Multiplayer Computer Games

Jouni Smed
Department of Information Technology
University of Turku
http://www.iki.fi/smed

Course Syllabus

- credits: 5 cp (3 cu)
- recommendable prerequisites:
  - Algorithms for Computer Games
  - knowledge on the basic concepts of computer networks
- assessment
  - examination only (no exercises)
- course web page: **http://www.iki.fi/smed/mcg/**

Lectures

- Lecture times:
  - Mondays 14–16
  - Wednesdays 16–18
  - Thursdays 14–16
- October 29 – November 29
  - N.B. no lecture on Nov. 8, Nov. 19, Nov. 21, Nov. 22, and Nov. 28
- Auditorium Alpha, ICT Building


Examinations 1 (2)

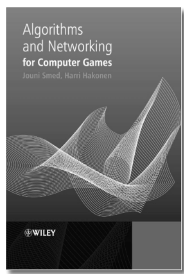
- examination dates
  1. ?? (possibly December, 2007)
  2. ?? (possibly January, 2008)
  3. ?? (possibly February, 2008)
- check the exact times and places at **http://www.it.utu.fi/opiskelu/tentit/**
- remember to enrol! **https://ssl.utu.fi/nettiopsu/**
- if you are a student of Åbo Akademi University, you must register to University of Turku to receive the credits
  - further instructions are available at **http://www.tucs.fi**

Examinations 2 (2)

- questions
  - based on both lectures and the textbook
  - two questions, à 5 points
  - to pass the examination, at least 5 points (50%) are required
  - grade: $g = \lceil p - 5 \rceil$
  - questions are in English, but you can answer in English or in Finnish
- remember to enrol in time!

Textbook

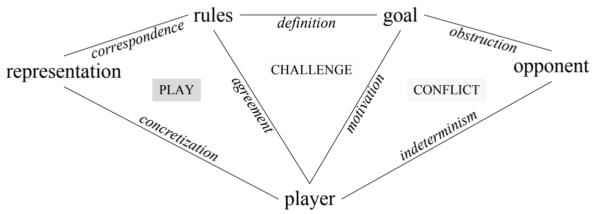
- Jouni Smed & Harri Hakonen: *Algorithms and Networking for Computer Games*, John Wiley & Sons, 2006.
- **http://www.wiley.com/go/smed**



Outline of the Course

8. Communication layers
   - physical platform
   - logical platform
   - networked application
9. Compensating resourse limitations
   - aspects of compensation
   - protocol optimization
   - dead reckoning
   - local perception filters
   - synchronized simulation
   - area-of-interest filtering
10. Cheating prevention
   - attacking the hosts
   - tampering with network traffic
   - look-ahead cheating
   - collusion
   - offending other players

Components, Relationships and Aspects of a Game

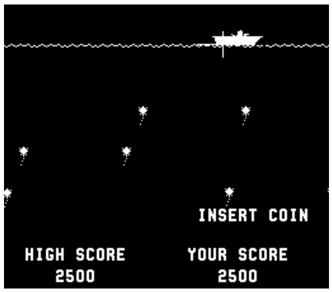


So What Is Multiplaying?

- multiplaying vs. single-playing
  - opponents are not controlled by a computer but other humans
- interaction amongst the multiple players
  - attempt-based
    - sports games
  - turn-based
    - board games, play-by-email games
  - real-time
    - real-time strategy games, first-person shooters

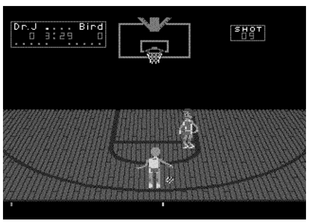
Answer 1: High-score List

- attempt-based interaction
- examples
  - pinball machines
  - *Sea Wolf* (1976)
  - *Asteroids* (1979)



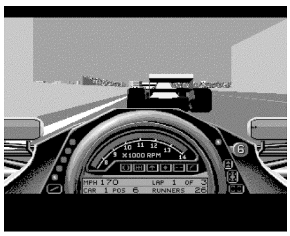
Answer 2: Multiple Game Controllers

- multiple players using the same computer
  - multiple controllers
  - split screen
- examples
  - *Pong* (1972)
  - *One on One* (1983)



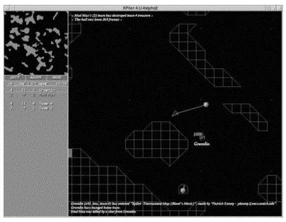
Answer 3: Time-slicing

- one active player at a time
  - one computer, one controller
  - multiple players taking turns
  - computer controls the passive players
- example
  - *Formula One Grand Prix* (1991)



