A Case Study on Designing Interfaces for Multiple Users in Developing Regions

Kurtis Heimerl
kheimerl@cs.berkeley.edu

Janani Vasudev
janani@ischool.berkeley.edu

Kelly G. Buchanan
kellybuchanan@berkeley.edu

Tapan Parikh
parikh@ischool.berkeley.edu

Eric Brewer
brewer@cs.berkeley.edu

University of California, Berkeley
Berkeley, CA 94720-1776

ABSTRACT

Computer assisted learning (or E-learning) is used broadly in the developed world. However, comparable technologies are only recently beginning to be used in rural and developing regions. In these environments, obstacles to the successful deployment of educational software include a lack of basic infrastructure, low student attendance, necessary sharing of resources, and the participants'—both teachers and students—unfamiliarity with communications technology. To illustrate these issues in detail, we present a design study for Metamouse, a system for sharing single user software on a single computer with multiple mice.

We designed Metamouse during a one and a half month long study in low-income primary schools in Bangalore, India. We iterated through two primary usage paradigms, competitive and collaborative, working with grade four and five students. In these populations, we found that students had widely varying mouse skills, and that even amongst competent users, interface confusion presented significant barriers. Given this, interface tasks that are known to have a cost in usability, such as mode switching or complicated interaction models, had a severe impact on students' ability to use the technology. We discuss interface issues that result from sharing practices that are the norm in these regions. We also discuss issues that generalize across educational application design in the developing world.

1. INTRODUCTION

Computer assisted learning in the developing world has been the focus of a great deal of research. It has the potential to mitigate a number of common concerns: low teacher attendance, limited literacy, high student absenteeism, as well as many others. Much formal work on interfaces for children in these areas focuses on game content [15, 30], rather than on appropriate basic interfaces for users in the developing world. Though content research is quite valuable, we believe that basic interface design has been neglected and is an important area for work.

ACM DEV'10, December 17–18, 2010, London, United Kingdom.

We advocate an approach that acknowledges the fact that educational games in the developing world are commonly *shared* rather than played individually. Single Display Groupware (SDG) [29] allows for multiple users to share a single application, monitor, and computer using multiple input devices. SDG interfaces have a long history in the developed world, traditionally focused on business and education applications. Recently, researchers brought this technology to students in the developing world and showed that these techniques are intuitive and usable [24], increase learning [23], and help focus and motivate users [28].

Despite these encouraging results, it is clear that designing shared applications is difficult. Multiple input devices have actually reduced learning in some populations [23], and our own experiences have shown the many limitations of this technique.

We conducted a design study of *Metamouse*, a system for sharing existing educational content using multiple mice. Metamouse allows for the sharing of any single-player educational game by multiple users. We evaluated a number of possible interaction paradigms, competitive [24] as well as different voting schemes [10] in a month and a half long study in two low-income schools in Bangalore, India. To conduct the design study, we partnered with the Azim Premji Foundation [6], a large non-profit focused on developing educational games as well as supporting computer-based education in India.

As we iterated our design, we observed a number of lessons relevant to the design of educational interfaces (multi-user and not) for the developing world. Firstly, we learned that application designers need to account for the high variability in children's mouse skills. Children of all competence levels participated, and designers must account for this fact. Secondly, we found that competitive models of interaction are difficult to implement in a way that supports collaboration and learning. Thirdly, we observed that complicated models of interaction (such as our original metacursor design [8]) were difficult for students and often limited learning. Lastly, we found a number of common single-player design idioms that fail miserably when shared among users, including timed games, multi-modal interactions, and inflexible navigation.

The rest of this paper is organized as follows: We begin by discussing previous work that related to this project. We follow with the initial goals for the project. We then discuss our partners, the Azim Premji Foundation, and the low-income Indian government schools for which Metamouse is designed. We then describe the format of the design study itself. We cover the design iterations next, from initial prototype to to finished system. Within each iteration we note the novice mouse user errors, user interface failings, and design considerations relevant to the iteration. We then summarize our findings and discuss the design implications for both single and multiple mouse environments.

2. RELATED WORK

As stated earlier, a great deal of work on computer-based education for the developing world focuses on educational content. One example is MILLEE [15, 30], which uses cell phones to provide English language learning software. Other work focuses on the environment created by these computers [21], and its impact on education. Our work is highly focused on the problem of creating appropriate interfaces for these regions, rather than these broader topics.

Single-display groupware (SDG) is a term for collaborative, multi-user applications using one computer and display [29]. There have been several SDG applications for education [25, 31, 32] as well as work focused on designing interfaces for multiple children [28]. However, these works do not take into account the unique difficulties of computing in the developing world.

There is also a wide variety of work on developing specialized interfaces for children. Many previous works have evaluated children's mouse usage, discovering a number of problems, including trouble controlling drag and drop [3], differentiating left and right mouse buttons [12], and inaccurate clicking [13]. Many of these issues appear to improve sometime by first grade in the developed world [4, 12, 14], though whether this improvement is due to motor development or practice is not clear [11]. We encountered these mouse skills issues, even when dealing with students well into adolescence.

Recently, there has been significant interest in using these SDG techniques in the developing world [22]. Preliminary results indicate that, with appropriate content, sharing multiple mice can be intuitive and usable [24] and increase learning gains [23]. There has been work on the specifics of designing interfaces for these technologies, including scaling to over thirty users [19], reducing dominant behavior [18] and providing multiple mouse text entry [1]. Our work supports their findings, to the degree that they are applicable with the constraint of supporting legacy applications. We continue this line of work by providing design recommendations, design experiences, and a deeper analysis of the challenges inherent in designing multi-user software in developing regions.

