i. Table of Contents

ii. Table of Figures

Introduction

1.1 Motivation

Cartography has been an extremely useful tool for over 4300 years. The Babylonians created the earliest known maps about 2300 B.C. on clay tablets. The maps of today contain much in common with those of the Babylonian time period; most are flat depictions of an irregular surface that retain focus on the place that they are portraying. However, technology now allows us to change the map dynamically, customizing the map to each user that interacts with it. Before one can utilize this ability, he must have an understanding of place and time, with a strong knowledge of how they affect a person. This need to comprehend the significance of place and time as they influence people forms the basis of this project.

The original reason for the placeMap project was to merge the ideas of maps and user based systems together. The hope was to better understand how people view time and place, and how this in turn affects them. CampusMap is the actual project that I undertook and aims to create an application to realize the goals of the placeMap project.

1.2 Goals

The CampusMap project has several main goals. The principal objective is to create a system that allows users to visualize events, people, and location together. Because this is going to be a research tool as well as a usable application, there must be an easy interface to allow for expansion. There must be several channels that allow effortless data collection, as this information will be the main criteria for future studies. As for the application itself, it should be generic so that other schools or groups may

employ it in their own location. To complement the option for others to implement the software, there should be expandability to allow for the display of other types of data.

In terms of usability, the application should allow customization. Because this is a user-based system, the user should have control over the data that is displayed such as filtering, sorting, and placement. Security of information is of low priority since the service will keep no crucial personal information, eliminating the need for high encryption processes such as Kerberos.

2. Design

2.1 Overview

The premise of the architecture was to build a user interaction layer on top of the Yahoo! Maps API. I was encouraged to use the Yahoo! API over other possible APIs because of the Adobe (formerly Macromedia) Flash[i] platform. The Flash Player[ii] is cross platform and ideal for web-based deployment solutions, ensuring that virtually anyone with an internet connection can view the application with the free viewing software. The platform holds one major advantage above competing open source mapping APIs in that are typically produced in plain JavaScript[iii] or Ajax[iv]. Flash uses vector graphics, promotes animation, and compresses file sizes. It has vector graphics that greatly reduce the file size and ease of producing high quality images that do not distort when resized. Its smaller fast transmission speeds and on-the-fly rendering simplify animation procedures by not forcing the developer to reduce user interaction for the sake of size. Using the provided functionality of the API also facilitates the ability for the user to choose a satellite, standard map, or hybrid view of that location, further customizing the

user's experience. These factors all aided in the decision to use the Yahoo! Maps API which implements the Flash platform.

2.2 Preliminary Design

The preliminary design was soon incompatible with the implemented system as a whole. Because new features were constantly being added to the project, the initial design became more outdated at each step the project matured. Classes and objects began to contain circular-dependencies causing both troubleshooting and project understanding to become nearly impossible. The project had no version control system in place, which made reverting to previous versions nearly impossible except for saved copies in .zip files. These two major reasons, among others, brought the development of placeMap to a necessary decision point.

The two options were to try and fix the current version, or to completely redesign and re-implement everything done so far. The easy answer was to not throw all work away. However, the smart choice was to start over with an intelligent design using the knowledge gained from the first attempt. Before jumping into the creation of a new design, several users participated in an ad hoc questioning session about the usability and feel of the user interface. Using the information from the first effort and the input from users, the new design was structured better and visually more appealing.

2.3 Reimplementation – A New Design

The new, current architecture of the system still lies on the Yahoo! Maps API, enabling placeMap to be lightweight and visually appealing. The placeMap project as a whole can be categorized into three distinct components: client software application, Jabber[v] server, and web services. The client software application is by far the largest

component, as it contains all of the tools for user interaction with the system. The Jabber server handles the chat exchange between users, providing some security in that user clients do not talk directly to one another. The web services are a set of ASP.NET tools that allow easy reference to data manipulation and extraction methods.

