

“본 강의 동영상 및 자료는 대한민국 저작권법을 준수합니다. 본 강의 동영상 및 자료는 상명대학교 재학생들의 수업목적으로 제작·배포되는 것이므로, 수업목적으로 내려받은 강의 동영상 및 자료는 수업목적 이외에 다른 용도로 사용할 수 없으며, 다른 장소 및 타인에게 복제, 전송하여 공유할 수 없습니다. 이를 위반해서 발생하는 모든 법적 책임은 행위 주체인 본인에게 있습니다.”

# 오리엔테이션

GPU Programming

2022학년도  
2학기



# 과목 개요

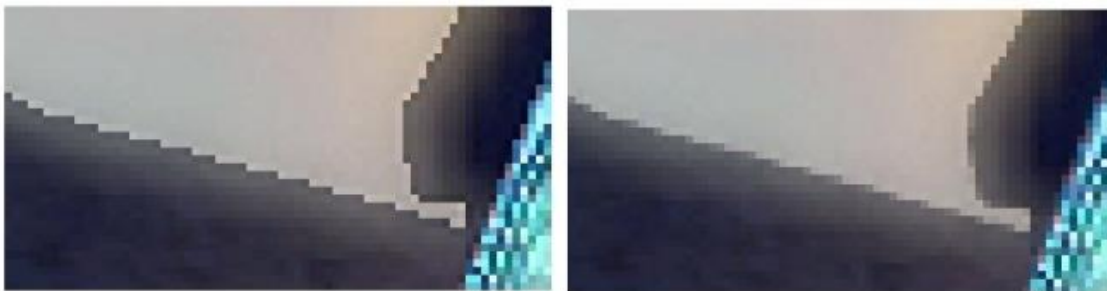
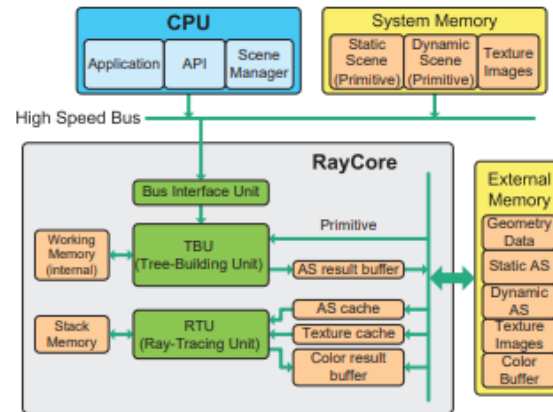
# GPU 프로그래밍 과목의 개요 및 목표

- 교과목 개요
  - 2022학년 2학기 신규 개설 과목
  - GPU(graphics processing unit)를 이용해 그래픽스 프로그래밍을 하는 방법을 공부하는 과목
- 교과목 목표
  - 컴퓨터 그래픽스, 그 중에서도 렌더링과 관련된 여러가지 이론 소개
  - 이러한 이론을 실제 프로그래밍에 적용해 보기 위해 OpenGL API 문법을 학습하고, 관련 예제를 통해 실습을 수행
  - 이론과 실습을 통해 궁극적으로 자신이 컴퓨터 화면에 원하는 것을 그릴 수 있는 능력을 배양
- 수강 대상: 컴퓨터과학과 또는 인공지능융합전공 3, 4학년
  - 필수 선수 과목은 없음
  - C++ 언어를 기반으로 하므로, C와 JAVA에 대한 경험이 있으면 좋음  
(관련 과목: C프로그래밍1/2, 객체지향프로그래밍)
  - 행렬 연산 등이 실제로 코드에 적용되기 때문에, 이와 관련된 기초 수학 지식이 있으면 좋음  
(관련 과목: 컴퓨터수학, 선형대수학 등)



# 담당 교수 연구 분야 및 약력

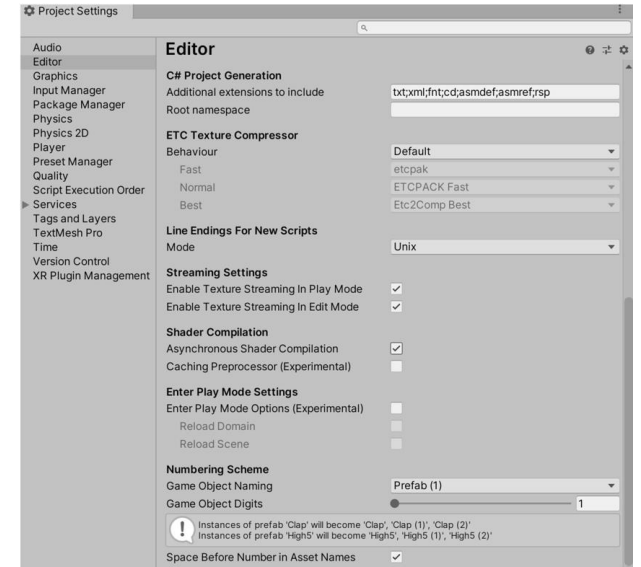
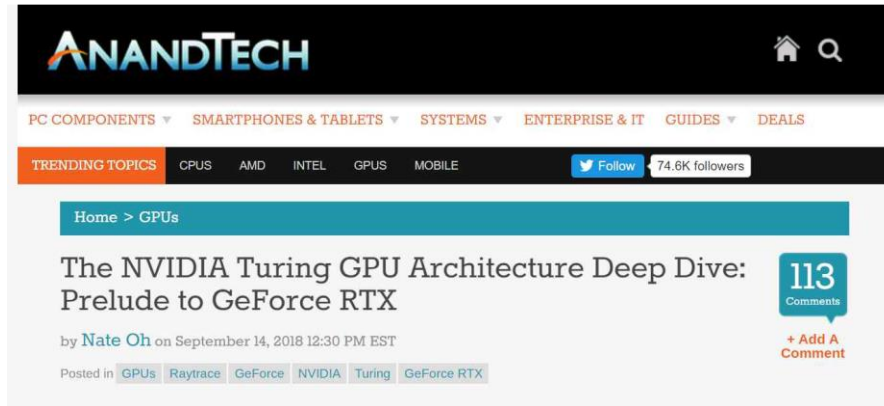
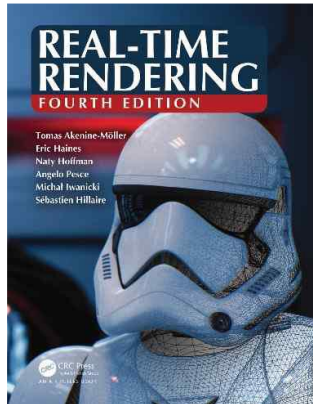
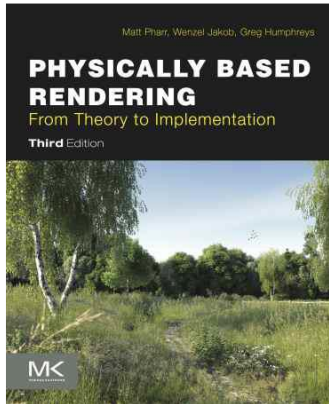
- 연구 분야: 컴퓨터 그래픽스, 그 중에서도 GPU 및 렌더링 위주로 연구
  - GPU 아키텍처, 레이 트레이싱, 안티 알리아싱, 텍스처 압축 등



ORIGINAL   OURS   ETCPACK  
0.14 ms   392 ms

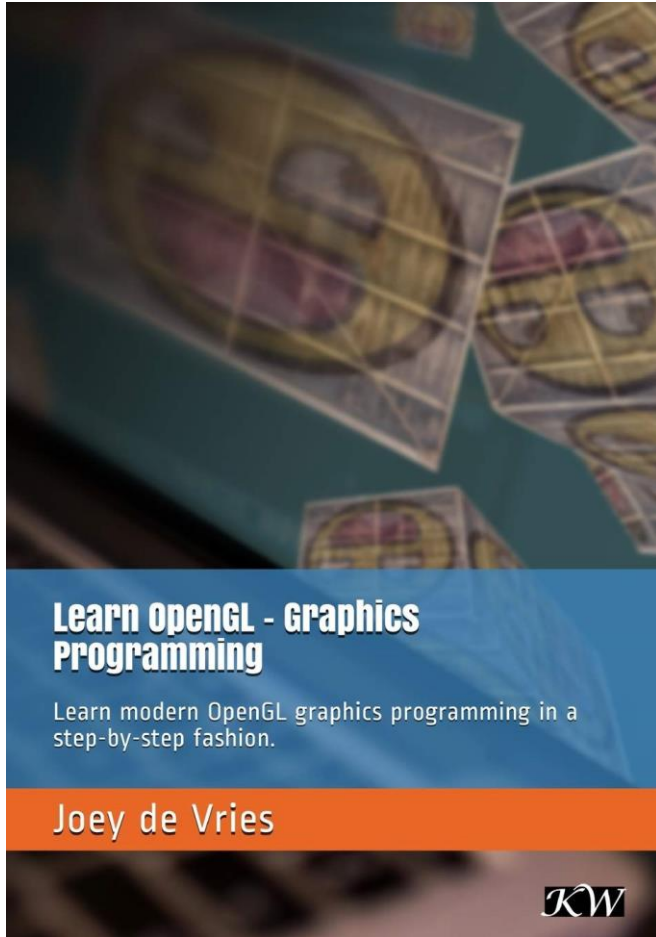
# 담당 교수 연구 분야 및 약력

- 그래픽스 분야의 최고 학회인 ACM SIGGRAPH/SIGGRAPH Asia에서 3편의 technical paper 발표
  - T&I Engine (2011), RayCore (2014), QuickETC2 (2020)
  - 첫번째와 두번째 논문은 렌더링 분야의 저명 교과서 및 AnandTech 기사에서 인용
  - 세번째 논문은 Unity에 내장되는 텍스처 인코더인 etcpak의 정식 버전(1.0)에 반영



