Cooperative Interaction Techniques for Graphical Objects in a Collaborative Activity[1]

Technical Report 97-04-03

Marla J. Baker, Lauren J. Bricker, Steven L. Tanimoto
Department of Computer Science and Engineering
University of Washington, Box 352350
Seattle, Washington 98195-2350 USA
{marla, bricker, tanimoto}@cs.washington.edu

Abstract

An interactive puzzle-solving program was implemented in two versions both permitting two players to cooperate in solving the puzzle. In one version, the users were permitted not only to individually pick up and place pieces into the puzzle but also to jointly pick up and position pieces with synchronous cooperative control of the piece. In the other version, the synchronous cooperative control was disabled. We present the rationale for cooperative control, the design of the puzzle activities, the results of user testing, and displays of the interaction histories for the activities. We found that although the cooperative control of pieces was more difficult than individual control, half of the users preferred the cooperative control --- a testimony to the added social value of cooperative controls in user interfaces.

[1] Supported in part by NSF Grant MDR-9155709, the Washington Technology Center, Ark Interface II, and an NSF graduate fellowship.

1 Introduction

The new group-centered culture in the modern workplace is forcing educators to take another look at team-oriented, collaborative activities. Teamwork skills are beneficial in a myriad of professional and social situations. Collaboration in a team environment encourages cooperation, communication of concepts, resolution of cognitive conflicts, and promotes problem solving and development of higher-order thinking skills.

Collaboration can also be an aid to teachers who have limited time and resources. With the current trend of increasing class sizes, it is helpful to use collaborative activities in which fellow students can take on the role of a teacher by sharing their knowledge with other students. Each student brings different knowledge and experience to an activity that can be beneficial to the team as a whole.

Another motivation for collaborative learning is that in most schools today, there are not enough state-of-the-art computers for students to all work on them individually and simultaneously. Computer-supported collaborative activities can help in this aspect since more than one student can share a single computer when working on a collaborative task.

Admittedly, collaboration can make a task more difficult than just doing the task by oneself, but the collaborative aspect may add facets of learning and enjoyment to the activity. For example, imagine two runners competing in a “three-legged race,” where the first runner has one leg bound to one of the second runner's legs. It would certainly be easier to run the race individually, but the activity has social, team-building, and entertainment value of its own.

Although collaborative activities are beginning to find their place in the classroom, the area of computer-supported collaboration is relatively new to both industrial and research communities. Thus, there is a need to investigate what types of collaborative activities can and should be supported on computers, and at a more rudimentary level, how multiple users can interact with the same objects in a computer environment. The latter is the focus of this research.

In this paper, we present some background information on sharing, previous work, and the development environment we used for building a particular collaborative activity. We then describe the collaborative puzzle activity developed in this work and the interaction techniques and history mechanism used to visualize information collected during the activity. Finally, we discuss some of the results from a user study on the puzzle activity.

2 Background

There are many ways in which multiple users can share an object. One relatively well documented type of sharing, *fine-grained sharing*, occurs when an object is divided into parts and each part is controlled by one user. For example, in Figure 1, a point is divided into its *x* and *y* coordinates, and one user manipulates the *x* coordinate while the other user manipulates the *y* coordinate (the left arrow cursor determining the *y* coordinate, and the bottom arrow determining the *x* coordinate).

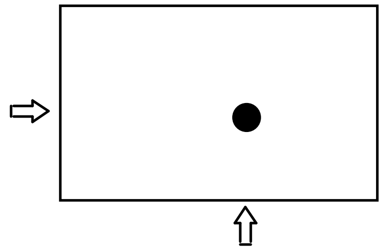

Figure 1: An example of fine-grained sharing.

A new paradigm for sharing, which we have termed *cooperative control*, occurs when multiple users manipulate an object in a simultaneous, integrated manner. For example, in Figure 2(a), we redefine the point to be the midpoint of a line segment, where each user is allowed to move an endpoint of the line segment. One of the distinguishing features of cooperative control is the complexity of the relationship between the users and the object. We might extend this interaction technique to include four users by defining the point to coincide with the intersection of two line segments, where each user can manipulate an endpoint of a line segment, as seen in Figure 2(b).

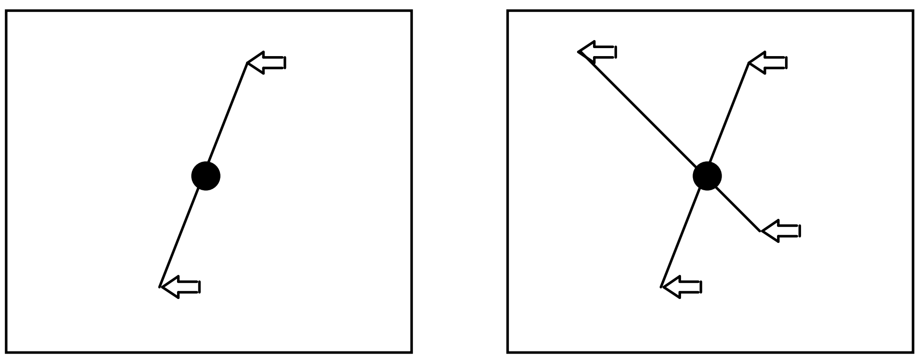

Figure 2: (a) An example of cooperative control, and (b) an extension for four users.

It would be misleading to say that there are only two types of sharing, when in reality there is a whole spectrum of sharing that consumes these two types. In the remainder of this section, we present some examples of collaborative environments and discuss the methods of sharing they use. We also describe the research on which this project is based.

2.1 Previous Work

We believe that degrees of sharing can be categorized along many dimensions. We are focusing on three dimensions including: simultaneity, complexity of control, and distance of the people interacting. The simultaneity dimension ranges from disjoint motions or manipulations to highly synchronous, perhaps even enforced, motions. The complexity of control dimension ranges from no control to very complex, integrated constraint-based control. Finally, the distance dimension is a measure of the physical distance between participants in the collaboration. The following collaborative systems demonstrate some of the variation that exists in sharing methods.

CSILE [Scardamalia94] is a collaborative system designed to support knowledge-building communities by encouraging alternative discourse patterns that lend themselves to building knowledge. Users produce materials, including textual and graphical notes, and contribute to a database which other users can search, comment on, or organize into more complex informational structures. This system is designed to accommodate distance collaboration ranging from individuals in a classroom to individuals in different geographic locations. The interactions are asynchronous, but might be considered fine-grained sharing in the sense that they are manipulating parts of a larger whole, the knowledge base itself.

