โครงงาน

6 th Floor

จัดทำโดย

6604062636704 นายสิรวิชญ์ ผาสุข

เสนอ

ผู้ช่วยศาสตราจารย์ สถิตย์ ประสมพันธ์

วิชา 040613204 การโปรแกรมเชิงวัตถุ(Object-Oriented Programming)

ภาคเรียนที่ 1 ปีการศึกษา 2566

ภาควิชาวิทยาการคอมพิวเตอร์และสารสนเทศ คณะวิทยาศาสตร์ประยุกต์

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

บทที่ 1 : บทนำ

ที่มาและความสำคัญของโปรเจ็ค

โครงงานนี้เกิดจากความชอบของผู้จัดทำที่ชอบเล่นเกมปริศนาและชอบเล่นแบบ Speedrun จึงได้สร้างผลงานชิ้นนี้ขึ้นเพื่อเป็นการวัดไหวพริบของผู้เล่น และยังได้วัดความสามารถและความรู้จากการเรียนในวิชา Object-Oriented Programming

ประเภทของโครงการ

เกม (Game)

ประโยชน์

1. ฝึกความใจเย็น
2. ฝึกวิธีแก้ปัญหา คิด วิเคราะห์
3. ฝึกสมาธิ
4. ฝึกการเขียนโปรแกรมเชิงวัตถุ

ขอบเขตของโครงการ

• รายละเอียดเกมส์

ผู้เล่นจะรับบทเป็นหญิงสาวที่พึ่งกลับมาจากการทำงานแล้วหลับลงไปทันที แต่กลับตื่นขึ้นมาในโลกแห่งความฝันที่เต็มไปด้วยปริศนาต่างๆ ผู้เล่นจะต้องค้นหาปริศนา แก้ปริศนาต่างๆ และใช้ไอเทมที่พบเพื่อแก้ไขปัญหา ผู้เล่นจะต้องสำรวจฉากต่างๆ เพื่อหาทางออกและกลับสู่โลกความเป็นจริงภายในเวลาที่กำหนด หากไม่สามารถแก้ปริศนาและหาทางออกได้ทันเวลา ผู้เล่นจะติดอยู่ในโลกแห่งความฝันที่วนลูปไปเรื่อยๆ

