Number: CS G148  
TITLE: Game Programming, Intermediate

ORIGINATOR: Omid Pourzanjani  
EFF TERM: Fall 2008

FORMERLY KNOWN AS:

DATE OF OUTLINE/REVIEW: 11-16-2006

CROSS LISTED COURSE:

CID:

SEMESTER UNITS: 4.0
HRS LEC: 54.0  
HRS LAB: 54.0  
HRS OTHER: 0.0
CONTACT HRS TOTAL: 108.0
STUDY NON-CONTACT HRS RECOMMENDED: 108.0

CATALOG DESCRIPTION:
This course will present intermediate techniques and principles in developing computer games. Students will study and develop custom game engines, experiment with custom shaders using HLSL (High-Level Shading Language), interact with gaming input devices, study object motion and collision, integrate sound, and build and import characters using animation software. Visual Studio .Net will be used to create managed DirectX programs on the Windows platform. ADVISORY: Computer Science G147

JUSTIFICATION FOR COURSE:

PREREQUISITES:

COREQUISITES:

ADVISORIES:

- CS G147: Game Programming, Introduction

ASSIGNED DISCIPLINES:

Computer science

MATERIAL FEE: Yes [ ] No [X] Amount: $0.00

CREDIT STATUS: Noncredit [ ] Credit - Degree Applicable [X]  Credit - Not Degree Applicable [ ]

GRADING POLICY: Pass/No Pass [X]  Standard Letter [X]  Not Graded [ ]  Satisfactory Progress [ ]

OPEN ENTRY/OPEN EXIT: Yes [ ] No [X]

TRANSFER STATUS: CSU Transferable[X]  UC/CSU Transferable[ ]  Not Transferable[ ]

BASIC SKILLS STATUS: Yes [ ] No [X]  LEVELS BELOW TRANSFER: Not Applicable

CALIFORNIA CLASSIFICATION CODES: Y - Not Applicable

NON CREDIT COURSE CATEGORY: Y - Not applicable, Credit Course

OCCUPATIONAL (SAM) CODE: C

REPEATABLE ACCORDING TO STATE GUIDELINES: No [X]  Yes [ ]
NUMBER REPEATS:

REQUIRED FOR DEGREE OR CERTIFICATE: No [ ] Yes [X]

Video Game Development(Associate in Arts)
Video Game Development(Certificate of Achievement)

GE AND TRANSFER REQUIREMENTS MET:

COURSE LEVEL STUDENT LEARNING OUTCOME(S) Supported by this course:

Explain the fundamentals of game physics and artificial intelligence (AI).
1. Explain the fundamentals of game physics and artificial intelligence (AI).

2. 1. Compare and contrast existing game engines.

3. Describe the internal components of a game engine and their interactions.

**COURSE OBJECTIVES:**
1. Design and create a custom 3D game engine.
2. Effectively use light sources and virtual cameras in the 3D space.
3. Create and apply object texturing, alpha blending, stenciling, transparency, and shadows.
4. Use animation software and .x files.
5. Detect objects and create collisions.
7. Create custom shaders with HLSL (High-Level Shading Language).
8. Render 3D Terrain.
10. Interact with user Input/Output devices.

**COURSE CONTENT:**

**LECTURE CONTENT:**

The following outline describes the course contents and objectives:

A. Design and develop custom game engines
B. Design and apply custom texturing and lighting to objects
C. Build complex meshes
D. Use virtual cameras in 3D space
E. Design 3D terrains
F. Develop algorithms for dealing with objects
   1. moving objects
   2. object collision
   3. object detection
   4. shooting
   5. explosions
G. Study the principles of gaming mathematics and physics
H. Use animation software to
   1. build models
   2. import and export models
   3. Utilize models in 3D game engines
I. Develop and integrate music for games
J. Process complex user commands
K. Evaluate and improve performance of 3D game algorithms and engines

**LABORATORY CONTENT:**

The following outline describes the course contents and objectives:

A. Design and develop custom game engines
B. Design and apply custom texturing and lighting to objects
C. Build complex meshes
D. Use virtual cameras in 3D space
E. Design 3D terrains
F. Develop algorithms for dealing with objects
   1. moving objects
   2. object collision
   3. object detection
   4. shooting

5. explosions
G. Study the principles of gaming mathematics and physics
H. Use animation software to
   1. build models
   2. import and export models
   3. Utilize models in 3D game engines
I. Develop and integrate music for games
J. Process complex user commands
K. Evaluate and improve performance of 3D game algorithms and engines

METHODS OF INSTRUCTION:

A. Lecture:
B. Lab:
C. Other simultaneous interactive:
D. Online:
E. Independent Study:

INSTRUCTIONAL TECHNIQUES:

COURSE ASSIGNMENTS:

Reading Assignments

Students will be assigned multiple chapters from the required textbook(s).
Additional material as assigned by instructor to encourage further studies into specific topics.
Students will be assigned to read a large amount of online material found on various web pages.
Various current handouts on the subjects of 3D games, game engines, animation, and multimedia
will be provided by the instructor.

Out-of-class Assignments

An optional library research paper will promote further study and research in 3D interactive game
development or other related topics selected by the student and approved by the instructor.
Guest lecturers will be invited to discuss the current subjects/trends in 3D game applications.
Students will be required to write summary papers on the presentations.

Writing Assignments

Students will be required to complete the following assignments:
Research paper on 3D game engines
4 Programming assignments
1 individual or group term-project

METHODS OF STUDENT EVALUATION:

Midterm Exam
Final Exam
Short Quizzes
Written Assignments
Objective Examinations
Report
Projects (ind/group)
Problem Solving Exercises
Oral Presentations
Skills Demonstration

Demonstration of Critical Thinking:

Class assignments will be presented to the students in the form of problems requiring students to devise
solutions in the form of software development projects.
Students will be given various game engine technology architectures to study and analyze.
Required Writing, Problem Solving, Skills Demonstration:
Students will be required to complete the following assignments:
Research paper on 3D game engines
4 Programming assignments
1 individual or group term-project

TEXTS, READINGS, AND RESOURCES:
TextBooks:

Other:
1. A syllabus and multiple reference materials will be distributed by the instructor.

LIBRARY:
Adequate library resources include:

Comments:

Attachments:
[Attached Files](#)