MULTIMEDIA SURVEY OF COLONIAL INDIA

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
Shruthi P. Rao
Fall 2013
SAN DIEGO STATE UNIVERSITY

The Undersigned Faculty Committee Approves the

Thesis of Shruthi P. Rao:

Multimedia Survey of Colonial India

[Signatures]

Carl Eckberg, Chair
Department of Computer Science

William A. Root
Department of Computer Science

Vadim Ponomarenko
Department of Mathematics & Statistics

Oct 23, 2013
Approval Date
Copyright © 2013
by
Shruthi P. Rao
All Rights Reserved
DEDICATION

I would like to dedicate this thesis to my parents, sister, my husband, parents-in-law and friends who have always supported, and stood by me in all my accomplishments. I also dedicate this thesis to my advisor and thesis chairperson Dr. Carl Eckberg who has been an invaluable source of support throughout my graduate studies.
ABSTRACT OF THE THESIS

Multimedia Survey of Colonial India

by

Shruthi P. Rao

Master of Science in Computer Science

San Diego State University, 2013

The thesis emphasizes on building an intuitive and robust GIS (Geographic Information systems) tool which gives an insight into the colonial era of India, including the advent of independence. The GIS tool incorporates various European colonies and their territories on the ancient Indian map. The user can know more about the European colonies by clicking on the map.

The ancient Indian Map also incorporates the major cities and territories of the European countries which had colonies in India. This includes the colonies of the British, the Portuguese, the French, the Dutch and the Danish. The tool also depicts the post independence territorial map of India, Pakistan and Bangladesh which was a part of British India.

The tool has HTML (Hypertext markup language) pages which provide information on various European colonies and their trade and administration in colonial India. Web hyperlinks have been provided for the users which give more detailed information on the colonial period of India.

The tool has been developed in core JAVA. MOJO (Map Objects Java Edition) is used for presenting the ancient India map. Map Objects is developed by ESRI. The major features shown on the Indian map is designed using Map Objects. ESRI library modules have been used to render various graphical features on the map.

The user interface design uses HTML for content rendering and CSS for the styling. Some of the key aspects are shown using pictures which make the tool more interesting for the users.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>ABSTRACT</th>
<th>.................................................................................................................................................. v</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIST OF TABLES</td>
<td>........................................................................................................................................... viii</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>............................................................................................................................................ ix</td>
</tr>
<tr>
<td>ACKNOWLEDGEMENTS</td>
<td>.............................................................................................................................................. x</td>
</tr>
<tr>
<td>CHAPTER</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>INTRODUCTION ........................................................................................................................................ 1</td>
</tr>
<tr>
<td>1.1</td>
<td>Overview ........................................................................................................................................ 1</td>
</tr>
<tr>
<td>1.2</td>
<td>About the Application .................................................................................................................. 2</td>
</tr>
<tr>
<td>1.3</td>
<td>Background ...................................................................................................................................... 2</td>
</tr>
<tr>
<td>2</td>
<td>REQUIREMENTS ..................................................................................................................................... 3</td>
</tr>
<tr>
<td>2.1</td>
<td>Data Requirements .......................................................................................................................... 3</td>
</tr>
<tr>
<td>2.2</td>
<td>Platform and Functional Requirements ......................................................................................... 3</td>
</tr>
<tr>
<td>3</td>
<td>TECHNOLOGIES ................................................................................................................................... 5</td>
</tr>
<tr>
<td>3.1</td>
<td>Java .................................................................................................................................................. 5</td>
</tr>
<tr>
<td>3.2</td>
<td>Java Swing ...................................................................................................................................... 6</td>
</tr>
<tr>
<td>3.3</td>
<td>Cascading Style Sheets (CSS) ........................................................................................................ 7</td>
</tr>
<tr>
<td>3.4</td>
<td>HTML5 .............................................................................................................................................. 8</td>
</tr>
<tr>
<td>3.5</td>
<td>Map Objects - Java Edition ............................................................................................................. 9</td>
</tr>
<tr>
<td>3.6</td>
<td>Eclipse IDE ...................................................................................................................................... 10</td>
</tr>
<tr>
<td>4</td>
<td>COLONIAL INDIA .............................................................................................................................. 11</td>
</tr>
<tr>
<td>4.1</td>
<td>Discovery of New Sea Route to India and Advent of Europeans ................................................... 11</td>
</tr>
<tr>
<td>4.2</td>
<td>Portuguese ...................................................................................................................................... 11</td>
</tr>
<tr>
<td>4.3</td>
<td>Dutch .............................................................................................................................................. 12</td>
</tr>
<tr>
<td>4.4</td>
<td>Danes .............................................................................................................................................. 13</td>
</tr>
<tr>
<td>4.5</td>
<td>French ............................................................................................................................................. 14</td>
</tr>
<tr>
<td>4.6</td>
<td>British ............................................................................................................................................ 15</td>
</tr>
<tr>
<td>4.7</td>
<td>Indian Freedom Movement and Partition of India .......................................................................... 16</td>
</tr>
</tbody>
</table>
SOFTWARE PROTOTYPE ..............................................................18
SOFTWARE ARCHITECTURE ......................................................21
ECLIPSE ENVIRONMENT SETUP ...............................................23
TOOLBARS AND BUTTONS .......................................................26
  8.1 ZoomPan Toolbar ...............................................................26
  8.2 Selection Toolbar ...............................................................26
  8.3 Custom Buttons ...............................................................26
SHAPEFILES AND FEATURE LAYERS ........................................30
  9.1 Creating Shapefiles ...........................................................30
  9.2 Feature layers .................................................................31
  9.3 Promoting Layers ............................................................33
SCREENSHOTS ...........................................................................35
CONCLUSION AND CHALLENGES ...........................................37
FUTURE ENHANCEMENTS .......................................................38
REFERENCES .............................................................................39
LIST OF TABLES

