



## Object-oriented Programming mini project

“The adventure of drake”

จัดทำโดย

6504062630189 นายพิสิษฐ์ ศรีโมอ่อน

เสนอ

ผู้ช่วยศาสตราจารย์สถิต ประสมพันธ์

วิชาการเขียนโปรแกรมเชิงวัตถุ (Object-oriented Programming)  
ภาควิชาวิทยาการคอมพิวเตอร์และสารสนเทศ คณะวิทยาศาสตร์ประยุกต์

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

ภาคเรียนที่ 1/2566

# บทที่ 1

## บทนำ

### 1.1 ที่มาและความสำคัญ

โครงงานนี้ถูกจัดทำขึ้นเพื่อส่งงานวิชา Object-oriented Programming เพื่อวัดผลความรู้ที่ได้จากการเรียน วิชา Object-oriented Programming ว่าสามารถนำมาใช้งานได้จริง ทางปฏิบัติ ในรูปแบบของเกม ที่มีเนื้อหาและ concept OOP

### 1.2 ประโยชน์

1.2.1 ฝึกการเขียนโปรแกรม

1.2.2 เพื่อฝึกความอดทน

1.2.3 ฝึกวิธีแก้ปัญหา

### 1.3 ขอบเขตของโครงงาน

เกม The adventure of drake เป็นเกมกระโดดข้ามสิ่งกีดขวางและยิงโครงกระดูกกับเก็บไอเทมเพื่อเก็บแต้ม เมื่อโดนสิ่งกีดขวางเลือดจะลดลงถ้าลดลงหมดจะจบเกมทันที

1.3.1 ประเภทของโครงงาน

โครงงานเป็นประเภทเกมที่สร้างจากภาษา java โดยมีเนื้อหาและ concept ของ OOP

## ตารางแผนการทำงาน

ลำดับ	รายการ	วันที่ 1-6	วันที่ 7-30	วันที่ 31-2
1	หารูปและฉากหลังต่างๆ			
2	ศึกษาเอกสารข้อมูลที่เกี่ยวข้อง			
3	ลงมือเขียนโปรแกรม			
4	จัดทำเอกสาร			
5	ตรวจสอบข้อผิดพลาดต่างๆ			

## 1.4 Storyboardและวิธีการเล่น

### รายละเอียดเกม

เดรดผู้ซึ่งเป็นลูกครึ่งคนครึ่งมังกรรู้สึกเบื่อหน่าย เลยเดินทางออกจากบ้านเกิดเพื่อผจญภัยรอบโลก และหาประสบการณ์ใหม่ๆ ในระหว่างผจญภัยก็จะเจอสิ่งกีดขวางและมอนสเตอร์โครงกระดูกมากมาย เดรดต้องจัดการมอนสเตอร์โครงกระดูกให้หมดไม่งั้นจะไม่สามารถผ่านไปได้

## วิธีการเล่น

1. กดลูกศรข้างบนบน keyboard เพื่อไปกระโดดข้ามสิ่งกีดขวาง
2. กด spacebar บน keyboard เพื่อยิงกระสุนโจมตีใส่โครงกระดูก
3. เกมจะนับคะแนนจนเกมจบ
4. เกมจะจบก็ต่อเมื่อเด้งโดนสิ่งกีดขวางหรือโดนโครงกระดูก

## Storyboard



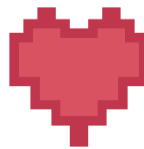
Drake (ตัวละคร)



