CSYE 7270
Building Virtual Environments
Course Syllabus

Course Information

Professor: Nik Bear Brown
Email: ni.brown@neu.edu
Office: 505A Dana Hall
Office hours:
Through Zoom by Appointment

Note: I am also a master's student at Northeastern. Do not send e-mail to my student e-mail brown.ni@husky.neu.edu as I almost never read that e-mail.

All classes will be held via Zoom through Canvas

Course website: Canvas

Course Prerequisites

Programming in some language like python, Java or C++.

Course Description

The course delves deep into the realm of real-time programming and design, aiming to impart skills for crafting captivating interactive experiences and constructing immersive virtual environments. Rooted in practicality, students will acquire the know-how to develop interactive real-time applications across a spectrum of domains including, but not limited to, virtual and augmented reality, games, data visualization, medicine, industrial design, film, and the burgeoning field of generative AI in media.

A pivotal component of the curriculum focuses on real-time 3D rendering engines and the intricate subsystems that breathe life into both 2D and 3D realms. Learners will gain insights into various facets encompassing rendering, animation, collision mechanics, physics, audio intricacies, trigger systems, intricate shading, control logic, the nuances of behavior trees, and the rudiments of artificial intelligence.

The course not only outlines the underpinnings of real-time graphics engines but also equips students with the skills to harness libraries, such as physics and graphics libraries, tailoring them to forge interactive real-time virtual environments. Key discussions revolve around the graphics pipeline, scene graph intricacies, adept level design, behavior scripting, the tenets of object-oriented game design, analytics, and the versatile scripting languages.

Moreover, in a distinctive module, the course will unravel the potential of generative AI in the realms of character design, storyboarding, and animation. Participants will be introduced to how AI can be employed to generate innovative character archetypes, envision dynamic storyboards, and even automate intricate animation sequences, pushing the boundaries of traditional media production.

By the culmination of this course, students will not only possess a profound understanding of real-time design mechanics but also an appreciation for the transformative role of AI in shaping the future of interactive media and storytelling.

Learning Objectives

- In this course, students will immerse themselves in the dynamic realm of Building Virtual Environments, making use of cutting-edge platforms like Unreal Engine and Unity 3D. The course will unravel the complex tapestry of systems that give life to 2D and 3D worlds, spanning rendering, animation, collisions, physics, and a host of other critical components. A deep dive into the operational mechanics of game and graphic engines will empower students to harness libraries tailored for physics and graphics, aiding in the crafting of virtual realms. In-depth discussions encompassing graphics pipeline, scene graph, the art of level design, the nuances of behavior scripting, the essence of object-oriented game design, the utility of world editors, and the vast world of scripting languages will be undertaken.
- Additionally, students will touch upon the modern realms of servers and delve into the nuances of mobile programming, focusing on platforms like Android and iPhone.
- Throughout the course journey:
- Students will engage in weekly assignments, spanning the development of compact games or tools, intricate 3D simulations, or virtual environments.
- Biweekly in-class progress presentations will keep the learning momentum high.
- The curriculum will entail both a group project and an individual project for a holistic learning experience.
- The curriculum's first half will pull students into the practical dimensions of programming for games, virtual environments, and real-time 3D in platforms like Unreal Engine or Unity 3D. The essence lies in "hands-on" experiential learning, fostering skills through the creation of a series of micro-projects. Students will begin with a rudimentary skeleton and then progressively enhance and extend this foundational structure with each assignment.
- As we transition into the course's latter half, the focal points will shift towards social/network gaming, the enchanting domains of virtual reality, and the transformative realm of augmented reality programming.
- The overarching learning objectives for this course are to:
- Master programming in both C++ and C# with clarity, robustness, efficiency, and safety.
- Understand game engine rendering intricacies.
- Dive deep into the world of animation, including generative AI-driven character design, storyboarding, and advanced animation techniques.
- Grasp game engine collision mechanics.
- Navigate the physics of game engines.
- Comprehend game engine audio systems.
- Delve into game engine trigger mechanisms.

- Explore the art of game engine shaders.
- Embrace augmented reality's transformative potential.
- Venture into the immersive realms of virtual reality.
- Engage with social/network gaming dynamics.
- Explore mobile gaming's nuances.
- Understand game servers' foundational principles.
- Dive into the basics of Game AI.
- Familiarize oneself with basic Game Analytics.

