ISYE/ME 4803/8803 Syllabus

Extended Reality for Engineering (XRE), 3 Credit Hours

TBD, TBD AM/PM, TBD

Instructor Information

Instructor	Email	Drop-in Hours & Location
Mohsen Moghaddam, PhD	mohsen.moghaddam@gatech.edu	TBD
Teaching Assistants	**Email**	**Drop-in Hours & Location**
TBD	TBD	TBD
TBD	TBD	TBD

General Course Information

Description

This course covers the fundamentals of the extended reality (XR) development lifecycle, with a special focus on engineering applications. Students will learn to use the Unity game engine to create XR apps compatible with various platforms. The course explores XR's affordances for human-machine interaction, immersive design visualization, virtual prototyping, training, remote operation, and real-time collaboration. Topics include overview of XR concepts, hardware, software, and industrial use cases; UX/UI design, storyboarding, bodystorming, and conceptual design; C# and object-oriented programming; Unity Editor basics, gameplay mechanics, version control, effects, animations, UI, and prototyping; VR development using XR Interaction Toolkit, covering interaction design, ergonomics, and optimization; and AR development using AR Foundation, exploring mobile and head-mounted AR, marker-based AR, plane detection, and interaction design. A semester-long project requires teams to design, develop, and test an XR apps addressing a real-world engineering problem, emphasizing hands-on experience and team-based problem-solving.

Prerequisites

1) CS 1371 or CS 1301.
2) ISYE 3770 or MATH 3670 or ECE 3077 or ISYE 3030.

Learning Outcomes

By the end of the semester, students will be able to:

- Describe fundamental concepts and principles of XR, including AR and VR hardware, software, and industrial applications, with an emphasis on engineering use cases such as human-machine interaction, immersive design visualization, virtual prototyping, industrial training, remote operation, and real-time collaboration.
- Design and prototype user-centered XR interfaces and interactions by integrating UX/UI principles, storyboarding, and immersive experience design methodologies.
- Implement interactive XR experiences by applying C# programming, object-oriented principles, and best practices in the Unity game engine.
- Develop XR apps using the XR Plugin Management system, including XR Interaction Toolkit, OpenXR Plugin, and AR Foundation, ensuring cross-platform compatibility and performance optimization.
- Assess user experience and system performance through structured testing, feedback analysis, and iterative refinement to improve usability and optimize XR apps.
- Synthesize principles of programming, user-centered design, and XR technology to design, develop, and validate a semester-long XR project that addresses a real-world engineering challenge.

Course Requirements & Grading

Grading Components*

Participation (5%)	Participation will be recorded via Canvas for each class session. Students are expected to communicate any absences to the instructor in advance and provide a legitimate reason to be excused.
Quizzes (10%)	Quizzes will be taken in class on an ad-hoc basis through Canvas. These quizzes are only available to students who attend class. Quiz topics will align with recent course material and count towards both quiz grades and participation.
Assignments (15%)	Weekly assignments will be given on programming, Unity development, and short essays. Students will occasionally deploy and test their apps on XR simulators and mobile devices, as outlined in the Course Schedule. Assignments must be submitted via Canvas and/or the course's GitHub within a one-week deadline.
Project (30%)	Students will work in groups on a semester-long project to identify and solve a real-world engineering problem using custom AR/VR apps. Milestones and deliverables are specified in the Course Schedule.
Exam (40%)	A comprehensive written final exam will be administered during finals week. The exam assesses students' understanding of course concepts, programming principles, and practical applications discussed throughout the semester.

* While the core curriculum and project structure remain consistent across both levels, **graduate students** will be expected to meet additional requirements that emphasize research integration and advanced technical competency. For the semester-long project, graduate students must conduct a literature review of at least five peer-reviewed sources, build their application to advance the identified research topics, fully develop the app, and prepare a formal human subjects study design to validate their system. Weekly assignments will include added research-oriented tasks focused on identifying open research questions in XR for engineering and proposing strategies to address them, supported by relevant literature. On the final exam, graduate students will complete additional advanced questions covering C# programming and XR development in Unity.

