GIS-BASED INTERACTIVE TOOL TO MAP THE EVOLUTION OF
AMERICAN AUTOMOBILE INDUSTRY

A Thesis
Presented to the
Faculty of
San Diego State University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
in
Computer Science

by
Saran Teja Reddy Alla
Spring 2015
SAN DIEGO STATE UNIVERSITY

The Undersigned Faculty Committee Approves the

Thesis of Saran Teja Reddy Alla:

GIS-Based Interactive Tool to Map the Evolution of American Automobile Industry

__________________________
Carl Eckberg, Chair
Department of Computer Science

__________________________
John Donald
Department of Computer Science

__________________________
Subrata Bhattacharjee
Department of Mechanical Engineering

Feb 19, 2015
Approval Date
Copyright © 2015
by
Saran Teja Reddy Alla
All Rights Reserved
ABSTRACT OF THE THESIS

GIS-Based Interactive Tool to Map Evolution of American Automobile Industry
by
Master of Science in Computer Science
Saran Teja Reddy Alla
San Diego State University, 2015

This is a GIS based tool for showcasing the history of the American Automobile Industry starting from 1900 to post Second World War. The tool is developed using simple language and is flexible to allow future enhancements.

The application consists of numerous images and textual information, and also some links that can be used by this software which depict various automobile companies including the steam automobile manufacturers at the American automobile inception. It covers various topics on evolution of different technologies in automobiles. All the automobile industries locations are shown on GIS (Geographical Information Science) maps.

The tool can be tweaked as per user necessities and is created utilizing JAVA and GIS Technology. The user interface is made utilizing JAVA and MOJO, which help promote learning and understanding of the topic. The user interface in this tool includes the usual zoom-in, zoom-out, as well as legend editing, printing, images, layer control and many menu items.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>CHAPTER</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td>iv</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>viii</td>
</tr>
<tr>
<td>ACKNOWLEDGEMENTS</td>
<td>x</td>
</tr>
<tr>
<td>1 INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Motivation</td>
<td>1</td>
</tr>
<tr>
<td>1.2 High Level Description</td>
<td>1</td>
</tr>
<tr>
<td>2 TECHNOLOGY</td>
<td>3</td>
</tr>
<tr>
<td>2.1 Java Standard Edition</td>
<td>3</td>
</tr>
<tr>
<td>2.1.1 Object Oriented Programming (OOP)</td>
<td>4</td>
</tr>
<tr>
<td>2.1.2 Enhanced Error Prevention and Checking</td>
<td>4</td>
</tr>
<tr>
<td>2.1.3 Threaded</td>
<td>4</td>
</tr>
<tr>
<td>2.2 Map Objects Java Edition (MOJO23)</td>
<td>4</td>
</tr>
<tr>
<td>2.3 Web Technologies (HTML, CSS, JAVASCRIPT)</td>
<td>5</td>
</tr>
<tr>
<td>2.4 Eclipse</td>
<td>6</td>
</tr>
<tr>
<td>3 REQUIREMENTS</td>
<td>8</td>
</tr>
<tr>
<td>3.1 Data Requirements</td>
<td>8</td>
</tr>
<tr>
<td>3.2 Functional Requirements</td>
<td>8</td>
</tr>
<tr>
<td>3.3 Platform Requirements</td>
<td>9</td>
</tr>
<tr>
<td>4 INFORMATION GATHERING AND ANALYSIS</td>
<td>10</td>
</tr>
<tr>
<td>4.1 Data</td>
<td>10</td>
</tr>
<tr>
<td>4.2 Shape Files</td>
<td>11</td>
</tr>
<tr>
<td>5 SOFTWARE IMPLEMENTATION</td>
<td>12</td>
</tr>
<tr>
<td>5.1 Application Start Page</td>
<td>12</td>
</tr>
<tr>
<td>5.2 Point Layers</td>
<td>13</td>
</tr>
<tr>
<td>5.3 Application’s Tool Bars</td>
<td>13</td>
</tr>
<tr>
<td>5.3.1 Zoom/Pan Toolbar</td>
<td>15</td>
</tr>
</tbody>
</table>
5.3.1.1 Previous Extent..........................15
5.3.1.2 Next Extent..............................16
5.3.1.3 Zoom to Active Layer......................16
5.3.1.4 Zoom to Full Extent..........................16
5.3.1.5 Zoom In..................................16
5.3.1.6 Zoom Out..................................17
5.3.1.7 Pan..................................17
5.3.1.8 Pan in One Direction......................17
5.3.1.9 Identify................................18
5.3.2 Selection Toolbar..........................18
5.3.2.1 Find................................18
5.3.2.2 Clear All Selection......................19
5.3.2.3 Attributes...............................19
5.3.3 Customized Toolbar.........................19
5.3.3.1 Print................................19
5.3.3.2 Add Layer...............................20
5.3.3.3 Remove Layer.............................21
5.3.3.4 Cursor (Arrow)............................21
5.3.3.5 Hotlink..................................21
5.3.3.6 Create Polyline Layer / XY Tool..............22
5.4 Application Menu Bar..........................23
5.4.1 Add Layer................................25
5.4.2 Remove Layer...............................25
5.4.3 Legend Editor................................25
5.4.3.1 Attribute Table............................25
5.4.3.2 Create Layer from Selection...............26
5.4.4 Layer Control: Promote/Demote Layer.............26
5.4.5 Help Topics................................28
5.4.6 Contact Us................................28
6 AMERICAN AUTOMOBILE LAYERS.....................31
6.1 Adding a Layer................................31
6.2 Automobiles Layer............................32
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.3</td>
<td>Steam Automobiles Layer</td>
<td>33</td>
</tr>
<tr>
<td>7</td>
<td>SUMMARY AND OBSTACLES</td>
<td>36</td>
</tr>
<tr>
<td>8</td>
<td>FUTURE WORK</td>
<td>37</td>
</tr>
<tr>
<td>9</td>
<td>TESTING SECTION</td>
<td>38</td>
</tr>
<tr>
<td></td>
<td>REFERENCES</td>
<td>39</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

