**พัฒนาเกมแนวยิงในอวกาศ (Cat in Space)**

**สุทธิภัทร เจียรศิริสิน 016630641008-4**

**รายงานนี้เป็นส่วนหนึ่งของการศึกษาวิชาการโปรแกรมคอมพิวเตอร์**

**สาขาวิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์และเทคโนโลยี**

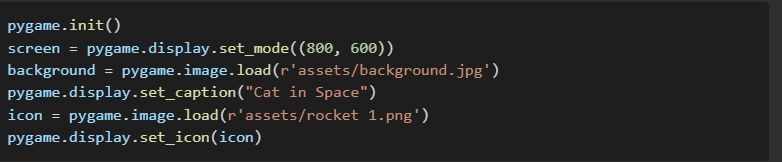
**มหาวิทยาลัยเทคโนโลยีราชมงคลตะวันออก**

**ภาคการศึกษาที่ 2 ปีการศึกษา 2567**

**1. องค์ประกอบหลักของเกม**

เกม “Cat in Space” เป็นเกมที่พัฒนาโดยใช้ภาษา Python และไลบรารี Pygame โดยเป็นเกมยิงศัตรูที่ให้ผู้เล่นควบคุมยานอวกาศ ยิงกระสุนไปทำลายศัตรู และเก็บคะแนน เกมมีระบบการเล่นที่ต่อเนื่องจนกว่าศัตรูจะมาถึงพื้นที่ของผู้เล่น ทำให้เกมจบลง

**1.1 การเริ่มต้น Pygame**



pygame.init() เรียกใช้เพื่อเริ่มต้นระบบ Pygame

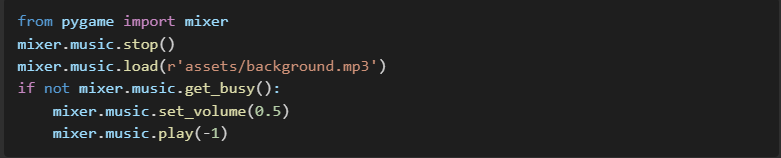
pygame.display.set\_mode((800, 600)) ใช้สร้างหน้าจอเกมขนาด 800x600 พิกเซล

โหลดรูปภาพพื้นหลังและกำหนดให้แสดงในเกม

ตั้งชื่อเกมให้แสดงบนแถบหน้าต่าง (pygame.display.set\_caption("Cat in Space"))

ตั้งค่าไอคอนของเกมให้เป็นรูปภาพที่กำหนด

**1.2 ระบบเสียง**



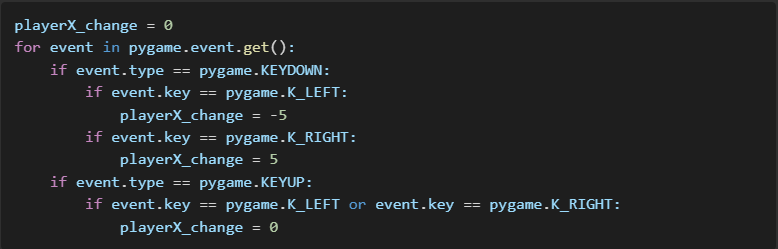
mixer.music.stop() หยุดเพลงที่อาจกำลังเล่นอยู่ก่อนหน้า

mixer.music.load(r'assets/background.mp3') โหลดไฟล์เสียงพื้นหลัง

เช็คว่าเพลงกำลังเล่นอยู่หรือไม่ (mixer.music.get\_busy()) หากไม่มีการเล่นเพลงอยู่ จะตั้งระดับเสียงที่ 50% และเล่นเพลงซ้ำไปเรื่อย ๆ (-1 หมายถึงเล่นซ้ำแบบไม่มีที่สิ้นสุด)

นอกจากนี้ยังมีการใช้เสียงเอฟเฟกต์อื่น ๆ เช่น เสียงกระสุน (laser.wav) และเสียงระเบิด (Boom.wav) ในเกม

**1.3 การควบคุมผู้เล่น**

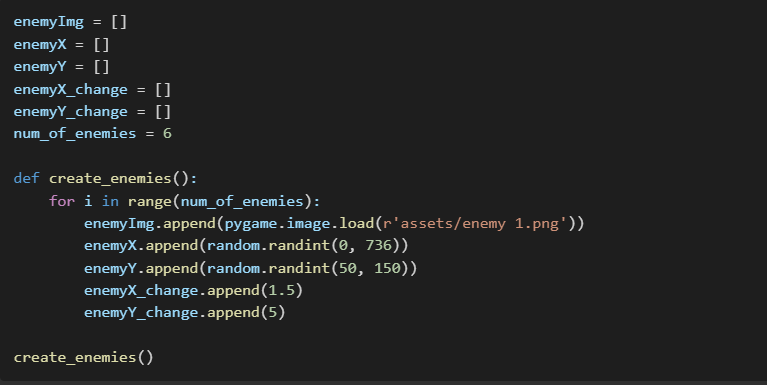


ผู้เล่นสามารถควบคุมยานอวกาศด้วยปุ่มลูกศรซ้าย (K\_LEFT) และขวา (K\_RIGHT)

สามารถยิงกระสุนด้วยปุ่ม SPACE BAR (K\_SPACE)

