



รายงาน

เรื่อง Project OOP (Gundam Battle)

เสนอ

ผู้ช่วยศาสตราจารย์สฤติย์ ประสมพันธ์

จัดทำโดย

นายศุภโชติ ปาลพันธ์

เลขประจำตัวนักศึกษา 6204062620054

คณะ วิทยาศาสตร์ประยุกต์

ภาคเรียนที่ 1 ปีการศึกษา 2564

บทนำ

ที่มาและความสำคัญของโปรเจ็ค

ได้แรงบันดาลใจมาจากการชอบการ์ตูน gundam ตอนเด็กๆ จึงคิดอยากลองทำเกมgundam ที่มีการเล่นผ่านด่านต่างๆไปเรื่อยๆ โดยผมอยากลองเขียนเกมนี้เนื่องจากผมมีความสงสัยว่าในแต่ละด่าน การสร้างมันจะยากขนาดไหนและมีอะไรที่ซับซ้อนบ้าง ผมจึงเลือกที่จะทำเกมแนวผ่านด่านที่ผมคิดผ่านและวิธีการเล่นเอง โดยมีเพื่อนๆคอยช่วยเหลือในส่วนของการคำนวณในแต่ละด่าน และทำมันออกมาจนเสร็จสมบูรณ์

ประเภทของโครงการ

โครงการเกี่ยวกับคอมพิวเตอร์

ประโยชน์

1. ฝึกความอดทน
2. ได้ความสนุกสนาน
3. มีไหวพริบในการแก้ไขปัญหา
4. ฝึกวิธีการเอาตัวรอด

ขอบเขตของโครงการ (Storyboard)

