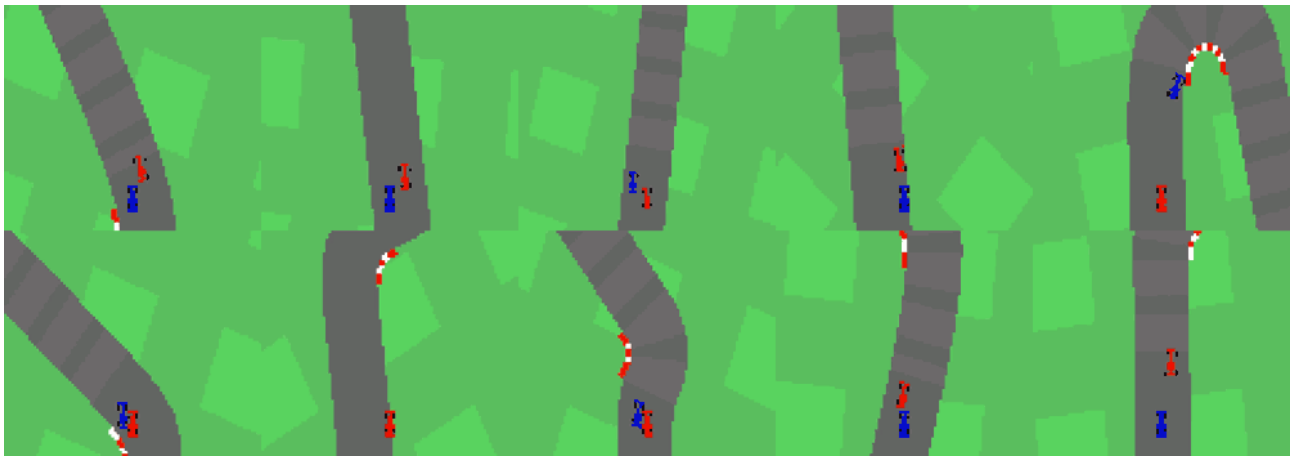
# Deep Reinforcement Learning

## Homework #3

**Due: 2024/4/23 (Tue.) 23:59**

### Problem Description



1. Continuous Action Space Algorithm ("**<Student_ID>_hw3_<train|test>.py**")
   - This environment is a simple continuous control task for a single agent. The state consists of 96x96x3 pixels representing the agent's surroundings. The agent receives a constant penalty of **-0.1 every timestep**, and a reward of **+1000/num_tiles for each unique tile it visits**. The motivation behind this reward structure is to provide a sufficiently dense reward signal for the agent to learn the basic driving skills, while encouraging it to explore and visit as many tiles as possible.
   - The starting point is random position of the track.
   - Installation of the project can be referred at **THIS REPOSITORY**.
2. Purpose of this Homework
   - In this assignment, you should implement algorithms handling continuous action space mentioned from the class. You are allowed to use some techniques, such as stack frames, skip frames, PER, HER, to help your agent.
   - During evaluation time, our baseline agent utilizes **SAC + Normal Replay Buffer**. Therefore, you need to find out your suitable strategy to beat TA's agent.
3. Detailed Rules for this Homework
   1. If your program outputs invalid moves, you lose and the game ends immediately. Please check the upper and lower bound of action space with env.action_space.high, env.action_space.low.
   2. Time limit for each move is 1 second, and the memory limit is **8 GB**. The model saved file size should be smaller than **20 MB**.

3. You are allowed to access an external file for loading your learned policy. You can read the file at the following path: "**<Student_ID>_hw3_data**". The external file should be smaller than **25 GB**.
4. You are allowed to use the following Python package:
   A. numpy, scipy, gym, pandas, tensorflow, pytorch and the packages mentioned in the environment's repo.
   B. You are allowed to use Python default installed packages. (e.g., sys, time, pickle, random, etc.)
   C. If you need to use other packages, state your reasons and post on **EECLASS**.
   D. You are not allowed to use ready made RL packages/framework for example : **stable_baselines, spinningup, tianshou,…**
5. This environment is not applicable on GitHub to render the view, we decide to directly announce the baseline score: 574. Therefore, all you need to do is to surpass the baseline.

## Program Submission

1. For each problem, please use Python to implement with a single source file.
2. Your files must be named as:
   (a) "**<Student_ID>_hw3_train.py**"
   (b) "**<Student_ID>_hw3_test.py**"
   (c) "**<Student_ID>_hw3_data**"
   (d) "**<Student_ID>_hw3_report.pdf**"
3. Your program will be ran in a GNU/Linux environment with Python 3.8.
4. 0 points will be given to Plagiarism. **NEVER SHOW YOUR CODE** to others and you must write your code by yourself. If the codes are similar to other people and you can't explain your code properly, you will be identified as plagiarism.
5. 0 points will be given if you violate the rules above.
6. If you use modularized / OOP code and want to use multiple files to keep your code structured, please email us beforehand.

## Report

1. TAs will not refer to your code when grading your report, so make sure you have taken a screenshot of the important code snippets.
2. The report filename must be "**<Student_ID>_hw3_report.pdf**" and please make sure that all characters of the filename are in lower case.

## Grading Policy

1. The project accounts for 15 points (tentative) of your total grade.
2. You must submit both your source code and report. Remember the submission rules mentioned above, or you will be punished on your grade. Late submission rules are specified in the Lecture 1 Slides.
3. Compress all your files directly (do not compress the folder containing your files) and upload to EECLASS before the deadline. (Single compressed file, Named as **<Student_ID>_hw3.zip**!!!)
4. The TA code will not be released. Your code will be tested against them after the submission deadline.

- **Random Policy Evaluation**
  - Surpass Baseline (60%)
  - Leaderboard Ranking (20%)
  - Report (Discussion) (20%)