WordList (1.0)

Kcits970

January 10, 2022

Sections

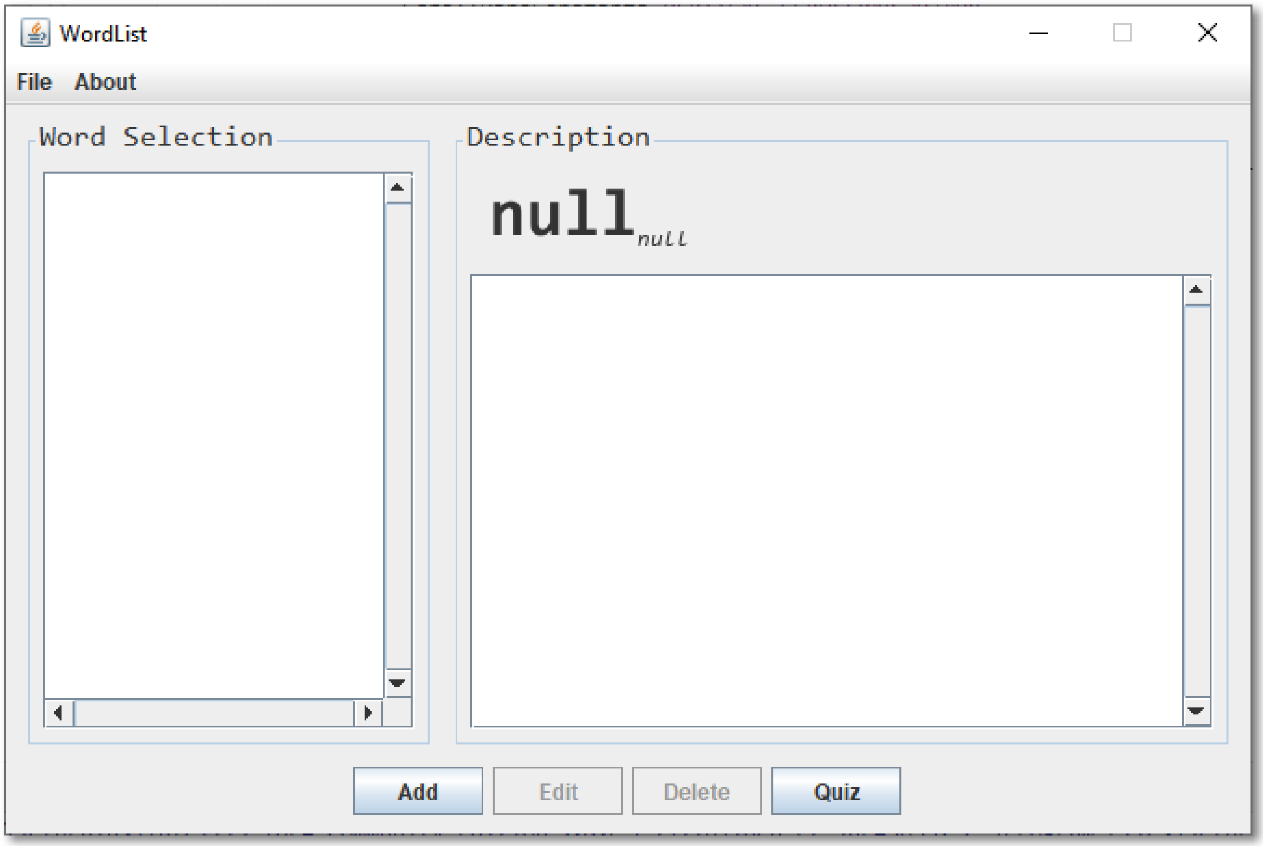
Section 1. Description

‘WordList’ is a vocabulary list interface. Its main purpose is to make vocabulary memorization slightly more entertaining for the average people. To achieve this purpose, ‘WordList’ provides an event-driven interface to store and view word objects, as well as a quiz dialog with a scoring system.

The program is written in Java language, and compiled using Java Compiler 15.0.1. Attempts to compile the given source code with a different compiler may result in unexpected or undefined behavior.

Section 2. Features

When the program is launched, an empty frame will be shown.

