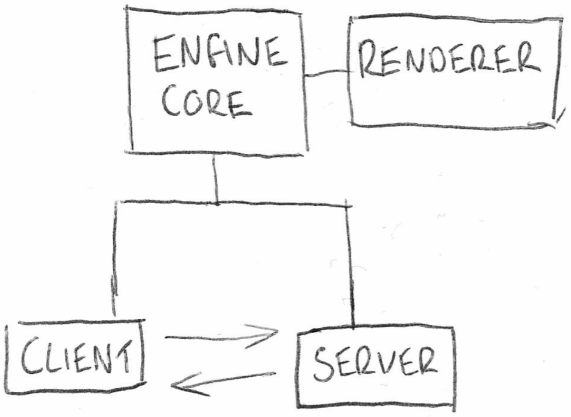
METAGINE ARCHITECTURE PROTOTYPES

and other STUFF

JAMES MUNRO
7th Dec 2008

At a glance...



Potential folder hierarchy:

/ –

Engine/	– Core functionality.
Renderer/	– Rendering system.
Client/	– Client game code.
Server/	– Server game code.
Public/	– Public class interfaces.
Tools/	– Tools code (if any)

Features of the engine core:

- Thread pooling and management.
- Virtual filesystem and asset management.
- Console system (Quake-style).
- Managing of game modules. AND COMMUNICATION BETWEEN THEM.
- Potentially a GUI system.
- User input.

Game modules should get passed a simple interface which facilitates access to other interfaces as required on a per-module basis.

Aims and objectives:

- Well documented.
- Avoid complication. AKA Keep it simple :)
- Rely on threads where possible.
- Cross-platform, but Linux as main development target.
- Clean code + design.
- Support development with personal reflections on LL blog.
- Priority of openmedia formats (OGG etc...)

Tools and Libraries:

- SDL for graphics + audio. (and potentially threading)
- Investigate third-party threading libs, such as Intel's "Thread Building Blocks".
- Doxygen for documentation generation.

* Sound functionality may be implemented as self-contained module, but more likely included in core engine.