



## เกม Super Slime Tournament

นาย บัณณพนต์	ตากำวัน
นาย จักรพรรดิ	จืดดวงจันทร์
นาย วัชรกร	เย็นทวีทรัพย์

อาจารย์ที่ปรึกษา

อาจารย์ อนิราช      มิ่งขวัญ

โครงการเล่มนี้เป็นส่วนหนึ่งของการศึกษาตาม หลักสูตรวิศวกรรมศาสตรบัณฑิต  
สาขาวิชา วิศวกรรมสารสนเทศและเครือข่าย  
คณะเทคโนโลยีและการจัดการอุตสาหกรรม  
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

หัวข้อโครงการ เกมส์ Super Slime Tournament

ชื่อผู้เขียน นาย บัณฑิต คำคำวัน  
นาย จักรพรรดิ จี๊ดดวงจันทร์  
นาย วัชรกร เย็นทวีทรัพย์

อาจารย์ที่ปรึกษา อาจารย์ อนิราช มิ่งขวัญ

สาขาวิชา วิศวกรรมสารสนเทศและเครือข่าย

ปีการศึกษา 2564

### บทคัดย่อ

โครงการนี้มีวัตถุประสงค์ เพื่อให้ผู้เล่นได้รู้จักการแข่งขัน ผ่านทางเกมส์ Super Slime Tournament เพื่อให้ผู้เล่นสนุกกับมัน และทำให้เกิดการแข่งขันขึ้นด้วย

การพัฒนาเกมส์ Super Slime Tournament ดำเนินการวิเคราะห์และออกแบบเกมส์โดยนิทานและนิยายในโลกที่มีเวทย์มนต์เป็นต้นแบบ ที่ผู้คนสามารถพบเห็นได้บ่อย ทั้งใน ภาพยนตร์ เกมส์ การ์ตูน หรือสื่ออื่นๆ เกมส์ที่สร้างขึ้นมา โดยใช้ภาษา Python และ Pygame

จากการพัฒนาเกมส์ Super Slime Tournament พบว่า เกมนี้เหมาะสำหรับผู้เล่นทุกเพศทุกวัยที่มีความชอบสนใจการเล่นเกมส์และการแข่งขัน โดยผู้เล่นเพลิดเพลินพร้อมกับการฝึกทักษะการเล่น การแข่งขัน ไปพร้อมกัน

## กิตติกรรมประกาศ

โครงการนี้สำเร็จลงได้ด้วยความกรุณาจาก อาจารย์อนิราช มิ่งขวัญ ที่ได้ให้มาเป็นอาจารย์ที่ปรึกษาโครงการที่ได้ให้คำแนะนำแนวคิด ตลอดจนแก้ไขข้อบกพร่องต่างๆ มาโดยตลอดจนโครงการเล่มนี้เสร็จสมบูรณ์ ผู้ศึกษาจึงกราบขอบพระคุณเป็นอย่างสูง

ขอบคุณเพื่อนๆ ที่ช่วยสนับสนุนในการทำโครงการนี้ที่ได้ให้ความร่วมมือในการแสดงความคิดเห็นจนทำให้โครงการสำเร็จลงไปได้ด้วยดี

ขอบคุณ เว็บไซต์ Stack Overflow และ Youtube ช่อง KidsCanCode ที่ให้คำแนะนำและความรู้ในการพลิกแพลงออกแบบเกมส์

บัณฑิต คำวัน

จักรพรรดิ จี๊ดดวงจันทร์

วัชรกร เย็นทวีทรัพย์

## สารบัญ

อธิบาย Concept เกมส์	1
วิธีการเล่นเกมส์	2
อธิบาย Concept การออกแบบ Code	3
อธิบาย Functions และ Flowchart	4
- World	4
- Enemy	8
- Lava , Door	10
- Platform	12
- Button	15
- Player	17
- Draw_Text	20
- Main Game	21
Code Program	24

## อธิบาย Concept ของเกมส์

เกมส์ Super Slime Tournament ได้รับแรงบันดาลใจมาจาก Super Mario bros . ซึ่งเป็นเกมส์ที่คนทั่วไปรู้จักกันและเป็นที่เกมส์ที่เราชอบด้วยในสมัยเด็ก แต่ในเกมส์ของพวกผมที่ได้นำมาดัดแปลงกันจะมีการแข่งขันแบบจับเวลา จัดอันดับ แบบ Tournament อันดับ 1 – 5 ซึ่งในปัจจุบัน ที่พวกผมคิดเกมส์มาในรูปแบบ Tournament ขึ้นมา เพราะมีการแข่ง E-sports ( Electronic sports ) มากขึ้นในปัจจุบัน ซึ่งประเทศไทยเราก็มีบรรจุกีฬา E-sports เป็นกีฬาแล้วเช่นกัน ซึ่งกฎกติกาการแข่งขันก็ง่ายๆ นั่นก็คือ ใครที่สามารถทำเวลาได้ดีที่สุดจะเป็นผู้ชนะ 1 – 5 อันดับแรก

ซึ่งเกมส์นี้จะช่วยฝึกไหวพริบและการคิดของผู้เล่นว่าจะต้องทำอะไรจึงจะพาเจ้าสไลม์ไปถึงหาออกให้ได้ และต้องไปถึงให้ไวที่สุดด้วย ซึ่งมีการทำเวลาแข่งกับผู้เล่นคนอื่นด้วย ซึ่งทำให้เกมส์มีความท้าทายยิ่งขึ้น มาช่วยเจ้าสไลม์ผ่านอุปสรรคและหาทางออกไปพร้อมกันเถอะ

## วิธีการในการเล่นเกมส์

เกม Super Slime Tournament ได้รับแรงบันดาลใจมาจาก Super Mario bros . ซึ่งเป็นเกมส์ที่ทุกคนทั่วไปรู้จักกันและเป็นที่พวกเราชอบด้วยในสมัยเด็ก แต่เกมส์ของพวกเราได้มาดัดแปลงจนเป็นเกมส์ที่มีความสนุกอาจน้อยกว่า Super Mario bros. เพราะพวกผมเน้นการแข่งขันจับเวลาและบันทึกจัดอันดับ 1-5 โดยระยะเวลาการเล่นของเราตั้งแต่ เริ่ม จนถึง ประตูนุ่เส้นชัย ว่าวิ่งไปกี่วินาที

## วิธีการควบคุมตัวละคร

ปุ่ม A : จะเป็นการเดินไป ทางซ้าย

ปุ่ม D : จะเป็นการเดินไป ทางขวา

ปุ่ม Spacebar : จะเป็นการกระโดดข้ามสิ่งกีดขวาง



### อธิบาย Concept การออกแบบ Code

เราจะทำการเขียน แต่ละ Functions เพื่อเรียกใช้เพื่อให้มันไม่สับสน และนำไปเรียกใช้ให้มันแสดงผล แต่ละ Functions บางตัวแสดงผลของวัตถุเช่น Player , Lava , World เพื่อให้มันตอบสนองต่อการนำมารวมกันเป็นเกมส์หนึ่งเกมส์ขึ้นมา Function แต่ละตัวจะมีข้อมูล ที่เหมือนกันจะเรียกมาใช้ เช่น Self , x , y , image Code ที่ผมได้นำมาเขียนรวมๆกัน ได้ IDEA มาจากเกมส์ Mario และศึกษาเพิ่มเติมจากเว็บไซต์ Stack overflow เพื่อนำมาประยุกต์ให้เข้ากับ Code ของเกมส์พวกผม การ run ตัวเกมส์พวกผมจะใช้เป็น Pygame ซึ่งน่าจะง่ายกว่าการใช้ Pgzzero

