

Games for Sketch Data Collection

Gabe Johnson

Computational Design Lab
Carnegie Mellon

Ellen Yi-Luen Do

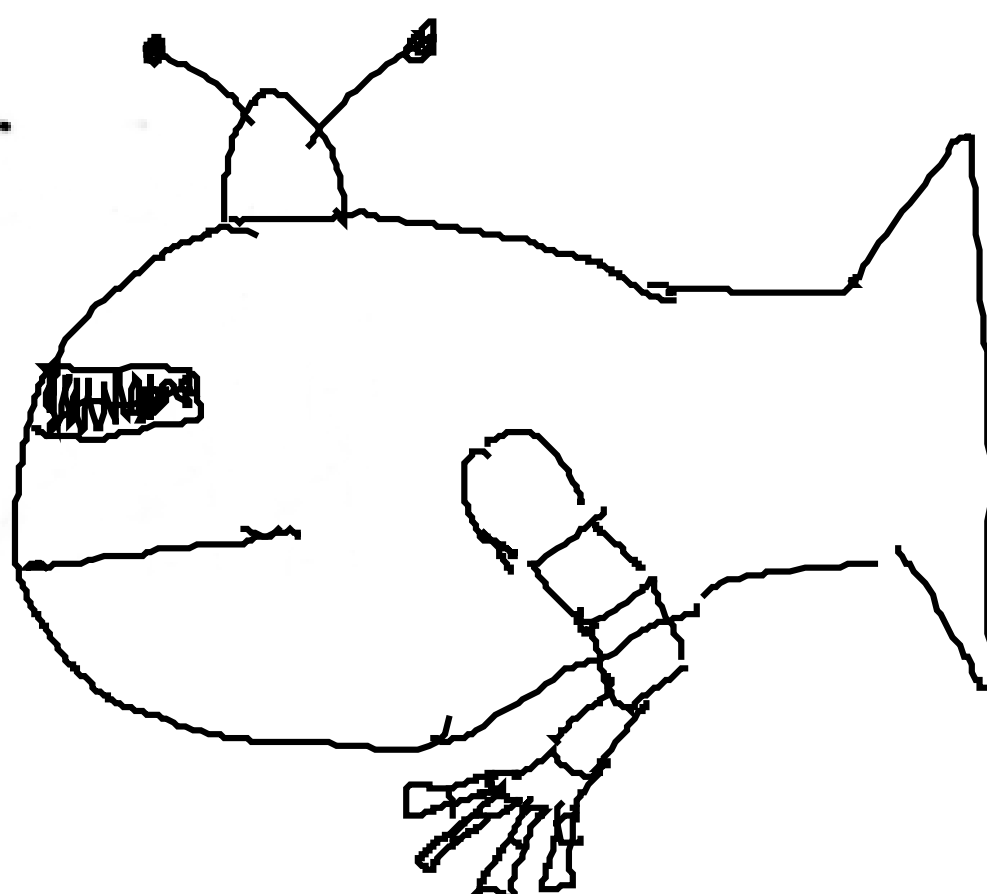
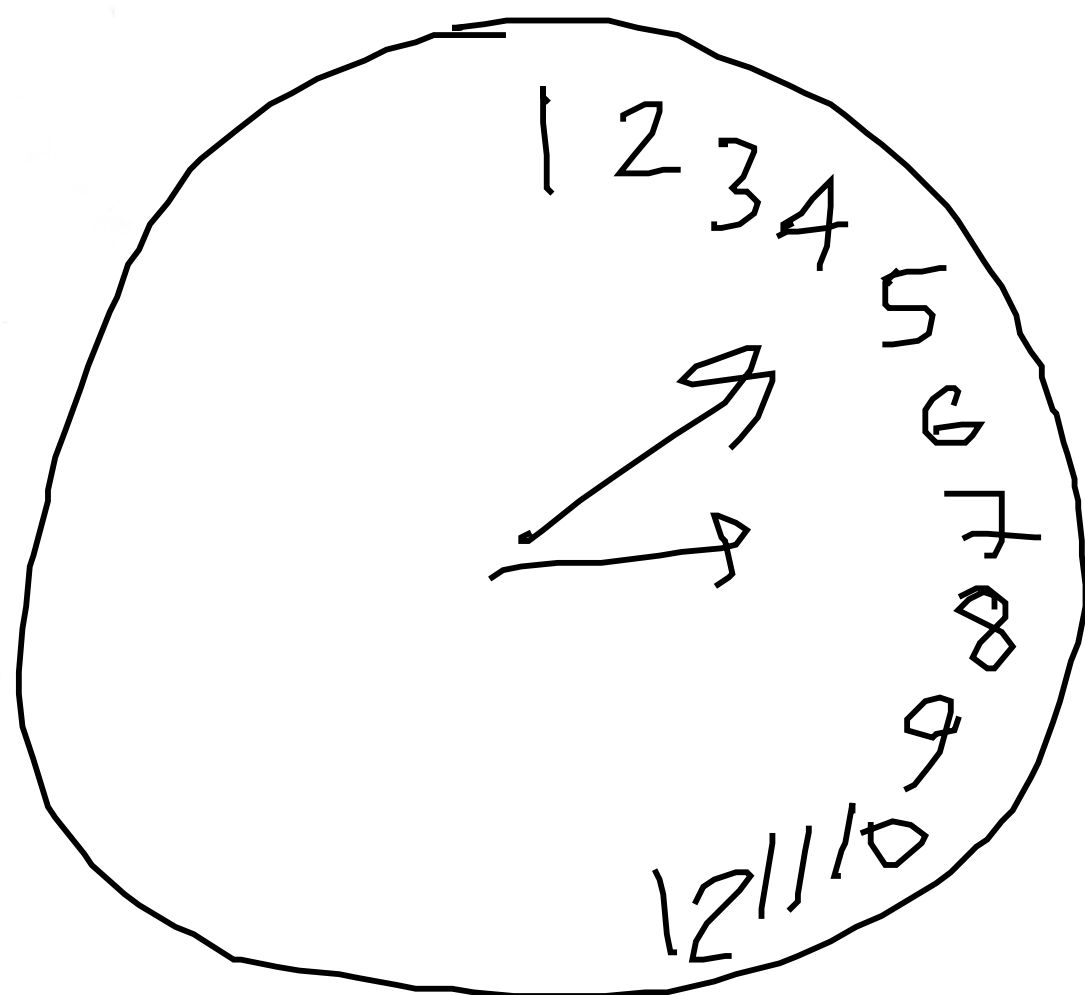
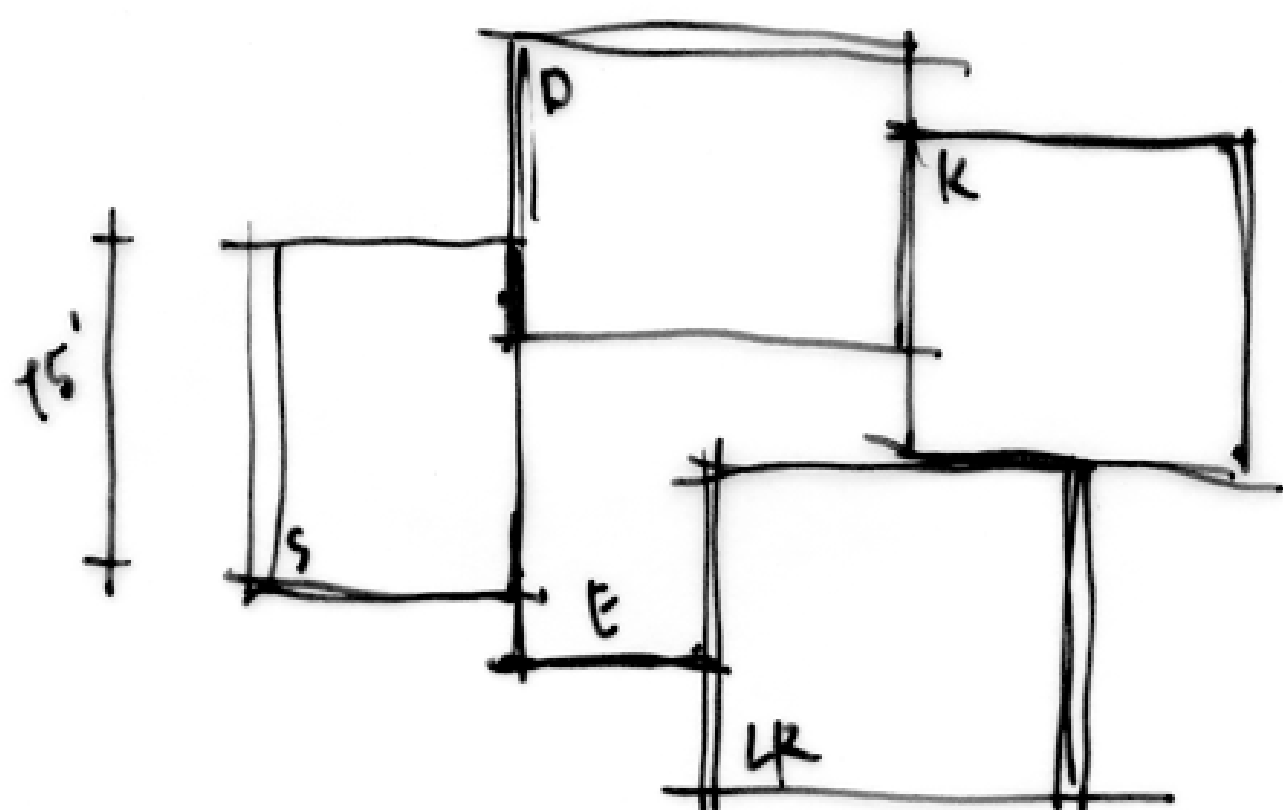
College of Architecture &
College of Computing
Georgia Tech

SBIM - August 2, 2009

1

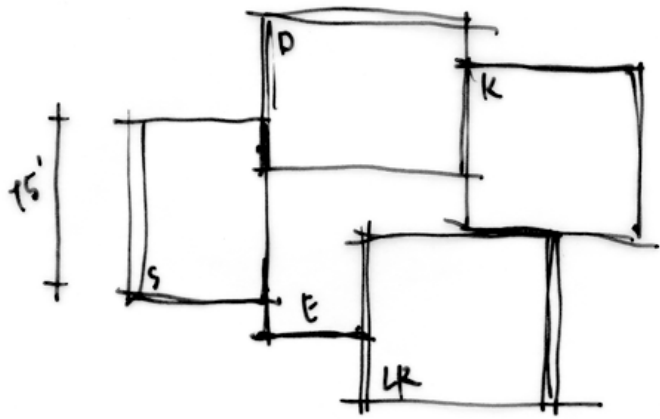
Hello. My name is Gabe Johnson, from the computational design lab at CMU. Today I'm going to present my work that explores the idea of using games to collect data on how people make and describe sketches. My advisor, Ellen Do helped shape the ideas and lent me some students for playtesting.

8

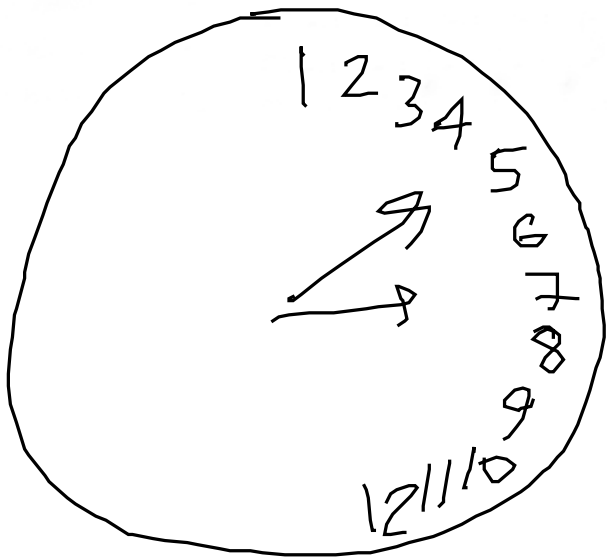


In the spirit of my topic, I want to start off the talk today with a little game. Pretend you are on the phone with a friend, and your task is to describe these pictures so your friend can draw them. They can't see these pictures, so the only way they can them is if you give them the right details. The closer their picture matches the original, the more points you both get. How would you describe them?

