

Game Engine I

Fall 2015

Gam 475

instructor: Ed Keenan
email: ekeenan2@cdm.depaul.edu
office hours: Wed 1-4pm 14EAS 505, 30 min before class or by email appointment
office: CDM 830
phone: (312)362-6747
website: piazza.com/depaul/fall2015/gam475 (Preferred communication)
lecture: 14EAS 505, Thursday 5:45-9:00pm
Desired to Learn (D2L): d2l.depaul.edu (Grades, Viewing lectures, Announcements)
Version Control: perforce: **140.192.39.61:1666**

Description:

Game engines provide an integrated programming platform for today's high performance games. This class explores C/C++ game engine programming, data structures, and practices. Topics include memory management, input devices, video rendering, file access, debugging, and application interface development.

Prerequisites:

- Required:
 - CSC 461: Optimized C++
 - Implies - C++, CSC 374 Computer Systems II, CSC 383 or CSC 393 data structures
- Recommended:
 - GPH 469: Computer Graphics Development
 - OpenGL or Direct X graphics programming
 - Exposure to 3D matrix math and basic polygonal graphics understanding

Learning Goals:

- Understand the components of a real-time Game Engine
 - Input, AI, Physics, Networking, UI, Graphics
- Understand real-time operating systems influence Game Engine performance
 - Such as Memory, File, Object, Math systems
- Understand real-time Graphics rendering
 - Transformation, Rendering, Camera, Polygons, Buffers
- Developing a large software architecture
 - Using Design patterns, Layers, and Libraries
- Developing and maintaining a very large project
 - Incremental development and debugging a 100K+ lines of code

Grading

50% - Milestone 1 Core Systems

- 10% - PA1 - PCS Tree (Object System)
- 10% - PA2 - Memory System part A: Heaps/tracking
- 10% - PA3 - Memory System part B: Alignment/fixed block
- 10% - PA4 - Math System
- 10% - PA5 - File System / PCSTree iterators

30% - Milestone 2 Graphics System

10% - Progression submissions (4 weekly submissions with video)

10% - Final Exam

Textbooks and printed resources

Additional course material will be many supplied through class notes, handouts or online links.

- 2 Required Books
 - Game Engine Architecture, 2nd edition, Gregory, A.K. Peters Ltd., 2015
 - ISBN: 978-1466560017
 - OpenGL® SuperBible: Comprehensive Tutorial and Reference 6th or 7th Edition
 - OpenGL® SuperBible: Comprehensive Tutorial and Reference, 6th edition, Wright, Lipchak, Sellers & Haemel, Addison-Wesley Prof./Pearson, 2014.
 - ISBN: 978-0321902948
 - 7th edition just released in August 2015
 - We are using 6th edition for this class
 - Uses a OpenGL 4.3, this new edition uses OpenGL 4.5 with new extensions
 - Either is OK for this class
- Recommended:
 - **Windows System Programming** (4th Edition), 2010, Johnson Hart,
 - ISBN: 978-0321657749
- Assumed you already have (please buy it if you don't have a copy):
 - ***The C++ Programming Language***- Bjarne Stroustrup

Additional Material

- Will be provided by the instructor
- Lectures, links, SDKs and other corresponding material

Software

- **Microsoft Visual Studio 2013 (ultimate edition - Recommended)**
 - [DePaul MSDNAA link Microsoft Visual Studio Ultimate 2013](#)
 - C++ and C# install (future classes)
 - Microsoft Visual Studio 2015 is not used in this class.
- Perforce - Visual Client (p4v)
 - www.perforce.com
- Download and configuration instructions will be provided in class
 - Server address: **140.192.39.61:1666**

Topics will include:

Core Systems

- Memory System
- File System
- Object System
- Math System

Primitive Graphics Systems

- Transformations
- Camera
- Polygons / Strips
- Vertex Buffers
- Scene Graph
 - Bounding Volumes
 - Hierarchy of scene
 - Level of Detail

Programming Assignments – 90%

Two distinctive milestones to this class. In the first milestone section, Core systems - we build individual components and libraries for the memory, file, object system, and math systems that will be used in the Graphics Engine. In the second milestone, we build a Graphics Engine from scratch, using the most primitives components on top of OpenGL. All the work complete in this class is further extended in Gam 575 Game Engine II.

