Mario Go!

LET'S PLAY

Mario Go!

โดย

นางสาว ภัทร์พลอย นิธิวาณิชย์ 6530300392

เสนอ

อาจารย์ กุลวดี สมบูรณ์วิวัฒน์

Fundamentals II รหัสวิชา 03603171

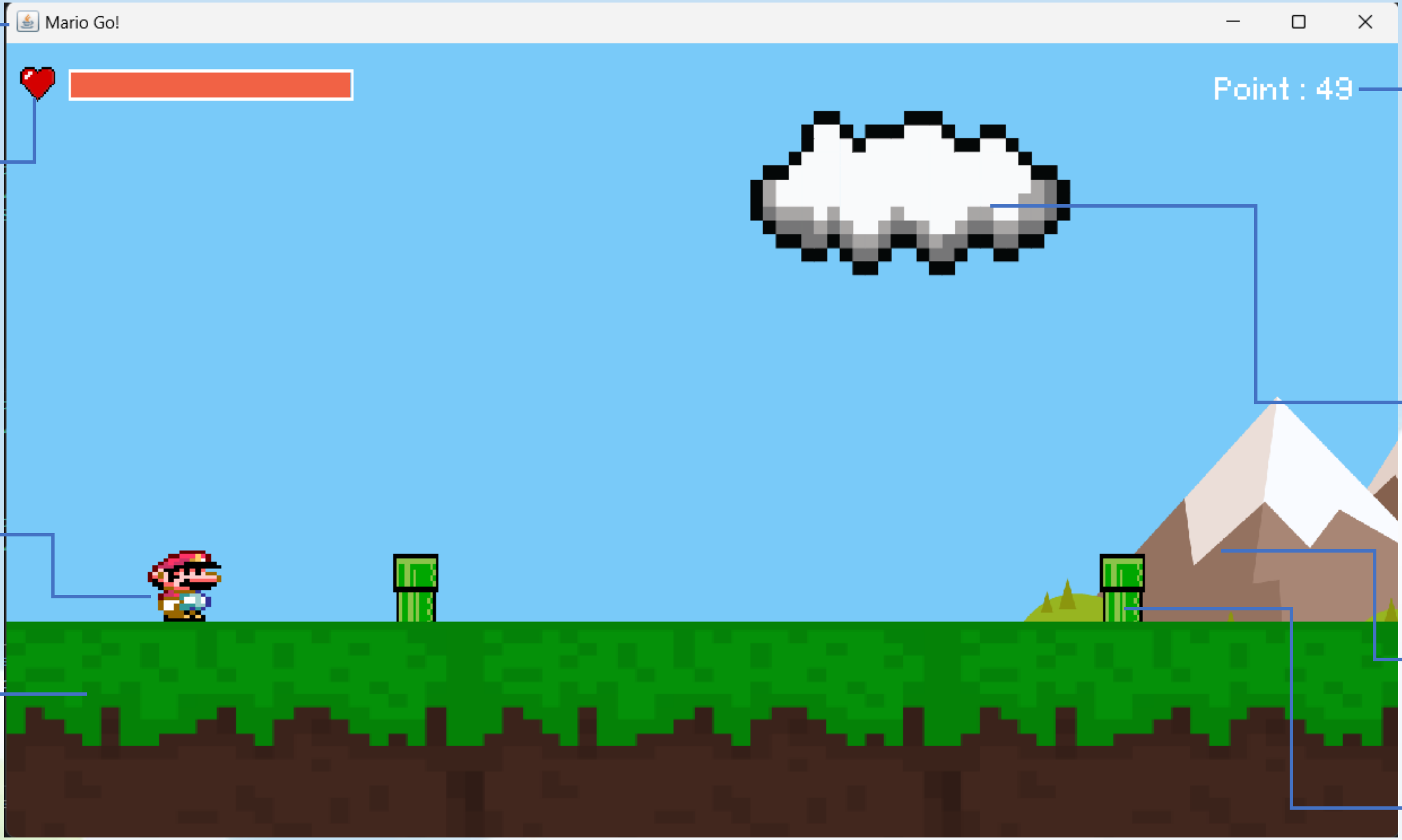
ภาคเรียนที่ 2 ปีการศึกษา 2565

มหาวิทยาลัยเกษตรศาสตร์ วิทยาเขตศรีราชา

เหตุผลการทำโปรเจค

- สร้างเกม Mario Go! เพื่อความสนุกสนาน
- เพื่อเป็นการทบทวนบทเรียนที่ผ่านมา
- เพื่อทดสอบความเข้าใจในภาษา Java เบื้องต้น
- เป็นการใช้เวลาว่างให้เกิดประโยชน์

