MIDOS: Multimodal Interactive DialOgue System

by

Aaron Daniel Adler

S.B., Massachusetts Institute of Technology (2001)
M.Eng., Massachusetts Institute of Technology (2003)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2009

Author .....
Department of Electrical Engineering and Computer Science
May 22, 2009

Certified by .....
Randall Davis
Professor
Thesis Supervisor

Accepted by .....
Professor Terry P. Orlando
Chair, Department Committee on Graduate Students

MIDOS: Multimodal Interactive DialOgue System

by

Aaron Daniel Adler

Submitted to the Department of Electrical Engineering and Computer Science
on May 22, 2009, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Computer Science

Abstract

Interactions between people are typically conversational, multimodal, and symmetric. In conversational interactions, information flows in both directions. In multimodal interactions, people use multiple channels. In symmetric interactions, both participants communicate multimodally, with the integration of and switching between modalities basically effortless.

In contrast, consider typical human-computer interaction. It is almost always unidirectional – we're telling the machine what to do; it's almost always unimodal (can you type and use the mouse simultaneously?); and it's symmetric only in the disappointing sense that when you type, it types back at you.

There are a variety of things wrong with this picture. Perhaps chief among them is that if communication is unidirectional, it must be complete and unambiguous, exhaustively anticipating every detail and every misinterpretation. In brief, it's exhausting.

This thesis examines the benefits of creating multimodal human-computer dialogues that employ sketching and speech, aimed initially at the task of describing early stage designs of simple mechanical devices. The goal of the system is to be a collaborative partner, facilitating design conversations.

Two initial user studies provided key insights into multimodal communication: simple questions are powerful, color choices are deliberate, and modalities are closely coordinated.

These observations formed the basis for our multimodal interactive dialogue system, or MIDOS. MIDOS makes possible a dynamic dialogue, i.e., one in which it asks questions to resolve uncertainties or ambiguities. The benefits of a dialogue in reducing the cognitive overhead of communication have long been known. We show here that having the system able to ask questions is good, but for an unstructured task like describing a design, knowing what questions to ask is crucial. We describe an architecture that enables the system to accept partial information from the user, then request details it considers relevant, noticeably lowering the cognitive overhead of communicating.

The multimodal questions MIDOS asks are in addition purposefully designed to

use the same multimodal integration pattern that people exhibited in our study.

Our evaluation of the system showed that MIDOS successfully engages the user in a dialogue and produces the same conversational features as our initial human-human conversation studies.

Thesis Supervisor: Randall Davis
Title: Professor

Acknowledgments

I have been at M.I.T. for 12 years, 4 as an undergraduate, and 8 years working towards my Ph.D. After countless late nights working on problem sets and research, digesting volumes of information (drinking from the proverbial fire hose), countless committees, thought-provoking lectures and seminars, wonderful memories, and friendships that will last a lifetime, I have finally finished. I could not have achieved what I did without the help, advice, and support from many professors, colleagues, and friends. For that, I will always be grateful.

I'd like to thank my thesis committee members Jim Glass and Rob Miller. Without their helpful suggestions and insightful questions, my research and thesis would not be as good as it is today. I am especially thankful for Jim's reminders that my thesis is not about the physics simulator, and Rob's help and insight into the user studies and system interaction.

I cannot imagine a better advisor than Randy Davis. Randy always asks the right questions at the right times, encouraged me when I needed it, and pushed me when I needed to be pushed. Our weekly conversations have been essential to this work. I will be forever grateful for all of his help. My only regret is not asking him for his opinion and help more often.

I'd also like to thank Howie Shrobe for his help over the years, especially as I began this journey working on the intelligent room.

I have had great officemates over the last eight years, and they have all become even better friends. Harold Fox, Gary Look, Max VanKleek, and I shared an office as we adjusted to being graduate students. Tracy Hammond and Mike Oltmans always provided wonderful suggestions, helpful advice, and most importantly lasting friendship. A special thanks to Sonya Cates and Tom Ouyang who are amazing officemates and friends, and had to put up with me talking to my computer, sometimes with increasing frustration. They were also especially supportive as I finished this thesis.

Thank you to Jacob Eisenstein for being a great friend and collaborator. The user

studies we conducted together and papers we wrote together were important parts of my research.

I would also like to thank the rest of the Multimodal Understanding Group for all their advice, suggestions, feedback, and friendship, over the years. These include Mark Foltz, Christine Alvarado, Oskar Bruening, Chih-yu Chao, Andrew Correa, Rebecca Hitchcock, An Ho, James Oleinik, Metin Sezgin, Yale Song, Kevin Stolt, Ying Yin, and Olya Vesselova.

Thank you to my fellow graduate students at CSAIL; I appreciate all of your help and friendship over the last n-years. A special thanks to Ali Mohammad, especially for the LaTeX tips.

Thank you to T!G for making sure I was warm, dry, and had computers and printers that worked. A special thanks to Anthony, Noah, Jon, Jack, and Garrett who were always happy to answer my questions.

Thank you to Nira for all of her help and for making sure we didn't eat pizza at every group meeting.

I owe the most thanks to my family for all of their advice, support, prodding, and love. Thank you Mom, Dad, Rachel, Dana, Wendy, and Andrew. A special thanks to my grandparents, Grandma, Oma, and Opa; I'm glad I finally finished so you can see me graduate.

Most importantly, thank you to my wife Leah. She has supported me on every step of this adventure, giving me invaluable encouragement when things weren't going well and sharing in the happiness and joy when they did. She's been supportive through all the late nights at lab and has dealt with all of my stress. Leah, I couldn't have done this without you and I'm eternally grateful.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Consider an ordinary conversation between two people. It typically involves multiple modalities including speech, gesture, sketching, and facial expressions. Often, it is symmetric in the sense that both participants communicate multimodally, with the integration of and switching between modalities basically effortless. In contrast, most current communication between a person and a computer is tedious, slow, and in most cases requires the use of a keyboard and mouse. This thesis examines the benefits of creating multimodal dialogues using sketching and speech between a computer and a person, in particular for domains involving early stage design tasks. The goal of the resulting system is to be a collaborative partner for a user describing these early stage designs.

1.1 The Power and Limitations of Sketching

Pen-based input, including sketching, has been an input method for computers for some time [54]. Previous computer systems have shown how sketching can provide an easy and powerful way to input data directly into the computer. There are many domains for which sketching is suitable for thinking about, communicating, and recording the early stages of designs: chemistry diagrams [46], road design [10], electrical circuit diagrams [6], floorplans, mechanical engineering sketches [30], PowerPoint slides [34], maps for military course of action diagrams, maps for real estate,

and more [19, 56, 16]. Some example sketches from these domains are shown in Figure 1-1. This thesis focuses on the domain of early stage designs of simple mechanical devices, but provides techniques and ideas that are more generally applicable to domains with graphical, verbal, and dynamic elements.

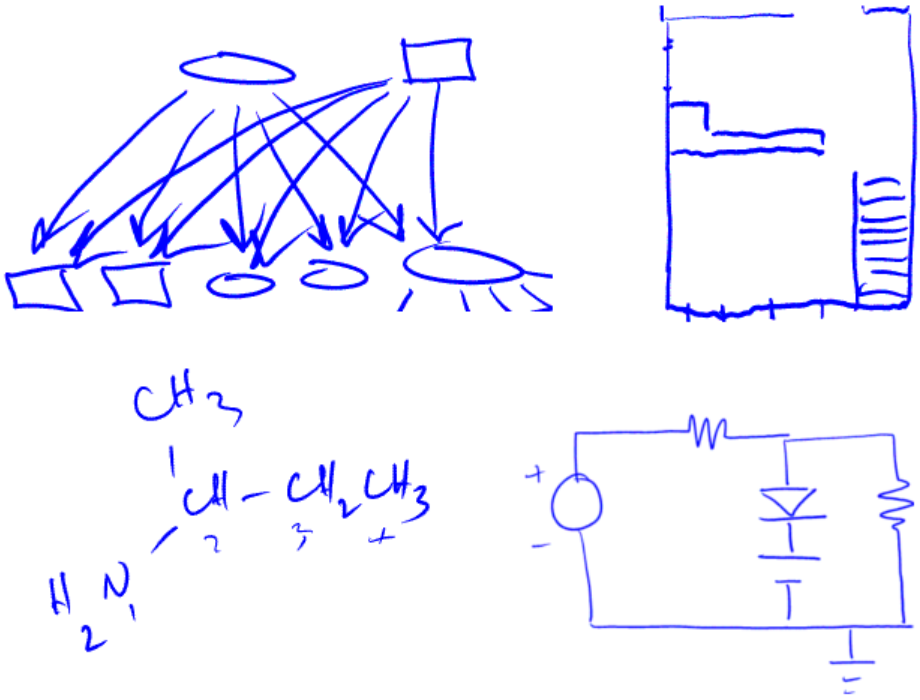


Figure 1-1: A few example sketches from different domains. Starting at the top left and going clockwise: a family tree, a floorplan, a circuit diagram, and a chemistry diagram.

Sketching allows users to easily and directly indicate particular components, features, or properties. In circuit design, for example, instead of drawing on paper then re-entering the design in an electrical CAD system, a user can directly enter an electrical circuit diagram by sketching. The digital input can then be used to produce a simulation of the circuit, quickly showing the user how the circuit functions and enabling the user to spot errors. Once a basic sketch has been drawn, the user can use sketching again to supply additional information. For example, after a user has sketched a simple mechanical device, she can indicate a component's angle, rotational direction, or motion distance by simply drawing a stroke.

Although sketching can be very useful, expressing every detail about a device using

sketching can be difficult or impossible. A simple example is Newton's Cradle (see Figure 1-2), a system of pendulums that consists of a row of metal balls on strings. When you pull back a number of balls on one end, after a nearly elastic collision the same number of balls will move outward from the other end of the system. Although this system seems simple enough to sketch, it is in fact nearly impossible to draw so that it operates properly. The device works because the pendulums are identical and the metal balls just touch each other. Sketching the pendulums to have these properties proves to be nearly impossible.

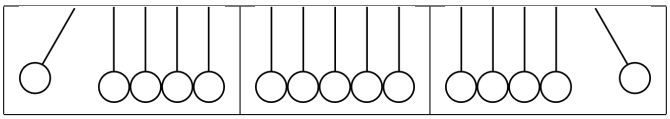


Figure 1-2: A sequence of images showing Newton's Cradle when one of the pendulums is pulled back and released.

A second shortcoming of sketching by itself is illustrated in Figure 1-3, drawn by a user study participant. The sketch is incomprehensible without knowing what was sketched. Knowing that it's a sketch of a Lego robot helps slightly, and seeing the photograph of the robot (Figure 1-4) helps more. Comparing the sketch and the photograph, the relationships between parts of the sketch and the robot can be more easily identified and the sketch makes considerably more sense. Still, there remain parts of the sketch that cannot be identified. For example, different parts of the sketch represent different perspectives of the robot.

1.2 Mixing in Speech

Sketching is powerful and expressive, but it does not provide the best way to communicate all information. Some information is notoriously difficult to express using sketching alone, as Newton's Cradle and the robot illustrate. A more comprehensive interaction can be created by combining sketching and another modality such as speech. The combination of modalities allows a user to communicate information more easily, aims to lower the cognitive load of communication, and forms the basis

Figure 1-3: A sketch of a robot.

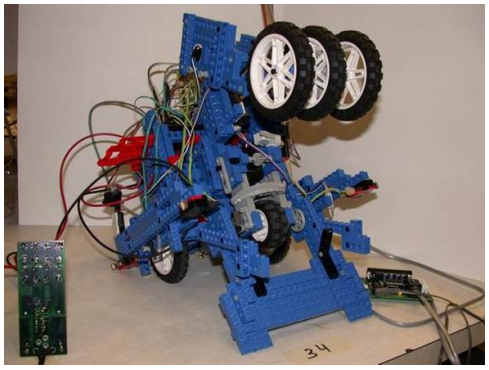


Figure 1-4: The robot sketched in Figure 1-3.

of a rich verbal and graphical communication with the user. With multiple modalities to choose from, the user can communicate using whichever modality seems most natural.

Why choose speech to complement the sketching? Sketching is often accompanied by speech that, although informal, conveys a considerable amount of information. People often use these sketches in a discussion about the design in which participants can all sketch, speak, and ask or answer questions about the design. The interaction about the design with another person helps work out details and uncover mistakes. The combination of speech and sketching provides more information than either input alone [9].

Returning to the earlier examples, speech provides an additional channel for information and makes descriptions of these devices possible. Adding speech to the description of Newton's Cradle clarifies the details of the device. The constraints can be expressed verbally: "There are five identical, evenly spaced and touching pendulums." Then the sketch can be updated appropriately [2].

In the second example, the original sketch was accompanied by detailed speech that described the different views of the robot, what the components of the robot were, and how the components worked together to accomplish a goal. These details are missing when the sketch is viewed alone. Figure 1-5 illustrates some of the different views and components of the original robot sketch.

Some information can be expressed more easily using sketching, including information about location and connectedness. Some information can be expressed more easily using speech, including information about properties of objects. By combining the two modalities, the user is free to use whichever one they find most natural.

1.3 Beyond Multimodal Input to Dialogue

The previous sections described several advantages to multimodal input that combines speech and sketching. Although multimodal input contains more information, it also has several drawbacks: interpreting the input is more complex, the potential sources

(a) Side view (b) Top view (c) Gear box

Figure 1-5: Several views and components contained within the original robot sketch.

of errors and ambiguities increase, and it introduces the possibility that a user can provide conflicting information in different modalities. First, errors might occur in either the speech input or in the sketch input, increasing the quantity and complexity of errors that a system must handle. Second, conflicts between modalities can arise either directly from the user's input or indirectly due to an incorrect recognition of the user's input. In any of these cases, the system can lack the information needed to interpret the input from the user.

A central problem with open-ended, free-form input is *information deficit*: a key piece of information is missing that prevents acting on the user's input. Information deficit can arise because the input modalities conflict, a reference is uncertain, or a design is under-specified. Examples will illuminate the possible situations in more detail.

Returning to Newton's Cradle, imagine a situation in which the system has a correct interpretation for most of what the user has expressed, but conflicting inputs prevent it from determining a definitive action to take. If the user refers to four pendulums but there are five drawn, there are two possible interpretations. The user could actually want to refer to all five pendulums or she could be referring explicitly to a particular subset of the pendulums. In a conversation between two people this

would be resolved with a question. Here the computer lacks a way to gather the required information and thus cannot update the sketch.

Another case of information deficit in Newton's Cradle could be a user referring to a pendulum without indicating which pendulum. For example, saying "This pendulum moves to the left" provides insufficient information because the system does not know what pendulum the speech utterance "this" refers to.

An under-specified design is shown in Figure 1-6. In this spring-block system, the direction of motion of the spring and the block cannot be determined from the sketch. Even with multimodal input, the system cannot query the designer for the needed information.

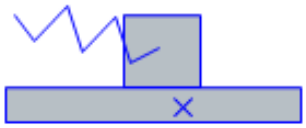


Figure 1-6: An under-specified spring/block system. The direction the spring will move in is not specified.

How do people solve this problem? They talk with each other. Both dialogue participants can use the same modalities to communicate. This thesis extends this principle to human-computer interaction and enables the computer to present ambiguities or uncertainties to the user in the same way a person would ask a clarifying question. The benefits of dialogue in lowering cognitive effort are well established: in the absence of a dialogue, the speaker must anticipate and preemptively eliminate every ambiguity and must ensure that the communication is both complete and unmistakably clear, an exhausting set of demands. Human conversation is (often) easy, in part, because we rely on the listener to ask when things are unclear. Similarly, the computer can leverage the interaction to ask questions of the user when necessary. The uncertainties that arise from the multimodal inputs can likewise be handled by

having the computer ask the user for clarification.

This thesis describes an approach that uses the computer to generate and ask the user questions to clarify the details of her design. This allows the user to use multiple modalities in her descriptions and leverages the computer's ability to interpret the device and focus the questions on the areas it needs help with. The dialogue approach enables the user to more easily communicate the required information as opposed to the alternative approach in which the user specifies a complete, unambiguous description all at once.

The benefit of allowing open-ended input using both speech and sketching is clear, but the limitations are also evident. Without providing an avenue for the system to clarify the user's intent or acquire new information, the system may not be able to act on the user's input.

1.4 Dialogue

The previous sections have made the case for supplementing sketching with speech and for having a dialogue with the user, although a dialogue system is not a new idea [36, 37, 49, 57], our approach is unique for the *dynamic* dialogues that it generates.

The task for the system in our target domain is to simulate the behavior of the device using a qualitative physics simulator. The system asks the user for additional information whenever it determines that the current physical situation is unclear or ambiguous, or when the user's input has not been understood. The user's answers (delivered multimodally) update the physical model or clarify a previous response allowing the simulator to take the next step which, in turn, affects which questions are asked next. The dialogue is thus driven from moment to moment by the physics, not by a prepared script or a set of fields that need to be filled in to run a database query.

Allowing the computer to ask the user for clarifications allows the system to combine its knowledge of the system with the additional information the user supplies in her answers. Partially understanding the situation allows the system to generate

appropriate questions to ask the user. These multimodal conversations allow the user to more easily describe the function and behavior, paths, trajectories, movements, and structure of the device.

Our goal is not to have perfect speech or sketch recognition – rather we want to do the best we can with both inputs and get a good idea of the user's intention. By making the system capable of a two-way dialogue, we can ask the user for clarifications. Benefits of having a two-way multimodal dialogue include encouraging a rich dialogue with the user, and making the interaction closer to the kind of interaction a user would have with another person. Asking the user questions will help keep the user engaged, help the user refine and clarify the design, help the system learn more about the sketch, and help make the system more of a partner in the design process. An example illustrating the dynamic nature of the dialogue can be found in Section 1.6.2.

1.5 Design Assistant

Our goal is to make the computer a collaborative partner for early design moving beyond a sketching system to a multimodal system that incorporates speech and dialogue capabilities. Although there are systems that allow the user to utter simple spoken commands to a sketch system [15, 20, 37], our goal is to move beyond simple commands to create a multimodal digital whiteboard that allows the user to have a more natural conversation with the computer. Instead of being limited to simple commands (like uttering "block" while pointing), the users should be able to say whatever comes to mind. Although speech recognizers may not be able to understand everything the user says, the goal is to have the system understand *enough* of the sketch and *enough* of the speech to engage the user in a sensible conversation [1].

Traditional dialogue and command-driven systems make many assumptions about what computer-human interaction should be like and typically involve quite structured dialogues. Although such approaches are tractable, well-understood, and sometimes quite useful, they might not be the optimal form of multimodal interaction for

open-ended domains such as design. To better understand the characteristics of open-ended dialogue in a design domain, we conducted a study of human-human dialogues to ascertain the requirements for a multimodal conversational design assistant.

The challenges faced in building such a design assistant include integrating the speech and sketching inputs, interpreting the input, and determining how certain the system is of the interpretation. The system must:

- understand and manage the dialogue,
- determine what questions to ask the user,
- determine how and when to ask these questions, and
- understand the responses.

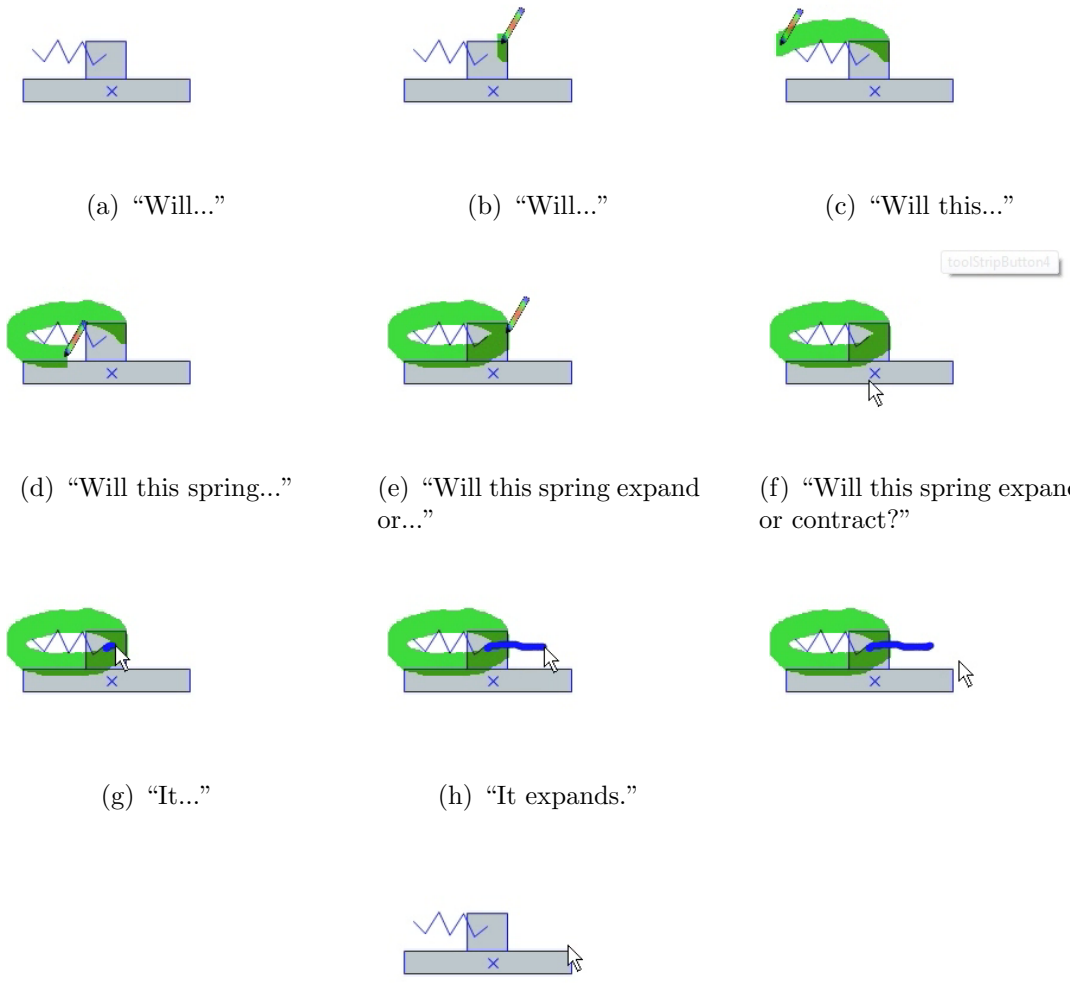
1.6 MIDOS

Based on results from user studies, the Multimodal Interactive DialOgue System, or MIDOS, was constructed. MIDOS simulates the behavior of the device, asks the user for additional information when necessary, and updates the physical model based on the user's answers. The conversation centers around changes that are occurring in the devices. MIDOS generates a computer-driven dialogue that asks the user open-ended questions in contrast to commercial systems, which can be computer driven, but essentially follow a flow chart.

1.6.1 Example of MIDOS

The system asks its questions by generating multimodal output, for example, circling a spring and asking aloud "Will this spring expand or contract?" Figure 1-7 illustrates the stroke drawn by the system and the accompanying speech. The figure also shows the user's multimodal response. As we discuss below, there are several challenges in generating coherent simultaneous speech and pen output and timing them properly: much like an orchestra score, both the correctness of the individual parts (sketching

and speech) and their timing are vital to the composition. Appendix B contains a complete example of a MIDOS dialogue.



(a) "Will..." (b) "Will..." (c) "Will this..."

(d) "Will this spring..." (e) "Will this spring expand or..." (f) "Will this spring expand or contract?"

(g) "It..." (h) "It expands."

Figure 1-7: A series of screen shots that illustrate MIDOS asking a multimodal question (1-7(a)-1-7(f)) and the user's multimodal response (1-7(g)-1-7(h)).

1.6.2 Dynamic Dialogue Example

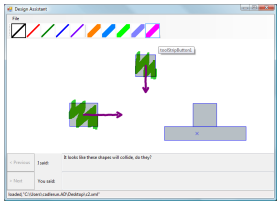
The dynamic nature of the conversation the system produces is illustrated by the sequence of snapshots in Figure 1-8. Figure 1-8(a) contains three bodies: a left body that has a velocity to the right, a middle body that has a downward velocity, and a right body that has no velocity. There are several possible collisions that may

occur; the system cannot figure out what collisions will or will not occur because the velocities do not have magnitudes. We illustrate two possible outcomes to show the dynamic nature of the dialogue.

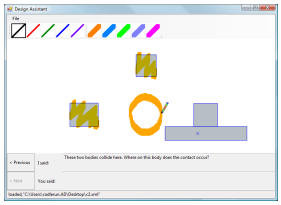
Assume that in the first scenario the user intends for the middle body to collide with the left body hitting it from above. But the initial situation is ambiguous: with the information given we cannot determine which collisions, if any, will occur. As a result, the system begins by asking "Do these two bodies collide?" while circling the left and middle bodies (Figure 1-8(a)). The user answers "yes," and the system continues by asking a series of questions to determine what happens next (Figure 1-8). In this case the user indicates exactly where the collision occurs (Figure 1-8(c)) and then what the velocity of the left body is after the collision (Figure 1-8(e)). The left body moves off the screen and so does the middle body due to the force of gravity. The right body is positioned as shown in Figure 1-8(i).

Alternatively, if the user indicates that the left and middle bodies do not collide, the system will ask questions about a collision between the left and right bodies (shown in Figure 1-9). The collision between those bodies results in Figure 1-9(d), in which the middle body has moved off the screen and the velocity of the left body has been transferred to the right body. The final positions in this case are illustrated in Figure 1-9(e) with only the left body still visible.

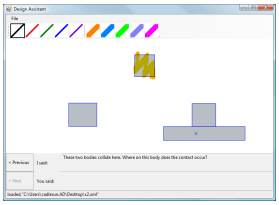
These two cases show how different the dialogue and the result can be based on the user's response to the system's questions. In one case the left body moved off the screen, and in the other case it is the only body still visible.



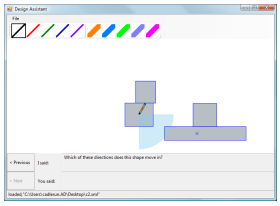
(a) System asks: "It looks like these shapes will collide, do they?"



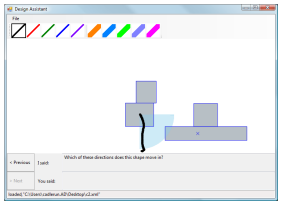
(b) After the user says "Yes," the system says "These two bodies collide here," while circling the collision location.



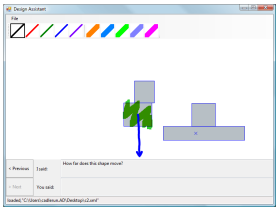
(c) The system continues: "Where on this body does the contact occur?" while highlighting the middle body.



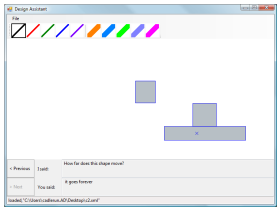
(d) The user indicates the contact location and the positions of the bodies are updated. The system inquires: "Which of these directions does this shape move in?"



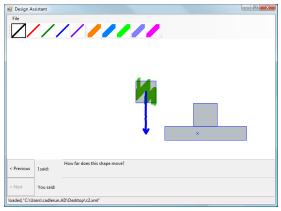
(e) The user replies using the shown stroke and says: "This direction."



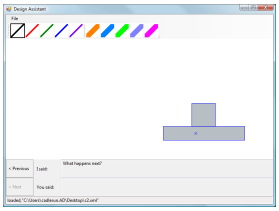
(f) The system asks: "How far does this shape move?"



(g) The user replies: "It goes forever." The system moves the body off the screen

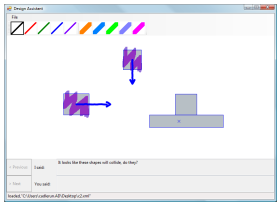


(h) The system asks about the other shape which starts to move again due to gravity.

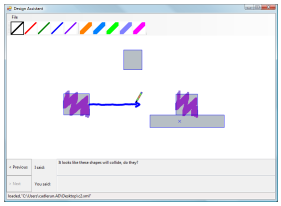


(i) The user replies and the system moves that the body off the screen too.

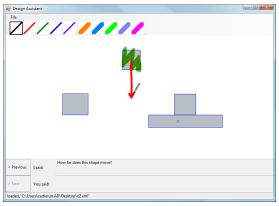
Figure 1-8: A series of screen shots that indicate some of the questions and answers when the user says the left and middle bodies collide.



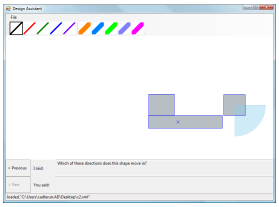
(a) System asks: "It looks like these shapes will collide, do they?"



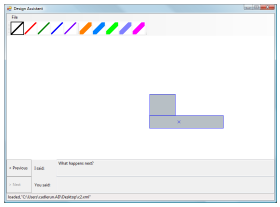
(b) After the user says "No" the system asks about the other collisions and asks: "It looks like these shapes will collide, do they?"



(c) This time the user answers affirmatively and the system proceeds to ask about the middle block. The system asks: "How far does this shape move?"



(d) The user tells the system that it goes forever and the middle block is moved off the screen. The system updates the position of the left and right bodies based on the collision. The right body has been moved to the right and the system now asks: "Which of these directions does this shape move in?"



(e) The user specifies a direction and states that the right block also moves off screen.

Figure 1-9: An alternative series of screen shots of questions and answers when the user says that the left and middle bodies do not collide.

1.7 Contributions

The principle contributions of this thesis are:

- the insights gathered from user studies that revealed useful facts about human multimodal conversations,
- MIDOS with its dynamic dialogue and novel interaction, and
- the evaluation of MIDOS.

First, the data gathered for MIDOS from the various studies provided key insights into how humans converse in a multimodal fashion using speech and sketching. Among the key findings from the studies: simple questions can initiate long, detailed responses, and complex coordination occurs between user's speech and sketching.

Second, the key findings from these studies led to the creation of MIDOS. The interaction style of MIDOS is novel: the user and the computer interact using the same modalities and the same space to sketch in. In the interaction, the computer asks sensible, tightly integrated, multimodal questions, in an attempt to elicit more complete answers from the user; by more complete we mean answers that are more complex than one-word yes/no answers. The user, in turn, can respond using a combination of speech and sketching, which the system interprets based on expected responses to the question it asked.

Another key principle of MIDOS is its dynamic nature. MIDOS determines the next question based only on the current state of the physics and the question history. If the physical layout or answers to the questions are different, then different questions will be asked. The questions are asked to gather enough information to move the simulation of the device to the next state. This information varies depending on the state of the particular device. Many current speech systems have a set number of specific fields that need to be filled in to run a query, in contrast to the open ended nature of the question MIDOS uses. The system leverages what it knows to find out what it doesn't.

Finally, we evaluate MIDOS to determine the strengths and weaknesses of the system. The key results from this evaluation study include a reduced number of ink color changes, detailed responses to questions, and an overall preference for MIDOS. Participants also provided some suggestions for future improvements to MIDOS.

1.8 Organization of this Thesis

Three major themes are woven through this thesis. *Multimodal dialogues* are constructed using rich verbal and graphical communication, and the advantages of these dialogues are discussed. The interaction is *computer driven* with the hypothesis that giving the system more initiative and the ability to ask sensible questions will elicit more complete answers from the user even if the system cannot completely understand them. The progression between user study and system and back again traces the *evolution* of the system.

Chapter 2 describes our initial user study that looked at speech and sketching input. The subsequent chapter describes the details, results, and analysis of our more recent study of multimodal dialogues. Chapter 4 provides a high-level view of the chapters that discuss MIDOS in detail. Chapter 5 describes the multimodal inputs to the system, and Chapter 6 describes the output modalities. The core of the system that connects the input modalities to the output modalities is discussed in Chapter 7. We then evaluate MIDOS and describe the results of the evaluation in Chapter 8. We conclude with chapters about related work (Chapter 9), future work (Chapter 10), and our contributions (Chapter 11).

Chapter 2

Multimodal Device Descriptions

People naturally use speech and sketching when describing devices. One of our goals is to allow the computer to understand and generate descriptions that are as similar as possible to the descriptions that people naturally use. To see how people naturally describe devices, a qualitative user study was conducted to examine the informal speech and sketching of users describing a mechanical system. This user study examines multimodal input; Chapter 3 discusses a study involving multimodal dialogues.

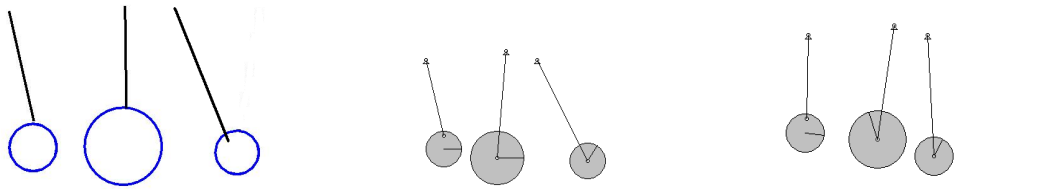
The purpose of this study was to identify the vocabulary used to describe mechanical systems and find out which features of the systems were described verbally and which were drawn. In addition, relationships identified between the speech and the sketching input data (e.g., timing, references, ordering) can be exploited to create a system that responds to the user's utterances. This chapter describes the study, the key results, and the resulting system. More details can be found in [1, 2].

2.1 Motivation

Why do we want to combine sketching with speech? Sketching is a powerful modality for capturing designs; enabling users to quickly draw a device in a modality that is very similar to the way they would use a pen and a piece of paper. Many components of devices are easily sketched, for example, the shape and location of components in

the mechanical domain can easily be sketched (a pulley, a block, a pivot, etc.). But there are limitations to the descriptive power of sketching. In particular, constraints or relationships between components of the device are difficult to describe by using sketching alone.

As mentioned in Chapter 1, Newton's Cradle illustrates this difficulty clearly (see Figure 2-1). For this device to correctly operate, all the component pendulums must be identical, and the spacing must be precise so that the pendulums just touch each other.



(a) The initial sketch

(b) The start of the simulation

(c) The simulation result

Figure 2-1: An initial sketch and the resulting simulation.

The limitations of sketching can be overcome by complementing the sketching with speech. The additional information for Newton's Cradle can be easily specified by saying "there are three identical and touching pendulums." The computer can then adjust the sketch accordingly and create a simulation of the device that functions correctly (see Figure 2-2).

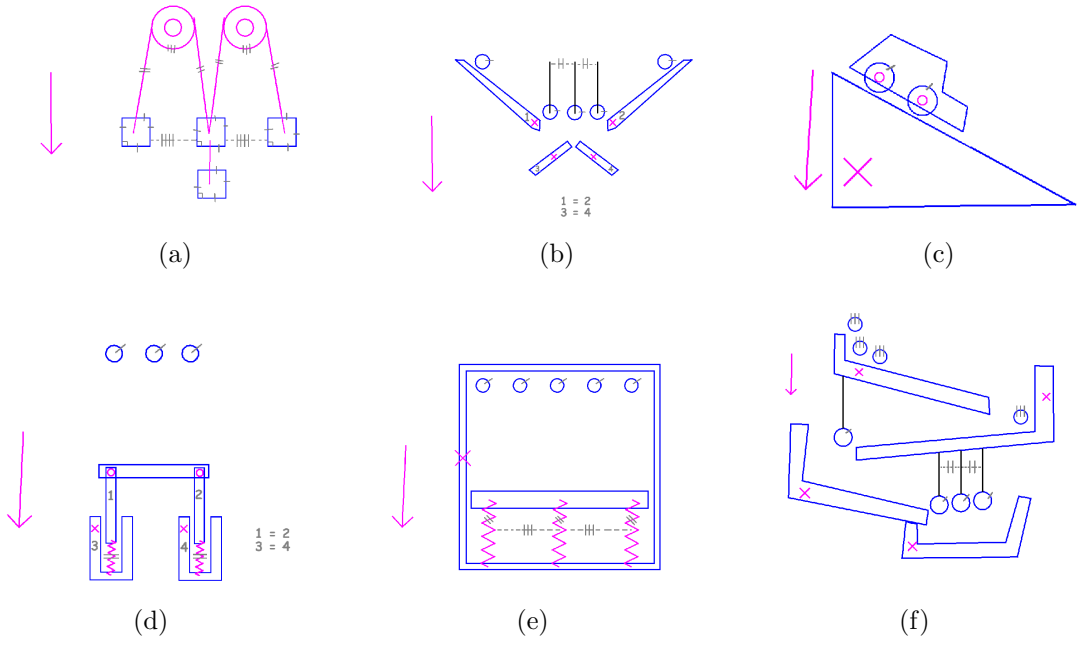
(a) The start of the simulation

(b) The simulation result

Figure 2-2: The corrected version of the simulation.

2.2 Study Setup

The study of multimodal input was designed so that it avoided biasing the participants toward any particular vocabulary. The participants were shown small versions of six mechanical devices and were instructed to draw enlarged versions of the devices on a whiteboard while providing a verbal description [2]. The participants were told to assume their audience was a small group of people, such as a physics tutorial. The figures had marks to indicate replicated components and equal distances (Figure 2-3). These graphical marks were provided to get an idea of how the participants would describe identical or equally spaced objects without inadvertently biasing their language with a written or spoken description that used a particular set of vocabulary. The sessions were videotaped to facilitate subsequent analysis. The six participants in the study were drawn from the M.I.T. community.



(a) (b) (c)

(d) (e) (f)

Figure 2-3: The devices that the participants sketched. The grey hash marks, grey numbers, and equivalences indicate congruent components.

2.3 Data Analysis

The data from the videos were analyzed by manually transcribing and assigning timestamps for individual speech events (roughly, phrases) and sketching events (part of a drawn object). Topic shifts and corresponding events were manually annotated in a second pass over the data. For example, for the device in Figure 2-3(b) the three sketched pendulums and the speech describing the three pendulums formed one topic. Speech phrases about different topics were separated; speech about the objects on the ramp in Figure 2-3(b) was separated from the speech describing the ramp itself.

2.4 Observations about the Data

The transcribed, timestamped, and grouped data were analyzed to collect the vocabulary the participants used and uncover any patterns in the integration of their speech and sketching. Several patterns emerged concerning disfluencies, key phases, consecutive instances of shapes, timing gaps, and cross-modality coherence.

- Disfluencies (e.g., "ahh" and "umm") were good indicators that the participant was still talking about the same topic. For example, a participant said: "And then we have this umm [draws rectangle] table." The word "table" occurs after the disfluency "umm" and is the conclusion of the sentence not the beginning of a new sentence.

- Key phrases such as "there are", "and", and "then" were indicators that the participant was starting a new topic. For example, a participant started a new topic beginning with "then" in the following utterance: "Then we have like a [drawing rectangles] divider in that box."

- Consecutive instances of the same drawn shape indicate that the shapes represent the same type of object and that the topic is the same for all of the instances. For example, in a device that contained two pulleys, most participants drew the pulleys consecutively.

- A gap (an absence of input) of more than about 0.8 seconds in both the speech and sketching inputs indicated a topic shift by the participant. In the following utterance, the participant separated two topics with a pause: "So now we have a box [draw box] with five circles [draw one circle] inside on the top [draws four circles] [pause] And then we have like a..."

- Participants never talked about one topic while sketching about another topic (a phenomenon we term *cross-modality coherence*). For example, participants did not speak about springs while drawing a ramp.

- Other patterns were composed of combinations of the above observations. For example, divisions between topics frequently occur when a pause precedes one of the key phrases. One topic segment might include three sketched springs and the speech phrase "[pause] And that's ahh filled with springs."

Recognizing the above patterns does not require any domain-specific knowledge about mechanical devices and could apply to other domains. However, domain-specific vocabulary is necessary for understanding the multimodal input and modifying the sketch. Linking the noun "pendulum" with the corresponding sketch components – a rod connected to a circular body – is critical to resolving references to "pendulum" or "pendulums." A deeper understanding of the structure and function of a pendulum is required to act upon references to modifier adjectives such as "identical" or "touching."

2.5 Multimodal Input System Overview

The observations from the user study formed the design of a multimodal input system that could modify a sketch created in ASSIST [4, 5] by combining speech recognition and sketch interpretation. This combination allows users to describe the structure of devices more completely and enables device descriptions that were not possible with sketching alone. We describe this initial system and then discuss the shortcomings that led to the subsequent user studies and MIDOS.

Returning to the pendulums in the Newton's Cradle example, the system enables users to say things such as "there are three identical equally spaced pendulums" while sketching several pendulums. The system will then respond by making the pendulums identical and spacing them equally, as shown in Figure 2-4.

The system has several components including speech recognition, a rule system, and an integration framework. The speech recognition sends an interpretation of the speech to the rule system. The rule system then groups the speech and the sketching into related units. These units are combined with domain specific knowledge in the integration framework to modify the sketch.

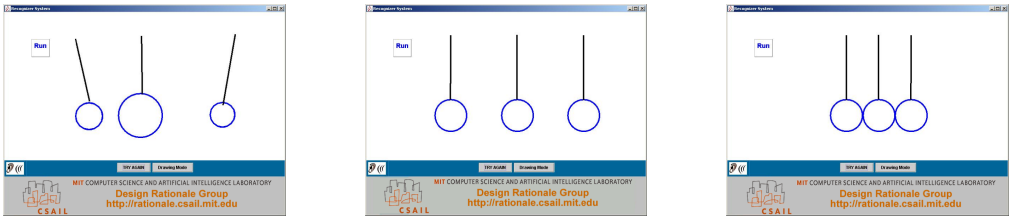


Figure 2-4: Three successive steps in our multimodal system. The first image shows the sketch before the user says anything. The second image shows the sketch after the user says "there are three identical equally spaced pendulums." The third image shows the sketch after the user says that the pendulums are touching.

2.5.1 Speech Recognition

The vocabulary and sentences from the transcribed videos, augmented with a few additional words (e.g., plurals and numbers), were used to create a speech recognizer for the system. The speech understanding is provided by part of Galaxy[32], a speaker-independent speech understanding system that functions in a continuous recognition mode. The system allows users to talk without prior calibration of the system and without the need to warn the system before each utterance. Both factors help create a natural user interface.

2.5.2 Rule System

One of the system's tasks is grouping the speech and sketching that are related to each other. This is accomplished using a rule system based on the observations and patterns described in Section 2.4. The system uses a manually derived set of approximately 50 integration rules that encapsulates the knowledge gathered from the user study.

The rules were created using 18 data sets and are independent of specific features and of the vocabulary of the mechanical engineering domain. The rules use properties of the speech and sketching such as:

- grouping objects that are the same shape (e.g., grouping consecutively drawn triangles),
- using the timing between the speech and sketching events to identify overlapping events and pauses between events,
- looking for the key words that were good indicators that the user started a new topic, and
- identifying key times that separate groups of related speech and sketching events.

For example, one rule identifies key times based on speech events and indicates a possible new group when a speech utterance starts with a key word that is preceded by a pause. The result of the rules is a determination of the key times that delineate groups of speech and sketching events that refer to the same objects. This might produce a group that included two sketched springs and the speech phrase "that's suspended by springs on the bottom."

2.5.3 Integrating Speech and Sketching

Integrating the speech into the sketching framework allows the user's utterances to affect the sketch. There are three steps to the processing of the speech and sketching:

the grouping of speech and sketching described above, followed by an evaluation step and an adjustment step.

After the rule system performs an initial partitioning of the speech and sketching input, the subsequent steps make adjustments using domain specific knowledge and vocabulary. In the second step, a search is conducted within a group found in the first step to align the speech and sketching events (e.g., match the speech event containing the word "pendulums" with any sketched pendulums). This step evaluates the speech and sketching to determine how well the different input modalities match.

The third step adjusts the speech and sketch groupings by searching adjacent groups when the speech and sketching inputs do not correspond exactly. This step relaxes the constraints determined by the rules to provide more flexibility in the grouping and accounts for domain specific vocabulary.

2.5.4 Sketch Modification

After the speech and sketching inputs are grouped, a grammar framework can be used to modify the sketch appropriately. The grammar framework recognizes certain nouns and adjectives and thereby produces a modest level of generality. For instance, one noun it can recognize is "pendulum." The system needs to be told what a pendulum looks like (i.e., a rod connected to a circular body), so that it can link the user's intentions (e.g., drawing three identical pendulums) to a modification of the sketch. Adjectives it can recognize include numbers and words such as "identical" and "touching." Adjectives are modifications to be made to the sketch (e.g., "touching"). The framework is general enough to allow the system to be extended to work with more examples.

In the Newton's Cradle example, we needed to space the pendulums equally and make them identical. Changing the sketch required performing a simple translation from the descriptions, such as "equally spaced," to a set of manipulation commands that were implemented in ASSIST. Figure 2-4 illustrates one possible interaction that results in a modification of the original sketch.

2.5.5 Mismatched Inputs

The system has a minimal coping strategy for dealing with mismatched input modalities. This situation arises when the number of objects identified in speech differs from the number of objects identified in the sketch. The process of segmenting and aligning the data also allows us, in a limited way, to use both modalities in interpretation. For example, if the user draws three pendulums and says there are two, the system must ignore the speech because it cannot identify the subset of pendulums. If, however, the user says that there are four pendulums, the system will wait for another pendulum to be drawn before attempting to group the speech and sketching events. The difficulties in dealing with mismatched inputs led directly to the user study described in Chapter 3, where we engage the participants in a dialogue and ask them questions to clarify their multimodal input.

2.6 Results

The performance of the system is discussed in more detail in [1]; here we summarize key results. To determine how well the rules work, the transcript files from the videos were parsed and run though the rule system with each speech and sketching action presented sequentially as if arriving from a user. The data used to test the system was separate from the data used to create the rule system.

The results of running the rules on the video transcripts were compared in detail to hand-generated results for 4 data sets that comprised the test set. There were 29 topic separation times in the hand-generated segmentations. The computer-generated segmentation matched on 24 of these (82%) while generating 18 false positives. The errors the system made fell into several different categories. Some of the false positives were immaterial to parsing, but some were due to the system's limited knowledge of objects and spatial relationships. For example, an anchor on an object was grouped with the object in the hand segmentation but not in the computer segmentation, because the rules do not have any knowledge of the meaning of the anchor. Similarly, the rules cannot take advantage of spatial relationships between different objects, for

example a set of ramps. To correctly group these objects, the system would need additional contextual information.

2.7 Limitations of the System

The speech and sketching system worked well for simple cases, but it is limited in several ways:

- it requires the modalities to be in numerical agreement,
- the manipulations of the sketch that the system can perform are limited,
- the system has no knowledge of the spatial layout of the sketch, and
- the system can not communicate with the user in a bidirectional fashion, preventing it from asking the user questions about ambiguities or the design.

First, the system can improve the sketch only if the interpreted speech and the sketch are numerically consistent. If, for example, the user draws and talks about three pendulums, but the system identifies only two of them in the sketch, it can not edit the sketch. Likewise, if the system recognizes three sketched pendulums but recognizes the speech as referring to a different number, it can not perform any manipulations of the sketch.

Second, the system requires time-consuming, hand-coded manipulations of the shapes for each word associated with a modification. For example, for pendulums to be "identical," the balls must be the same diameter, the rods must be the same length, and they must be connected the same way. Additionally, the angle of the pendulums is important, as is the starting location of the connected rods. These manipulations are specific to pendulums; different manipulations would be required for other adjectives or different objects.

Third, the system lacks knowledge about the spatial relationships between different sketched shapes, such as shapes drawn inside other shapes. For example, the system did not know that a particular anchor was drawn inside of a ramp. Knowledge

of these spatial relationships is necessary for recognizing and modifying more complex sketches.

Fourth, the system can communicate in only one direction – listening to the user; the system can not ask the user any questions. The system has no way to cope with conflicting information from different input modalities. Bidirectional communication would allow the system to converse with the user to resolve ambiguities and ask questions about the design.

2.8 Broader Implications

This study produced a number of interesting observations about the language and timing people use when describing simple mechanical devices. Three of these proved to be particularly applicable to MIDOS are:

- Disfluencies ("ahh", "umm", etc.) indicated that participant was still talking about the same topic,
- A substantial pause in both modalities was likely an indication of a topic change,
- People display *cross-modality coherence*, i.e., they do not talk about one topic while sketching another.

This study led to an initial system [1] capable of handling sketching and speech, but the initial system lacked the conversational capabilities to resolve uncertain inputs. As Chapter 4 illustrates, MIDOS seeks to overcome these shortcomings.

Chapter 3

Human Multimodal Dialogue Study

The shortcomings described in Chapter 2, notably the inability to query the user about ambiguities or uncertainties in her input or in her sketch, led to the idea of a dialogue with the user instead of a one-way interaction. When people engage in conversation, they naturally exchange information in an efficient and effective manner. These are characteristics that we hope to use to make communicating with a computer as easy and beneficial as communicating with another person. Learning more about the characteristics of human-human dialogues will help us construct a computer system capable of having a similar type of conversation with a user.

We conducted a study to gather data about human-human multimodal dialogues to illuminate the interaction characteristics of dialogues concerning the behavior of a device. This chapter discusses the setup, execution, and results of this study, as well as the implications of the study for MIDOS.

The intent of the study was to examine questions like: what are the characteristics of bidirectional interaction; what questions are asked; how is the sketching surface used to ask questions; how to learn new, out-of-vocabulary terms; how to handle disfluency; how prosody reveals cues about the speakers intentions; how conversations are structured; and how often and when it is okay to interrupt the user.

3.1 Study Setup

The domain for this study was electric circuit diagrams. Eighteen subjects participated, all of them students in the Introductory Digital System Laboratory class at M.I.T. The two conversers, the experimenter and subject, sat across a table from each other (Figure 3-1), each with a Tablet PC. The Tablet PC was equipped with software that provided a window to sketch in. The sketched strokes in each window were replicated in real time on the other tablet, in effect giving the participants a single, shared sketching surface usable by two people at once. The experimenter and subject communicated with each other using only verbal communication and by sketching on the Tablet PCs.

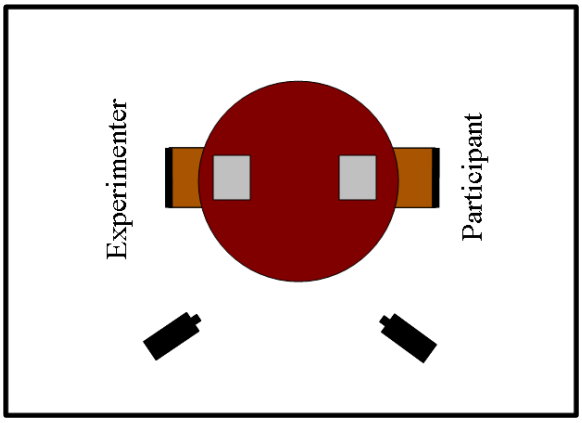


Figure 3-1: Overhead view of the user study layout.

This software allowed the users to sketch and annotate the sketch by using a pen and a highlighter. Buttons above the sketching area allowed users to switch between five pen colors and five highlighter colors (Figure 3-2). Another button allowed users to switch into or out of a pixel-based erase mode, allowing either user to erase parts of any stroke. Finally, there was a button that allowed either user to create a blank page.

The software recorded the (x, y) position, time, and pressure data for each point in every stroke drawn by either user. To enhance the feeling of naturalness, strokes were rendered so that they were thicker when the user applied more pressure. The sketching data was recorded in real time and saved to a file.

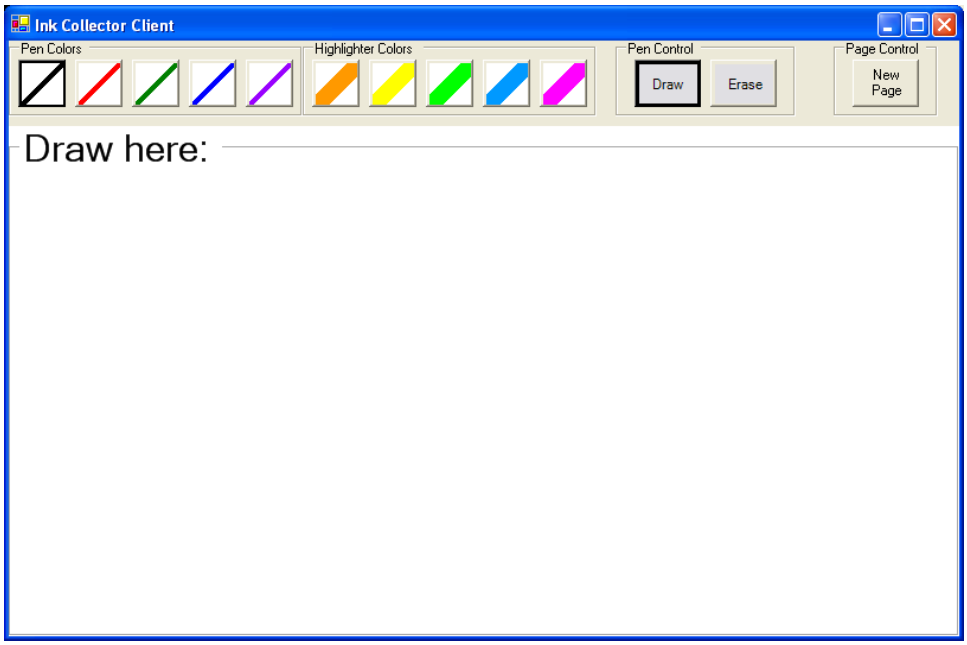


Figure 3-2: The window that the users sketched in.

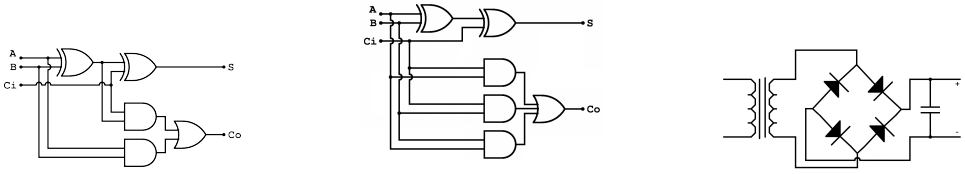
Two video cameras and headset microphones, one for each person, were used to record the study, with the audio and video synchronized. Each camera and the audio from the corresponding microphone were connected to a hardware-based video encoding card (Hauppauge WinTV-PVR 250), and the audio/video stream was digitized using MPEG2 compression (720x480 pixels, 30 frames per second).

A physical barrier between the conversers was considered but not used, because it would have created an unnatural environment and obstructed the video recording. In order to encourage all communication to be done by interacting with the drawing surface, the experimenter looked at his tablet and avoided eye contact with the subject.

The user study software provided a variety of services, including ensuring that the timestamps for the sketch data were synchronized with the audio and video data (using the Network Time Protocol [41]), gathering data about study participants by using a computer-based questionnaire, and displaying instructions. Having synchronized data streams allows us to replay the study as it happened and facilitated analyzing the timing of the speech and sketching events. To ensure participant anonymity the data was anonymized using a random number.

3.1.1 Domain

Participants were instructed to sketch and talk about four different items: a floor plan for a room or apartment with which they were familiar, the design for an AC/DC transformer, the design for a full adder, and the final project they built for their digital circuit design class. In addition, there were instructions and a warm-up condition to familiarize the participants with the system and the interface. The floor plan sketch was used to collect a few sketches in a different domain and to ensure that the subjects were familiar with the interface before they had to describe the more complex circuits. For the AC/DC transformer and the full adder, the participants were given a text description of the circuit and a list of suggested components. They had the option of viewing a schematic of the transformer or adder circuit (Figure 3-3) before they began drawing, but the schematic was not visible while they were drawing.



(a) Full Adder Schematic I (b) Full Adder Schematic II (c) AC/DC Transformer Schematic

Figure 3-3: Schematic views of the full adder and the AC/DC transformer that the participants could choose to view.

During the participant's explanations, the experimenter added to the sketch and asked simple questions about different components in it. The participants were compensated with a movie gift certificate valued at $10.

We would ideally have conducted a Wizard-of-Oz study in which responses to the participant would appear to be coming from a computer. We determined that this was too difficult, given the open-ended nature of the speech and sketching in the study, and instead used the study protocol described below.

Other systems that let users sketch and speak are typically limited in one or more of the following dimensions:

- Command-based speech – The user talks to the system using one or two words,

not natural speech, e.g. [47].

- Unidirectional communication – The system cannot ask questions or add things to the sketch, e.g. [5, 37].

- Annotation instead of drawing – The user can only annotate an existing representation, not use free-form drawing, e.g. [13, 36].

- Fixed set of graphical symbols – The user has to know the fixed symbol vocabulary, e.g. [10].

3.2 Study Analysis

This section describes our initial data annotation process and our qualitative results from the dialogue study. Section 3.3 describes the quantitative results from the study.

3.2.1 Data Annotation

At the conclusion of the study, the collected files included two movie files (one for the participant and one for the experimenter) for each of the four items the users drew, along with one XML file for each page of sketching. The XML files contained a full record of the sketching by both the participant and the experimenter, with precise timestamps for each point. These data files can be used to replay all of the events and interactions that occurred in the study.

The software also allowed us to select parts of the audio tracks for playback and transcription. This transcript was passed to the Sphinx speech recognizer [35] forced-alignment function, which produced precise timestamps for each word. The transcripts were verified by playing the segment of the audio file and confirming that it contained the correct word.

3.2.2 Study Statistics

Data from 6 of the 18 participants were processed as described above. Only 6 participants' data was analyzed due to the time-consuming nature of the transcription process. Each of the 6 datasets contains data from each of the tasks (i.e., the warm-up and four sketching tasks). The total length of the data is approximately 105 minutes; about 17.5 minutes of data for each participant. Cumulatively, the six participants drew 3206 strokes, 74 erase strokes, and spoke 10,848 instances of 1177 words. Cumulatively, the experimenter drew 156 strokes, 3 erase strokes, and uttered 2282 instances of 334 words.

The participants varied in age from 20 to 22, with an average age of 21. There were 14 male participants and 4 female participants. Fifteen of the participants were right-handed. Two of the participants owned Tablet PCs, 11 reported having tried one, and 5 reported never having used one.

3.2.3 Initial Results

The analysis of the study focused on how speech and sketching work together when people are interacting with each other. Figure 3-4 shows one of the sketches, and Figure 3-5 illustrates the type of speech that accompanied it. In general, the sketches contained the circuit itself and additional strokes related to its function or identification of its components. In Figure 3-6 the sketch contains the AC/DC converter and strokes indicating the flow of current through the circuit in each of two operating conditions. In addition, there are several highlighter strokes used to identify components in the circuit.

Our qualitative analysis of the recorded and transcribed data has led to a series of observations divided into five categories: sketching, language, multimodal interaction, questions, and comments. Although these categories aren't mutually exclusive, they help organize the observations and the subsequent discussion.

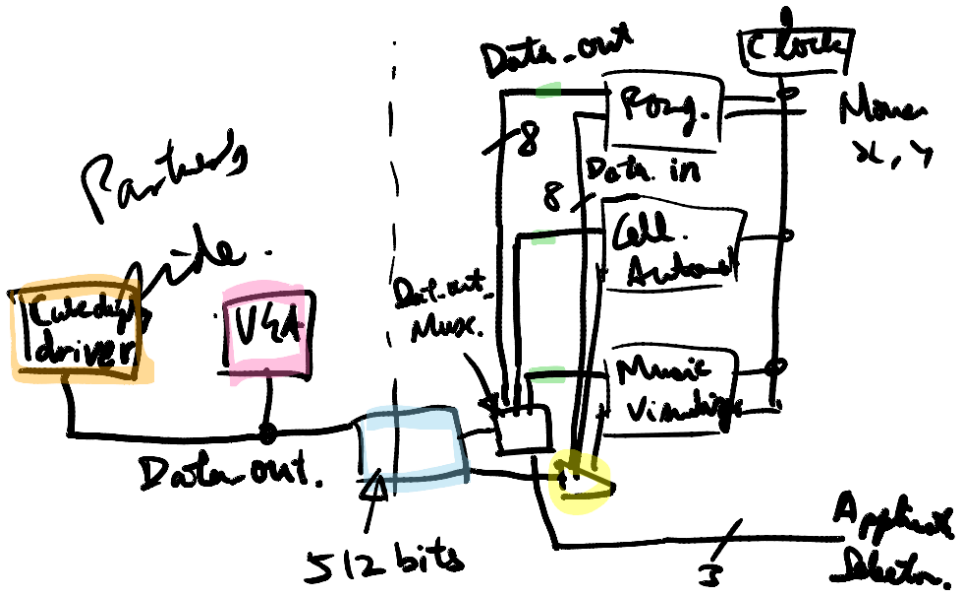


Figure 3-4: A sketch of a participant's project from the dialogue user study.

Experimenter:	so all these outputs are are they all the same these outputs
Participant:	um they're not the same they are the individual um um data out connectors of each of the different um well actually i shouldn't be drawing that that at all

---

Experimenter:	so then what's what's um this piece what's that
Participant:	that would be the mux for the data input actually

---

Participant:	that was a uh uh yeah a memory bank with five hundred and twelve um yep five hundred and twelve bits this ah i could that i had read and write access to

Figure 3-5: Three fragments of the conversation about a participant's project (Figure 3-4). Notice the disfluencies and repeated words (discussed in Section 3.2.5).

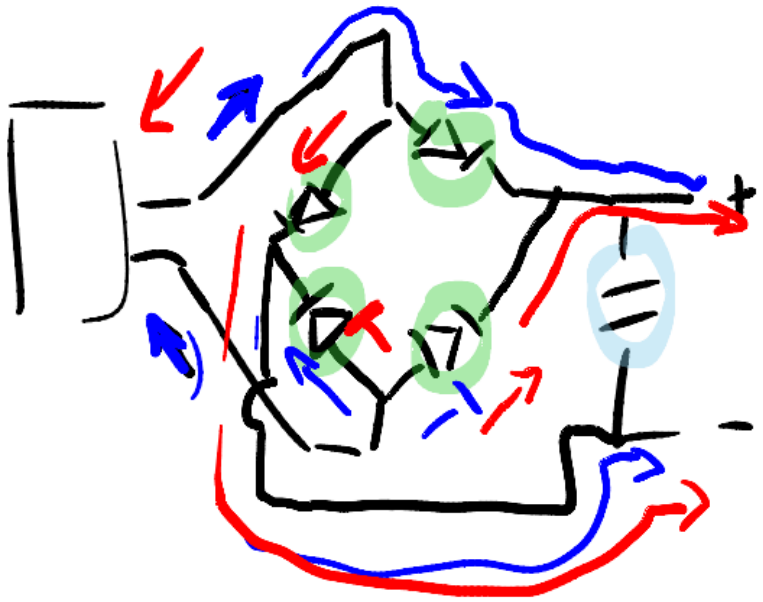


Figure 3-6: A sketch from the dialogue study of an AC/DC transformer.

3.2.4 Observations about Sketching

Ink color was used in several different ways in the sketches:

- To identify regions that were already drawn
- To differentiate objects
- To add an "artistic" character

Identifying Regions

Color was frequently used to refer back to existing parts of the sketch and/or to link different parts of the sketch together. In Figure 3-7, color was used to indicate the location of rooms on a lower floor of the building. In Figure 3-8, three different colors were used to indicate the correspondence between different parts of the sketch – the labeled inputs in the left part of the sketch are highlighted with the same color as the numeric input values in the right half of the sketch. In Figures 3-7 and 3-8, color was critical for identifying references to or connections between parts of the sketch.

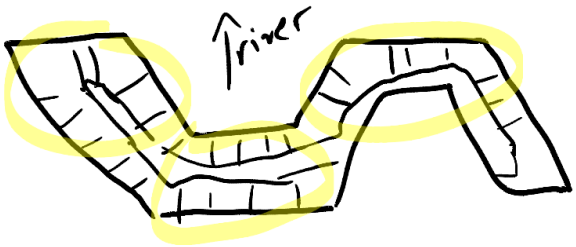


Figure 3-7: Yellow highlighter was used to highlight locations of rooms on another floor in a sketch of Next House dormitory.

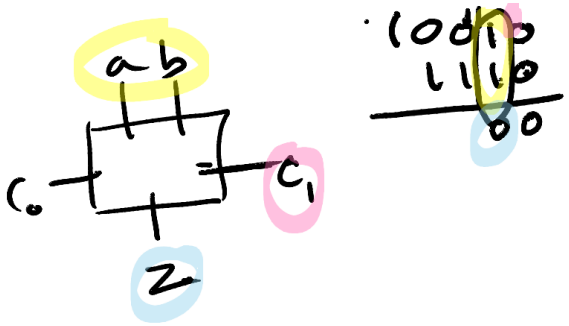


Figure 3-8: Color was used to indicate corresponding areas of the sketch.

Differentiating Objects

Although some participants switched colors while drawing a circuit, different colors were used more often in drawing floor plans to differentiate items. When it does happen, the change in color is an excellent indication that the user is starting a new object. This information would greatly aid sketch segmentation. Figures 3-9 and 3-10 are clear examples of a switch in color used to distinguish objects.