PAGE

Figure 2.1. Adding Jar files to Eclipse ................................................................. 7
Figure 4.1. CSV files for the “Car Companies Layer” ............................................. 11
Figure 5.1. Application launch ........................................................................... 12
Figure 5.2. CSV files for automobile companies layer ........................................ 14
Figure 5.3. The launch of the tool with one of the layers added .......................... 15
Figure 5.4. Zoom/pan toolbar ............................................................................. 15
Figure 5.5. Previous extent tool button ............................................................... 16
Figure 5.6. Next extent tool button ..................................................................... 16
Figure 5.7. Zoom to active layer tool button ....................................................... 16
Figure 5.8. Zoom to full extent tool button .......................................................... 16
Figure 5.9. Zoom in tool button ......................................................................... 17
Figure 5.10. Zoom out tool button ...................................................................... 17
Figure 5.11. Pan tool button .............................................................................. 17
Figure 5.12. Pan in one direction tool button ..................................................... 17
Figure 5.13. Pan in one direction’s drop down list ........................................... 18
Figure 5.14. Identify tool button ....................................................................... 18
Figure 5.15. Selection toolbar .......................................................................... 18
Figure 5.16. Find tool button ............................................................................ 19
Figure 5.17. Clear all selection tool button ....................................................... 19
Figure 5.18. Attributes tool button .................................................................. 19
Figure 5.19. Print tool button ........................................................................... 19
Figure 5.20. Add layer tool button .................................................................... 20
Figure 5.21. Adding the shape file to the map ................................................... 20
Figure 5.22. Remove layer tool button .............................................................. 21
Figure 5.23. Cursor (arrow) tool button ............................................................ 21
Figure 5.24. Hotlink tool button ..................................................................... 21
Figure 5.25. Webpage for Hotlink tool button .................................................. 22
Figure 5.26. XY tool button. ................................................................. 22
Figure 5.27. Selecting the steam cars CSV files. ........................................ 23
Figure 5.28. File menu................................................................. 23
Figure 5.29. Theme menu............................................................. 24
Figure 5.30. Layer control menu...................................................... 24
Figure 5.31. Automobile components menu ......................................... 24
Figure 5.32. Help menu .............................................................. 24
Figure 5.33. Remove layer................................................................ 25
Figure 5.34. Legend editor tool button................................................. 25
Figure 5.35. Layers selected for the table of contents.............................. 26
Figure 5.36. Steam properties legend editor ......................................... 27
Figure 5.37. Attribute table............................................................ 27
Figure 5.38. Create layer from selection............................................... 27
Figure 5.39. Creating a layer on the map using attribute table ................. 28
Figure 5.40. Help topics. ............................................................... 29
Figure 5.41. Contacting the developer of the application.......................... 30
Figure 6.1. Application started using Java IDE. ...................................... 31
Figure 6.2. Dialog after clicking XY button........................................... 32
Figure 6.3. CSV file of layer automobile companies................................. 33
Figure 6.4. After adding a layer ........................................................ 34
Figure 6.5. HTML page, after using hotlink tool. .................................... 34
Figure 6.6. HTML page of Buckeye Manufacturing Company of automobile layer. ........ 35
Figure 6.7. HTML page of Stanley Steamer ........................................ 35
ACKNOWLEDGEMENTS

I want to thank Dr. Carl Eckberg for consenting to be my supervisor and for his proposal to tackle this undertaking. His help, learning, persistence and consultation were pivotal for the finishing of this work.