- 기타 유명 학회/논문지에 다수 논문 발표 및 심사
  - 그래픽스 분야 3대 논문지(ACM TOG, IEEE TVCG, CGF)에 논문 발표
  - 그래픽스 분야 3대 학회(ACM SIGGRAPH, ACM SIGGRAPH Asia, EUROGRAPHICS) 논문 심사
- 주저자로 쓴 모든 논문은 개인 홈페이지에서 다운 가능 [Nah, Jae-Ho \(nahjaeho.github.io\)](http://Nah, Jae-Ho (nahjaeho.github.io))

# 주교재 소개



- Joey de Vries, Learn OpenGL: Learn modern OpenGL graphics programming in a step-by-step fashion, Kendall & Welling, 2020.
  - [learnopengl.com](https://learnopengl.com)에서 책 pdf 및 소스 코드 다운로드 가능
- 종이책 구입을 원할 때에는 쿠팡 직구 이용 가능
  - [쿠팡! | Learn OpenGL: Learn modern OpenGL graphics programming in a step-by-step fashion. Paperback \(coupang.com\)](https://coupang.com)
- 개인 번역본도 존재 (일부 챕터만 번역)
  - <https://heinleinsgame.tistory.com/category/OpenGL>
- 강의 자료는 100% 손수 제작
  - 교재 내용 대비 다소 간소화된 부분이 있을 수 있습니다.



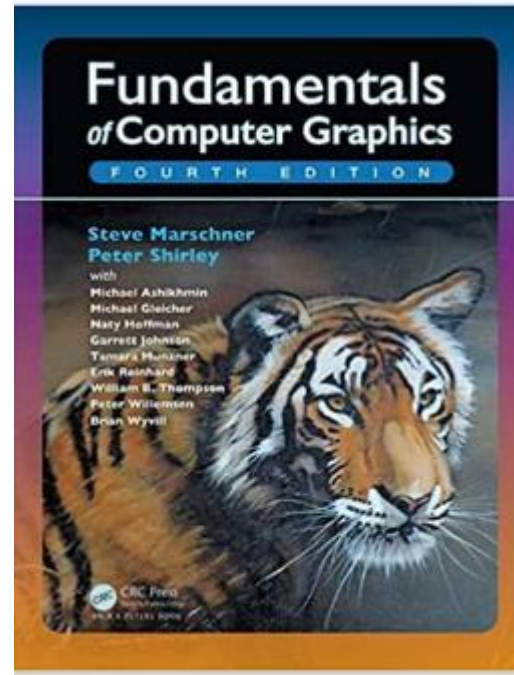
# 부교재 소개 (강의자료에 일부 내용 발췌 가능)

SUNG-EUI YOON, KAIST

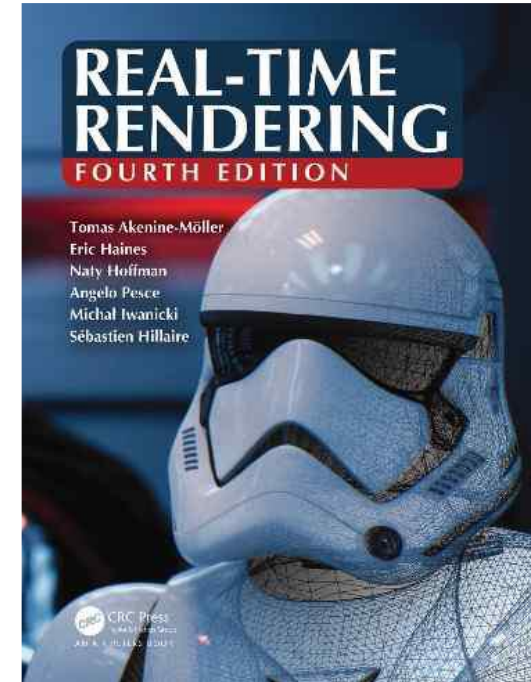
RENDERING

FREELY AVAILABLE ON THE INTERNET

[Rendering \(kaist.ac.kr\)](http://rendering.kaist.ac.kr)



**Fundamentals of Computer Graphics, 4th Edition**



**Real-Time Rendering, 4th Edition**

# 수업 방법

- 이론/실습 혼합
  - 목요일 8, 9교시: G210 (이론)
  - 화요일 6교시: G206 (실습)
- 실습 방식
  - 본인의 노트북을 가져와서 실습해도 되고, 실습실 컴퓨터를 사용해도 됨  
(GPU가 내장되어 있지 않은 컴퓨터는 없으므로, 어느 노트북을 사용해도 무방)
  - 실습실 컴퓨터를 사용할 경우, 1학기동안 가능한 한 같은 자리 사용을 권장 (프로그램 설치 등의 이유).



# 성적 평가

- 성적 평가 비율
  - 중간고사 30%
  - 기말고사 40%
  - 프로젝트 20%
  - 출석 10%
- 등급 비율은 타 과목과 동일 (상대평가1)
  - A 30% 이하, B 40% 이하, C/D/F 30% 이상
  - 결석이 1/4 이상(중간/기말고사 포함)일 경우 F
  - 중간&기말고사 모두 아예 응시하지 않으면 F, 응시했어도 결과가 0점이면 F 가능성

# 성적 평가

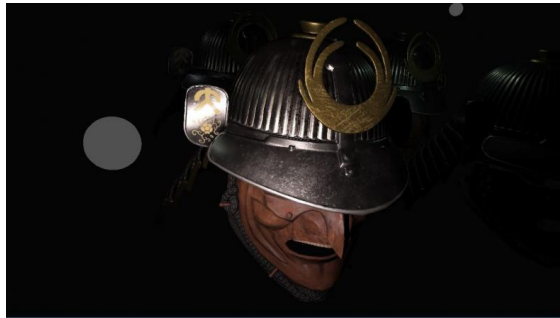
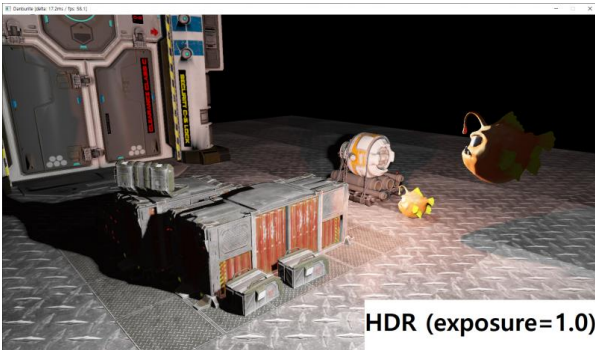
- 중간/기말고사는 해당 주간 목요일 8, 9교시에 오프라인 지필고사로 볼 예정
- 출석
  - 결석 1회당 1점씩 감점
  - 지각 3회 = 결석 1회
  - 결석 1회를 모두에게 허용하며 (감점 없음), 1주차 출석은 출석점수에 포함 안 함
  - 공결신청시 att.smu.ac.kr에 증빙서류 첨부 (코로나19 격리시에도 '질병으로 인한 미출석' 선택)
- 프로젝트 외 별도 과제물은 없음
  - 매 주 이캠퍼스에 코딩 실습할 예제를 업로드할 예정

# 프로젝트

- 개인이 진행하거나 2인 단위 팀 구성
  - 가능한 한 개인 프로젝트와 2인 프로젝트 결과물을 따로 채점
- 자유 주제
  - 단, 교재 예제를 확장한 형태여야 함 (C++ / OpenGL / Assimp 사용)
  - 오픈 소스 사용 무방 (출처는 반드시 명시)
- 중간고사 이후 9주차에 프로젝트 계획 제출
  - 팀 구성 내역 (개인 또는 2인), 주제, 간단한 구현 계획을 서술
  - 계획서 내용은 평가에는 반영되지 않음
- 12주차까지 코드 및 발표자료 제출, 13주차 목요일에 발표
  - 팀당 발표시간을 2-3분으로 제한 (30여팀 \* 3분 = 90분)
  - 주제 소개 및 실제 데모 시연
  - 발표력은 평가하지 않고 구현 내용으로만 평가
  - 한국어 대신 영어로 발표해도 무방

# 프로젝트

- 프로젝트 주제 예시
  - 최신 발표된 논문의 기법을 적용하여 교재의 코드 대비 성능 또는 화질을 향상
  - 다른 모델들을 import하고 교재의 여러 코드를 조합하여 현실감 있는 화면을 렌더링
  - 셰이더 코드 수정을 통한 여러가지 post-processing 효과 적용
  - 게임 엔진에서 구현한 내역을 OpenGL을 이용해 low-level로 다시 구현



learnopengl.com의 코드를 개발자들이 직접 적용한 후 게시글의 답글로 첨부한 화면

- 공개 모델 다운로드
  - [McGuire Computer Graphics Archive \(casual-effects.com\)](http://casual-effects.com)
  - [KAIST Model Benchmarks](http://kaist.ac.kr/model-benchmarks)
  - [Ingo Wald: UT AnimRep \(utah.edu\)](http://www.cs.utah.edu/~wald/)



