

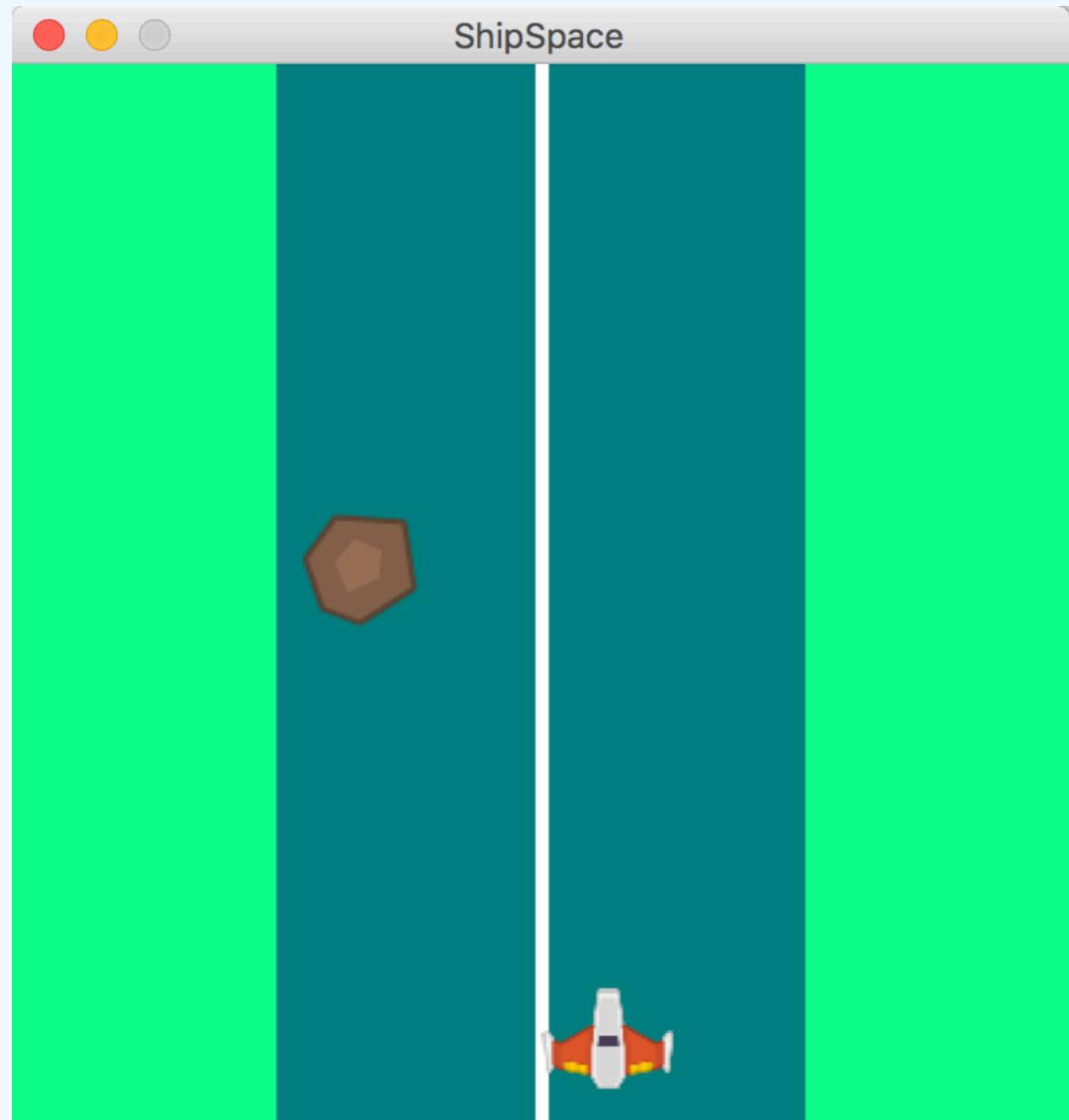
Topic

03-sprite.py [Tutorial]

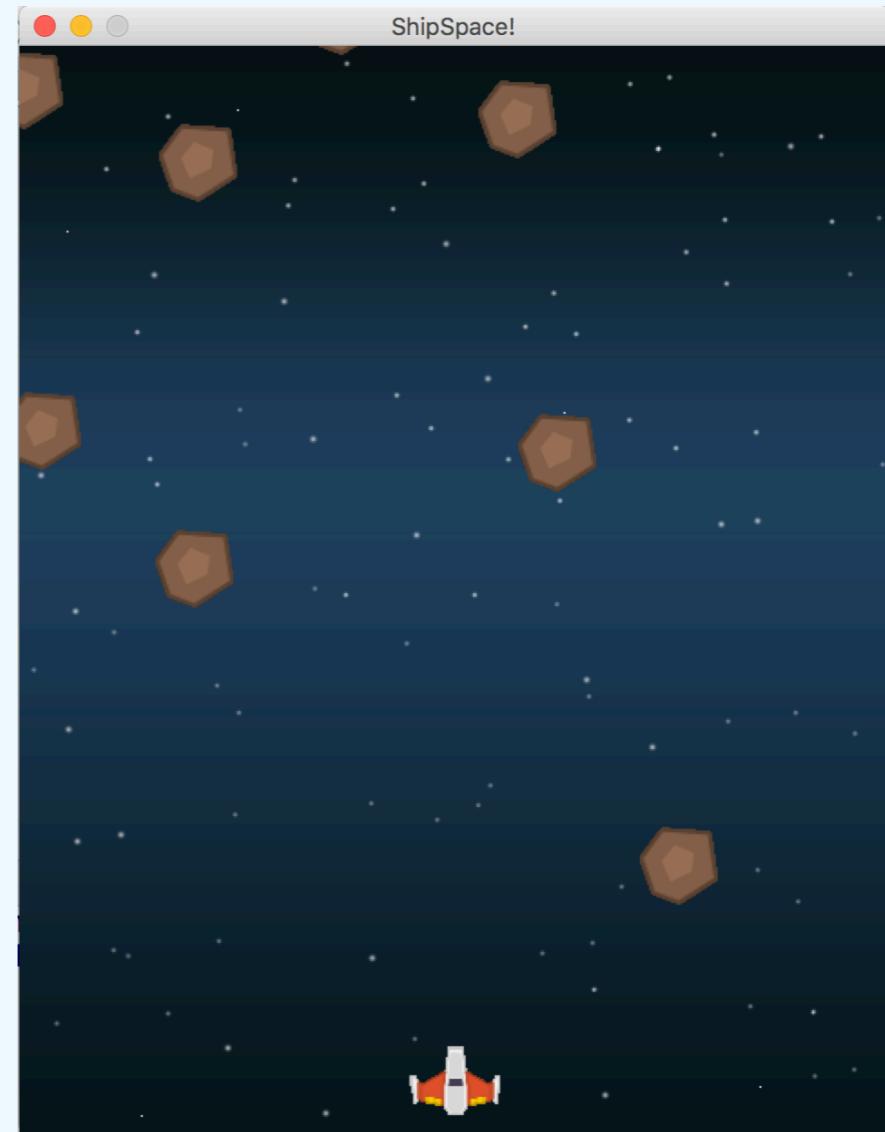
04-Lab_ship_space.py [HW]

