

A+ Computer Science

PYTHON

Graphics & PyGAME

Using Libraries

There's already a ton of code written called libraries that we can use by importing. One that we use often is random

```
import random
```

```
print random.randint(2, 22)
```

Random

frequently used methods

Name	Use
<code>random.randint(a, b)</code>	Get a random integer between a and b, including a and b
<code>random.random()</code>	Get a random number between 0.0 and 1.0

Using Libraries

**To use the code we have to use
filename.methodName()**

```
import random
```

```
print random.randint(2, 22)    #python 2.x
```

```
print ( random.randint(2, 22) ) #python 3.x
```

Using Libraries

The library we will be using for graphics is pygame.

For now, most of the work to display the window is done for you in the runner. You will have to create, change or add things in another file

Pygame Basics

A pygame surface is like a piece of paper. You can draw lines, circles and other shapes on it. Sometimes you put other surfaces on it.

In the draw methods, it called window.

Pygame Basics

all imports go at the top

import pygame, sys

from pygame.locals import *

Pygame Basics

create the screen

```
pygame.init()
```

```
screen = pygame.display.set_mode((800,600))
```

Colors the screen white

```
screen.fill((255,255,255))
```

Updates the display

```
pygame.display.update()
```


Pygame Basics

```
# python 2.x
```

```
while True :
```

```
    for event in pygame.event.get():
```

```
        # Check the if the user closed window
```

```
        if event.type==QUIT :
```

```
            # End the program
```

```
            pygame.quit()
```

```
            sys.exit()
```

Pygame Basics

```
# python 3.x  
while ( True ):  
    for event in pygame.event.get():  
        # Check the if the user closed window  
        if ( event.type==QUIT ) :  
            # End the program  
            pygame.quit()  
            sys.exit()
```

pygame_basics.py

Colors

Colors are made using 3 integer parameters.

white = (255, 255, 255)

black = (0, 0, 0)

blue = (0, 0, 255)

red = (255, 0, 0)

Random Colors

Colors are made using 3 integer parameters.

r = random.randint(0, 256)

g = random.randint(0, 256)

b = random.randint(0, 256)

randColor = (r, g, b)

