TIME LINE HELP PACKAGE FOR MAP OBJECTS, JAVA EDITION

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
Nomitha Mudireddy

Spring 2014
SAN DIEGO STATE UNIVERSITY

The Undersigned Faculty Committee Approves the

Thesis of Nomitha Mudireddy:

Time Line Help Package for Map Objects, Java Edition

Carl Eckberg, Chair
Department of Computer Science

William Root
Department of Computer Science

Gary Girty
Department of Geological Sciences

March 24, 2014
Approval Date
Copyright © 2014
by
Nomitha Mudireddy
All Rights Reserved
DEDICATION

I would like to dedicate my thesis to my husband Mr. Srinivas Mudireddy, my father Dr. M. Nirmal Reddy, my mother Mrs. Shobha Nandikonda, my entire family who has always cared for me and most importantly Prof. Carl Eckberg who has always given me endless support.
ABSTRACT OF THE THESIS

Time Line Help Package For Map Objects, Java Edition
by
Nomitha Mudireddy
Master of Science in Computer Science
San Diego State University, 2014

The timeline tool is an interactive tool that helps create a product where end-users can view various series of events that happened over time. It is a visual representation of the events that took place over time on the space. It is a simple tool that takes input from the users to represent the data on the map in timely manner. This tool is designed keeping in mind a vast set of users like from regular users to teacher, students, computer science students, professors etc. This tool can be mainly used for school projects by the teachers to help the students know more about history or any events that took place over time. Important inventions can be displayed in space vs. time on the map like displaying the “History of pi” which is greatly useful for mathematics students.

This tool is very interactive in the sense that the user is asked to enter the start and end year, also interval between the years for the period he wants to see the time line for. Depending on the data entered by the user, the slider bar on the map is represented with start and end years and the user given interval is used to mark the interval between the years on the slider bar. Users have control over the timeline they want to see and also the distance or gap between the years. Users also enter data for each event point like latitude, longitude, description and further data related to the event point. Users also upload pictures related to the event, wiki links and video links for the event points. Users can see a clear representation of the points on the map as the time line slider bar is moved. For each and every point user can click on the point to know more about the event point. Information, description, picture, wiki and video links can be seen for the point by clicking it.

A lot of map objects toolbars and custom tool bars have been added to make the tool more interactive and give the users a choice of playing around or getting more knowledge from the tool. One of the interesting addition to this tool is that user can enter data either by going point to point or by directly uploading the csv file. The second option is given keeping in mind computer science students or any user who has good computer knowledge and doesn’t want to take time by entering data for each and every point. Hence this tool is flexible and useful for wide range of users. There is also a help menu which explains the users on how to enter data at each and every step and how to use the tool. This tool is based on Java and Map Objects Java Edition 2. IDE Eclipse has been used to help write code.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>CHAPTER</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ABSTRACT</strong></td>
<td>v</td>
</tr>
<tr>
<td><strong>LIST OF FIGURES</strong></td>
<td>viii</td>
</tr>
<tr>
<td><strong>CHAPTER</strong></td>
<td></td>
</tr>
<tr>
<td>1  INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>2  REVIEW OF EXISTING SYSTEMS</td>
<td>3</td>
</tr>
<tr>
<td>2.1 Time Lapse Videos</td>
<td>3</td>
</tr>
<tr>
<td>2.2 Google Earth – Time Slider</td>
<td>5</td>
</tr>
<tr>
<td>2.3 Dipity</td>
<td>5</td>
</tr>
<tr>
<td>2.4 ArcGIS.com – Time –Enabled Maps</td>
<td>7</td>
</tr>
<tr>
<td>2.5 Dynamic Projection of Data on Maps Based On Timelines</td>
<td>7</td>
</tr>
<tr>
<td>2.6 Limitations of Existing Systems</td>
<td>8</td>
</tr>
<tr>
<td>3  IMPLEMENTATION</td>
<td>9</td>
</tr>
<tr>
<td>3.1 Technology Used</td>
<td>9</td>
</tr>
<tr>
<td>3.2 Swing Components</td>
<td>9</td>
</tr>
<tr>
<td>3.3 Map Objects</td>
<td>10</td>
</tr>
<tr>
<td>3.4 HTML</td>
<td>11</td>
</tr>
<tr>
<td>3.5 Java</td>
<td>12</td>
</tr>
<tr>
<td>3.6 Timelines</td>
<td>12</td>
</tr>
<tr>
<td>4  TOOL DESCRIPTION WITH SCREENSHOTS</td>
<td>14</td>
</tr>
<tr>
<td>4.1 Front End Screens</td>
<td>14</td>
</tr>
<tr>
<td>4.2 TimeLine Maps</td>
<td>20</td>
</tr>
<tr>
<td>4.3 Slider Bar</td>
<td>20</td>
</tr>
<tr>
<td>4.4 A Tailored Example - Super bowl</td>
<td>20</td>
</tr>
<tr>
<td>4.5 Application Used – History of Pi</td>
<td>23</td>
</tr>
<tr>
<td>5  MAP OBJECTS TOOLBARS</td>
<td>26</td>
</tr>
<tr>
<td>5.1 Zoom Pan Tool Bar</td>
<td>26</td>
</tr>
<tr>
<td>5.2 Selection Tool Bar</td>
<td>27</td>
</tr>
<tr>
<td>Section</td>
<td>Title</td>
</tr>
<tr>
<td>---------</td>
<td>-------</td>
</tr>
<tr>
<td>5.3</td>
<td>Query Builder</td>
</tr>
<tr>
<td>5.4</td>
<td>Select Features</td>
</tr>
<tr>
<td>6</td>
<td>CUSTOM TOOLBAR AND MENUS</td>
</tr>
<tr>
<td>6.1</td>
<td>Pointer or Release Tool</td>
</tr>
<tr>
<td>6.2</td>
<td>Add Layer</td>
</tr>
<tr>
<td>6.3</td>
<td>Remove Layer</td>
</tr>
<tr>
<td>6.4</td>
<td>Hot Link Tool</td>
</tr>
<tr>
<td>6.5</td>
<td>Help Tool</td>
</tr>
<tr>
<td>7</td>
<td>LIMITATIONS AND FUTURE ENHANCEMENTS</td>
</tr>
<tr>
<td>8</td>
<td>CONCLUSIONS AND OBSTACLES</td>
</tr>
<tr>
<td>REFERENCES</td>
<td></td>
</tr>
</tbody>
</table>
LIST OF FIGURES

