



Object-oriented Programming mini project

“Desty Space”

จัดทำโดย

6504062630197 ฟ้าใส แต้มฤทธิ์

เสนอ

ผู้ช่วยศาสตราจารย์สถิต ประสมพันธ์

วิชาการเขียนโปรแกรมเชิงวัตถุ (Object-oriented Programming)

ภาควิชาวิทยาการคอมพิวเตอร์และสารสนเทศ คณะวิทยาศาสตร์ประยุกต์

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

ภาคเรียนที่ 1/2566

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญ

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

1.2 ประโยชน์

1.2.1 ฝึกวิธีแก้ปัญหา

1.2.2 เพื่อฝึกความอดทน

1.2.3 ฝึกการเขียนโปรแกรม

1.2.4 ฝึกคิดเป็นระบบขั้นตอน

1.3 ขอบเขตของโครงการ

เกม Desty Space เป็นเกมยิงสิ่งกีดขวาง เพื่อทำการเก็บแต้ม และหลบสิ่งกีดขวางเพื่อไม่ให้ชีวิตลด ถ้าชีวิตหมดเกมก็จะจบทันที

1.3.1 ประเภทของโครงการ

โครงการเป็นประเภทเกมที่สร้างจากภาษา java โดยมีเนื้อหาและ concept ของ OOP

ตารางแผนการทำงานเดือนตุลาคม - พฤศจิกายน

ลำดับ	รายการ	วันที่ 1-12	วันที่ 13-31	วันที่ 1-7
1	หารูปและฉากหลังต่างๆ			
2	ศึกษาเอกสารข้อมูลที่เกี่ยวข้อง			
3	ลงมือเขียนโปรแกรม			
4	จัดทำเอกสาร			
5	ตรวจสอบข้อผิดพลาดต่างๆ			

1.4 Storyboard และวิธีการเล่น

- รายละเอียดเกม

เกม Desty Space คุณต้องทำการบังคับควบคุม คอยหลบและยิงอุกกาบาต ที่พร้อมจะทำร้ายคุณ เมื่อคุณยิงอุกกาบาตจะมีการสะสมแต้มไปเรื่อย ซึ่งอุกกาบาตขนาดต่างกัน แต้มก็ต่างกันด้วย และคุณสามารถเก็บเกาะป้องกัน เพื่อป้องกันไม่ให้จรวดของคุณพังทลายได้

- วิธีการเล่น

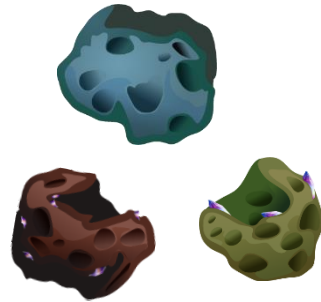
ใช้ปุ่ม “A” , “D” หรือปุ่มลูกศรซ้ายขวา เพื่อบังคับทิศทาง

