

โครงงาน

Monster Hunter

จัดทําโดย

6604062636241 ธนโชติ สุนทรกำจรพานิช

เสนอ

ผู้ช่วยศาสตราจารย์ สถิตย์ ประสมพันธ์

รายงานนี้เป็นส่วนหนึ่งของวิชา 040613204 Object-Oriented Programming

ภาควิชาวิทยาการคอมพิวเตอร์และสารสนเทศ คณะวิทยาศาสตร์ประยุกต์

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

ภาคเรียนที่ 1 ปีการศึกษา 2567

**บทที่ 1 บทนำ**

**ที่มาและความสำคัญ**

โครงงานนี้จัดขึ้นเพื่อวัดความรู้ในการเรียนวิชา Object Oriented Programming โดยการนําเรื่องที่เรียนมาสร้างเป็นชิ้นงานในรูปแบบของเกมโดยใช้แนวคิดการเขียนโปรแกรมแบบเชิงวัตถุ

**ประเภทของโครงการ**

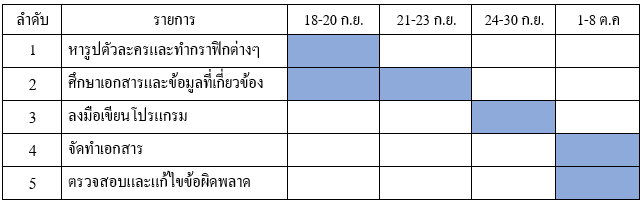
เกม 2 มิติ

**ประโยชน์**

1.ฝึกความแม่นยำ

2.ฝึกไหวพริบ

3.นำเนื้อหาที่เรียนมาประยุกต์ใช้

**ขอบเขตของโครงการ**

**บทที่ 2 ส่วนการพัฒนา**

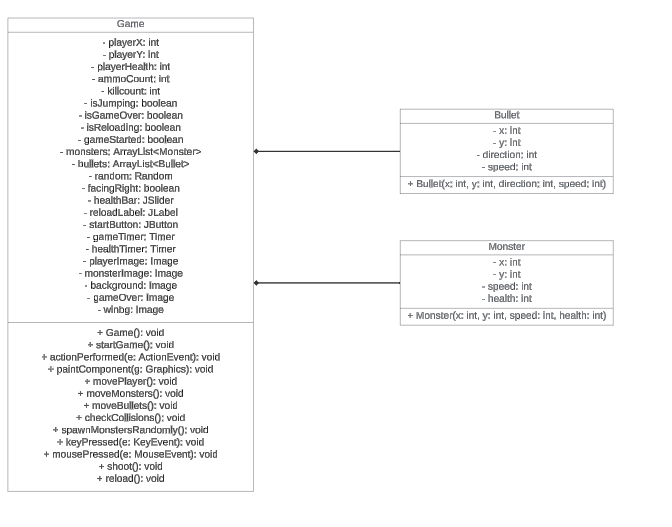
**เนื้อเรื่องย่อ**

ผู้เล่นสวมบทบาทเป็นตัวละครผู้ที่มีความสามารถและได้รับเลือกให้ปกป้องมนุษยชาติและต้องเผชิญหน้ากับมอนสเตอร์โดยการผจญภัยไปในเมืองที่ถูกทำลาย

**วิธีการเล่น**

ใช้ปุ่ม A,W,D เพื่อใช้ในทำให้ตัวละครเดินไป ซ้าย กระโดด และ ขวา ตามลำดับ ใช้ปุ่มคลิกซ้ายของเมาส์เพื่อทำการยิงปืนและปุ่ม R เพื่อ Reload กระสุนปืน โดยต้องยิงมอนเตอร์ให้หมดจึงจะชนะเกม

