



โครงการ

Project game OOP

จัดทำโดย

ปรเมษฐ์ พุกุลานนท์ 6404062620010

อาจารย์ผู้สอน

ผู้ช่วยศาสตราจารย์ สถิต ประสมพันธ์

วิชา Object Oriented Programming

ภาคเรียนที่ 1/2565

บทที่ 1 ที่มาและความสำคัญ

โครงการนี้จัดทำขึ้นเพื่อวัดผลการเรียนรู้ในวิชา Object Oriented Programming โดยการนำเรื่องที่เรียนมาสร้างเป็นโครงการในรูปแบบเกมโดยทางผู้จัดได้จัดทำเกมนี้ขึ้นเพื่อฝึกไหวพริบของผู้เล่น โดยระดับความยากจะเพิ่มมากขึ้น เมื่อผู้เล่นสามารถเก็บแต้มได้เพิ่มขึ้น และทางผู้จัดได้ตั้งชื่อเกมนี้ว่า “เกมเด็กปั้มเก็บน้ำมัน”

ประโยชน์ของเกม

- ฝึกความอดทน
- ฝึกไหวพริบ
- คลายเครียด
- สามารถนำเรื่องที่เรียนมาประยุกต์ใช้

ขอบเขตโครงการ

-รายละเอียดเกม

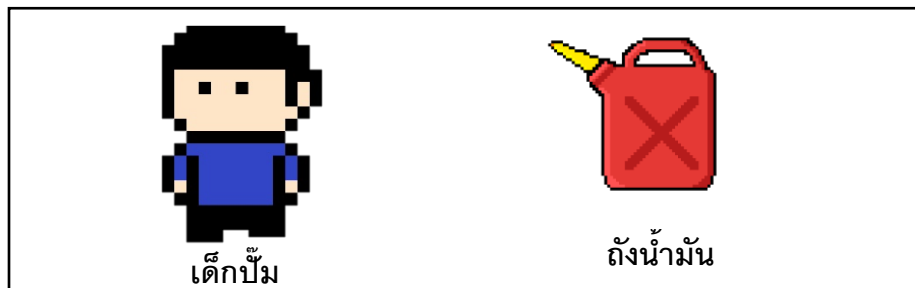
สมชายเป็นเด็กปั๊มแห่งหนึ่ง ซึ่งเขาได้ก่อเรื่องเอาไว้คือเขาได้ทำถังน้ำมันตกไว้ตามทาง ถ้าเขาจึงต้องตามเก็บถังน้ำมันตามทาง ถ้าเขาเก็บพลาดจะทำให้เขาต้องตกงาน

-วิธีการเล่น

ใช้ลูกขรบนเป็นคีย์บอร์ดบังคับทิศทางช่วยเด็กปั๊มเก็บน้ำมัน และระวังอย่าให้พลาด ถ้าเขาพลาดจะทำให้เกมจบทันที