## อธิบาย Functions และ Flowchart

### Function : World

```
class World():
    def __init__(self, data):
        self.tile_list = []

        #class object
        row_count = 0
        for row in data:
            col_count = 0
            for tile in row:
                if tile == 1:
                    img = pygame.transform.scale(dirt, (tile_size, tile_size))
                    img_rect = img.get_rect()
                    img_rect.x = col_count * tile_size
                    img_rect.y = row_count * tile_size
                    tile = (img, img_rect)
                    self.tile_list.append(tile)
                if tile == 2:
                    img = pygame.transform.scale(grass, (tile_size, tile_size))
                    img_rect = img.get_rect()
                    img_rect.x = col_count * tile_size
                    img_rect.y = row_count * tile_size
                    tile = (img, img_rect)
                    self.tile_list.append(tile)
                if tile == 3:
                    ghost = Enemy(col_count * tile_size, row_count * tile_size)
                    blob_group.add(ghost)
                if tile == 4:
                    lava = Lava(col_count * tile_size, row_count * tile_size + (tile_size//2))
                    lava_group.add(lava)
                if tile == 5:
                    door = Door(col_count * tile_size, row_count * tile_size)
                    door_group.add(door)
                if tile == 6:
                    platform = Platform(col_count * tile_size, row_count * tile_size, 1, 0)
                    platform_group.add(platform)
                if tile == 7:
                    platform = Platform(col_count * tile_size, row_count * tile_size, 0, 1)
                    platform_group.add(platform)

            col_count += 1
            row_count += 1

    def draw(self):
        for tile in self.tile_list:
            screen.blit(tile[0], tile[1])
            #pygame.draw.rect(screen, (255, 255, 255), tile[1], 2)
```

```
# 1= ดิน # 6 = ดินขยับแนวนอน
# 2= หญ้า # 7 = ดินขยับแนวตั้ง
# 3= ศัตรู
# 4= ลาวา
# 5= ประตู
world_data = [
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1],
[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 5, 1],
[1, 0, 0, 0, 0, 0, 6, 0, 0, 0, 6, 0, 0, 2, 1],
[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1],
[1, 0, 7, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1],
[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1],
[1, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1],
[1, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1],
[1, 0, 0, 0, 0, 0, 0, 7, 0, 0, 0, 3, 0, 0, 1],
[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 2, 2, 0, 1],
[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1],
[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 1],
[1, 0, 2, 0, 0, 6, 0, 0, 2, 0, 0, 6, 0, 0, 1],
[1, 2, 1, 4, 4, 4, 4, 4, 1, 4, 4, 4, 4, 4, 1],
]
```

### อธิบาย

เป็นหลักการสร้างโลกภายในเกมของเรา 1 = ดิน 2 = หญ้า 3 = ศัตรู 4 = ลาวา 5 = ประตู 6 = ดินขยับแนวนอน 7 = ดินขยับแนวตั้ง

### หลักการทำงานของ Code

เราจะสร้าง Function ที่ชื่อว่า `def __init__` โดยให้ค่า `self` กับ `data` ผ่านมาได้ นำข้อมูลจาก `self` มาเพิ่มด้วยตัวแปรที่ชื่อว่า `tile_list` ตั้งไว้รอค่า ต่อมา ตั้งตัวแปรที่ชื่อว่า `row_count = 0` เพื่อกำหนดแถวแนวนอน แล้วต่อมาทำ Loop for ใน `data` ต่อมา สร้าง ตัวแปรที่ชื่อว่า `col_count = 0` เพื่อกำหนดแถวแนวตั้ง ต่อมาลงด้วย Loop for `tile` ใน `row` สร้าง if เพื่อที่จะดักจับว่า Object ต่างๆ ยกตัวอย่างเช่น `if tile == 1` ก็คือจะสร้างบล็อกดินขึ้นมา โดยใช้คำสั่ง `pygame.transform.scale ( รูปภาพ , ( ขนาด กว้าง * ยาว )`) ต่อมา สร้างตัวแปรที่ชื่อว่า `img_rect = img.get_rect()` เพื่อที่จะให้ `img` ใช้คำสั่ง `get_rect` ต่อมาสร้าง `img_rect.x` และ `y` เพื่อที่จะกำหนดขนาดของบล็อกที่จะเกิดขึ้นมา ใช้คำสั่งสร้างตัวแปร `tile = ( img , img_rect )` มันจะเซฟค่า

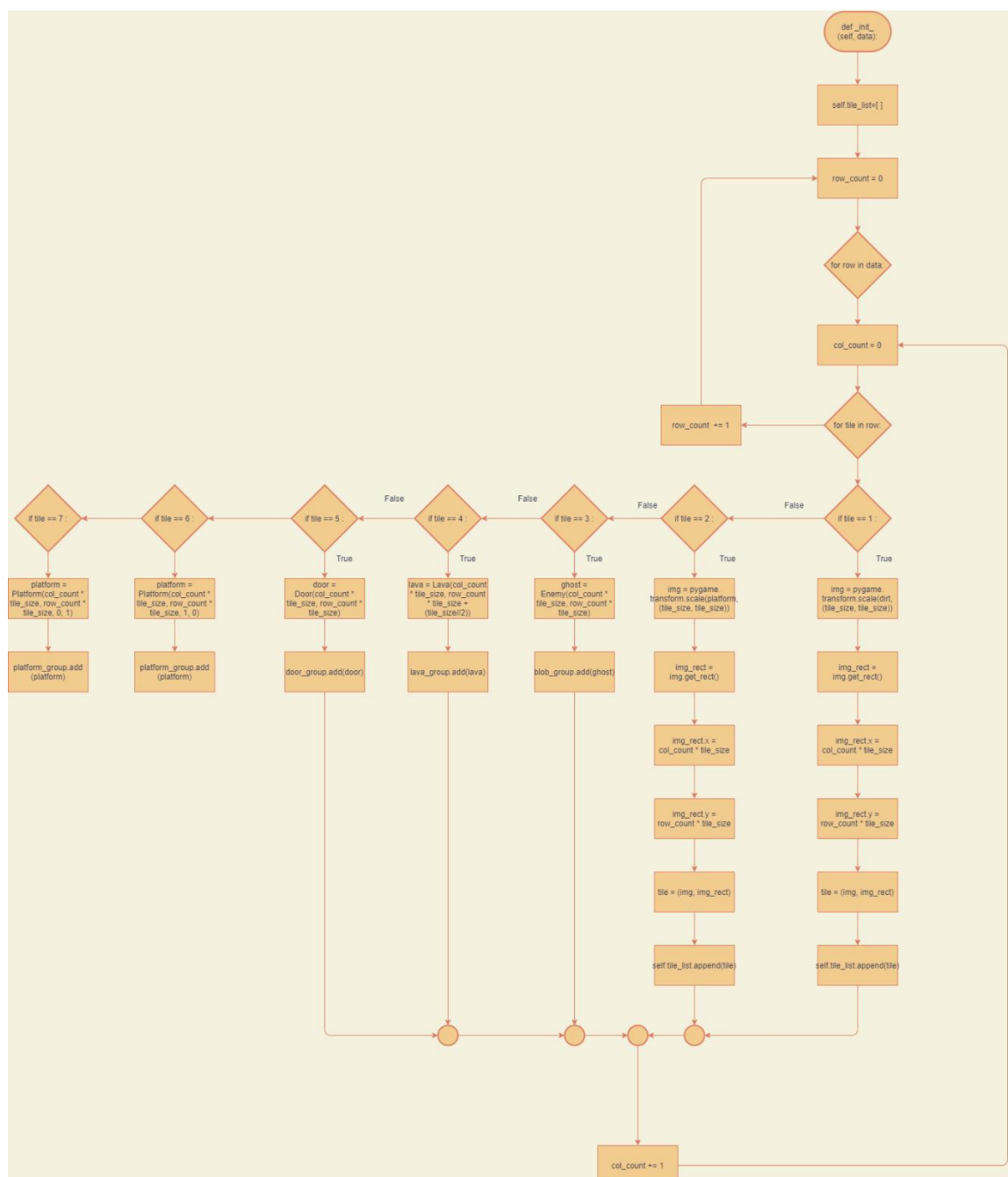


ไปใน self ด้วย จะเป็นคำสั่ง `self.tile_list.append(tile)` ก็คือเอา tile ชัดเข้าไปใน self และ if tile == 2 ก็ทำเช่นกัน ส่วน if tile == 3 – 7 จะเป็นการสร้างตัวแปรของแต่ละอย่างและ แอดเข้าไป

ยกตัวอย่างเช่น

if tile == 3 จะสร้างตัวแปร `ghost = Enemy ( แถวตั้ง * ขนาด , แถวแนวนอน * ขนาด )` จะใช้คำสั่งที่เคยเตรียมเอาไว้ ก็คือ `blob_group.add(ghost)` และทำแบบนี้คล้ายๆกัน จนถึง 3 – 7 จะออกมาจาก if ต่อมาจะบวก `col_count += 1` ออกมาจาก loop for จะบวก `row_count += 1`

