Metamouse: Improving Multi-user Sharing of Existing Educational Applications

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*—Many children, especially in the developing world, must share a computer at school. Often, more advanced or aggressive students dominate, leaving others frustrated and disengaged. One promising approach is to provide each student with their own input device, usually a mouse, while sharing a common computer and display. Previous multiple mouse sharing efforts for education have relied on developing custom applications that encourage collaboration — for example, by requiring that all users click on an icon or option to proceed. Implementing this requires access to application source code, which is unlikely to be available, and would require significant engineering effort to adapt even if it was. To address these limitations, we developed *Metamouse*. Metamouse only conveys clicks to the application when users have already agreed on a screen location, within a pre-defined tolerance (a novel sharing technique that we call "location-voting") and requires no access to the application source code. We have implemented two versions of location-voting — one that requires all users to agree ("Consensus"), and another where only a majority is required ("Majority"). To evaluate Metamouse, we conducted a user study with 24 fifth-grade students in a low-income school in Bangalore, India. Our results demonstrate that Metamouse is intuitive, usable and has the potential to outperform other sharing strategies in terms of user engagement, generating discussion, and overall satisfaction. We also show that the Majority approach provides all of these benefits as effectively as Consensus with less frustration for faster users and less embarrassment for slower ones.**

***Index Terms*—Education, Developing Regions, Single Display Groupware, Shared Computers**

I. Introduction

Many children, especially in the developing world, must share a computer while at school [1]. Often, more advanced or aggressive students dominate, leaving others frustrated and disengaged. One approach that has appeared effective is providing each student with their own mouse while sharing a computer and display [2]. Previous research has shown that these techniques are intuitive and usable [3], increase learning [4], and help focus and motivate users [5].

Despite these encouraging results, several barriers remain for deploying such techniques in real-world educational settings. First and foremost is the lack of appropriate content. Moreover, encouraging collaboration is a difficult challenge in the presence of multiple (excited) kids, each with their own mouse. Previous efforts have relied on custom applications that enforce collaboration — for example, by requiring that all users click on an icon or option to proceed (something we call "click voting") [4], [6]). Implementing this approach requires access to source code including the underlying application-specific widgets. These are unlikely to be available for existing educational content, and would require significant engineering effort even if source code was available.

Supporting *existing* applications would open up a huge library of high-quality content, much of which is integrated into existing lesson plans. However, prior efforts for supporting multiple mice with legacy applications have relied on basic interaction strategies that do not enforce any particular collaboration requirements, and could even impede learning. In one model, each mouse pointer acts autonomously and all user clicks are passed through to the underlying application [7], [8]. When testing this method, researchers found that it encouraged students to race to click first (we call this approach "click first") [4]. Another strategy disables all of the individual users' pointers, and places a new pointer at their average location ("location averaging"). In our own early trials we found that this confused students, making it seem that the mouse was under *no one's* control.

To address these limitations, we have developed *Metamouse*. Metamouse only conveys clicks to the application when users have already agreed on a location, within a predefined tolerance ("location-voting"). We have implemented two versions of location-voting — one that requires all users to agree ("Consensus"), and another where just a majority is required ("Majority"). Metamouse also subtly redefines the semantics of single-user interaction primitives such as point-and-click, drag-and-drop and mouse-over to effectively support multiple mice. To evaluate Metamouse, we conducted a user study with 24 fifth-grade students in a low-income school in Bangalore, India. Our results demonstrate that Metamouse is intuitive, usable and has the potential to outperform other sharing strategies in terms of user engagement, generating discussion and overall satisfaction. We also show that the Majority approach provides all of the benefits of Consensus, with less frustration for faster users, and less embarrassment for slower ones.

The specific contributions of this paper are:

- A novel "location-voting" technique for sharing multiple mice with legacy applications.
- An evaluation demonstrating that Metamouse is intuitive, usable and outperforms simpler sharing strategies.
- A comparison of Consensus versus Majority agreement for location-voting; demonstrating that the latter obtains all of the benefits of the former, with reduced frustration and embarrassment. To our knowledge, this is the first system to implement and evaluate a multiple mouse sharing strategy where only a *subset* of users are required to agree.

The rest of this paper is organized as follows. We begin with a discussion of related work, and our motivations and goals for this research. We then describe Metamouse and the location-voting approach that we implemented. Next, we evaluate Metamouse, including the Consensus and Majority versions, by comparing them with each other as well as with traditional sharing of a single mouse. We conclude with a discussion of these results, some areas for future work and a summary of the main results of the paper.

