

## Java Applets

Nan Niu ([nn@cs.toronto.edu](mailto:nn@cs.toronto.edu))  
CSC309 -- Summer 2008

## Java History

- Created by James Gosling et al. at Sun Microsystems in 1991 "The Green Team"



- Gosling created a processor-independent language for "StarSeven", a 2-way wireless remote-control device
  - Called the language 'Oak'



2

## Java History (cont'd)

- Could not find a market for the technology.
- During a Sun offsite with Bill Joy in 1994:
  - "why not use it on the Internet?"
  - Started development of "WebRunner", later to be renamed "HotJava" browser
    - a browser capable of downloading and running Java bytecode.
- Folks were impressed with interactive Web pages
  - e.g., spinning molecules, sorting demos
  - many downloads of HotJava
    - a big success
- Marc Andersen of Netscape agrees to support Java in Netscape browsers in 1995

3

## Applet Execution

- A JVM (Java Virtual Machine) running within the context of the client browser loads and runs the applet classes.
- Applet is not trusted
  - Limited access to system resources (file system, network)

4

## How Java Applets Work

- `<embed>`, `<applet>`, or `<object>`
- Java ".jar" files downloaded to client machine
  - contains
    - class files
    - other resources (images, sounds, property files, ...)
- Class indicated in the tag (must derive from `java.applet.Applet`) is invoked

5

## A Trivial Applet

- Old (and convenient way) of invoking applets (invokes built-in jre)
  - [TrivialApplet.html](#)
- Official (new way) of invoking applets (invokes SUN's jre1.5 plug-in)
  - [TrivialObject.html](#)

6

## TrivialApplet.html

```
<html>
<head>
<title>Applet Example</title>
</head>

<body>
<h1>Below the line is an applet</h1>
<hr/>
<applet code="Trivial.class" width="300" height="100"/>
</body>
</html>
```

7

## TrivialObject.html

```
<html>
<head>
<title>Applet Example Firefox Version</title>
</head>
<body>
<h1>Below the line is an applet</h1>
<hr/>
<object classid="java:Trivial.class"
type="application/x-java-applet"
height="100" width="300" />
</body>
</html>
```

8

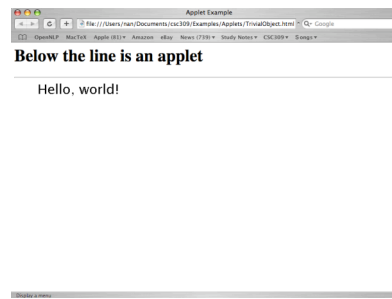
## Trivial.java

```
import java.applet.Applet;
import java.awt.Graphics;
import java.awt.Font;

public class Trivial extends Applet {
    public void paint(Graphics g) {
        g.setFont( new Font("Default", Font.PLAIN,
28) );
        g.drawString("Hello, world!", 50, 25);
    }
}
```

9

## Result



10

## java.applet.Applet lifecycle methods

- void init()
  - invoked when applet is initially loaded
- void start()
  - invoked when started or restarted as a result of user flipping Web pages
- void stop()
  - invoked when user switches away from Web page
- void destroy()
  - invoked (usually) on browser termination

11

## Applet contextual methods

- String getParameter(String)
- AudioClip getAudioClip(URL)
- Image getImage(URL)
- URL getCodeBase()
- URL getDocumentBase()
- void showStatus(string)

12

## Applet GUI

- Inherit from Applet and use the AWT GUI library
- Inherit from JApplet and use the JFC/Swing GUI
  - [GUI.html](#)

13

## GUI.html

```
<html>
<head>
<title>Applet Example</title>
</head>

<body>
<h1>Between the lines is an applet</h1>
<hr>
<applet code="GUI.class" width="330" height="370">
  <param name="image" value="bear.jpg">
  <param name="sound" value="bearsound.au">
</applet>
<hr>
</body>
</html>
```

14

## Java/JavaScript Communications

- JavaScript can call methods defined in Java Applets
  - [JS.html](#)
- JavaScript vs. Applets

15

## JS.java

```
import java.applet.*;
import java.awt.*;

public class JS extends Applet {
  String text = "error";

  public void init() {
    text = getParameter("text");
  }

  public void paint(Graphics g) {
    int style = Font.BOLD + Font.ITALIC;
    g.setFont(newFont("TimesRoman", style, 36));
    g.drawString(text, 50, 50);
  }

  public String getText() {
    return text;
  }
}
```

16

## JS.html

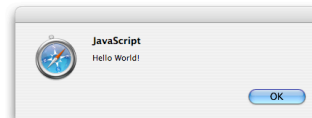
```
<html>
<head>
<title>Test Applet</title>
</head>
<body>
<h1>This is a test of applets</h1>
<hr/>
<applet name="jsapplet" code="JS.class" height="300" width="300">
  <param name="text" value="Hello World!"/></param>
  Text displayed by non-java enabled browsers
</applet>
<hr/>
<form>
  <input type="button"
    onclick="alert(document.jsapplet.getText())"
    value="Get Data From Applet">
  </form>
</body>
</html>
```

17

## Result

This is a test of applets

*Hello World!*



Get Data From Applet

18

## 10 Reasons for Applet's Downfall

1. Permission pop-up
2. Coarse-grain whole-sale permission levels
  - Network ,file systems
3. All applets need to be analyzed for potential security threat
  - If network or file system permission is to be granted
  - Third-party security analysis
    - 3 browsers vs. 1000s applets?
    - Buffer overflow, unauthorized access, general exploits

19

## 10 Reasons (cont'd)

4. Trust issues
  - Privacy conscious?
  - Bug free?
5. Certificate signing
  - Authority chain: is the certificate trust-worthy?
  - Complexity for both developer and end user
6. User cannot easily inspect code
7. Required to install JVM on client

20

## 10 Reasons (cont'd)

8. Critical business logic should run on server
  - To ensure users do not interfere with business assumptions by hacking the applet
  - Servlet is a better choice
9. Graphic capabilities can be supplied via new content types and browser plug-ins
  - Flash
10. Ajax provides more superb interaction between client and server

21