

Lecture 17 Graphics กราฟิก

17 ตุลาคม 2557 11:05

เราได้เรียนพื้นฐานการเขียนโปรแกรมตั้งแต่ input output ตัวแปร variable , loops, lists , functions, objects, และ modules ต่อไปจะเริ่มเรียนสิ่งที่จะทำให้สนุกคือการสร้างเกม ทุกคนเมื่อเล่นเกมจะมีความสุข
วันนี้เราจะเรียน พื้นฐานการวาด(draw)บนหน้าจอ (screen) วาดรูปร่าง (shape) สี และสร้างสิ่งให้เคลื่อนไหว (animation) ซึ่งจะใช้เป็นพื้นฐานในการสร้างเกมต่อไป

Pygame



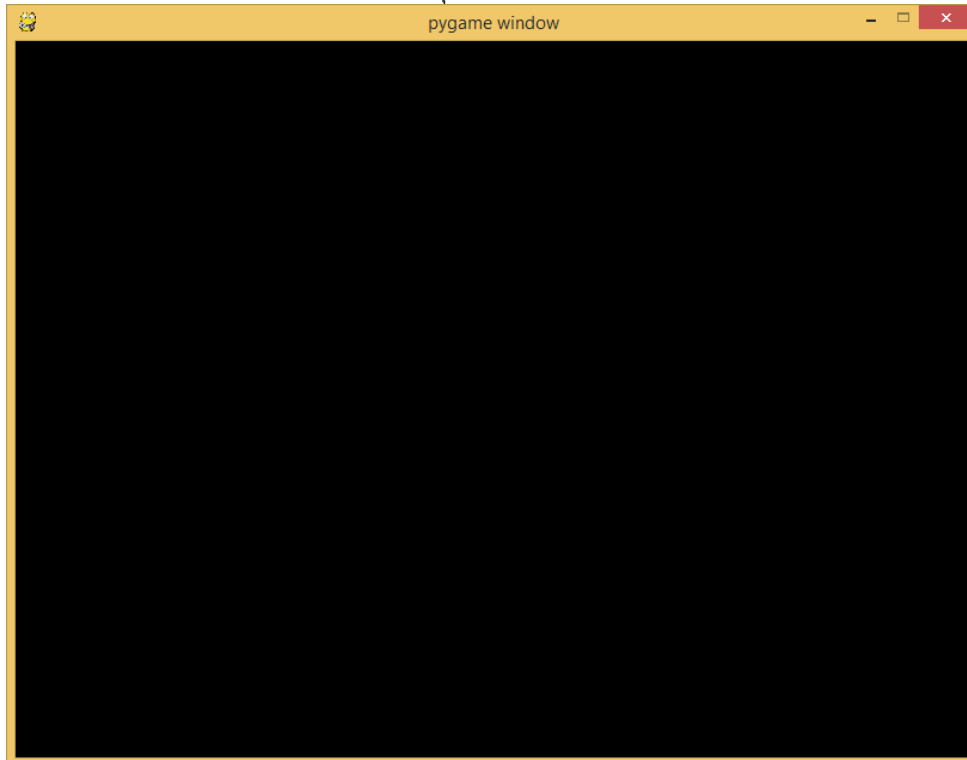
เป็นโมดูลที่จะช่วยเราเขียนเกมส์ ถ้าเราเขียนโปรแกรมให้แสดงผลกราฟฟิคเอง โปรแกรมจะค่อนข้างซับซ้อนเพราะจะต้องติดต่อกับเครื่องคอมพิวเตอร์โดยตรง เป็นการดีที่มีตัวช่วยอย่าง pygame ค่อยอำนวยความสะดวกให้เรา แทนที่เราจะเขียนโค้ดหลายสิบบรรทัดเอง เรากลับเรียกใช้ฟังก์ชันใน pygame ไม่กี่บรรทัดในการแสดงผล เราสามารถดูรายละเอียด pygame เพิ่มเติมได้ใน เว็บ www.pygame.com สามารถย้อนกลับไปได้ดูเนื้อหาการติดตั้ง pygame ร่วมกับ python ได้ที่ lecture 2

เริ่มที่การสร้างหน้าต่าง A pygame window

เราจะสร้างหน้าต่าง (screen) ขนาด กว้าง 800 ยาว 600 ได้ด้วยโค้ดนี้

```
import pygame
pygame.init()
screen = pygame.display.set_mode([800, 600])
```

โปรแกรมจะขึ้นหน้าจอ ขนาดเท่ากับที่เราระบุ



แต่ตอนนี้ โปรแกรมจะไม่สามารถปิดเองได้ ถ้าเราต้องการให้ปิดเองได้ให้ใช้คำสั่ง ที่ shell

```
>>> pygame.quit()
```

ถ้าเราใส่ pygame.quit() ไปทันทีโปรแกรมจะปิดทันที เราควรจะวนเพื่อรอดูว่า มีการกดปุ่มปิดก่อนหรือไม่ แล้ววนดูเหตุการณ์ต่างๆ ที่เกิดขึ้นในเกมส์ (event) ถ้ามี event ไหนที่เป็นชนิด (type) ปิด QUIT ให้หยุด ดังโค้ดนี้

```
import pygame
pygame.init()
screen = pygame.display.set_mode([800, 600])

running = True
while running:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False

pygame.quit()
```

