**End Game Assignment**

**States space:**

1. 32x32 image around the car

160x160 map mask image around the car is taken. Car mask (a triangle) is overlaid on the 160x160 image to indicate the car orientation

1. +orientation: Orientation of the car wrt to target
2. -orientation: Negative orientation of car wrt to target
3. Distance\_x : Distance of car from target in the x-axis
4. Distance\_y : Distance of car from target in the y-axis

**Action space:**

Theta: Angle at which the car should turn

**Velocity:**

1. On Sand: Lesser velocity
2. On road: More velocity

**Reward policy:**

1. On road and closer to target: 1
2. On road and farther from target: -1
3. On sand and closer to target : -2
4. On sand and father from target: -4
5. At the boundary: -10

**Episode end condition:**

1. Reward threshold : Episode reward lesser than -50000 will end the episode
2. When the car reaches the boundary, episode is ended

**Things tried and difficulties faced:**

1. Approach1:

* Implemented TD3 in assignment7
* Removed sensor data as state variables
* Trained a separate CNN to provide sand densities around the car at 10 points around the car
* Training of the CNN was done separately than the TD3
* During TD3 training, CNN was used only in inference mode
* Used the 10 sand densities around the car, orientation and distance from target as input to the TD3 model
* The model worked fairly well and the car learnt to go the roads and reach the target
* Downside: Realized that this was not the expectation of the assignment

1. Approach2:

* Provide the map patch around the car as state to the TD3 model
* Merged the CNN to the TD3 actor-critic models
* Training was extremely slow
* Ported the training part to Colab which speeded up the process
* Gradient explosion problem : After 5 episodes: Added batch normalization and weight decay.
* Gradient explosion happens after 18 episodes
* Ongoing experiments : Trying to play with rewards to see if it helps the gradient explosion problem
  + Step1 : Use plain map and fix rewards to move only to the target
  + Step2: Use map image and train to move only on roads
  + Step3: Combine step2 and step3