



Object-oriented Programming mini project

“SHOOT IT!”

จัดทำโดย

6504062630138 ธนิตศักดิ์ สกุลรัศมิ์หิรัญ

เสนอ

ผู้ช่วยศาสตราจารย์สถิต ประสมพันธ์

วิชาการเขียนโปรแกรมเชิงวัตถุ (Object-oriented Programming)
ภาควิชาวิทยาการคอมพิวเตอร์และสารสนเทศ คณะวิทยาศาสตร์ประยุกต์

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

ภาคเรียนที่ 1/2022

สารบัญ

หัวข้อ	หน้า
บทที่ 1 บทนำ	
1.1 ที่มาและความสำคัญของโปรเจค	1
1.2 ประโยชน์	1
1.3 ขอบเขตของโครงการ	1
บทที่ 2 ส่วนของการพัฒนา	
2.1 Storyboardและวิธีการเล่น	3
2.2 class diagram	7
2.2 คลาสที่มีการใช้คุณสมบัติของ OOP	8
2.3 ส่วนการทำงานของโปรแกรม	11
2.4 Event Handling ในหน้าจอ	12
2.5 อัลกอริทึมที่สำคัญEnemy Generation Algorithm:	13
บทที่ 3 สรุป	
3.1 ปัญหาที่พบระหว่างการพัฒนา	14
3.2 จุดเด่นที่ไม่เหมือนใคร	14
3.2 คำแนะนำอาจารย์	14

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญ

โครงการนี้ถูกจัดทำขึ้นเพื่อส่งงานวิชา Object-oriented Programming เพื่อวัดผลความรู้ที่ได้จากการเรียน วิชา Object-oriented Programming ว่าสามารถนำมาใช้งานได้จริง ทางปฏิบัติ ในรูปแบบของเกม ที่มีเนื้อหาและ concept OOP

1.2 ประโยชน์

1.2.1 เพื่อความสนุกสนาน เพลิดเพลิน

1.2.2 ฝึกทักษะการตัดสินใจ กระบวนการทางความคิดว่าจะเลือกกำจัดศัตรูตัวไหนก่อน

1.2.3 ฝึกไหวพริบ ทักษะการสังเกต การเอาตัวรอด

1.2.4 การประยุกต์ หลักการจากวิชาต่างๆเช่นคณิตศาสตร์มาช่วยสร้างเกม

1.3 ขอบเขตของโครงการ

เกม SHOOT IT! เป็นเกมที่ไม่มีสิ้นสุด จนกว่า ตัวผู้เล่นเองนั้น จะ โดน มอนสเตอร์บุกถึงที่ฐานที่พของตนเอง

1.3.1 ประเภทของโครงการ

โครงการเป็นประเภทเกมที่สร้างจากภาษา java โดยมีเนื้อหาและ concept ของ OOP

โปรเจกต์เกมยิงปืนเป็นโครงการที่น่าสนใจและท้าทายในการพัฒนาเกม

1.3.2 ระบบเกมพื้นฐาน

- การสร้างตัวละครเปรียบเสมือนผู้เล่น(ป้อมปืน)/ศัตรู
- การคุมตัวละครด้วยเมาส์

- ระบบพลังชีวิตจากการตรวจสอบ ตำแหน่ง ศัตรู ว่าเข้าใกล้ป้อมปืนหรือยัง

1.3.3 กราฟิก

- การสร้างกราฟิกสวยงามสำหรับโลกเกมและตัวละคร.

