

โครงการ



เรื่อง เกมสเต็มกริมทาง
(Steak Street Food)

จัดทำโดย

6204062610024 นางสาวณศรา ห่อจันทร์

เสนอ

ผู้ช่วยศาสตราจารย์สถิต ประสมพันธ์

วิชา Object Oriented Programming

ภาคเรียนที่ 1/2565

ภาควิชาวิทยาการคอมพิวเตอร์และสารสนเทศ คณะวิทยาศาสตร์ประยุกต์
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

บทที่ 1

ที่มาและความสำคัญของโครงการ

โครงการนี้จัดขึ้นเพื่อวัดผลความสามารถในการเรียนวิชา Object Oriented Programming โดยการนำเรื่องที่เรียนมาสร้างเป็นชิ้นงานในรูปแบบเกม ผู้จัดทำได้สร้างเกมนี้ขึ้นมา เพื่อให้ผู้เล่นในทุก ๆ ช่วงวัยสามารถเล่นได้ ไม่มีความซับซ้อนมากเกินไป

จึงนำมาสู่แนวคิดการสร้างเกมที่สามารถทำความเข้าใจได้ง่าย เล่นง่าย และใช้ภาษาจาวาในการพัฒนา อีกทั้งยังใช้ความรู้ที่เรียนมาช่วยประยุกต์ใช้ได้ โดยผู้เล่นจะได้สัมผัสประสบการณ์การเป็นเจ้าของร้านอาหารริมทางที่แสนสนุกได้ผ่านเกมนี้

ประเภทโครงการ :

- โครงการพัฒนาเกม

ประโยชน์ :

- 1.ฝึกการวางแผนการทำงานเพื่อให้ทันเวลา
- 2.เพื่อความสนุก
- 3.เพื่อเป็นสื่อการศึกษาการพัฒนาเกมภาษาจาวา

ขอบเขตโครงการ :

- 1.สามารถพัฒนาเกมจากภาษาจาวาได้
- 2.ผู้เล่นสามารถเลือกเล่นเกมได้มากกว่า 1 ด้าน(level)
- 3.ในแต่ละด้าน(level) ของเกม ผู้เล่นสามารถชนะได้ก็ต่อเมื่อทำอาหารครบภายในเวลาที่กำหนด

ตารางแผนการทำงานเดือนตุลาคม :

ลำดับ	รายการ	9-16	17-23	24-30
1.	ทำกราฟิกอาหารและพื้นหลังต่างๆ			
2.	ศึกษาเอกสารและข้อมูลที่เกี่ยวข้อง			
3.	ลงมือเขียนโปรแกรม			
4.	จัดทำเอกสาร			
5.	ตรวจสอบและแก้ไขข้อผิดพลาด			

บทที่ 2

ส่วนการพัฒนา

เนื้อเรื่องย่อ :

คุณคือเจ้าของร้านสเต็กริมทางที่จะต้องรับหน้าที่ย่างสเต็กและเสิร์ฟสลัดให้กับลูกค้าภายในเวลาที่กำหนด โดยคุณจะต้องขายสเต็กและสลัดให้ได้ครบตามจำนวนที่กำหนดไว้ในแต่ละด่าน เพื่อที่จะพัฒนาขยายร้านเพิ่มเมนูของคุณให้ได้มากยิ่งขึ้น

วิธีการเล่น :

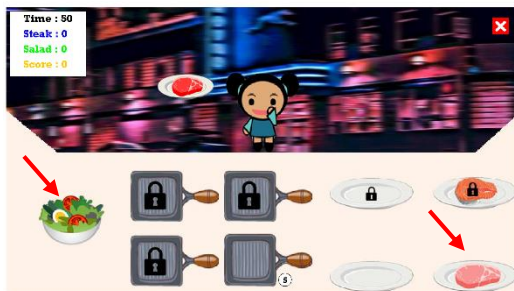
1. เริ่มจาก กดปุ่ม “START” เพื่อเข้าสู่เกม



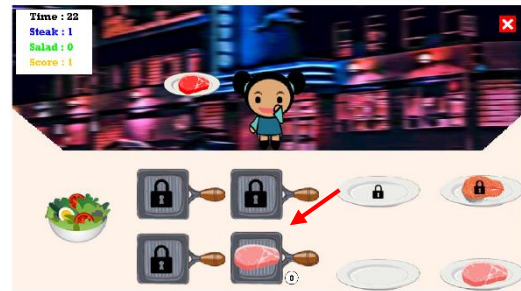
2. เลือกด่าน(Level)ที่ต้องการเล่น ซึ่งแต่ละด่านจะมีภารกิจและเวลาที่แตกต่างกัน



