

## 1. Identificação dos Alunos e Horas Despendidas

### Grupo 06

| Aluno            | Número | Horas Despendidas |
|------------------|--------|-------------------|
| Rodrigo Frutuoso | 61865  | 27 horas          |
| Simão Alexandre  | 61874  | 25 horas          |
| Tiago Leite      | 61863  | 28 horas          |

**Total do grupo: 80 horas**

## 2. Descrição da Aplicação e Mecanismos de Interação

### 2.1 Visão Geral

A aplicação desenvolvida simula um veículo automóvel que pode circular livremente num ambiente 3D e entrar numa garagem. O projeto foi implementado em **Python** utilizando **OpenGL 2.1** através das bibliotecas **PyOpenGL** e **GLUT**.

### 2.2 Requisitos Principais Implementados

#### *Veículo com Rodas Diferenciadas*

- **Rodas traseiras:** Raio de 1.5 unidades, espessura de 0.6 unidades
- **Rodas dianteiras:** Raio de 1.2 unidades, espessura de 0.5 unidades (mais estreitas)
- Todas as rodas incluem **jantes detalhadas com 5 raios radiais**
- Material: Borracha preta (0.05, 0.05, 0.05)

#### *Portas Interativas*

- **2 portas laterais** (esquerda e direita) que abrem por interação do utilizador
- **Tecla Q:** Abre/fecha porta esquerda (rotação de 60°)
- **Tecla E:** Abre/fecha porta direita (rotação de 60°)
- Animação instantânea com rotação no eixo Y
- Pivot point preciso para rotação realista

#### *Volante Funcional*

- Volante modelado com **torus** (aro) e **3 raios radiais**
- **Rotação automática** baseada no ângulo de direção (steering) do carro
- Multiplicador visual de **3×** para efeito mais perceptível
- Posicionamento realista no lado esquerdo do condutor

#### *Rotação das Rodas durante Movimento*

- As **4 rodas giram automaticamente** quando o veículo se desloca
- Velocidade de rotação calculada com base na velocidade linear e raio da roda
- Fórmula:  $\text{rot\_speed} = (\text{vel} / (2\pi \times \text{raio})) \times 360^\circ$
- **Rodas dianteiras:** Incluem rotação de direção (steering) além da rotação de rolamento
- **Rodas traseiras:** Apenas rotação de rolamento

### *Garagem com Portão Interativo*

- Garagem completa com **3 paredes laterais e teto**
- Portão animado que abre/fecha verticalmente
- **Tecla G:** Abre/fecha o portão da garagem (rotação de 90°)
- **Sistema de colisão:** Impede fechamento do portão quando o carro está a atravessar
- **Animação suave:** Movimento gradual a 20°/s
- Dimensões: 20×20 unidades (profundidade×largura)

### *Sistema de Iluminação Completo (6 Luzes)*

**Iluminação Diurna:** - **GL\_LIGHT0** (Sol - Luz Direcional): - Posição: (0.45, 0.9, 0.35, 0.0) - Cor: Branca (1.0, 1.0, 1.0) - Ambiente: Azulado suave (0.18, 0.18, 0.22)

- **GL\_LIGHT1** (Luz da Garagem - Posicional):
  - Posição: (-25.0, 8.0, 0.0, 1.0)
  - Cor: Laranja/quente (1.0, 0.7, 0.4)
  - Especular: (0.8, 0.6, 0.3)
- **GL\_LIGHT2 e GL\_LIGHT3** (Postes de Rua):
  - Posições: (-7.0, 9.3, ±13.0)
  - Cor: Branco quente (1.0, 1.0, 0.9)
  - Atenuação: Constante=0.5, Linear=0.05, Quadrática=0.01

**Iluminação Noturna (adicional):** - **GL\_LIGHT4 e GL\_LIGHT5** (Faróis do Carro - Spotlight): - Cor: Amarelo quente (1.0, 1.0, 0.8) - Ângulo do cone: 40° - Atenuação: Constante=0.3, Linear=0.08, Quadrática=0.015 - Direção dinâmica: Acompanha rotação do carro

