

Doom 3 Bfg Concrete Architecture

Basement Gurus

Isaac Chan

Daniel Elmer

Matthew Sherar

Yohanna Gadelrab

Dylan Liu

Kainoa Llyod

Overview

1. Introduction
2. Derivation Process
3. Updated Conceptual Architecture
4. Concrete Architecture
 - ◇ Dependencies
 - ◇ Reflexion Analysis
5. Alternatives
6. Game Logic Subsystem
7. Sequence Diagram
8. Concurrency
9. Limitations
10. Lessons Learned
11. Summary

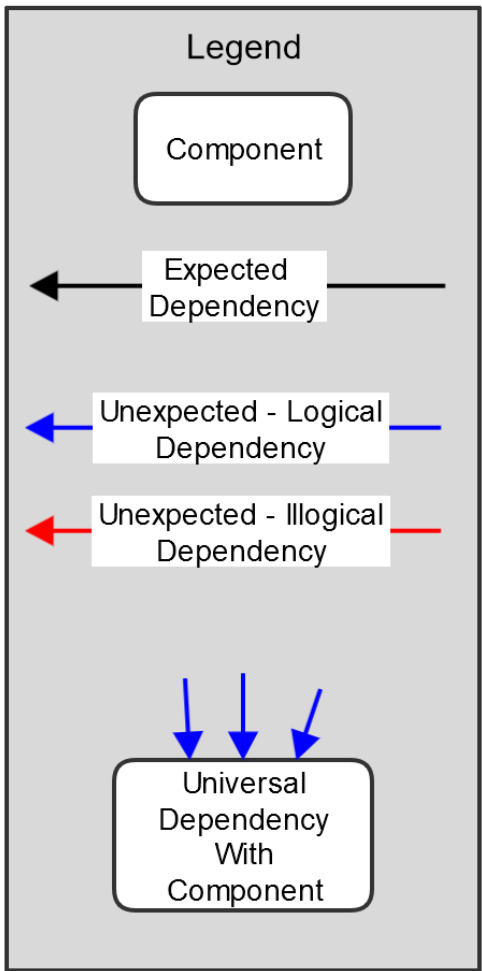
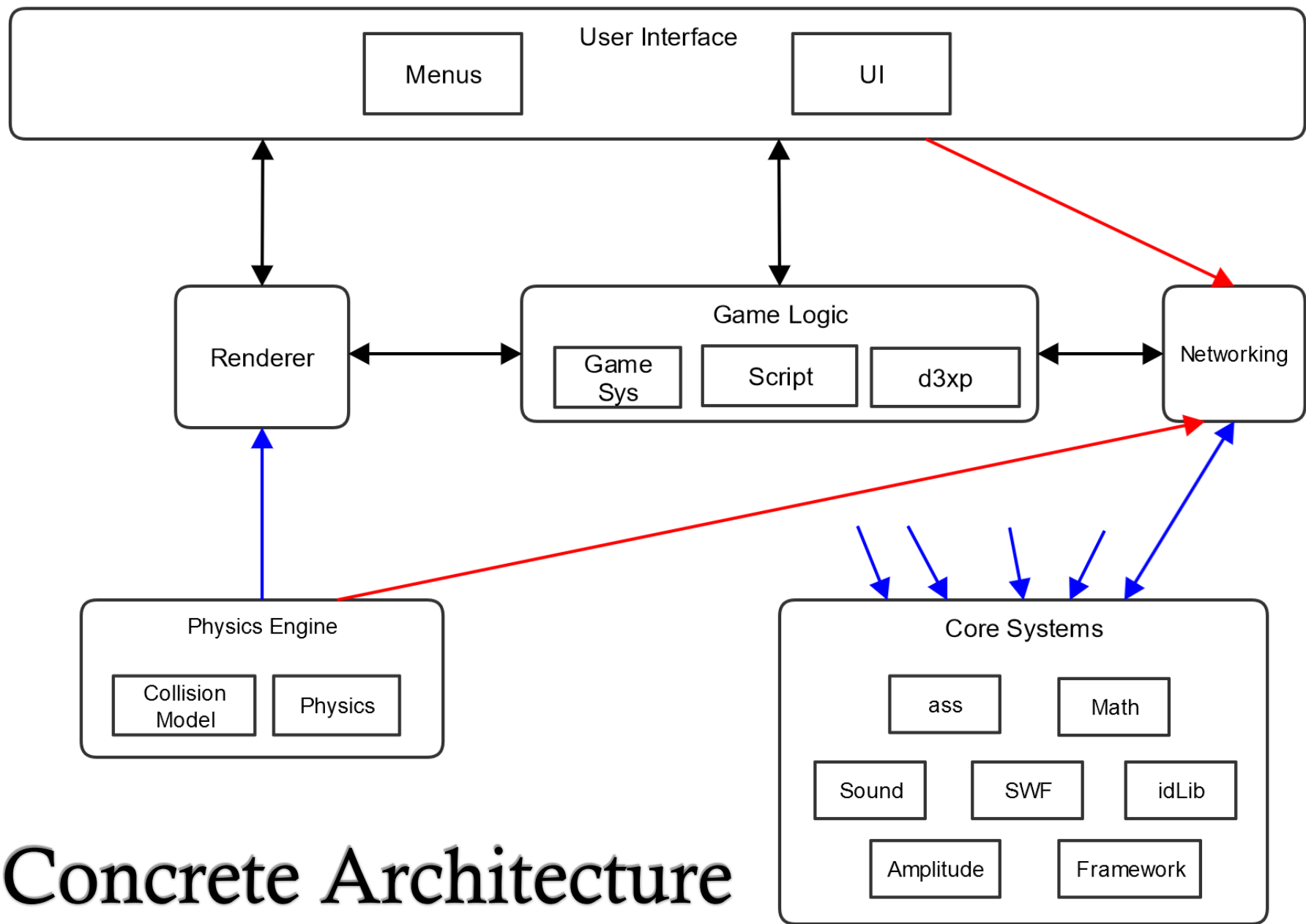
Introduction

