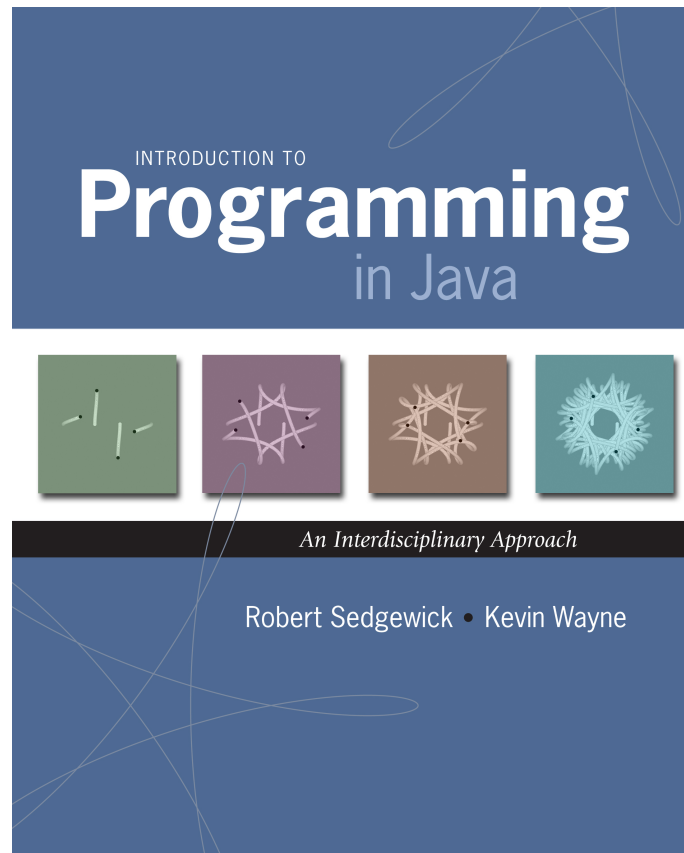


1.5 Input and Output (II)



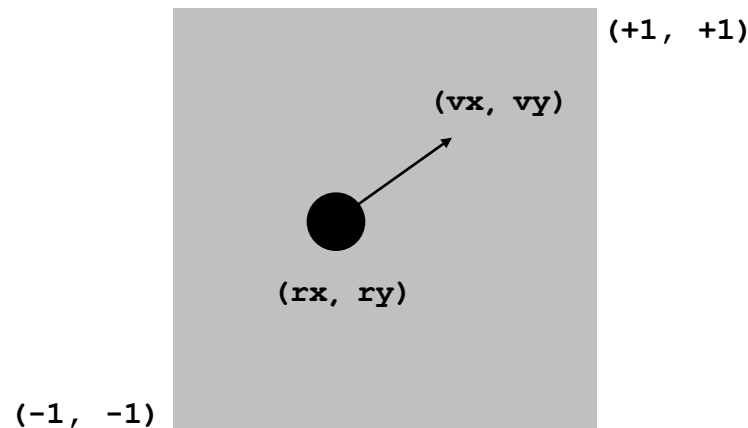
Animation

Animation loop. Repeat the following:

- Clear the screen.
- Move the object.
- Draw the object.
- Display and pause for a short while.

Ex. Bouncing ball.

- Ball has position (rx, ry) and constant velocity (vx, vy) .
- Detect collision with wall and reverse velocity.



Bouncing Ball

```
public class BouncingBall {  
    public static void main(String[] args) {  
        double rx = .480, ry = .860;  
        double vx = .015, vy = .023;  
        double radius = .05;  
  
        StdDraw.setXscale(-1.0, +1.0);  
        StdDraw.setYscale(-1.0, +1.0);
```

position
constant velocity
radius

rescale coordinates

```
while(true) {  
    if (Math.abs(rx + vx) + radius > 1.0) vx = -vx;  
    if (Math.abs(ry + vy) + radius > 1.0) vy = -vy;
```

bounce

```
    rx = rx + vx;  
    ry = ry + vy;
```

update position

```
    StdDraw.setPenColor(StdDraw.GRAY);  
    StdDraw.filledSquare(0.0, 0.0, 1.0);  
    StdDraw.setPenColor(StdDraw.BLACK);  
    StdDraw.filledCircle(rx, ry, radius);  
    StdDraw.show(20);
```