มอนสเตอร์โครงกระดูก



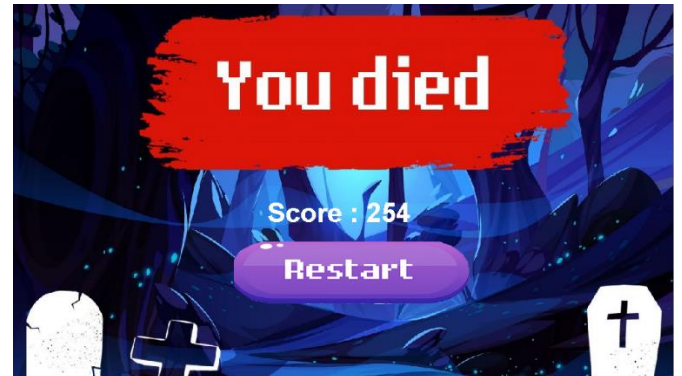
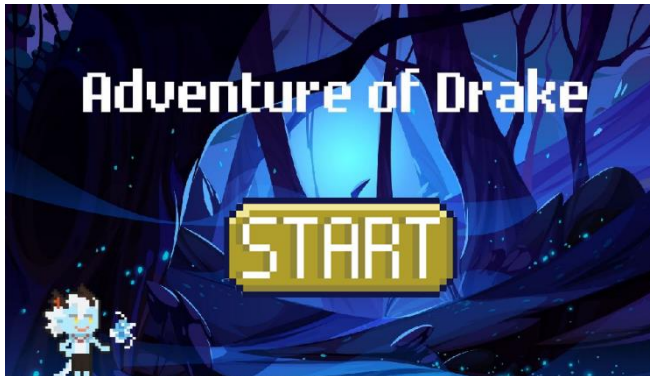
สิ่งกีดขวาง(-10HP)



ไอเทม(ได้คะแนน + ได้เลือด)

## ฉากภายในเกม

### 1. หน้าเริ่มต้น และ หน้าจบเกม



### 2. ฉากภายในเมื่อเริ่มเกม

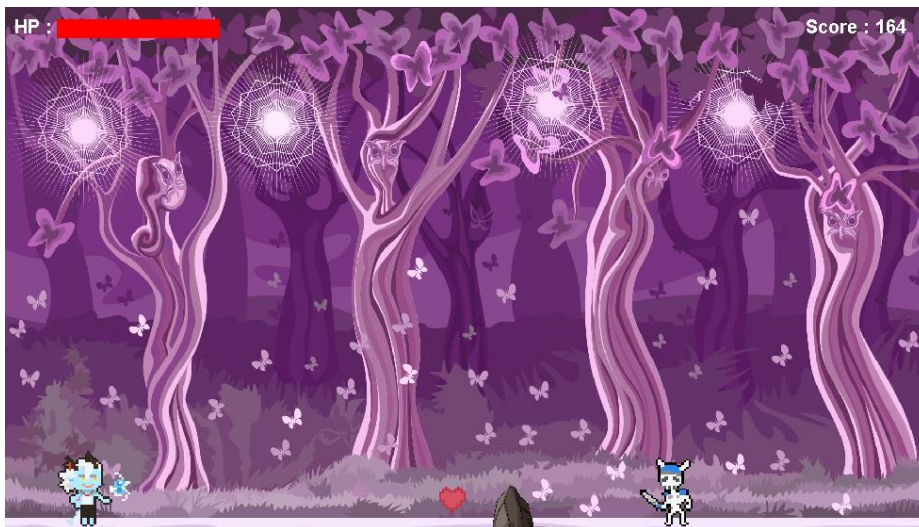
#### 2.1 เมื่อคะแนน มากกว่าเท่ากับ 0 หรือน้อยกว่า 100



