Script:  
  
  
**1)   Introduction to the project**   
  
Hi, this is Varenya Prasad and Ritu Pradhan’s final project for our Scientific Computing Class. For our final project, we decided to delve into the Octave-Zenity package. Our motivation for the project stemmed from the lack of a GUI in Octave compared to MATLAB  
  
**2)    what is Zenity**  
Firstly, what is Zenity? Zenity is a compilation of functions for creating simple graphical user interfaces. The package currently supports multiple things like creation of file selection dialogs, or message windows. The package also lets you show notification icons, display a calendar, progress bars and other cool features.  
  
  
**3)   Installing Zenity**   
Now that we introduced zenity, lets go into the installing part.  
1. The installation part was a little trickier than we thought it would be. But anyways, lets get started.  First, download the Zenity package from Octave-Forge which is available at Octave Forge.  
  
2.  We even have to download the exe available at Placella url which is the Zenity for Windows.  
  
3.  After Zenity package has been downloaded, it can be installed from the Octave prompt with the command pkg **install zenity-0.5.7.tar**  
  
4. After installing package, to use and load the function provided by the Zenity package, type***pkg load zenity*** in the octave prompt.  
  
5. Finally, modify the system path by adding an environment variable “ZENITY DATADIR” and assign it a value corresponding to the share folder within Zenity. For example:   
After you export the environment variable "ZENITY\_DATADIR" and assign it a value  
C: \Program Files(x86)\zenity nshare \  
  
**4)    Our Application**  
Now we are ready to use the Zenity package in octave.  
To demonstrate the use of Zenity Octave package functions, we developed an interactive gaming environment that allows the user to play a mind reading game. The users are given the options to Play the Game, Plot the statistics, View Calendar and quit the process.  
  
Lets start out application:  
1.  In the octave command prompt, go into the respective file and type **final\_project.**  
2. The text entry dialogue box appears where user enters their name. Then graphical message dialogue box pops up displaying the player name.  
3. As we already said, the user are given the options to choose between Playing the Games, Plotting statistics, displaying calendar and quitting the application.  
4. Let’s start with the option of playing the game.  
  
a. Are you ready for fun? Off course YES!  
b. Firstly, About the Game(Guessing Number): Our game challenges the player that it can read their mind, by asking the player to think of a number between 1 an 1000. Once the player has thought of a number, the computer attempts to guess the ‘thought’ by displaying the player’s number within a maximum of 10 guesses.  
c. Ok, let’s start …..Think of a number between 1 and 1000. We will guess the number in maximum 10 moves. For the demo purpose, my guess is **907**. - 5 moves.  
  
At the end of the game, we record the statistics. Using the data collected from the end of game, user can view the game history selecting the option PLOT STATISTICS. Even though, the progress bar is not progressing. We included here just to show the graphical progress bar provided by Zenity package. This plot is drawn for Player against Number of moves. This is a little dry graph. But still we are showing just to demonstrate that we could plot using Zenity within octave.  
  
Users are also given the option to view the calendar, the default date is today’s date, and we can switch between dates and finally quitting the application where the application ends.  
  
**5)      Limitations of Zenity**   
Right off the bat, we noticed that the package was not built as robustly as one would prefer for it to be.  The file selection dialog function mistakenly gives the impression of having the ability to open a file, while all it does is return the absolute pathname of the selected file. Similarly the calendar function displays the calendar properly, but if one was to pick a date, it produces errors. The error eventually disrupted the flow of our application.  
  
One other issue we faced, which we were fortunately able to address was the values returned by Zenity. User responses were returned as vectors, which blocked any sort of textual comparison. To overcome this, we built a program that does a character by character comparison of the provided vector and string. We were then able to receive input from Zenity and parse it as required.