Lab 11: Car Project

CSE/IT 107

NMT Department of Computer Science and Engineering

"When I see a bird that walks like a duck and swims like a duck and quacks like a duck, I call that bird a duck."

— James Whitcomb Riley, Duck Enthusiast

"Imagination is more important than knowledge."

— Albert Einstein

"The limits of my language mean the limits of my world."

— Ludwig Wittgenstein

---

1 Car Project

Cars Are Hard

Warning: The following exercise was designed with little regard for how cars and physics actually work in the real world. Sorry.

Exercise 1.1 (car.py, tire.py, engine.py, race.py).
For this project, you will be submitting four files:

`car.py` will contain the definition of `class Car`. A Car will have an `Engine` and a `Tire` (representing all 4) as attributes. Additionally, each car should have an attribute representing its name. Additionally, a Car should have a method `travel_time(self, time)` that returns the distance that the Car travels in the time given as well as a method `travel_dist(self, dist)` that returns the time it takes for the Car to travel the given distance. Both methods should assume that the car starts at a standstill and accelerates throughout the travel time. The results should take into account the attributes of the Car's Tire and Engine.

`tire.py` will contain the definition of `class Tire`. Each Tire will have two attributes: traction and radius. The traction of the wheel indicates the maximum ideal acceleration the wheel can provide. If the car's acceleration would be above this number, then it is instead the average of this number and the ideal acceleration. The radius of the tire determines how far it causes the car to travel per rotation. This is important because Engine strength is measured in rotations.

Traction and radius are both measured in meters.

`engine.py` will contain the definition of `class Engine`. Each Engine will have two attributes: max speed and acceleration. Max speed is in terms of tire rotations per second. Acceleration is a value between 0 and 1 that indicates how much closer the current speed gets to the max speed each second. Each second, the new speed is determined by the following formula:

$$newSpeed = oldSpeed + acceleration \cdot (maxSpeed - oldSpeed)$$

Since engine speed is measured in rotations per second, the actual acceleration of the car must take into account the engine's speed, the tire's traction, and the tire's radius to determine how far it travels each second. Because of the tire's traction, it is possible that the car is not experiencing the full speed that might be indicated by $engineSpeed \cdot tireCircumference$. Thus, the car's speed cannot be purely calculated from the engine's speed each second.

`race.py` will contain your `main()` function and will manipulate the classes from the other files. The program should call the constructors for each of the classes described above in order to create several complete cars, then print out data comparing the performance of the cars. The data displayed should include every attribute about every car, followed by a table comparing the performance of each car in a trial comparing either how far each car travels in a given time or how long each car takes to travel a given distance. These should use the relevant methods described above. You should ask the user whether they want a table of distances gone in fixed times or times taken to go fixed distances. You should be able to compare 1, 2, 3, or 4 cars in one table. Figure 1.1 shows the data you need to show when comparing distances given a fixed time. You should be able to make a similar table the other way around.

Car Name	Traction	Tire Radius	Max Speed	Acceleration
Rabbit	0	0.5	1000	0.1
Hare	100	0.1	500	1.0

Table 1: Data to print out about cars.

Time	Rabbit	Hare
0	0.00	0.00
1	0.00	0.00
2	157.08	207.08
3	534.07	517.70
4	1148.25	831.86
5	1995.54	1146.02
6	3062.44	1460.18
7	4331.90	1774.34
8	5786.12	2088.50
9	7407.85	2402.65
10	9180.95	2716.81

Table 2: Data to display about how far cars go in given times.

When simulating the cars, you should use a time delta of 1 second. This means that you simulate time in chunks of one second at a time. Thus, a car should be considered to have the same speed throughout the entirety of each second. This can result in some inaccurate results (such as each car always traveling a distance of 0 for the first second), but is fine for our purposes. As an example of how a car's travel might work, see Table 1.1, which is a breakdown of the stats of a car at each second of a 10 second trip. The car's tires have a radius of $\frac{1}{\pi}$ (giving a circumference of 2) and a traction of 10. The car's engine has a max speed of 100 and an acceleration of 0.5.

Time (s)	Distance (m)	Car Speed ($\frac{m}{s}$)	Ideal Car Speed ($\frac{m}{s}$)	Engine Speed ($\frac{rot}{s}$)
0	0.00	0.00	0.00	0.00
1	0.00	55.00	100.00	50.00
2	55.00	107.50	150.00	75.00
3	162.50	146.25	175.00	87.50
4	308.75	171.88	187.50	93.75
5	480.63	187.81	193.75	96.88
6	668.44	196.88	196.88	98.44
7	865.31	198.44	198.44	99.22
8	1063.75	199.22	199.22	99.61
9	1262.97	199.61	199.61	99.80
10	1462.58	199.80	199.80	99.90

Table 3: Example of a Car's Performance

It should be noted that the Distance column of Table 1.1 represents the return value of Car's `distance_time` method when given the corresponding Time column as an argument.

2 Submitting

You should submit your code as a tarball. It should contain all files used in the exercises for this lab. The submitted file should be named

```
cse107_firstname_lastname_lab11.tar.gz
```

Upload your tarball to Canvas.

List of Files to Submit