3D Game Programming

Instructor : DJ Kehoe
Course : IT-466-001
Email: kehoed@njit.edu
Office: GITC 3200
Office Hours: TWR: 1:00 - 2:00

Academic Integrity

The NJIT Honor Code will be upheld at all times. The work you do and submit is expected to be the result of your effort only. You may discuss the high level (general) solution of a design problem, however, cooperation should not result in one or more students having possession of copied graphics, code, or any other project element created by another student. Any violations of the NJIT Honor code will be brought to the immediate attention of the Dean of Students.
The latest version of this can be found here:
https://www5.njit.edu/policies/sites/policies/files/NJIT-University-Policy-on-Academic-Integrity.pdf

Objective

This course will be an intensive deep dive into the relationship between the CPU and the GPU. We will be working in Vulkan, the latest industry standard graphics API from the Kronos Group (the people behind OpenGL and other industry standards). In addition to working with vulkan we will be building our 3D graphics based game engines and working with all of the topics related to that. Throughout the semester each student will work on a single large game project of their own design. There will be a major midterm milestone that will serve as a technical proof of concept of the game envisioned, a number of support applications, a research component, and ultimately a final playable game demo. Students will learn proper version control, programming practices, agile development, c/c++ programming, data modeling, file parsing, matrix math, and a variety of other related skills.

Use of Artificial Intelligence

All work submitted is expected to be the result of your efforts and your efforts alone. Use of AI is not permitted for submitted works.

Grading

- Milestones (~3):5%
- Project Design Document & Scrum Journal: 10%
- Midterm Exam: 10%
- Midterm Project Milestone: 20%

- Final Design Challenge: 15%
- Tool Chain: 10%
- Final Project: 30%

Course Materials

- Canvas Access
- A vulkan capable laptop - Re-image it before the semester begins.
- Discord
  - MIXR Lab: https://discord.gg/VGVQ3nrG5Y
  - Class: https://discord.gg/5nGQNz34AB
  - Note: Make sure your username is appropriate for class and change your nickname to your real name.
- A Github Account www.github.com
- https://github.com/engineerOfLies/gf3d - The seed project for this course
- GIT-Bash (for windows, or just use linux)
- Visual Studio or GCC
- https://vulkan-tutorial.com/
- SDL www.libsdl.org
- Updated Drivers with Vulkan Support (does not need to be top of the line, most cards should be supported, but will not have all features available)

Submission Criteria

All projects for the class must follow a set of submission guidelines to be eligible for grading. All projects must include the following:

- All Projects must be pre-approved with the professor or they will **not** be graded. A project proposal must be one to two page paper submitted through canvas. After submission a 1-on-1 meeting will be scheduled to discuss your project's specific deliverables. Grading of the project will be based on the amount of completed deliverables. Deliverables will be a combination of your project specific deliverables as well as common deliverables that will be posted to canvas
- All programming projects are to be submitted through www.github.org. To submit a project, post a link to the project repository along with the name of the branch you wish to have graded to your moodle project thread on canvas.
- You are required to present your projects in class as part of the project deliverables for grading. Make sure you perform a desktop share. These presentations are not formal presentations, merely demonstrations of requirements. Do not prepare slides.

Late Policy

Any projects that are submitted late will have a penalty of 1 point (of its percent value towards your final grade) per day late. Any project more than 7 days late will not be considered for grading. No exceptions.

Course Topics

Week	Topic	Due	Checkpoint
1	Course Overview Course Materials Introduction to Vulkan Programming Practices (Scrum / Commenting) 3D Rendering Concepts Review		Re-image your work machine, its past due
2	Compiling the Seed Project The Vulkan Overview 3D Game Engine Architecture		DemonstrateProject Compiles and runs
3	Swap Chains, Command Buffers, Pipelines, Descriptor Sets & Queues Resource Managers Entity Systems	Design Document Due/ 1 on 1 meetings begin	Able to make visual changes
4	3D Model Processing Shader Programming (GLSL) 3D Collision Detection		Functioning Entity System Collisions Detected
5	Additional Pipelines Event Driven Design Callbacks User Interface Programming	Scrum Journal Checkpoint	Rudimentary HUD Basic Gameplay Features
6	Lighting & Animation		Player Interacting with world
7	Midterm Exam Midterm Project Presentations	Midterm Project Scrum Journal Checkpoint	Entities interacting with Player
8	Midterm Project Presentations (Continued)		
9	Loading from Configuration (JSON, GLTF) Building your own Tools		
10	Common 3D Data Structures and Algorithms		

11	Research & Development Processes	Scrum Journal Checkpoint	
12	Lab Work		
13	Lab Work		
14	Final Project Presentations	Final Exam Scrum Journal Due Final Projects Due	

Schedule subject to change, attend class to keep up to date.