TRABALHO 01 - COMPUTACAO GRÁFICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO
UNIVERSIDADE FRANCISCANA – UFN. 2025-01.

PROFESSOR: André F. dos Santos.
Nome do aluno:_____.
Data: ___/___/_____. Peso 3,0.

Parte I- Nesta atividade, você deve responder às perguntas abaixo com base em uma pesquisa sobre OpenGL. Enviar na atividade da aula de hoje no formato pdf, com nome e data preenchidos.

1- O que é o OpenGL?

OpenGL é uma API gráfica que permite a criação de aplicativos gráficos e jogos em 3D. É uma das mais utilizadas na indústria, é independente de sistema operacional e sistema de janelas.

2- Qual é o objetivo principal do OpenGL?

O objetivo principal do OpenGL é permitir a criação de gráficos 2D e 3D em diferentes sistemas operacionais, facilitando a portabilidade entre sistemas operacionais.

3- Quem criou o OpenGL? Qual empresa foi responsável?

A Silicon Graphics, Inc. (SGI) desenvolveu o OpenGL e o lançou em 1992

4- Em que ano o OpenGL foi criado?

O OpenGL foi lançado no ano de 1992.

5- O que significa a sigla “OpenGL”?

OpenGL é uma sigla do nome Open Graphics Library.

6- Quais são os principais recursos oferecidos pelo OpenGL?

O OpenGL nos permite criar imagens complexas a partir de pontos e linhas, gerenciar texturas de objetos, iluminação e escalabilidade, dentre outras funções.

7- Em que linguagens de programação o OpenGL pode ser usado?

O OpenGL pode ser utilizado em C, C++, C#, Java, Python e Lua.

8- O OpenGL é uma API de alto ou baixo nível? Explique.

O OpenGL é uma API de baixo nível, pois permite que você manipule diretamente aspectos detalhados do processo gráfico, como buffers de vértices, shaders e transformações geométricas, entre outros.

9- O OpenGL é multiplataforma? Quais sistemas operacionais suportam?

OpenGL é multiplataforma. Ele suporta sistemas operacionais como Windows, Linux, macOS e também pode ser usado em dispositivos móveis com Android e iOS.

10- Qual a diferença entre o OpenGL clássico e o OpenGL moderno?

A principal diferença é que o OpenGL clássico usa um modelo baseado em state machine e funções de alto nível, enquanto o OpenGL moderno utiliza shaders e um controle mais direto sobre a pipeline gráfica, oferecendo maior flexibilidade

11- Qual foi a maior mudança trazida com a versão 3.0 do OpenGL?

A maior mudança trazida com a versão 3.0 do OpenGL foi a introdução de um modelo de renderização baseado em shaders, eliminando o uso do pipeline fixo (state machine) e tornando o OpenGL mais flexível e eficiente.

12- O que é o OpenGL ES e onde ele é utilizado?

O OpenGL ES é uma versão simplificada e otimizada do OpenGL, projetada para dispositivos com recursos limitados, como smartphones, tablets, dispositivos embarcados e videogames portáteis, e ele é utilizado em plataformas móveis, como Android e iOS.

13- Quais as diferenças entre o OpenGL e o Vulkan?

O OpenGL é uma API de alto nível, oferecendo abstração do hardware e facilidade de uso, enquanto o Vulkan é uma API de baixo nível, proporcionando maior controle sobre o hardware e melhor desempenho com menor overhead. O Vulkan permite um uso mais eficiente de múltiplos núcleos de processador, enquanto o OpenGL tem suporte limitado a multithreading. Por mais que ambos sejam multiplataforma, o Vulkan oferece maior flexibilidade e desempenho, mas exige mais conhecimento técnico, podendo ser mais difícil de lidar dependendo da experiência do usuário.

14- Onde o OpenGL é utilizado no mundo real? Dê exemplos de áreas ou softwares.

O OpenGL é amplamente utilizado em áreas como jogos, onde é usado para renderização gráfica principalmente para PC. Também é utilizado em software gráfico como o Blender, para modelagem e visualização 3D.

15- Quais são algumas vantagens e desvantagens do OpenGL?

O OpenGL oferece várias vantagens, como ser multiplataforma, código aberto, amplamente adotado em indústrias e suporte a gráficos 2D e 3D, também possui boa documentação e recursos de aprendizado. No entanto, apresenta desvantagens, como desempenho inferior em comparação com APIs de baixo nível como Vulkan, complexidade no controle de recursos gráficos e a obsolescência de versões antigas, além de não fornecer uma abstração de alto nível como outros motores gráficos.

16- Existe alguma ferramenta ou biblioteca que facilita o uso do OpenGL? Quais?

Existem várias ferramentas e bibliotecas que facilitam o uso do OpenGL, como GLFW, que auxilia na criação de janelas e no gerenciamento de contexto OpenGL, GLEW, que facilita o uso de extensões OpenGL, GLM, para operações matemáticas 3D, SDL, que oferece suporte a OpenGL em jogos e multimídia, e Assimp, que facilita a importação e exportação de modelos 3D

Parte II

Trabalho 01 – Parte II: Interatividade com Objetos em OpenGL

Objetivo

Implementar, usando PyOpenGL e pygame, um programa com menu de opções no terminal. A cada opção, diferentes objetos devem ser renderizados na tela com controle via teclado. O foco é praticar renderização, movimentação, rotação e interação via teclado.

