

SAVING SIR NGHĨA

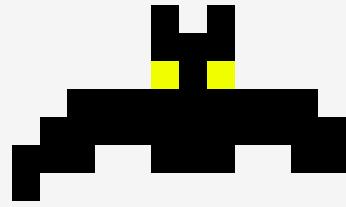
1M5 PLUS GROUP

200
192
184
176
168
160
152
144
136
128
120
112
104
96
88
80
72
64
56
48
40
32
24
16
8





TEAM MEMBERS



DO THANH BAO ANH-ITDSIU22175

NGUYEN MINH KHA-ITITIU22079

PHAM TUAN DANG KHOA-ITITIU22087

TRAN THIEN PHU-ITITIU22124

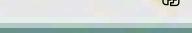


TABLE OF CONTENTS

1 DEMO

2 INTRODUCTION

3 GRAPHICS DESIGN

4 SYSTEM DESIGN

5 CODE FLOW

6 CONCLUSION AND
FUTURE WORK



□□

□E□

02

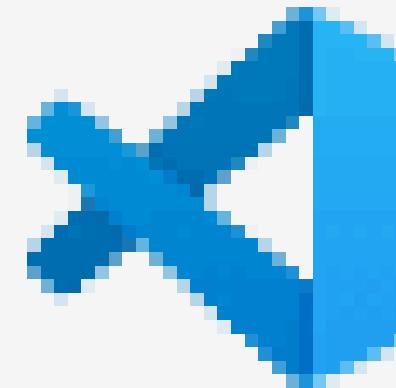
INTRODUCTION

Objectives

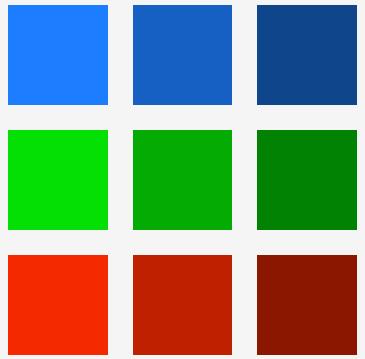
- +Helps people better understand Game Development Concepts such as: game loops, collision detection, rendering, and game physics...
- +Apply knowledge of java programming language and Object-Oriented Programming to create a game that can bring attraction and entertainment to players.
- +Experience the processes of ideating, creating, managing and improving games.

