A layered approach

- **Intelligence** is spread across layers, with higher concentration at the top and lower concentration in the graphics hardware at the bottom.
- **Interfaces** between layers provide scripting opportunities, sharing of components, different control paths and levels of detail, as well as being the glue holding a character together.
<table>
<thead>
<tr>
<th>1. Rendering Engine</th>
<th>Rendering of graphics primitives onto the screen</th>
</tr>
</thead>
<tbody>
<tr>
<td>2. Scene Manager</td>
<td>Representation and transformation of world geometry and sounds</td>
</tr>
<tr>
<td>3. Body Model</td>
<td>Articulated skeleton or armature, inverse kinematics, constraints</td>
</tr>
<tr>
<td>4. Motion Engine</td>
<td>Motion controllers and motion script executors, meaningful motion primitives</td>
</tr>
<tr>
<td>5. Behavior Engine</td>
<td>Mapping from functional description to behavior description and behavior script executor</td>
</tr>
<tr>
<td>6. Functional Engine</td>
<td>Task / Gameplay Behavior Planning, Gameplay Rules</td>
</tr>
<tr>
<td>7. Existential Engine</td>
<td>Character goals, drives and deliberative high level action selection</td>
</tr>
<tr>
<td>8. Avatar Engine (and Interface)</td>
<td>Player character interface and action selection</td>
</tr>
<tr>
<td>9. Scenario Engine</td>
<td>Game and scenario directives from game designers</td>
</tr>
</tbody>
</table>

**INTERFACE:**
- **Standard 3D Rendering API** (DirectX, OpenGL)
- **Standard Scene Graph API** (Open Scene Graph, Panda 3D)
- **Standard 3D Body Specification** (H-Anim, various animation tools)
- **Behavior Markup Language** (New)
- **Low-Level Function Markup Language** (New)
- **High-Level Function Markup Language + State Description**
- **Game Director Scripting**
1. Rendering Engine
Rendering of graphics primitives onto the screen
INTERFACE: Standard 3D Rendering API (DirectX, OpenGL)

2. Scene Manager
Representation and transformation of world geometry and sounds
INTERFACE: Standard Scene Graph API (Open Scene Graph, Panda 3D)

3. Body Model
Articulated skeleton or armature, inverse kinematics, constraints
INTERFACE: Standard 3D Body Specification (H-Anim, various animation tools)

4. Motion Engine
Motion controllers and motion script executors, meaningful motion primitives
INTERFACE: Behavior Markup Language (New)

5. Behavior Engine
Mapping from functional description to behavior description and behavior script executor
INTERFACE: Low-Level Function Markup Language (New)

6. Functional Engine
Social / Interpersonal Social Interaction Planning, Social Rules
Task / Gameplay Behavior Planning, Gameplay Rules
(And other specialized functionality)
INTERFACE: High-Level Function Markup Language + State Description

7. Existential Engine
Character goals, drives and deliberative high level action selection

8. Avatar Engine (or Interface)
Player character interface and action selection
INTERFACE: Game Director Scripting

9. Scenario Engine
Game and scenario directives from game designers

HELPFUL: Sensors and Arbitrators
The body always knows where it is and what it is doing, it shouldn’t do anything dumb!

Crucial: We never stop moving!
The motion engine needs to make sure the body is in constant motion or risk breaking the illusion.
1. Rendering Engine
Rendering of graphics primitives onto the screen
INTERFACE: Standard 3D Rendering API (DirectX, OpenGL)

2. Scene Manager
Representation and transformation of world geometry and sounds
INTERFACE: Standard Scene Graph API (Open Scene Graph, Panda 3D)

3. Body Model
Articulated skeleton or armature, inverse kinematics, constraints
INTERFACE: Standard 3D Body Specification (HAnim, various animation tools)

4. Motion Engine
Motion controllers and motion script executors, meaningful motion primitives
INTERFACE: Behavior Markup Language (New)

5. Behavior Engine
Mapping from functional description to behavior description and behavior script executor
INTERFACE: High-Level Function Markup Language + State Descriptions

6. Functional Engine
Social / Interpersonal Social Interaction Planning, Social Rules
Task / Gameplay Behavior Planning, Gameplay Rules
(And other specialized functionality)
INTERFACE: High-Level Function Markup Language + State Descriptions

7. Existential Engine
Character goals, drives and deliberative high level action selection

8. Avatar Engine (and Interface)
Player character interface and action selection
INTERFACE: Game Director Scripting

9. Scenario Engine
Game and scenario directives from game designers

**Note:** Many ways to do the same thing!
The behavior engine can provide many different behavior sequences to achieve the same function.
1. Rendering Engine
Rendering of graphics primitives onto the screen
INTERFACE: Standard 3D Rendering API (DirectX, OpenGL)

2. Scene Manager
Representation and transformation of world geometry and sounds
INTERFACE: Standard Scene Graph API (Open Scene Graph, Panda 3D)

3. Body Model
Articulated skeleton or armature, inverse kinematics, constraints
INTERFACE: Standard 3D Body Specification (H-Anim, various animation tools)

4. Motion Engine
Motion controllers and motion script executors, meaningful motion primitives
INTERFACE: Behavior Markup Language (New)

5. Behavior Engine
Mapping from functional description to behavior description and behavior script executor
INTERFACE: Low-Level Function Markup Language (New)

6. Functional Engine
Social/Interpersonal Social Interaction Planning, Social Rules
Task/Gameplay Behavior Planning, Gameplay Rules
(And other specialized functionality)
INTERFACE: High-Level Function Markup Language + State Description

7. Existential Engine
Character goals, drives and deliberative high level action selection

8. Avatar Engine (and Interface)
Player character interface and action selection
INTERFACE: Game Director Scripting

9. Scenario Engine
Game and scenario directives from game designers

**Important:** Re-use, re-use, re-use!
Think of this as a fairly standard skill set that most people (and characters) have.

Opportunity: Individualization
A description of personality, mood, attitudes and a personal history can influence all layers below through tweaking parameters.
<table>
<thead>
<tr>
<th>1. Rendering Engine</th>
<th>Rendering of graphics primitives onto the screen</th>
</tr>
</thead>
<tbody>
<tr>
<td>2. Scene Manager</td>
<td>Representation and transformation of world geometry and sounds</td>
</tr>
<tr>
<td>3. Body Model</td>
<td>Articulated skeleton or armature, inverse kinematics, constraints</td>
</tr>
<tr>
<td>4. Motion Engine</td>
<td>Motion control and motion script execution, meaningful motion primitives</td>
</tr>
<tr>
<td>5. Behavior Engine</td>
<td>Mapping from functional description to behavior description and behavior script execution</td>
</tr>
<tr>
<td>7. Existential Engine</td>
<td>Character goals, drives and deliberative high level action selection</td>
</tr>
<tr>
<td>8. Avatar Engine (and Interface)</td>
<td>Player character interface and action selection</td>
</tr>
<tr>
<td>9. Scenario Engine</td>
<td>Game and scenario directives from game designers</td>
</tr>
</tbody>
</table>

**Goal:** The right level of control

More degrees of freedom does not mean better control!