Idaho State University | Computer Science

CS 4458 | CS 5558 – Fall 2019
Computer Graphics

Instructor: Isaac Griffith

Office: BA 315
Phone: (208) 282-4876
Email: grifisaa@isu.ed
URL: https://www2.cose.isu.edu/~grifisaa/
Office Hours:

- **TR**: 1430 – 1530
- By appointment scheduled at: https://isaac-griffith.youcanbook.me/

Course Information

Meeting Time: TR: 16:00 – 17:15 **Room:** Pocatello: BA 506, Idaho Falls: TA 286

Final Exam Dates

Section: 4458/5558-01 – Pocatello
Date: Tuesday Dec 10, 2019
Time: 1730 – 1930
Room: BA 506

Section: 4458/5558-02 – Idaho Falls
Date: Tuesday Dec 10, 2019
Time: 1730 – 1930
Room: Idaho Falls Testing Center

Prerequisites:

- MATH 2240
- CS 4412

Textbooks:

- **Computer Graphics Through OpenGL: From Theory to Experiments, 3rd Edition** – Guha. ISBN:

978-1138612648

Course Description

Graphics, transformation matrices, lighting models, object hierarchies, visible surface determination, ray tracing.

Technology

This is a Software Engineering course and thus will heavily rely upon the use of current technology. As a student you are expected to have at your disposal a computer system capable of running a recent operating system such as MacOS X, Windows 10, or Linux. To program in OpenGL your computer will need a graphics card capable of running OpenGL 3.0+. In addition, you will be required to use Git, a professional grade IDE (e.g., JetBrains CLion or Visual Studio) and build system. Furthermore, you, as a future technologist, are responsible for installing and learning to use these tools.

Learning Outcomes

The following outcomes will be evaluated via homework assignments and exams

GV/Fundamental Concepts

- Identify common uses of digital presentation to humans (e.g., computer graphics, sound). [F]
- Explain in general terms how analog signals can be reasonably represented by discrete samples, for example, how images can be represented by pixels. [F]
- Explain how the limits of human perception affect choices about the digital representation of analog signals. [F]
- Describe the differences between lossy and lossless image compression techniques, for example as reflected in common graphics image file formats such as JPG, PNG, MP3, MP4, and GIF. [F]
- Describe color models and their use in graphics display devices. [F]
- Describe the tradeoffs between storing information vs. storing enough information to reproduce the information, as in the difference between vector and raster rendering. [F]
- Describe the basic process of producing continuous motion from a sequence of discrete frames (sometimes called “flicker fusion”). [F]
- Describe how double-buffering can remove flicker from animation. [F]

GV/Basic rendering

- Discuss the light transport problem and its relation to numerical integration i.e., light is emitted, scatters around the scene, and is measured by the eye. [F]
- Describe the basic graphics pipeline and how forward and backward rendering factor in this. [F]
- Contrast forward and backward rendering. [A]
- Explain the concept and applications of texture mapping, sampling, and anti-aliasing. [F]

- Explain the ray tracing/rasterization duality for the visibility problem. [F]
- Compare and contrast the different rendering techniques [A]
- Computer space requirements based on resolution and color coding. [A]
- Computer time requirements based on refresh rates, rasterization techniques. [A]

GV/Geometric Modeling

- Contrast modeling approaches with respect to space and time complexity and quality of image [A]

GV/Advanced Rendering

- Demonstrate how an algorithm estimates a solution to the rendering equation [A]
- Prove the properties of a rendering algorithm, e.g., complete, consistent and unbiased [A]
- Analyze the bandwidth and computation demands of a simple algorithm [A]
- Discuss how a particular artistic technique might be implemented in a renderer. [F]
- Explain how to recognize the graphics techniques used to create a particular image. [F]

GV/Computer Animation

- Describe the tradeoffs in different representations of rotations [A]
- Discuss the basic ideas behind some methods for fluid dynamics for modeling ballistic trajectories, for example for splashes, dust, fire, or smoke [F]

GV/Visualization

- Describe the basic algorithms for scalar and vector visualization [F]
- Describe the tradeoffs of visualization algorithms in terms of accuracy and performance [A]
- Propose a suitable visualization design for a particular combination of data characteristics and application tasks [A]
- Analyze the effectiveness of a given visualization for a particular task [A]
- Design a process to evaluate the utility of a visualization algorithm or system [A]
- Recognize a variety of applications of visualization including representations of scientific, medical, and mathematical data; flow visualization; and spatial analysis [F]

The following outcomes will be evaluated via homework assignments and the course project

GV/Fundamental Concepts

- Construct a simple UI using a standard API. [U]

GV/Basic rendering

- Create a program to display 3D models of simple graphics images. [U]
- Derive linear perspective from similar triangles by converting points (x, y, z) to points (x/z, y/z, 1). [U]
- Obtain 2D and 3D points by applying affine transformations. [U]
- Apply 3D coordinate system and the changes required to extend 2D transformation operations to handle transformations in 3D. [U]