- ◆ Concrete Architecture
 - ◆ Object Oriented Layered Architecture
 - ◆ Subsystem: Game Logic Subsystem

Derivation Process

- ◆ Conceptual Architecture as a basis
- ◆ Analysis of source code using Understand
- ◆ Identified subsystems and decided on Game Logic subsystem
- ◆ Analyzed discrepancies

Updated Conceptual Architecture



Concrete Architecture

Unexpected Dependencies

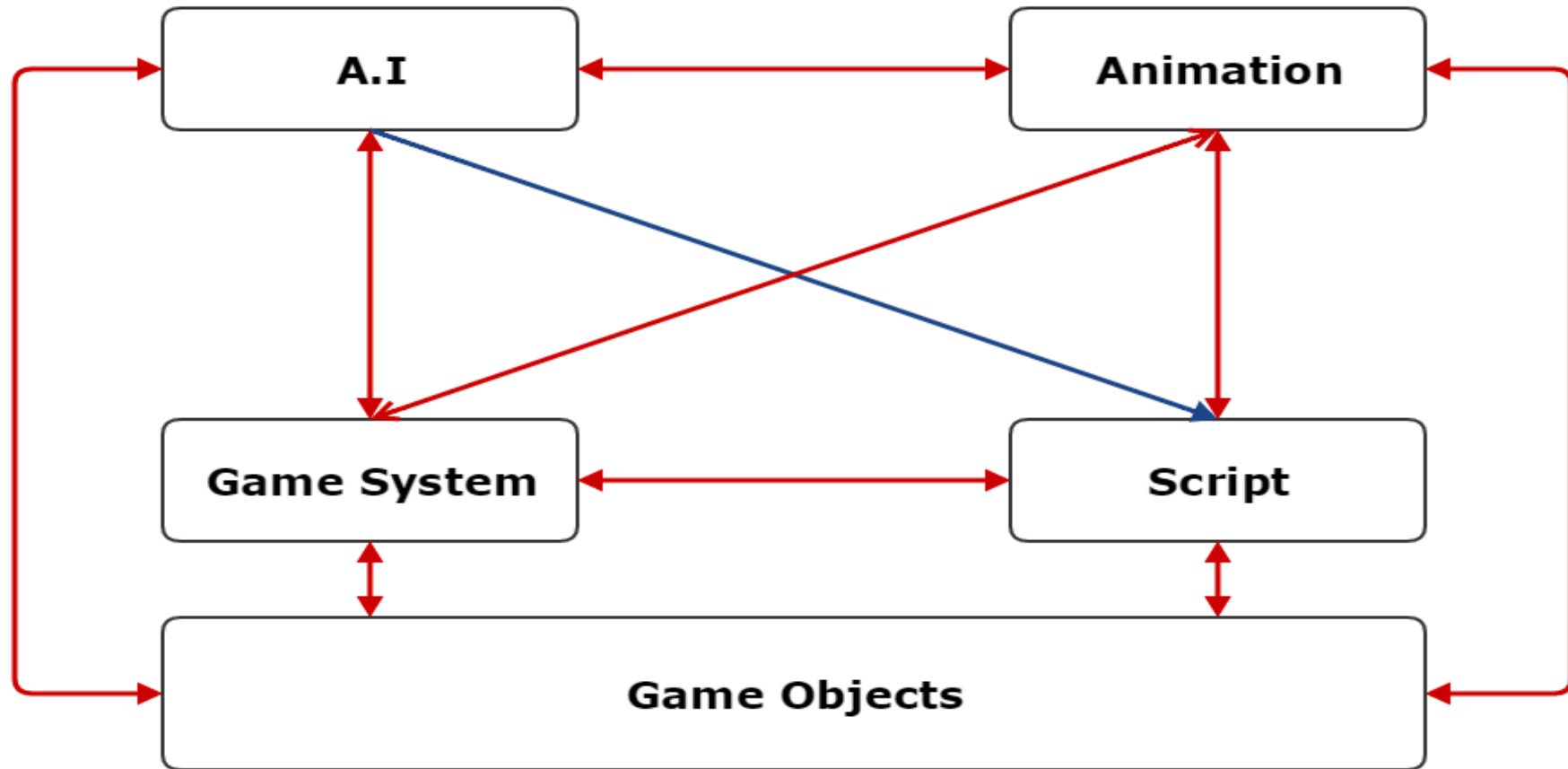
◆ **Logical:**

- ◆ Universal bidirectional dependency with **Core System**
- ◆ Bidirectional dependency between **Networking** and **Core System**
- ◆ Unidirectional dependency between **Physics** and **Renderer**

◆ **Illogical:**

- ◆ Dependencies between the **UI**, **Renderer**, and **Physics** subsystems and **Networking**

Internal Dependencies



Legend:

Dependency

Data Path

Component

Reflexion Analysis

Alternative Architectures

- ◆ Initially had platform independence layer
- ◆ Initially had game assets as it's own component
- ◆ Tried including idLib in Game Logic (precompiled.h includes many files found in Game Logic Component)
- ◆ Tried gamesys in core systems but too tightly coupled with other game logic components

Subsystem

Sequence Diagram

Player Shooting

Weapon.cpp
GL - d3xp

Game_local.cpp
GL - d3xp

Projectile.cpp
GL - d3xp

Player.cpp
GL - d3xp

snd_emitter.cpp
CS - sound

Physics_base.cpp
Physics

render_world.cpp
Renderer

SpawnEntityDef(projectile)
Tell game to spawn

SpawnProjectile()
GetEyePosition()
return pos

CreateRidgidBody(vect1, velocity, entity)

LaunchProjectile()
Checks created succesfully

StartSound()

