Lab 6

Q1

NetworkVariable<int>

Used for synchronizing an integer value like a players health in a multiplayer game

Example: when a player takes damage all players can see their new health value

NetworkVariable<Vector3>

Used for synchronizing a position value like a players position in a game

Example: used so that all players can know where another player is at all times

Q2

[ServerRpc]: Marks a function as a server-side Remote Procedure Call (RPC). When a client calls this function, the server executes it

Purpose: A player shoots a projectile, and the server handles the spawn and validation

[ClientRpc]: Marks a function as a client-side RPC. The server calls this function, and it executes on all connected clients

Purpose: The server notifies all clients that a power-up has been collected and should disappear

[Server]: Ensures that the marked function can only be executed on the server

Purpose: Prevents unauthorized changes to game-critical logic, such as modifying player scores

[Client]: Ensures that the marked function can only be executed on a client

Purpose: Local player input handling that should not run on the server

[NetworkVariable]: Marks a variable as synchronized across the network

Purpose: A shared game timer that must be consistent across all clients

Q3

ServerRpc: Called by a client and executed on the server. Useful for validating player actions before updating game state

ClientRpc: Called by the server and executed on all clients. Used for broadcasting events like visual or sound effects

Example Use

In a multiplayer shooter game

1 player fires a bullet by pressing the fire button

2 client calls a ServerRpc to inform the server

3 the server verifies the shot and applies damage then calls a ClientRpc to update all clients

Q4

The NetworkManager in Unity Netcode is responsible for managing the overall networking lifecycle, including:

>Starting and stopping network sessions (acting as host, server, or client)

>Handling player connections (tracking connected clients)

>Synchronizing game state between the server and clients

>Providing utilities like spawn management and scene transitions

It is the backbone of multiplayer games, ensuring proper communication between all networked objects.