I might likewise want to thank Prof. John D Donald and Prof. Subrata Bhattacharjee for consenting to be a piece of my Thesis Committee and for taking the time to peruse this work.
CHAPTER 1

INTRODUCTION

1.1 MOTIVATION

Certain topics are common and almost obsessive, for people inclined towards hobby collecting information, which is a large group of people.

Two such, mostly with males are the civil war and automobiles. So both these topics are temptations for map based multimedia story telling. Hobbyists are called buffs, as in civil war buffs or vintage car buffs. An other thesis was done about the civil war, and this vintage car thesis deals with may be the second most popular history based hobby topic in the united state of America.

The civil war is by its nature map based, but both topics lend themselves to multimedia dynamic maps at some level and a well designed information tool can give a coherent view of a major era of history which coalesces information which would otherwise be scattered and hard to piece together.

1.2 HIGH LEVEL DESCRIPTION

The Main purpose for this GIS application is to give an intuitive learning tool for the American automobile industry from the early to mid 20th century, or what we now refer to as (very) classic cars. The tool clarifies advances in diverse automobile parts used in automobiles during their early evolution. It likewise covers different steam autos, and the companies making them. It covers all the major activities at the time.

This GIS tool represents American manufactured automobiles with map layers, webpages, pictures and textual data. All the data is visually shown in HTML pages. Although the majority of the well-known car companies have been detailed, there are still some missing connections, which can be added to the tool at a later point of time, also mentioned in the Future Improvements chapter. The tool is made in such a way that it is easy for users to change and alter the layers, and webpages according to their requirements. The users likewise have the option of adding extra data, and HTML pages to the current layers.
The application was developed as a Geographical Information System (GIS) in order to present to the user different types of data in the form of interactive maps [1]. The steps followed in the development process of this application are explained in the coming chapters and they go from requirements to future work.

The second chapter in this document portrays all the technologies utilized as a part of the usage of the tool, highlighting the fundamental motivation behind why they were picked.

The “software requirements” chapter will go over the methodology followed for the collection of the requirements for this project. It will list, describe and classify all of them.

The fourth chapter, “Data Collection and Analysis”, will cover the tools and methods used for the collection and preparation of the data used to create all the map layers presented in this application.

The fifth chapter, “Software Implementation”, talks about how each part of this tool was implemented. It contains several screen shots and some code snippets that will guide the reader in the understanding of this process.

The last two chapters of this document, “Summary and Obstacles” and “Future Work”, will include lessons learned during this process, the obstacles or difficulties found in the implementation stage and possible future enhancements to the developed application as well as areas of research.
CHAPTER 2

TECHNOLOGY

Below are listed the major technologies used in developing the tool.

- **Programming language**: Java(TM) SE Runtime Environment.
- **Application Programming Interface**: Map Objects Java Edition.
- **Web Technologies**: Hypertext Markup Language (HTML), Cascade Style Sheets (CSS).
- **Integrated Development Environment (IDE)**: Eclipse

### 2.1 Java Standard Edition

Java Standard Edition is a platform used by many servers as well as desktop users for development and deployment of portable applications. Typically Java is compiled to byte code (class file) that can run on any Java Virtual Machine (JVM) regardless of computer architecture. So this feature gives you platform independence. This means that code that runs on one platform does not need to be recompiled to run on another. This platform independence feature makes this language ideal for the development for this project.

Java is licensed under the GNU General public license therefore making it free/open source software and a very cost-effective option. Another added advantage of using the Java language for developing this project is the ease with which we can deploy Java applications. Java uses JAR (Java Archive) file format which enables you to bundle multiple files in a single archive file that contains all the class files (compiled code) and all the auxiliary resources associated with your application. Most importantly it eliminates the dependence on windows installers, which in turn makes your applications portable.

Java has evolved with versions 1.1 through 1.8, and this tool should work with 1.6 and up. Some specific features of java are important to this thesis are object oriented, enhanced error prevention and checking exceptions, threaded( gis applications are usually threaded and mojo is), easily deployed( executables jar files) [2].
2.1.1 Object Oriented Programming (OOP)

Java is a computer programming language that is concurrent, class-based and object-oriented. The advantages of object oriented software development are shown below:

- Modular development of code, which leads to easy maintenance and modification.
- Reusability of code.
- Improved reliability and flexibility of code.
- Increased understanding of code.

Object-oriented programming contains many significant features, such as encapsulation, inheritance, polymorphism and abstraction [2].

2.1.2 Enhanced Error Prevention and Checking

Java exception handling enables Java applications to handle errors sensibly. When an error occurs in a Java program it usually results in an exception being thrown. How you throw, catch and handle these exception matters. There are several different ways to do so. Not all are equally efficient and fail-safe [2].

