

컴퓨터 그래픽스

2023학년도 1학기

담당교수 : 마 준



- 연구실 : M508
- 이메일 : majun@sch.ac.kr
- 선행 과목
 - 알고리즘, C/C++ Programming, 자료구조, 선형 대수
- 교재
 - OpenGL 프로그래밍 가이드, Addison Wesley



숙 제: 10%

퀴 즈: 10%

출 석: 10%

중간 고사: 30%

기말 고사: 40%

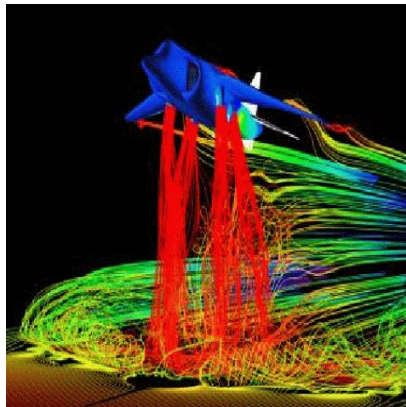
- 과제 열심히 해서 제출 할 것(가끔)
- 절대 Cheating 불가!!
 - Cheating 적발 시 "F"학점



- C++/C programming 실력?
- Window programming 경험?
- OpenGL programming 경험?
- Final exam or project?



컴퓨터 그래픽스 소개

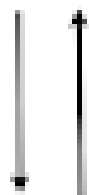




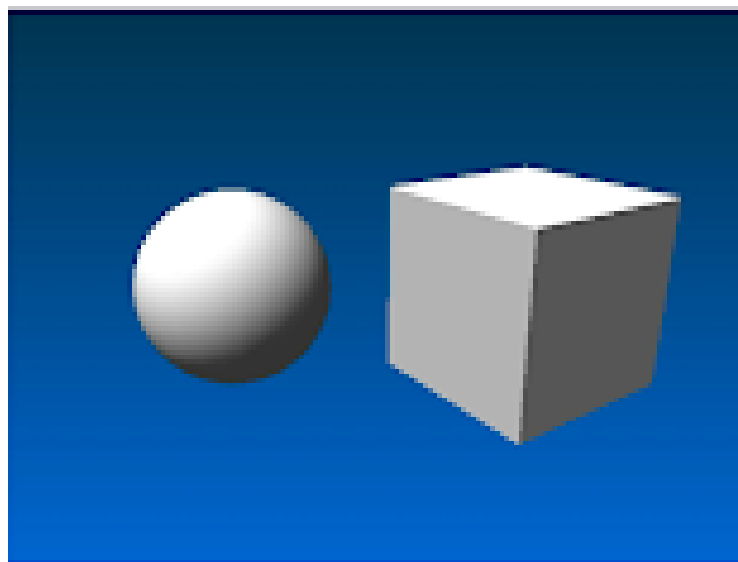
{cube, size, x_0 , y_0 , z_0 , θ_{xz} , θ_{xy} , θ_{yz} , ...}