GroupKit [Roseman92] is a toolkit designed to speed the development of multi-user applications for distance collaboration. This toolkit was used to create a group sketchpad and a text editing program [Gutwin95] in which multiple cursors are used to support awareness of other users. Application designers are given control over how closely the multiple views are tied together, i.e., when or how often synchronization points occur. Like CSILE, these activities demonstrate a degree of fine-grained sharing since users are editing different parts of a document, but unlike CSILE, these activities are designed for a more synchronous distance situation in which users are working on the document at the same time.

The Color Matcher activity [Bricker95] demonstrates fine-grained sharing in a synchronous activity where all of the users are co-located (in fact, using the same computer with a multiple mouse input device). Here, each of three users is given control over a single parameter of a color, one controlling each of the red, green, and blue dimensions. The three users work together, trying to reach a target color. This activity encourages a level of communication among users (about relative amounts of color, and about the additive color system in general) which is not facilitated in a single-user version of the program or in a distance situation.

The Midpoint activity [Bricker95] was designed to encourage communication about geometric concepts. In two player mode, the players are charged with matching the midpoint of a line segment to a point on the screen. Each user has control of one endpoint of the line segment. In three player mode, each user has control of a vertex in a triangle, and the users are charged with matching the centroid of the triangle to a point on the screen. A level of complexity is added by confining each of the users' cursors to a certain area on the screen. This activity demonstrates a higher degree of complexity in the users' interactions, which we later termed cooperative control. The users are simultaneously defining the midpoint or centroid of a geometric object in a tightly integrated manner.

2.2 Project Background

CoImage is an application shell which contains a number of collaborative activities, including the Image Warper [Bricker96], a drawing activity reminiscent of the Etch-a-sketch $_{TM}$ toy, an image transformer, and the puzzle activity which is described below.

CoImage and the activities it contains were created using a hierarchy of objects, called CO-API, that facilitate the development of collaborative activities. Included in this hierarchy are concepts that are specific to a multi-user platform, such as users and groups, access permissions, and ownership.

This collaborative system is constrained to a specific hardware and operating system platform, because of the hardware technology that was available at the time it was conceived. We use a multiple mouse device connected to a single personal computer over an input bus called *Access.Bus*, from Computer Access Technology Corporation. The interface can have up to eight ports for input devices. The multiple-user version of CoImage is currently supported in only Windows 3.1, since the input hardware was developed for that operating system, but the single user version of CoImage also runs in Windows 95.

3 Description of Puzzle Activity

We have developed a collaborative puzzle activity using both the CO-API object hierarchy and the CoImage application shell. This activity was designed for a number of reasons. First, we wanted to test the usability of the CO-API object hierarchy. Second, we needed an activity in which to explore different interaction techniques and in which to investigate the use of a history/assessment mechanism.

The program that provides the puzzle activity takes a bitmap and splits it into square puzzle pieces. It then presents the pieces on the screen as can be seen in Figure 3 below. The users are then confronted with the task of putting the bitmap back together in a simultaneous, collaborative manner.

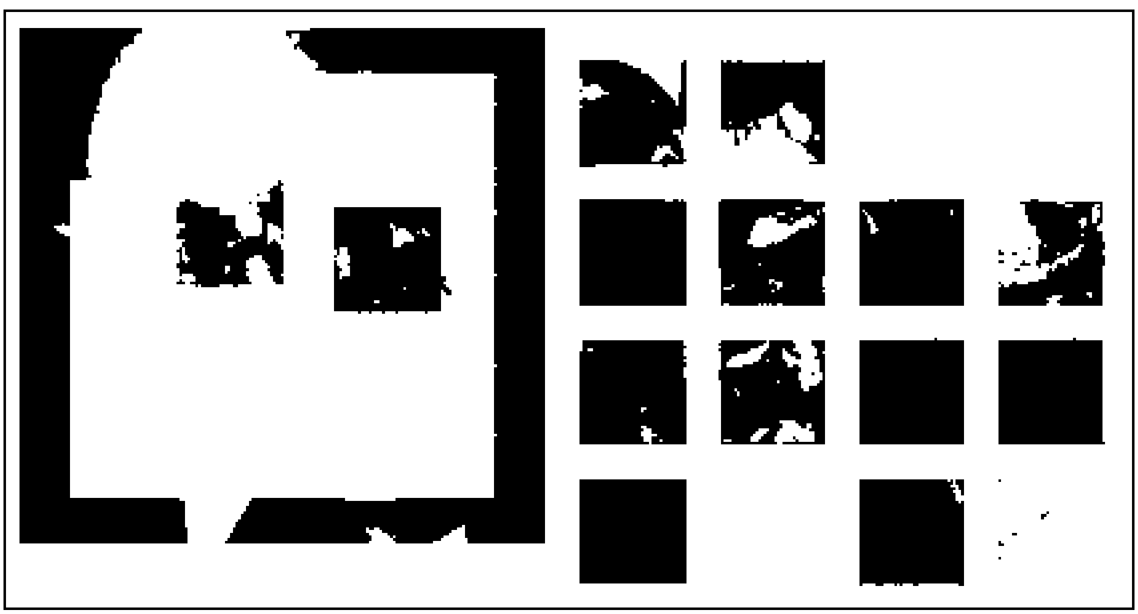

Figure 3: A view of the screen at the beginning of the puzzle activity.

This task differs from a traditional, hands-on jigsaw puzzle in a number of ways. First, it has square puzzle pieces which makes it more difficult to find the correct solution, because there is no way to distinguish or match the pieces solely by their shapes. Second, the puzzle pieces can appear to be in the correct position, even when they're not, since the pixels mesh together in a way that is not possible with the obvious borders on traditional puzzle pieces. The puzzle activity also allows us to do something we cannot easily do with a traditional puzzle: keep a history of information on the user interactions. This will be discussed further in Section 5.

4 Interaction Techniques

We have identified three means of control in a collaborative activity. First, *parallel control* is when the users are allowed to manipulate objects at the same time, but only a single user can manipulate a given object at once. The first version of the puzzle activity implements this type of control. Each user can translate or rotate puzzle pieces by clicking and/or dragging their mouse, but no two users can manipulate the same puzzle piece simultaneously.

The second type of control, *fine-grained control,* o[illegible]n the users are allowed to manipulate the same object simultaneously, using f[illegible]d sharing methods where each user manipulates only one part of the object. [illegible]le, a square can be defined by two points, say the top-left and bottom-right points. One user could control the top-left point of the square while the other user controls the bottom-right point. In Figure 4, the users have performed a translation towards the bottom-left and a 45 degree clockwise rotation. We might also want to enforce collaboration by requiring the users to act simultaneously in order to change the position of the square.

