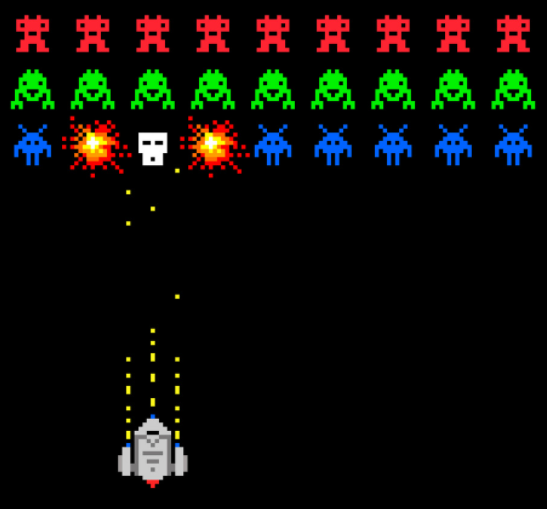
**#Final Game** นายชินพัฒน์ ลิ้มประธาน 64015030



Alien of typing เป็นเกม arcade คล้ายกับเกม space invaders คือเป็นเกมยิงเอเลียนที่จะเข้ามาโจมตียานอวกาศของเรา โดยเราต้องปกป้องตัวเองด้วยการยิงกระสุนเลเซอร์ใส่เอเลียนด้วยการ พิมพ์คำศัพท์ที่อยู่ด้านใต้ยานของเอเลียนแต่ละตัว

เหตุผลที่ทำเกมนี้เพราะว่า อยากทำเกมแนว space invaders แต่อยากเปลี่ยนจากกดยิง มาเป็นพิมพ์ดีด เพื่อนำมาเป็นเกมเอาไว้ฝึกพิมพ์คำภาษาอังกฤษ

**ตัวอย่างเกม**

**Space Invader** **Alien of typing**

**วิธีการเล่น**

**หน้าเมนู**

**หน้าเลือกความยาก**

**หน้าต่างในเกมเมื่อเล่น**

**หน้า pause game**

**หน้าจบเกม**

**การควบคุม**

* Keyboard ใช้เพื่อพิมพ์คำที่แสดงขึ้นมาตรงศัตรูแต่ละตัว
* ปุ่ม Esc เพื่อ Pause เกมหรือ ออกจากการ Pause เกม

**การเก็บคะแนน**

เมื่อพิมพ์คำศัพท์ที่แสดงขึ้นมาตรงศัตรูแต่ละตัวจนครบคำ จะเป็นการจัดการกับศัตรูและจะได้คะแนนมา 1 คะแนน คูณกับ combo ที่เราสะสมมาได้ การเก็บ combo เกิดจากการพิมพ์ต่อเนื่องโดยไม่พิมพ์ผิด โดย combo จะเริ่มที่ 1 และบวกไปทีละ 0.25 เมื่อพิมพ์ต่อเนื่อง 5 ตัวอักษรโดยไม่พิมพ์ผิด ถ้าเกิดพิมพ์ผิดจะเป็นการ reset combo กลับมาเป็นค่าเริ่มต้นคือ 1

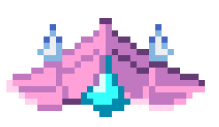
**เงื่อไขการจบเกม**

* ผู้เล่นจะจบเกมก็ต่อเมื่อกดออกจากเกมหรือ พลังชีวิตของผู้เล่นหมด

**ศัตรู**



ชื่อ: enemy-small  
ความยาวตัวอักษร: <= 3 ตัวอักษร



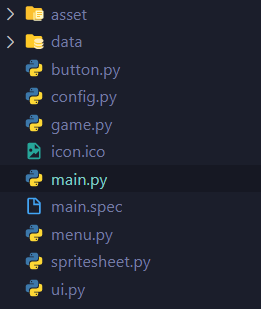
ชื่อ enemy-medium  
ความยาวตัวอักษร: <= 5 ตัวอักษร



ชื่อ enemy-big  
ความยาวตัวอักษร: > 5 ตัวอักษร

**เนื้อหาโค้ด**

**องค์ประกอบภายใน folder**



ไฟล์หลักที่กำหนดว่าอยู่หน้าไหน

เก็บคะแนนสูงสุดและคลังคำศัพท์

ไฟล์เก็บ ui ในหน้าเกมทั้งหมด

ไฟล์ที่ใช้สำหรับทำ animation ให้กับรูปภาพ

ไฟล์เก็บโครงสร้าง ui หน้าเมนูหลัก หน้าเลือกระดับความยาก

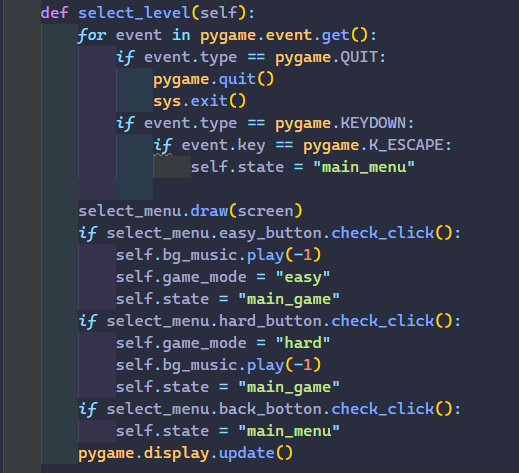
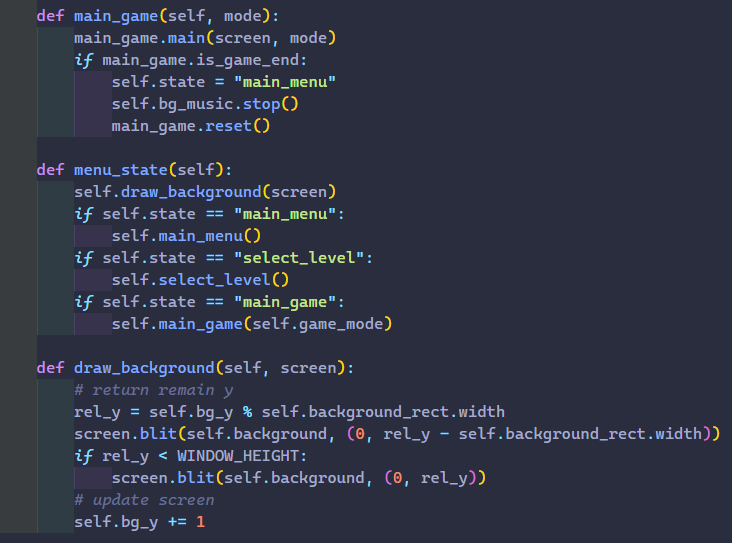
ไฟล์หน้าในเกม