{sphere, radius, x_1 , y_1 , z_1 , ...}

Computer
Graphics

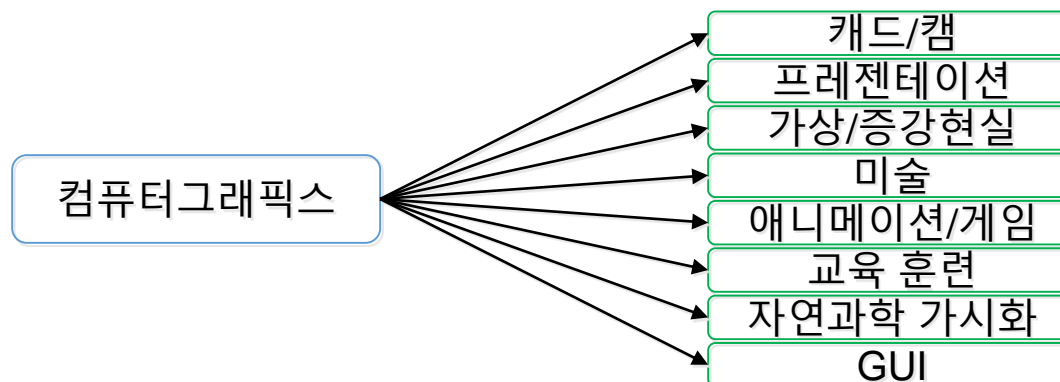


Computer
Vision





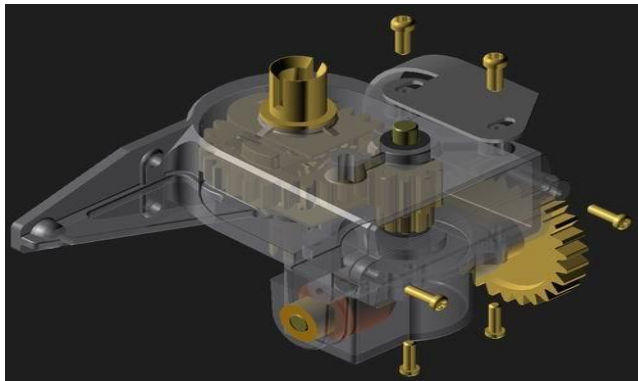
- 컴퓨터 그래픽스
 - 컴퓨터를 사용하여 그림을 생성하는 기술
 - 수작업으로 그림을 그리던 과거와 달리 경제적인 비용으로 그림을 생성 or 창조
- 컴퓨터 비전
 - 그림을 통해 데이터를 분석하는 기술
- 영상처리: 이미 있는 그림을 개선하거나 인식하는 분야
- 그래픽스 응용분야



[그림 1-1] 그래픽스 응용분야



- CAD(Computer Aided Design): 건물, 자동차, 비행기, 선박 등의 설계에 이용, 시뮬레이션도 가능
 - 과거 제도 용구를 사용한 수작업을 컴퓨터 그래픽으로 대체하여 설계에 필요한 인력, 시간, 노력 등을 단축하여 설계효율 향상
- CAM(Computer Aided Manufacturing)
 - CAD로 설계된 데이터가 직접 Numerical Control Machine으로 입력되어 제품가공
- 자동생산에 따른 경제적 효율과 가공의 정밀도 향상



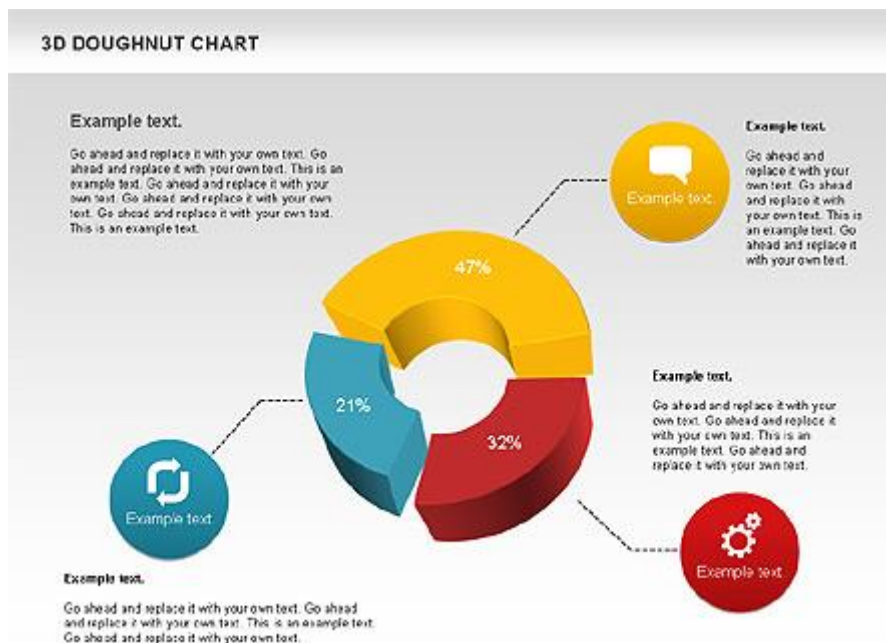
[그림 1-2] 기계부품 렌더링



[그림 1-3] 자동차 겉모습 렌더링



- 막대 차트(Bar Chart), 선 그래프(Line Chart), 파이 차트(Pie Chart), 입체 그래프(Surface Graph)
- 백문(百聞)이 불여일견(不如一見)



[그림 1-4] 시연용 슬라이드 예



[그림 1-5] 레이 트레이싱에 의한 렌더링



- Virtual Reality: 존재하지 않는 가상의 환경을 구성하되 그것이 마치 현실과 똑같이 느껴지도록 만드는 것
 - 공사 이전에 설계된 건물을 미리 가상현실 기술로 구현
- Augmented Reality: 실제하는 환경 위에 가상의 물체를 띄워 혼재하도록 만드는 것
- 기술적 요소
 - 사실적인 입체화면, 3차원 입체 음향, 촉각을 느끼게 하는 장비
 - 장면 데이터베이스, 그래픽 소프트웨어



[그림 1-6] HMD



[그림 1-7] 데이터 글로브



[그림 1-8] 데이터 글러브



- 상업적 또는 순수 예술 목적으로 그래픽을 활용
- 무선 스타일러스 펜: 누르는 압력으로 선의 굵기, 명암 조절가능
- 그래픽 소프트웨어를 통해 수채화나 파스텔, 유화 효과 가능