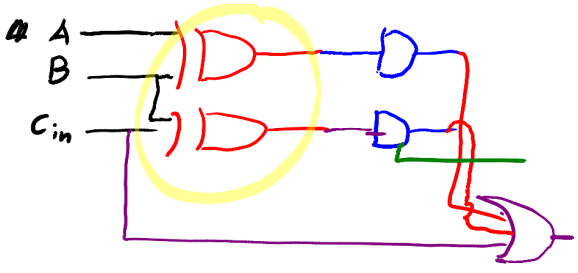


Figure 3-9: Color was used to differentiate the circuit components.

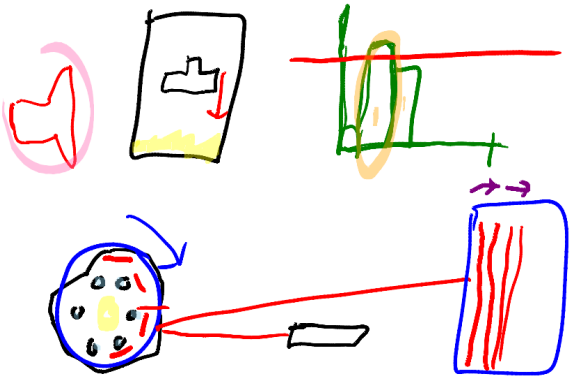


Figure 3-10: Notice that each item in the sketch is a different color.

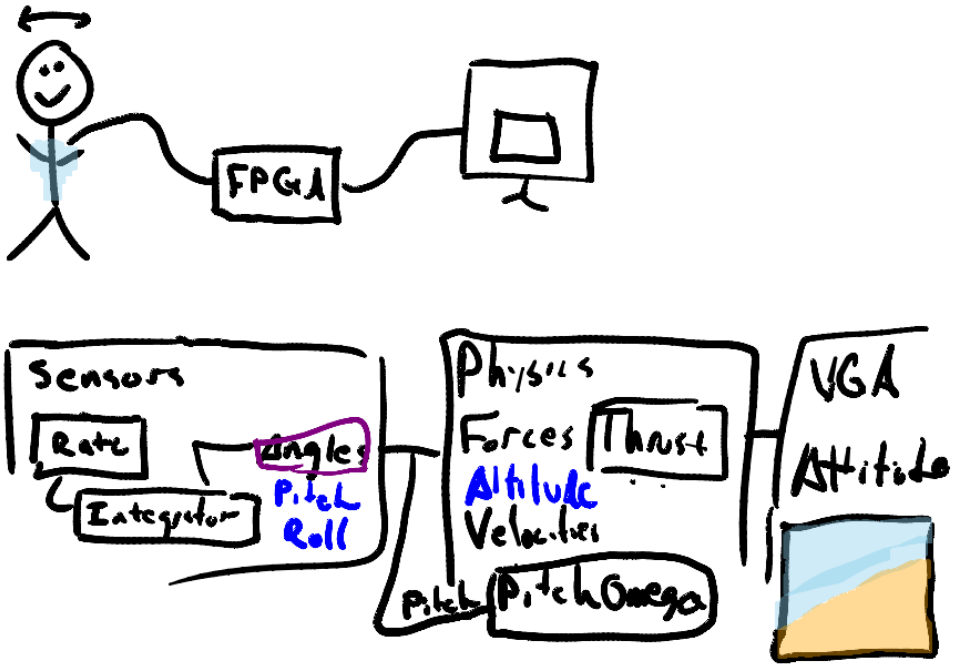


Figure 3-11: Notice the artistic use of blue and orange in the square in the lower-right of this sketch.

Adding Artistic Character

The lower-right corner of Figure 3-11, a sketch of a flight simulator, illustrates an artistic use of color. In this case, the user was describing the operation of the attitude indicator. The lower part is brown, indicating the ground, and the upper part is blue, indicating the sky, just as in real attitude indicators. In other sketches, participants used blue to indicate bodies of water, imitating the color of the real world.

Colors used like this still aid in segmenting the input, but also have deeper meaning because they relate to real-world objects and associations. Matching the colors with the references in the speech is one way the system can make connections between the two inputs. For example, one participant drew a blue rectangle in his floor plan sketch and referenced it by referring to the color: "this one's blue [sic] is a sink."

3.2.5 Language

The language chosen by participants provided several valuable insights. The most readily apparent observation is that the speech tended to be highly disfluent, with frequent word and phrase repetition. This phenomenon appears to occur more frequently when participants are thinking about what to say. Second, participants' responses to questions posed to them tended to reuse words from the question. Third, not unexpectedly, the speech utterances are related to what is currently being sketched. Each of these observations is addressed in turn.

Disfluent, repetitious speech

The repetition of words or phrases in the speech occurred more frequently when participants were thinking about what they wanted to say. One participant who was describing the output "R" of a circuit said: "the result will be R, whereas... if so let's let's eh the result will be R... is that if the carry in is carry eh if the carry in is one, then the result here will be R, this is in case the carry in is one." The speech here is ungrammatical, disfluent, and repetitive, clearly making it more difficult for a speech recognition system. However, the repetition of the key words "result," "carry

in," and "R" should allow us to identify them as the key concepts being discussed. The repetition could also provide evidence that the user is thinking about what to say. This evidence about user uncertainty could help a system better assist the user by asking questions or making suggestions.

Question responses

Participants' responses to questions tended to reuse vocabulary from the question. For example, when asked "so is this the, is that the diode?," the participant replied: "this is the diode, yeah." A system could learn to expect a response to questions to have phrasing similar to the question, facilitating the speech recognition task.

Speech relates to current sketching

Not unexpectedly, the participants' speech relates to what they are currently sketching. For example, in one sketch the participant is drawing a box and while drawing it says "so let's see, we got the power converter over here;" the box is the representation of the power converter he is talking about. This may facilitate matching the sketching and speech events as they are occurring at roughly the same time.

3.2.6 Multimodal

This section discusses three varieties of multimodal interactions between the speech and sketching inputs exhibited by the study subjects: referencing lists of items, referencing written words, and coordination between input modalities.

Referencing lists of items

Participants in the study would often verbally list several objects and sketch the same objects using the same order in both speech and sketching. For example, when sketching a floor plan, one participant said "eh so here I got a computer desk, here I got another desk, and here I got my sink," while sketching the objects in the same order. In another sketch, a participant drew a data table and spoke the column labels

aloud in the same order that he sketched them. The consistent ordering of objects in both modalities provides another method for associating sketched objects with the corresponding speech.

Referencing written words

Participants who wrote out words such as "codec" or "FPGA" referenced these words in their speech, using phrases such as "so the the codec is pretty much built in, into the, like uh standard, um, eh, standard, uh FPGA interface." If the handwriting can be recognized, this information can help identify the words in the speech input, as has been done in [37]. Participants also wrote abbreviations for spoken words, for example, "Cell." for "Cellular." Recognizing these textual abbreviations will also help find correspondences between the sketch and the speech.

Coordination between input modalities

As noted, the speech often roughly matches whatever is currently being sketched. Subjects indicated a tendency to enforce this coordination: if a subject's speech got too far ahead of his sketching, he typically slowed down or paused his speech to compensate.

There were many examples in the study where the participant paused his speech to finish drawing an object, and then continued talking. For example, one participant said "and that's also a data out line" and then finished writing "Data out" before continuing the speech. In another case, a participant said "um, you come in and" and then paused while he finished drawing an arrow to indicate the entrance to the room. These observations provide additional data that the two modalities are closely coordinated. This relationship can be used in a system to help match speech utterances with sketching.

3.2.7 Questions

When the experimenter asked the participants questions, the participants made revisions or explained their design in more depth. This section describes the types of responses that participants gave.

Revision

Some questions caused the participant to make the sketch more accurate. Consider Figure 3-12(a), when the experimenter asked if the three outputs, highlighted in green were the same, the participant realized that the original sketch was inaccurate, prompting him to revise it by replacing one data output line with three separate lines (Figure 3-12(b)).

(a) Original (b) Revised

Figure 3-12: Left: the original sketch, right: after revision. One data output line in the original image has been replaced by three in the revised image.

Broader explanation

Questions about one part of the sketch also spurred explanations about other, unrelated parts of the sketch, as participants apparently decided other parts of the sketch might be confusing as well, based on the question asked. When one participant was asked about a label for a column in a data table, he not only clarified that label, he explained the other four labels in the table.

Comparison questions also encouraged participants to explain the sketch in more detail by explaining how the parts were or were not similar. For example, participants were asked if several different gates in the full adder were the same. One participant's reply was that both were AND gates, while another indicated that one was an AND gate and one was an OR gate.

These elaborate answers to questions were an unexpected result of the study. Asking questions keeps the participant engaged and encourages them to continue talking. The resulting additional speech and sketching data would give a system a better chance to understand the sketch. The interaction also appears to encourage the participants to provide more information about the sketch, and it appears to cause the participants to think more critically about the sketch so that they spot and correct errors or ambiguities. Even simple questions like "Are these ___ the same?" seems to be enough to spark an extended response from the participant, especially if there is a subtle aspect of the objects that was not previously revealed.

3.2.8 Comments

Participants made several comments during the study that did not relate directly to the sketch, but still provided valuable information. Uncertainty was indicated through the use of phrases such as "I believe" or "I don't remember." Some comments related to the user interface, for example, "I'll try to use a different color." Other comments referenced the appearance of the sketch. Two examples of this type of comment are: "it's all getting a little messy" and "I'll draw openings like this. I don't know... I draw li... I drew like a switch before." These comments still provide insight into the

participant's actions, but don't relate directly to what they are sketching. Recognizing the uncertainty or other comments could help create a more natural interface for the users.

Another observation from the study is that both the participant and the experimenter are expected to be able to fill in words that their partner forgot. For example, one participant expected the experimenter to help with forgotten vocabulary, and another participant filled in a word that the experimenter forgot. This might be another way that a system could interact with the user, saying something like "And this is ah..." and pausing, prompting the user to identify the object.

3.3 Quantitative Analysis

Work in [49] reports on a series of user studies in which users interacted multimodally with a simulated map system. They examined the types of overlap that occurred between the speech and sketching, finding that the sketch input preceded the speech input a large percentage of the time. The studies used a click-to-talk model for the audio input, but further work showed that this did not affect the results.

A similar analysis was conducted on the data from our dialogue study. Software was used to match corresponding sketching and speech events in the transcripts. For example, the speech utterance "so we have an arrangement of four diodes" was matched with the strokes making up the concurrently sketched diodes. The speech was segmented into phrases based on pauses in the participants' speech; we call these groups *phrase groups*.

Phrase groups are subdivided into groups containing only a word and the strokes it was referring to; for example, the word "diode" and the strokes making up the diode. We call these groups *word groups*. These two types of groups were generated in light of differences in the nature of overlap between the speech and the sketching events as compared to the results from [49]. The overlap for the *word groups* matches the results in [49] (sketch input preceded the speech input), but the results for the *phrase groups* do not.

The analysis of the nature of the overlap between the sketching and speech events was also taken a step further. The start time of the sketching was compared with the start time of the speech, and the end time of the sketching was compared with the end time of the speech. Table 3.1 shows the nine possible ways the speech and sketching can overlap and the percentage of time each occurred for the phrase groups. Table 3.2 shows the same thing for the word groups. The enumeration of overlap possibilities is the same as in [49].

Speech Precedes (82%)	Sketch Precedes (16%)	Neither Precedes (2%)
Sketch Speech (1%)	Sketch Speech (1%)	Sketch Speech (0%)
Sketch Speech (30%)	Sketch Speech (5%)	Sketch Speech (0%)
Sketch Speech (51%)	Sketch Speech (11%)	Sketch Speech (2%)

Table 3.1: The temporal overlap patterns for the phrase groups. The alignment of the speech and sketching is illustrated in each table cell. The percentage of phrase groups in each category is also noted.

Speech Precedes (26%)	Sketch Precedes (71%)	Neither Precedes (3%)
Sketch Speech (0%)	Sketch Speech (1%)	Sketch Speech (0%)
Sketch Speech (24%)	Sketch Speech (14%)	Sketch Speech (3%)
Sketch Speech (2%)	Sketch Speech (55%)	Sketch Speech (0%)

Table 3.2: The temporal overlap patterns for the word groups. The alignment of the speech and sketching is illustrated in each table cell. The percentage of word groups in each category is also noted.

Unlike the videotape analysis used in [49] to determine the overlap between speech

and sketching, the analysis presented here is based on precise timing data for speech and timestamped points from the pen input, both measured in milliseconds. By analyzing the video of several speech/sketching groups whose overlap difference was very small, 50 milliseconds was determined to be a reasonable threshold to use for calling two events simultaneous. The video was recorded at 30 frames per second which is approximately one frame every 33 milliseconds.

The graphs in Figure 3-13 and Figure 3-14 illustrate the overlap between the speech and sketching events groups in the data. The x-axis is the time in milliseconds that the start of the sketching preceded the start of the speech. A negative number means that the speech preceded the sketching. Similarly, the y-axis represents the number of milliseconds that the end of the sketching preceded the end of the speech. A negative number here means that the sketching ended after the speech. The words in the corners of the graph provide a visual depiction of the overlap of the speech and sketching in that quadrant.

Figure 3-13, depicting time differences for the word groups, shows that in most cases (71%), the sketching precedes the word spoken; these data points are in the right half of the graph. The plot has few groups (2%) in the upper-left quadrant, i.e., very few instances of speech that starts first and ends last. Only 20% of the data points are in the lower-left quadrant, i.e., speech that starts and ends first. The graph further illustrates a dense cluster in the upper right. This represents groups where sketching events precede the speech but the speech ends after the sketching. The data is also tightly clustered near the origin; this shows that sketching occurred temporally near the speech that referenced it.

The results for the *word groups* match the results reported by [49]. They reported that 57% of the time writing preceded speech (our data shows 71%). The most frequent overlap category they had was sketching starting first and ending first; this was also our highest category for the word groups (55%).

The overlap that occurred in the *phrase groups* was also examined, as shown in Table 3.1 and Figure 3-14. The phrase plot shows a different relationship from the word plot. Most of the data points are in the left half of the graph (82%), representing

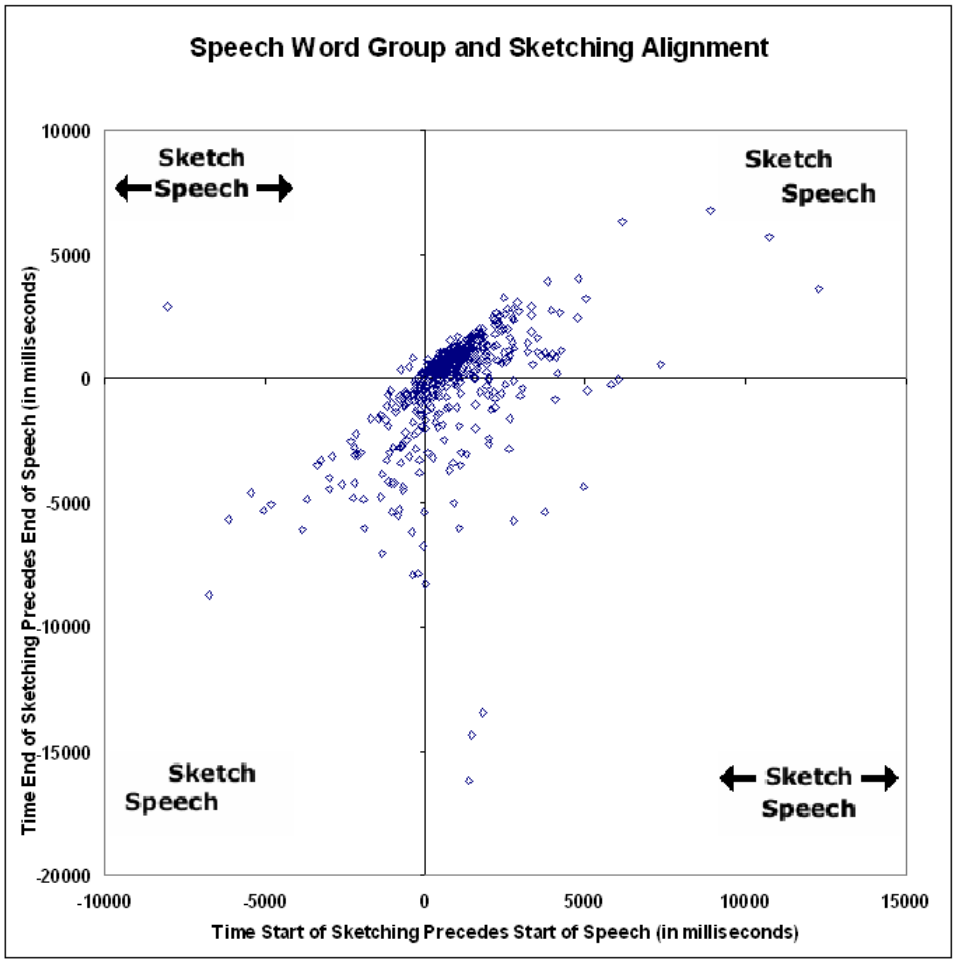


Figure 3-13: A graph depicting the time differences between the start and end times of the speech and sketching in each word group. The x-axis is the time in milliseconds that the start of the sketching preceded the start of the speech. The y-axis represents the number of milliseconds that the end of the sketching preceded the end of the speech. The words in the corners of the graph give a visual depiction of the overlap of the speech and sketching in that quadrant.

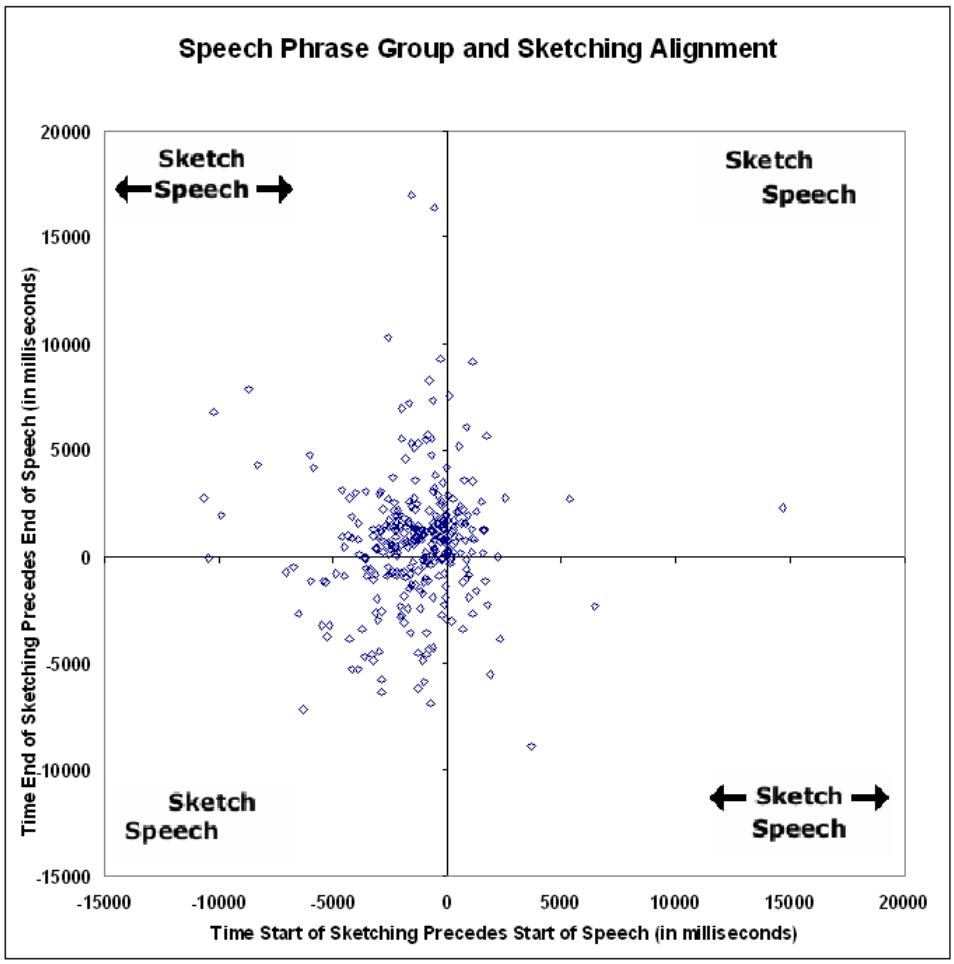


Figure 3-14: A graph depicting the time differences between the start and end times of the speech and sketching in each phrase group.

phrases in which the speech preceded the sketching. Further, many of the data points are in the upper-left quadrant, representing phrases in which the speech started before the sketching and ended after it (51%).

This is the opposite of the data reported in [49], which reported that sketching usually preceded the speech. There are several possible explanations for this difference. Their study looked at users sketching on an existing map, however, our study examined users drawing on a blank page. Our users explained the function of the various parts of their designs – something that doesn't happen when locating places on a map. Also, [49] used a Wizard-of-Oz study, so the participants were talking to a computer instead of a person across the table. The interactive conversation in our study could also have had an effect on the timing of the type of speech and sketching data that was observed.

We tested whether the mean of the difference between speech onset and sketch onset in Figures 3-13 (word group data) and 3-14 (phrase group data) was statistically different from zero. The word data mean different is 795 ms and is significant ($t(495) = 9.93$, $p<.01$); likewise, the phrase data mean difference is -1427 ms and is significant ($t(313) = -10.7$, $p<.01$).

3.4 Implications for MIDOS

Several results from the dialogue user study have important implications for MIDOS, in particular, pen color, speech characteristics, and complex replies to questions.

This study revealed that pen color is important in interpreting the user's intention. Pen color was used in the sketches for several purposes:

- to refer back to existing parts of the sketch or link parts of the sketch together as illustrated in Figure 3-8,
- to indicate a new topic as shown in the red and blue current paths in Figure 3-6,
- to reflect real world colors of objects.

The importance of color changes provides evidence that MIDOS needs to attend to color changes by both the user and the computer. However, our evaluation study of MIDOS (Chapter 8) revealed that user color changes were less prevalent when ink was automatically erased after each question.

The speech observations from the dialogue study echo the findings in our first study. First, the participants' speech was disfluent, especially when they appeared to be thinking about what to say. Second, the responses to questions reused some of the vocabulary contained in the question. Finally, concurrent speech and sketching always referred to the same objects, we call this *cross-modality coherence*. This last observation is particularly relevant for MIDOS because it provides an interpretation for simultaneous input from different modalities.

Interesting answers resulted from the questions posed by the experimenter in the study. Although the questions were simple, they produced lengthy, in-depth replies that went beyond simply answering the question. The participants also revised the sketches in response to the questions to make corrections or clarifications. These observed responses suggest that engaging the user in a conversation will do more than just resolve uncertainties in the physics simulation; we hypothesize that asking the user questions will engage them more deeply in the sketch and help them correct errors or clarify the design.

Chapter 4

MIDOS: An Overview

Several key results and observations from the multimodal input study (Chapter 2) and the dialogue study (Chapter 3) guided the development of our Multimodal Interactive DialOgue System (MIDOS). In particular, the utility of even simple questions, the concurrent nature of user's speech and sketching (cross-modality coherence), pen color changes, and extensive user replies hinted at some of the benefits of a multimodal dialogue system. The goal for the system is to provide an easy and natural way for the user to convey key information to the system. Instead of having the user try to guess what information the system needs, MIDOS engages the user in a dialogue to acquire the information necessary to proceed with a simulation, while leveraging the system's knowledge to pick reasonable questions to ask.

The initial domain for MIDOS is simple mechanical devices, similar to Rube-Golderberg machines. Objects in the domain include bodies, springs, pulleys, weights, pivots, and anchors. The combination of the user input and computer output creates a two-way multimodal dialogue as both the user and computer use speech and sketching. A benefit of the interaction with the user is that the system can take advantage of the user's knowledge; the system need only generate sensible questions and the user can aid the simulation by answering them. Instead of having to simulate the entire device at once, the system can generate a question and have the user supply more detailed physics knowledge about the next state of the device.

4.1 User Study Result – Simple Questions, Long Answers

An important and interesting result from the dialogue user study was that simple questions could elicit complex, ungrammatical, detailed, and long responses. These long responses contrast with the simple, short answers in many current dialogue systems. Although understanding the entirety of such responses is beyond the capabilities of current speech recognition and natural language systems, the ability to capture this information would be quite important. In the future, this information will allow a system to capture the rationale for designs. The rationale could then be recalled when analyzing a design or referencing the design process at a later time.

4.2 User Study Result – Pen Color

Participants in the studies used the different colored pens and highlighters when describing devices, changing color when switching topics or to indicate correlations between different parts or different perspectives. This feature is replicated in MIDOS, both as something the computer can do in its output and as something the user can do with their input. However, the color usage in MIDOS is less frequent than in the study because the computer can easily erase ink in MIDOS– part of the reason that users may have switched colors in the study is to differentiate two concepts without having to erase any ink.

4.3 User Study Result – Cross-Modality Coherence

Another key aspect of the studies was the tight integration of the user's speech and sketching. MIDOS should expect cross-modality coherence and should be able to generate output that also exhibits this property. In other words, the output of MIDOS needs to carefully integrate the sketching and speech modalities.

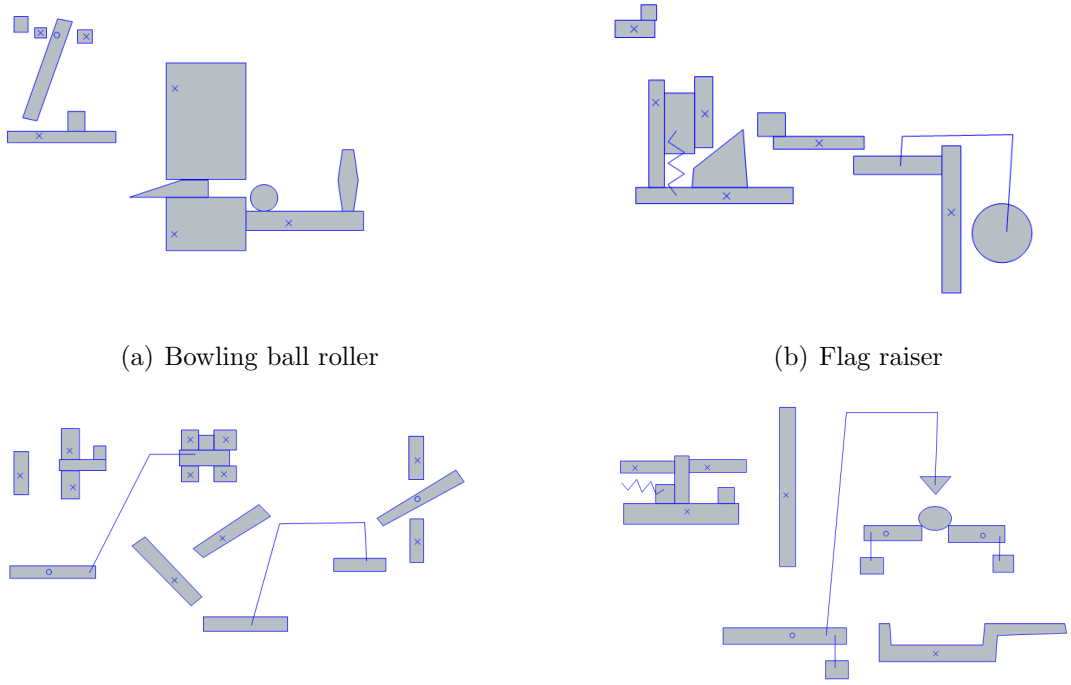
4.4 Example Devices

MIDOS can have conversations about mechanical devices, such as the four pictured in Figure 4-1 that were used in the evaluation user study. Figure 4-1(a) is a bowling ball roller. The block at the top-left falls, setting into motion a chain of actions that cause the horizontal pin to hit the bowling ball that in turn causes the bowling pin to fall down. The block at the top-left in Figure 4-1(b) is pushed to the right and falls onto the spring. This eventually causes the other block to slide to the right and onto the left block of the pulley. The right side of the pulley goes up and raises the circular flag. The platform at the top-left of the switch flipper in Figure 4-1(c) is moved to the left. The block at the top of the device falls down and hits the rotating platform. The platform rotates pulling the middle stopper to the left. This allows the other block to fall and slide down the ramps. Eventually the other block pushes the left side of the pulley down, causing the right side of the pulley to move upward and flip the switch. Finally, Figure 4-1(d) shows an egg cracker. The stopper on the left is pulled up, then the spring pushes the block at the edge of the platform. The block falls down causing the platform to rotate counter-clockwise. This in turn causes the triangular knife to move downward pushing the egg into the frying pan.

4.5 MIDOS Goals

As discussed in previous chapters, there is information about a design that is not supplied by the user in her initial sketch. The information could include properties of the device or a missing component. Traditionally, because systems cannot ask questions, the user must think of this information ahead of time and provide the information all at once. Accurately describing the details of a device in this way is error-prone and awkward.

We propose a different approach. The user provides a first approximation of the device, and then the system determines the information it needs and asks the appropriate questions. This approach requires less cognitive overhead and is both



(a) Bowling ball roller

(b) Flag raiser

(c) Switch flipper

(d) Egg cracker

Figure 4-1: Four devices that MIDOS can discuss.

more natural and more like discussing the design with another person.

One of the goals of MIDOS is to offer interaction that replicates, as closely as possible, a design discussion with another person. MIDOS currently is applied to the domain of simple mechanical devices, however, the techniques described in this thesis are also applicable to other domains. In particular, it is applicable to domains that have ambiguity in them as well as graphical, verbal, and dynamic elements. The system needs to be able to both determine the information it requires and generate appropriate questions to ask the user.

Several new issues are raised by multimodal interactive dialogue, including the timing necessary for the smooth integration of the outgoing speech and sketching, and the semi-persistent nature of the ink drawn on a shared drawing surface. MIDOS creates a novel interaction by combining the speech and sketching modalities in a symmetric interaction, i.e., both participants communicate multimodally.

A long-term goal of the multimodal interactions like the one created in MIDOS

is for people to interact more naturally with computers. Our hypothesis is that as the response from the computer is more human-like, humans will be more willing to provide detailed, rich, answers to the system.

4.6 MIDOS Components

MIDOS is built from several components: input acquisition, output synthesis, and the core components, which include the user interface, qualitative physics simulator, question selection, and dialogue control components. These components are described briefly here, and in more detail in the subsequent chapters. Figure 4-2 illustrates how these components are connected.

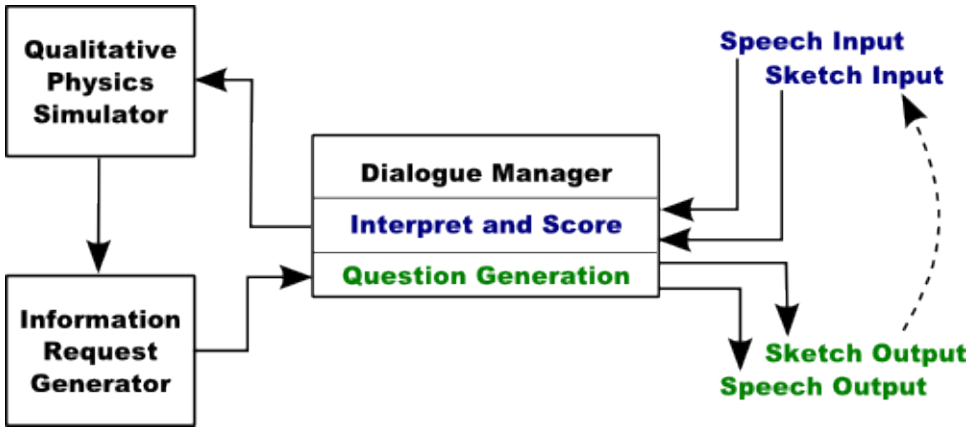


Figure 4-2: An overview of the MIDOS components and how they are connected.

4.6.1 Input Acquisition

MIDOS has two input modalities: speech and sketching, acquired separately using the Microsoft Speech Recognizer and a sketch recognition framework developed by other members of the Multimodal Understanding Group [51]. Each of the recognition systems returns an n-best list of interpretations. The results are combined with the possible expected responses from the user. Then MIDOS selects one of the possible interpretations. This process is described in detail in Chapter 5.

4.6.2 Output Synthesis

The output modalities in MIDOS are likewise (synthesized) speech and (synthesized) sketching. Speech is generated using the AT&T Natural Voices Speech Synthesizer; synthesized strokes are generated with a synthesizer developed for MIDOS. The observations from the studies revealed a tight integration and coordination of participant's speech and sketching. The dynamic nature of the questions that MIDOS asks and the varying size and position of the shapes in the sketch necessitate different outgoing strokes and different timing of the output modalities for each instance of a question. Replicating the tight integration and cross-modality coherence in the synthesized output, while accommodating the dynamic timing of the output, required the creation of a language to describe the relationship between the speech and sketching. The details of the output synthesis are discussed in Chapter 6.

4.6.3 Core Components

Several core components of MIDOS bridge the gap between input acquisition and the output synthesis: the user interface, the qualitative physics simulator, the question selection, and the dialogue control. These components are discussed briefly here and in more depth in Chapter 7.

User Interface

Figure 4-3 shows the MIDOS user interface. Based on the observations about color changing in the dialogue user study, several pen and highlighter colors are available to the user. The bottom of the window displays the computer's outgoing speech and the user's recognized speech. The interface is written in C# and integrates the input acquisition and output synthesis components. The data is sent to or from the rest of the core system which is written in Java.

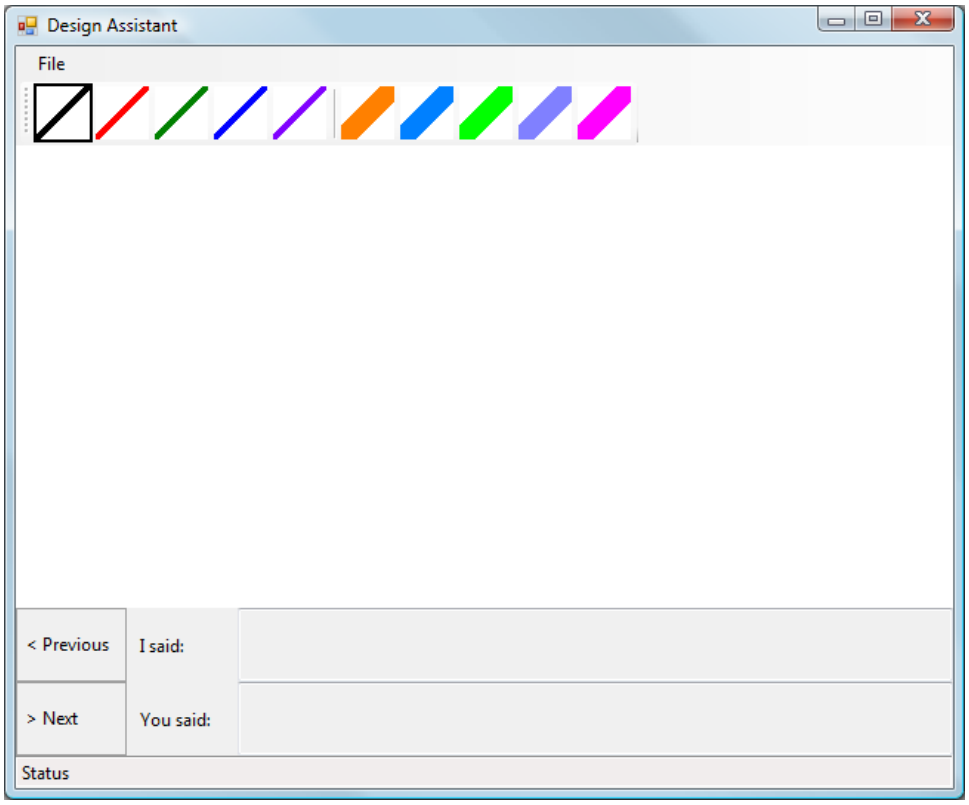


Figure 4-3: The MIDOS user interface.

Qualitative Physics Simulator

The physics simulator acts as a kind of inference engine, taking the current state of the world and trying to predict the next state. When the next state cannot be determined unambiguously, the system creates a set of *information requests* that are eventually turned into questions. The user's answers to these questions provide data used to update the system's physics model, allowing it to continue simulating the device.

Question Selection

One of the many information requests that the physics simulator generates must be selected and turned into a question to ask the user. The question selection component of the system accomplishes both of these tasks. The next information request is selected based on the types of the potential requests and recently requested information. After a particular request has been selected, a question is formed and sent to the output synthesizers.

Dialogue Control

Dialogue control handles the high-level functions of MIDOS. For example, the dialogue control is responsible for stopping the outgoing modalities if the user interrupts the system. In addition, this part of the system processes the result of the user input. This includes everything from sending an acknowledgment to the user to updating the state of the physics model to determining that the user hasn't provided enough input and a follow-up question is needed.

Chapter 5

Multimodal Input Acquisition

Users can communicate with Midos using sketching and speech. This chapter describes the acquisition process for each input modality and the procedure for combining them.

5.1 Speech Recognition

Speech recognition systems vary along several dimensions including amount of training, dictation performance, user input constraints, and ease of integration. Midos needed a speech recognition system with limited training, good performance in dictation mode, flexibility in allowed user input, and easy integration with the rest of the system. Based on these parameters, the Microsoft speech recognizer was chosen. It is easily integrated with the other user interface code in C#, requires little training for the user, and provides reasonable recognition results without limiting the user's speech.

The speech recognizer is used in dictation mode, which does not force the user to conform to a predefined grammar. Dictation mode allows the user to say anything she wants, presenting a difficult task for the speech understander. Midos takes a measured approach to this problem by focusing on matching the user's speech to an expected utterance. It requires that the user's speech roughly match one of these preset phrases. This tradeoff restricts the user less than a grammar would, and frees

MIDOS from having to do extensive natural language understanding. As a consequence, MIDOS cannot understand everything that the user might say such as speech containing negations or complex utterances (see Chapter 10 for more discussion). All of the expected utterances are listed in Appendix A.

The speech recognizer returns a ranked n-best list of possible spoken utterances. MIDOS then calculates a score for the utterance by comparing the spoken speech with the list of possible speech phrases that the system expects in response to the question it asked. The scoring metric is a percentage calculated as:

$$MatchPercentage = \frac{MatchedWords - 0.5 * ExtraWords}{ExpectedWords}$$

where $MatchedWords$ is the number of words in the user's speech that matched words in the expected speech and $ExtraWords$ is the number of words in the user's speech that do not match any word in the expected speech excluding the words: the, a, in, to. $ExpectedWords$ is the number of words in the expected speech utterance. Note that the formula penalizes missing words as they are excluded from $MatchedWords$.

The entries in the n-best list of recognized speech are first given a base score according to the position in the n-best list. The top entry is given a score of 10, and each subsequent entry's score is reduced by 1. The match percentage is calculated for each entry in the n-best list and multiplied by the entry's base score. The score is scaled so that it is between 0 and 1000, to match the scoring metric used for sketching. This process is repeated for each expected speech utterance, and the best overall score is kept. Two examples of n-best lists and the score calculations are shown in Table 5.1 and Table 5.2.

5.2 Sketch Recognition

Sketch data is most effectively captured in C#, by using the Microsoft Tablet PC API to get high sample rates and pressure data. Each stroke the user draws is captured as a series of points, and each point contains the x and y coordinates in himetric units,

Recognized Speech	Base Score	Match Percentage	Scaled Score
it moves in this direction	10	1.0	1000
it moves this direction	9	0.8	720
it moves and this direction	8	0.7	560
it moved this direction	7	0.5	350
it moves on this direction	6	0.7	420
it moves its direction	5	0.5	250
it moved its direction	4	0.2	80
it moves his direction	3	0.5	150
it moved his direction	2	0.2	40
it owns this direction	1	0.5	50

Table 5.1: The n-best list from the speech recognizer matched against the expected phrase "It moves in this direction."

Recognized Speech	Base Score	Match Percentage	Scaled Score
No	10	1.0	1000
no	9	1.0	900
know	8	0.0	0
noe	7	0.0	0
new	6	0.0	0
noh	5	0.0	0
knew	4	0.0	0
nau	3	0.0	0
dough	2	0.0	0
doe	1	0.0	0

Table 5.2: The n-best list from the speech recognizer matched against the expected phrase "No."

the timestamp, and the pressure.

Sketch recognition is handled by a low-level stroke recognizer developed by our group [51] that classifies the stroke into one of several primitive shapes (e.g., lines, arcs, ellipses, and polylines). MIDOS adds a higher level classification of these primitives, allowing it to recognize strokes as either a location, path, or selection. A location indicates a point on an object or a new position for an object. A path shows how an object moves (e.g., a line that indicates a distance or direction, or an arc that indicates a direction or distance of rotation). A selection identifies a particular object (by circling, marking, or filling in an object). Figure 5-1 shows an example of each type of stroke.

(a) Location (b) Location (new position) (c) Path (d) Selection

Figure 5-1: Illustration of the three different types of input strokes.

The scoring metric used for sketching is similar to the scoring metric for speech. Like the speech recognizer, the sketch recognizer returns an n-best list of interpretations, scored in the range (0-1000). MIDOS combines these scores with the sketching it is expecting based on the question it asked. Each expected stroke has a type of stroke and optionally a target shape. If the drawn stroke matches one of the expected strokes, it keeps its base score. If it doesn't match it gets a score of zero. If it matches some of the expected shapes, it gets a score equivalent to the base score times the percentage of shapes that it matched. The expected strokes are listed in Appendix A.

5.3 Combining Inputs

The user's speech and sketching potentially overlap temporally and in content. The first step in figuring out what the user intended is to find corresponding speech and sketching. The user studies we conducted provide two key insights about segmen-

tation: concurrent speech and sketching are typically about the same topic (cross-modality coherence) and a pause indicates a new topic. The system uses these heuristics to group concurrent speech and sketching together; concurrent speech and sketching are assumed to be about the same topic. Furthermore, no input is processed while there is ongoing speech or sketching. MIDOS waits until there is a pause of 300 milliseconds in both input modalities before it attempts to process the user's input.

After the system has grouped some speech and sketching, it must determine the user's intention. This task is guided by two assumptions. First, we assume the user is answering a question posed by the system. Second, as mentioned above, the system assumes the answer will fall within a known variety of possibilities for each modality. Although the user's answer does not have to exactly match any expected answer, the expected answers help the system interpret the user's response and determine whether it is valid. The user's multimodal input is processed in seven steps:

1. Ask the user a question.
2. Group user's speech and sketching as described above.
3. Match and score the user's speech against the expected speech.
4. Match and score the user's sketching against the expected sketching.
5. Maximize the sum of speech and sketching scores.
6. Evaluate the best scoring combination, checking for missing or conflicting input.
   (a) If the combination is successful, go to the next step.
   (b) If the combination is unsuccessful, ask the user a follow-up question with more guidance about the expected answer. Return to the first step.
7. Generate statements based on the new information and update current state and the physics appropriately.

We illustrate these steps with a simple example of a block connected to a spring, Figure 5-2.

First, the system runs the physics simulator and determines that the situation is ambiguous (is the spring in tension or compressed?). Next, it generates an appropriate question for the user: "Will this spring expand or contract?" (Figure 5-3). The generation of the speech and sketching output is discussed in the Chapter 6.

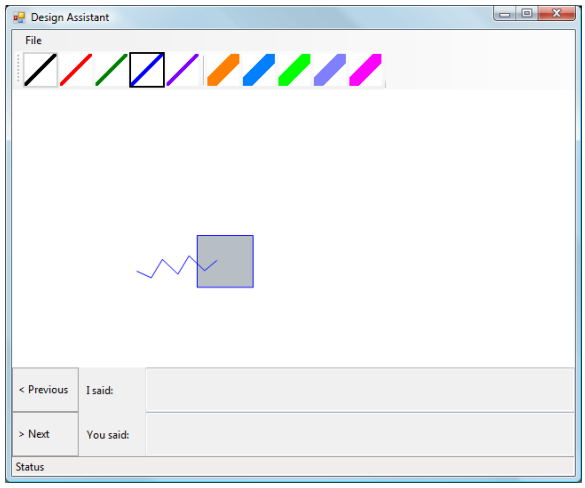


Figure 5-2: The initial configuration of the block and the spring.

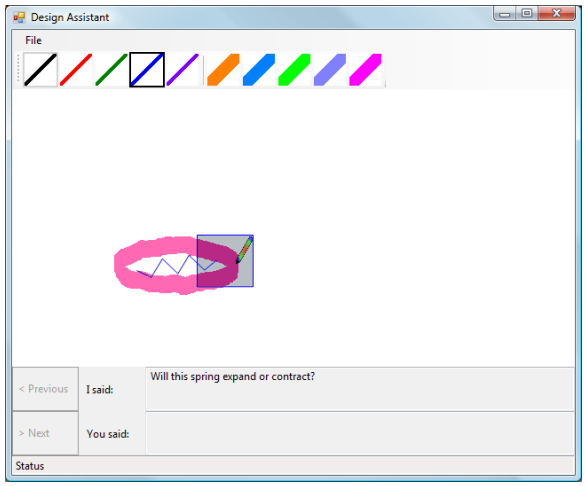


Figure 5-3: The system asks the user which direction the spring is going to move in: "Will this spring expand or contract?"

The system knows to expect several possible responses from the user: The user might say "it expands," "the spring gets longer," or she might simply draw a line to indicate the direction in which the spring moves. Alternatively, she might combine speech and sketching and say "it moves in this direction" while drawing a stroke. The

response from the user could contain only speech, only sketching, or a combination of both. The system anticipates and understands any of these possibilities.

After scoring both the speech and sketching input, the system next evaluates cross-modal consistency. For example, is the line indicating a direction for the spring consistent with the speech utterance? Table 5.3 visually summarizes possible results from the consistency check; the possibilities are explored in more depth below.

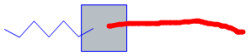
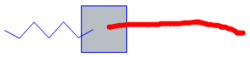
User Sketching	User Speech	Consistency Check Result
	"It moves in this direction"	Insufficient
	"It contracts"	Conflict
	"It expands"	Success

Table 5.3: A visually summary of possible consistency check results.

Figure 5-4 shows an example in which the user drew a stroke indicating the spring compresses, while saying "it expands." The system noticed this inconsistency and asked a follow-up question to resolve the inconsistency. As shown in Figure 5-5, MIDOS said "I could not understand your speech and sketching. Does the spring expand, contract, or is it at rest?"

Replies may also be insufficient or at odds with what the simulator knows about the physics of the situation. An insufficient reply arises if, for example, the user says "it moves in this direction" without drawing a stroke (Figure 5-5). Without a stroke the speech cannot be translated into a direction.

A reply that is consistent across modalities is illustrated in Figure 5-6.

The user's input may also make sense on the surface, but the underlying physics is impossible or at least impossible for our physics simulator to compute. This can occur, for example, if the system asks the user where a collision point is located on a body and the user indicates a point that cannot be the first collision point, as illustrated in Figure 5-7.

The result of the cross-modality consistency checking is used to update the state

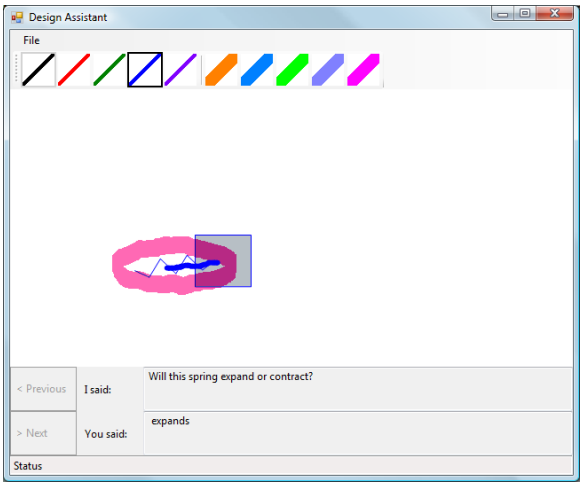


Figure 5-4: The user provides a conflicting answer by drawing the shown stroke and saying "It expands." Note that the UI displays the best result from the speech recognizer, in this case it displays "expands."

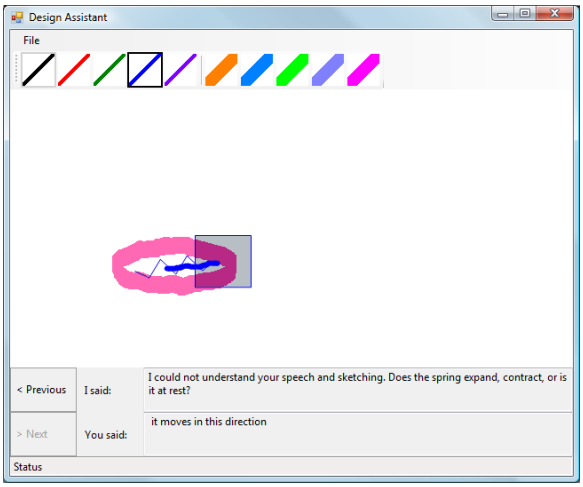


Figure 5-5: The user provides an insufficient answer to the computer's question. The computer asked: "I could not understand your speech and sketching. Does the spring expand, contract, or is it at rest?" This time the user answered, "It moves in this direction," but did not draw a new stroke. The blue stroke was drawn in response to the previous question (Figure 5-4).

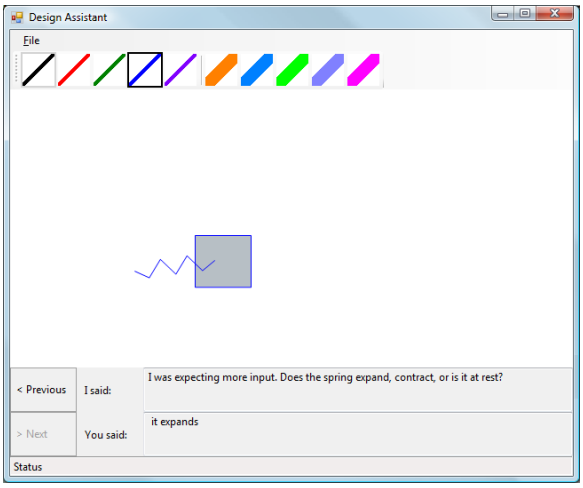


Figure 5-6: The user provides an acceptable answer by saying "It expands." The system then updates the velocity of the body accordingly and removes the stroke that was used in the question.

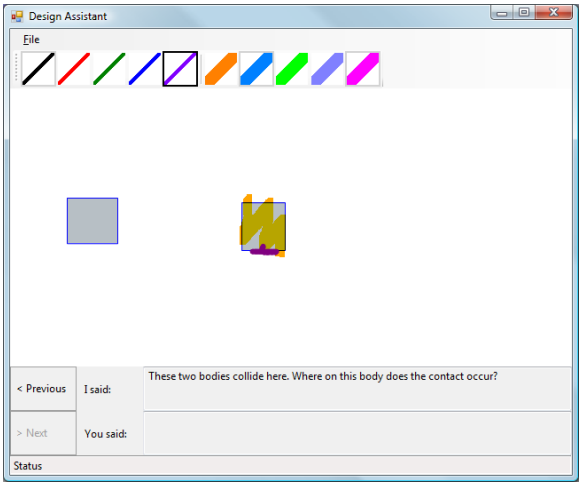


Figure 5-7: The two bodies pictured here are moving towards each other. MIDOS asks the user where the contact occurs on the block highlighted in orange. With the purple pen, the user indicates the bottom face which is a physically impossible location for the collision.

of the system, in this case setting the force exerted by the spring on the block in the appropriate direction. In other cases, properties of the stroke itself are used to compute the update, for example, the angle of a stroke may be used to update the velocity direction of a body. Each information request is responsible for processing the set of corresponding speech and sketching input. Chapter 7 describes information requests in more detail.

5.3.1 Matching the Input

The examples above illustrated the different possible outcomes for combinations of speech and sketching input. The expected response is different for each information request. The system handles the general case by comparing the inputs it receives with a table of possibilities. Table 5.3 provides a visual summary of matching process.

Each line of the table indicates which of the expected strokes and speech must be present, not present, or have a specific value. The specific values are calculated based on the speech or the sketching in the user's input. These requirements can also be marked as optional or required. Each row in the table indicates the outcome of matching that row: success, insufficient, conflict, or other. For stroke input, the information request can specify a calculation, the result of which can be used in the table.

The table for the question about whether a spring expands, contracts, or stays stationary is shown in Table 5.4. For this question, the value computed for the stroke input corresponds with whether the stroke indicated an expanding (positive value), contracting (negative value), or stationary (0.0 value) state for the spring.

If the result of matching the user's input is not "success," the system asks a follow-up question that indicates explicitly the type of answer it is expecting. If possible, it tries to specify exactly what was missing from the answer. Figure 5-6, for example, shows the computer's response when the user's answer is missing some important piece of information: "I was expecting more input. Does the spring expand, contract, or is it at rest?" If on the other hand the match is successful, a statement or set of statements are created that capture the information. The physics simulator then uses

these statements when it calculates how to update the simulation. In our example, the force of the spring would be updated so that the spring expands.

Expands Speech	Contracts Speech	Stationary Speech	Multimodal Speech	Direction Stroke	Result
Not Present	Not Present	Not Present	Present	Not Present	Insufficient
Present	Present	Optional	Optional	Optional	Conflict
Optional	Present	Present	Optional	Optional	Conflict
Present	Optional	Present	Optional	Optional	Conflict
Optional	Optional	Present	Present	Optional	Conflict
Not Present	Present	Optional	Optional	Positive Value	Conflict
Not Present	Present	Optional	Optional	0.0	Conflict
Present	Not Present	Optional	Optional	Negative Value	Conflict
Present	Not Present	Optional	Optional	0.0	Conflict
Not Present	Not Present	Present	Optional	Positive Value	Conflict
Not Present	Not Present	Present	Optional	Negative Value	Conflict
Not Present	Present	Not Present	Optional	Optional Negative Value	Success
Present	Not Present	Not Present	Optional	Optional Positive Value	Success
Not Present	Not Present	Present	Optional	Optional 0.0	Success
Not Present	Not Present	Not Present	Optional	Positive Value	Success
Not Present	Not Present	Not Present	Optional	Negative Value	Success

Table 5.4: The full table of expected inputs for a question about the direction a spring moves. Entries not marked "optional" are required.

The combination of the low-level recognizers, our matching and scoring functions, and our consistency checking table allow the system to determine the user's intended behavior.

5.3.2 Disfluencies

Our research has shown that disfluencies play an important role. For example, disfluencies in the speech input seem to indicate that the user is still thinking about

the same topic. MIDOS handles disfluencies the same way it handles other speech input; they are recognized as incoming speech and cause the system to wait before processing of the user's input. The delay gives the user more time to finish her speech and sketching about the current topic, thus achieving the desired result. Ideas for more detailed handling of disfluencies are discussed in Chapter 10.

Chapter 6

Multimodal Output Synthesis

Output synthesis can be thought of in two parts: synthesizing the individual outputs and joining the parts together. The individual modalities can be thought of as instruments in an orchestra score. The score is not complete without synchronizing the different instruments, or in this case synchronizing the different output modalities. We created a simple language to easily write a multimodal score.

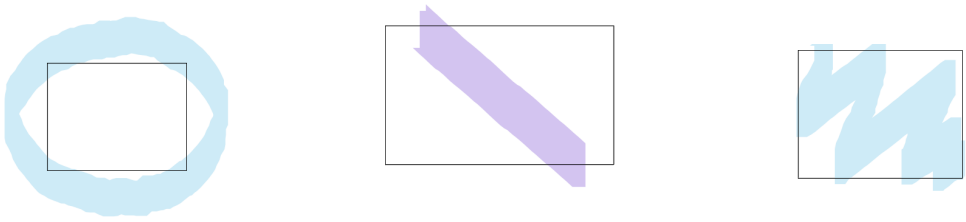
As one of the goals of MIDOS is to create a symmetric interaction, each of the computer-generated modalities should replicate the equivalent user modality as closely as possible. Additionally, users frequently combine speech and sketching when responding to questions about designs. The computer should replicate this behavior too.

In Chapter 5 we discussed how the input to MIDOS is handled. This chapter explores the generation of both output modalities and the process MIDOS uses to combine them to form the complete output.

6.1 Sketch Synthesis

In the same way that users identify objects in a sketch by highlighting, circling, or otherwise marking the object being discussed (see Chapter 3), computer-generated output also needs to graphically identify objects in the sketch. The sketch synthesizer used to identify objects takes several different approaches, each of which mirrors the

user's methods of identifying areas: circling it, drawing a stroke through it, or filling it in (see Figure 6-1).



(a) Circling (b) Stroke (c) Filling in

Figure 6-1: Three methods of identifying areas.

Identification strokes like these are straightforward to generate. The circling stroke is created by constructing an ellipse that encloses the bounding box of the shape. A single stroke through a shape is generated by making a randomly oriented line through the center of the bounding box of the shape and using the longest portion of that line that is contained within the shape. Finally, filling in a shape is accomplished by starting at the top-left corner of the shape and drawing connected line segments at alternating angles through the shape.

The remainder of this section discusses the more complex aspects of synthesizing the sketched portion of the output, including selecting an identification method, adjusting the strokes so they appear natural, and generating output with the appropriate properties.

6.1.1 Selecting an Identification Method

MIDOS takes several factors into account when deciding which identification method to use: applicability to the situation, system preference, and drawing time. The drawing time of the strokes for the different identification methods is particularly important. Preferably, the strokes should be drawn within the amount of time that the generated utterance takes to speak so that pauses in the utterance are not required. The speech synthesizer produces more natural sounding speech when it synthesizes

speech in longer phrases. We have discussed how users closely coordinate their speech and sketching input and pause the appropriate modality to keep the modalities synchronized; the synthesized outputs should do the same.

By default, the different identification techniques are ordered in decreasing precision: filling in a shape, drawing a single stroke through a shape, and circling a shape. Filling in a shape is the most precise method, and circling a shape is the least precise. The full set of techniques is not applicable in every situation. In such cases, only the applicable selection techniques are used. Some shapes, like springs, cannot be filled in or marked by a single stroke, so are always circled. Computer-generated questions may also refer to a region of space that does not have an underlying shape or object, for example, the region where a collision will occur. These areas are also circled. It is always possible for the system to generate a stroke that surrounds an object or area; therefore, circling serves as the default method. Although circling is a good default, it is imprecise; the size of the enclosing ellipse often vastly exceeds the size of the shape or region. The ellipse is generated using the bounding box of a region and does not closely track the border of a region. This potentially causes the ellipse to enclose areas that are not part of the target region. Each requested identification stroke can specify list of preferred techniques. For each technique, an appropriate stroke is generated, and then MIDOS determines which stroke should be used.

MIDOS uses two factors to choose the identification stroke to use: the position in the preference list, and the amount of time (if any) that drawing the stroke exceeds the duration of the speech associated with it. The penalty for using a particular stroke is calculated by this formula:

$$Penalty = Max(0, Stroke_{TIME} - Speech_{TIME}) + 250(P - 1)$$

where $Stroke_{TIME}$ is the time in milliseconds that the stroke takes to draw, $Speech_{TIME}$ is the time in milliseconds that the speech takes to say, and P is the stroke's (1-based) position in preference list. If the stroke is not in the preference list, P is set to 5. The determination of $Stroke_{TIME}$ is discussed below. The weight

for the preference component of the formula was chosen by examining the *Penalty* value for several examples and comparing the value to the desired behavior. The identification stroke with the smallest penalty is used to identify the shape or region.

6.1.2 Timing and Adjusting Points

As previously discussed, a goal for the sketch synthesizer is that its output should appear plausibly humanly generated, as opposed to obviously machine generated. One aspect of the effort is to make sure that the computer strokes are drawn at the same drawing speed as user strokes and that the strokes appear in the same way as user strokes – point-by-point. Each of the identification strokes has a range of pen speeds that attempt to approximate the pen speed a person would use to draw the stroke. Our qualitative observations from the dialogue user study indicated that people draw fill strokes the fastest, followed by the circling stroke. The single stroke line was the slowest stroke users drew. The range of speeds allows MIDOS some flexibility in fitting the stroke into the duration of the associated speech.

Additionally, the path for each stroke is adjusted to introduce variations and errors to more closely match human-drawn strokes. This is a two-step process. First, the stroke is resampled so that the distance between points does not exceed a threshold. Some strokes initially have very sparse points; resampling allows the system to introduce variations while preserving the overall shape of the path. Second, the points are shifted by a random amount so that the lines are not perfectly straight and the curves are not perfectly round. Figure 6-2 illustrates the process of drawing a stroke and the variation that is introduced.

6.1.3 Color Selection

The user studies revealed that pen and highlighter color consistency is important. MIDOS picks the pen and highlighter color using the following criteria:

- Use the same color for the entire question.
- If the question type is the same as the last question asked, use the same color.

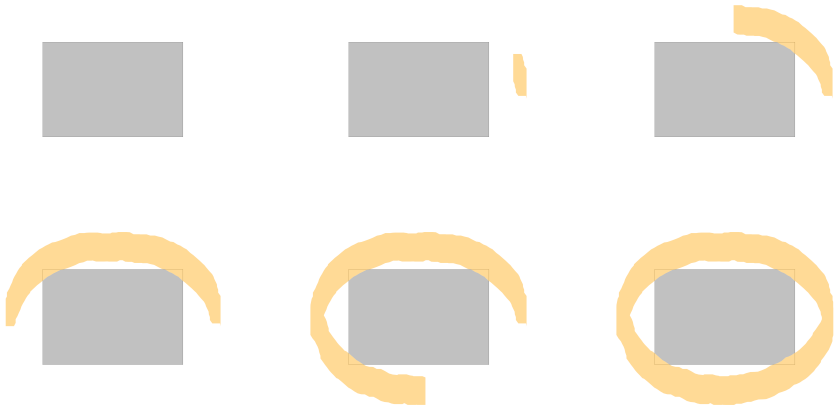


Figure 6-2: A body in the process of being circled.

- If the question type is not the same as the last question, use a different color.

- If a pen and highlighter are used in the same question, use contrasting ink colors.

6.1.4 Pie Wedges

In addition to strokes, the computer can also draw pie wedges to indicate a (sometimes) small range of possible angles to the user, as shown in Figure 6-3. For example, after a collision between two bodies, we need to know what direction the bodies move in (due to the qualitative nature of the physics simulator this cannot be calculated exactly). Specifying a range of angles by using a stroke proved to be difficult. Pie-shaped areas clearly indicate the allowable range of angles and seemed to be the best solution, although this aspect of the interaction is not symmetric because the user cannot draw pie wedges. Pie wedges appear at specified times like the strokes, but appear all at once instead of being drawn slowly. Fifty milliseconds is used as the duration of the pie wedges for the calculations that are used to align the speech and sketching.

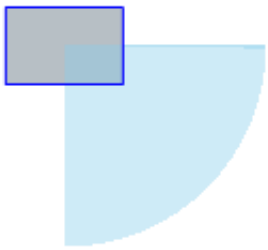


Figure 6-3: A pie wedge.

6.1.5 Pen Image

Some of the identification strokes happen in a short amount of time. In order to draw the user's attention to a particular computer-generated stroke, an image of a pen is displayed, follows the stroke as the stroke is being drawn, and stays on the screen for up to 2.5 seconds after the stroke is complete, as shown in Figure 6-4. The pen image is drawn at the location of the last point, or at the center of the pie wedges. In addition to drawing visual attention to the stroke, the image also alludes to the idea that the computer is drawing these strokes in the same manner that the user is drawing.

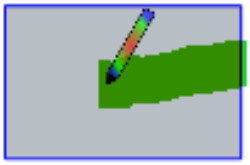


Figure 6-4: The pen drawing a stroke through a shape.

The pen stays visible for 2.5 seconds at the conclusion of drawing or until the pen is displayed at a new location with a new stroke. Future work (see Chapter 10) includes more detailed drawing considerations to more accurately and closely reflect human drawing styles and properties.

6.1.6 Motion Indicators

Some of the questions MIDOS asks are about potential events: possible collisions, motion distances, or rotation amounts. MIDOS indicates potential events using straight and curved arrows to show motion paths of objects as shown in Figure 6-5. These arrows serve as a primitive form of animation that enable MIDOS to convey the expected path of an object to the user. For example, the arrows allow MIDOS to show the user how the system anticipates bodies will collide or to show the predicted trajectory or rotation of a body. The arrow strokes, like the identification stokes, have a range of possible drawing speeds. The arrows are rendered almost as quickly as the fill strokes so that they do not delay the associated speech utterance.

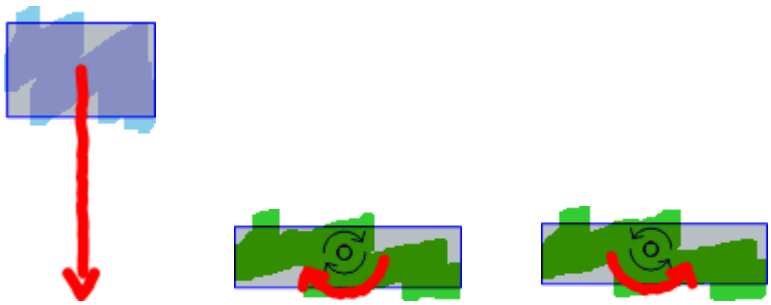


Figure 6-5: Arrows indicating a direction and rotations.