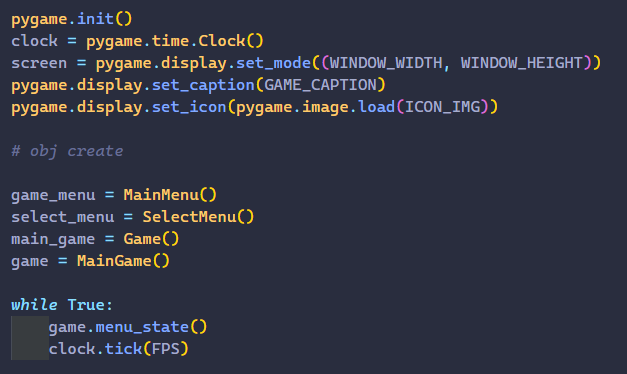
เก็บการตั้งค่าในเกม และที่อยู่ของไฟล์ต่างๆในเกม

ใช้สร้างปุ่มในเกม

**Code ภายในไฟล์**

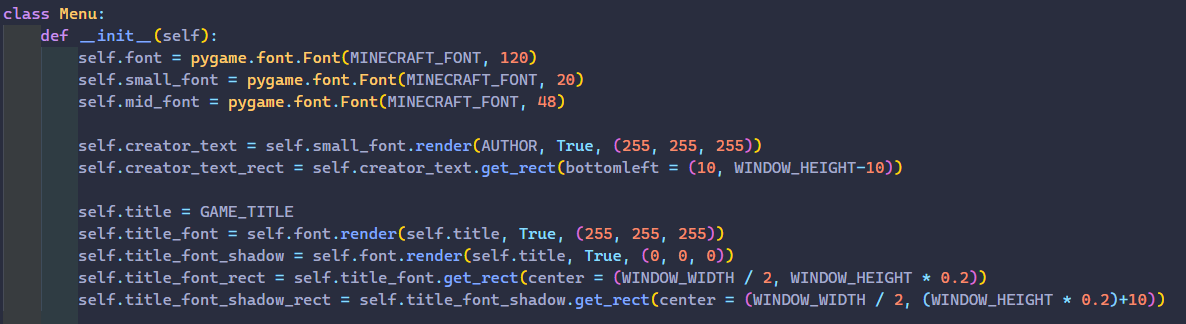
**main.py**

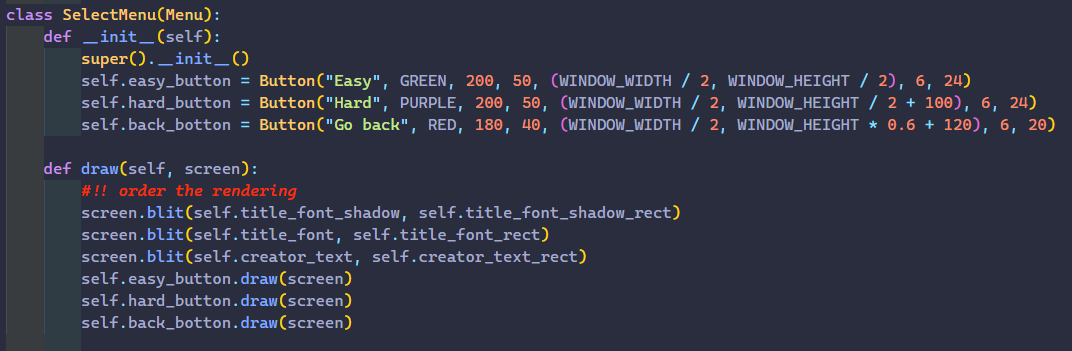




เป็นไฟล์หลักที่ใช้จัดการ สถานะหน้าของเกมว่าตอนนี้กำลังรันหน้าไหนอยู่ โดยที่เริ่มต้นมาจะ run main\_menu() คือหน้าหลัก และวาดปุ่มเริ่มเกมกับปุ่มออกเกม โดยที่ถ้ากดเริ่มเกมให้เปลี่ยน self.state = “select\_level” คือให้ไปหน้าเลือก level และถ้าเลือก level ได้แล้วจะทำการเปลี่ยนเป็น “main\_game” ซึ่งระบบจะเช็คว่าตอนนี้ต้อง run หน้าไหนโดย menu\_state() ซึ่งจะ run อยู่ใน while loop ตลอด เป็น loop หลัก

