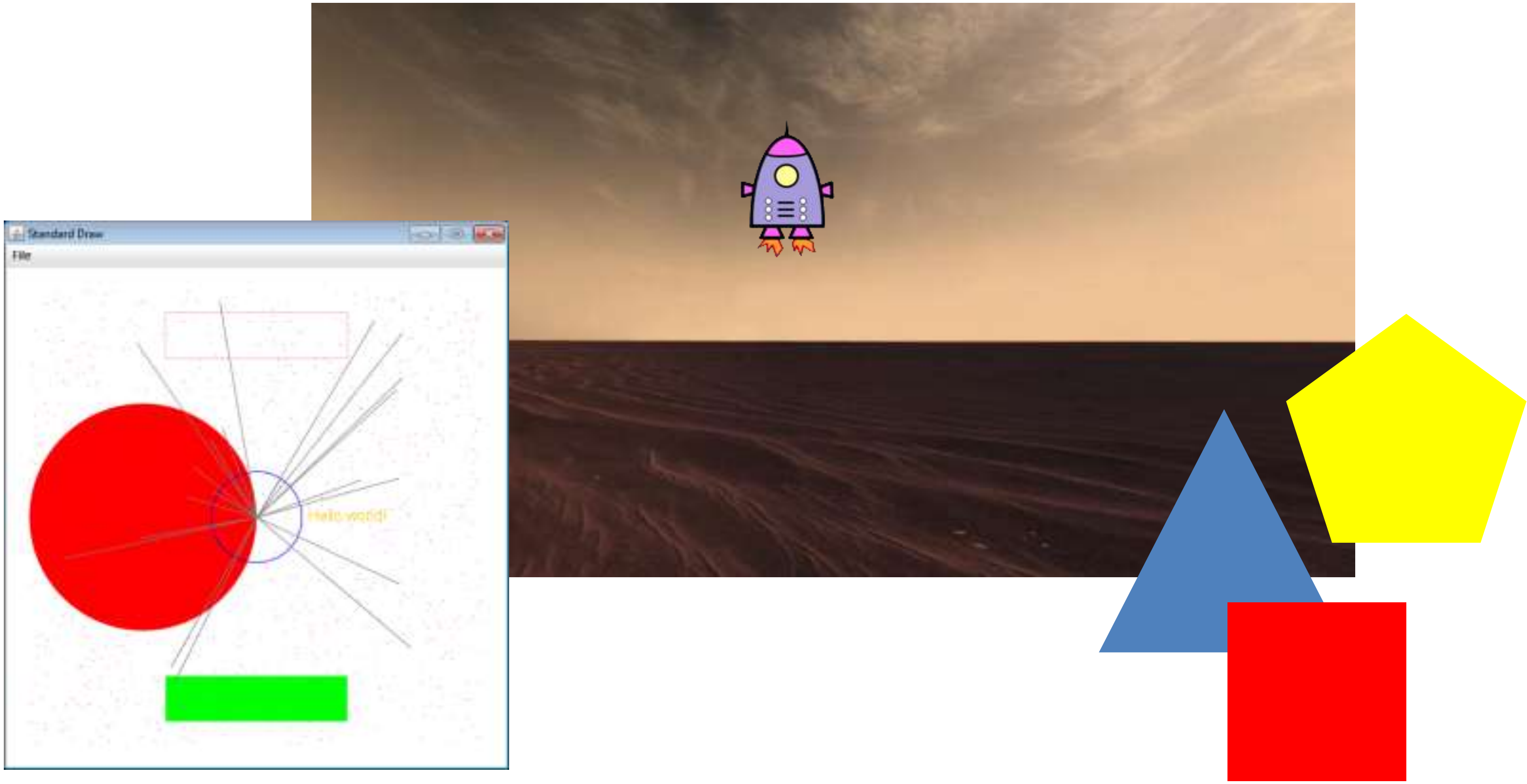
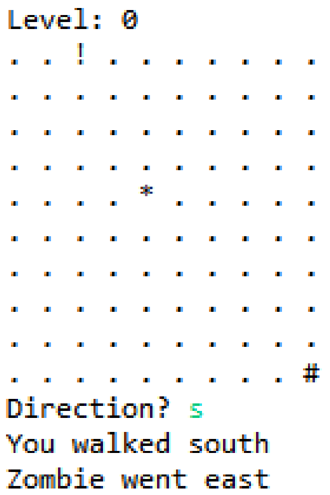
Graphics and sound