Tools Used

VS CODE



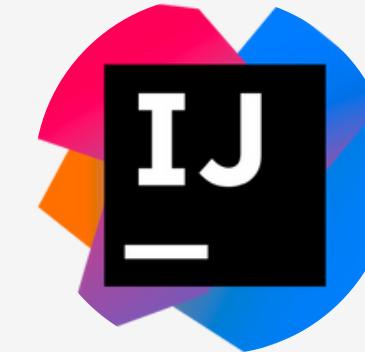
PISKEL



ECLIPSE



GITHUB



INTELLIJ



FACEBOOK





GRAPHICS DESIGN

THE DESIGN

Playing Screen



Player



Skin 1: Angry Grandma



Skin 2: Colorful Boy



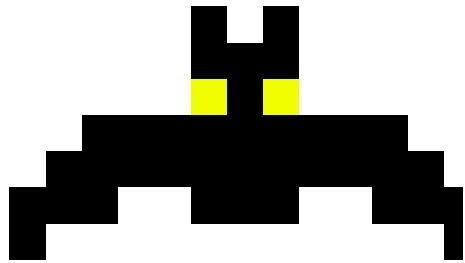
Skin 3: Turtle Ninja

THE DESIGN MONSTER

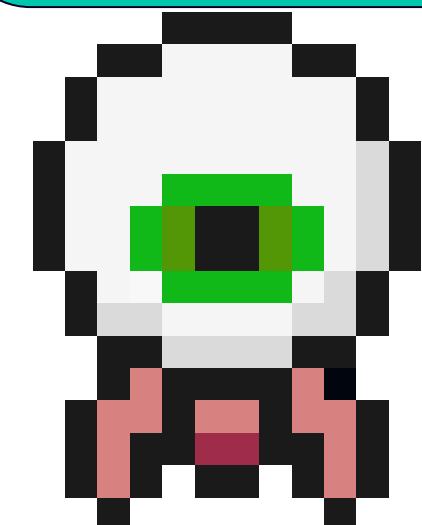
Mini-Groot



Bat



OneEyeCanFly



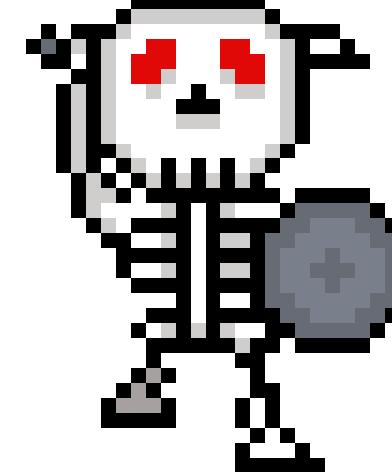
Orc



Skeleton-Lord
Phase 1

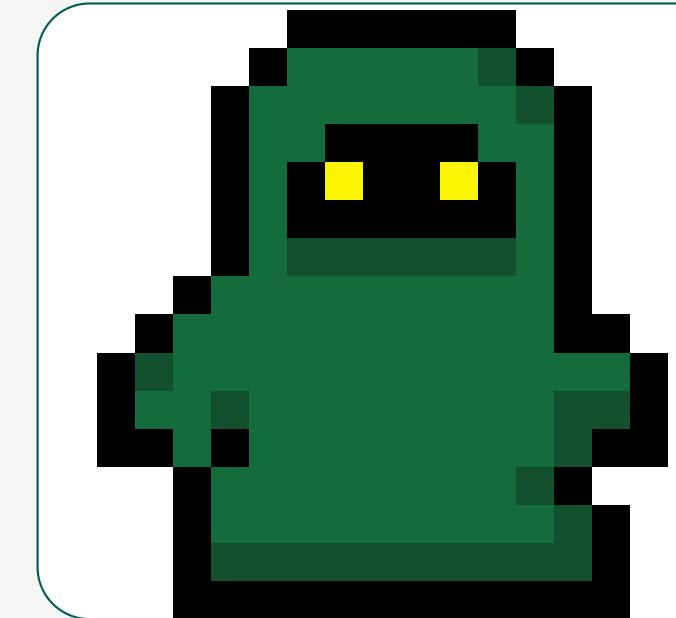


Skeleton-Lord
