Asset Management for JVM Applications

Hawk Weisman
weismanm@allegheny.edu
http://hawkweisman.me

Department of Computer Science
Allegheny College

February 19, 2015

Asset Management

- Application software has to manage a diverse range of assets
  - Graphics
  - Sound
  - Natives
- Challenges
  - Assets stored in zip/jar archives
  - Modularity and overwriting
  - Consistency across platforms
  - Security

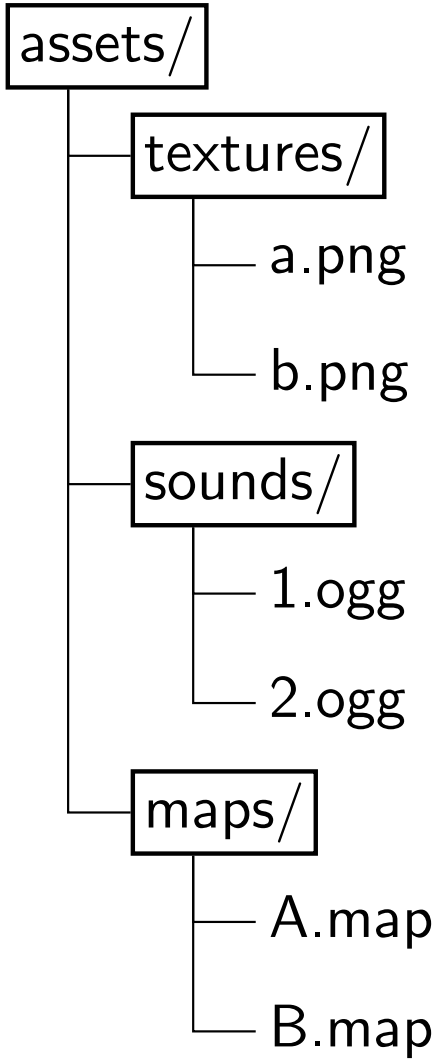
The Virtual File System

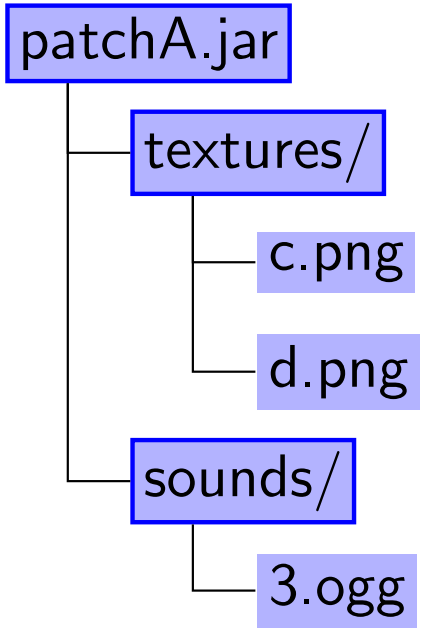
Idea

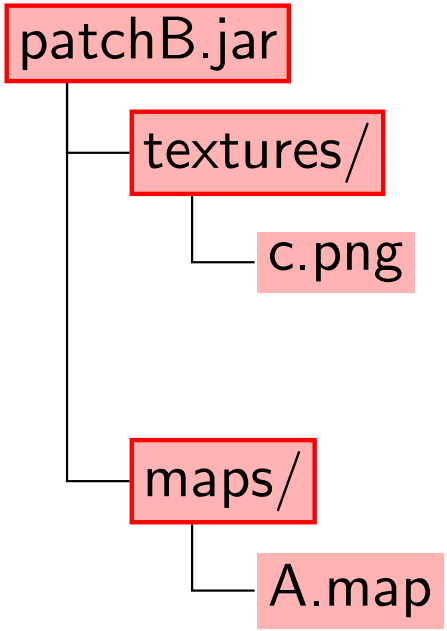
Construct a virtual file system that fuses multiple directories into a new file system tree.