Input and output thus far

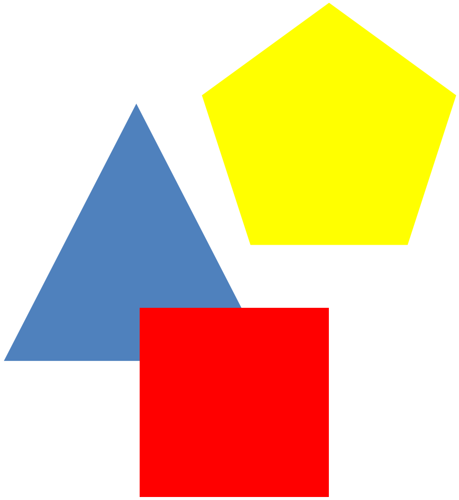
- Input
  - Parsing command line arguments
  - Reading from a file using standard input
  - Reading text typed by a user
- Output
  - Display text to console



```
Level: 0
. . ! . . . . . . .
. . . . . . . . . .
. . . . . . . . . .
. . . . . . . . . .
. . . . * . . . . .
. . . . . . . . . .
. . . . . . . . . .
. . . . . . . . . .
. . . . . . . . . .
. . . . . . . . . #
Direction? s
You walked south
Zombie went east
```

New input/output capabilities

- StdDraw
  - – Draw shapes and images
  - – Make animated programs
  - – Get real-time keyboard input



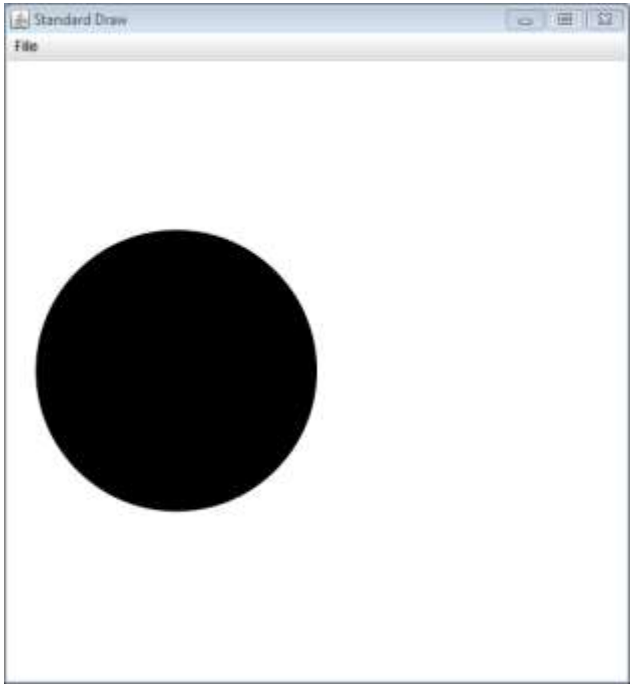
- StdAudio
  - – Playback of record sounds
  - – Generate your own sounds

StdDraw overview

- StdDraw
  - Like `StdIn`, 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);
   }
}
```



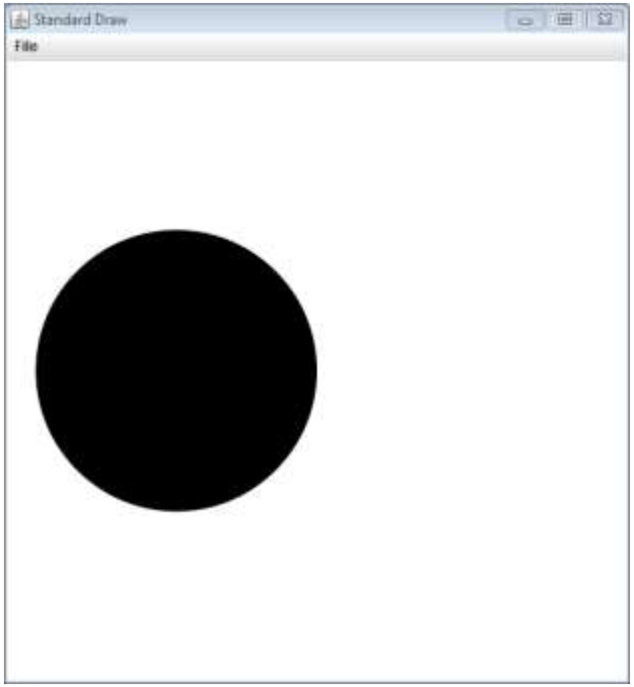
Arguments to `filledCircle()`

