Game Programming II (Wrath of Khan)

Winter 2019

GAM 575

instructor:	Ed Keenan
email:	ekeenan2@cdm.depaul.edu
office hours:	Wed 1:30-3:00pm CDM 400, 9-10 Classroom or by email appointment
office:	CDM 830
phone:	(312)362-6747
website:	piazza.com/depaul/winter2018/gam575 (Preferred communication)
lecture:	CDM 206, 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 Engine Programming II (Wrath of Khan) is being offered this spring. This class continues to explore C/C++ game engine programming, data structures, and practices. Topics include audio, network access, threads and multi-processor systems, profiling, scripting, content libraries, animation, and a survey of game engines. The previous quarter's game engine will be furthered strengthen with more systems integrated into our framework.

Prerequisites:

- GAM 475 Game Programming Engine I

Learning Goals:

- Students should be able to design and implement a 3D keyframe animation system.
- Students should be able to create asset conversion tools for 3D models and animations.
- Students be able to use 3D Math (Matrices, Vectors, Quaternions) to solve Graphics and simulations problems
- Students should be able to implement an efficient graphics rendering system using data friendly buffers, such as Vertex Buffer Objects.
- Students should understand the architecture and layout of a real-time game engine
- Students will be able to design and implement a large software system

Grading

Milestone 1

- 10 % PA1 - Archiver
- 10% PA2 - Math
- 15% PA3 - Converter for models
- 15% PA4 - Model Viewer

Milestone 2

- 20% PA5 - Converter for animation and bones
- 20% PA6 - Skeletal Animation Engine
- 10% PA7 - Playback engine

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 edition,Wright, Lipchak, Sellers & Haemel, Addison-Wesley Prof./Pearson, 2014.
    - ISBN: 978-0321902948
- 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 2017 Enterprise Edition (not Community)***
  - MSDNAA Depaul – Visual Studio 2017 Enterprise
    - C++ and C# install (future classes)
  - Any other variants are not used in this class
  - Students are responsible keeping their development tools working
- ***Perforce Server***
  - Download and configuration instructions will be provided in class
  - ***Perforce – Helix Visual Client (p4v)* -** www.perforce.com
    - Server address: ***140.192.39.61:1666***

Topics will include:

- ***Main Lectures:***
    - 2nd pass on Graphics engine
        - Run time formats
        - Texture manager
        - Camera Culling
        - Bounding volumes
    - FBX
        - Model conversion
        - Animation extraction
        - Skeleton
        - Skinning
    - Transformations
        - Interpolation - Linear, slerp, blerp
        - Hierarchy relative vs flat transformations
        - Quaternions
    - Animation
        - Key frame vs motion capture
        - Animation Controllers
        - Skeletons
        - Blending / Mixing
        - Morphing
        - Move the animation by game control
        - Puppet-ting
    - Skinning
        - Rigid body
        - EA technique
        - Midway Technique
    - Object
        - Cloning
        - Replication
        - Scene Graph

- ***Secondary Lectures (if time permits)***
    - Level of Detail
    - Multiple rendering targets
    - Input / Events trigger
    - Sound System
    - UI
    - Threading

Programming Assignments – 100%

2 major milestones, each milestone builds on the previous milestone. There are mini check points to make sure the student is on convergence path between milestones.

- ***Milestone 1***: Model Converter and more… - 50% (due Week 6)
    - Write a generic file Archiver
        - Takes loose binary files, adds headers and formatting info to create chunks
        - Packages the chunks together into a single binary package
        - Extracts chunks on demand from the package
    - Write a FBX converter to convert to run-time file format.
        - Place data into the file archiver
        - VBO with VAO format
        - Convert 3 FBX models to Game Engine runtime format from a batch/script file.
        - Models need to be
            - In FBX format
            - Exceeding 200 polygons each
            - Contain textures / Models need to be lit in game engine
    - Model viewer
        - Load models from the archiver
        - Rotate camera 360 around the viewer
        - Zoom
        - Add ground plane
        - Bounding volume culling and viewing
        - Demo the game engine to display the 3 supplied models + 3 others
    - Quaternion library
        - Write and integrate the quaternions into the math library
        - Validate against unit tests
- ***Milestone 2***: Animation Engine 50% (due Week 11)
    - FBX animation exporter
        - Write a converter to extract Animation data (Skeleton and animation) from an fbx file
        - Convert 3 FBX animations to a run-time file format
    - Animation engine
        - Demo the game engine to display the 3 different animations
        - Be able to dynamically interpolate the play back of the animation
        - Each animation needs at least 5 or more bones
        - At least 20 or more key frames
            - Each keyframe containing rotation and translation
    - Playback engine
        - Animation should be able to:
            - Play forward / backwards
            - Loop

            - Faster or slower playback rate
        - Transition to different animations

Perforce Submissions

- Everyone is expected to submit several <u>submissions</u> to perforce a week.
    - Minimum of ***five*** significant (real) submissions on ***three*** 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(byte) of the design
- Both are minimized with this <u>*Perforce RULE*</u>
- Even my simplest programs take 10-20 submissions.
    - For these project assignments, my average is 40-400 submissions, so five will be no problem.
- Detailed perforce changelist comments are expected

Piazza Discussion forum

- Statistics show: students who participate more and help other students <u>do better</u>!
    - 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 <u>***expected***</u> 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 will benefit from the collective knowledge of your classmates and instructors. I encourage you to ask questions when you are 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 time when you will need help from the class (trust me).

