



สร้างเกม 2 มิติด้วย Pygame
(Python : Game Development)

Pygame คืออะไร



Pygame คือ ไลบรารีสำหรับ
การพัฒนาเกม 2 มิติโดยใช้
ภาษาไพธอน (Python)

ต้องมีพื้นฐานอะไรบ้าง

ต้องมีพื้นฐานอะไรบ้าง

- พื้นฐานการเขียนโปรแกรมภาษา Python
- เข้าใจโครงสร้างข้อมูลในภาษา Python (List , Dictionary)
- พื้นฐานการเขียนโปรแกรมเชิงวัตถุด้วยภาษา Python
(Class , Method , Object)

เครื่องมือ

- Python
- Visual Studio Code





ติดตั้ง Python



คำสั่งสำหรับตรวจสอบเวอร์ชัน

```
python --version
```



ติดตั้ง Visual Studio Code

ติดตั้ง Pygame

ติดตั้ง Package

```
pip install pygame #สำหรับ python 3.6 - 3.10
```

```
pip install pygame --pre #สำหรับ python 3.11
```

สร้างหน้าจอเกม



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>

สร้างหน้าจอเกมส์

```
import pygame
```

```
#เริ่มการทำงานของ Pygame
```

```
pygame.init()
```

```
#กำหนดข้อความที่ Titlebar
```

```
pygame.display.set_caption("My Game")
```

```
#สร้างหน้าจอเกม
```

```
screen = pygame.display.set_mode((ความกว้าง,ความสูง))
```

สร้างหน้าจอเกมส์

```
running=True
```

```
while running: #วนลูป While เพื่อควบคุมกิจกรรมในการ์นเกม
```

```
    for event in pygame.event.get(): #ตรวจสอบ event ระหว่างรันเกม
```

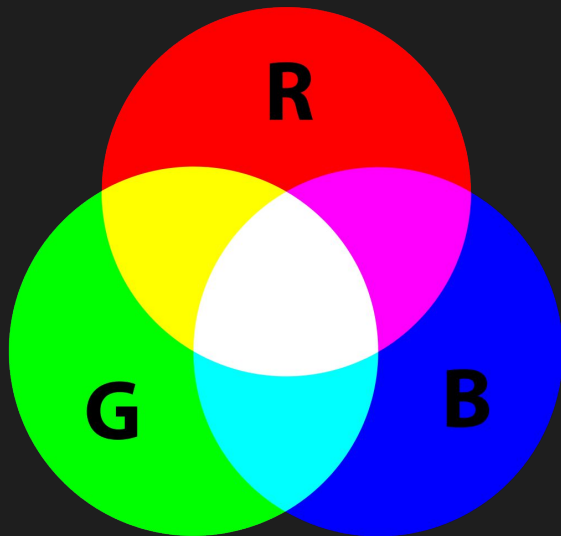
```
        if event.type == pygame.QUIT: #ถ้าเกิด event ปิดหน้าจอเกม
```

```
            running=False #หยุดการวนลูป While
```

```
pygame.quit() #ออกจากเกม
```

กำหนดสี (Color)

การกำหนดสี



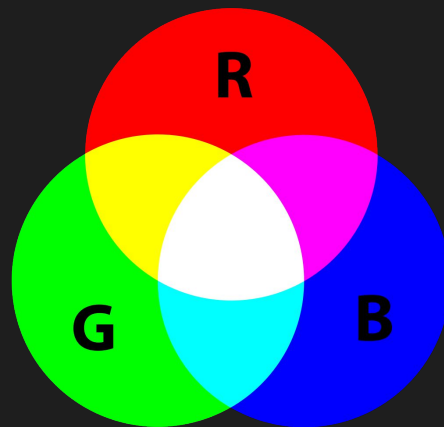
การกำหนดสีใน pygame จะใช้
ระบบสีแบบ RGB เป็นรูปแบบ Tuple
โดยค่า RGB คือระดับสีแดง (RED) ,
สีเขียว (GREEN) , สีน้ำเงิน (BLUE)