menu.py

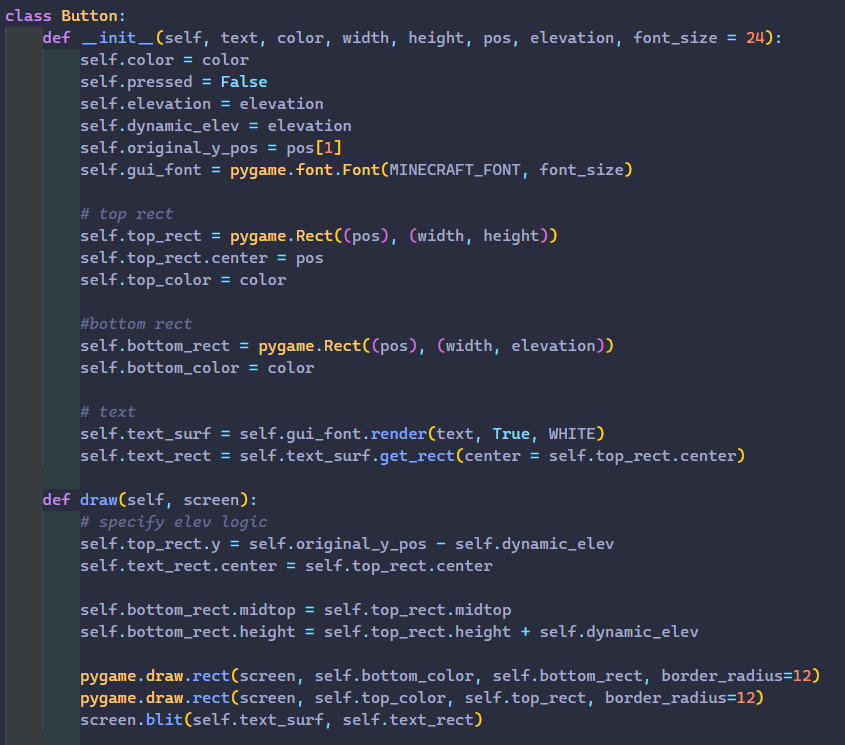


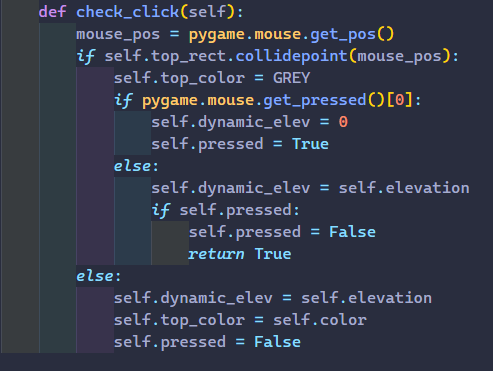




เป็นไฟล์ที่ใช้เก็บเมนูของหน้าหลัก ในเมนูนั้นก็จะเก็บปุ่มและพวก ui ต่างๆของหน้านั้น

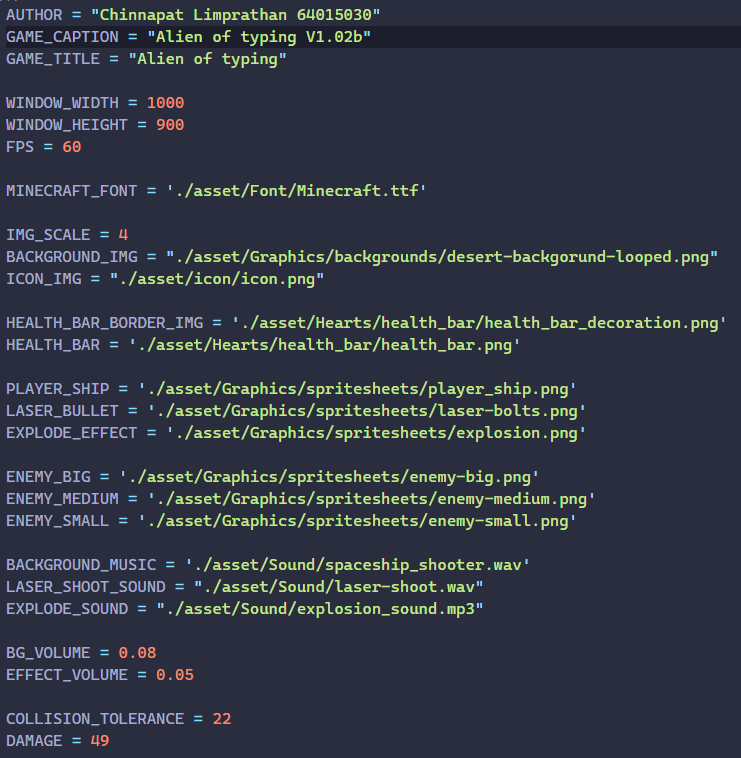
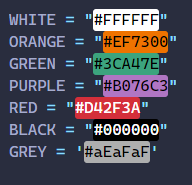
ซึ่งจะถูกเรียกใช้โดย main.py

button.py



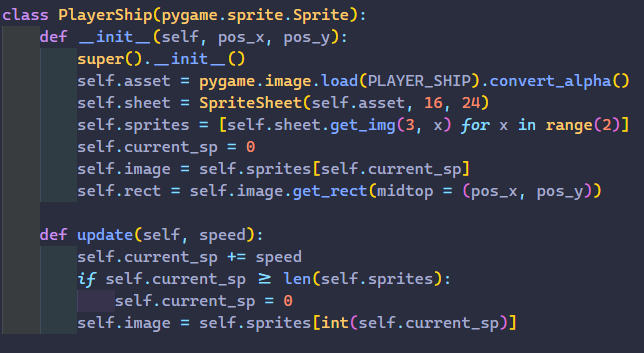
เป็นไฟล์ที่ไว้สำหรับทำปุ่มต่างๆ ของ ui ใน game และ เมนูต่าง

