

CENTER FOR PROFESSIONAL DEVELOPMENT**VIDEO GAME DESIGN AND DEVELOPMENT**

500 Hours/8 Months/Instructor-Facilitated

Course Code: **CPD126****OVERVIEW**

Excelsior College has partnered with ed2go to bring you the Video Game Design and Development program. Video game design and development is challenging, but the rewards are worth it. With this unparalleled comprehensive training program, you'll master skills that open doors to the growing video game industry.

Using a comprehensive and analytical approach to game engine architectures, this program offers you the opportunity to learn how to effectively implement game ideas. The curriculum is divided into eight modules covering five major areas of study: programming languages, mathematics skills, 3D graphics pipeline programming, real-time game engine architectures, and artificial intelligence algorithms. Developing and managing the complex environments for games and related visualization applications is a challenging task, but with the right training and professional guidance, the challenge becomes much less daunting.

OBJECTIVES

As a student, you will learn the following:

- Understand the core programming language skills necessary for game development
- Have a sufficient level of proficiency in the areas of mathematics common to almost all game projects
- Gain a thorough understanding of the main concepts involved in real-time 3D graphics programming

MATERIALS INCLUDED

Excelsior College / ed2go will provide the required textbook, *Physics for Game Developers*.

OUTLINE**I. C++ Programming for Game Developers I**

- A. How to Create a C++ Program, Console Input and Output, Variable, and Arithmetic Operators
- B. Logical Operators, Controlling Program Flow, Repetition, and Arrays
- C. Functions
- D. References and Pointers
- E. Classes and Object-oriented Programming Design
- F. Strings
- G. Operator Overloading

- H. File Input and Output
 - I. Inheritance and Polymorphism
- II. C++ Programming for Game Developers II**
- A. Template Classes and Template Functions
 - B. Error Handling
 - C. Number Systems, Data Representations, and Bit Operations
 - D. The Standard Template Library
 - E. Introduction to Windows Programming
 - F. Menus and Drawing With GDI
 - G. Dialog Boxes
 - H. Timing, Animation, and Sprites
 - I. Designing and Implementing a 2D Game
- III. Math Primer**
- A. Real Numbers
 - B. Algebra
 - C. Analytic Geometry
 - D. Triangles
- IV. Graphics Programming With DirectX 9 I**
- A. 3D Mathematics
 - B. The Transformation and Lighting Pipeline
 - C. Initializing Direct3D
 - D. Vertex and Index Buffers
 - E. Single and MultiTexture Effects
 - F. Camera Management Systems
 - G. Compressed Textures
 - H. Texturing and the Texture Blending Cascade
 - I. Alpha Blending
 - J. Loading GILES Scenes
- V. Game Mathematics**
- A. Set Theory and Functions
 - B. Polynomials
 - C. Trigonometry
 - D. Vector Mathematics
 - E. Matrix Mathematics
 - F. Quaternion Algebra
 - G. Linear Transformations
 - H. Analytic Geometry
- VI. Graphics Programming With DirectX 9 II**
- A. The D3DX Mesh Containers
 - B. Progressive Meshes
 - C. Scene Hierarchies
 - D. Keyframe Animation
 - E. Indexed and Non-indexed Vertex Blending
 - F. Software and Hardware Skinning Techniques
 - G. Skeletal Animation
 - H. Tree Animation and Rendering

- I. X Files
- J. Motion Blending
- K. Collision Detection and Response
- L. Quadtrees, Octrees, and kD-Trees
- M. Binary Space Partitioning (BSP) Trees and Potential Visibility Sets (PVS)

VII. Artificial Intelligence for Game Developers

- A. Decision Making
- B. Grid Traversal and Search Algorithms
- C. Path finding with A*
- D. Flocking Behaviors
- E. Finite State Machines
- F. Scripting
- G. Squad Level AI
- H. Waypoint Networks
- I. AI Engine Integration

VIII. Physics for Game Developers

- A. The Principles of Newtonian Physics
- B. Simulating Gravity
- C. Simulating Friction
- D. Modeling Acceleration and Velocity
- E. Trajectories
- F. Kinematics and Motion Control
- G. Collision Detection and Response
- H. Animation Technique

COMPUTER REQUIREMENTS

This program is compatible with the Windows XP and later operating systems and IE 7 and later browsers.

Prior to enrolling in this program, please ensure that your computer meets the following hardware and software requirements:

Hardware Requirements:

- Processor: Pentium III+/AMD Athlon (750MHz or greater)
- System Memory: 256 MB+
- Video: 3D Hardware Accelerator w/ 128MB+ on-board memory
DirectX 9.0/OpenGL compliant
- Disk: 1 GB+ uncompressed
- Sound: DirectSound compliant sound card

Software Requirements:

- Microsoft Visual C++ 6.0 or higher (.NET recommended) DOWNLOAD AT:
<http://www.microsoft.com/express/vc/>

- Microsoft DirectX 9.0c Software Development Kit
DOWNLOAD AT:
<http://www.microsoft.com/downloads/details.asp>

PREREQUISITES

To enroll in this program, you need to have a reasonable familiarity with computers and a background in high school-level mathematics. No prior game or graphics programming experience are necessary.