Table 8.1. ZoomPan Toolbar Tools .................................................................27
Table 8.2. Selection Toolbar Tools .................................................................28
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1</td>
<td>Application startup page.</td>
<td>19</td>
</tr>
<tr>
<td>5.2</td>
<td>Map base layer.</td>
<td>19</td>
</tr>
<tr>
<td>5.3</td>
<td>Colonial India webpage.</td>
<td>20</td>
</tr>
<tr>
<td>6.1</td>
<td>Architecture diagram.</td>
<td>21</td>
</tr>
<tr>
<td>7.1</td>
<td>Adding ESRI library.</td>
<td>23</td>
</tr>
<tr>
<td>7.2</td>
<td>Adding external JAR files.</td>
<td>24</td>
</tr>
<tr>
<td>7.3</td>
<td>Selecting the JAR files.</td>
<td>24</td>
</tr>
<tr>
<td>7.4</td>
<td>Setting the main class and configuring other run time parameters.</td>
<td>25</td>
</tr>
<tr>
<td>8.1</td>
<td>MapObjects – toolbars.</td>
<td>26</td>
</tr>
<tr>
<td>8.2</td>
<td>ZoomPan toolbar</td>
<td>27</td>
</tr>
<tr>
<td>8.3</td>
<td>Selection toolbar.</td>
<td>28</td>
</tr>
<tr>
<td>8.4</td>
<td>Custom buttons.</td>
<td>28</td>
</tr>
<tr>
<td>8.5</td>
<td>Selecting CSV file after enabling XY button.</td>
<td>29</td>
</tr>
<tr>
<td>8.6</td>
<td>Feature layer added after selecting CSV file.</td>
<td>29</td>
</tr>
<tr>
<td>9.1</td>
<td>Create layer from selection.</td>
<td>30</td>
</tr>
<tr>
<td>9.2</td>
<td>British India layer.</td>
<td>31</td>
</tr>
<tr>
<td>9.3</td>
<td>Dutch India layer.</td>
<td>32</td>
</tr>
<tr>
<td>9.4</td>
<td>French and Danish colonies.</td>
<td>33</td>
</tr>
<tr>
<td>9.5</td>
<td>Post independent India and Pakistan.</td>
<td>34</td>
</tr>
<tr>
<td>9.6</td>
<td>Promoting/demoting layers.</td>
<td>34</td>
</tr>
<tr>
<td>10.1</td>
<td>Danish India webpage.</td>
<td>35</td>
</tr>
<tr>
<td>10.2</td>
<td>Dutch India webpage.</td>
<td>35</td>
</tr>
<tr>
<td>10.3</td>
<td>French India webpage.</td>
<td>36</td>
</tr>
<tr>
<td>10.4</td>
<td>Freedom movement webpage.</td>
<td>36</td>
</tr>
</tbody>
</table>
ACKNOWLEDGEMENTS

I thank my advisor and Thesis chair person Professor Dr. Carl Eckberg, San Diego State University for providing me an opportunity to work on this thesis project and supporting and guiding me to accomplish this thesis.

I thank Professor William Root, San Diego State University for his acceptance and being part of this Thesis committee.

I am also grateful to Professor Vadim Ponomarenko, San Diego State University for showing interest in knowing about the application and being part of this Thesis committee.
CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

Colonial India refers to the part of the Indian subcontinent which was under the control of European colonial powers, through trade and conquest. The main aim behind developing this tool is to provide a software program to students who want to know more about Indian history, the colonial era in particular.

In the era of web learning where the method of knowledge acquisition has shifted from traditional paper books to computer based systems, a software program could ideally provide information on the colonial period of India, suitable for self study or to supplement a standard text.

The objective behind the development of this tool is to garner interest in history among the target audience and make learning history more fun. This is accomplished with the use of pictures, maps, web pages, external web links, audio and video clips etc.

The GIS tool is developed using JAVA and Map Objects Java edition which is a product of ESRI (Environmental Science Research Institute). ESRI has the major market share in the field of GIS and geo database management. The term GIS, can be thought of as geospatial information studies which refer to the academic discipline of working with geographic information systems and is a large domain within the broader academic discipline of Geo informatics [1]. GIS is the merging of cartography, statistical analysis, and computer technology [2].

The remainder of this chapter talks in detail about the application tool. Chapter 2 tells about the requirement specifications for the software application tool.

The technologies used to develop this application, Java, MOJO, HTML, CSS and Java Swing are discussed in detail in Chapter 3. Chapter 4 provides more light on the advent of the Europeans, the Colonial Era of India and the transition to independence. This chapter also discusses how the tool renders this information to the end users. Chapter 5 and Chapter 6 discuss about the software prototype and the architecture design of the application. Chapter 7
tells how to set up the java environment and the ESRI Jar files. Chapter 8 explains about custom menus and toolbars. Chapter 9 tells about the shape files and layer creation, Chapter 10 has some additional screenshots of the application. Chapters 11 and 12 discuss the challenges and the future enhancements respectively.

1.2 ABOUT THE APPLICATION

This application tool has been developed using Map Objects Java Edition, a product of ESRI. ESRI offers a bottom up Java solution for adding dynamic maps to applications. Map Objects allows the user to build custom map applications or applets using the open source Java programming language. Map Objects–Java Edition includes prebuilt JavaBeans that are easily used in integrated development environments such as Net Beans and Eclipse. Map Objects can also be used to perform a variety of geography based display, query and data retrieval activities, which are typical of GIS products [3, 4].

The IDE used for developing this application is Eclipse 4.2 which is a flexible Java developer’s environment to develop applications. Each map layer is associated with HTML pages and external web links which provide the user with appropriate information. Map Objects supports image formats like BMP, TIFF, PNG, JPG. Map Objects aids in the display of shape files as map layers, provides standard GIS toolbars etc.

1.3 BACKGROUND

Earlier theses written with guidance in history provided by Professor Mahdavi of San Diego State University History department, are available at www-rohan.sdsu/~eckberg and included these periods of Indian history, Ancient Harappan civilization [5], Delhi Sultanate [6], Mauryan Empire [7] and Mughal dynasty [8].

This thesis fills an obvious and an important gap in these chronicles.
CHAPTER 2

REQUIREMENTS

This thesis project is a tool developed for young high school or college history enthusiasts to aid them to learn about the colonial period of India. This tool is developed under the guidance of Dr. Carl Eckberg of Computer Science Department, San Diego State University. The requirements mainly came from the thesis supervisor, Dr. Carl Eckberg. The requirements are classified into two categories, namely: Data requirements, Platform and Functional requirements.

2.1 DATA REQUIREMENTS

The data requirements refer to the information presented in the application. The following milestone events during the colonial period of India are identified to be a part of data requirements.

- The advent of Europeans and the discovery of a sea route to India by Vasco Da Gama.

- Formation of British East India company and depiction of British territories on the ancient India map.

- French East India company and French colonies.

- Dutch East India company and Dutch colonies.

- Portuguese East India company and colonies.

- Danish presence and Danish occupied cities.

- Partition of India and Freedom movement.

2.2 PLATFORM AND FUNCTIONAL REQUIREMENTS

This section refers to the functional and platform details of the application.

- The tool should come up with the ancient India map when launched.

- Legend table of contents should display all the layers and allow the user to select the layer to be viewed.

- The tool should have buttons to provide an access to the HTML pages.
The HTML pages should contain a brief description about each of the layers and external hyperlinks which provide more information.

