RLController Code Explanation

Initialization

The `__init__` method initializes the RLController class, which inherits from the SumoEnv class. This initialization sets up various parameters for the traffic light controller and ramp meters.

1. Base Times:

   - tg = 10: Base green time for ramp meters.

   - tr = 2: Red time for ramp meters.

2. Shapes and Thresholds:

   - dtse_shape: Defines the shape of the state representation (3D).

   - sum_delay_sq_min: Tracks the minimum sum of squared delays.

3. Schedulers and IDs:

   - scheduler: Schedules traffic light events.

   - next_tl_id: Tracks the next traffic light ID.

4. Ramp Meters and Actions:

   - ramp_meter_ids: Example IDs for ramp meters.

   - edge_after_ramp: Edge ID after the ramp.

   - action_space_n: Number of actions (e.g., 0, 1, 2).

   - observation_space_n: Shape of the observation space.

5. Thresholds and Mappings:

   - density_threshold and flow_threshold: Density and flow thresholds.

RLController Code Explanation

- max_queue_length: Maximum allowable queue length.

- ramp_lane_mapping: Mapping from traffic light IDs to lane IDs.

RLController Code Explanation

Reset Method

The `reset` method initializes the simulation environment and sets up the scheduler for traffic light events.

1. Simulation Reset:

   - Calls simulation_reset() to reset the simulation environment.

2. Scheduler Initialization:

   - Initializes the scheduler with ramp meter IDs.

   - Retrieves the next traffic light ID.

3. Simulation Steps:

   - Steps through the simulation for the base green time (tg).

RLController Code Explanation

Step Method

The `step` method executes an action by controlling the traffic light phases and scheduling the next events.

1. Action Processing:

   - Computes green time based on the action: green_time = tg * (action + 1).

2. Traffic Light Control:

   - Sets the phase and duration for the traffic light.

   - Schedules the next traffic light event.

3. Simulation Execution:

   - Steps through the simulation until the next event.

RLController Code Explanation

Observation Method

The `obs` method retrieves the current state of the environment.

1. Retrieve Metrics:

   - Density: density = total_vehicles / total_length.

   - Flow: Number of vehicles passing through the edge.

   - Queue Length: Number of vehicles in the queue.

   - Speed: Average speed of vehicles.

2. Return Observation:

   - Returns an array of density, flow, queue length, and speed.

RLController Code Explanation

Reward Method

The `rew` method calculates the reward based on the current state.

1. Delay Calculation:

   - Sum of squared delays: sum_delay_sq = sum (1 - (vehicle_speed / v_max_speed)^2).

2. Reward Computation:

   - Normalized reward: rew = 0 if sum_delay_sq_min == 0 else 1 + sum_delay_sq / sum_delay_sq_min.

   - Penalization based on thresholds for density, flow, and queue length.

3. Return Clipped Reward:

   - Clips the reward to be between 0 and 1.

RLController Code Explanation

Done Method

The `done` method checks if the simulation should end.

1. Simulation End Check:

   - Returns true if the simulation end condition is met.