This particular paper describes the design portion of a larger study as well as recommendations for the design of single- and multi-user educational games. Earlier works [8, 9] describe the first designs of the system, prior to this design study. Our immediately following work [10] describes a quantitative evaluation of our final Metamouse design, including an analysis of learning benefits.

A number of our design choices have been seen before. albeit in different contexts. The consensus "location-voting" scheme is very similar to "click-voting" schemes seen in prior SDG literature [5, 23] but requires no access to source code to implement, making it more feasible to support legacy applications in developing regions. Our initial "mouse-averaging" technique, as seen in the "metacursor" feature of the original design [8, 9], is highly informed by prior work [2, 20]. However, we found this model to be unfeasible for children with few mouse skills. See Section 6.3 for further discussion.

3. INITIAL GOALS

Before conducting our contextual inquiry or design study, we had a number of high-level goals for this project. These goals are detailed in this section.

3.1 Backward Compatibility

Working with existing software is a primary goal of the Metamouse project. There are thousands of existing educational games, crossing language barriers and integrating into existing lesson plans. For this reason, any interaction paradigms we develop must allow for the sharing of existing educational content.

3.2 Encourage Collaboration and Engagement

Collaboration has been shown to improve learning outcomes [16, 27]. Peer-to-peer interaction is even more important in impoverished schools in the developing world, where teachers are often unfamiliar with computers or content, if they teach at all. To this end, we hope to provide interfaces that encourage group interactions.

4. EDUCATIONAL CONTEXT

Working with the Azim Premji Foundation (APF) [6] educational technology initiative, we observed students in six schools in and around Bangalore, India to motivate the design for Metamouse. In this section, we detail APF's initiatives, as well as the computing context in the schools we visited.

4.1 Azim Premji Foundation

The Azim Premji Foundation is the largest developer of educational software in India. They have developed over 120 games under various subject categories such as English, Math, Science, and Hindi, serving grades one through eight in six different Indian languages. The foundation also provides computer hardware to the schools, including battery backup for the entire computer lab. These computer labs consist of between four and ten Windows computers, each loaded with APF content. These computers are generally not networked together, nor connected to the Internet, though we did see one school that had installed a local area network. Our primary goal was to allow multiple users to share all of Azim Premji's existing educational content.

4.2 Schools

All of the schools we visited were government-run primary schools. Government schools are free and generally cater to students of low socioeconomic status. The schools ranged in size from approximately one hundred children to well over four hundred. All of the schools taught grades one through seven, with some grades exceeding eighty students. Computing infrastructure (four to eight computers) was supplied to all of the schools by APF, allowing each to operate a full educational computer lab.

A number of smaller "feeder" schools exist which teach only grades one through five. Few of these have computer equipment. Upon finishing grade five, the students transfer to one of the larger primary schools, which is much more likely to have a computer lab. For this reason, students coming from "feeder" schools are likely to be significantly behind their peers in terms of computer skills.

In the six schools we visited, we observed many different models for sharing computing resources. Some variations include how students were paired, what role teachers play, and how students shared games. We describe these observations below.

4.2.1 Student Pairing

Students usually sat in groups of three to five per computer. We observed various ways in which students were grouped. Most schools used the class roll to organize the groups. These rolls were generally sorted alphabetically, and sometimes also split by gender, leading to the formation of single-gender groups. In some schools, there was a concerted effort to pair weaker students with stronger ones. Another method we observed was self-grouping, where the teacher did not explicitly organize the children, and they chose their own groups.

4.2.2 Teaching Structure

There were three primary models for computer instruction we found in our initial investigations. In the most common model, a dedicated computer teacher managed the computer classes. This teacher ran only the computer classroom, taking children from their primary teacher for a period of time. The instructor also assisted the other teachers in computer tasks, such as developing lesson plans and computer grading. Nevertheless, there remains some disconnect, and the computer class often does not directly support what is being taught in more traditional classes.

Alternatively, the primary teachers may instruct the computer class as well as their other classes. This allows the teacher to tightly integrate the computer learning with the rest of the lessons. However, the teacher may not be computer literate, limiting the effectiveness of the computer class. Also, class sizes often dwarf the number of computers, leading the teacher to split the class during computer hours. In this case, the remaining students would go to the library or take a recess.

Finally, primary teachers may split the class and send half to the computer room with no instructor. We were unable to observe this, as it was practiced at just one school where the computer lab was not operational during our visit.

4.2.3 Sharing Models

The schools had various ways to organize the student groups. APF suggests that the most technically skilled student take on a dominant role in group dynamics. They are designated the "monitor" and their job is to ensure the mouse is switched between every game. This technique is uncommon in practice; we saw it only in one of the six schools visited.

The other schools used a less strict sharing model; children are encouraged to share, without this explicit enforcement. If a child feels like there isn't enough sharing, they can voice their complaint to the instructor (if there is one) who will then remedy the situation. This worked if the students were confident enough to voice a complaint. Unfortunately, inexperienced students often lack this confidence.

5. STUDY DESIGN

We selected two schools for a participant observation and design study. To ensure that our designs were applicable to the wide educational context, we selected two schools that were markedly different. School A sorted the children by gender and alphabetical order. This school also had a dedicated computer instructor. School B sorted the kids alphabetically with no gender separation and had no instructor. The children either managed themselves in the classroom (the common case) or were led by their primary teacher. Both schools used only informal structures to enforce sharing; we had not discovered the "monitor" model (See section 4.2.3) at design time.

Our team consisted of three researchers and one assistant. The researchers were fluent in English and Tamil, both spoken by children on occasion. The assistant was a native speaker of Kannada and translated for the researchers when the children spoke in a language that was not understood.

