RIT-Dubai Fall 2024

GCIS-123

Class Activity & Problem Solving #02

Turtle Drawing

## Goals of the Assignment

The goal of this assignment is to give you more practice using loops and basic string processing. You will write a basic command-line interface that will prompt the user to enter strings of characters, and will interpret those strings into works of pixel art.

## **Before starting**

Please be advised that you are required to apply ***Incremental Development*** in your project. Failure to utilize this approach when developing the methods will result in a deduction of 50% of the marks for the corresponding function.

## **Part 1 (50%)**

1. **(5%)** Add a function to your pixart module named **get\_color().** This function takes a character as parameter (i.e. ‘0’, ‘1’, etc.) and will return the corresponding string for this character as in the table below.
   1. For example, **get\_color(‘5’)** will return the string ‘green’.
   2. For example, **get\_color(‘2’)** will return the string ‘red.
   3. For example, **get\_color(‘X’)** will return **None** as ‘X’ has no corresponding color.

|  |  |  |
| --- | --- | --- |
| **Character** | **Turtle Color** | **Color** |
| 0 | 'black' |  |
| 1 | 'white' |  |
| 2 | 'red' |  |
| 3 | 'yellow' |  |
| 4 | 'orange' |  |
| 5 | 'green' |  |
| 6 | 'yellowgreen' |  |
| 7 | 'sienna' |  |
| 8 | 'tan' |  |
| 9 | 'gray' |  |
| A | 'darkgray' |  |

1. **(10%)** Add the function **draw\_color\_pixel(color\_string, turta)** to pixart module. This function takes color\_string (i.e. string) and a turtle object as parameters, and it draws one pixel of that particular color.
   1. For example: draw\_color\_pixel(‘red’, turta) will draw a red pixel:



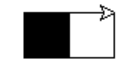
1. **(10%)** Add the function **draw\_pixel(color\_string, turta)** to pixart module. This function takes color\_string (i.e. string) and a turtle object as parameters, and it draws one pixel of that particular color. Noting that this function must use the function **draw\_color\_pixel(‘red’, turta)** to draw the colored pixel.
   1. For example: **draw\_pixel(‘5’, turta)** will draw a green pixel:



1. **(15%)** Add the function **draw\_line\_from\_string(color\_string, turta)** that, given a string and a turtle object, will draw one pixel for each character in the string that corresponds to one of the colors in the table above. This function must use the fuction draw\_pixel defined in part 3.
   1. For example, given the string '01A753421' your function **draw\_line\_from\_string ('01A753421', turta)** should produce a row of pixels like the one shown in the image below and return True as all of the colors are valid.



* 1. If the string contains an invalid color, your function should stop drawing immediately and return False. For example, given the string '01Y753421' your function **draw\_line\_from\_string ('01Y753421', turta)** should produce a row of pixels like the one shown in the image below:

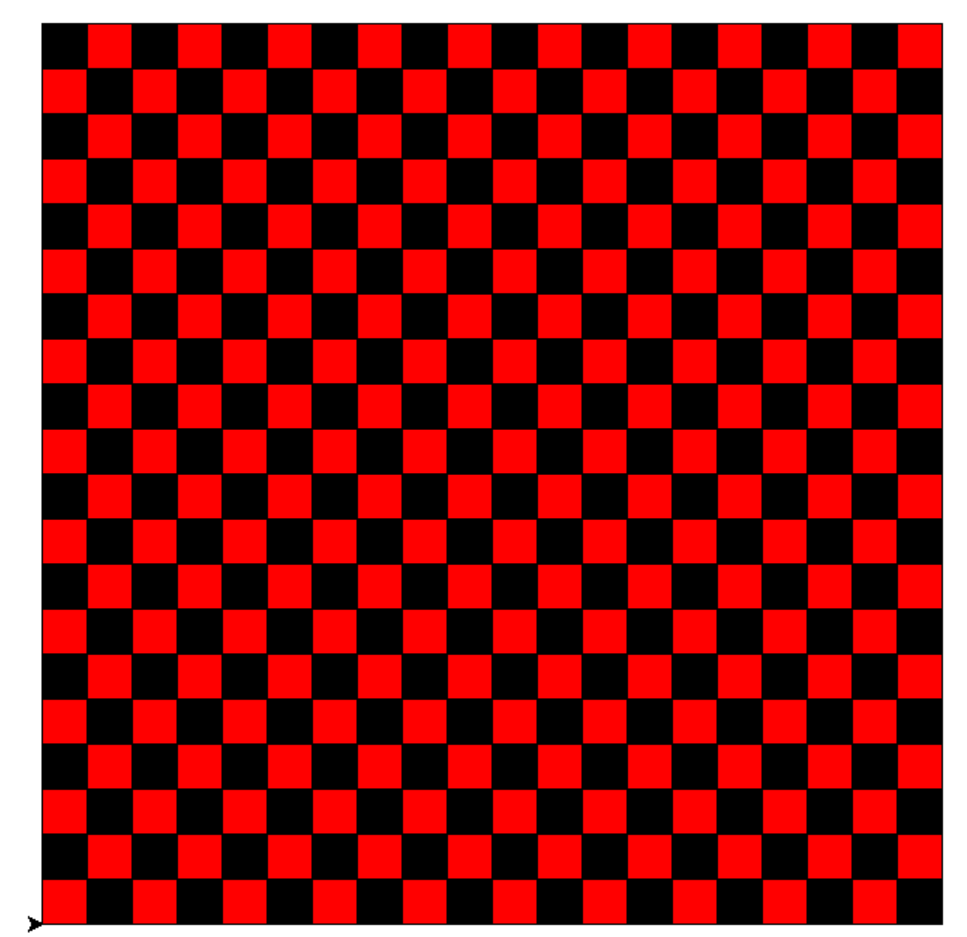


1. **(20%)** Add the **draw\_shape\_from\_string(turta)** function that takes a turtle object as parameter, and uses a loop to prompt the user to enter color strings. For each string that the user enters, this function will to draw the corresponding sequence of colored pixels by calling the function **draw\_line\_from\_string**. You should always move to the start of the next row after drawing a string. Your function should continue prompting the user until:

* an invalid color is entered, or
* the user enters an empty string.