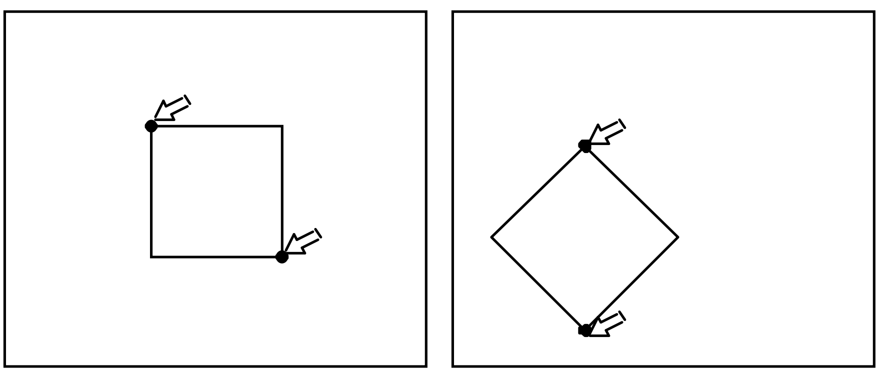

Figure 4: Fine-grained control of a square.

Finally, in *cooperative control*, the users are allowed to manipulate an object simultaneously as in fine-grained control, but using cooperative control methods where users simultaneously define all aspects of an object in an integrated manner. For example, see Figure 5. Here we have a square that is controlled by the position of a line segment. The centroid of the square is constrained to coincide with the midpoint of the line segment. Two users can manipulate the line segment, and thus the position of the square, by each moving an endpoint of the line segment. In the single cooperative interaction shown below the users have applied a number of transformations to the object, including a translation downward and to the left, and a 45 degree counterclockwise rotation, in an integrated manner.

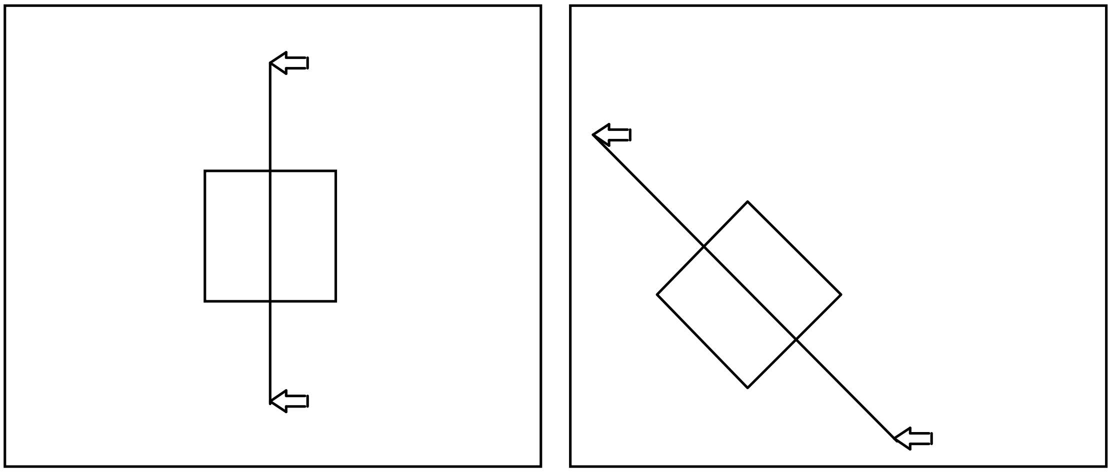

Figure 5: Cooperative control of a square.

4.1 Constraint-based Implementation

In our implementation of cooperative control techniques, we use a very simple constraint system. In general, we define a constraint as an object with multiple input and multiple output objects and some relationship between those objects. Actions performed on one object may affect the state of other objects in the constraint based on the definition of their relationship.

In the puzzle activity, we use one-way constraints [Bharat95], which means that the input objects may have an effect on the output objects, but not vice-versa. For example, let's look again at the cooperative control of a square by a line segment, as presented in Figure 5. Here there is only one input object, the line segment, and one output object, the square. When a user performs an action on the line segment, the square is sent a message to move and redraw itself based on the relationship between the two objects in the constraint.

If we extend the cooperative control technique used in Figure 5, we can create a constraint with multiple input objects. Let us redefine the position of the square based on the intersection between two line segments so that the centroid of the square corresponds to the intersection of the line segments. We now have a constraint with two input objects, the two line segments, but only one output object, the square. Now, when a user performs an action on either of the line segments, the square is sent a message to move and redraw itself based on the intersection constraint.

4.2 Puzzle Activity with Cooperative Control

The cooperative interaction technique shown in Figure 5 is used in a version of the puzzle activity by simply replacing the square with a puzzle piece. Figure 6 presents a screen snapshot of the activity. Here, users can only manipulate a puzzle piece by manipulating the control line (or line segment) connected to it. In this figure, two users are moving a puzzle piece simultaneously by manipulating its control line.

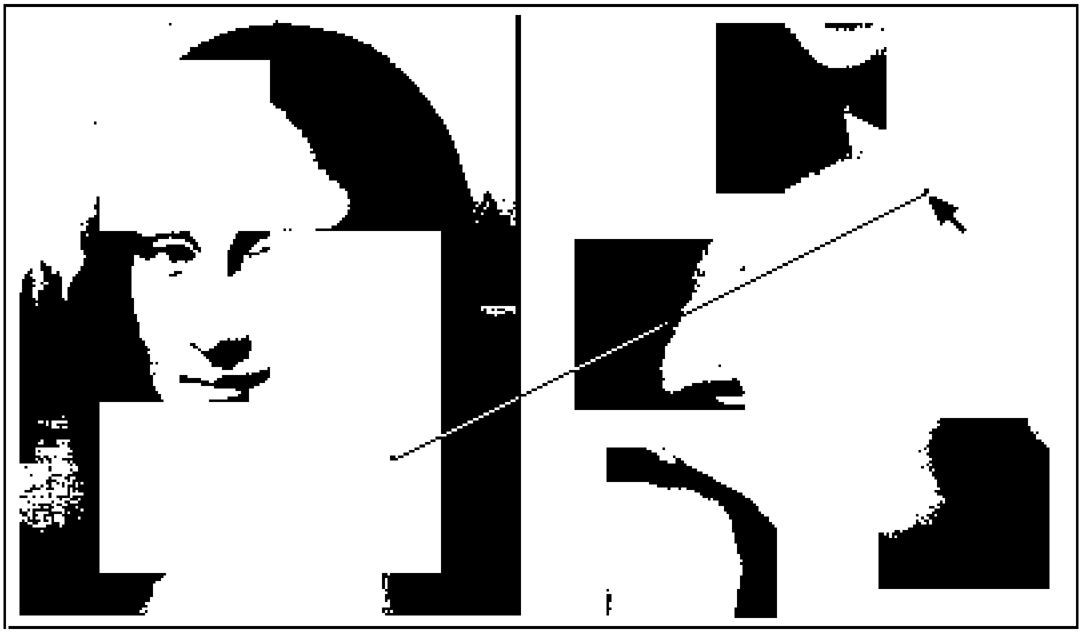

Figure 6: Version of the puzzle activity in which users employ a line segment to constrain the motion of a piece.

