DITAA Design

Version 1.0

November 25, 2020

Diagrams Through ASCII Art Specifications

Kevin Porter
Adam Shiveley

Table of Contents

1.0. Introduction

1.1 Purpose

The purpose of this document is to describe the overall design for the Designs Through ASCII Art (DITAA) program. The DITAA program has been explained in full in a previous documents named "Software Requirements Specifications" and "DITTA Specs" found at the GitHub address: https://github.com/dimeandpenny/CS_7140_Group_Project.

1.2 Scope of Document

This document will describe the general design of the DITAA program. The document will include the overall flow of the program and the Graphical User Interface (GUI) that the program will implore.

2.0. System Architecture

2.1 Architectural Design

The program will be written in Java version 8. No custom of special plugins will be used. The design goal is to allow the program to operate on any modern computer without installing various need packages and addons.

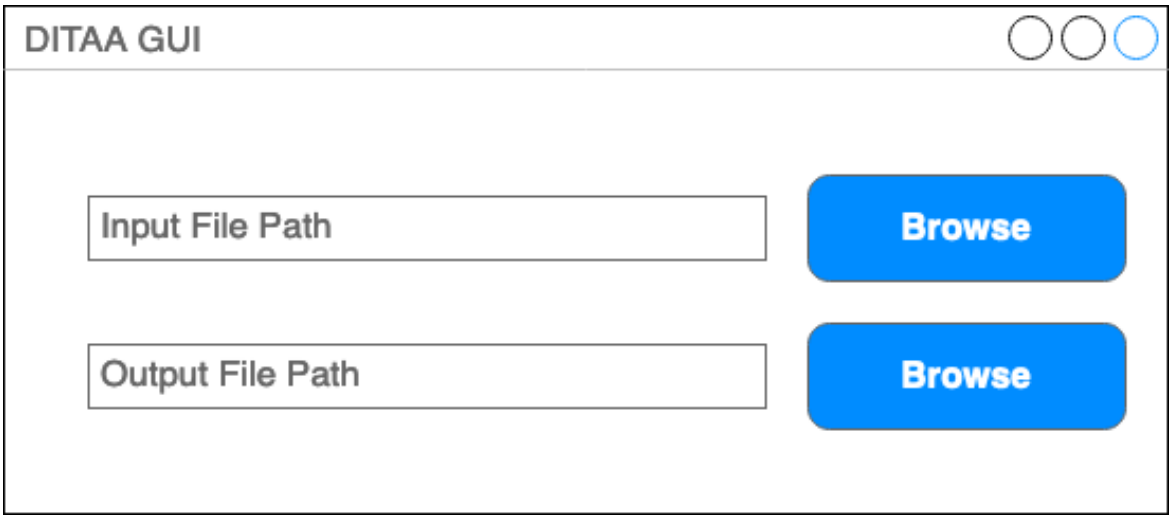
2.2 Design Rational

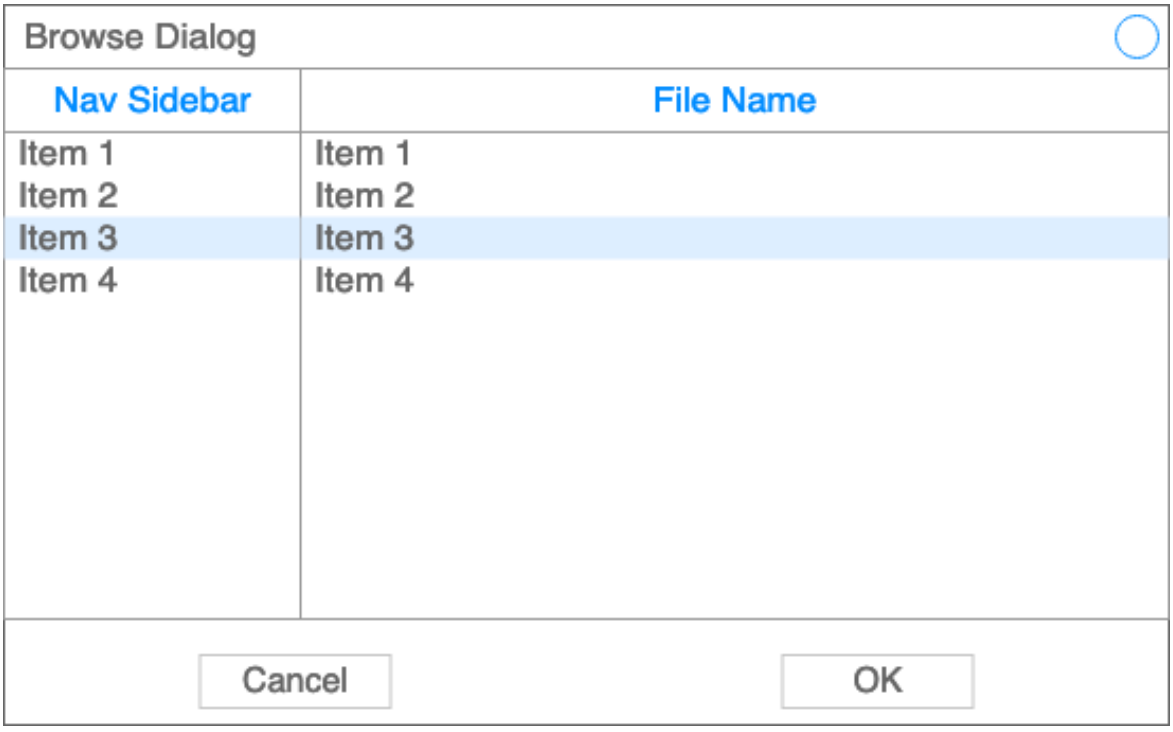
The DITAA program can be divided into 2 separate parts: User Interface and computational program. The User Interface section is described in detail in Section 3.0 of this document. The computational portion of the program will be described in this section. The general requirements and specifications of the program have been described earlier as noted in Section 1.1 of this document. The program will use the Very High Level Language (VHLL) Pseudo code from the specification document to convert the ASCII art to a graphical representation.

3.0. Human Interface Design

The DITAA program will implore a Graphical User Interface (GUI) to facilitate the ease of user input. The GUI will consist of multiple fields. One field, a text box, will be used in

