

Background

Visually impaired and blind individuals often need to rely on other people for way-finding or limit travel to familiar areas. Some examples of these dependent relationships are obtaining verbal directions, ordering a static tactile map much before travel begins, and soliciting the help of a sighted guide. Although often reliable means of way-finding, none of these methods allow for complete independence. Other factors to consider in evaluating the quality of a way-finding method are cost, ease of use, range of locations covered, and richness of street information available. Keeping these and other considerations in mind, we have developed the *Street Sound* web application.

Our Concept

Street Sound is a free web application for the visually impaired to access street information. The digital nature of the system also allows anyone with a computer and internet connection to access the information online for free and customize the map on the fly. The only tools needed beyond the internet connection are *Firefox* web browser with the *FireVox* text-to-speech plug-in and an audio output device (e.g. headphones or speakers). Optionally, a pen tablet can be used to explore the map because it seems like a more natural interface than a mouse, especially for blind users who may not be familiar with the mouse.

Using only sound (and not force or tactile feedback), we believe the individual can explore the map and create a lasting mental image of the area to orient him/herself before arriving on scene. When the user follows a street on the map using a mouse or another pointing device the street names, intersections, and nearby venues are verbalized. If the cursor deviates from the road, a beep notification will sound. This feature allows the user to follow the street and become familiar with the area without sight. Now what a sighted person can see is made audible; that is, the ability to “navigate with your ears” is enabled.

Features

Below is a screenshot highlighting some of the features of *Street Sound*. Like all Google Maps web applications, the user can center the map at any address, zoom and scroll the map, and find local businesses within the visible area. Unique to *Street Sound*, though, is the ability to verbalize the plethora of information embedded in the image.

A sighted person would not have difficulty determining where the edge of the map is or whether the map was finished loading after shifting the field of view, but these are difficult events to realize with limited sight. As a result, *Street Sound* will audibly alert the user of these actions.

A pop-up window system working with the cursor's latitude-longitude coordinate location and the *Geonames* database allows the *FireVox* text-to-speech plug-in to speak street name, intersection, and local business information.

Additionally, all commands can be done with keystrokes (see <http://code.google.com/apis/maps/documentation/reference.html> - GKeyboardHandler for chart) so that the user does not need to depend on visual information to interact with the *Street Sound* map interface.

If the individual desires, he/she can purchase a pen tablet like the Wacom *Intuos3* 9x12" one we experimented with to interact with *Street Sound*. After a short installation process and minimal personalization of pen settings, the user can use their fingers to control the cursor location with a motion similar to reading Braille or discovering another tactile surface.

Capitalizing off of the normal activities of a visually impaired individual (e.g. Internet browsing with a text-to-speech aid, keystroke interaction, finger scanning to learn new information), we feel that the *Street Sound* web application is a promising method for way-finding with an extensive list of future development goals and even applications beyond the scope of the system itself.



Figure 1: Screenshot highlighting some of the features of *Street Sound*. The green overlays make visible the objects that contain the street information. A callout box with three curved lines in the upper corner indicates auditory feedback from the system. *Street Sound* can be accessed at <http://italia.cse.ucsc.edu/~StreetSound/>

Implementation

Street Sound makes use of the Google Maps framework and number of other external free open source scripts for a nearly completely JavaScript-based system (see Figure 2 for chart summary). Attributes are easily modified within this framework so as to meet the developer's needs. For example, we have disabled draggable scrolling with the line `map.disableDoubleClickZoom();`. Further exploration of the code and comments, as well as the Google Maps API (see Additional Resources section) will reveal other options that can be enabled or disabled as needed.

Script Title	Purpose
Google Maps (default)	Define all attributes of map and appropriate interaction behavior
Google Maps Local Search	Allow searchability for local venues and return location details (e.g. address, reviews)
BDDCPolyline	Improve speed of mouse on/off response over default GPolylines (no longer in use)
JavaScript Object Notation	Process script returned by <i>Geonames</i> database
SoundManager	Play sound (.mp3) clips with the support of an XML file and Flash movie (.swf)

Figure 2: Summary of external scripts implemented in *Street Sound*.