The class structure intelligently enforces a lack of circular dependencies and a central data-based architecture. Figure 2 below depicts the modular dependency diagram in which you can see the consistent hierarchy. Figure 1 shows the various Data Brokers that the Yahoo Map depends on for information to display. Each of the four brokers is responsible for a specialized type of data storage and recollection, namely event, person, location, and user data. The individual classes will be described later in section 3.

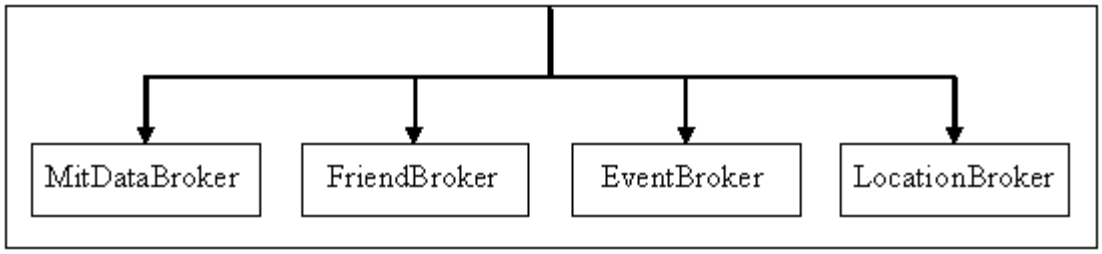

Figure 1. Diagram of the Data Brokers.