Weekly Schedule

Week 1
Game Design Document or GDD
Generative AI for Game Conceptualization and Content Ideas

Week 2
Unity Game Engine Basics
Generative AI for Character Style

Week 3
Unity Game Engine Basics (Advanced)
Generative AI for Character Design

Week 4
Unreal Engine Basics
Generative AI for background generation

Week 5
Unreal Engine Basics (Advanced)
Generative AI for 3D modeling

Week 6
Shaders & Materials
Generative AI for Shader and Material Design

Week 7
Advanced Shaders & Materials
Generative AI for Storybaords

Week 8
Shaders & Materials (Applied Techniques)
Generative AI for Adaptive Shader Creation

Week 9
Particle Effects

Generative AI for Procedural Particle Effect Generation

Week 10
Animation
Generative AI for Character Animation and Motion Synthesis

Week 11
Advanced Animation Techniques
Generative AI for Sprite Based-Animation

Week 12
Audio
Generative AI for Music Composition and Voice Over Synthesis

Week 13
Profiling/Optimization
Generative AI-driven Techniques for Dynamic Content Generation

Week 14
Game AI Basics
Generative AI for Game AI

Week 15
Final Projects
Generative AI for creating a Game Trailer

Course Software

For games and real-time 3D students will their choice of one of two game engines Unreal Engine or Unity 3D

If your choice is Unreal Engine, you need to download and install the free and open-source Unreal Engine https://www.unrealengine.com/en-US/

If your choice is Unity 3D you need to download and install the free and closed-source Unity 3D https://store.unity.com/download

We will cover both extensively the first couple of weeks of class so students understand the differences and make the right choice for them.

For VFX all students will be using the free version of SideFX: Houdini https://www.sidefx.com/

While optional the free the free and open-source 3D creation suite Blender would be helpful for all students. https://www.blender.org/download/

Course GitHub

The course GitHub (for all lectures, assignments and projects):

https://github.com/nikbearbrown/CSYE_7270

nikbearbrown YouTube channel

Over the course of the semester I'll be making and putting additional data science and machine learning related video's on my YouTube channel.

https://www.youtube.com/user/nikbearbrown

The purpose of these videos is to put additional advanced content as well as supplemental content to provide additional coverage of the material in the course. Suggestions for topics for additional videos are always welcome.

Teaching assistants

The Teaching assistants are:

TBA

Programming questions should first go to the TA's. If they can't answer them then the TA's will forward the questions to the Professor.

Learning Assessment

Achievement of learning outcomes will be assessed and graded through:

- Quizzes
- Exams
- Completion of assignments
- Completion of term projects

Reaching out for help

A student can always reach out for help to the Professor, Nik Bear Brown ni.brown@neu.edu. In an online course, it's important that a student reaches out early should he/she run into any issues.

Grading Policies

A point system is used. Everything that you are expected to turn in has points. Points can range from 1 point to 1000 points. For example, every class you are expected to make class notes and upload them by 11:59PM the day of the lecture and that is worth 5 points. A quiz can range from 25 to 100 points and an exam might be 250 points. Assignments that are worth 50 points or less get a 50% deduction for each day they are late rounded up. For example, a late 5 point class notes would get 3 points (2.5 rounded to 3). Assignments more than 50 points get a 10% deduction for each day they are late rounded up. Exams cannot be made up unless arraignments are made before the exam.

I expect to use the following grading scale at the end of the semester. You should not expect a curve to be applied; but I reserve the right to use one.

Score	Grade
93 – 100	A
90 – 92	A-
88 – 89	B+
83 – 87	B
80 – 82	B-
78 – 79	C+
73 – 77	C
70 – 72	C-
60 – 69	D
<60	F

Scores in-between grades. For example, 82.5 or 92.3 will be decided based on the exams.

* Note the score is calculated using the grading rubric and IS NOT the average of the assignments that is displayed by Canvas.

There will be two major projects due near the end of the semester worth approximately 50% of the total grade. A VFX assignment in Houdini and a game/real-time 3D assignment in either Unreal Engine or Unity 3D.