# Q & A

---

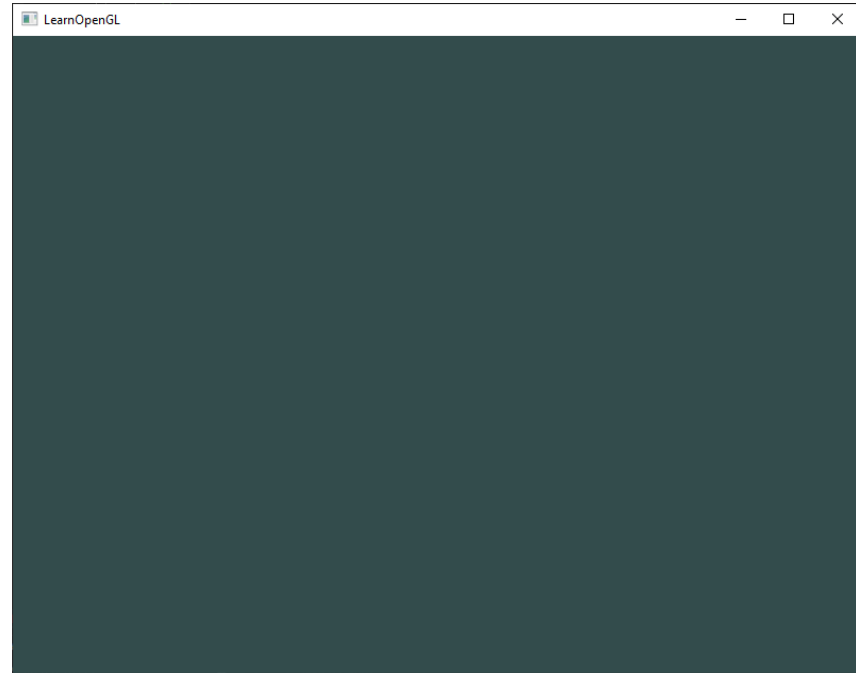
- 과목 운영에 대해 문의 사항은 e-Campus 상의 Q&A 게시판 또는 1:1 메시지로 질문 바랍니다.
- 면담 약속을 잡고 싶을 경우 이메일([jaeho.nah@smu.ac.kr](mailto:jaeho.nah@smu.ac.kr))로 연락 바랍니다.



# 주차별 강의 내용

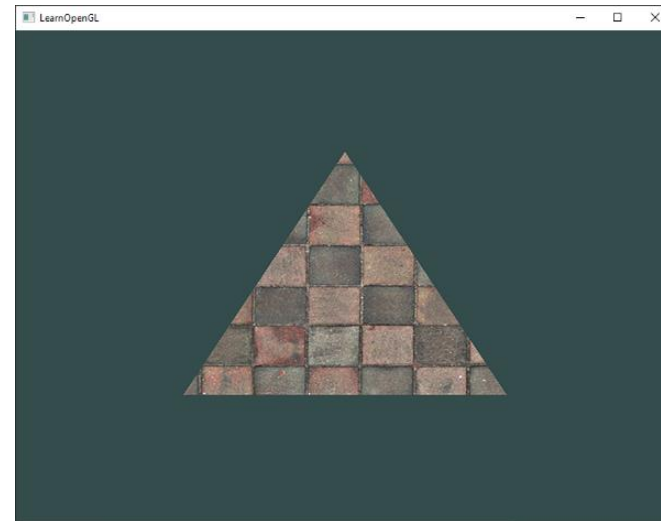
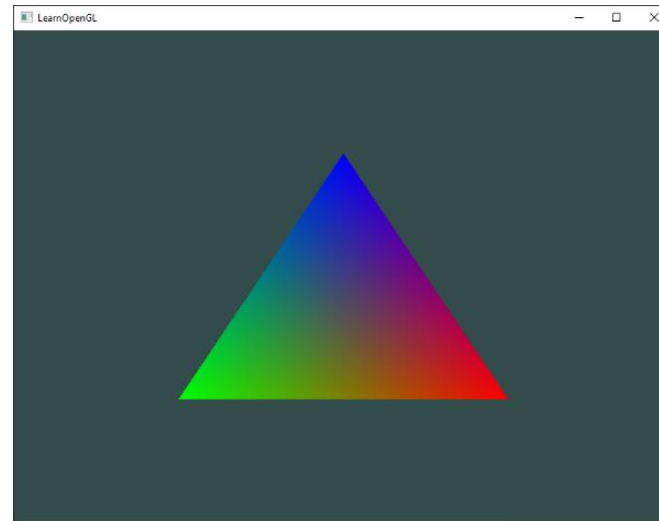
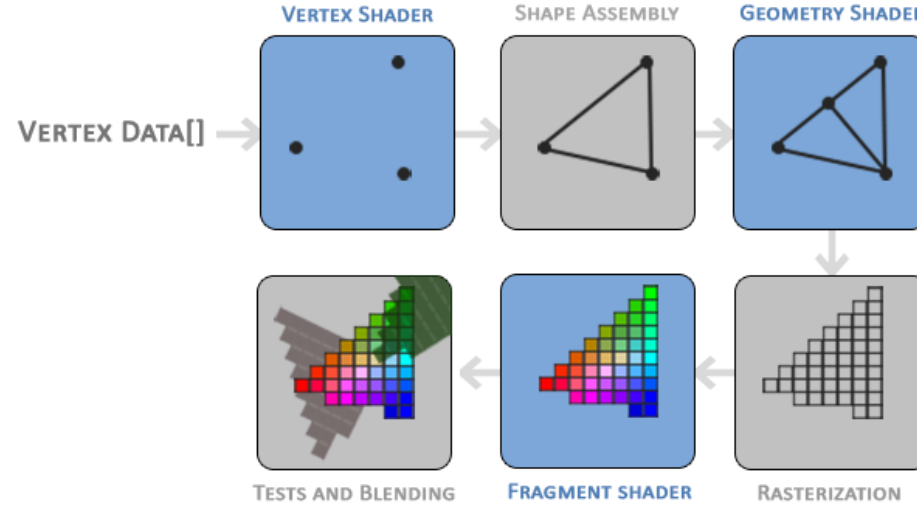
# 1주차