In the design of this version of the puzzle activity, we created a control line for each user. This gives the users the freedom to work individually or cooperatively. Note that this implementation does not require two users to manipulate a control line at the same time. If only one endpoint is selected, the other endpoint will remain fixed during the interaction. A single user could move one endpoint and then the other in an iterative manner to achieve the same result as can be accomplished with two users. This design allows us to study whether users prefer to work individually or cooperatively, or possibly with a mix of both interaction styles. One could easily imagine a situation where users were forced to collaborate on moving a puzzle piece, but that was not our goal in this project.

One of our hopes is that cooperatively controlled objects such as those presented here could be designed to help students learn mathematical and other concepts in useful new ways. The cooperative line segment shown in Figure 5 may help users to understand three mathematical concepts: the midpoint of a line segment, translation of an object, and rotation of an object. This complex interaction technique may also help users to learn about the center of rotation, and how it changes as the position of the line segment changes.

5 History and Assessment Mechanism

Since computer-supported collaborative activities are relatively new to computing research and to industry, there has been very little work published on how to assess these activities. For example, the following are some worthwhile questions to which answers may provide help for future researchers, teachers, and developers of collaborative applications:

- Do users prefer collaborative methods for solving a problem over individual methods?
- What aspects of collaboration do users like or dislike?
- What information about users' interactions would be useful for assessment?
- Do user interactions (i.e., talking, pointing, etc.) correspond to events in the activity?
- Do users get better at working together? If so, how quickly, and what influences it?
- How does the cooperative control version compare to the parallel control version? Does it change how they interact with each other in significant ways? How does their task performance compare in the two modes?
- How do students divide up the task? Does this differ for same sex and mixed sex groups?
- Do students find ways around collaborating with others? Do they find new ways of collaborating that had not been anticipated?

Many other questions could be added to this list, but these demonstrate the need for a method to collect and assess data in collaborative activities.

5.1 History Mechanism

With this in mind, we have developed a mechanism to store history information for each object in the CO-API hierarchy. This allows us to record a history for graphical objects, user objects, and user interface objects, amongst others.

The difficulty is in deciding what information is interesting and useful. Currently, we have concentrated on tracking information on graphical objects that are a part of the collaborative activity (i.e., the puzzle pieces in the puzzle activity). There are currently three events that are monitored for each puzzle piece: selection, translation, and rotation. Each of these events is recorded with a time stamp and user identification and other information that is relevant (i.e., to which point a puzzle piece was translated). This information allows us to ask some interesting questions:

- How many interactions were recorded for each puzzle piece? For each user?
- How much time was spent manipulating each puzzle piece? For each user?
- In what order were the puzzle pieces touched? And by which users?
- How did the users divide up the task?
- Did the users work together to find the correct position for a puzzle piece?
- How much time did it take to find the correct position for each puzzle piece? To solve the puzzle?
- How does their performance compare in the different versions or over time?

There are surely other useful pieces of information that could be tracked. We expect that future user studies will give us more insight into what data and interactions would be interesting to record.

5.2 Visualization Techniques

This history information is of little use if we cannot represent or visualize it in an intuitive way. We have chosen to take advantage of the familiar structure of the puzzle activity itself in order to represent the data, based on a visualization principle presented in *Envisioning Information* [Tufte90]. Each puzzle piece is colored according to some statistic. We chose to use a value scale because value is a dimension of color that is relatively easy to perceive [Tufte90].

Figure A3 in Appendix A depicts a visualization of the interaction totals for each puzzle piece, the brightest puzzle piece being the most active one. Here, each interaction corresponds to a selection, translation, or rotation. We can see from this image where the "hot spots" in the puzzle were. These hot spots might correspond to more difficult, possibly more detailed or ambiguous areas of the puzzle, or they may correspond to areas where there were conflicts between different users. The "cool spots" or less active areas of the puzzle may correspond to less difficult areas of the puzzle, or possibly to more familiar structures or objects.

In order to display information about multiple users, we need to add another dimension to our display. See Figure A4 in Appendix A for an example of interaction count by user. Here different colors are used for each user to differentiate between actions that correspond to different users. One user is represented with red while the other is represented with blue. The color used to represent a given user is the same color as that used to paint their on-screen cursor throughout the activity. Any puzzle piece that appears to be a combination of red or blue carries the implication that both users participated in the placement of that puzzle piece. The relative amount of each color is proportional to each user's participation in the placement of the piece. The brightness of the color again indicates the number of interactions performed on each puzzle piece.

Other visualizations in the puzzle activity depict the time spent manipulating each puzzle piece (in totality or by user) and the order in which puzzle pieces were first touched. They are visualized in the same manner by coloring each puzzle piece along a value scale, according to the selected statistic. We give examples of these types of visualizations in the following section.

6 User Study

A small user study was carried out with the puzzle activity in order to get some feedback on the collaborative process, the educational value of such an activity, and the usability of the interface. This user data also helps to assess the usefulness of the visualization techniques described above.

First, let us describe the procedure used during the study, and then let us present some results and visualizations of the history information that was collected.

6.1 Procedure

In the study, five pairs of graduate students were asked to solve a puzzle in two different versions of the software. The first version, referred to as the *parallel version*, allows the users to work simultaneously but on different puzzle pieces. An example of the parallel version of the activity is depicted in Figure 3. The second version, referred to as the *constraint version*, allows users to simultaneously manipulate a puzzle piece. Note that this version facilitates simultaneous interactions, but does not require them. An example of the constraint version of the activity is pictured in Figure 6.

Each group in the study first solved a puzzle in the parallel version and then solved a different puzzle in the constraint version. There were two images used for the puzzle throughout the study, the Crater image in Figure A1 of Appendix A, and the Mountain image in figure A2. (All figures from the user study can be found in Appendix A.) Each of these two images were split up into 25 puzzle pieces. The image chosen for the parallel and constraint versions was varied between groups.

Before beginning each activity, the users were given instructions and a quick demonstration on how to manipulate a puzzle piece in the appropriate version of the software. In the parallel version, they were told to click and drag with the left mouse button to move a puzzle piece, and to click with the right mouse button to rotate a piece. In the constraint version, they were told that they must manipulate the endpoints of the line segment in order to move and/or rotate a puzzle piece. They were given a demonstration of two ways of doing so, one with a single user manipulating the puzzle piece, and the other with two users simultaneously manipulating a piece.