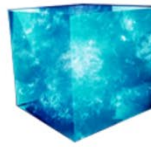
ตัวละคร



ไอเทมในเกม



คั่นโยนสำหรับเปิดประตู



ลูกบาศก์มิติ ใช้เพื่อวาร์ปไปจุดต่างๆ



ประตูผ่านด่าน

ฉากเกม

หน้าแรกของเกม



หน้าอธิบายวิธีการเล่น



ด่านที่ 1



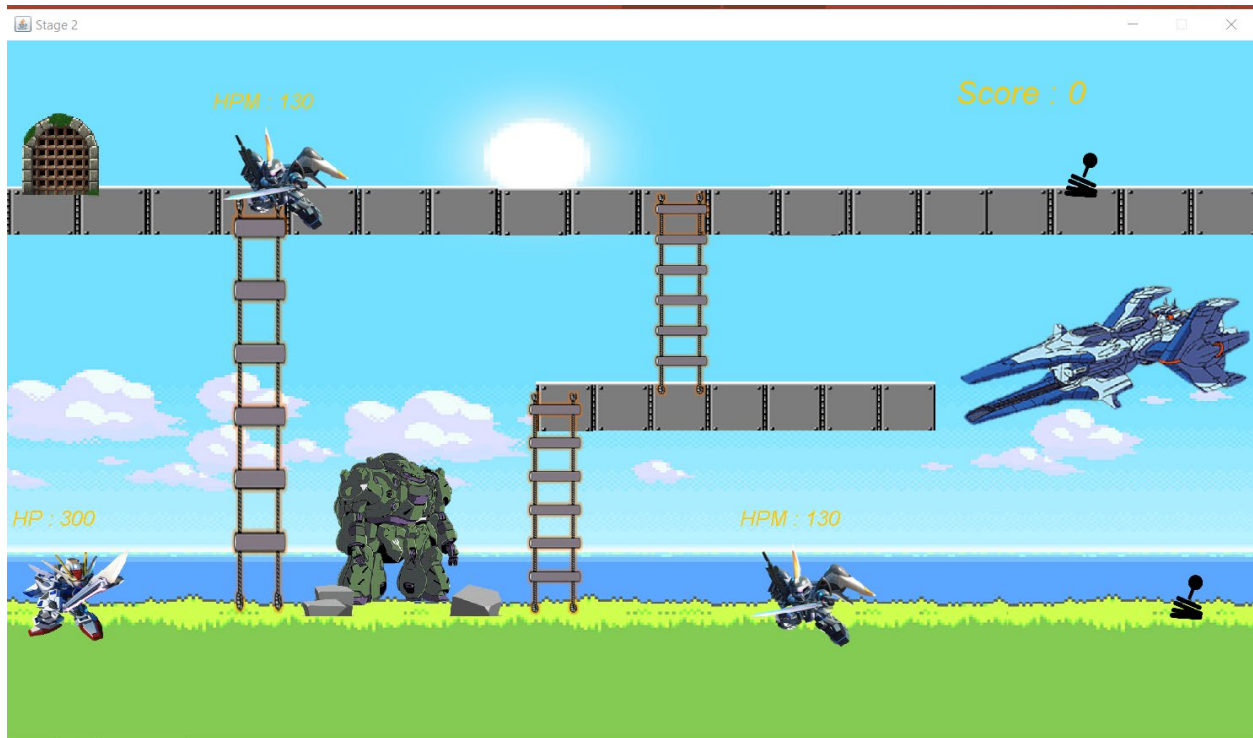
ผ่านด่านที่ 1



แพ่ด่านที่ 1



ด่านที่ 2



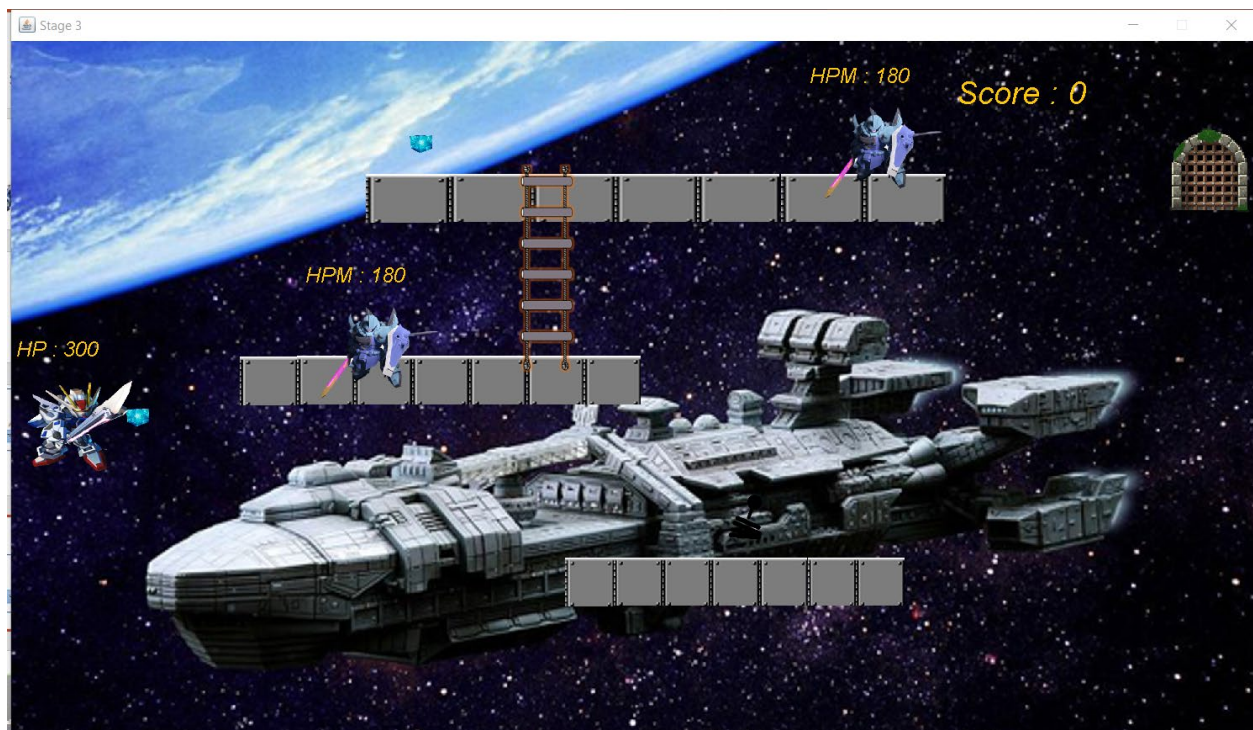
ผ่านด่านที่ 2



แพ่ด่านที่ 2



ด่านที่ 3



ผ่านด่านที่ 3



แพ้ด่านที่ 3



เงื่อนไขการเล่น

- ผู้เล่นต้องไปโยกคั่นโยกของแต่ละด้านเพื่อเปิดประตู
- ในด่านที่ 3 ผู้เล่นต้องทำลาย mobile suit ฝ่ายศัตรู ก่อนแล้วประตูมิติจึงจะปรากฏ
- ผู้เล่นต้องทำลาย mobile suit ฝ่ายศัตรู ในแต่ละด้านแล้วเข้าประตูเพื่อผ่านไปด่านต่อไป

ตารางแผนการทำงานเดือนตุลาคม

ลำดับ	รายการ	1 -8 ต.ค.	9– 22 ต.ค.	23 – 27 ต.ค.
1	หารูปตัวละครและทำกราฟิกต่างๆ			
2	ศึกษาเอกสารและข้อมูลที่เกี่ยวข้อง			
3	ลงมือเขียนโปรแกรม			

ตารางแผนการทำงานเดือนพฤศจิกายน

ลำดับ	รายการ	12 -19 พ.ย.
1	จัดทำเอกสาร	
2	ตรวจสอบและแก้ไขข้อผิดพลาด	
3	ทำวิดีโอประกอบการอธิบาย	

ส่วนการพัฒนา

เนื้อเรื่องย่อ

เกมส์นี้มีชื่อว่า Gundam Battle ซึ่งเป็นเกมเล่นผ่านด่าน โดยผู้เล่นจะทำการบังคับ mobile suit ไปทำภารกิจเปิดประตูเพื่อ ผ่านไปยังด่านอีกแห่ง โดยขณะที่เล่น จะมี mobile suit ของฝ่ายศัตรู มาขวางการทำภารกิจ ผู้เล่นจะต้องทำลาย mobile suit ฝ่ายศัตรูให้หมด เพื่อผ่านอุปสรรคนั้น ไปยังอีกด่าน โดยในแต่ละด่านจะมีไอเทมที่แตกต่างกันออกไป

