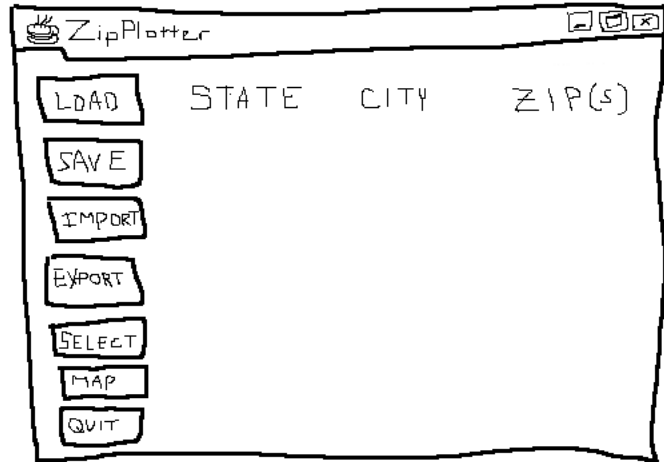


# Project #3 GUI

Computer Science 2334  
Spring 2008

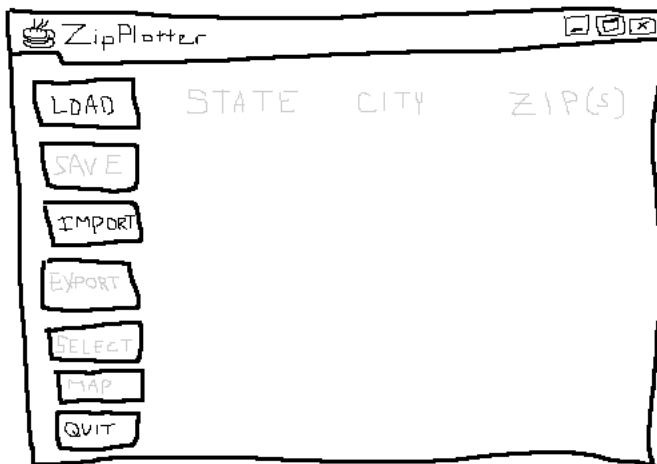
## Description:

The basic Graphical User Interface for Project #3 will have a main window with seven buttons down its left side and space for three lists in three columns to the right of the buttons. The seven buttons will be (in order) load, save, import, export, select, map, and quit. The three columns will be state, city, and zip(s). A rough sketch of this is shown in the figure to the right. (The figure is intentionally rough so that you will not attempt to copy it exactly.)



This main window will be augmented by additional windows that will pop up to help with file selection and mapping. In particular, a *JFileChooser* window will pop up to help with file selection for loading, saving, importing, or exporting data; and a browser will pop up to display the resulting map. Code for creating JavaScript to allow the browser to display the resulting map will be provided to you.

When your program is run initially, it should be clear from looking at the GUI that only load, import, and quit are possible. To make this clear, the save, export, select, and map buttons will be displayed differently (e.g., with the letters “grayed out”). Likewise, until data is loaded or imported, the state, city, and zip(s) columns will be similarly displayed in such a way that it should be clear there is nothing that can be done with respect to them. See, for example, the figure at left.

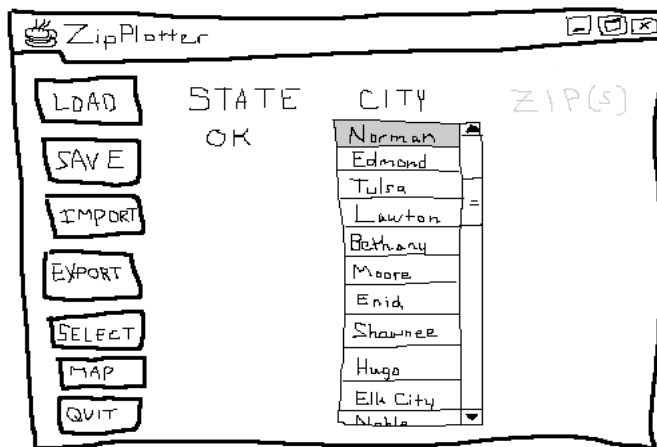
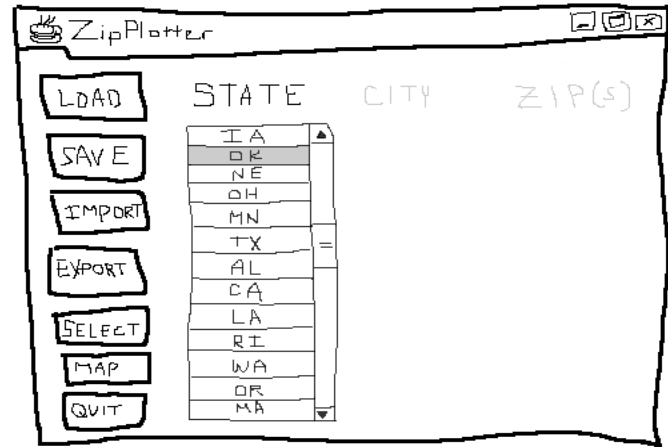


When the load or import button is selected, a *JFileChooser* window will pop up to help with file selection.

