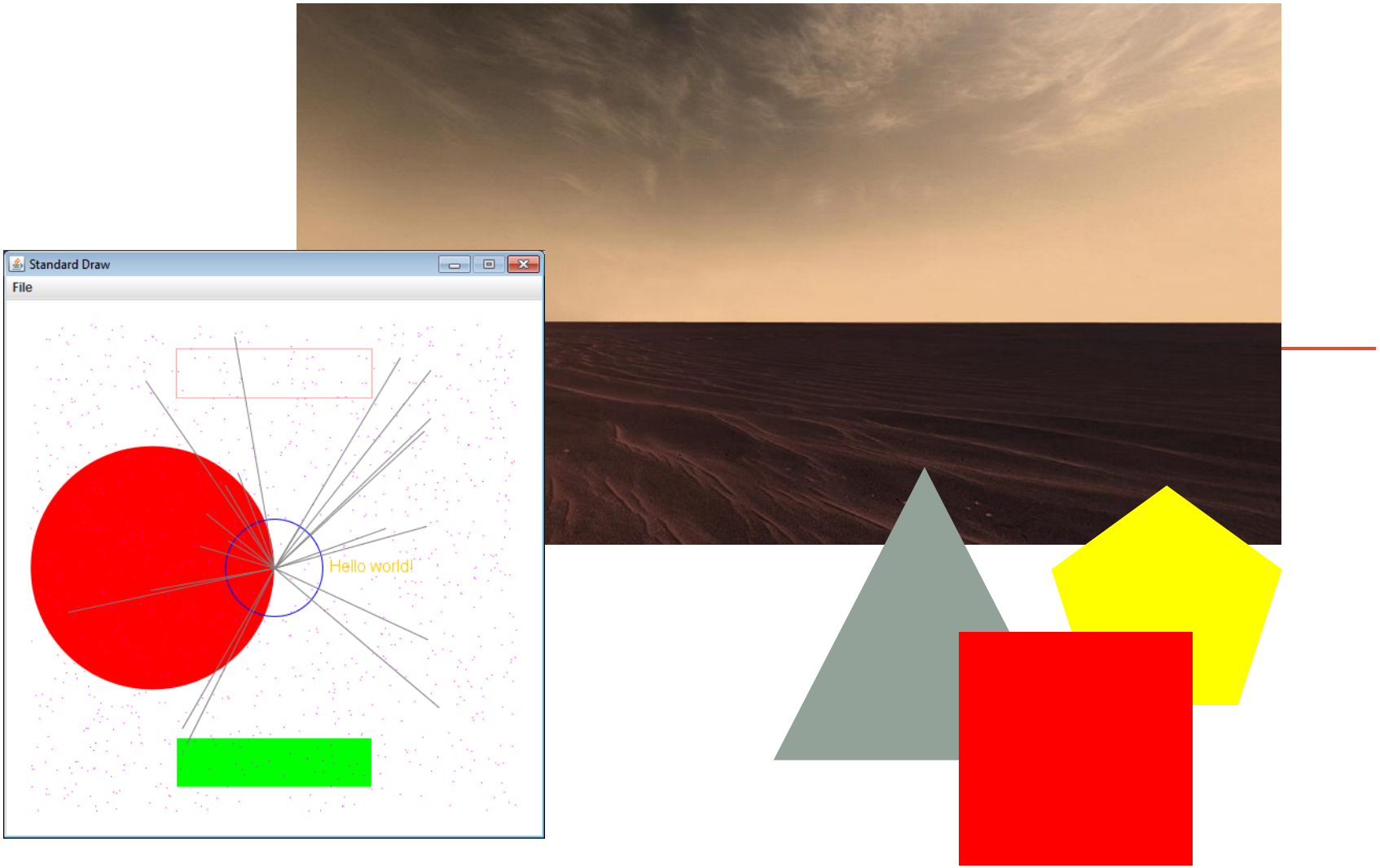
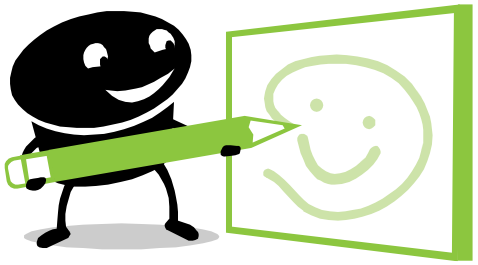
GRAPHICS PRACTICE