- Storyboard

ตัวละคร

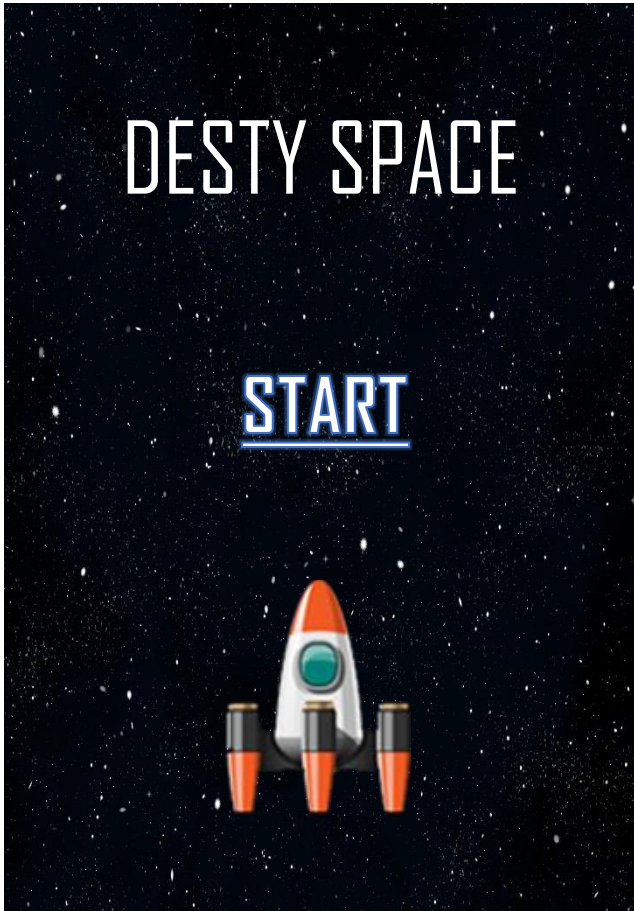


จรวด

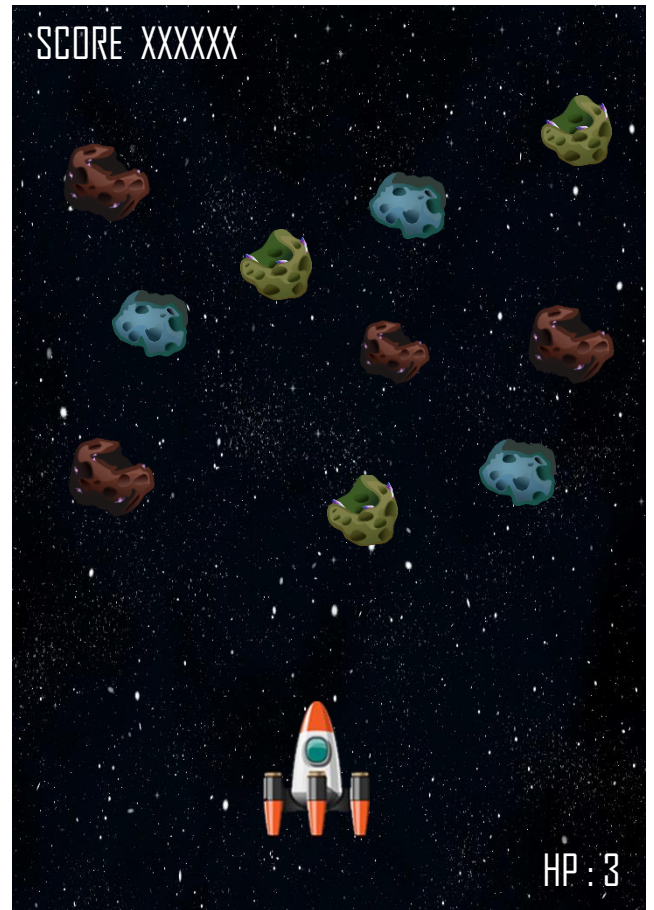


สิ่งกีดขวาง

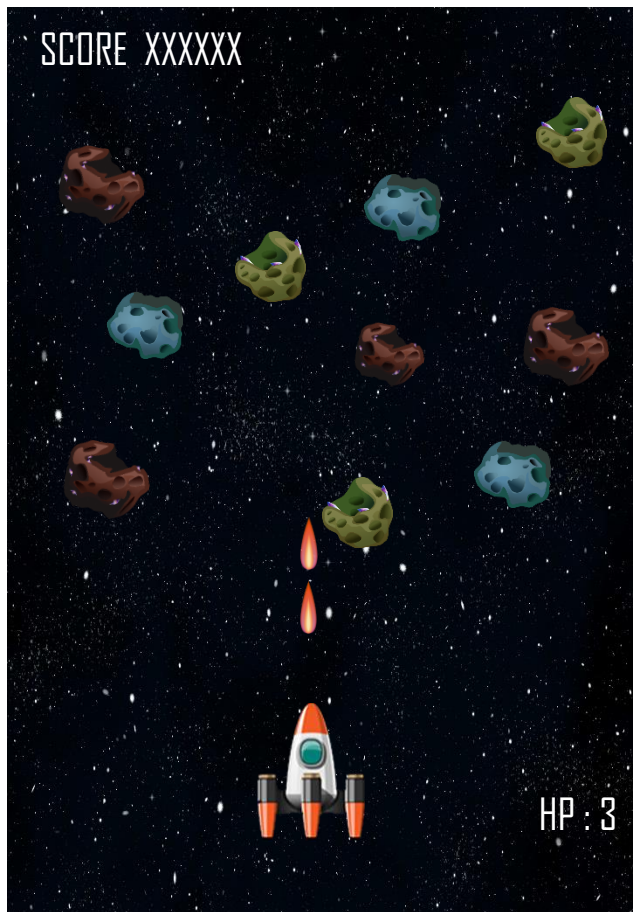
ฉาก



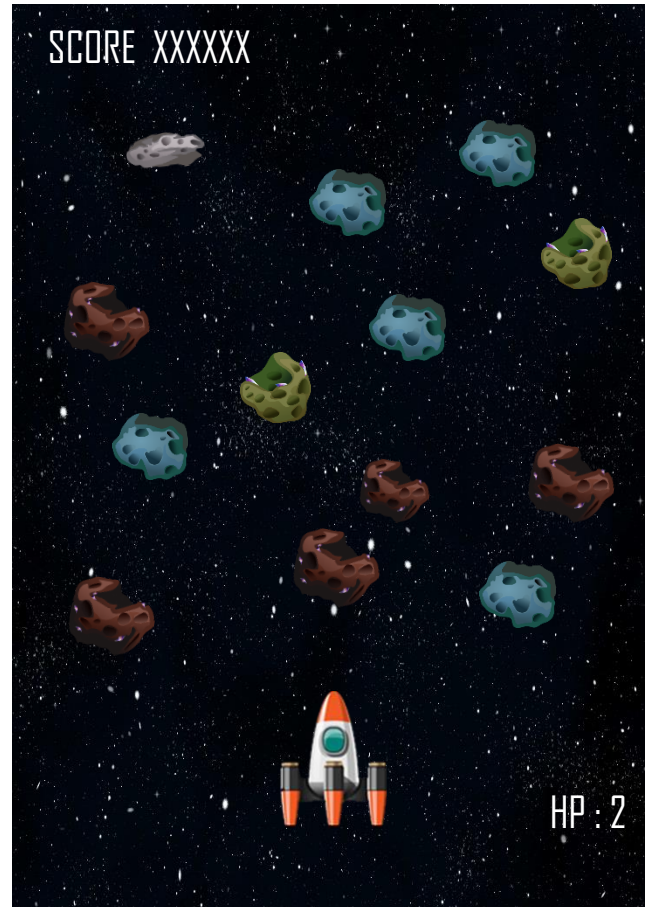
เริ่มเกม



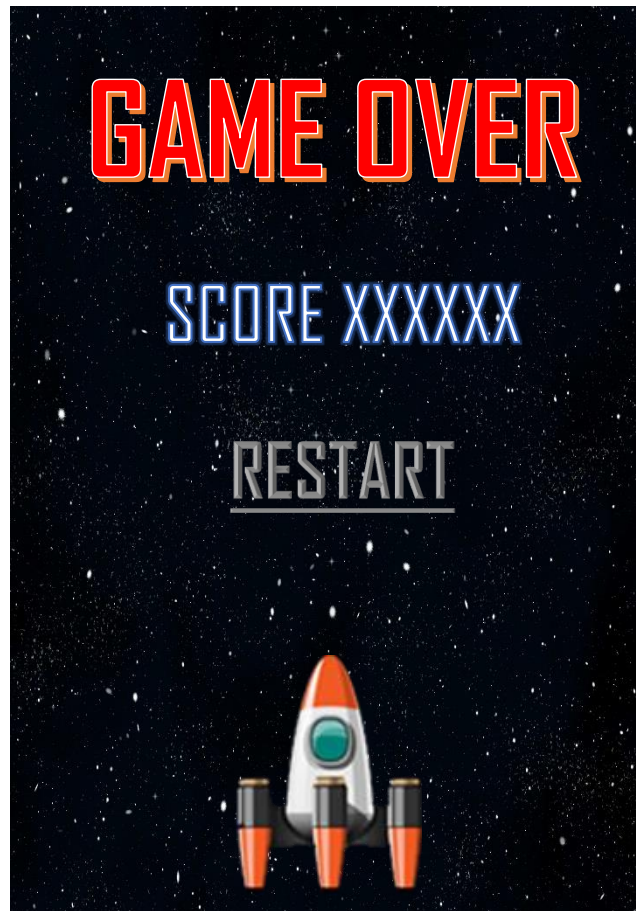
ทำการบังคับซ้ายขวาเพื่อหลบสิ่งกีดขวาง



สามารถยิงสิ่งกีดขวางได้



เมื่อจรวดชนกีดขวาง HP ก็จะลด



เมื่อจรวดชนสิ่งกีดขวาง ก็จะระเบิดแล้วจบเกมทันที

- ประโยชน์ของเกม

- เพื่อความสนุกสนาน

- ช่วยผ่อนคลาย

- ตารางแผนการทำงานเดือนกันยายน และตุลาคม

ลำดับ	รายการ	16-30	1-10	11-20	21-31