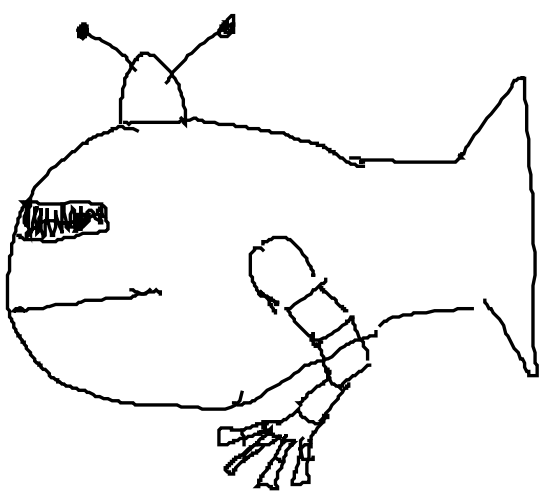
8



rectangles - floor plan -
15 feet - drawing number 8 -
dining room adjacent to
kitchen and entryway



clock - messed up clock -
6:20 - circle with numbers 1
to 12 inside the right side



robot fish - mechanical whale
- spaceship with a face, arms,
fish tail, and antenna

If you really wanted to be verbose, you could measure the coordinates of all the marks and tell your friend exactly where to draw. But from experience we know that people don't really do that. Instead, you are probably going to the drawing in higher-level terms. For example, geometric elements like rectangles or circles, or directional or perceptual qualities like numbers along the inner right side of the circle, or that two lines seem to come together at the center of a circle. Or you might refer to familiar things like kitchens, clocks, fish, or antennas. Or you might simply state what type of drawing it is, such as a floor plan layout.

Summary of this talk:

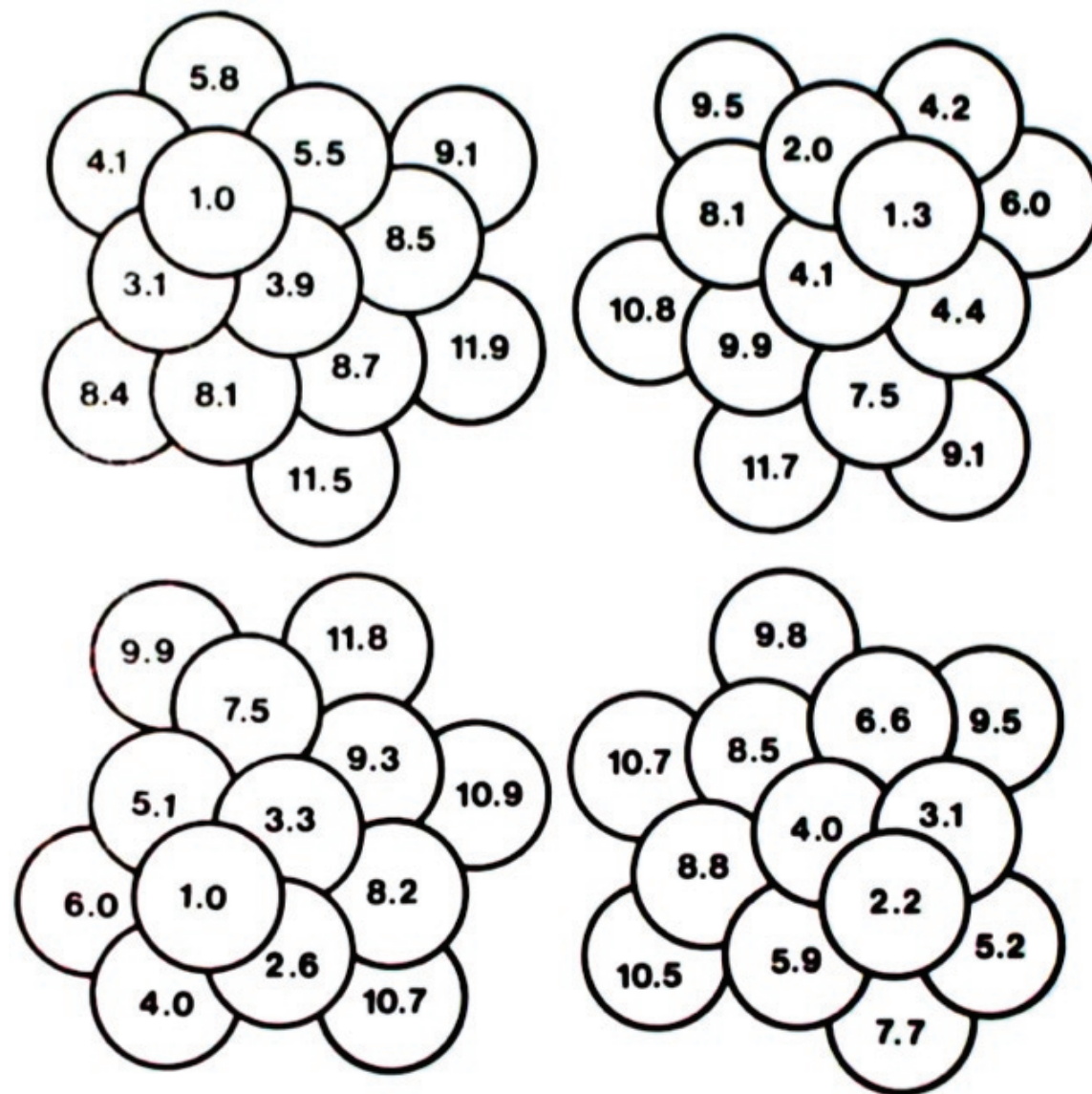
We can use games to collect information about how people make and describe sketches.

- Why collect sketch data?
- How others have approached sketch collection
- Two examples: *Picturephone* and *Stellasketch*
- This is not the final word on games+sketch

The work I am presenting today is an exploration of how we can use simple games as a way to get people to provide information about how people make drawings, and how people might describe those drawings. I'll be showing two particular web-based games called *Picturephone* and *Stellasketch*. I want to stress that the point of this work is to think about what kind of information we can gather using games, and how the format of the game itself dictates the type and quality of information collected. This is by no means a stopping point on the topic.

Why collect sketch data?

I. Mechanical, cognitive act of sketching



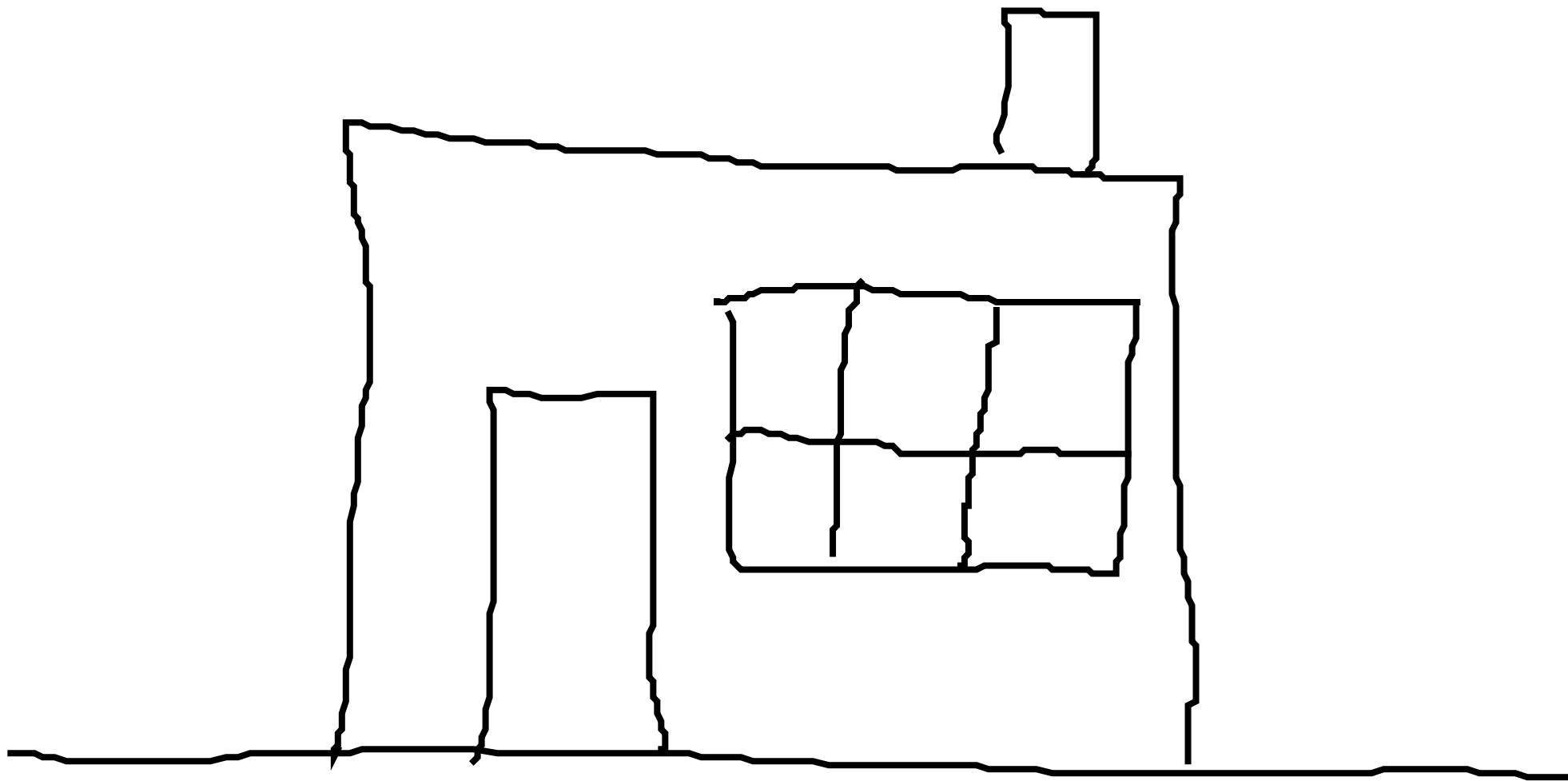
5

Whenever I show these games to other researchers, they usually say “That’s neat, but why do you want to collect all this data?” So before jumping into a discussion of the systems I’ve built, I should explain that, and I’m going to spend a little extra time on that topic.

Some people have done work on the mechanical, cognitive act of sketching. van Sommers studied this topic for many years and conducted a lot of tests wherein participants were asked to trace a given drawing, or draw something based on a description. This picture that looks like a cluster of grapes is one of the drawings van Sommers’ participants were asked to trace. The number indicates when, on average, that grape’s boundary was drawn. As you can see, people tended to focus on drawing the grape that was ‘on top’ first, and worked outward from there. That’s just one strategy people tend to use when drawing. This work, and similar work, has in turn informed the design of some sketch recognizers.

Why collect sketch data?

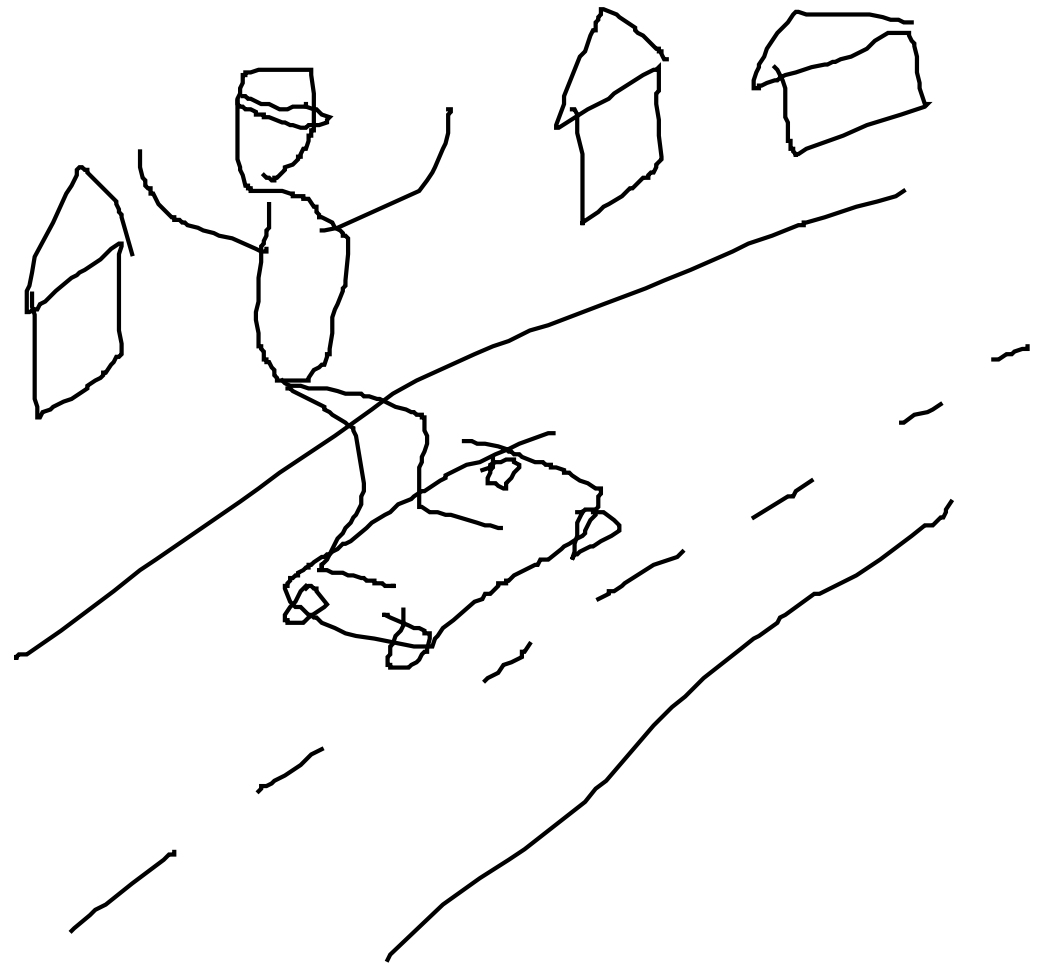
2. Which elements are drawn, how they relate



6

Another reason to collect data about sketches is to analyze the components of a drawing and how they relate. For example, if asked to draw a house, many people will include rectangular windows and doors. Many popular sketch understanding algorithms leverage contextual clues: if you can confidently recognize one element, that knowledge might be enough to suggest the identity of other elements. So in this drawing, you just have a bunch of rectangles, but we can probably agree this picture is of a house by looking at where the rectangles are in relation to one another. We know something about houses, doors, and windows, and that knowledge---those domain rules---inform how we interpret the drawing. Currently, sketch recognition systems that incorporate domain rules like that are usually be programmed by hand. And that is based on the programmer's knowledge about how people make drawings, which is probably fairly biased or incomplete.