PAGE

Figure 2.1. Time lapse video of ancient western civilizations timeline 3500BC-476AD...........................................................................................................................4
Figure 2.2. Time lapse video of nuclear explosions timeline from 1945........................................4
Figure 2.3. Google Earth...........................................................................................................5
Figure 2.4. Word War 2 timeline..............................................................................................6
Figure 2.5. Word War 2 projected on map.................................................................................6
Figure 2.6. US Presidents projected on map.............................................................................7
Figure 4.1. First UI screen......................................................................................................14
Figure 4.2. First UI screen with data.......................................................................................15
Figure 4.3. Help tool..............................................................................................................15
Figure 4.4. Map display screenshot.........................................................................................16
Figure 4.5. 2nd screen............................................................................................................17
Figure 4.6. 3rd screen............................................................................................................17
Figure 4.7. Upload picture screenshot....................................................................................17
Figure 4.8. Upload Csv file screenshot..................................................................................18
Figure 4.9. First Csv file screenshot.......................................................................................18
Figure 4.10. Second Csv file screenshot..............................................................................19
Figure 4.11. Slider Bar Screenshot.........................................................................................21
Figure 4.12. Super bowl Csv Screenshot...............................................................................22
Figure 4.13. Super bowl map screenshot...............................................................................23
Figure 4.14. History of Pi dialog screenshot...........................................................................24
Figure 4.15. History of Pi map screenshot..............................................................................25
Figure 4.16. History of Pi map and dialogs screenshot..........................................................25
Figure 5.1. Zoom pan tool bar..............................................................................................26
Figure 5.2. United States identification using identify tool....................................................27
Figure 5.3. Selection tool bar..............................................................................................27
Figure 5.4. Find tool.............................................................................................................28
Figure 5.5. Find window......................................................................................................29
Figure 5.6. Query builder window.................................................................30
Figure 5.7. Buffer window with 500 miles distance around Los Angeles........30
Figure 6.1. Custom tool bar. ............................................................................31
Figure 6.2. Select theme tool. ........................................................................32
Figure 6.3. Hot link window............................................................................34
Figure 6.4. Help tool. ......................................................................................36
CHAPTER 1

INTRODUCTION

Geographic Information Systems are information systems for geographic problems. We use GIS to model some aspects of the real world that we are dealing with and make use of the GIS to help us make important decisions related to our problem. Of course we resort to GIS for complex geographic problems only. GIS acts as a tool which one can use for exploring and analyzing data that are recorded for specific locations in geographical space. GIS provides a broader framework for analyzing spatial data.

GI Science is concerned with the collections or capture of spatial data by such methods as satellite remotely sensed images, GPS, surveys of people and/or land, light detection and ranging, aerial photographs, and spatially coded digital video. The key element is to capture not only attribute information but also accurate information on the location of each measurement of that attribute. Once spatial data has been captured they need to be stored and transmitted. This can create some challenges as some spatial data sets can be extremely large. So this huge amount of data is to be turned into useful information. Two main sets of techniques exist to turn data into information: visualization and statistical mathematical modeling [1].

Web mapping solutions are the biggest advantage provided to the users. They help us to visualize the sequence of events that took place over a time period. For example, if we take the time line of San Diego county, it would be really interesting to know how the place changed over years. The time line maps also are very educative in the sense that if we take the time line of history, it would walk us through the years. Google has shown a new face of the GIS to its users by developing various GIS products like: Google maps and Google earth. As this is given to users free of cost, it has increased the users see and use GIS. Google maps are a huge revolution in the day to day usage. These systems enable users to view maps without any need to update or update any shape files or to install any software. However with all these advantages, there is a lack of many features that map objects might provide and less scope for flexibility.
The purpose of this thesis project is to develop a tool that gives users a choice of building their own timeline map. This is developed keeping in mind teachers, students, geography experts and also common users such that they can build their own time line map for anything with little effort. It is much useful even for GIS students, since I have seen lot of my friends working on creating time lines for their GIS projects. The tool is built using java and map objects. The map objects provides various features which makes the map more interactive to the user. Users can click on the map and can see various features which help them to know more about any place over any year or any event. There is time slider bar which user can slide through and see all the events that happen over each year. The thesis is demonstrated by taking an example, which is “Super bowl” [2]. As the super bowl happens every year the time slider bar represents all the years from the start of super bowl to the year 2013. User can slide through and see the super bowl event mapped on the map every year. With the help of map objects like hotlink tool, user can click on the event point and know more about that place. Similarly user can create any time line using this tool, like timeline for US presidents, time line for US possessions, timeline for Olympics etc.

This thesis report is organized into 6 chapters. The 2nd chapter discusses the current existing GIS time line systems, the advantages and disadvantages of these current existing systems. The 3rd chapter explains all the technologies used to create this project which made the tool very interactive to the user. The next chapter displays screenshots of the tool, this is useful to understand how the tool works. The next chapter explains the menu bars provided by the map objects that help us to play around on the map. The final chapters discuss the conclusion, obstacles faced and scope for future enhancements.
CHAPTER 2

REVIEW OF EXISTING SYSTEMS

Timelines help us to visualize the data in a sequential manner. They give us an idea of how the sequence of events took place one after the other. Timelines can be created for various events like timelines for presidents, timelines for major earthquakes or volcanoes in United States or an example like timeline for super bowl. If we take the example of a timeline for the earthquakes, as each year progress timelines show each year where the major earthquake happened in United States. This gives the user a clear visual of patterns of events. The data of the events are clearly represented on the maps to give a better idea of the events. There are several existing systems that can be used to create such timeline maps.

