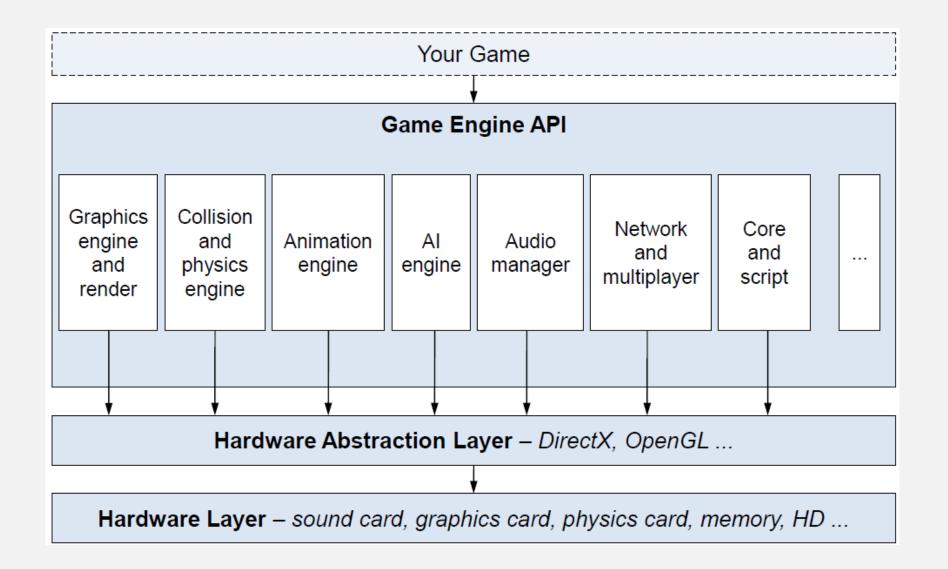
게임 엔진 프로그래밍

# 제2강 게임 엔진 구조

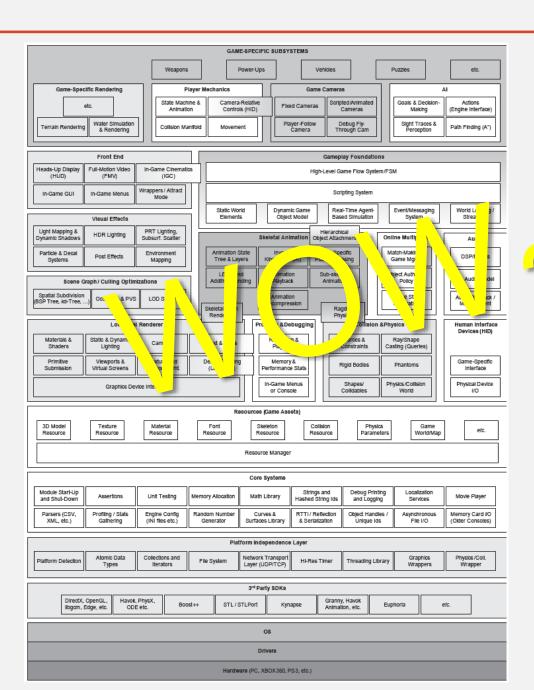


### 학습 안내

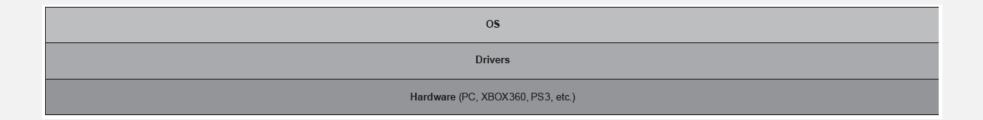
- 게임 엔진 구조
- 게임 엔진 구성 요소



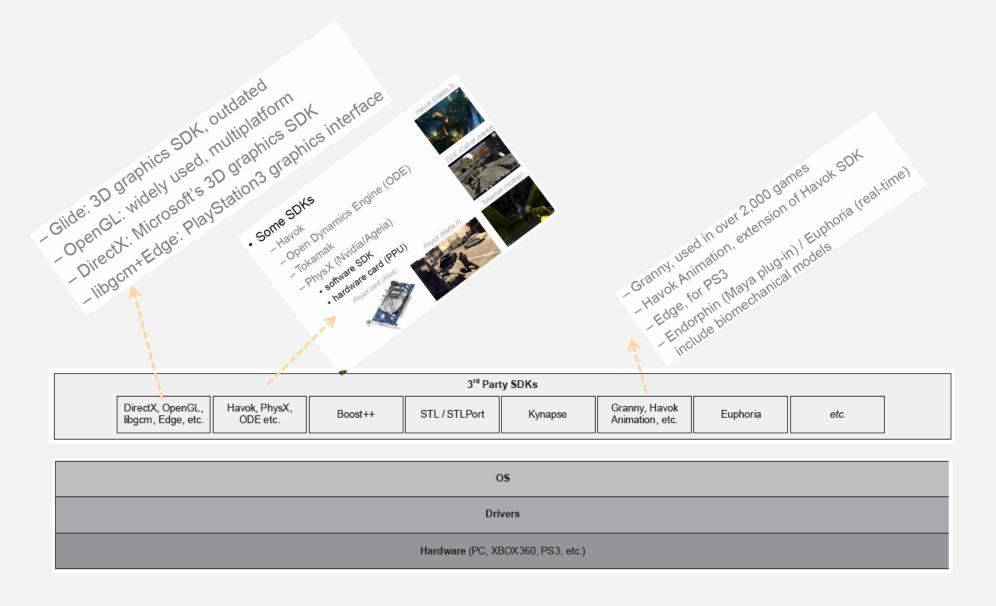
#### 제대로 살펴보면…



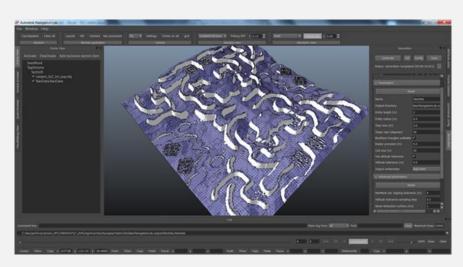
#### Hardware - Drivers - OS

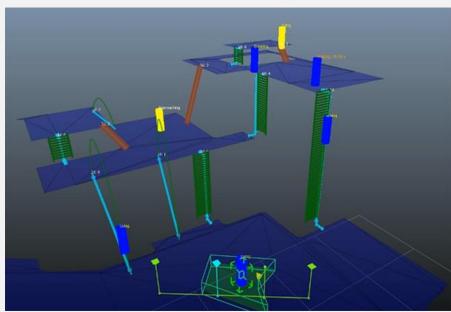


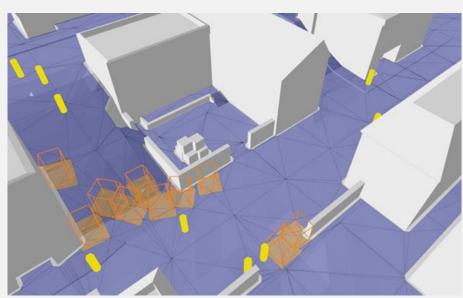
#### 써드파티 SDK



### **Kynapse** → **Gameware Navigation**

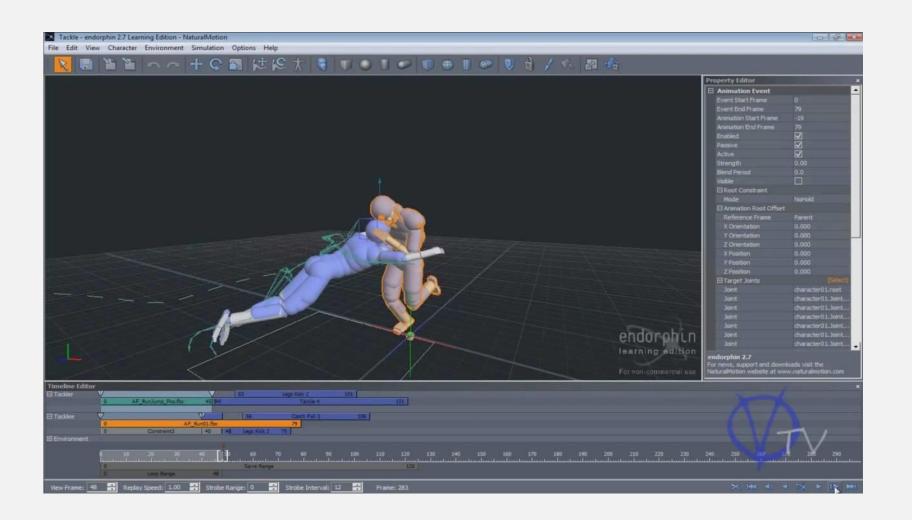






