โครงงาน (game project)

จัดทำโดย

6204062630343 นาย ปณิธิ แก้วเจริญ

เสนอ

ผู้ช่วยศาสตราจารย์สถิต ประสมพันธ์

วิชา Object Oriented Programming

ภาคเรียนที่ 1/2566

ภาควิชาวิทยาการคอมพิวเตอร์และสารสนเทศ คณะวิทยาศาสตร์ประยุกต์

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

บทที่ 1 ที่มาและความสำคัญของโครงการ

โครงงานนี้จัดขึ้นเพื่อวัดผลความสามารถในการเรียนวิชา Object Oriented Programming โดยการนำเรื่องที่เรียนมาสร้างเป็นชิ้นงานในรูปแบบเกม

ประเภทโครงการ

โปรแกรม java game

ชื่อโปรเจค: NinjaDogJump project

นำเสนอโดย: นาย ปณิธิ แก้วเจริญ

อาจารย์ผู้สอน: ผู้ช่วยศาสตราจารย์สถิต ประสมพันธ์

ประโยชน์

1.เพื่อความสนุกในการคลายเครียด

2.เพื่อฝึกสมาธิ

3.ช่วยทำให้รู้จักนำหลักการoopมาใช้

ตารางแผนการทำงานเดือนตุลาคม-พฤศจิกายน

	รายละเอียด	1-5	6-17	18-31	1-10
1	ศึกษาและสร้างแพลนเกมคร่าวๆ	✓			
2	ลงมือเขียนโปรแกรม		✓		
3	แก้ไขโปรแกรมใหม่เนื่องจากใช้library มาช่วยมาเกินไป			✓	
4	ทำเอกสาร				✓

บทที่ 2 ส่วนการพัฒนาโปรแกรม

2.1 รูปแบบการพัฒนา

2.2.1 มีฟังก์ชันการเริ่มเกม การเกิดของobjต่างๆ

2.2.2 มีการใช้หลัก OOP

2.2 อธิบายส่วนของโปรแกรมที่มีการใช้หลักการ OOP

มีการใช้ constructor

จากภาพมีการสร้างคลาส DogJump และมีการสร้าง constructor เพื่อเก็บขนาดของเกม

```
public class DogJump extends JPanel implements Runnable, KeyListener {
    final int WIDTH = 500;
    final int HEIGHT = 800;

    boolean isRunning;
    boolean gameover;
    Thread thread;
    BufferedImage view, background, platform, dog;

    PlatformPosition[] platformsPosition;
    int x = 100, y = 100, h = 500;
    float dy = 0;
    boolean right, left;
    int score = 0;
    boolean restart = false;

    public DogJump() {
        setPreferredSize(new Dimension(WIDTH, HEIGHT));
        addKeyListener(this);
    }
```

มีการ extends JPanel (ซึ่งใช้หลัก polymophism)

```
public class DogJump extends JPanel implements Runnable, KeyListener {
    final int WIDTH = 500;
    final int HEIGHT = 800;
```

มีการ implements Runnable,KeyListener (ซึ่งเป็น interface มาใช้งาน)

```
public class DogJump extends JPanel implements Runnable, KeyListener {
    final int WIDTH = 500;
    final int HEIGHT = 800;
```

มีส่วนการใช้งาน GUI JFrame

```
public static void main(String[] args) {
    JFrame w = new JFrame("Ninja Dog Jump");
    w.setResizable(false);
    w.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    w.add(new DogJump());
    w.pack();
    w.setLocationRelativeTo(null);
    w.setVisible(true);
}
```

มีการใช้ Thread ในโปรแกรม

```
@Override
public void addNotify() {
    super.addNotify();
    if (thread == null) {
        thread = new Thread(this);
        isRunning = true;
        thread.start();
    }
}
```

ส่วนฟังก์ชันของเกม

จะมีการกำหนดค่าการขยับของตัวละครให้ไปซ้ายไปขวา

```
public void update() {
    if (restart) {
        restartGame();
    }

    if (right) {
        x += 3;
    } else if (left) {
        x -= 3;
    }
    dy += 0.2;
    y += dy;
    if (y >= 800) {
        gameover = true;
        isRunning = false;
    }
```

ฟังก์ชันในการสร้าง platform ใหม่เรื่อยเมื่อ มีplatform อันนึงลงไปอยู่ต่ำกว่าframe จะทำการสร้าง platform อันใหม่ขึ้นมาอยู่บนสุดของ frame

```
if (y < h) {
    for (int i = 0; i < 10; i++) {
        y = h;
        platformsPosition[i].y = platformsPosition[i].y - (int) dy;
        if (platformsPosition[i].y > 800) {
            platformsPosition[i].y = 0;
            platformsPosition[i].x = new Random().nextInt(450);
        }
    }
}
```

ฟังก์ชันของการที่จะทำให้ตัวละครเมื่อเหยียบ platform แล้วจะมีการกระโดดขึ้นไปโดยตัวแปร dy คือค่าการกำหนดระยะการกระโดดของตัวละคร และจะมีการเพิ่ม score ทุกครั้งที่เหยียบกับ platform

```
for (int i = 0; i < 10; i++) {
    if ((x + 50 > platformsPosition[i].x) &&
            (x + 20 < platformsPosition[i].x + 40) &&
            (y + 70 > platformsPosition[i].y) &&
            (y + 70 < platformsPosition[i].y + 50) &&
            (dy > 0)) {
        dy = -12;
        score++;
    }
```

