



# Game Project

ถูกเก็บของ

จัดทำโดย

6404062630546 นางสาวชิรณัฐ พรรณา

เสนอ

ผู้ช่วยศาสตราจารย์ สถิต ประสมพันธ์

## Object Oriented Programming

คณะวิทยาศาสตร์ประยุกต์ ภาควิชาวิทยาการคอมพิวเตอร์และสารสนเทศ

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

ภาคการศึกษาที่ 1 ปี 2565

# บทที่ 1 บทนำ

## ที่มาและความสำคัญของโครงการ

โครงการนี้เกิดขึ้นจากการนำสิ่งที่ได้เรียนรู้ตลอดทั้งภาคการเรียนที่ 1 ปีการศึกษา 2565 มาจัดทำขึ้นเป็นเกม เพื่อช่วยให้ผู้จัดทำมีความเข้าใจในด้านของเนื้อหาที่ได้เรียนมาตลอดภาคการศึกษามากยิ่งขึ้น โดยเกมที่ได้จัดทำนั้นเป็นเกมเกี่ยวกับการเก็บของที่หล่นลงมาจากฟ้า สอดแทรกความสนุกสนาน เพลินเพลิน ผีกสมาธิ และชมความสวยงามของคาแรคเตอร์และฉากในเกม โดยโปรแกรมที่ได้นำมาใช้คือ Intelij ที่ใช้ในการเขียนโค้ด

## ประเภทโครงการ

โปรแกรมเกม

ประโยชน์ของโครงการ

1. เพื่อความสนุกสนาน เพลิดเพลิน ผ่อนคลายในเวลาว่าง
2. ผีกความมีไหวพริบ รวดเร็ว
3. ผีกสมาธิ

## ขอบเขตของโครงการ

1. ตัวละครในเกมเคลื่อนย้ายในแนวแกน X
2. เคลื่อนไหวตัวละครด้วยการใช้ปุ่มกด A และ D
3. เกมมีทั้งหมด 5 level
4. เมื่อหลุดเลือดหรือคะแนนเท่ากับศูนย์ เกมจะสิ้นสุด

เกมสื่ กูกูกูเกบของ

## รายละเอียดเกมส์

เกมส์ ถูกเก็บของ ผีน้อยตัวหนึ่งต้องเก็บของที่ตกลงมาจากฟ้า โดยต้องเก็บของที่ช่วยเพิ่มคะแนนหรือค่าในหลอดเลือดเพื่อให้ผีน้อยไปเกิดในภพภูมิที่ดี ซึ่งจะมี ลูกอม ดาว และเหรียญหัวใจที่ช่วยเพิ่มคะแนนและหัวใจกับยา จะช่วยเพิ่มหลอดเลือด แต่ถ้าผีน้อยเก็บได้ลูกระเบิด กะโหลก และปัสสาวะ จะทำให้หลอดเลือดและคะแนนของผีน้อยลดลง และถ้าหากหลอดเลือดหมดหรือคะแนนเท่ากับ 0 จะถือว่า Game Over ทันที และผีน้อยจะไม่ได้ไปเกิดในภพภูมิที่ดี

### วิธีเล่น

ใช้ 'A' และลูกศร '←' ในแบบพิมพ์ ถ้าต้องการเคลื่อนที่ไปทางซ้าย

ใช้ 'D' และลูกศร '→' ในแบบพิมพ์ ถ้าต้องการเคลื่อนที่ไปทางขวา

ตัวละคร

๗. ฝึ้นอัย



ดาว ลูกอม และเหรียญเพิ่มคะแนน



ตัวลดคะแนนและ

ลดหย่อนเลือด



ยาและหัวใจชุกชีวิต

ผีน้อยต้องเก็บลูกอม ดาวและเหรียญเพื่อเพิ่มคะแนน

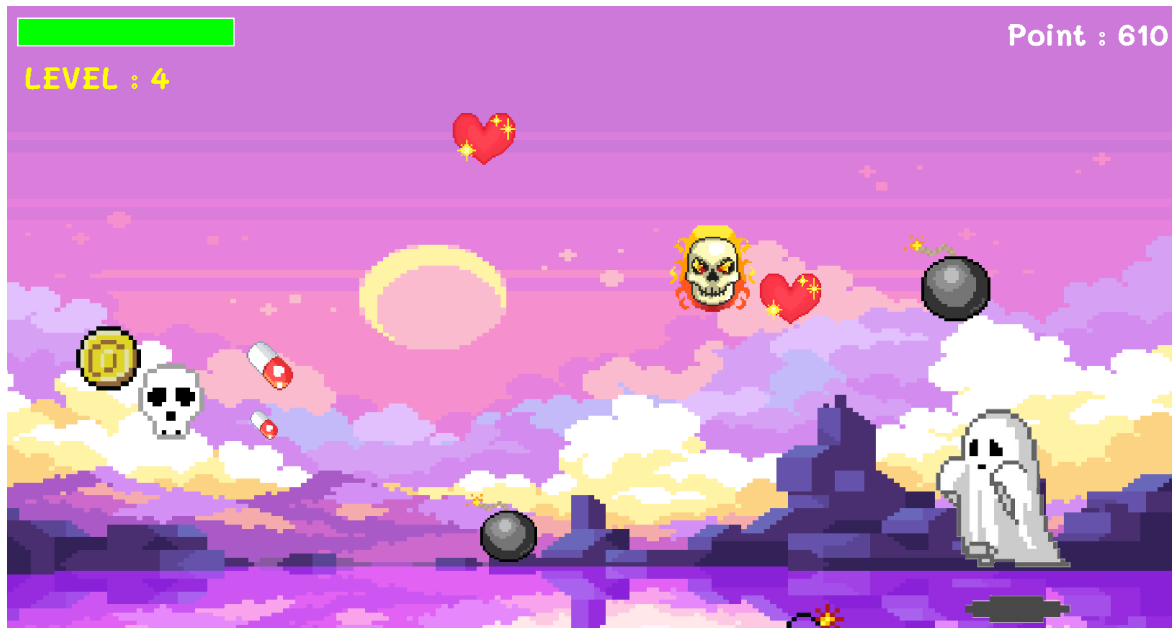


ถ้าโดนระเบิดหลุดเลือดจะลดลงและคะแนนก็จะลดลงด้วย

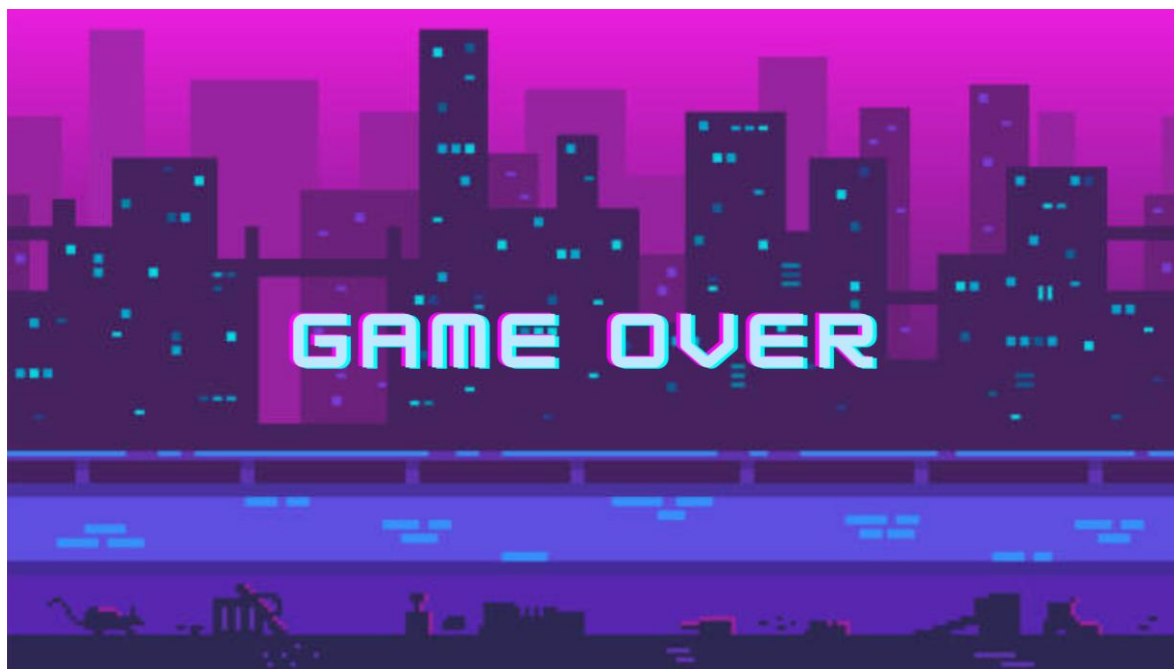
ซึ่งถ้าอยากให้หลุดเลือดเพิ่มต้องเก็บหัวใจไม่กี่ยา



