



## รายงาน

Mini ping pong game

จัดทำโดย

นางสาวชนาธิยา      บริบูรณ์หิรัญ

รหัสนักศึกษา 6304062630172

เสนอ

อาจารย์สถิตย์      ประสมพันธ์

รายงานนี้เป็นส่วนหนึ่งของรายวิชา Object Oriented Programming

(รหัสวิชา 040613222)

ภาคเรียนที่ 1 ปีการศึกษา 2564

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ (วิทยาเขตกรุงเทพฯ)

## บทที่ 1 บทนำ

### ที่มาและความสำคัญของโปรเจกต์

ปัจจุบันเนื่องจากประเทศไทยมีความก้าวหน้าทางด้านเทคโนโลยีเป็นอย่างมากและในช่วงอายุของเด็กปัจจุบันมักใช้ไปกับการหาความเพลิดเพลินโดยการเล่นเกมส์จึงเป็นเหตุผลในการพัฒนาเกมมินิปีปองนี้ขึ้นมา

### ประเภทของโครงการ มินิเกม

#### ประโยชน์

-เพื่อความสนุกสนาน

-ฝึกความอดทน

-ฝึกความไหวพริบ

-มีน้ำใจนักกีฬา

#### ขอบเขตของการศึกษา

1. การศึกษาครั้งนี้มุ่งเน้นไปในวิชา Object Oriented Programming โดยใช้เนื้อหาในรายวิชา Object Oriented Programming ของคณะวิทยาศาสตร์ประยุกต์ สาขาวิทยาการคอมพิวเตอร์
2. ระยะเวลาที่ใช้ในการศึกษาครั้งนี้ดำเนินการในภาคเรียนที่ 1 ปีการศึกษา 2564 ตั้งแต่วันที่ 21 กันยายน 2564 ถึงวันที่ 19 พฤศจิกายน 2564

## บทที่ 2 ส่วนการพัฒนา

### Mini ping pong game

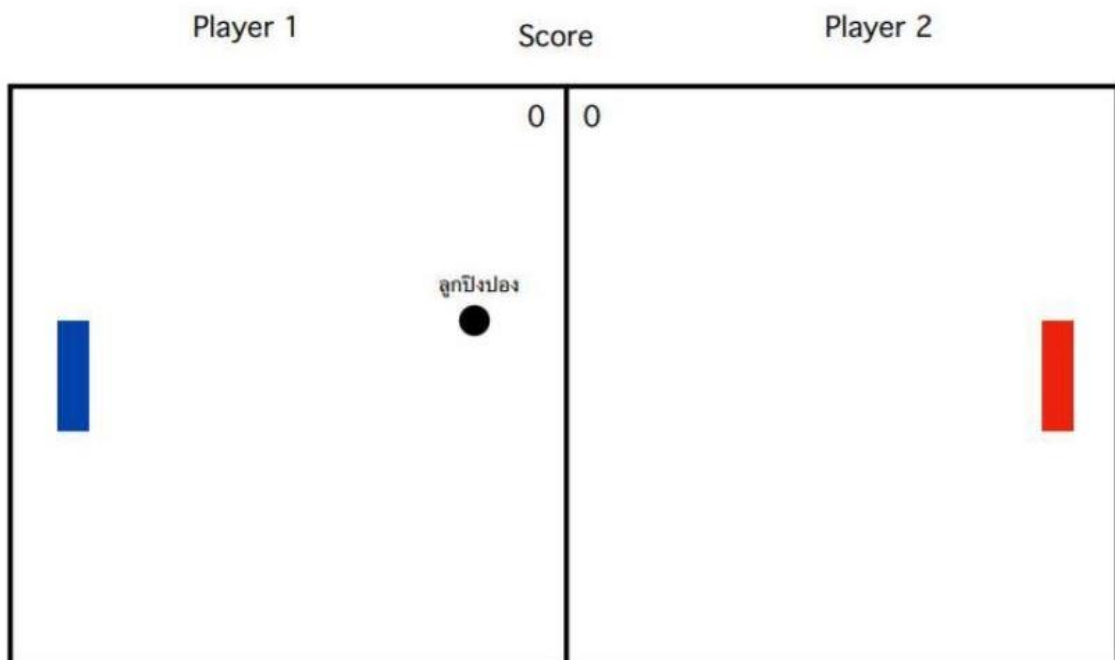
#### รายละเอียดเกม

เกมส้มินิปิงปอง เมื่อผู้เล่นสองคนต้องตีปิงปองโดยถ้าฝ่ายใดตีลูกปิงปองพลาดอีกฝ่ายจะได้คะแนนทันทีและยิ่งตีปิงปองนานเท่าใดความเร็วของลูกปิงปองจะเร็วขึ้นเท่านั้น