- The HTML pages should contain a brief description about each of the layers and external hyperlinks which provide more information.
- The hotlink tool should be provided in order to provide brief information about the layer clicked.
- The tool should be interactive and designed in such a way that it provides collective information through pictures, audio, and video clips.
- All necessary tools should be provided to the user to interact with the application and gain enough knowledge.
- Tool should be flexible and accommodate the dynamic addition of layers.
- The tool has to be platform independent.
- The tool is developed using Java, and Map Objects Java Edition is used to provide GIS solution which makes the tool platform independent.
- The application has to be neatly packaged into a JAR file which can be downloaded and run on any machine.

There have been other master theses which chronologically cover Indian history, including ancient Harappan civilization, Mauryan dynasty, Delhi Sultanate and Mughal empire [5-8]. The motivation to develop this tool, about colonial India, is to fill the gap which witnessed European colonialism and the Indian freedom movement.

All the theses projects about India can be packaged into a software application and this interactive software application can be a great source of information for high school students learning history by self study in a multimedia fashion.
CHAPTER 3

TECHNOLOGIES

This section introduces the technologies and platform used to develop the application tool. It describes the advantages of each of these technologies and thus justifies their usage in this application tool. JAVA Swing is used to develop the application GUI. Platform independence and ease of use make Java a natural choice for programming. The other technologies used are Map Objects-Java Edition, HTML and CSS which are discussed in detail later in this section.

3.1 JAVA

JAVA is a general-purpose, concurrent, object-oriented language based on C/C++, designed for easy Web/Internet applications and developed by James Gosling of Sun Microsystems. It is specifically designed to have as few implementation dependencies as possible and has gained widespread acceptance from the programming community. It follows the philosophy of "write once, run anywhere" (WORA), meaning code that runs on one platform does not need to be recompiled to run on another [9].

Salient Features of Java:

- Simple
  
  Java has a relatively simple syntax and is easy to program; it has done away with some of the rarely used, confusing features of C++ like operator overloading, multiple inheritance, and extensive automatic coercions. It has no pointers but provides for automatic garbage collection. It also provides a rich set of predefined Class Libraries to support application development in Java.

- Object oriented

  Primary focus in Java is on the data (objects) and methods that manipulate the objects. Almost all data types are objects (files, strings, etc.) and all functions are associated with objects providing for better code organization and reuse. Unlike C++, everything in Java must be placed in a class.

- Interpreted and Portable

  Java compilers generate byte-codes which are platform-independent. Java byte-codes are translated on the fly to machine readable instructions in runtime by a Java Virtual Machine. So each application runs on all platforms; the sizes of the
primitive data types are always the same and the libraries define portable interfaces.

- **Reliable and Secure**
  
  Java provides for extensive compile-time and runtime error checking; as it has no pointers memory corruptions or unauthorized memory accesses are impossible; also automatic garbage collection tracks objects usage over time. A lot of emphasis has been placed on security which enables the construction of virus-free, tamper-free systems. The authentication techniques are based on public-key encryption. The memory allocation model makes it impossible for applications to forge unauthorized access to data structures or to access private data in objects.

- **Multithreaded and Dynamic**
  
  Java utilizes a sophisticated set of synchronization primitives (based on monitors and condition variables paradigm) and enables multiple concurrent threads of executions to run simultaneously. Threads are very handy for GIS since redrawing map layers can very time consuming.

- **Easily Deployed**
  
  Platform independence and executable JAR files make it easy to deploy Java applications to almost all environments. Java is designed to adapt to evolving environment; libraries can freely add new methods and instance variables without any effect on their clients.

### 3.2 JAVA SWING

Swing is essentially a **Java widget Toolkit** with a set of extensible GUI components to enable developers to rapidly develop powerful Java front ends for commercial applications. It is an improvement over the earlier **Abstract Window Toolkit (AWT)**; in addition to familiar components such as buttons, check boxes and labels, Swing provides several advanced components such as tabbed panel, scroll panes, trees, tables, and lists [10].

**Salient Features of Swing:**

- **Pluggable look-and-feel architecture**
  
  Swing is a highly modular-based architecture, which allows for the "plugging" of various custom implementations of specified framework interfaces: users can provide their own custom implementation(s) of these components to override the default implementations using Java's inheritance mechanism.

- **Model View Controller Design**
  
  Swing architecture is loosely based on the **model-view-controller (MVC)** design in which a visual application is broken up into three separate parts: a **model** that represents the data for the application, a **view** that is the visual representation of that data and a **controller** that handles the user input. In swing the View and
Controller objects are collapsed in to a single User Interface object known as UI Delegate

- Platform Independent

Swing is implemented entirely in Java to promote cross-platform independence, consistency and easier maintenance. It provides a single API capable of supporting multiple look-and-feels so that developers and end-users would not be locked into a single look-and-feel.

- Customizable

The Swing framework allows users to exercise a fine control over the rendering of a component. As a general pattern, the visual representation of a Swing component is a composition of a standard set of elements, such as a border, inset, decorations, and other properties. Users can customize a standard Swing component by assigning specific borders, colors, backgrounds, opacities, etc. The core component will then use these properties to render itself. A typical GIS package does not offer this degree of customization.

- Lightweight UI

Swing "paints" its controls using the Java 2D APIs, rather than calling a native user interface toolkit. Thus, a Swing component does not have a corresponding native OS GUI component, and is free to render itself in any way that is possible with the underlying graphics GUIs. In this context Swing components are described as lightweight because they do not require allocation of native resources in the operating system's windowing toolkit unlike their AWT counterparts.

### 3.3 Cascading Style Sheets (CSS)

CSS is a style sheet language that defines the layout of HTML documents. While HTML is used to structure content, CSS is used for formatting structured content. CSS allows the separation of presentation from structure which makes site maintenance a lot easier. CSS can define color, font, text alignment, size, borders, spacing, layout etc., and can do so independently for on-screen and printed views. CSS also defines non-visual styles such as the speed and emphasis with which text is read out by text readers. CSS is supported by most of the currently available browsers. CSS has gone through many versions like CSS1, CSS2, and CSS3 and currently CSS4 is under development [11].

- CSS Syntax

CSS has a simple syntax and uses a number of English keywords to specify the names of various style properties. A style sheet consists of a list of rules. Each rule or rule-set consists of one or more selectors, and a declaration block. Selectors are used to specify which part of the markup a style applies to and may be applied to all elements of a specific type, or to elements specified by an
attribute. A declaration block consists of a list of declarations in braces. Each declaration itself consists of a property, a colon (:), and a value. If there are multiple declarations in a block, a semi-colon (;) must be inserted to separate each declaration.