### *Sistema Dia/Noite*

- **Tecla L:** Alterna entre modo dia e noite
- **Modo Dia:**
  - Céu azul claro (0.5, 0.7, 1.0)
  - Todas as luzes ativas exceto faróis
  - Ambiente normal (0.2, 0.2, 0.2)
- **Modo Noite:**
  - Céu escuro (0.05, 0.05, 0.15)
  - Apenas postes e faróis do carro ligados
  - Ambiente muito escuro (0.05, 0.05, 0.1)
  - Faróis com efeito emissivo

### *Cinco Materiais Diferentes*

1. **Carroçaria** (Pintura Texturada):
  - Textura: Pintura.jpg
  - Base: Branco (1, 1, 1) para não alterar textura
2. **Rodas** (Borracha):
  - Cor: Preto mate (0.05, 0.05, 0.05)
  - Sem brilho especular
3. **Jantes** (Metal):
  - Aro: Cinza claro (0.7, 0.7, 0.7)
  - Centro: Cinza escuro (0.3, 0.3, 0.3)
  - Raios: Cinza médio (0.4, 0.4, 0.4)
4. **Vidro Frontal** (Transparente):
  - Cor: Azulado (0.7, 0.8, 0.9, 0.15)

- Material especular: (0.3, 0.3, 0.3)
  - Shininess: 20.0
  - Alpha blending para transparência
5. **Faróis:**
- Corpo: Cinza escuro (0.2, 0.2, 0.2)
  - Lente dianteira: Amarela brilhante com emissão no modo noite
  - Lente traseira: Vermelha (1.0, 0.0, 0.0) com emissão no modo noite

### *Controlo da Câmara pelo Utilizador*

Três modos de câmara (Tecla **C** para alternar):

1. **Modo Livre** (padrão):
  - **Setas**  $\leftarrow/\rightarrow$ : Rotação horizontal ( $-5^\circ/+5^\circ$ )
  - **Setas**  $\uparrow/\downarrow$ : Rotação vertical ( $-5^\circ/+5^\circ$ , limite:  $\pm 80^\circ$ )
  - **Page Up/Down**: Zoom in/out (distância: 5-200 unidades)
  - Posição inicial: Distância 35, Ângulo H:  $45^\circ$ , Ângulo V:  $20^\circ$
2. **Modo 3ª Pessoa**:
  - Câmara segue o carro a **25 unidades** de distância
  - Altura: **12 unidades** acima do carro
  - Acompanha rotação e posição do veículo automaticamente
  - Look-at aponta para o carro
3. **Modo 1ª Pessoa**:
  - Câmara posicionada no interior do veículo (lado esquerdo do condutor)
  - Visão para a frente na direção de movimento
  - Simula perspectiva realista do condutor

### *Chão e Estrada Texturados*

- **Chão**: Textura de mosaico (Mosaico\_Chao.png)
  - Área:  $200 \times 200$  unidades
  - Repetição:  $50 \times 50$  tiles
  - Mapeamento com GL\_REPEAT
- **Estrada**: Textura de asfalto (asphalt\_clean.png)
  - Dimensões: De  $X=-29.5$  até  $X=100$ ,  $Z=\pm 10$
  - Ligeiramente elevada ( $y=0.02$ ) para evitar z-fighting
  - Repetição adequada para continuidade visual

## **2.3 Requisitos Secundários Implementados**

### *Movimento com Viragem Realista*

- **Tecla W**: Acelerar para frente (velocidade máxima: 30 unidades/s)
- **Tecla S**: Recuar (velocidade máxima: 18 unidades/s, 60% da frontal)
- **Tecla A**: Virar à esquerda (máx:  $35^\circ$ )
- **Tecla D**: Virar à direita (máx:  $35^\circ$ )
- **Barra de Espaço**: Travão de mão (força: 15 unidades/s<sup>2</sup>)
- **Sistema físico**:
  - Aceleração: 2.0 unidades/s<sup>2</sup>
  - Fricção natural: 2.5 unidades/s<sup>2</sup> (quando sem input)
  - Modelo de Ackermann para viragem realista
  - Recentragem automática da direção ( $80^\circ/s$ )

