

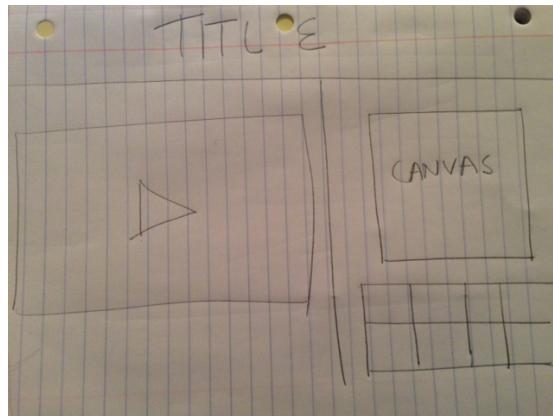
### I. Development of Design

In the following section of my design document, I will explain the thought process behind the creation of the design for Signal.Me, my gesture-based video player.

#### 1) First thoughts – how do people interact with video players?

When I first set eyes on the problem at hand, I asked myself this very simple question. When I use software such as VLC Media Player or use the internet to watch videos on Hulu.com, my main goal is (to state the obvious) to watch a video with minimal interruption. I also wish to use familiar controls to adjust my viewing experience to a custom setting. Once I achieve my ideal setting, I want little more than to simply lay back and watch till the video is done.

This simple use case is what inspired me to create, first and foremost, a *simple* application with no frills. I aimed at designing a website that has minimal text, easy-to-see colors and a smooth video-watching experience.



(Figure 1)

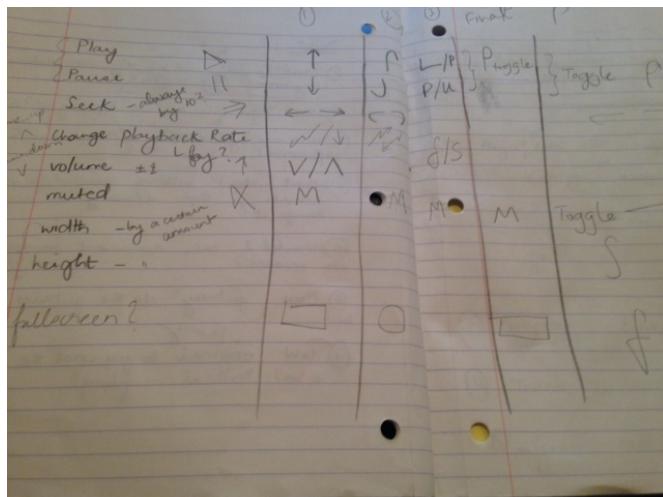
#### 2) A few things that I had to consider before I started creating my gestures were:

- Should they imitate video players?

I initially decided to settle on a very familiar set of gestures that imitate generic video players (see Figure 2). This made sense to me since it was following Nielson's heuristic #2 (match between the system and the real world). I reasoned that people would find it easier to acclimatize themselves to the user interface if they didn't have to keep referring to the gesture chart.

However, as I thought more about it, I realized that using gesture-based controls was an opportunity to break out from the traditional designs of video players. After all, why would someone want to swipe "||" for pause when they could simply press the spacebar? The gestures *had* to be far more intuitive and easy to use.

As I started using the gestures I was developing (at this point, both single and multi-stroke), I used the timer on my iPhone to time how long it took me to draw the gestures, have them recognized and give me feedback. On average, when I used the gestures that had multiple strokes with another stroke to signal the recognizer that I was done with my gesture, it took about 5.7 seconds. I realized that it was far too long a period of time for a user to be spending simply *wanting* something to happen. On the other hand, it took about 1.4 seconds for the single stroke gestures. This drastic difference is one of the reasons I decided to eliminate multi-stroke gestures.



(Figure 2)

- b. Should they be single swipe or multi-stroke?