The Video Game Design and Development program is for you if you seek a professional career as a game developer. It's also well-suited for enthusiastic amateurs and gamers looking to explore this exciting field as a recreational endeavor.

INSTRUCTOR BIO

Stan Trujillo has been a developer in the computer games programming field since 1981. During the early '90s, he spent much of his time writing DOS and Windows game class libraries, some of which would go on to become the subject of best-selling game programming books. *C++ Game Programming* and *Windows 95 Game Programming* were published in 1995, co-authored with friend and long time Dr. Dobb's C columnist, Al Stevens. The next year, Stan followed this up with two successful solo books, *Cutting-Edge Direct3D Programming* and *High Performance Windows Graphics Programming*. He went on to spend two years at Sierra Online working on game software. More recently, Stan has been working as a full-time technical consultant and freelance developer for both online and offline technology companies in the Seattle, Washington, area.

John DeGoes began writing software and designing digital logic circuits during the early 1980s. He has been actively involved in the fields of computer science, mathematics, and game development for more than fifteen years. John graduated at the top of his class in undergraduate studies in applied mathematics at Montana State University-Billings. He has authored two successful games programming books, *3D Game Programming with C++* and *3D Game Programming with C++ Gold Edition*, and he's co-authored a third book. John has also written and published articles for popular programming magazines and technical journals. He developed software for commercial programs such as "qED," "Pro Bass Fishing," "Mansion of Mystery," along with other assorted titles. Currently, John is on the road to acquiring his doctorate in applied mathematics, and he intends to research computational and mathematical methods for the study of complex dynamical systems.

Gary Simmons started games programming in 1981. He was always fascinated more by how games worked than actually playing them (although he will admit to playing too). In 1985 Gary became the host of "Video and Chips," a United Kingdom television program dedicated to exploring future computer technology. Over the years, Gary has dedicated most of his time to researching and developing cutting-edge game techniques. In May 2000, Gary founded Mr.GameMaker.com, a teaching site dedicated to helping game programmers (hobbyists and professionals alike) learn cutting-edge game programming techniques. Over the years, Gary has published dozens of full-length game development papers and tutorials. He has been teaching programs at the Game Institute for over four years and also serves as the faculty director.

Adam Hoult got his start programming in the early 1980s on the ZX-80/81. After developing several demos for the Commodore 64, Adam got a job writing software for a local company, specializing in accounting and production software. He went on to develop 3D multimedia presentations demonstrating client production systems. This experience rekindled his earlier passion for games programming. Over the years, Adam has worked on many projects, but he's focused his efforts mainly on engine and tool design for games. He spent time running a successful tools production company as well as his own game programming Web site. Adam later teamed up with fellow faculty member Gary Simmons and developed the successful Mr.GameMaker.com teaching Web site. The site gathered a loyal following as it provided information and tutorials for beginners and professionals alike on subjects that were often considered difficult to find on the Internet or in books. Adam is currently the lead technology developer at the Game Institute. He's working on some exciting upcoming 3D game projects in addition to teaching programs.

Brian Hall is currently an engine and AI programmer at Midway Amusement Games, where he's working on advanced AI algorithms for an upcoming action-adventure console title. He has also designed and written parametric airport generation software for SimAuthor Inc, as well as a real-time CLOD terrain system using real-world satellite imagery and elevation data. Previously Brian was a senior engineer at Accurate Automation Corporation, where he developed real-time learning systems for detecting pilot-induced oscillations in aircraft and intelligent path-finding for cruise-missile type applications.

David Bourg is a naval architect and marine engineer, David Bourg performs computer simulations and develops analysis tools that measure things such as hovercraft performance and the effect of waves on the motion of ships and boats. He teaches at the college level in the areas of ship design, construction, and analysis. On occasion, David also lectures at high schools on topics such as naval architecture and software development. In addition to his practical engineering background, David is professionally involved in computer game development and consulting through his company, Crescent Vision Interactive. Current projects include a massive multiplayer online role-playing game, several Java-based multiplayer games, and a couple of PC-to-Macintosh game ports. David is currently finishing his Ph.D. in engineering and applied sciences.

Frank Luna has been programming interactive 3D computer graphics for more than six years, and he's been using DirectX since its fifth iteration. He has over eight years of C++ programming experience and has also programmed in Java, C#, and MATLAB. He has worked as a contractor, with Hero Interactive, and on the open source Scorch 3D engine. In 2003 his book, *Introduction to 3D Game Programming with DirectX 9.0*, was published and became a bestseller in its genre. Frank also maintains a game and graphics development Web site, www.moon-labs.com, where he provides tutorials and sample codes on DirectX and game-related issues.