• Storyboard

ตัวละคร

หญิงสาว

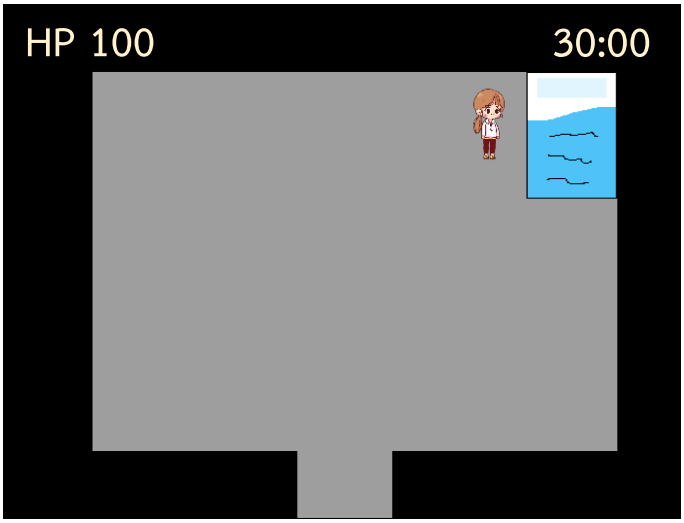
ผีดี

ผีร้าย

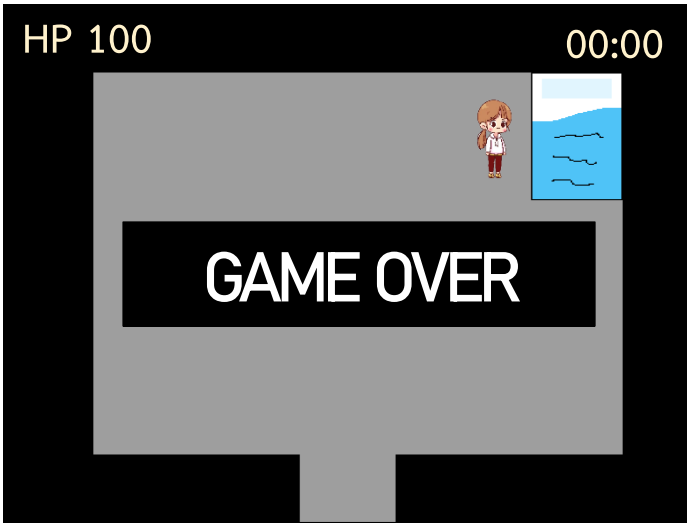
ฉาก

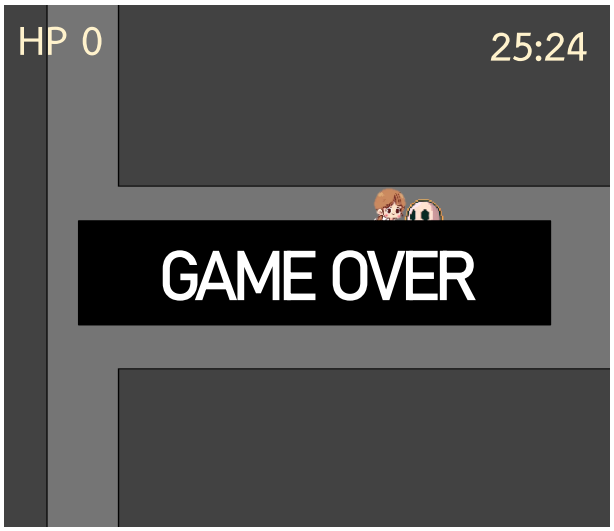
- หน้าแรก

- เริ่มเกม

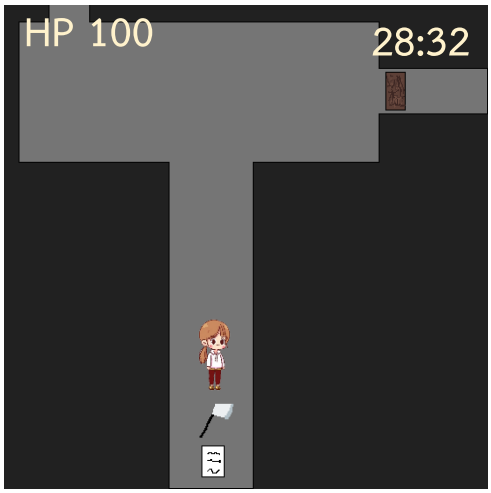


