DITAA Implementation Document

Existing Implementation

DITAA was originally implemented as a headless Java application. The application is built by compiling the Java source into classes and bundling them into a runnable JAR file. The JAR file manifest specifies a main class, CommandLineConverter, which is invoked when the JAR is run. The CommandLineConverter logic is contained mostly in a large main method, which parses command line parameters and then runs the DITAA conversion on either the single input file or the input files referenced in HTML if running with the --html flag.

Classes

Below is a listing of the major classes in the DITAA implementation along with a brief description.

CommandLineConverter:
Entry point for the program. Parses command-line arguments and calls the other major parts of the program to turn ASCII art into bitmaps. This primarily entails constructing a TextGrid from the input, using it to construct a Diagram, and feeding that Diagram to a BitmapRenderer.

ConfigurationParser
Reads and parses a config.xml. Used by ConversionOptions

ConversionOptions
A struct holding miscellaneous options. Holds a ProcessingOptions and a RenderingOptions.

DebugUtils
Can get the line number currently being executed.

DocBookConverter
Unused, and deleted from our version of DITAA.

FileUtils
Boilerplate for handling file/path names.

HTMLConverter

Has only one public method, convertHTMLFile, which loops through an HTML document, converts any ASCII art found in tags labeled “textdiagram”, and creates a new version of the HTMLFile where those diagrams are real pictures.

JavadocTaglet
Allows ditaa diagrams to be involved in javadoc comments. Not actually used by any other part of the current implementation.

Pair
Generic class representing a 2-tuple.

PerformanceTester
Contains a main() which tests how fast DITAA is.

ProcessingOptions
A struct with miscellaneous options about how TextGrids will be turned into Diagrams, but also other eclectic settings of the program--basically anything that doesn’t go in RenderingOptions.

RenderingOptions
Small struct with options affecting how Diagrams are drawn by BitmapRenderer. Examples include whether to use drop shadows and anti-aliasing.

Shape3DOrderingComparator
Comparator that allows DiagramShapes to be compared for sorting purposes. Used by BitmapRenderer to make sure the right things are rendered on top of the right things.

VisualTester
Has a main() which runs some tests.

BitmapRenderer
One of the most important classes. Responsible for taking Diagrams and turning them into bitmaps.

CompositeDiagramShape
A DiagramShape which wraps an array of other DiagramShapes. This is used in practice to represent ‘spiderwebs’, or connected components of open lines with intersections and multiple dead ends.

CustomShapeDefinion

A small struct which encodes how to render a custom (user-defined) shape, usually by pointing to an SVG file somewhere.

Diagram
The god class of DITAA. Contains the bulk of the code for actually processing input. Once constructed, this class exposes collections of DiagramShapes and TextObjects which can be used by clients like BitmapRenderer to easily draw any diagram; the constructor is enormous, and does most the heavy-lifting of taking raw ASCII text and turning it into DITAA's internal representation of diagrams. We do a lot of work on this class, including refactoring and several bug fixes.

DiagramComponent
The base class of all 'parts' of a Diagram. Once the input is processed, a diagram basically comprises a collection of these.

DiagramShape
A DiagramComponent that represents shapes (non-text entities, basically) that can exist in a diagram. Has a should-be-an-enum indicating its type--an arrow, a trapezoid, etc. Can also return a 'renderPath', an outline essentially, which is most of what you need to know to draw a given shape.

DiagramText
A DiagramComponent representing a block of text.

FontMeasurer
Figures out how big strings will appear, in pixels, when rendered in a certain font. Used by Diagram.

ImageHandler
Singleton class that has to do with making SVGs. Used by BitmapRenderer. Handles pre-defined shapes like 'trapezoids' that are packaged with DITAA as SVGs.

OffscrenSVGRenderer
Renders SVGs.

ShapeEdge
A (directed, though it may not always matter) edge belonging to a particular DiagramShape.

ShapePoint

An x,y point with some methods for checking geometric properties about it.

AbstractCell

This is used by Abstraction Grid to create ‘upsampled’ representations of TextGrids. It has 3x3 int[] representations for various shapes.
A horizontal line, for example, is
000
111
000

Whereas a cross is
010
111
010

AbstractionGrid

When you instantiate an AbstractionGrid, it takes a TextGrid and creates an ‘upsampled’ 3x3 representation of it. Every character in the original TextGrid is represented by a 3x3 block of characters which are either blanks or asterisks.

This allows flood-filling algorithms to detect the space in between adjacent line or corner characters. It also allows for the adjacency to adjacent characters to be detected intelligently: two adjacent pipes “||” are not part of the same component, because the AbstractionGrid reveals that they don’t actually “touch” the way two dashes “--” do.

It’s actually rather clever, and a crucial part of the logic in the Diagram constructor.

CellSet

A set of TextGrid.Cells. Despite the simple premise, this is a relatively large class, as it contains many methods for performing both manipulations of the set and querying high-level properties, such as whether the set represents an open or closed curve. A non-trivial chunk of the logic for processing ASCII inputs ultimately lives in this file.

GridPattern

This is essentially a regular expression for 2D text, which can be used to detect certain multi-character shapes in TextGrids.

GridPatternGroup

A static library of predefined GridPatterns for the kinds of patterns DITAA is interested in, such as ‘dead ends’ and different orientations of corners.

StringUtils
Simple utilities for Strings.

TextGrid
While we believe Diagram is DITAA's "God class", TextGrid is physically the largest class in DITAA (with over 100 methods!).
In a nutshell, it represents a 2D grid of ASCII text. But it also contains methods for all sorts of complex manipulations and queries on this grid, ultimately containing a significant chunk of DITAA's logic.

Maintainability Improvements

In re-writing to improve maintainability, we refactor Diagram and BitmapRenderer, primarily by decomposing their largest methods into several methods of more manageable size, and documenting existing methods with contracts. In the process, we fix several bugs in Diagram resulting from the interaction of all the shape processing strategies (these fixes are detailed in Testing).

