



โครงการ
Mini Project - Game

จัดทำโดย

6504062610099 ณัฐวิษณุ กอมณี

เสนอ

ผู้ช่วยศาสตราจารย์ สกิต ประสมพันธ์

วิชา Object Oriented Programming

ภาคเรียนที่ 1/2567

ภาควิชาวิทยาการคอมพิวเตอร์และสารสนเทศ คณะวิทยาศาสตร์ประยุกต์

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

เกี่ยวกับโครงการ

ชื่อโปรเจค: Fazo The Maze Runner
 นำเสนอโดย: นาย ณัฐวิษญ์ กอมณี
 อาจารย์ผู้สอน: ผู้ช่วยศาสตราจารย์สถิต ประสมพันธ์

บทที่ 1

ที่มาและความสำคัญของโปรเจค

โครงการนี้จัดขึ้นเพื่อวัดผลความสามารถในการเรียนวิชา Object Oriented Programming โดย การนำ เรื่องที่เรียนมาสร้างเป็นชิ้นงานในรูปแบบโปรเจกเกม ผู้จัดทำได้สร้างเกมนี้ขึ้นมาเพื่อความสุข และ เพื่อการศึกษา

ประเภทโครงการ

โครงการทางวิทยาศาสตร์และเทคโนโลยี
 โปรเจกเกม

ประโยชน์

1. ฝึกไหวพริบ ในการเอาตัวรอดจากเจ้ามอนสเตอร์
2. ฝึกความจำ ในการจำทางเขาวงกต
3. มีความสนุก ผ่อนคลาย

ตารางแผนงานการทำงาน

ลำดับ	รายการ	1 - 5	6 - 9	9 - 22	23 - 30
1	วาดตัวละครและฉากพื้นหลัง				
2	ศึกษาวิธีการเขียนโปรแกรม				
3	ลงมือเขียนโปรแกรม				
4	จัดทำเอกสาร				
5	ตรวจสอบและแก้ไขข้อผิดพลาด				

