

Halloween Run

บทนำ

ที่มาและความสำคัญ

การเรียนรู้วิชา Object Oriented Programming เป็นการศึกษาเกี่ยวกับการเขียนโปรแกรมในรูปแบบของ การมองหลายๆอย่าง เป็น Object ที่แตกต่างจากหลายๆภาษาที่เรียนมา ซึ่งการที่จะสามารถทำให้เราเข้าใจการเรียนรู้วิชานี้มากขึ้นการสร้างเกมส์จากภาษา จาวา เป็นทางเลือกหนึ่งที่ทำให้เข้าใจในการเรียนการสอนได้มากยิ่งขึ้นเนื่องจากเกมส์มีคุณสมบัติหลายๆอย่างที่เหมะมากับการมองอย่างเป็น Object ซึ่งเกมส์ Halloween Run นั้นได้รับแรงบัลดาลใจมาจากเกมส์ที่โด่งดังอย่าง Cookie Run

ประเภทของโครงการ : โครงการทดลองและวิจัย

ประโยชน์

เพื่อเพิ่มความเข้าใจเกี่ยวกับวิชา Object Oriented Programming มากยิ่งขึ้น และเป็นตัวอย่างสำหรับคนอื่นที่อยากจะศึกษาเหมือนกัน

ขอบเขตของโครงการ

ทำเกมส์ที่มีความสามารถและรูปแบบการเล่นเหมือนเกมส์ Cookie Run โดยลดความสามารถของเกมส์มาระดับหนึ่งเพื่อให้เหมาะสมในการพัฒนาด้วยทรัพยากรที่มีอยู่