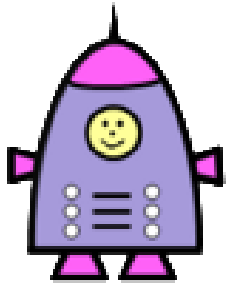
Outline

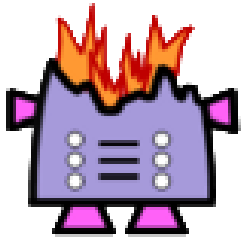
- Graphics
  - Practice
    - Shapes
    - Color
    - Window Coordinates
    - Background Images
    - Animation
- Sound



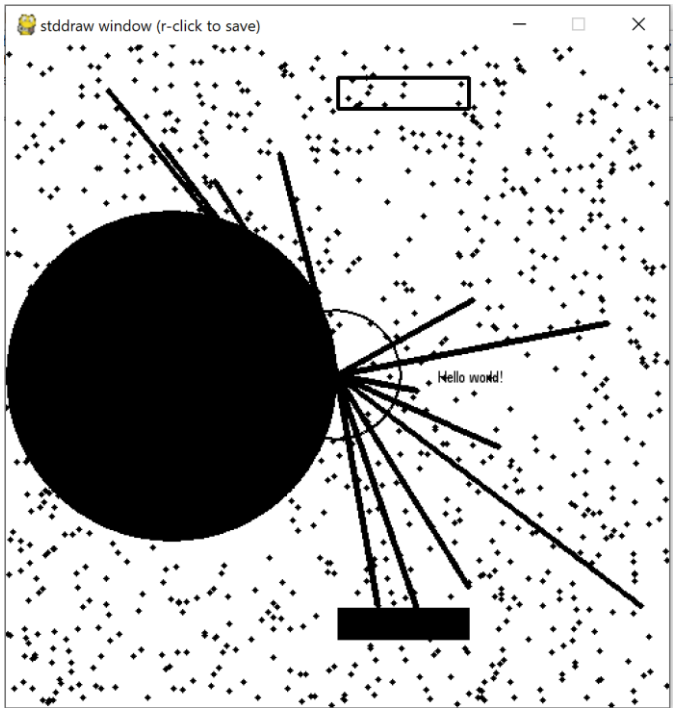
StdDraw Overview

- StdDraw
  - Like random and sys, we'll use another library: StdDraw
  - Put StdDraw.py in directory/folder with your program
    - You will also need:
      - stdarray.py
      - stdio.py
      - picture.py
      - color.py
    - These are all downloadable from the class website





Shapes



- Circles – x, y is center

```
StdDraw.circle(x, y, r)
StdDraw.filledCircle(x, y, r)
```

- Rectangles – x, y is lower left corner

```
StdDraw.rectangle(x, y, w, h)
StdDraw.filledRectangle(x, y, w, h)
```

- Points

```
StdDraw.point(x, y)
```

- Lines

```
StdDraw.line(x1, y1, x2, y2)
```

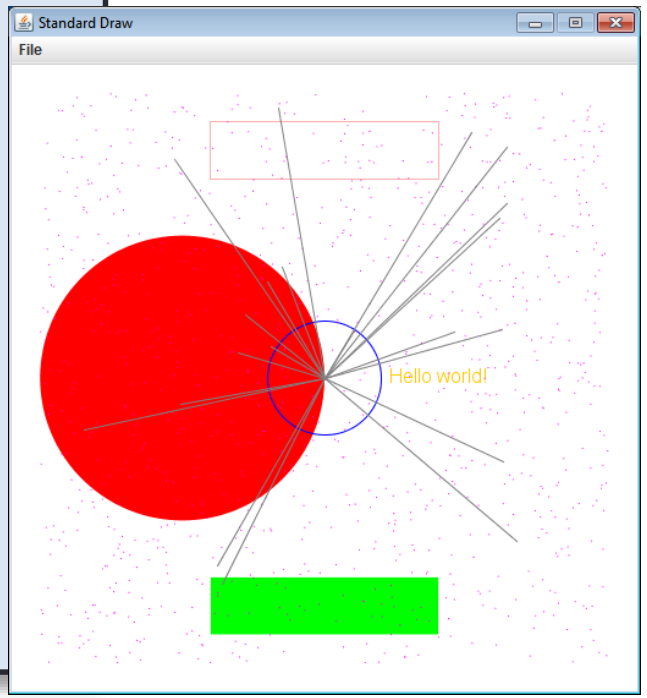
- Polygons – first list is x coordinates, second is corresponding y

```
StdDraw.polygon([x1, x2, …, xn], [y1, y2, …, yn])
StdDraw.filledPolygon([x1, x2, …, xn], [y1, y2, …, yn])
```