6.1.7 Technical Considerations

Although the C# Tablet PC API provides realistic and detailed rendered strokes in response to user pen input, the ability to programmatically generate rendered strokes on a point-by-point basis does not exist. Extensive bookkeeping code compensates for this by generating a series of strokes of increasing length while removing the previous shorter stroke. This allows the strokes to be displayed as if the computer were drawing them point-by-point, like the user draws. The display uses multiple layers of ink to support simultaneous drawing by the user and the computer. This allows the user to start drawing while the computer is drawing strokes.

The output that MIDOS creates is sent to the user interface all at once. It is then added to a set of time-release queues that add the points to the strokes (or send

the speech to the synthesizer) at the appropriate time. This allows the system to create the effect of the computer actually drawing the strokes, coordinated with the generated speech.

C# records the pen input points in himetric units with millisecond timestamps. Himetric units are used when the files are saved, but pixel coordinates are used by the physics simulator for its calculations. Additionally, the timestamps on the points must be adjusted to account for inaccurate timestamps. The timestamps that are originally assigned contain some points with duplicate times and unevenly spaced timestamps. The corrections are made as the data is received in C#.

6.2 Speech Synthesis

The synthesized speech is generated by an off-the-shelf synthesizer: AT&T Natural Voices. MIDOS sends the synthesizer a sentence or phrase at the desired output time and the synthesizer generates the corresponding realistic speech. Although generating the outgoing speech is easier than generating the outgoing sketching, more work is involved in timing the speech output correctly.

Ideally, the synthesizer would report its progress in the utterance, which would enable us to coordinate the speech and sketching output. However, this recognizer provides limited feedback: indicating only whether it is actively producing speech. To circumvent this issue, all expected output phrases are timed in an initialization step. Then the timing data is used to approximate the progress of the synthesizer. Details of the coordination with the sketching are discussed below.

6.3 Synchronizing Outputs

The previous sections have discussed the synthesis of the individual output modalities. Any nontrivial use of simultaneous sketching and speech requires synchronizing the two modalities. For example, the use of two deictic gestures in the same sentence ("Does this block hit this block?") is impossible without close coordination of the

two modalities.

The coordination of the different modalities is based on the observations from the user studies. In particular, the study participants would pause their speech or sketching to keep the two modalities synchronized, helping to ensure that the two modalities were always referring to the same topic and objects (cross-modality coherence).

6.3.1 Synchronization Language

Producing cross-modality coherence in MIDOS is difficult because the system has numerous questions it can ask, each of which can be applied to a very large number of physical situations. As a result we face a non-trivial problem in determining how much time is needed to draw any required strokes. The time depends on both the length of the stroke and the stroke speed, which itself depends on the type of the stroke and the accompanying speech. The variations in physical situation also mean that the timing constraints between the speech and sketching will vary between different instances of the same question. In order to specify how the speech and sketching should be coordinated in a manner independent of these variations, MIDOS uses a small language that expresses the chronological relationship between the system's speech and pen strokes. The small set of features in the language provide a rich set of possible outputs, while limiting the effort required to generate them.

The main capabilities of the language are:

- drawing one stroke with an accompanying phrase,
- drawing zero or more strokes with an accompanying phrase,
- pausing both modalities for a short or long period of time,
- deleting a single stroke or multiple strokes that have been displayed,
- automatically computing the timing of the modalities,
- automatically adjusting the grammatical number of the words based on the number of assigned strokes.

The syntax of the language is shown in Table 6.1. Parentheses and braces are used to indicate a phrase that should be concurrent with a stroke or a group of strokes, respectively. Also shown is the syntax for pauses and deletion of strokes. The short pause is 700 milliseconds, the long pause is 3 seconds. Several examples are shown in Section 6.3.4.

Function	Annotation
Associating a word or group of words with one stroke	( )
Associating a word or group of words with zero or more strokes	{ }
Short pause in the output	<short pause>
Long pause in the output	<long pause>
Clearing all the computer-generated strokes	<clear strokes>
Clearing the last computer-generated stroke	<clear stroke>

Table 6.1: The timing annotations for the speech and sketching output.

6.3.2 Pointing

MIDOS can "point" at objects by combining identification strokes and pauses. The identification strokes allow MIDOS to point out a shape or several shapes. A subsequent pause inserts a delay that allows the user to absorb the output.

The "pointing" can be temporary by using the deletion operations after the pause to remove the strokes. This leaves the display in a state where the same objects can be circled a second time during the same question. Clearing the initial strokes allows the system to focus the question on one or more shapes. An example of pointing is illustrated in Section 6.3.4.

6.3.3 Automatic Timing

Generating the timing for the combined output is accomplished using the following steps:

1. Break the speech string into fragments based on the braces and parentheses.

2. Calculate the number of strokes that should be assigned to each fragment.

3. Check to make sure there are enough strokes.

4. Assign the strokes to each fragment.

5. Adjust grammatical number of each fragment.

6. Determine the time needed for the speech in each fragment.

7. Determine the time needed for each stroke.

8. Determine the timing for each fragment, merging fragments where possible.

9. Determine the timing for the entire question.

For the purposes of these steps, pie wedges are counted as strokes. The steps are described in more detail below.

Allocating Strokes to Fragments

The first step is to break the speech string into fragments so that the strokes can be assigned to each fragment. The annotated speech string is broken into fragments, using the braces and parentheses as guides. Each fragment has either zero, one, or many (zero or more) strokes associated with it. The speech string is accompanied by lists of strokes, pie wedges, and deletions to assign to the fragments.

The strokes (and pie wedges) are assigned to the fragments in an iterative process. First, each fragment that requested exactly one stroke is assigned one stroke. The remaining strokes are assigned to the fragments that requested multiple strokes so that the strokes are as evenly distributed as possible. For example, each fragment is assigned one stroke before any fragment is assigned two strokes.

The strokes and speech are specified separately, so MIDOS performs two sanity checks on the fragments. It ensures that there are enough strokes to satisfy the total number of assigned strokes and that there are no unassigned strokes remaining.

A similar process is followed for the deletion annotations. Deletion strokes are assigned (to <clear stroke> and <clear strokes> requests) until there are no more deletion strokes available. A sanity check is made to ensure that the number of

deletion strokes requested by the language is not greater than the number of available deletion strokes and that there are no extra deletion strokes.

At this point, the number of strokes (and deletions) for each fragment is determined. The actual strokes (and deletions) then must be assigned to each fragment. The strokes are assigned based on their position in the list of strokes, with pie wedges added to the end of the list. Likewise, deletions are assigned based on their position in the deletion list. None of the current questions require mixing strokes with pie wedges in the same question. If a situation arose that called for mixing these types, the system could be modified to assign strokes and pie wedges to fragments from a single, merged list.

Automatic Grammatical Number Adjustment

Now that the strokes are assigned to the fragments, the grammatical number agreement (singular vs. plural) of the fragments can be updated. The grammatical number of the fragment actually affects a larger section of speech, because the verb tenses also need to be adjusted accordingly and may not be contained in the fragment with the stroke(s). Thus, if any fragment is plural, the entire set of fragments under consideration is changed to plural form. For example, the sentence "{This shape} causes a clockwise rotation." has two fragments, "{This shape}" and "causes a clockwise rotation." If there are multiple shapes that cause a clockwise rotation, there will be multiple strokes associated with the fragment "this shape." The first fragment must be updated to "{These shapes}," and the second fragment must be updated to "cause a clockwise rotation." In the future, a more careful analysis of which words should be updated will be needed, but this admittedly simplistic analysis was sufficient for the questions that MIDOS currently asks. This limitation can be circumvented by aligning smaller groups of speech and sketching at a time (parts of a sentence or question). MIDOS will consider each set of speech and sketching separately so the grammatical number of one set of inputs will not affect another set.

Timing Computation

After the grammatical number is used to adjust the form of the words, the exact timing for the speech phrase can be calculated. This step is important because there can be variations allowed in the timing of the sketching that depend on the speech timing being determined. Once the words in the speech are set, the time for the phrase can be looked up in the list of precalculated speech-timing information. This time is then used to determine how quickly the strokes will be drawn, as described in Section 6.1.

With the timing determined for the speech and the sketching, the overall timing for the fragment can be calculated. If the sketching can occur entirely within the duration of the speech, the fragment can be combined with the subsequent fragment. If not, then the subsequent fragment must be delayed until the sketching for the current fragment has concluded. If the sketching part of the fragment is shorter than the associated speech, the sketching is centered in the fragment.

After the timing for each fragment is determined, a similar calculation is used to compute the timing for the entire question. The speech sounds natural and flows smoothly if the synthesizer speaks an entire sentence at once. As many fragments as possible are merged together so that the speech that is sent to the speech synthesizer is in the largest units possible. The fragments of the question are timed relative to the start time of the entire question. When the system is ready to ask the question, the offsets allow the system to compute the absolute time for each question component by simply adding the current time to the offsets.

Time Estimation Accuracy for Phrases

The estimates for the length of time that the speech synthesizer will take to speak a particular phrase are not exact. Is it better if the time is over or under estimated? If this time is overestimated, the system might mistakenly believe that an associated stroke or strokes would be finished before the end of the speech phrase. If the speech is shorter than expected, these stokes would overlap with the next speech utterance. An

underestimate of the speech would instead cause the start of the next speech phrase to be delayed until the strokes were completely drawn. Although underestimating may cause the calculated start time of the next utterance to be too early, the system can only speak one utterance at a time, so this is not a problem.

6.3.4 Examples

The questions the system asks range from simple: "Do {these two bodies} collide?" to complex: "(These two) (bodies) collide (here.) <long pause> <clear strokes> Where on (this) body does the contact occur?" Both of those utterances are accompanied by strokes that identify the bodies in question, and, for the second question, the region where the collision occurs. The second example is illustrated in Figure 6-6.

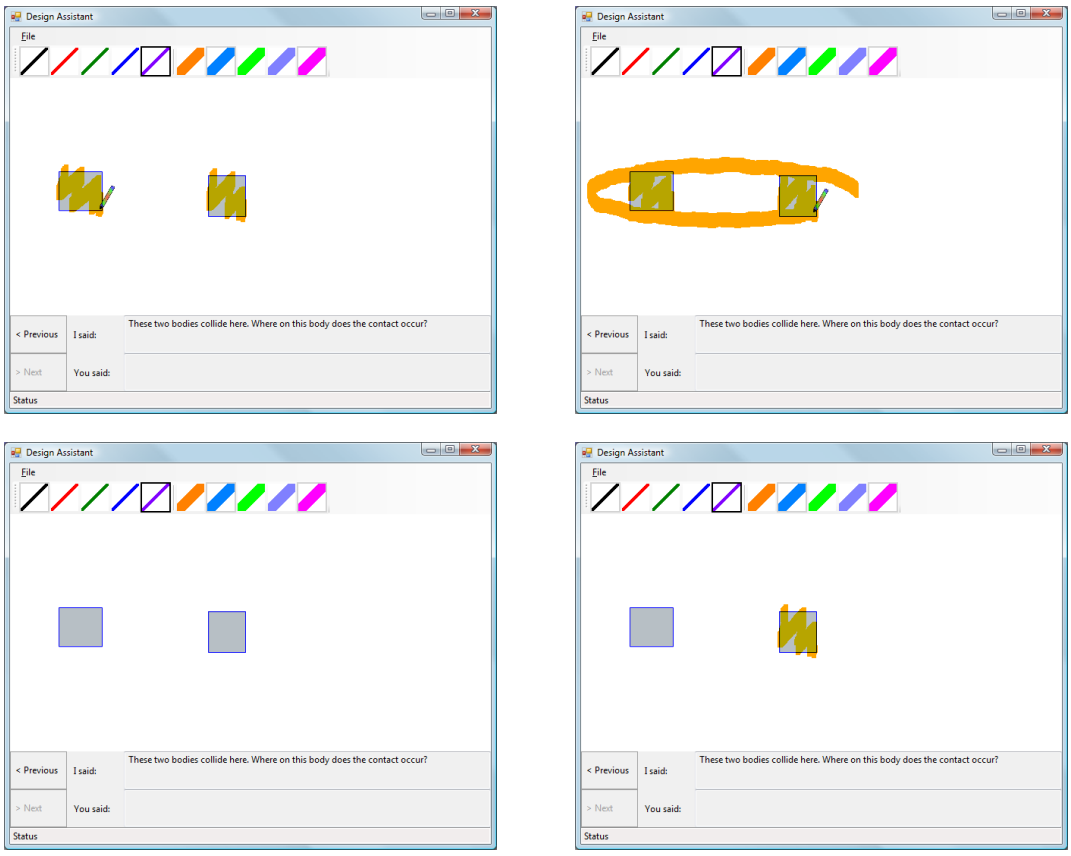


Figure 6-6: An example of the generated output for the question "(These two) (bodies) collide (here.) <long pause> <clear strokes> Where on (this) body does the contact occur?" Notice MIDOS pointing to bodies using identification strokes and deletion operations.

6.4 Interruption and Input Acknowledgment

MIDOS handles two other pieces of dialogue control. First, if the user starts to draw or talk, the system will halt its output. This allows the user to start to answer a question before the system has finished asking it. This proves to be especially useful with lengthy questions, some of which can take more than 15 seconds to finish.

The second piece of dialogue control is input acknowledgment. The system will acknowledge the user's input if the score for the user's input is above a threshold. This acknowledgment takes the form of a brief speech utterance to let the user know that the system is processing their input. The acknowledgment phrases include: "Got it," "I got it," "Uh huh," "Okay," and "Thanks."

Chapter 7

MIDOS: Core Components

Several core components of MIDOS tie together the input (Chapter 5) and output (Chapter 6). These core components include the user interface, the physics simulator, and the information request processing.

Figure 7-1 shows an overview of the components and their connections. The qualitative physics simulator analyzes the elements in the sketch, updates properties, and generates trajectories for the bodies. The information request generator analyzes the results of the simulation and generates a list of needed information. The topics of these requests include collisions, missing information, and under-specified information. The dialogue manager then selects the question to ask and generates the appropriate speech and sketching. After asking the question (Chapter 6) and receiving the result (Chapter 5), the dialogue manager makes the appropriate updates to the physical situation and the physics simulator is run again.

7.1 User Interface

MIDOS users interact with the system via the interface shown in Figure 7-2. The interface has a row of buttons across the top that allow the user to change between a pen and a highlighter of various colors, motivated by the user studies (Chapter 2 and Chapter 3) that revealed the importance of allowing the users to switch the pen style (pen / highlighter) and the ink color. The bottom of the screen shows the user's

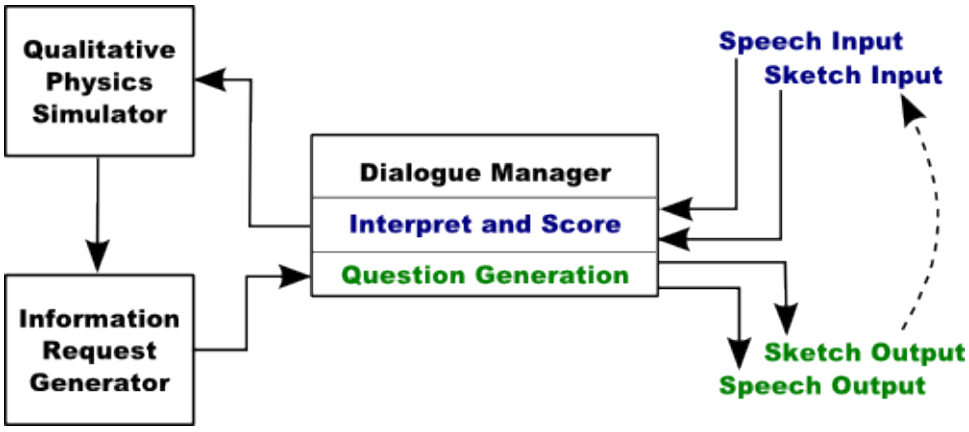


Figure 7-1: An overview of the MIDOS components and how they are connected.

recognized speech and the computer's generated speech. This visual version of the question complements the auditory version and allows the user to review the question MIDOS asked. The user can scroll through the previous questions and replies by using the arrow buttons at the bottom left area of the screen.

The base sketch that contains the various mechanical components is created in a separate part of the program. An example base sketch is shown in Figure 7-3. Users can select different shapes and add them to the diagram by using the mouse. After the base sketch is created, MIDOS will ask questions and simulate the device. Starting with a neat base sketch allowed us to focus on the dialogue aspects of the interface. An area of future work for MIDOS is to allow the user to create the base sketch using freehand drawing. Figure 7-4 illustrates the difference between the neat base sketch and a freehand version.

7.1.1 Technical Details

The user interface allows the user to save the current state to a XML-formatted file. The file contains the shapes in the sketch, the strokes drawn by the user and the computer, and a text form of the user's and the computer's speech. The file saves the history of the entire sketch, and the interaction can be replayed using the saved data. The file format is an extension of the ETCHASketches format [44] that saves the speech data in addition to the sketching data. The user can open or save these

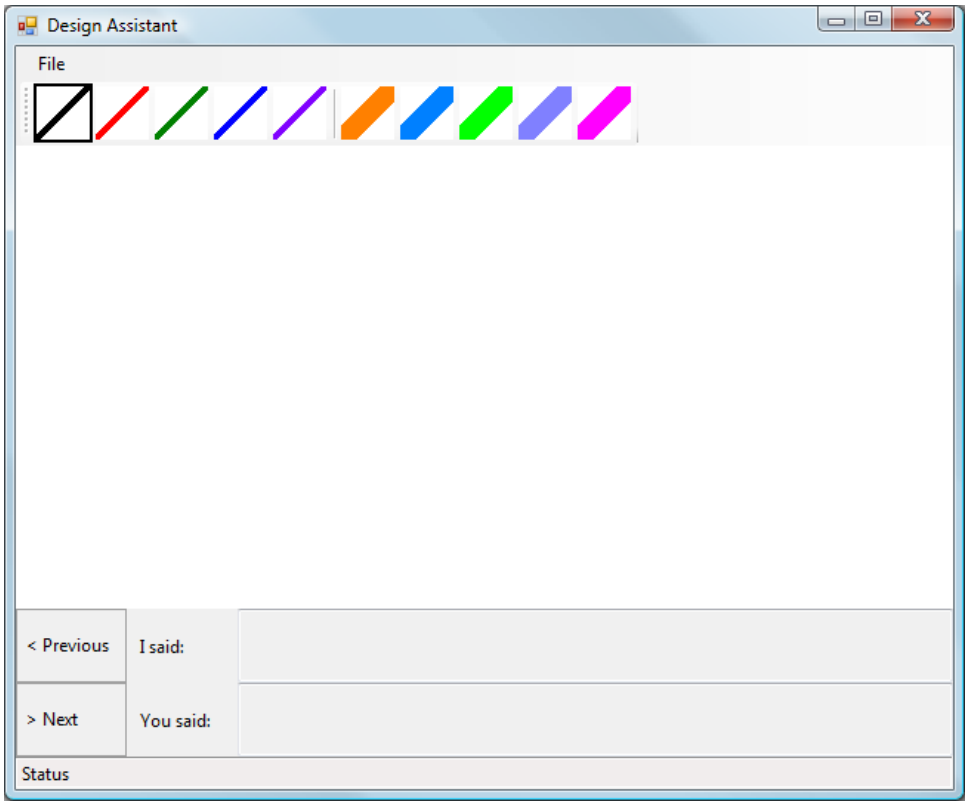


Figure 7-2: The user interface of MIDOS.

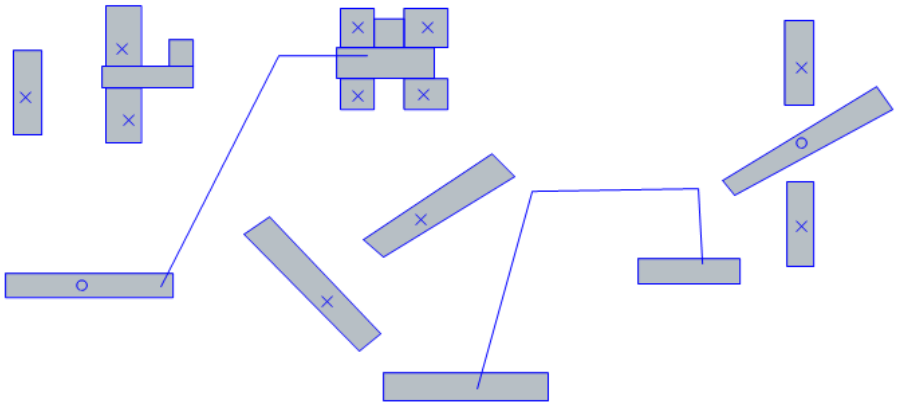
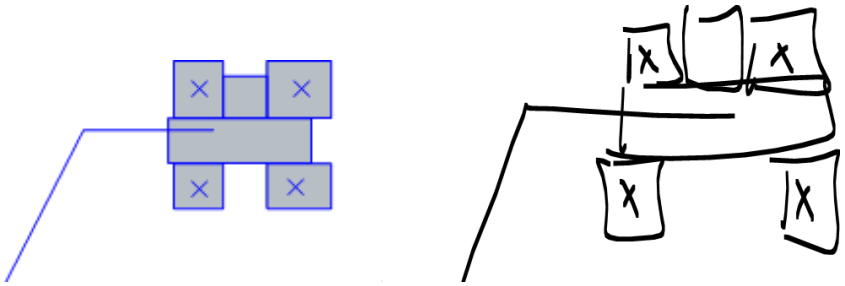


Figure 7-3: A base sketch for a switch flipper.



(a) Neat base sketch (b) Freehand sketch

Figure 7-4: A neat and freehand version of part of a base sketch.

files at any time by using the menu in the user interface.

Although the user interface is written in C#, the rest of the core components of the system are written in Java. The connection between the two is made using several socket connections. The incoming user speech and sketching are captured in C# and then handled by Java. The output consisting of the computer speech and sketching is sent back to be displayed or spoken in C#.

7.2 Qualitative Physics Simulator

The physics simulator is a major component of MIDOS. It acts as a kind of inference engine taking the current state of the world and trying to predict the next state. The simulator serves two purposes: updating the device to the next state and determining the needed information to get to the next state. The physics simulator itself has two important properties: it is qualitative and modest.

As a qualitative simulator, it uses only directions of velocities and accelerations not their magnitudes. This is still useful as the system is designed to allow users to describe early-stage designs, a stage when requiring them to enter precise velocities, masses, and properties like elasticity and friction coefficients would detract from this goal.

The simulator is modest in the sense that we have made a number of simplifying assumptions. We do not claim that it can handle every situation nor that it is an accurate representation of real world physics. We do claim that it can generate a

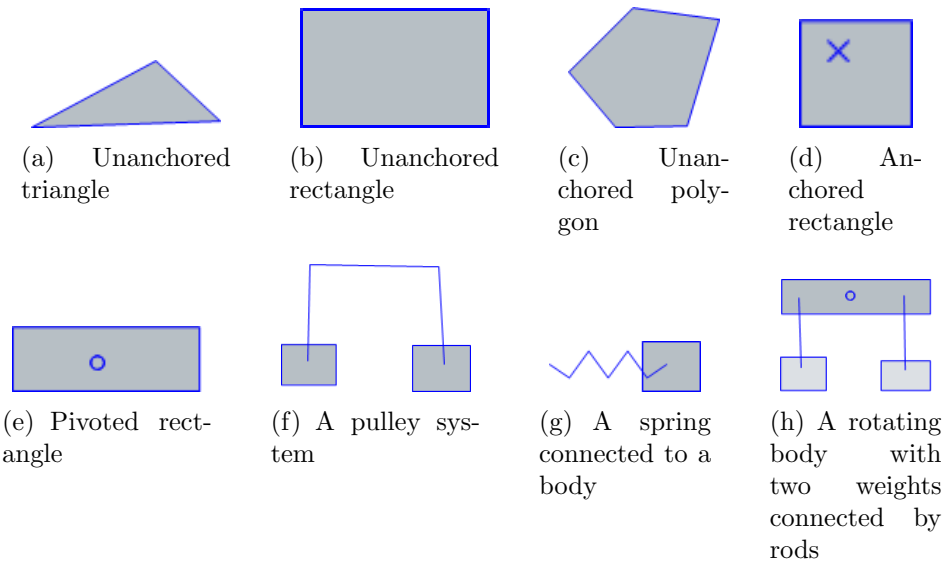
series of sensible questions and update its model of the world appropriately.

Our simplifying assumptions include handling rotations and translations of bodies, but not simultaneous translation and rotation, or friction. Our goal was a simulator sufficient to identify physical ambiguities and to generate sensible questions, rather than one capable of making extensive and subtle inferences. The simulator is sufficient to allow us to focus on the interaction that is created with the user. The simulator can generate a set of reasonably complex questions that engage the user in a discussion about the device. This allows us to focus on the dialogue and how the questions are asked and answered. The evaluation of MIDOS (Chapter 8) showed that the simulator is sufficient to create an engaging dialogue with the user, although some expert users requested that it handle more complex physics.

The simulator runs in real time and attempts to update the device to the next state. If the next state cannot be determined unambiguously, the system creates a set of possible questions to ask the user, in the form of *information requests*. The requests are turned into questions; the answers provide additional data that updates the system's model and allows it to continue simulating the device. As the device is updated, new information requests are generated based on the new state using a variety of techniques described below.

7.2.1 Supported Shapes

The physics simulator handles polygons of various forms, from triangles to quadrilaterals to arbitrary polygons, as well as ellipses (which are approximated as polygons). The polygons can be anchored, free moving, or have a pivot. The system supports pulley systems, springs, and weights connected by rods. A weight is created by attaching a body to a rotating body with a line that only has one segment. The system recognizes this as a rod and weight. The shapes recognized by the system are displayed in Figure 7-5.



(a) Unanchored triangle

(b) Unanchored rectangle

(c) Unanchored polygon

(d) Anchored rectangle

(e) Pivoted rectangle

(f) A pulley system

(g) A spring connected to a body

(h) A rotating body with two weights connected by rods

Figure 7-5: The various supported shapes.

7.2.2 Scope of the Simulator

The physics simulator makes simplifications that make the physics tractable while still allowing MIDOS to generate sensible questions for a conversation with the user.

Some simplifications are made by not handling various aspects of real-world physics. The simulator does not handle simultaneous rotation and translation of objects. It does not account for friction and assumes that ropes never have any slack. Although the user can draw concave polygons and the physics simulator makes an attempt to handle these shapes, the results it returns are not always accurate. The simulator, however, does account for gravity.

One shortcoming of the physics simulator that was evident on a few occasions during the user study is that it assumes all collisions are perfectly elastic, i.e., a collision will transfer all velocity from one shape to the other. This does not allow shapes to move together. For example, if a body collides with another body, the initial body will transfer all of its velocity and stop. Sometimes the desired behavior is to have both bodies move together after the collision.

The physics simulator handles polygons of various forms, as stated previously. Weights, rods, and ropes cannot collide with other shapes as they are essentially 2 dimensional shapes in a 2.5 dimensional world. Weights and rods have the additional

property that the weight and rod always hang vertically from whatever they are attached to. When the simulator computes rotational collisions, it uses a finite set of angles, allowing it to try all possible rotations to find collisions.

The system's model of springs is fairly simplistic. A spring has a maximum length to which it can stretch, a minimum length to which it can compress, and a current motion direction (or it can be stopped).

These simplifying assumptions are not always beneficial. Conflicting calculations between approximations and exact calculations can sometimes cause the physics simulator to behave erroneously. For example, a fixed set of angles are used to approximate rotational collisions, but an exact distance is used to determine if two shapes are touching. The rotation approximation can indicate that a shape has rotated as far as it can without overlapping with another shape, although the distance between the two shapes is still too large for the shapes to be considered as touching. The differing calculations can prevent the simulator from realizing that the shapes have collided. Chapter 10 discusses ideas for how the simulator can be improved.

7.2.3 Calculation Techniques

The physics simulator approaches the physics calculations from several directions. We discuss each of the techniques below.

Base Constraints

The simulator starts by applying some basic constraints to all of the shapes, including setting the velocity of an anchored body to zero, ensuring that pivoted shapes have only rotational motion, and adding the force of gravity to unanchored shapes. Gravity is applied by giving effected shapes a downward acceleration.

Additionally, the simulator creates statements that reflect the current properties of the device. This includes statements about shapes that are anchored or are attached to a spring, rope, or weight. These statements are used to find shapes with specific properties and generate the appropriate information requests. If a spring or rope is

at the limit of its motion – it is stretched or compressed as far as it can go or at the end of its rope – a statement is generated to reflect this. This statement is used in the degrees of freedom calculation.

Degrees of Freedom

Another technique is to examine the degrees of freedom that each shape has. The degrees of freedom are a way to represent in which directions a shape can translate or rotate. We calculate two sets of degrees of freedom for each shape, one for translational motion and one for rotational motion. One of the simplifications in the physics simulator is that shapes can have only translational or rotational motion, but not both. All of the shapes, however, affect the calculations of the degrees of freedom for both types of motion. We discuss how the degrees of freedom calculations are used and then describe the calculations themselves.

The degrees of freedom are used to calculate how far shapes can move along a surface, when a shape might change direction, what direction or rotations are permissible for a shape, and how the velocity of a shape is transferred when it is involved in a collision, among other computations.

The system calculates the degrees of freedom as it updates the position of shapes. When the degrees of freedom change, a direction change is possible and an information request is generated to determine the new direction of the shape. This can occur when a shape reaches the end of a body or when a shape moves away from a surface with which it was in contact. For example, if a shape is resting on a surface, its degrees of freedom are restricted in the downward direction, and it cannot move down. If the shape is then moved to a position above this surface, its degrees of freedom are now unrestricted, and it can move in any direction. If the shape is under the influence of gravity, its motion is now uncertain because it could either continue to move up or it could move down. Similarly, if a body slides off the end of a surface, its degrees of freedom will change because it is now free to move in a downward direction.

For rotations, the degrees of freedom are used to determine how far a shape can rotate. If there is an anchored body in the path of the rotating shape, the rotating

shape cannot rotate past that body.

When a collision occurs, the velocity of the colliding body is transferred to the body with which it collides. Here, the degrees of freedom are used to calculate the new velocities of the bodies. The collision direction and the degrees of freedom of the body that is hit are taken into account. It is possible that the hit body cannot move in the direction that the collision pushes. In this case the colliding body might bounce or stop. An information request to determine what will happen would be generated by the system.

If the movement of a body will change the degrees of freedom – for example, separating from another body or moving past the edge of another body – the body will be updated a distance of 50 himetric units (as long as this is physically possible). This provides some visual separation so that the user knows that the shape has been moved.

The translational degrees of freedom are calculated as follows:

1. Assign degrees of freedom to easily calculated shapes. This includes anchored shapes that cannot move and shapes that are not touching anything that can move in any direction. Pivoted shapes are set to have no restrictions on their translational degrees of freedom, a temporary setting for the calculations.

2. Shapes that are attached to a spring or pulley that cannot move any further in a particular direction are set to have the appropriate degrees of freedom.

3. Groups of touching shapes are found.

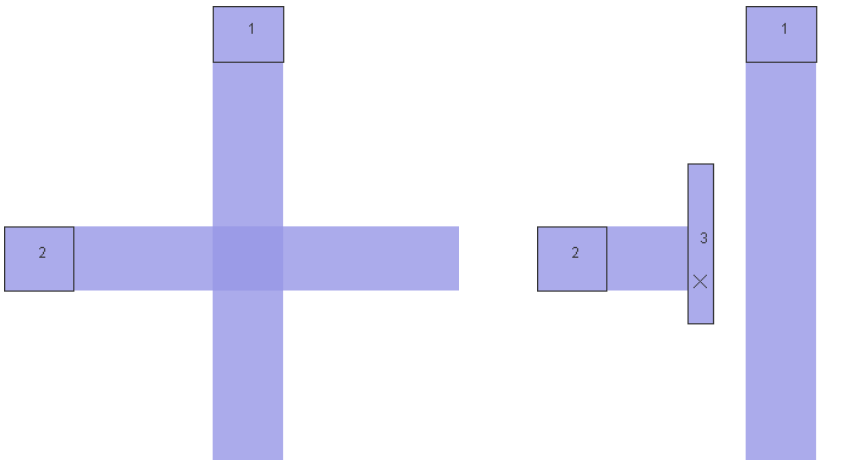
4. The degrees of freedom are calculated for each group. If a group of shapes has no shapes with restricted movement, all the shapes in the group can move in any direction. Otherwise, the degrees of freedom are propagated between the shapes in the group based on the current degrees of freedom and the contact surfaces of the shapes.

5. After all the degrees of freedom are set, the pivoted shapes are set to have no translational degrees of freedom.

The rotational degrees of freedom are calculated in a more direct way. The shape is rotated to a fixed number of angular positions and checked for any overlap with anchored shapes or shapes with which it collides as determined by user responses to information requests. The degrees of freedom are set so that the shape can rotate to any position without overlap.

Projections

Collisions also play a significant role in the physics simulator. The system uses projections to calculate possible collisions. Projections are polygons that represent the area that a shape moves through as it follows its trajectory or rotation path. Projections for translating shapes have a finite (but large) length limit of 200,000 himetric units, which allows the simulator to calculate possible intersections. Once the projections are determined, MIDOS finds all the intersections of the projection areas. These intersections represent possible collisions. Figure 7-6(a) shows two shapes and their intersecting projections indicated by the shaded regions. The intersection of the projections leads to a predicted possible collision between shapes 1 and 2.



(a) A potential collision is predicted for shapes 1 and 2.

(b) A potential collision between shapes 1 and 2 is no longer predicted due to anchored shape 3.

Figure 7-6: Two examples of two translating shapes and their projections indicated by the shaded regions.

The projections are refined to reduce the number of predicted collisions that can-

not occur. Projections can be limited by anchored shapes, pulleys, and springs that constrain the path of a moving shape. Figure 7-6(b) shows how the projection of the shape numbered with 2 is stopped by anchored block numbered 3. A collision with shape 1 is no longer predicted by MIDOS.

The qualitative simulator can determine if objects might collide because of intersecting projections, but it cannot be certain because the speed of the objects is unknown. Each potential collision has a matching information request that is generated to determine if the collision actually occurs.

The system assumes that if a shape can move all the way to an anchored shape that it will do so, and the system will not ask if the two shapes will collide. This assumption works for the current system, but refinements may be necessary in the future because sometimes the shape will change direction before it reaches the anchored block. Currently, for example, gravity only affects shapes when the degrees of freedom of the shape are changed. If a shape moves across a long distance to an anchored body, it is possible that the path or direction might change over that distance. The current system does not take this into account. The solid red line in Figure 7-7 is the path MIDOS currently predicts and results in a collision. The dashed green line shows a more accurate path that takes gravity into account and does not result in a collision.

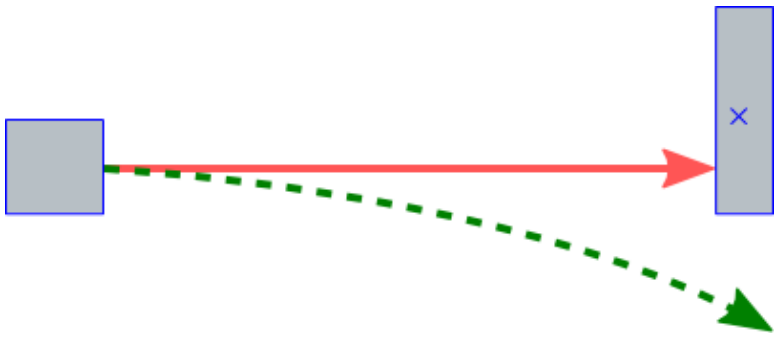


Figure 7-7: The solid red line indicates the path MIDOS currently predicts and results in a collision. The dashed green path indicates a more accurate path that would take gravity into account and does not result in a collision.

Statements

Statements represent a piece of knowledge about a property or behavior of the device, and are used to keep track of bits of information as the simulation proceeds. Some statements represent basic information about the device, such as a statement about a shape that is connected to a spring. Other statements store the result of an information request, such as a statement that two bodies collide with each other. The statements can be generated and used by different components of the physics system.

Statements can be generated and consumed at different points of the simulation. Statements about bodies connected to a spring will persist throughout the simulation. Other statements, such as those about balanced shapes or collisions, will persist only as long as they are relevant. If a body is balanced by several other bodies, a collision involving any of those bodies will cause the statement about the balanced body to be consumed. Similarly, a statement that two bodies will collide will be consumed after the collision between those shapes occurs.

Determining when the information represented by the statements is no longer relevant poses an interesting challenge, this is also known as the frame problem [40]. In particular, there is tension between keeping collision information for too long and not keeping it long enough. If the information is kept too long, out-of-date collision information might be used. If it is not kept long enough, the user will be asked the same collision questions repeatedly. We strike a balance by more aggressively forgetting the collisions that do not occur while keeping the information about the collisions that do occur.

Modifiers, Uncertainty, and Conflict Resolution

The physics simulator performs updates in two phases. The first phase calculates the new velocity of all the shapes based on the collisions, springs, weights, and pulleys. The second phase analyzes the results of the first phase and identifies any conflicting updates. If there are no conflicts, the velocities of all the shapes are updated. If there are conflicts, an information request is created to obtain the desired velocity from the

user.

The physics simulator handles uncertainty when a precise value for a motion direction (translational or rotational) cannot be determined. A shape can be updated to have a range of possible directions or an unknown rotation. Similar to the conflicting updates, an information request will be generated to ask the user for the information. This allows the system to ask a question about whether a shape will move in a clockwise or counterclockwise direction, or ask a question about the angle of motion of a shape.

User Provided Input

MIDOS is not driven exclusively by the physics simulator. In addition to the simulator, the user is prompted for critical information in a series of questions derived from the information requests. Asking the user questions allows the system to overcome some of its shortcomings and limitations by updating the physical situation based on the user provided data. For example, the physics simulator is qualitative and does not know the magnitude of any velocities, making it impossible to determine quantitatively if a collision between two shapes will occur. The system asks the user if the collision will occur and overcomes this shortcoming.

7.2.4 Generating Information Requests

The techniques described in the previous section generate information that can update the physics simulation to the next state, and generate information requests that are used to form questions to ask the user.

Each type of information request analyzes the current information in the physics simulator and the current set of statements, and creates the appropriate set of requests. Each request encapsulates the information needed to ask the question and interpret the answer from the user. The information in the statements allows the information requests to determine which information has already been obtained so that questions are not repeated. The information requests are described in more detail in

Section 7.3.

Auto Answering

MIDOS attempts to answer a few of the information requests without asking the user. In particular, when a body collides with a set of previously balanced bodies, MIDOS will attempt to determine which way the (now unbalanced) pivoted body will rotate. Thus, the system can avoid asking the user this particular question.

7.2.5 Shortcomings

The system assumes that a shape will move as far as it can until it hits something or the degrees of freedom change. This is true in the system's physics model because it does not handle friction. The user, however, may wish to move the block a smaller amount than the system thinks it will move. In the current system, it is not possible to convey this information.

The system does not give the user an easy way to go back and make corrections. For example, if the user specifies the length of a spring and does not quite have it stretch far enough for an intended collision to occur, there is no way to go back and tell the system that the spring stretches further.

Another difficulty with the current system is that it forces the user's answer into one of the answer choices that the system computed. Sometimes the user may want to correct the system and provide an updated position or velocity for a particular body. In the current system this cannot be done.

7.3 Information Request Processing

Each type of *information request* analyzes the current state of the system and the physics by using the positions of the shapes and the collection of *statements* about the system that have already been determined.

Information requests encapsulate pieces of information that MIDOS needs to acquire. For each request, MIDOS:

- Analyzes the current physics state, statements, and pending updates; then determines if it needs any information from the user,
- Generates the question to ask the user.
- Anticipates what speech and sketching to expect in the user's reply.
- Interprets the user's reply and updates the physics state appropriately.

Matching the user's input was described in Chapter 5, but each information request knows how to handle the speech and sketching it receives.

Table 7.1 and Table 7.2 list some details about the information requests used in MIDOS. The order the requests are listed in the table reflects the priority of different information requests. The following sections describe the process for selecting the next request, generating the question, and interpreting the user's reply.

7.3.1 Determining the Next Request

Several factors are considered in the attempt to ask questions in a reasonable order and avoid redundancy. First, information requests are not created if there is already an answer in one of the statements. Second, information requests can store information for other information requests of the same type. Currently, this is used for collision requests to store the distance to the collision. MIDOS uses the distances to ask about the collisions that happen at short distances first. After the user verifies that a collision happens, MIDOS will not ask about collisions involving those shapes that happen at longer distances.

The list of possible information requests is generated in an order that reflects the interdependence of the questions. For example, the system will ask about a body's trajectory before inquiring about a collision involving that body. The answer to the trajectory question could render the collision question moot.

Although there is a specific order to the information requests (as shown in Table 7.1 and Table 7.2), the order is modified in several ways:

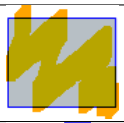
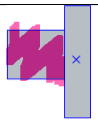
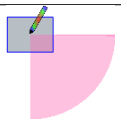
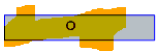
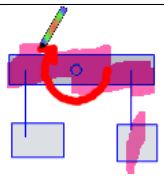
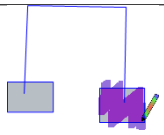
Type	Image	Description & Sample Text
anchor		ask if a not anchored, not moving block is anchored
		"Is (this shape) anchored?"
bounce		does a shape bounce after colliding with another shape
		"Does (this shape) bounce after the collision?"
angle		what angle is the shape moving in (or is it stationary)
		"Which of {these directions} does this shape move in?"
rotation direction		which direction (or no direction) does the shape rotate in
		"What direction does (this shape) rotate in?"
rotational velocity		pivoting shape might rotate or might be balanced
		"I can not determine the rotation of (this shape) now. This shape causes a (clockwise rotation.) <short pause> <clear stroke> This shape causes a (counterclockwise rotation.) <short pause> <clear stroke> <short pause> <clear strokes> At this instant what direction does (this rotate in) or is it balanced?"
pulley		what direction does the pulley move or is it balanced
		"What direction does (this shape) move in at this instant?"
distance		how far does a shape move along its trajectory (or is it stationary)
		"How far does (this shape) (move?)"

Table 7.1: Part 1: The information requests, a sample question, and an image from the question being asked.

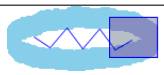
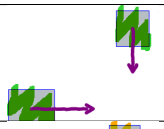
Type	Image	Description & Sample Text
rotation distance		how far does the shape rotate around the pivot
		"How far does (this shape) (rotate?)"
spring direction		which way is the spring force
		"Will (this spring) expand or contract?"
spring length		how far can the spring expand or contract
		"How far does (this spring) stretch?"
spring end		what happens when the spring expands or contracts all the way, does it change direction
		"(This spring has) reached its maximum length. What happens next?"
collision		do two shapes collide
		"It looks like {these shapes} {will collide, do they?}"
collision location		where do two shapes collide
		"(These two) (bodies) collide (here.) <long pause> <clear strokes> Where on (this) body does the contact occur?"
next		what is the next thing that happens or is that the end of the simulation
		"What happens next?"

Table 7.2: Part 2: The information requests, a sample question, and an image from the question being asked.

1. Requests that have already been asked and requests that are no longer valid are removed from the list.

2. If the user did not successfully answer the question from the last information request, try that request again.

3. If there is a request of the same type that involves one of the same shapes as in the previous question, use that as the next request. If possible, we want to keep asking about the same shapes.

4. Use the next request in the list.

7.3.2 Generating the Question

Asking the user questions about the design engages the user, the user supplies more details, and MIDOS receives the information necessary to continue the simulation. Each information request knows how to form an appropriate question to acquire the information it needs from the user. These questions vary depending on the current state of the physics. For example, a question about a body's rotational direction uses information about the different forces and the shapes that exert these forces in its question. In addition, if a question is asked multiple times, the question is changed to reflect this repetition by adding words such as "now" or "again." For example, MIDOS will ask "Does this shape bounce again after the collision?" if the shape has already bounced after a previous collision. If the user does not provide an acceptable answer the first time the question is asked, she is given additional guidance when the question is asked again. For example, if MIDOS asks about a collision a second time, it will ask: "Do the bodies collide? Yes or no?" Providing the user additional guidance to encourage an answer that the system will understand has been incorporated into other systems [27, 28, 31, 33]. MIDOS has the advantage of knowing the question that was asked and the expected answers. The details of generating the speech and sketching output were discussed in Chapter 6.

7.3.3 Processing the Reply

The initial processing of the user's response based on the speech and sketching that the information request is expecting is discussed in Chapter 5. When a match is found, the information request generates a set of statements that reflect the information contained in the user's response. If the user was asked about the direction of a body and supplied a particular angle for the direction, the velocity of the body will be updated. If the user specified that a shape was balanced, the information request will update the shapes that are balanced by setting the velocities appropriately. It will also add a statement that specifies the shapes that are balanced so that they will not be updated until the balance is disturbed.

Users provide answers of varying length. As in the original set of user studies, some of the answers are long. If the system cannot process the user's input, the request for information is made again with additional guidance about the type of answer the system is expecting. Statements are added to the physics simulation only when a match with the user's input has been successfully made, as described in Chapter 5.

Chapter 8

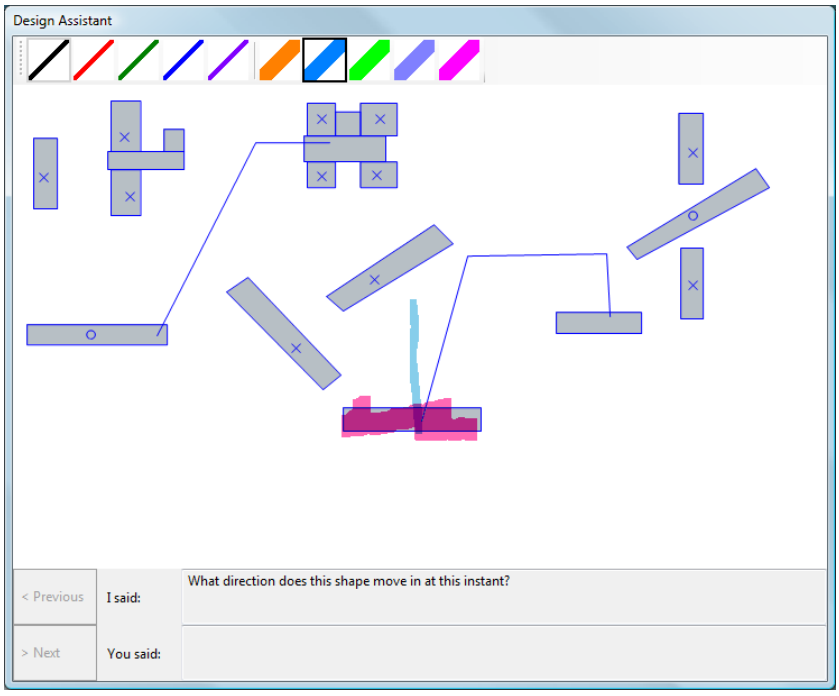
MIDOS Evaluation

This chapter discusses the evaluation study of MIDOS. MIDOS has sufficient capabilities to be evaluated by real users to determine how closely the dialogue in MIDOS matches the observations in the original studies that guided its development (Chapters 2 and 3). The evaluation study showed that users preferred interacting with MIDOS more than a text version of the system. Although some users perceived that the text version was faster, this was not supported by the data.

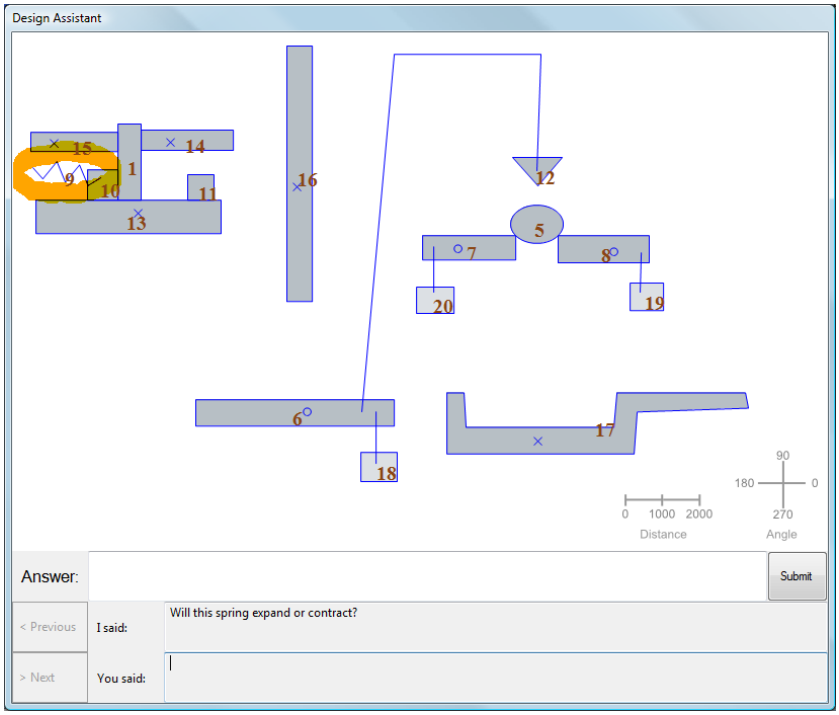
8.1 Setup

The primary purpose of the study was to test the dialogue capabilities of MIDOS and to observe how well the user interacted in a dialogue with the system. Participants in the study interacted with two versions of the system. In both versions the computer asked the user multimodal questions using speech and sketching. In one version the user could respond to the questions using sketching and speech (MIDOS version). In the other version, the bodies were labeled with numbers and user had to type her answer (text version). The two user interfaces are shown in Figure 8-1. The text version provides a basis of comparison for the multimodal version of the system.

There were 12 participants in the study who responded to advertisements on the M.I.T. campus or responded to emails. The participants ranged in age from 20 to 39 with an average age of 26.3. Seven participants were male and five were female. One



(a) MIDOS version user interface.



(b) Text version user interface.

Figure 8-1: The user interfaces for the study participants for the MIDOS and text conditions.

participant owned a Tablet PC, and four participants had used one many times. Five people had used a tablet only once before and two people had never used one. The participants were given two movie tickets as compensation.

Each participant used a Tablet PC which ran a slightly modified version of the C# MIDOS code. The Java part of MIDOS ran on the experimenter's computer. The speech recognition and generation were performed on the participant's tablet. Participants wore a headset microphone/headphone that allowed participants to hear the computer generated speech and provide verbal input.

The participant's microphone was connected to the laptop and to an audio mixer. Similarly, the audio output of the participant's tablet was connected to the participant's headset and to the audio mixer. The mixed audio feed was distributed to the experimenter and to a video camera allowing the experimenter to listen to the questions and answers while it was recorded. The study was videotaped primarily to record the audio for further analysis, but also to have a visual record. The video was recorded to tape and then subsequently digitized and stored electronically.

Figure 8-2 shows an overhead view of the study layout. The participant and the experimenter were in separate rooms. This provided a noise-free environment, prevented the experimenter from affecting the participant's responses, and allowed the experimenter to use the wizard interface undetected.

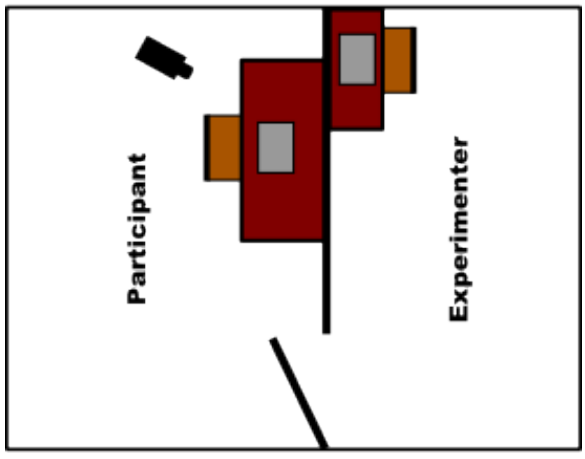


Figure 8-2: Overhead view of the MIDOS evaluation user study layout.

8.1.1 Wizard Details

Several adjustments to the MIDOS system were made for the user study. The largest change was the addition of a wizard [17] to the system. The job of the wizard was to interpret the participant's speech or text input. We chose to run a Wizard-of-Oz experiment, to avoid problems with speech or text recognition accuracy and to keep the focus on the dialogue component of the interaction. Understanding the unrestricted text the participants entered required the use of a wizard, and to keep the two experimental conditions as similar as possible, we also used a wizard in place of automatic speech recognition. The interface for the wizard is shown in Figure 8-3.

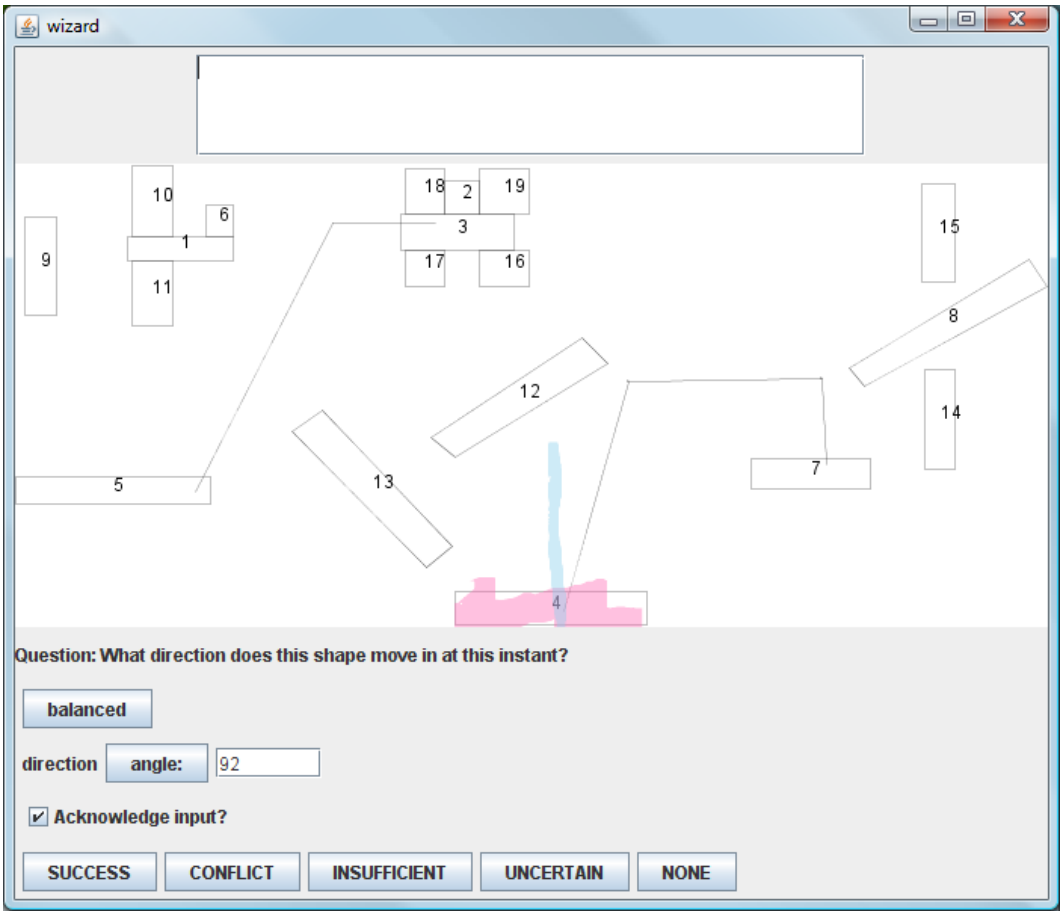


Figure 8-3: The controls the wizard used in the evaluation study.

MIDOS selects the questions and the wizard interface allows the experimenter to enter the user's answer using a set of buttons and fields. The fields allow the experimenter to adjust or enter numbers that indicate properties such as the direction

of the user's strokes. The wizard can also select whether the question was answered successfully or indicate that the answer is insufficient or in conflict. The wizard interface does not change other aspects of the system. The system still waits for the user to finish her input and pause before it will acknowledge the input and update the physical situation. Users can still interrupt the computer as it is asking the question.

In the text version, the wizard sees what the participant types in real time, and can enter the interpretation of the user's input at any time. The system will not act on the participant's text input until they press the submit button. Similarly, in the MIDOS version, the wizard hears the participant's speech and can see any strokes the participant draws when they lift the pen. In the MIDOS version, the system attempts to calculate the appropriate parameters from the participant's strokes to help improve the responsiveness of the wizard. The wizard can again enter an answer at any time, but the system will not act on it until the participant pauses her sketching and speech.

8.1.2 Study Procedure

The participant first filled out a pre-study questionnaire that included demographic information questions. She was then given a set of written directions that explained how to use the interface. The experimenter answered any questions the participant had. After the participant understood how to use the system, the warmup condition was run. Following the warmup device, the experimenter again answered any questions the participant had.

The devices used in the study are shown in Section 8.2. For each device the participant first watched a movie of a Working Model simulation of the functioning device. The video could be viewed as many times as the participant wanted. Then the participant answered the multimodal questions about the device by using either the MIDOS or text version of the system. The warmup condition was always the MIDOS version to help familiarize users with the novel interface. The remaining four devices alternated between the text version and the MIDOS version. The order of the versions and devices was randomized to avoid any ordering effects. For the MIDOS conditions, the Tablet PC was placed in the flat, slate mode. For the text conditions,

it was placed in laptop mode.

At the start of each condition, the participant's screen flashed and the laptop emitted several beeps. These indicators are used to synchronize the sketch file with the video tape so that the data can be replayed together.

Immediately after answering each question, the participant was asked to rate the question on a 1-5 scale. This dialog box was modal, so that the participant was forced to answer this question before the system would continue. The dialog box is shown in Figure 8-4. The wizard does not see these scores and therefore cannot inadvertently affect the results.

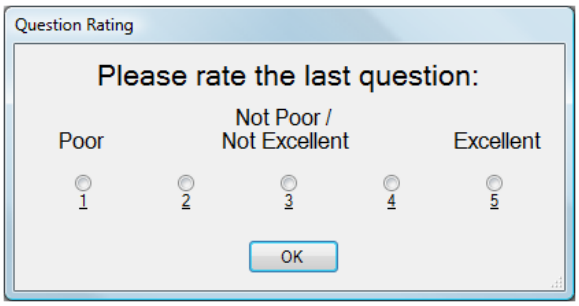
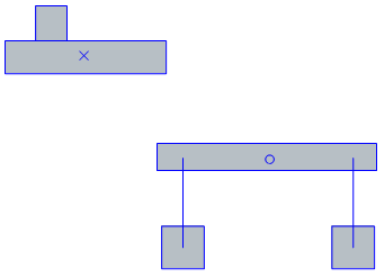


Figure 8-4: The rating dialog box used in the evaluation study.

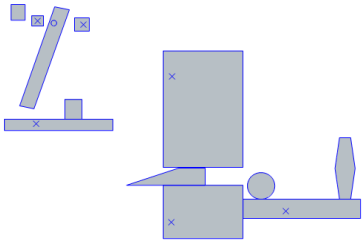
Data files that captured all of the input and timing data for the questions and responses were saved after each device. After all of the conditions were completed, the user completed a second questionnaire that gathered data about her impressions of the two versions of the system.

8.2 Devices

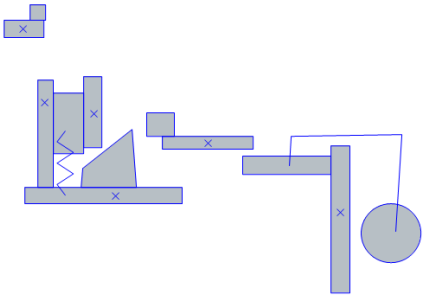
The five devices used in the study are shown in Figure 8-5. For the text conditions, the bodies were labeled with numbers that the participants could reference in their descriptions. The devices, other than the warmup device, were chosen to have a clear purpose, for example knocking over a bowling pin.



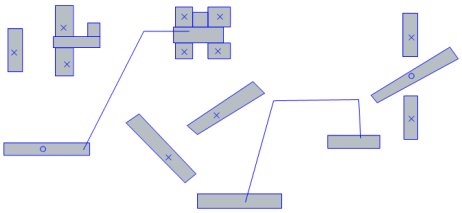
(a) Warmup



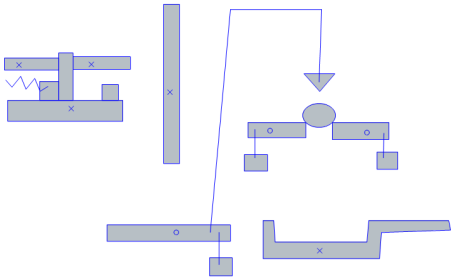
(b) Bowling ball roller



(c) Flag raiser



(d) Switch flipper



(e) Egg cracker

Figure 8-5: The five devices that the participants in the study described.

8.3 Qualitative Results

The qualitative results from the study fall into several categories, including sketching observations, speech observations, and general observations.

8.3.1 Sketching Observations

One participant starting doodling after answering the questions. This caused the system to seem unresponsive because it was waiting for the participant to finish drawing. After several instances of this behavior, the experimenter informed the participant that the doodling was delaying the system response.

Some participants started writing words such as "Yes" and "No" when verbally answering the question. When possible the wizard would indicate that the answer could not be understood in an effort to discourage this behavior, however, this had a limited effect.

8.3.2 Speech Observations

The vocabulary participants used was varied. Some participants used words such as "affirmative" as a synonym for "yes." Other answers included phrases such as "dude" and "my friend" in reference to the computer.

Participants in both the MIDOS version and the text version described distances and angles using relative terms. They used words and phrases such as: up, down, down and to the right, and half the length of shape five. Even in the text version that provided a direction and length key (the scales at the bottom right in Figure 8-1(b)), participants almost exclusively used relative language.

The tone and speed of the participant's voice in the MIDOS version varied as the participant was more or less confident in her answer. A more advanced speech recognizer might be able to take advantage of this and provide MIDOS with an indication that the confidence in the answer is low or that the question needs to be clarified.

8.3.3 General Observations

Several minor technical difficulties occurred during the user study. The data for several sketches were lost, however, the data related to question and response timing were preserved. Occasionally a limitation of the physics simulator would cause odd behavior to occur or cause the system to get stuck. These instances were infrequent and in most cases the experimenter could help get the system back on track. In the few cases that the system got stuck, only a few questions remained. In these cases, the condition was ended before the device was completely simulated.

The egg cracker required the longest explanation, followed by the switch flipper. Participants spent less time explaining the bowling ball roller and the flag raiser. The shorter explanations contained more questions that could be answered with a simple yes or no and less opportunities to use sketching.

Many of the participants were confused when they started the warmup condition. After some uncertainty they became more comfortable with the speech and sketching interface. A demonstration video may be helpful if a similar system is deployed. Although a video was considered for the study, it would have biassed the interaction and the responses.

Participants responses tended to provide more information than the system could handle (see Figure 8-6). These long explanations are similar to the explanations in the dialogue user study (Chapter 3). These extensive descriptions occurred more frequently at the start of each interaction. After the system reacted to only a small portion of the provided input, the participant provided shorter responses. The long explanations were also present in the typed responses (see Figure 8-7). The length of participant's MIDOS answers was limited because a small pause enabled the system to process the participant's input and then move on to the next question. In the text version, this was not the case. The computer could only process the response after the participant clicked the submit button.

Some of the descriptions provided by participants could not be handled by the system. These limitations include:

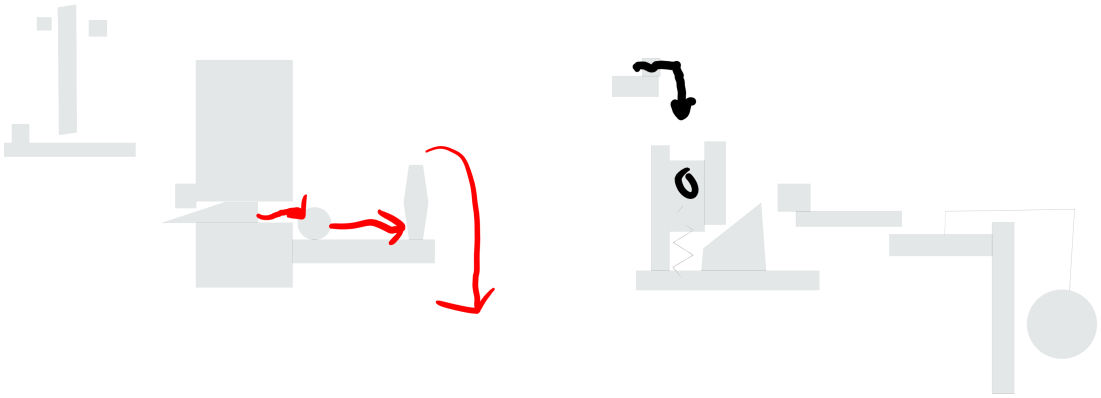


Figure 8-6: Two examples of the user providing more data at once than the system can handle.