โครงสร้างคำสั่ง

ชื่อตัวแปร = (R , G , B)

ตัวอย่าง

- BLACK = (0 , 0 , 0)
- WHITE = (255 , 255 , 255)
- GREEN = (0 , 255 , 0)
- RED = (255 , 0 , 0)



ตั้งค่าสีหน้าจอ

```
WHITE = (255,255,255) #กำหนดสี
```

```
#สร้างหน้าจอ
```

```
screen = pygame.display.set_mode((WIDTH,HEIGHT))
```

```
screen.fill(WHITE) #แสดงสีพื้นหลังในหน้าจอ
```

```
pygame.display.update() #อัปเดตการแสดงผลไปที่หน้าจอ
```

การวาดรูปทรงต่างๆ

ระบบพิกัดในคอมพิวเตอร์กราฟิก

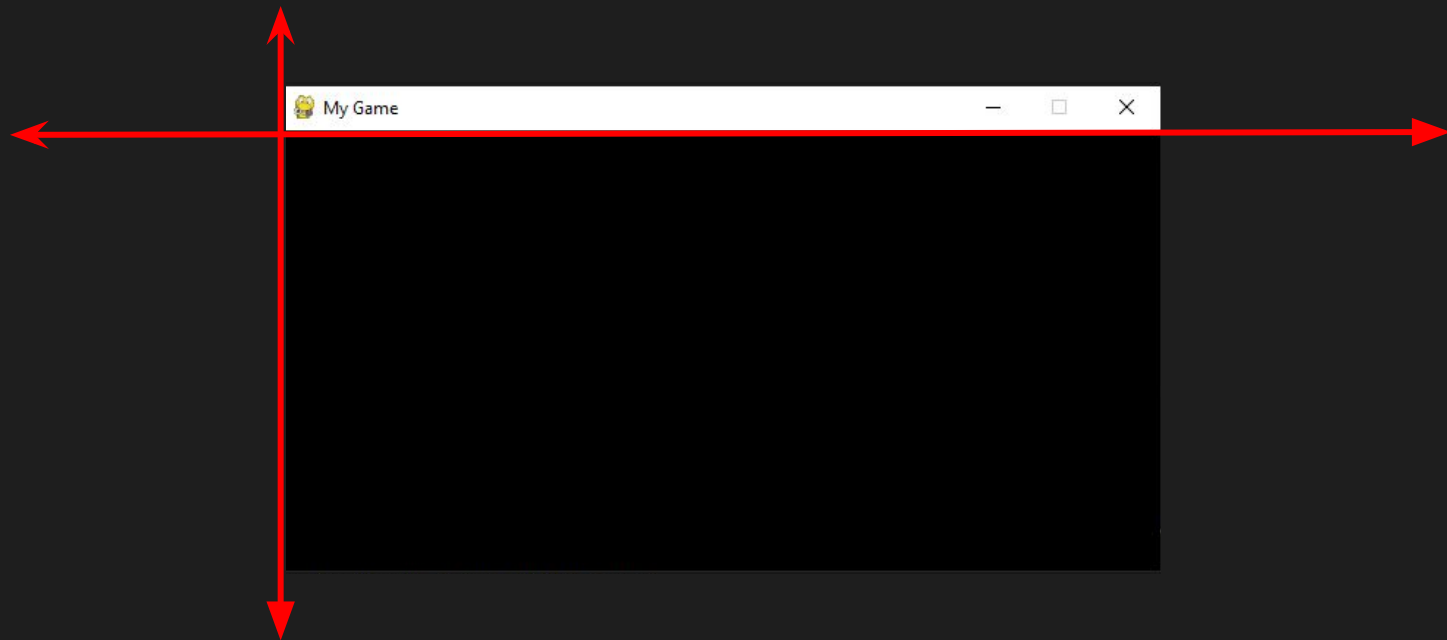


<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>

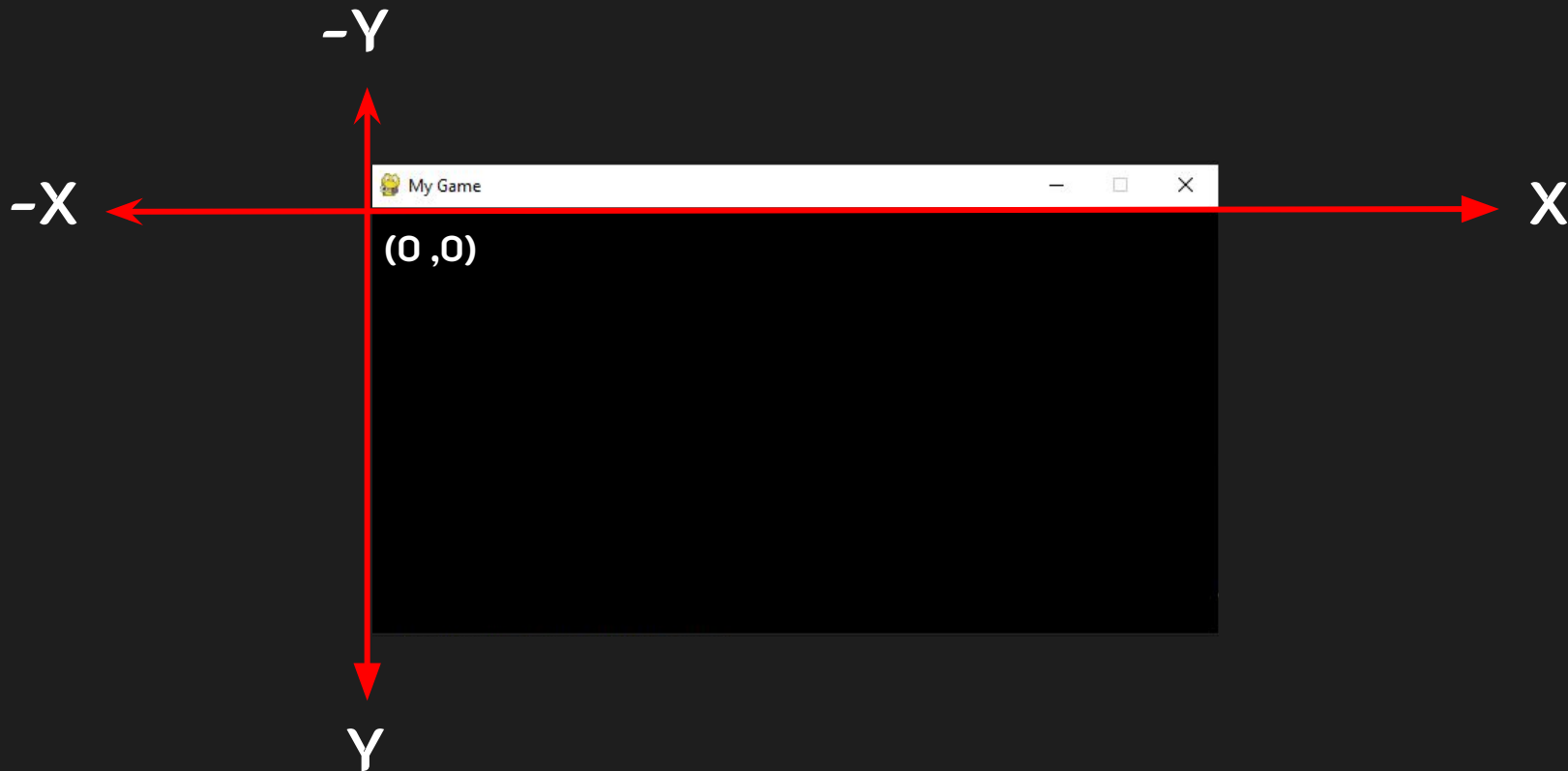
ระบบพิกัดในคอมพิวเตอร์กราฟิก



ระบบพิกัดในคอมพิวเตอร์กราฟิก



ระบบพิกัดในคอมพิวเตอร์กราฟิก



ระบบพิกัดในคอมพิวเตอร์กราฟิก



การวาดเส้นตรง

โครงสร้างคำสั่ง

`pygame.draw.line(surface,color ,start,end ,width)`

- `surface` คือ หน้าจอที่ต้องการวาดรูปทรง
- `color` คือ สีของเส้นตรง
- `start , end` คือ จุดเริ่มต้น และ จุดสิ้นสุดการวาดเส้นตรง
- `width` คือ ขนาดของเส้นตรง (ค่าเริ่มต้น = 1)

การวาดรูปสี่เหลี่ยม

โครงสร้างคำสั่ง

`pygame.draw.rect(surface,color,(x, y , width , height))`

- `surface` คือ หน้าจอที่ต้องการวาดรูปทรง
