CMSC 143: Object-Oriented Programming with Robots

Lab 10: Programming with Class
Due December 3, 2015

In this lab, we will improve upon the small *Frogger* game we started in lecture. In the first part of the lab, you are asked to create class diagrams of the existing classes in the game. 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)

In the second part you should improve the game by designing a new class for a new object(s) in the game. For example, you could create a timer for the game, a treasure object that rewards the player when collected, a speed booster, or create a new type of enemy. Feel free to change the current Player and Car class.

First write the class diagram for the new class before writing any python code.

Learning Objectives

○ Create Class Diagrams ○ Design Classes ○ Implement Classes ○ Create a Game

Deliverables

Submit an electronic copy of your lab using moodle. 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.

```
cmsc143-lab10-LASTNAME-FIRSTNAME/
   lab10.pdf-- Your class diagrams
   lab10.py -- Your program
```