CSE360 Team Project

CSE360

2019 Fall

Jonathan Cahal
Tingyu Luo
Anna McDonald
Ruijun Yang

Table of Contents

Introduction – description of this document

The following user guide describes the requirements, defaults, and format of a program that loads and saves files, as well as presents any errors that are created in this process. The program here is going to read a .txt file with one or more format flags before some text. This program will only read ASCII text and then output a file with the text in the file in accordance with any format flags. In this program you are presented with a window that has three actions: Choose File, Preview and Save As. There is also a Preview Window and a Error Log that provides users with the ability to preview their inputted .txt file as well as any errors that come up in the program.

Overview of the program

To use the program you click Choose File to begin. After a chosen txt file is inputted into the program, you can click on the Preview button to see what the formatted file looks like in the File Preview area. If you are happy with the text you got from the File Preview area, the Save As button will allow you save the formatted file. The user also has the option to go back into your .txt file and add/remove flags, save the file and then press the "Preview" button again.

All the text will be formatted based on the format flags before the text. If there is a text without any format flags before it, it will run based on the default. The default formats are 80 characters per line, single space, left justify, and it will merge lines together if the first line doesn't have 80 characters. The following are the format flags can be used inside the txt file:

-r	right justify
-c	center
-l	left justify
-t	title, centered, no justification
-n	remove indentation
-d	double space
-s	single space
-i	indent on first line, 5 spaces, left only
-b	indent for multiple lines, 10 spaces
-2	2 columns, 35 characters + 10 spaces + 35 characters
-1	1 column
-e	blank line

If there are improper flags in the input file there will be an error message in the error log. An error message will also be presented if the preview button is pressed before a file is chosen.

Format Overwrites

Formats are overwritten as flags are called. In formatting conflicts the bottom flag wins

External requirements to execute the program

To run the program you need a .txt file containing text and one or more format flags. The flags must be placed before or after blocks of text, on their own line. The text immediately following the flag(s) will be formatted according to those flags. Here are some examples of valid and invalid input files.

Valid input file

```
-c
Lorem ipsum dolor sit amet
-e
-i
-l
-d
-b
Consectetur adipiscing elit,sed do eiusmod tempor.
incididunt ut labore et dolore magna aliqua.
Ut enim ad minim veniam, quis nostrud.
exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure
dolor in reprehenderit
in voluptate velit esse cillum dolore eu fugiat nulla
pariatur.
-e
-b
Excepteur sint occaecat cupidatat non proident,
sunt in culpa qui officia deserunt mollit anim id est laborum.
```

