

Computer Graphics and Programming

Lecture 1

Jeong-Yean Yang

2020/10/22

The Goal of Graphics in this class is

Developing your own
C/C++ Graphics Engine

by Mathematical Modeling



3DMax, Maya,
or
Even Sketch?

These are NOT graphics, but CAD.
You can learn it from Youtube.

How to develop CAD program
is our goal.

How about using
Unity-like
Graphics engine?

Unity is NOT Graphics.

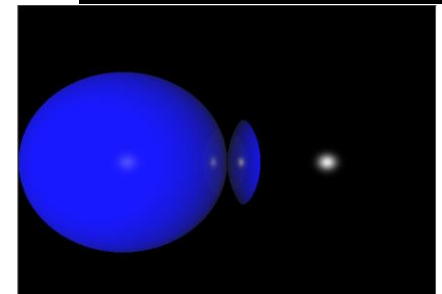
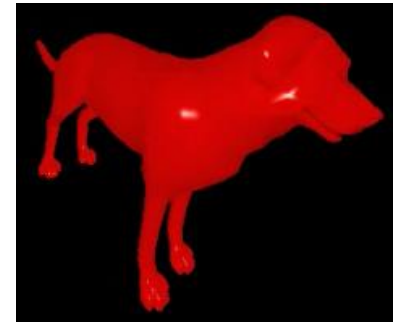
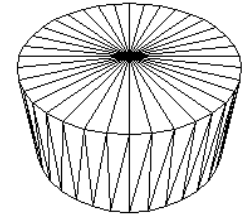
What you want
is NOT in Unity,
but in Graphics Theory.

I am foolish
in C/C++
programming?

Try harder
or
Drop the course.

Introduction

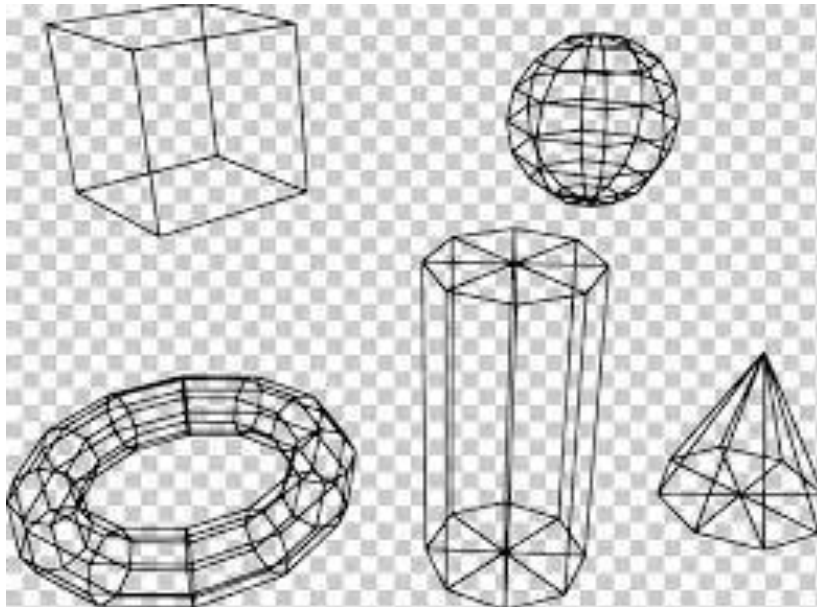
- 3Dim. Vector space and Homogeneous Transform
 - Viewing frustum based clipping and Perspective Mapping
 - Wire frame based Polyhedron Modeling with Vertices and Faces
 - Quaternion and Homogeneous Transform
 - Baricentric Interpolation and Hidden surface Removal
 - Modeling of Various Types of Polyhedron as Primitives
-
- OpenGL 3D Modeling
 - Vertex Buffer Object and Elements Handling
 - Illumination Model with Normal Vector for Shadings
 - Multi object modeling with Hierarchical Structure
-
- Ray Tracing with Reflection and Refraction
 - Understanding GLSL Concept by Ray Tracing
 - GLSL in OpenGL
 - Particle Effect



