



โครงการ

แจ็คเป็นคาวบอย

(Jack is Cowboy)

จัดทำโดย

6604062630498 นาย วรชน มีมูล

เสนอ

ผู้ช่วยศาสตราจารย์ สติติย์ ประสมพันธ์

วิชา **040613204 Object-Oriented Programming**

ภาคเรียนที่ 1 ปีการศึกษา 2567

ภาควิชาวิทยาการคอมพิวเตอร์และสารสนเทศ คณะวิทยาศาสตร์ประยุกต์

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

เกี่ยวกับโครงงาน

ชื่อโปรเจก : แจ็คเป็นคาวบอย (Jack is Cowboy)

นำเสนอโดย : นายวรณ มีมูล

อาจารย์ : ผู้ช่วยศาสตราจารย์สฤติย์ ประสมพันธ์

Source Code : <https://github.com/stamp070/2dgames>

บทที่ 1 ที่มาและความสำคัญของโครงงาน

โครงงานนี้จัดขึ้นเพื่อวัดผลความสามารถในการเรียนวิชา Object Oriented Programming

โดยการนำเรื่องที่เรียนมาสร้างเป็นชิ้นงานในรูปแบบของเกมโดยใช้แนวคิดการเขียนโปรแกรมเชิงวัตถุและยังช่วยให้ผู้จัดทำเรียนรู้อุปกรณ์และเครื่องมือ ผู้จัดทำจึงได้สร้างเกมนี้ขึ้นมา

ประเภทโครงงาน

เกม(game)

ประโยชน์

1. ฝึกไหวพริบในการเอาตัวรอด
2. ฝึกการแก้ไขปัญหา
3. ฝึกความอดทน

ขอบเขตโครงการ

1. ตารางเวลาการดำเนินโครงการ (Project Schedule)

ลำดับ	รายการ	21 - 26	27 - 30	31 - 30
1	หารูปตัวละครและทำกราฟิกต่างๆ			
2	ศึกษาเอกสารและข้อมูลที่เกี่ยวข้อง			
3	ลงมือเขียนโปรแกรม			
4	จัดทำเอกสาร			
5	ตรวจสอบและแก้ไขข้อผิดพลาด			

บทที่ 2 การพัฒนา

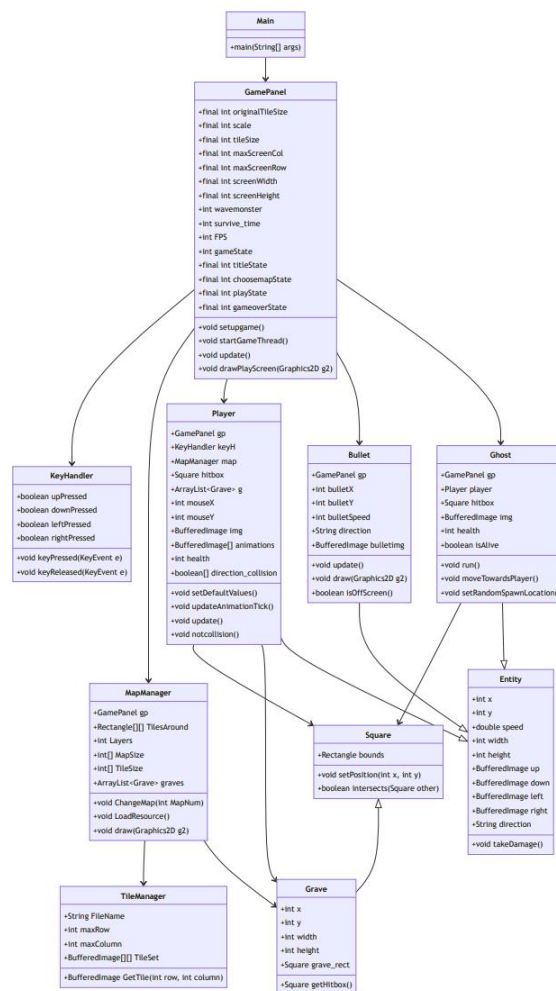
2.1 รายละเอียดโครงการงาน

เป็นเกมที่ตัวผู้เล่นจะรับบทเป็นคาวบอยผู้ปราบผี โดยตัวผีจะออกมาจาก 4 ทิศ เหนือ, ใต้, ตะวันออก, ตะวันตก โดยที่แจ็คจะมีหลอดเลือดอยู่ 3 หลอด เกมจะจบลงเมื่อแจ็คเลือดหมดหลอด