การวาดด้านใน Drawing in the window

เราจะลองเพิ่มโค้ด เพื่อวาดบางสิ่ง ให้สีเป็นสีขาว (Red, Green, Blue) RGB ถ้าสีขาวจะเป็น [255, 255, 255] fill คือการเติมสี

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

วาดวงกลม circle ลง screen สีแดง [255, 0, 0] ตำแหน่ง (x, y) = [100, 100] ขนาด 30

```
pygame.draw.circle(screen, [255, 0, 0], [100, 100], 30, 0)
```

การวาดกราฟฟิค จะไม่นิยามวาดสิ่งของแล้วแสดงผลทีละอย่าง แต่จะวาดจำนวนวัตถุเยอะ ก่อนแล้วค่อยแสดงผล เพื่อเป็นการประหยัดเวลา และเพิ่มความเร็ว ดังนั้น pygame จะมีคำสั่ง flip() เมื่อวาดหลายวัตถุแล้ว พร้อมทั้งจะแสดงผลแล้วจริง เรียกคำสั่งด้านล่างนี้

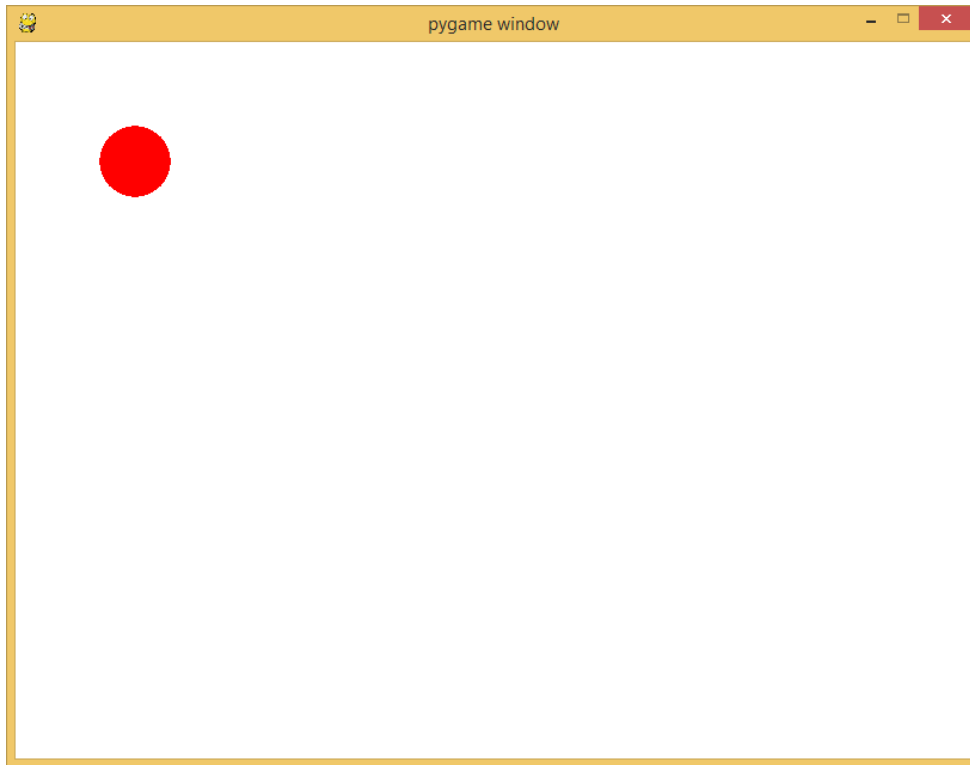
```
pygame.display.flip()
```

เมื่อรวมกันเป็นโค้ดจะได้ดังนี้

```
import pygame
pygame.init()
screen = pygame.display.set_mode([800, 600])
screen.fill([255, 255, 255])
pygame.draw.circle(screen, [255, 0, 0], [100, 100], 30, 0)
pygame.display.flip()
running = True
while running:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False

pygame.quit()
```

เมื่อแสดงผลจะได้ดังนี้



การวาดวงกลมจะระบุอยู่ 5 สิ่งด้วยกัน pygame.draw.circle()

- พื้นที่ที่ต้องการวาด ที่นี้เราเลือก screen
- สีระบุเป็น (R, G, B) สีแดง ให้เติมสี 255 ส่วนเขียว 0 และน้ำเงิน 0 ดังนั้นจะได้ [255, 0, 0]
- ตำแหน่ง x y บนหน้าจอ x = 100, y=100 จะต้องให้ค่า [100, 100]
- ขนาด size ของวงกลม รัศมี 30
- ขนาดของเส้นขอบ width = 0 ไม่วาดเส้น

x, y ตำแหน่งที่เราอ้างอิงบนหน้าจอ เราเรียกว่าหนึ่งจุดบนหน้าจอ (pixel)

Colors in python

เราสามารถที่ใช้สีจากชื่อได้โดย นำเข้า THECOLORS

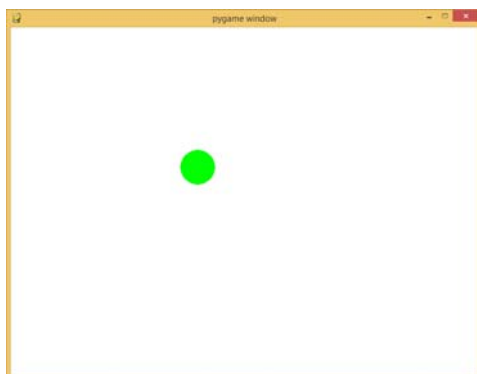
```
from pygame.color import THECOLORS
```