2.1.3 Threaded

Thread is basically a lightweight sub-process, a smallest unit of processing. Multiprocessing and multithreading, both are used to achieve multitasking [2]. But we use multithreading than multiprocessing because threads share a common memory area. They don't allocate separate memory area so saves memory, and context switching between the threads takes less time than process. GIS applications are threaded and MOJO is.

2.2 Map Objects Java Edition (MOJO23)

This API for Java is a product of ESRI (Environmental Systems Research Institute), headquartered in Redlands, California.

The version used for this project was 2.3. This version package contains several libraries developed in Java that a programmer can use to create as well as customize Java applications in the area of Geographic Information System (GIS) which covers systems designed to capture, store, manipulate, analyze, manage and present all types of geographical data [3].
One functionality this API provides the programmer with is the possibility of displaying and manipulating maps as one wants (based on geographical data) and also performing queries on its spatial information [4].

Map Objects also gives you the freedom of merging two java-based Graphical User Interface (GUI) frameworks, AWT (Abstract Windowing Toolkit) and Swing. The use of these powerful frameworks gives the developer the option of customizing the GUI elements provided by the Map Objects API and creating new ones.

As the Map Objects library was written in the Java language it complements the language chosen for this project.

MOJO contains 17 Jar files which effectively extend the java language to do GIS. The hundreds of classes and thousand of methods in these jar files cause MOJO to be viewable as a “Component Architecture”, akin to Visual Basic. Indeed ESRI has a Visual Basic version Map Objects [5].

2.3 WEB TECHNOLOGIES (HTML, CSS, JAVASCRIPT)

HTML (hyper text markup language) is the main markup language used for creating and developing web pages in a web browser. Web browsers also refer to Cascading Style Sheets (CSS) in order to define the look and layout of text and other material that is part of documents written in HTML.

HTML5 is a programming markup language to create apps and web page content that conform to mobile and desktop browsers. HTML5 was an advancement of legacy HTML, incorporating a variety of new features including advanced streaming video and audio features as well as more integration with device specific tools and additional storage capacity. HTML5 is often used for producing the same application on multiple devices, but that approach comes with both benefits and challenges [6].

JavaScript is a very widely used programming language that can be used for HTML, web, computers, servers, tablets and much more. JavaScript is a prototyped-based scripting language, which makes it dynamic, weakly typed, and has first-class functions. It is primarily used in the form of a client side script language, which helps enhance user interfaces and dynamic websites.
The above three languages were used in the creation of the web pages that are an important part of the user application.

HTML, CSS and JavaScript are open standards and interpreted to a certain degree, in most of today's web browsers in use.

One advantage of CSS is that, it can be used to give the same look and feel to many web pages, thus saving repetitive code in the webpages.

2.4 Eclipse

Eclipse is an integrated development environment (IDE). It consists of a base workspace and an extensible plug-in system for customizing the environment. It is mostly written in Java and used for developing applications. Since we can have various plug-ins, we can use Eclipse to develop applications in other languages like C, C++, ABAP, Ada, Fortran, Haskell, JavaScript, Lasso, Natural, Perl, PHP to name a few. It is also used to develop packages for the software mathematics. Development environments include the Eclipse Java development tools (JDT) java and scala, Eclipse PDT for PHP, among others. The initial codebase has its origin from IBM Visual age. The Eclipse software development kit (SDK), which includes the java development tools, is meant for Java developers. Users can use its abilities extendedly by installing plug-ins written for the Eclipse platform like development toolkits for other programming languages, and can write and contribute their own plug-in modules. In using Eclipse for MOJO, we have to use a minimum two of the MOJO Jar files to eclipse [7].

Figure 2.1 is the screenshot of adding MOJO Jar files to the eclipse
Figure 2.1. Adding Jar files to Eclipse.
CHAPTER 3

REQUIREMENTS

The requirements in my thesis are classified into 3 types, data requirements, functional requirements and software and platform requirements. The major contributions in the above 3 mentioned requirements were by Professor. Carl Eckberg.

3.1 DATA REQUIREMENTS

In order to accomplish, this project enormous amounts of data were required. All this data was gathered from books, websites, visiting a vintage car show, and discussions with Prof Carl Eckberg [5] and a few other people.

There are two layers in my tool. The primary layer is the automobiles layer and the secondary layer is the steam automakers. The majority of the data included in these layers was gathered from Wikipedia and different online resources.

Also, speaking to people attending the vintage car shows helped me collect the data for the evolution of automobiles. Also discussions with various people owning a vintage car have given a good insight of the automobiles industry, parts used in its early stages. Various inputs provided by Professor Eckberg have also made me to dive deep and understand the early days in the automobile industry.