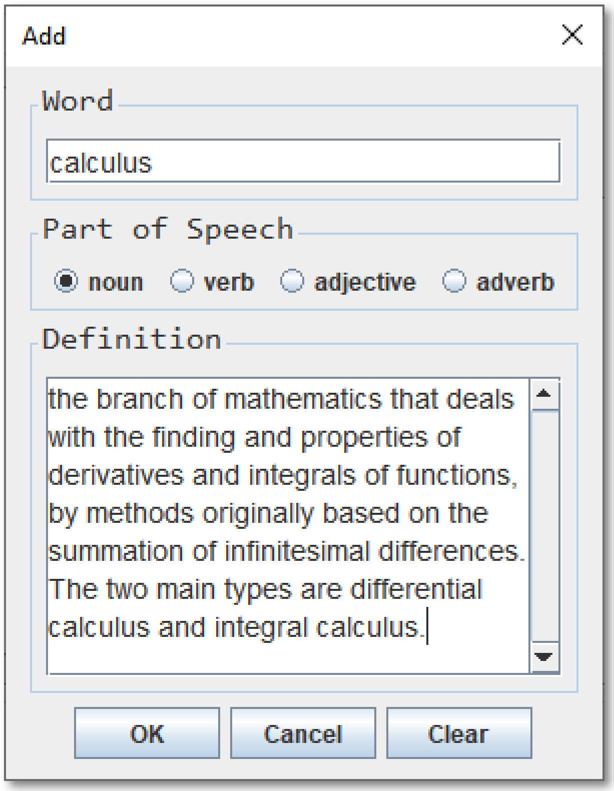


(The initial frame at program launch.)

Adding/Deleting Words

To add a new word to the list, click the [Add] button. A dialog containing a text field, 4 radio buttons, and a text area will appear. Simply enter the word properties[1] into the fields and click the [OK] button. The entered word will be visible[2] in the list.

[1] name of word, part of speech, and definition.
[2] It will be visible in the list *if and only if* the entered word is valid. (all fields must be completed, and duplicate words are not allowed.)


Add

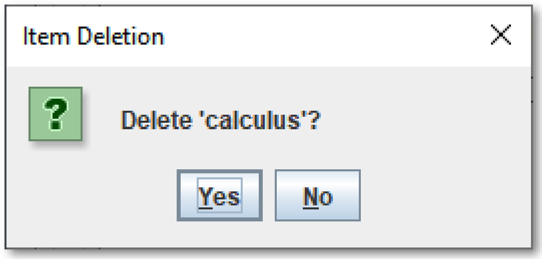
Word: calculus

Part of Speech: (● noun) ○ verb ○ adjective ○ adverb

Definition: the branch of mathematics that deals with the finding and properties of derivatives and integrals of functions, by methods originally based on the summation of infinitesimal differences. The two main types are differential calculus and integral calculus.

(The 'Add' Dialog)

To delete an existing word on the list, select any item in [Word Selection], and click the [Delete] button. A confirmation dialog will pop up asking for the user's confirmation to delete the selected word. Click [Yes] to delete the selected word.

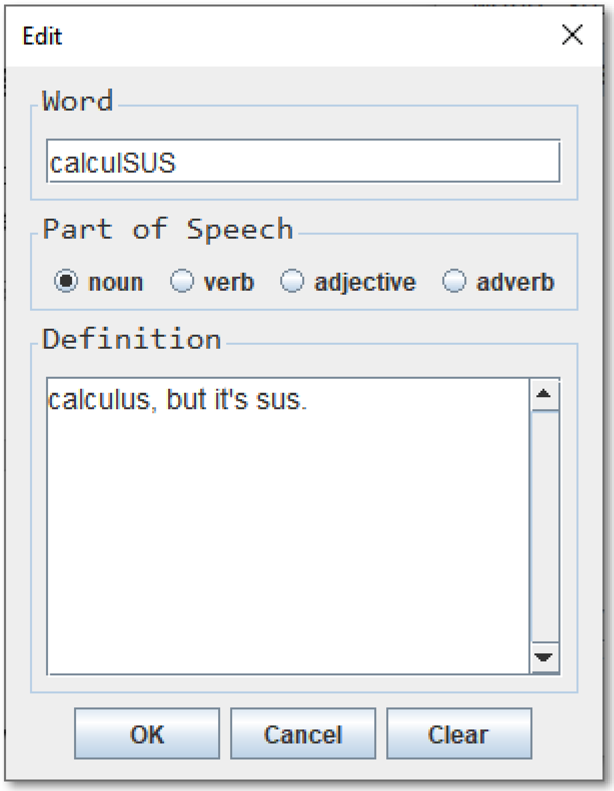

Item Deletion

Delete 'calculus'?