The design study itself consisted of student group interviews. We selected three children from either grade 5, 6 or 7 (with all students from the same grade) and had them play an existing APF educational game using *Metamouse*. We assumed they had experience with single-mouse before, as their computer labs were either functional or had recently (within the month) been functional. The children were asked to play two games for about twenty minutes using one of the interaction models described below. The models were counterbalanced to limit any learning bias. During this time, our assistant translated the student discussion while the researchers observed their interactions. We followed this up with a brief focus group interview, after which the students were allowed to play a different game with an alternate interaction model. Upon completion, we once again conducted a focus group interview, where we asked the students to compare the two models and their experiences with single mouse sharing.

In addition to this exercise, we conducted contextual inquiries and observations during normal lab hours. The students spent their lab hours playing APF educational games according to an established lesson plan. They shared with 3-4 kids per computer. Upon detecting particularly interesting behavior from an observation, we would bring that group in for testing and interviewing. These observations happened only at school A, as school B's computer lab was under renovation.

The students were selected in differing ways at the different schools. In school A, we selected randomly from pre-formed groups. This allowed us to examine Metamouse in existing social structures. At school B, we allowed the teacher to select the students. This led to groups of students of comparable skill level, as the teachers tended to pull their brightest students first.

6. METAMOUSE

In this section, we describe the design iterations of Metamouse. Many of the incremental design changes were actually done in parallel, so the iterations below are not strictly chronological, but instead primarily conceptual.

6.1 Initial Designs

At the beginning of the design study, we implemented two conceptual models of sharing we hoped to evaluate. The first is a *competitive* model, where the students would compete to be the one to successfully navigate the game. The second is a *collaborative* model where the students were required to work together to make progress.

6.1.1 Competitive

The competitive model is conceptually quite simple, and has been implemented in a number of multi-user games [23]. In this model, each user is given their own mouse cursor and operates completely independently; one user could play the entire game by herself without any involvement from the other users.

Of course, the other users would probably not appreciate that. For this reason, we expected to see competitive behavior from the students, competing to be the one to select the correct answer and make progress in the game. This competition would hopefully spur discussion and collaboration, as users attempt to optimize their own strategies.

This model is one of the first offered by most multiple input device software developers. It is simple to understand and implement, and users tend to understand it quickly.

6.1.2 Collaborative

The basic premise of our collaborative model is that we wish to encourage the users to work together to progress through the game. This should engender more user discussion and collaboration,

Figure 1. The original version of Metamouse. The user cursors are blue, black and white and the central cursor is the "metacursor" and it is green, signaling agreement.

which have been shown to improve learning outcomes [16, 27]. To accomplish this, we provide each user with their own cursor, as in other models, but also draw a *metacursor*. This metacursor is controlled by all users equally; it is placed at the average location of all user cursors. All interactions with the game happen through this single shared cursor; the users use it to click on answers, select text boxes, and skip content.

However, while this resolves one issue, that of mapping many mice down to the one cursor an application expects, it introduces another. If the user cursors are far apart, the metacursor is near none of them; it is somewhere in between. This means that it may be on buttons or content, or may be at some nearly random location on the screen. For this reason, we drop all clicks unless the mice are near each other. Lastly, we signal the agreement with color changes; the metacursor is green when in agreement and red when not. Figure 1 demonstrates this early version of Metamouse.

This system allows for group control of an existing application. It requires all users to agree on an answer to make progress. Our initial tests in India found it to be intuitive and usable, and we hoped to see gains in collaboration and discussion in our larger design study.

See our earlier work [8, 9] for a deeper analysis and discussion of this model.

6.2 First Iteration: Failure of the Competitive Model

As we began testing the two models, it became abundantly clear that the competitive model is simply incorrect given our constraints. There are two primary reasons for this. Firstly, the existing content doesn't adequately support competitive behavior, leading to "race-clicking". Secondly, the competitive model seems to greatly reduce the amount of discussion and collaboration.

6.2.1 Race-Clicking

Firstly, the students were competing to *complete* the game, rather than *understand* the content. This had been seen in prior work [24], where children would compete to click first (called "race-clicking"). This was exacerbated by our requirement to support existing games. These applications often had games that would progress when the student provided any answer, even if the answer wasn't correct. This was a common model, seemingly simulating a test. However, here it meant that children would compete to click on *any* answer first, completely disregarding the content.

Though there may be solutions to this "race-clicking" phenomenon, but they required very specific controls in the classroom. For instance, in [26] it was found that seating children of comparable mouse skills together improved the effectiveness of competitive models. This is infeasible, as the high variability in teacher quality in these low-income schools means that we cannot expect any consistent grouping from the teachers.

We wish to note that competition itself is not the problem, but rather competitive sharing. In [26] they discuss having the groups compete in a tournament-like system. This is potentially compatible with Metamouse–each group can work cooperatively and compete together against other groups.

6.2.2 Discussion and Collaboration

The competitive model seemed to heavily reduce the amount of discussion between the students. Students were more focused on the competition than on their fellow students. Though this can be valuable given the right underlying interface (for instance if each student was required to answer a question) we could not devise a way to encourage discussion given the requirement of supporting a wide variety of existing games.

The key to understanding why collaboration decreased is to note how sharing of physical objects is different from sharing media. With the physical object (sharing the single mouse), the students have a clear inequality. They view it as unfair if just one user dominates this object. If a student is not allowed to use the mouse, it is *not* their fault. However, if we introduce multiple mice, this dynamic changes; every student now has equal access and opportunity with the software. This equality allows students to compete in a much more selfish way. If the other students cannot keep up, it is as one student told us, "their fault". This problem is exacerbated by the high variance in mouse skills among users.