บทที่ 2

ส่วนการพัฒนา

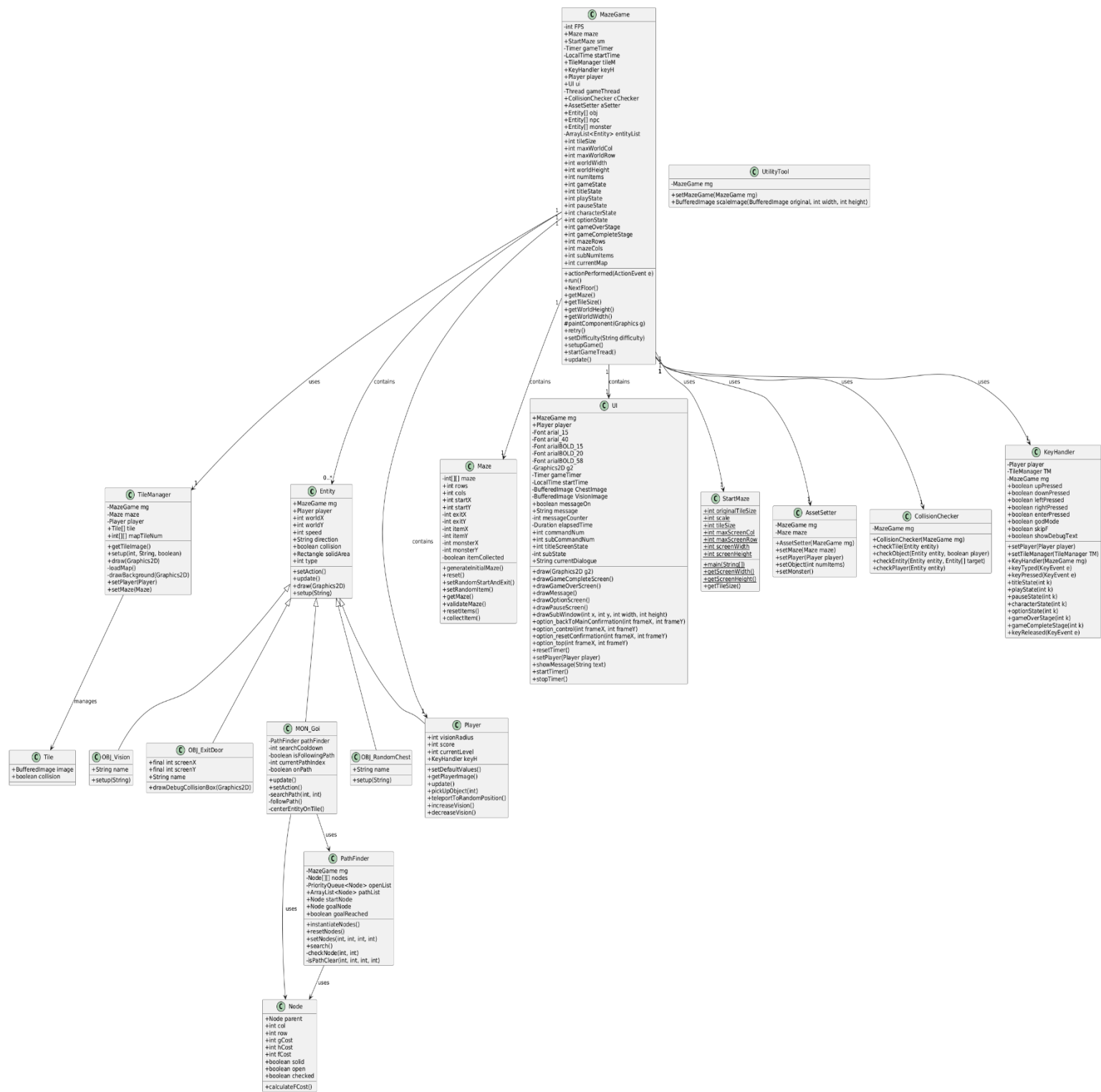
เนื้อเรื่องย่อ

Fazo เป็นนักวิ่งเขาวงกตที่เก่ง เค้ชอบความท้าทาย วันหนึ่งเค้ถูกจับมาไว้ที่เขาวงกตปริศนาที่ไม่มีทางออกเหมือนเขาวงกตทั่วไป และยังมีเจ้ามอนเตอร์ Goi ที่คอยไล่ล่าเค้ทุกครั้งเมื่อเค้ก้าวเข้ามาในเขาวงกตแห่งนี้ เขาวงกตจะเปลี่ยนรูปแบบทุกครั้งที่เข้ามา มีเพียงวิธีเดียวที่จะออกไปได้คือทางออกลึกลับที่จะซ่อนอยู่บนพื้นของทางเดิน Fazo จะต้องวิ่งหาประตูลับพร้อมกับต้องหลบหนีไม่ให้เจ้ามอนเตอร์ Goi จับได้

วิธีการเล่น

ใช้ปุ่มลูกศร W A S D บน ล่าง ซ้าย และ ขวา ตามลำดับในการควบคุมทิศทางการเดิน

Class Diagram



คำอธิบาย Class Diagram

Class Entity เป็นคลาสหลักในการจัดเก็บ และควบคุมพฤติกรรมของ ตัวละคร, Monster และ Object ต่างๆ ภายในเกม

Class Player เป็นคลาสที่จัดเก็บข้อมูลและพฤติกรรมของตัวละครหลักที่ผู้เล่นบังคับ

Class MON_Goi เป็นคลาสที่จัดเก็บข้อมูลและพฤติกรรม ของ Monster ในเกม

Class OBJ_RandomChest , OBJ_ExitDoor , OBJ_Vision จะเป็นคลาสที่จัดเก็บข้อมูลของ Object ภายในเกม

Class Maze เป็นคลาสหลักในการสร้างเขาวงกต และ กำหนดจุดต่างๆ เช่น จุดเกิดผู้เล่น,จุดทางออก จุดเกิดมอนสเตอร์และ จุดเกิด RandomChest

Class TileManager , Tile เป็นคลาสที่จัดการ Tile ต่างๆภายในเกม

Class Pathfinder , Node เป็นคลาสใช้ในการหาเส้นทางของมอนสเตอร์ เพื่อทำให้ผู้เล่น

Class AssetSetter เป็นคลาสที่มีหน้าที่รับผิดชอบในการจัดวางสิ่งต่างๆ ภายในเกม

Class CollisionChecker เป็นคลาสที่เป็นตัวกลางในการตรวจสอบการชนกันของ Entity ต่างๆ ภายในเกม