1.3.4 ควบคุมและอินเตอร์เฟซ

- การออกแบบอินเตอร์เฟซผู้เล่น, เช่น เมนูหลัก, การตั้งค่า, และอื่น ๆ

1.3.3 การจัดการโปรเจกต์

- การวางแผนและจัดการโปรเจกต์เพื่อให้มันสามารถเสร็จสิ้นได้ตามเวลา

ตารางแผนการทำงาน

ลำดับ	รายการ	16-23 กันยายน	24 กันยายน -1 ตุลาคม	2ตุลาคม -31 ตุลาคม
1	ออกแบบตัวละคร อินเตอเฟซ			
2	ศึกษา กลไกเกม ข้อมูลต่างๆ			
3	Implementation			
4	จัดทำ เอกสาร			
5	ตรวจสอบ ข้อผิดพลาด			

บทที่ 2

ส่วนของการพัฒนา

2.1 Storyboardและวิธีการเล่น

รายละเอียดเกม

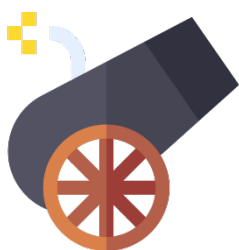
ผู้เล่นต้องเป็นปกป้อง ปรากฏตนเอง จากศัตรูกลายพันธุ์ จากอีกมิติ โดย ป้อมปืนของผู้เล่นนั้นจะถูกทำลายลงก็ต่อเมื่อมีศัตรู เข้ามาประชิด ตัวผู้เล่น หรือปืนของผู้เล่นได้ ผู้เล่นจะได้รับประสบการณ์ที่ดีและสวยงามจากการกำจัดศัตรูพร้อมกับการพบเจอเรื่องราวต่างๆในการผจญภัย

วิธีการเล่น

ใช้ เมาส์ ขยับเล็งไปที่ศัตรูจากนั้นทำการกดปุ่ม M1/M2(คลิกซ้าย หรือขวาก็ได้)

ที่ศัตรูเพื่อกำจัดศัตรู ก่อนที่ จะเข้ามาประชิด ปรากฏของเรา

Storyborad



สุดยอดปืนพิฆาตศัตรู (ผู้เล่น)



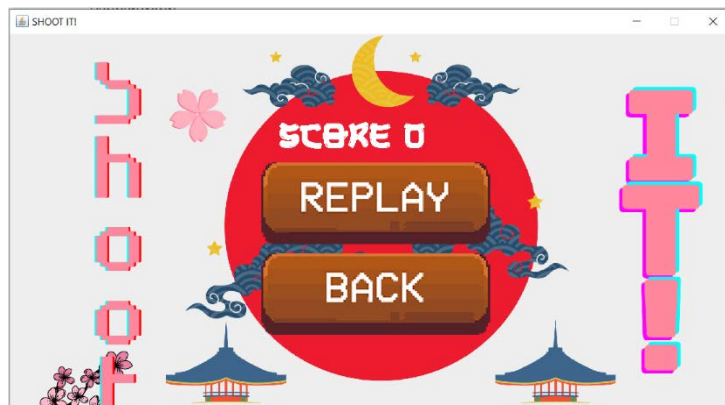
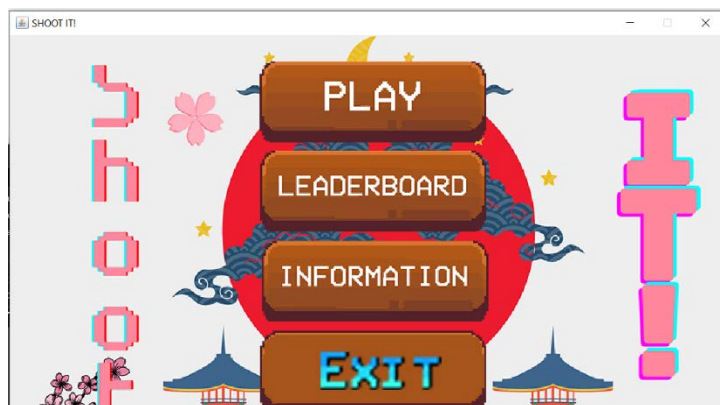
ผึ้งกลายพันธุ์ (HP 1,Score 100)



ค้างคาวกลายพันธุ์ (HP 3,Score 200)