2.2 วิธีการเล่น

กดปุ่ม w, a, s, d

เพื่อควบคุมทิศทางในการเคลื่อนที่และมีการใช้เมาส์เพื่อเลือกทิศทางที่จะยิงปืนและกดคลิกซ้ายเพื่อเป็นการยิงปืน



2.4 การทำงานของส่วนต่างๆ

2.4.1 Constructor

```
1 public GamePanel(){
2     this.setPreferredSize(new Dimension(screenWidth,screenHeight));
3     this.setBackground(Color.black);
4     this.setDoubleBuffered(true);
5     this.addKeyListener(keyH);
6     this.setFocusable(true);
7
8 }
```

กำหนดขนาดหน้าจอและเปลี่ยนพื้นหลังให้เป็นสีดำใช้งาน Double Buffering ด้วยเพื่อให้การ

แสดงผลกราฟิกมีความราบรื่นและลดการกระพริบของภาพนอกจากนี้ยังเพิ่ม KeyListener

เพื่อให้สามารถควบคุมการเล่นเกมนผ่านคีย์บอร์ดได้ และ setFocusable(true) เพื่อให้สามารถ

ตอบสนองต่อการกดคีย์บอร์ดได้

```
1 public Player(GamePanel gp, KeyHandler keyH, MapManager map) {
2     this.gp = gp;
3     this.keyH = keyH;
4     this.map = map;
5     setDefaultValues();
6     getPlayerImage();
7     gp.addMouseMotionListener(this);
8     gp.addMouseListener(this);
9     img = down;
10
11     this.hitbox = new Square(x+ pixelMC , y +pixelMC , width, width);
12 }
```

กำหนด GamePanel, keyHandler, MapManager พร้อม set ค่าที่ extends entity

พร้อมทั้งดึงรูป player ออกมาและเพิ่ม MouseEvent จากนั้นเช็คค่า Hitbox ของตัวละคร

```
1 public Ghost(GamePanel gp, Player player) {
2     this.gp = gp;
3     this.player = player;
4     spawnLocations = new int[][] {
5         {100, 100},
6         {gp.screenWidth - 200, 100},
7         {100, gp.screenHeight - 200},
8         {gp.screenWidth - 200, gp.screenHeight - 200},
9         {gp.screenWidth / 4, gp.screenHeight / 4},
10        {3 * gp.screenWidth / 4, gp.screenHeight / 4},
11        {gp.screenWidth / 4, 3 * gp.screenHeight / 4},
12        {3 * gp.screenWidth / 4, 3 * gp.screenHeight / 4}
13    };
14    setRandomSpawnLocation();
15    setDefaultValues();
16
17    try {
18        img = ImageIO.read(getClass().getResourceAsStream("/res/ghost/ghost-Sheet.png"));
19        loadAnimations();
20    } catch (IOException e) {
21        e.printStackTrace();
22    }
23    this.hitbox = new Square(x+sethitbox, y+sethitbox, width, height);
24 }
```

กำหนด GamePanel และ Player เพื่อใช้ Method ของ Player เพื่อคิดฟังก์ชันเดินไปหาผู้เล่น
จากนั้นเช็ค spawnLocations จาก Variable ของ GamePanel และใช้ Method สุ่มค่า
SpawnLocation เป็นค่าที่ Ghost เกิด จากนั้นก็เช็คค่าที่ extends จาก entity และทำการโหลด
ภาพ Ghost และสร้าง Hitbox ของ Ghost

```
1 public Bullet(GamePanel gp,int x,int y, String direction){
2     this.gp = gp;
3     this.bulletX = x;
4     this.bulletY = y;
5     this.direction = direction;
6 }
```

กำหนด GamePanel และ เช็คค่า x, y, direction ในคลาส Bullet

```

1 public Square(int x, int y, int width, int height) {
2     bounds = new Rectangle(x, y, width, height);
3     this.x = x;
4     this.y = y;
5 }