3.2 FUNCTIONAL REQUIREMENTS

Functional requirements of this GIS tool were gathered with the help of Prof. Carl Eckberg. It refers to the functional details of the tool and the Graphical User Interface (GUI) requirements. The functional requirements added in this Tool are:

- The software should come up with a default map of world.
- The software shall provide standard tools like zoom and pan.
- The software is capable of adding/removing layers. This is suitable for possible homework assignments.
- The places and points of interest are identified with the help of an identification tool.
- A custom toolbar including a print tool and a hotlink tool.
• The software comes with the Print option.
• Different layers are shown when the tool is launched and the user should be capable of enabling and disabling different layers in the tool.
• The soft link tool is to be used to get detailed information about particular automobile companies via HTML.
• In addition a number of menus are required, including a Help menu to inform the user about how to use the dynamic map features like the hotlink tool and the Legend editor.

3.3 Platform Requirements

Platform requirements refer to the development programming language used to develop this tool. This information is collected from Professor Dr. Carl Eckberg. The entire set up starts with setting up the windows platform as it has its own advantages. The user need not have knowledge about IDE’s. The user just needs to have knowledge about command prompts to work with GIS files. The different technologies used in developing this GIS tool are the following:

• JAVA as a programming language
• MOJO (Map Object Java Editor) as GIS API
• Web Technology – HTML5 & CSS

Finally for deployment purposes, a Zip File must be made available at an SDSU website, containing and executable jar file for launching the application.
CHAPTER 4

INFORMATION GATHERING AND ANALYSIS

4.1 DATA

Before gathering the data related to my thesis, I needed to read and understand the complete history of the American automobile industry from its inception. American automobile industry is very vast. Going with the vastness I have classified the automobile companies into 2 types: automobiles that run on gasoline, automobiles that run on steam.

In order to gather information I read the books “Treasury of Early American Automobiles 1877-1925” by Floyd Clymer [8] and “American Automobile Trademarks 1900-1960” by C.H. Wendel [9]. For the implementation of this project a great amount of data was required [10]. The sources for all the data collected for the implementation of this application are listed and briefly discussed in the next few sections.

The steps below were followed in the creation of all the map layers.

- Collection of geographical coordinates for the physical features of all the maps presented and to collect the latitude and longitude of the places presented in the project http://www.worldatlas.com/aatlas/latitude_and_longitude_finder.html was used.

- CSV (Comma-Separated Values) files were generated using the previous collected coordinates. CSV files store tabular data (numbers and data) in plain text format. They consist of any number of records, separated by line breaks of some kind; each record consists of fields, separated by some character or string, most commonly a literal comma or tab. See Figure 4.1. Similar CSV files, like the one presented in Figure 4.1 were prepared for all the map layers presented as part of this application. Implementation of Java code was used to read the CSV files and generate the appropriate type of layer (point, polyline or polygon) in SHP format (shape file) in order to be displayed using the Map Objects API. The shape file format is a geospatial vector data format for GIS software that was developed by ESRI. Note: The XY tool in the application can add layers based on CSV files created by the user.
Figure 4.1. CSV files for the “Car Companies Layer”.

4.2 SHAPE FILES

ESRI created the shape file as a geographical layer format. A shape file so called in reality must have three required files, e.g. Country.shp, Country.shx and Countries.dbf [5].

The .shp has locational information

The .shx allows a quick index into .shp. Since country boundary descriptions very greatly in length

The .dbf file is an old format useable by database products and Excel. It contains non-locational attributes of the GIS Layer. It can be created from CSV files by a process described elsewhere in this thesis.
CHAPTER 5
SOFTWARE IMPLEMENTATION

In this chapter the process of software implementation will be discussed. Several screen shots are presented to provide clear understanding and a visual guide to the user.

5.1 APPLICATION START PAGE

This is the page displayed as soon you launch the application (Figure 5.1). The main page or the launch page of the application is a window with a map of the world and has the following contents in it.

![Application Launch](image)

*Figure 5.1. Application launch.*

The “Table of Contents” is located to the left of the map area and it contains a list of all the map’s layers currently displayed, it also serves as the legend’s definition that identifies
each feature on the map. The “Status Bar” at the bottom displays the coordinates in the map to which the cursor currently points, and it is dynamically updated as the mouse’s pointer moves.

One important reason to open with a world map is to allow future extensions, including autos imported from Europe and Asia, or exported to them, during the time frame used in the tool.

5.2 POINT LAYERS

For the creation of the point layers, the method described in the chapter “Data Collection and Analysis” was followed. Part of the code used to parse the CSV files, like the one presented, can be seen in Figure 5.2.

The code snippet seen in Figure 5.2 was modified depending on the point layer that was being created, in order to process the appropriate data for it. In this case, the CSV file being processed was the one for the automobiles, so the data to be parsed was the GPS coordinate (longitude, latitude), name of the automobile company and the HTML page associated with the company which holds more information. The launch of the tool for the CSV file is shown in Figure 5.3.