- 오리엔테이션
- Introduction
- OpenGL
- Creating a Window
- Hello Window



# 2주차

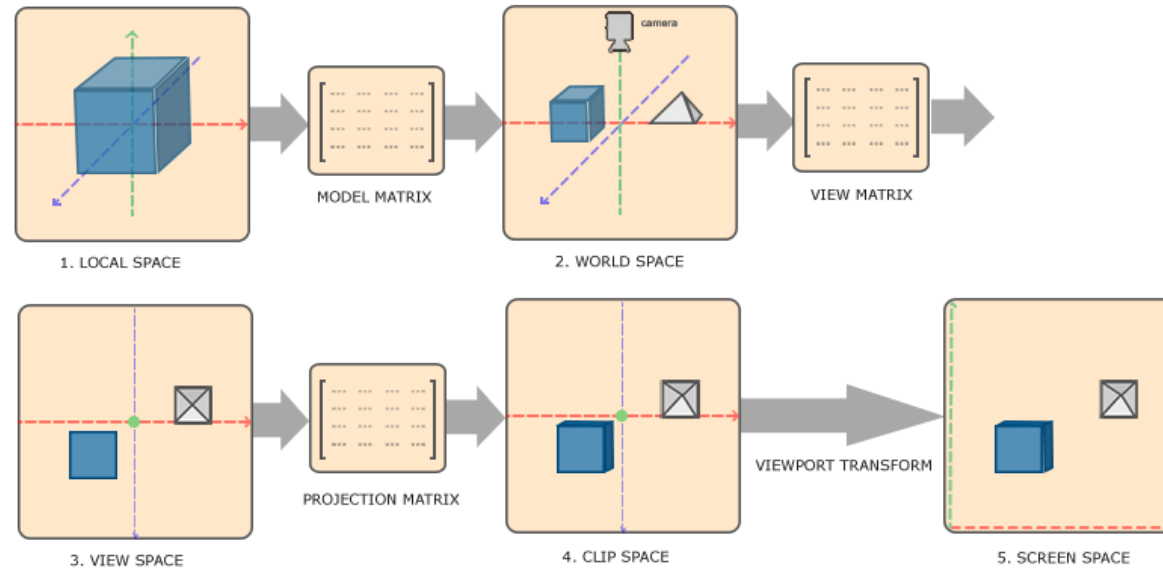
- Hello Triangle
- Shaders
- Textures





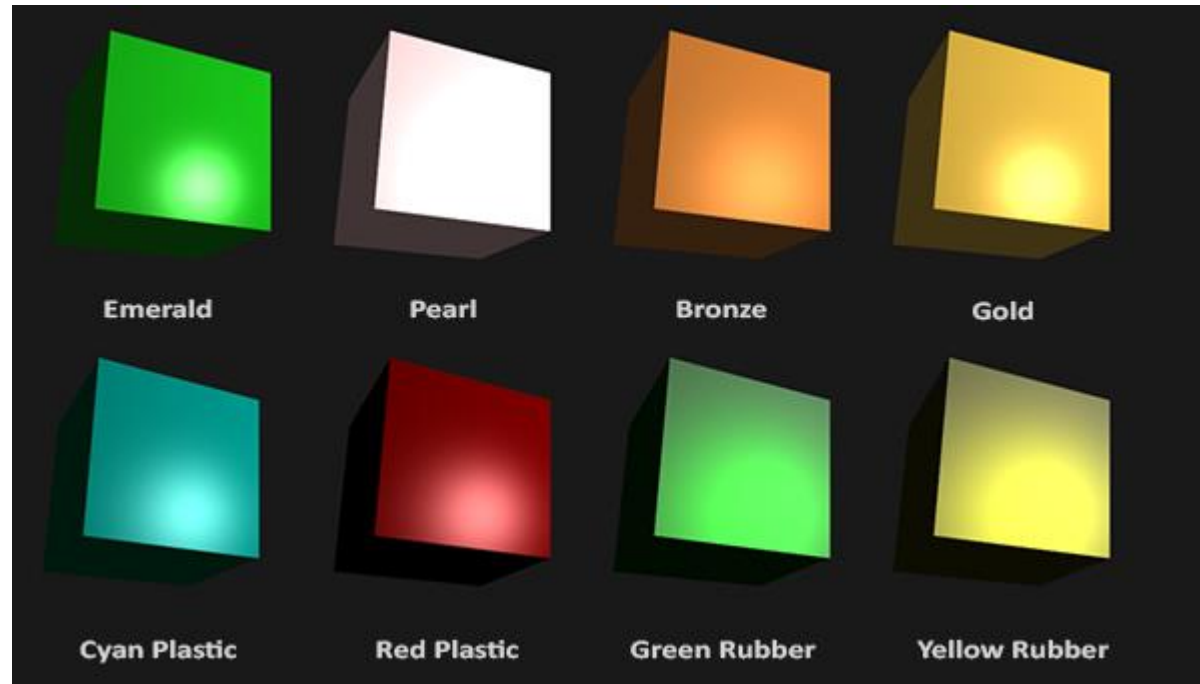
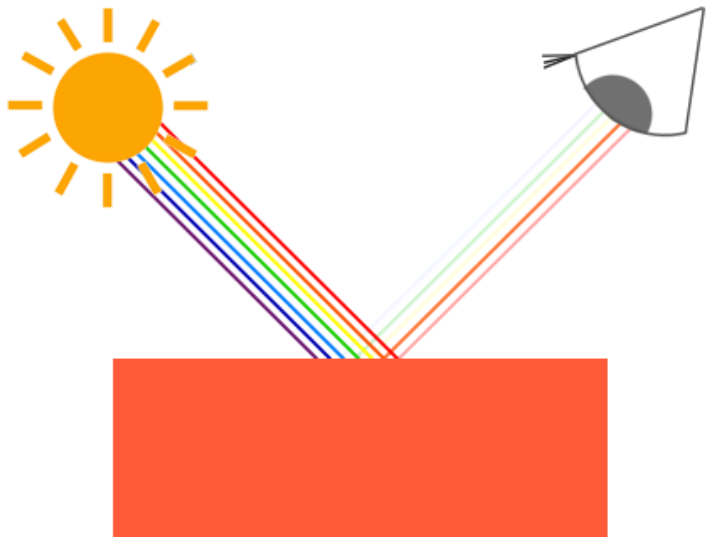
# 3주차

- Transformations
- Coordinate Systems
- Camera



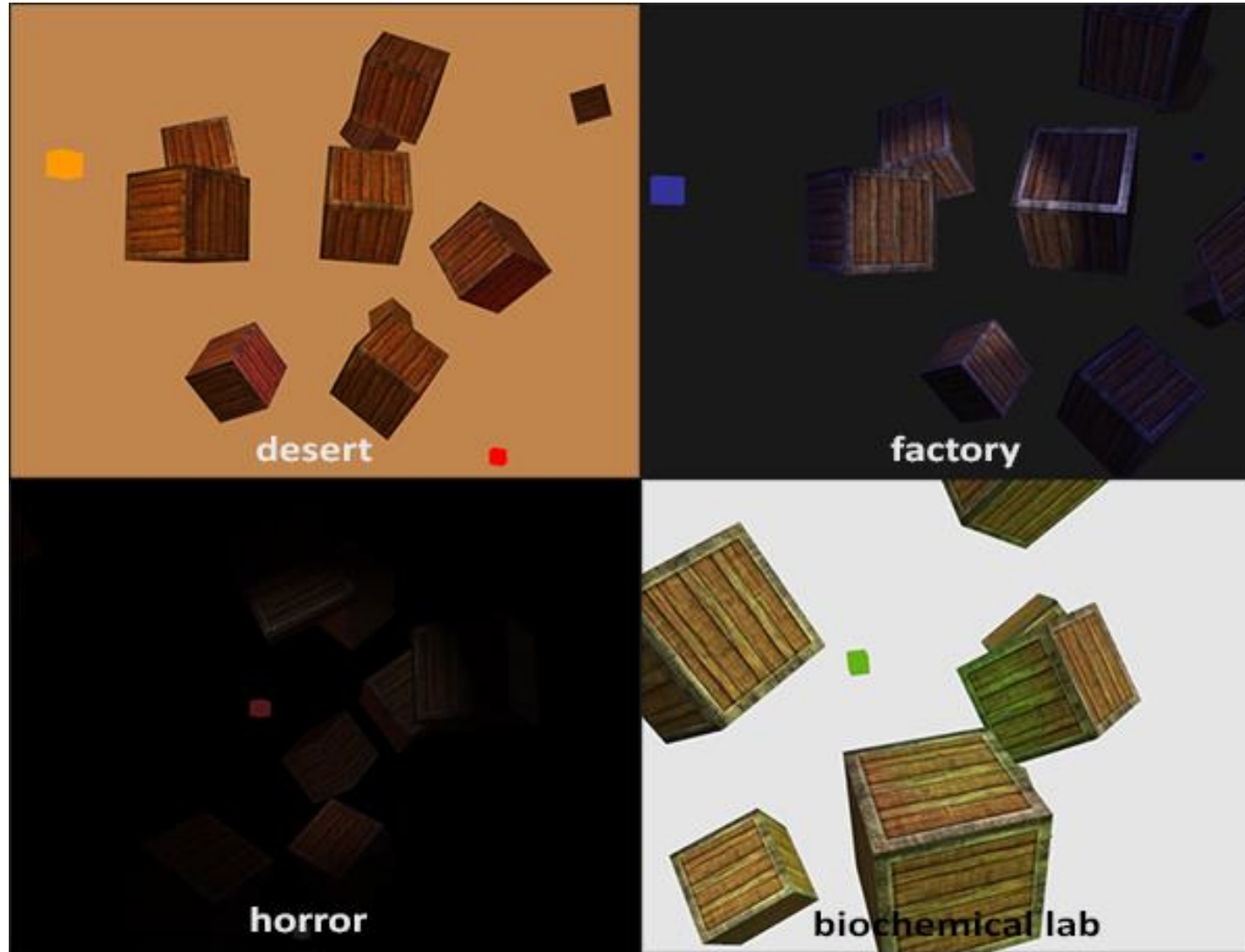
# 4주차

- Colors
- Basic Lighting
- Materials



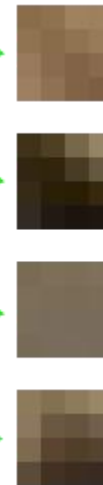
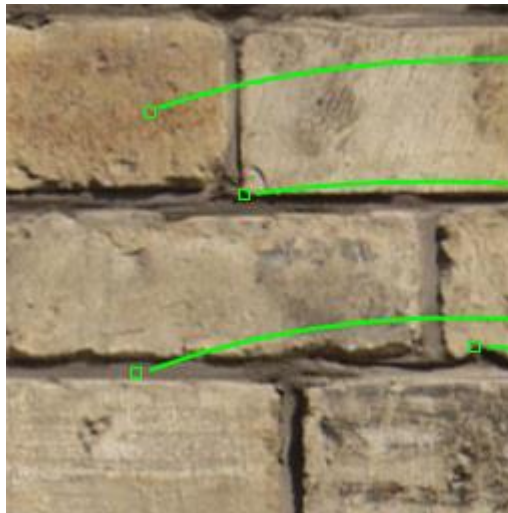
# 5주차

- Lighting Maps
- Light Casters
- Multiple Lights



# 6주차

- Assimp
- Mesh
- Model
- Texture Compression

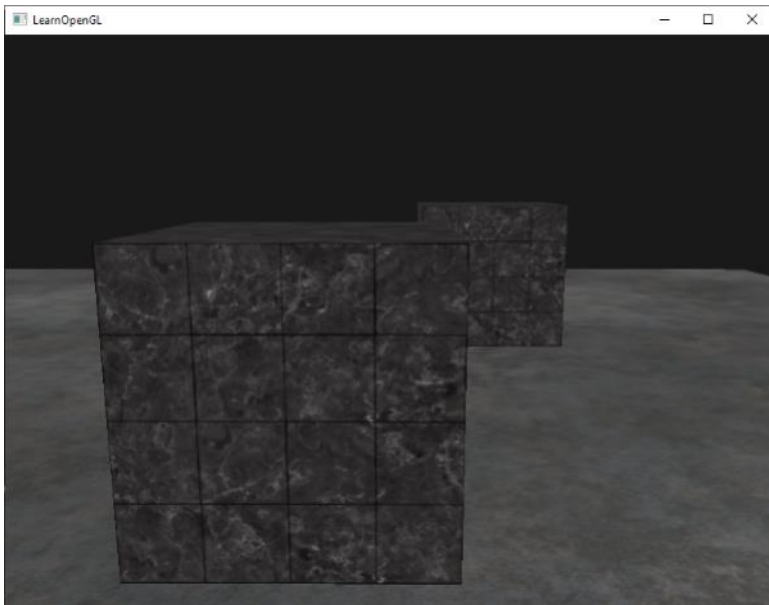


[Understanding BCn  
Texture Compression  
Formats – Nathan Reed's  
coding blog  
\(reedbeta.com\)](#)