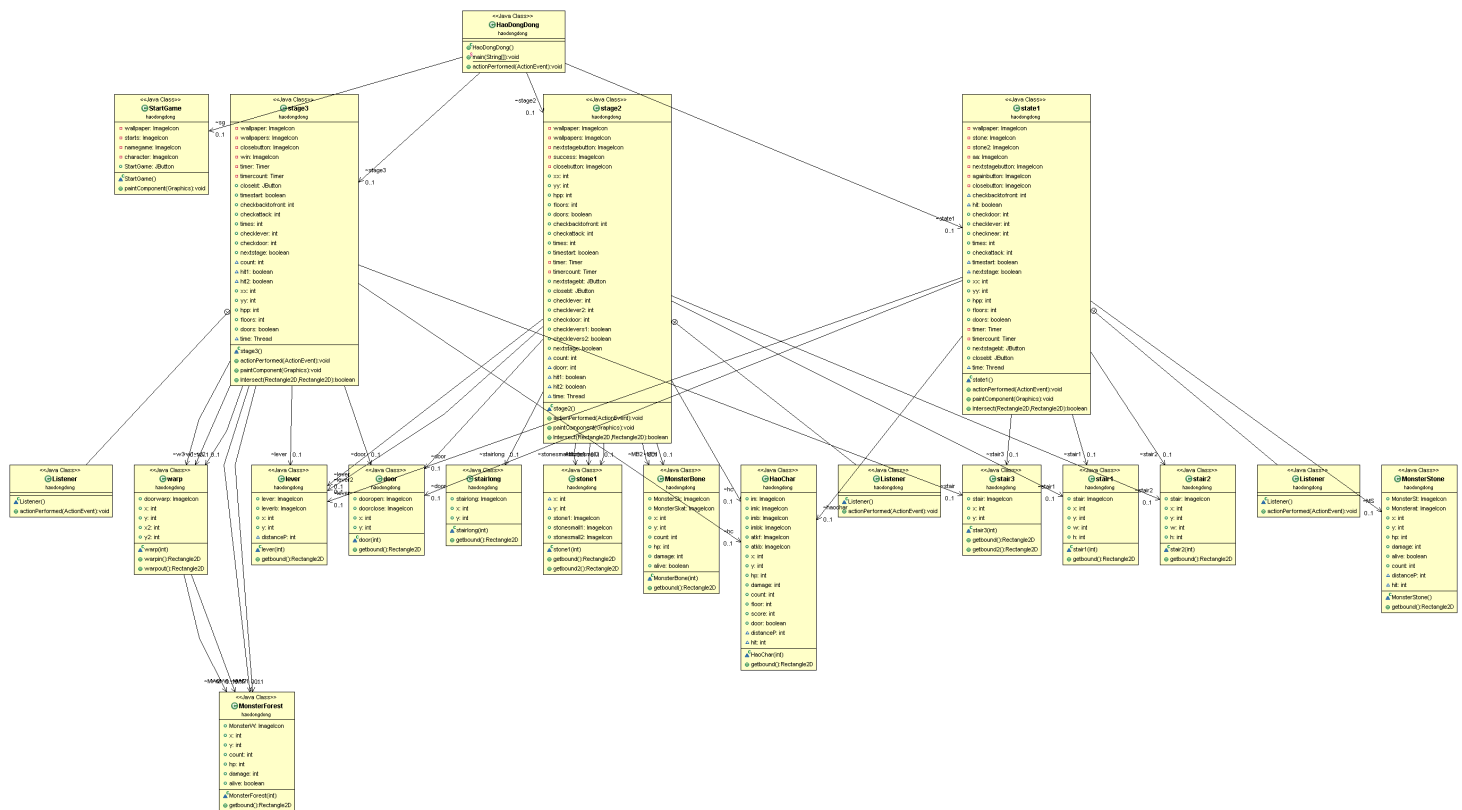
วิธีการเล่น

ใช้ปุ่ม ลูกศรขึ้น-ลง เพื่อบังคับตัวละคร เดินขึ้นหรือเดินลง ส่วน ลูกศรซ้าย-ขวา เพื่อบังคับตัวละคร เดินไปซ้ายหรือขวา

ใช้ปุ่ม Spacebar ในการใช้ดาบโจมตี

ใช้ปุ่ม F ในการโยกคันโยกและเข้าประตู

คลาสไดอะแกรม



การทำงานก็คือน่าจะมีหน้า maingamegundam เป็นหน้า Main โดยการที่เราจะสร้าง class แยกไว้แต่ละหน้าโดยการ สร้าง stage1, stage2, stage3, mobile suit1, mobile suit1, mobile suit1, stair1, stair2, stair3, stairlong, StartGame, Mygundam, lever, door, stone1, warp , howto play , SoundEffect และอื่นๆจากนั้นเราจะทำการสร้าง Object โดยเรียกไปที่แต่ละ Class นั้นๆ เพื่อดึงมาใช้ใน main ส่วนหน้าอื่นๆหากมีความจำเป็นที่จะต้องใช้ class นั้นเราก็จะใช้การ Object และเรียกไปที่ Class นั้นเพื่อเรียกใช้

