



## ***Syllabus***



### ***Interactive Visualization***

# ***Course Information***

**KUCG**

- **Credit**
  - 3.0
- **Schedule**
  - 14:00, Mon/Wed
- **Attendee**
  - Senior
- **Text Book**
  - Lecture Notes @BlackBoard
- **Prerequisite**
  - Data structure
  - Algorithms
  - Linear algebra
  - Computer graphics

# Schedule

KUCG

일정	강의 내용
9/03	Syllabus
9/05	Installing Tools: CUDA, GPU Server, OpenGL
9/10	HelloWorld: GPGPU, Primitive Drawing
9/12	Basic Drawing <b>Assignment #1</b>
9/17	Transformations
9/19	Scene Graph
9/24	<b>추석 공휴일</b>
9/26	
10/01	Shader Basic
10/03	<b>개천절</b>
10/08	Shader Data Transfer
10/10	Shader Lights
10/15	Shader Texture
10/17	Advanced Shader <b>Assignment #2</b>
10/22	<b>Mid-Term Exam</b>
10/24	

일정	강의 내용
10/29	GPGPU Basic
10/31	GPGPU Thread (Vector, Matrix)
11/05	GPGPU Memory Model <b>Assignment #3</b>
11/07	GPGPU Advanced Computing
11/12	VR Platform
11/14	VR Platform with GPGPU <b>Assignment #4</b>
11/19	Ray Tracing Algorithm ( <b>Mini Project 공지</b> )
11/21	Advanced Data Structure
11/26	Ray Tracing with GPU
11/28	Particle Basic
12/03	Particle Simulation
12/05	Particle System
12/10	Application examples
12/12	Lecture Summary
12/17	<b>Project Due Date</b>
12/19	

# ***In this course, you will learn...***

**KUCG**

## **I. Basic Visualization: OpenGL**

- Visualization with Serial Processing

## **II. Advanced Visualization: GLSL**

- OpenGL Shading Language
- Visualization with Parallel Processing(GPU Computing)

## **III. General-Purpose computing on GPU: CUDA**

- GP computing for Parallel Processing with GPU
- Students can use public GPU Server computer.

## **IV. Visualization Algorithm: Ray Tracing**

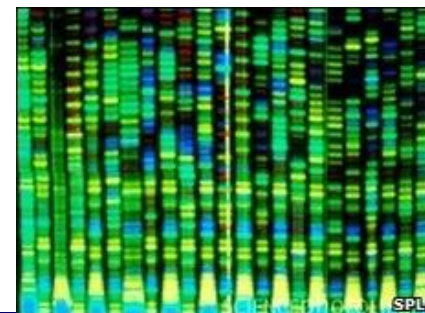
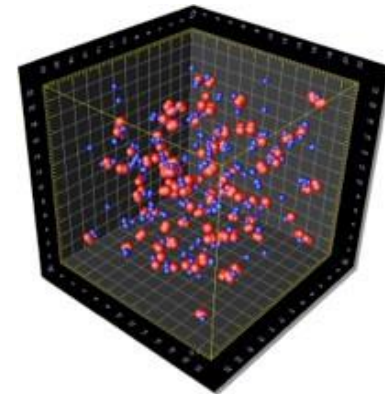
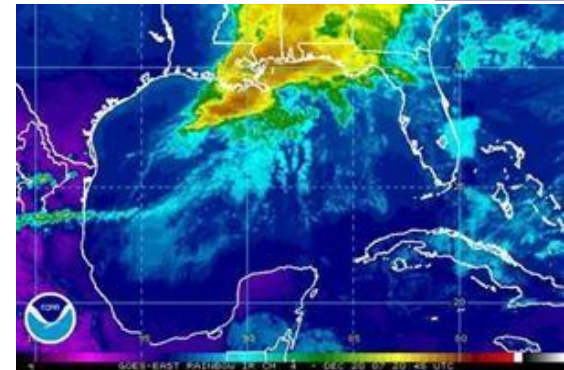
- High Quality & High Performance with GPU

## **V. Application: Particle Systems**

# Why Visualization

KUCG

- Simulations of physical phenomena such as:
  - Weather forecasting
  - Earthquake forecasting
  - Galaxy formation
  - Oil reservoir management
  - Molecular dynamics
- Data Mining: Finding needles of critical information in a haystack of data such as:
  - Bioinformatics
  - Signal processing
  - Detecting storms that might turn into hurricanes
- Visualization: turning a vast sea of data into pictures that scientists can understand.
- At its most basic level, all of these problems involve many, many **complex operations**.