Phase 2



NPCs

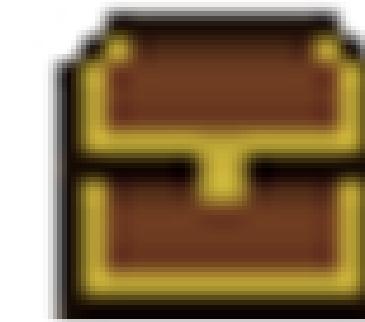
MECHANT



OLDMAN



OBJECT



Chest



Door



Heart



Key



Lantern

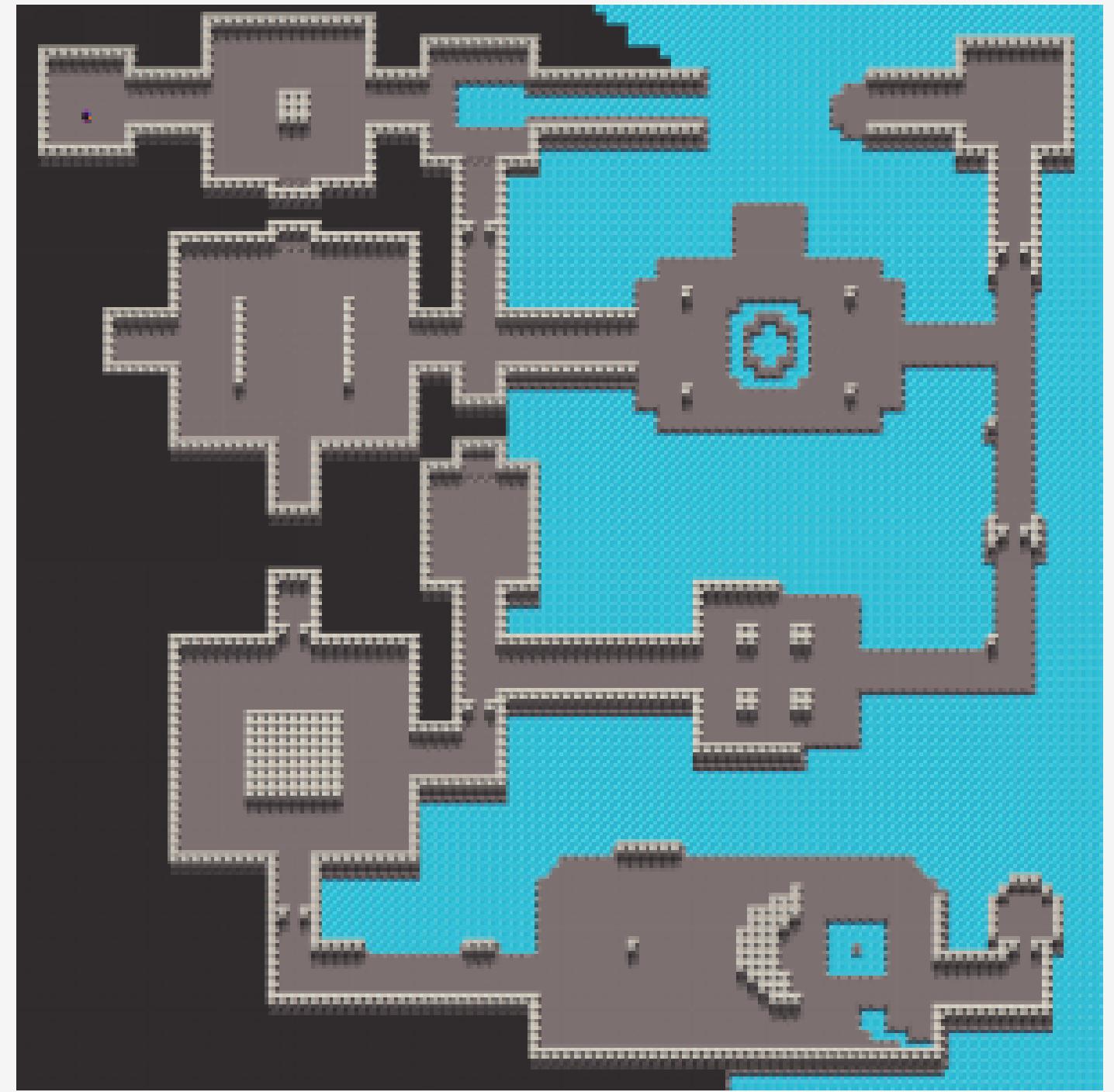


Mana-Crystal



Red-Potion

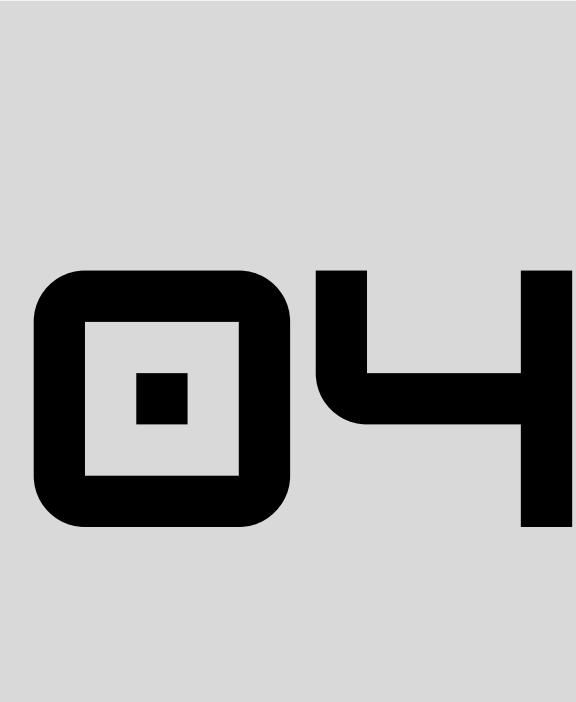
Overview Map







