



Project Game

เกม : DINOREX RUN

โดย

นาย กัณฑ์ จิตรแก้ว

รหัสนักศึกษา 6404062610456 ตอนเรียนที่ 1

รายงานนี้เป็นส่วนหนึ่งของวิชา 040613222

Object Oriented Programming

ภาควิชาวิทยาการคอมพิวเตอร์และสารสนเทศ คณะวิทยาศาสตร์ประยุกต์

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

ภาคการศึกษาที่ 1 ปีการศึกษา 2565

สารบัญ

บทที่ 1 บทนำ	2-5
บทที่ 2 ส่วนการพัฒนา	6-15
บทที่ 3 สรุป	16

บทที่ 1 : บทนำ

ที่มาและความสำคัญ

โปรแกรมที่ผู้เขียนได้จัดทำขึ้นได้แรงบันดาลใจมาจาก เกมกระโดดบน บราวเซอร์ google chrome ผู้จัดทำจึงต้องการพัฒนาเกมด้วยภาษา JAVA ในการพัฒนาเกม DINOREX RUN เพื่อให้เกิดความเพลิดเพลินและความสนุก ทั้งยังได้รับประโยชน์จากการเล่นและพัฒนาเกมอีกด้วย

ประเภของโครงการ

เกมประเภท ARCADE GAME SINGLE PLAYER โดยไม่ใช้อินเตอร์เน็ต

ประโยชน์

1. ฝึกสมาธิและไหวพริบในการมองเห็น
2. เพื่อความสนุกสนาน และ เพลิดเพลิน
3. เพื่อฝึกกระบวนการคิดในสถานการณ์ฉุกเฉิน
4. ฝึกความอดทน

ขอบเขต

เกม : DINOREX RUN

● รายละเอียดเกม

เกม : DINOREX RUN เป็นเกมที่ทำให้ผู้เล่นได้ควบคุมตัวละคร โดยการกระโดดและสไลด์ เพื่อวิ่งหลบสิ่งศัตรู รวมถึงการเก็บไอเทมต่าง ๆ ที่มีในในเกม เนื่องจากตัวละครที่เราได้ควบคุมนั้นน่ารักเกินไปในหมู่ไดโนเสาร์ ไดโนเสาร์ตัวอื่นจึงไม่ชอบและต้องการจะทำร้าย จึงต้องวิ่งหนีเพื่อไม่ให้ไดโนเสาร์ตัวอื่นมาทำร้ายร่างกาย

● วิธีเล่น

ใช้ปุ่มลูกศรบนแป้นพิมพ์ , ปุ่ม w, ปุ่ม s , ปุ่ม space bar , ปุ่ม ctrl เพื่อกำหนดให้ตัวละครกระโดดหรือหมอบลง เพื่อหลบสิ่งกีดขวางและศัตรูทำให้ตัวละครมีชีวิตอยู่ได้นานที่สุด ปุ่มลูกศรขึ้น ปุ่มw ปุ่มspace bar ใช้ควบคุมเพื่อให้ตัวละครกระโดด และ ปุ่มลูกศรลง ปุ่มs ปุ่มctrl ใช้ควบคุมเพื่อให้ตัวละครสไลด์