Selectors can be combined in many ways, to achieve great specificity and flexibility.

- **CSS Sources**

  CSS information can be provided from different sources. CSS style information can be applied from a separate document or it can be embedded in a HTML document. Multiple style sheets can also be imported. Different styles can be applied depending on the output device being used like screen or print. The style sheet with the highest priority controls the content display. Declarations not set in the highest priority source are passed on to a lower source which is known as **cascading**.

**Benefits of CSS:**

- Ability to control layout of many documents from one single style sheet;
- Allows more precise control of document layout
- Ability to apply different layouts to different media-types (screen, print, voice etc.);
- Provides for many advanced and sophisticated techniques

### 3.4 HTML5

**HTML5** is a markup language used for structuring and presenting content for the World Wide Web and a core technology of the Internet. HTML5 is the next major revision of the HTML standard superseding HTML 4.01, XHTML 1.0, and XHTML 1.1 and is a joint effort of World Wide Web Consortium (W3C) and the Web Hypertext Application Technology Working Group (WHATWG). Many features of HTML5 have been built keeping low-powered devices such as smart phones and tablets in mind and thus HTML5 is a potential candidate for cross-platform mobile applications.

With HTML5, one can make use of a wide variety of graphics elements, such as animation, games, movies, etc. Even intense graphics effects such as lightning and shadows, 3D, special effects, vector graphics and so on are supported.

The latest versions of Apple Safari, Google Chrome, Mozilla Firefox, and Opera all support many HTML5 features. The mobile web browsers that come pre-installed on iPhones, iPads, and Android phones all have good support for HTML5 [12].

**Salient New Features of HTML5**
• **New syntactic features:** These include the new `<video>`, `<audio>` and `<canvas>` elements, as well as the integration of scalable vector graphics (SVG) content and MathML for mathematical formulas. Other new elements, such as `<section>`, `<article>`, `<header>` and `<nav>`, are designed to enrich the semantic content of documents.

• **Forms 2.0:** Improvements to HTML web forms where new attributes have been introduced for `<input>` tag.

• **Offline Web Applications:** HTML5 will allow one to create apps that function even when not connected, or when system is offline. The HTML 5 specification provides two solutions: a **SQL-based database API** for storing data locally, and an **offline application HTTP cache** for ensuring applications are available even when the user is not connected to their network.

• **WebSocket:** A next-generation bidirectional communication technology for web applications.

• **Server-Sent Events:** HTML5 introduces events which flow from web server to the web browsers and they are called Server-Sent Events (SSE).

• **Canvas:** This supports a two-dimensional drawing surface that one can program with JavaScript.

• **Audio & Video:** Support for embedding audio or video in web pages without resorting to third-party plugins.

• **Geolocation:** Visitors can choose to share their physical location with a web application.

• **Microdata:** Creation of own vocabularies beyond HTML5 and extension of web pages with custom semantics.

• **Drag and drop:** Ability to drag and drop the items from one location to another location on the same webpage.

### 3.5 Map Objects - Java Edition

Map Objects - Java Edition is a product of ESRI (Environmental Systems Research Institute, leader of Geographic Information system technology). ESRI offers a pure Java solution for adding dynamic maps to applications through Map Objects. This is a collection of Java-based GIS and mapping developer components that can be used to build custom, cross-platform GIS applications. Map Objects--Java Standard Edition also includes prebuilt JavaBeans that can be easily used in the integrated development environment [3, 4].

Key features of Map Objects include:

• Capability to combine multiple data sources (local, Internet, and Intranet) to create customized maps;
• connectivity with ArcIMS

• Compatibility with many data sources including shape files, ArcSDE layers, and a variety of image formats such as BMP, TIFF, PNG, JPEG, and GIF.

• Ability to create feature layers from custom data sources.

Many resources are available on Map Objects such as a guide to building applications, a programmer's reference, Javadoc (standard interface and class descriptions), many sample applications including source code, quick-start tutorials, and a comprehensive object model diagram.

The book written by Professor Dr. Carl Eckberg, San Diego State University is a valuable resource for developing GIS applications using Map Objects. This book provides numerous source code examples to implement various features of Map Objects [3].

### 3.6 Eclipse IDE

Eclipse is a multi-language Integrated development environment (IDE) comprising a base workspace and an extensible plug-in system for customizing the environment. It is written in Java. It can be used to develop applications in Java and, by means of various plug-ins, other programming languages like C, C++, JavaScript, Perl, PHP, Python, Ruby (including Ruby on Rails framework) [13].

The Eclipse SDK includes the Java development tools and is actually meant for Java developers. However users can also extend its abilities by installing plug-ins such as development tool kits written for the Eclipse Platform for other programming languages. Eclipse SDK is free and open source software.

Eclipse provides views, perspective and editors. There is a debug perspective which helps in debugging the code while giving break points. The JVM in Eclipse is faster and the auto suggestions for bug fixing makes Eclipse is one of the efficient choices for the developers. Netbeans is another popular alternative.
CHAPTER 4

COLONIAL INDIA

The next five sections briefly document that a large number of European countries had a strong presence in various parts of India, which inevitably led to conflicts and an ebb and flow of influence. To appreciate this turbulent period better, a map based strategy is necessary.

4.1 DISCOVERY OF NEW SEA ROUTE TO INDIA AND ADVENT OF EUROPEANS

The accounts of Marco polo and other travelers about the immense wealth of India kindled the interest of European communities. There was also a heavy demand in Europe for Indian commodities like spices, calicoes, silk and precious stones. Arab traders who controlled major land routes to India had a monopoly over India’s external trade and charged Europeans exorbitantly for Indian goods. All these resulted in European communities trying to discover a new sea route to India. The search for the wealth and prosperity of India led to the accidental "discovery" of the Americas by Christopher Columbus in 1492. Bartholomew Diaz rounded the Cape of Good Hope, or the Stormy Cape, as he called it, in 1487; and Vasco da Gama found out a new route to India and reached the famous port of Calicut on the 17th May, 1498.

Portuguese, Dutch, Danish, English and French were some of the major European communities who arrived in India. Initially their arrival was solely for the purpose of conducting trade but later their geo-political ambitions and weakened state of India resulted in many of them conquering Indian territories and subjugating the native rulers [14].