> 3 falls of off 8 and crashes into 4. 1 (the spring) contracts and causes 4 to crash into 5. 5 is pushed into 2 from the angle of 4 falling into it. 2 is then slid across 12 from the reaction of 5 slamming into it. 2 then falls off of 12 and lands on 6. while 6 moves down, 2 slides back and forth on 6 until 7 slows down 6. this is the end.

Figure 8-7: An example of a user providing more text at once than the system can handle.

- The system can handle only the action it asked about; it cannot handle more than one piece of information at a time.
- The system does not handle paths, only specific directions (a particular angle).
- The system does not model small movements, such as oscillations.
- The system does not handle moving a short distance along a surface (friction).

8.3.4 Questionnaire Ratings

The MIDOS version was preferred by 10 of the 12 participants. Participants were more divided on the other preference questions. Seven people thought the MIDOS version was easier to use and five thought the text version was easier. Surprisingly, seven people thought that the text version was faster. We analyze this result in more depth in the next section.

Eight participants thought the multimodal system was more accurate, and eight

people trust the MIDOS version, while 6 trust the text version. Six people would use the MIDOS for their own tasks and five would use the text system.

8.3.5 Questionnaire Comments

At the conclusion of the study, participants provided comments on the systems:

- "strengths was typing, weak was sketching"
- "I gave negative ratings when questions were worded in a confusing way (mostly talk about "collisions" when two bodies were in contact) or seemed irrelevant... Typing was faster, especially since I could answer a long question before the voice finished reading. Drawing was more accurate for describing directions of movement."
- "The sketching system is more intuitive and allows more freedom. The typing system does not allow the same degree of expression."
- "Drawing... too slow sometimes, a picture is worth a million words. Vice versa [for typing]"
- "I gave negative ratings to question that had little to no bearing on the purpose of the device and conversely positive ratings to those critical. For each of the systems, it seemed less able to cope with describing small details."
- "I gave negative ratings for question I found irrelevant"
- "I gave negative ratings to repetitive question. Typing was faster than speech & sketching. The sketches were more precise and intuitive."
- "I gave questions that were unclear or repetitive negative ratings, and positive ratings for questions that were especially clear or good "Mastering Physics" type question. ... does "falls down" work or does the system really need angles and lengths?"

- "The shorter the question the better! The weakness of typing was directions... The strength was the sketch stylus allowed direct and simple communication with the computer!"

- "I liked the speech and sketching interface. Obviously some tasks are easier for the speech and sketching interfaces while some tasks are easier for the keyboard interface. Text required a lot of verbosity for some questions."

- "The system as a whole would be more fluid if I could provide more information at once. I did like the questions where it knew qualitatively what should happen, just not the right angles / distances to use. They make it feel like the system "gets it." Typing is precise, and it's very hard to tell if it got the speech right."

- "There were two types of bad questions - ambiguous and "stupid" - either the system was asking a question where the answer was a simple extension of the question, or repeatedly asking the same question. Excellent questions made use of sketching and had answers which required some thought. ... Sketching / Speech - it was much easier to specify directions, associations, etc. Sketching / Speech felt much more fluid and intuitive. With text I felt like I had to spend more time thinking about what I wanted to say."

Participants also provided their thoughts about improving the system:

- "Let the system decide some of the actions rather than asking the user."

- Suggested combining the sketching and typing (instead of speech)

- "At times I felt impatient with the speed of the computer's speech."

- "I would have appreciated more feedback from the system so I'd know that it understood what I was saying."

- "Faster response time for speech & sketching"

- "It needs to show me what it's thinking. Once or twice it got on the wrong track, but it took a while to realize and I can't really correct it. But, don't ask

me if it got it wrong every time, just let me answer the new question or tell it to back up."

Participants in the study wanted the physics to handle more complicated paths and motions, were mixed in their opinions about whether text or speech and sketching was better, and pointed out that sketching was particularly valuable in some cases. The next section discusses the quantitative results from the study.

8.4 Quantitative Results

This section discusses five sets of quantitative results from the study, speech and sketching timing, speech and text statistics, color usage, perceived interface speed, and question ratings.

8.4.1 Speech and Sketching Timing

In the MIDOS conditions, participants could use a combination of speech and sketching. We conducted an analysis similar to the one in Section 3.3. The current study was set up in such a way that we could easily obtain information about the phrase-level grouping of the speech and sketching. The participants responded to 510 questions. Many of those responses were unimodal, 41 used only sketching and 334 used only speech[1]. There were 135 multimodal responses; 71% of these responses started with speech. Table 8.1 shows more detailed results which are comparable to the results from our dialogue study. The data is also shown in the graph in Figure 8-8. Our results on a phrase level again differ from the results reported in [49], which reported that sketching usually preceded the speech. Instead they show that a speech utterance precedes sketching in most of the multimodal input to MIDOS.

[1]The high number of speech only responses is likely due to the large number of questions that could be answered with "Yes" or "No."

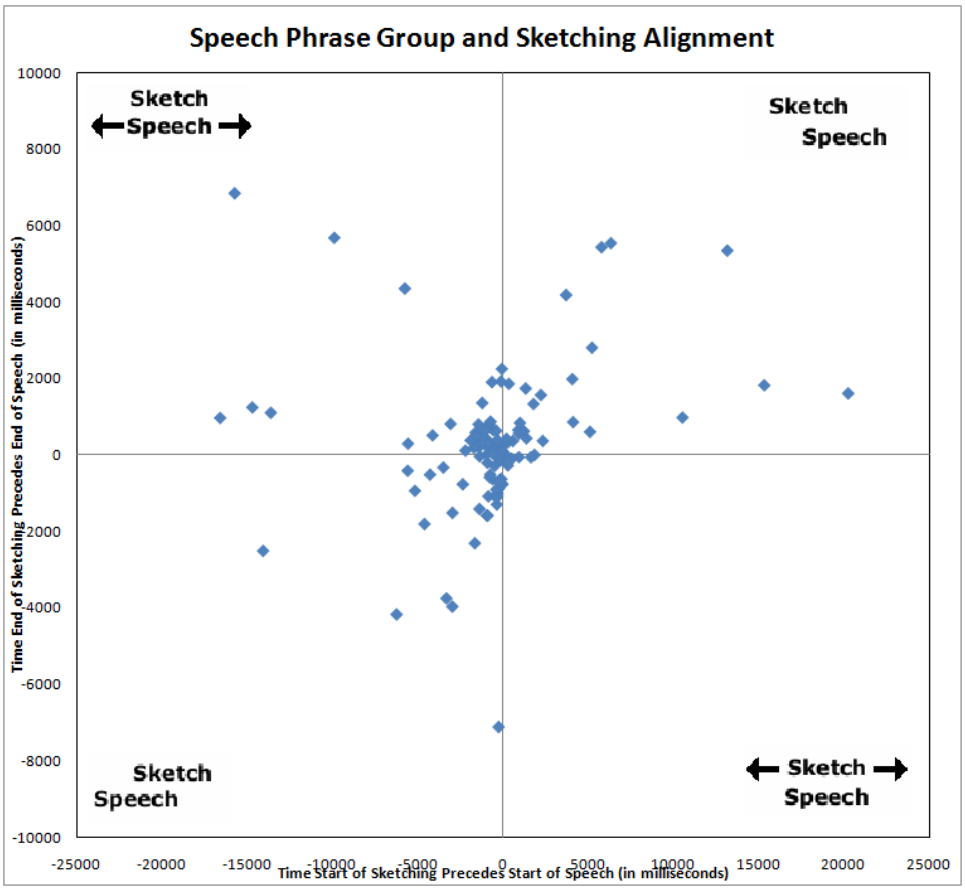


Figure 8-8: A graph depicting the time differences between the start and end times of the speech and sketching in each phrase group. The x-axis is the time in milliseconds that the start of the sketching preceded the start of the speech. The y-axis represents the number of milliseconds that the end of the sketching preceded the end of the speech. The words in the corners of the graph give a visual depiction of the overlap of the speech and sketching in that quadrant.

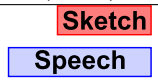
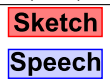
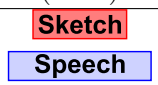
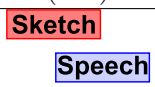
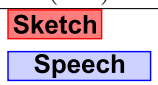
Speech Precedes (71%)	Sketch Precedes (28%)	Neither Precedes (2%)
 (2%)	 (2%)	 (0%)
 (32%)	 (5%)	 (1%)
 (36%)	 (21%)	 (1%)

Table 8.1: The temporal overlap patterns for the phrase groups for the MIDOS study. The alignment of the speech and sketching is illustrated in each table cell. The percentage of phrase groups in each category is also noted.

8.4.2 Speech and Text Word Counts

Participants in the dialogue user study were voluble. The participant's replies in the evaluation study were analyzed to see how long the text input and speech utterances were.

There were a large number of questions that could be answered with a simple "yes" or "no" response and this is reflected in the data. There were 531 text input replies, of which 314, or 59.1%, were one word. Of the speech utterances, 228 of the 497, or 45.8%, were one word. Some user responses were quite lengthy: the maximum number of words in a text input was 75 and the maximum in a speech utterance was 67 words. The average number of words in a speech utterance was 3.7, and the average number of words in a text input was 3.3. Figure 8-9 shows a histogram of the distribution of the length of the speech utterances and text input. Note that there are fewer two word utterances than three word utterances.

The number of questions and responses for each condition is also informative. The average number of speech utterances used to describe a device was 20.7, and the average number of text inputs was 23.8. The egg cracker took the largest number of interactions on average to describe (30.5 speech utterances and 35.2 text inputs). The egg cracker also had the longest individual interaction length, 43 speech utterances

and 55 text inputs. The bowling ball roller took the fewest number of interactions to describe (13.0 speech utterances and 17.7 text inputs).

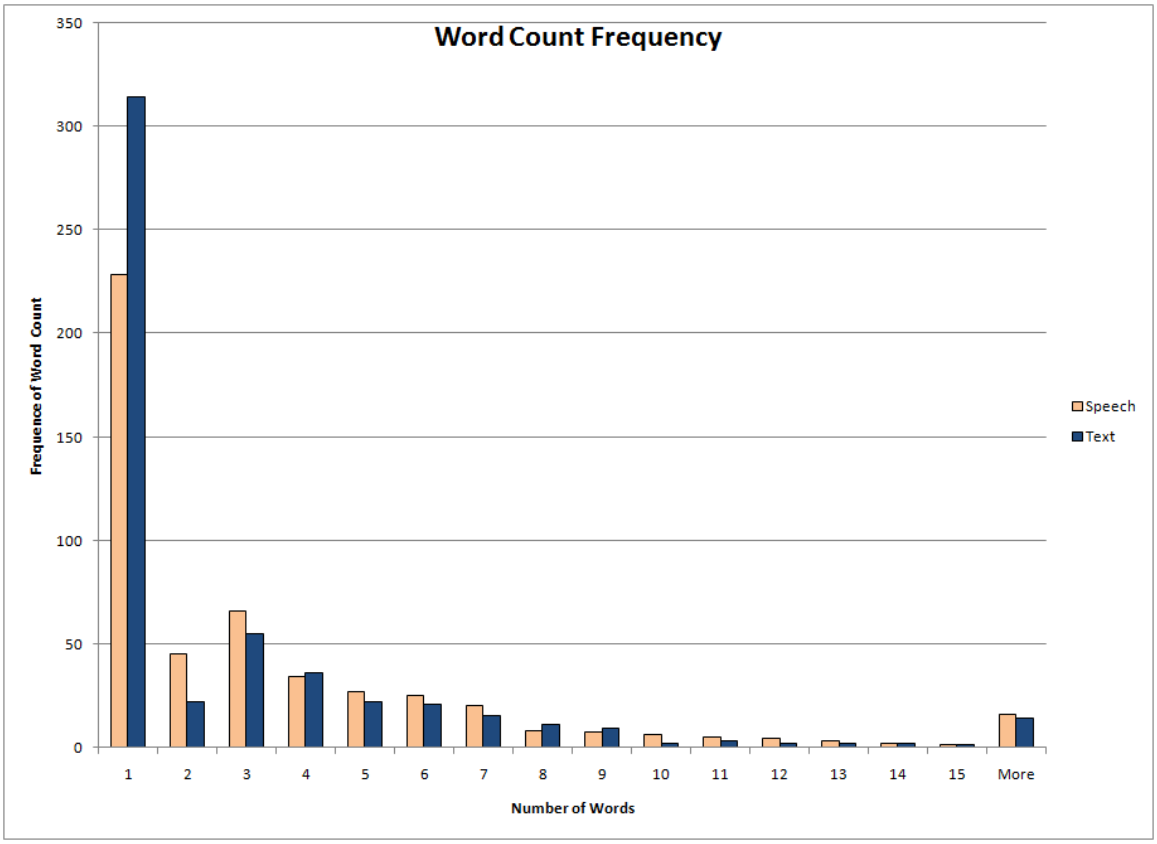


Figure 8-9: A histogram showing the word count frequencies for speech utterances and text input. The x-axis represents the number of words in the speech utterance or text input. The y-axis represents the frequency of the counts.

8.4.3 Color Usage

An interesting qualitative result from the study was that participants did not switch ink color very often. Analyzing the color data in more detail we found that 258 of the 468 strokes were drawn with the default black pen. Participants switched colors only 18 times, and those switches were all done by four users. One user accounted for the vast majority (10 switches). This user had never used a Tablet PC before and started doodling part way through the experiment, although she still answered the questions. The other three users who switched pen colors all had extensive experience with Tablet PCs.

There are several possible explanations for the difference in behavior between the dialogue study and this study. In the dialogue study, the ink was persistent unless explicitly removed by one of the participants. In the evaluation study, the ink is automatically removed after each question is answered. As a result, there is less need for a contrasting color of ink because there are fewer colors in the sketch at any one time. Additionally, the basic device did not have to be drawn by the study participant. In the dialogue study, the participant started with an empty screen. Switching ink color may have been more useful when the participant needed to differentiate the basic components of the design.

8.4.4 Perceived Interface Speed

A reoccurring comment from the study participants was that the text interface was faster or that the MIDOS interface was not as responsive. Curiously, this is not supported by the timing data gathered in the study and therefore must be a perception issue. Most participants did not realize they could interrupt the computer speech and sketching because the directions did not mention this capability. Participants did type answers in the text condition as the computer was speaking, but the participants were careful not to submit the answer until the computer finished the question. Perhaps the text condition seemed faster because the participant was actively engaged typing instead of waiting passively while the computer finished asking the question. The reason this was possible is that typing is a separate channel, so the two can be used simultaneously without conflict (i.e., interrupting).

We investigated the timing data to determine whether or not the text version was faster. First, we examined the duration of the participant's responses. The response duration is the duration of all of the participant's speech, sketching, and typing for a question. The mean response duration for the MIDOS version is 3248ms; the response duration mean for the text version is 7702ms. This difference is significant ($t(688) = 6.604$, $p<.01$). The duration of the speech responses is about half as long as the duration of the text responses.

Similarly, the total duration of the questions, measured from the question begin-

ning to the response end, is smaller for the MIDOS version. The average question and response duration is 13.9s for the text version and 8.6s for the MIDOS version. This is also significant ($t(688) = 6.77$, $p<.01$).

A final possibility is that the wizard took longer to respond in one of the versions. It was easier for the wizard to anticipate the participant's reply in the text version because the wizard could see the text as the user typed. Occasionally a user in the speech and sketching condition would change her answer at the last second; a quick answer change in the text version was impossible. The data shows that the wizard's response time was not significantly different between the two version, in fact, the response time was slightly faster in the MIDOS version. The mean was 8125ms for the MIDOS version and 8559ms for the text version ($t(688)=-0.877$).

All of these statistics show that the MIDOS version is faster than the text version even if some users do not perceive it that way.

8.4.5 Rating Data Results

We collected rating data for every question that was asked in the study. We analyzed the data for differences based on the number of times the question was repeated, differences between the MIDOS and text versions, and differences in ratings between types of questions.

Not surprisingly, the ratings for questions decreased as the question was repeated. The mean rating for a question the first time it was asked was 3.45, the second time it was asked it was 3.14, and third time and beyond was 2.87. The difference between the first and second time is statistically significant ($t(902)=3.88$, $p<.05$), as is the difference between the second and the third time and beyond repetitions ($t(389)=2.33$, $p<.05$).

There was no difference in ratings between the text and MIDOS versions; the rating average for the two versions were nearly identical. The mean rating for the MIDOS version was 3.296, and the mean for the text version was 3.298. This indicates that the users were rating the questions and not the method they were using to answer them.

Type	Mean	Median	Variance
angle	3.30	3	1.38
bounce	3.20	3	1.10
collision	3.32	3	1.35
collision location	2.87	3	0.84
distance	3.08	3	1.31
next	3.58	4	0.89
pulley	3.21	3	1.03
rotation distance	3.25	3	0.41
rotational velocity	3.22	3	1.37
spring direction	3.84	4	1.06
spring end	2.92	3	1.74
spring length	3.46	4	0.87

Table 8.2: Question ratings for the different types of information requests.

We analyzed the questions ratings to determine if there were significant differences between question types. An ANOVA analysis reveals that the differences are significant ($F = 1.83$, $F_{critical} = 1.80$, $p < .05$). Looking at the average ratings, shown in Table 8.2, the question about how far a spring stretches and the question about what happens next get high ratings. The questions that ask the user to specify how a collision occurs or what happens when a spring stretches as far as it can, get low ratings. Both the question ratings and the feedback from participants show that asking intelligent, interesting questions is valued by the user, although asking repetitious questions is not.

8.5 Study Summary

The results of the MIDOS evaluation study show:

- Participants had a natural conversation with the system about the design with long detailed answers similar to the descriptions in our dialogue study.
- The physics simulator was good enough to support the interaction.
- Participants preferred MIDOS to the text only version.

- Some participants perceived that the text version was faster, which was demonstratably false.
- Participants gave lower ratings to repeated questions.

Chapter 9

Related Work

9.1 Our Previous Work

Our research group has focused on sketching interfaces. Sketching is a powerful modality for capturing designs, enabling users to quickly draw a device in a familiar modality. To date our group has developed systems that understand sketches in a variety of domains [5, 29, 52, 46].

Our previous system, ASSIST [5], lets users sketch in a natural fashion and recognizes mechanical components (e.g., springs, pulleys, axles, etc.). Sketches can be drawn with any variety of pen-based input (e.g., Tablet PC). ASSIST (see Figure 9-1) displays a "cleaned up" version of the user's sketch and interfaces with a simulation tool to show users their sketch in action.

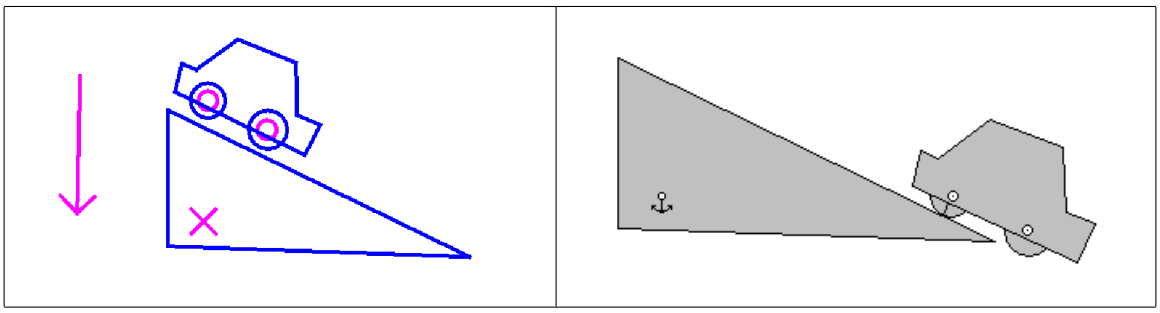


Figure 9-1: The left image shows the sketch in ASSIST. The right image shows the simulation.

ASSISTANCE[45] was a previous effort in our group to combine speech and sketch-

ing. It built on ASSIST[5] by letting the user describe the behavior of the mechanical device with additional sketching and voice input. More recently we built a system [2] that let users simultaneously talk in an unconstrained manner and sketch. This system had a limited vocabulary and could not engage the user in a dialogue, limiting its ability to interpret the user's input.

9.2 Multimodal User Interfaces

Multimodal user interfaces originated with Bolt's "Put-That-There" system. Working in the domain of rescue mission planning, Bolt's system used pointing gestures to resolve designated keywords in the speech [11]. The field has gradually grown to include more interesting and complex non-verbal input.

QuickSet [47] is a collaborative multimodal system that recognizes sketched icons built on an agent-based architecture. The user can create and position items on a map using voice and pen-based gestures. For example, a user could say "medical company facing this way <draws arrow>." QuickSet is command-based, targeted toward improving efficiency in a military environment. This differs from our goal of creating the most natural user interface possible. In contrast to our system where the user starts with a blank screen, QuickSet is a map-based system and the user starts with a map to refer to. QuickSet uses a continuous speaker-independent speech recognition system like MIDOS. QuickSet differs from our system in several ways: it provides users a map to refer to, and does not provide the multimodal dialogue capabilities for the computer.

There are several other related projects[22, 47] that involve sketching and speech, but they are focused more on a command-based interaction with the user. In our system, speech augments the sketching; in other systems, the speech is necessary to the interaction.

Several existing systems allow users to make simple spoken commands to the system [20, 37]. We had many instances of users writing words and speaking them, which is very similar to the types of input that [37] handles. Kaiser et al. describe

how they can add new vocabulary to the system based on handwritten words and their spoken equivalents of the type that appear in Gantt schedule-charts [38].

All of these systems have benefitted from a series of empirical studies of multimodal communication. Oviatt et al. document users' multimodal integration patterns across speech and pen gestures in [49].

9.3 Multimodal Dialogues

Focusing explicitly on managing multimodal dialogues, Johnston et al. describe MATCH in [36]. MATCH includes finite state transducer-based component for combining multimodal inputs, including speech, sketch, and handwriting, in the domain of map-based information retrieval. MATCH's dialogue manager enables a goal-directed conversation, using a speech-act dialogue model similar to [50]. This tool provides some multimodal dialogue capabilities, but it is not a sketching system and has only text recognition and basic circling and pointing gestures for the graphical input modality. Some recent work on multimodal reference resolution uses a greedy algorithm that uses linguistic and cognitive principles to efficiently resolve the references [14].

Another system [13] allows users to query a real estate database with a multimodal user-driven dialogue (speech and sketching), employing a probabilistic graph-matching approach to resolve multimodal references. In a user study, this approach proved effective in resolving ambiguous gesture inputs. Their study, like ours, highlighted the importance of disfluencies in the user's speech.

There has been significant work on multimodal output, but it has focused on generating combinations of speech, images, text, and gestures by an avatar or robot. Mel the robotic penguin uses speech and gestures in a conversation with a user to explain a device [53]. The modalities used are different from our work and Midos users provide the system with the information that guides the dialogue. The COMIC system [24] focuses on generating multimodal dialogues for an avatar including speech output and pointing gestures. Most relevant is their work on interleaving speech and

avatar animation [23] which takes a similar approach in timing the outgoing speech and aligning the other modalities accordingly. However, the main focus of their work is on supporting parallel output and planning of the multimodal dialogue. Our system does not require this level of planning to produce the required output. We produce the strokes to display along with speech instead of avatar animations and speech.

The system in [12] is used to animate two agents that communicate using speech and coordinated gestures. The two output modalities are different from our system, but have some important similarities. In both systems, both modalities influence the combined output. Specifically, both systems must adjust the output based on the duration of different output events – the speech and the gestures and the speech and the sketching.

Multimodal output is also used in several other systems. The WIP system generates device instructions that are multimodal illustrated texts [7] containing images and text. The text includes references to the images. Another system [8] uses a set of rules and heuristics to produce a page layout of text and images. The structure and references in the text determine the sections and the formatting. Related sections of text are displayed using similar styles. This is analogous to our use of the same highlighter color when identifying similar objects. WIP and the system in [8] deal with text layout and images instead of our system's generated questions and identification of objects on a shared drawing surface.

Medical images, a text display and speech are coordinated by the multimodal system in [18]. The layout of the visual display provides constraints on the spoken output. Textual data is highlighted as it is verbally referenced; one of the constraints is that the text should be highlighted in coherent areas. The information displayed differs from our system, but the coordination between the display and speech is similar. In our system, we focus on the highlighting parts of the dynamic sketched objects. Again, our output is a shared medium and we ask the user questions based on the system state rather than presenting fixed data to the user.

Giuliani [25] provides a way to specify speech and gestures for a human-robot interaction in an XML format. The format supports specific start and end time

information for the gestures. Since our modalities require less information to generate the output, we can use our simple format and calculate the exact timing information as needed.

Spoken dialogue management is an important component of natural user interaction. We build upon the previous research [3, 26, 50], some of which is relevant to managing a multimodal dialogue about design. Ideas such as turn taking and determining which information you still need from the user, are still applicable. The situation is made more complicated by the additional modalities and our desire to make the interaction as natural as possible. In particular, the open ended interaction means that the information the system requires is not known a priori as it is in other domains such as airline reservations.

9.4 Querying the User

MIDOS is not the first system to ask the user questions to clarify their intent. The Peridot system [42] is a tool for creating user interfaces. Based on a set of inference rules, Peridot generates text-based questions to clarify the user's design. The user responds by typing an answer, and Peridot may ask follow-up questions. For each question, it specifies what the acceptable answer choices are, for example "yes," "no," "almost," and "quit." The system shares many challenges with MIDOS including selecting the most relevant question, asking the question in a straight-forward manner, and interpreting the user's response. MIDOS differs in both the multimodal nature of its questions and the user's reply and its ability to accept open-ended user responses.

9.5 Wizard-of-Oz Studies

Wizard-of-Oz studies [17] are common and have been conducted in situations where the wizard simulates both pen and speech data [48, 49]. In those studies, the pen input was not open ended, and the wizard had a good idea of what the user would draw.

In our dialogue study, the pen input was open ended and each participant had a unique design project that they described. These factors would have made it difficult for a wizard to create a smooth and natural interaction.

The evaluation study of MIDOS used a fixed set of devices and had a limited number of questions and possible responses. This allowed the study to use a wizard for both speech and text recognition without affecting the responsiveness of the system for the study participants.

9.6 Qualitative Physics Simulators

Our approach to the physics simulator drew ideas and approaches from several qualitative simulators. In particular, we used degree of freedom analysis from [39], order based math computations from [43], and got the inspiration for physics "events" from [55]. Our physics simulator differs from other qualitative simulators because it depends on the user to supply additional information and therefore can make do with less elaborate physics computations.

Chapter 10

Future Work

MIDOS takes a novel approach to generating a multimodal dialogue using a qualitative physics simulator to generate questions about simple mechanical devices. Our work has many possible extensions and avenues for new research. This chapter discuses future work in the sketch and speech input, the sketch and speech output, the core components of MIDOS, and new domains.

10.1 Sketch Input

There are several ways that the sketch input to MIDOS can be improved. Principally, MIDOS can be extended to handle the initial sketch of the device. This could be accomplished by integrating an existing sketch recognition system such as LADDER [30] with MIDOS. The shapes in MIDOS's domain are within the capabilities of current sketch recognition systems.

In order to add this capability, several key issues will need to be addressed, including differentiating strokes and accounting for inaccuracies in the sketch. MIDOS would have to determine whether a stroke is part of an object in the sketch or an annotation stroke. The distinction is important because object strokes are persistent, but annotation strokes should be deleted when they are no longer needed. MIDOS currently starts with a clean sketch enabling it to easily determine which shapes are touching each other. Starting with sketched input makes such determinations more

difficult because of the inherent inaccuracies in the sketch.

MIDOS can currently recognize complex paths that the user draws, but treats them as though they were a line from the starting point to the ending point. Several of the participants in the evaluation study (see Chapter 8) drew complex paths for objects in the sketch to follow. The usability of MIDOS could be improved if it more accurately interpreted the user's complex paths.

The initial user studies revealed the importance of ink color in explanations (Section 3.2.4). Ink of the same color indicated a relationships between components, and ink of a different color indicated a new topic. Our more recent evaluation study found color was used less frequently in MIDOS (Section 8.4.3). Even if it has a reduced role in MIDOS, the system should pay special attention to the user's choice of color. Color choice may play an increased role if the system is extended to handle the initial sketches of the devices because of the need to differentiate objects and annotations.

MIDOS currently lacks editing features once shapes are created. In the future, the user should have the ability to move and resize shapes using the stylus and/or speech. These capabilities would be important if the user needs to correct inaccuracies in an initial sketch of a device created in MIDOS. The physics simulator could be connected to these manipulations so that if one end of a pulley was moved, the other side would move appropriately. Or, for example, if a user moved a block that was connected to a spring, the spring could stretch or compress as the block moved.

The ability to directly manipulate the device components would allow for a different variety of questions and response. Questions could be answered by dragging the bodies to a new location. Questions could be asked by moving bodies to new locations and showing an animated version of collisions. Showing animations would clarify some of the more complex questions that MIDOS asks by visually demonstrating how a collision would occur. This ability might make the interaction more engaging for the user while maintaining the symmetric quality of the interaction.

Handwriting recognition is another area where the capabilities of MIDOS could be improved. Some multimodal systems tie handwriting and speech together, notably [37]. The evaluation study had one instance of a user writing words in addition

to speaking them. Although the simple mechanical devices did not require handwriting, our earlier dialogue study (Chapter 3) had many instances of handwritten text. Design sketches in some domains contain important details in handwriting such as component values in electric circuit diagrams. In order to integrate handwriting recognition with MIDOS, the system will have to determine which parts of the sketch are objects and which parts are handwriting. The location, drawing characteristics, and surrounding context may help with this task.

10.2 Speech Input

The matching that we do between the incoming speech recognition results and the expected speech could be improved to allow the users to say a more diverse set of utterances. For example, WordNet [21] could be used to take a base form of a word and determine alternative phrases. This would allow MIDOS to recognize synonyms for words like "yes" without having to enumerate them explicitly. Similarly, the recognized speech could be parsed and stemmed to isolate verbs and nouns to aid in the matching.

The system should also use natural language processing techniques to parse multi-part utterances from the user and separate them into their component pieces. Using timing information for the speech pieces and any sketching input, the system could form groups of related speech and sketching and proceed in a similar fashion to the current system.

During the studies, users provided clues to their uncertainty and provided narrations that didn't relate directly to the sketch (as described in Section 3.2.8). For example, a user said "now I'm going to switch ink colors." Although it is not directly about the sketch, this information can provide the system another alignment point between the speech and the sketching. The users also expressed uncertainty about the designs by saying things like "I'm not sure this is right..." A challenge for the system is to take this uncertainty or conflicting information into account.

The speed and prosody of the user's speech could be another source of information

for the system. Qualitative observations from the dialogue and MIDOS user studies showed that if a user is speaking quickly, she most likely had a great deal that she wanted to say about an idea. The more quickly and loudly she spoke, the more confident she was with her answers. If she was speaking slowly then she was more likely to be thinking about the design or what to draw next. A more advanced speech recognizer might be able to take advantage of these patterns. Changes in tone could be used to detect new topics, determine user confidence in an answer, or decide that a question needs to clarified.

The MIDOS evaluation study results showed that the users preferred to use relative terms to describe directions and distances. MIDOS should be modified so that it can translate phrases such as "up," "down," "down and to the left," and "half the distance to this block" into numerical values that can be processed.

When two humans discuss a design, they can establish new terms and vocabulary to refer to different parts of a design. MIDOS should have a similar capability that would allow it associate a particular word with a particular symbol in the sketch. Current multimodal systems are capable of learning new vocabulary based on speech and handwriting [38]. If the system recognized that an unknown word was spoken, it should ask the user to identify the component of the sketch that should be associated with that word. If the system needed to ask a question about that component later in the dialogue, it could use the newly learned word.

10.3 Sketch Output

The computer generated strokes should resemble human strokes as closely as possible. Currently, the different highlighter strokes are drawn as isolated strokes without any higher level of organization. The current method could be extended to include a concept of a computer "hand" that was drawing the strokes. The strokes that are drawn could reflect the current location of the "hand" and minimize the distance the "hand" would have to move.

Other factors could be used to render the computer strokes so that they appear

even more human-like. This could be accomplished by varying the speed of the stroke within the stroke itself, e.g., to slow down at corners. The computer generated strokes could also use varying pressure instead of the current constant pressure which would result in strokes that varied in thickness like user generated strokes.

Currently, if multiple strokes have to be drawn in a fixed window of time, the time is allocated evenly to all of the strokes. Instead, the relative length and time required to draw the strokes should be part of the calculation. If one stroke takes very little time to draw and another takes a significantly longer time to draw, the interaction could flow more smoothly if the allocation of time to strokes was globally optimized so that the overall drawing time was minimized.

MIDOS currently provides verbal acknowledgments of the user's input. It could also provide visual feedback, for example, MIDOS might briefly flex a spring to show that it has understood that the spring expands. These actions are more complex than the current user interface supports. The interface currently only displays the objects and strokes as they are drawn. Each update to the display is stored as part of the sketch. These fleeting updates would require a modified architecture and a change in how the objects are displayed on the screen.

MIDOS can draw arrows to indicate motion. In the future arrows could be replaced or augmented with animations, dotted paths, or by giving the user the ability to drag shapes around on the screen. Any or all of these methods might prove to be effective ways of communicating motion to the user.

Digital ink persistence is an additional area of research. As design sketches get more complicated, determining when to keep and when to erase computer and user generated strokes also becomes an increasingly complex problem. A user study could be conducted to determine if, when, and how quickly digital ink should be erased and determine when it should be persistent.

10.4 Speech Output

During the evaluation study of MIDOS, participants were very averse to interrupting the computer's speech. MIDOS can handle interruptions and will stop the speech output, but users did not try this. The computer cannot determine that the user wants to say something, so the user just waits until the computer finishes speaking. If the computer's speech output were to increase in length, this issue becomes more critical because the longer the computer's speech, the more likely it is that the user will want to interrupt. One way to resolve this situation might be to insert pauses into the outgoing speech to allow the user more opportunities to jump in and interrupt the computer.

User's responses to questions can vary in length and vocabulary. It is possible that by varying the way a question is asked, the system could control the user's response. In the evaluation study, user's responses to questions that could be answered with "yes" or "no" tended to be short, but responses to open ended questions, such as "What happens next?," tended to be longer. At certain points in the interaction, it could be desirable to have a short reply from the user and at other points in the interaction a lengthy explanation might be more appropriate. If MIDOS could exert some influence on the user's replies, the interaction could be more efficient.

10.5 Core System

MIDOS currently handles responses to the question it has asked. Participants in the studies, however, consistently tried to provide more information at once than the system could handle. The user studies showed that even simple questions can elicit lengthy, in-depth replies. This extra information could contain answers to questions that have not yet been asked. For example, the user might say that a shape is not anchored and neither are any of these other shapes. The system should be able to parse this more complex answer and update its physics knowledge accordingly, resulting in a decrease in user frustration and enabling MIDOS to interact with the

user more naturally.

MIDOS currently asks many questions that seem obvious to the user. One way to reduce the number of questions MIDOS asks would be to have the system guess answers to questions and narrate these guesses to the user. If the user detects a misstep, she could interrupt the system and have it rewind to the point where the incorrect decision was made. It is also possible, however that the user just wants to see that part of the simulation again. The system must determine which of these actions the user means, and if the user is pointing out an error, the system must also determine what it guessed wrong. In this interaction style, it would be important to encourage the user to interrupt the system to make the necessary corrections in a timely manner.

MIDOS already stores frequent snapshots of the state. The more challenging part of this interaction is providing an easy interface for the user to control the rewinding of the simulation and developing the natural narration dialogue that would explain the decisions that MIDOS was making to the user. Other challenges in this interaction include deciding which information to ask about, which information to guess about, and what guesses to make.

The physics simulator is currently limited in scope. In the future, it could be extended to handle more shapes, or handle physics calculations with more accuracy, for example, extending the simulator to handle friction or simultaneous rotation and translation. Currently, the physics simulator analyzes the state and determines how far it can move all the bodies. This creates a "jumpy" animation of the device. It might be possible to create a smooth animation of the bodies, however, the lack of timing information in the qualitative simulation may make this difficult. MIDOS uses the physics simulator to generate interesting questions to ask the user; any improvements to the simulator should keep the overall objective in mind.

10.6 New Domains

MIDOS is built in such a way that the same principles can be applied to other domains.

Instead of the physics simulator driving the interaction, domain specific knowledge can be leveraged to come up with questions to ask the user. Research is currently underway to apply the ideas and code from MIDOS to the domain of software programmable radios.

The two main components that need to be changed to apply MIDOS to a new domain are the question generation and the question selection components. Additional interaction techniques and user interface adjustments may also be required.

In order to use MIDOS in a new domain, the physics simulator would need to be replaced with a new information generation component. This component needs to gather the information necessary to form questions about the objects in the new domain. Also, new questions would need to be devised. The system needs to know when the question is relevant, how to phrase the question, and the speech and sketching that are expected in response. The currently available techniques for identifying components and indicating motion may or may not be sufficient in a new domain. Once all the possible questions are generated, the system must select the appropriate question to ask next. The question selection techniques used in the simple mechanical device domain should provide a guide for new domains, but additional factors may be important.

Other parts of the system would also need to be modified. Most of the domain dependent data structures are contained in the physics components, but some modifications to data structures would be required. For example, the system currently writes the system state to an XML file so it can be reloaded and replayed. The file format and data structures would need to be extended to support the objects in a new domain. Additionally, the current C# user interface may require the addition of new features for the new domain.

Chapter 11

Contributions

MIDOS creates a novel, symmetric, multimodal, interaction using speech and sketching as the modalities for both the computer and the user. Users describe early stage mechanical devices, and the computer asks a series of questions about the device based on the current state and the output of a qualitative physics simulator. The conversation resolves uncertainties and ambiguities in the sketch, and allows MIDOS to simulate the function of the device.

MIDOS and the user studies revealed several key concepts:

- Simple questions are powerful: Asking simple questions can lead to long, detailed responses from the user.

- Cross-modality coherence: Speech and sketching that occur at the same time are about the same topic. Users will keep the modalities synchronized by pausing a modality as necessary.

- Color choices are deliberate: Using the same pen color indicates similarity, and changing the pen color indicates a new topic or component.

The output MIDOS generates mirrors the user's input as closely as possible, using the same modalities and integrating them in the same manner as the user, working to ensure that the modalities pause at the appropriate times. MIDOS selects questions based only on its physics simulator, the current state, and the recently asked

questions. This creates a dynamic dialogue that depends entirely on the answers the user provides.

Our evaluation study showed that MIDOS produces the same types of conversations as our initial human-human interaction studies. Participants provided long, detailed answers to the computer generated questions. The users preferred using MIDOS to a text based alternative.

MIDOS brings us a step closer to having a computer design partner that can ask questions about a design in a similar fashion as a human, and by doing so help the user clarify and refine her ideas.

Appendix A

Expected Speech and Sketching

This appendix contains the expected speech and expected sketching for each information request. A sample question and image are also shown for each information request. The order the requests are listed in reflects the priority of the different information requests.

A.1 Anchor Information Request

Figure A-1: Anchor information request question: "Is (this shape) anchored?"

Yes Speech
yes
yeah
of course
affirmative

Table A.1: Anchor information request expected yes speech.

No Speech
no
nah
of course not
negative

Table A.2: Anchor information request expected no speech.

A.2 Bounce Information Request

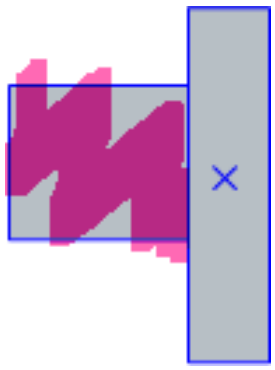


Figure A-2: Bounce information request question: "Does (this shape) bounce after the collision?"

Bounce Speech
yes it bounces
yeah
yes
of course
of course it bounces
it bounces
bounces
it does bounce

Table A.3: Bounce information request expected bounce speech.

Stop Speech
no it stops
no
no it does not bounce
no it doesn't bounce
it does not bounce
it doesn't bounce
stops
does not bounce
doesn't bounce
it stops
no it stops
it doesn't move
it does not move
does not move
it stays
stays
it stays there
of course not

Table A.4: Bounce information request expected stop speech.

A.3 Angle Information Request

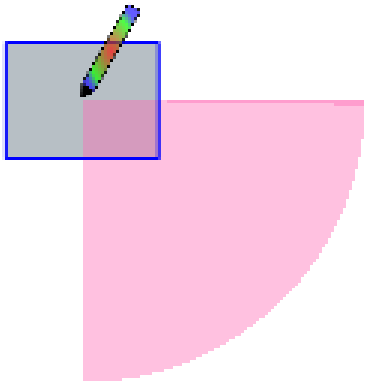


Figure A-3: Angle information request question: "Which of {these directions} does this shape move in?"

The expected stroke type for the angle information request is a path.

Angle Speech
this direction
like this
it moves in this direction
it goes in this direction
it goes like this
it moves like this

Table A.5: Angle information request expected angle speech.

Stationary Speech
it doesn't move
it does not move
does not move
it stays
stays
it stays there
it stops
stops
it stops there
it is stationary
it's stationary
stationary
it does not go in any direction
it doesn't go in any direction

Table A.6: Angle information request expected stationary speech.

A.4 Rotation Direction Information Request

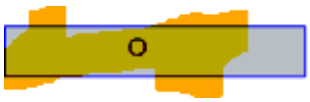


Figure A-4: Rotation direction information request question: "What direction does (this shape) rotate in?"

The expected stroke type for the rotation direction information request is a path.

Direction Speech
this direction
like this
it moves in this direction
it goes in this direction
it goes like this
it moves like this

Table A.7: Rotation direction information request expected direction speech.

Clockwise Speech
clockwise
it moves clockwise
it rotates clockwise
it goes clockwise
it moves in a clockwise direction
it rotates in a clockwise direction
it goes in a clockwise direction

Table A.8: Rotation direction information request expected clockwise speech.

Counterclockwise Speech
counter clockwise
it moves counter clockwise
it rotates counter clockwise
it goes counter clockwise
it moves in a counter clockwise direction
it rotates in a counter clockwise direction
it goes in a counter clockwise direction
counterclockwise
it moves counterclockwise
it rotates counterclockwise
it goes counterclockwise
it moves in a counterclockwise direction
it rotates in a counterclockwise direction
it goes in a counterclockwise direction

Table A.9: Rotation direction information request expected counterclockwise speech.

Stationary Speech
it doesn't move
it does not move
does not move
it stays
stays
it stays there
it stops
stops
it stops there
it is stationary
it's stationary
stationary
it does not go in any direction
it doesn't go in any direction

Table A.10: Rotation direction information request expected stationary speech.

A.5 Rotational Velocity Information Request

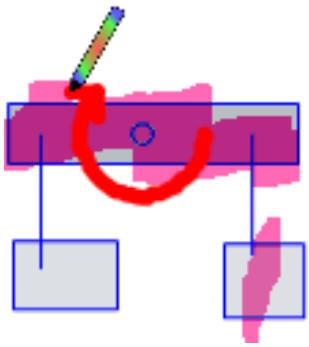


Figure A-5: Rotational velocity information request question: "I can not determine the rotation of (this shape) now. This shape causes a (clockwise rotation.) <short pause> <clear stroke> This shape causes a (counterclockwise rotation.) <short pause> <clear stroke> <short pause> <clear strokes> At this instant what direction does (this rotate in) or is it balanced?"

The expected stroke type for the rotational velocity information request is a path.

Direction Speech
this direction
like this
it moves in this direction
it goes in this direction
it goes like this
it moves like this

Table A.11: Rotational velocity information request expected direction speech.

Clockwise Speech
clockwise
it moves clockwise
it rotates clockwise
it goes clockwise
it moves in a clockwise direction
it rotates in a clockwise direction
it goes in a clockwise direction

Table A.12: Rotational velocity information request expected clockwise speech.

Counterclockwise Speech
counter clockwise
it moves counter clockwise
it rotates counter clockwise
it goes counter clockwise
it moves in a counter clockwise direction
it rotates in a counter clockwise direction
it goes in a counter clockwise direction
counterclockwise
it moves counterclockwise
it rotates counterclockwise
it goes counterclockwise
it moves in a counterclockwise direction
it rotates in a counterclockwise direction
it goes in a counterclockwise direction

Table A.13: Rotational velocity information request expected counterclockwise speech.

Balanced Speech
balanced
it is balanced
it doesn't move
it does not move
it doesn't rotate
it does not rotate
it doesn't move
it does not move
does not move
it stays
stays
it stays there
it stops
stops
it stops there
it is stationary
it's stationary
stationary
it does not rotate in any direction
it doesn't rotate in any direction

Table A.14: Rotational velocity information request expected balanced speech.

A.6 Pulley Information Request

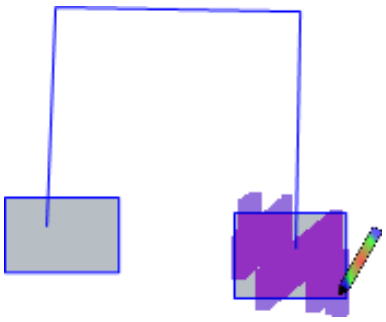


Figure A-6: Pulley information request question: "What direction does (this shape) move in at this instant?"

The expected stroke type for the pulley information request is a path.

Direction Speech
this direction
like this
it moves in this direction
it goes in this direction
it goes like this
it moves like this

Table A.15: Pulley information request expected direction speech.

Balanced Speech
balanced
it is balanced
it doesn't move
it does not move

Table A.16: Pulley information request expected balanced speech.

A.7 Distance Information Request

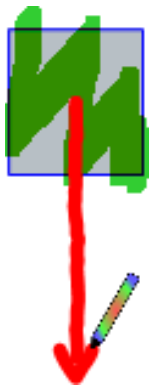


Figure A-7: Distance information request question: "How far does (this shape) (move?)"

The expected stroke type for the distance information request is a path or location.

Forever Speech
it goes forever
it goes off the screen
it keeps going
it moves forever
it moves off the screen
it keeps moving
forever
off the screen

Table A.17: Distance information request expected forever speech.

Distance Speech
it goes here
it goes this far
it goes to here
it moves here
it moves this far
it moves to here
here
this far
stops here
it stops here

Table A.18: Distance information request expected distance speech.

Stationary Speech
it doesn't move
it does not move
does not move
it stays
stays
it stays there
it stops
stops
it stops there
it is stationary
it's stationary
stationary
it does not go in any direction
it doesn't go in any direction

Table A.19: Distance information request expected stationary speech.

A.8 Rotation Distance Information Request

Figure A-8: Rotation distance information request question: "How far does (this shape) (rotate?)"

The expected stroke type for the rotation distance information request is a path.

Forever Speech
it goes forever
it keeps going
it moves forever
it keeps moving
it rotates forever
it keeps rotating
forever

Table A.20: Rotation distance information request expected forever speech.

Rotation Speech
it rotates this far
it rotates like this
it goes this far
it goes like this
it moves this far
it moves like this
this far
like this

Table A.21: Rotation distance information request expected rotation speech.

Stationary Speech
it doesn't move
it does not move
does not move
it stays
stays
it stops
stops
it stays there
it stops there
it is stationary
it's stationary
stationary
it does not go in any direction
it doesn't go in any direction

Table A.22: Rotation distance information request expected stationary speech.

A.9 Spring Direction Information Request

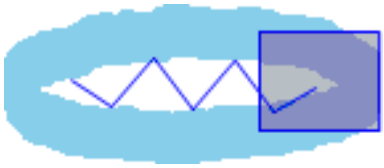


Figure A-9: Spring direction information request question: "Will (this spring) expand or contract?"

The expected stroke type for the spring direction information request is a path.

Expands Speech
it expands
it gets longer
expands
longer
expand

Table A.23: Spring direction information request expected expands speech.

Contracts Speech
it contracts
it gets shorter
contracts
shorter
contract

Table A.24: Spring direction information request expected contracts speech.

Multimodal Speech
it goes in this direction
it moves in this direction
in this direction
this direction
like this

Table A.25: Spring direction information request expected multimodal speech.

Stationary Speech
neither
neither, it stays the same
it stays the same
stays

Table A.26: Spring direction information request expected stationary speech.

A.10 Spring Length Information Request

Figure A-10: Spring length information request question: "How far does (this spring) stretch?"

The expected stroke type for the spring length information request is a path or location.

Multimodal Speech
this far
to here
here
like this

Table A.27: Spring length information request expected multimodal speech.

Expands Speech
it expands to here
it stretches to here
expands here
stretches here
expands
stretches

Table A.28: Spring length information request expected expands speech.

Contracts Speech
it contracts to here
it compresses to here
contracts
compresses

Table A.29: Spring length information request expected contracts speech.

A.11 Spring End Information Request

Figure A-11: Spring end information request question: "(This spring has) reached its maximum length. What happens next?"

The expected stroke type for the spring end information request is a path.

Expands Speech
it expands
it gets longer
expands
longer
expand

Table A.30: Spring end information request expected expands speech.

Contracts Speech
it contracts
it gets shorter
contracts
shorter
contract

Table A.31: Spring end information request expected contracts speech.

Multimodal Speech
it goes in this direction
it moves in this direction
in this direction
this direction
like this

Table A.32: Spring end information request expected multimodal speech.

Stationary Speech
neither
neither, it stays the same
it stays the same
it stops

Table A.33: Spring end information request expected stationary speech.

Indifferent Speech
it doesn't matter
it does not matter
doesn't matter
does not matter

Table A.34: Spring end information request expected indifferent speech.

Reverses Speech
it reverses direction
it reverses
reverses

Table A.35: Spring end information request expected reverses speech.

A.12 Collision Information Request

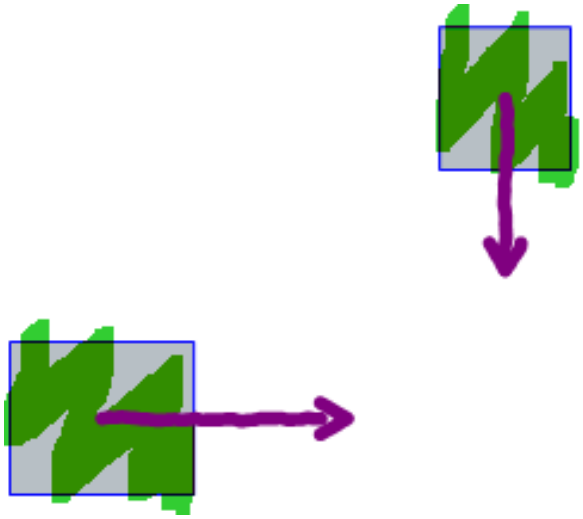


Figure A-12: Collision information request question: "It looks like {these shapes} {will collide, do they?}"

Yes Speech
yes
yeah
of course
affirmative

Table A.36: Collision information request expected yes speech.

No Speech
no
nah
of course not
negative

Table A.37: Collision information request expected no speech.

A.13 Collision Location Information Request

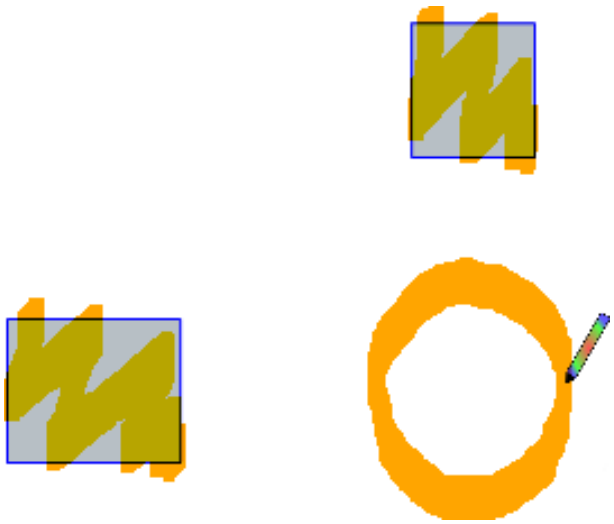


Figure A-13: Collision location information request question: "(These two) (bodies) collide (here.) <long pause> <clear strokes> Where on (this) body does the contact occur?"

The expected stroke type for the collision location information request is a path or location.

Multimodal Speech
it hits here
here
this is the collision location
it collides here
it contacts here
this is
this is the contact location

Table A.38: Collision location information request expected multimodal speech.

A.14 Next Information Request

The question for the next information request is "What happens next?" and the expected stroke type is a path.

Multimodal Speech
this moves in this direction
this moves
this rotates
this way
moves
like this
like goes this like this

Table A.39: Next information request expected multimodal speech.

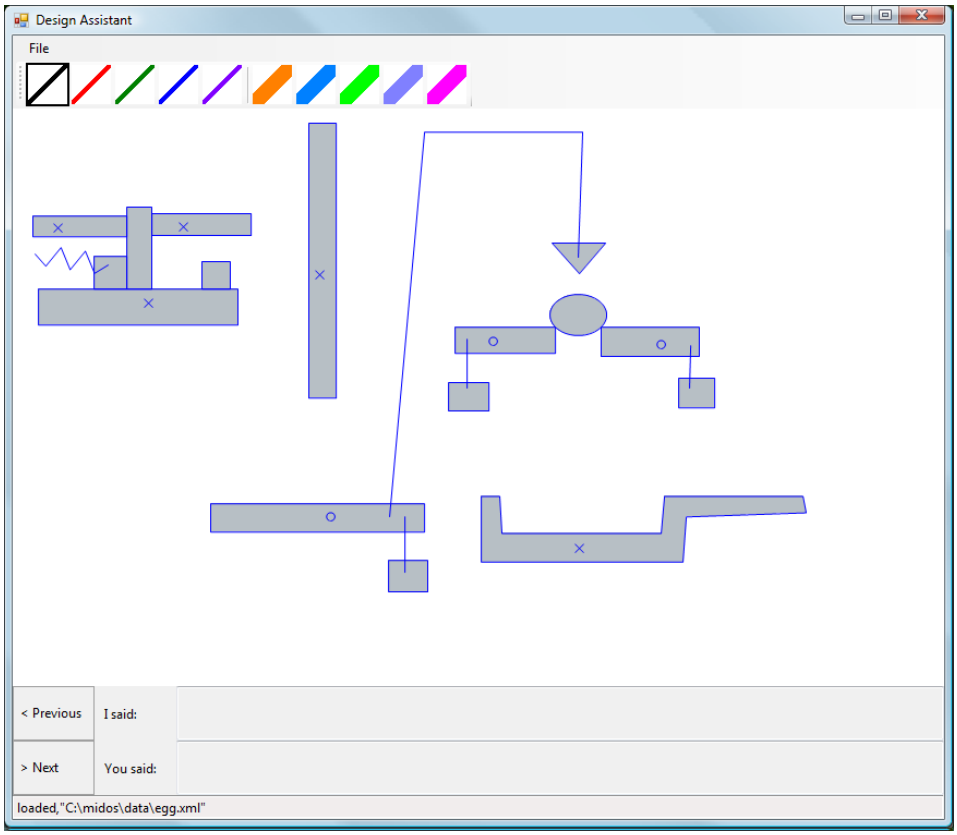
End Speech
that's it
that's the end
that is it
that is the end
nothing else happens

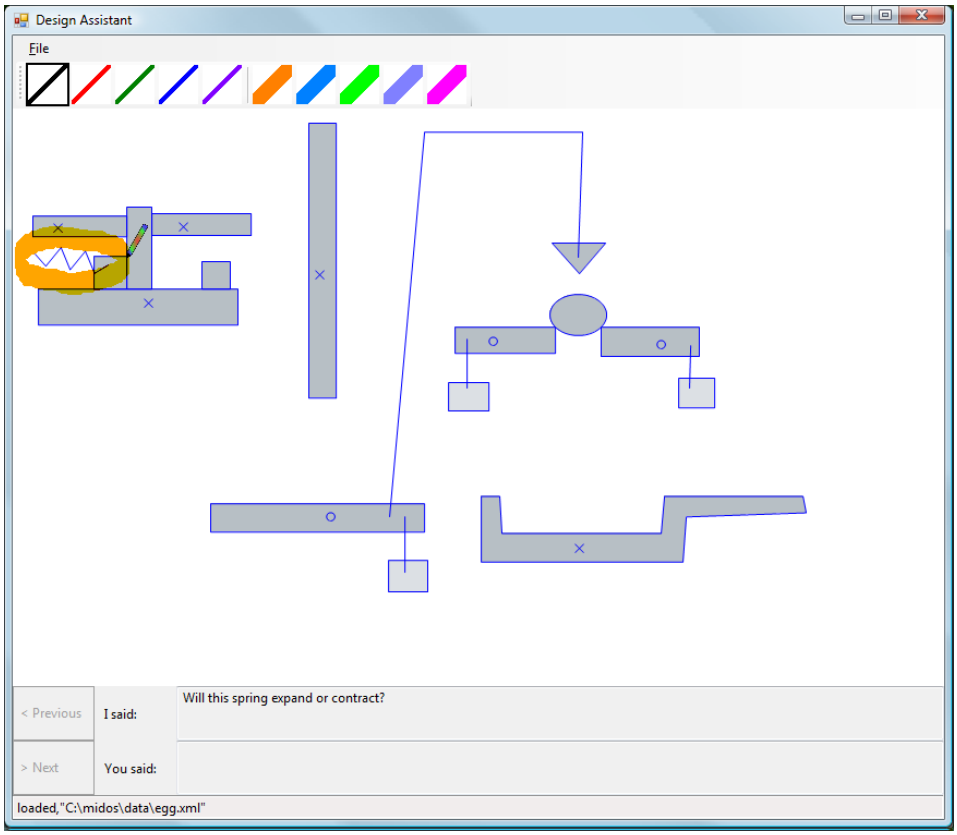
Table A.40: Next information request expected end speech.

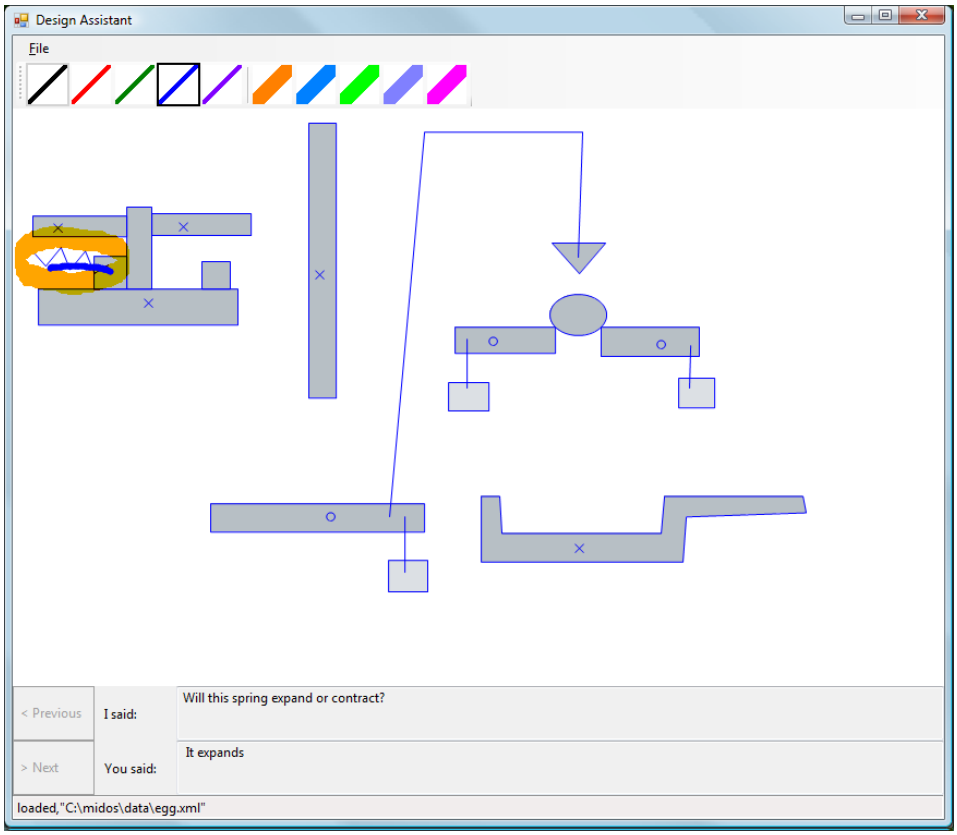
Appendix B

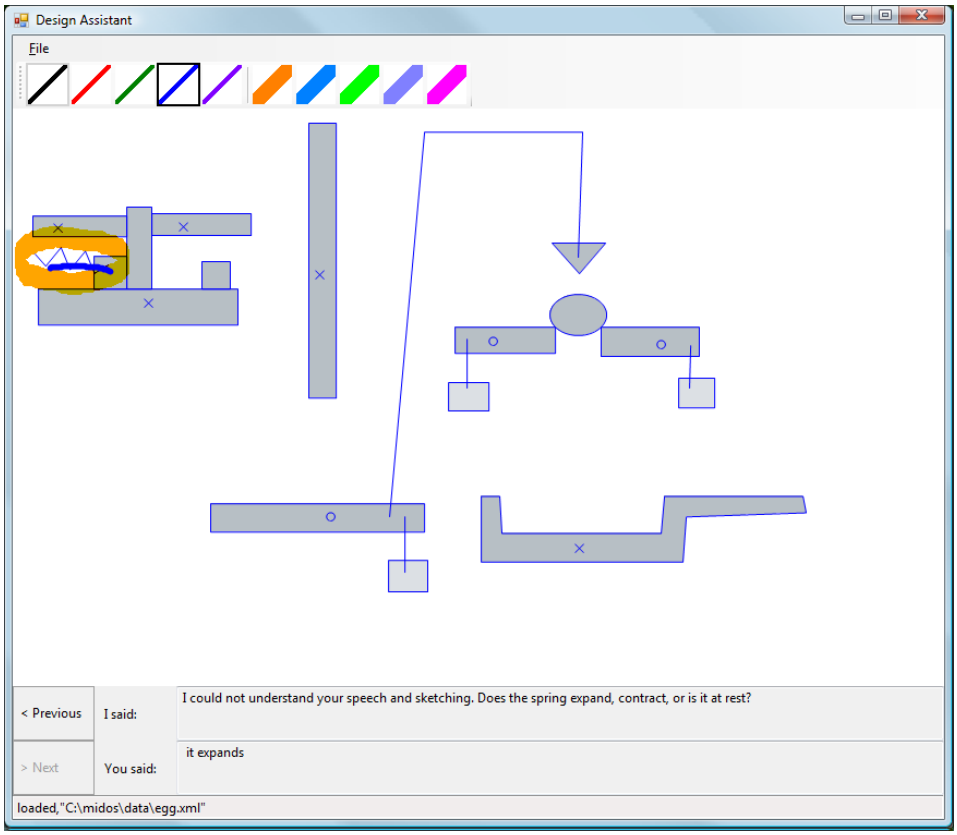
Egg Cracker Example

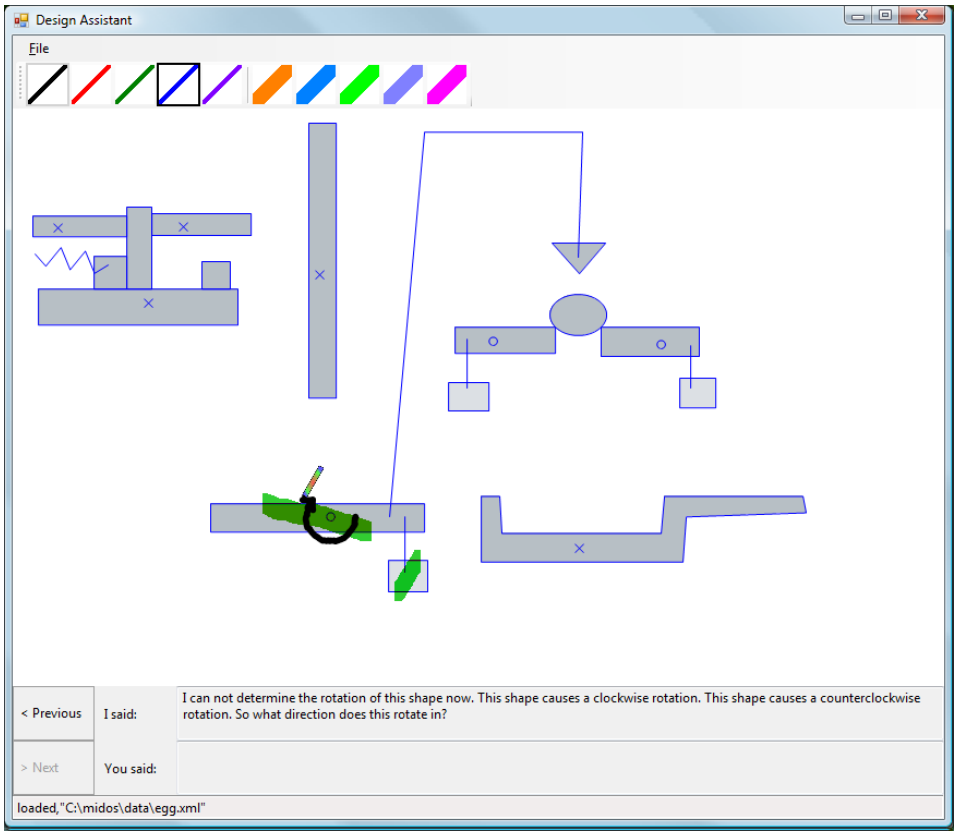
The sequence of images in this appendix illustrates an example of a full interaction with MIDOS. The user describes how the egg cracker device functions and MIDOS successfully simulates it.