2.2 เมื่อคะแนน เท่ากับ 100 หรือ มากกว่า 100

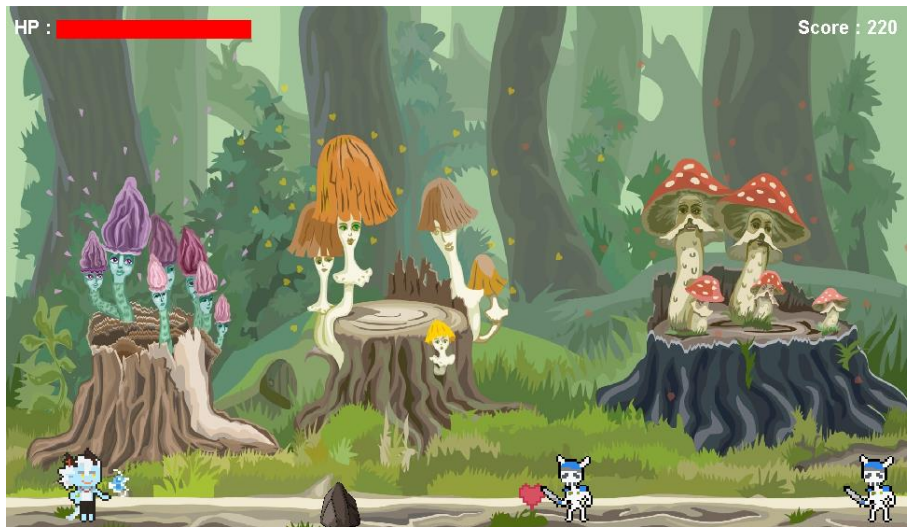


2.3 เมื่อคะแนน เท่ากับ 150 หรือ มากกว่า 150





2.4 เมื่อคะแนน เท่ากับ 200 หรือ มากกว่า 200



3.ฉากยิงกระสุนภายในเกม



## บทที่ 2

### ส่วนของการพัฒนา

#### เนื้อเรื่องย่อ

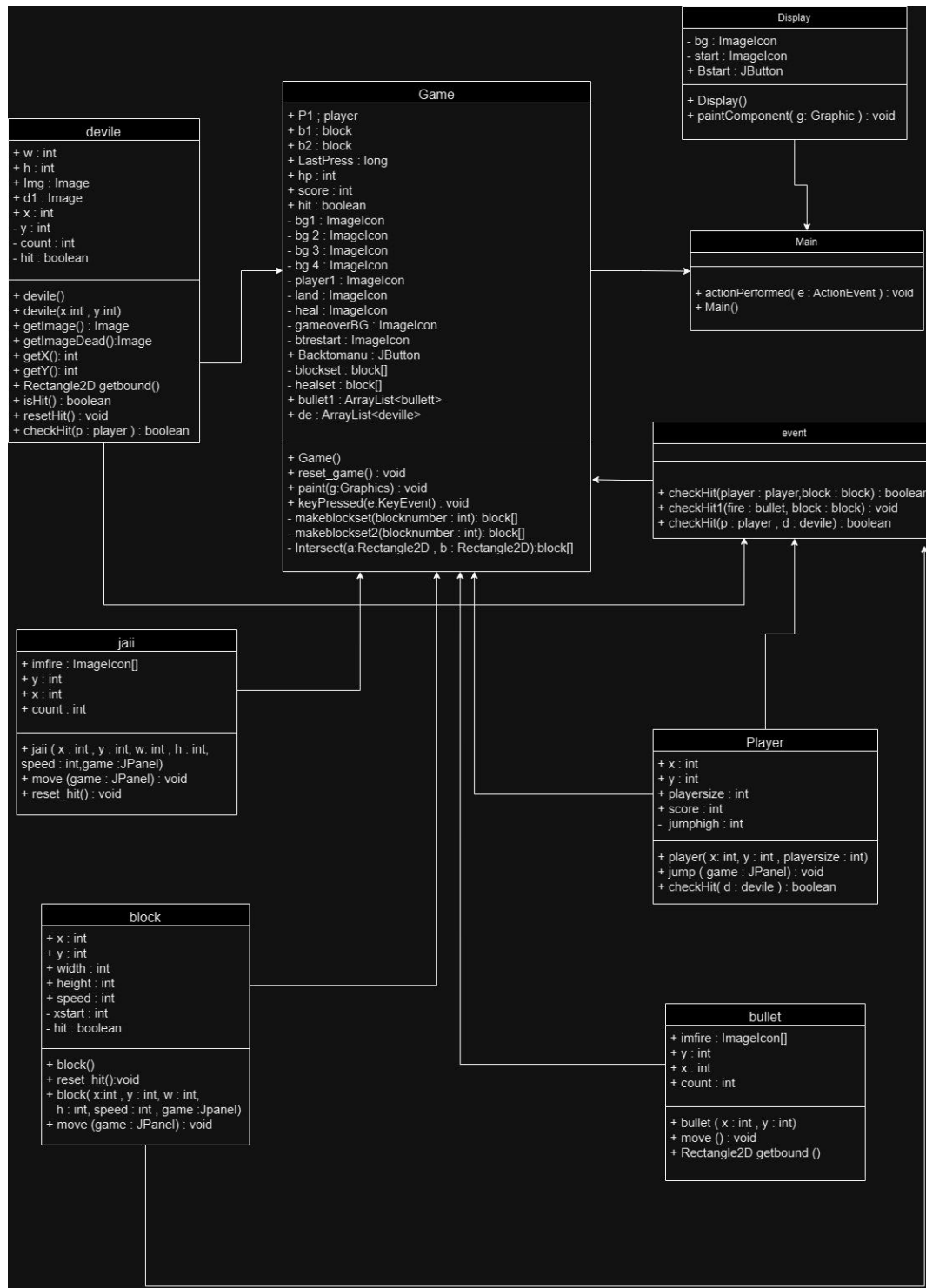
เดรดผู้ซึ่งเป็นลูกครึ่งคนครึ่งมังกรรู้สึกเบื่อหน่าย เลยเดินทางออกจากบ้านเกิดเพื่อผจญภัยรอบโลก และหาประสบการณ์ใหม่ๆ ในระหว่างผจญภัยก็จะเจอสิ่งกีดขวางและมอนสเตอร์โครงกระดูกมากมาย เดรดต้องจัดการมอนสเตอร์โครงกระดูกให้หมดไม่งั้นเลือดจะลดเรื่อยๆจนตาย