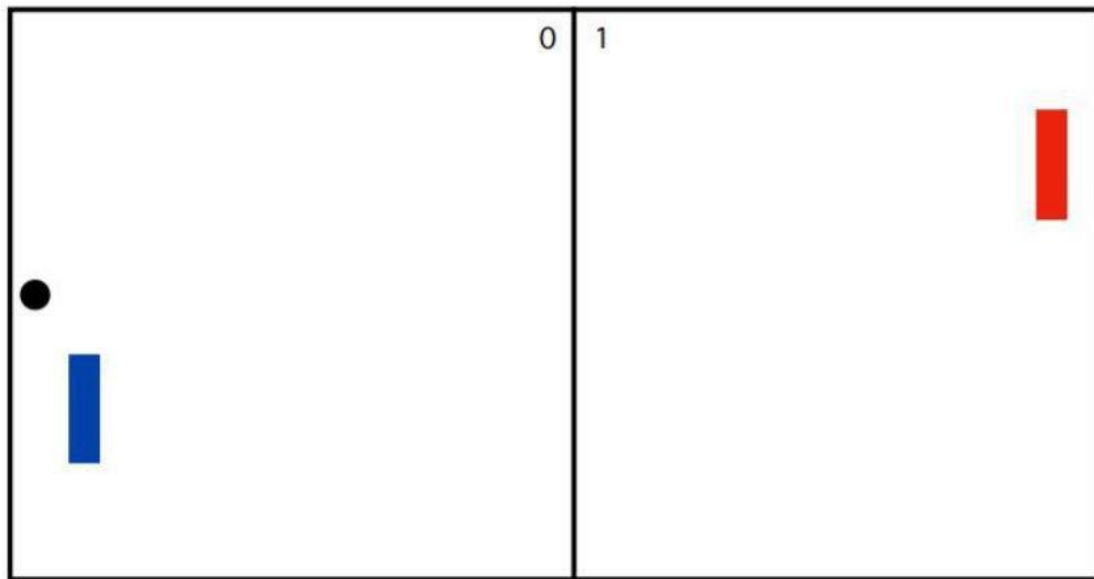
#### วิธีการเล่น

**Player 1** ฝ่ายด้านซ้าย (สีน้ำเงิน) จะใช้ตัวอักษร W บนแป้นคีย์บอร์ด ในการบังคับขึ้นและตัวอักษร S ในการบังคับลง

**Player 2** ฝ่ายด้านขวา (สีแดง) จะใช้ลูกศรขึ้นบนแป้นคีย์บอร์ดเพื่อบังคับขึ้น และลูกศรลงในการบังคับลง

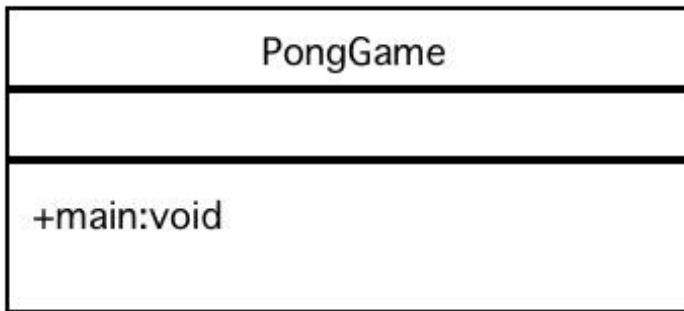


ฝ่าย **Player 1** (สีน้ำเงิน) รับลูกปิงปองพลาสติก **Player2** (สีแดง) จึงได้รับคะแนน

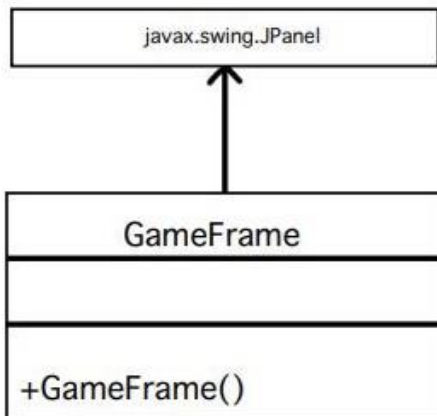


## Class Diagram

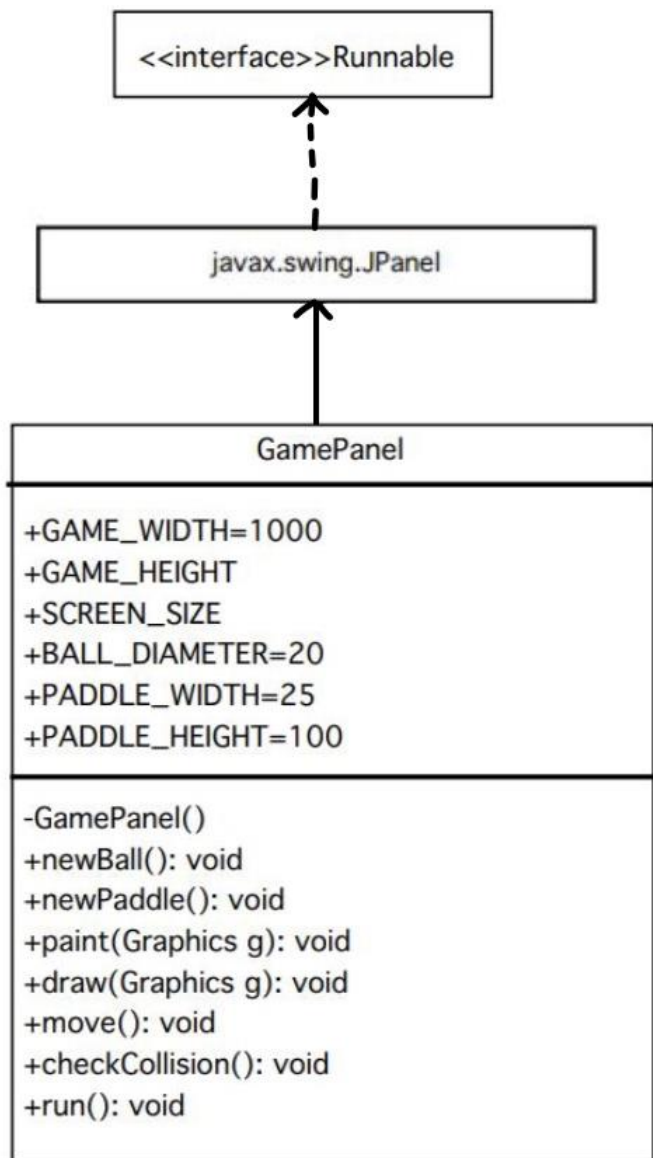
แผนภาพที่แสดง class ทั้งหมดที่ใช้ในการพัฒนา Application