Grading Scale

A: 90–100%, B: 80–89%, C: 70–79%, D: 60–69%, F: 0–59%.

Course Materials

Course Text

GitHub: https://github.gatech.edu/mmoghaddam3/XRE/wiki.

Recommended Resources

- Norman, D. (2013). *The Design of Everyday Things: Revised & Expanded Edition*. Basic Books.
- Dix, A., Finlay, J., Abowd, G. D., Beale, R. (2004). *Human-Computer Interaction* (3rd Ed). Harlow, England; New York: Pearson / Prentice-Hall.
- Rogers, Y., Sharp, H., Preece, J. (2023). *Interaction Design: Beyond Human-Computer Interaction* (6th Ed). John Wiley & Sons, Inc., Hoboken, New Jersey.
- Microsoft Learn, https://learn.microsoft.com/en-us/training.
- Unity Learn, https://learn.unity.com.

Software

- Unity Hub, https://unity.com/download.
- Visual Studio Code, https://code.visualstudio.com/download.
- X Code (for Mac XR Dev Only), https://developer.apple.com/xcode.

Course Policies, Expectations, & Guidelines

Academic Integrity

Georgia Tech aims to cultivate a community based on trust, academic integrity, and honor. Students are expected to act according to the highest ethical standards. Review Georgia Tech's Honor Code (https://catalog.gatech.edu/policies/honor-code) and Code of Conduct (https://catalog.gatech.edu/rules/18). Any student suspected of cheating or plagiarizing on a quiz, exam, or assignment will be reported to the Office of Student Integrity, who will investigate the incident and identify the appropriate penalty for violations.

Accommodations for Students with Disabilities

If you are a student with learning needs that require special accommodation, contact the Office of Disability Services (http://disabilityservices.gatech.edu, 404-894-2563) as soon as possible to make an appointment to discuss your special needs and to obtain an accommodations letter. Please also e-mail me as soon as possible to set up a time to discuss your learning needs.

Participation

Participation is evaluated based on multiple criteria, including punctuality, active participation in class activities, respectful listening during discussions, and maintaining full engagement in learning by avoiding distractions such as texting, checking your phone or email, or using other digital devices inappropriately. Students should recognize that their active participation not only enhances their own learning but also contributes to the success of their classmates in mastering the material. Students are expected to communicate any absences to the instructor in advance and provide a legitimate reason to be excused. Review Georgia Tech's Attendance Rule (https://catalog.gatech.edu/rules/4) for further information.

Assignments

Assignments will be posted on Canvas. Submissions must be made individually via Canvas and/or the course's GitHub within a one-week deadline. Each student is required to submit their own homework assignments on time. While discussing homework problems with classmates is encouraged, copying is strictly prohibited. Assignments will be reviewed for excessive similarities, and any violations will result in a score of zero for the assignment. Late submissions are accepted with a 20% penalty per day.

Project

A semester-long, hands-on project will be assigned to small groups, with group assignments made at the beginning of the semester. It is each student's responsibility to attend group meetings, contribute actively to the project, and notify the instructor early in the semester if they are unable to attend. The project will include a report, a final presentation, and a demonstration in classroom during the final week of the semester. Each group will identify a real-world engineering challenge that can be addressed using XR, propose their idea and planned activities by the second week of the semester, and work toward achieving their objectives under the guidance of the instructor and teaching assistants. The project grade will include an anonymous peer evaluation, which will account for 25% of the final score, as assessed by other group members.

Exam

A comprehensive final exam will be administered during finals week, covering all course material. The exam will be in a closed-book format and will focus on key concepts and a general understanding of the material. A review session will be held during the last week of classes. Cheating of any kind, including copying another student's work, constitutes a direct violation of the Georgia Tech Academic Honor Code and will be addressed in accordance with Georgia Tech policy.