This shift immediately reduces the amount of collaboration among the users. Comparable behavior has been seen in a number of other works [23]. Though literature exists on how to mitigate this behavior, it is extremely difficult to do in an application-agnostic way. For instance, in [17] students are directed to explain to partners when they click on an answer. In Metamouse we are unable to determine when a user has clicked on an answer. We know only that they clicked, and thus cannot use similar techniques.

For these reasons, we decided that the competitive model could not be modified to appropriately support sharing existing single-player educational games.

6.3 Second Iteration: Removal of the Metacursor

The final design of Metamouse no longer features the metacursor. While this model was intuitive for proficient mouse users in the United States, it did not translate well to our target users. In this section we discuss the findings that led us to this decision. Though a simple idea for people with extensive mouse skills, the metacursor was difficult for unskilled users and encouraged "gaming" of the system.

6.3.1 Original Metacursor Design

The original Metamouse collaborative design consisted of color-coded user cursors and a special metacursor which was placed at the average location of the user cursors. All user clicks were directed through this special cursor, rather than their own, when the user mice were in within some distance tolerance [8] of each other. This metaphor was too complex for most of the students we worked with. Even after an explanation of how Metamouse works, they struggled to focus on more than one cursor on the screen and missed agreement cues.

An unforeseen complication arose from a group where only one

Figure 2. The Metamouse system using no metacursor. The mice are in agreement, as signaled by their green outline.

student moved her mouse at a time. In this case, the metacursor seems to be a normal cursor, moving in the same direction as her personal cursor, but at a slower speed. This was very confusing, as the system *seemed* to map directly to traditional single-mouse usage. Of course, the user could not click without agreement.

6.3.2 Metacursor Upon Agreement

We noted that the metacursor was of limited value when the users were not in agreement. The students couldn't click through it, so during this time it merely served to add cursor confusion. In line with this reasoning, we removed the metacursor from general game-play and only showed it when the users achieved proximity agreement.

This allowed users to recognize their own cursors, but without the gestalt experience of real-time location averaging, students did not think of the metacursor as a cursor. Instead, they viewed it as merely an indicator of agreement. Students would then try to click through their own cursors rather than the metacursor. This difference was sometimes irrelevant; if a button was large, most students did not realize their error. However, some games are designed with very small targets—on the order of 10px—and many students lacked the precision mouse control to click these targets, even without the additional clutter of the metacursor.

6.3.3 Removing the Metacursor

We next noted that, if the users were just using the metacursor for signaling, a more direct signaling method might be more effective. Towards this end, we removed the metacursor entirely and designed new user cursors with a thick green outline that appeared when the cursors were in agreement. This improved student performance, but revealed that even without the metacursor to add confusion, many of the students did not grasp that clicks originate at the "point" of the cursor. A student might have their entire cursor on a button, except for the tip, while other cursors in proximity agreement might be off of the button entirely. In this case, repeated clicking and slight mouse movements would eventually lead to one student successfully clicking the button. This gives the impression of a non-deterministic system where some clicks mysteriously lead to actions and others do not, a perception which will further hinder the development of expert mouse skills. Figure 2 demonstrates the Metamouse system with no metacursor.

We ultimately chose to keep the standard cursor shape in the

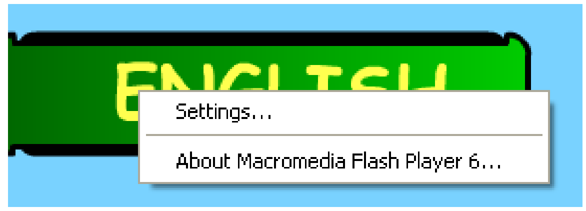

Figure 3. The contextual menu brought up when a student accidentally right clicks in an APF game

hopes that students will eventually pick up on the fact that clicks originate at the point of the arrow. This was one of many design decisions where we chose to support industry standard UI conventions over what might be strictly easiest for our users to understand.

6.4 Third Iteration: Dealing with Low Computer Literacy

There were a wide variety of issues we discovered when dealing with non-proficient users. These problems include right clicks, machine-gun clicking, and accidental drag-and-drop. In this section we will primarily discuss our solutions to these common user behaviors, but we also discuss the design challenges in implementing collaborative drag and drop—a task which exposed many user behavior issues.

6.4.1 Right Clicks

Differentiating between the various buttons on a PC mouse is a common novice error. While designers in the developed world may expect children to be familiar with the distinction between buttons by early grammar school age [12], right clicks were quite common amongst our students in grades five through seven. In fact, we saw a few kids clicking on the rotating center button instead of either.

In APF's educational flash games, right clicking leads to a contextual menu that did not make sense to the children and blocked content. Figure 3 demonstrates this menu. Once a contextual menu has been brought up, it freezes the game until you left click away from the menu. In Metamouse, students had the additional complication of requiring agreement before users were able to left-click. Upon accidentally right clicking, students were often unable to sort this problem out and had to call to the instructor.

To remedy this, Metamouse intercepts and ignores all right clicks. This is a configurable option, as it is possible that other games may make use of the right click.

6.4.2 Multiple Simultaneous Clicks

When three kids are sharing three mice, it is quite common that all three will click on a button at approximately the same time. Though this is often not a problem (most button functions are idempotent) there are occasions where it is crippling. One example are "skip" buttons which allow you to skip content. Three clicks skips three pieces of content, rather than one, as desired. As another example, some games immediately place new questions where old ones were, and the following clicks accidentally provide incorrect answers.

To solve this, we created a click timer which drops all clicks following the first click for some amount of time. This lets all kids click on the content, while the application receives just one click event. This remedies the problem in most cases. However, if the initial clicker is off the content (as in the small button example earlier) the users who *were* on the content would be prevented from clicking on the correct target. To limit this problem, we now only drop each user's first click. This allows them to correct an error if by clicking a target multiple times. Though we still occasionally see unintentional double clicks, this has reduced the problem to a manageable level.