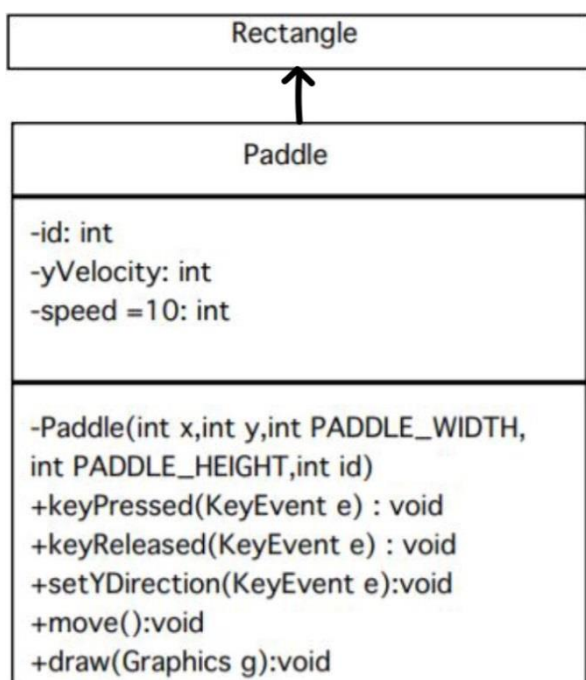
กำหนด Class PongGame เป็นเมนคลาส



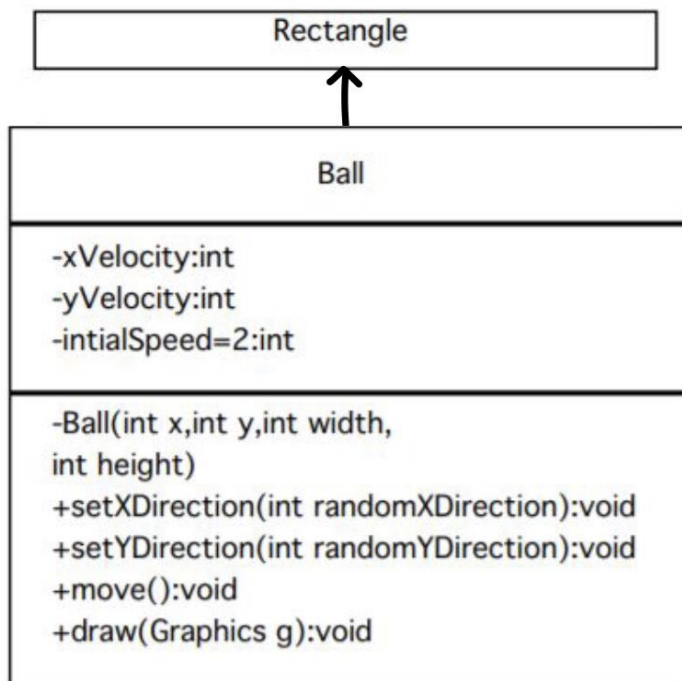
กำหนดให้ class GameFrame extend JFrame เพื่อสร้าง class container และเซตกำหนดตำแหน่งของ components



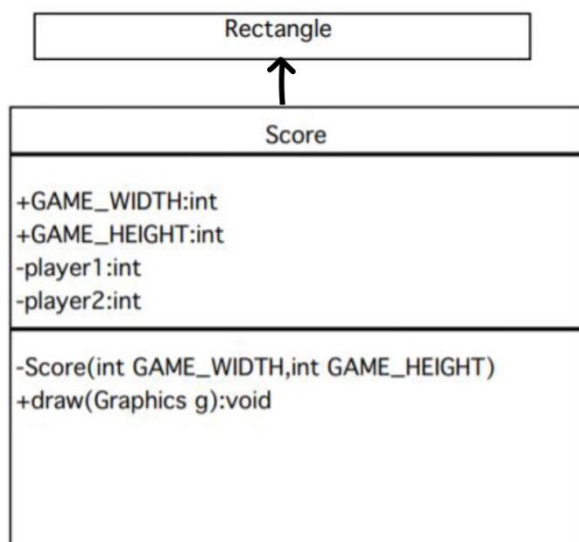
สร้าง class GamePanel ที่ extend JFrame และ implements Runnable เป็นคลาสที่เขียนอัลกอริทึมสำคัญของแอปไว้ เช่น ในส่วนของ ไม้ปิงปอง ลูกปิงปอง คะแนน อัลกอริทึมการเพิ่มความเร็วของลูกปิงปอง



Class Paddle ทำการ extends Rectangle เป็นส่วนของไม้ปิงปอง คลาสนี้เป็น event classes ทั่วไป key Event



Class Ball ทำการ extends Rectangle เป็น  
คลาสที่กำหนด object ของลูกบ๊องปอง กำหนด  
สี ความเร็วและการเคลื่อนที่



Class Score ทำการ extends Rectangle  
คลาสนี้ทำหน้าที่โชว์คะแนนของผู้เล่น

# รูปแบบการพัฒนา Application

## ส่วนของโปรแกรม

### Constructor

คอนสตรัคเตอร์ คือ สิ่งที่มีไว้สำหรับกำหนดค่าเริ่มต้นให้กับตัวแปรคลาสของอ็อบเจ็กต์

```
public class GameFrame extends JFrame{  
    GamePanel panel;  
    GameFrame(){  
        panel = new GamePanel();  
        this.add(panel);  
        this.setTitle("Pong Game");  
        this.setResizable(false);  
        this.setBackground(Color.white);  
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        this.pack();  
        this.setVisible(true);  
        this.setLocationRelativeTo(null);  
    }  
}
```

คลาส **GameFrame** ทำการสืบทอดคุณสมบัติจากคลาส **JFrame** มี **Constructor** **GameFrame()** ซึ่งใน Constructor นี้ทำการ

- นำ complement จากคลาส **GamePanel** ใส่ในคลาส **GameFrame** ที่เป็นหน้าจอหลัก

- เช็ต title ให้ชื่อว่า Pong Game

- กำหนด Resizable เป็น false เพื่อไม่ให้ย่อขยาย

- เช็ต background ให้เป็นสีขาว

- เช็ต locationRelativeTo เป็น null เพื่อให้หน้าจออยู่ตรงกลาง

ทั้งหมดนี้อยู่ในคลาส **GameFrame** จะทำการสร้างหน้าจอขึ้นมา



```

public class GamePanel extends JPanel implements Runnable{

    static final int GAME_WIDTH = 1000;
    static final int GAME_HEIGHT = (int)( GAME_WIDTH* (0.5555));
    static final Dimension SCREEN_SIZE = new Dimension(GAME_WIDTH,GAME_HEIGHT);
    static final int BALL_DIAMETER = 20;
    static final int PADDLE_WIDTH = 25;
    static final int PADDLE_HEIGHT = 100;
    Thread gameThread;
    Image image;
    Graphics graphics;
    Random random;
    Paddle paddle1;
    Paddle paddle2;
    Ball ball;
    Score score;

    GamePanel() {
        newPaddles();
        newBall();
        score = new Score(GAME_WIDTH,GAME_HEIGHT);
        this.setFocusable(true);
        this.addKeyListener(new AL());
        this.setPreferredSize(SCREEN_SIZE);

        gameThread = new Thread(this);
        gameThread.start();
    }
}

```

คลาส **GamePanel** ทำการสืบทอดคุณสมบัติจากคลาส **JPanel** ใช้งานเทรดในรูปแบบ implements จากอินเตอร์เฟส **Runnable** มี **Constructor** **GamePanel()** ซึ่งใน Constructor นี้ทำการ

-สร้าง object score

-เช็ค Focusable เป็น true

-add KeyListener เพื่อรับฟังเหตุการณ์จากคีย์บอร์ด

-สร้าง object thread ระบุเป็น this เพื่อให้มาทำงานที่เมทอด run

```

public class Ball extends Rectangle{

    Random random;
    int xVelocity;
    int yVelocity;
    int initialSpeed = 2;

    Ball(int x, int y, int width, int height){
        super(x,y,width,height);
        random = new Random();
        int randomXDirection = random.nextInt(2);
        if(randomXDirection == 0)
            randomXDirection--;
        setXDirection(randomXDirection*initialSpeed);

        int randomYDirection = random.nextInt(2);
        if(randomYDirection == 0)
            randomYDirection--;
        setYDirection(randomYDirection*initialSpeed);
    }
}

