



โครงการ

ToGoHome

จัดทำโดย

6504062663061 จิรัชฎาญจน์ ชูจันทร์

เสนอ

ผู้ช่วยศาสตราจารย์สถิต ประสมพันธ์

วิชา **Object Oriented Programming**

ภาคเรียนที่ 1/2566

ภาควิชาวิทยาการคอมพิวเตอร์และสารสนเทศ คณะวิทยาศาสตร์ประยุกต์

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

## เกี่ยวกับโครงงาน

ชื่อโปรเจค: ToGoHome

นำเสนอ: นาย จิรัชฎาญจน์ ชูจันทร์

อาจารย์ผู้สอน: ผู้ช่วยศาสตราจารย์สถิต ประสมพันธ์

### บทที่ 1: บทนำ

ที่มาและความสำคัญของโปรเจ็ค ประเภทของโครงการ ประโยชน์

### บทที่ 2: ส่วนการพัฒนา

เนื้อเรื่องย่อหรือวิธีการเล่น (กรณีเกมส์)

Class Diagram

รูปแบบการพัฒนา Application / Applet

อธิบายส่วนของโปรแกรมที่มี

Constructor

Encapsulation

Composition

Polymorphism

Abstract

Inheritance

หน้าจอ GUI

### บทที่ 3: สรุป

ปัญหาที่พบระหว่างการพัฒนา จุดเด่นของโปรแกรมที่ไม่เหมือน

## บทที่ 1 ที่มาและความสำคัญของโครงการ

โครงการนี้จัดขึ้นเพื่อวัดผลความสามารถในการเรียนวิชา Object Oriented Programming โดยการนำเรื่องที่เรียนมาสร้างเป็นชิ้นงานในรูปแบบเกมส์

### ประเภทโครงการ

โปรแกรมเกมส์แอปพลิเคชัน

### ประโยชน์

- 1.ทำให้เกิดความสนุก
- 2.ฝึกไหวพริบ
- 3.ฝึกความคิดสร้างสรรค์

### ขอบเขตของโครงการ

- 1.เคลื่อนไหวตัวละครโดยการใช้ปุ่ม A เพื่อเดินถอยหลัง,  
D เพื่อเดินหน้า, Space Bar และ W เพื่อกระโดด
- 2.เกมมี 3 level
- 3.เมื่อชน Monster ก็จะเสียเลือด