```
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

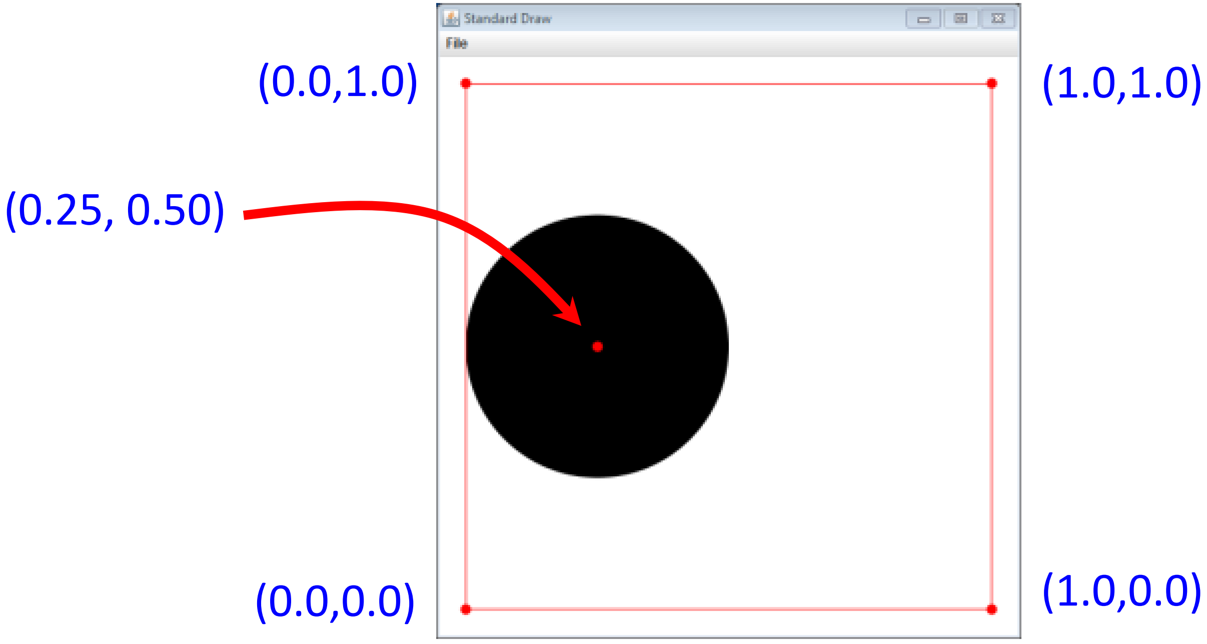
Put the circle at y coordinate 0.5

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);
    }
}
```

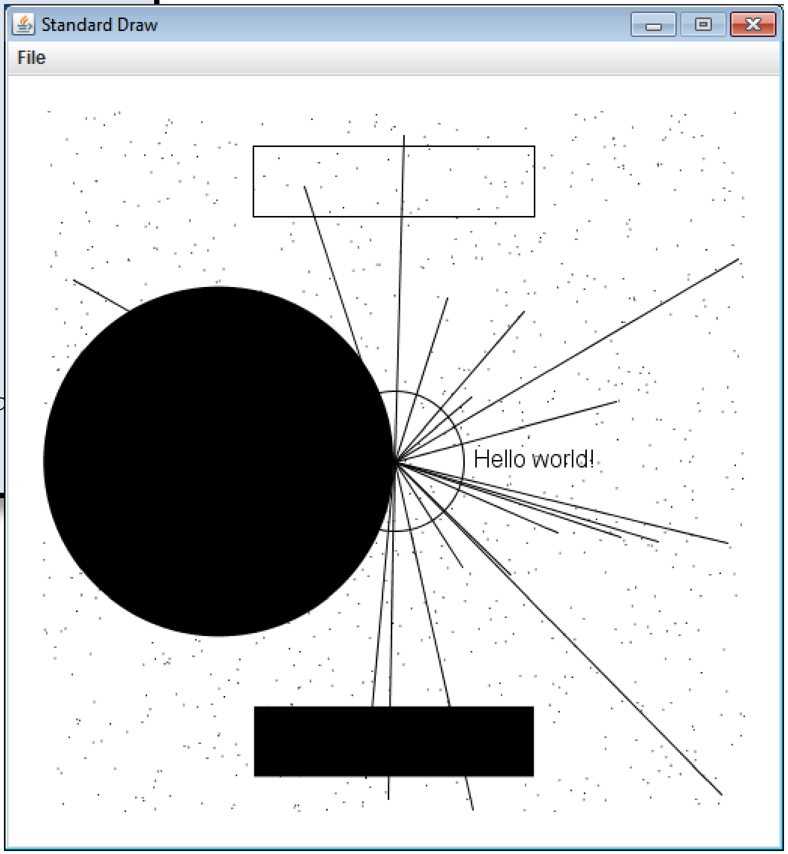


Other shapes and text