2.1 TIME LAPSE VIDEOS

Time lapse videos are a collection of pictures or videos or data put together to represent the flow of events in the order that they took place. When all the data recorded is taken and put together it forms a video which shows the events in order. The main goal is to show a long period of time in a short time. It is used in applications like weather changes, changes in a city over a year, sunrise and sunset videos, traffic and crowd videos etc. [3].

Although there are many advantages and applications of time line videos there are many disadvantages too. There is no specific framework to generate timeline maps and also the application is not scalable. The users will not able to interact with the system and make their own changes in the application.

Below shows the example of a time lapse video of Ancient Western civilization in YouTube depicting the sequence of civilizations that happened between 3500BC to 500 AD (Figure 2.1 [4]).

Another example for time lapse videos is the nuclear explosion from 1965 (Figure 2.2 [5]). This timeline example shows all the nuclear explosions on map over time.

2.2 GOOGLE EARTH – TIME SLIDER

Google earth helps to travel back in time with historical imagery. It helps us to view any place like our neighborhood, home town or any other place how they have changed over time (See Figure 2.3 [6]).

![Google Earth](https://www.youtube.com/watch?v=eQ8h_jgl8c)


2.3 DIPITY

Dipity is a timeline website that allows users to organize the web content by date and time. Dipity timelines are interactive and they integrate text, links, locations timestamps, and video audio to create a timeline [7]. Dipity allows users to create free timelines on their website to bring back history to life. Dipity is a chronological time builder that takes information over internet like social media, search services and integrates all this data into a single interface which is easy for the users. They help users manually create timeline events or easily import content from YouTube, face book, Google, twitter etc. to the timeline [8, 9]. Users can also modify the timelines using advanced settings like timeline zoom, theme style etc.

The Figures 2.4 [9] and 2.5 [8] show Dipity timeline representing the Word War 2.

2.4 ArcGIS.com – Time-Enabled Maps

ArcGIS provides a feature which is the ability to “time-enable” your data in ArcMap. A time slider bar is provided for the user which enables the users to watch the events based on their date of occurrence. The time properties are saved with the map document. This time visualization on the map can be shared with others too. It also provides facility that if the other people whom you share the data with don’t have the ArcMap then this time enabled map can be quickly exported to an .AVI file, which can be viewed by a windows media player. This application is not open sourced and is licensed by ESRI [10].

2.5 Dynamic Projection of Data on Maps Based on Timelines

This is the thesis developed by SDSU student Swagath Manda. The tool is developed keeping in mind wide range of users. The tool integrates with the Google map. A slider is provided similarly, using which users can slider over and get idea of sequence of events that happened (example shown in Figure 2.6 [11]).

2.6 Limitations of Existing Systems

The existing systems mentioned above have various disadvantages like they lack portability, flexibility. Some of them are not open source and hence needs to be purchased. And some of the projects do not have much flexibility.

This project is aimed at making an open source tool which addresses the issues of above disadvantages. We have used java, map objects, esri mojo libraries, Java swing API’s which enable us to generate a time sensitive data.
CHAPTER 3

IMPLEMENTATION

Java Swing components are used to make the front end UI screens. Swing components like JDialog, JFileChooser are used to project and grab the user entered data. Java is used in the back end to convert the user entered data into projectable map points. Map objects are used to project the points created on the map. Swing components are used to project the map and give various set of controls like zoom panel tool bar, and other tools.

Html is used to display the messages and also the dialog box structure which asks the user to enter data about the time line they want to create. Swing components like JDialog, Feature Layer are used to create the feature layers of the points and display the timeline. Components like JLabel are used to display the headings and data in the JDialog. ImageIcon is used to use display the images of various event points. BorderLayout is used to place the labels and text boxes in the JDialog. Swing component JSlider is used to display the slider bar, it slides and as the user slides through the years the points are displayed in the map.

3.1 TECHNOLOGY USED

The following section explains different technologies used to develop this thesis. Some of the technologies are from ESRI, a technology company which develops GIS solutions and some of them are generic technologies which are chosen depending on their integration with ESRI.

3.2 SWING COMPONENTS

Swing, which is an extension library to the AWT, includes new and improved components that enhance the look and functionality of GUIs. Swing can be used to build Standalone swing GUI Apps as well as Servlets and Applets. It employs model/view design architecture. Swing is more portable and more flexible than AWT. Swing is built on top of AWT and is entirely written in Java, using AWT’s lightweight component support. In particular, unlike AWT, the architecture of Swing components makes it easy to customize both their appearance and behavior. Components from AWT and Swing can be mixed,
allowing you to add Swing support to existing AWT-based programs. For example, swing components such as JSlider, JButton and JCheckbox could be used in the same program with standard AWT labels, text fields and scrollbars [12]. There are various components used in this project like:

- JButton
- JLabel
- JSlider
- JMenuItem
- JButton
- JPanel: - The panel is used to display the map and all other components of this project. Everything that we see in this project is projected onto the JPanel.

### 3.3 Map Objects

Map objects is a java application and a GIS development tool that is written in java by ESRI. It’s propriety of ESRI and is available as the JAR files that can be installed to work with java. This Map objects using Java Objects is called MOJO and MOJO uses component architecture. Components architecture examples are Visual Basic, VB.Net and also VB version of Map Objects by ESRI.

As MOJO uses component architecture, there are various advantages of this architecture. It is flexible, meaning that the user can change the original functionality for which the code was written. User can have the component and override it to have his own control. The user can modify various factors of the component like color, shape etc. User can use the component as much as he wants or as little as he wants. He can either override what is there in the component or use only few portion of the component. And this map objects can easily be integrated with multiple IDES like eclipse, and also Visual Basic.