6.4.3 Mouse Drifting and Long Clicks

Students with limited mouse skills had a tendency to "drift" with their cursors. When selecting a target, or trying to stay in an area, the student would fail to hold the mouse steady. Drifting appeared to be primarily the result of low motor control or limited familiarity with mice.

These students would also hold clicks (the mouse trigger) down for longer than is necessary. Long clicks sometimes seemed to be purposeful, as if the user intended to nudge the computer more forcefully in hopes of a more successful outcome. Longer clicks, too, may be explained by a lack of fine motor skill required to click rapidly.

Long clicks in combination with drifting or nudging sometimes resulted in accidental dragging. Students would hold the click longer than necessary and then drift away from the initial target, instigating a drag from the perspective of the system. This has a negative impact on collaboration, as well as game-play, so we added a number of heuristics that would lessen the impact of accidental dragging. We noted that while standard user clicks were often of a longer duration than those of an expert user, students' clicks were longer still when they were attempting to drag or pick up an item. We were able to raise the minimum threshold of time between down-click and up-click to initiate a drag, and we found no adverse effects in usage. Purposeful drags also moved a much larger average distance than accidental ones. We also raised the distance threshold for initiating a drag. These two changes substantially decreased the dragging associated with long clicks and drifting.

6.4.4 Mitigating Accidental Drag-and-drop

In our original drag-and-drop model, the system cursor was always assigned to the metacursor, meaning that dragged items were not "owned" by any individual. After we removed the metacursor, we assigned dragged items to the user cursor who first initiated the down-click. Alternative assignments (such as the metacursor model) had proven too confusing, but our solution requires that the original user who instantiated the drag must also end it. Even with the heuristics to mediate long clicks and drifting, students still sometimes initiated drags accidentally, and then found that they could not drop the item without achieving agreement first. Essentially, users could accidentally pick an item but had to coordinate to put it back down. This led to an untenable situation, as unskilled users were more likely to freeze or become flustered when unexpected actions occurred.

We certainly could have diffused this situation entirely by not requiring agreement to drop an item, and simply letting the user who initiated the drag complete it on her own. However, in games that use drag and drop as a primary decision making action, it is often the dropping portion where the actual decision is made. We felt that not requiring collaboration at this point would undermine the collaborative intent of Metamouse.

Instead, we chose to add a feature which reduces the conflict of accidental drags. We were able to do this by taking advantage of some regularities in drag and drop game-play to reinforce desirable user behavior. As mentioned before, most successful drags are over a considerable distance, and last for much longer than any clicks. Additionally, when we set the heuristics such that being below either threshold negated a drag (versus requiring both) students began "putting items back" after dragging accidentally. They would recognize that the item was stuck to their cursor, and return it to the location it was originally. They were able to accomplish this even after large spans of time, and without agreement. Moving the item to a new location still required agreement. This proved to be very helpful in diffusing the frustration of failed collaborative dragging attempts.

6.4.5 Complications of click-move-click

Users frequently made incomplete dragging gestures–clicking and dragging to pick up an item, and letting it go as soon as it has become associated with the cursor. This partial dragging may be caused by fine motor control issues. However, this behavior also seems to be related to ignorance of the standard dragging model and possibly a learned behavior. Some of the applications that students used featured a "click-move-click" method that attached an item to your cursor without requiring the user to hold down a mouse button. This procedure is often used in education games with the assumption that students will make fewer errors than they will with drag and drop. Evidence does not seem to support this hypothesis [3]. Given its ubiquity in the games we sampled, we feel that the "click-move-click" procedure may have influenced some students to expect that they do not need to hold down the mouse button while moving items.

In order to require collaboration for drag and drop options, Metamouse must require agreement for dropping in addition to picking up. Unfortunately this leads to perpetuating non-standard user behavior with regards to partial drags. Metamouse does not transmit an up-click to the system if it takes place out of agreement, so if a user begins a drag and up-clicks before achieving agreement, the item will remain attached to the cursor, as in "click-move-click". The user must then click again with agreement and the resulting up-click will end the drag. We witnessed students clicking needlessly at the end of drags, even if they successfully dragged the item and released with agreement. It is unclear if Metamouse created this behavior or if it is a permutation of the student confusion around "click-move-click".

7. RESULTS AND DISCUSSION

In this section, we first discuss open issues, related specifically to collaborative systems with multiple mice, that we feel were not resolved by our design process. We then follow with observations and conclusions on how to design shared educational games, for use with one or multiple mice.

7.1 Open Issues

7.1.1 Dragging Errors

As discussed in Section 6.4, we encountered several behavioral artifacts surrounding drag and drop. Users who picked up an item in a game may release their click before reaching agreement. As our design requires that the users agree before dropping an item, we ignore that up-click, leading to a situation where the user has to click down and then up (while in agreement) to release an item. This is not a user behavior we feel contributes positively to learning mouse control and standard application behavior.

It is unclear to us how to resolve the confusion around drag and drop. We observed students extraneously clicking when dropping an item they had successfully dragged, and this seems to be clearly related to our flawed design. Acquired behaviors like this one will not map well to a single-mouse model. Within the intended context of Metamouse, these behaviors may be preferable to dropping the item without agreement, leading to less discussion, or dropping the item when agreement is reached but without any direct user action. In this case, users would accidentally drop whenever their cursors hovered too close together.

This issue appears to be unresolvable given our constraint of supporting legacy applications. We hope that students will be able to adapt to the two environments. However, this behavioral eccentricity is an excellent example of the interface problems designers must consider in designing user interfaces for children in developing regions.