Referring to the reasons stated in (a) above, I decided to keep my entire gesture set as single swipe with no multi-stroke components. As you will see in Figure 2, my designs slowly evolved to become, in my opinion, more fluid and easier to draw. There was also less room for error as I started to create very different, individual designs for each control.

One of the important factors that went into the evolution of the design process was my assumption that the user should be able to use the canvas with their dominant and non-dominant hand. I decided it would be unwise to assume that the user should only use his/her

dominant hand, and thus discounted a lot of the more complex gestures that I had earlier planned on using. For more detailed explanation on the meaning of each gesture, please jump to point 4 below.

c. Should users be allowed to change them?

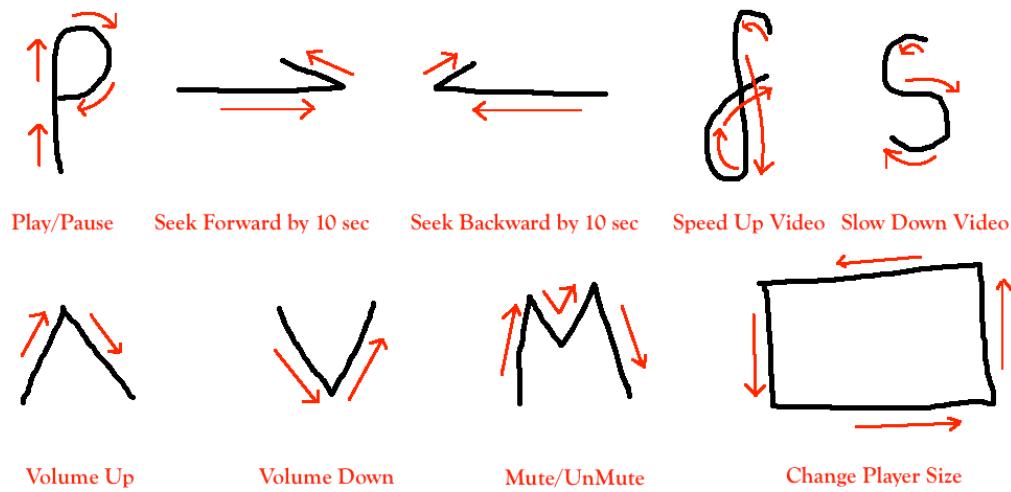
I decided that users would *not* be allowed to change existing or add custom gestures because I would want them to learn and recognize the gestures already present, rather than have to recall the gestures they added and get confused (Nielsen heuristic #6 – Recognition rather than recall).

3) \$1, \$P or \$N?

For reasons explained above, I decided to use the \$1 recognizer instead of the \$P or \$N, because I intended to add only single-stroke gestures. Also, I set the “use protractor” option to ‘true’ permanently for faster recognition results over the default ‘golden section search’ method.

4) Final Gestures

As you can see from Figure 3, the final set of gestures that I designed has very specific patterns and uses. I decided (following Neilson’s heuristic to adhere to aesthetic and minimalist design) that I would have some of my controls attached to the same gesture with a toggle mode.



(Figure 3)

- The Play/Plause controls are toggled by the gesture ‘P’. After going through several gestures, I decided that ‘P’ was the most suitable since

Darshana Umakanth  
COMS4170 – Assignment 3

it was the characterizing letter in both controls, but was also easy to draw with both hands on the canvas.