II. Related Work

Single-display groupware (SDG) is a general term for collaborative, multi-user applications using a single computer and display [2]. There have been several SDG applications for education [9], [10]. Inkpen et al. [11] is an early example studying the educational benefits of collaboration in SDG applications. Prior research in education also supports the assertion that collaboration between students can improve learning and child development [12], [13]. Stewart at al. [14] found that the users collaborated more when sharing using multiple mice.

More recently, there has been significant interest in exploring the benefits of using multiple mice for sharing computers in the developing world. Preliminary results indicate that, with appropriate content, sharing multiple mice can be intuitive and usable [3], increase learning gains [4], help focus and motivate users [5], and scale to literally dozens of users [15]. Most of these results were obtained using software that was specifically designed for multiple users by enforcing very specific collaboration requirements, such as requiring that every user must click on an answer or option to select it (we call this approach "click voting") [4], [6]. Researchers have discussed how it is difficult to develop custom SDG applications, and that this is one of the primary reasons that this approach has not been widely adopted [16], [17]. This is particularly problematic for our context, given the lack of experienced educational content developers in the developing world.

Attempts to use multiple mice with legacy applications (educational or otherwise) have typically modified the input driver to support multiple mice by capturing each mouse's individual input and issuing combined messages to the application software [7], [18]. The "click-voting" approach is impossible to implement at this level, as the input driver has no knowledge of application buttons and other widgets. As a result, most of these tools have implemented either very rudimentary or no sharing strategies. The simplest is allowing each mouse to have independent control, while passing all user clicks and other actions through to the application [7], [8], [18], [19], [20]. Not only is this approach confusing, but it leads to competition among students to see who can "click first'. In early Multimouse work [4] they found that this competitive model negatively impacts learning for boys, actually performing worse than the sharing a single mouse. In our own early studies of this model, we found that students quickly became frustrated and paid little attention to the application content. An alternative is aggregating the individual mouse positions, and locating the system cursor at the average location ("location averaging") [21], [22], [23]. In our studies we found that users quickly became confused with this; as the pointer corresponded to no one user's movements, they felt that nobody was controlling it.

Several researchers have compared various sharing approaches. Druin et al. [6] compared a "click-first" to a "click-voting" model, using the same educational content with each. They found that "click first" resulted in more discussion about the application content, while "click voting" increased discussion about user interface elements. Multimouse [4], [24] compared these same two models in India, obtaining a very different result. They found that "click-voting" was better for encouraging discussion and learning, but that "click-first" frequently devolved into students racing to be the first to click. Students (especially the boys) often viewed participation in the game as being more important than actual learning, which led to increased clicking, even if students did not know (or even read) the answer.

III. Motivation and Goals

We summarize Metamouse's main design goals in this section.

A. Backward Compatibility

There are lots of high-quality educational software applications designed for single users, and more are developed every year. Moreover, teachers have already been trained and have experience with these applications, which means they are well-integrated into lesson plans and curricula. Our goal is to make these titles more accessible to children working in shared computing settings.

B. Encourage Collaboration and Engagement

Collaboration has been shown to improve learning outcomes [25], [26]. Supporting effective collaboration and group problem solving is even more important in public schools in the developing world, where teachers are often absent and/or unfamiliar with computers. On the other hand, children become frustrated or disengaged if they are ignored and/or actively excluded from the interaction. In general, competition and fighting degrades the educational experience for everyone.

C. Easy to Learn

In our work with schools in India, we have found that even instructors in computing classes sometimes have little practical computer experience. Wide disparities in prior computing experience are also common among the students. Some may have attended primary schools where computers were commonly used, or may even have a computer at home; while others may not have even touched a computer before. We would like to ensure that novice teachers and students can use the system with little to no external instruction, ideally without frustrating or limiting more advanced students.

D. No New Teaching Policies

In some schools, proactive teachers have instituted formal policies for sharing a single mouse. For example, some teachers we met required that the weakest student sit in the middle, or that a "monitor" be appointed from the students to enforce a particular sharing strategy. Some education researchers [27] have suggested using comparable techniques to encourage sharing inside of competitive or collaborative SDG models. For instance, requiring the students discuss each question after completion. Unfortunately, we observed that these guidelines were not always followed. In general, it would be extremely difficult to institute a consistent set of sharing or seating policies across schools. As a result, our system should work in as many classroom contexts as possible, regardless of how the teacher chooses to group or seat students.