[그림 1-9] 촬영 영상



[그림 1-10] 파스텔 처리



■ Ants



[그림 1-11] 벅스 라이프



[그림 1-12] 슈렉

소요 내역	양
총 프레임(정지화면) 수	119,592 개
주당 렌더링에 소요된 시간	275,000 시간
평균 정지 프레임 크기	6 MB
렌더링에 사용된 실리콘 그래픽 서버 수	270 대
프로세싱에 사용된 데스크 탑 컴퓨터 수	166 대
프로세서 당 평균 메모리 용량	156MB
1개의 프로세서로 제작할 경우의 소요시간	약 54년
영화 저장을 위한 보조기억 장치 용량	3.2 TB
매 순간 온 라인으로 공유된 프레임 수	75,000 개

[표 1-1] Ants의 제작에

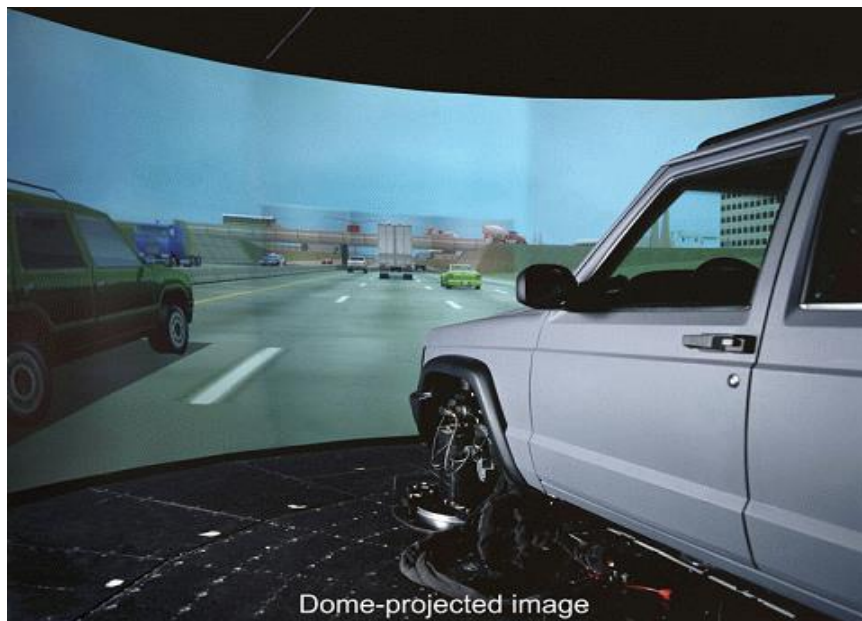


[그림 1-13] 3D 온라인 게임

- 캐릭터, 배경화면, 애니메이션 등에 모두 컴퓨터 그래픽스 기술 사용
- 사용자와 프로그램 사이의 상호작용 설계가 아주 중요
- 상호작용에 걸리는 시간의 최소화가 중요 요소
- 시장성 면에서 볼 때 무한한 가능성 존재



- 시뮬레이션: 다양한 상황들을 경험 가능



[그림 1-14] 운전 시뮬레이션

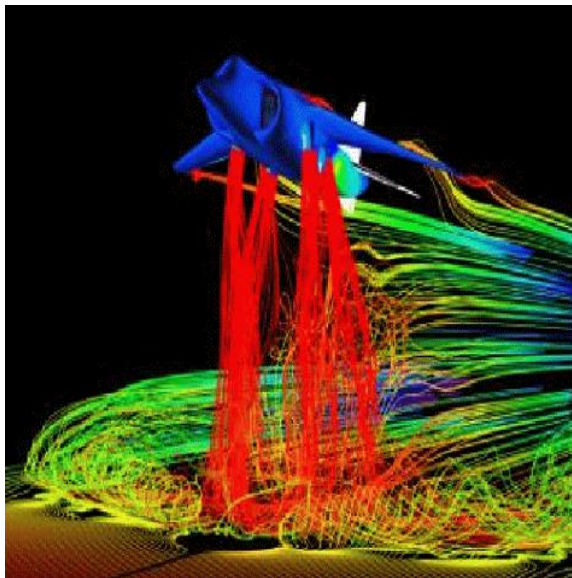
[그림 1-15] 비행 시뮬레이션



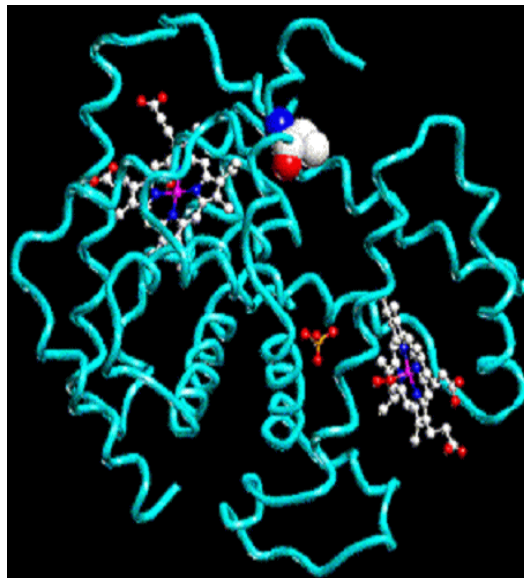


■ SCI VIS(Scientific Visualization)