Preview Output

```
                Lorem ipsum dolor sit amet


    Consectetur adipiscing elit,sed do eiusmod tempor. incididunt ut labore et

dolore magna aliqua.Ut enim ad minim veniam, quis nostrud.exercitation ullamco

laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor

inreprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.


        Excepteur sint occaecat cupidatat non proident, sunt in culpa qui

            officia deserunt mollit anim id est laborum.
```
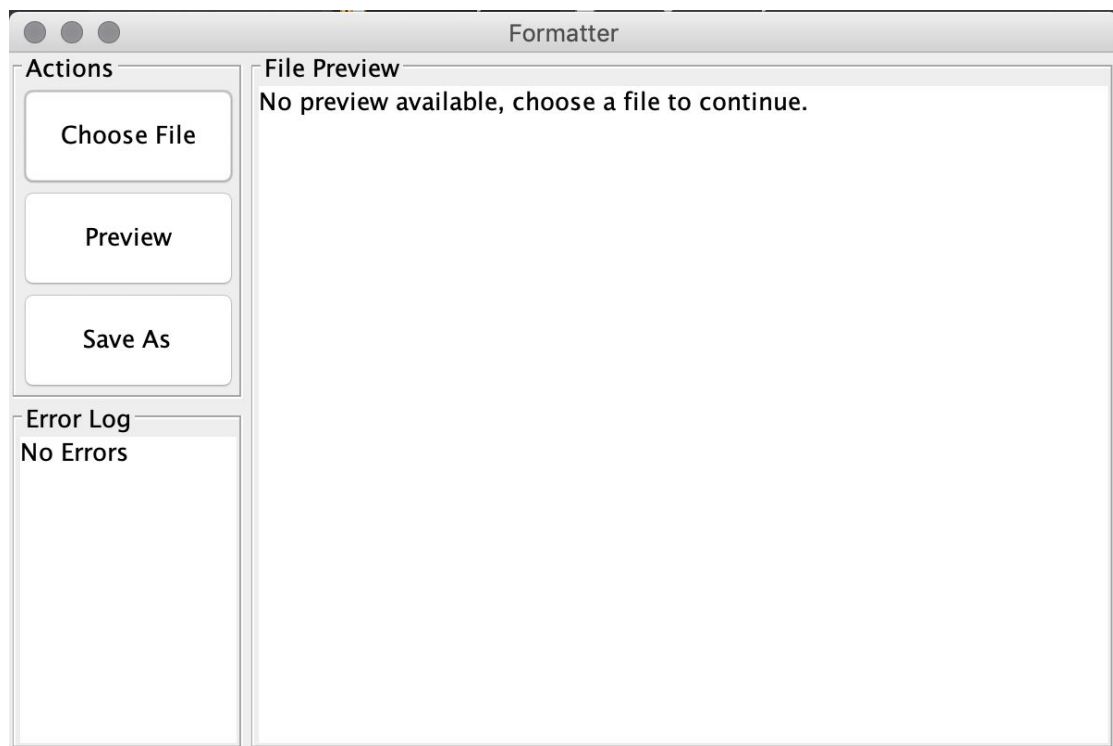
Invalid Input file

```
-i
-!
Lorem ipsum dolor sit amet, consectetur. Voluptate velit esse cillum dolore.
```

Error Log Output

```
Flag error
```

Getting started

Upon opening the program you will be greeted with a window like the figure below. The main area of user interaction is the "Actions" panel on the top left. The other two areas provide the end-user with dynamic feedback regarding the input file and actions that use. If there's a problem with the input or an action used, feedback will be displayed in the "Error Log". A formatted preview of the input file is displayed in the "File Preview" panel once the "Preview" action has been used.