Instruções Gerais

1- O programa deve exibir um menu no terminal, com pelo menos 6 opções numeradas.

2- Com base na opção escolhida pelo usuário, diferentes objetos devem ser renderizados na janela OpenGL.

3- O usuário deve conseguir movimentar e rotacionar os objetos com as teclas indicadas.

4- Utilize teclas específicas para cada modo de interação.

5- Cada opção deve funcionar de forma independente.

Menu de Opções – O que implementar

Opção 1 – Cubo

- Renderizar um **cubo 3D colorido**.
- Deve permitir movimentação com: ○ W, S → cima / baixo ○ A, D → esquerda / direita ○ Q, E → rotacionar eixo X ○ R, F → rotacionar eixo Y ○ Z, X → zoom in / out

Opção 2 – Triângulo

- Renderizar um **triângulo 2D** com cor definida.
- Deve permitir movimentação com as mesmas teclas da opção 1.

Opção 3 – Cubo + Triângulo

- Renderizar **ambos os objetos lado a lado**.
- Ao movimentar, **ambos devem se mover juntos** com 'WASD', e as mesmas teclas anteriores para rotação 'QERF', e zoom 'ZX'.

Opção 4 – Pirâmide

- Renderizar uma **pirâmide 3D com base quadrada**.
- Movimentação padrão com WASD, QERF, ZX.

Opção 5 – Cubo + Triângulo + Pirâmide

- Renderizar os **três objetos lado a lado**.
- Todos devem **se mover e rotacionar juntos** com:
  - W, S, A, D ○ Q, E, R, F ○ Z, X

Opção 6 – Controle individual

- Renderizar os três objetos, **cada um com controle independente**.
- Teclas de movimentação:
  - **Cubo:**
    I, K → cima/baixo
    J, L → esquerda/direita ○

    Triângulo:

    G, B → cima/baixo
    V, N → esquerda/direita ○

    Pirâmide:

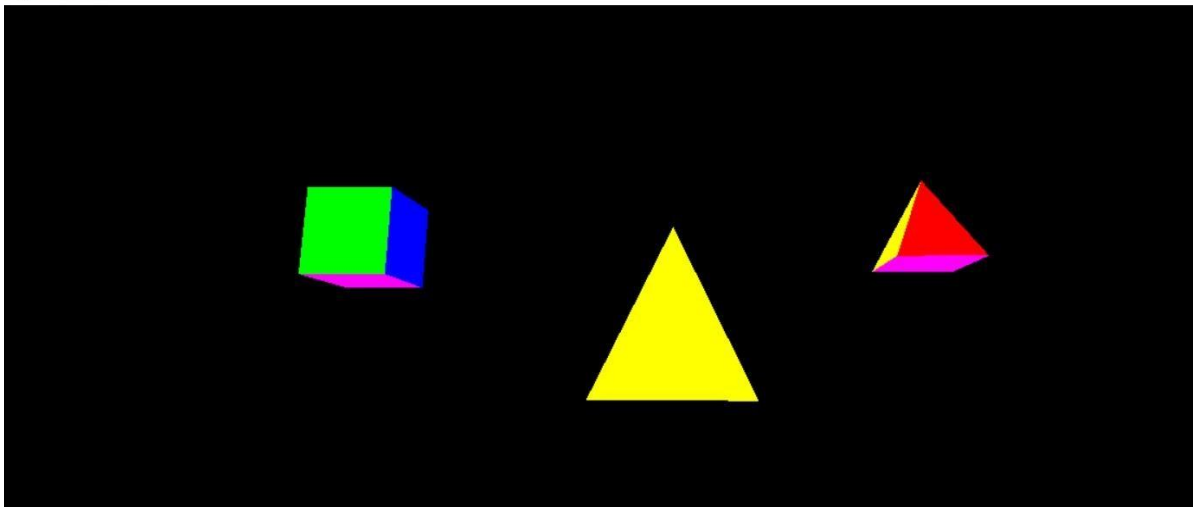
    ↑, ↓ : cima/baixo (direcional do teclado)
    ←, → : esquerda/direita (direcional do teclado) ○ **Zoom e rotação global:** Z, X, Q, E, R, F

Exemplo do Menu em execução:

```
pygame 2.6.1 (SDL 2.28.4, Python 3.12.1)
Hello from the pygame community. https://www.pygame.org/contribute.html
=== MENU DE FORMAS ===
1 - Cubo
2 - Triângulo
3 - Cubo + Triângulo
4 - Apenas Pirâmide
5 - Todos juntos (WASD para todos)
6 - Todos com controle individual
Escolha uma opção: 6
```

Exemplo se opção 6 escolhida (movimentação independente de cada objeto):

pygame window



Avaliação do Professor será ao final da aula quando o aluno estiver pronto para apresentar!!

Logo após submeter as respostas da Parte I e parte II (código) na atividade da aula de hoje. A parte do código pode ser um link do github ou ser enviado em anexo junto.