After completing the study, users were asked to fill out a questionnaire on the activity. The questionnaire is attached in Appendix B. Also, interaction data was collected automatically by the software as the activity took place, using the history mechanism described previously in Section 5.

6.2 Results

The results from the study were fairly encouraging. The constraint version of the activity seemed to encourage users to communicate and collaborate, and have more educational value than the parallel version. The feedback from the questionnaire also seemed to indicate that many of the subjects thought the constraint version was more fun than the parallel version, and that it was enjoyable to work in a collaborative environment.

From general observations throughout the study and feedback on the questionnaire, we infer that the constraint version of the activity seems to encourage significantly more communication than the parallel version. In the questionnaire responses, users agreed strongly that the interface for the constraint version encouraged them to communicate more (both verbally and with gestures). The communication that did occur seemed to be more focused on the common task and goals. For example:

> "We should try and do the edges first and move in."
> "Hey, go this way." <gesturing with cursor>
> "I'll go down, you go up."
> "Let's carry it back."

Each manipulation required coordinating their actions, which sometimes led to conflicts. Oftentimes, the two users in a team developed their own little language to communicate their intentions. Some of the users' discourse also suggested that they felt a developing sense of teamwork. For example, some of the remarks heard were:

> "That looks good."
> "Hey, we're not doing too bad."
> "We're awesome!"

According to feedback on the questionnaire, a large majority of the users (if not all of them) found it easier to manipulate puzzle pieces cooperatively, as opposed to individually, in the constraint version of the activity. Initially, some users wanted to try moving pieces on their own, but they seemed to concede fairly quickly that it was easier to collaborate. For instance, here is a dialog from one pair of users:

> User 1: "You can just take it [the puzzle piece], since it doesn't need to be rotated."
> User 2: "Okay."
> <tried moving piece alone>
> User 1: "Nah - it's easier if we both do it."

Another group struggled a bit over which method to use, as demonstrated in the following dialog:

> User 1: "It'd be harder to do both endpoints at the same time. Let's just do what we want."
> <tried working alone>
> User 2: "This is much harder."
> User 1: "Help me! I can't go anywhere."

Based on the event history information collected during the puzzle activity, users collaboratively placed more puzzle pieces in the constraint version than in the parallel version of the activity. Collaboration in the parallel version is defined as both users performing at least one transformation on a puzzle piece (i.e., a move or rotation). In this version, users collaborated in placing 7.8 pieces on the average (with 3 pieces being the minimum and 15 pieces being the maximum).

In the constraint version of the activity, collaboration is defined as both users acting in a simultaneous manner to place a puzzle piece. Users collaborated in placing 24.5 pieces on average (with 23 pieces being the minimum and 25 pieces being the maximum) in the constraint version. We should note that a few users tried to place a piece on their own for a while before their partners joined in to help.

Though the human subjects of this study were computer science graduate students who presumably are knowledgeable in graphics and mathematics, there is some indication that there is an educational value to this activity. From feedback on the questionnaire, it appeared that the subjects thought there was something to be learned about translation, rotation, and midpoints in the constraint version of this activity. One might expect that the parallel version would teach just as much about rotation, but, as one user stated on the questionnaire, "[the constraint version] required more thinking about rotations instead of just blind rotations to test all possible placements."

Half of the users preferred to use the constraint version over the parallel version for its entertainment value even though it was generally agreed upon that puzzle pieces in the constraint version were more difficult to move and rotate. This indicates that the users enjoyed working more closely with their partners, as does much of the feedback on the questionnaire. Some subjects noted that simultaneously manipulating a puzzle piece can be frustrating and sometimes caused conflicts, but that the conflicts were easy to resolve. Some users even found conflicts to be funny and thought they added to the enjoyment of the activity.

6.3 Visualizations of Event Histories

Throughout the study, the software tracked the event history for each of the puzzles that were solved. In this section, we present some of the visualizations of this data, and draw some insights from them.

The first type of visualization depicts the number of interactions that were performed on each puzzle piece, the brightest piece being the most active one. In Figure A3 in Appendix A, there is a visualization of the total interaction counts for group 5 using the parallel version of the activity. In this puzzle, which was generated using the Crater image, the brighter area, towards the base of the crater, required the most manipulations.

The next visualization, shown in Figure A4, is based on the same data (interaction counts for group 5 using the parallel version) with one additional dimension to depict this information for each user. One user is represented with red, and the other user with blue. Any puzzle piece that is a combination of red and blue was manipulated by both users. One can see in this figure that the users chose to divide up the task by area, one user taking the top half of the picture while the other worked on the bottom. As they approached the center of the puzzle, it appears that there was a bit more collaboration in placing the pieces. This sort of division of the task was fairly typical in the parallel version of the activity.

In contrast, the visualization in Figure A5 shows the interaction counts for the same group using the constraint version of the activity to solve a puzzle generated from the Mountain image. Notice that almost all of the puzzle pieces were cooperatively placed by both users. There are two exceptions to this pattern: one red and one blue puzzle piece in the upper left corner of the puzzle. These pieces were placed solely by one user.

The second type of visualization depicts the amount of time spent manipulating each puzzle piece. Figure A6 shows a visualization of the total time spent on each piece for a group using the parallel version of the activity to solve the Mountain puzzle. Figure A7 shows a visualization of the time spent on each piece by each user in the constraint version of the puzzle, generated from the Crater image. One might speculate that more time seems to be spent on the more homogeneous pieces, for example, the sky or internal portions of the mountain/crater. The boundary pieces that separate one object from another (for instance, sky from mountain) seem to be placed more quickly.

The final type of visualization depicts the order in which the puzzle pieces were first manipulated. Figure A8 shows the order in which the pieces were touched by a group using the parallel version of the activity to solve the Crater puzzle. The brightest piece was the first one to be manipulated while the darkest piece was the last. This group chose to first tackle the sky and the skyline-crater boundary, saving the more complex regions for last.

In summary, these visualizations do help to reveal some information about how a group divides the task of solving a puzzle, which areas appeared to be more difficult to solve, how much each user participated in the solution, and how much the users collaborated to reach their solution. In Section 8 on future work, we discuss some ways to augment this visualization technique and alternative techniques for visualizing the event histories that may give us more insight into the activity and the collaborative process.

7 Development Experience

This project was used as a test for the CO-API object hierarchy which was created, in part, to speed the development of collaborative activities. CO-API made the job of developing the puzzle software a lot easier in a number of ways. First, it hid a lot of the details that are necessary to a multiple user system, including displaying multiple cursors on the screen, the mapping of users to cursors, and creating intermediate messages for event handlers that contained pertinent information like which user caused an event to fire. It also took care of a number of other conceptually difficult tasks like managing the multiple layers of graphical objects that must be drawn to the screen in a particular order.

