

Video Game Design and Development

500 hours

Course Overview/Description

Are you ready to take on a challenge that will always be interesting, push you to your intellectual limits, and still be fun? Video Game Design and Development is challenging, but the rewards are worth it. With our unparalleled comprehensive training program, you will master skills that open doors to the growing video game industry.

Developing and managing the complex environments for games and related visualization applications is a challenging task, but with the right training and with professional guidance, the challenge becomes much less daunting. Using a comprehensive and analytical approach to game engine architectures, this program offers students the opportunity to learn how to effectively implement game ideas. A reasonable familiarity with computers and a background in high school level mathematics are all that is assumed. No prior game or graphics programming experience are necessary.

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.

The Video Game Design and Development program is an appropriate starting point for students who seek a professional career as a game developer. It is also well-suited for enthusiastic amateurs and gamers looking to explore this exciting field as a recreational endeavor.

Course Objectives

The objectives of the Video Game Design and Development online program are:

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

Course 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 & 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 / 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



Prerequisites/Audience

A reasonable familiarity with computers and a background in high school level mathematics are all that is assumed. No prior game or graphics programming experience are necessary.

The Video Game Design and Development program is an appropriate starting point for students who seek a professional career as a game developer. It is also well-suited for enthusiastic amateurs and gamers looking to explore this exciting field as a recreational endeavor.

PC Requirements/Materials Included

This course is currently compatible with Windows Vista Operating System.

Your computer should meet the following hardware and software requirements upon registration:

Hardware Requirements

- Operating System: Microsoft Windows 98/2000/Millennium/XP
- 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>

Materials Included:

Students will be provided the required textbook, *Physics for Game Developers*.

Instructor Bio

Stan Trujillo (C++ Programming for Games) has been a developer in the computer games programming field since 1981. During the early 90's, 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 (Game Mathematics I/II) began writing software and designing digital logic circuits during the early 1980's. 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 as well as co-authoring a third. 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, where he intends to research computational and mathematical methods for the study of complex dynamical systems.

Gary Simmons (Graphics Programming I) started games programming in 1981. He was always fascinated more with how games worked than with 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 (hobbyist and professional 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 courses at the Game Institute for over four years and also serves as the Faculty Director.

Adam Hoults (Graphics Programming I) got his start programming in the early 1980's 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 has 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 website. Adam later teamed up with fellow faculty member Gary Simmons and developed the successful Mr.GameMaker.com teaching website. 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 and is working on some exciting upcoming 3D game projects in addition to teaching courses.

Brian Hall: (Artificial Intelligence) is currently an Engine and AI Programmer at Midway Amusement Games, where he is 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 Sr. Engineer at Accurate Automation Corporation where he developed real-time learning systems for detecting Pilot Induced Oscillations in aircraft, as well as intelligent path-finding for cruise-missile type applications.

David Bourg: (Physics for Game Developers) As 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 massively 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: (C++) has been programming interactive 3D computer graphics for more than six years, and has 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/graphics development website, www.moon-labs.com, where he provides tutorials and sample codes on DirectX and game related issues.