**Lab 10**

(a) Interest Management reduces bandwidth and processing by ensuring players receive only relevant game state updates.

1. Spatial Partitioning:

The game world is divided into regions possibly using a grid.

Players and objects are only updated if theyre within a certain distance or region around a player.

Example: A player only receives updates about nearby enemies, NPCs or loot within a 100 meter radius.

2. Visibility-Based Filtering

Takes into account what a player can actually see, factoring in obstacles, line of sight, and camera orientation.

Updates about objects behind walls, inside buildings, or blocked in any way are not sent.

Benefits to the Game:

Reduces network traffic by sending fewer updates.

Improves performance on both client and server side.

Scales better for large player counts.

(b) Potentially Visible Set (PVS) vs Static Zones

Potentially Visible Set:

A set of areas that are potentially visible from a given region or camera location.

Used in 3D environments where visibility is not determined by distance alone, but by occlusion (walls and around corners for example).

Static Zones:

The world is divided into fixed zones like grid cells.

Players in one zone receive updates only from nearby or connected zones.

Differences:

Pvs - Whats visable is updated

SZ - Updates based on its spacial location

Pvs - Dynamic and updated at runtime.

SZ - Static or preconfigured areas are updated depending where you are.

Pvs - High percision difficult to implement.

SZ - Doest increase performance as much but easy to implement.

Both improve scalability, reduce lag, and enable smoother experiences in massively multiplayer games.

(c) Server side attack example: SQL Injection

How the Attack Works:

An attacker sends malicious SQL code in input fields or network packets to manipulate the game servers database.

Example: A player sends the input ' OR '1'='1 in a login field.

The server may execute this as:

sql

Copy

Edit

SELECT \* FROM users WHERE username='' OR '1'='1';

This returns all users, allowing the attacker to log in without a valid account.

Impact:

Unauthorized access to accounts.

Manipulation or deletion of game data.

Data leaks.

Prevention Mechanism:

Use Prepared Statements/Parameterized Queries: These ensure input is treated as data, not executable SQL.

Input Validation and Escaping: Sanitize all inputs from users.

Least Privilege Principle: Limit database access privileges for game server accounts.

(d) Example of Cheating: Speed Hacking

How It Works:

A cheater modifies the game clients perception of time or alters movement variables to move faster than allowed.

This can be done by:

Changing memory values via tools like Cheat Engine.

Exploiting weaknesses in client side validation.

Impact:

Unfair gameplay such as dodging bullets or faster travel.

Frustration for honest players.

Server integrity compromised if movement is not validated.

Prevention Mechanism:

Authoritative Server Model: The server determines valid movement and position, not the client.

Movement Verification: Server checks whether position updates violate maximum speed thresholds.

Time Synchronization: Server maintains control over the game clock, preventing time exploits.