**การสร้างปุ่มแบบ Interactive และการจัดการเรื่องเสียงใน PyGame**

ใน PyGame เราสามารถสร้างปุ่มโดยการวาดวัตถุเป็นรูปสี่เหลี่ยมบนหน้าจอ และเขียนโค้ดเพื่อบริหารจัดการรูปสี่เหลี่ยม (ปุ่มบนหน้าจอ) ให้มีลักษณะ Interactive เมื่อเราทำการคลิ๊กเมาส์บนปุ่มเพื่อให้เกมส์เริ่ม หรือ ออกจากเกมส์ และเพื่อรองรับการหยุดระหว่างเล่นเกมส์เราสามารถเขียนโค้ดให้คอยตรวจจับการกดปุ่ม อาทิ กดปุ่ม p เพื่อให้เกมส์หยุด และแสดงข้อความเช่น Game Paused และมีปุ่มให้เลือกกดว่าจะ continue หรือจะ quit เป็นต้น

อันดับแรกเราต้องมีสร้างฟังก์ชันรองรับการสร้างปุ่มในเกมส์ก่อน โดยมีโค้ดสำหรับการวาดปุ่มในเกมส์ดังนี้

|  |
| --- |
| def button(msg,x,y,w,h,ic,ac,action=None): mouse = pygame.mouse.get\_pos() #สำหรับหาค่าตำแหน่งของเมาส์ click = pygame.mouse.get\_pressed() #การกดปุ่มเมาส์ if x+w > mouse[0] > x and y+h > mouse[1] > y: # ถ้าเมาส์อยู่ในพื้นที่ของปุ่ม pygame.draw.rect(gameDisplay, ac,(x,y,w,h)) # ให้วาดปุ่มสี่เหลี่ยมและกำหนดสถานะเป็น active # (ac) สีของปุ่มจะเปลี่ยน if click[0] == 1 and action != None: # ถ้ามีการกดปุ่มเมาส์ action() # เรียกฟังก์ชัน action ที่ส่งเข้ามา else: pygame.draw.rect(gameDisplay, ic,(x,y,w,h)) # ถ้าเมาส์ไม่อยู่บนปุ่ม ก็กำหนดสถานะเป็น# inactive smallText = pygame.font.SysFont(None,20) #กำหนดฟอนท์และเขียนฟอนท์ในปุ่ม textSurf, textRect = text\_objects(msg, smallText) textRect.center = ( (x+(w/2)), (y+(h/2)) ) gameDisplay.blit(textSurf, textRect) |

จากสัปดาห์ก่อนในฟังก์ชัน def crash(): ที่มีการเรียกแสดงข้อความ เราจะทำการปรับปรุงโค้ดเพื่อรองรับการใช้งานปุ่มที่สร้างเมื่อ เพื่อถามว่าจะเล่นเกมส์ต่อ หรือ จะออกจากเกมส์ โดยสามารถเพิ่มโค้ดต่อไปนี้ เมื่อเกิดการชนกันและมีการเรียกฟังก์ชัน crash() ในตัวฟังก์ชัน crash() จะสร้างปุ่มขึ้นมาสองปุ่มคือ Play Again และ Quit

|  |
| --- |
|  while True: for event in pygame.event.get(): #เช็ค event ว่ามีการปิดเกมส์หรือไม่ ถ้าใช่ก็ออกจากเกมส์ if event.type == pygame.QUIT: pygame.quit() quit() #gameDisplay.fill(WHITE) button("Play Again",150,450,100,50,GREEN,BRIGHTGREEN,gameLoop)  button("Quit",550,450,100,50,RED,BRIGHTRED,quitgame) pygame.display.update() clock.tick(15) |

คำอธิบายการเรียกใช้ฟังก์ชัน button()

 button("Play Again",150,450,100,50,GREEN,BRIGHTGREEN,gameLoop)

* Play Again คือข้อความที่ส่งเข้าไปในฟังก์ชัน button()
* x, y, w, h คือตำแหน่งที่จะวางปุ่ม และความสูงและกว้างของปุ่ม
* GREEN, BRIGHTGREEN คือสีที่เราจะให้ปุ่มแสดงตอนสถานะของปุ่มว่าเป็น active หรือ inactive
* gameLoop() คือฟังก์ชันที่เราส่งเข้าไปเมื่อมีการเรียกฟังก์ชัน action() ในนั้นก็จะทำการเรียก gameLoop()

**1) ให้นักเรียนปรับปรุงโค้ดสำหรับ gameIntro() ให้นอกเหนือจากแสดงข้อความ Welcome ให้รองรับการสร้างปุ่ม สำหรับเริ่มเล่นเกมส์ (START) หรือหยุด (QUIT) Hint: หลักการคล้ายกับ crash() ที่เพิ่งปรับปรุงไปก่อนหน้านี้**

ในระหว่างการเล่นเกมส์เราอาจจะต้องหยุดเกมส์เป็นการชั่วคราว ดังนั้นจำเป็นต้องสร้าง logic สำหรับการ pause และ unpause ในเกมส์ โดยเราจะสร้างเป็นฟังก์ชัน อันดับแรกก็กำหนดฟอนท์และขนาดของฟอนท์ **และให้นักเรียนทำส่วนที่เหลือโดยใช้หลักการเดียวกันกับด้านบนคือสร้างสองปุ่ม Continue และ Quit**

|  |
| --- |
| def pause(): largeText = pygame.font.SysFont("freesansbold.ttf",115) TextSurf, TextRect = text\_objects("Paused", largeText) TextRect.center = ((displayWidth/2),(displayHeight/2)) gameDisplay.blit(TextSurf, TextRect) |