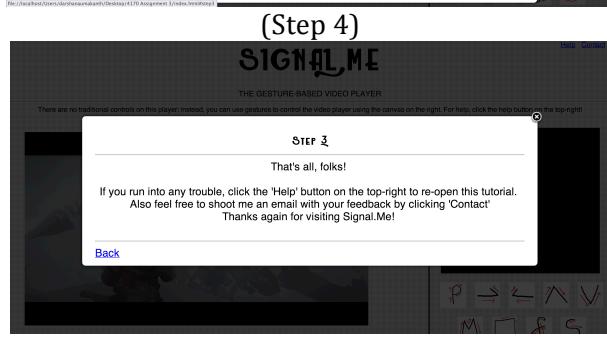
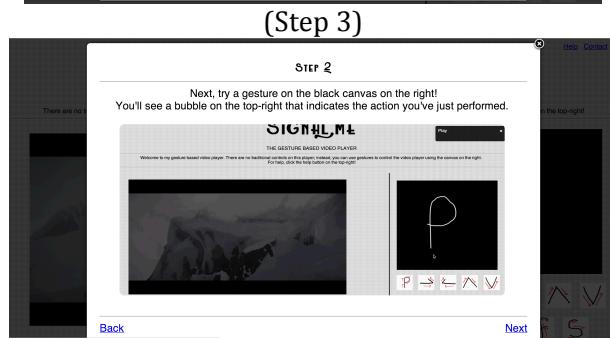
- The Mute/UnMute gestures are very similar to that of the Play/Pause gesture. I chose the letter ‘M’ because it mirrored the first letter in Mute and was also easily recognizable and stuck in my memory.
- The Seek Forward/Backward gestures are mutants of the arrow symbol. My initial inspiration came from the reversible reaction symbol in Chemistry, with one half of the arrow pointing forward and the other half pointing back. As you can see, they are also easy to draw and hard to mistake for any other gesture.
- The Volume Up/Down controls are similar to the seek gestures, in that they mirror each other. I had initially chosen the ‘v’ symbol to mean volume up and the ‘^’ to mean volume down because I was looking at ‘v’ as the letter ‘V’ which stands for volume.

However, after I beta tested the website with a few friends from non-tech backgrounds, it was pointed out to me that it made more sense *visually* to have ‘v’ stand for volume down since it was an arrow head pointing downwards and vice-versa for ‘^’. This was an example of how testing opened my eyes to a problem I would not otherwise have noticed.

- The Toggle Video Size gesture is drawn from geometric sensibilities. I used the rectangle because it is an easy shape for the user to recognize since the player itself is rectangular. The gesture toggles between three different sizes (Small, Medium (default) and Large).
- The Increase Playback Speed gesture is alphabetically based. I used the ‘f’ from the word fast to create a cursive form of the small letter ‘f’.
- The Decrease Playback Speed gesture is the letter ‘S’, which stands for slow. In keeping with my theme of having the letter as the gesture, I decided to use to allow the player the decrease the playback speed of the video.