An advanced option employing AJAX (asynchronous JavaScript and XML) is the *local search* module. Inclusion of previously developed scripts from Google and feature customization declarations is all that are needed to get local venue addresses, phone numbers, location, etc. on the map.

As seen at the top of the screen, a form processes the user's request for where to center the map (default is the Metro Station in downtown Santa Cruz) and sends this information to a database to determine the latitude-longitude coordinate that corresponds to the address. This process—called *geocoding*—is handled by another script developed by Google.

Once the map with all its associated attributes are loaded, what are called *overlays* can be added to the map. Examples of overlays are *polylines* (like the green street segments), location markers, and polygons (not used in *Street Sound*). Like many other objects embedded in the map, these polylines have *listeners* for actions like click, mouse on, and mouse off. If any of these events occur while the cursor is on a polyline (or wherever the listener is), a function is executed. (See

<http://code.google.com/apis/maps/documentation/events.html> for more information.)

Through this multitude of listeners *Street Sound* determines what the user is doing and how to respond.

The most complex part of the code lies in the two-dimensional array data structure for keeping track of which coordinates have already been explored. Thinking of a two-dimensional grid superimposed upon the map image, we efficiently make calls to the database by using only the coordinates at the grid intersections. When the field of view shifts due to a scroll or resolution change, only the newly exposed regions along the gridlines will be sent to the database for street information because we have a record of the area that has already been explored; there's no need to redo all the work. Through trial and error experimentation, the grid size was determined to be a square with side .0034 degrees.

Overview of Basic Operations

Given this technical background, it is now easier to understand the mechanics of *Street Sound*. The initial load of the page draws the map (centered at the Metro Station in Santa Cruz, default location) and sets the features we have determined would be best for the visually impaired (i.e. small control for field of view alteration, disabled double-click zoom, disabled popup info windows, disabled draggable map moving, local search results displayed without zooming to include more markers, and enabled keyboard shortcuts). Listeners for activities that change the field of view (i.e. zooming or scrolling) and for the cursor leaving the map area are also added since these are very visually dependent actions.

Then at each intersection on the grid, the corresponding latitude-longitude coordinate is sent to the *Geonames* database to return a script containing a JavaScript Object Notation object with nearby street segment information to the initial point. These segments are enhanced with listeners for mouse on/off behavior (as needed for the beep notification through the SoundManager script) before being added to the map. Once all the points in the grid have been processed, the system is ready for interaction.

The listeners handle the segment interaction (i.e. mouse on/off) as well as handle the click command to determine where the cursor currently rests. The coordinate for wherever the map is clicked is extracted by Google Maps and sent to the *Geonames* database for detailed street information. This process is called *reverse geocoding* because instead of starting from an address to get a latitude-longitude coordinate, we do the reverse and request further information from the coordinate. In the scenario depicted in Figure 3, the mouse is at coordinate (36.9722, -122.028) and the database would return that the cursor is the

intersection of Lincoln St and Cedar St, and near, for example. Lincoln St, Cedar St, New St, and Walnut Ave. (Note, this is done through two separate calls to the web service—one for intersection information and the other for nearby streets.) Finally, this result is verbalized with the *FireVox* plugin by passing it in the title tag of a new window since the reader natively reads this information.



Figure 3: The cursor is at the latitude-longitude coordinate (36.9733, -122.028), which is sent to the *Geonames* database to get more street information. This process of gathering data from a coordinate is called reverse-geocoding.

Improvements and Future Applications

Street Sound was originally developed as a proof-of-concept system. As such, there are many opportunities for improvement in the current implementation and even more future applications that have sprouted from this kernel of an idea.

Maintaining the web-based system we already have, we plan on creating a local database of the street data to break our dependence on the *Geonames* server and the clutter on the page caused by so many scripts. (Each call to the server returns one script. Despite writing a function to keep the number of scripts on the page less than 20, many internet connections still return errors for having too many scripts to process.) We believe that this improvement would expedite all event processing and initial load time. This database is free and available for download, but it's important to note that the *Geonames* information is not the same database Google uses to generate its maps. This difference explains why the green segments do not always align with the Google Maps image. Ideally, gaining access to the same database Google uses would remedy the situation. (Note, though, that even if the user clicks on an area of the map without a street segment, the street information can still be found because the database only needs is a latitude-longitude coordinate.)