### Flowchart : การทำงานของ World



## Function : Enemy

```
class Enemy(pygame.sprite.Sprite):
    def __init__(self, x, y):
        pygame.sprite.Sprite.__init__(self)
        self.image = pygame.image.load('images/ghost.png')
        self.rect = self.image.get_rect()
        self.rect.x = x
        self.rect.y = y
        self.move_direction = 1
        self.move_counter = 0

    def update(self):
        self.rect.x += self.move_direction
        self.move_counter += 1
        if self.move_counter > 50:
            self.move_direction *= -1
            self.move_counter *= -1
```

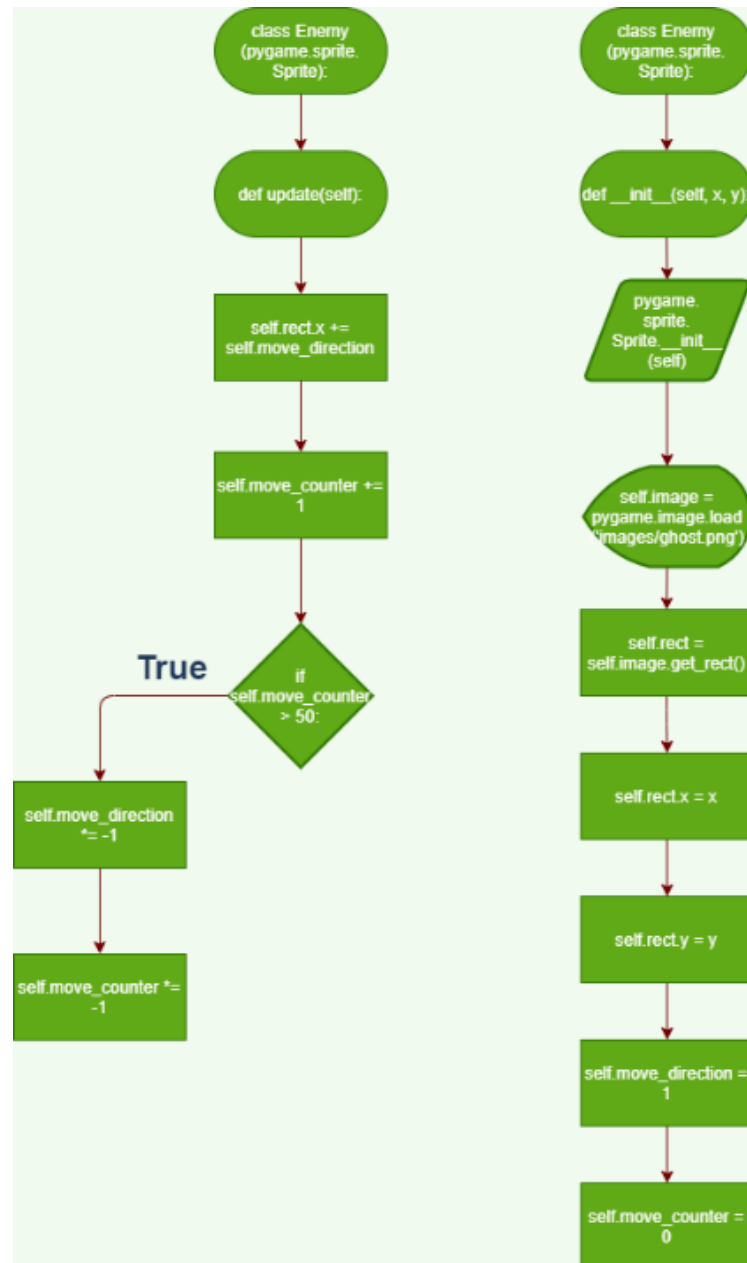
### อธิบาย

เป็นการสร้างศัตรูขึ้นมาในเกมส์เพื่อ เป็นอุปสรรคในการเล่นของเรา

### หลักการทำงานของ Code

จะให้ pygame.sprite.Sprite. ผ่านเข้ามาต่อมาจะสร้าง def \_\_init\_\_ ด้วยโดยที่ ค่า self , x , y ผ่านเข้ามา จะเอาข้อมูลใน self เข้าไปใน pygame.sprite.Sprite โดยผ่าน Function def และจะสร้างตัวแปรที่ชื่อ image = pygame.image.load( รูปภาพ ) ต่อมาสร้างจะตัวแปร self.rect = self.image.get\_rect () สร้าง self .rect.x = รับค่า x และ สร้าง self.rect.y = รับค่า y สร้าง self.move\_direction = 1 และสุดท้ายในฟังก์ชันจะสร้าง self.move\_counter = 0 ต่อมาจะเก็บค่าทุกตัวแปรที่สร้างขึ้นมาไว้ใน self พอเก็บค่าทุกอย่างไว้ใน self ต่อไปจะสร้าง def update โดยที่ค่า self จะผ่านเข้ามาในฟังก์ชัน จะเอา self.rect.x += self.move\_direction ถัดไปจะนำ self.move\_counter += 1 ต่อมาจะเข้าเงื่อนไข if self.move\_counter > 50 : ถ้ามันเข้าเงื่อนไขของ if จะนำ self.move\_direction \*= -1 และ self.move\_counter \*= -1 สุดท้ายจะเก็บค่า self ไว้

### Flowchart : การทำงานของ Enemy



## Function : Lava , Door

### อธิบาย

Code Function Lava และ Door จะเป็นเขียนสร้าง ลาวาที่เป็นอุปสรรคขวางทางกันเราอยู่ และ ประตูจะหมายถึง การเข้าเส้นชัย

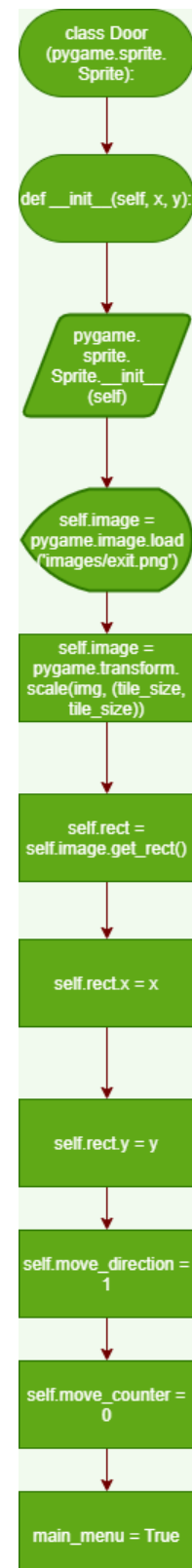
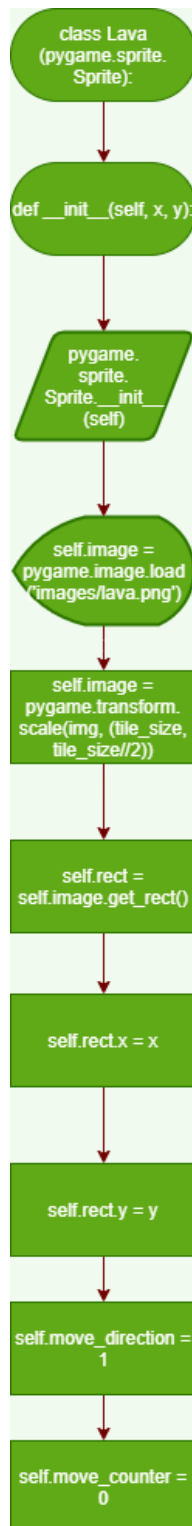
### หลักการทำงานของ Code

โดยทั้ง Lava , Door ฟังก์ชัน จะให้ pygame.sprite.Sprite. ผ่านเข้ามาต่อมาจะสร้าง def \_\_init\_\_ ด้วย โดยที่ ค่า self, x, y ผ่านเข้ามา จะเอาข้อมูลใน self เข้าไปใน pygame.sprite.Sprite โดยผ่าน Function def และจะสร้างตัวแปรที่ชื่อ image = pygame.image.load( รูปภาพ ) โดยฟังก์ชันนี้จะทำงานเหมือนกันกับ Enemy จะแตกต่างกันตรงที่ self.image = pygame.transform.scale ( img , ( ขนาด , ขนาด // 2 ) )

```
class Lava(pygame.sprite.Sprite):
    def __init__(self, x, y):
        pygame.sprite.Sprite.__init__(self)
        img = pygame.image.load('images/lava.png')
        self.image = pygame.transform.scale(img, (tile_size, tile_size//2))
        self.rect = self.image.get_rect()
        self.rect.x = x
        self.rect.y = y
        self.move_direction = 1
        self.move_counter = 0
```

```
class Door(pygame.sprite.Sprite):
    def __init__(self, x, y):
        pygame.sprite.Sprite.__init__(self)
        img = pygame.image.load("images/exit.png")
        self.image = pygame.transform.scale(img, (tile_size, tile_size))
        self.rect = self.image.get_rect()
        self.rect.x = x
        self.rect.y = y
        self.move_direction = 1
        self.move_counter = 0
        main_menu = True
```