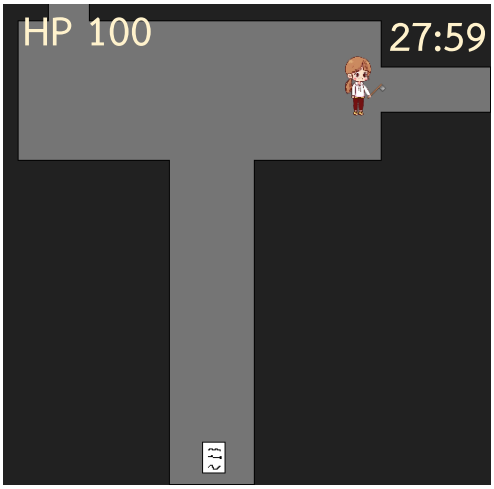
- หมดเวลาหรือ HP ของผู้เล่นหมด

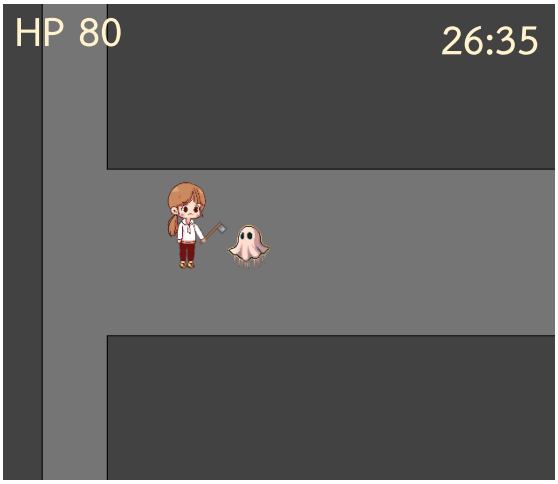




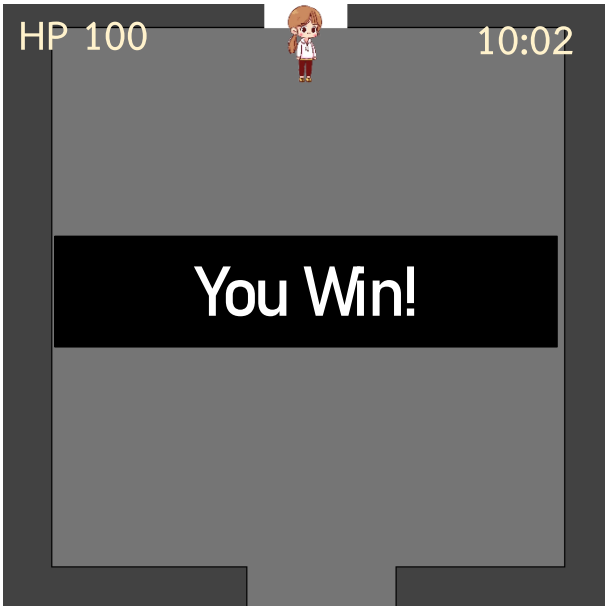
- ใช้อาวุธเพื่อทำลายสิ่งกีดขวางหรือโจมตีผีร้าย







- แก้ไขปริศนาจากคำให้ที่มีอยู่ตามฉาก



