The New QuizMaster: Teaching An Old Dog New Tricks

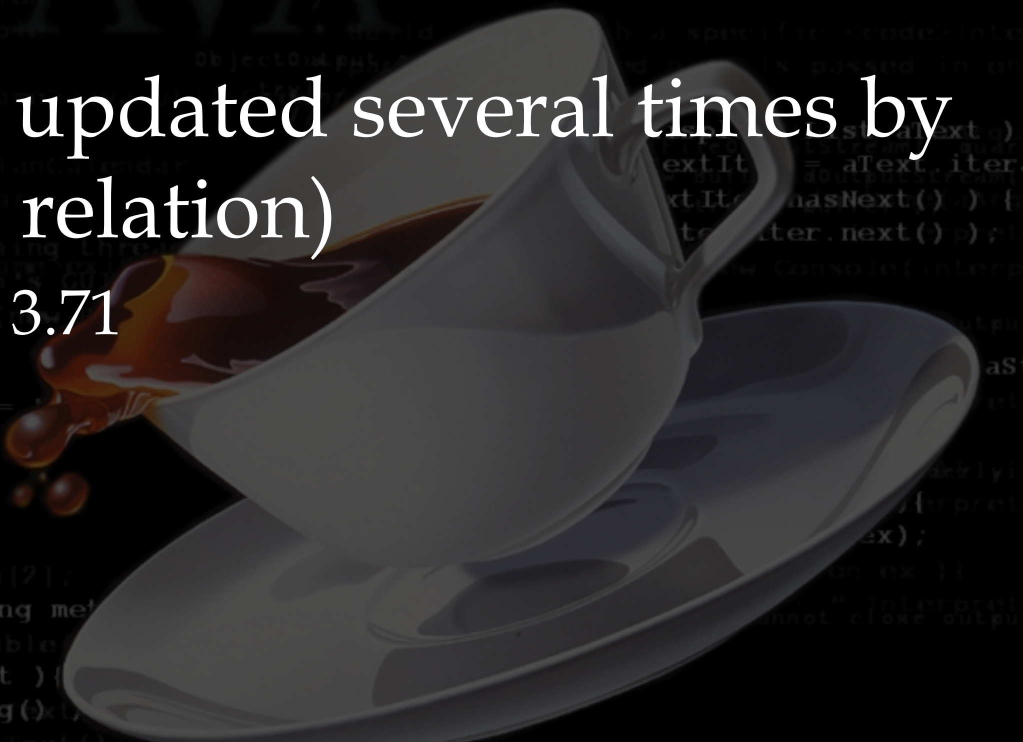
by Willie George

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