The application shell, CoImage, in which the puzzle activity was developed, was also helpful in abstracting away many of the details of coding a Windows application. For example, one did not have to worry about creating a window, a toolbar, menus, dialogs to open a bitmap file, etc. It also provided numerous examples of how to handle the interactions, separate the document and view, and add to the user interface.

Of course, there are always difficulties in using a system that is new to the programmer and evolving during the course of the implementation. There was some overhead in trying to understand the objects and how they fit into the hierarchy, and in learning a new programming environment (MS Visual C++ for Windows). Probably the most difficult part of working with CO-API was the fact that it was (and is) still an evolving system, and, as is often the case with a non-stable system, the documentation was somewhat scarce and sometimes out-of-date.

8 Future Work

There are three general directions for this work that could be pursued in the future. First and foremost, it is critical to do more formal user studies in order to assess the interaction techniques and to determine what information might be interesting to record in the object histories.

One could also develop new and interesting ways to cooperatively control an object. For instance, we would like to implement an interaction technique that could be termed a "herding stick" that is based on a technique which uses a stick-like tool to align objects in a drawing program [Roope96]. In the puzzle activity, the herding stick could be used to move puzzle pieces or groups of puzzle pieces around the screen, and even to rotate the pieces. Two users could control the stick by each manipulating one end of the stick, as seen in Figure 9. There are a lot of user interface issues here like should the stick dynamically change size, how can a user avoid picking up everything in the sticks path if they do not want to, how do puzzle pieces interact with the stick, etc.

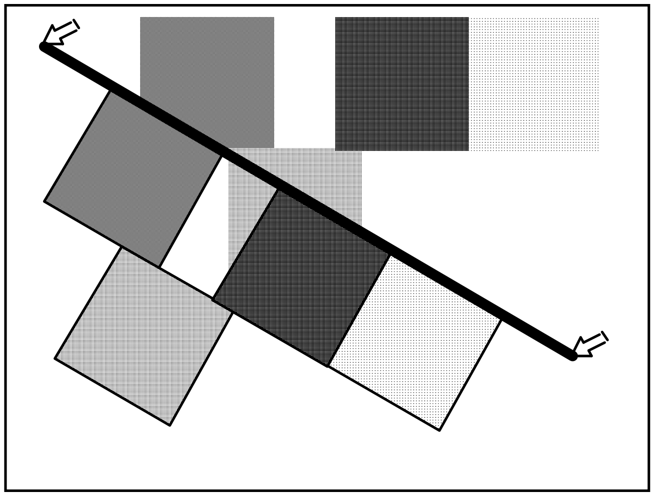

Figure 9: example of "herding stick" interaction

Finally, we would like to find alternative ways to display history and assessment information. For example, in the visualization method presented above, it becomes somewhat difficult to determine the relative interaction count when we represent multiple users with different colors (i.e. How does one compare the intensity of blue to the intensity of red or purple?). One might try to use a different method or an additional dimension to encode the interaction counts. For example, we could use height to redundantly encode interaction counts, where the height of a puzzle piece corresponds to the number of interactions performed on that puzzle piece. We could take this a step further by turning the height dimension into a sort of bar chart where a section of each bar indicates a specific user's interaction count and is colored correspondingly. Of course, we would then have problems with occlusion since some puzzle pieces might be hidden by others, but we could add enough white space in between the puzzle pieces to compensate.

It would also be interesting to visualize the history for each puzzle piece. Let's say you are looking at a visualization of interaction counts by user, like in Figure 8. You may decide that the purple puzzle piece is interesting because more than one user manipulated it. Maybe you would like to see how these users interacted. Did the two users act in a totally disjoint manner, or did their interactions intermix, possibly indicating some communication or other interaction? We might want to allow the user to click on a particularly interesting puzzle piece to get a more detailed view of its history, as is presented in Figure 10.

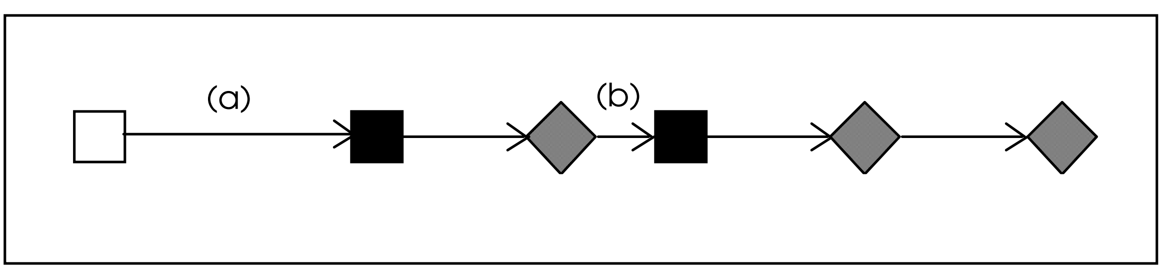

Figure 10: A more detailed view of a puzzle piece's history.

In this case, we have five interactions, each denoted by a small square (the original position denoted by a white box). A non-rotated square depicts a translation while a rotated square depicts a rotation. Each square is filled in with the color corresponding to the user that performed the transformation. The length of the arrow connecting two squares denotes the relative distance a puzzle piece traveled in a translation. So you can see that the first translation, labeled (a), covered a greater distance than the second translation, labeled (b). From this visualization, we can also see that the users did have somewhat intermixed interactions, and we might also notice that one user did all of the translations while another performed all of the rotations (which might be interesting from a pedagogical standpoint).

9 Conclusions

We have described a collaborative puzzle activity in which users can share objects in a simultaneous, integrated manner. The activity is designed to encourage students to work more closely together, encouraging communication and keeping their focus on a common task. The hope is that these cooperatively controlled objects can be created in such a way to help students develop a better understanding of certain mathematical and other concepts. It would require rigorous user studies to determine if these goals were actually met, but we have cited some observations from a user study that seem to suggest progress in this direction.

In order to assess the data from user studies, it is helpful to have a history mechanism and some means of making sense of the recorded data. We have implemented a history mechanism that records specific events, and a way to visualize the data in the context of the puzzle activity itself. Although this mechanism is helpful, it is still some distance away from helping to answer many of our questions about collaborative activities and cooperative control techniques. Unfortunately, much of the data needed to answer these questions is of a qualitative nature, and is therefore difficult to track in any computer system.

Finally, the experience in designing and developing this collaborative activity was an affirmation of the value of the CoAPI and CoImage software infrastructure. The concepts and abstractions provided by the CO-API object hierarchy and the CoImage application shell were great assets in the successful completion of the project.

Acknowledgments

We would like to thank all of the volunteers who participated in the user study, and Ross Ortega for his idea to add depth to the puzzle's history visualizations.

