CS1007 lecture #16 notes

tue 26 mar 2002

- news
- networks
- applets
- graphics
- reading: ch 1.3, 1.6, 2.9-2.10, 4.7, appendix J

news.

- homework #4 due: THU MARCH 28
- homework #3 — tournament results next class!
- midterm #2 changed to: TUE APRIL 9

networks (1).

- two or more computers connected to each other
- networked computers can share information
- and resources, e.g.:
  - printer
  - file server
- example: CUNIX system
- connections:
  - *point-to-point* — computers are directly connected to each other
    - message speed is fast
    - adding computers is expensive
  - single communication line
    - message speed can be slow
    - adding computers is cheap

networks (2).

- *network address*
  - uniquely identifies each computer on the network
- *packet*
  - long messages are split into pieces
  - each piece is a packet, sent individually along the network
  - improves message speed
- *local-area network* (LAN)
  - designed to span short distances
- *wide-area network* (WAN)
  - designed to span longer distances
  - connects multiple LANs

networks (3).

- the *Internet*
  - developed in the 1970s as ARPANET
  - the ultimate WAN — a network of networks
- *protocol*
  - set of rules governing communication
  - TCP/IP (transmission control protocol / internet protocol)
- IP address = network address on the Internet
  - numeric, e.g., 204.192.116.2
  - also have text equivalents called *Internet addresses*, which are comprised of local computer names (i.e., name of computer on LAN) plus domain names (i.e., name of LAN on WAN)
  - domain names are controlled by the Internet Naming Authority
- *domain name system* (DNS)
  - translates between IP address and Internet address

networks (4).

- the *World Wide Web* (WWW)
- provides standard method of interfacing to the Internet from the user level
- uses *hypertext*
  - non-linear method of organizing information
  - refers only to textual information
- *hypermedia*
  - refers to non-textual information, such as sound, video and graphics
- *browser*
  - user program that provides method of viewing WWW documents
  - early browsers included: Archie, Gopher
  - *Mosiac*
    - first graphical WWW browser
    - released in 1993
    - became Netscape

networks (5).

- *HyperText Markup Language* (HTML)
  - standard format for WWW documents
- *Uniform Resource Locator* (URL)
  - unique document address on the WWW
- *HypterText Transfer Protocol* (http)
  - protocol used for transfering HTML documents
  - provides *one-way* transfer from *server* to *client*
- other protocols include: ftp, telnet
  - these provide *two-way* transfer between *server* and *client*
- Java
  - grew out of the above
  - allows *two-way* transfer
  - text, graphics, sound

networks (6).

- *client-server* architecture
- comes from operating system design
- methodology by which tasks are divided onto different processors according to functionality
- programs can be divided into:
  - computation portion
  - drawing or output portion
- each portion can be executed on a different CPU
- X windows
  - windowing system used under UNIX
  - with X windows, the drawing is done on the *client*, although the execution may be happening on a different physical machine, the *server*

applets (1).

- Java programs can run as *applications* or *applets*
- *application:*
  - executed using the *java* command
  - server and client can be the same machine or different machines
  - client invokes JVM which interprets classes and runs them
- *applet:*
  - must be executed using a browser, like Netscape, or the *appletviewer* command
  - server sends applet to the client, in the form of class files; applet invokes JVM which interprets classes and runs them on the client
  - there are two parts:
    - an HTML file used to invoke the applet
    - Java class file(s) that contain the applet code

applets (2).

- file name = hi.html

```
<html>
<title>
sample applet page
</title>

the applet will be shown below...

<applet code="hi.class" width=400 height=400>
</applet>

</html>
```


applets (3).

- file name = hi.java

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

public class hi extends Applet {

  public void paint( Graphics g ) {

    g.drawString( "hi",10,10 );

  } // end of paint()

} // end of class hi
```


applets (4).

- `java.awt` package
  - *Abstract Windowing Toolkit* (AWT)
  - classes that support graphical user interfaces (GUI)
  - includes `java.awt.Component` method:
    - `public void paint()`
- `java.applet.Applet` class
  - `public void init()`
  - `public void start()`
  - `public void stop()`

graphics.

- `java.awt.Graphics` class
- X-windows coordinate system
- drawing primitives:
  - lines
  - Strings
  - rectangles
  - ovals
  - arcs
- color

graphics (2).

- simple methods from the `java.awt.Graphics` class
  - `void drawLine( int x1, int y1, int x2, int y2 );`
    - draws a line connecting (x1,y1) and (x2,y2);
  - `void drawString( String str, int x, int y );`
    - draws the text in "str", with its lower left corner at (x,y)

graphics (3).

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

public class ex16_1 extends Applet {

  public void paint ( Graphics g ) {
    g.drawString( "hello world!",10,10 );
    g.drawLine( 0,400, 400,0 );
  } // end of paint()

} // end of class ex16_1()
```


graphics (4).

- bounding rectangles
  - coordinates of origin (upper left corner)
  - extent (width and height)
- arcs
  - measured in degrees
  - starting from 0° (along positive X-axis, like hw#3)
  - extent (total angle of arc)

graphics (5).

- more methods from the `java.awt.Graphics` class
  - `void drawRect( int x, int y, int width, int height );`
    - draws a rectangle with its upper left corner at (x,y), extending the specified "width" and "height"
  - `void drawOval( int x, int y, int width, int height );`
    - draws an oval circumscribed in the bounding rectangle with its upper left corner at (x,y), extending the specified "width" and "height"
  - `void drawArc( int x, int y, int width, int height, int startAngle, int arcAngle );`
    - draws an arc whose oval is circumscribed in the bounding rectangle with its upper left corner at (x,y), extending the specified "width" and "height", where the arc starts at the "startAngle", measured in degrees (where 0°) is horizontal along the positive x-axis), extending for "arcAngle" degrees

graphics (6).

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

public class ex16_2 extends Applet {

  public void paint ( Graphics g ) {
    g.drawRect( 10,300,25,25 );
    g.drawOval( 10,250,25,25 );
    g.drawArc( 10,200,25,25,45,90 );
  } // end of paint()

} // end of class ex16_2()
```


graphics (7).

- `java.awt.Color` class
- color is defined using the "RGB" methodology
- "Red", "Green", "Blue"
- each is an integer between 0 and 255, where 0 means no color and 255 means maximum color
- so white is: red=255 green=255 blue=255 or the ordered triple (255,255,255)
  - and black is: red=0 green=0 blue=0
  - and red is: red=255 green=0 blue=0
  - and green is: red=0 green=255 blue=0
  - and blue is: red=0 green=0 blue=255
- make up your own colors...

graphics (8).

- even more methods from the `java.awt.Graphics` class
  - `void setColor( Color color );`
    - sets the foreground (pen) color to the specified color
  - `void fillRect( int x, int y, int width, int height );`
    - draws a filled rectangle with its upper left corner at (x,y), extending the specified "width" and "height"
  - `void fillOval( int x, int y, int width, int height );`
    - draws a filled oval circumscribed in the bounding rectangle with its upper left corner at (x,y), extending the specified "width" and "height"
  - `void fillArc( int x, int y, int width, int height, int startAngle, int arcAngle );`
    - draws a filled arc whose oval is circumscribed in the bounding rectangle with its upper left corner at (x,y), extending the specified "width" and "height", where the arc starts at the "startAngle", measured in degrees (where 0°) is horizontal along the positive x-axis), extending for "arcAngle" degrees