TUI, GUI, HUI: Is a bimodal interface truly worth the sum of its parts?

Amanda Strawhacker, Amanda Sullivan, and Marina Umaschi Bers
The DevTech Research Group
Tufts University
105 College Ave Medford, MA 02144
{amanda.strawhacker, amanda.sullivan, marina.bers}@tufts.edu

ABSTRACT
This study aims to explore the relative differences in efficacy of three different computer programming interfaces for controlling robots designed for early childhood education. A sample of N=36 kindergarten students from 3 different classrooms participated in this research. Each classroom was randomly assigned to one of the following three conditions: a tangible user interface, a graphical user interface, and a hybrid user interface. Comparisons between the three conditions focus on which interface yields better understanding of the programming concepts taught. Implications for designing developmentally appropriate computer programming interfaces for early childhood education are discussed.

Categories and Subject Descriptors
H.5.2 [**Information Interfaces and Presentation**]: User Interfaces; K.3.1 [**Computers and Education**]: Computer Uses in Education

Keywords
programming interfaces, robotics, early childhood, tangible programming, kindergarten

1. INTRODUCTION
While the majority of research on robotics and computer programming in education focuses on middle school and beyond, teaching these subjects during the foundational early childhood years can be an engaging and rewarding experience for young learners [1]. Furthermore, when children program, their role in the learning process changes from passive listener to active, self-directed learner, a learner with a personal purpose [10]. Previous research has shown that children as young as 4 years old can build and program simple robotics projects [4; 5; 11], and that kindergarteners can successfully build computational thinking skills through engineering and computer programming activities [2]. In turn, learning how to program has been shown to have a positive impact on young children's sequencing abilities, a foundational math and literacy skill set [8; 9]. Children exercise sequencing when organizing words in a sentence, sorting colors, recognizing patterns, and performing procedural thinking tasks from a variety of cognitive domains [9].

IDC '13, June 24-27 2013, New York, NY, USA

In order to make programming robots developmentally appropriate and therefore more accessible in early childhood education, the DevTech Research Group at Tufts University developed a unique programming language called CHERP (Creative Hybrid Environment for Robotic Programming) [3; 7]. The CHERP software allows children to create physical programs using interlocking wooden blocks or onscreen icon-based programs. The CHERP language is composed of easily-understood icons that represent actions for a simple robot to perform. For example, a block with the word "SPIN" and a picture of an arrow swirling in circles will make the robot spin around. Based on the way the wooden blocks and the onscreen icons fit together, children never encounter a syntax error. To make testing easier, programs can be compiled and sent to the robot in a matter of seconds [3].

CHERP was inspired by early ideas from tangible programming [11] which have inspired decades of exploration and a variety of tangible languages for children in research labs around the world [12; 13]. In contrast to graphical programming, which relies on pictures and words on a computer screen, tangible programming uses physical objects to represent the same concepts. CHERP provides a system that allows children to construct physical computer programs by connecting interlocking wooden blocks. CHERP's wooden blocks contain no embedded electronics or power supplies. Instead, children use CHERP's blocks to create the program for their robot and then take a picture of it using a standard webcam connected to, or embedded within a computer. The picture is converted into digital code using the TopCodes computer vision library and downloaded to the robot [3]. Although CHERP can be used to program a variety of educational robotics kits, this study uses the commercially available LEGO®WeDo™robotics construction set (see Figure 1).

With CHERP, children can choose to use either the tangible wooden blocks, the graphical onscreen program, or some combination of both of these interfaces, in order to program their robots. Anecdotal evidence from pilot studies (including researcher observations and conversations with

Figure 1: A sample robot with tangible and graphical interfaces

classroom teachers) indicate that problem solving strategies employed by students, as well as their subsequent mastery of programming concepts, may be directly influenced by the interface. However, as of yet, little research has been done comparing student learning using these different interfaces. The present study fills in this gap by examining students' mastery of basic computer programming concepts across three CHERP interfaces: tangible, graphical, and hybrid.

2. METHOD

The purpose of this study is to compare student learning using three different CHERP programming interfaces: a Tangible User Interface (TUI), Graphical User Interface (GUI), and Hybrid User Interface (HUI).