Similarly we can also the other layers to the Automobile layer and project it on the GIS Tool.

5.3 APPLICATION’S TOOL BARS

This application has the following main toolbars. The Zoom/Pan Toolbar and the Selection Toolbar are the two standard ones that should be present, in some form, in most GIS applications that provide the user with an interactive map. These two toolbars are supported by the Map Objects API and can be customized, in some way, for a particular GIS application. The third toolbar was specifically created and customized for this project. In the next few sections the previously mentioned toolbars will be briefly discussed. The customized toolbar and its importance with respect to my application will be explained in detail.
Figure 5.2. CSV files for automobile companies layer.
Figure 5.3. The launch of the tool with one of the layers added.

5.3.1 Zoom/Pan Toolbar

The Zoom/Pan Toolbar contains nine standard buttons. The functionality of each one of them is briefly described in this section.

The toolbar seen in Figure 5.4 provides the user with the ability to manipulate the current map by zooming on it and moving it in a specific direction. It also allows to identify each feature on the current map.

Figure 5.4. Zoom/pan toolbar.

5.3.1.1 Previous Extent

Every time before the user modifies the scale of the map by using any of the zoom or pan tools, a view of the map with its current scale gets saved to the extent history. So when
this tool button is used (see Figure 5.5), it zooms the map to the previous extent stored in the extent history (it works like an undo for maps).

![Figure 5.5. Previous extent tool button.]

### 5.3.1.2 Next Extent

Every time before the user modifies the scale of the map by using any of the zoom or pan tools, a view of the map with its current scale gets saved to the extent history. So when this tool button is used (see Figure 5.6), it zooms to the next extent stored in the extent history (it works like a redo for maps).

![Figure 5.6. Next extent tool button.]

### 5.3.1.3 Zoom to Active Layer

This tool zooms the map to the extent of all the features that are part of the selected layer(s). This way the view of the map, after this button is pressed (see Figure 5.7), it’s scaled in a way that includes all the selected layer’s features and not more.

![Figure 5.7. Zoom to active layer tool button.]

### 5.3.1.4 Zoom to Full Extent

This tool zooms the map to the extent of all layers within the map. This means that after pressing this tool button (see Figure 5.8) we get an eagle eye view of the entire map changing the scale of it appropriately. It can undo previous zooms.

![Figure 5.8. Zoom to full extent tool button.]

### 5.3.1.5 Zoom In

Provides a tool for clicking or dragging a rectangle on the map in order to zoom in.
The scale for the current map is modified accordingly. When using this tool (see Figure 5.9), the cursor changes to a “zoom in magnifier” to indicate that the tool is active. In order to unselect the tool, the cursor tool button needs to be pressed.

Figure 5.9. Zoom in tool button.

5.3.1.6 ZOOM OUT

Provides a tool for clicking or dragging a rectangle on the map in order to zoom out. The scale for the current map is modified accordingly. When using this tool (see Figure 5.10), the cursor changes to a “zoom out magnifier” to indicate that the tool is active. In order to unselect the tool, the cursor tool button needs to be pressed (see section 5.3.3.5).

Figure 5.10. Zoom out tool button.

5.3.1.7 PAN

This button provides a tool for dragging the map to a new location without altering the map’s scale. When using this tool (see Figure 5.11), the cursor changes to a hand shape to indicate that the tool is active. In order to unselect the tool, the arrow (cursor) tool button needs to be pressed.

Figure 5.11. Pan tool button.

5.3.1.8 PAN IN ONE DIRECTION

Pans the map in one of four directions: north, south, east, or west. When pressing this tool button (see Figure 5.12), the drop down list, seen in Figure 5.13, is displayed.

Figure 5.12. Pan in one direction tool button.
Figure 5.13. Pan in one direction’s drop down list.

The list includes the options for panning the map north, south, east or west. The percentage to pan is based on the current map’s extent and it is applied toward the clicked pan bar direction.

5.3.1.9 IDENTIFY

Performs an identify function on the features that are part of the currently selected layer, which shows the non-locational attributes of the feature. When using this tool (see Figure 5.14), the cursor changes to a smaller pointer that has an information sign on top, to indicate that the tool is active. In order to unselect the tool, the arrow (cursor) tool button needs to be pressed.

Figure 5.14. Identify tool button.

5.3.2 Selection Toolbar

The Selection Toolbar (See Figure 5.15) contains seven standard buttons. The functionality of six of them will be briefly described in this section.

Figure 5.15. Selection toolbar.

5.3.2.1 FIND

This tool (see Figure 5.16) opens a dialog to locate features on the current map whose attributes contain an end-user provided string. It doesn’t have to be an exact match. As long
as the provided string is contained in any database field of any feature that is part of the selected layer, it will return the first match and the feature will be highlighted on the map.

