**การแสดงข้อความใน PyGame และการนับคะแนน**

ใน PyGame มี built-in ฟังก์ชันรองรับการแสดงข้อความ Text บนหน้าจอ เราสามารถนำเอา built-in function มาเขียนไว้ในฟังก์ชันใหม่ที่เราสร้างขึ้นมาดังตัวอย่างต่อไปนี้

def text\_objects(text, font):

textSurface = font.render(text, True, RED) #กำหนดสีของฟอนท์

return textSurface, textSurface.get\_rect() #กำหนด surface ที่จะแสดงฟอนท์เป็น #rectangular

def message\_display(text):

largeText = pygame.font.Font('freesansbold.ttf',115) # กำหนด font และขนาด

TextSurf, TextRect = text\_objects(text, largeText) #การสร้าง text ให้เป็น object

TextRect.center = ((displayWidth/2),(displayHeight/2)) #กำหนดตำแหน่งที่ text จะปรากฏ

gameDisplay.blit(TextSurf, TextRect) #แสดงผลบนหน้าจอ

pygame.display.update() #อัพเดทหน้าจอ

time.sleep(2) #หยุด 2 วินาที

ในการเรียกใช้ฟังก์ชัน นักเรียนสามารถเรียกได้ดังตัวอย่างต่อไปนี้

message\_display(‘Welcome)

message\_display(‘Game Over’)

**1) ให้นักเรียนลองแสดงข้อความ Welcome ก่อนเริ่มเกมส์ (ในฟังก์ชัน game\_loop ก่อนเริ่มเกมส์) )และแสดงข้อความ Game Over (ในเงื่อนไขที่ตรวจจับวัตถุชนกัน) เมื่อตัวละครในเกมส์ชนกับวัตถุ (เช่น อุกกาบาต หรือ วัตถุอื่นที่นักเรียนใช้ในเกมส์)**

ในอนาคตนักเรียนอาจต้องการเขียนฟังก์ชันเพิ่มเติมเพื่อจัดการส่วนของ Game Quit, Intro ของเกมส์ หรือแม้กระทั่ง การหยุดระหว่างเกมส์ (pause) และเราสามารถเรียกใช้ฟังก์ชัน message\_display( ) ในฟังก์ชันที่เราเขียนได้ อาทิเช่น

def crash():

message\_display(‘You Crashed!!!’)

def game\_intro():

message\_display(‘Welcome’)

**2) ให้นักเรียนเรียกใช้สองฟังก์ชันนี้แทนในตำแหน่งที่ต้องการแสดงผลข้อความเมื่อเริ่มเกมส์ และ เมื่อเกิดการชนกัน**

ในกรณีที่เราต้องการแสดงผลข้อความที่มีการเปลี่ยนแปลงเมื่อเกิดเหตุการณ์ขึ้น เช่น การนับคะแนน การแสดงผล Level หรือข้อความอื่นๆ ระหว่างเกมส์ เราสามารถเขียนฟังก์ชันอย่างง่ายขึ้นมาเพื่อจัดการ

def scoreCount(count):

font = pygame.font.SysFont(None, 25) # ไม่ระบุชนิดฟอนท์ ขนาด 25

text = font.render("Score: "+str(count), True, BLACK) # สร้างฟอนท์ Score และค่าจาก count

gameDisplay.blit(text,(0,0)) # แสดงข้อความบนจอที่ตำแหน่ง 0,0

**3) ฟังก์ชัน scoreCount(count) ด้านบนใช้สำหรับแสดง text และค่า count ที่ต้องการ ดังนั้นนักเรียนต้องทำการปรับแก้ code ในส่วนที่ตรวจจับการชนกันของวัตถุ เช่น ในฟังก์ชัน fallingFruit( ) ต้องส่งค่า count เป็น argument เพิ่ม และ return เมื่อจบฟังก์ชัน และในเงื่อนไขที่ตรวจจับการชนกันค่า count จะเพิ่มอีก 1 เมื่อเงื่อนไขถูกต้อง หลังจากนั้น (นอกเงื่อนไข) ให้เรียกฟังก์ชัน scoreCount(count) ในฟังก์ชัน gameLoop( ) นักเรียนต้องประกาศตัวแปร count = 0**