# 7주차

- Depth Testing
- Stencil Testing
- Blending
- Face Culling



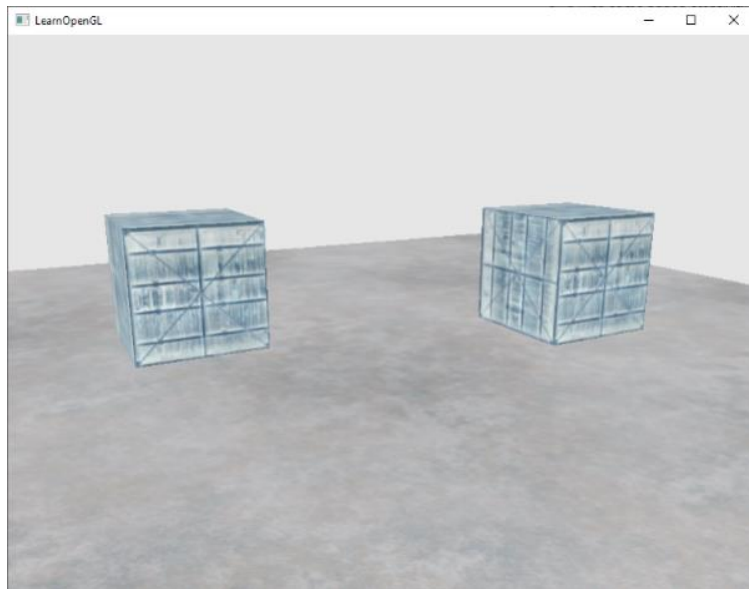
Full transparent window



Partially transparent window

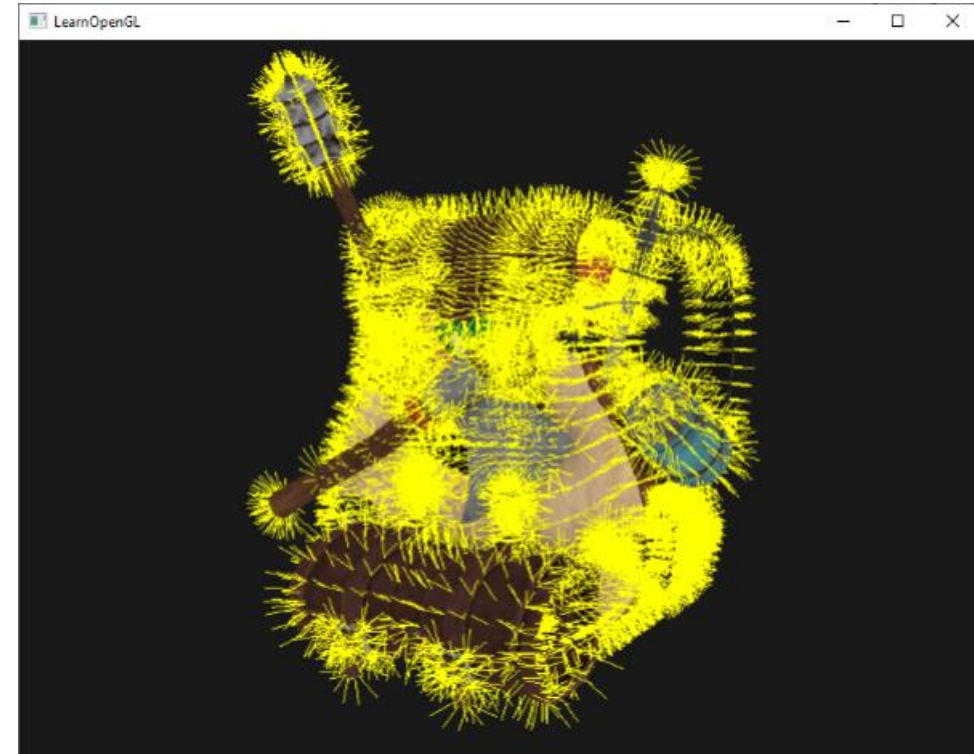
# 8, 9주차

- 중간고사 (8주차)
- Framebuffers (9주차)
- Cubemaps ( " )



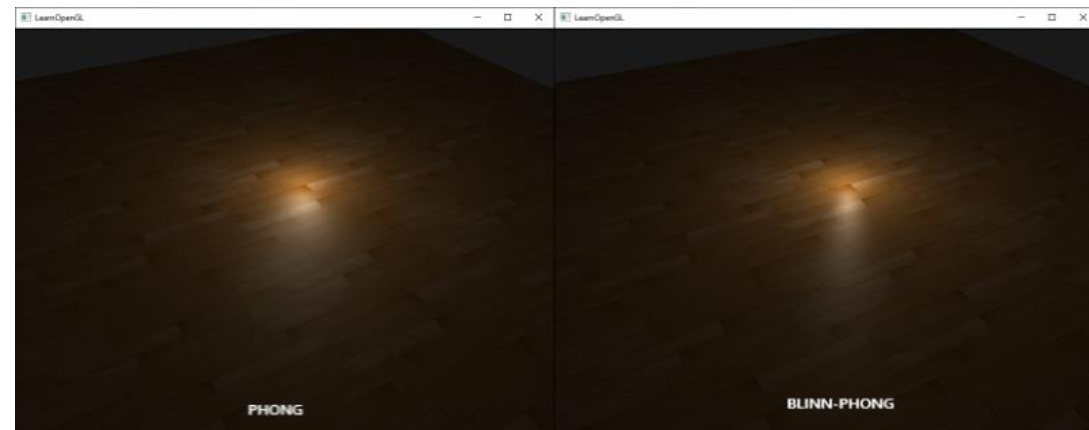
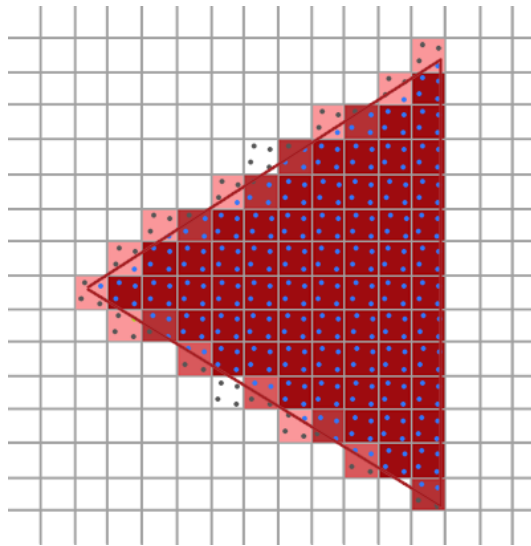
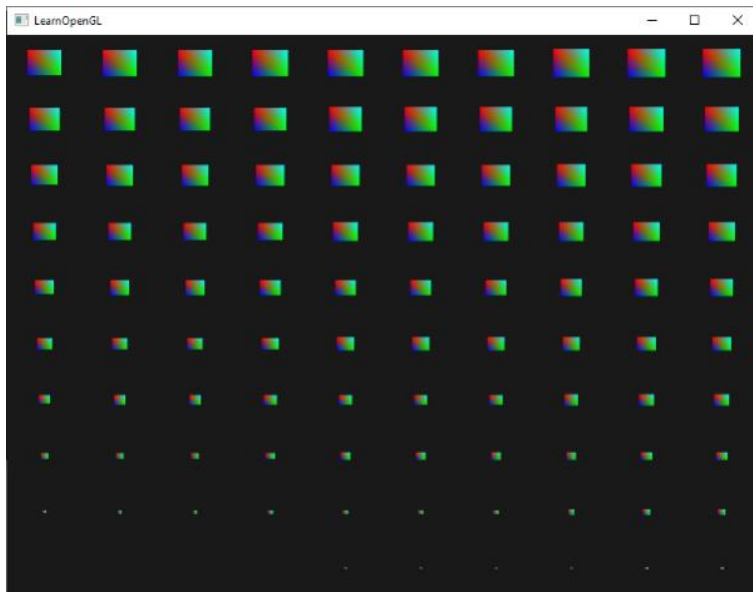
# 10주차

