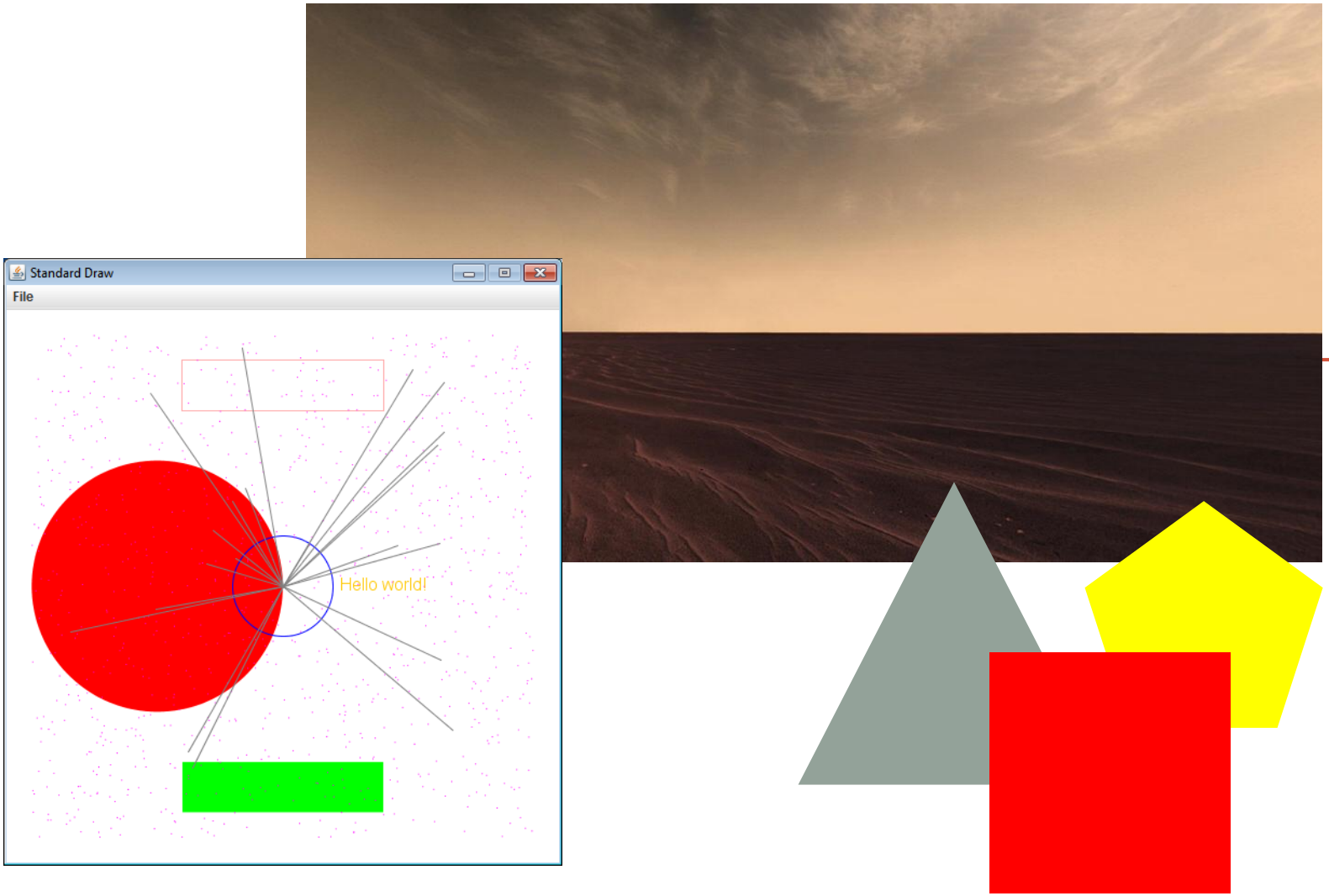
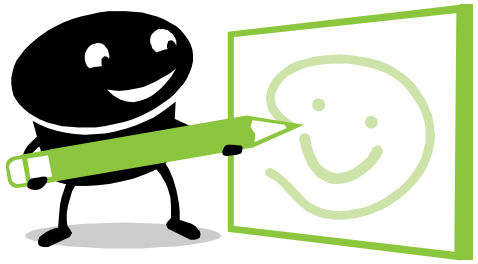
GRAPHICS AND SOUND