4.2 PORTUGUESE

The Portuguese were the first European community to establish direct trade with India with the arrival of Vasco da Gama’s ship San Gabriel at the busy port of Calicut in South India. He established a factory at Cochin in 1502. It was Alfonso de Albuquerque who was the architect of Portuguese power in India. He first came to India in 1503 as the
commander of a squadron, and was appointed Governor of Portuguese affairs in India in 1509. In November, 1510, he captured the rich port of Goa, then belonging to the Bijapur Sultanate, and during his rule did his best to strengthen the fortifications of the city and increase its commercial importance. With a view to securing a permanent Portuguese population, he encouraged his fellow-countrymen to marry Indian wives. During the earlier half of the sixteenth century Indian external trade was dominated by the Portuguese, who controlled the sea routes on the western shores of India, effectively blockading any ships belonging to other European nations.

Alphonso De Albuquerque and his Portuguese successors established their settlements at Diu, Daman, Salsette, Bassein, Chaul and Bombay, San Thome near Madras and Hugli in Bengal. Their Authority also extended over the major part of Ceylon. They built the St. Angelo Fort at Kannur to guard their possessions in North Malabar. Most of their provinces with the exception of Goa, Daman and Diu were lost to the Maratha Empire in early 18th Century, whereas Mumbai was gifted as dowry to King Charles II of England in 1661.

Portugal retained control of Goa till 1961 when it was liberated by Indian forces and became part of Indian nation. Several causes led to the decline of Portuguese power in India, such as their religious intolerance which provoked the hostility of the Indian rulers, their clandestine practices in trade, the discovery of Brazil that attracted Portugal to move away from India, and their failure to compete successfully with the other European Companies who came in their wake [15].

4.3 Dutch

The lucrative spice trade which brought prosperity to Portugal compelled the Dutch to seek direct access to the spice markets in South-East Asia and undertook a series of voyages since 1596. The Dutch East India Company was formed by a charter of Dutch parliament with powers to make wars, conclude treaties, acquire territories and build fortresses in March, 1602.

Dutch established factories in Gujarat, on the Coromandel Coast and in Bengal, Bihar and Orissa, entering deep into the interior of the lower Ganges valley. The more important of their factories in India were at Masulipatam (1605), Pulicat (1610), Surat (1616), Bimlipatam
(1641), Karikal (1645), Chinsura (1653), Cassimbazar, Baranagore, Patna, Balasore, Negapatam (1658) and Cochin (1663). Slowly the Dutch replaced the Portuguese as the most important naval power and practically monopolized the spice trade in the East throughout the seventeenth century. They also became the carriers of trade between India and the islands of the Far East. They fortified Pulicat with the permission of the native ruler and conquered Ceylon from the Portuguese.

At Surat the Dutch traded in large quantities of indigo, manufactured in Central India and the Jumna valley, and from Bengal, Bihar, Gujarat and Coromandel they exported raw silk, textiles, saltpetre, rice and Gangetic opium. After 1690, Negapatam instead of Pulicat became the chief seat of the Dutch on the Coromandel.

The years 1630-1658 formed a period of expansion for the Dutch on the Coromandel Coast and extension of their trade in other regions. Their policy of monopolizing trade brought the Dutch in to intense conflict with the British. During the years 1672-1674 the Dutch frequently clashed with the British and captured three English vessels in the Bay of Bengal. The Dutch later became less involved in India, as they had the Dutch East Indies (now Indonesia) as their prized possession and in fact exchanged Chinsura for British held Sumatra in 1825 and exited India for good [16].

4.4 DANES

The Danes had a brief presence in India. The Danish East India Company formed in 1616 established a colony known as Fredericknagore, near Serampore, West Bengal in 1699 which failed as a commercial venture. It established trading outposts in Tranquebar, Tamil Nadu (1620) and the Nicobar Islands (1750s). In 1777, after the Danish company went bankrupt, Serampore became a Danish crown colony. Though a commercial failure, Serampore emerged as an important center on the cultural front. As British had banned missionary activities in their settlements Serampore supported missionary activities in India.

In 1799, Reverend William Carey and two fellow Baptist missionaries established the first printing press in Asia, in Serampore to print copies of the Bible. Serampore College, the first institution to impart western style higher education in Asia was also started in 1819 by Reverend Carey. In 1827, a Royal Charter by the King of Denmark declared it as a university at par with those in Copenhagen and Kiel.
In 1845, Denmark ceded Serampore to Britain, thereby ending the nearly 150 years of Danish presence in Bengal [17].

4.5 French

The French East India Company was formed in 1664 at the request of Minister Colbert in the reign of Luis XIV. The French company was created, controlled and financed by the state and it differed from the British East India Company which was a private enterprise. The French were the last of the European powers to compete for commercial gains in the East with the rest of the European Companies. They established their first trading outpost at Surat in 1668. Later they started several trading centers at Masulipatnam, Chandranagore, Yanam, Mahe, and Karikal. In 1673 Francois Martin obtained a little village from the Muslim governor of Valikondapuram and developed it into an important place. Thus was Pondicherry born, which was their most important settlement and served as their seat of power and capital in India.

Initially the Dutch and the French fought bitterly in India as a result of their European rivalries. In 1672 the French seized San Thome, from the Dutch but were defeated in the very next year and had to surrender it again to the Dutch. In retaliation Pondicherry was captured by the Dutch in 1693 but returned back to the French by the Treaty of Ryswick in 1697. The fortunes of the French fluctuated widely during this period and entered a very bad phase by the early part of 18th century. The French closed their factories at Bantam, Surat and Masulipatam and even sold their licenses to others. The company was reconstituted in 1720 and prospered between 1720 and 1742 under the wise administration of their governors Lenoir and Dumas. The French at this point in time were pursuing only their commercial objectives and had no political ambitions. They enlisted troops and fortified their factories only as a safeguard against the attacks from the Dutch and the English.

In 1741 Joseph François Dupleix was appointed the governor of French possessions in India. He was an ambitious man who desired to create a French Empire in India and dabbled in local politics. Dupleix's army successfully controlled a large area between Hyderabad and Cape Comorin. Dupleix's ambition clashed with British interests in India and resulted in a series of military skirmishes and political intrigues. Battle of Plassey in 1757, the Battle of Wandiwash, and the siege of Pondicherry in 1760 established
the military superiority of the British and the French lost control of their possessions. The enclaves of Pondicherry, Karaikal, Yanam, Mahé and Chandernagore were returned to France in 1816 by the British. On November 1, 1954, the four enclaves of Pondichéry, Yanam, Mahe, and Karikal were transferred to the Indian Union and became an integral part of Indian Republic bringing the curtain down on French rule in India [18].

