Your final system must run correctly on CASlab Linux, regardless of where you develop it.

You need not retain the old functionality. However, retaining both requires a deeper understanding of the design patterns used, so is a better result. The following directions assume you're replacing the old functionality; if you want to retain .txt editing, you'll need to study the existing code more thoroughly.

2 Getting Started

First, verify that the example editor works in your preferred development environment:

1. Create a new subdirectory and copy the contents of the `editor` subdirectory into it.
2. If you are running directly on CASlab Linux with access to a GUI, run `make` to compile and run `TextEditor`, the main class.
3. If you are running in your own environment, you must accomplish the equivalent (running Java on the main class, which compiles all the classes it depends on, and then running the result with a particular `.jar` file in the classpath).
4. Use the editor menus to make read and minor changes to `sampleText.txt`, then save it to a new file.

3 Version Management

You will need to apply some form of version management to develop your system incrementally and make sure you can revert to previous versions if changes introduce a bug. If you already are famliar with such a system, use it. Otherwise apply the following very simple ancient strategy.

1. Create a subdirectory named `V0` and move all the text editor files into it.
2. Copy the files with names starting with `Text` back into your main directory, renaming then to start with whatever is appropriate for your file type.
3. Edit the contents to change the class names
4. Before any significant new changes, create a `V1`, `V2`, etc. subdirectory and copy all the current files into them.

4 Sketch of Necessary Changes

I may augment this section in response to questions and comments.

4.1 MainPanel.java

This sets up the initial window used before any files have been opened. You shouldn't need to change it.

4.2 TextType.java

1. Delete the contents of `actionPairs`; this is for menu actions that already exist, rather than new ones you create.
2. Delete the body of the **`try`** clause in `getStaticMenu` but leave the **`try/catch`**. This is where