**Shooting Around Corners**

Imagine a shooting game where there are two player, A and B, they are facing each other around a corner. Player A moves out from behind it, shoots at player B, and moves back. Due to latenct the time it takes for data to travel from Player A’s computer, to the server, and finally to player B’s computer, Player B might see player A pop out slightly later, if player B reacts by shooting back, their action might be too late on their own screen and player A might have already moves back to safety.

A diagram of a computer system

Description automatically generated

This demonstrates a consistenct issue where the game state known to Player A is vastly different from player B due to network delays.

**Consistenct Management Techniques**

The dumb client model is a conservative category consistenct management technique. In a dumb client setup, the client does minimal processing and entirely depends on the server for the game state. The server calculates everything and sends updates to the client, this method axes most cheating opportunities and inconsistencies but can feel less responsive especially for high latency clients.

Client side predition is a technique catagorised under optimistic. It involves the client predicting the outcomes of actions without waiting for the server confirmation, this makes the game feel more responsive to the player. The client will adjust the game state based on the server’s updates when they arrive. This may lead to corrections like positions snaps and such.

**Cheating**

Speed hacking is a form of cheating that involves changing the game client to increase the speed of a player’s movement, allowing them to move faster than normal. This gives an unfair advantage in most games. To prevent it the server can implement checks that validate player movement against expected macimum speeds. Automated systems can penalise the player if cheating is suspected.

Wallhacking allows a player to see through solid objects which provide unfair awareness advantages. They can use this to see where other players are without the visual or audio cues. To prevent this, the server can use game logic to manage which player locations are sent to each client by ensuring the data about a player not visible to the client is not sent to them in the first place.