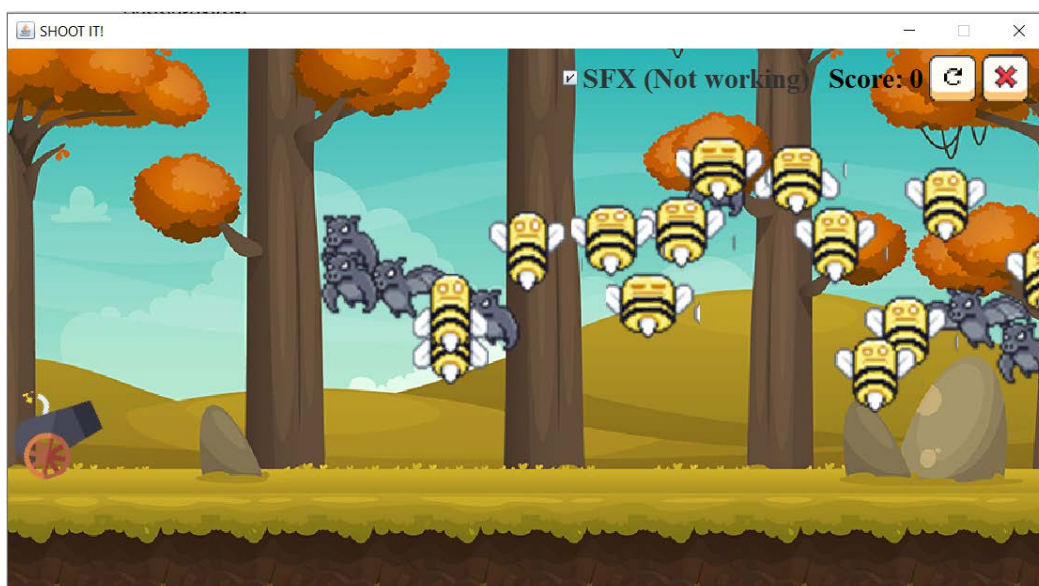
ฉากภายในเกม

1. หน้าเริ่มต้น และ หน้าจบเกม



2. ฉากภายในเมื่อเริ่มเกม

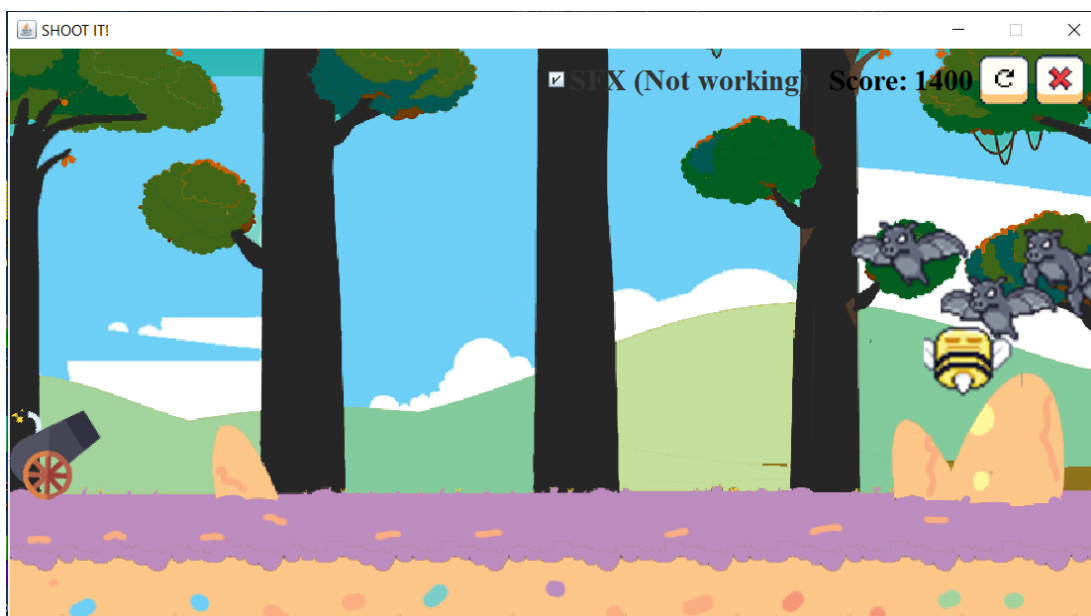
2.1 เมื่อคะแนน เท่ากับ 0 หรือน้อยกว่า 500