Canvas

You will submit your assignments via Canvas *and* Github. Click the title of assignment (Canvas -> assignment -> <Title of Assignment>), to go to the submission page. You will know your score on an assignment, project or test via Canvas. Canvas only represents only the raw scores. Not normalized or curved grades. A jupyter notebook file ALONG with either a .DOC or .PDF rendering of that jupyter notebook file must be submitted with each assignment.

Your name MUST be part of your submission, for example Sanchez_Rick_Assignment_1.zip

Multiple files must be zipped. No .RAR, .bz, .7z or other extensions.

Assignment file names MUST start with students last name then first name OR the groups name and include the class number and assignment number.

Assignment MUST estimate the percentage of code written by the student and that which came from external sources.

Assignment MUST specify a license at the bottom of each notebook turned in.

All code must adhere to a style guide and state which guide was used.

Due dates

Due dates for assignments at midnight on due date of the assignment.

Five percent (i.e. 5%) is deducted for each day an assignment is late. Solutions will be posted the following Monday. Assignments will receive NO CREDIT if submitted after the solutions are posted. Any extensions MUST be granted via e-mail and with a specific new due date.

Course Materials

Computational Art Lab

The Computational Art Lab stands as a testament to the future of game design, where the worlds of traditional development methodologies and innovative Generative AI techniques converge. Here, students are provided a platform to harness computation, merging it seamlessly with artistry to create stunning, dynamic content for games.

Delving into the lab, students embark on a multifaceted journey that intertwines the power of renowned engines like Unity and Unreal with the expansive capabilities of Generative AI. As they master the essentials of game engines, they simultaneously explore how Generative AI can revolutionize character style, depth of design, and the creation of evocative backdrops. This layered approach empowers them to craft vibrant in-game worlds, rich with intricate objects, scenes, and personas. Beyond the static visuals, the lab deepens the experience with a focus on motion. Students are introduced to the intricate dance of light through shaders, materials, and advanced animation techniques, all while leveraging AI to synthesize lifelike movements, adaptive shaders, and procedural particle effects.

The lab doesn't just stop at visual splendor; it ventures into the auditory realm, teaching students the art of game audio. Here, Generative AI becomes an indispensable ally, assisting in music composition and voice-over synthesis to capture the perfect mood and tone for every scenario. Additionally, the platform pushes boundaries by delving into game profiling, optimization, and AI-driven dynamics, ensuring students grasp the critical elements that promise smooth and immersive gameplay experiences. By the culmination of their journey, students are not merely passive learners but trailblazing creators, adept at harnessing the potential of Generative AI to usher in a new era of game design and interactive art.

ML-Agents

The Unity Machine Learning Agents Toolkit (ML-Agents) is an open-source Unity plugin that enables games and simulations to serve as environments for training intelligent agents. Agents can be trained using reinforcement learning, imitation learning, neuro-evolution, or other machine learning methods through a simple-to-use Python API. We also provide implementations (based on TensorFlow) of state-of-the-art algorithms to enable game developers and hobbyists to easily train intelligent agents for 2D, 3D and VR/AR games. These trained agents can be used for multiple purposes, including controlling NPC behavior (in a variety of settings such as multi-agent and adversarial), automated testing of game builds and evaluating different game design decisions pre-release. The ML-Agents toolkit is mutually beneficial for both game developers and AI researchers as it provides a central platform where advances in AI can be evaluated on

ML-Agents

The Unity Machine Learning Agents Toolkit (ML-Agents) is an open-source Unity plugin that enables games and simulations to serve as environments for training intelligent agents. Agents can be trained using reinforcement learning, imitation learning, neuro-evolution, or other machine learning methods through a simple-to-use Python API. We also provide implementations (based on TensorFlow) of state-of-the-art algorithms to enable game developers and hobbyists to easily train intelligent agents for 2D, 3D and VR/AR games. These trained agents can be used for multiple purposes, including controlling NPC behavior (in a variety of settings such as multi-agent and adversarial), automated testing of game builds and evaluating different game design decisions pre-release. The ML-Agents toolkit is mutually beneficial for both game developers and AI researchers as it provides a central platform where advances in AI can be evaluated on Unity's rich environments and then made accessible to the wider research and game developer communities.

https://github.com/Unity-Technologies/ml-agents

Most textbooks related to Unity 3D and Unreal Engine are all available for free to NEU students via SpringerLink (http://link.springer.com/). the there are no ***required*** textbooks.