- `color` คือ สีพื้นหลังของสี่เหลี่ยม
- `x , y` คือ ตำแหน่งมุมบนซ้ายของสี่เหลี่ยม
- `width , height` คือ ความกว้าง และ ความสูง

การวาดวงกลม

โครงสร้างคำสั่ง

`pygame.draw.circle(surface,color, center ,radius,width)`

- `surface` คือ หน้าจอที่ต้องการวาดรูปทรง
- `color` คือ สีของวงกลม หรือ ขอบวงกลม
- `center` คือ จุดศูนย์กลางของวงกลม (`x` , `y`)
- `radius` คือ รัศมีหรือขนาดวงกลม
- `width` คือ ความหนาของเส้นขอบ

การแสดงข้อความ



<https://www.youtube.com/c/KongRuksiamOfficial/>



<https://www.facebook.com/KongRuksiamTutorial/>

แสดงข้อมูลฟอนต์ (System Font)

```
fonts = pygame.font.get_fonts()
```

```
for font in fonts:
```

```
    print(font)
```



ตั้งค่าฟอนต์และข้อความ

```
sys_font = pygame.font.SysFont(name,size)
```

```
message_text = sys_font.render(text,antialias,color)
```

- name คือ ชื่อฟอนต์
- size คือ ขนาดฟอนต์
- text คือ ข้อความ
- antialias คือ ลบรอยหยัก (True / False)
- color คือ สีข้อความ

แสดงข้อความ

```
sys_font = pygame.font.SysFont(name,size)
```

```
message_text = sys_font.render(text,antialias,color)
```

```
screen.blit(message_text , dest)
```