#### วิธีการเล่น

1. กด spacebar บน keyboard เพื่อไปกระโดดข้ามสิ่งกีดขวาง
2. กดลูกศรข้างบน บน keyboard เพื่อยิงกระสุนโจมตีใส่โครงกระดูก
3. เกมจะนับคะแนนจนเกมจบ
4. ยิงโครงกระดูกคะแนนจะบวกเพิ่ม และ เก็บหัวใจจะได้คะแนนแล้วก็เพิ่มเลือด
4. เกมจะจบก็ต่อเมื่อเดรดโดนสิ่งกีดขวางหรือโดนโครงกระดูกจนเลือดหมด



## 2.1 Class diagram



### 2.1.1 Class Main

เป็นคลาสที่มีการทำงานในการดึงหน้าpanel ในส่วนต่างๆของเกมมาแสดง และกำหนด ActionListener เช่น event การกดปุ่ม

### 2.1.2 Class Game

เป็นคลาสในส่วนpanelของการเล่นเกม มีหน้าที่กำหนดจัดการควบคุม สิ่งที่เกิดขึ้นภายในเกม เช่นการสร้างโครงกระดูก กำหนดเงื่อนไขที่จะสามารถ ยิงกระสุนออกมาถึงโครงกระดูก มีการกำหนดการนับคะแนน แสดงคะแนน แสดงเลือด มีการตรวจจับ event keypress เพื่อควบคุมเล่นตัวละครในเกม

### 2.1.3 Class Display

เป็นการสร้างPanelกำหนดปุ่มในหน้าstartก่อนเข้าเกม

### 2.1.4 Class Bullett, Class block, Class jaii และ Class devile

คลาสในหัวข้อนี้อาจสร้างขึ้นมากำหนดการสร้างและสุ่มเกิดของ ไอเทม โครงกระดูก และสิ่งกีดขวาง และยังกำหนดให้เคลื่อนที่ในคลาสนี้อีกด้วย

### 2.1.5 Class player

คลาสนี้เป็นการเซตค่าเพื่อสร้างตัวละครและกำหนดตัวแปรเพื่อทำไปใช้เช็คeventต่างๆในเกม

### 2.1.6 Class event

คลาสนี้เป็นคลาสที่สร้างขึ้นมาเพื่อสร้างการเช็คชนกันของไอเทมต่างๆภายในเกมให้คลาสGame เรียกใช้เพื่อนับคะแนน

