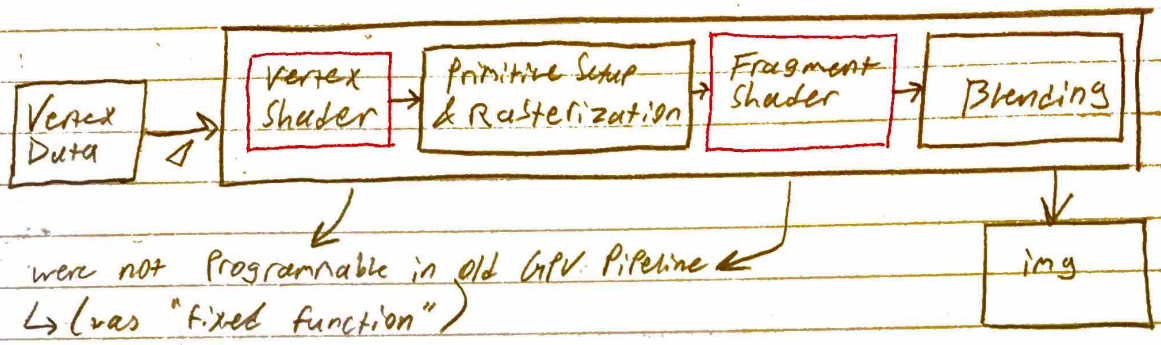
L05 - Intro to Modern OpenGL

GPU Pipeline



Vertex Shader

(from model view)

- Responsible for transforming vertices into the canonical view volume
- each vertex shader execution sees 1 vertex at a time, but 3 executions take place at once to process a whole triangle at once

eg.

GLSL

```
#version 330 core
layout(location = 0) in vec3 Pos;
uniform mat4 mvp;
void main() {
    gl_Position = mvp * vec4(Pos, 1);
}
```

Fragment Shader

- determines the color for each fragment/pixel of the triangle

eg.

GLSL

```
#version 330 core
layout(location = 0) out vec4 color;
void main() {
    color = (1, 0, 0, 1); // red
}
```

Rasterization

- generates pixels / fragments of each triangle → Rasterization Algs

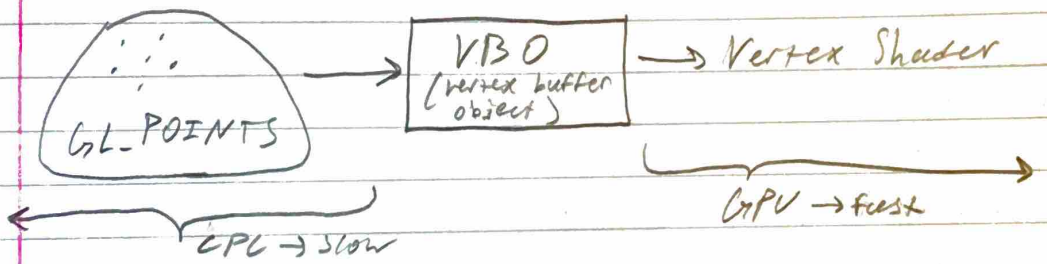
CPU Stuff

- the CPU compiles the shaders @ runtime (using openGL) so that the shaders are optimized for our particular hardware

Compile Shaders

< Code @ 37:13 >

Vertex Buffer Object




```
float positions[] = { f, f, f,
                      f, f, f, ... etc }

GLuint buffer;
glGenBuffers(1, &buffer)
glBindBuffer(GL_ARRAY_BUFFER, buffer)   → after this, all GLs will use 'buffer'
glBufferData(GL_ARRAY_BUFFER,
             sizeof(positions),
             positions,
             GL_STATIC_DRAW);
```

↳ how to tell OpenGL how to interpret the data in buffer for drawing?

↳ how to actually send to vertex shader? →

will get the location of "pos" from vertex shader
↳ we set this to zero

```
GLuint pos = glGetAttribLocation(program, "pos");
glEnableVertexAttribArray(pos);   → indicate that 'pos' is where the data to draw is.
glVertexAttribPointer(pos,        → start at pos
                      3,          } → each item has 3 floats
                      GL_FLOAT    }
                      0,
                      (GLvoid*)0);
```

uses 'buffer' since we binded it previously

Rendering

```
glClear
glUseProgram(program);
glDrawArrays(GL_POINTS,   → draw points
             0,           → starting at 0
             num_vertices); → # points
glSwapBuffers();
```

Vertex Array Object

- Stores connection between Vertex Buffer Objects (VBOs) & vertex attributes

<more @ 1:04:03>
↳ w/ code