Keyword Overview

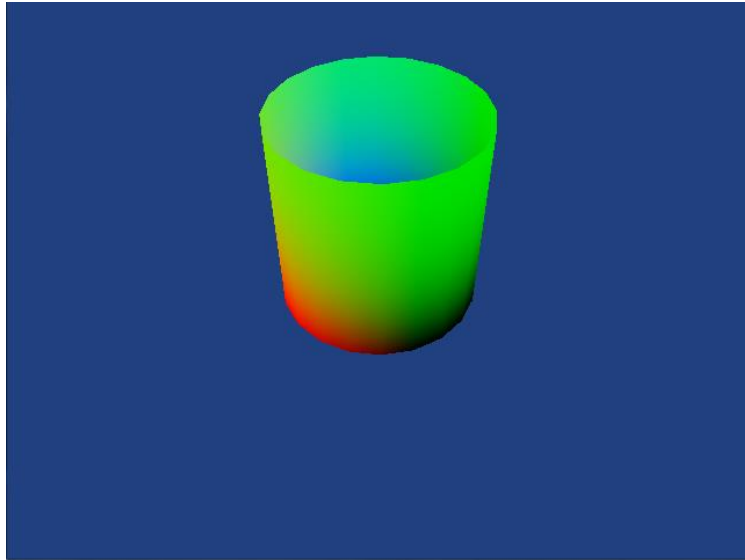
- Step1: Line-based Graphics with Windows Programming
- Step2: Polygon-based Graphics with OpenGL
- Step3: Ray Tracing and GLSL

Step1. Line-based Graphics

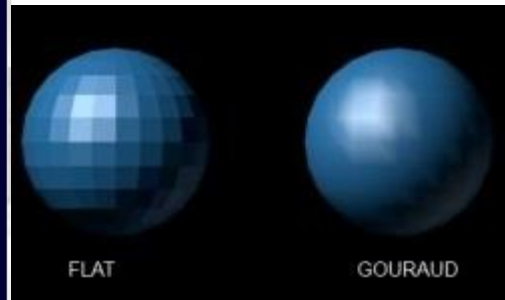
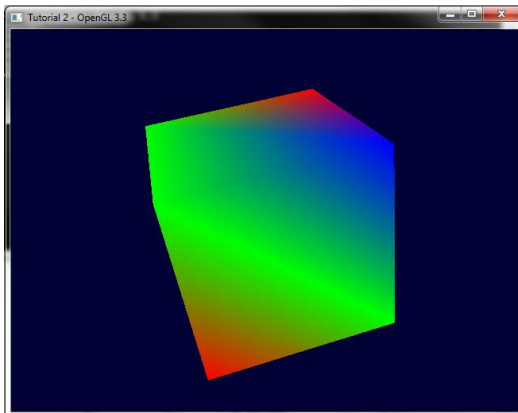


- Everything is rendered by Line Command
 - MoveTo, LineTo in MFC
- In other words, Everything is calculated by Math.
 - 3D Vector space
 - Homogeneous Transform
 - Perspective Transform
 - Projection
 - Clipping

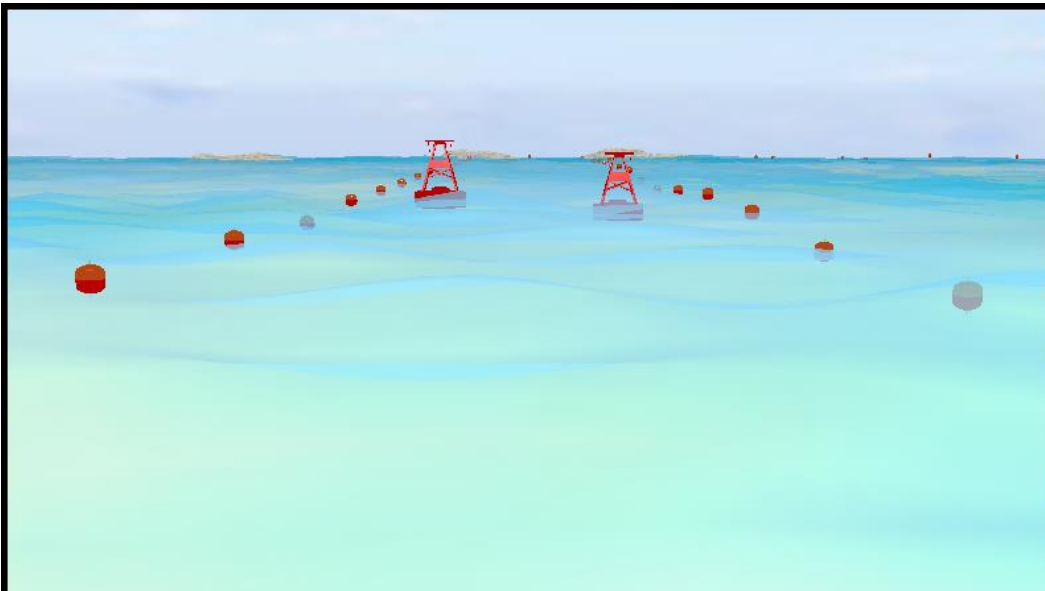
Step2. Polygon-based Graphics with OpenGL



