CSCI 4836: Game Development Fundamentals

Basic Information

- **ADA University, Spring Semester, 2026**
- **CSCI 4836: Game Development Fundamentals (6 Credits)**
- **Course meeting times and location**

Section CRN	Mondays	Fridays
20910	10:00 – 11:15	10:00 – 11:15
20619	11:30 – 12:45	11:30 – 12:45

- **Instructor:** Azar Aliyev (MS in Computer Science/Data Analytics), Room B010 CeDAR Main Office
- **How to contact instructor**
  - **In-person office hours:** Students are expected to actively use the course discussions forum for asking their questions.
  - **Email address:** araaliyev@ada.edu.az
  - **Preferred mode of communication:** Course discussion forum
  - **Optional:** Drop an email to agree on a meeting
- **Course Web page URL:**
  - **Blackboard:**
    - CRN 20910: https://ada.blackboard.com/ultra/courses/_18257_1/outline
    - CRN 20619: https://ada.blackboard.com/ultra/courses/_18018_1/outline
  - **GitHub Classroom:** https://github.com/ADA-SITE-CSCI4836-Spring-2026

Course Description

- **Prerequisites: CSCI 2324 - Computer Science**

- **Technology requirements**
  - Equipment (Hardware): Students must have access to a personal computer (Windows or macOS) capable of running a 3D Game Engine. Operating System should be Windows 10/11 (64-bit) or macOS 11.0+ (Big Sur or later). Minimum of 8GB RAM (16GB recommended), and a dedicated graphics card (GPU) is highly recommended for 3D rendering. **Important Note:** Chromebooks, iPads, and Tablets are NOT compatible with the Unity Editor and cannot be used for this course.
  - Software (Required): Unity Hub and **Unity 6** (or Unity 2022 LTS, Exact version number will be provided in the first lecture to ensure class-wide compatibility*),* **Visual Studio Code** (with C# Dev Kit extension) OR **Visual Studio Community 2022,**

Git (bundled with Unity or standalone) and a GUI client like GitHub Desktop, Software to record assignment demos (e.g., OBS Studio, QuickTime, or Windows Game Bar).

- **Overview of course:** This course explores the engineering foundations of modern game development using **Unity** and **C#**. Moving beyond basic design, students will master real-time simulation, vector mathematics, and component-based software architecture. Through a **"Modular Sandbox"** approach, the course covers physics, AI, and decoupled systems, culminating in a team-based **Capstone Project** focused on agile development and version control.

- **Student learning objectives:** Students will gain a comprehensive understanding of game engine architecture, real-time simulation logic, and the mathematical principles governing physics and motion. Through the practical application of C# and the Unity engine, the course builds proficiency in scripting artificial intelligence, implementing decoupled software patterns, and managing complex asset pipelines. Furthermore, students will practice industry-standard workflows, including version control and agile team collaboration, to successfully architect and deploy interactive software.

- **Methods of instruction:** The class will be taught through lectures, including discussions around class examples and course homework. Discussions based on student contributions add a vital and dynamic element to the class. Students are expected to come to the class with comments or questions from the course readings.

- **Workload**: It is estimated that the students will need to spend 3-4 hours of study and preparation for the classes every week. Estimated amount of time to spend on course homework is additional 3-4 hours per week

Policies

- **Grading Procedures**
  - The students will be graded on absolute scale.
  - The course grade will be calculated from the following components:

Attendance – 5%	Asg1 – 5%	Asg 2 – 10%	Asg 3 – 15%	Asg 4 – 10%	Midterm – 20%	Final Project – 35%

- **Attendance and tardiness:** Attendance is an indispensible element of the educational process. In compliance with Azerbaijani legislation, instructors are required to monitor attendance and inform the Registrar and the Dean of the respective School when students miss significant amounts of class time. Azerbaijani legislation mandates that students who fail to attend at least 75% of classes will fail the course.
  - Starting from the 2nd week of the semester (right after the Add/Drop period on 30 Jan 2026) in the first half of the class. Thus, the students have extra time to make it to the classroom to be marked as ‘present’. Otherwise, they will be marked as ‘absent’. They also are required to be present 70% of the class.
- **Missed or late assignments/extensions:** No missed or late assignments will be accepted later.

- **Standards for academic honesty and penalties for infractions:** If a student found guilty of academic dishonesty for the first time after the "Forgiveness Period", he or she would fail the course. If the case of repeated violations, the student will be suspended or permanently expelled. For more information, please read the Honor Code

Requirements

- **Exams and quizzes:** Students will take 1 exam (midterm). It will be closed book test(s) consisting of multiple-choice, fill-in-the-blank and open-ended questions. The Midterm exam's time and place will be communicated during the term and it is currently planned to be within the 30 March 2026 week.
  - Weekly self-check quizzes (participation activity) are for students to take after the class sessions to check how well they learned the material. Each one is an individual non-graded quiz with limited 3 attempts. Hence, the suggestion is to not mechanically retake the quiz until you get all answers right, rather to focus on your mistakes and make sure you go through the corresponding material or ask relevant questions in the Discussions forum. Students unlock the next week material only if they score at least 50% in the self-check quiz for this week.
- **Assignment/problem sets/projects/reports/research papers:** Homework(s) in a form of individual assignments and a team project will be given during the semester. Each homework assignment will be based on the material covered in the class. The team project will be culminated with the presentation of the prototype at the end of the term. Teams of 4±1 students will be formed. Detailed information and the exact date will be communicated during the term. The students will submit the homework assignments online and in hard copy (as required). The homework(s) will be graded based either on clarity, technical soundness, thoroughness and coverage, relevance and utilization of resources, or on correct functioning, quality of documentation, and cleanness of the code. Presentation(s) will be graded based on quality of content and delivery, level of participation, organization and design knowledge.

Schedule

- **Last day to withdraw from the course:** 04 April 2026

	Week of	Theme	Topics	Learning Outcomes	Reading
1	19 Jan	Part 1: The basics, History & Analysis **(Assignment 1)**	History of technical constraints (Pong to Doom) | Game Deconstruction Workshop.	• Define the relationship between hardware history and mechanics. • Analyze the rules/logic of a classic arcade game. • Deconstruct a game into entities and rules.	**MDA Framework**
2	26 Jan	Unity & Deployment	Editor Basics & The Build Pipeline | "Hello World" – Building a	• Navigate the Unity Interface (Scene, Hierarchy, Inspector). • Create a standard WebGL Build.	**Unity Manual: WebGL**

		(Assignment 1 Due)	cube scene and uploading to itch.io.	• Deploy a Unity project to a public web URL.	
3	2 Feb	The Loop & Input **(Assignment 2)**	The Game Loop (Update vs FixedUpdate) | Moving objects with Vectors and Input.	• Explain frame independence (Time.deltaTime). • Implement basic movement using the Input System. • Apply vector addition to transform objects.	**Game Prog. Patterns: Loop**
4	9 Feb	Physics 101	Rigidbodies, Colliders, and Triggers. | Building the "Arcade Physics" prototype.	• Distinguish between Kinematic and Dynamic Rigidbodies. • Use Colliders to detect interactions. • Implement OnCollisionEnter logic.	**Unity Manual: Physics 2D**
5	16 Feb	PART 2: Gameplay / Object Management **(Assignment 2 Due)**	Instantiation and Destruction. | Spawning enemies/projectiles.	• Instantiate Prefabs dynamically via script. • Manage object lifecycle to prevent memory leaks. • Implement basic random generation logic.	**Unity Manual: Prefabs**
6	23 Feb	Camera & Audio	Cinemachine (Camera tracking). | Adding background music and SFX.	• Implement Cinemachine for smooth camera tracking. • Apply AudioSources and Spatial Sound. • Configure audio rendering settings.	**Unity Manual: Audio**
7	2 Mar	Simple AI (Logic) **(Assignment 3)**	Logic Blocks (if/else) vs State Machines. | Using NavMesh for movement.	• Construct simple AI logic using conditional statements. • Implement Unity NavMesh for pathfinding. • Create an enemy that chases the player.	**Game Prog. Patterns: State**
8	9 Mar	Assessment & Vision	Raycasting Lite ("Lasers" and detection).	• Explain core Unity lifecycle and C# OOP concepts. • Use Raycasting to detect objects. • Calculate simple distances for AI vision.	**Unity Manual: Raycast**

9	16 Mar	PART 3: Systems **(Assignment 3 Due)**	The Singleton Pattern. | Using UnityEvents for decoupling.	• Implement a Singleton Game Manager. • Use UnityEvents to connect systems visually. • Decouple UI logic from Player logic.	**Game Prog. Patterns: Singleton**
10	30 Mar	UI & Menus **(Assignment 4)** **(Midterm)**	Canvas System and Anchors. | Building a Main Menu and Pause Screen.	• Construct responsive UI using RectTransform. • Script interactions for Buttons and Sliders. • Manage Game States (Pause/Resume).	**Unity Manual: UI Toolkit**
11	6 Apr	Data Persistence	PlayerPrefs (Saving Data). Saving/Loading High Scores.	• Use PlayerPrefs to store simple data. • Implement a High Score system. • Reload scenes using SceneManager.	**Unity Scripting API**
12	13 Apr	Polish ("Juice") **(Assignment 4 Due)**	Particle Systems & Asset Store. | Importing art and adding "feel."	• Apply Particle Systems for visual feedback. • Integrate third-party assets. • Implement basic visual tweens.	**Unity Manual: Particles**
13	20 Apr	PART 4: Advanced Topics & Production	Multiplayer Networking (Theory: Latency, Client-Server). | Procedural Generation (Theory: Noise, Algorithms).	• Explain the difference between Client-Server and P2P. • Describe the concept of Latency and Prediction. • Explain how algorithms (Perlin Noise) generate terrain.	
14	27 Apr	The Industry: Business & Ethics	Monetization & Success Stories.	• Distinguish between Premium, Freemium, and Ad models. • Analyze case studies of successful indie games. • Explain IP rights and Copyright basics.	
15	4 May	Showcase	Project Presentations	• Demonstrate a complete, playable software product.	

				• Articulate technical challenges and solutions.	

Disclaimer

- **The course schedule is subject to change as necessary**