รูปแบบการพัฒนา

เป็น Application

อธิบายส่วนของโปรแกรมที่มี

Constructor

```
public maingamegundam (){  
    this.setSize(1200,700);  
    this.add(sg);  
    sound.FileName = "Blank.wav";  
    sound.sound();  
    sg.requestFocusInWindow();  
    sg.StartGame.addActionListener(this);  
    state1.closebt.addActionListener(this);  
    stage2.closebt.addActionListener(this);  
    stage3.closebt.addActionListener(this);  
    state1.nextstagebt.addActionListener(this);  
    stage2.nextstagebt.addActionListener(this);  
    sg.HowButton.addActionListener(this);  
    http.backtoentergame.addActionListener(this);  
}
```

เป็นส่วนของ Public class ซึ่งการทำงานจะทำการเซตหน้า JFrame หน้านี้เป็นขนาด 1200x700 จากนั้นให้ทำการ Add หน้าต่าง sg(StartGame) เข้ามา และให้โฟกัสไปที่หน้าต่าง sg(StartGame) และ ทำการดึงไฟล์เสียงเพลงประกอบเข้ามาเล่นโดย ใช้ .WAV ทำการลงทะเบียนปุ่ม ActionListener ให้กับปุ่ม StartGame ในหน้าต่าง sg(StartGame)

ทำการลงทะเบียนปุ่ม ActionListener ให้กับปุ่มกดปิดของหน้าต่างที่ 1

ทำการลงทะเบียนปุ่ม ActionListener ให้กับปุ่มกดปิดของหน้าต่างที่ 1

ทำการลงทะเบียนปุ่ม ActionListener ให้กับปุ่มกดปิดของหน้าต่างที่ 2

ทำการลงทะเบียนปุ่ม ActionListener ให้กับปุ่มกดปิดของหน้าต่างที่ 3

ทำการลงทะเบียนปุ่ม ActionListener ให้กับปุ่มกดผ่านด้านของหน้าต่างที่ 1

ทำการลงทะเบียนปุ่ม ActionListener ให้กับปุ่มกดผ่านด้านของหน้าต่างที่ 1

ทำการลงทะเบียนปุ่ม HowButton ให้กับปุ่มกดเพื่อไปยังหน้าวิธีเล่น

ทำการลงทะเบียนปุ่ม backtoentergame ให้กับปุ่มกดย้อนกลับ


```

state1() {
    timer.start();
    time.start();
    timercount.start();
    this.setFocusable(true);
    this.setLayout(null);

    this.addKeyListener(new KeyAdapter() {
        @Override
        public void keyPressed(KeyEvent e) {
            int a = e.getKeyCode();
            if(a==KeyEvent.VK_RIGHT){
                checkbacktofront = 0;
                if(checkbacktofront==0){
                    if(haochar.floor == 0){
                        if(haochar.x>=1100){
                            haochar.x = 1100;
                        }
                        else if(Intersect(stair1.getbound(),haochar.getbound())){
                            if(haochar.y>=565){
                                haochar.x+=10;
                            }
                        }
                        else if(Intersect(stair3.getbound(),haochar.getbound())){
                            if(haochar.y>=565){
                                haochar.x+=10;
                            }
                        }
                        else{
                            haochar.x+=10;
                            haochar.count++;
                        }
                    }
                }
            }
        }
    });
}

```

ของด่านที่ 1 เป็นการทำให้โฟกัสหน้า Level 1 และสร้าง AddKeyListener เพื่อรองรับการกดปุ่มเข้ามาโดยก็จะมีการทำงานในแต่ละปุ่มทำให้เกิดอะไรขึ้น แล้วก็จะมีการทำงานแบบนี้ไปถึง stage3

Encapsulation

```

public class stage1 extends JPanel implements ActionListener{
    private ImageIcon wallpaper = new ImageIcon(this.getClass().getResource("v1.jpg"));
    private ImageIcon day = new ImageIcon(this.getClass().getResource("day.png"));
    private ImageIcon died = new ImageIcon(this.getClass().getResource("died.png"));
    private ImageIcon stone = new ImageIcon(this.getClass().getResource("stone.png"));
    private ImageIcon stone2 = new ImageIcon(this.getClass().getResource("stone2.png"));
    private ImageIcon aa = new ImageIcon(this.getClass().getResource("aa.png"));
    private ImageIcon nextstagebutton = new ImageIcon(this.getClass().getResource("nextstage.png"));
    private ImageIcon againbutton = new ImageIcon(this.getClass().getResource("click to play again.png")); //agin
    private ImageIcon closebutton = new ImageIcon(this.getClass().getResource("closebt.png"));
}

```

เป็น attribute ที่ถูกห่อหุ้มไว้ใช้ได้เฉพาะในหน้า class ของ stage 1 เท่านั้นทำให้หน้าอื่นไม่สามารถเข้าถึงได้โดยตรง

Composition

```
Mygundam haochar = new Mygundam(1);  
MobileSuit1 MS = new MobileSuit1();  
lever lever = new lever(1);  
door door = new door(1);  
stair1 stair1 = new stair1(1);  
stair2 stair2 = new stair2(1);  
stair3 stair3 = new stair3(1);
```