Class KeyHandler เป็นคลาสที่เป็นตัวกลางในการจัดการระบบป้อนข้อมูลจากผู้ใช้ผ่านแป้นพิมพ์ และนำข้อมูลไปใช้ในการควบคุมตัวละคร หรือ เปลี่ยนสถานะเกมต่างๆ

Class UtilityTool เป็นคลาสที่ตัวช่วยสำหรับการจัดการภาพ Tile ภายในเกม

Class UI เป็นคลาสที่แสดงผลของอินเทอร์เฟซผู้ใช้ในเกม

Class MazeGame เป็นคลาสหลักในการเรียกใช้งานวัตถุต่างๆ

Class StartMaze เป็นคลาสที่ใช้ในการเริ่มเล่นเกม

รูปแบบการพัฒนา

- ภาษา: Java
- GUI: javax.swing
- โปรแกรมวาดรูป: Pixel Studio

อธิบายส่วนของโปรแกรมที่มี

Constructor

```
public MazeGame(int rows, int cols, int numItems, int tileSize) {

    this.numItems = numItems;
    this.tileSize = tileSize;

    this.worldWidth = tileSize * maxWorldCol;
    this.worldHeight = tileSize * maxWorldRow;

    maze = new Maze(rows, cols);

    this.setDoubleBuffered(true);
    this.addKeyListener(keyH);
    this.setFocusable(true);

    gameState = titleState;
    ui = new UI(this);
    aSetter = new AssetSetter(this);
    tileM = new TileManager(this);

}
```

Constructor ของคลาส MazeGame จะมีการตั้งค่าพื้นฐานที่จำเป็นในการเริ่มต้นเกม เช่น ขนาดเขาวงกต จำนวนไอเทม ขนาดของไทม์ สร้างออบเจกต์ที่จำเป็น และกำหนดสถานะเริ่มต้นของเกม

```
public Player(int startX, int startY, MazeGame mg, KeyHandler keyH) {
    super(mg);

    this.worldX = startX * mg.getTileSize();
    this.worldY = startY * mg.getTileSize();

    this.keyH = keyH;

    screenX = StartMaze.screenWidth / 2 - (mg.getTileSize() / 2);
    screenY = StartMaze.screenHeight / 2 - (mg.getTileSize() / 2);

    solidArea = new Rectangle();
    int padding = mg.getTileSize() / 4;
    solidArea.x = padding;
    solidArea.y = padding * 2;
    solidAreaDefaultX = solidArea.x;
    solidAreaDefaultY = solidArea.y;
    solidArea.width = mg.getTileSize() - (padding * 2);
    solidArea.height = mg.getTileSize() - (padding * 2);

    setDefaultValues();
    getPlayerImage();

}
```

Constructor ของคลาส Player จะมีการกำหนดตำแหน่งเริ่มต้นของผู้เล่น กำหนดพื้นที่ชน และตั้งค่า Key Handler รวมถึงกำหนดค่าเริ่มต้นต่างๆ ของผู้เล่น และตั้งค่าตั้งต้นกับโหนดรูปภาพ

```

public TileManager(MazeGame mg) {

    this.mg = mg;
    tile = new Tile[10];
    mapTileNum = new int[mg.maxWorldCol][mg.maxWorldRow];

    getTileImage();

}

```

Constructor ของคลาส TileManager จะมีเชื่อมโยง MazeGame และโดยกำหนดอาร์เรย์สำหรับเก็บประเภทไทม์และตำแหน่งในแผนที่ และโหลดรูปภาพของไทม์ที่จะใช้

```

public UI(MazeGame mg) {
    this.mg = mg;

    arial_15 = new Font("Times New Roman", Font.PLAIN, 15);
    arial_40 = new Font("Times New Roman", Font.PLAIN, 40);
    arialBOLD_15 = new Font("Times New Roman", Font.BOLD, 15);
    arialBOLD_20 = new Font("Times New Roman", Font.BOLD, 20);
    arialBOLD_58 = new Font("Times New Roman", Font.BOLD, 58);
    OBJ_RandomChest rChest = new OBJ_RandomChest(mg);
    OBJ_Vision vision = new OBJ_Vision(mg);
    ChestImage = rChest.image;
    VisionImage = vision.image;

    startTime = LocalTime.now();
    elapsedTime = Duration.ZERO;
    gameTimer = new Timer(1000, e -> {
        LocalTime now = LocalTime.now();
        elapsedTime = Duration.between(startTime, now);
    });
}

