
Computer Graphics

1 - Course Intro

Yoonsang Lee
Hanyang University

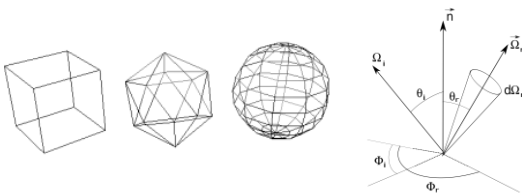
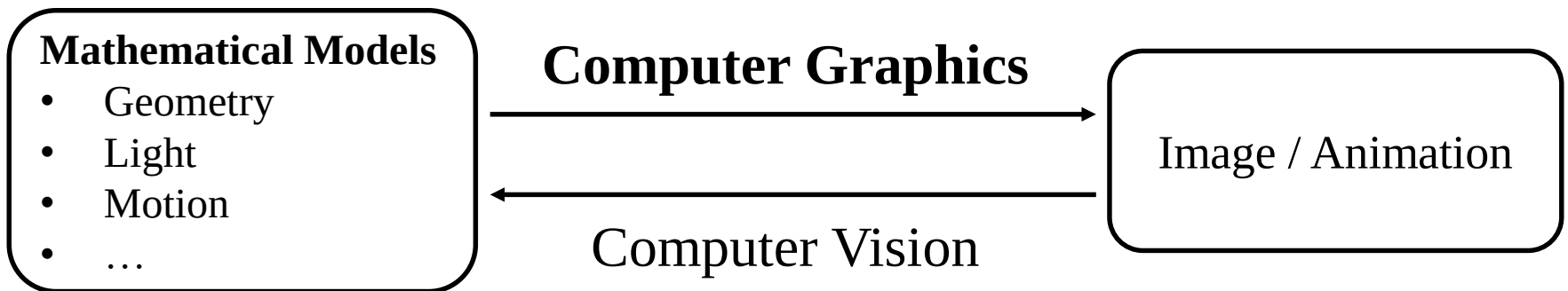
Spring 2026

Course Information

- Instructor: Yoonsang Lee (이윤상)
 - yoonsanglee@hanyang.ac.kr
- Teaching Assistants:
 - Yumin Lee (이유민), Kyungeon Byeon (변경언)
 - You can find their email addresses on <https://cgrhyu.github.io/1-people.html>
- Course Homepage
 - The LMS course homepage at portal.hanyang.ac.kr (or learning.hanyang.ac.kr)
 - Slides will be uploaded to Weekly Learning(주차학습), probably *just before the lecture*. So, please **download lecture slides at the beginning of each lecture**.
 - If you want to study the lecture slides **in advance**, please refer to last year's lecture slides (They won't change much): <https://cgrhyu.github.io/courses/2025-spring-cg.html>

What is Computer Graphics?

- Computer graphics is the study of creating, manipulating, and displaying visual content.
- Core idea:
Computer graphics generates **images** from **mathematical models of a scene**.



Computer Graphics: Practice vs Computer Science

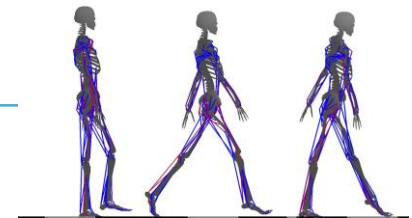
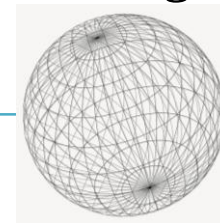
Computer Graphics in **Practice** (Artists / Designers)

- Unreal Engine
- Photoshop
- Blender
- Maya
- ...



Computer Graphics in **Computer Science**

- Geometry models
- Rendering algorithms
- Shading models
- Animation algorithms
- ...



These tools implement those algorithms.

➔ This course studies the algorithms and mathematical models behind these tools.

Where Do We See Computer Graphics?

- Applications of Computer Graphics

- Games
- Movies & Animation
- Virtual / Augmented Reality
- Scientific Visualization
- Medical Imaging
- Robotics & Simulation
- Design & Architecture



Where Do We See Computer Graphics?

- Applications of Computer Graphics

- Games
- **Movies & Animation**
- Virtual / Augmented Reality
- Scientific Visualization
- Medical Imaging
- Robotics & Simulation
- Design & Architecture



Where Do We See Computer Graphics?