## 2.2 รูปแบบการพัฒนา Application / Applet

### 1. Application Format (รูปแบบการพัฒนาแอปพลิเคชัน):

- Main Class (Application Entry Point): โปรแกรมจะมีคลาสหลักที่เป็นจุดเริ่มต้นของแอปพลิเคชัน ในที่นี้คือคลาส Main ที่สืบทอดจาก JFrame และมีเป็น Entry Point ของโปรแกรม โดยจะสร้างหน้าต่างแสดงผลและควบคุมการเปลี่ยนแปลงระหว่าง Display และ Game

- Display Class: คลาส Display ใช้สำหรับแสดงหน้าต่างเริ่มต้นของแอปพลิเคชัน ซึ่งสืบทอดจาก 'JPanel' และใช้ในการแสดงภาพพื้นหลังและปุ่มเริ่มเล่นเกม

- Game Class (Main Game Logic): คลาส Game เป็นคลาสหลักที่รับผิดชอบควบคุมเกมและหน้าต่างเกม โดยสืบทอดจาก 'JPanel' และมีการใช้ Composition เพื่อรวมหลายคลาสและองค์ประกอบ เช่น player, block, bullet, devil เข้าด้วยกันในการสร้างเกมและควบคุมการทำงานของเกม

### 2. Composition (การใช้งานสำหรับการรวมองค์ประกอบ):

โปรแกรมใช้ Composition เพื่อรวมหลายคลาสและองค์ประกอบเข้าด้วยกันในโปรแกรมหลัก Game ซึ่งจะสร้างและควบคุมสิ่งต่าง ๆ ที่เกี่ยวข้องกับเกม เช่น ผู้เล่น (player), object (block, bullet, devil), และองค์ประกอบเสริมอื่น ๆ

## 2.3 คลาสที่มีการใช้คุณสมบัติของ oop

### 2.2.1 คลาสที่มีการใช้ Constructor

```
public Game() {  
  
    this.setBounds(x:0, y:0, width:1000, height:600);  
    this.addKeyListener(this);  
    this.setFocusable(focusable:true);  
    this.setLayout(mgr:null);  
  
}
```

ในรูปเป็นตัวอย่างจากคลาสGame ซึ่งในโปรเจกต์มีการใช้ Constructor เกือบทุกคลาส

### 2.2.2 คลาสที่มีการใช้Encapsulation

```
public class Game extends JPanel implements ActionListener, KeyListener {  
  
    player P1 = new player(x:50, y:465, playersize:90);  
    block b1 = new block();  
    block b2 = new block();  
  
    long LastPress = 0;  
    int hp = 200;  
    int score = 0;  
    boolean hit = false;  
  
    private ImageIcon bg1 = new ImageIcon(this.getClass().getResource(name:"bg3.jpg"));  
    private ImageIcon bg2 = new ImageIcon(this.getClass().getResource(name:"5.jpg"));  
    private ImageIcon bg3 = new ImageIcon(this.getClass().getResource(name:"6.jpg"));  
    private ImageIcon bg4 = new ImageIcon(this.getClass().getResource(name:"7.jpg"));  
    private ImageIcon player1 = new ImageIcon(this.getClass().getResource(name:"p3.png"));  
    private ImageIcon land = new ImageIcon(this.getClass().getResource(name:"stone.png"));  
    private ImageIcon heal = new ImageIcon(this.getClass().getResource(name:"heart.png"));  
    private ImageIcon gameoverBG = new ImageIcon(this.getClass().getResource(name:"bgover.jpg"));  
    private ImageIcon btrestart = new ImageIcon(this.getClass().getResource(name:"btrestart.png"));  
}
```

