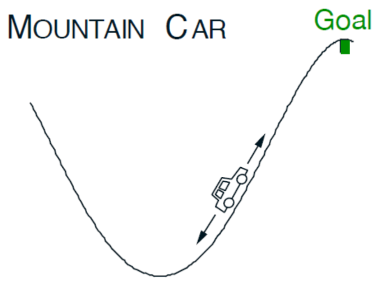
Homework 8: Policy Gradient
Due: Wednesday, December 2nd 11:59 pm

The purpose of this project is to study different properties of Policy Gradient algorithms with Function Approximation.

Task Description
Consider the task of driving an underpowered car up a steep mountain road, as suggested by the diagram in the upper left of the following figure. The difficulty is that gravity is stronger than the car's engine, and even at full throttle the car cannot accelerate up the steep slope. The only solution is to first move away from the goal and up the opposite slope on the left. Then by applying full throttle the car can build up enough inertia to carry it up the steep slope even though it is slowing down the whole way.



This is a continuous control task where things have to get worse in a sense (farther from the goal) before they can get better. The reward in this problem is -1 on all time steps until the car moves past its goal position at the top of the mountain, which ends the episode. There are three possible actions: full throttle forward (+1), full throttle reverse (-1), and zero throttle (0). The car moves according to a simplified physics. Its position x_t and velocity $\dot{x}_t$ are updated by

$$x_{t+1} = bound[x_t + \dot{x}_{t+1}]$$

$$\dot{x}_{t+1} = bound[\dot{x}_t + 0.001A_t - 0.0025\cos(3x_t)]$$

where the *bound* operation enforces $-1.2 \leq x_{t+1} \leq 0.5$ and $-0.07 \leq \dot{x}_{t+1} \leq 0.07$. In addition, when x_{t+1} reached the left bound, $\dot{x}_{t+1}$ was reset to zero. When it reached the right bound, the goal was reached and the episode was terminated. Each episode starts from a random position $x_t \in [-0.6, -0.4)$ and zero velocity.

You have been given a simple implementation of the Mountain Car task. You can use your implementation of the function approximation from Homework 6, or implement a new one.

Part I
Select and implement of the following algorithms (+10 for implementing both):

1. REINFORCE with Baseline (p. 330)
2. Actor-Critic with Eligibility Traces (p. 332)

Part II
Use the algorithm to learn the Mountain Car task. Tune the step-size parameter (α), the Function Approximation order, the discount factor (γ), and the λ–value. **Note:** you can consider the problem undiscounted.

1. Plot step-per-episode (in log scale) vs. number of episodes. This plot should be averaged over 50-100 runs.

2. Plot total reward on episode vs. number of episodes. This plot should be averaged over 50-100 runs.
3. Show an animation of the task for the final episode.

Evaluation: we will grade your submission according to the following table:

Item	COMP4600	COMP5300
Part I	40	40
Part II	60	60

Note 1: The parts marked with * are optional for COMP4600 (undergraduates) but mandatory for COMP5300 (graduates). There are no optional parts in this homework.

Note 2: We do not accept code in any programming language other than Python 3 (do not use Python 2).

Note 3: For the implementation of the algorithms, you are not allowed to rely on any python library other than `numpy` and `matplotlib` (for plotting), and `pygame`.

Note 4: All the code, plots, explanations should be included in a single Jupyter Notebook (`.ipynb`) file. Include your name as part of the filename and submit through Blackboard.

Submission: By 11:59pm on Wednesday, December 2nd 2020, submit your `student_name.ipynb` file on Blackboard. Make sure everything is entirely contained within this file and it runs without any error.