**Engine Subsystems**

Entries highlighted in green have been completed. Entries in yellow are incomplete; non-highlighted entries have not been worked on.

1. **Platform Independence**
   1. Platform Detection (Win32 for now)
   2. Datatypes
   3. File System
   4. Timer
   5. Threading System
   6. Graphics Wrapper (using OpenGL for now)
   7. Physics System/Wrapper
2. **Core**
   1. Memory Alloc.
   2. Module Startup & Shutdown
   3. Assertion System
   4. Debug Logging
   5. RNG
   6. Math Libraries
   7. Strings & Hashed Strings
   8. Engine Configuration
   9. Profiler/Stat Gatherer
   10. Object Handles & IDs
   11. Curves & Surfaces
   12. Optional Subsystems
       * *Reflection & Serialization*
       * *Localization*
       * *Async File I/O*
       * *Movie Player/Replay System*
       * *Debug Menu/Console*
3. **Resource Manager**
   1. Resource Importers
      * *Models*
      * *Textures*
      * *Materials*
      * *Fonts*

(3.1.1-4 have importing systems, but their import procedures are still subject to change.)

* + - *Colliders*
    - *Physics Params*
    - *Maps*
    - *Skeletons?*

1. **Input**
   1. Device I/O
   2. I/O Wrapper
2. **Collision Detection/Physics**

(Collision detection is currently handled by Bullet; it seems to work, but it hasn't been tested in a very active environment.)

* 1. Forces & Constraints
  2. Raycasting, maybe Shapecasting
  3. Rigid bodies
  4. Collision bodies
  5. Phantoms
  6. Physics/Collision world

1. **Low-Level Renderer**
   1. Materials & Shaders

(Materials exist as part of the Material class, which records diffuse, specular, and emissive properties of a mesh's material. Shaders are handled by OpenGL.)

* 1. Lighting

(6.2 has a Gouraud shader that supports point lights.)

* 1. Camera
  2. Primitive Submission System
  3. Viewport/Virtual Screen
  4. Texture & Surface Management
  5. Text & Fonts
  6. Debug Primitives

(Cylinders and toruses don't have a drawing function, and since 6.4 is not implemented the debug calls force the system to switch to the debug shader and setup matrices every time a debug primitive needs to be drawn. Not a major concern, however, since 6.4 needs to be implemented along with Section 9 anyway.)

* 1. Device Interface

1. **Profiler & Debug Utilities**
   1. Memory & Perf. Stats.
2. **Audio**
   1. Playback / Manager
   2. DSP & Effects (distortion, low priority)
   3. 3D Audio
3. **Scene Graph**
   1. Space Subdivision (OctTree?)
   2. *LOD*
   3. *Occlusion*

(9.2-3 are pretty low priority at the moment.)

1. **Front End**
   1. HUD
   2. Game GUI & Menu Screens
   3. Attract Mode?
2. **Gameplay Base**
   1. Game Flow System
   2. Scripting
      * Incomplete, see “Scripting Notes”.
   3. Static World
   4. Dynamic Object Model (Actor and Components)
   5. Agent System (AI hooks)
   6. Event / Messaging System
   7. World Loading

Right now, I'm working on the bug in Item 6.7. Next in priority is Item 9.1, then 6.4 so that we can render a large number of objects. After 6.4 is 8.1 and 8.3; I'm not sure about what audio library should be used, as free libraries are either unreliable or pretty platform-dependent. Next is 11.2, followed by 10.1-2, and finally the rest of Section 11. If all of those are finished, we can consider making a small demo that uses the engine.