Syllabus – CS 371 – Computer Graphics – Spring 2016

INSTRUCTOR: Tom Naps

OFFICE: Halsey 214, phone 424-1388

EMAIL: naps@uwosh.edu

OFFICE HOURS: MWF 9:30 - 10:30, Thursday 9:15 - 11:15

REFERENCES:

- Daily class handouts available by 4:00am MWF on D2L. They comprise a high-level, but not detailed, outline of what we will cover during class. Get a copy of them before class, organize them, take notes on and about them. Handouts that are not liberally saturated with your own explanatory notes will likely prove useless when you need them most
- Lots of programs that will allow you to do "what-if" experimentation (I use these myself to make up devilish test questions)
- Review problems posted on D2L at the end of each class. Complete them by 4:00am on the day of the following class meeting and be ready to discuss the answers you have submitted (e.g., review problems posted Wednesday are due by 4:00am Friday).
- Optional but good references: (1) *OpenGL Programming Guide*, a.k.a. the "Red Book" by Woo, Neider, David, and Schreiner, (2) *Interactive computer graphics : a top-down approach with WebGL* by Edward Angel, and (3) *Computer Graphics Using OpenGL* by F.S. Hill and S.K. Kelley

Topic Coverage

1. Graphics systems in general – the fixed function pipeline (OpenGL), the programmable pipeline (GLSL), and the "algorithmic rendering ladder"
2. Two-dimensional graphics
3. Math necessities beyond the course prerequisites
4. Transformations in two- and three-dimensions
5. Hierarchical modeling of 3-D objects
6. 3-D Viewing with the synthetic camera
7. Modeling 3-D shapes with polygon meshes
   - Meshes obtained from various data collections
   - Meshes obtained from "pure" mathematical surfaces
   - Approximating/interpolating curves and surfaces
8. Lighting and Shading
   - The Phong reflection model
   - Gouraud shading in the fixed function pipeline
   - Texture mapping in the fixed function pipeline
   - Vertex and Fragment shaders – using the programmable pipeline
   - Phong shading
   - Bump mapping
   - Ray tracing
   - Photon mapping
   - Radiosity
9. Raster Algorithms – as time allows

Learning Outcomes

Given our coverage of these topics, you will be expected to ...

1. Identify and define the purpose of each component in the graphics pipeline that transforms a vertex in world coordinates to a pixel location
2. To perform in manual fashion the transformation carried out by the graphics pipeline on points in two-dimensional and three-dimensional world coordinate space
3. Discuss the relationship between the aspect ratio of a scene and the viewport in which it is rendered
4. Trace and describe a selection of classic raster algorithms such as the Cohen-Sutherland two-dimensional clipping algorithm on points in world-coordinate space, Bresenham's line-drawing algorithm in two-dimensional space, and scan-line conversion of polygons.
5. Define and discuss the role of double buffering in real-time animations
6. Apply linear affine transformations such as scaling, translation, and rotation to points in two- and three-dimensional space and analyze the effects of such transformations on the points in a rendered scene
7. Define and perform the perspective and orthographic projections on points and scenes in three-dimensional space
8. Compare scenes rendered by perspective and orthographic projections
9. Plan and design scenes animated by an underlying hierarchical model
10. Identify the role of the model-view transformation and its matrix representation in rendering hierarchical models

