



โครงการ
รับเลี้ยงน้องแมว

จัดทำโดย
6504062610269 นายศิริพงษ์ พฤติเครือวงศ์

เสนอ
ผู้ช่วยศาสตราจารย์สฤติย์ ประสมพันธ์
วิชา 040613204 Object-Oriented Programming
ภาคเรียนที่1 ปีการศึกษา 2567

ภาควิชาวิทยาการคอมพิวเตอร์และสารสนเทศ คณะวิทยาศาสตร์ประยุกต์
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

บทที่ 1 ที่มาและความสำคัญของโครงการ

โครงการ “รับเลี้ยงน้องแมว” ถูกจัดทำขึ้นเพื่อประยุกต์ความรู้ในการเขียนโปรแกรมเชิงวัตถุ (Object-Oriented Programming) ในรูปแบบของเกม โดยผู้จัดทำเลือกใช้การเขียนโค้ดภาษา Java เพื่อพัฒนาเกมประเภท 2 มิติ ที่มีการควบคุมการเคลื่อนไหวของผู้เล่น โดยต้องคอยหลบสิ่งกีดขวาง และสะสมคะแนนผ่านการเก็บแมวที่เป็นมิตร ซึ่งเกมนี้จะช่วยให้ผู้จัดทำได้ฝึกฝนการใช้โครงสร้างของโปรแกรมเชิงวัตถุและการออกแบบคลาส รวมถึงฝึกการจัดการกับ GUI และ Event Handling อย่างมีประสิทธิภาพ

ประเภทของโครงการ

เกม (Game)

ประโยชน์ของโครงการ

1. ฝึกฝนการคิดแก้ปัญหาเชิงตรรกะ
2. พัฒนาทักษะในการเขียนโปรแกรมเชิงวัตถุ (OOP)
3. ช่วยสร้างความบันเทิงและผ่อนคลายความเครียด
4. ฝึกฝนการจัดการกับ GUI และ Event Handling ใน Java

ขอบเขตของโครงการ

1. การควบคุมการเคลื่อนไหวของตัวละครหลัก (ผู้เล่น) ให้สามารถกระโดดหลบสิ่งกีดขวาง
2. การสร้างระบบเก็บสะสมคะแนนจากการเก็บแมวที่เป็นมิตร
3. การออกแบบฟังก์ชันยิงกระสุนจากอาหารแมวเพื่อเปลี่ยนแมวจรจัดให้เป็นแมวที่เป็นมิตร
4. การแสดงผลภาพพื้นหลังและองค์ประกอบต่าง ๆ บนหน้าจอ รวมถึงตัวนับคะแนนและจำนวนอาหารที่เหลือ
5. การตั้งค่าหน้าจอเริ่มต้นของเกมและหน้าจอแสดงผล Game Over เมื่อผู้เล่นแพ้เกม

บทที่ 2 การพัฒนา

เนื้อเรื่องย่อ

ในเกม “รับเลี้ยงน้องแมว” ผู้เล่นจะควบคุมตัวละครเพื่อกระโดดหลบแมวจรจัดและถึงขยะที่เข้ามาใกล้ รวมถึงสามารถเก็บอาหารแมวเพื่อใช้ยิงแมวจรจัด ทำให้พวกมันกลายเป็นแมวที่เป็นมิตร และเพิ่มคะแนนให้กับผู้เล่น