- เนื้อหาของเกม

- ตัวละคร



ตัวละครที่ควบคุมโดยผู้เล่น



ศัตรู



กล่องไอเทม



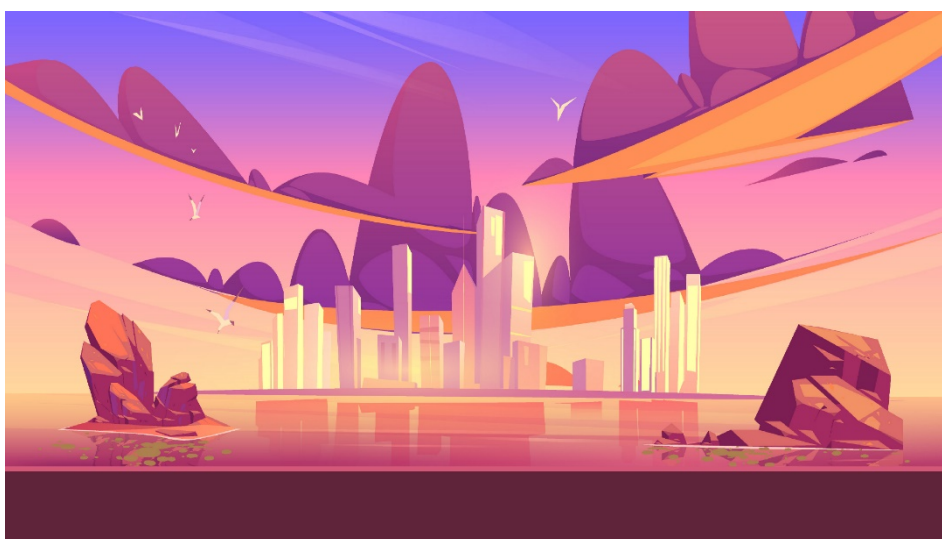
