

Assignment: Incorporate Turn Lanes with Dedicated Signals into the Traffic Light Simulator

Objective: The purpose of this assignment is to add another layer of realism to the traffic light simulator by integrating turn lanes with dedicated signals. This addition will simulate the complexities of actual traffic intersections where turn lanes operate on separate signal phases from through lanes.

Requirements:

1. **Turn Lane Model:** Extend the `Intersection` struct to include turn lanes. Create a `TurnLaneSignal` struct which will manage the state of turn-specific traffic lights.
2. **Updated State Machine Logic:** Modify the state machine to include logic for the turn signals, ensuring that they are synchronized with the main traffic lights but operate on independent cycles where appropriate.
3. **Enhanced Intersection Model:** The `Intersection` struct now must manage additional `TurnLaneSignal` instances and coordinate them with the existing `TrafficLight` and `PedestrianSignal` structs.
4. **UX/UI Expansion:** Update the user interface to display the turn lane signals, providing clear indications of when it is safe for vehicles to turn. Ensure that the UI reflects the complexity of the intersection in an easy-to-understand manner.
5. **Interaction Complexity:** Implement additional UI controls that allow the simulation of vehicles arriving at and occupying turn lanes, requiring signal changes.

Deliverables:

1. **Code:** Submit the updated Rust source files with your implementation of the `TurnLaneSignal` and the modified `Intersection` struct.
2. **UI Demonstration:** A gif or video of the UI showing the intersection with its turn lane signals during a turn phase.
3. **Documentation:** A revised README file that documents your turn lane signal integration, including design decisions and how turn lane signals were synchronized with other traffic and pedestrian signals.

Evaluation Criteria:

- **Functionality:** The simulator must accurately model the operation of dedicated turn lane signals in coordination with other intersection signals.
- **Code Quality:** The code should maintain high standards of organization and readability, with a structure that reflects the intersection's increased complexity.
- **Usability:** The UI should clearly display the state of all signals and be intuitive for users, even with the added complexity of turn lanes.

Extra Credit Features:

1. **Real-Time Traffic Statistics:** Implement a feature in the UI that displays real-time statistics, such as the number of vehicles turning, waiting times, and queue lengths for each turn lane.
2. **Interactive Signal Timing Adjustment:** Create a UI component that allows the user to manually adjust the timing of the turn signals and observe the effects on traffic flow, simulating the work of traffic engineers.

Hints:

- Consider the different types of turns at intersections (right on red, left turn only on arrow, etc.) when designing your state machine logic.
- Take into account the safety intergreen period required when transitioning between conflicting traffic movements.

Submission Deadline: [Insert deadline]

Please submit the revised codebase and other deliverables by the specified deadline to [insert submission email or upload link].

This step moves us closer to a fully functional SCADA-like system for traffic light control, providing a comprehensive educational tool on traffic signal operations. Your continued attention to detail and system safety is essential.

Best of luck, and remember to simulate realistic turn lane scenarios for a true-to-life experience!