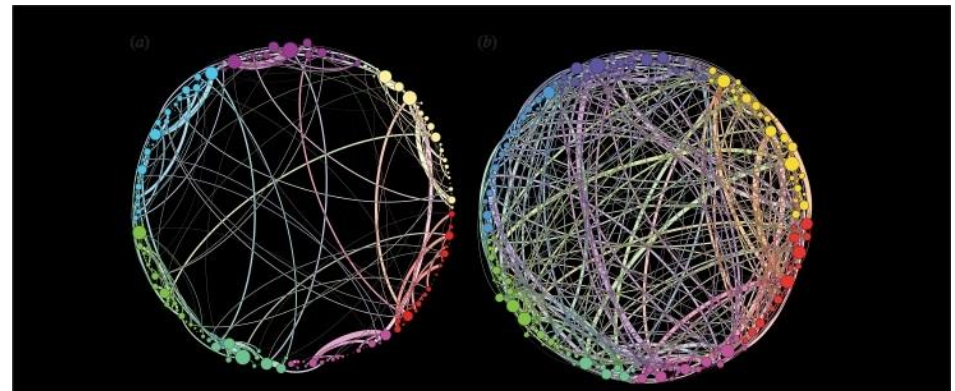
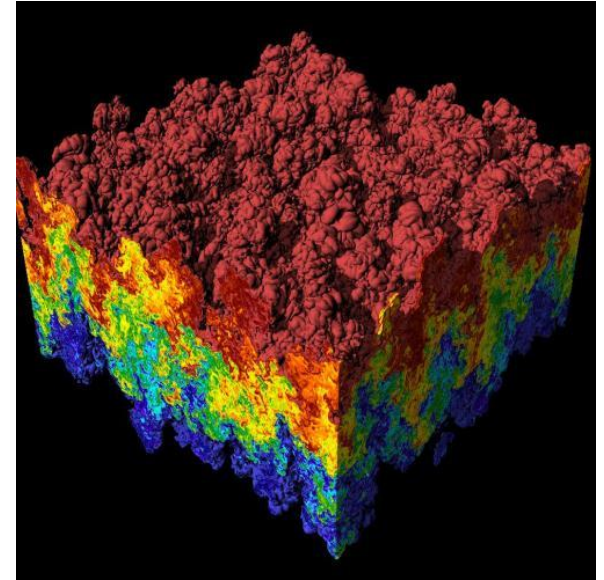
- Applications of Computer Graphics
 - Games
 - Movies & Animation
 - **Virtual / Augmented Reality**
 - Scientific Visualization
 - Medical Imaging
 - Robotics & Simulation
 - Design & Architecture



Where Do We See Computer Graphics?

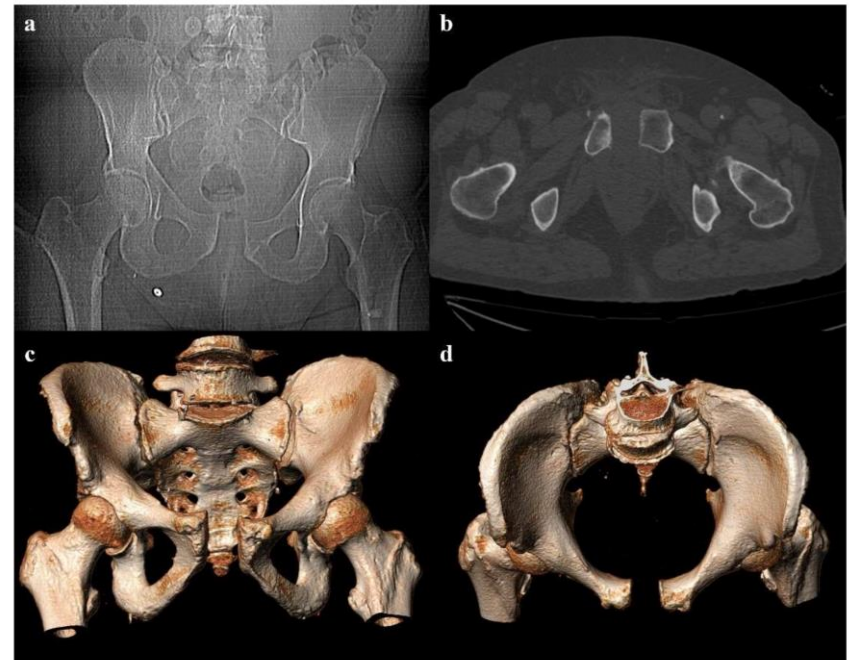
- Applications of Computer Graphics

- Games
- Movies & Animation
- Virtual / Augmented Reality
- **Scientific Visualization**
- Medical Imaging
- Robotics & Simulation
- Design & Architecture



Where Do We See Computer Graphics?