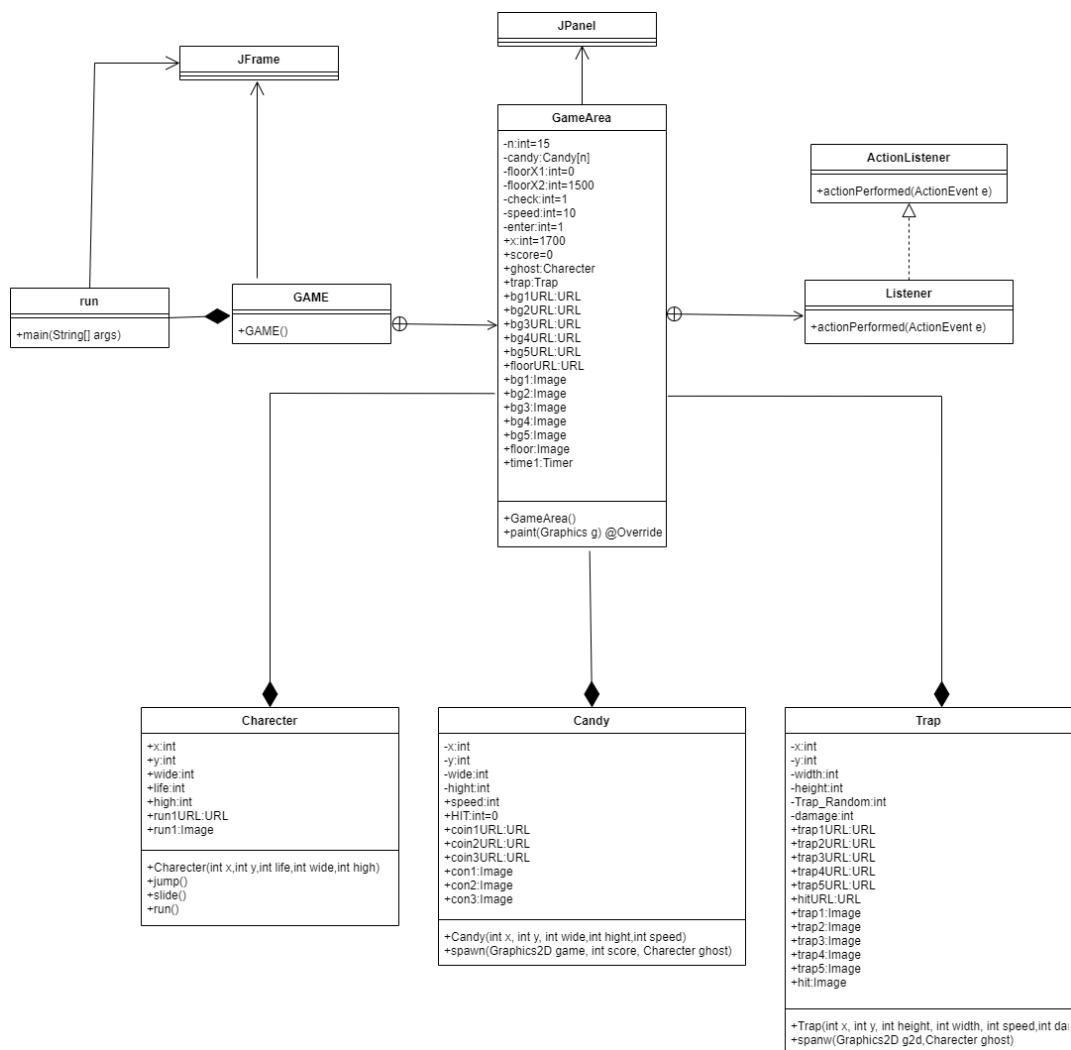
ส่วนการพัฒนา

เนื้อเรื่อง

คำคืนวันฮาโลวีน เหล่าภูตผีปีศาจจากโลกหลังความตายจะออกมาไล่ล่าลูกอมในดินแดนมนุษย์ เพื่อนำกลับไปในโลกหลังความตายของตน แต่มนุษย์นั้นไม่ยอมให้เหล่าภูตผีนำลูกอมไปง่ายๆ จึงได้วางกับดักปราบเหล่าภูตผีปีศาจเพื่อไม่ให้แย่งลูกอมไป GHOST MAN ผู้ได้รับหน้าที่แย่งชิงลูกอมไปจากเหล่ามนุษย์ จำเป็นต้องนำลูกอมกลับไปยังดินแดนหลังความตายให้มากที่สุดเท่าที่จะทำได้

วิธีการเล่น

Ghost man ต้องวิ่งเก็บลูกกวาดไปเรื่อยๆ ใช้ปุ่มบนแป้นคีย์บอร์ดปุ่ม J และ F เพื่อกระโดดและหลบกับดักที่ มนุษย์สร้างขึ้น จนกว่าพลังชีวิตจะหมด



คลาสไดอะแกรม

มีคลาส GAME extend JFrame ซึ่งมี คลาส GameArea extend JPanel อยู่ข้างใน

และมี Class Listener ที่ implements interface Action listener อยู่ภายใน GameArea

มีclass Character Trap Candy run สำหรับ สร้าง ตัวละคร กับดัก ลูกอม และรัน โปรแกรม ที่เป็น ส่วนประกอบของ GameArea

รูปแบบการพัฒนา : Application

อธิบายโปรแกรม

-Constructor

```
public GAME(){  
    add(new GameArea());  
}
```

Constructor ของ class game ที่ extends JFrame ทำการเพิ่ม new Game Area(); ลงไป

```
public GameArea(){  
    // CREATE CANDY OBJECT  
    for(int i=0;i<15;i++){  
        candy[i]=new Candy(x, y: 545, wide: 100, high: 100, speed);  
        x+=100;  
    }  
    timer1.start();  
    setFocusable(true);  
  
    //KEYBOARD CONTROL  
    addKeyListener(new KeyListener() {  
        @Override public void keyTyped(KeyEvent e) {}  
  
        @Override  
        public void keyPressed(KeyEvent e) {  
            if(e.getKeyCode()==74){  
                ghost.jump();  
            }  
            if(e.getKeyCode()==70){  
                ghost.slide();  
            }  
            if(e.getKeyCode()==27){  
                timer1.stop();  
            }  
            if(enter==1){  
                if(e.getKeyCode()==KeyEvent.VK_ENTER){  
                    enter=0;  
                }  
            }  
        }  
        @Override  
        public void keyReleased(KeyEvent e) {  
            timer1.start(); //BACK TO DEFAULT VALUE  
            ghost.y=545;  
            ghost.wide=100;  
            ghost.high=100;  
        }  
    })  
}
```

Constructor ของ class GameArea ที่ extends JPanel

มีการสร้าง object candy Type เป็น Candy ไว้ใน array ทั้งหมด 15ตัว

เริ่มนับเวลา และ setFocusable(true);

