USC
VITERBI
SCHOOL OF
ENGINEERING

Programming Game Engines
ITP 485 (4 Units)

Spring 2012

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

Contacting the Instructor *Email:* madhav@usc.edu (Preferred in almost all cases)

Office Hours TBD

Lab Assistant

Lecture 3-4:50PM T @ KAP 107

Lab 3-4:50PM Th @ KAP 107

Course Structure Every week, we will cover a variety of game engine topics during lecture.

Throughout the course of the semester, students will be building out basic game engine systems in their lab assignments. Students will be allowed to work alone, or with a partner.

There are two exams which are comprehensive of all topics covered.

Textbooks **Required:**
Game Engine Architecture. Jason Gregory. ISBN: 978-1568814131.

Optional:
Mathematics for 3D Game Programming & Computer Graphics. Eric Lengyel. ISBN: 978-1584502777.

Effective C++ (3rd Edition). Scott Meyers. ISBN: 978-0321334879.

Large-Scale C++ Software Design. John Lakos. ISBN: 978-0201633627.

Grading

The course is graded with the following weights:

Lab Assignments (15% each)	60%
Exams (20% each)	40%
TOTAL POSSIBLE	100%

There is no curving. Students will receive the grade they earn.

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.

Late Projects: 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. Please contact your instructor for specific times and days for the current semester.

Academic Integrity

The use of unauthorized material, communication with fellow students during an examination, attempting to benefit from the work of another student, and similar behavior that defeats the intent of an examination or other class work is unacceptable to the University. It is often difficult to distinguish between a culpable act and inadvertent behavior resulting from the nervous tension accompanying examinations. When the professor determines that a violation has occurred, appropriate action, as determined by the instructor, will be taken.

Although working together is encouraged, all work claimed as yours must in fact be your own effort. Students who plagiarize the work of other students will receive zero points and possibly be referred to Student Judicial Affairs and Community Standards (SJACS).

All students should read, understand, and abide by the University Student Conduct Code listed in SCampus, and available at: http://www.usc.edu/student-affairs/SJACS/nonacademicreview.html

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 your 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. The phone number for DSP is (213) 740-0776.

Course Outline

Week 1 (1/10 and 1/12) - Introduction, Assembly, and SIMD

- Course introduction
- What is a game engine? What are its subsystems?
- Look at x86 Assembly
- SIMD

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

Lab: Begin work on Lab 1

Week 2 (1/17 and 1/19) - 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 (1/24 and 1/26) **-** 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 1/29 @ 11:59PM

Week 4 (1/31 and 2/2) - 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

Week 5 (2/7 and 2/9) - 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, §10.2.6 - §10.2.7

Lab: Continue work on Lab 2

Week 6 (2/14 and 2/16) - Advanced Rendering, Part 2

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

Lab: Continue work on Lab 2

Week 7 (2/21 and 2/23) – **Exam 1**

- Exam 1 during **lecture hours**

Lab: Finish work on Lab 2

Lab 2 DUE 2/26 @ 11:59PM

Week 8 (2/28 and 3/1) – Advanced Sound Programming

- Sound System Programming
- Low level programming
- Discussion of DSP effect techniques

Reading: §10.3 - §10.5

Lab: Begin work on Lab 3

Week 9 (3/6 and 3/8) - Hardware Considerations and Advanced 3D Math

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

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

Spring Break (3/13 – 3/16)

Week 10 (3/20 and 3/22) - Advanced Game Physics

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

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

Week 11 (3/27 and 3/29) – Animation System Architecture

- Quick review of animation basics
- Blending and compression techinques
- Procedural animations, IK, and other post-processing
- Animation system architecture and pipeline

Reading: *Gregory*: §11.6 - §11.12
Lab: Continue work on Lab 3
Lab 3 DUE 4/1 @ 11:59PM

Week 12 (4/3 and 4/5) - 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

Week 13 (4/10 and 4/12) - Multiplayer

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

Reading: None
Lab: Continue work on Lab 4

Week 14 (4/17 and 4/19) – Advanced C++

- Templates
- Function Objects
- Other C++ Overflow Topics

Lab: Continue work on Lab 4

Week 15 (4/24 and 4/26) - Conclusion

- Overflow topics as necessary
- Review/QA

Lab: Continue work on Lab 4
Lab 4 DUE 4/29 @ 11:59PM

Week 16 – **Exam 2** (5/8 @ 2:00PM)