Example use showing all functionality

Before using the program you have to start with a text file including format flags. Here's an example.
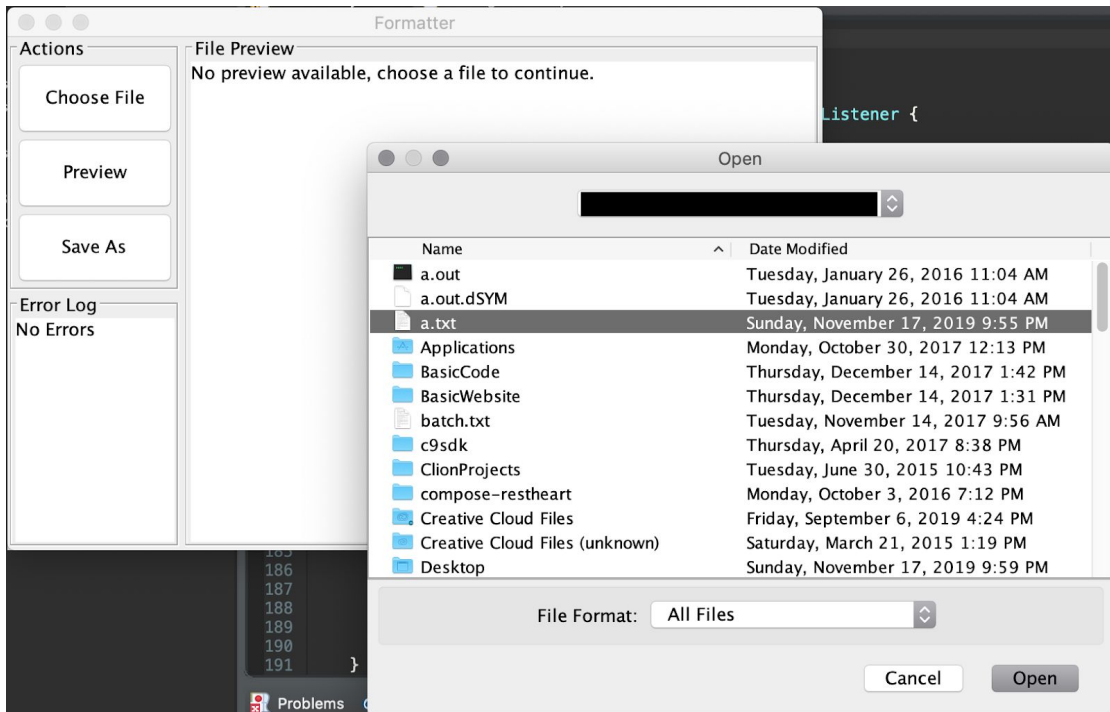
a.txt

```
-i
-d
Lorem ipsum dolor sit amet, consectetur.
adipiscing elit, sed do eiusmod tempor.
incididunt ut labore et dolore magna aliqua.
Ut enim ad minim veniam, quis nostrud exercitation.
```
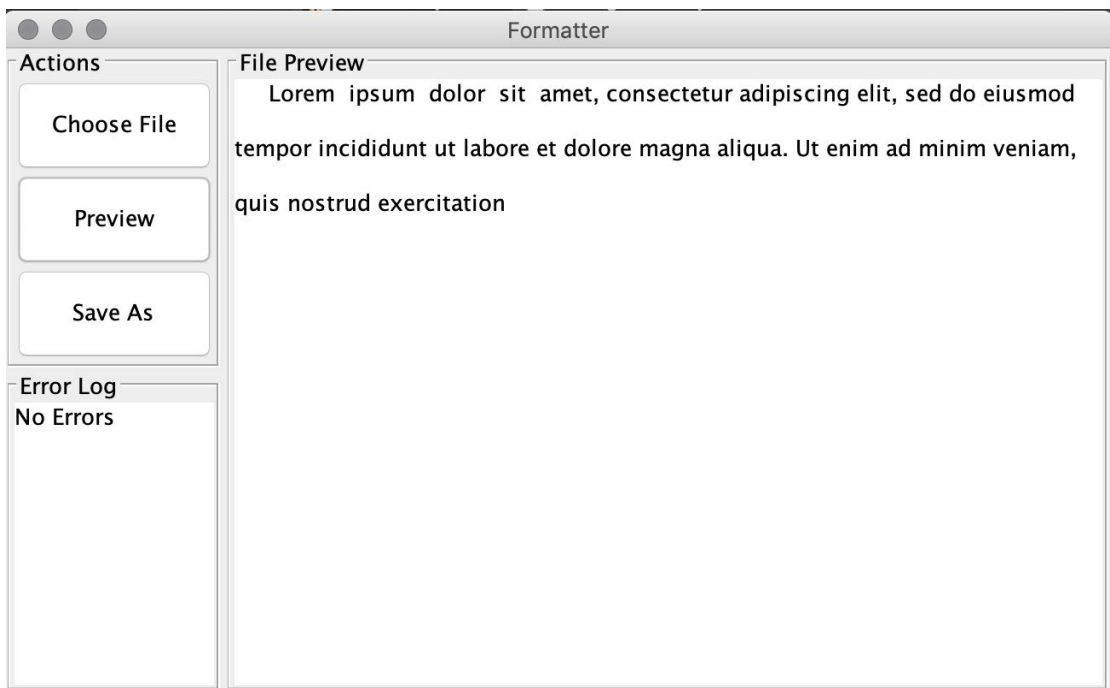
After processing, based on the format flags provided, this text file should have the following attributes.