### Flowchart : การทำงาน Lava และ Door



## Function : Platform

```
class Platform(pygame.sprite.Sprite):
    def __init__(self, x, y, move_x, move_y):
        pygame.sprite.Sprite.__init__(self)
        img = pygame.image.load('images/platform.png')
        self.image = pygame.transform.scale(img, (tile_size, tile_size // 2))
        self.rect = self.image.get_rect()
        self.rect.x = x
        self.rect.y = y
        self.move_counter = 0
        self.move_direction = 1
        self.move_x = move_x
        self.move_y = move_y

    def update(self):
        self.rect.x += self.move_direction * self.move_x
        self.rect.y += self.move_direction * self.move_y
        self.move_counter += 1
        if abs(self.move_counter) > 50:
            self.move_direction *= -1
            self.move_counter *= -1
```

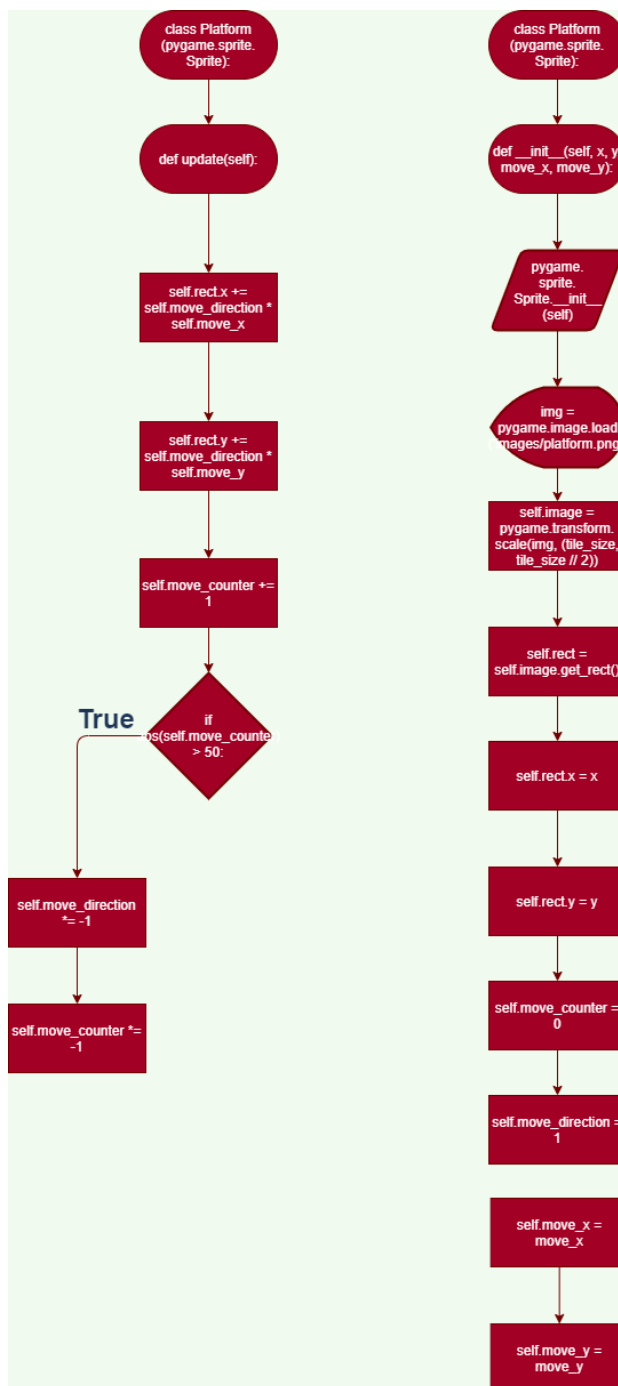
### อธิบาย

เป็นการสร้าง Platform ขยับตามแนวนอนและแนวตั้ง

### หลักการทำงานของ Code

ในฟังก์ชันของ Platform จะทำงานเหมือนกับ Lava และ Door แต่แตกต่างกันที่ def \_\_init\_\_ จะรับค่าเพิ่มเติมมาก็คือ move\_x และ move\_y โดยที่ค่า self.move\_x = move\_x และ self.move\_y = move\_y จะทำการเก็บไว้ที่ค่า self ต่อมาเราจะอยู่ในฟังก์ชัน def update(self) โดยในฟังก์ชันจะมี self.rect.x += self.move\_direction \* self.move\_x และ y จะทำการ self.move\_counter += 1 ถัดไปจะมีเงื่อนไขของ if abs(self.move\_counter) > 50: ถ้ามันจริงมันจะเข้าเงื่อนไขของ if ภายใน if จะมี self.move\_direction \*= -1 และ self.move\_counter \*= -1

## Flowchart : การทำงานของ Platform



## Function : Button

### อธิบาย

เป็นการสร้างปุ่มกด และการกระทำของเราเวลาคลิกเข้าไปในปุ่ม

### หลักการทำงานของ Code

การทำงานของฟังก์ชัน Button ปุ่มกด จะรับค่า self, x, y, image โดยการทำงานของฟังก์ชัน `__init__` จะสร้างตัวแปร `self.image = image` จะทำงานคล้ายๆ กับ Lava, Door, Enemy แต่จะมีเพิ่มมาคือ `self.clicked = False` ถัดไปจะสร้าง ฟังก์ชัน `def draw(self)` ก็คือให้ self ผ่านเข้ามา โดยที่ การทำงานของ ฟังก์ชัน `action = False` ต่อมาจะสร้างตัวแปร `pos = pygame.mouse.getpos()` การกระทำของเมาส์เรา จะเข้าสู่ การเขียน `if self.rect.collidepoint(pos)` มันจะเอาไว้ตรวจว่า เมาส์ ไปทำอะไร ช้างใน `if` จะมีอีกเงื่อนไข `if` คือ `if pygame.mouse.get_pressed()[0] == 1 and self.clicked == False` เราจะเช็คว่าเอาเมาส์ไปวางหรือยัง ถ้า มันจริง `action = True` และ `self.clicked = True` ถัดจากนั้นจะเขียน `if pygame.mouse.get_pressed()[0] == 0` ภายในเงื่อนไขโดยถ้ามันจริง `self.clicked = False` ถัดไปจะเป็นการเขียน Button โดยใช้คำสั่ง `screen.blit(self.image, self.rect)` และจะส่งค่า `return action` กลับไป

```
class Button():
    def __init__(self, x, y, image):
        self.image = image
        self.rect = self.image.get_rect()
        self.rect.x = x
        self.rect.y = y
        self.clicked = False

    def draw(self):
        action = False
        #get cilck
        pos = pygame.mouse.get_pos()

        #check clicked:
        if self.rect.collidepoint(pos):
            if pygame.mouse.get_pressed()[0] == 1 and self.clicked == False:
                action = True
                self.clicked = True

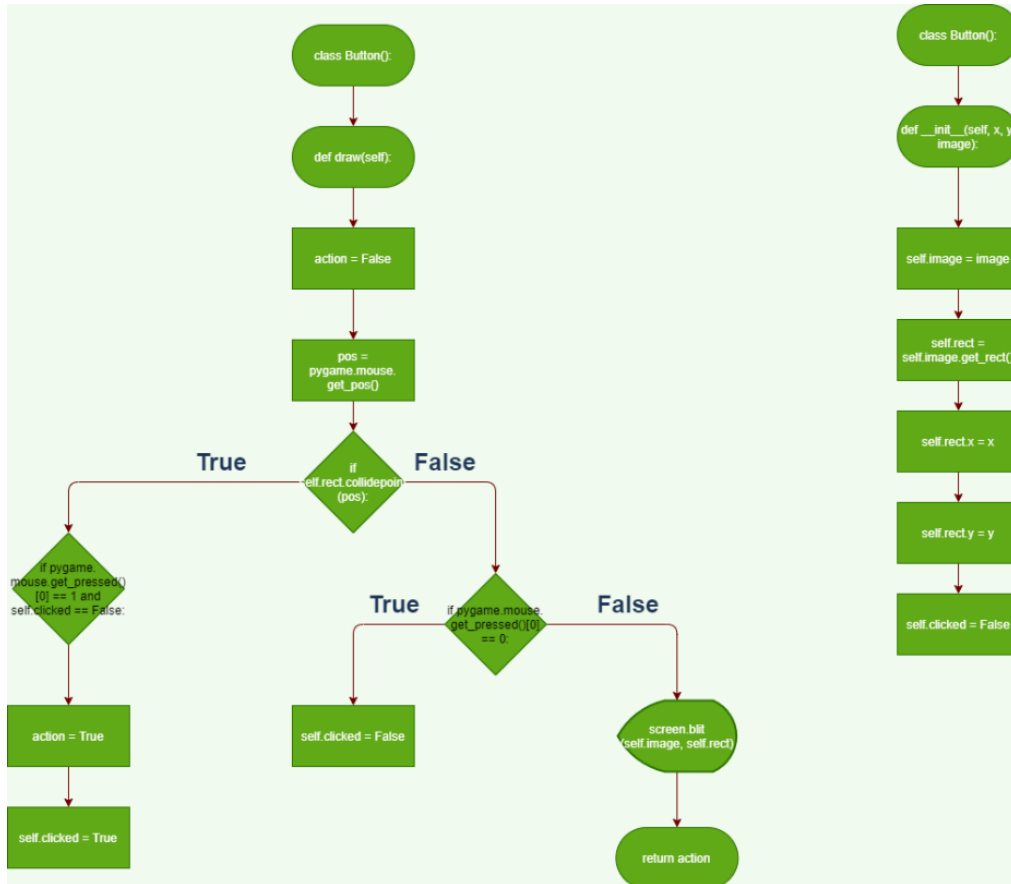
            if pygame.mouse.get_pressed()[0] == 0:
                self.clicked = False

        #draw button
        screen.blit(self.image, self.rect)

        return action
```