11. Define the roles of the eye point, look point, and up vector parameters in the synthetic camera's view of a three-dimensional scene and to perform the computations necessary to illustrate how these parameters affect the model-view transformation matrix
12. Trace the depth-buffer (Z-buffer) algorithm as it is used to determine hidden points and surfaces in a rendered scene
13. Define and compare the variety of transformations used in texture mapping to identify a point in texture space with a point in world coordinate space
14. To discuss the mathematics underlying two- and three-dimensional interpolating curves and surfaces (for example, Bezier curves and surfaces)
15. Discuss the roles played by color, lighting, and material parameters in the progression of increasingly sophisticated shading models - flat, smooth, Gouraud, Phong, ray-tracing, radiosity, and photon-mapping
16. Using visual clues, differentiate between scenes rendered by a variety of shading models such as flat, smooth, Gouraud, Phong, ray-tracing, radiosity, and photon-mapping
17. Analyze the relationship between computational rendering algorithms for increasingly sophisticated shading models - flat, smooth, Gouraud, Phong, ray-tracing, radiosity, and photon-mapping - and the time required to render the scene using that algorithm
18. Using a graphics library such as OpenGL, implement three-dimensional animations rendered in real-time using an appropriate lighting model built into the fixed function pipeline.
19. Using a shader language such as GLSL implement scenes that employ lighting and shading algorithms that are above and beyond the fixed function pipeline.

Course Grading Policies

Your grade for the course will be based on the following weighted factors:

Factor	**Weight**
6-8 Assignments	45% in total
Class participation and preparation	15%
Exams:	40%

There will be four exams during the semester. To get the 40% contribution to your grade from the four exams, I will use the formula:

$$E = 0.05 \times E_{worst} + 0.15 \times E_{best} + 0.10 \times E_{other1} + 0.10 \times E_{other2}$$

where E_{worst} is your worst exam score and E_{best} is your best exam score.

Unlike other courses I teach, in this course meeting the specifications on a programming assignment will only guarantee a 90% grade. To get 100% (or qualify for golly-gee-whiz points), you must provide add-on features that are clearly explained and enumerated in the opening documentation block that accompanies your program.

At the end of the term, your work in all of these areas will contribute to a numerical grade for the course based on a 100-point scale. Grade cutoff levels on this final scale are:

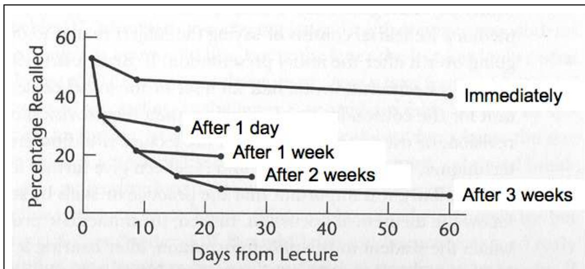
A $\geq$ 92	B $\geq$ 82	C $\geq$ 72	D $\geq$ 62
A- $\geq$ 90	B- $\geq$ 80	C- $\geq$ 70	D- $\geq$ 60
B+ $\geq$ 88	C+ $\geq$ 78	D+ $\geq$ 68	F $<$ 60

FAQ

Do I have to come to class? You are expected to arrive prepared to ALL the course sessions. Furthermore you are expected to participate in the classroom discussions and activities to the best of your abilities. This includes being ready to defend your answer to the review problems from the previous class (more on that later). It is difficult to envision a student missing and/or arriving unprepared to a number of the class sessions and still succeeding in the course.

How much time will this course take? Figure about three hours outside of class for each hour in class. That heuristic makes being a full-time student pretty much equivalent to holding a full-time job, so this is really good preparation for the real world that awaits you after graduation.

How can I best prepare for the exams? We've known what the following graph illustrates since 1968:



Bassey M. (1968), in *Learning methods in tertiary education*

More interesting evidence . . .

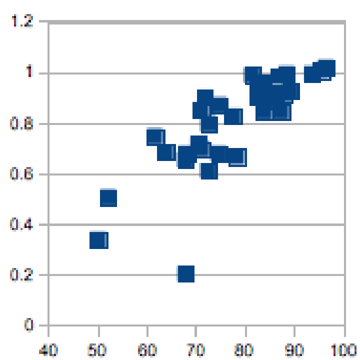
> "Research has demonstrated that after a lecture, students recall 62% of the information. However, only 45% is recalled by students after 3-4 days and in 8 weeks only 24% of the information is recalled. If a quiz or exam was administered after the lecture, recall was doubled at the 8-week period. It is interesting that many faculty members appear to ignore the potential impact which quizzes and tests can have upon learning." – Bonwell C.C., Eison J.A.: Active Learning: Creating Excitement in the Classroom. Washington, DC: George Washington University, 1991.

