



Mini Project

เกม : FLYING KOALA

โดย

นาย กรณ์มงคล ฤสิทธิ์พิชัย

รหัสนักศึกษา 6404062610391 ตอนเรียนที่ 1

รายงานนี้เป็นส่วนหนึ่งของวิชา 040613222

Object Oriented Programming

ภาควิชาวิทยาการคอมพิวเตอร์และสารสนเทศ

คณะวิทยาศาสตร์ประยุกต์

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

ภาคการศึกษาที่ 1 ปีการศึกษา 2565

สารบัญ

บทที่ 1 : บทนำ	1
ที่มาและความสำคัญ	1
ประเภทของโครงการ	1
บทที่ 2 : ส่วนการพัฒนา	4
วิธีเล่น	4
Class Diagrams	5
รูปแบบการพัฒนา	6
หน้าจอ GUI และ Event handling	9
อัลกอริทึมที่สำคัญ	15
บทที่ 3 : สรุป	16
ปัญหาที่พบระหว่างการพัฒนา	16
จุดเด่นของโปรแกรมที่ไม่เหมือนใคร	16

บทที่ 1 : บทนำ

ที่มาและความสำคัญ

โปรแกรมที่ผู้เขียนได้จัดทำมีแรงบันดาลใจมาจากเกมในโปรแกรม Google Chrome ที่จะแสดงให้เห็นผู้ใช้งานขณะที่ไม่มีการเข้าถึงของอินเทอร์เน็ต ผู้เขียนจึงมีความคิดที่จะจำลองการเขียนโปรแกรมที่มีลักษณะคล้ายกับเกมดังกล่าวขึ้นด้วยการใช้ภาษาจาวาเพื่อพัฒนาโปรแกรมขึ้นมาเพื่อให้เกิดความสนุกและเพลิดเพลิน

ประเภทของโครงการ

เกมประเภท Arcade เล่นคนเดียวโดยไม่ใช้อินเทอร์เน็ต

ประโยชน์

1. ฝึกไหวพริบในการมองเห็น
2. เพื่อความสนุกสนาน
3. เพื่อฝึกกระบวนการคิดวิเคราะห์สถานการณ์ต่าง ๆ

ขอบเขต

เกมโคอาล่าบินๆ

(Flying Koala)

*****หมายเหตุ รูปใช้เพื่อการโฆษณาตอนทำอาจมีการเปลี่ยนแปลงเล็กน้อยตามความสามารถของผู้ทำ**

● รายละเอียดเกม

เกมโคอาล่าบินๆ เป็นเกมที่จะให้ผู้เล่นจำลองตัวเองเป็นหมีโคอาล่าที่กำลังหลงอยู่ในป่า ซึ่งหมี

โคอาล่าตัวนี้ต้องหลบหลีกสัตว์ป่าดุร้ายที่จะเข้ามาทำร้ายมัน ไม่ว่าจะเป็นสัตว์ร้ายตัวไหนหมีโคอาล่าก็จะ

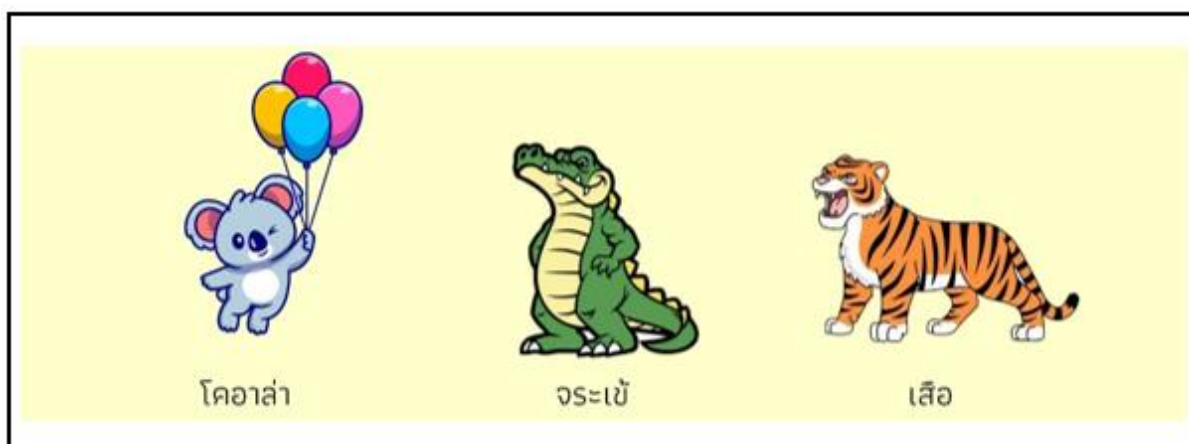
ต้องหลบให้ได้ หากหลบไม่ได้จะ Game Over ทันที