## Flowchart : การทำงานของ Button



## Function : Player

```

3  class Player():
4  def __init__(self, x, y):
5      self.reset(x, y)
6
7  def update(self, game_over , ):
8      dx = 0
9      dy = 0
10     col_thresh = 20
11
12     if game_over == 0:
13         # กดปุ่มลัด
14         keys = pygame.key.get_pressed()
15         if keys[pygame.K_a]:
16             dx -= 5
17         if keys[pygame.K_d]:
18             dx += 5
19         if keys[pygame.K_SPACE] and self.jumped == False:
20             self.vel_y = -15
21             self.jumped = True
22         if keys[pygame.K_SPACE] == False and self.in_air == False:
23             self.jumped = False
24
25         self.vel_y += 1
26         if self.vel_y > 10:
27             self.vel_y = 10
28         dy += self.vel_y
29
30         self.in_air = True
31         for tile in world.tile_list:
32             # เดินชนบล็อก
33             if tile[1].colliderect(self.rect.x + dx, self.rect.y, self.width, self.height):
34                 dx = 0
35             # ชนบนบล็อก
36             if tile[1].colliderect(self.rect.x, self.rect.y + dy, self.width, self.height):
37                 if self.vel_y < 0 :
38                     dy = tile[1].bottom - self.rect.top
39                 elif self.vel_y >= 0 :
40                     dy = tile[1].top - self.rect.bottom
41                     self.vel_y = 0
42                     self.in_air = False
43
44         if pygame.sprite.spritecollide(self, blob_group, False):
45             game_over = -1
46
47         if pygame.sprite.spritecollide(self, lava_group, False):
48             game_over = -1
49
50         if pygame.sprite.spritecollide(self, door_group, False):
51             game_over = -2
52
53         for platform in platform_group:
54             #เดินไม่ติด X
55             if platform.rect.colliderect(self.rect.x + dx, self.rect.y, self.width, self.height):
56                 dx = 0
57             #เดินไม่ติด Y
58             if platform.rect.colliderect(self.rect.x , self.rect.y + dy, self.width, self.height):
59                 #ติดใต้ platform
60                 if abs((self.rect.top + dy) - platform.rect.bottom) < col_thresh:
61                     self.vel_y = 0
62                     dy = platform.rect.bottom - self.rect.top
63                 #ติดบน platform
64                 elif abs((self.rect.bottom + dy) - platform.rect.top) < col_thresh:
65                     self.rect.bottom = platform.rect.top - 1
66                     self.in_air = False
67                     dy = 0
68                 if platform.move_x != 0:
69                     self.rect.x += platform.move_direction

```

```

        self.rect.x += dx
        self.rect.y += dy
    elif game_over == -1:
        if self.rect.y > 200:
            self.rect.y -= 5

    screen.blit(self.image, self.rect)
    #pygame.draw.rect(screen, (255, 255, 255), self.rect, 2)

    return game_over
def reset(self, x, y,):
    load_player = pygame.image.load("images/blob.png")
    self.image = pygame.transform.scale(load_player,(45,37))
    self.rect = self.image.get_rect()
    self.rect.x = x
    self.rect.y = y
    self.width = self.image.get_width()
    self.height = self.image.get_height()
    self.vel_y = 0
    self.jumped = False
    self.in_air = True
    mixer.music.play()

```

## อธิบาย

ในฟังก์ชันของ Player จะเกี่ยวกับการควบคุมตัวละครให้เดิน ซ้าย เดิน ขวา กระโดด หลบข้ามอุปสรรคภายในเกมส์ และ จะมีตัวแปรที่เอาไว้ตอบสนองต่อชนบล็อก หรือ ศัตรู ต่างๆ ถ้าสมมติเราตายจะไปในที่เดิม โดยกดปุ่ม Reset ที่เราตั้งไว้

## หลักการทำงานของ Code

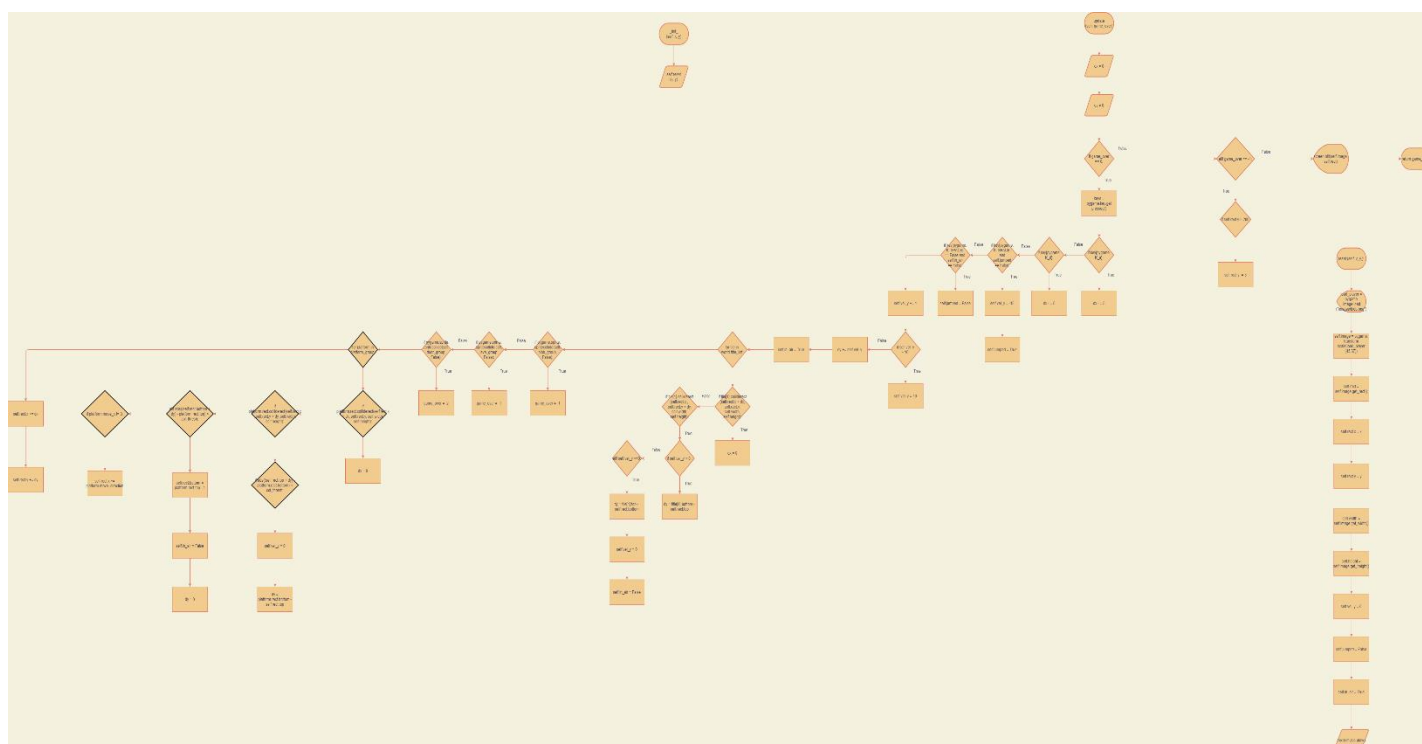
การทำงานของฟังก์ชัน Player ภายในฟังก์ชัน Player จะมีฟังก์ชัน 3 อันคือ `__init__` , `update` และ `reset` ฟังก์ชัน `__init__` จะมีการรับค่า `self` , `x` และ `y` ลงไปในฟังก์ชัน ซึ่งในฟังก์ชันก็จะมี `self.reset ( x , y )` เก็บค่าเอาไว้ ถัดมาจะมาอยู่ในฟังก์ชัน `update` ในฟังก์ชันจะมี