Currently *Street Sound* is predominantly a click-based system whereby street names or intersection information is only pronounced if the user clicks. Eliminating the need to click for feedback (e.g. announce street names as soon as the cursor changes streets or intersections and perhaps using the click command as a way to get a reminder of the cursor's location) would require much faster event processing that will be possible once a local

version of the *Geonames* database is made. AJAX technology might be a useful tool to achieve this.

Using some of these same concepts, a means to guide the user to a location on the map (e.g. search result marker or initial address) would also be helpful to the visually impaired individual. From the cursor's current location, sound could be used in the form of words or tones to steer the individual.

Detecting street shape and orientation is also a difficult task (e.g. whether street runs North-South, curves at the end, etc.), so devising an algorithm to use the segment coordinates to approximate the shape and orientation as well as convey this information to the user would be another valuable improvement.

Even before we begin to make these improvements it is important to recognize that at print time *Street Sound* does not accurately notify the user whether he/she is on a street. There exists a buffer zone around each segment where no beep sounds when the cursor deviates from it (For an explanation of Google's algorithm for cursor-polyline interaction, see <http://googlemapsapi.blogspot.com/2007/09/v288-clickable-polylines-polygons.html>). This means that the system is essentially indicating that the cursor is on the street by not beeping when, in fact, the cursor is not on a street; a critical flaw in the implementation. Implementing the BDCCPolyline script written by Bill Chadwick to override the native GPolyline class solved the buffer zone problem but at the same time introduced erratic mouseon-mouseoff behavior—particularly for intersecting segments—so this was solution was abandoned.

Algorithms and code aside, we are still experimenting with the best interfacing device to interact with *Street Sound*. We currently use the pen tablet, but the new research and presentations we have seen on multi-touch displays present an appealing alternative interface. Work nearly began this summer creating a multi-touch projected screen using the hardware inside Nintendo's Wii-motes because the game system is much cheaper and easier to find. (See TED Talk by Jeff Han from 2006 <http://www.youtube.com/watch?v=QKh1Rv0PIQQ> and Carnegie Mellon graduate Johnny Lee's work <http://www.cs.cmu.edu/~johnny/projects/wii/> for more information.)

Branching outside this original framework, we can envision using the technology in a GPS-enabled mobile device in which the user's physical latitude-longitude location would replace the cursor's contribution and the device would verbally update as the user moved. Even without the GPS information, a mobile or iPhone application would also be a valuable development. Any of the previously listed improvements could also benefit from Bluetooth

voice interaction because this would be another way to remove visual dependence (e.g. state desired location instead of using a keyboard).

With so many future application ideas and the continuing need to make maps and street information available to the visually impaired, work will continue on *Street Sound* beyond the 2008 SURF-IT program in hopes of reducing the difficult task of way-finding as a visually impaired individual.

Additional Technical Resources

Google Maps Information

- <http://code.google.com/apis/maps/> (includes API Reference, Articles, Blog, Forum, Examples)
- <http://code.google.com/apis/maps/documentation/>
- http://en.wikipedia.org/wiki/Google_Maps
- <http://www.bdcc.co.uk/Gmaps/BdccGmapBits.htm>
- <http://maps.forum.nu/>

Geonames Information

- <http://geonames.wordpress.com/>
- <http://www.geonames.org/export/>

Setup Notes

- In the *FireVox* options, check only 'Speak Events' and 'Brief Mode'
- In *Firefox*, be sure the 'search for text when I start typing' is not selected because this will disable the already sometimes finicky keyboard shortcuts for the map.
- Also in *Firefox*, hide unnecessary toolbars and the status bar to maximize the screen area for the map.
- For the pen, choose the option which disables clicking when dragging over the tablet since this would result in a continual call to the database for street names. Only clicking the button on the side should trigger a click event.