2.1 Sample

Our sample was derived from three Kindergarten classes from a public early childhood school in urban Boston. The school hosts Montessori-style classrooms as well as traditional-education ones. According to the 2012-13 Massachusetts Department of Early and Secondary Education Accountability Data Report, the student body (200 children in PreK-2nd grade) is 72.5% Hispanic, 69% Limited English Proficient, 65% Free or Reduced Lunch, and 15.5% Special Education. All participating classrooms were inclusive education settings, but due to confidentiality reasons, the school could not release information about specific students with cognitive, behavioral, or learning disabilities. For this reason, we cannot address the number and distribution of students with any type of disability in our sample.

2.2 Procedure

A sample of N= 36 Kindergarteners in three classes at one public, early childhood school in the Boston area participated in this research. Each classroom completed the same 13 hour introductory robotics and programming curriculum, taught by the same research assistant over 13 one-hour sessions. Children worked in groups of 2-3 to complete different programming tasks assigned throughout the lessons. Over the 13 sessions, two programming units were implemented to present two specific programming concepts to the children: sequencing and repeat-loops. Each unit was built around a programming project that encouraged students to use the skills covered in the lessons (for example, the sequencing unit project challenged students to program their robots to "dance" the Hokey Pokey). The robotics curriculum was taught as one of their regularly scheduled "Special" classes (e.g. music, art, etc.), and was tailored to align with a curriculum on animals that was already being taught in the kindergarten classrooms. The three participating classrooms were assigned to either the TUI, HUI, or GUI condition.

2.2.1 TUI Classroom

The TUI classroom consisted of N=15 kindergarten students in a traditional public school setting. Children in this group completed the programming curriculum described above using only the CHERP tangible wooden blocks. Because CHERP requires a computer to upload and compile the tangible programs to the WeDo^{TM}robots, there were two stations set up in the classroom where children could upload their tangible programs and observe their robots performing the actions. At the testing stations, children were not permitted to edit their programs using the computer, although they were able to observe the teachers using the computers embedded webcam to upload/compile their tangible programs.

2.2.2 GUI Classroom

The GUI classroom consisted of N=7 kindergarten students in a Montessori instructional setting. The sample size for this group was smaller than that of the TUI and HUI groups because these seven children represented only a subset of their Montessori classroom. Montessori learning environments typically incorporate mixed-aged groups. For the purpose of this study, we used only those children aged 5-6 years, the same age as the children in the other two classrooms. The GUI class completed the programming curriculum described above using only the CHERP graphical interface. Each group of children shared one computer for testing and programming their robot.

2.2.3 HUI Classroom

The HUI classroom consisted of N=12 kindergarten students in a traditional public school setting. They worked in small groups using both the graphical and tangible programming interfaces. At each groups work station, they shared one computer and a large supply of tangible programming blocks for building, uploading, and testing their programs. Children could freely switch between the two interfaces at any point in the programming task and could even use both while working toward a single goal. Unlike the TUI group, which used teacher-managed testing stations to test their tangible programs, children in the HUI classroom used their groups computer to compile and test all programs on their robot.

2.3 Assessments

Children worked in groups during their lessons, but we wanted to measure each individual child's level of programming knowledge during the final assessments. In order to do this, all students completed programming tasks called "Solve-Its." These Solve-Its, developed by the DevTech Research Group, were designed to address children's understanding of the concepts taught in the curriculum. The tasks were comprised of two sequencing tasks (one hard and one easy), two

repeat loop tasks (one hard and one easy), and a culminating task. For four Solve-Its, students were given all the programming icons they would need to complete the challenge. In the culminating task, children were given extra instructions, and told that they would need to select the correct components to solve the task. Each Solve-It had only one correct answer and childrens' programs were scored on how precisely they matched this answer. In order to determine the difficulty of the tasks and the accessibility of the stories, each Solve-It was pilot-tested with Kindergarten children from a mixed age Montessori classroom that had completed a robotics and programming curriculum.

The Solve-Its were presented as games in which the researcher tells the children a story about a robot. Working individually, children used paper CHERP programming icons to sequence the program described in the story. Prior research has demonstrated that young children have an easier time building a sequence when there is a context that they can understand and relate to from their own experience [6; 9]. Because of this, all of the Solve-Its used real-world examples that a young child would be familiar with. For example, one Solve-It asked children to recall the song "Wheels on the Bus," as part of the context of the story.

