Computer Science I — CSCI-141
Draw Your Song! — Homework 5

09/09/2020

1 Problem Description

Write a program that reads the lyrics of a song from a text file and uses the turtle package to draw a picture of the song with a colorful square for each letter. The length of each line defines the 'shape' of the text and shall be preserved by a ragged-right-edge display in the picture.

1.1 Implementation Requirements

- The program shall be in a single file named `songs.py`.
- The `square()` function shall take one parameter as input, a string that is a color, and draw a single square of size 10×10 filled with the given color.
- The `paint_line()` function shall take one parameter as input, a string representing a line from a text file, and draw a row of colorful squares corresponding to the characters in the line.
  `paint_line()` shall use five different colors defined as follows:
  - color 1 for any character, `ch`, such that $0 \leq$ `ord(ch)` < 70,
  - color 2 for any character, `ch`, such that $70 \leq$ `ord(ch)` < 100,
  - color 3 for any character, `ch`, such that $100 \leq$ `ord(ch)` < 110,
  - color 4 for any character, `ch`, such that $110 \leq$ `ord(ch)` < 122, and
  - color 5 for any character, `ch`, such that `ord(ch)` ≥ 122.

  These are *fixed color settings* embedded into your program, and it is your decision on what to make them.
  Before drawing the row, `paint_line()` shall remove the newline and any trailing whitespace from the line.
  After drawing the row, `paint_line()` shall move the turtle to the beginning of the next 'row' on the canvas to be ready to paint the next line of the file, if any.
- The `picture()` function takes one parameter as input, a file name, opens the file, and draws the picture by calling the `paint_line()` function for each line of text in the file. The `picture()` starts drawing in the upper left corner of the canvas.
- The `main()` function prompts the user for a file name and calls `picture()` to do the work, and waits for the user to close the canvas before terminating.

1.2 Constraints

The following are prohibited.

- Do **not** use the Python `readlines()` function for input.

1.3 Sample Runs

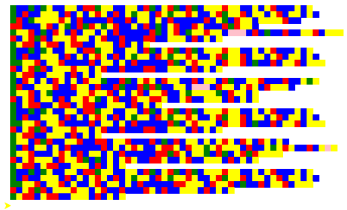
For example, suppose the file `song.txt` contains the following:

```
Strawberries, cherries and an angel kissing spring
My summer wine is really made from all these things
I walked in town on silver spurs that jingled to
A song that I had only sang to just a few
She saw my silver spurs and said let's pass some time
And I will give to you, summer wine
Oh. oh, oh, summer wine
```

Using color values that approximate "red", "green", "yellow", "blue", and "pink", the program run produced the result shown below.

Below is output from a solution when run on a longer song file.

```
Strawberries, cherries and an angel kissing spring
My summer wine is really made from all these things
I walked in town on silver spurs that jingled to
A song that I had only sang to just a few
She saw my silver spurs and said let's pass some time
And I will give to you, summer wine
Oh. oh, oh, summer wine
Strawberries, cherries and an angel kissing spring
My summer wine is really made from all these things
Take off your silver spurs and help me pass the time
And I will give to you, summer wine
Oh, summer wine
My eyes grew heavy and my lips they could not speak
I tried to get up but I couldn't find my feet
She reassured me with the unfamiliar line
And then she gave to me, more summer wine
Woh, woh, oh, summer wine
Strawberries, cherries and an angel kissing spring
My summer wine is really made from all these things
Take off your silver spurs and help me pass the time
And I will give to you, summer wine
Mm, summer wine
When I woke up, the sun was shining in my eyes
My silver spurs were gone, my head felt twice its size
She took my silver spurs, a dollar and a dime
And left me craving for, more summer wine
Oh, oh, summer wine
Strawberries, cherries and an angel kissing spring
My summer wine is really made from all these things
Take off those silver spurs, help me pass the time
And I will give to you my summer wine
Oh, oh, summer wine
```



1.4 Available Text Files

In the same folder as this assignment document are some text files that you could use for development. Click on the name of the file, and the browser will display it. Then right-click or use the menu to "Save as", and direct the browser to save the file into the same directory as your source file.

You may also test with a source code file.

1.5 Notes and Tips

- The `paint_line()` function needs to move to the next line after drawing the last element on the current line. One way to do this is to count the number of characters just drawn and multiply that by the size of the squares to get the amount to back up along the horizontal dimension. Then simply move down the size of the square side to position the turtle at the location where the next line should be.
- If the file read has very long lines, it is possible that the line runs over the right edge of the canvas. This is acceptable, but the viewer would not see the ends of very long lines.
- **Do not be concerned if your program produces different results.** Output will be different for any/all of these reasons:
  - different color selections.
  - different positioning near the upper left of the canvas.
  - different file content.
- The text file to open must be in the same directory, or folder, that has the program. Otherwise, the program will not be able to find the file when it runs. The `open()` function will then fail and report the error.
- Use the `turtle.goto()` function to start the drawing somewhere in the upper left corner of the canvas.
- You can set the turtle speed to 0 when first developing the solution. That will speed things up, but `tracer()` is even *faster*.
- The function `turtle.tracer()` is useful to accelerate the drawing of complex graphics. You can use following arguments to the call:
  `turtle.tracer(0, 0)`. This turns off the interaction to vastly speed things up by hiding the rendering.

2 Grading

- 15%: The implementation outputs the expected picture.
- 15%: The `square()` function draws a single square of size 10×10 filled in with a color value passed as an argument.
- 30%: The `paint_line()` function loops over characters in a string and draws a single line of squares using the five colors.
- 5%: The drawing starts near the upper left corner of the canvas.

- 30%: The `picture()` function opens the named file, reads the content line by line, and processes each line using a loop.
- 5%: The `main()` function gets the file name from the user and passes it to `picture()`, which does all the work.

- *Minus Points up to 5% off the top*
  Naming and Style: The files and functions have the specified names, and the code follows the course style and format guidelines for author, docstrings, etc.
  Also the submission used the correct *zip* and upload process.

3 Submission

Use zip to create the file `hw05.zip`, that contains the `songs.py` solution. Zip only the program file. Graders will use their own text files.

Upload the zip file to the correct MyCourses assignment dropbox by the deadline.