

**เกม Super Slime Tournament**

**นาย บัณณพนต์ ตาคำวัน**

**นาย จักรพรรดิ จ๊อดดวงจันทร์**

**นาย วัชรากร เย็นทวีทรัพย์**

**อาจารย์ที่ปรึกษา**

**อาจารย์ อนิราช มิ่งขวัญ**

โครงงานเล่มนี้เป็นส่วนหนึ่งของการศึกษาตาม หลักสูตรวิศวกรรมศาสตรบัณฑิต

สาขาวิชา วิศวกรรมสารสนเทศและเครือข่าย

คณะเทคโนโลยีและการจัดการอุตสาหกรรม

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

**หัวข้อโครงงาน** เกมส์ Super Slime Tournament

**ชื่อผู้เขียน** นาย บัณณพนต์ ตาคำวัน

นาย จักรพรรดิ จ๊อดดวงจันทร์

นาย วัชรากร เย็นทวีทรัพย์

**อาจารย์ที่ปรึกษา** อาจารย์ อนิราช มิ่งขวัญ

**สาขาวิชา** วิศวกรรมสารสนเทศและเครือข่าย

**ปีการศึกษา** 2564

**บทคัดย่อ**

โครงงานนี้มีวัตถุประสงค์ เพื่อให้ผู้เล่นได้รู้จักรการแข่งขัน 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**

โครงงานนี้สำเร็จลุล่วงได้ด้วยความกรุณาจาก อาจารย์อนิราช มิ่งขวัญ ที่ได้ให้มาเป็นอาจารย์ที่ปรึกษาโครงงานที่ได้ให้คำเสนอแนะแนวคิด ตลอดจนแก้ไขข้อบกพร่องต่างๆ มาโดยตลอดจนโครงงานเล่มนี้เสร็จสมบูรณ์ ผู้ศึกษาจึงกราบขอบพระคุณเป็นอย่างสูง

ขอบคุณเพื่อนๆ ที่ช่วยสนับสนุนในการทำโครงงานนี้ที่ได้ให้ความร่วมมือในการแสดงความคิดเห็นจนทำให้โครงงานสำเร็จลุล่วงไปได้ด้วยดี

ขอบคุณ เว็บไซต์ Stack Overflow และ Youtube ช่อง KidsCanCode ที่ให้คำแนะนำและความรู้ในการพลิกแพลงออกแบบเกมส์

บัณณพนต์ ตาคำวัน

จักรพรรดิ จ๊อดดวงจันทร์

วัชรากร 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

**1**

**อธิบาย Concept ของเกมส์**

เกม Super Slime Tournament ได้รับแรงบันดาลใจมาจาก Super Mario bros . ซึ่งเป็นเกมส์ที่คนทั่วไปรู้จักกันและเป็นที่เกมส์พวกเราชอบด้วยในสมัยเด็ก แต่ในเกมส์ของพวกผมที่ได้นำมาดัดแปลงกันจะมีการแข่งขันแบบจับเวลา จัดอันดับ แบบ Tournament อันดับ 1 – 5 ซึ่งในปัจจุบัน ที่พวกผมคิดเกมส์มาในรูปแบบ Tournament ขึ้นมา เพราะมีการแข่ง E-sports ( Electronic sports ) มากขึ้นในปัจจุบัน ซึ่งประเทศไทยเราก็บรรจุกีฬา E-sports เป็นกีฬาแล้วเช่นกัน ซึ่งกฎกติกาการแข่งขันก็ง่ายๆ นั้นก็คือ ใครที่สามารถทำเวลาได้ดีที่สุดจะเป็นผู้ชนะ 1 – 5 อันดับแรก

ซึ่งเกมนี่จะช่วยฝึกไหวพริบและการคิดของผู้เล่นว่าจะต้องทำอย่างไรจึงจะพาเจ้าสไลม์ไปถึงหาออกให้ได้และต้องไปถึงให้ไวที่สุดด้วย ซึ่งมีการทำเวลาแข่งกับผู้เล่นคนอื่นด้วย ซึ่งทำให้เกมมีความท้าทายมายิ่งขึ้น

มาช่วยเจ้าสไลม์ผ่านอุปสรรคและหาทางออกไปพร้อมกันเถอะ

**2**

**วิธีการในการเล่นเกมส์**

เกม Super Slime Tournament ได้รับแรงบันดาลใจมาจาก Super Mario bros . ซึ่งเป็นเกมส์ที่พวกคนทั่วไปรู้จักรกันและเป็นที่พวกเราชอบด้วยในสมัยเด็ก แต่เกมส์ของพวกได้มาดัดแปลงจนเป็นเกมส์ที่มีความสนุกอาจน้อยกว่า Super Mario bros. เพราะพวกผมเน้นการแข่งขันจับเวลาและบันทึกจัดอันดับ 1-5 โดยระบุเวลาการเล่นของเราตั้งแต่ เริ่ม จนถึง ประตูเส้นชัย ว่าวิ่งไปกี่วินาที