```

กำหนด bounds เป็น Rectangle และ เช็ตค่า x, y ในคลาส Square

2.4.2 Encapsulation & Composition

```

1 public class GamePanel extends JPanel implements Runnable{
2     final int originalTileSize = 16;
3     final int scale = 2;
4
5     public final int tileSize = originalTileSize * scale;
6     public final int maxScreenCol = 30;
7     public final int maxScreenRow = 21;
8     public final int screenWidth = tileSize * maxScreenCol;
9     public final int screenHeight = tileSize * maxScreenRow;
10
11     int wavemonster = 1;
12     int survive_time = 0;
13
14     int FPS = 60;
15
16     public int gameState = 0;
17     public final int titleState = 0;
18     public final int choosemapState = 1;
19     public final int playState = 2;
20     public final int gameoverState = 3;
21
22     OBJ_heart heart = new OBJ_heart(this);
23     MapManager map = new MapManager(this,0);
24     KeyHandler keyH = new KeyHandler();
25     Thread gameThread;
26     Player player = new Player(this,keyH,map);

```

Attribute และ Method เกือบทุกคลาสจะเป็น public เพื่อให้ง่ายต่อการเข้าถึงและในคลาสต่างๆ

จะมีการใช้ Composite มาจากคลาสอื่น เช่น คลาส OBJ_heart, Player etc.

2.4.3 Polymorphism

ขณะนี้ยังไม่มีการใช้ Polymorphism แต่สามารถประยุกต์ใช้ในอนาคตได้

2.4.4 Abstract

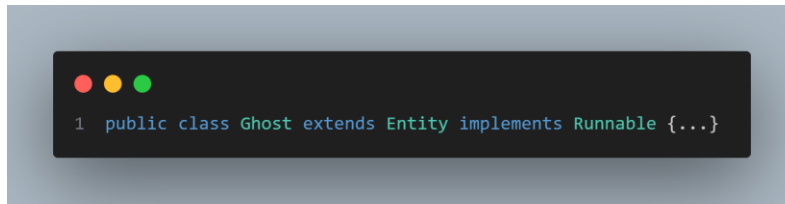
```
1 public abstract class Entity {  
2     public int x,y;  
3     public double speed;  
4     public int width,height;  
5  
6     public BufferedImage up, down, left, right;  
7     public String direction;  
8  
9     public abstract void takeDamage();  
10 }
```

สร้าง Abstract คลาส Entity และ Abstract Method takeDamage()

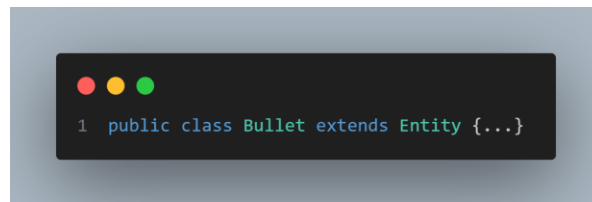
2.4.5 Inheritance

```
1 public class Player extends Entity implements MouseMotionListener,MouseListener {...}
```

คลาส Player มีการสืบทอดคลาสมาจากคลาส Entity



คลาส **Ghost** มีการสืบทอดคลาสมาจากคลาส **Entity**



คลาส **Bullet** มีการสืบทอดคลาสมาจากคลาส **Entity**

2.5 ส่วนประกอบ GUI

2.5.1 หน้า Title



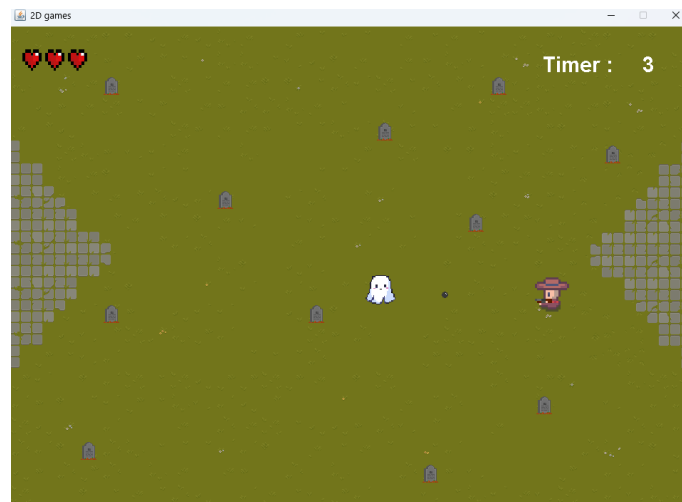
Title ประกอบด้วย Image Background Component, Player Component, String Component, Button Component

