

## Trabalho 2

### Simulador de Tráfego em Malha Viária

#### Objetivo do Trabalho

Desenvolver um simulador de tráfego. Neste simulador há veículos que se movem pelas vias de uma malha viária. A malha é informada através de arquivo.

O trabalho pode ser realizado **em duplas** (trios são aceitos, mas com diferenciais conforme abaixo).

#### Especificação dos Veículos

- Cada veículo deve ser um *thread*. ATENÇÃO: não utilizar *ExecutorService* para rodar as *threads*.
- O veículo se movimentará pela malha, uma posição por vez, respeitando o sentido de fluxo da pista. O veículo só pode se mover caso a posição à frente esteja livre.
- Ao se deparar com um cruzamento:
  - Deve escolher, aleatoriamente, uma das vias de saída do cruzamento para seguir viagem. A escolha deve ser feita antes do veículo ingressar no cruzamento.
- Só deve se mover pelo cruzamento se todas as posições por onde vai passar estiverem totalmente livres (exclusão mútua).
- Não deve bloquear o cruzamento de outros veículos (ficar parado no cruzamento).
- Novos veículos são inseridos nos pontos de entrada da malha (ver especificação da malha)
- Ao atingir um ponto de saída (ver especificação da malha), o veículo deve ser encerrado.
- Veículos possuem velocidades diferentes (tempo de *sleep* da *thread* a cada passo).

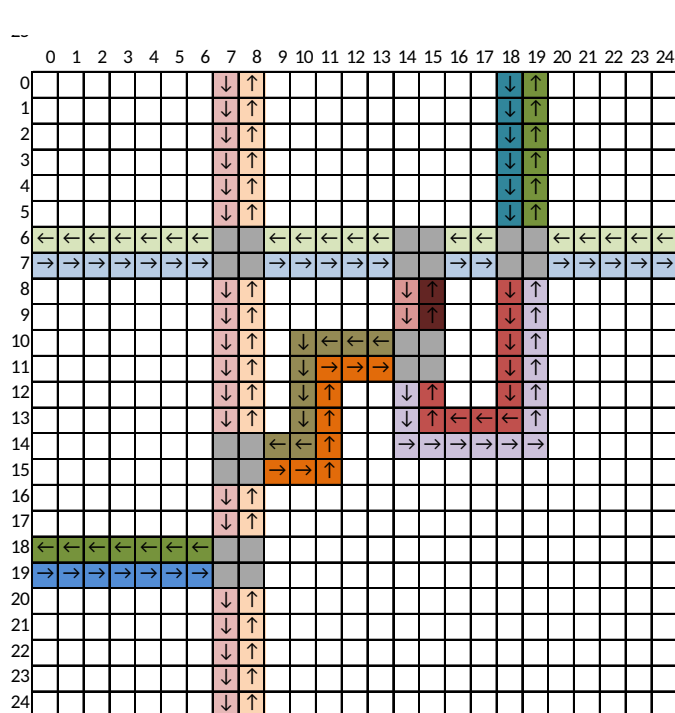
#### Especificação da Malha Viária

- Deve ser carregada de um arquivo texto.
- Nas duas primeiras linhas estão a quantidade de linhas e colunas da malha, respectivamente.
- As próximas linhas especificam o tipo de cada segmento (célula) da malha:

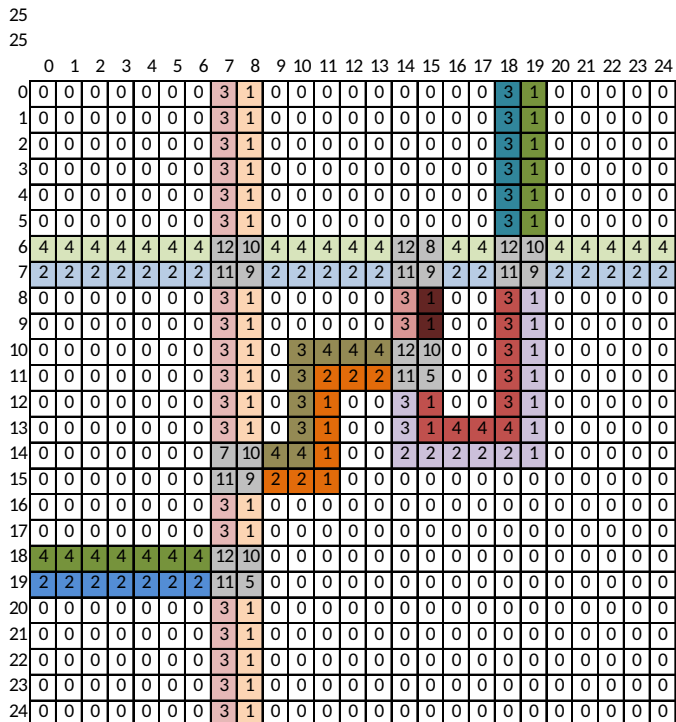
Valor (int)	Tipo de segmento (movimentos permitidos)
0	Nada (célula não usada pela malha)
1	Estrada Cima
2	Estrada Direita
3	Estrada Baixo
4	Estrada Esquerda
5	Cruzamento Cima
6	Cruzamento Direita
7	Cruzamento Baixo
8	Cruzamento Esquerda
9	Cruzamento Cima e Direita
10	Cruzamento Cima e Esquerda
11	Cruzamento Direita e Baixo
12	Cruzamento Baixo e Esquerda