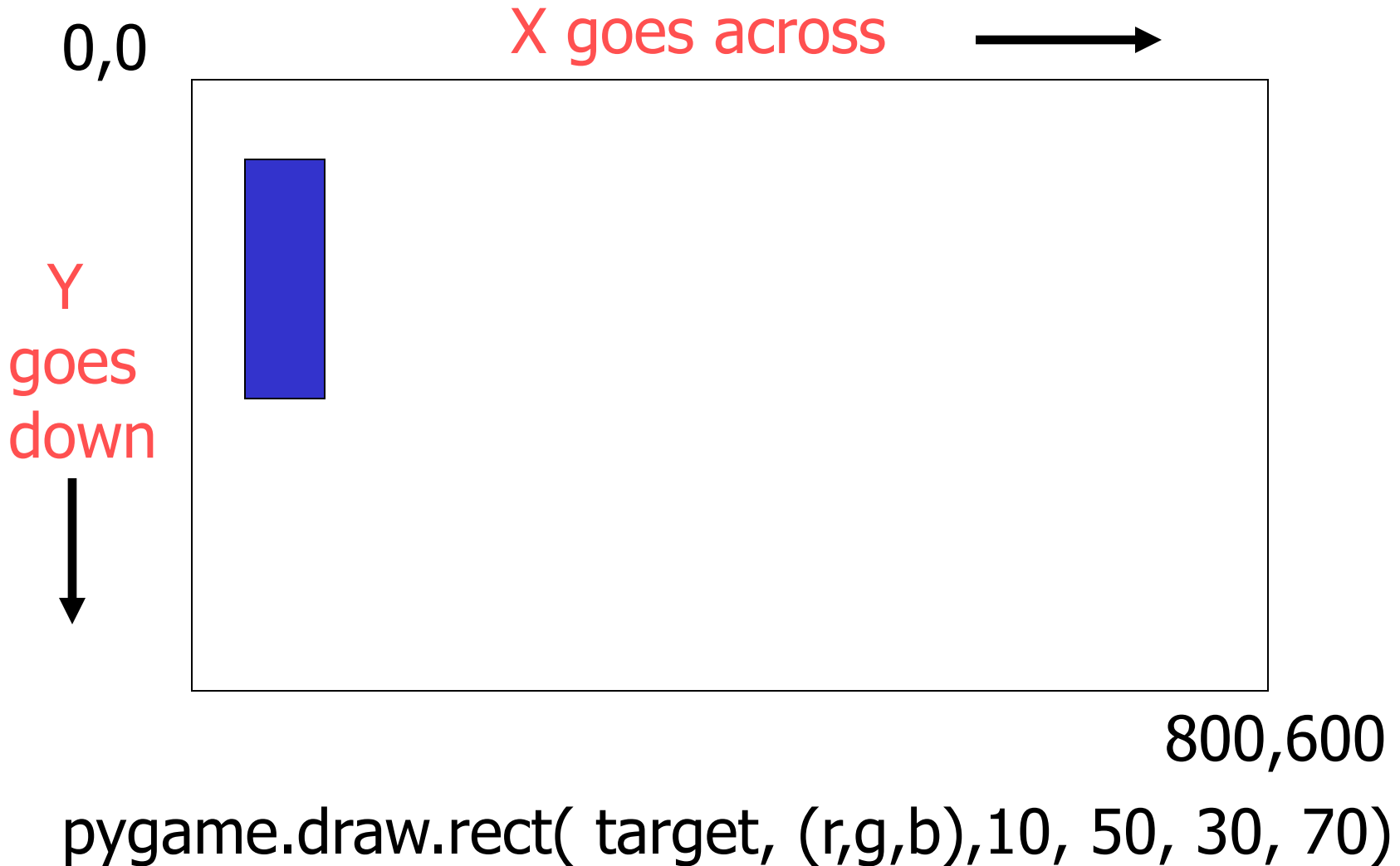
pygame_colors.py

Pygame Methods

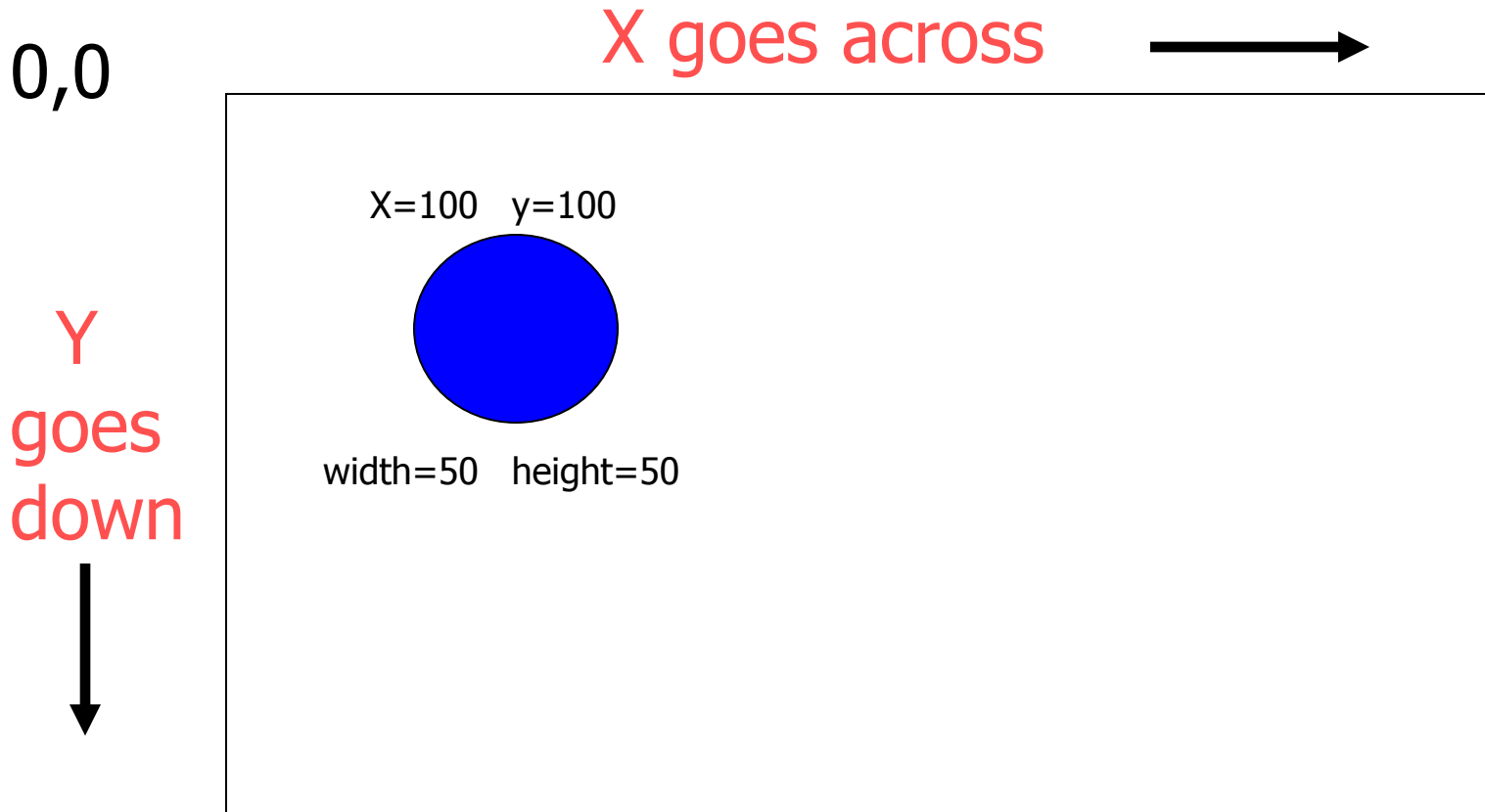
frequently used methods

Name	Use
draw.line	Connects two points
draw.rect	Draws a rectangle between 2 points
draw.circle	Draws a circle at a point