(The Deletion Dialog)

Editing Words

The process of editing is similar to adding. To edit a video, select any item in [Word Selection], and click the [Edit] button. Make any desired changes to the fields, and click the [OK] button. If the edits are valid, the changes will be reflected on the main window.



Word: calculSUS

Part of Speech: [✓ noun] verb adjective adverb

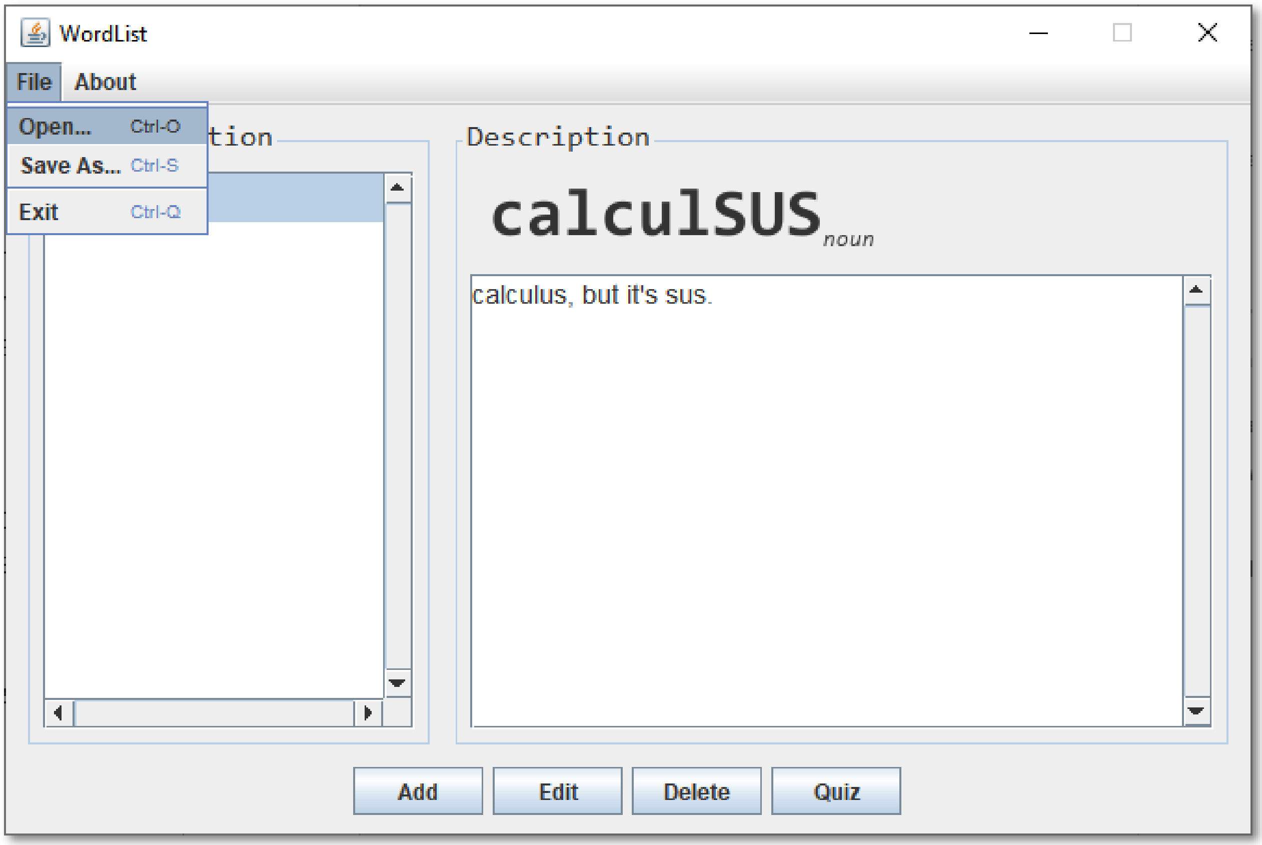
Definition: calculus, but it's sus.