1	ทำกราฟิกตัวละครและออกแบบ				
2	ศึกษาวิธีการเขียน โปรแกรม และค้นคว้าเอกสารที่เกี่ยวข้อง				
3	ลงมือเขียน โปรแกรม				
4	จัดทำเอกสาร				
5	ตรวจสอบ และแก้ไขข้อผิดพลาด				

บทที่ 2

ส่วนของการพัฒนา

2.1 เนื้อเรื่องย่อ และวิธีการเล่น

รายละเอียดเกม

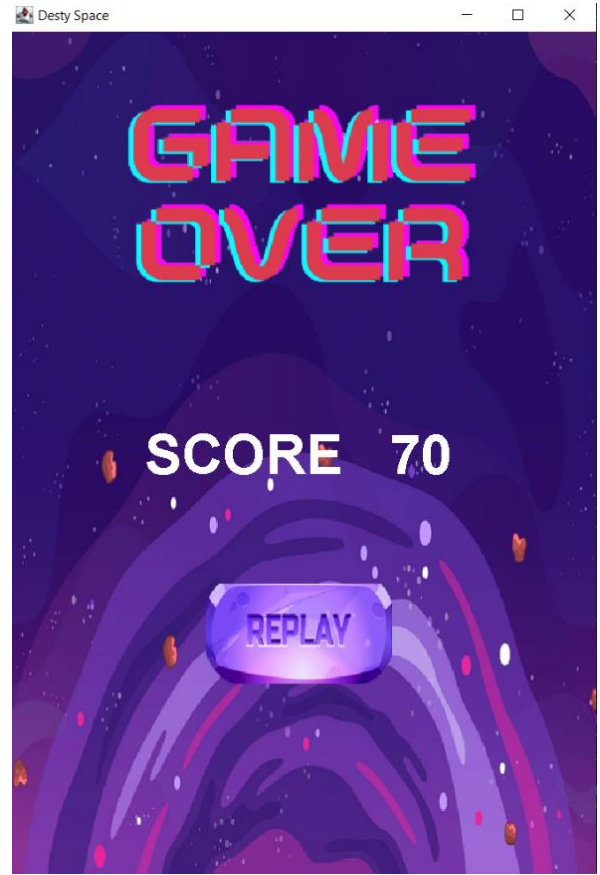
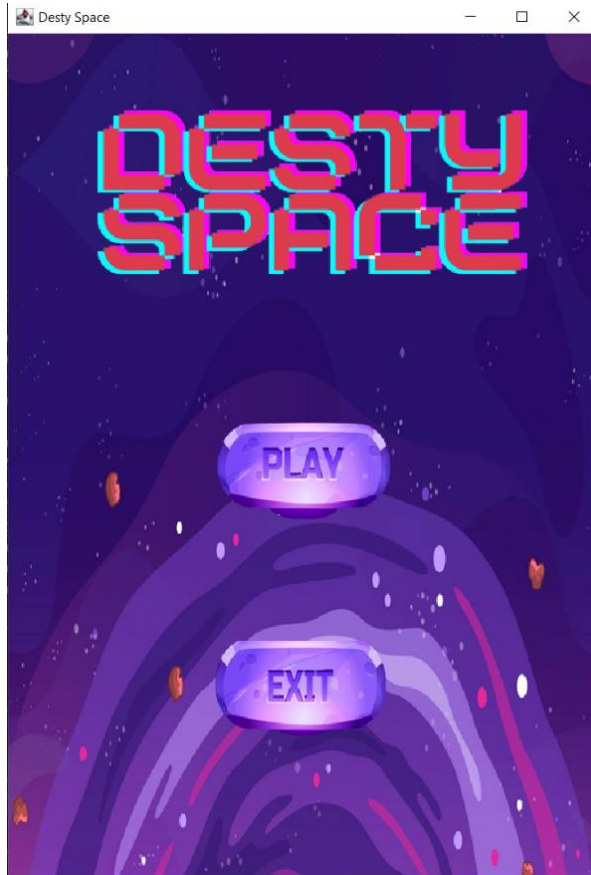
เกม Desty Space คุณต้องทำการบังคับควบคุม คอยหลบและยิงอุกกาบาต ที่พร้อมจะทำร้ายคุณ เมื่อคุณยิงอุกกาบาตจะมีการสะสมแต้มไปเรื่อย ๆ แต่ถ้าคุณยิงโดนระเบิด แต้มก็จะลด แล้วก็ต้องคอยหลบอุกกาบาต และระเบิดด้วย เพื่อไม่ให้ชีวิตลดหมดจนตาย