E. Impart Transferable Mouse Skills

As mentioned, students in the developing world often have widely varying prior computer experience. On the other hand, computing skills are highly valued for the employment possibilities they can create [28]. Some 5th grade students we met did not even possess basic mouse skills. Other students were unwilling to share the mouse with them, as they were prone to make a mistake or to proceed slowly. Even if a weaker student was able to gain control of the mouse (perhaps through teacher or monitor intervention), they were quickly harassed or coerced into giving up control. As a result, less experienced children had little opportunity to improve their mouse proficiency. Learning to use a mouse is a fundamental requirement for computer literacy. We would like to ensure that the skills students learn while using Metamouse transfer to a normal single-mouse setting.

IV. Metamouse

To achieve these goals, we developed *Metamouse*, a system for sharing legacy single-player educational content between multiple children using multiple mice with a single computer and display. In this section, we describe the design and implementation of Metamouse.

Fig. 1. The users (Blue, White and Yellow) have not agreed. Each cursor has a distinct color.

Fig. 2. User agreement is signaled by outlining the agreeing mice in green.

A. Location-Voting

At the core of Metamouse is the novel "location-voting" sharing strategy. Specifically, we require that users agree on a location before any clicks are passed on to the application. None of the individual cursors are "active" until they are within some pre-defined distance from the other cursors. After some testing, we set this tolerance to be 60 pixels. When all cursors meet this requirement, we say that the cursors are in *agreement*.

Each of the cursors is assigned a distinct color at the outset. When cursors agree, they gain a green outline, providing immediate feedback to the user (see Figure 2). If any one user clicks while green, the click is passed through to the application and the appropriate action is taken (note that not all users must click as in click-voting). All clicks are ignored until this agreement is achieved. It turns out that this can be done with knowledge only of the individual cursor locations. This means that "location-voting" can be implemented without any knowledge of application widgets or state.

In early testing we found that location-voting was intuitive and efficient for performing most basic point-and-click interactions. However, we found that it required some subtle modifications to effectively support drag-and-drop and mouse-over actions.

1) Drag-and-drop: During our early testing we found that it was difficult for users to remain in agreement throughout a drag-and-drop action, especially when there was variation in mouse proficiency. We considered placing the dragged item at the average cursor location throughout the drag, similar to "location-averaging" [23]. However, while testing this approach, we again found that users thought of the object as being under no one's control, as they were not able to relate the movement of the object to the movement of their own mouse and cursor.

To address this, we associate the object with the user whose click initiated the selection. Note that users must still agree on a location before the click is passed through. However, once the selection is made, the object is carried exclusively by the initiating user. Users must agree again when dropping the object. Sometimes agreement is not reached by the time the lead user releases the mouse. If this happens, we ignore the up-click and wait for another down-click by the lead user while oulined in green (meaning the other mice are in agreement). In the meantime, the object remains associated with the original user, and moves wherever their pointer moves. During our testing we found that users understood this metaphor quickly, some with no instruction at all.

2) Mouse-overs: There were a subset of games that we tested in which mouse-over information was critical, such as one game which involved searching an image for relevant content. In this game, application items were very small, and mouse-over information was used to highlight this potentially clickable content. However, only one mouse-over callback can be active for an application at any given time. Thus, all of the cursors could not trigger simultaneous mouse-overs at different locations. To address this, mouse-over events were triggered using location averaging [22], by placing the system cursor at the average user cursor location, but *only* when they were in agreement.

B. Consensus vs. Majority

Prior systems that have implemented multiple mouse sharing techniques (for example, "click-voting") have generally required *all* users to agree. However, children have often seemed frustrated by this requirement [6]. In our own early testing, we found that faster students often became frustrated with slower users who were holding up the game, and that slower users were embarrassed or discouraged because of this.

To address this, we implemented the following two variants of Metamouse:

- Consensus: In this scheme, *all* of the users must agree on location. Figure 2 illustrates consensus agreement.
- Majority: In this scheme, only a *majority* of users must agree. For instance, if there are 3 users, 2 must agree on the location. Figure 3 shows agreement by a majority. To our knowledge, this is the first system to implement a multiple mouse sharing strategy where only a *subset* of users are required to agree.

The key distinction is whether users have to wait for everyone to proceed. Its possible that in the Majority condition, faster users will simply ignore slower ones and proceed at their own pace. On the other hand, if slower users are still found to be engaged and active, it would indicate that the same benefits as Consensus could be achieved with reduced frustration for faster users.

Fig. 3. In Majority mode, only a majority of users (Blue and Yellow in this case) must converge to make progress.