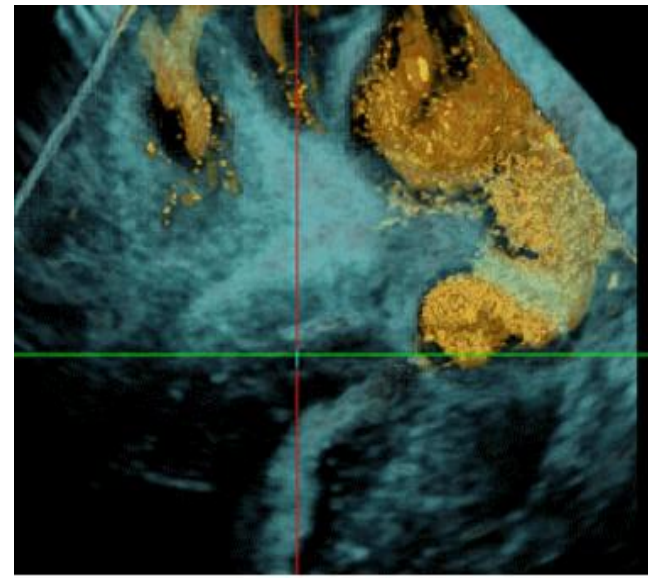
- 과학 및 공학 분야에서 발생하는 제반 현상을 그래픽으로 분석
- 대용량의 정보분석에 용이
- 자연현상을 시각화하여 현상 내부의 패턴이나 추세를 직관적으로 파악