clear background

draw the ball

```
    }
```

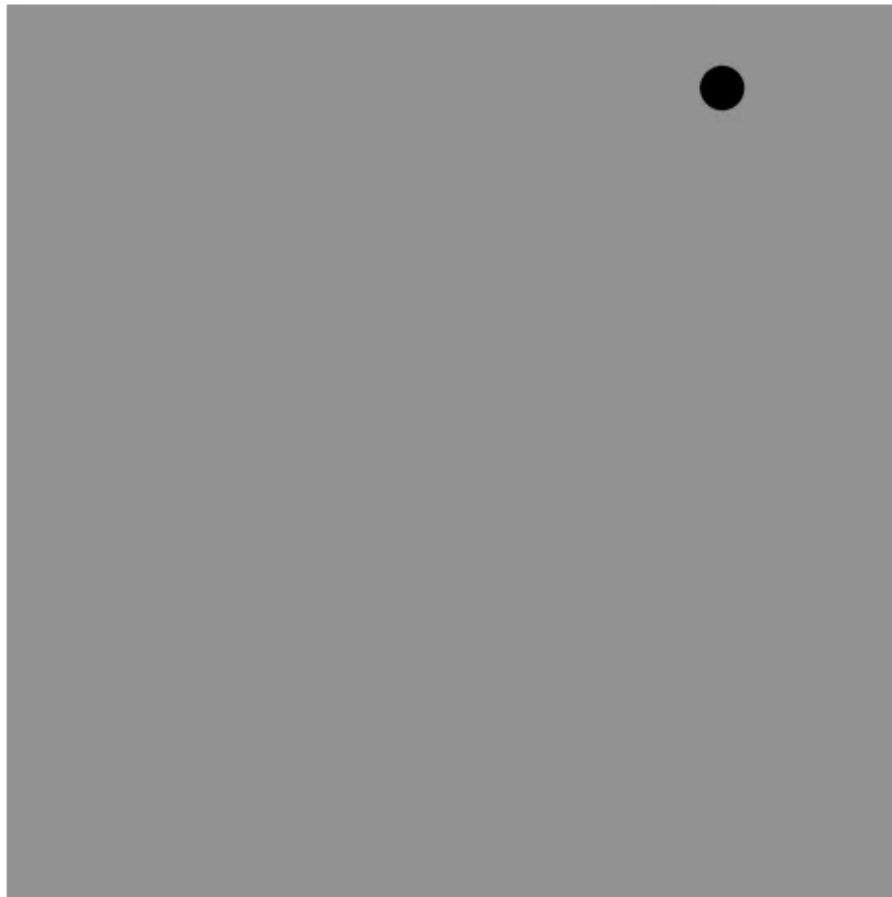
```
}
```

```
}
```

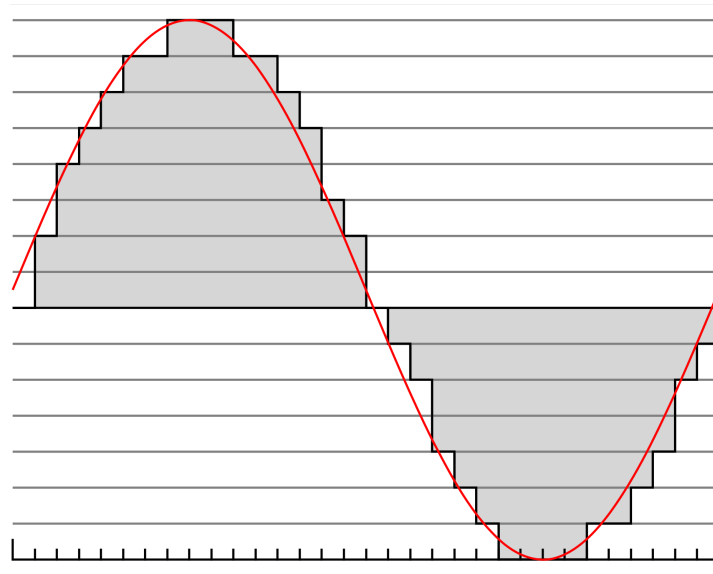
turn on animation mode:
display and pause for 20ms

