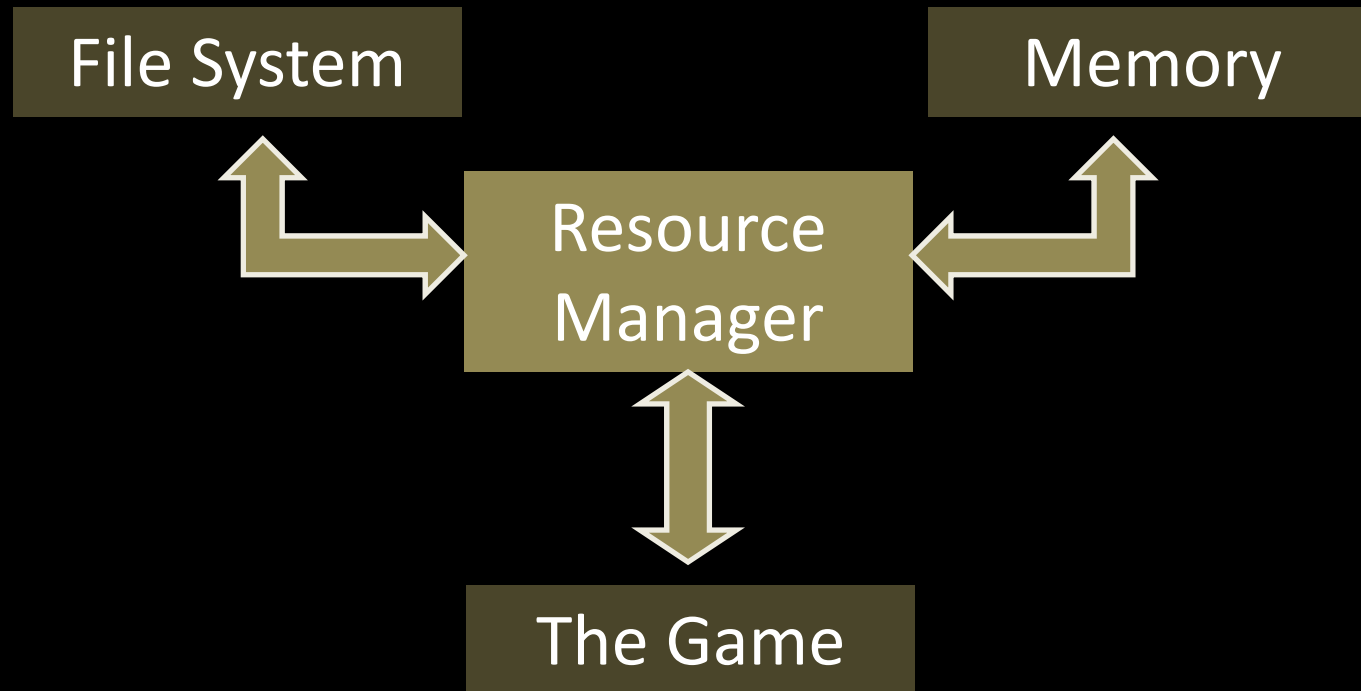




Resources and the File System

hannes@ru.is

Overview



OS Path Differences

- „/“ vs. „\“
- Volume specifier („C:“ or „\\“ in MS)
- File extensions (e.g. „.exe“ in MS)
- Certain characters not allowed
- Current Working Dir (CWD) (one per volume in MS)
- Current Working Volume (only MS)
- Certain predefined special volumes (e.g. „/dev_bdvd“ on PS3)

Absolute vs. Relative Path

- Absolute

C:\Windows\System32

D:\

\

\games\assets\animations\walk.anim

\\joe-dell\Shared-Files\Images\foo.jpg

/usr/local/bin/grep

/game/src/audio/effects.cpp

/

Absolute vs. Relative Path

- Relative
 - System32
 - x:animation\walk.anim
 - bin/grep
 - src/audio/effects.cpp

Search Path

- Tells a system where to find certain files, for example:
 - Executables (system search path)
 - Resources (in Ogre3D resources.cfg)
- Not a particularly efficient way to manage files. Preferably only search and find once at beginning.

Path API

- Tool for interpreting and manipulating paths
- System specific, e.g. `shlwapi.dll` on Windows
- Better to write your own
 - To isolate engine from platform
 - You may not need the full system specific API

Basic File I/O

- Buffers
 - Required
 - Needed to mediate between program and disk
- Standard C library provides two APIs
 - Buffered (streamed functions)
 - Unbuffered (manage your own for efficiency)

Basic File I/O

- System specific File I/O APIs always available
- Wrap them yourself
 - To isolate file system from engine
 - You may not need the full system API
 - You may need extra features

Basic File I/O

- Synchronicity of I/O functions
 - Synchronous: Block program while executing
 - Asynchronous: Program continues while executing
- Both standard C libraries (buffered and unbuffered) are synchronous

Basic File I/O

- Asynchronous File I/O, i.e. non-blocking
 - Allows data to be streamed, i.e. loaded in the background
 - Good for large chunks, i.e. audio, textures and whole levels
 - Can provide a seamless play experience
 - Implemented with threads
- Additional functionality
 - Wait calls (block after doing something else), deadlines, priorities

Resource Manager

- Manages the whole of your asset creation process off-line, but also the on-line use of those assets
- Sometimes a single unified system, but also sometimes a collection of sub-systems

Off-Line Resource Manager

- Need to consider revision control issues
 - Binary files
 - Large files
- Commercial and home-made solutions
 - Alienbrain
 - Use UNIX network file system and symbolic links

Off-Line Resource Manager

- Resource Database (metadata on all assets)
 - Multiple filetypes
 - Create, delete, inspect, modify
 - Relocate
 - Cross-reference
 - Referential integrity
 - Revision history
 - Search / Query

Off-Line Resource Manager

- Unreal Ed
 - + In-game engine
 - + One place for all
 - + Validation
 - Large binary packages
 - Dummy remapping objects
- ND uses MySQL + GUI

Asset Conditioning Pipeline (tool chain)