**คลาสไดอะแกรม**



**รูปแบบการพัฒนา**

-ภาษาที่ใช้ Java พัฒนาแบบ Java Application

**Constructor**

Constructor ของคลาสนี้จะทำการกำหนดค่าพื้นฐานของสถานะเกมและส่วนประกอบของ GUI โดยทำการตั้งค่าต่างๆ และ Listeners ดังนี้



-กำหนด Image ต่างๆในเกม

-ตั้ง Focusable และ preferred size

-เพิ่ม KeyListener MouseListenter สำหรับคีย์บอร์ดและเมาส์

-สร้าง healthBar และ reloadLabel ทำการกำหนดค่าต่างๆ

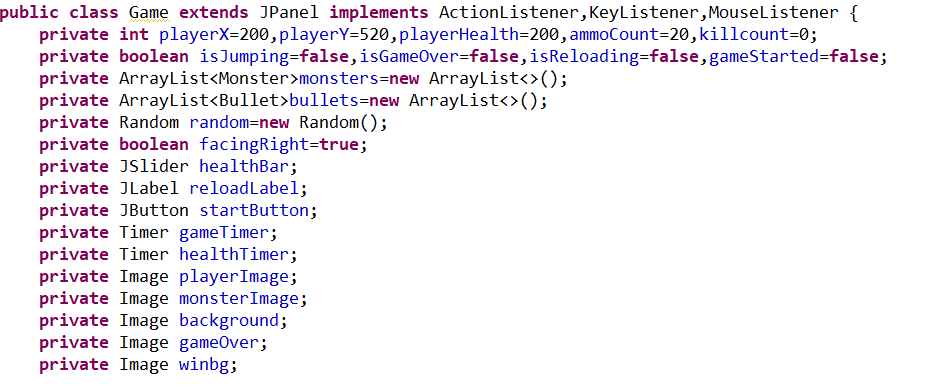
-สร้าง startButton เป็นปุ่มเริ่มเกม

-เริ่ม gameTimer สำหรับ Timer หลักของเกมและ healthTimer สำหรับ Timer ของเลือดผู้ เล่น

-เรียกใช้ spawnMonstersRandomly() เพื่อสุ่มจุดเกิดและสร้างมอนสเตอร์ครั้งแรกในเกม

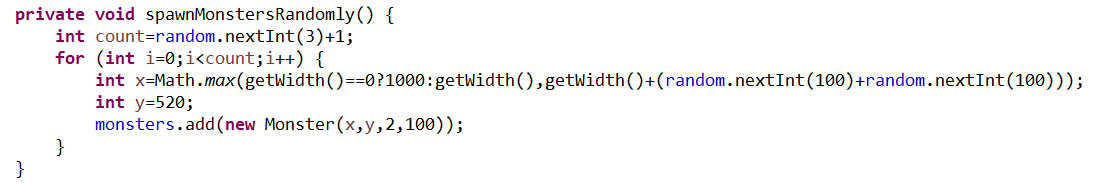
**Encapsulation**

Encapsulation ทำให้ข้อมูลอยู่ภายในคลาสและเข้าถึงได้ผ่านเมธอด โดยใช้ตัวแปร private เพื่อไม่ให้เข้าถึงได้จากภายนอก



**-**Data Field ต่างๆ playerX, playerY, playerHealth, ammoCount ที่เป็น private



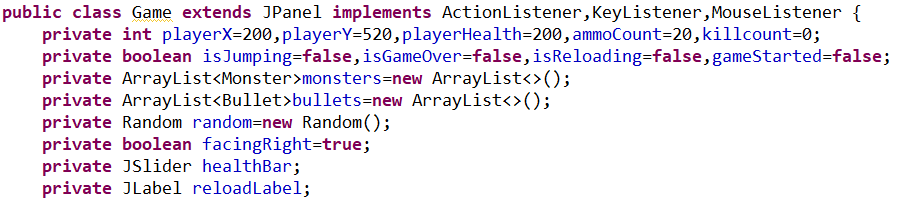




-การเข้าถึงและการเปลี่ยนค่าของตัวแปรที่ทำผ่านเมธอด เช่น movePlayer() moveMonsters() moveBullets()checkCollisions() spawnMonstersRandomly() shoot() reload()