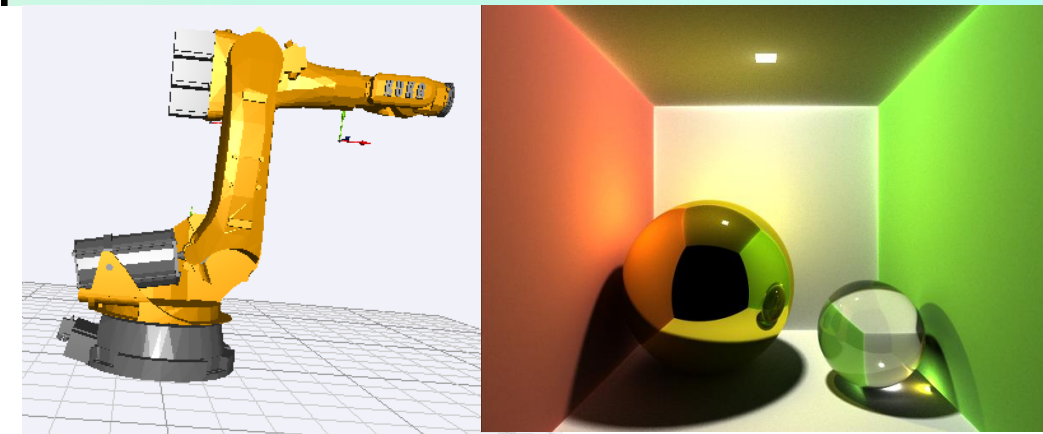
- VBO from OpenGL
- Light and material
- Shading
- Polygon and Polyhedra for OpenGL



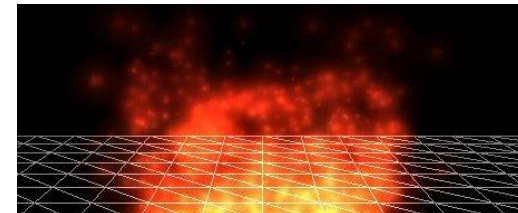
Step3. Ray Tracing and GLSL



- Light and Material modeling
- Ray Tracing by math
- GLSL – Advanced shading language