ศัตรูทางอากาศ



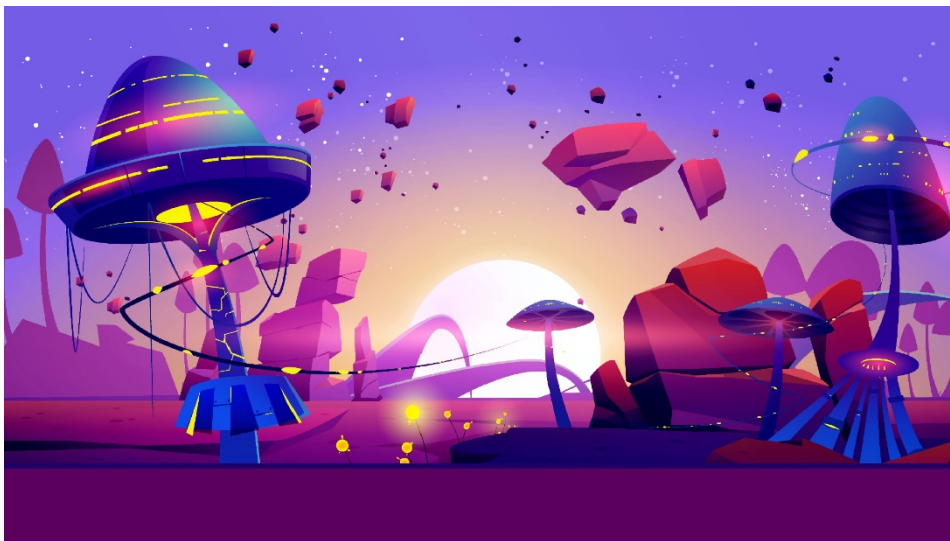
ศัตรูตัวBoss

- ฉาก

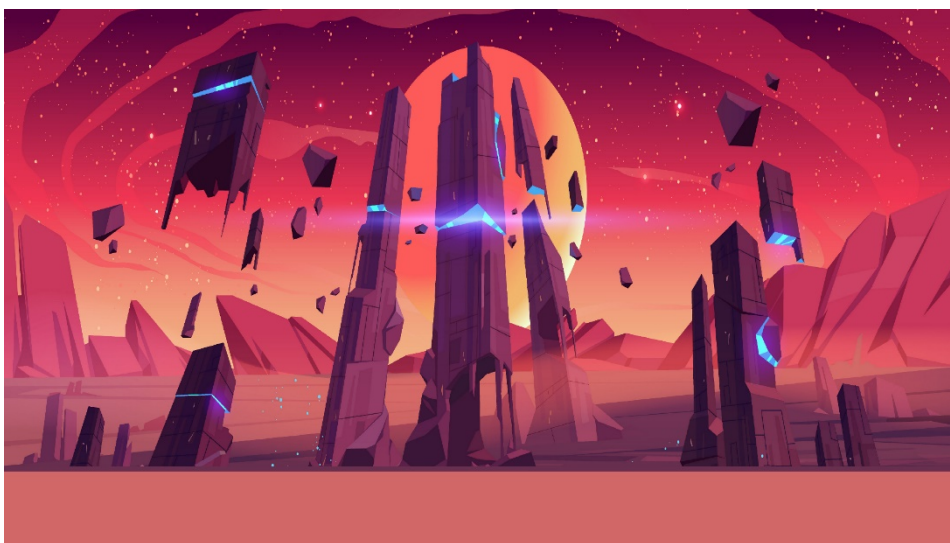
ด่านที่ 1



ด่านที่ 2



