영상적 효과 II:
The Revenge

케빈 비욕 **(Kevin Bjorke)**

개요

- 영화와 게임:차이점과 유사점
  - 비주얼 퀄리티와 “Look Development”(외양의 개발)
  - 제작 규모
- 영화에서 얻은 아이디어의 현실화
  - 새로운 도구, 쉐이더, 아이디어
  - 실시간 데모 예제들
  - 게임 엔진에 적용
  - 아트 파이프라인에 적용
- 소스 코드!
  - http://developer.nvidia.com에서 여러 소스코드 정보를 제공.

텍스쳐 좌표의 “MRT” 비쥬얼 화

“Revenge(복수)”???

- *지금까지의 진행상황은...**
  - 프로그래밍이 가능한 쉐이딩을 통한 영상 효과는 지금까지 가장 강력한 게임용 미술도구
  - 그러나 고전을 면치 못함.
    - 구현과 실험이 어려움
    - 게임 엔진 내부에 적용하기 어려움
    - 디버깅은 더욱 어려움
- 수확을 할 시간.

애니메트리스의 “Thad” – Character © Silver Pictures

* Part I은 다음 사이트에서 제공되고 있음 http://developer.nvidia.com/

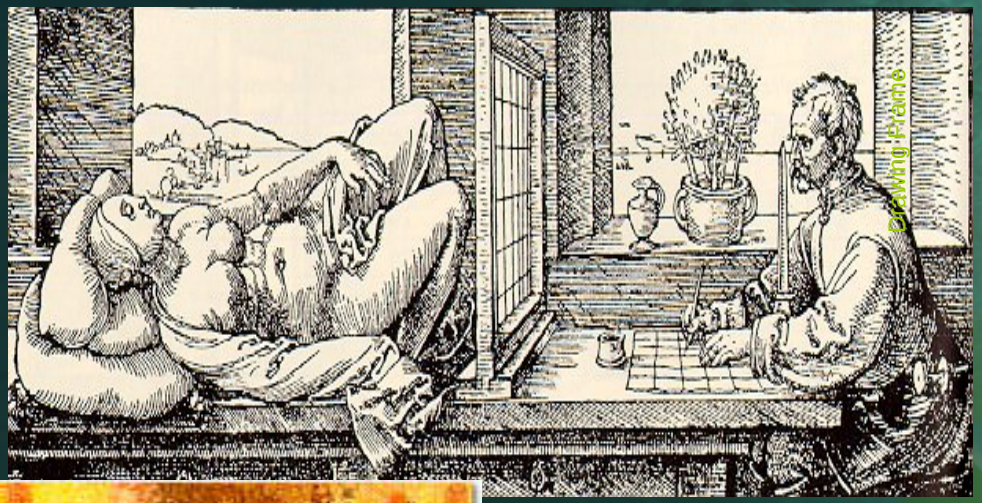
시각적 예술과 게이밍

- 두 가지는 항상 연결되어 있다.
- 세계에서 가장 오래된 예술작품은 아마 “고 득점 기록”!

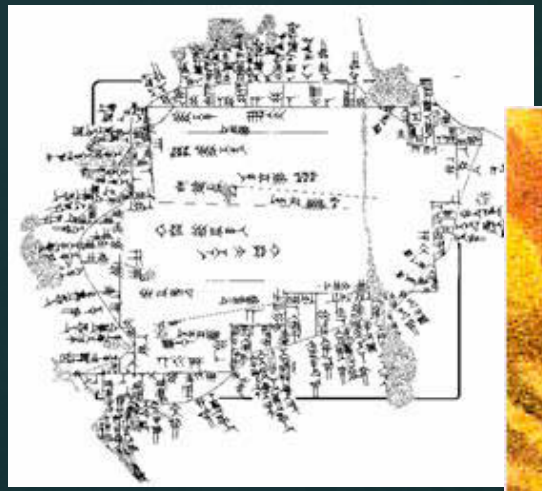
쇼베동굴 – 기원전 25,000 B.C으로 추정.

기하학과 빛

- 컴퓨터 그래픽은 가장 최근의 개발형태
- 영화와 사진
- 광학과 기하학
- “측정된 시각(視覺)”



Albrecht Durer



메소포타미아의 측량도, 기원전 2500년 경

쇼베동굴 – 기원전 25,000 추정

영화와 현실

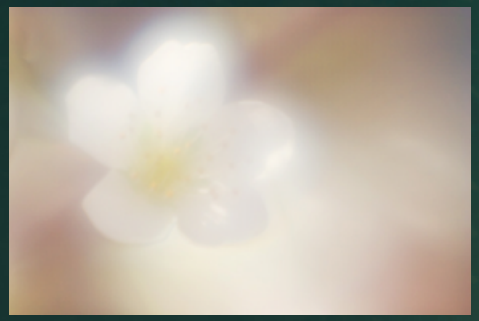
- 영화는 다큐멘터리가 아니다.
- 영화는 역동적이며 스타일이 가미된 일러스터레이션
  - 주관적, 비객관적
  - “실물보다 더 생생함”
- “다큐멘터리 스타일” 은 – *이러한 스타일*
  - “Reality TV” 각색된 비밀
  - BBC의 “The Office” 명백히 대본에 의한 것

Sylvia ©2003 Focus Features

CGI 과 “포토리얼리즘”

- “포토리얼리즘은 ” 또 다른 스타일
- 사진도 추상적일 수 있다!
- 자연은 인간의 방정식으로 재현할 수 있는 것보다 훨씬 더 많은 요소로 구성됨
- 영화는 이미지 창조의 개념을 초창기 매체로부터 많이 빌려옴

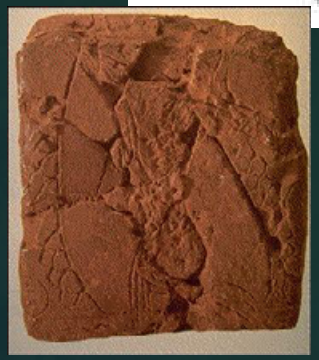
Vogue cover, 1950 - Penn



정밀과 추상의 혼합

- 지도 및 원근도 (문학을 포함하여) 는 회계에서 개발됨
- 정확한 묘사도 중요했으나 소유권이나 과세 등 추상적인 문제보다는 중요도가 낮았음
- 그 당시에도 “예술적 생략” 은 중요하였음

Mesopotamian Survey Map, ca. 2500 B.C.



CGI, 필름, 그리고 회화

- 필름은 회화와 같은 매체에서 빛과 구성법을 차용한다.
- 빛은 시선을 끈다.
- 빛은 감정적 상태를 좌우한다.

리들리 스콧감독의 *블레이드 러너*

라파엘의 *변모도*

외양의 개발 (look Development)

- “외양의 개발”이란 어떤 프로젝트(혹은 프로젝트의 일부에서) 무엇이 중요한지 (혹은 무엇이 중요하지 않은지)를 결정하고 스타일의 요소를 배치하는 것이다.
- 개발과정에서 “look”(외양)의 결정이 빠르면 빠를 수록, 좋다 (그리고 사용에 있어서도 비용절감이 된다).