- Características gerais das vias:
  - São sempre horizontais ou verticais (não haverá vias em diagonal).
  - São de mão dupla, ou seja, possuem duas pistas.
  - Nas bordas, só haverá vias perpendiculares.
  - Entre vias paralelas, haverá sempre ao menos uma linha ou coluna em branco (ou seja, não haverá vias "grudadas" umas nas outras).
- Identificação de pontos de entrada e de saída de veículos:
  - Entrada:** posição inicial da pista que está em uma das bordas da malha.
  - Saída:** posição final da pista que está em uma das bordas da malha.

Exemplo de malha e arquivo:



Malha



Arquivo

## Especificação Geral do Sistema

- A malha e os veículos devem ser visualizados em uma interface gráfica.
- O sistema **deve obrigatoriamente** possuir opções para:
  - limitar quantidade de veículos: usuário informar a quantidade máxima de veículos que poderão estar circulando **simultaneamente** na malha. O sistema fica inserindo veículos até atingir esta quantidade. Assim que veículos saírem da malha, novos veículos devem ser inseridos para manter a quantidade máxima circulando simultaneamente.
  - iniciar simulação: fica inserindo veículos, que se movimentam na malha, respeitando a quantidade descrita anteriormente no item (i).
  - encerrar simulação: para de inserir e *encerra imediatamente* todos os veículos.
- O sistema **pode** possuir opções para:
  - usuário informar um intervalo de inserção de veículos (de quanto em quanto tempo um novo veículo é inserido na malha).
  - após encerrada a simulação, aguardar os veículos existentes saírem da malha.
- O sistema deve suportar dois mecanismos de exclusão mútua (semáforos e monitores), devendo existir uma opção para o usuário escolher qual destes mecanismos será utilizado.
  - Equipes de três integrantes devem implementar um terceiro mecanismo de gerenciamento da região crítica, denominado **troca de mensagens** (consultar vídeo explicativo [nesse link](#)).

## CRITÉRIOS DE AVALIAÇÃO

- Funcionalidades do sistema (deve atender todas as especificações acima).
- Ausência de erros. A movimentação dos veículos deve estar livre de impasses e sobreposições.
  - Na apresentação, *breakpoints* devem ser utilizados para evidenciar que o comportamento dos veículos e a exclusão mútua garantem a ausência de impasses e sobreposições.
- Adoção de técnicas de engenharia de software quando necessário ("ausência de gambiarras").
- Elaboração e uso de slides organizados, claros e com linguagem correta.
- Exposição do projeto do sistema (no mínimo diagrama de classes), das técnicas/padrões de projeto utilizados, dificuldades encontradas e soluções adotadas.
- Linguagem verbal adequada, termos e expressões relacionados ao assunto.
- Aproveitamento do tempo (conforme tempo máximo estabelecido e divulgado previamente).

- Demonstração do software desenvolvido, com execução passo-a-passo (modo debug) evidenciando ausência de impasses e sobreposições de veículos.
- No dia da apresentação, outras malhas poderão ser usadas pelo professor para testar o sistema.

### **ATENÇÃO**

As notas de implementação e apresentação serão individualizadas por aluno, em função da clareza, coerência, domínio do assunto e do software, e respostas às perguntas.

---

## **DATAS E ENTREGAS**

### **Até 20/4**

- Criar repositório no Github e adicionar o professor como colaborador (usuário: [santos-fernando](#))
- O arquivo README.MD deve conter o nome de todos os integrantes da equipe.
- **Perda de 2.0 pontos por semana de atraso na criação e compartilhamento do repositório.**

### **13/5 até 19:00h push final para correção**

- Nesta data/horário os repositórios serão clonados para correção final. Portanto a equipe deve ter feito *push* de todo o conteúdo (apresentação, documentos, códigos fonte, etc) no repositório até esta data/horário
- **Perda de 1.0 ponto por dia de atraso na entrega do trabalho (conforme data do *github*).**

### **13/5 a 20/5 a partir das 19:00h**

- Apresentações dos trabalhos. As apresentações serão por equipes voluntárias, ou sorteio caso não haja equipes voluntárias.