- Implement simple procedures that perform transformation and clipping operations on simple 2D images. [U]
- Implement a simple real-time render using a rasterization API (e.g., OpenGL) using vertex buffers and shaders. [U]

GV/Geometric Modeling

- Represent curves and surfaces using both implicit and parametric forms. [U]
- Create simple polyhedral models by surface tessellation. [U]
- Generate a mesh representation from an implicit surface. [U]
- Generate a fractal model or terrain using a procedural method. [U]
- Generate a mesh from data points acquired with a laser scanner. [U]
- Construct CSG models from simple primitives, such as cubes and quadric surfaces [U]

GV/Advanced Rendering

- Implement a non-trivial shading algorithm (e.g., toon shading, cascaded shadow maps) under a rasterization API. [U]
- Implement any of the specified graphics techniques using a primitive graphics system at the individual pixel level. [U]
- Implement a ray tracer for scenes using a simple (e.g., Phong model) BRDF plus reflection and refraction. [U]

GV/Computer Animation

- Compute the location and orientation of model parts using a forward kinematic approach. [U]
- Compute the orientation of articulated parts of a model from a location and orientation using an inverse kinematic approach. [U]
- Implement the spline interpolation method for producing in-between positions and orientations [U]
- Implement algorithms for physical modeling of particle dynamics using simple Newtonian mechanics, for example Witkin & Kass, snakes and worms, symplectic Euler, Stormer/Verlet, or midpoint Euler methods [U]
- Use common animation software to construct simple organic forms using metaball and skeleton [U]

Student Expectations

The above objectives cannot be met unless you, the student, take an active role in your education. Thus you are expected to:

- **Attend class** on a regular basis and devote your attention to the material presented
- Prepare for each and every class by **reading** the assigned material and completing both **pre- and post-lecture assignments**
- Devote the necessary **time** to preparing assignments and turning them in on time.
  - **Computer Science** is time-intensive
  - You should be prepared to give the time need for each assignment
  - As the class progresses the time required for assignments will increase, be prepared.

- **Time Management** is a requirement. Do not procrastinate, as the amount of time required for any given assignment and any given student cannot be estimated. For this reason you are encourage to begin assignments at the earliest possible date so that you will be able to complete them on time.

This is a 3 credit, as a student, you should expect to put forth on average 6 – 9 hours of additional effort outside the classroom towards this course. Given that, I expect to utilize this completely.

Valid Excuses

This course requires that you attend class and participate in classroom activities (including exams). The student handbook notes the following can be considered viable excuses for class absence, in consultation with your instructor:

- Serious Illness
- Severe Weather
- Religious Holidays
- Approved University Activities (e.g., extracurricular athletics, performance groups, student government)
- Emergency Family Issues

Invalid excuses include anything else, but I will specifically note the following:

- Minor Illness
- Typical Winter Weather Conditions
- Work
- Non-emergency Family Issues
- You or your family Purchased Plane Tickets (non-refundable)

Moodle

Course material including lectures, assignment requirements, handouts, and solutions can be viewed using your Moodle account. Announcement and Help forums will also be available on Moodle. Students are expected to access their Moodle account on a daily basis to keep apprised of course developments.

Attendance

Attendance in the course is mandatory. Missing 9 class meetings (without an acceptable excuse as defined by University Policy) during the course will result in failing the course, and a resulting X grade (IAW CoSE policy). Attendance will be monitored by in class activities, quizzes, etc.

Outreach

As this is a upper division Computer Science course, each student is required to complete 10 Hours of Computer Science Outreach within the community, or 15 Hours of unrelated Community Outreach. If you are enrolled in multiple upper division Computer Science courses you are required to only complete a total (across all courses) 10 or 15 hours. This will constitute a total of 10% of your grade in this course.

For more information please see: `https://www2.cose.isu.edu/~bodipaul/outreach/csServes/`.

Assignments

Homework assignments are due as assigned on Moodle. Do the homework, it helps. Late homework will be accepted up to the last day of the course, but each day after the due date will incur a linearly increasing penalty. This penalty will result in a 0 grade at midnight of the last day of class.

This course is based upon four components:

- Attendance at lectures and participation during in class exercises.
- Reading assignments
- Project
- Exams

Exams

You will be tested on your mastery of topics listed above as taken from lecture, readings, and assignments. Any information addressed by a component of this class will be considered for exams. There will be two exams, a mid-term and a final. The **final** will be a **comprehensive** exam.

Note: I do not give exams on any date other than the assigned date. The only exceptions to this are to accommodate exceptions noted above. These exceptions will only be accommodated if provided in advance and will require a minimum of **1 week advanced notice** unless an emergency.