มีเมทอดคักจับการกดปุ่มของเรา j f esc enter เพื่อ โดด สไลด์ หยุดเกมส์ เข้าเกมส์ และปล่อยปุ่ม

```
public Candy(int x, int y, int wide,int hight,int speed) {  
    this.x = x;  
    this.y = y;  
    this.wide=wide;  
    this.hight=hight;  
    this.speed=speed;  
}
```

Constructor ของ class Candy มีการรับค่าเพื่อกำหนดพิกัด ความกว้าง สูง และความเร็วให้กับ class Candy

```
public Trap(int x, int y, int height, int width, int speed,int damage) {  
    this.x = x;  
    this.y = y;  
    this.height = height;  
    this.width = width;  
    this.speed = speed;  
    this.damage=damage;  
}
```

Constructor ของ class Trap Candy มีการรับค่า เพื่อกำหนดพิกัด ความกว้าง สูง ความเร็ว และ damage ให้กับ class Trap

```
public Charecter(int x,int y,int life,int wide,int high) {  
    this.x=x;  
    this.y=y;  
    this.life=life;  
    this.wide=wide;  
    this.high=high;  
}
```

Constructor ของ class Trap Candy มีการรับค่า เพื่อกำหนดพิกัด ความกว้าง สูง และพลังชีวิตให้กับ class Charecter

-Encapsulation

มีตัวแปร แบบ public และ private สำหรับใช้นอกคลาสและภายในคลาส

และ method แบบ public ทั้งหมดสำหรับไปใช้นอกคลาสได้

-Composition

```
public static class GameArea extends JPanel{  
  
    private int n=15;  
    private Candy candy[]= new Candy[n];  
    private int floorX1=0;  
    private int floorX2=1500;  
    private int check=1;  
    private int speed=10;  
    private int enter=1;  
    public int x=1700;  
    public int score=0;  
  
    //NEW OBJECT  
    Charecter ghost=new Charecter( x: 200, y: 545, life: 200, wide: 100, high: 150);  
    Trap trap = new Trap( x: 1400, y: 545, height: 150, width: 100,speed, damage: 2);  
}
```

คลาส GameArea มีส่วนประกอบ ที่เป็น Class อื่นๆ มี Candy Character Trap เพื่อใช้ในการสร้าง Object ที่เป็นตัวละคร กับดัก และ ลูกอมมาใช้งาน

-inheritance

```
public class GAME extends JFrame {
```

Class GAME extends JFram เพื่อทำการใช้ในการสร้างหน้าต่าง JFrame

```
public static class GameArea extends JPanel{
```

Class GameArea extends JPanel เพื่อใช้ในการแปะหน้ากระดาษเข้าไปใน JFram เพื่อที่จะสามารถวาดองค์ประกอบต่างๆลงไปได้

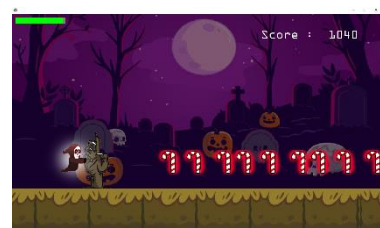
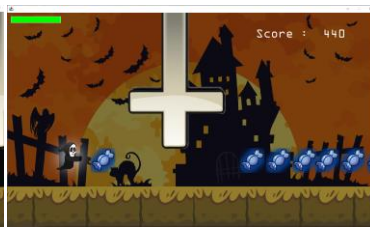
หน้าจอ GUI



```
//PAINT
@Override
public void paint(Graphics g) {

    //HOME
    if (enter == 1) {
        g.drawImage(bg4, x: 0, y: 0, getWidth(), getHeight(), observer: this);
    }
}
```

สร้างหน้าแรกใน method paint โดยใช้



```
//GAME GRAPHIC
if (check == 1 & enter == 0) {
    super.paint(g);
    Graphics2D g2d = (Graphics2D) g;
    //BACKGROUND
    if (score < 300) {
        g.drawImage(bg1, x: 0, y: 0, getWidth(), getHeight(), observer: this);
    } else if (score < 900) {
        g.drawImage(bg2, x: 0, y: 0, getWidth(), getHeight(), observer: this);
        this.speed = 15;
        for (int i = 0; i < n; i++) {
            candy[i].speed = 15;
        }
        trap.speed = 15;
    } else if (score >= 900) {
        g.drawImage(bg3, x: 0, y: -150, getWidth(), getHeight(), observer: this);
        this.speed = 20;
        for (int i = 0; i < n; i++) {
            candy[i].speed = 20;
        }
        trap.speed = 20;
    }
}
```