การรับค่า self , game\_over ไว้ ในฟังก์ชันก็จะมี 2 ตัวแปร dx = 0 กับ dy = 0 ต่อมาจะมี คำสั่ง if game\_over == 0: ถ้ามันเป็นจริงจะทำงาน

ซึ่งใน if จะมีตัวแปรที่ชื่อว่า keys = pygame.key.get\_pressed() ถือเป็นคำสั่งที่จะเอาไว้ควบคุมทิศทางตัวละครของเรา ใน if ก็จะมีเด่น ๆ ในการควบคุมก็คือ if keys[pygame.K\_a] เมื่อเรากดตัว A นั้นเองจะเป็นการเดินไปทางซ้าย ซึ่งใน if keys[pygame.K\_a] ก็จะมี dx -= 5 และ if keys[pygame.K\_d] ก็จะทำงานเหมือนกับ if keys[pygame.K\_a] เช่นกันแต่จะแตกต่างที่ dx จะเป็น dx += 5 แทน ส่วน if keys[pygame.K\_space] == False and self.in\_air == False จะเป็นการกระโดด แต่จะมีตัวแปรเพิ่มเติมมาคือ self.jumped = False ตัวแปรถัดไปจะเป็นการทำให้มันกระโดดขึ้นมาตามวิธีรับค่าของการกระโดดก็คือ self.vel\_y += 1 ถ้าเป็นจริงจะเข้าเงื่อนไข if self.vel\_y > 10 ภายในเงื่อนไขก็จะมี self.vel\_y = 10 จะมีตัว self.vel\_y = 10 และข้างนอกของ if จะมีตัวแปร dy += self.vel\_y ถัดไปจะมี 3 เงื่อนไขคือ if pygame.sprite.spritecollide(self, blob\_group, False) , if pygame.sprite.spritecollide(self, lava\_group, False) , if pygame.sprite.spritecollide(self, door\_group, False) จะเป็น สัมผัส เมื่อชนกับ 3 สิ่งนี้ถัดไปจะมาอยู่ใน Loop ของ for platform in platform\_group ก็จะมี if อยู่ 2 ตัวคือ if platform.rect.colliderect(self.rect.x + dx, self.rect.y, self.width, self.height): คือจะเป็นการเดินไม่ติด X และ if platform.rect.colliderect(self.rect.x , self.rect.y + dy, self.width, self.height): จะเป็นการเดินไม่ติด Y ภายใน if ในการเดินไม่ติด Y จะมีคำสั่งติดได้ platform คือคำสั่ง if abs((self.rect.top + dy) - platform.rect.bottom) < col\_thresh: และอีกอันคือการติดบน platform elif abs((self.rect.bottom + dy) - platform.rect.top) < col\_thresh: พอออกมาจาก Loop for แล้วจะเป็นการสร้างตัวแปร 2 ตัวคือ self.rect.x += dx และ self.rect.y += dy ถัดไปจะเป็นการออกจาก คำสั่ง if game\_over == 0 จะเข้าเงื่อนไข elif game\_over == -1 ภายใต้อำนาจ if self.rect.y > 200 และใน if self.rect.y > 200 จะมีเงื่อนไข self.rect.y -= 5 และสุดท้ายก็คือ display ภาพขึ้นจอ เกมส์ screen.blit(self.image, self.rect) ต่อมาจะเป็นฟังก์ชันการทำงานของ def reset(self, x, y)

ฟังก์ชันของ reset จะเป็นเริ่มต้นใหม่ด้วย reset ตัวละครสโตร์ของเรา เมื่อเวลาเราชนกับบางสิ่งที่ทำให้เราตาย ก็จะวนอยู่ในฟังก์ชัน reset ของเรา

### Flowchart : การทำงานของ Player



### Function : draw text

```
def draw_text(text, font, text_col, x, y):
    img = font.render(text, True, text_col)
    screen.blit(img, (x, y))
```

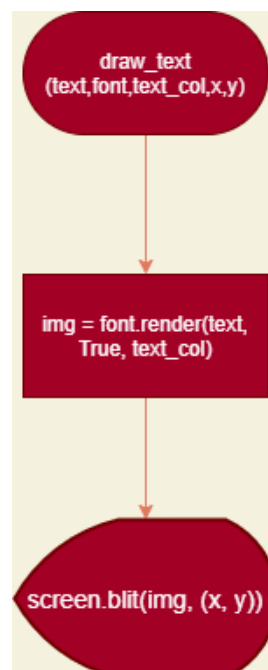
### อธิบาย

เป็นการเขียน ตัวอักษร text ขึ้นบนจอในเกมส์ของเรา

### หลักการทำงานของ Code

ในฟังก์ชัน draw\_text จะมีการรับค่าผ่าน text , font , text\_col , x , y ถัดไปเราจะสร้างตัวแปรที่ชื่อ img = font.render(text , True , text\_col) และสุดท้ายจะทำให้มัน เอาภาพมันขึ้นจอในเกมส์ของเรา

### Flowchart : การทำงานของ draw\_text



## Main Game

```

run = True

#เปิดเกม

while run:

    clock.tick(fps)
    screen.blit(bg, (0, 0))
    if game_over == 0:
        t1 = time.time()
        dt = t1 - t0
        dt = format(dt, '.0f')
        if main_menu == True:
            draw_text('HIGHSORE 1 : ' + list_score[0], font_obj, white, (screen_width // 2) - 150, (screen_height // 2) - 350)
            draw_text('HIGHSORE 2 : ' + list_score[1], font_obj, white, (screen_width // 2) - 150, (screen_height // 2) - 300)
            draw_text('HIGHSORE 3 : ' + list_score[2], font_obj, white, (screen_width // 2) - 150, (screen_height // 2) - 250)
            draw_text('HIGHSORE 4 : ' + list_score[3], font_obj, white, (screen_width // 2) - 150, (screen_height // 2) - 200)
            draw_text('HIGHSORE 5 : ' + list_score[4], font_obj, white, (screen_width // 2) - 150, (screen_height // 2) - 150)
            if exit_button.draw():
                run = False

            if start_button.draw():
                main_menu = False
                t0 = t1

        else:
            world.draw()
            game_over = player.update(game_over)

            if game_over == 0:
                blob_group.update()
                platform_group.update()

            blob_group.draw(screen)
            lava_group.draw(screen)
            door_group.draw(screen)
            platform_group.draw(screen)

```

```

if game_over == -1:

    draw_text('YOU LOSE WITH TIME : ' + (format(dt)), font_obj, red, (screen_width // 2) + -200, (screen_height // 2) - 50)
    mixer.music.stop()

    if restart_button.draw():
        player.reset(100, screen_height - 130)
        game_over = 0
        t0 = t1
        main_menu = True

if game_over == -2:

    draw_text('YOU WIN WITH TIME : ' + (format(dt)), font_obj, font_color, (screen_width // 2) + -200, (screen_height // 2) - 50)

    mixer.music.stop()

    if restart_button.draw():
        player.reset(100, screen_height - 130)
        game_over = 0
        t0 = t1
        main_menu = True
        list_score.append(dt)
        list_score = sorted(list_score)
        open_file3 = open('highscore.txt', 'w')

        for un3 in range(len(list_score)):
            open_file3.write(list_score[un3] + '\n')
        open_file3.close()
        print(list_score)

    count = font_obj.render("Time : " + (format(dt)), 0, font_color)
    screen.blit(count, (300, 50))

    pygame.time.delay(0)

for event in pygame.event.get():
    if event.type == pygame.QUIT:
        run = False

