**Lista 1**

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?**

GLSL é a linguagem de programação que usaremos para o desenvolvimento de shaders dentro do pipeline do OpenGL, GLSL vem de GL Shading Language e é baseada em C. Os dois tipos de shaders obrigatórios no pipeline são o Vertex Shader e o Fragment Shader, sendo o primeiro para o tratamento do vértice e o segundo para o tratamento da área.

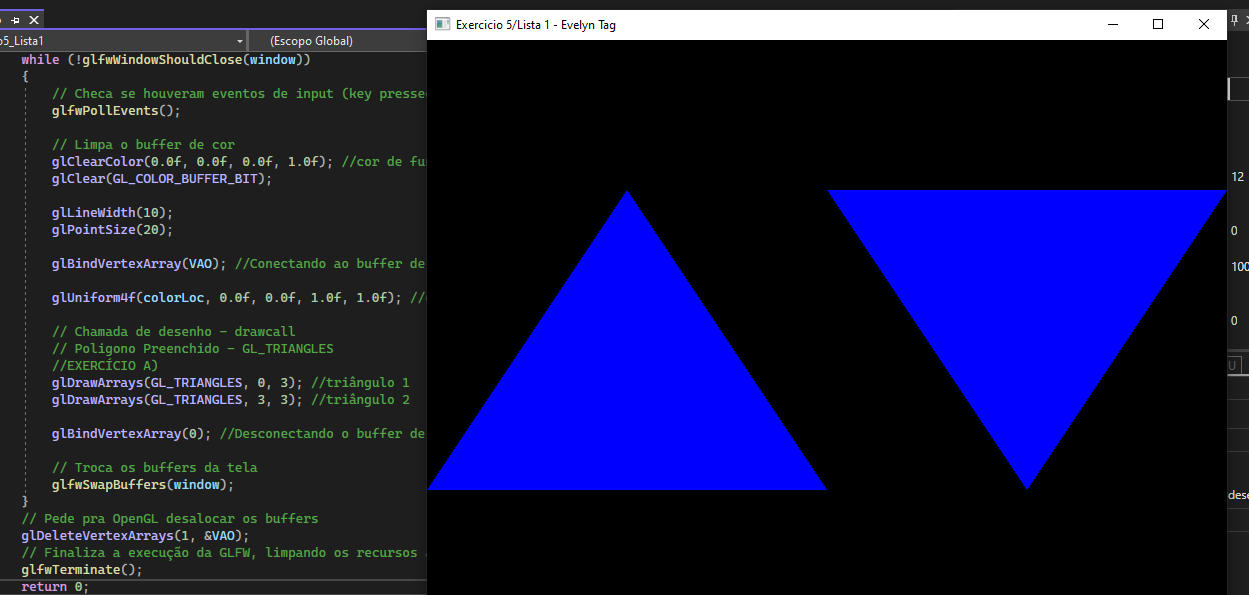
1. **O que são primitivas gráficas? Como fazemos o armazenamento dos vértices na OpenGL?**

São elementos básicos que compõem um desenho no OpenGL contamos com os seguintes: GL\_POINTS, GL\_LINES, GL\_LINE\_STRIP, GL\_LINE\_LOOP, GL\_TRIANGULOS, GL\_TRIANGLE\_STRIP e GL\_TRIANGLE\_FAN. Vertex Buffer Objects (VBO) são usados para armazenar dados de vértices (coordenadas, cores, normais e outros dados relevantes), os quais são necessários para desenhar geometria na tela.