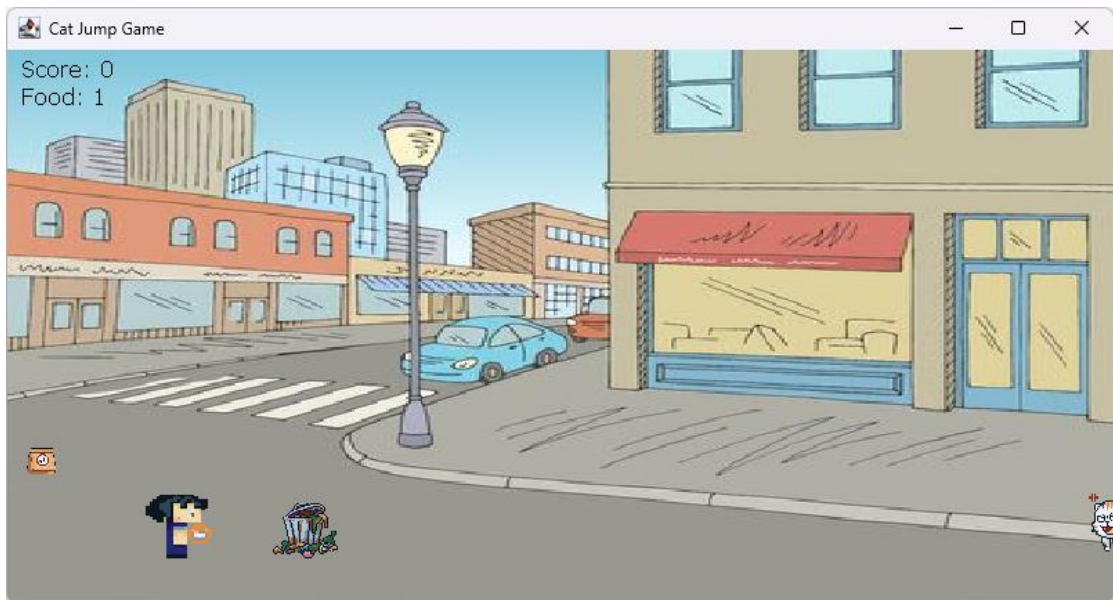
วิธีการเล่น

- ใช้ปุ่ม **Spacebar** เพื่อกระโดดหลบแมวจรจัดและถึงขยะ
- ใช้ปุ่ม **F** เพื่อยิงกระสุนอาหารใส่แมวจรจัด เมื่อแมวกลายเป็นมิตร ผู้เล่นสามารถเดินไปเก็บคะแนนได้
- ใช้ปุ่ม **R** เพื่อเริ่มเกมใหม่เมื่อจบเกม

Storyboard



ผู้เล่นเริ่มเกมและเคลื่อนที่ไปทางด้านขวาของหน้าจอ



แมวจรจัดและถังขยะจะปรากฏขึ้นแบบสุ่ม ผู้เล่นต้องกระโดดหลบ



อาหารแมวจะปรากฏเป็นครั้งคราว ผู้เล่นสามารถเก็บเพื่อเพิ่มจำนวนกระสุนได้



เมื่อผู้เล่นชนกับแมวจรจัดที่ยังไม่ได้ยิง หรือชนกับถังขยะ เกมจะจบลงและแสดงคะแนนที่ได้รับ

เกมนี้จะมีคลาสหลัก 5 คลาสคือ:

1. คลาส Player - เก็บข้อมูลของผู้เล่น รวมถึงการกระโดด เก็บอาหาร และยิงกระสุน
2. คลาส Obstacle - ใช้สำหรับสิ่งกีดขวางที่เป็นแมวจรจัดและถังขยะ
3. คลาส Food - แทนอ็อบเจกต์อาหารแมวที่ผู้เล่นสามารถเก็บได้
4. คลาส Bullet - ใช้แทนกระสุนอาหารที่ยิงไปยังแมวจรจัด
5. คลาส Game - คลาสหลักที่ทำหน้าที่ควบคุมการแสดงผล การอัปเดตเกม และจัดการ Event Handling ทั้งหมด

รูปแบบการพัฒนาโครงการ

- ภาษา: Java
- GUI: javax.swing
- การออกแบบภาพ: Pixilart.com

แนวความคิดเขียนโปรแกรมเชิงวัตถุ

1. Encapsulation (การห่อหุ้มข้อมูล)

การ Encapsulation ใช้เพื่อควบคุมการเข้าถึงข้อมูลและฟังก์ชันภายในคลาส โดยเก็บข้อมูลเป็น private และให้เข้าถึงผ่าน method ที่กำหนด เช่น getX() และ getY() ของคลาส Player และ Obstacle ซึ่งปกป้องข้อมูลจากการเข้าถึงโดยตรงจากภายนอก และลดข้อผิดพลาดที่อาจเกิดขึ้นได้

```
class Player {
    private int x;
    private double y, vy;
    private int food;
    private boolean isJumping;

    public Player(int x, int y) {
        this.x = x;
        this.y = y;
        this.vy = 0;
        this.food = 0;
        this.isJumping = false;
    }

    public void jump() {
        if (!isJumping) {
            vy = -15;
            isJumping = true;
        }
    }
}
```