References

[Bharat95] Bharat, Krishna, and Scott Hudson, "Supporting Distributed, Concurrent, One-way Constraints in User Interface Applications," Proceedings of The Symposium on User Interface Software and Technology, Pittsburgh, PA, November 14-17, 1995.

[Bricker96] Bricker, Lauren J., Steven L. Tanimoto, and Evan McLain, "CoImage - a Cooperatively Controlled Image Warper," presented at The Conference on Educational Multimedia and Hypermedia, Boston, Mass., June 17-22, 1996.

[Bricker95] Bricker, Lauren J., Steven L. Tanimoto, Alex I. Rothenberg, Danny C. Hutama, and Tina H. Wong, "Multiplayer Activities that Develop Mathematical Coordination," Proceedings of The Conference on Computer Support for Collaborative Learning, Bloomington, Indiana, October 17-20, 1995.

[Gutwin95] Gutwin, Carl, Gwen Stark, and Saul Greenberg, "Support for Workspace Awareness in Educational Groupware," Proceedings of The Conference on Computer Support for Collaborative Learning, Bloomington, Indiana, October 17-20, 1995.

[Roope96] Roope, Raisamo, and Kari-Jouko Raiha, "A New Direct Manipulation Technique for Aligning Objects in Drawing Programs," Proceedings of The Symposium on User Interface Software and Technology, Seattle, Washington, November 6-8, 1996.

[Roseman92] Roseman, M., and S. Greenberg, "GroupKit: A Groupware Toolkit for Building Real-time Conferencing Applications," Proceedings of The Conference on Computer Supported Cooperative Work, Toronto, Ontario, 1992.

[Scardamalia94] Scardamalia, Marlene, and Carl Bereiter, "Computer Support for Knowledge-Building Communities," The Journal of the Learning Sciences, 1994, 3(3), pp. 265-283.

[Tufte90] Tufte, Edward R., *Envisioning Information*, Graphics Press, Cheshire, Conn., 1990.

Appendix A: Visualizations of Event Histories from User Studies

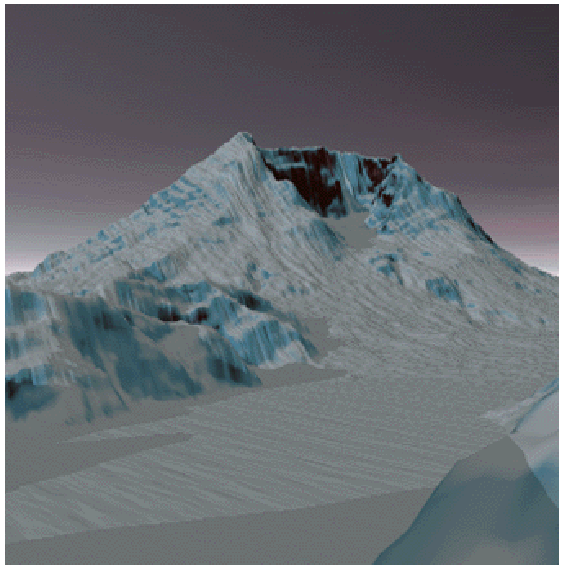

Figure A1: Crater image used in study.

Figure A2: Mountain image used in study.

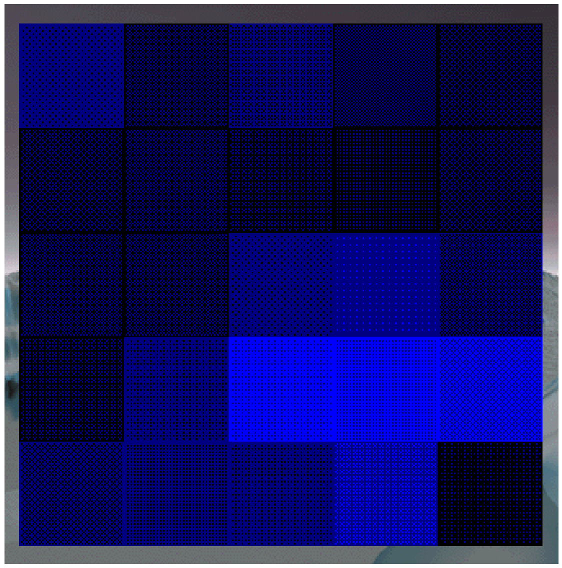

Figure A3: Interaction totals for group 5 using the parallel version.

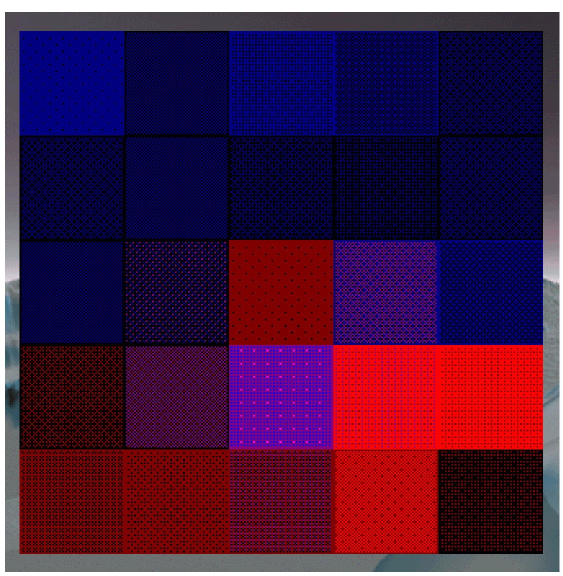

Figure A4: Interaction totals by user for group 5 using the parallel version.

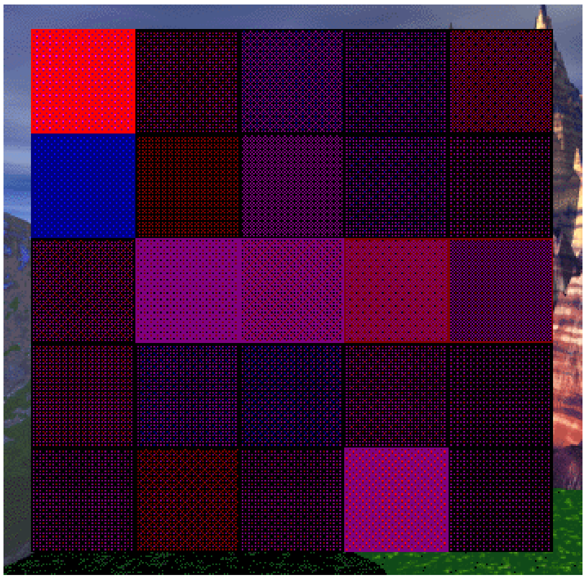

Figure A5: Interaction totals by user for group 5 using the constraint version.