```

คลาส **Ball** ทำการสืบทอดคุณสมบัติจากคลาส **Rectangle** มี **Constructor** **Ball()** ซึ่งใน Constructor นี้มีหน้าที่สร้าง object **Ball** ที่เป็นลูกปิงปองโดยเมื่อเริ่มเกมจะกำหนดทิศทางของลูกบอลโดย random

```

public class Paddle extends Rectangle{

    int id;
    int yVelocity;
    int speed = 10;

    Paddle(int x, int y, int PADDLE_WIDTH, int PADDLE_HEIGHT, int id){
        super(x,y,PADDLE_WIDTH,PADDLE_HEIGHT);
        this.id=id;
    }
}

```

คลาส **Paddle** ทำการสืบทอดคุณสมบัติจากคลาส **Rectangle** มี **Constructor** **Paddle()** ซึ่งใน Constructor นี้มี argument **x,y,PADDLE\_WIDTH,PADDLE\_HEIGHT**

```

public class Score extends Rectangle{

    static int GAME_WIDTH;
    static int GAME_HEIGHT;
    int player1;
    int player2;

    Score(int GAME_WIDTH, int GAME_HEIGHT){
        Score.GAME_WIDTH = GAME_WIDTH;
        Score.GAME_HEIGHT = GAME_HEIGHT;
    }
}

```

คลาส **Score** ทำการสืบทอดคุณสมบัติจากคลาส **Rectangle** มี **Constructor** **Score()** ซึ่งใน **Constructor** นี้มี argument 2 ตัวคือ **int GAME\_WIDTH, int GAME\_HEIGHT** โดยคลาสนี้มีหน้าที่นับคะแนนและส่งคะแนนไปยังหน้าจอของเกมและคอนโซล

## Composition

ใน class object สามารถมีชนิดข้อมูลเป็นชนิด object ได้

```
public class GamePanel extends JPanel implements Runnable{

    static final int GAME_WIDTH= 1000;
    static final int GAME_HEIGHT= (int)( GAME_WIDTH* (0.5555));
    static final Dimension SCREEN_SIZE = new Dimension( GAME_WIDTH, GAME_HEIGHT);
    static final int BALL_DIAMETER= 20;
    static final int PADDLE_WIDTH= 25;
    static final int PADDLE_HEIGHT= 100;
    Thread gameThread;
    Image image;
    Graphics graphics;
    Random random;
    Paddle paddle1;
    Paddle paddle2;
    Ball ball;
    Score score;

    GamePanel(){
        newPaddles();
        newBall();
        score = new Score( GAME_WIDTH, GAME_HEIGHT);
        this.setFocusable(true);
        this.addKeyListener(new AL());
        this.setPreferredSize( SCREEN_SIZE);

        gameThread = new Thread(this);
        gameThread.start();
    }
}
```

ในคลาส GamePanel ที่สืบทอดคุณสมบัติมาจากคลาส JPanel มี static attributes ชนิด int และ attributes ชนิด object ที่ประกอบไปด้วย

-ball จากคลาส Ball

-paddle1 paddle2 จากคลาส Paddle

-score จากคลาส Score

## Polymorphism

คือการที่ออบเจ็กต์สามารถมีได้หลายรูปแบบ ซึ่งเกิดจากการสืบทอดจาก super class และมันยังคงรักษาสภาพและคุณสมบัติของ super class ไว้

```
public class PongGame {  
    public static void main(String[] args) {  
        GameFrame frame = new GameFrame();  
    }  
}
```

ในคลาสเมนทำการ new object frame มีชนิดข้อมูลเป็น GameFrame

```
public class GameFrame extends JFrame{  
    GamePanel panel;  
    GameFrame(){  
        panel = new GamePanel();  
        this.add(panel);  
        this.setTitle("Pong Game");  
        this.setResizable(false);  
        this.setBackground(Color.white);  
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        this.pack();  
        this.setVisible(true);  
        this.setLocationRelativeTo(null);  
    }  
}
```

ซึ่ง GameFrame extend มาจาก JFrame จึงทำให้ frame มีไทม์ของทั้ง GameFrame และ JFrame ไปด้วย

## Inheritance

เป็นคุณสมบัติในการเขียนโปรแกรมเชิงวัตถุที่เรียกว่าคุณสมบัติการสืบทอด โดยที่คลาสสามารถสืบทอดสมาชิกของมันจากคลาสหลัก (superclass) ไปยังคลาทย่อย (subclass)

```
public class GameFrame extends JFrame{  
    JPanel panel;
```

คลาส GameFrame เป็น subclass(คลาสลูก) ทำการสืบทอดคุณสมบัติมาจาก JFrame ที่เป็น superclass

```
public class Paddle extends Rectangle{  
    int id;  
    int yVelocity;  
    int speed = 10;
```

คลาส Paddle เป็น subclass(คลาสลูก) ทำการสืบทอดคุณสมบัติมาจาก Rectangle ที่เป็น superclass

```
public class Ball extends Rectangle{  
    Random random;  
    int xVelocity;  
    int yVelocity;  
    int initialSpeed = 2;
```

คลาส Ball เป็น subclass(คลาสลูก) ทำการสืบทอดคุณสมบัติมาจาก Rectangle ที่เป็น superclass

```
public class Score extends Rectangle{  
    static int GAME_WIDTH;  
    static int GAME_HEIGHT;  
    int player1;  
    int player2;
```

คลาส Score เป็น subclass(คลาสลูก) ทำการสืบทอดคุณสมบัติมาจาก Rectangle ที่เป็น superclass

```
public class GamePanel extends JPanel implements Runnable{  
    static final int GAME_WIDTH = 1000;  
    static final int GAME_HEIGHT = (int)(GAME_WIDTH * (0.5555));  
    static final Dimension SCREEN_SIZE = new Dimension(GAME_WIDTH, GAME_HEIGHT);  
    static final int BALL_DIAMETER = 20;  
    static final int PADDLE_WIDTH = 25;  
    static final int PADDLE_HEIGHT = 100;  
    Thread gameThread;
```

คลาส GamePanel เป็น subclass(คลาสลูก) ทำการสืบทอดคุณสมบัติมาจาก Rectangle ที่เป็น superclass

# GUI

(Graphical User Interface)