Polymorphism

NULL

Abstract

NULL

Inheritance

NULL

Thread

```
Thread time = new Thread(new Runnable(){
    public void run(){
        int check=0;
        while(true){
            try{

                if(Intersect(MS.getbound(),haochar.getbound())){
                    MS.count=1;
                }
                else{
                    if(MS.alive == true){
                        if(MS.x==400){
                            check = 0;
                            MS.x+=5;
                        }
                        else if(MS.x==500){
                            check = 1;
                            MS.x-=5;
                        }
                        else{
                            if(check==0){
                                MS.x+=5;
                            }
                        }
                    }
                }
            }
        }
        Thread.sleep(100);
    }catch(Exception e){
    }

    if(timestart == false){
        repaint();
    }
}
});
```

หน้าจอ GUI อธิบายส่วนประกอบของ GUI โครงสร้างของ GUI ประกอบด้วย Component อะไรบ้าง

```
public void paintComponent(Graphics g){
    super.paintComponent(g);
    g.drawImage(wallpaper.getImage(), 0, 0, 1200, 665, this);
    g.drawImage(stone.getImage(), 0, 425, 350, 45, this);
    g.drawImage(stone.getImage(), 900, 380, 350, 45, this);
    g.drawImage(stone2.getImage(), 0, 200, 700, 45, this);
    g.drawImage(stair1.stair.getImage(), stair1.x, stair1.y, 80, 215, this);
    g.drawImage(stair2.stair.getImage(), stair2.x, stair2.y, 80, 240, this);
    g.drawImage(stair3.stair.getImage(), stair3.x, stair3.y, 80, 260, this);
    g.setColor(Color. ORANGE);
    g.setFont(new Font("Fixedsys Regular",Font. HANGING_ BASELINE,30));
    g.drawString("Score : "+ haochar.score,900, 60);
}
```

ในส่วนแรกจะเป็นการวาด wallpaper , stone ,stair จากนั้นจะทำการปรับปากกาเป็นสีส้ม และเปลี่ยน font เป็น Fixedsys Regular โดยขนาด font คือ 30 และให้วาด score โดยดึงค่ามาจากตัวแปร score จาก class HaoChar

```
if(checklever==1){
    g.drawImage(lever.leverb.getImage(), lever.x, lever.y, 40, 45, this);
    g.drawImage(door.dooropen.getImage(), door.x+2, door.y, 82, 80, this);
}
else if(checklever==0){
    g.drawImage(lever.lever.getImage(), lever.x, lever.y, 40, 45, this);
    g.drawImage(door.doorclose.getImage(), door.x, door.y, 86, 80, this);
}
```

ในส่วนนี้จะเช็คว่าการเปิดหรือยัง ถ้ายังไม่ถูกเปิดจะวาด leverปิด กับ doorปิด แต่ถ้าถูกเปิด จะวาด leverเปิด กับ doorเปิด

```

if(checkbacktofront==0){
    if(haochar.count==0){
        if(checkattack == 1){
            g.drawImage(haochar.atkf.getImage(), haochar.x+16, haochar.y, 120, 100, this);
            g.drawString("HP : "+haochar.hp, haochar.x, haochar.y-15);
            checkattack = 0;

            if(MS.hp==0){
                MS.alive = false;
                haochar.score+=100;
                if(haochar.score==100){
                    haochar.score=100;
                }
            }
        }
        else {
            g.drawImage(haochar.im.getImage(), haochar.x, haochar.y, 120, 100, this);
            g.drawString("HP : "+haochar.hp, haochar.x, haochar.y-15);
        }
    }
    else {
        g.drawImage(haochar.imk.getImage(), haochar.x, haochar.y, 120, 100, this);
        g.drawString("HP : "+haochar.hp, haochar.x, haochar.y-15);
    }
}
}
else if(checkbacktofront==1){
    if(haochar.count==0){
        if(checkattack == 1){
            g.drawImage(haochar.atkb.getImage(), haochar.x-41, haochar.y, 120, 100, this);
            g.drawString("HP : "+haochar.hp, haochar.x, haochar.y-15);
            checkattack = 0;
            if(MS.hp==0){
                MS.alive = false;
                haochar.score+=100;
                if(haochar.score==100){
                    haochar.score=100;
                }
            }
        }
        else {
            g.drawImage(haochar.imb.getImage(), haochar.x, haochar.y, 120, 100, this);
            g.drawString("HP : "+haochar.hp, haochar.x, haochar.y-15);
        }
    }
    else{
        g.drawImage(haochar.imbk.getImage(), haochar.x, haochar.y, 120, 100, this);
        g.drawString("HP : "+haochar.hp, haochar.x, haochar.y-15);
    }
}
}
else if(checkbacktofront==2){