● วิธีเล่น

ใช้ลูกศรบนแป้นคีย์บอร์ดเพื่อควบคุมเจ้าหมีโคอาล่าให้หลบหลีกสัตว์ร้าย

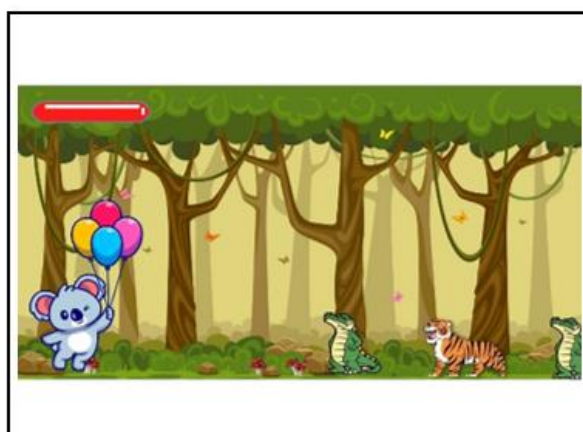
● เนื้อหาของเกม

- ตัวละคร



***หมายเหตุ ตัวละครต่าง ๆ อาจมีการเปลี่ยนแปลง

- ฉาก
 - เริ่มเกม



- เมื่อตัวละครชนกับสิ่งกีดขวาง หรือ ชนกับศัตรูจะ จบเกม ทันทีพร้อมสรุปคะแนน



***หมายเหตุ ฉากและตัวเกมจะมีการเปลี่ยนแปลงอย่างเน่ำนอน ภาพข้างต้นเป็นเพียงภาพประกอบ

- ประโยชน์

1. ฝึกไหวพริบในการมองเห็น
2. เพื่อความสนุกสนาน
3. เพื่อฝึกกระบวนการคิดโรสถานการณ์ต่าง ๆ

- แผนการทำงาน

ลำดับ	รายการ	สัปดาห์ที่ 1	สัปดาห์ที่ 2	สัปดาห์ที่ 3	สัปดาห์ที่ 4
1	ทำกราฟิกภายในเกมและตัวละคร	←→			
2	ศึกษาเอกสารและข้อมูลที่เกี่ยวข้อง	←→	→		
3	ลงมือเขียนโปรแกรม		←→	→	
4	ตรวจสอบและแก้ไขข้อผิดพลาด			←→	→
5	จัดทำเอกสาร				←→

บทที่ 2 : ส่วนการพัฒนา

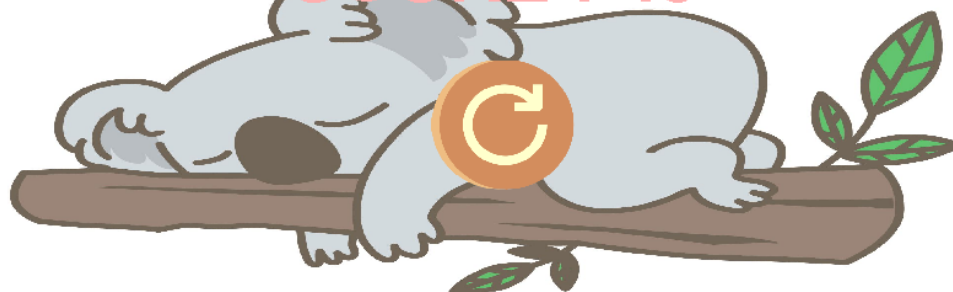
วิธีเล่น

ใช้ลูกศรบนแป้นพิมพ์เพื่อกำหนดให้ตัวละครกระโดดหรือหมอบลงเพื่อหลบสิ่งกีดขวางและศัตรูทำให้ตัวละครมีชีวิตรอยู่ได้นานที่สุด (ลูกศรขึ้นและลงเพื่อกระโดดและหมอบตามลำดับ)