NOTE: Do <u>***NOT***</u> post until you have watched the entire lecture <u>***FIRST***</u> (in class or online)

This will prevent frustration on all sides (members asking or answering questions)

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 or website
    - 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 or Project setting 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.
    - By the process of tuition, you "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."***

Miscellaneous

- ***Late Policies***
  - Due dates and times are verified by the submission record on the Perforce Server
    - No extensions are allowed
  - All assignments need to be compiling without warnings
    - Failure to compile "as-is" results in a 0 for the grade

- ***Crashing***
  - For assignments that work for a set duration (long enough to demo all the features) but then crash after time.
    - A deduction of 20% is applied to the grade of that assignment
    - Crash – program locking up or quitting unexpectedly

- ***Memory Leaking***
  - For assignments that have memory tracking enabled
    - If an assignment is determined that its leaking memory
      - A deduction of 20% is applied to the grade of that assignment
  - Leaking status is provided during development

Tentative Schedule:

Week	Lecture	Assign	Due
1	Syllabus, Class Overview	PA1 - Archiver	
2	Quaternions Keenan-Mods Quaternions Quaternions Interface	PA2 - Math Quaternions	PA1 – Archiver
3	FBX Converter Level of Details	PA3 - FBX converter *Models*	PA2 – Math
4	Converter Hands on demo Model Viewer	PA4 - Model Viewer	PA3 – FBX Converter
5	Keyframe Animation Starting Animation development demo		PA4 – Model Viewer
6	Timer (Discrete time for animations) Animation System	Animation Engine start	Milestone 1 PA1 - PA4
7	Keyframe animation Animation converter Design demos	PA5 - FBX converter *Anim data* *Skeleton*	
8	Hierarchy Animation Skeletons Design demos	PA6 - Animation System	PA5 - FBX converter
9	Skinning Pimpl	PA7 - Playback Instance	PA6 - Animation System
10	Animation System issues Loading / unloading assess		PA7 -Playback
11			Milestone 2 PA5-PA7

University Dates (Drop, Withdrawal, Audit, Exam)

- https://academics.depaul.edu/calendar/Pages/default.aspx

University 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. Please see https://resources.depaul.edu/teaching-commons/teaching/Pages/online-teaching-evaluations.aspx for additional information.

Academic Integrity and Plagiarism

This course will be subject to the university's academic integrity policy. More information can be found at https://resources.depaul.edu/teaching-commons/teaching/academic-integrity/Pages/default.aspx.

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:
http://www.cdm.depaul.edu/Current%20Students/Pages/PoliciesandProcedures.aspx

Incomplete Grades

An incomplete grade is a special, temporary grade that may be assigned by an instructor when unforeseeable circumstances prevent a student from completing course requirements by the end of the term and when otherwise the student had a record of satisfactory progress in the course. All incomplete requests must be approved by the instructor of the course and a CDM Associate Dean. Only exceptions cases will receive such approval. Information about the Incomplete Grades policy can be found at
http://www.cdm.depaul.edu/Current%20Students/Pages/Grading-Policies.aspx

Students with Disabilities

Students seeking disability-related accommodations are required to register with DePaul's Center for Students with Disabilities (CSD) enabling them to access accommodations and support services to assist with their success. There are two office locations:

- Loop Campus – Lewis Center #1420 – (312) 362-8002
- Lincoln Park Campus – Student Center #370 – (773) 325-1677

Students who register with the Center for Students with Disabilities are also invited to contact Dr. Gergory Moorhead, Director of the Center, privately to discuss how he may assist in facilitating the accommodations to be used in a course. This is best done early in the term. The conversation will remain confidential to the extent possible.

Please see https://offices.depaul.edu/student-affairs/about/departments/Pages/csd.aspx for Services and Contact Information.

Proctored exams for OL courses (if applicable)

If you are an online learning student living in the Chicagoland area (within 30 miles of Chicago), you will need to come to the Loop campus to take an exam. Online learning students outside of the Chicagoland area are required to locate a proctor at a local library, college or university. You will need to take the exam within the window your instructor gives. Students should examine the course syllabus to find exam dates and the instructor's policy on make-up exams. Detailed information on proctored exams for online learning students can be found at http://www.cdm.depaul.edu/onlinelearning/Pages/Exams.aspx

Online office hours for OL courses (if applicable)

Faculty should be accessible to online students via phone, email and/or Skype.