C. Implementation

We used Microsoft's Raw Input Device libraries [29] and *DLL Injection* [30] to add multiple mouse support to existing applications. The Metamouse DLL is inserted into each running process's address space. This DLL intercepts all incoming mouse events and redraw messages, allowing our code to draw the user cursors, and decide which events to pass on to the application.

V. EVALUATION

We conducted the following study to evaluate the usability and benefits of Metamouse, and to compare the Consensus and Majority variants. The study was conducted over a period of two weeks at a low-income government school near the city of Bangalore, Karnataka, India. We tested both the Consensus and Majority variants of Metamouse, and compared both of them to sharing a single mouse. In early testing we found that the "click-first" and "location-averaging" approaches were largely unusable for learning in our context, so we chose not to include them in our evaluation.

A. Context

This study was conducted in collaboration with the Azim Premji Foundation (APF) [31], a large education-focused non-profit organization headquartered in Bangalore. APF has developed over 120 educational games suitable for grades one through eight, in six different Indian languages. The foundation also provides computer hardware to selected schools.

For this study, we worked with a class of fifth grade students at a low-income government school. This school was burglarized last year, leading to a loss of most of their computing equipment. The computer equipment was eventually replaced, but the school had not yet restarted its computer classes, and as a result students had not played any games since the previous year. Students who had been at the school since before the burglary had some prior experience with the Azim Premji educational content. Students were previously organized into groups of three to four when using the computers. One student from each group was designated as the "monitor", whose role it was to make sure that the mouse was equitably shared. The monitor was selected on the basis of his or her computer literacy and mouse proficiency.

For our experiment, we selected games that tested a wide variety of interactions, and were representative of APF's overall content library. The games covered both point-and-click and drag-and-drop actions, with content such as physiology, math, social studies and health. Different selections of games were used for training and testing. The training games were "Moti's Lessons on Safety", "Visit to an Orchid", "Division", and "Trip to Village Fair". The test games were "Choo Matar Returns", "Perimeter", and "Africa".

B. Participants

Thirty-six fifth grade students were selected at random for our study. We chose a between-subjects experimental design. We organized children into twelve groups of three students each, with four groups assigned to each sharing mode: Single Mouse, Majority, or Consensus. Before the training, we tested each student's mouse proficiency by asking them to complete a simple point-and-click game with moving targets, and by dragging twenty files from one folder to another. Students who were unable to complete both tasks were judged to be non-mouse-proficient, of which there were nine total students (out of 36). These were typically new students transferring from smaller schools that did not have computers. Each group was assigned a maximum of one non-proficient student, with three assigned to each of the three testing conditions.

Unfortunately, this study was conducted during a recent swine flu scare. This, coupled with widespread absenteeism in government-run Indian schools, led to absence rates approaching 75% on specific days. We were also forced to remove data for one group from the single mouse condition. After the study concluded, we discovered that this group, which had been randomly selected, consisted entirely of students who had worked together before as a group, artificially improving their performance. Because of these reasons, the analysis in the next section only includes data for three of four groups for the Consensus and Majority conditions, and two of four groups for single-mouse, covering a total of 24 students.

We also note that our experiment covers a relatively small range of sharing arrangements. In particular, we tested only with groups of three, where at least two of the three users had some prior computer literacy. We have also informally tested Metamouse with groups of four and five, without any noticeable degradation in performance or satisfaction. However, for the final evaluation we decided to limit group size to three, as that was the number that could comfortably sit at one workstation.

C. Training and Testing Setup

We spent one full week allowing children to play a selection of APF games to become familiar with the testing scenario, and with Metamouse (if they were using Metamouse). All of the groups within a single testing condition were tested in the same lab, at the same time. Students were allowed to converse with the other groups, and did so both during the training and test sessions.

Three researchers acted as computer instructors during the testing and training. Students often asked these researchers to enforce sharing (in the single mouse condition), and for general assistance while playing the games. After finishing training, students were asked to play three different games under testing conditions. Each training and test run was fifty minutes long. The students could leave earlier if they completed the game, but could not continue after fifty minutes had expired.

D. Data Capture

We logged all mouse events — including clicks and movement, for each user. This allowed us to calculate how much each user interacted with the application. For the single-mouse condition, we placed one researcher behind each single-mouse group and recorded each transfer of the mouse. We also audio recorded all user sessions. These were translated and transcribed from Kannada by a native speaker fluent in English. We also conducted exit interviews with a majority of the students we tested.

