Computer Science Department
cs.salemstate.edu

CSC 425 Computer Graphics and Games **4 cr.**

Instructor: TBA **Office:** location **Phone:** (978) 542-extension
email: TBA@salemstate.edu **Office Hours:** days and times

Section	Time	Room	Final Exam
nn	days and times	location	Date and time

Catalog description:

This course covers fundamental principles and applications underlying computer graphics and computer games. The course presents key aspects of computer graphics including graphics pipeline, scene graphs, 2D/3D geometric objects and transformations, viewing, shading, and modeling. Topics related to computer game development include game engines, animation, and behavior and interaction. The course will also introduce basic concepts of collision detection, illumination, game design and implementation, and will emphasize the application of the topics in game-related computer graphics programming projects with the use of graphics libraries and game engines and toolkits. Four lecture hours per week, plus programming work outside of class.

Prerequisites: MAT 108 or MAT 150 or any MAT course numbered 208 or above, plus CSC 260.

Goals:

The purpose of this course is to develop students' understanding of modern computer graphics and game development. The goals of this course are:

CG01: to develop an appreciation for the modern computer graphics and game design;
CG02: to understand the basic mathematics behind computer graphics;
CG03: to understand the fundamental concepts in creating graphical contents;
CG04: to present the design issues for game development;
CG05: to give students experience in the construction of visual elements and environment (including games).

Objectives:

Upon successful completion of the course, a student will have:

CO01: demonstrated knowledge of the mathematical foundations of computer graphics;
CO02: demonstrated knowledge of the rendering pipeline in computer graphics;
CO03: described scene graphs, 2D/3D transformations, 2D/3D viewing, lighting, and clipping;
CO04: utilized at least one graphics library and at least one programming language to construct 2D/3D graphical objects and apply transformations, material properties, illuminations, and camera models to the objects to create a graphical environment;
CO05: identified texturing, shading, curves and surfaces, rasterization, and z-buffer;
CO06: become acquainted with basic game programming theory and tools, game engine, game GUI and input devices, collision detection;
CO07: demonstrated the ability to design and develop an interactive game with the application of computer graphics techniques.

Student Outcome (SO) vs. Course Objectives matrix

SO	CO01	CO02	CO03	CO04	CO05	CO06	CO07
SO-1	✓	✓	✓	✓	✓	✓	✓
SO-2				✓			✓
SO-3							✓
SO-4							✓
SO-5							✓
SO-6	✓	✓	✓	✓	✓	✓	✓

Notes:

SO-1: Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.
SO-2: Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline.
SO-3: Communicate effectively in a variety of professional contexts.
SO-4: Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.
SO-5: Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline.
SO-6: Apply computer science theory and software development fundamentals to produce computing-based solutions.

Topics:

- Introduction **GV1(0.5, 1, 0), GV2(0, 0, 2), GV4(0, 0, 0.5)**
  - graphics system (hardware and software)
  - graphics pipeline
  - rasterization
  - scene graph
  - graphics libraries and APIs
  - computer graphics programming
  - GPU
  - interactive graphics
  - mathematical fundamentals of computer graphics
- Geometric objects and transformations **GV3(0, 0, 5)**
  - scalars, points, vectors
  - graphics primitives
  - coordinate systems and frames
    - representations and N-tules
    - homogeneous coordinates
  - transformations
    - affine transformations
    - translation
    - rotation
    - scaling
    - shear
    - transformations in homogeneous coordinates
  - modeling shapes with polygonal meshes
  - concatenation of transformations
- Viewing **GV2(0, 0, 2)**
  - positioning of the camera
  - projections
    - orthogonal projections

        - perspective projections
    - clipping
    - z-buffer
- Lighting and texturing **GV2(0, 0, 3), GV3(0, 0, 2), GV4(0, 0, 1)**
    - light and matter
    - light sources
        - color sources
        - ambient light
        - point sources
        - spotlights
        - distant light sources
    - Phong reflection model
        - ambient reflection
        - diffuse reflection
        - specular reflection
        - use of Phong model in graphics API
        - translation
    - texture mapping
- Interactive graphics and games **GV5(0, 0, 8), HCI2(0, 1, 0), HCI3(0, 0, 2), HCI10(0, 0, 3), IS2(0, 2, 0), SE3(0, 1, 0)**
    - input and interaction
        - input devices
        - event-driven programming
        - picking
        - design of interactive programs
    - fundamentals of game design
        - design components and processes
        - user interfaces
        - genres of games
    - game development
        - collision detection
        - artificial intelligence in games
        - game engine patterns

Assignments: Five to eight homework assignments will be given to help understand computer graphics and games topics and their related mathematical basics. Six to eight programming projects will be given to reinforce the concepts and techniques covered in the class. Lab exercises are preparations to and part of the programming assignments.

