Sockets and Networking

4/22/2008

Opening Discussion

- Let's look at some solutions to the interclass problem.
- Do you have any questions about the assignment?

Power of Serialization

- Now we can take the next step. I want our drawing application to have the ability to save and load full drawing. What do we need to change in the code to make this happen? We basically have to take the entire object for our tree and write it out to file one element at a time.
- The task of converting an object into a stream of bytes is called serialization. In most languages it is a tough thing to do. Fortunately, Java has built in functionality to provide serialization.

Serializing in Java

- To make it so that an object can be serialize we simply inherit from the interface Serializable. This is a “mix-in” interface that doesn't have any methods.
- The ObjectOutputStream and ObjectInputStream can be used to write and read whole objects that are Serializable. If it, or some part of it, isn't Serializable an exception will be thrown.
- Elements that you don't want written (or that can't be written) can be labeled as transient.

Challenges

- “With great power comes great responsibility.”
- Serialization is truly powerful, but you shouldn't just make everything Serializable because there are costs.
- Anything that inherits from a Serializable class/interface is itself Serializable.
- The default serialization can be expensive and potentially leads to security holes where people can find out about details of your objects that are otherwise private.

Networking

- You all understand the power of networking. When your computer isn't plugged into the internet you feel like it has lost a lot of its functionality.
- So many of the application that you use regularly use the network and a lot of the time you aren't even aware of it. It has simply become transparent.
- Given this, you need to know how to write networking code. Fortunately, Java makes it fairly easy with the java.net package.

Sockets

- When computers talk to one another, they do so through sockets. There are different types of sockets. The java.net package includes classes for TCP sockets and UDP sockets.
- Most communication is done with TCP sockets where the socket have something of a client/server model. A server waits on a port on the computer and clients can connect to that port. Each side “talks” through a socket. With TCP there is error checking so if the message doesn't get through, you are told about it.
- UDP is used less frequently because it doesn't tell you if messages are lost. It is faster though.

Using Networking

- Now you get to really see the power of the streaming model. Sockets in Java have methods that give you input and output steams. All the things we did with files can be done with networking. In fact, if you write code to work with an InputStream or an OutputStream, you can give it a file stream or a network stream and it won't care.
- Let's add a bit of networking code to our application. The goal would be to have the ability to do collaborative drawing work. This means we need to be able to send chat messages and drawings.

Minute Essay

- Java makes socket communication rather simple to do. What are some of the challenges that you can see arising with the use of sockets?
- Assignment #6 is due on today.
- Interclass Problem – Write a simple chat room client-server program. You can have it use console I/O instead of a GUI.