depth study will be needed. Another observation that may merit additional study was the desire to build the longest program possible with the blocks. With the GUI approach, visitors often ran simple programs ('left-hop') of one or two commands to move the frog in a dead reckoning manner, whereas the TUI seemed to encourage length by the physical presence of additional blocks.

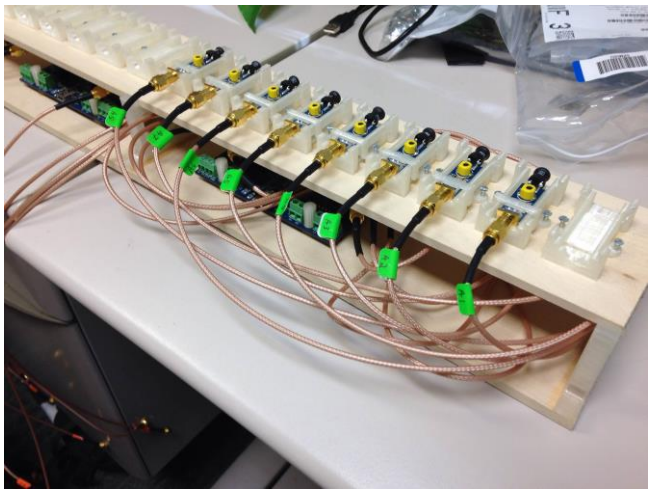

Figure 8: Internal components of TUI prototype

FUTURE WORK

Moving forward there are number are areas of research we would like to explore. First off we plan to iterate on the attitude survey and collect additional data to evaluate the performance of the Frog Pond experience with a pure graphical interface. A major limitation in this study was the age discrepancies between female baseline and female exhibit subject age so in the following studies we will ensure to eliminate that bias so baseline and exhibit conditions can be comparable and used as an independent variable.

Furthermore, in this pilot study we observed and evaluated a 'programing' activity with the preexisting GUI Frog pond exhibit. This work serves as the foundation for future work that will incorporate a tangible programming interface with the Frog Pond exhibit. We will compare GUI and TUI implementations of the learning activity to better understand how TUIs contribute to learning and attitude outcomes over and above traditional GUI interfaces. We will also investigate changes in engagement or attitudes, particularly with respect to engagement time and gender attitudes.

CONCLUSION

The graphical user interface of Frog Pond showed promising results as an APE activity, with a typical holding time distribution for informal learning environments and positive associations with prior programming experiences. In regards to the pilot study, the survey was helpful in uncovering a knowledge bias in the typical CHM visitor, often having more prior knowledge about programming then a typical educational intuition visitor. We have also proposed a design and study for a tangible interface that offers similar affordances the graphical version but extends this work by incorporating cultural forms and offering a more collaborative platform. Future work will evaluate this tangible format for leveraging the benefits of physical interactions compared to a GUI interface.

ACKNOWLEDGMENTS

We thank Dr. Michael Horn, Richard Davis, Alex Lux, David Weintrop, Sarah R. D'Angelo, and the Computer History Museum. We gratefully acknowledge support from the NSF (#1234-2012-ABC). Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

1. Horn, M. and Jacob, R. Designing Tangible Programming Languages for Classroom Use. *In Proc. Tangible and Embedded Interaction TEI'07*, ACM Press (2007), 159–162.
2. Horn, M., Solovey, E., and Jacob, R. Designing Tangible Programming Languages for Classroom Use. *In Proc. Interaction Design and Children IDC'08*, ACM Press (2008), 194–201.
3. Horn, M., Weintrop, D., and Routman, E. Programming in the Pond: A Tabletop Computer Programming Exhibit. *Proceedings ACM Human Factors in Computing Systems (CHI'14 extended abstracts)*, (2014).
4. Horn, M.S., Leong, Z.A., Block, F., et al. Of BATs and APEs : An Interactive Tabletop Game for Natural History Museums. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'12)*, ACM Press (2012).
5. Horn, M.S., Solovey, E.T., Crouser, R.J., and Jacob, R.J.K. Comparing the use of tangible and graphical programming languages for informal science education. *Proceedings of the 27th international conference on Human factors in computing systems CHI 09 32*, (2009), 975.
6. Horn, M.S. The role of cultural forms in tangible interaction design. *Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction - TEI '13*, (2013), 117.
7. Humphrey, T. and Gutwill, J.P. *Fostering Active Prolonged Engagement: The art of creating APE exhibits.* Exploratorium, 2005.
8. Kaczmarczyk, L. and Dopplick, R. *Pathway to Success Rebooting the Pathway to Success.* ACM, 2014.
9. Margolis, J. and Fisher, A. *Unlocking the clubhouse: Women in computing*. MIT press, 2003.
10. Marshall, P., Cheng, P.C., Luckin, R., and Keynes, M. Tangibles in the Balance : a Discovery Learning Task with Physical or Graphical Materials. *In Proc. Tangible, Embedded, and Embodied Interaction TEI'10*, 153–160.
11. Marshall, P. Do tangible interfaces enhance learning? *Proceedings of the 1st international conference on Tangible and embedded interaction*, (2007), 163–170.
12. Montemayor, J., Druin, A., Farber, A., Simms, S., Churaman, W., and D'Amour, A. Physical programming: designing tools for children to create physical interactive environments. *Proceedings of the SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves Minneapoli*, 4 (2002), 299–306.
13. Oh, H., Deshmane, A., Li, F., Han, J.Y., Stewart, M., and Tsai, M. The Digital Dream Lab: Tabletop Puzzle Blocks for Exploring Programmatic Concepts. *Proc. TEI'13*, (2013), 51–56.

14. Pausch, R. and Dann, W. ALICE : A 3-D Tool for Introductory Programming Concepts. *Journal of Computing Sciences in Colleges 15*, 5 (200AD), 107–116.

15. Resnick, M., Maloney, J., Monroy-hernández, A., et al. Scratch : Programming for Everyone. (2007).

16. Sapounidis, T. and Demetriadis, S. Touch your program with hands: Qualities in tangible programming tools for novice. *Proceedings - 2011 Panhellenic Conference on Informatics, PCI 2011*, (2011), 363–367.

17. Schweikardt, E. and Gross, M. The robot is the program: interacting with roBlocks. *Proceedings of the 2nd international conference on Tangible and embedded interaction*, ACM (2008), 167–168.

18. Wang, D., Wang, T., and Liu, Z. A tangible programming tool for children to cultivate computational thinking. *The Scientific World Journal 2014*, (2014).

19. Xie, L., Antle, A., and Motamedi, N. Are tangibles more fun?: comparing children's enjoyment and engagement using physical, graphical and tangible user interfaces. *Proceedings of the 2nd international conference on Tangible and embedded interaction.*, ACM (2008), 191–198.

20. Blockly. https://code.google.com/p/blockly/. Visited Dec. 2014

Item 1:

```
define setup
    begin
        create 300 birds
    end



define go
    begin
        repeat forever [
            ask birds [ fly ]
        ]
    end



define fly
    begin
        if distance-to closest-bird < 1 [
            turn-away-from closest-bird
        ]

        if any other birds nearby [
            turn-towards one bird nearby
        ]

        forward 1
    end
```

GLOSSARY

define is used to create a new program *function*

setup is called when you press the *setup* button

go is called when you press the *go* button

create adds birds in random places

birds are objects that have a position on the screen and a direction that they're pointing in

ask tells each individual bird to do something

if is used to ask a question in a program

distance-to returns the distance between the current bird and another bird

turn-away-from turns the current bird a little bit away from another bird

turn-towards turns the current bird to move closer to another bird

forward moves the current bird forward a little