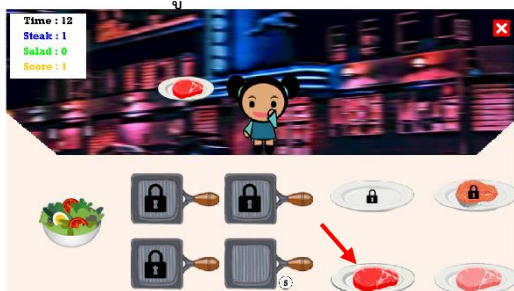
3. เริ่มทำสเต็กด้วยการคลิกที่วัตถุดิบทางด้านขวาหรือคลิกที่จานสลัดถ้าลูกค้าสั่งสลัด



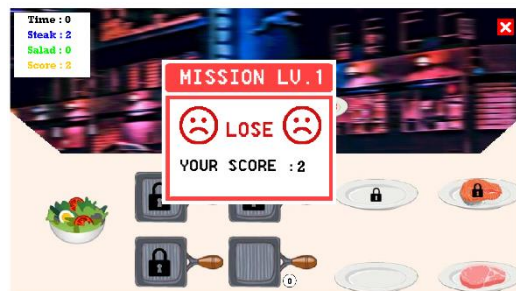
4. รอสเต็กสุก 5 วินาทีแล้วคลิกสเต็กที่สุกแล้วเพื่อเอาออกจากกระทะ



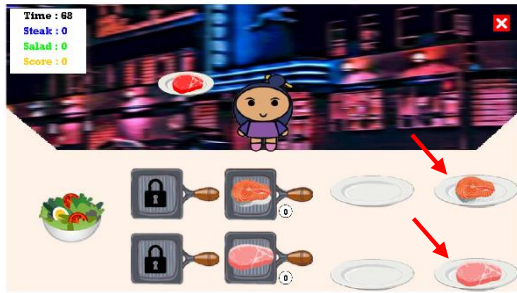
5. คลิกสเต็กที่สุกแล้วเสิร์ฟให้ลูกค้า หากเสิร์ฟผิดสเต็กชิ้นนั้นจะถูกทิ้งทันที



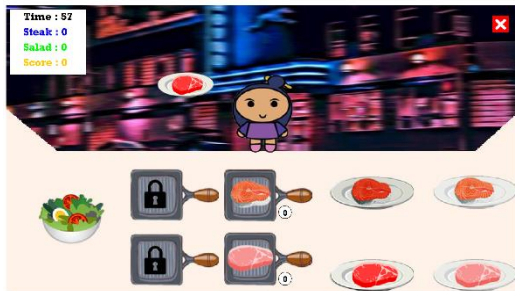
6. หากทำภารกิจไม่ครบภายในเวลาที่กำหนด คุณจะแพ้ทันที



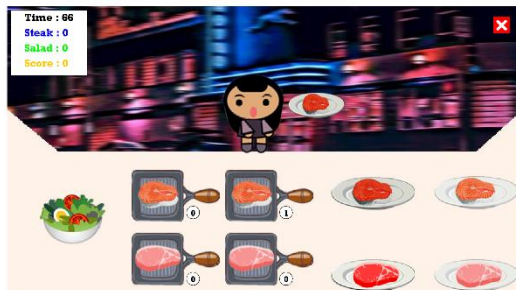
7. ในด่านที่2จะมีวัตถุดิบและกระทะเพิ่มขึ้น โดยทำการคลิกวัตถุดิบทางขวาเพื่อย่างสเต็ก



8. สามารถทำสเต็กเพื่อรอเสิร์ฟลูกค้าได้ แต่ถ้าหากคลิกเสิร์ฟผิดสเต็กจะถูกทิ้งทันที



9. ด่านที่3 ทำหายที่สุดเพราะจะมีเวลาให้เท่ากับด่านที่2 แต่ต้องเสิร์ฟสเต็กให้ได้มากขึ้น ผู้เล่นจะต้องบริหารร้าน จัดการเวลาเพื่อเสิร์ฟสเต็กให้ครบ