- **dest** คือ ตำแหน่งพิกัดมุมบนซ้ายของข้อความ (x , y)

กำหนดฟอนต์ข้อความ (Custom Font)

ตั้งค่าฟอนต์และข้อความ

```
custom_font = pygame.font.Font(name.ttf,size)
```

```
message_text = custom_font.render(text,antialias,color)
```

- name คือ ชื่อฟอนต์
- size คือ ขนาดฟอนต์
- text คือ ข้อความ
- antialias คือ ลบรอยหยัก (True / False)
- color คือ สีข้อความ

แสดงข้อความ

```
custom_font = pygame.font.Font(name.ttf,size)
```

```
message_text = custom_font.render(text,antialias,color)
```

```
screen.blit(message_text , dest)
```

- **dest** คือ ตำแหน่งพิกัดมุมบนซ้ายของข้อความ (x , y)

การใช้งานรูปภาพ (Image)

การใช้งานรูปภาพ

นำภาพเข้ามาทำงาน

- `img = pygame.image.load("ตำแหน่งภาพ")`

ปรับขนาดภาพใหม่

- `img = pygame.transform.scale(img,(width,height))`

แสดงผลภาพ

- `screen.blit(img,(x,y))`

รู้จักกับ `getRect()`

รู้จักกับ `getRect()`

`get_rect()` คือ คำสั่งที่ใช้ดึงข้อมูลพื้นฐานจากวัตถุที่เราสนใจ เช่น ความกว้าง , ความสูง , ตำแหน่งกึ่งกลาง

```
SCREEN_W = 600
```

```
SCREEN_H = 300
```

```
screen = pygame.display.set_mode((SCREEN_W,SCREEN_H))
```

```
screen_rect = screen.get_rect() #เก็บข้อมูลหน้าจอ
```

