Document: <https://docs.gl/>

Before drawing triangle, we need to create a buffer in GPU and put our data into it. We need to tell GPU how to interpret that data and show it on the screen in shader program.

GPU is like a state machine, you just need to set a series of state. In stead of draw a triangle directly, you need to select a buffer, a shader, and say draw a triangle

So, we will need to place our buffer code outside the while loop, because we only need to set the buffer once.

Text

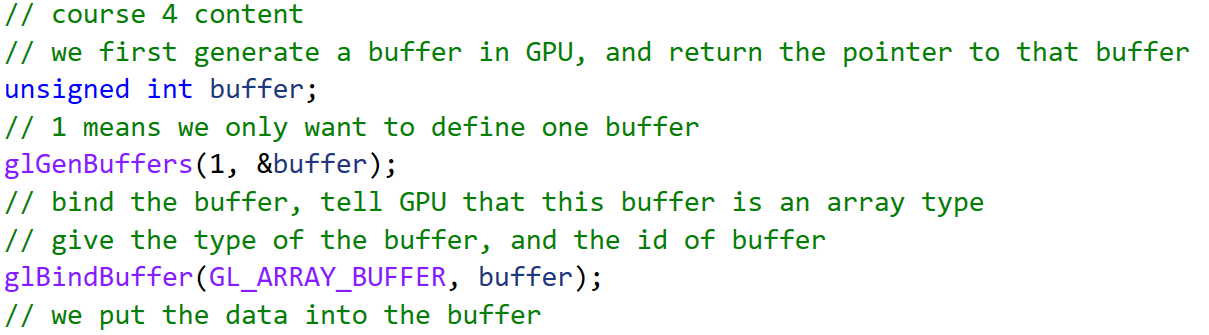
Description automatically generated

Add a sleep to prevent it keep calculating

Text

Description automatically generated

Define buffer



Remember, OpenGL is contextual based. Here we state bind buffer to “buffer”, so all of the operation below would be on “buffer”, and we don’t need to pass “buffer” as a parameter

Data type:

Graphical user interface, text, application, email

Description automatically generated

Stream: we don’t use too much

Static: we define it once and use it many times.

Dynamic: we will modify it

And for the suffix, we have:

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text

Description automatically generated

Now we don’t have a shader to tell GPU how to draw this buffer, we will cover it in the next lecture.

We draw the triangle in the while loop

Graphical user interface, text, application

Description automatically generated

We have such parameters:

Graphical user interface, text, application, email

Description automatically generated

We still have another function to draw triangle:

Graphical user interface, text, application

Description automatically generated

Now we still can’t draw a triangle since we only finish 30% of work.

OpenGL is a state machine, you tell OpenGL to draw the triangle, and actually the buffer it draw is the buffer you bind before.