draw.ellipse	Draws an ellipse between points
draw.polygon	Draws a polygon between points

The Graphics Screen



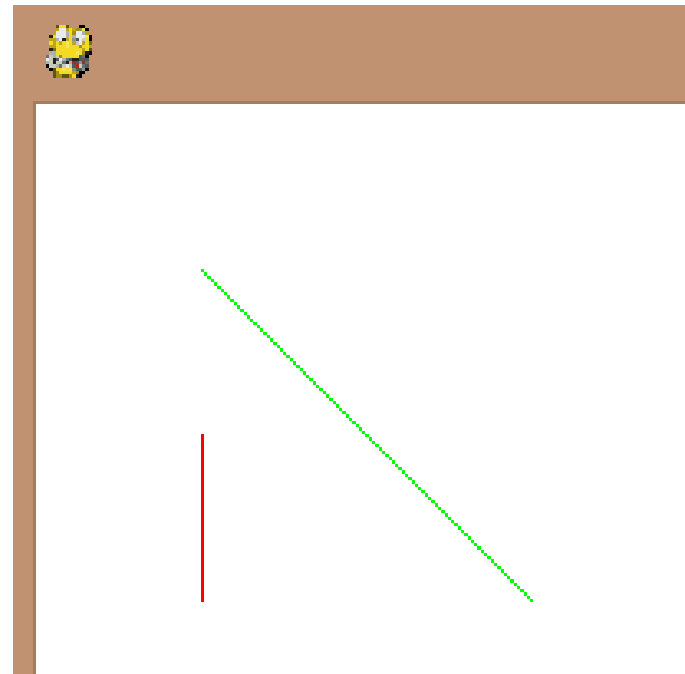
The Graphics Screen



```
pygame.draw.ellipse( target, (r,g,b),100, 100, 50, 50 );
```