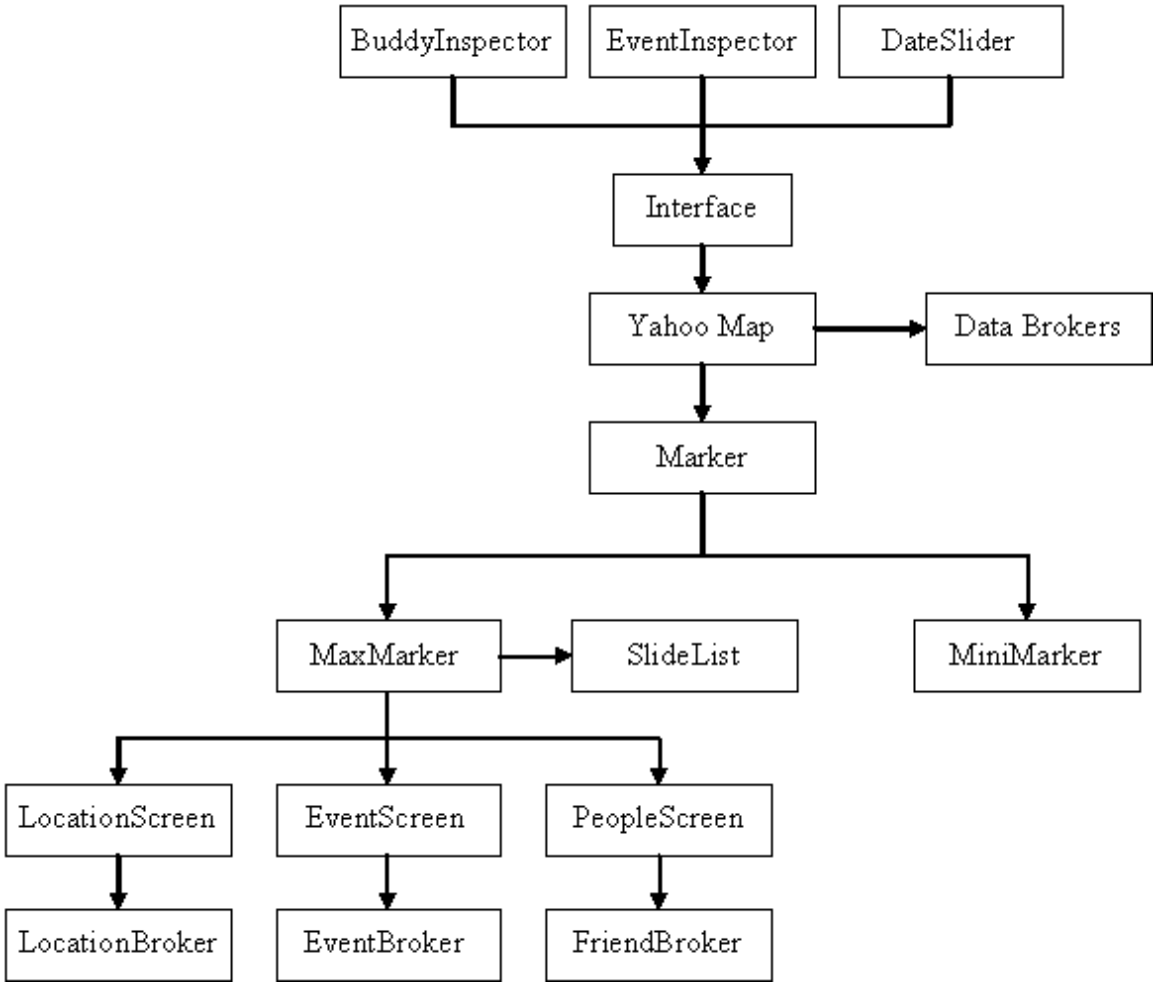

Figure 2. Module dependency diagram. Note that no circular dependencies exist

3. Architecture

3.1 Client Software Application

The client software application was programmed in ActionScript[vi] class files, as opposed to within the movie time frames. The main extension that placeMap adds to Yahoo!'s Map API is the novel marker described below.

3.1.1 Marker

The first figure, Figure 3, shows a screenshot of an unexpanded marker. The program displays this as an "icon" to let the user know of the underlying information. The program tinted one of the markers maroon to imply to the user that this is the user's current geographical location. Simply rolling over the marker with the mouse will cause the marker to animate into the expanded state. Figure 4 shows a brief glimpse of the marker expanding from small to enlarged views.