• ตารางแผนการทำงาน

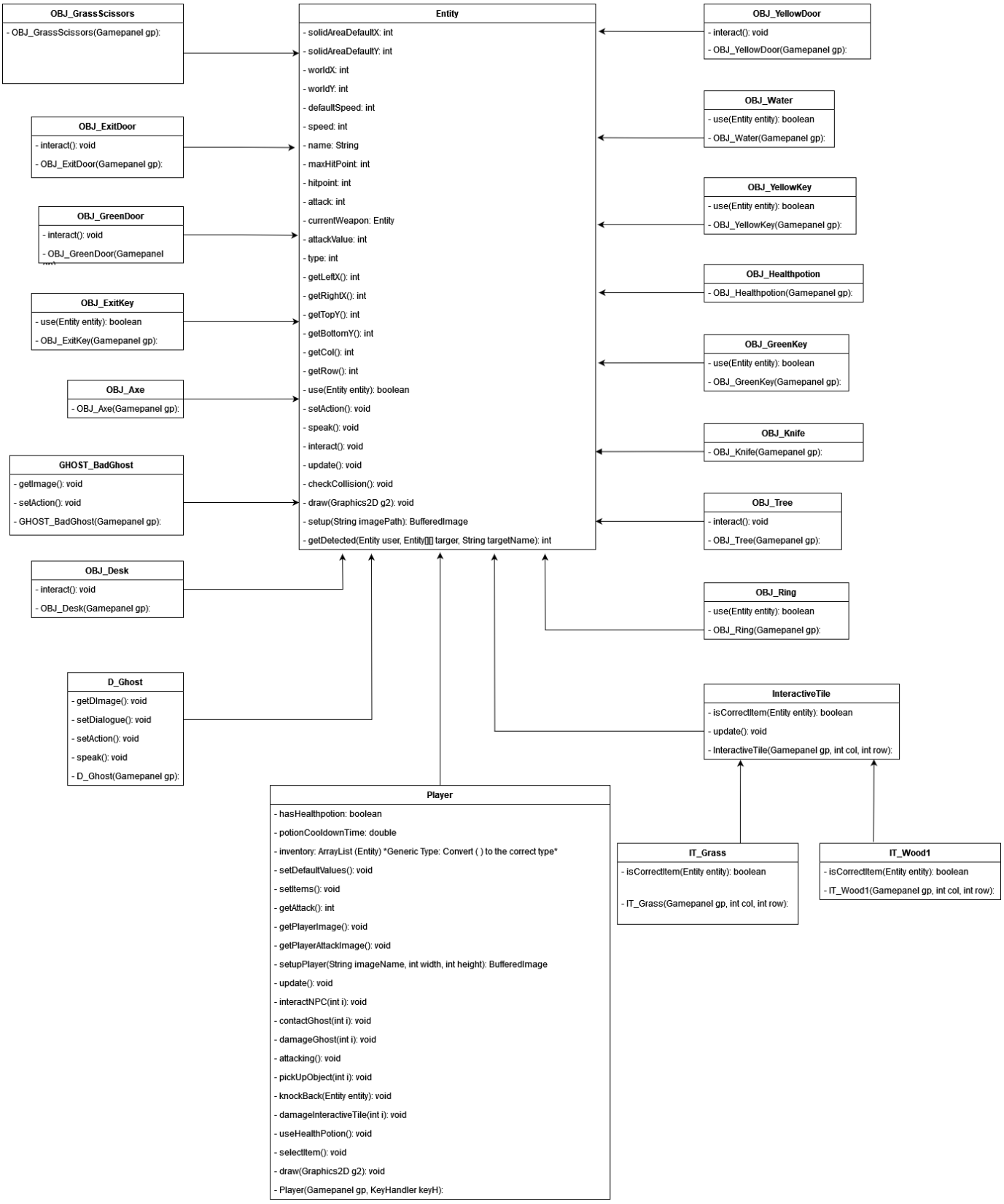
ลำดับ	รายการ	1 – 4	5 – 20	21 – 27	28 – 30
1	ออกแบบและทำกราฟิกตัวละคร และ ฉาก				
2	ศึกษาเอกสารและข้อมูลที่เกี่ยวข้อง				
3	เขียนโปรแกรม				
4	ตรวจสอบข้อผิดพลาดและทำการแก้ไข				
5	จัดทำเอกสาร				