Line / Lines

green = (0, 255, 0)
p1 = (50, 50)
p2 = (150, 150)



pygame.draw.line(screen, green, p1, p2)

pygame_lines.py



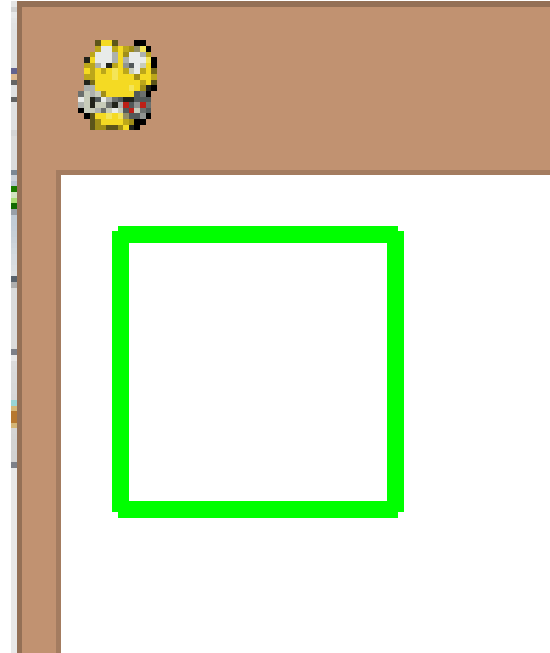
Rectangles

```
pygame.draw.rect(window, (r, g, b),  
                  (x, y, width, height))
```

***x* and *y* are the coordinates where the upper left corner of the rectangle will be placed**

***Width* and *height* are the width and height of the rectangle you want to draw**

Rectangles



green = (0, 255, 0)

box = (10,10,50,50)

pygame.draw.rect(screen, green, box, 3)

pygame_rectangles.py



Circles

```
pygame.draw.circle(window, (r, g, b),  
                    (x, y), rad)
```

**This draws a circle around the point (x, y)
with a radius of rad**

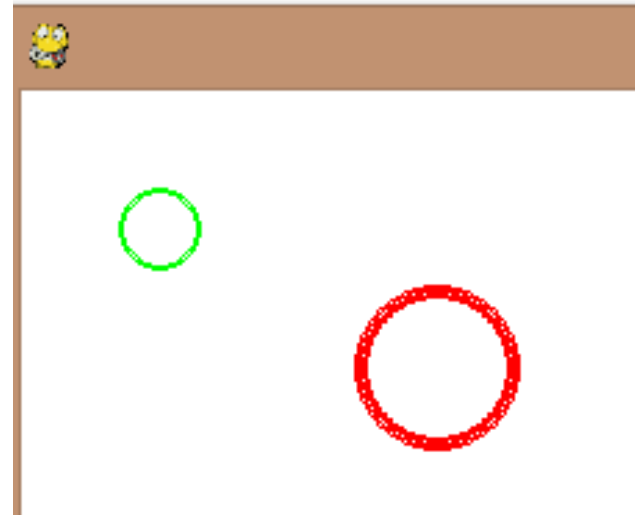
Circles

green = (0, 255, 0)

red = (255, 0, 0)

p1 = (50, 50)

p2 = (150, 100)



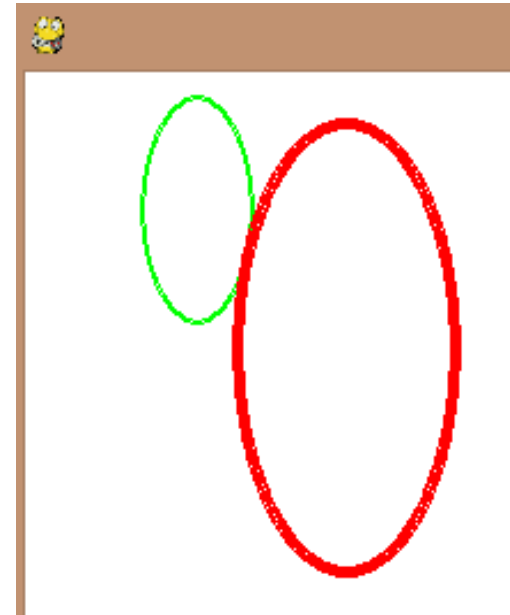
pygame.draw.circle(screen, green, p1, 15, 2)

pygame.draw.circle(screen, red, p2, 30, 5)

pygame_circles.py

Ellipses

green = (0, 255, 0)
red = (255, 0, 0)
r1 = (50, 10, 50, 100)
r2 = (90, 20, 100, 200)