VI. Results

Our results demonstrate that Metamouse is usable, imparts useful mouse skills, and can increase engagement and discussion across all users. We also found that the Majority variant provides all of the benefits of, and sometimes outperforms, the Consensus version, while reducing apparent user conflict and frustration. In the following section, we detail these and other findings resulting from our study.

A. User Engagement

We segmented each trial into 25-minute intervals. (For context, in the single mouse condition, users exchanged the mouse an average of once every 3.5 minutes, meaning there was plenty of time for multiple exchanges). During each interval, we categorized each user as being either primary (most mouse actions), secondary, or laggard (least mouse actions). For each of these, we calculated the total mouse actions that each user contributed during the time period, normalized as a percentage of the total actions [1]. After throwing out high and low values, we calculated the average percentage of mouse actions for the primary, secondary and laggard users in each of the three conditions.

In Figure 4 we compare the percentage of mouse movements for the primary, secondary and laggard users. Both Consensus and Majority significantly increased the relative activity of the laggard user, and reduced the relative activity of the primary user ($p<0.01$). This is not surprising; simply giving each user a mouse is bound to increase their relative participation. However, it is interesting to note that the laggard moves almost as much in the Majority model as in Consensus.

Figure 5 shows the equivalent clicking data. This is particularly notable, as Metamouse does not require all users

[1] While we acknowledge that mouse activity is an imperfect proxy for user engagement, we assert that the *relative* amount of activity is a reasonable proxy for relative engagement, especially considering that almost all groups played every game to its successful completion.

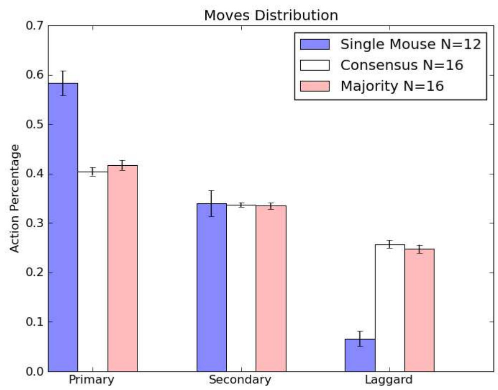

Fig. 4. Average percentage of moves for each user in each condition for each 25-minute interval, with standard error. The difference between single and both Metamouse modes is significant for the primary and laggard users ($p<0.01$).

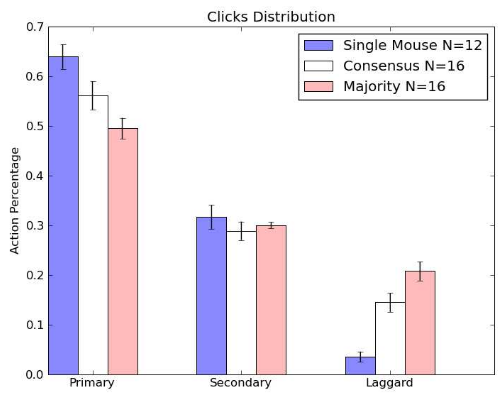

Fig. 5. Average percentage of clicks for each user in each condition for each 25-minute interval, with standard error. Both Metamouse modes led to significantly more relative activity for the laggard, when compared to the single mouse condition ($p<0.01$). Only the Majority model led to a significant reduction in the primary user's relative activity. ($p<0.01$).

to click to make progress. We again demonstrate significant increases in relative activity for the laggard user in both the Metamouse modes, when compared to sharing a single mouse ($p<0.01$). Even more surprisingly, the Majority condition leads to relatively more clicks for the laggard then in Consensus ($p<0.05$). Both Metamouse modes lead to a relative reduction in the activity of the primary user, but only the Majority difference is significant at $p<0.01$.

B. Group Discussion

Based on analysis of audio transcripts, productive group discussion also seems to have increased when using Metamouse. One reason is that Metamouse inherently requires coordination between users. Students must discuss basic mouse tasks, and then coordinate their actions to achieve desired goals. As such, an extremely complicated or unusable interface could also increase discussion, without leading to effective learning. However, in our case we found that this coordination often also involved discussions of learning content. For example, students said: *"Mmm, come to respiratory system"*, *"That was wrong. Hmm, I think its the circulatory. I'm telling you its the circulatory system. Come here one of you!"*, *"Come to the feet now and click."* and *"We have to click on the body part"*.

In fact, discussions about movement sometimes led to discussions involving content. For example, in a Majority group:

Student 1: "Come to the first one."
Student 2: "Not that one."
Student 3: "Ok come here."
Student 3: "We have to click on the head first, then hands."
Student 1: "Yeah, he is correct"

Metamouse also seemed to create a more egalitarian learning environment. Users may have felt that they had more power and were therefore more willing to speak up. As an example, all of the students in the following Consensus group asserted their opinion:

Student 1: "Click on this"
Student 3: "That is the head."
Student 3: "Who is that? Come here."
Student 2: "Not that one. Come here."

