## **CS 535 Computer Graphics**

#### Syllabus (Fall 2025)

Required/Elective: Elective Prerequisites: CS315, CS335, (CS321/CS322)

## Syllabus (Fall 2025)

**General Information Topics Covered Computer Facilities** Grading Policy Course Summary & Program Outcomes Plagiarism & Cheating Important Dates CS Dept, UK

Location: Mine and Minerals Research Bldg, Rm 112 Time: TR 12:30 – 1:45pm Instructor: Dr. Fuhua (Frank) Cheng OFFICE: DMB 303 OFFICE HOURS: TR 3:30pm-5:30pm and by appointment PHONE: (859) 257-6760 E-MAIL: cheng@cs.uky.edu CS Dept, UK

3

#### CLASS WEBSITE:

http://www.cs.uky.edu/~cheng/cs535/CS535-HomePage-2025f.htm

Or, go to my personal WEBSITE first: http://www.cs.uky.edu/~cheng/

Then scroll down to 'Teaching' and click on 'CS 535'

## KENTUCKY

Will be followed in the following order.

- Intro 8 2D Raster algorithms
  OpenGL and Shaders
  Example programs (1) (2) (3) (4)
  (5) (5) (5)
- 3D Viewing Example program (1)
   3D Data Structures, 3D Data
- Management and 3D Models Example programs (1-1) (1-2) (1-3) (1-4) (2) (3) (4) (5) (6) (7)
- Texture Mapping I
  Example Program (1)
- Hidden Surface Elimination
- Lighting and Shadows I Exemple programs (1-1) (1-2) (4-3)
- Lighting and Shadows II Sample program (2) Sample program (3)
- Ray Tracing I
  Example programs (1) (2) (3+4)
  (5) (5) (7) (8) (9) (10) (11) (12)
  (13+14+15+15)
- Ray Tracing II (Solid Modeling)
- Texture Mapping II
  Example program (1b)
- Curves and Surfaces V (Surface Modeling)

#### Related Sites

#### <u>Stanford</u> (Undergrad/Grad) MIT(Undergrad)

- CMU(Grad) Berkeley(Undergrad/Grad)
- Callech(Grad)
  UIUC (Undergrad/Grad)
- Comel (Undergrad/Grad) Columbia (Undergrad)
- USC<u>(Undergrad)</u> (Grad)

#### Final Exam

Game Download



#### Creator: Seth Parker CS535-2021f)

This is the home page of CSS35: Intermediate Computer Graphics.

Instructor:

#### Dr. Fuhua (Frank) Cheng (cheng@cs.uky.edu)

This course covers 3D graphics primitives and OpenGL 3D shader programming. It will help you achieve the following educational objectives: (1) understand the concept of 3D viewing, lighting and rendering process; (2) understand the concept of hidden surface elimination, and know techniques that can be used for such a process, as well as offens to determine if a method is appropriate for particular hidden surface elimination problems; (3) be familiar with the shading process; (4) understand the concept of shadow generation and know how to choose shadowgeneration methods for 3D rendering problems; (5) understand how to use ray tracing technique to generate a high quality image, how to use CBG trees to represent solids, and how to render a CBG-represented solid; (6) understand how to use B-splines in 3D shape representations; (7) be able to write OpsnGL programs to use shaders to render 3D scenes consisted of polygonal objects and objects bounded by tree-form surfaces.

- The Gurse <u>SVIIabus</u> is available CHTML format. <u>Campus resources</u>, including <u>engineering tutoring</u>, <u>UK Counseling Center</u> and <u>Center for Support and</u> <u>Intervention</u>, you can use (for free).
- see <u>Title IV Regulation</u> as
- Homework Assignments Homework Solution Sets
  - Programming Assignments
  - -----
  - Takehome Exams



Exam dates:

C \$535: Intermediate Computer Graphics

- Android's web site for OpenGL Android's web site for related OpenGL instructions categories References on OpenGL
- Computer Grephics through OpenGL's website
- Segmentation Anything Model

Sample Programs for Programming Assignments

#### CS Dept, UK

CLASS

WEBSITE:

#### TEXTBOOK: Fundamentals of Computer Graphics (5th Edition)

#### by Steve Marschner and Peter Shirley

(publisher: A K Peters/CRC Press) (ISBN-10:0367505037 ISBN-13:978-0367505035)

and my notes (can be downloaded from the class website)