เปลี่ยนฉากหลัง ความเร็วต่างๆภายในเกมส์

```

//DRAW GHOST TRAP CANDY
ghost.run(g2d);
//SPAWN CANDY
for(int i=0;i<n;i++){
    candy[i].spawn(g2d,score,ghost);
}
trap.spawn(g2d,ghost);
//DEDUCT TRAP
if(trap.deduct==1&&trap.HIT==1){
    score-=10;
} trap.HIT=0;
//CHECK HIT CANDY
for (int i = 0; i < n; i++) {
    if(candy[i].HIT==1){
        score+=10;
    }candy[i].HIT=0;
}

```

วาดตัวละคร สร้างกับคั๊กและลูกอม โดยเรียก ghost.run(g2d); trap.spawn(g2d);candy[i].spawn(g2d);

```

//FLOOR
g.drawImage(floor,floorX1,y,700,width:1520,height:200,observer:this);
g.drawImage(floor,x:floorX2-1,y:700,width:1520,height:200,observer:this);
floorX1-=speed;
floorX2-=speed;
if(floorX1<-1500 ){
    floorX1=1500;
}
if(floorX2<-1500 ){
    floorX2=1500;
}

//HP
g.setColor(Color.GRAY);
g.fillRect(x:15,y:15,width:200,height:25);
g.setColor(Color.GREEN);
g.fillRect(x:15,y:15,ghost.life,height:25);
g.setColor(Color.WHITE);
g.drawRect(x:15,y:15,width:200,height:25);
}

```

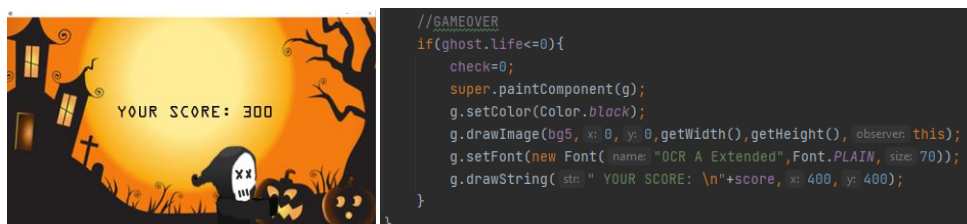
วาดพื้นด้านล่างและหลอดเลือด

```

//SCORE
g.setFont(new Font( name: "OCR A Extended",Font.PLAIN, size: 50));
g.setColor(Color.white);
g.drawString( str: "Score : "+score,x:1000,y:100);
g.setColor(Color.RED);

```

วาดคะแนน



วาดหน้าต่างจบเกมส์เมื่อพาลังชีวิตหมด

Event handling

```
//KEYBOARD CONTROL
addKeyListener(new KeyListener() {
    @Override public void keyTyped(KeyEvent e) {}

    @Override
    public void keyPressed(KeyEvent e) {
        if(e.getKeyCode()==74){
            ghost.jump();
        }
        if(e.getKeyCode()==70){
            ghost.slide();
        }
        if(e.getKeyCode()==27){
            time1.stop();
        }if(enter==1){

            if(e.getKeyCode()==KeyEvent.VK_ENTER){
                enter=0;
            }
        }
    }
}
@Override
```

มีการตรวจจับการกดต่างๆ ปุ่ม j กระโดด f หมอบ esc หยุดเกมส์ enter เข้าเกมส์

```

    @Override
    public void keyReleased(KeyEvent e) {

        time1.start();          //BACK TO DEFAULT VALUE
        ghost.y=545;
        ghost.wide=100;
        ghost.high=150;
    }
}
```