กำหนดให้ไม่สามารถยิงกระสุนได้ต่อเนื่องเร็วเกินไปโดยใช้ตัวแปร bullet\_cooldown

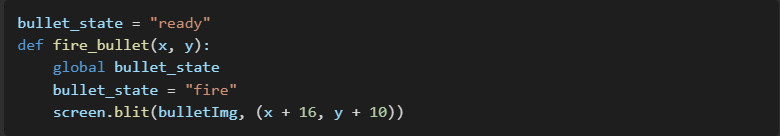
**1.4 ระบบศัตรู**



ศัตรูถูกสร้างขึ้นเป็นอาร์เรย์ของตำแหน่งและภาพ ศัตรูเคลื่อนที่ไปมาและเคลื่อนที่ลงมาเมื่อถึงขอบหน้าจอ

ความเร็วของศัตรูเพิ่มขึ้นเมื่อผู้เล่นกำจัดศัตรูได้

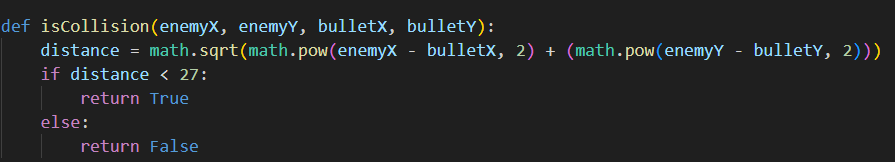
**1.5 ระบบกระสุน**

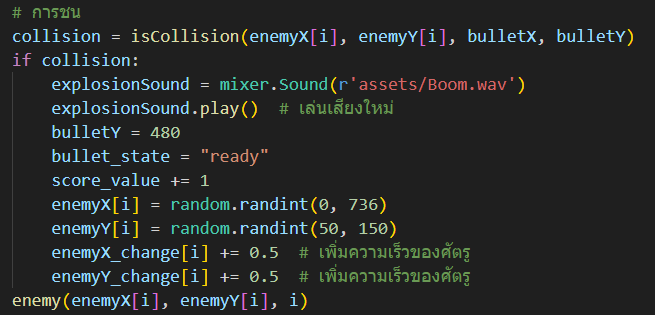


ใช้สถานะ bullet\_state เพื่อตรวจสอบว่ากระสุนพร้อมยิง ("ready") หรือกำลังเคลื่อนที่ ("fire")

กระสุนเคลื่อนที่ขึ้นไปและรีเซ็ตเมื่อพ้นขอบบนของหน้าจอ

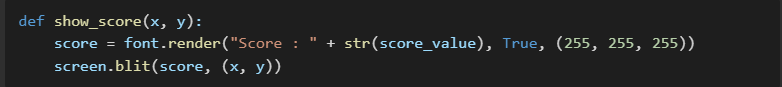
**1.6 การตรวจสอบการชนกัน**

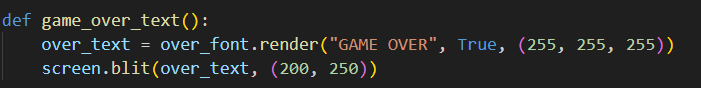


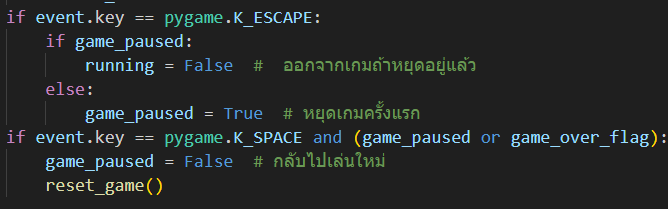


ใช้ฟังก์ชัน isCollision(enemyX, enemyY, bulletX, bulletY) ในการตรวจสอบว่ากระสุนชนศัตรูหรือไม่ ถ้าชนกัน ศัตรูจะถูกทำลาย และคะแนนของผู้เล่นเพิ่มขึ้น มีเสียงระเบิดเมื่อศัตรูถูกทำลาย

**1.7 ระบบคะแนนและการจบเกม**

****

****

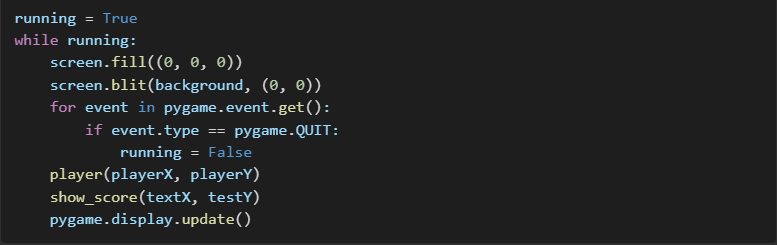
****

แสดงคะแนนของผู้เล่นที่มุมซ้ายบนของหน้าจอ

ถ้าศัตรูเคลื่อนที่ถึงผู้เล่น เกมจะจบลง และแสดงข้อความ "GAME OVER"

ผู้เล่นสามารถกดปุ่ม ESC เพื่อออกจากเกม หรือกด SPACE เพื่อเริ่มเกมใหม่หลังจากจบเกม

**1.8 การทำงานของลูปหลักของเกม**

****

1. เคลียร์หน้าจอและแสดงภาพพื้นหลัง

2. ตรวจจับอินพุตของผู้เล่น เช่น การกดปุ่มเพื่อเคลื่อนที่หรือยิงกระสุน

3. อัปเดตตำแหน่งของผู้เล่น กระสุน และศัตรู

4. ตรวจสอบการชนกันของกระสุนและศัตรู

5. แสดงผู้เล่น ศัตรู กระสุน และคะแนนบนหน้าจอ

6. ตรวจสอบเงื่อนไขการจบเกม และอัปเดตเฟรมเกม