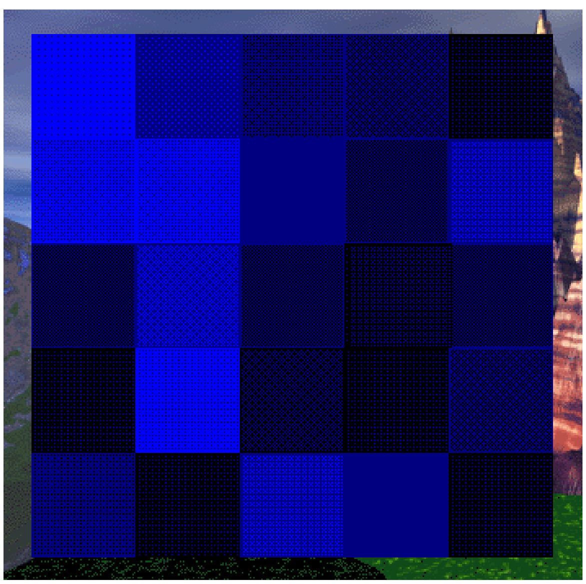

Figure A6: Time totals for group 1 using the parallel version.

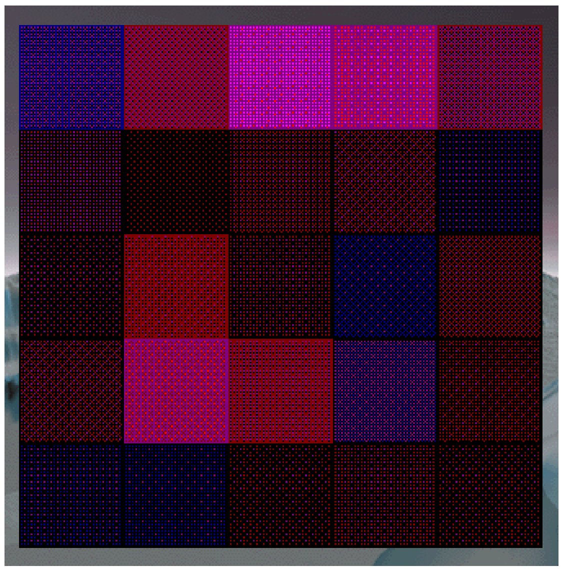

Figure A7: Time by user for group 1 using the constraint version.

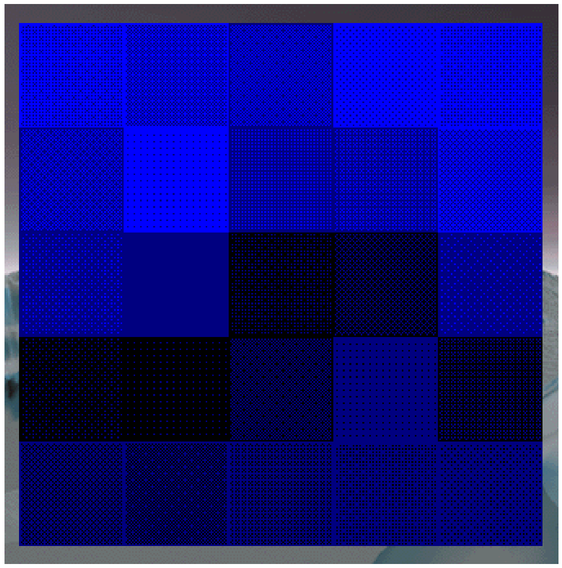

Figure A8: Order in which pieces were touched for group 4 using the parallel version.

Appendix B: Puzzle Activity Questionnaire

Sex:	male	female			Date:
Education:	grade ___	high-school	college	graduate school	
Group:	female-female	male-male	female-male		

Cursor color:	blue	red
First puzzle (in parallel version):	mountain	crater
Second puzzle (in constraint version):	mountain	crater

agree no disag
strongly
opinion
strongly

1.	I like solving jigsaw puzzles.	1	2	3	4	5	6	7
2.	I thought the interface for the ___ was easy to use.							
	parallel version	1	2	3	4	5	6	7
	constraint version	1	2	3	4	5	6	7
3.	I liked using the ___ with a partner.							
	parallel version	1	2	3	4	5	6	7
	constraint version	1	2	3	4	5	6	7
4.	I felt the interface for the ___ encouraged me to communicate more.							
	parallel version	1	2	3	4	5	6	7
	constraint version	1	2	3	4	5	6	7
5.	Given a choice, I'd rather use the ___ by myself.							
	parallel version	1	2	3	4	5	6	7
	constraint version	1	2	3	4	5	6	7
6.	It was easy to move a puzzle piece in the ___ .							
	parallel version	1	2	3	4	5	6	7
	constraint version	1	2	3	4	5	6	7
7.	It was easy to rotate a puzzle piece in the ___ .							
	parallel version	1	2	3	4	5	6	7
	constraint version	1	2	3	4	5	6	7
8.	Only being able to move one part of the constraint hindered me.	1	2	3	4	5	6	7

parallel cons
version
neither
version

9.	It was easier to solve a puzzle in the ___.	1	2	3	4	5	6	7
10.	I enjoyed solving a puzzle in the ___ more.	1	2	3	4	5	6	7

11.	My partner and I talked more in the ___.	1	2	3	4	5	6	7
12.	My partner and I used gestures more in the ___.	1	2	3	4	5	6	7
13.	The ___ helped further my understanding of translation.	1	2	3	4	5	6	7
14.	The ___ helped further my understanding of rotation.	1	2	3	4	5	6	7
15.	The ___ helped further my understanding of a midpoint.	1	2	3	4	5	6	7
16.	The ___ seemed most like the task of solving a physical puzzle.	1	2	3	4	5	6	7
17.	I spent more time manipulating puzzle pieces in the ___.	1	2	3	4	5	6	7
18.	I helped place more puzzle pieces in the ___.	1	2	3	4	5	6	7

19. If you were asked to solve a puzzle as quickly as possible, which would you use?
    parallel
    constraint
20. If you were solving a puzzle just for entertainment, which would you use?
    parallel
    constraint
21. Which version did you prefer working with? Why?

22. What strategy did you use to try to solve the puzzle in the . . .
    parallel version?

    constraint version?

23. In what ways did you feel the activity encouraged you to communicate?

24. In the constraint version with line segments,
    (a) did you prefer manipulating the puzzle pieces by yourself or with your partner? Why?

    (b) was there any advantage to manipulating a puzzle piece together?

25. What did you like or dislike about solving puzzles in a collaborative environment?

26. Do you have any suggestions on how to make the puzzle activity more interesting to use? Or on how to make the interface more intuitive?

Other Comments:

Bugs: