Unity Coding Challenge 🎮

Objective

Welcome to the **Unity Coding Challenge!** This activity is designed to test your **problem-solving skills, debugging abilities, and feature implementation in Unity.**

You will complete a series of coding challenges, earning points based on their difficulty. 🏆

Rules & Guidelines:

✅ Each student must complete at least **5 challenges**.

✅ Challenges are categorized based on skill type.

✅ Points range from **10 to 100**, depending on difficulty.

✅ Your goal is to **maximize your score within 4 hours** (including a break).

✅ **Solutions must be tested and demonstrated to an instructor** for validation.

Challenge Categories

1. Bug Solving (Debugging & Problem-Solving)

- Identify and fix **bugs** in existing Unity scripts.
- Use **debugging tools** and **log outputs** to analyze and resolve issues.

2. Feature Development (Game Engine & Programming Skills)

- Complete **partial scripts** by implementing missing functionality.
- Apply **Unity components**, **C# scripting**, and **best practices** for game development.

3. Resume Builders (Interview-Oriented Tasks)

- Implement **real-world programming problems** commonly used in Unity interviews.
- Focus on **data management, system design, and clean code practices.**

4. Custom Made (Game Mechanics & System Design)

- Create an **original** game mechanic or system from scratch.
- Showcase your ability to **design and implement features independently**.

5. Showcase (Demonstrating Core Principles)

- Complete a task that highlights a **specific programming principle**.

Task	Points
Implement a **pause screen** that freezes and resumes the game on input.	**10 pts**
Implement an **energy bar** using the Unity UI system, with automatic refill over time.	**15 pts**
Showcase an example of **Polymorphism** in action.	**15 pts**
Showcase an example of **Inheritance** in action.	**15 pts**
Demonstrate the **SOLID Interface Segregation Principle** in a Unity-based scenario.	**15 pts**
Create a **simple camera shake effect** using C# scripting.	**15 pts**
Implement a **2D parallax scrolling background**.	**15 pts**
Implement the **Strategy Pattern** to allow different movement behaviors.	**40 pts**
Prototype a **basic procedural level generation system** using prefabs.	**40 pts**
Implement a **physics-based rope system** using Unity's LineRenderer and joints.	**40 pts**

6. Research & Development (R&D)

- Investigate **advanced Unity topics** and implement a working demo.

Task	Points
Implement **serialization and deserialization** of game data using JSON.	**30 pts**
Research and showcase **Unity's garbage collection system**, optimizing an inefficient script vs. an efficient one.	**50 pts**
Select a **Unity component** (e.g., Cinemachine, NavMesh, LineRenderer, AudioMixer, ProBuilder) and create a **small**	**50 pts**

interactive demo showcasing its use.	

Scoring & Submission

🏆 Each completed challenge earns **points**.

🛠️ **All solutions must be tested and validated by an instructor** before submission.

🥇 The **highest-scoring student** will receive **honorary recognition**! 🚀

Good luck, and happy coding! 🎉