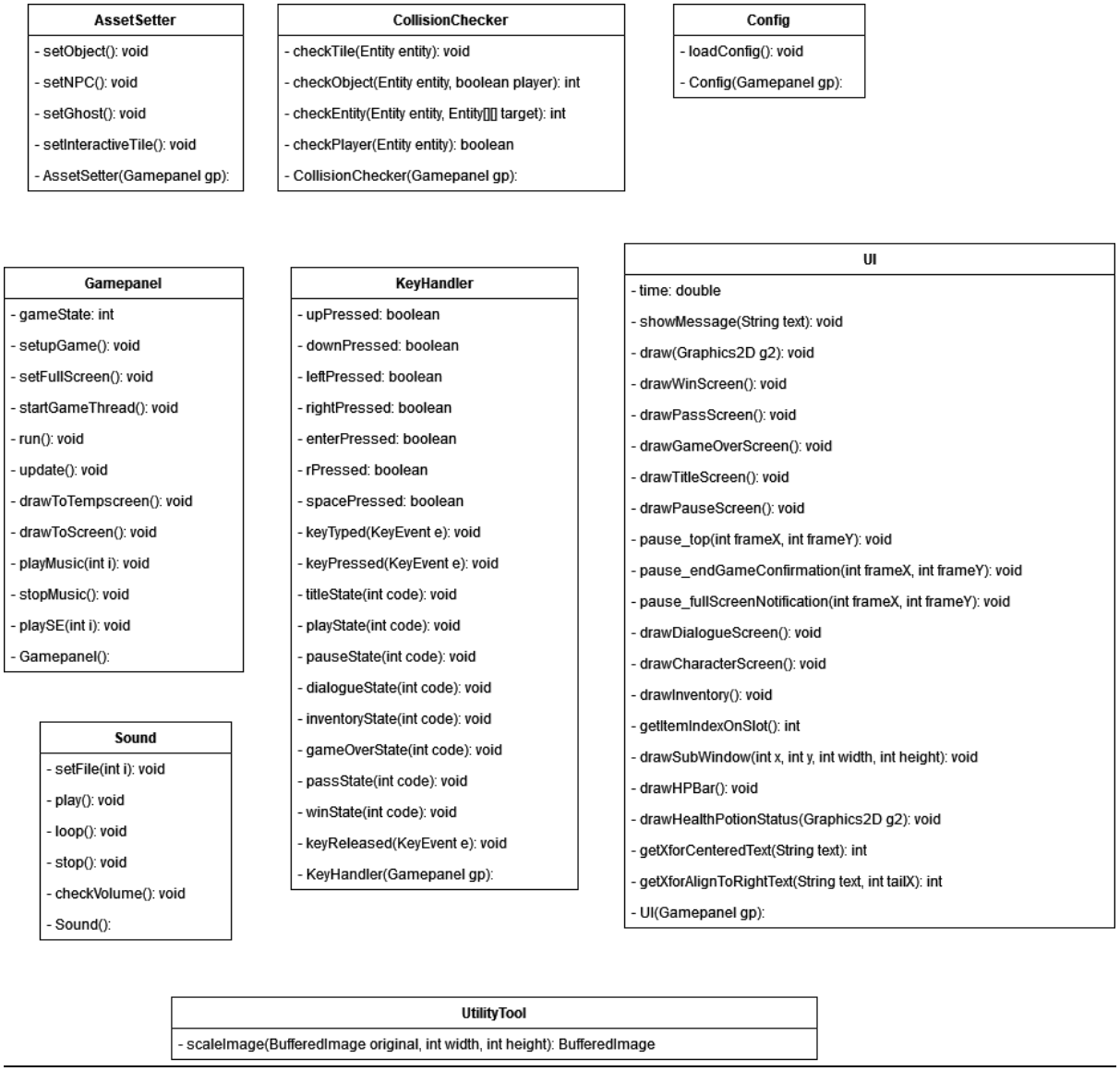
บทที่ 2 : ส่วนของการพัฒนา

วิธีการเล่น

W เดินขึ้นบน A เดินไปทางซ้าย S เดินไปลงข้างล่าง D เดินไปทางขวา กด space bar เพื่อใช้อาวุธที่เก็บมาเพื่อทำลายสิ่งกีดขวางหรือโจมตีผีร้าย และสิ่งกีดขวางแต่ละแบบก็ต้องใช้อาวุธที่แตกต่างกันออกไป กด ENTER เพื่อพูดคุยกับผีดีหรือตรวจสอบสิ่งของ กด I เพื่อเปิดช่องเก็บของ กด R เพื่อใช้ขวดยาซึ่งจะเพิ่ม HP ทีละ 50 หน่วย มีคูลดาวน์อยู่ที่ 45 วินาที ในการใช้กุญแจเพื่อเปิดประตูจะต้องเดินไปที่หน้าประตู เปิดช่องเก็บของและกด ENTER เพื่อใช้งานกุญแจหรือสิ่งของอื่นๆ และสิ่งที่ผู้เล่นต้องทำก็คือ ตามหาขวดน้ำ แล้วนำน้ำไปรดน้ำต้นไม้แล้วจะได้แหวนมา แล้วเอาแหวนที่ได้มาไปให้ผีดีแล้ว ผีดีจะไปให้กุญแจตอบแทนกลับมาให้ แล้วนำกุญแจไปเปิดประตูเพื่อออกจากโลกแห่งความฝันก่อนจะหมดเวลา ซึ่งจะมีเวลาทั้งหมด 15 นาที ในเกมจะมีด่านทั้งหมดอยู่ 2 ด่านซึ่งทั้ง 2 ด่านจะใช้เวลา การนับคะแนนจะนับจากการเอาเวลาที่เหลือมาคูณ 10 แล้วถ้าหมดเวลาหรือ HP หมดก็จะ Game Over ทันที โดยระหว่างทางก็จะมีผีร้ายเข้ามาโจมตีใส่ผู้เล่น โดยผู้เล่นสามารถจะหลีกเลี่ยงหรือโจมตีใส่ผีก็ได้

คลาสไดอะแกรม





คลาสหลักที่จำเป็น

1. Player คือตัวผู้เล่น
2. OBJ คือสิ่งของเครื่องใช้ต่างๆที่จะมาช่วยผู้เล่น
3. D_Ghost คือ NPC ที่จะมาช่วยบอกวิธีผ่านด่าน
4. InteractiveTile คือ สิ่งกีดขวางที่จะต้องใช้อาวุธเฉพาะเพื่อทำลาย
5. Entity คือ คลาสแม่ของ 4 ข้อแรก ทำหน้าที่เป็นตัวเก็บสิ่งที่ 4 ข้อแรกใช้บ่อย
6. KeyHandler ทำหน้าที่คอยรับ Input ต่างๆ
7. Sound รับและสร้างเสียงต่างๆในเกม
8. Config ทำหน้าที่เก็บการตั้งค่าในเกม
9. UI ไว้แสดงหน้าจอต่างๆตามสถานะ
10. Gamepanel เป็นตัวที่คอยวาดภาพต่างที่อยู่ในเกม
11. AssetSetter กำหนดจุดเกิดของ NPC , GHOST และ สิ่งกีดขวาง
12. CollisionChecker คอยเช็คว่า Hitbox ของสิ่งต่างๆชนกันรึเปล่า
13. UtilityTool กำหนดขนาดของภาพ