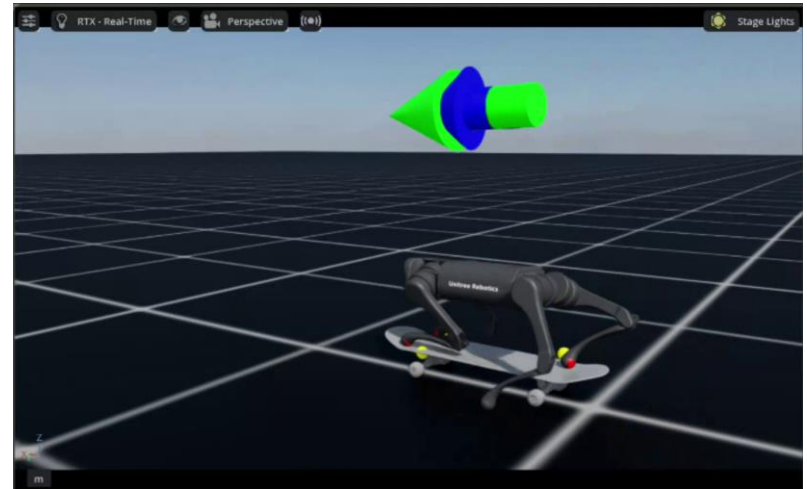
- Applications of Computer Graphics
 - Games
 - Movies & Animation
 - Virtual / Augmented Reality
 - Scientific Visualization
 - **Medical Imaging**
 - Robotics & Simulation
 - Design & Architecture



Where Do We See Computer Graphics?

- Applications of Computer Graphics

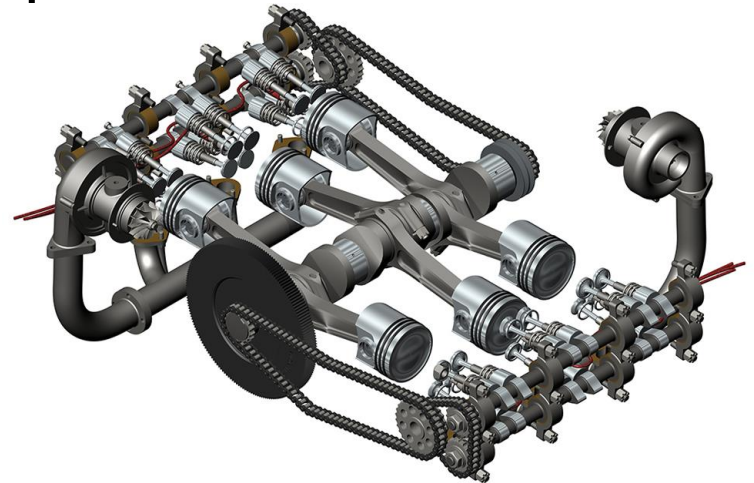
- Games
- Movies & Animation
- Virtual / Augmented Reality
- Scientific Visualization
- Medical Imaging
- **Robotics & Simulation**
- Design & Architecture



Where Do We See Computer Graphics?

- Applications of Computer Graphics

- Games
- Movies & Animation
- Virtual / Augmented Reality
- Scientific Visualization
- Medical Imaging
- Robotics & Simulation
- **Design & Architecture**



Course Overview

- Computer Graphics: Mathematics made visible
- In this course, you will
 - Explore **fundamental computer graphics & mathematics concepts**
 - Learn how to use **modern OpenGL**
- You will **not**
 - Learn sophisticated computer graphics techniques
 - Learn or use **legacy OpenGL**

Course Overview

- As “implementation examples” of the fundamental computer graphics concepts, we’ll use:
- On-line WebGL demos (Lectures)
- Modern OpenGL API (Labs)
- Blender: An open-source 3D graphics tool (sometimes)
- These examples will help you understand the concepts more clearly.

Prerequisites

- Basic knowledge of linear algebra
 - Understanding of basic concepts and operations on vectors and matrices,
 - Such as matrix-matrix multiplication, matrix-vector multiplication, identity matrix, dot product, cross product, etc.
 - If you know these things, it is enough to take this course.
- Basic knowledge of programming skill
 - We'll use Python in this class.
- You don't need artistic sense!