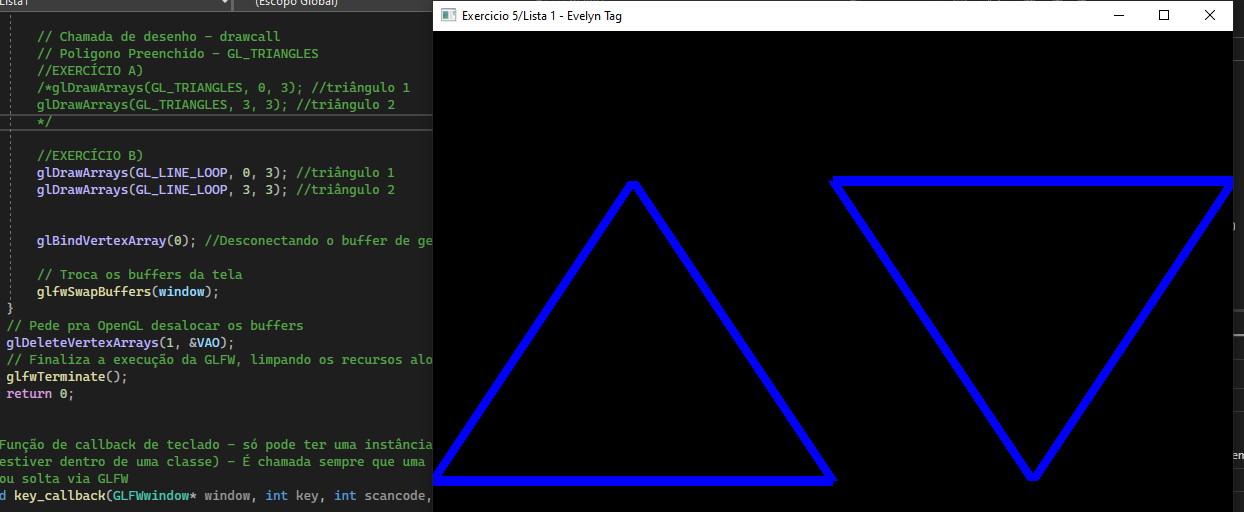
1. **Explique o que é VBO, VAO e EBO, e como se relacionam (se achar mais fácil, pode fazer um gráfico representando a relação entre eles).**

Vertex Buffer Objects (VBO) é um array de dados dos vértices que será enviado a GPU, o Vertex Array Objects (VAO) define qual VBO será utilizado, a localização dos dados do mesmo e o formato dos dados. Já Element Buffer Objects (EBO), associa a ideia de índices e evita a especificação de vértices replicados no VBO.

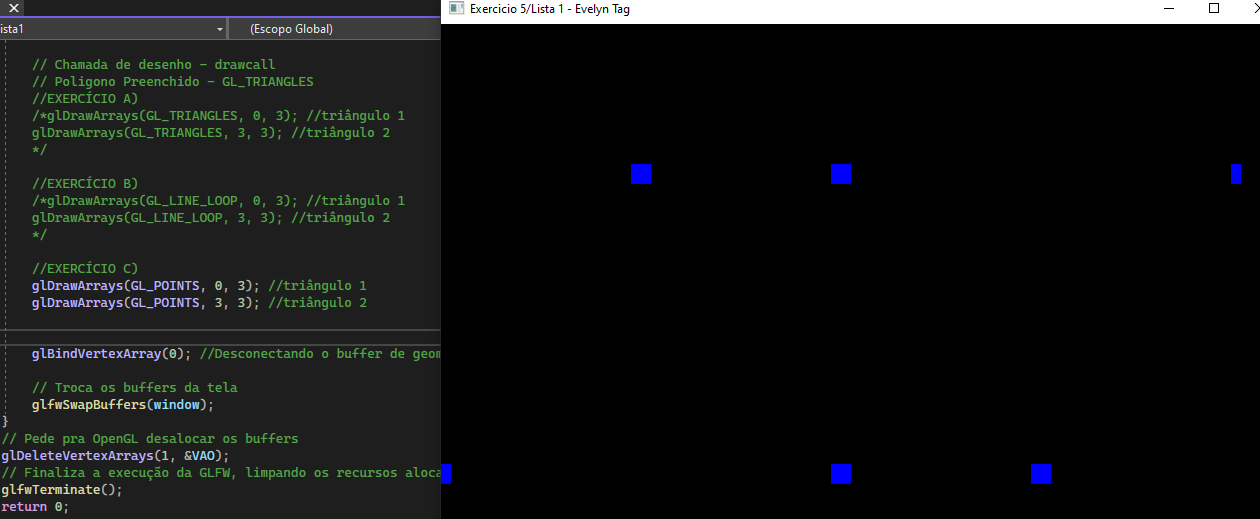
1. **Analise o código fonte do projeto Hello Triangle. Localize e relacione os conceitos de shaders, VBOs e VAO apresentados até então. Não precisa entregar nada neste exercício.**
2. **Faça o desenho de 2 triângulos na tela. Desenhe eles:**
   1. Apenas com o polígono preenchido