We make a public CellSet method (addAll) private because clients should only use addSet instead.

We remove unused code from BitmapRenderer.

We add asserts reflecting the contracts in Diagram and BitmapRenderer where feasible.

Other opportunities that were identified for improvement but not completed include:

- Turning the ints in DiagramShape into actual enums
- Why is the factory for closed shapes in DiagramComponent but the one for open shapes in CompositeDiagramShape? This seems inconsistent.

Java Version

The existing project was implemented against Java 1.6, which is extremely outdated (released in 2006). While this limits the amount of "modern" Java conveniences that are available, it does maximize the compatibility of DITAA in dated environments. As a compromise to retain maximum compatibility, which may be an important requirement for an ascii-based diagramming tool, the project was only incremented a single version to use Java 1.7 (released in 2011). This allows use of now-standard constructs such as try-with-resource blocks and handling multiple exceptions in a single catch clause.

Implementation of New Plugins

New Color Codes

Human-Readable color codes in DITAA are implemented using an in-code static dictionary from human-readable codes to hex RGB codes. This dictionary is statically initialized in the TextGrid class. We expand this table with additional entries using the same syntax.

GUI

Overview

The existing DITAA application is a headless application with a command line interface. To increase the usability of the tool, a GUI was developed. The GUI is accessed in one of two ways, either by running DITAA with no arguments (no input filename) or by running DITAA with the --gui/-g argument. The latter method supports the use of additional arguments to DITAA which will be used by the GUI each time a new source file is loaded.

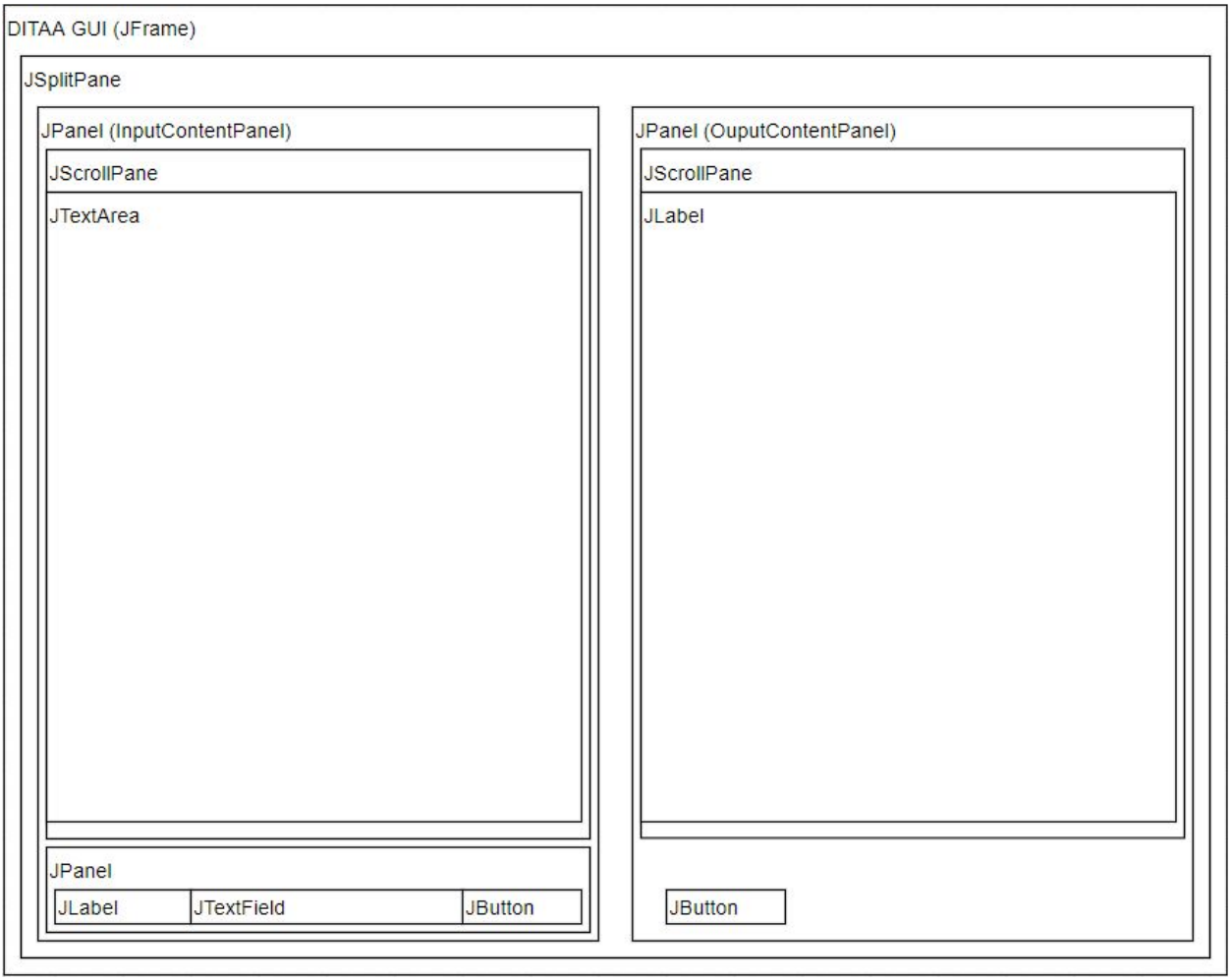
A single new class called DitaaGUI was added to encapsulate the GUI. The logic contained in the GUI is all private to this class and should not be exposed. The only public-facing method in the class is the method which shows the application, openGUI().

Creating and opening the GUI can easily be done in a single line, e.g.

```
new DitaaGUI(options).openGUI();
```

Swing Components

The GUI is implemented using the Swing library. This library provides a number of widget implementations (called “Components”) that can be used to build GUIs. The DITAA GUI implementation is a relatively simple arrangement of Components:



A JFrame contains the entire GUI in an external window. Within that JFrame is a JSplitPane, which provides the left (input) and right (output) content sections. Each section is contained within its own JPanel. The input text preview is provided by a JTextArea and the output image preview is provided by a JLabel. Each of these are wrapped in a JScrollPane which is configured to provide scrollbars if the content size exceeds the container size.