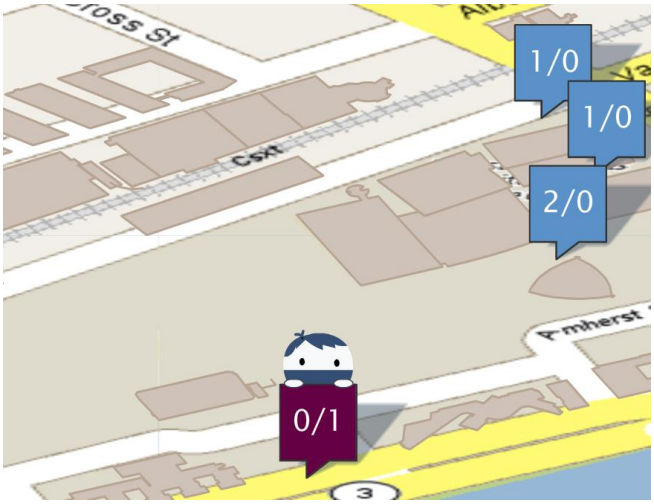

Figure 3. Unexpanded marker

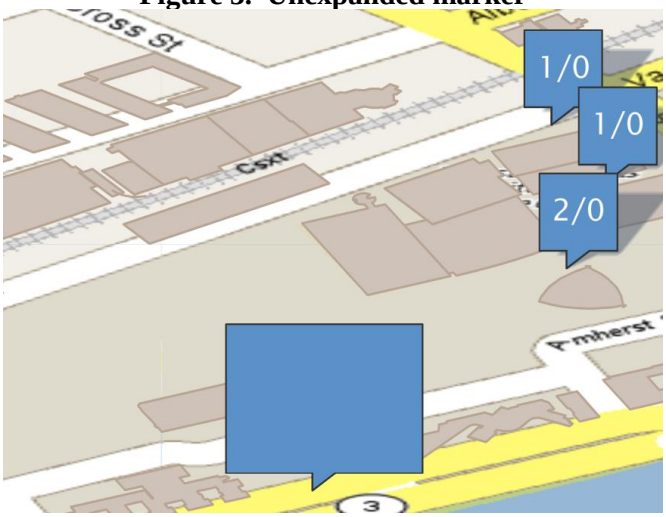

Figure 4. Marker in transitional state during animation

Once the animation is complete, the expanded marker is visible to the user and depicts information relevant to the selected location. In the example of Figure 5, a user is at the location in question. Since this is actually the current user, only the basic information is provided. In the case that different person is selected, a chat window overrides the information screen and allows the user to chat with the other person via the Jabber instant messaging system. When the marker does not have the focus of the mouse, the small marker "throbs" to alert the user about an incoming message. This throb effect is achieved by animating the marker to become bigger and smaller repetitively until the user checks the message. By using this visually pulsating effect, the location not only catches one's attention, but actually appears to be reaching out to the user. In this respect, instant messaging is no longer communication with a person at an unknown place, but a conversation between places.

Figure 5. Marker in expanded state with people view open

When a person is not at the location in question, the events screen is shown, which displays the details of an event. In Figure 6, a marker is expanded showing a

prayer and bible study at the student center. The image displayed as its icon is a randomly generated image from the web, acquired from Yahoo! Image Search.[vii] Through the service, images matching event keywords are returned and a random picture is selected as the icon for the event. This random generation yielded some problems which will be discussed later in the paper.

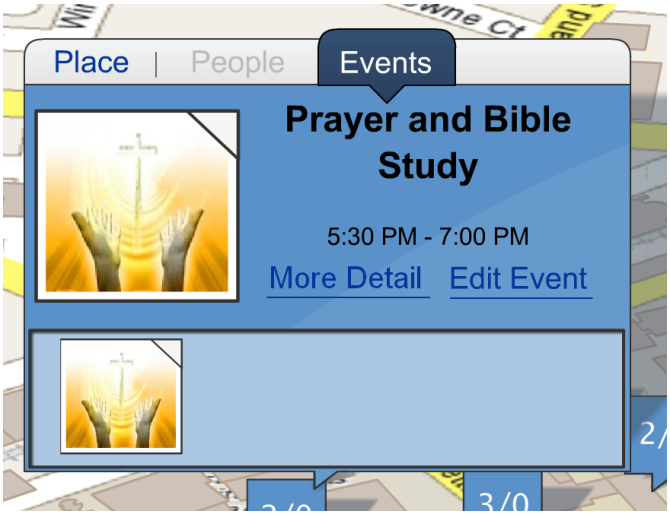

Figure 6. Marker in expanded state with event view open

3.1.2 DataBrokers

In this project, a data broker is defined as an object that's sole responsibility is to collect, store, format, and distribute data between a web service and the internal application. placeMap has four data brokers – FriendBroker, LocationBroker, EventBroker, and MitDataBroker. The MitDataBroker is just a container class to hold the data brokers, while the other three perform all tasks related to retrieving information from the centralized server, and storing it internally to the application for efficient use.

3.1.2.a FriendBroker

The FriendBroker class is responsible for handling all of the user information, including friends, current user, and potential friends. In this respect, it is dependent on the Person class, and inherently on the Location class. The FriendBroker allows for such functionality as getting a list of friends, adding/removing friends, and updating user information. One of the most important functions of the FriendBroker is the ability to determine both location of the user, as well as the locations of his friends. The correct placement of all person markers is dependant on this information.

3.1.2.b LocationBroker

The LocationBroker handles all information regarding locations stored in the database memory. This is heavily dependent on the Location class that stores all information about a given Location. Functions the LocationBroker is highly needed for are the methods to retrieve all buildings on the map, and reverse lookup of buildings based on building number. An example of this would be placing a friend on the map after just receiving their data.