(The 'Edit' Dialog)

Loading/Saving Words

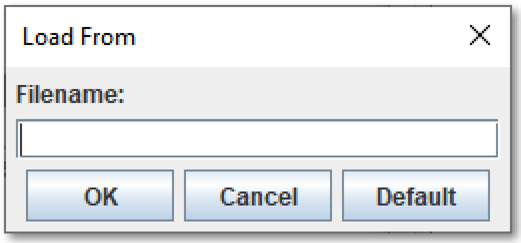
The words can be loaded from or saved to an external file. 'WordList' provides a default file that the user can directly load into the program. To load from this file, navigate to the above menu bar, and go to [File] -> [Open...][3]. The program will respond with a filename dialog.

[3] ctrl+o can also be used as a shortcut.



calculus, but it's sus.

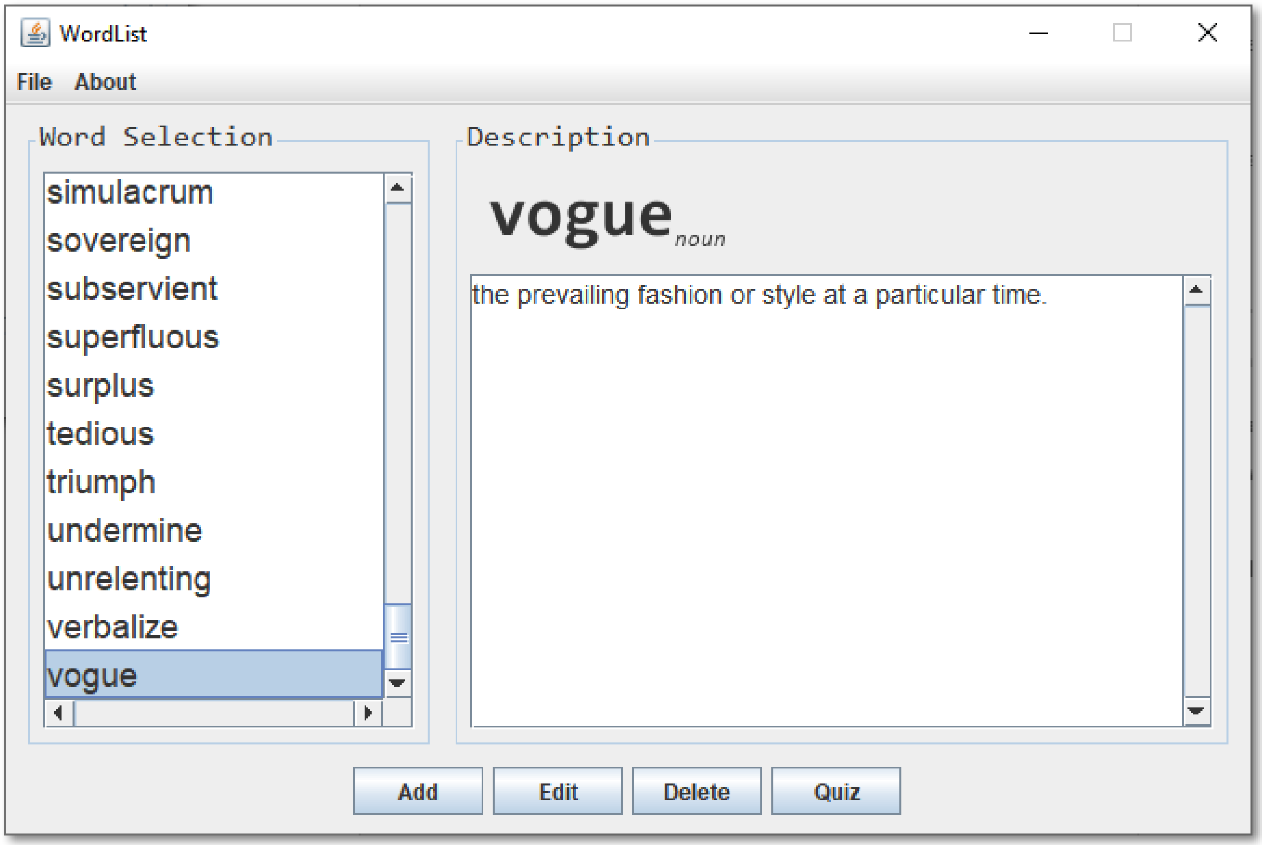
(Navigating to the [Open...] Menu)



(The Filename Dialog)

Click [Default] and then [OK] to load from the default file. If successfully loaded, the main window should update and display the contents of the file.