โดยถ้า level สูงขึ้น ระดับความยากก็มากขึ้น



ถ้าหลอดเลือดหมดหรือคะแนนเท่ากับ 0 จะถือว่า Game Over ทันที



## ประโยชน์

1. เพื่อความสนุกสนาน เพลิดเพลิน ผ่อนคลายในเวลารว่าง
2. ฝึกความมีไหวพริบ รวดเร็ว
3. ฝึกสมาธิ

## ตารางการแผนทำงานเดือนตุลาคม – พฤศจิกายน

ลำดับ	รายการ	2-9 ตุลาคม	10-24 ตุลาคม	25-31 พฤศจิกายน	1-3 พฤศจิกายน
1	หารูปตัวละครและทำ กราฟิกต่างๆ				
2	ศึกษาเอกสารและข้อมูล ที่เกี่ยวข้อง				
3	ลงมือเขียนโปรแกรม				
4	จัดทำเอกสาร				
5	ตรวจสอบและแก้ไข ข้อผิดพลาด				
6	ทำการนำเสนอ				

## บทที่ 2 ส่วนการพัฒนา

### เนื้อเรื่องย่อของเกม

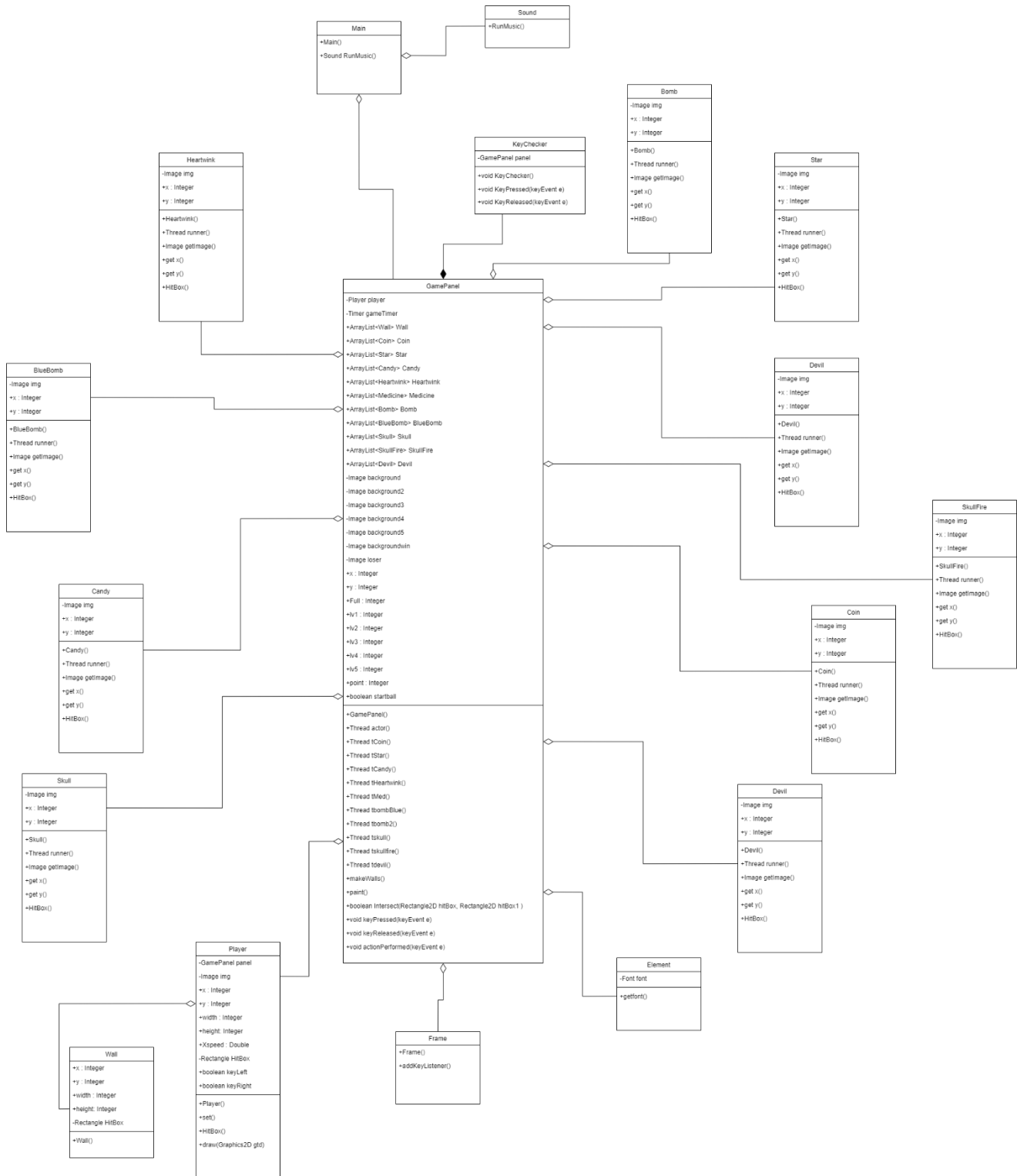
เกมส์ กู้กู่เก็บของ ฝีน้อยตัวหนึ่งต้องเก็บของที่ตกลงมาจากฟ้า โดยต้องเก็บของที่ช่วยเพิ่มค่าคะแนนหรือค่าในหลอดเลือดเพื่อให้ฝีน้อยไปเกิดในภพภูมิที่ดี ซึ่งจะมี ลูกอม ดาว และเหรียญหัวใจที่ช่วยเพิ่มคะแนนและหัวใจกับยา จะช่วยเพิ่มหลอดเลือด แต่ถ้าฝีน้อยเก็บได้ลูกระเบิด กะโหลก และปีศาจ จะทำให้หลอดเลือดและคะแนนของฝีน้อยลดลง และถ้าหากหลอดเลือดหมดหรือคะแนนเท่ากับ 0 จะถือว่า Game Over ทันที และฝีน้อยจะไม่ได้ไปเกิดในภพภูมิที่ดี

### วิธีการเล่น

ใช้ 'A' และลูกศร '←' ในแป้นพิมพ์ ถ้าต้องการเคลื่อนที่ไปทางซ้าย

ใช้ 'D' และลูกศร '→' ในแป้นพิมพ์ ถ้าต้องการเคลื่อนที่ไปทางขวา

# Class Diagram





## อธิบายโปรแกรม

### Public Class Main

- เป็นคลาสที่ใช้ในการรัน โปรแกรม โดยกำหนดขนาดจอที่มีความกว้าง 1100 ซม. และความสูง 620 ซม.  
กำหนดชื่อจอว่า กุ๊กกู่เก็บของ นั่นคือชื่อเกม และ add music ใน class นี้

```
package javagame;
import javax.swing.*;

public class Main {

    public static void main(String[] args) {
        Frame Ghost = new Frame();
        Ghost.setTitle("กุ๊กกู่เก็บของ");
        Ghost.setSize( width: 1100, height: 620);
        Ghost.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        Ghost.setLocationRelativeTo(null);
        Ghost.setResizable(false);
        Ghost.setVisible(true);
        //Add song
        Sound.RunMusic( path: "Music/songgame.wav");
    }
}
```

## Public Class BlueBomb

- Public Class BlueBomb เป็นคลาสระเบิดสีฟ้า มีการกำหนดรูปภาพ กำหนดทิศทางของวัตถุที่จะตกลงมาข้างล่าง และตำแหน่งในการเกิดของวัตถุแบบสุ่ม

```
package javagame;

import java.awt.*;
import java.awt.geom.Rectangle2D;
import java.net.URL;

3 usages
public class BlueBomb {
    3 usages
    Image img;
    public int y= (int)(Math.random()*300)+20;
    public int x=(int)(Math.random()*900)+20;
    1 usage
    BlueBomb(){
        String imageLocation = "img/BlueBomb.png";
        URL imageURL = this.getClass().getResource(imageLocation);
        img = Toolkit.getDefaultToolkit().getImage(imageURL);
        runner.start();
    }
    Thread runner = new Thread(new Runnable() {
        @Override
        public void run() {
            while(true){
                y+=2;
                if(y>=1000){
                    img = null;
                    runner = null;
                    x = -500;
                    y = -500;
                }
                try{
                    runner.sleep( millis: 10);
                }catch(InterruptedException e){}
            }
        }
    });

    public Image getImage() { return img; }

    public int getX() { return x; }
    public int getY() { return y; }
    public Rectangle2D HitBox() { return (new Rectangle2D.Double(x,y, w: 30, h: 30)); }
}
```