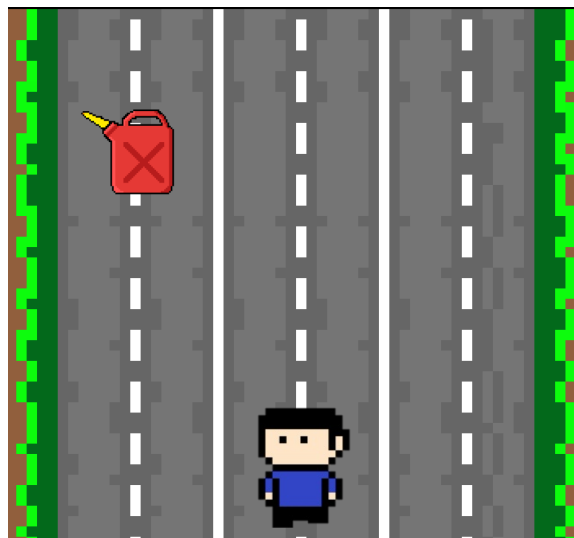
-storyboard

ตัวละคร

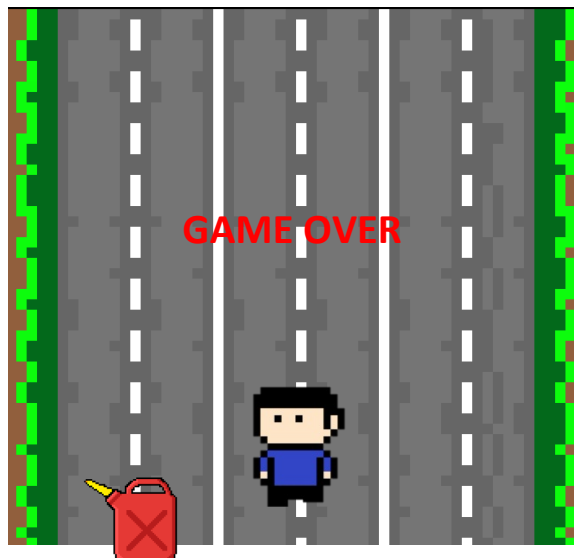


ฉาก

เริ่มเกม



- ถ้าหากเขาเก็บถึงน้ำมันไม่ทันก็จะจบเกมทันที และต้องไปเริ่มใหม่



- ถ้าเขาเก็บถึงน้ำมันได้ จะได้ score +1

-ประโยชน์

1. ฝึกไหวพริบ
2. เพื่อความสนุกสนาน
3. ฝึกการตัดสินใจ

-ตารางการทำงาน

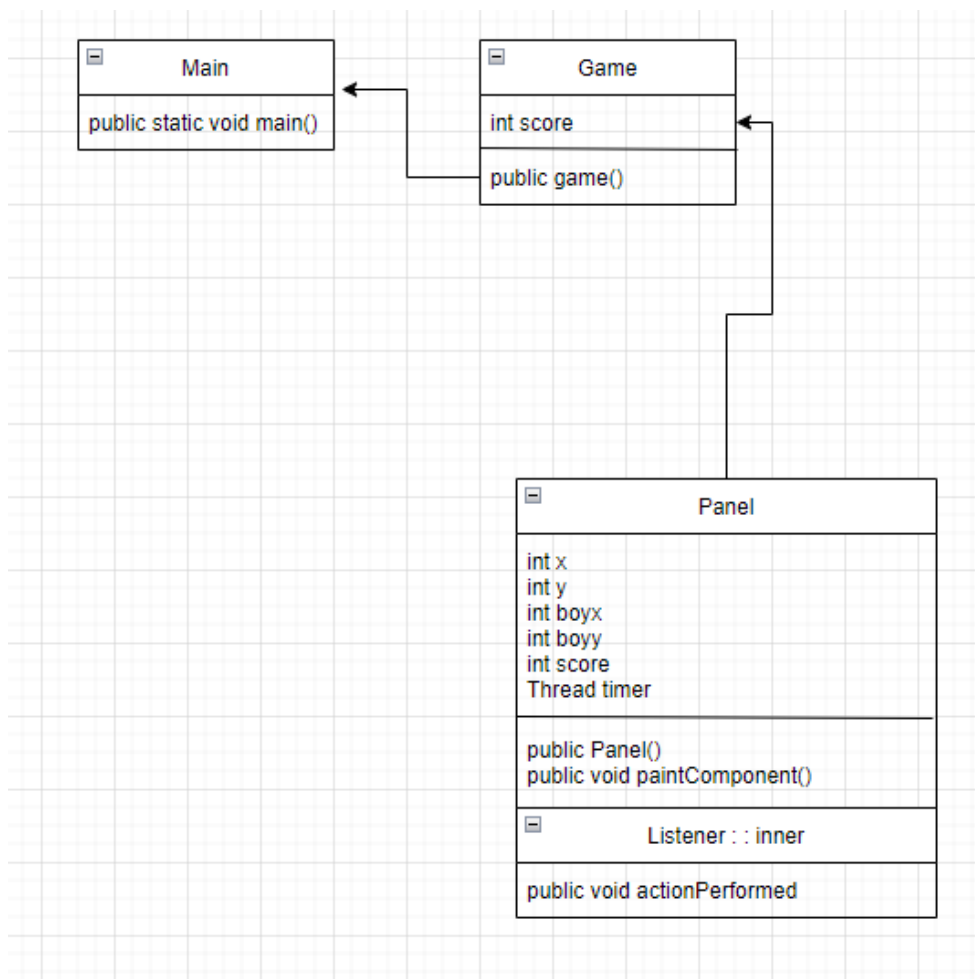
ลำดับ	รายการ	1-7	8-26	27-30
1	หารูปตัวละคร			
2	ศึกษาข้อมูลที่เกี่ยวข้องกับการทำเกม			
3	เขียนโปรแกรม			
4	ทำเอกสาร			
5	ทดลองและแก้ไข			

บทที่ 2 ส่วนของการพัฒนา

เนื้อเรื่องย่อ

สมชายเป็นเด็กป่วนคนหนึ่ง ซึ่งเขาได้ก่อเรื่องเอาไว้คือเขาได้ทำถังน้ำมันตกไว้ตามทาง ถ้าเขาจึงต้องตามเก็บถังน้ำมันตามทาง ถ้าเขาเก็บพลาดจะทำให้เขาต้องตกงาน