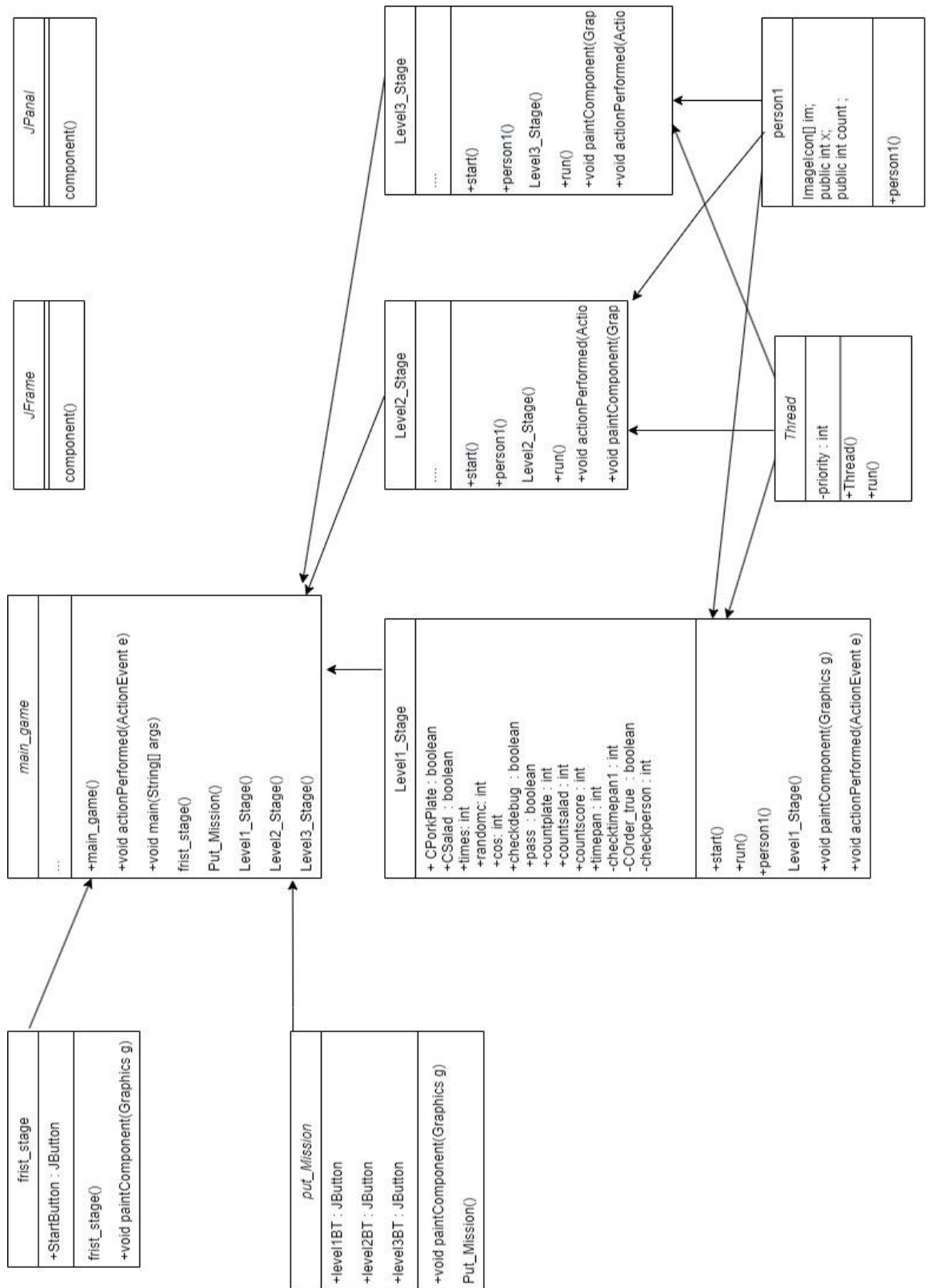


10. หากทำภารกิจในแต่ละด่านครบตามที่กำหนด จะแสดงหน้าต่าง MISSION COMPLETE ซึ่งหมายความว่าผู้เล่นทำภารกิจสำเร็จ



คลาสไดอะแกรม : (ต่อ)

คลาสไดอะแกรม : (ต่อ)



รูปแบบการพัฒนา :

