Game Networking
Game Networking Topology

• Star (client/server) Topology
  – All devices are connected to a central hub (server)
  – Nodes communicate across the network by passing data through the hub
Game Networking Topology

• Mesh (peer to peer) Topology
  – every node has a connection to every other node in the network.
  – provides faster communication
  – burden on clients to enforce application rules
  – minimize size of messages
What is communicated across networks?
- player locations
- player states
- player orientation
- voice communications

Note: send the minimum amount of data necessary
- minimize data type sizes (bytes are nice)
Networked games must be:
- fast
- reliable

Which should be used (client/server or peer-to-peer) for MMOs?
- it depends, but mostly client/server

Architectures in use:
- pure client/server
- pure peer to peer
- hybrid client/server & peer to peer
Game Networking

- Latency
  - a difficult issue in multiplayer games
  - How do you handle communication delays?
  - Where is the delay?
    - lag is commonly due to overloaded networks
What UDP (User Datagram Protocol) gives you:

- unreliable delivery
- no retransmission, packets not ACKnowledged, no reordering
- no congestion control
- speed & low overhead (good for streaming audio/video)
TCP vs. UDP

- Both transport layer protocols on top of IP.
- TCP is lossless.
- What TCP (Transmission Control Protocol) gives you:
  - reliable delivery
  - retransmission and reordering
  - congestion control
TCP VS. UDP

- Some game data can be *interpolated*.
  - if packet is lost, the game can continue.
- Some game data must be guaranteed to be received.
  - some things can’t and shouldn’t be interpolated.
- Understand TCP is slower, but safer.
- Both are used, both have virtues, both have drawbacks.
- Programmers are always looking for UDP opportunities.
- Alternative: use a “Reliable UDP” and use for certain types of info sent.
Ogre Network Plugins

1. RakNet
2. OpenTNL
Open TNL

- OGRE page
- Sourceforge
Using RakNet

- Site
- Overview
- Bitstream
  - static BitStream bs;
  - bs.Write(PACKET_ID_LINE);
  - bs.Write(mCamera->getPosition());

- rakconnection->Send(&bs, HIGH_PRIORITY, RELIABLE_ORDERED, 0, UNASSIGNED_SYSTEM_ADDRESS, true);

- Example code
Class Projects

Topics
- Animation
- Gui's
- Network
- Physics
- Artificial Intelligence
- Sound
- Other
Class Projects

- Add a feature to our basic OgreBullet neworked code (Assignment 7)
  - Can work in groups if project is sufficiently ambitious, but each group members tasks must be clearly delineated from the start.
Class Projects

- The project will be submitted in the form of a tutorial that explains new features and adds the parts incrementally.
- Project plan due 3/6.