แนวคิดการออกแบบ UI



Title ชื่อเกม

Health พลังชีวิต

Character Mario

Ground

Point

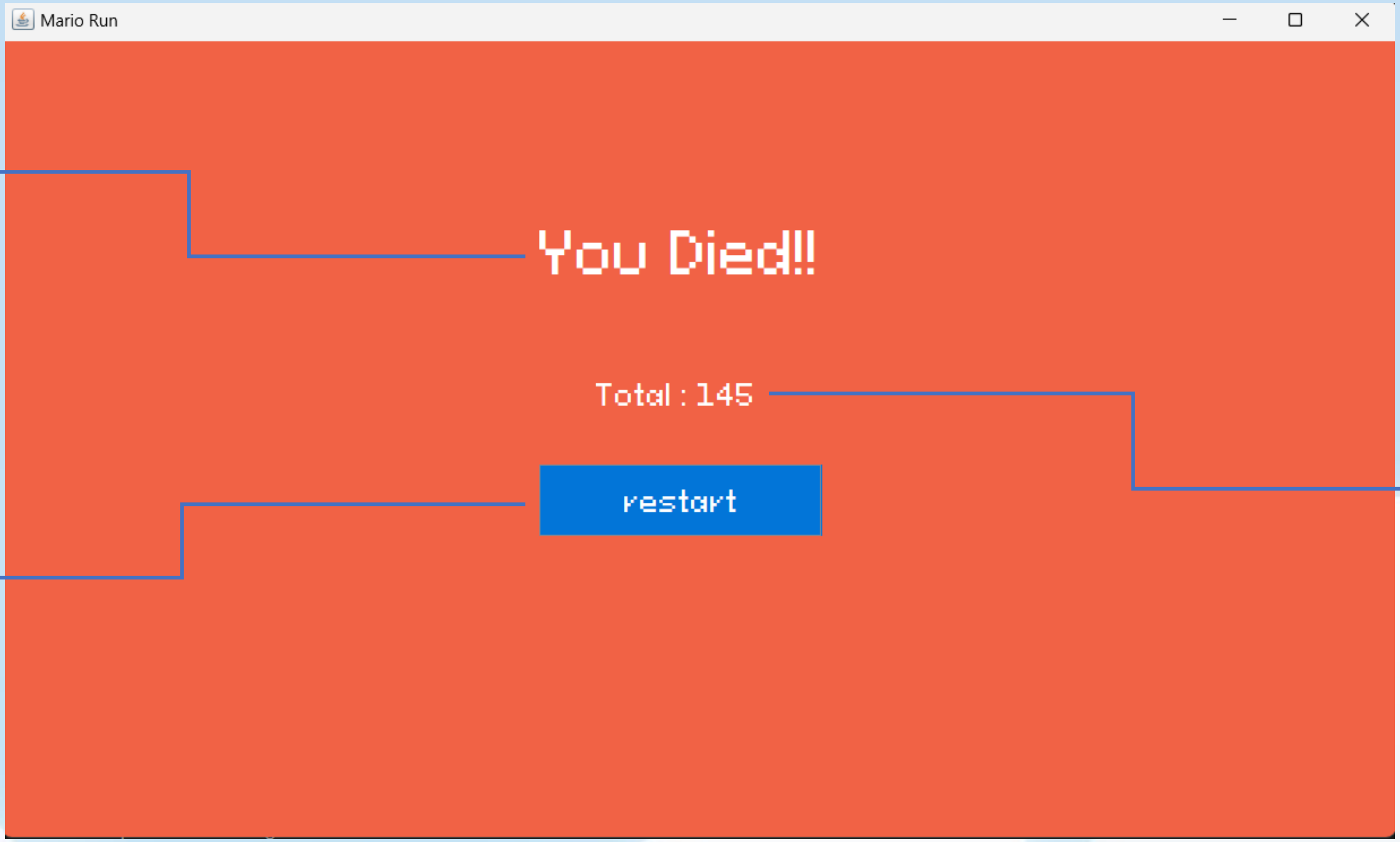
แต้มที่ได้จากการ
กระโดดข้ามสิ่งกีดขวาง

Background

Background

Enemy pipe

แนวคิดการออกแบบ UI



status

สถานะ

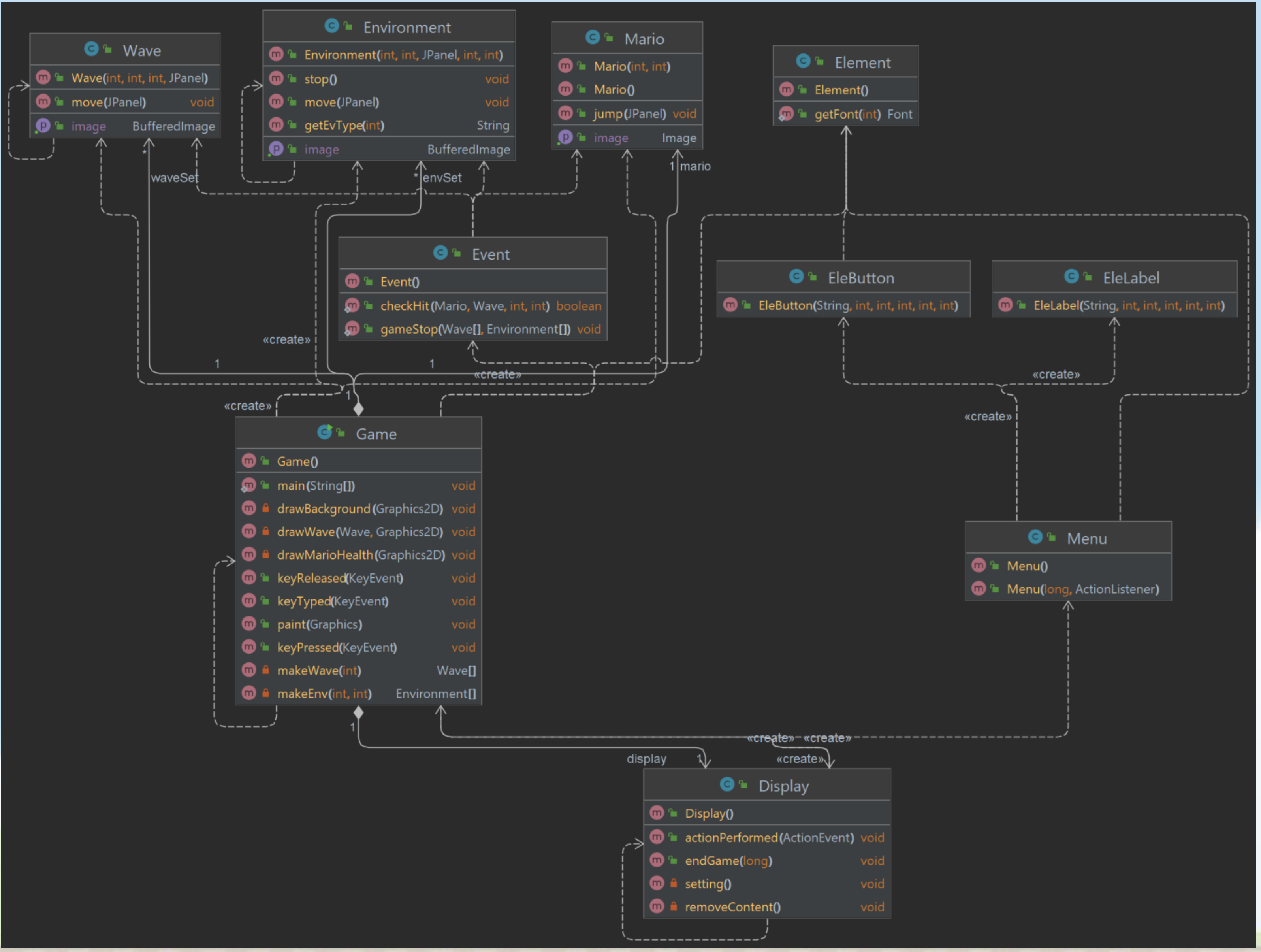
Restart

เมื่อกดจะเริ่มเกมใหม่

Showpoint

แสดงแต้ม

Class Diagram



Wave
- Wave(int, int, int, JPanel)
- move(JPanel) void
- image BufferedImage

Environment
- Environment(int, int, JPanel, int, int)
- stop() void
- move(JPanel) void
- getEvType(int) String
- image BufferedImage

Mario
- Mario(int, int)
- Mario()
- jump(JPanel) void
- image Image

Element
- Element()
- getFont(int) Font

Event
- Event()
- checkHit(Mario, Wave, int, int) boolean
- gameStop(Wave[], Environment[]) void

EleButton
- EleButton(String, int, int, int, int, int)

EleLabel
- EleLabel(String, int, int, int, int, int)

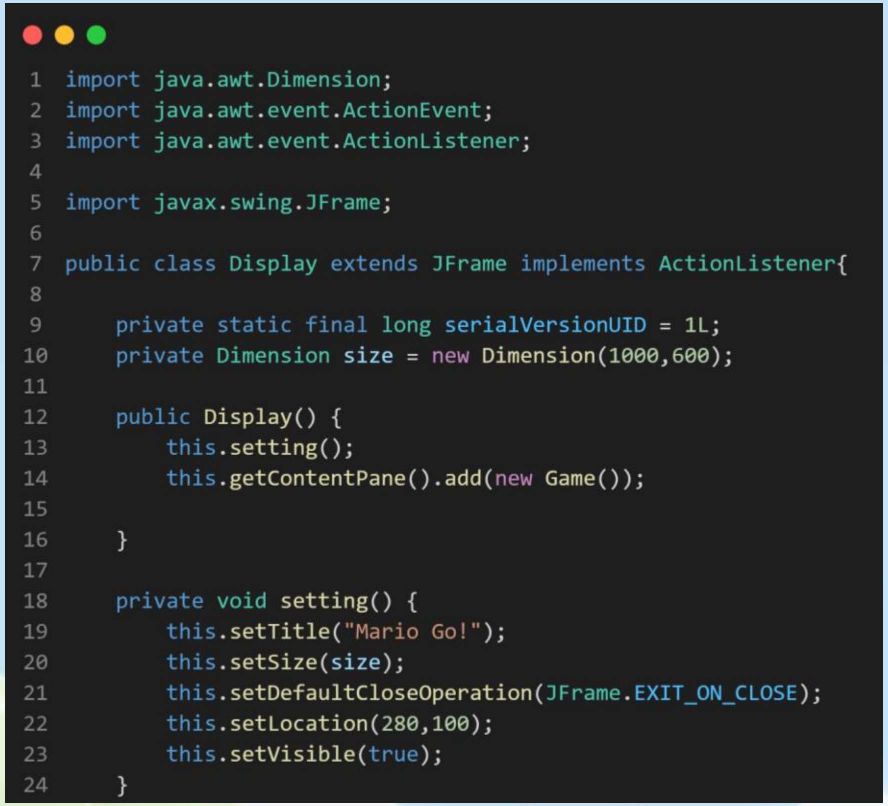
Game
- Game()
- main(String[]) void
- drawBackground(Graphics2D) void
- drawWave(Wave, Graphics2D) void
- drawMarioHealth(Graphics2D) void
- keyReleased(KeyEvent) void
- keyTyped(KeyEvent) void
- paint(Graphics) void
- keyPressed(KeyEvent) void
- makeWave(int) Wave[]
- makeEnv(int, int) Environment[]

