WebGL Pipeline

The WebGL or graphics pipeline is a serious of steps that must be taken to create images on our displays [3]. The pipeline is a process of converting vertices into pixels and it accomplishes this goal by taking the data through a series of steps including; vertex processor, clipper and primitive assembler, rasterizer, and fragment processor [1]. These steps all take place in the GPU and rely on data that is passed to the GPU from the CPU [2]. Both the vertex processor and fragment processor steps of the pipeline can be programmer controlled through the use of shaders [1]. These shaders which are made by programmers, give us direct control over exactly what we want to happen with the data in the vertex processor and fragment processor. The vertex shader is called and executed for each vertex and it calculates the position of each vertex and puts it into a varying called gl\_position [3]. Then, the clipper and primitive assembler draws the triangle objects made by the vertices and clips any parts of the objects which will not be seen on our display [2 & 3]. Then the rasterizer comes into play. The rasterizer takes the triangles and vertices and uses them to determine which pixels in our rasterizer (or display) should be turned on [1 & 3]. This essentially converts our vertices that we have been dealing with up until now into the pixels that the actual display or monitor uses to display and make images. The last step is the fragment processor. This step utilizes a fragment shader that, again, programmers are able to directly write code for and control. The fragment shader runs on each pixel and determine things such as color for each pixel [1 & 3]. Both the vertex and fragment shaders require input and output data. The vertex and fragment shaders both can use textures and uniforms as input data, but the vertex shader can also use attributes as input data and the fragment shader can also use varyings as input data [1]. Varyings are a way of passing data from the vertex shader to the fragment shaders [1]. Through the WebGL pipeline, the graphics and images we see on our screens everyday our able to come to life and be realized. Each step of the pipeline is an integral piece of the puzzle and serves its role to make our images a reality.

Sources:

1. Class lecture slides (Feb. 9 and 11)
2. <https://runestone.academy/runestone/books/published/learnwebgl2/01_the_big_picture/3_3d_graphics_pipeline.html>
3. <https://www.tutorialspoint.com/webgl/webgl_graphics_pipeline.htm>