ในรูปเป็นตัวอย่างจากคลาสGame ซึ่งในโปรเจกต์มีการใช้ Encapsulation ในทุกคลาส

### 2.2.3 คลาสที่มีการใช้ Inheritance

```
public class Game extends JPanel implements ActionListener, KeyListener {  
  
    player P1 = new player(x:50, y:465, playersize:90);  
    bullet1 = new bullett(x:115, P1.y+40);  
}
```

```
@Override  
public void keyPressed(KeyEvent e) {  
    // TODO Auto-generated method stub  
    if (System.currentTimeMillis() - LastPress > 600) {  
        if (e.getKeyCode() == KeyEvent.VK_SPACE) {  
            // Musicclass s1 = new Musicclass();  
            // s1.Bgsound();  
            System.out.println(x:"jump");  
            P1.jump(this);  
            this.repaint();  
        }  
  
        LastPress = System.currentTimeMillis();  
    }  
  
    if (e.getKeyCode() == KeyEvent.VK_RIGHT) {  
        bullet1.add(new bullett(x:115, P1.y+40));  
    }  
}
```

ในรูปเป็นตัวอย่างจากคลาสGame ซึ่งในโปรเจกมีการใช้ Inheritance จาก JPanel

### 2.2.4 คลาสที่มีการใช้Polymorphism

ไม่พบการใช้Polymorphismในทุกคลาสของโปรเจก

### 2.2.4 คลาสที่มีการใช้ interface-implement

```
public class Main extends JFrame implements ActionListener {  
    Display mg = new Display();  
    Game all = new Game();  
}
```

ในรูปเป็นตัวอย่างจากคลาสMain ซึ่งในโปรเจกมีการใช้ interface-implement ใน คลาส Main , Display และคลาส Game



### 2.2.5 คลาสที่มีการใช้ Composition

```
public class Main extends JFrame implements ActionListener {  
    Display mg = new Display();  
    Game all = new Game();  
}
```

ในรูปเป็นตัวอย่างจากคลาส Main ซึ่งมีการสร้าง object จากคลาส Display และ คลาส Game

## 2.4 หน้าจอ GUI

### 2.3.1 ส่วนประกอบของ GUI

1. JPanel (display,Game) สร้าง JPanel ที่ใช้เป็นพื้นที่ในการทำงานหลักของGUI ซึ่งในที่นี้ถูกใช้เพื่อแสดงภาพ,สร้างการแสดงผล และจัดการกับเหตุการณ์ต่างๆ เช่น KeyEvents

2.JFrame(Main) สร้าง JFrame ที่เป็นหน้าต่างหลักของแอปพลิเคชัน GUI โดยที่มีการกำหนดขนาด,ชื่อ และ ตำแหน่งของหน้าต่าง

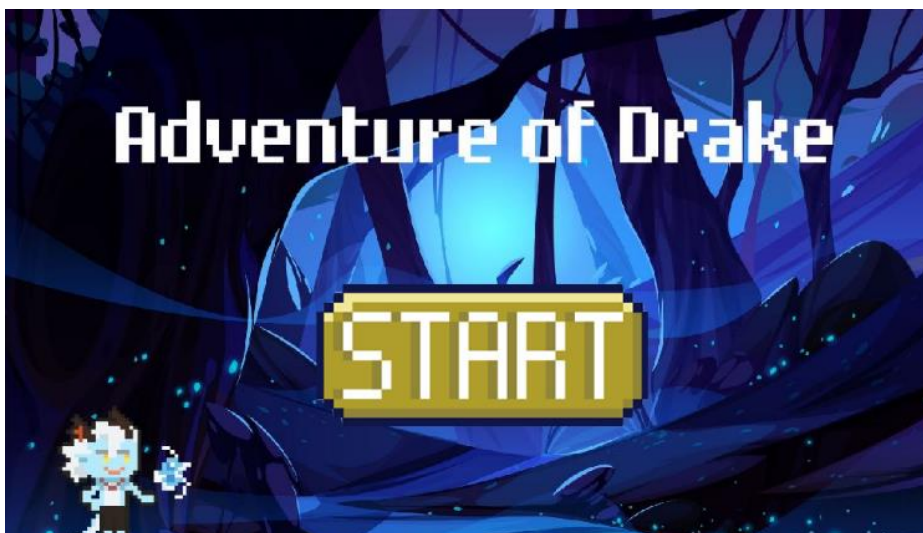
3.Graphic2D สำหรับการวาดภาพและกราฟิกต่างๆ บนPanel