## Public Class Bomb

- Public Class Bomb เป็นคลาสระเบิด มีการกำหนดรูปภาพ กำหนดทิศทางของวัตถุที่จะตกลงมาข้างล่าง และตำแหน่งในการเกิดของวัตถุแบบสุ่ม

```
package javagame;

import java.awt.*;
import java.awt.geom.Rectangle2D;
import java.net.URL;

3 usages
public class Bomb {
    3 usages
    Image img;
    public int y= (int)(Math.random()*300)+20;
    public int x=(int)(Math.random()*900)+20;
    1 usage
    Bomb(){
        String imageLocation = "img/Bomb.png";
        URL imageURL = this.getClass().getResource(imageLocation);
        img = Toolkit.getDefaultToolkit().getImage(imageURL);
        runner.start();
    }

    Thread runner = new Thread(new Runnable() {
        @Override
        public void run() {
            while(true){
                y+=2;
                if(y>=1000){
                    img = null;
                    runner = null;
                    x = -500;
                    y = -500;
                }
                try{
                    runner.sleep( millis: 10);
                }catch(InterruptedException e){}
            }
        }
    });

    public Image getImage() { return img; }

    public int getX() { return x; }
    public int getY() { return y; }
    public Rectangle2D HitBox() { return (new Rectangle2D.Double(x,y, w: 30, h: 30)); }
}
```

## Public Class Candy

- Public Class Candy เป็นคลาสแคนดี้ มีการกำหนดรูปภาพ กำหนดทิศทางของวัตถุที่จะตกลงมา ข้างล่าง และตำแหน่งในการเกิดของวัตถุแบบสุ่ม

```
package javagame;
import java.awt.*;
import java.awt.geom.Rectangle2D;
import java.net.URL;
3 usages
public class Candy {
    3 usages
    Image img;
    public int y = (int) (Math.random() * 100) + 20;
    public int x = (int) (Math.random() * 1000) + 20;
    1 usage
    Candy() {
        String imageLocation = "img/candy.png";
        URL imageURL = this.getClass().getResource(imageLocation);
        img = Toolkit.getDefaultToolkit().getImage(imageURL);
        runner.start();
    }
    Thread runner = new Thread(new Runnable() {
        @Override
        public void run() {
            while (true) {
                y += 2;
                if (y >= 500) {
                    img = null;
                    runner = null;
                    x = -500;
                    y = -500;
                }
                try {
                    runner.sleep(10);
                } catch (InterruptedException e) {
                }
            }
        }
    });
    public Image getImage() { return img; }
    public int getX() { return x; }
    public int getY() { return y; }
    public Rectangle2D HitBox() { return (new Rectangle2D.Double(x, y, 50, 50)); }
}
```

## Public Class Coin

- Public Class Coin เป็นคลาสเหรียญ มีการกำหนดรูปภาพ กำหนดทิศทางของวัตถุที่จะตกลงมาข้างล่าง และตำแหน่งในการเกิดของวัตถุแบบสุ่ม

```
package javagame;

import javax.swing.*;
import java.awt.*;
import java.awt.geom.Rectangle2D;
import java.net.URL;

3 usages
public class Coin {
    3 usages
    Image img;
    public int y = (int) (Math.random() * 100) + 20;
    public int x = (int) (Math.random() * 1000) + 20;

    1 usage
    Coin() {
        String imageLocation = "img/coin.png";
        URL imageURL = this.getClass().getResource(imageLocation);
        img = Toolkit.getDefaultToolkit().getImage(imageURL);
        runner.start();
    }

    Thread runner = new Thread(new Runnable() {
        @Override
        public void run() {
            while (true) {
                y += 2;
                if (y >= 1000) {
                    img = null;
                    runner = null;
                    x = -500;
                    y = -500;
                }
                try {
                    runner.sleep(10);
                } catch (InterruptedException e) {}
            }
        }
    });

    public Image getImage() { return img; }
    public int getX() { return x; }
    public int getY() { return y; }
    public Rectangle2D HitBox() { return (new Rectangle2D.Double(x, y, 50, 50)); }
}
```

## Public Class Devil

- Public Class Devil เป็นคลาสปีศาจ มีการกำหนดรูปภาพ กำหนดทิศทางของวัตถุที่จะตกลงมาข้างล่าง และตำแหน่งในการเกิดของวัตถุแบบสุ่ม

```
package javagame;

import java.awt.*;
import java.awt.geom.Rectangle2D;
import java.net.URL;

3 usages
public class Devil {

    3 usages
    Image img;
    public int y = (int) (Math.random() * 100) + 20;
    public int x = (int) (Math.random() * 1000) + 20;

    1 usage
    Devil() {
        String imageLocation = "img/Devil.png";
        URL imageURL = this.getClass().getResource(imageLocation);
        img = Toolkit.getDefaultToolkit().getImage(imageURL);
        runner.start();
    }

    Thread runner = new Thread(new Runnable() {
        @Override
        public void run() {
            while (true) {
                y += 2;
                if (y >= 1000) {
                    img = null;
                    runner = null;
                    x = -500;
                    y = -500;
                }
                try {
                    runner.sleep( millis: 10);
                } catch (InterruptedException e) {
                }
            }
        }
    });

    public Image getImage() { return img; }
    public int getX() { return x; }
    public int getY() { return y; }
    public Rectangle2D HitBox() { return (new Rectangle2D.Double(x, y, w: 50, h: 50)); }
}
```

## Public Class Heartwink

- Public Class Heartwink เป็นคลาสหัวใจ มีการกำหนดรูปภาพ กำหนดทิศทางของวัตถุที่จะตกลงมาข้างล่าง และตำแหน่งในการเกิดของวัตถุแบบสุ่ม

```
package javagame;

import java.awt.*;
import java.awt.geom.Rectangle2D;
import java.net.URL;

3 usages
public class Heartwink {
    3 usages
    Image img;
    public int y = (int) (Math.random() * 100) + 20;
    public int x = (int) (Math.random() * 1000) + 20;

    1 usage
    Heartwink() {
        String imageUrl = "img/hw.png";
        URL imageUrl = this.getClass().getResource(imageLocation);
        img = Toolkit.getDefaultToolkit().getImage(imageURL);
        runner.start();
    }

    Thread runner = new Thread(new Runnable() {
        @Override
        public void run() {
            while (true) {
                y += 2;
                if (y >= 1000) {
                    img = null;
                    runner = null;
                    x = -500;
                    y = -500;
                }
                try {
                    runner.sleep(10);
                } catch (InterruptedException e) {}
            }
        }
    });

    public Image getImage() { return img; }
    public int getX() { return x; }
    public int getY() { return y; }
    public Rectangle2D HitBox() { return (new Rectangle2D.Double(x, y, 50, 50)); }
}
```

## Public Class Medicine

- Public Class Medicine เป็นคลาส มีการกำหนดรูปภาพ กำหนดทิศทางของวัตถุที่จะตกลงมาข้างล่าง และตำแหน่งในการเกิดของวัตถุแบบสุ่ม

```
package javagame;

import java.awt.*;
import java.awt.geom.Rectangle2D;
import java.net.URL;

3 usages
public class Medicine {
    3 usages
    Image img;
    public int y = (int) (Math.random() * 100) + 20;
    public int x = (int) (Math.random() * 1000) + 20;

    1 usage
    Medicine() {
        String imageUrl = "img/medicine.png";
        URL imageUrl = this.getClass().getResource(imageLocation);
        img = Toolkit.getDefaultToolkit().getImage(imageURL);
        runner.start();
    }

    Thread runner = new Thread(new Runnable() {
        @Override
        public void run() {
            while (true) {
                y += 2;
                if (y >= 500) {
                    img = null;
                    runner = null;
                    x = -500;
                    y = -500;
                }
                try {
                    runner.sleep(10);
                } catch (InterruptedException e) {}
            }
        }
    });

    public Image getImage() { return img; }
    public int getX() { return x; }
    public int getY() { return y; }
    public Rectangle2D HitBox() { return (new Rectangle2D.Double(x, y, 50, 50)); }
}
```



## Public Class Skull

- Public Class Skull เป็นคลาสกระดูก มีการกำหนดรูปภาพ กำหนดทิศทางของวัตถุที่จะตกลงมาข้างล่าง และตำแหน่งในการเกิดของวัตถุแบบสุ่ม

```
package javagame;

import java.awt.*;
import java.awt.geom.Rectangle2D;
import java.net.URL;

3 usages
public class Skull {
    3 usages
    Image img;
    public int y = (int) (Math.random() * 100) + 20;
    public int x = (int) (Math.random() * 1000) + 20;

    1 usage
    Skull() {
        String imageLocation = "img/skull.png";
        URL imageURL = this.getClass().getResource(imageLocation);
        img = Toolkit.getDefaultToolkit().getImage(imageURL);
        runner.start();
    }

    Thread runner = new Thread(new Runnable() {
        @Override
        public void run() {
            while (true) {
                y += 2;
                if (y >= 1500) {
                    img = null;
                    runner = null;
                    x = -600;
                    y = -600;
                }
                try {
                    runner.sleep(10);
                } catch (InterruptedException e) {}
            }
        }
    });

    public Image getImage() { return img; }
    public int getX() { return x; }
    public int getY() { return y; }
    public Rectangle2D HitBox() { return (new Rectangle2D.Double(x, y, 50, 50)); }
}
```