รูปแบบการพัฒนา

พัฒนาในรูปแบบของ Waterfall Model เพราะมีการทำงานเป็นลำดับง่ายต่อการพัฒนา

Constructor

เมื่อมีการสร้าง object จาก class Sound ก็จะทำการทำการรับเสียงจากไฟล์ที่เก็บไว้

```
public class Sound {
        Clip clip;
         URL[] soundURL = new URL[30];
        FloatControl fc;
        int volumeScale = 3;
        float volume;
        public Sound() {
                this.soundURL[0] = this.getClass().getResource("/res/Sound/GhostAttack.wav");
                this.soundURL[1] = this.getClass().getResource("/res/Sound/Pickup.wav");
                this.soundURL[2] = this.getClass().getResource("/res/Sound/PlayerAttack.wav");
                this.soundURL[3] = this.getClass().getResource("/res/Sound/UnlockDoor.wav");
                this.soundURL[4] = this.getClass().getResource("/res/Sound/press.wav");
                this.soundURL[5] = this.getClass().getResource("/res/Sound/hit.wav");
                this.soundURL[6] = this.getClass().getResource("/res/Sound/drink.wav");
                this.soundURL[7] = this.getClass().getResource("/res/Sound/ready.wav");
                 this.soundURL[8] = this.getClass().getResource("/res/Sound/select.wav");
                 this.soundURL[9] = this.getClass().getResource("/res/Sound/Hit_wood.wav");
                 this.soundURL[10] = this.getClass().getResource("/res/Sound/gameover.wav");
                 this.soundURL[11] = this.getClass().getResource("/res/Sound/grow.wav");
                this.soundURL[12] = this.getClass().getResource("/res/Sound/footstep1.wav");
                this.soundURL[13] = this.getClass().getResource("/res/Sound/footstep2.wav");
          }
```

เมื่อมีการเรียกใช้ TileManager จะทำการไปอ่านไฟล์แผนที่

```
public TileManager(Gamepanel gp) {

    this.gp = gp;

    this.tile = new Tile[20];

    this.mapTileNum = new int[10][50][50];

    this.getTileImage();

    this.loadMap("/res/Map/Worldmap00.txt", 0);

    this.loadMap("/res/Map/Worldmap01.txt", 1);

  }
```

Encapsulation

แต่ละคลาสต้องมีการเข้าถึงข้อมูลของกันและกันจึงใช้ public กันเกือบทั้งหมด

Composition

player , obj, npc ,ghost ,interactiveTile เป็นส่วนหนึ่งของ Gamepanel

```
public class Gamepanel extends JPanel implements Runnable {
    public Player player = new Player(this, this.keyH);

    public Entity[][] obj = new Entity[maxMap][20];;

    public Entity[][] npc = new Entity[maxMap][1];

    public Entity[][] ghost = new Entity[maxMap][100];

    public InteractiveTile[][] iTile = new InteractiveTile[maxMap][100];
```