**Composition**

Composition มีให้เห็นโดย Object ของคลาสอื่นๆ เป็น Field ของคลาสของเกมนี้



**-**ArrayList<Monster> และ ArrayList<Bullet> ใช้สำหรับการเก็บมอนสเตอร์และกระสุน

-JSlider (healthBar) และ JLabel (reloadLabel) เป็นส่วนประกอบของ GUI เมื่อ JFrame ถูกปิดจะทำให้ JSlider (healthBar) และ JLabel (reloadLabel) หายไปด้วย

**Polymorphism**

โค้ดในเกมนี้ไม่ได้มีการใช้งาน Polymorphism

**Abstract**

โค้ดในเกมนี้ไม่ได้มีการใช้งาน abstract คลาส หรือ abstract เมธอด

**Inheritance**

คลาสนี้ทำการ extends JPanel เลยเป็นการ Inheritance



-การInheritance จาก JPanel ของในคลาสนี้และทำการ override เมธอด paintComponent(Graphics g) เพื่อทำการวาด

**ส่วนประกอบของ GUI**

GUI ประกอบด้วยส่วนประกอบดังนี้

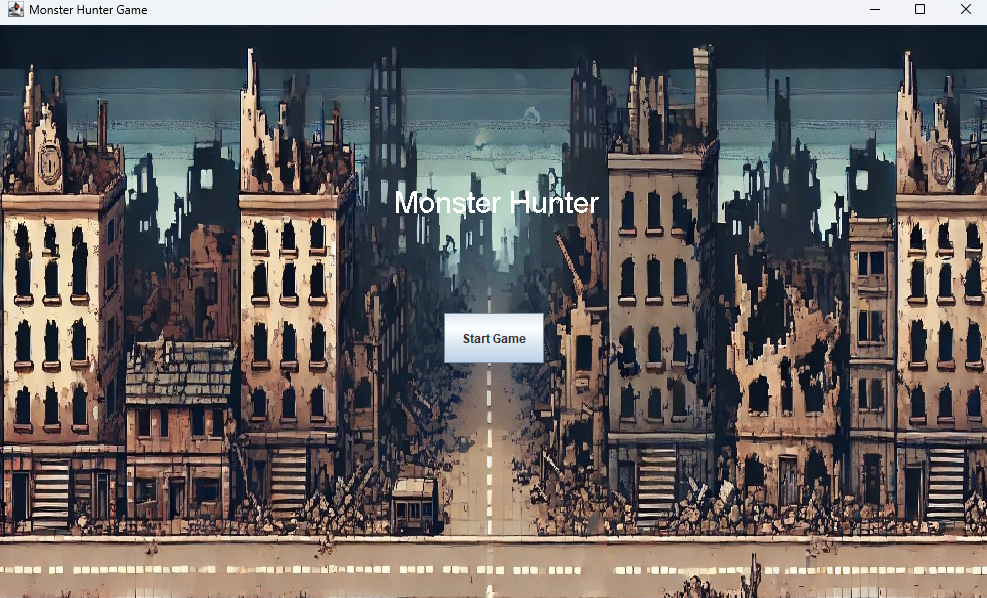
**-JPanel** เป็น panel ที่กำหนดเองของเกมนี้