Integration with DITAA

It was desirable for the implementation of the GUI to rely as much as possible on the existing DITAA implementation. That is, care was taken to ensure that code was not duplicated from the existing CommandLineConverter class. This guarantees that as the command-line version of DITAA is updated, those updates are carried over and function equivalently in the GUI version.

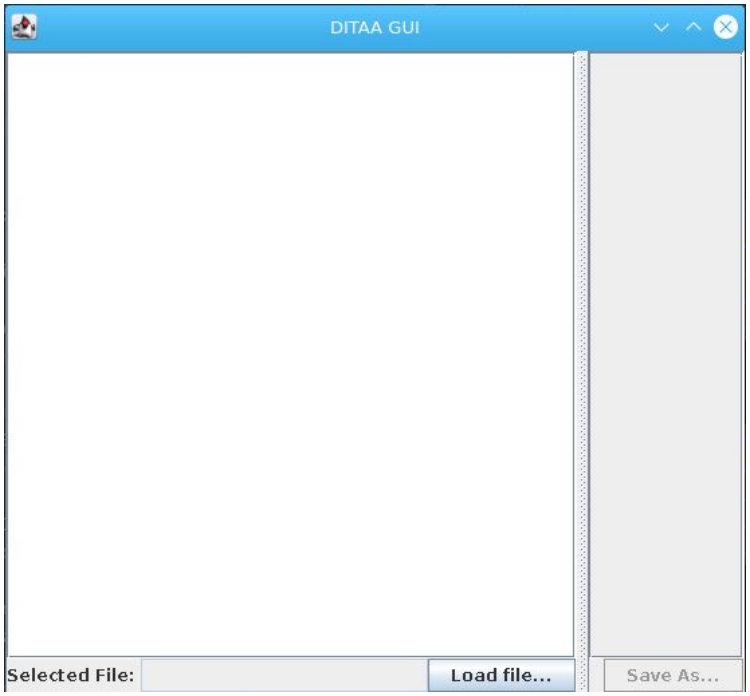
The CommandLineConverter class contained a main function that was run when invoking the DITAA JAR headlessly. Unfortunately, this main function contained a lot of the logic of parsing command line arguments, setting options, and then running DITAA. In order to re-use this implementation, the logic involved with invoking DITAA and saving the output was moved to a new runSimpleMode method. (In this context, “simple mode” indicates running DITAA without the --html option, which is incompatible with the GUI since it does many things at once).

The existing main function was augmented to include the new ‘--gui’ argument and corresponding usage description as well as the logic to invoke the GUI if the option was provided in the command line arguments.

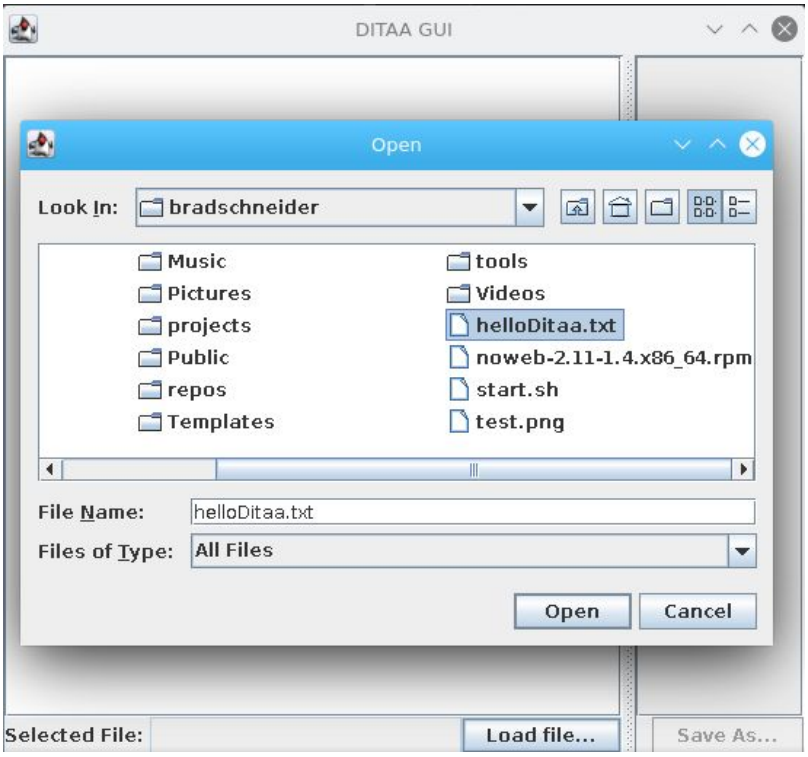
With these changes, it was possible for the GUI implementation to run DITAA programmatically in the exact same way that the headless interface does, maximizing code re-use and maintainability. Another option would have been to implement a completely separate GUI executable which delegated DITAA processing to a compiled version of DITAA (e.g. by running a process which called the JAR file in the background), but this architecture suffers from limited error information being available between processes and was avoided.