Bouncing Ball Demo

```
% java BouncingBall
```



Standard Audio

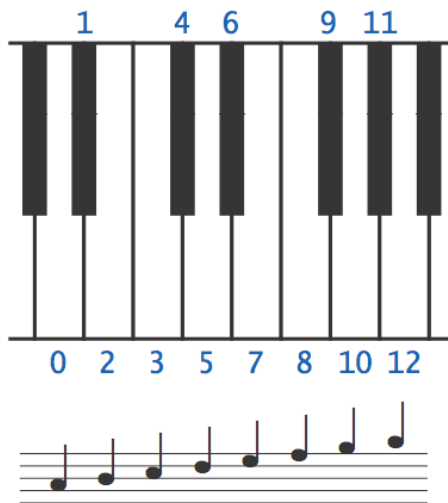


Crash Course in Sound

Sound. Perception of the **vibration** of molecules in our eardrums.

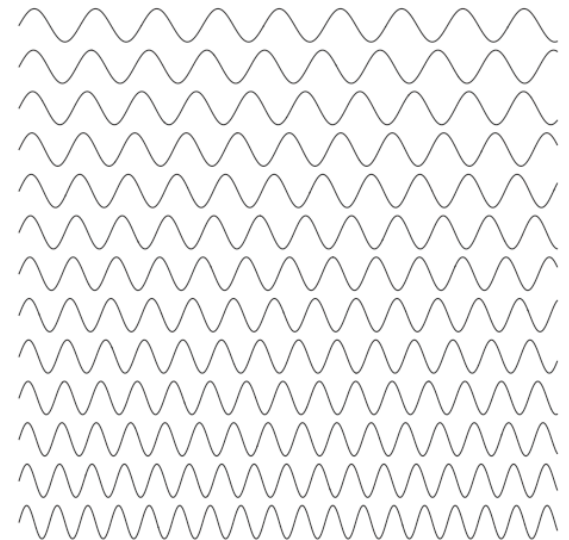
Concert A. Sine wave, scaled to oscillate at 440Hz.

Other notes. 12 notes on chromatic scale, divided logarithmically.



<i>note</i>	<i>i</i>	<i>frequency</i>
A	0	440.00
A# or B _b	1	466.16
B	2	493.88
C	3	523.25
C# or D _b	4	554.37
D	5	587.33
D# or E _b	6	622.25
E	7	659.26
F	8	698.46
F# or G _b	9	739.99
G	10	783.99
G# or A _b	11	830.61
A	12	880.00

$440 \times 2^{i/12}$



Notes, numbers, and waves

Digital Audio

Sampling. Represent curve by sampling it at regular intervals.

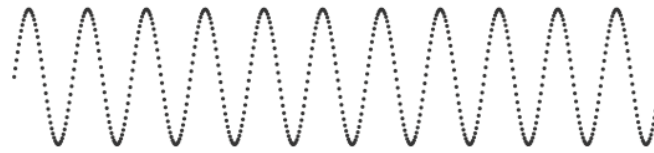
5,512 samples/second, 137 samples



11,025 samples/second, 275 samples

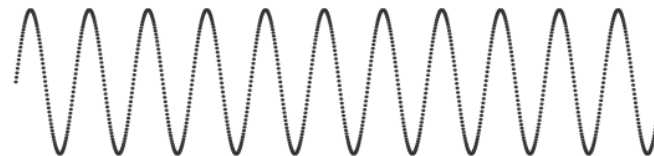


22,050 samples/second, 551 samples



44,100 samples/second, 1,102 samples

audio CD



$$y(i) = \sin\left(\frac{2\pi \cdot i \cdot 440}{44,100}\right)$$

Digital Audio in Java

Standard audio. Library for playing digital audio.

```
public class StdAudio
```

```
    void play(String file)
```

play the given .wav file

```
    void play(double[] a)
```

play the given sound wave

```
    void play(double x)
```

play sample for 1/44100 second

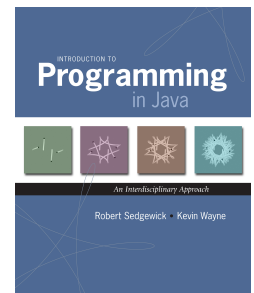
```
    void save(String file, double[] a)
```

save to a .wav file

```
    double[] read(String file)
```

read from a .wav file

library developed
for this course
(also broadly useful)



