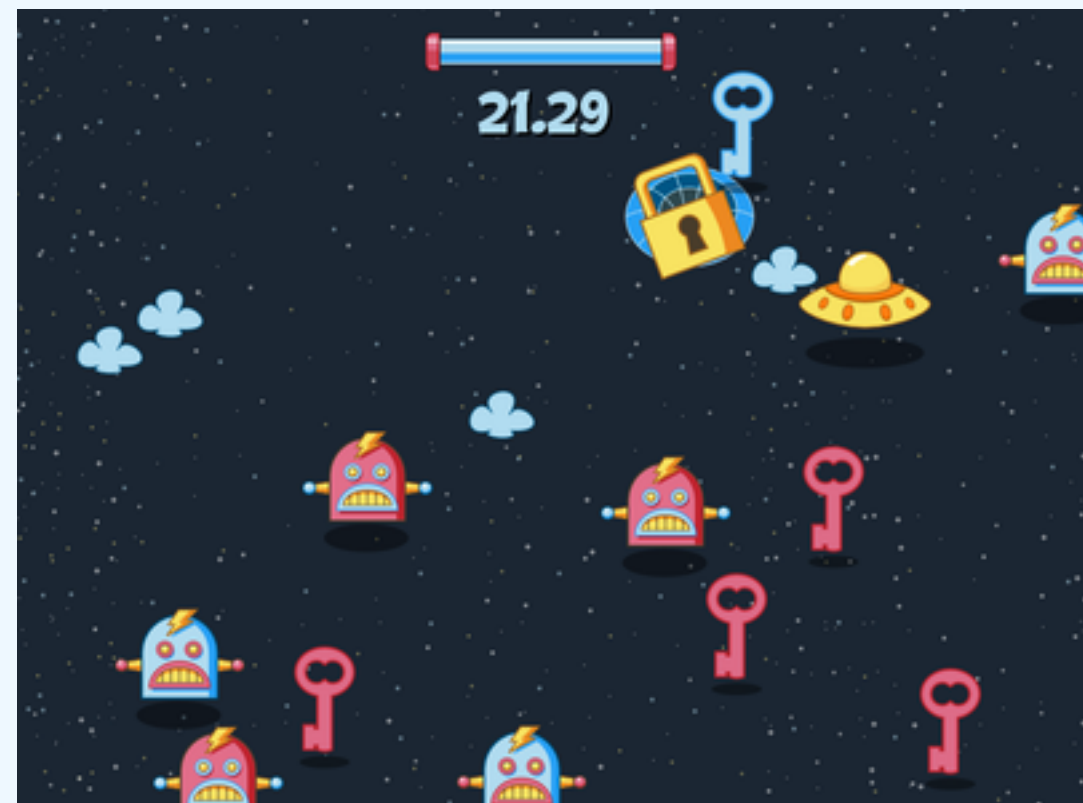
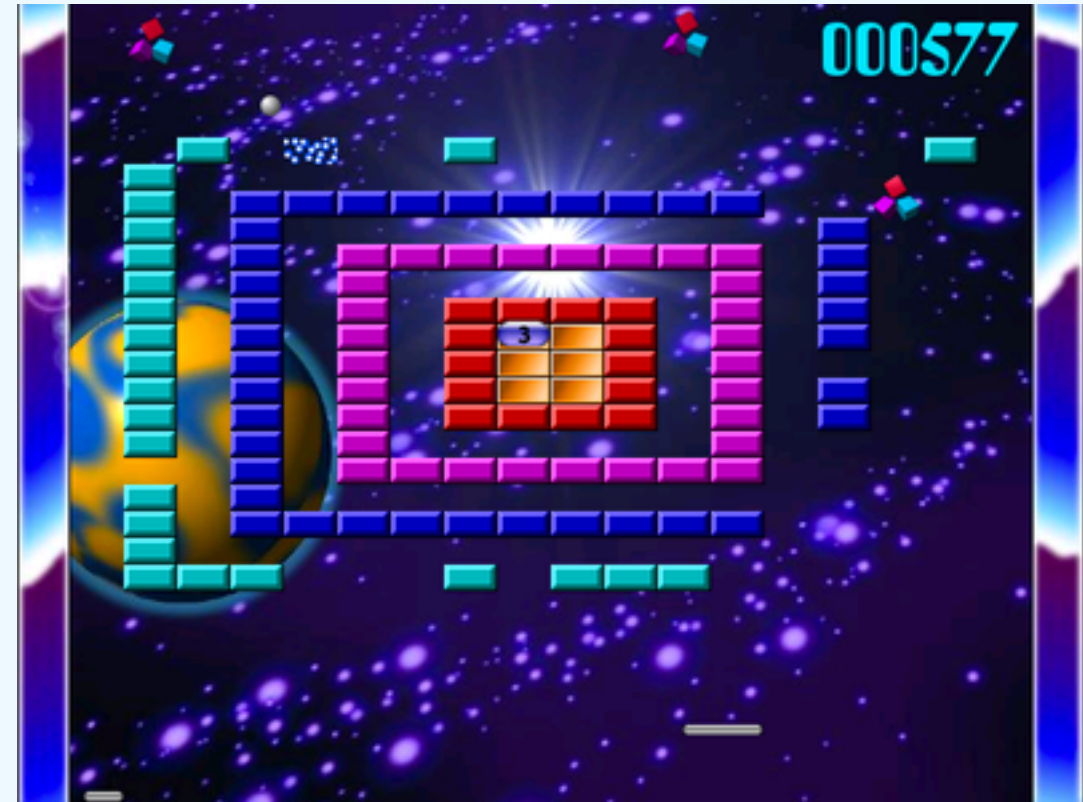
