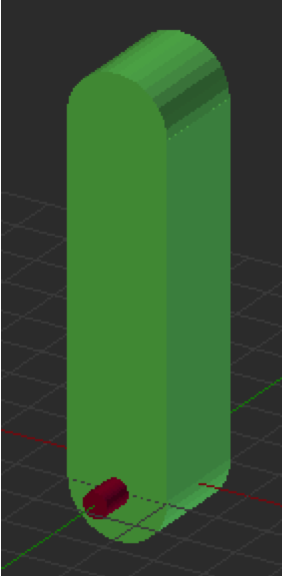
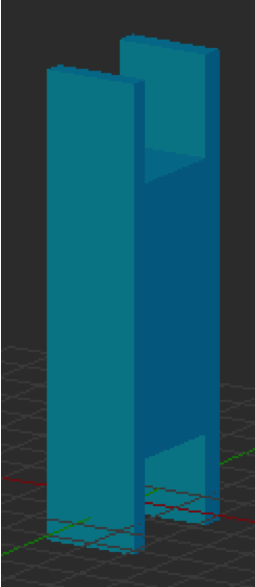
The green axis (Z axis)

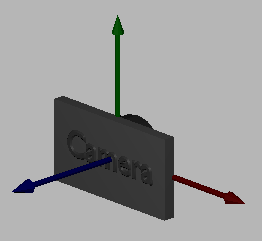
will be the rotation axis. The pivot point is at the origin.



The green axis (Z axis)

will be the rotation axis. The pivot point is at the origin.





The graphic pipeline’s

***fragment shader*** processes

the fragments used to

represent the triangle,

one at a time.

The graphic pipeline’s

***vertex shader*** processes

the triangle’s vertices,

one at a time.

Rendering Diagrams

RAM

JavaScript

Program

Uniform  
Variables

GPU

Shader

Program

VOB\_1

VOB\_2

VOB\_3

Render Buffer

(Output image)

1. Copy **uniform** values to the shader program.
2. Attach a vertex object buffer to the shader program for the **attribute** values.
3. Execute the shader program on the data: **gl.drawArrays()**
4. Graphics pipeline outputs colors to the rendered image.

1.

2.

3.

4.

Vertex Object Buffers (VOB’s)

arrays\_to\_gpu.js

ModelArraysGpu()

ModelArrays

object

ply\_to\_arrays.js

CreateModelsFromPLY ()

\*.PLY file

text description of a model

\*.OBJ file

text description of a model

obj\_to\_arrays.js

CreateModelsFromOBJ ()

Scene pre-processing sends the model arrays to the GPU.

initializeRendering()

converts text to ModelArrays objects.

SceneDownload() downloads the models from the server.

Rendering is performed by a rendering function that uses a GpuModelArray object and a shader program.

GpuModelArrays()

object

shader program

object