Topic

03-sprite.py [Tutorial]



04-Lab_ship_space.py



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
```

while loop :

```
clock.tick(FPS)    # ทำให้เกมแสดงผลด้วยความเร็วที่เหมาะสม

for event in pg.event.get():    # รอรับ action จากผู้ใช้ผ่านทาง mouse หรือ keyboard
    if event.type == pg.QUIT:    # ตรวจสอบว่าผู้ใช้คลิกปุ่มปิดหรือไม่
        pg.quit()                # ใส่ในกรณีที่ไม่ได้เขียนคำสั่ง Quit ใน event ปิดโปรแกรม

pg.display.flip()    # อัพเดท 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()

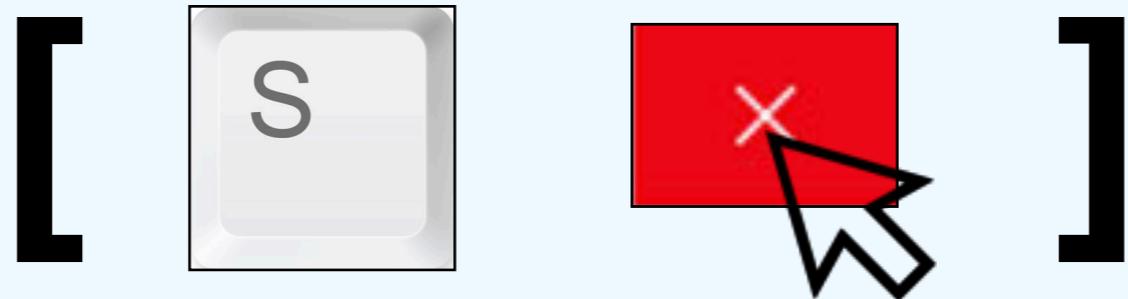
while loop :

    clock.tick(FPS)

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

    pg.display.flip()
```

การรับและตรวจสอบ event



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

คำสั่งสำหรับการ fill สี และ draw shape

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

หรือ

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

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 ECN", 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
```

ทำไมต้องกำหนด FPS ?

```
import pygame
```

```
pg.init()
```

```
width = 700
```

```
height = 500
```

```
FPS = 60
```

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

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

```
while loop :
```

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

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

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

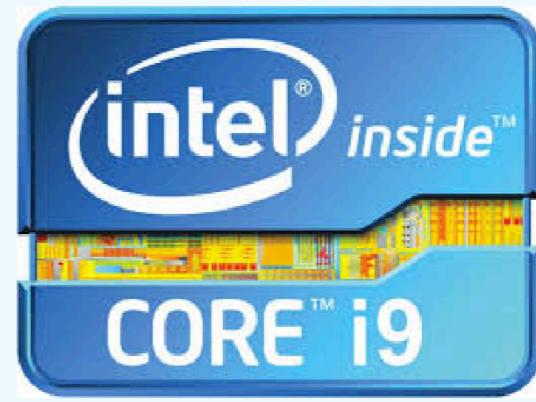
```
            pygame.quit( )
```

```
    pygame.display.flip( )
```

ทำไมต้องกำหนด FPS ?



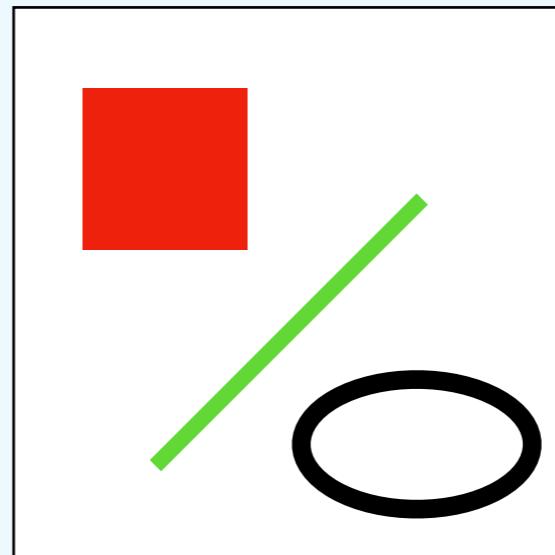
```
import pygame  
  
pygame.init()  
  
width = 700  
height = 500  
FPS = 60  
screen = pygame.display.set_mode( ( width , height ) )  
clock = pygame.time.Clock()  
  
while loop :  
  
    clock.tick(FPS)  
  
    for event in pygame.event.get( ):  
        if event.type == pygame.QUIT:  
            pygame.quit( )  
  
    pygame.display.flip( )
```