### *Volante Controlado pela Direção*

- Rotação automática do volante baseada no steering angle
- Multiplicador de 3× para visibilidade
- Recentragem automática quando não há input de direção
- Sincronizado com rotação das rodas dianteiras

### *Câmara Segue o Veículo*

- Modo 3ª pessoa com seguimento suave
- Atualização contínua da posição e rotação
- Distância e altura configuráveis

### *Câmara no Interior do Veículo*

- Posição do condutor (lado esquerdo)
- Visão frontal na direção de movimento
- Offset lateral de -3 unidades para realismo
- Altura de 4 unidades (nível dos olhos)

### *Texturas Aplicadas (5 Texturas)*

1. **Matrícula.png:** Matrículas frontal e traseira
2. **Pintura.jpg:** Carroçaria do carro
3. **Mosaico\_Chao.png:** Chão/relva
4. **asphalt\_clean.png:** Estrada
5. **tree\_bark.png:** Tronco das árvores

Todas com: - Mipmaps automáticos (gluBuild2DMipmaps) - Filtros lineares para suavização - Mapeamento UV correto

### *Vidro Frontal Transparente*

- Transparência com alpha blending ( $\alpha = 0.15$ )
- Tom azulado (0.7, 0.8, 0.9)
- Material com reflexos especulares (shininess: 20)
- Renderização bidirecional (sem face culling)
- Desabilitação temporária de depth write

### *Sistema de Colisão Completo*

1. **Colisão com portão da garagem:**
  - Verifica posição da frente e traseira do carro
  - Impede entrada/saída quando portão fechado
  - Mensagem de erro quando tenta fechar com carro no meio
2. **Colisão com paredes da garagem:**
  - 3 paredes (2 laterais + 1 traseira)
  - Detecção AABB (Axis-Aligned Bounding Box)
  - Para o carro instantaneamente
3. **Colisão com obstáculos:**
  - Árvores (2): Raio de 5 unidades
  - Postes (2): Raio de 2.5 unidades
  - Detecção circular (distância euclidiana)
4. **Limites do mundo:**
  - Área delimitada: -100 a +100 em X e Z
  - Clamp automático de posição

### Elementos de Cenário

1. **Árvores (2 unidades):**
  - Tronco texturado (tree\_bark.png)
  - 2 cones verdes para folhagem
  - Posições: (-15, 0, ±17)
2. **Postes de Iluminação (2 unidades):**
  - Poste metálico cinza
  - Braço horizontal
  - Lâmpada esférica com material emissivo
  - Luz posicional (GL\_LIGHT2 e GL\_LIGHT3)
  - Posições: (-10, 0, ±13)

### Faróis Funcionais

1. **Faróis Dianteiros (2):**
  - Corpo: Cilindro cinza (raio: 0.7)
  - Lente: Esfera amarela (raio: 0.6)
  - Posições: (-6.0, 2.5, ±4.8)
  - Iluminação: GL\_LIGHT4 e GL\_LIGHT5 (spotlight)
  - Ligam automaticamente à noite
2. **Faróis Traseiros (2):**
  - Corpo: Cilindro cinza (raio: 0.5)
  - Lente: Esfera vermelha (raio: 0.45)
  - Posições: (6.0, 2.5, ±4.8)
  - Efeito emissivo vermelho à noite

### Assentos no Interior

- **2 assentos** (condutor e passageiro)
- Compostos por:
  - Base (1.5×0.3×1.5)
  - Encosto (1.5×1.6×0.3)
  - Apoio de cabeça (1.0×0.5×0.3)
- Cor: Cinza escuro (0.2, 0.2, 0.2)
- Posições: (1.5, 3, ±3.0)

## 3. Descrição do Grafo de Cena

### 3.1 Estrutura do Grafo

O grafo de cena foi implementado através da classe **Node**, que permite criar uma hierarquia transformacional:

**class** Node:

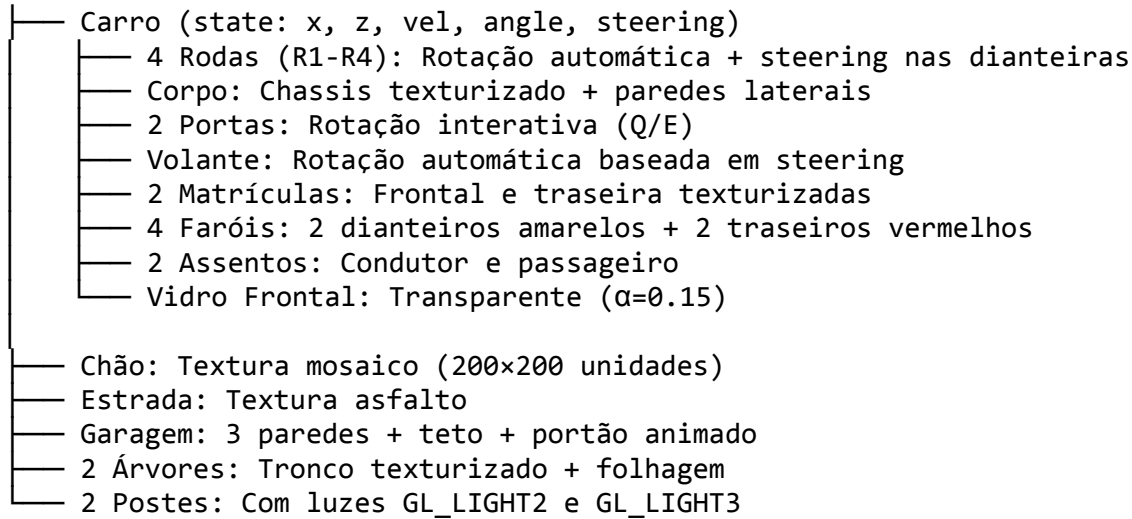
- name: Identificador do nó
- geom: Função de geometria a desenhar
- transform: Função de transformação (translação, rotação, escala)
- updater: Função de atualização por frame
- state: Dicionário de estado (posição, ângulos, velocidade)
- children: Lista de nós filhos

**Métodos principais:** - add(\*kids): Adiciona filhos ao nó - update(dt): Atualiza estado e propaga para filhos - draw(): Renderiza geometria e filhos (com glPush/PopMatrix)

## 3.2 Hierarquia Simplificada

### Estrutura principal do grafo de cena:

World



**Componentes das geometrias principais:** - **Rodas:** Cilindro (pneu) + jantes com 5 raios radiais - **Volante:** Torus (aro) + esfera central + 3 raios - **Faróis:** Cilindro (corpo) + esfera (lente) com emissão noturna - **Garagem:** Portão com animação gradual (20°/s)

## 4. Screenshots Principais ([Pasta com Screenshots](#))

### Screenshot 1: Vista Geral - Modo Dia

- Veículo, garagem, estrada texturizada, árvores e postes
- Iluminação diurna completa (sol + luz garagem + postes)
- Céu azul, chão com textura mosaico

### Screenshot 2: Modo Noite com Faróis

- Faróis dianteiros amarelos e traseiros vermelhos acesos
- Cones de luz visíveis (spotlight 40°)
- Postes de rua iluminando, ambiente escuro

### Screenshot 3: Portas Abertas e Interior

- Portas laterais abertas (60°) - teclas Q/E
- Interior visível: volante, assentos, vidro transparente

### Screenshot 4: Câmara 1ª Pessoa

- Vista do condutor, volante rotacionável
- Vidro frontal transparente ( $\alpha=0.15$ )

### Screenshot 5: Câmara 3ª Pessoa

- Seguimento do veículo (dist: 25, altura: 12)
- Rodas em movimento, matrículas visíveis