## Public Class SkullFire

- Public Class SkullFire เป็นคลาสกระดูกมีไฟ มีการกำหนดรูปภาพ กำหนดทิศทางของวัตถุที่จะตกลงมาข้างล่าง และตำแหน่งในการเกิดของวัตถุแบบสุ่ม

```
package javagame;

import java.awt.*;
import java.awt.geom.Rectangle2D;
import java.net.URL;

3 usages
public class SkullFire {
    3 usages
    Image img;
    public int y = (int) (Math.random() * 100) + 20;
    public int x = (int) (Math.random() * 1000) + 20;

    1 usage
    SkullFire() {
        String imageLocation = "img/skullonfire.png";
        URL imageURL = this.getClass().getResource(imageLocation);
        img = Toolkit.getDefaultToolkit().getImage(imageURL);
        runner.start();
    }

    Thread runner = new Thread(new Runnable() {
        @Override
        public void run() {
            while (true) {
                y += 2;
                if (y >= 1000) {
                    img = null;
                    runner = null;
                    x = -500;
                    y = -500;
                }
                try {
                    runner.sleep(10);
                } catch (InterruptedException e) {}
            }
        }
    });

    public Image getImage() { return img; }
    public int getX() { return x; }
    public int getY() { return y; }
    public Rectangle2D HitBox() { return (new Rectangle2D.Double(x, y, 50, 50)); }
}
```

## Public Class Star

- Public Class Star เป็นคลาสดาว มีการกำหนดรูปภาพ กำหนดทิศทางของวัตถุที่จะตกลงมาข้างล่าง และตำแหน่งในการเกิดของวัตถุแบบสุ่ม

```
package javagame;

import java.awt.*;
import java.awt.geom.Rectangle2D;
import java.net.URL;

3 usages
public class Star {
    3 usages
    Image img;
    public int y = (int) (Math.random() * 100) + 20;
    public int x = (int) (Math.random() * 1000) + 20;

    1 usage
    Star() {
        String imageLocation = "img/medicine.png";
        URL imageURL = this.getClass().getResource(imageLocation);
        img = Toolkit.getDefaultToolkit().getImage(imageURL);
        runner.start();
    }

    Thread runner = new Thread(new Runnable() {
        @Override
        public void run() {
            while (true) {
                y += 2;
                if (y >= 500) {
                    img = null;
                    runner = null;
                    x = -500;
                    y = -500;
                }
                try {
                    runner.sleep(10);
                } catch (InterruptedException e) {
                }
            }
        }
    });

    public Image getImage() { return img; }
    public int getX() { return x; }
    public int getY() { return y; }
    public Rectangle2D HitBox() { return (new Rectangle2D.Double(x, y, 50, 50)); }
}
```

## Public Class Player

```
package javagame;

import javax.swing.*;
import java.awt.*;
import java.awt.geom.Rectangle2D;
import java.net.URL;

2 usages
public class Player extends Rectangle {
    4 usages
    GamePanel panel;
    1 usage
    URL Sauces = this.getClass().getResource( name: "img/Ghost.png");
    1 usage
    Image Image = new ImageIcon(Sauces).getImage();
    6 usages
    int x,y,width,height;
    14 usages
    double Xspeed;
    13 usages
    Rectangle HitBox ;
    9 usages
    boolean keyLeft,keyRight;
```

- Class Player extends Rectangle เป็นแบบ Inheritance โดย Class Player จะสามารถเรียกใช้งาน method ตัวแปรต่างๆ ที่อยู่ใน Class Rectangle ได้ ซึ่ง class Rectangle ก็คือ Superclass เป็น class แม่ในเวลาที่มีการ extends ส่วน Subclass คือ class Player หรือ class ลูกที่ได้ Attribute และ Function จากคลาสแม่

- เป็นคลาสของตัวละครผู้เล่น และมีการกำหนด speed ของตัวละครเวลาที่มีการเคลื่อนที่ซ้าย-ขวา

```

if (Xspeed >7) Xspeed = 7;
if(Xspeed<-7) Xspeed = -7;

if(keyLeft){
    HitBox.x ++;
    for(Wall wall: panel.walls){
        if(wall.HitBox.intersects(HitBox)) Xspeed = -5;
    }
    HitBox.x--;
}
HitBox.x += Xspeed;
for(Wall wall: panel.walls){
    if(HitBox.intersects(wall.HitBox)){
        HitBox.x -=Xspeed;
        while(!wall.HitBox.intersects(HitBox)) HitBox.x += Math.signum(Xspeed);
        HitBox.x -= Math.signum(Xspeed);
        Xspeed = 0;
        x = HitBox.x;
    }
}
x+=Xspeed;
HitBox.x = x;
HitBox.y = y;
}

public Rectangle2D HitBox() { return(new Rectangle2D.Double( x: x+60, y: y+160, w: 75, h: 75)); }
5 usages
public void draw(Graphics2D gtd){

    gtd.drawImage(Image, x: x+40 , y: y+70 , width: 200, height: 210,panel);
}
}

```

## Public Class Sound

- เป็นคลาสสำหรับใส่เสียงในเกม

```
import javax.sound.sampled.*;
import java.io.File;
import java.io.IOException;

1 usage
public class Sound {
    1 usage
    public static void RunMusic(String path){
        try {
            AudioInputStream inputStream = AudioSystem.getAudioInputStream(new File(path));
            Clip clip = AudioSystem.getClip();
            clip.open(inputStream);
            clip.loop( count: 0);
        }catch (UnsupportedAudioFileException e){
            e.printStackTrace();
        }catch (IOException e) {
            e.printStackTrace();
        }catch (LineUnavailableException e){
            e.printStackTrace();
        }
    }
}
```

## Public Class Element

- เป็นคลาสที่กำหนดรูปแบบของ Font ในเกม เช่นคำว่า Point , Level

```
package javagame;

import java.awt.Font;
import java.io.File;

17 usages
public class Element {

    17 usages
    public static Font getFont(int size){
        Font font = null;
        try {
            font = Font.createFont(Font.TRUETYPE_FONT,new File( pathname: "font\\Mali-Bold.ttf"));
            return font.deriveFont((float)size);
        } catch (Exception e) {
            e.printStackTrace();
        }
        return font;
    }
}
```

Public Class KeyChecker extends KeyAdapter

- Class KeyChecker extends KeyAdapter เป็นแบบ Inheritance โดย Class KeyChecker จะสามารถเรียกใช้งาน method ตัวแปรต่างๆ ที่อยู่ใน Class KeyAdapter ได้ ซึ่ง class KeyAdapter ก็คือ Superclass เป็น class แม่ในเวลาที่มีการ extends ส่วน Subclass คือ class KeyChecker หรือ class ลูกที่ได้ Attribute และ Function จากคลาสแม่

```
package javagame;

import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;

1 usage
public class KeyChecker extends KeyAdapter {
    3 usages
    JPanel panel;
    1 usage
    public KeyChecker(JPanel panel){
        this.panel = panel;
    }
    @Override
    public void keyPressed(KeyEvent e){
        panel.keyPressed(e);
    }
    public void keyReleased(KeyEvent e){
        panel.keyReleased(e);
    }
}
```



## Public Class Frame extends JFrame

- Class Frame extends JFrame เป็นแบบ Inheritance โดย Class Frame จะสามารถเรียกใช้งาน method ตัวแปรต่างๆ ที่อยู่ใน Class JFrame ได้ ซึ่ง class JFrame ก็คือ Superclass เป็น class แม่ในเวลาที่มีการ extends ส่วน Subclass คือ class Frame หรือ class ลูกที่ได้ Attribute และ Function จากคลาสแม่
- เป็นคลาสที่กำหนดหน้าจอของ Frame

```
package javagame;

import javax.swing.*;
2 usages
public class Frame extends JFrame {
    1 usage
    public Frame(){
        this.setSize( width: 1100, height: 620);
        GamePanel panel = new GamePanel();
        panel.setLocation( x: 0, y: 0);
        panel.setSize(this.getSize());
        panel.setVisible(true);
        this.add(panel);
        addKeyListener(new KeyChecker(panel));
    }
}
```

## Public Class Wall

- เป็นคลาสที่กำหนดช่องตอนรับของที่ตกลงมาจากฟ้า

```
package javagame;

import java.awt.*;

6 usages
public class Wall {
    1 usage
    int x,y,width,height;
    4 usages
    Rectangle HitBox;

    2 usages
    public Wall(int x ,int y,int width,int height){
        this.x=x;
        this.y=y;
        this.width = width;
        this.height = height;

        HitBox = new Rectangle(x,y,width,height);
    }
}
```