#### **Endorphin**

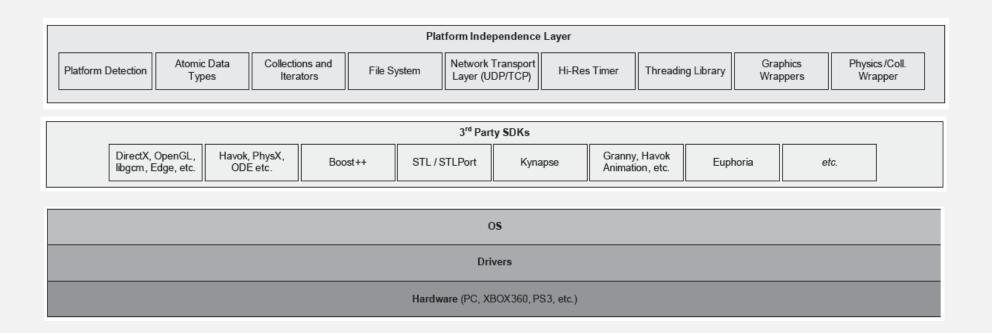
■ 무게중심, 무게분포, 중력 작용 등에 기반한 생체역학적 시뮬레이션 결과를 애니메이션으로 Export



### DirectX VS. OpenGL

	Direct3D	OpenGL				
장점	<ul> <li>마이크로소프트의 강력한 지원</li> <li>게임에 최적화된 빠른 속도</li> <li>최신의 3D 기술을 반영한 업그레이드</li> <li>많은 하드웨어 드라이버 지원</li> </ul>	■ 긴 역사를 통한 검증된 구조 ■ 플랫폼 독립적인 프로그래밍이 가능함. ■ 쉬운 문법과 깔끔한 구조 ■ 게임 외 다양한 3D 그래픽 애플리케이션 작성에 용이 ■ PC뿐만 아니라 스마트폰, 리눅스 등 다양한 플랫폼을 지원함.				
