CSci 4611: Programming Interactive Computer Graphics and Games

Spring Semester 2018, 3 Credits
Course Website: https://canvas.umn.edu/courses/39610

Professor: Dr. Daniel Keefe
Email: dfk@umn.edu
Office: Keller Hall 6-211
Phone: 612-626-7508

TA: Jung Nam
Email: namxx054@umn.edu

TA: Dan Orban
Email: dtorban@umn.edu

Contact
Office Hours: See webpage for weekly prof and TA hours or email for appointment.

The best way to contact us is through the email alias: csci4611@cs.umn.edu, which will forward to all of us.

Important Dates
Lectures: 4:00pm – 5:15pm, Mon & Wed (Jan 17, 2018 – May 2, 2018), Keller Hall 3-230
Spring Break: Mar 12 – Mar 16
Final Exam: 4:00pm – 6:00pm, Monday, May 7

Course Description
Computer graphics is an exciting field within computer science that has seen dramatic recent growth. The impact of graphics on our culture and on our daily lives is far-reaching, as we can see through applications in art, design, education, games, movies, science, and medicine. This course covers the tools and techniques used today for programming games and other interactive computer graphics applications. Some of the core concepts covered include: event loops, rendering and animation, polygonal models, texturing, and physical simulation. This is a heavy programming course, and there is an emphasis on graphics toolkits and C/C++ programming. Other topics briefly covered include the history and future of computer games technologies and the social impact of interactive computer graphics.

Learning Outcomes
In this course, you will learn to:

- **Understand** basic concepts and algorithms relevant to computer graphics programming.
- **Identify, define, and solve** 2D and 3D graphics programming problems.
- **Critically evaluate** and select the right graphics toolkit to solve new problems.
- **Communicate** information through visual means using computers.

Prerequisites
The prerequisites for the course are CSci-2021. Please contact me if you have any questions about whether the course is a good fit for your interests and background.

Course Structure

Class Time
Class will meet twice each week. Class time will be a mix of lecture/discussion and hands-on activities. In general, one day of the week will be devoted to lecture/discussion and the other to developing hands-on graphics programming skills. On hands-on days, we will need laptops… at least enough for each pair of students to share a laptop, so please bring yours with you to class.

Class discussion and the lecture notes are the definitive references for both the assignments and the exams. (You are responsible for knowing the information presented and discussed in lectures.) The textbook is a very useful supplement, but the lectures draw upon several sources in addition to the textbook. Lecture slides will generally be available for download via the calendar section of the course webpage immediately after class on the day of the lecture.

In class activities will typically include some sort of hand-in (e.g., a google form "quiz" on the reading, solution to an in-class exercise). These will be the basis for your participation grade in the class also a means of checking attendance.

Exams
There will one midterm and one final exam for the course that will cover material presented in class as well as fundamental algorithms and techniques utilized in the assignments. You are required to take the exam at the official, course and university-scheduled time, except in the case of a University-excused absence (you must let me know ahead of time) or for documented emergency reasons. Dates and times for the exams will be announced ahead of time on the course webpage.

Assignments
The assignments for this class are challenging, and they are also the most exciting and rewarding aspect of the course (and your chance to impress your friends with some sweet 3D graphics). We will introduce this year's specific assignment topics on the first day of class. Briefly, the assignments will cover a broad range of applications of computer graphics programming, including: data visualization, character animation for movies and games, realistic real-time 3D rendering, art-inspired rendering, and more. You will use both 2D and 3D graphics programming toolkits to implement the assignments. Your goal in these assignments should be to come away with a great practical understanding of how to tackle new computer graphics programming challenges using the latest and greatest programming toolkits.

Assessment
Final course grades will be calculated based upon the following percentages:

- **50% -- Programming Assignments** (divided equally between each assignment)
- **45% -- Exams (Midterm and Final, weighted equally)**
- **5% -- Participation in Classroom and Web-based activities**

This course will follow the University's Uniform Grading and Transcription Policies, which are described on the web at
http://policy.umn.edu/Policies/Education/Education/GRADINGTRANSCRIPTS.html.

Calendar & Course Webpages

The course webpage is hosted on Canvas. All students registered for the course should already have access to it. We will make extensive use of this website, so it is a requirement of the course to check it regularly to stay on top of course announcements and assignments. A detailed calendar of lecture topics and assignment milestones/due dates is provided on the course webpage. As assignment dates and lecture topics may change slightly, I expect you to check the webpage regularly for updates. The webpage has an "announcements" section, which we will use to post official notices. In the past, this has been useful to clarify assignment details if any changes are necessary after an assignment has been handed out, so check the webpage regularly as you work on your assignments. One of the first announcements posted will be the time and location of our regularly scheduled office hours. This will also be the place that we will post an announcement if for any reason the normal office hours need to be cancelled or rescheduled.