There are various features of the map objects that are cited from the reference [13] are:

- Wide range of GIS capabilities. Applications which are built using Map Objects Java Edition can support activities such as labeling map features, thematic mapping, Panning and zooming through multiple map layers, specifying projections, querying Spatial and attribute data, performing geometric operations, measuring distances, displaying real time geographic data, and creating layouts.
- Ability to combine Multiple Distributed Data Sources. By using applications created with Map Objects Java Edition, end users can combine local data with Internet and Intranet data to create their own customized maps and easily integrate these with existing application architectures. Supported data formats include, industry standard shape files, ArcSDE layers, and ArcIMS image and feature services, image formats such as BMP, TIFF, PNG, JPG, GIF and MrSID, CAD (DGN, DXF, and DWG), military formats, and much more. You can also use Map Objects Java Edition to access files from your own custom, industry specific data sources for easy integration.

- Ability to deploy ones Applications over the Web. One can easily distribute Map Objects Java Edition applications over an internet or intranet through browser hosted applets or simplify web delivery of your application with the use of Sun Java web start technology, the industry standard launching mechanism for distributing Java applications over the Web.

- Helpful tools for building a user interface. By using the Swing components included with Map Objects Java Edition, you can quickly build applications that include functional toolbars, dynamic symbol control, query dialogs, overview and insert maps, and intelligent legends that make your custom applications easy to use and even easier to develop.

- Server side (J2EE) components. The server side map components in Map Objects Java Edition allows developers to build map services, Java Server Pages (JSPs) and Servlets, or custom Enterprise Java Beans (EJB) Web based mapping applications. Several extensive sample applications based on JSPs and EJB are provided to Demonstrate how to build robust Web applications with the server side map objects. Lot of the features are used in this project to project the event points on the map and display as the slider bar moves.

### 3.4 HTML

HTML or Hypertext Markup Language is a markup language used to create web pages. It is a simple language used to create front end pages for any websites. It consists of various tags like `<h1>` `</h1>` or `<p>` .. `<p>`, so all the data is in the tags. We can mention how we want this data to be. If we want the data inside the tags to be label, we have `<Label>` tag. On the other hand if you want the data to be displayed as a paragraph then `<p>` is used and `<h1>`, `<h2>`, `<h3>` for the headers. Html provides wonderful tools for taking user input by using “text area”. When the user inputs some data into this text area and clicks on the “Submit” button then the data is submitted to the action or the data can be stored anywhere. Submit button is one of the important functionality of the html. We can create whatever buttons we want and the action required for that button can be written in java. In this project we have used the text fields to take user input like latitudes, longitudes, event years etc. And
lot of buttons like Upload, submit, next are used to upload the data and move on to the next screens. Many versions of HTML exists, the most recent one supports cross platform mobile applications.

### 3.5 Java

Java is an open source programming language. It is used widely for desktop applications, web applications [7]. There are many advantages of using java over other languages for this thesis project. The most important advantage is that it integrates very well with ESRI’s Map Objects Java Edition. Swing and awt APIs have been used extensively in this thesis project. Following are some other advantages of Java:

- **Serialization and Deployment**: Java allows serialization of objects. In this thesis project, a database has been created in the form of serialized objects. Serialized objects made this tool run standalone. Users can download this tool and run the application just by double clicking on the executable jar file. All the data will be stored on the user machine. The users should not edit that serialized file. The editing of the serialized file may prevent the application from running properly.

- **Integration with MOJO**: Map Objects Java Edition is a set of jar files provided by ESRI to develop GIS applications. Those jar files have methods which can be called easily through java code. ESRI gives you about 17 jar files with MOJO. In my thesis project only 2 jar files have been used and that made my tool size small.

- **Platform Independence**: Java is platform independent. It is a very big advantage of the Java language. It means class files can run on any platform, be it Windows, Mac or Linux. Class files are compiled java files. Class files contain Byte Code which is interpreted by the Java Runtime Environment. Users can run the tool created in this thesis project on any platform.

- **Mature Language**: Java is a pretty mature language. It is very stable. Heavy support is available online for Java. When I started this thesis project, I was not expert in Java. Whenever I had some coding issues, I would search online and most of the times, I would find answers very easily.

### 3.6 Timelines

This thesis project is implemented by asking the user to enter the data by 2 routes: either by giving latitude, longitude, description and other details of the event. Then this data is converted into a csv file by CsvFileCreation.java class. The converted csv file is then used to create a shape file using the ShapeFileWriter in ShapeFileCreation.java class. This shape file is the base for displaying the map. TimeLineUI.java is the front end which has dialog boxes that take the input from the user. Then the entire base map and the entire data user
entered is processed and displayed on the map. All the processing of the data like feature layer creation is done in SuperBowl.java.
CHAPTER 4

TOOL DESCRIPTION WITH SCREENSHOTS

4.1 FRONT END SCREENS

The following screenshots show the homepage or the first screen of the application. This (Figure 4.1) screen is developed using java and map objects.

![Figure 4.1. First UI screen.](image)

This screen is developed using java swing objects like JDialog which is used to display this screen and JLabel is used to display the titles and JTextArea is used to input data. The Drop down list shown in Figure 4.2 is the list of base shape files for the different continents and some of the important countries. More shape files for other counties or anything else can be added easily (Figure 4.3).

The screen asks for the start year, end year, the interval, and chooses the base shape file. This data is used to setup the base for the timeline tool. The start and end year are used to decide the starting and ending year in the timeline slider bar. And the interval determines the gap between featured years (See Figure 4.4: Slider Bar Screenshot which shows a 4 year gap). Depending on the start and end year, the user enters the interval logistically which helps them view the years clearly. A large gap between the start and end years and a small interval may lead to too many marked years, not easily represented. For example, a span of 1000 years with a 2 year gap would ask for 500 labels on the slider. So endDate – startDate /
Figure 4.2. First UI screen with data.

Figure 4.3. Help tool.
interval should probably be between 10 and 35. A warning is issued if the endDate –
startDate / interval is less than 5 or >50.

