

CMPE 485 - Spring '22

Introduction to Game Programming

Term Project

Deadline: 12.04.2022 (Phase I) / 15.05.2022 (Phase II)

Project Mechanics: You can work alone or in groups of max 2.

DEFINITION: You have two options for the Term Project. Each option has its own requirements and expectations. So, read the details given below carefully and decide accordingly.

Option A (Single Challenge): If you choose this option, you will not design and implement a complete game but rather you will focus on a single technical challenge from the game programming domain. (It will be harder to create the challenge on the real game). You will do an in-depth analysis regarding your "technical challenge" and report your findings based on your experiment results.

Technical Challenge: Something (a nice feature of the game or game engine) that when not properly used causes excessive use of system resources (can be CPU, GPU, memory etc). Examples: Soft body dynamics in games, Controlling objects in multiplayer games, raytracing.

Here are tentative steps to follow for Option A:

- Decide on your "Technical Challenge".
- Design a hypothetical setting in which the scene is on-purposely designed to demonstrate the effect of your challenge. You need a controlled, repeatable experiment design. (Like the one given in ICP #1)
- Design repeatable experiments, clearly state your parameters.
- In-depth performance evaluation where you use measurements your code and the unity profiler to see the effect of your parameters on FPS, CPU usage, GPU usage (or whatever is relevant)

Grading:

Grading Criteria	Grading Weight	Further Explanations
Rationale	15%	Discussion on why the chosen technical challenge is significant in game programming. (You can give case analysis and discussion from game development forums, blog articles or academic papers)
Experiment Design	35%	The quality of the setting and scene in terms of its capacity to reflect the challenging behavior. Are all the relevant parameters chosen? Can they be varied properly in the scene? Can enough game objects be easily added? Has optimization techniques been included to help solving challenge?
Performance Evaluation	30%	Metrics chosen for the performance evaluation. Methods/tools used. How effective profiler is used. Has the challenge clearly/numerically demonstrated in terms of system resources.
Report	20%	Figures & tables clearly showing the effect of experiment parameters on your challenge. Technical explanation of the results.

Option B (Simple Game): If you choose this option, you will design and implement a complete simple game from scratch. You may inspire from an existing game mechanic, but you need to modify it in order to keep some level of authenticity. Of course, you are always welcome to create your own game mechanic.

You can use audio and visual elements from asset store or similar open sources; however, whatever you use should be free, no pay-to-win 😊 (Plus, we don't want to pay to test your projects)

Apart from visuals in game characters and scene objects, all software of your game should belong to you. You should clearly indicate any base code examples, tutorial videos etc. that your project depends on.

In this option, technical challenges will play an important role, too. However, here you will not do an in-depth analysis like in Option A. Instead, you will go over technical challenges that you deal with in your code, explain why are they a challenge? How did you solve it? Could there be alternative solutions?

You will also use profiler effectively to evaluate the performance of your game based on max three parameters that you chose.

Here are tentative steps to follow for Option B:

- Formally define your game mechanics (rules, aims etc).
- Decide on the software design approach: FSM, Game Manager, Delegates & events etc
- Decide on which libraries to use: For instance, if it is a networked game, use Netcode vs Mirror?
- Iteratively code the game. Define three technical challenges as your code progresses.
- Use unity profiler to evaluate the performance of your game. See the effect of your technically challenging part of your project on FPS, CPU usage, GPU usage (or whatever is relevant)

Grading:

Grading Criteria	Grading Weight	Further Explanations
Aesthetics (*)	5%	Audio visual design of player characters, scene objects etc.
Game Mechanics (*)	10%	Is it fun to play this game?
User experience	25%	Does the game provide a smooth user experience throughout the game?
Software design & implementation	40%	Are the used approaches appropriate and effective? Are the alternative implementation methods discussed? Are the technical challenges correctly identified?
Performance Evaluation	20%	Has the profiler been used correctly to assess the performance? Correct metrics chosen? Effect of challenging part demonstrated?

(*): *Aesthetics* and *Game Mechanics* will be evaluated collectively by the whole class.