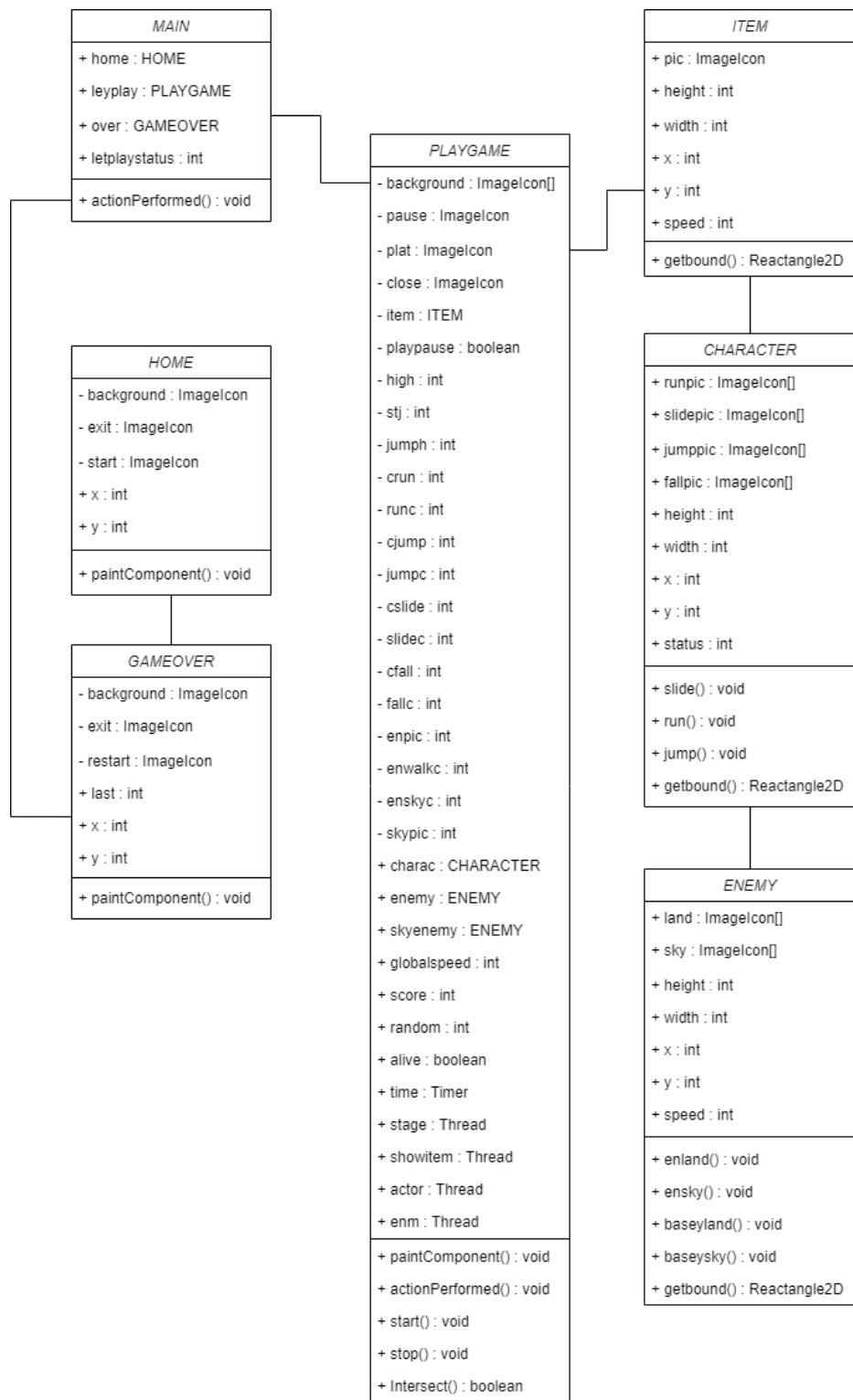


GAME OVER

SCORE : 43



Class Diagrams



Class MAIN extends JFrame implements ActionListener

Class PLAYGAME extends JPanel implements ActionListener

Class HOME & GAMEOVER extends JPanel

รูปแบบการพัฒนา

เกมเป็นภาษา java แบบ java application

Constructor

ตัวอย่าง

```
MAIN(){

    ImageIcon icon = new ImageIcon(this.getClass().getResource("Face 01.png"));

    this.setIconImage(icon.getImage());

    add(home);

    time.start();

}
```