3.1.2.c EventBroker

Possibly the most important of the data brokers, the EventBroker communicates with the database to retrieve and process the events from the database. EventBroker handles all events as it retrieves, sorts by location, and displays the events on the map. This forms the basic functionality of placeMap.

3.1.3 User Interface

The user interface has three main components other than the actual Yahoo! Map. The first is the DateSlider that enables the user to view events that occur on different days. Next is BuddyInspector which acts as a "buddy list" that shows current users and the online/offline status as well as allows a user to add or remove friends. Last is the EventInspector which is the most complicated, yet most informative of the group.

3.1.3.a DateSlider

In order to view days other than the current one, the DateSlider provides the functionality to select a desired day to view. Its default and initial position is "closed" in the bottom right hand corner of the screen, simply showing the date that is currently being viewed. If a user clicks on the DateSlider, it slides to the left, expanding to show six days forward and backward from the current date. Clicking the date that you want will slide the DateSlider back to its compact state in the corner and refresh the view on the map to display the events on the newly selected date. Figure 7 shows an example of the DateSlider in expanded view with Thursday the 18th selected.

Figure 7. Screenshot of the DateSlider when expanded

3.1.3.b BuddyInspector

As is typical in a program that allows instant messaging, a user needs to know the online/offline status of his friends, and how to reach them. The BuddyInspector handles this need by displaying the current user and his friends along with their locations. The current user is tinted maroon to avoid confusion between friends. When a user is offline,

they are smaller, faded out, and italicized, unlike the online state that is black, non-italicized and of bigger font. Clicking on a user centers the map on their location so the user can easily find them or talk to them, as well as opening the Billboard to display all of the known information about the user. The BuddyInspector also has an options menu that allows the user to add or remove friends from his buddy list.

Figure 8. Screenshot of the BuddyInspector

3.1.3.c EventInspector

When acquiring events from online sources, not all have a mapped location on the system. This means that not all events are visible as markers on the map. To assure that a user can find any event that is in our system, the EventInspector was added to the map. As is shown in Figure 9, the EventInspector concisely presents the title of the event along with the start and end times. Because ease of use is of utmost priority, a tooltip-like explanation window follows the cursor and contains the description of the event the mouse is hovering over. Clicking on an event will have a two-fold effect similar to that of the BuddyInspector. First, the map will center on the location of that event if the

system knows it. Secondly, the Billboard will display all of the known information about the event.

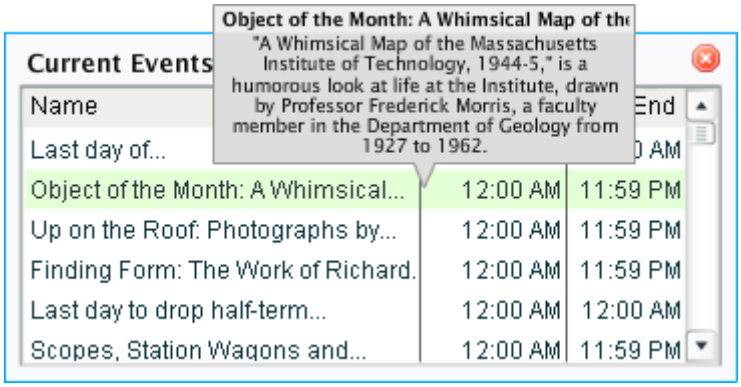

Figure 9. Screenshot of the EventInspector

3.2 Jabber Chat System

Jabber is an open source, server-based protocol of instant messaging between users. Its communication is XML-based, which allows it to be easily extended to handle customized message types. The key factor in deciding to use a Jabber server as the means for inter-user chat was the ease of creating a private, or public, network for users to join. This capability is one of the highlights of the Jabber project, allowing anyone to setup a server, create users, and control who can communicate and how. The open ended possibility of linking networks together is even more exciting, as this could become a future goal of the placeMap project.