4.6 British

With a view to challenge Portugal's monopoly of trade with Asia and carry on the lucrative spice trade, the British East India Company was formed in 1600. In 1608 the Company dispatched Captain Hawkins to the court of the Mughal emperor Jahangir who granted them the permission to start a factory at Surat in 1613. Later on they established several factories at Agra, Broach, Hariharpur, Balasore, Hugly and Kalighat which eventually developed as today’s Kolkata. In 1639 they established another factory at Madras and in 1688 Bombay was transferred to the company by Charles II and grew prosperous by leaps and bounds. By the end of eighteenth century the British emerged supreme and marginalized all other European trading communities. The ports of Kolkota in the east, Bombay in the west and Madras in south were well developed, fortified and contributed in great measure to Company’s thriving trade.

The battle of Plassey fought in 1757 in which the British forces under Robert Clive defeated the combined forces of the French and the navab of Bengal proved to be a turning point in the history of modern India. This battle changed British perspective as they realized their strength and potential to conquer smaller Indian kingdoms; it thus marked the beginning of the imperial or colonial era.

A series of Anglo-Mysore wars were also fought in South India to end Tipu Sultan’s rule; the kingdom of Mysore was annexed by the British in 1799, after the defeat of Tipu Sultan in the fourth Anglo-Mysore War. A Governor General was appointed by the Company to administer its possessions who operated out of Culcutta, the official capital of British India.

British policy during the 19th century was chiefly concerned with expanding and protecting its hold on India, viewed as its most important colony and the key to the rest of Asia. In the 1840s and 1850s, under the governor-generalship of Dalhousie and then
Canning, more territories were absorbed into British India, either on the grounds that the native rulers were corrupt, inept, and notoriously indifferent about the welfare of their subjects, The East India Company drove the expansion of the British Empire in India so much so that by the middle of the nineteenth century the British had direct or indirect control over all of present-day India [19].

4.7 INDIAN FREEDOM MOVEMENT AND PARTITION OF INDIA

Indian Freedom movement refers to a broad spectrum of activities carried out by political organizations, popular uprisings, various national and regional campaigns over a period of 90 years starting from 1857 to overthrow the British rule and secure Independence of India.

The British rule left the populace of India seething with discontent and disgust due to the discriminations practiced by the British which burst out in a revolt by the 'sepoys' at Meerut in 1857 which came to be known as the First war of Indian Independence. [20, 21]

- The Indian Mutiny of 1857 (First war of Indian Independence)

The Indian Rebellion of 1857 was a major turning point in the history of modern India. While affirming the military supremacy of the British, it led to significant change in how India was to be ruled by them. Under the Government of India Act passed in 1858, the Indian Territories came under the direct authority of the British government. In a royal proclamation, Queen Victoria promised equal opportunity of public service under British law, and restoration of the rights of the native princes.

- Gandhian Era and Non-violence (1920-1942)

Mohandas Karamchand Gandhi (Mahatma Gandhi) had been a prominent leader of the anti-discrimination movement in South Africa, and had perfected the concept of satyagraha, a form of peaceful, non-violent struggle to fight for one’s rights. Gandhi’s vision brought millions of regular Indians into the movement, transforming it from an elitist struggle to a national one.

Mahatma Gandhi launched the Non-Cooperation Movement in 1920 followed by the Civil Disobedience movement in 1930 which aimed to withdraw nation’s cooperation from the British Government and cripple the administrative setup of the country. These two movements were great successes as they got massive support from millions of Indians and shook the British Government.
During this movement it was decided that India would celebrate 26th January as Independence Day all over the country. The British Government tried to repress the movement and resorted to brutal firing, killing hundreds of people. Meanwhile the Muslim League in 1940 adopted the *Lahore Resolution*, demanding the division of India into two separate sovereign states, one Muslim, the other Hindu, thus sowing the seeds of partition.

- **Quit India Movement and Partition of India**

  In August 1942, Gandhi launched his famous ‘Quit India Movement' calling upon the British to immediately leave the country and gave a clarion call to his countrymen “*Do or Die*” in their efforts to oust British from India.

  At the conclusion of the Second World War, the Labor Party which came to power in Britain was sympathetic towards the desire of the Indian people for freedom. An interim Government was formed headed by Jawaharlal Nehru. However, the Muslim League refused to participate in the government and pressed for the separate state for Pakistan. Lord Mountbatten, the Viceroy of India, presented a plan for the division of British India into India and Pakistan. At 11:57 on 14 August 1947 Pakistan was declared a separate nation, and at 12:02, just after midnight, on 15 August 1947, India also became an independent nation. Pakistan comprised of two distinct and separate parts the West Pakistan and the East Pakistan which gained independence in 1971 and became the present day Bangladesh.
CHAPTER 5

SOFTWARE PROTOTYPE

Software prototyping is an important part of software development process. Prototypes are incomplete versions of the software application being developed. A prototype simulates initial requirements of the product and may undergo several refinements along the development process. Prototyping is a part of agile software development process.

Prototyping has many advantages: The software design and implementation can get feedback early in the development process. It also allows the software application developer to get some insight into initial project estimates and project completion deadlines.

Dr. Eckberg provided his feedback in the early phase of development which helped estimating project completion deadlines.

The initial prototype for this application was developed as shown in the following figures.

Figure 5.1 shows the initial startup screen of the tool. The startup screen has two navigation buttons. The Colonial India Interactive Tool button takes the user to the actual application which has all the map rendering and map layers.

The Website button takes the user directly to the HTML pages which provide more information about the colonial period of India and the freedom movement.

Figure 5.2 is the screen shot of the map tool of colonial India. The tool comes with the base map layer of ancient India. The user can select other map layers to know more about the European nations who had colonized India.

The tool also has many menu items whose functionality is explained in detail in the further sections of this document. The tool also contains many button which links to HTML pages which provide all the information about European colonies in India.

Figure 5.3 is the screen shot of the HTML pages which is a part of the colonial India website. This HTML page talks about the advent of Europeans with the discovery of a sea route by Portuguese sea farer Vasco Da Gama. The website also includes audio and video clips about some important milestones of the Indian freedom movement.
Figure 5.1. Application startup page.

Figure 5.2. Map base layer.
Figure 5.3. Colonial India webpage.

Vasco da Gama (1469 - 1524) was the Portuguese explorer who discovered the sea route to India from Europe through the Cape of Good Hope. Vasco da Gama sailed his ships out into the Atlantic Ocean and eventually coming within 600 miles of South America. The fleet sailed through St. Helena Bay and continued on to the Cape of Good Hope. He then made his way up the eastern coast of Africa and reached Malindi. He then left Malindi and set out to cross the Indian Ocean. Da Gama was successful in crossing the Indian Ocean and anchored off the city of Calicut, India in 1497.