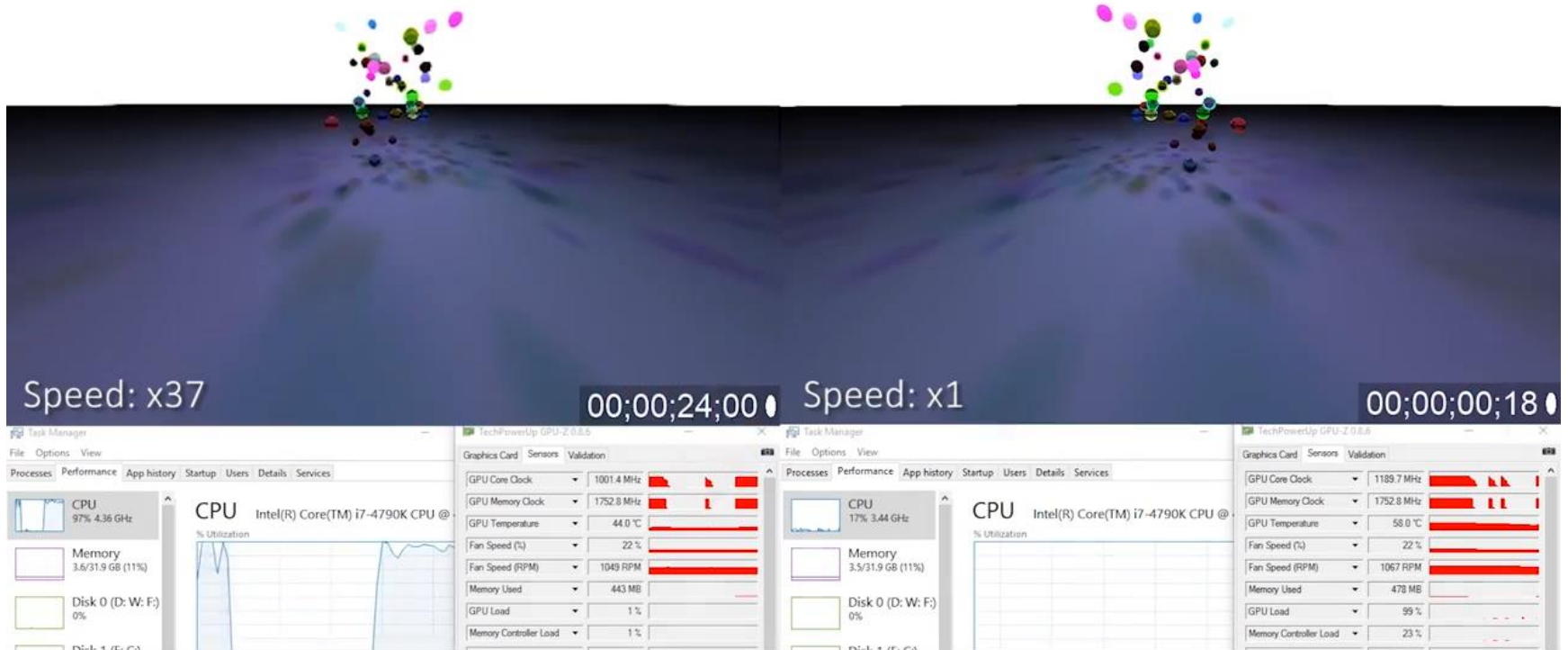


# Ray Tracing@GPU

KUCG

CPU

GPU



# ***Realtime Visualization***

**KUCG**

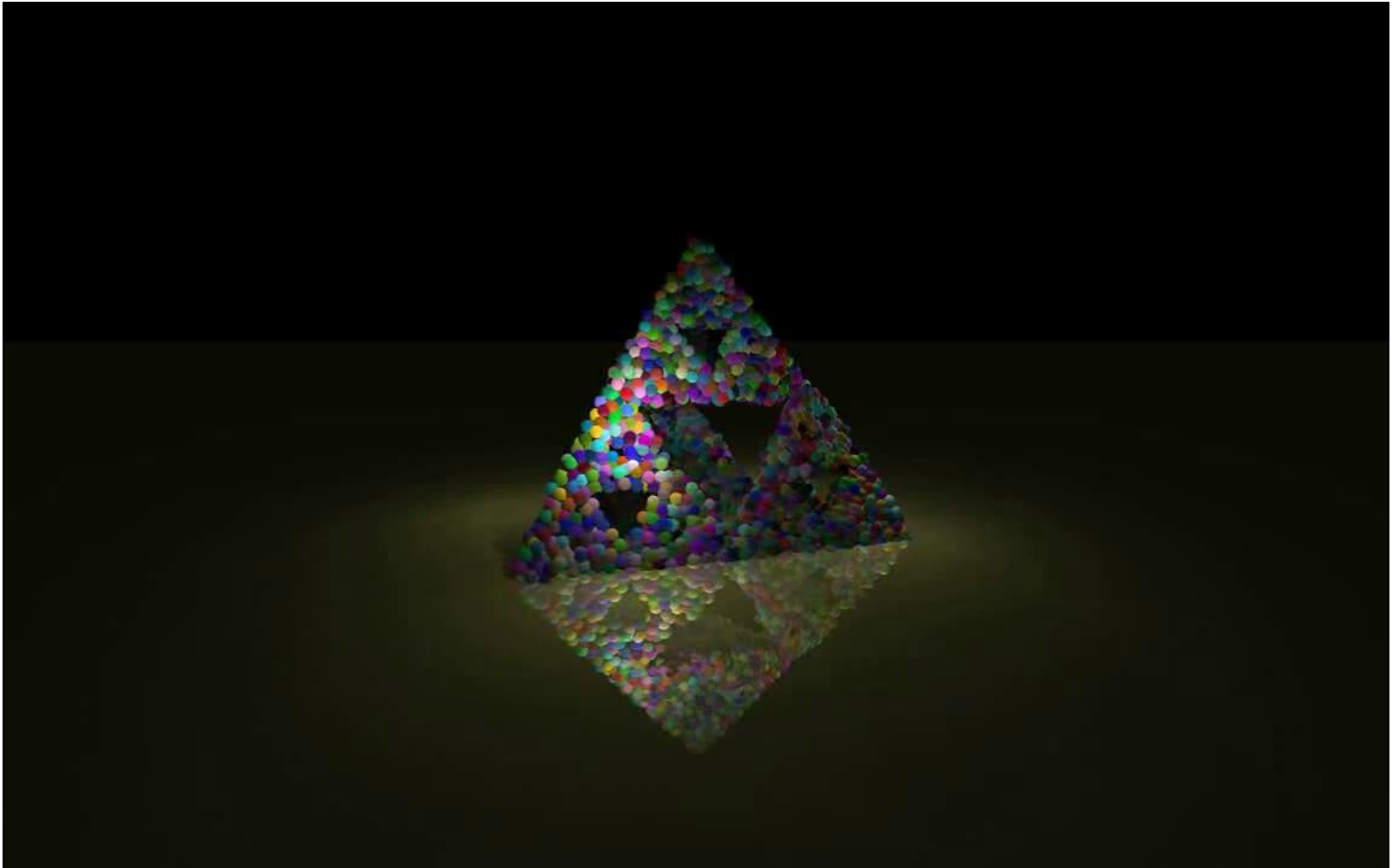




# ***Mini Project:***

## ***Particle+Raytracing+Physics***

**KUCG**





# ***Particles***

**KUCG**



# ***Particle: Collision & Response***

**KUCG**

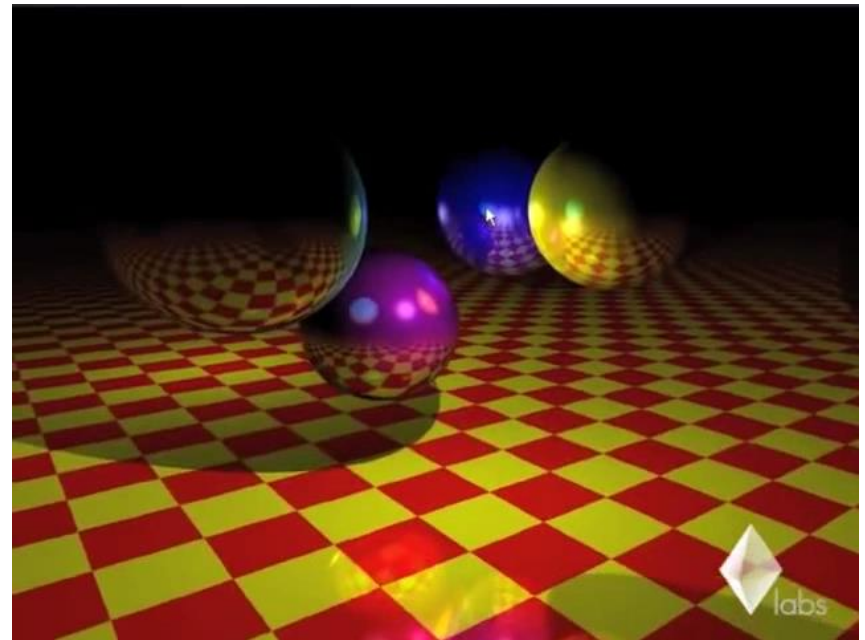
www.fraps.com



# ***Particle: Ray Tracing***

KUCG

GPU



# ***Particle System Applications***

**KUCG**



# ***Course Evaluation***

**KUCG**

- Attendance(10%)
- Midterm exam(25%)
- Programming assignments(40%)
  - 1<sup>st</sup>(10%) + 2<sup>nd</sup>(10%) + 3<sup>rd</sup>(10%) + 4<sup>th</sup>(10%)
- mini project(25%)