## **Part 2 (15%)**

1. Add a function **draw\_grid(turta)** to **pixart**. This function takes a turtle object as parameter and draw a 20x20 checkerboard like the one shown in the image below. This function must use the **draw\_line\_from\_string** defined previously.



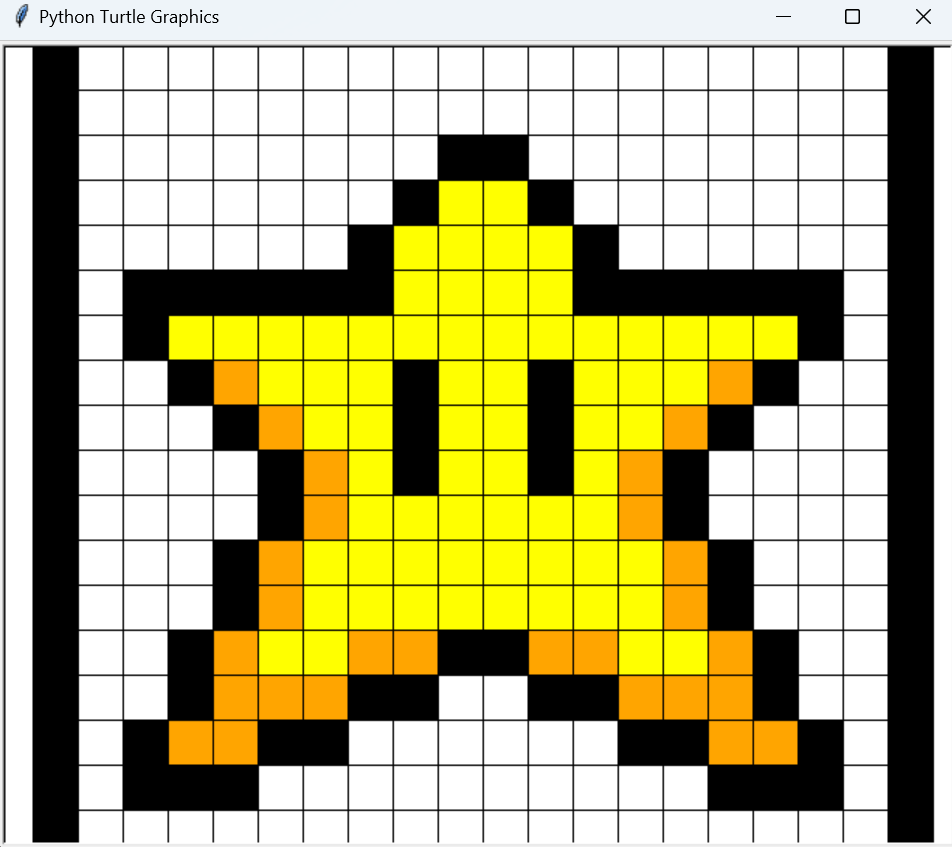
## **Part 3 (15%)**

Add a new function to pixart that use the function **draw\_shape\_from\_file(turta)** to prompt the user for the name of a text file, read its content and generate the image.

For example: by executing the draw\_shape\_from\_file(turta), the function asks the user to enter the user to enter the path of .txt file:

**Enter the path of the file that you want to read its content:** drawing01.txt

Let say the user enters the drawing01.txt, then the function draws the following image:



Test your code for the four provided files: drawing01.txt, drawing02.txt, drawing03.txt, and drawing04.txt

|  |  |
| --- | --- |
| drawing01.txt | drawing02.txt |
|  | *This image is larger than 20x20 and will not be centered on the turtle’s canvas.* |
|  |  |
| drawing03.txt | drawing04.txt |
|  |  |

## **Part 4 (10%)**

Please create a module named drawing.py in which you will import the relevant functions from the **pixart.py** module. Additionally, include a **main** function in **drawing.py**, and implement the Python construct **if \_\_name\_\_ == "\_\_main\_\_":** to control the execution of code within the script. This construct differentiates between when a Python file is run directly and when it is imported as a module in another script. Consequently, **pixart.py** should contain only the function definitions.

## **Submission Instructions**

1. Include the appropriate internal-documentation (i.e. comments & docstring) for each function you implemented in this work.
2. Include the docstring for your modules.
3. Upload **ALL the file (including the provided text files)** to the MyCourses Assignment box as (group\_number\_activity02.zip) (only one team-member needs to submit to MyCourses).
4. You are not allowed to use any concepts not explained in class this term. Otherwise, -10% will be applied.
5. For late submission, 10% of the work’s assigned mark will be deducted, for every calendar day without exception, for a maximum of 2 days, after which it will be graded as zero.
6. Be sure that you have pushed **ALL the file (including the provided text files)** to GitHubClassroom. Be sure that you have pushed **ALL the file** to your GitHub classroom repository (**for each group)**. Otherwise -5% will be applied. DON’T WAIT UNTIL THE LAST DAY TO CHECK YOUR GITHUB AND FIX YOUR PROBLEMS.
7. You are required to provide a comprehensive explanation of all aspects of your activity, including the components developed by your teammate.
8. Please note that you need to present your code with your group and answer all the questions even though that part is not your responsibility. (10% will be deducted for not presenting)
9. You can't use the concepts we did not cover in the class; otherwise, 10% will be deducted.