Degree Applicable

Glendale Community College
January 2012

COURSE OUTLINE

Computer Science/Information Systems 153
3D Game Engine Application Development

I. Catalog Statement

Computer Science/Information Systems 153 is an introduction to 3D programming concepts and the application of those concepts using 3D game engines. 3D game programming concepts presented will include coordinate systems, transformations, and rendering. Emphasis will be on understanding and using the large libraries of code that make up a 3D game engine to build 3D game applications. A real- world game engine will be used in the course to code and debug a simple 3D game.

Units – 4.0
Lecture/Demonstration Hours – 4.0

Prerequisite: Computer Science/Information Systems 135, or equivalent.

II. Course Entry Expectations

Skills Level Ranges: Reading 4; Writing 4; Listening/Speaking 4; Math 4.

Prior to enrolling in the course, the student should be able to:

1. analyze a programming task in order to develop and communicate efficient algorithms to implement that task;
2. evaluate the workability of a programming solution based on a knowledge of computing machinery internal operations and their connections with peripheral equipment;
3. recognize programming problems on a function-by-function basis and develop structured/procedural code based on this approach;
4. demonstrate an understanding of object-oriented programming concepts and object-oriented design;
5. design, code, and debug basic object-based programs;
6. program in the C++ language including use of objects, pointers, and structures.

III. Course Exit Standards

Upon successful completion of the required coursework, the student will be able to:

1. explain basic 3D game programming concepts, basic 3D programming techniques, and the role 3D game engines play in 3D game programming and design;
2. code basic game scripts using the 3D game engine scripting language;
3. explain the organization and elements of the 3D game engine and how the game elements are organized to create a game;
4. explain the role of the artist in game development and utilize the 3D game engine to incorporate the game assets the artist provides into a game;
5. design, code, and debug a basic 3D game using 3D game engine scripting language.

IV. Course Content **Total Contact Hours = 64**

A. Introduction to 3D Game Development 3 hours
   1. The 3D game programming industry
   2. 3D game elements (engine, scripts etc.)

3. Installing the 3D game engine

B. 3D Game Programming Concepts 4 hours
1. Coordinate systems, transformations, rendering, scene graphs, and 3D audio
2. 3D programming concepts illustrated in the 3D game engine

C. 3D Engine Overview–Engine Organization 4 hours
1. Client server architecture
2. Event-driven simulator
3. Additional features (input architecture, game assets organization)

D. Introduction to the 3D Game Engine Scripting Language 9 hours
1. Good scripting language features
2. Reasons for using a scripting language
3. Setting up to program using the 3D Engine scripting language
4. 3D Engine script programming concepts
a. Variables, data types, operators,
b. Control structures, loops, functions
c. Objects
d. Console methods
e. Namespaces and datablocks

E. The 3D Game Engine Basic Game Structure 7 hours
Understanding and organizing modules
1. Server vs. client design issues
2. Modules: root main and control main
3. Modules: client, server, and player
4. Using the modules to create a basic game with an avatar

F. Intermediate 3D Game Engine Control Modules 6 hours
1. Client control modules (creating player interfaces)
2. Server control modules (handling players and game objects)
3. Adding additional features to the basic game (intermediate game)

G. Creating the Game Network (Linking Client and Server) 8 hours
1. Direct messaging
2. Triggers
3. Game connection messages
4. Finding servers
5. Enhancing the intermediate level game

H. Common Scripts 4 hours
1. Game Initialization
2. Selected common server modules
3. Selected common client modules

I. Working with the Artist–Textures, Skins, Terrains 4 hours
1. The role of textures in the game
2. The role of skins in the game
3. The role of terrains in the game
4. Incorporating textures and skins into the game

J. Creating GUI Elements and Adding them to the Game 4 hours

K. Working with the Artist–3D Models and Animation 4 hours

L. Incorporating Sounds into the Game 4 hours
1. Object sounds
2. Music

M. Overall Game Development Process 3 hours

1. Roles of programmer, artist, and designer
2. 3D game engine tools for the game designer

V. Methods of Presentation

The following instructional methodologies may be used in the course:

1. lecture/demonstration;
2. hands-on activities;
3. multimedia;
4. online.

VI. Assignments and Methods of Evaluation

1. Midterm examination.
2. Final examination.
3. Programming projects (e.g. debugging a basic 3D game).

VII. Textbooks

Gregory, Game Engine Architecture, Current edition.
Wellesley, MA: A. K. Peters, Ltd., 2009.
10th Grade Textbook Reading Level. ISBN: 978-1-56881-413-1.

Maurina, E. F. III, The Game Programmer's Guide to Torque, Current edition.
Wellesley, MA: A. K. Peters, Ltd., 2006.
10th Grade Textbook Reading Level. ISBN: 978-1-5688-1284-7.

VIII. Student Learning Outcomes

Upon successful completion, the student will be able to:

1. explain basic 3D game programming concepts, basic 3D programming techniques, and the role 3D game engines play in 3D game programming and design.
2. code basic game scripts using the 3D game engine scripting language.
3. explain the organization and elements of the 3D game engine and how the game elements are organized to create a game.

.