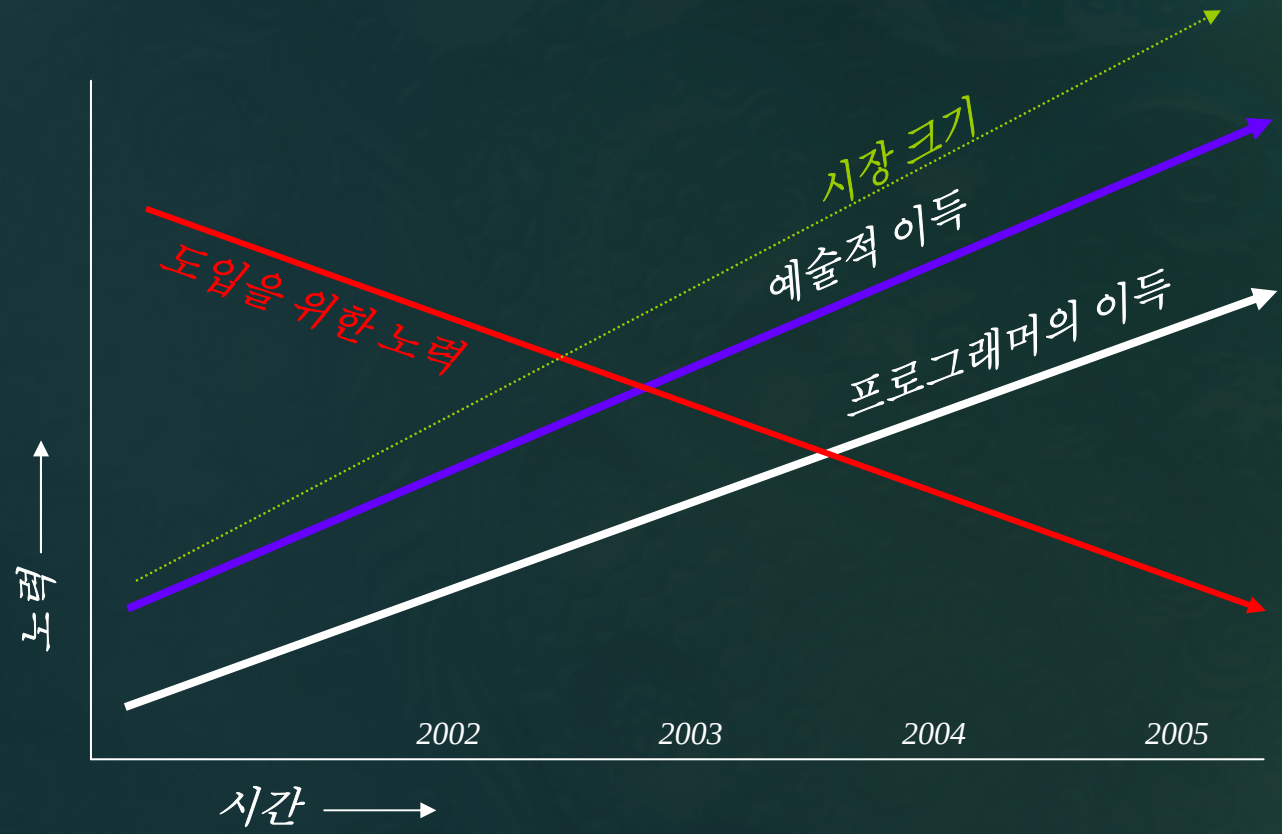
게임에서 외양(Look)의 개발

- 게임에서 외양은 종종 게임엔진 설계의 부산물이다.
  - 아티스트는 “최소한의 공통분모” 이외에 어떤 것도 추측하기 어려움
  - 디자인은 보수적이고 안전한 방향으로 진행하려는 경향이 있다.
  - 기술적 제약에 신경을 씀
- 영화의 경우는 초기 개발은 예산과 같은 “ 세부사항”에 대한 철저한 주의 없이 수행되는 것이 일반적
  - 아티스트는 완전히 자유로움
  - 스토리에 신경을 씀

Test from *Animatrix* – © Silver Pictures

프로그래밍 가능한 쉐이딩: 언제?

- 작업 흐름에 도입 시 대가와 장점이 있음
- 각 제작사에 자체적인 "손익분기점"이 발생할 것임



쉐이더 개발: 프로그래머

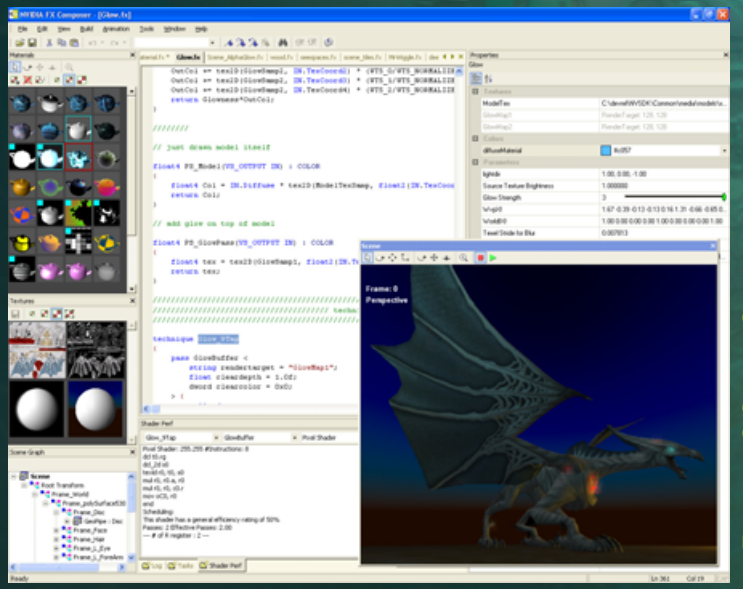
- 쉐이딩 도구는 프로그래머와 디자이너 모두에게 중요
- 현대적 게임 엔진을 완벽히 지원하려면 도구(Tool)는 다음 기능을 갖추어야 한다.
  - Render-To-Texture (RTT) (텍스쳐에 렌더링하기)
  - Multiple Render Targets (MRT) (다중 렌더 타겟)
  - 스텐실, 알파 블렌드 등의 렌더링 스테이트
  - 변형 텍스쳐 맵(예:노멀라이제이션 큐브, 노이즈)
  - 복잡한 배열이 어떠한 특정 게임 엔진의 렌더링 루프와도 들어맞을 수 있다는 것을 보장하는 관리기능
  - 스크립트 가능
- 제작의 모든 단계에 게임 엔진으로의 도입 및 제거 시 결과를 어떻게 확인할 것인가?