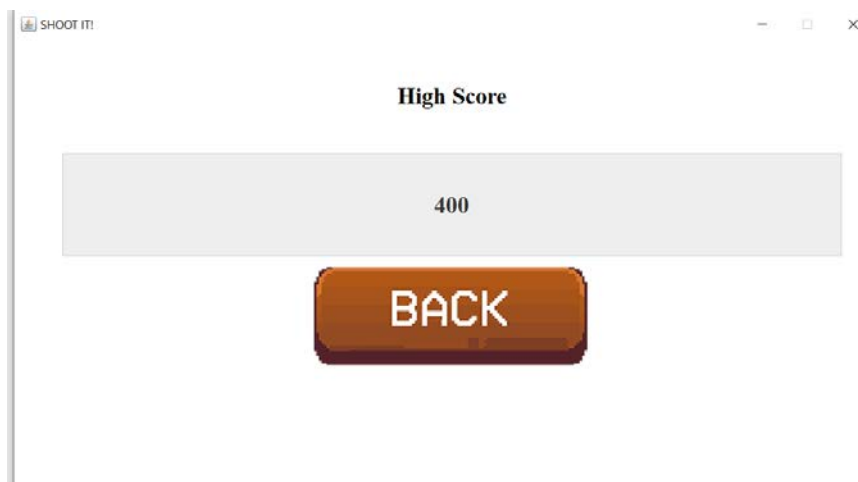
2.2 เมื่อคะแนน เท่ากับ 500 หรือน้อยกว่า 1200