```
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());
  }
}
```

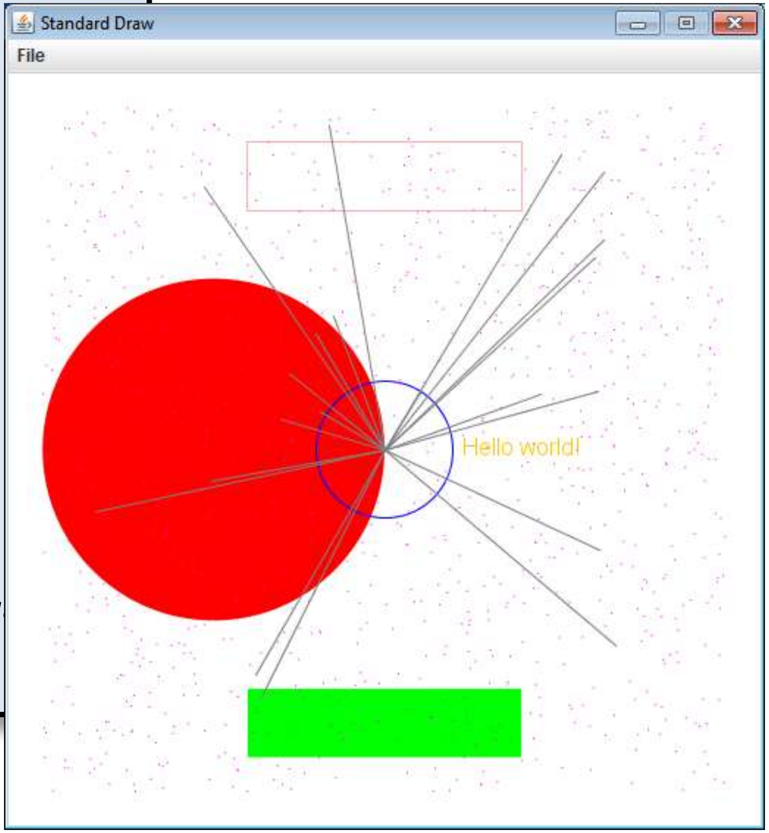
Other shapes and text

```
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.rand
   }
}
```



Adding color

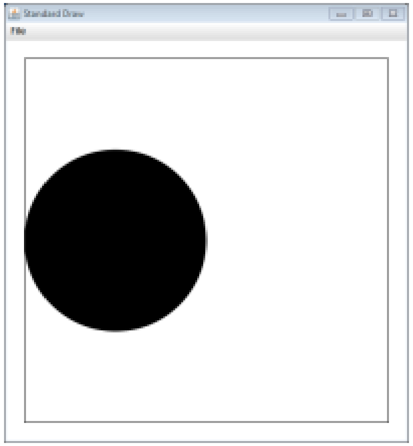
```
public class DrawShapesColor
{
   public static void main(String [] args)
   {
      StdDraw.setPenColor(StdDraw.RED);
      StdDraw.filledCircle(0.25, 0.5, 0.25);
      StdDraw.setPenColor(StdDraw.BLUE);
      StdDraw.circle(0.5, 0.5, 0.1);

