

## DEPARTMENT OF COMPUTER SCIENCE

### Course Proposal

**Title:** CSCI 475 – 3D Game Development and Computer Animation.

**Credits:** 4 s.h.

**Description:**

This course will provide students with the skills necessary to design, develop, and deploy: 3D games and related entertainment technology applications. The intent of the course is to provide the student with a solid technical foundation for developing, animating and controlling articulated systems used in interactive computer games, virtual reality simulations and high-end animation applications. Topics include: 3D game programming, 3D graphics, programming video game controllers, collision detection, force and motion calculations, networking multiplayer games, manipulating sound objects, physical modeling, projectiles, particle systems, physical constraints, deformation of virtual 3D objects, surface deformation, computer animation, forward and inverse kinematics; keyframe, motion capture and procedural animation; behavior-based animation and control; smart characters and intelligent agents.

PREREQUISITE: CS362 - Data Structures II

**Rationale:**

Interactive entertainment and computer-animated visual effects are now part of our mainstream culture. Sixty percent of all Americans older than the age of 6, or about 145 million people currently play video games. During the fiscal year, (2003-2004) for the first time in history the annual gross sales of 3D games exceeded the annual gross sales of all movie box office receipts. The total 3D game gross sales figure is expected to exceed 12 billion dollars this next fiscal year. This trend and these statistics are too large to be ignored. Currently, most computer science curricula are ill prepared to handle the needs of the entertainment technology and game development industry. Only a very small fraction of CS curricula nationally have a course in game programming. This course proposal will help to broaden the delivery of education in video game development and entertainment technologies.

Projected enrollment 25.

## OBJECTIVES AND ASSESMENT

Assessment of student learning will be conducted through conceptual diagnostic tests (CDT) through a midterm examination and a final examination, and performance assessments (PA) through weekly hands-on computer software laboratory assignments. The student who completes this course will be able to:

1. Create dynamic effects such as surface deformation, deformation of virtual objects, mass-springs-damper physics, particle systems, and smoke, fog, and fire.
2. Create humanoid body kinematics and articulated joint hierarchies in 3D animated character models,
3. Design and develop real 3D interactive games.
4. Write software to program basic physics modeling in 3D game programming.
5. Explain collision detection algorithms and modeling in 3D game programming.
6. Explain 3D graphics such as: coordinate transformations, vector and matrix computations, and scene graph architectures,
7. Create a behavioral animation for an animated character using boids with behaviors such as: swarm, flee, wander, avoid, and flocking,
8. Create animation interpolation for motion using Bezier curves or Catmull-Rom splines.

Objective	Assessment Technique
Create dynamic effects such as surface deformation, deformation of virtual objects, mass-springs-damper physics, particle systems, and smoke, fog, and fire.	Performance Assessment: Students will write software laboratory programming assignments.
Create humanoid body kinematics and articulated joint hierarchies in 3D animated character models,	Performance Assessment: Students will write software laboratory programming assignments
Design and develop real 3D interactive games.	Performance Assessment: Students will write software laboratory programming assignments
Write software to program basic physics modeling in 3D game programming	Performance Assessment: Students will write software during laboratory programming assignments.
Explain collision detection algorithms and modeling in 3D game programming.	Conceptual Diagnostic Test: Examination questions designed to evaluate the student's understanding of basic algorithms and underlying mathematics of collision detection.
Explain 3D graphics such as: coordinate transformations, vector and matrix computations, and scene graph architectures,	Conceptual Diagnostic Test: Examination questions designed to evaluate the student's understanding of coordinate transformations, vector and matrix computations, and scene graph architectures.
Create a behavioral animation for an animated character using boids with behaviors such as: swarm, flee, wander, avoid, and flocking.	Performance Assessment: Students will write software laboratory programming assignments to create a behavioral animation for an animated character using boids with behaviors such as: swarm, flee, wander, avoid, and flocking.
Create animation interpolation for motion using Bezier curves or Catmull-Rom splines.	Performance Assessment: Students will write software laboratory programming assignments to create animation interpolation for motion using Bezier curves or Catmull-Rom splines.

## Course Topics

- I. Introduction
  - a. course organization
  - b. background
  - c. motivation for course
  - d. basic concepts
  - e. terminology
  - f. demos
  
- II. History of Computer Games
  - a. successes
  - b. failures
  - c. game design principles
  
- III. Game Genre
  - a. role-playing
  - b. real-time strategy
  - c. sports
  - d. adventure
  - e. action
  - f. simulations
  - g. real-time versus turn-based games
  
- IV. Game Play Elements:
  - a. what is game play
  - b. storytelling
  - c. challenge
  - d. conflict
  - e. interactivity
  - f. relationships
  - g. supporting learning with consistent challenges and appropriate feedback
  - h. communicating with the player regarding challenges
  - i. actions and abilities within the game world
  