Usage

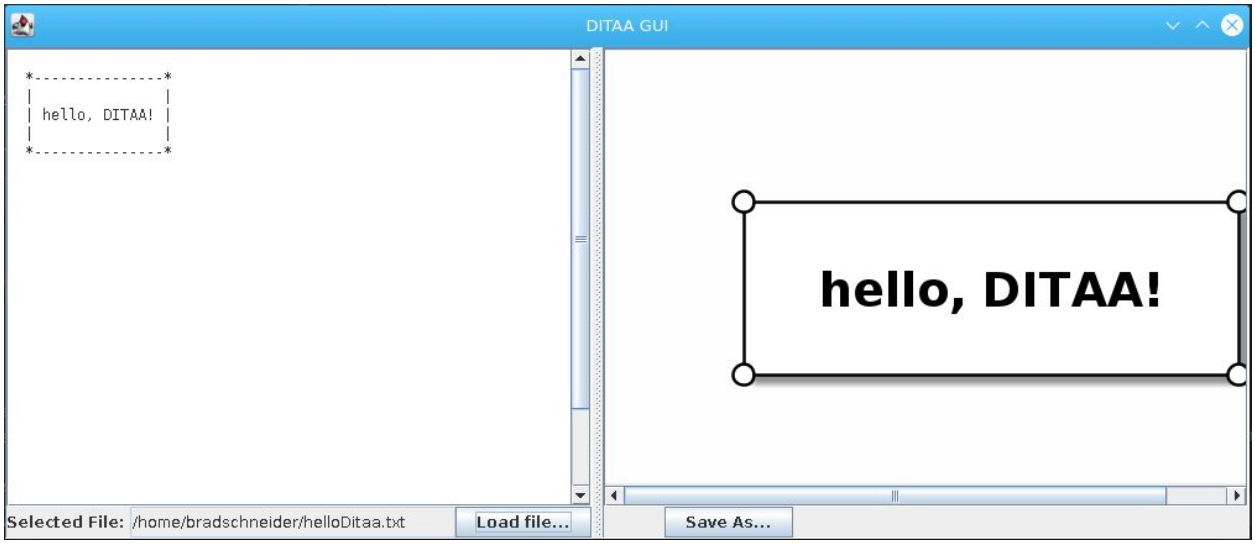
The GUI opens to blank input/output panels as shown below:



The “Load File” and “Save As” buttons are implemented with standard Swing FileChoosers, which are familiar to most users:



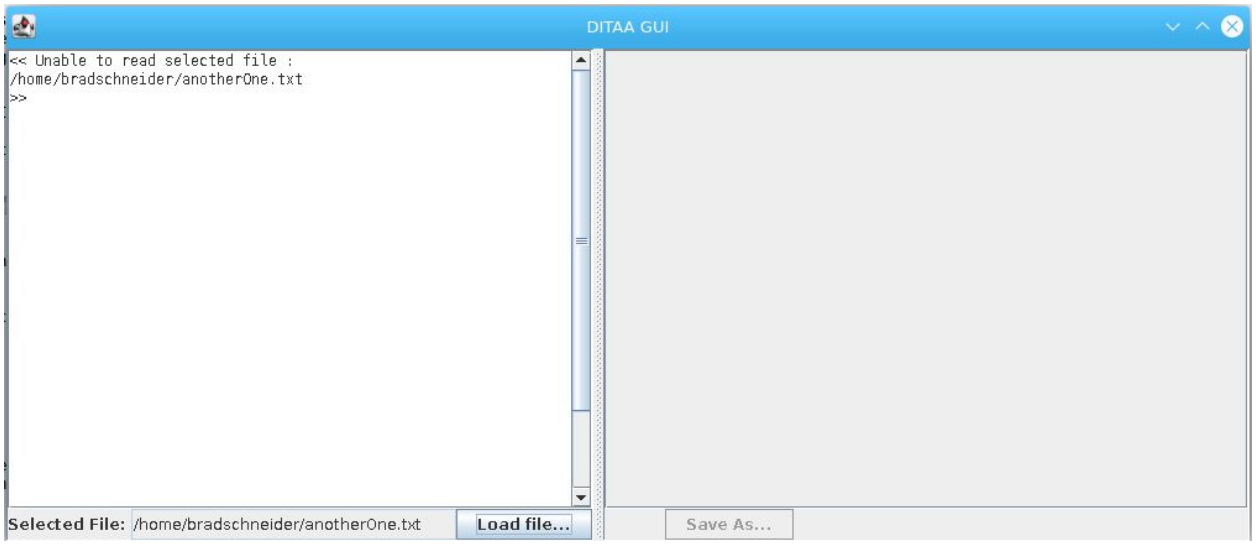
Once a text file is loaded using the “Load file” button, it’s contents are displayed in the file preview pane on the left. DITAA is automatically run in the background and the output is saved to a temporary directory for preview, which is loaded in the panel on the right side of the layout:



It is worth noting that both the input and output preview areas are scrollable if they exceed the window's current size, and the divider between the two panels is adjustable so that the ratio between the left and right panes is able to be modified for easier viewing.

The output images are computed in a temporary file space. If the user wishes to save the output graphic, the "Save As" button will allow the user to select a location to save the graphic to and provide a name for the output file.

In the event that there is an error opening or reading the selected file, an error message is displayed in the input file preview pane:



Building and Running

Our implementation of DITAA, including the updates mentioned in this document as well as the added features, can be found in the following location on Github: https://github.com/CJMenart/CS-7140-2020/tree/master/DITAA

The existing DITAA project can be built using Apache Ant, a build configuration tool. No additional configuration is needed to build the added features.

See Appendix A: File Listing for a list of source files included in the project.

On your local machine, navigate to a desired directory. Clone the git repository to your local machine or unzip the source tarball:

```
>> cd /home/username/repositories
>> git clone https://github.com/CJMenart/CS-7140-2020.git
```

Or

```
>> tar -zxvf ditaa_source.tar
```

Navigate to the build directory:

```
>> cd CS-7140-2020/DITAA/build
```

Run Ant using the provided release.xml file:

```
>> ant -f release.xml
```

This compiles the Java code and produces a new subdirectory in the DITAA directory called ‘releases’. This will contain the compiled runnable JAR file.

```
>> cd ../releases
>> java -jar ditaa0_9.jar
```

This will open the DITAA GUI. Additional command line arguments may be provided as long as the --gui command is also provided. For example, the following runs the GUI and passes the scale argument to DITAA for all files run in the GUI:

```
>> java -jar ditaa0_9.jar --gui --scale 2.5
```

Smoke Testing

A quick ‘smoke test’ can be performed on the software after it has been built. This section documents the recommended smoke test procedure. This is not intended to be a complete test of the system, but is a quick check that it is operating.