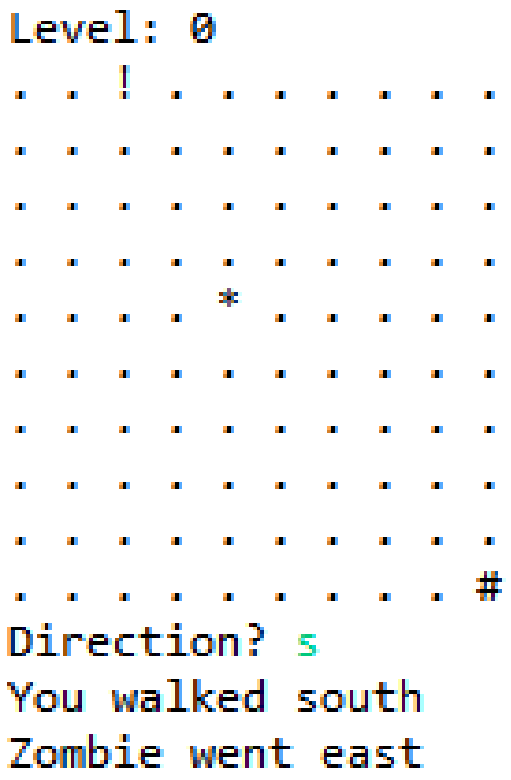
Outline

- File Input
- Graphics
  - `StdDraw.java`
    - Draw primitive shapes
    - Draw images from a file
    - Create animation loops
    - Get keyboard input from users
- Audio
  - StdAudio.java
    - Play audio files



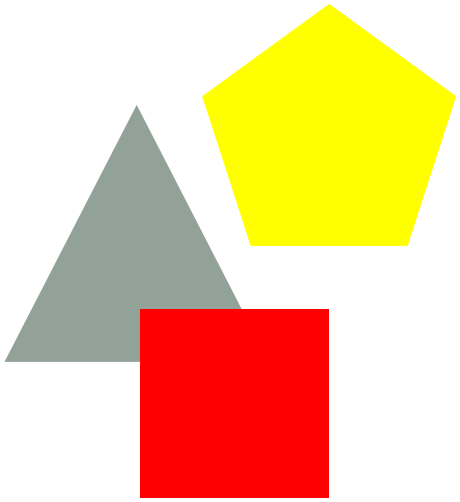
Input and Output Thus Far

- Input
  - Parsing command line arguments
  - Reading interactively from user
- Output
  - Display text to console



New Input/Output Capabilities

- `File Input`
- `StdDraw`
  - Draw shapes and images
  - Make animated programs
  - Get real-time keyboard input
- `StdAudio`
  - Playback of record sounds
  - Generate your own sounds



Input from Files

- What if..
  - There are too many values for a user to type interactively?
  - These values are stored in a text file?
- Can our program read these values from a file?
  - Yep! ☺

Java File Input

- We've used the Scanner class before

```
Scanner scanner = new Scanner(System.in);
```

  - The (System.in) part means get these values from standard input (the keyboard)
  - To get these values from a file, all we need to do is change the source:

```
Scanner scanner = new Scanner(new File("My File.txt"));
```

  - Or, we can have the user supply the file name as command line input:

```
Scanner scanner = new Scanner(new File(args[0]));
```

- Is it really that easy?
- Well… there's one hitch…

Java File Input

- Reading a file is a risky thing
  - Maybe the file doesn't exist
  - Since it's risky, Java requires you to handle that problem
  - If something goes wrong, the code around a file will throw an exception
  - Your code must deal with it by "trying" to open the file, and "catching" the exception if it didn't work
  - Use a try… catch construct

```
try
{
   // the risky thing
}
catch (FileNotFoundException e)
{
  //what to do if the risky thing fails
}
```

Java File Input

- You also need to import additional Java libraries
  - Just like you did to use the Scanner class
  - These go at the top of your program, before any Java code