Quizzes, Tests and Examinations: there will be two examinations: a midterm exam administered in the middle of the semester and a comprehensive final exam administered during the final exam period.

Grading: The course grade will be determined using the following approximate weights: homework, 20%; programming assignments, 55%; midterm and final exams, 25%. There will penalties for late submissions of homework and programming assignments.

Assessment Mechanism / Course Objective matrix

	Exam Questions	Homework Problems	Programming Projects (individual and group)	Programming Exercises
CO01	✓	✓	✓	✓
CO02	✓	✓	✓	✓
CO03	✓	✓	✓	✓

Assessment Mechanism / Course Objective matrix

	Exam Questions	Homework Problems	Programming Projects (individual and group)	Programming Exercises
CO04	✓	✓	✓	✓
CO05	✓	✓	✓	✓
CO06	✓	✓	✓	✓
CO07		✓	✓	✓

Bibliography:

Main texts:

Ammeraal, Leen; Zhang, Kang. **Computer Graphics for Java Programmers. Third Edition.** Springer, 2017.

Angel, Edward. **Interactive Computer Graphics: A Top-down Approach Using OpenGL. Fifth Edition**. Addison-Wesley, 2009.

Angel, Edward. **OpenGL: A Primer. Third Edition**. Addison-Wesley, 2007.

Blundell, Barry G. **An Introduction to Computer Graphics and Creative 3-D Environments**. Springer, 2008.

Davison, Andrew. **Killer Game Programming in Java**. O'Reilly Media, 2005.

Davison, Andrew. **Pro Java 6 3D Game Development: Java 3D, JOGL, JInput, and JOAL APIs**. Apress, 2007.

De Byl, Penny. **Holistic Game Development with Unity 3e: An All-in-One Guide to Implementing Game Mechanics, Art, Design and Programming.** Third Edition. CRC Press, 2019.

Godtland, Annette; Darst, Leah. **Do-It-Yourself Java Games: An Introduction to Java Computer Programming (Volume 1). Second Edition.** Create Space Independent Publishing, 2015.

Godtland, Annette. **Advanced Do-It-Yourself Java Games: An Introduction to Java Threads and Animated Video Games (Volume 3).** Create Space Independent Publishing, 2016.

Harbour, Joanthan, S. **Beginning Game Programming**. **Fourth Edition**. Cengage Learning PTR, 2014.

Hearn, Donald D.; Baker, M. Pauline. **Computer Graphics with OpenGL**. **Fourth Edition**. Pearson, 2010.

Zhang, Hong; Liang, Y. Daniel. **Computer Graphics Using Java 2D and 3D**. Pearson & Prentice Hall, 2007.

Computer Graphics/Games References:

Adam, Ernest. **Fundamentals of Game Design**. **Third Edition**. New Riders Press, 2013.

Bailey, Mike; Cunningham, Steve. **Graphics Shaders: Theory and Practice. Second Edition**. A K Peters, 2011.

Eberly, David H. **Game Physics (Interactive 3D Technology Series**). Morgan Kaufmann, 2003.

Eberly, David H. **3D Game Engine Design: A Practical Approach to Real-time Computer Graphics (The Morgan Kaufmann Series in Interactive 3D Technology**). Morgan Kaufmann, 2006.

Ericson, Christer. **Real-Time Collision Detection (The Morgan Kaufmann Series in Interactive 3D Technology**. Morgan Kaufmann, 2005.

Gordon, V. Scott; Clevenger, John L. **Computer Graphics Programming in OpenGL with JAVA. Second Edition.** Mercury Learning & Information, 2018.

Gregory, Jason. **Game Engine Architecture. Third Edition**. A K Peters, 2018.

Harbour, Joanthan, S. **Advanced 2D Game Development**. Course Technology PTR, 2008.