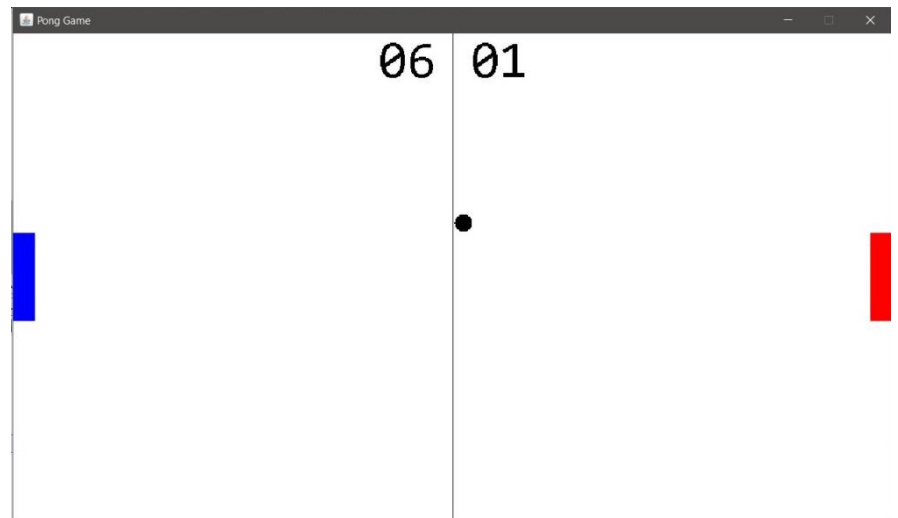
ส่วนประกอบของ GUI

-Container

-Window

-JFrame

-JPanel



สร้าง Container Components จากคลาส JFrame ที่สืบทอดคุณสมบัติมาจาก  
คลาส **JFrame** ทำการตั้งชื่อ title ว่า Pong Game กำหนดให้สี Components background เป็นสี  
ขาวและทำการ new object panel จากคลาส JPanel

```
public class GameFrame extends JFrame{  
    JPanel panel;  
    GameFrame(){  
        panel = new JPanel();  
        this.add(panel);  
        this.setTitle("Pong Game");  
        this.setResizable(false);  
        this.setBackground(Color.white);  
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        this.pack();  
        this.setVisible(true);  
        this.setLocationRelativeTo(null);  
    }  
}
```



คลาส `GamePanel` สืบทอดคุณสมบัติมาจากคลาส `JPanel` ใช้ `focusable` เป็น `true` ทำการ `new` object `score` จาก `Score` เพื่อแสดงคะแนนที่หน้าจอ มีอินเทอร์เฟซประเภท `Listener` คือ `KeyListener` เป็นอินเทอร์เฟซสำหรับ object ของคลาส `KeyEvent`

```
public class GamePanel extends JPanel implements Runnable{

    static final int GAME_WIDTH = 1000;
    static final int GAME_HEIGHT = (int)(GAME_WIDTH * 0.5555);
    static final Dimension SCREEN_SIZE = new Dimension(GAME_WIDTH, GAME_HEIGHT);
    static final int BALL_DIAMETER = 20;
    static final int PADDLE_WIDTH = 25;
    static final int PADDLE_HEIGHT = 100;
    Thread gameThread;
    Image image;
    Graphics graphics;
    Random random;
    Paddle paddle1;
    Paddle paddle2;
    Ball ball;
    Score score;

    GamePanel(){
        newPaddles();
        newBall();
        score = new Score(GAME_WIDTH, GAME_HEIGHT);
        this.setFocusable(true);
        this.addKeyListener(new AL());
        this.setPreferredSize(SCREEN_SIZE);

        gameThread = new Thread(this);
        gameThread.start();
    }
}
```

## Event handling

```
public class Paddle extends Rectangle{

    int id;
    int yVelocity;
    int speed = 10;

    Paddle(int x, int y, int PADDLE_WIDTH, int PADDLE_HEIGHT, int id){
        super(x,y,PADDLE_WIDTH,PADDLE_HEIGHT);
        this.id=id;
    }

    public void keyPressed(KeyEvent e) {
        switch(id) {
            case 1:
                if(e.getKeyCode()==KeyEvent.VK_W) {
                    setYDirection(-speed);
                }
                if(e.getKeyCode()==KeyEvent.VK_S) {
                    setYDirection(speed);
                }
                break;
            case 2:
                if(e.getKeyCode()==KeyEvent.VK_UP) {
                    setYDirection(-speed);
                }
                if(e.getKeyCode()==KeyEvent.VK_DOWN) {
                    setYDirection(speed);
                }
                break;
        }
    }

    public void keyReleased(KeyEvent e) {
        switch(id) {
            case 1:
                if(e.getKeyCode()==KeyEvent.VK_W) {
                    setYDirection(0);
                }
                if(e.getKeyCode()==KeyEvent.VK_S) {
                    setYDirection(0);
                }
                break;
        }
    }

    public void keyReleased(KeyEvent e) {
        switch(id) {
            case 1:
                if(e.getKeyCode()==KeyEvent.VK_W) {
                    setYDirection(0);
                }
                if(e.getKeyCode()==KeyEvent.VK_S) {
                    setYDirection(0);
                }
                break;
            case 2:
                if(e.getKeyCode()==KeyEvent.VK_UP) {
                    setYDirection(0);
                }
                if(e.getKeyCode()==KeyEvent.VK_DOWN) {
                    setYDirection(0);
                }
                break;
        }
    }

    public void setYDirection(int yDirection) {
        yVelocity = yDirection;
    }

    public void move() {
        y = y + yVelocity;
    }

    public void draw(Graphics g) {
        if(id==1)
            g.setColor(Color.blue);
        else
            g.setColor(Color.red);
        g.fillRect(x, y, width, height);
    }
}
```

Paddle class ที่สืบทอดคุณสมบัติมาจาก Rectangle ทำหน้าที่สร้างในส่วนของไม้ปิงปองของทั้งสองผู้เล่น กำหนดสี/ความยาว/ความกว้าง/ความเร็ว ของไม้ปิงปอง ตั้งค่าให้ไม้ปิงปองเคลื่อนที่ในแนวแกน Y