```

Constructor ของคลาส UI จะมีเชื่อมโยง MazeGame และมีการกำหนดค่าให้กับฟอนต์ต่าง ๆ มีการสร้าง OBJ_RandomChest และ OBJ_Vision object และนำภาพของมันเก็บไว้ใน ChestImage และ VisionImage และมี การเริ่มต้นเวลาสำหรับ gameTimer

Encapsulation & Composition

```

Maze maze;
StartMaze sm;
Timer gameTimer;
LocalTime startTime;
public TileManager tileM;
public KeyHandler keyH = new KeyHandler(this);
public Player player;
public UI ui;
Thread gameThread;
public CollisionChecker cChecker = new CollisionChecker(this);
public AssetSetter aSetter = new AssetSetter(this);
public Entity obj[] = new Entity[10];
public Entity npc[] = new Entity[10];
public Entity monster[] = new Entity[20];
ArrayList<Entity> entityList = new ArrayList<>();

private int tileSize;
public final int maxWorldCol = 62;
public final int maxWorldRow = 62;
public int worldWidth;
public int worldHeight;
public int numItems;

```

มีการกำหนด access modifier ให้กับตัวแปรต่าง ๆ ใน MazeGame class เช่น tileSize, maze, ui, aSetter, tileM, keyH, player, obj, npc, monster ช่วยให้สามารถควบคุมการเข้าถึงและการแก้ไขค่าของตัวแปรเหล่านี้ได้

Polymorphism & Inheritance

```
public void setAction() {

}
```

(setAction() ใน Entity Class)

```
public void setAction() {
    if (onPath) {
        int goalCol = (mg.player.worldX + mg.player.solidArea.x) / mg.getTileSize();
        int goalRow = (mg.player.worldY + mg.player.solidArea.y) / mg.getTileSize();
        searchPath(goalCol, goalRow);
    } else {

        actionLockCounter++;
        if (actionLockCounter == 120) {
            Random random = new Random();
            int i = random.nextInt(100) + 1;

            if (i <= 25) {
                direction = "up";
            } else if (i <= 50) {
                direction = "down";
            } else if (i <= 75) {
                direction = "left";
            } else {
                direction = "right";
            }

            actionLockCounter = 0;
            speed = BASE_SPEED;
            centerEntityOnTile();
        }
    }
}
```

(setAction() ใน MON_Goi Class)

Entity class เป็น superclass และ Player, Monster เป็น subclasses setAction() ที่ถูกกำหนดใน Entity class แต่อาจถูกนิยามใหม่ในแต่ละ subclass เพื่อให้เหมาะสมกับแต่ละประเภท

Abstract

ยังไม่มีการใช้ Abstract

หน้าจอ GUI

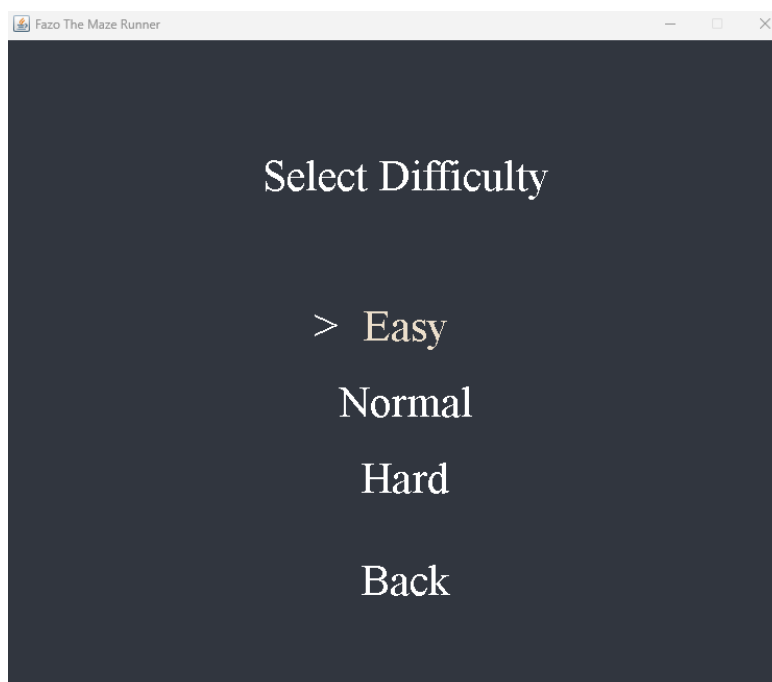
- หน้า Main Menu

ประกอบไปด้วย : ชื่อเกม, เมนูหลัก (New Game, Load Game, Quit)