Menu
- Menu()
- Menu(long, ActionListener)

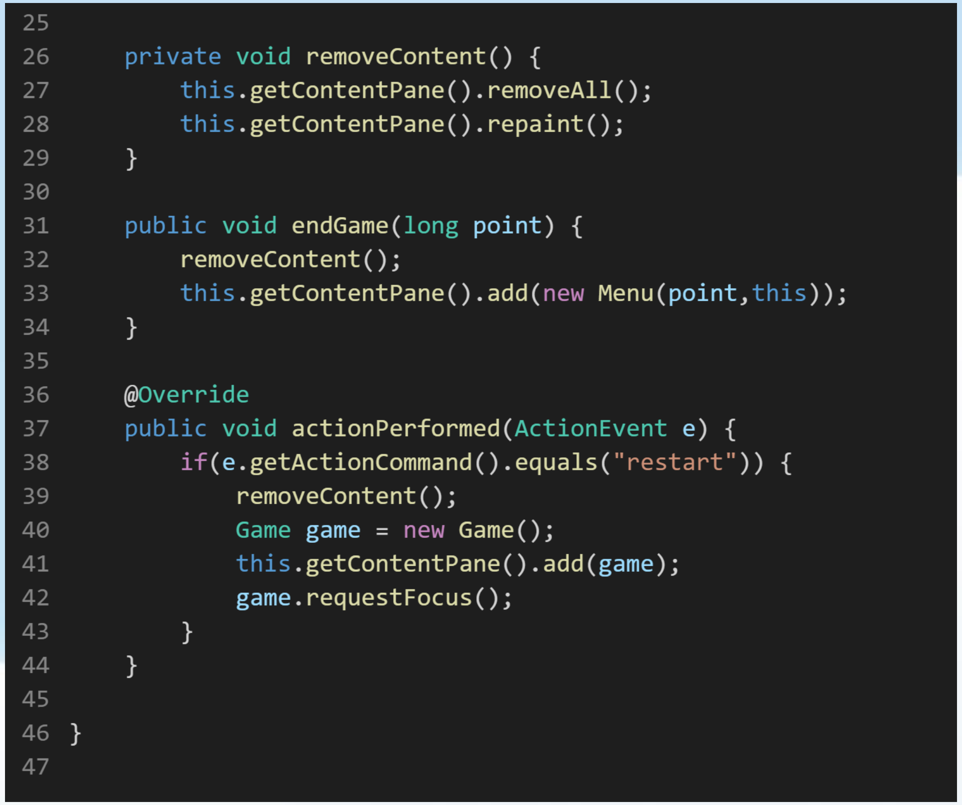
Display
- Display()
- actionPerformed(ActionEvent) void
- endGame(long) void
- setting() void
- removeContent() void

Algorithm

Display

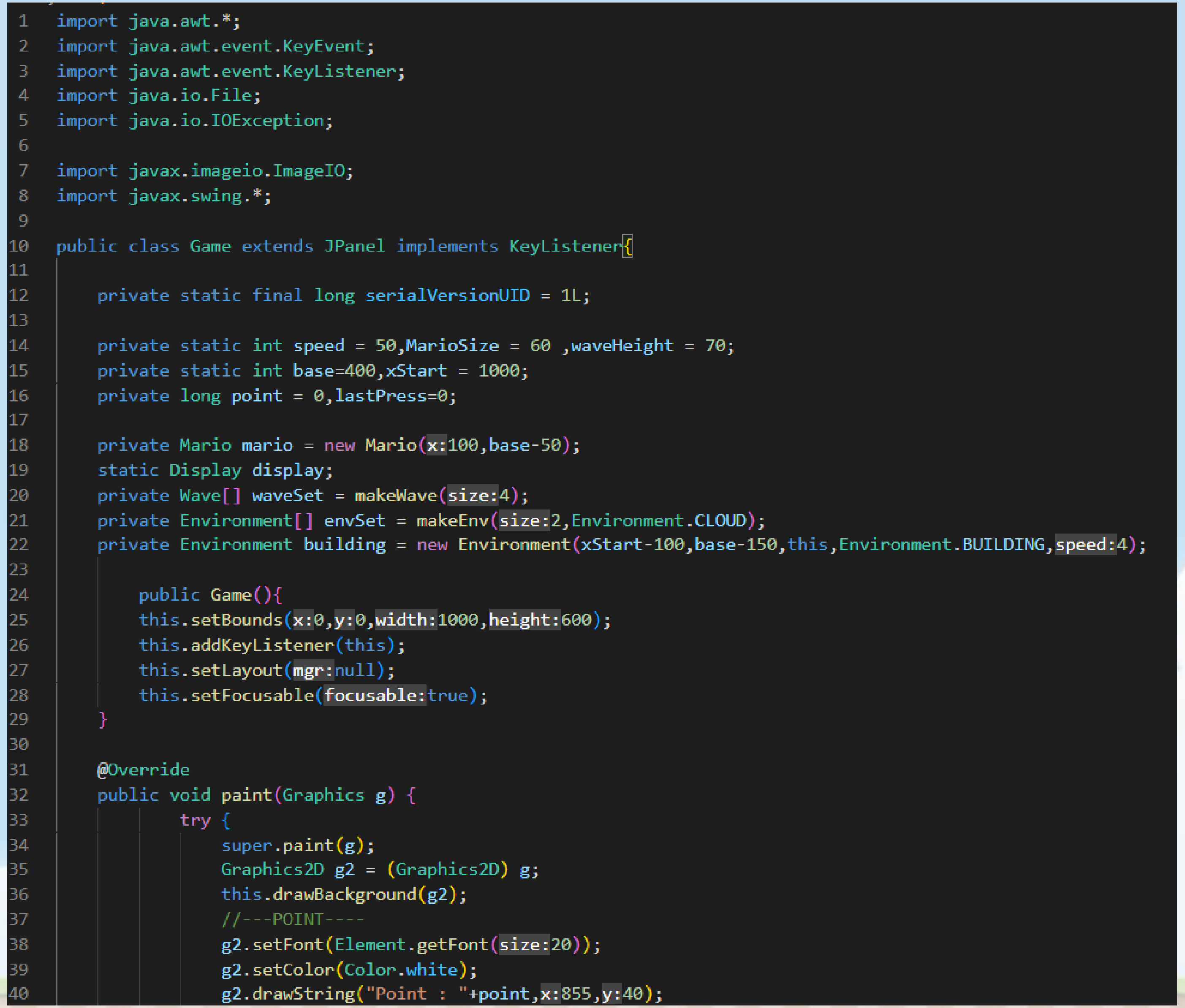


```java
import java.awt.Dimension;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.JFrame;

public class Display extends JFrame implements ActionListener{

    private static final long serialVersionUID = 1L;
    private Dimension size = new Dimension(1000,600);

    public Display() {
        this.setting();
        this.getContentPane().add(new Game());

    }

    private void setting() {
        this.setTitle("Mario Go!");
        this.setSize(size);
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setLocation(280,100);
        this.setVisible(true);
    }
```