쉐이더 개발 : 아티스트

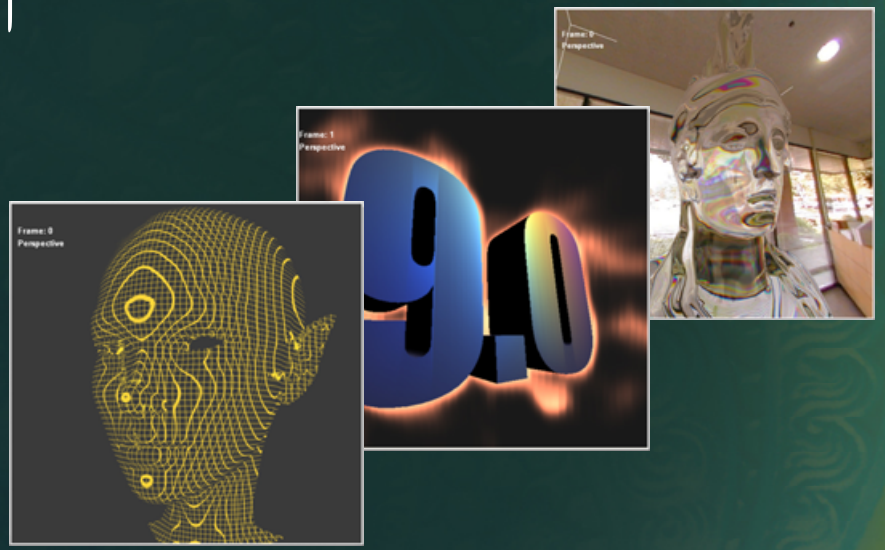
- 아티스트는 나중에 어떻게 보일지 추측만 하는 것보다는 자신의 디자인을 직접 보기를 원함
- 올바른 모델 외에도 정확한 라이팅(lighting)환경을 사용해야만 개발된 쉐이더와 모델이 실제로 게임에 사용될 수 있다.
- 렌더링 구현은 대개 DCC 애플리케이션(Maya, Max, XSL 등)마다 다름. 그러나 실제의 게임과 일치하는 것은 없음
- 콘솔 게임 디자이너의 요청도 수긍하려 함- 즉, 같은 게임의 다른 버전의 모델 (DX9, DX8, Xbox, PS…)을 보여 줄 방법을 제공

HLSL 및 FX Composer

- 작업을 위해 만들어진 도구
- 쉐이더들을 결합
- 쉐이더 사용자 정의
- 코드의 재작성 및 추가의 SDK 혹은 런타임 레이어 없이 이동 가능
- 성능 튜닝 도구
- C# 및 VB 스크립팅
- http://www.fxcomposer.com/



Everquest® Content Courtesy
Sony Online Entertainment Inc.

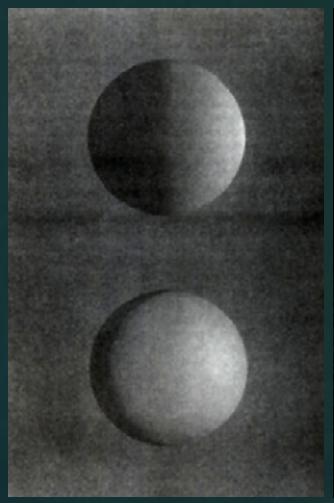


쉐이딩 스케치북

- FX Composer 는 아티스트와 프로그래머들이 복잡한 개념의 실험을 위해 C++ 로 게임 엔진 전체를 작성하지 않고도 탐구해 볼 수 있는 환경을 제공

쉐이더 라이브러리 구축

- FX Composer 에는 많은 샘플 쉐이더가 들어 있음
  - 아무 HLSL FX 쉐이더나 사용 가능하며, 다른 쉐이더 도구에서 작성된 것도 사용 가능
- 실험을 해보십시오! 그리고 결과를 저장/보관하십시오
  – 언젠가는 사용을 하게 될 것입니다
- 저장, 교환, 수집하십시오.



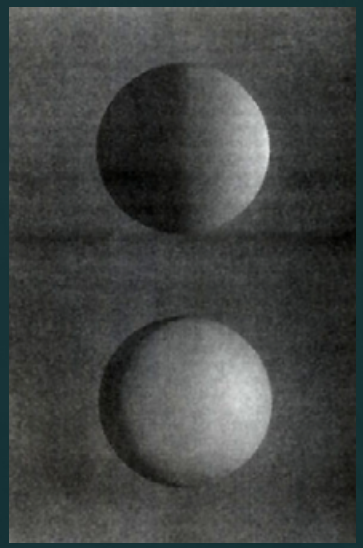
Ruskin's Shading Exercises, 1877

Bjorke's Dumb Mistake, 2003

스케치북 예제:
연필 스케치를 쉐이더로 변환

- 쉐이딩된 구는 쉐이더로 변환하기 쉬움
- 색 참조에 유용함
- JPEG 노이즈 등 사소한 부분에 주의, 얼룩으로 나타남

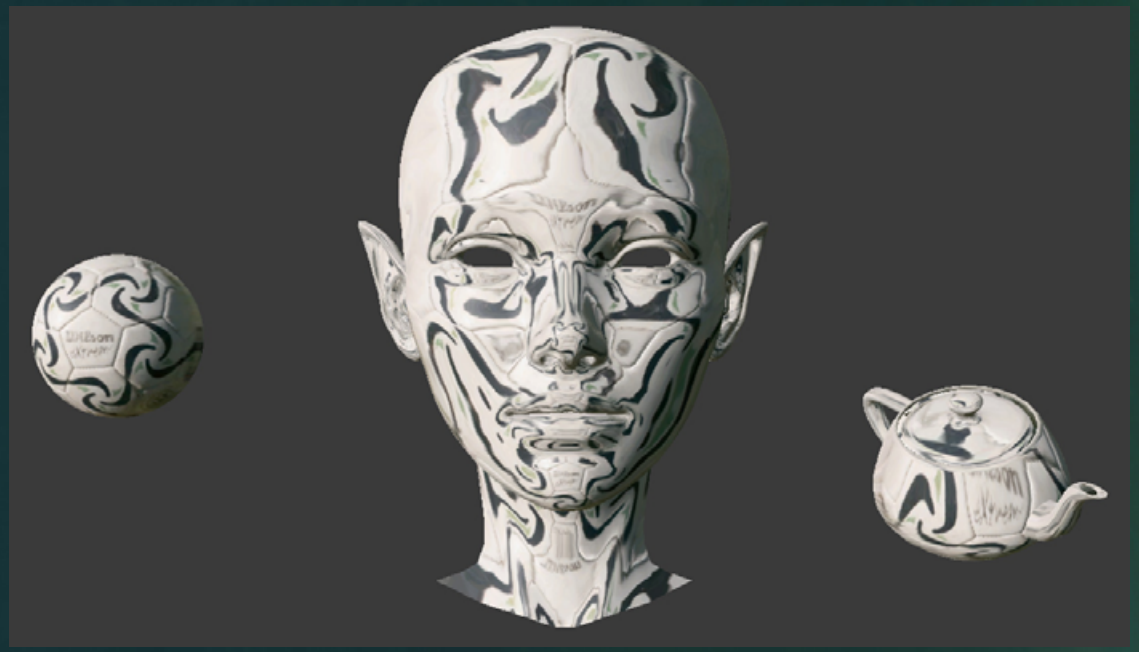
1877



2004

멋진 예제

- 사진은 왜곡됨
- 별 쓸모는 없겠지만 *저렴함* – 단 한 사이클!
- 이것을 어디엔가 유용하게 쓸 수는 없을까?