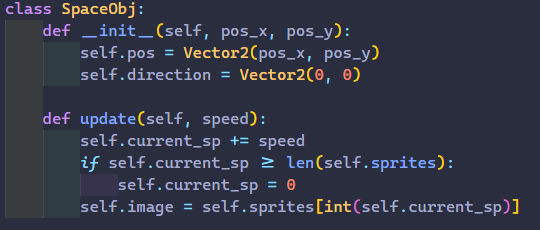
config.py

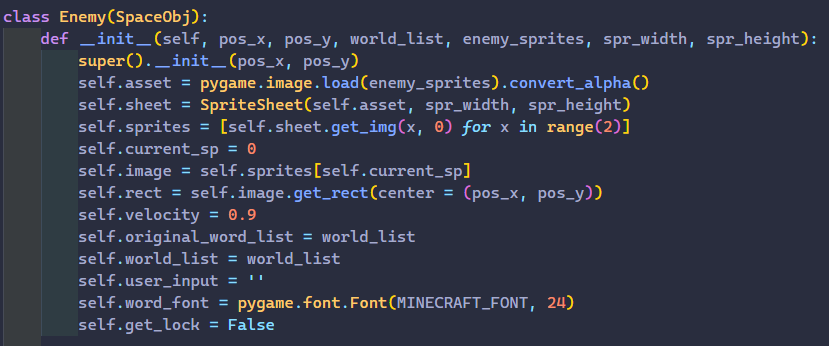


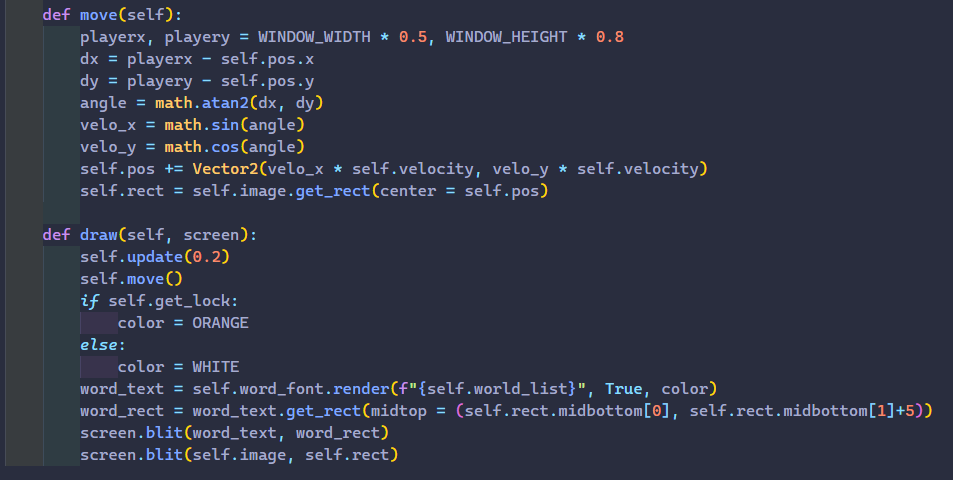
เป็นไฟล์ไว้สำหรับเก็บการตั้งค่าต่างๆของเกม เก็บที่อยู่ไฟล์ เพื่อความง่ายในการนำไปใช้ในแต่ละไฟล์

