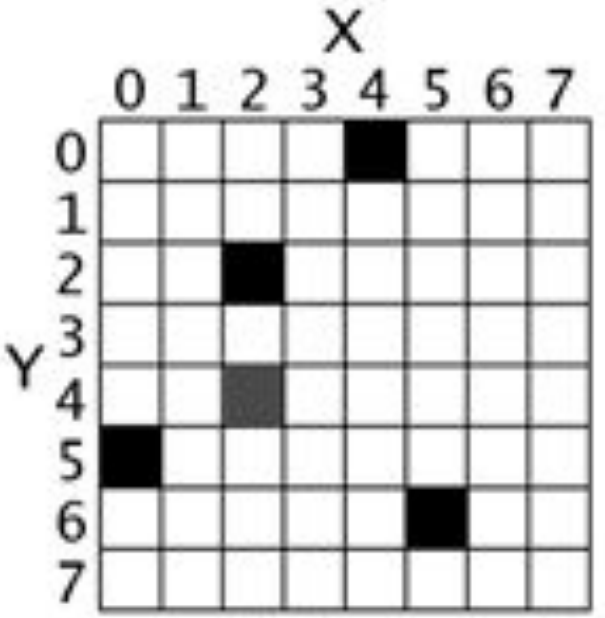
Bouncing Ball Game

Today

- Pygame basics
  - Surface object
  - Rect object
  - Drawing images
- Bouncing ball game

2D Pixel Coordinates

- 2D pixel coordinates (x, y)
  - Black: (4,0), (2,2), (0,5), (5,6)
  - gray: (2,4)

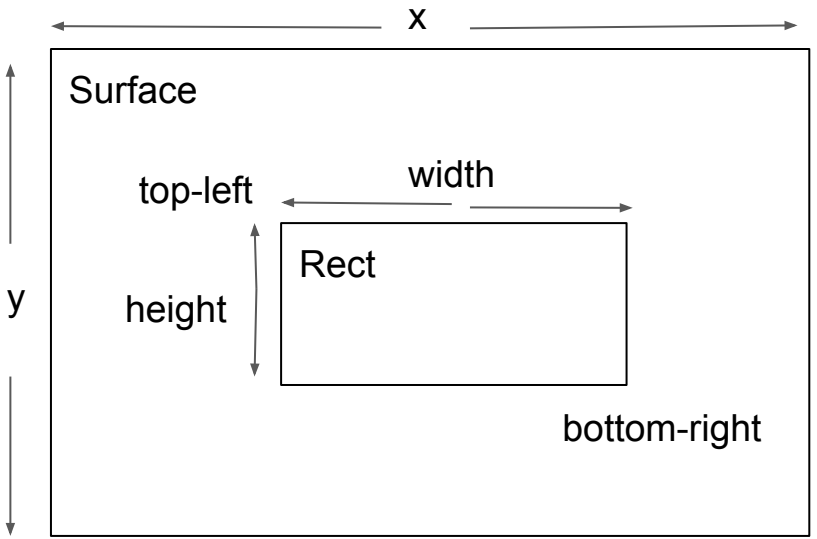


Surface Object

- Surface objects represent rectangle 2D images
  - Display surface:
    pygame.display.set_mode((width, height))
    pygame.display.set_caption(‘Hello World’)