The second screen asks the user choice how he wants to enter the data for timeline.
The choices given are (Figure 4.5):

- Enter Data for each and every event point -- Data is entered for each and every point by giving longitude, latitude, name and description of the event, year of the event, Wikipedia link, video link, and also related picture for that event. If the picture is not uploaded then there is a default picture shown for the point. And same is the case with the video link and wiki link. If the user has not entered any of the links for any of the event points then the defaults links are opened. If the user has entered the links then when the user clicks on the event point with the hotlink tool then the dialog box opens up with the picture, description, year, wiki link, video link. And when the user clicks on these links they are opened in a browser.

Clicking on the next will take you to the next screen (Figure 4.6), user needs to clear the old data and enter the new data for the next point and click on next again until all the points are entered, user keeps on clicking the next and click on finish finally.

*Upload Button:* This is used to upload the picture for the event. A picture selector is opened and user can select any picture from the user’s machine (Figure 4.7). Then on licking the Finish button, all the data entered is converted into a csv file. And this csv file is converted to the shape file which displays these event points on the map.
Figure 4.5. 2\textsuperscript{nd} screen.

Figure 4.6. 3\textsuperscript{rd} screen.

Figure 4.7. Upload picture screenshot.
• Add csv file option—Instead of entering the data directly, if the user is well aware of csv files, he/she can directly upload data placed in csv files (Figure 4.8).

![Figure 4.8. Upload Csv file screenshot.](image)

On clicking the “Add Csv” button the next screen which appears is shown above. This screen displays 2 upload buttons for the user to upload the data in the form of 2 csv files. The first csv file has the format of longitude, latitude, place name and also the year. On clicking the Upload 1st Csv Button a File chooser opens up which allows the user to select a file from his machine. For example see Figure 4.9.

![Figure 4.9. First Csv file screenshot.](image)
The second csv (Figure 4.10) file can be uploaded by clicking on the “upload 2\textsuperscript{nd} Csv” button. Similar to the first one a file chooser JDialog opens up and user can pick the file from his/her machine. The format of the 2\textsuperscript{nd} file has some more additional information along with the longitude, latitude, place name, year of event. The additional information includes picture link, YouTube link and also Wikipedia link for the event. Since each of the events has a different picture, the user needs to have the path to this picture in the csv file. Two csv files are asked to be uploaded by user since, for each of the event point we need the path to the picture of the event, their wiki and YouTube links, and for the shape file creation we need the csv file only to have the longitude, latitude, place name and year. So if the user uploads only one file, the code should have to split the single csv file into 2 csv files. Instead as the user is well aware of the csv file creation it is easier on the user’s part to create 2 separate csv files and upload them.

![Figure 4.10. Second Csv file screenshot.](image.png)

Instructions are given to the user clearly as to what is the format of 1\textsuperscript{st} csv file and also the 2\textsuperscript{nd} csv file. In the shape file dbf file, only longitude, latitude, year and name are kept. ESRI has a limit on string on length for a dbf field, and the last 4 fields above might sometimes exceed that limit. Hence the first csv file has only these 4 fields, because it is used to create the dbf file/ shape file.
4.2 **Timeline Maps**

On uploading the csv files or entering the data and clicking on the finish the maps opens up in a new window with the points displayed. Here I have taken Super bowl as the example (see Figure 4.4) where starting year is 1966 and ending year is 2013. The interval between the years is 4 since, this gap makes the years clearly visible. The map shows the base map as the US map and the points of the events are displayed as the feature layer. Each event point is marked or highlighted in the map as a red dot so that it is easily noticeable to the user. Layers are shown on the left side of the map and we can choose to display or not display those layers by check marking the check box beside each layer. On the right side of the map a slider bar is displayed with all the years marked and there is arrow shaped thing, which can be moved along the slider bar. As this arrow shaped thing is moved over the years then the points relevant to the year appear on the map.

4.3 **Slider Bar**

Slider bar is the important part of this project. As we move the slider bar over the years the points on the map get highlighted according to the years (Figure 4.11). The start year and end year are taken from the data entered by the user. The picture below shows a major tick every 4 years, this was mentioned by the user. Interval can be controlled by the user. The small ticks are for every year are calculated depending on the interval that is given by the user.

4.4 **A Tailored Example - Super Bowl**

I have taken super bowl as an example to code through the entire thesis. The application starts by asking the start year, end year, and interval between the years and also choosing the base map which is US map. For the super bowl, I have given start year as 1966, end year as 2013 and interval between the years as 4, since this gap would show the years clearly, and base map is chosen as the US map. On clicking next there is screen which asks for the choice to be selected, whether to “Enter the data” or to “Upload the csv file”. If we choose to go with the first option, then data is to be entered for each and every point. For super bowl, the first point entered is, -118.15 (longitude), 34.04 (latitude), point name or city name (Los Angeles), description about Los Angeles, year of event – 1966, wiki link –
http://www.youtube.com/watch?v=bTvr_2v-0HI [15]. And there is an upload button for each point, on clicking that a file chooser opens and we can choose any file from the desktop. On clicking next we can enter the next point and super bowl as it happens every year, for the year 1967 all the data is entered similarly as described above. Until the year 2013, we can enter data for each and every year until it finishes, and click on Finish. Once this is done shape file is created internally for this super bowl data and it is added as a layer names “super
bowl” onto the US base map. Beside each layer there is a check box, we need to tick mark the check box and highlight the super bowl layer. On highlighting now on the right side of the map, there is a slider bar which has start year 1966, end year 2013. There is an arrow provided on the slider bar and as we move through the years 1966, 1967, 1968 etc., Los Angeles (1966), Miami (1967), and Miami (1968) get highlighted on the map marked as a red circle. On clicking on any of these points when highlighted using the hotlink too |(Thunderbolt symbol) a dialog box opens which has all the information about these points. It has description, place name, year, picture of the place, and buttons for wiki and video links. On clicking the wiki or video buttons the wiki page and video links open in the browser if connected to internet. Hence we can open and see the wiki link of super bowl in Los Angeles, Miami etc. and learn more information about these events.