```

ในส่วนนี้จะเช็คเดินหน้า ถอยหลัง โจมตี ถ้าเดินหน้าแล้วโจมตีจะวาด ตัวละครโจมตีไปด้านหน้า ถ้าเดินหน้าแล้วไม่ได้โจมตีก็จะวาดตัวละครที่ไม่ได้โจมตี ส่วนเดินถอยหลังแล้วโจมตีก็จะวาด ตัวละครโจมตีไปด้านหลัง ถ้าเดินถอยหลังแล้วไม่ได้โจมตีก็จะวาดตัวละครที่ไม่ได้โจมตี


```

if(MS.count == 1){
    if(MS.alive == true){
        g.drawImage(MS.Monsterat.getImage(), MS.x, MS.y, 120, 100, this);
        g.drawString("HPM : "+MS.hp, MS.x-10, MS.y-15);
        MS.count = 0;
        hit = true;
    }
}
else {
    if(MS.alive == true){
        g.drawImage(MS.MonsterSt.getImage(), MS.x, MS.y, 120, 100, this);
        g.drawString("HPM : "+MS.hp, MS.x-10, MS.y-15);
    }
}

```

ในส่วนนี้จะเช็คว่ามี mobile suit ทำการโจมตีมั้ย ถ้า โจมตีก็จะวาดรูป monster โจมตี ถ้า
ไม่ได้โจมตีก็จะวาดรูป mobile suit ที่ไม่ได้โจมตี

```

if(haochar.hp <= 0){
    this.setLayout(null);
    g.setColor(Color.RED);
    g.drawImage(wallpaper.getImage(), 0, 0, 1200, 800, this);
    g.drawImage(died.getImage(), 100, 100, 1000, 200, this);
    setLayout(null);
    closebt.setBounds(600, 450, 500, 120);
    closebt.setBorderPainted(false);
    closebt.setBorder(null);
    closebt.setOpaque(false);
    closebt.setContentAreaFilled(false);
    add(closebt);

    timer.stop();
    timercount.stop();
}

```

ในส่วนนี้จะเช็คว่ามีตัวละครเลือดเท่ากับ 0 หรือยัง ถ้าเลือดน้อยกว่าหรือเท่ากับ 0 จะวาด
wallpaper แล้ววาดข้อความ You died แล้ว วาดปุ่ม close เพื่อใช้ในการกดปิด

```

    if(nextstage == true){
        this.setLayout(null);
        g.drawImage(wallpaper.getImage(), 0, 0, 1200, 800, this);
        g.drawImage(day.getImage(), 450, 40, 400, 220, this);

        haochar.x = xx;
        haochar.y = yy;
        haochar.hp = hpp;
        haochar.floor = floors;
        haochar.door = doors;
        setLayout(null);
        nextstagebt.setBounds(400, 400, 460, 70);
        nextstagebt.setBorderPainted(false);
        nextstagebt.setBorder(null);
        nextstagebt.setOpaque(false);
        nextstagebt.setContentAreaFilled(false);
        add(nextstagebt);

        timer.stop();
        timercount.stop();
    }
} //drawscreen

```

ในส่วนนี้จะเช็คว่าคุณผ่านด่านหรือยัง ถ้าผ่านด่านแล้วจะวาด wallpaper แล้ววาด day แล้วทำการเช็ทค่าตัวละครใหม่ทั้งหมดเพื่อจะไปใช้ในด้านต่อไป แล้ววาดปุ่ม Nextstage เพื่อกดเปลี่ยนด่าน

อธิบาย Event handling ที่มีในหน้าจอ

```
if(nextstage == true){  
    this.setLayout(null);  
    g.drawImage(wallpaper.getImage(), 0, 0, 1200, 800, this);  
    g.drawImage(day.getImage(), 450, 40, 400, 220, this);  
  
    haochar.x = xx;  
    haochar.y = yy;  
    haochar.hp = hpp;  
    haochar.floor = floors;  
    haochar.door = doors;  
    setLayout(null);  
    nextstagebt.setBounds(400, 400, 460, 70);  
    nextstagebt.setBorderPainted(false);  
    nextstagebt.setBorder(null);  
    nextstagebt.setOpaque(false);  
    nextstagebt.setContentAreaFilled(false);  
    add(nextstagebt);  
  
    timer.stop();  
    timercount.stop();  
}
```

โดยในส่วนนี้จะมีการวาดปุ่ม NextLevel ซึ่งได้ทำการ addActionListener ไว้ที่ Constructor เรียบร้อยแล้ว หากเราทำการกดปุ่มนี้ก็จะเกิดการ Event handling ให้ไปดำเนินต่อไปทันที