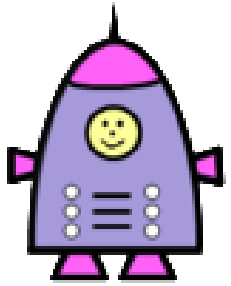
```
import java.io.File;
import java.io.FileNotFoundException;
```

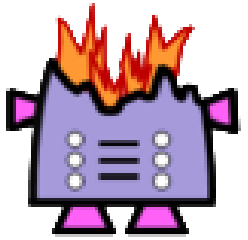
File Input

```
import java.io.File;
import java.io.FileNotFoundException;
import java.util.Scanner;

public class AvgNumsFile
{
   public static void main(String [] args)
   {
      double sum = 0.0;
      long count = 0;

        try
        {
            Scanner scanner = new Scanner(new File(args[0]));
            while (scanner.hasNext())
            {
               sum += scanner.nextDouble();
               count++;
            }
            scanner.close();
            System.out.println(sum / count);
         }
         catch (FileNotFoundException e)
         {
               System.out.println("Failed to open file!");
         }
   }
}
```

StdDraw Overview

- StdDraw
  - Like Math and System, we'll use another class: StdDraw
  - Put StdDraw.java in directory with your program
  - Draw simple things:
    - Rectangles, circles, lines, polygons, text
    - Make them different colors
  - Draw images loaded from a file:
    - e.g. spaceship, Mars background, etc.
  - Animate things:
    - e.g. bouncing ball, video games





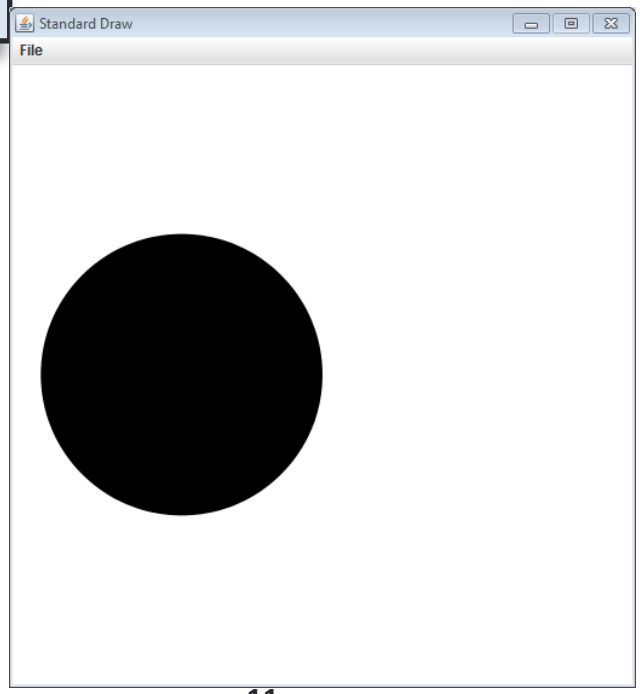
Hello Drawing!

```
public class HelloDraw
{
   public static void main(String [] args)
   {
      StdDraw.filledCircle(0.25, 0.5, 0.25);
   }
}
```