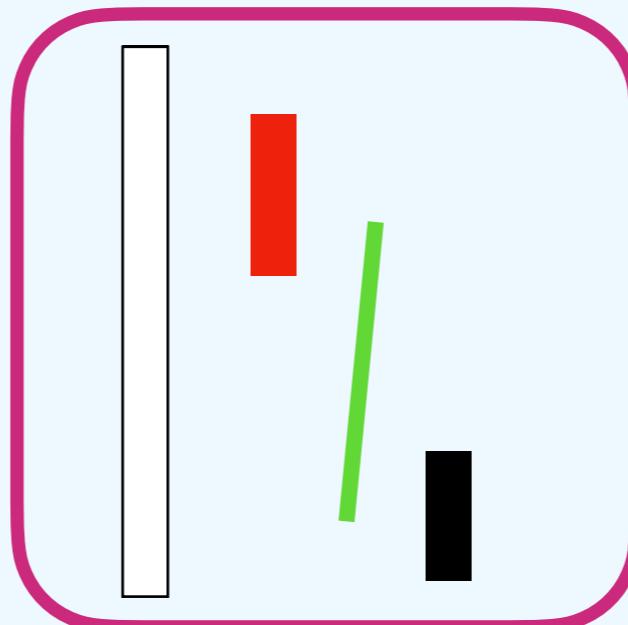
```
import pygame  
  
pygame.init()  
  
width = 700  
height = 500  
FPS = 60  
screen = pygame.display.set_mode( ( width , height ) )  
clock = pygame.time.Clock()  
  
while loop :  
  
    clock.tick(FPS)  
  
    for event in pygame.event.get( ):  
        if event.type == pygame.QUIT:  
            pygame.quit( )  
  
    pygame.display.flip( )
```

การ pack surface โดย .flip()

```
import pygame  
  
pygame.init()  
  
black = (0,0,0)  
white = (255,255,255)  
green = (0,255,0)  
red = (255,0,0)  
  
width = 700  
height = 500  
FPS = 60  
screen = pygame.display.set_mode((width,height))  
pygame.display.set_caption("ComProg Sec 5")  
clock = pygame.time.Clock()  
  
while True:  
  
    clock.tick(FPS)  
  
    for event in pygame.event.get():  
        if event.type == pygame.QUIT:  
            pygame.quit()  
  
    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)  
  
    pygame.display.flip()
```



Surface



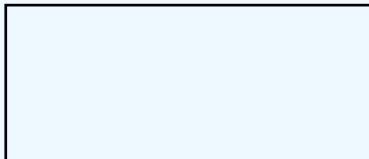
การโหลดภาพและการสร้าง object ให้กับภาพ

Surface



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

Object (มีความสามารถพิเศษ)



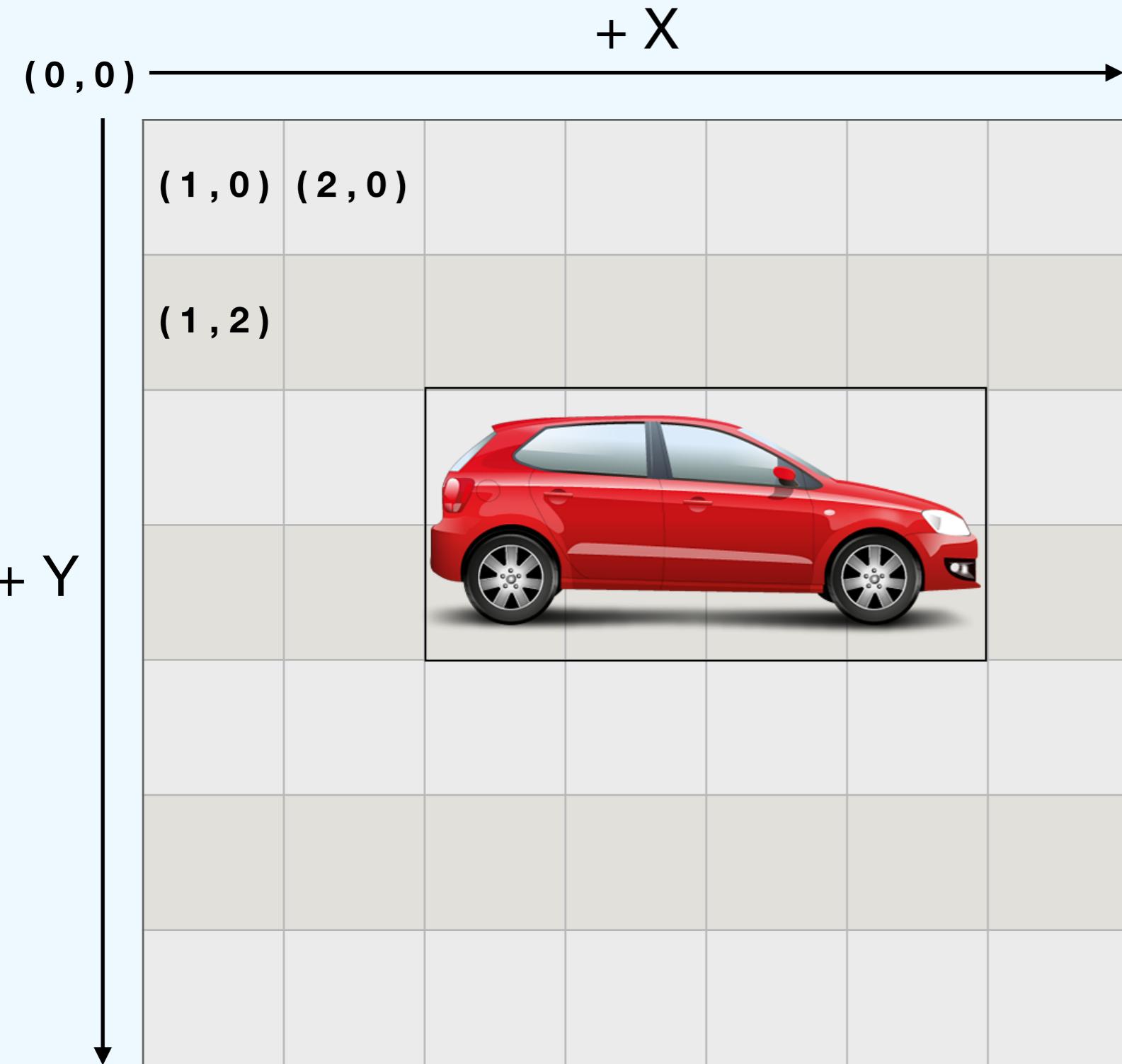
```
car_rect = car_img.get_rect()
```

Pack image with Object



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

attribute และ function ของ object จาก get_rect()