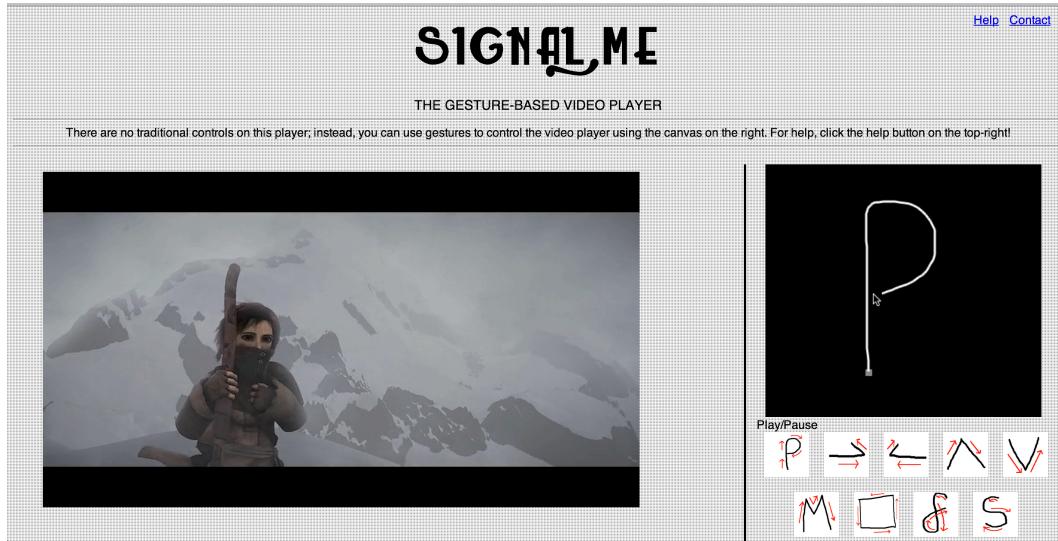
## II. What Signal.Me Does

- When the user first opens the website, there is a modal tutorial that opens up. It guides the user through the steps of using the website, i.e. what the pictures below the canvas stand for and how to use the canvas to control the player. The tutorial includes GIFs that show a demo of how one should go about using the above-mentioned features. Once the tutorial ends, it can be re-opened at any time by pressing the ‘Help’ button on the top right of the screen.



(Figure 4)

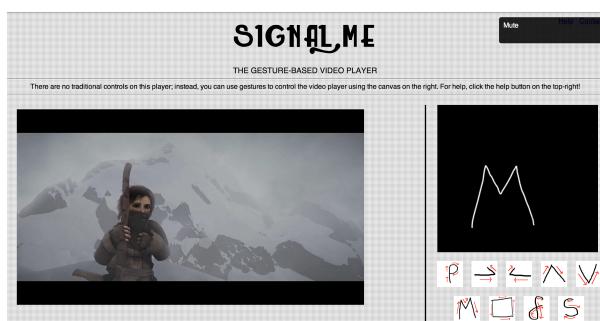
- 2) The user can then proceed to viewing *how* to draw the gestures by hovering over the pictures below the canvas. On hover, the canvas has a seamless overlay that shows the user how to use the mouse to draw the gesture, which disappears as soon as the user stops hovering.



(Figure 5)

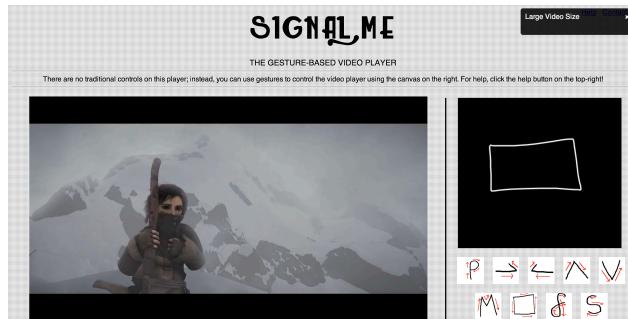
- 3) Once the user has seen the demos for how to use the canvas, he/she can start drawing on the canvas to control the player. The following gestures are recognized by the \$1 recognizer:
- Play/Pause
  - Seek Front/Back
  - Mute/UnMute
  - Volume Up/Down
  - Faster/Slower
  - Video Size Toggle (with default=medium size)

In order to allow for maximum feedback and flexibility, I added the growl notifications to give the user feedback about what action is being performed after each gesture.



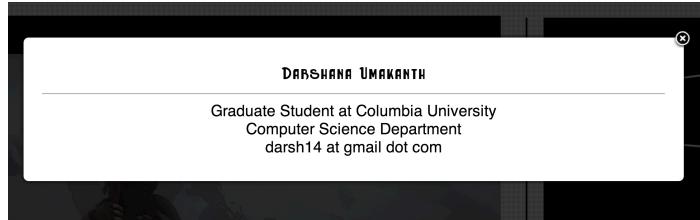
(Figure 6)

Darshana Umakanth  
COMS4170 – Assignment 3



(Figure 7)

