**Awesome Alphabet**

**Software Design Document (SDD)**

Vivek Goyal - Michael Grant - Jaleel Kazi - Mark Musante - Levi Paul

Professor Yuting Zhang

Boston University

Spring 2013

MET-CS-673-W4

Software Engineering

February 24, 2013

# Revision History

|  |  |  |
| --- | --- | --- |
| **Revision** | **Date** | **Description** |
| 1.0 | Feb 24, 2012 | Original Version. Matches current Iteration 1 software. |

# Introduction

The document discusses the design of the Awesome Alphabet educational software. This includes the architecture, design patterns, classes, and methods used in the software.

Since Awesome Alphabet is continually evolving, this is a living document. As the software is changed, this document will be updated.

For additional design and project information, please consult the Unified Modeling Language (UML) diagram and other project documentation.

# User Interface

The Awesome Alphabet user interface currently consists of three main pages: the Title Page, the Alphabet Page, and the Letter Page.

## Title Page

The Title Page is the first page the user will see when the application launches. On the page, the user will see the application’s name and version number and will be able to navigate to the Alphabet Page.

## Alphabet Page

On the Alphabet Page, the 26 letters of the English alphabet will be displayed in upper and lower-cases. The user can navigate to the corresponding Letter Page by clicking on of the letters. In addition, the user can play the alphabet song and navigate back to the Title Page.

## Letter Page

On the Letter Page, the user will be able to see a letter in both upper and lower-case. When the user first sees the page, the name of the letter will be pronounced. Accompanying the letter, will be a word and picture. Any time the user clicks on the letter, a phonetic pronunciation of the letter will be played. In addition, when the word or picture is clicked, the user will hear the name of the word. The user may see another word and picture by clicking on the Get Next Example button. Finally, the user may navigate to the previous letter, next letter, Alphabet Page, and Title Page from the Letter Page.

# Architecture and Design Patterns

As stated in other documents, Awesome Alphabet will be programmed using the Java language, targeting the Java Runtime Environment (JRE), and using Java Swing for the user interface (UI).

The UI will consist of a main window containing a Card Layout. The Card Layout will manage the different pages and allow a single page to be displayed at any given time.

To separate the application data from the views, a Model-View-Controller (MVC) architecture will be used. Each UI page will be represented by a View class (i.e. AlphabetPageView). In addition, there will be a series of model classes, including the Alphabet class, Letter class, and other supporting classes. To act as an intermediary between the views and models, there will be a series of Controller classes, including AlphabetPageController, LetterPageController, etc. For now, there will be a single controller associated with each view.

To assist with the MVC architecture, the Observer Pattern will be used. This pattern will allow the models to notify the views when they have changed. The views can then update themselves by requesting the latest data values. The Observer Pattern will also allow the controllers to request a page change in an abstract manner so that they do need to know the details of the main window.

In addition to the design approaches listed above, inheritance will be used to consolidate common features among classes. For example, all views will inherit from the abstract PageView class and all controllers will inherit from the abstract PageController class. This will eliminate duplication of page name and panel management code in the view classes and page navigation code in the controller classes.

The final design topic worth addressing is the models. The primary two models for Awesome Alphabet are the Alphabet model and Letter model, implemented as the Alphabet and Letter classes, respectively. The Alphabet model consists for 26-Letter objects, one for each letter in the English alphabet. In addition, it has the ability to keep track of the currently selected letter and change the letter selection. The Letter model keeps track of the letter it represents and stores a list of words, with associated pictures and sounds, which starts with that letter. It also keeps track of the most recently viewed word and provides methods for playing sounds and obtaining model data.

# Classes and Methods

To see the classes and methods for Awesome Alphabet, please consult the Javadoc located on the GitHub project page or on Google Drive.