Joe Broder

Mr. Tucker

CS 172

23 June 2017

Project Specifications

The problem I am solving is that there is no desktop image viewing application that specializes in reading manga.

My manga reader application will require the following capabilities at a **minimum:**

* A simple GUI.
* Ask the user for the name of a directory (where a book is located).
* The ability to display images.
* The ability to sort, and then display the directory’s images in the GUI.
* Buttons to show the next and previous page of a manga.

Based on the development process, **extra features may include:**

* Browsing for a directory.
* Displaying thumbnails for images.
* Allow the user to click on a thumbnail.
* Save the location of a book (using a text file).
* Save the page left off on for each book.

To meet the **minimum requirements**, my app will need an image loader class, along with any other classes required for GUI objects. I am currently learning Qt for GUI development. The Image loader will need the ability to access files in a given directory, and store the paths to those files in a vector of strings. The image loader will then need to supply GUI elements (whatever displays the image) with the path to the desired file(page in the book). This class will contain an integer value to keep track of the current page.

As seen in the UML diagram, the image loader class (ImageLoader) will have two values, a vector<string> to store paths of images, and an integer to store the current page(used to index the vector). The class will have accessor and mutator methods for the integer, a method to return a path to an image for a given index of the vector, and a method to clear and then populate the vector with paths to images for a given directory.

Joe Broder

Mr. Tucker

CS 172

29 June 2017

UPDATED Project Specifications

Throughout the development process, many of my project specifications were modified. I was able to meet all the **minimum** requirements for my project; however, I was also able to include the following **additional** features:

* Open a dialogue to browse for a manga directory, rather than have to manually type in a file path.
* Buttons to jump to the first and last pages of the manga.
* A button and textbox to jump to any page in the manga.
* Ability to zoom in and out on images.
* Ability to reset the zoom of images (set back to 100%).
* Ability to scroll on large images.
* Ability to pan/kinetic-scroll on large images via mouse clicking and dragging.

The classes used for this application also had to be heavily modified to meet these features. First, a **MainWindow** class was added to render the main window of the application in Qt. This class contained an object of the **ImageLoader** class (used to load manga), a **scene** variable used to hold the current image being displayed, a method to set the page image to that of the current **Imageloader** page, and the required methods to represent ui events (button clicks, ect.).

The **ImageLoader** class had to be heavily modified as well. I changed the main data structure used to hold images from a vector<string>, to a vector<QPixmap>. The QPixmap class is how Qt represents images. The load method utilized QDir (used by Qt to represent a directory) to create a QPixmap for each item in the directory, and pushes it back on the vector. A scaleFactor was added for zooming (along with mutator methods), and upon returning a page (QPixmap type), the object has it’s size multiplied by the current scale factor.