Why collect sketch data?



3. Train or test segmentation techniques

7

People can look at sketches such as these and identify elements such as a person, a skateboard, a parked car, the road, and houses. Our brains are really good at turning this visual input into semantically meaningful things, so we see the whole picture, and whole items within the picture, and we might not consciously think about individual lines or curves and what they go with. But for many sketch recognition techniques to work, the drawing has to be segmented before the system can identify individual parts, which is what our brains do so well. When developing segmenters, it should be helpful to have access to a wide variety of drawings for testing.

Why collect sketch data?

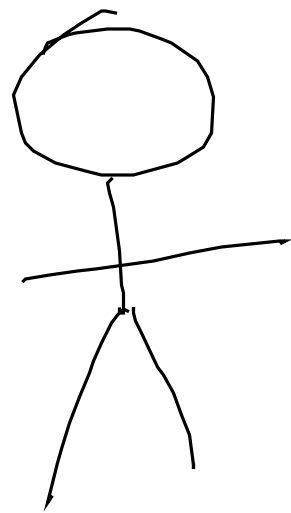


4. Train or test recognizers and rectifiers

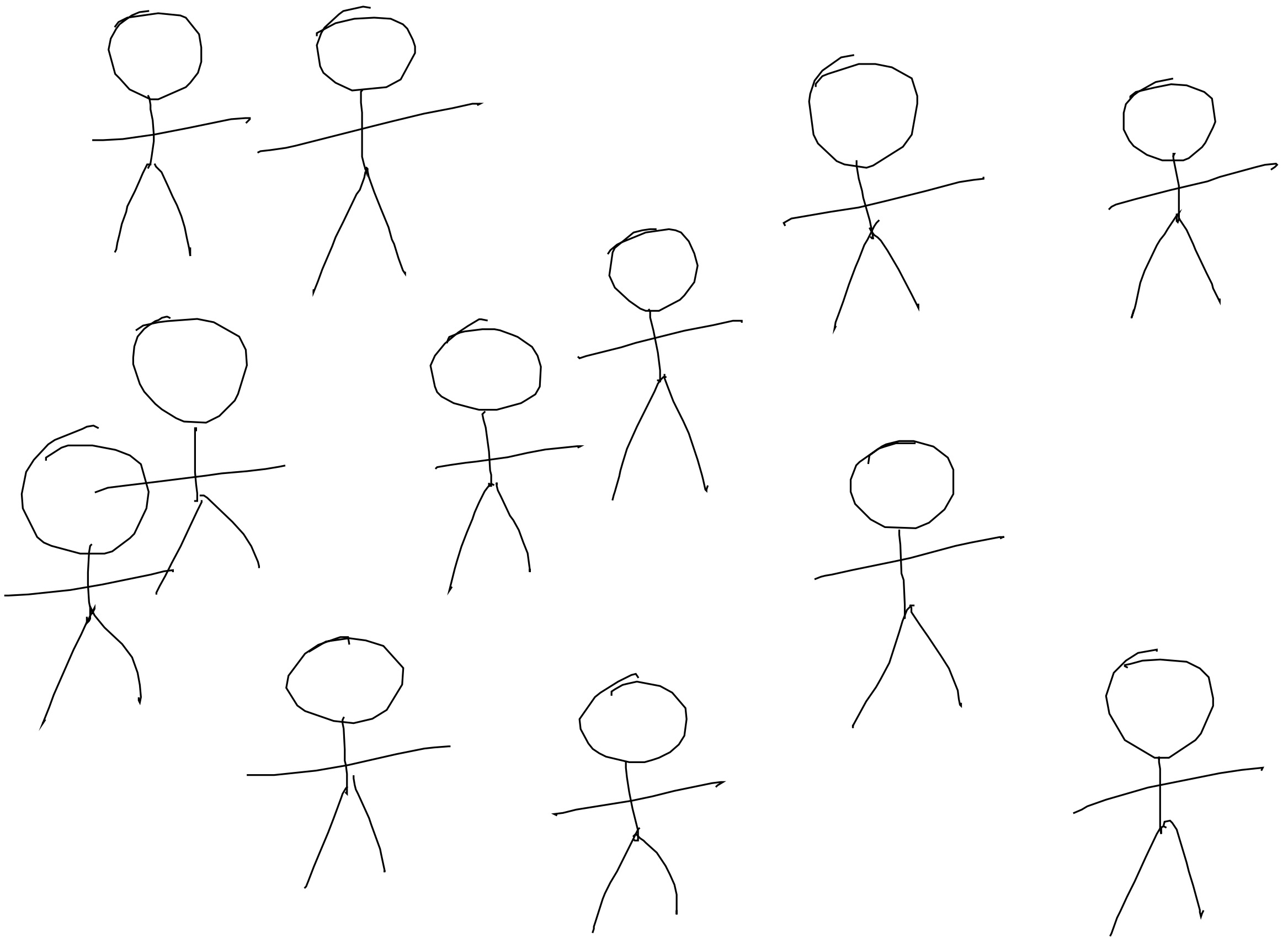
8

A sketch based system might include recognition---finding clocks, arrows, stick figures and the like. If your system should recognize houses, you could use one of the sketches from the previous slide and (using a segmentation or labeling tool) pull out the three houses, and use them for training or testing. Instead of recognition, a system might simply involve beautifying drawings by making straight lines straight, smoothing out curves, or making similar looking objects identical. Or the system might do both, by first recognizing elements, then rectifying them. This was a main reason why I built these games, and I'm using the data I have collected to test the sketch based systems I'm working on right now.

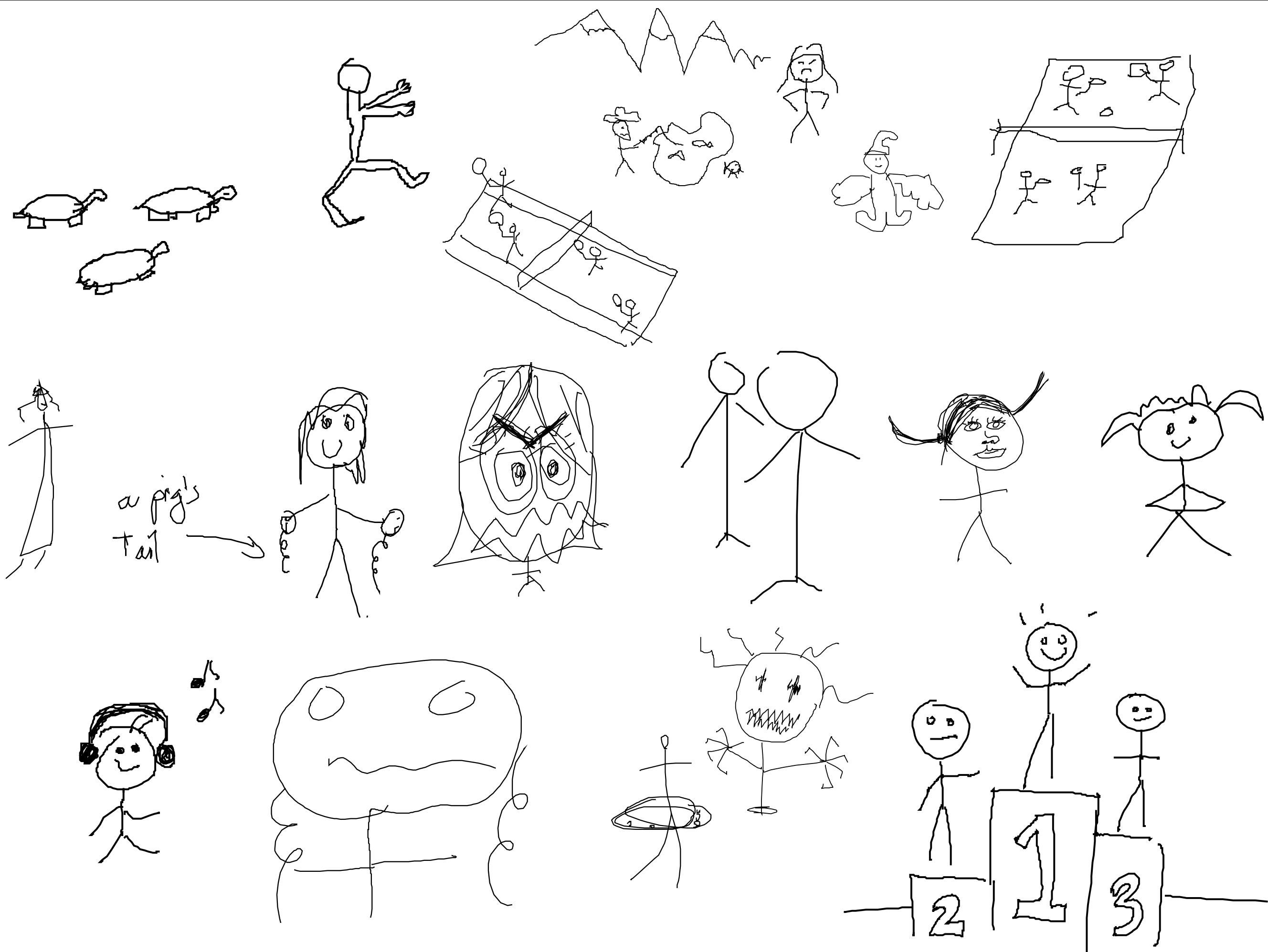
Why collect data from so many people?



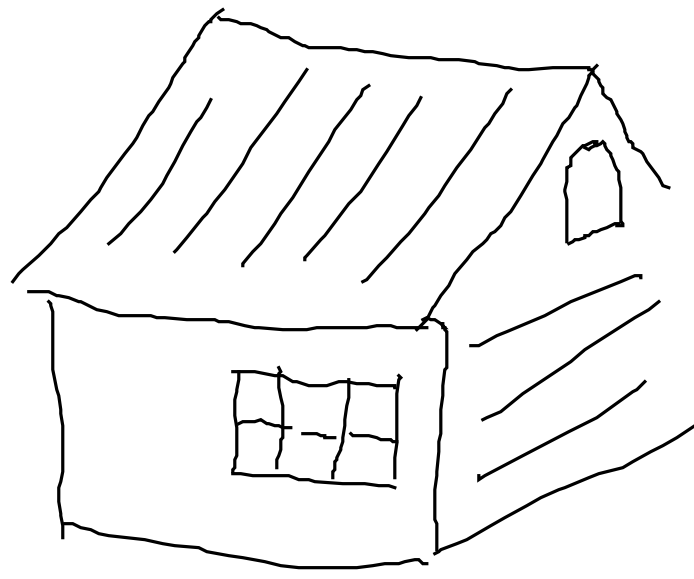
So why should we collect data from so many people? It is easy to build a sketch recognizer that works just great when you test it with your own drawings. If you've only tested with your own input, it is fairly certain that it will completely fail when somebody else picks up the pen. So we ask a few other people to give sample input. One common way of doing this is to draw N copies of the same element. But when asked to draw many copies of something, people make the drawing the same way, every time.



But in context of a real sketching situation, figures representing people are often not drawn like this at all, especially if the person drawing intends to convey something special about the figure.