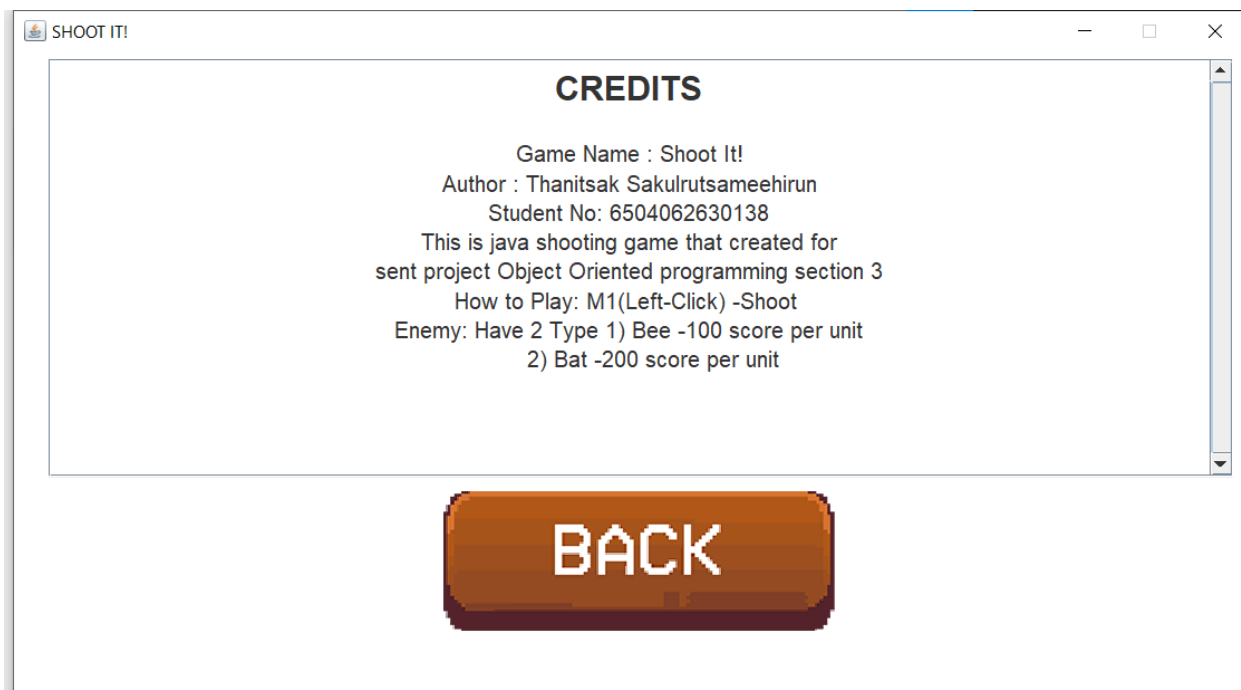
2.3 เมื่อคะแนน เท่ากับ 1200



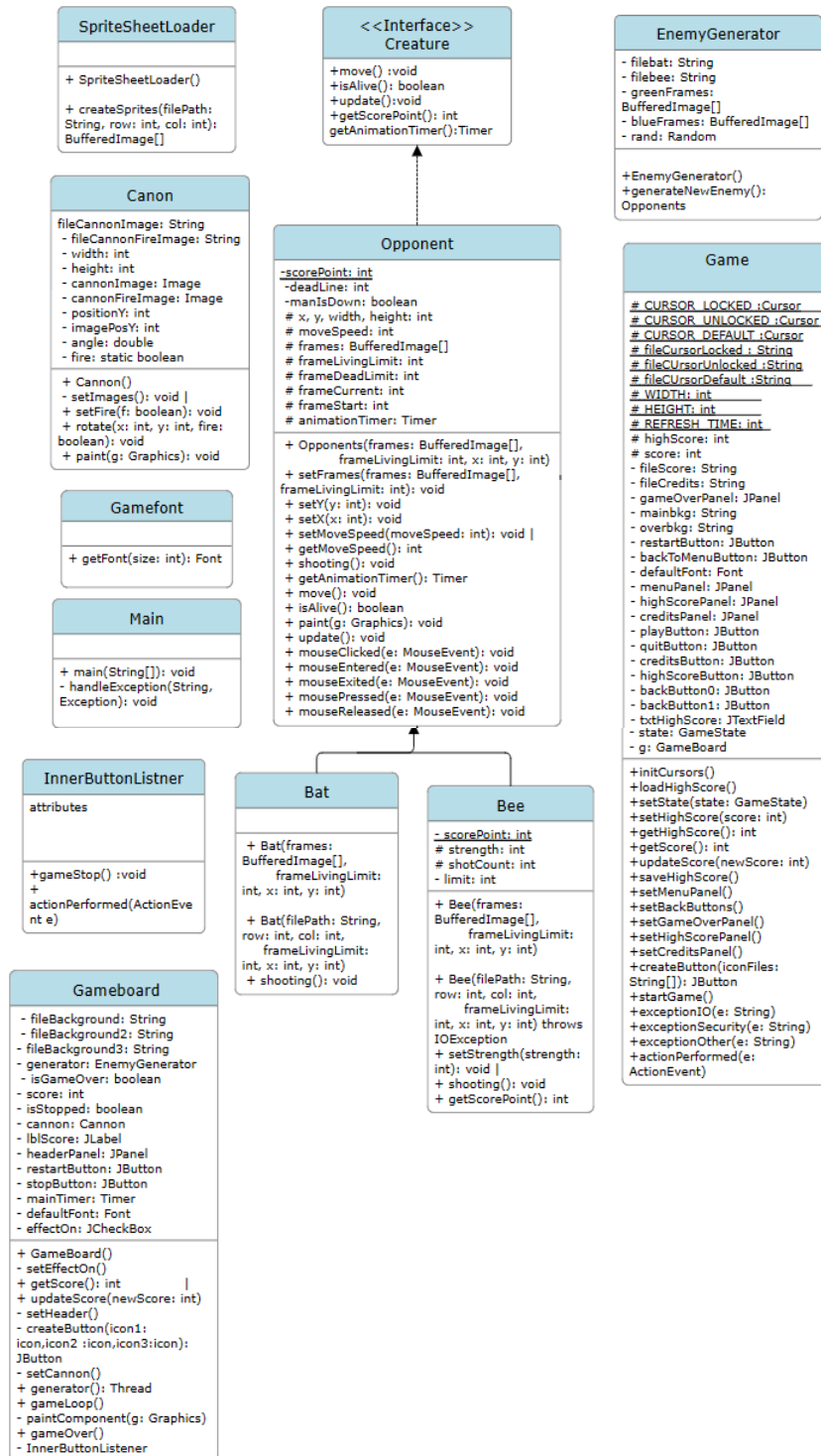
3. หน้า Leaderboard



4. หน้า Information



2.1 Class diagram



2.2 คลาสที่มีการใช้คุณสมบัติของ oop

2.2.1 คลาสที่มีการใช้Encapsulation

```
public class GameBoard extends JPanel {
    private final String fileBackground = "images/backgrounds/background.png";
    private final String fileBackground2 = "images/backgrounds/background2.png";
    private final String fileBackground3 = "images/backgrounds/lastbkg.png";
    private EnemyGenerator generator;
    protected static boolean isGameOver = false;
    public int score;

    private boolean isStopped;

    private Cannon cannon;
    private JLabel lblScore;
    private JPanel headerPanel;
    private JButton restartButton, stopButton;
    private Timer mainTimer;
    private Font defaultFont;
    private JCheckBox effectOn;
}
```

นี่คือตัวอย่างการใช้ Encapsulation จาก Gameboard

2.2.2 คลาสที่มีการใช้ Inheritance

```
✓ public class Bee extends Opponents {
  ⚡ private final int scorePoint = 200;

  protected int strength, shotCount;
  private int limit; // will be used as second living limit.

  // Constructor calls Class Bee's constructor with given arguments and
  // half of the given frameLivingLimit argument because Bat has
  // a strength value which means that after one shot its shape will be cha
  // but it will not die untill the shotCount value reaches to strength val
  ✓ public Bee(BufferedImage[] frames, int frameLivingLimit, int x, int y) {
    super(frames, frameLivingLimit / 2, x, y);
  }
}
```

ตัวอย่างการสืบทอด ฟังก์ชันการสืบทอดความสามารถจากศัตรู