Put the circle at x coordinate 0.25

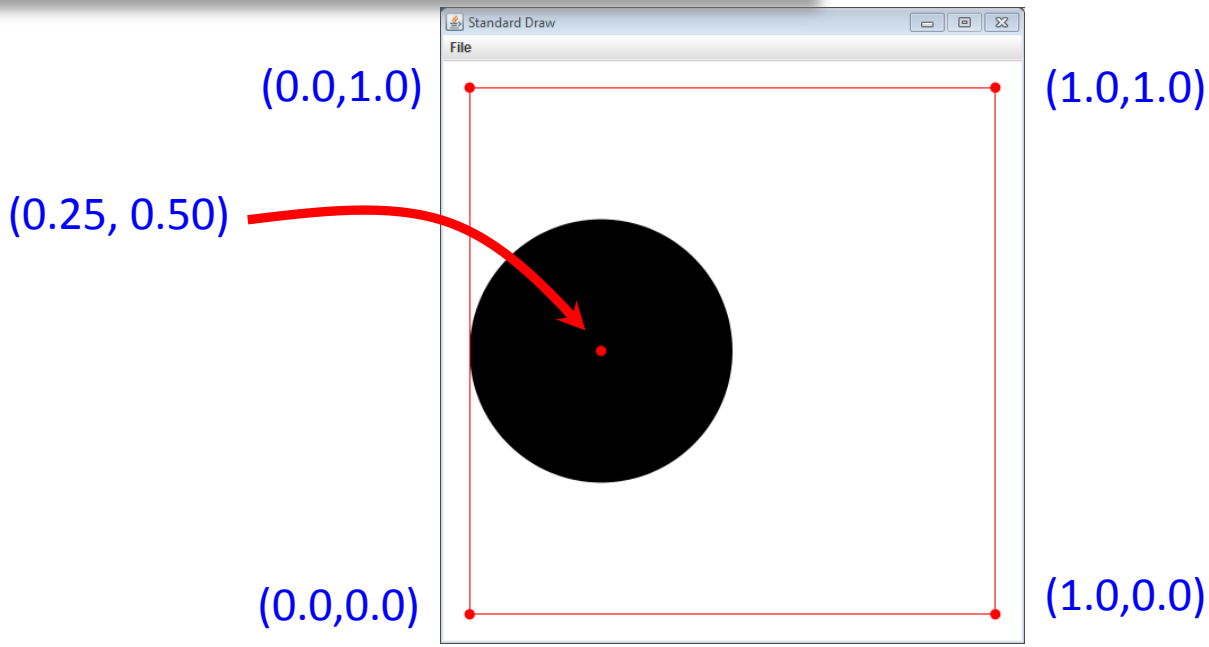
Put the circle at y coordinate 0.5

Make the circle be of radius 0.25



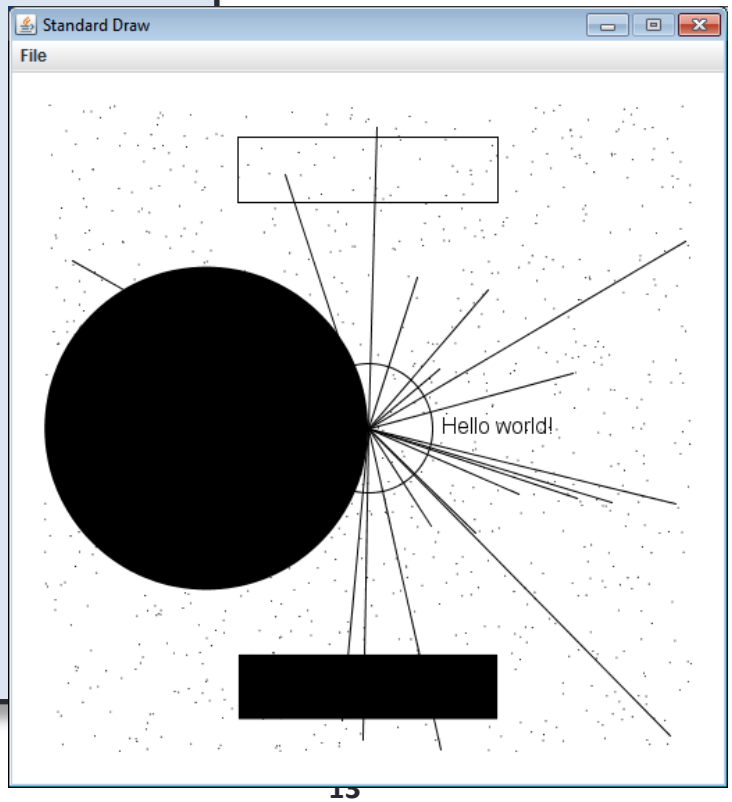
Default Coordinate System

```
public class HelloDraw
{
   public static void main(String [] args)
   {
      StdDraw.filledCircle(0.25, 0.5, 0.25);
   }
}
```