      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.rand
   }
}
```



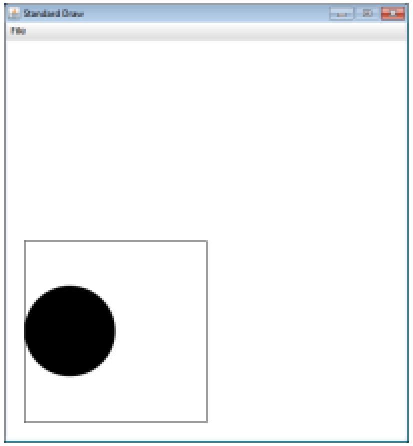
Changing coordinate size

- 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)`

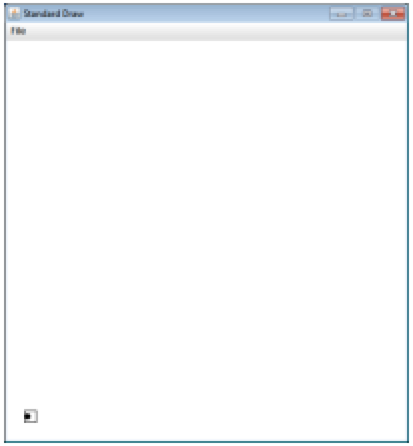
```
StdDraw.filledCircle(0.25, 0.5, 0.25);
StdDraw.rectangle(0.5, 0.5, 0.5, 0.5);
```



```
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)`

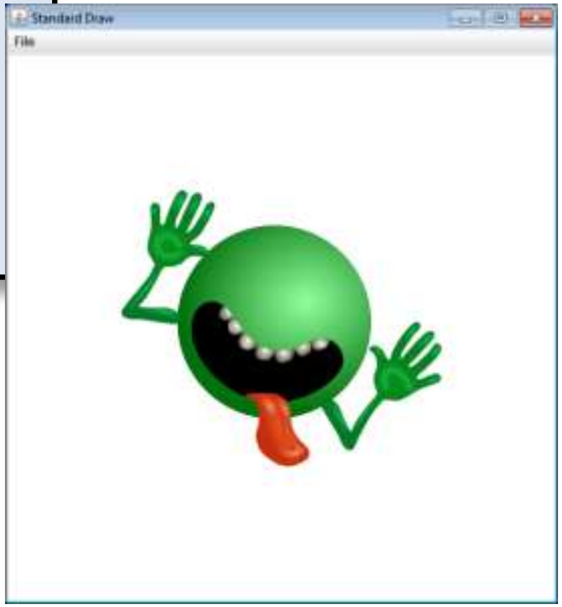
```
public class DrawImage
{
   public static void main(String [] args)
   {
      StdDraw.picture(0.5, 0.5, args[0]);
   }
}
```

Drawing images

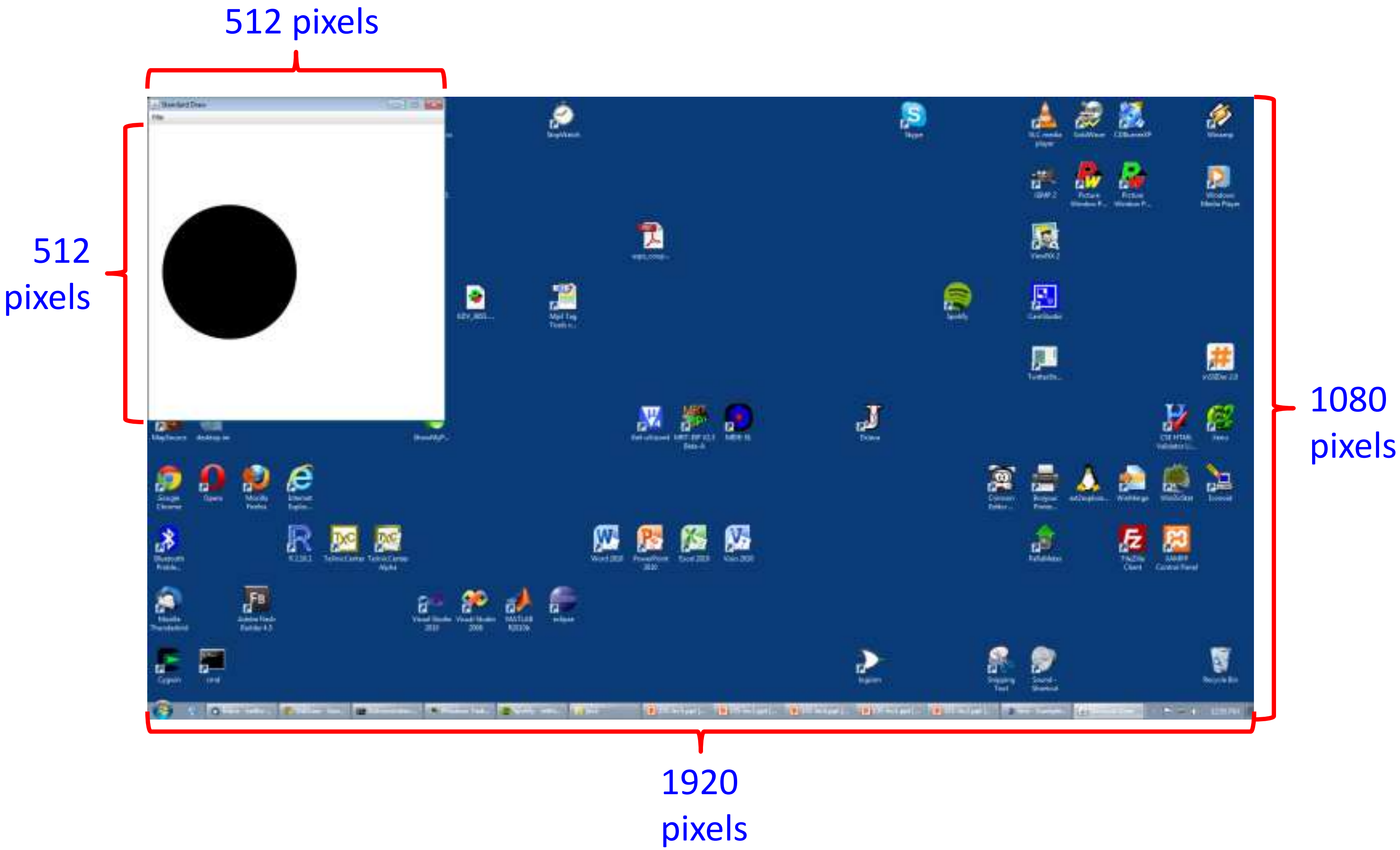
- Loading image from file
  - Supports various formats such as JPG and PNG
  - Put image files in same directory with program
  - `StdDraw.picture(double x, double y, String filename)`