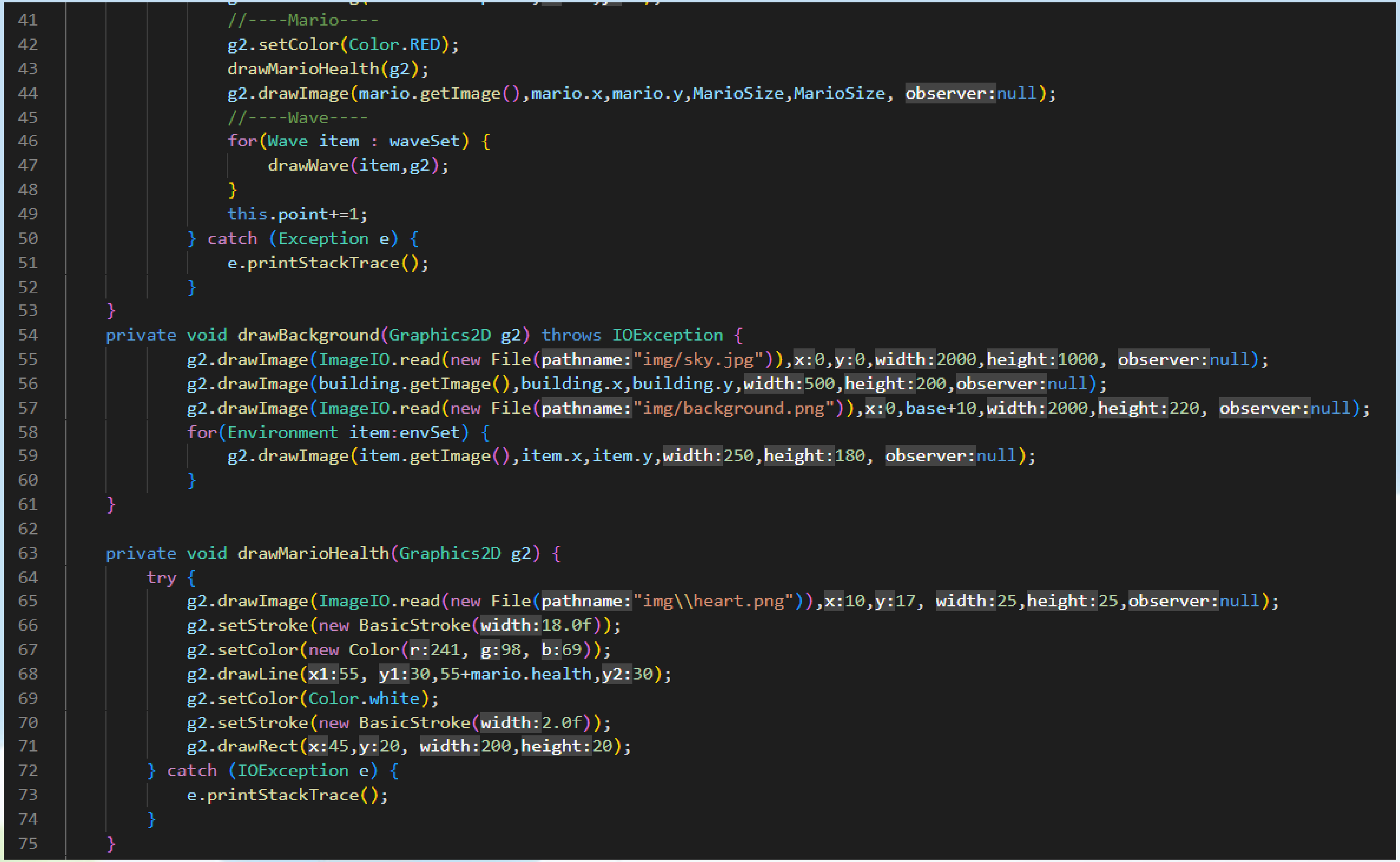


```java

    private void removeContent() {
        this.getContentPane().removeAll();
        this.getContentPane().repaint();
    }

    public void endGame(long point) {
        removeContent();
        this.getContentPane().add(new Menu(point,this));
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        if(e.getActionCommand().equals("restart")) {
            removeContent();
            Game game = new Game();
            this.getContentPane().add(game);
            game.requestFocus();
        }
    }

}

