Processamento Gráfico 2022/2 – Lista 1

Matheus Henrique Schmökel

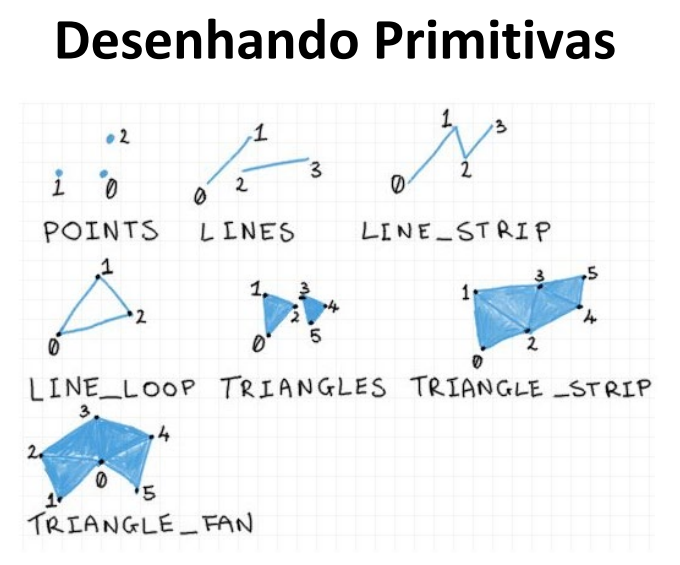
**Lista 1 – Questões teóricas**

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?**

OpenGL Shading Language (GLSL) é uma linguagem de sombreamento de alto nível com uma sintaxe baseada em C. Os dois tipos obrigatórios de shaders para o OpenGL 4 são o Vertex Shader, que descreve como tratar um vértice, e o Fragment Shader, que descreve como tratar uma área.

1. **O que são primitivas gráficas? Como fazemos o armazenamento dos vértices na OpenGL?**

As primitivas básicas são os elementos mais básicos que podem ser criados.



* GL\_POINTS: pontos.
* GL\_LINES: linha a cada 2 pontos do array.
* GL\_LINE\_STRIP: linha entre todos os pontos do array do primeiro ao ultimo.
* GL\_LINE\_LOOP: linha entre todos os pontos do array.
* GL\_TRIANGULOS: triangulo a cada 3 pontos do array.
* GL\_TRIANGLE\_STRIP: triangulos grudados.
* GL\_TRIANGLE\_FAN: triangulo a cada 2 pontos do array sendo que o terceiro ponto é sempre o primeiro do array.

Os vértices podem ser armazenados em um array do tipo float.

1. **Explique o que é VBO, VAO e EBO, e como se relacionam (se achar mais fácil, pode fazer um gráfico representando a relação entre eles).**

Vertex Buffer Object (VBO) é um buffer que armazena um array de dados (posição, vetores, cores etc) diretamente na memória da GPU, permitindo uma renderização mais rápida. Vertex Array Object (VAO) faz a ligação dos atributos de um vértice, ele define que VBO vai ser usado e a localização e o formato desses dados. E por fim, Element Buffer Object (EBO) é um buffer, similar ao VBO, também utilizado para o armazenamento de índices.

1. **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.**

Analisado.