For a vertex, it does not only contain positions, but also has many other attributes, like texture coordinate, normal, color

How does OpenGL know our data layout? We need to tell OpenGL

Text

Description automatically generated

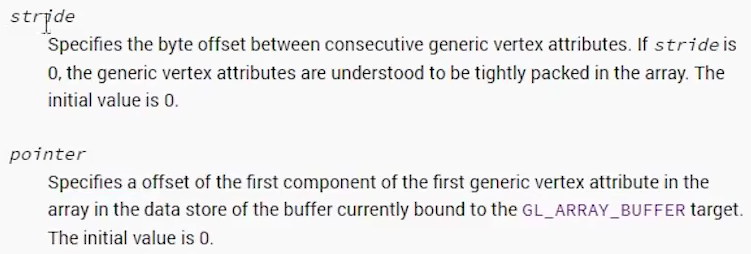
When assign vertex attributes:

Graphical user interface, text, application, email

Description automatically generated

Size parameter has nothing to do with the buffer size. If we have 2 values (x coordinate and y coordinate) for each point, we need to set size = 2. Instead, in 3D coordinate, we would like to set size = 3

For normalize, we don’t need to consider this parameter so far, as out points are between 0 and 1. But for color parameter range from 0 to 255, we need to normalize it. We can do it on CPU side, but you can tell GPU to do this if you are lazy.



Stride: the number of bytes between each vertex. Suppose we have position, texture coordinate, and normal. Position is a 3-component vector, texture coordinate is 2, normal is 3. They will be (3 + 2 + 3) \* sizeof(float) = 32

Pointer: the offset. Take the previous example. Let’s say we want to define normal. Before normal there are 5 component which have 20 bytes. So the pointer would be 20. You can define macros to avoid use number each time

And before drawing, we still need to set the vertex is enable

Text

Description automatically generated

Now we get it

A picture containing shape

Description automatically generated

But we still not provide a shader program. Some GPU has default shader programs. That is based on your GPU driver

If we draw a line instead of a triangle

A picture containing graphical user interface

Description automatically generated