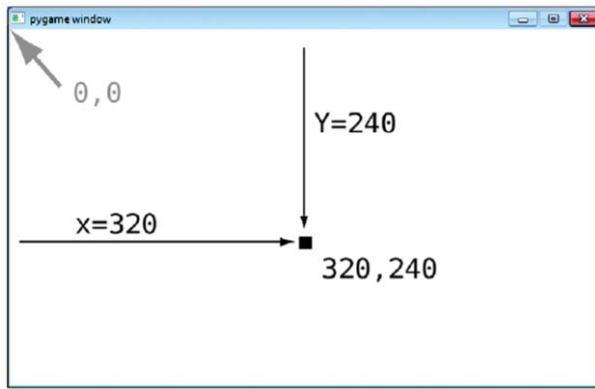
แล้วอ้างอิงสีจากชื่อดังนี้

```
screen.fill(THECOLORS["white"])
pygame.draw.circle(screen, THECOLORS["green"], [100, 100], 30, 0)
```

ตำแหน่งบนหน้าจอ Location - Screen coordinates

ถ้าเปลี่ยนตำแหน่งจาก 100,100 เป็น [320, 240]





ด้วยโค้ด

```
pygame.draw.circle(screen, THECOLORS["green"], [320, 240], 30, 0)
```

ขนาดของวัตถุ Size of Shapes

การวาดวงกลมเราจะกำหนดขนาดของวงกลมเป็น 30 คือรัศมี แต่รูปที่สามารถวาดได้ใน pygame อาจจะเป็น สีเหลี่ยม rectangle ย่อในโค้ดเป็น rect ในการวาดเราสามารถระบุ ด้านซ้ายทาง left ด้านบนทาง top ความกว้าง width ความยาว height ดังโค้ดรูปแบบนี้

```
Rect(left, top, width, height)
```

เราจะสร้างวัตถุสีเหลี่ยมของเราเองได้ดังนี้

```
my_rect = Rect(250, 150, 300, 200)
```

โค้ดที่ใช้วาดสีเหลี่ยมที่มีสีน้ำเงิน [0, 0, 255] มีดังนี้

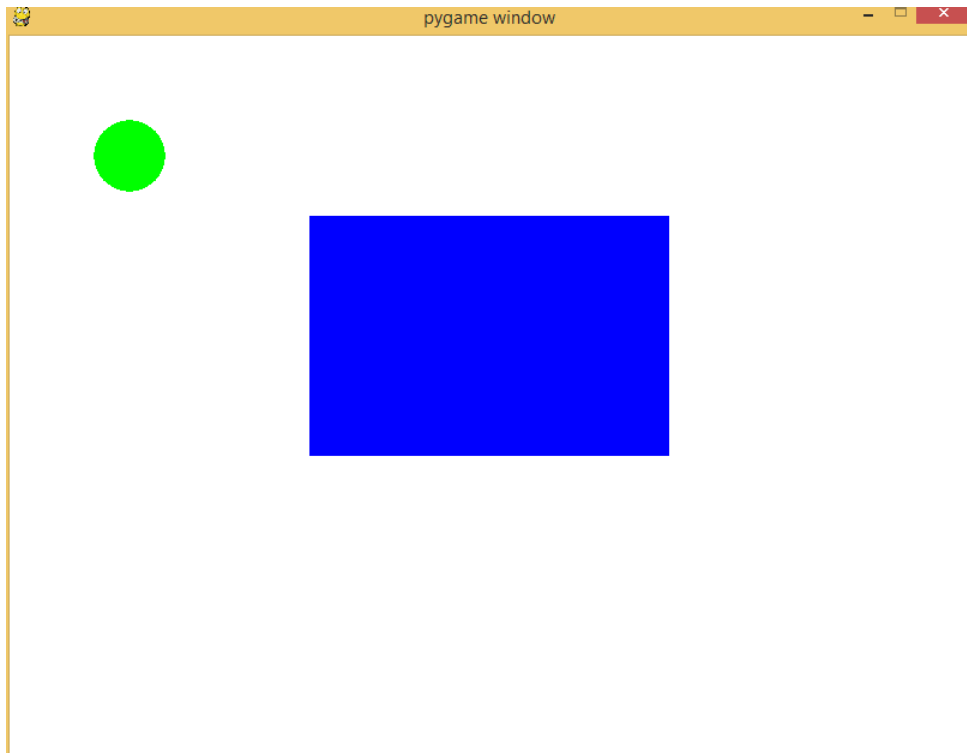
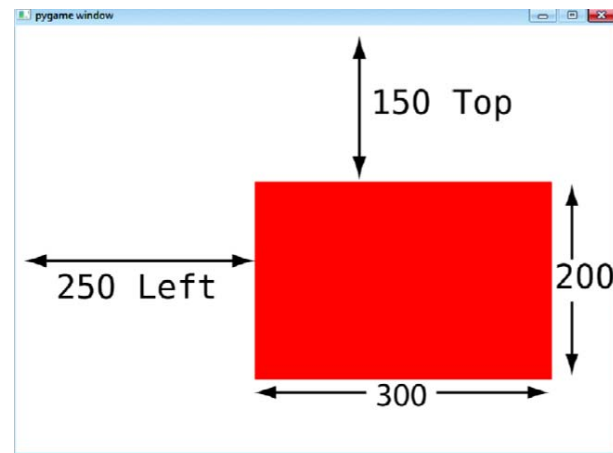
```
pygame.draw.rect(screen, [0, 0, 255], [250, 150, 300, 200], 2)
```