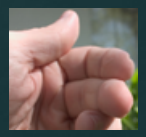


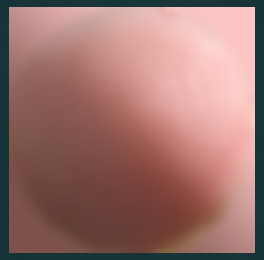
형태가 반드시 구(Sphere)여야 하나?

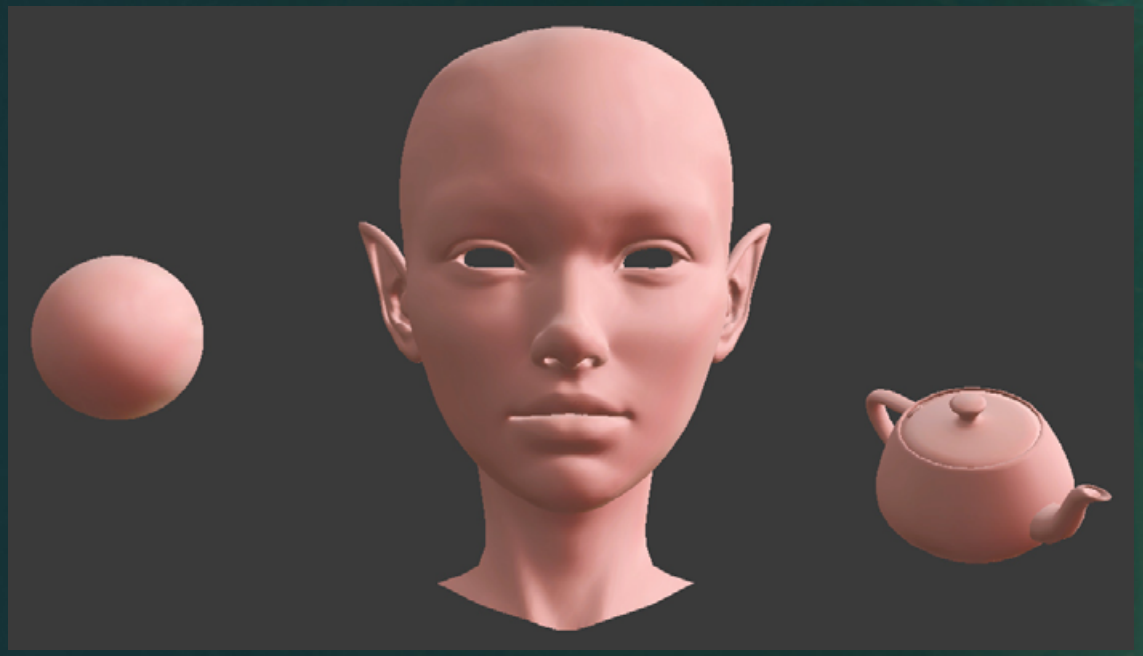
- 포토샵으로 작업할 용의가 있다면 그렇지 않음
  - 본인은 “Liquify” 및 Smudge/Stamp 도구를 선호