- Advanced Data
- Advanced GLSL
- Geometry Shader



# 11주차

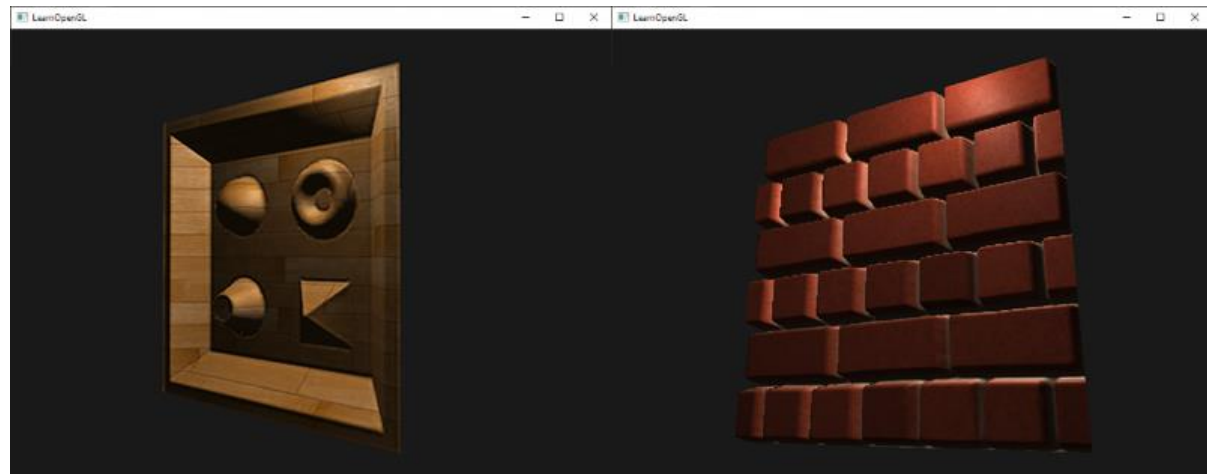
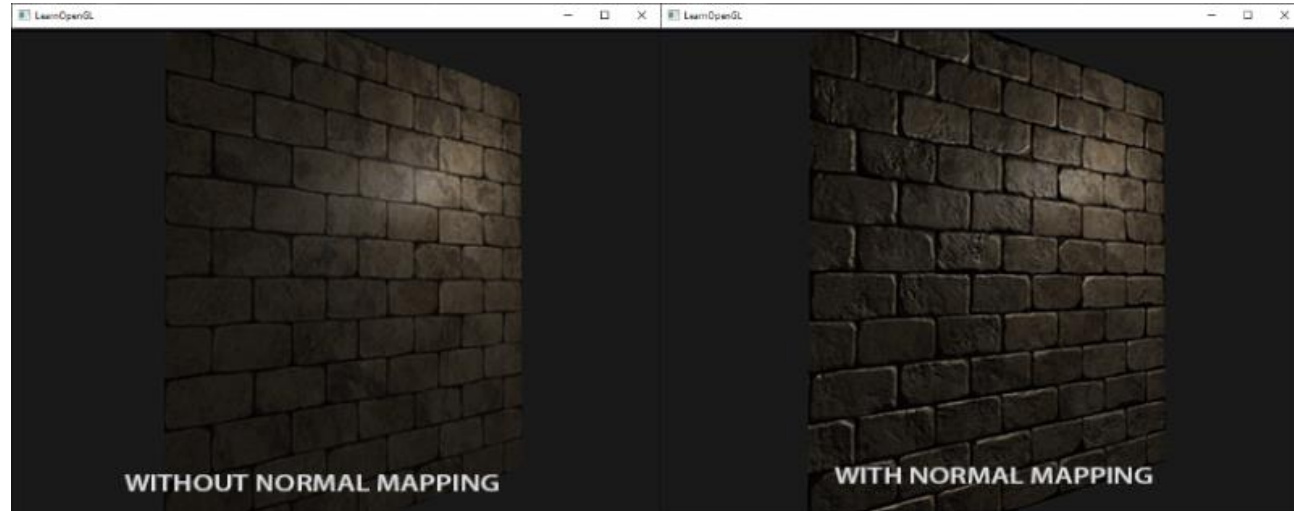
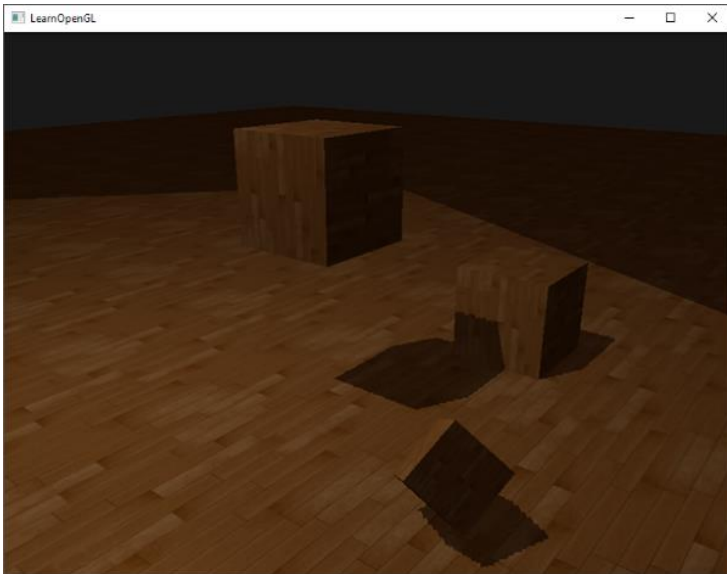
- Instancing
- Anti-Aliasing
- Advanced Lighting
- Gamma Correction





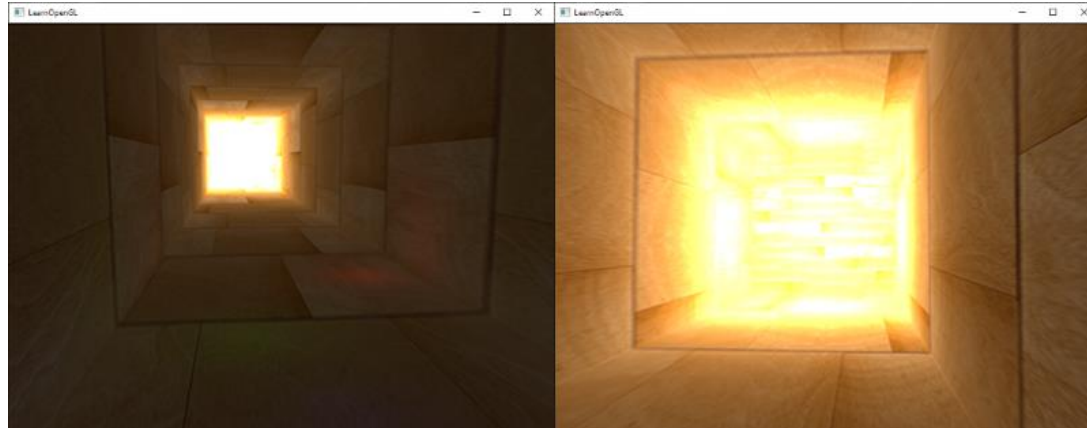
# 12주차

- Shadow Mapping
- Point Shadows
- Normal Mapping
- Parallax Mapping



# 13주차

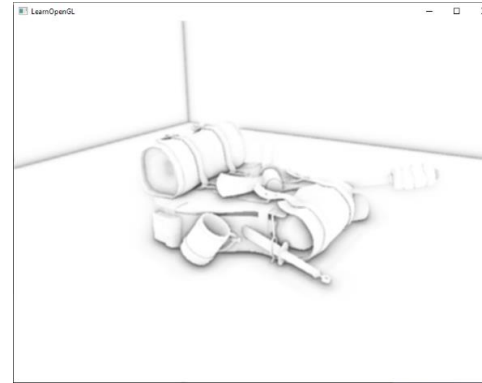
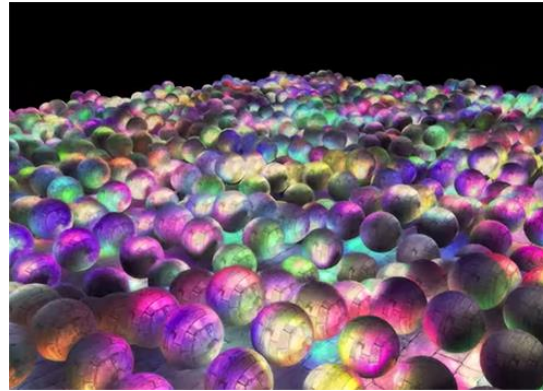
- 프로젝트 발표 (목요일)
- HDR (화요일)
- Bloom ( " )



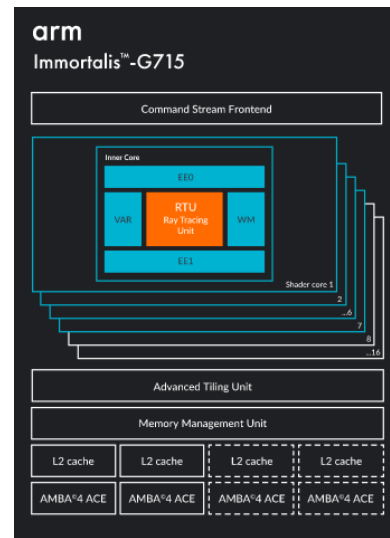
# 14주~16주차

- 14주차에 학회 참석으로 휴강할 경우 15주차에 보강합니다.