Setup

Before smoke testing, create a small sample text file called helloDitaa.txt with the contents below:

```
/---------------+
|               |
|  Hello, DITAA *---->
|               |
+---------------/
```

The expected output for this file is the following diagram:

Test Case 1: Load Valid File

1. Run DITAA as described in the ‘Building and Running’ Section
   ```
   >> java -jar ditaa0_9.jar
   ```
   a. Confirm that the DITAA GUI is displayed
2. In the DITAA GUI, load the helloDitaa.txt file that was created in setup.
   a. Click the “Load file...” button
   b. In the file chooser, navigate to the helloDitaa.txt file and click “open”
   c. Confirm that
      i. Contents of the file are loaded in the left pane
      ii. DITAA diagram is displayed in the right pane
      iii. DITAA diagram matches the expected output
3. Save the output diagram
   a. Click the “Save As…” button
   b. In the filie chooser, navigate to a directory of choice
   c. Type a file name into the box, e.g. ditaaDiagram.png
   d. Confirm that
      i. The file containing the diagram has been created on the filesystem in the chosen location

Appendix A: File Listing

```
.:
total 56K
drwxr-xr-x. 2 bradschneider 4.0K Nov 29 20:57 build
drwxr-xr-x. 3 bradschneider 4.0K Nov 29 10:45 classes
-rw-r--r--. 1 bradschneider  18K Nov 29 10:45 COPYING
-rw-r--r--. 1 bradschneider 6.6K Nov 29 10:45 ditaa.iml
-rw-r--r--. 1 bradschneider  448 Nov 29 10:45 HISTORY
drwxr-xr-x. 2 bradschneider 4.0K Nov 29 10:45 lib
drwxr-xr-x. 2 bradschneider 4.0K Nov 29 20:43 releases
drwxr-xr-x. 5 bradschneider 4.0K Nov 29 10:45 src
drwxr-xr-x. 3 bradschneider 4.0K Nov 29 10:45 tests

./build:
total 12K
-rw-r--r--. 1 bradschneider   54 Nov 29 10:45 ejp-filter
-rw-r--r--. 1 bradschneider 4.1K Nov 29 17:01 release.xml

./classes:
total 4.0K
drwxr-xr-x. 3 bradschneider 4.0K Nov 29 10:45 artifacts

./classes/artifacts:
total 4.0K
drwxr-xr-x. 2 bradschneider 4.0K Nov 29 15:54 ditaa_jar

./classes/artifacts/ditaa_jar:
total 22M
-rw-r--r--. 1 bradschneider 22M Nov 29 15:54 ditaa.jar

./lib:
total 4.5M
-rw-r--r--. 1 bradschneider 361K Nov 29 10:45 batik-awt-util.jar
-rw-r--r--. 1 bradschneider 272K Nov 29 10:45 batik-bridge.jar
-rw-r--r--. 1 bradschneider 233K Nov 29 10:45 batik-css.jar
-rw-r--r--. 1 bradschneider  86K Nov 29 10:45 batik-dom.jar
-rw-r--r--. 1 bradschneider  46K Nov 29 10:45 batik-extension.jar
-rw-r--r--. 1 bradschneider  75K Nov 29 10:45 batik-ext.jar
-rw-r--r--. 1 bradschneider 135K Nov 29 10:45 batik-gui-util.jar
-rw-r--r--. 1 bradschneider 167K Nov 29 10:45 batik-gvt.jar
-rw-r--r--. 1 bradschneider  37K Nov 29 10:45 batik-parser.jar
-rw-r--r--. 1 bradschneider  37K Nov 29 10:45 batik-script.jar
-rw-r--r--. 1 bradschneider 417K Nov 29 10:45 batik-svg-dom.jar
```

```
-rw-r--r--. 1 bradschneider 161K Nov 29 10:45 batik-svggen.jar
-rw-r--r--. 1 bradschneider 136K Nov 29 10:45 batik-swing.jar
-rw-r--r--. 1 bradschneider  62K Nov 29 10:45 batik-transcoder.jar
-rw-r--r--. 1 bradschneider  80K Nov 29 10:45 batik-util.jar
-rw-r--r--. 1 bradschneider  20K Nov 29 10:45 batik-xml.jar
-rw-r--r--. 1 bradschneider  30K Nov 29 10:45 commons-cli-1.0.jar
-rw-r--r--. 1 bradschneider  41K Nov 29 10:45 commons-cli-1.2.jar
-rw-r--r--. 1 bradschneider  57K Nov 29 10:45 jericho-html-1.4.jar
-rw-r--r--. 1 bradschneider 584K Nov 29 10:45 js.jar
-rw-r--r--. 1 bradschneider 473K Nov 29 10:45 pdf-transcoder.jar
-rw-r--r--. 1 bradschneider 913K Nov 29 10:45 xerces_2_5_0.jar
-rw-r--r--. 1 bradschneider 105K Nov 29 10:45 xml-apis.jar

./releases:
total 0

./src:
total 12K
drwxr-xr-x. 3 bradschneider 4.0K Nov 29 10:45 au
drwxr-xr-x. 2 bradschneider 4.0K Nov 29 10:45 META-INF
drwxr-xr-x. 3 bradschneider 4.0K Nov 29 10:45 org

./src/au:
total 4.0K
drwxr-xr-x. 3 bradschneider 4.0K Nov 29 10:45 id

./src/au/id:
total 4.0K
drwxr-xr-x. 3 bradschneider 4.0K Nov 29 10:45 jericho

./src/au/id/jericho:
total 4.0K
drwxr-xr-x. 3 bradschneider 4.0K Nov 29 10:45 lib

./src/au/id/jericho/lib:
total 4.0K
drwxr-xr-x. 2 bradschneider 4.0K Nov 29 10:45 html

./src/au/id/jericho/lib/html:
total 336K
-rw-r--r--. 1 bradschneider 5.7K Nov 29 10:45 Attribute.java
-rw-r--r--. 1 bradschneider  16K Nov 29 10:45 Attributes.java
-rw-r--r--. 1 bradschneider 2.4K Nov 29 10:45 BlankOutputSegment.java
-rw-r--r--. 1 bradschneider  64K Nov 29 10:45 CharacterEntityReference.java
```

```
-rw-r--r--. 1 bradschneider  17K Nov 29 10:45 CharacterReference.java
-rw-r--r--. 1 bradschneider 2.8K Nov 29 10:45 CharOutputSegment.java
-rw-r--r--. 1 bradschneider 7.0K Nov 29 10:45 Element.java
-rw-r--r--. 1 bradschneider 6.4K Nov 29 10:45 EndTag.java
-rw-r--r--. 1 bradschneider  16K Nov 29 10:45 FormControlType.java
-rw-r--r--. 1 bradschneider  11K Nov 29 10:45 FormField.java
-rw-r--r--. 1 bradschneider  11K Nov 29 10:45 FormFields.java
-rw-r--r--. 1 bradschneider 6.0K Nov 29 10:45 IntStringHashMap.java
-rw-r--r--. 1 bradschneider 2.3K Nov 29 10:45 IOutputSegment.java
-rw-r--r--. 1 bradschneider  11K Nov 29 10:45 NumericCharacterReference.java
-rw-r--r--. 1 bradschneider 5.4K Nov 29 10:45 OutputDocument.java
-rw-r--r--. 1 bradschneider 1.6K Nov 29 10:45 OutputSegmentComparator.java
-rw-r--r--. 1 bradschneider 2.2K Nov 29 10:45 package.html
-rw-r--r--. 1 bradschneider 3.0K Nov 29 10:45 SearchCache.java
-rw-r--r--. 1 bradschneider  14K Nov 29 10:45 Segment.java
-rw-r--r--. 1 bradschneider  21K Nov 29 10:45 Source.java
-rw-r--r--. 1 bradschneider 4.9K Nov 29 10:45 SpecialTag.java
-rw-r--r--. 1 bradschneider  36K Nov 29 10:45 StartTag.java
-rw-r--r--. 1 bradschneider 2.7K Nov 29 10:45 StringOutputSegment.java
-rw-r--r--. 1 bradschneider   20 Nov 29 10:45 stylesheet.css
-rw-r--r--. 1 bradschneider  26K Nov 29 10:45 Tag.java

./src/META-INF:
total 4.0K
-rw-r--r--. 1 bradschneider 92 Nov 29 10:45 MANIFEST.MF

./src/org:
total 4.0K
drwxr-xr-x. 3 bradschneider 4.0K Nov 29 10:45 stathissideris

./src/org/stathissideris:
total 4.0K
drwxr-xr-x. 6 bradschneider 4.0K Nov 29 10:45 ascii2image

./src/org/stathissideris/ascii2image:
total 16K
drwxr-xr-x. 2 bradschneider 4.0K Nov 29 17:03 core
drwxr-xr-x. 2 bradschneider 4.0K Nov 29 10:45 graphics
drwxr-xr-x. 2 bradschneider 4.0K Nov 29 10:45 test
drwxr-xr-x. 2 bradschneider 4.0K Nov 29 10:45 text

./src/org/stathissideris/ascii2image/core:
total 108K
-rw-r--r--. 1 bradschneider 9.2K Nov 29 15:54 CommandLineConverter.java
```