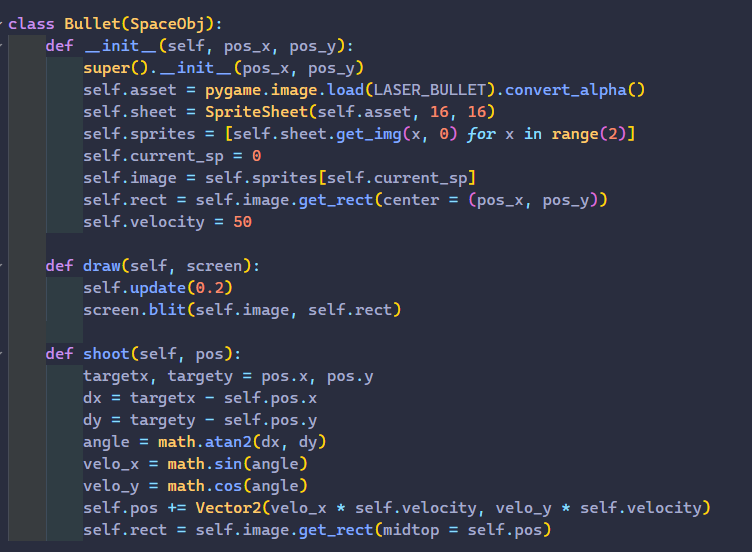
game.py

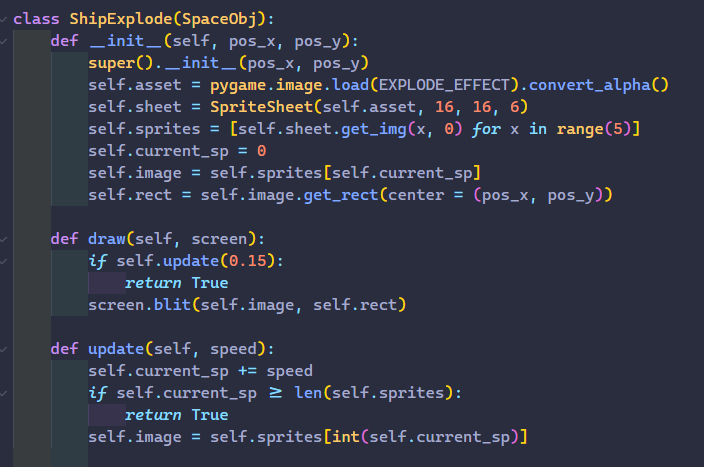


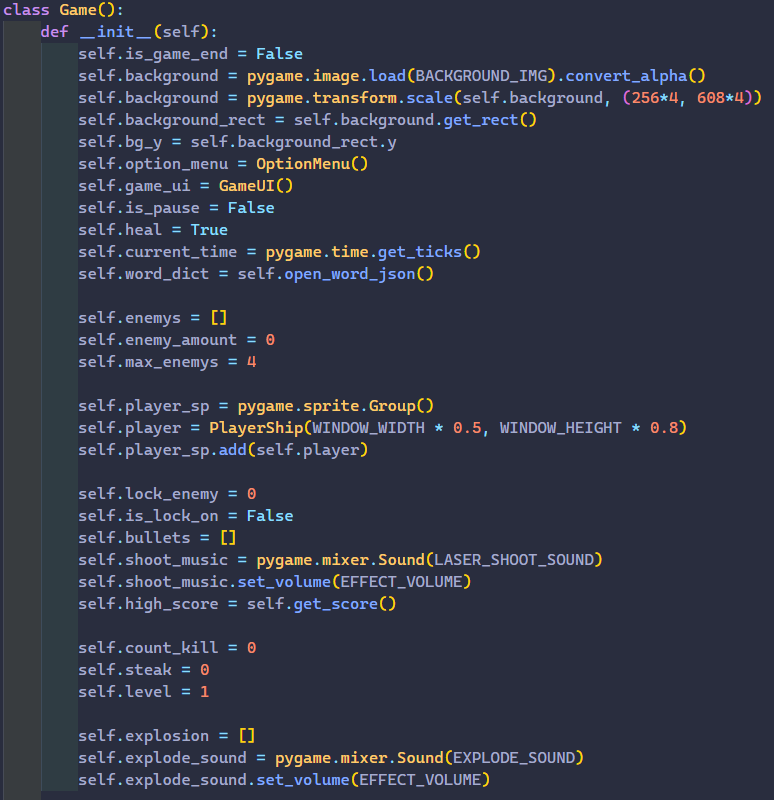


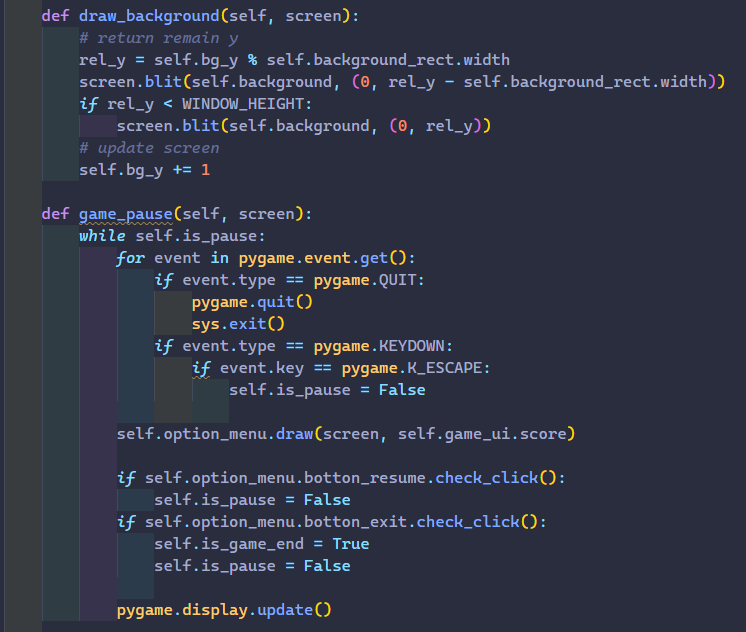


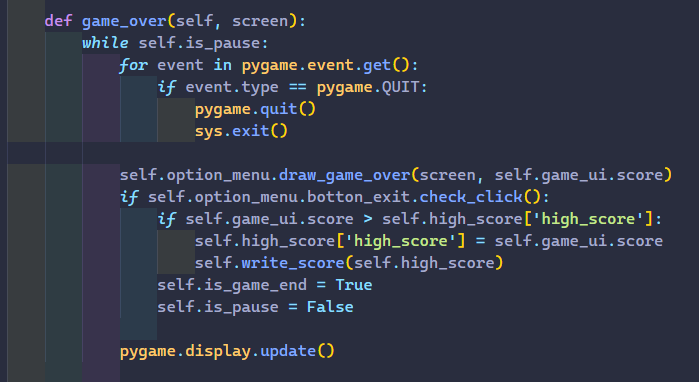


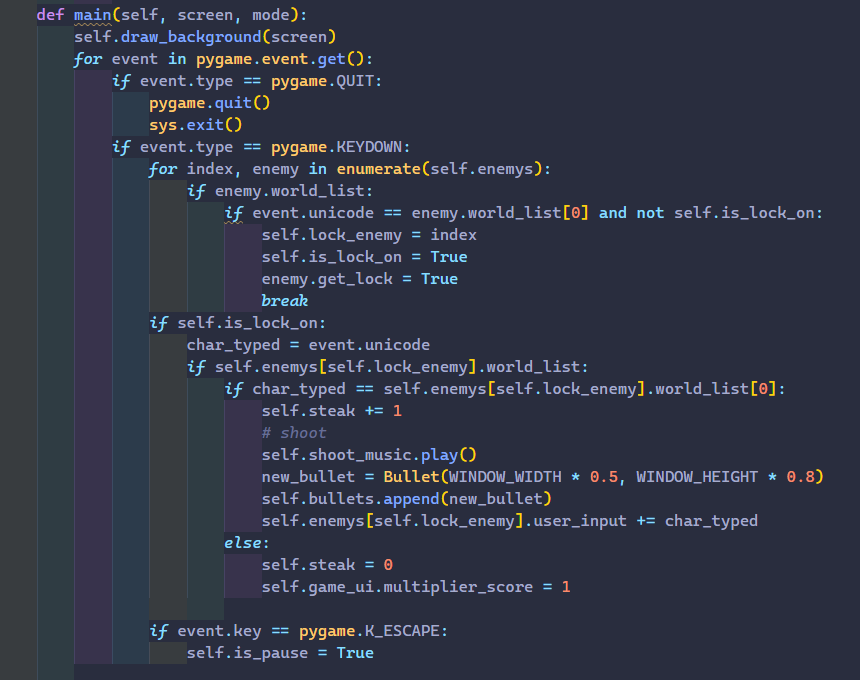


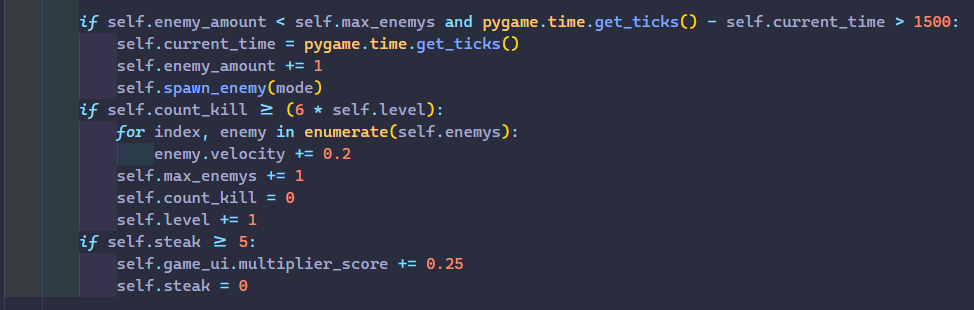


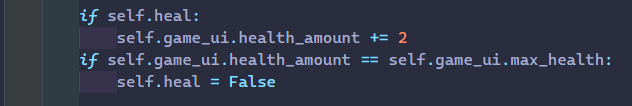


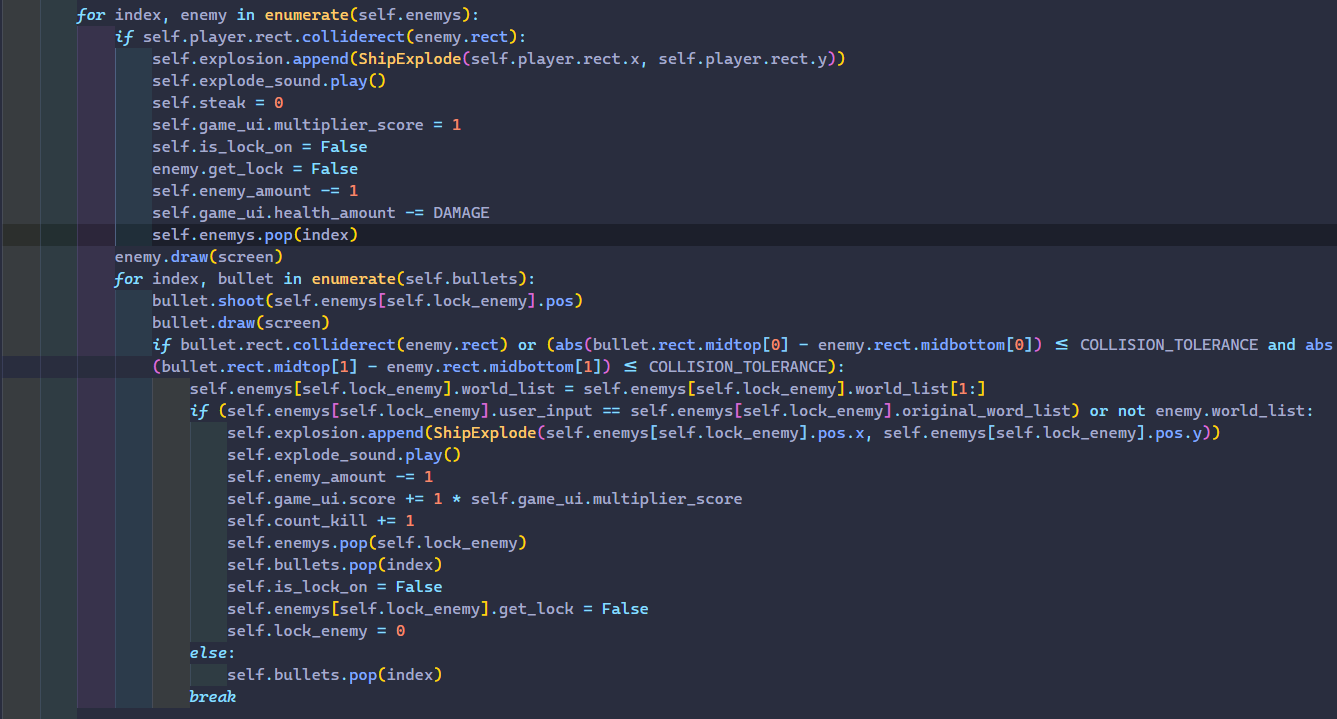


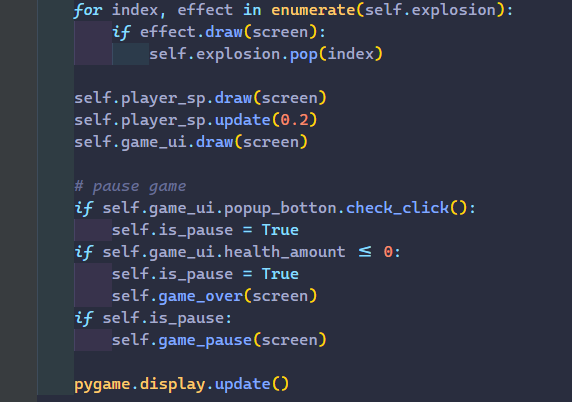


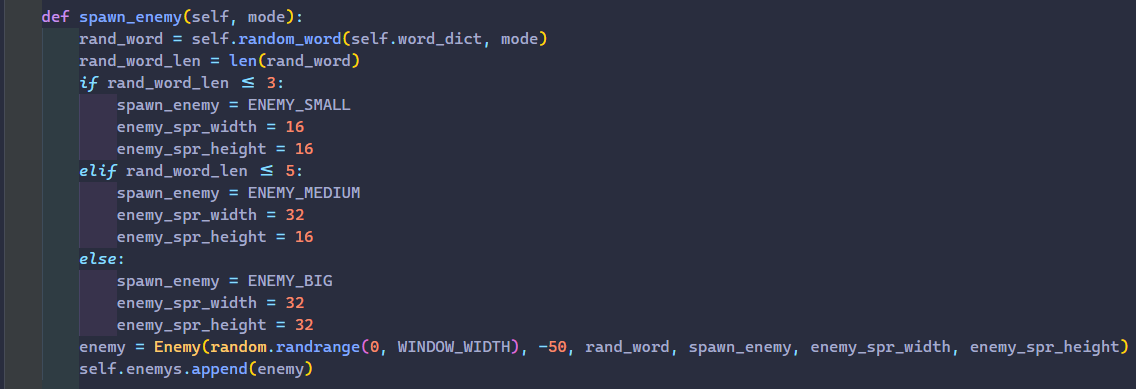


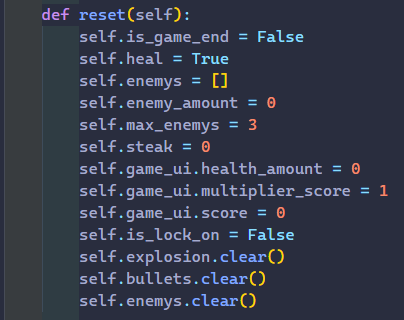


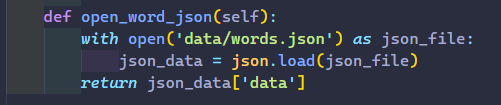


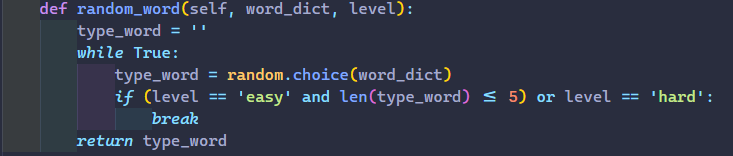


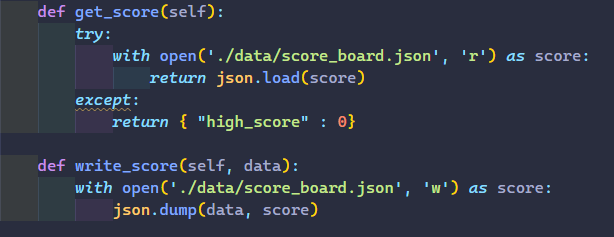










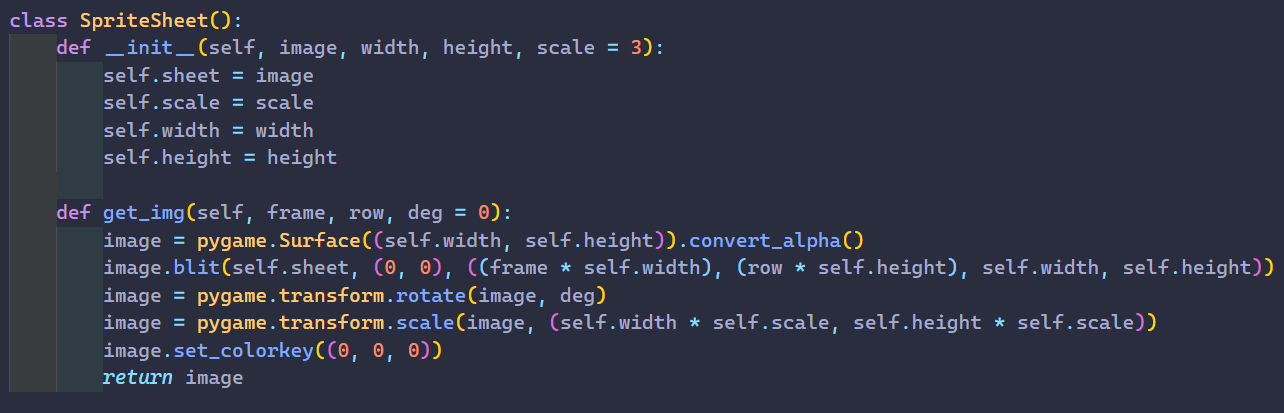


เป็นไฟล์ที่เกี่ยวข้องกับระบบของเกมทั้งหมด โดยที่จะมี class Game() เป็น class หลักในการจัดการเกม ซึ่งระบบของเกมจะรันอยู่ในฟังก์ชั่น main() โดยการทำงานของฟังก์ชั่น main คือ

* ดักจับ event ของ keyboard
