GAMEPLAY SCRIPTING & IMPLEMENTATION I

GAME 240 (K01) – Spring 2024

Instructor: John P. Doran

Email: jdoran6@gmu.edu

Discord: netravelr#2081 (preferred communication method)

Class meetings: Tuesday & Thursday: 1:30 PM -2:45 PM

Office: #G551

Office Hours: Tuesday & Thursday: 10:15 AM -11:45 PM

Or by appointment*

* Tue., and Fri. are best for appointments. The best way to reach the instructor is via email.

Mason Game Design Mission Statement

The Mission of the Computer Game Design Program at George Mason University is to prepare students for employment and further study in the computer game design and development field, doing so with a curriculum designed to reflect the gaming industry's demand for an academically rigorous technical program coupled with an understanding of the artistic and creative elements of the evolving field of study.

Catalog Description

This class applies previously learned programming concepts, data structures, and techniques specifically relevant to game design within the context of professional game engines. Learners will use tools provided by the selected game engine to implement features commonly found in games.

Course Overview

In this course, you will use the tools provided by a professional game engine to implement features commonly used in games. Lessons will be provided on the common features provided by game engines followed by application of those concepts within the game engine selected for the course. The final project will focus on the implementation of several interrelated features.

Student Learning Objectives

Upon completion of this course, students will:

- Understand user-level features common to most modern game engine toolsets
- Apply game engine tools to implement features common in game development such as interactive player characters, collision response, non-player characters/objects, and player feedback
- Synthesize game engine features to create rudimentary games

Required Texts

There’s no required text for this course. Necessary materials will be distributed online if applicable.

Recommended Texts

Nystrom, R. (2014). *Game programming patterns*. Genever Benning.
(https://gameprogrammingpatterns.com/contents.html)

Required Software

Students will be required to have access to the following software in order to complete the required assignments:

- Unity Game Engine V2021.3.25f1 LTS
  - (I suggest using whatever is installed in the classroom so you know for sure that there will be no issues when moving between computers)
  - You can download any prior version of Unity via the Unity Download Archive at https://unity.com/releases/editor/archive.
- Visual Studio Community 2022

Course Structure

This course is structured around 13 week-long units accompanied by conceptual lessons, quizzes and other activities to assess conceptual understanding. The conceptual portion of a unit will be followed by the application of the learned concepts to a holistic game dev task or problem, typically requiring the student to implement a feature within an existing game. While completing the 13 lessons students will also spend time working on their final synthesis project which requires them to implement multiple interrelated features within the same game.

Concept	Application
Conceptual “abstract” lessons	**Procedural “how to” lessons**
Game engines typically provide a way to spawn multiple instances of the same game object. For example, to support this functionality the Unity Engine provides Prefabs while the Unreal Engine provides Blueprints.	*In Unity, you can create a Prefab from an object in the scene by dragging it from the Hierarchy Window into the Prefabs folder.*

Each project will include ***reach goals*** to ensure understanding of the project beyond what is provided in the application lessons.

Several reach goals will be provided so students may select those that align with their interests and ability level. Students may also create their own reach goals with prior approval from the instructor.

Reach Goals are **not** extra credit. They provide an opportunity to build on the core requirements needed to achieve a passing grade on the project in order to demonstrate problem solving skills and achieve an A or B on the project.

The Units / projects that make up this course are:

Unit 1: Editor Basics
Unit 2: Scripting
Unit 3: Player-Controlled Objects
Unit 4: Collision Detection and Response
Unit 5: Asset Management
Unit 6: 3D Model Animation
Unit 7: Camera Controls
Unit 8: Spawning Objects
Unit 9: Autonomous Object Behavior I
Unit 10: Autonomous Object Behavior II
Unit 11: Basic Level Layout and Lighting
Unit 12: HUDs and Data Management Across Scenes
Unit 13: Juice and Feedback
Final: Synthesis Project

Types of Coursework

There are three major types of work that you will do in this course: Projects, Lessons, and Class Participation.

Lessons (Concept)

Each Unit is made up of multiple lessons. A lesson will generally have some lecture material (like a video) as well as a short, graded activity (like a quiz) related to the lesson material. You are expected to complete all the graded lesson activities in this course.

Projects (Application and Reach Goals)

These are the projects that you will be completing during the week-long units of the course and the multi-week final synthesis project. You will be expected to complete a Unit's Project before proceeding to the next Unit.