We will often use the online forum or similar features within the class webpage. The purpose of these forums is to provide a place for the TA and I to respond quickly to general questions that arise during the semester, particularly as you work on the programming assignments. When posting a question to the forum, it is essential that you keep in mind the public nature of the communication. In particular, the forum is not a place to discuss a grading decision or a personal concern, contact me directly with these questions. Rather, use the forum to ask general questions that apply to the entire class (e.g. a clarification about an assignment, a technical question on C++ programming, a question about a topic discussed in class). The assignments are all individual assignments, so it is also important to not distribute part of your solution to the class when posting or responding to a post of the forum.

Readings
Weekly readings will come from a main technical text and also from complementary materials that focus on contemporary topics in game design. An estimate of 20-30 pages of reading per week is expected. The required text for the course is:

3D Graphics for Game Programming, by JungHyun Han, CRC Press, 2011.

The text is available at the university bookstore. We will post additional links to references on the course webpage.

Late Policy
A 10% per day (24 hour period from the due time) late penalty will be applied to programming assignments turned in late. No assignment that is more than five days late will be accepted. Smaller class and web-based activities included in your participation grade will not be accepted late, but the lowest 2 grades in this category will automatically be dropped in anticipation of each student requiring a couple of days for a legitimate absence during the course of the semester. Exams must be taken at the scheduled times.

Use of Web-Based Resources and Public Presentations of Student Work
In this class, our use of technology will make students' names, U of M Internet IDs, and/or coursework (e.g., images, videos, and descriptions of the computer graphics projects you produce) visible within course-related websites, including the main course website. Similarly, after the class is complete, we would like to use some of the best visuals (images, video, etc.) produced during the class on our websites and in other public presentations to drum up excitement for future students and demonstrate to others what a great job our students at the U of M do. Some of these websites are not secure and can be accessed by anyone. If you have

concerns about the visibility of your name, Internet ID, or images and descriptions of your coursework appearing in any of these public forms, please contact me for further information.

Course Topics in Detail

(see the course calendar for most up-to-date information)

Planned Week-by-Week Topics

Week 1: First hands-on graphics programing
- Use of a lightweight graphics scripting toolkit, such as *Processing.org*
- Graphics primitives: points, lines, polygons
- Representing color in computer graphics
- Responding to user input

Week 2: The History and Future of Computer Graphics and Games
- Ivan Sutherland and the history of computer graphics
- Early games
- Current trends in graphics hardware
- The future of real-time graphics and futuristic human-computer interfaces

Week 3: Intro to a modern C++ based graphics toolkit
- Practical introduction to programming with a graphics toolkit used in games or related industries
- Hands on experience

Week 4: Visual Debugging with Graphics Toolkits
- Software engineering concepts and tools for computer graphics
- Emphasis on using visual outputs to understand the function of programs

Week 5: Graphics Math in More Detail / Linear Algebra Refresher
- Refresher on transformation matrices
- Advanced graphics math at the toolkit level (e.g., ray-triangle intersection routines, object vs. world space)

Week 6: Polygonal Modeling and Scene Graphs (Using C++ Toolkit)
- Mesh and spatial data structures
- Scene graphs and hierarchical transformations

Week 7: Creating Effective Virtual Worlds
- Schell's elemental triad for effective game design
- The relationship between characters, scenes, and worlds
- Automated terrain generation and other technical tools for building worlds

Week 8: Realism in Interactive Computer Graphics
- Tradeoffs between speed and realism
- Current trends in industry and real-world applications
- Serious games
- Intro to part 2 of the course viewed as many forms of realism (texture, animation, physics, user experience)

Week 9: Texture and Bump Mapping for Realism (Using C++ Toolkit)
- Texture coordinates and different forms of texture mapping
- Impact of speed and realism
- Artistic use of texturing / contemporary texturing in the games

Week 10: Characters and Animation
- Simulation and animation loops/threads
- Motion capture vs. physically-based simulation vs. key-frame animation
- Developing effective characters

Week 11: Lighting Design and Implementation
- Local vs. global illumination
- OpenGL shaders

Week 12: Designing for the User Experience in Games and Interactive Graphics
- Interdisciplinary design practices in game development
- Interface genres and input available during game play
- Examples from outside of games: virtual reality, CAD tools, 3D modeling tools

Week 13: Event Loops and Graphical User Interaction
- Implementing effective user interfaces with 3D graphics toolkits

Week 14: Physics Engines and Real-Time Simulation
- Integrating 2D and 3D physics toolkits with graphics toolkits
- How to manage your loops: rendering, physics, events, networking, etc.

Week 15: The Social Impact of Interactive Computer Graphics
- Games and graphics in our culture
- Games for healthcare
- Online and multi-player games
- The cognitive psychology of avatars
- Anthropological Examples, e.g., *Coming of Age in Second Life*

Planned Assignments (2 weeks each)

Assignment 1: Use *Processing.org* toolkit to create an interactive art installation similar to Text Rain by Camille Utterback & Romy Achituv,1999.