Color

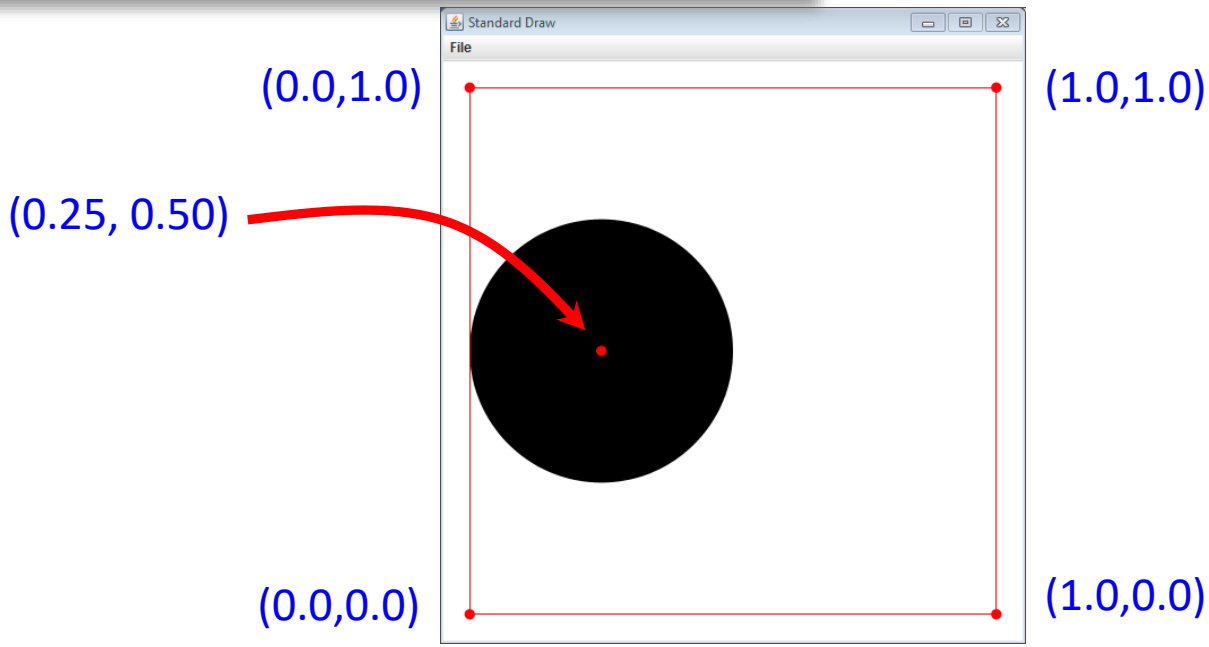
```
import StdDraw
import random

# Drawing circles involves sending in an (x,y) center plus a radius
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.show(0.0)

# Rectangles are drawn at an (x,y) center with the passed in half
# width and half height argument (the distance to the edge from the
# (x,y) center).
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.show(0.0)

# Text is drawn centered at the given (x,y)
StdDraw.setPenColor(StdDraw.ORANGE)
StdDraw.text(0.7, 0.5, "Hello world!")
StdDraw.show(0.0)

# Scatter a 1000 random points around the screen
StdDraw.setPenColor(StdDraw.MAGENTA)
for i in range(0, 1000):
    StdDraw.point(random.random(), random.random())
StdDraw.show(0.0)

# 20 random lines radiating from the center
StdDraw.setPenColor(StdDraw.GRAY)
for i in range(0, 20):
    StdDraw.line(0.5, 0.5, random.random(), random.random())
StdDraw.show(0.0)
```