วิธีการเล่น

1. ใช้ปุ่ม “A” และ “D” เพื่อทำการบังคับทิศทางของจรวด
2. กดลูกศรขึ้นบน เพื่อทำการยิงกระสุนยิงอุกกาบาต
3. เกมจะนับ Score จากการที่เรายิงอุกกาบาตได้ แต่ถ้ายิงโดนระเบิด Score ก็จะเป็น -10
4. ต้องบังคับจรวดเพื่อหลบสิ่งกีดขวางด้วย ถ้าโดน HP จะลดและจบเกมเมื่อ HP หมด

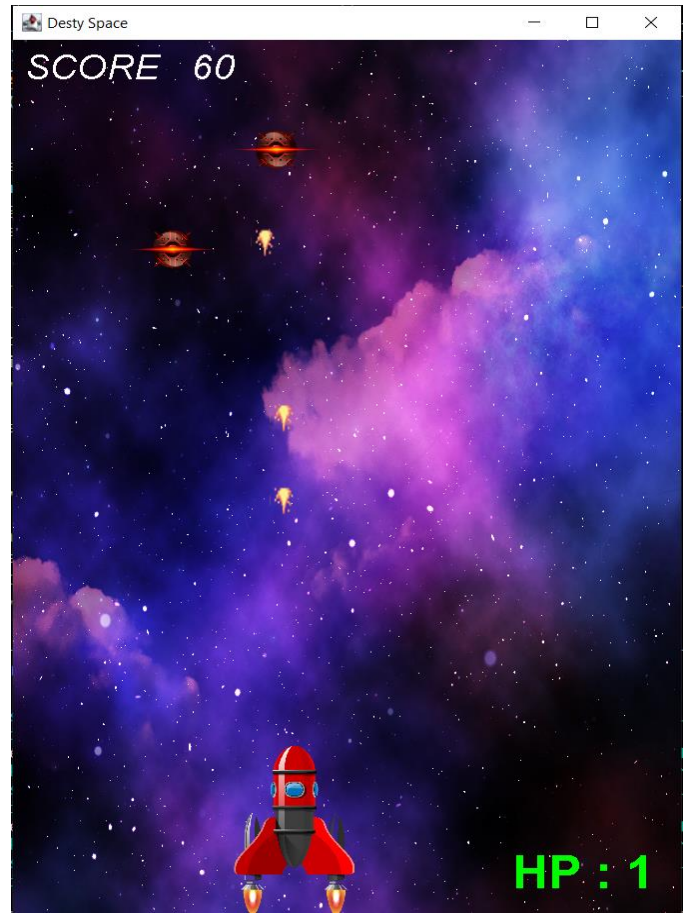
ฉากภายในเกม

1. หน้าเริ่มเกม และหน้าจบเกม

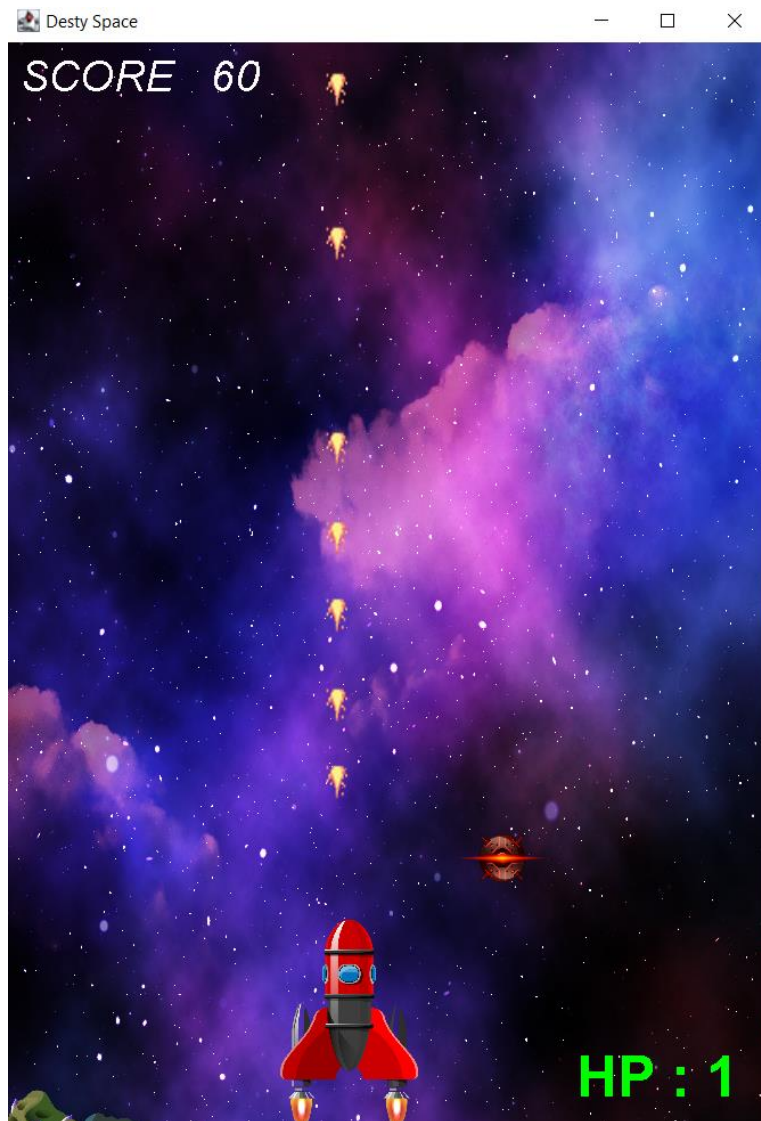


2. ฉากภายในเมื่อเริ่มเกม

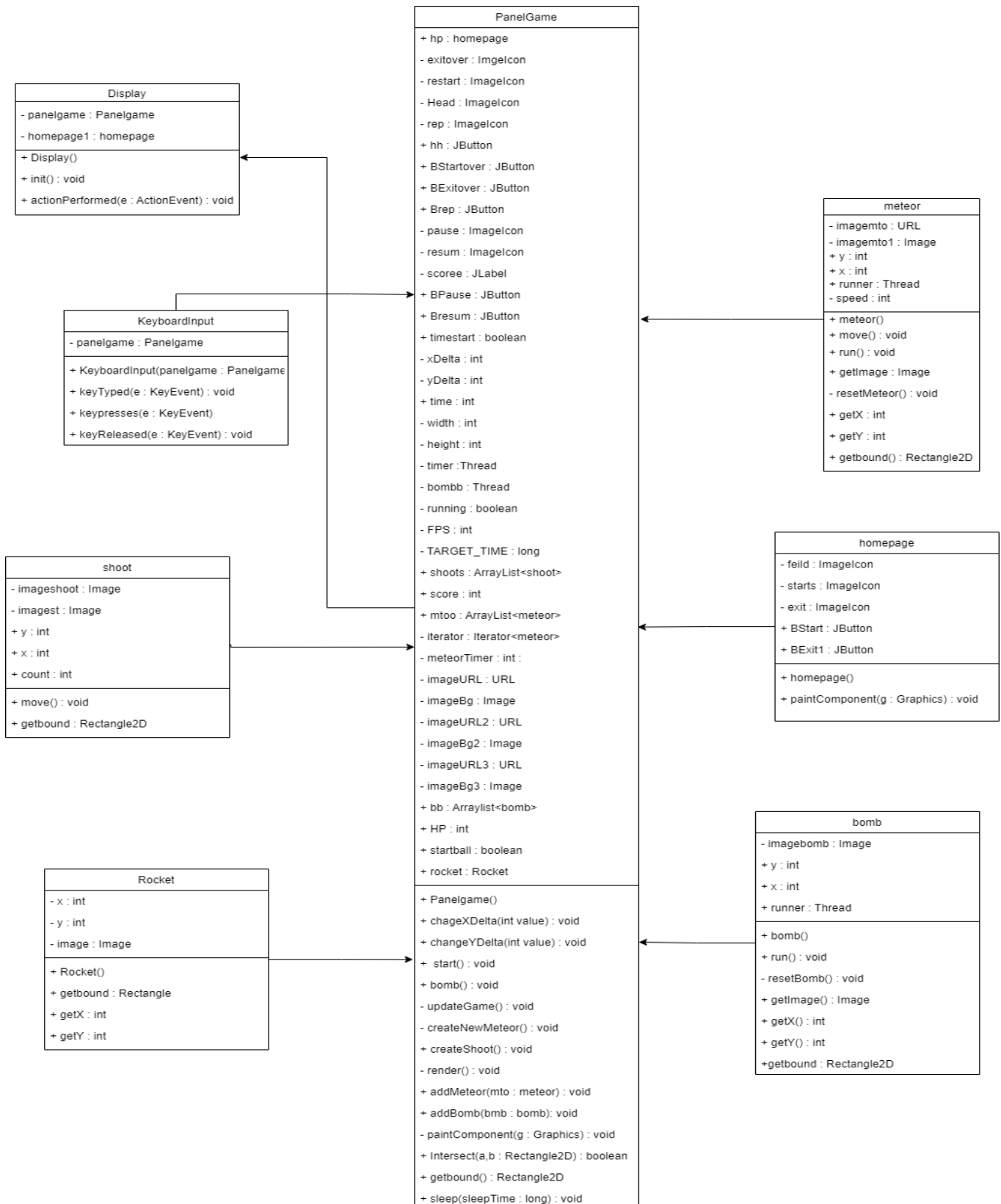
เมื่อคะแนนเท่ากับ 0 หรือน้อยกว่า 50 และ คะแนนมากกว่าหรือเท่ากับ 50