1. สร้างคลาสในแต่ละด้านของเกม
2. สร้างคลาสลูกค้า
3. เพิ่มฟังก์ชันสำหรับการแสดงรูปภาพวัตถุดิบ สเต็ก กระทั่ง และสลัด และการตรวจสอบสเต็กว่าเสิร์ฟตรงตามความต้องการของลูกค้าหรือไม่
4. เพิ่มการตกแต่งให้เกมมีความสวยงามมากยิ่งขึ้นด้วยการแสดงคะแนน และการนับจำนวนอาหารที่เสิร์ฟ
5. เพิ่มอัลกอริทึมในการสุ่มลูกค้าและเมนูอาหารที่สั่ง
6. สร้างคลาสหลัก(mainclass)เพื่อรันโปรแกรม
7. พัฒนาคลาสแต่ละด้านให้ถูกต้องครบถ้วน

อธิบายส่วนของโปรแกรมที่เกี่ยวข้องกับหลักการOOP :

```
public class main_game extends JFrame implements ActionListener{

    frist_stage enterGame = new frist_stage();
    Put_Mission enterLevel = new Put_Mission();
    Level1_Stage Level1Stage = new Level1_Stage();
    Level2_Stage Level2Stage = new Level2_Stage();
    Level3_Stage Level3Stage = new Level3_Stage();

    public main_game(){
        this.setSize(1200,700);
        this.add(enterGame);
        enterGame.requestFocusInWindow();
        enterGame.StartButton.addActionListener(this);
        enterLevel.level1BT.addActionListener(this);
        enterLevel.level2BT.addActionListener(this);
        enterLevel.level3BT.addActionListener(this);

    public static void main(String[] args){
        JFrame gui = new main_game();
        gui.setSize(1200,700);
        gui.setVisible(true);
        gui.setTitle("Steak Street Food");
        gui.setDefaultCloseOperation(EXIT_ON_CLOSE);
        gui.setLocationRelativeTo(null);
        gui.setResizable(false);
    }
}
```

- **Constructor** : ใน class main_game มีการใช้ constructor main_game();

```
public class frist_stage extends JPanel{
    private ImageIcon wallpaper = new ImageIcon(this.getClass().getResource("PlayBG.jpg"));
    private ImageIcon starts = new ImageIcon(this.getClass().getResource("PlayButton.png"));
    public JButton StartButton = new JButton(starts);

    frist_stage(){
        setLayout(null);
        StartButton.setBounds(350, 480, 450, 150);
        StartButton.setBorderPainted(false);
        StartButton.setBorder(null);
        StartButton.setOpaque(false);
        StartButton.setContentAreaFilled(false);
        add(StartButton);
    }
}
```

- **Encapsulation** : ใน class frist_stage มีการห่อหุ้มตัวแปร (กำหนดการเข้าถึงตัวแปร) แบบ private และ public


```

public class Level1_Stage extends JPanel implements ActionListener {

    Person1 person = new Person1();

    if(checkperson == 1){
        g.drawImage(PorkSteak.getImage(), 350, 155, 130, 60, this);
        g.drawImage(person.im[0].getImage(), 500, 140, 180, 200, this);
        CPorkPlate=true;
    }
}

```

- **Composition** : มีการสร้าง object ของ class Person1 และใน class Level1_Stage ก็ประกอบด้วย object ของ class Person1 มีการเรียกใช้งาน method constructor ของ Person1

```

public class frist_stage extends JPanel{
    private ImageIcon wallpaper = new ImageIcon(this.getClass().getResource("PlayBG.jpg"));
    private ImageIcon starts = new ImageIcon(this.getClass().getResource("PlayButton.png"));
    public JButton StartButton = new JButton(starts);
}

```

- **Polymorphism** : class frist_stage มีการสืบทอดมาจาก class JPanel

```

public void paintComponent(Graphics g){
    super.paintComponent(g);
    g.drawImage(wallpaper.getImage(), 0, 0, 1200, 665, this);
}