[그림 1-17] 기류 분석



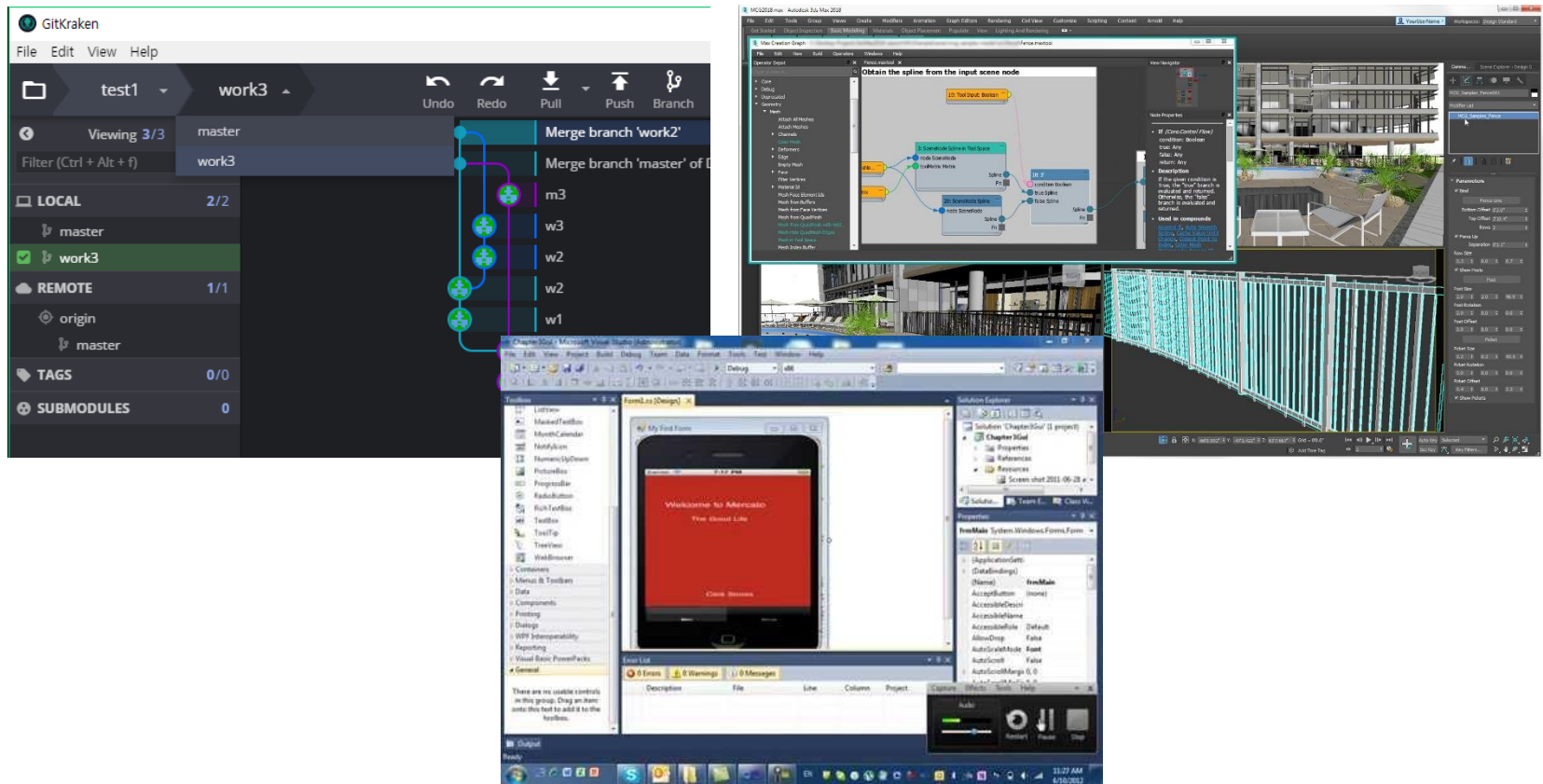
[그림 1-18] 분자 구조 를 가시화



[그림 1-21] 초음파 촬영 3차원 렌더링



- GUI(Graphic User Interface): 메뉴, 스크롤 바, 아이콘, ... VS DOS
- 사용자 편의를 고려해 렌더링을 수행해야 함



[그림 1-22] 다양한 상용 툴의 GUI 환경

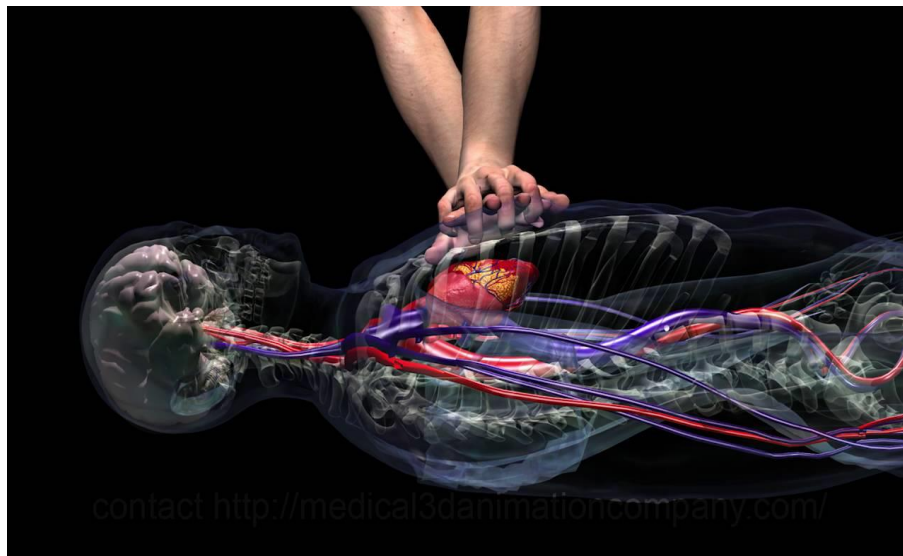
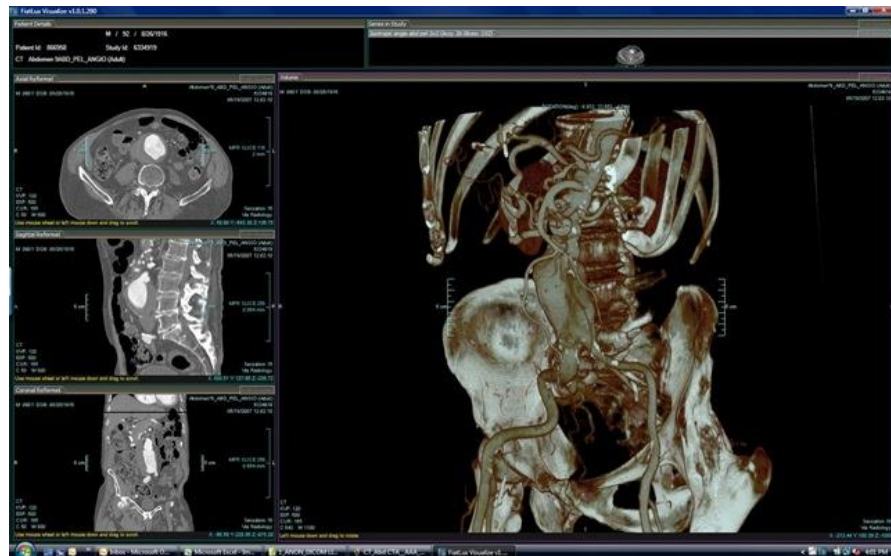