단 점	■ 너무 잦은 업그레이드로 기술 습득 및 축적의 어려움 ■ 윈도우에서만 사용 가능	■ 다이렉트3D에 비해 최적화에 어려움 ■ 최신의 기술로 업그레이드가 늦음 ■ 다양한 하드웨어 지원이 약함				

### 플랫폼 독립 계층



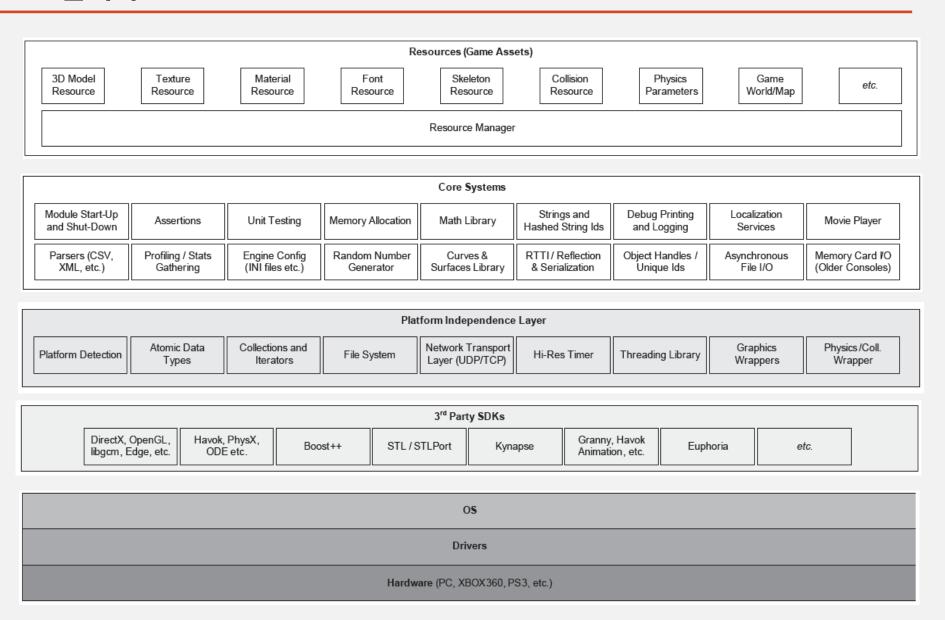
Core Systems								
Module Start-Up and Shut-Down	Assertions	Unit Testing	Memory Allocation Math Library		Strings and Hashed String Ids	Debug Printing and Logging	Localization Movie Play	
Parsers (CSV, XML, etc.)			RTTI/ Reflection & Serialization	Object Handles / Unique Ids	Asynchronous File I/O	Memory Card IO (Older Consoles)		

