

Hangman Game

September 27, 2015

0.1 Python Code for the Hangman Game

```
In [8]: # 6.00 Problem Set 3
#
# Hangman game
#

# -----
# Helper code
# You don't need to understand this helper code,
# but you will have to know how to use the functions
# (so be sure to read the docstrings!)

import random
import string

WORDLIST_FILENAME = "words.txt"

def loadWords():
    """
    Returns a list of valid words. Words are strings of lowercase letters.

    Depending on the size of the word list, this function may
    take a while to finish.
    """
    print "Loading word list from file..."
    # inFile: file
    inFile = open(WORDLIST_FILENAME, 'r', 0)
    # line: string
    line = inFile.readline()
    # wordlist: list of strings
    wordlist = string.split(line)
    print " ", len(wordlist), "words loaded."
    return wordlist

def chooseWord(wordlist):
    """
    wordlist (list): list of words (strings)

    Returns a word from wordlist at random
    """
    return random.choice(wordlist)

# end of helper code
```

```

# -----

def isWordGuessed(secretWord, lettersGuessed):
    '''
    secretWord: string, the word the user is guessing
    lettersGuessed: list, what letters have been guessed so far
    returns: boolean, True if all the letters of secretWord are in lettersGuessed;
        False otherwise
    '''
    # FILL IN YOUR CODE HERE...
    lettersMatch = True
    for char in secretWord:
        if char not in lettersGuessed:
            lettersMatch = False
            break
    return lettersMatch

def getGuessedWord(secretWord, lettersGuessed):
    '''
    secretWord: string, the word the user is guessing
    lettersGuessed: list, what letters have been guessed so far
    returns: string, comprised of letters and underscores that represents
        what letters in secretWord have been guessed so far.
    '''
    # FILL IN YOUR CODE HERE...
    stringGuessed = ''
    for num in range(len(secretWord)):
        if secretWord[num] in lettersGuessed:
            stringGuessed = stringGuessed + secretWord[num]
        else:
            stringGuessed = stringGuessed + '_'

    return stringGuessed

def getAvailableLetters(lettersGuessed):
    '''
    lettersGuessed: list, what letters have been guessed so far
    returns: string, comprised of letters that represents what letters have not
        yet been guessed.
    '''
    # FILL IN YOUR CODE HERE...
    newString = ''
    myString = string.ascii_lowercase
    for num in range(len(myString)):
        if myString[num] not in lettersGuessed:
            newString = newString + myString[num]

    return newString

def hangman(secretWord):
    '''

```

secretWord: string, the secret word to guess.

Starts up an interactive game of Hangman.

** At the start of the game, let the user know how many letters the secretWord contains.*

** Ask the user to supply one guess (i.e. letter) per round.*

** The user should receive feedback immediately after each guess about whether their guess appears in the computers word.*

** After each round, you should also display to the user the partially guessed word so far, as well as letters that the user has not yet guessed.*

Follows the other limitations detailed in the problem write-up.
'''

FILL IN YOUR CODE HERE...

```
print "Welcome to the game, Hangman!"
print "I am thinking of a word that is " + str(len(secretWord)) + " letters long"
print "-----"
print "You have 8 guesses left."
print "Available letters:" + string.ascii_lowercase
myGuess = raw_input("Please guess a letter:")
myGuessLower = myGuess.lower()
num = 8
myS = []
while num > 0:
    if myGuessLower in secretWord:
        if myGuessLower in myS:
            print 'Oops! You\'ve already guessed that letter: '
            + getGuessedWord(secretWord,myS)
            print "-----"
            print "You have " + str(num) + " guesses left."
            print "Available letters:" + getAvailableLetters(myS)
            myGuess = raw_input("Please guess a letter:")
            myGuessLower = myGuess.lower()

        else:
            myS.append(myGuessLower)
            print "Good Guess:" + getGuessedWord(secretWord,myS)
            print "-----"
            if isWordGuessed(secretWord,myS):
                print "Congratulations, you won!"
                break
            else:
                print "You have " + str(num) + " guesses left."
                print "Available letters:" + getAvailableLetters(myS)
                myGuess = raw_input("Please guess a letter:")
                myGuessLower = myGuess.lower()

    else:
        if myGuessLower not in myS:
            myS.append(myGuessLower)
```

```

print "Oops! That letter is not in my word: " + getGuessedWord(secretWord,myS)
print "-----"
num -= 1
if num == 0:
    print "Sorry, you ran out of guesses! The word was " + secretWord
    break
print "You have " + str(num) + " guesses left."
print "Available letters:" + getAvailableLetters(myS)
myGuess = raw_input("Please guess a letter:")
myGuessLower = myGuess.lower()

elif myGuessLower in myS:
    print 'Oops! You\'ve already guessed that letter: '
    + getGuessedWord(secretWord,myS)
    print "-----"
    print "You have " + str(num) + " guesses left."
    print "Available letters:" + getAvailableLetters(myS)
    myGuess = raw_input("Please guess a letter:")
    myGuessLower = myGuess.lower()

```

In [9]: *#When you've completed your hangman function, uncomment these two lines
and run this file to test! (hint: you might want to pick your own
#secretWord while you're testing)*

```

secretWord = chooseWord(loadWords()).lower()
hangman(secretWord)

```

Loading word list from file...

55909 words loaded.

Welcome to the game, Hangman!

I am thinking of a word that is 8 letters long

You have 8 guesses left.

Available letters:abcdefghijklmnopqrstuvwxyz

Please guess a letter:e

Good Guess:_ _ _ _ _ e_

You have 8 guesses left.

Available letters:abcdefghijklmnopqrstuvwxyz

Please guess a letter:a

Oops! That letter is not in my word: _ _ _ _ _ e_

You have 7 guesses left.

Available letters:bcdfghijklmnopqrstuvwxyz

Please guess a letter:n

Good Guess:_ _ _ _ _ ne_

You have 7 guesses left.

Available letters:bcdfghijklmnopqrstuvwxyz

Please guess a letter:d

Oops! That letter is not in my word: _ _ _ _ _ ne_

You have 6 guesses left.

Available letters:bcdfghijklmnopqrstuvwxyz

Please guess a letter:m

```

Good Guess:_ _ m_ _ ne_
-----
You have 6 guesses left.
Available letters:bcfghijklpqrstuvwxyz
Please guess a letter:p
Oops! That letter is not in my word: _ _ m_ _ ne_
-----
You have 5 guesses left.
Available letters:bcfghijklqrstuvwxyz
Please guess a letter:h
Oops! That letter is not in my word: _ _ m_ _ ne_
-----
You have 4 guesses left.
Available letters:bcfghijklqrstuvwxyz
Please guess a letter:g
Oops! That letter is not in my word: _ _ m_ _ ne_
-----
You have 3 guesses left.
Available letters:bcfijklqrstuvwxyz
Please guess a letter:b
Good Guess:_ _ mb_ ne_
-----
You have 3 guesses left.
Available letters:cfijklqrstuvwxyz
Please guess a letter:y
Oops! That letter is not in my word: _ _ mb_ ne_
-----
You have 2 guesses left.
Available letters:cfijklqrstuvwxyz
Please guess a letter:l
Oops! That letter is not in my word: _ _ mb_ ne_
-----
You have 1 guesses left.
Available letters:cfijkoqrstuvwxyz
Please guess a letter:f
Oops! That letter is not in my word: _ _ mb_ ne_
-----
Sorry, you ran out of guesses! The word was combines

```

In []: