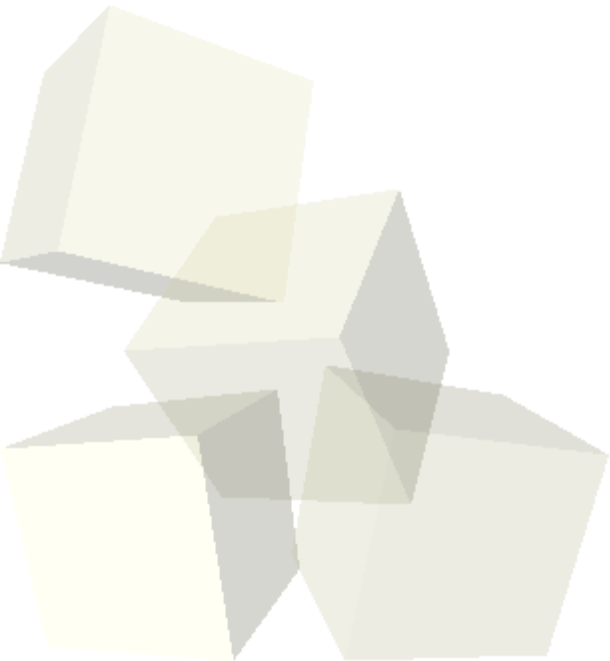




Platform independent engine for volume visualization

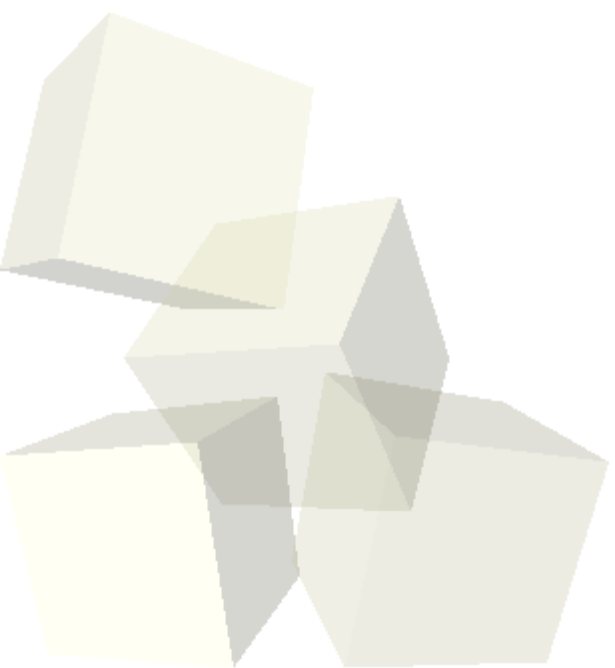


Michal Hučko





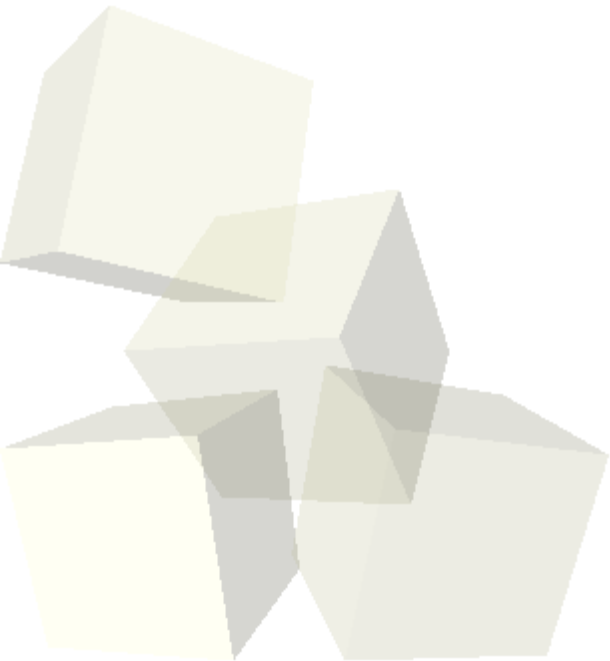
- Requirements
- Existing solutions
- Proposed architecture
- Usage scenarios





Requirements

- engine
 - ◆ UI-less application
 - ◆ defines interface
 - ◆ enables various methods of visualization
- platform independency
- based on scene graph





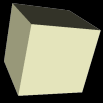
- Requirements
- Existing solutions
- Proposed architecture
- Usage scenarios





- volume rendering extension for the OpenSG scene graph
 - ◆ single node that can be used in any OpenSG application
 - ◆ provides texture based (2D/3D) volume visualization
 - ◆ ability to use shaders
 - ◆ bricking

Klein, Weiler, Ertl (2003) – A volume rendering extension for the OpenSG scene graph API