#### Fundamentals of Computer Graphics

FOURTH EDITION

# <complex-block><complex-block>

#### **REFERENCE BOOKS:**

Computer Graphics: Principles and Practice,

2nd edition in C by Foley, van Dam, Feiner, and Hughes





THE SYSTEMS PROGRAMMING SERIES

REFERENCE BOOKS: COMPUTER GRAPHICS PROGRAMMING *in OpenGL with C++* 3rd Edition

by V. Scott Gordon and John Clevenger

#### COMPUTER GRAPHICS PROGRAMMING IN OPENGL WITH C++



#### **REFERENCE BOOKS:**

OpenGL Programming Guide: The Official Guide to Learning OpenGL,

Versions 4.0 (8th Edition)

by Dave Shreiner, Graham Sellers, John Kessenich, Bill Licea-Kane



John Kessenich \* Graham Sellers \* Dave Shreiner The Khronos OpenGL ARB Working Group

#### Weeks 1:

- **Basics:** graphics systems, modeling, rendering, input and interaction
- **2D Graphics**: concept of RGBα and rasterization of polygons

#### Weeks 2-3:

10

**OpenGL and Shaders**: basic structure, utility libraries, callback function prototypes, examples

Week 3-4:

**3D Viewing:** projections, geometric transformations, graphics transformation

Week 4-5:

**3D Data Structures, 3D Data Management and 3D Models:** face-table based, winged-edge data structure, scene graphs, Managing 3D data, and 3D models

Weeks 6:

Hidden Surface Elimination: overwriting, backface culling, Z-buffer, scan-line method, BSPtree method

#### Weeks 7-8:

Illumination, Shading (lighting) and Shadows: Gouraud shading, Phong shading, shadow volume method, shadow map method

Weeks 9-10:

**Ray Tracing I:** ray tracing, shadow generation, specular reflection, refraction, instancing

Weeks 10-11:

Ray Tracing II: Solid modeling, ray casting

Week 12: **Texture Mapping**: Week 13-14: **Curves and Surfaces**:

## **Computer Facilities:**

You can either use your own computer or the computers in the Multilab to do programming assignments for this class.

Your userid for the Multilab will be mailed to you during the first week of the semester. If you did not get it, please send me email so I can send your userid to you again.

This is a Linux laboratory administered by the Computer Science Dept.

## **Computer Facilities:**

Lab Location: Hardyman Building

15

You may use alternative computer systems for developing and testing your work. But your submitted work must compile and run under the proper software environment.

This course uses OpenGL to support graphics operations.

## Get Prepared:

Follow the instructions given underneath "Get Prepared" to install all the header files and things you need for this class

#### KENTÜCKY

#### Lecture Notes

- fill be followed in the following order.
- Intro & 2D Raster algorithms
  OpenGL and Shaders
  Example programs (1) (2) (3) (4)
  (5) (5) (5)
- <u>3D Viewing</u> <u>Example program (1)</u>
   <u>3D Data Structures, 3D Data</u> <u>Management and 3D Models</u> <u>Example programs (1-1) (1-2)</u> (<u>1-3) (1-4) (2) (3) (4) (5) (6) (7)</u>
- <u>Texture Mapping I</u>
  <u>Example Program (1)</u>
- Hidden Surface Elimination
  Lighting and Shadows [
- Example programs (1-1) (1-2) (1-3) Lighting and Shadows II
- Lighting and Shadows II Sample program (2) Sample program (3)
- <u>Ray Tracing I</u> <u>Example programs (1) (2) (3+4)</u> (5) (6) (7) (8) (9) (10) (11) (12) (13+14+15+16)
- Ray Tracing II (Solid Modeling)
- Texture Mapping II
  Example program (1b)
- <u>Curves and Surfaces V (Surface</u> <u>Modeling)</u>

#### Related Sites

- Stanford (Undergrad/Grad)
- <u>MIT(Undergrad</u>)
  CMU(Grad)
- Berkeley(Undergrad/Grad)
- <u>Caltech</u>(Grad)
  <u>UIUC</u> (Undergrad/Grad)
- Cornell (Undergrad/Grad)
- Columbia (Undergrad)
- USC(Undergrad) (Grad)