หรือแยกออกเป็นชื่อดัชนี list ของสีเหลี่ยมได้

```
my_list = [250, 150, 300, 200]
pygame.draw.rect(screen, [0, 0, 255], my_list, 2)
```

หรือสร้างวัตถุสีเหลี่ยมดังนี้

```
my_rec = pygame.Rect(250, 150, 300, 200)
pygame.draw.rect(screen, [0, 0, 255], my_rec, 2)
```



ลองโค้ดต่อไปนี้

```

import pygame
import random
from pygame.color import THECOLORS

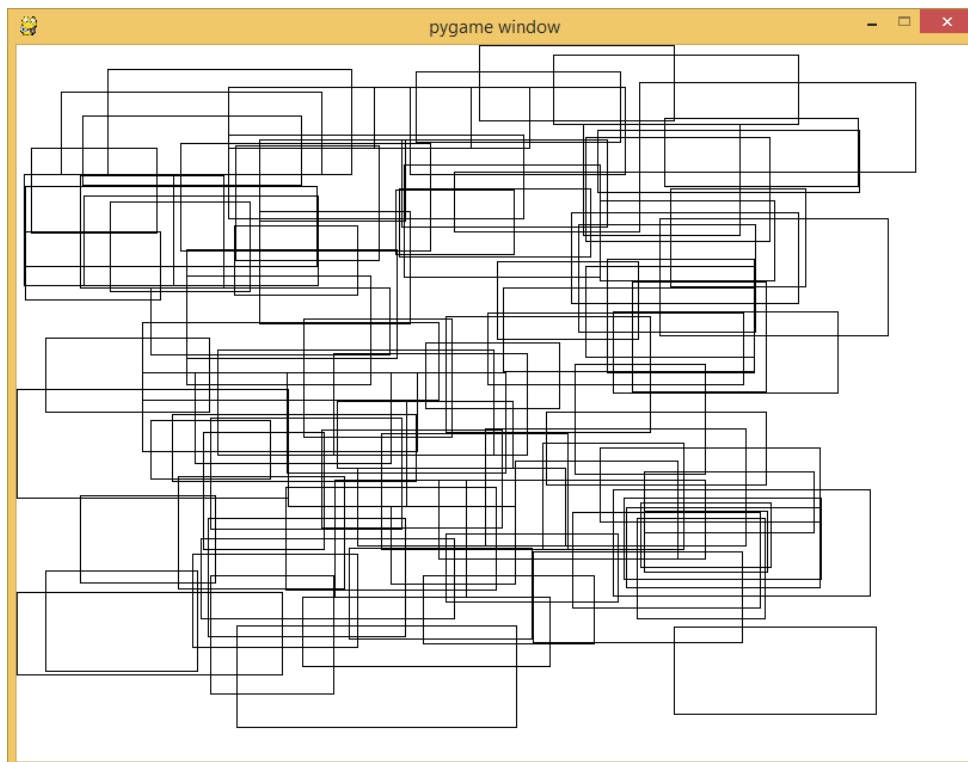
pygame.init()
screen = pygame.display.set_mode([800, 600])
screen.fill(THECOLORS["white"])

for i in range(100):
    width = random.randint(100, 250)
    height = random.randint(50, 100)
    top = random.randint(0, 500)
    left = random.randint(0, 550)
    my_rec = pygame.Rect(left, top, width, height)
    pygame.draw.rect(screen, [0,0,0], my_rec, 1)
pygame.display.flip()
running = True
while running:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False

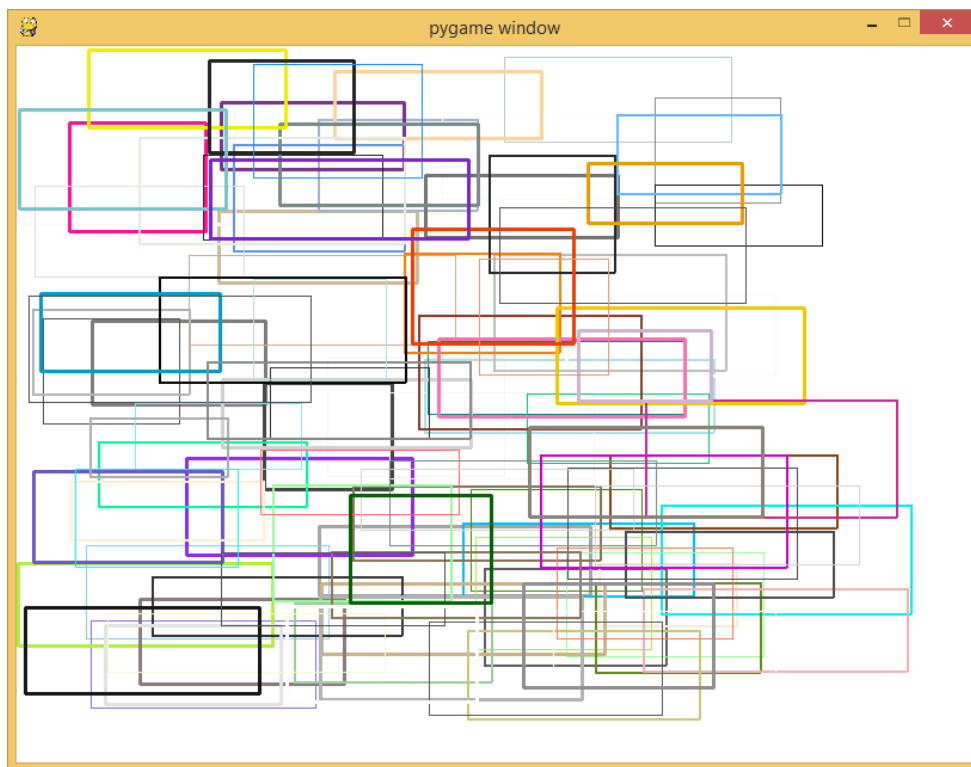
pygame.quit()