Polymorphism

-

Abstract

-

Inheritance

public class Player extends Entity

public class D_Ghost extends Entity

public class GHOST_BadGhost extends Entity

public class OBJ_Axe extends Entity

public class OBJ_Desk extends Entity

public class OBJ_ExitDoor extends Entity

public class OBJ_ExitKey extends Entity

public class OBJ_GrassScissors extends Entity

public class OBJ_GreenDoor extends Entity

public class OBJ_GreenKey extends Entity

public class OBJ_Healthpotion extends Entity

public class OBJ_Knife extends Entity

public class OBJ_Ring extends Entity

public class OBJ_Tree extends Entity

public class OBJ_Water extends Entity

public class OBJ_YellowDoor extends Entity

public class OBJ_YellowKey extends Entity

public class InteractiveTile extends Entity

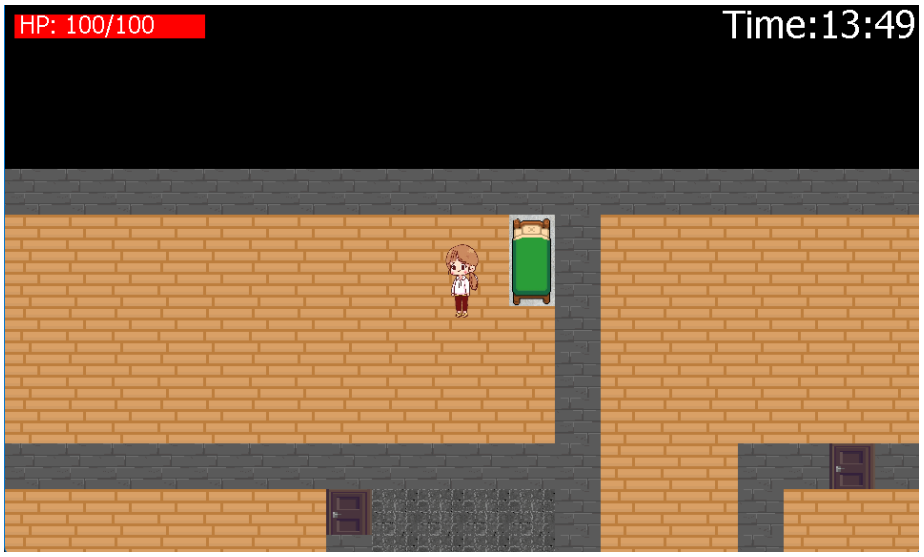
คลาส Player , OBJ ,InteractiveTile และ Ghost ได้รับการ extends มาจากคลาส Entity

หน้าจอ GUI และ Event handling

TitleScreen

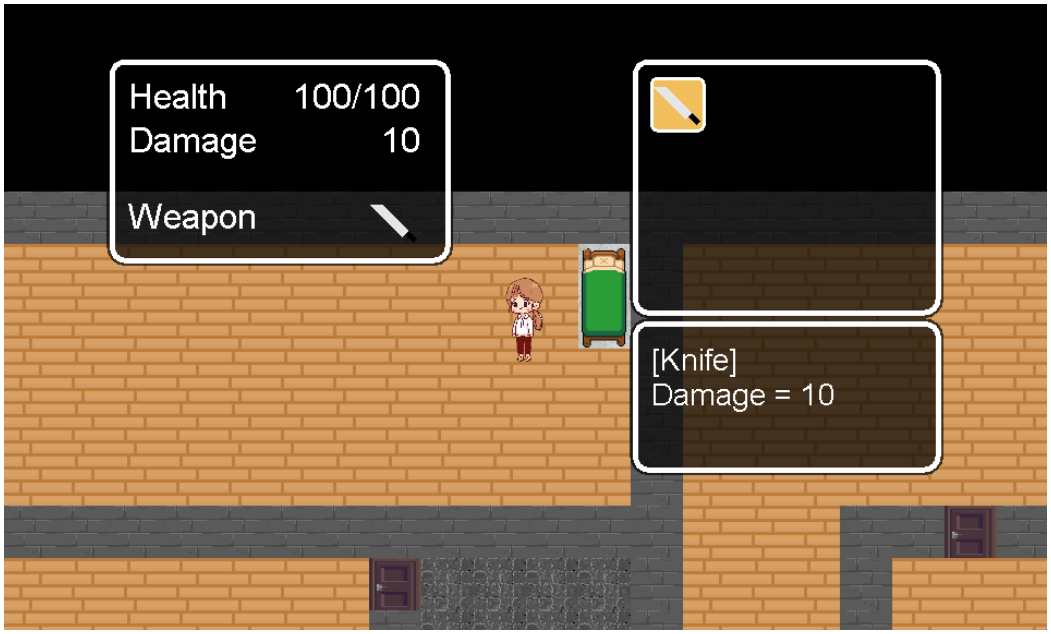
ประกอบด้วย background Image, START, EXIT Event handling จะมีการดักจับปุ่ม W เพื่อเลื่อนขึ้น , ปุ่ม S เพื่อเลื่อนลง และ ENTER เพื่อกดปุ่ม