## Public Class GamePanel extends JPanel implements ActionListener

```
public class GamePanel extends JPanel implements ActionListener {  
    49 usages  
    Player player;  
    2 usages  
    Timer gameTimer;  
    4 usages  
    ArrayList<Wall> walls = new ArrayList<Wall>();  
  
    //----- List Add Score -----  
    36 usages  
    ArrayList<Coin> coin1 = new ArrayList<Coin>();  
    15 usages  
    ArrayList<Star> star1 = new ArrayList<Star>();  
    36 usages  
    ArrayList<Candy> cd = new ArrayList<Candy>();  
  
    //----- List Add HP -----  
    36 usages  
    ArrayList<Heartwink> heartwink1 = new ArrayList<Heartwink>();  
    15 usages  
    ArrayList<Medicine> med1 = new ArrayList<Medicine>();  
  
    //----- List Delete Score -----  
    36 usages  
    ArrayList<Bomb> bm2 = new ArrayList<Bomb>();  
    29 usages  
    ArrayList<BlueBomb> bmbblue = new ArrayList<BlueBomb>();  
    22 usages  
    ArrayList<Skull> sk1 = new ArrayList<Skull>();  
    15 usages  
    ArrayList<SkullFire> sk2 = new ArrayList<SkullFire>();  
    8 usages  
    ArrayList<Devil> devil1 = new ArrayList<Devil>();  
}
```

- Class GamePanel extends JPanel implements ActionListener เป็นแบบ Inheritance โดย Class GamePanel จะสามารถเรียกใช้งาน method ตัวแปรต่างๆ ที่อยู่ใน Class JPanel ได้ ซึ่ง class JPanel ก็คือ Superclass เป็น class แม่ในเวลาที่มีการ extends ส่วน Subclass คือ class GamePanel หรือ class ลูกที่ได้ Attribute และ Function จากคลาสแม่ ส่วน ActionListener เป็น Interface ซึ่งจะประกาศเป็น public โดยเข้าถึงได้จากภายนอกคลาส
- มีการสร้าง ArrayList ของที่ตกลงมาจากฟ้า โดยจะมีของที่ช่วยเพิ่มคะแนน เพิ่มค่า HP และของที่จะเป็นตัวลดคะแนน

- กำหนดภาพพื้นหลังของแต่ละด่าน ภาพพื้นหลังตอนชนะเกมและภาพพื้นหลัง Game Over ตอนที่คะแนนเท่ากับ 0 หรือตอนที่หมดเลือดหมด

```
//----- BG -----  
1 usage  
URL bg = this.getClass().getResource( name: "img/bg.jpg");  
1 usage  
URL bg2 = this.getClass().getResource( name: "img/bg2.png");  
1 usage  
URL bg3 = this.getClass().getResource( name: "img/bg3.jpg");  
1 usage  
URL bg4 = this.getClass().getResource( name: "img/bg4.png");  
1 usage  
URL bg5 = this.getClass().getResource( name: "img/bg5.png");  
1 usage  
URL win = this.getClass().getResource( name: "img/win.png");  
1 usage  
URL lose = this.getClass().getResource( name: "img/LOST.png");  
  
Image background2 = new ImageIcon(bg2).getImage();  
1 usage  
Image background3 = new ImageIcon(bg3).getImage();  
1 usage  
Image background4 = new ImageIcon(bg4).getImage();  
1 usage  
Image background5 = new ImageIcon(bg5).getImage();  
1 usage  
Image backgroundwin = new ImageIcon(win).getImage();  
6 usages  
Image loser = new ImageIcon(lose).getImage();
```

```

int x=400;
1 usage
int y=300;
103 usages
public int Full = 200;

//----- Level -----
1 usage
public int lv1 = 1;
1 usage
public int lv2 = 2;
1 usage
public int lv3 = 3;
1 usage
public int lv4 = 4;
1 usage
public int lv5 = 5;
44 usages
public int point = 0;
20 usages
boolean startball = false;

1 usage
Thread actor = new Thread(new Runnable() {
    @Override
    public void run() {
        while (true) {
            try {
                Thread.sleep(1);
            } catch (Exception e) {}
            repaint();
        }
    }
});

```

กำหนดค่าแกน x = 400, y = 300, หลอดเลือด = 200, เลเวล 1-5 ส่วน Thread actor เป็น thread ของตัวละคร

## Thread ( Add Score )

```
//----- add score -----  
1 usage  
Thread tCoin = new Thread(new Runnable() {  
    @Override  
    public void run() {  
        while (true) {  
            try {  
                if (startball == false) {  
                    Thread.sleep( (long) (Math.random() * 100) + 2000);  
                }  
            } catch (InterruptedException e) {  
                e.printStackTrace();  
            }  
            if (startball == false) {  
                coin1.add(new Coin());  
            }  
        }  
    }  
});
```

Thread tCoin เป็น thread ที่ทำให้เหรียญตกลงมา

```
Thread tStar = new Thread(new Runnable() {  
    @Override  
    public void run() {  
        while (true) {  
            try {  
                if (startball == false) {  
                    Thread.sleep( (long) (Math.random() * 500) + 2000);  
                }  
            } catch (InterruptedException e) {  
                e.printStackTrace();  
            }  
            if (startball == false) {  
                star1.add(new Star());  
            }  
        }  
    }  
});
```

Thread tStar เป็น thread ที่ทำให้ดาวตกลงมา

```
Thread tCandy = new Thread(new Runnable() {  
    @Override  
    public void run() {  
        while (true) {  
            try {  
                if (startball == false) {  
                    Thread.sleep( (long) (Math.random() * 500) + 2000);  
                }  
            } catch (InterruptedException e) {  
                e.printStackTrace();  
            }  
            if (startball == false) {  
                cd.add(new Candy());  
            }  
        }  
    }  
});
```

Thread tCandy เป็น thread ที่ทำให้แคนดี้ตกลงมา

## Thread ( Add HP )

```
//----- Add HP -----  
1 usage  
Thread tHeartwink = new Thread(new Runnable() {  
    @Override  
    public void run() {  
        while (true) {  
            try {  
                if (startball == false) {  
                    Thread.sleep( millis: (long) (Math.random() * 10000) + 2000);  
                }  
            } catch (InterruptedException e) {  
                e.printStackTrace();  
            }  
            if (startball == false) {  
                heartwink1.add(new Heartwink());  
            }  
        }  
    }  
});
```

Thread tHeartwink เป็น thread ที่ทำให้หัวใจตกลงมา

```
Thread tMed = new Thread(new Runnable() {  
    @Override  
    public void run() {  
        while (true) {  
            try {  
                if (startball == false) {  
                    Thread.sleep( millis: (long) (Math.random() * 10000) + 2000);  
                }  
            } catch (InterruptedException e) {  
                e.printStackTrace();  
            }  
            if (startball == false) {  
                med1.add(new Medicine());  
            }  
        }  
    }  
});
```

Thread tMed เป็น thread ที่ทำให้ยาตกลงมา

## Thread ( Delete Score )

```
//----- delete score -----  
1 usage  
Thread tbombBlue = new Thread(new Runnable() {  
    @Override  
    public void run() {  
        while (true) {  
            try {  
                if (startball == false) {  
                    Thread.sleep( millis: (long) (Math.random() * 1000) + 2000);  
                }  
            } catch (InterruptedException e) {  
                e.printStackTrace();  
            }  
            if (startball == false) {  
                bmbblue.add(new javagame.BlueBomb());  
            }  
        }  
    }  
});
```

Thread tbombBlue เป็น thread ที่ทำให้ลูกระเบิดสีฟ้าตกลงมา

```
Thread tbomb2 = new Thread(new Runnable() {  
    @Override  
    public void run() {  
        while (true) {  
            try {  
                if (startball == false) {  
                    Thread.sleep( millis: (long) (Math.random() * 1000) + 2000);  
                }  
            } catch (InterruptedException e) {  
                e.printStackTrace();  
            }  
            if (startball == false) {  
                bm2.add(new javagame.Bomb());  
            }  
        }  
    }  
});
```

Thread tbomb2 เป็น thread ที่ทำให้ลูกระเบิดสีดำตกลงมา

```
Thread tdevil = new Thread(new Runnable() {  
    @Override  
    public void run() {  
        while (true) {  
            try {  
                if (startball == false) {  
                    Thread.sleep( millis: (long) (Math.random() * 700) + 2000);  
                }  
            } catch (InterruptedException e) {  
                e.printStackTrace();  
            }  
            if (startball == false) {  
                devil1.add(new javagame.Devil());  
            }  
        }  
    }  
});
```

Thread tdevil เป็น thread ที่ทำให้ปีศาจตกลงมา



```

Thread tskull = new Thread(new Runnable() {
    @Override
    public void run() {
        while (true) {
            try {
                if (startball == false) {
                    Thread.sleep( (long) (Math.random() * 700) + 2000);
                }
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
            if (startball == false) {
                sk1.add(new javagame.Skull());
            }
        }
    }
});

```

Thread tskull เป็น thread ที่ทำให้กระดูกตกลงมา

```

Thread tskullfire = new Thread(new Runnable() {
    @Override
    public void run() {
        while (true) {
            try {
                if (startball == false) {
                    Thread.sleep( (long) (Math.random() * 700) + 2000);
                }
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
            if (startball == false) {
                sk2.add(new javagame.SkullFire());
            }
        }
    }
});