```

แสดงผลรัน



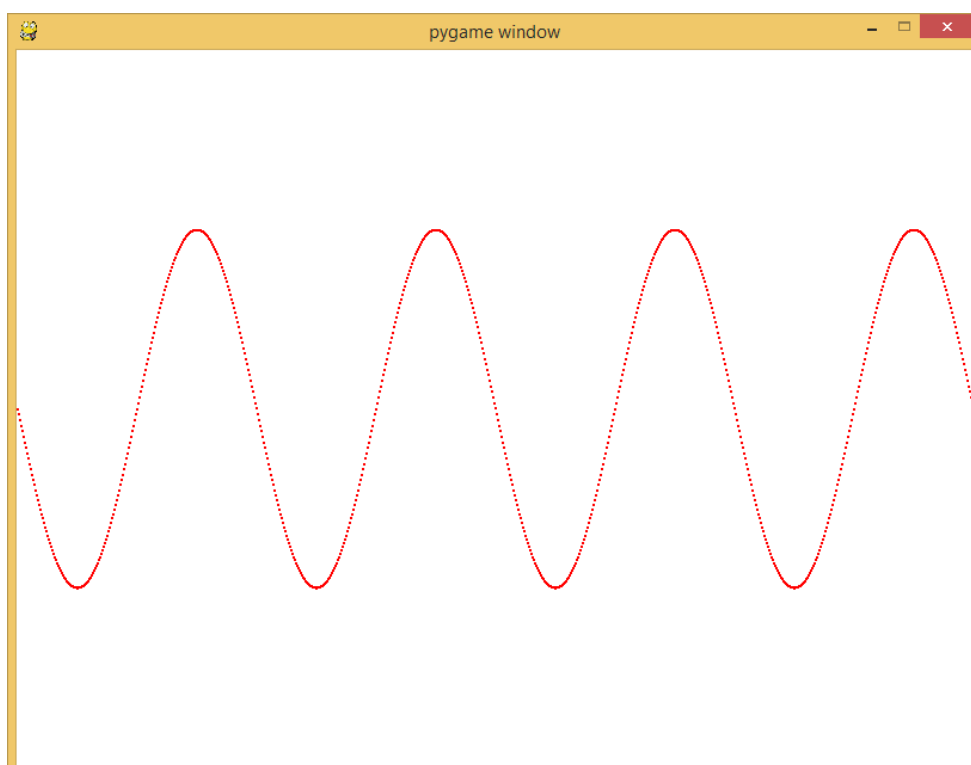
เราจะทำให้มีสีเส้นโดยแก้ไขโค้ดเดิม



โดยโค้ด

```
colors = list(THECOLORS.keys())
for i in range(100):
    width = random.randint(100, 250)
    height = random.randint(50, 100)
    top = random.randint(0, 500)
    left = random.randint(0, 550)
    color_name = random.choice(colors)
    color = THECOLORS[color_name]
    line_width = random.randint(1,3)
    my_rec = pygame.Rect(left, top, width, height)
    pygame.draw.rect(screen, color, my_rec, line_width)
```

การวาดทีละ pixel



เราจะวาดสมการ $y = \sin(x)$ y อยู่ในช่วง $(-1, 1)$ x อยู่ในช่วง $(0, 2\pi)$
เขียนเป็นโค้ดได้ดังนี้