ด่านที่ 3



ฉากเมื่อจบเกม



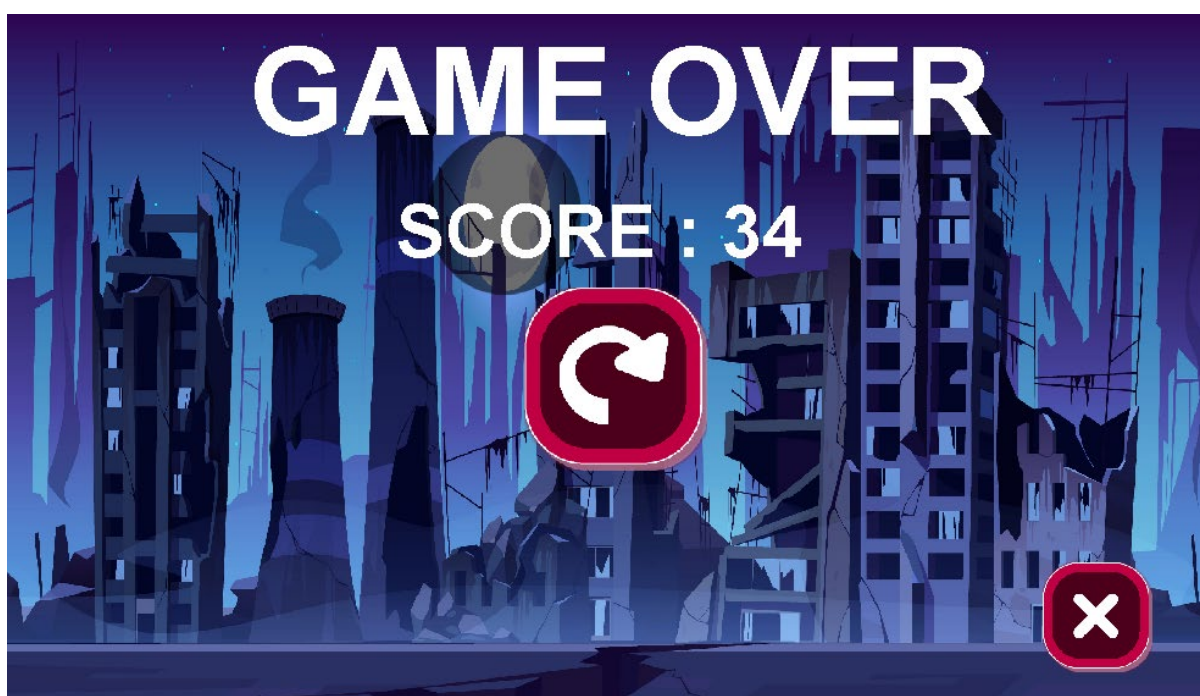
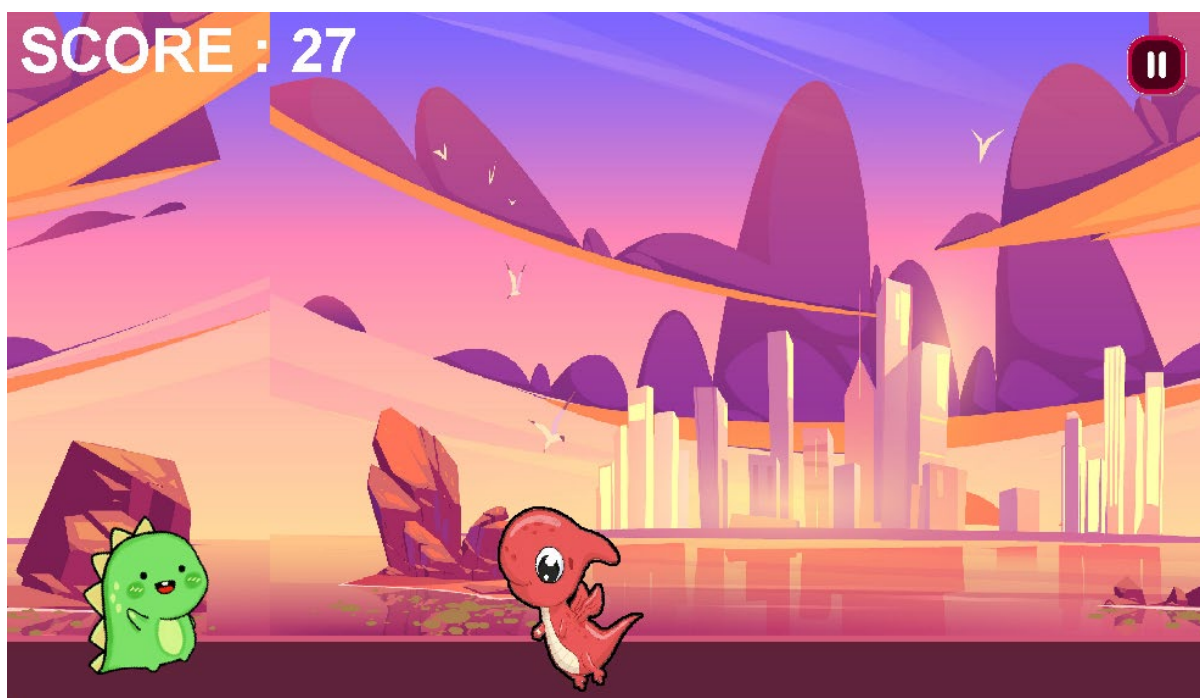
- แผนการทำงาน