- Deferred Shading
- SSAO
- GPU Architecture
- Ray Tracing
- AI-assisted Graphics



- 기말고사 (16주차)



© arm



© Disney, Intel

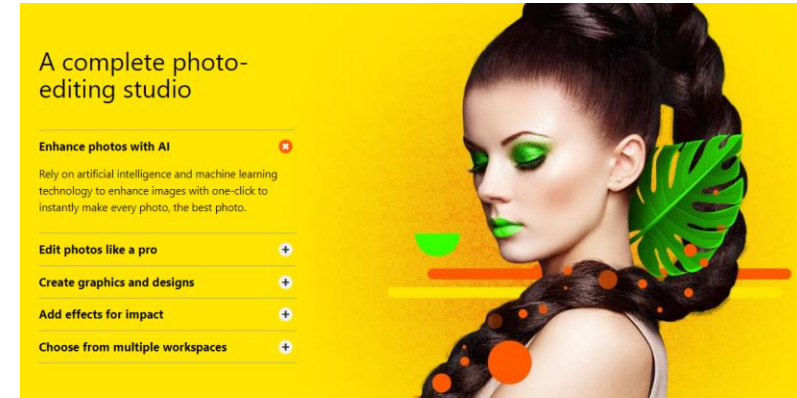


# 컴퓨터 그래픽스란?

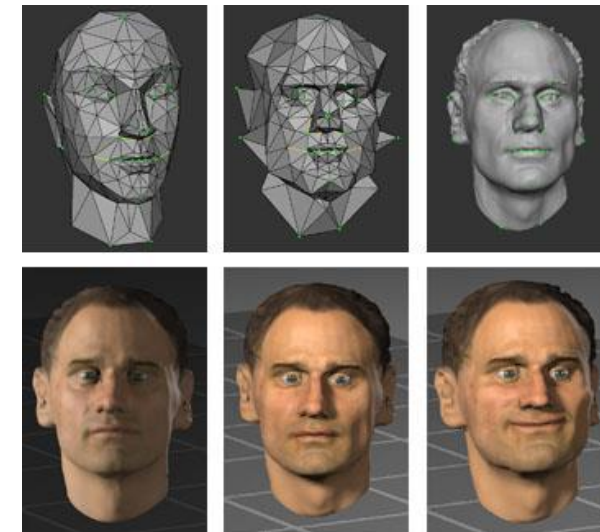


# 컴퓨터 그래픽스의 정의

- 컴퓨터를 사용하여 이미지를 만들고 조작하는 행위
- 2차원 그래픽스
  - 컴퓨터를 이용하여 2차원 평면에 그림을 그리는 것
  - 관련 소프트웨어: Photoshop, Paintshop Pro 등
- 3차원 그래픽스
  - 3차원 가상 공간에 물체를 만들고 (모델링), 움직이도록 하고 (애니메이션), 이를 화면에 표현 (렌더링)
  - 관련 소프트웨어: Blender, Maya, 3ds Max 등
- 본 과목은 3차원 그래픽스 중에서도 렌더링을 중점적으로 다룸



[Powerful, New Photo Editing Software – Try PaintShop Pro 2023](#)



[Modeling and Rendering for Realistic Facial Animation \(cornell.edu\)](#)

# 컴퓨터 그래픽스의 활용 분야 - 게임

- [Forza Horizon 5 | BMW M4 - YouTube](#)





# 컴퓨터 그래픽스의 활용 분야 - 영화

- [Avatar: The Way of Water | Official Teaser Trailer - YouTube](#)



# 컴퓨터 그래픽스의 활용 분야 - VR/AR/메타버스

- [신아영 + 장동민 넘어짐 레전드ㅋㅋㅋㅋㅋㅋ - 예능 짹짹 : 퍼펙트센스 VR - YouTube](#)



# 컴퓨터 그래픽스의 활용 분야 - 시뮬레이션

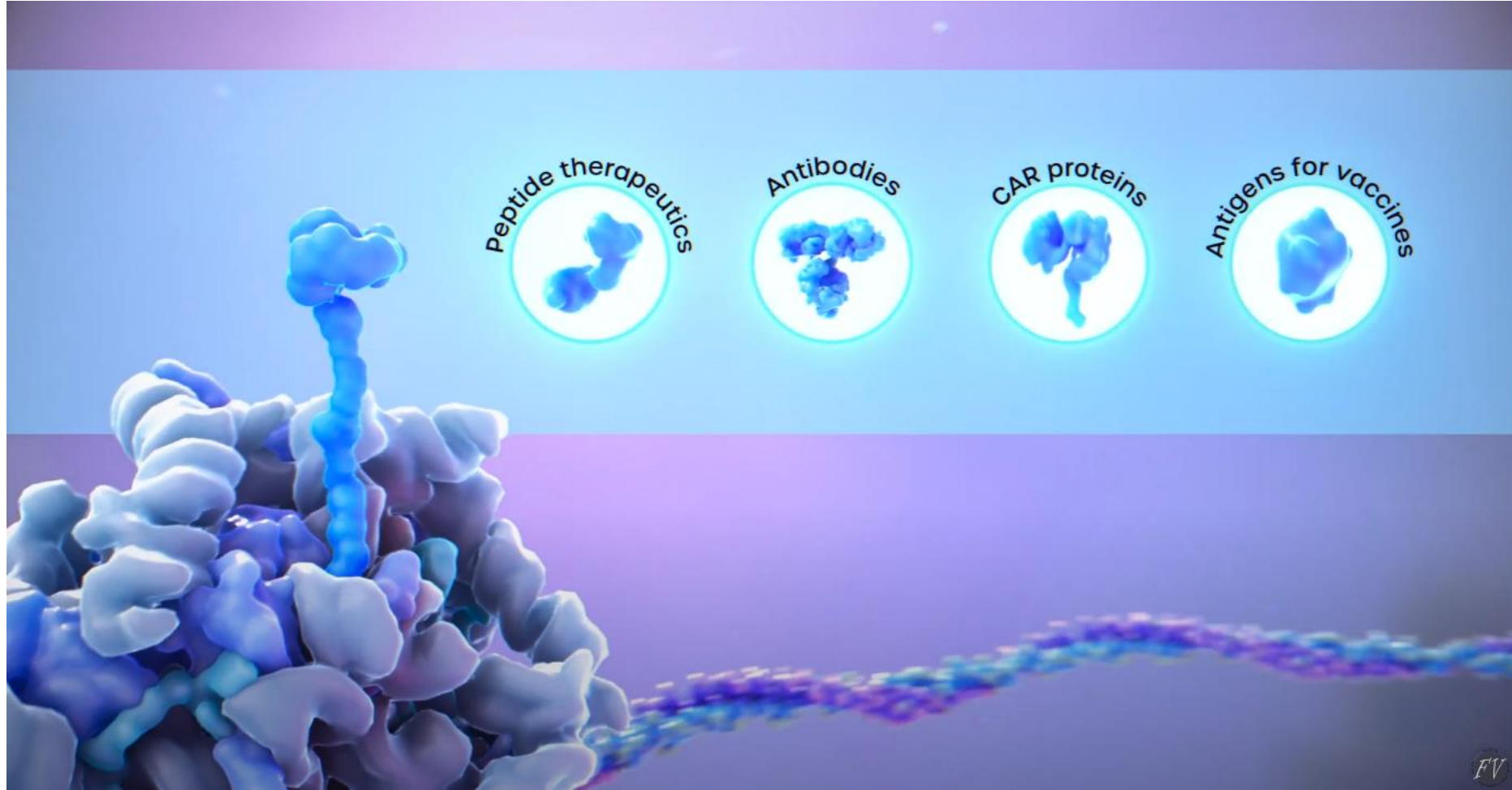
- [How Zoox Uses Simulation to Test and Validate its Self-Driving Technology - YouTube](#)





# 컴퓨터 그래픽스의 활용 분야 - 의료 영상

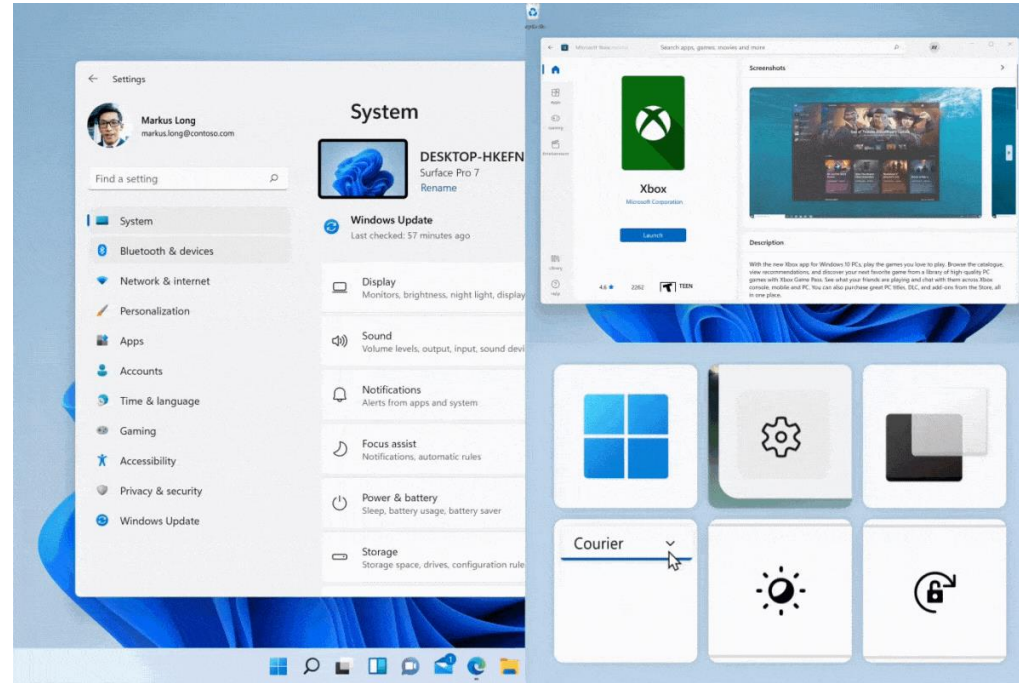
- [Scientific Animation / 메디컬 과학 애니메이션, 의학 생명과학 3D 의료 영상 - YouTube](#)