Textbook

- **Textbook : Lecture slides**

Schedule (subject to change)

Week	Topic	Mon
1	1 - Course Intro / 1 - Lab - Environment Setting	3/9
2	2 - Rendering Basics / 2 - Lab - Hello Triangle!	3/16
3	3 - Transformations / 3 - Lab - Sending Data to Shaders	3/23
4	4 - Affine Space / Frame / Matrix / 4 – Lab	3/30
5	5 - Vertex Processing 1 / 5 – Lab	4/6
6	6 - Vertex Processing 2 / 6 – Lab	4/13
7	7 - Hierarchical Modeling, Mesh / 7 – Lab	4/20
8	8 - Lighting / 8 – Lab	4/27
9	Midterm Exam	5/4
10	9 - Orientation & Rotation / 9 – Lab	5/11
11	10 - Character Animation / 10 – Lab	5/18
12	No Class (public holiday)	5/25
13	11 - Curves / 11 – Lab	6/1
14	12 - More Lighting, Texture / 12 – Lab	6/8
15	13 - Scan Conversion, Visibility / 13 - Lab	6/15
16	Final Exam	6/22

Lectures & Labs

- Each pair of a lecture and a lab will be held on the same day.
- Lectures (by instructor)
 - Lectures for fundamental computer graphics concepts
 - Does not depend on specific libraries or software
- Labs
 - Part 1: OpenGL concepts and demos (by instructor)
 - Part 2: In-class practice (Lab Check)
- One [Lecture+Lab] session is a **continuous 4-hour class**.
- **No break in between!**

Lab Check

- One **Lab Check** per lab session.
- Submit your work through LMS during the lab session.
- An **in-class practice activity**.
 - The goal is to **expose you to and let you explore** key concepts from lectures and demos.
 - This is a short in-class practice activity, **not a graded assignment**.
- You may ask TAs questions
 - about **technical issues**
 - or **high-level conceptual clarification**
 - **But not about whether your submission is correct or complete.**

What is a Lab Check?

- A Lab Check may involve:
 - **a small task** related to the lecture (e.g., implementing or experimenting with simple code)
 - exploring behaviors or results
 - thinking about underlying concepts
- It is **not evaluated as a problem-solving assignment,**
and **correctness is not graded at this stage.**

Lab Check - How to submit

- Submit through **LMS during the lab session.**
- You may **leave the classroom after submitting to LMS.**
- TAs record Lab Check statuses on LMS.
 - This is **not a correctness approval.**
- Each Lab Check is recorded as one of the following:
 - **Pass**
 - **Fail**
 - **Not Submitted**
 - Fail indicates that the submission does not show a meaningful attempt to engage with the Lab Check task (e.g., empty, irrelevant, or clearly incomplete submissions).

Lab Check – Grading

- Only submissions **during the lab session** are recorded.
 - The lab session ends **exactly at the LMS submission deadline**.
 - No submissions are accepted after the deadline.
- Important notes:
 - **Passing does NOT mean your understanding is correct.**
 - Lab Checks are for practice and exposure.
 - **Concepts from Lab Checks and Projects will be evaluated in exams.**
- In the next lab session, we may briefly review the previous Lab Check.

Projects

- **Three individual projects** will be given during the semester.
(No team projects.)
- Projects are **challenging programming tasks** where **multiple valid solutions are possible**.
- Each project may require you to:
 - make **design choices** among multiple valid options
 - implement your chosen approach
 - **explain and justify** your decisions and trade-offs
- The due date will be **2–3 weeks** after each project is given.

Policy for Projects

- **Late submissions are not accepted.**
 - Submit before the deadline.
- **Academic integrity**
 - Projects must represent **your own work and decisions**.
 - You may use external resources (including AI tools) **as support**, but you are responsible for:
 - **understanding the code you submit**
 - **explaining your design choices and trade-offs** in your own words
- **NO SCORE will be given if:**
 - the submission is largely copied from external sources **with little or no meaningful modification**
 - you **cannot clearly explain** how your implementation works
 - multiple submissions are substantially identical **without independent work**

Use of AI Tools in This Course

- You can use AI tools (e.g., ChatGPT, code assistants) in this course to help you write, debug, and explore code.
- However, AI tools **do not replace your responsibility** to:
 - understand the code you submit
 - make your own design decisions
 - explain and justify your choices in your own words
- In this course:
 - **Lab Checks:** AI use is allowed for exploration and practice
 - **Projects:** AI use is allowed, but **design choices and explanations are evaluated**
 - **Exams:** AI use is **not allowed**
- Submitting AI-generated content **without understanding or justification** may result in **no score**, as stated in the project policy.
- **You should be able to explain any code you submit.**

Grading

Midterm exam	40%
Final exam	40%
Projects	10%
Lab Checks	5%
Attendance	5%

- You will get "F" for more than 5 absences in four-hour sessions of (lecture + lab)s.
- Absences from the midterm or final exam → F

