USC
SCHOOL OF
ENGINEERING

Game Engine Development
CSCI 522 (4 Units)

Objective The purpose of this course is to familiarize the student with the principles of developing game engines targeted at modern PC and game console hardware.

Concepts This course will provide basic and advanced techniques for designing and developing a game engine suitable for use on a wide variety of next-generation gaming hardware. Through implementing the essential systems of a high-performance, multi-threaded, cross-platform, real-time 3D game engine, the student will gain familiarity with the characteristics of such engines.

First, the students will learn the principles and techniques associated with software development of a game engine. The students will learn to use and/or understand the tools needed to make games and build the game engine libraries and executables using MS Visual Studio Integrated Development Environment.

Each week the student will learn about every game engine subsystem and be exposed to advanced techniques currently used in the video game industry.

Then each student will be assigned a game engine module to design, develop and document around the USC Gamepipe Game Engine, GGE. The game engine will be based around OGRE 3D, an existing and popular Open Source Render API.

Through class projects the students will create software demos and documentation on their assigned game modules. The lessons learned will be shared with other students through class discussions and online documentation using Wiki pages.

Final project will consist of final presentation of the game engine at work including a peer presentation on how to use it as part of the GGE.

Prerequisite None

Lecture 3 hours a week

Textbook Required

Real-Time Rendering, Third Edition
by Tomas Akenine-Moller and Eric Haines

Textbook Optional

Pro OGRE 3D Programming
by GregoryJunker

3D Game Engine Design, 2nd Edition
by David H. Eberly

Game Programming Gems
Edited by Mark DeLoura

Grading

The following point structure will be used in determining the grade for the course. Final grade will be based upon the total points received, the highest total in the class, and the average of the class.

Attendance	200
Project 1	200
Project 2	100
Project 3	200
Final Project	300
TOTAL POSSIBLE	1000

Policies

- Make-up policy for exams: In order to make up for a missed exam, the student must provide a satisfactory reason along with proper documentation. Usually make-ups are allowed only under extraordinary circumstances.

- Late Projects: Only one project may be turned in late (and receive full points). No other late projects will be accepted.

- Though working together is encouraged, the projects must be your own effort. “Duplicate” projects will all receive zero points and possible referral to the Office for Student Conduct.

- All students should read, understand and abide by the University Student Conduct Code http://www.usc.edu/dept/publications/SCAMPUS/governance/gov03.html

Game Engine Development

CSCI 522 (4 Units)

Course Outline

Week 1 – **Course overview and introduction to Game Engine Development**

- Game Engine Modules Overview
- Development Tools Overview
- Review of Existing Demos

Reading:
None
Project: Project 1 (Due Week 3)

Week 2– Development Environment

- Coding Standards and Documentation
- Source Control and Versioning
- Visual Studio 2005 IDE
- Building and Debugging GGE Demo
- Debugging vs. Release Code

Reading: *Pro Ogre 3D Programming* – Chapter 1-4
3D Game Engine Design – Chapter 1,18

Week 3 – Gamepipe Game Engine, pt 1

- Project 1 Review
- Scene Graph Management
- Materials
- Resource Management
- Render Target
- Camera System

Reading:
Pro Ogre 3D Programming – Chapter 5-8
3D Game Engine Design – Chapter 2-4
Game Programming Gems 2 – Section 4
Project: Project 2 Proposal (Due week 4)

Week 4 – **Gamepipe Game Engine, pt 2**

- Project 2 Proposal Review
- Skeletal Animation
- Morph Animation
- Pose/Blend Shape Animation
- Facial Animation

Reading:
Pro Ogre 3D Programming – Chapter 9

3D Game Engine Design – Chapter 5, 17
Game Programming Gems 1 – Section 4, 13-15

Week 5 – **Gamepipe Game Engine, pt 3**

- User Interface
- Overlays
- Billboards
- Particles
- 2D Animations
- Special Effects

Reading:
Pro Ogre 3D Programming – Chapter 10, 12
3D Game Engine Design – Chapter 7

Week 6 – **Gamepipe Game Engine, pt 4**

- Project 2 Approval
- Collada File Format
- Input Devices
- FMV Cinematics
- Audio

Reading:
Pro Ogre 3D Programming – Appendix A
Game Programming Gems 2 – Section 6
Game Programming Gems 3 – Section 6
Project: Project 3 (Due week 8)

Week 7 – **Gamepipe Game Engine, pt 5**

- Physics
- Collision Detection

Reading:
3D Game Engine Design – Chapter 8-9
Game Programming Gems – Section 2

Week 8– **Gamepipe Game Engine, pt 6**

- AI

Reading:
Pro Ogre 3D Programming – Appendix B
Game Programming Gems 1,2 – Section 3

Week 9 – **Advance Computer Graphics Techniques**

- Visual Appearance
- Realistic and Non-Photo realistic Rendering
- Shadows

Reading:
Pro Ogre 3D Programming – Chapter 11
3D Game Engine Design - Chapter 20

Real-time Rendering - Chapter 4, 6, 7
Game Programming Gems 1 – Section 5

<u>Week 10</u> – **Networking**

- Peer to Peer vs. Client/Server
- Designing Game Data Structures for a Network Layer
- Latency
- Network Packet Structure
- Determine what to keep in synch and what not to care for
- Game Networking Best Practices

Reading:
Game Programming Gems 3 – Section 5

<u>Week 11</u> – **Multi-threading, Cross-Platform development, Optimizations**

- Determine what is using CPU time, physics, AI, graphics…
- Incremental processing and real-time operations
- Variety and Popularity of Different Game Consoles and Media
- Platform dependent vs. platform independent code
- "Refactoring" your engine for easier porting
- GPU Growing Power
- Pipeline Optimization
- CPU and GPU Profiling Tools

Reading:
3D Game Engine Design - Chapter 19
Real-time Rendering - Chapter 10-12,15
Project: Final Project (Due week 13)

Week 12 – **Growing with the Engine**

- Development of new Tools and Gameplay Widgets
- Expanding Level Editor with Real-time Update
- Optimize, Rewrite and Add new Features to Existing Engine
- Expand Low-Level Hardware for new Consoles

Reading:
Real-time Rendering - Chapter 16

Week 13 – **Final Project**

- Final Project and Peer Presentations
- Final Project Documentation

Reading:
None

The following list summarizes the potential student game engine module projects targeted for this semester.

- GGE Integration of Intel Havok Physics, Animation and Behavior SDK
- GGE Live Scene Editor
- GGE Networking
- GGE Performance, Multi-Core Threading
- GGE Shader Pipeline and Collada SupportnGGE Build Installer/Website Integration
- xGGE MS X360 XNA Game Engine
- iGGE Apple iPhone Game Engine
- LSF Game