Class Participation

Attend and actively participate in all required class sessions.

Multiple Attempts and Late Submission Policy

For Graded Lesson Activities

In general, I will accept multiple submissions for assignments and quizzes that are embedded in lessons. I will use the latest attempt for calculating your grade. So, if you were to take a quiz and miss a question, by all means, TAKE IT AGAIN to improve your grade!

While graded lesson activities don't have specific deadlines week by week, I recommend you do them as soon as you encounter them. It's possible to leave until the end of the course, but they tend to pile up quickly and you'd only be hurting yourself in the process.

For Projects

Unlike graded lesson activities, projects have set deadlines defined in the class schedule. Reach goals typically account for 30% of each project and are intended only for students who complete the core requirements of the projects early. Therefore, reach goals will not be graded for projects that are submitted past the deadline. No other penalty will be applied to late submissions. **No submissions for any project will be accepted after June 4th.**

Students may re-submit projects to address feedback and improve their scores.

Grading & Assessment Overview

Your grade will be based on weighted categories:

55%	Unit projects (13 projects)
15%	Synthesis Project
20%	Quizzes & Graded Lesson Activities
10%	Class Participation

Projects are graded on a combination of technical and creative competence.

Letter grades are assigned according to the following scale:

A: 90% - 100%

B: 80% - 90%

C: 70% - 80%

D: 60% - 70%

F: 0% - 60%

POLICIES AND IMPORTANT INFORMATION

Communication and Privacy

Students must use their MasonLIVE or gmu.edu email accounts to receive important University information, including messages related to this class. See http://masonlive.gmu.edu for more information.

Incomplete Grades

As per GMU policy for Incompletes: https://registrar.gmu.edu/topics/incomplete

> “A student who is **passing** a course may be unable to complete **scheduled** coursework due to a cause **beyond reasonable control**. In such a case, the instructor may assign a temporary grade of incomplete (IN).”

Please do not request an Incomplete if you do not have a passing grade, do not have a cause beyond reasonable control for not being able to complete upcoming work, or in hopes of completing coursework that is past due.

Students with Disabilities

If you are a student with a disability and you need academic accommodations, please see me and contact the Office for Disability Services (ODS) at 993-2474, http://ds.gmu.edu. All academic accommodations must be arranged through the ODS.

Honor Code

George Mason University has an Honor Code that requires all members of this community to maintain the highest standards of academic honesty and integrity. Cheating, plagiarism, lying, and stealing are all prohibited. All violations of the Honor Code will be reported to the Honor Committee.

See more information at https://oai.gmu.edu/mason-honor-code/full-honor-code-document

Use of AI

Any material generated by an artificial intelligence (AI) generation tool (such as ChatGPT or Stable Diffusion) is not accepted in this class as “the student’s own work,” and so will be considered similarly to text published on paper or online or text composed or significantly edited/altered by another person. The use of such text/images/etc. without proper attribution is a violation of academic integrity.

Assisting Other Students

In this course, we encourage you to help your fellow students, but you are not obligated to. If someone asks you for help troubleshooting a problem, you can help them or direct them to ask the instructor. Game development is a collaborative activity, and helping fellow students troubleshoot their problems can improve your own programming abilities. If you like helping other students but are worried that your help is crossing the line into cheating, you can reach out to your instructor for clarification.

University Catalog

Students are to be aware of the policies listed in the University Catalog: http://catalog.gmu.edu. University dates concerning withdrawal are not negotiable by the professor.

Additional Resources

GMU Student information and resources: http://www.gmu.edu/mlstudents/

The University Libraries maintain info guides for various majors. You can find links to various game design resources on the Computer Game Art & Design info guide: http://infoguides.gmu.edu/games

If you are a student with a disability and you need academic accommodations, please see me and contact the Disability Resource Center (DRC) at 703-993-2474. All academic accommodations must be arranged through that office. Students must inform the instructor at the beginning of the semester, and the specific accommodation will be arranged through the Disability Resource Center.

Disclaimer

In this class, I reserve the right to show a broad range of course materials, some of which assume the audience to be adults in age and demeanor. Should you at any time in the course of the class feel offended by something you have seen or heard, we would appreciate you staying to be part of a dialogue. If you feel that you cannot stay, remove yourself from the classroom as discreetly as possible. You may be asked to report on your response.