Note: If you do not earned a **60% or higher average** on the exams in this course the highest grade you will receive in this course is a **60%** which equates to a **D−**

Grade Distribution

Grades for all assignments, exams, and the final grade will be assigned according to the following table:

Grade	–		+
A	90.00 – 92.99	93.00 – 100.0	
B	80.00 – 82.99	83.00 – 86.99	87.00 – 89.99
C	70.00 – 72.99	73.00 – 76.99	77.00 – 79.99

Grade	–		+
D	60.00 – 62.99	63.00 – 66.99	67.00 – 69.99
F		00.00 – 59.99	

The final grade calculation for this course will be allocated as follows:

CS 4458

Grade Event Type	Percent of Final Grade
Homework	30%
Project	35%
Exams	25%
Outreach	10%

CS 5558

Grade Event Type	Percent of Final Grade
Homework	40%
Project	35%
Exams	25%

Learning Environment

We are all committed to maintaining an inoffensive, non-threatening learning environment for every student. Class members (including the instructor) are thus to treat each other politely–both in word and deed. Offensive humor and aggressive personal advances are specifically forbidden. If you feel uncomfortable with a personal interaction in class, see your instructor for help in solving the problem.

Content

I reserve the right to change the content as needed to fit the flow of the class and experience of the students. Changes will be reflected on Moodle.

Policies & Procedures

Academic Integrity

Academic Integrity is expected at Idaho State University and the College of Science and Engineering. All forms of academic dishonesty, including cheating and plagiarism, are strictly prohibited, the penalties for which range up to permanent expulsion from the university with “Expulsion for Academic Dishonesty” noted on the student’s transcript.

Academic Integrity violations are a scourge at any University. I detest them with a passion. Anyone found to be violating the academic integrity code on any assignment or exam will be dealt with with extreme prejudice. The standard remedy, in this course, **for any Academic Integrity Violation will be FAILURE of the course**. Please note that in accordance with the Policy at Idaho State University that attempts to withdraw from the course to avoid such punishment will work to no avail.

Academic dishonesty includes, but is not limited to:

1. Cheating on Exams
2. Plagiarism
3. Collusion
4. Sharing solutions or code on programming assignments

Definitions

Cheating on an examination include:

- Copying from another’s exam, any means of communication with another during an exam, giving aid to or receiving aid from another during an exam;
- Using any material during an examination that is unauthorized by the proctor;
- Taking or attempting to take an exam for another student or allowing another student to take or attempt to take an exam for oneself
- Using, obtaining, or attempting to obtain by any means the whole or any part of an administered exam.
- Talking to anyone other than the professor during an exam.

Plagiarism is the unacknowledged incorporation of another student’s work into work which the student offers for credit.

- The use of the source code of another person’s program, even temporarily, is considered plagiarism. This includes attempting to hide the plagiarism by changing variable names, method names, or class names.
- Copying material from another project (including open source projects) without attributing (citing) that project.

Collusion is the unauthorized collaboration of another in preparing work that a student offers for credit.

- Allowing another person to use your source code, even temporarily, is considered collusion.

Other types of academic dishonesty include:

- Using other student’s content from their assignments, disk, etc.
- Performing any act designed to give unfair advantage to a student or the attempt to commit such acts

Exceptions

In this course, the specific exceptions given below are not considered scholastically dishonest acts:

- Discussion of the algorithm and general programming techniques used to solve a problem.
- Giving and receiving aid in debugging.
- Discussion and comparison of program output **(output only not code)**

Student Notification

All students are responsible for checking Moodle and their email on a regular basis, preferably daily, for notification of any class scheduling changes or assignment clarification. Often such notifications will be posted late at night.

Instructor Availability

The instructor will be available during posted office hours, but additional efforts are made to increase accessibility to the students. If the instructor is not available at the telephone number above, the student can leave a detailed voicemail message. However, the instructor's email is checked at a minimum of twice a day and often the student will get an immediate response to questions submitted by email. Email is usually the most reliable means of contact.

Note that I am a working research scientist, thus I may need to attend conferences both with my local colleagues and with my international colleagues. Thus, I am constantly in meetings, both here and abroad. That being said, I work 7 days a week 12 months a year. If you need my help and all other means of scheduling have led to no avail, please do not hesitate to contact me. Note that I am not an emergency service, I will require that you contact me 24 hours prior to an assignment being due, for which you need help. If you do not plan ahead, I cannot help.

Email Etiquette

Email is the best possible method of reaching me outside of my office hours. Note that I have certain expectations for communicating with me via email, as follows:

- DO NOT use chat or SMS shorthand in your messages.
- Use full words, and full sentences.
- Maintain the frame of mind that your are communicating to a professional, and that a professional demeanor is required.

Failure to abide by these requirements will result in your email being deleted and forgotten.

Disability and Special Needs

The Computer Science program at Idaho State University is committed to ensuring that all students achieve their potential. If you have a disability (physical, hearing, vision, psychiatric, or learning disability) that may need a reasonable accommodation, please contact the ADA & Disabilities Resource Center located in the Rendezvous Complex, Room 125, 282-3599, as early as possible.

Closed Week Policy

Information about the ISU Closed Week Policy can be found online. Note that the policy does not prevent the presentation of new material during closed week.

CoSE X Grade Policy

In the College of Science and Engineering, a student who earns a failing grade via course work (exams, homework, etc.) and has unexcused absences that total more than 30% of class meetings will receive a grade of “X”.