Musical Tone

Concert A. Play concert A for 1.5 seconds using stdAudio.

$$a(i) = \sin\left(\frac{2\pi \cdot i \cdot \text{hz}}{44,100}\right)$$

```
double hz = 440.0;
double seconds = 1.5;

int SAMPLE_RATE = 44100;
int N = (int) (seconds * SAMPLE_RATE);
double[] a = new double[N+1];
for (int i = 0; i <= N; i++) {
    a[i] = Math.sin(2 * Math.PI * i * hz / SAMPLE_RATE);
}
StdAudio.play(a);
```

Play That Tune

Play that tune. Read in pitches and durations from standard input;
sonify using standard audio.

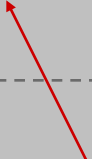
```
% more elise.txt  
7 .125  
6 .125  
7 .125  
6 .125  
7 .125  
2 .125  
5 .125  
3 .125  
0 .25
```

```
% java PlayThatTune < elise.txt
```



Play That Tune

```
public class PlayThatTune {  
    public static void main(String[] args) {  
        while (!StdIn.isEmpty()) {  
            int pitch = StdIn.readInt();  
            double seconds = StdIn.readDouble();  
            double hz = 440.0 * Math.pow(2, pitch / 12.0);  
  
            int SAMPLE_RATE = 44100;  
            int N = (int) (seconds * SAMPLE_RATE);  
            double[] a = new double[N+1];  
            for (int i = 0; i <= N; i++) {  
                a[i] = Math.sin(2 * Math.PI * i * hz / SAMPLE_RATE);  
            }  
            StdAudio.play(a);  
        }  
    }  
}
```



code as before

1.5 Extra Slides

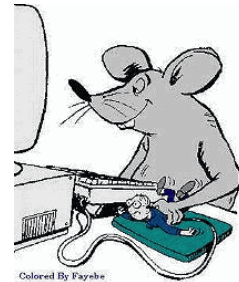
User Interfaces

Command line interface.

- User types commands at terminal.
- Easily customizable.
- Extends to complex command sequences.

Point and click.

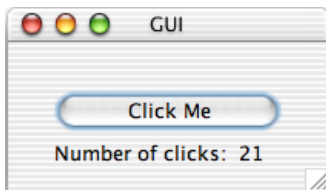
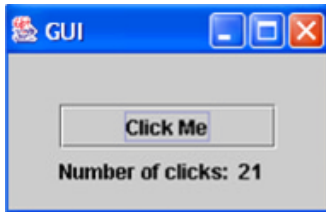
- User launches applications by clicking.
 - `File` → `Open` → `HelloWorld.java`
- Restricted to pre-packaged menu options.



Swing Graphical User Interface

"Swing" is Java's GUI.

- Buttons.
- Menus.
- Scrollbars.
- Toolbars.
- File choosers.



```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class GUI implements ActionListener {
    private int clicks = 0;
    private JFrame frame = new JFrame();
    private JLabel label = new JLabel("Number of clicks: 0");
    public GUI() {
        JButton button = new JButton("Click Me");
        button.addActionListener(this);
        JPanel panel = new JPanel();
        panel.setBorder(BorderFactory.createEmptyBorder(30, 30, 10, 30));
        panel.setLayout(new GridLayout(0, 1));
        panel.add(button);
        panel.add(label);
        frame.add(panel, BorderLayout.CENTER);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setTitle("GUI");
        frame.pack();
        frame.show();
    }

    public void actionPerformed(ActionEvent e) {
        clicks++;
        label.setText("Number of clicks: " + clicks);
    };

    public static void main(String[] args) {
        GUI gui = new GUI();
    }
}
```

a sample Swing application

Ignore details.