```

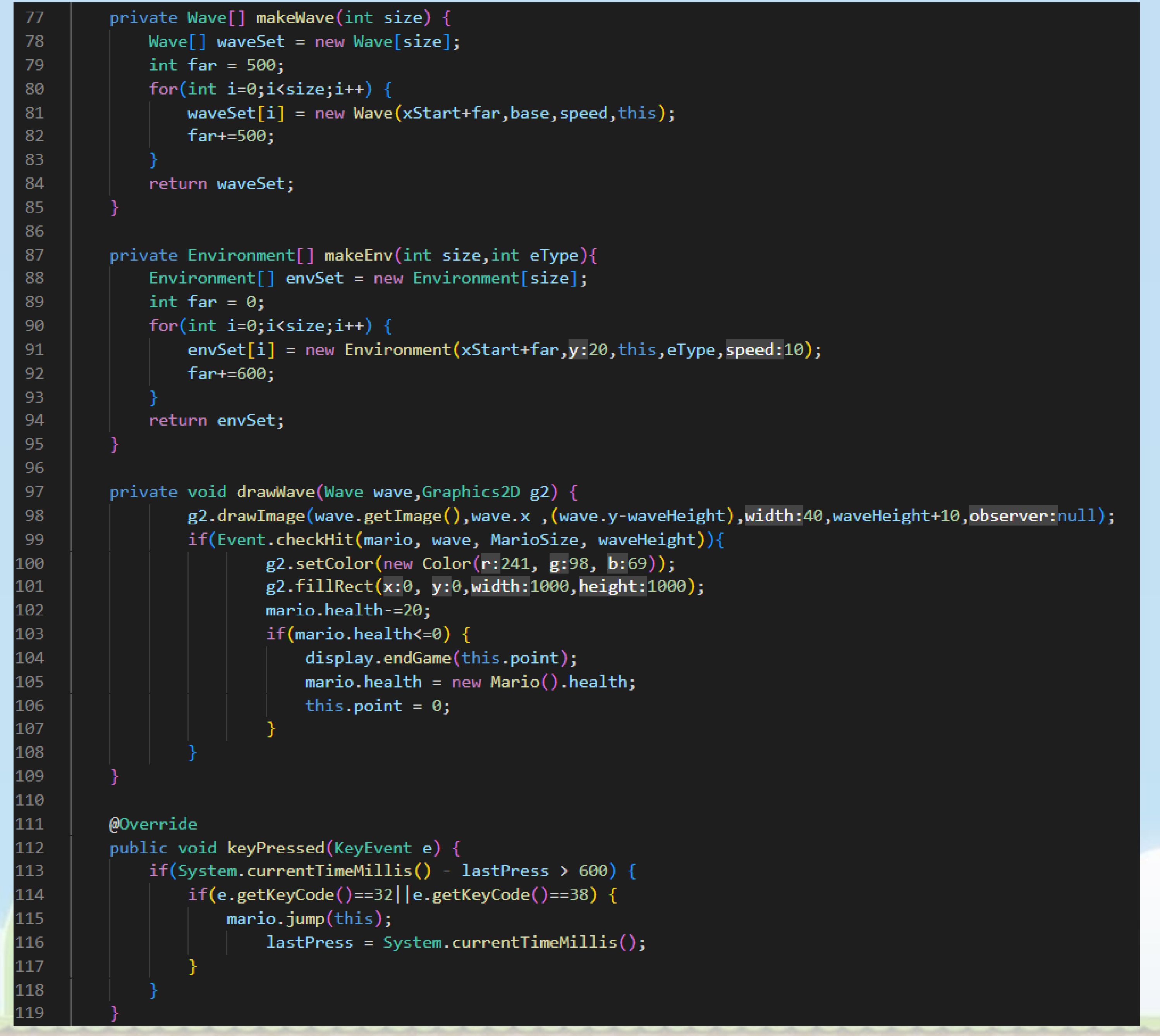
Game



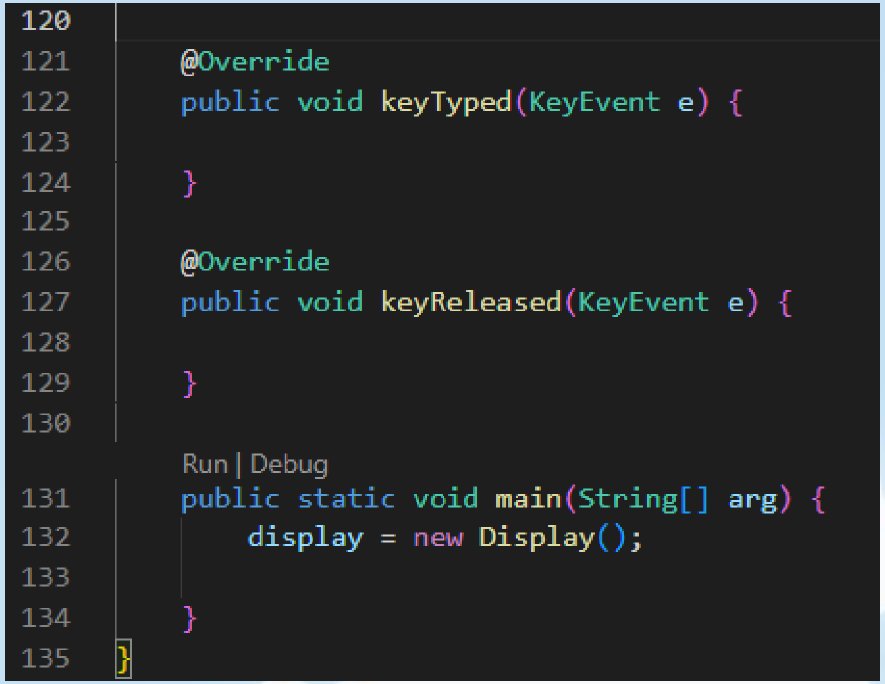
```java
import java.awt.*;
import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;
import java.io.File;
import java.io.IOException;

import javax.imageio.ImageIO;
import javax.swing.*;

public class Game extends JPanel implements KeyListener{

    private static final long serialVersionUID = 1L;

    private static int speed = 50,MarioSize = 60 ,waveHeight = 70;
    private static int base=400,xStart = 1000;
    private long point = 0,lastPress=0;

    private Mario mario = new Mario(x:100,base-50);
    static Display display;
    private Wave[] waveSet = makeWave(size:4);
    private Environment[] envSet = makeEnv(size:2,Environment.CLOUD);
    private Environment building = new Environment(xStart-100,base-150,this,Environment.BUILDING,speed:4);

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

    @Override
    public void paint(Graphics g) {
            try {
                super.paint(g);
                Graphics2D g2 = (Graphics2D) g;
                this.drawBackground(g2);
                //---POINT----
                g2.setFont(Element.getFont(size:20));
                g2.setColor(Color.white);
                g2.drawString("Point : "+point,x:855,y:40);
```



```
                //----Mario----
                g2.setColor(Color.RED);
                drawMarioHealth(g2);
                g2.drawImage(mario.getImage(),mario.x,mario.y,MarioSize,MarioSize, observer:null);
                //----Wave----
                for(Wave item : waveSet) {
                    drawWave(item,g2);
                }
                this.point+=1;
            } catch (Exception e) {
                e.printStackTrace();
            }
    }
    private void drawBackground(Graphics2D g2) throws IOException {
            g2.drawImage(ImageIO.read(new File(pathname:"img/sky.jpg")),x:0,y:0,width:2000,height:1000, observer:null);
            g2.drawImage(building.getImage(),building.x,building.y,width:500,height:200,observer:null);
            g2.drawImage(ImageIO.read(new File(pathname:"img/background.png")),x:0,base+10,width:2000,height:220, observer:null);
            for(Environment item:envSet) {
                g2.drawImage(item.getImage(),item.x,item.y,width:250,height:180, observer:null);
            }
    }

    private void drawMarioHealth(Graphics2D g2) {
        try {
            g2.drawImage(ImageIO.read(new File(pathname:"img\\heart.png")),x:10,y:17, width:25,height:25,observer:null);
            g2.setStroke(new BasicStroke(width:18.0f));
            g2.setColor(new Color(r:241, g:98, b:69));
            g2.drawLine(x1:55, y1:30,55+mario.health,y2:30);
            g2.setColor(Color.white);
            g2.setStroke(new BasicStroke(width:2.0f));
            g2.drawRect(x:45,y:20, width:200,height:20);
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
```

