# Game Programming II (Wrath of Khan)

Spring 2015

GAM 575

| | |
|---|---|
| **instructor:** | Ed Keenan |
| **email:** | ekeenan2@cdm.depaul.edu |
| **office hours:** | R  4-5pm, 9-10pm after class or by email appointment |
| **office:** | CDM 830 |
| **phone:** | (312)362-6747 |
| **website:** | piazza.com/depaul/spring2015/gam575  (Preferred communication) |
| **lecture:** | CDM 200, Thursdays, 5:45-9:00pm |
| **Course OnLine (COL):** | col.cdm.depaul.edu |

## 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 (Matrixs, 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

80% - 2 milestones (grouping of assignments)

10% - Perforce Submissions (at least 10 per week with comments)

10% - Final Exam

## Textbooks and printed resources

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

- 2 Required Books
  - o Game Engine Architecture, 2nd edition, Gregory, A.K. Peters Ltd., 2015
    - ▪ ISBN: 978-1466560017
  - o OpenGL® SuperBible: Comprehensive Tutorial and Reference, 6th edition,Wright, Lipchak, Sellers & Haemel, Addison-Wesley Prof./Pearson, 2014.
    - ▪ ISBN: 978-0321902948
- Recommended:
  - o **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):
  - o ***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 2012 (ultimate edition - Recommended)***
  - o [DePaul MSDNAA link Microsoft_Visual_Studio_Ultimate_2012](#)
  - o C++ and C# install (future classes)
  - o Microsoft Visual Studio 2013 is not used in this class.
- Perforce - Visual Client (p4v)
  - o [www.perforce.com](http://www.perforce.com)
- Download and configuration instructions will be provided in class
  - o Server address: ***140.192.39.61:1666***

## Topics will include:

- ***Main Lectures:***
  - o 2$^{nd}$ pass on Graphics engine
    - ▪ Run time formats
    - ▪ Texture manager
    - ▪ Camera Culling
    - ▪ Bounding volumes
  - o 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)***
    1) Level of Detail
    2) Multiple rendering targets
    3) Input / Events trigger
    4) Sound System
    5) UI
    6) Threading
    7) Serialization / Networking

## Programming Assignments – 80%

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… - 40% (due beginning 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

- o 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
- o 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
- o Quaternion library
    - Write and integrate the quaternions into the math library
    - Validate against unit tests
- **Milestone 2**: Animation Engine 40% (due Week 11)
    - o 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
    - o 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
    - o Playback engine
        - Animation should be able to:
            - Play forward / backwards
            - Loop
            - Faster or slower playback rate
        - Transition to different animations
    - o Skinning
        - Write a converter to extract the skin and bone information from a fbx file
            - Fbx data needs to have simple animation
        - Display the animation with vertex skinning

## Perforce Submissions & Piazza Participation - 10%

- Everyone is expected to submit <u>at least 10 submissions</u> 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 <u>do better</u>!
    - 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 (Standford 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!***

## Final Exam – 10% (11<sup>th</sup> week)

Take home final exam covering the concepts and the material of the class.

- Take home exam
  - Not an essay, but actual engineering dialog

## Schedule:

| Week | Lecture | Assign | Due |
|---|---|---|---|
| 1 | Syllabus, Class Overview | PA1 - Archiver | |
| 2 | Quaternions<br>Keenan-Mods Quaternions<br>Quaternions Interface | PA2 - Math<br>    Quaternions | (PA1 - tracking date) |
| 3 | FBX Converter<br>Level of Details | PA3 - FBX converter<br>    *Models* | (PA2 - tracking date) |
| 4 | Converter Hands on demo<br>Model Viewer | PA4 - Model Viewer | (PA3 - tracking date) |
| 5 | Keyframe Animation<br>Starting Animation development demo | | (PA4 - tracking date) |
| 6 | Timer (Discrete time for animations)<br>Animation System | | <span style="color:red">Beginning PA1-PA4</span> |
| 7 | Keyframe animation<br>Animation converter<br>Design demos | PA5 - FBX converter<br>    *Anim data*<br>    *Skeleton* | |
| 8 | Hierarchy Animation<br>Skeletons<br>Design demos | PA6 - Animation<br>    System | (PA5 - tracking date) |
| 9 | Skinning<br>Pimpl | PA7 - Skinning | (PA6 - tracking date) |
| 10 | Animation System issues<br>Loading / unloading assess | Final Exam | (PA7 - tracking date) |
| 11 | | | <span style="color:red">Final Project<br>Final Exam<br>PA5-PA7</span> |

| | |
|---|---|
| April 3, 2015 | Last day to add classes to SQ2015 schedule |
| April 10, 2015 | Last day to drop classes with no penalty, Last day to select pass/fail option |
| April 11, 2015 | Grades of "W" assigned for SQ2015 classes dropped on or after this day |
| April 18, 2015 | Last day to select auditor status |
| May 15, 2015 | Last day to withdraw from SQ2015 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