***You will fail if you miss just one!!***

# ***Assignment Specification***

**KUCG**

## ■ Themes

- #1) HelloWorld!(CUDA, OpenGL) : 9/12
- #2) Shader Programming : 10/17
- #3) GPGPU Programming : 11/05
- #4) Dynamic Link with VR Platform : 11/14

# Assignment Themes

KUCG

## Assignment #1 : HelloWorld!(CUDA, OpenGL)

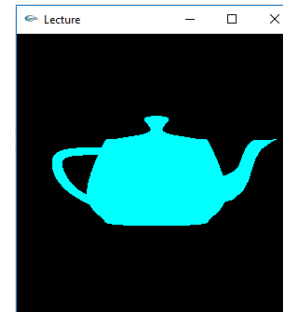
CUDA(with GPU Server), OpenGL의 기본 사용 및 이해

```
#include "cuda_runtime.h"
#include <iostream>
using namespace std;

__global__ void addKernel()
{
    //Do nothing
}

int main()
{
    addKernel
    cout <<
    return 0;
}
```

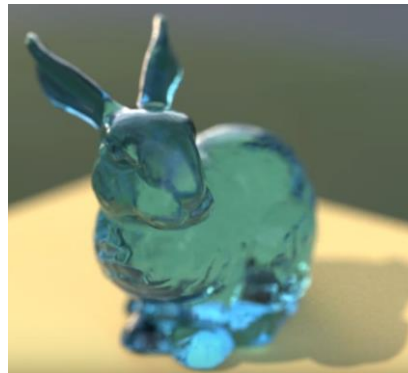
<CUDA: Kernel Calls  
in GPU Server>



<OpenGL: Drawing Teapot>

## Assignment #2 : Shader Programming

Mesh 파일을 읽어서 shader로 Rendering 구현

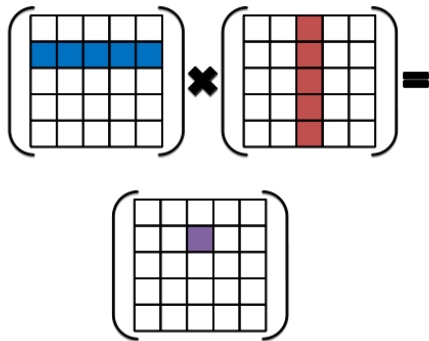




# Homework Themes

KUCG

## Assignment #3 : GPGPU Programming



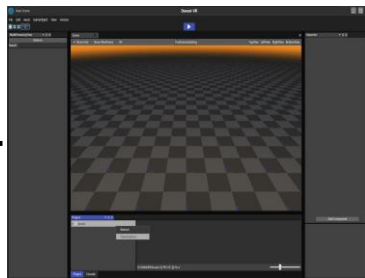
```
__global__ void matrixMultiplication(float* m1/*input*/,  
float* m2/*input*/,  
float* m3/*output*/,  
int sizeN){  
  
    //Write Matrix Multiplication Function with Shared Memory  
  
}
```

## Assignment #4 : Dynamic Link with VR Platform



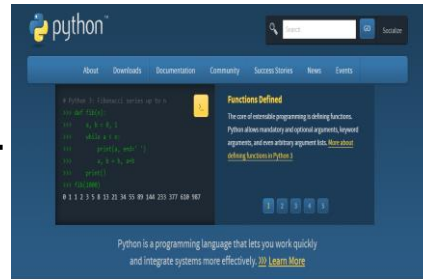
App. DLL

+



VR Platform

+



Dynamic Link with Python

=



- Particle System
  - GPGPU Coding
  - Ray Tracing
  - Do not refer to Other Source Code(Ex: Github.)
- Extra Points
  - Advance Data Structure(Octree, K-d Tree,,, etc.)
  - Collision & Response
  - Dynamic Link with VR Platform
  - Number of Particles in Real-time
- Theme Spec. Notification : 11/26
- Submission Due Date : 12/17

- 담당교수
  - 김창현 교수([chkim@korea.ac.kr](mailto:chkim@korea.ac.kr))
  - 심윤식 교수([neuronomicon@Hotmail.com](mailto:neuronomicon@Hotmail.com))
  
- Teaching assistant
  - Qimeng Zhang([zoe1024@korea.ac.kr](mailto:zoe1024@korea.ac.kr))
  - 박지혁 ([wlgur1014@korea.ac.kr](mailto:wlgur1014@korea.ac.kr))
  
- 그래픽스연구실
  - 우정정보통신관 407B 호