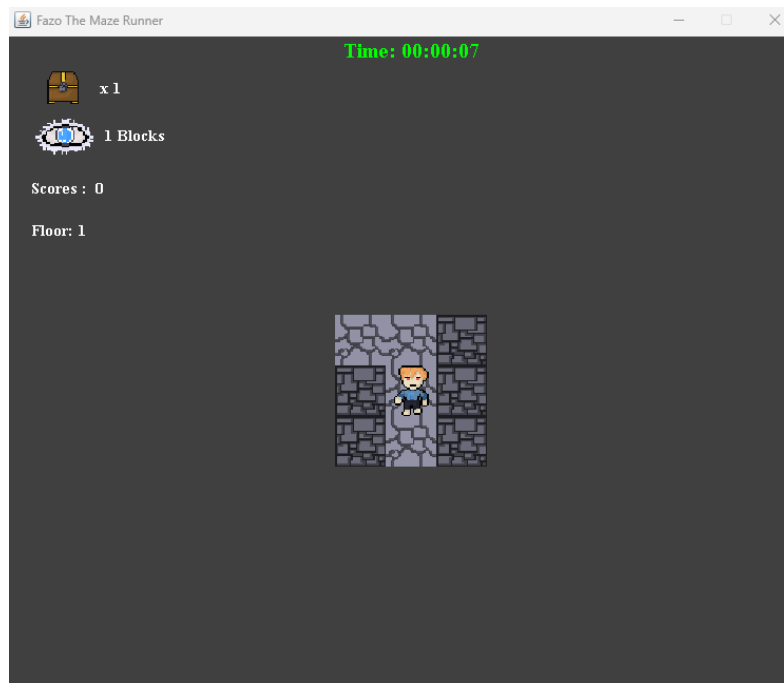
- เมื่อกดเลือก NEW GAME จะขึ้นหน้าจอเลือกระดับ

ประกอบไปด้วย : เมนูเลือกระดับความยาก (Easy, Normal, Hard) และปุ่ม Back



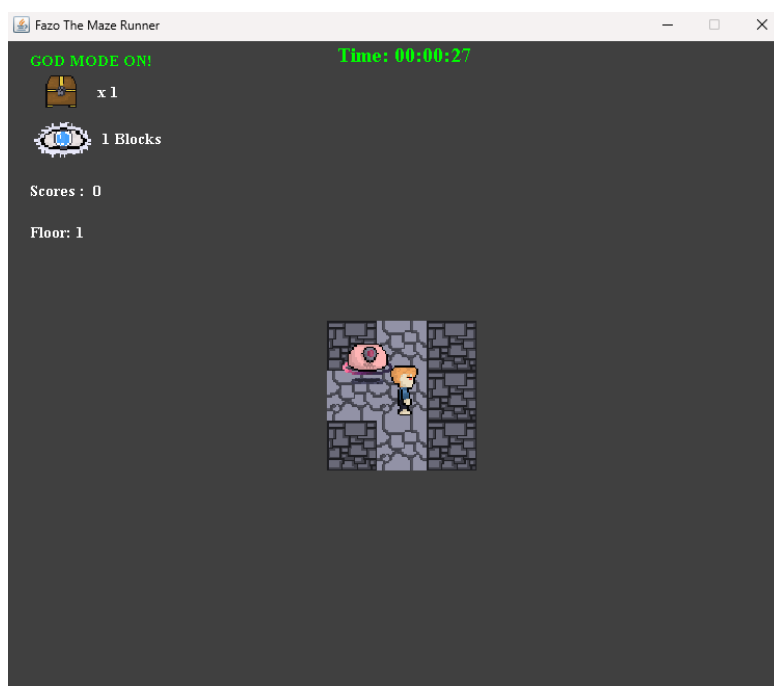
- เมื่อเลือกระดับความยากแล้ว จะเข้าสู่หน้าเกมหลัก