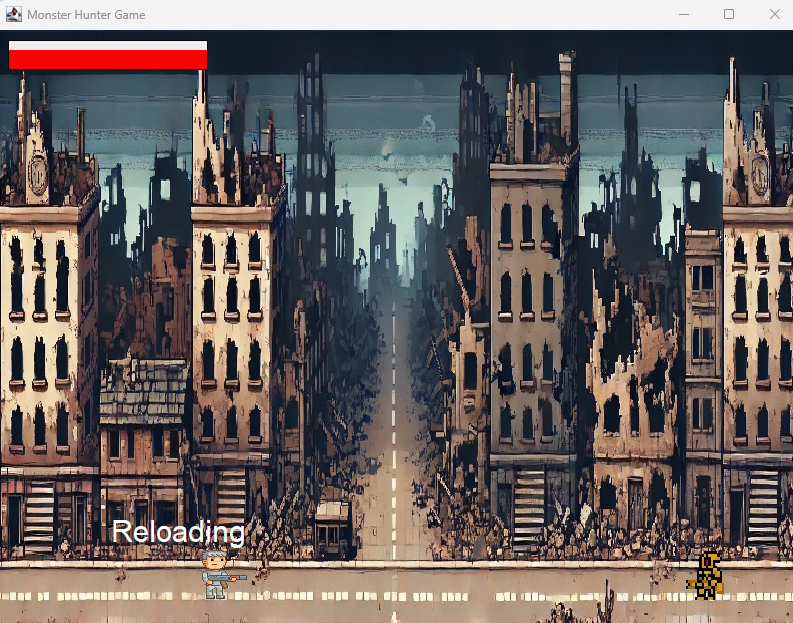
**-JFrame** เป็น container สำหรับเกมนี้

**-JSlider (healthBar)** แสดงสถานะเลือดปัจจุบันของผู้เล่น

**-JLabel (reloadLabel)** แสดงข้อความ “Reload” เมื่อมีการกดปุ่มรีโหลด

**-JButton (Start Game)** เป็นปุ่มไว้ทำการเริ่มเกม

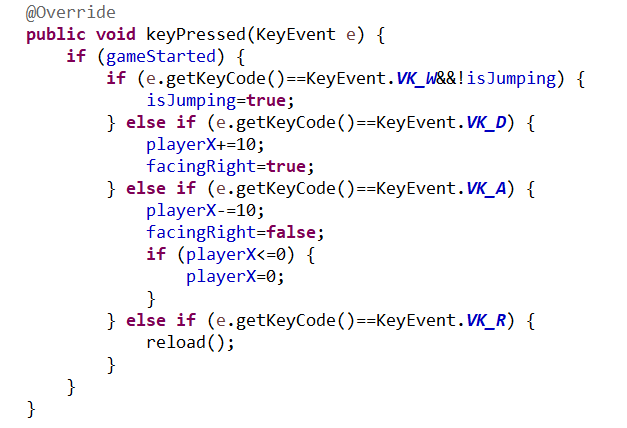


****

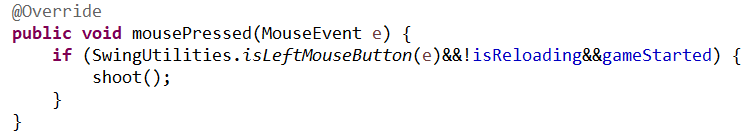
**Event Handling**

จัดการกับเหตุการณ์ถูกใช้ผ่าน Listener หลาย Listener ดังนี้

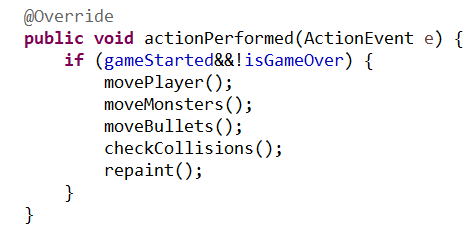
**-KeyListener:** จัดการปุ่มบนคีย์บอร์ดเพื่อให้ผู้เล่นเคลื่อนที่ปุ่ม W, A, D และ Reload R



**-MouseListener:** ควบคุมการคลิกเมาส์เพื่อยิงกระสุนปืน

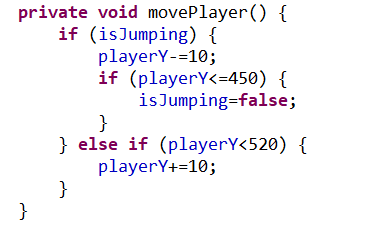


**-ActionListener:** ใช้เพื่อดักจับว่าตอนนี้อยู่ที่หน้าก่อนเริ่มเกมไหมถ้าไม่จะทำการเริ่มเกมเลยและเป็นเมธอดที่จัดการเกมทั้งหมด