รู้จักกับ `getRect()`

ค่าพื้นฐานที่อยู่ในวัตถุ

- `width` คือ ความกว้างของวัตถุ (`SCREEN_W`)
- `height` คือ ความสูงของวัตถุ (`SCREEN_H`)
- `size` คือ ขนาดวัตถุ (`SCREEN_W` , `SCREEN_H`)
- `centerx` คือ จุดกึ่งกลางแกน x (`SCREEN_W // 2`)
- `centery` คือ จุดกึ่งกลางแกน y (`SCREEN_H // 2`)
- `center` คือ จุดกึ่งกลาง (`SCREEN_W // 2` , `SCREEN_H // 2`)

รับค่าจากเมาส์ (Mouse)

รับค่าจากเมาส์ (Mouse)

โครงสร้างคำสั่ง

- `MOUSEBUTTONDOWN` คือ event การกดเมาส์
- `event.pos[0]` คือ ตำแหน่งเมาส์ในพิกัดแกน x
- `event.pos[1]` คือ ตำแหน่งเมาส์ในพิกัดแกน y

รับค่าจากคีย์บอร์ด

รับค่าจากคีย์บอร์ด (Keyboard)

โครงสร้างคำสั่ง

- `pygame.key.get_pressed()` คือ คำสั่งสำหรับดึงรายการปุ่มบนคีย์บอร์ดที่มีการกดปุ่มทั้งหมดมาใช้งาน
- `pygame.K_ชื่อปุ่ม` เช่น
 - `pygame.K_LEFT` คือ ปุ่มลูกศรซ้ายมือ
 - `pygame.K_RIGHT` คือ ปุ่มลูกศรขวามือ

กำหนด Frame Rate

กำหนด Frame Rate

Frame Rate คือ จำนวนเฟรมที่จะแสดงผลใน 1 วินาที (Frame Per Second) หรือ ความเร็วในการแสดงหรืออัปเดตหน้าจอเกม

ถ้ากำหนดเป็น 30 หมายถึง จะมีการอัปเดตหน้าจอจำนวน 30 ครั้งใน 1 วินาที (30 FPS)

กำหนด Frame Rate

ใน pygame การกำหนด Frame Rate สามารถดำเนินการผ่านคลาส `Clock()` โดยใช้คำสั่ง

- `clock = pygame.time.Clock()`
- `clock.tick(FPS)`

การชนของวัตถุ

การชนของวัตถุ

การชน (Collision) เป็นระบบพื้นฐานของเกม อาจจะ
ใช้การชนเพื่อเก็บคะแนนในเกม หรือ เพื่อตัดสินแพ้-ชนะ
ในเกม ขึ้นอยู่กับการออกแบบระบบเกมนั้นๆ

การชนนั้นจะเป็นการตรวจสอบผิวสัมผัสของวัตถุว่า
มีการซ้อนทับ หรือ สัมผัสกันหรือไม่

คอร์สเรียนบน Udemy

วิทยากร

Kong Ruksiam

Programmer , Developer


ผู้เรียนทั้งหมด รีวิว

1,912 **450**

เกี่ยวกับฉัน

โปรแกรมเมอร์และนักพัฒนาเกม รวมถึงสอนเกี่ยวกับการเขียนโปรแกรมในช่องยูทูป KongRuksiam , KongRuksiam Official และเป็นเจ้าของแฟนเพจ KongRuksiam มีผู้ติดตามมากกว่า 50,000 คน



 Facebook

 Youtube

<https://www.udemy.com/user/kong-ruksiam/>