Consequently at the end of most of our class periods you will find on D2L a small set of review problems covering what we discussed in class. The time to work on these review problems is immediately after the material is covered in class. You are also encouraged to discuss review problems with your classmates in a spirit of mutual help toward better understanding of how to solve them.

We will always discuss the review problems at the beginning of the class following their distribution. Your solutions to the review problems are due no later than 4:00am on the Monday, Wednesday, or Friday following the day of their distribution. (e.g., review problems posted Wednesday are due by 4:00am Friday.) These solutions should be submitted using the link to a blank "quiz" form for the review problems that you will find on D2L.

If you have participated in class the day the review problem was distributed, have made a good faith effort to work on the review problem, and are "stuck" on it, I will be more than happy to help you with it if you come my office anytime within three days after you have received the review problem in class. After those three days (not counting weekends), *because you have made the choice to not learn effectively*, you are on your own in terms of grappling with these review problems.

Although the review problems only count 15% of your grade, the following correlation from a previous course between review-problem-percentage (on a 0 to 1.0 vertical scale) and overall percent in the course (on a horizontal scale of 0 to 100) is indicative of their true importance.



What if I'm late in submitting an assignment for evaluation? Each assignment will carry with it a due date. If you are late in submitting it for evaluation, it will be accepted but will be penalized at the rate of 10% of point value the first day late, *an additional* 20% the second, *an additional* 30% the third . . .

Is there any way I can carelessly lose points in the course? Yes . . .

- Be late in submitting your work for evaluation on assignments.
- Don't participate in and prepare for the class.

What is this class participation/preparation stuff? How does it add up to 15% of my grade? . . .

- Be sure to get those review problems done and submitted on time
- Exhibit your knowledge when called on to explain your answer to a review problem.
- Exhibit your knowledge when called on to respond to other questions in class

Is there any way I can get some bonus points? . . .

- Take advantage of various bonus "challenges" that will accompany some of your programming assignments
- Do an outstanding job when called on to explain your answer to a review problem.
- You can get "golly-gee-whiz" points by being creative and going *significantly* beyond the plain vanilla specifications for your programming assignments. This will enable you to fare well in the vote for the "best program" on each assignment.

Can I get an extension on work that is due on a specified date? Only if you're ill enough to provide a signed note from the attending physician or have other reasons serious enough that the Dean of Students Office is willing to provide a written note justifying the extension.

If I miss a test, can I make it up? If you are unable to take a scheduled exam, it may be possible to take a make-up exam provided that you do BOTH of the following, which are then subject to my approval:

- Make arrangements prior to the scheduled exam (for last minute emergencies, telephone me at 424-1388 or leave a message at the computer science office, 424-2068). No after-the-fact notifications will be accepted ... *AND*
- Have a written medical excuse signed by the attending physician OR have a note of justification from the Dean of Students Office.

Only one make-up exam will be given. It will be a rigorous comprehensive exam given at an arranged time during the last week of the semester.

Can I work with others on assignments?

On "regular" assignments No, not in the sense of two people working on the same program. However, it is acceptable to consult another student for help in debugging a program that you have authored yourself and that is not producing the result you expected. It is also acceptable to cut-and-paste code snippets from in-class demos or examples you find on the internet *provided* that you cite the sources of these code snippets in the introductory documentation block at the beginning of your program.

On the one "special project" assignment This assignment will be given about eight weeks into the semester. You will be able to work in teams of two on it. You task will be to create a game given a framework that I provide as a starting point or create an appropriate interactive 3-D visualization of data from another discipline. It will count twice the weight of a regular assignment. Each team will be required to demonstrate their project in class during the final week of the semester.