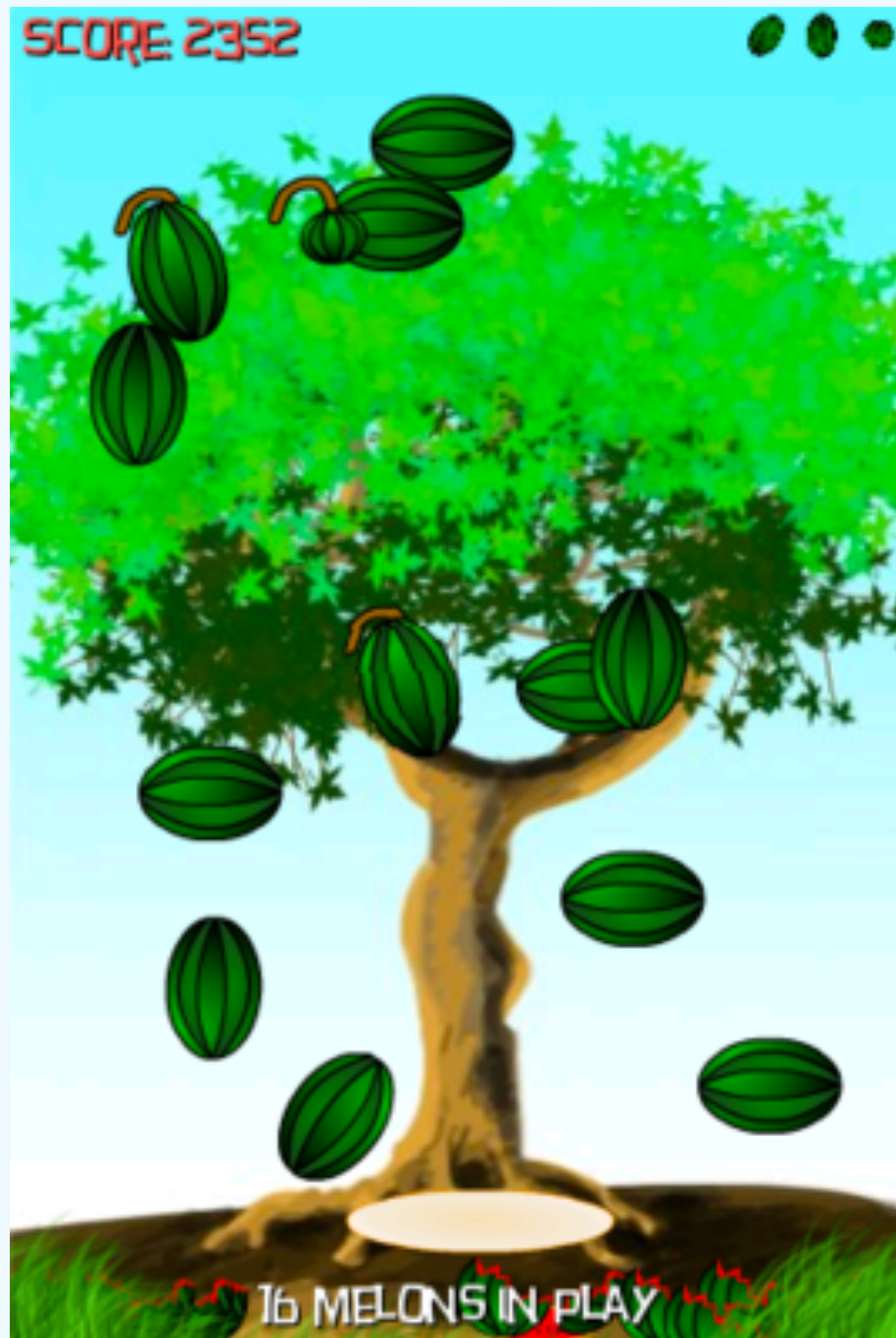