- framework for remote 3D-visualization
 - ◆ applications based on OpenInventor or Cosmo3D can be adapted for remote versions (using the framework)
 - ◆ client events are transferred to adapted application using CORBA interface
 - ◆ results are sent back via sockets
 - ◆ client modules in Java
 - ◆ possibility to work on server and broadcast rendered images to multiple observing clients
 - ◆ uses loss-less compression for final images
 - ◆ fast enough even on 56k modem connections

Engel, Sommer, Ertl (2000) – A framework for interactive hardware accelerated remote 3D-visualization



- Requirements
- Existing solutions
- **Proposed architecture**
- Usage scenarios





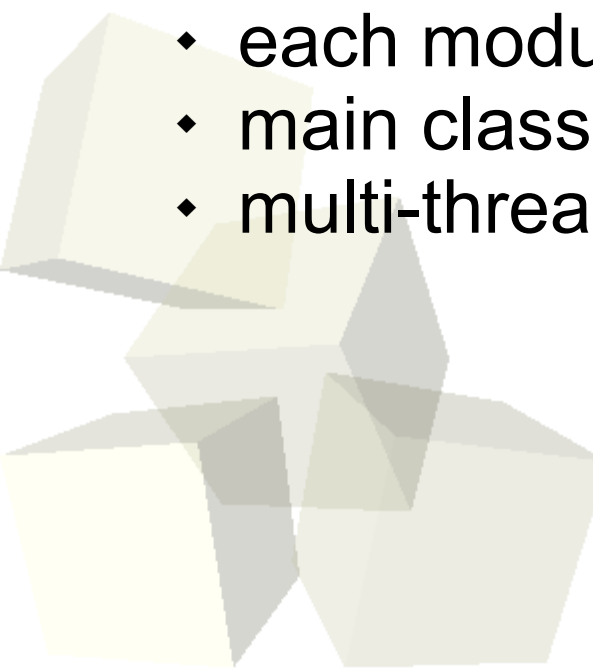
Proposed architecture

■ additional requirements

- ◆ possibility to act as a server (multiple autonomous clients)
- ◆ expandibility of the engine with new rendering methods without need to change engine's code
- ◆ ability to specify various special rendering parameters

■ solution

- ◆ engine modules responsible for various functions
- ◆ each module is highly configurable
- ◆ main class managing modules
- ◆ multi-threading





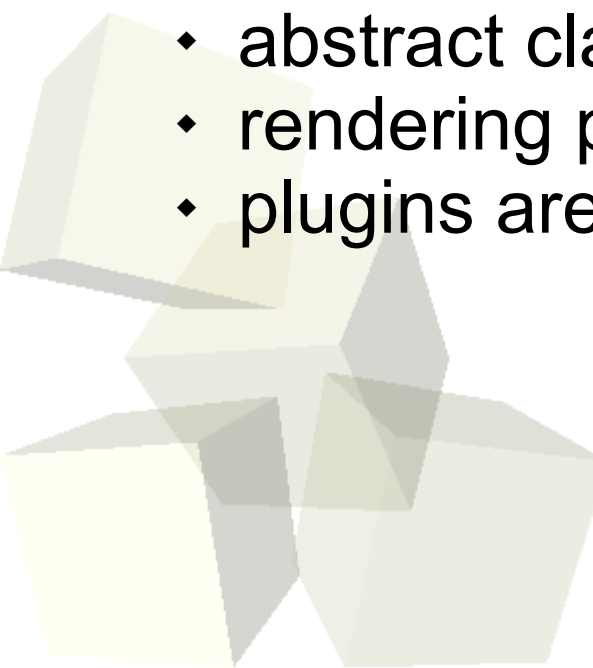
Proposed architecture

■ data manager

- ◆ responsible for loading, storing and freeing data
- ◆ ability to load data into textures (2D/3D) or into main memory (plugin can choose)
- ◆ textures are in OpenSceneGraph object
- ◆ takes care of bricking

■ plugin manager

- ◆ plugins represents rendering methods
- ◆ abstract class defining interface
- ◆ rendering plugin = overriding class
- ◆ plugins are stored in dynamically linked libraries

