USC Viterbi School of Engineering

Programming Game Engines

ITP 485 (4 Units)

Objective	This course provides students with an in-depth exploration of 3D game engine architecture. Students will learn state-of-the-art software architecture principles in the context of game engine design, investigate subsystems typically found in a real game engine, survey engine architectures from actual shipped games, and explore how the differences between game genres can affect engine design. Students will participate in individual hands-on lab exercises to reinforce these concepts.
Concepts	Engine subsystems including rendering, audio, collision, physics, and game world models. Large-scale C++ software architecture in a games context. Tools pipelines for modern games.
Prerequisites	ITP 380
Instructor	Sanjay Madhav
Contact	**Email:** madhav@usc.edu
Office Hours	M/W 3:30–4:50PM and T 2-3PM, all in OHE 530H
Lab Assistant	TBA
Lecture	T 12-1:50PM in KAP 107
Lab	Th 12-1:50PM in KAP 107
Course Structure	Throughout the semester, students will work either by themselves to build out a skeleton game engine framework, which by the end of the semester will feature per-pixel lighting and animation, among other features. These assignments must be completed *individually*. There will periodically be “pop” quizzes. These quizzes will generally be short 20-30 minute programming assignments, and can occur during any class session. There are a total of six quizzes, with the lowest quiz grade being dropped. Towards the end of the semester, students will schedule a one-on-one mock interview with the instructor. During this mock interview, questions related to topics covered in class will be asked in a format which is similar to actual industry interviews. This is to ensure the students can verbalize and explain the concepts they should have learned during the semester. There are two exams which are comprehensive of all topics covered.
Textbooks	**Required:** *Game Engine Architecture.* Jason Gregory. ISBN-13: 978-1568814131. **Optional:** *Effective C++ (3rd Edition)*. Scott Meyers. ISBN-13: 978-0321334879.

Grading

The course is graded with the following weights:

Lab Assignments (10% each)	40%
Mock Interview	10%
Quizzes (6 total, lowest grade dropped)	10%
Midterm Exam	20%
Final Exam	20%
TOTAL POSSIBLE	100%

If a lab assignment receives a grade lower than 50%, the student(s) are required to resubmit it with a satisfactory amount of completion, or they will automatically receive an F in the course.

Grading Scale

Letter grades will be assigned according to the following scale:

93%+	A
90-92%	A-
87-89%	B+
83-86%	B
80-82%	B-
77-79%	C+
73-76%	C
70-72%	C-
69	D+
67-68	D
66	D-
65 and below	F

Half percentage points will be rounded up to the next whole percentage. So for instance, 89.5% is an A-, but 89.4% is a B+.

There is no curving. Students will receive the grade they earn. Extra credit is generally not offered.

Policies

Make-up policy for exams: To make up for a missed exam, the student must provide a satisfactory reason (as determined by the instructor) along with proper documentation. Make-up exams are only allowed under extraordinary circumstances. Note that quizzes cannot be made up.

Late Assignments: As there are only four due dates the entire semester, late projects will not be accepted unless the student(s) meet the same criteria for making up exams.

Before logging off a computer, students must ensure that they have emailed or saved projects created during the class or lab session. Any work saved to the computer will be erased after restarting the computer. ITP is not responsible for any work lost.

ITP offers Open Lab use for all students enrolled in ITP classes. These open labs are held beginning the second week of classes through the last week of classes. Hours are listed at: http://itp.usc.edu/labs/.

Academic Integrity	USC seeks to maintain an optimal learning environment. General principles of academic honesty include the concept of respect for the intellectual property of others, the expectation that individual work will be submitted unless otherwise allowed by an instructor, and the obligations both to protect one's own academic work from misuse by others as well as to avoid using another's work as one's own. All students are expected to understand and abide by these principles. *SCampus*, the Student Guidebook, (www.usc.edu/scampus or http://scampus.usc.edu) contains the University Student Conduct Code (see University Governance, Section 11.00), while the recommended sanctions are located in Appendix A. Students will be referred to the Office of Student Judicial Affairs and Community Standards for further review, should there be any suspicion of academic dishonesty. The Review process can be found at: http://www.usc.edu/student-affairs/SJACS/. Information on intellectual property at USC is available at: http://usc.edu/academe/acsen/issues/ipr/index.html. In this class, all code submissions will be ran against current, previous, and future students using MOSS, which is a code plagiarism identification tool. If your code significantly matches another student's submission, you will be reported to SJACS. Generally, the rule of thumb is that it is acceptable to discuss solutions to problems with other students, but once you are looking at someone else's code, it crosses over into the realm of cheating. It does not matter if this code is online or from a student you know, it is cheating in all situations. Do not share your code with anyone else in this or a future section of the course, as allowing someone else to copy off your code carries the same penalty as you copying the code yourself.
Students with Disabilities	Any student requesting academic accommodations based on a disability is required to register with Disability Services and Programs (DSP) each semester. A letter of verification for approved accommodations can be obtained from DSP. Please be sure the letter is delivered to me (or to TA) as early in the semester as possible. DSP is located in STU 301 and is open 8:30 a.m.–5:00 p.m., Monday through Friday. Website and contact information for DSP: http://sait.usc.edu/academicsupport/centerprograms/dsp/home_index.html, (213) 740-0776 (Phone), (213) 740-6948 (TDD only), (213) 740-8216 (FAX) ability@usc.edu.
Emergency Preparedness	In case of a declared emergency if travel to campus is not feasible, USC executive leadership will announce an electronic way for instructors to teach students in their residence halls or homes using a combination of Blackboard, teleconferencing, and other technologies. Please activate your course in Blackboard with access to the course syllabus. Whether or not you use Blackboard regularly, these preparations will be crucial in an emergency. USC's Blackboard learning management system and support information is available at blackboard.usc.edu.