Answer 4: Server and (Dumb) Clients

- multiple computers
  - the game runs on a server
  - the clients display the output and convey the input
- examples
  - *Multi-User Dungeon* (1978)
  - *Xpilot* (1991)



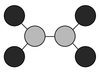
Answer 5: Players as Peers

- multiple computers
  - the same game runs on each participating computer
  - players' decisions are conveyed via a network
- example
  - *Doom* (1993)

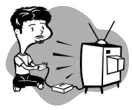
§8 Communication Layers

- physical platform

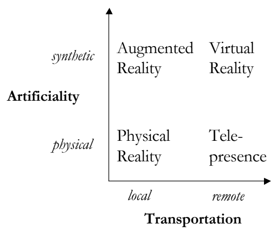
- logical platform



- networked application



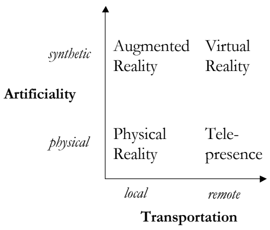
Classification of Shared-Space Technologies 1 (2)



Benford et al., 1998

- Physical reality
  - resides in the local, physical world
  - here and now
- Telepresence
  - a real world location remote from the participant's physical location
  - a remote-controlled robot

Classification of Shared-Space Technologies 2 (2)



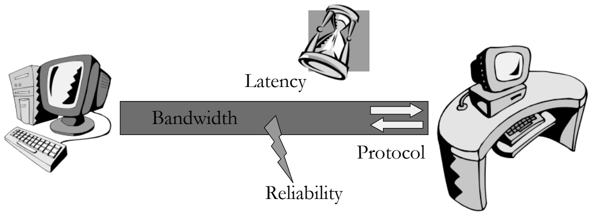
Benford et al., 1998

- Augmented reality
  - synthetic objects are overlaid on the local environment
  - a head-up display (HUD)
- Virtual reality
  - the participants are immersed in a remote, synthetic world
  - multiplayer computer game

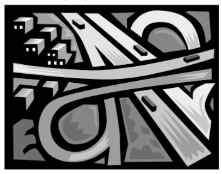
§8.1 Physical Platform

- resource limitations
  - bandwidth
  - latency
  - processing power for handling the network traffic
- transmission techniques and protocols
  - unicasting, multicasting, broadcasting
  - Internet Protocol, TCP/IP, UDP/IP

Network Communication

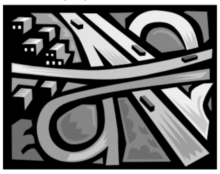


Fundamentals of Data Transfer 1 (3)



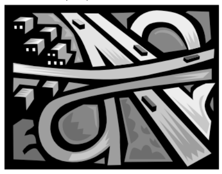
- Network latency
  - network delay
  - the amount of time required to transfer a bit of data from one point to another
  - one of the biggest challenges:
    - impacts directly the realism of the game experience
    - we cannot much to reduce it
  - origins
    - speed-of-light delay
    - endpoint computers, network hardware, operating systems
    - the network itself, routers

Fundamentals of Data Transfer 2 (3)



- Network bandwidth
  - the rate at which the network can deliver data to the destination host (bits per second, bps)
- Network reliability
  - a measure of how much data is lost by the network during the journey from source to destination host
  - types of data loss:
    - dropping: the data does not arrive
    - corruption: the content has been changed

Fundamentals of Data Transfer 3 (3)



- Network protocol
  - a set of rules that two applications use to communicate with each other
  - packet formats
    - understanding what the other endpoint is saying
  - packet semantics
    - what the recipient can assume when it receives a packet
  - error behaviour
    - what to do if (when) something goes wrong

Internet Protocol (IP)

- Low-level protocols used by hosts and routers
- Guides the packets from source to destination host
- Hides the transmission path
  - phone lines, LANs, WANs, wireless radios, satellite links, carrier pigeons,…
- Applications rarely use the IP directly but the protocols that are written on top of IP
  - Transmission Control Protocol (TCP/IP)
  - User Datagram Protocol (UDP/IP)

TCP versus UDP

Transmission Control Protocol (TCP/IP)

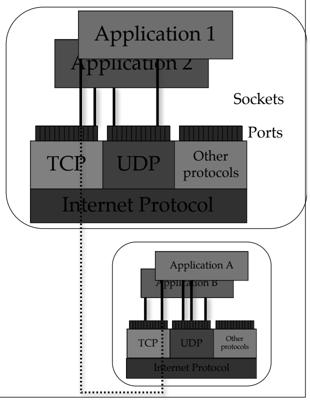
- Point-to-point connection
- Reliable transmission using acknowledgement and retransmission
- Stream-based data semantics
- Big overhead
  - data checksums
- Hard to ‘skip ahead’

