**Simulação de Trânsito Inteligente**

**Introdução**

Este projeto implementa uma simulação de controle de tráfego urbano utilizando semáforos inteligentes. A interface gráfica foi criada com a biblioteca 'Tkinter' em Python, e o sistema de semáforos é gerenciado por um algoritmo que prioriza direções com mais veículos ou aquelas previamente definidas como prioritárias.

**Funcionalidades**

* Semáforos que alternam entre verde, amarelo e vermelho com base no tempo e nas filas de veículos.
* Interface gráfica que exibe os semáforos e as filas de veículos esperando.
* Veículos são gerados de forma aleatória e direcionados para diferentes filas de semáforos.
* sistema de controle de tráfego toma decisões com base em regras e prioridades definidas.
* Relatório de fim de simulação mostrando o status dos semáforos e o tempo de espera dos veículos.

**Estrutura do Projeto**

* \*\*`traffic\_light.py`\*\*: Define a classe `TrafficLight`, que representa um semáforo com operações para abrir, fechar e alternar estados.
* \*\*`traffic\_control\_system.py`\*\*: Implementa a lógica de controle dos semáforos, gerenciando seus estados e determinando quando trocar os sinais.
* \*\*`vehicle.py`\*\*: Define a classe `Vehicle`, representando os veículos que aguardam nos semáforos.
* \*\*`app.py`\*\*: Cria a interface gráfica da simulação utilizando `Tkinter`. Controla o ciclo de vida da simulação e atualiza o estado visual dos semáforos e das filas de veículos.

**Funcionamento do Sistema**

* \*\*Semáforos\*\*: Cada semáforo pode estar nos estados verde, amarelo ou vermelho. O tempo que o semáforo permanece verde é controlado pelo sistema, e o tempo de amarelo é fixo (3 segundos). Se o semáforo permanecer aberto por muito tempo ou tiver poucos veículos aguardando, ele fecha para que outros possam abrir.
* \*\*Veículos\*\*: Os veículos são gerados aleatoriamente e colocados na fila correspondente à sua direção (norte, sul, leste ou oeste). Cada veículo tem um tempo de espera que aumenta enquanto o semáforo está vermelho.
* \*\*Sistema de Controle\*\*: A lógica de controle decide qual semáforo abrir com base no número de veículos aguardando e nas direções prioritárias (norte e sul, por padrão). A cada ciclo, o sistema atualiza o tempo de espera dos veículos e realiza a troca de sinais quando necessário.

**Customizações**

* \*\*Direções Prioritárias\*\*: O sistema prioriza certas direções (norte e sul) quando há muitos veículos esperando. Isso pode ser alterado diretamente no arquivo `traffic\_control\_system.py`, no atributo `priority\_directions` da classe `TrafficControlSystem`.
* \*\*Tempos dos Semáforos\*\*: O tempo padrão que o semáforo permanece verde pode ser ajustado no arquivo `traffic\_control\_system.py`, nos atributos `default\_green\_time` e `max\_green\_time`.

**Exemplo de Execução**

Ao iniciar a simulação, uma janela será aberta mostrando os semáforos e as filas de veículos. “Veículos” azuis são adicionados aleatoriamente e são exibidos próximos ao semáforo correspondente. Os semáforos mudam de cor conforme o tempo passa, e os veículos são removidos quando o sinal fica verde.

**Conclusão**

A simulação de trânsito inteligente fornece uma estrutura robusta para o gerenciamento de tráfego urbano com base na densidade de veículos e nas prioridades das direções. A flexibilidade da simulação permite a personalização dos tempos e das configurações de prioridade, tornando-a uma ferramenta valiosa para explorar estratégias de gestão de tráfego.