Assignment 2: Car soccer (animation, simulation, and collision detection) in a modern graphics toolkit.

Assignment 3: Rendering and navigating the Earth (polygonal modeling of terrain, texture mapping, camera controls).

Assignment 4: Animating characters using motion capture data (hierarchical transformations, data-driven animation, realism).

Assignment 5: Artistic rendering for non-photorealistic stylistic graphics with shaders.

Assignment 6: Pencil physics – implement a game similar to crayon physics (www.crayonphysics.com). Combines several skills / areas of study throughout the semester: (1) geometric modeling, (2) shaders for non-photorealistic rendering, (3) physics-based simulation, and (3) a sketch-based gestural user interface.

Academic Integrity

Collaboration Policy
All work submitted for this course is required to be your original work. You are expected to do your own thinking, your own design, and your own coding. You are encouraged to discuss the content of the lectures and the texts with your peers. With respect to programming assignments, you are also permitted to discuss (and post to the forum regarding) programming in C++ in general (e.g., a syntax error you are stuck on, missing include file). However, your discussions with others must stop before discussing a solution to the homework or assignment. If you have any question about whether discussing something with peers might go beyond what is permitted, then stop and ask us first for clarification on the policy.

Use of Web & External Resources
The web will be one of your best learning tools and sources of documentation for working with the processing.org and OpenGL graphics toolkits used in the class, but you are not allowed to use web or other external resources to find solutions to the core computer graphics problems in your assignments.

What you can do: You are expected/encouraged to use the web to lookup questions about the APIs used and even to look at example programs written with processing.org and OpenGL.

What you cannot do: You are not allowed to search for solutions to assignments. For example, if you assignment is to develop your own texture mapping routine for a sphere, you can lookup the OpenGL or processing.org API documentation to learn the right function to call and the right syntax to use to set the texture coordinate for each vertex in a mesh. What you cannot do is google for “texture map a sphere” and find out how to setup a for loop and use cos() and sin() equations to determine the correct texture coordinates for each vertex of the sphere. This is the difference between using the web as a programming toolkit reference vs. using the web as a reference for the computer graphics algorithms/solutions that are being taught in the course. With the exception of the small snippets of code that are found in the official processing and OpenGL documentation, you should never copy code like a routine, class, object, etc. from a website and include it in your project. Never post questions about your assignment or portions of the code that you have written or that we have given you for the class to an online forum other than the class forum on our class webpage.

Scholastic Conduct
Scholastic dishonesty includes violating the course policies outlined here; plagiarizing; cheating on assignments or examinations; engaging in unauthorized collaboration on academic work; taking, acquiring, or using test materials without faculty permission; submitting false or incomplete records of academic achievement; acting alone or in cooperation with another to falsify records or to obtain dishonestly grades, honors, awards, or professional endorsement; altering, forging, or misusing a University academic record; or fabricating or falsifying data, research procedures, or data analysis. Within the course, a student responsible for scholastic dishonesty can be given a penalty, including an "F" or "N" for the course, and further disciplinary

action may occur. See the University of Minnesota conduct code: http://www1.umn.edu/regents/policies/academic/Conduct.html.

Additional Information

Disability Information
University policy is to provide, on a flexible and individualized basis, reasonable accommodations to students who have documented disability conditions (e.g., physical, learning, psychiatric, vision, hearing, or systemic) that may affect their ability to participate in course activities or to meet course requirements. Students with disabilities are encouraged to contact Disability Services and their instructors to discuss individual needs for accommodations. Disability Services, McNamara Alumni Center, Suite 180, 200 Oak Street, East Bank. Staff can be reached at http://ds.umn.edu or by calling (612) 626-1333 (voice or TTY).

Mental Health Information
As a student you may experience a range of issues that can cause barriers to learning, such as strained relationships, increased anxiety, alcohol/drug problems, feeling down, difficulty concentrating and/or lack of motivation. These mental health concerns or stressful events may lead to diminished academic performance or reduce your ability to participate in daily activities. University of Minnesota services are available to assist you with addressing these and other concerns you may be experiencing. You can learn more about the broad range of confidential mental health services available on campus via http://www.mentalhealth.umn.edu.

Equal Access and Opportunity
The University of Minnesota shall provide equal access to and opportunity in its programs, facilities, and employment without regard to race, color, creed, religion, national origin, gender, age, marital status, disability, public assistance status, veteran status, sexual orientation, gender identity, or gender expression.

Sexual Harassment
University policy prohibits sexual harassment as defined in the University Policy Statement adopted on December 11, 1998. Complaints about sexual harassment should be reported to the University Office of Equal Opportunity, 419 Morrill Hall, East Bank.
http://www1.umn.edu/regents/policies/humanresources/SexHarassment.html