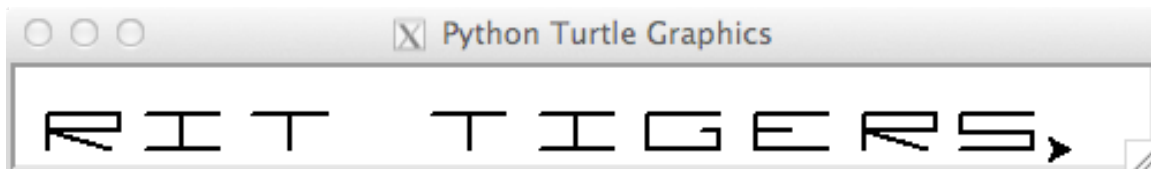


Problem-Based Intro. to Computer Science Banner Typography Lab

09/06/2011

1 Problem

Using Python's turtle graphics module, we would like to design a program that draws the textual message "RIT Tigers" as shown below.



1.1 Problem-solving Session (20%)

You will work in a group of five to six students as determined by your instructor. Each group will work together to complete the following activities. Each activity is worth 4%.

1. Draw on a piece of graph paper how you want the letters to appear. Label enough points with coordinates so that the exact size of the letters, and the spacing between them, is clear.
2. Write pseudocode to draw the letter "S".
3. Develop a design for writing messages where the turtle always starts at the same position and orientation relative to where the letter will appear. Write down what the assumed initial orientation of the turtle will be, and where the turtle will be when the drawing of the letter is done and it is ready for the next letter.
4. Re-write the letter "S" pseudocode as a function that obeys your decisions above (if it does not do so already).
5. Write the letter "R" pseudocode as a function that obeys your design decisions.

At the end of problem-solving, hand in your work, one copy per group, and number each item.

1.2 Implementation (80%)

Each student will *individually implement* and submit their own solution to the problem as a Python program named `rit_tigers.py`. Also you will submit a text document, `readme.txt`, which contains a description of your design, pseudocode, and test cases.

1.2.1 Program Operation

The program shall be runnable from the command line of a terminal window as follows:

```
python3 rit_tigers.py
```

When run, the program must pause after drawing the figure and wait for the user to press the ENTER/RETURN key before the program terminates.

1.2.2 Grading

Your grade is based on these factors:

- 55%: The program draws a message containing the phrase “RIT TIGERS” fitting within the canvas on one line. (Your program may draw other words or items after this; see below.)
- 10%: The design promotes code reuse by defining reusable functions.
- 5%: The code follows the style guidelines on the course web site.
For example, the code should be commented well enough so that it would be fairly easy for someone to reuse your code to (a) spell a different word with the letters provided, and (b) add a function for a new letter.
- 10%: The `readme.txt` file contains appropriate design documentation, run and test information. Make sure things are well documented here so that the grader of your work knows what to examine and do. The file should have the following items/sections:
 - A brief summary of your design; this should be no more than 4 sentences.
 - Instructions describing how to run the program.
 - Description of additional tests and features in the code, including answers to these questions:
 - * What variations of drawing position did you test?
 - * What other words did you experiment with?
 - * How did you test your functions to make sure they work in any order?
 - * What are the standard, expected outputs from running the program?
 - * Are there additional, non-required outputs from running the program? (Your program may display more than the required text in one run.)
 - * What sections of code, if any, could be changed to make your program do different things.

1.3 Submission

Compress your `rit_tigers.py` and `readme.txt` into one file named **rit_tigers.zip**, and submit that to the **myCourses dropbox** for this lab.

On machines that have the zip compression program, the command is shown here:

```
zip rit_tigers.zip rit_tigers.py readme.txt
```