2. Inheritance (การสืบทอด)

การสืบทอด (Inheritance) คือการสร้างคลาสใหม่โดยใช้คลาสที่มีอยู่เป็นแม่แบบ เพื่อให้คลาสลูกสามารถใช้ข้อมูลและพฤติกรรมของคลาสแม่ได้ ในตัวอย่างนี้ คลาส Obstacle ถูกใช้แทนแมวจรจัดและถึงขยะ ซึ่งสามารถแบ่งประเภทได้โดยการกำหนดค่า type เช่น 0 สำหรับแมวจรจัด และ 2 สำหรับถึงขยะ

```
class Obstacle {
    private int x, y;
    private int type;
    private boolean friendly = false;
    private int colorIndex;

    public Obstacle(int x, int y, int type, int colorIndex) {
        this.x = x;
        this.y = y;
        this.type = type;
        this.colorIndex = colorIndex;
    }

    public void update() { x -= 3; }

    public int getType() { return type; }
    public int getX() { return x; }
    public void setFriendly(boolean friendly) { this.friendly = friendly; }
    public boolean isFriendly() { return friendly; }
}
```

3. Polymorphism (พหุรูป)

Polymorphism ช่วยให้วัตถุในคลาสสามารถเปลี่ยนพฤติกรรมได้ตามสถานะหรือประเภท เช่น การวาดภาพของ Obstacle เมื่อเป็นแนวจอดหรือแนวที่เป็นมิตร ใช้ Polymorphism เพื่อเปลี่ยนการแสดงผลภาพใน method draw()

```
public void draw(Graphics g) {  
    if (type == 0) {  
        g.drawImage(friendly ? friendlyCatImg[colorIndex] : strayCatImg[colorIndex], x, y,  
    } else if (type == 2) {  
        g.drawImage(trashCanImg, x, y, width:50, height:50, observer:null);  
    }  
}
```

4. Abstraction (นามธรรม)

การใช้ Abstraction ช่วยให้ซ่อนรายละเอียดที่ไม่จำเป็นต่อการใช้งาน โดยกำหนดเฉพาะโครงสร้างหลักที่จำเป็นต้องใช้ในโปรแกรม เช่น ในการสร้างคลาส Bullet และ Food โดยไม่แสดงรายละเอียดภายในของการอัปเดตและการวาด

```
class Food {  
    private int x, y;  
    public Food(int x, int y) { this.x = x; this.y = y; }  
  
    public void update() { x -= 2; }  
  
    public int getX() { return x; }  
    public Rectangle getBounds() { return new Rectangle(x, y, width:30, height:30); }  
    public void draw(Graphics g) { g.drawImage(foodImg, x, y, width:30, height:30, observer:null); }  
}  
  
class Bullet {  
    private int x, y;  
    public Bullet(int x, int y) { this.x = x; this.y = y; }  
  
    public void update() { x += 6; }  
  
    public int getX() { return x; }  
    public Rectangle getBounds() { return new Rectangle(x, y, width:10, height:10); }  
    public void draw(Graphics g) { g.drawImage(bulletImg, x, y, width:10, height:10, observer:null); }  
}
```

บทที่ 3 สรุป

ปัญหาที่พบระหว่างการพัฒนา

1. การตั้งค่าการเคลื่อนไหวของแมวจรจัดและถึงขยะให้เข้ามาตามลำดับที่กำหนดอย่างเหมาะสม
2. การจัดการการชนระหว่างผู้เล่นและสิ่งกีดขวางให้ทำงานอย่างถูกต้องโดยไม่เกิดข้อผิดพลาด
3. การจัดการการยิงกระสุนเพื่อเปลี่ยนสถานะของแมวจรจัดให้กลายเป็นแมวที่เป็นมิตร

จุดเด่นของโปรแกรม

1. เกมนี้มีการควบคุมที่เข้าใจง่าย ใช้การกระโดดหลบสิ่งกีดขวางและการยิงกระสุน
2. มีการใช้แนวคิดเชิงวัตถุอย่างครบถ้วน รวมถึงการสร้าง GUI และ Event Handling ด้วย Java
3. เป็นเกมที่ทำง่ายและช่วยฝึกทักษะการแก้ปัญหาและการเขียนโปรแกรมเชิงวัตถุ