

CS228: Human Computer Interaction Final Project

Monday, December 1, 2014

Nick Strayer

Introduction:

In this project the goal was to introduce a completely new user to the first ten American Sign Language (ASL) numbers. Using principles from human computer interaction (such as object consistency[1]) and cognitive psychology (the generation effect [2]) I designed a game to minimize the time and maximize the true understanding in the learning of ASL numbers. In addition, I decided to take the project in the direction of a web application using OpenGL. This decision maximizes the reach of the program as one only needs a modern web browser and a Leap Motion to use the software as opposed to a clunky and unfamiliar command line interface.

Additions:

Hand-bivalence: (<https://www.youtube.com/watch?v=p2GIPCWgc4U&list=UUnonD6A2upIGsOiggrhXwIA>)

The program does not care if you use your left hand or your right hand, if you sign in the upper right, the lower left, or anywhere in between, or how big your hand is.

Realistic Hand(s): (<https://www.youtube.com/watch?v=VX3GZnQwQZ8>)

As opposed to sticking with an abstracted hand composed of homogeneous lines I utilized OpenGL to render and draw to the browser a (semi) realistic hand. (As a cool side-note: this uses hardware acceleration in the browser.) I did this because, not only does it look cooler, but it eliminates confusion that can come from issues such as drawing order of fingers and things simply being too close together. In addition, it allows the implementation of glowing fingers...

Glowing Fingers: (<https://www.youtube.com/watch?v=kxKz3034u5E>)

As opposed to bashing the user over the head with instructions a la a non-dynamic animation I decided that a more subtle, but also more informative, approach would be better. If a user is having trouble signing a number (or simply is new to it) the fingers that the user is supposed to extend will glow green. This allows the game space to remain uncluttered and non distracting.

Celebration Animation: (See glowing fingers video)

I added a rather inconsequential (but kinda cool,) d3.js success animation to the game. When a user successfully signs a number an arc appears from beneath the user's hand and "explodes" onto the page along with the success message. It is a pretty reward for success. This very well might be unnecessarily complicating the game, but user feedback I have received is positive.

Walk-through:

- First the user accesses the web app by going to the URL (www.uvm.edu/~nstrayer/signLanguage).
- They are greeted by a "Welcome Page". Here they enter their name and proceed. If they are a new user the system will initiate an entry into the HTML5 localStorage database.
- Upon logging in the user is greeted with the main game page. The progress bars animate to their positions (current number, overall, and past performance metrics) and an animation shows a user interacting with the system.
- A small image prompts the user to put their hands over the leap motion, upon doing so the animation hands become the user's hands and the game starts.
- The number they are supposed to sign is represented by a floating bubble on a vertical number line. Next to it is a bubble indicating the number that the user is

currently signing. Upon successfully signing a success animation displays and a new number is chosen. If the user runs out of time a “fail” animation shows and a new number is chosen. At the beginning of a new number all of the performance metric bars animate to their new positions. This gives the user a physical sense of progress (object consistency [1]).

- The amount of time that the user has to sign the number (demonstrated by to the user by a background color change) and the amount of hints (as glowing fingers) decrease smoothly as the user’s success rate and familiarity with a given number increases. In addition, the system will slowly introduce new numbers to the user as they play and master the previous ones.
- The user ends the game by simply closing the tab. All data is saved automatically while playing so sudden log-offs or crashes are not harmful. Upon returning the game picks up where it left off.

User Testing:

The system was tested on a large number of my classmates and family members which had a massive impact on my design process. The design of the database lends itself well to user testing. With a time based representation of performance.

One aspect that I did not expect user testing to have such a large role was in the colors of the game. At first I used the standard encoded colors such as “red” and “blue” but then a user mentioned that they thought the colors were too jarring and made the game uncomfortable to look at. I then switched to a divergent color set from Cynthia Brewer’s Color Brewer [3].

The following is a dump of some of the more recent (and brief) user testing sessions. These took place over thanksgiving with relatives.

Sample user data:

```
console.log(localStorage["Annika"])
```

```
{“0”:[0,1,0,1,0,1,1,0,0,1,1,1,0,0,1,0,0,1,1,1,1,0,1,1,1,1,1,1],  
  “1”:[1,0,0,1,0,1,1,0,1,1,0,1,1,1,1,1,1,1,1,1,1,1,1,1],  
  “2”:[1,0,1,1,0,0,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,0],  
  “3”:[1,1,1,1,1,1,1,1,1,0,1,1,0],  
  “4”:[0,1,1,1] ... ,  
  “logins”:2, “lastSuccessRate”:70}
```

```
console.log(localStorage["Thadeus"])
```

```
{“0”:[1,1,1,0,1,1,1,1,1,1,1,1,1,1],  
  “1”:[0,1,1,0,1,1,1,1,1,1,1,1,1,1,1],  
  “2”:[0,1,1,0,1,1,1,1,1], ...  
  “logins”:1, “lastSuccessRate”:90}
```

You can see a progression through the data of initial confusion with a number, followed by a rapid acclimation. A statement from user “Annika” after testing was *“At first I was very confused, but then it became clearer and very quickly too easy and [boring].”* This statement reinforced my new interface for what number the user is to sign and what they are signing. In addition, it influenced me to develop the more snazzy success animation. Due to database dumps

necessitated by feature changes I lost most of the actual raw data from user testing. User Testing played a major role in the development process, especially after implementing new features such as the progress visualizations.

A point of note: User “Thadeus” (my 86 year old grandfather) has a daughter who is deaf, but he never learned sign language, however, he saw enough to be familiar with the letters. He said that using the program brought the numbers back very quickly. The failures after the first attempt for the numbers are actually due to leap motion hand model errors.

What next?

While this was a fascinating introduction to the sign language there are a lot of things I would want to do further:

Cool progress visualization.

It would be nice to have a stand alone visualization of your progress through the system. More complicated than is possible in the game. This would take the form of a time plot of a running average success rate for a given number, overall, and compared to others.

Letters/ words:

Numbers are great, but they are an absolutely inconsequential subset of ASL. I would like to expand to letters and words as well. Motion Savvy (<http://www.motionsavvy.com/>) is attempting to build a product that can translate words and letters from ASL using the Leap Motion. I fear however that, due to the similarity of many of the hand positions, combined with a huge number of signs, this will prove very difficult. I think that a game to teach would be easier, as you could dynamically tune the classifier so that it is more likely to detect a successful attempt as a letter

or word that looks very similar to another. It would be fantastic if Motion Savvy opened their training sets to the public (they are already crowd sourcing the generation of training sets so hopefully they will).

Make it more of a game:

Since I already have the program running in OpenGL adding much more complex scenes with things like physics would not be hard. I think a simple game with something akin to doors at the end of each level that wont open unless you sign increasingly difficult things (starting with simple numbers and letters up to full sentences) would be pretty fun and motivating. Disguise the learning!

Hosted Game:

www.uvm.edu/~nstrayer/signLanguage

Github Repo:

<https://github.com/nstrayer/cs228/tree/master/final>

Citations

1. Use of Animation in Information Visualization. G. Robertson. Microsoft Faculty Summit 2010.
http://research.microsoft.com/en-us/um/redmond/events/fs2010/presentations/robertson_vizualization_and_interaction_rfs_071210.pdf
2. The generation effect: Delineation of a phenomenon. Slamecka, Norman J.; Graf, Peter
Journal of Experimental Psychology: Human Learning and Memory, Vol 4(6), Nov 1978, 592-604. <http://dx.doi.org/10.1037/0278-7393.4.6.592>
3. <http://colorbrewer2.org/>