

Relatório

Projecto: Simulação Inteligente de Fluxo de Tráfego

1. Introdução

Este relatório técnico detalha o desenvolvimento e a implementação de um sistema de simulação de tráfego. O objetivo principal do projeto é modelar o comportamento de veículos e semáforos em intersecções rodoviárias, explorando a aplicação de princípios de Programação Orientada a Objetos (POO) e padrões de design para criar uma simulação flexível e extensível. A simulação permite a visualização do fluxo de tráfego e a avaliação de diferentes estratégias de controlo de semáforos, incluindo ciclos fixos e adaptativos, bem como a gestão de veículos de emergência.

2. Arquitetura do Sistema

O sistema foi concebido com base no padrão arquitetónico **Model-View-Controller (MVC)**, que promove a separação de responsabilidades, facilitando a manutenção, escalabilidade e testabilidade do código. Esta abordagem divide o sistema em três componentes principais:

- **Model:** Engloba a lógica de negócio e os dados da simulação. Inclui classes que representam as entidades do domínio, como **Vehicle** (Veículo), **Road** (Estrada), **Intersection** (Intersecção) e **TrafficLight** (Semáforo), bem como a classe **World** (Mundo), que gere o estado global da simulação e a interação entre as entidades.
- **View:** Responsável pela apresentação visual do estado da simulação ao utilizador. Implementada em **JavaFX**, utiliza um **CanvasView** para renderização gráfica de alto desempenho dos veículos, estradas e semáforos, e um **ControlPanel** para fornecer uma interface de utilizador interativa para controlo da simulação.
- **Controller:** Atua como intermediário entre o Model e a View, processando as entradas do utilizador e as atualizações do Model. Gere a lógica de simulação, as estratégias de controlo de tráfego e a interação com a interface gráfica.

3. Modelagem de Domínio

A simulação é construída sobre um conjunto de classes que representam os elementos fundamentais de um ambiente de tráfego:

- **World:** A classe central que orquestra a simulação. Contém listas de **Roads**, **Intersections** e **Vehicles**. É responsável por avançar a simulação em

cada 'tick' de tempo, gerando veículos, atualizando o estado dos semáforos e movendo os veículos. Também integra a recolha de métricas de desempenho.

- **Road** : Representa um segmento de estrada com um comprimento definido. Contém uma lista de veículos atualmente na estrada e gere a sua movimentação ao longo do segmento. Pode ter uma intersecção no seu final.
- **Intersection** : Representa um cruzamento de estradas. Contém um mapa de semáforos, associando cada estrada de entrada a um TrafficLight específico. É o ponto onde as estratégias de controlo de tráfego são aplicadas.
- **Vehicle** : A classe base para todos os veículos na simulação. Possui propriedades como ID, velocidade, posição e a estrada atual. Gere o seu movimento e interação com outros veículos e semáforos. A sua subclasse EmergencyVehicle implementa um comportamento prioritário.
- **TrafficLight** : Representa um semáforo individual. Utiliza o padrão State para gerir os seus diferentes estados (Vermelho, Amarelo, Verde) e a transição entre eles. Mantém a duração de cada estado e o tempo atual no estado.

4. Padrões de Design Utilizados

O projeto faz uso de dois padrões de design cruciais para garantir a flexibilidade e a manutenibilidade do sistema:

4.1. State Pattern (Padrão de Estado)

O **State Pattern** é empregado para gerir os diferentes estados dos semáforos (TrafficLight). Em vez de usar múltiplas instruções condicionais (if-else ou switch) para determinar o comportamento do semáforo em cada estado, o padrão delega essa responsabilidade a objetos de estado dedicados. As classes RedState , YellowState e GreenState implementam a interface LightState , cada uma definindo como o semáforo se comporta e transita para o próximo estado quando o método tick() é invocado. Isso simplifica a lógica do TrafficLight e facilita a adição de novos estados no futuro.

4.2. Strategy Pattern (Padrão de Estratégia)

O **Strategy Pattern** é utilizado para implementar diferentes políticas de controlo de tráfego. A interface Strategy define um contrato para as estratégias de controlo, e as classes FixedCycle e AdaptiveCycle fornecem implementações concretas. O World da simulação pode alternar dinamicamente entre estas estratégias, permitindo testar e comparar o desempenho de diferentes abordagens de gestão de tráfego sem modificar o código central da simulação. A

estratégia `FixedCycle` utiliza durações pré-definidas para os semáforos, enquanto a `AdaptiveCycle` ajusta as durações com base nas condições de tráfego em tempo real.

5. Funcionalidades Avançadas

5.1. Controlo Adaptativo de Semáforos

A estratégia `AdaptiveCycle` representa uma funcionalidade avançada que simula um sistema de semáforos inteligente. Esta estratégia monitoriza o número de veículos em fila (`vehiclesInQueue`) em cada estrada de entrada de uma intersecção. Com base nesta informação, a duração do estado verde (`GreenState`) é ajustada dinamicamente para otimizar o fluxo de tráfego. Um maior número de veículos em fila resulta num tempo verde prolongado, dentro de limites mínimos e máximos (`MIN_GREEN` , `MAX_GREEN`), para permitir que mais veículos passem. Esta abordagem visa reduzir o tempo de espera e melhorar a eficiência do tráfego.

5.2. Veículos de Emergência

A classe `EmergencyVehicle` estende a classe `Vehicle` e introduz um comportamento prioritário. Veículos de emergência são visualmente distintos (por exemplo, através de uma cor diferente na interface gráfica) e, crucialmente, **ignoram semáforos vermelhos**. Embora ainda respeitem a distância de segurança para o veículo à sua frente, a sua capacidade de atravessar intersecções sem parar em sinais vermelhos simula a prioridade que lhes é dada em situações reais, melhorando a resposta a emergências na simulação.

5.3. Métricas e Estatísticas

O sistema integra a recolha de métricas de desempenho através da classe `Metrics` . Esta classe é responsável por calcular e registar informações importantes sobre o fluxo de tráfego, como o tempo médio de espera dos veículos e o número total de veículos que passaram por uma intersecção. Além disso, a funcionalidade de exportação para **CSV** (Comma Separated Values) permite que os dados recolhidos sejam facilmente analisados externamente, facilitando a avaliação e comparação de diferentes cenários de simulação e estratégias de controlo.

6. Interface Gráfica (JavaFX)

A interface gráfica do utilizador (GUI) é desenvolvida utilizando **JavaFX**, uma plataforma robusta para aplicações desktop. A `MainApp` inicia a aplicação, enquanto a `CanvasView` é o componente principal para a renderização visual da simulação. Utiliza um `Canvas` para desenhar os elementos gráficos (estradas, veículos, semáforos) de forma eficiente. O `ControlPanel` oferece controlos interativos para iniciar/parar a simulação, ajustar

parâmetros (como o intervalo de geração de veículos) e alternar entre as estratégias de controlo de tráfego. A escolha do JavaFX permite uma visualização clara e interativa do comportamento da simulação.

7. Conclusão

O projeto de simulação de tráfego demonstra com sucesso a aplicação de conceitos avançados de engenharia de software, incluindo Programação Orientada a Objetos e padrões de design como State e Strategy. A arquitetura MVC garante uma base sólida para o desenvolvimento e a manutenção. As funcionalidades de controlo adaptativo de semáforos e a gestão de veículos de emergência adicionam realismo e complexidade à simulação, enquanto a recolha de métricas e a interface gráfica interativa fornecem ferramentas valiosas para análise e experimentação. Este sistema serve como uma plataforma eficaz para estudar e otimizar o fluxo de tráfego em ambientes simulados.