**อัลกอริธึมที่สำคัญ**

**-การเคลื่อนที่ของผู้เล่น:**

****

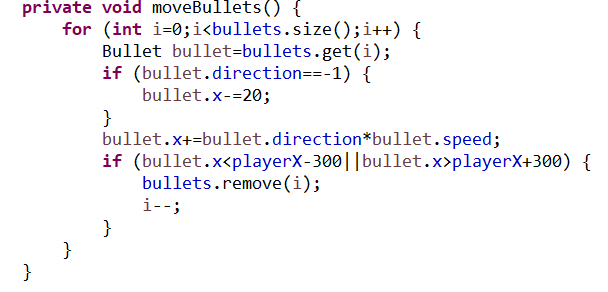
-เมธอด movePlayer() จะจัดการการกระโดดตามสถานะ isJumping

**-การเคลื่อนที่และการชนของมอนสเตอร์:**

****

-เมธอด moveMonsters() จะเคลื่อนที่มอนสเตอร์ไปหาผู้เล่นและตรวจสอบว่าถ้ามอนสเตอร์เข้าใกล้ผู้เล่นน้อยกว่า 30 พิกเซลจะทำการลดเลือดผู้เล่นทีละ 1 และเมื่อเลือดผู้เล่นหมดเกมจะแพ้ทันที

**-การเคลื่อนที่ของกระสุนและการชน:**

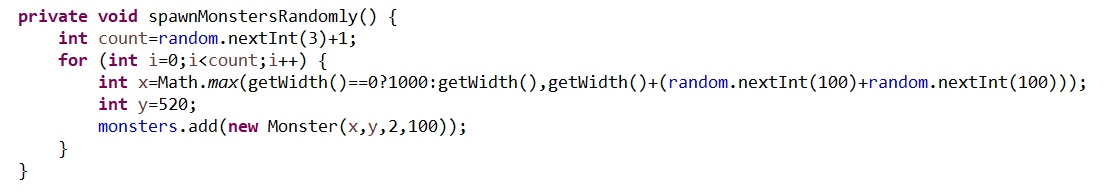
****

-เมธอด moveBullets() จะทำการขยับกระสุนและลบกระสุนหลังห่างจากผู้เล่นไป 300 พิกเซล



-เมธอด checkCollisions() จะตรวจสอบว่ากระสุนถูกมอนสเตอร์หรือไม่ โดยถ้ากระสุนโดนจะลดเลือดมอนสเตอร์ทีละ 10 และทำการลบมอนสเตอร์ออกหากเลือดหมด

**-การสร้างมอนสเตอร์:**

****

-เมธอด spawnMonstersRandomly() จะสร้างมอนสเตอร์ในตำแหน่งสุ่มด้านนอกจอ เพื่อเพิ่มความท้าทายในเกมและในเมธอดยังมีการสุ่มจำนวนการเกิดของมอนเตอร์ในแต่ละครั้งด้วย

**บทที่ 3 สรุป**

**ปัญหาที่พบระหว่างการพัฒนา**

-กระสุนยิงไปแล้วเลือดมอนสเตอร์ไม่ลด

-เมื่อถูกมอนสเตอร์ตีจะไม่สามารถตีมอนสเตอร์กลับได้

**จุดเด่นของโปรแกรมที่ไม่เหมือนใคร**

-การเกิดของมอนสเตอร์แบบสุ่มและจำนวนการเกิดแบบสุ่ม

-มีพื้นหลังที่สมจริง

**คำแนะนำสำหรับผู้สอนที่อยากให้อธิบาย**

-