เมื่อกดปุ่ม Continue ก็ให้เรียกฟังก์ชัน unpause() ซึ่งจะกล่าวถึงต่อไป หรือเมื่อกด Quit ก็ให้เรียกฟังก์ชัน quitGame()

|  |
| --- |
| def unpause(): global pause pause = False |

ในฟังกัชัน gameLoop() เราต้องทำการปรับปรุงเพื่อตรวจจับการกดปุ่มเมื่อต้องการ pause ในกรณีเราจะใช้ปุ่ม p สำหรับกดเพื่อ pause game

ภายใต้เงื่อนไข if event.type == pygame.KEYDOWN: ใน gameLoop() นักเรียนต้องเพิ่มโค้ดต่อไปนี้ เมื่อมีการกดปุ่ม p ค่าตัวแปร pause จะเป็น True และเราก็ทำการเรียกฟังก์ชัน pause() ที่ทำไว้ก่อนหน้านี้

|  |
| --- |
| if event.key == pygame.K\_p: pause = True paused() |

เพื่อให้ตัวแปร pause สามารถใช้งานได้แบบ Global (นักเรียนไม่ต้องส่งค่าเข้าฟังก์ชัน และ return ค่าออกมา) ใน gameLoop() เราต้องทำการประการตัวแปรดังนี้

|  |
| --- |
| global pause |

**การจัดการเรื่องเสียงใน PyGame**

การจัดการเรื่องเสียงใน PyGame สามารถทำได้อย่างง่ายเนื่องจากมีฟังก์ชันใน PyGame ให้เรียกใช้ได้ทันที สิ่งที่เราต้องทำคือการโหลดไฟล์เสียง (เป็นไฟล์ .wav) ที่เราต้องการเล่นไว้ในเกมส์โดยใช้โค้ดต่อไปนี้ โดยก่อนใช้งานฟังก์ชันเสียงเราต้องทำการ Initialize ตัว mixer ก่อน

|  |
| --- |
| pygame.mixer.init()crash\_sound = pygame.mixer.Sound("crash.wav") # นักเรียนต้องโหลดเสียงสำหรับ action อื่นด้วย |

หลังจากที่เราโหลดไฟล์เสียงเราสามารถ play sound ที่เราโหลดไว้แล้วได้ดังนี้ (ตัวอย่างนี้เป็นการ play sound เมื่อเราชนกับอุกกาบาต) โดยเราเพิ่มโค้ดไว้ในฟังก์ชัน crash( ) ที่เราทำการปรับปรุงมาก่อนหน้านี้

|  |
| --- |
|  pygame.mixer.Sound.play(crash\_sound) pygame.mixer.music.stop() #time.sleep(1) #สามารถกำหนดการหน่วงเวลาถ้าต้องการ |

**ให้นักเรียนทำการ play sound สำหรับ**

* **การเก็บทุเรียน (หรือวัตถุอื่นที่นักเรียนใช้ในโปรแกรม) ไม่ต้องหน่วงเวลา**
* **เมื่อ Level Up ไม่ต้องหน่วงเวลา**

**สำหรับเพลงที่เราต้องการให้เล่นเป็น background ตลอดเวลาเช่นตั้งแต่เริ่มเกมส์ (เวลาเรียก gameLoop()) และต้องการให้เพลงของเราเล่นต่อเนื่อง) เราสามารถทำได้ดังนี้ 1) โหลดเสียง และ 2)เล่นเสียงนั้นโดยกำหนด option ให้เป็น -1 คือเล่นต่อเนื่อง**

|  |
| --- |
|  **############** **pygame.mixer.music.load('pacman\_backmusic.wav')** **pygame.mixer.music.play(-1)** **############** |

**Hint: ทั้งนี้ PyGame ไม่สามารถรองรับไฟล์ .wav ที่มี format บางอย่างได้ หากนักเรียนต้องการทดสอบว่าไฟล์ .wave ไหนใช้ใน PyGame ได้บ้างให้ทดสอบโดยใช้โค้ดต่อไปนี้ (แนะนำให้เปิด Notebook ใหม่สำหรับการทดสอบ อย่าทดสอบในโค้ดเกมส์ของนักเรียน) ให้เขียนแยกเป็น 2 cell ตามด้านล่าง และกดรัน cell แรกเพิ่มเล่นเสียง และกดรัน cell ที่ 2 เพื่อหยุดเล่น และทำการปิด mixer**

****

**ท้ายนี้หากเราต้องการให้เสียงเพลง background หยุดเล่นส์เมื่อเรากด p สำหรับ pause ใน PyGame ก็มีฟังก์ชันสำหรับรูปให้เรียกใช้คือ pygame.mixer.music.pause() และ pygame.mixer.music.unpause() สำหรับเล่นเสียงต่อ ให้นักเรียนเพิ่มโค้ดด้านบนในฟังก์ชัน pause() และ unpause() ที่นักเรียนได้พัฒนามาก่อนหน้านี้**