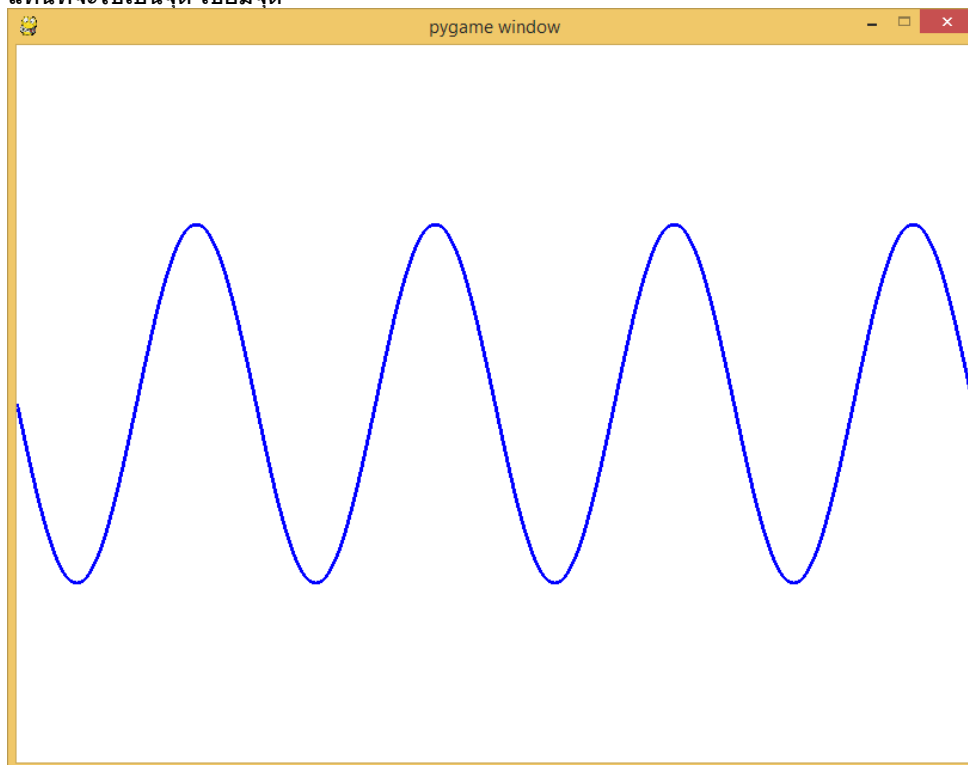
```
import pygame
import random
from pygame.color import THECOLORS
import math

screen_width = 800
screen_height = 600
pygame.init()
screen = pygame.display.set_mode([screen_width, screen_height])
screen.fill(THECOLORS["white"])

n = 4
for x in range(0, screen_width):
    angle = x/screen_width * n * math.pi * 2
    y = int(math.sin(angle)*screen_height/4 + screen_height/2)
    pygame.draw.rect(screen, THECOLORS["red"], [x, y, 2, 2], 0)
pygame.display.flip()
running = True
while running:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False

pygame.quit()
```

แทนที่จะใช้เป็นการจุด เชื่อมจุด



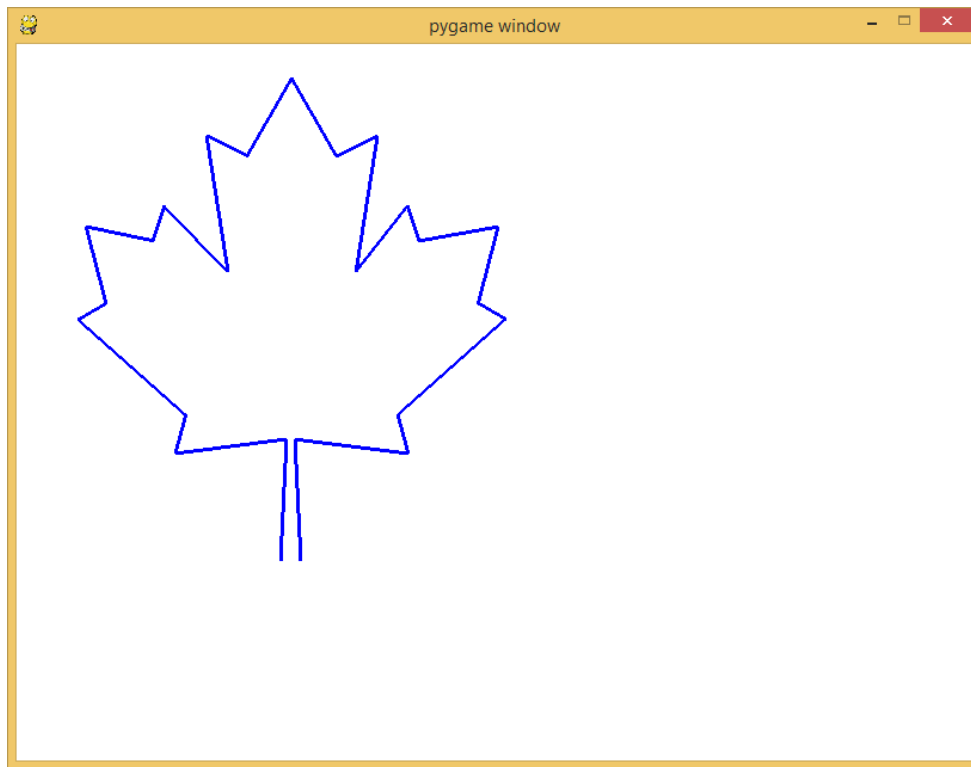
โดยใช้ การวาดเส้นจาก list ของจุด

```
pygame.draw.lines(screen, THECOLORS["blue"], False, plotPoints, 3)
```