3. ฉากยิงกระสุนภายในเกม



2.2 Class diagram



2.3 อธิบายการทำงานของโปรแกรม

2.3.1 Class Display

Display เป็นคลาสหลักของแอปพลิเคชัน ซึ่ง extends จาก JFrame และมีการรวม PanelGame และ homepage ซึ่งเป็นคลาสอื่น ๆ ที่ใช้สร้างหน้าต่างและส่วนต่าง ๆ ของ Application

2.3.2 Class PanelGame

PanelGame เป็นส่วนสำคัญในโปรแกรม "Desty Space" ที่ใช้ Swing สำหรับสร้างพื้นที่เกมและแสดงเกม มี Method สำคัญดังนี้

- รับควบคุมจรวดและยิงกระสุน
- จัดการ meteors ที่สร้างขึ้น และปล่อยลงมาจากด้านบนของหน้าจอ
- สร้างกระสุนและตรวจสอบการชนกับหินและระเบิด
- แสดงคะแนนและจำนวนชีวิตที่เหลือบนหน้าจอ

2.3.3 Class homepage

คลาส homepage คือคลาสที่สืบทอดมาจาก JPanel คือโค้ดที่สร้างหน้าจอหลักของเกม และปุ่มเพื่อให้ผู้ใช้สามารถเริ่มเกมหรือออกจากเกมได้ โดยแสดงภาพพื้นหลังและกำหนดรูปแบบของปุ่มให้ดูดีและใช้งานง่าย

2.3.4 Class KeyboardInput

คลาส KeyboardInput ที่ implements KeyListener เพื่อจัดการอีเวนต์ที่เกิดขึ้นเมื่อกดปุ่มบนคีย์บอร์ด นอกจากนี้ยังรับ PanelGame เข้ามาใน constructor เพื่อทำให้สามารถเปลี่ยนสถานะของ

ถ้ากดปุ่ม A, จะเรียกเมทอด changeXDelta(-15) ของ PanelGame เพื่อเปลี่ยนค่าพิกัด X ทางซ้าย

ถ้ากดปุ่ม D, จะเรียกเมทอด changeXDelta(15) ของ PanelGame เพื่อเปลี่ยนค่าพิกัด X ทางขวา

2.3.5 Class shoot

คลาส shoot ที่ extends จาก JPanel ใน Java Swing เพื่อแสดงรูปภาพของการยิงในเกม จะถูกใช้ในการพัฒนาเกมที่เกี่ยวข้องกับกราฟิก โดยวัตถุนี้สามารถเคลื่อนที่ขึ้นมามบนหน้าจอและมีการใช้ในการตรวจสอบการชนกับวัตถุอื่น ๆ ในเกม

2.3.6 Class Rocket

คลาส Rocket ที่ใช้ในการจัดการข้อมูลและการวาดรูปภาพของ "จรวด" ในเกม สามารถสร้างขึ้นและนำไปใช้ในการแสดงผลบนหน้าจอได้

public Rectangle getbound() สร้างและคืนค่า Rectangle ที่เป็นกล่องคำนวณขอบเขตของใช้เพื่อตรวจสอบการชนกับวัตถุอื่น ๆ ในเกม.

2.3.7 Class meteor

การทำงานสำคัญ ใน Constructor (public meteor()), ทำการกำหนดค่าเริ่มต้นของตำแหน่ง x และ y, และเริ่ม thread (runner.start()) สำหรับการเคลื่อนที่

public void move(): เคลื่อนที่ลงมาด้านล่างด้วยความเร็วที่กำหนด, และเมื่อmeteorอยู่นอกขอบเขตหรือชนขอบสูงสุด, จะทำการรีเซ็ตตำแหน่ง

private void resetMeteor(): ทำการรีเซ็ตตำแหน่ง x และ y ใหม่เมื่ออยู่นอกขอบเขตหรือชนขอบ

public Rectangle2D getbound(): คืนค่า Rectangle2D ที่เป็นกล่องคำนวณขอบเขต

2.3.7 Class bomb

คลาสนี้ทำหน้าที่ควบคุมการเคลื่อนไหวของระเบิดที่ตกลงจากข้างบนบน มีการใช้ AWT ในการจัดการกราฟิกและเชอร์คเพื่อควบคุมเคลื่อนไหวของระเบิด เชอร์คทำงานในรูปแบบไปเรื่อย ๆ, อัปเดตตำแหน่ง y ของระเบิดให้มันเคลื่อนลง, และถ้าตกลงมานอกขอบเขตหรือนอกกรอบทางนอน, จะทำการรีเซ็ตตำแหน่ง. มีการตั้ง sleeptime เพื่อช่วยให้ระเบิดตกลงอย่างช้าๆ

2.4 คลาสที่มีการใช้คุณสมบัติของ OOP

2.4.1 คลาสที่มีการใช้ Constructor

```
public class PanelGame extends JPanel{

    public homepage hp = new homepage();
    private ImageIcon exitover = new ImageIcon(this.getClass().getResource(name:"photo/exist.png"));
    private ImageIcon restart = new ImageIcon(this.getClass().getResource(name:"photo/start.png"));
    private ImageIcon Head = new ImageIcon(this.getClass().getResource(name:"photo/head.png"));
    private ImageIcon rep = new ImageIcon(this.getClass().getResource(name:"photo/replay.png"));
    public JButton hh = new JButton(Head);
    public JButton BStartover = new JButton(restart);
    public JButton BExitover = new JButton(exitover);
    public JButton Brep = new JButton(rep);
```

ในรูปแบบเป็น Constructor จากคลาส Panelgame

```
public Display(){
|   init();
}
```