ลำดับ	รายการ	สัปดาห์ที่ 1	สัปดาห์ที่ 2	สัปดาห์ที่ 3	สัปดาห์ที่ 4
1	หาตัวละครและจัดทำองค์ประกอบต่างๆ	←→			
2	ศึกษาเอกสารและข้อมูลที่เกี่ยวข้อ	←→	→		
3	เขียนโปรแกรม		←→	→	
4	ตรวจสอบและแก้ไขข้อผิดพลาด			←→	→
5	จัดทำเอกสาร				←→

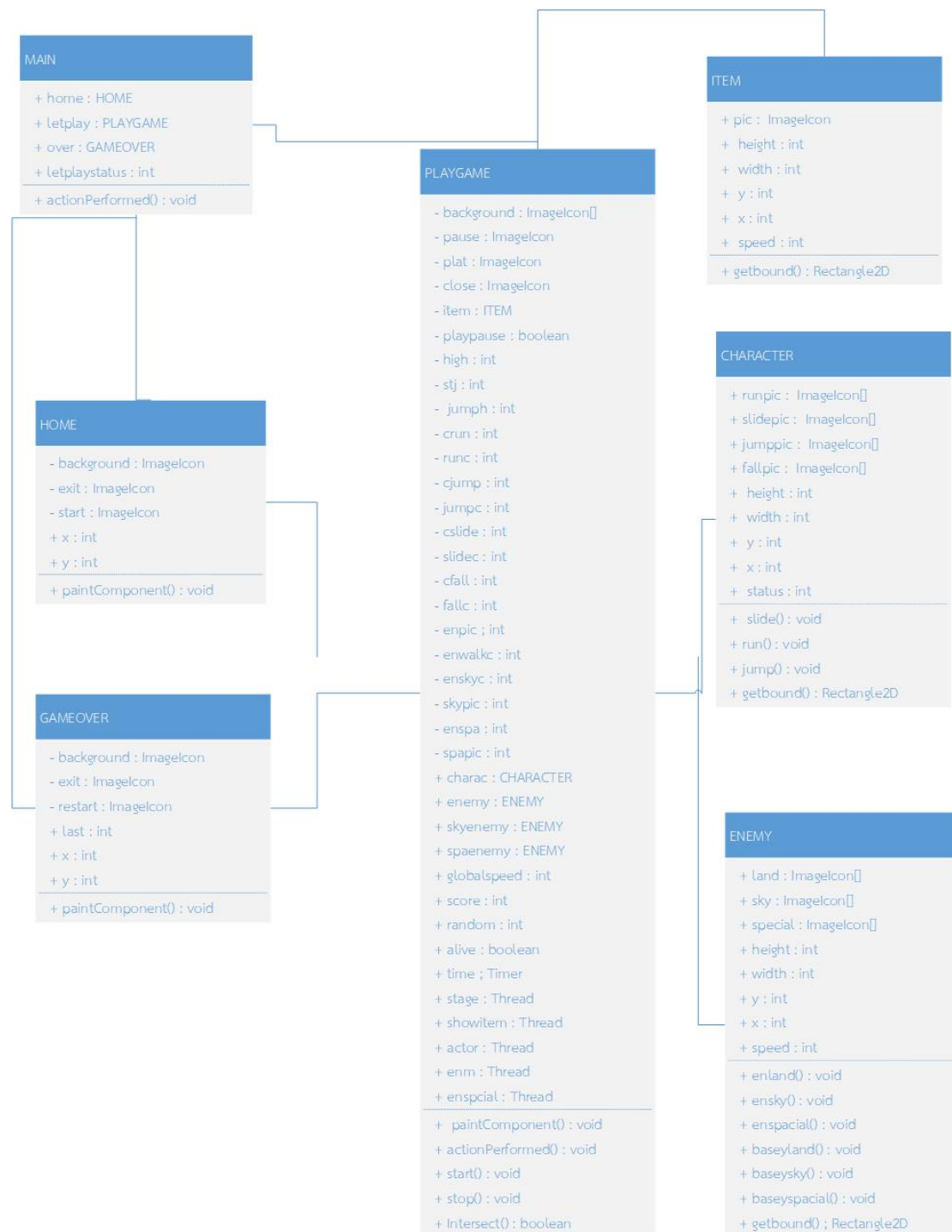
บทที่ 2 : ส่วนการพัฒนา

วิธีเล่น

ใช้ปุ่มบนแป้นพิมพ์ที่กำหนดเพื่อกำหนดให้ตัวละครกระโดดหรือหมอบลงเพื่อหลบสิ่งกีดขวางและศัตรู
ทำให้ตัวละครมีชีวิตรอยู่ได้นานที่สุด



คลาสไดอะแกรม



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( url: this.getClass().getResource( name: "Face 01.png" ));
    this.setIconImage( image: icon.getImage() );
    add( comp: home );
    time.start();
}
```

Encapsulation

ตัวอย่าง

```
private ImageIcon[] background = new ImageIcon[3];
private ImageIcon pause = new ImageIcon( url: this.getClass().getResource( name: "pause.png" ));
private ImageIcon play = new ImageIcon( url: this.getClass().getResource( name: "play.png" ));
private ImageIcon close = new ImageIcon( url: this.getClass().getResource( name: "close.png" ));
public int random = 0;
public int score = 0;
private int jump = 300;
private int high = 0;
private int stj = 0;
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();
public int globalspeed = 10;
public boolean alive = true;
```

Composition

ตัวอย่าง

```
if (letplay.alive == false) {
    over.last = letplay.score;
    this.remove( comp: letplay );
    this.add( comp: over );
    over.requestFocusInWindow();
}
```

Polymorphism

ตัวอย่าง

```
public void paintComponent(Graphics g) {
    super.paintComponent(g);
    g.drawImage (img: background.getImage(), x: 0, y: 0, width: 1000, height: 600, observer: this);
    g.drawImage (img: start.getImage(), x: 425, y: 225, width: 150, height: 150, observer: this);
    g.drawImage (img: exit.getImage(), x: 850, y: 450, width: 90, height: 90, observer: this);
    g.setFont (new Font (string: "Playtime", i: Font.CENTER_BASELINE, i1: 90));
    g.setColor (c: Color.WHITE);
    g.drawString (str: "DINOREX RUN", x: 180, y: 130);
}
```

Abstract

```
abstract class GraphicsObject{
    abstract void run();
    abstract void jump();
    abstract void slide();
}
```

Inheritance

ตัวอย่าง