Course Outline

Week 1 (8/27 and 8/29) - Introduction, Assembly, and SIMD

- Course introduction
- Look at x86 Assembly
- SIMD

Reading: Blackboard: "SIMD Tutorial," *Gregory*: §1.2 - §1.6, §4.7

Lab: Begin work on Lab 1: SIMD and Pool Allocator

Week 2 (9/3 and 9/5) - Software Engineering for Games

- Custom memory allocators
- Errors, exceptions, and assertions
- Data structures and design patterns

Reading: *Gregory*: §3.2.2 - §3.2.5, §3.3, §5.2 - §5.4

Lab: Continue work on Lab 1

Week 3 (9/10 and 9/12) - Tools of the Trade

- The compiler
- Optimization
- C++ constructs

Reading: *Gregory*: §2.1 - §2.5, §3.1

Lab: Finish work on Lab 1

Lab 1 DUE Sunday, 9/15 @ 11:59PM

Week 4 (9/17 and 9/19) - Gameplay Foundation Systems

- Components of the gameplay layer
- Runtime object model architectures
- Introduction to Windows Game Loop

Reading: *Gregory*: §14.2; §14.4, §6.2.2.7, §7.3

Lab: Begin work on Lab 2: Component Model and Level Loading

Week 5 (9/24 and 9/26) - Advanced Rendering, Part 1

- The rendering pipeline and DirectX
- Visibility determination and scene graphs
- Render states, sorting, alpha blending and Z pre-pass

Reading: *Gregory*: §10.1, §10.2.1 - §10.2.5

Lab: Continue work on Lab 2

Week 6 (10/1 and 10/3) - Advanced Rendering, Part 2

- Introduction to Shaders
- Advanced Shaders
- Global illumination and other techniques

Reading: *Gregory*: §10.2.6 - §10.2.7

Lab: Continue work on Lab 2

Week 7 (10/8 and 10/10) – **Midterm Exam**

- Midterm Exam during **lecture hours**

Lab: Finish work on Lab 2

Lab 2 DUE Sunday, 10/13 @ 11:59PM

Week 8 (10/15 and 10/17) – Advanced Sound Programming

- Sound System Programming
- Low level programming

Reading: §10.3 - §10.5
Lab: Begin work on Lab 3: Per-Pixel Lighting

Week 9 (10/22 and 10/24) - Hardware Considerations and Advanced 3D Math

- Integer and IEEE floating-point formats
- Console hardware overview and writing cross-platform code
- Quaternions, Splines, and Pseudo-Random Number Generation

Reading: *Gregory*: §12.3.5 - §12.5
Lab: Continue work on Lab 3

Week 10 (10/29 and 10/31) - Advanced Game Physics

- Quick review of physics basics
- Fast moving bodies and the bullet-through-paper problem
- GJK and AABB prune/sweep algorithms
- Typical physics/collision system architectures

Reading: *Gregory*: §12.3.5 - §12.5
Lab: Continue work on Lab 3

Week 11 (11/5 and 11/7) – Animation System Architecture

- Quick review of animation basics
- Blending and compression techinques
- Animation system architecture and pipeline

Reading: *Gregory*: §11.6 - §11.12
Lab: Continue work on Lab 3
Lab 3 DUE Sunday, 11/10 @ 11:59PM

Week 12 (11/12 and 11/14) - Engine Subsystem Integration

- Updating a multi-object simulation in real time
- Integrating rendering, physics, and animation into the game loop
- Multiprocessor game loops

Reading: *Gregory*: §7.1 - 7.6; §14.6
Lab: Begin work on Lab 4: Skeletal Animation and Demo Camera

Week 13 (11/19 and 11/21) - Multiplayer

- UDP, TCP/IP, ICMP, and the socket layer
- Client/Server, Peer-to-Peer, and other networking models
- Platform services (Xbox Live, PSN, Etc)

Reading: Multiplayer whitepaper on Blackboard
Lab: Continue work on Lab 4

Week 14 (11/26) – Advanced C++

- Templates
- Function Objects

Reading: C++ reading on Blackboard
Lab: Continue work on Lab 4
No class 11/28 due to Thanksgiving.

Week 15 (12/3 and 12/5) – Content/Engine Pipelines

- DCC pipelines
- Engine framework software design

Lab: Continue work on Lab 4
Lab 4 DUE Friday, 12/6 @ 11:59PM

Final Exam 12/17 @ 11AM