The 3D graphics pipeline is an assembly line of sorts. The pipeline first starts with taking in a vector graphics representation of objects and finishes with a 2D raster image, before the image can be displayed the vector graphics representation of objects goes through a few processes. The first step of the pipeline is the input of shader attributes and shader uniforms from the CPU. This data is passed down the pipeline to the vertex shader in the GPU were the data is transformed into normalized device coordinates. In the vertex shader step the coordinates are positioned in front of a virtual camera and then projected onto a unit cube. After the Vertex shader has completed and sent the information down the line the Clipping step occurs. The Clippings step consist of cutting away all the data the resides outside of the virtual camera prepared in the step previous. After the short step of Clipping has occurred the Viewport Transformation maps all of the model data from the normalized device coordinates, created in the Vertex shader step, into a viewport defined by pixels. After all of the previous steps have been completed the computer will finally start the process of making the imagines. In the rasterization step the GPU determines which pixels are going to be used in the outline of the shape. The rasterization step turns the data into the pixel outlines that will be utilized by the fragment shader. Once the rasterization step is done the Fragment Shader uses the outline from the step previous and composes a geometric primitive and a color value for each pixel.

Once we have the geometric primitive and the colors of each pixel, they need to be combined with the colors of the pixels already assigned to the output draw buffer. After the compositing step the image is ready to be displayed and used. Take special note that this pipeline will mainly occur in the Graphics Processing Unit using GPU memory. The only part of this pipeline that takes places in the Central Processing Unit is the is the primary step of data setup. After this step the GPU is the sole organizer of the data from the Vertex Shader all the way until the image is displayed on the appropriate monitor.

Sources:

https://www.hebergementwebs.com/webgl-tutorial/webgl-graphic-pipeline

pcmag.com/encyclopedia/term/graphics-pipeline

https://runestone.academy/runestone/books/published/learnwebgl2/01\_the\_big\_picture/3\_3d\_graphics\_pipeline.html