EE 4702—GPU Programming

Midterm Exam Review

When / Where

Wednesday, 18 October 2017, 11:30-12:20 CDT

1218 PFT (Here)

Conditions

Closed Book, Closed Notes

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

No use of communication devices.

Format

Several problems, short-answer questions.

Resources

Lecture “slides” used in class: `http://www.ece.lsu.edu/koppel/gpup/ln.html`

Solved tests and homework:. `http://www.ece.lsu.edu/koppel/gpup/prev.html`

Topics for Exam

Everything up to and including shader programming.

Study Recommendations

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

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.

Emphases

Vertex specification: `glBegin` v. `glDrawArray` . . .
. . . client arrays v. buffer objects

Rendering pipeline stages, and their inputs and outputs.

Shader inputs v. uniform variables v. buffer object arrays.

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.

Geometric Operations

Should be able to write code such as:

Given two points, find 7 equally spaced points between them.

Given points A and B, find points on a circle with center A passing through B.

Given points A, B, and C, find points on the surface of a cylinder whose axis passes through A and B, and whose surface passes through C.

Given points of a triangle, find triangle normal.

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 Arrays and Buffer Objects

Difference between glBegin(PRIM), glDraw(PRIM)

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

Amount of Data

Estimate amount of data required for individual items.

Examples:

`pCoor`, four floats $= 4 \times 4\,\mathrm{B} = 16\,\mathrm{B}$.

`glVertex3f`, three floats $= 3 \times 4\,\mathrm{B} = 12\,\mathrm{B}$.

Estimate amount of data sent between CPU and GPU.

Uniform v. Vertex Shader Input.

Buffer Object v. Client Array.

Unchanging shader input. (E.g., `glColor` as used in class.)

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. Compute

For Each One:

Inputs, Outputs.

Conventional functionality.