Project 1 - A Turtle Graphics scene

What you will demonstrate

1. Ability to break a problem down into smaller problems
2. Ability to represent solutions as combinations of blocks, conditionals, loops and functions
3. Ability to write Python code to implement your solutions
4. Ability to use good programming style

What you will produce

For this project, you will use Turtles to create a scene by combining shapes. The scene will have some background element(s) and some recognizable object(s) in the foreground.

You'll design a scene and then "solve the problem" of how to draw it with at least two Turtles. You will then write a main function in Python (named **main**) to call on those other functions to draw the background and your shapes.

Your scene must include text indicating the name of your scene.

Part 1 is Design; Part 2 is Plan; Part 3 is Coding (and revision of design and plan as needed.)

Design

On an 8.5 x 11 sheet of paper, draw out your scene. Remember that it should have a background with some repeated shape (snowflakes, stars, etc.) and a primary foreground object that is created from simple shapes (e.g, rocket, boat, house, building, etc.) Keep it simple for the project.

Take a picture or scan of the sheet and upload to Gradescope.

This is taking care of Step 2 of the process at https://docs.google.com/document/d/1ItTRvpKJqIucHIwRj5wdvMWpwXHaQgTYs6V_zUh3Q2I/edit#heading=h.qdnl7d51evn2

Plan

On another sheet of paper, plan out your code. This is Step 4 of the process at https://docs.google.com/document/d/1ItTRvpKJqIucHIwRj5wdvMWpwXHaQgTYs6V_zUh3Q2I/edit#heading=h.qdnl7d51evn2 .

(What about Step 3? For *this* project, Step 3 - "describe the solution in programming terms" - is basically "write turtle graphics commands to generate the drawing")

In this step, you want to think about the functions and variables you'll need and how you'll organize your code. This doesn't have to be formal, but time spent organizing your thoughts and planning on paper will mean less time spent coding (which is Step 5 of the process.)

Code Requirements

Your code must contain

A. the following functions to draw shapes, all of which must accept (and use) parameters for **turtle (**name the parameter **t**)**, x, y, heading, pensize, pen_color,** and **fill_color,** along with the parameters specified below:

   a. **draw_rectangle** (**length** and **width**);

   b. **draw_triangle (side_length);**

   c. **draw_circle (radius, extent, steps);**
   see https://docs.python.org/3/library/turtle.html#turtle.circle
   (the use of the **steps** parameter will let you draw regular regular polygons)

   d. **draw_shape** (up to you)
   this function should draw some other shape you'll use in your drawing (e.g., trapezoid, bird, cloud, wave, parallelogram)

   For all these functions, if the **fill_color** argument is "none", the shape should be drawn unfilled.

B. a function named **draw_object** that combines multiple calls to the above functions to make a composite object (e.g., student, castle, boat, car, spaceship, elephant); this must involve at least 6 calls to your other functions, meaning that it must be comprised of at least 6 shapes.

C. a function named **draw_background** that draws some background elements (e.g., mountains, trees, stars);

D. **draw_background** includes a **for** loop and the use of random numbers (e.g., a loop that draws some number of stars at random locations);

E. at least one conditional expression;

F. the use of a list (e.g., a list of colors, a list of turtles, a list of words to write);

G. a **write()** statement to write the title of your scene on the image (in a readable font-size and color);

H. a print statement to output your name and UIN to the console

You can certainly include other functions and language elements as needed.

Project Collaboration

You are allowed to receive help on this project from other students who are also taking CS111. Each student must still complete and submit his/her own project. You will be required to include a **Collaboration Statement** somewhere on your project if you receive help. This statement should list each helping student's name in a comment in the header of your Python file that includes your main() function. Take a look at the example in the sample header below.

If you didn't get help from any student, remove those lines from the header.

Programming Style

This project is to be written using good programming style. Elements of good programming style include:

- Meaningful Variable Names
- Proper Indentation of Code
- Blank Lines between Code Sections
- Use of Functions
- In-Line Commenting
- Header Comment for the File
- Docstring for each function

Submission

Make sure your submitted version sets the tracer to 0 and calls update at the end. While it can be useful to watch the drawing happen as *you* are developing, *we* don't need to spend the time watching the drawing happen.