-rw-r--r--. 1 bradschneider 6.2K Nov 29 10:45 ConfigurationParser.java
-rw-r--r--. 1 bradschneider 5.8K Nov 29 10:45 ConversionOptions.java
-rw-r--r--. 1 bradschneider 182 Nov 29 10:45 DebugUtils.java
-rw-r--r--. 1 bradschneider 9.1K Nov 29 17:03 DitaaGUI.java
-rw-r--r--. 1 bradschneider 2.0K Nov 29 10:45 DocBookConverter.java
-rw-r--r--. 1 bradschneider 4.4K Nov 29 10:45 FileUtils.java
-rw-r--r--. 1 bradschneider 6.4K Nov 29 10:45 HTMLConverter.java
-rw-r--r--. 1 bradschneider 9.7K Nov 29 10:45 JavadocTaglet.java
-rw-r--r--. 1 bradschneider 193 Nov 29 10:45 Pair.java
-rw-r--r--. 1 bradschneider 1.9K Nov 29 10:45 PerformanceTester.java
-rw-r--r--. 1 bradschneider 4.7K Nov 29 10:45 ProcessingOptions.java
-rw-r--r--. 1 bradschneider 2.2K Nov 29 10:45 RenderingOptions.java
-rw-r--r--. 1 bradschneider 1.7K Nov 29 10:45 Shape3DOrderingComparator.java
-rw-r--r--. 1 bradschneider 5.4K Nov 29 10:45 VisualTester.java

./src/org/stathissideris/ascii2image/graphics:
total 144K
-rw-r--r--. 1 bradschneider 15K Nov 29 10:45 BitmapRenderer.java
-rw-r--r--. 1 bradschneider 9.1K Nov 29 10:45 CompositeDiagramShape.java
-rw-r--r--. 1 bradschneider 2.1K Nov 29 10:45 CustomShapeDefinition.java
-rw-r--r--. 1 bradschneider 4.1K Nov 29 10:45 DiagramComponent.java
-rw-r--r--. 1 bradschneider 30K Nov 29 10:45 Diagram.java
-rw-r--r--. 1 bradschneider 29K Nov 29 10:45 DiagramShape.java
-rw-r--r--. 1 bradschneider 4.0K Nov 29 10:45 DiagramText.java
-rw-r--r--. 1 bradschneider 6.1K Nov 29 10:45 FontMeasurer.java
-rw-r--r--. 1 bradschneider 4.6K Nov 29 10:45 ImageHandler.java
-rw-r--r--. 1 bradschneider 5.2K Nov 29 10:45 OffScreenSVGRenderer.java
-rw-r--r--. 1 bradschneider 6.2K Nov 29 10:45 ShapeEdge.java
-rw-r--r--. 1 bradschneider 3.1K Nov 29 10:45 ShapePoint.java