ในรูปแบบเป็น Constructor จากคลาส Display

```
public homepage(){
|   setLayout(mgr:null);
|   BExit1.setBounds(x:180, y:550, width:170,height:90);
|   add(BExit1);
|   BExit1.setBackground(new Color(r:0, g:0, b:0, a:0));
|   BExit1.setBorderPainted(b:false); // ปิดการวาดเส้นขอบ
```

ในรูปแบบเป็น Constructor จากคลาส homepage

```
public bomb() {
|   String imageLocation = "photo/PNG/Sprites/Bombs/bomb1.png";
|   URL imageURL1 = this.getClass().getResource(imageLocation);
|   imagebomb = Toolkit.getDefaultToolkit().getImage(imageURL1);
|   runner.start();
}
```


ในรูปแบบ Constructor จากคลาส bomb

```
public meteor(){  
    x = (int) ((Math.random() * 300) + 20);  
    y = 0;  
    runner.start();  
}
```

ในรูปแบบ Constructor จากคลาส meteor

```
public Rocket() {  
    URL imageUrl = this.getClass().getResource(name:"photo/rocket.png");  
    this.image = new ImageIcon(imageUrl).getImage();  
}
```

ในรูปแบบ Constructor จากคลาส Rocket

2.4.2 คลาสที่มีการใช้ Encapsulation

```
public class PanelGame extends JPanel{  
    public homepage hp = new homepage();  
    private ImageIcon exitover = new ImageIcon(this.getClass().getResource("photo/exist.png"));  
    private ImageIcon restart = new ImageIcon(this.getClass().getResource("photo/start.png"));  
    private ImageIcon Head = new ImageIcon(this.getClass().getResource("photo/head.png"));  
    private ImageIcon rep = new ImageIcon(this.getClass().getResource("photo/replay.png"));  
    public JButton hh = new JButton(Head);  
    public JButton BStartover = new JButton(restart);  
    public JButton BExitover = new JButton(exitover);  
    public JButton Brep = new JButton(rep);  
  
    private ImageIcon pause = new ImageIcon(this.getClass().getResource("photo/pause.png"));  
    private ImageIcon resum = new ImageIcon(this.getClass().getResource("photo/start.png"));  
  
    private JLabel scoree = new JLabel();  
    public JButton BPause = new JButton(pause);  
    public JButton Bresum = new JButton(resum);  
    public boolean timestart = true;
```

ในรูปแบบตัวอย่างจากคลาส Panelgame ซึ่งในโปรเจกมีการใช้ Encapsulation ในทุกคลาส

2.4.3 คลาสที่มีการใช้ Inheritance

```
public class Display extends JFrame implements ActionListener{
    private PanelGame panelgame = new PanelGame();
    private homepage homepage1 = new homepage();

    public Display(){
        init();
    }
}
```

ในรูปเป็นตัวอย่างจากคลาส Display ซึ่งในโปรเจกต์มีการใช้ Inheritance ในคลาส Panelgame , Display , homepage

2.4.4 คลาสที่มีการใช้ interface-implement

```
public class KeyboardInput implements KeyListener {
    private PanelGame panelgame;
    public KeyboardInput(PanelGame panelgame){
        this.panelgame = panelgame;
    }
    @Override
    public void keyTyped(KeyEvent e) {

    }
}
```

ในรูปเป็นตัวอย่างจากคลาส KeyboardInput ซึ่งในโปรเจกต์มีการใช้ interface-implement ในคลาส Display , KeyboardInput

2.5 GUI

GUI ในเกมสร้างขึ้นด้วย Java Swing

หน้าจอหลัก (PanelGame): มีพื้นหลังที่ใช้รูปภาพด้านบนของหน้าจอ, แบ่งเป็นสองส่วน โดยมีรูปภาพเป็นพื้นหลังในแต่ละส่วน จอเกมแสดงผลตัวละครหลัก (Rocket), meteor, กระจุน, bomb, และคะแนน มีปุ่มควบคุมเกม เช่นปุ่มเริ่มเกมใหม่, ปุ่มหยุดเกม, และปุ่มเล่นใหม่

หน้าจอหน้าแรก (homepage): มีพื้นหลังที่ใช้รูปภาพเตรียมไว้สำหรับหน้าแรกของเกม ประกอบด้วยปุ่มสำหรับเริ่มเล่นเกม (Start) และปุ่มออกจากเกม (Exit)

การจัดการเหตุการณ์ (Event Handling): ใน KeyboardInput class มีการใช้ KeyListener เพื่อรับเหตุการณ์ที่เกิดขึ้นจากคีย์บอร์ดสำหรับการควบคุมตัวละคร มีการจัดการเหตุการณ์ของปุ่มใน GUI เพื่อเริ่มเกม, หยุดเกม, และเล่นใหม่

การแสดงผลและกราฟิก: มีการใช้รูปภาพที่ประกอบเกี่ยวกับตัวละคร, meteor, bomb, และพื้นหลังเพื่อสร้างประสบการณ์ทางภาพ ใช้ JLabel เพื่อแสดงคะแนน, ข้อความ, และคำสั่ง

การจัดการเวลา: ใช้ Thread เพื่อควบคุมการทำงานของเกมและวัตถุต่าง ๆ ระบบเวลาทำให้เกมทำงานได้อย่างเรียลไทม์

2.6 Event handling

KeyboardInput Class: ใช้ KeyListener เพื่อตรวจจับการกดปุ่มบนคีย์บอร์ด

ถ้าปุ่ม A ถูกกด, จะเรียก panelgame.changeXDelta(-15) เพื่อเปลี่ยนค่า xDelta ใน PanelGame ลดลง 15