After you get your project working in Repl.it, download the zip file, and upload it to Gradescope.

```
#######################################################
# Project: Project 1
```

```
# Student Name:  <your name: last, first>
# UIN: <your UIN>
# URL: <address to your Repl.it for the project>

# either list the students who provided help or remove the comment lines below

For this project, I received help from the following members of CS111.

Student1name, netID 12345678: help with background loop

Student2name, netID 87654321: help with turtle heading and function parameters


##########################################################

<imports>

<function definitions>

def main():

  <variable declarations> (turtles, screen)>

  <initial settings for variables>

  <lots of code, including calls to other functions>


main()




## information for scorers

## on what line number is the required for loop?
## <line number>

## on what line number is the required use of a random number?
## <line number>

## on what line number is the required use of a conditional statement?
## <line number>
```

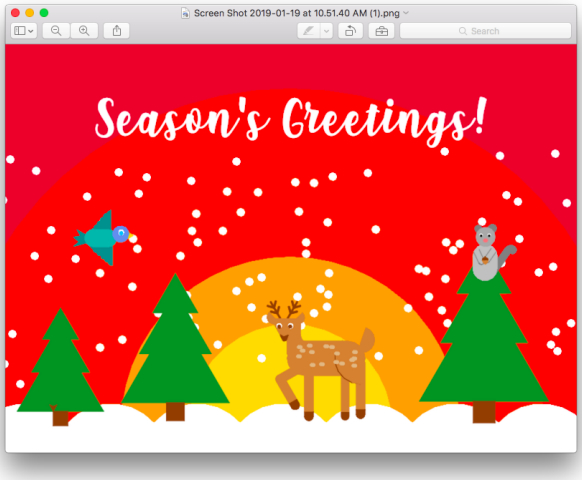
```
## on what line number is the required use of a list?
## <line number>
```

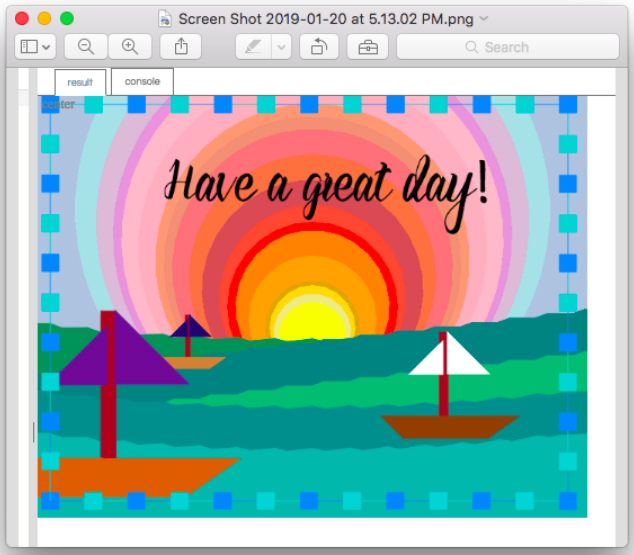
Rubric

	Missing	Partial	Complete
Background function			
Includes **draw_background** function	0	1	2
draw_background uses a loop	0		1
draw_background uses randomization	0		1
Shape functions			
Includes **draw_rectangle** function, as specified	0	1	2
Includes **draw_triangle** function, as specified	0	1	2
Includes **draw_circle** function, as specified	0	1	2
Includes **draw_shape** function, as specified	0	1	2
Includes **draw_object** function, as specified	0	1	2
Other code requirements			
Includes a **main** method (function) which runs the whole program	0		1
Uses at least two Turtles to draw the scene.	0		1
Uses a conditional statement	0		1
Uses a list	0		1
Includes specified text (a title) on drawing	0		1
Includes print statement with Name and UIN	0		1
For final submission, sets tracer to 0	0		1
Programming Style			
Variable and function names are meaningful	0	1	2
Variable and function names follow Python standards	0		1

Includes meaningful in-line commenting	0		1
Includes header comment for the file	0		1
Includes docstring for each function	0	1	2
Total			28

Here are a couple of examples from a high school teaching colleague. Your scene does not need to be as complex, but these both demonstrate how we can combine simple shapes to make a picture. The trees and the sails are triangles, and both have used concentric circles of different colors.




Screen Shot 2019-01-20 at 5.13.02 PM.png
Search
result
console
Have a great day!