Photo Realistic Images





Forrest Gump



- 아이반 서더랜드(Ivan Sutherland)
 - 컴퓨터 그래픽의 창시자
 - 대화형 컴퓨터 그래픽 개념: 라이트 펜으로 의사전달
 - 스케치패드 프로젝트: 컴퓨터가 스케치 북 역할을 대신
 - 직선, 원호 등 기본적 그래픽 요소를 사용하여 물체를 표현하는 방법
 - 계층구조 모델링: 기본물체를 조합하여 큰 물체를 모델링
 - 물체를 선택하여 이동하는 방법(Dragging)
 - 팝업 메뉴에 의한 사용자 입력

1960	William Fetter	“컴퓨터 그래픽”이란 용어를 최초로 사용
1963	Ivan Sutherland	컴퓨터 그래픽의 제반 개념을 확립
1963	Douglas Englebart	최초의 마우스 프로토타입
1965	Jack Bresenham	선분 그리기 알고리즘을 개발



- 그래픽 알고리즘의 시기: 많은 알고리즘의 개발로 그래픽 국제 표준의 필요성 논의

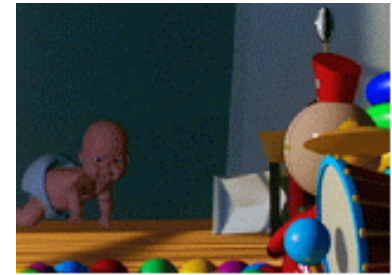
1971	Gouraud	구로 셰이딩 알고리즘
1973	John Whitney Jr.	컴퓨터 그래픽에 의한 최초의 영화 "West World"
1974	Edwin Catmuff	텍스처 매핑, 지-버퍼 알고리즘
1974	Bui-Tong Phong	전반사에 의한 하이라이트 알고리즘
1975	Martin Newell	베지어 표면 메쉬를 사용한 차 주전자 모델
1975	Benoit Mandelbrot	프랙탈 이론
1976	Jim Blinn	주변 매핑, 범프 매핑 이론
1977	Steve Wozniak	컬러 그래픽 PC: Apple II
1977	Frank Crow	앤티 에일리어싱 알고리즘
1979	Kay, Greenberg	최초로 투명한 물체 면을 그려냄



- PC의 시대
- 래스터 그래픽 하드웨어
- 기하 엔진(Geometry Engine) 출현



[그림 1-24] Luxo Junior



[그림 1-25] Tin Toy

1980	Turner, Whitted	광선 추적 알고리즘
1982	Steven Lisberger	3차원 그래픽 애니메이션 "Tron"
1982	John Walkner, Dan Drake	"AutoCAD"
1983	Jaron Lanier	데이터 장갑을 사용한 가상현실 영화
1985	Pixar	"Luxo Junior"
1985	NES	가정용 게임 "Nintendo"
1986	Steve Jobs	Lucasfilm사의 Pixar 그래픽 그룹을 인수
1987	IBM	VGA 그래픽 카드
1989	IBM	SVGA 그래픽 카드
1989	Pixar	"Tin Toy" 아카데미상 수상



- 사실적(Photo-realistic) 그래픽 영상에 주력



[그림 1-26] NVIDIA GeForce 256

1990	Pixar: Hanrahan, Lawson	렌더링 소프트웨어 "Renderman" 개발
1990	Gary Yost	3-D Studio 개발
1991	Disney and Pixar	"Beauty and the Beast"
1992	Silicon Graphics	OpenGL 사양 발표
1993	Steven Spielberg	"Jurassic Park"
1995	Pixar	"Toy Story" 100% 컴퓨터그래픽으로 제작
1995	Microsoft	DirectX API 사양 발표
1996	John Carmack, Michael Abrash	Quake 그래픽 엔진 개발
1999	NVIDIA	GeForce 256 GPU



- 영화, 게임 응용의 시기



2001	Square	"Final Fantasy: The Spirits Within"
2003	Timothy Purcell	광선추적 기법을 GPU에 적용
2004	Id Software	Doom Engine 발표
2004	DirectX, OpenGL	DirectX SDK, OpenGL 2.0 발표
2007	OpenGL	OpenGL ES 2.0 발표
2009	DirectX	DirectX 11 발표



- 병렬처리 / 게임 엔진의 보급화



2011	OpenGL	OpenGL 4.2 발표
2012	OpenGL	OpenGL 4.3 발표 / 병렬처리 셰이더 추가
2014	Unity 3D	Unity 4 발표 및 멀티플랫폼 기능 제공
2015	Unreal	Unreal Engine 4 무료화



■ 2가지 구성요소



[그림 1-28] 그래픽스 구성요소

- 모델링: 무엇을 그릴까?
 - 장면 내부 물체를 정의하는 작업
 - 선분의 끝점, 다각형의 정점을 정의
 - 2,3차원 물체를 표현할 수 있는 자료구조와 이를 처리할 수 있는 알고리즘 포함
- 렌더링: 정의된 물체를 어떻게 그릴까?
 - 조명의 위치, 관찰자의 위치, 3차원 물체를 어떻게 2차원으로 사상, 물체표면 재질 등
 - 와이어 프레임 렌더링, 솔리드 렌더링



