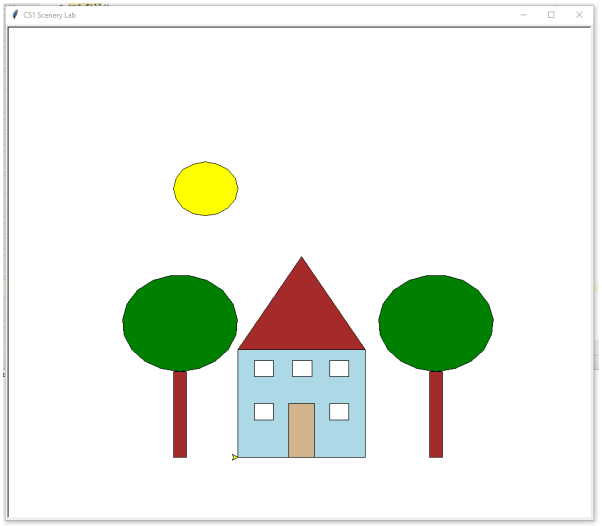
Computer Science I
Turtle Scenery

CSCI-141
Lab 1

08/22/2021

1 Problem

Using Python's turtle graphics module, design a program that draws several different shapes using re-usable functions. The goal of this lab is to become familiar with various Python Turtle commands and use them to create a module (file) with basic Python program structure.



2 PSS: Problem-solving Session (15%)

Do not use a computer for the PSS; use paper or the whiteboard. Work in a team of students as determined by the instructor. Each team should complete the following activities.

1. Write a Python function to draw a simple square with side length of 100 units. The turtle must end in the same location and orientation it started in.
2. Write a Python function to draw a simple triangle with side length of 100 units. The turtle must end in the same location and orientation it started in.
3. Write Python code to draw a shape of a house, a triangle on top of a square. You must *reuse* the functions defined in the first two tasks.
4. Design a tree you would like the turtle to draw; no need to write code. You have freedom to make this tree as simple or complex as desired. Explain how you would define the tree drawing function using Turtle. What shapes would you use? Where does the turtle start and in what state does the turtle finish?

Each team shall number the items and deliver results at the end of problem-solving.

There will be another handout for the In-lab session.

3 In-lab Session (10%)

The Instructor and SLI can assist you in starting to use the CS department lab systems and learn these things:

- How to log in to CS lab systems;
- How to start a browser and look up Python 3 documentation;
- How to change your password with 'knockknock';
- How to open a terminal window for starting PyCharm, Idle or Python;
- Where and how to save files;
- How to contact the SLI to ask questions via email; and
- How to get help at the system administrators' office.

The In-lab goals are to learn the CS systems as a backup in case your system fails and to start implementation of the lab.

4 Implementation (75%)

Each student must individually implement and submit their own solution to the problem as a Python program named `scenery.py`.

When run, the program must pause after drawing the figure and wait for the user to close the turtle window. Wait for this acknowledgement by inserting a `turtle.done()` statement.

You need to research and use various turtle commands so that you can make a Python program that draws a basic scenery drawing.

The scenery drawing must include the following items:

- A house with:
  - A facade (the front of the house);
  - Several windows;
  - One door; and
  - One roof.
- Two trees.
- A Sun.

You must fill the shapes with colors. Research turtle color fill to assist with this portion of the task.

You may follow the example picture, or make your own design.

You must make reusable functions to draw each of the scene elements. It is poor practice to draw the entire scene in one long function.

Note*: You must retest code when making even the smallest of changes.*

4.1 Hints and Tips

- Turtle Doc Website: `https://docs.python.org/3.3/library/turtle.html`
- Turtle can use numbers or words for choosing colors.
- Design functions for each element and test them first. Then draw the larger picture using these functions.
- `https://www.cs.rit.edu/~csci141/resources.html` is the location of resource page for the course. There is also a link found near the top of the main course page.

4.2 Grading

- 40%: Scene-drawing Functionality
  - 20%: Draws the scene with the required content.
  - 10%: Fills shapes with multiple, appropriate colors.
  - 5%: Drawing fits within the canvas window.
  - 5%: Displays the drawing for the user to view and waits for the user before exiting the program.
- 20%: Design and implementation that draws the scene elements. Drawing the entire scene in one long function or with no function at all is bad design and subject to a 50% deduction.
  - 6% A function that draws the house as a complete unit
  - 8% Functions that draw house components (e.g. door, windows)
  - 4% A function to draw a single tree as a complete unit
  - 2% A function to draw the Sun in the sky
- 10%: Style and Documentation
  - 5%: Follows the code 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 functions to draw another scene.
  - 5%: A *docstring* at the top of the source file module includes at least the author's full name. This is a *file documentation block* that should use *triple-quote* syntax as shown in the course examples.
- 5%: Submission used zip properly and uploaded correctly. (See below.)

4.3 Submission

Create a zip file named `lab01.zip` containing the `scenery.py` file. Submit the zip file to the **myCourses dropbox** for this assignment.

Use *only the zip program*. Do not use rar, 7zip or other compression tools.

Here is a command line example: `zip lab01.zip scenery.py`

In graphical file browsers, you also might be able to select a file, right-click, and choose a compress option, but the compression tool is *not necessarily zip*.