

## Trabalho **4** – Cidade em OpenGL

### Descrição:

Implemente um programa em C++, usando a API OpenGL, um programa que simule um veículo 3D andando em uma cidade 3D. O usuário deve poder guiar o veículo no cenário. A cidade pode ser composta por cubos (simulando prédios) e modelos geométricos mais detalhados.

A aplicação não necessita ter botões ou interfaces. A animação do veículo deve ser suave (não usar refresh do teclado).

O código deve estar bem estruturado em C++ e deve fazer uso de vetores e transformações geométricas.

Requisitos básicos (Max: 9.0)

- O veículo pode ser formado por qualquer geometria (exceto esfera).
- A movimentação do veículo deve ser feita **obrigatoriamente** com vetores (direção, posição), caso contrario o trabalho **não será aceito**.
- Todos os modelos devem ter vetores normais. Deve-se obrigatoriamente incluir alguma iluminação na cena.
- Deve-se poder selecionar o tipo de câmera e posição em relação ao carro: visão interna, frontal, traseira, de cima, etc.
- Controle de FPS

Avançados (valores máximos por item)

- Modelagem de objetos com sweep (2 pts)
- O relevo definido por uma malha curva, gerada com alguma técnica de modelagem de superfícies curvas (Bezier ou B-spline) (2 pts).
- Carregamento de modelos 3D usando qualquer API (2 pts)
- Incluir textura (1 pt)
- Simular veículo com Aceleração, física, etc. (1 pt)
- Sinaleiras nas ruas. (2 pt)
- Adicionar outros veículos autônomos no cenário (2 pts)

### Data e Formato de Entrega:

- Data: até 12/07/2017 – Apresentação individual. Agende um horário.

- No email e no cabeçalho do arquivo, devem conter o nome completo e matrícula do aluno. O arquivo deve ser enviado para [pozzer3@gmail.com](mailto:pozzer3@gmail.com) com o *subject* “CG – T4”. Deve-se enviar fontes, projeto (codeblocks).
- O programa deve ser enviado em um arquivo compactado **fulano.rar** (fulano = login ou nome do aluno). Dentro deste arquivo deve haver um diretório com o mesmo nome do arquivo e dentro deste diretório os arquivos do trabalho.
- **Ex:** o arquivo **pozzer.rar** deve conter um diretório chamado pozzer, e dentro do diretório devem estar os arquivos do trabalho, como por ex: math.dev, math.cpp, math.h, aux.cpp, ...

#### **Critério de Avaliação:**

- **documentação:** descrever no cabeçalho de cada arquivo a idéia geral do código e detalhes específicos de partes que mereçam uma explicação – não comente por exemplo o que faz b++.
- **pontualidade:** Trabalhos não entregues na data não serão avaliados e receberão nota zero.
- **legibilidade:** nome de variáveis, estruturação do código. O código digital a ser entregue deve ter **3 espaços de indentação** e não deve possuir tabulações.
- **clareza:** facilidade de compreensão – evite códigos complexos e desnecessários. Adote a solução mais simples possível.
- **funcionalidade:** o programa deve satisfazer todos os requisitos. Programas que não compilarem ou que não atenderem nenhum requisito receberão nota 0 (zero).