Assignment #8  Drawing Shapes with Turtle Graphics

This assignment asks you to use word search to understand instructions for drawing shapes on a Turtle canvas. (Make sure you look at the next page for guidance!) You are to define a small vocabulary that includes at least 3 shapes, at least 3 sizes, at least 3 colors, and at least 3 positions. For example, a vocabulary could be:

- **Shapes:** square, triangle, hexagon
- **Sizes:** small, medium, large
- **Colors:** red, blue, green
- **Positions:** top, center, bottom

Write a program that repeatedly asks the user what to draw, using the turtle textinput() function. Your program should interpret each response, which may be a free-form sentence that includes words from your vocabulary, and then it should draw what is described.

For example, your code should be able to process requests like:

“Draw a small red triangle in the center of the canvas.”

and recognize the vocabulary words “small”, “red”, “center”, and “triangle”. Your program should know how to interpret this specification and draw the specified shape. (Note: This example uses my vocabulary words --- your vocabulary words may be different!)

If the user types something that does not include words from your vocabulary, then your program should indicate that it doesn’t understand, or ask a further question to clarify. You can use random and randint to vary the output, so that the shapes don’t pile up on each other.

**IMPORTANT:** Be sure to specify what your vocabulary is in the comment at the beginning of your program.

Your program must contain a loop that asks for input. You may also use functions called from within the loop.
**Extra credit:** If you do something very creative and spectacular, up to 5 points extra credit are possible. (Adding another color does not count as spectacular.)

**Implementation guidance:**

You can use these programs for reference on Turtle graphics relevant to this assignment:

http://cs.nyu.edu/courses/spring14/CSCI-UA.0002-008/hw/random_poly.py

http://cs.nyu.edu/courses/spring14/CSCI-UA.0002-008/hw/random_text.py

To interpret a free-form sentence, you can use the Python `in` and `not in` string operators to search for vocabulary words. Here is sample code that has a small vocabulary (“stomach”, “head”, “muscle”, “back”, “sharp”), and interprets a user's medical complaints.

```python
sentence = input("Where does it hurt? ")

if "stomach" in sentence :
    if "sharp" not in sentence :
        print("Take some alka-selzer")
    else :
        print("Go to the doctor to check for appendicitis.")

elif "head" in sentence :
    print("Take some tylenol or advil")

elif "muscle" in sentence :
    print("Try applying some heat.")

elif "back" in sentence :
    print("Don't do anything too strenuous for a few days")

else :
    print("I don't understand.")
```