ปล่อยปุ่ม แล้วค่าต่างๆจะกลับไปเป็นเหมือนเดิม

อัลกอริทึมที่สำคัญ

```
//SPAWN TRAP
public void spawn(Graphics2D g2d,Charecter ghost) {
    x -= speed;
    if (x < -300) {
        x = 1700;
        Trap_Random = rand.nextInt( bound: 5); //RANDOM
    }if(Trap_Random==0){
        g2d.drawImage(trap1, x: x-100, y: y-70, observer: null); //TRAP1
        damage=2;
        y=0;
        height=560;
        deduct=0;
    }else if(Trap_Random==1){
        g2d.drawImage(trap2, x: x-150, y: y-80, observer: null); //TRAP2
        damage=5;
        y=0;
        height=560;
        deduct=0;
    }else if(Trap_Random==2){
        g2d.drawImage(trap3,x, y: y-70, observer: null); //TRAP3
        damage=2;
        y=545;
        height=150;
    }
```

```
}else if(Trap_Random==3){
    g2d.drawImage(trap4, x: x-20, y: y-50, observer: null); //TRAP4
    damage=0;
    y=545;
    height=150;
    deduct=1;
}

}else if(Trap_Random==4){
    g2d.drawImage(trap5, x: x-30, y: y-70, observer: null); //TRAP5
    damage=200;
    y=545;
    height=150;
    deduct=0;
}
}

//HITBOX
if((ghost.x<=x&&ghost.x+ghost.wide>=x)|| (x+width>=ghost.x&&x+width<=ghost.x+ghost.wide)) {
    if ((ghost.y + ghost.high >= y&&ghost.y + ghost.high<=y+height)|| (ghost.y>=y&&ghost.y<=y+height)) {
        g2d.drawImage(hit, x: ghost.x-150, y: ghost.y-100, observer: null);
        ghost.life-=damage;
        HIT=1;
    }
}
}
```

Spawn จาก class Trap

Trap_Random = rand.nextInt(5);

สุ่มกับดักให้เกิด โดยจะมีการสุ่มกับดักมาในหน้าจอ 5 ชนิด โดยแต่ละอันจะมีรูปร่างและความสามารถแตกต่างกัน ตายทันที ลดคะแนน หรือ ลดพลังชีวิต และตรวจจับ hitbox ถ้าชนก็จะเกิดผลตามแต่ละกับดักที่สุ่มได้และตัวจะกลายเป็นสีแดง

```

//SPAWN CANDY
public void spawn(Graphics2D game, int score, Charecter ghost) {
    x -= speed;
    if (x < -300) {
        x = 1700;
        y=545;
    }
    //CHANGE CANDY
    if(score<300){
        game.drawImage(coin1,x,y,wide,height, observer: null);
    }else if(score<900){
        game.drawImage(coin2,x,y,wide,height, observer: null);
    }else if(score>=900){
        game.drawImage(coin3,x,y,wide,height, observer: null);
    }

    //HITBOX
    if((ghost.x<=x&&ghost.x+ghost.wide>=x)|| (x+wide>=ghost.x&&x+wide<=ghost.x+ghost.wide)) {
        if ((ghost.y + ghost.high >= y&&ghost.y + ghost.high<=y+ high)|| (ghost.y>=y&&ghost.y<y+high)) {
            HIT=1;
            y=-100;
        }
    }
}
}

```

Spawn จาก class Candy

วาดลูกอมในหน้าจอหากวิ่งไปทางซ้ายเรื่อยๆ และเกิดใหม่เมื่อชนกับผู้เล่นคะแนนก็จะเพิ่ม และเมื่อคะแนนถึงกำหนดก็จะเปลี่ยนรูปลูกอม

สรุป

ปัญหาที่พบระหว่างการพัฒนา

การกำหนดค่า x และ y ให้กับรูปภาพเป็นสิ่งที่ลำบากมากที่สุดถ้าไม่มีการจัดเตรียมรูปภาพมาอย่างดี รูปภาพอาจเล็กไป บ้าง ไม่เท่ากับ HIT BOX บ้างเหลื่อมบ้าง จึงทำให้ต้องกำหนดค่าให้ใหม่ในหลายๆรูป

จุดเด่นของโปรแกรมที่ไม่เหมือนใคร

มีการสุ่มเกิดของกับดักของเกมส์ที่ไม่เหมือนกันสักครั้งในแต่ละรอบการเล่นทำให้ผู้เล่นไม่สามารถเดาได้เลยว่าจะมีอะไรรอเราอยู่

สิ่งที่อยากให้อาจารย์เพิ่มให้แก่น้องคือ

เรื่องเล็กๆที่ไม่อาจมองข้ามเช่น memory ของโปรแกรมให้กับน้องๆ เพราะสำหรับผมคิดว่ามันเป็นเรื่องที่ใกล้ตัวมากและยังไม่เข้าใจพอเท่าที่ควร การเขียนโปรแกรมของผมเลยอาจจะมีการใช้ทรัพยากรอย่างสิ้นเปลือง