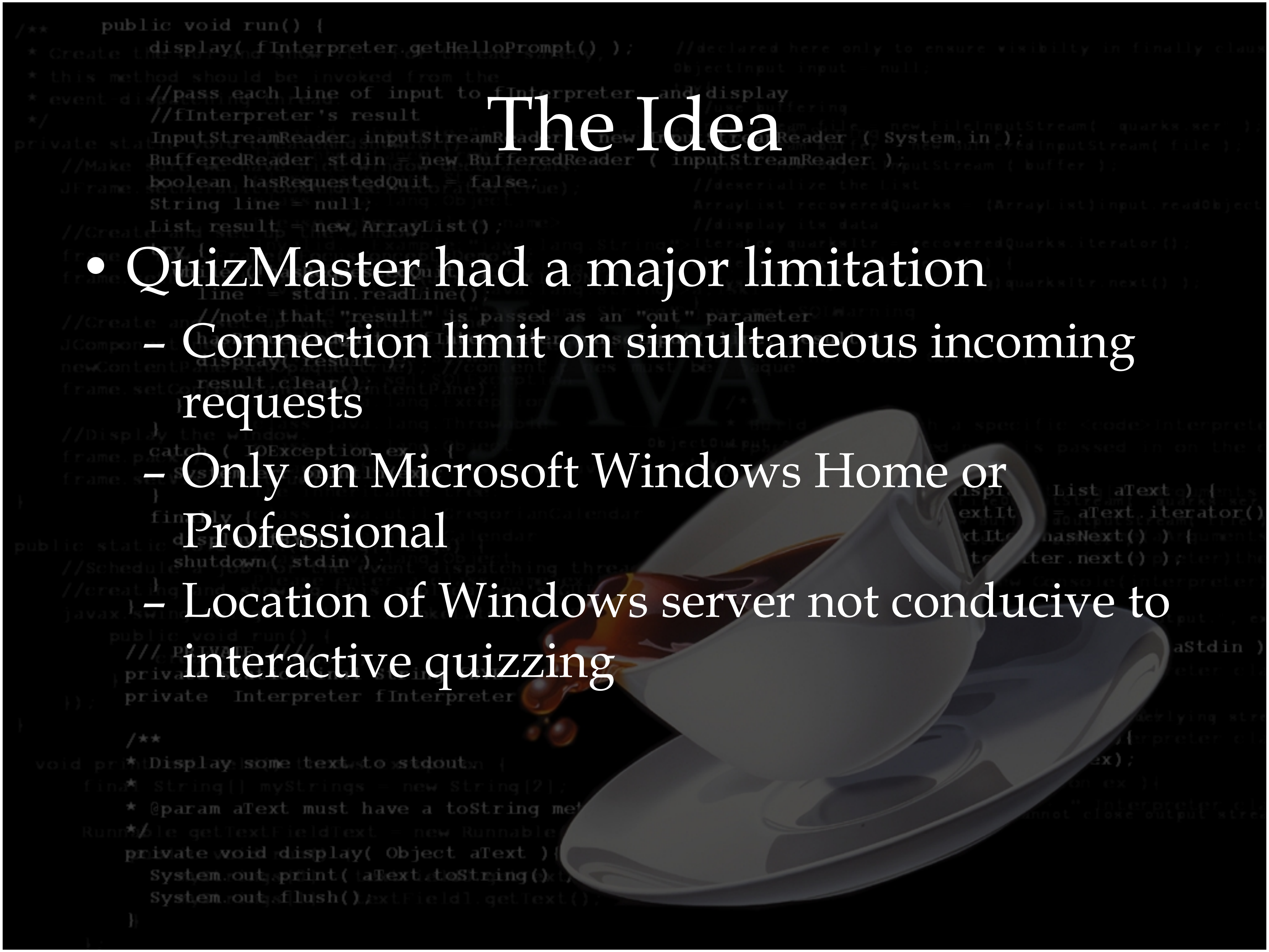


The Background

- 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

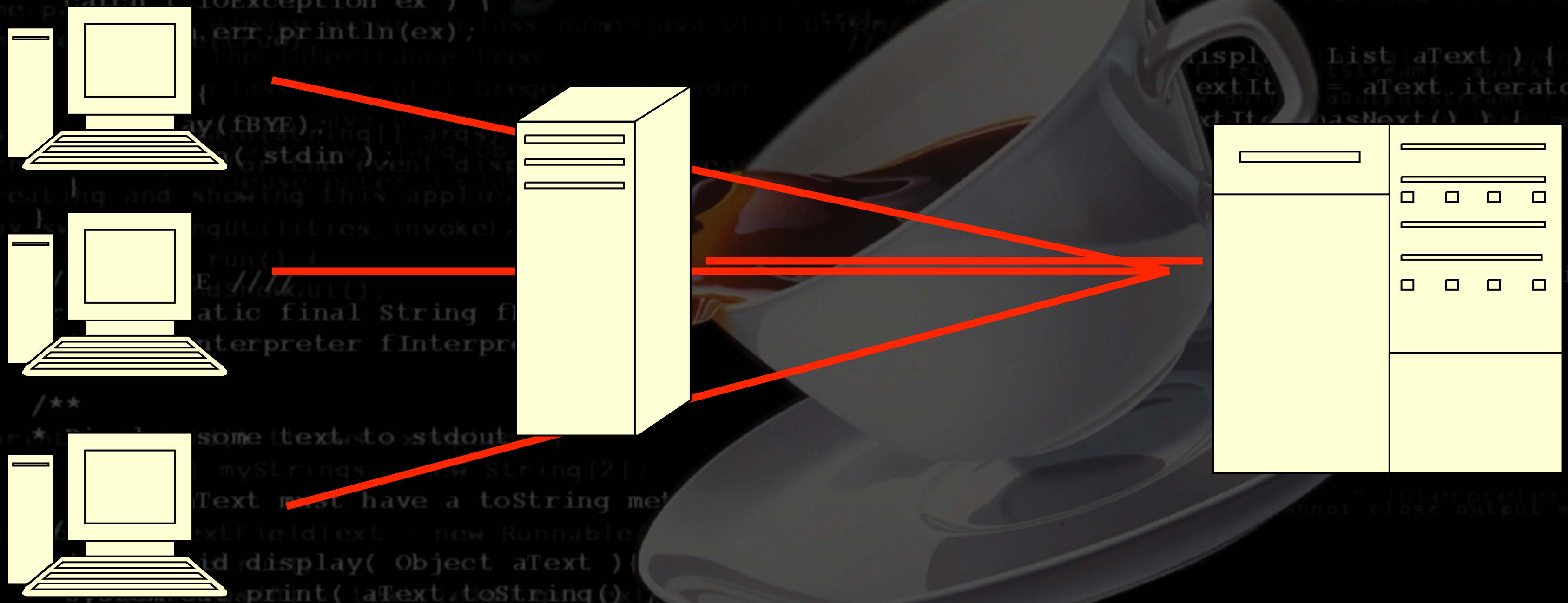


The Idea