Grading Policy

- Basic principle: Separating the grades where there is a big gap between points.
- Guideline:

A	20%~25%
B	25%~35%
C	40%

Language

- Lectures will be conducted in English.
- **Please ask questions in English** so that all students can follow the discussion.
 - **Answers may be given in both English and Korean.**
- Unlike last year, I will not repeat the **full explanation** in Korean after each slide.
 - If something is unclear, please ask questions.
 - For **important or complex concepts**, I may provide **additional explanation in Korean** when helpful.
- In lab part 2, TAs will mainly use English for announcements to all students.
 - You may ask TAs individually in **English or Korean.**

Questions – Slido.com

- After lectures, please post your questions on the LMS Q&A Board (게시판 - Q&A 게시판).
 - TAs will check and respond at least once a day.
- In lectures, we'll use an online, anonymous Q&A platform (slido.com) to encourage questions.

Just Try Asking a Question!

- Go to <https://www.slido.com/>
- Join #cg-ys
- You do not need to log in to slido.com.
- **Do not bookmark a slido event page** because new events will be created every week!
- Ask any questions in English!

Questions – Slido.com

- On slido.com, you can
 - **Ask** your own questions anonymously
 - **Upvote** other questions
- We'll use the slido Q&A **only during lecture & lab sessions led by the instructor.**
 - Not after lecture or lab session
 - Not during TA lab time (lab part 2)
 - No written responses
- Please ask questions **anonymously.**
 - Just leave your name blank when posting a question.

Quiz & Attendance – Slido.com

- Three quiz questions per lecture (using slido.com polls)
- Very simple questions – you must submit your answer in two minutes.
- **Attendance will be recorded based on quiz submissions.**

Quiz – Slido.com

- You **MUST** submit your answer in the following format:
 - **Student ID: Your answer**
 - **e.g. 2024123456: 4**
 - Whether your answer is correct or not **has nothing to do with your attendance.**

Attendance	Number of submissions in the format - 3 times && You are in the classroom
Late	Number of submissions in the format – 1~2 times && You are in the classroom
Absence	Number of submissions in the format – 0 times You are NOT in the classroom

- **3 lates count as 1 absence.**

Quiz & Attendance – Slido.com

- If submitting a quiz answer without attending the class is detected,
- I think they have been also absent from the previous lecture.
- → Counted as “Absence” for these two lectures

Just Try a Quiz!

- Go to <https://www.slido.com/>
- Join #cg-ys
- Click “Polls”
- Submit your answer in the following format:
 - **Student ID: Your answer**
 - e.g. 2024123456: 4)
- Note that your quiz answer must be submitted **in the above format** to be counted as attendance.

About Laptop

- Bring your laptop to lectures and labs.
 - Lecture slides contains many WebGL demos, so you'll need a laptop to run them and listen to the lecture.
 - In labs, you will need your laptop for practice and lab checks.
- Ensure your laptop supports **OpenGL 3.3 or higher**.
 - We'll use **OpenGL 3.3 Core Profile**.

Checking OpenGL Version

- Windows, Mac: Use "OpenGL Extensions Viewer"
 - <https://support.esri.com/en/technical-article/000011375>

- Ubuntu: Use "glxinfo" (need to install mesa-utils)

```
sudo apt-get install mesa-utils  
glxinfo | grep version
```

- Find 'OpenGL core profile' version (we'll use the core profile).
- Reference: Supported APIs for Intel® Graphics
 - <https://www.intel.com/content/www/us/en/support/articles/000005524/graphics.html>
 - Intel CPUs with integrated graphics that do not support OpenGL 3.3 are quite old CPUs released in 2011.

Classroom Etiquette

- **DO NOT negatively affect other students** in the classroom. For example,
 - Doing other things with your laptop (e.g. playing games)
 - Looking at your phone for a long time
 - Private conversation
 - Sleeping on a desk

My Recommendation

- This class is NOT recommended to those who...
 - want an easy A.
 - expect lectures in perfect English.
- This class is recommended to those who...
 - wonder how movies and games render scenes.
 - are interested in how objects or characters move.
 - like to visualize things using code.
 - have a passion for computer graphics!

Lab Session

- We don't have a lab session today.
- However, you must set up the environment using “1- Lab - Environment Setting.pdf” to follow the next lecture.
- Submit Assignment 1 (“1 – Assignment.pdf”) through LMS before the next lecture.

Lastly...

- If you agree on all these policies, see you next time!
- If not, please consider taking another class instead.