Game Engine I

Fall 2016

Gam 475

instructor:	Ed Keenan
email:	ekeenan2@cdm.depaul.edu
office hours:	Wed/Thur 3:30-5pm, 9-9:30pm in classroom or by email appointment
office:	CDM 830 (see Piazza for office hour location)
phone:	(312)362-6747
website:	piazza.com/depaul/fall2016/gam475 (Preferred communication)
lecture:	CDM 220, 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

40% - Milestone 2 Graphics System

- 35% - Required features (Graphics, Shaders, Camera, Textures, Lighting, Scene Graph,…)
- 5%   - 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*** 7th Edition, Sellers, Wright, & Haemel, Addison-Wesley Professional; 7th edition (July 31, 2015)
      - ISBN: 978-0672337475
    - 6th or 7th edition is ok for this class
      - 6th uses OpenGL 4.3, 7th edition uses OpenGL 4.5
- 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 2015 Enterprise Edition***
  - Visual Studio Enterprise 2015 with update 3 32/64-bit
  - 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: 35%
        - 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: 5%
  - 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

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!
    - Want to be a Master programmer so master it!
- Everyone is ***expected*** and encouraged to participate on the Piazza discussion forum. All class-related discussion here this term.
  - At least one real question or response per week from every student.
- Everyone is expected to keep up with the material on Piazza and are responsible for its content. Critical class updates and directions will be presented there.
  - Not participating or reading the material on Piazza is not an excuse.
- All correspondence that is not personal in nature should be vectored through Piazza
  - Sensitive material, use Piazza private note, not email.
- 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.
- 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)

Perforce Submissions

- Everyone is expected to submit several submissions to perforce a week.
    - Minimum 5 significant (real) submissions on 3 separate days.
    - To promote incremental development and prevent last day rush.
    - Grade deduction will occur if not followed
- The biggest reason students get into trouble with software design:
    - Not starting the project early
    - Not working on the material frequently enough
    - Taking too large of a bite of the design
- Both are minimized with this *Perforce RULE*
- Even my simplest programs take 10-20 submissions.
    - For these project assignments my average is 40-400 submissions, so 5 will be no problem.
- Detailed perforce changelist comments are expected

Collaborating together on programming assignments

- You are encouraged to work together
    - Use the Piazza forums heavy
    - Even share your material with others in the common directory
        - Obviously not the answers
- 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.
    - I will pursue any plagiarism/integrity violations aggressively, arguing for full expulsion from the university for the offenders.
        - Don't put me or you in this scenario
- 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!***

- Material was uniquely created for this Class.
    - You indirectly by the process of tuition, "paid" for the contents and material of this class.
        - Do not share this ***copyrighted*** material in any form
        - It is design for your personal use, while enrolled in the Class.
    - Do ***NOT*** post any content or revealing material to any external website or forum outside of this class.
        - The Class Piazza forum is provided for this service, ask questions there, not on the internet (i.e. StackOverflow and other software forums)
- After you leave this class
    - You are expressly ***FORBIDDEN*** to provide or share the content with others.
    - Academic Integrity Violations can still be applied to students who provide material support to other students even after completion of the class.
- Just follow the golden rule:
    - ***"I have neither given, nor received, nor have I tolerated others' use of unauthorized aid."***

Tentative Class Schedule

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

September 13, 2016	Last day to add classes to AQ2016 schedule
September 20, 2016	Last day to drop classes with no penalty, Last day to select pass/fail option
September 21, 2016	Grades of “W” assigned for SQ2016 classes dropped on or after this day
September 27, 2016	Last day to select auditor status
October 25, 2016	Last day to withdraw from AQ2016 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