- **Milestone 1:** Core System - 50%
 - Composed of the following core systems:
 - 10% - PCS Tree (Object System)
 - 10% - Memory System part A: Heaps/tracking
 - 10% - Memory System part B: Alignment/fixed block
 - 10% - Math System
 - 10% - File System / PCSTree iterators

- **Milestone 2:** Graphics Engine - 40%
 - Graphics Engine: 30%
 - Composed of the following base systems:
 - Transformations
 - Camera
 - Polygons / Strips
 - Vertex Buffers
 - Texture
 - Model management
 - Composed of the following secondary systems:
 - Bounding Volumes
 - Hierarchy of scene
 - Level of Detail
 - Demos using your new engine
 - Progression submissions: 10%
 - 4 progression submissions (1 weekly)
 - Showcase incremental development
 - Video and source drops weekly

Final Exam – 10%

- Final exam covering the concepts and the material of the class (Take home exam).
 - Open book, Open notes, Open computer – Yes, I said it.
 - Essay format 5+ pages
 - Post-mortem of your design process and understanding of engine development

Perforce Submissions

- Everyone is expected to submit at least 10 submissions a week to perforce.
- The biggest reason students get into trouble with software design:
 - Not working on the material frequently enough
 - Taking too large of a bite of the design
- Both are fixed with this Perforce RULE
- Even my simplest programs take 10-20 submissions.
 - For these project assignments my average is 40-400 submissions, so 10 will be no problem.
- Detailed perforce changelist comments are expected

Piazza Discussion forum

- Statistics show: students who participate more and help other students do better!
 - The correlation is ridiculous!
 - Poor understanding / poor participation.
 - Great understanding / Great participation

- As you master the material, help others learn!
 - You're in the master's program so master it!
- Everyone is **expected** and encouraged to participate on the Piazza discussion forum. All class-related discussion here this term.
- The quicker you begin asking questions on Piazza (rather than via emails), the quicker you'll benefit from the collective knowledge of your classmates and instructors. I encourage you to ask questions when you're struggling to understand a concept.
- All correspondence that is not personal in nature should be vectored through Piazza
- Sensitive material, use Piazza private note, not email.
- Keep the forum professional and positive, help each other out.
 - Karma really pays off here.
 - Help each other whenever you can.
 - There will be a section where you'll need help (trust me).

NOTE: Do **NOT** post until you have watched the entire lecture **FIRST** (in class or online)
This will prevent frustration on all sides (members asking or answering questions)

Collaborating together on programming assignments

- You are encourage to work together
 - Use the Piazza forums heavy
 - Even share your material with others in the common directory
- Everyone is 100% responsible for the work they do.
 - If you get help with a section of code,
 - Please refactor the code the **snot out of it**
 - Comment and understand that material
 - Transform the code to **make it yours**
 - Be able to answer **any** question regarding the code you commit
- System for Detecting Software Plagiarism
 - We will be using MOSS - Measure of Software Similarity (Stanford University)
 - Indicates possible code infringements (plagiarism)
 - MOSS - will detect the similarity independent of naming convention, indentation style or formatting, it compares abstract syntax tree of your code.
- If you gain significant support / help from another student
 - Fully disclose the support / help you had in a Readme.txt file submitted with your assignments.
 - Disclosing the help, is **not permission** for copying the code.
 - Only there to clarify and acknowledge help you were given from a fellow student.
- Modifying any Unit Test to alter the outcome results is also an **Academic Integrity Violation**
- If you are stuck and find yourself even tempted to plagiarize
 - Ask for help !!!!
 - Use on Piazza -> Visit during offices hours, make an appointment
 - **Don't ever compromise your integrity!**