- V. Review of 3D Graphics:
  - a. 3D coordinate systems
  - b. linear algebra review
  - c. Vector spaces
  - d. coordinate transformations
  - e. texture mapping
  - f. 3D models
  
- VI. Using a Game Engine:
  - a. API programming
  - b. OpenGL programming
  - c. simulation techniques and modeling
  - d. camera models

- e. mouse programming
- VII. 3D objects in the scene
  - a. 3D models geometry
  - b. texture mapped objects
  - c. movement
  - d. human animation
- VIII. Collision Detection
  - a. bounding boxes collision detection
  - b. ray – plane collision,
  - c. spherical collision detection
  - d. visibility and occlusion
  - e. quad trees
  - f. BSP trees
- IX. First Person Shooter Games
  - a. camera manipulation
  - b. explosions
  - c. projectiles
  - d. bullets
- X. Sounds:
  - a. Loading sound files
  - b. playing sound files
  - c. manipulating sound objects
  - d. 3D sound
- XI. Basic Physics
  - a. Newton’s Laws of Motion
  - b. angular velocity
  - c. Inertia
- XII. Computer Animation
  - a. Keyframe animation
  - b. Motion capture animation
  - c. Procedural animation
  - d. Forward Kinematics
  - e. Inverse Kinematics
  - f. Interpolation,
    - i. Bezier curves,
    - ii. Catmull-Rom Splines
  - g. Introduction to Behavioral Animation
    - i. Boids
    - ii. Flocking,
    - iii. Fleeing,
    - iv. Avoid,
    - v. Swarm,

vi.Wander

- XIII. Networking for multiplayer computer games massive multi-player online games
  - a. client-server software
  - b. UDP vs TCP/IP
  - c. socket programming
  - d. network packet or message sending
  
- XIV. Game Design Process
  - a. player motivation
  - b. teasing the user
  - c. hidden secrets
  - d. longevity
  - e. perceptive boundaries
  - f. reward structures
  - g. reward growth, and stories
  - h. creating the right balance of obstacles/aids penalties/rewards
  
- XV. User Interface Design:
  - a. rules and boundaries,
  - b. specific hardware constraints
  - c. controllers (force feedback)
  - d. keyboards,
  - e. joysticks
  - f. headsets
  
- XVI. Simple Artificial Intelligence (AI) techniques
  - a. finite state machines in games
  - b. min-max trees
  - c. path finding
  
- XVII. Dynamic Effects
  - a. surface deformation
  - b. deformation of virtual objects
  - c. physical modeling
  - d. mass-springs-damper physics
  - e. particle systems
  - f. special effects: smoke, fog, fire

**Evaluation of Student Performance**

Students will be evaluated using an appropriate combination of tests and laboratory assignments. The following is a sample evaluation plan:

Midterm Examination	1/3
Final Examination	1/3

## Laboratory Software Assignments 1/3

Laboratory Software Assignments : There will be six software laboratory assignments each of equal worth comprising 1/3 of the total grade.

### Suggested Texts

Rick Parent, *Computer Animation: Algorithms and Techniques*, Morgan Kaufmann Publishers, San Francisco, California, 2002.

David Astle, and K. Hawkins, *Beginning OpenGL Game Programming*, Thomson Course Technology, Boston, Massachusetts, 2004.

Paul Steed, *Animating Real-Time Game Characters*, Charles River Media, Hingham, Massachusetts, 2003.

### Related Readings

James Van Verth, L. Bishop, *Essential Mathematics for Games and Interactive Applications*, Morgan Kaufmann Publishers, San Francisco, California, 2004.

David Eberly, *3D Game Engine Design: A Practical Approach to Real-Time Computer Graphics*, Morgan Kaufmann Publishers, Los Altos, California, 2004.

Jonathan Harbour, *Beginning Game Programming*, Thomson Course Technology, Boston, Massachusetts, 2004.

David Bourg, *Physics for Game Developers*, O'Reilly Press, Newton, Massachusetts, 2002.

Alan Watt and F. Policarpo, *3D Games: Volume I: Real-Time Rendering and Software Technology*, Addison-Wesley, Reading, Massachusetts, 2003.

Alan Watt and F. Policarpo, *3D Games: Volume II: Animation and Advanced Real-Time Rendering*, Addison-Wesley, Reading, Massachusetts, 2003.

Kevin Oxland, *GamePlay and Design*, Addison-Wesley, Reading, Massachusetts, 2004.

Daniel Sanchez-Crespo Dalmau, *Core Techniques and Algorithms in Game Programming*, New Riders Press, 2004.

David Eberly, *Game Physics*, Morgan Kaufmann Publishers, Los Altos, California, 2004.

Wendy Stahler, *Beginning Math and Physics for Game Programmers*, New Riders Press, 2004.

David Eberly, *Game Physics*, Morgan Kaufmann Publishers, Los Altos, California, 2004.