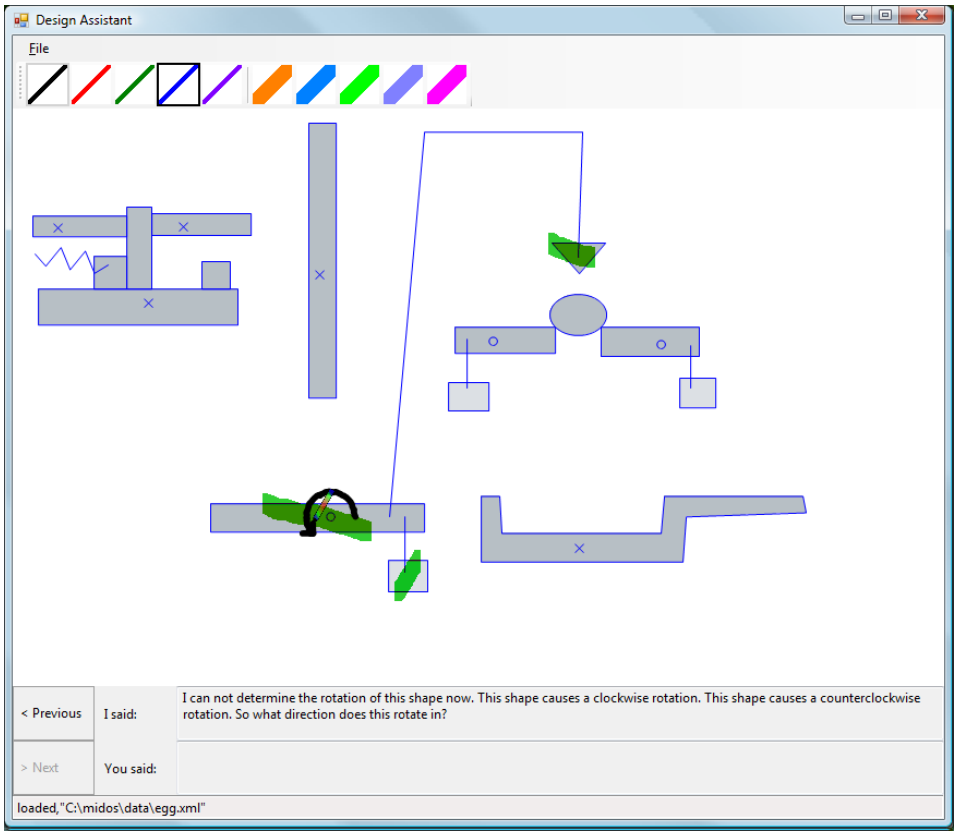


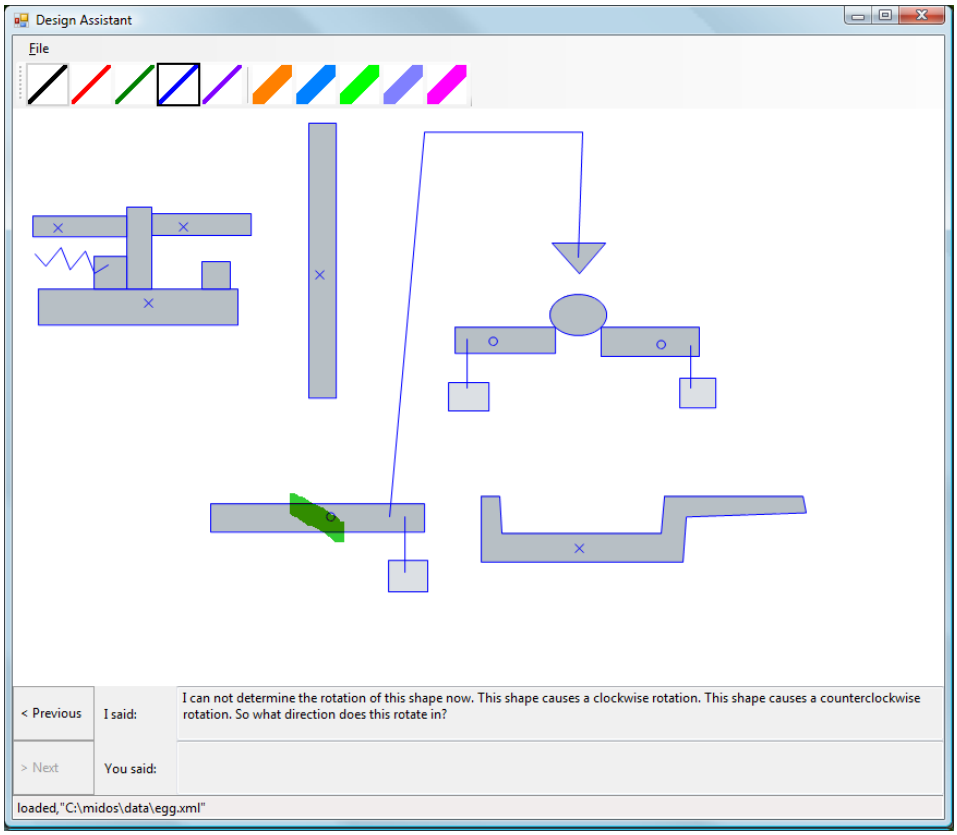

Design Assistant
File
< Previous
I said:
Will this spring expand or contract?
> Next
You said:
loaded,"C:\midos\data\egg.xml"

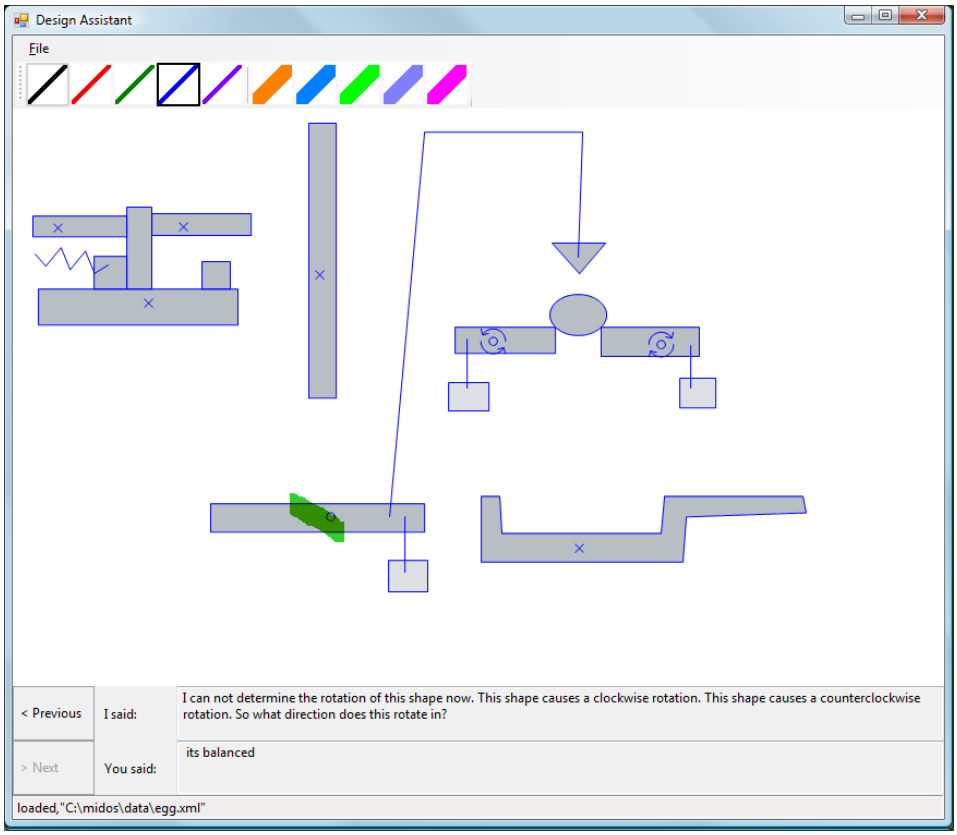

Design Assistant
File
< Previous
I said:
Will this spring expand or contract?
> Next
You said:
It expands
loaded,"C:\midos\data\egg.xml"

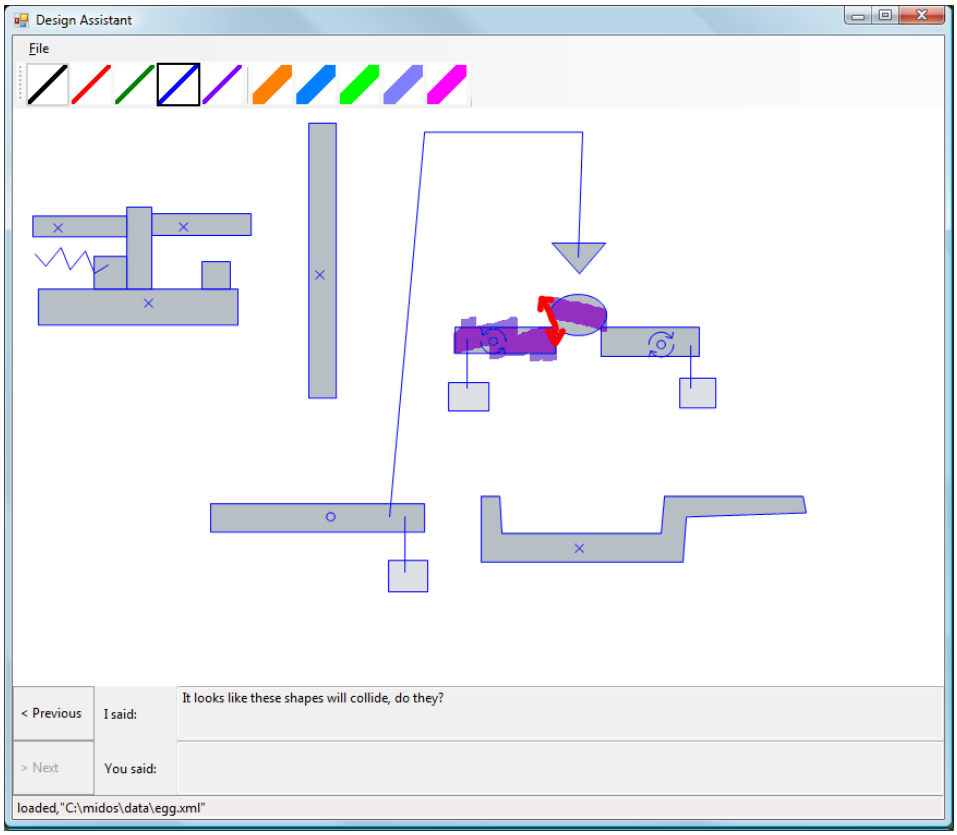

Design Assistant
File
< Previous
I said:
I could not understand your speech and sketching. Does the spring expand, contract, or is it at rest?
> Next
You said:
it expands
loaded,"C:\midos\data\egg.xml"

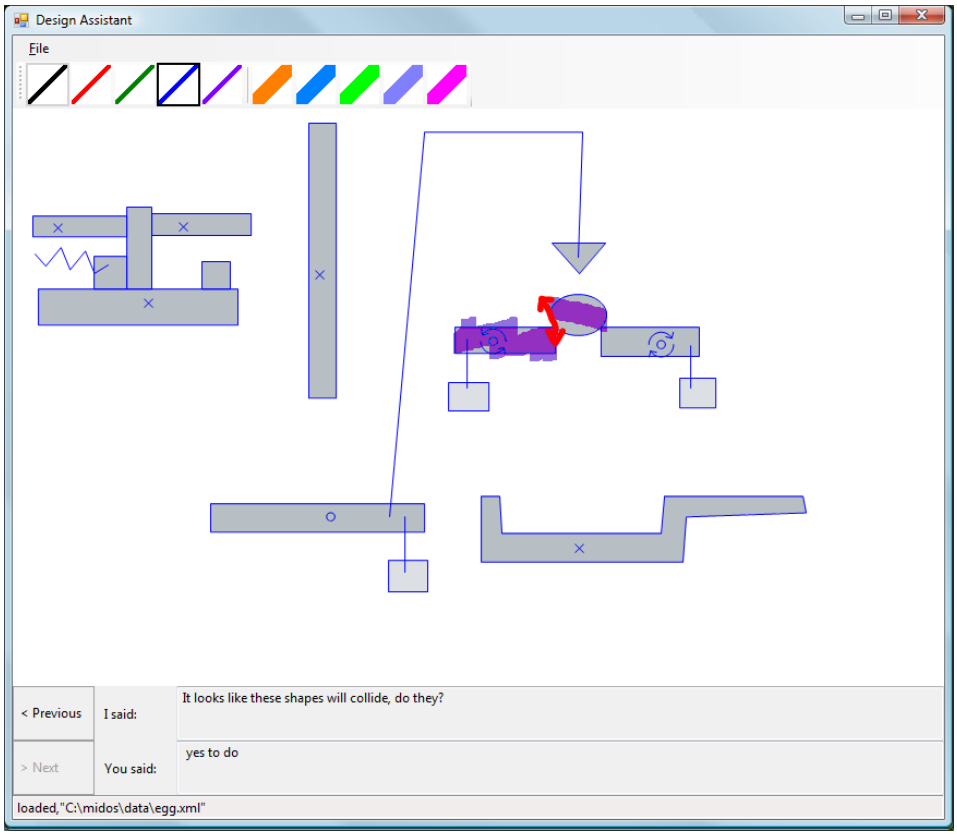

Design Assistant
File
< Previous
I said:
I can not determine the rotation of this shape now. This shape causes a clockwise rotation. This shape causes a counterclockwise rotation. So what direction does this rotate in?
> Next
You said:
loaded,"C:\midos\data\egg.xml"

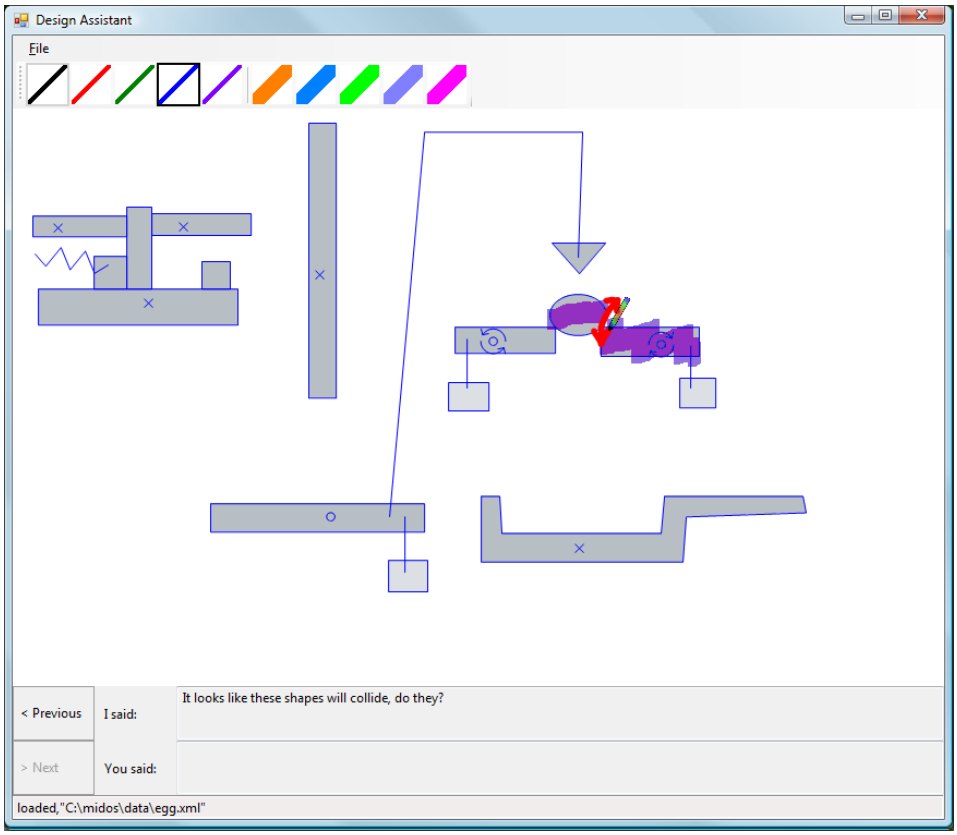

Design Assistant
File
< Previous
I said:
I can not determine the rotation of this shape now. This shape causes a clockwise rotation. This shape causes a counterclockwise rotation. So what direction does this rotate in?
> Next
You said:
loaded,"C:\midos\data\egg.xml"

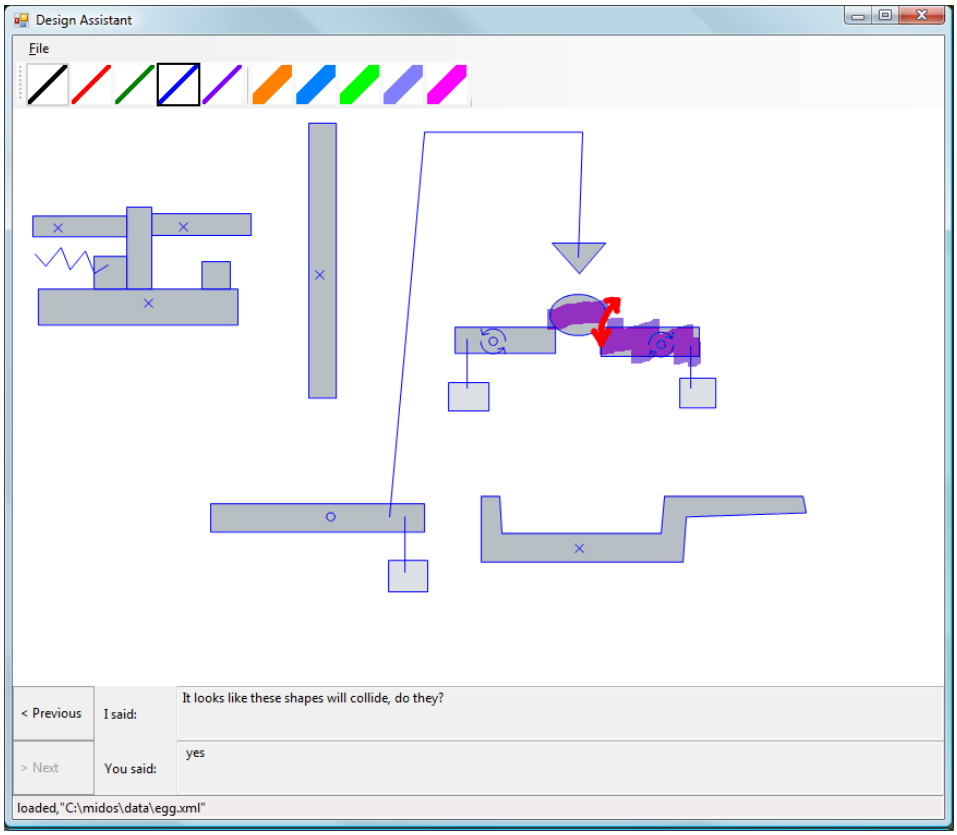

Design Assistant
File
< Previous
I said:
I can not determine the rotation of this shape now. This shape causes a clockwise rotation. This shape causes a counterclockwise rotation. So what direction does this rotate in?
> Next
You said:
loaded,"C:\midos\data\egg.xml"

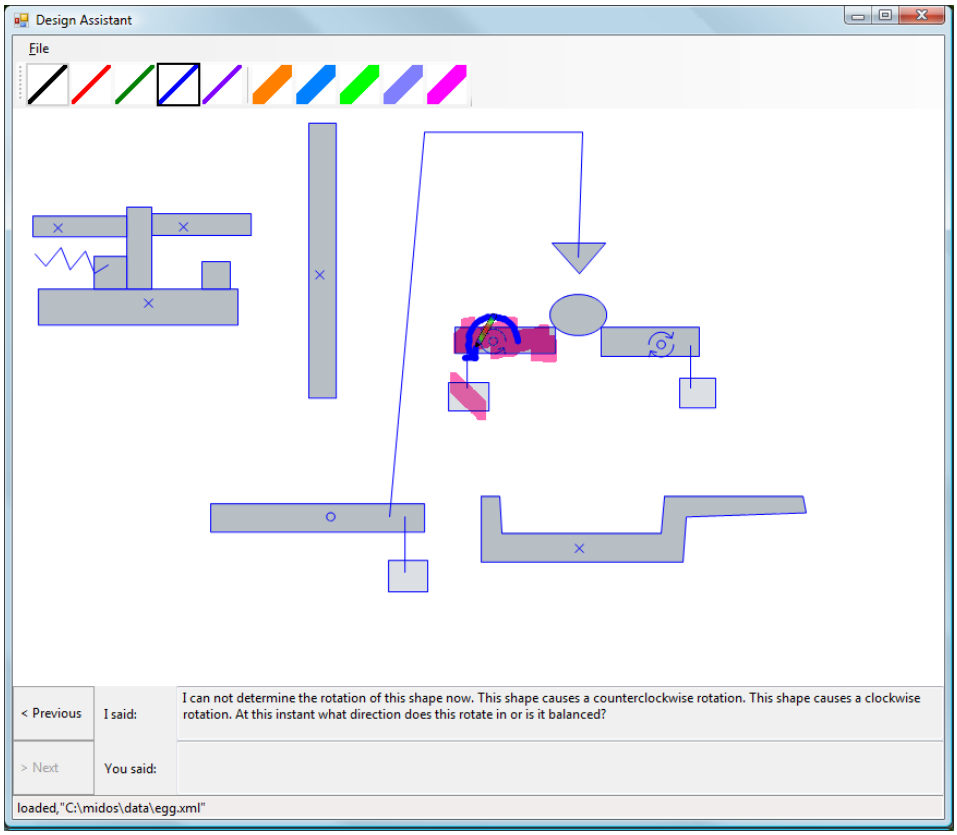

Design Assistant
File
< Previous
I said:
I can not determine the rotation of this shape now. This shape causes a clockwise rotation. This shape causes a counterclockwise rotation. So what direction does this rotate in?
> Next
You said:
its balanced
loaded,"C:\midos\data\egg.xml"

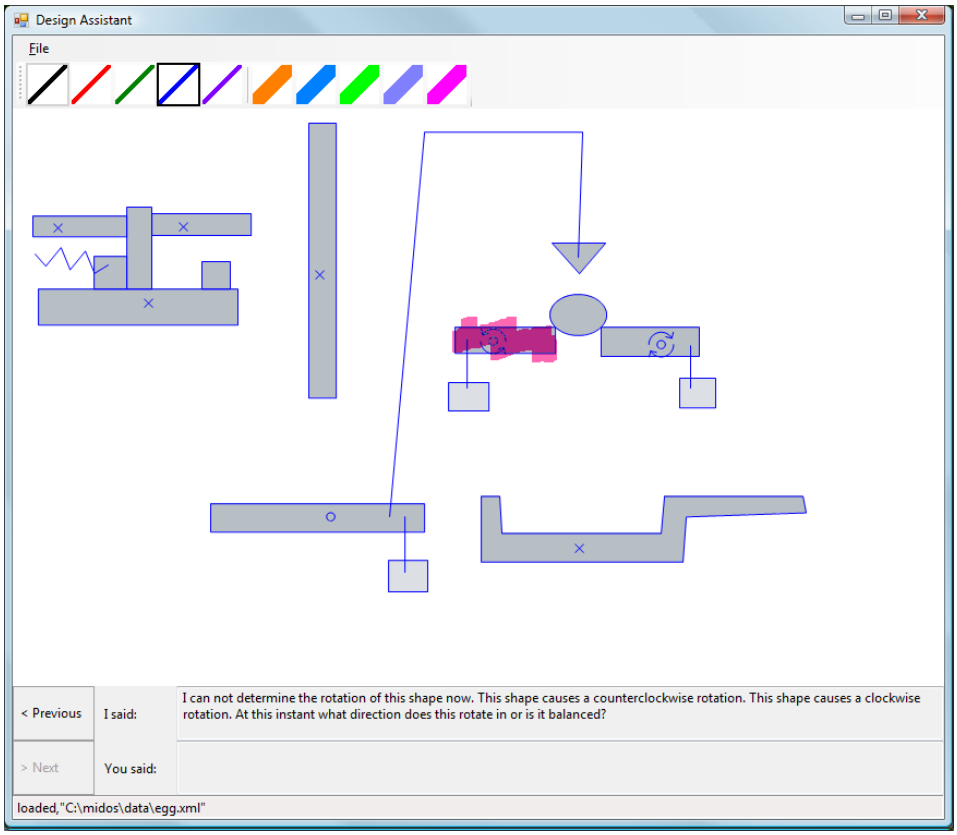

Design Assistant
File
< Previous
I said:
It looks like these shapes will collide, do they?
> Next
You said:
loaded,"C:\midos\data\egg.xml"

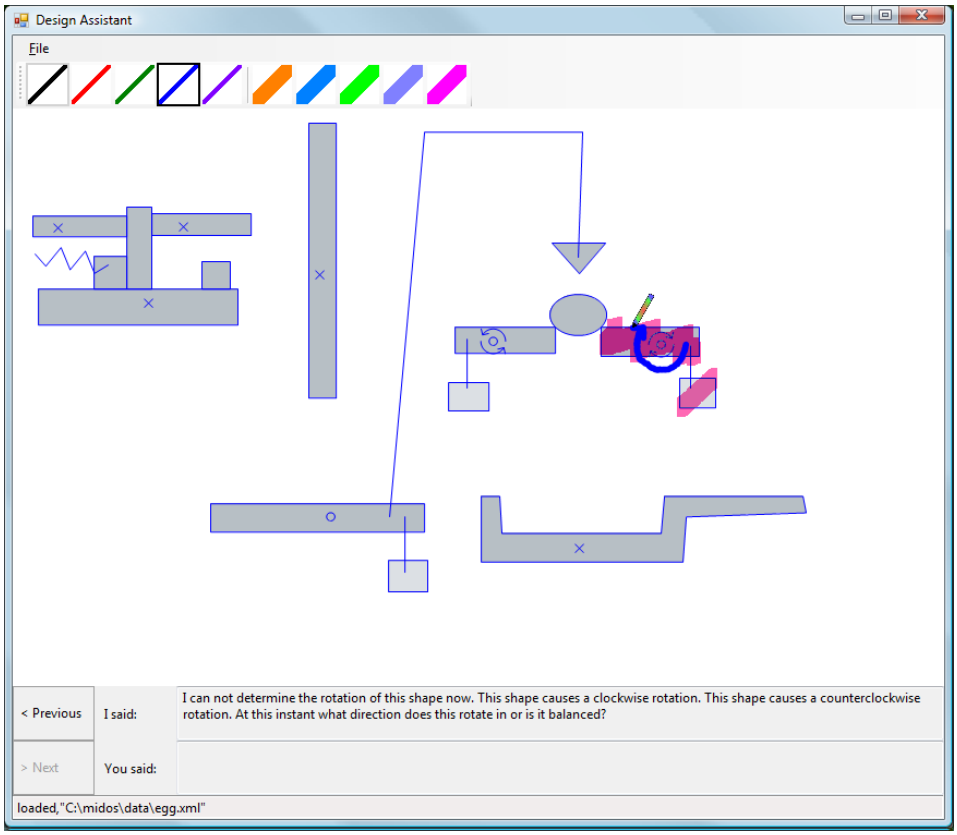

Design Assistant
File
< Previous
I said:
It looks like these shapes will collide, do they?
> Next
You said:
yes to do
loaded,"C:\midos\data\egg.xml"

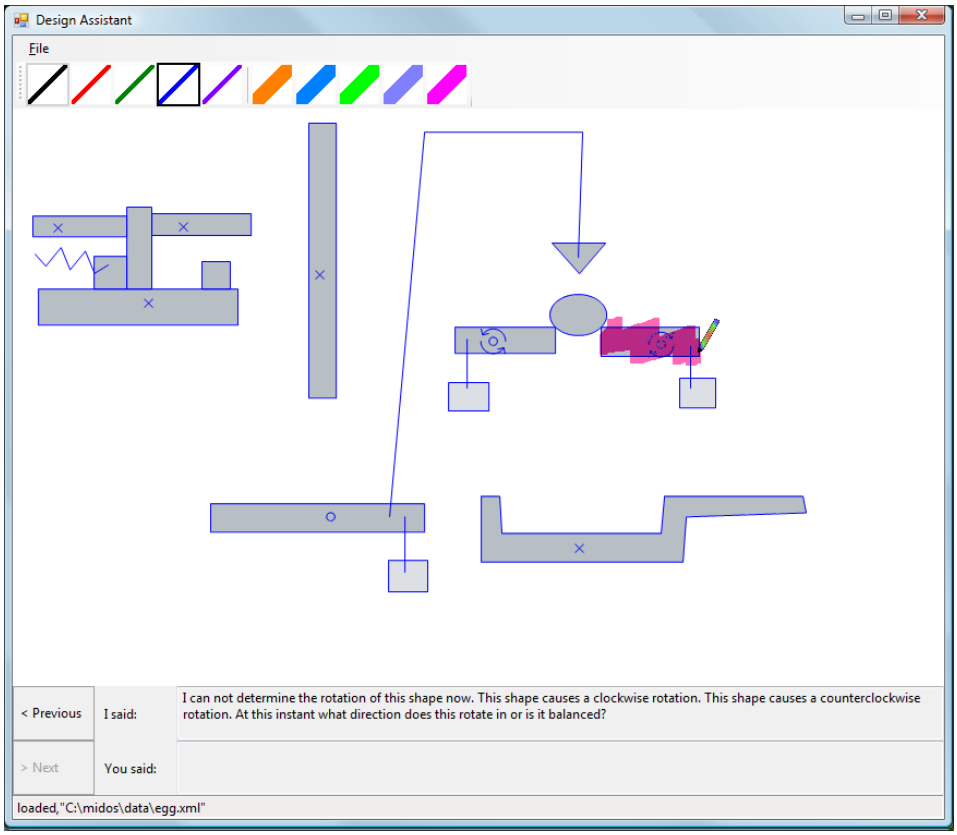

Design Assistant
File
< Previous
I said:
It looks like these shapes will collide, do they?
> Next
You said:
loaded,"C:\midos\data\egg.xml"

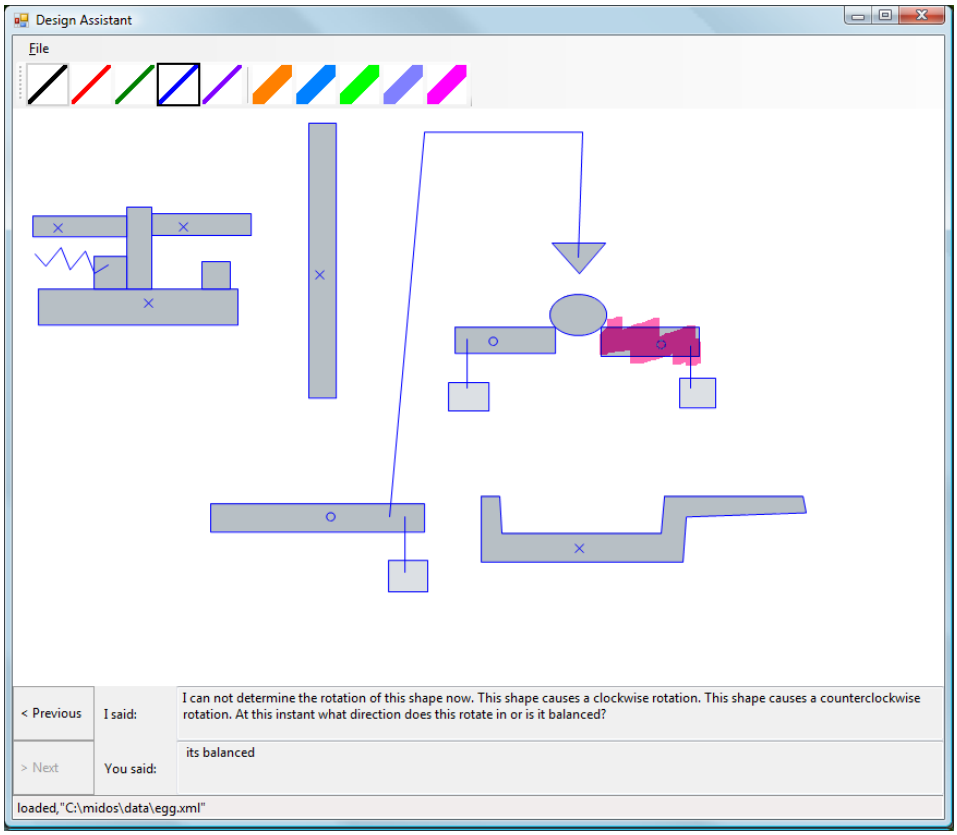

Design Assistant
File
< Previous
I said:
It looks like these shapes will collide, do they?
> Next
You said:
yes
loaded,"C:\midos\data\egg.xml"

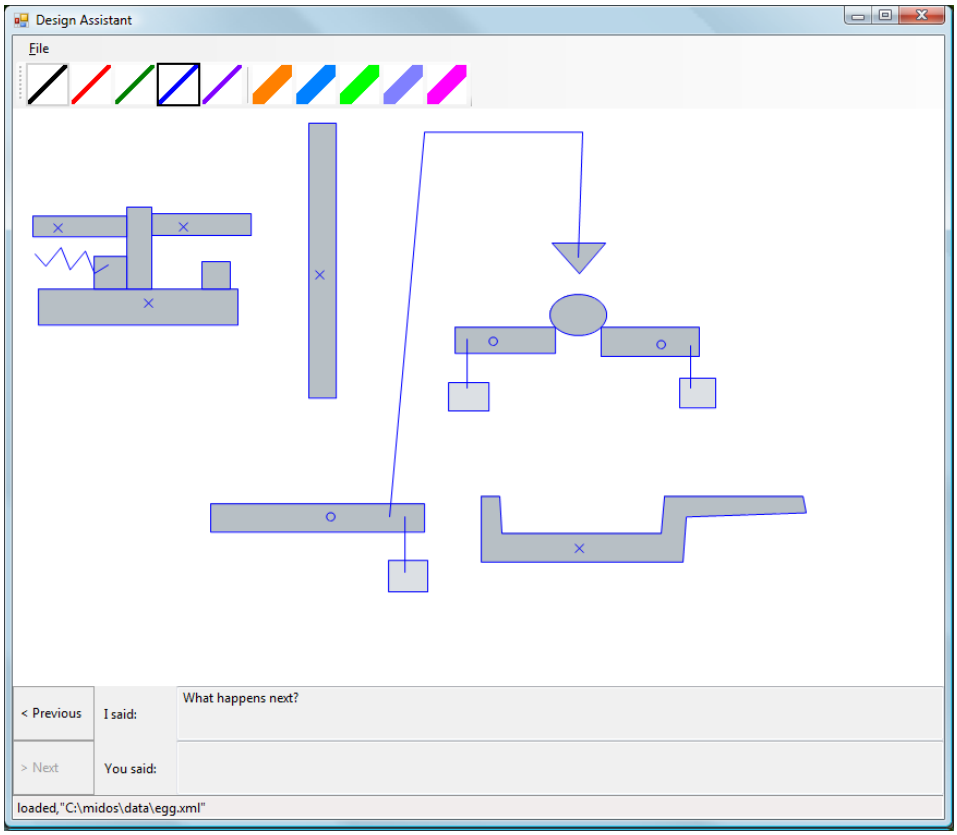

Design Assistant
File
< Previous
I said:
I can not determine the rotation of this shape now. This shape causes a counterclockwise rotation. This shape causes a clockwise rotation. At this instant what direction does this rotate in or is it balanced?
> Next
You said:
loaded,"C:\midos\data\egg.xml"

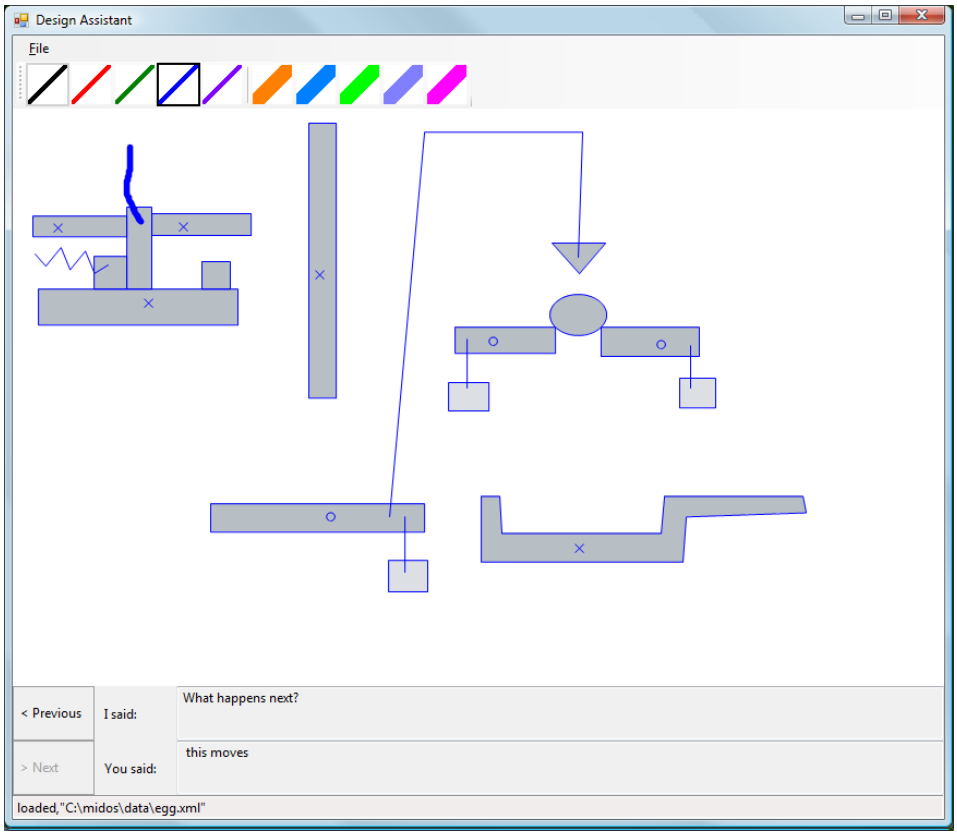

Design Assistant
File
< Previous
I said:
I can not determine the rotation of this shape now. This shape causes a counterclockwise rotation. This shape causes a clockwise rotation. At this instant what direction does this rotate in or is it balanced?
> Next
You said:
loaded,"C:\midos\data\egg.xml"

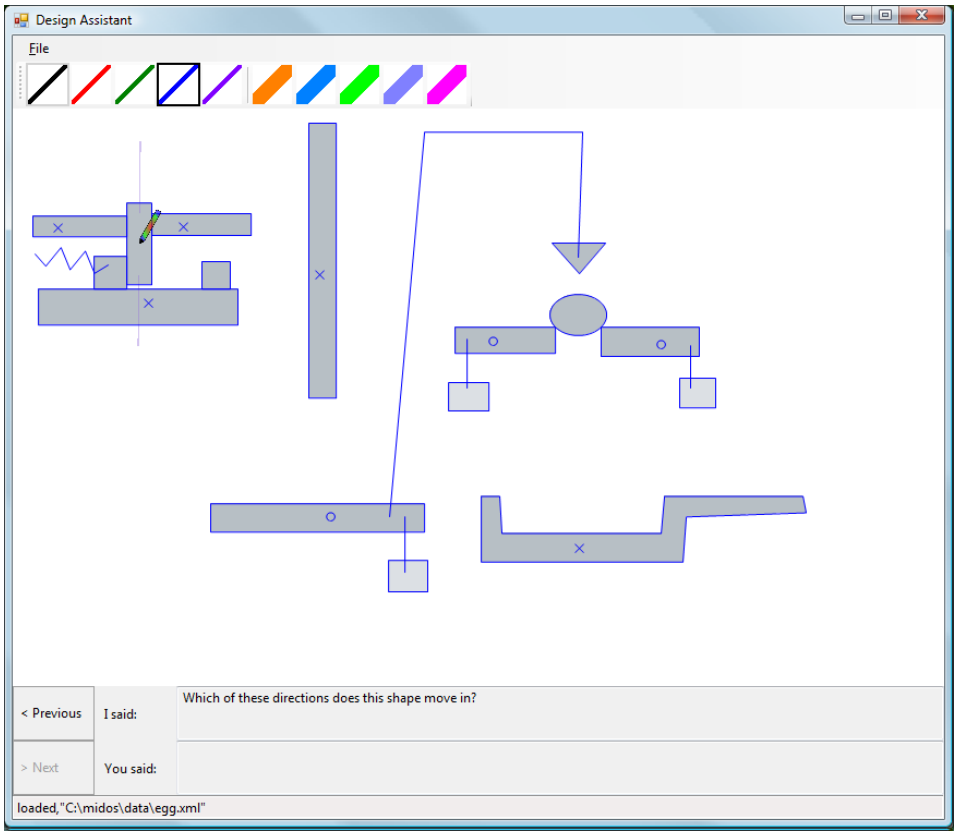

Design Assistant
File
< Previous
I said:
I can not determine the rotation of this shape now. This shape causes a clockwise rotation. This shape causes a counterclockwise rotation. At this instant what direction does this rotate in or is it balanced?
> Next
You said:
loaded,"C:\midos\data\egg.xml"

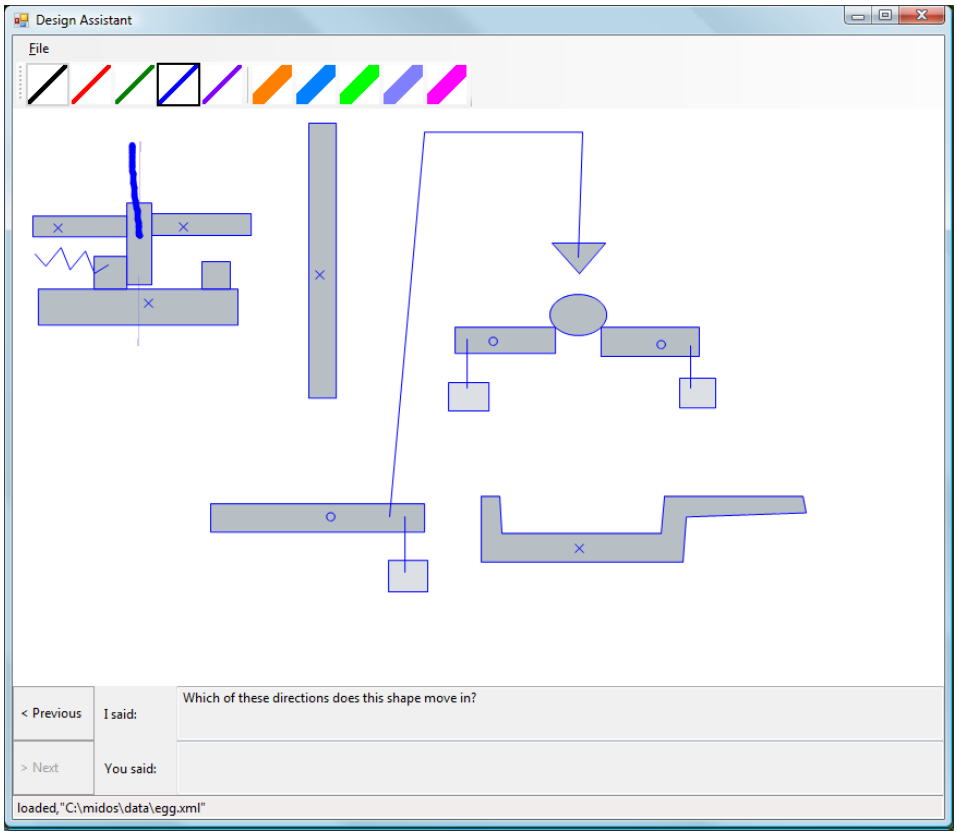

Design Assistant
File
< Previous
I said:
I can not determine the rotation of this shape now. This shape causes a clockwise rotation. This shape causes a counterclockwise rotation. At this instant what direction does this rotate in or is it balanced?
> Next
You said:
loaded,"C:\midos\data\egg.xml"

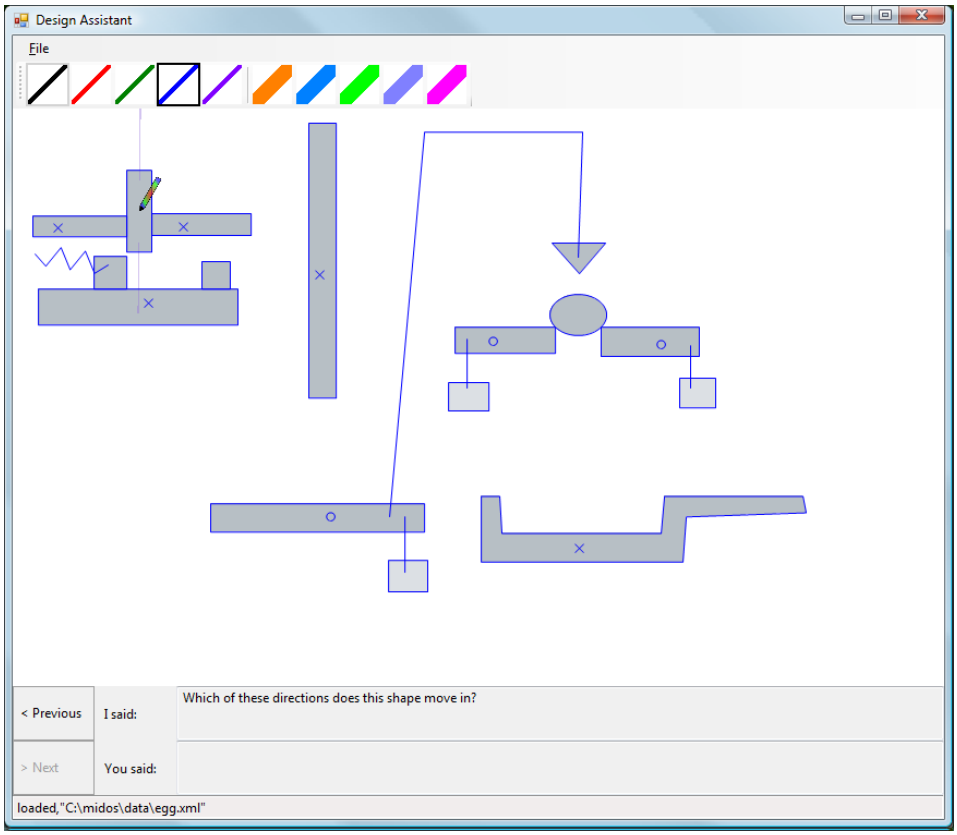

Design Assistant
File
< Previous
I said:
I can not determine the rotation of this shape now. This shape causes a clockwise rotation. This shape causes a counterclockwise rotation. At this instant what direction does this rotate in or is it balanced?
> Next
You said:
its balanced
loaded,"C:\midos\data\egg.xml"

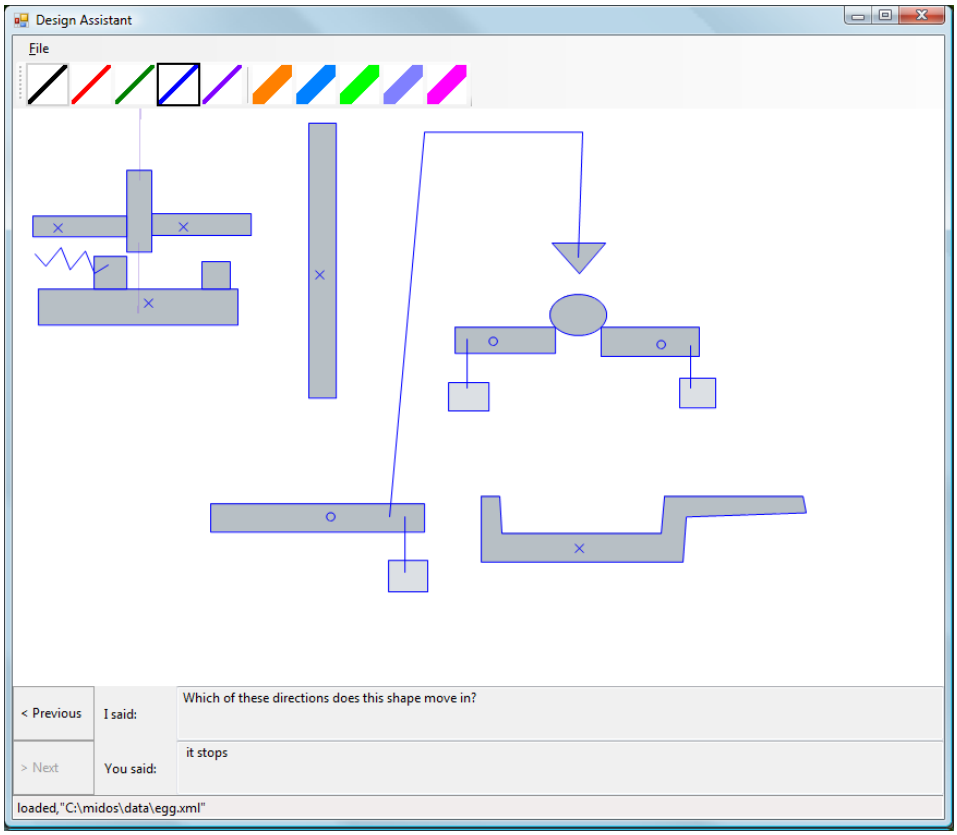

Design Assistant
File
< Previous
I said:
What happens next?
> Next
You said:
loaded,"C:\midos\data\egg.xml"

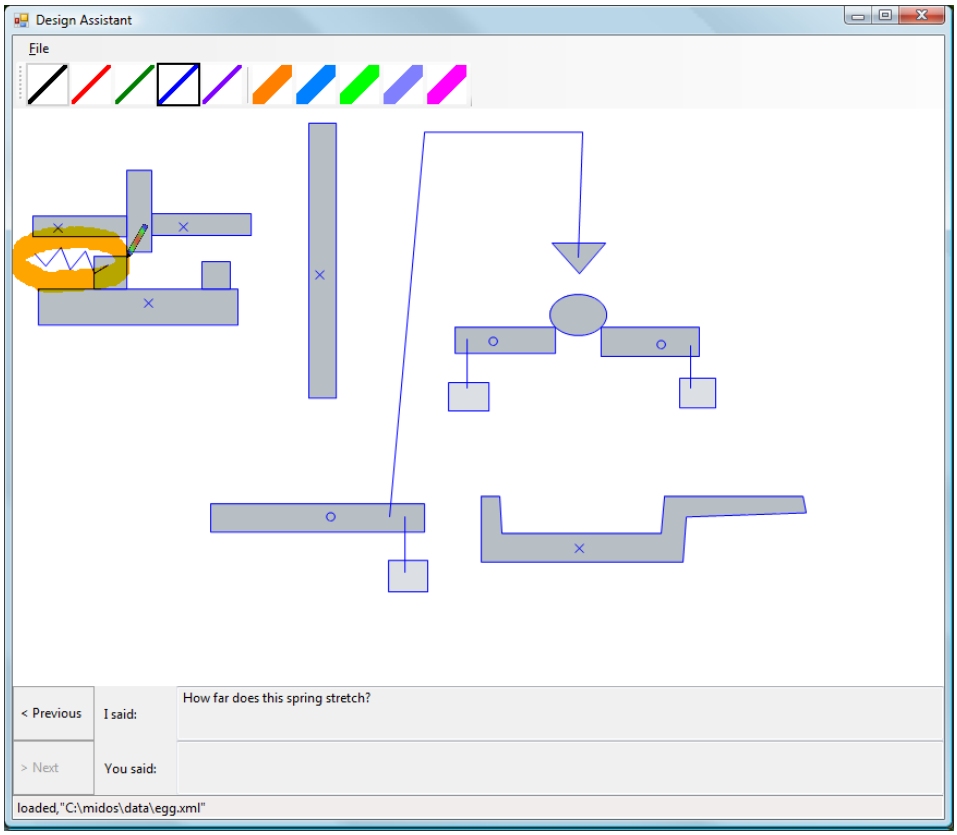

Design Assistant
File
< Previous
I said:
What happens next?
> Next
You said:
this moves
loaded,"C:\midos\data\egg.xml"

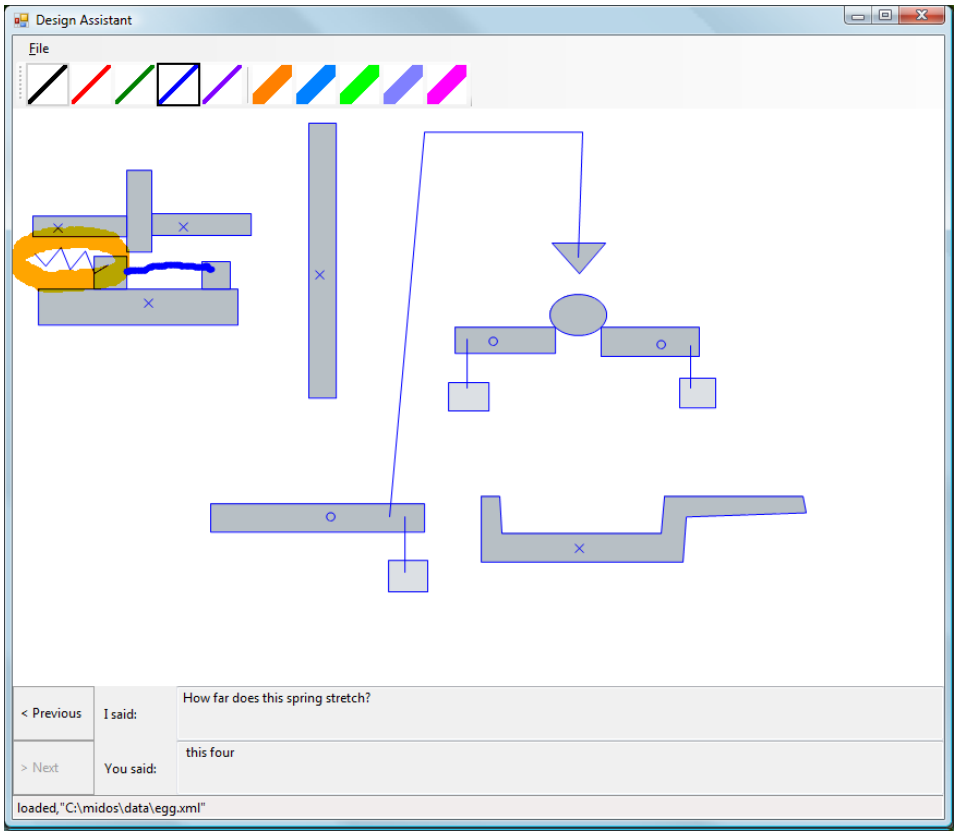

Design Assistant
File
< Previous
I said:
Which of these directions does this shape move in?
> Next
You said:
loaded,"C:\midos\data\egg.xml"

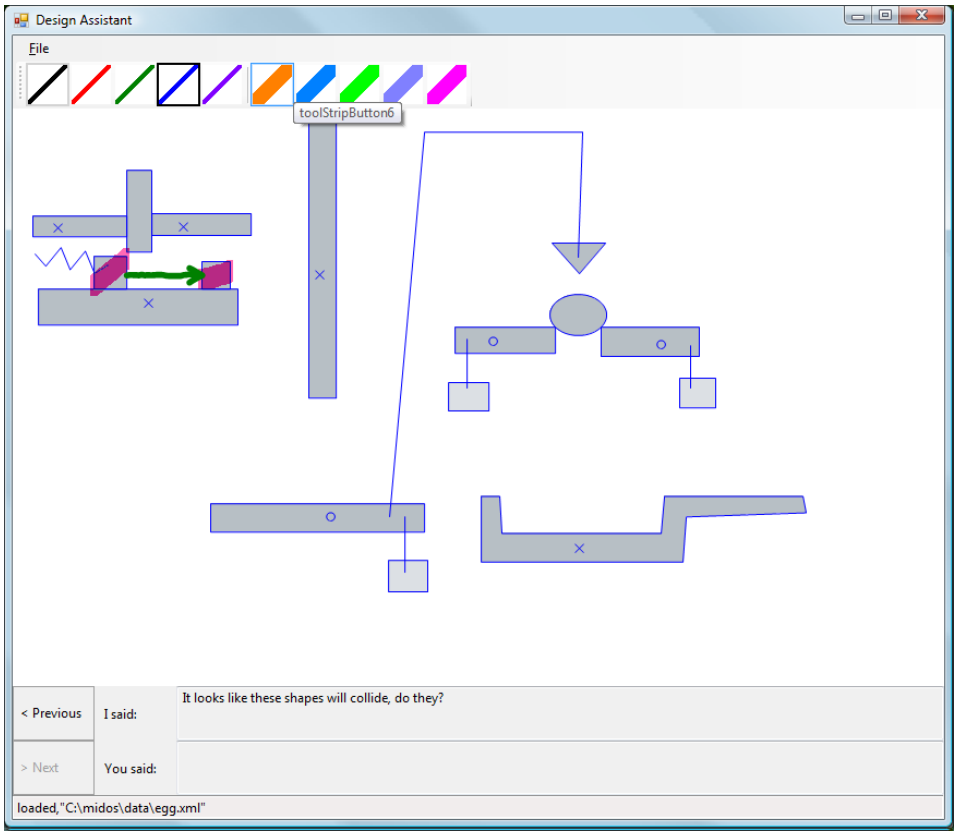

Design Assistant
File
< Previous
I said:
Which of these directions does this shape move in?
> Next
You said:
loaded,"C:\midos\data\egg.xml"

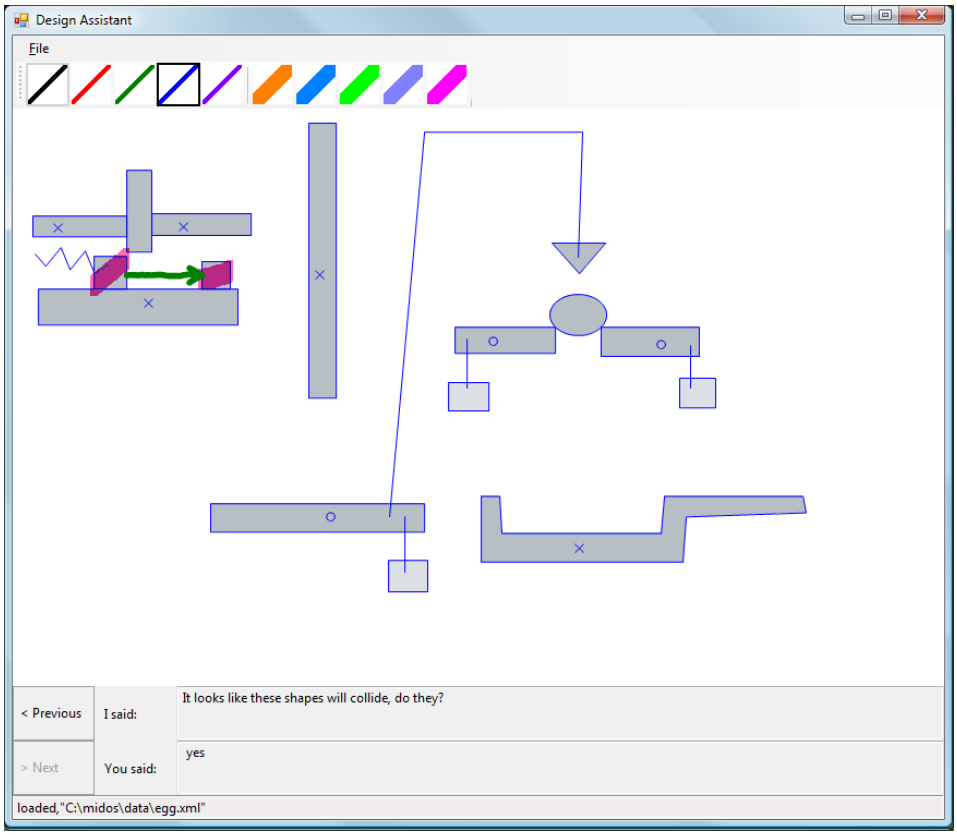

Design Assistant
File
< Previous
I said:
Which of these directions does this shape move in?
> Next
You said:
loaded,"C:\midos\data\egg.xml"

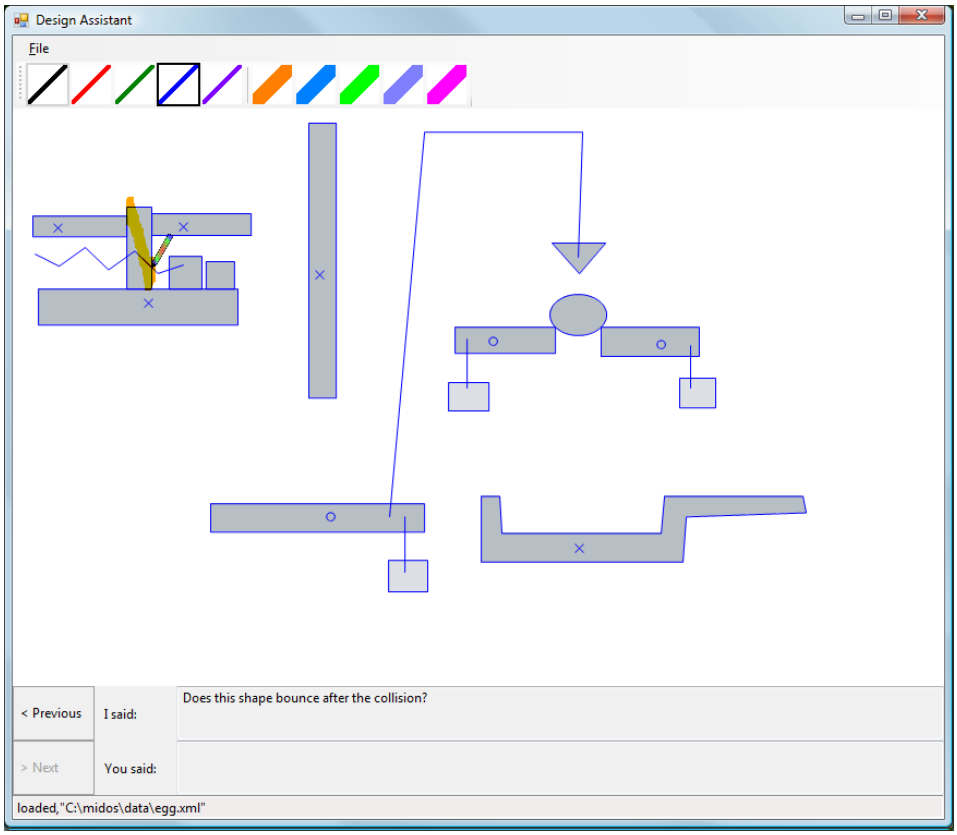

Design Assistant
File
< Previous
I said:
Which of these directions does this shape move in?
> Next
You said:
it stops
loaded,"C:\midos\data\egg.xml"