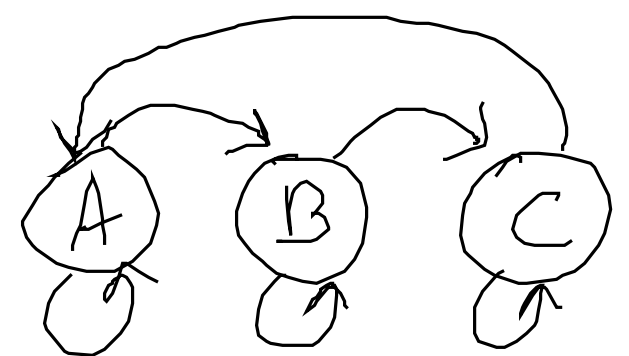
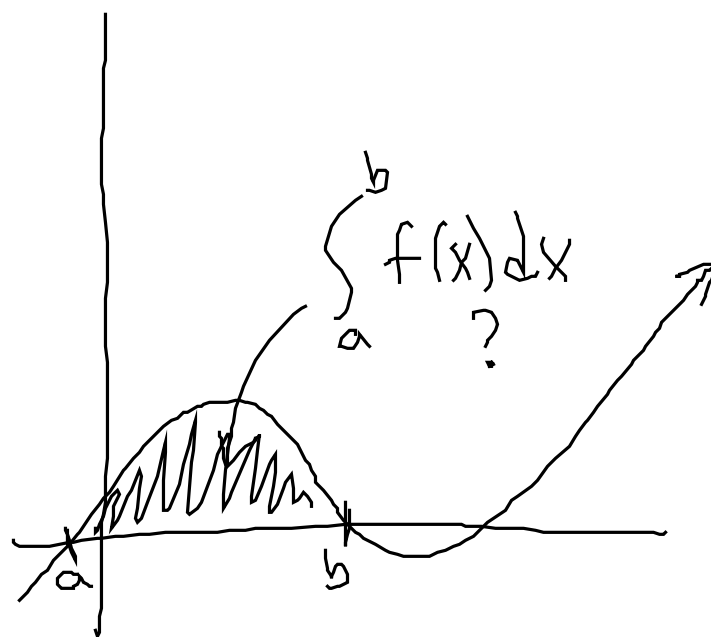
At the same time, recognizing diagram languages is only one part of the broader field of sketch based modeling tools.



Ideally we'd like to recognize aspects of all of these things, even if it is just geometric or perceptual constraints.

Things to do -

- write proposal
- propose
- write dissert -
- defend



We might bring the power of computation to support freehand sketching in lots of ways beyond recognizing diagrams, like the finite state machine in the bottom corner. For example, it is probably a little crazy at this point to create a recognizer that knows about aliens in spaceships. I'm not sure how useful that would be. But there are aspects of that drawing that we might benefit from recognizing. In that drawing, there are geometric and perceptual qualities like symmetry and co-linearity of the circles. If you are interested in building systems that recognize those properties, it makes a whole lot of sense to use a corpus of sketching data, because as I mentioned it is easy to delude yourself into thinking your algorithms are better than they really are if you are only testing on your own data, or sketches made by your labmates.

Related Work: Human Computation

- People teaching computers about the world: Cyc, Open Mind Commons
- Using computer games as bait:
 - OMC's 20 Questions
 - von Ahn's ESP game (& Google Image Labeler)

14

So, that's why I think it is important to collect sketch data. Now I'm going to cover some related work both in human computation, and on other games that involve sketching.

Over the years there have been a number of efforts to develop intelligent systems that leverage the natural intelligence of humans. Over the past few years people have started calling this "Human Computation". Doug Lenat's Cyc system is a prime example of this, which consists of a large library of true statements that were typed in by humans. For example, Cyc may contain the statement "A dog is an animal", and "All animals must eat", from which other true things can be derived, like "All dogs must eat". But it is time consuming to teach computers in this way, and you might end up with a system that is biased because only a few people were involved in teaching the system new rules. So others have looked at ways to make the process more appealing to the humans who have to do the work in order to gather more data from more people. One way of doing this is to turn the teaching process into a game.

For example, some people at MIT have made a 20 questions game where the computer tries to guess the object you are thinking about by asking simple yes/no questions of the player. If the computer can not correctly guess the object, it asks the user to reveal what it was, and it incorporates that information into its knowledge base. Another example is von Ahn and Dabbish's ESP Game. It is probably the most well known in this category.

I'll give an example of the ESP Game for those who haven't seen it. This is a video of the Google Image Labeler, which is basically the same thing as the ESP game. It works by forming pairs of human players who don't know each other. Both players are shown the same picture, and must type in labels for that picture. When the two players agree, the players get points. This is a really effective way to teach computers what is in a photograph. After enough people have identified the obvious things, it lists those words as off-limits, which forces you to be more creative in your labeling. I know I've killed a lot of valuable time doing this.

You and a **guest** scored **360** points.
Your cumulative score is **12740**.

[Start Again](#)

Today's Top Pairs

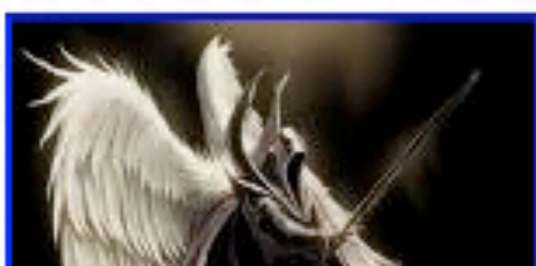
1. guest - DiecastHobbyUSA	460
2. guest - guest	400
3. Gabe - guest	360
4. cbelle - guest	330
5. guest - guest	290

All-time Top Co

1. drive big
2. PS
3. Zip
4. FD
5. MW

Thanks for your contribution. It will help us improve the relevance of image search results so that you and other Google users can labeled and the sites they came from, click on any of the images below.

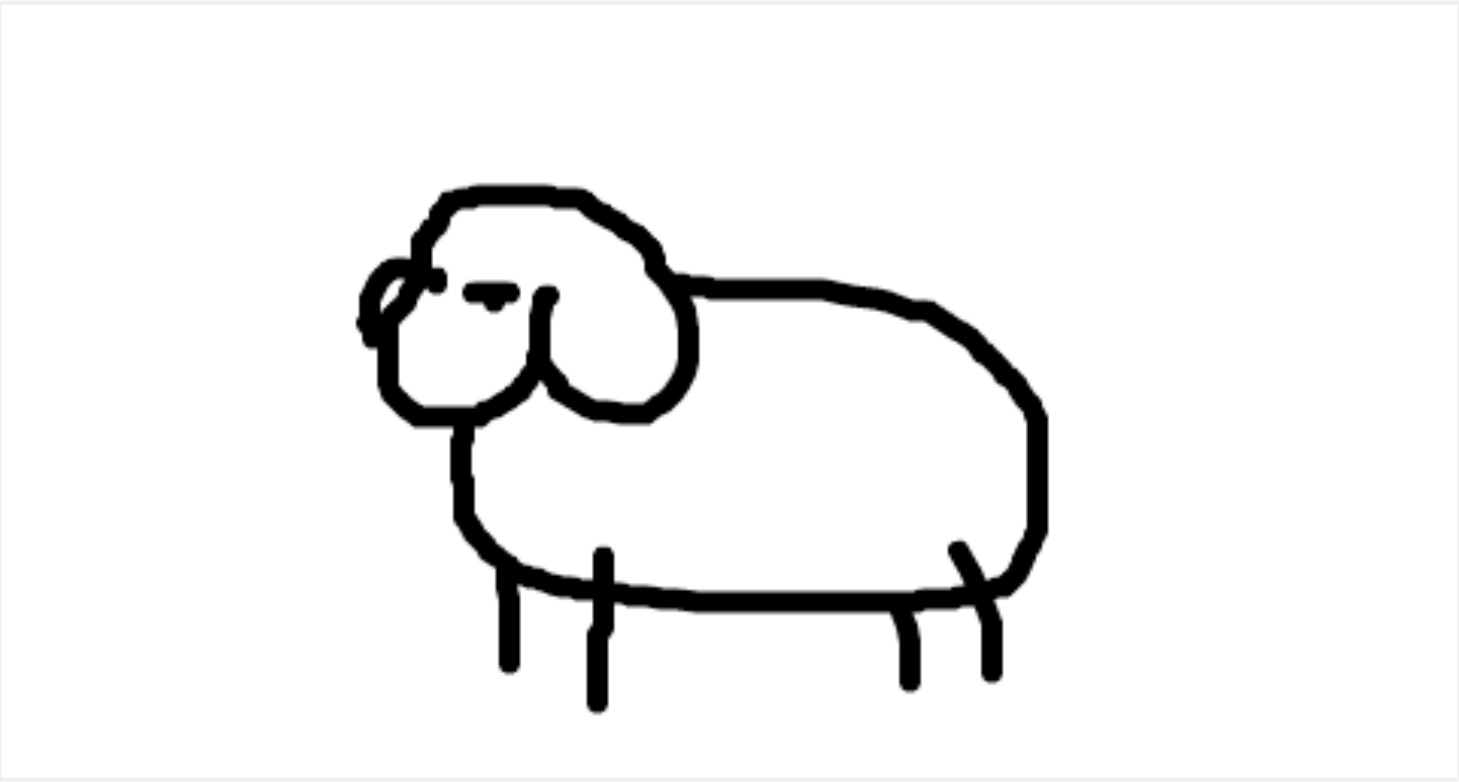
Images labeled - [Click on any image below to find out more](#)



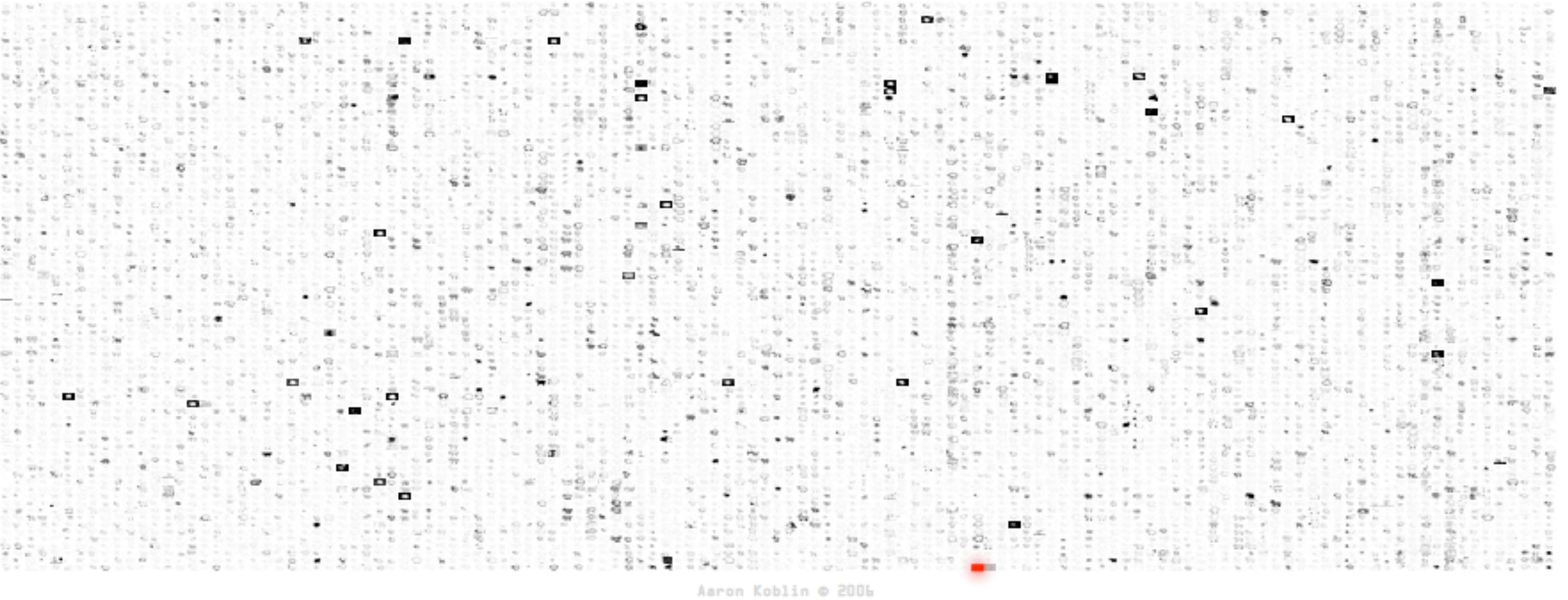
I'll give an example of the ESP Game for those who haven't seen it. This is a video of the Google Image Labeler, which is basically the same thing as the ESP game. It works by forming pairs of human players who don't know each other. Both players are shown the same picture, and must type in labels for that picture. When the two players agree, the players get points. This is a really effective way to teach computers what is in a photograph. After enough people have identified the obvious things, it lists those words as off-limits, which forces you to be more creative in your labeling. I know I've killed a lot of valuable time doing this.