ประกอบไปด้วย : แสดงข้อมูลต่างๆ ในระหว่างเล่นเกม เช่น จำนวนของ, vision radius, scores, floor, และเวลา การแสดงภาพ RandomChest และ Vision ใช้ BufferedImage และภาพ Tile รอบตัวในระยะการมองเห็น



- เมื่อมีมอนสเตอร์เข้ามาในระยะการมองเห็น

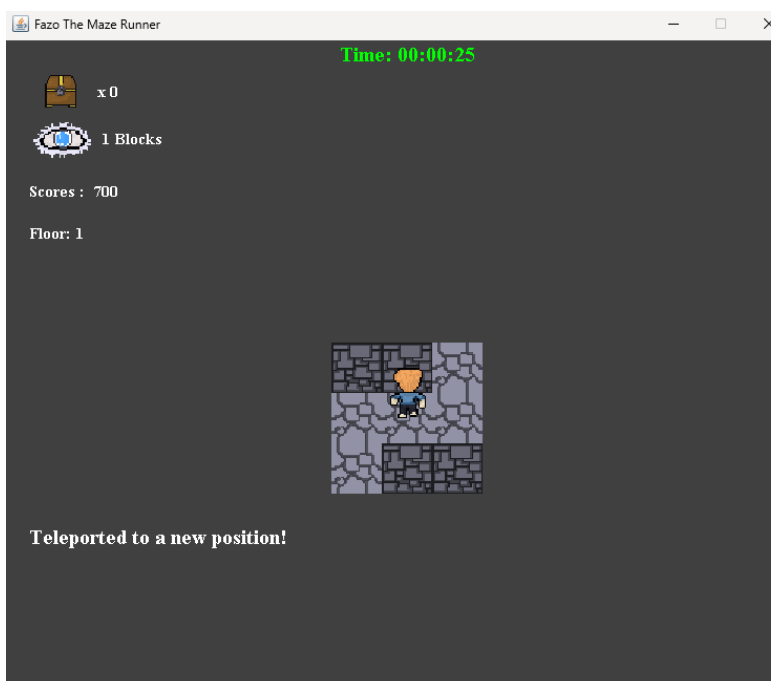
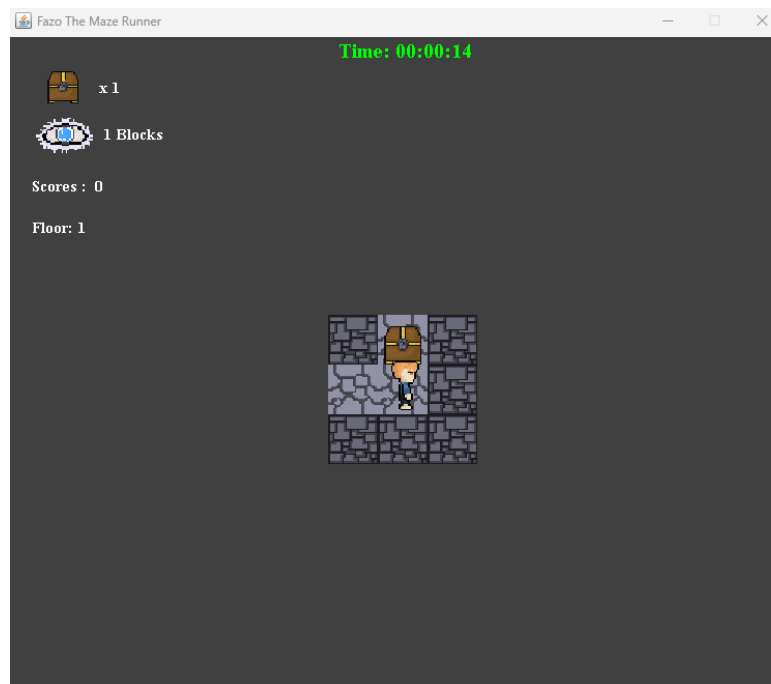
ประกอบไปด้วย : การแสดง MON_Goi ใช้ BufferedImage ในระยะการมองเห็น