* ตรวจสอบ หรือ ทำเงื่อนไขของเกม
* การ draw component ต่างๆ ลงใน game

1. เกมจะดักจับการกดปุ่มของ keyboard โดยที่จะมีอยู่ 3 ระบบย่อยคือ  
   1.1 ตรวจหาเป้าหมายว่าตัวอักษรแรกที่พิมพ์ตรงกับศัตรูตัวไหนที่อยู่ไกล้ไหม โดยถ้าเจอ ให้ทำการ lock เป้าหมายไว้  
   1.2 ระบบจะตรวจสอบว่ามีการ lock เป้าหมายไว้ไหม ถ้ามีการ lock จะตรวจสอบว่า ตัวอักษรที่พิมพ์มาตรงกับตัวอักษรตัวแรกของศัตรูที่ lock ไว้ไหม ถ้าตรงให้ทำการยิงใส่ศัตรูและขยับไปตัวอักษรถัดไป  
   1.3 ดักจับปุ่ม Esc เพื่อทำการ pause เกม
2. เกมทำการตรวจสอบเงื่อนไขดังนี้  
   2.1 ถ้าศัตรูที่มีอยู่น้อยกว่าจำนวนสูงสุดที่ศัตรูมีได้ และเวลาผ่านไป 1500 ms นับจากการเกิดตัวแรกของศัตรู ให้ทำการ spawn\_enemy() หรือก็คือเรียกศัตรูให้เกิดใหม่  
   2.2 ถ้า kill ศัตรูเกิน 6 ตัวขึ้นไป ให้ทำการเพิ่มความเร็วของศัตรู 0.2 และเพิ่มจำนวนสูงสุดของศัตรู และทำการขยับ kill ที่ใช้เพิ่มทั้งสองอย่างไป 2 เท่า  
   2.3 ถ้าพิมพ์ติดกัน 5 ตัวอักษรโดยไม่ผิดให้ทำการเพิ่มแต้มคูณ combo ไป 0.25  