Platform Independence Layer								
Platform Detection	Oetection Atomic Data Collections and Iterators		File System Network Transport Layer (UDP/TCP) Hi-Res Timer			Threading Library	Graphics Wrappers	Physics /Coll. Wrapper

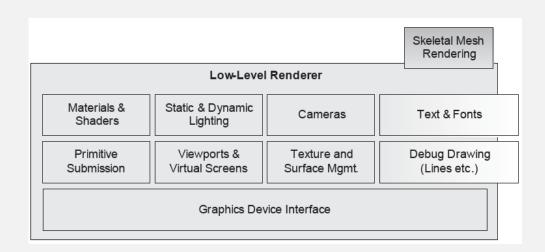
3 <sup>rd</sup> Party <b>\$</b> DKs								
DirectX, OpenGL, libgcm, Edge, etc.	Havok, PhysX, ODE etc.	Boost++	STL/STLPort	Kynapse	Granny, Havok Animation, etc.	Euphoria	etc.	

os		
Drivers		
Hardware (PC, XBOX360, PS3, etc.)		

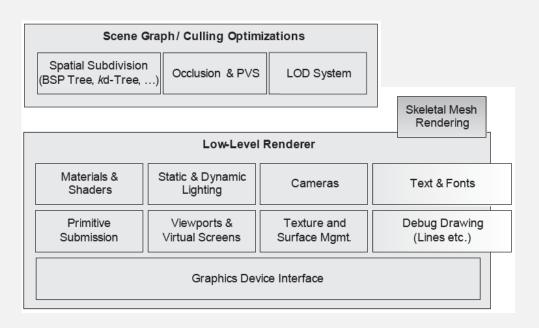
#### 리소스 관리자



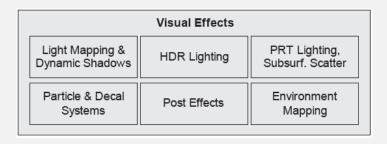
### 렌더링 – 로우레벨 렌더러(Low-Level Renderer)

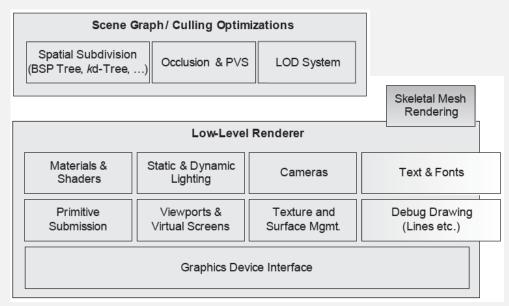


### 렌더링 – 씬그래프(Scene Graph) 및 컬링(Culling) 최적화

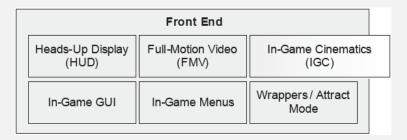


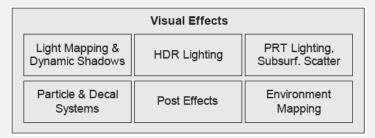
### 렌더링 – 시각효과(Visual Effects)

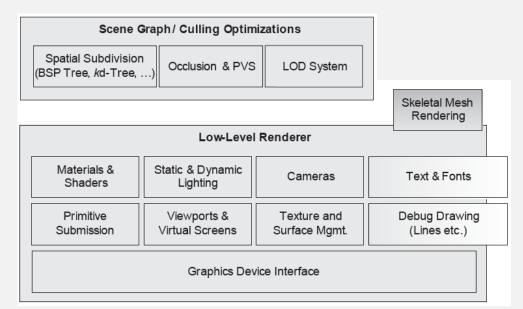




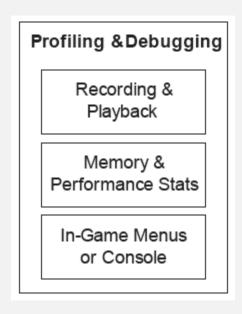
#### 렌더링 - 전단부(Front End)