4.Character Package มีคลาส Player devile jaii และ block ที่ใช้ในการสร้างวัตถุต่างๆในเกม

5.Element Package ประกอบด้วยองค์ประกอบที่ใช้ในการสร้างข้อความและภาพต่างๆ

6.Button มีการใช้ปุ่มbutton โดยการใช้ keylistener

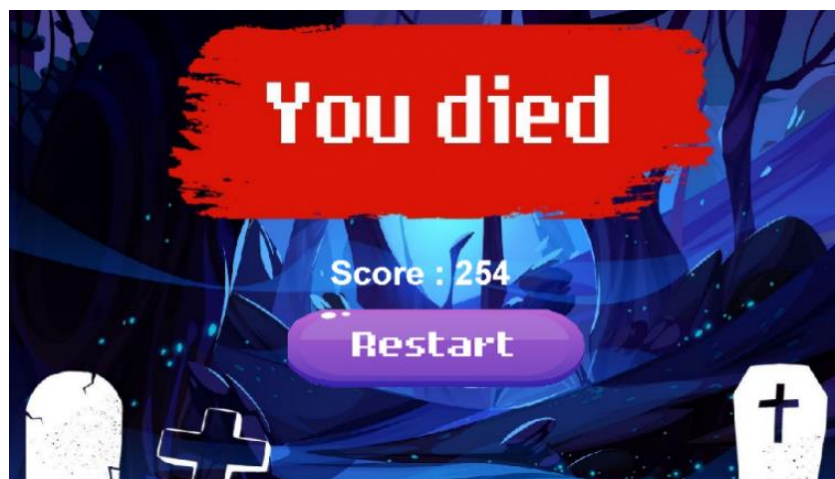
-หน้าจอ Play ก่อนเริ่มเกม จะมีปุ่ม PLAY เพื่อให้ผู้เล่นกด และ มีการวางชื่อเกม,ตัวละคร



-หน้าเล่นเกมก็จะมีการวาง HP อยู่บนซ้าย Score อยู่บนขวาและมีการวางตัวละคร และ object อยู่ในพื้นที่ของฉาก  
แล้วก็จะเปลี่ยนbackgroundเมื่อscore ถึงที่ตั้งไว้



-ส่วนหน้าจอก็จะมีการจัดlayoutข้อความไวตรงกลาง มีคะแนน และมีปุ่มRestartให้กดกลับไปเริ่มเกมใหม่



## 2.5 Event handling

- Main Class (Main): คลาสหลักที่สืบทอดจาก JFrame และใช้ในการสร้างหน้าต่างหลักของแอปพลิเคชัน โดยมีปุ่มเริ่มเกมและสลับไปยังหน้าจอหลักเมื่อผู้ใช้คลิกปุ่ม.
- Display Class (mg): คลาสที่ใช้ในการสร้างหน้าจอเริ่มเกม โดยมีปุ่ม "เริ่มเกม" ที่ผู้ใช้สามารถคลิกเพื่อเริ่มเกม.
- Game Class (all): คลาสที่ใช้ในการสร้างหน้าจอเกม และจัดการกับการแสดงผลเกม, การเคลื่อนที่ของตัวละคร, การสร้างวัตถุ, การจับเหตุการณ์ของคีย์บอร์ด, และการตรวจสอบการชน.
- ActionListener และ KeyListener: การจัดการกับเหตุการณ์การคลิกปุ่มและการกดคีย์บอร์ด โดยใช้ ActionListener และ KeyListener เพื่อตรวจสอบเหตุการณ์และดำเนินการตามที่กำหนด.
- รูปภาพและไอคอน: การโหลดรูปภาพและไอคอนที่ใช้ในการแสดงผลบนหน้าจอ.
- การสร้างอ็อบเจกต์เกม: การสร้างอ็อบเจกต์ที่ใช้ในการควบคุมเกม เช่น ตัวละครของผู้เล่น, วัตถุในเกม, และกระบวนการที่เกี่ยวข้องกับเกม.
- การจัดการกับการแสดงผล: การใช้ Graphics และ Graphics2D เพื่อวาดรูปภาพและแสดงผลบนหน้าจอ GUI.
- การจัดการกับเหตุการณ์การชน: การตรวจสอบการชนของอ็อบเจกต์ในเกม เช่น การชนของกระสุนกับศัตรู.

