

Relatório Técnico do Repositório: Grafo

Visão Geral

O repositório Grafo é um projeto desenvolvido em Java que simula o funcionamento de uma cidade utilizando estruturas de grafo, veículos, semáforos e uma interface gráfica interativa implementada com JavaFX. O sistema representa ruas como arestas e cruzamentos como vértices, permitindo a movimentação de veículos de forma realista, controlada por regras de trânsito, semáforos e heurísticas para otimização de deslocamento.

Funcionalidades Principais

[OK] Simulação de Tráfego Urbano:

- O projeto modela veículos que percorrem rotas definidas dentro de um grafo.
- Cada veículo possui lógica de movimentação que respeita regras de trânsito, como a parada em semáforos e a escolha de caminhos.

[OK] Controle de Semáforos:

- Implementação de semáforos nos cruzamentos, que alternam seus estados (aberto/fechado) conforme uma heurística definida.
- As heurísticas simulam de forma simplificada a gestão de fluxo, buscando evitar congestionamentos.

[OK] Movimentação Baseada em Grafo:

- Estrutura de dados orientada a grafos, com vértices e arestas, representa de maneira fiel o mapa da cidade.
- Os veículos percorrem as arestas com controle de posição, respeitando distâncias mínimas entre si.

[OK] Interface Gráfica com JavaFX:

- Visualização em tempo real da cidade, com veículos em movimento, mudanças de estado dos semáforos e feedback visual das operações.
- Elementos gráficos bem posicionados e representativos, como retângulos para veículos e nós

destacados para os cruzamentos.

[OK] Flexibilidade na Configuração:

- O projeto consome arquivos JSON para importar as configurações do grafo, permitindo uma fácil alteração e expansão do mapa sem necessidade de recompilar o código.
- Facilita testes com diferentes layouts urbanos e cenários.

Dificuldades e Desafios

[!] Consumo de JSON para Geração do Grafo:

- A leitura de arquivos JSON com descrições complexas de vértices e arestas apresentou desafios, especialmente no mapeamento correto das conexões entre nós.
- A necessidade de validar dados e garantir a integridade da estrutura do grafo adicionou complexidade ao processo de inicialização.

[!] Heurísticas e Semáforos Quebrados:

- Implementar heurísticas eficientes para alternância dos semáforos foi um desafio, pois exige equilíbrio entre fluxo contínuo e prevenção de congestionamentos.
- O sistema precisa lidar com casos de "semáforos quebrados" ou mal configurados, o que pode causar travamentos ou rotas inviáveis na simulação.

[!] Complexidade do Código:

- A integração entre múltiplos componentes - grafo, veículos, semáforos e interface gráfica - aumentou significativamente a complexidade do código.
- Garantir a sincronização entre a lógica de movimentação e a renderização gráfica demandou um planejamento cuidadoso.

[!] Implementação da Interface Gráfica com JavaFX:

- Embora poderosa, a biblioteca JavaFX exige um bom domínio para implementação eficiente de animações e interações.
- O gerenciamento do ciclo de atualização da interface, especialmente com múltiplos veículos em movimento simultâneo, foi uma das partes mais desafiadoras do projeto.

Pontos Fortes do Projeto

- [*] Estrutura bem organizada, modular e orientada a objetos.
- [*] Uso de boas práticas de desenvolvimento, como separação de responsabilidades.
- [*] Simulação realista e visualmente clara com JavaFX.
- [*] Flexibilidade e escalabilidade, permitindo expansão do grafo e inserção de novas regras.

Sugestões de Melhoria

- Implementação de logs ou painel de status na interface para depuração e monitoramento em tempo real.
- Adição de testes automatizados para verificar a consistência do grafo gerado via JSON.
- Otimização das heurísticas de semáforo, incorporando algoritmos baseados em fluxo de veículos.

Conclusão

O projeto Grafo é uma simulação robusta e didática do funcionamento de um sistema urbano baseado em grafos, que integra algoritmos, estruturas de dados e interface gráfica com excelência. Apesar dos desafios naturais associados à manipulação de dados estruturados e à complexidade de sistemas gráficos em Java, o resultado final é um software funcional, extensível e com potencial para aplicações acadêmicas e de pesquisa.