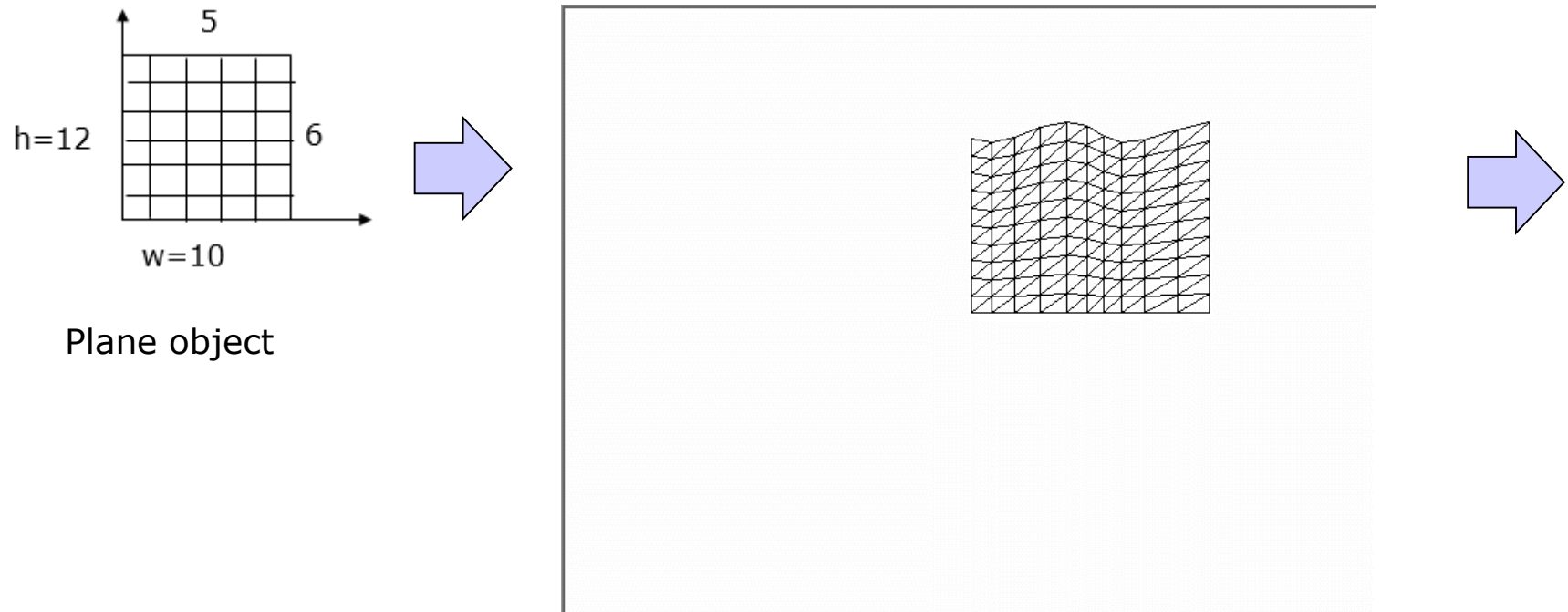


Step3. Minor Part Particle Effect

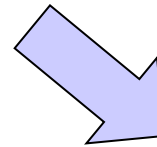


Graphics is an Intuitive Coursework

- Your seeing ability will be improved
 - by learning the coursework and understanding the graphics



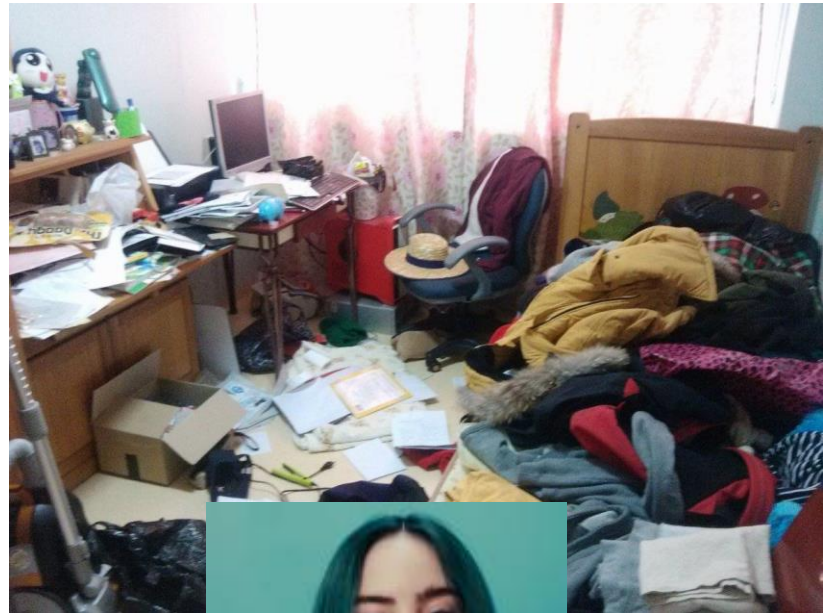
Con't



Object Oriented Programming (OOP)

Alternative Goal in this class

- What is OOP?
 - It is difficult to learn what OOP is



Good kid for Organization



How to organize my room?