컬러 조정

- 색 분리를 위하여 텍스쳐에 Gaussian Blur를 적용
- 다른 쉐이딩 모델들과 혼합하기에 좋음

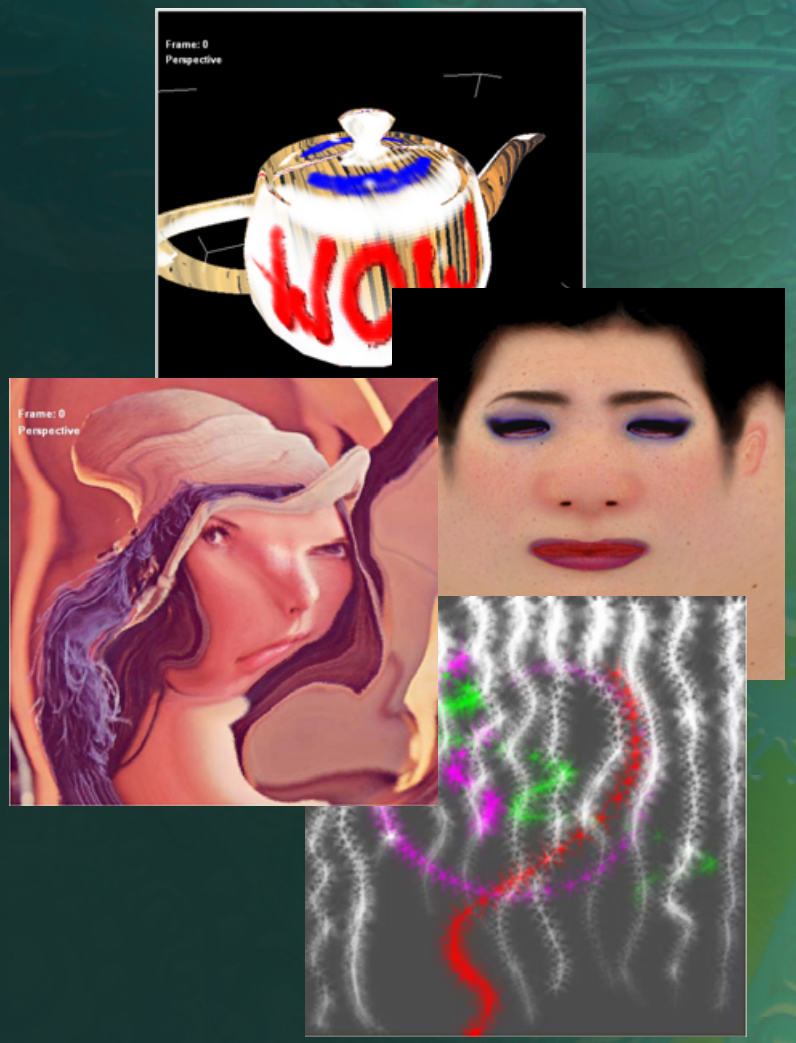






FX Composer로 스케치

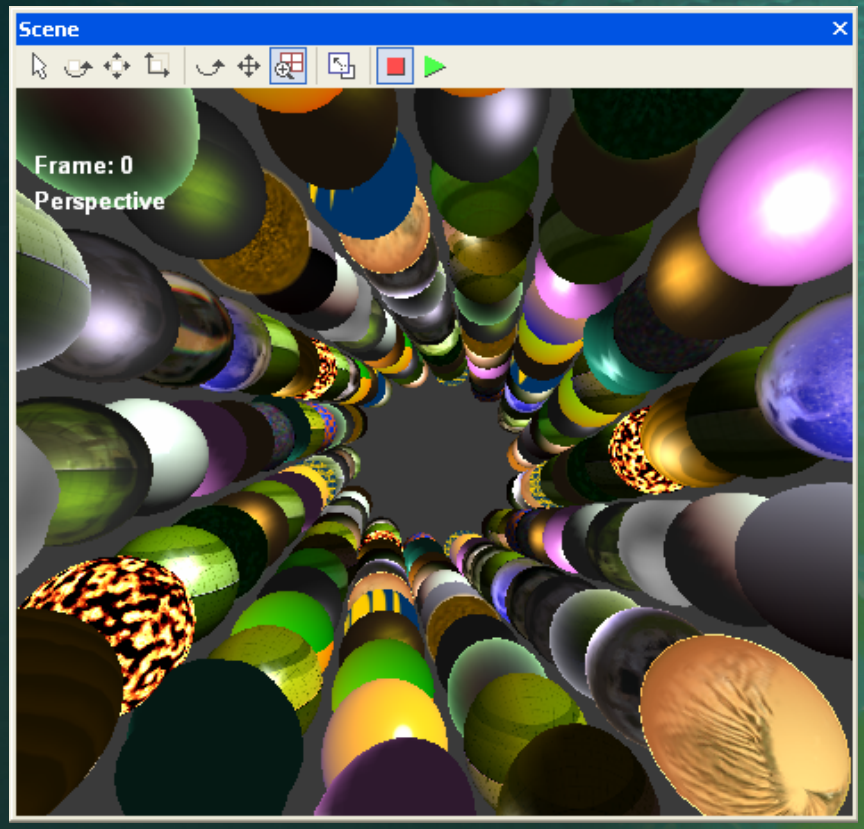
- 스케칭에 관한 토의:
- FX Composer 에서는 마우스 이벤트를 가로채는 것이 가능
- FX 쉐이더만 가지고도 미니 애플리케이션을 개발 가능



동영상: 스케일 관리

- 영화에는 대량의 스케일이 있음
- 수많은 도형
- *많은 쉐이더*
  - *토이스토리: 1300개의 쉐이더*
  - *벅스: 두 배*
  - *몬스터 주식회사: "수천 개"*
- 많은 혼합 레이어 (경우에 따라서 수백 개)
- 스크립팅 도구 (Perl, C#, VB, Python, Mel 등 다종다양!)
- *긴 스케줄*
  - 순간 렌더링 기능으로 스케줄 단축…

FX Composer: 스케줄 관리

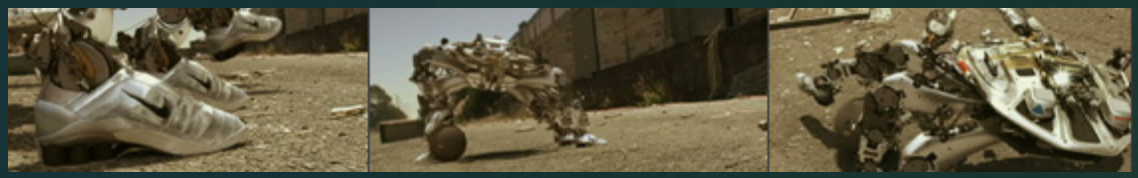


"See all the shaders in a directory"
-- Scene Generated by C# Script

- 게임의 스케일도 증가 추세
- 수 많은 쉐이더와 모델의 관리는 따분한 작업
- FXComposer 는 .NET 어셈블리를 사용함으로 .NET이 FX Composer를 제어하여 장면 제작, 이미지 익스포팅, 쉐이더 지정, 데이터 익스포팅등을 내보내기 등을 신속하게 할 수 있음
- C# 또는 Visual Basic 사용

긴 스케줄

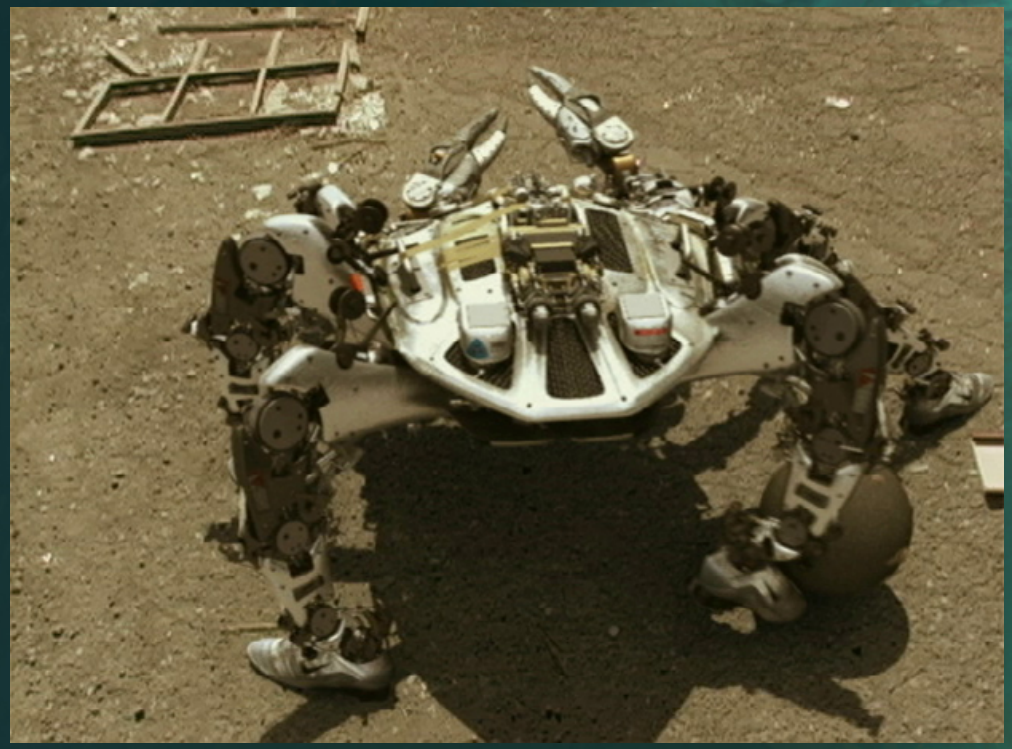
- 자금과 시간이 풍부한 영화 쪽에서는 첨단 기술을 개발할 잠재력이 있음
- 그러나! 그러한 기술들이 상당히 초기에 결정되어야만 영화 제작 마지막 날의 촬영분이 첫 날의 촬영분과 비슷하게 보이도록 해야 함
- 이것이 혁신에 대한 제약을 야기함
- 가장 빠른 혁신: *TV 광고*



