CS 112 Lab Assignment

Instructor: Dan Fleck
Lab: File IO

Overview

This lab will familiarize you with reading and writing files.

Reading Files
To read a file in Python you must open the file, and then read in the file (in chunks usually) and then close the file. In programming when you "open" a file, the file is not automatically read, you must read it yourself!

```
infile = open("myfile.txt","r") #Open in 'r'ead mode

# Read from the file until you read the end of file (EOF)
while infile:
      lineFromFile = infile.readline()
      if lineFromFile == '':
             break
      print "The line is: "+lineFromFile
```

infile.close() # Close the file… you must do this when you are done reading

Writing Files

To create a new file:

```
outfile = open("myfile.txt", "w")
```

WARNING: If myfile.txt already exists, it will be essentially deleted and you will start with an empty file!! Beware!

To append to an existing file (for example to add lines):

```
outfile = open("myfile.txt", "a")
```

So, to create a new file with three lines of text in it:

```
outfile = open("myfile.txt", "w")
outfile.write("Roses are red \n")
outfile.write("Violet are blue \n")
outfile.write("Python is cool \n")
outfile.write("And CS112 is to! \n")
outfile.close()
```

Where to get more information
The tutorial has more information also:

http://docs.python.org/tut/node9.html

The functions you can call with a file are documented here:
http://docs.python.org/lib/bltin-file-objects.html

And of course your book has more information as well.

Assignment

In this assignment you will be updating the Hangman game we created in class to store and retrieve how long each person solved the game. This is like a high score list, but with times instead.

When a user solves a word the Hangman game calculates how many seconds it took for them to solve the puzzle. At that point in the program you must:
1. Ask the user for their name
2. Store the time it took to solve, their name and word in a **besttimes.txt** file
Do not store names unless they solved the puzzle! You should append to the file, so if you run the program (and solve the puzzle) 10 times, you should have 10 lines in the file. If you then run it again and solve the puzzle, you should have 11 lines, etc…
3. Print out the table below

Then, after the game is done (whether they won or lost) you must display the best times file nicely formatted. It should look like this:

```
           Time    Person                      Word
-----------------------------------------------------------
       0m 15sec    George Washington           democracy

        2m 0sec    Barack Obama                president

        2m 0sec    Hillary Clinton             president

       0m 55sec    Jon Stewart                 funnyguy

       0m 19sec    Grover                      ha

       1m 17sec    Jay Leno                    tonight
```

Other Requirements

When storing data in a file, usually you format it for storage and then parse the file while you read it. This helps allow different programs to create output in any way they choose and keeps the data file compact.

So, for this project you need to format the file like this:

number of seconds~name~word

So a sample file would look like this:

```
15~George Washington~democracy
120~Barack Obama~president
120~Hillary Clinton~president
55~Jon Stewart~funnyguy
19~Grover~ha
77~Jay Leno~tonight
```

After reading a line of text you must split the line on the "~" character in order to get each individual field. Then you can format for screen output using each field (number of seconds, name, word) separately.

Hangman code is located at:
http://cs.gmu.edu/~dfleck/classes/cs112/spring08/labs/file_io/hangman.py

You also need the wordlist.txt file located at:
http://cs.gmu.edu/~dfleck/classes/cs112/spring08/labs/file_io/wordlist.txt
(The wordlist is all possible words for the game.)

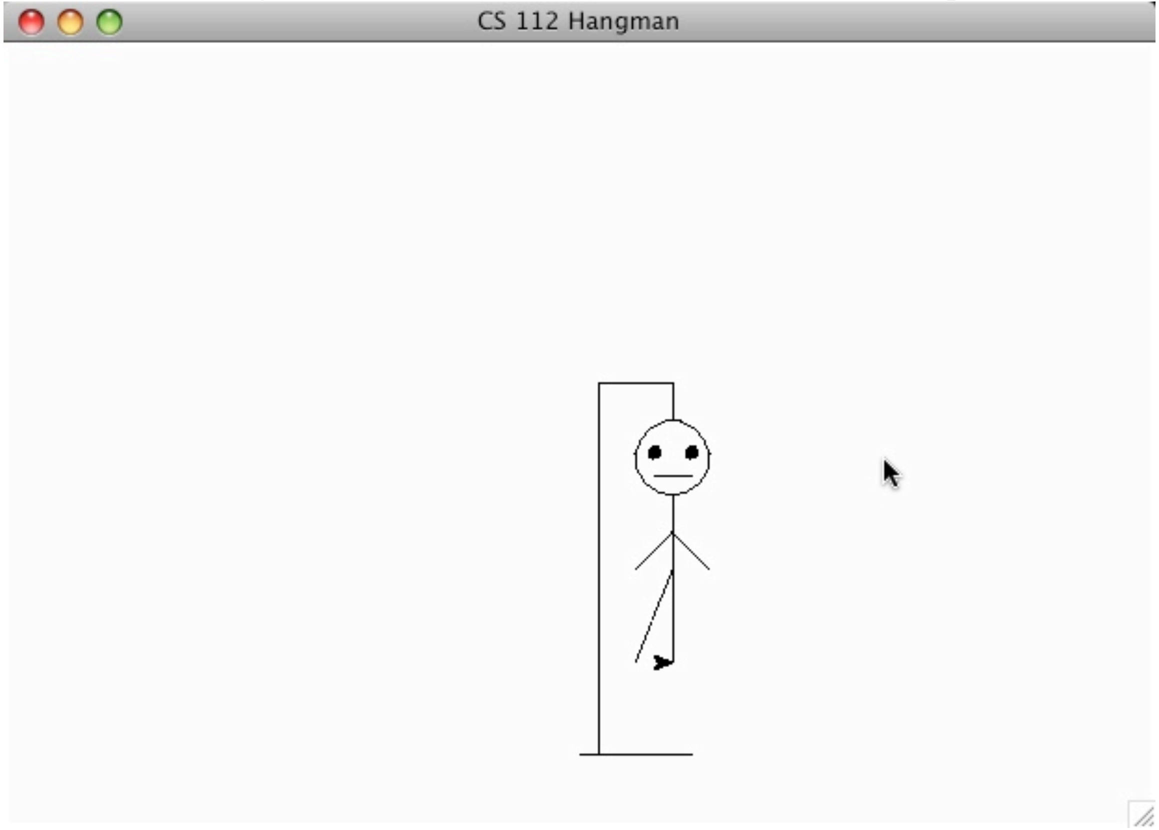
Challenges

- Convert from seconds to minutes/seconds
- Ask the user for their name
- Store the besttimes.txt file
- Read the besttimes.txt file
- Print out a nice table (as below) of the besttimes.txt file

Sample Output from Hangman:

```
>>> ================================ RESTART
================================
>>>
Enter a word to guess: test
_ _ _ _
Enter a letter or 1 to quit ->t
t _ _ t
Enter a letter or 1 to quit ->e
t e _ t
Enter a letter or 1 to quit ->s
```

```
 *** YOU WIN!! Good job! ***


Please, enter your name: Dan Fleck
User result is saved
           Time    Person                      Word
-----------------------------------------------------------
       0m 15sec    George Washington           democracy

        2m 0sec    Barack Obama                president

        2m 0sec    Hillary Clinton             president

       0m 55sec    Jon Stewart                 funnyguy

       0m 19sec    Grover                      ha

       1m 17sec    Jay Leno                    tonight

        0m 3sec    Dan Fleck                   test


Done. Please close the graphics window.

Note, the Hangman should look about like this if you lose:
```



What to turn in:

1. Modified hangman.py source code
2. Output from your program running
3. A sample besttimes.txt output file you created by running your program