- 컴퓨터 그래픽은 실재(reality)를 속이는 것
- 실재의 물체와 분간이 안 되도록 컴퓨터 그래픽을 이용한 합성 물체를 만들어 내는 것
- 그렇게 생성된 물체들을 사실감 있도록 움직이게 하는 것
- 이러한 작업들은 가능하다면 실시간으로 실행적이고, 과학적으로 보이도록 하는 것임



- 모델링 (Modeling)
 - 물체를 어떻게 형성할 것인가?
 - 물체들을 어떻게 디자인 할 것인가?
 - 3차원 세계에서 물체들의 형태를 기하학적 좌표를 이용하여 생성하고 표현
- 애니메이션 (Animation)
 - 물체가 움직이는 것을 표현하고 통제
 - 시간 변화에 물체가 어떻게 변하는지를 기술
- 렌더링 (Rendering)
 - 이미지가 어떻게 형성될지를 표현
 - 2차원의 물체 이미지들을 생성



- 물체를 어떻게 표현할 것인가?
 - 기하학적 모델링 (Geometric Modeling)
 - 기본 요소 (Primitives : point, line, face, curve, curved surface)
 - Convex, Non-convex objects
 - 관절로 연결된 물체 표현 (Articulated Objects)
 - 광도 모델링 (Photometrical Modeling)
 - 빛 (Light)
 - 색 (Color)
 - 반사 (Reflection), 굴절 (Refraction)



- 정의 모델링 (Declarative modeling)
 - 정점 (Vertex in 3D as x, y, z)
 - 선 (Line in 3D as two vertices)
- 대화식 모델링 (Interactive modeling)
 - free form drawing
- 센싱 (Sensing)
 - Image scanning
 - Volume scanning



- 물체가 어떻게 움직일지를 모델
- 움직임을 어떻게 표현할 것인가?
 - 이미지들의 연속
 - 변수에 의한 표현
- 움직임에 대한 서술
 - manually : 관찮게 보이도록 미세 조정
 - Key framing
 - 규칙에 의한 표현 (rule based)
 - physically based
 - motion capture



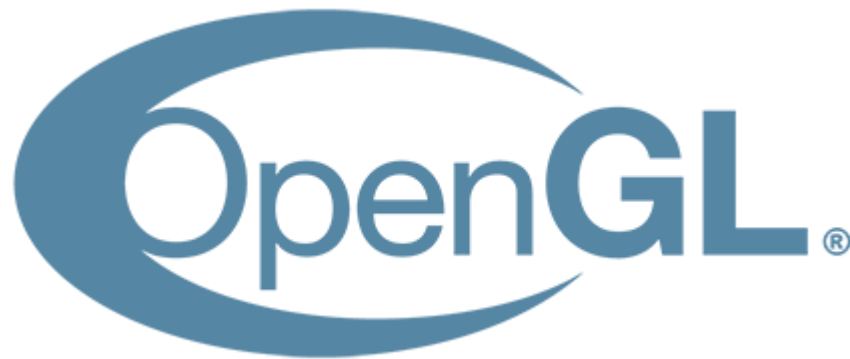
- 이미지란?
 - 2차원 필름에 빛 에너지가 분포
- 어떻게 이미지를 저장하고 표현?
 - sampled array of pixels
- 어떻게 광경으로부터 이미지를 생성?
 - Input : description of scenes, sensing
 - Solve light transport through environments
 - Project to camera's viewpoint
 - ray tracing



- Direct X
 - Microsoft사에서 배포한 그래픽 라이브러리
 - Windows에 종속적인 특징
- Vulkan
 - OpenGL NG, glNext 등으로 불리던 그래픽 라이브러리
 - 크로스 플랫폼 2D/3D 그래픽 라이브러리
 - 정보가 적은 편



- 범용적 그래픽 렌더링 API
 - 그래픽 소프트웨어 라이브러리
 - 그래픽 하드웨어에 접근하기 위한 소프트웨어
 - 1980년대 실리콘 그래픽스(SGI)에 의해서 개발됨
 - 2006년 크로노스 그룹에 의해 관리받는 것으로 이전됨





- Windows system에 독립적
- Operating system에 독립적
- Graphics programming의 표준
- 많은 platforms에서 사용 가능
- 빠르고 이식성이 좋은 open library
- 성능이 좋은 (hardware-accelerated) 그래픽 API



- 기하 기본요소 (Geometric primitives)
 - Points, lines, polygons
- 이미지 기본요소 (Image primitives)
 - Images and bitmaps
 - Separate pipeline for images and geometry
 - Texture mapping
- State dependent rendering
 - Color, materials, light sources etc.