4.เมื่อเสียเลือดครบ 3 ก็จะแพ้เกมส์

5.ตัวละครเคลื่อนย้ายในแนวแกน X และ Y

## บทที่ 2 ส่วนการพัฒนา

### เนื้อเรื่องย่อ

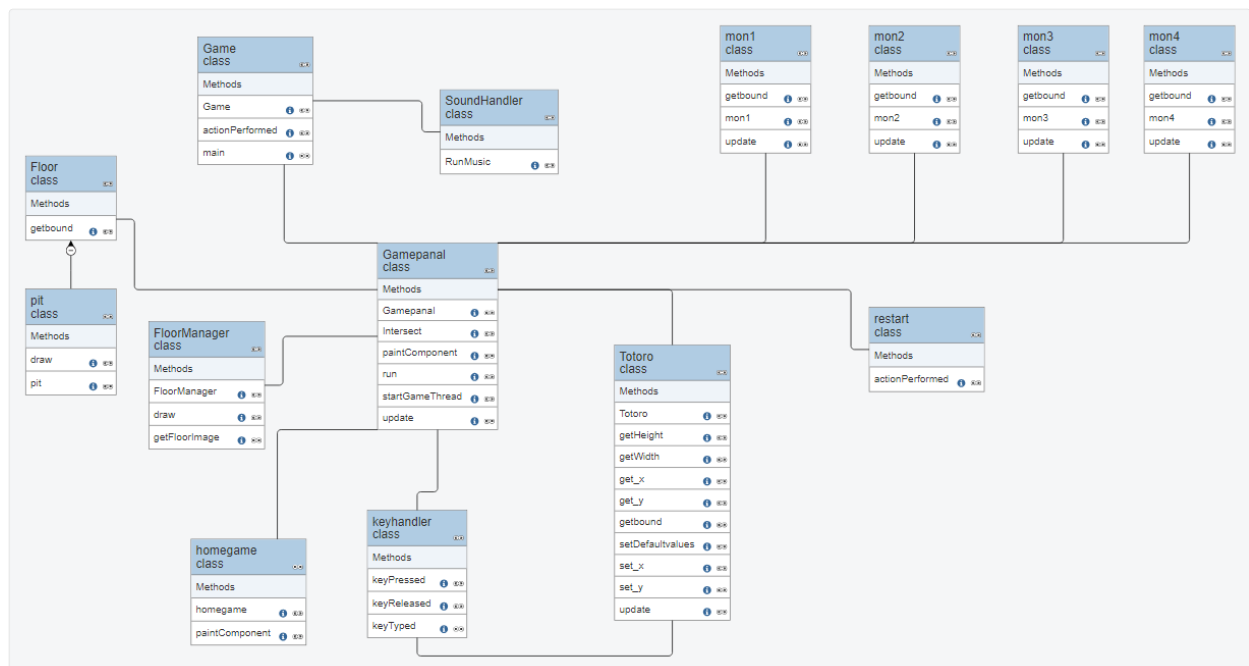
#### เกม ToGoHome

โทโทโร่ผู้เป็นภูติป่าต้องออกไปหาอาหารในตอนค่ำแต่กว่าจะกลับก็เป็นกลางคืนไปซะแล้ว จึงต้องหาทางเอาตัวรอดเพื่อกลับมายังบ้านให้ได้อย่างปลอดภัย

### วิธีเล่น

ใช้ปุ่ม A และ D ในการเคลื่อนที่ไปข้างหน้าและข้างหลัง ใช้ W และ Space Bar ในการกระโดด ตัวละครจะมี HP = 3 และเมื่อเดินชนมอนสเตอร์ก็จะเสีย HP ที่ละ 1 แต่ถ้าตกหลุมก็จะจบเกมทันที

# Class Diagram



ส่วนของโปรแกรม

## Construtor

```
public Totoro(Gamepanel gp, keyhandler keyH) {  
  
    this.gp = gp;  
    this.keyH = keyH;  
  
    setDefaultvalues();  
  
    for(int i=0;i<im.length;i++){  
        im[i] = new ImageIcon("t"+(i+1)+".png");  
    }  
  
    for(int i=0;i<imb.length;i++){  
        imb[i] = new ImageIcon("t"+"b"+(i+1)+".png");  
    }  
  
}
```

มี constructor ที่มีการรับพารามิเตอร์ชนิด Gamepanel ,keyhandler

```

public Game() {

    this.add(comp: home);

    home.Bstart.addActionListener(l: this);
    home.Bexit.addActionListener(l: this);
    gamepanel.Brestart.addActionListener(l: this);
    gamepanel.startGameThread();

    //play music//
    try {
        SoundHandler.RunMusic();
    } catch (LineUnavailableException ex) {
        Logger.getLogger(name: Game.class.getName()).log(level: Level.SEVERE, msg: null, thrown: ex);
    } catch (UnsupportedAudioFileException ex) {
        Logger.getLogger(name: Game.class.getName()).log(level: Level.SEVERE, msg: null, thrown: ex);
    } catch (IOException ex) {
        Logger.getLogger(name: Game.class.getName()).log(level: Level.SEVERE, msg: null, thrown: ex);
    }
}

```

มี constructor ที่รับไม่มีการรับพารามิเตอร์

```

homegame() {
    setLayout (mgr: null);
    Bstart.setBounds (x: 600, y: 410, width: 224, height: 120);
    Bexit.setBounds (x: 600, y: 550, width: 224, height: 120);
    add (comp: Bstart);
    add (comp: Bexit);

}

```

มี constructor ที่รับไม่มีการรับพารามิเตอร์

```

public mon1(Gamepanel gp,int x,int y,int w,int h) {

    this.gp = gp;
    width = w;
    height = h;
    this.x = x;
    this.y = y;
    for(int i=0;i<im.length;i++){
        im[i] = new ImageIcon("mon"+(i+1)+".png");
    }

}

public mon2(Gamepanel gp,int x,int y,int w,int h) {

    this.gp = gp;
    width = w;
    height = h;
    this.x = x;
    this.y = y;
    for(int i=0;i<4;i++){
        imm[i] = new ImageIcon("mon2m"+(i+1)+".png");
    }
    for(int i=0;i<3;i++){
        ima[i] = new ImageIcon("mon2a"+(i+1)+".png");
    }

}

```



```

public mon3(Gamepanel gp,int x,int y,int w,int h) {

    this.gp = gp;
    width = w;
    height = h;
    this.x = x;
    this.y = y;
    for(int i=0;i<im.length;i++){
        im[i] = new ImageIcon("mon3a"+(i+1)+".png");
    }

}

public mon4(Gamepanel gp,int x,int y,int w,int h) {

    this.gp = gp;
    width = w;
    height = h;
    this.x = x;
    this.y = y;

    im = new ImageIcon(filename: "monmon.png");

}