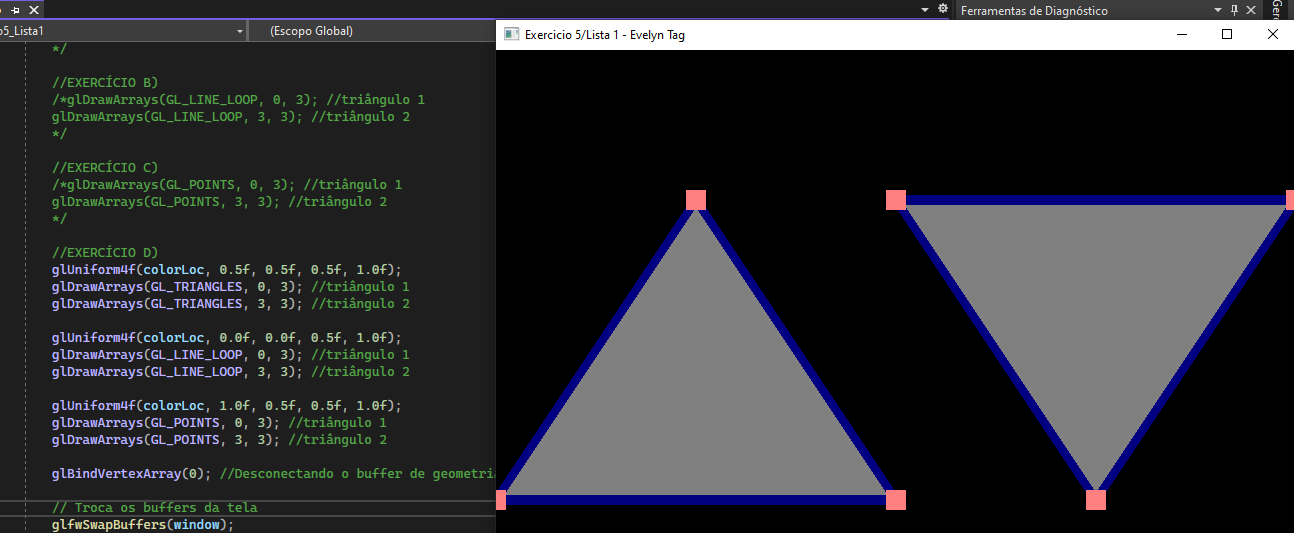
* 1. Apenas com contorno



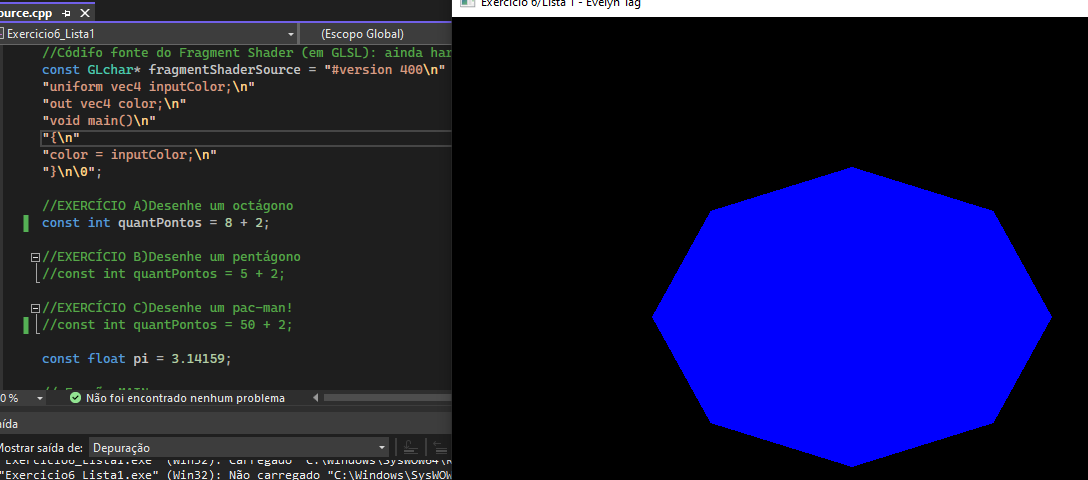
* 1. Apenas como pontos



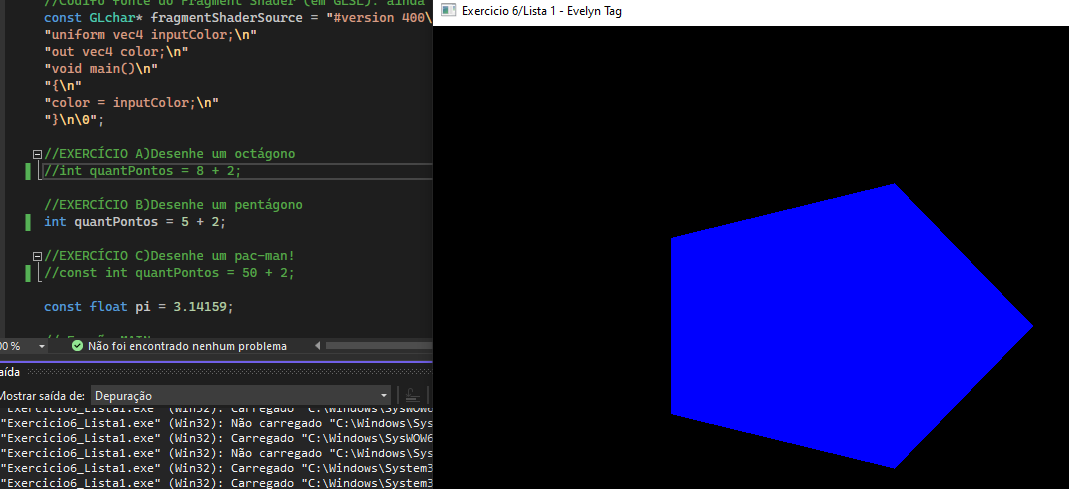
* 1. Com as 3 formas de desenho juntas