อธิบาย Event handling

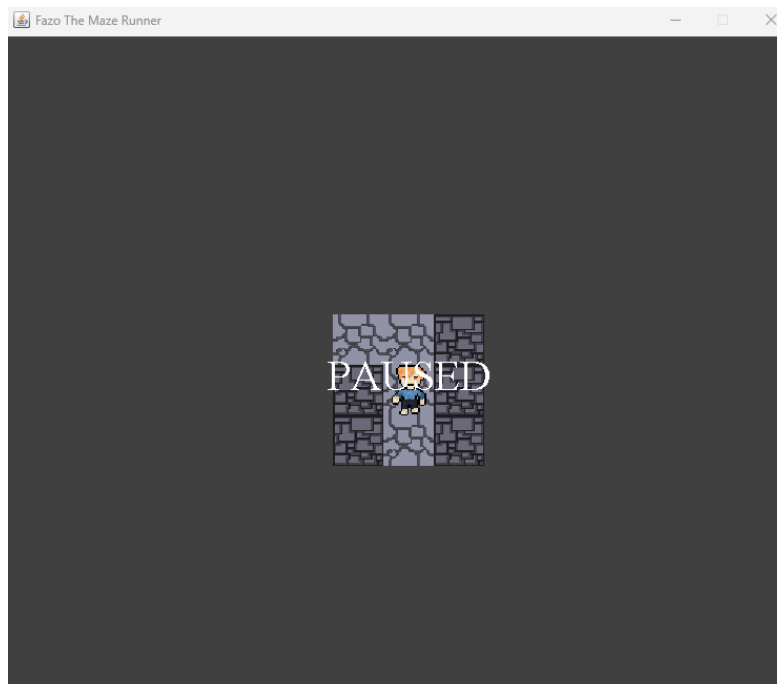
- เมื่อเก็บกล่อง

เหตุการณ์ที่เกิดขึ้น : จะสุ่มสามอย่าง เพิ่มระยะการมองเห็น, ลดระยะการมองเห็น และ สุ่มวาร์ปไปยังพื้นที่อื่นในเขาวงกต



- เมื่อกดปุ่ม P

เหตุการณ์ที่เกิดขึ้น : ข้อความ "PAUSED"