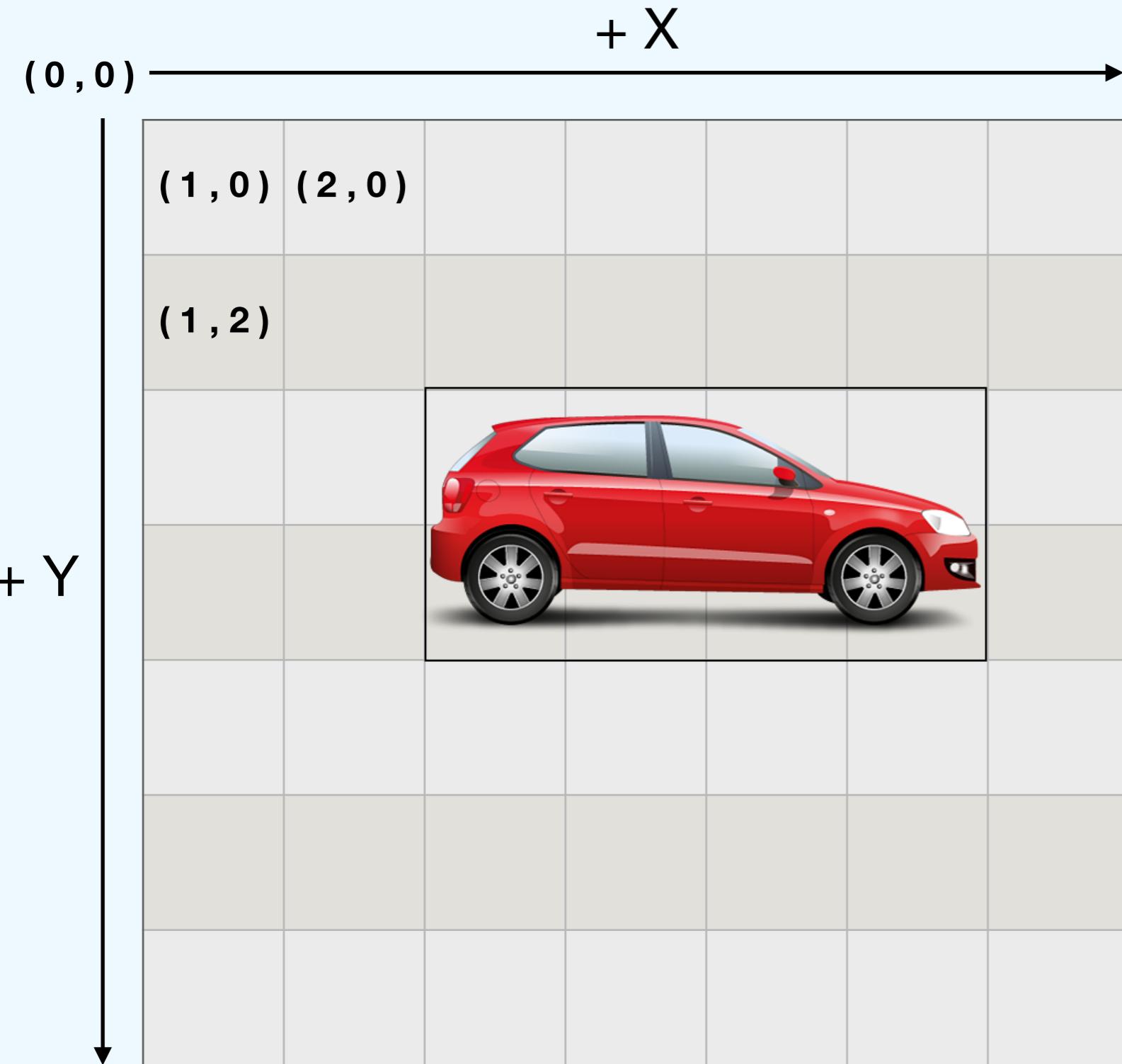


| | | |
|-----------------|------------------------|---|
| car_rect.left | บอกค่า x ของซ้ายของรูป | 2 |
| car_rect.right | บอกค่า x ของขวาของรูป | 6 |
| car_rect.top | บอกค่า y ของบนของรูป | 2 |
| car_rect.bottom | บอกค่า y ของล่างของรูป | 4 |