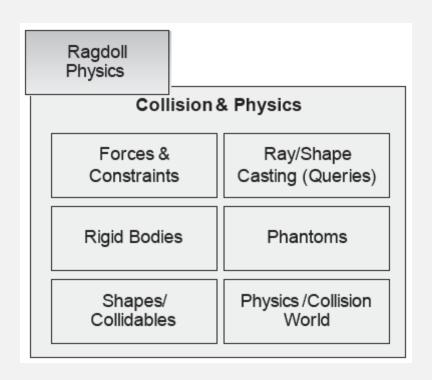






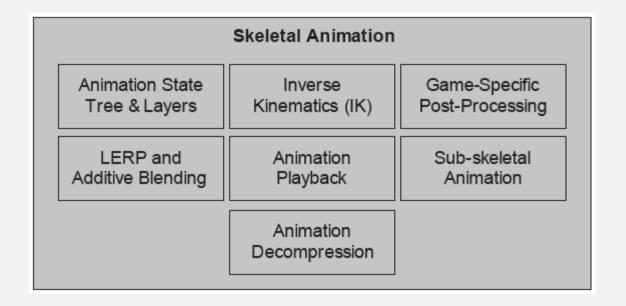
### 성능분석(프로파일링:profiling)과 디버깅



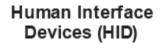


#### 애니메이션

- 게임에서 캐릭터와 기하학적 객체들의 자연스러운 움직임을 묘사해내지 못하면 재미가 없다
  - □따라서 캐릭터와 기하학적 객체들의 자연스러운 움직임을 보여줄 필요성이 있게 됨
  - □물리 엔진 및 렌더링 엔진과 밀접하게 연동이 됨.



## 휴먼 인터페이스 장치(HID)

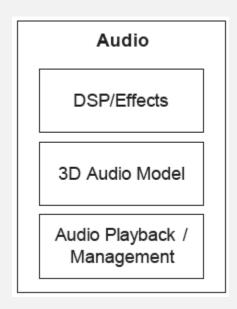


Game-Specific Interface

Physical Device I/O

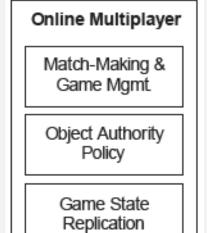
#### 오디오

- 다른 부분들에 비해 상대적으로 간과되는 경향
- But, 훌륭한 게임은 반드시 좋은 오디오 엔진이 요구됨!



### 온라인 멀티플레이 및 네트워킹

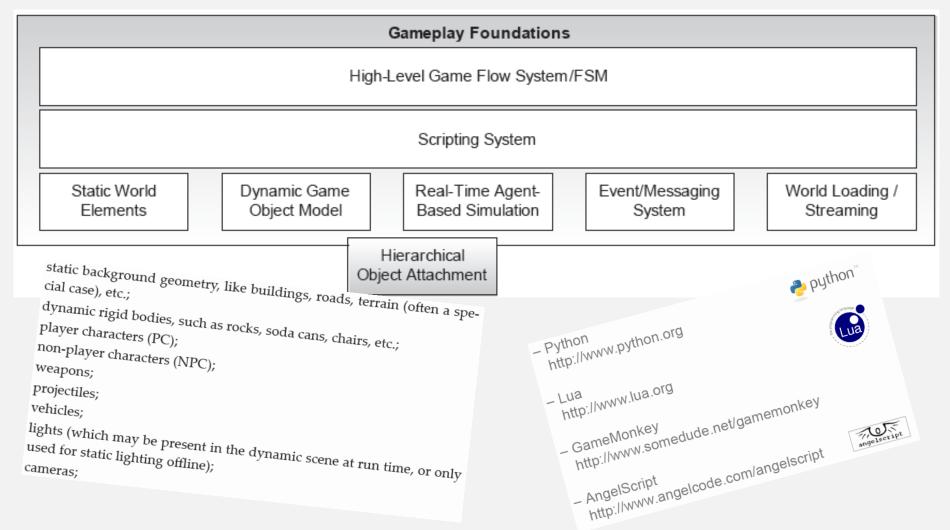
- 단일 스크린 멀티플레이어
- 분할 화면 멀티 플레이어
- 네트워크 멀티 플레이어
- MMOG