```

- **Abstract** : มีการเรียกใช้ public abstract class Graphics

```

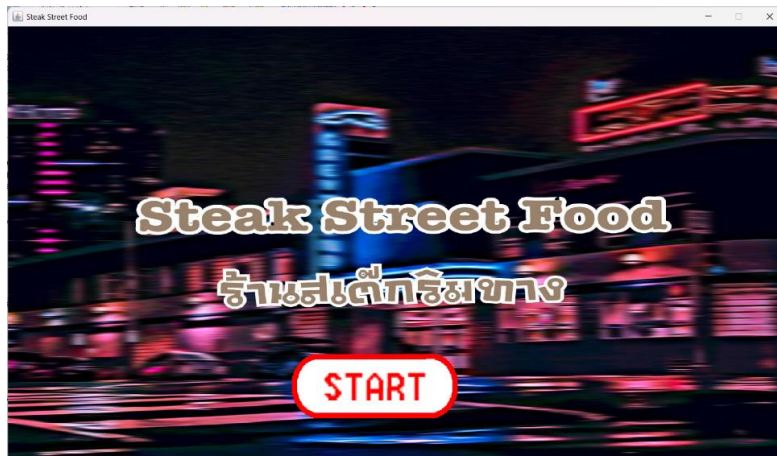
Thread timecount = new Thread(new Runnable(){
    public void run() {
        while(true){
            try{
                times--;
                if(checkpan1 == true && checktimepan1 > 0){
                    checktimepan1--;
                }

                Thread.sleep(1000);
            }catch(InterruptedException e){
            }
            if(checkdebug == true){
                repaint();
            }
        }
    }
}

```

- **Inheritance** : มีการสืบทอดสมบัติและเมธอดจาก class Thread

หน้าจอ GUI:



หน้าจอ GUI แรกของเกม เริ่ม จากคลาส main_game ประกอบด้วย component

- JFrame
- setSize(1200,700); ตั้งขนาดของกรอบวินโดว์
- setDefaultCloseOperation (EXIT_ON_CLOSE) กดกากบาทโปรแกรมก็จะปิดการทำงาน
- setVisible(true); คือทำให้หน้าจอวินโดว์นั้นมองเห็นใช้ได้
- setTitle("Steak Street Food") คือกำหนดหัวข้อด้านบนจอวินโดว์
- add(entergame) ; คือแสดงหน้าจอของคลาส frist_stage
- JButton ในคลาส frist_stage คือการกำหนดปุ่มกด START

```
public class main_game extends JFrame implements ActionListener{

    frist_stage enterGame = new frist_stage();
    Put_Mission enterLevel = new Put_Mission();
    Level1_Stage Level1Stage = new Level1_Stage();
    Level2_Stage Level2Stage = new Level2_Stage();
    Level3_Stage Level3Stage = new Level3_Stage();

    public main_game(){
        this.setSize(1200,700);
        this.add(enterGame);
        enterGame.requestFocusInWindow();
        enterGame.StartButton.addActionListener(this);
        enterLevel.level1BT.addActionListener(this);
        enterLevel.level2BT.addActionListener(this);
        enterLevel.level3BT.addActionListener(this);
    }

    public static void main(String[] args){
        JFrame gui = new main_game();
        gui.setSize(1200,700);
        gui.setVisible(true);
        gui.setTitle("Steak Street Food");
        gui.setDefaultCloseOperation(EXIT_ON_CLOSE);
        gui.setLocationRelativeTo(null);
        gui.setResizable(false);
    }
}
```


อธิบาย Event handling ที่มีในหน้าจอ :



มี `ActionEvent` สำหรับจัดการกับการคลิกโดยใช้คำสั่ง `JButton` และอินเตอร์เฟซ `ActionListener`

```
public main_game() {  
    this.setSize(1200,700);  
    this.add(enterGame);  
    enterGame.requestFocusInWindow();  
    enterGame.StartButton.addActionListener(this);  
    enterLevel.level1BT.addActionListener(this);  
    enterLevel.level2BT.addActionListener(this);  
    enterLevel.level3BT.addActionListener(this);  
  
    Level1Stage.BackButton.addActionListener(this);  
  
    Level2Stage.BackButton.addActionListener(this);  
  
    Level3Stage.BackButton.addActionListener(this);  
}
```

```
public class Put_Mission extends JPanel {  
  
    private ImageIcon bglevel = new ImageIcon(this.getClass().getResource("BackgroundLevel.jpg"));  
    private ImageIcon level1 = new ImageIcon(this.getClass().getResource("level1.png"));  
    private ImageIcon level2 = new ImageIcon(this.getClass().getResource("level2.png"));  
    private ImageIcon level3 = new ImageIcon(this.getClass().getResource("level3.png"));  
  
    public JButton level1BT = new JButton(level1);  
    public JButton level2BT = new JButton(level2);  
    public JButton level3BT = new JButton(level3);  
  
    Put_Mission() {  
  
        setLayout(null);  
        level1BT.setBounds(123, 360, 177,175);  
        level1BT.setBorderPainted(false);  
        level1BT.setBorder(null);  
        level1BT.setOpaque(false);  
        level1BT.setContentAreaFilled(false);  
        add(level1BT);  