นอกจากนี้ คลาส **Paddle** ยังเป็นส่วนรับฟังเหตุการณ์จากการทำงานของคีย์บอร์ด โดยมี method คือ

```
public void keyPressed(KeyEvent e) {  
    switch(id) {  
        case 1:  
            if(e.getKeyCode()==KeyEvent.VK_W) {  
                setYDirection(-speed);  
            }  
            if(e.getKeyCode()==KeyEvent.VK_S) {  
                setYDirection(speed);  
            }  
            break;  
        case 2:  
            if(e.getKeyCode()==KeyEvent.VK_UP) {  
                setYDirection(-speed);  
            }  
            if(e.getKeyCode()==KeyEvent.VK_DOWN) {  
                setYDirection(speed);  
            }  
            break;  
    }  
}
```

method keyPressed จะทำงานเมื่อ key ถูกกด

```
public void keyReleased(KeyEvent e) {  
    switch(id) {  
        case 1:  
            if(e.getKeyCode()==KeyEvent.VK_W) {  
                setYDirection(0);  
            }  
            if(e.getKeyCode()==KeyEvent.VK_S) {  
                setYDirection(0);  
            }  
            break;  
        case 2:  
            if(e.getKeyCode()==KeyEvent.VK_UP) {  
                setYDirection(0);  
            }  
            if(e.getKeyCode()==KeyEvent.VK_DOWN) {  
                setYDirection(0);  
            }  
            break;  
    }  
}
```

method keyReleased จะทำงานเมื่อ key ถูกปล่อยจากการกด

# อัลกอริทึมที่สำคัญในโปรแกรม

## Paddle หรือ ไม้ปิงปอง

```
public class GamePanel extends JPanel implements Runnable{

    static final int GAME_WIDTH = 1000;
    static final int GAME_HEIGHT = (int)(GAME_WIDTH * 0.5555);
    static final Dimension SCREEN_SIZE = new Dimension(GAME_WIDTH, GAME_HEIGHT);
    static final int BALL_DIAMETER = 20;
    static final int PADDLE_WIDTH = 25;
    static final int PADDLE_HEIGHT = 100;
    Thread gameThread;
    Image image;
    Graphics graphics;
    Random random;
    Paddle paddle1;
    Paddle paddle2;
    Ball ball;
    Score score;

    GamePanel(){
        newPaddles();
        newBall();
        score = new Score(GAME_WIDTH, GAME_HEIGHT);
        this.setFocusable(true);
        this.addKeyListener(new AL());
        this.setPreferredSize(SCREEN_SIZE);

        gameThread = new Thread(this);
        gameThread.start();
    }
}
```

จากคลาส GamePanel ได้กำหนด

static final int PADDLE\_WIDTH = 25; คือความกว้างของไม้ปิงปอง

static final int PADDLE\_HEIGHT = 100; คือความยาวของไม้ปิงปอง

และใน constructor ได้ทำการเรียกใช้ newPaddles();

```
public void newPaddles() {
    paddle1 = new Paddle(0, (GAME_HEIGHT/2)-(PADDLE_HEIGHT/2), PADDLE_WIDTH, PADDLE_HEIGHT, 1);
    paddle2 = new Paddle(GAME_WIDTH-PADDLE_WIDTH, (GAME_HEIGHT/2)-(PADDLE_HEIGHT/2), PADDLE_WIDTH, PADDLE_HEIGHT, 2);
}
```

ซึ่งเมธอด newPaddles จะทำการสร้างและเซตตำแหน่งไม้ปิงปองของ player 1, player 2

```

public class Paddle extends Rectangle{

    int id;
    int yVelocity; //ความเร็วของไม้ปิงปองที่ยับขึ้นลงเมื่อผู้เล่นกดปุ่ม
    int speed = 10;

    Paddle(int x, int y, int PADDLE_WIDTH, int PADDLE_HEIGHT, int id){
        super(x,y,PADDLE_WIDTH,PADDLE_HEIGHT);
        this.id=id;
    }
}

```

คลาส Paddle ทำการสืบทอดคุณสมบัติจาก Rectangle โดยมี constructor ใช้เรียก super constructor รับ argument x,y,PADDLE\_WIDTH,PADDLE\_HEIGHT

```

public void draw(Graphics g) {
    if(id==1)
        g.setColor(Color.blue);
    else
        g.setColor(Color.red);
    g.fillRect(x, y, width, height);
}

```

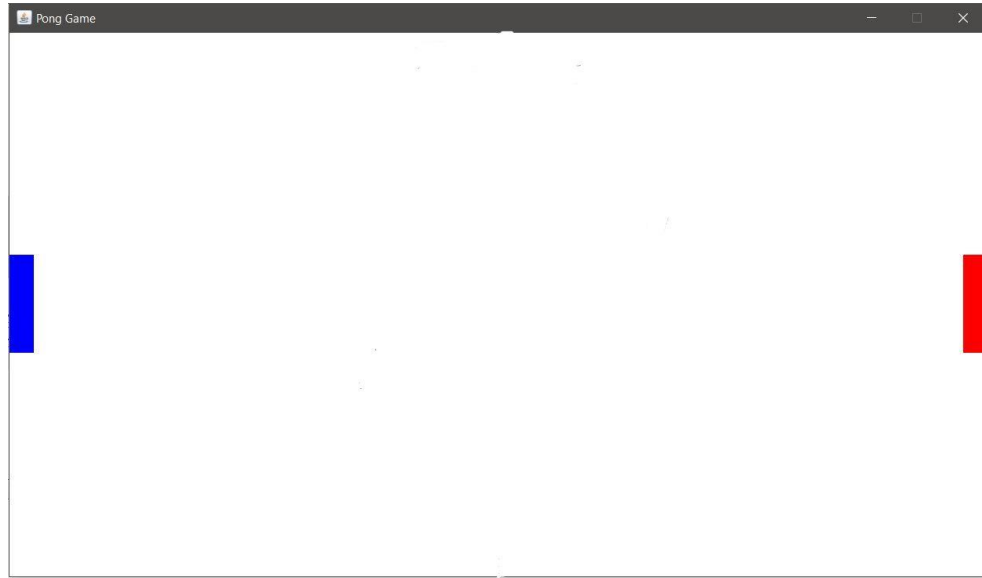
ในคลาส Paddle กำหนดสีให้กับไม้ปิงปองโดยกำหนดให้ id = 1 จะเป็นสีน้ำเงิน ถ้าไม่เท่ากับ 1 ไม้ปิงปองอีกอันจะเป็นสีแดง

```

public void draw(Graphics g) {
    paddle1.draw(g);
    paddle2.draw(g);
    ball.draw(g);
    score.draw(g);
    Toolkit.getDefaultToolkit().sync();
}

```

สร้างเมธอด draw (Graphics g) ในคลาส GamePanel เรียกใช้ paddle1,paddle2 เป็นการวาดไม้ปิงปองของผู้เล่นลงไปหน้าจอ



เมื่อทำการ run แล้วจะได้หน้าจอที่มีไม้ปิงปองของ player 1 เป็นสีน้ำเงิน player 2 เป็นสีแดง