On the other hand if the user doesn’t want to take the path for entering all the data for each and every point, then we can click on “Upload data” and directly upload the csv file that contains all the information. 2 files need to be uploaded. The first file has longitude, latitude, place name and year (see Figure 4.12). The second file contains longitude, latitude, place name, year, and path of the picture, wiki link and video link in a csv file.

![Figure 4.12. Super bowl Csv Screenshot.](image)
On uploading both the files and clicking on finish, the following map opens up. The red circle is the first point in the year 1966, Los Angeles i.e. the first year super bowl was hosted (see Figure 4.13).

Figure 4.13. Super bowl map screenshot.

4.5 APPLICATION USED – HISTORY OF PI

Another example where time tool was demonstrated or used is in the project “History of pi”. The timeline tool was used in demonstrating the history of pi for a mathematics class in SDSU. As this tool is given to a group of students who has to demonstrate the sequence of discoveries that lead to “pi” formula and value today, the shape file for this is already created unlike the application where the user initially enters the start and end dates. The application when clicked opens the map with points already marked on it. And the slider bar shows the range of date from 250 AD to 1850 with the arrow pointer at the start year which is 250 AD. As soon as we move the slider bar over from 250 AD, a line starts to draw from the starting point for the 250 AD and a dialog box appears with Archimedes (see Figure 4.14). This is the first invention about pi, by Archimedes in 250 AD, he approximated its value. There are few buttons given like “learn more” and “See Video”, these buttons on clicking open the web page to show more information about Archimedes

http://www.math.utah.edu/~alfeld/Archimedes/Archimedes.html [16]. Also the video link opens up which gives more information about Archimedes.
Another example where time tool was demonstrated or used is in the project “History of pi”. Similarly as we slide along the slider bar across year 455 AD another dialog box pops up for Zu Chongzui calculated the value of pi using polygons. Similarly more information can be found by clicking on the see video or learn more buttons. Now we can see that there is a line drawn from the first point 250 AD to 455 AD (see Figure 4.15), and as the slider bar moves more lines are drawn between the previous point and next points.

All the tools which are available in the time line tool like hotlink, zoom in, zoom out, query, pan over, identify tool, select features tool, all of these are available similarly for the History of pi too. Various tool bars some of which are custom made and some of them are from map objects are displayed above which help the user to use them for various functionalities. Also there is a distance tool provided below the map which calculates the x and y coordinates as the mouse pointer is moved across. As we move across to the end of the slider bar, line between all the points is drawn and all the dialog boxes appear beside each other (see Figure 4.16). This helps the user to view all the events that lead to discovery of pi in a sequence and see all of them at once.
Figure 4.15. History of Pi map screenshot.

Figure 4.16. History of Pi map and dialogs screenshot.
CHAPTER 5

MAP OBJECTS TOOLBARS

Map Objects Java Edition 2 comes with a couple of toolbars like Selection tool bar, Zoom pan tool bar. These in built tool bars provide some standard functionality like zoom in, zoom out, pan etc. In this thesis project, we have used two standard tool bars; Selection tool bar and Zoom pan tool bar. This chapter discusses these two tool bars in detail and how these two tool bars have been integrated into this thesis project. We will also provide the code snippets to show you the integration.

5.1 ZOOM PAN TOOL BAR

There are 9 standard tools provided by Zoom pan tool bar as shown in Figure 5.1. We will discuss each in detail as following starting with the left most one:

![Zoom Pan Tool Bar](image)

**Figure 5.1. Zoom pan tool bar.**

- *Previous Extent:* It zooms the map to the previous extent stored in the extent history. It does not make any difference to the map if you have not zoomed in or zoomed out the map and you use this previous extent tool.

- *Next Extent:* This tool zooms the map to the next extent stored in the history. When you launch the application, this tool is inactive. It becomes active only after you have used the previous extent tool.

- *Zoom to active layer:* This tool zooms the selected layer. This tool is inactive until you select one of the layers from the TOC.

- *Zoom to full extent:* This tool zooms the map to the extent of all layers within the map. After you have zoomed in or zoomed out a particular point and now you want to get back to the original map size, you can use this tool.

- *Zoom In:* A user can zoom in on the map using this tool. A user can drag a rectangle on the map and the selected area will get zoomed. A user can also click on the map to zoom that particular area.

- *Zoom out:* This is opposite to Zoom in. You can use this tool in similar way as of zoom in. You can draw a rectangle or click on the map to zoom out the area.
- **Pan**: It is used to pan the map to any direction. A user can drag the map to any direction he wants. This tool is useful only after you have zoomed in the map and you are not able to see some part of the map.

- **Pan one direction**: It does the same thing as Pan Tool, but just in one specified direction.

- **Identify**: This tool lets you click on the map. Location of the clicked point is identified and the properties of that point are shown. The following screenshot as shown in Figure 5.2 explains that.

- **Code snippet to show zoom pan tool bar integration**: 
  ```java
  ZoomPanToolBar zptb = new ZoomPanToolBar(); zptb.setMap(map); jPanel.add(zptb);
  ```

Figure 5.2. United States identification using identify tool.

### 5.2 Selection Tool Bar

There are 7 tools in the selection tool bar as shown in Figure 5.3. We will be explaining each of them starting with the left most one:

- **Search**: It opens a window for locating features based on predefined stored query.

- **Find**: It opens a window for searching the values in the selected layers. The following screenshot as shown in Figure 5.4 shows that Los Angeles was searched after selecting the Super bowl layer and two results came which are shown in the right panel. You can notice that Super bowl - Los Angeles has been
selected on the map and the color of Los Angeles has been changed to yellow. Yellow is the default selection color of Map Objects Java Edition 2.

5.3 Query Builder

The users can build their own queries using this tool. This tool is inactive initially. After you select a layer from the TOC, this tool becomes active. You can see all the fields of the selected layer in the left panel and then using buttons on the screen, you can build your own query and the result of the query is shown at the bottom side of the window. The result set of the query gets selected on the map and shown in yellow color.