THE SHEEP MARKET

10,000 sheep created
by online workers.
More...



6320/10000



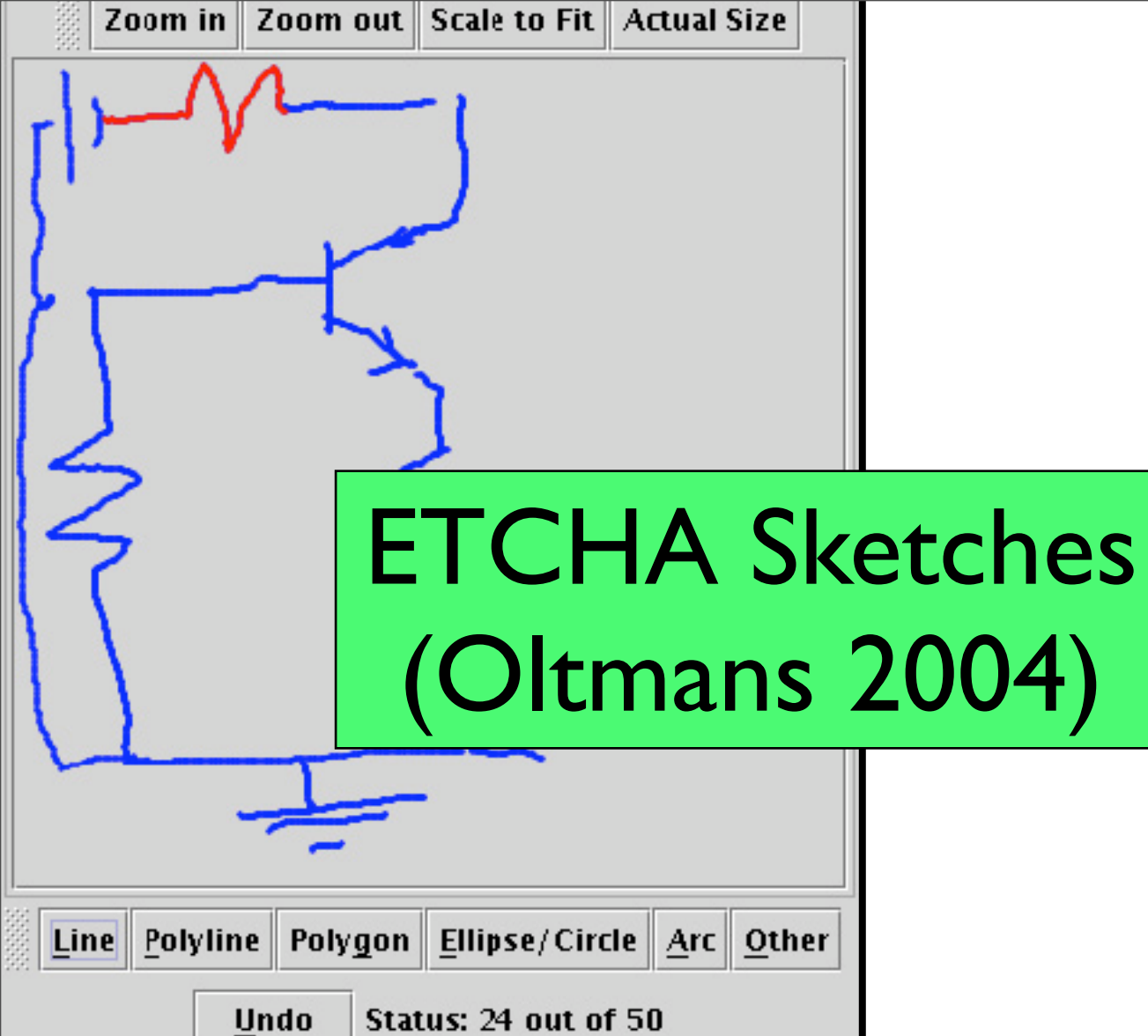
Aaron Koblin © 2006

Both the 20 questions and ESP game gather textual information from players. But there’s also some effort to collect drawings as well. Here’s a crazy art project called the Sheep Market that uses Amazon’s Mechanical Turk to pay people a few pennies to make drawings. In this case, they collected ten thousand drawings of “a sheep facing left”. While this might seem like a trivial use of human labor, the results are interesting because it shows a huge variety of ways such a simple thing can be drawn, and what extra elements are given, such as cartoon speech bubbles that let the sheep say something. But even given the variety, there still remains something invariant about most of these drawings that lets a human observer identify them (or at least, portions of them) as depicting sheep.

Related Work: Collecting Sketch Data

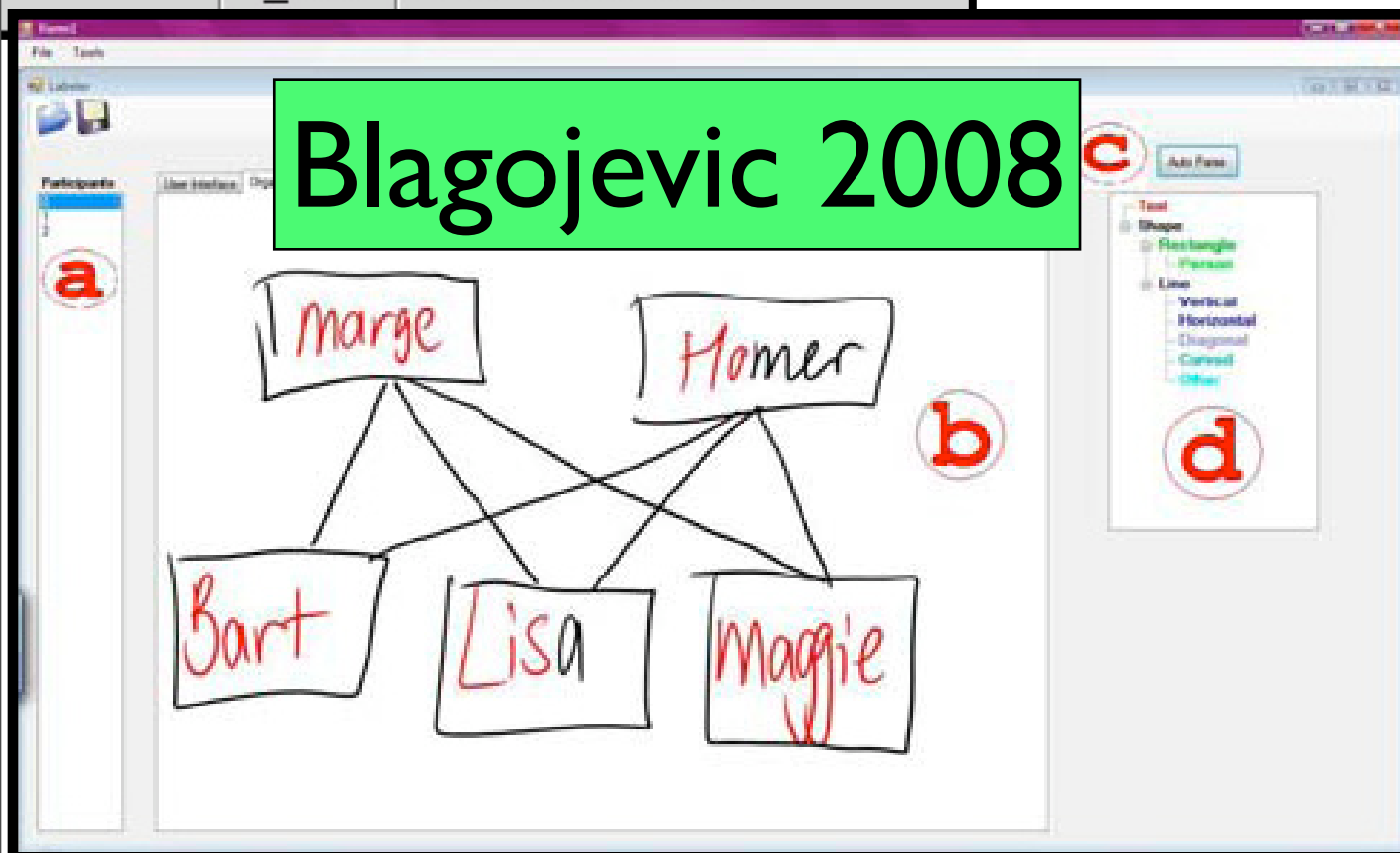
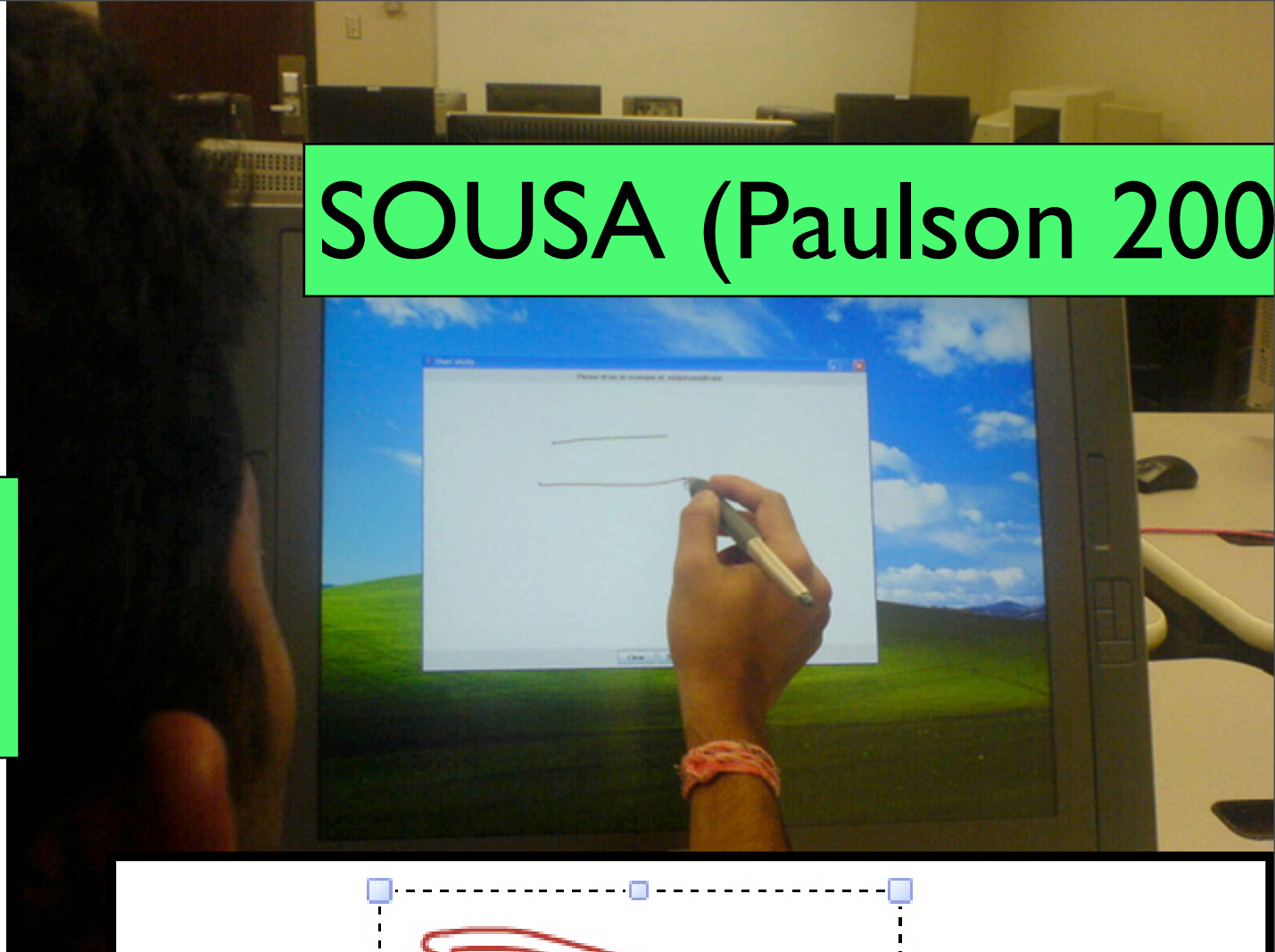
- Oltmans 2004: ETCHA Sketches domain-centric corpus
- Wolin 2007
- Paulson 2008: SOUSA
- Blagojevic 2008

Many areas of machine learning have well-known data sets that researchers around the world can use to train, test, or validate their work. For sketch-based systems, there are a few examples.

