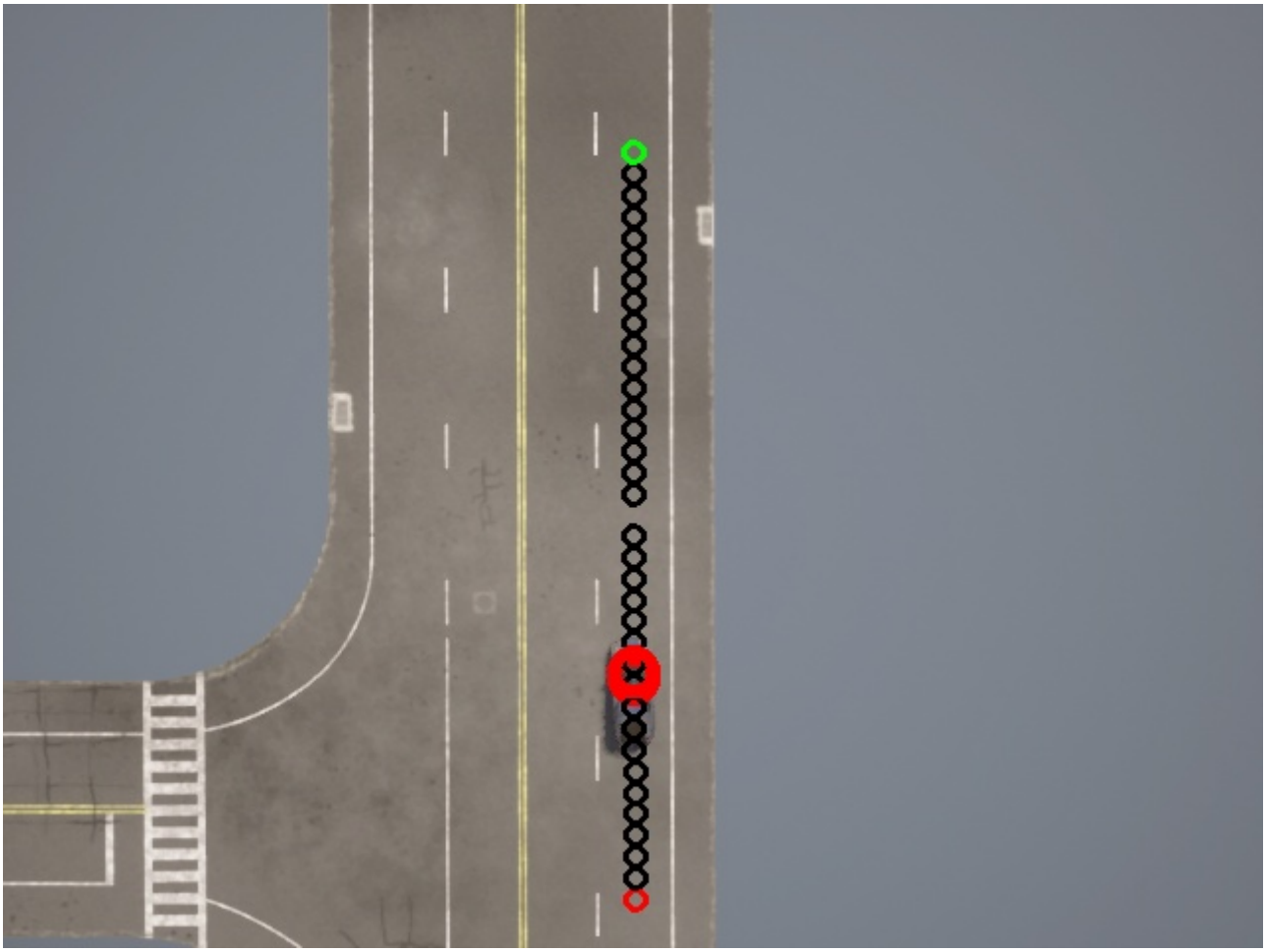

Instruction

Pre-requisite:
Requirements stated in the `README.md` file should be met

- Open conda environment terminal. And open `CARLA` server.
- Open a Jupeter Notebook and connect it to the conda environment.
- Import relevant libraries:

```
from RL_envdirectcontrol import CarENV
import numpy as np
import time
import random
from stable_baselines3 import PPO
from stable_baselines3 import DQN
import os
import stable_baselines3
```

- Create an actor, i.e. object of CARENV

```
actor = CarENV(show_local_view=True, Debugger= False , start_end_dic=
start_end_dic, action_space_type='Discrete')
```

- **Parameters:**
  - **show_local_view** (Bool): If True then images of an episode get saved in images folder. NOTE: At each new episode the images of the older episodes get over written
  - **no_render_mode** (Bool): CARLA environment goes into no render mode. No render mode is less computationally expensive.
  - **Debugger** (Bool): If True then more information is outputted at each time-step, i.e. action taken, reward, etc.
  - **start_end_dic** (Dictionary): A dictionary which has the index of start spawn points as key and a list of index for end spawn points as value of the dictionary. This wil decide the training behaviour. For example currently the model was only trained on straight route.
  - **action_space_type** ('Discrete'/'BOX'). It is set according to the action space allowed by the `stable_baselines3`. DQN doesnot allow for `BOX` type action space.

- **Callback:**
  - A callback is when the model should stop training.
  - `stable_baselines3` provides multiple callbacks. Check their documentation for more options. I used the max number of episode call back

    ```
    from stable_baselines3.common.callbacks import
    StopTrainingOnMaxEpisodes
    callback_max_episodes = StopTrainingOnMaxEpisodes(max_episodes= 50,000,
    verbose=1)
    ```

- Create Model Directory and Tensor Board log Directory

  ```
  logdir = f"trainingdir/checking_box"
  Attempt =  'triall'

  if not os.path.exists(models_dir):
      os.makedirs(models_dir)

  if not os.path.exists(logdir):
      os.makedirs(logdir)
  ```

- Create a DRL stable_baselines3 Model. Check SB3 documenetation for a comprehensive description of parameters

  ```
  ```

```
model = DQN('CnnPolicy',actor, verbose=1, buffer_size=10000, start_learning
= 1000, tensorboard_log=logdir)
```

- Start model training.

```
model.learn(total_timesteps=10000000, reset_num_timesteps=False ,
callback=callback_max_episodes,log_interval = 4)
```

  - Save the model: `model.save(f'{models_dir}/model')`
  - Then load the model: `model = DQN.load(f'{models_dir}/model.zip')`
  - check performance using the following code:

```
obs,info = actor.reset()

for i in range(10):
    actions = []
    while True:s
        action, _states= model.predict(observation = obs,
deterministic=True)
        actions.append(action)
        obs, reward,_, terminated, info = actor.step(action)

        if terminated:
            print(actions)
            obs, info = actor.reset()
            break


if actor.actor_lst:
    actor.destroy_all_actors()
```