Other Shapes and Text

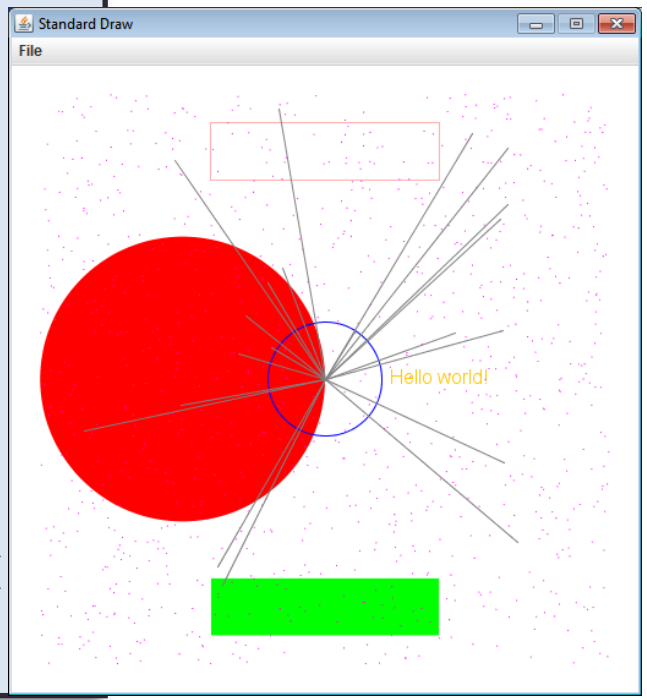
```
public class DrawShapes
{
   public static void main(String [] args)
   {
      StdDraw.filledCircle(0.25, 0.5, 0.25);
      StdDraw.circle(0.5, 0.5, 0.1);

      StdDraw.filledRectangle(0.5, 0.1, 0.2, 0.05);
      StdDraw.rectangle(0.5, 0.9, 0.2, 0.05);

      StdDraw.text(0.7, 0.5, "Hello world!");

      for (int i = 0; i < 1000; i++)
         StdDraw.point(Math.random(),
                       Math.random());

      for (int i = 0; i < 20; i++)
         StdDraw.line(0.5,
                      0.5,
                      Math.random(),
                      Math.random());
   }
}
```



Adding Color

```
public class DrawShapesColor
{
   public static void main(String [] args)
   {
      StdDraw.setPenColor(StdDraw.RED);
      StdDraw.filledCircle(0.25, 0.5, 0.25);
      StdDraw.setPenColor(StdDraw.BLUE);
      StdDraw.circle(0.5, 0.5, 0.1);

      StdDraw.setPenColor(StdDraw.GREEN);
      StdDraw.filledRectangle(0.5, 0.1, 0.2, 0.05);
      StdDraw.setPenColor(StdDraw.PINK);
      StdDraw.rectangle(0.5, 0.9, 0.2, 0.05);

      StdDraw.setPenColor(StdDraw.ORANGE);
      StdDraw.text(0.7, 0.5, "Hello world!");

      StdDraw.setPenColor(StdDraw.MAGENTA);
      for (int i = 0; i < 1000; i++)
         StdDraw.point(Math.random(), Math.random());

      StdDraw.setPenColor(StdDraw.GRAY);
      for (int i = 0; i < 20; i++)
         StdDraw.line(0.5, 0.5, Math.random(), Math.random(
   }
}
```

```
StdDraw.BLACK
StdDraw.BLUE
StdDraw.CYAN
StdDraw.DARK_GRAY
StdDraw.GRAY
StdDraw.GREEN
StdDraw.LIGHT_GRAY
StdDraw.MEGENTA
StdDraw.ORANGE
StdDraw.PINK
StdDraw.RED
StdDraw.WHITE
StdDraw.YELLOW
```



Changing Coordinate Size

- Often convenient to use different coordinates
  - 0.0 to 1.0 is default x-size and y-size
  - Change x-size `StdDraw.setXscale(double min, double max)`
  - Change y-size `StdDraw.setYscale(double min, double max)`