- 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

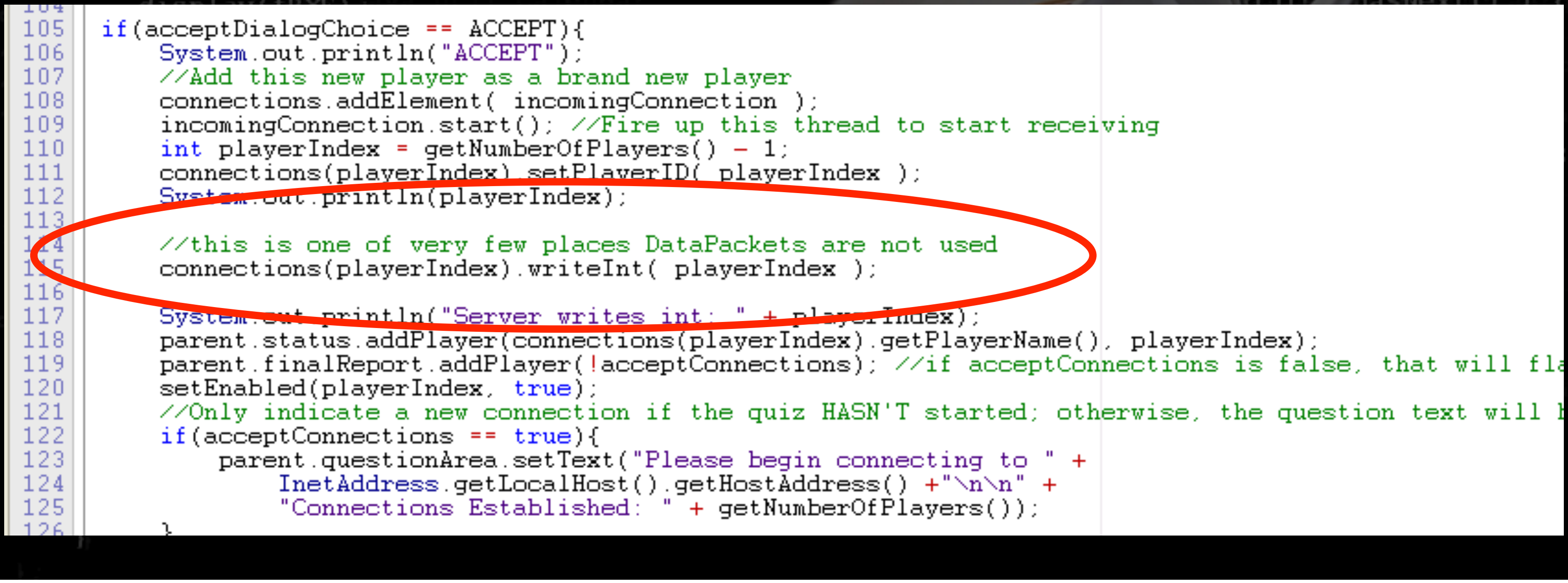
The First Attempt

- 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:



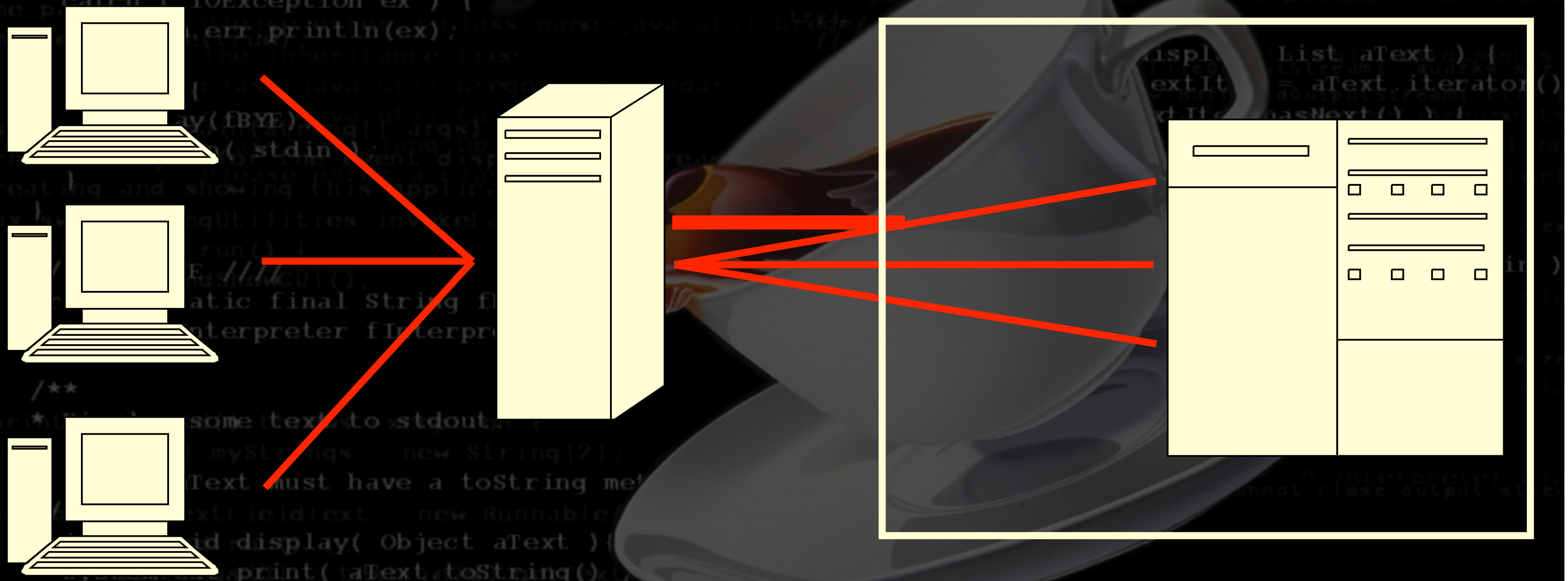
```
if(acceptDialogChoice == ACCEPT){
    System.out.println("ACCEPT");
    //Add this new player as a brand new player
    connections.addElement( incomingConnection );
    incomingConnection.start(); //Fire up this thread to start receiving
    int playerIndex = getNumberOfPlayers() - 1;
    connections(playerIndex).setPlayerID( 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 fla
    setEnabled(playerIndex, true);
    //Only indicate a new connection if the quiz HASN'T started; otherwise, the question text will
    if(acceptConnections == true){
        parent.questionArea.setText("Please begin connecting to " +
            InetAddress.getLocalHost().getHostAddress() +"\n\n" +
            "Connections Established: " + getNumberOfPlayers());
    }
```

The Second Attempt

- Byte level proxy
- Still needed to eliminate the connection problem



QuizMaster Demo

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