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])

```
import pygame
pygame.init()

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

screen = pygame.display.set_mode((width,height))

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

ballrect = ball.get_rect()

while True:

    clock.tick(FPS)

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

    ballrect = ballrect.move(speed)

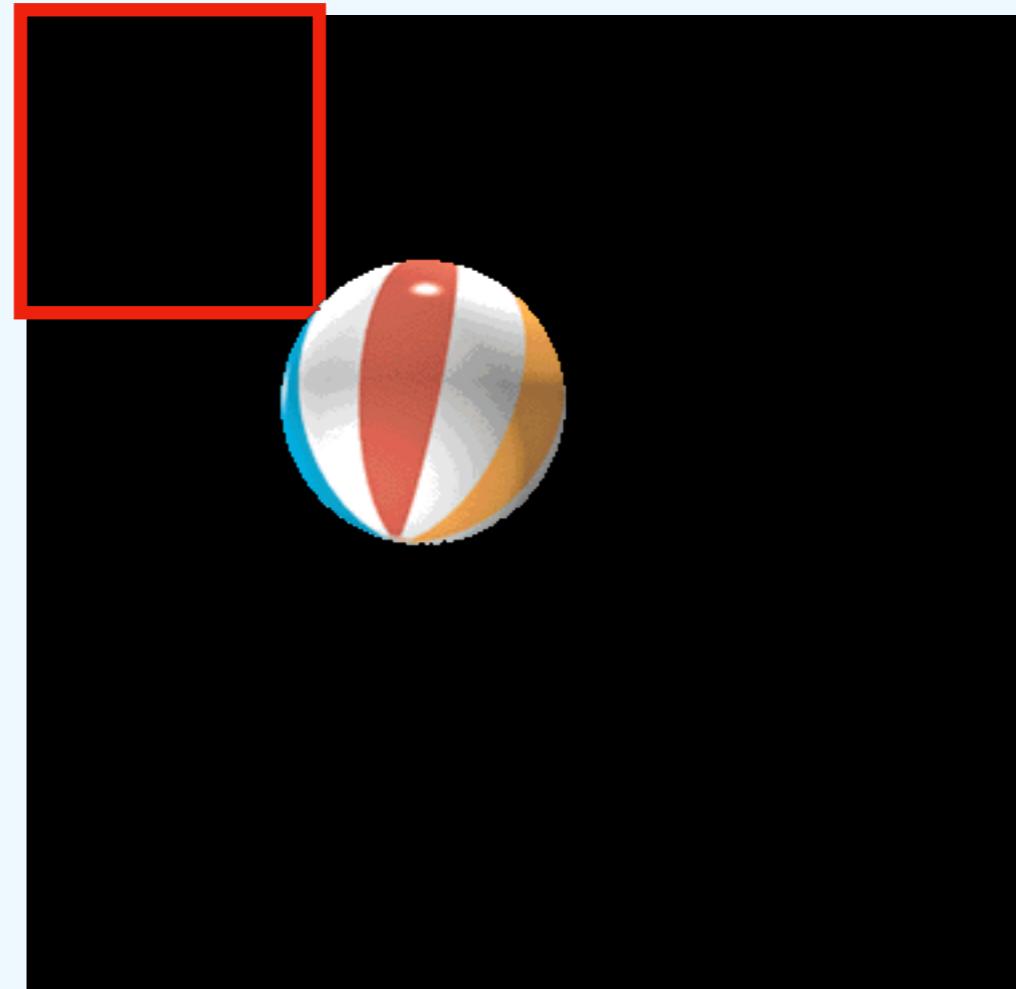
    if ballrect.left < 0 or ballrect.right > width:
        speed[0] = -speed[0]

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

    screen.fill(black)

    screen.blit(ball, ballrect)

    pygame.display.flip()
```



```
import pygame
pygame.init()

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

screen = pygame.display.set_mode((width,height))

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

ballrect = ball.get_rect()

while True:

    clock.tick(FPS)

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

    ballrect = ballrect.move(speed)

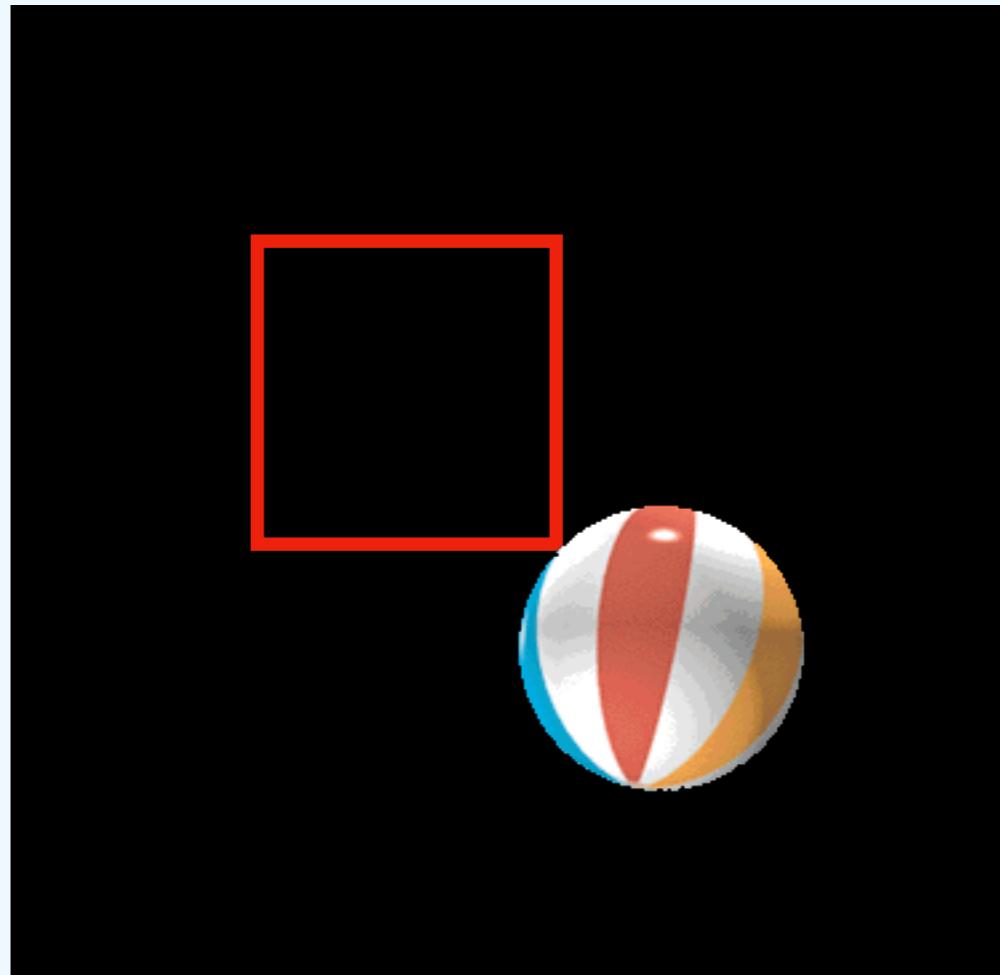
    if ballrect.left < 0 or ballrect.right > width:
        speed[0] = -speed[0]

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

    screen.fill(black)

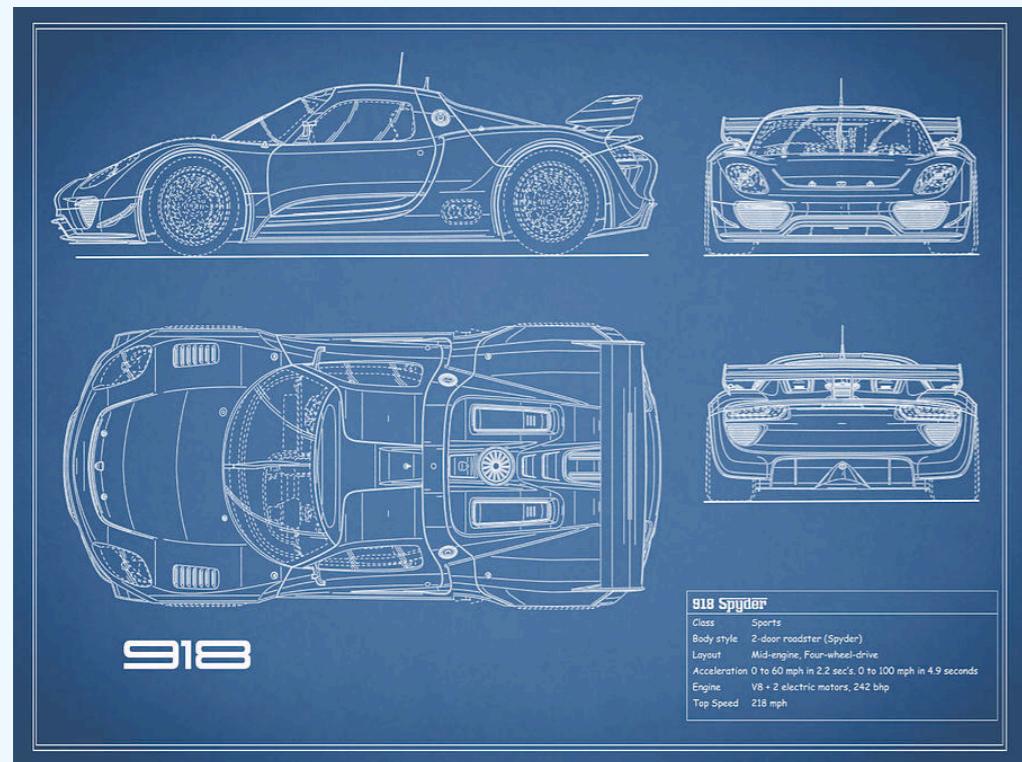
    screen.blit(ball, ballrect)

    pygame.display.flip()
```