In the single mouse condition, dominance by one or two users often lead others to become distracted, upset or disengaged. In one case, a student without control of the mouse said the following:

Student: "Come here, here."
Student: "No, that didn't work"
Student: "This one. This one. Pick up this one."
Student: "You are playing all games. This is not fair."

Following this, the same student did not speak again for a few minutes, presumably because they had given up on being involved. In fact, in one complete 25-minute single-mouse run, one user said absolutely nothing at all. Even when this user was handed the mouse, the only thing he/she did is listen to advice from the others.

In contrast, we saw many instances of advanced students teaching others while using Metamouse. Students had a clear incentive to help, as they needed other students to keep up to be able to proceed. This was made explicit in this quote from a Majority group: *"Ask us when you don't understand. Don't randomly go somewhere."* Or, *"[Laggard] does not know how to play. We have to watch out for him."*

In the single mouse condition, when users attempted to influence the mouse holder, it tended to manifest as demands, pleads, or insults, rather than teaching or encouragement as we see above. For example, frustrated students in the single-mouse groups said all of the following: *"I will tell you [what to do], you [do it]."*, *"This knee. I'm telling you it's this one and you're clicking on the other one. You idiot"*, *"I'll beat you if you don't [move there]"* and *"Click, I am telling you!"*.

C. Mouse Proficiency

We again tested for mouse proficiency after the study, using the same test as described earlier. Four out of five non-proficient users in the Metamouse groups were able to complete both tasks by the end of the study[2]. Another student was unable to complete the point-and-click test, but was able to complete the drag-and-drop test. We also asked three of these users if they preferred Metamouse or playing by themselves. All of them preferred Metamouse. They felt that they were not skilled enough to operate the computer on their own, but did like having some control over the application.

Students did pick up one idiosyncrasy from Metamouse, related to drag-and-drop. Because users often had to click to drop an item after a failed up-click, they sometimes also re-clicked to drop an item when using the mouse by themselves. We are considering alternate designs for drag-and-drop that avoid this misunderstanding.

D. Frustration and Embarrassment

Four out of the six proficient students from the Consensus condition expressed some frustration with their partners during the exit interview. Moreover, two out of the three non-proficient users in the Consensus condition indicated that another student had grabbed their mouse at some point. In contrast, none of the students in the Majority condition reported any such examples of frustration.

Consensus sometimes lead to embarrassment for the laggard. For example, in the first fifteen minutes of one test there were 7 instances of the laggard being verbally pushed (i.e. *"[Laggard] come here!"*) by the other two students. As another example, the two dominant users said the following in the presence of the laggard:

> Student 1: "You have to come to where we are. See, we are done."
> Student 2: "[Laggard] has to come"
> Student 1: "She is very slow. [Other Student] was better."

In contrast, conversations with the laggard in the Majority condition were usually not so critical. Simply put, if at least two users were on the right track, they didn't have to worry about being held up by the third. If users did talk to the laggard, it was generally a more positive exchange. For example, one user said *"[Laggard], you can also join him if you want"*.

E. User Preference

In our exit interviews, Metamouse was unanimously preferred to single-mouse sharing. Students listed a variety of reasons:

- Less Fighting: Students indicated that fighting over mice took a significant amount of time away from playing. They also said that fighting was stressful. Metamouse reduced fighting by giving each student their own mouse. We saw no instances of fighting among the Majority groups, and only a few instances in Consensus (like when a dominant user would grab a laggard's mouse). Fighting was more common in the single-mouse condition.

[2]Unfortunately, all but one non-proficient user from the single-mouse condition was absent during the post-test. This student also passed both proficiency tests.

Fig. 6. Testing the Consensus Group

- Everyone Participates: Students indicated that allowing everyone to participate made them feel better. This was not only a selfish view; dominant students also indicated they felt bad about slower users not participating.
- No Monitor: All students have the same amount of power in the Metamouse condition. Students appreciated this, noting that the current practice of appointing a "monitor" creates conflict and opportunities for collusion.
- Play All Games: Some groups in the single-mouse condition implemented sharing by handing the mouse off after every game, reducing their playing time. Using Metamouse, students were excited to be able to play every game without having to wait.