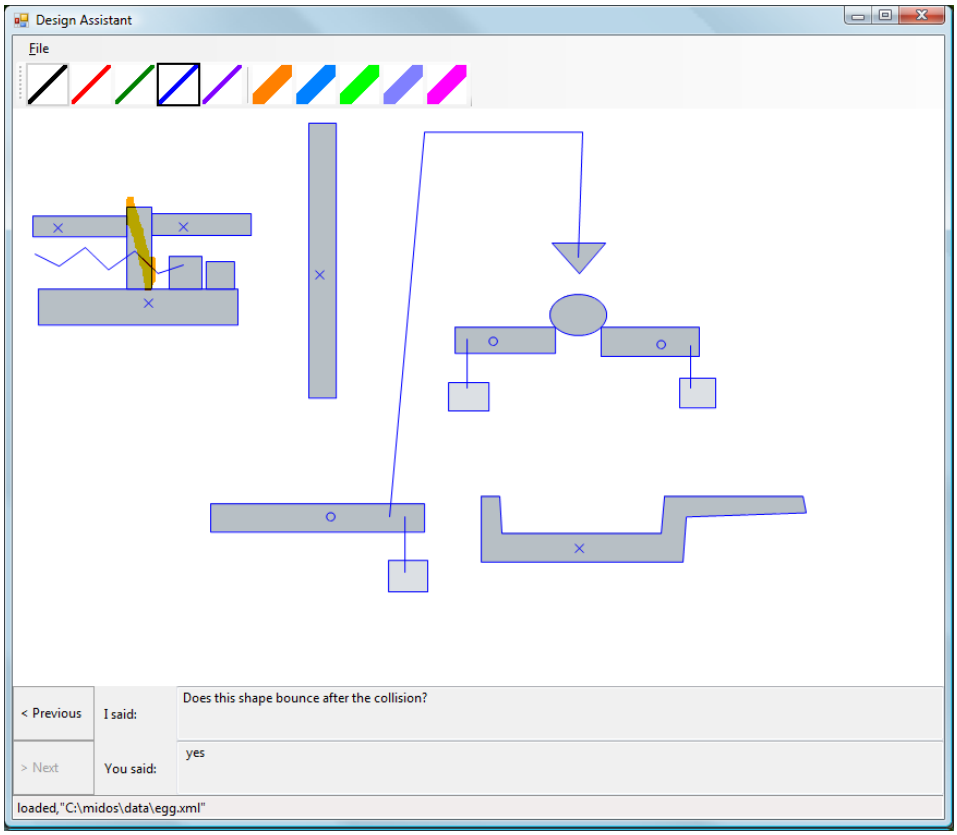

Design Assistant
File
< Previous
I said:
How far does this spring stretch?
> Next
You said:
loaded,"C:\midos\data\egg.xml"

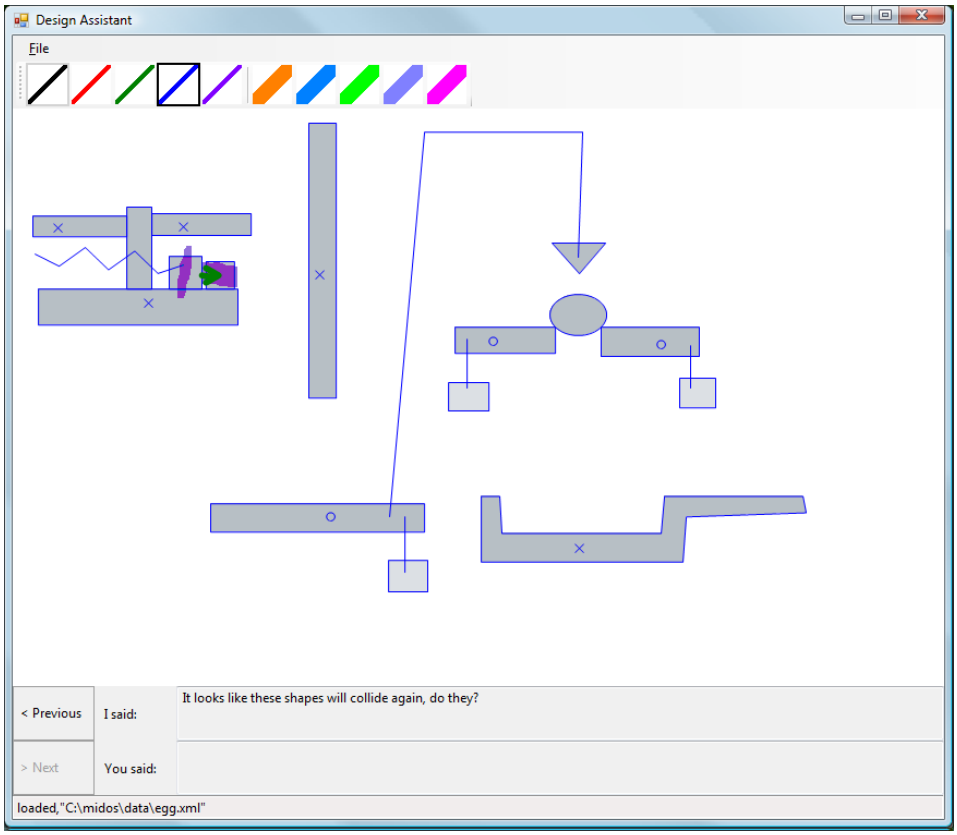

Design Assistant
File
< Previous
I said:
How far does this spring stretch?
> Next
You said:
this four
loaded,"C:\midos\data\egg.xml"

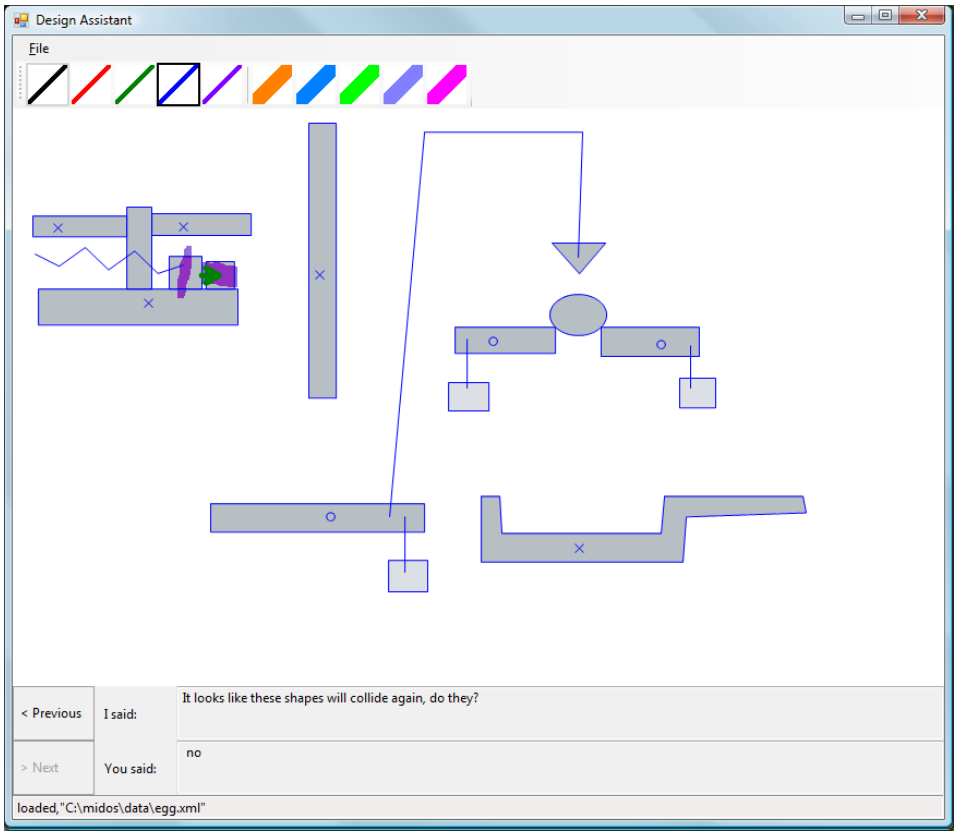

Design Assistant
File
toolStripButton6
< Previous
I said:
It looks like these shapes will collide, do they?
> Next
You said:
loaded,"C:\midos\data\egg.xml"

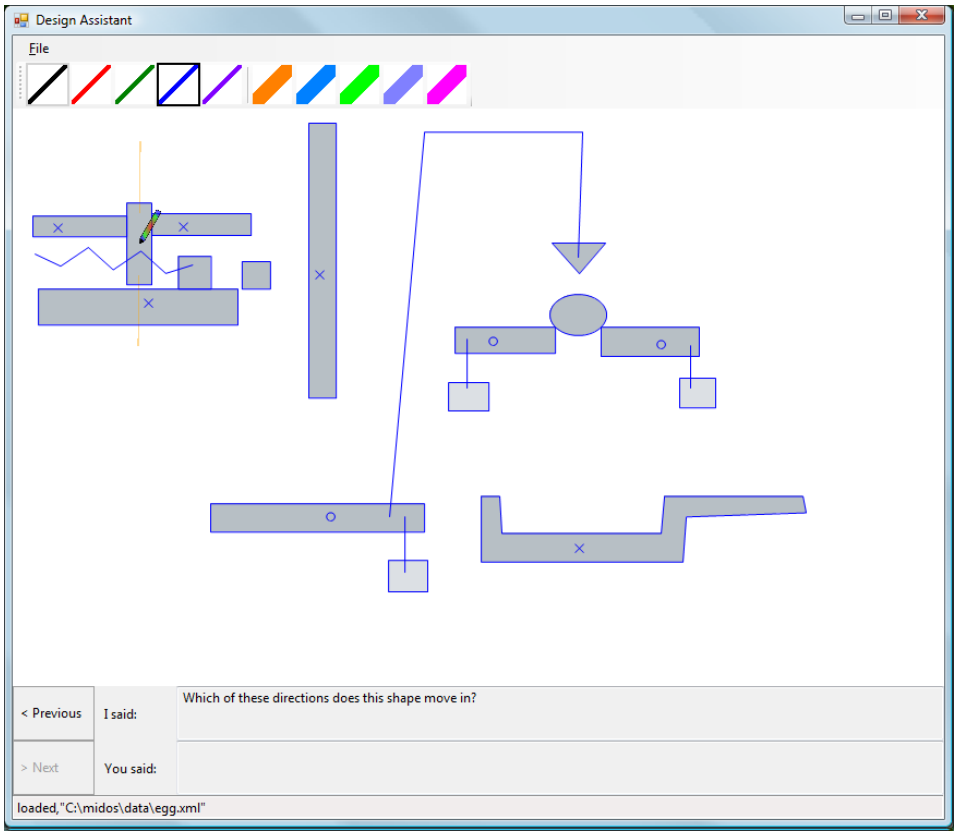

Design Assistant
File
< Previous
I said:
It looks like these shapes will collide, do they?
> Next
You said:
yes
loaded,"C:\midos\data\egg.xml"

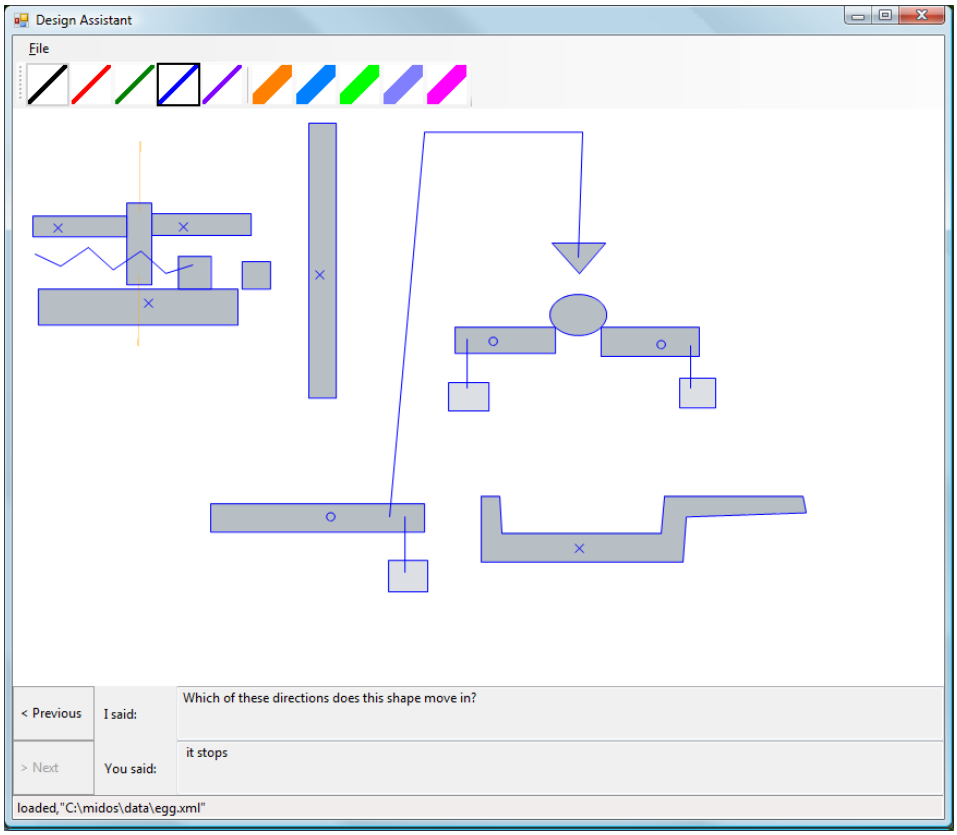

Design Assistant
File
< Previous
I said:
Does this shape bounce after the collision?
> Next
You said:
loaded,"C:\midos\data\egg.xml"

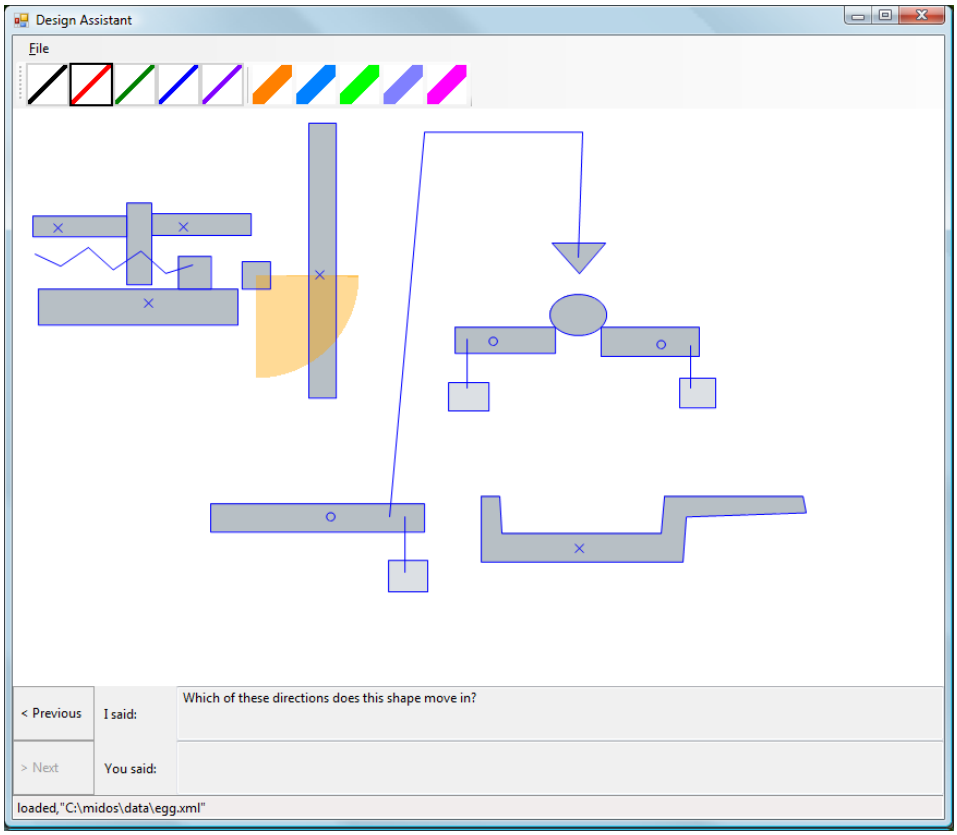

Design Assistant
File
< Previous
I said:
Does this shape bounce after the collision?
> Next
You said:
yes
loaded,"C:\midos\data\egg.xml"

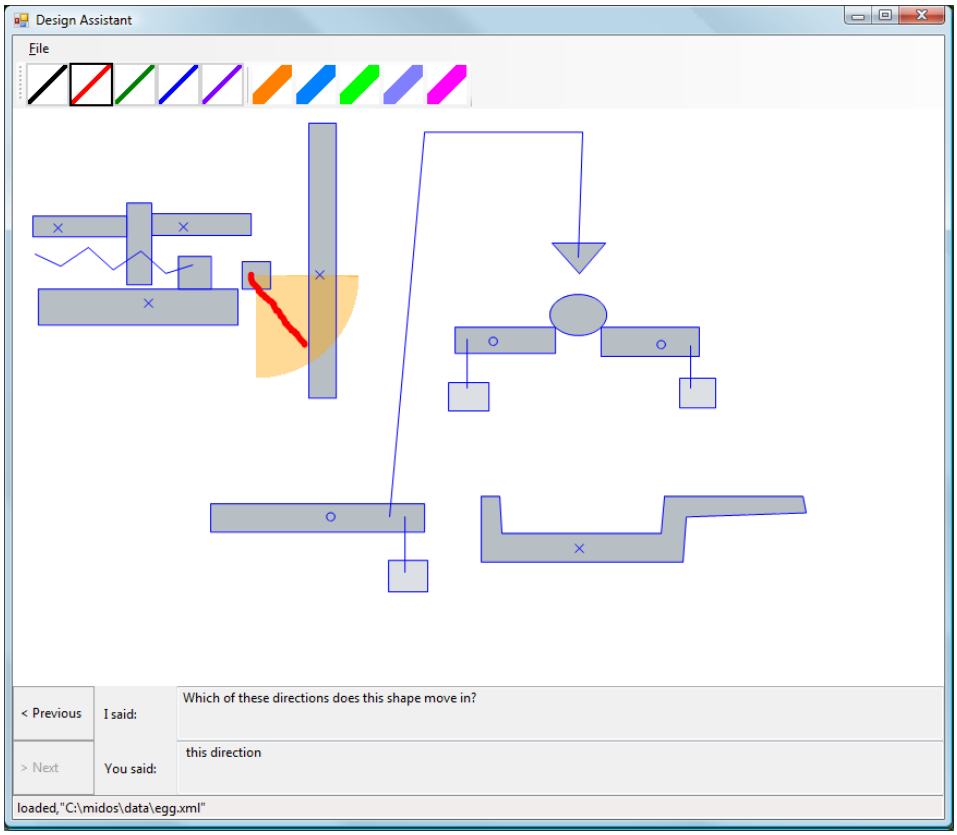

Design Assistant
File
< Previous
I said:
It looks like these shapes will collide again, do they?
> Next
You said:
loaded,"C:\midos\data\egg.xml"

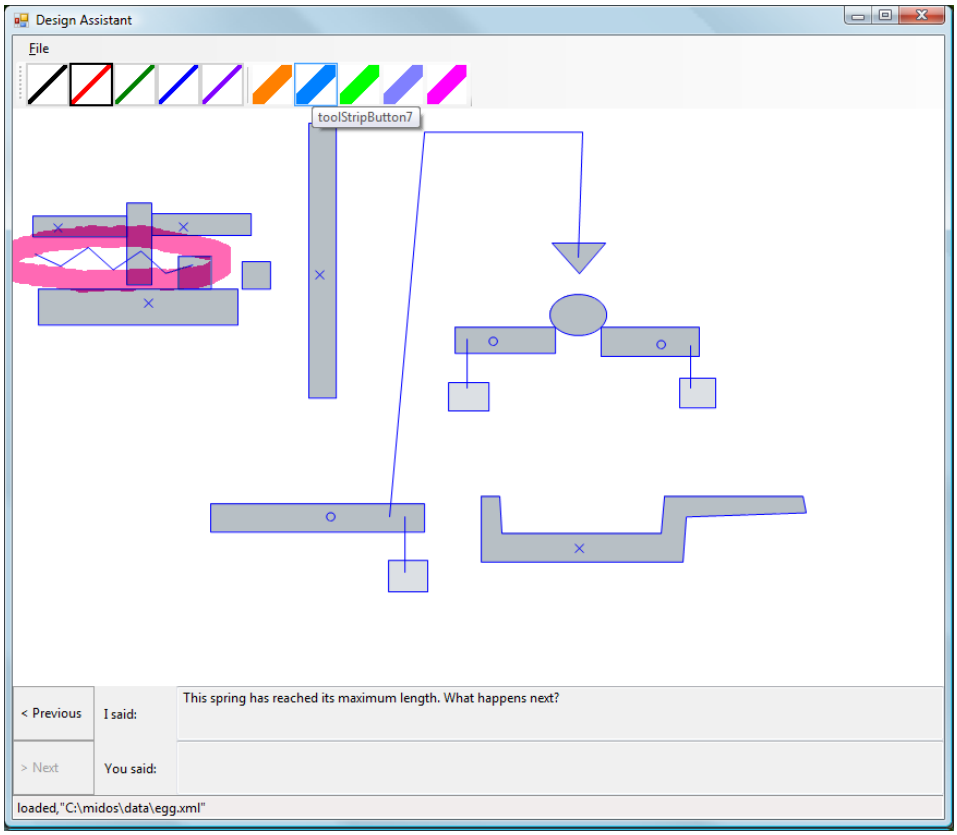

Design Assistant
File
< Previous
I said:
It looks like these shapes will collide again, do they?
> Next
You said:
no
loaded,"C:\midos\data\egg.xml"

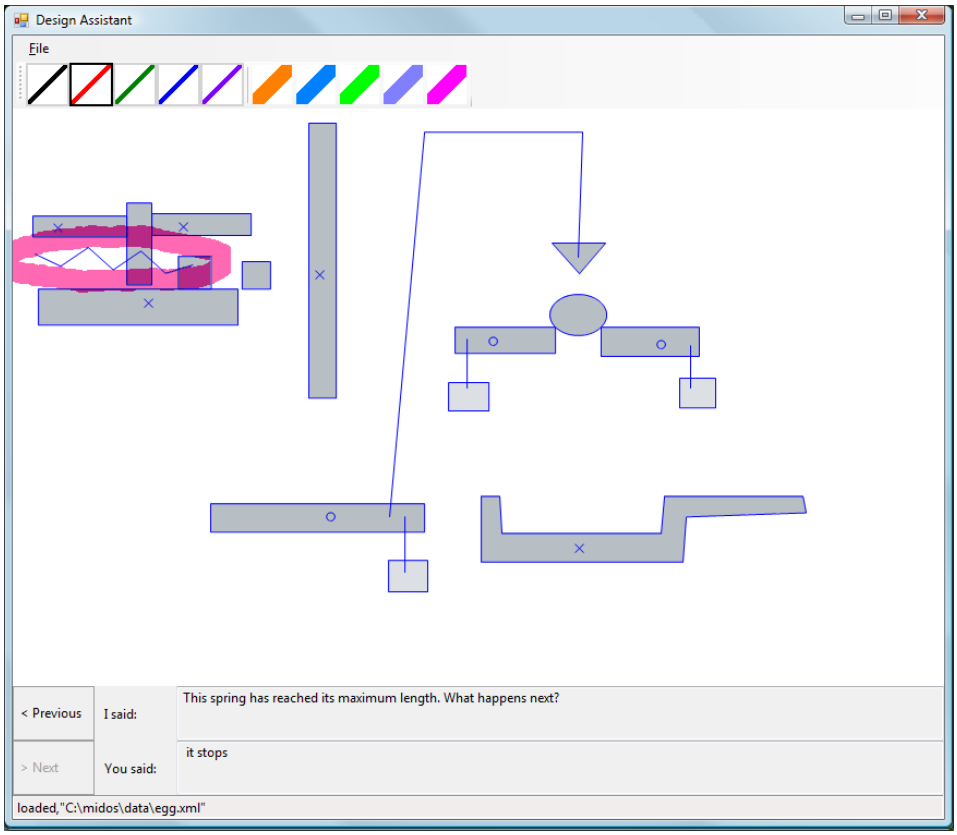

Design Assistant
File
< Previous
I said:
Which of these directions does this shape move in?
> Next
You said:
loaded,"C:\midos\data\egg.xml"

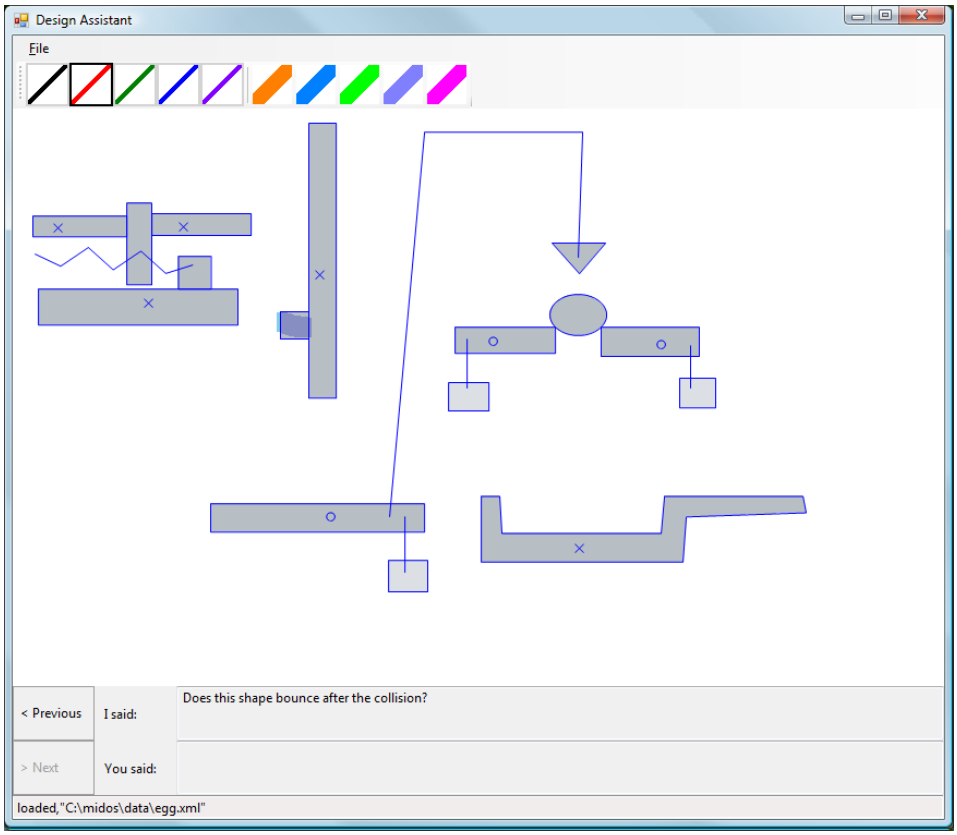

Design Assistant
File
< Previous
I said:
Which of these directions does this shape move in?
> Next
You said:
it stops
loaded,"C:\midos\data\egg.xml"

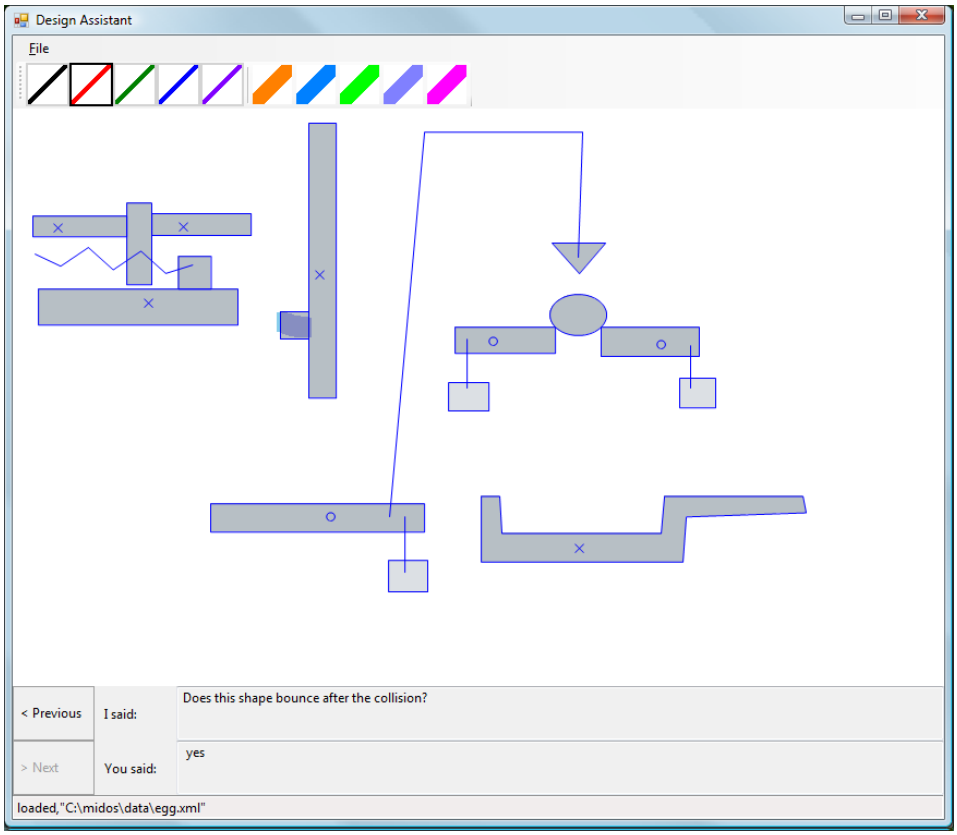

Design Assistant
File
< Previous
I said:
Which of these directions does this shape move in?
> Next
You said:
loaded,"C:\midos\data\egg.xml"

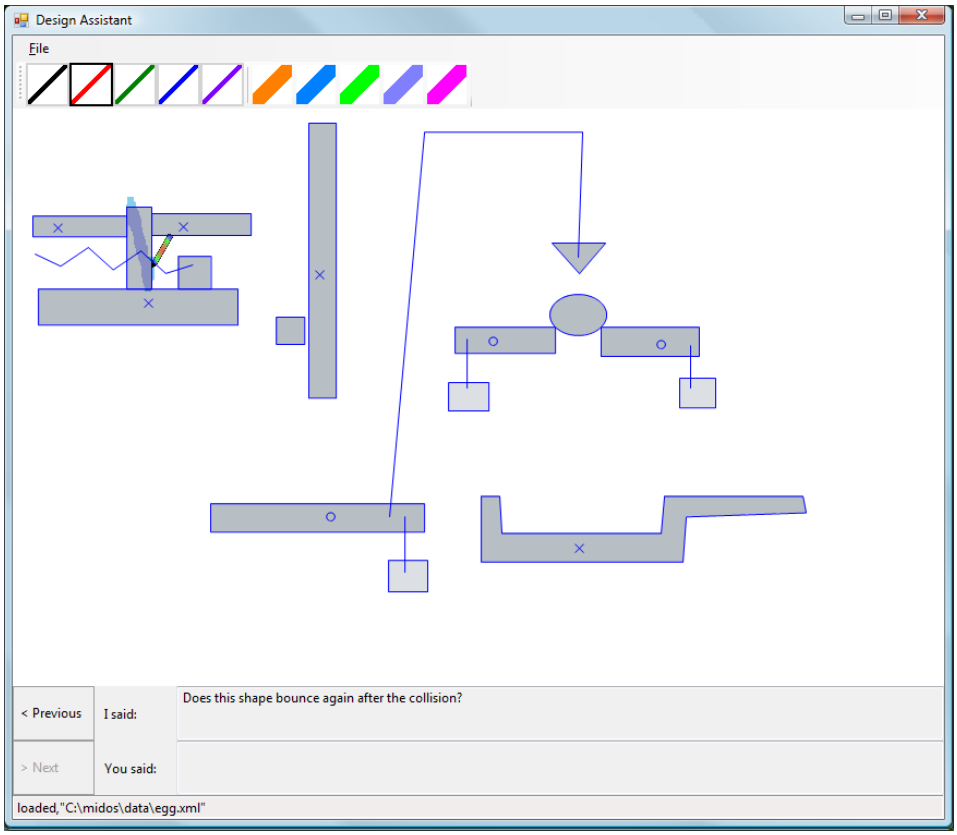

Design Assistant
File
< Previous
I said:
Which of these directions does this shape move in?
> Next
You said:
this direction
loaded,"C:\midos\data\egg.xml"

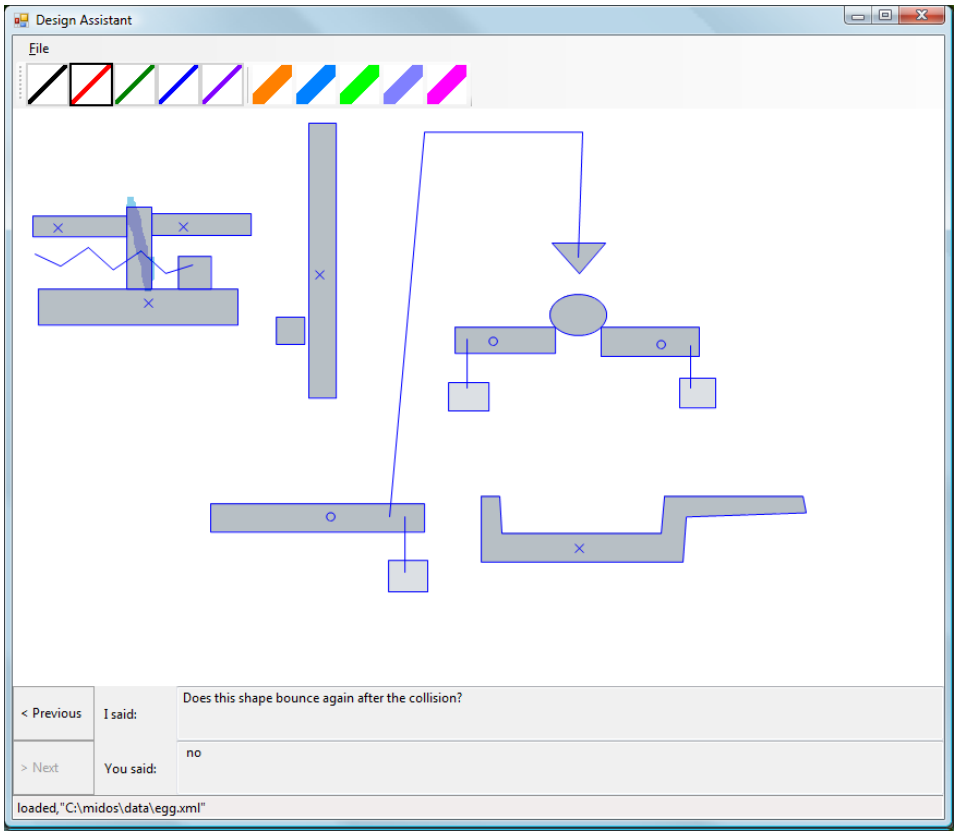

Design Assistant
File
toolStripButton7
< Previous
I said:
This spring has reached its maximum length. What happens next?
> Next
You said:
loaded,"C:\midos\data\egg.xml"

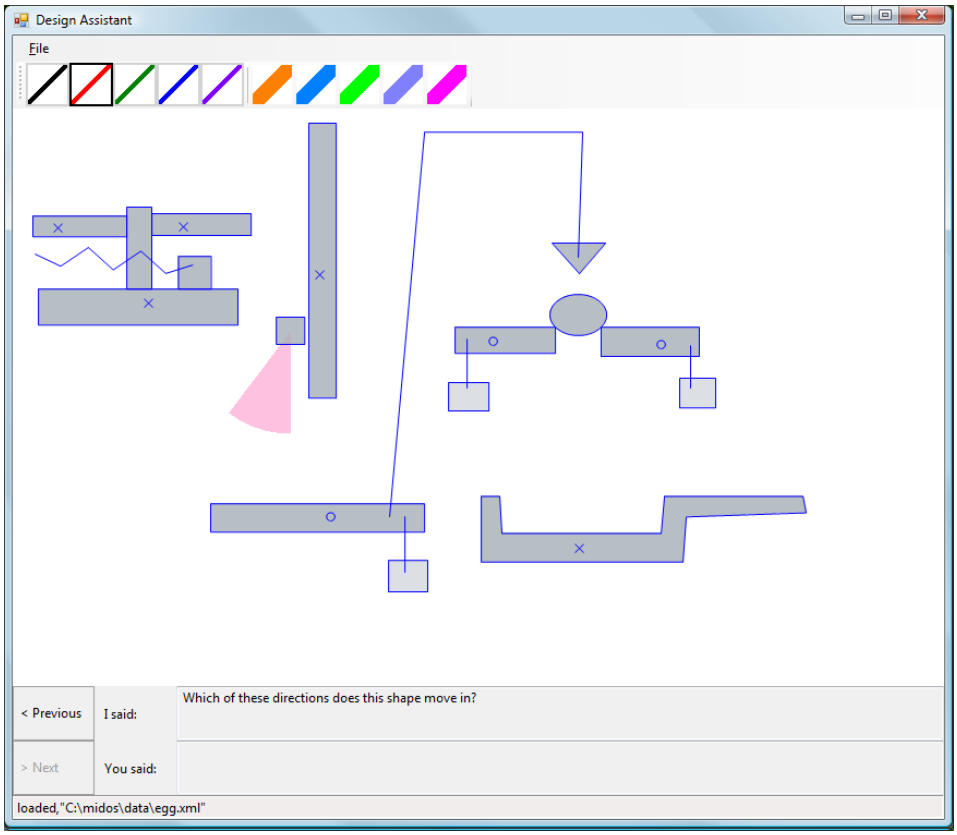

Design Assistant
File
< Previous
I said:
This spring has reached its maximum length. What happens next?
> Next
You said:
it stops
loaded,"C:\midos\data\egg.xml"

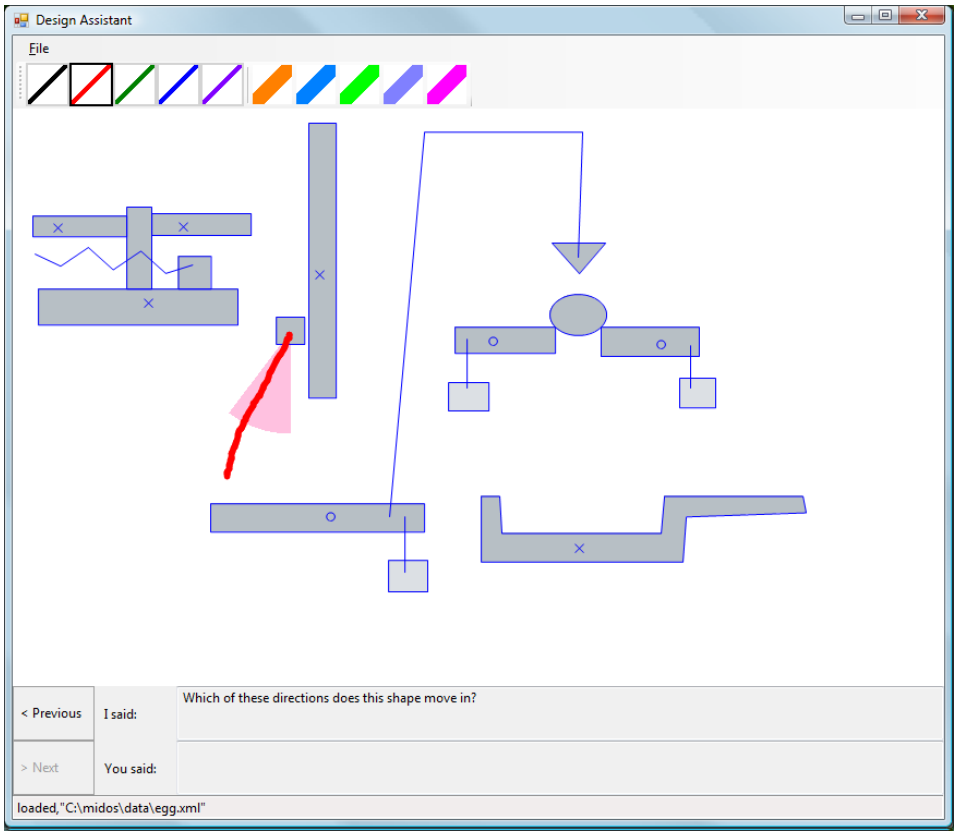

Design Assistant
File
< Previous
I said:
Does this shape bounce after the collision?
> Next
You said:
loaded,"C:\midos\data\egg.xml"

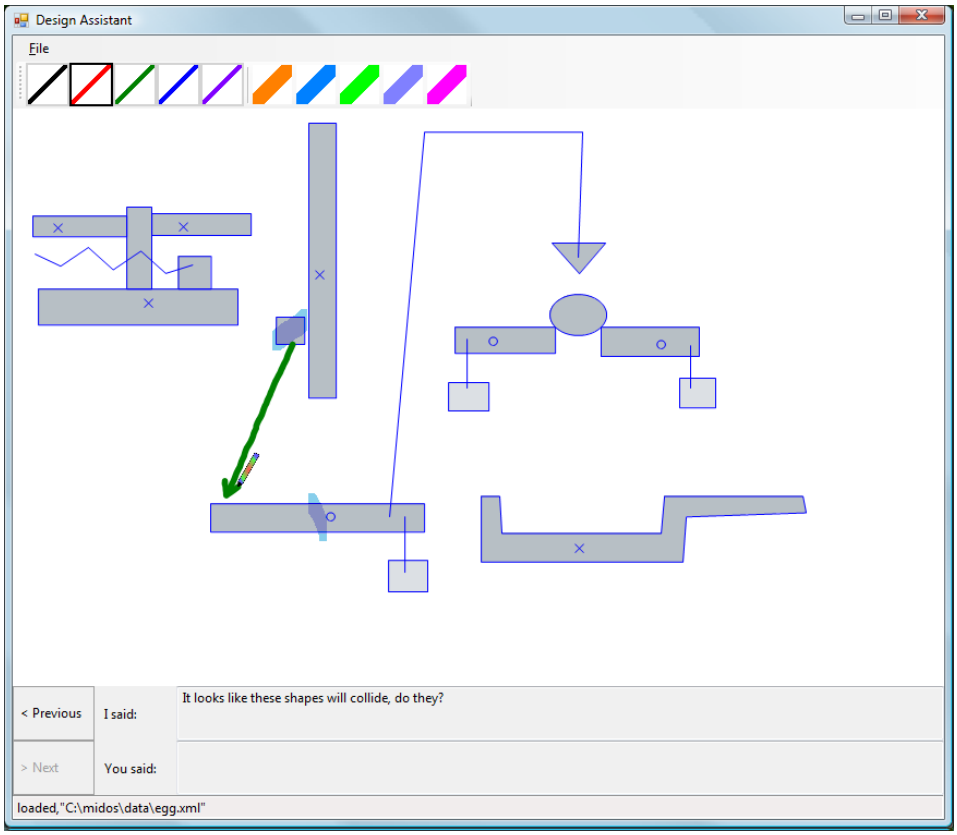

Design Assistant
File
< Previous
I said:
Does this shape bounce after the collision?
> Next
You said:
yes
loaded,"C:\midos\data\egg.xml"

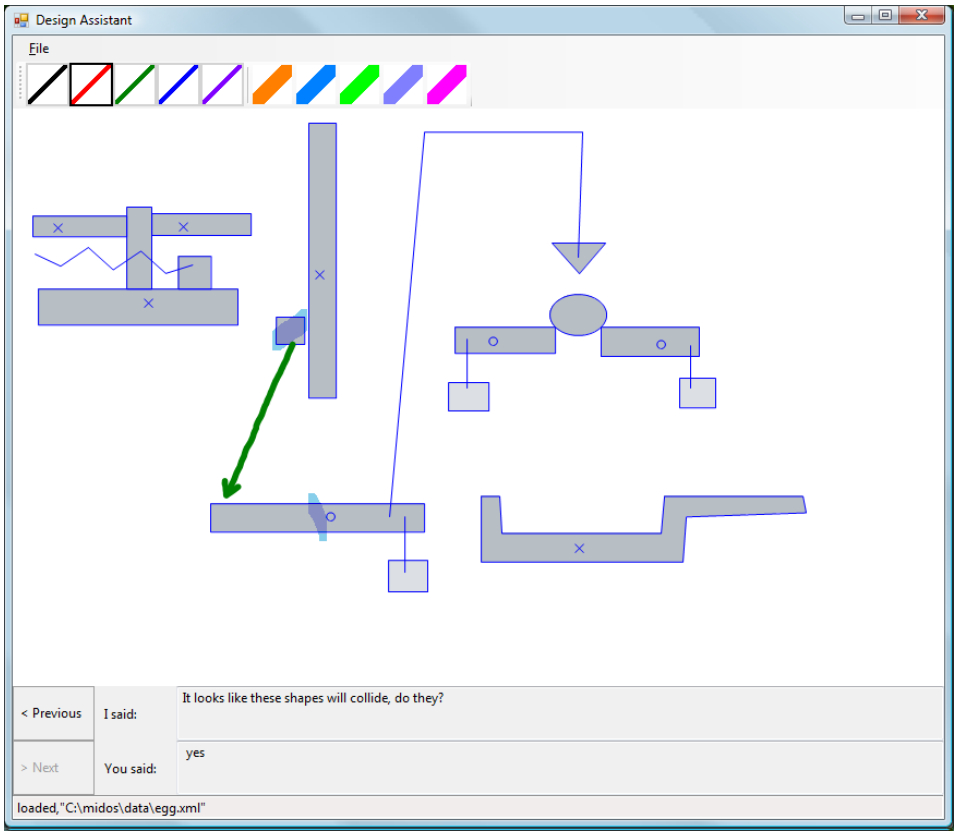

Design Assistant
File
< Previous
I said:
Does this shape bounce again after the collision?
> Next
You said:
loaded,"C:\midos\data\egg.xml"

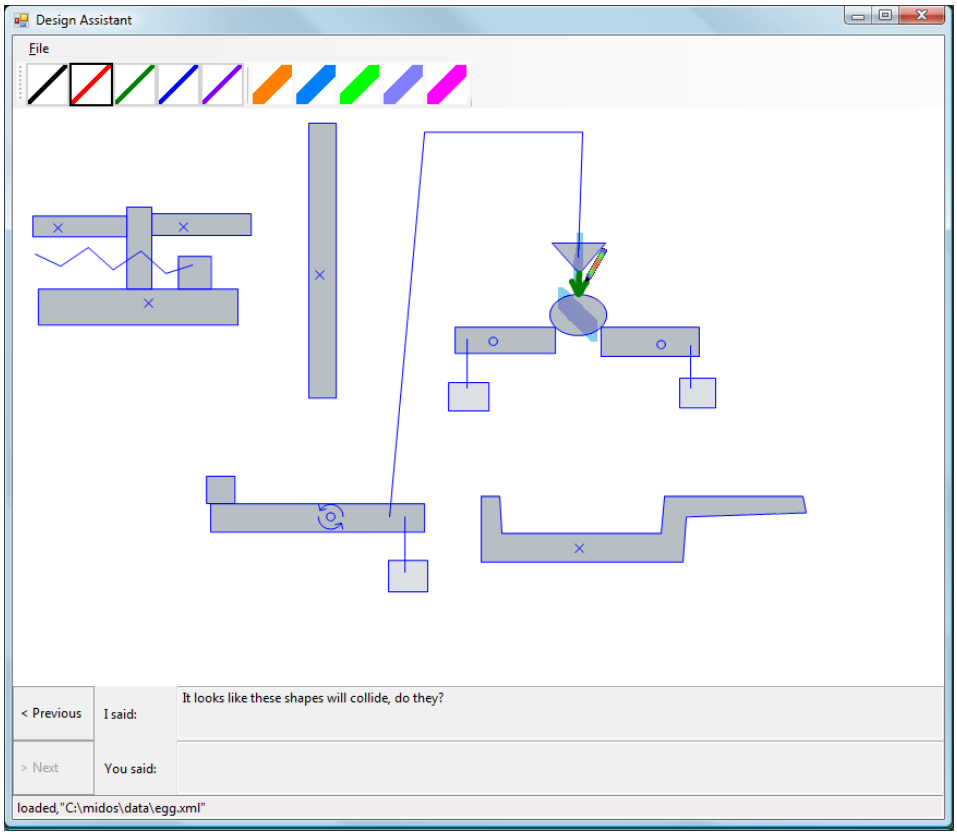

Design Assistant
File
< Previous
I said:
Does this shape bounce again after the collision?
> Next
You said:
no
loaded,"C:\midos\data\egg.xml"

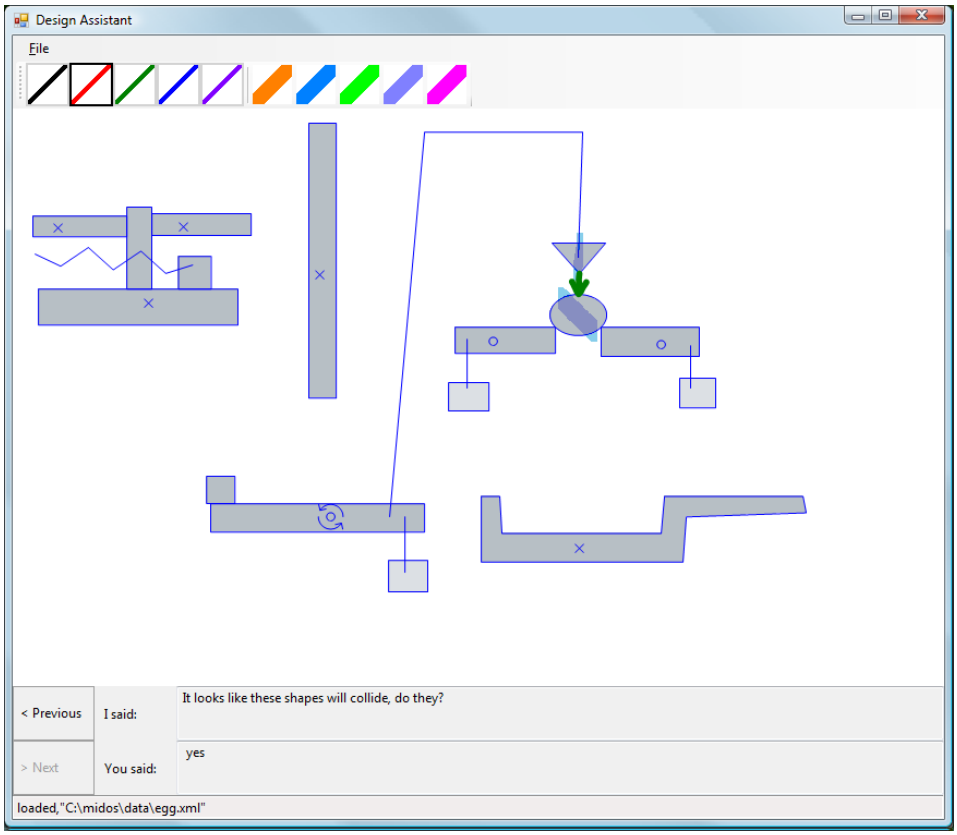

Design Assistant
File
< Previous
I said:
Which of these directions does this shape move in?
> Next
You said:
loaded,"C:\midos\data\egg.xml"

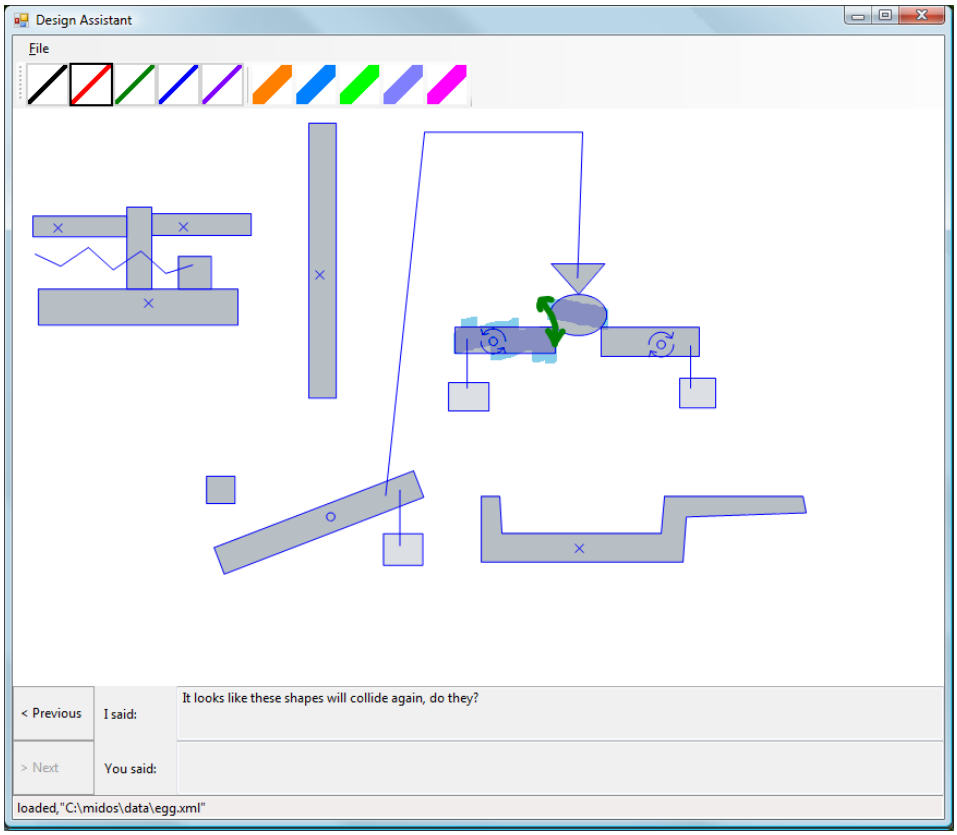

Design Assistant
File
< Previous
I said:
Which of these directions does this shape move in?
> Next
You said:
loaded,"C:\midos\data\egg.xml"

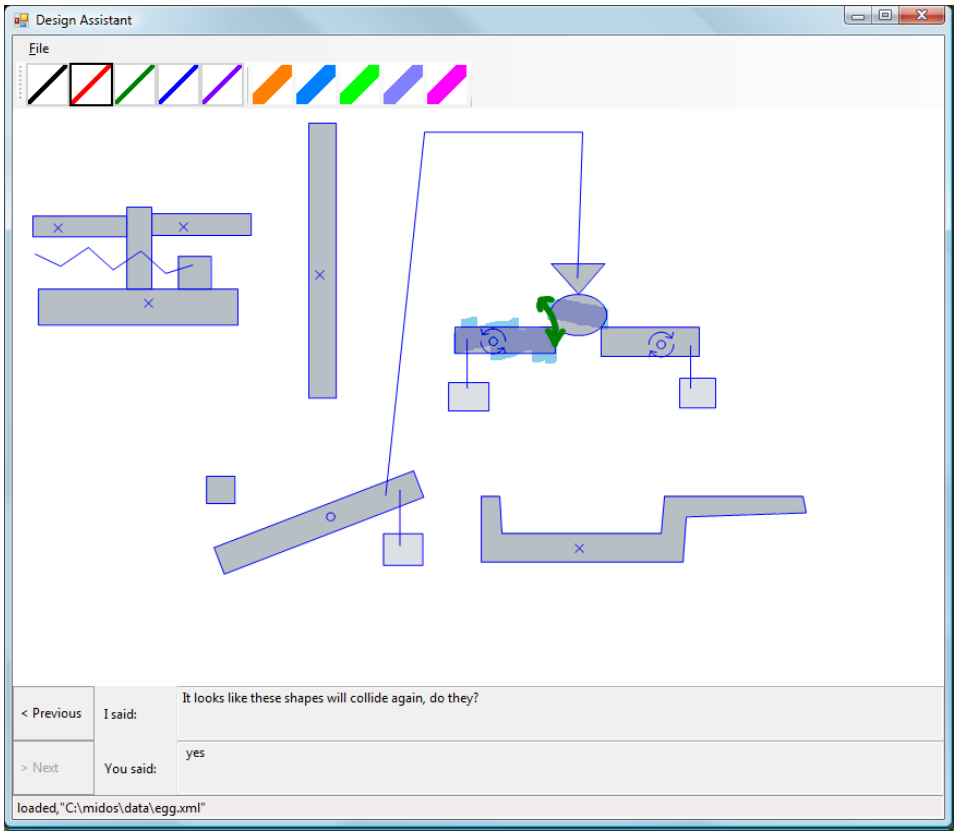

Design Assistant
File
< Previous
I said:
It looks like these shapes will collide, do they?
> Next
You said:
loaded,"C:\midos\data\egg.xml"

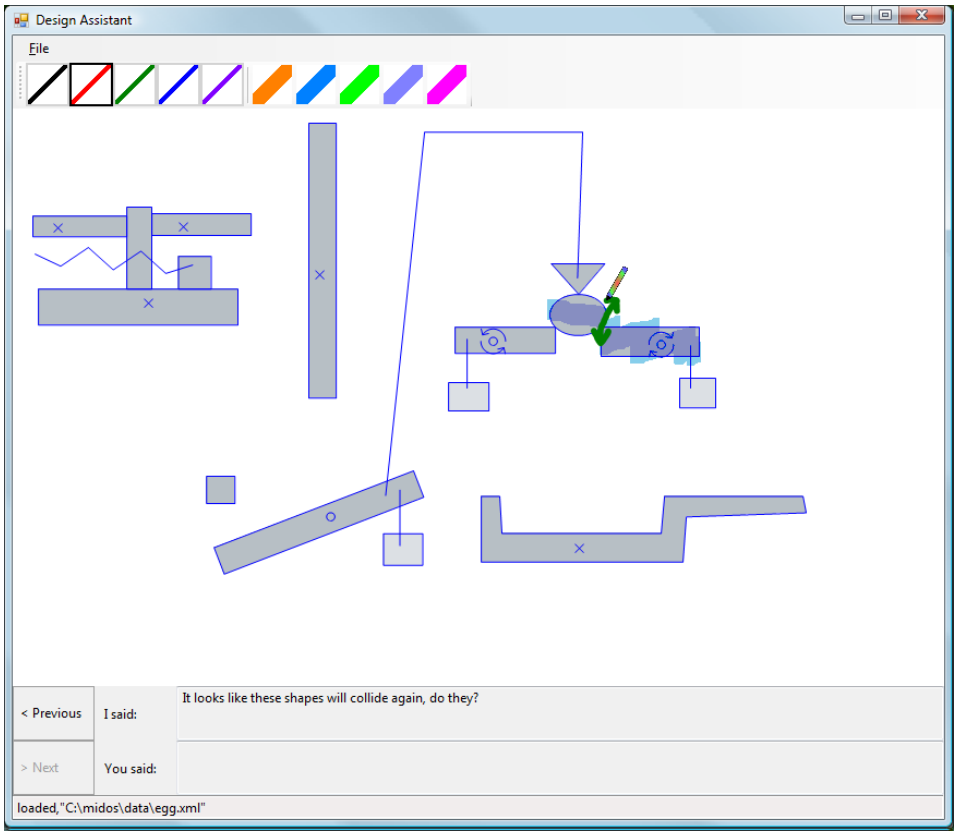

Design Assistant
File
< Previous
I said:
It looks like these shapes will collide, do they?
> Next
You said:
yes
loaded,"C:\midos\data\egg.xml"

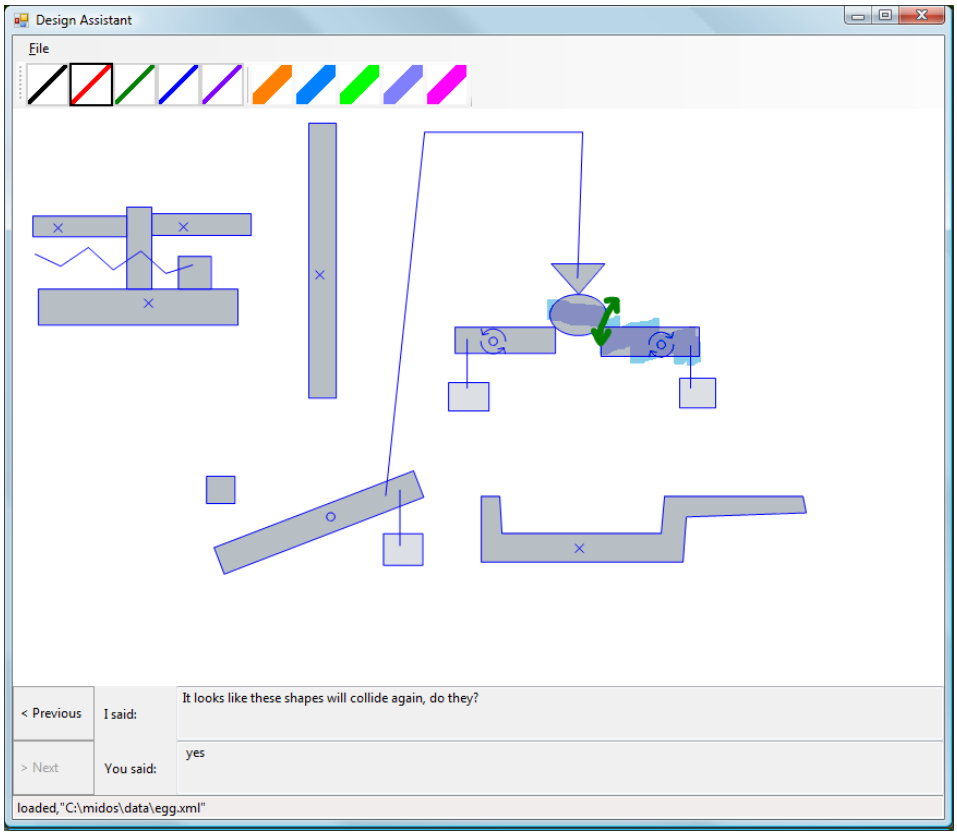

Design Assistant
File
< Previous
I said:
It looks like these shapes will collide, do they?
> Next
You said:
loaded,"C:\midos\data\egg.xml"

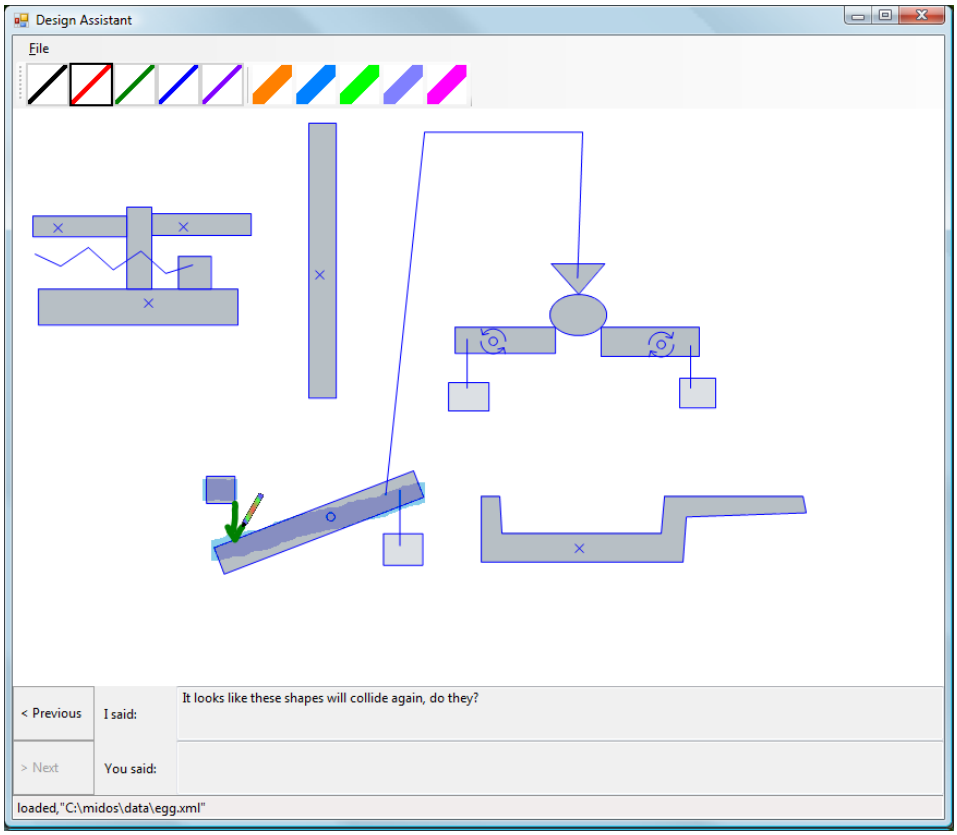

Design Assistant
File
< Previous
I said:
It looks like these shapes will collide, do they?
> Next
You said:
yes
loaded,"C:\midos\data\egg.xml"

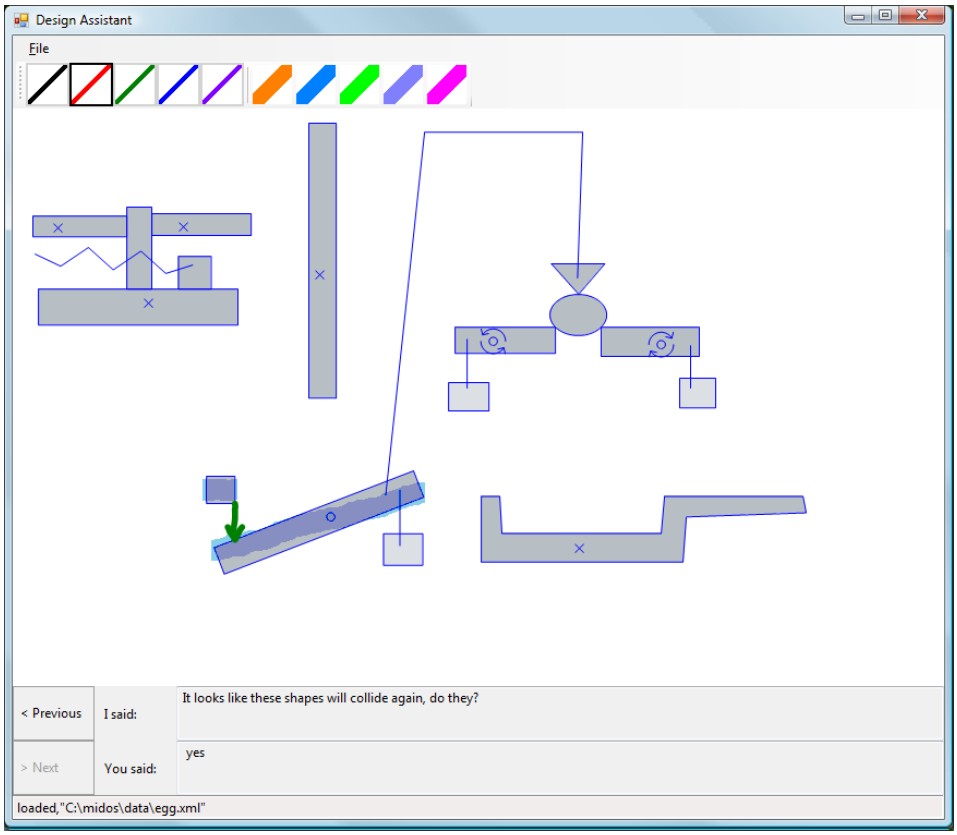

Design Assistant
File
< Previous
I said:
It looks like these shapes will collide again, do they?
> Next
You said:
loaded,"C:\midos\data\egg.xml"

