Game Programming III (Search for Sprites)

Spring 2018

GAM 576

instructor:	Ed Keenan
email:	ekeenan2@cdm.depaul.edu
office hours:	Wednesday 4:00-5:30pm, 9-10pm in classroom or by email appointment
office:	CDM 830
phone:	(312)362-6747
website:	piazza.com/depaul/spring2018/gam576 (Preferred communication)
lecture:	CDM 216, Thursdays, 5:45-9:00pm
learning system (D2L):	d2l.depaul.edu

Description:

Game Engine Programming III (Search for Sprites) is being offered this term. This class continues to explore C/C++ game engine programming. This is a topics class focusing on advanced game engine issues such as Skinning, Scene Graphs, Object management, Cameras and other material previous not covered in Game Engine I and II. The previous quarter's game engine will be furthered strengthen with more systems integrated into our framework.

Prerequisites:
- GAM 575 Game Programming Engine II

Learning Goals:
- Students should be able to design and implement a skinned animation system.
- Students should be able to create asset conversion tools for 3D models and skinned animations with offline compression.
- Students should be able to have the ability to identify, research and implement new game technologies.
- 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 required milestones (40% each)
- o Skinned Model and Animation Compression - Conversion tool, compression and runtime implementation for skinned models with keyframed animation data

20% - Individual Research project
- o Project proposed by student and approved by instructor

Textbooks and printed resources

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

- 1 Required Book
  - o <u>Game Engine Architecture</u>, 2nd edition, Gregory, A.K. Peters Ltd., 2015
    - ISBN: 978-1466560017
- Various GPU and Shading cookbooks
  - o To be researched and determined
  - o Online material

Software

- ***Microsoft Visual Studio 2017 Enterprise Edition (not Community)***
  - o [MSDNAA Depaul – Visual Studio 2017 Enterprise](#)
    - C++ and C# install (future classes)
  - o Any other variants are not used in this class
  - o Students are responsible keeping their development tools working
- ***Perforce Server***
  - o Download and configuration instructions will be provided in class
  - o ***Perforce – Helix Visual Client (p4v) -*** www.perforce.com
    - Server address: ***140.192.39.61:1666***

Topics will include:

- ***Tentative Lecture Topics:***
  - o Skinning Models
    - extraction of data
    - reducing the number of influence
    - runtime layout
    - compression
  - o Advanced Keyframe Animation
    - compression
    - data layout
    - CPU/GPU issues
    - runtime formats
  - o Scene Graph
    - agnostic tree layout for heterogeneous data types
    - models, skeletons, animations, cameras
    - Spacial layout
    - Quad/Octal trees, Feudal, Binary Spacial Partitioning
    - Bounding volumes
  - o Object Management
    - Layout / organization

- reflection / communication
- Self describing
- Protocol buffers
    - Cameras
        - Movement - paths
        - Auto-targeting
        - Spring (dynamic follow)

Required Programming Assignments – 80%

2 required 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.

- **Req. Milestone 1**: Model Skinning - 40% (due Week 5)
    - Write exporter tool to extract Skinned model data
        - Reduce the number of influences from N to 4 per vertex
        - Create a runtime format
    - Create the corresponding GPU code for the skinned data
        - Rework the game Engine to load the skinned data
        - Create the corresponding GPU systems to have the skinned data automated
- **Req. Milestone 2**: Advanced Animation - 40% (due Week 11)
    - Write a an animation extractor tool that exports compressed keyframe animation
        - Export animation data for a FBX file
        - Compressed the data using an offline tool
        - Convert the runtime to process the compressed data
            - Combination of CPU / GPU code
        - Replay animations using GPU code
        - Add blending between 2 or more separate animations

Individual Programming Assignments – 20%

1 individual project. There are mini checkpoints to make sure the student is on convergence path between milestones.

- **Indv. Milestone 1**: Student's choice - 20%
    - Student picks a topic that requires research and software experimentation
        - Will look at current industry/academic practices
        - Write a summary / present summary of research material
    - Create a prototype of technology
    - Project needs to be approved by professor

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 *Perforce RULE*
- 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 **_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 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 **_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 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

Tentative Schedule:

Week	Lecture	Assign	Due
1	Engine Review Clean-Up Refactor Topics / Layout	Project Selection Engine CleanUp	
2	Exporter Skinning GPU issues / layout	(Skinning GPU)	Project Selection Engine CleanUp
3	Skinning GPU coding GPU and data implementation design		
4	Animation Compression Skinning cleanup		
5	GPU animation	(Animation GPU)	(Skinning GPU)
6	Blending Animations Puppet animations		
7	Scene Graphs Presentation training	1st pass of slides	
8	Object Management Protocol Buffers		Feedback Review
9	Camera Systems / Movements Presentations (4)		Student Presentations begin
10	Open Topics Presentations (6)		
11	Remaining presentations (6)		(Animation GPU) Student write up

April 1, 2018	Last day to add classes to SQ2018 schedule
April 6, 2018	Last day to drop classes with no penalty, Last day to select pass/fail option
April 7, 2018	Grades of "W" assigned for SQ2018 classes dropped on or after this day
April 13, 2018	Last day to select auditor status
May 11, 2018	Last day to withdraw from SQ2018 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 or COLWeb 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