COMSATS UNIVERSITY ISLAMABAD, ABBOTTABAD CAMPUS

Department of Computer Science
Terminal Examination – Spring 2025

Class:	BCS 7C & 7D	Date:	July 12, 2025
Subject:	Game Development (CSC495)	Instructor:	Mazhar Bukhari
Time Allowed:	3 Hours	Max Marks:	60

1. Answer the following questions to demonstrate your understanding of Unity concepts, systems, and game logic. [CLO-1] [15 Marks]
   - 1.1. What are Post-Processing Effects in Unity? Name and briefly explain any three common post-processing effects.
   - 1.2. Explain the purpose of Radius, and Slope settings in a NavMesh Agent while configuring a navmesh surface. Analyze how these values affect enemy navigation on a terrain-based level.
   - 1.3. Define a Behavior Graph and the role of the Blackboard in controlling AI logic.

2. Use your Unity skills to construct scenes, scripts, and components in a game development context. [CLO-2] [20 Marks]
   - 2.1. You are designing a Hover Dash level with platforms, oscillating spike traps, fuel crates, and a key to collect. Draw and label an Onion Architecture showing core features, mid-tier features, and extra polish. Explain how each feature feeds into the core gameplay.
   - 2.2. Construct a C# script that checks if the player has collected all three keys before opening a final door. Use basic conditional logic and tagging.
   - 2.3. Relate the concept of ProBuilder shapes to Unity's built-in shapes (e.g., Cube, Sphere). Design an example of modifying a cube into a custom object using ProBuilder tools.
   - 2.4. Outline the complete steps required to import and rig a character from Mixamo into Unity. Include necessary rig settings and import configurations.
   - 2.5. Configure a GameObject (e.g., platform) to become visible or active only after a specific key is collected. Describe how to do this using Unity's inspector and scripting.

3. Answer the following to demonstrate your ability to create animation logic and visual transitions. [CLO-3] [Marks 10]
   - 3.1. Design and draw a state diagram for an Enemy AI Animator that includes four states: Idle, Walk, Run, Punch Also, list the parameters and specify the transition conditions.
   - 3.2. Demonstrate how to create a Blend Tree for a third-person character that blends between Idle, Walk, and Run based on player speed. Why is this approach preferred over using separate transitions?

4. Apply AI, visual scripting, and advanced Unity systems in practical development scenarios. [CLO-4] [Marks 15]
   - 4.1. Scenario: A behavior graph already contains a Switch node with four states: Idle, Chase, Patrol, Attack. Assume all except Patrol are working.
     Design a Patrol Node with the following:
     - List steps to configure PatrolPoints in Unity.
     - Add a Condition node that checks if the enemy is still in patrol mode — if not, it should switch to the appropriate state.
   - 4.2. Implement camera switching in Unity's Starter Assets – Third Person Controller to activate a Cinemachine Aim Camera when the Right Mouse Button is pressed.
     - Demonstrate the required C# code logic to toggle between the default and aim camera based on input.
     - Configure the necessary Cinemachine setup, camera priorities, and input bindings within the Unity Editor.
   - 4.3. Apply the concepts of Unity's NavMeshAgent to develop a script that enables an NPC to follow the player's position using pathfinding. Include how you would use SetDestination() and reference the player position dynamically.

CamScanner