

CS 125

Lab 3

Write a Python program that draws a house. You're going to first write some functions that will be useful.

For this lab, we'll use IDLE. Start it up and then do File/New File from the menu to open a file editing window. Save your file into your cs125 directory as house.py.

Your file should start with

```
import turtle
```

You will need to create a turtle with

```
t = turtle.Turtle()
```

at the bottom of the file (after the functions).

As you write each of the functions below, you should test them by doing Run/Run Module from the Menu on the editing window. Test them one at a time, rather than trying to write them all and then test. To test a function, go back to the Python shell window and call the function.

1. Write the following function called `move` that will move the turtle without drawing a line as it's moving.

```
def move(t : turtle.Turtle, x : float, y : float):
    """
    Move the turtle without drawing.

    Parameters:
        t -- the turtle
        x -- the x coordinate of the new position
        y -- the y coordinate of the new position
    """
    t.penup()
    t.goto(x, y)
    t.pendown()
```

Save your file. Then test the function by going back to the Python shell window and call the function like this:

```
move(t, 100, 200)
```

You should see the turtle move.

2. Write a function called `draw_rectangle` that will draw a rectangle. The current position should be at the center of the rectangle. The parameters will be the turtle and the width and height of the rectangle. Put it after the `move` function but before the line that creates the turtle. Example:

```
draw_rectangle(t, 200, 400)
```

will draw a rectangle of width 200 and height 400. There are several ways to do this. Here's a hint. Start by moving to the upper left hand corner. You can do this with

```
move(t, t.xcor() + width / 2, t.ycor() + height / 2)
```

because `t.xcor()` and `t.ycor()` give you the x and y coordinates of the turtle's current position. Then you can use `t.setheading(180)` to make the turtle turn to the west. After that, use `forward` and `left` to go around the rectangle.

3. Write a function called `fill_rectangle` that fills a rectangle with color. The parameters will be the turtle, the width and height, and the color. Example:

```
fill_rectangle(t, 200, 400, "blue")
```

will draw a blue rectangle of width 200 and height 400 filled with the color blue. Use the `draw_rectangle` function rather than repeating the code from `draw_rectangle`. You'll need `color`, `begin_fill`, and `end_fill`.

4. Write a function called `draw_triangle` that draws a triangle. The parameters will be the turtle, and the x and y coordinates of the vertices of the triangle. For example,

```
draw_triangle(t, 50, 100, 75, 150, 0, 125)
```

will draw a triangle that connects the points (50, 100), (75, 150) and (0, 125). Hint: use `move` to get the first point. Then use `goto` to draw the lines.

5. Write a function called `fill_triangle` that uses `draw_triangle` to fill a triangle with a color. You'll need `begin_fill`, and `end_fill`.
6. Write a function called `draw_house` that uses the other functions to draw a house with a triangle roof, at least one door, and at least two windows. Use colors to make it pretty. Its only parameter will be the turtle. Example:

```
draw_house(t)
```

should draw the entire house.

Because the builtin turtle graphics methods are methods belonging to the turtle, you have to call them like this:

```
t.forward(100)
```

But your functions are functions rather than methods. You call them like this:

```
draw_rectangle(t, 50 100)
```

Your main program (the last thing in the file) should be:

```
draw_house(t)
t.hideturtle()
```

```
turtle.getcanvas().postscript(file="house.ps")
```

Then running the program will draw the entire house. The last instruction writes out the picture as a postscript file. Turnin will then run a command called ps2pdf to convert it to a pdf. Finally turnin will collect house.py and house.pdf.

When you ssh to groot to turn this one in, you need to do it like this:

```
ssh -XY groot
```

Here is a link to the turtle graphics documentation:

<https://docs.python.org/3/library/turtle.html>