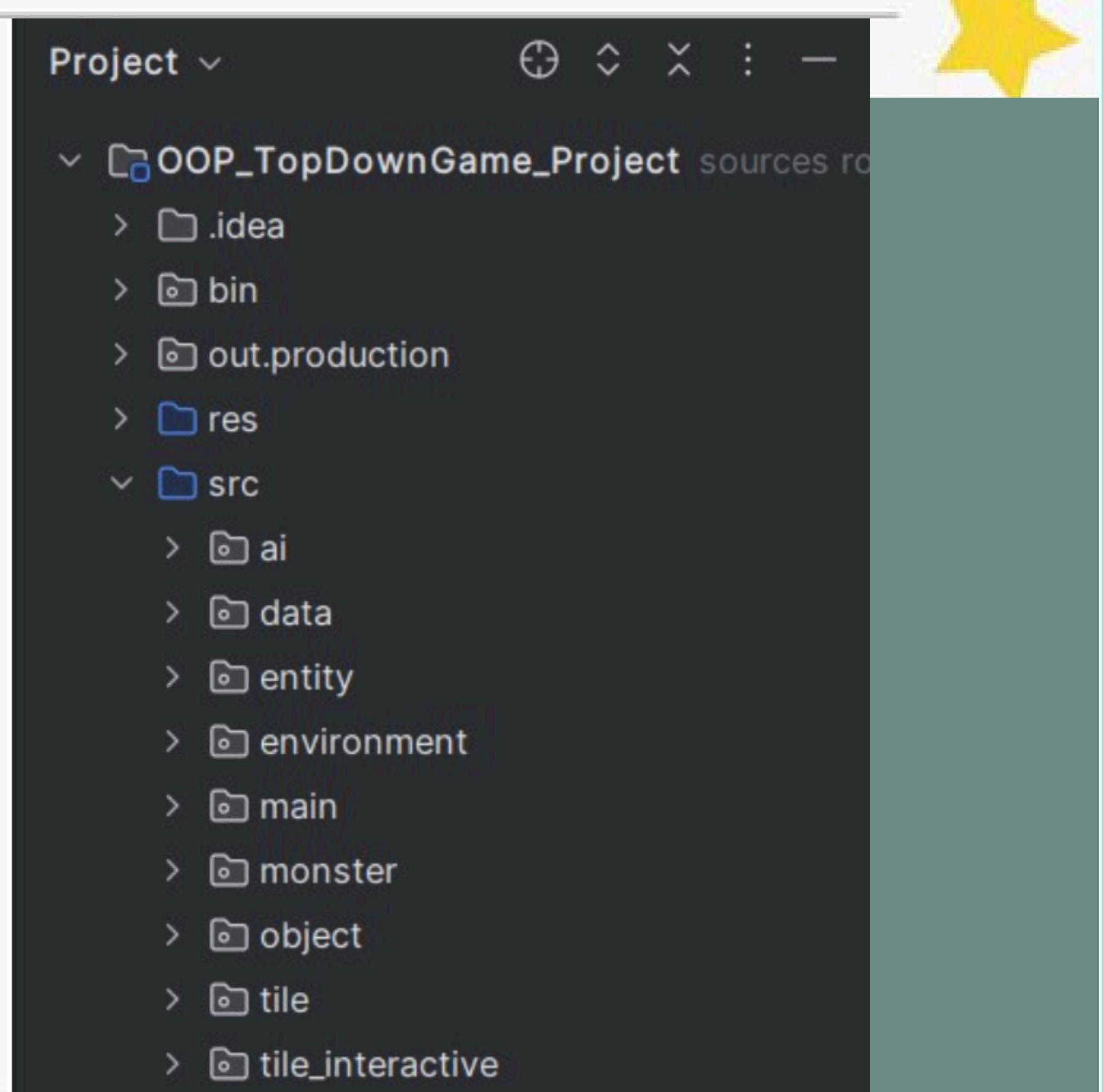
04 SYSTEM DESIGN

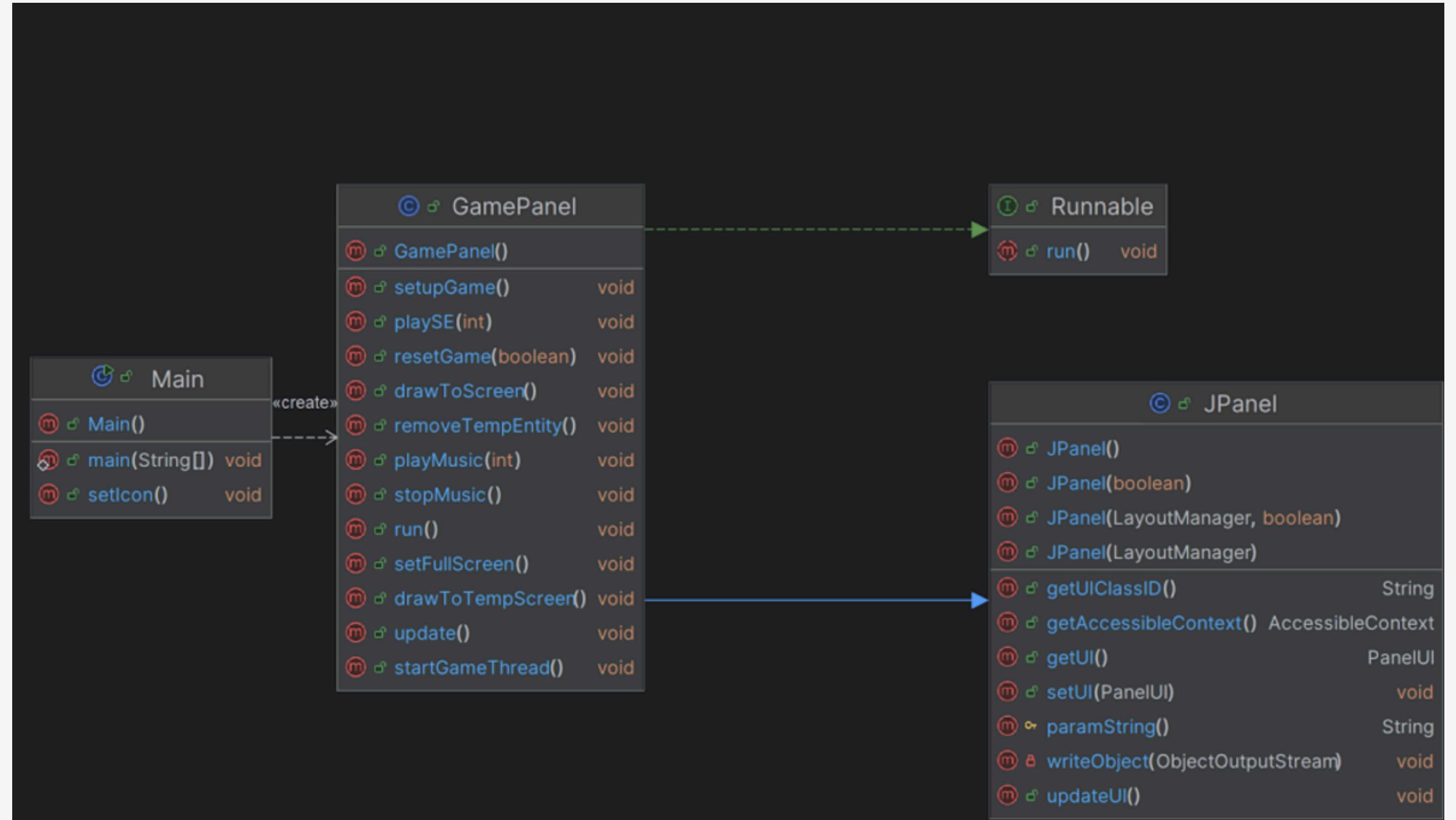


PACKAGE STRUCTURE

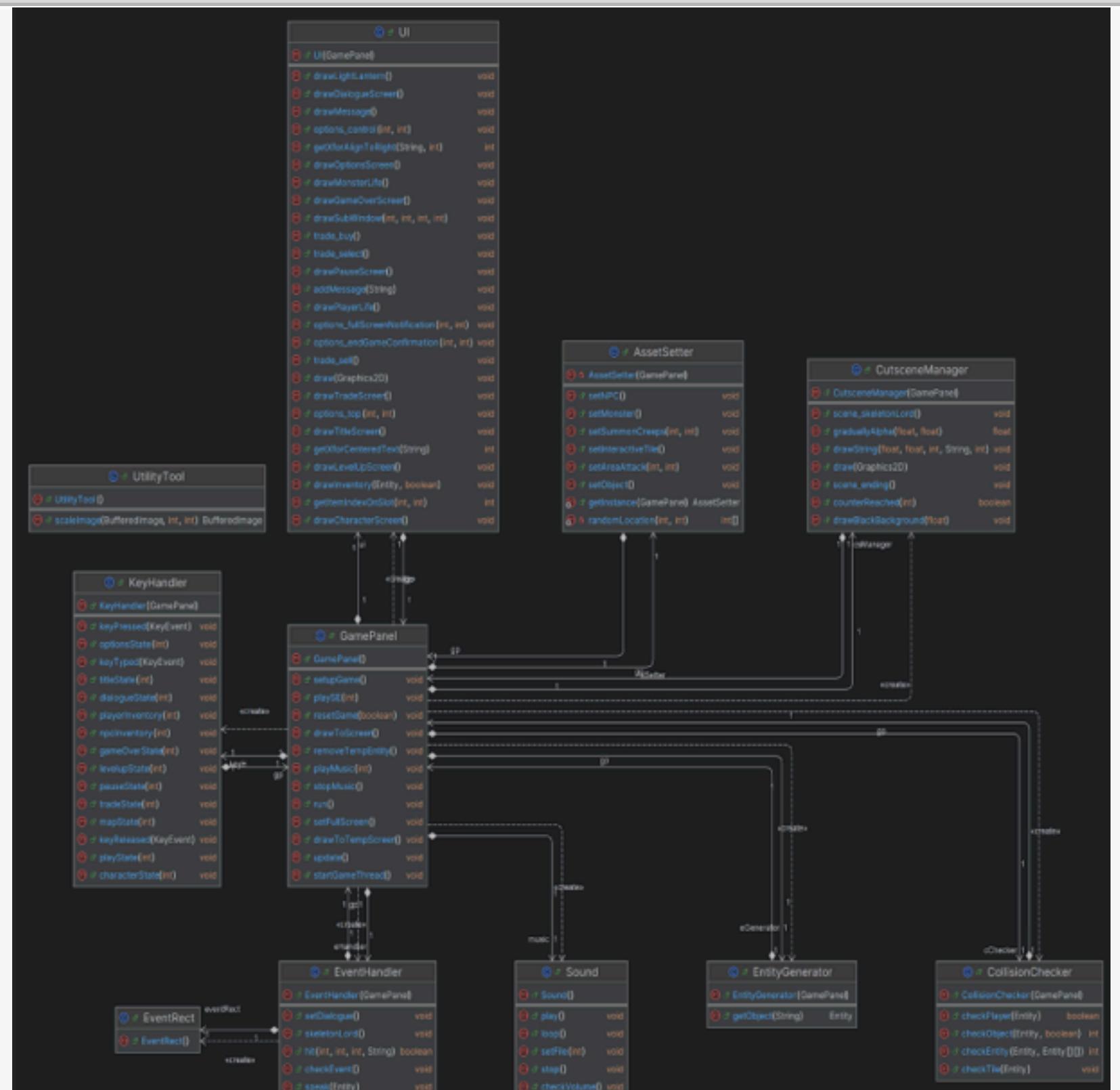
To facilitate the management of the classes, we have organized them into distinct groups, include:

- 9 packages
- 47 classes

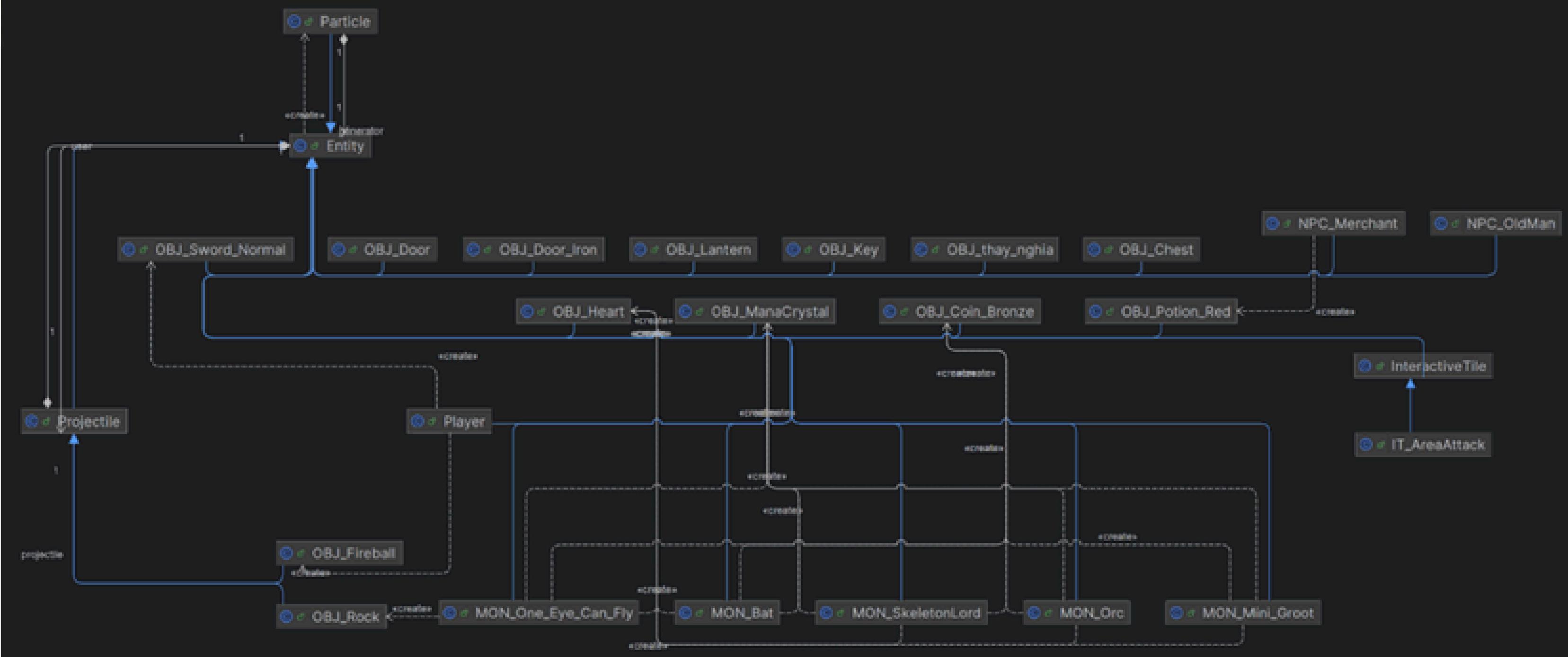




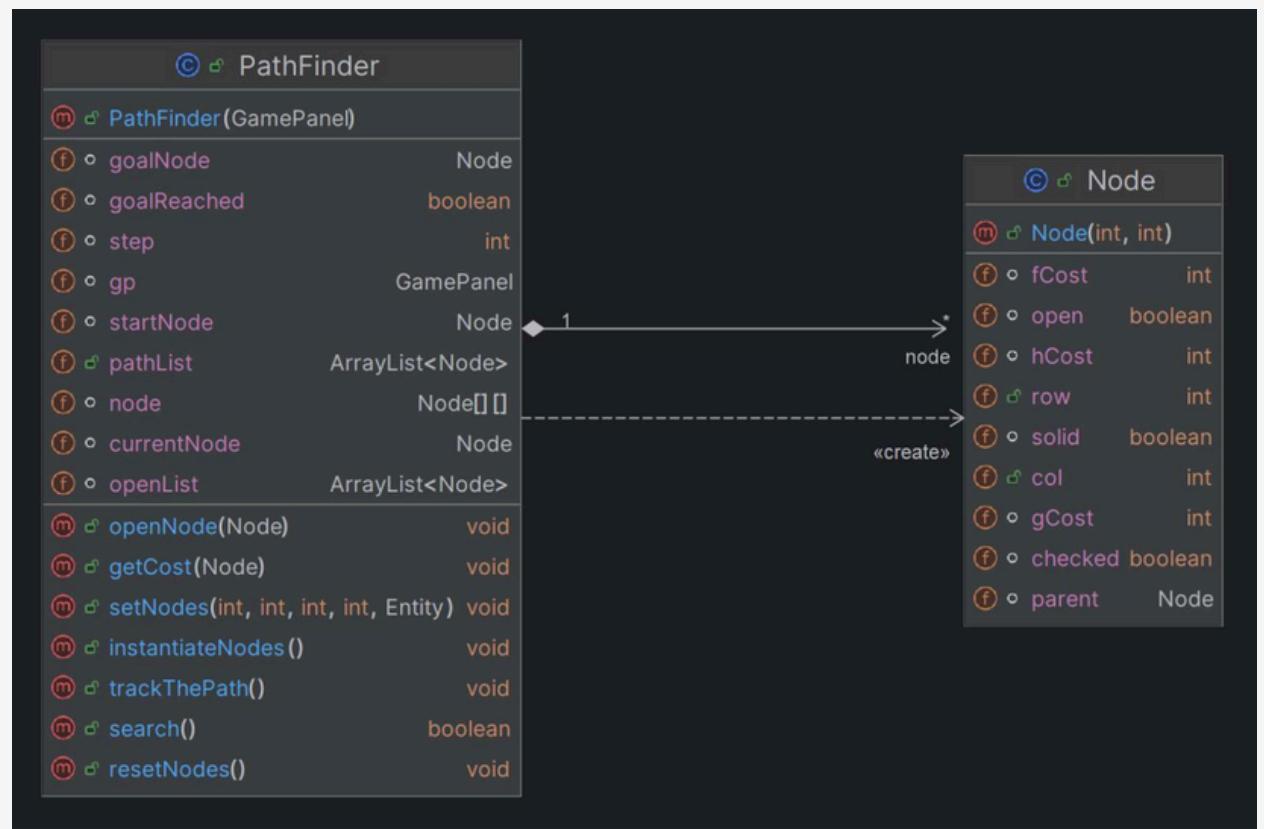
Main Diagram



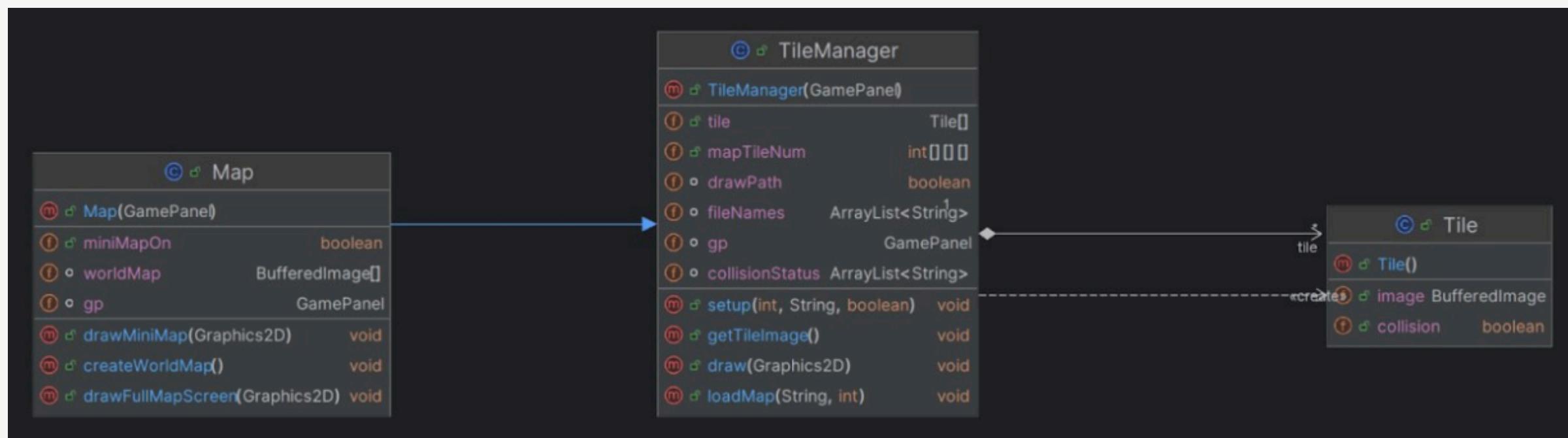
Panel Diagram



ENTITY DIAGRAM



PATH FINDING ALGORITHM DIAGRAM



MAP DIAGRAM

05 CODE FLOW



FOR STARTING GAME

```
> Main {  
  
    public static JFrame window; 13 usages  
  
>    public static void main(String[] args) {  ↳ Khoa Tap Hat +1  
        window = new JFrame();  
        window.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        window.setResizable(false); // Cant Resizable  
        window.setTitle("Saving Sir.Nghia\n"); // Window Name  
        new Main().setIcon();  
        GamePanel gamePanel = new GamePanel();  
        window.add(gamePanel);  
  
        //gamePanel.config.loadConfig();  
        if(gamePanel.fullScreenOn == true)|  
        {  
            window.setUndecorated(true);  
        }  
  
        window.pack(); // Resizes to preferred size and prevents overflow.  
  
        window.setLocationRelativeTo(null); // Starts center of screen  
        window.setVisible(true);  
  
        gamePanel.setupGame(); // Setting up the game before starts  
        gamePanel.startGameThread();  
    }  
    public void setIcon() 1 usage  ↳ Khoa Tap Hat  
    {  
        ImageIcon icon = new ImageIcon(getClass().getClassLoader().getResource( name: "player/boy_down_1.png"));  
        window.setIconImage(icon.getImage());  
    }  
}
```