Workshop 1 :

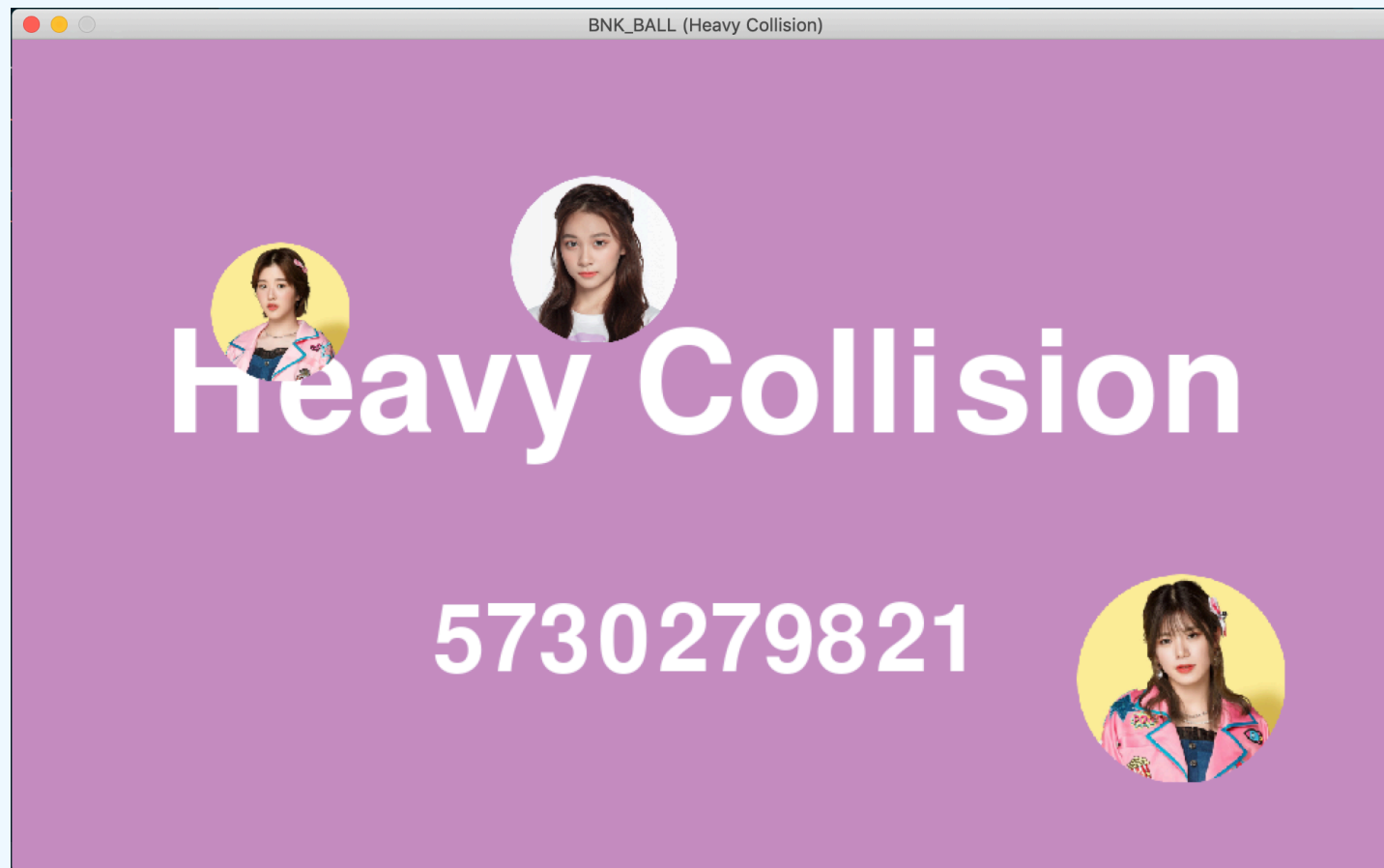


```
import pygame
```

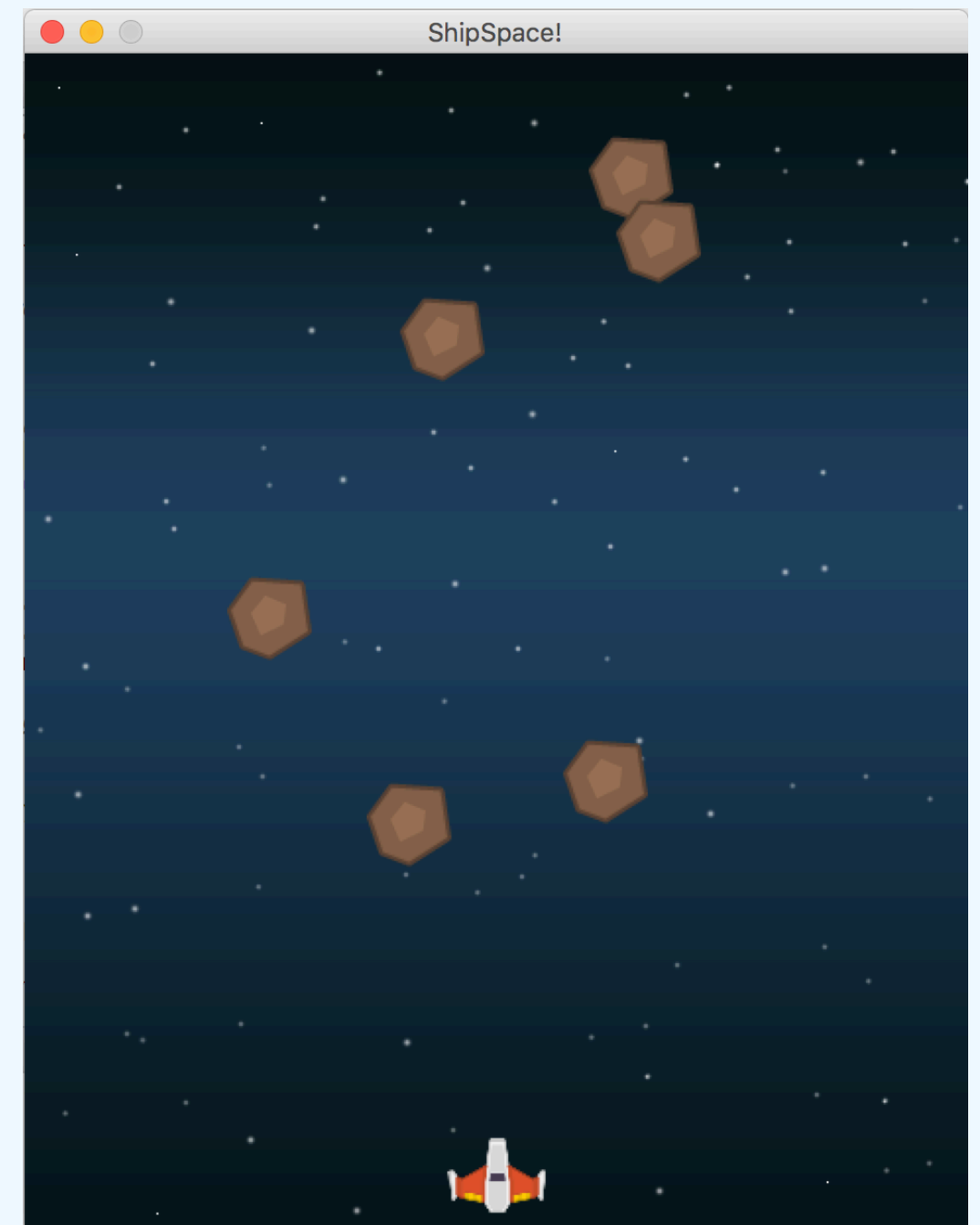
Examples of pygame project



Workshop 1 :



<https://youtu.be/hJb2mgR7nV0>



Workshop 1 :



Goal :

- ใช้ while , for , if-else ให้เป็น
- การสร้าง object และเรียกใช้ method

Topic

00-start_pygame.py

01-get_rect.py

02-Lab_bnk_ball.py

03-sprite.py

04-Lab_create_screen.py

05-Lab_ship_space.py

Topic

00-start_pygame.py

01-get_rect.py

02-Lab_bnk48_ball.py

Review

if - elif - else

```
if a > 10 :  
    b += 1
```

```
elif a > 5 :  
    b += 2
```

```
else :  
    b += 3
```

```
if a > 10 :  
    b += 1
```

```
if a > 5 :  
    b += 2
```

```
else :  
    b += 3
```

Review

for

```
for i in range( 0 , 8 ):  
    print( i )
```

ทำ 8 รอบ
แสดงผลดังนี้

0
1
2
3
4
5
6
7

```
box = [ 3 , 1 , 4 , 2 ]
```

```
for c in box:  
    print ( c )
```

หยิบของใน box ออกมาทีละชิ้น
แสดงผลลัพท์ดังนี้

3
1
4
2

Review

while

```
i = 1
while i < 6 :
    print( i )
    i+=1
```

แสดงผลดังนี้

1
2
3
4
5

```
i = 1
while True:
    print(i)
    i+=1
    if i >= 6 :
        break
```

ใช้ while True และ break
แสดงผลลัพท์ดังนี้

1
2
3
4
5

```
i = 1
running = True
while running:
    print(i)
    i+=1
    if i >= 6 :
        running = False
```

กำหนด Boolean
เพื่อควบคุมการทำงานของ while
แสดงผลลัพท์ดังนี้

1
2
3
4
5

Pygame Structure

```
import pygame as pg # import library pygame
```

```
pg.init() # เพื่อ initialize pygame modules
```

```