Class

Class เปรียบเสมือนพิมพ์เขียว



class Car(pygame.sprite.Sprite):

```
def __init__(self, color, width, height):
    super().__init__()
    self.image = pygame.image.load("source/img/Cat.png")
    self.image = pygame.transform.scale(self.image,(50,38))
    self.rect = self.image.get_rect()
```

```
def moveRight(self, pixels):
    self.rect.x += pixels
```

```
def moveLeft(self, pixels):
    self.rect.x -= pixels
```

Object



Class เรียกใช้งานตรงๆ ไม่ได้ แต่ Object ต่างหาก ที่ใช้งานได้

ภาษาใน Class มีอะไรบ้าง ?

```
class Car(pygame.sprite.Sprite):
```

```
    def __init__(self, width, height):  
        super().__init__()
```

```
        self.image = pygame.image.load("source/img/Car.png")  
        self.image = pygame.transform.scale(self.image,(width,height))  
        self.rect = self.image.get_rect()
```

```
    def moveRight(self, pixels):  
        self.rect.x += pixels
```

```
    def moveLeft(self, pixels):  
        self.rect.x -= pixels
```



สร้าง Object จาก Class ได้อย่างไร ?

```
class Car(pygame.sprite.Sprite):
```

```
    def __init__(self, width, height):  
        super().__init__()
```

```
        self.image = pygame.image.load("source/img/Car.png")  
        self.image = pygame.transform.scale(self.image,(width,height))  
        self.rect = self.image.get_rect()
```

```
    def moveRight(self, d):
```

```
        self.rect.x += d
```

```
    def moveLeft(self, d):
```

```
        self.rect.x -= d
```

```
myCar = Car( 50 , 38 )
```

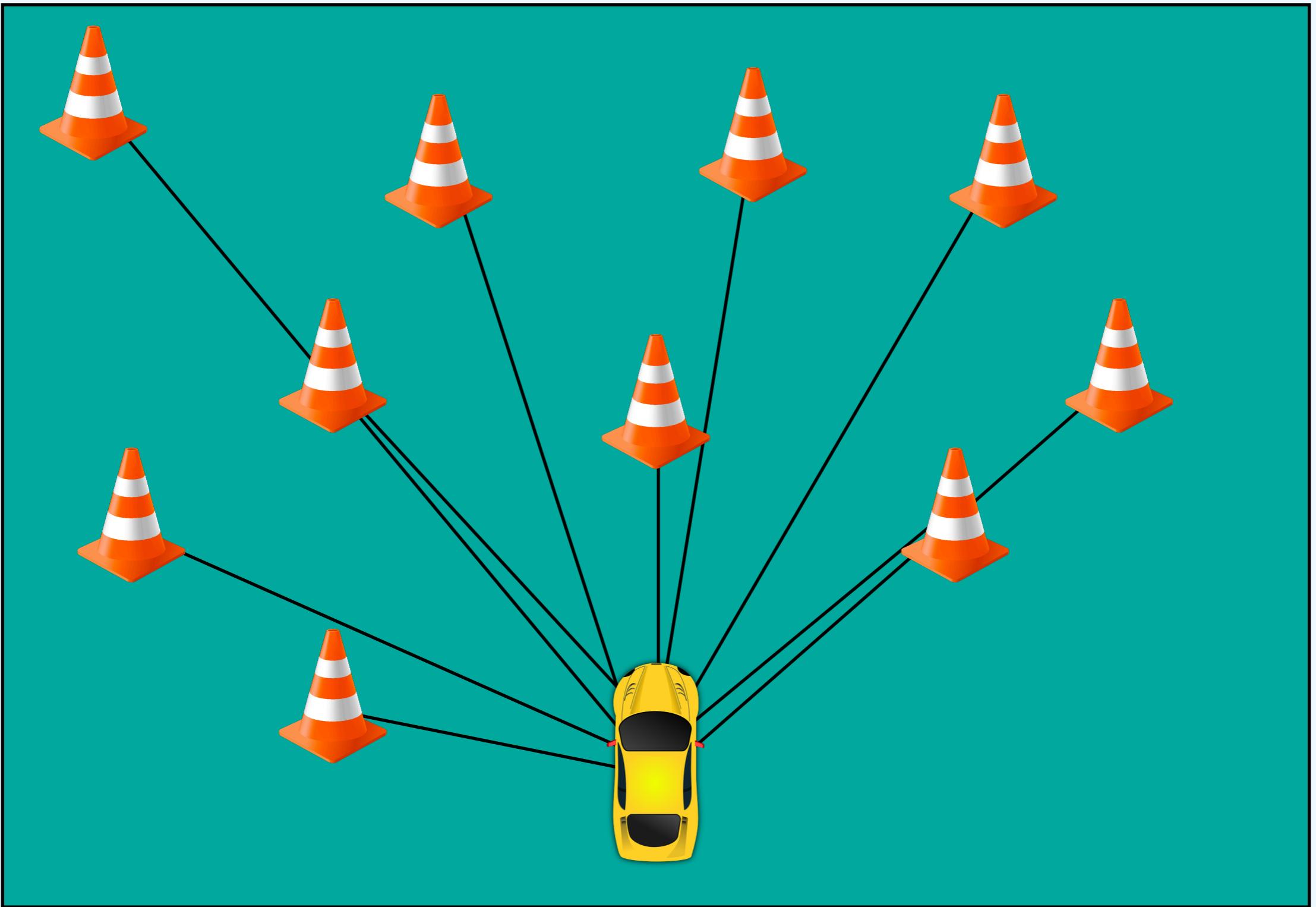
```
myCar.moveRight( 5 )
```

```
myCar.moveLeft( -3 )
```

