게임 엔진

1. 게임 엔진의 존재 목적 : 생산성. 시간 없고 기술 안되면 돈으로라도 메꾸라고 있는 것

1> 바퀴부터 다시 만들 필요가 있을까?

2> 기본은 렌더러, 익스포터, 쉐이더/애니메이션 툴

3> 프레임워크, 메타데이터, 파일 시스템, 개발 도구, 버그 리포팅 시스템, 프로파일러, 월드 에디터

4> UI, 물리, 사운드, AI 등의 서드 파티 라이브러리의 엔진화 + 테스트 프레임워크.

2. 특화엔진

1> 렌더러 : 렌더웨어, 겜브리오

2> 물리 : 하복, 피직스X

3> UI : 스케일폼

4> 애니메이션 : 캐릭터 스튜디오

5> 장르 특화 엔진

3. 통합엔진

1> 언리얼, 유니티, 비젼 etc…

2> Asset market

4. 엔진에 대한 논의는 더 이상 필요 없고, 그냥 **공짜 되버린 언리얼 쓰세요!**

5. 해묵은 괴담 1 : 엔진이 업글되면 그래픽 때깔이 좋아질까?

1> 엔진은 도구일 뿐 최종 결과는 오로지 그래픽 디자이너의 능력에 좌우

2> 하지만 같은 기능 구현 비용은 확실히 asset이 충실한 엔진이 유리

3> 신규 기능 구현비용의 차이가 존재

4> 가끔 엔진 default material의 기본 값을 엔진의 색감이라 착각하는 사람들이 있는데 그냥 웃어주길

5> 오히려 고급 엔진일수록 파이프라인 중간에 개입되는 변수를 제거하기가 힘듬. 장점이자 동시에 단점

6. 엔진은 꼭 수정해 써야 한다?

1> 오륙년 전까지만 해도 당연히 여겨졌던 일

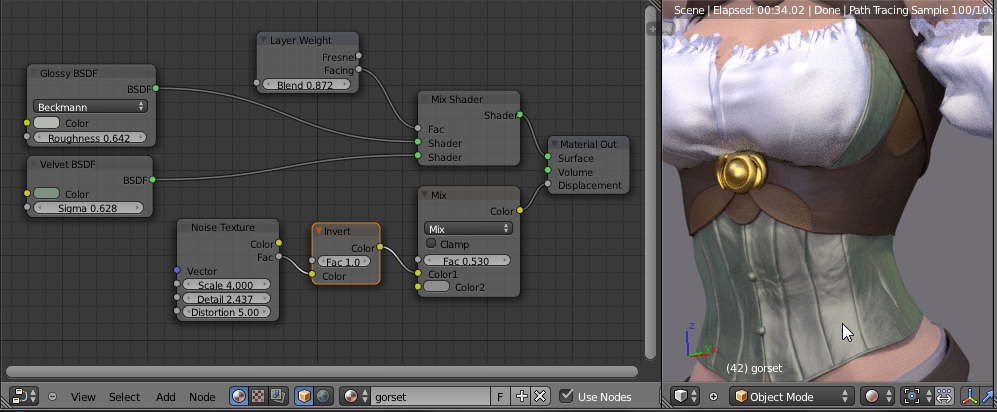
2> 하지만 엔진을 사 쓰는 의미가 줄어드는데다 엔진 업데이트시마다 갖은 문제 발생

3> 엔진 자체도 많은 발전을 해서 현재는 수정하기보단 그대로 쓰고 대신 asset을 늘리는 방식으로 진화

7. 엔진 구성 요소

1> 렌더러 & 익스포터 -> 렌더러는 사실상 언리얼 3이후 고정. 익스포터는 FBX로 대동단결 중

2> 쉐이더 : fx style script, visual / node base shader editor



3> 스크립트, asset management

4> 툴! 범용성을 위한 plugin 제공

5> 엔진 특성 : 인스턴싱(군중 처리), 실내 특화, 광역 필드 특화, 포탈, 범용성

6> 기타 등등은 보너스

8. 오픈 엔진

1> 다 죽고 Ogre 하나만 명맥 유지 중

2> 언리얼도 소스 공개 끝났음. Game over;;

9. 엔진 개발사는 거의 북미와 유럽이다보니 그에 따른 관점 차이

1> 3D는 art가 아닌 engineering

2> 엔진 관련 프로그래머는 당연히 3D 툴(보통 3DS MAX)을 만질 줄 알고

3> 3D 디자이너는 당연히 관련 기술 요소를 모두 알고 있으며

4> 기획자들은 당연히 툴이나 스크립트를 능숙하게 다룰 수 있다고 전제

5> Unicode(다국어 처리)에 대한 무지

6> code보다 asset 중심!

10. 엔진 철학

1> 방대한 양에 co-op가 기본이다보니 일관성 유지가 중요한 미덕

2> excute : 실행에 대한 뼈대. 코드 혹은 스크립트

3> rule : 실행 규칙 : 스크립트 혹은 data

4> data : 말 그대로 데이터

5> rule과 data를 합쳐 asset이라 부름(덤으로 unity의 최고 약점)

6> 철학 유지는 중수와 고수를 가르는 잣대