./src/org/stathissideris/ascii2image/test:
total 20K
-rw-r--r--. 1 bradschneider 688 Nov 29 10:45 CellSetTest.java
-rw-r--r--. 1 bradschneider 707 Nov 29 10:45 GridPatternTest.java
-rw-r--r--. 1 bradschneider 11K Nov 29 10:45 TextGridTest.java

./src/org/stathissideris/ascii2image/text:
total 112K
-rw-r--r--. 1 bradschneider 3.0K Nov 29 10:45 AbstractCell.java
-rw-r--r--. 1 bradschneider 5.3K Nov 29 10:45 AbstractionGrid.java
-rw-r--r--. 1 bradschneider 18K Nov 29 10:45 CellSet.java
-rw-r--r--. 1 bradschneider 12K Nov 29 10:45 GridPatternGroup.java
-rw-r--r--. 1 bradschneider 8.9K Nov 29 10:45 GridPattern.java
-rw-r--r--. 1 bradschneider 4.8K Nov 29 10:45 StringUtils.java

```
-rw-r--r--. 1 bradschneider  47K Nov 29 10:45 TextGrid.java

./tests:
total 4.0K
drwxr-xr-x. 2 bradschneider 4.0K Nov 29 10:45 text

./tests/text:
total 432K
-rw-r--r--. 1 bradschneider  16K Nov 29 10:45 art10.png
-rw-r--r--. 1 bradschneider  554 Nov 29 10:45 art10.txt
-rw-r--r--. 1 bradschneider 1.4K Nov 29 10:45 art11.png
-rw-r--r--. 1 bradschneider  148 Nov 29 10:45 art11.txt
-rw-r--r--. 1 bradschneider  132 Nov 29 10:45 art12.txt
-rw-r--r--. 1 bradschneider  492 Nov 29 10:45 art13.png
-rw-r--r--. 1 bradschneider   44 Nov 29 10:45 art13.txt
-rw-r--r--. 1 bradschneider 1.9K Nov 29 10:45 art14.png
-rw-r--r--. 1 bradschneider   73 Nov 29 10:45 art14.txt
-rw-r--r--. 1 bradschneider   42 Nov 29 10:45 art15.txt
-rw-r--r--. 1 bradschneider  176 Nov 29 10:45 art16.txt
-rw-r--r--. 1 bradschneider  280 Nov 29 10:45 art17.txt
-rw-r--r--. 1 bradschneider 6.0K Nov 29 10:45 art18.png
-rw-r--r--. 1 bradschneider  410 Nov 29 10:45 art18.txt
-rw-r--r--. 1 bradschneider    7 Nov 29 10:45 art19.txt
-rw-r--r--. 1 bradschneider  12K Nov 29 10:45 art1.png
-rw-r--r--. 1 bradschneider 1.7K Nov 29 10:45 art1.txt
-rw-r--r--. 1 bradschneider  795 Nov 29 10:45 art20.txt
-rw-r--r--. 1 bradschneider   12 Nov 29 10:45 art21.txt
-rw-r--r--. 1 bradschneider 2.8K Nov 29 10:45 art22_2.png
-rw-r--r--. 1 bradschneider 2.6K Nov 29 10:45 art22_3.png
-rw-r--r--. 1 bradschneider 2.9K Nov 29 10:45 art22_4.png
-rw-r--r--. 1 bradschneider 2.4K Nov 29 10:45 art22.png
-rw-r--r--. 1 bradschneider  108 Nov 29 10:45 art22.txt
-rw-r--r--. 1 bradschneider  978 Nov 29 10:45 art23.png
-rw-r--r--. 1 bradschneider  130 Nov 29 10:45 art23.txt
-rw-r--r--. 1 bradschneider 1.2K Nov 29 10:45 art24.png
-rw-r--r--. 1 bradschneider  226 Nov 29 10:45 art24.txt
-rw-r--r--. 1 bradschneider 1.2K Nov 29 10:45 art25.png
-rw-r--r--. 1 bradschneider  138 Nov 29 10:45 art2_5.txt
-rw-r--r--. 1 bradschneider  184 Nov 29 10:45 art25.txt
-rw-r--r--. 1 bradschneider 1.3K Nov 29 10:45 art26.png
-rw-r--r--. 1 bradschneider  184 Nov 29 10:45 art26.txt
-rw-r--r--. 1 bradschneider  828 Nov 29 10:45 art27.png
-rw-r--r--. 1 bradschneider  116 Nov 29 10:45 art27.txt
-rw-r--r--. 1 bradschneider  811 Nov 29 10:45 art28.png
```

```
-rw-r--r--. 1 bradschneider  116 Nov 29 10:45 art28.txt
-rw-r--r--. 1 bradschneider  646 Nov 29 10:45 art29.png
-rw-r--r--. 1 bradschneider   75 Nov 29 10:45 art29.txt
-rw-r--r--. 1 bradschneider  11K Nov 29 10:45 art2.png
-rw-r--r--. 1 bradschneider  708 Nov 29 10:45 art2.txt
-rw-r--r--. 1 bradschneider  530 Nov 29 10:45 art30.png
-rw-r--r--. 1 bradschneider   75 Nov 29 10:45 art30.txt
-rw-r--r--. 1 bradschneider  748 Nov 29 10:45 art31.png
-rw-r--r--. 1 bradschneider  113 Nov 29 10:45 art31.txt
-rw-r--r--. 1 bradschneider 4.4K Nov 29 10:45 art32.png
-rw-r--r--. 1 bradschneider  226 Nov 29 10:45 art32.txt
-rw-r--r--. 1 bradschneider 1.1K Nov 29 10:45 art3_5.png
-rw-r--r--. 1 bradschneider  133 Nov 29 10:45 art3_5.txt
-rw-r--r--. 1 bradschneider 1.1K Nov 29 10:45 art3.png
-rw-r--r--. 1 bradschneider  130 Nov 29 10:45 art3.txt
-rw-r--r--. 1 bradschneider  357 Nov 29 10:45 art4.png
-rw-r--r--. 1 bradschneider   11 Nov 29 10:45 art4.txt
-rw-r--r--. 1 bradschneider 3.1K Nov 29 10:45 art5.png
-rw-r--r--. 1 bradschneider  154 Nov 29 10:45 art5.txt
-rw-r--r--. 1 bradschneider  104 Nov 29 10:45 art6.txt
-rw-r--r--. 1 bradschneider  101 Nov 29 10:45 art7.txt
-rw-r--r--. 1 bradschneider 1.2K Nov 29 10:45 art8.png
-rw-r--r--. 1 bradschneider   86 Nov 29 10:45 art8.txt
-rw-r--r--. 1 bradschneider  666 Nov 29 10:45 art_text.txt
-rw-r--r--. 1 bradschneider   38 Nov 29 10:45 bug10.txt
-rw-r--r--. 1 bradschneider    7 Nov 29 10:45 bug11.txt
-rw-r--r--. 1 bradschneider   52 Nov 29 10:45 bug12.txt
-rw-r--r--. 1 bradschneider    5 Nov 29 10:45 bug13.txt
-rw-r--r--. 1 bradschneider  138 Nov 29 10:45 bug14.txt
-rw-r--r--. 1 bradschneider   54 Nov 29 10:45 bug15.txt
-rw-r--r--. 1 bradschneider   38 Nov 29 10:45 bug1.txt
-rw-r--r--. 1 bradschneider  134 Nov 29 10:45 bug2.txt
-rw-r--r--. 1 bradschneider  214 Nov 29 10:45 bug3.txt
-rw-r--r--. 1 bradschneider 1.8K Nov 29 10:45 bug4.png
-rw-r--r--. 1 bradschneider   64 Nov 29 10:45 bug4.txt
-rw-r--r--. 1 bradschneider  318 Nov 29 10:45 bug5.txt
-rw-r--r--. 1 bradschneider   46 Nov 29 10:45 bug6.txt
-rw-r--r--. 1 bradschneider 3.0K Nov 29 10:45 bug7.png
-rw-r--r--. 1 bradschneider  204 Nov 29 10:45 bug7.txt
-rw-r--r--. 1 bradschneider  228 Nov 29 10:45 bug8.txt
-rw-r--r--. 1 bradschneider 1.8K Nov 29 10:45 bug9_5.png
-rw-r--r--. 1 bradschneider  108 Nov 29 10:45 bug9_5.txt
-rw-r--r--. 1 bradschneider 2.0K Nov 29 10:45 bug9.png
-rw-r--r--. 1 bradschneider  168 Nov 29 10:45 bug9.txt
```

