8.8 *Lab:* Lipograms

According to *Wikipedia*,

> a *lipogram* (from Greek lipagrammatos, "missing letter") is a kind of writing with constraints or word game consisting of writing paragraphs or longer works in which a particular letter or group of letters is missing, usually a common vowel, the most common in English being *e*.
>
> "*Gadsby* is a notorious book by Californian author E. V. Wright, circa 1939. It was Wright's fourth book. It is famous for consisting only of words not containing any e's. Gadsby is thus a lipogram, or a display of constraint in writing. It is 50,100 words long. Wright informs us in Gadsby's introduction of having had to impair his own typing contraption to avoid slipups."

The *Lipogrammer* program, shown in Figure 8-5, helps to create and verify lipograms. It shows the original text, below it the same text with all letters *e* replaced with #, and to the right, the list of all 'offending' words (with an *e* in them). The user can load a lipogram text from a file or type it in or cut and paste it from another program. There is also a menu command to save the text. In this lab, you will write the `LipogramAnalyzer` class for this program.

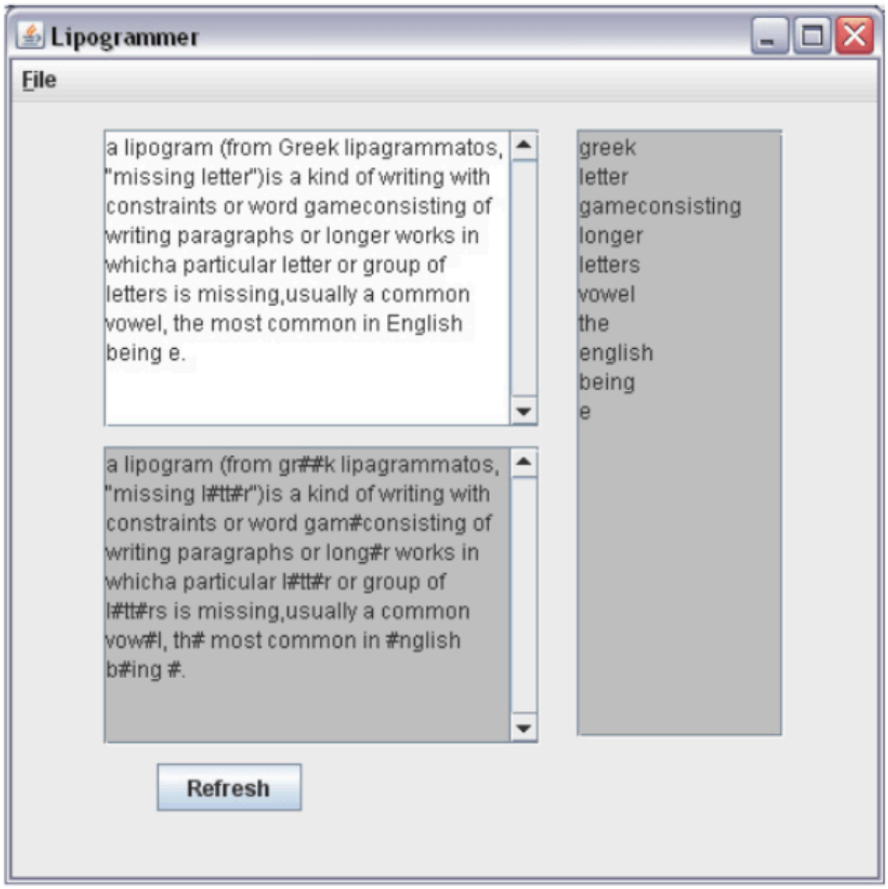

Figure 8-5. The *Lipogrammer* program

The `LipogramAnalyzer` should have the following constructor and two public methods:

Constructor Summary	
`public LipogramAnalyzer(String text)` Saves the `text` string.	
Method Summary	
`String`	`mark(char letter)` Returns the saved `text` string with all characters equal to `letter` replaced with `'#'`.
`String`	`allWordsWith(char letter)` Returns a string that concatenates all "offending" words from `text` that contain `letter`; the words are separated by `'\n'` characters; the returned string does not contain duplicate words: each word occurs only once; there are no punctuation or whitespace characters in the returned string other than `'\n'` characters.

Hint: write a private method to extract and return the word that contains the character at a specified position in `text`. Find the boundaries of the word by scanning the text to the left and to the right of the given position.

Combine in one project your `LipogramAnalyzer` class with the `Lipogrammer` and `LipogrammerMenu` GUI classes from the `JM\Ch08\Lipogrammer` folder. Test the program.