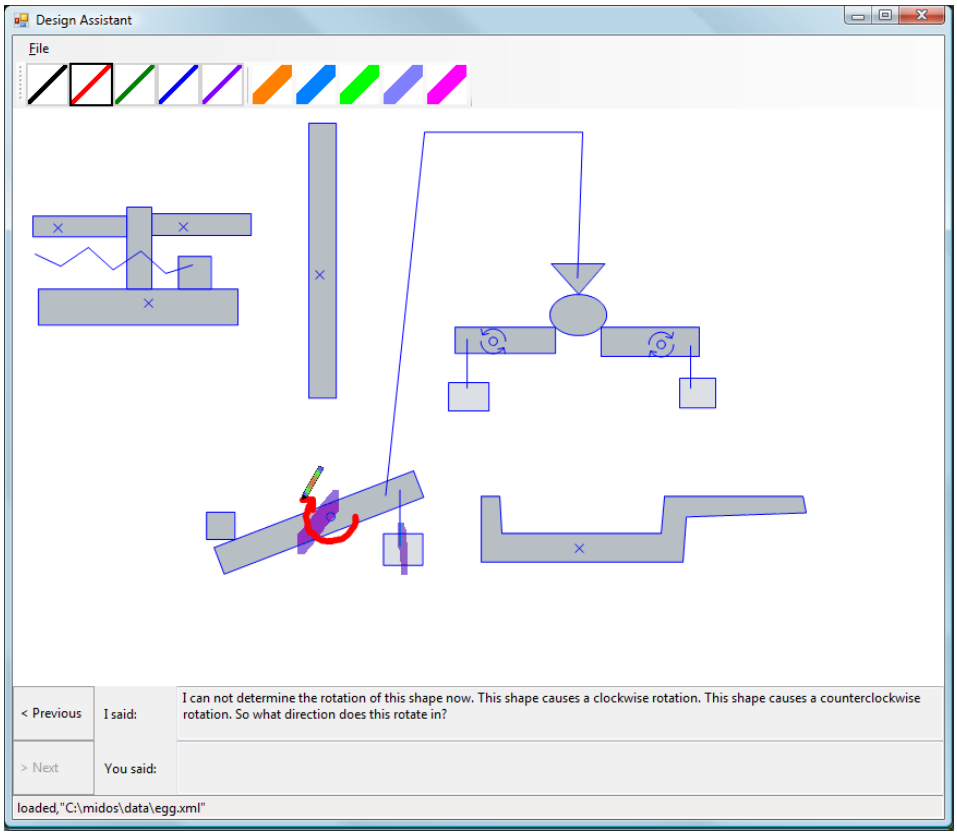

Design Assistant
File
< Previous
I said:
It looks like these shapes will collide again, do they?
> Next
You said:
yes
loaded,"C:\midos\data\egg.xml"

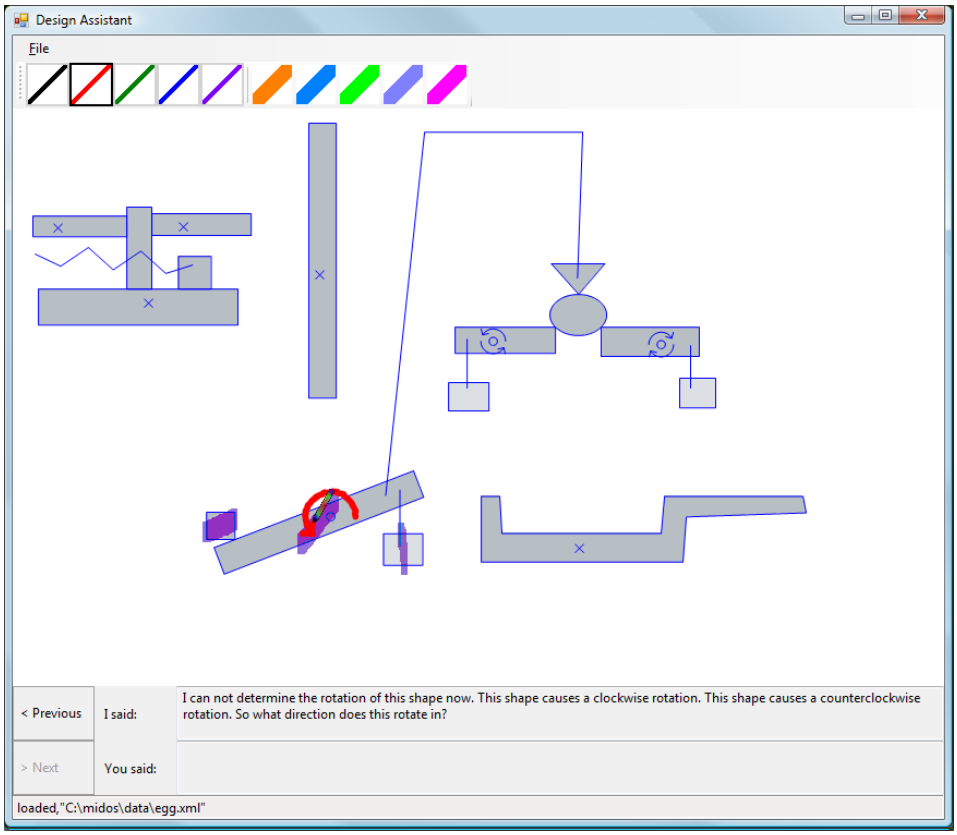

Design Assistant
File
< Previous
I said:
It looks like these shapes will collide again, do they?
> Next
You said:
loaded,"C:\midos\data\egg.xml"

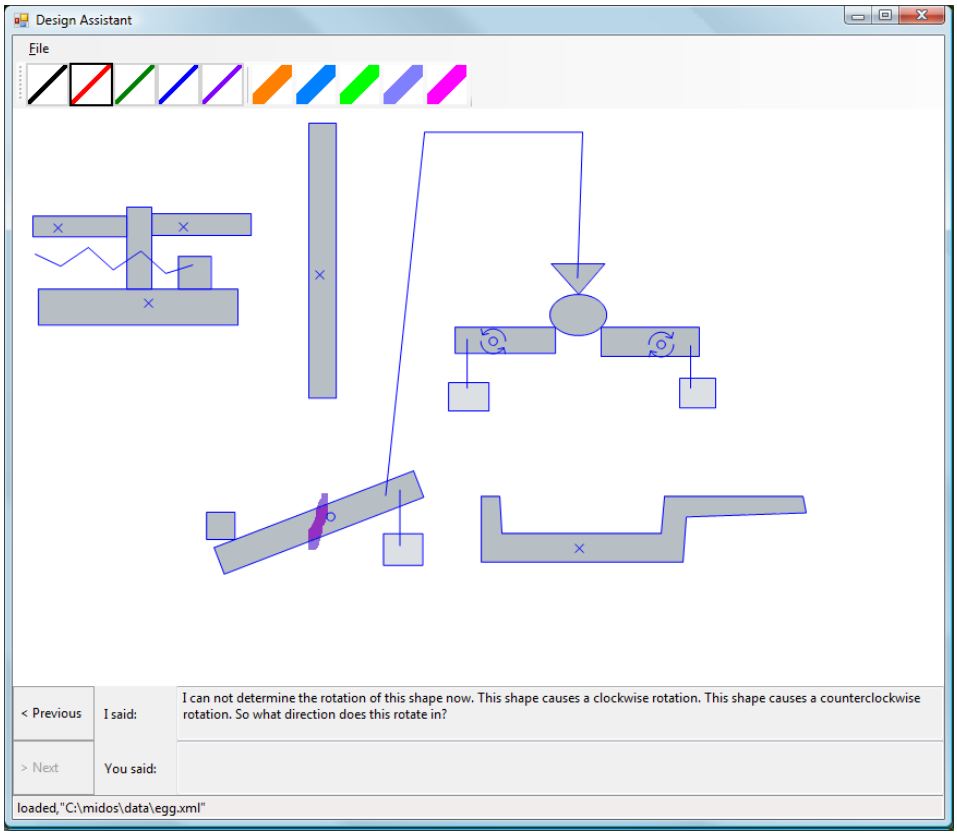

Design Assistant
File
< Previous
I said:
It looks like these shapes will collide again, do they?
> Next
You said:
yes
loaded,"C:\midos\data\egg.xml"

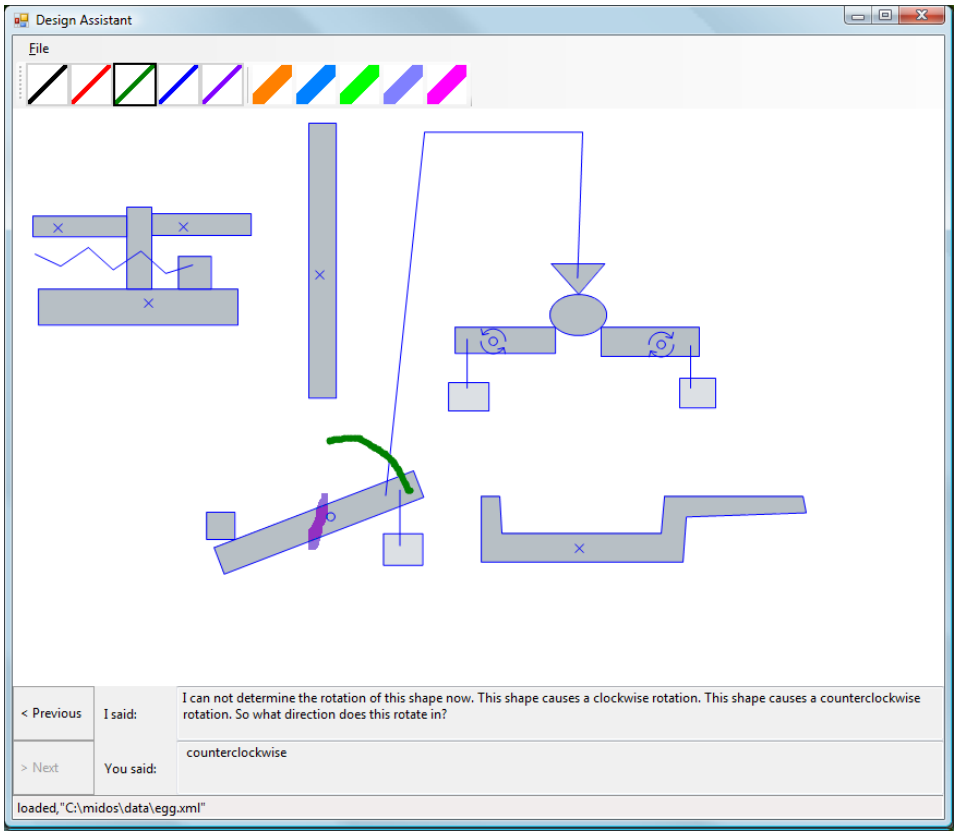

Design Assistant
File
< Previous
I said:
It looks like these shapes will collide again, do they?
> Next
You said:
loaded,"C:\midos\data\egg.xml"

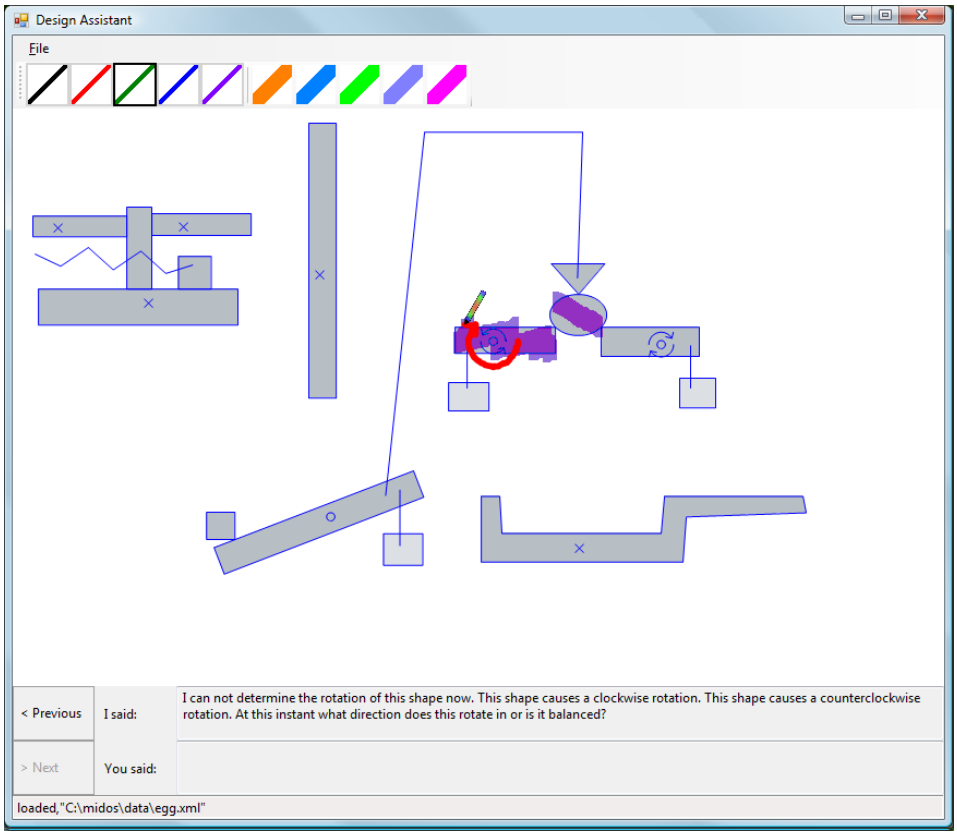

Design Assistant
File
< Previous
I said:
It looks like these shapes will collide again, do they?
> Next
You said:
yes
loaded,"C:\midos\data\egg.xml"

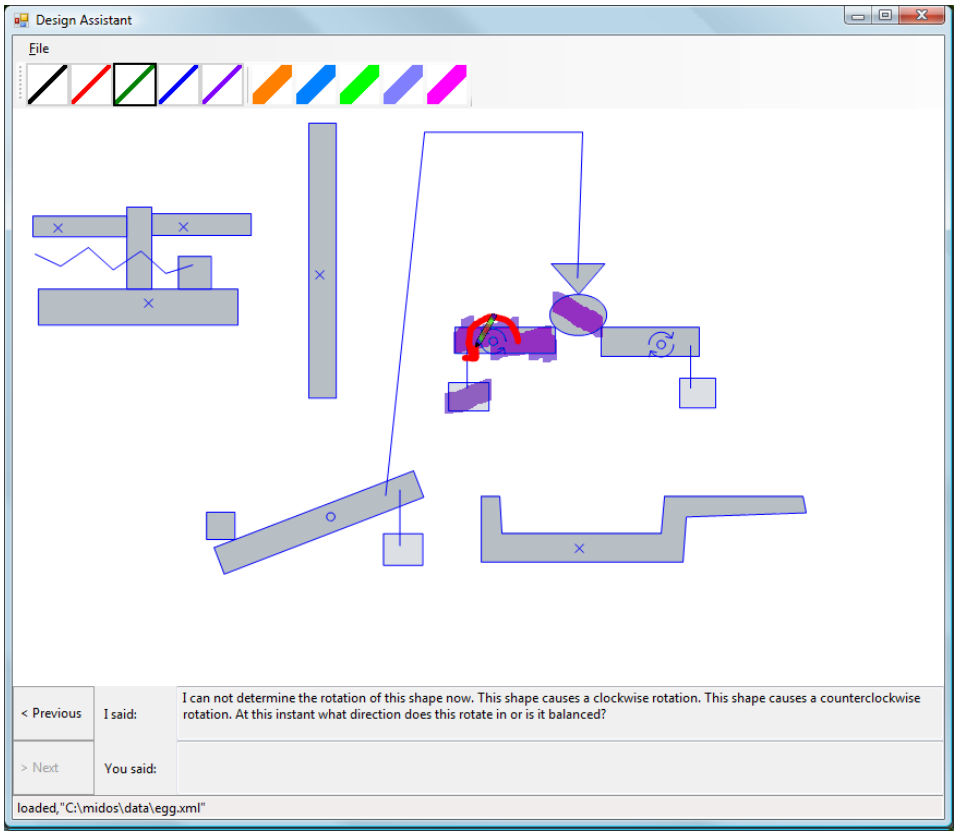

Design Assistant
File
< Previous
I said:
I can not determine the rotation of this shape now. This shape causes a clockwise rotation. This shape causes a counterclockwise rotation. So what direction does this rotate in?
> Next
You said:
loaded,"C:\midos\data\egg.xml"

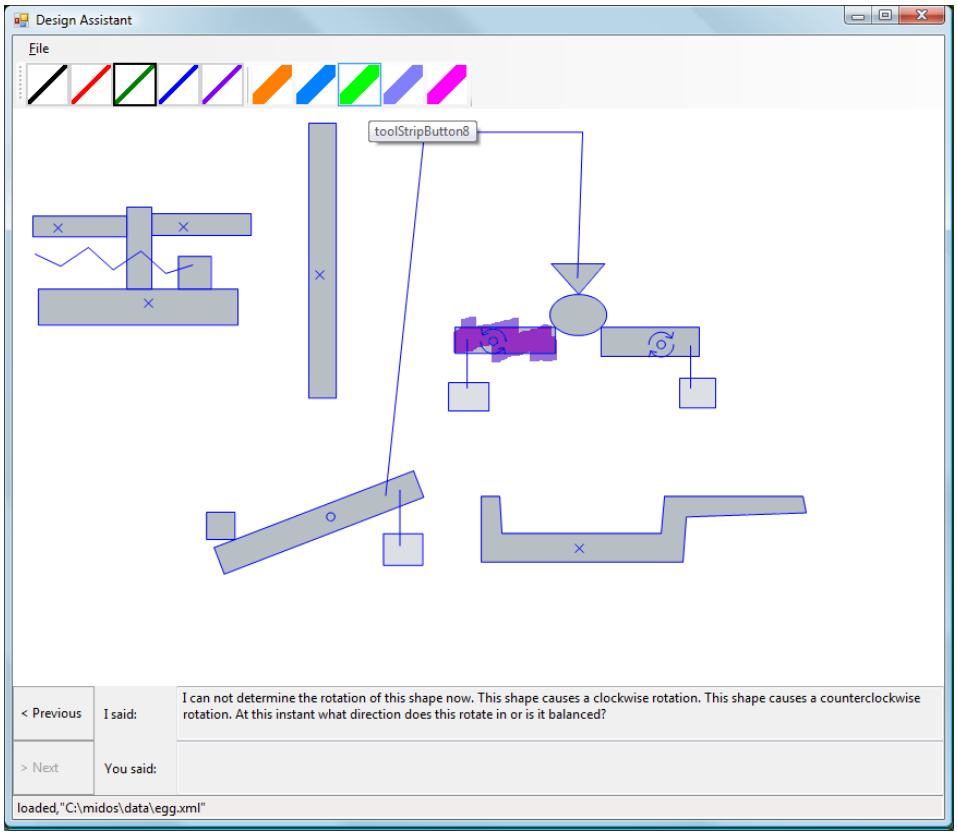

Design Assistant
File
< Previous
I said:
I can not determine the rotation of this shape now. This shape causes a clockwise rotation. This shape causes a counterclockwise rotation. So what direction does this rotate in?
> Next
You said:
loaded,"C:\midos\data\egg.xml"

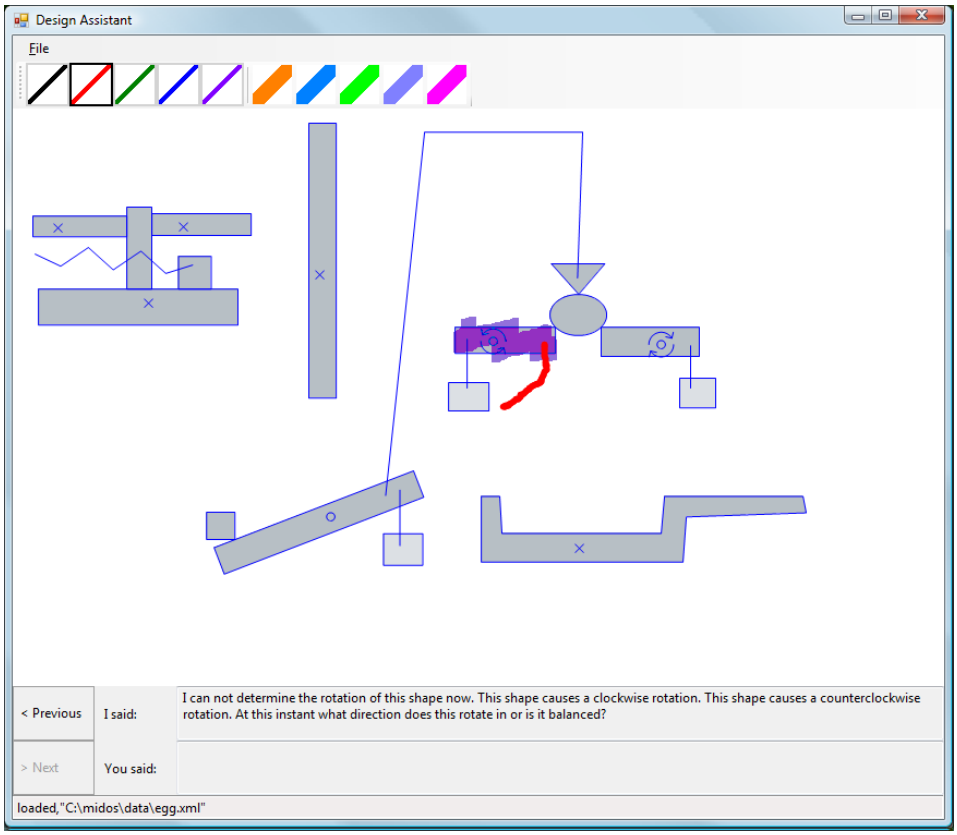

Design Assistant
File
< Previous
I said:
I can not determine the rotation of this shape now. This shape causes a clockwise rotation. This shape causes a counterclockwise rotation. So what direction does this rotate in?
> Next
You said:
loaded,"C:\midos\data\egg.xml"

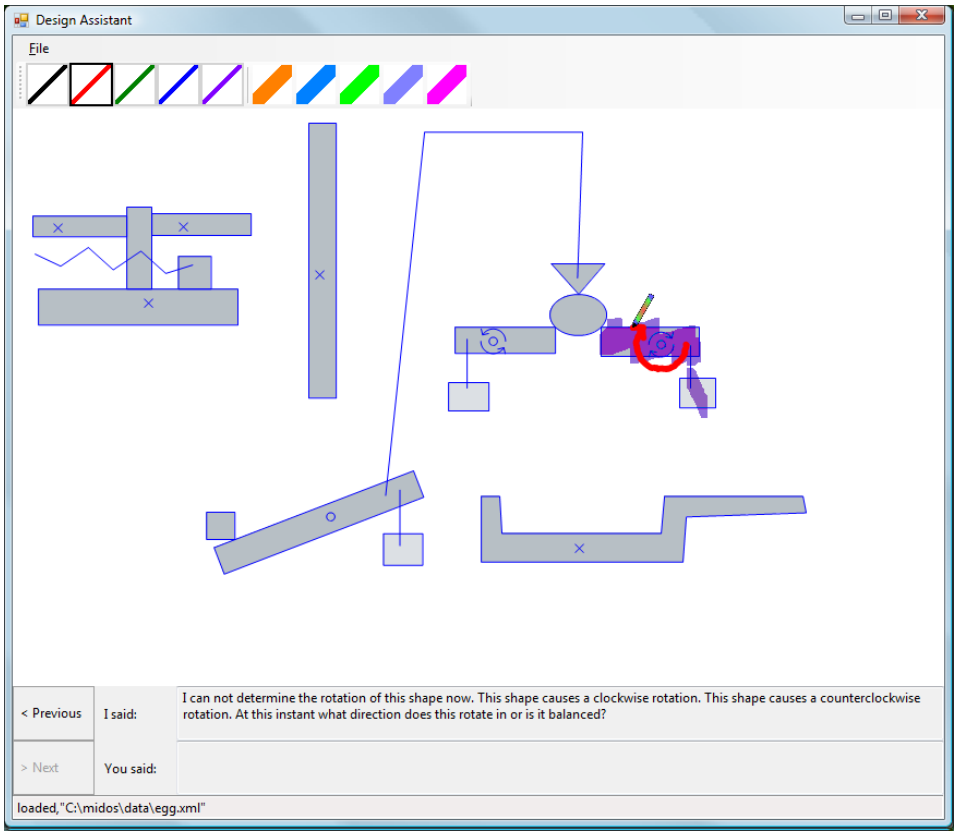

Design Assistant
File
< Previous
I said:
I can not determine the rotation of this shape now. This shape causes a clockwise rotation. This shape causes a counterclockwise rotation. So what direction does this rotate in?
> Next
You said:
counterclockwise
loaded,"C:\midos\data\egg.xml"

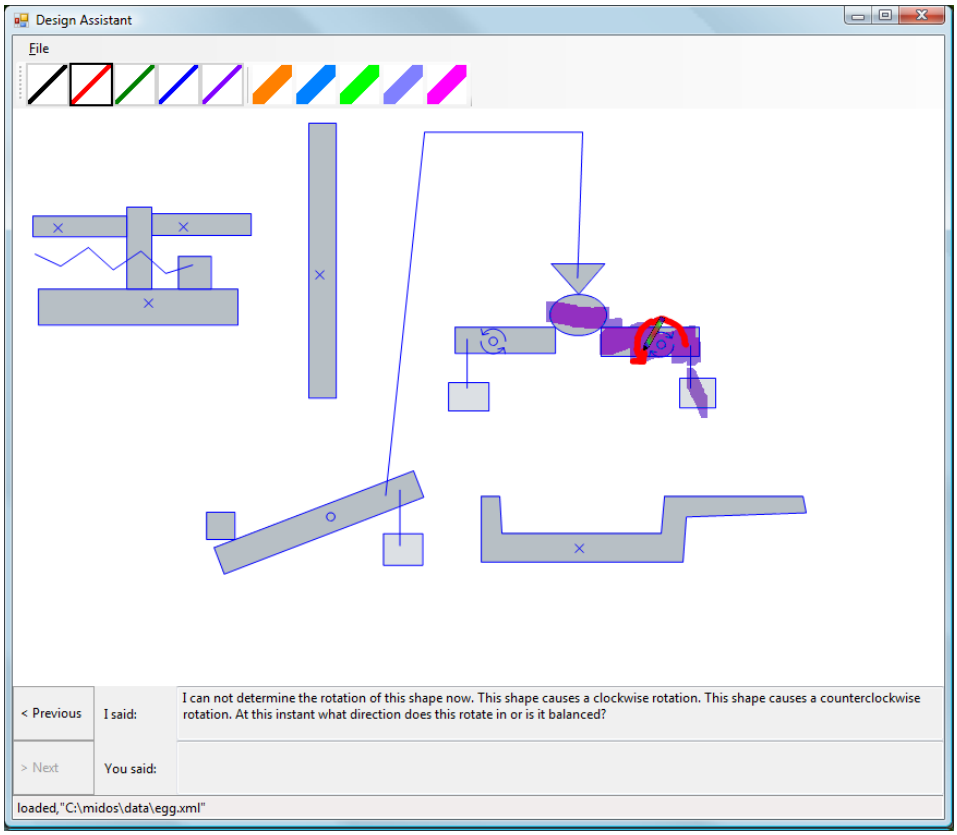

Design Assistant
File
< Previous
I said:
I can not determine the rotation of this shape now. This shape causes a clockwise rotation. This shape causes a counterclockwise rotation. At this instant what direction does this rotate in or is it balanced?
> Next
You said:
loaded,"C:\midos\data\egg.xml"

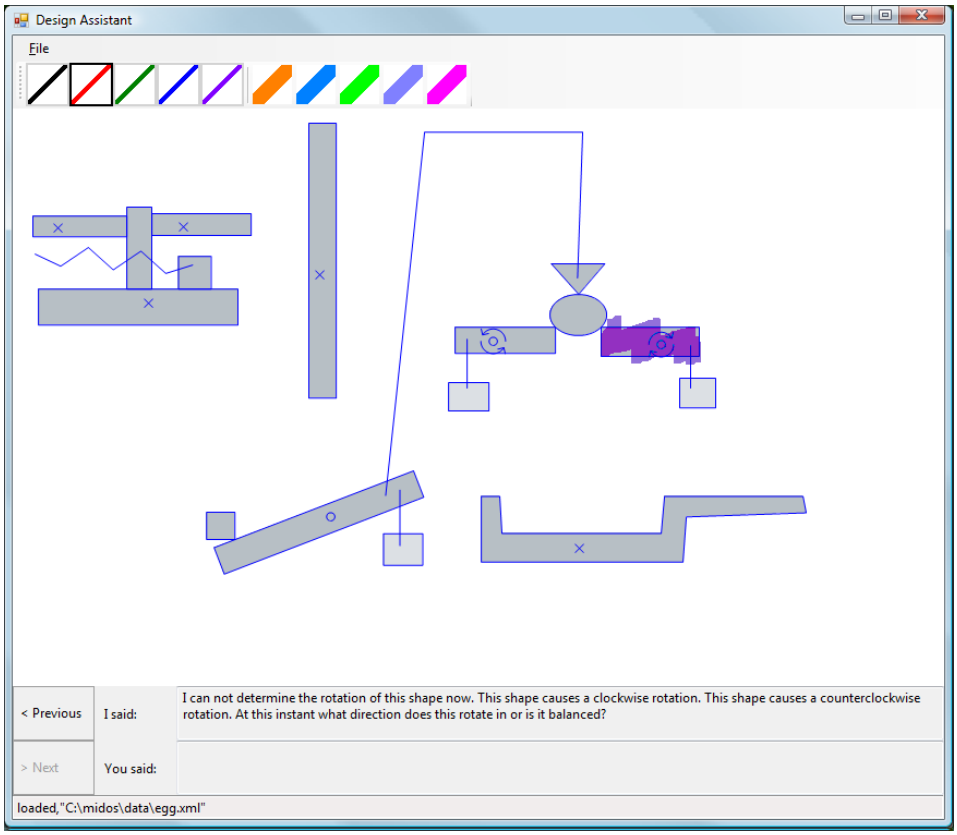

Design Assistant
File
< Previous
I said:
I can not determine the rotation of this shape now. This shape causes a clockwise rotation. This shape causes a counterclockwise rotation. At this instant what direction does this rotate in or is it balanced?
> Next
You said:
loaded,"C:\midos\data\egg.xml"

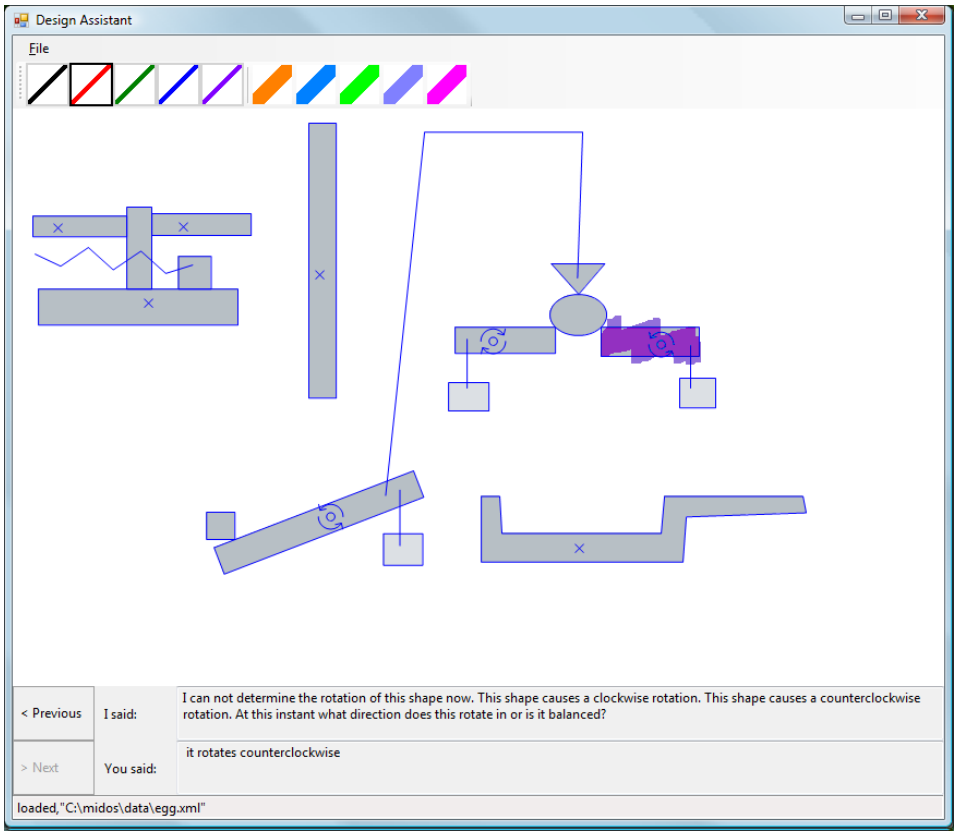

Design Assistant
File
toolStripButton8
< Previous
I said:
I can not determine the rotation of this shape now. This shape causes a clockwise rotation. This shape causes a counterclockwise rotation. At this instant what direction does this rotate in or is it balanced?
> Next
You said:
loaded,"C:\midos\data\egg.xml"

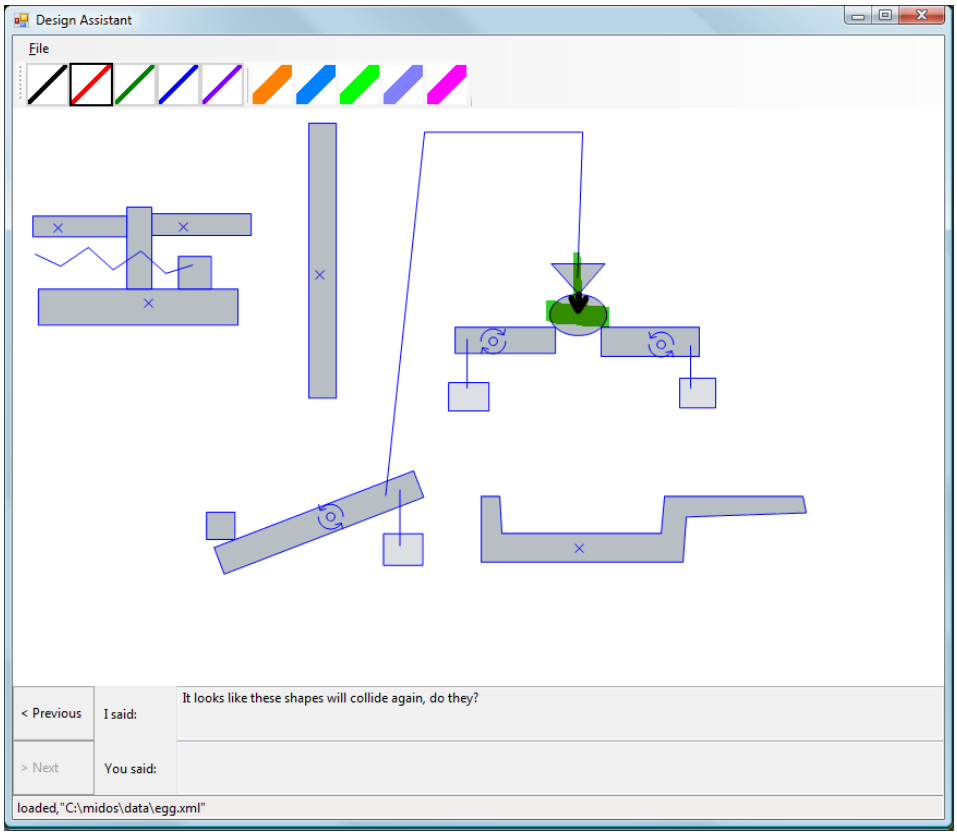

Design Assistant
File
< Previous
I said:
I can not determine the rotation of this shape now. This shape causes a clockwise rotation. This shape causes a counterclockwise rotation. At this instant what direction does this rotate in or is it balanced?
> Next
You said:
loaded,"C:\midos\data\egg.xml"

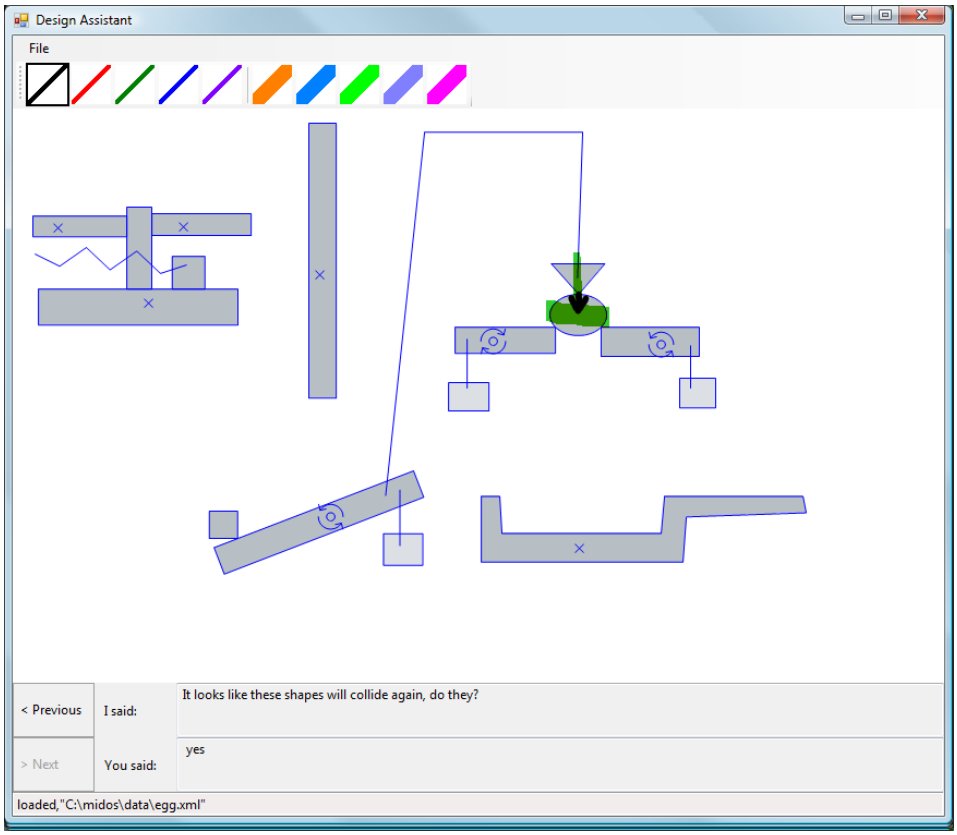

Design Assistant
File
< Previous
I said:
I can not determine the rotation of this shape now. This shape causes a clockwise rotation. This shape causes a counterclockwise rotation. At this instant what direction does this rotate in or is it balanced?
> Next
You said:
loaded,"C:\midos\data\egg.xml"

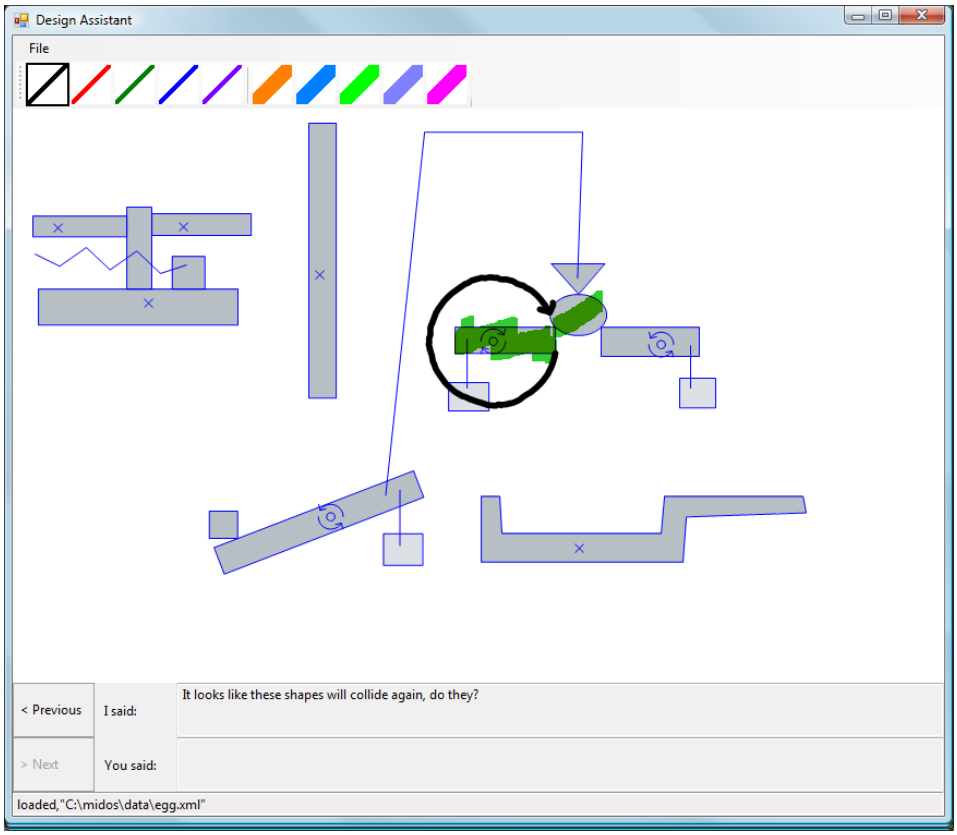

Design Assistant
File
< Previous
I said:
I can not determine the rotation of this shape now. This shape causes a clockwise rotation. This shape causes a counterclockwise rotation. At this instant what direction does this rotate in or is it balanced?
> Next
You said:
loaded,"C:\midos\data\egg.xml"

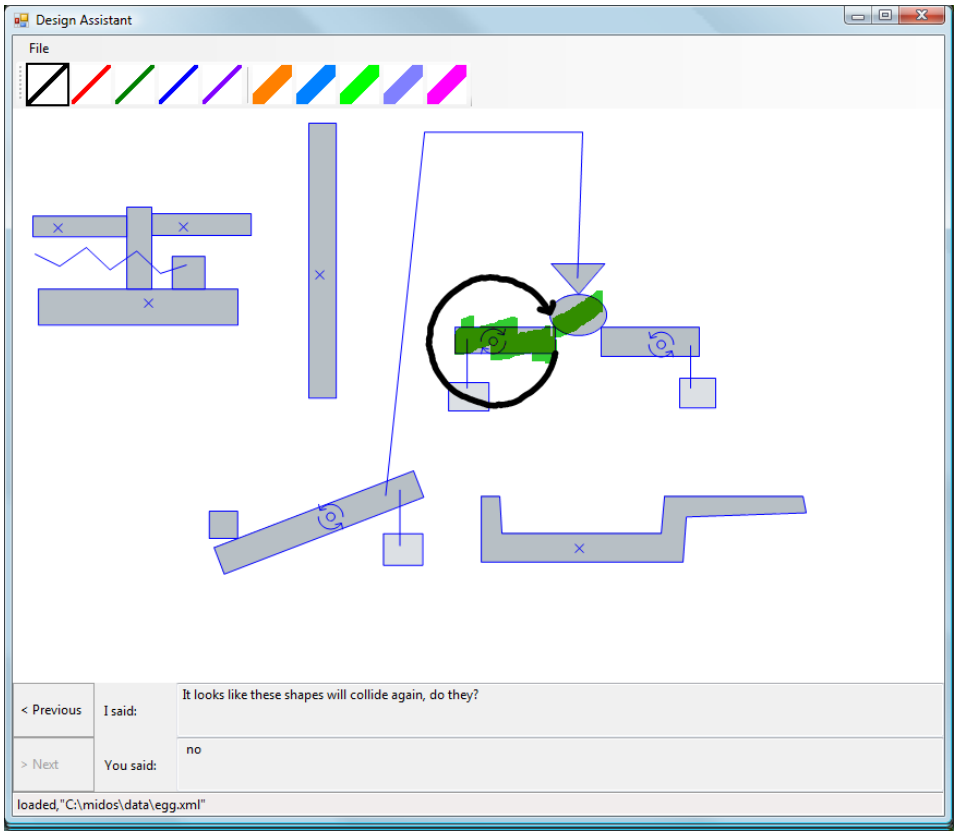

Design Assistant
File
< Previous
I said:
I can not determine the rotation of this shape now. This shape causes a clockwise rotation. This shape causes a counterclockwise rotation. At this instant what direction does this rotate in or is it balanced?
> Next
You said:
loaded,"C:\midos\data\egg.xml"

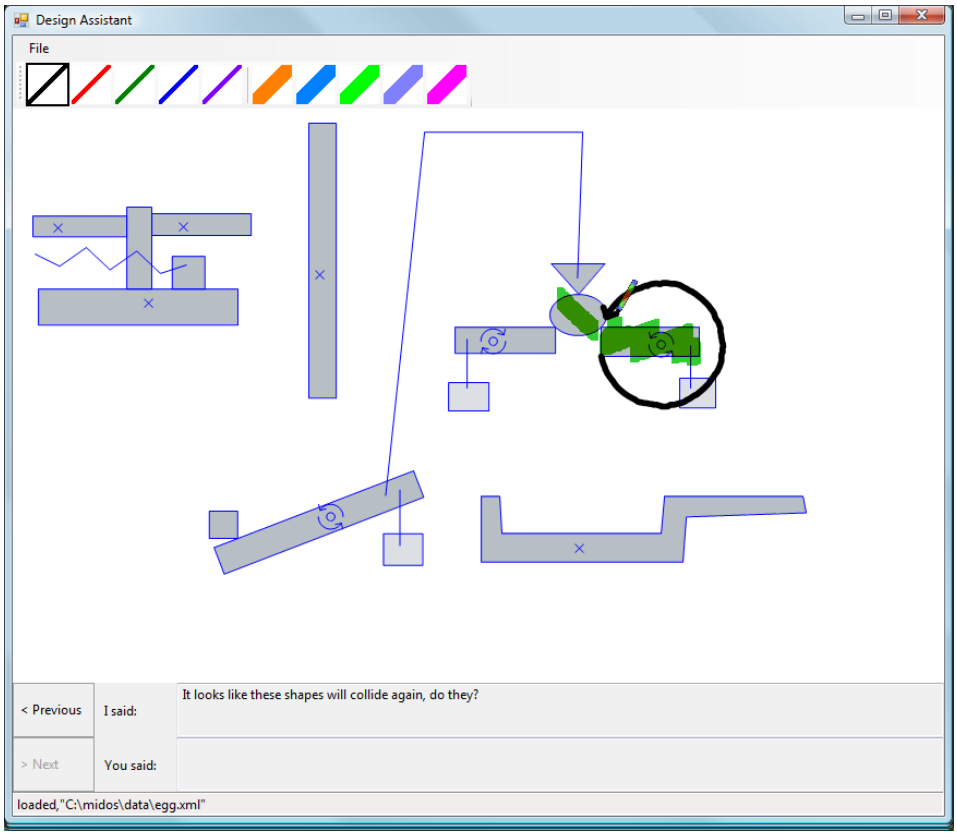

Design Assistant
File
< Previous
I said:
I can not determine the rotation of this shape now. This shape causes a clockwise rotation. This shape causes a counterclockwise rotation. At this instant what direction does this rotate in or is it balanced?
> Next
You said:
it rotates counterclockwise
loaded,"C:\midos\data\egg.xml"

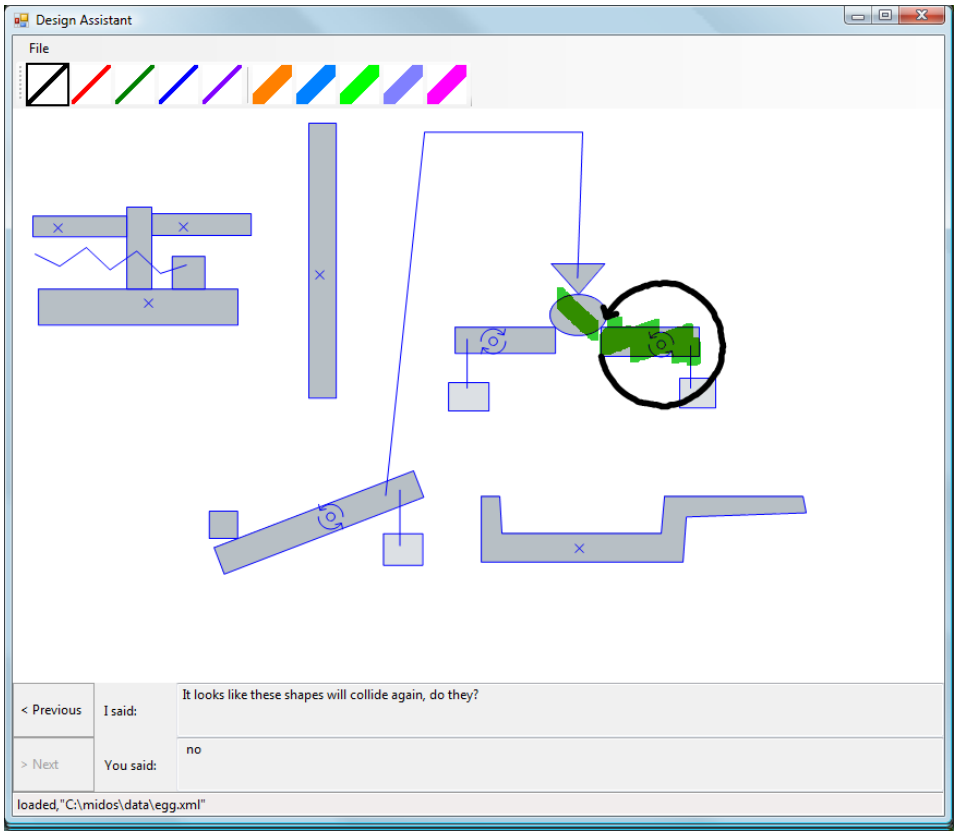

Design Assistant
File
< Previous
I said:
It looks like these shapes will collide again, do they?
> Next
You said:
loaded,"C:\midos\data\egg.xml"

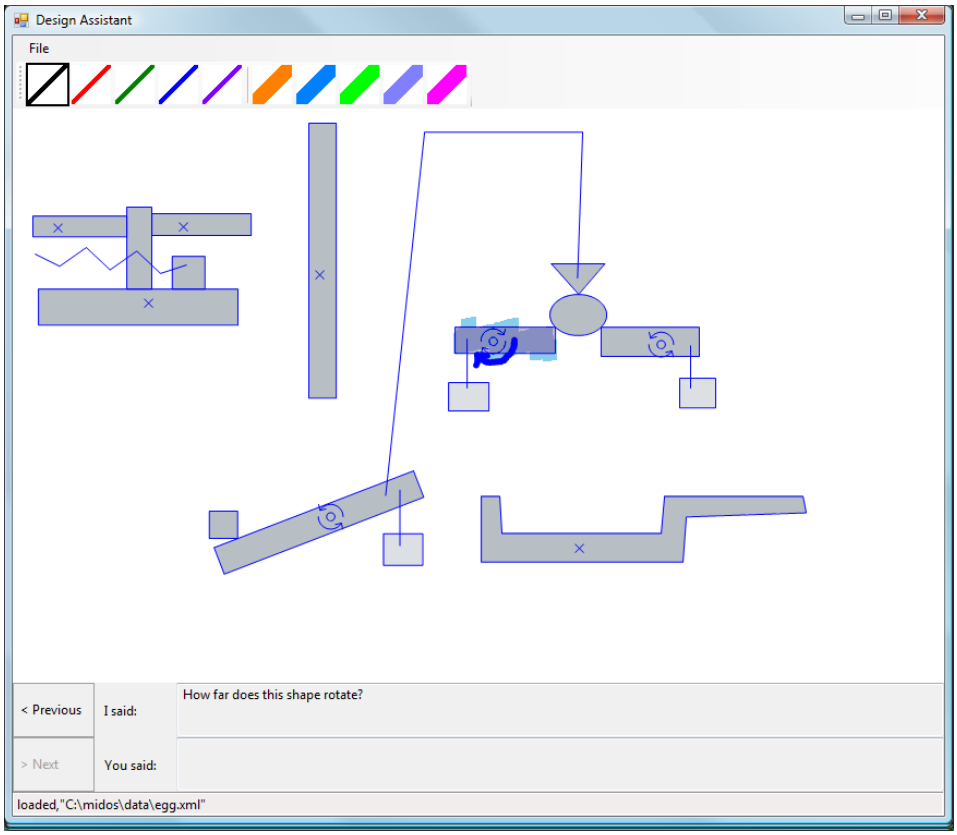

Design Assistant
File
< Previous
I said:
It looks like these shapes will collide again, do they?
> Next
You said:
yes
loaded,"C:\midos\data\egg.xml"

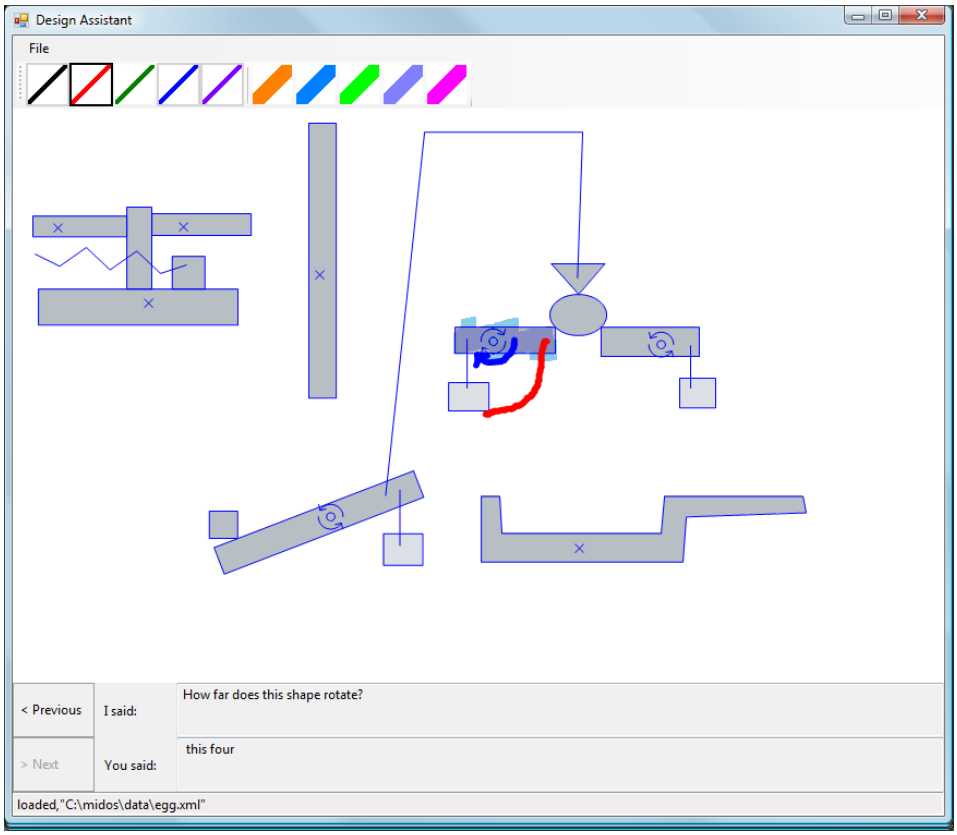

Design Assistant
File
< Previous
I said:
It looks like these shapes will collide again, do they?
> Next
You said:
loaded,"C:\midos\data\egg.xml"

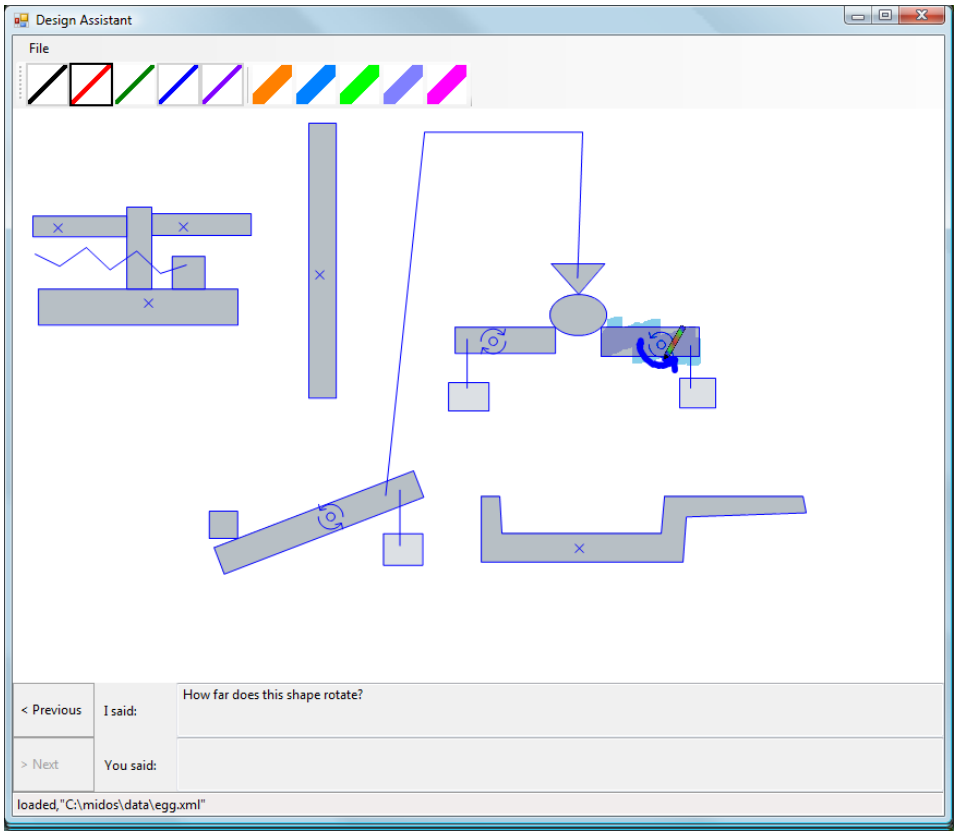

Design Assistant
File
< Previous
I said:
It looks like these shapes will collide again, do they?
> Next
You said:
no
loaded,"C:\midos\data\egg.xml"

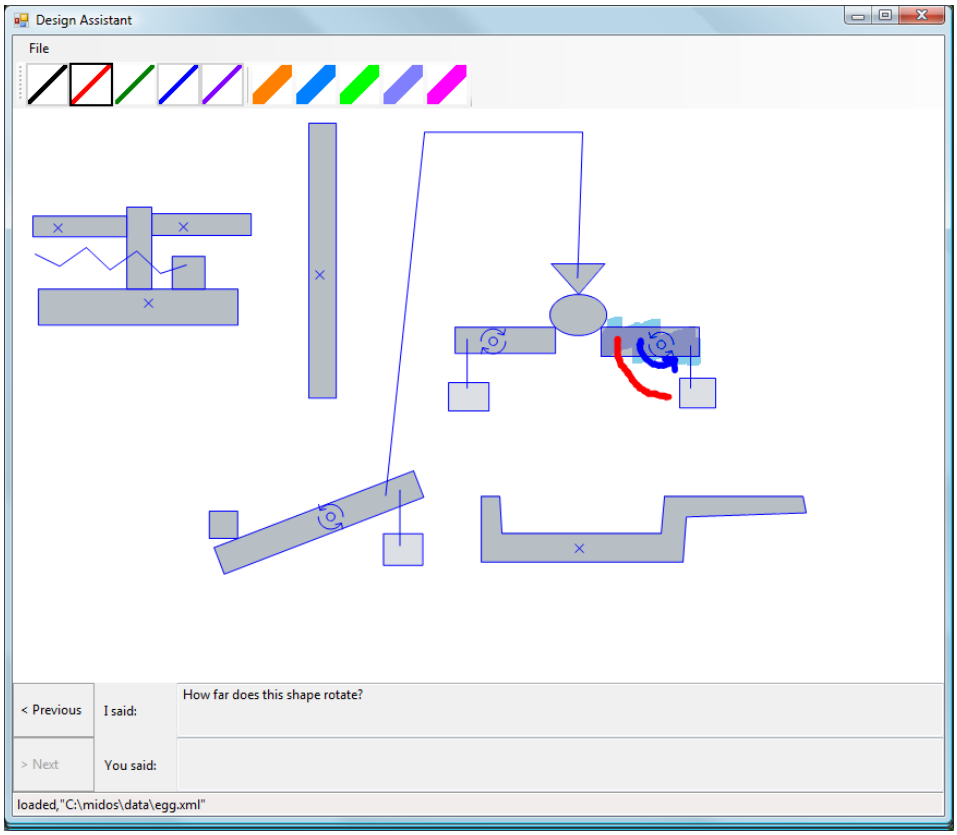

Design Assistant
File
< Previous
I said:
It looks like these shapes will collide again, do they?
> Next
You said:
loaded,"C:\midos\data\egg.xml"

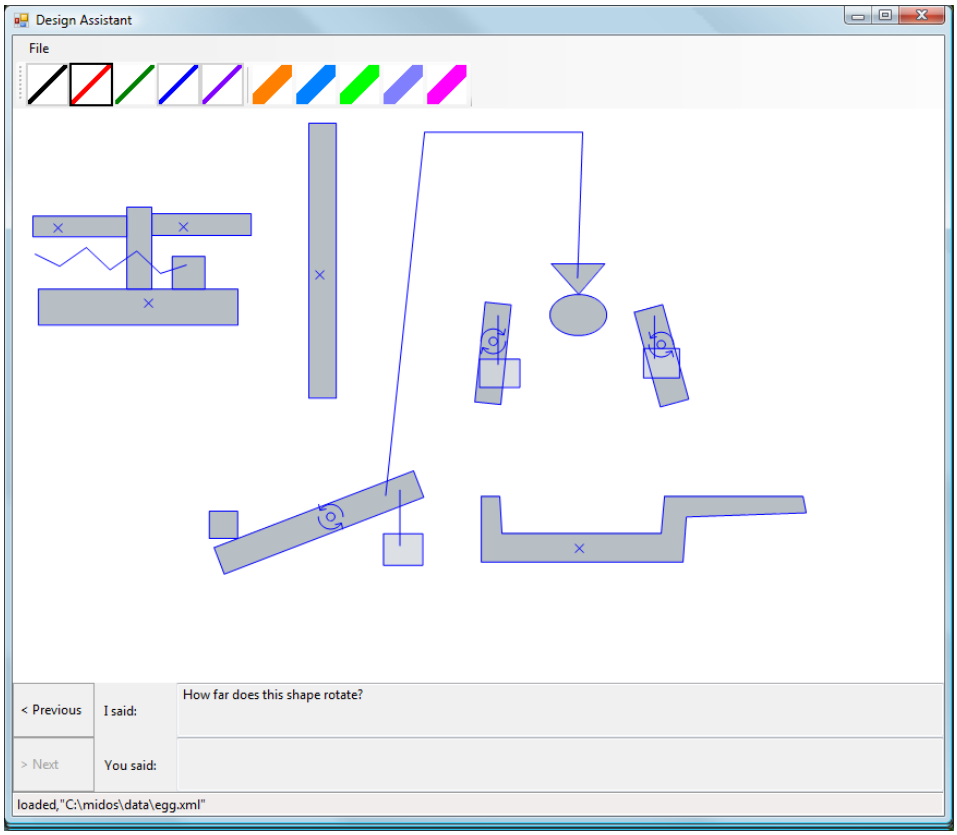

Design Assistant
File
< Previous
I said:
It looks like these shapes will collide again, do they?
> Next
You said:
no
loaded,"C:\midos\data\egg.xml"

