In essence, the rendering engine is quite straightforward. It follows the patterns of basic OpenGL applications, but it separates components out, making things more flexible and adaptable. It strives to add a layer of abstraction, providing a common-sense framework through which models and scenery can be set up with high-level commands. The gameplay engine operates under a similar convention: it strives to make things as straightforward and as easy as possible. The code is designed under software design principles, most notably a lack of inter-dependencies that make development more difficult and bugs more likely.

Game

- **Voxels**
  - Chunk – no implementation at the moment
  - Cube – has type, has mesh, normal, and program caches, is Model
- Game – singleton, has 1 Scene and 1 Player
- Player – has pointer to Scene

Modeling

- **DataBuffers**
  - **Sampled Buffers**
    - SampledBuffer – has image and coordinate map, is an OptionalDataBuffer
    - TextureBuffer – has GLSL attribute for texture, is an SampledBuffer
    - BumpMap – no implementation right now, is a SampledBuffer
    - CoordinateMapReader – no data, implements a method
  - DataBuffer – declares methods, is a ShaderUtilizer
  - VertexBuffer – has vector of vertices, is a DataBuffer
  - IndexBuffer – has vector triangles, is a DataBuffer
  - OptionalDataBuffer – categorical class, is a DataBuffer
  - NormalBuffer – has vector of vertex normals, is an OptionalDataBuffer
- **Mesh**
  - Mesh – must have VertexBuffer, can have IndexBuffer, is DataBuffer
  - PlyParser – no data, implements collection of methods
- **Shading**
  - Program – namespace 5400, has Gluint handle
  - Shader – namespace 5400, has GLuint handle
  - ShaderManager – no data, implements collection of methods
  - ShaderSnippet – has fields, methods, and main body code strings
  - ShaderUtilizer – no data other than two method declarations
- Model – has 1 Mesh, has zero or more OptionalDataBuffers, has 1 Program, has 1 model matrix, has 1 rendering mode.
- Triangle – has a, b, and c indexes.

World

- Camera – has look direction, position, up vector, FOV, near/far clip, and aspect ratio. Is a ShaderUtilizer
- Light – has position, color, and power. Is a ShaderUtilizer
- Scene – has zero or more Models, has zero or more Lights, has 1 Camera, has ambient light