Nike.Com campaign, Weiden + Kennedy, Dir Neill Blomkamp http://www.theembassyvfx.com/

NikeLab.COM

- Embassy Visual Effect의 제작
- 4주 만에 완성!
- Lightwave, Shake,NVIDIA Quadro GPU 사용



Nike campaign, Weiden + Kennedy, Dir Neill Blomkamp
http://www.theembassyvfx.com/

그림자

- 조명(illumination)보다 그림자가 중요한 경우가 많음
- 그림자를 일단 그려 넣으면 되돌리기 어려움

1998

2004

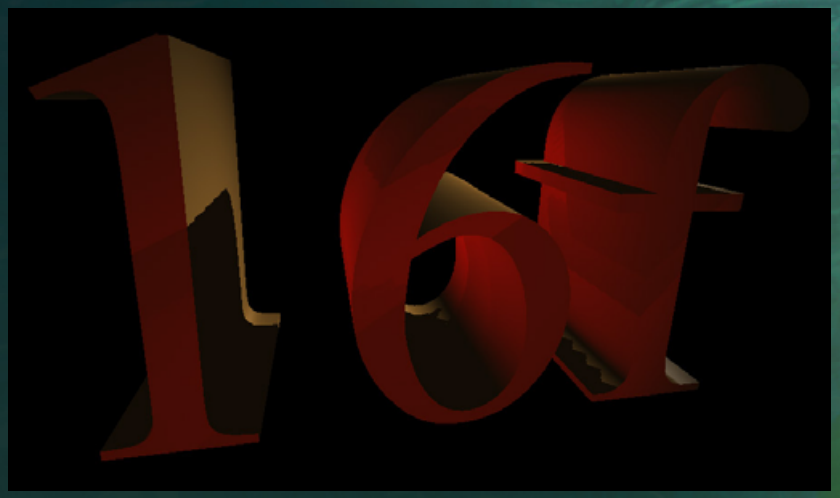
기술 레슨: 그림자

- 단순 그림자 : 스텐실 볼륨 또는 텍스쳐 렌더링
- 조명의 위치는?
- 조명의 공유
- “Advanced Shading” 에 대한 토의에서 이 문제도 일부 언급할 것임

The Art Lesson

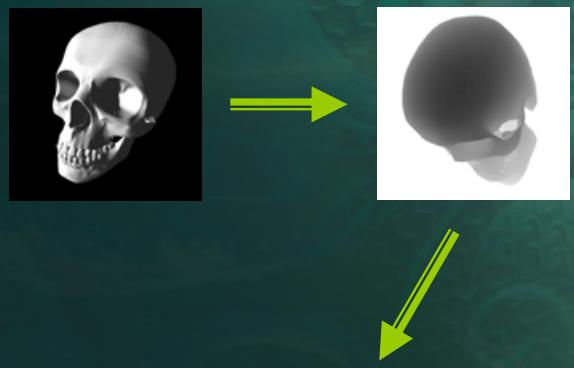
화려한 그림자 -- 반투명

- 그림자의 Z 값이 물체 뒤쪽에서도 사용 가능
- “Advanced Shading” 시간에 이 기술 및 기타내용 상세히 논의



DXSAS – 스크립팅 가능한 FX/HLSL

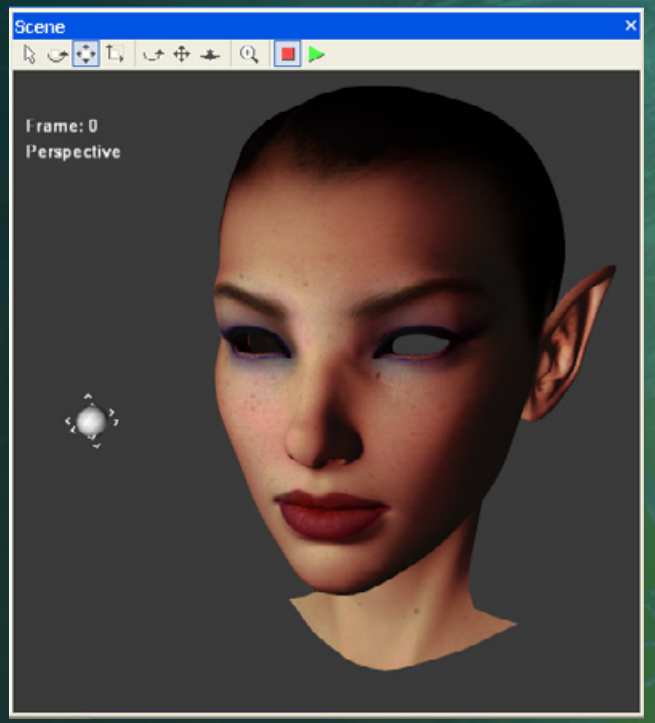
- DXSAS = “DirectX Standard Annotations and Semantics” 그리고 Microsoft의 XNA의 표준* 부분
- 모든 패스 및 기법에 “Script” 세만틱(semantic) 포함
- 스크립트를 통한 렌더링 대상 정의, 반복(loop) 가능, 상호 호출 가능
- HLSL “Virtual Machine” (VM) 행열수학과 같은 숫자처리도 함



The Art Lesson

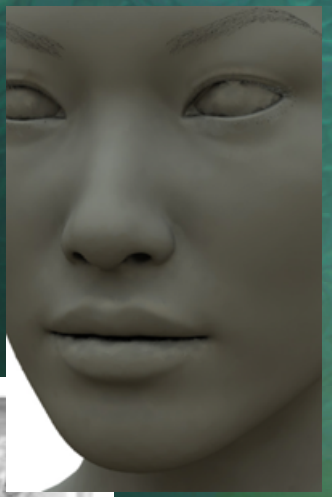
스킨과 그림자

- 서브서피스 스캐터링의 산란 효과(저렴한 비용):
  - 그림자 산란(diffuse-shading) 계산에서 “(N·L)” 을 “((N·L)+w)/(1+w)” 로 재매핑하면 개체의 윤곽선 주위를 빛으로 ‘감쌀’ 수 있음
  - (수학적인 부분은 예제가 있으므로 걱정 끝!)
  - 이것이 산란 조명의 전부이므로 경우에 따라 버텍스 쉐이더에서 작업해도 무방



피부와 직접 반사

- 젊을수록 피부가 좋음
- 생생한 피부 세포는 고양이 눈의 반사체처럼 반사율이 높음
- 따라서 매끈한 피부톤 = 젊음
- Oren-Nayar 쉐이딩 (고가) 와 “grisaille” 쉐이딩(저렴!)
- 개념 결합



One Modern Variation

Traditional Grisaille Relief

조명

- 조명을 받는 물체를 음영처리함 – 조명이 없는 것에 음영처리를 하는 것이 아님
- PS_3_0 초기 제품 사용
  - 보너스: “if” 사용하면 배치 크기 면에서도 유리
  - 쉐이더 하나를 작성하여 ps_3 또는 ps_2용으로 컴파일
- 지연(deferred) 쉐이딩의 경우, 조명 받는 픽셀에만 쉐이드 적용
- “글루미넌스(Gloominance)” 는 부동점 픽셀의 모든 경우에 대하여 항상 완벽히 안전함