Encapsulation

ตัวอย่าง

```
private boolean playpause = true;

public CHARACTER charac = new CHARACTER();

public ENEMY enemy = new ENEMY();

public ENEMY skyenemy = new ENEMY();

public ENEMY spaenemy = new ENEMY();

private ITEM item = new ITEM();
```

Composition

ตัวอย่าง

```

    if (letplay.alive == false) {

        over.last = letplay.score;

        this.remove(letplay);

        this.add(over);

        over.requestFocusInWindow();

    }

```

Polymorphism

ตัวอย่าง

```

public void paintComponent(Graphics g){

    super.paintComponent(g);

    g.drawImage(background.getImage(),0,0,1000,600,this);

    g.drawImage(restart.getImage(),425,225,150,150,this);

    g.drawImage(exit.getImage(),850,450, 90,90 ,this);

    g.setFont(new Font("2005_iannnnnTKO",Font.CENTER_BASELINE,100));

    g.setColor(Color.red);

    g.drawString("GAME OVER",300,100);

    g.setColor(Color.PINK);

    g.setFont(new Font("2005_iannnnnTKO",Font.CENTER_BASELINE,60));

    g.drawString("SCORE : "+last,340,200);

}

```

Abstract

ตัวอย่าง

```
abstract class GraphicObject{

    abstract void run();

    abstract void jump();

    abstract void slide();

    abstract Rectangle2D getbound();

}

public class CHARACTER extends GraphicObject {

    public void run(){

    }

    public void jump(){

    }

    public void slide(){

    }

    public Rectangle2D getbound(){

    }

}
```

Inheritance

ตัวอย่าง

```
public class MAIN extends JFrame

public class GAMEOVER extends JPanel
```

หน้าจอ GUI และ Event handling

หน้าจอเริ่มต้นเมื่อกดเริ่มตรงกลางจะเริ่มเกมและเมื่อกดปิดจะปิดเกม โดยมี MouseListener เป็นตัวดักฟังการกระทำต่างๆในหน้าจอ