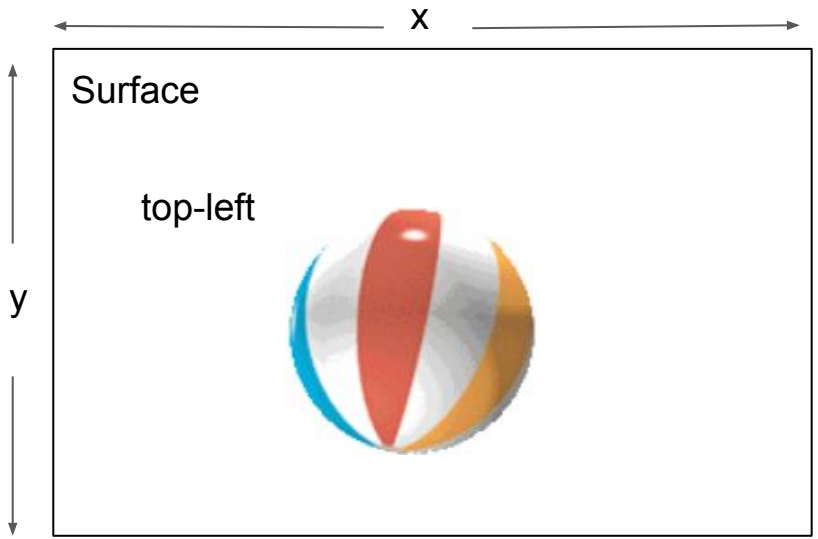
Rect Object

- Tuple of 4 integers
  - X coordinate of the top left corner (pixel)
  - Y coordinate of the top left corner (pixel)
  - Width of the rectangle (pixel)
  - Height of the rectangle (pixel)
- Create Rect Object
  - pygame.Rect(10, 20, 200, 300)



Draw image

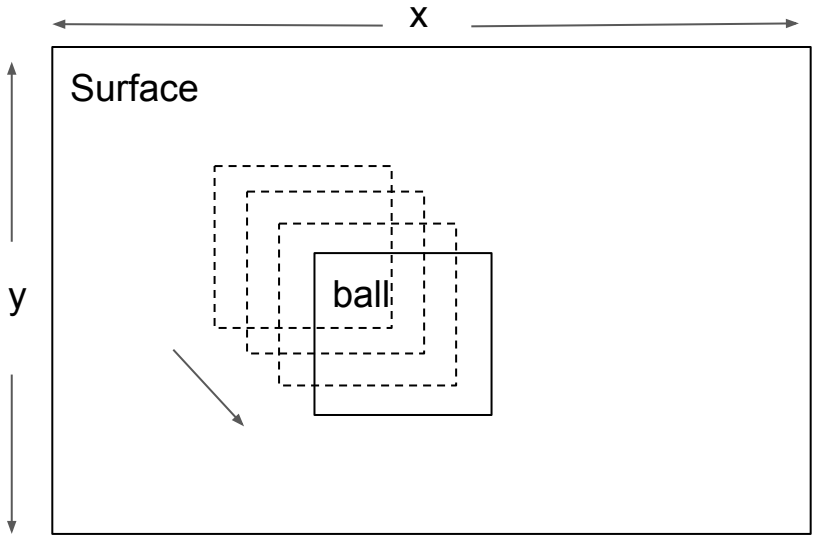
- Load images to surface objects from PNG, JPG, GIF and BMP files
  - ball=pygame.image.load(image_file)
- Get the object rectangle
  - ball_rect = ball.get_rect()
- Copy to display surface
  - screen.blit(ball, rect_object)



Animation

Animated images are the result of drawing an image on the screen, then a split second later drawing a slightly different image on the screen.

- Erase the screen
  screen.fill(black)
- Move object
  ballrect = ballrect.move((speed_x, speed_y))
- Copy the object to screen
  screen.blit(ball, ballrect)
- Change speed: next page
- Update monitor
  pygame.display.update()
- Pause
  time.sleep(0.01)
- Go to the first step



Bouncing Ball Game (2)

Bounce back on the boundary

```
if ballrect.left < 0 or ballrect.right > width:
    speed[0] = -speed[0]
if ballrect.top < 0 or ballrect.bottom > height:
    speed[1] = -speed[1]
```

code

```
import pygame, sys
from pygame.locals import QUIT
import time

speed = [1,1]
black = (0, 0, 0)
delay = 0.01
width, height = 500, 400

pygame.init()
screen = pygame.display.set_mode((width, height))
pygame.display.set_caption('Bouncing Ball')
ball = pygame.image.load('intro_ball.gif')
ballrect = ball.get_rect()

while True:
    for event in pygame.event.get():
        if event.type == QUIT:
            pygame.quit()
            sys.exit()

    screen.fill(black)
    ballrect = ballrect.move(speed)
    if ballrect.left < 0 or ballrect.right > width:
        speed[0] = -speed[0]
    if ballrect.top < 0 or ballrect.bottom > height:
        speed[1] = -speed[1]

    screen.blit(ball, ballrect)

    pygame.display.update()
    time.sleep(delay)
```

Download ball image file

https://github.com/zhihongzeng2002/pythongame/blob/master/2/intro_ball.gif

Save to the work folder

Intro_ball.gif