If the user desires to load from a different file, manually type in the name of the file and click [OK].



simulacrum
sovereign
subservient
superfluous
surplus
tedious
triumph
undermine
unrelenting
verbalize
vogue

vogue *noun*

the prevailing fashion or style at a particular time.

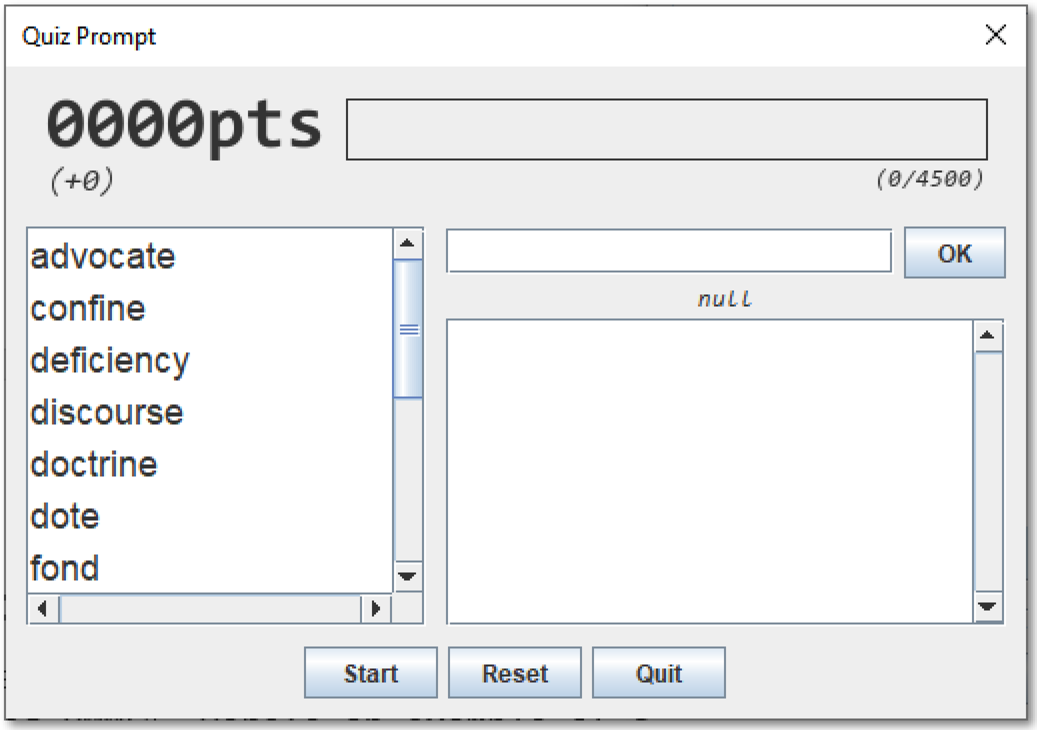
(The frame after loading from the default file.)

To save the current list to a file, navigate to the above menu bar, and go to [File] -> [Save As...][4]. The saving process follows the same process of loading. The current list can be saved to the default file, or the user can specify a different save location by manually typing it in the text field.

Taking a Quiz

Taking a quiz is perhaps the most notable feature of 'WordList'. To take a quiz with the current set of words, click the [Quiz] button. A quiz dialog[5] will appear.

[4] ctrl+s can also be used as a shortcut.
[5] The quiz dialog prompts 15 (at maximum) unique random words from the original list.