7.1.2 Filtering extraneous clicks

Metamouse drops all right clicks and filters out repeated clicking because our user population is using only simple single-click flash games. This property is configurable. If we were sharing content that required the double or right clicks, this feature would be disabled.

Much of the benefit of these clicking heuristics are seen in the initial stages of deploying Metamouse, when students first encounter computers, mice, and Metamouse, and are struggling through these initial learning curves. However, given the consistently low level of mouse proficiency we encountered, it is likely that most new deployments of educational software in the developing world will need to take similar precautions to make widespread adoption feasible. We are currently undertaking a longitudinal study of Metamouse. It is possible that, in the long run, these heuristics may become unnecessary or even damaging. We do not expect any of our current heuristics to be damaging, given that we have observed students using Metamouse over several weeks. However, overly aggressive compensation for poor mouse skills could clearly hamper the acquisition of traditional mouse skills. This remains an important question to us, and one we continue to focus on.

7.2 Designing for Sharing Using Multiple Mice

In this section, we describe some simple design lessons for producing multi-user educational games.

7.2.1 Employ Competition Wisely

With the constraint of working with existing games, race-clicking [23] crippled our design. Games often moved forward even on wrong answers and did not provide explanations when answers were correct. Because of this, one aggressive user can dominate the gameplay even without superior knowledge of the material. Correctly incentivizing users to *learn* rather than just *play* is very difficult. Designers of competitive games should be careful to ensure that their interfaces encourage learning and positive discussion amongst users.

7.2.2 Use Simple Models of Interaction

The original metacursor design [8] is an example of a complex usage model which was ultimately abandoned. However, we witnessed students struggling with simpler, more standard interactions such as drag and drop, differentiating system cursors, and clicking through the point (rather than the center) of the cursor. While a clear understanding of these user interface elements is part of baseline computer literacy in the developed world, young computer learners in developing countries have less experience with user interface norms, and naturally may take longer to appreciate their subtleties. Designers should strive to use a small set of simple interaction paradigms (point and click, drag and drop), even at the expense of content.

In addition to the wide disparities in computer literacy amongst students, we have also found that instructors in computing classes sometimes have little practical computer experience. Even in schools with a designated computer lab instructor, the instructor's computer literacy is often not what technology designers might consider standard. Because of this and general under-staffing problems, it is likely that students will not always have teacher intervention to fall back on when they struggle. This set of constraints necessitates simple designs and even simpler error messages. Designers should assume their users are "on an island" with the software, operating with absolutely no external feedback or instruction.

7.2.3 Work With Any Pedagogy

Some education researchers [26] have suggested instituting collaboration policies in the classroom. While this can certainly be helpful, our experiences have shown that such policies can not be reliably implemented in the field. APF, the Indian educational organization we worked with, had many recommendations for teachers to best organize their students to maximize learning and engagement. However, outside designers cannot consider all of the factors in the basic organization of the teacher's day, and teachers often disregard their suggestions. Only one of the five schools we looked at while planning this study used APF's suggested "monitor" model, where one strong student in the group enforces sharing. Designers should not assume that any complicated sharing strategies will be enforced. Students will steal mice, sit with peers of varying backgrounds, and ignore on-screen instructions. Educational software must work when dropped into any classroom, regardless of the class structure.

7.2.4 Impart Transferable Mouse Skills

Some of the most generalizable findings from the design study include the unexpectedly thorough, widespread issues with mouse proficiency. In the two schools that participated in the user study, one quarter of the students (9 out of 36 grade 5 students) [10] did not know what to do with a mouse when seated alone at a computer. Computing skills are highly valued for the employment possibilities they can create [21]. Learning to use a mouse is a fundamental requirement for computer literacy. However, most games are focused on teaching a lesson, assuming mouse competence from the users. We believe that designers should strive to teach not just content, but also basic mouse skills that are of use outside of just their specific application.

7.2.5 Impart Mouse Skills on All Users

Many of the most proficient users in our study were unaware of the subtleties of mouse actions in a traditional single mouse setup. This was in schools where computer learning was *required* and students had been playing games for *years*. While most studies examining mouse proficiency in children in developed areas see behavioral issues improve or resolved between the ages of four and five [4, 12, 14], our students still seemed to be struggling at ages 12 to 14.

The widespread mouse proficiency issues observed in our study clearly indicate that single mouse sharing is not sufficient to establish computer literacy with any kind of equanimity. Habitual sharing of computers can serve to broaden the gap in mouse skills, rather than reduce it. Even with multiple mice, students from feeder schools who have no computer experience may be shunted into the role of the slow user, and thus may not have many opportunities to improve their mouse skills. To mitigate this, designers should provide interfaces that allow for participation of all users, all of the time. Students can learn basic mouse skills by simply manipulating the mouse, irrespective of the actions taken in the game. Our own results support this [10], with students who were not required to participate still improving mouse proficiency.

7.3 Designing Single-Player Games for Sharing

Many of our findings inform general game design in developing regions. The user needs in these areas differ from those of the developed world in many ways. In this section, we focus on just one: omnipresent sharing.

Few designers of single player applications take multi-user sharing into mind when designing their games. In games intended for use in low-income, developing regions, sharing is by far the most common method of use. Clearly, designing multi-player games for multiple mice is an excellent option to pursue, but there is considerable value in improving the design of single-player games; making them more sharable with just a single mouse. We suggest three basic ideas that allow for better sharing of single-player games.

7.3.1 Avoid Timers

Quite a few educational games have timed sections: areas where the user has to complete a task within a certain amount of time. This has a basis in traditional educational pedagogy; educators commonly wish to prevent students from looking up answers or train students to respond quickly. However, when groups of people are involved, tasks simply take longer. We witnessed groups of students continually disappointed after performing admirably on a task only to "lose" when an inappropriately brief timer went off. In many cases, poor mouse skills played into these "losses" as well.