   2.4 ถ้าเริ่มเกมใหม่ระบบเพิ่มเลือดจะ run ครั้งเดียวเพื่อเป็นการเพิ่มเลือดจาก 0 ไป 4 หน่วย และถ้าเพิ่มเต็มให้หยุดเพิ่ม  
   2.5 ทำการวน loop ในศัตรูแต่ละตัวเพื่อให้ศัตรูตรวจหาดังนี้  
   ทำการตรวจสอบว่าศัตรูได้ชนกับตัวของ player หรือยัง ถ้าชนให้เกิดการระเบิดเกิดขึ้นและ reset ค่า combo เป็น 1 และ ตรวจสอบว่า กระสุนที่ player ยิงมานั้น ถึงตัวของศัตรูหรือยังถ้าถึงแล้วจะเช็คอีกว่า กระสุนที่ยิงเข้ามานั้นตรงกับตัวสุดท้ายที่พิมพ์หรือไม่ ถ้าตรงให้ทำลายศัตรูตัวนั้นและเกิดการระเบิดขึ้น แต่ถ้าไม่ ให้กระสุนนั้นหายไปแทน  
   2.6 ทำการเช็คว่ามีการกดปุ่ม pause เกมหรือไม่ ถ้ามีให้ทำการ pause เกม  
   2.7 ทำการเช็คว่าเลือดนั้นเหลือ 0 หรือยัง ถ้าเหลือ 0 ให้ gameover ทันที
3. ส่วนของการ draw component จะ rendering order ดังนี้  
   3.1 วาด background  
   3.2 วาดศัตรูที่มี  
   3.3 วาดกระสุนที่มีการยิง  
   3.4 วาด effect ระเบิดที่มีการเกิดขึ้น  
   3.5 วาด player  
   3.6 วาด game ui หน้าต่าง หลอดเลือด score ปุ่มในเกมต่างๆ