Based on the Jabber protocol, XIFF is an open source chat client written in Flash ActionScript. Because it is open source, its code was modified to ease the integration of XIFF into the placeMap architecture. The chat-handling module distributed with XIFF was too bulky and bogged down with many unnecessary features, as well as lacking certain features we need. For this reason, an entire new chat-handler was built, which controls all communication between placeMap and the Jabber server.

3.3 Web Services

3.3.1.a SOAP-based web services

The web services of the placeMap project are an integral part of the system as a whole. They not only create an easy method for centralizing user information, but abstract away the database completely to the client application. From the viewpoint of the client application, the web services are merely another method that is available to call. Written in C# .NET, the web services are an extremely fast communication link to extract database information. To interface with Flash, the web services use XML-based SOAP calls to correspond between computers. Although typically slower than binary protocols, the SOAP messages provide an extremely easy method for transfer of small data messages.

3.3.1.b Billboard – website driven information

One of the most unique features of placeMap is its look and feel. One goal was to make the user as comfortable in the application environment as they would be in a normal setting. Several visual modifications assist in this; however, the Billboard is the most useful of them. Figure 10 shows the opening screen of the Billboard which alerts the user to some random events today and his friends that are currently logged in. A more prominent purpose of the Billboard is to "advertise" the information about a selected person, place, or event. As previously stated, when a user or event is selected in their respective inspector, a webpage displaying all of their information is expanded in the Billboard. Figure 11 displays an example screenshot of the previously shown event.

