Brooklyn College
Department of Computer and Information Sciences

CISC 3660 [54.1] Game Programming
3 hours; 3 credits

Game programming techniques. 2D and 3D games. Data representations of virtual elements. Visualizing the 3D game environment. Controlling motion and behaviors. Interaction control. Game architectures, including multi-player games and message passing. Managing complexity. Teamwork to create a 3D game using a 3D multi-player game engine. Interaction with game development professionals regarding state-of-the-art hardware and software technology for game creation and adaptation.

Objectives

Computer-based games represent a rapidly growing segment of the computer software industry. For students who grew up playing computer games, game programming may seem like the ideal job. Yet in order to provide an immersive experience for the player(s), game developers must make use of the most efficient algorithms and techniques available. Game programming makes use of advances in computer graphics, human-computer interaction, visual simulation, networking (message passing), artificial intelligence, and multimedia.

- Students will learn current practices in game programming.
- Students will use skills and techniques learned in computer graphics (CIS 41) and apply them to the specific problem of game programming.
- Students will be able to design and develop their own, albeit simple, 2D and 3D games.
- Students complete two significant projects: a 2D game, and a 3D game.
- Students will gain experience in working within groups as well as individually.
- Students will learn how to document large programming projects.

Syllabus:

Week 1:	Introduction to game programming. The game programming market. Problems and issues.
Weeks 2-3:	2D game programming. Data representations of virtual elements. Efficient 2D rendering algorithms.
Weeks 4-5:	3D modeling. Polygonal models and meshes. Level of detail modeling.
Weeks 6-9:	3D rendering algorithms. The graphics pipeline. Texture mapping. Visibility processing. Lighting and shading. Shadows. Multi-pass algorithms.
Weeks 10-12:	Control of objects. Computer animation processes. Dynamic simulation. Collision detection. Interactive control. Autonomous agents.
Weeks 13-14:	Game architectures. Multi-player games and message passing. Game engine architectures. Managing complexity.

Textbook:

A. Watt and F. Policarpo, 3D Games: Volume I: Real-Time Rendering and Software Technology, Addison Wesley, 2000.

Bibliography:

A. Kirmse (ed), Game Programming Gems 4, Charles River Media, 2004.

T. Alexander (ed), Massively Multiplayer Game Development, Charles River Media, 2003.

R. Bartle, Designing Virtual Worlds, New Riders, 2003.

J. Mulligan and B. Patrovsky, Developing Online Games: An Insider's Guide, New Riders, 2003.

C. Crawford, Chris Crawford on Game Design, New Riders, 2003.

S. Rabin (ed), AI Game Programming Wisdom, Charles River Media, 2002.

D.M. Bourg, Physics for Game Developers, O'Reilly & Assoc., 2001.

Proceedings of the Game Developers' Conference