- Security
  - Disallow reads/writes outside of VFS root (/)
- Modularity
  - Patches can be 'spliced in' by overwriting existing files
  - Overwriting controlled through a load order policy
- Compatibility
  - Archives mounted as directories
  - Platforms with multiple content roots (i.e. Android) fused into one directory tree
  - VFS handles OS or platform-specific details

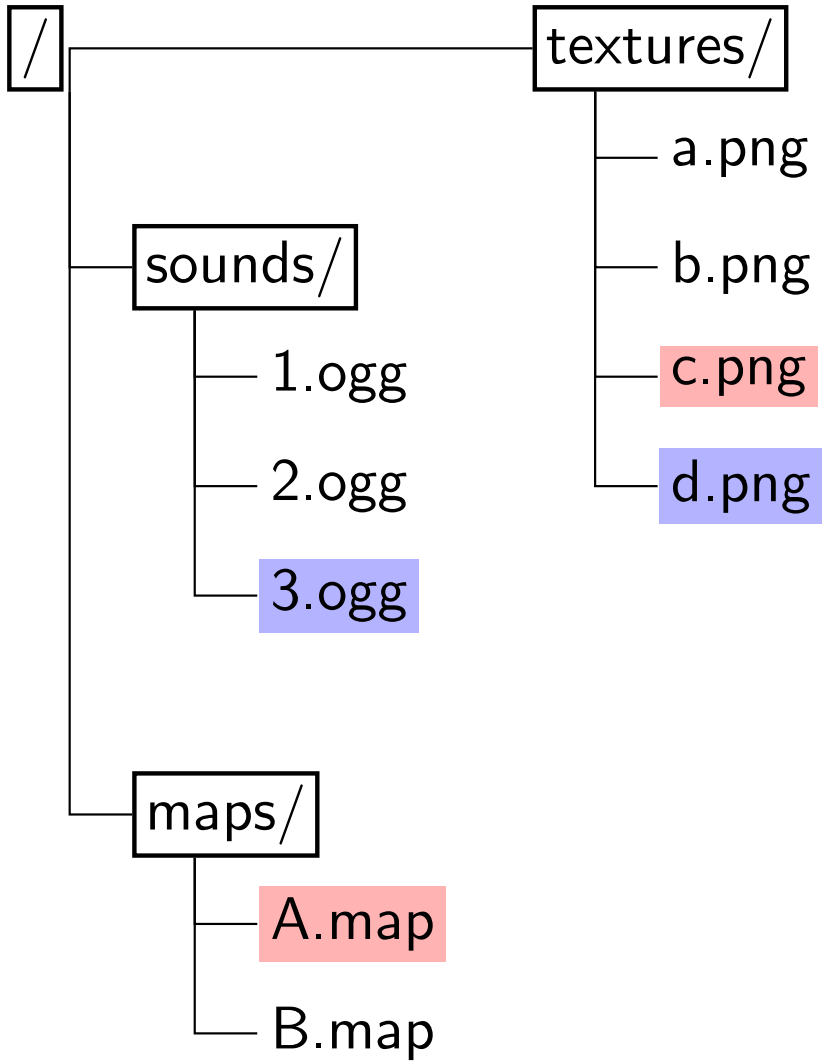
Example: Before Melding







Example: After Melding



Previous Work

- UnionFS
  - `http://unionfs.filesystems.org`
  - Kernel-level extension for POSIX operating systems
  - Multiple open-source implementations
  - C
- Pathway Resource Manager
  - `https://github.com/MeteorCode/Pathway`
  - Component of the Pathway game engine
  - Implemented by Hawk Weisman and Max Clive at MeteorCode Labs
  - Scala/Java

Proposal

- Expand Pathway resource manager to a standalone library
  - Improve core algorithm and architecture
  - Use Java Security API to disallow writes outside of virtual file system
  - Extend functionality to handle additional platforms, media types, natives
  - Add idiomatic Scala functional API
- Evaluation
  - Integration testing with edge cases
  - Testing with ‘malicious’ code to ensure security
  - Release as open-source library and collect user feedback