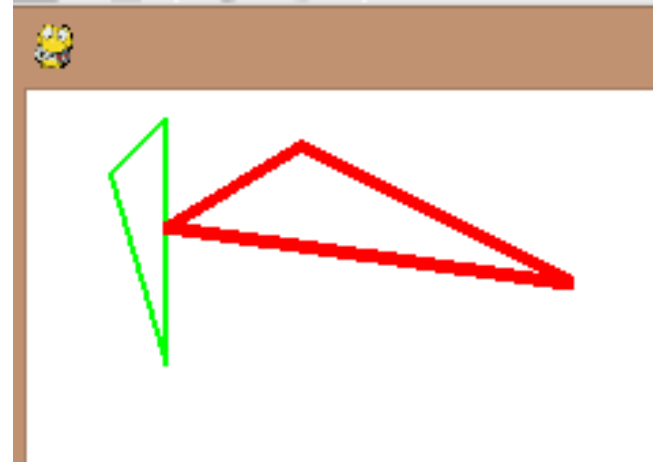


pygame.draw.ellipse(screen, green, r1, 2)
pygame.draw.ellipse(screen, red, r2, 5)

pygame_ellipses.py



Polygons



green = (0, 255, 0)

red = (255, 0, 0)

r1 = ((50, 10), (50, 100), (30, 30))

r2 = ((100, 20), (200, 70), (50, 50))

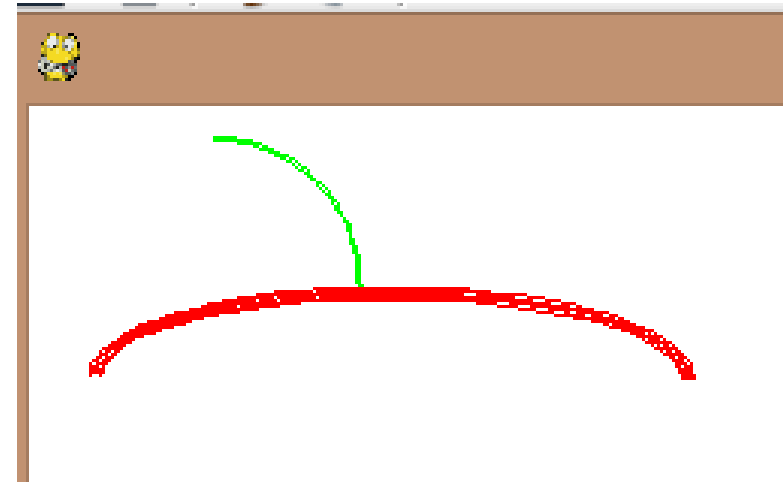
pygame.draw.polygon(screen, green, r1, 2)

pygame.draw.polygon(screen, red, r2, 5)

pygame_polygons.py

Arcs

```
green = (0, 255, 0)  
red = (255, 0, 0)  
r1 = ((10, 10), (100, 100))  
r2 = ((20, 60), (200, 60))
```



```
pygame.draw.arc(screen, green, r1, 0, pi/2, 2)  
pygame.draw.arc(screen, red, r2, 0, pi, 5)
```