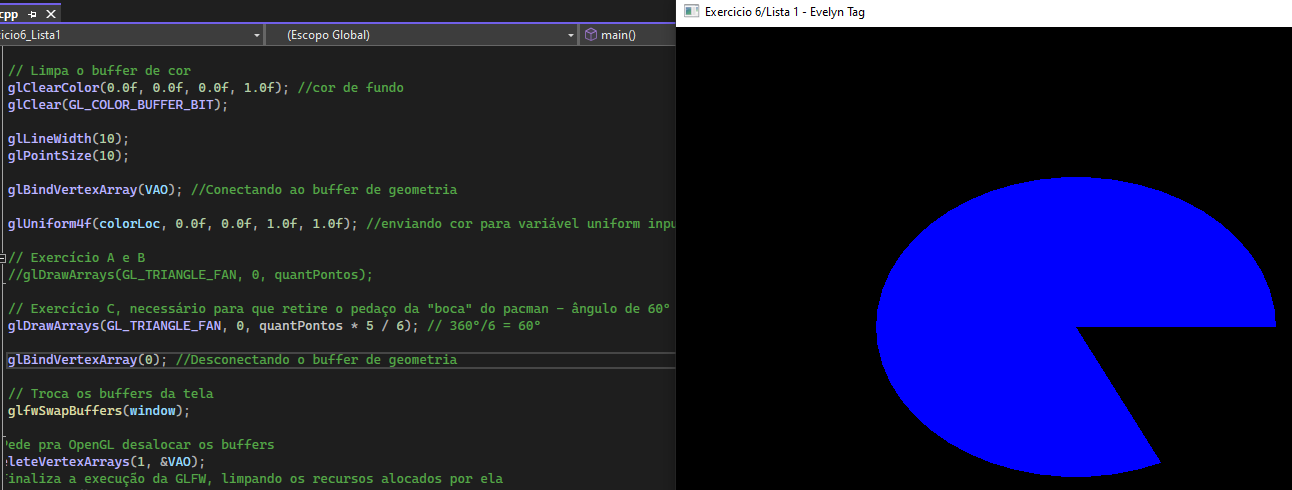
1. **Faça o desenho de um círculo na tela, utilizando a equação paramétrica do círculo para gerar os vértices. Depois disso:**
   1. Desenhe um octágono