แก้ไขโค้ดและเพิ่มเติมโค้ดดังนี้

ลองเชื่อมจุดเหล่านี้

```
dots = [ [221, 432], [225, 331], [133, 342], [141, 310],
          [51, 230], [74, 217], [58, 153], [114, 164],
          [123, 135], [176, 190], [159, 77], [193, 93],
          [230, 28], [267, 93], [301, 77], [284, 190],
          [327, 135], [336, 164], [402, 153], [386, 217],
          [409, 230], [319, 310], [327, 342], [233, 331],
          [237, 432]]
```



การวาดสีที่จุด

เราสามารถที่จะอ่านค่าสีจากหน้าจอ ด้วยคำสั่งที่ตำแหน่ง x,y ใดๆได้

```
color = screen.get_at([x, y])
```

และสามารถให้ค่าสีที่ตำแหน่งใดๆ [x, y] และสี

```
screen.set_at([x, y], [r, g, b])
```

จงดูผลลัพธ์โค้ดนี้

```
print(screen.get_at([0, 0]))
screen.set_at([221, 431], [255, 0, 0])
screen.set_at([221, 432], [255, 0, 0])
screen.set_at([222, 431], [255, 0, 0])
screen.set_at([222, 432], [255, 0, 0])
pygame.display.flip()
```

การใส่รูปภาพ

นอกจากเราจะวาดด้วยการวาดแต่ละ pixel วงกลม สีเหลี่ยมและเส้นแล้ว เรายังแสดงผลโดยใช้รูปภาพ ด้วยฟังก์ชัน image ดังโค้ดต่อไปนี้


```

import pygame
import sys
from pygame.color import THECOLORS

screen_width = 800
screen_height = 600
pygame.init()
screen = pygame.display.set_mode([screen_width, screen_height])
screen.fill(THECOLORS["green"])

myBrid = pygame.image.load("angry.png")
screen.blit(myBrid, [50, 50])

pygame.display.flip()
running = True
while running:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False

pygame.quit()

```

เมื่อรันจะได้รูปดังนี้



ถ้าหากขนาดใหญ่เกินไป เราสามารถย่อได้โดยระบุขนาดใหม่ โดยใช้คำสั่ง `pygame.transform.scale()` แล้วระบุขนาดใหม่

```

myBrid = pygame.image.load("angry.png")
myBrid = pygame.transform.scale(myBrid, [100, 100])
screen.blit(myBrid, [50, 50])

```

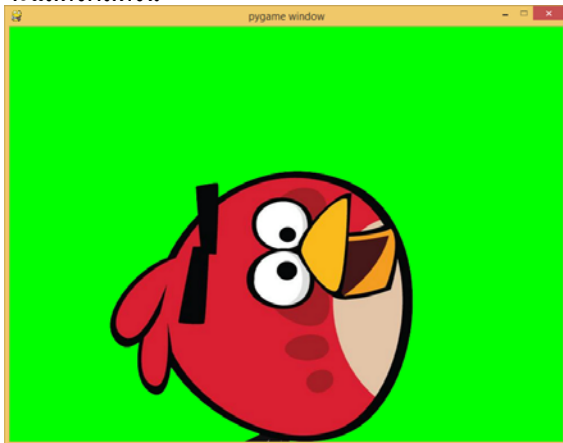
แสดงผลได้ดังนี้



หรือจะหมุนด้วยคำสั่ง `pygame.transform.rotate()` และระบุภาพ และ องศาเข้าไป

```
myBrid = pygame.image.load("angry.png")
myBrid = pygame.transform.rotate(myBrid, 90)
screen.blit(myBrid, [50, 50])
```

จะแสดงผลดังนี้



การวาดหลายภาพ

```
myBrid = pygame.image.load("angry.png")
myBrid = pygame.transform.scale(myBrid, [100, 100])
screen.blit(myBrid, [50, 50])
pygame.display.flip()
pygame.time.delay(2000)
screen.blit(myBrid, [150, 50])
pygame.display.flip()
```

เมื่อรันจะได้ผลดังนี้



ก่อนที่จะได้เห็นนกตัวที่สองแสดงผลเราได้หยุดไปสองวินาที ด้วยคำสั่ง `pygame.time.delay(2000)`

การทำการเคลื่อนไหว (Animation)

สมมติว่าเราจะทำให้นกเคลื่อนที่ ลบนกตัวเก่า คือการวาดนกใหม่ที่ตำแหน่งใหม่ จะทำให้เหมือนกับนกเคลื่อนที่ การลบเราจะวาดสีที่เหมือนกับสีของฉากหลังดังนี้

```
myBrid = pygame.image.load("angry.png")
myBrid = pygame.transform.scale(myBrid, [100, 100])
screen.blit(myBrid, [50, 50])
pygame.display.flip()
pygame.time.delay(2000)
pygame.draw.rect(screen, THECOLORS["green"], [50, 50, 100, 100])
screen.blit(myBrid, [150, 50])
pygame.display.flip()
```

หรืออีกวิธีเราอาจจะลบทั้งหน้าจอแล้ววาดใหม่
ลองดูตัวอย่างโค้ดต่อไปนี้ที่วาดทีละนิด

```

myBrid = pygame.image.load("angry.png")
myBrid = pygame.transform.scale(myBrid, [100, 100])
x = 50
y = 50
screen.blit(myBrid, [x, y])
pygame.display.flip()
for i in range(1000):
    pygame.time.delay(20)
    pygame.draw.rect(screen, THECOLORS["green"], [x,y, 100,100])
    x = x + 1
    screen.blit(myBrid, [x, y])
    pygame.display.flip()