3. RESULTS

Statistical comparisons were drawn between the three classrooms (TUI, GUI, HUI) in order to explore whether the programming interface was a mediating factor on student performance on individual programming tasks (Solve-It). Descriptive statistics show that the TUI group had a higher proportion of correct answers for all but one of the programming tasks (see Table 1).

A one-way analysis of variance (ANOVA) was performed in order to compare differences between the three interface groups (TUI, GUI, HUI) on Solve-It scores. Statistically significant differences between the groups were found only on the easy sequencing task [$F(2, 21) = 4.409$, $p = 0.025$]. There were no significant differences between the classes on any of the other tasks.

Post-hoc testing was used to further explore the differences between classrooms on the easy sequencing task. Tukey Test results indicated that although there was no significant difference between the TUI and HUI classrooms, or HUI and GUI classrooms, there was a significant difference between the TUI and GUI classrooms (p¡.05) on the easy sequencing task.

4. DISCUSSION

A cursory glance at the data reveals that the group with the tangible interface curriculum outperformed both the graphical and the hybrid groups in terms of easy and hard sequencing assessments, easy repeat loop assessments, and the culminating assessment . The GUI group performed the highest of the three on the hard repeat loop assessment, but relatively poorly on all the rest. The HUI group averages did not exceed the averages of the other two classrooms on any of the five Solve It tasks.

It was interesting to see that in the HUI group, having the option to choose their interface did not seem to provide the children with any educational advantage, and actually seemed to hinder their understanding of some programming concepts. For example, they scored the lowest averages of all three groups on the hard sequencing Solve It, which indicates that their understanding of long (5-instruction) chains of ordered commands was not as well developed as either the GUI or the TUI group. They also performed the worst of the three groups on the culminating task, displaying a lack of understanding about repeat loops generally, and especially about when to be selective in choosing programming instructions. One possible explanation for these results is that the HUI group was so inundated with interface options for building their programs that they had a harder time learning to par down programs to necessary instructions. This could also have led them to have difficulty distinguishing between multiple program instructions as their robot performed them, leading to a disordered understanding of sequential actions.

As evidenced by their assessment scores, the TUI group developed a deeper understanding of the core concepts of sequencing a program and repeat-loop syntax. This could be due to the different instructional styles of the regular teachers in each group, or possibly the proportion of inclusive-education students in each group. However, another explanation is that the simultaneous introduction of a dual interface was simply overwhelming for the HUI group and impeded their ability to focus on the programming concepts behind the tools. The TUI group was able to interact with an interface that leveraged already-familiar blocks, and consequently the students may have been able to devote more attention and cognitive resources to understanding the relationship between the blocks and the movements of their robots.

The low scores of the GUI group served to underscore the importance of having an alternative way to make the programming ideas concrete. Even introducing something as low-tech as wooden blocks or pieces of paper with programming icons on them can enable the children to experiment with program building. The manipulability of tangible interfaces apparently helps children to mentally model abstract programming ideas, and the more time spent using tangible interfaces, the deeper their demonstrated learning.

4.1 Limitations

Part of the low performance by the GUI group may be explained by the fact that the students in this classroom usually received Montessori instruction, which is highly focused on tangible learning environments. In this study, however, these children were not exposed to the tangible interface at all, which may have been challenging for them. Also, since Montessori classrooms are mixed-age and our study called only for participants aged 5-6, the GUI group was composed of only those members of the Montessori class who fit the age criteria, approximately one-third of their total classroom. The HUI and TUI groups, both traditional education style classrooms, were each made up of an entire classroom, and not just a portion of one.

Another possible limitation of this study was the inability

	Easy Sequencing	Hard Sequenging	Easy Repeats	Hard Repeats	Culminating Task
TUI	73.3% (n=15)	73.3% (n=15)	46.7% (n=15)	0% (n=15)	33.3% (n=15)
HUI	66.7% (n=12)	33.3% (n=12)	41.7 % (n=12)	0% (n=12)	18.2% (n=11)
GUI	12.5% (n=7)	50% (n=6)	66.7% (n=6)	16.7% (n=6)	16.7% (n=6)

Table 1: Percentage of Correct Answers on Programming Task by Interface

to distinguish which students in the sample had cognitive or behavioral disorders. The school from which the sample was derived is committed to inclusive education, and approximately 15% of the student body is diagnosed with one or more cognitive or behavioral disorders. All three groups included students with varying degrees and types of learning disabilities, but the specific identities of these children were not revealed to the instructing researchers.