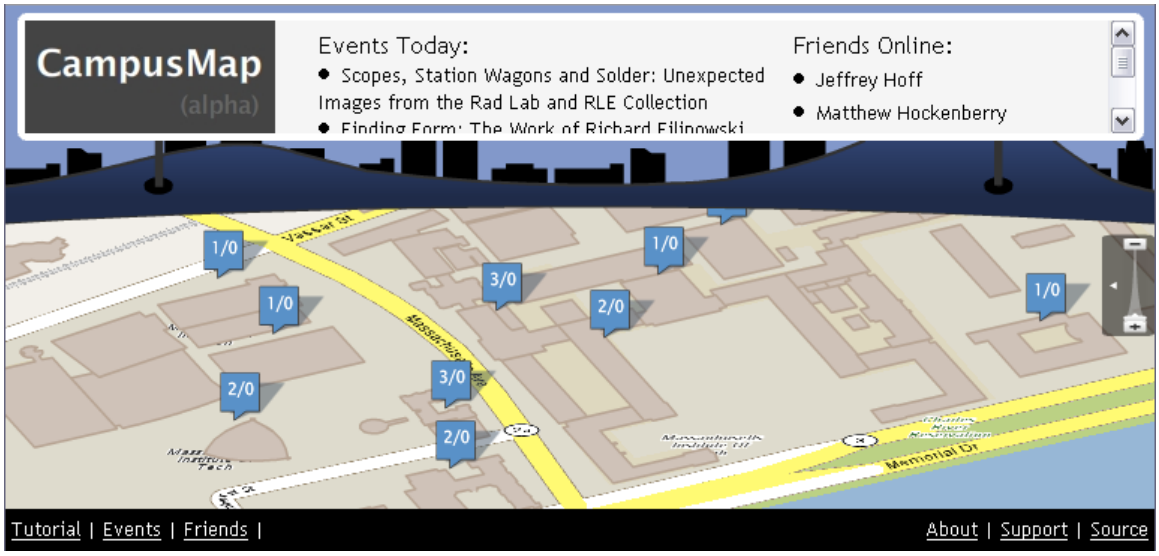

Figure 10. Screenshot showing the Billboard

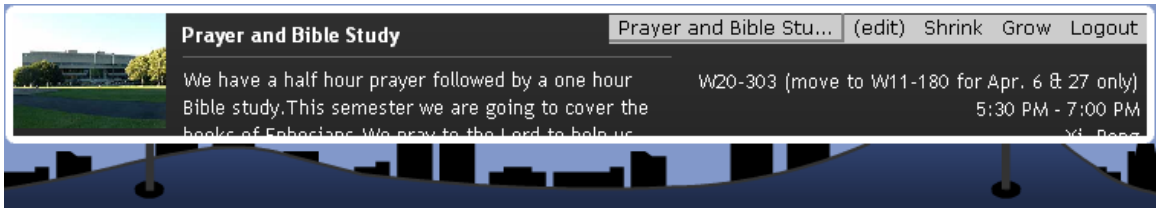

Figure 11. Screenshot showing the Billboard

4. Possible Applications

There are many possible groups that would use the placeMap project and benefit from its uses. The primary focus group for future use was universities, as the initial application was made for the Massachusetts Institute of Technology. Events on campus have multiple media on which they are conveyed such as poster board, radio, websites, and emails. The placeMap application allows all events to be gathered together in one space, visually placing them at their respective locations. Other than events, the project is set up to become an information exchange, not only of where and when events are happening, but of who is going and what they think about the event is.

Aside from individual groups, major companies are highly interested in the project. Pepsi and Schlumberger are attracted to the idea that they could visually track

vehicles, equipment, and people through one application, as well as use it as a medium for meeting scheduling. This interest shows the diversity of the possible applications this project can accommodate.

5. Conclusions

By looking at how a user would like to have a map tailored to their needs, placeMap contains a new type of map altogether. The user-based map wraps itself around the user, as opposed to the user conforming to the maps perspective. Minor details such as perspective skewing allow user to feel more comfortable in the application environment as it is more natural. The most exciting conclusion is that placeMap shows the ability for user based mapping to work and be accessible to many types of applications. The many possible applications realized and suggested attest to this conclusion.

6. Future Work

The placeMap project is now a good platform for user research on human-place interaction, human perception of place, and how place affects human decision making. Possible future work should include human studies in these areas. Also, broadening the aspect view of the project would probably prove to be beneficial to study. Currently, the project is user-based, but what would happen if it became group-based? Integrating GPS would enable real-time tracking of objects through the world, however because of current licensing restrictions, Yahoo! does not allow this.

7. Acknowledgements

I would like to thank several people for their help along the way in this project. First, Shawn Sullivan connected me with the Context Aware Computing group at MIT's

Media Lab, where I did my project. Secondly, Matthew Hockenberry was a great friend and mentor throughout the entire project. I am truly privileged to be able to work with on the placeMap project. Lastly, Professor Ted Selker trusted me enough to give me responsibilities over such big project. For that, his guidance, and all the opportunities that he has provided me with, I thank him.

8. Endnotes

[i] Adobe (formerly Macromedia) Flash® is a development tool that was used to create the animated online website. Macromedia Flash is produced and trademarked by Adobe Systems Incorporated [http://www.adobe.com].

[ii] Adobe (formerly Macromedia) Flash Player® is the free client to view Flash files.

[iii] JavaScript™ is a scripting language used mostly in websites to aid in user event handling and is a registered trademark of Sun Microsystems, Inc.

[iv] Ajax, short for Asynchronous JavaScript and XML (though its creator Jesse James defends that is not an acronym), is a technique used to use asynchronous calls to a server to reload only small parts instead of the webpage as a whole, saving both time and transfer traffic. Through this Ajax enhances the user experience by making data transition appear seamless.

[v] Jabber® is an open source, server-based method of instant text-messaging using XML-based communication.

[vi] ActionScript is the scripting language used to program Flash. Its roots are based on the same standard as JavaScript, assisting in fluency in the other if one is known.

[vii] Yahoo! Image Search is a web search utility to find images on the web with certain descriptive keywords. For more information or to use the free service, please visit [http://search.yahoo.com/images].