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

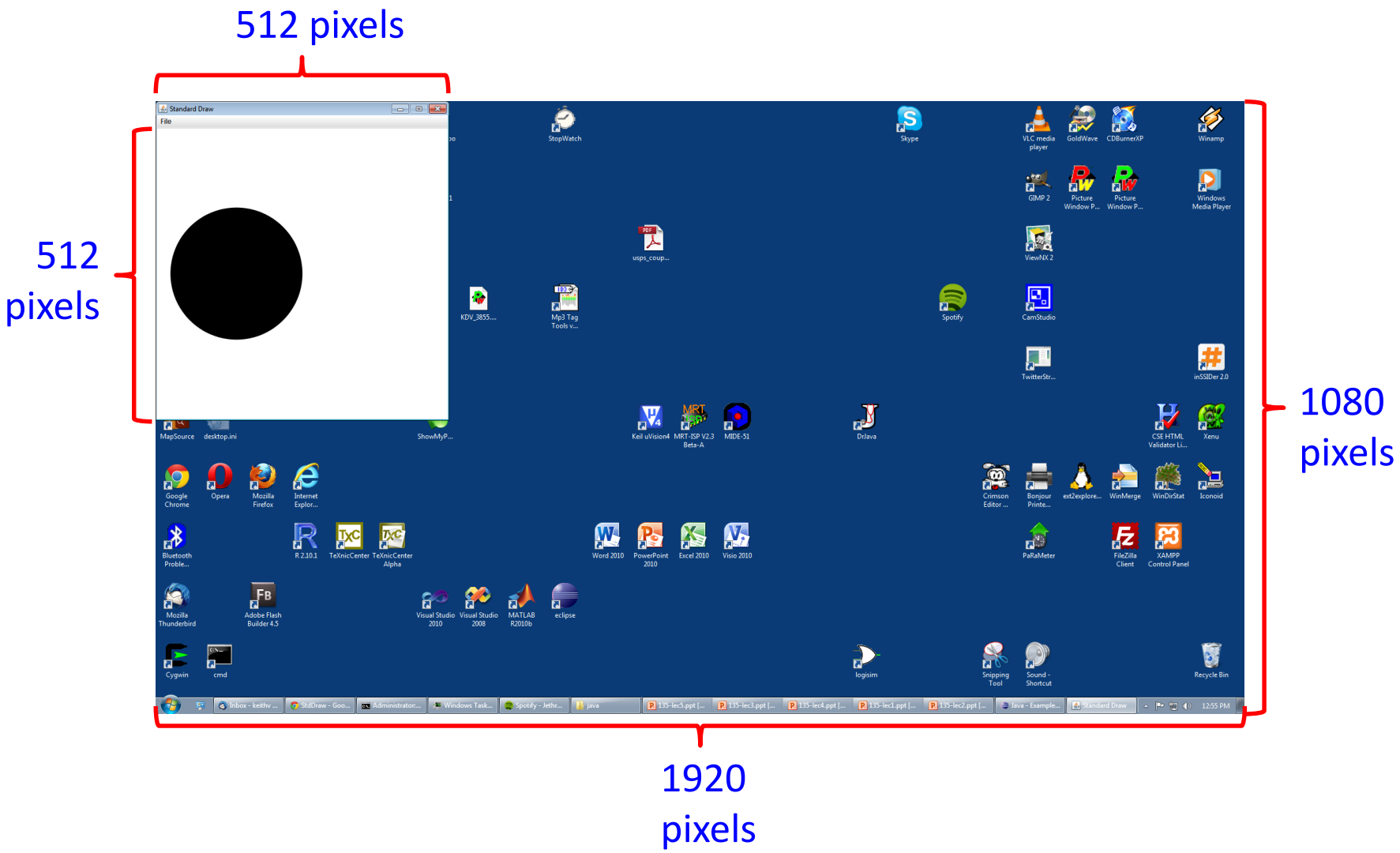


Default Coordinate System

```
import StdDraw

StdDraw.filledCircle(0.25, 0.5, 0.25)
StdDraw.show(0.0)
```

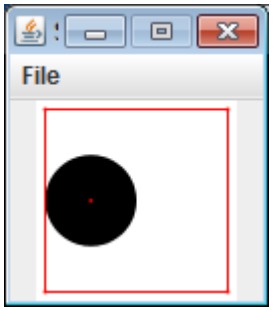


Window Size

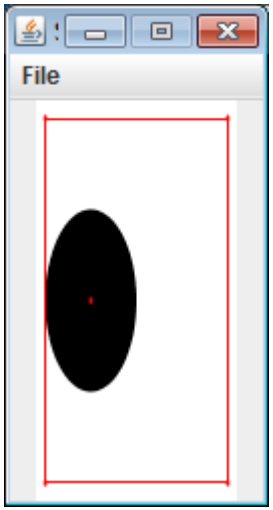

512 pixels
512 pixels
1080 pixels
1920 pixels
Standard Draw
File

Changing Window Size

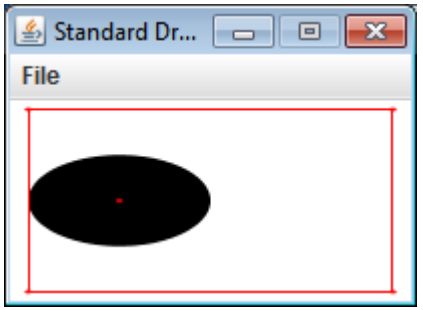
- Window size
  - Default size: 512 x 512 pixels
  - Set different size:
    - `StdDraw.setCanvasSize(width, height)`
  - Call just once at start of program