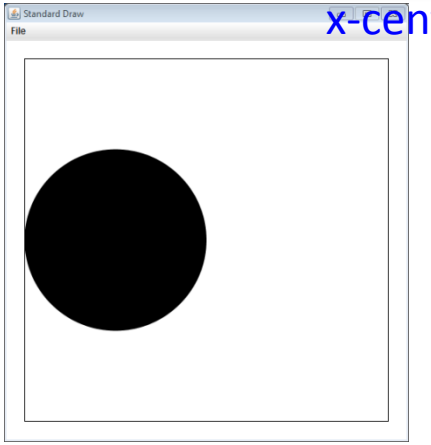
```
StdDraw.filledCircle(0.25, 0.5, 0.25);
StdDraw.rectangle(0.5, 0.5, 0.5, 0.5);
```

x-center of rectangle

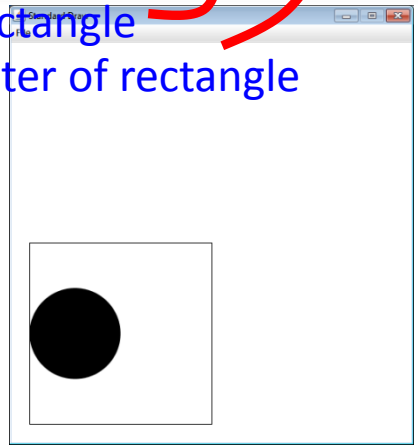
y-center of rectangle

half width

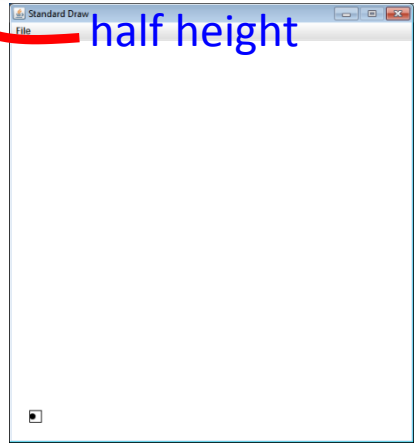
half height



```
StdDraw.setXScale(0.0,1.0);
StdDraw.setYScale(0.0,1.0);
```



```
StdDraw.setXScale(0.0,2.0);
StdDraw.setYScale(0.0,2.0);
```



```
StdDraw.setXScale(0.0,30.0);
StdDraw.setYScale(0.0,30.0);
```

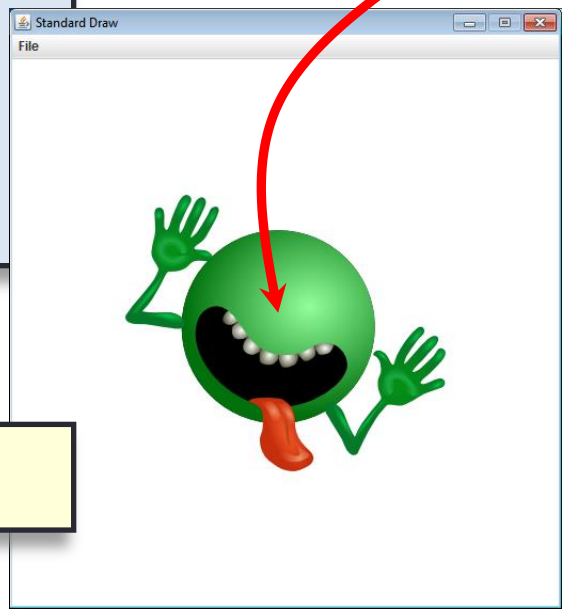
Drawing Images

- Loading image from file
  - Supports various formats such as JPG and PNG
  - Put image files in same directory with program
  - `StdDraw.picture(double x, double y, String filename)`

