

Homework 3 - Readings Ch5

1. What is the difference between using multiple buffers to store vertex data vs. interleaving vertex data in one buffer?

Using multiple buffer is best for handling small amounts of data, but more difficult to manage with multiple complex 3d structures. Interleaving vertex data is the process in which in one array stores all necessary info in one array and "skips" around the array to get the necessary info.

2. Describe the two process that take place between the vertex and fragment shaders.

The process for the vertex shader is the following:

1. The vertex shader is first invoked where then the buffer object is passed to the attribute variable, and once assigned to position the first coordinate is communicated to the geometric shape assembler.
2. This process is repeated for the rest of the vertices
3. After all vertices are added, geometric shape assembly starts and depending on the type (eg. `gl.POINTS`, `gl.TRIANGLES`, `gl.LINES`, `gl.TRIANGLE_STRIP`) it will then decide how the primitives will be assembled

Lastly it will start rasterization

The process for the fragment shade is that once rasterization ends the fragment shader will be invoked to process each of the generated fragments. Where all the fragments are fed one by one to the fragment shader, and for each fragment, the fragment shader sets the color and writes its output to the color buffer. Once the last fragment shader is completed the final output is displayed.

3. In the context of the fragment shader, describe how varying variables can be used to interpolate data among fragments.

When using varying variables, each vertex is assigned its own color, and the color is interpolated between vertices. This is different from uniform variables where each vertex is assigned the same color. Varying variables allow for per-vertex data to be interpolated across the surface of a primitive.

4. In the context of texture mapping, what is a magnification method? Enumerate and explain different methods.

The magnification method magnifies a texture image when you map the texture to a shape whose drawing area is larger than the size of the texture. It is called under `gl.TEXTURE_MAG_FILTER` which has the default value of `gl.LINEAR` that uses the weighted average of four texels that are nearest the center of the pixel, and `gl.NEAREST` which calculates the value of the texel that is nearest the center of the pixel being textured.

5. In the context of texture mapping, what is a minification method? Enumerate and explain different methods.

The minification method miniaturizes a texture image when you map the texture to a shape whose drawing area is smaller than the size of the texture. It called under `gl.TEXTURE_MIN_FILTER` which has the default value of `gl.NEAREST` which calculates the value of the texel that is nearest the center of the pixel being textured `gl.LINEAR` that uses the weighted average of four texel that are nearest the center of the pixel.