Figure 5.5 shows what the Query builder window looks like and what options users have on this window.

5.4 Select Features

This tool lets users select features on the map by rubber banding a shape in the map. The users can select one of the following four shapes:

1. Polygon
2. Rectangle
3. Circle
4. Line

One of the layers must be selected in the TOC. This tool is inactive until one of the layers is selected from the TOC. The features can be selected only from the selected layer.

- **Clear all selections:** This tool is pretty simple to use. It clears all the selections. The selected features color is yellow. After you click this tool, the yellow color goes away or in other words, the selected features are no longer selected. One of the features on the map must be selected to make this tool active. Initially, this tool is inactive because no features are selected.

- **Buffer:** This tool lets users construct a buffer polygon around the currently selected features. This tool becomes active only if one or more features are selected on the map. Figure 5.6 shows a 500 mile buffer around the selected feature.

- **Attributes:** This tool shows you all the attributes of the selected features. Figure 5.7 shows you all the attributes of the 8 selected features. This tool becomes active only if one or more features are selected on the map.

- **Code snippet to show selection tool bar integration:**
  ```java
  SelectionToolBar stb = new SelectionToolBar();
  stb.setMap(map);
  jPanel.add(stb);
  ```
Figure 5.6. Query builder window.

Figure 5.7. Buffer window with 500 miles distance around Los Angeles.
CHAPTER 6
CUSTOM TOOLBAR AND MENUS

In addition to the toolbars provided by Map Objects, a custom toolbar was created to enhance the functionality of the GIS tool. The custom toolbar looks like Figure 6.1.

All the tools in the toolbar have tooltips to help users know more about the tool. The following list describes each tool in detail starting from the left most one. We are showing the code snippets also for the tool.

6.1 POINTER OR RELEASE TOOL

This tool is used to release other tools. When a tool is selected from either of the Map Objects toolbars or from the custom toolbar, the cursor icon is changed to the selected tool icon. If a user wants to go back to the original state or the state when no tool is selected, the users will click on the pointer tool.

Code snippet for Pointer tool: Java class named Arrow has been created to achieve the functionality of the pointer tool.

```java
public class Arrow extends Tool {
    // Override
    public void mouseClicked(MouseEvent arg0) {
        // Code
    }
}
```

The following code shows how the button is created and an action is attached to that.

```java
JButton btnPointer = new JButton(new ImageIcon("icons/pointer.gif"));
btnPointer.setToolTipText("Pointer");
```
btnPointer.addActionListener(this);
public void actionPerformed(ActionEvent ae) {
    if (ae.getSource() == btnPointer) {
        map.setSelectedTool(new Arrow());
    }
}

6.2 ADD LAYER

This tool gives the ability to the users to add new layers to the map. When the application starts, there are seven active layers in the map. The users can add more layers to the map using this tool. As soon as the layer is added, users can see the newly added layer in the TOC and map is updated also.

Figure 6.2 shows how the layer can be added to the map.

![Select theme tool](image)

**Figure 6.2. Select theme tool.**

Users can browse to the location of the shape file and then click OK to add that layer to the map.

Code snippet for Add layer tool:
JButton btnAddLayer = new JButton(new ImageIcon("icons/addtheme.gif"));
btnAddLayer.addActionListener(this);
public void actionPerformed(ActionEvent ae) {
    if (ae.getSource() == btnAddLayer) {
        AddLyrDialog aldlg = new AddLyrDialog();
        aldlg.setMap(map);
        aldlg.setVisible(true);
    }
}

Following are the key points to be noted from the AddLyrDialog class.

com.esri.mo2.ui.bean.CustomDatasetEditor cus = new com.esri.mo2.ui.bean.
CustomDatasetEditor();
getContentPane().add(cus,BorderLayout.CENTER);
panel1.add(ok);
panel1.add(cancel);
getContentPane().add(panel1,BorderLayout.SOUTH);
map.getLayerset().addLayer(cus.getLayer());

6.3 REMOVE LAYER

This tool is used to delete the selected layer. As soon as the layer is deleted, the TOC
and the map is refreshed to show the changes. When the application launches, this tool is
inactive because no layer is selected at that time. Once a user selects a layer from the TOC,
this tool becomes active and then the user can click on it.

Code snippet for remove layer tool:
JButton btnRemLayer = new JButton(new ImageIcon("icons/delete.gif"));
com.esri.mo2.map.dpy.Layer dpylayer = legend.getLayer();
map.getLayerset().removeLayer(dpylayer);
map.redraw();
6.4 Hot Link Tool

This tool is used to find Super bowl location on the map. The users can click on the map to see what is the point displayed in red on the map. When a user clicks on the map, on the point a window opens up to show information about the point clicked. There is a text about that place and there are 2 buttons in that window. When the users click on the button “Learn More”, a web page opens up the wiki link of the place. There is another button “See Video” which opens up the YouTube link on a new webpage (Figure 6.3).

![Figure 6.3. Hot link window.](image)

The bolt looking icon in the tool bar is Hot Link tool.

Code Snippet for Hot Link tool:

```java
JButton btnHotLink = new JButton(new ImageIcon("icons/hotlink.gif"));
btnHotLink.addActionListener(this);
Identify hotlink = new Identify();
MyPickAdapter picklis = new MyPickAdapter();
hotlink.addPickListener(picklis);
map.setSelectedTool(hotlink);
class MyPickAdapter implements PickListener {
    public void beginPick(PickEvent pe) {
    }
};
```
public void endPick(PickEvent pe) {}
public void foundData(PickEvent pe) {
    //fires only when a layer feature is clicked
    FeatureLayer flayer2 = (FeatureLayer) pe.getLayer();
    com.esri.mo2.data.feat.Cursor c = pe.getCursor();
    Feature f = null;
    Fields fields = null;
    if (c != null)
        f = (Feature)c.next();
    int columnIndex = getInfoColumnIndex(pe.getLayer().getName(), f);
    mystate = (String)f.getValue(3);
    try {
        HotPick hotpick = new HotPick();
        hotpick.setVisible(true);
    } catch(Exception e) {
    }
};

6.5 Help Tool

This tool gives the ability to users to learn about the other tools. The users can left click on this tool to activate this tool. Once the help tool is selected, the users can right click on the other tools and then a window will open up which tells users about how to use the selected tool. Figure 6.4 shows the window which opens up when a user right clicks on the hot link tool after selecting the help tool.

