

Lab 11: Turtle Power

Due November 17, 2015

An Improved Turtle

Add the following methods to the `Turtle` class we designed and implemented in class. You will have to add new attributes as well.

1. `penUp()`: turn trail drawing off.
2. `penDown()`: turn trail drawing on.
3. `isPenDown()`: return `True` if currently drawing.
4. `moveBy(t, r)`: turn the turtle left by `r` degrees, move forward by `t` units, similar to `Myro.move(t,r)`. Then you can drive the turtle with the gamepad like so:

```
from Myro import *
from Graphics import *

win = Window(500, 500)
turtle = Turtle(win)
for time in timer(60):
    r, t = getGamepadNow("axis")
    turtle.moveBy(5*t, 2*r)
    wait(.1)
```

5. `moveTo(x, y)`: move the turtle to the location `x, y` in the window (drawing a trail if the pen is down).
6. `dropCircle(radius)`: draw a `Graphics Circle` at the turtle's current position.
7. `dropRectangle(width, height)`: draw a `Graphics Rectangle` at the turtle's current position.
8. `setPenColor(cname)`: change the trail color to `cname`.
9. `getPenColor()`: return current trail color.
10. `setLineWidth(width)`: change the trail line width (using the `setWidth()` method for shapes).

Turtle Testing

Create a test function `turtleTest()` that uses all the methods you create.

Class Diagram

Create a class diagram for the final Turtle class. A class diagram describes the attributes and methods of the class as well as type information. For example, below is a class diagram for a toy Duck class.

```
class Duck:

    ducks = 0

    def __init__(self, name):
        self.name = name
        self.wings = 2
        self.body = Scribbler()
        Duck.ducks = Duck.ducks + 1

    def quack(self, voice):
        setVoice(voice)
        speak(self.name + " says quack")

    def getName(self):
        return self.name

    def waddle(self, times):
        for i in range(times):
            self.body.move(1, -0.5)
            self.body.move(1, 0.5)
            self.body.stop()
```

Class Name	Duck
Class Attributes	<u>ducks: int</u>
Object Attributes	name: string wings: int body: Scribbler
Methods	«constructor»Duck(name: string) quack(voice: string) getName(): string waddle(times: int)

Learning Objectives

- Work with Classes
- Create Methods
- Create Attributes

Deliverables

Submit an electronic copy of your lab using moodle and bring a hardcopy of both files to class. Your program should have your name, email, assignment description, the date, and collaboration statement at the top of the file as a comment. Your submission should be a zip file that expands to a folder with a single file.

cmssc143-lab11-LASTNAME-FIRSTNAME

lab11.py
lab11.pdf