conjunction with a “Browse” button to enable users to select which ASCII file they want to convert. Another field will contain a text box field and a “Browse” button to enable the user to choose where they will save the final result. Wireframe mockups of the GUI elements are given below.

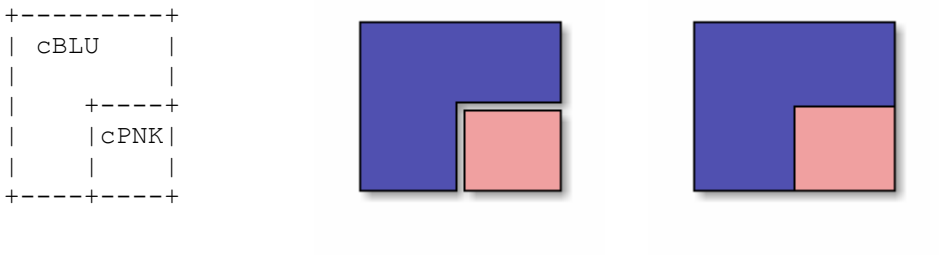




The DITAA program will also facilitate the usage of a command line input feature. This mode of the program will allow a more advanced user to control various elements of the program through command line flags. An example and description of these flags can be found below:

```
-A,--no-antialias          Turns anti-aliasing off.
-d,--debug                 Renders the debug grid over the resulting
```

	image.
-E,--no-separation	Prevents the separation of common edges of shapes. You can see the difference below:



Before processing | Common edge separation (default) | No separation (with the `-E` option)

-e,--encoding <ENCODING>	The encoding of the input file.
-h,--html	In this case the input is an HTML file. The contents of the <pre class="textdiagram"> tags are rendered as diagrams and saved in the images directory and a new HTML file is produced with the appropriate <img> tags. See the HTML section.
--help	Prints usage help.
-o,--overwrite	If the filename of the destination image already exists, an alternative name is chosen. If the overwrite option is selected, the image file is instead overwriten.
-r,--round-corners	Causes all corners to be rendered as round corners.
-s,--scale <SCALE>	A natural number that determines the size of the rendered image. The units are fractions of the default size (2.5 renders 1.5 times bigger than the default).
-S,--no-shadows	Turns off the drop-shadow effect.
-t,--tabs <TABS>	Tabs are normally interpreted as 8 spaces but it is possible to change that using this option. It is not advisable to use tabs in your diagrams.
-v,--verbose	Makes ditaa more verbose.

4.0. Testing

The DITAA program will be tested against known test cases. Those test cases are not yet fully developed.