```

```

    keys = pygame.key.get_pressed()
    if keys[pygame.K_f]:
        pygame.display.set_mode((screen_width,screen_height),pygame.FULLSCREEN)
    if keys[pygame.K_ESCAPE]:
        exit()

    pygame.display.update()
pygame.quit()

```

## อธิบาย

ใน Main Game จะเป็นตัวเปิดและ run ตัวเกมส์ Super Slime Tournament ที่จะมีทั้งตัวแสดงจัดอันดับ 5 อันดับที่วิ่งไปที่วินาที และการวิ่งเข้าเส้นชัยหรือติดอุปสรรคในการ วิ่งขึ้นไปถึงเส้นชัยที่เกมส์กำหนดเอา และผลที่เราวิ่งไปที่วินาที พอที่จะจัดอันดับหรือไม่ ใน 5 อันดับ คำสั่งเกมส์ที่พวกผมใช้ในการเขียนเกมส์ตัวนี้ขึ้นมาก็คือ pygame ครับ





## Code Program

```
1  import pygame
2  from pygame.locals import *
3  from pygame import mixer
4  import time
5  import os
6  from os import path
7
8  t0 = time.time()
9  x = 0
10 adwwdasfwess = 0
11 pygame.init()
12
13
14 dt = 0
15 clock = pygame.time.Clock()
16 fps = 60
17
18 screen_width = 800
19 screen_height = 750
20
21 list_score = []
22
23 tile_size = 50
24 game_over = 0
25
26 main_menu = True
27 topscore = []
28
29 second = 1
30 screen = pygame.display.set_mode((screen_width, screen_height))
31
32 pygame.display.set_caption("Super Slime Tournament")
33
34 icon = pygame.image.load("images/blob.png")
35 pygame.display.set_icon(icon)
36 bg = pygame.image.load("images/sky.png")
37 dirt = pygame.image.load("images/dirt.png")
```

```

38 grass = pygame.image.load("images/platform.png")
39 load_player = pygame.image.load("images/blob.png")
40 start_img = pygame.image.load("images/start_btn.png")
41 exit_img = pygame.image.load("images/exit_btn.png")
42 restart_img = pygame.image.load("images/restart_btn.png")
43
44 player = pygame.transform.scale(load_player,(45,37))
45 mixer.music.load("sounds/Corona320bit.wav")
46
47
48
49 font_color=(34,139,34)
50 font_obj=pygame.font.Font("font/2005_iannnnnCPU.ttf",60)
51 font_time = pygame.font.SysFont("font/2005_iannnnnCPU.ttf",60)
52 white = (255, 255, 255)
53 red = (255,0,0)
54
55 √ if os.path.exists('highscore.txt'):
56     open_file = open('highscore.txt')
57     bar = 0
58     √ for un in open_file:
59         bar = bar + 1
60     open_file2 = open('highscore.txt')
61     √ for un2 in range(bar):
62         read_score = open_file2.readline().replace('\n','')
63
64         list_score.append(read_score)
65     list_score = sorted(list_score)
66 √ else:
67     list_score = [0,0,0,0,0]
68
69
70
71 √ def draw_text(text, font, text_col, x, y):
72     img = font.render(text, True, text_col)
73     screen.blit(img, (x, y))
74

```

```

74
75 v def draw_grid():
76 v     for line in range(0, 6):
77         pygame.draw.line(screen, (255, 255, 255), (0, line * tile_size), (screen_width, line * tile_size))
78         pygame.draw.line(screen, (255, 255, 255), (line * tile_size, 0), (line * tile_size, screen_height))
79
80 v class World():
81 v     def __init__(self, data):
82         self.tile_list= []
83
84         #class object
85         row_count = 0
86 v         for row in data:
87             col_count = 0
88 v             for tile in row:
89 v                 if tile == 1:
90                     img = pygame.transform.scale(dirt, (tile_size, tile_size))
91                     img_rect = img.get_rect()
92                     img_rect.x = col_count * tile_size
93                     img_rect.y = row_count * tile_size
94                     tile = (img, img_rect)
95                     self.tile_list.append(tile)
96 v                 if tile == 2:
97                     img = pygame.transform.scale(grass, (tile_size, tile_size))
98                     img_rect = img.get_rect()
99                     img_rect.x = col_count * tile_size
100                    img_rect.y = row_count * tile_size
101                    tile = (img, img_rect)
102                    self.tile_list.append(tile)
103 v                 if tile == 3:
104                     ghost = Enemy(col_count * tile_size, row_count * tile_size)
105                     blob_group.add(ghost)
106 v                 if tile == 4:
107                     lava = Lava(col_count * tile_size, row_count * tile_size + (tile_size//2))
108                     lava_group.add(lava)
109 v                 if tile == 5:
110                     door = Door(col_count * tile_size, row_count * tile_size)
111                     door_group.add(door)

```

```

112
113         if tile == 6:
114             platform = Platform(col_count * tile_size, row_count * tile_size, 1, 0)
115             platform_group.add(platform)
116         if tile == 7:
117             platform = Platform(col_count * tile_size, row_count * tile_size, 0, 1)
118             platform_group.add(platform)
119
120         col_count += 1
121         row_count += 1
122
123     def draw(self):
124         for tile in self.tile_list:
125             screen.blit(tile[0], tile[1])
126             #pygame.draw.rect(screen, (255, 255, 255), tile[1], 2)

```

```

128  ✓ class Enemy(pygame.sprite.Sprite):
129  ✓     def __init__(self, x, y):
130         pygame.sprite.Sprite.__init__(self)
131         self.image = pygame.image.load('images/ghost.png')
132         self.rect = self.image.get_rect()
133         self.rect.x = x
134         self.rect.y = y
135         self.move_direction = 1
136         self.move_counter = 0
137
138  ✓     def update(self):
139         self.rect.x += self.move_direction
140
141         self.move_counter += 1
142  ✓         if self.move_counter > 50:
143             self.move_direction *= -1
144             self.move_counter *= -1
145
146  ✓ class Platform(pygame.sprite.Sprite):
147  ✓     def __init__(self, x, y, move_x, move_y):
148         pygame.sprite.Sprite.__init__(self)
149         img = pygame.image.load('images/platform.png')
150         self.image = pygame.transform.scale(img, (tile_size, tile_size // 2))
151         self.rect = self.image.get_rect()
152         self.rect.x = x
153         self.rect.y = y
154         self.move_counter = 0
155         self.move_direction = 1
156         self.move_x = move_x
157         self.move_y = move_y
158
159  ✓     def update(self):
160         self.rect.x += self.move_direction * self.move_x
161         self.rect.y += self.move_direction * self.move_y
162         self.move_counter += 1
163  ✓         if abs(self.move_counter) > 50:
164             self.move_direction *= -1
165             self.move_counter *= -1

```

```

169 class Lava(pygame.sprite.Sprite):
170     def __init__(self, x, y):
171         pygame.sprite.Sprite.__init__(self)
172         img = pygame.image.load('images/lava.png')
173         self.image = pygame.transform.scale(img, (tile_size, tile_size//2))
174         self.rect = self.image.get_rect()
175         self.rect.x = x
176         self.rect.y = y
177         self.move_direction = 1
178         self.move_counter = 0
179
180 class Door(pygame.sprite.Sprite):
181     def __init__(self, x, y):
182         pygame.sprite.Sprite.__init__(self)
183         img = pygame.image.load("images/exit.png")
184         self.image = pygame.transform.scale(img, (tile_size, tile_size))
185         self.rect = self.image.get_rect()
186         self.rect.x = x
187         self.rect.y = y
188         self.move_direction = 1
189         self.move_counter = 0
190         main_menu = True