# 컴퓨터 그래픽스의 활용 분야 - UI/UX



[향후 예측 뒤로 동작 탐색을  
지원하도록 앱 업데이트 \(google.com\)](https://www.google.com)



[Windows 11의 동작 - Windows apps | Microsoft Docs](https://www.microsoft.com/windows/windows-11)

# 컴퓨터 그래픽스의 활용 분야 – Computational Photography

- [Total Relighting: Fast Forward \(SIGGRAPH 2021\) - YouTube](#)





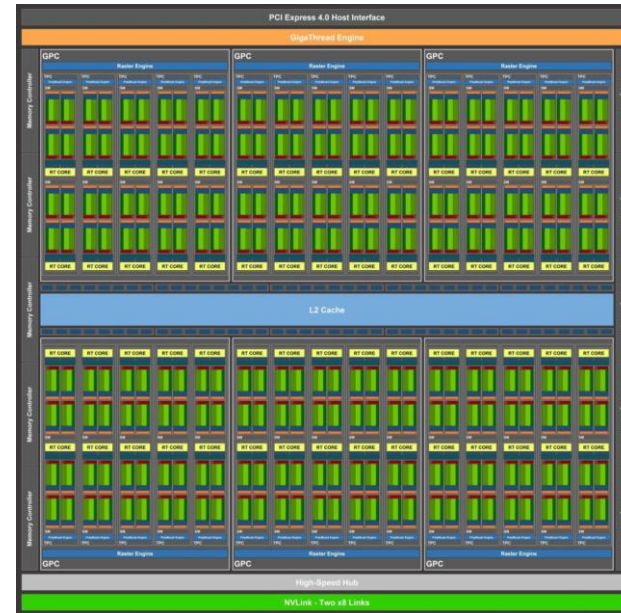
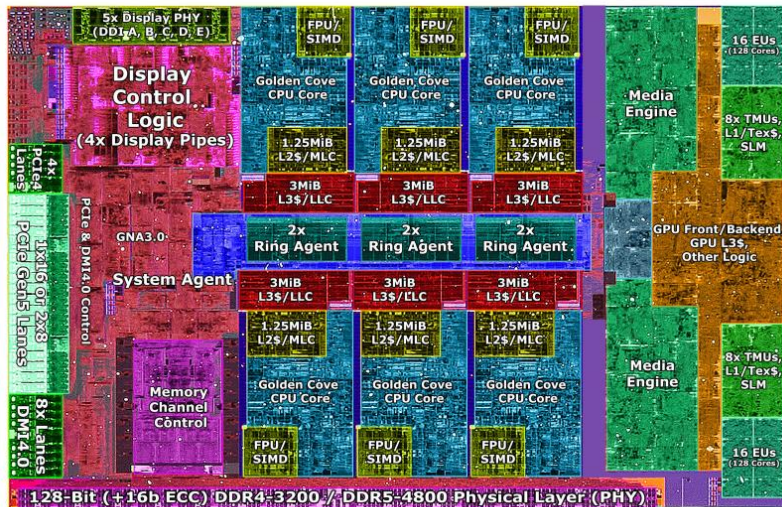
# 컴퓨터 그래픽스의 역사

- [A History of Computer Graphics – YouTube](#)



# GPU (Graphics Processing Unit)

- 그래픽스 관련 연산을 처리하는 장치
- 디스플레이와 연결 가능한 대부분의 컴퓨터 및 장치는 이러한 GPU를 내장
  - 외장 그래픽카드가 장착이 안 된 컴퓨터라도 CPU 안에 GPU가 내장되어 있음
  - 모바일/자동차/가전기기 등에 들어가는 AP(application processor)에도 GPU가 내장
- GPU 성능의 발전과 함께 실시간 렌더링 품질도 비약적으로 향상



Locuza 님의 트위터: "A mockup of the Alder Lake S 6+0 die: <https://t.co/aeMKfC4cdZ>  
<https://t.co/OLaqYTsnNT>" / 트위터 (twitter.com)

NVIDIA 암페어 GPU의 다이 크기, 스펙, 아키텍처, 기타등등 - 컴퓨터 / 하드웨어 - 기글하드웨어 ([gigglehd.com](http://gigglehd.com))

# GPU (Graphics Processing Unit)

- GPGPU (General-Purpose computing on Graphics Processing Units)
  - 초기에는 GPU가 그래픽스 가속만을 위해 쓰여졌으나, 현재는 다양한 용도로 사용
  - AI 가속, 코인 채굴, 과학 시뮬레이션 등
  - 높은 floating-point 연산 성능 덕분

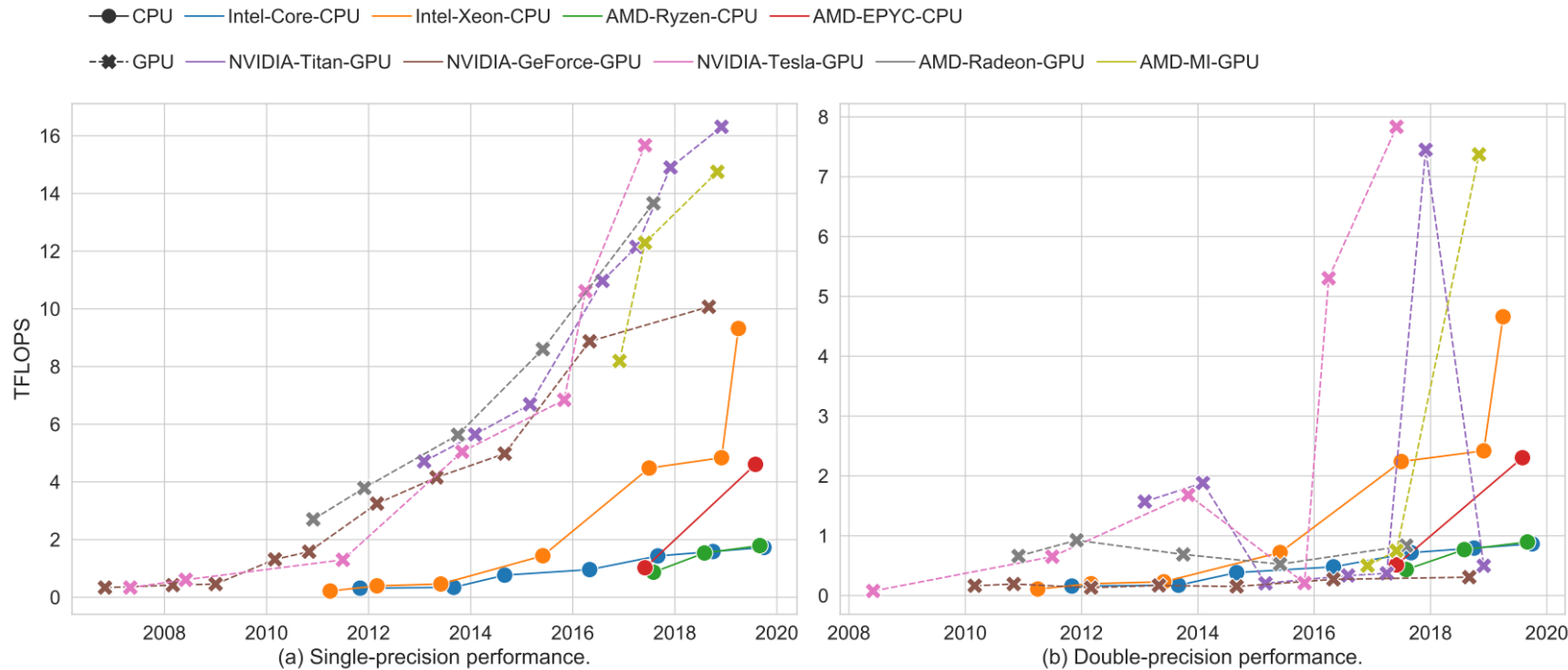


Fig. 6. Comparing single-precision and double-precision performance of CPUs and GPUs.

[1911.11313] Summarizing CPU and GPU Design Trends with Product Data ([arxiv.org](https://arxiv.org/abs/1911.11313))

# 그러면 GPU 프로그래밍은 어떻게 하나요?

- 일반적인 그래픽스 프로그래밍을 위해서는 아래와 같은 API를 사용
  - OpenGL (데스크탑용)
  - OpenGL ES (모바일/임베디드용)
  - WebGL (웹용)
  - Vulkan (범용)
  - DirectX (윈도우 전용)
  - Metal (맥/iOS 전용)
- GPGPU를 위해서는 아래와 같은 API를 사용
  - CUDA (엔비디아 GPU 전용)
  - OpenCL (범용)
  - Metal (맥/iOS 전용)
- 본 과목에서는 OpenGL 사용법 및 이와 관련된 그래픽스 이론을 한 학기동안 배우게 됩니다.





# 마무리

# 마무리

- 이번 시간에는 아래와 같은 내용을 살펴보았습니다.
  - 과목 소개
  - 주차별 강의 계획
  - 컴퓨터 그래픽스의 정의 및 활용 분야
  - GPU의 개념 및 관련 API 소개
- 다음 주 화요일 실습 시간에는 아래 내용을 다룹니다.
  - Introduction, OpenGL, Creating a Window, Hello Window
  - OpenGL 라이브러리 설정을 실습할 예정이므로,  
학기 중 학교 컴퓨터를 이용하지 않을 것이라면 개인 노트북 지참
  - 화요일 강의실은 G206이므로 유의