

Final Exam Review

When / Where

Tuesday, 4 December 2018, 12:30-14:30 (12:30 PM - 2:30 PM) CST

Room 226 Tureaud Hall ([Here](#))

Conditions

Closed Book, Closed Notes

Bring one sheet of notes (both sides), 216 mm \times 280 mm.

No use of communication devices.

Format

Several problems, short-answer questions.

Resources

Lecture “slides” used in class: <https://www.ece.lsu.edu/koppel/gpup/ln.html>

Solved tests and homework: <https://www.ece.lsu.edu/koppel/gpup/prev.html>

It’s important to study the solutions.

Study Recommendations

Study this semester's homework assignments. Similar problems may appear on the exam.

For CUDA material look at:

- Fall 2012 Homework 4 (CUDA Basics).

Solve Old Problems—memorizing solutions **is not the same** as solving.

Following and understanding solutions **is not the same as** solving.

Use the solutions for brief hints and to check your own solutions.

Mathematics

Coordinates, Points, Vectors, Homogeneous Coordinates

Dot and Cross Products

Line / Plane Intercept

Transformations

Projections

Coordinate and Vector Classes

pVect, pCoor, pNorm, pMatrix

Use these for basic computations.

Simple Physical Simulation.

Understand how world modeled.

Point masses, ideal springs, gravity field.

Time Step

Updating velocity and position.

Forces

Gravity.

Ideal spring.

Simple Collisions.

OpenGL Coordinate Spaces

Object, Eye, Clip, Window

OpenGL Primitives and Vertex Specification

Primitives

Triangles, triangle strips, etc.

Vertex attributes.

Vertex (coordinate), color, normal, etc.

Estimate amount of data needed.

OpenGL Rendering Passes; Arrays and Buffer Objects

Rendering Passes

```
glBegin( PRIM )
```

```
glDrawArrays( PRIM, ... )
```

```
glDrawArraysInstanced( PRIM, ... )
```

For each: Where do vertex shader inputs come from?

Difference between array on CPU (client) and buffer object.

Estimate amount of data sent between CPU and GPU.

OpenGL Textures

Basic Idea

Texture Filtering: Minification, Magnification, mipmap levels.

Linear/Nearest

Texture application.

OpenGL Rendering Pipeline

The Stages.

Fixed Functionality v. Programmable Stage.

Shader Programming

Programmable Shaders

Vertex, Geometry, Fragment.

For Each One:

Inputs, Outputs.

Conventional functionality.

CUDA

Address Spaces

Aspects

How to declare a variable so that it uses the address space.

Scope: thread, block, all.

Read/write or read only.

Speed: Fast or really slow.

- Global
- Constant
- Shared

Thread Organization

Threads and Thread Groupings:

Thread

Warp

Block

Grid

Global Memory Use

Pointer declaration.

Allocation

GPU/CPU movement.

Efficient access patterns.

Programming for 1D Array Access

How to assign threads to array elements.

How to access memory efficiently.