**4) ให้นักเรียนสร้างฟังก์ชัน def levelCount(level): เพื่อแสดงข้อความ Level ในเกมส์ โดยที่เมื่อคะแนนที่เก็บได้ครบ 10 คะแนน ให้ปรับ Level ขึ้น 1 ระดับ**

**5) ให้นักเรียนปรับฟังก์ชัน fallingFruit( ) หรือในชื่ออื่นๆที่นักเรียนเลือกใช้ โดยกำหนดให้มีการสุ่มการตกของผลไม้ต่างสีกัน และมีคะแนนต่างกัน**

Hint: นักเรียนสามารถสร้าง fallingFruit2() ขึ้นมาสำหรับผลไม้ต่างสี ภายในฟังก์ชันในส่วนของเงื่อนไขการตรวจจับการชนกัน ให้เพิ่มค่า count มากกว่า (คะแนนตามแต่ที่นักเรียนจะกำหนด เช่น ผลไม้ปกติ 1 คะแนน ผลไม้พิเศษ 3 คะแนน)

ใน gameLoop( ) ก็ทำการเรียกฟังก์ชัน fallingFruit2( ) ภายใต้เงื่อนไขด้านล่างเหมือนกันการเรียก fallingFruit( )

if not gameExit:

เนื่องจากเราจะมี fallingFruit( ) และ fallingFruit2( ) จึงจำเป็นที่จะต้องกำหนดค่าตัวแปร (เช่น flag) ขึ้นมาเพื่อจะบอกว่าถ้าเกิดกันชนกันของวัตถุ ค่าตัวแปรนี้จะเปลี่ยนไป เช่น จาก 0 เป็น 1 ในฟังก์ชันนึง และเปลี่ยนจาก 0 เป็น 1 ในอีกฟังก์ชันนึง เพื่อที่เราจะสามารถใช้ตั้งเงื่อนไขใน gameLoop ได้ว่าถ้าเป็น 0 ให้เรียก fallingFruit( ) ถ้าเป็น 1 ให้เรียก fallingFruit2( ) เป็นต้น ทั้งนี้เราจำเป็นต้อง return ค่า flag ออกมาจาก fallingFruit( ) และ fallingFruit2( ) ด้วย และกำหนดตัวแปร flag เริ่มต้นใน gameLoop() ให้เท่ากับ 0

ตัวอย่างการกำหนดเงื่อนไขเพื่อใช้เรียกระหว่าง fallingFruit( ) และ fallingFruit2( ) เช่น ภายในเงื่อนไข

if not gameExit:

if flag == 0: # เงื่อนไขสำหรับตรวจสอบว่าเราจะเรียก fallingFruit( ) ไหน

เรียก fallingFruit( )

if not gameExit: #เงื่อนไขสำหรับเรียกฟังก์ชัน fallingFruit2( )

if flag == 1:

เรียก fallingFruit2( )

ถ้านักเรียนกำหนดเงื่อนไขข้างบนถูกต้องใน gameLoop( ) ผลไม้ที่ตกลงและตัวละครหลักของเราเก็บได้ ผลไม้ก็จะตกสลับกันระหว่างผลไม้ปกติ กับผลไม้พิเศษ ถ้าเก็บไม่ได้ผลไม้แบบเดิมก็จะตกลงมาใหม่

หากเราต้องการให้ผลไม้พิเศษตกลงมาในจำนวนน้อยครั้งกว่าเมื่อเทียบกับผลไม้ปกติ สามารถกำหนดเงื่อนไขเพิ่มเติมอาทิเช่น สุ่มตัวเลขมาระหว่าง 1 ถึง 8 และเรา % ด้วย 5 แล้วเท่ากับ 0 แสดงว่าอัตราการตกของผลไม้พิเศษ (โดยเฉลี่ยคือ) 7 ต่อ 1