Student-Faculty Expectations Agreement

At Georgia Tech, we believe that it is important to strive for an atmosphere of mutual respect, acknowledgement, and responsibility between faculty and the student body. The Student-Faculty Expectations Rules (https://catalog.gatech.edu/rules/22) articulate some basic expectations that you can have of the instructor

and vice versa. In the end, simple respect for knowledge, hard work, and cordial interactions will help build the environment we seek. Therefore, students are encouraged to remain committed to the ideals of Georgia Tech while in this class.

Use of AI

Students may use generative AI tools, such as ChatGPT and VS Code Copilot, to assist with coding and content creation. These tools should supplement, not replace, the learning process. Students must critically review and adapt AI-generated outputs to demonstrate their understanding and align with course objectives. Over-reliance without comprehension may violate academic expectations and result in penalties. Proper acknowledgment of AI assistance is encouraged when applicable.

Undergraduate Student Academic Success Resources

Students looking for additional assistance outside of the classroom are advised to consider working with a peer tutor through Knack. Georgia Institute of Technology has partnered with Knack to provide students with access to verified peer tutors who have previously aced this course. Visit gatech.joinknack.com and sign in with your student account to view available tutors.

Graduate Student Academic and Professional Success Resources

A list of resources for graduate students is given on the Office of Graduate and Postdoctoral Education website (https://gradpostdoc.gatech.edu). Information for current students (https://grad.gatech.edu/current-students) include:

- Academic Resources (https://grad.gatech.edu/academic-resources), such as the Communications Center, Language Institute, Library, Catalog, Registrar, resources for conducting research, Advocacy and Conflict Resolution resources;
- Student Resources (https://grad.gatech.edu/student-resources), such as Campus Services, Child Care/Family programs, Health & Wellness, Career Services, and the Student Resource Guide; and
- Professional Development (https://grad.gatech.edu/career-development), such as the programming from the Career Center and other professional development resources and events.

Student Well-Being

At Georgia Tech, we are concerned about your overall physical, social, and mental well-being. A comprehensive list of wellness related resources has been compiled and maintained by the Office of the Vice President for Student Engagement and Well-being (https://students.gatech.edu/student-resource-guide).

Course Schedule

Module A: Foundations of XR for Engineering

A1. Advancing Engineering with XR
Introduces the spectrum of XR (VR, AR, MR), their roles in industry, hardware and software ecosystems, and real-world case studies from sectors like manufacturing, aerospace, healthcare, and retail. Covers the XR technology stack including spatial computing, rendering engines, and sensors.

A2. Architecting XR Experiences
Focuses on designing for XR users with principles from Norman's design to XR-specific ergonomics, spatial layout, interaction, complexity and occlusion management. Introduces prototyping, usability testing, and best practices in iterative design.

Module B: Unity for XR Development

B1. Building Real-Time 3D Apps
Covers Unity fundamentals, including installing Unity Hub, setting up projects, and navigating the Unity Editor. Explores scene management, GameObjects, prefabs, and basic Unity panels like Scene, Hierarchy, Project, and Inspector for creating structured interactive 3D environments.

B2. Components of GameObjects
Introduces the component-based architecture of Unity, covering transforms, renderers, lighting, physics, animation systems, audio sources, UI elements, particle systems, and custom scripting.

B3. Physics & Animation
Explains Unity's physics engine, covering rigidbody dynamics, collision detection, joints, and physics materials. Introduces animation workflows including keyframe animation, Animator Controllers, and importing and refining animations for interactive GameObjects.

B4. Audio & UI Prototyping
Covers Unity's audio system including spatial audio, audio mixers, reverb zones, and prioritization of sounds. Explores UI design elements such as canvas scaling, layout systems, interactive buttons, sliders, toggles, and best practices for responsive XR interfaces.

