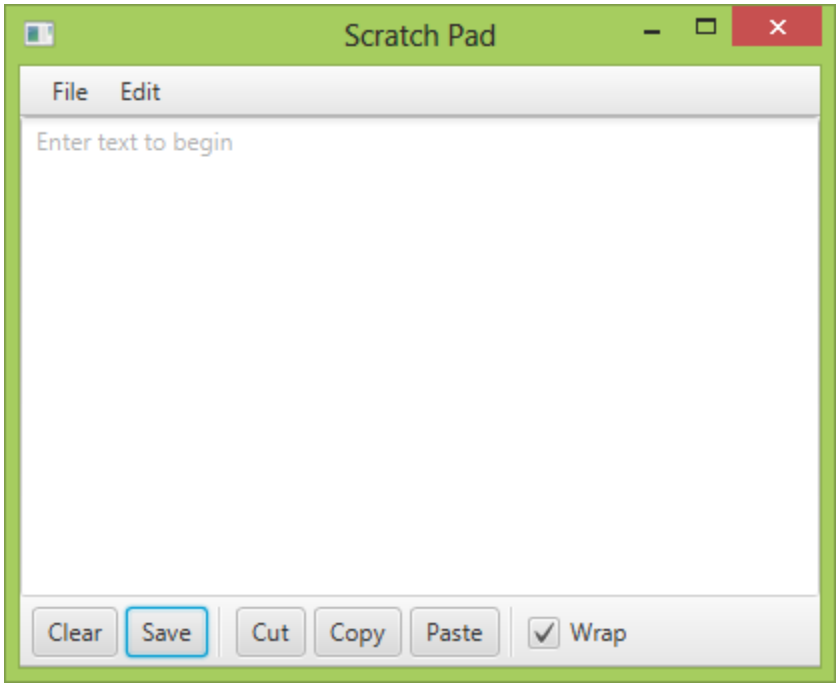
CIS 494 – Business Systems Development with Java – Spring 2015

Take-Home Midterm 1: Mini Project

Due on Blackboard: Friday, March 20 by 11:59 PM

Develop a basic text editor application, like Windows Notepad, using JavaFX. This application should have the following features



Requirements

- A visual interface implemented using JavaFX.
- Basic editing features like clear, cut, copy, paste.
- Option to wrap/unwrap text.
- Implementation of save and load features
    - The program should be able to save text to an internal text file. The program should be able to load data from the internal text file.
    - You do not need to provide the ability to save to user-specified file names.
- A "toolbar" that offers quick access to editing features.
- A menu system, with at least a File and Edit Menu.
- Functionality within the project should be properly distributed using appropriate methods.
- The project should be properly commented and follow good programming conventions such as using camel case, using descriptive variable/method names, etc.

Use Existing JavaFX Controls

- You will find most of text editor features already built into the TextArea control, which allows you to have multiple lines of text (in contrast to a TextField, which allows only single line of input). In effect, you only need to call the appropriate methods.

- Implement the save and load features using the File I/O procedures covered in class.

- Implement the menu system using the MenuBar, Menu, and MenuItem classes.
    - Have a MenuBar. Add Menus to the MenuBar. Add MenuItems to Menus. Add listeners for each MenuItem.
    - Implement Listeners using *Lambda expressions* to simply event handling.
    - Add the MenuBar to a pane of some sort and add the pane eventually to the Scene, i.e. MenuBar will not automatically display itself – follow same procedures as that for any JavaFX control to display the MenuBar.

- Implement the toolbar using the ToolBar class. The toolbar accepts standard Buttons as its toolbar buttons.
    - Implement Listeners for each Button.

- It is recommended to try out the BorderPane for organizing controls within your Scene; however, that is not a requirement. Feel free to use any organizing structure you want.

Optional Advanced Enhancements

- Use the HTMLEditor control to implement an advanced editor that can implement formatting
- Use small buttons or icons to offer formatting functionality.

Submitting Files

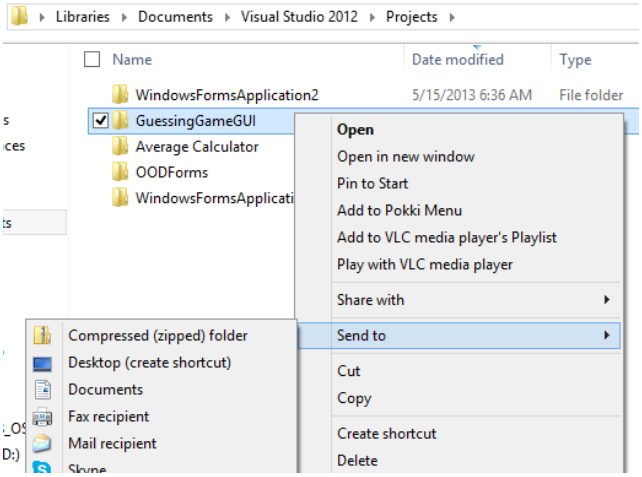
Submission should be made using a zip file that contains the entire Eclipse project folder. You will need to ***zip the entire project folder***. The folder will automatically contain the class source files as well as the compiled .class files.

ZIP file should be named: **MiniProject.zip**

.

How to Properly ZIP and Submit Your Project Files:

- Go to the folder within {Eclipse Workspace}\
- ZIP the entire top-level folder for your project by right-clicking your project folder and selecting Send to | Compressed (zipped) folder.
- Finally, submit the ZIP file using the submission link on Blackboard by the due-date and time listed on the assignment. Upload the ZIP file.



Using built-in windows zip tools: http://windows.microsoft.com/en-us/windows/compress-uncompress-files-zip-files

Verify your files BEFORE and AFTER submission:

- Check for actual class files being present in the folder before you zip it.
- ***Ensure that you are not zipping a short-cut to the folder.***
- After zipping, check file size. A file size under 4K likely does not contain all the files.
- Unzip, extract all files, and verify you see actual files, not a solitary short-cut.

- Uncompress your zip file before submitting and verify that files are present.
- Make sure you have ***submitted*** your file and not just saved a draft on Blackboard. ***A blue clock indicates a submission in progress, i.e. a draft, not a submission. The draft is accessible only to you. You will get a ZERO if you only ever save a draft on Blackboard and never submit your files.***

- Download your zip file after submitting, uncompress, and again verify that your files are present. Test your files in Visual Studio after uncompressing.

This takes an extra couple of minutes. Please do it if your grade is important to you. If you do this, you will not end up submitting a bad file. If you submit an empty file, or one containing only a shortcut, or a bad zip file, or a bad project file, you will receive a score of zero and your only recourse will be to do the makeup assignment at the end of the semester.