2.2.3 คลาสที่มีการใช้ Polymorphism

```
@Override
public void shooting() {
    // After the first shot the animation will start from 6th frame so
    // the shape will be changed.
    // When shotCount value equals to strength value the manIsDown will be become to true,
    // so it will be dead.
    if(++shotCount < strength) {
        frameStart = 6;
        frameLivingLimit = limit;
    }
}
```

2.2.4 คลาสที่มีการใช้ interface-implement

```
public abstract class Opponents extends JPanel implements Creature, MouseListener {
    private final int scorePoint = 100;

    protected int deadline; // game will over when Opponents get that position.
    protected boolean manIsDown; // if true animation will changed and will disappear.
    protected int x, y, width, height;
    protected int moveSpeed;
}
```

```
public interface Creature {
    public void move(); // Calculates new position of the Creature.
    public boolean isAlive();
    public void update(); // Decides which frame should be current.
    public int getScorePoint();
    Timer getAnimationTimer();
}
```

ตัวอย่างการinterface ว่าทุกๆศัตรูมีการขยับ ที่มี scoreแต่ละตัว

2.2.5 คลาสที่มีการใช้ abstract

```
public abstract void shooting();
```

```
public abstract class Opponents extends JPanel implements Creature, MouseListener {
    private final int scorePoint = 100;
```

ตัวอย่างการการนำ abstract มาใช้ ในโปรแกรม ใช้ในส่วน bee,bat

2.2.6 คลาสที่มีการใช้ composition

```
public static GameState state;
GameBoard g = new GameBoard();

public Game() throws HeadlessException, IOException {
    this(title:"SHOOT IT!");
}
```