Harbour, Joanthan, S. **Game Programming All in One**. **Third Edition**. Course Technology PTR, 2006.

House, Donald.; Keyser, John C. **Foundations of Physically Based Modeling and Animation.** A K Peters, 2016.

Lengyel, Eric. **Mathematics for 3D Game Programming and Computer Graphics. Edition.** Cengage Learning PTR, 2011.

McShaffry, Mike. **Game Coding Complete**. **Fourth Edition**. Cengage Learning PTR. 2012.

Millington, Ian. **Game Physics Engine Development: How to Build a Robust Commercial-Grade Physics Engine for your Game. Second Edition**. CRC Press, 2010.

Nystrom, Robert. **Game Programming Patterns.** Genever Benning, 2014.

OpenInventor Group. **Open Inventor C++ Reference Manual: The Official Reference Document for Open Inventor, Release 2**. Addison-Wesley, 1994.

Palmer, Grant. **Physics for Game Programmers**. Apress, 2006.

Parent, Rick. **Computer Animation: Algorithms and Techniques. Third Edition.** Morgan Kaufmann, 2012.
Shreiner, Dave; The Khronos **OpenGL ARB Working Group. OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 3.0 and 3.1. Seventh Edition.** Addison-Wesley, 2009.
Stemkoski, Lee. **Java Game Development with LibGDX: From Beginner to Professional. Second Edition.** Apress, 2018.
Verth, James M. Van; Bishop, Lars M. **Essential Mathematics for Games and Interactive Application. Third Edition.** A K Peters, 2015.
Wernecke, Josie; Open Inventor Architecture Group. **The Inventor Mentor: Programming Object-Oriented 3D Graphics with Open Inventor, Release 2**. Addison-Wesley, 1994.
Wernecke, Josie; Open Inventor Architecture Group. **The Inventor Toolmaker: Extending Open Inventor, Release 2**. Addison-Wesley, 1994.
Wolff, David. **OpenGL 4 Shading Language Cookbook: Build high-quality, real-time 3D graphics with OpenGL 4.6, GLSL 4.6 and C++17. Third Edition.** Packt Publishing, 2018.

Academic Integrity Statement:

"Salem State University assumes that all students come to the University with serious educational intent and expects them to be mature, responsible individuals who will exhibit high standards of honesty and personal conduct in their academic life. All forms of academic dishonesty are considered to be serious offences against the University community. The University will apply sanctions when student conduct interferes with the University primary responsibility of ensuring its educational objectives." Consult the University catalog for further details on Academic Integrity Regulations and, in particular, the University definition of academic dishonesty.

The Academic Integrity Policy and Regulations can be found in the University Catalog and on the University website (http://catalog.salemstate.edu/content.php?catoid=13&navoid=1295#Academic_Integrity). The formal regulations are extensive and detailed - familiarize yourself with them if you have not previously done so. A concise summary of and direct quote from the regulations: "Materials (written or otherwise) submitted to fulfill academic requirements must represent a student's own efforts". *Submission of other's work as one's own without proper attribution is in direct violation of the University's Policy* and will be dealt with according to the University's formal Procedures. *Copying without attribution is considered cheating in an academic environment - simply put,* ***do not do it!***

University-Declared Critical Emergency Statement:

In the event of a university-declared emergency, Salem State University reserves the right to alter this course plan. Students should refer to www.salemstate.edu for further information and updates. The course attendance policy stays in effect until there is a university-declared critical emergency.

In the event of an emergency, please refer to the alternative educational plans for this course, which will be distributed via standing class communication protocols. Students should review the plans and act accordingly. Any required material that may be necessary will have been previously distributed to students electronically or will be made available as needed via email and/or Internet access.

Equal Access Statement:

"Salem State University is committed to providing equal access to the educational experience for all students in compliance with Section 504 of The Rehabilitation Act and The Americans with Disabilities Act and to providing all reasonable academic accommodations, aids and adjustments. **Any student who has a documented disability requiring an accommodation, aid or adjustment should speak with the instructor immediately.** Students with Disabilities who have not previously done so should provide documentation to and schedule an appointment with the Office for Students with Disabilities and obtain appropriate services."

Note: This syllabus represents the intended structure of the course for the semester. If changes are necessary, students will be notified in writing and via email.