x-center

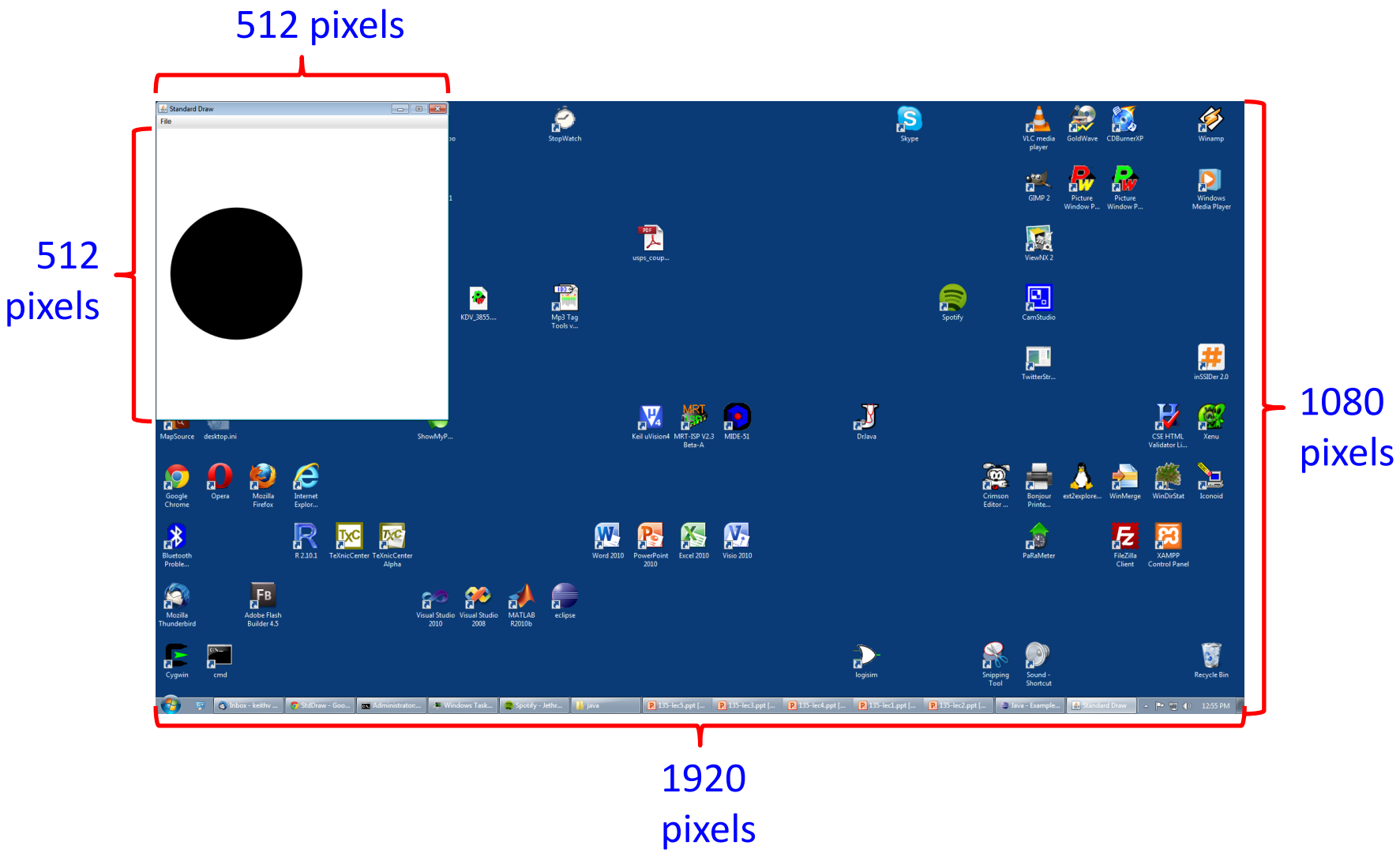
y-center

```
public class DrawImage
{
   public static void main(String [] args)
   {
      StdDraw.picture(0.5, 0.5, args[0]);
   }
}
```



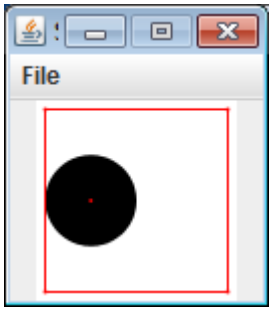

```
% java DrawImage dont_panic.png
```

