Published

Week 8: Modules 4 of 5

8 Week 8 Assignment

Computer SCIENCE

You will complete one programming task in this assignment based on the new things you learned in this unit. Make sure to use proper naming conventions for your project and name the files for each task as specified below.

Objectives

You will complete three tasks to give you experience with

1. Function design
2. Functions
3. Positional, keyword, default parameters
4. Modules

Task 1

Starter: **assn4-task3-starter.py** (https://usu.instructure.com/courses/681553/files/81729303/download?download_frd=1)

Video: **Patterns.mp4** (https://usu.instructure.com/courses/681553/files/81729306/download?download_frd=1)

Requirements

You will be submitting three files: task1.py (rename the starter code file), pattern.py, and plan1.txt. Use the assn4-task3-starter.py file to get started. Ignore the name of the starter code, it's correct, but named for the 7 week version of the class. You should add to this file in the areas indicated, but not modify any existing code. This should give insight to the function definitions you need to write. You need to watch the video demo get understand the full instructions.

You are going to create a program that draws cool patterns. The user will have three modes to choose from:

1. Rectangle Pattern - A circular pattern created by drawing multiple rectangles
2. Circle Pattern - A circular pattern created by drawing multiple circles
3. Super Pattern - A random selection of some number of Rectangle and Circle Patterns

Coding Notes

1. task1.py - Use the starter file given and only fill in the appropriate areas. Do not modify any code that is there already, only add to it. This file will help you with defining the functions in pattern.py.
2. pattern.py - You must define the following functions
    - reset() - Erase all of the patterns and start over
    - setup()
        - Configure turtle to draw quickly
        - Configure turtle to have a window of 1000 x 800
    - drawRectanglePattern()
        - Use appropriate parameters
        - See additional information below
        - It should call drawRectangle()
    - drawRectangle()
        - Use appropriate parameters
        - Should draw a SINGLE rectangle
    - drawCirclePattern()
        - Use appropriate parameters
        - See additional information below
    - drawSuperPattern()
        - Use appropriate parameters
        - Randomly draw Rectangle and Circle patterns. Each pattern should based on random values.
        - Use reasonable random values (some can be negative) so patterns are drawn on the screen
    - setRandomColor()
        - Do not use any parameters
        - Set turtle to draw in a random color
        - Use at least 4 colors
    - done()
        - Called when user quits
        - Keeps the turtle window open

Make sure to read the rubric to see the additional requirements.

Drawing Rectangle Pattern

The Rectangle Pattern has 6 parts to it

1. Center position - This is the x, y center point of the circular pattern that is drawn
2. Offset - This is the distance from the center position to the starting corner of each rectangle. It can be a positive or negative number
3. Height - The height of a rectangle
4. Width - The width of a rectangle
5. Count - The number of rectangles to draw within the 360 degree pattern.
6. Rotation - The number of degrees each rectangle is rotated in relation to the line from the Center Position to the starting corner of the rectangle

Each individual rectangle should be a new random color. You should use at least 4 different colors.

Drawing Circle Pattern

The Circle Pattern has 4 parts to it

1. Center position - This is the x, y center point of the circular pattern that is drawn
2. Offset - This is the distance from the center position to starting drawing point of each circle. It can be a positive or negative number. Note that the center point of each circle should be 'radius + offset' distance from the Center Position of the pattern.
3. Radius - The radius of the circle
4. Count - The number of circles to draw within the 360 degree pattern.

Each individual circle should be a new random color. You should use at least 4 different colors.

Common Errors

- The rotation for the rectangle pattern is a common error. The rotation happens at the corner where drawing starts for the rectangle. Make sure to watch the video for a detailed explanation.
- The offset for the circle is frequently missed. Make sure you review what that means.

Points 100

Submitting a file upload

Due	For	Available from	Until
Oct 29, 2021	Everyone	-	-

Unit 4 (2)

Criteria	Ratings		Pts
Task 1: Software Development Plan	**3 pts Full Marks**	**0 pts No Marks**	3 pts
Task 1: Get input Get input properly from users. You may assume the users give valid input	**3 pts Full Marks**	**0 pts No Marks**	3 pts
Task 1: main(), use starter code main() is the only function defined in task1.py, which also calls main() Use the starter code properly	**3 pts Full Marks**	**0 pts No Marks**	3 pts
Task 1: pattern.py, functions Create a module called pattern.py. This contains all of the functions associated with drawing the patterns	**5 pts Full Marks**	**0 pts No Marks**	5 pts
Task 1: reset() Proper implementation and usage of reset()	**5 pts Full Marks**	**0 pts No Marks**	5 pts
Task 1: setup() Proper implementation and usage of setup()	**5 pts Full Marks**	**0 pts No Marks**	5 pts
Task 1: drawRectanglePattern() Proper implementation and usage of drawRectanglePattern() location, offset, width, height, count, and rotation are all used and drawn properly	**20 pts Full Marks**	**0 pts No Marks**	20 pts
Task 1: drawRectangle() Proper implementation and usage of drawRectangle()	**10 pts Full Marks**	**0 pts No Marks**	10 pts
Task 1: drawCirclePattern() Proper implementation and usage of drawCirclePattern() as defined above location, offset, radius, and count are all used and drawn properly	**20 pts Full Marks**	**0 pts No Marks**	20 pts

Criteria	Ratings		Pts
Task 1: drawSuperPattern() Proper implementation and usage of drawSuperPattern()	**10 pts** **Full Marks**	**0 pts** **No Marks**	10 pts
Task 1: setRandomColor() Proper implementation and usage of setRandomColor(). There should be at least four colors that can be randomly selected.	**8 pts** **Full Marks**	**0 pts** **No Marks**	8 pts
Task 1: done() Proper implementation and usage of done()	**5 pts** **Full Marks**	**0 pts** **No Marks**	5 pts
Task 1: Follow proper coding conventions	**3 pts** **Full Marks**	**0 pts** **No Marks**	3 pts
			Total Points: 100