**วิธีการควบคุมตัวละคร**

ปุ่ม A : จะเป็นการเดินไป ทางซ้าย

ปุ่ม D : จะเป็นการเดินไป ทางขวา

A picture containing graphical user interface

Description automatically generatedปุ่ม Spacebar : จะเป็นการกระโดดข้ามสิ่งกีดขวาง

**3**

**อธิบาย Concet การออกแบบ Code**

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

**4**

**อธิบาย Functions และ Flowchart**

**Function : World**

Text

Description automatically generatedBackground pattern

Description automatically generated

**อธิบาย**

เป็นหลักการสร้างโลกภายในเกมส์ของพวกเรา 1 = อิฐ 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 เพื่อกำหนดแถวแนวตั้งต่อมาลงด้วย Loop for tile ใน row สร้าง if เพื่อที่จะดักจับว่า Object ต่างๆ ยกตัวอย่างเช่น if tile == 1 ก็คือจะสร้างบล็อกดินขึ้นมา โดยใช้คำสั่ง pygame.tranform.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

**7**

ไปใน self ด้วย จะเป็นคำสั่ง self.tile\_list.append(tile) ก็คือเอา tile ยัดเข้าไปใน 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**

A picture containing diagram

Description automatically generated

**8**

**Function : Enemy**

**Text

Description automatically generated**

**อธิบาย**

เป็นการสร้างศัตรูขึ้นมาในเกมส์เพื่อ เป็นอุปสรรคในการเล่นของเรา

**หลักการทำงานของ Code**

จะให้ pygame.sprtie.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

**9**

**Flowchart : การทำงานของ Enemy**

**Diagram

Description automatically generated**

**10**

**Function : Lava , Door**

**อธิบาย**

Code Function Lava และ Door จะเป็นเขียนสร้าง ลาวาที่เป็นอุปสรรคขวางทางกันเราอยู่ และ ประตูจะหมายถึง การเข้าเส้นชัย

**หลักการทำงานของ Code**

โดยทั้ง Lava , Door ฟังก์ชัน จะให้ pygame.sprtie.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 ) โดยฟังก์ชันนี้จะทำงานเหมือนกันกับ Enemy จะแตกต่างกันตรงที่ self.image = pygame.tranform.scale ( img , ( ขนาด , ขนาด // 2 ))

**A screenshot of a computer

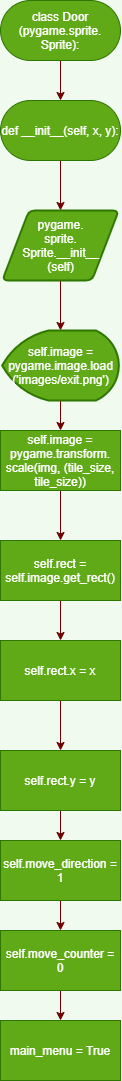
Description automatically generated with medium confidence**

Text

Description automatically generated

**11**

**Flowchart : การทำงาน Lava และ Door**

****

**12**

**Function : Platform**

**Text

Description automatically generated**

**อธิบาย**

เป็นการสร้าง Platform ขยับตามแนวนอนและแนวตั้ง

**หลักการทำงานของ Code**

ในฟังก์ชันของ Platform จะทำงานเหมือนกับ Lava และ Door แต่แตกต่างตรงที่ 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 ถัดไปจะมีเงื่อนไข ของ if abs(self.move\_counter) >50: ถ้ามันจริงมันจะเข้าเงื่อนไขของ if ภายใน if จะมี self.move\_direction \*= -1 และ self.move\_counter \*= -1

**14**

**Flowchart : การทำงานของ Platform**

**Diagram

Description automatically generated**

**15**

**Function : Button**

**อธิบาย**

เป็นการสร้างปุ่มกด และการกระทำของเราเวลาคลิกเข้าไปในปุ่ม

**หลักการทำงานของ Code**

การทำงานของฟังก์ชัน Button ปุ่มกด จะรับค่า self , x , y, image โดยการทำงานฟังก์ชัน \_\_init\_\_