เพื่อที่จะทำให้ไม้ปิงปองขยับขึ้นลงได้จึงต้องสร้าง AL class สำหรับ Action listener โดยคลาสนี้ทำการ extend Keyadapter

สร้างเมธอด keypress และ keyReleased เพื่อทำการเรียกใช้ฟังก์ชัน

```
public class AL extends KeyAdapter { //คลาสประเภท event adapter
    public void keyPressed(KeyEvent e) {
        paddle1.keyPressed(e);
        paddle2.keyPressed(e);
    }
    public void keyReleased(KeyEvent e) {
        paddle1.keyReleased(e);
        paddle2.keyReleased(e);
    }
}
```



```

public void keyPressed(KeyEvent e) {
    switch(id) {
        case 1:
            if(e.getKeyCode()==KeyEvent.VK_W) {
                setYDirection(-speed);
            }
            if(e.getKeyCode()==KeyEvent.VK_S) {
                setYDirection(speed);
            }
            break;
        case 2:
            if(e.getKeyCode()==KeyEvent.VK_UP) {
                setYDirection(-speed);
            }
            if(e.getKeyCode()==KeyEvent.VK_DOWN) {
                setYDirection(speed);
            }
            break;
    }
}

public void keyReleased(KeyEvent e) {
    switch(id) {
        case 1:
            if(e.getKeyCode()==KeyEvent.VK_W) {
                setYDirection(0);
            }
            if(e.getKeyCode()==KeyEvent.VK_S) {
                setYDirection(0);
            }
            break;
        case 2:
            if(e.getKeyCode()==KeyEvent.VK_UP) {
                setYDirection(0);
            }
            if(e.getKeyCode()==KeyEvent.VK_DOWN) {
                setYDirection(0);
            }
            break;
    }
}

```

ในคลาส Paddle จะทำการสร้างเมธอด  
**keyPressed** เพื่อทำการกำหนดปุ่ม โดย

case 1 เป็นของ player 1 เมื่อกดคีย์ W  
ไม้ปิงปองจะขยับขึ้น เมื่อกดคีย์ S ไม้  
ปิงปองจะขยับลง

case 2 เป็นของ player 2 เมื่อกดคีย์ลูกศร  
ขึ้น ไม้ปิงปองจะขยับขึ้น เมื่อกดคีย์ลูกศร  
ลง ไม้ปิงปองจะขยับลง

**KeyReleased** จะทำงานเมื่อ key ถูก  
ปล่อยจากการกด

```

public void setYDirection(int yDirection) {
    yVelocity = yDirection;
}

public void move() {
    y = y + yVelocity;
}

```

สร้างเมธอด setYDirection กำหนดความเร็วของไม้ปิงปอง  
สร้างเมธอด move ทำให้ไม้ปิงปองขยับ

เมื่อทำการ run ไม้ปิงปองทั้งสองด้านจะขยับได้แต่ขยับเกินขอบหน้าจอจึงต้องเพิ่มเมธอดในการ  
กำหนดเขตการขยับของไม้ปิงปองให้อยู่ในหน้าจอ

```

public void checkCollision() {
    if(paddle1.y<=0)
        paddle1.y=0;
    if(paddle1.y >= (GAME_HEIGHT-PADDLE_HEIGHT))
        paddle1.y = GAME_HEIGHT-PADDLE_HEIGHT;
    if(paddle2.y<=0)
        paddle2.y=0;
    if(paddle2.y >= (GAME_HEIGHT-PADDLE_HEIGHT))
        paddle2.y = GAME_HEIGHT-PADDLE_HEIGHT;
}

```

ทำการสร้างเมธอด checkCollision ในคลาส  
GamePanel เพื่อกำหนดให้ไม้ปิงปองขยับอยู่ในเขต  
จอ โดยให้ paddle1 = player 1 paddle 2 = player2

## Ball หรือ ลูกปิงปอง

```
public class GamePanel extends JPanel implements Runnable{

    static final int GAME_WIDTH = 1000; //ความกว้างของโต๊ะ
    static final int GAME_HEIGHT = (int)(GAME_WIDTH * (0.5555)); //ความยาวโต๊ะ
    static final Dimension SCREEN_SIZE = new Dimension(GAME_WIDTH, GAME_HEIGHT); //จุดศูนย์กลางของโต๊ะ
    static final int BALL_DIAMETER = 20; //จุดศูนย์กลางของลูกปิงปองกำหนดให้เป็น 20
    static final int PADDLE_WIDTH = 25; //ความกว้างของไม้ปิงปอง
    static final int PADDLE_HEIGHT = 100; //ความยาวของไม้ปิงปอง
    Thread gameThread; //กำหนดเธรด
    Image image;
    Graphics graphics;
    Random random;
    Paddle paddle1; //ไม้ปิงปองของผู้เล่นคนที่ 1
    Paddle paddle2; //ไม้ปิงปองของผู้เล่นคนที่ 2
    Ball ball; //ลูกปิงปอง
    Score score; //คะแนน

    GamePanel(){
        newPaddles();
        newBall();
        score = new Score(GAME_WIDTH, GAME_HEIGHT);
        this.setFocusable(true); //set focus ถ้าคลิกจะโฟกัสและอ่าน
        this.addKeyListener(new AL()); //เพิ่ม action listener
        this.setPreferredSize(SCREEN_SIZE);

        gameThread = new Thread(this); //สร้างเธรดตรงนี้จะทำงานอยู่ตลอดเวลา
        gameThread.start();
    }

    public void newBall() {
        random = new Random();
        ball = new Ball((GAME_WIDTH/2)-(BALL_DIAMETER/2), random.nextInt(GAME_HEIGHT-BALL_DIAMETER), BALL_DIAMETER, BALL_DIAMETER);
    }
}
```

จากคลาส GamePanel ได้สร้างเมธอด newBall ที่มีการ new object random และ new object ball ขึ้นมา และมี argument เพื่อส่งไปยังคลาส Ball

```
public class Ball extends Rectangle{

    Random random;
    int xVelocity; //การขยับของลูกปิงปองในแนวนอนx
    int yVelocity; //การขยับของลูกปิงปองในแนวแกนy
    int initialSpeed = 2;

    Ball(int x, int y, int width, int height){
        super(x,y,width,height);
        random = new Random();
        int randomXDirection = random.nextInt(2);
        if(randomXDirection == 0)
            randomXDirection--;
        setXDirection(randomXDirection*initialSpeed);

        int randomYDirection = random.nextInt(2);
        if(randomYDirection == 0)
            randomYDirection--;
        setYDirection(randomYDirection*initialSpeed);
    }
}
```