pygame_arcs.py

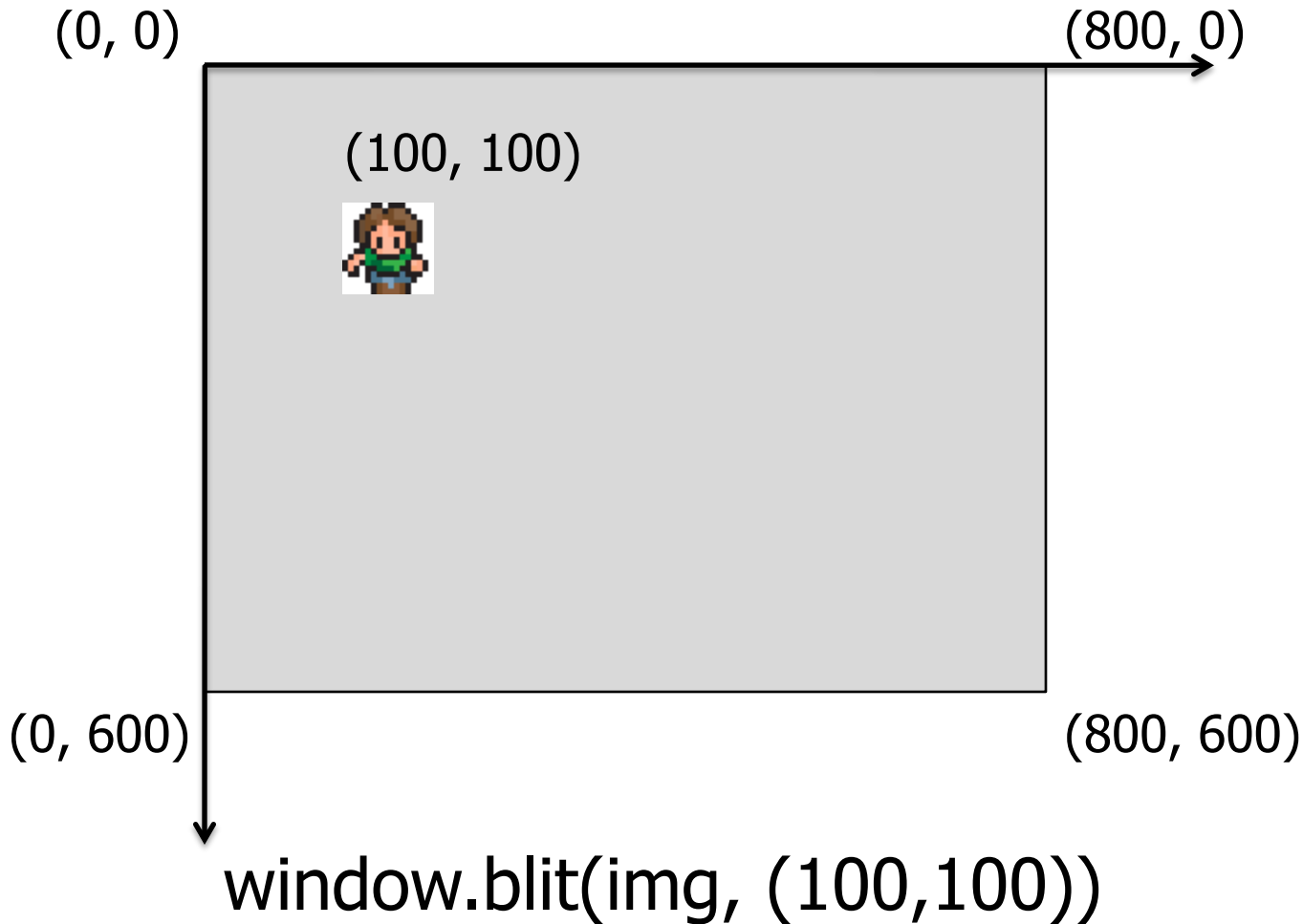


Pygame

frequently used methods

Name	Use
<code>pygame.image.load(file)</code>	This loads an image from a file and returns it as a surface
<code>surface.blit(img, (x,y))</code>	This draws an surface on another surface
<code>pygame.draw.rect(surface, color, Rect, width=0)</code>	Draws a rectangle on a surface
<code>pygame.draw.circle(surface, color, pos, radius, width=0)</code>	Draws a circle on a surface

Screen Coordinates



Pygame Methods

```
img = pygame.image.load("dude.gif")
```

pygame.image.load() loads an image as a surface

"dude.gif" is the file name of the image. It should be stored in the folder with your code