5.3.2.2 CLEAR ALL SELECTION

This tool (see Figure 5.17) provides the user with the ability to clear all the selected features that result from using the previously discussed tools that are part of the Selection toolbar.

5.3.2.3 ATTRIBUTES

This tool (see Figure 5.18) displays the attributes of the currently selected features. All the database records of the current layer are displayed in the table.

5.3.3 Customized Toolbar

The Customized Toolbar contains six buttons. Their functionality is briefly described in this section.

5.3.3.1 PRINT

This tool (see Figure 5.19) provides the user with the usual printing functionality found in most user applications. It calls the native print dialog used in the operating system. It prints the current map on display.
5.3.3.2 ADD LAYER

This tool (see Figure 5.20) opens a dialog to be able to browse for the map layer (in SHP format) that wants to be added to the map currently displayed. See Figure 5.21. After browsing and selecting the desired layer to be added to the map, the new layer is added, the map is refreshed and the legend for the new layer appears on top of the Table of Contents area, located to the left of the map.

Figure 5.20. Add layer tool button.

Figure 5.21. Adding the shape file to the map.
5.3.3.3 REMOVE LAYER

In order to use this tool (see Figure 5.22), the layer to be removed should be selected. To select a layer is necessary to click its legend on the table of contents area. After the layer is selected, pressing the Remove Layer button will automatically remove the layer, update the map and the table of contents.

Figure 5.22. Remove layer tool button.

5.3.3.4 CURSOR (ARROW)

This tool (see Figure 5.23) provides a way for the user to unselect a map tool (that changes the shape of the cursor like Zoom In, Zoom Out, Pan, Identify, Hotlink and Help Tool) by changing the cursor back to the normal pointer.

Figure 5.23. Cursor (arrow) tool button.

5.3.3.5 HOTLINK

This button (see Figure 5.24) provides the user with a tool (the cursor changes to a lightning bolt) to identify the features on the selected layer by clicking on them and displaying a web page link that contains interesting information.

Figure 5.24. Hotlink tool button.

The Hotlink window seen in Figure 5.24 need to come up with fig contains, at the bottom, a “Site Information” button. When pressed, this button will open a browser window with the appropriate web page created for the site. All the web pages created for this application have the format seen in Figure 5.25.

In order to unselect this tool, the arrow (cursor) tool button needs to be pressed (see section 5.3.3.4).
5.3.3.6 CREATE POLYLINE LAYER / XY TOOL

This button (see Figure 5.26) was the GUI (Graphical User Interface) used to run the code implemented for the creation of polyline layers. This button opens a dialog to browse for the appropriate CSV file.

Figure 5.26. XY tool button.

Figure 5.27 shows the dialog that opens after the XY button is pressed. The CSV file is selected from here and then it appears as a new layer on the map and can be seen and edited from the table of contents panel. The user can easily add his own point layers to augment the application.
Figure 5.27. Selecting the steam cars CSV files.

5.4 APPLICATION MENU BAR

The Menu bar has four menus, Theme, Layer Control, Display and Help. I will discuss each of these, their sub parts, functionality and how to use them. File, Theme and Layer Control, Display drop down options are shown in Figures 5.28, 5.29 and 5.30, 5.31 and 5.32, respectively. All the menu were coded in Java and not supplied by MOJO.
Figure 5.29. Theme menu.

Figure 5.30. Layer control menu.

Figure 5.31. Automobile components menu.

Figure 5.32. Help menu.
5.4.1 Add Layer
Add layer menu item is similar to the Add Layer button discussed in Section 5.3.3.2

5.4.2 Remove Layer
Remove layer simply removes a layer from the map. The icon for remove layer is shown in Figure 5.33. Select the layer on the table of contents i.e. on the left side of the map, then go to file menu and then click remove layer. Now the cities layer is removed from the map.

Figure 5.33. Remove layer.

5.4.3 Legend Editor
This tool (see Figure 5.34) allows the user to modify the appearance of each layer that is part of the map currently displayed. The layer to be modified needs to be selected first. The legend editor provides the user with the option of modifying the layer’s color and labels through a dialog with different tabs.

Figure 5.34. Legend editor tool button.

In Figure 5.35 the layer is being selected from the table of contents bar for which we are performing the action of the legend editor.

The Figure 5.36 is the legend editor. Above are the properties which we can edit to the legend.

5.4.3.1 Attribute Table
An attribute table contains data about the layer that will be in the form of rows and columns. This attribute table is stored in the format of .dbf files. These files come along with shape files. The size of these depends upon the information they want to provide along with that layer. To see the attribute table, select a layer on the table of contents and then go to the theme menu and click on the open attribute table option that is shown in Figure 5.37.
5.4.3.2 **CREATE LAYER FROM SELECTION**

Using this option you can create a new layer from the selections made on the map using the select features functionality which is in Figure 5.18. Make selections as explained in Section 5.3.2.3 and as shown in Figure 5.18. The icon for this option is as shown in Figure 5.38.