Da Gama's second voyage to India was in 1502 and was made up of 20 ships. During this voyage, he landed on the city of Calicut. He was able to sign trade treaties with the rajahs in the cities of Cochin and Cannanore and thus established trade relations between Portugal and India. The discovery of sea route from Europe to India led to the establishment of European colonies in India.
CHAPTER 6

SOFTWARE ARCHITECTURE

The software architecture refers to the high level structure of any software application. It defines the properties and components of the high level subsystems. It also denotes the relationships between the subsystems.

This chapter talks about the architecture details of this application. The architecture diagram (Figure 6.1) shows all the components of the colonial India application.

- The startup window has two buttons: colonial India tool and colonial India website.
- The UI of colonial India tool is divided into the left panel consisting of the table of contents whose legends name and list the map layers, a top panel having all the menus and toolbars, and the right panel consisting of buttons to link to the colonial India website, and finally the bottom panel displaying the coordinates of the mouse pointer.
- Each legend in the TOC corresponds to a shape file(.shp) which is deployed on to the map using Map Objects. Map rendering also requires .dbf and .shx files.

Figure 6.1. Architecture diagram.
• Some shape files were created using ArcView and Diva GIS shape file editor.
• The End of British Rule button when clicked dynamically adds 3 more layers to the legend panel which depicts the partition into India, East Pakistan and West Pakistan.
CHAPTER 7

ECLIPSE ENVIRONMENT SETUP

This section talks about how to configure the Eclipse developer environment. The executable file of Eclipse IDE and the installation instructions can be found from the Eclipse website. Next is installing the MapObjects Java Edition.

MapObjects has to be integrated to Java by importing the ESRI MapObjects Libraries. This requires importing of two JAR files. Finally run configurations include defining the main class and environment variables if needed.

The Figures 7.1, 7.2, 7.3 and 7.4 show the step by step setup configuration.

![Figure 7.1. Adding ESRI library.](image-url)
Figure 7.2. Adding external JAR files.

Figure 7.3. Selecting the JAR files.
Figure 7.4. Setting the main class and configuring other run time parameters.
The toolbars are implemented to provide a better interaction for the users with the application. The toolbars provide more options and can offer many interactive functionalities.

This section talks about the components of the toolbar and their respective functionalities. Figure 8.1 shows the MapObjects toolbars.

![MapObjects toolbars](image)

**Figure 8.1. MapObjects – toolbars.**

The toolbars are categorized into three types based on the functionality they provide. Each of them is explained in detail in the following sections.

### 8.1 ZoomPan Toolbar

ZoomPan toolbar is implemented to provide important capabilities like zoom in and zoom out for the displayed map. There is also a pan tool. Table 8.1 describes the different features of ZoomPan toolbar. Figure 8.2 shows ZoomPan toolbar.

### 8.2 Selection Toolbar

Selection toolbar provides the capabilities of creating queries on the selected layer. The user can also search for a specific information about a layer with the help of selection toolbar tools. Table 8.2 describes the different features of Selection toolbar. Figure 8.3 shows the Selection toolbar.

### 8.3 Custom Buttons

This tool bar is created to add additional functionalities for an user make a more powerful application tool. This toolbar has options for adding new layers, drawing points on a map, projecting CSV (Comma separated values) files on to the map etc. The Figure 8.4 shows the custom tools available.
Table 8.1. ZoomPan Toolbar Tools

<table>
<thead>
<tr>
<th>Tool</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Previous Extent</td>
<td>Zooms the map to the previous extent stored</td>
</tr>
<tr>
<td>Next Extent</td>
<td>Zooms the map to the next extent</td>
</tr>
<tr>
<td>Zoom to Active layer</td>
<td>Zooms the map to all the selected features of the current layer</td>
</tr>
<tr>
<td>Zoom to full extent</td>
<td>Brings the active layer to original size</td>
</tr>
<tr>
<td>Zoom In</td>
<td>Zooms in the map</td>
</tr>
<tr>
<td>Zoom Out</td>
<td>Zooms out of the map</td>
</tr>
<tr>
<td>Pan</td>
<td>Moves the map in any direction, useful when dealing with larger maps</td>
</tr>
<tr>
<td>Pan in one direction</td>
<td>Pans in any one of the direction selected</td>
</tr>
<tr>
<td>Identify</td>
<td>Identifies the area pointed in the active layer and provides a brief tabular description of the feature layer</td>
</tr>
</tbody>
</table>

Figure 8.2. ZoomPan toolbar.

The following are the functions of Custom Toolbar:

- **Point tool:** Provides the user with the capability to place a point on the map.
- **XY tool:** This tool facilitates to create a layer dynamically from a CSV file. The code to add a CSV file is taken from Dr. Eckberg's book about GIS and it works for 3 fields: latitude, longitude and city. The following screenshots show how to add a CSV file [3]. The Figure 8.5 shows how to add a CSV file. See Figure 8.6, which shows the feature layer added after selecting the CSV file.
- **Add layer tool:** This tool lets the user add a shape file layer to the existing application. This supports .shp formats only.
- **Arrow tool:** Used to get back the cursor to the default mode, after using other tools from the tool bars.
- **Hotlink tool:** The lightning bolt icon is used in many GIS applications. The hotlink tool provides additional information about the selected layer. It opens a
### Table 8.2. Selection Toolbar Tools

<table>
<thead>
<tr>
<th>Tool</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Find</td>
<td>Opens up a dialog box with all the layers for locating features based on the user inputs</td>
</tr>
<tr>
<td>Search</td>
<td>Opens a dialog to locate features based on a predefined stored query</td>
</tr>
<tr>
<td>Query Builder</td>
<td>Lets the user create a query and displays the query results on the map layer</td>
</tr>
<tr>
<td>Select</td>
<td>Selects feature by choosing any of the polygons available by making a layer active and drag the mouse to select the portion of map.</td>
</tr>
<tr>
<td>Buffer</td>
<td>Creates a buffer polygon for currently selected features</td>
</tr>
<tr>
<td>Attributes</td>
<td>Displays attributes of currently selected layer</td>
</tr>
</tbody>
</table>

**Figure 8.3. Selection toolbar.**

**Figure 8.4. Custom buttons.**

- dialog box which can contain images, description or external hyperlinks which provide more information about the layer.
- **Print tool:** This a very important and a powerful feature of Map Objects application which lets the user to print the required data from the application, when the user is connected to printer source.
Figure 8.5. Selecting CSV file after enabling XY button.

Figure 8.6. Feature layer added after selecting CSV file.
CHAPTER 9

SHAPEFILES AND FEATURE LAYERS

9.1 CREATING SHAPEFILES

There are various tools available to create shape files given a set of geographical coordinates.

During the development of this application, the shape files were created using ArcView and Diva GIS softwares. These softwares take a CSV file or a special .dbf file as input and generate the required shape file.