ในคลาส Game มีการนำ GameBoard มาใช้

2.2.7 คลาสที่มีการใช้ constructor

```
public Game() throws HeadlessException, IOException {
    this(title:"SHOOT IT!");
}

public Game(String title) throws HeadlessException, IOException {
    super(title);
    initCursors();
    loadHighScore();
    setBackButtons();

    defaultFont = new Font(Font.SERIF, Font.BOLD, size:24);

    setMenuPanel();
    setHighScorePanel();
    setCreditsPanel();
    setGameOverPanel();

    // Set layout as null since every component's position are
    setLayout(manager:null);
    setPreferredSize(new Dimension(WIDTH, HEIGHT));
    setBackground(color.WHITE);
    setDefaultCloseOperation(EXIT_ON_CLOSE);
    setResizable(resizable:false);
    setVisible(b:true);
    pack();
}
```

```
y {
    Game game = new Game(title:"SHOOT IT!");
```

ตัวอย่างการ ทำ constructor และการเอาไปใช้ ระหว่าง game กับ main

2.3 หน้าจอ GUI

โปรแกรมนี้มี GUI ที่ประกอบด้วยหลาย Component สำหรับเกมและหน้าต่างต่าง ๆ ซึ่งเราสามารถแบ่งออกมาดังนี้:

JFrame (Game class):

เป็นหน้าต่างหลักของแอปพลิเคชัน GUI เกม.

ประกอบไปด้วยหลาย JPanel สำหรับแสดงหน้าต่างต่าง ๆ ของเกม.

JPanel (menuPanel, gameOverPanel, highScorePanel, creditsPanel):

เป็นพื้นที่แสดงข้อมูลหรือปุ่มต่าง ๆ ในหน้าต่างต่าง ๆ ของเกม.

มีการกำหนดพื้นหลังจากรูปภาพ (background image) ที่ถูกโหลดจากไฟล์รูปภาพต่าง ๆ.

ประกอบไปด้วย JButton สำหรับเป็นตัวควบคุมต่าง ๆ ที่ผู้เล่นสามารถคลิก.

JButton (playButton, highScoreButton, creditsButton, quitButton, restartButton, backToMenuButton, backButton0, backButton1):

ปุ่มต่าง ๆ ที่ให้ผู้เล่นคลิกเพื่อทำการต่าง ๆ เช่น เริ่มเกม, ดูคะแนนสูงสุด, ดูเครดิต, ออกจากเกม, รีเซ็ต, กลับไปที่เมนู, กลับไปที่หน้าก่อนหน้า.

JTextField (txtHighScore):

ช่องข้อความที่แสดงคะแนนสูงสุด.

JTextPane (txtPane):

กล่องข้อความที่ใช้แสดงเครดิตของเกม.

JScrollPane (scrollPane): ใช้สำหรับเลื่อนเนื้อหาของ txtPane ในกรณีที่ข้อความมีมากเกินไป.

2.4 Event Handling ในหน้าจอ

MouseMotionListener และ MouseAdapter:

ใน GameBoard มีการใช้ MouseMotionListener เพื่อติดตามตำแหน่งของเมาส์ ทำให้ตัวปืนหันไปทางตำแหน่งของเมาส์:

playButton:

เมื่อผู้เล่นคลิกที่ปุ่ม "Play", โปรแกรมจะเรียกเมทอด actionPerformed ใน Game class.

ในเมทอดนี้, ระบบจะลบ gameOverPanel, และเปลี่ยนสถานะของเกมเป็น NEW เพื่อเริ่มต้นเกมใหม่.

highScoreButton:

เมื่อผู้เล่นคลิกที่ปุ่ม "High Score", โปรแกรมจะเรียกเมทอด actionPerformed ใน Game class.

ในเมทอดนี้, ระบบจะแสดงหน้าจอ High Score โดยลบ menuPanel และเปลี่ยนสถานะของเกมเป็น HIGHSCORE.

backButton0, backButton1:

เมื่อผู้เล่นคลิกที่ปุ่ม "Back" ในหน้า High Score หรือ Credits, โปรแกรมจะเรียกเมทอด actionPerformed ใน Game class.