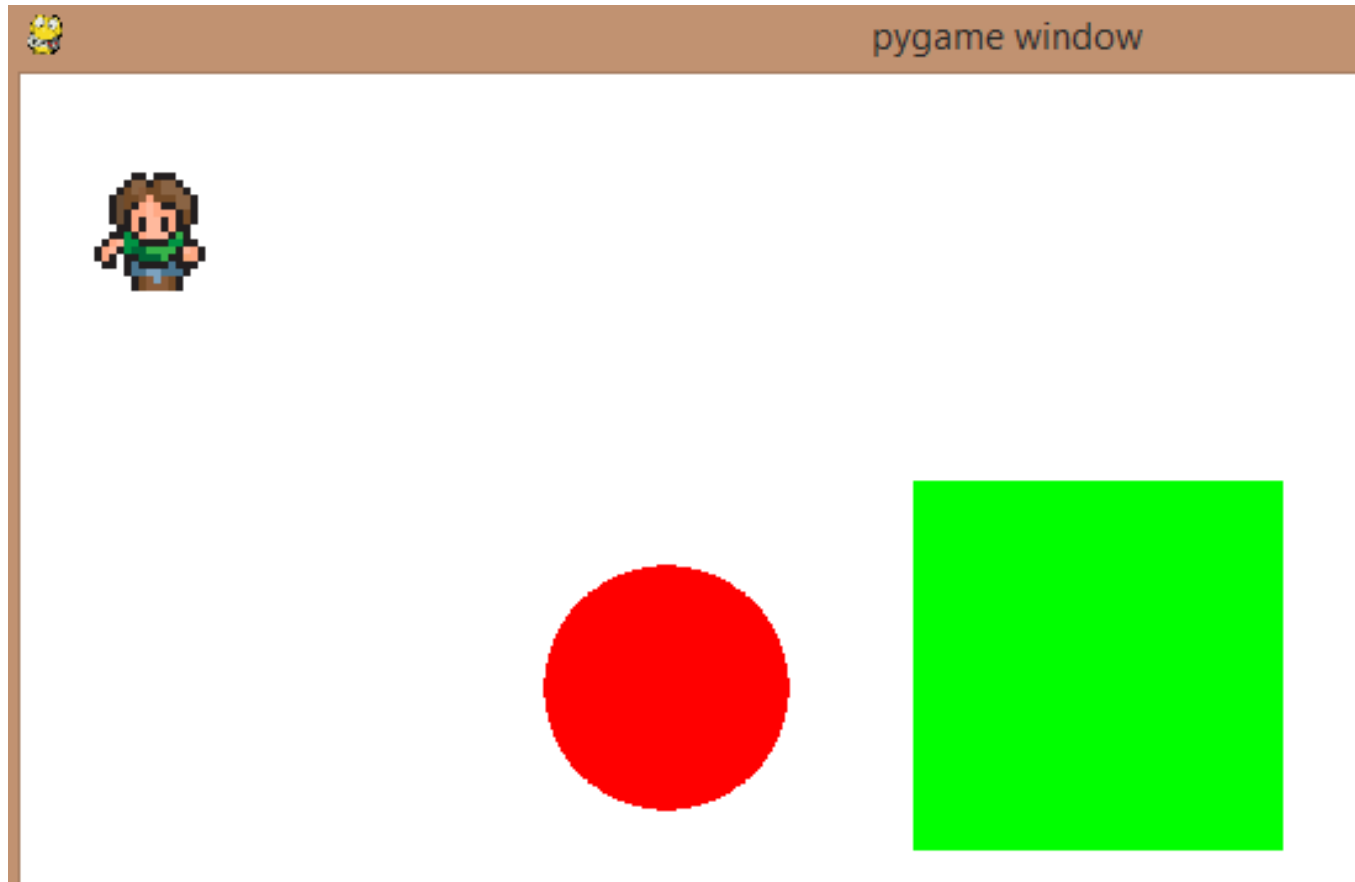
Pygame Methods

```
window.blit(img, (x, y))
```

Draws the surface *img* onto the surface *window*.

***x* and *y* are the coordinates where the upper left corner of the image will be placed**

Pygame Methods



pygame_image.py
graphics.py

Work on Programs!

Crank

Some Code!

A+ Computer Science

PYTHON

Graphics & PyGAME