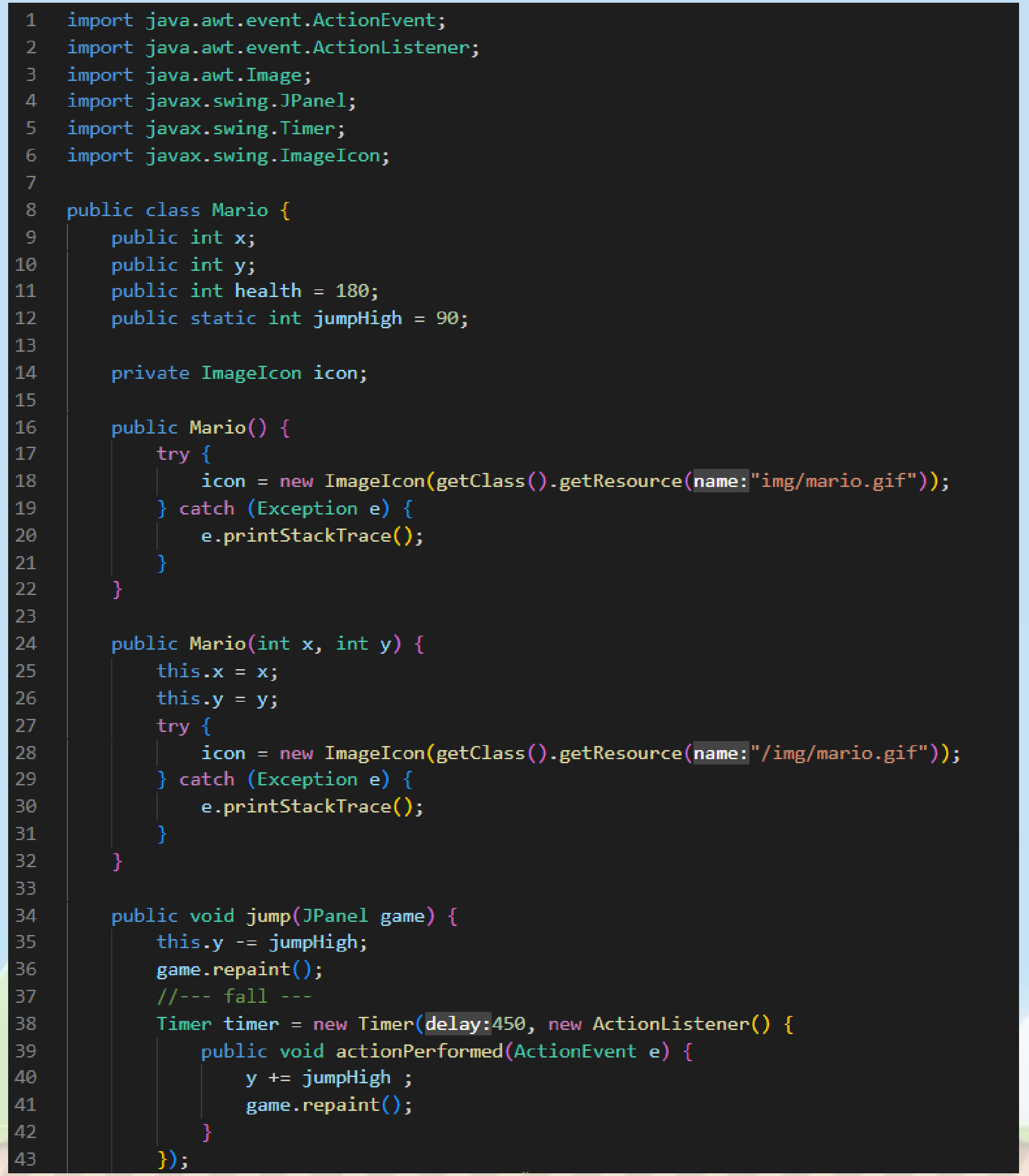


```
    private Wave[] makeWave(int size) {
        Wave[] waveSet = new Wave[size];
        int far = 500;
        for(int i=0;i<size;i++) {
            waveSet[i] = new Wave(xStart+far,base,speed,this);
            far+=500;
        }
        return waveSet;
    }

    private Environment[] makeEnv(int size,int eType){
        Environment[] envSet = new Environment[size];
        int far = 0;
        for(int i=0;i<size;i++) {
            envSet[i] = new Environment(xStart+far,y:20,this,eType,speed:10);
            far+=600;
        }
        return envSet;
    }

    private void drawWave(Wave wave,Graphics2D g2) {
            g2.drawImage(wave.getImage(),wave.x ,(wave.y-waveHeight),width:40,waveHeight+10,observer:null);
            if(Event.checkHit(mario, wave, MarioSize, waveHeight)){
                    g2.setColor(new Color(r:241, g:98, b:69));
                    g2.fillRect(x:0, y:0,width:1000,height:1000);
                    mario.health-=20;
                    if(mario.health<=0) {
                        display.endGame(this.point);
                        mario.health = new Mario().health;
                        this.point = 0;
                    }
            }
    }

    @Override
    public void keyPressed(KeyEvent e) {
        if(System.currentTimeMillis() - lastPress > 600) {
            if(e.getKeyCode()==32||e.getKeyCode()==38) {
                mario.jump(this);
                    lastPress = System.currentTimeMillis();
            }
        }
    }
```

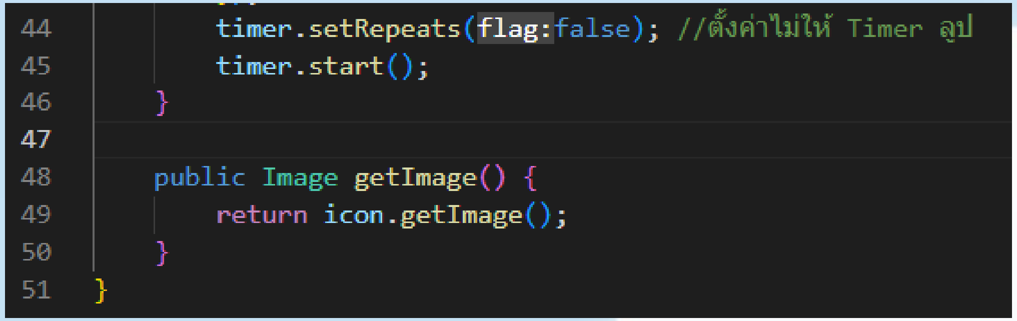


```

    @Override
    public void keyTyped(KeyEvent e) {

    }

    @Override
    public void keyReleased(KeyEvent e) {

    }

    Run | Debug
    public static void main(String[] arg) {
        display = new Display();

    }
}
```

Mario

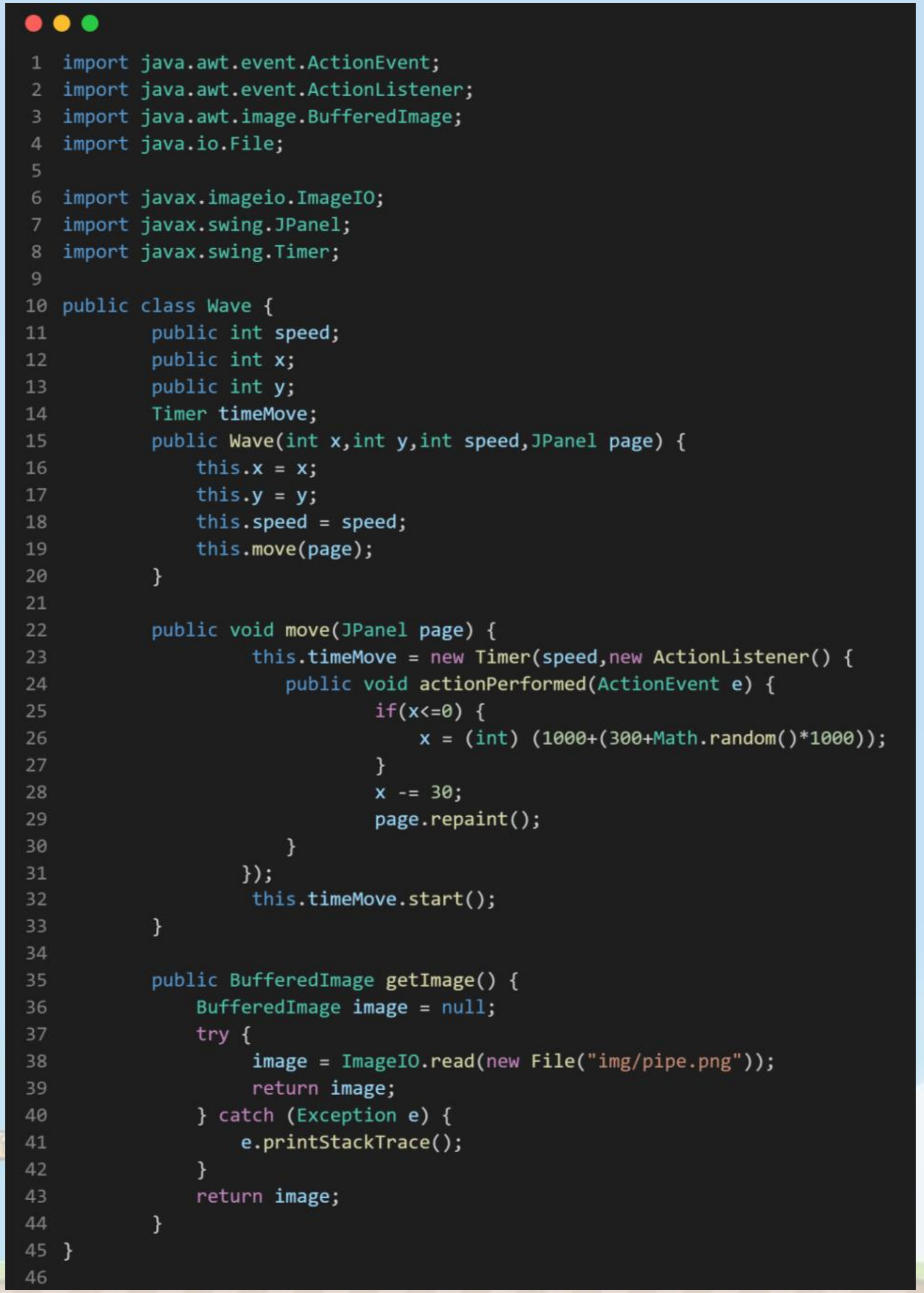


```java
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.Image;
import javax.swing.JPanel;
import javax.swing.Timer;
import javax.swing.ImageIcon;

public class Mario {
    public int x;
    public int y;
    public int health = 180;
    public static int jumpHigh = 90;

    private ImageIcon icon;

    public Mario() {
        try {
            icon = new ImageIcon(getClass().getResource(name:"img/mario.gif"));
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public Mario(int x, int y) {
        this.x = x;
        this.y = y;
        try {
            icon = new ImageIcon(getClass().getResource(name:"/img/mario.gif"));
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public void jump(JPanel game) {
        this.y -= jumpHigh;
        game.repaint();
        //--- fall ---
        Timer timer = new Timer(delay:450, new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                y += jumpHigh ;
                game.repaint();
            }
        });
```