```

อธิบายอัลกอริทึมที่สำคัญในโปรแกรม :

```
//*****//
//*****put Pork on the pan*****//
//*****//
if(e.getSource() == PorkBT){
    if(checkpan1 == false && checkpan3 == false){
        this.remove(Pan1BT);
        setLayout(null);
        PorkPan1BT.setBounds(460,490, 292, 187);
        PorkPan1BT.setBorderPainted(false);
        PorkPan1BT.setBorder(null);
        PorkPan1BT.setOpaque(false);
        PorkPan1BT.setContentAreaFilled(false);
        add(PorkPan1BT);
        PorkPan1BT.addActionListener(this);
        checkpan1 = true;
    }
    else if(checkpan1 == true && checkpan3 == false){
        this.remove(Pan3BT);
        setLayout(null);
        PorkPan2BT.setBounds(244,490, 292, 187);
        PorkPan2BT.setBorderPainted(false);
        PorkPan2BT.setBorder(null);
        PorkPan2BT.setOpaque(false);
        PorkPan2BT.setContentAreaFilled(false);
        add(PorkPan2BT);
        PorkPan2BT.addActionListener(this);
        checkpan3 = true;
    }
    else if(checkpan1 == false && checkpan3 == true){
        this.remove(Pan1BT);
        setLayout(null);
        PorkPan1BT.setBounds(460,490, 292, 187);
        PorkPan1BT.setBorderPainted(false);
        PorkPan1BT.setBorder(null);
        PorkPan1BT.setOpaque(false);
        PorkPan1BT.setContentAreaFilled(false);
        add(PorkPan1BT);
        PorkPan1BT.addActionListener(this);
        checkpan1 = true;
    }
}
```

บทที่ 3

สรุป

ปัญหาที่พบระหว่างการพัฒนา

- 1.ในตอนแรกตั้งใจที่จะให้มีวัตถุติดในร้านสเต็กมากขึ้นกว่านี้ แต่พบปัญหาในการเขียนโปรแกรมจึงต้องลดวัตถุติดลงและเพิ่มด่าน(level) เพื่อให้ผู้เล่นได้เล่นหลากหลายมากขึ้น
- 2.ภาพตัวละคร และภาพอื่น ๆ ไม่ชัดเล็กน้อยบางภาพ เนื่องจากขั้นตอนการวาดเริ่มก่อนที่จะเขียนโปรแกรมทำให้การวาดรูปออกมาค่อนข้างมีขนาดเล็ก
- 3.เมื่อเล่นภารกิจในแต่ละด่านจบแล้วจะไม่สามารถกลับไปเล่นซ้ำอีกได้
- 4.สามารถสร้างคลาสObjectเพิ่มได้อีกเพื่อลดขนาดของโปรแกรม/ลดการเขียนซ้ำ แต่เนื่องจากจัดการเวลาผิดพลาดจึงไม่สามารถทำได้ จึงแก้ไขปัญหาให้ดีที่สุดเพื่อให้เกมสามารถเล่นได้สนุกสนาน

จุดเด่นของโปรแกรมที่ไม่เหมือนใคร

- 1.มีภารกิจให้ทำหลายแบบ แล้วแต่ผู้เล่นจะเลือก
- 2.ฝึกการบริหารร้านอาหาร และบริหารเวลาให้ทันเพราะการจะเสิร์ฟสเต็กต้องรออย่างให้สุก5วินาที หากจัดการไม่ดีเวลาอาจจะหมดก่อนทำภารกิจครบ
- 3.เล่นง่ายไม่ซับซ้อน ไม่ต้องใช้ความจำ ไม่ต้องทำความเข้าใจยุ่งยาก

คำแนะนำสำหรับผู้สอน

อยากให้เพิ่มสัดส่วนคะแนนโปรเจกต์ให้มากกว่านี้ และลดสัดส่วนคะแนนสอบปลายภาค ถ้าสามารถทำได้ ส่วนเรื่องการสอนและเนื้อหาในรายชั่วโมงอาจารย์สอนละเอียดและการทำLabในคาบเรียนเหมาะสมกับเนื้อหาดีมากค่ะ