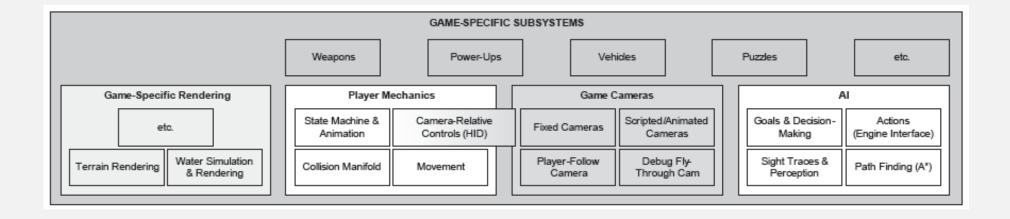
#### 게임 플레이 기반 시스템

#### ■ 게임 플레이

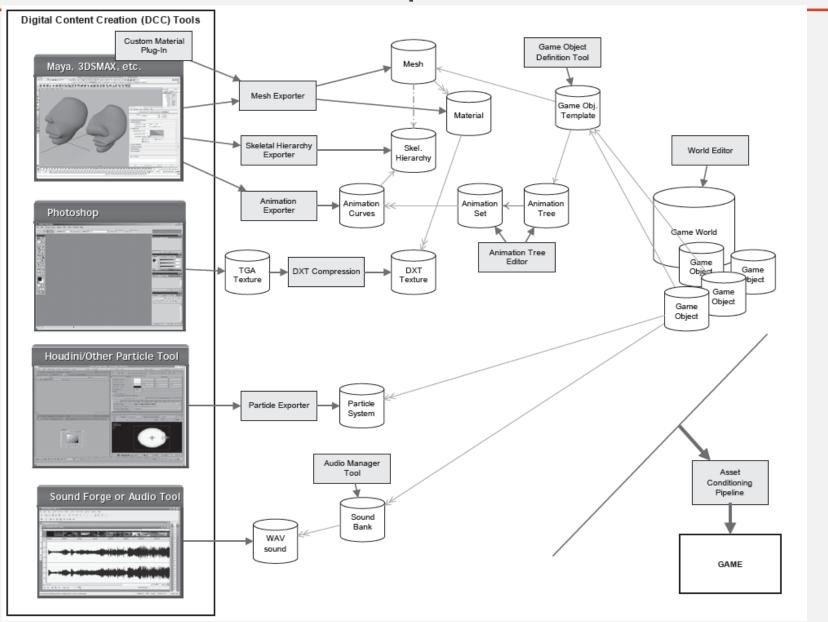
□플레이어의 행동, 게임 세계의 규칙, 플레이어 캐릭터의 능력, 플레이어의 목표 및 목적 …



### 게임 특화 서브시스템

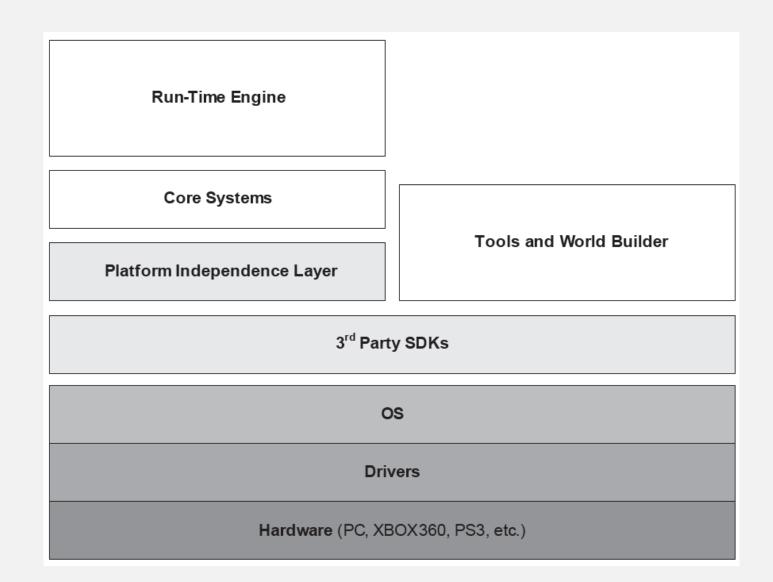


### **Digital Contents Creation / Asset Pipeline**



#### **Asset Condition Pipeline**

- DCC로 만든 데이터를 게임에 바로 사용 X
  - □필요 이상으로 복잡함
  - □로딩에 시간이 걸리고, 저작권의 이슈도 있음.
- Export
  - □ DCC로 만들어진 데이터를 게임에서 활용되는 포맷으로 변환 저장하는 과정
  - □ DCC 안에서 플러그인 형식으로 존재하거나, 또는 포맷 변환툴이 사용됨.



### 도구 역시 프레임워크를 공유하는 구조

Run-Time Engine	Tools and World Builder				
Core Sy	ystems				
Platform Independence Layer					
3 <sup>rd</sup> Party SDKs					
os					
Drivers					
Hardware (PC, XBOX360, PS3, etc.)					

### 런타임 엔진과 에디터의 통합 구조 - UnreadEd, Unity3D