จะสร้างตัวแปร self.image = image จะทำงานคล้ายๆ กับ Lava , Door , Enemy แต่จะมีเพิ่มมาคือ self.clicked = False ถัดไปจะสร้าง ฟังก์ชัน def draw( self ) ก็คือให้ self ผ่านเข้ามา โดยที่ การทำงานของฟังก์ชัน action = False ต่อมาจะสร้างตัวแปร pos = pygame.mouse.getpos() การกระทำของเมาส์เรา จะเข้าสู่การเขียน if self.rect.collidepoint(pos) มันจะเอาไว้ตรวจว่า เมาส์ ไปทำอะไร ข้างใน if จะมีอีกเงื่อนไข if คือ

if pygame.mouse.get\_pressed()[0] == 1 and self.clicked == False เราจะเช็คว่าเอาเมาส์ไปวางหรือยัง ถ้ามันจริง 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

**Text

Description automatically generated**

**16**

**Flowchart : การทำงานของ Button**

**Diagram

Description automatically generated**

**17**

**Function : Player**

**Text

Description automatically generated**

**18**

**Text

Description automatically generated**

**อธิบาย**

ในฟังก์ชันของ Player จะเกี่ยวกับการควบคุมตัวละครให้เดิน ซ้าย เดิน ขวา กระโดด หลบข้ามอุปสรรคภายในเกมส์ และ จะมีตัวแปรที่เอาไว้ตอบสนองต่อชนบล็อก หรือ ศัตรู ต่างๆ ถ้าสมมติเราตายจะไปเกิดในที่เดิม 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

**19**

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

ซึ่งใน if จะมีตัวแปรที่ชื่อว่า keys = pygame.key.get\_pressed() คือเป็นคำสั่งที่จะเอาไว้ควบคุมทิศทางตัวละครของเรา 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 อยุ่ 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 จะมีคำสั่งติดใต้ platform คือคำสั่ง if abs((self.rect.top + dy) - platform.rect.bottom) < col\_thresh: และอีกอันคือการติดบน platfrom 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 ภายใต้คำสั่ง if self.rect.y > 200 และใน if self.rect.y > 200 จะมีเงื่อนไข self.rect.y -= 5 และสุดท้ายก็คือ display ภาพขึ้นจอเกมส์ screen.blit(self.image,self.rect) ต่อมาจะเป็นฟังก์ชันการทำงานของ def reset(self,x,y)

**19**

ฟังก์ชันของ reset จะเป็นเริ่มต้นใหม่ด้วย reset ตัวละครสไลม์ของเรา เมื่อเวลาเราชนกับบ้างสิ่งที่ทำให้เราตาย ก็จะวนอยู่ในฟังก์ชัน reset ของเรา

**Flowchart : การทำงานของ Player**

**Chart

Description automatically generated**

**20**

**Function : draw text**

**Text

Description automatically generated**

**อธิบาย**

เป็นการเขียน ตัวอักษร text ขึ้นบนจอในเกมส์ของเรา

**หลักการทำงานของ Code**

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

**Flowchart : การทำงานของ draw\_text**

Diagram

Description automatically generated

**21**

**Main Game**

Text

Description automatically generated

Text

Description automatically generated

**22**

**Text

Description automatically generated**

**อธิบาย**

ใน Main Game จะเป็นตัวเปิดและ run ตัวเกมส์ Super Slime Tournament ที่จะมีทั้งตัวแสดงจัดอันดับ 5 อันดับที่วิ่งไปกี่วินาที และการวิ่งเข้าเส้นชัยหรือติดอุปสรรคในการ วิ่งขึ้นไปถึงเส้นชัยที่เกมส์กำหนดเอา และผลที่เราวิ่งไปกี่วินาที พอที่จะจัดอันดับหรือไม่ ใน 5 อันดับ คำสั่งเกมส์ที่พวกผมใช้ในการเขียนเกมส์ตัวนี้ขึ่นมาก็คือ pygame ครับ

**23**

**Flowchart : การทำงานของ Main Game**

**Diagram, schematic

Description automatically generated**

**24**

**Code Program**

Text

Description automatically generated

**25**

Text

Description automatically generated

**26**

Text

Description automatically generated

Text

Description automatically generated

**27**

**Text

Description automatically generated**

**Text

Description automatically generated**

**28**

**Text

Description automatically generated**

**Background pattern

Description automatically generated**

**29**

**Text

Description automatically generated**

**Text

Description automatically generated**

**30**

**Text

Description automatically generated**

**Text

Description automatically generated**

**31**

**Text

Description automatically generated**

**Text

Description automatically generated**

**Text

Description automatically generated**

**32**

**Text

Description automatically generated**