width = 700 # กำหนดความกว้างของจอภาพ
```

```
height = 500 # กำหนดความสูงของจอภาพ
```

```
FPS = 60 # กำหนดความเร็ว FPS
```

```
screen = pg.display.set_mode((width,height)) # กำหนดขนาดของหน้าจอ
```

```
clock = pg.time.Clock() # สร้าง clock
```

```
pygame.error: font not initialized
```

while loop :

```
clock.tick(FPS) # ทำให้เกมแสดงผลด้วยความเร็วที่เหมาะสม
```

```
for event in pg.event.get(): # รอรับ action จากผู้ใช้งานทาง mouse หรือ keyboard
```

```
    if event.type == pg.QUIT: # ตรวจสอบว่าผู้ใช้คลิกปุ่มปิดหรือไม่
```

```
        pg.quit() # ใส่ในกรณีที่ไม่ได้เขียนคำสั่ง Quit ใน event ปิดโปรแกรม
```

```
pg.display.flip() # อัปเดต content ลงใน screen
```

Pygame Structure

```
import pygame as pg
```

```
pg.init()
```

```
width = 700
```

```
height = 500
```

```
FPS = 60
```

```
screen = pg.display.set_mode( ( width , height ) )
```

```
clock = pg.time.Clock( )
```

```
while loop :
```

```
    clock.tick(FPS)
```

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

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

```
            pg.quit( )
```

```
    pg.display.flip( )
```

00-start_pygame.py

คำสั่งสำหรับการ fill สี และ draw shape

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

หรือ

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

```
pygame.draw.rect( screen , red , [ 55 , 200 , 100 , 70 ] , 0 )  
pygame.draw.line( screen , green , [ 20 , 35 ] , [ 250 , 160 ] , 5 )  
pygame.draw.ellipse( screen , black , [ 340 , 100 , 250 , 100 ] , 2 )
```