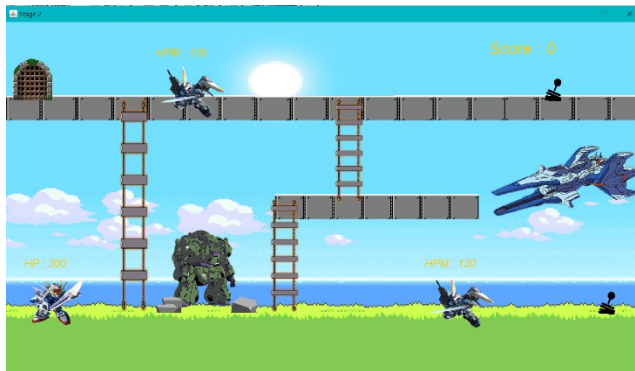



```

this.addKeyListener(new KeyAdapter() {
    @Override
    public void keyPressed(KeyEvent e) {
        int a = e.getKeyCode();
        if (a == KeyEvent.VK_RIGHT) {
            checkbacktofront = 0;
            if (checkbacktofront == 0) {
                if (haochar.floor == 0) {
                    if (haochar.x >= 1100) {
                        haochar.x = 1100;
                    }
                    else if (Intersect(stair1.getbound(), haochar.getbound())) {
                        if (haochar.y >= 565) {
                            haochar.x += 10;
                        }
                    }
                    else if (Intersect(stair3.getbound(), haochar.getbound())) {
                        if (haochar.y >= 565) {
                            haochar.x += 10;
                        }
                    }
                }
                else {
                    haochar.x += 10;
                    haochar.count++;
                }
            }
            else if (haochar.floor == 1) {
                if (haochar.x >= 290) {
                    haochar.x = 290;
                }
                else if (Intersect(stair1.getbound(), haochar.getbound())) {
                    if (haochar.y <= 355) {
                        haochar.x += 10;
                    }
                }
            }
        }
    }
});

```

ตรงนี้จะเป็นส่วนของการเดินโดยหากเรากดปุ่ม ลูกศรขวา ก็จะเกิดการ Event handling โดยให้ตัวละครของเราขยับพิกัด x ไปทางขวา



```

else if(a==KeyEvent.VK_LEFT){
    checkbacktofront = 1;
    if(haochar.floor==0){
        if(haochar.x <= 0){
            haochar.x = 0;
        }
        else if(Intersect(stair1.getbound(),haochar.getbound())){
            if(haochar.y >= 565){
                haochar.x-=10;
            }
        }
        else if(Intersect(stair3.getbound(),haochar.getbound())){
            if(haochar.y >= 565){
                haochar.x-=10;
            }
        }
        else{
            haochar.x-=10;
            haochar.count++;
        }
    }
    else if(haochar.floor==1){
        if(haochar.x <= 0){
            haochar.x = 0;
        }
        else if(Intersect(stair1.getbound(),haochar.getbound())){
            if(haochar.y <= 355){
                haochar.x-=10;
            }
        }
        else if(Intersect(stair2.getbound(),haochar.getbound())){
            if(haochar.y >= 355){
                haochar.x-=10;
            }
        }
        else{

```

ตรงนี้จะเป็นส่วนของการเดินโดยหากเราดปุ่ม ลูกศรซ้าย ก็จะมีการ Event handling โดยให้ตัวละครของเราขยับพิกัด x ไปทางซ้าย



```

else if(a == KeyEvent.VK_SPACE){
    if(checkattack == 0){
        checkattack = 1;
    }
    if(Intersect(MS.getbound(),haochar.getbound())){
        if(MS.hp==0){
            MS.hp=0;
        }
        MS.hp-=haochar.damage;
    }
}
}

```

ตรงนี้เป็นในส่วนของการโจมตีโดยหากเรากดปุ่ม spacebar ก็จะมีการ Event handling โดยจะสร้างตัวเช็คให้วาดรูปโจมตี

```

else if(a == KeyEvent.VK_F){

    if(Intersect(lever.getbound(),haochar.getbound())){
        checklever = 1;
        checkdoor = 1;
    }
    if(Intersect(door.getbound(),haochar.getbound())){
        if(checkdoor == 1){
            nextstage = true;
        }
        else nextstage = false;
    }
    else nextstage = false;
}
}

```

ตรงนี้เป็นในส่วนของการโจมตีโดยหากเรากดปุ่ม F ก็จะมีการ Event handling โดยถ้าเข้าใกล้คั่นโยกแล้วกดจะโยกคั่นโยกแล้วประตูจะเปิด ถ้าเข้าใกล้ประตูที่เปิดแล้วกดจะผ่านด่าน


```

else if(a==KeyEvent.VK_UP){
    if(haochar.floor==0){
        if(Intersect(stair1.getbound(),haochar.getbound())){
            if(haochar.y<=355){
                haochar.y=355;
                haochar.floor=1;
            }
            else{
                haochar.y-=10;
            }
        }
        else if(Intersect(stair3.getbound(),haochar.getbound())){
            if(haochar.y<=315){
                haochar.y=310;
                haochar.floor=2;
            }
            else{
                haochar.y-=10;
            }
        }
    }
    else if(haochar.floor==1){
        if(Intersect(stair2.getbound(),haochar.getbound())){
            if(haochar.y<=135){
                haochar.y = 130;
                haochar.floor=3;
            }
            else{
                haochar.y-=10;
            }
        }
    }
}
}
}

```

ตรงนี้จะเป็นส่วนของการเดินโดยหากเรากดปุ่ม ลูกศรขึ้น ก็จะมีการ Event handling โดยให้ตัวละครของเราขยับพิกัด Y ไปด้วย