Class Diagram



ส่วนการทำงานของโปรแกรม

-Constructor

Code	อธิบาย
<pre>class Panel extends JPanel { JButton EXITGame = new JButton("EXIT"); int x = 100; int y = 0; private int boyx = 320; private int boyy = 460; int score = 0; Thread timer; public Panel() { timer = new Thread(new Runnable() { public void run() { while (true) { try { Thread.sleep(10); repaint(); } catch (Exception e) { } } } }); timer.start(); } }</pre>	เมื่อมีการเรียกใช้คลาส Panel จะมีการเรียกใช้ constructor Panel() โดยใน constructor จะเป็นการเรียกใช้ Thread

-Encapsulation

Code	อธิบาย
<pre>class Panel extends JPanel { JButton EXITGame = new JButton("EXIT"); public int x = 100; public int y = 0; private int boyx = 320; private int boyy = 460;</pre>	เป็นการประกาศตัวแปร ถ้าหากเป็น public จะทำให้สามารถใช้ได้ในทุกคลาส แต่หากเป็น private จะใช้ได้แค่ในคลาสตัวเอง

-Inheritance

Code	อธิบาย
<pre>class Panel extends JPanel { JButton EXITGame = new JButton("EXIT"); public int x = 100; public int y = 0; private int boyx = 320; private int boyy = 460;</pre>	Class Panel ใช้คุณสมบัติสืบทอด (Inheritance) จาก คลาส JPanel

-Graphics

Code	อธิบาย
<pre>public void paintComponent(Graphics g) { Random rand = new Random(); super.paintComponent(g); try { g.drawImage(ImageIO.read(new File("img\\footpath.png")), 0, 0, this.getWidth(), this.getHeight(), null); } catch (IOException ex) { Logger.getLogger(Game.class.getName()).log(Level.SEVERE, null, ex); } try { g.drawImage(ImageIO.read(new File("img\\boy.png")), boyx, boyy, 80, 80, null); } catch (IOException ex) { Logger.getLogger(Game.class.getName()).log(Level.SEVERE, null, ex); } try { g.drawImage(ImageIO.read(new File("img\\fuel.png")), x, y, 80, 80, null); } catch (IOException ex) { Logger.getLogger(Game.class.getName()).log(Level.SEVERE, null, ex); } g.setColor(black); g.setFont(new Font("Jokerman", Font.PLAIN, 30)); g.drawString("SCORE : " + score, 35, 30);</pre>	ตัวอย่างการสร้างหน้าจอเกม และ component ต่างๆ เช่น Background ตัวละคร และ ถังน้ำมัน

-GUI

Code	อธิบาย
<pre>class Panel extends JPanel { JButton EXITGame = new JButton("EXIT"); public int x = 100; public int y = 0; private int boyx = 320; private int boyy = 460;</pre>	มีการสร้างปุ่มชื่อ EXIT โดยใช้ JButton

-Event

Code	อธิบาย
<pre>addKeyListener(new KeyAdapter() { @Override public void keyPressed(KeyEvent e) { switch (e.getKeyCode()) { case KeyEvent.VK_LEFT: boyx -= 25; break; case KeyEvent.VK_RIGHT: boyx += 25; break; } repaint(); } });</pre>	มีการ Check Event การกดปุ่ม โดยถ้าหากกดปุ่มลูกศรซ้ายจะให้ตัวละครเลื่อนแกน x ที่ละ 25 ไปทางซ้ายและทำการ repaint แต่ถ้าหากกดปุ่มลูกศรขวาจะให้ตัวละครเลื่อนแกน x ที่ละ 25 ไปทางขวาและทำการ repaint

-Polymorphism

Code	อธิบาย
<pre>class Listener implements ActionListener{ @Override public void actionPerformed(ActionEvent e) { if(e.getSource() == EXITGame){ System.exit(0); } } }</pre>	Override เป็นคุณสมบัติของ Polymorphism

อัลกอริทึมที่สำคัญในโปรแกรม

Code	อธิบาย
<pre>double con = (Math.pow(boyx - x, 2)) + (Math.pow(boyy - y, 2)); double Collision = Math.sqrt(con); if (Collision <= 75) { y = 0; x = rand.nextInt(650); score += 1; } }</pre>	เป็นการเช็คการชน โดยใช้หลักการทาง คณิตศาสตร์ โดยการหาระยะระหว่างจุด ถ้าห่างกันไม่เกิน 75 จะถือว่าวัตถุชนกัน

บทที่ 3 สรุป

ปัญหาที่พบระหว่างการพัฒนา

- หน้าจอ Game Over ขึ้นมาแบบกะพริบเพราะ Check ค่า $Y = 580$ แก้ปัญหาโดย Check ค่า $Y > 580$
- ทำได้ไ้ที่ครบที่ตั้งเป้าเพราะเวลาที่ทำตรงกับช่วงสอบ

จุดเด่นของโปรแกรม

- มีการเพิ่ม speed หาก score เพิ่มขึ้น เพื่อเพิ่มความยาก