Module C: C# Programming for XR

C1. C# for Interactive XR
Introduces C# programming for Unity, covering IDE setup, variables, data types, string manipulation, mathematical operations, data conversion, and Unity-specific data handling methods.

C2. Decision Logic & Methods
Covers control structures including if statements, switch-case logic, loops, modular programming using methods with parameters and return values, and best practices for code reuse and execution flow.

C3. Collections & OOP Patterns
Explores C# collections such as arrays, lists, and dictionaries. Introduces object-oriented programming (OOP) concepts like classes, inheritance, polymorphism, encapsulation, and abstraction.

C4. MonoBehaviour & Namespaces
Covers Unity's scripting lifecycle with MonoBehaviour methods like Start, Update, and event-driven methods. Explores namespaces relevant to Unity, XR, UI, and audio systems.

Module D: Developing Virtual Worlds in VR

D1. Developing Virtual Worlds in VR
Explains the principles of VR, its engineering use cases, and challenges such as comfort, safety, and performance optimization. Provides an overview of the VR development lifecycle.

D2. VR Technology Essentials
Details the VR development process in Unity including scene setup, XR Toolkit configuration, OpenXR integration, input systems, XR device simulation, and deployment to VR hardware.

D3. Navigating in VR
Covers coordinate systems, spatial awareness, user tracking, and locomotion systems including snap turn, teleportation, and continuous movement using Unity's XR rig.

D4. Grabbing Objects in VR
Introduces object manipulation in VR using XR Grab Interactable, covering configuration, interaction refinement, grip points, and performance considerations.

D5. Attaching Objects in VR
Explores socket-based interaction using XR Socket Interactor, covering setup, interaction layers, and reusable socket configurations for object placement and attachment.

D6. Interacting with Objects in VR
Covers activation events, direct and ray-based interactors, and how to apply these interactions for manipulating and activating objects within VR environments.

D7. Hearing and Sensing in VR
Introduces multimodal feedback through haptics and spatial audio, covering implementation of event-driven sounds, 3D audio positioning, and reverb zones for immersive VR experiences.

D8. Designing UI in VR
Covers the creation of VR-specific user interfaces using Unity's canvas system, text elements, interactive buttons, and XR Toolkit components with a focus on usability and accessibility.

D9. Enhancing VR Experiences
Addresses VR comfort features like fade transitions and snap turning and covers performance optimization techniques such as draw call reduction, polycount management, and post-processing.

Module E: Augmenting the Physical World with AR

E1. Enriching Physical World with AR
Introduces AR concepts, hardware and software tools including AR SDKs, spatial mapping, and game engines. Discusses AR's value for immersive augmentation in engineering and its limitations.

E2. AR Foundation
Covers setting up AR Foundation in Unity, deploying to Android/iOS, and managing AR subsystems like session management and tracking.

E3. World Understanding
Explains plane detection, point clouds, meshing, occlusion, and environment probes, covering their implementation and performance considerations in AR experiences.

E4. Image Tracking
Covers setting up reference image libraries, configuring image tracking managers, and building AR experiences that respond to recognized images with dynamic content placement.

E5. Object Tracking
Introduces AR object tracking, anchors, and bounding boxes. Covers AR anchor manager configuration for persistent object placement and management of anchored content.

E6. User Tracking
Covers face, body, and multi-user tracking using AR Foundation's manager for face tracing and body tracking features, with example use cases for interactive avatars and shared AR spaces.

E7. Spatial AR (Part 1)
Focuses on spatial AR development for Magic Leap 2 and similar platforms, covering plane detection, meshing, occlusion, spatial audio, and in-context testing for advanced spatial experiences.

E8. Spatial AR (Part 2)
Covers advanced interaction systems including spatial anchors, controller inputs, voice commands, hand tracking, and eye tracking for fully immersive spatial AR applications.

XRE Showcase: Project Presentations and Demos

Project presentations and demonstrations in class, participation in peer review of other projects and own teammates, and submission of project report, including design, development process, and implementation highlights.