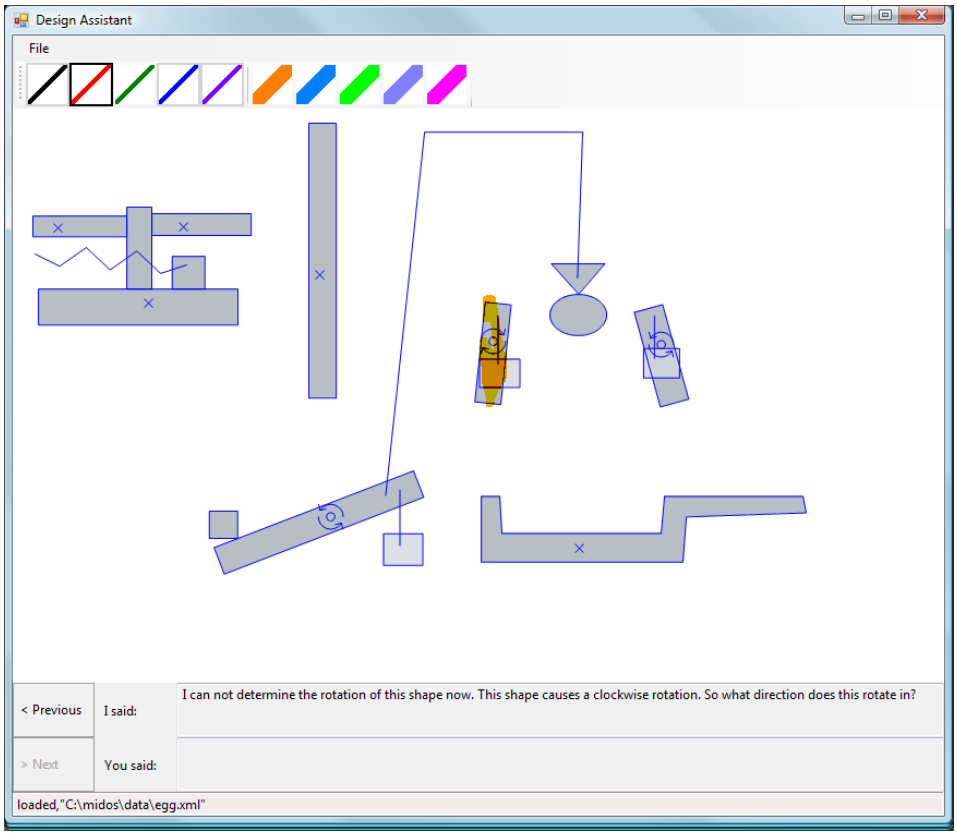

Design Assistant
File
< Previous
I said:
How far does this shape rotate?
> Next
You said:
loaded,"C:\midos\data\egg.xml"

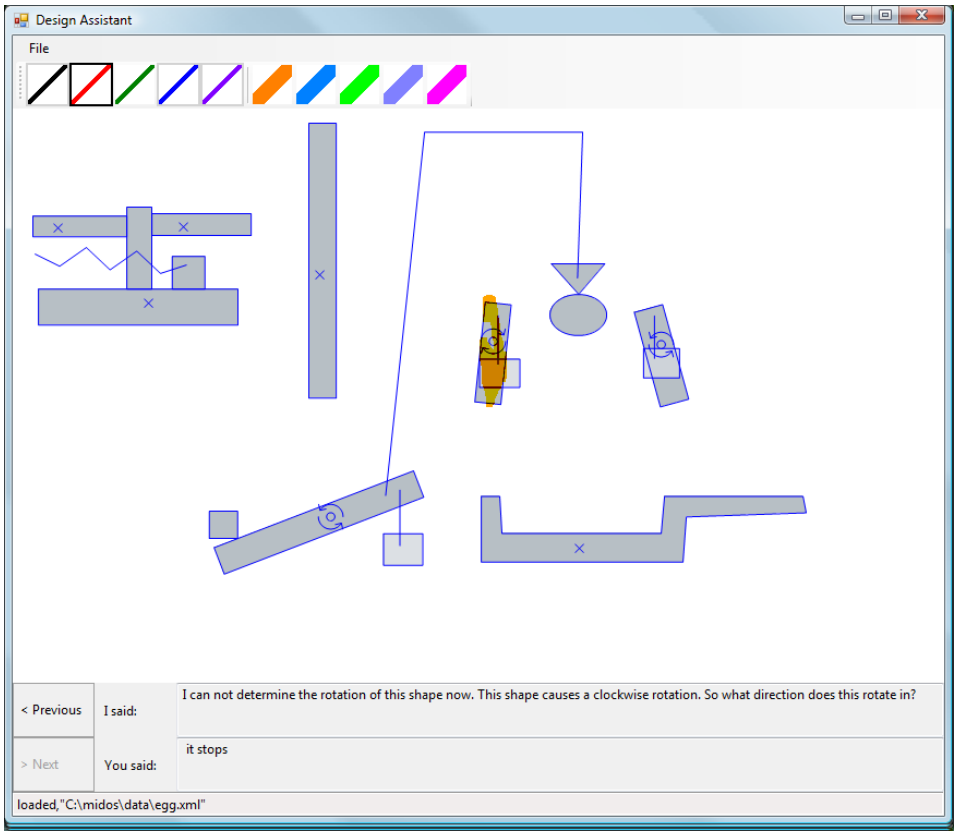

Design Assistant
File
< Previous
I said:
How far does this shape rotate?
> Next
You said:
this four
loaded,"C:\midos\data\egg.xml"

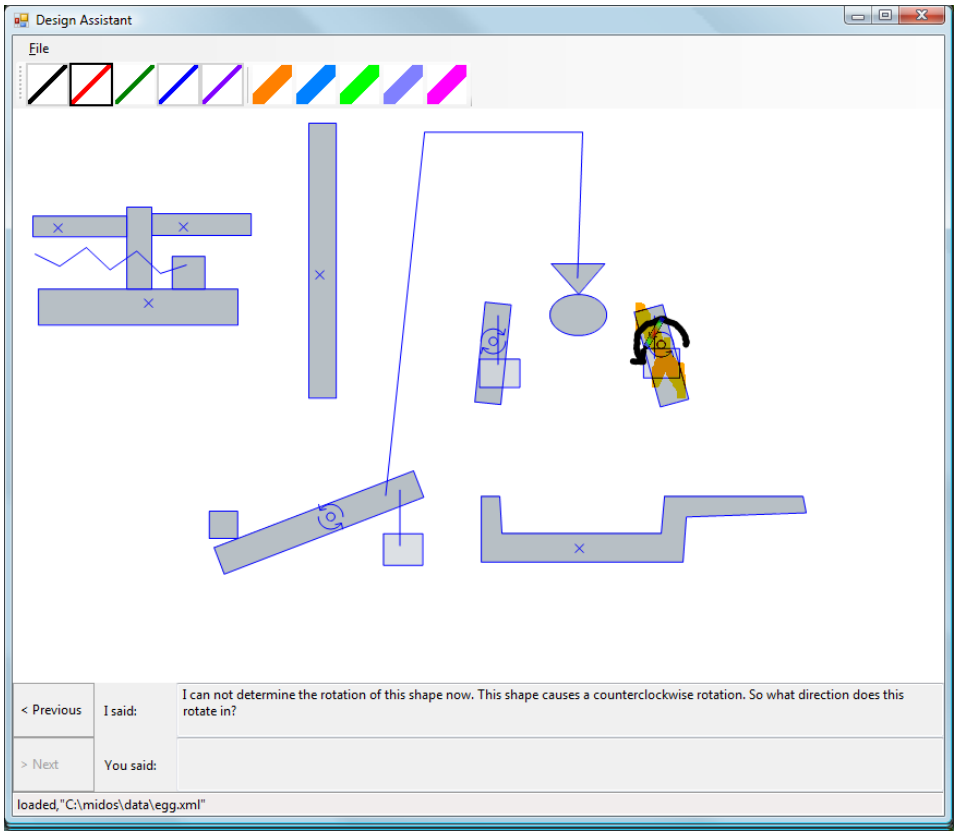

Design Assistant
File
< Previous
I said:
How far does this shape rotate?
> Next
You said:
loaded,"C:\midos\data\egg.xml"

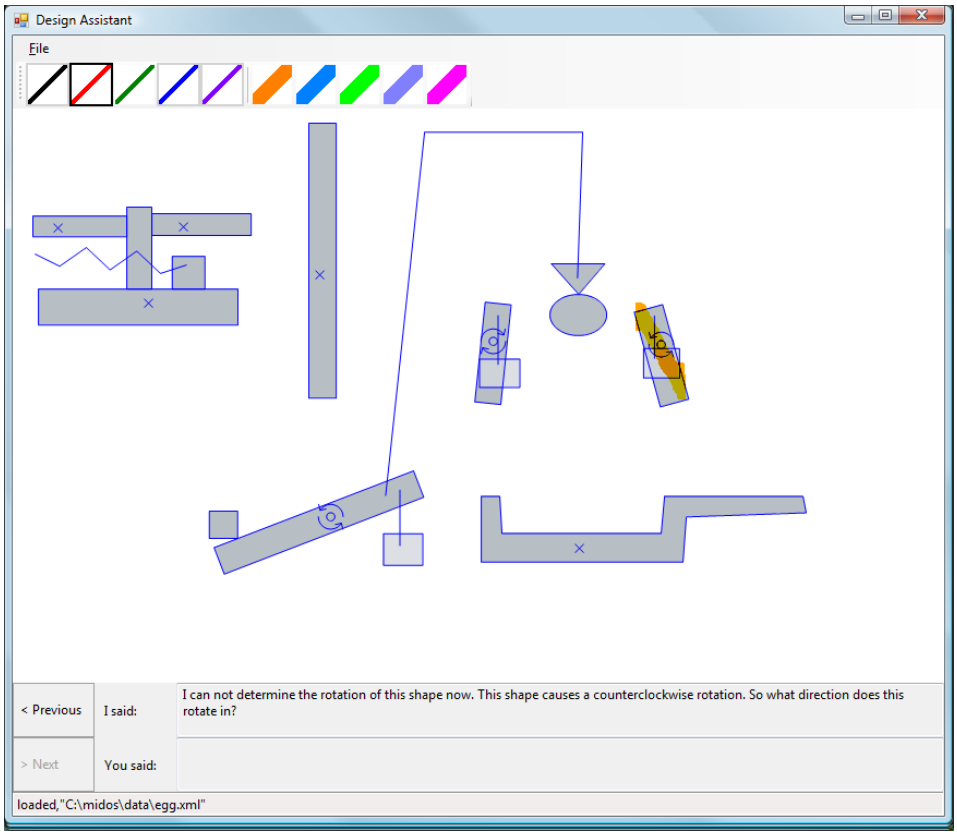

Design Assistant
File
< Previous
I said:
How far does this shape rotate?
> Next
You said:
loaded,"C:\midos\data\egg.xml"

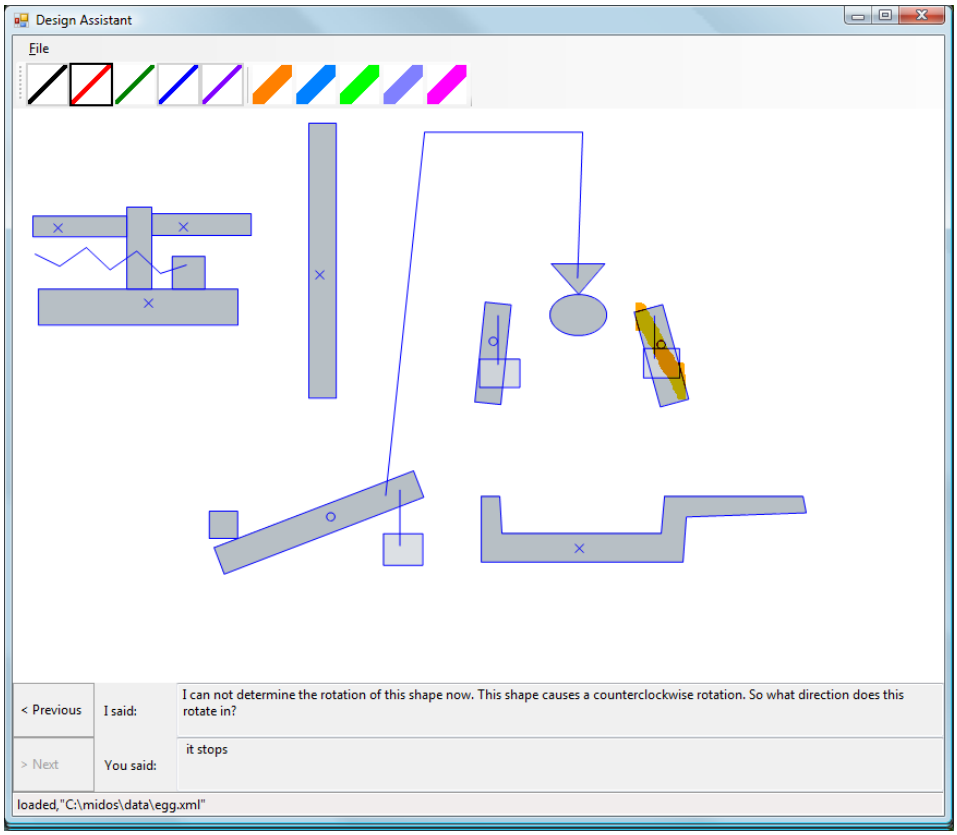

Design Assistant
File
< Previous
I said:
How far does this shape rotate?
> Next
You said:
loaded,"C:\midos\data\egg.xml"

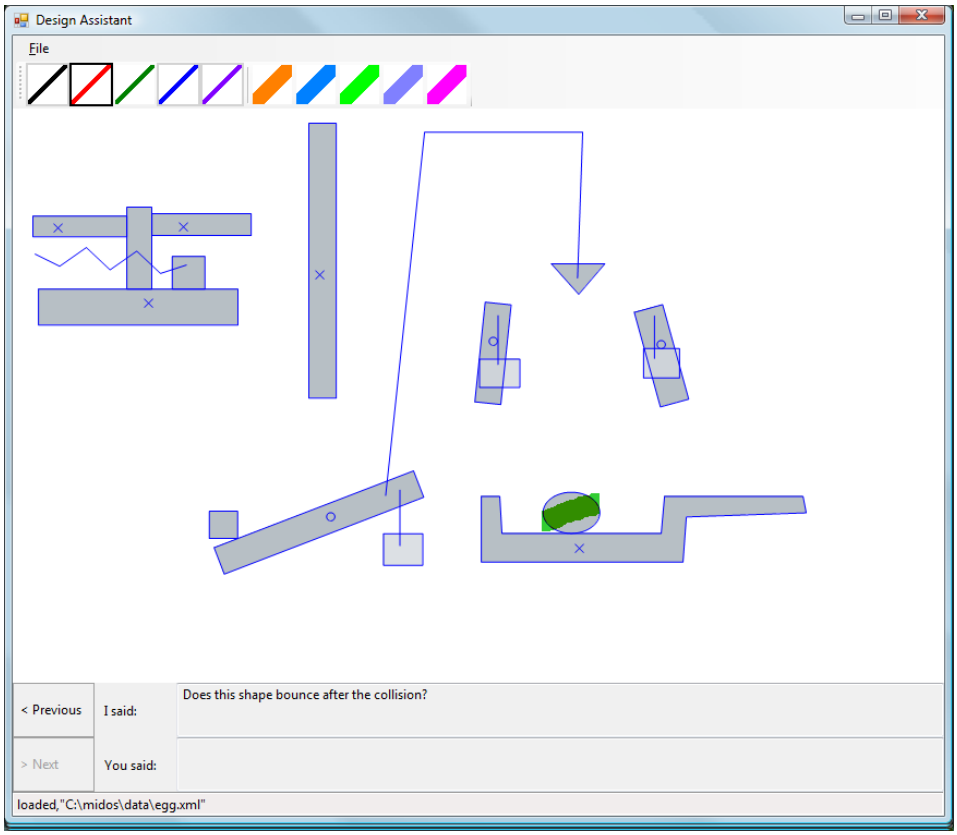

Design Assistant
File
< Previous
I said:
I can not determine the rotation of this shape now. This shape causes a clockwise rotation. So what direction does this rotate in?
> Next
You said:
loaded,"C:\midos\data\egg.xml"

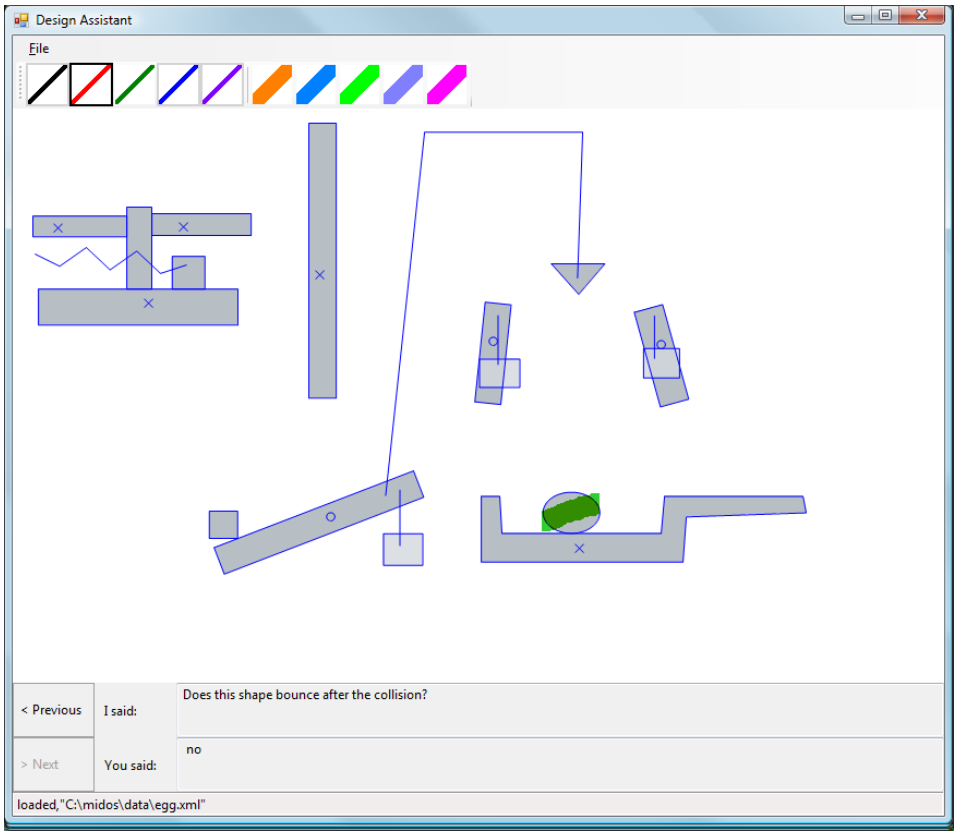

Design Assistant
File
< Previous
I said:
I can not determine the rotation of this shape now. This shape causes a clockwise rotation. So what direction does this rotate in?
> Next
You said:
it stops
loaded,"C:\midos\data\egg.xml"

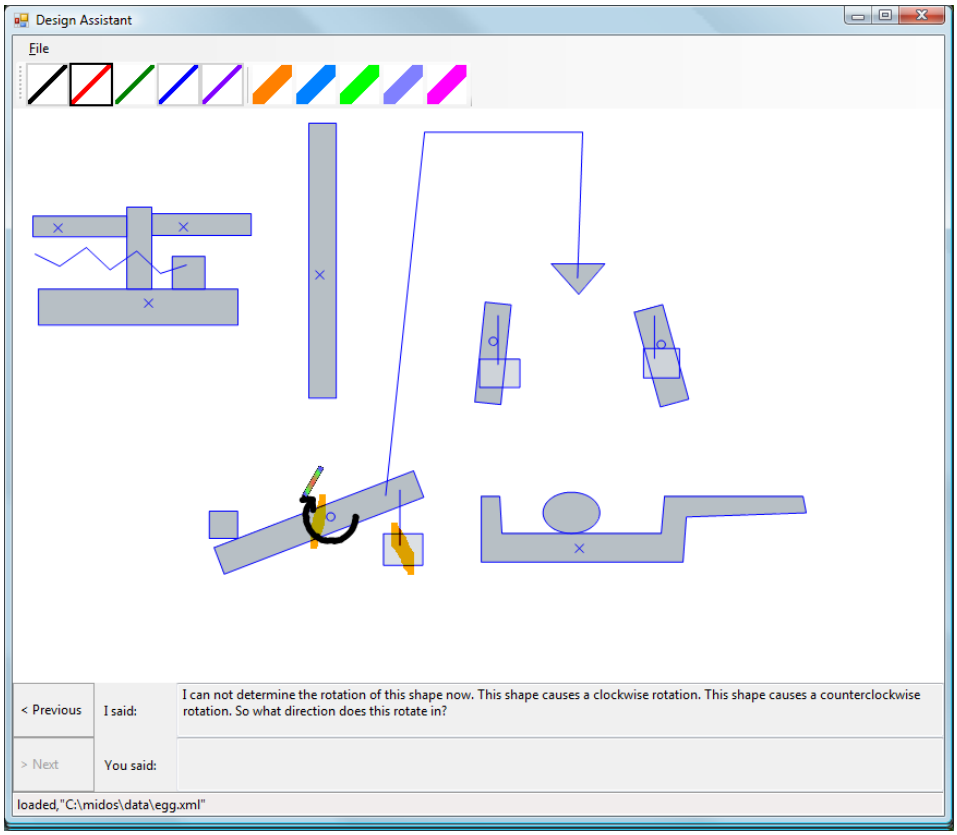

Design Assistant
File
< Previous
I said:
I can not determine the rotation of this shape now. This shape causes a counterclockwise rotation. So what direction does this rotate in?
> Next
You said:
loaded,"C:\midos\data\egg.xml"

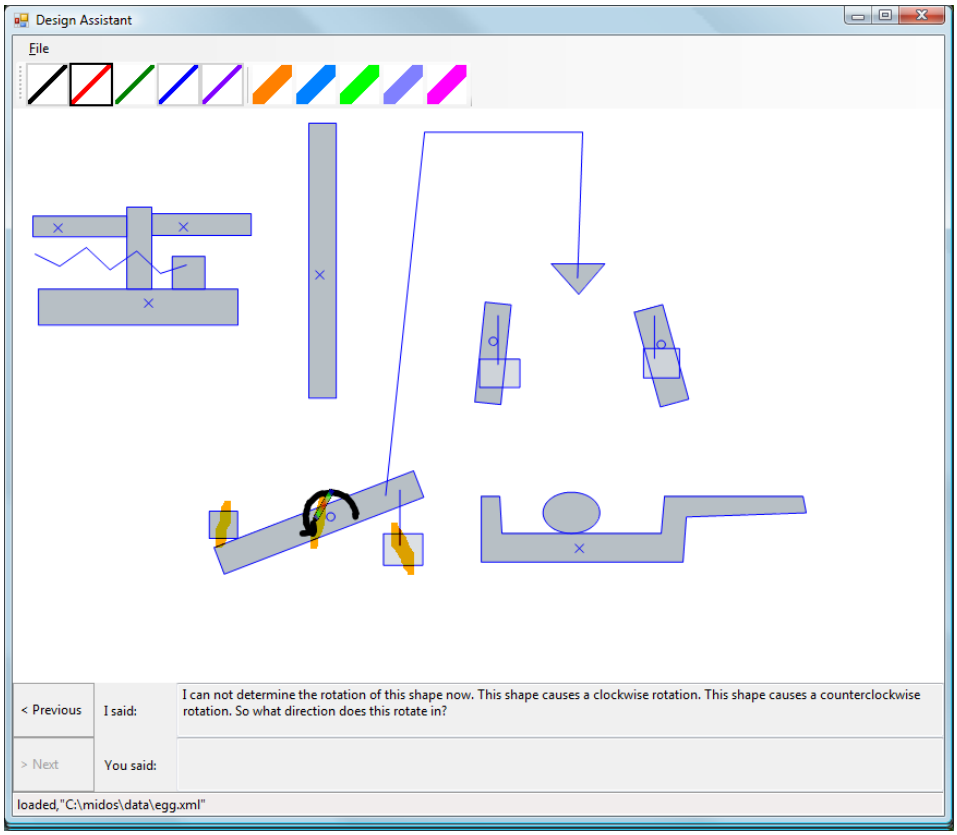

Design Assistant
File
< Previous
I said:
I can not determine the rotation of this shape now. This shape causes a counterclockwise rotation. So what direction does this rotate in?
> Next
You said:
loaded,"C:\midos\data\egg.xml"

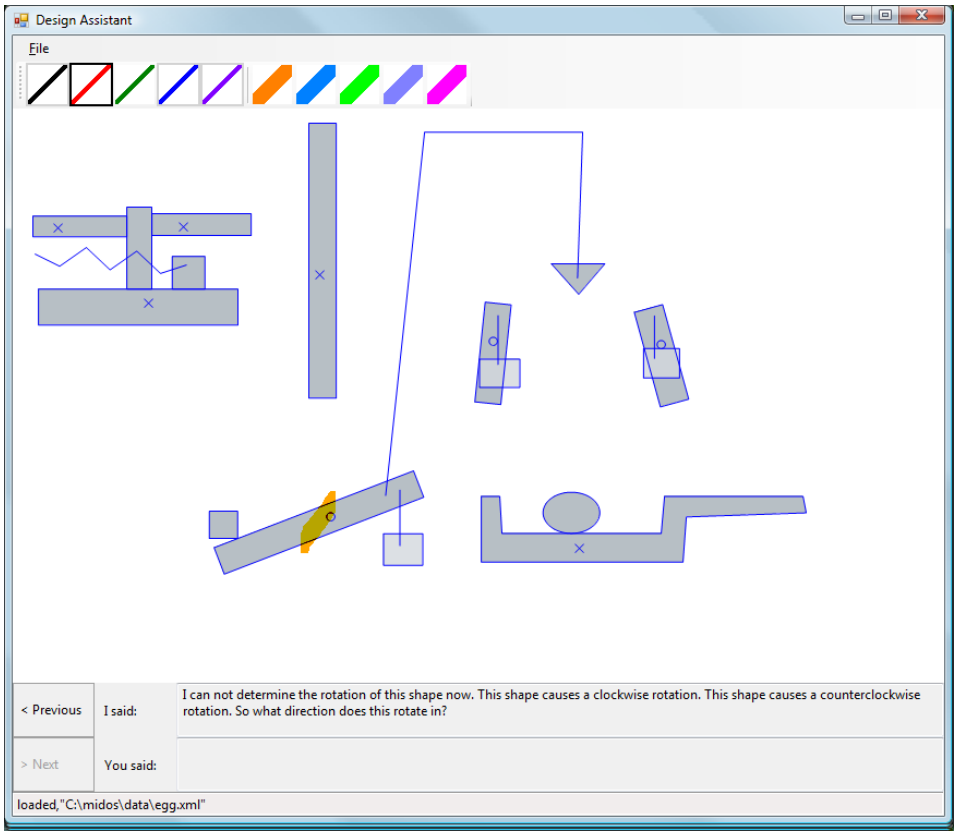

Design Assistant
File
< Previous
I said:
I can not determine the rotation of this shape now. This shape causes a counterclockwise rotation. So what direction does this rotate in?
> Next
You said:
it stops
loaded,"C:\midos\data\egg.xml"

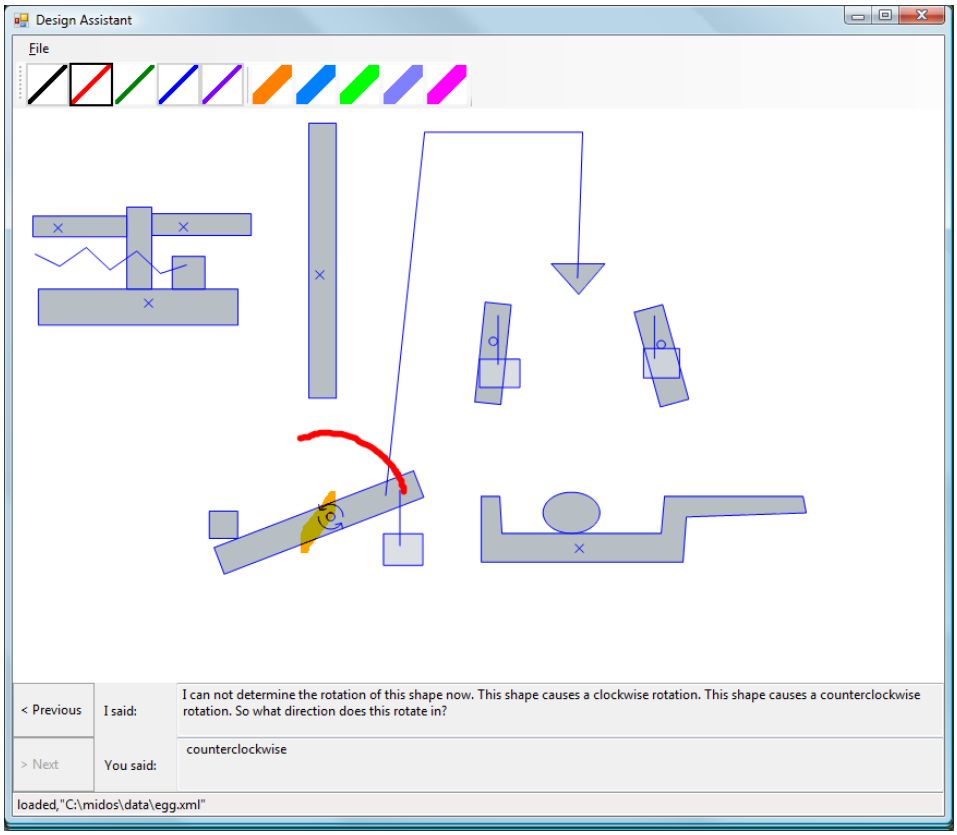

Design Assistant
File
< Previous
I said:
Does this shape bounce after the collision?
> Next
You said:
loaded,"C:\midos\data\egg.xml"

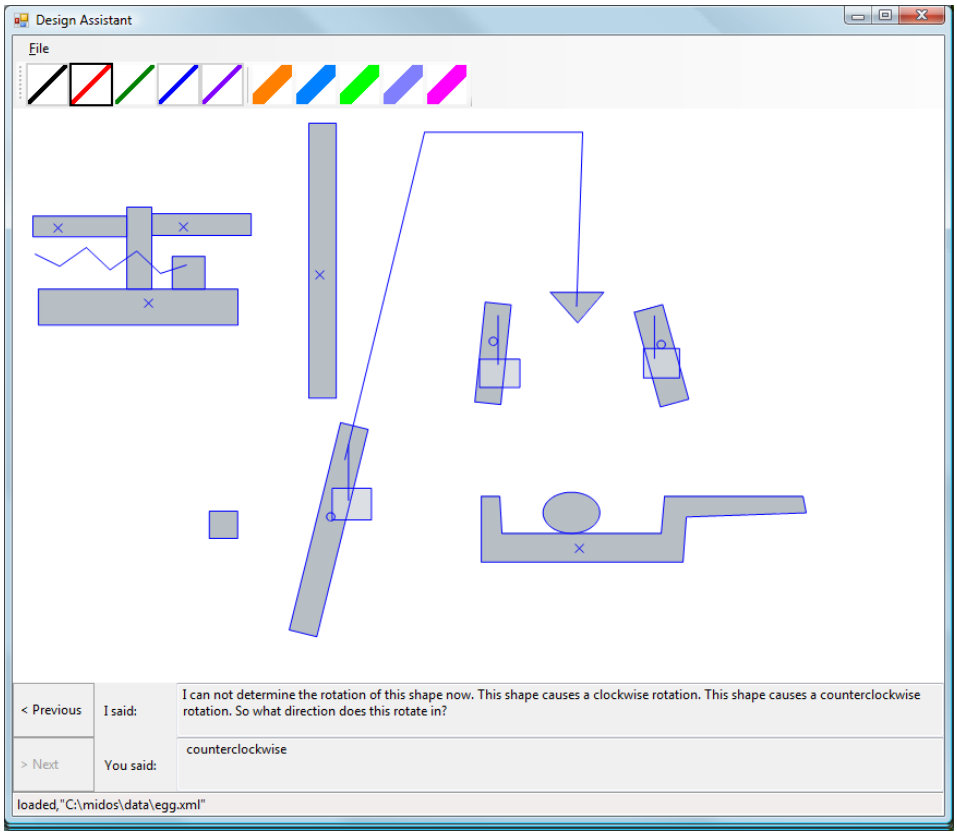

Design Assistant
File
< Previous
I said:
Does this shape bounce after the collision?
> Next
You said:
no
loaded,"C:\midos\data\egg.xml"

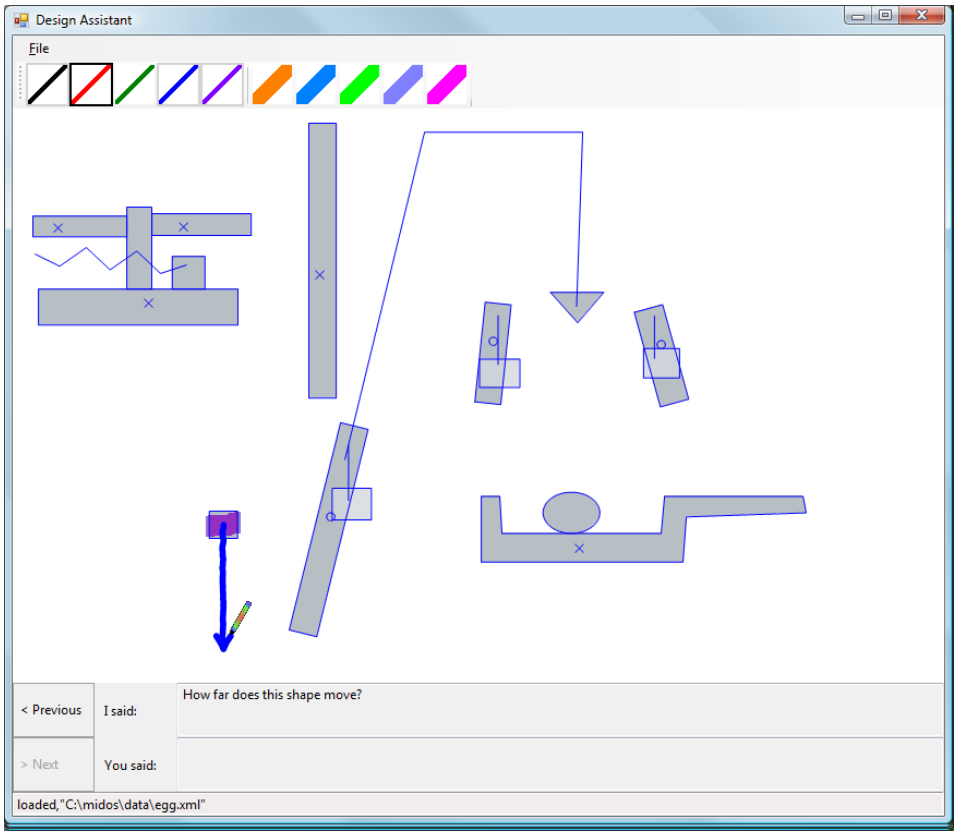

Design Assistant
File
< Previous
I said:
I can not determine the rotation of this shape now. This shape causes a clockwise rotation. This shape causes a counterclockwise rotation. So what direction does this rotate in?
> Next
You said:
loaded,"C:\midos\data\egg.xml"

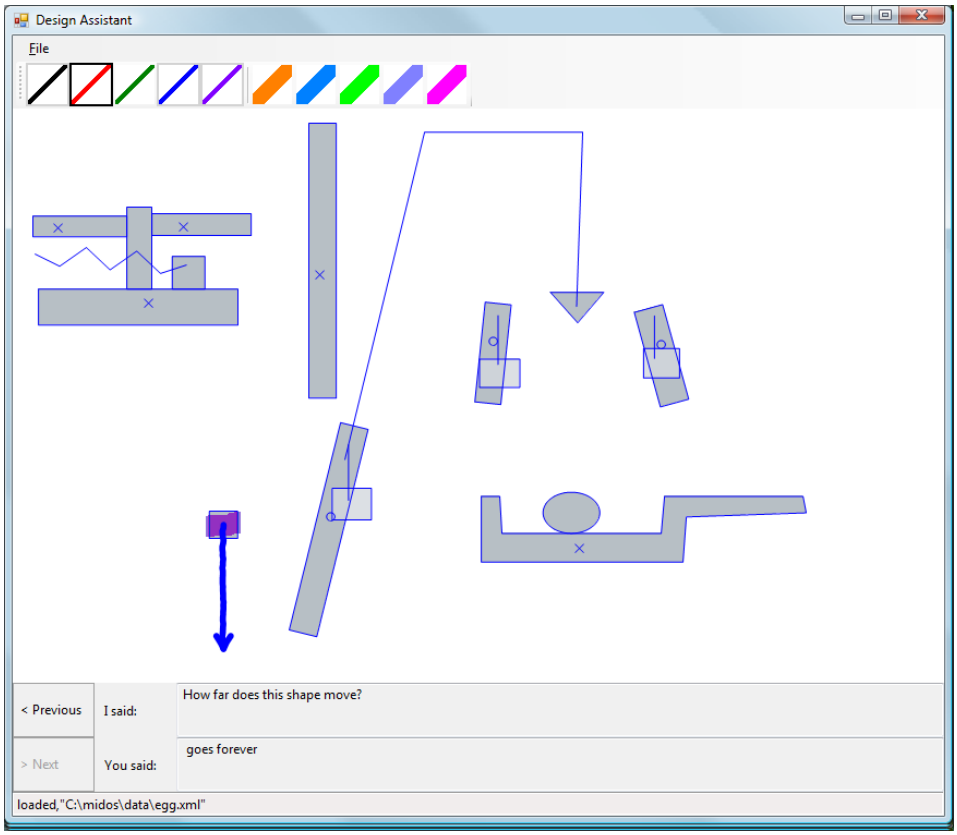

Design Assistant
File
< Previous
I said:
I can not determine the rotation of this shape now. This shape causes a clockwise rotation. This shape causes a counterclockwise rotation. So what direction does this rotate in?
> Next
You said:
loaded,"C:\midos\data\egg.xml"

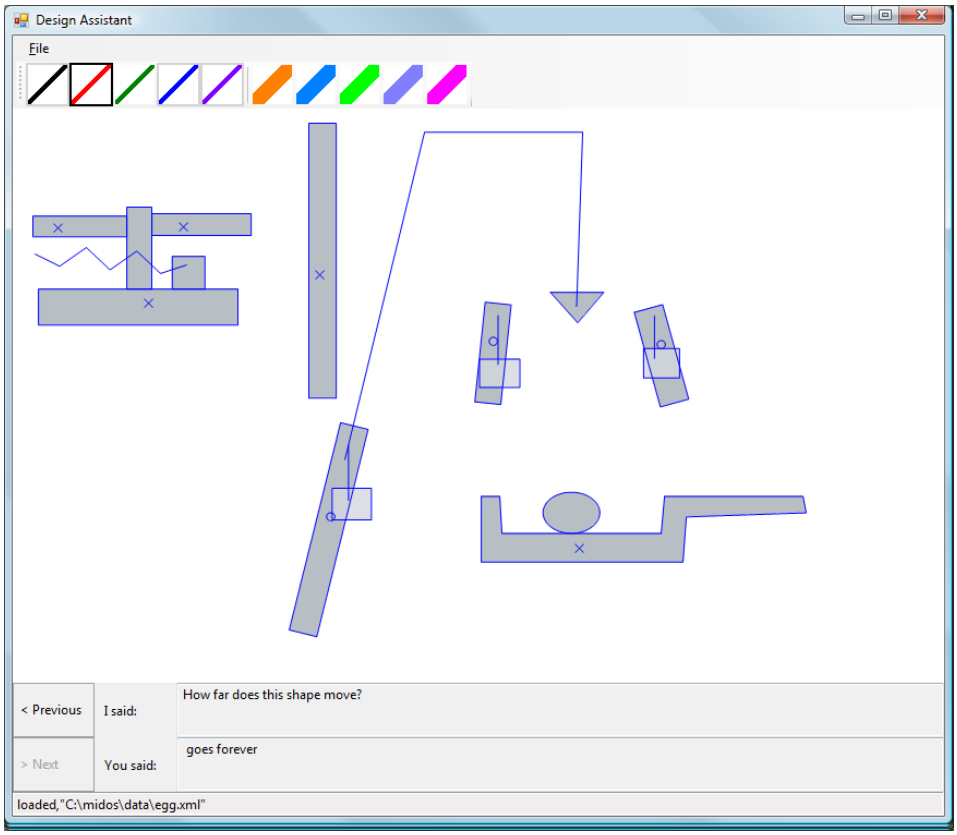

Design Assistant
File
< Previous
I said:
I can not determine the rotation of this shape now. This shape causes a clockwise rotation. This shape causes a counterclockwise rotation. So what direction does this rotate in?
> Next
You said:
loaded,"C:\midos\data\egg.xml"

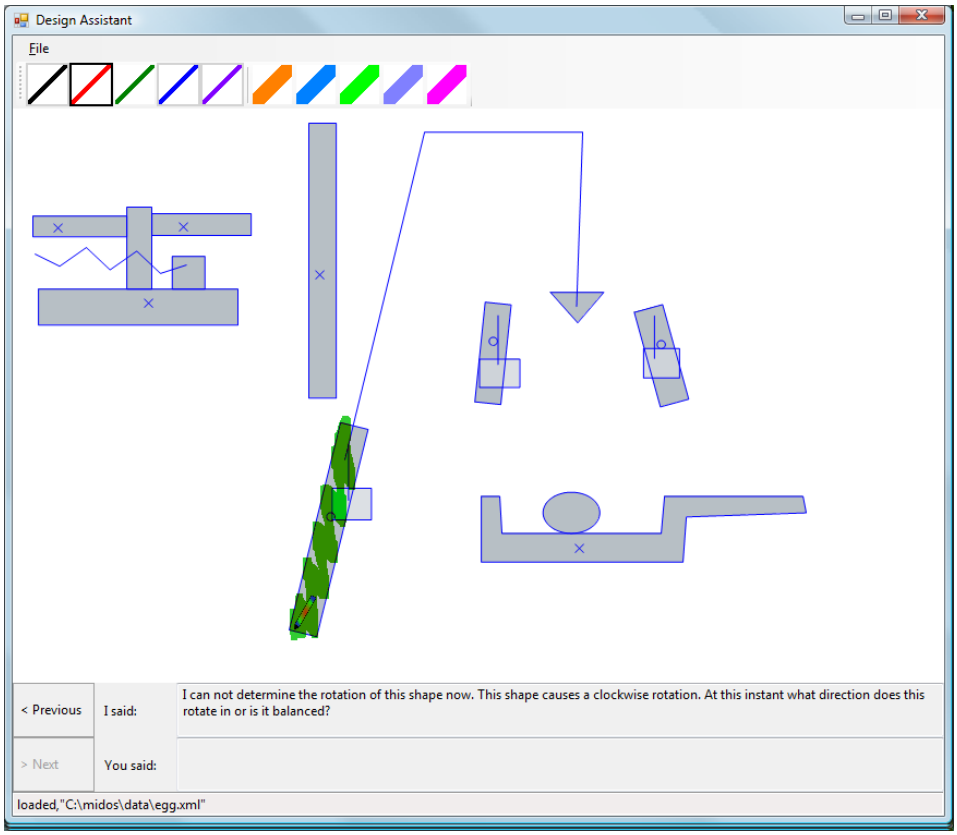

Design Assistant
File
< Previous
I said:
I can not determine the rotation of this shape now. This shape causes a clockwise rotation. This shape causes a counterclockwise rotation. So what direction does this rotate in?
> Next
You said:
counterclockwise
loaded,"C:\midos\data\egg.xml"

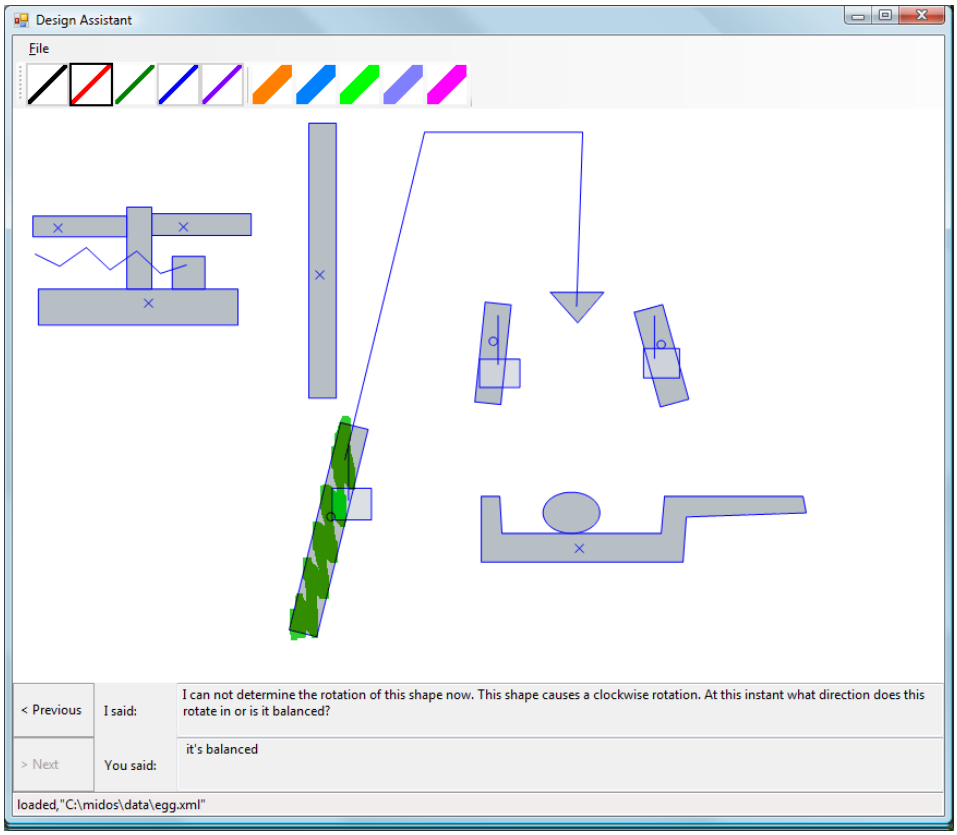

Design Assistant
File
< Previous
I said:
I can not determine the rotation of this shape now. This shape causes a clockwise rotation. This shape causes a counterclockwise rotation. So what direction does this rotate in?
> Next
You said:
counterclockwise
loaded,"C:\midos\data\egg.xml"

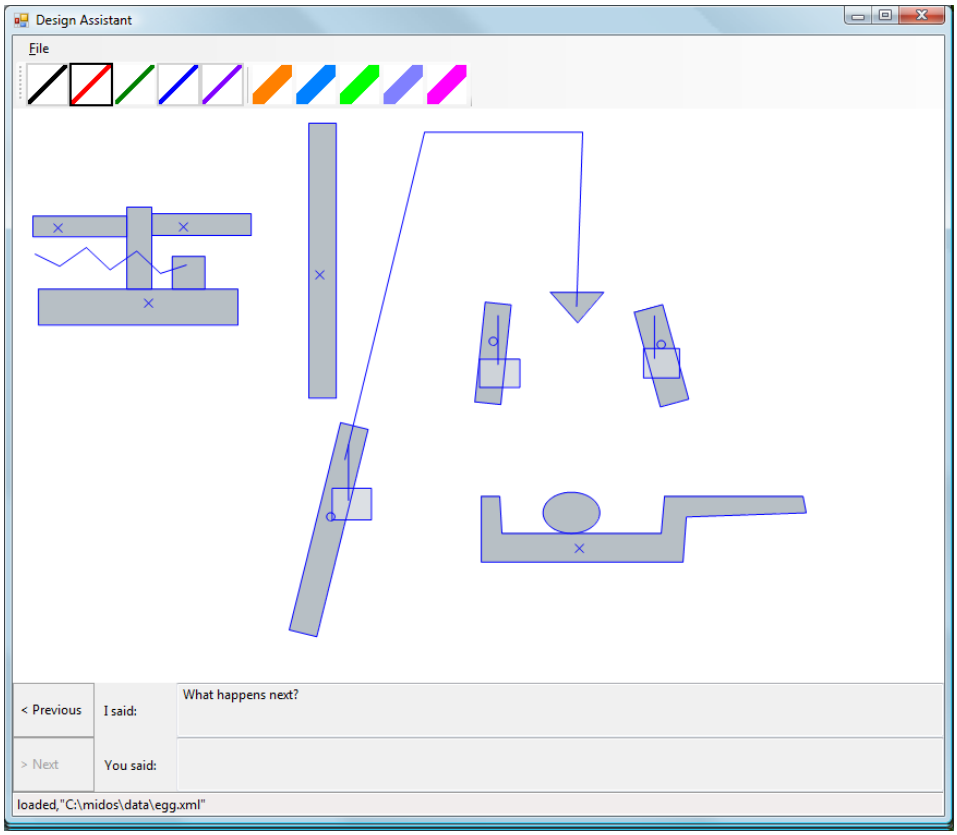

Design Assistant
File
< Previous
I said:
How far does this shape move?
> Next
You said:
loaded,"C:\midos\data\egg.xml"

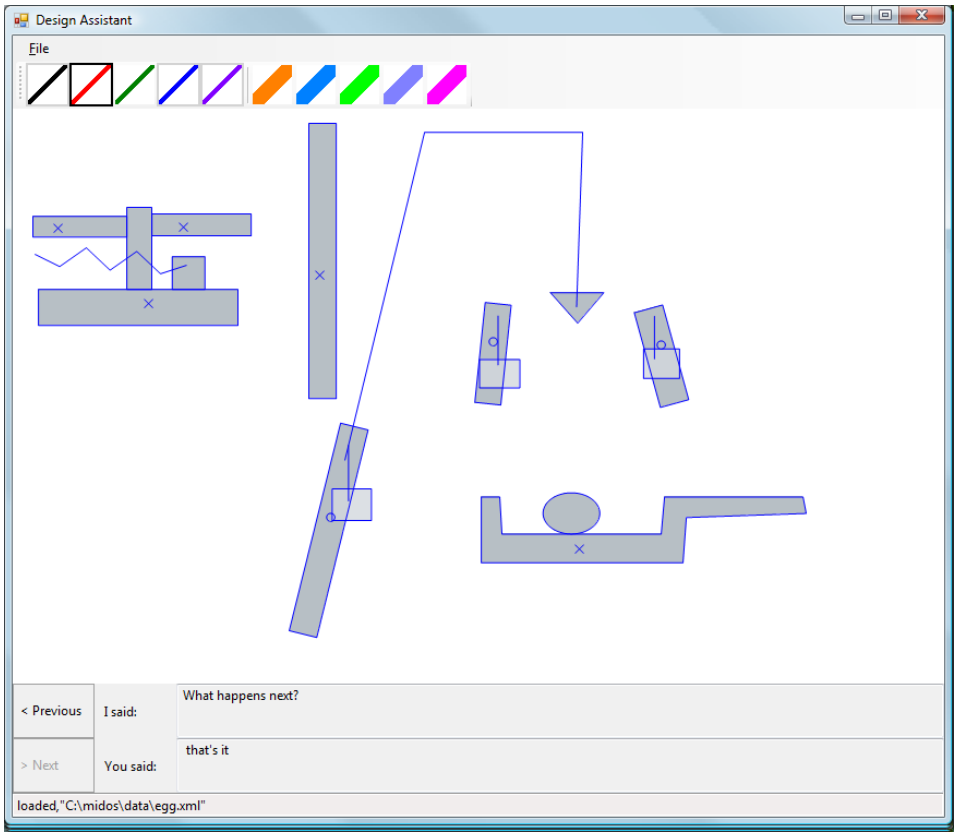

Design Assistant
File
< Previous
I said:
How far does this shape move?
> Next
You said:
goes forever
loaded,"C:\midos\data\egg.xml"


Design Assistant
File
< Previous
I said:
How far does this shape move?
> Next
You said:
goes forever
loaded,"C:\midos\data\egg.xml"


Design Assistant
File
< Previous
I said:
I can not determine the rotation of this shape now. This shape causes a clockwise rotation. At this instant what direction does this rotate in or is it balanced?
> Next
You said:
loaded,"C:\midos\data\egg.xml"


Design Assistant
File
< Previous
I said:
I can not determine the rotation of this shape now. This shape causes a clockwise rotation. At this instant what direction does this rotate in or is it balanced?
> Next
You said:
it's balanced
loaded,"C:\midos\data\egg.xml"


Design Assistant
File
< Previous
I said:
What happens next?
> Next
You said:
loaded,"C:\midos\data\egg.xml"


Design Assistant
File
< Previous
I said:
What happens next?
> Next
You said:
that's it
loaded,"C:\midos\data\egg.xml"

Bibliography

[1] Aaron Adler. Segmentation and alignment of speech and sketching in a design environment. Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, February 2003.

[2] Aaron Adler and Randall Davis. Speech and sketching for multimodal design. In *Proceedings of the 9th International Conference on Intelligent User Interfaces*, pages 214–216. ACM Press, 2004.

[3] James F. Allen, Donna K. Byron, Myroslava Dzikovska, George Ferguson, Lucian Galescu, and Amanda Stent. Towards conversational human-computer interaction. *AI Magazine*, 22(4):27–38, 2001.

[4] Christine Alvarado. A natural sketching environment: Bringing the computer into early stages of mechanical design. Master's thesis, MIT, 2000.

[5] Christine Alvarado and Randall Davis. Resolving ambiguities to create a natural sketch based interface. In *Proceedings of IJCAI-2001*, August 2001.

[6] Christine Alvarado and Randall Davis. Dynamically constructed bayes nets for multi-domain sketch understanding. In *Proceedings of IJCAI-05*, pages 1407–1412, San Francisco, California, August 1 2005.

[7] Elisabeth André, Wolfgang Finkler, Winfried Graf, Thomas Rist, Anne Schauder, and Wolfgang Wahlster. *Intelligent Multimedia Interfaces*, pages 75–93. AAAI Press, 1993.

[8] John Bateman, Jörg Kleinz, Thomas Kamps, and Klaus Reichenberger. Towards constructive text, diagram, and layout generation for information presentation. *Computational Linguistics*, 27(3):409–449, 2001.

[9] David Bischel, Thomas Stahovich, Eric Peterson, Randall Davis, and Aaron Adler. Combining speech and sketch to interpret unconstrained descriptions of mechanical devices. In *Proceedings of the 2009 International Joint Conference on Artificial Intelligence (IJCAI)*, Pasadena, California, 2009.

[10] Alexander Blessing, T. Metin Sezgin, Relja Arandjelovic, and Peter Robinson. A multimodal interface for road design. In *2009 Intelligent User Interfaces Workshop on Sketch Recognition*. ACM Press, February 2009.

[11] Richard A. Bolt. Put-that-there: Voice and gesture at the graphics interface. In *Proceedings of the 7th Annual Conference on Computer Graphics and Interactive Techniques*, pages 262–270, 1980.

[12] Justine Cassell, Catherine Pelachaud, Norman Badler, Mark Steedman, Brett Achorn, Brett Douville, Scott Prevost, and Matthew Stone. Animated conversation: Rule-based generation of facial expression, gesture and spoken intonation for multiple conversational agents. In *Proceedings of SIGGRAPH '94*, pages 413–420, 1994.

[13] Joyce Y. Chai, Pengyu Hong, and Michelle X. Zhou. A probabilistic approach to reference resolution in multimodal user interfaces. In *Proceedings of 2004 International Conference on Intelligent User Intefaces (IUI'04)*, pages 70–77, 2004.

[14] Joyce Y. Chai, Zahar Prasov, Joseph Blaim, and Rong Jin. Linguistic theories in efficient multimodal reference resolution: An empirical investigation. In *IUI '05: Proceedings of the 10th international conference on intelligent user interfaces*, pages 43–50, New York, NY, USA, January 2005. ACM Press.

[15] P. R. Cohen, M. Johnston, D. R. McGee, S. L. Oviatt, J. Pittman, I. Smith, L. Chen, and J. Clowi. QuickSet: Multimodal interaction for distributed applications. In *Proceedings of Mutimedia '97*, pages 31–40. ACM Press, 1997.

[16] Pedro Company and Peter Ashley Clifford Varley. Operating modes in actual versus virtual paper-and-pencil design scenarios. In *2009 Intelligent User Interfaces Workshop on Sketch Recognition*. ACM Press, February 2009.

[17] Nils Dahlback, Arne Jonsson, and Lars Ahrenberg. Wizard of Oz studies - why and how. *Intelligent User Interfaces (IUI93)*, pages 193–200, 1993.

[18] Mukesh Dalal, Steven Feiner, Kathleen McKeown, Shimei Pan, Michelle Zhou, Tobias Höllerer, James Shaw, Yong Feng, and Jeanne Fromer. Negotiation for automated generation of temporal multimedia presentations. In *MULTIMEDIA '96: Proceedings of the fourth ACM international conference on Multimedia*, pages 55–64. ACM, 1996.

[19] Randall Davis. Sketch understanding in design: Overview of work at the MIT AI lab. *Sketch Understanding, Papers from the 2002 AAAI Spring Symposium*, pages 24–31, March 25-27 2002.

[20] David Demirdjian, Teresa Ko, and Trevor Darrell. Untethered gesture acquisition and recognition for virtual world manipulation. *Virtual Reality*, 8(4):222–230, September 2005.

[21] Christiane Fellbaum, editor. *WordNet: an electronic lexical database*. MIT Press, 1998.

[22] Kenneth Forbus, R. Ferguson, and J. Usher. Towards a computational model of sketching. In *Intelligent User Interfaces '01*, pages 77–83, 2001.

[23] Mary Ellen Foster. Interleaved planning and output in the COMIC fission module. In *Proceedings of the ACL 2005 Workshop on Software*, Ann Arbor, June 2005.

[24] Mary Ellen Foster and Michael White. Assessing the impact of adaptive generation in the COMIC multimodal dialogue system. In *Proceedings of the IJCAI 2005 Workshop on Knowledge and Reasoning in Practical Dialogue Systems*, Edinburgh, 2005.

[25] Manuel Giuliani. Representation of speech and gestures in human-robot interaction. In *IEEE Ro-Man 2008 Workshop: Towards Natural Human-Robot Joint Action*, Munich, Germany, August 2008.

[26] James Glass, Eugene Weinstein, Scott Cyphers, Joseph Polifroni, Grace Chung, and Mikio Nakano. A framework for developing conversational user interface. In *Proceedings of the 4th International Conference on Computer-Aided Design of User Interfaces*, pages 354–365, January 2004.

[27] Genevieve Gorrell, Ian Lewin, and Manny Rayner. Adding intelligent help to mixed-initiative spoken dialogue systems. In *Proceedings of ICSLP 2002*, pages 2065–2068, 2002.

[28] Alexander Gruenstein. *Toward Widely-Available and Usable Multimodal Conversational Interfaces*. PhD thesis, Massachusetts Institute of Technology, June 2009.

[29] Tracy Hammond and Randall Davis. Tahuti: A geometrical sketch recognition system for UML class diagrams. *AAAI Spring Symposium on Sketch Understanding*, pages 59–68, March 25-27 2002.

[30] Tracy Hammond and Randall Davis. LADDER, a sketching language for user interface developers. *Elsevier, Computers and Graphics*, 28:518–532, 2005.

[31] Helen Wright Hastie, Michael Johnston, and Patrick Ehlen. Context-sensitive help for multimodal dialogue. In *Proceedings of the 6th ACM International Conference on Multimodal Interfaces ICMI 2004*, pages 93–98. ACM PRess, 2002.

[32] Timothy J. Hazen, Stephanie Seneff, and Joseph Polifroni. Recognition confidence scoring and its use in speech understanding systems. In *Computer Speech and Language*, pages 49–67, 2002.

[33] Beth Ann Hockey, Oliver Lemon, Ellen Campana, Laura Hiatt, Gregory Aist, James Hieronymus, Alexander, Gruenstein, and John Dowding. Targeted help for spoken dialogue systems: intelligent feedback improves naive users' performance. In *EACL '03: Proceedings of the tenth conference on European chapter*

of the Association for Computational Linguistics, pages 147–154, Morristown, NJ, USA, 2003. Association for Computational Linguistics.

[34] Heloise Hwawen Hse and A. Richard Newton. Recognition and beautification of multi-stroke symbols in digital ink. In *Making Pen-Based Interaction Intelligent and Natural*, pages 78–84, Menlo Park, California, October 21-24 2004. AAAI Fall Symposium.

[35] Xuedong Huang, Fileno Alleva, Hsiao-Wuen Hon, Mei-Yuh Hwang, and Ronald Rosenfeld. The SPHINX-II speech recognition system: an overview. *Computer Speech and Language*, 7(2):137–148, 1993.

[36] Michael Johnston, Patrick Ehlen, Marilyn Walker, Steve Whittaker, and Preetam Maloor. Match: An architecture for multimodal dialogue systems. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 276–383, 2002.

[37] Edward C. Kaiser. Multimodal new vocabulary recognition through speech and handwriting in a whiteboard scheduling application. In *IUI '05: Proceedings of the 10th international conference on intelligent user interfaces*, pages 51–58, New York, NY, USA, January 2005. ACM Press.

[38] Edward C. Kaiser. Using redundant speech and handwriting for learning new vocabulary and understanding abbreviations. In *ICMI '06: Proceedings of the 8th International Conference on Multimodal Interfaces*, New York, NY, USA, November 2006. ACM.

[39] Glenn A. Kramer. Using degrees of freedom analysis to solve geometric constraint systems. In *SMA '91: Proceedings of the first ACM symposium on Solid modeling foundations and CAD/CAM applications*, pages 371–378, New York, NY, USA, 1991. ACM.

[40] John McCarthy and Patrick J. Hayes. Some philosophical problems from the standpoint of artificial intelligence. In B. Meltzer and D. Michie, editors, *Machine Intelligence 4*, pages 463–502. Edinburgh University Press, 1969.

[41] David L. Mills. Internet time synchronization: the network time protocol. *IEEE Transactions on Communications*, 39:1482–1493, 1991.

[42] Brad A. Myers. *Creating user interfaces by demonstration.* Academic Press Professional, Inc., San Diego, CA, USA, 1988.

[43] Paul Nielsen. A qualitative approach to mechanical constraint. In *Proceedings of AAAI 1988*, pages 270–274, 1988.

[44] Michael Oltmans, Christine Alvarado, and Randall Davis. ETCHA sketches: Lessons learned from collecting sketch data. In *Making Pen-Based Interaction Intelligent and Natural*, pages 134–140, Menlo Park, California, October 21-24 2004. AAAI Fall Symposium.

[45] Michael Oltmans and Randall Davis. Naturally conveyed explanations of device behavior. In *Workshop on Perceptive User Interfaces*, 2001.

[46] Tom Y. Ouyang and Randall Davis. Recognition of hand drawn chemical diagrams. In *Proceedings of AAAI*, pages 846–851, 2007.

[47] Sharon Oviatt, Phil Cohen, Lizhong Wu, John Vergo, Lisbeth Duncan, Berhard Suhm, Josh Bers, Thomas Holzman, Terry Winograd, James Landay, Jim Larson, and David Ferro. Designing the user interface for multimodal speech and pen-based gesture applications: State-of-the-art systems and future research directions. In *Human Computer Interaction*, pages 263–322, August 2000.

[48] Sharon Oviatt, Philip Cohen, Martin Fong, and Michael Frank. A rapid semi-automatic simulation technique for investigating interactive speech and handwriting. In *Proceedings of the International Conference of Spoken Language Processing*, pages 1351–1354. University of Alberta, 1992.

[49] Sharon Oviatt, Antonella DeAngeli, and Karen Kuhn. Integration and synchronization of input modes during multimodal human-computer interaction. In *Conference Proceedings on Human Factors in Computing Systems*, pages 415–422. ACM Press, 1997.

[50] Charles Rich and Candace Sidner. COLLAGEN: A collaboration manager for software interface agents. *User Modeling and User-Adapter Interaction*, 8(3–4):315–350, 1998.

[51] Tevfik Metin Sezgin, Thomas Stahovich, and Randall Davis. Sketch based interfaces: Early processing for sketch understanding. In *The Proceedings of 2001 Perceptive User Interfaces Workshop (PUI'01)*, Orlando, FL, November 2001.

[52] Chang She. A natural interaction reasoning system for electronic circuit analysis in an educational setting. Master's thesis, MIT, June 2006.

[53] Candace L. Sidner, Christopher Lee, Louis-Philippe Morency, and Clifton Forlines. The effect of head-nod recognition in human-robot conversation. In *HRI '06: Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*, pages 290–296, New York, NY, USA, 2006. ACM.

[54] Ivan B. Sutherland. Sketchpad, a man-machine graphical communication system. *Proceedings of the Spring Joint Computer Conference*, pages 329–346, 1963.

[55] T.Stahovich, R. Davis, and H. Shrobe. Qualitative rigid body mechanics. *Artificial Intelligence*, 2000.

[56] David G. Ullman, Stephen Wood, and David Craig. The importance of drawing in the mechanical design process. *Computers and Graphics*, 14(2):263–274, 1990.

[57] Sy Bor Wang. A multimodal galaxy-based geographic system. Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, June 2003.