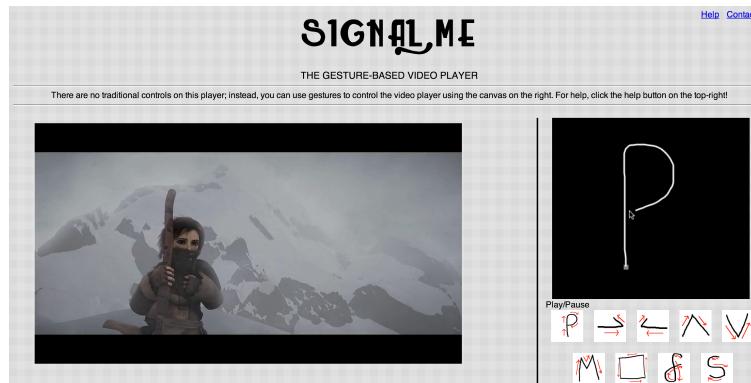
- 4) I added a 'Contact' link at the top of the website to allow the user to send me their feedback about the player, so I can continue improving it.



(Figure 8)

### III. Signal.Me - Final Design Decisions Explained

- 1) Basic website interface: I decided to go for a simple block-type design for the basic elements on the site. The colors are monochromatic black and white shades, so as not to draw the attention of the user away from the video player (Neilson's Heuristic: Minimalist Design). Even the canvas is black with bright white cursor marks, so the user can clearly see the gesture they are making. There is very little text on the main page itself, because I wanted to keep only *useful* text in view at all times. For example, I give the two most important pieces of information in the blurb below the main title – that the player can be controlled using gestures on the canvas, and that the tutorial can be accessed under 'Help'. I used blue to highlight the 'Help' and 'Contact' buttons so they would stand out in case the user needed them (Nielsen's Heuristic: Help & Documentation)



(Figure 9)

- 2) Since the primary use of the website is to watch and control the video player, I decided to "de-clutter" the design by having features embedded into each other or placed discreetly above one another. For example, my first impulse was to have a separate window open when the user hovers over each pictorial gesture below the canvas; however, I soon realized that users would need to (at least initially) refer to them several times, perhaps even *while* the video is playing. Thus, having a separate overlay pop up that covers the video would be intrusive. For these reasons, I decided to hide the demo GIF files on initial load, and only have them appear seamlessly above the canvas when the user hovers over each picture. This way, the video is unobstructed and there is minimal visual change on the site itself. This, I believe, allows the user to minimize errors in using the program and also gives the user experience flexibility (Nielsen Heuristics #5, 7).

Also, the ability to open and dismiss the tutorial also, I believe, achieves a minimalist design.

- 3) jGrowl – I believe that Nielson's Heuristic #1 (Visibility of System Status) is one of the most important principles to be followed in UI design. To that end, I decided to use a handy plugin called jGrowl that displays sleek "Growl" or bubbled notifications to the user as he/she draws gestures on the canvas. This allows the user to see if the gesture has been recognized, and what function it corresponds to on the video player. Thus, *if* the gesture happens to be recognized incorrectly by the recognizer, the user is at least aware of what action was performed on the player, and is able to reverse it (Nielsen's heuristic: Help users diagnose & recover from errors easily).



(Figure 10)

- 4) Tutorial – In order to ease the user's transition from using a traditional video player to one that is controlled only by gestures, I decided to have a modal overlay guide the user through a step by step tutorial of how to use the website. The tutorial is embedded with lightweight GIFs that show the user how to use the gesture pictures below the canvas and also guides them through using the canvas to control the player. The tutorial's main aim is to have the user recognize and familiarize herself with the website *before* even having the chance to use it, thus boosting the chances that the interface will be easier to navigate on first use. (See Figure 4 series in Section II)

Note: Due to a small bug with the canvas where upon scrolling down on the website, the cursor position and the drawing position no longer align, I initially decided to disable scrolling on the website and have all my elements fit into one view; however, I realized that this is more of a hindrance for those who wish to keep custom window sizes, and so I left it as is.

Note: Libraries/Technologies used – jGrowl, jQuery Modal, Fonts, HTML5 video, canvas, LICEcap for GIF capture.