Spotlight

스마트 조명 배치

- Magy Seif El-Nasr's "ELE": 표현형 조명 엔진
- http://ist.psu.edu/SeifElNasr/
- 로봇 공학의 로드 밸러싱 방정식을 이용하여 제한된 조명 세트만으로도 가시성과 "분위기"를 극대화

Mirage, El-Nasr et al, CIRA

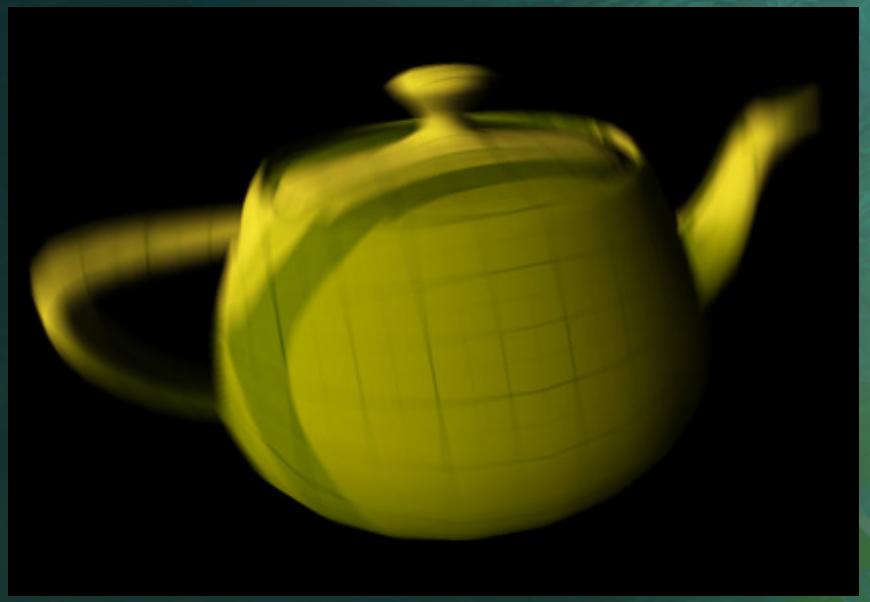
반사

- 상황에 따라 모든 Specular를 교체 가능
- CUBE 맵 생성을 위하여 VM 사용 가능
- 유한 반경 (finite radius) 가능 (나중에 설명)
- 2차 낙하(quadratic falloff)가 있는 거리 사용 가능 (나중에 설명)

환경매핑이 적용된 배경, 반사된 카드 모양의 광원,
overbright bloom의 16비트 블렌딩

새로운 영역: 카메라 효과

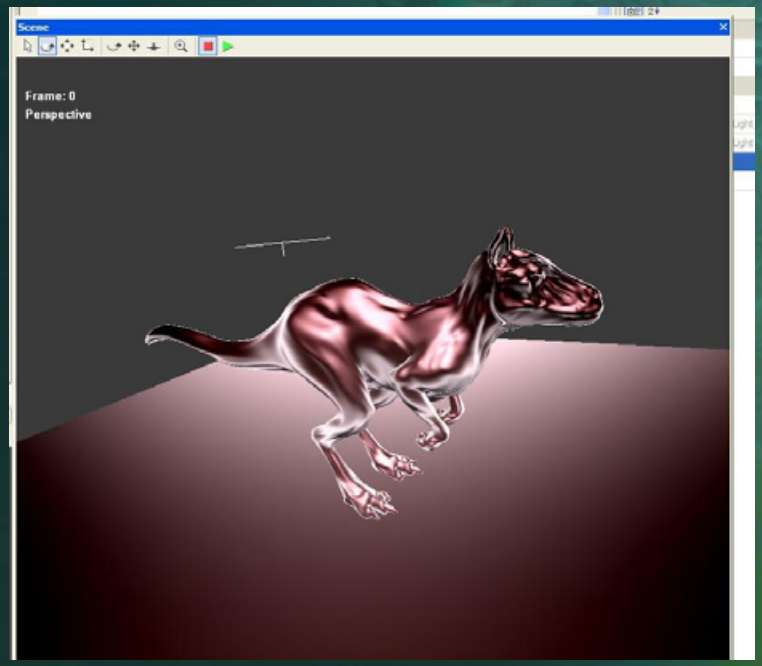
- "누적 버퍼" 기술의 장점:
  - 모션 블러
  - 필드 깊이
  - 부드러운 그림자
  - 기타
- 특별한 쉐이딩 요건은 없지만 쉐이더는 반드시 *빨라야* 함



Motion blur

Direct X VM 활용의 극대화

- 텍스쳐 생성
- CPU상의 “텍스쳐 쉐이더” 로 이미지 생성 또는 예측 가능한 기능을 포함하는 텍스쳐 생성 가능
- HLSI intrinsic을 이용한 행열 조작을 통해 복잡한 음영 처리도 기능적 현실화가 가능



Dinosaur with Physically-based car paint BRDF

조합 및 2D Effects

- FP 버퍼의 강력한 힘
- 재미…
- 색 제어
- 최종 “마무리”
- 모드 혼합
- 2D/3D 스프라이트 혼합
- 부동점(floating point) 픽셀