ถ้าปุ่ม D ถูกกด, จะเรียก panelgame.changeXDelta(15) เพื่อเปลี่ยนค่า xDelta ใน PanelGame เพิ่มขึ้น 15

ถ้าปุ่ม UP arrow ถูกกด, จะเรียก panelgame.createShoot() เพื่อสร้างกระจุนใหม่

PanelGame Class: ใช้ KeyListener เพื่อรับข้อมูลจากคีย์บอร์ด

ในเมทอด updateGame(), มีการตรวจสอบ meteorTimer เพื่อสร้าง meteor ใหม่

ในเมทอด paintComponent(), มีการตรวจสอบการกดปุ่ม UP arrow และทำการเรียก createShoot() เพื่อสร้างกระสุน

```
private void updateGame() {  
    meteorTimer++;  
    if (meteorTimer >= 100) {  
        meteorTimer = 0;  
        createNewMeteor();  
    }  
  
    for (int i = 0; i < mtoo.size(); i++) {  
        meteor mto = mtoo.get(i);  
        mto.move();  
    }  
}
```

PanelGame Class (paintComponent method): ในการตรวจสอบการชน ระหว่างกระสุนกับ meteor และ bomb ในการตรวจสอบการชนระหว่างจรวดกับ meteor และ bomb

```
//กระสุนชนmeteor  
if (!shoots.isEmpty()){  
    for (int i = 0; i < shoots.size(); i++) {  
        for (int j = 0; j < mtoo.size(); j++) {  
            if (Intersect(shoots.get(i).getbound(), mtoo.get(j).getbound())) {  
                mtoo.remove(j);  
                shoots.remove(i);  
                i--;  
                score += 10;  
                g.drawString(str:"+10", xDelta+210 , y:310);  
                break;  
            }  
        }  
    }  
}
```

PanelGame Class (constructor): เรียก addKeyListener(new KeyboardInput(this)) เพื่อเพิ่ม KeyListener ใน PanelGame

```
public class PanelGame extends JPanel{

    public homepage hp = new homepage();
    private ImageIcon exitover = new ImageIcon(this.getClass().getResource("photo/exist.png"));
    private ImageIcon restart = new ImageIcon(this.getClass().getResource("photo/start.png"));
    private ImageIcon Head = new ImageIcon(this.getClass().getResource("photo/head.png"));
    private ImageIcon rep = new ImageIcon(this.getClass().getResource("photo/replay.png"));
    public JButton hh = new JButton(Head);
    public JButton BStartover = new JButton(restart);
    public JButton BExitover = new JButton(exitover);
    public JButton Brep = new JButton(rep);

    private ImageIcon pause = new ImageIcon(this.getClass().getResource("photo/pause.png"));
    private ImageIcon resum = new ImageIcon(this.getClass().getResource("photo/start.png"));

    private JLabel scoree = new JLabel();
    public JButton BPause = new JButton(pause);
    public JButton Bresum = new JButton(resum);
    public boolean timestart = true;
```

2.7 Algorithm

การเคลื่อนที่ของวัตถุ: ในคลาส meteor, bomb, และ shoot มี algorithm ที่ใช้ในการอัปเดตตำแหน่งของวัตถุต่าง ๆ โดยใช้ค่าความเร็ว, เวลาผ่านไป, และตำแหน่งปัจจุบันเพื่อคำนวณตำแหน่งใหม่.

การตรวจสอบการชน: ในคลาส PanelGame, มี method Intersect(Rectangle2D a, Rectangle2D b) ที่ใช้ในการตรวจสอบการชนระหว่างสองวัตถุ Method นี้ใช้ Rectangle2D เพื่อระบุขอบเขตของวัตถุและตรวจสอบว่ามีการชน

การสร้างวัตถุ: ในคลาส PanelGame, มี method createNewMeteor() ที่ใช้ในการสร้างวัตถุ meteor ใหม่ วัตถุจะถูกสร้างในตำแหน่งสุ่มบนแนวนอน

การจัดการเวลา: ในเมทอด start() ของคลาส PanelGame, มี algorithm ที่ใช้ในการควบคุมการทำงานของเกมในแต่ละ frame โดยใช้ Thread เพื่อคำนวณเวลาที่ใช้ในการอัปเดตและแสดงผล

การสร้างกระสุน: ในเมทอด createShoot() ของคลาส PanelGame, มี algorithm ที่ใช้ในการสร้างกระสุนใหม่

บทที่ 3

สรุป

3.1 ปัญหาที่พบระหว่างการพัฒนา

1. Meteor ไม่ปล่อยออกมา
2. มีปัญหากับ Arraylist
3. จรวดยังไม่ออก
4. ปัญหาเรื่อง UI
5. ความรู้การใช้ OOP ไม่เพียงพอ

3.2 จุดเด่นของโปรแกรม

มีจรวดที่สามารถยิงได้ ระบบการปล่อย meteor และ bomb ที่ทำลาย มีการจัดการความยากขึ้นเรื่อย ๆ ด้วยการเพิ่ม meteor และ bomb ลงมา

3.3 คำแนะนำสำหรับผู้สอนที่อยากให้อธิบาย หรือที่เรียนแล้วไม่เข้าใจ หรืออยากให้เพิ่มสำหรับน้องๆรุ่นต่อไป

อาจารย์สอนดี ละเอียด เข้าใจง่ายแล้วค่ะ แต่อยากให้มีความพักระหว่างเรียนสักนิดนึงค่ะ