(Creator: Seth Parker CS535-2021f)

This course covers 3D graphics primitives and OpenGL 3D It will help you achieve the following educational objectives: (1) understand the concept of 3D viewing, lighting and rendering process; (2) understand the concept of hidden surface elimination, and know techniques that can be used for such a process, as well as criteria to determine if a method is appropriate for particular hidden surface elimination problems; (3) be familiar with the shading process; (4) understand the concept of shadow generation and know how to choose shadowgeneration methods for 3D rendering problems; (5) understand how to use ray tracing technique to generate a high quality image, how to use CSG trees to represent solids, and how to render a CSG-represented solid; (6) understand how to use B-splines in 3D shape representation and how to perform subdivision and tessellation on B-spline based representations; (7) be able to write OpenGL programs to use shaders to render 3D scenes consisted of polygonal objects and objects bounded by free-form surfaces.

This is the home page of CS535: Intermediate Computer

The course  $\underline{Syllabus}$  is available in HTML format.

#### Campus resources, including engineering tutoring,

- UK Counseling Center and Center for Support and Intervention, you can use (for free).
- See Title IV Regulation here.



Exam dates:



## **Grading Policy:**

Programming Assignments (3 assignments)---- 40% Midterm ----- 20% Final ----- 20% Class attendance (extra credit) ----- 5% Homework (6-8 assignments)----- 20%

• You get the attendance credit (5 points) if you miss at most two lectures in the semester

## **Grading Policy:**

\*\* Programming assignments may be done in C++. Example programs in C++ will be provided.

\*\*\* You may use ChatGPT or any GAI to help with your HW or programming assignments.

#### LATE PENALITY:

I will accept programs and homework up to two days late for a penalty of 20% (10% each day).

## **Grading Policy:**

## SCALE (graduate students):

- 90 -105 .... A
- 80 89 ..... B
- 70 79 ..... C

SCALE (for under-

- graduate students)
  - 86 -105 .... A
- 76 85 ..... B
- 66 75 ..... C
- 56 65 .... D
- 0 55 ..... E

## **Course Summary & Program Outcomes:**

This course covers 3D graphics primitives such as 3D viewing, 3D data structures, hidde line/ surface elimination, illumination and shading, and more advanced topics such as ray tracing, solid modeling, texture mappings, curves and surfaces, advanced raster graphics architecture and algorithms, and advanced modeling techniques if time permits.

# Course Summary & Program Outcomes:

Specific skills:

- An understanding of the graphic system and shader programming (input devices, scan conversion, graphics storage, graphic processing unit, output devices, graphics pipeline and shaders)
  - An understanding of 3D viewing (projections, geometric/graphics transformations)
  - An ability to use appropriate data structures (facetable based, winged-edge data structure, scene graphs) to represent 3D objects and to manage 3D data and building 3D models

## **Course Summary & Program Outcomes:**

#### Specific skills:

- An ability to apply appropriate techniques (overwriting, Z-buffer, scan-line method, BSP-tree method) to elliminate hidden lines/surfaces in 3D rendering process
- An ability to apply appropriate techniques (Gouraud shading, Phong shading, shadow volume method, shadow map method) to shade objects and to create shadows in the rendering process

## **Course Summary & Program Outcomes:**

#### Specific skills:

- An ability to use appropriate mapping techniques to create textures for objects in the rendering process
- A fluency in graphic and shader (especially compute shader) programming using OpenGL as the supporting graphic system.

## **Plagiarism & Cheating:**

Consult the following links for information on what constitutes an academic offense and on applicable penalties:

http://www.uky.edu/Ombud/ http://www.uky.edu/Ombud/Plagiarism.pdf

## **Important Dates:**

25

First day of class - 8/26/2025 (Tuesday) Last day to drop without a W or change grading option - 9/12/2025 (Friday) Midterm - 10/16/2025 (Thursday) Fall Break - 10/27/2025-10/28/2025 (M&T) Thanks Giving Holidays - 11/26/2025-11/29/2025 (Wednesday-Saturday) Last day of classes - 12/09/2025 (Tuesday) Final Exam – 10:30-12:30, 12/16/2025 (Tuesday)

