

LUCAS FORTOLAN SAMPAIO

**REPRESENTAÇÃO TRIDIMENSIONAL DE UM VEÍCULO DE
FÓRMULA 1 EM PYOPENGL COM CONTROLE DE
CÂMERA E MOVIMENTAÇÃO**

ORIENTADO: Prof. Dr Maurício Acconcia Dias

**ARARAS/SP
Novembro/2025**

INTRODUÇÃO

O Tyrrell 007 foi o carro utilizado pela equipe Tyrrell Racing Organisation na temporada de Fórmula 1 de 1974. Projetado por Derek Gardner como sucessor do modelo 006, o 007 marcou uma fase de transição importante para a equipe, apresentando uma aerodinâmica mais limpa e um chassi de geometria simplificada. Essa combinação conferiu ao carro uma identidade visual muito característica, típica da F1 dos anos 1970.

Na Imagem 1, observa-se o Tyrrell 007 de 1974.

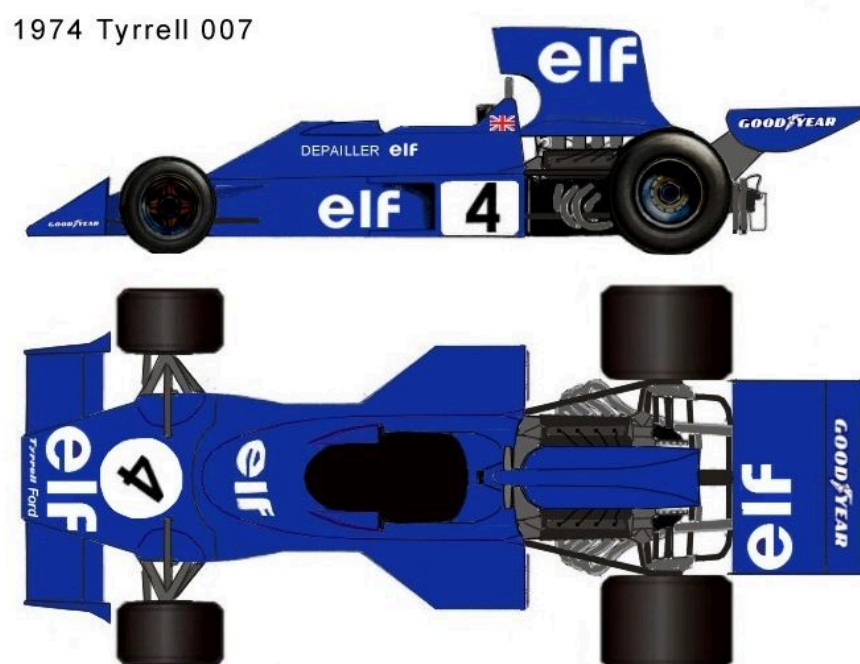


Imagem 1 — Tyrrell 007 (1974)

O modelo se destaca como um carro de Fórmula 1 clássico, com formas bem definidas e superfícies amplas. Essas características tornam sua construção especialmente adequada para modelagem 3D baseada em primitivas geométricas, o que o transforma em uma excelente escolha para estudos introdutórios de computação gráfica e representação tridimensional.

DESENVOLVIMENTO

O código desenvolvido implementa uma cena tridimensional utilizando exclusivamente PyOpenGL para renderização e pygame para gerenciamento da janela, captura de eventos e exibição de textos. O processo de modelagem iniciou-se pela análise estrutural do carro real, traçando sobre sua imagem as formas geométricas que compõem sua silhueta, conforme visualizado na Figura 2. Esse procedimento permitiu identificar quais partes poderiam ser representadas por primitivas básicas, facilitando o planejamento da construção tridimensional. Na figura foram desenhados retângulos, que na prática correspondem à junção de dois triângulos, evidenciando a decomposição estrutural da forma do veículo. Com essa etapa concluída, iniciou-se o desenvolvimento do código responsável por gerar essas primitivas em PyOpenGL, traduzindo cada componente do carro para sua respectiva forma geométrica e servindo como base para a montagem progressiva do modelo 3D.



Imagem 2 — Identificação das figuras geométricas sobre a imagem do carro

Para possibilitar a construção da cena, foram utilizadas as seguintes bibliotecas: **pygame**, empregada para criar e gerenciar a janela da aplicação, capturar eventos do teclado e escrever textos na tela; **pygame.locals**, que fornece constantes que facilitam o tratamento de eventos; **OpenGL.GL**, que contém as funções principais do OpenGL responsáveis pelo desenho da cena 3D, incluindo criação de primitivas, configurações de material, transformações e controle do pipeline gráfico; **OpenGL.GLU**, que oferece utilitários gráficos adicionais e simplifica a configuração da câmera e da perspectiva; e **math**, biblioteca padrão do Python utilizada para operações matemáticas auxiliares, especialmente cálculos de ângulos, rotações e funções trigonométricas empregadas na animação.