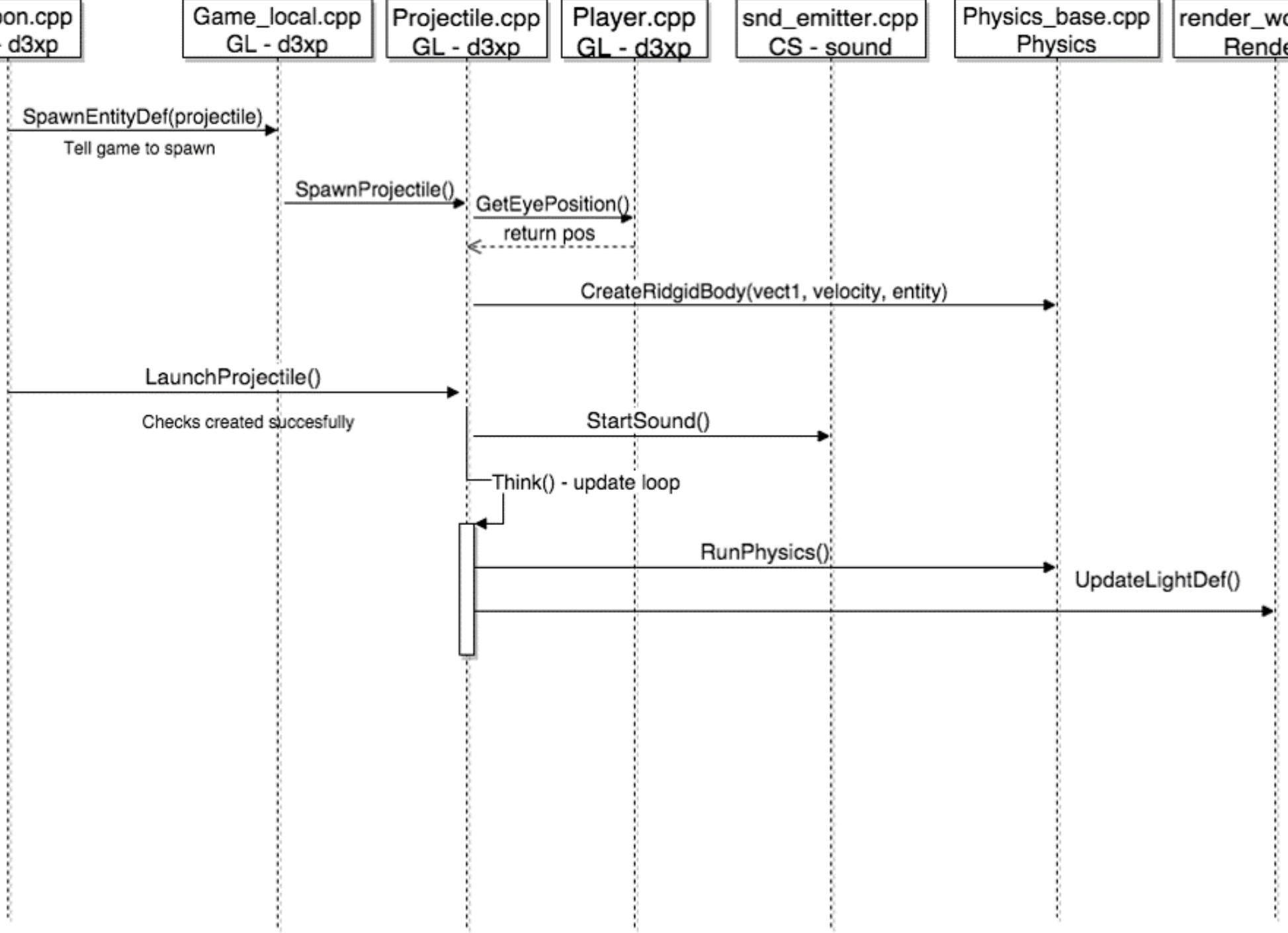
Think() - update loop



RunPhysics()

UpdateLightDef()


Fire Projectile



Concurrency

- ◆ Follows the conceptual triple-threaded multi-threading model with support from Worker threads.
- ◆ Code responsible for maintaining multi-threading exists in the **Core System** under `idLib\sys`.
- ◆ Core System is called by all subsystems – multi-threading is ubiquitous in the program.

Limitations

- ◊ Knowledge of the Understand program
- ◊ Understand Crashing

Lessons Learned

- ◆ Use a program to determine dependencies
- ◆ Unexpected dependencies

Summary

◆ DOOM 3 BFG uses an Object Oriented Layered Architecture