Halftoning Patterns

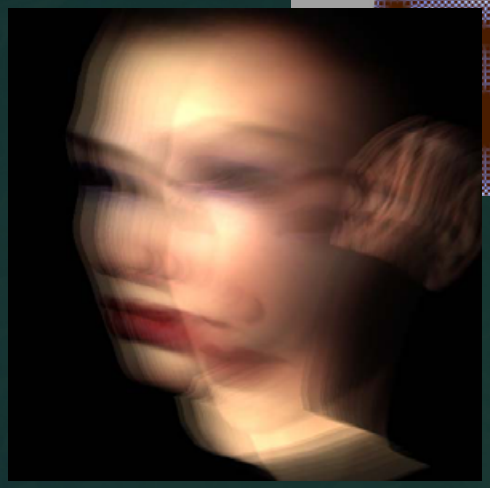
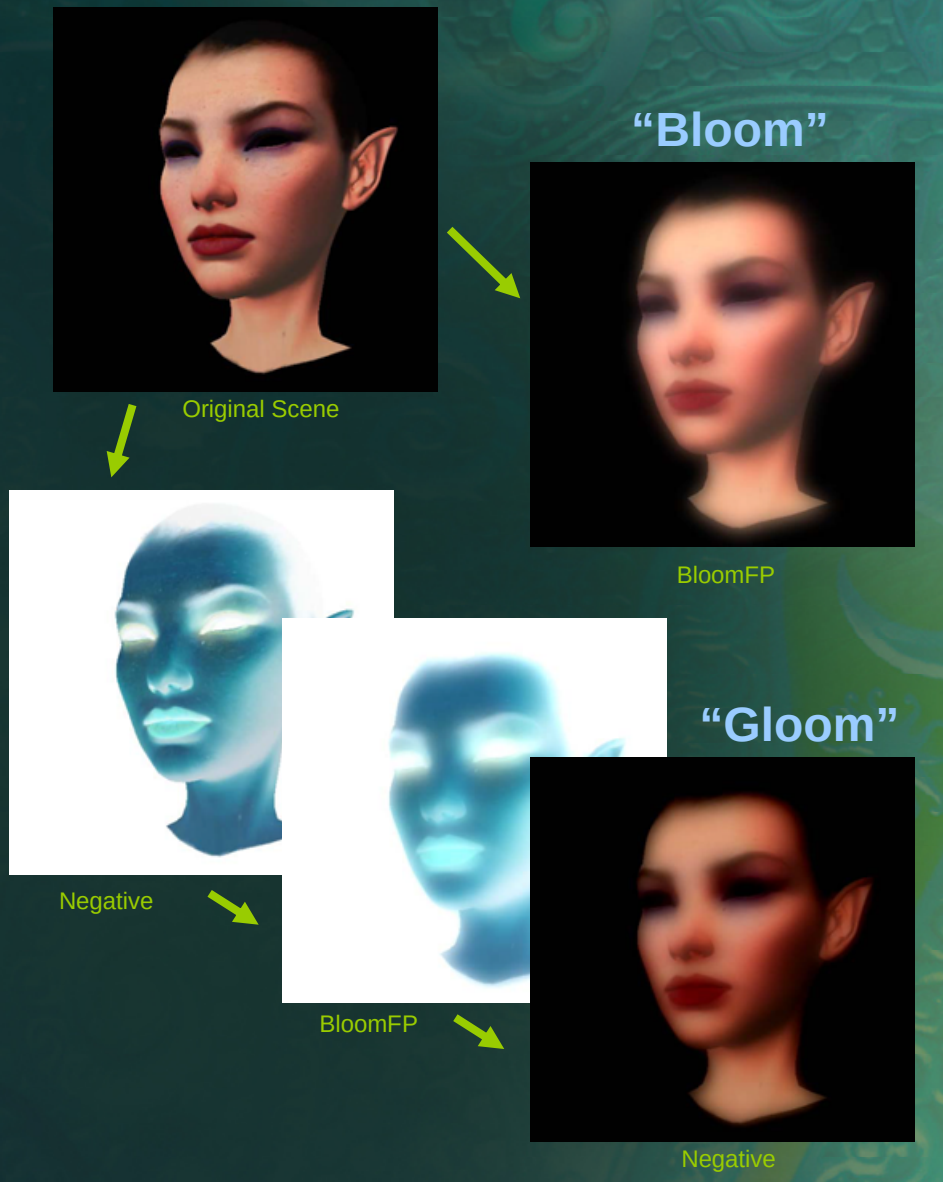


Image Trails

사후 처리:환함(bloom)&어두움(gloom)

- Glow와 “Overbright Bloom”로 복잡성과 스케일에 대한 환상 효과
  - 꼭 사용하게 된다는 점에서 녹음 스튜디오의 “ 에코”와 약간 비슷
- FX Composer 에서 이미지 효과를 겹침으로써 새롭고 더 복잡한 효과가 가능



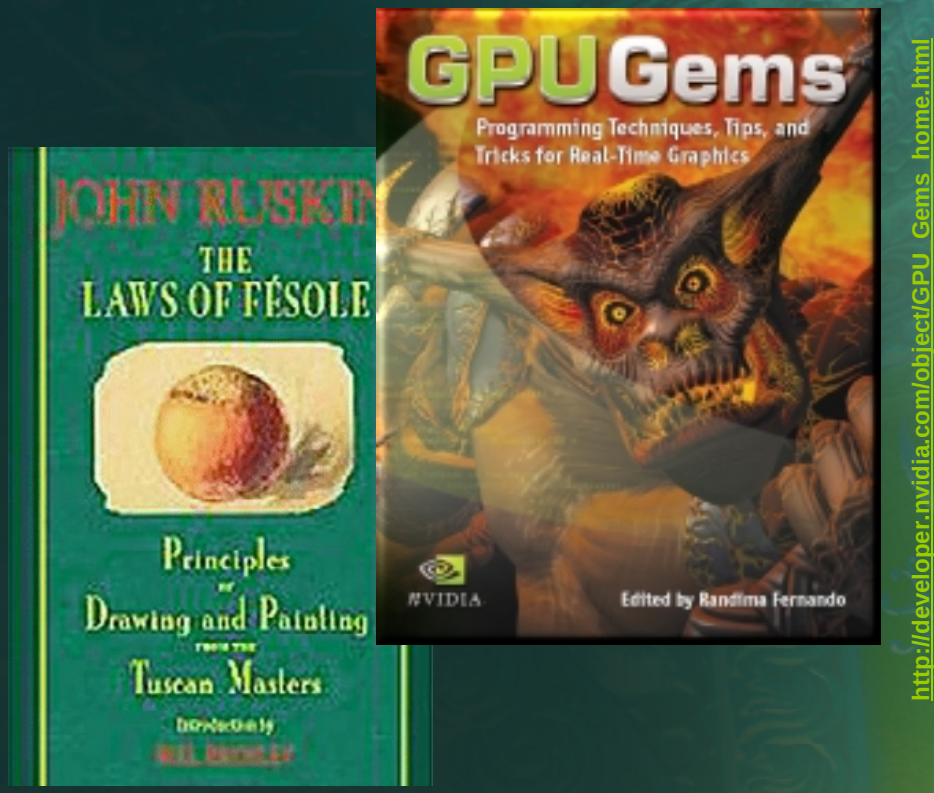
마무리!

The End

- 이제 게임도 영화수준의 쉐이딩에 필적할 만한 능력을 보유 – 영화와 비교하여 픽셀 대 픽셀 수준의 비교 뿐만 아니라 캐릭터의 그림자 처리면에서 우위를 가리기 힘들어지고 있음.
  - 수많은 쉐이더에 적응하십시오
  - 가지고 놀 수 있는 도구를 장만하십시오
  - http://www.fxcomposer.com/
  - 쉐이더를 가지고 뭐든지 실험해 보면서 유용한 아이디어의 " 스케치북" 마련

권장서적

- Randima Fernando: *GPU Gems*
- John Alton: *Painting with Light*
- Jon Ruskin: *The Laws of Fésole, Priciples of Drawing and Painting from the Tuscan Masters*



http://developer.nvidia.com/object/GPU_Gems_home.html

추가정보

- http://developer.nvidia.com/
- http://www.fxcomposer.com/
- http://developer.nvidia.com/object/sdk_effects.html
- kbjorke@nvidia.com