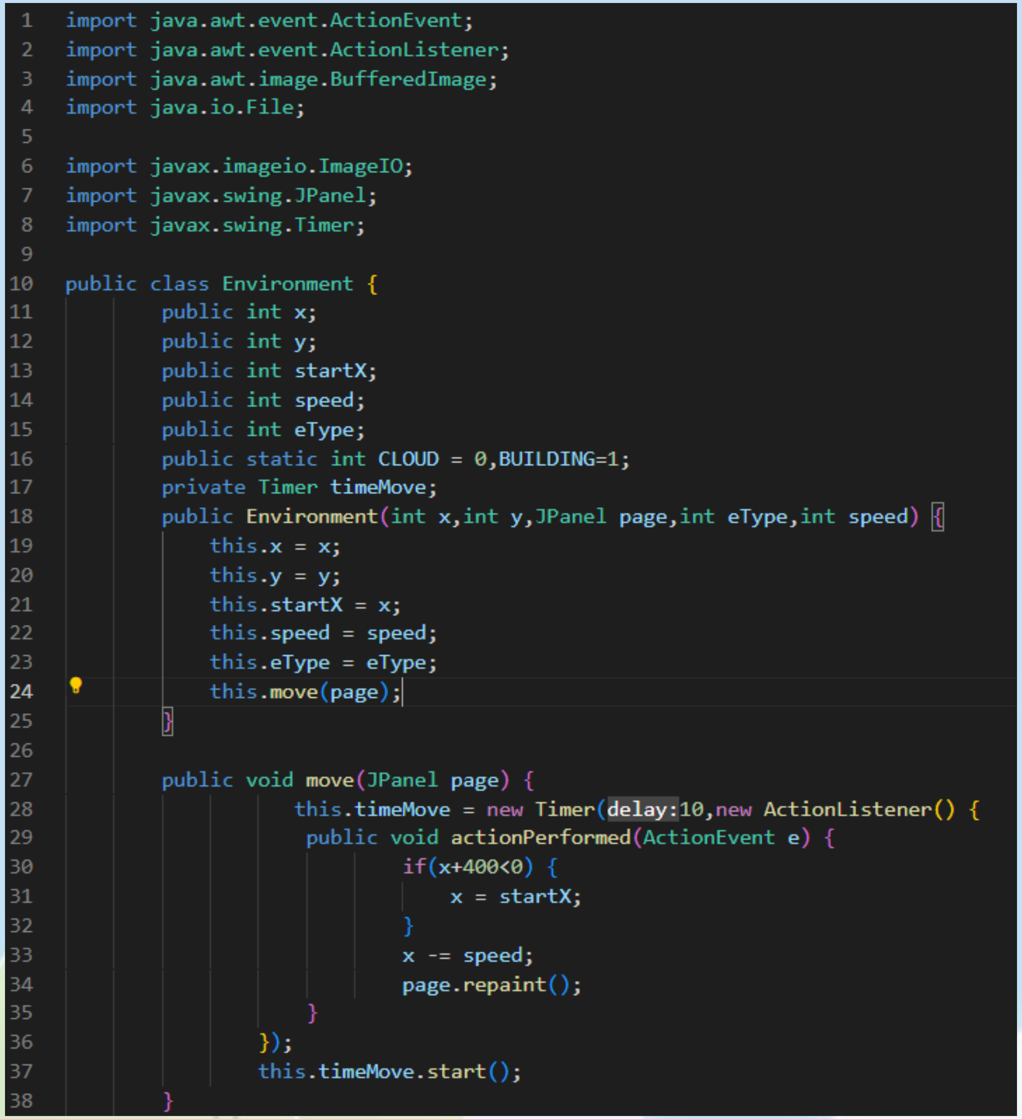
```java
        timer.setRepeats(flag:false); //ตั้งค่าไม่ให้ Timer ลูป
        timer.start();
    }

    public Image getImage() {
        return icon.getImage();
    }
}
```