Finally, though none of the users were tested with both Majority and Consensus, we described both and asked users which they would prefer. Most students preferred Consensus, saying that it better reflected the ideal of equal participation. However, a somewhat greater percentage of students from the Majority condition liked Consensus than did those who actually had experience with it, leading one to surmise that the latter group had a better sense of its limitations. The teachers that we interviewed also preferred the Majority model, saying that Consensus could cause unnecessary frustration by slowing down faster users.

VII. Future Work

In this section we discuss some areas for future work: namely, comparing Metamouse to other SDG approaches, scaling Metamouse to a larger number of users, and conducting a longer-term longitudinal evaluation of Metamouse.

A. Comparison to other Multiple Mouse Sharing Techniques

There are any number of custom sharing strategies that will work well for specific applications [21], [23]. For example, a painting game may benefit by allowing parallel drawing interactions. Some of these will obviously perform better then Metamouse for specific games. However, each new approach will incur additional overhead in terms of design iteration, programming and explanation to students. Metamouse provides a consistent, usable and easy-to-learn sharing model that can be used *across* applications, including legacy ones.

However, it would still be interesting to compare location-voting to an equivalent click-voting strategy [3], [6], [32]. It is possible that location-voting could lead to *greater* collaboration, due to the fact that users have to agree *before* any individual makes a decision by clicking.

B. Scaling to More Users

Recent work has explored scaling multiple mouse applications up to 32 mice, while using a shared classroom projector [15]. Metamouse's ability to scale to more then 3-4 users has not been tested in depth. Intuitively, it is clear that the consensus model would be difficult to scale. It is hard enough to achieve consensus among 3 children, much less 32. However, a generalization of the majority approach could fare better. Specifically, the percentage of users that are required to agree could be reduced from 50% to other values. This could even lead to situations where teams of consensus groups compete with each other to achieve tasks. For example, in our study, groups in each condition often competed to complete games the fastest.

C. Longitudinal Evaluation

We have begun a longer-term study of Metamouse at another set of four schools in the greater Bangalore area. This study will last over 6 months, and evaluate the possible learning benefits of using Metamouse. We hope to show learning gains, as well as the impact of Metamouse on sharing *outside* the computer center.

VIII. Conclusion

We have developed *Metamouse* for using multiple mice with existing single-player educational games. This is done by injecting a DLL into an already running application. Users see multiple differently colored cursors, one for each mouse. Metamouse intercepts clicks and only forwards them to the application when users agree on their location, within some tolerance. Mouse *agreement* is indicated when the respective cursors all gain green outlines. We call this technique *location-voting*, and it allows for the use of existing educational applications with multiple mice without access to the underlying application source code or widget libraries.

We developed and tested two distinct interaction strategies based on this location-voting approach. In the *Consensus* model, all users are required to agree before user clicks are processed. In *Majority*, only a majority must agree. We compared engagement and group discussion between these two models, and sharing of a single mouse, in a 5th grade class in a low-income government school in Bangalore, India. Our findings indicate that Metamouse increases engagement and discussion when compared to sharing a single mouse. Moreover, we find that the Majority model provides all of the benefits of Consensus, without frustrating more proficient users. For these reasons, we have recommended the Majority version of Metamouse to the Azim Premji Foundation for wider adoption and use.

We believe that Metamouse could have an enormous impact on education in the developing world. We are continuing to partner with APF to allow Metamouse to be used with all of their application titles, and by the tens of thousands of kids supported by the foundation. We are also searching for new partners all over the world, and hope that this will allow for more equitable use of computers by all children.

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.

References

[1] 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*. New York, NY, USA: ACM, 2006, pp. 781–789.

[2] 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*. New York, NY, USA: ACM, 1999, pp. 286–293.

[3] U. S. Pawar, J. Pal, and K. Toyama, "Multiple mice for computers in education in developing countries," in *IEEE/ACM Intl Conf. on Information and Communication Technologies for Development*, 2006.

[4] 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.

[5] 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*. ACM Press, 2002, pp. 342–351.

[6] 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 childrens collaborative behaviors and dialogue. j comput assist learn 19(2):239224 elkonin db," Department of Computer Science, University of Aarhus Jordan B, Henderson A, Tech. Rep., 2003.