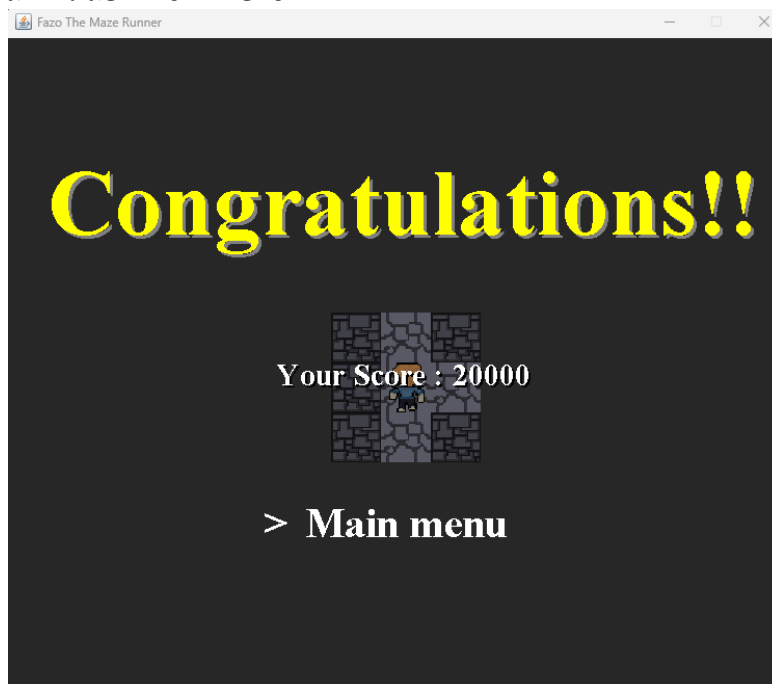
- เมื่อกดปุ่ม ESC

เหตุการณ์ที่เกิดขึ้น : หน้าจอตั้งค่าของเกม เช่น Control, Reset, Main Menu



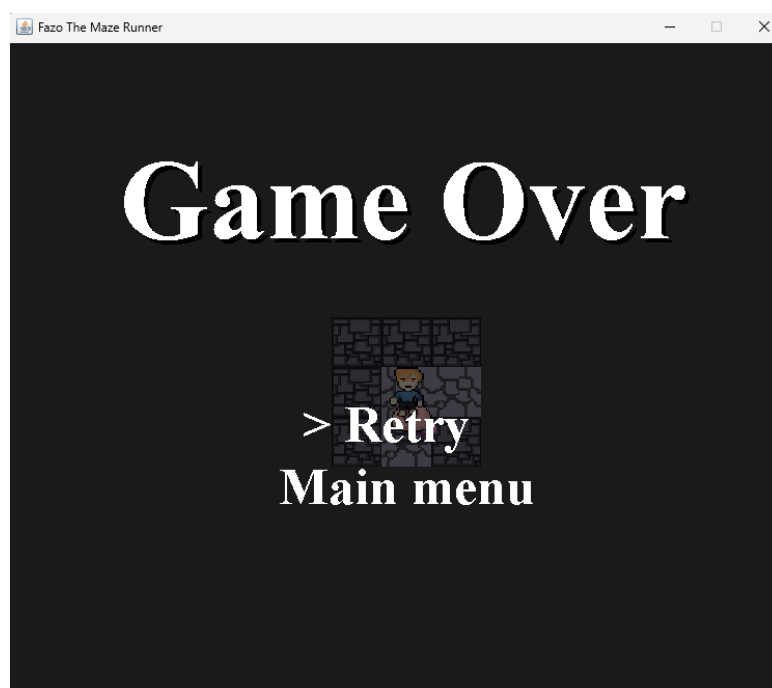
- เมื่อชนะเกม

เหตุการณ์ที่เกิดขึ้น : ข้อความ "Congratulations!!" ขนาดใหญ่ตรงกลางหน้าจอ, แสดงคะแนนสุดท้ายของผู้เล่น และ ตัวเลือก Main Menu



- เมื่อแพ้เกม

เหตุการณ์ที่เกิดขึ้น : ข้อความ "Game Over" ขนาดใหญ่ตรงกลางหน้าจอ, ตัวเลือก Retry และ ตัวเลือก Main Menu



อธิบายอัลกอริทึมที่สำคัญ

```
private void generateMaze(int row, int col) {
    maze[row][col] = 0;

    Integer[] directions = {0, 1, 2, 3};
    Collections.shuffle(Arrays.asList(directions));

    for (int i = 0; i < 4; i++) {
        int direction = directions[i];
        int newRow = row + dRow[direction] * 2;
        int newCol = col + dCol[direction] * 2;

        if (isValid(newRow, newCol) && maze[newRow][newCol] == 1) {
            maze[row + dRow[direction]][col + dCol[direction]] = 0;
            generateMaze(newRow, newCol);
        }
    }
}

private boolean isValid(int row, int col) {
    return row > 0 && row < rows - 1 && col > 0 && col < cols - 1;
}
```

อัลกอริทึมในการสร้างเขาวงกตใช้การสำรวจแบบลึก (Depth-First Search หรือ DFS) ร่วมกับการสุ่ม เพื่อสร้างเส้นทางในเขาวงกตแบบลัดเลาะ

บทที่ 3

สรุป

ปัญหาที่พบระหว่างการพัฒนา

- ปัญหาบั๊กเล็กๆ น้อยๆ และจุดเล็กน้อย ที่ทำให้โค้ดรันไม่ได้
- การเลือกใช้วิธีการดำเนินการของส่วนต่างๆ ในตัวเกม

จุดเด่นของโปรแกรมที่ไม่เหมือนใคร

- การผจญภัยเอาตัวรอดให้ผ่าน ในเขาวงกตที่ถูกสร้างขึ้นอย่างดีตามระดับความยาก
- การเอาตัวรอดจากมอนสเตอร์ เจ้า Goi ที่ไล่ล่าเราอย่างสนุกสนาน

คำแนะนำสำหรับผู้สอนที่อยากให้อธิบาย หรือที่เรียนแล้วไม่เข้าใจ หรืออยากให้เพิ่มสำหรับน้อง ๆ รุ่นต่อไป

- อยากให้ลดคะแนนสอบแล้วเพิ่มคะแนนเก็บขึ้นสักหน่อยกับคะแนนโปรเจคครับ +-5 ประมาณนั้นครับ
- ขอขอบคุณที่ได้ทำโปรเจกเกมดีๆ แบบนี้ครับ สำหรับใครที่อยากสร้างเกมคงได้ความรู้ไปเยอะ

~ขอบคุณครับผม~