100 x 100



100 x 200



200 x 100

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(min, max)`
  - Change y-size `StdDraw.setYscale(min, max)`

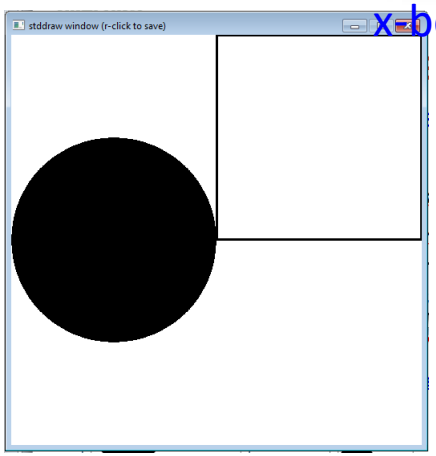
```
StdDraw.filledCircle(0.25, 0.5, 0.25)
StdDraw.rectangle(0.5, 0.5, 0.5, 0.5)
```

x-bottom of rectangle

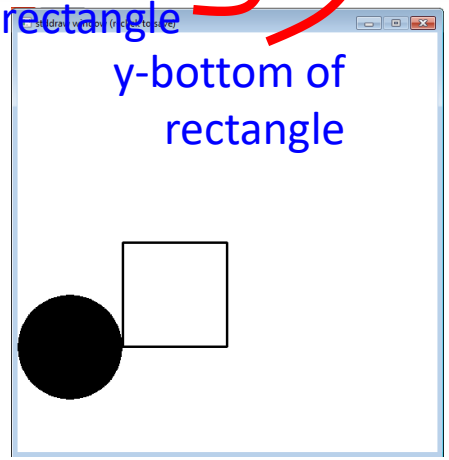
y-bottom of rectangle

width

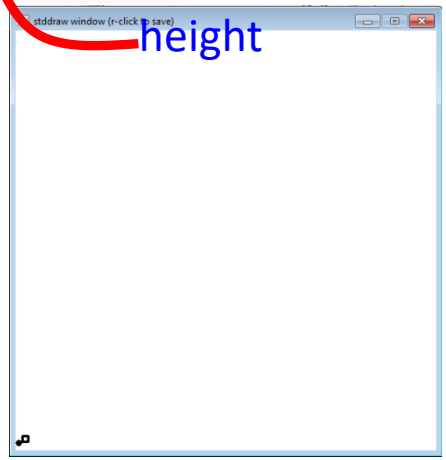
height



`StdDraw.setXscale(0.0,1.0)`
`StdDraw.setYscale(0.0,1.0)`



`StdDraw.setXscale(0.0,2.0)`
`StdDraw.setYscale(0.0,2.0)`



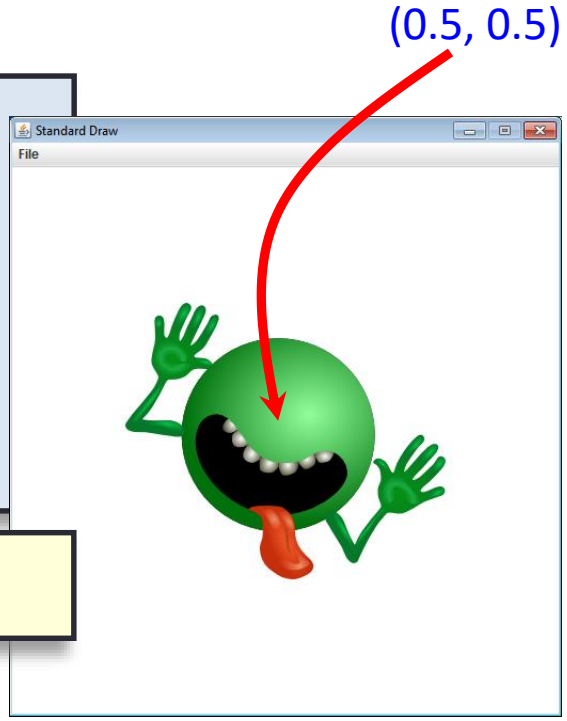
`StdDraw.setXscale(0.0,30.0)`
`StdDraw.setYscale(0.0,30.0)`