คำสั่งสำหรับการ draw Text

font_name = pg.font.match_font('arial')	# กำหนดชื่อ Font
font = pg.font.Font(font_name, 80)	# กำหนดขนาด font
text_surface = font.render("CP NEE PAI", True, black)	# กำหนด Text และสี
text_rect = text_surface.get_rect()	# แปลง Surface เป็น object
text_rect.midtop = (350, 250)	# ระบุตำแหน่งของ text
screen.blit(text_surface, text_rect)	# เอา Text ใส่ลงใน object ของ Text นั้น

การ pack surface โดย .flip()

```
import pygame as pg
```

```
pg.init()
```

```
black = (0,0,0)
```

```
white = (255,255,255)
```

```
green = (0,255,0)
```

```
red = (255,0,0)
```

```
width = 700
```

```
height = 500
```

```
FPS = 60
```

```
screen = pg.display.set_mode((width,height))
```

```
pg.display.set_caption("ComProg Sec 5")
```

```
clock = pg.time.Clock()
```

```
while True:
```

```
    clock.tick(FPS)
```

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

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

```
            pg.quit()
```

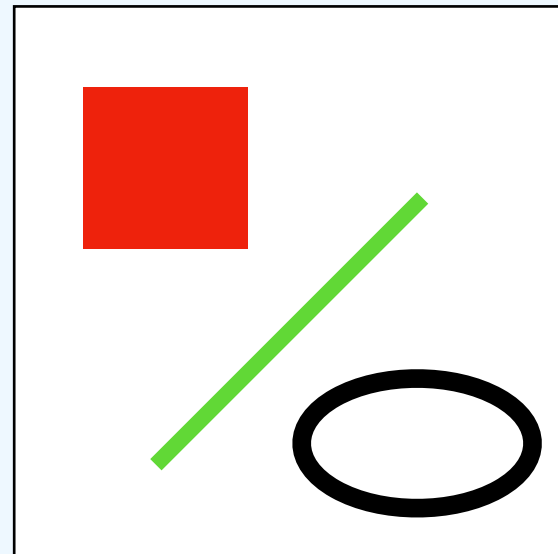
```
    screen.fill(white)
```

```
    pg.draw.rect(screen,red,[55,200,100,70],0)
```

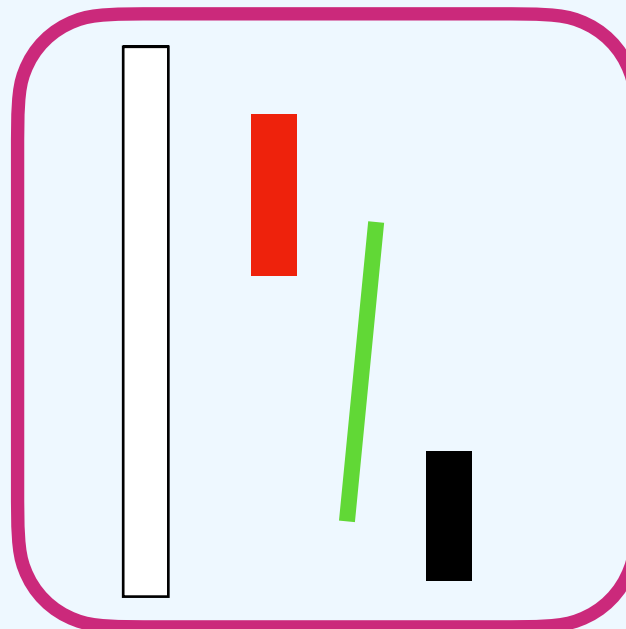
```
    pg.draw.line(screen, green ,[20,35],[250,160],5)
```

```
    pg.draw.ellipse(screen , black , [340,100,250,100], 2)
```

```
    pg.display.flip()
```



Surface



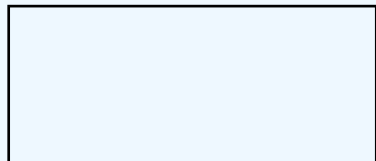
การโหลดภาพและการสร้าง object ให้กับภาพ

Surface



```
car_img = pg.image.load("source/img/Car.png").convert_alpha()
```

Object (มีความสามารถพิเศษ)