Code ของการดักฟังในหน้าจอเริ่มแรก

```
if (home.x >= 425 && home.x <= 575 && home.y >= 225 && home.y <= 375){

    home.x = 0;

    home.y = 0;

    this.remove(home);

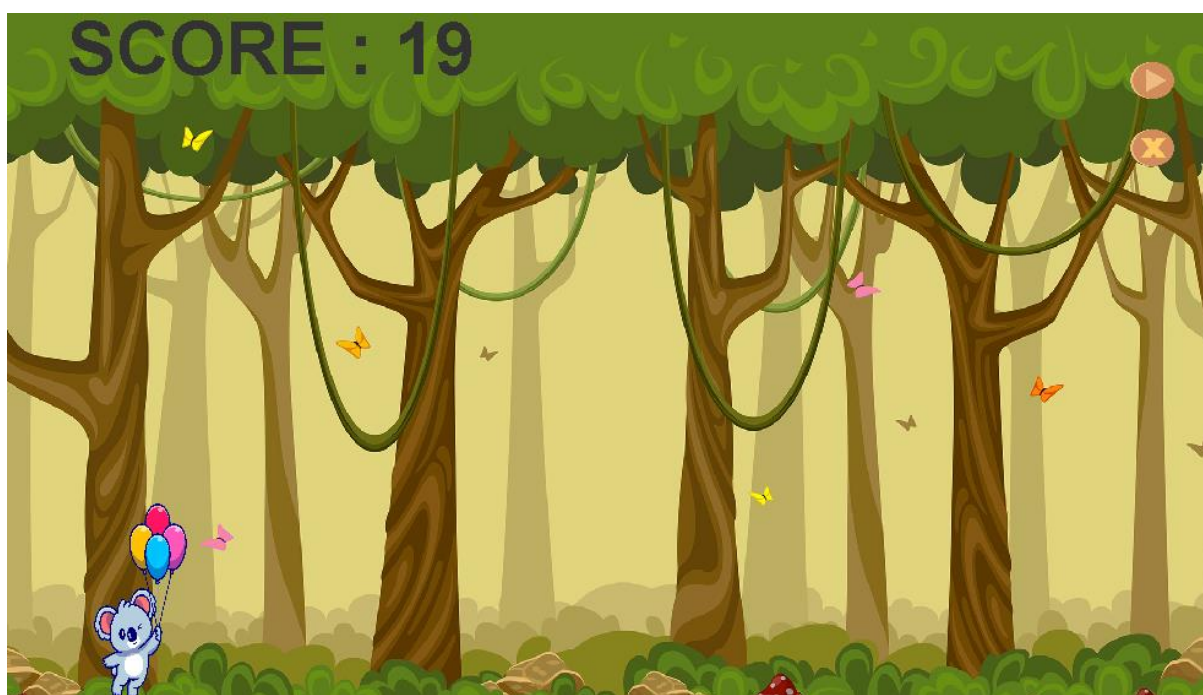
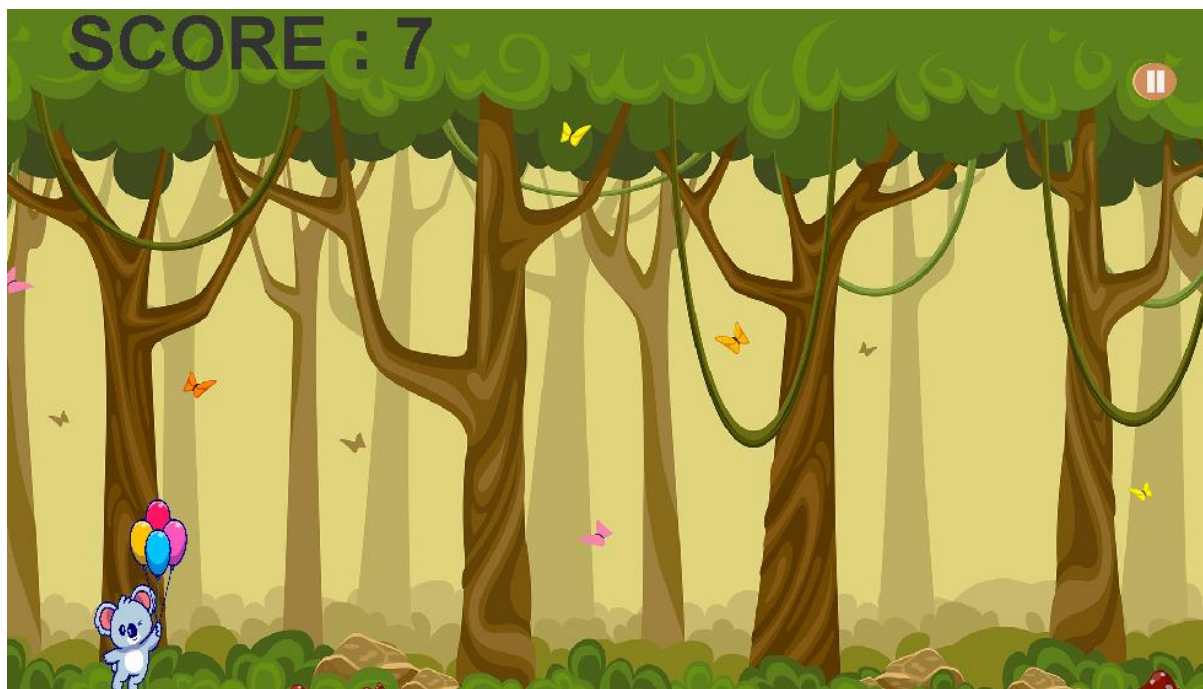
    this.add(letplay);

    letplay.requestFocusInWindow();

    letplay.start();

}
```