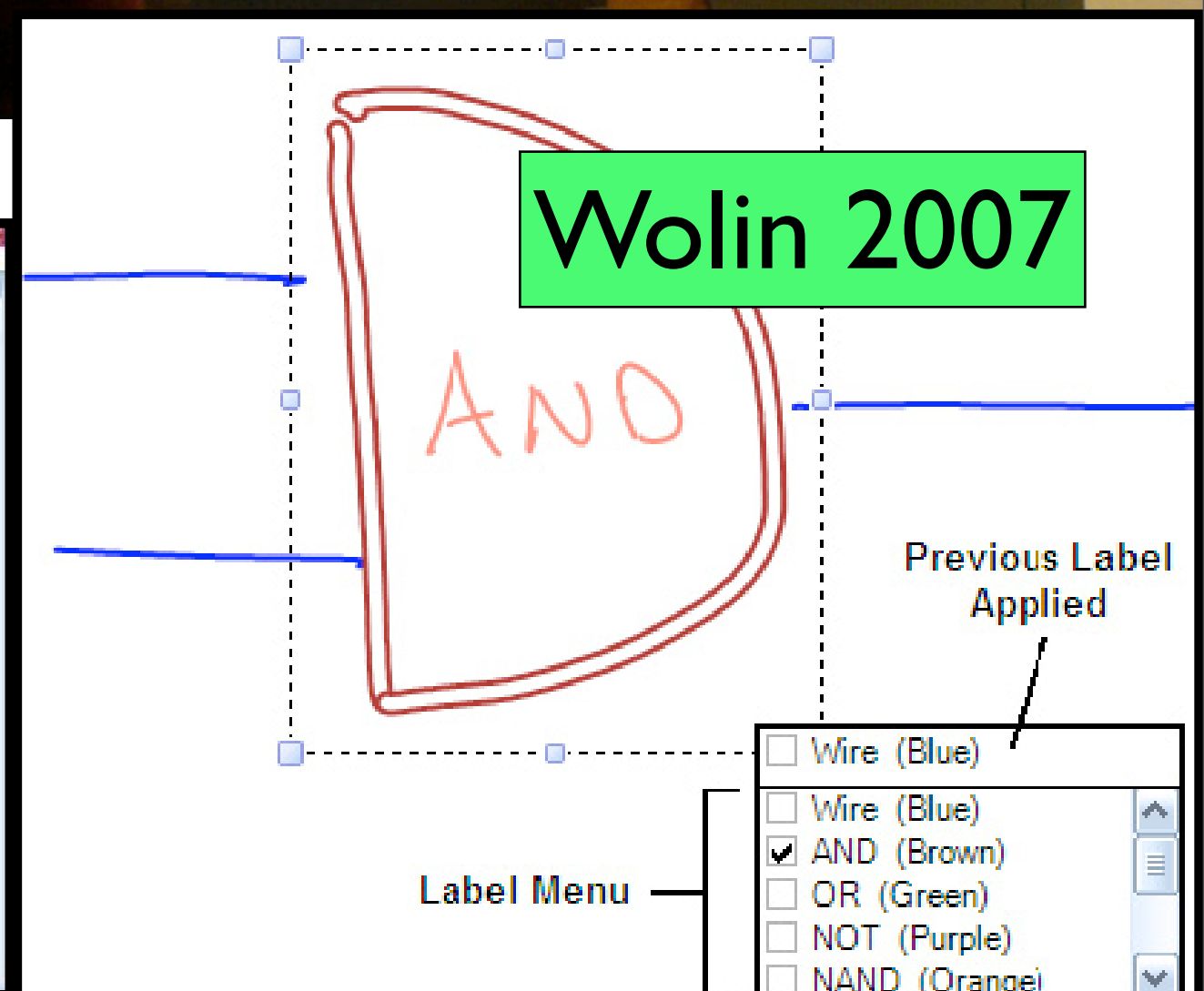


ETCHA Sketches (Oltmans 2004)

SOUSA (Paulson 2000)



Blagojevic 2008



Wolin 2007

The most widely cited data corpus is the ETCHA Sketches set that contains drawings made in diagrammatic domains like circuit layout or family trees. More recent efforts, and many have been presented at this conference in previous years, have aimed to make it easier for researchers to collect and label diagrammatic sketch data in other domains. A primary difference between these systems and the work I am presenting today is the level of fidelity. The systems mentioned on this slide are for collecting domain-specific data that is meticulously processed afterward, with each region of digital ink having been labeled according to its function. This makes the data suitable for direct use in training recognizers.

Related Work: Sketching Games

- Parlor games
 - Pictionary
 - Telephone Pictionary

Now, on to the topic of games. There have been parlor games that involve drawing for many years. You might be familiar with Pictionary, where one person is secretly given a description of something, and their team must guess what it is they are attempting to draw before a time limit expires. There is a similar game that has become popular in recent years, though it doesn't seem to have a commercial name. One common name for it is Telephone Pictionary. The first player is given a textual description of something, and they must draw it the best they can and give the sheet of paper to the next player. That player only sees the drawing, but not the description. The second player then writes down a text description of the drawing, and gives that description to the third player, who draws it, and so on. By the time the paper gets back to the first player, the content has changed dramatically and usually humorously.

Related Work: Sketching Games

- Internet games
 - iSketch
 - Draw My Thing
 - ... many others

There are of course online drawing games, and you can easily find a dozen of them on the Internet without looking very hard. The games I looked at, like iSketch, are geared toward entertainment for sure, but it seems the data is not saved or otherwise available after a game is done. Further, the topics used in these games are intended to be particularly challenging to depict. For example, a player may be asked to draw abstract concepts such as ‘frustration’, which are difficult to draw.

Picturephone and Stellasketch

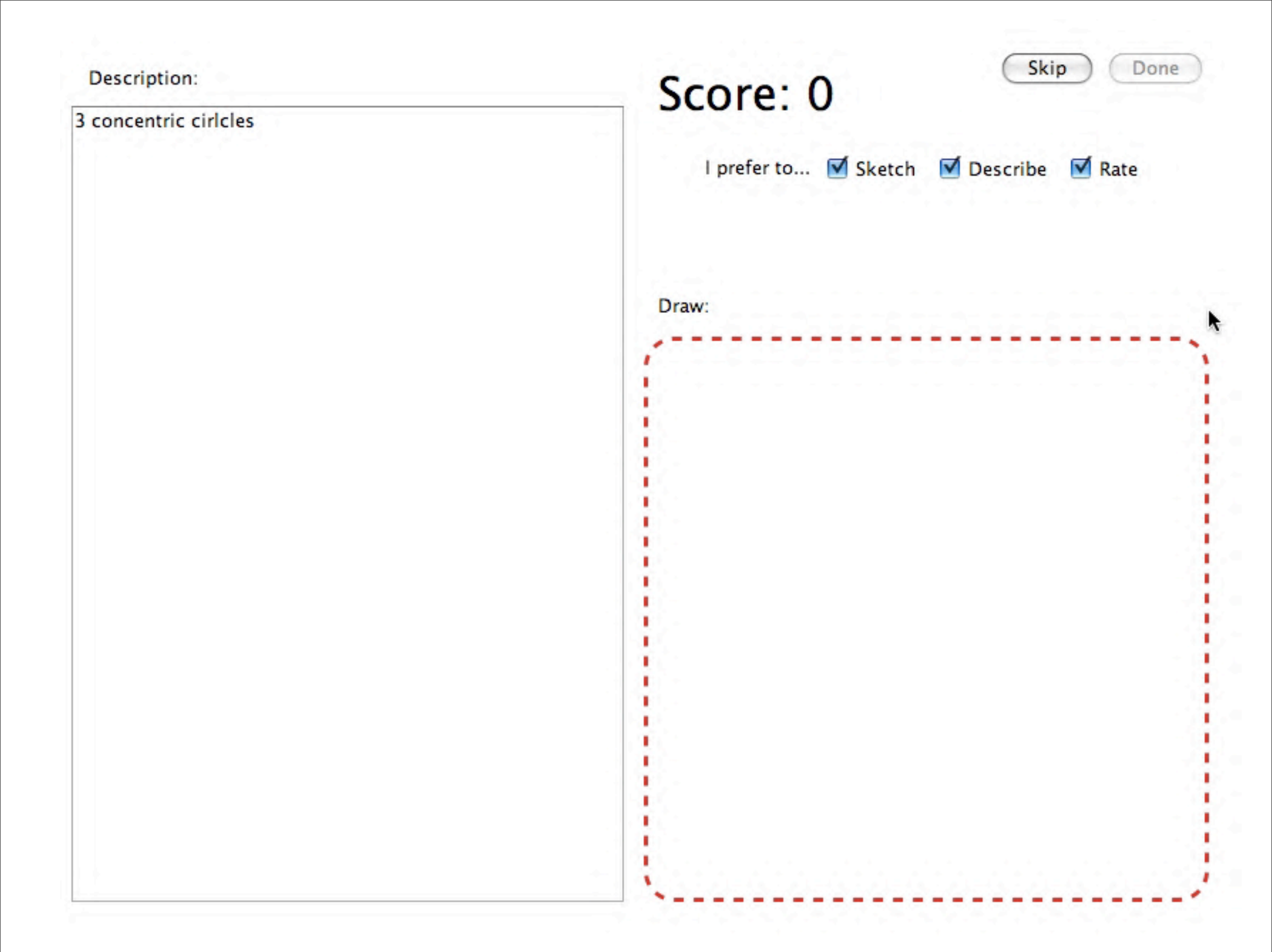
- Make it entertaining for people to provide drawings and descriptions
- Java applets coordinated via Java web server
- Each tested by ~20 players for an hour
- All drawings may be browsed and downloaded in PNG, PDF, raw data

Today I'm presenting two online, multi-player sketching games designed to collect data about how people make drawings and how those drawings might be described. The games were both implemented as Java applets, which are coordinated by a web-based server application, which was also written in Java. Ellen and I conducted playtesting of each game using about 20 students, with each test lasting about one hour. In addition to gathering quite a bit of sketch data and labels, we also got a lot of feedback about what was good and not so good about the games. All of the data collected by the games so far is available on the game sites in various formats including a raw text format that is usable by other sketching systems.

Picturephone

22

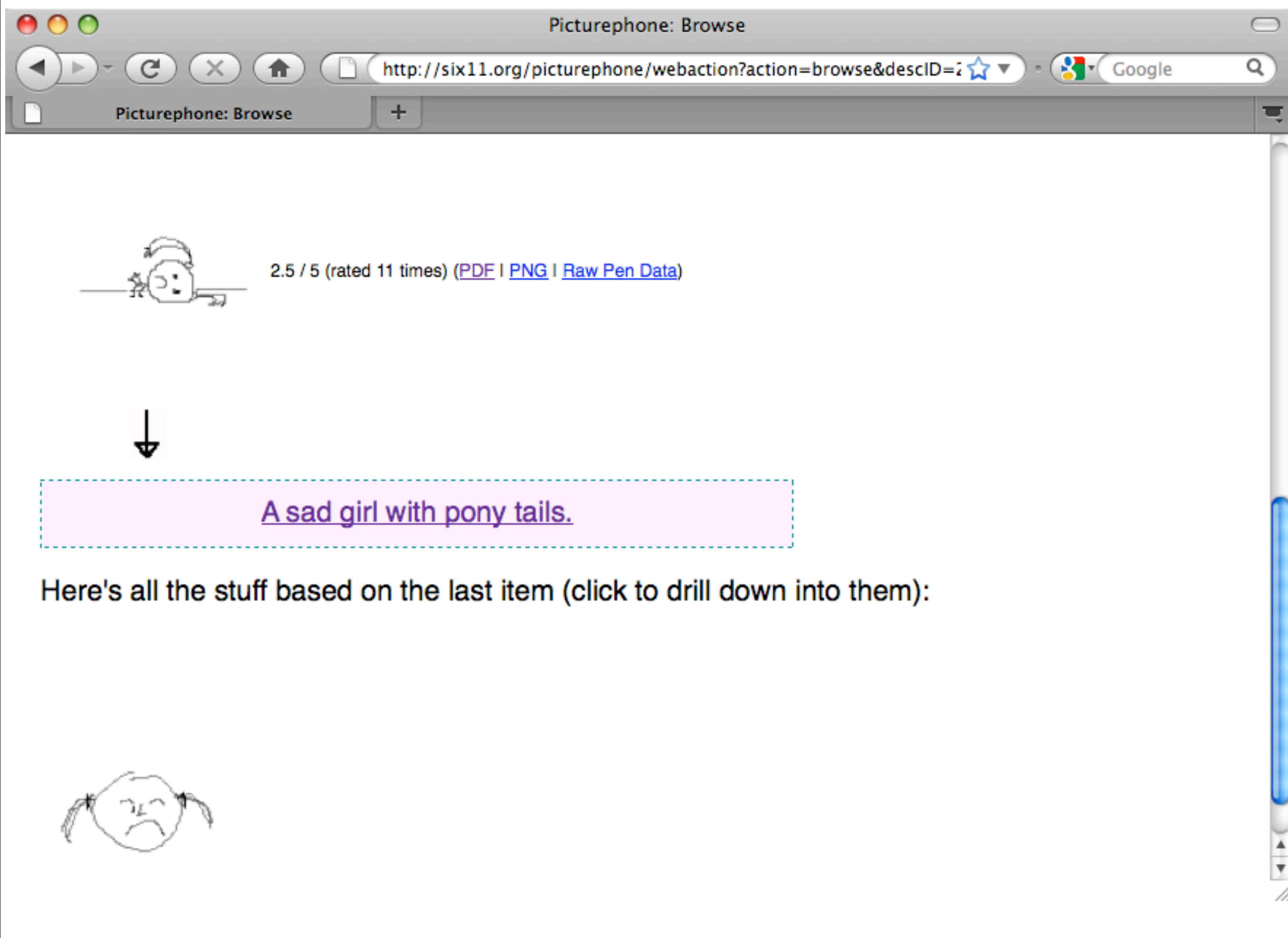
Here is a short video of somebody playing Picturephone. Here it asks the player to draw using the description “three concentric circles”. After completing the drawing, the player hits ‘Done’. The game chooses among your preferred modes: sketching, describing, or rating. The user gets a few ‘rate’ phases in a row here. It simply asks the player to rate how closely the two pictures match. The picture on the left was the basis for a description, and the one on the right was a sketch made based on that description. When a player rates these two drawings, points are assigned to the people who made both sketches as well as whoever made the mediating description. As you can see, the task of rating is rather subjective, but since it collects lots of ratings for the same pair it seems to work out fairly well. Eventually the game picks ‘describe’ mode, so here you see the player typing in a description about a floor plan layout of a square house with a bathroom and kitchen in the corners. This process continues until the player chooses to end their session.