Wave

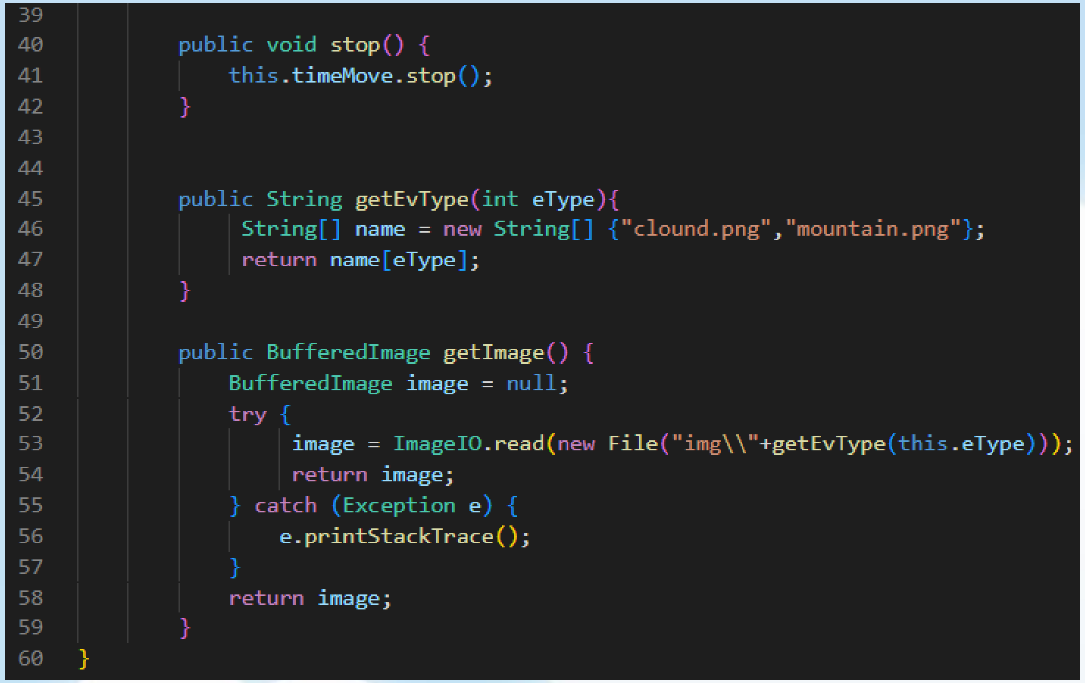


```java
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.image.BufferedImage;
import java.io.File;

import javax.imageio.ImageIO;
import javax.swing.JPanel;
import javax.swing.Timer;

public class Wave {
        public int speed;
        public int x;
        public int y;
        Timer timeMove;
        public Wave(int x,int y,int speed,JPanel page) {
            this.x = x;
            this.y = y;
            this.speed = speed;
            this.move(page);
        }

        public void move(JPanel page) {
                 this.timeMove = new Timer(speed,new ActionListener() {
                    public void actionPerformed(ActionEvent e) {
                            if(x<=0) {
                                x = (int) (1000+(300+Math.random()*1000));
                            }
                            x -= 30;
                            page.repaint();
                    }
                });
                 this.timeMove.start();
        }

        public BufferedImage getImage() {
            BufferedImage image = null;
            try {
                 image = ImageIO.read(new File("img/pipe.png"));
                 return image;
            } catch (Exception e) {
                e.printStackTrace();
            }
            return image;
        }
}
```

Environment

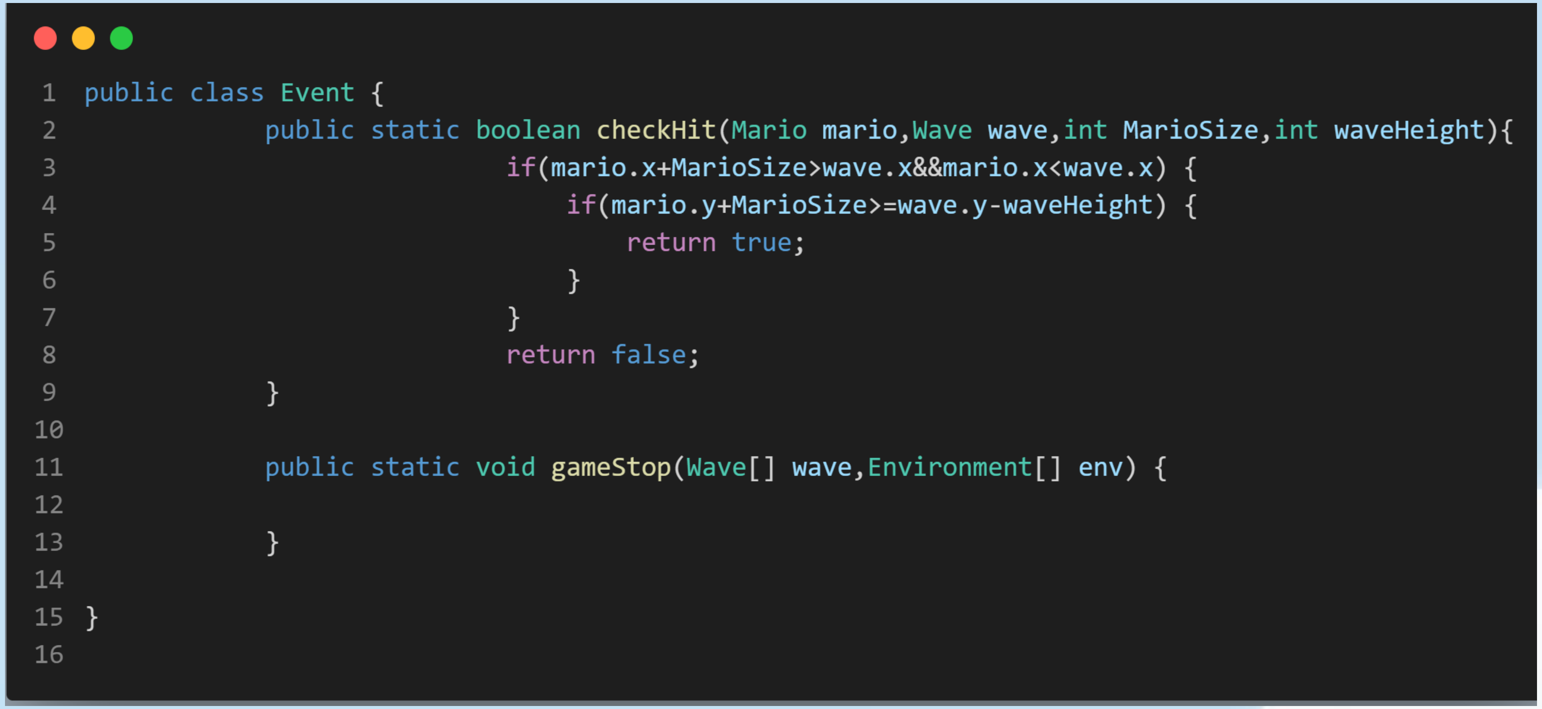


```java
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.image.BufferedImage;
import java.io.File;

import javax.imageio.ImageIO;
import javax.swing.JPanel;
import javax.swing.Timer;

public class Environment {
        public int x;
        public int y;
        public int startX;
        public int speed;
        public int eType;
        public static int CLOUD = 0,BUILDING=1;
        private Timer timeMove;
        public Environment(int x,int y,JPanel page,int eType,int speed) {
            this.x = x;
            this.y = y;
            this.startX = x;
            this.speed = speed;
            this.eType = eType;
            this.move(page);
        }

        public void move(JPanel page) {
                   this.timeMove = new Timer(delay:10,new ActionListener() {
                    public void actionPerformed(ActionEvent e) {
                            if(x+400<0) {
                                x = startX;
                            }
                            x -= speed;
                            page.repaint();
                    }
                });
                this.timeMove.start();
        }
```



```java

        public void stop() {
            this.timeMove.stop();
        }


        public String getEvType(int eType){
             String[] name = new String[] {"clound.png","mountain.png"};
             return name[eType];
        }

        public BufferedImage getImage() {
            BufferedImage image = null;
            try {
                 image = ImageIO.read(new File("img\\"+getEvType(this.eType)));
                 return image;
            } catch (Exception e) {
                e.printStackTrace();
            }
            return image;
        }
}
```

Event



```java
public class Event {
            public static boolean checkHit(Mario mario,Wave wave,int MarioSize,int waveHeight){
                            if(mario.x+MarioSize>wave.x&&mario.x<wave.x) {
                                if(mario.y+MarioSize>=wave.y-waveHeight) {
                                    return true;
                                }
                            }
                            return false;
            }

            public static void gameStop(Wave[] wave,Environment[] env) {

            }

}
```

อ้างอิง

https://youtu.be/lItydZs4voo

https://github.com/patploy2546/MARIO_GOGO.git