[7] P. Hutterer and B. H. Thomas, "Enabling co-located ad-hoc collaboration on shared displays," in *AUIC '08: Proceedings of the ninth conference on Australasian user interface*. Darlinghurst, Australia, Australia: Australian Computer Society, Inc., 2008, pp. 43–50.
[8] G. B. Shoemaker and K. M. Inkpen, "Middesktop: An application framework for single display groupware investigations," School of Computing Science, Simon Fraser University, Report No. TR 20001-01, Tech. Rep., April 2001.
[9] 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*, 2002, pp. 333–341.
[10] E. Tse and S. Greenberg, "Rapidly prototyping single display groupware through the sdgtoolkit," in *Proceedings of the Fifth Australasian User Interface Conference*, vol. 28, 2004.
[11] K. M. Inkpen, W. ling Ho-ching, O. Kuederle, S. D. Scott, G. B. Shoemaker, and B. D. Shoemaker, ""this is fun! we're all best friends and we're all playing.": Supporting children's synchronous collaboration," 1999, pp. 252–259.
[12] K. Topping, "Cooperative learning and peer tutoring: An overview," vol. 5, 1992, pp. 151–157.
[13] D. Wood and C. O'Malley, "Collaborative learning between peers," in *Educational Psychology in Practice*, vol. 11, 1996, pp. 4–9.
[14] J. Stewart, E. M. Raybourn, B. Bederson, and A. Druin, "When two hands are better than one: enhancing collaboration using single display groupware," in *CHI '98: CHI 98 conference summary on Human factors in computing systems*. New York, NY, USA: ACM, 1998, pp. 287–288.
[15] 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*. New York, NY, USA: ACM, 2009, pp. 2157–2166.
[16] J. Grudin, "Why cscw applications fail: problems in the design and evaluation of organizational interfaces," in *CSCW '88: Proceedings of the 1988 ACM conference on Computer-supported cooperative work*. New York, NY, USA: ACM, 1988, pp. 85–93.
[17] K. L. Kraemer and J. L. King, "Computer-based systems for cooperative work and group decisionmaking: status of use and problems in development," in *CSCW '86: Proceedings of the 1986 ACM conference on Computer-supported cooperative work*. New York, NY, USA: ACM, 1986, pp. 353–375.
[18] M. Westergaard, "Supporting multiple pointing devices in microsoft windows," in *Proceedings of Microsoft Summer Workshop for Faculty and PhDs*, 2002.
[19] P. Hutterer, B. S. Close, and B. H. Thomas, "Tidl: mixed presence groupware support for legacy and custom applications," in *AUIC '06: Proceedings of the 7th Australasian User interface conference*. Darlinghurst, Australia, Australia: Australian Computer Society, Inc., 2006, pp. 117–124.
[20] WonderWurks, "Teamplayer," http://www.wunderworks.com/education/, accessed on 3/2010.
[21] 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)99*, 1999, pp. 587–592.
[22] 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*. New York, NY, USA: ACM, 2009, pp. 3853–3858.
[23] N. Osawa, "Aggregate pointers to support large group collaboration using telepointers," in *CHI '06: CHI '06 extended abstracts on Human factors in computing systems*. New York, NY, USA: ACM, 2006, pp. 1169–1174.
[24] 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*. International Society of the Learning Sciences, 2009, pp. 360–364.
[25] Y. Lou, P. C. Abrami, and S. d'Apollonia, "Small group and individual learning with technology: a meta-analysis," *Review of Educational Research*, vol. 71, no. 3, pp. 449–521, 2001.
[26] R. E. Slavin, "Research on cooperative learning and achievement: What we know, what we need to know." *Contemporary Educational Psychology*, vol. 21, pp. 43–69, 1996.
[27] R. Slavin, "Cooperative learning," *Review of Educational Research*, vol. 50, no. 2, pp. 315–342, June 1980.
[28] J. Pal, M. Lakshmanan, and K. Toyama, ""my child will be respected": Parental perspectives on computers and education in rural india," *Information Systems Frontiers*, vol. 11, no. 2, pp. 129–144, 2009.
[29] Microsoft, "Raw input device," http://msdn.microsoft.com/en-us/library/ms645536(VS.85).aspx, accessed on 3/2010.
[30] J. Shewmaker, "Analyzing dll injection," GSM Presentation, 2006.
[31] Azim Premji Foundation, http://www.azimpremjifoundation.org, accessed on 3/2010.
[32] S. Benford, B. B. Bederson, K.-P. Akesson, V. Bayon, A. Druin, P. Hansson, J. P. Hourcade, R. Ingram, H. Neale, C. O'Malley, K. T. Simsarian, D. Stanton, Y. Sundblad, and G. Taxén, "Designing storytelling technologies to encouraging collaboration between young children," in *CHI '00: Proceedings of the SIGCHI conference on Human factors in computing systems*. New York, NY, USA: ACM, 2000, pp. 556–563.