## 2.6 Algorithm

1. การจัดการตำแหน่งและเคลื่อนที่ของตัวละคร : โค้ดมีคลาส player ที่จัดการกับตำแหน่งและเคลื่อนที่ของตัวละครผู้เล่น (P1) ในเกม. อัลกอริทึมนี้เป็นรูปแบบการควบคุมตำแหน่งของตัวละครและการเคลื่อนที่ของตัวละครเมื่อผู้เล่นกระทำ
2. การสร้างวัตถุในเกม : โค้ดมีการสร้างวัตถุอื่น ๆ เช่น block (ลักษณะของสิ่งกีดขวางในเกม), bullet (กระสุนที่ผู้เล่นยิง), และ devil (ศัตรูในเกม)

3. การตรวจสอบการชน : โค้ดมีการตรวจสอบการชนระหว่างอ็อบเจกต์ต่าง ๆ ในเกม เช่น การตรวจสอบการชนระหว่างตัวละครผู้เล่นกับวัตถุที่สร้างขึ้น หรือ การตรวจสอบการชนระหว่างกระสุนและศัตรู. อัลกอริทึมนี้ช่วยให้โปรแกรมรู้ว่าอ็อบเจกต์ไหนที่ชนกัน.

4. การจัดการกับเหตุการณ์การคลิกและการกดคีย์บอร์ด: โค้ดใช้ ActionListener และ KeyListener เพื่อตรวจจับเหตุการณ์ที่เกิดขึ้นเมื่อผู้เล่นคลิกปุ่มหรือกดคีย์บอร์ด. อัลกอริทึมนี้ช่วยให้โปรแกรมสามารถตรวจจับการกระทำของผู้เล่นและรับคำสั่ง.

5. การจัดการกับแสดงผลบนหน้าจอ: โค้ดใช้ Java Swing ในการจัดการกับการแสดงผลบนหน้าจอ GUI โดยใช้ Graphics และ Graphics2D เพื่อวาดรูปภาพ, ตัวละคร, และวัตถุบนหน้าจอ.

6. การสร้างเกมลูป : โค้ดใช้ Thread เพื่อสร้างเกมลูปที่รันอัลกอริทึมที่เกี่ยวข้องกับการเคลื่อนที่ของศัตรู และการอัปเดตสถานะเกมในระหว่างเกม.

7. การสร้างหน้าจอเริ่มเกม : โค้ดมีการสร้างหน้าจอเริ่มเกม (Start Screen) โดยใช้ JFrame และปุ่ม "เริ่มเกม" ซึ่งเมื่อผู้เล่นคลิกจะสลับไปยังหน้าจอหลัก.

## บทที่ 3

### สรุป

#### 3.1 ปัญหาที่พบระหว่างการพัฒนา

1. ความรู้พื้นฐานยังไม่แน่นพอ
2. โค้ดเยอะมากจนทำให้ตาลาย
3. บางครั้งเขียนโค้ดผิดจนทำให้เกมรวน
4. จัด layout ได้ไม่สวยและจัดยาก
5. ใส่เสียงแล้ว error

#### 3.2 จุดเด่นของโปรแกรมที่ไม่เหมือนใคร

- มีกราฟิกที่สวยงามและสีสันที่น่าสนใจ

#### 3.2 คำแนะนำ

อยากให้อาจารย์ไม่ต้องสอนรวบเดียวจบ ให้มีเวลาพักสัก 5-10 นาที ค่อยสอนต่อครับ