Drawing Images

- Loading image from file
  - Supports various formats such as JPG and PNG
  - Put image files in same directory with program
  - `StdDraw.picture(picture, x, y)`
- `Must be run from the command window, not the Idle shell!`

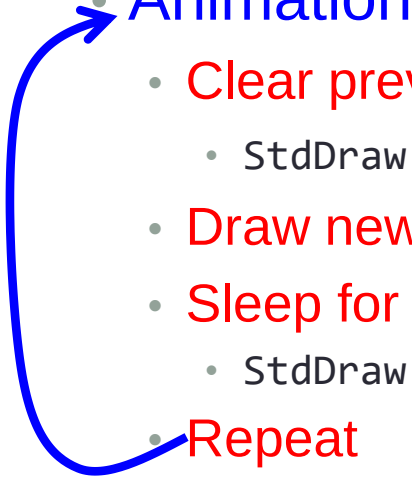
```
import StdDraw
import sys
import picture as p

pic = p.Picture(sys.argv[1])
StdDraw.picture(pic, 0.5, 0.5)
while True:
    StdDraw.show(0.0)
```

```
% python DrawImage.py dont_panic.png
```



Animating Things

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



```
import StdDraw
import sys
import picture as p

x, y = 0.5, 0.5
xOffset, yOffset = 0.01, 0.01
pic = p.Picture(sys.argv[1])
while True:
    StdDraw.clear()
    StdDraw.picture(pic, x, y)
    x += xOffset
    y += yOffset
    if x > 1.0 or x < 0.0:
        xOffset *= -1
    if y > 1.0 or y < 0.0:
        yOffset *= -1
    StdDraw.show(50)
```

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 change x direction on ‘x'
    - Make image change y direction on ‘y'
    - Go back to default on any other key

Interactive Bouncing Image

```
import StdDraw
import sys
import picture as p

x, y = 0.5, 0.5
xDirection, yDirection = 0.0, 0.0
xOffset, yOffset = 0.01, 0.01
pic = p.Picture(sys.argv[1])
while True:
    StdDraw.clear()
    StdDraw.picture(pic, x, y)
    if StdDraw.hasNextKeyTyped():
        ch = StdDraw.nextKeyTyped()
        if ch == 'a':
            xOffset += 0.05
        elif ch == 's':
            yOffset += 0.05
    x += xOffset
    y += yOffset
    if x > 1.0 or x < 0.0:
        xOffset *= -1
    if y > 1.0 or y < 0.0:
        yOffset *= -1
    StdDraw.show(50)
```

Mouse Input

- Responding to mouse input:
  - StdDraw.mousePressed()
  - x = StdDraw.mouseX()
  - y = StdDraw.mouseY()

```
while True:
    if StdDraw.mousePressed():
        StdDraw.filledCircle(StdDraw.mouseX(),StdDraw.mouseY(), .02)
StdDraw.show(0.0)
```

Adding Sound

- `StdAudio`
  - Plays sound files in .wav format
    - Plays one time
    - `StdAudio.playFile(filename)`
  - Make sure the filename you use does not include the .wav extension
    - Say you have a file frog.wav
    - `StdAudio.playFile("frog")`
  - Example, add audio to our frog image:

```
import StdAudio
…
 else:
            xOffset = 0.01
            yOffset = 0.01
            StdAudio.playFile(sys.argv[2])
```

Additional information

- Many more functions in `StdDraw` and `StdAudio`
  - Can look at the individual programs to see the functions you can call:

```
line(x0, y0, x1, y1)
point(x, y)
circle(x, y, radius)
filledCircle(x, y, radius)
rectangle(left_x, bottom_y, width, height)
filledRectangle(left_x, bottom_y, width, height)
square(left_x, bottom_y, length)
filledSquare(left_x, bottom_y, length)
Polygon(listOfXs, listOfYs)
text(x, y, string)
setFontFamily(fontFamily)
setPenRadius(radius)
setPenColor(color)
...
```

Summary

- Graphics
  - Practice
    - Shapes
    - Color
    - Window Coordinates
    - Background Images
    - Animation
- Sound

Your Turn

- Draw me a face. Your face outline should be a circle with eyes, nose and a mouth. You can choose what shapes to use for each facial feature, and what color to draw them. Your face should look different from the one I did in class.

- Name your program Face.py and submit it to the Moodle Activity04 dropbox. You get 1 extra credit point for turning something in, 2 points for something correct.