```
public class MAIN extends JFrame implements ActionListener{
    public HOME home = new HOME();
    public PLAYGAME letplay = new PLAYGAME();
    public GAMEOVER over = new GAMEOVER();
    public int letplaystatus = 0;
    Timer time = new Timer (i: 100, a1: this);
}
```

หน้าจอ GUI และ Event handling

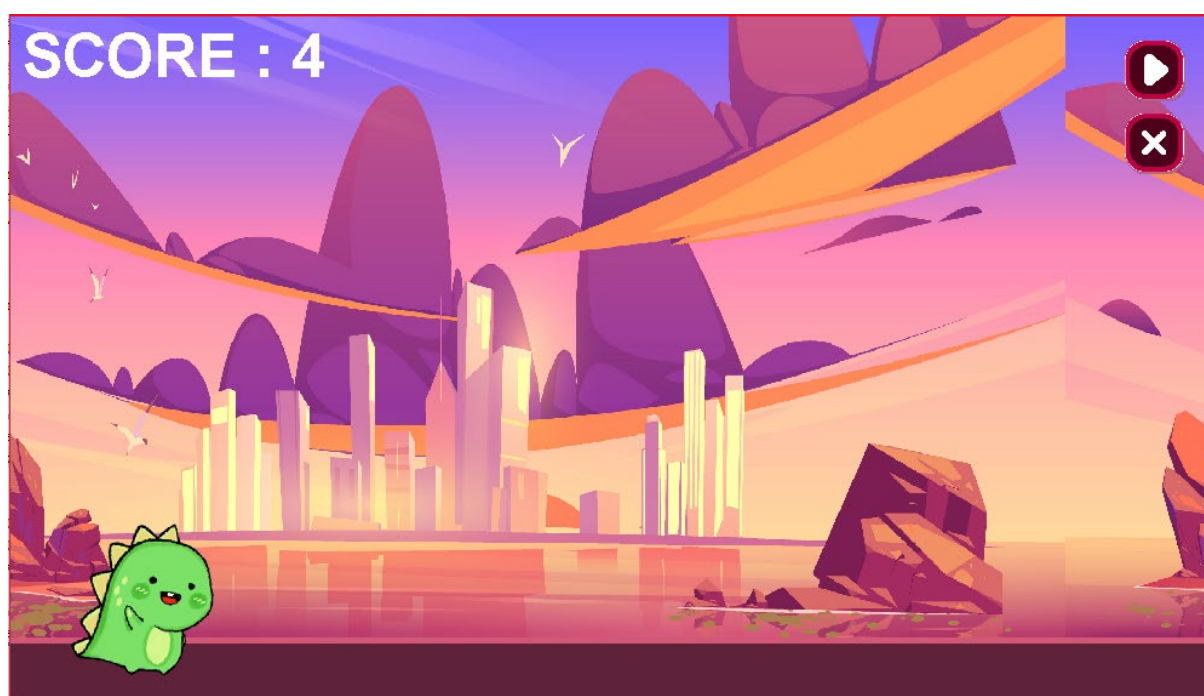
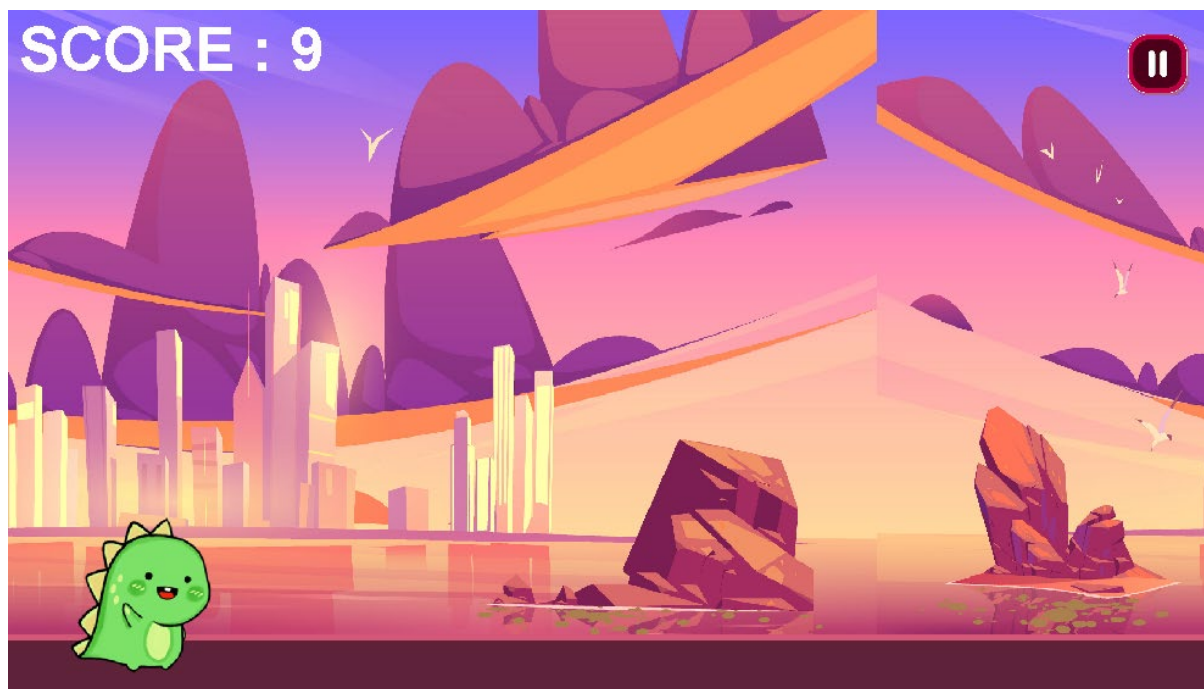
หน้าเริ่มต้นเมื่อกดเริ่มตรงกลางจะเริ่มเกมและเมื่อกดปิดจะปิดเกม โดยมี MouseListener เป็นตัวดักฟังการกระทำต่างๆในหน้านี้



Code ของการดักฟังในหน้าจอเริ่มแรก