```

ทั้ง 4 คลาสมี constructor ที่รับมีการรับพารามิเตอร์ชนิด Gamepanel และมีชนิด int x,y เพื่อกำหนดตำแหน่ง int w เพื่อกำหนดความกว้าง และ int h เพื่อกำหนดความสูง

## Encapsulation

```
public Game w;  
public int x,y;  
public int width,height;  
public int speedx,speedy,speedc ;  
public int speed;  
public int ground = 480;  
public int jumphigh ;  
public ImageIcon[] im = new ImageIcon[6];  
public ImageIcon[] imb = new ImageIcon[6];  
public int countf = 0;  
public int countb = 0;  
public int side = 0;  
public boolean isJumping = false;  
public boolean falling = false;  
public boolean climbing = false;  
public boolean blessed = false;  
Gamepanel gp;  
keyhandler keyH;
```

ทุกตัวแปรในคลาส Totoro ถูกกำหนดเป็น Public

ทั้งหมดเนื่องจากต้องมีการเข้าถึงตัวแปรจากภายนอกคลาส Totoro

```
private int minute = 0;
private int second = 0;
private boolean gamewin = false;
private final ImageIcon shield = new ImageIcon(filename:"shield.png");
private final ImageIcon imgstate1 = new ImageIcon(filename:"state1.png");
private final ImageIcon imgstate2 = new ImageIcon(filename:"state2.png");
private final ImageIcon imgstate3 = new ImageIcon(filename:"map3.png");
private final ImageIcon win = new ImageIcon(filename:"You win.png");
private ImageIcon bggameover = new ImageIcon(filename:"You Lose.png");
private final ImageIcon h1 = new ImageIcon(filename:"h1.png");
private final ImageIcon h2 = new ImageIcon(filename:"h2.png");
```

มีการใช้ **private** เพื่อกำหนดขอบเขตของตัวแปรเนื่องจากไม่อนุญาตให้คลาสภายนอกเข้าถึงตัวแปร

```
public boolean gover = false ;
public int state = 1;
public int HP=3;
```

```
public boolean state1 = false;
public boolean state2 = false;
public boolean state3 = false;
```

ทุกตัวแปรในคลาส ถูกกำหนดเป็น **Public** ทั้งหมดเนื่องจากต้องมีการเข้าถึงตัวแปรจากภายนอกคลาส

## Composition

```
Gamepanal gp;  
keyhandler keyH;
```

```
public Totoro(Gamepanal gp, keyhandler keyH) {  
  
    this.gp = gp;  
    this.keyH = keyH;  
  
    setDefaultvalues();  
}
```

```
public class FloorManager {  
  
    Gamepanal gp;  
    Floor[] f;
```

มีการใช้ composition เพื่อกำหนดความสามารถให้กับตัวแปร gp เป็น Gamepanal และ f เป็น Array ซึ่งแต่ละตัวใน Array มีชนิดเป็น Floor

## Polymorphism

```
public class Gamepanal extends JPanel implements Runnable{
```

```

@Override
public void run() {

    double drawInterval = 10000000000/FPS;
    double nextDrawTime = System.nanoTime() + drawInterval;

    while(gameThread != null)
    {

        update();
        repaint();

        try{
            double remainingTime = nextDrawTime - System.nanoTime();
            remainingTime = remainingTime/1000000;

            if(remainingTime < 0){
                remainingTime = 0;
            }
            Thread.sleep((long)remainingTime);

            nextDrawTime += drawInterval;

        }catch(Exception e){
            e.printStackTrace();
        }
    }
}

```

มีการoverride method run() จากการ implements Runnable

```

public class restart implements ActionListener{

    @Override
    public void actionPerformed(ActionEvent e) {
        if(e.getSource() == Brestart){
            HP = 3;
            gover = false;
            state1 = true;
            state2 = false;
            state3 = false;
            totoro.x = 0;
            totoro.y = 480;
            totoro.isJumping = false;
            totoro.speedx = 0;
            totoro.speedy = 0;
            minute = 0;
            second = 0;
            timestart = false;
            gamewin = false;
        }
    }
}

```

มีการoverride method actionPerformed() จากการ implements  
ActionListener

```
@Override  
public void paintComponent(Graphics g) {  
    ...  
}
```

มีการoverride method paintComponent() จากการ extends JPanel

## Inheritance

```
public class Game extends JFrame {  
    ...  
}
```

Game สืบทอดจากคลาส JFrame

```
public class Gamepanel extends JPanel {  
    ...  
}
```