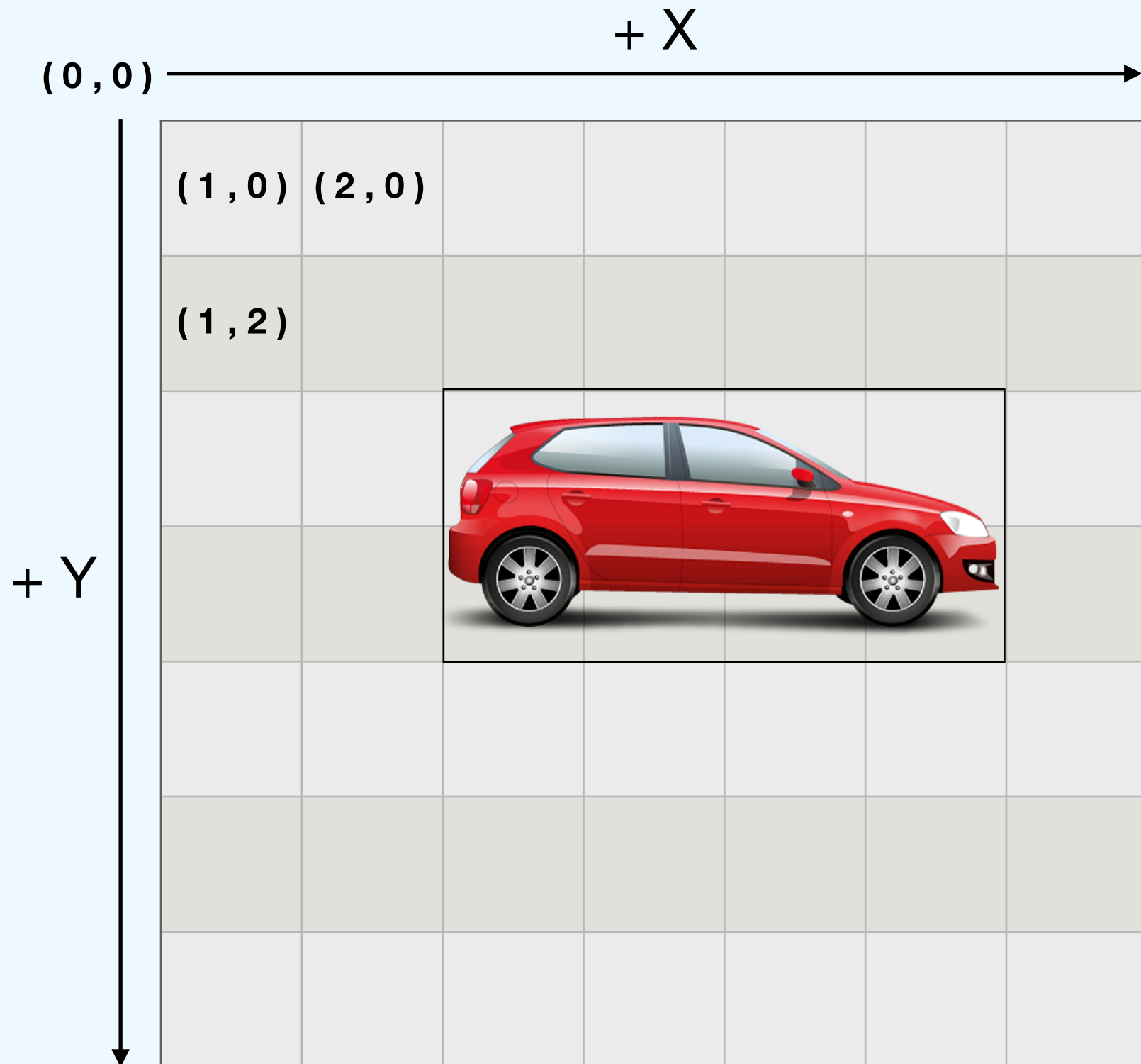
```
car_rect = car_img.get_rect()
```

Pack image with Object



```
screen.blit( car_img , car_rect )
```


attribute และ function ของ object จาก get_rect()



`car_rect.left` บอกค่า x ของซ้ายของรูป **2**

`car_rect.right` บอกค่า x ของขวาของรูป **6**

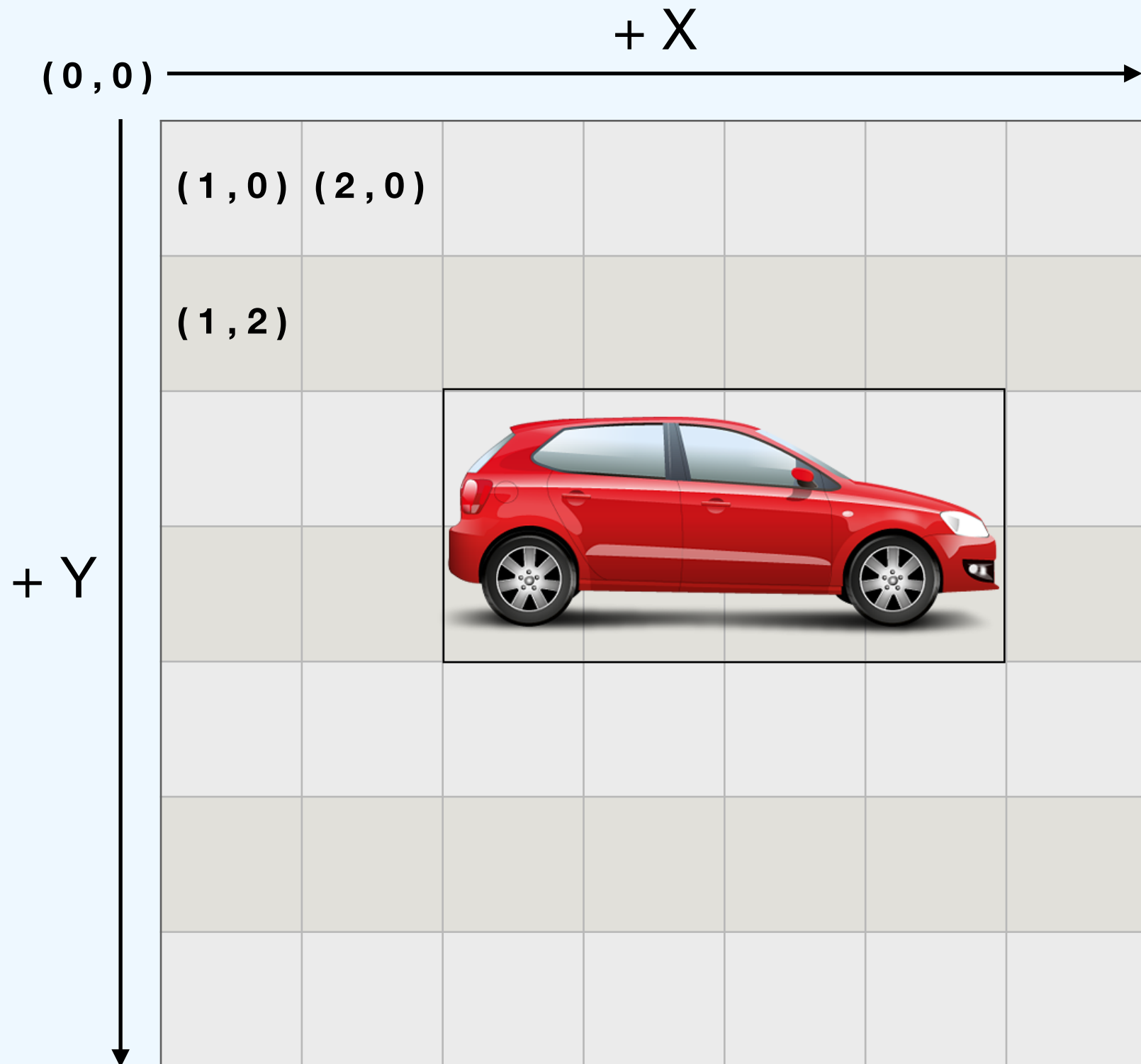
`car_rect.top` บอกค่า y ของบนของรูป **2**