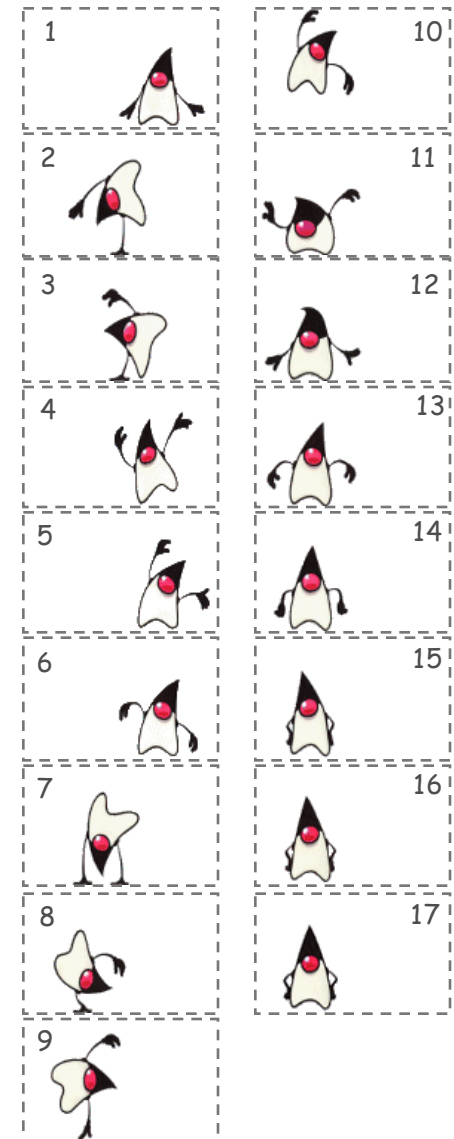
Computer Animation

Computer animation. Display a sequence of closely related images in rapid succession to produce the illusion of movement.

Frame rate. Use 15-70 frames per second to "trick" human eye and brain into seeing smooth motion.

Ex 1. Television and motion pictures.

Ex 2. Java mascot Duke cart-wheeling.



Java Implementation

```
public class Duke {  
    public static void main(String[] args) {  
        int images = 17;  
        int WIDTH = 130, HEIGHT = 80;  
        StdDraw.setCanvasSize(WIDTH, HEIGHT);  
  
        for (int t = 0; true; t++) {  
            int i = 1 + (t % images);  
            String file = "T" + i + ".gif";  
            StdDraw.picture(0.5, 0.5, file);  
            StdDraw.show(100);  
        }  
    }  
}
```

T1.gif - T17.gif



Operating System Specific Details

Common OS abstractions.

Operation	Windows XP	OS X	Unix
Cycle through recent command	Up, down arrows	Up, down arrows	Up, down arrows
File name completion	Tab	Tab	Tab
End of file	Ctrl-z	<Enter>Ctrl-d	Ctrl-d
Newline character	\r\n	\n or \r	\n
Scroll through text, one screenful at a time	more	more less	more less
List files in current directory	dir	ls	ls
Redirection, pipes	<, >,	<, >,	<, >,
File system	C:\introcS\Hi.java	/u/introcS/Hi.java	/u/introcS/Hi.java

Unix means Unix variants (Linux, Solaris, Aix)

Most Windows XP commands also supported in other version of Windows.

Twenty Questions

Twenty questions. User thinks of an integer between one and 1 million. Computer tries to guess it.

```
public class TwentyQuestions {  
    public static void main(String[] args) {  
        int lo = 1, hi = 1000000;  
        while (lo < hi) {  
            int mid = (lo + hi) / 2;  
            StdOut.println("Is your number <= " + mid + "?");  
            boolean response = StdIn.readBoolean();  
            if (response) hi = mid;  
            else          lo = mid + 1;  
        }  
        StdOut.println("Your number is " + lo);  
    }  
}
```

Binary search. Each question removes half of possible remaining values.

Consequence. Always succeeds after 20 questions.

$2^{20} \approx 1 \text{ million}$

invariant: user's number
always between lo and hi

Digital Michelangelo Project

Goal. Precise 3D description of the David.

- Laser rangefinder.
- 5,000 hours of scanning, 32 Gigabytes !