After selecting some areas on the map using the shift key and select features tool click on create layer from selection. It will show a prompt to click on and do it. It adds automatically a new layer to the map. We made some selections and created a layer on the map that you can see in Figure 5.39. You can change the name of the layer using the legend editor, which we discussed about earlier.

5.4.4 **Layer Control: Promote/Demote Layer**

Using these options that are present under the menu layer control as shown in Figure 5.30 we can change the order of layers displayed on the map.
Figure 5.36. Steam properties legend editor.

Figure 5.37. Attribute table.

Figure 5.38. Create layer from selection.
The Help topics window presents some useful information as to how to use some important features. This can be further expanded and detailed information can also be provided here. See Figure 5.40.

5.4.6 Contact Us

Contact Us window provides contact details about the developer in case the user wants to contact the developer when there is an issue with the software. The contact us window is shown in Figure 5.41.
Figure 5.40. Help topics.
Figure 5.41. Contacting the developer of the application.
CHAPTER 6

AMERICAN AUTOMOBILE LAYERS

The American Automobile application is divided into two layers, general and steam automobile layers for easy understanding.

The two layers presented in this tool can be seen in the Figure 6.1. The picture contains all the layers added to the map.

![Image of a world map with layers added](image)

Figure 6.1. Application started using Java IDE.

6.1 ADDING A LAYER

The section describes the step by step process of adding layers after the tool is launched. The initial launch of the application is shown in Figure 6.1.
In Figure 6.1, none of the layers are added and we proceed on adding the layers/CSV files. On clicking the “XY” button on the map the dialog opens where I have all my CSV files (See Figure 6.2).

![Figure 6.2. Dialog after clicking XY button.](image)

Now I select the layer to be added. And Figure 6.3 is the snapshot of the CSV files of one of the layer.

Once CSV file is selected, and processed the map appears as shown in Figure 6.4:

On clicking on a point on the map, using hotlink tool an HTML page opens as shown in Figure 6.5.

**6.2 Automobiles Layer**

Listed in this layer, are the general automobile companies during this period. Figure 6.5 and Figure 6.6 are a few of the webpages.
Steam automobiles played a major role around the early 19th century before the emerging of gasoline powered automobiles [11]. This layer explains a few important automobile companies that are benefited by U.S and the major exports to the U.S. Figure 6.7 shows the HTML page of Stanley steamer.
Figure 6.4. After adding a layer

Figure 6.5. HTML page, after using hotlink tool.
Figure 6.6. HTML page of Buckeye Manufacturing Company of automobile layer.

Figure 6.7. HTML page of Stanley Steamer.
CHAPTER 7

SUMMARY AND OBSTACLES

There have been several challenges in developing this tool. I being a foreigner to America it took some time initially to understand the American automobile and its inception. More time was spent in attending vintage car shows and reading books and articles relating to this subject.

There was a lot of information on this topic, as the period is a budding stage for different auto companies in the world. Gathering the information and summarizing has been a challenging task.

Even though I had knowledge in developing GIS applications during my course work, I still encountered some roadblocks while developing the tool but was able to resolve soon with the help of the GIS notes which was provided during the course work. There were several occasions where the HTML pages did not span accordingly and I had to spend a lot of manual time aligning the components of the HTML page in order to provide a seamless view of information.

I would like to conclude that, in spite of all these challenges, it was an interesting and adventurous topic to study a totally different subject and increase my knowledge. In the end, nothing is as fulfilling as providing an educational software project for Users.
CHAPTER 8

FUTURE WORK

The Map Objects API utilized right now, permits the application to run just on desktop machines and laptops. A gigantic difference can be made, if API's for tablets running on Windows, Android and iOS stages can be included. In the event that the application can be conveyed on tablets and phones then it would be exceptionally valuable for the users.

Including of extra layers is one of the critical future works for this application. The simplicity of extending and changing makes it more positive for this. To give more extensive of data or to give more itemized data in the application, the accompanying extra layers can be included.

A layer of automobile companies, which exported the cars to different nations, can be added as separate layer.

A layer of merging between various automobile companies has happened during this period.

A layer of the cars that are imported from Europe during the period can also be added to the group.

Extra inputs can be taken from the genuine end users in the wake of testing the application. In order to receive feedback, a contact us info is included the menu.
CHAPTER 9

TESTING SECTION

The application has been shown to some vintage car buffs and include their feedback has been included

Sample comments:

“Lost Heritage Found” – David who lives in Fremont.

“I only know half of the automobile companies that are projected on your tool” - wislet
REFERENCES