```

Thread tskullfire เป็น thread ที่ทำให้กระดูกมีไฟตกลงมา

## Public GamePanel

- เป็นส่วนที่จะทำให้แต่ละ thread ทำงาน และมีการ paint graphics

```
public GamePanel() {
    player = new Player(x, y, panel: this);
    makeWalls();
    gameTimer = new Timer();
    gameTimer.schedule(new TimerTask() {
        @Override
        public void run() {
            player.set();
            repaint();
        }
    }, delay: 0, period: 10);
    actor.start();
    tCoin.start();
    tHeartwink.start();
    tCandy.start();
    tMed.start();
    tStar.start();
    tbomb2.start();
    tbombBlue.start();
    tskull.start();
    tskullfire.start();
    tdevil.start();
}

public void makeWalls(){
    for(int i = 50; i<650; i+=50){
        walls.add(new Wall( x: 933,i, width: 50, height: 50));
    }
    for(int i = 50; i<650; i+=50){
        walls.add(new Wall( x: -150,i, width: 50, height: 50));
    }
}

public void paint(Graphics g) {
    super.paint(g);
}
```

## LEVEL 1

```
// ----- Level1 -----

if((point<200 && point>=0)&& (Full>0)){
    Graphics2D gtd = (Graphics2D) g;
    g.drawImage(background, x: 0, y: 0, this.getWidth(), this.getHeight(), observer: this);
    player.draw(gtd);

    for (int i = 0; i < cd.size(); i++) {
        g.drawImage(cd.get(i).getImage(), cd.get(i).getX(), cd.get(i).getY(),
            width: 60, height: 60, observer: this);
    }
    for (int i = 0; i < cd.size(); i++) {
        if (Intersect(player.HitBox(), cd.get(i).HitBox())) {
            cd.remove(i);
            point += 10;
            Full = Full;
        }
    }
}

for (int i = 0; i < coin1.size(); i++) {
    g.drawImage(coin1.get(i).getImage(), coin1.get(i).getX(), coin1.get(i).getY(),
        width: 60, height: 60, observer: this);
}

for (int i = 0; i < coin1.size(); i++) {
    if (Intersect(player.HitBox(), coin1.get(i).HitBox())) {
        coin1.remove(i);
        point += 20;
        Full = Full;
    }
}

for (int i = 0; i < bm2.size(); i++) {
    g.drawImage(bm2.get(i).getImage(), bm2.get(i).getX(), bm2.get(i).getY(),
        width: 80, height: 80, observer: this);
}

for (int i = 0; i < bm2.size(); i++) {
    if (Intersect(player.HitBox(), bm2.get(i).HitBox())) {
        bm2.remove(i);
        point -= 10;
        Full = Full-10;
    }
}

g.setColor(Color.GRAY);
g.fillRect(x: 15, y: 15, width: 200, height: 25);

g.setColor(Color.GREEN);
g.fillRect(x: 15, y: 15, Full, height: 25);

g.setColor(Color.WHITE);
g.drawRect(x: 15, y: 15, width: 200, height: 25);

g.setFont(Element.getFont(size: 20));
g.setColor(Color.white);
g.drawString(str: "Point : "+point, x: 930, y: 40);

g.setFont(Element.getFont(size: 20));
g.setColor(Color.yellow);
g.drawString(str: "LEVEL : " +lv1, x: 20, y: 80);

if(Full<=0 || point<0){
    g.drawImage(loser, x: 0, y: 0, this.getWidth(), this.getHeight(), observer: this);
}
}
```

ถ้าคะแนนมากกว่าเท่ากับ 0 และคะแนนน้อยกว่า 200 จะอยู่ที่ level 1 โดยจะประกอบไปด้วยเคนดี้ เฮอร์มู และลูกระเบิดสีดำที่ตกลงมา ถ้าเหลือคะแนนน้อยกว่าเท่ากับ 0 หรือคะแนน = 0 จะ Game Over

## LEVEL 2

```
/// ----- Level2 -----

else if( (point<400 && point>=200)){
    Graphics2D gtd = (Graphics2D) g;
    g.drawImage(background2, x 0, y 0, this.getWidth(), this.getHeight(), observer, this);
    player.draw(gtd);

    for (int i = 0; i < cd.size(); i++) {
        g.drawImage(cd.get(i).getImage(), cd.get(i).getX(), cd.get(i).getY(),
            width: 60, height: 60, observer, this);
    }

    for (int i = 0; i < cd.size(); i++) {
        if (Intersect(player.HitBox(), cd.get(i).HitBox())) {
            cd.remove(i);
            point += 10;
            Full = Full;
        }
    }
}

for (int i = 0; i < coin1.size(); i++) {
    g.drawImage(coin1.get(i).getImage(), coin1.get(i).getX(), coin1.get(i).getY(),
        width: 60, height: 60, observer, this);
}

for (int i = 0; i < coin1.size(); i++) {
    if (Intersect(player.HitBox(), coin1.get(i).HitBox())) {
        coin1.remove(i);
        point += 20;
        Full = Full;
    }
}

for (int i = 0; i < heartwink1.size(); i++) {
    g.drawImage(heartwink1.get(i).getImage(), heartwink1.get(i).getX(), heartwink1.get(i).getY(),
        width: 60, height: 60, observer, this);
}

for (int i = 0; i < heartwink1.size(); i++) {
    if (Intersect(player.HitBox(), heartwink1.get(i).HitBox())) {
        heartwink1.remove(i);
        Full = Full;
        if(Full == 200){
            Full = Full;
        }
        else{
            Full = Full+10;
        }
    }
}

for (int i = 0; i < bm2.size(); i++) {
    g.drawImage(bm2.get(i).getImage(), bm2.get(i).getX(), bm2.get(i).getY(),
        width: 80, height: 80, observer, this);
}

for (int i = 0; i < bm2.size(); i++) {
    if (Intersect(player.HitBox(), bm2.get(i).HitBox())) {
        bm2.remove(i);
        point -= 10;
        Full = Full-10;
    }
}

for (int i = 0; i < bmbLue.size(); i++) {
    g.drawImage(bmbLue.get(i).getImage(), bmbLue.get(i).getX(), bmbLue.get(i).getY(),
        width: 120, height: 100, observer, this);
}

for (int i = 0; i < bmbLue.size(); i++) {
    if (Intersect(player.HitBox(), bmbLue.get(i).HitBox())) {
        bmbLue.remove(i);
        point -= 20;
        Full = Full-20;
    }
}
}
```

ถ้าคะแนนมากกว่าเท่ากับ 200 และคะแนนน้อยกว่า 400 จะอยู่ที่ level 2 โดยจะประกอบไปด้วย  
แคนดี้ เหริยญ หัวใจ ลูกกระเบิดสีดำและลูกกระเบิดสีฟ้าที่ตกลงมา

```

g.setColor(Color.GRAY);
g.fillRect( x: 15, y: 15, width: 200, height: 25);

g.setColor(Color.GREEN);
g.fillRect( x: 15, y: 15, Full, height: 25);

g.setColor(Color.WHITE);
g.drawRect( x: 15, y: 15, width: 200, height: 25);

g.setFont(Element.getFont( size: 28));
g.setColor(Color.white);
g.drawString( str: "Point : "+point, x: 930 , y: 40);

g.setFont(Element.getFont( size: 28));
g.setColor(Color.yellow);
g.drawString( str: "LEVEL : " +Lv1, x: 20, y: 80);

if(Full<=0 || point<0){
    g.drawImage(loser, x: 0, y: 0, this.getWidth(), this.getHeight(), observer: this);
}
}

```

และถ้าหลอดเลือดน้อยกว่าเท่ากับ 0 หรือคะแนน = 0 จะ Game Over

## LEVEL 3

```

/// ----- Level3 -----

else if (point<600 && point>=400){

    Graphics2D gtd = (Graphics2D) g;
    g.drawImage(background3, x: 0, y: 0, this.getWidth(), this.getHeight(), observer: this);
    player.draw(gtd);

    for (int i = 0; i < cd.size(); i++) {
        g.drawImage(cd.get(i).getImage(), cd.get(i).getX(), cd.get(i).getY(),
            width: 60, height: 60, observer: this);
    }

    for (int i = 0; i < cd.size(); i++) {
        if (Intersect(player.HitBox(), cd.get(i).HitBox())) {
            cd.remove(i);
            point += 10;
            Full = Full;
        }
    }
}

for (int i = 0; i < coin1.size(); i++) {
    g.drawImage(coin1.get(i).getImage(), coin1.get(i).getX(), coin1.get(i).getY(),
        width: 60, height: 60, observer: this);
}

for (int i = 0; i < coin1.size(); i++) {
    if (Intersect(player.HitBox(), coin1.get(i).HitBox())) {
        coin1.remove(i);
        point += 20;
        Full = Full;
    }
}

for (int i = 0; i < heartwink1.size(); i++) {
    g.drawImage(heartwink1.get(i).getImage(), heartwink1.get(i).getX(), heartwink1.get(i).getY(),
        width: 60, height: 60, observer: this);
}

for (int i = 0; i < heartwink1.size(); i++) {
    if (Intersect(player.HitBox(), heartwink1.get(i).HitBox())) {
        heartwink1.remove(i);
        Full = Full;
        if(Full == 200){
            Full = Full;
        }
        else{
            Full = Full+10;
        }
    }
}
}