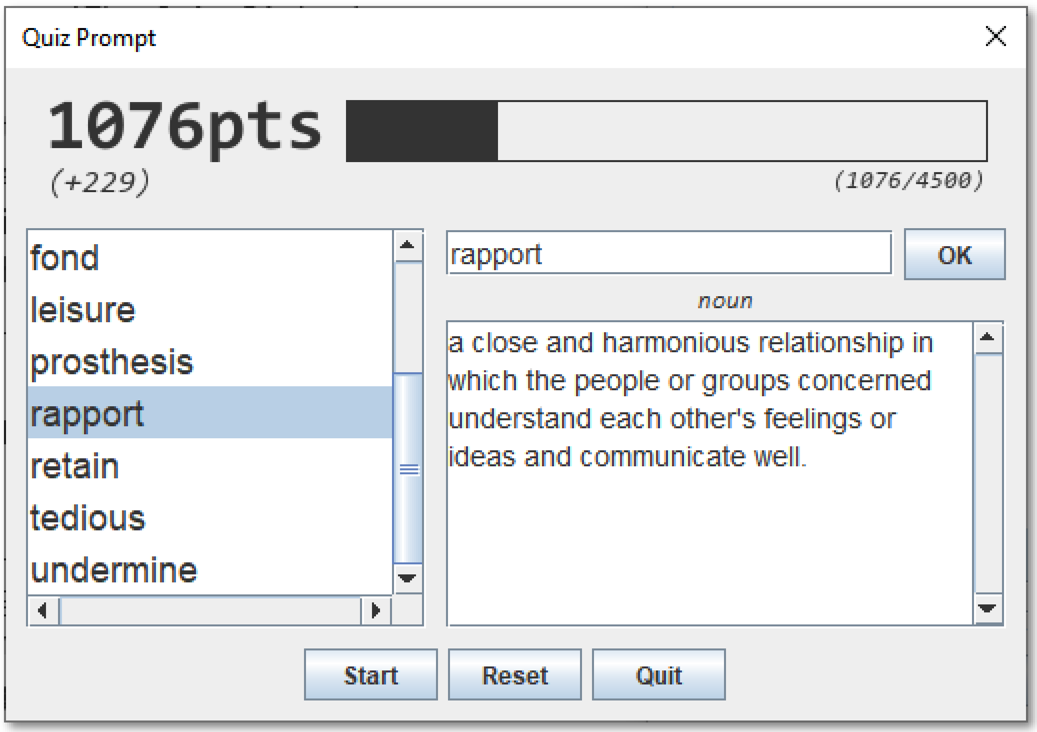
(The Quiz Dialog)

The list at the left contains all of the words that will be prompted. To start the quiz, click the [Start] button. The dialog will select a random word from the list, and display its pos[6] and definition. It is now up to the user to find the matching word from the list and enter it into the text field.

To fill in the text field, the user can either manually type the word in, or click any word from the list. The dialog will automatically fill in the text field when an item is selected.

Once the text field is filled, either click [OK] or press enter to check the guess. If the guess is correct, the user will be awarded with points that range between 100 and 300. If the guess is wrong, the user will lose 100 points.

[6] part of speech.


Quiz Prompt

1076pts

(+229)

(1076/4500)