Here is a short video of somebody playing Picturephone. Here it asks the player to draw using the description “three concentric circles”. After completing the drawing, the player hits ‘Done’. The game chooses among your preferred modes: sketching, describing, or rating. The user gets a few ‘rate’ phases in a row here. It simply asks the player to rate how closely the two pictures match. The picture on the left was the basis for a description, and the one on the right was a sketch made based on that description. When a player rates these two drawings, points are assigned to the people who made both sketches as well as whoever made the mediating description. As you can see, the task of rating is rather subjective, but since it collects lots of ratings for the same pair it seems to work out fairly well. Eventually the game picks ‘describe’ mode, so here you see the player typing in a description about a floor plan layout of a square house with a bathroom and kitchen in the corners. This process continues until the player chooses to end their session.



One of the fun aspects of both these games is the browsing feature, where you can see the drawings other players have made.



If you click on a picture it takes you to this screen, that shows you the description on which that picture was based, and all the drawings and descriptions that came before. You can see how well the sketches are rated. Importantly, it also allows you to download the sketch in two image formats as well as a raw data format for use in other sketching environments.

Stellasketch - Choose Theme

Pick a theme!

Animals (0 votes)

Athletics (0 votes)

Bacon! (0 votes)

Big Things (0 votes)

Household objects (0 votes)

Suprise me (0 votes)

Gabe has joined the Lobby
People in this room: askdelia, Gabe
ElBlarko has joined the Lobby
Gabe has left the Lobby (gone to Romper Room)
Gabe has joined Romper Room
People in this room: Gabe, ElBlarko

Now, on to Stellasketch, the second game. To play Stellasketch you need to have at least two players in the same game room. At the beginning of a game, all players can vote for a theme. All the clues in the rest of that game will be taken from that theme. This helps give players a little bit of context. Once a theme is chosen the game begins. Each game has three rounds, where one player is randomly chosen to draw, and all other players must guess what they think the drawing is of.

Stellasketch - Sketch Phase

When a player is chosen to sketch, this is what they see. The game gives them the clue based on the current theme, in this case it is simply the word “Star”, which is in the theme “Big Things”. The player draws on this canvas, and all the other players see this drawing being made in real time. There is a time limit, but in this case the object is easy enough to draw that the player finishes early. To speed things along, the sketcher can tell the game the drawing is finished by clicking the “Done” button at the bottom. After that the game sends a warning to everybody so they can finish typing their guesses.

Stellasketch - Sketch Phase

Your turn to sketch! Don't screw up!

This round's text:

Star

(lobby) Gabe has joined the Lobby
People in this room: (lobby) Gabe, ElBlarko
(lobby) Gabe has left the Lobby (gone to Romper Room)
(lobby) Gabe has joined Romper Room
People in this room: (lobby) Gabe
(lobby) Gabe: hello
Gabe has joined Romper Room
Gabe is going to sketch next!
YOU are going to sketch the next round!

When a player is chosen to sketch, this is what they see. The game gives them the clue based on the current theme, in this case it is simply the word “Star”, which is in the theme “Big Things”. The player draws on this canvas, and all the other players see this drawing being made in real time. There is a time limit, but in this case the object is easy enough to draw that the player finishes early. To speed things along, the sketcher can tell the game the drawing is finished by clicking the “Done” button at the bottom. After that the game sends a warning to everybody so they can finish typing their guesses.

Stellasketch - Label Phase

27

This is what the labeling phase looks like. It tells you to get ready to label, and it shows the sketch being made. You type in your guess into a text field and hit Enter. All the other players are doing this at the same time, but you don't see their guesses until later. Like earlier, there is a timer that lets you know how much time you have left. When time is up, it shows you the guesses made by all other participants in the order they were received.

Stellasketch - Label Phase

A new round is starting in a few seconds!

Gabe has joined the Lobby
People in this room: Gabe, askdelia, karolina
Gabe has left the Lobby (gone to Gabe's Romper Room)
Gabe has joined Gabe's Romper Room
People in this room: Gabe
karolina has joined Gabe's Romper Room
YOU are going to sketch the next round!
ElBlarko has joined Gabe's Romper Room
karolina: map
karolina: paper
karolina: book
karolina: box'
karolina: bookshelf\
karolina: bookshelf
ElBlarko: shelves

This is what the labeling phase looks like. It tells you to get ready to label, and it shows the sketch being made. You type in your guess into a text field and hit Enter. All the other players are doing this at the same time, but you don't see their guesses until later. Like earlier, there is a timer that lets you know how much time you have left. When time is up, it shows you the guesses made by all other participants in the order they were received.

Discussion: Type of Data Collected

- All sketch input has position and time
- Picturephone: labels apply to entire sketch; ratings suggest quality of drawings and description
- Stellasketch: time-stamped labels apply to sketch as it looked at one point in time

The kind of data that is collected depends partly on the nature of the game. Both of the games I'm showing today have time-stamped sketching data that involves millisecond-level accuracy. But while both systems involve text labels, they have different levels of granularity. In Picturephone, labels apply to the entire sketch, but in Stellasketch they apply only to what the sketch looked like at one point. And Picturephone has ratings applied by many players that helps to suggest how good the drawings and descriptions are. Stellasketch does not have a rating system at all.

Playtesting

- Java sometimes a problem (solution: Flash?)
- “Hard to draw with a track pad”

We put these games in front of undergraduate and graduate students in one of Ellen's courses at Georgia Tech. There were a couple of issues common to both games. First, some of the students had difficulty getting Java to work, since they were using machines in a campus computer lab, and they didn't have Java installed. So, delivery was a problem, and that could probably be addressed by using another application type, such as Flash, or maybe HTML 5 in the future. Also, many of the students either didn't have access to a tablet, or chose to use their mouse or trackpad. If you've ever tried to use a trackpad like it was a pen, you can understand how difficult that is to use.

Playtesting (Picturephone)

- Used IRC to coordinate test—users loved the social connection with each other
- Browsing and sharing links was half the fun
- Players shouldn't feel like they're doing homework

30

The Picturephone playtesting session involved about twenty users, and lasted a little over an hour. Just before the test was to begin, I decided almost on a whim to ask the students to all join an Internet Relay Chat channel so we could coordinate. And I probably should not have been surprised by this, but they really loved the chat function. As they played the game they would use IRC to chat with each other, sending links to amusing drawings and the like.

On the whole players liked the game, but sometimes felt like the describing was a little bit like “doing homework” as one put it.

Playtesting (Stellasketch)

- Had all players in one big game room (they preferred it that way)
- Had more fun than with Picturephone (many same players)
- Continued to play after test 'stopped', despite technical problems.

31

I implemented Stellasketch after the Picturephone playtesting session, so I incorporated a lot of the lessons I learned. For the most part that involved making the game seem like a fun place to hang out, so I spent some time making chat work quite well.

When we did the playtesting for Stellasketch, I asked the players to try to spread out evenly into a few different game rooms, but they all preferred to go into one big game room. The participants in this case were for the most part the same from the Picturephone session. They mostly reported that Stellasketch was much more fun than Picturephone. After I had to end the session and go offline, some of the students continued to play for about ten minutes after I had left. This was in spite of the technical problems we were seeing due to the stress on my server.

Discussion: Game Mechanics

- Funny/ironic is more entertaining than ‘correct.’



- Time:
 - Time limits: hasty sketching
 - No time limit: fancy sketching

Other game mechanics also have an effect on data quality. People know that their drawings will be seen and used by others, so there is a tendency to use the medium to amuse their fellow players, and draw things in unexpected ways because it is funny.

Time limits are another mechanic that I originally did not expect to be such a huge factor. But it turns out that when there is a time limit, people feel quite pressured to complete the drawing quickly, even if they actually have plenty of time. Originally I planned to have a time limit in Picturephone, but I left it out because gameplay seemed fine without it, and it promoted some players to make elaborate drawings.

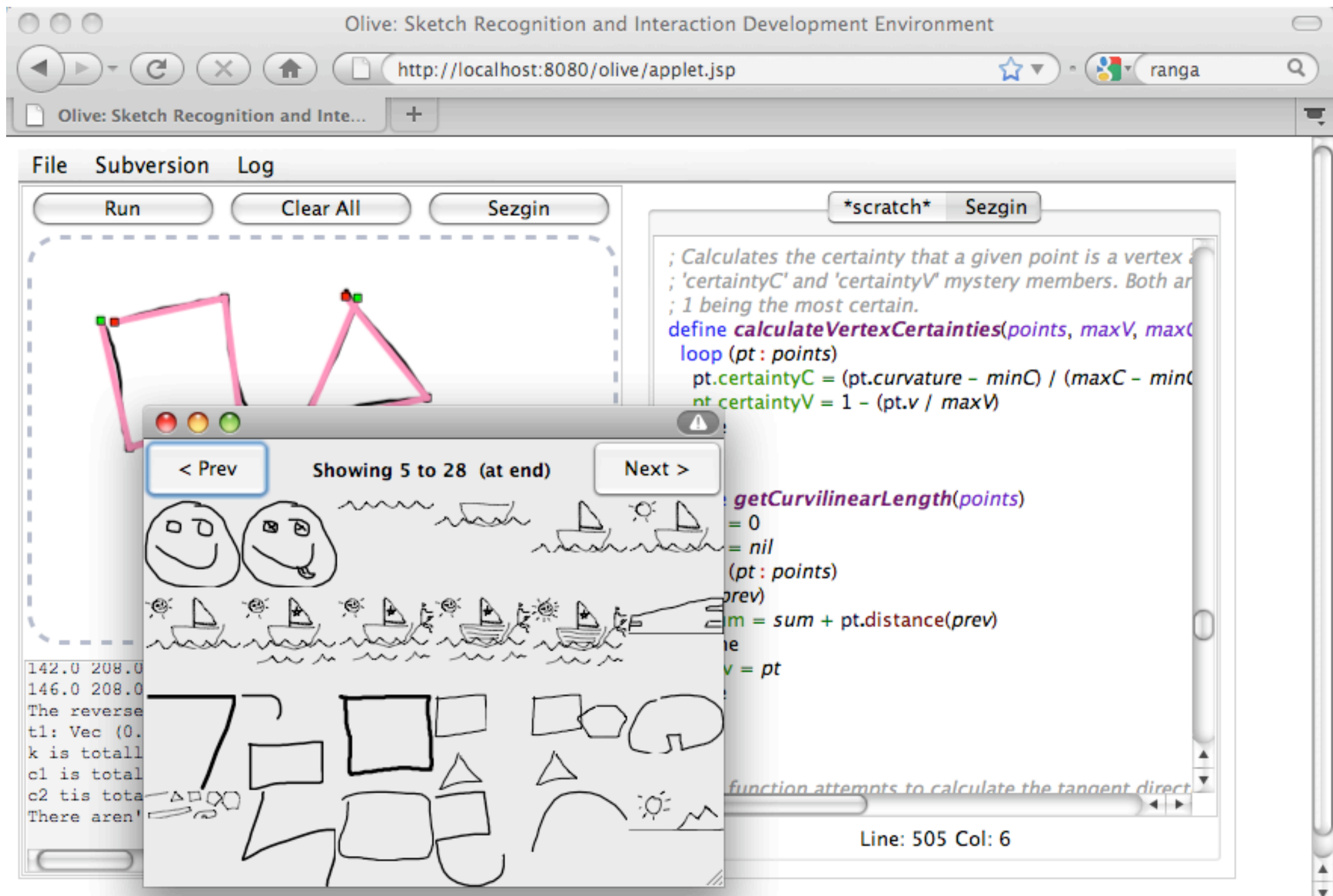
Discussion: Game Mechanics

- Asynchronous (Picturephone): Play at your own pace, choose favorite modes
- Synchronous (Stellasketch): Play with other people. It puts you on the spot! Social interaction is half the fun.

