COURSE OUTLINE OF RECORD

Number: CS G149  TITLE: Game Programming, Advanced

ORIGINATOR: Omid Pourzanjani  EFF TERM: Summer 2010
FORMERLY KNOWN AS:  
CROSS LISTED COURSE:  

SEASONAL 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 advanced techniques and principles in developing multiplayer computer games. Students will study and develop advanced game engines, optimized 3D rendering sub-systems, Level Editors, and artificial intelligence algorithms for multiplayer online games. Visual Studio .Net and DirectX will be used to create multiplayer games on the Windows platform. ADVISORY: Computer Science G148

JUSTIFICATION FOR COURSE:

PREREQUISITES:

COREQUISITES:

ADVISORIES:
- CS G148: Game Programming, Intermediate

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: B

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:
1. Understand the basic principles of computer networking.
2. Describe the components of a Multiplayer Online Game and the distribution and interaction of these components on the network.
3. Explain the tools used by teams of developers to create Massively Multiplayer online Games (MMOGs).

**COURSE OBJECTIVES:**
1. Demonstrate an understanding of Client-Server application fundamentals.
2. Develop networked applications using socket-based programming libraries and techniques.
3. Integrate DirectPlay functionality into games.
4. Demonstrate and understanding of game server principles and components.
5. Manage real-time and interactive game data.
6. Implement 3D models, terrains, and special effects (fog, water, fire, clouds, etc.).

**COURSE CONTENT:**

**LECTURE CONTENT:**

A. Overview
   1. Multiplayer game internals
   2. Networking basics
   3. Client-Server applications
   4. Richness of data versus transmission efficiency

B. Custom Game Engine
   1. Componentization of a game engine
   2. A distributed approach to game physics
   3. Strategies in managing real-time and interactive game data
   4. DirectPlay functionality

C. 3D Models and Terrains
   1. Model/Character interactions in a multiplayer game
   2. Lighting and texturing considerations
   3. Static versus dynamic scenes and terrains
   4. Camera/View considerations

D. Game Mathematics and Physics
   1. Moving objects
   2. Object detection
   3. Object collision
   4. Shooting
   5. Explosions
   6. Special effects for fog, water, fire, clouds, etc.

E. Animation software
   1. Model (game asset) construction
   2. Format conversion
   3. Texturing and UV mapping

**LABORATORY CONTENT:**

A. Overview
   1. Multiplayer game internals
   2. Networking basics
   3. Client-Server applications
   4. Richness of data versus transmission efficiency

B. Custom Game Engine
   1. Componentization of a game engine
   2. A distributed approach to game physics
3. Strategies in managing real-time and interactive game data
4. DirectPlay functionality

C. 3D Models and Terrains
1. Model/Character interactions in a multiplayer game
2. Lighting and texturing considerations
3. Static versus dynamic scenes and terrains
4. Camera/View considerations

D. Game Mathematics and Physics
1. Moving objects
2. Object detection
3. Object collision
4. Shooting
5. Explosions
6. Special effects for fog, water, fire, clouds, etc.

E. Animation software
1. Model (game asset) construction
2. Format conversion
3. Texturing and UV mapping

METHODS OF INSTRUCTION:
A. Lecture:
B. Lab:
C. Independent Study:

INSTRUCTIONAL TECHNIQUES:

COURSE ASSIGNMENTS:

Reading Assignments
Such as:
Multiple chapters from the required textbook
Additional material as assigned by instructor to encourage further studies into specific topics
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 multiplayer online games
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)
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 multiplayer online games 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](#)