Window Size

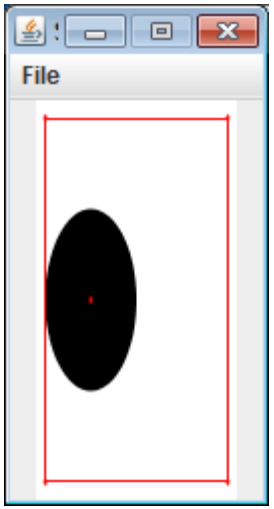

512 pixels
512 pixels
1080 pixels
1920 pixels
Standard Draw
File

Changing Window Size

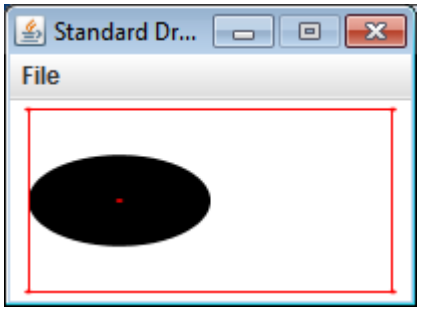
- Window size
  - Default size: 512 x 512 pixels
  - Set different size:
    - `StdDraw.setCanvasSize(int width, int height)`
  - Call just once at start of program



100 x 100



100 x 200



200 x 100

Animating Things

- Animation loop
  - Clear previous drawing
    - `StdDraw.clear()` (or draw a picture over the screen)
  - Draw new stuff
  - Sleep for awhile
    - `StdDraw.show(int timeMs)`
  - Repeat

```
public class SpinningImage
{
   public static void main(String [] args)
   {
      int degrees = 0;
      while (true)
      {
         StdDraw.clear();
         StdDraw.picture(0.5, 0.5, args[0], degrees);
         degrees = (degrees + 1) % 360;
         StdDraw.show(10);
      }
   }
}
```

Keyboard Input

- Responding to keyboard input
  - Problem: Interactive input waits for text then enter key
  - `StdDraw` gives us real-time keyboard input
    - Check if key was pressed: `StdDraw.hasNextKeyTyped()`
    - Find out the key: `StdDraw.nextKeyTyped()`
  - Note: must click on drawing window first
  - Example:
    - Make image spin clockwise on 'a'
    - Make image spin counterclockwise on 's'
    - Stop spinning on any other key

Interactive Spinning Image

```
public class SpinningImageKey
{
   public static void main(String [] args)
   {
      int degrees = 0;
      int direction = 0;
      char ch = '\0';
      while (ch != 'q')
      {
         StdDraw.clear();
         StdDraw.picture(0.5, 0.5, args[0], degrees);

         if (StdDraw.hasNextKeyTyped())
         {
            ch = StdDraw.nextKeyTyped();
            if (ch == 'a')
               direction = 1;
            else if (ch == 's')
               direction = -1;
            else
               direction = 0;
         }
         degrees = (degrees + direction) % 360;
         StdDraw.show(10);
      }
   }
}
```

Adding Sound

- `StdAudio`
  - Plays sound files in .wav, .au, .mid format
    - Plays one time
    - `StdAudio.play(String filename)`
  - Also can play raw audio in double []
    - For creating your own sounds
  - Example, add audio to our spinning image:

```
public class SpinningImageKeyAudio
{
   public static void main(String [] args)
   {
      StdAudio.play(args[1])
      ...
```

Additional information

- Many more methods in `StdDraw` and `StdAudio`
  - Full documentation:
  - http://introcs.cs.princeton.edu/java/stdlib/javadoc/StdDraw.html
  - http://introcs.cs.princeton.edu/java/stdlib/javadoc/StdAudio.html

```
void line(double x0, double y0, double x1, double y1)
void point(double x, double y)
void circle(double x, double y, double r)
void filledCircle(double x, double y, double r)
void square(double x, double y, double r)
void filledSquare(double x, double y, double r)
void polygon(double [] x, double [] y)
void filledPolygon(double [] x, double [] y)
void text(double x, double y, String s)
void setFont(Font f)
void setPenColor(Color c)
...
```

Summary

- File Input
- Graphics
  - `StdDraw.java`
    - Draw primitive shapes
    - Draw images from a file
    - Create animation loops
    - Get keyboard input from users
- Audio
  - StdAudio.java
    - Play audio files