The biggest difference between the two games is that with Picturephone, it is asynchronous, meaning you can play without other people. That contrasts with Stellasketch, where synchronous interaction with other people is a fundamental part of the game. Some of the players had strong preferences towards or against either of these. The asynchronous style lets people play at their own pace and not feel responsible for keeping up with others. The synchronous style lets people hang out with friends while occasionally being put on the spot—some people like that, while some really don't.

Current Work

(Using Sketching Games data in the Olive IDE to test algorithms implemented in my Slippy sketching language.)



34

Currently, I am developing an IDE for creating algorithms related to sketch interaction. The environment is called Olive, and the programming language is called Slippy, which stands for Sketching Language In Progress. To test my algorithms, I use data pulled from my games.

Potential Future Work

- Expand player capabilities: pen tools (color, eraser, fill), add fine-grain labeling
- Other games: Memory, ‘tipping point’

Players often request expanded drawing capabilities such as color, erasers, and pen thickness. But adding these capabilities only makes sense if we would like to use the data it provides. And it isn't too hard to think of more game-like environments that can gather interesting information about sketches. For example, imagine a 'memory' game, where you are shown a drawing, and you must later sketch it from memory—what you draw is likely to represent the salient parts of the original. Or consider a game that plays back a sketch as it is made, but in slow motion, and players are asked to label it. At some point, the drawing suddenly becomes recognizable—so determining what exactly made the drawing identifiable may be worth knowing if we could leverage that information in sketch recognizers.

Potential Future Work

- Well-engineered sketching games, possibly on social sites like Facebook
- Sketches as CAPCHAs

36

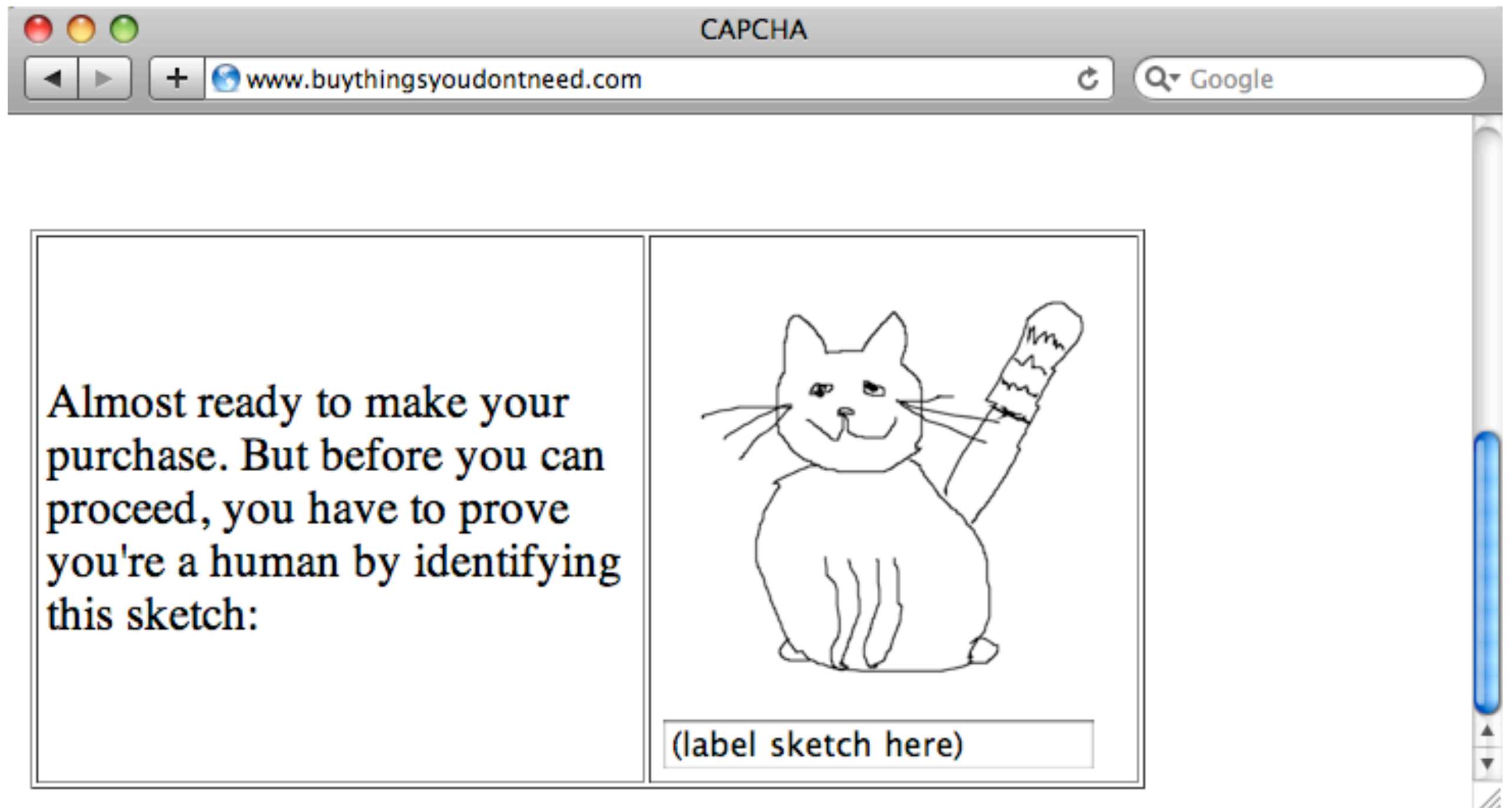
To really be useful, human computation requires a lot of users. My games have had quite limited exposure—40 users on each game, and only about 600 sketches in total. I imagine a well-engineered, debugged sketching game on social networking sites such as Facebook would likely be played quite a bit.

And the idea of using human computation to label sketches might even serve as an effective type of CAPCHA.



This is an example of a CAPCHA. It is a little puzzle that should be easy for a human to solve, but is currently difficult for computers to solve. It is used to prevent software agents from doing things that only ought be done by people, like buying things from online stores, or signing up for new web mail accounts.

This would be cool



But instead of a CAPCHA based on deciphering warped text, you might be asked to identify what a sketch is about. Obviously there should be room for interpretation. This is most likely a cat, but “kitty” should work, as well as other animals that look sort of like this, such as “raccoon”.

On the Web!

`six11.org/picturephone`

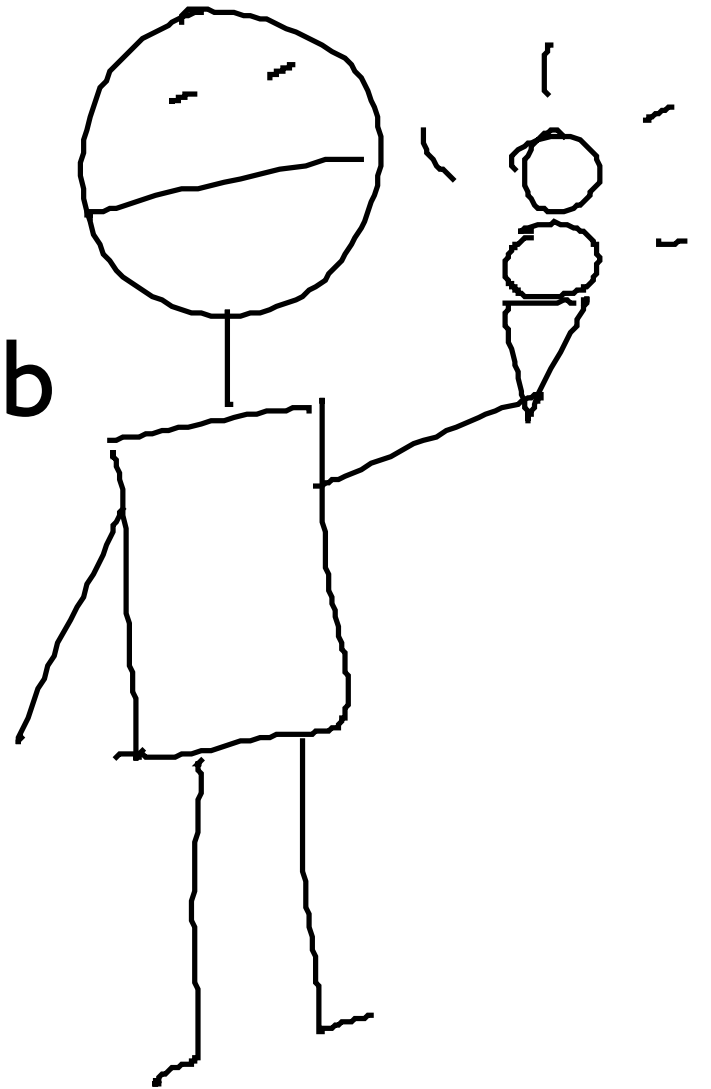
`six11.org/ss`

Games for Sketch Data Collection

SBIM - August 2, 2009

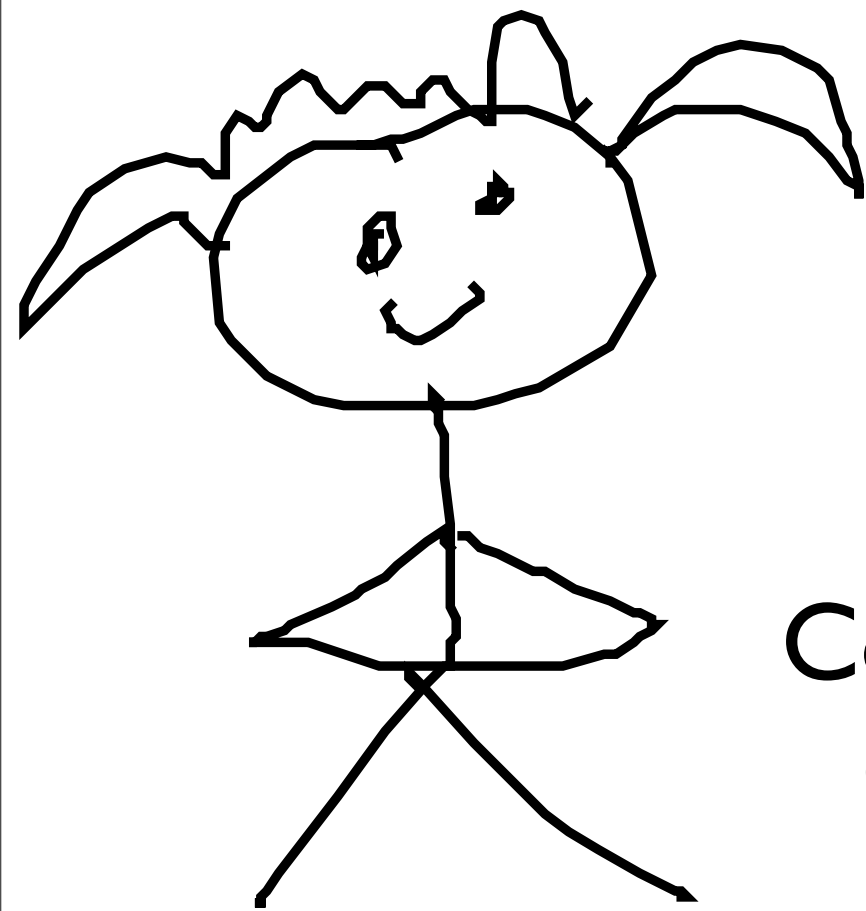
Gabe Johnson

Computational Design Lab
Carnegie Mellon



Ellen Yi-Luen Do

College of Architecture &
College of Computing
Georgia Tech



I want to thank my advisor, Mark Gross, for unwittingly contributing to this work early on, and my other advisor Ellen Do for her guidance and support with the playtesting. I also want to thank my girlfriend Karolina who also happens to be the funding source for this work.