เกม -> ต้องสามารถรับคำสั่งจากผู้เล่นได้

if  ชน  :
score += 10





รู้จักกับ Sprite (Sprite Group) ใน Pygame

Sprite เปรียบเสมือน List ที่เก็บ Object

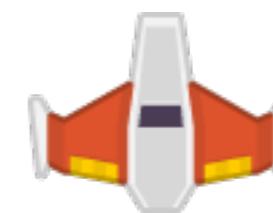
[

]

รู้จักกับ Sprite (Sprite Group) ใน Pygame

```
class Ship(pygame.sprite.Sprite):  
  
    def __init__(self, width, height):  
        super().__init__()  
        self.image = pygame.image.load("source/img/Ship.png")  
        self.image = pygame.transform.scale(self.image, (width, height))  
        self.rect = self.image.get_rect()  
  
    def moveRight(self, pixels):  
        self.rect.x += pixels  
  
    def moveLeft(self, pixels):  
        self.rect.x -= pixels
```

myShip = Ship(20, 30)



```
class Meteor(pygame.sprite.Sprite):  
  
    def __init__(self):  
        super().__init__()  
        self.image = pygame.image.load("source/img/meteor_med.png").convert_alpha()  
        self.rect = self.image.get_rect()
```

myMeteor = Meteor()



รู้จักกับ Sprite (Sprite Group) ใน Pygame

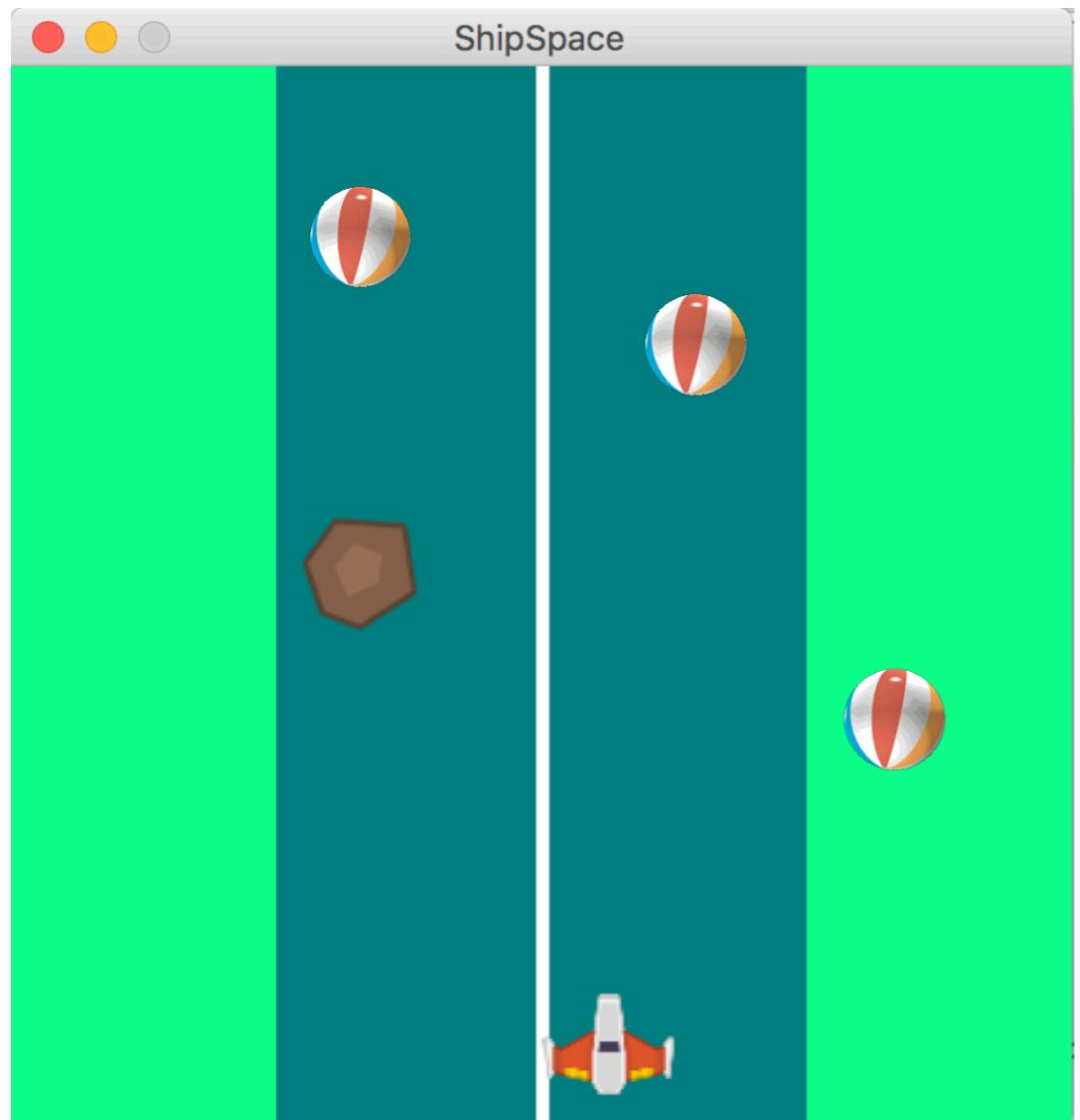
```
all_sprites = pg.sprite.Group()
```

การเพิ่ม Object ลงไปใน Sprites

```
all_sprites.add(myShip)  
all_sprites.add(myMeteor1)  
all_sprites.add(myMeteor2)  
all_sprites.add(myMeteor3)
```

```
all_sprites = [     ]
```