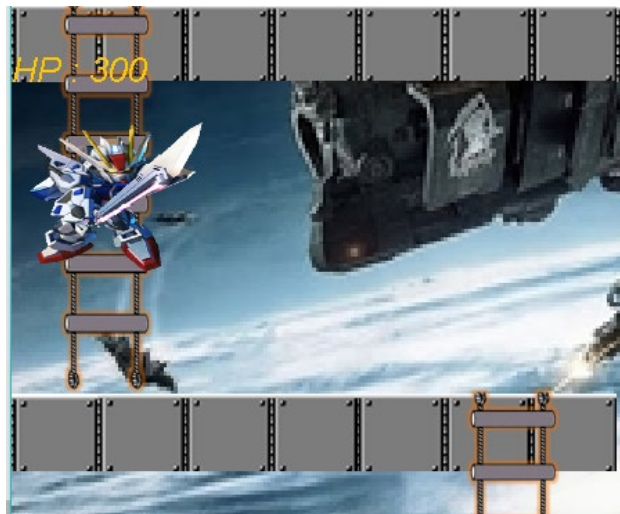


```

else if(a==KeyEvent.VK_DOWN){
    if(haochar.floor==1){
        if(Intersect(stair1.getbound(),haochar.getbound())){
            if(haochar.y>=565){
                haochar.y=565;
                haochar.floor=0;
            }
            else{
                haochar.y+=10;
            }
        }
    }
    else if(haochar.floor==3){
        if(Intersect(stair2.getbound(),haochar.getbound())){
            if(haochar.y>=355){
                haochar.y=355;
                haochar.floor=1;
            }
            else{
                haochar.y+=10;
            }
        }
    }
    else if(haochar.floor==2){
        if(Intersect(stair3.getbound(),haochar.getbound())){
            if(haochar.y>=565){
                haochar.y=565;
                haochar.floor=0;
            }
            else{
                haochar.y+=10;
            }
        }
    }
}
}
}

```

ตรงนี้เป็นในส่วนของการเดินโดยหากเราดปุ่ม ลูกศรซ้าย ก็จะมีการ Event handling โดยให้ตัวละครของเราขึ้นพิคัด Y ไปด้านล่าง



อธิบายอัลกอริทึมที่สำคัญในโปรแกรม

```
public boolean Intersect(Rectangle2D a, Rectangle2D b){  
    return (a.intersects(b));  
}
```

```
public Rectangle2D getbound(){  
    return (new Rectangle2D.Double(x,y,75,70));  
}
```

ในส่วนของอัลกอริทึมที่สำคัญมี 2 ส่วน คือ 1.การเช็ค Intersect 2.การสร้างกรอบ 2D

ตัวอย่าง Intersect

```
if(Intersect(door.getbound(),haochar.getbound())){  
    if(checkdoor == 1){  
        nextstage = true;  
    }  
    else nextstage = false;  
}
```

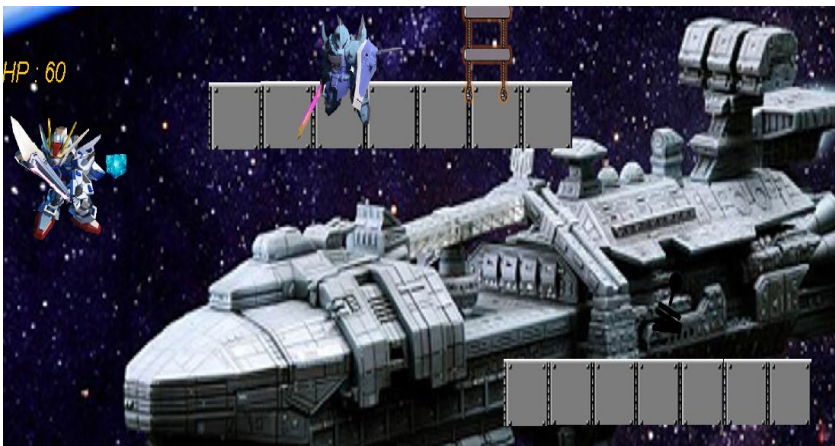
เมื่อตัวละคร Intersect ประตุ แล้วกด F ก็จะสามารถผ่านด่าน

สรุป

ปัญหาที่พบระหว่างพัฒนา

ปัญหาในการทำมีอยู่ 2 อย่างคือ อย่างแรก เกมไม่สามารถย้อนกลับไปหน้าเริ่มเกมได้เนื่องจากThread ที่ใช้ไม่สามารถ stop ได้ และ อย่างที่สองคือเมื่อมีการเพิ่มขนาดของตัวละคร ให้ใหญ่ขึ้น มากกว่า 120x100 จะเกิดBug ทำให้ตัวละครค้าง หรือเดินในตำแหน่งที่ผิด

จุดเด่นของโปรแกรมที่ไม่เหมือนใคร



มีการเก็บลูกบาศก์มิติ เพื่อวาบไปยังจุดต่างๆ ภายในด่าน เพื่อเพิ่มความสนุก

คำแนะนำสำหรับผู้สอนที่อยากให้อธิบาย หรือที่เรียนแล้วไม่เข้าใจ หรืออยากให้เพิ่มสำหรับน้อง ๆ รุ่นต่อไป

สำหรับคำแนะนำให้ผู้สอน

อาจารย์สอนดีอยู่แล้ว สอนเข้าใจและนำไปใช้ประโยชน์ได้จริง แต่เสียดายที่ไม่ได้เรียนในห้อง เพราะเรียนในห้องคงจะสนุกกว่านี้