`car_rect.bottom` บอกค่า y ของล่างของรูป **4**

function

`car_rect.move([1,1])`

attribute และ function ของ object จาก get_rect()



`car_rect.left` บอกค่า x ของซ้ายของรูป **2**

`car_rect.right` บอกค่า x ของขวาของรูป **6**

`car_rect.top` บอกค่า y ของบนของรูป **2**

`car_rect.bottom` บอกค่า y ของล่างของรูป **4**

function

`car_rect.move([-1 , 2])`

01-get_rect.py

```
import pygame as pg
pg.init()
```

```
width= 320
height = 240
FPS = 60
speed = [2, 2]
black = 0, 0, 0
```



```
screen = pg.display.set_mode((width,height))
```

```
ball = pg.image.load("source/img/intro_ball.gif")
clock = pg.time.Clock()
```

```
ballrect = ball.get_rect()
```

```
while True:
```

```
    clock.tick(FPS)
```

```
    for event in pg.event.get():
        if event.type == pg.QUIT:
            pg.quit()
```

```
        ballrect = ballrect.move(speed)
```

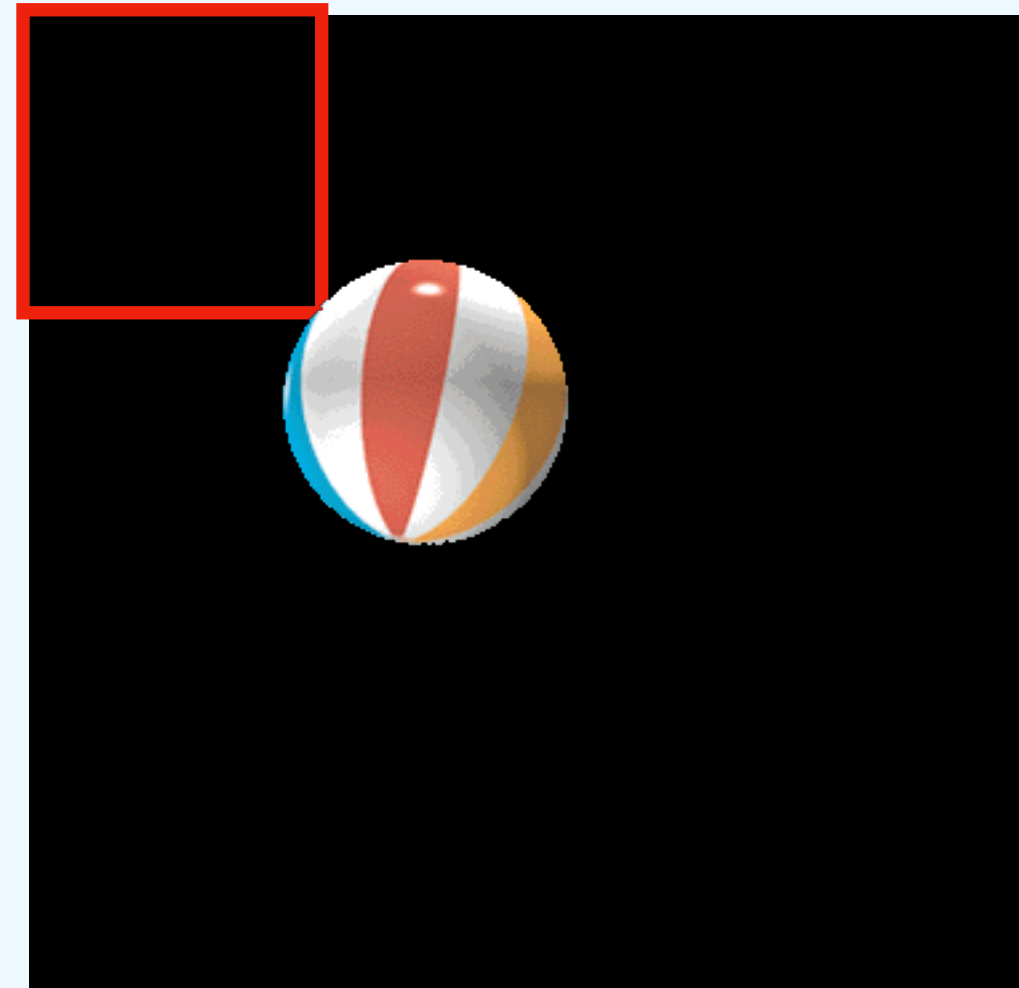
```
        if ballrect.left < 0 or ballrect.right > width:
            speed[0] = -speed[0]
```