คลาส Ball ทำการสร้าง constructor ที่มีพารามิเตอร์ int x, int y, int width, int height และทำการเรียก superclass constructor หลังจากนั้นทำการ new object random ซึ่งเป็น local variable

int randomXDirection = random.nextInt(2)

ถ้า randomXDirection = 0 ลูกปิงปองจะไปซ้ายกับแกน X

randomXDirection = 1 ลูกปิงปองจะไปขวากับแกน X

ในส่วนของแกน Y เขียนคำสั่งเหมือนกันกับแกน X

```
public void setXDirection(int randomXDirection) {
    xVelocity = randomXDirection;
}
public void setYDirection(int randomYDirection) {
    yVelocity = randomYDirection;
}
public void move() {
    x += xVelocity;
    y += yVelocity;
}
public void draw(Graphics g) {
    g.setColor(Color.black);
    g.fillOval(x, y, height, width);
}
```

คลาส Ball ทำการสร้างเมธอด setXDirection กำหนดให้ขยับตามแกน X

ทำการสร้างเมธอด setYDirection กำหนดให้ขยับตามแกน Y

สร้างเมธอด move ให้ลูกปิงปองขยับได้

สร้างเมธอด draw กำหนด ขนาดลูกปิงปองและสี



```

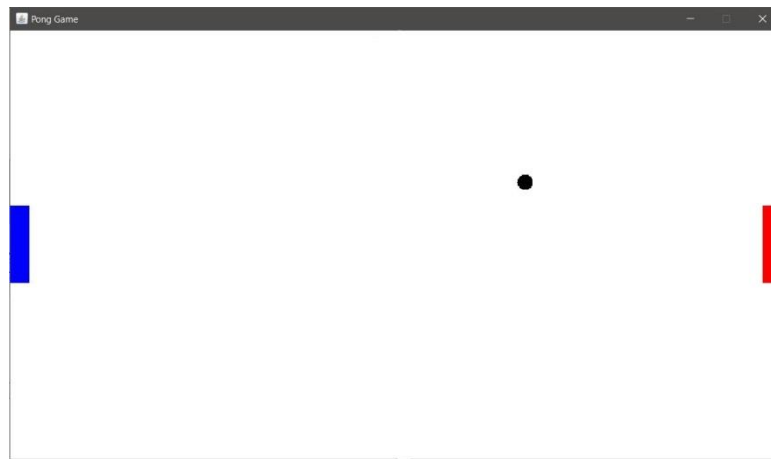
public void draw(Graphics g) {
    paddle1.draw(g);
    paddle2.draw(g);
    ball.draw(g);
    score.draw(g);
    Toolkit.getDefaultToolkit().sync();
}

public void move() {
    paddle1.move();
    paddle2.move();
    ball.move();
}

```

ในคลาส **GamePanel** ทำการเขียนคำสั่ง  
เรียกใช้ **draw** และ **move** จากคลาส **Ball**

เมื่อทำการ run ลูกปิงปองจะขยับได้แต่ขยับเกินขอบหน้าจอจึงต้องเพิ่มเมธอดในการกำหนดเขต  
การขยับของลูกปิงปองให้อยู่ในหน้าจอ



```

public void checkCollision() {
    if(ball.y <= 0) {
        ball.setYDirection(-ball.yVelocity);
    }
    if(ball.y >= GAME_HEIGHT-BALL_DIAMETER) {
        ball.setYDirection(-ball.yVelocity);
    }
}

```

จากเมธอด **checkCollision** ในคลาส **GamePanel** เพื่อ  
กำหนดให้ลูกปิงปองขยับอยู่ในเขตจอโดยถ้า

$ball.y \leq 0$  ให้ `ball.setYDirection(-ball.yVelocity)`

$ball.y \geq GAME\_HEIGHT-BALL\_DIAMETER$  ให้  
`ball.setYDirection(-ball.yVelocity)`

ลูกปิงปองที่เค้งโดนขอบจอจะกระเด็นไปยังทิศตรงข้าม

```

public void checkCollision() {
    if(ball.intersects(paddle1)) {
        ball.xVelocity = Math.abs(ball.xVelocity);
        ball.xVelocity++;
        if(ball.yVelocity>0)
            ball.yVelocity++;
        else
            ball.yVelocity--;
        ball.setXDirection(ball.xVelocity);
        ball.setYDirection(ball.yVelocity);
    }
    if(ball.intersects(paddle2)) {
        ball.xVelocity = Math.abs(ball.xVelocity);
        ball.xVelocity++;
        if(ball.yVelocity>0)
            ball.yVelocity++;
        else
            ball.yVelocity--;
        ball.setXDirection(-ball.xVelocity);
        ball.setYDirection(ball.yVelocity);
    }
}

```

จากเมธอด checkCollision ในคลาส GamePanel เพื่อ  
กำหนดให้ลูกปิงปองเด้งออกหลังจากถูกไม้ปิงปองตีจะทำ

การแยกเป็นสองเคส โดย case 1 = player 1 , Case 2 =  
player 2

ถ้า  $ball.yVelocity > 0$  จะทำให้  $ball.yVelocity$  เพิ่มขึ้น 1

แต่หากไม่ตรงตามเงื่อนไข  $ball.yVelocity$  จะลบ 1

อธิบายได้ดังนี้ เมื่อไม้ปิงปองตีโดนลูกปิงปอง ลูกปิงปอง  
จะทำการเพิ่มความเร็วขึ้นจึงทำให้ยังดีนนานเท่าใดลูก  
ปิงปองจะเร็วขึ้นเท่านั้น

## บทที่ 3 สรุป

### ปัญหาที่พบระหว่างการพัฒนา

บางเรื่องที่ต้องใช้ในการพัฒนาเกมยากจนเกินความสามารถที่มีซึ่งต้องใช้เวลาในการฝึกฝนเมื่อนำความสามารถที่มีมาพัฒนาเกมจึงทำให้ติดขัดไปบ้างและเวลาในการพัฒนาเกมค่อนข้างกระชั้นชิดกับการสอบ Final จึงทำให้ประสิทธิภาพของเกมไม่ดีเท่าที่ได้ตั้งเป้าหมายเอาไว้

### จุดเด่นของโปรแกรม

เนื่องจากพัฒนาเกมโดยการใช้ keyEvent จากคีย์บอร์ดแบ่งเป็นสองฝั่งจึงทำให้หนึ่งเกมเล่นได้ถึงสองคนเพิ่มความสนุกสนานและความเพลิดเพลิน