PlayScreen



ประกอบไปด้วย HP Bar, Timer และ Potion Cooldown Bar เมื่อมีการเก็บยาขึ้นมา Event handling จะมีการดักจับปุ่ม WASD ที่ใช้เดิน ปุ่ม R เพื่อใช้ยา ปุ่ม I เพื่อเปิดกระเป๋า ปุ่ม SPACEBAR เพื่อโจมตี ปุ่ม ESC เพื่อเปิด Menu

InventortScreen



จะมี Status, Item Slots, Item Details, Use Item Button Event handling มีการดักจับปุ่ม WASD เลือกไอเทม ปุ่ม ENTER ใช้ของที่เลือกอยู่ กด I เพื่อปิดกระเป๋า

GameOverScreen

ประกอบไปด้วย Status Message ,Restart, Quit Event handling มีการดักจับปุ่ม W เลื่อนขึ้น S เลื่อนลงENTER เพื่อกดปุ่ม

ExitScreen

ประกอบด้วย Status Message, Next Level, Quit Event handling มีการดักจับปุ่ม W เลื่อนขึ้น S เลื่อนลงENTER เพื่อกดปุ่ม

อัลกอริทึมที่สำคัญในโปรแกรม

ถ้า HP หมดหรือเวลาหมด จะ GameOver

```
if (hitpoint <= 0 || gp.ui.time <= 0.0D) {

        this.gp.playSE(10);

        gp.gameState = gp.gameOverState;

 }
```

ตรงจับเมื่อตีสิ่งกีดขวางว่าใช้อาวุธถูกประเภทไหมถ้าถูกทำดาเมจใส่สิ่งกีดขวาง

```
public void damageInteractiveTile(int i) {

    if (i != 999 && gp.iTile[gp.currentMap][i].destructible && gp.iTile[gp.currentMap][i].isCorrectItem(this) &&
!gp.iTile[gp.currentMap][i].invincible) {

        gp.playSE(9);

        gp.iTile[gp.currentMap][i].hitpoint -= attack;

        gp.iTile[gp.currentMap][i].invincible = true;

        if (gp.iTile[gp.currentMap][i].hitpoint == 0) {

            gp.iTile[gp.currentMap][i] = null;

        }

    }

  }
```

ใช้วาดแผนที่เฉพาะพื้นที่บนหน้าจอของเราเท่านั้น

```
public void draw(Graphics2D g2) {

    int worldCol = 0;

    int worldRow = 0;


    while(true) {

        if (worldCol >= 50) {

            break;

        }


        if (worldRow >= 50) {

            break;

        }
```

```
        int tileNum = mapTileNum[gp.currentMap][worldCol][worldRow];
        int worldX = worldCol * gp.tileSize;
        int worldY = worldRow * gp.tileSize;
        int screenX = worldX - gp.player.worldX + gp.player.screenX;
        int screenY = worldY - gp.player.worldY + gp.player.screenY;
        if (worldX + gp.tileSize * 2 > gp.player.worldX - gp.player.screenX) {
            if (worldX - 48 * 2 < gp.player.worldX + gp.player.screenX) {
                if (worldY + 48 * 2 > gp.player.worldY - gp.player.screenY) {
                    if (worldY - 48 * 2 < gp.player.worldY + gp.player.screenY) {
                        g2.drawImage(tile[tileNum].image, screenX, screenY, (ImageObserver)null);
                    }
                }
            }
        }

        ++worldCol;
        if (worldCol == 50) {
            worldCol = 0;
            ++worldRow;
        }
    }

}
```

บทที่ 3 : สรุป

ปัญหาที่พบระหว่างพัฒนา

- เวลาได้รับดาเมจจากผีบางทีจะไม่มีเสียงขึ้นมา
- ถ้าในตัวเรามียา เมื่อทำการ Restart จะมีสถานะของยาขึ้นอยู่
- เมื่อมี object สักอย่างทะลุออก Map ไปเกมจะ Crash ทันที
- ไม่สามารถ spawn ผีได้จำนวนมากไม่งั้นเกมจะโหลดนานมาก

จุดเด่นที่ไม่เหมือนใคร

- อาวุธแต่ละแบบมีความสามารถเฉพาะ

คำแนะนำสำหรับผู้สอนที่อยากให้อธิบาย หรือที่เรียนแล้วไม่เข้าใจ หรืออยากให้เพิ่มสำหรับน้อง ๆ รุ่นต่อไป

-