Tentative Class Schedule

Date	Lecture	Activity	Due
Week 1 10-Sep	Overview Architecture Design Object System	PA1 - Object system	
Week 2 17-Sep	Memory Overview Memory System	PA2 - Memory system Part A: Heaps/Track	PA1 - Object system
Week 3 24-Sep	Fixed Block Memory Data Alignment	PA3 - Memory system Part B: Align/Fixed Block	PA2 - Memory system Part A: Heaps/Track
Week 4 1-Oct	Vector / Matrix Transformations Math Library	PA4 - Math system	PA3 - Memory system Part B: Align/Fixed Block
Week 5 8-Oct	File System Load in Place Forward/Reverse Iterators	PA5 - File / PCSTree Iterators	PA4 - Math system
Week 6 15-Oct	Graphics Overview Refactoring Math Lib integration	Demos from SuperBible Refactor Spinning cube	PA5 - File / PCSTree Iterators
Week 7 22-Oct	Game Loop Shader Manager Model Manager	Different models Different shaders	Graphics Progress #1
Week 8 29-Oct	VAO/VBO Graphics Object Texture Manager	Rework models Add new model Changing textures	Graphics Progress #2
Week 9 5-Nov	Camera Manager Culling Bounding Volumes Scene Graph	Camera movement Culling	Graphics Progress #3
Week 10 12-Nov	Loading from a File Attribute switching Camera transitions	Loading models from file Demo tweaking	Graphics Progress #4
Week 11 19-Nov	Final Exam (take home)		Milestone II - due Final Exam due

September 15, 2015 Last day to add classes to AQ2015 schedule

September 22, 2015 Last day to drop classes with no penalty, Last day to select pass/fail option

September 23, 2015 Grades of "W" assigned for AQ2015 classes dropped on or after this day

September 29, 2015 Last day to select auditor status

October 27, 2015 Last day to withdraw from AQ2015 classes

Course Policies

Changes to Syllabus

This syllabus is subject to change as necessary during the quarter. If a change occurs, it will be thoroughly addressed during class, posted under Announcements in D2L and sent via email.

Online Course Evaluations

Evaluations are a way for students to provide valuable feedback regarding their instructor and the course. Detailed feedback will enable the instructor to continuously tailor teaching methods and course content to meet the learning goals of the course and the academic needs of the students. They are a requirement of the course and are key to continue to provide you with the highest quality of teaching. The evaluations are anonymous; the instructor and administration do not track who entered what responses. A program is used to check if the student completed the evaluations, but the evaluation is completely separate from the student's identity. Since 100% participation is our goal, students are sent periodic reminders over three weeks. Students do not receive reminders once they complete the evaluation. Students complete the evaluation online in [CampusConnect](#).

Academic Integrity and Plagiarism

This course will be subject to the university's academic integrity policy. More information can be found at <http://academicintegrity.depaul.edu/>. If you have any questions be sure to consult with your professor.

Academic Policies

All students are required to manage their class schedules each term in accordance with the deadlines for enrolling and withdrawing as indicated in the [University Academic Calendar](#). Information on enrollment, withdrawal, grading and incompletes can be found at: cdm.depaul.edu/enrollment.

Students with Disabilities

Students who feel they may need an accommodation based on the impact of a disability should contact the instructor privately to discuss their specific needs. All discussions will remain confidential.

To ensure that you receive the most appropriate accommodation based on your needs, contact the instructor as early as possible in the quarter (preferably within the first week of class), and make sure that you have contacted the Center for Students with Disabilities (CSD) at: csd@depaul.edu.

Lewis Center 1420, 25 East Jackson Blvd.

Phone number: (312)362-8002

Fax: (312)362-6544

TTY: (773)325.7296

Retroactive withdrawal

This policy exists to assist students for whom extenuating circumstances prevented them from meeting the withdrawal deadline. During their college career students may be allowed one medical/personal administrative withdrawal and one college office administrative withdrawal, each for one or more courses in a single term. Repeated requests will not be considered. Submitting an appeal for retroactive withdrawal does not guarantee approval. Information on enrollment, withdrawal, grading and incompletes can be found at:

<http://www.cdm.depaul.edu/Enrollment-Policies.aspx>