



Introduction to Game Programming

ITP 380 (4 Units)

Objective This course provides students with an introduction to the technologies and software engineering practices used in the video game industry today. Students will learn the basics of creating a PC game based on Microsoft® DirectX®, through lecture material, hands-on labs, and a final project in which the students will actually build a simple game from the ground up.

Concepts Modern game software architecture. 3D mathematics for games. Working with DirectX9®. The 3D graphics pipeline. Game worlds and game objects. Collision detection. Introduction to game physics. Events and scripting. Introduction to animation and A.I. Implementing game play. Getting a job in the game industry.

Prerequisite Proficiency in C/C++ programming for the Windows platform.

Lecture 2 hrs/week

Lab 2 hrs/week

Textbooks

1. *Game Programming Complete*. Mike McShaffry. ISBN 1-932111-75-1. Publisher: Paraglyph Press Inc.
2. (optional) *Mathematics for 3D Game Programming and Computer Graphics*. Eric Lengyel. ISBN 1-584500-37-9. Publisher: Charles River Media.

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.

Lab 1: Project set-up and unit testing	50
Lab 2: Writing a Vector Class	50
Lab 3: Loading and Drawing .x File Meshes	50
Lab 4: Building a Game Camera	50
Lab 5: Texturing and Lighting	50
Lab 6: Modeling the Game World and Objects	50
Lab 7: Collision Detection	50
Midterm	300
<u>Final Project: Complete Your Game</u>	<u>350</u>
TOTAL POSSIBLE	1,000

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.

Labs: It is the student's responsibility to turn in lab projects on or before deadlines as set by the instructor.

Late Labs: Rules for late lab or final project submission will be established by the instructor.

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 instructor determines a violation has occurred, appropriate action, as determined by the instructor, will be taken.
- 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>

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.

Introduction to Game Programming

ITP 380 (4 Units)

Course Outline

Week 1 – Introduction and course overview:

- Working in the electronic entertainment industry
- Overview of modern game software architecture

Week 2 – Foundations of game programming:

- Review of software engineering basics
- Project organization, source control
- Unit testing
- Core engine systems

Lab 1: Project set-up and unit testing

Week 3 – Mathematics for games:

- Matrix and vector properties
- Matrix operations
- Atomic transformations
- Coordinate systems, representations of rotation

Lab 2: Writing a Vector Class

Week 4 – Windows® games: The main game loop:

- Window creation, destruction, message handling
- Writing a game loop, integrating with Windows® message loop
- Direct3D basics

Lab 3: Loading and drawing .x file meshes

Week 5 – The 3D graphics pipeline:

- Typical pipeline
- The view frustum
- Coordinate spaces and transformations between them
- Simple culling and clipping

Lab 4: Building a game camera

Week 6 – More DirectX

- More D3D: render states, vertex buffers, materials, lighting basics
- DirectInput
- DirectSound

Lab 5: Texturing and Lighting

Week 7 – Midterm prep

- Review, Q&A

Week 8 – MIDTERM EXAM

Week 9 – Game worlds and game objects

- Discussion of game world / game object class hierarchies
- Types of game objects: visible, helpers, cameras, lights, etc.
- File formats for game world representation

Lab 6: Modeling the game world and objects

Week 10 – Collision

- Collision detection versus collision response
- Simple collision detection: sphere vs. sphere
- Sphere vs. line collision; sphere vs. plane collision
- Some simple collision response approaches

Lab 7: Collision Detection

Week 11 – Game Physics

- Animation vs. arcade physics vs. semi-realistic simulations
- Linear dynamics
- Simple numerical integration
- Physical collision resolution: impulse
- Introduction to rotational dynamics

Lab: Begin Work on Final Project

Week 12 – Event System and Scripting

- Review of event-driven programming
- A simple event system implementation; some common problems
- Using scripting languages for game logic

Lab: Work on Final Project

Week 13 – Animation and A.I.

- Introduction to skeletal/skinned animation
- Overview of modern animation system features
- Basic responsibilities of the A.I. system; state machines
- Simple path finding: A*
- Linkage between A.I. and animation

Lab: Work on Final Project

Week 14 – Game Play and the Asset Pipeline

- Approaches to implementing game play logic
- Overview of a typical asset pipeline
- Choosing asset file formats

Lab: Work on Final Project

Week 15 – Getting Into the Game Industry

- What it's like being a game programmer: pros and cons
- How to succeed in an interview: sample interview questions

Lab: Complete Your Final Project!

Week 16 – Review of final projects