```

การชนขอบจอ

จากตัวอย่างโค้ดนี้ เติมนจนหลุดหน้าจอ เราควรทำให้ด้งกลับได้ เมื่อถึงขอบดงโค้ดทั้งหมดนี้ เราจะประกาศ vx ทิศทางการเคลื่อนที่ x จะเปลี่ยนตาม vx เมื่อถึงขอบขวามือ ตำแหน่ง x จะมีค่าเท่ากับ ขนาดของ screen_width ให้เราลกลับทิศของการเคลื่อนที่ และกลับรูปภาพด้วยคำสั่ง pygame.transform.flip() เลือกกลับเฉพาะตามแนวแกน x

```

x = x + vx
screen_width = screen.get_width()
if x > screen_width - 100 or x < 0:
    vx = -vx
    myBrid = pygame.transform.flip(myBrid, True, False)

```

เราย้ายโค้ดของการเคลื่อนที่ทั้งหมดมาไว้ระหว่างรันโปรแกรม ที่ while ดังนั้นโค้ดทั้งหมด จะเป็นดังนี้

```

import pygame
import sys
from pygame.color import THECOLORS

screen_width = 800
screen_height = 600
pygame.init()
screen = pygame.display.set_mode([screen_width, screen_height])
screen.fill(THECOLORS["green"])

myBrid = pygame.image.load("angry.png")
myBrid = pygame.transform.scale(myBrid, [100, 100])
x = 50
y = 50
vx = 2
screen.blit(myBrid, [x, y])
pygame.display.flip()
running = True
while running:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False

    pygame.time.delay(20)
    pygame.draw.rect(screen, THECOLORS["green"], [x,y, 100,100])
    screen.blit(myBrid, [x, y])
    pygame.display.flip()

    x = x + vx
    screen_width = screen.get_width()
    if x > screen_width - 100 or x < 0:
        vx = -vx
        myBrid = pygame.transform.flip(myBrid, True, False)

pygame.quit()

```

เราสามารถให้เคลื่อนที่ได้ทั้ง แกน x, y ดังนี้ โดยกำหนดตัวแปร vy ขึ้นมาอีก

```

x = x + vx
y = y + vy
screen_width = screen.get_width()
screen_height = screen.get_height()
if x > screen_width - 100 or x < 0:
    vx = -vx
    myBrid = pygame.transform.flip(myBrid, True, False)
if y > screen_height - 100 or y < 0:
    vy = -vy

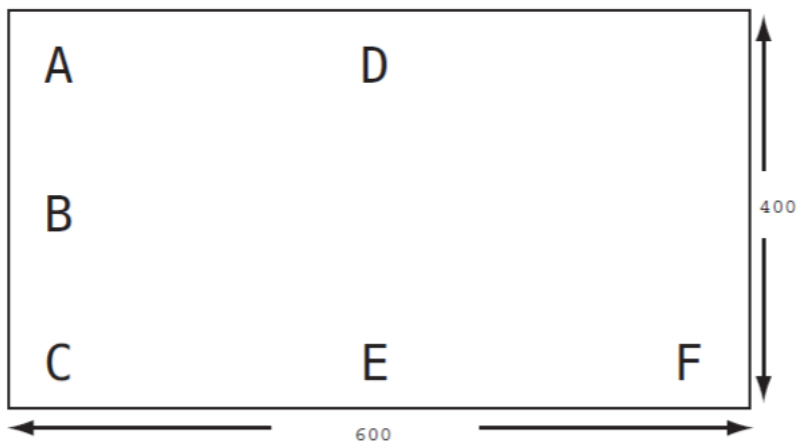
```

ทบทวน

- การใช้ pygame
- การวาดรูปด้วยคำสั่ง pygame.draw
- ระบบสี rgb
- การวาดรูปที่ต้องการ image
- การสร้างโปรแกรมเคลื่อนไหวที่รูป และให้ชนขอบ

ทดสอบความรู้

1. ค่าสีเท่ากับ [255, 255, 255] สีที่แสดงคืออะไร
2. ค่าสีเท่ากับ [0, 255, 0] สีที่แสดงคืออะไร
3. จงแสดงคำสั่งวาดรูปสี่เหลี่ยมด้วย pygame
4. จงแสดงคำสั่งวาดเส้นจากจุด
5. Pixel หมายถึงอะไร
6. จงวาดหน้าจอและชี้ที่ตำแหน่ง [0, 0] อยู่ตรงไหน
7. ถ้าหน้าจอขนาดกว้าง 600 และสูง 400 จงบอกว่า ตำแหน่ง [50, 200] อยู่ที่ตัวอักษรอะไร
8. จากข้อเจ็ด ตำแหน่ง [300, 50] คือตัวอักษรใด



9. จงแสดงโค้ดการวาด image
10. นอกขั้นตอนการทำให้รูปเคลื่อนไหวที่ได้อย่างไร

จงเขียนโปรแกรมต่อไปนี้

1. จงเขียนโปรแกรม โดยใช้รูปเป็นหน้าตัวเอง แล้วทำให้ดังต่อไปนี้ ลองเปลี่ยนตัวแปรเหล่านี้
 - a. ความเร็วในแนวกแกน x และ y
 - b. ระยะเวลา delay