Further Unity 3D and Unreal Engine have many tutorials on their websites. The class is very hands-on and we will be extending and adapting the Unity 3D and Unreal Engine learning materials.

Participation Policy

Participation in discussions is an important aspect on the class. It is important that both students and instructional staff help foster an environment in which students feel safe asking questions, posing their opinions, and sharing their work for critique. If at any time you feel this environment is being threatened—by other students, the TA, or the professor—speak up and make your concerns heard. If you feel uncomfortable broaching this topic with the professor, you should feel free to voice your concerns to the Dean's office.

Collaboration Policies

Students are strongly encouraged to collaborate through discussing strategies for completing assignments, talking about the readings before class, and studying for the exams. However, all work that you turn in to me with your name on it must be in your own words or coded in your own style. Directly copied code or text from any other source MUST be cited. In any case, you must write up your solutions, in your own words. Furthermore, if you did collaborate on any problem, you must clearly list all of the collaborators in your submission. Handing in the same work for more than one course without explicit permission is forbidden.

Feel free to discuss general strategies, but any written work or code should be your own, in your own words/style. If you have collaborated on ideas leading up to the final solution, give each other credit on what you turn in, clearly labeling who contributed what ideas. Individuals should be able to explain the function of every aspect of group-produced work. Not understanding what plagiarism is does not constitute an excuse for committing it. You should familiarize yourself with the University's policies on academic dishonesty at the beginning of the semester. If you have any doubts whatsoever about whether you are breaking the rules – ask!

Any submitted work violating the collaboration policies WILL BE GIVEN A ZERO even if "by mistake." Multiple mistakes *will be sent to OSCCR for disciplinary review.*

To reiterate: **plagiarism and cheating are strictly forbidden. No excuses, no exceptions.** *All incidents of plagiarism and cheating will be sent to OSCCR for disciplinary review.*

Assignment Late Policy

Assignments are due by 11:59pm on the due date marked on the schedule. It is your responsibility to determine whether or not it is worth spending the extra time on an assignment vs. turning in incomplete work for partial credit without penalty. Any exceptions to this policy (e.g. long-term illness or family emergencies) must be approved by the professor.

Assignments will receive NO CREDIT if submitted after the solutions are posted. Any extensions MUST be granted via e-mail and with a specific new due date.

Only ONE extension will be granted per semester.

Student Resources

Special Accommodations/ADA: In accordance with the Americans with Disabilities Act (ADA 1990), Northeastern University seeks to provide equal access to its programs, services, and activities. If you will need accommodations in this class, please contact the Disability Resource Center (www.northeastern.edu/drc/) *as soon as possible* to make appropriate arrangements, and please provide the course instructors with any necessary documentation. The University requires that you provide documentation of your disabilities to the DRC so that they may identify what accommodations are required, and arrange with the instructor to provide those on your behalf, as needed.

Academic Integrity: All students must adhere to the university's Academic Integrity Policy, which can be found on the website of the Office of Student Conduct and Conflict Resolution (OSCCR), at http://www.northeastern.edu/osccr/academicintegrity/index.html. Please be particularly aware of the policy regarding plagiarism. As you probably know, plagiarism involves *representing anyone else's words or ideas as your own*. It doesn't matter where you got these ideas—from a book, on the web, from a fellow-student, from your mother. It doesn't matter whether you quote the source directly or paraphrase it; if you are not the originator of the words or ideas, *you must state clearly and specifically where they came from*. Please consult an instructor if you have any confusion or concerns when preparing any of the assignments so that together. You can also consult the guide "Avoiding Plagiarism" on the NU Library Website at http://www.lib.neu.edu/online_research/help/avoiding_plagiarism/. If an academic integrity concern arises, one of the instructors will speak with you about it; if the discussion does not resolve the concern, we will refer the matter to OSCCR.

Writing Center: The Northeastern University Writing Center, housed in the Department of English within the College of Social Sciences and Humanities, is open to any member of the Northeastern community and exists to help any level writer, from any academic discipline, become a better writer. You can book face-to-face, online, or same day appointments in two locations: 412 Holmes Hall and 136 Snell Library (behind Argo Tea). For more information or to book an appointment, please visit http://www.northeastern.edu/writingcenter/.