หน้าจอเมื่อกดหยุดเกมจะหยุดและสร้างปุ่มที่สามารถกดเพื่อปิดได้ โดยมี MouseListener เป็นตัวดักฟังการกระทำต่างๆในหน้าจอ และยังมี KeyListener ที่ควบคุมปุ่มตัวละครให้กระโดดและหมอบโดยลูกศรขึ้นลง



Code ของ MouseListener และ KeyListener ที่ดักฟังในหน้านี้

```
this.addKeyListener(new KeyAdapter() {

    @Override

    public void keyPressed(KeyEvent e) {

        super.keyPressed(e);

        if (e.getKeyCode() == KeyEvent.VK_UP) {

            if (charac.status == 0) {

                charac.jump();

                stj = 1;

                repaint();

            }

        } else if (e.getKeyCode() == KeyEvent.VK_DOWN) {

            charac.slide();

            high = 0;

            repaint();

        } else if (e.getKeyCode() == KeyEvent.VK_RIGHT) {

            score+=200;

            globalspeed++;

        }

    }

}

@Override

public void keyReleased(KeyEvent e) {

    super.keyReleased(e);

    if (e.getKeyCode() == KeyEvent.VK_DOWN) {

        charac.run();

        repaint();

    }

}
```

```

    }

}

});

this.addMouseListener(new MouseAdapter() {

    @Override

    public void mouseClicked(MouseEvent e) {

        super.mouseClicked(e);

        if (e.getX() >= 920 && e.getX() <= 970 && e.getY() >= 20 && e.getY() <= 60) {

            if (playpause == true) {

                playpause = false;

                time.stop();

                stage.suspend();

                actor.suspend();

                enm.suspend();

                ensky.suspend();

                showitem.suspend();

                enspcial.suspend();

                repaint();

            } else {

                time.start();

                playpause = true;

                stage.resume();

                actor.resume();

                ensky.resume();

                enm.resume();

                showitem.resume();

```

```
        enspcial.resume();

        repaint();

    }

}

if (e.getX() >= 920 && e.getX() <= 970 && e.getY() >= 80 && e.getY() <= 130) {

    System.exit(0);

}

}

});

]
```


หน้าจอเกมเมื่อกดเริ่มใหม่ตรงกลางจะเริ่มเกมใหม่และเมื่อกดปิดจะปิดเกม โดยมี MouseListener เป็นตัว
ดักฟังการกระทำต่างๆในหน้านี้



Code ของ MouseListener

```
if (over.x >= 425 && over.x <= 575 && over.y >= 225 && over.y <= 375){

    over.x = 0;

    over.y = 0;

    this.remove(over);

    letplay = new PLAYGAME();

    this.add(letplay);

    letplay.requestFocusInWindow();

    letplay.start();

}
```

อัลกอริทึมที่สำคัญ

อัลกอริทึมที่สำคัญคือการสร้างศัตรูให้ออกมาโดยไม่มีค่า y ไกลเคียงกันเกินไป

Code

```
if (spaenemy.x > 10 && spaenemy.x < getWidth() - 1) {

    if (skyenemy.x == getWidth() || skyenemy.x == getWidth() - 1) {

        skyenemy.x = getWidth();

        skyenemy.speed = 0;

    }

}

if (enemy.x == getWidth()-1 && enemy.x - skyenemy.x < 300 && skyenemy.x
>=10 && skyenemy.x <= getWidth()-1) {

    enemy.x = getWidth();

}
```

คือการสั่งให้ตัวบินที่จะออกมาวัดค่าของ y กับ ตัวเดินและให้รอจนถึงระยะที่กำหนดจึงเปลี่ยนความเร็ว

บทที่ 3 : สรุป

ปัญหาที่พบระหว่างการพัฒนา

ศัตรูตัว 2 – 3 ตัวออกมาพร้อมกันทำให้ผู้เล่นไม่สามารถหลบได้

จุดเด่นของโปรแกรมที่ไม่เหมือนใคร

โปรแกรมเป็นเกมวิ่งแบบแอคตปกติแต่ว่าเกทนี้มีการใส่บอสที่วิ่งเร็วและตัวใหญ่ออกมาเป็นระยะๆ