

Networking

Many places at once

HONESTLY..

- Networking is not my area.
- I've never personally implemented a networked game.
- Still, I can describe the problem, and some approaches.

WHAT'S THE NET?

- Two kinds of transports.
- TCP/IP
 - Gets your packets there.
 - ..eventually.
- UDP
 - Possibly faster.
 - But doesn't guarantee anything.

DERIVATIVES

- UDP can be used to build your own "TCP/IP"
 - Out of order packets, for instance.
- Sometimes you don't need to care about lost packets!
 - If you spam the network with position data, for instance, losing some of these might not be a problem.
- Different kinds of network traffic for different kinds of games.
 - Chess vs FPS vs RTS

- What's the time?
 - PC timers run at slightly different speeds.
 - Audio clocks run at slightly different speeds (i.e. not exactly 44.1khz for instance)
 - > Can't trust computers' clocks to run at the same speed!
- Need constantly resynchronized "network time" of some sort.
 - Which is kind of like mailing someone a letter saying "let's synchronize our watches!"

KEEPING IN SYNC

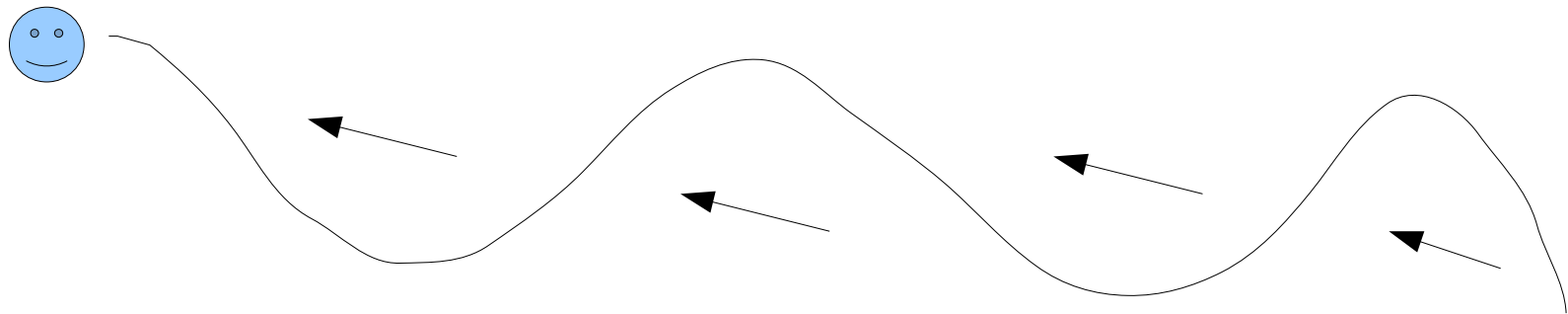
- Approach 1: peer-to-peer
 - Run the exact same simulation on all nodes.
 - Can't trust clock to run at the same speed.
 - Can't trust floating point ops to calculate exactly same values!
 - Keep machines in sync frame by frame and send keypresses around.
 - Locked to the lowest FPS (or worse)
 - Requires a lot of bandwidth and zero ping
 - Only feasible on LAN

KEEPING IN SYNC

- Approach 2: client-server
 - Server knows the 'truth'.
 - Clients run simulation, predicting future state.
 - Server updates state to all clients periodically.
 - If simulation and "the truth" vary too much, warp objects.
 - For performance reasons, may want to trust client on some things (such as player's position).
 - No player sees the "actual" game situation, but with low enough ping, "close enough"

PREDICTION FUN

- In a complex case you may end up having the following on a client:
 - Current simulation state.
 - Latest known truth.
 - Latest prediction target.



INTERNET FUN

- Apart from variable lag and lost (and out of order) packets:
 - IPv4: today
 - Majority of users behind NAT.
 - May need a lobby server for NAT punching.
 - IPv6: tomorrow
 - Can't expect users to type in IPv6 addresses.
 - 1234:5678:9012:3456:7890:1234:5678:9012
 - Lobby server for matchmaking again.

WHO TO TRUST?

- Cheating rampant in networked games.
 - Hey, developing cheats may actually be fun.
- Some ways to fight this:
 - Strict client-server split. Server owns the "truth".
 - Possible performance issues.
 - Server tells clients things on a need-to-know basis.
 - Possible performance issues; client may not be able to do predictive simulation properly.
 - Encryption, client binary validation, anti-debugging techniques, etc.