fond
leisure
prosthesis
rapport
retain
tedious
undermine

rapport

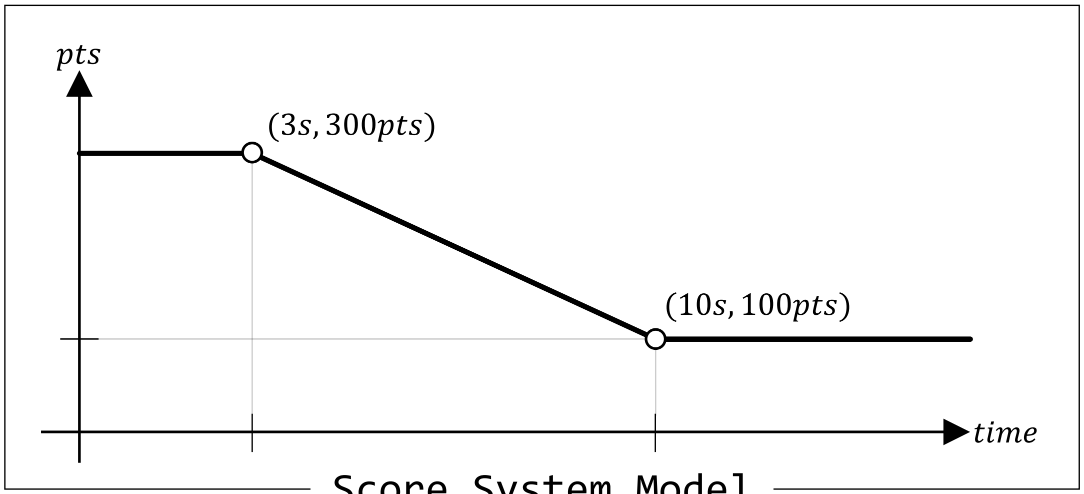
noun

a close and harmonious relationship in which the people or groups concerned understand each other's feelings or ideas and communicate well.

(Taking the Quiz)

Quiz Scoring System

The quiz scores are purely based on time. If the correct guess is made within 3 seconds, 300 points are awarded. If the correct guess is made, but the taken time exceeds 10 seconds, only 100 points are awarded. Correct guesses made between 3 - 10 seconds follow a linear model.



Score System Model

Because the quiz prompts at maximum 15 words for each round, this means the theoretical maximum achievable score is

$$300pts \times 15 = 4500pts\,.$$

If anyone’s interested, here is my attempt at scoring 4500pts with the default list.

Section 3. File Format

The files that are used to save/load words all follow a CSV[7] format$^{\text{(name, part of speech, definition)}}$.

Commas and newlines are special delimiter characters for csv files, so they must be escaped if the value itself uses them. ‘WordList’ uses an ampersand(‘&’) to escape these characters, so the ampersand itself must be escaped as well.

<Escape Formatting>

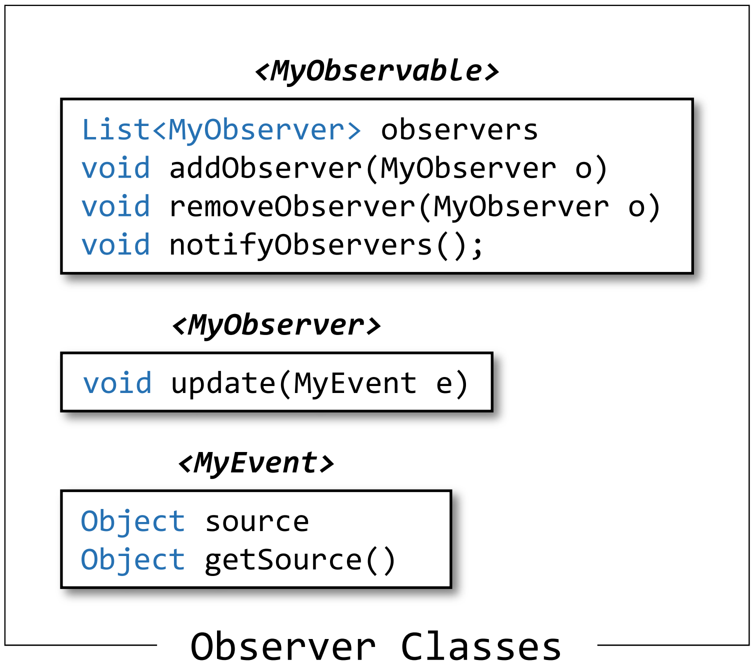
comma(,)	&com
newline(\n)	&ln
ampersand(&)	&amp

Sometimes, it may be faster to manually type the words into the file than to use the built-in [Add] button. If the user desires to apply this manual change, ensure to properly escape the special characters using the table above. Otherwise, the parser may fail to evaluate the properties of the words.

[7] Comma-Separated Values

Section 4. Design Pattern

This program follows the observer pattern. I've implemented two classes to follow this design pattern[8].