```

```

for (int i = 0; i < bm2.size(); i++) {
    g.drawImage(bm2.get(i).getImage(), bm2.get(i).getX(), bm2.get(i).getY(),
        width: 80, height: 80, observer: this);
}
for (int i = 0; i < bm2.size(); i++) {
    if (Intersect(player.HitBox(), bm2.get(i).HitBox())) {
        bm2.remove(i);
        point -= 10;
        Full = Full-10;
    }
}

for (int i = 0; i < bmbLue.size(); i++) {
    g.drawImage(bmbLue.get(i).getImage(), bmbLue.get(i).getX(), bmbLue.get(i).getY(),
        width: 120, height: 100, observer: this);
}
for (int i = 0; i < bmbLue.size(); i++) {
    if (Intersect(player.HitBox(), bmbLue.get(i).HitBox())) {
        bmbLue.remove(i);
        point -= 20;
        Full = Full-20;
    }
}

for (int i = 0; i < sk1.size(); i++) {
    g.drawImage(sk1.get(i).getImage(), sk1.get(i).getX(), sk1.get(i).getY(),
        width: 70, height: 70, observer: this);
}
for (int i = 0; i < sk1.size(); i++) {
    if (Intersect(player.HitBox(), sk1.get(i).HitBox())) {
        sk1.remove(i);
        point -= 30;
        Full = Full-30;
    }
}

g.setColor(Color.GRAY);
g.fillRect( x: 15, y: 15, width: 200, height: 25);

g.setColor(Color.GREEN);
g.fillRect( x: 15, y: 15, Full, height: 25);

g.setColor(Color.WHITE);
g.drawRect( x: 15, y: 15, width: 200, height: 25);

g.setFont(Element.getFont( size: 28));
g.setColor(Color.white);
g.drawString( str: "Point : "+point, x: 930 , y: 40);

g.setFont(Element.getFont( size: 28));
g.setColor(Color.yellow);
g.drawString( str: "LEVEL : " +lv1, x: 20, y: 80);

if(Full<=0 || point<0){
    g.drawImage(loser, x: 0, y: 0, this.getWidth(), this.getHeight(), observer: this);
}
}
}

```

ถ้าคะแนนมากกว่าเท่ากับ 400 และคะแนนน้อยกว่า 600 จะอยู่ที่ level 3 โดยจะประกอบไปด้วย  
 แคนดี้ เหริยญ หัวใจ ลูกระเบิดสีดำ ลูกระเบิดสีฟ้า และกระดุกที่ตกลงมา และถ้าหมดเลื้อคนน้อย  
 กว่าเท่ากับ 0 หรือคะแนน = 0 จะ Game Over

## LEVEL 4

```
/// ----- Level4 -----

else if (point<800 && point>=600){
    Graphics2D gtd = (Graphics2D) g;
    g.drawImage(background4, 0, 0, this.getWidth(), this.getHeight(), observer, this);
    player.draw(gtd);

    for (int i = 0; i < cd.size(); i++) {
        g.drawImage(cd.get(i).getImage(), cd.get(i).getX(), cd.get(i).getY(),
            width: 60, height: 60, observer: this);
    }

    for (int i = 0; i < cd.size(); i++) {
        if (Intersect(player.HitBox(), cd.get(i).HitBox())) {
            cd.remove(i);
            point += 10;
            Full = Full;
        }
    }
}

for (int i = 0; i < coin1.size(); i++) {
    g.drawImage(coin1.get(i).getImage(), coin1.get(i).getX(), coin1.get(i).getY(),
        width: 60, height: 60, observer: this);
}

for (int i = 0; i < coin1.size(); i++) {
    if (Intersect(player.HitBox(), coin1.get(i).HitBox())) {
        coin1.remove(i);
        point += 20;
        Full = Full;
    }
}

for (int i = 0; i < star1.size(); i++) {
    g.drawImage(star1.get(i).getImage(), star1.get(i).getX(), star1.get(i).getY(),
        width: 60, height: 60, observer: this);
}

for (int i = 0; i < star1.size(); i++) {
    if (Intersect(player.HitBox(), star1.get(i).HitBox())) {
        star1.remove(i);
        point += 40;
        Full = Full;
    }
}

for (int i = 0; i < heartwink1.size(); i++) {
    g.drawImage(heartwink1.get(i).getImage(), heartwink1.get(i).getX(), heartwink1.get(i).getY(),
        width: 60, height: 60, observer: this);
}

for (int i = 0; i < heartwink1.size(); i++) {
    if (Intersect(player.HitBox(), heartwink1.get(i).HitBox())) {
        heartwink1.remove(i);
        Full = Full;
        if(Full == 200){
            Full = Full;
        }
        else{
            Full = Full+10;
        }
    }
}

for (int i = 0; i < med1.size(); i++) {
    g.drawImage(med1.get(i).getImage(), med1.get(i).getX(), med1.get(i).getY(),
        width: 100, height: 100, observer: this);
}

for (int i = 0; i < med1.size(); i++) {
    if (Intersect(player.HitBox(), med1.get(i).HitBox())) {
        med1.remove(i);
        Full = Full;
        if(Full == 200){
            Full = Full;
        }
        else{
            Full = Full+30;
        }
    }
}
}
```

```

for (int i = 0; i < bm2.size(); i++) {
    g.drawImage(bm2.get(i).getImage(), bm2.get(i).getX(), bm2.get(i).getY(),
        width: 80, height: 80, observer: this);
}

for (int i = 0; i < bm2.size(); i++) {
    if (Intersect(player.HitBox(), bm2.get(i).HitBox())) {
        bm2.remove(i);
        point -= 10;
        Full = Full-10;
    }
}

for (int i = 0; i < bmbblue.size(); i++) {
    g.drawImage(bmbblue.get(i).getImage(), bmbblue.get(i).getX(), bmbblue.get(i).getY(),
        width: 120, height: 100, observer: this);
}

for (int i = 0; i < bmbblue.size(); i++) {
    if (Intersect(player.HitBox(), bmbblue.get(i).HitBox())) {
        bmbblue.remove(i);
        point -= 20;
        Full = Full-20;
    }
}

for (int i = 0; i < sk1.size(); i++) {
    g.drawImage(sk1.get(i).getImage(), sk1.get(i).getX(), sk1.get(i).getY(),
        width: 70, height: 70, observer: this);
}

for (int i = 0; i < sk1.size(); i++) {
    if (Intersect(player.HitBox(), sk1.get(i).HitBox())) {
        sk1.remove(i);
        point -= 30;
        Full = Full-30;
    }
}

for (int i = 0; i < sk2.size(); i++) {
    g.drawImage(sk2.get(i).getImage(), sk2.get(i).getX(), sk2.get(i).getY(),
        width: 80, height: 80, observer: this);
}

for (int i = 0; i < sk2.size(); i++) {
    if (Intersect(player.HitBox(), sk2.get(i).HitBox())) {
        sk2.remove(i);
        point -= 50;
        Full = Full-50;
    }
}

g.setColor(Color.GRAY);
g.fillRect(x: 15, y: 15, width: 200, height: 25);

g.setColor(Color.GREEN);
g.fillRect(x: 15, y: 15, Full, height: 25);

g.setColor(Color.WHITE);
g.drawRect(x: 15, y: 15, width: 200, height: 25);

g.setFont(Element.getFont(size: 28));
g.setColor(Color.white);
g.drawString(txt: "Point : "+point, x: 930, y: 40);

g.setFont(Element.getFont(size: 28));
g.setColor(Color.yellow);
g.drawString(txt: "LEVEL : " +lv1, x: 20, y: 80);

if(Full<=0 || point<0){
    g.drawImage(loser, x: 0, y: 0, this.getWidth(), this.getHeight(), observer: this);
}
}
}

```

ถ้าคะแนนมากกว่าเท่ากับ 600 และคะแนนน้อยกว่า 800 จะอยู่ที่ level 4 โดยจะประกอบไปด้วย  
 แคนดี้ เหริยญ ดาว หัวใจ ยา ลูกระเบิดสีดำ ลูกระเบิดสีฟ้า กระดุก และกระดุกที่มีไฟที่ตกลงมา และ  
 ถ้าหลุดเล็ดน้อยกว่าเท่ากับ 0 หรือคะแนน = 0 จะ Game Over