2.5.2 หน้าเลือก Map



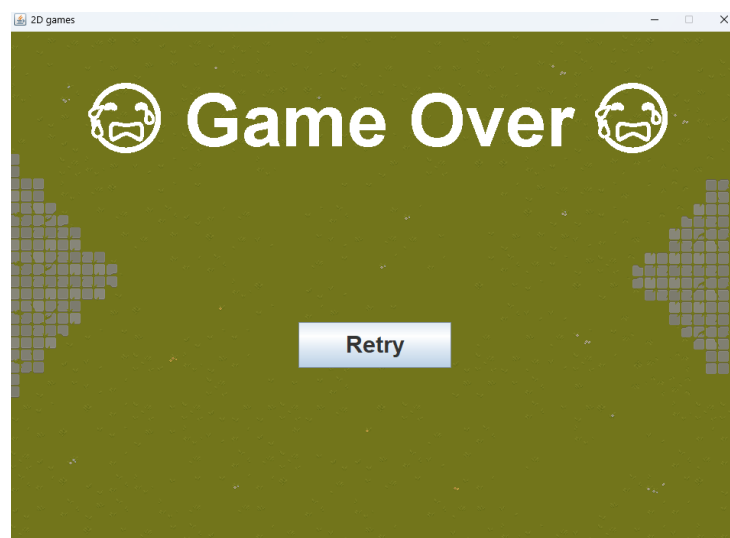
หน้าเลือก Map ประกอบด้วย Image Background Component, 3 Button Component

2.5.3 หน้า Game (The Graveyard 1)



หน้าเกมประกอบไปด้วย Image heart Component, Image Player Component, Image Ghost Component, String Component 2 อัน

2.5.4 หน้า Gameover



หน้า Gameover ประกอบไปด้วย String Component, Button Component

2.6 Event handling

```
1 public class KeyHandler implements KeyListener{
2
3     public boolean upPressed, downPressed, leftPressed, rightPressed;
4
5     @Override
6     public void keyTyped(KeyEvent e) {
7     }
8
9     @Override
10    public void keyPressed(KeyEvent e) {
11
12        int code = e.getKeyCode();
13
14        if(code == KeyEvent.VK_W){
15            upPressed = true;
16        }
17        if(code == KeyEvent.VK_A){
18            leftPressed = true;
19        }
20        if(code == KeyEvent.VK_S){
21            downPressed = true;
22        }
23        if(code == KeyEvent.VK_D){
24            rightPressed = true;
25        }
26    }
27
28    @Override
29    public void keyReleased(KeyEvent e) {
30
31        int code = e.getKeyCode();
32
33        if(code == KeyEvent.VK_W){
34            upPressed = false;
35        }
36        if(code == KeyEvent.VK_A){
37            leftPressed = false;
38        }
39        if(code == KeyEvent.VK_S){
40            downPressed = false;
41        }
42        if(code == KeyEvent.VK_D){
43            rightPressed = false;
44        }
45    }
46 }
47
48 }
```