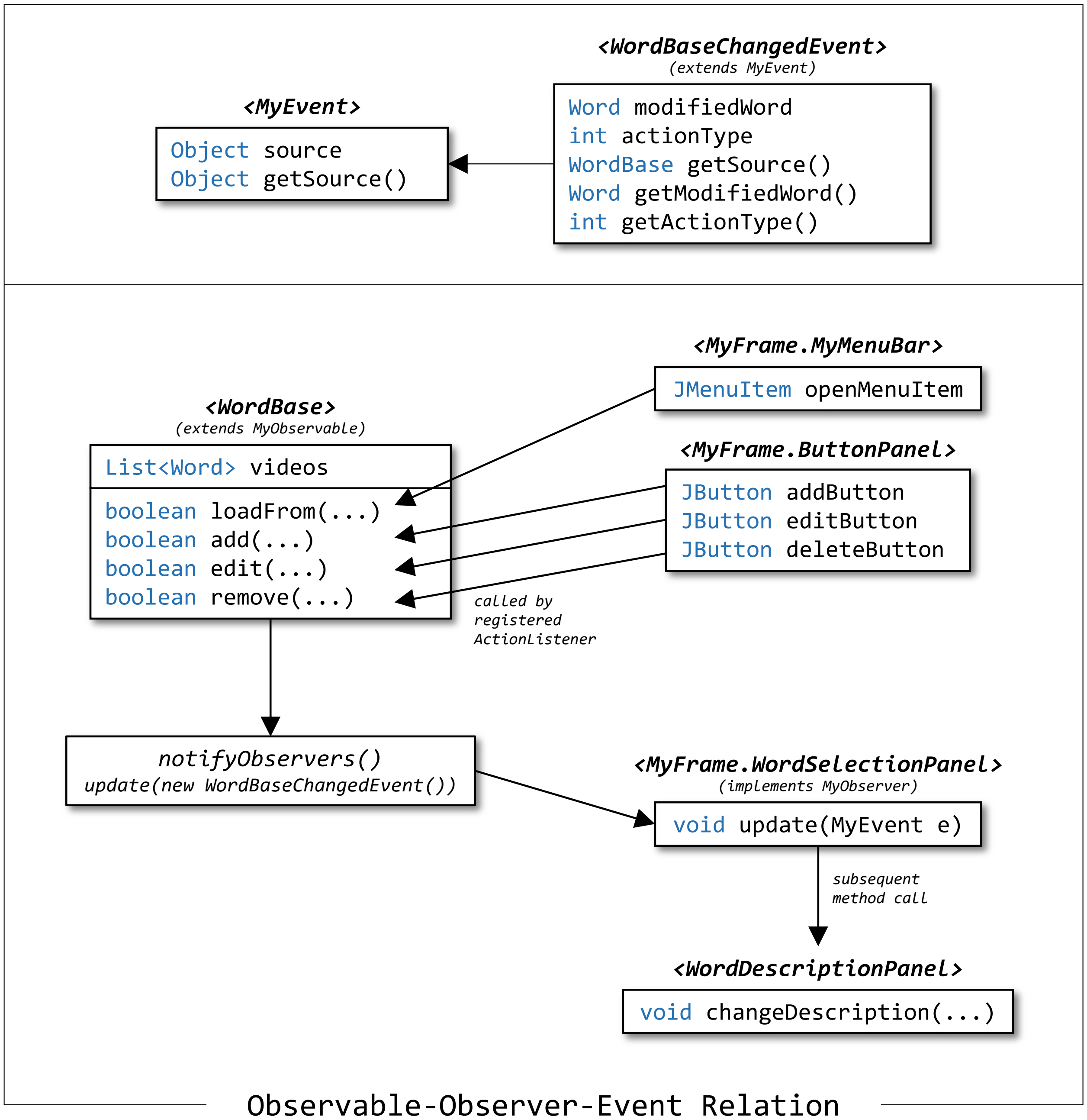


```
<MyObservable>
List<MyObserver> observers
void addObserver(MyObserver o)
void removeObserver(MyObserver o)
void notifyObservers();

<MyObserver>
void update(MyEvent e)

<MyEvent>
Object source
Object getSource()
```

Observer Classes

The diagram in the next page provides a basic illustration of how I applied this pattern to 'WordList'.

[8] Since the release of Java 9, the observer classes from java.util have been deprecated. To go around this issue, I simply rewrote the classes on my own. MyObservable replaces java.util.Observable, and MyObserver replaces java.util.Observer.



Observable-Observer-Event Relation

Calling any of the 4 methods:

```
WordBase.loadFrom(String filename)
WordBase.add(Word word, Boolean notify)
WordBase.edit(Word originalWord, Word newWord)
WordBase.remove(Word word)
```

notify the registered observers, which allows the frame to display the correct updated data from the base.