User Datagram Protocol (UDP/IP)

- Lightweight data transmission
- Differs from TCP
  - connectionless transmission
  - ‘best-efforts’ delivery
  - packet-based data semantics
- Packets are easy to process
- Transmission and receiving immediate
- No connection information for each host in the operating system
- Packet loss can be handled

The BSD Sockets Architecture

- A socket is a software representation of the endpoint to a communication channel
- Reliable/unreliable communication, single/multiple destinations, etc.
- Includes several pieces of information, such as
  - protocol
  - destination host and port
  - source host and port



BSD = Berkeley Software Distribution

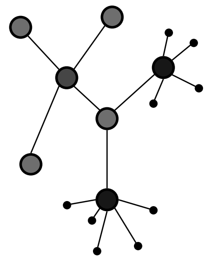
Transmission Techniques

- Unicasting
  - single receiver
- Multicasting
  - one or more receivers that have joined a multicast group
- Broadcasting
  - all nodes in the network are receivers

IP Broadcasting

- Using a single UDP/IP socket, the same packet can be sent to multiple destinations by repeating the send call
  - 'unicasting'
  - great bandwidth is required
  - each host has to maintain a list of other hosts
- IP broadcasting allows a single transmission to be delivered to all hosts on the network
  - a special bit mask of receiving hosts is used as a address
- With UDP/IP, the data is only delivered to the applications that are receiving on a designated port
- Broadcast is expensive
  - each host has to receive and process every broadcast packet
- Only recommended (and only guaranteed) on the local LAN
- Not suitable for Internet-based applications

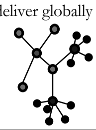
IP Multicasting 1 (3)



- Packets are only delivered to subscribers
- Subscribers must explicitly request packets from the local distributors
- No duplicate packets are sent down the same distribution path
- Original 'publisher' does not need to know all subscribers
- Receiver-controlled distribution

IP Multicasting 2 (3)

- 'Distributors' are multicast-capable routers
- They construct a multicast distribution tree
- Each multicast distribution tree is represented by a pseudo-IP address (multicast IP address, class D address)
  - 224.0.0.0–239.255.255.255
  - some addresses are reserved
  - local applications should use 239.0.0.0–239.255.255.255
- Address collisions possible
  - Internet Assigned Number Authority (IANA)
- Application can specify the IP time-to-live (TTL) value
  - how far multicast packets should travel
  - 0: to the local host
  - 1: on the local LAN
  - 2–31: to the local site (network)
  - 32–63: to the local region
  - 64–127: to the local continent
  - 128–254: deliver globally



IP Multicasting 3 (3)

- Provides desirable network efficiency
- Allows partitioning of different types of data by using multiple multicast addresses
- The players can announce their presence by using application's well-known multicast address
- Older routers do not support multicasting
- Multicast-aware routers communicate directly by 'tunneling' data past the non-multicast routers (Multicast Backbone, Mbone)
  - Participant's local router has to be multicast-capable

Selecting a Protocol 1 (4)

- Multiple protocols can be used in a single system
- Not which protocol should I use in my game but which protocol should I use to transmit *this piece of information*?

- Using TCP/IP
  - reliable data transmission between two hosts
  - packets are delivered in order, error handling
  - relatively easy to use

  - point-to-point limits its use in large-scale multiplayer games
  - bandwidth overhead

Selecting a Protocol 2 (4)

- Using UDP/IP
  - lightweight
  - offers no reliability nor guarantees the order of packets
  - packets can be sent to multiple hosts
  - deliver time-sensitive information among a large number of hosts
  - more complex services have to be implemented in the application (serial numbers, timestamps)
  - recovery of lost packets
    - positive acknowledgement scheme
    - negative acknowledgement scheme (more effective when the destination knows the sources and their frequency)
  - transmit a quench packet if packets are received too often

Selecting a Protocol 3 (4)

- Using IP broadcasting
  - design considerations similar to (unicast) UDP/IP
  - limited to LAN
  - not for games with a large number of participants
  - to distinguish different applications using the same port number (or multicast address):
    - Avoid the problem entirely: assign the necessary number
    - Detect conflict and renegotiate: notify the participants and direct them to migrate a new port number
    - Use protocol and instance magic numbers: each packet includes a magic number at a well-known position
    - Use encryption

Selecting a Protocol 4 (4)

- Using IP multicasting
  - provides a quite efficient way to transmit information among a large number of hosts
  - information delivery is restricted
    - time-to-live
    - group subscriptions
  - preferred method for large-scale multiplayer games
  - how to separate the information flows among different multicast groups
    - a single group/address for all information
    - several multicast groups to segment the information