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

```
    screen.fill(black)
```

```
    screen.blit(ball, ballrect)
```

```
    pg.display.flip()
```



```
import pygame as pg
pg.init()
```

```
width= 320
height = 240
FPS = 60
speed = [2, 2]
black = 0, 0, 0
```

```
screen = pg.display.set_mode((width,height))
```

```
ball = pg.image.load("source/img/intro_ball.gif")
clock = pg.time.Clock()
```

```
ballrect = ball.get_rect()
```

```
while True:
```

```
    clock.tick(FPS)
```

```
    for event in pygame.event.get():
        if event.type == pg.QUIT:
            pg.quit()
```

```
        ballrect = ballrect.move(speed)
```

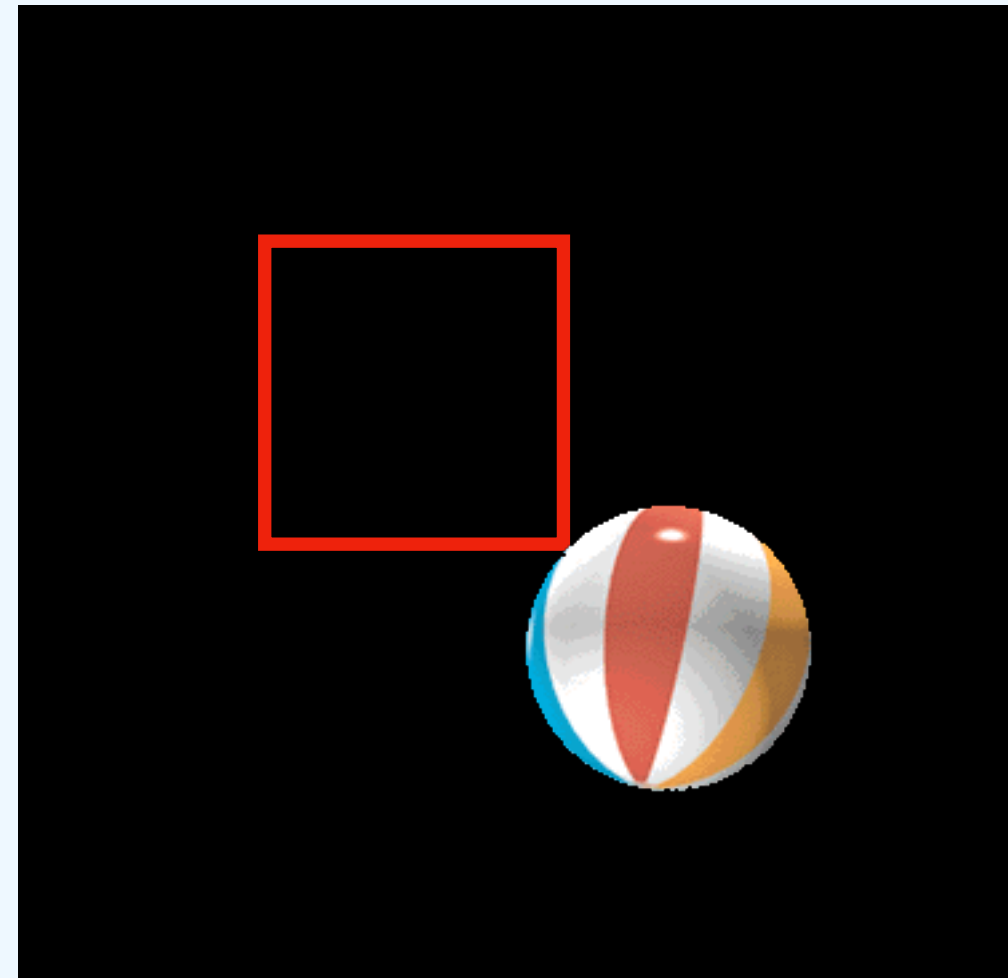
```
        if ballrect.left < 0 or ballrect.right > width:
            speed[0] = -speed[0]
```

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

```
    screen.fill(black)
```

```
    screen.blit(ball, ballrect)
```

```
    pg.display.flip()
```



02-Lab_bnk48_ball.py



