EE 4702—GPU Programming

Final Exam Review

When / Where

Solve-Home. Due Friday, 10 December 2021 CST

Conditions

No collaboration.

Can search for and use background material.

Do not search for answers to specific questions.

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.

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.

Coordinate Spaces

OpenGL: Object, Eye, Clip, Window

Vulkan Rasterization: Clip, Window

Vulkan Ray Tracing: Object, World, Clip, Window

Vulkan Primitives and Vertex Specification

Primitives (Primitive Topology)

Vulkan 1.2 Section 21.1 (`https://www.khronos.org/registry/vulkan/specs/1.2-extensions/html/chap21.html`)

- `vk::PrimitiveTopology::eTriangleList`
- `vk::PrimitiveTopology::eTriangleStrip`
- `vk::PrimitiveTopology::eLineList`
- `vk::PrimitiveTopology::eLineStrip`
- `vk::PrimitiveTopology::ePointList`

Vertex Shader Inputs.

- Vertex (coordinate), color, normal, etc.

Estimate amount of data needed.

Vulkan and OpenGL Textures

See `https://www.ece.lsu.edu/koppel/gpup/2021/demo-08-texture.cc.html`

Basic Idea

Texture Filtering: Minification, Magnification, mipmap levels.

Linear/Nearest

Texture application.

Vulkan Rasterization Rendering Pipeline

The Stages: Vertex, Geometry, Fragment

Fixed Functionality v. Programmable Stage. (See `https://www.ece.lsu.edu/koppel/gpup/2021/set-3-rend-pipe.pdf`)

Shader Programming

Programmable Shaders

Vertex, Geometry, Fragment.

For Each One:

Inputs, Outputs.

Conventional functionality.

Ray Tracing

Ray Generation Shader

Role as starting point for rendering.

Role as writer of the frame buffer.

Ray Traversal

Shaders called during traversal:

Intersection, Any Hit, Closest Hit, Miss

Brute force traversal (check all geometry for each ray) v. accelerated traversal (use a bounding-volume hierarchy or other tree-like structure to reduce amount of checked geometry).