Code snippet for Help Tool:
JButton helpjb = new JButton(new ImageIcon("icons/help2.gif"));
if (ae.getSource() == helpjb) {
    helpToolOn = true;
    map.setSelectedTool(new HelpTool());
}
Figure 6.4. Help tool.
CHAPTER 7

LIMITATIONS AND FUTURE ENHANCEMENTS

The Time line tool that we have created is a useful tool for both people who have little idea about programming and also for those who have good programming knowledge. Yet there are few limitations for this tool. The major limitation for this tool is the validation between latitude and longitudes of each point. When the user enters the latitude and longitude for a point there is no way that we can validate which is the latitude and which is the longitude. We can only validate if the user has entered the correct format of the latitude and longitude like numbers only allowed and no characters allowed. But if the user by mistake enters the latitude in place of longitude and vice versa the map displays weird points. Hence user needs to be careful when he enters the latitudes and longitudes. As to err is the human nature there is no tool or any round to identify the user data entry mistake.

Another disadvantage is there is no cross platform compatibility. This tool is not designed to be deployed on mobile phones. In order for this tool to be deployed on the mobile phones we need to use jQuery and other things like Google maps instead of using the ESRI map objects. Hence the ESRI objects need to be installed first and they need to be integrated with the eclipse and the java libraries for the ESRI to work on the machine. And the ESRI is not free software, user needs to buy license in order for this map to work. And also there is some ground work needed to be done by the user if he wants to use a new shape file that is not present in the tool. The advantage though is that ESRI map objects use the shape files that are readily available on Internet. But if the shape file is not present then user might to create one. ESRI provides a tool called ArcView which can be easily sued to create shape files. The scenarios where user might want to create a shape himself is, if we want to display time line for the San Diego county school then the shape file might not be available since only most commonly used shape files are available in the internet.

This tool implementation only allows certain parameters to be entered in the csv file. Since the data from the csv file is read according to the predefined file format and any
change in that will cause errors in the time line map. And the file format that may be uploaded is only the csv file. Other file types are not accommodated.

Further enhancement in this area would allow users to provide custom defined data fields and also allow SQL imported data to be supported. If the user is able to import the SQL data instead of filling the data in the csv file, huge amounts of data can be accommodated. And also it eases for the user since most of the applications today have data stored in the databases or something huge. So this would provide the tool a capability to display large volumes of data. Also allowing user to have other file types will help user feel more comfortable since, sometimes there are file types like KLM files or word files etc. which can store the data.

JSlider is the most important component of this project. It is provided by ESRI map objects. This is quite useful and flexible in the sense that it allows user to give start year, end year and also the interval between the years. Even though it has various advantages it has limited itself in the appearance. For example, if the user has entered the data that he wants the time line between 500 BC to 2 AD. So here the time gap is more than a 100 years. In this case the time line maps display the data effectively, but the years on the slider bar are too close, it seems like overlapping and not clearly visible. This may be to the user not understanding on what year he has moved the pointer too. Hence enhancement in the appearance of the slider bar when there is more data will help the users to clearly understand and see the years over the slider. Not only that it will encourage the user to work on huge volumes of data instead of being limited and splitting up a big chunk of time frame into 2 small time frames. Also more flexibility in the years display will be good enhancement. In some scenarios there might be a situation where there is no event taking place in a certain period of time. If we take the time line for scientific discoveries in united states, there might be few years where there are no discoveries at all and since slider bar displays years sequentially that amount of space is waste in the time line slider bar, instead if only the years where discoveries are made are displayed on the slider bar then it would save lot of space and accommodate more new data. All these enhancements will make the tool more powerful and convenient to the users.

Another few enhancements thought of is the allowance for large text entries so that the user is not limited for the large number of years. Also for each point only one image is
allowed to upload now, but allowance for multiple images upload for a single point must be
good enhancement. And similarly multiple YouTube links [5], multiple audio links per event
point and multiple wiki links and so forth will be reasonable enhancements for this tool.
CHAPTER 8

CONCLUSIONS AND OBSTACLES

This tool developed as a part of GIS thesis, is a very interactive and user friendly tool that enables the user to create their own time line Maps. These maps display time sensitive data which gives a user an idea of the sequence of events that took place. The time line bar is quite sensitive and a handy tool that helps user to view the data that changes with time dynamically. This tool is quite interactive with the user, as it takes data from the user and the data displayed on the map changes dynamically according to that. The tool works both ways for the users with little programming knowledge or for the one who have good experience with the computers. It achieves this functionality by providing two paths to the users where he/she can choose to enter the data for each event point, if they are not aware of the file systems. But the user with good computer knowledge can avoid this path, instead of wasting his/her time by entering the data for each and every point this user can directly upload the csv file with all the related data for the event points.

This tool is useful for wide range for users like from school teachers who wants to show their students time line of the US states acquisition to a normal person who wants to just see how his community's events over a period of time. The tool can be used to display time lines of various examples like super bowl, US Presidents, British empire etc. They can be used effectively in the history, geology and other areas involving spatial data. Users also can connect to the wiki links or the video links to know more about a certain point.

Several obstacles were faced during the creation of this project. Various shapes files are needed for this project which may be used as the base shape file. Shape files for all continents and few important countries are being created since some of there were not available directly on the internet. So this shape file creation was done as the pre work for this project, and it had some troubles doing that. The shape file writer class provided by ESRI is not very consistent and gives weird errors lot of times.
REFERENCES