- According to Defaults
  - 80 chars per/line
  - Left justified
- Set by flags
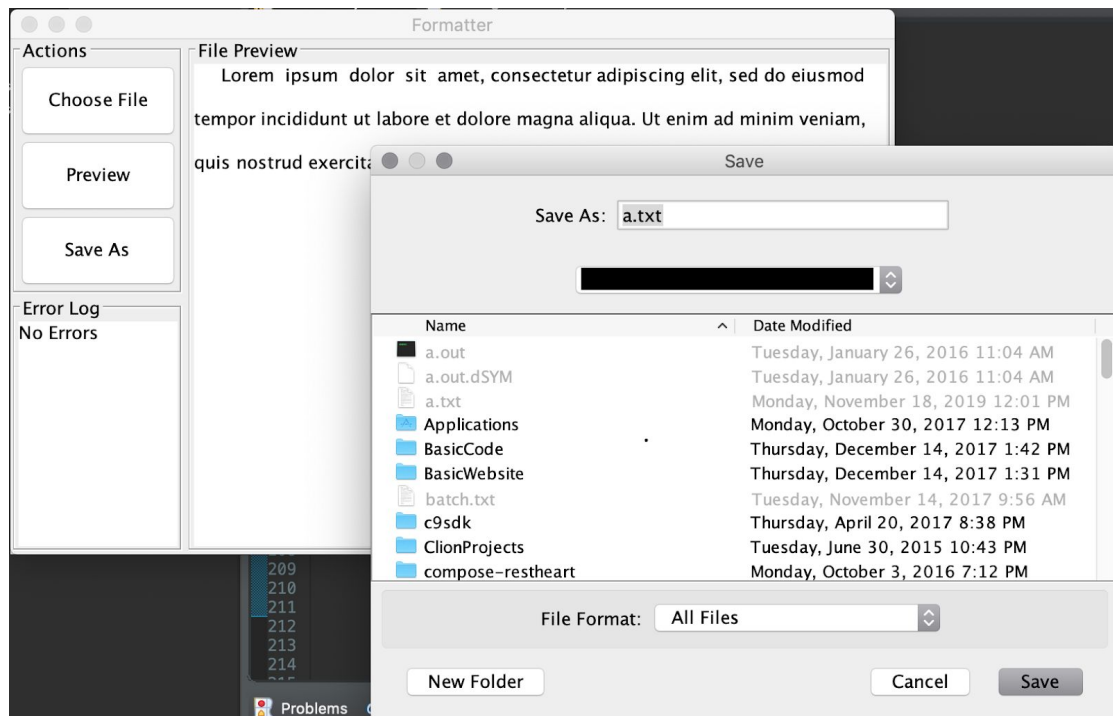 - First-line indented
  - Double spaced

Once opening the program, you will need to choose a file to format, do this by pressing the "Choose File" button. You should see a window open similar to the figure below.