ในเมทอดนี้, ระบบจะลบ Panel ปัจจุบันและแสดง menuPanel, จากนั้นเปลี่ยนสถานะของเกมเป็น WAIT.

quitButton:

เมื่อผู้เล่นคลิกที่ปุ่ม "Quit", โปรแกรมจะเรียกเมทอด actionPerformed ใน Game class.

ในเมทอดนี้, ระบบจะลบ gameOverPanel และเปลี่ยนสถานะของเกมเป็น QUIT, ทำให้โปรแกรมจบการทำงาน.

restartButton:

เมื่อผู้เล่นคลิกที่ปุ่ม "Restart" หลังจากเกม Game Over, โปรแกรมจะเรียกเมทอด actionPerformed ใน Game class.

ในเมทอดนี้, ระบบจะลบ gameOverPanel, สร้าง GameBoard ใหม่, และเปลี่ยนสถานะของเกมเป็น NEW เพื่อเริ่มต้นเกมใหม่.

backToMenuButton:

เมื่อผู้เล่นคลิกที่ปุ่ม "Back to Menu" หลังจาก Game Over, โปรแกรมจะเรียกเมทอด actionPerformed ใน Game class.

ในเมทอดนี้, ระบบจะลบ gameOverPanel, แสดง menuPanel, และเปลี่ยนสถานะของเกมเป็น WAIT.

2.5 อัลกอริทึมที่สำคัญ Enemy Generation Algorithm:

ใน generator() method ของ GameBoard มีอัลกอริทึมที่ทำหน้าที่สร้างศัตรู (Enemy) ใหม่ในเกมทุกๆ ระยะเวลาที่กำหนด. มีการใช้ loop และ sleep เพื่อควบคุมการสร้างศัตรูใหม่.

Collision Detection Algorithm:

ใน gameLoop() method ของ GameBoard มีการตรวจสอบการชนกันของวัตถุต่างๆ ในเกม, เช่น ถ้า ศัตรูตาย (ไม่มีชีวิต), จะเพิ่มคะแนนและลบวัตถุนั้นออกจาก GUI.

Mouse Tracking Algorithm:

ใน MouseMotionListener มีอัลกอริทึมที่ติดตามตำแหน่งของเมาส์และทำให้ปืนหันไปทางตำแหน่งของเมาส์. ในที่นี้ใช้ความสัมพันธ์ระหว่างตำแหน่งเมาส์และตำแหน่งปืนเพื่อหามุมการหันที่ถูกต้อง.

บทที่ 3

สรุป

3.1 ปัญหาที่พบระหว่างการพัฒนา

1. การเรนเดอร์ปุ่มและด้านที่ไม่ดี: ส่งผลให้บางครั้งปุ่มซ้อนกับหน้าเล่นเกม
2. ปัญหาการทำงานของภาพและการเคลื่อนไหว: พบปัญหาการ วาดภาพใหม่ของปืนโดยติดปัญหาที่พื้นหลังของปืนที่ไม่คงที่ปัจจุบันยังติดอยู่
3. การเกิดของศัตรูที่ล่องหน: ทำให้บางทีเกมจบโดยที่เราไม่รู้ด้วยซ้ำเพราะอะไร
4. ไฟล์เสียงไม่ทำงานในเกม

3.2 จุดเด่นของโปรแกรมที่ไม่เหมือนใคร

โปรแกรมนี้นี้มีจุดเด่นหลายอย่างการใช้หลักการแมทริกซ์ในการช่วยให้เกิดภาพเคลื่อนไหวของศัตรูเรนเดอร์ปืน โดยใช้ ตรีโกณมิติ มาเข้าช่วย กราฟิกที่ดูค่อนข้างลื่นไหล พื้นหลังที่ไม่ทำให้รู้สึกว่บรรยากาศต่างจากเดิมเสียเท่าไร

3.3 คำแนะนำ

อาจารย์ มีการ สอน และการบรรยายที่ดีครับ แต่บางคลาสมีเนื้อหาที่ค่อนข้างหนักหน่วงทำให้อาจจะตามเนื้อหาไม่ทันบ้างครับ