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

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

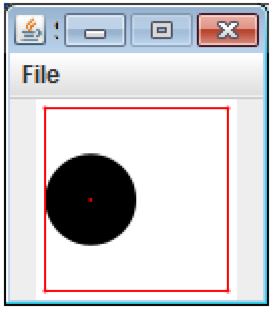


Window size

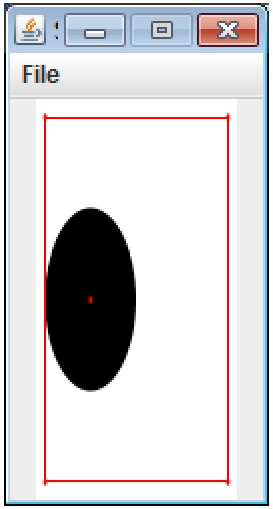

512 pixels
512 pixels
1080 pixels
1920 pixels
Standard Draw
File

Changing window size

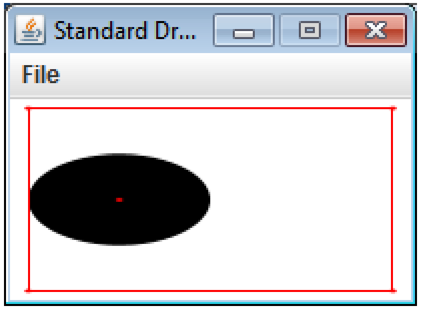
- Window size
  - Defaults size 512 x 512 pixels
  - To set different size:
    - `StdDraw.setCanvasSize(int width, int height)`
  - Call just once at start of program



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 using `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: `StdIn` 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;
      while (true)
      {
         StdDraw.clear();
         StdDraw.picture(0.5, 0.5, args[0], degrees);

         if (StdDraw.hasNextKeyTyped())
         {
            char 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 program:

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

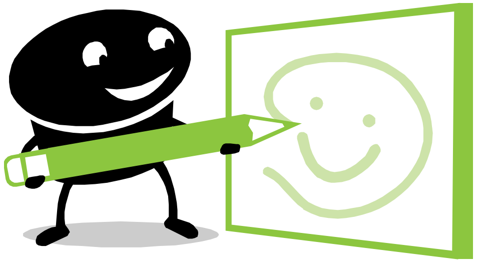
Additional information

- 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

- Drawing
  - Easy to do with `StdDraw.java`
  - Draw primitive shapes
  - Draw images from a file
  - Create animation loops
  - Get keyboard input from users



- Audio
  - Play audio files