A Graphical Interface to Octave using Zenity

Ritu Pradhan and Varenya Prasad

May 9th 2011

1 Final Project Description

We will be working on package focused project where we will explore and demonstrate the functionality of the Zenity package available for Octave.

Since the Zenity package adds the potential to present data in Octave with a graphical interface, we intend to explore the practicality of this package within the infrastructure of Octave commands we have worked with in class.

2 Package Description

There are number of bindings from Octave to Zenity that can be used for a Graphical User Interface interaction. The following features are available through the Zenity package:

- <u>Creating calendar windows</u> - Displays a date selection window
- <u>Displaying a text entry dialog</u> - Displays a text entry dialog.
- <u>Displaying a file selection dialog</u> - allows user to open a file/directory of files, and save a file
- <u>Displaying a graphical list of data</u> - options include a checklist, a radiolist and an editable list that allows users to modify values.

- A message box that can be used to display the status of interactions with Octave, for example, error messages and warnings.
- A notification icon that serves the same functionality as the message box above, except the message is displayed as an icon in the notification area.
- A text box that can be used to display data as read only or as read-/write so that the user can edit it.
- A progress bar to monitor how far Octave has progressed in executing a command.
- A selection scale window that allows the user to choose a parameter within the set ranges, and sets default value, and step sizes.

The above set of functions can be used for creating simple graphical user interfaces for octave.

3 Installing Zenity On Windows

1. Download the Zenity package from Octave-Forge available at :`http://octave.sourceforge.net/zenity/index.html`
2. After zenity package has been downloaded,assuming package is available in the file zenity-0.5.7.tar, it can be installed from the Octave prompt with the command pkg install zenity-0.5.7.tar
3. After loading package, it is possible to use the function provideds by the package by using the Octave command: *pkg load zenity*
4. Since the Zenity package executes as system calls, it needs to be added to the system path. To do so, download the .exe available at:`http://www.placella.com/software/zenity/#downloads`
   At this point, you are ready to modify the system path by adding an environemnt variable "*ZENITY_DATADIR*" to a value corresponding to the share folder within Zenity. For example: 'C: \Program Files (x86)\zenity \share \'

4 How To Use Zenity

Some of the Zenity package functions implemented in our project include:

- **zenity_entry(s = zenity_entry(text, entry_text, password))**
  The function zenity_entry displays a text entry dialog. We used this function as a prompt which asks for user to enter his/her name.

  For example:
  prompt = zenity_entry(”Welcome, Please Enter Your Name: ”);

- **zenity_message(zenity_message (text, type))**
  The function zenity_message displays a graphical message dialog. We implemented this function in two ways, one as a regular message dialog and other as an question dialogue which gives the user to allow selecting Yes or No option.

  For example:
  zenity_message(confirm,”Are you Ready for the Fun?”)
  zenity_message(’Thanks for joining us.GoodBye,Come Back Again.....’)

- **zenity_list(s = zenity_list(title, columns, data, options1, ...))**
  zenity_list displays a graphical list of data. The variable title sets the title of the list. The variable columns must be a cell array of strings of length N containing the headers of the list. The variable data must be cell array of strings of length NxM containing the data of the list.

  An example of the function implemented in following way:
  zenity_list(”Please Select an Option!”,”Select an Option”,
  ”Play!”,”PlotStatistics”,”Calendar”,”Quit”);
  The above statement displays the list of options for the user and the list include Play!, Plot Statistics, Calendar and Quit. The user can select the row/options and go into respective process.

- **zenity_calendar(d = zenity_calendar(title, day, month, year))**

  This function as name suggest displays the date selection window. The variable title sets the title of the calendar. The optional arguments day, month, and year changes the standard selected date.

An example of how we used this function:
zenity_calendar('Better Check the Dates!');
This displays the calendar pop up box with the title 'Better Check the Dates! and todays dates selected.

- **zenity_progress(h = zenity_progress(title, option1, option2))**
  This function displays a graphical progress bar. We used this function while loading the graphs.

  Example:
  zenity_progress('Plotting Graph....')

5 Zenity Within Our Project

To demonstrate the use of Zenity, we designed an interactive gaming environment that allows the user to play a mind reading game.
The user is given the followint options:

1. Play the Game
2. Plot the statistics
3. View Calendar, and
4. Quit the process.

At the end of the game, we record statistics for each instantiation of the game. When the user is finished playing, we use the data collected to graphically display the ranking of the player amongst those who have played the game before.

6 Game Description

Read Your Mind
This game challenges the player that they can read their mind, by asking the player to think of a number between 1 and 1000. Once the player has thought of a number, the computer attempts to guess the 'thought' by displying the player's number within a maximum of 10 guesses.

Instructions:

- The player decides on an integer between 1 and 1000, but does not type this number into the computer.
- The computer will make an initial guess of the number.
- For each guess the computer makes, the player must indicate if the guess is too high, too low or correct.
- The computer is only allowed 10 chances to guess the player's number.
- The player can quit at any time in the game.
- If the computer cant guess within 10 moves, the player WINS!

For this game, we collect data on how many guesses it took the computer to guess the player's number. The number of guesses is recorded only if the game isnt aborted midway.

7 Implementation Details and Limitations of Zenity

- Return Values from Zenity

  We noticed that the values we receive from the text entry dialog box is a vector. Given this vector, we needed to create a module that would take this vector value and compare it to a string value. The implementation of this is available in the file *vector_string_cmp.m* This function takes in a vector and a string and does a character comparison element by element.
  Input Parameters: The first parameter is the vector and the second parameter is the string.
  Return Values: Returns true if the vector and string are equal, false if not.

- Errors in Zenity calendar function

  When we created the zenity calendar using the in-built function, the calendar is displayed correctly. But when the user clicks on a date or selects between the "ok' and 'cancel' buttons, the function returns an error stating that none of the standard formats match the date string that is returned. This resulted in our implementation of the calendar option to be ineffectual.

Eventually this affected the manner in which our main program runs and the flow of execution.

8 Exercises:

Exercise 1: Using Zenity open a .txt file to read in the values stored in the file. Display the contents of the file in a graphical interface provided by the package.

Solution:

```
function y=open_display_file()
        pkg load zenity #Load the package

        # Open and read data from the file
        fid=fopen("words.txt","r");
        tline = fgets(fid);

        data = ""
        # Create a string to display the contents of the file.
        while ischar(tline)
                data = strcat(data,tline);
                tline = fgets(fid);
        end

        # Display data in the zenity provided GUI
        y=zenity_text_info('File Content',data)

        fclose(fid);      #Close file.
end
```

Note - For the given code, filename should be 'words.txt' and the line demiliter is a new line. Look at the attached words.txt for reference.

Exercise 2: Create a text-entry dialog box that prompts the user to input their name.

Solution: $prompt = zenity\_entry(''Please\,enter\,your\,name:'');$

Exercise 3: Using Zenity, create a list menu for the user to select between

- Entering your name, and
- Opening a file.

Solution: $zenity\_list(''Please\,Select\,an\,Option!'',''Select\,an\,Option'',$
$''Enter\,your\,name!'',''Open\,a\,file'');$

The workflow of the program after the user selects between the two options is left up to your discretion

References

[1] HappyMutant, *A Quick and Dirty Guide to LaTeX*, available at `http://happymutant.com/latex/latex.html#intro`.