Finally, all three classrooms had very different group dynamics. The individual styles of each lead teacher and the personalities of the children resulted in different emphases on classroom expectations and priorities (i.e., high self-regulation, respectful interactions with adults and peers, creative expression). Because of this, the three groups had each cultivated a distinct classroom culture, which contributed to the learning environment of each student.

4.2 Future Work

The work presented in this paper provides exploratory findings related to childrens experiences and programming concept mastery using various CHERP interfaces. In order to gather more generalizable data on concept retention and classroom experiences, further research should include a larger sample size, collect longitudinal data, and examine both individual learning trajectories and classroom trends.

As part of the National Science Foundation funded Ready for Robotics Project (DRL-1118897) the DevTech Research Group is currently expanding the present study in several ways. Case study data is currently being collected from the children in this study to examine common and outlying learning trajectories. Additionally, the curriculum is being extended to include a unit in which all three groups complete programming challenges using both interfaces. Data collected from this wave of the study will highlight any order effects based on which interface children were introduced to first.

5. CONCLUSION

Prior research has demonstrated that young children can master basic computer programming concepts. Perhaps more importantly, new technologies can facilitate students' problem-solving abilities, computational thinking skills, and social-emotional development. Research is essential to understanding the impact of these new technologies on student learning. Future work should concentrate on both designing new technologies for young learners and continuing to investigate the most developmentally appropriate applications of those tools.

6. FINANCIAL SUPPORT

This project received generous funding from the National Science Foundation (NSF Grant No. DRL-1118897, DRL-0735657).

7. REFERENCES

[1] M. U. Bers. *Blocks, Robots, and Computers: Learning about Technology in Early Childhood*. New York, Teacher's College Press, 2008.

[2] M. U. Bers. The tangiblek robotics program: Applied computational thinking for young children. *Early Childhood Research and Practice*, 12, 2010.

[3] M. U. Bers and M. Horn. *Tangible programming in early childhood: Revisiting developmental assumptions through new technologies*. Greenwich CT, Information Age Publishing, 2010.

[4] M. U. Bers, I. Ponte, K. Jeulich, A. Viera, and J. Schenker. Teachers as designers: Integrating robotics into early education. *Information Technology in Childhood Education*, pages 123–145, 2002.

[5] E. Cejka, C. Rogers, and M. Portsmore. Kindergarten robotics: Using robotics to motivate math, science, and engineering literacy in elementary school. *Int. J. Eng. Educ.*, 22:711–722, 2006.

[6] R. Fivush and J. M. Mandler. Developmental changes in the understanding of temporal sequence. *Child Development*, 56:1437–1446, 1985.

[7] M. S. Horn, R. J. Crouser, and M. U. Bers. Tangible interaction and learning: The case for a hybrid approach. *Special Issue on Tangibles and Children, Personal and Ubiquitous Computing*, 16:379–389, 2011.

[8] E. Kazakoff and M. U. Bers. Programming in a robotics context in the kindergarten classroom: The impact on sequencing skills. *Journal of Educational Multimedia and Hypermedia*, 12:371–391, 2012.

[9] E. Kazakoff, A. Sullivan, and M. U. Bers. The effect of a classroom-based intensive robotics and programming workshop on sequencing ability in early childhood. *Early Childhood Education Journal*, 2012.

[10] S. Papert. *Mindstorms: Children, Computers, and Powerful Ideas*. New York, Basic Books, 1980.

[11] R. Perlman. Using computer technology to provide a creative learning environment for preschool children. *Logo memo no 24*, page 260, 1976.

[12] H. Suzuki and H. Kato. Interaction-level support for collaborative learning: Algoblock – an open programming language. *Proceedings Computer Support for Collaborative Learning CSCL*, pages 349–355, 1995.

[13] P. Wyeth. How young children learn to program with sensor, action, and logic blocks. *International Journal of the Learning Sciences*, 17:517–550, 2008.