-rw-r--r--. 1 bradschneider 3.8K Nov 29 10:45 color_codes.png
-rw-r--r--. 1 bradschneider 222 Nov 29 10:45 color_codes.txt
-rw-r--r--. 1 bradschneider 866 Nov 29 10:45 corner_case01.png
-rw-r--r--. 1 bradschneider 128 Nov 29 10:45 corner_case01.txt
-rw-r--r--. 1 bradschneider 914 Nov 29 10:45 corner_case02.png
-rw-r--r--. 1 bradschneider 127 Nov 29 10:45 corner_case02.txt
-rw-r--r--. 1 bradschneider 512 Nov 29 10:45
dak_orgstruktur_vs_be.ditaa.OutOfMemoryError.2.txt
-rw-r--r--. 1 bradschneider 512 Nov 29 10:45
dak_orgstruktur_vs_be.ditaa.OutOfMemoryError.3.txt
-rw-r--r--. 1 bradschneider 512 Nov 29 10:45
dak_orgstruktur_vs_be.ditaa.OutOfMemoryError.4.txt
-rw-r--r--. 1 bradschneider 140 Nov 29 10:45
dak_orgstruktur_vs_be.ditaa.OutOfMemoryError.edit.txt
-rw-r--r--. 1 bradschneider 512 Nov 29 10:45
dak_orgstruktur_vs_be.ditaa.OutOfMemoryError.txt
-rw-r--r--. 1 bradschneider 512 Nov 29 10:45 dak_orgstruktur_vs_be.ditaa.txt
-rw-r--r--. 1 bradschneider 610 Nov 29 10:45 ditaa_bug2.txt
-rw-r--r--. 1 bradschneider 2.3K Nov 29 10:45 ditaa_bug.txt
-rw-r--r--. 1 bradschneider   65 Nov 29 10:45 garbage.txt
-rw-r--r--. 1 bradschneider 359 Nov 29 10:45 logo.txt
-rw-r--r--. 1 bradschneider 183 Nov 29 10:45 simple_S01.txt
-rw-r--r--. 1 bradschneider   40 Nov 29 10:45 simple_square01.txt
-rw-r--r--. 1 bradschneider 133 Nov 29 10:45 simple_U01.txt

Team Member Journals

Christopher Menart

11/29/20

- Looking into refactors
    - Start planning to decompose TextGrid class, but closer inspection casts doubt on the wisdom of this
    - Start breaking up diagram constructor instead
    - This triggers an (existing?) bug, which we spend the rest of the day fixing

11/30/20

- Formatting notes on DITAA classes for inclusion in Implementation document

Brad Schneider

11/29/20

- Initial implementation of DITAA GUI code
- Documented structure of implementation in implementation doc
- Identify smoke test steps

11/30/20

- Identified class notes in design doc that were good candidates for moving to implementation.
- Configure/document ant build steps

12/1/20

- Review existing implementation draft against requirements for the deliverable
- Proofreading, editing of implementation doc draft
- Creation of File listing index