CSE 40439 Game Development

Prerequisites

- CSE 20312
- CSE 20289
- CSE 30332

Where/When

Room TBD

T/Th, 2:00 – 3:15 PM

Instructor

Dan Rehberg

Office:

Office Hours:

Email:

Teaching Assistant(s)

TBD

Office Hours: TBD

Description:

This course introduces topics within game development to have students prototype, iterate, and present a final semester game project to their peers. These topics include emphasizing applied vector mathematics for all parts of game creation (interaction, audio, mechanics, and graphics), discussing the software behind game engines, numerical methods and coherency over discrete time, UX/UI/accessibility, and some advanced concepts in systems/software and performance/accuracy tradeoffs.

The course will rely on the Godot Engine as well as the Unofficial OpenGL Math library for software development. Dev logs will rely on Obsidian notes. Students will be expected to plan their workflow, track and log jobs, and play a role as both an observer and tester for other student projects as they design and prototype their own games. The course should present new perspectives on the power of linear algebra, the importance of game testing, and how a systems perspective is important to performance and user experience.

Topics:

Week 1

- Software and tools overview
- Linear Algebra and Geometric analysis with vectors

- Dot products, projections, matrices, and quaternions

Week 2

- Discrete-event simulation and coherency over discrete time
- Input handling and interactive events

Week 3

- Programmable graphics pipeline (w/PBR) and coherency over discrete space
- Building assets, programmer art, shaders, and tangent space

Week 4

- Finite Automata for Simple Reflex Agent and Parametric Paths with Interpolation
- AI complexity and FSM vs ML vs NPC freedom

Week 5

- Rapid prototyping and testing with game engines
- Outlining new gameplay features and mechanics

Week 6

- Art vs Engineering and target audience, marketing, safety, UI/UX
- Tropes vs Uniqueness in gameplay and visuals

Week 7

- Interactive audio and history
- Monaural sound manipulation for object-based rendering, volumetric audio, ambisonics, spatial sounds and FX, and foley design

Week 8 (Resuming after break)

- Kinematics, armatures, and numerical methods
- Accuracy and stability of integration methods, input-driven animations, and physically-based animations

Week 9

- Advanced texture effects, varying coordinate spaces (e.g., SSAO), and direct use of normalized device coordinates
- Screen-space UI, reflections/redirections, and scrolling textures

Week 10

- Interactive physics simulations and collision detection
- Discrete vs Continuous, historical boundary conditions (pixel-per axis), and issues with impulse-based dynamics

Week 11

- Interactive CG with shaders and particle FX
- Heightmaps and tessellation, texture splatting, geometry shaders and fur, imposters, and partial-differential equations/vector fields

Week 12

- Framerate and frame times, asynchronous discretized time
- Issues with JIT shader handling, instanced rendering, and render state changes

Week 13

- Scene graphs and advanced graphics techniques (e.g., parallax mapping)
- 3D CG processing vs Screen-space FX, deferred and forward rendering, ray tracing

Week 14-15

- Final presentations

Week 16

- Final

The scheduled topics might be shuffled if some topics are found to be more useful earlier in the course.

Grading — Points

- Obsidian development logs - Total 100
  - Interstate game — 50
  - Course project — 50
- Q/A Testing - Total 100
  - Tester — 50
  - Observer — 50
- Revisions of course project - Total 200
- Quizzes - Total 100
- Interstate game programming - Total 100
- Project presentation - Total 200
- Exams - Total 200

The development logs (dev logs) will persist throughout the course and are judged based on organization, planning and execution, and accountability of tasks within the course project and example game assignments. Q/A testing is expected to occur at least 3 times in the semester, and will require reiterating/revising your project(s) based on tester feedback. Quizzes are open to reattempts until they close (on their respective

deadlines) following a “low stakes” quizzing model (https://eric.ed.gov/?id=ED611620). This practice has been shown to increase learning of new material by providing active memory rehearsals sessions. Please feel free to take quizzes when they open to make use of this rehearsal process.

Course Project Conditions

Your course project is allowed to be built in another game engine not explicitly used in the course. However, the same rules apply with generative AI – for example, if you choose another engine that utilizes generative AI trained on internet assets to (for instance) automate the creation of textures, then you can not use that utility OR the entire engine if it is a baseline feature. Your game must also be playable on department machines in order to ensure it can be tested by other students, TA(s), or myself.

Late Policy

Late submissions will only be accepted if you reach out to me in advance (more than a week), or given an exception allotted by the University. Exams are less applicable to being made up without an explicit University exception.

A one-time, 3 day late submission is available for any assignment that is not due at a semester checkpoint. That is, two primary assignment dates cannot be changed, which are for the midterm assessment of your term project, and a final exam as well as the final assessment for your term project. Otherwise, email me for a ***no-questions-asked*** *3-day extension to a pending assignment.*

Honor Code

You are expected to abide by the Notre Dame University Undergraduate Academic Code of Honor (https://honorcode.nd.edu/). This extends to generative AI policies added to the Code of Honor.

Game development requires conceptualizing certain topics to understand how to build art assets and write code. You are welcome to collaborate with peers to illustrate ideas either verbally or by drawing diagrams or geometric concepts. However, you are not allowed to share art assets or code for your assignments and course project. *If, for instance, you solved a problem that someone else is facing, then you may share the process of thinking through that problem but not by handing them final code.* The reasoning follows two arguments. First, in game development production you will have the opportunity to read manuals or cutting-edge research to understand how to solve

problems, but these will typically be conceptual and require *you* to build a solution. Secondly, even with things like open-source code, the way the code runs dictates how a user experience and game feel will be provided – meaning the specific implementations of a programming solution (in the infinite and non-unique solutions of a Turing model) dictate how a game will look and "feel."

For these reasons, only conceptual information is allowed to be shared and not assets. The exception to this is any code provided during lectures. ***You are not allowed to use generative AI to create your arts assets or code, this issue relates to intellectual property and the resources used to train AI models.***

Attendance

General attendance is expected in this course. We cover conceptual information in class that should be relevant to you working on the assignments and developing your term project. Of course, life-happens and crunch can be real. Please try to take breaks between work, and if you have an emergency situation or University exemption event it is understandable to miss class.

Content Warning

This is a University setting, so expressive freedom is allowed, but for content you make in your games, please refrain from building anything that would be banned on Steam (https://steamcommunity.com/games/593110/announcements/detail/1666776116200553082). ***I will present content that may contain adult themes, violence, or gore;*** **so please let me know in advance if there are any concerns about such content.** For your own games, be considerate of people and provide warnings about violence, gore, etc.. when having others test your game. Consent is required from your play testers to be observed during a Q/A session. If you have any questions about your game contents, do not hesitate to contact me.

Syllabus Revisions

The schedule might get rearranged, and content might be dropped in favor of accelerating your time to work on your semester projects. If changes occur, the syllabus version will be updated and if the revision was major (i.e., not just a spelling/formatting), then a Canvas announcement will be made noting that this document has been modified.