

Lista de Exercícios 1 – Fundamentos de Computação Gráfica

## Introdução à OpenGL Moderna – Primitivas Gráficas, *Shaders* & *Buffers*

### Entrega

#### Individual

DATA DE ENTREGA: até 03/05/2025, via Moodle

Instruções para envio: link para o repositório contendo o código-fonte e o README.md

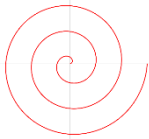
### Parte 1 - Teoria

1. O que é a GLSL? Quais os dois tipos de *shaders* são obrigatórios no pipeline programável da versão atual que trabalhamos em aula e o que eles processam?
2. O que são primitivas gráficas? Como fazemos o armazenamento dos vértices na OpenGL?
3. Explique o que é VBO e VAO, e como se relacionam (se achar mais fácil, pode fazer um gráfico representando a relação entre eles).
4. Como são feitas as chamadas de desenho na OpenGL? Para que servem as primitivas de desenho?
5. Analise o código fonte do projeto Hello Triangle. Localize e relacione os conceitos de *shaders*, VBOs e VAO apresentados até então. Não precisa entregar nada neste exercício.

### Parte 2 - Programação

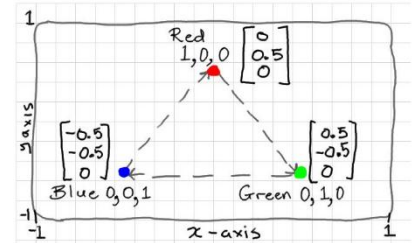
6. Faça o desenho de 2 triângulos na tela. Desenhe eles:
  - a) Apenas com o polígono preenchido
  - b) Apenas com contorno
  - c) Apenas como pontos
  - d) Com as 3 formas de desenho juntas
7. Faça o desenho de um círculo na tela, utilizando a equação paramétrica do círculo para gerar os vértices. Depois disso:
  - a) Desenhe um octágono
  - b) Desenhe um pentágono
  - c) Desenhe um pac-man!
  - d) Desenhe uma fatia de pizza
  - e) DESAFIO: desenhe uma “estrela”

8. Desenhe uma espiral, assim:

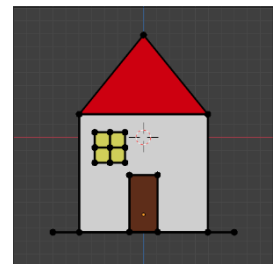


9. Considere um triângulo cujos vértices possuem, respectivamente, as cores vermelho, verde e azul.

- Descreva uma possível configuração dos buffers (VBO, VAO) para representá-lo.
- Como estes atributos seriam identificados no *vertex shader*?
- Agora **implemente**!



10. Faça um desenho em um papel quadriculado (pode ser no computador mesmo) e reproduza-o utilizando primitivas em OpenGL. Neste exercício você **poderá criar mais de um VAO** e fazer **mais de uma chamada de desenho** para poder utilizar primitivas diferentes, se necessário.



## Referências

- <https://learnopengl.com/#!Getting-started/Hello-Triangle>
- <https://learnopengl.com/#!Getting-started/Shaders>
- <https://antongerdelan.net/opengl/hellotriangle.html>
- AKENINE-MÖLLER, Tomas; HAINES, Eric; HOFFMAN, Naty. *Real-Time Rendering*. 4. ed. Natick, MA: A. K. Peters, 2018. Cap. 2 – The Graphics Rendering Pipeline, p. 11–27.
- Playlist sobre OpenGL Moderna: [The Chernobyl – YouTube – OpenGL](#)
- Playlist sobre C++: [The Chernobyl – YouTube – C++](#)

**Dica:** Utilize fontes confiáveis, como sites oficiais das ferramentas, documentação e artigos técnicos. Evite informações desatualizadas ou sem referências.

**BONS ESTUDOS!!!** 😊

"O único jeito de aprender matemática é fazendo matemática." — Paul Halmos