COSC 341 Program 3 Python Program

Distributed 3/2/09 Due 4/7/09

1. Using a dictionary, do Exercise 3 from the on-line text How to Think Like a Computer Scientist (http://openbookproject.net//thinkCSpy/ch12.xhtml#auto18)

Write a program called alice_words.py that creates a text
file named alice_words.txt containing an alphabetical listing
of all the words found in
http://openbookproject.net//thinkCSpy/resources/ch10/alice_in_wonderland.txt
The words should be case-insensitive.

The first 10 lines of your output file should look something like this:

```
Word          Count
======================
a              631
a-piece        1
abide          1
able           1
about          94
above          3
absence       1
absurd        2
```

Add a loop which asks the user for the desired keyword and then returns the count. End the loop in any reasonable way.

2. Use Turtle graphics to draw a sprite at location (0,0). Then move the sprite to location specified by the user (e.g., 100, 0) dropping a piece of gold every 10 pixels.
The sprite is at least a small rectangle abutting a small filled circle. You have your choice of colors and sizes on the sprite, just make it less than 10X10. Gold is represented as a small filled circle (radius = 4) color yellow.

Turtle graphics:

```
import turtle
```

Use `help(turtle)` to get documentation. Test these commands out in the idle window before attempting them in a program.

`turtle.up()` set the pen up
`turtle.down()` put the pen down
`turtle.right(degrees), turtle.left(degrees)`: turn the direction the turtle is facing
`turtle.forward(distance)`: move the turtle forward the amount indicated
`turtle.backward(distance)`: move the turtle backward
`turtle.goto(x, y)`: move turtle to specified point
`turtle.color(r, g, b)`: Set the color of the pen. The arguments are floating point numbers (between 0.0 and 1.0) indicating the amount of red, green and blue respectively.
`turtle.circle(radius)`: draw a circle of the indicated radius.
`turtle.write(string)`: Write a string
`turtle.fill(flag)`: Before drawing a figure, use turtle.fill(1), then draw the figure, then use turtle.fill(0).

IDLE doesn't play well with other graphics windows. Some windows may hang. To avoid this, `import os` and then use `os._exit(1)` as the last line in your program.

If you're having trouble with the turtle graphics window closing too quickly, `import time` and use `time.sleep(seconds)` The program will wait the number of seconds indicated before proceeding to the next instruction.

Turn in for each problem:
Source code
Screen shot of each program running
Program (1): ask for the count for 'alice', 'queen', 'rabbit'
Program (2): after all the gold has been dropped.

Grade based on:

Satisfying specifications	70%
Elegance and style	15%
Documentation and user-friendliness	15%