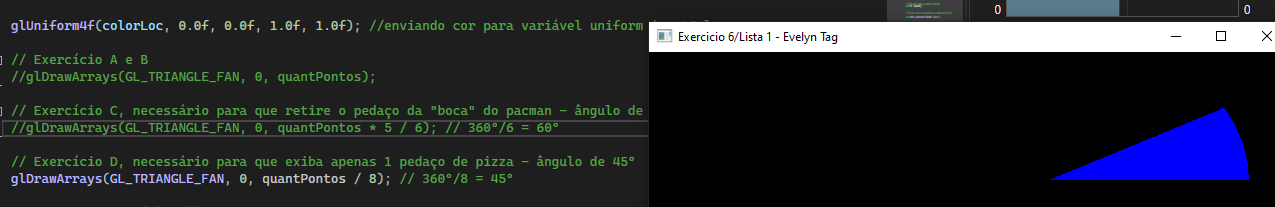
* 1. Desenhe um pentágono



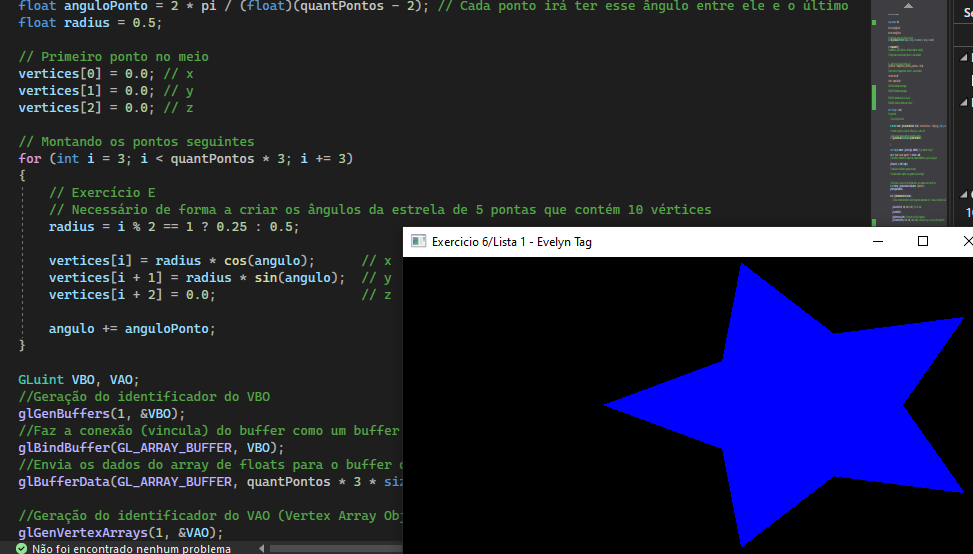
* 1. Desenhe um pac-man!