Once a file has been selected it's now time to preview what the formatted file will look like. To do this, press the "Preview" button. The output will look like this figure below.
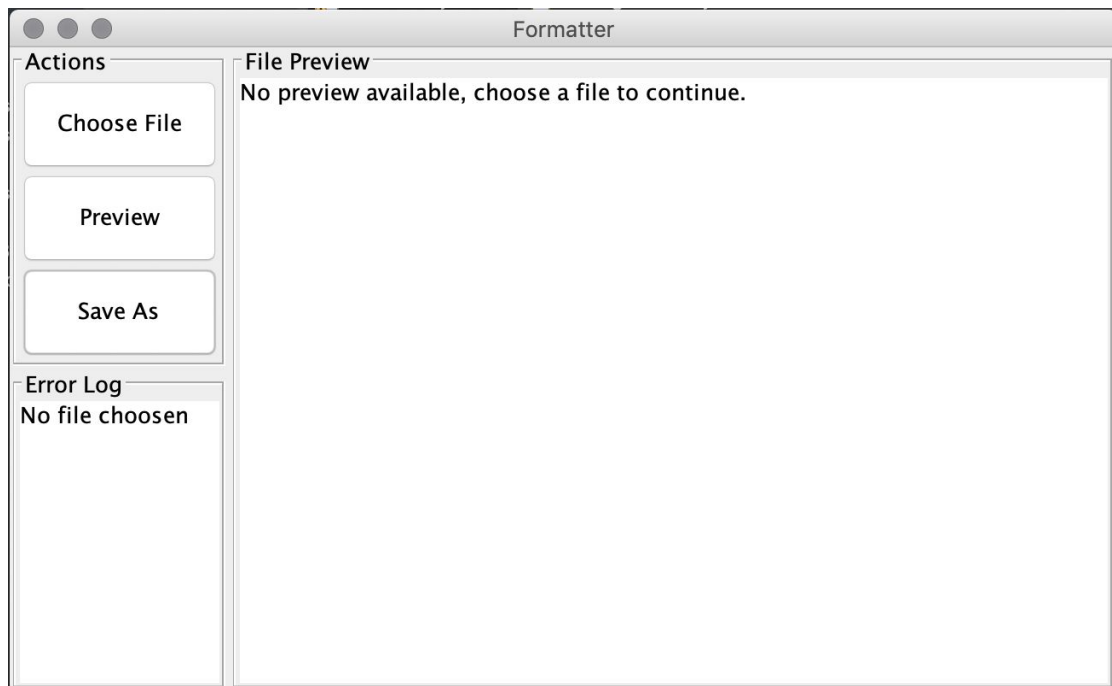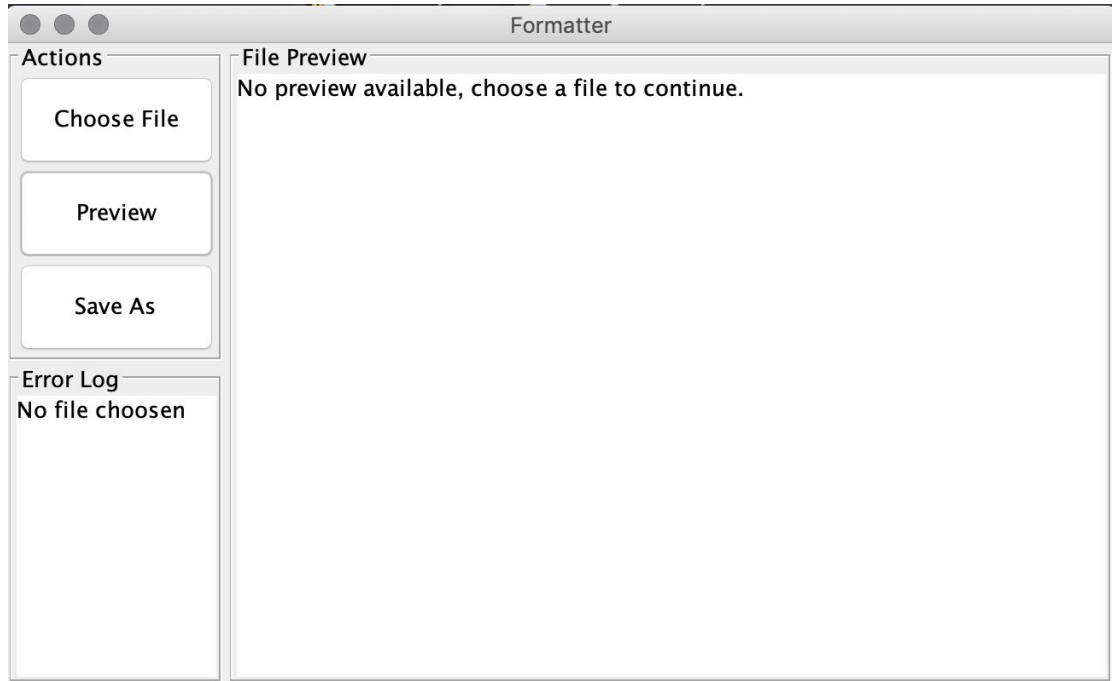
At this point, if you are unhappy with what the preview shows, you may go back into your .txt file and add/remove flags, save the file and then press the "Preview" button again. Once a file is chosen, there is no need to re-choose it even if changes are made using a different program. If you're happy with the format of your file as shown in the preview area, you can now press the "Save As" button to save your file in the format shown. This action will look similar to the figure below.

Examples of error conditions and error messages

Pressing the "Preview" or "Save As" button(s) prior to selecting a file will result in an error being displayed in the error log. See the figures below.

Ending the program

To end the program select "Quit Formatter" from the application menu, shown

in the figure below, or simply press the button on the top of the window.