```
if (home.x >= 425 && home.x <= 575 && home.y >= 225 && home.y <= 375) {
    home.x = 0;
    home.y = 0;
    this.remove( comp:home);
    this.add( comp:letplay);
    letplay.requestFocusInWindow();
    letplay.start();
}
```

หน้าจอเมื่อกดหยุดเกมจะหยุดและสร้างปุ่มที่สามารถกดเพื่อปิดได้ โดยมี MouseListener เป็นตัวดักฟังการกระทำต่างๆในหน้านี้ 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++;
        } else if (e.getKeyCode() == 32) {
            if (charac.status == 0) {
                charac.jump();
                stj = 1;
                repaint();
            }
        } else if (e.getKeyCode() == 87) {
            if (charac.status == 0) {
                charac.jump();
                stj = 1;
                repaint();
            }
        } else if (e.getKeyCode() == 83) {
            charac.slide();
            high = 0;
            repaint();
        } else if (e.getKeyCode() == 17) {
            charac.slide();
            high = 0;
            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();
                enspecial.suspend();
                repaint();
            } else {
                time.start();
                playpause = true;
                stage.resume();
                actor.resume();
                ensky.resume();
                enm.resume();
                showitem.resume();
                enspecial.resume();
                repaint();
            }
        }
        if (e.getX() >= 920 && e.getX() <= 970 && e.getY() >= 80 && e.getY() <= 130) {
            System.exit(status:0);
        }
    }
}

```


หน้าจอเกมเมื่อกดเริ่มใหม่ตรงกลางจะเริ่มเกมใหม่และเมื่อกดปิดจะปิดเกม โดยมี MouseListener เป็นตัวดักฟังการกระทำต่างๆในหน้านี้



Code ของ MouseListener

```
if (over.x >= 425 && over.x <= 575 && over.y >= 225 && over.y <= 375) {
    over.x = 0;
    over.y = 0;
    this.remove ( comp: over );
    letplay = new PLAYGAME ();
    this.add ( comp: letplay );
    letplay.requestFocusInWindow ();
    letplay.start ();
}
```

Algorithm

อัลกอริทึมที่สำคัญคือเมื่อ score ผ่านไปถึง 500 จะมีศัตรูบินโผล่มา

Code

```

skyenemy.ensky();
if (score == 0) {
    skyenemy.speed = 0;
    skyenemy.x = getWidth();
    skyenemy.baseysky();
}
if (score < 500) {
    skyenemy.speed = 0;
    skyenemy.x = getWidth();
    skyenemy.baseysky();
}
if (skyenemy.x - enemy.x < 400 && skyenemy.x == getWidth() ) {
    if (score > 500) {
        skyenemy.x = getWidth();
        continue;
    }
}
else if (skyenemy.x == getWidth()) {
    int x = (int) (Math.random() * 10) % 10;
    if (x >= 6) {
        skyenemy.x = getWidth() - 1;
        skyenemy.y = 280 - ((int) (Math.random() * 100) % 280);
        skyenemy.speed = globalspeed;
    } else {
        skyenemy.speed = 0;
        if (score != 0) {
            Thread.sleep( millis:1000 );
        }
    }
}

```

บทที่ 3 : สรุป

ปัญหาที่พบระหว่างการพัฒนา

ศัตรูโผล่มาเป็นคู่ตัวบินกับตัวเดินพร้อมกัน

ผู้เล่นไม่สามารถหลบศัตรูได้เนื่องจากศัตรูโผล่มาพร้อมกัน

จุดเด่นของโปรแกรมที่ไม่เหมือนใคร

โปรแกรมเป็นเกมวิ่งแบบแอคตปกติแต่ว่าเกทนี้มีการใส่บอสที่วิ่งเร็วและตัวใหญ่ออกมาเป็นระยะๆ

กราฟฟิกที่สวยงาม

ตัวละครที่น่ารัก