การสร้าง Object platform,background,dog

platformจะถูกใช้เป็นarray เพื่อนำไปใช้ในการ random สร้างขึ้น

```
public void start() {
    try {
        view = new BufferedImage(WIDTH, HEIGHT, BufferedImage.TYPE_INT_RGB);

        background = ImageIO.read(getClass().getResource("background.png"));
        platform = ImageIO.read(getClass().getResource("platform.png"));
        dog = ImageIO.read(getClass().getResource("dog.png"));

        platformsPosition = new PlatformPosition[10];
        for (int i = 0; i < 10; i++) {
            platformsPosition[i] = new PlatformPosition();
            platformsPosition[i].x = new Random().nextInt(450);
            platformsPosition[i].y = new Random().nextInt(800);
        }
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

ส่วนของ Graphic

```
public void draw() {
    Graphics2D g2 = (Graphics2D) view.getGraphics();
    g2.drawImage(background, 0, 0, WIDTH, HEIGHT, null);
    g2.drawImage(dog, x, y, dog.getWidth(), dog.getHeight(), null);
    for (int i = 0; i < 10; i++) {
        g2.drawImage(platform, platformsPosition[i].x, platformsPosition[i].y, platform.getWidth(),
    }
    if (gameover) {
        Font font = new Font("Arial", Font.BOLD, 36);
        g2.setFont(font);
        g2.setColor(Color.RED);
        String message = "Game Over";
        int messageWidth = g2.getFontMetrics().stringWidth(message);
        int messageX = (WIDTH - messageWidth) / 2;
        int messageY = HEIGHT / 2;
        g2.drawString(message, messageX, messageY);

        String scoreMessage = "Score: " + score;
        int scoreWidth = g2.getFontMetrics().stringWidth(scoreMessage);
        int scoreX = (WIDTH - scoreWidth) / 2;
        int scoreY = HEIGHT / 2 + 50;
        g2.drawString(scoreMessage, scoreX, scoreY);

        String restartMessage = "Press 'R' to restart";
        int restartWidth = g2.getFontMetrics().stringWidth(restartMessage);
        int restartX = (WIDTH - restartWidth) / 2;
        int restartY = HEIGHT / 2 + 100;
        g2.drawString(restartMessage, restartX, restartY);
    }
```

ส่วนของ KeyListener

ในการรับข้อมูล Keyboard ที่ใช้ในการควบคุมตัวละคร

```
@Override
public void keyPressed(KeyEvent e) {
    if (e.getKeyCode() == KeyEvent.VK_RIGHT) {
        right = true;
    }
    if (e.getKeyCode() == KeyEvent.VK_LEFT) {
        left = true;
    }
    if (e.getKeyCode() == KeyEvent.VK_R) {
        restart = true;
    }
}

@Override
public void keyReleased(KeyEvent e) {
    if (e.getKeyCode() == KeyEvent.VK_RIGHT) {
        right = false;
    }
    if (e.getKeyCode() == KeyEvent.VK_LEFT) {
        left = false;
    }
    if (e.getKeyCode() == KeyEvent.VK_R) {
        restart = false;
    }
}
```

คลาสที่มีการ Override คือมีการ implements มาจาก interface

เช่น keyPressed,Thread เป็นต้น

บทที่ 3 สรุป

ปัญหาที่พบระหว่างการพัฒนา

1. ปัญหาช่วงแรกคือการทำโปรเจคโดยมีการเรียนการสร้างเกมมาจากข้างนอกซึ่งมีการใช้ library ตัวอื่นมาช่วยเยอะเกินไป จึงเริ่มทำใหม่

2.แบ่งเวลาให้โปรเจคน้อยเกินไป พอมาทำใหม่จึงทำได้ไม่ทัน

3.เกิดบัคในการเก็บ SCORE เพื่อที่จะเปลี่ยนแมพ พอSCORE ถึง50จะมีการเปลี่ยนฉากเกมแต่เกิดบัค พอเปลี่ยนฉากทำให้ตัวเกมไม่สามารถเล่นต่อได้จึงนำออก

จุดเด่นของโปรแกรม

เรียบง่ายเล่นได้เรื่อย คลายเครียดระหว่างรอรถหรือรอเรียน ฝึกสมาธิไปในตัว

คำแนะนำสำหรับผู้สอนที่อยากให้อธิบาย หรือที่เรียนแล้วไม่เข้าใจ หรืออยากให้เพิ่มสำหรับน้องๆรุ่นต่อไป

อยากให้อาจารย์มี Quiz ทีเป็นเรื่องของทฤษฎีเพิ่มเข้ามาเป็นคะแนนเก็บอีกสัก5-10คะแนนครับ เรื่องของการสอนช่วงเขียนโค้ดยกตัวอย่างอยากให้ช้าลงหน่อยนึงครับอาจารย์ บางทีตามไม่ทันแล้วผมหลุดไปเลย ขอบคุณครับ