### Screenshot 6: Garagem com Portão Aberto

- Veículo entrando, portão animado (90°)
- Sistema de colisão ativo, iluminação interna

## 5. Link para o Código Fonte

Repositório GitHub: <https://github.com/SimaoAlexandre/Computacao-Grafica>

### Estrutura do projeto:

```
/
├── PrCG.py           # Código principal (1230 linhas)
├── Matrícula.png     # Textura da matrícula
├── Mosaico_Chao.png  # Textura do chão
├── Pintura.jpg        # Textura da carroçaria
├── asphalt_clean.png  # Textura da estrada
├── tree_bark.png      # Textura do tronco das árvores
├── RELATORIO.md       # Este relatório
└── README.md         # Instruções de execução
```

### Dependências:

- Python 3.x
- PyOpenGL
- PyOpenGL-accelerate
- Pillow (PIL)
- FreeGLUT

## 6. Implementação Técnica

### 6.1 Física do Veículo

- **Modelo de Ackermann:** Wheelbase 10 unidades, steering máx  $\pm 35^\circ$
- **Velocidades:** Frente 30 u/s, Ré 18 u/s
- **Aceleração:**  $2.0 \text{ u/s}^2$ , Fricção:  $2.5 \text{ u/s}^2$ , Travagem:  $15.0 \text{ u/s}^2$
- **Rotação das rodas:**  $\omega = v_{el} / (2\pi r)$  com raios específicos

### 6.2 Sistema de Iluminação

- **6 Luzes OpenGL** com atualização dinâmica
- **Faróis:** Posicionamento calculado com vetores forward/right
- **Spotlight:** Direção atualizada continuamente, cone  $40^\circ$
- **Dia/Noite:** Luzes alternadas, ambiente e céu ajustados

### 6.3 Texturas e Transparências

- **5 Texturas** com mipmaps (gluBuild2DMipmaps)
- **Wrapping:** GL\_REPEAT (chão/estrada), GL\_CLAMP\_TO\_EDGE (matrícula)
- **Vidro:** Alpha blending ( $\alpha=0.15$ ), glDepthMask(GL\_FALSE), face culling desabilitado

### 6.4 Sistema de Colisões

- **Portão:** AABB + verificação de estado, impede fechamento com carro no meio
- **Paredes:** AABB com 3 paredes da garagem
- **Obstáculos:** Detecção circular para árvores (raio 5) e postes (raio 2.5)
- **Limites:** Mundo delimitado a  $[-100, 100]$  em X e Z

### 6.5 Controlo e Câmara

- **Input contínuo:** Conjunto keys\_pressed, processamento em cada frame
- **3 Modos de câmara:** Livre (setas + Page Up/Down), 3ª pessoa (seguimento), 1ª pessoa (interior)
- **Delta time:** Movimento independente de FPS ( $\sim 60 \text{ FPS}$ )

## 7. Conclusões

### 7.1 Objetivos Alcançados

Todos os requisitos principais e maioria dos secundários implementados Sistema de cena hierárquico robusto (classe Node) Animações suaves com física realista (Ackermann steering) Sistema de iluminação complexo (6 luzes + dia/noite) Transparências e 5 texturas corretamente aplicadas Sistema de colisão completo (portão, paredes, obstáculos)

### 7.2 Dificuldades e Soluções

- **Transparências:** Resolvido com ordenação de renderização + glDepthMask
- **Faróis dinâmicos:** Cálculo com vetores forward/right do carro
- **Colisão com portão:** Verificação de posição frente e traseira
- **Rotação das rodas:** Fórmula correta  $\omega = v / (2\pi r)$  com raios específicos

### 7.3 Melhorias Futuras

- Sombras projetadas (shadow mapping)
- Mais veículos, sistema de partículas
- HUD com velocímetro
- Luzes de travão e indicadores de direção

## 8. Referências

- [OpenGL 2.1 Reference](#)
- [PyOpenGL Documentation](#)
- Aulas de Computação Gráfica (DI-FCUL)
- [Ackermann Steering](#)