ดักเมื่อกดปุ่ม w,a,s,d ค่า boolean ของแต่ละปุ่มจะเป็น true แสดงว่ากดอยู่เพื่อนำไปใช้ในคลาส Player เพื่อใช้ในการเคลื่อนที่ของ Player และเมื่อกดปุ่มออกค่า boolean จะเป็น false ทำให้ Player หยุดเคลื่อนที่

```

1  @Override
2      public void mouseMoved(MouseEvent e) {
3          mouseX = e.getX();
4          mouseY = e.getY();
5
6          double angle = Math.atan2(mouseY - y - pixelMC, mouseX - x - pixelMC); // Math.atan2 = find radians
7          if (Math.abs(angle) < Math.PI / 4) { // angle < 45°
8              img = right;
9              direction = "right";
10         } else if (Math.abs(angle) > Math.PI * 3 / 4) { // angle > 135°
11             img = left;
12             direction = "left";
13         } else if (angle < 0) {
14             img = up;
15             direction = "up";
16         } else {
17             img = down;
18             direction = "down";
19         }
20     }
21 }
22
23 @Override
24 public void mouseDragged(MouseEvent e) {
25 }
26
27 @Override
28 public void mouseClicked(MouseEvent e) {
29 }
30 }
31
32 @Override
33 public void mousePressed(MouseEvent e) {
34     long currentTime = System.currentTimeMillis();
35
36     // Check if the cooldown period has passed
37     if (currentTime - lastShotTime >= shootingCooldown) {
38         gp.fireBullet(x + pixelMC, y + pixelMC, direction);
39
40         lastShotTime = currentTime;
41
42         System.out.println("Bullet fired.");
43     }
44 }

```

เป็นการ Override Method mousePresseed(), mouseMoved() ในคลาส Player

ใน Method mousePress() เมื่อกด Click ก็จะทำให้การเรียก Method fireBullet() ใน Gamepanel

เพื่อยิงกระสุนออกไป และใน Method mouseMove() จะทำการเรียก Method getX(), getY()

เพื่อเก็บค่า x, y ของ mouse จากนั้นไปคำนวณมุมระหว่างผู้เล่นกับ mouse เพื่อเช็คทิศทาง

ที่ Player ควรจะหันไปในทิศทางเดียวกับที่ mouse ชี้อยู่

2.7 Algorithm

```
1 public void notcollision(){
2     int playerX = this.getplayerX();
3     int playerY = this.getplayerY();
4     for (Grave grave : g) {
5         int graveX = grave.getX();
6         int graveY = grave.getY();
7         Square graveHitbox = grave.getHitbox();
8         Square playerHitbox = this.getHitbox();
9
10
11         if (playerHitbox.intersects(graveHitbox)) {
12             // System.out.println("Collision detected with grave!");
13             double angle = Math.atan2(graveY - playerY - 32, graveX - playerX - 32);
14
15             if (Math.abs(angle) < Math.PI / 4) { // right
16                 direction_collision[0] = false;
17             } else if (Math.abs(angle) > Math.PI * 3 / 4) { //left
18                 direction_collision[1] = false;
19             } else if (angle < 0) { // up
20                 direction_collision[2] = false;
21             } else { // down
22                 direction_collision[3] = false;
23             }
24         }
25     }
26 }
```

เริ่มด้วยการนำค่า PlayerX, PlayerY จากนั้นทำการวนลูปตามจำนวนของ Grave ใน Map

และเก็บค่า GraveX, GraveY, graveHitbox, playerHitbox จากนั้นก็ทำการเช็คค่า Hitbox

ของทั้งคู่ นั้น intersect กันหรือไม่ ถ้า intersect กันก็ให้คำนวณมุมของ Player กับ Grave

เพื่อดูว่า Player อยู่ด้านไหนของ Grave โดยเมื่ออยู่ด้านใดด้านหนึ่งของ Grave แล้ว

ก็จะทำการเช็คค่า direction_collision ให้เป็น false และนำไปใส่ Method เดินของ Player

เพื่อให้ Player ไม่สามารถเดินไปในด้านใดด้านหนึ่งของ Grave ได้

บทที่ 3 สรุป

ปัญหาที่พบระหว่างการพัฒนา

1. ทำงานไม่ได้ตามแผนที่วางไว้เนื่องจากมีงานวิชาอื่นหลายวิชาทำให้เกมไม่ออกมาดีเท่าที่ควร
2. มีความเข้าใจในการทำงานของ Graphic น้อยจึงทำให้ใช้เวลาในการแก้บัคและทำความเข้าใจนาน
3. การออกแบบ Class และการคิด Algorithm ต่างๆในเกม

จุดเด่น

1. เป็นเกมที่ Player จะหันหน้าโดยมีทิศทางตาม mouse และมีสิ่งกีดขวางไม่ให้ Player สามารถเดินได้สะดวก

ข้อเสนอแนะ

-