Once data has been loaded or imported, three more buttons will become active: save, export, and map, as shown below. These will provide the user with three corresponding options. (1) The user may choose to save the data, as described in the main project description. If the user makes this choice, an appropriate dialog will pop up to allow the user to specify a file name and to browse directories to determine where to save the file. (2) The user may choose to export the data, as described in the main

project description. Again, if the user makes this choice, an appropriate dialog will pop up to handle file name and directory selection. (3) The user may choose to create a map of the locations specified in the data. If this choice is made at this time, a browser window will pop up to display the resulting map.

Additionally, once data has been loaded or imported, the state column will become active. The contents of the state column will depend on the data. If there is a single state present in the data, the contents of the state column will simply be a display of that state abbreviation. If there are multiple states present in the data, the contents of the state column will be a *JList* of all the state abbreviations in the data. With this *JList*, the user should be restricted to picking a single state, as shown above.

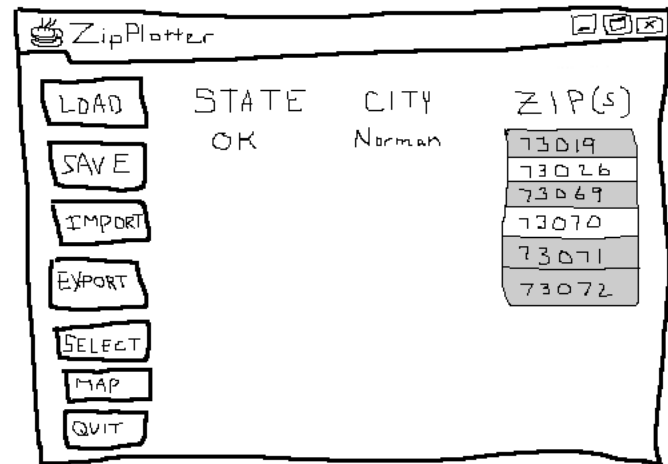


Once the user has chosen a single state (or if there was originally just a single state in the data), the select button will become active, as shown at left. If the user chooses the select option at this time, your program will keep a record of this selection. This means that if the user chooses to save, export, or map the data at this point, only the data relevant to this one state will be saved, exported, or mapped. To help the user see that a selection has been recorded, the state list will be replaced by a single state name, as shown at left.

The second result of the user has choosing a single state is that the city column will become active. (Also shown at left.) As with the state column, the contents of the city column will be determined by the data. If there is a single city for the state chosen, the contents of the city column will simply be a display of that city name. If there are multiple cities present for the chosen state, the contents of the city column will be a *JList* of all the city names associated with the state in the data. With this *JList*, the user should be restricted to picking a single city as shown above.

Note that the city column will list cities by name. It is possible to have multiple cities with the same city name in the database. That is okay. Within each state, the city names will be unique, so your *JList* of cities will only show a city name one time (at most).

After choosing a city from the city list (or if there was a single city associated with the selected state in the data), the user may press the select button to select a single city from the data. Again, what this means is that if the user chooses to save, export, or map the data at this point, only the data relevant to this one city in this one state will be saved, exported, or mapped. As with the state selection, to help the user see that a city selection has been recorded, the city list will be replaced by a single city name, as shown at right.

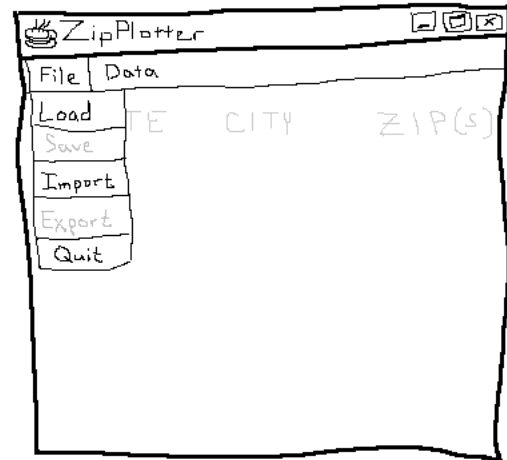


Also, after choosing a city from the city list, the zip(s) column will become active. As with the previous two columns, the contents of the zip(s) column will be determined by the data, with one data item simply being displayed and multiple data items requiring a *JList* for user selection. With this *JList*, the user should be allowed to select one **or more** zips as shown in the figure above.

*Extra Credit:*

There are several options for extra credit.

For 5 points of extra credit, you may replace the buttons from the description above with a menu bar. If you go this route, the menu bar will have a file menu with options to load, save, import, export, and quit, as shown in the figure to the right. It will also contain a data menu with options to select and map. All the functionality of this GUI will be the same as the functionality of the button-based GUI described above.



For 5 points of extra credit, you may allow the user to back up during the selection process, as well as go forwards. This means that after selecting the state, the user could go back and select a different state; after selecting the state and city, the user could go back and select a different state and/or city; and after selecting the state, city, and zip, the user could go back and select a different state and/or city and/or different zip codes.

For 3 points of extra credit, you may sort the state, city, and zip code lists.