A estrutura do programa foi organizada de forma modular, permitindo que cada elemento da cena fosse construído de maneira independente, mas integrada ao conjunto. As primitivas geométricas — como caixas, cilindros e superfícies circulares — serviram de base para a modelagem das partes do carro, que incluem rodas, chassi, aerofólios, sidepods, cockpit e demais componentes estruturais. A pista infinita foi implementada por meio de uma estrutura que se desloca continuamente

no eixo longitudinal, gerando a sensação de movimento constante enquanto o carro permanece relativamente fixo no enquadramento da câmera.

O funcionamento da simulação depende de um conjunto de variáveis globais responsáveis por manter o estado da cena, como rotação e aproximação da câmera, velocidade do carro, posição na pista e ângulo de giro das rodas. A animação é controlada pelo usuário, que pode iniciar o movimento, ajustar a velocidade ou encerrar a aplicação. Uma vez ativada, a animação prossegue automaticamente, utilizando transformações de translação, rotação e escalonamento para atualizar a cena a cada quadro. A câmera é configurada por meio de funções da biblioteca GLU, permitindo acompanhar o deslocamento da pista e manter o carro como foco principal da visualização.

Além da modelagem geométrica, o código inclui o uso de texturas aplicadas a elementos específicos, como logos de patrocinadores, geradas a partir de superfícies de texto criadas com pygame. Esse recurso adiciona riqueza visual e aproxima o modelo do veículo real representado. A renderização dos textos exibidos na interface também é realizada através da conversão de superfícies pygame em texturas OpenGL, integrando informações diretamente na tela sem interromper a fluidez da animação.

Além da construção geométrica, o código implementa comandos interativos destinados ao controle do modelo durante a execução. A função *desenhar_texto_na_tela* exibe na interface as instruções essenciais para a navegação. Sendo os comandos: espaço para iniciar ou parar a movimentação do carro; setas esquerda e direita do teclado para a direção das rodas do carro, enquanto a tecla **R** restaura o modelo à posição inicial. O movimento da câmera é ajustado por meio do mouse, permitindo rotação do ponto de visão, e o zoom é controlado pelo scroll, tornando a exploração da cena mais intuitiva. Esses comandos garantem maior interação e realismo, complementando a visualização do modelo tridimensional.

De modo geral, o programa combina modelagem tridimensional, animação contínua, aplicação de texturas, manipulação de câmera e interação por teclado, resultando em uma cena coerente, funcional e alinhada aos requisitos do trabalho. A decomposição inicial do veículo em formas geométricas simples, associada à construção modular do código, demonstra domínio dos conceitos fundamentais de computação gráfica e contribui para a clareza, escalabilidade e facilidade de manutenção do projeto.

O código-fonte realizado neste trabalho está disponível publicamente no repositório do próprio autor¹, possibilitando a reprodução.

¹<https://github.com/LucasFortolan/PYOPENGL-TYRELL-007-1974>

CONCLUSÃO

O desenvolvimento do modelo tridimensional do carro, aliado à implementação de uma pista contínua e de um sistema interativo de navegação, permitiu demonstrar de forma prática a aplicação dos princípios fundamentais de computação gráfica. A decomposição do objeto real em estruturas geométricas simples mostrou-se essencial para organizar o processo de modelagem e garantir fidelidade visual ao veículo original. O uso integrado das bibliotecas *pygame*, *OpenGL.GL*, *OpenGL.GLU* e *math* possibilitou tanto a renderização eficiente das primitivas quanto a criação de um ambiente interativo capaz de simular movimentação, controle e visualização dinâmica da cena.

Os comandos inseridos — controle das rodas pelas setas do teclado, redefinição da posição inicial pela tecla **R**, manipulação de câmera pelo mouse e ajuste de zoom via scroll, contribuíram para uma experiência mais completa, permitindo ao usuário explorar o modelo sob diferentes perspectivas. Desse modo, o projeto atendeu aos objetivos propostos ao demonstrar, de forma clara e funcional, como técnicas de modelagem, renderização e interação podem ser combinadas para a construção de um sistema gráfico educacional e progressivamente expansível. Caso o projeto seja continuado, há potencial para aprimoramentos, como a aplicação de texturas, iluminação avançada ou simulação física, expandindo ainda mais sua complexidade e realismo.