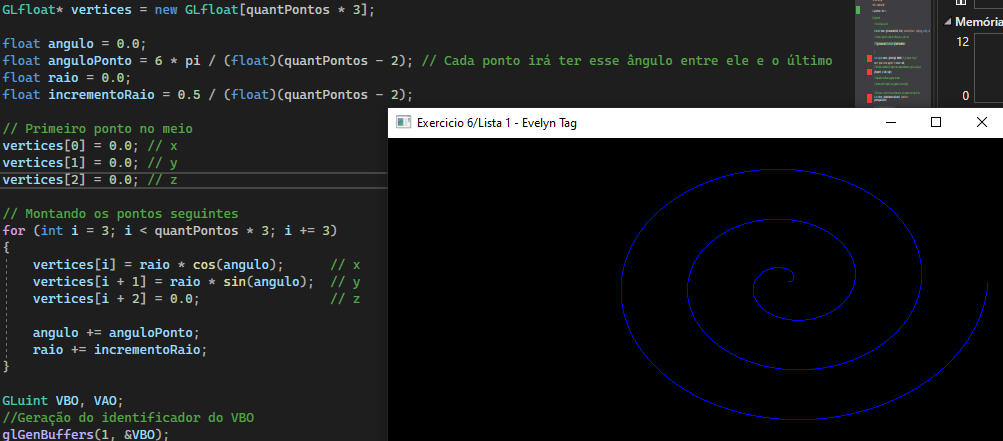
* 1. Desenhe uma fatia de pizza



* 1. DESAFIO: desenhe uma “estrela”



1. **Desenhe uma espiral, assim:**

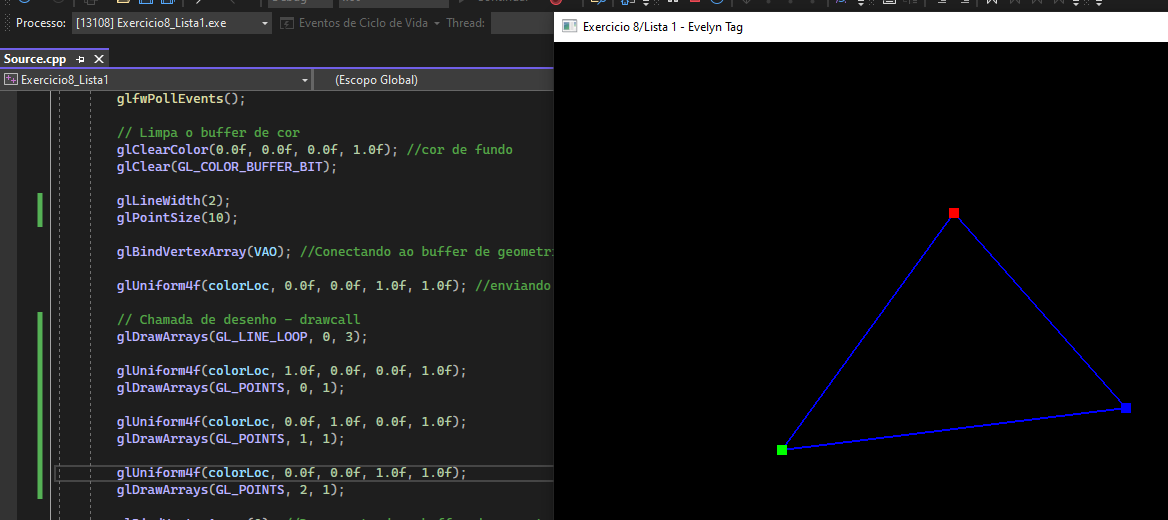


1. **Considerando o seguinte triângulo abaixo, formado pelos vértices P1, P2 e P3, respectivamente com as cores vermelho, verde e azul.**

a.Descreva uma possível configuração dos buffers (VBO, VAO e EBO) para representá-lo.

b. Como estes atributos seriam identificados no vertex shader?

Agora implemente!



1. **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.**