Gamepanel สืบทอดจากคลาส JPanel

```
public class pit extends Floor {  
    ...  
}
```

Pit สืบทอดจากคลาส Floor

## GUI

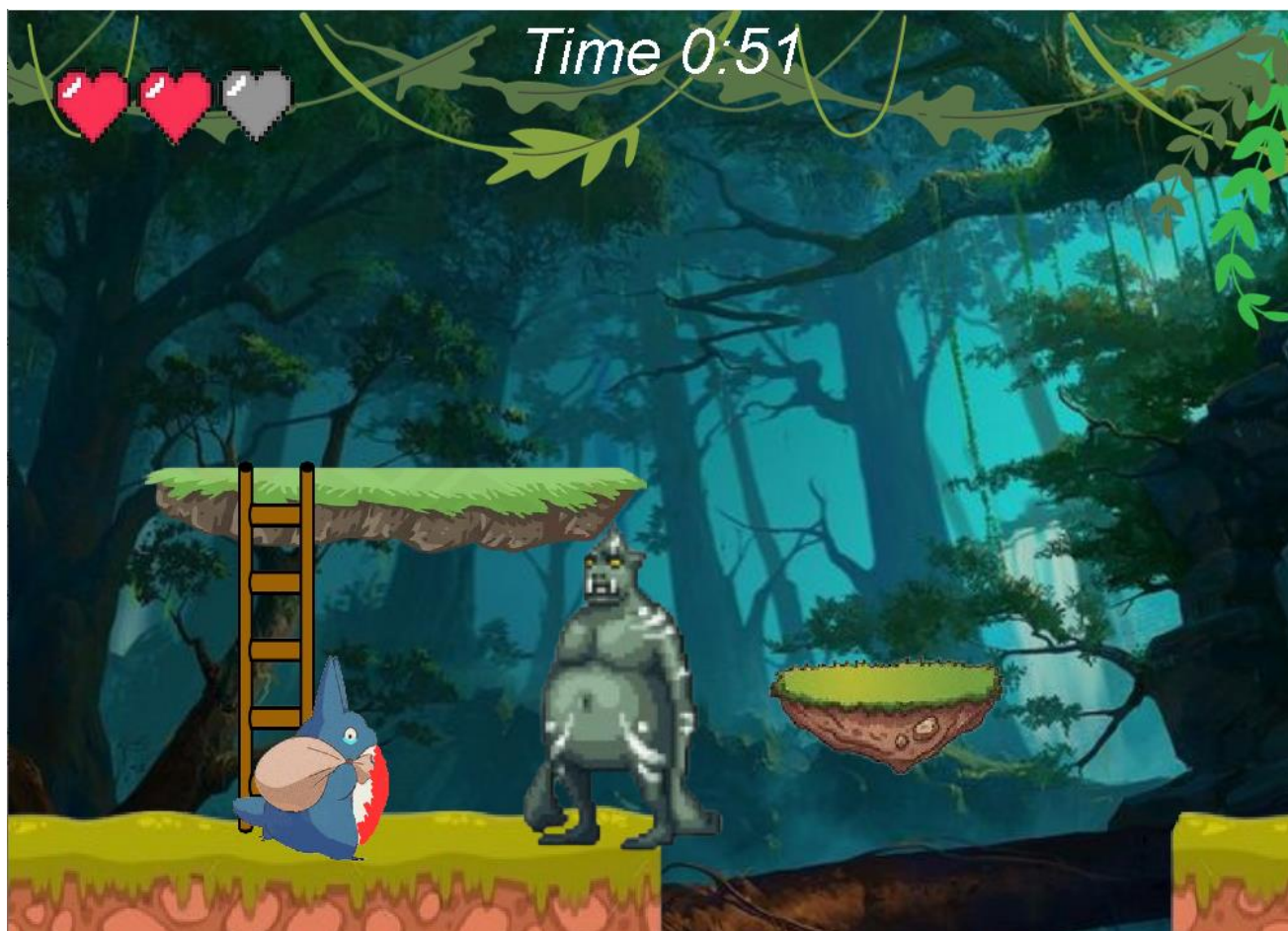




หน้าเมนูเกมที่มีปุ่ม **start** เพื่อเริ่มเกม และปุ่ม **Exit** เพื่อปิดเกม



เมื่อเริ่มเกมก็จะเข้าสู่ด่าน 1



เมื่อถูกมอนสเตอร์โจมตีก็จะเสียหัวใจไป





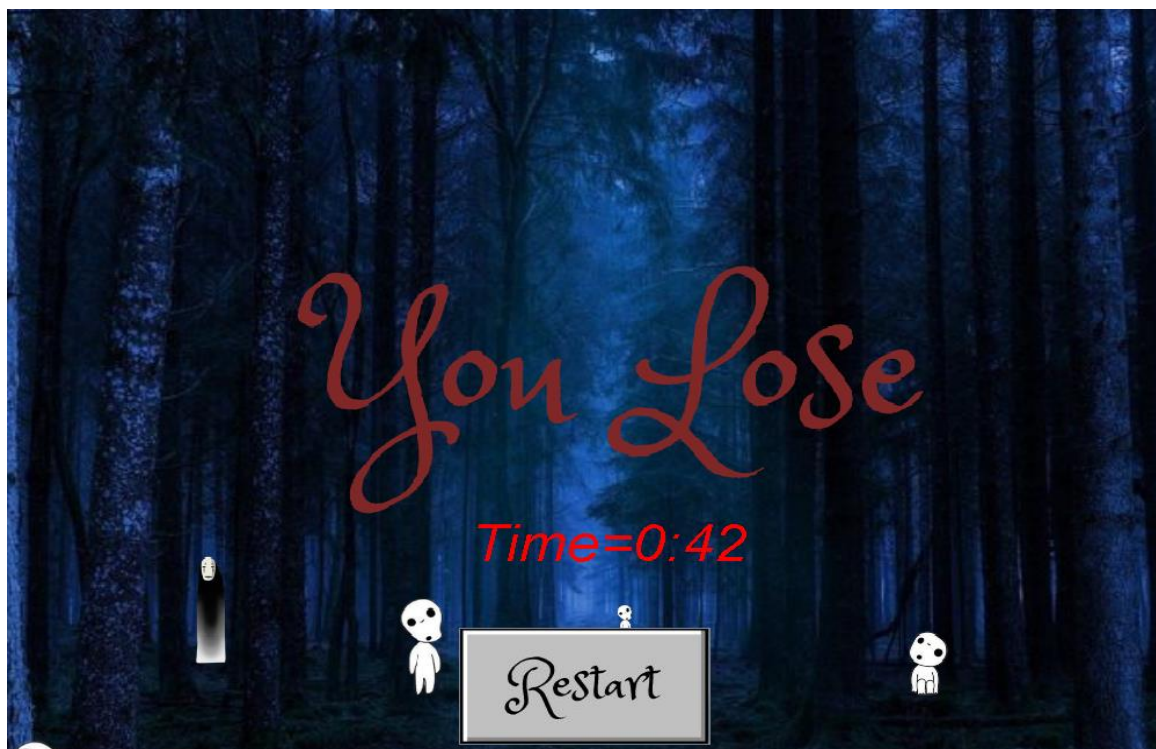
เมื่อผ่านด่าน 1 ก็เข้าสู่ด่านที่ 2



เมื่อกดปุ่ม E ตรงรูปปั้นก็จะได้รับเกราะป้องกันที่สามารถป้องกันการโจมตีของมอนสเตอร์ได้ 1 ครั้ง



เมื่อผ่านด่าน 2 ก็เข้าสู่ด่านที่ 3



เมื่อหัวใจหมดก็จะขึ้นหน้าgameover





เมื่อผ่านด่านทุกด่านก็จะขึ้นหน้าชนะเกมขึ้นมา

## บทที่ 3 สรุป

### ปัญหาที่พบระหว่างการทำงาน

เนื่องจากมีความรู้เรื่องการสร้างเกมด้วย java

ไม่มากพอจึงทำให้เสียเวลาในการทำความเข้าใจในแต่ส่วนและค่อยเป็นค่อยไปทำให้ทำงานออกมาเสร็จได้ล่าช้ากว่ากำหนด

### จุดเด่นของโปรแกรม

ใช้ Gui ที่เป็นภาพการ์ตูนที่มีสีสันทำให้ดูสวยงาม

คำแนะนำสำหรับผู้สอนที่อยากให้อธิบาย หรือที่เรียนแล้วไม่เข้าใจ  
หรืออยากให้เพิ่มสำหรับน้อง ๆ รุ่นต่อไป

อยากให้อาจารย์สอนและเฉลยข้อที่เป็นงานให้ห้องและการบ้านแบบละเอียดที่ละงานเพื่อที่จะได้เข้าใจมากขึ้นและจะได้รู้ว่าผิดตรงไหนต้องแก้ตรงไหนและเข้าใจล่อจิดได้ถูกต้อง