OOP is meaningless for Simple Code



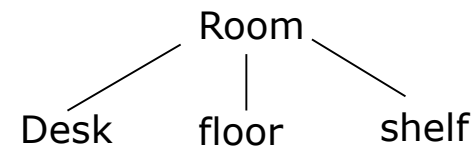
```
void test()
{
    print("babo")
}

main()
{
    test();
}
```

**It is
TOO simple
to learn OOP.**



**We start to
think
How to organize**



Why OOP is hard to learn?

- Complexity and Quantity are the key for OOP.
- You don't see a lot of codes as well as you don't write complex codes.
 - That's why students cannot understand and professors cannot teach OOP concept.
 - (Even most professors have not experience for developing large project)
- Graphics is the mathematical modeling from bottom to top. (Well Categorized Problem)
 - Polyhedron – polygons – lines – vertices.
 - Best example for understanding OOP in our Department.



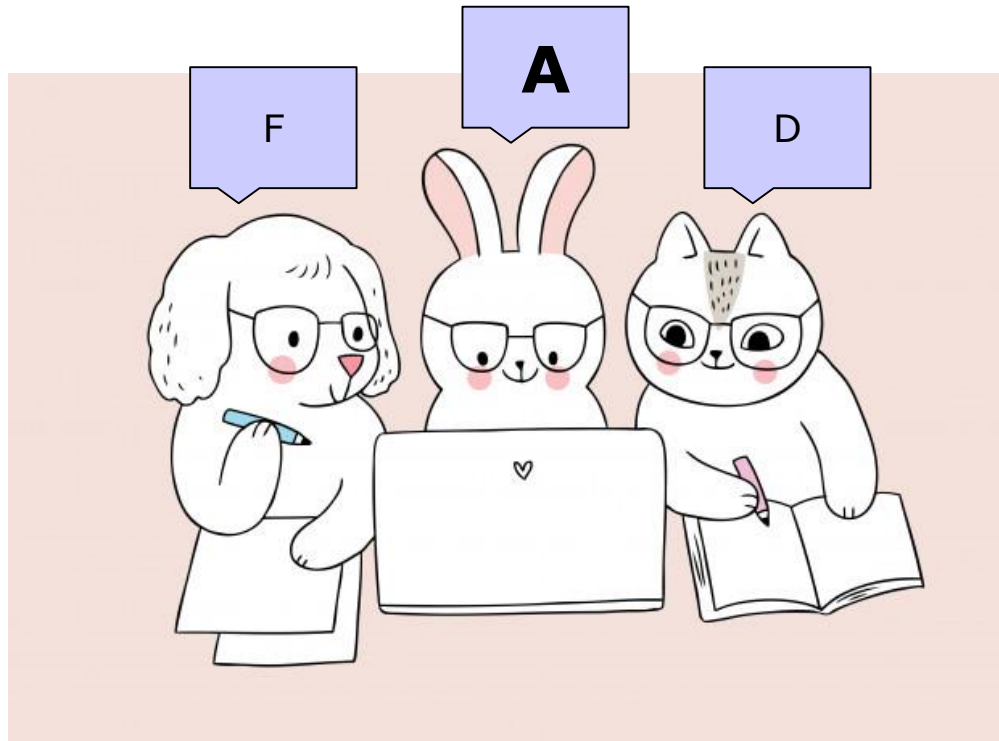
Overview of Grading and Homework

- 1. All HW should be written in C or C++
 - Free version of Visual Studio 2019 will be used.
 - <https://visualstudio.microsoft.com>
- 2. You do Presentation with your Result in every HW.
 - HW with Presentation (100%)



You Can't Get Good Score with **Copied** Materials

- When you do Presentation with friend's HW, Everyone knows you copied HW.



Warning!!

Graphics is a Very Tough Class

- We will cover Step 1,2, and 3 within few months.
- You must do more than 100 homework.



Hundred
Homework!!

Puhahaha...

Grading Policy

- Attendance(0%)
- Presentation with HW (100%)

- My office number is D313-1
 - Please visit my office in the afternoon.

- Download Every course materials from next FTP server
 - IP address: 175.126.99.177
 - Port: 21
 - ID: graphics
 - PW: 123456
 - Be careful that the server is opened only inside university.



Thank you for Listening Graphics Class
Welcome to Homework Hell