```

```

195 # 1= อิฐ # 6 = ดินขยับแนวนอน
196 # 2= หิน # 7 = ดินขยับแนวตั้ง
197 # 3= ศัตรู
198 # 4= ลาวา
199 # 5= ประตู
200 world_data = [
201     [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
202     [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1],
203     [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 5, 1],
204     [1, 0, 0, 0, 0, 6, 0, 0, 0, 6, 0, 0, 0, 2, 2, 1],
205     [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1],
206     [1, 0, 7, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1],
207     [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1],
208     [1, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1],
209     [1, 0, 0, 0, 7, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1],
210     [1, 0, 0, 0, 0, 0, 0, 7, 0, 0, 0, 3, 0, 0, 0, 1],
211     [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 2, 2, 0, 0, 1],
212     [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1],
213     [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 1],
214     [1, 0, 2, 0, 0, 6, 0, 0, 2, 0, 0, 6, 0, 0, 1, 1],
215     [1, 2, 1, 4, 4, 4, 4, 4, 1, 4, 4, 4, 4, 4, 1, 1],
216 ]

```

```

218 class Button():
219     def __init__(self, x, y, image):
220         self.image = image
221         self.rect = self.image.get_rect()
222         self.rect.x = x
223         self.rect.y = y
224         self.clicked = False
225
226     def draw(self):
227         action = False
228         #get click
229         pos = pygame.mouse.get_pos()
230
231         #check clicked:
232         if self.rect.collidepoint(pos):
233             if pygame.mouse.get_pressed()[0] == 1 and self.clicked == False:
234                 action = True
235                 self.clicked = True
236
237             if pygame.mouse.get_pressed()[0] == 0:
238                 self.clicked = False
239
240         #draw button
241         screen.blit(self.image, self.rect)
242
243         return action

```

```

246 class Player():
247     def __init__(self, x, y):
248         self.reset(x, y)
249
250     def update(self, game_over, ):
251         dx = 0
252         dy = 0
253         col_thresh = 20
254
255         if game_over == 0:
256             # คมด้วละคร
257             keys = pygame.key.get_pressed()
258             if keys[pygame.K_a]:
259                 dx -= 5
260             if keys[pygame.K_d]:
261                 dx += 5
262             if keys[pygame.K_SPACE] and self.jumped == False:
263                 self.vel_y = -15
264                 self.jumped = True
265             if keys[pygame.K_SPACE] == False and self.in_air == False:
266                 self.jumped = False
267
268             self.vel_y += 1
269             if self.vel_y > 10:
270                 self.vel_y = 10
271             dy += self.vel_y
272
273             self.in_air = True
274             for tile in world.tile_list:
275                 # เดินชนบล็อก
276                 if tile[1].colliderect(self.rect.x + dx, self.rect.y, self.width, self.height):
277                     dx = 0
278                 # ขึ้นชนบล็อก
279                 if tile[1].colliderect(self.rect.x, self.rect.y + dy, self.width, self.height):
280                     if self.vel_y < 0 :

```

```

281         dy = tile[1].bottom - self.rect.top
282     elif self.vel_y >= 0 :
283         dy = tile[1].top - self.rect.bottom
284         self.vel_y = 0
285         self.in_air = False
286
287     if pygame.sprite.spritecollide(self, blob_group, False):
288         game_over = -1
289
290     if pygame.sprite.spritecollide(self, lava_group, False):
291         game_over = -1
292
293     if pygame.sprite.spritecollide(self, door_group, False):
294         game_over = -2
295
296     for platform in platform_group:
297         #เดินไม่ติด X
298         if platform.rect.colliderect(self.rect.x + dx, self.rect.y, self.width, self.height):
299             dx = 0
300         #เดินไม่ติด Y
301         if platform.rect.colliderect(self.rect.x, self.rect.y + dy, self.width, self.height):
302             #ติดได้ platform
303             if abs((self.rect.top + dy) - platform.rect.bottom) < col_thresh:
304                 self.vel_y = 0
305                 dy = platform.rect.bottom - self.rect.top
306             #ติดบน platform
307             elif abs((self.rect.bottom + dy) - platform.rect.top) < col_thresh:
308                 self.rect.bottom = platform.rect.top - 1
309                 self.in_air = False
310                 dy = 0
311             if platform.move_x != 0:
312                 self.rect.x += platform.move_direction

```

```

315         self.rect.x += dx
316         self.rect.y += dy
317     elif game_over == -1:
318         if self.rect.y > 200:
319             self.rect.y -= 5
320
321
322     screen.blit(self.image, self.rect)
323     #pygame.draw.rect(screen, (255, 255, 255), self.rect, 2)
324
325     return game_over
326 def reset(self, x, y):
327     load_player = pygame.image.load("images/blob.png")
328     self.image = pygame.transform.scale(load_player, (45, 37))
329     self.rect = self.image.get_rect()
330     self.rect.x = x
331     self.rect.y = y
332     self.width = self.image.get_width()
333     self.height = self.image.get_height()
334     self.vel_y = 0
335     self.jumped = False
336     self.in_air = True
337     mixer.music.play()
338
339
340 player = Player(100, screen_height - 130)
341
342 platform_group = pygame.sprite.Group()
343 blob_group = pygame.sprite.Group()
344 lava_group = pygame.sprite.Group()
345 door_group = pygame.sprite.Group()
346
347 world = World(world_data)

```



```

349 #ทำ button
350 restart_button = Button(screen_width // 2-70, screen_height // 2+100, restart_img)
351 start_button = Button(screen_width // 2 -350, screen_height // 2 , start_img)
352 exit_button = Button(screen_width // 2 + 100, screen_height // 2 , exit_img)

361 run = True
362
363 #เปิดเกม
364
365 while run:
366     clock.tick(fps)
367     screen.blit(bg, (0, 0))
368     if game_over == 0:
369         t1 = time.time()
370         dt = t1 - t0
371         dt = format(dt, '.0f')
372     if main_menu == True:
373         draw_text('HIGHSCORE 1 : ' + list_score[0], font_obj, white, (screen_width // 2) - 150, (screen_height // 2) - 350)
374         draw_text('HIGHSCORE 2 : ' + list_score[1], font_obj, white, (screen_width // 2) - 150, (screen_height // 2) - 300)
375         draw_text('HIGHSCORE 3 : ' + list_score[2], font_obj, white, (screen_width // 2) - 150, (screen_height // 2) - 250)
376         draw_text('HIGHSCORE 4 : ' + list_score[3], font_obj, white, (screen_width // 2) - 150, (screen_height // 2) - 200)
377         draw_text('HIGHSCORE 5 : ' + list_score[4], font_obj, white, (screen_width // 2) - 150, (screen_height // 2) - 150)
378         if exit_button.draw():
379             run = False
380
381         if start_button.draw():
382             main_menu = False
383             t0 = t1
384
385     else:
386         world.draw()
387         game_over = player.update(game_over)
388
389         if game_over == 0:
390             blob_group.update()
391             platform_group.update()
392
393         blob_group.draw(screen)
394         lava_group.draw(screen)
395         door_group.draw(screen)
396         platform_group.draw(screen)
397
398         if game_over == -1:
399             draw_text('YOU LOSE WITH TIME : ' + (format(dt)), font_obj, red, (screen_width // 2) + -200, (screen_height // 2)- 50)
400             mixer.music.stop()
401
402             if restart_button.draw():
403                 player.reset(100, screen_height - 130)
404                 game_over = 0
405                 t0 = t1
406                 main_menu = True
407
408         if game_over == -2:
409             draw_text('YOU WIN WITH TIME : ' + (format(dt)), font_obj, font_color, (screen_width // 2) + -200, (screen_height // 2)- 50)
410
411             mixer.music.stop()
412
413             if restart_button.draw():
414                 player.reset(100, screen_height - 130)
415                 game_over = 0
416                 t0 = t1
417                 main_menu = True
418                 list_score.append(dt)
419                 for un4 in range(bar):
420                     list_score[un4] = int(list_score[un4])

```

```

433     list_score = sorted(list_score)
434     for un4 in range(bar):
435         list_score[un4] = str(list_score[un4])
436     open_file3 = open('highscore.txt','w')
437
438
439     for un3 in range(bar):
440         open_file3.write(list_score[un3]+'\\n')
441     open_file3.close()
442     #print(list_score)
443
444
445     count = font_obj.render("Time : "+(format(dt)),0,font_color)
446     screen.blit(count,(300,50))
447
448
449     pygame.time.delay(0)
450
451     for event in pygame.event.get():
452         if event.type == pygame.QUIT:
453             run = False
454     keys = pygame.key.get_pressed()
455     if keys[pygame.K_f]:
456         pygame.display.set_mode((screen_width,screen_height),pygame.FULLSCREEN)
457     if keys[pygame.K_ESCAPE]:
458         exit()
459
460     pygame.display.update()
461     pygame.quit()
462

```