The New QuizMaster: Teaching An Old Dog New Tricks

by Willie George
The Current QuizMaster

• Student Response System
• Internet-based “Ask the Audience”
• Similar to:
  – Smartroom’s “Beyond Question”
  – Sun-tech’s “XPRESS”
  – InterWrite’s “PRS (Personal Response System)"
• Written entirely in Sun’s Java programming language
The History

• First developed as the Senior Seminar project of Philip Smith
  • Version 1.0
• Taken over and updated several times by Allen Smith (no relation)
  • Versions 1.01 – 3.71
The Background

• QuizMaster uses java.net.Socket for communication
  - TCP protocol connection
  - Lets the network stack tend to verification that the bytes that are sent arrive at their destination
  - Easily wrapped with standard InputStream and OutputStream classes
  - Drastic improvement from the originally used java Datagrams
The Background

- Most communication is done through passing of DataPacket objects
  - DataPacket is a custom class implemented in version 2.0 by Allen Smith
  - Another major improvement from version 1.0
• QuizMaster had a major limitation
  – Connection limit on simultaneous incoming requests
  – Only on Microsoft Windows Home or Professional
  – Location of Windows server not conducive to interactive quizzing
The First Attempt

- Rewrite StreamConnection and ClientConnectionThread classes to communicate through a proxy
The First Attempt

- Rewrite StreamConnection and ClientConnectionThread classes to communicate through a proxy
- First big snag:

```java
if (acceptDialogChoice == ACCEPT){
    System.out.println("ACCEPT");
    connections.addElement((incomingConnection);
    incomingConnection.start(); //Fire up this thread to start receiving
    int playerIndex = getNumberOfPlayers() - 1;
    connections(playerIndex) setPlayerID(playerIndex);
    System.out.println(playerIndex);
    //this is one of very few places DataPackets are not used
    connections(playerIndex) writeInt(playerIndex);
    System.out.println("Server writes int: "+ playerIndex);
    parent.status.addPlayer(connections(playerIndex).getPlayerName(), playerIndex);
    parent.finalReport.addPlayer("acceptConnections"); //if acceptConnections is false, that will fl
    setEnabled(playerIndex, true);
    //Only indicate a new connection if the quiz HASN'T started; otherwise, the question text will f
    if(acceptConnections == true){
        parent.questionArea.setText("Please begin connecting to "+
            InetAddress.getHostAddress()+"\n\n" +
            "Connections Established: " + getNumberOfPlayers());
```
The Second Attempt

• Byte level proxy
• Still needed to eliminate the connection problem
public void run() {
    String line = null;
    boolean hasRequestedQuit = false;
    try {
        BufferedReader stdin = new BufferedReader(new InputStreamReader(System.in));
        while (!hasRequestedQuit) {
            line = stdin.readLine();
            if (line != null) {
                // Display the input
                display(line);
            }
        }
    } catch (IOException ex) {
        System.err.println(ex);
    }
}

private static void display(String line) {
    System.out.println(line);
}

public static void main(String[] args) {
    // Schedule a shutdown
    stdin = new BufferedReader(new InputStreamReader(System.in));
    while (true) {
        display(stdin.readLine());
    }
}

private static void display(String line) {
    System.out.println(line);
}
References

• Allen Smith – Comments in the existing QuizMaster code

java.sun.com – Extensive online documentation for the Java platform 1.4.2 Software Development Kit

• Kevin Hieb – Reminding me how to use Vectors in Java