## Level 5

```
/// ----- Level5 -----

else if (point<1000 && point>=800) {
    Graphics2D gtd = (Graphics2D) g;
    g.drawImage(background5, x: 0, y: 0, this.getWidth(), this.getHeight(), observer: this);
    player.draw(gtd);

    for (int i = 0; i < cd.size(); i++) {
        g.drawImage(cd.get(i).getImage(), cd.get(i).getX(), cd.get(i).getY(),
            width: 60, height: 60, observer: this);
    }

    for (int i = 0; i < cd.size(); i++) {
        if (Intersect(player.HitBox(), cd.get(i).HitBox())) {
            cd.remove(i);
            point += 10;
            Full = Full;
        }
    }
}

for (int i = 0; i < coin1.size(); i++) {
    g.drawImage(coin1.get(i).getImage(), coin1.get(i).getX(), coin1.get(i).getY(),
        width: 60, height: 60, observer: this);
}

for (int i = 0; i < coin1.size(); i++) {
    if (Intersect(player.HitBox(), coin1.get(i).HitBox())) {
        coin1.remove(i);
        point += 20;
        Full = Full;
    }
}

for (int i = 0; i < star1.size(); i++) {
    g.drawImage(star1.get(i).getImage(), star1.get(i).getX(), star1.get(i).getY(),
        width: 60, height: 60, observer: this);
}

for (int i = 0; i < star1.size(); i++) {
    if (Intersect(player.HitBox(), star1.get(i).HitBox())) {
        star1.remove(i);
        point += 40;
        Full = Full;
    }
}

for (int i = 0; i < heartwink1.size(); i++) {
    g.drawImage(heartwink1.get(i).getImage(), heartwink1.get(i).getX(), heartwink1.get(i).getY(),
        width: 60, height: 60, observer: this);
}

for (int i = 0; i < heartwink1.size(); i++) {
    if (Intersect(player.HitBox(), heartwink1.get(i).HitBox())) {
        heartwink1.remove(i);
        Full = Full;
        if(Full == 200){
            Full = Full;
        }
        else{
            Full = Full+10;
        }
    }
}

for (int i = 0; i < med1.size(); i++) {
    g.drawImage(med1.get(i).getImage(), med1.get(i).getX(), med1.get(i).getY(),
        width: 100, height: 100, observer: this);
}

for (int i = 0; i < med1.size(); i++) {
    if (Intersect(player.HitBox(), med1.get(i).HitBox())) {
        med1.remove(i);
        Full = Full;
        if(Full == 200){
            Full = Full;
        }
        else{
            Full = Full+30;
        }
    }
}
}
```

```

for (int i = 0; i < bm2.size(); i++) {
    g.drawImage(bm2.get(i).getImage(), bm2.get(i).getX(), bm2.get(i).getY(),
        width: 80, height: 80, observer: this);
}

for (int i = 0; i < bm2.size(); i++) {
    if (Intersect(player.HitBox(), bm2.get(i).HitBox())) {
        bm2.remove(i);
        point -= 10;
        Full = Full-10;
    }
}

for (int i = 0; i < bmbblue.size(); i++) {
    g.drawImage(bmbblue.get(i).getImage(), bmbblue.get(i).getX(), bmbblue.get(i).getY(),
        width: 120, height: 100, observer: this);
}

for (int i = 0; i < bmbblue.size(); i++) {
    if (Intersect(player.HitBox(), bmbblue.get(i).HitBox())) {
        bmbblue.remove(i);
        point -= 20;
        Full = Full-20;
    }
}

for (int i = 0; i < sk1.size(); i++) {
    g.drawImage(sk1.get(i).getImage(), sk1.get(i).getX(), sk1.get(i).getY(),
        width: 70, height: 70, observer: this);
}

for (int i = 0; i < sk1.size(); i++) {
    if (Intersect(player.HitBox(), sk1.get(i).HitBox())) {
        sk1.remove(i);
        point -= 30;
        Full = Full-30;
    }
}

for (int i = 0; i < sk2.size(); i++) {
    g.drawImage(sk2.get(i).getImage(), sk2.get(i).getX(), sk2.get(i).getY(),
        width: 80, height: 80, observer: this);
}

for (int i = 0; i < sk2.size(); i++) {
    if (Intersect(player.HitBox(), sk2.get(i).HitBox())) {
        sk2.remove(i);
        point -= 50;
        Full = Full-50;
    }
}

for (int i = 0; i < devill1.size(); i++) {
    g.drawImage(devill1.get(i).getImage(), devill1.get(i).getX(), devill1.get(i).getY(),
        width: 70, height: 70, observer: this);
}

for (int i = 0; i < devill1.size(); i++) {
    if (Intersect(player.HitBox(), devill1.get(i).HitBox())) {
        devill1.remove(i);
        point -= 70;
        Full = Full-70;
    }
}

g.setColor(Color.GRAY);
g.fillRect( x: 15, y: 15, width: 200, height: 25);

g.setColor(Color.GREEN);
g.fillRect( x: 15, y: 15, Full, height: 25);

g.setColor(Color.WHITE);
g.drawRect( x: 15, y: 15, width: 200, height: 25);

g.setFont(Element.getFont( size: 28));
g.setColor(Color.white);
g.drawString( str: "Point : "+point, x: 930, y: 40);

g.setFont(Element.getFont( size: 28));
g.setColor(Color.yellow);
g.drawString( str: "LEVEL : " +lv1, x: 20, y: 80);

if(Full<=0 || point<=0){
    g.drawImage(loser, x: 0, y: 0, this.getWidth(), this.getHeight(), observer: this);
}
}

```

ถ้าคะแนนมากกว่าเท่ากับ 800 และคะแนนน้อยกว่า 1000 จะอยู่ที่ level 5 โดยจะประกอบไปด้วยแคนดี้ เหริญดา ดาว หัวใจ ยา ลูกกระเบิดสีดำ ลูกกระเบิดสีฟ้า กระดุก กระดุกที่มีไฟ และปีศาจที่ตกลงมา และถ้าหลอดเลือดน้อยกว่าเท่ากับ 0 หรือคะแนน = 0 จะ Game Over

```

else if(point>1000){

    g.drawImage(backgroundwin, x: 0, y: 0, this.getWidth(), this.getHeight(), observer: this);
    g.setFont(Element.getFont( size: 28));
    g.setColor(Color.yellow);
}

else{
    g.drawImage(loser, x: 0, y: 0, this.getWidth(), this.getHeight(), observer: this);
}
}

```

กรณีที่คะแนนมากกว่า 1000 จะถือว่าชนะ แต่ถ้าคะแนนน้อยกว่า 1000 หรือไม่เข้าเงื่อนไขใดๆ ทั้งหมดที่กล่าวไว้ข้างต้น จะถือว่า Game Over

```

private boolean Intersect(Rectangle2D hitBox, Rectangle2D hitBox1) {

    return (hitBox.intersects(hitBox1));
}

1 usage
void keyPressed(KeyEvent e) {
    if(e.getKeyCode() == 'A') player.keyLeft = true;
    if(e.getKeyCode() == 'D') player.keyRight = true;
    if (e.getKeyCode()==VK_LEFT) player.keyLeft = true;
    if (e.getKeyCode()==VK_RIGHT) player.keyRight = true;
}

1 usage
void keyReleased(KeyEvent e) {
    if(e.getKeyCode() == 'A') player.keyLeft = false;
    if(e.getKeyCode() == 'D') player.keyRight = false;
    if (e.getKeyCode()==VK_LEFT) player.keyLeft = false;
    if (e.getKeyCode()==VK_RIGHT) player.keyRight = false;
}

@Override
public void actionPerformed(ActionEvent ae) {

}
}

```

การทำให้ตัวละครเคลื่อนที่ซ้าย-ขวา โดย 'A' และลูกศร '←' ในแป้นพิมพ์ สำหรับการเคลื่อนที่ไปทางซ้าย และ 'D' และลูกศร '→' ในแป้นพิมพ์ สำหรับการเคลื่อนที่ไปทางขวา

## บทที่ 3 สรุป

### ปัญหาที่พบระหว่างการพัฒนา

- มีกระตุกบ้างเป็นบางครั้งเมื่อมีการเปลี่ยนฉาก
- เนื่องจากการใช้เงื่อนไข if-else ในการเปลี่ยนฉาก ทำให้บางครั้งคะแนนที่มีค่าน้อยกว่าเงื่อนไขที่กำหนดเปลี่ยนไปจากก่อนหน้า ถ้าคะแนนเพิ่มก็เปลี่ยนไปจากปัจจุบัน สลับกันไปมา

### จุดเด่นของโปรแกรมที่ไม่เหมือนใคร

- มีหลาย level
- เป็นเกมที่สร้างเพื่อความบันเทิง
- ฝึกสมาธิ

### คำแนะนำสำหรับผู้สอน

- อาจารย์ใจดีค่ะ มีโค้ดที่ทำในคาบเรียนให้ สอนละเอียด แต่อยากให้เว้นช่วงให้จดคิดนึ่งค่ะ บางทีจดตามที่อาจารย์เขียนหรืออธิบายไม่ทัน