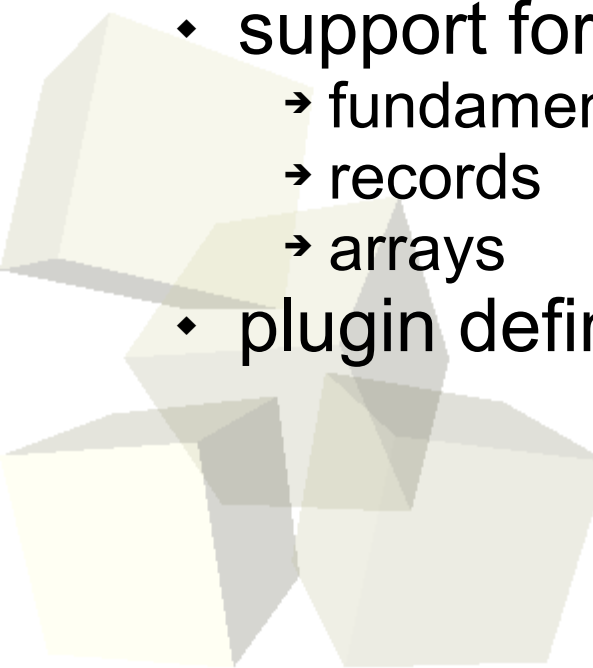


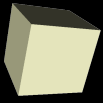


Proposed architecture

■ scene manager

- ◆ stores configuration of scene plus rendering parameters
- ◆ submodules for common scene elements (camera position, clipping geometry, etc.)
- ◆ submodules for common visualization algorithm parameters (transfer function, threshold, etc.)
- ◆ plugin can choose which submodules are to be used
- ◆ support for extra parameters
 - fundamental types (integer, string, etc.)
 - records
 - arrays
- ◆ plugin defines which extra parameters it needs

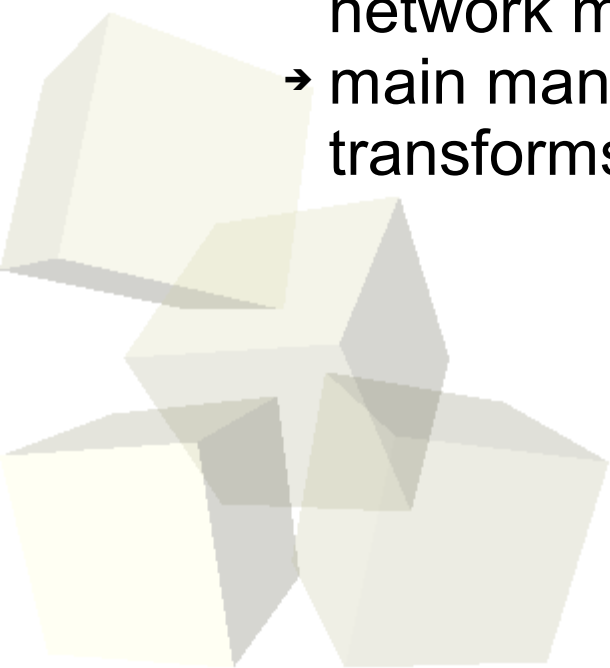




Proposed architecture

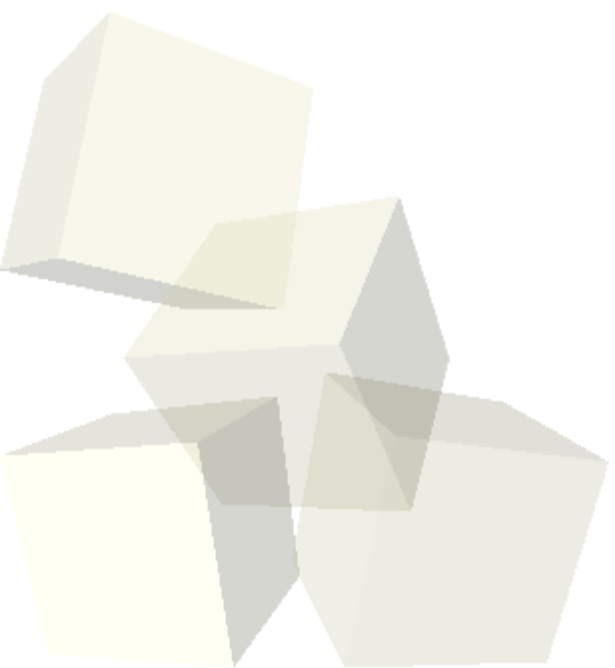
■ main manager

- ◆ integrates previously mentioned managers
- ◆ defines interface through which user communicates with engine
- ◆ used as a main engine class when including engine into project
- ◆ when engine is used as a server
 - proxy class on client side – transforms function calls to network messages
 - main manager preceded by communication class – transforms network messages to function calls





- Requirements
- Existing solutions
- Proposed architecture
- Usage scenarios





- Single application
 - ◆ engine statically linked into GUI
 - ◆ highest efficiency, need to recompile application with engine update
- Local client/server
 - ◆ engine running as a server (locally)
 - ◆ client on the same machine communicates with the server via network (loopback)
 - ◆ high efficiency, full separation of GUI and rendering engine, possible lower memory consumption
- Remote client/server
 - ◆ engine running as a server on dedicated hardware
 - ◆ clients communicates with server via network
 - ◆ efficiency highly dependent on client and server hw, network bandwidth and server load



Thank you for your attention