รู้จักกับ Sprite (Sprite Group) ใน Pygame



```
ball_sprites = pygame.sprite.Group()
```

```
[ [Ball1, Ball2, Ball3] ]
```

- Logic Control
- Graphic Control

สร้าง sprites สำหรับควบคุม Logic และ Graphic ของเกม

- ใส่ทุก object ที่มีลงไปในนี้

```
all_sprites = pygame.sprite.Group()
```

```
[ [Ship, [Meteorite, ball1, ball2, ball3], ball4] ]
```

สร้าง sprites สำหรับควบคุม Logic ของ meteor

- ใส่แยกตามประเภทของ Object

```
meteor_sprites = pygame.sprite.Group()
```

```
[ [Meteorite] ]
```

ฟังก์ชันที่ใช้จัดการกับ Sprite

`pygame.sprite.spritecollide(sprite, group, dokill)`

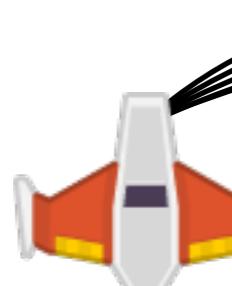
`myShip =`



`meteor_sprites = [`



`pygame.sprite.spritecollide(myShip, meteor_sprites, True)`



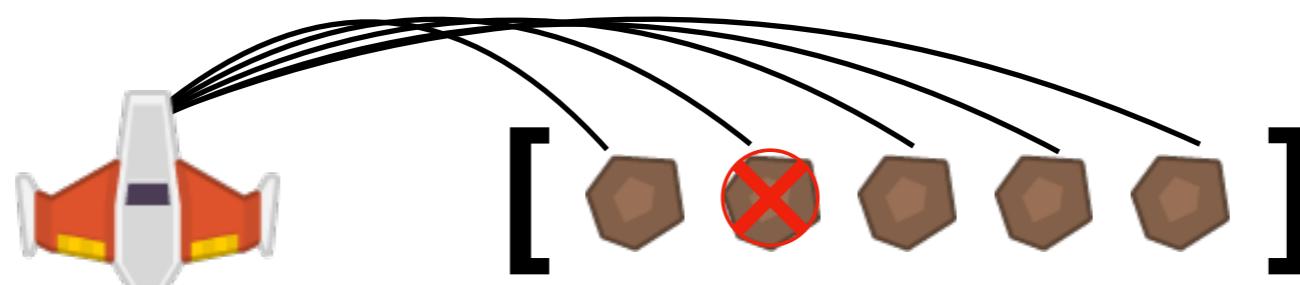
`[`



`]`

ฟังก์ชันที่ใช้จัดการกับ Sprite

```
pygame.sprite.spritecollide(myShip,meteor_sprites,True)
```



myShip =



meteor_sprites =



```
hits = pygame.sprite.spritecollide(myShip,meteor_sprites,True)
```

hits = [( , )]

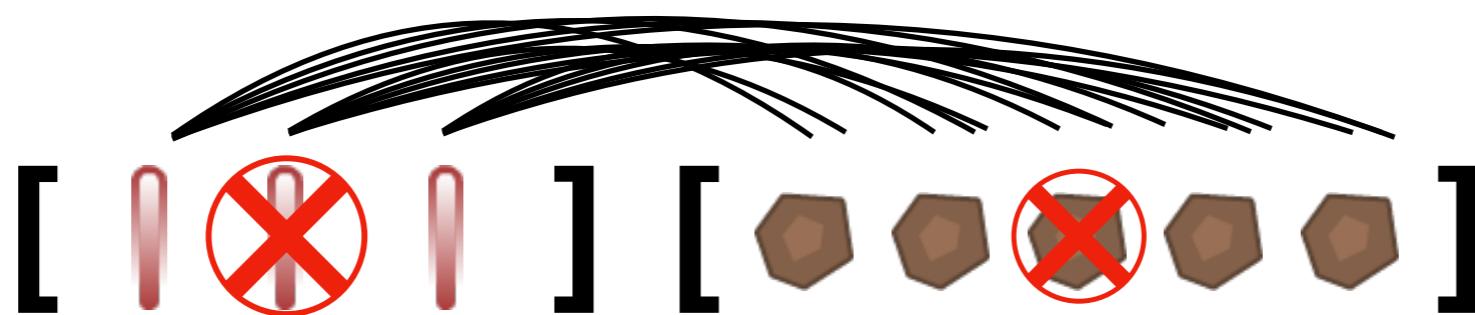
ฟังก์ชันที่ใช้จัดการกับ Sprite

```
pygame.sprite.groupcollide(group1,group2,dokill1,dokill2)
```

```
bullet_sprites = [  ]
```

```
meteor_sprites = [  ]
```

```
pygame.sprite.groupcollide(bullet_sprites,meteor_sprites,True,True)
```



03-sprite.py

04-Lab_ship_space.py

```
screen.fill(color)      # ล้างภาพหน้าจอตามสีที่กำหนด  
screen.blit(bg,bg_rect) # แพ็ค surface กับ object  
pygame.display.flip() # เอาทุก element ลงใน screen  
#####  
pygame.image.load("path to image").convert_alpha() # โหลดภาพที่เป็น .png  
pygame.transform.scale(image,(width,height)) # ปรับขนาดของภาพที่โหลดมา  
#####  
all_sprites = pygame.sprite.Group() # คำสั่งในการสร้าง spriteGroup  
pygame.sprite.spritecollide(sprite,group,dokill) # การชนกันระหว่าง sprite กับ group  
pygame.sprite.groupcollide(group1,group2,dokill1,dokill2) # การชนกันระหว่าง group1 กับ group2  
all_sprites.draw(screen) # วาด sprite ที่อยู่ใน all_sprites ลงใน screen
```