Additionally, timed games create negative group interactions. Students are more tense and less gracious of one another. When a strong user realizes that the game is timed, they will often assume complete control of the game, providing less proficient users with no chance to participate. Even if students continue trying to collaborate, discussion nearly ceases, which is an unfortunate loss of a good avenue for learning [33].

Designers should avoid timers in general. There is no way to know how many users are in front of their game, and thus the appropriate timer length.

7.3.2 Mouse-Centric Designs

Using the keyboard as a primary controller of game-play was common in the early days of computer gaming. While arrows may occasionally be an appropriate control for a game, many modern designs use mouse gestures or on-screen arrows to achieve an equally intuitive interface. We found that most of the time the keyboard is a distraction and a hindrance for multiple users. Keyboards are difficult to share, and taking turns is basically the only option. Some games alternate between keyboard and mouse inputs, providing more opportunity for users to quibble when the input method changes. Single keyboard sharing has many of the same failings as single mouse sharing: one dominant user tends to control the input.

Designers should stick to just one type of input idiom per game. Switching between them is a constant source of strife, providing more opportunity for dominant users to take control.

Projects have explored providing multiple keyboards [7], which we believe to be a promising research direction. We hope that a system comparable to Metamouse, utilizing multiple keyboards in existing games, could be developed in the future.

7.3.3 Flexible Navigation

When sharing a game or working with students who have low mouse proficiency, inflexible navigation is another factor that can frustrate users. Some of the games we encountered only allowed students to move through games sequentially. Students would inevitably encounter a task that was unplayable as a shared game, or which was simply far above the mouse proficiency of the players. In this case, they would be forced to complete the task in order to make progress. Often students replayed individual games, but with a different user controlling the mouse. This was impossible if the game did not provide a "restart" or "back" option. The best game designs we encountered featured a website-like model with a "homepage" and games which can be played or replayed in any order. Designers should try to design in a flexible way, allowing for students to quickly gain access to exactly the content they seek.

8. CONCLUSION

In this study, we described the design process for *Metamouse*, a system for using multiple mice with existing single-player games on one computer. During this study, we observed a number of facts relevant to the design of shared educational (both single- and multi-mouse) interfaces in the developing world.

Mouse proficiency was widely variable and on average considerably lower than what technology designers might expect given the literature on mouse proficiency in very young children. This necessitated special consideration in designing the basic interaction protocols for collaborative mouse movement. In implementing these procedures, we found that simple usage models are best, and that complex metaphors often build off of expert knowledge of the existing mouse paradigm that young users may not have yet developed.

Many of our design study findings generalize to shared single mouse environments in the developing world. We recommend several basic design standards for educational applications, including flexible navigation paired with games that feature simple, mouse-centric movement, and which avoid features that will create tension or strife when shared. Given that sharing computers is the norm in many developing regions, we urge designers of educational content to embrace collaborative, sharable designs.

9. ACKNOWLEDGMENTS

The authors would like to thank the Azim Premji Foundation, particularly Sukumar Anikar, S Santhosh, and Bharathish Kumar, for their fantastic assistance in the field. We would also like thank Matt Kam for his help in developing some of the techniques used in this paper, Joyojeet Pal and Divya Ramachandran for their aid in earlier iterations of this work, and the teachers, students, and headmasters at the schools for their time, help, and patience. Lastly, we would like to thank Telemundo, Anuj Tewari, Kuang Chen, Alice Lin, Dori Garman, Peder Reiland, Valerie Hoagland, Denali Kerr, and innumerable other people for their time spent playing children's games and emotional support.

This work was funded in part by National Science Foundation Grant No. 0326582, a National Science Foundation Graduate Research Fellowship, the Center for Information Technology Research in the Interest of Society (CITRIS), and the Blum Center for Developing Economies.

10. REFERENCES

[1] S. Amershi, M. R. Morris, N. Moraveji, R. Balakrishnan, and K. Toyama. Multiple mouse text entry for single-display groupware. In *CSCW '10: Proceedings of the 2010 ACM*

conference on Computer supported cooperative work, pages 169–178, New York, NY, USA, 2010. ACM.

[2] L. J. Bricker, M. J. Baker, E. Fujioka, and S. L. Tanimoto. Colt: A system for developing software that supports synchronous collaborative activities. In *Proceedings of Educational Media (EdMedia)*, pages 587–592, 1999.

[3] A. Donker and P. Reitsma. Drag-and-drop errors in young children's use of the mouse. *Interacting with Computers*, 19:257–266, 2007.

[4] A. Donker and P. Reitsma. Young childrenâĂŹs ability to use a computer mouse. *Computers & Education 48*, 48:602âĂŞ617, 2007.

[5] A. Druin, G. Revelle, B. B. Bederson, J. P. Hourcade, A. Farber, J. Lee, and D. Campbell. A collaborative digital library for children: a descriptive study of children's collaborative behaviors and dialogue. Technical report, Department of Computer Science, University of Aarhus Jordan B, Henderson A, 2003.

[6] A. P. Foundation. http://www.azimpremjifoundation.org. Accessed on 3/2010.

[7] S. Garg, C. Robinson, C. Tseng, H. Underwood, R. Anderson, and J. Pal. Multimath: numeric keypads for math learning on shared personal computers. In *ICTD'09: Proceedings of the 3rd international conference on Information and communication technologies and development*, pages 492–492, Piscataway, NJ, USA, 2009. IEEE Press.

