**GAM 330 Final Project Overview Notes**

Dave Notes (final project overview)

Example: You could develop a game where lots of players connect, ready up, server progresses to now we are playing the game, during the actual gameplay...

* You play one game (someone wins)...
* Now round 2, shuffles players to a different role
* Kick off new game / new round without disconnecting anyone from the server
* 4 objects with network syncs (player movement) and communication via RPCs (abilities, fire a fireball)

Any sort of thing where a player does a specific action or changes the state of something would use an RPC. Server updates the state of the game with an RPC

(optional): Go back to role selection screen without popping all the way out of server, so players don't have to reconnect

**Dave on Monkey Ball Network Physics difficult**:

Make it so one client (the one controlling the ball) is simulating physics.

Two people building tracks (God views) would use RPCs to put tracks on the level, server sends it down to the one monkey ball player.

Two god views wouldn't simulate physics, they would have an object for the monkey ball w/ a network sync on it, but its collision and rigidbody would be disabled. The monkey balls movement would be based on the networks syncs coming from that client telling the server the monkey’s position.

The server would then send the GOD view clients the position updates to the dummy monkey object in the GOD clients scene’s with a local network sync on it. That same dummy object would not have a RB on it though.

**Question on having god objects with a local object with Network Sync on it:**

**RYAN:** I guess, why would it need an object in their local scene at all if the server would be sending them the RPC information in regards to that monkey ball client?

**DAVE:**  Let me put it this way... when both of us connect to a game and play together, we each might have a player object that runs around. On your computer, you have a version of my player, but the controls are disabled (because I'm telling you where the object is via the NetworkSync component).

Essentially, you have a version of my player, but you aren't simulating where it should be... because you are relying on me to tell you where I am. Then, when you're talking about physics, it's the same deal. The physics system is a way to move an object around based on physical properties. Your client isn't in charge of where my player is, so it shouldn't be running physics on it. It should just be putting it where I tell it to go via the NetworkSync's sync messages (not through RPCs)

**RYAN: S**o, essentially, the god mode client don't need any info in their local scenes about the monkey ball object ahead of time, because the RPC info is coming from the Server

**DAVE:** Well, they'll need a game object that has the visuals, etc... but information about it's position and rotating should all come from the client that is actually controlling the ball.

No problem. Most of the time when you have networked game objects, you have "local" and "remote" versions. My local player has scripts on it to move based on my input, your remote version of my player object has scripts on it to move based on my client sending you position updates about it.

**Multiple Players simulating physics common strategy:**

Similar to the concept of ownership (I own this object, it has a network sync, I’m telling you where it’s position is). In the same way, the common way to deal with physics is to have ownership over multiple physical objects.

“I’m going to simulate the physics on the objects that are right next to me. You’re the one that is going to simulate the physics on those objects, because you’re interacting with them (objects close to you).”

Then, there is also going to be some logic for transferring between them.

* If I take a ball and throw it to you, I started simulating the physics (owned that object), and then once it reaches a certain range to the player it is being thrown to, then that player receiving the ball toss will start simulating physics on that object (gain ownership)

**Deterministic Lock Step** (<http://gafferongames.com/networked-physics/deterministic-lockstep/>)

* Cubes are in two different locations when they fall
* This is why you only want one player/client simulating physics on a particular object

Def: A way to simulate physics, and send the data to the other clients.