



# Sockets and Networking

4/10/2007





# Opening Discussion

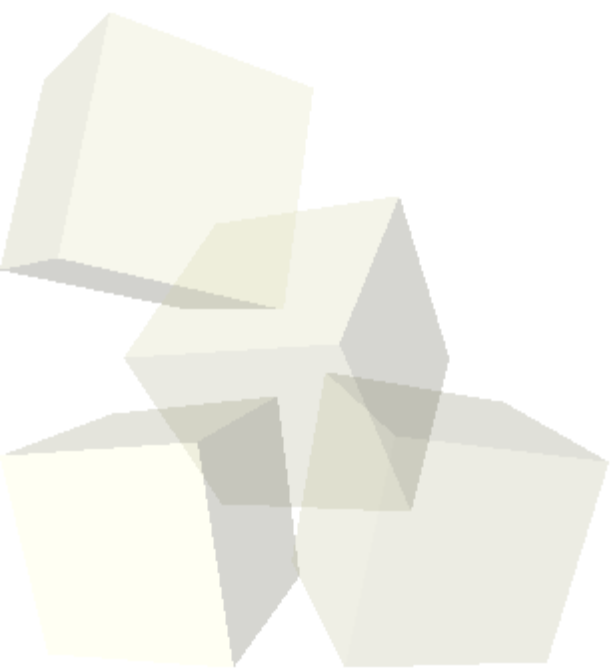
- What types of applications might use UDP instead of TCP? How do you have to change your programming style when you use UDP?
- Do you have any questions about the assignment?

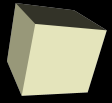




# A Chat Program

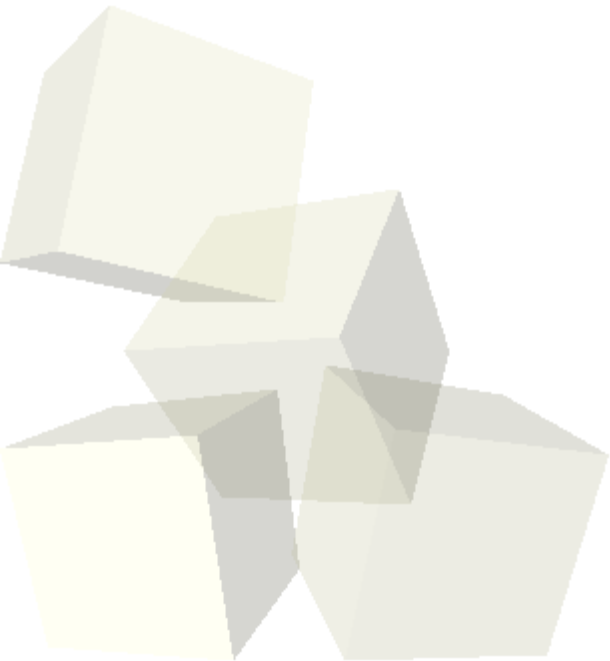
- I want you to write a chat program. Let's work through this together with one of you doing the typing. We can do something really basic and either do a pier-to-pier or a client-server model. I like the client-server model where multiple users can connect to the server.





# Sending Drawings

- Now we can play a little bit with our drawing program and get it so that we can send drawings from one computer to another.





- When you implemented the priority queue for your game you used a sorted linked list. This data structure has the disadvantage that adding takes  $O(n)$  time. For data structures we try to avoid  $O(n)$  if at all possible for the common operations because it typically leads to  $O(n^2)$  overall performance.
- There is another data structure we can use to implement that priority queue called a heap. Using a heap none of the operations will be worse than  $O(\log n)$ .
- There are actually many types of heaps. We will be doing the simplest type, the binary heap.



# Binary Heaps as Trees

- You can view a binary heap as a type of binary tree, though our final implementation will not actually use a binary tree.
- Binary heaps are complete trees that have heap ordering. A complete tree is one that only starts filling the next row after the one above it is completely full. We will fill rows from left to right adding in new nodes until we have to move down to the next row.
- Heap ordering is simply the property that each node has a higher priority than its children. So in your game each node would have an entity with a lower update time than the children.



# Adding and Removing

- When we add to a binary heap we simply place the new element at the next free spot on the tree. We then have to shift things to restore heap ordering. We refer to the new node as a bubble which moves up through the tree until it comes to rest.
- The only item we can remove from the tree is the root, which has the highest priority thanks to the heap ordering. To keep the heap complete we move the last element up to the root and let it sink down through the tree until it comes to rest.
- Both of these operations are always  $O(\log n)$  because the heap is complete.



# Heaps as Arrays

- The tree structure is nice for visualizing the heap, but it is extremely inefficient for implementation.
- If we order the nodes starting at 1 you will see nice mathematical relationships between parents and children. We use this numbering as an index into a array and store the heap in an array or an array based list.







- How much do you think using a heap would improve the performance of your game?
- I will be around from 2-5 today except when I'm advising.
- Remember that there is no class on Thursday.
- Assignment #6 is due on Thursday.