[8] K. Heimerl, D. Ramachandran, J. Pal, E. Brewer, and T. Parikh. Metamouse: multiple mice for legacy applications. In *CHI EA '09: Proceedings of the 27th international conference extended abstracts on Human factors in computing systems*, pages 3853–3858, New York, NY, USA, 2009. ACM.

[9] K. Heimerl, D. Ramachandran, J. Pal, E. Brewer, and T. Parikh. Metamouse: Multiple mice for legacy applications. In *Information and Communication Technologies and Development (ICTD)*, pages 490 –490, April 2009.

[10] K. Heimerl, J. Vasudev, K. Buchanan, E. Brewer, and T. Parikh. Metamouse: Improving multi-user sharing of existing educational applications. In *IEEE/ACM International Conference on Information and Communication Technologies for Development*, 2010.

[11] J. P. Hourcade. Interaction design and children. *Foundations and Trends in Human-Computer Interaction*, 1:277–392, 2008.

[12] J. P. Hourcade, B. B. Bederson, and A. Druin. Preschool children's use of mouse buttons. In *CHI '04: CHI '04 extended abstracts on Human factors in computing systems*, pages 1411–1412, New York, NY, USA, 2004. ACM.

[13] J. P. Hourcade, B. B. Bederson, A. Druin, and F. Guimbretière. Differences in pointing task performance between preschool children and adults using mice. *ACM Trans. Comput.-Hum. Interact.*, 11(4):357–386, 2004.

[14] J. P. Hourcade, M. Crowther, and L. Hunt. Does mouse size affect study and evaluation results?: a study comparing preschool children's performance with small and regular-sized mice. In *IDC '07: Proceedings of the 6th international conference on Interaction design and children*, pages 109–116, New York, NY, USA, 2007. ACM.

[15] M. Kam, A. Mathur, A. Kumar, and J. Canny. Designing digital games for rural children: a study of traditional village games in india. In *CHI '09: Proceedings of the 27th international conference on Human factors in computing systems*, pages 31–40, New York, NY, USA, 2009. ACM.

[16] Y. Lou, P. C. Abrami, and S. d'Apollonia. Small group and individual learning with technology: a meta-analysis. *Review of Educational Research*, 71(3):449–521, 2001.

[17] E. Mazur. *Peer Instruction: A User's Manual*. Series in Educational Innovation. Prentice Hall, Upper Saddle River, 1997.

[18] A. Moed, O. Otto, J. Pal, U. P. Singh, M. Kam, and K. Toyama. Reducing dominance in multiple-mouse learning activities. In *CSCL'09: Proceedings of the 9th international conference on Computer supported collaborative learning*, pages 360–364. International Society of the Learning Sciences, 2009.

[19] N. Moraveji, K. Inkpen, E. Cutrell, and R. Balakrishnan. A mischief of mice: examining children's performance in single display groupware systems with 1 to 32 mice. In *CHI '09: Proceedings of the 27th international conference on Human factors in computing systems*, pages 2157–2166, New York, NY, USA, 2009. ACM.

[20] N. Osawa. Aggregate pointers to support large group collaboration using telepointers. In *CHI '06: CHI '06 extended abstracts on Human factors in computing systems*, pages 1169–1174, New York, NY, USA, 2006. ACM.

[21] J. Pal, M. Lakshmanan, and K. Toyama. "my child will be respected": Parental perspectives on computers and education in rural india. *Information Systems Frontiers*, 11(2):129–144, 2009.

[22] J. Pal, U. S. Pawar, E. A. Brewer, and K. Toyama. The case for multi-user design for computer aided learning in developing regions. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 781–789, New York, NY, USA, 2006. ACM.

[23] U. S. Pawar, J. Pal, R. Gupta, and K. Toyama. Multiple mice for retention tasks in disadvantaged schools. In *CHI '07: Proceedings of SIGCHI conference on Human factors in computing systems*. ACM, 2007.

[24] U. S. Pawar, J. Pal, and K. Toyama. Multiple mice for computers in education in developing countries. In *IEEE/ACM International Conference on Information and Communication Technologies for Development*, 2006.

[25] S. D. Scott, R. L. M, K. M. Inkpen, and E. Lab. Understanding children's interactions in synchronous shared environments. In *Proceedings of Computer Supported Cooperative Learning*, pages 333–341, 2002.

[26] R. E. Slavin. Cooperative learning. *Review of Educational Research*, 50(2):315–342, June 1980.

[27] R. E. Slavin. Research on cooperative learning and achievement: What we know, what we need to know. *Contemporary Educational Psychology*, 21:43–69, 1996.

[28] D. Stanton, H. Neale, V. Bayon, and N. Rd. Interfaces to support children's co-present collaboration: Multiple mice and tangible technologies. In *PROC. CSCL*, pages 342–351. ACM Press, 2002.

[29] J. Stewart, B. B. Bederson, and A. Druin. Single display groupware: a model for co-present collaboration. In *CHI '99: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 286–293, New York, NY, USA, 1999. ACM.

[30] F. Tian, F. Lv, J. Wang, H. Wang, W. Luo, M. Kam, V. Setlur, G. Dai, and J. Canny. Let's play chinese characters: mobile learning approaches via culturally inspired group games. In

CHI '10: Proceedings of the 28th international conference on Human factors in computing systems, pages 1603–1612, New York, NY, USA, 2010. ACM.

[31] E. Tse and S. Greenberg. Rapidly prototyping single display groupware through the sdgtoolkit. In *Proceedings of the Fifth Australasian User Interface Conference*, volume 28, 2004.

[32] WonderWurks. Teamplayer. http://www.wunderworks.com/education/. Accessed on 3/2010.

[33] D. Wood and C. O'Malley. Collaborative learning between peers. In *Educational Psychology in Practice*, volume 11, pages 4–9, 1996.