Map Objects Java Edition also provides some modules to create a shape file from an existing shapefile. The code for implementing this feature is taken from Dr. Carl Eckberg's book about GIS applications and Map Objects. The Figure 9.1 shows how to create a new shape file using Map Objects modules [3].

Figure 9.1. Create layer from selection.

The user has to enable the Select tool to select the parts of the layer from which a new shape file has to be created and then navigate through theme->create layer from selection.
The shapefile records can also be edited using some of the popular DBF editors and the records can be appended with the required information at a later point of time. While editing the records, the user can add only columns but not rows.

9.2 FEATURE LAYERS

A layer conceptually represents a slice of geographical data of a particular area. In this application each legend item in the legend TOC corresponds to each display layer. A feature layer describes a set of feature data. The feature data represents geographic entities in the form of points, lines, and polygons.

In this application each European colonial power is represented as a feature layer. Each feature layer. Figure 9.2 shows British India feature layer.

![British India layer](image)

Figure 9.2. British India layer.

Figure 9.2 shows the British provinces in India; this is a polygon layer and the rendering of the layer is done with the help of Map Objects modules. The following code snippet shows how to access the renderer of a polygon shapefile, and change it to be rendered in red.

```java
java.util.List list=toc.getAllLegends();
com.esri.mo2.map.dpy.Layer lay1=((Legend)list.get(0)).getLayer();
FeatureLayer flay1=(FeatureLayer)lay1;
```
Figure 9.3 shows Dutch territories in India; this is a point shape file layer. Each point layer has a different marker symbol, which makes it more simple for the user to identify a particular layer.

![Figure 9.3. Dutch India layer.](image-url)

The French colonies are marked with black triangle markers and the Danish colonies are marked with pink circle markers. The following code snippet shows the rendering of point shapefiles. Figure 9.4 shows French and Danish colonies.

```java
java.util.List list = toc.getAllLegends();
com.esri.mo2.map.dpy.Layer lay1 = ((Legend)list.get(0)).getLayer();
FeatureLayer flay1 = (FeatureLayer)lay1;
BaseSimpleRenderer bsr5 = (BaseSimpleRenderer)flay5.getRenderer();
SimpleMarkerSymbol sms2 = new SimpleMarkerSymbol();
sms2.setType(SimpleMarkerSymbol.TRIANGLE_MARKER);
sms2.setSymbolColor(java.awt.Color.black); // green
sms2.setWidth(9);
bsr5.setSymbol(sms2);
```
Figure 9.4. French and Danish colonies.

Figure 9.5 shows the partition of pre-independent India. These layers are added dynamically with the click of a button. The **End Of British Rule** button is clicked and the map layers are added to the TOC. The map rendering shows post-independent India shaded in red and east and west Pakistan shaded in blue. These are thatched semi-transparent layers.

### 9.3 Promoting Layers

The TOC contains legends for all the layers; the ordering of these layers can be manipulated using the layer control tool. A layer that is selected can be promoted or demoted, i.e., can be moved up or down in the TOC. The Figure 9.6 shows the layer control.
Figure 9.5. Post independent India and Pakistan.

Figure 9.6. Promoting/demoting layers.
CHAPTER 10

SCREENSHOTS

This section shows the various screen shots of the application. Figures 10.1, 10.2, 10.3 and 10.4 show the screen shots of the colonial India website.

Figure 10.1. Danish India webpage.

Figure 10.2. Dutch India webpage.
Figure 10.3. French India webpage.

The French East India Company was formed in 1664 at the request of Minister Colbert in the reign of Louis XIV. The French company was created, controlled and financed by the state and it differed from the British East India Company which was a private enterprise. The French were the last of the European powers to compete for commercial gains in the East with the rest of the European Companies. They established their first trading outpost at Surat in 1668. Later they started several trading centers at Masulipatam, Chandranagore, Yanam, Mahe, and Karikal. In 1673 Francois Martin obtained a little village from the Muslim governor of Valikondapuram and developed it into an important place. Thus was Pondicherry born, which was their most important settlement and served as their seat of power and capital in India.

Figure 10.4. Freedom movement webpage.

Mohandas Karamchand Gandhi (Mahatma Gandhi) had been a prominent leader of the anti-discrimination movement in South Africa, and had perfected the concept of satyagraha, a form of peaceful, non-violent struggle to fight for one’s rights. Gandhi’s vision brought millions of regular Indians into the movement, transforming it from an elitist struggle to a national one. Mahatma Gandhi launched the Non-Cooperation Movement in 1920 followed by the Civil Disobedience movement in 1930 which aimed to withdraw nation’s cooperation from the British Government and cripple the administrative setup of the country. These two movements were great successes as they got massive support from millions of Indians and shook the British Government. During this movement it was decided that India would celebrate 26th January as Independence Day all over the country. The British Government tried to repress the movement and resorted to brutal firing, killing hundreds of people. Meanwhile the Muslim League in 1940 adopted the Lahore Resolution, demanding the division of India into two separate sovereign states, one Muslim, the other Hindu, thus sowing the seeds of partition.
CHAPTER 11

CONCLUSION AND CHALLENGES

This thesis presents an interactive GIS based application to provide information about the colonial era of India and the freedom movement. This application represents the geographical boundaries of each of the European nations that held colonies in India. There is also a website which is a part of this application which contains a brief description about each of these European colonies. The website is also supported with images, audio, and video clips which makes the learning more attractive and entertaining.

Finally, this application is successful in capturing and presenting most of the important events of the colonial era and the freedom movement of India. This GIS application is easily deployable on any machine and can be extended to be a part of various history applications.

The main challenge faced during the development of this application was collection of data for the shape files. It was hard to find the shape files representing the geographical territories, of the European colonies. To overcome this hurdle, all the coordinates were carefully collected and each shape file was created. Some shape files were created using ArcView, some were created using Diva GIS and rest of the shape files were created using Map Objects modules.

Another challenge was to concisely present the relevant information to the users who are totally new to Indian history. Given the vastness of this topic, it was quite hard to choose the relevant details which give a bird's eye view of the colonial era of India.
CHAPTER 12

FUTURE ENHANCEMENTS

This GIS application provides information about the colonial period of India, satisfies all the requirements envisioned at the start of this project. But there is also a scope for future enhancements which can refine the learning process and aid in imparting knowledge.

- The application tool can be made available in many languages to provide access to the users with limited knowledge of English.
- This application can be extended for tablets and mobile phones as a web application.
- Quizzes can be made a part of the application, which makes the learning more fun.
- Timelines can be added to the maps which provide more information about a particular period.
- Feedback and blog sections increase active participation and discussion among the users and is also a valuable source of suggestions for the improvement of the application.
REFERENCES