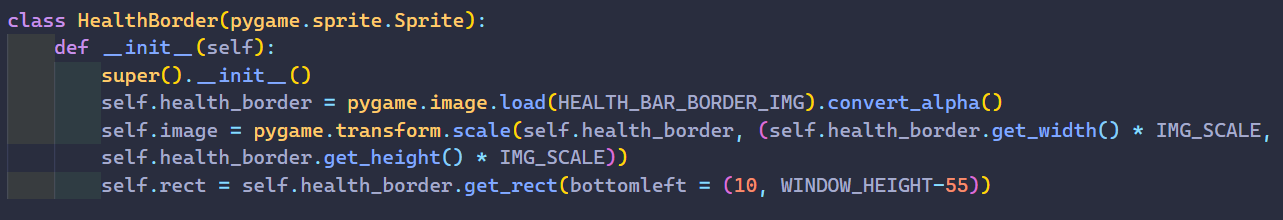
spritesheet.py



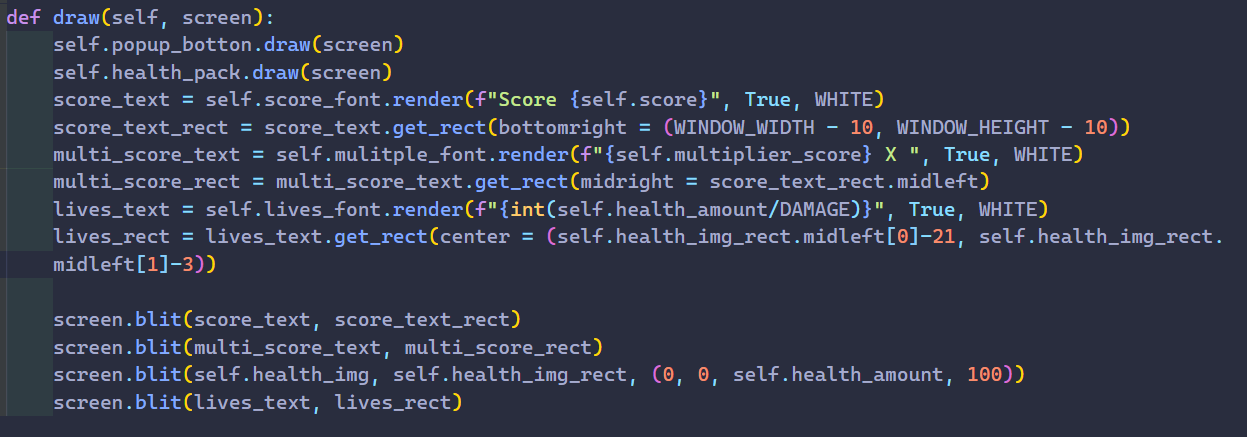
 ไฟล์นี้จะมี class SpritesSheet เพื่อเอาไว้ใช้สร้าง animation จากรูปภาพดังตัวอย่างนี้

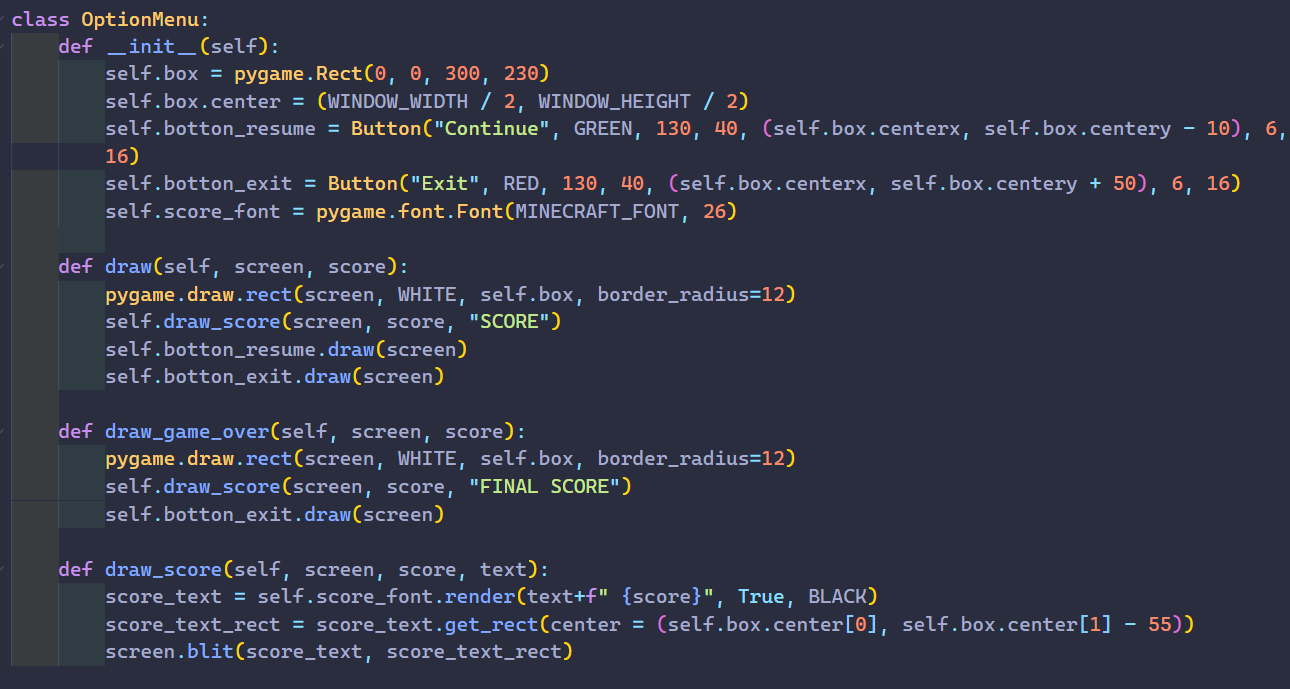
โดยทำการ crop รูปภาพให้ได้ size พอเหมาะกับ frame เดียวแล้วไล่ทำไปจนครบรูปที่มีแล้วสร้างเป็น list เอาไว้ใช้ทำ animation ต่อไป

ui.py





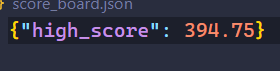




ไฟล์นี้เก็บหน้า ui component โครงสร้างทั้งหมด เช่นหลอดเลือด score หน้าต่าง pause ไว้ในไฟล์นี้ เพื่อให้ file game.py ไปเรียกใช้ต่อ

data/score\_board.json

ใช้เก็บคะแนน high score



data/words.json

ใช้เก็บคลังคำศัพท์ที่เอาไว้นำไปใช้ในเกม