- GLU (OpenGL Utility Library)
 - 많은 모델링을 제공: quadric surfaces, NURBS, tessellators
- GLUT (OpenGL Utility Toolkit)
 - 윈도우와 user interface API
 - 편리한 멋진 형태의 primitives 제공 (torus, teapot, cube)
- GLEW (OpenGL Extension Wrangler Library)
 - OpenGL 확장 라이브러리로 셰이더 프로그램을 작성할 때 사용
- GLFW (OpenGL Library for Windows)
 - Windows 프로그램을 생성하기 위해 사용하는 라이브러리 (GLUT와 유사)



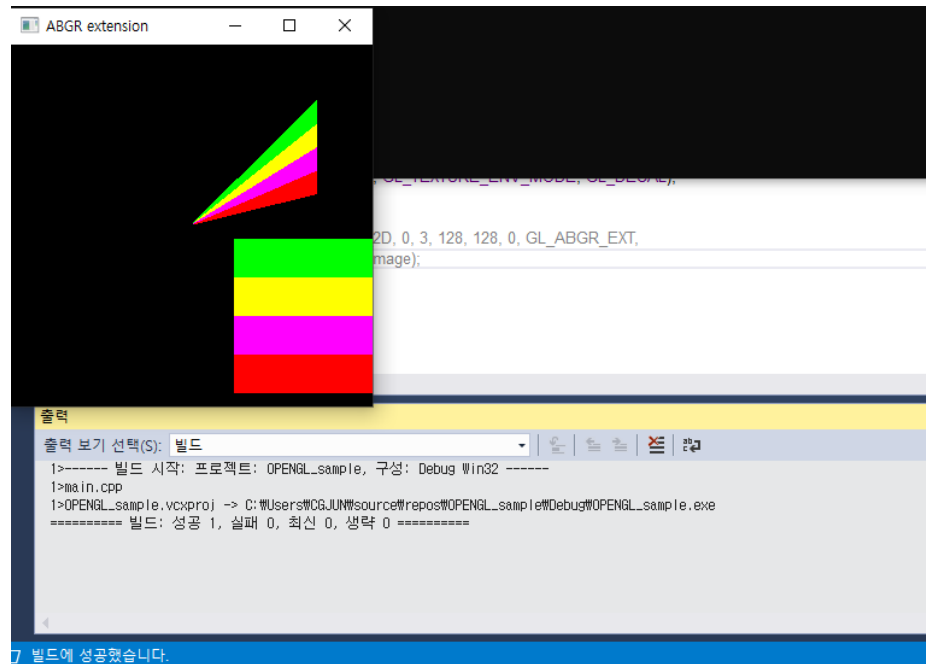
- GLUT을 과목 사이트에서 다운 받을것

각 파일 해당하는 경로에 붙여 넣기

- glut.h -----> C:\Program Files (x86)\Microsoft Visual Studio\2019\Community\VC\Tools\MSVC\{14.16.27023}\include\GL
- glut.dll, glut32.dll -----> C:\Windows\SysWOW64
- glut32.dll -----> C:\Windows\System32
- glut.lib -----> C:\Program Files (x86)\Microsoft Visual Studio\2019\Community\VC\Tools\MSVC\{14.16.27023}\lib\x64
- glut32.lib-----> C:\Program Files (x86)\Microsoft Visual Studio\2019\Community\VC\Tools\MSVC\{14.16.27023}\lib\x86



- 예제 코드 복사
- https://www.opengl.org/archives/resources/code/samples/glut_examples/examples/abgr.c
- ctrl + f5 로 구동 확인





- OpenGL의 공식 사이트(영문)
 - <http://www.opengl.org>
- OpenGL API Tutorial을 제공하는 nehe.gamedev.net(영문)
 - <http://nehe.gamedev.net>
- 현직 개발자 개인 사이트, OpenGL에 대한 자료와 nehe.gamedev.net의 번역자료도 보유하고 있음
 - <http://www.gisdeveloper.co.kr/category/Programming/OpenGL>
- 델파이를 통한 OpenGL – OpenGL의 기본 지식에 대해 자세히 설명되어있음
 - http://www.delphinara.com.ne.kr/open_r.htm
- 소엔 랩
 - <http://soen.kr/>



- 3D Studio Max를 이용하여 본인이 원하는 멋진 모형을 삼각형 메쉬를 이용하여 3D로 모델링 할 것(학생 버전 사용)
- 모델링한 결과를 obj파일 형식으로 저장하여 obj 파일을 eclass 시스템에 올릴 것(~03/14)
- 제작 과정 및 모델링 결과, 느낀점은 레포트 형식으로 작성하고 출력하여 3월 15일(수) 수업시간에 제출 할 것

THANK

YOU