MAIN CLASS

GAME PANEL CLASS

```
public void setupGame() 1 usage • Khoa Tap Hat +1
{
    aSetter.setObject();
    aSetter.setNPC();
    aSetter.setMonster();
    aSetter.setInteractiveTile();
    //The interface
    lighting.setup();

    eManager.setup();

    /*playMusic(0); // 0 = BlueBoyAdventure.wav
    stopMusic();*/
    gameState = titleState;
    //FOR FULLSCREEN
    tempScreen = new BufferedImage(screenWidth,screenHeight,BufferedImage.TYPE_INT_ARGB); //b
    g2 = (Graphics2D) tempScreen.getGraphics(); // g2 attached to this tempScreen. g2 will dra
    if(fullScreenOn == true)
    {
        setFullScreen();
    }
}
```

GAMEPANEL

```
public void run()
{
    double drawInterval = 1000000000/FPS;
    double delta = 0;
    long lastTime = System.nanoTime();
    long currentTime;
    //long timer = 0;
    //int drawCount = 0;

    while(gameThread != null)
    {
        currentTime = System.nanoTime();

        delta += (currentTime - lastTime) / drawInterval;
        //timer += currentTime - lastTime;
        lastTime = currentTime;
        if(delta >= 1)
        {
            update();
            /*repaint(); COMMENTED FOR FULL SCREEN*/
            drawToTempScreen(); //FOR FULL SCREEN - Draw everything to the buffered image
            drawToScreen(); //FOR FULL SCREEN - Draw the buffered image to the screen
            delta--;
            //drawCount++;
        }
        //SHOW FPS
        /*if(timer >= 1000000000)
        {
            System.out.println("FPS:" + drawCount);
            drawCount = 0;
            timer = 0;
        }*/
    }
}
```

```
public void update() 1 usage ▾ Khoa Tap Hat +1
{
    if(gameState == playState)
    {
        ...
    }

    if(gameState == pauseState)
    {
        //nothing, just pause screen
    }
}
```

```
public void drawToTempScreen() 1 usage ▾ Khoa Tap Hat +1
{
    //DEBUG
    long drawStart = 0;
    if(keyH.showDebugText == true)
    {
        drawStart = System.nanoTime();
    }

    //TITLE SCREEN
    if(gameState == titleState)
    {
        ui.draw(g2);
    }
    //MAP SCREEN
    else if(gameState == mapState)
    {
        map.drawFullMapScreen(g2);
    }
    //OTHERS
    else
    {
        ...
    }
}
```

GAMEPANEL

```
//GAME STATE
public int gameState; 58 usages
public final int titleState = 0; 6 usages
public final int playState = 1; 18 usages
public final int pauseState = 2; 4 usages
public final int dialogueState = 3; 6 usages
public final int characterState = 4; 3 usages
public final int optionsState = 5; 3 usages
public final int gameOverState = 6; 3 usages
public final int transitionState = 7; no usages
public final int tradeState = 8; 3 usages
public final int sleepState = 9; 1 usage
public final int mapState = 10; 3 usages
public final int cutsceneState = 11; 5 usages
public final int levelupState = 12; 3 usages
```

1st title screen

```
if(titleScreenState == 0) {
    //TITLE NAME
    g2.setFont(g2.getFont().deriveFont(Font.BOLD, size: 96F));
    g2.drawImage(gp.player.mainInterface, x: 0, y: 0, gp.screenWidth, gp.screenHeight, observer: null);
    String text = "Saving Sir.Nghia\n";
    int x = getXforCenteredText(text);
    int y = gp.tileSize * 3;
    //SHADOW
    g2.setColor(Color.gray);
    g2.drawString(text, x: x+5, y: y+5);
    //MAIN COLOR
    g2.setColor(Color.white);
    g2.drawString(text, x, y);

    //MENU
    g2.setFont(g2.getFont().deriveFont(Font.BOLD, size: 48F));

    text = "PLAY GAME";
    x = getXforCenteredText(text);
    y += gp.tileSize * 3.5;
    g2.drawString(text, x, y);
    if(commandNum == 0) {
        g2.drawString(str: ">", x: x - gp.tileSize, y);
    }

    text = "QUIT";
    x = getXforCenteredText(text);
    y += gp.tileSize;
    g2.drawString(text, x, y);
    if(commandNum == 1) {
        g2.drawString(str: ">", x: x - gp.tileSize, y);
    }
}
```

/SECOND SCREEN

```
(gp.ui.titleScreenState == 0) {
    if (code == KeyEvent.VK_W) {
        gp.ui.commandNum--;
        if (gp.ui.commandNum < 0) {
            gp.ui.commandNum = 2;
        }
    }
    if (code == KeyEvent.VK_S) {
        gp.ui.commandNum++;
        if (gp.ui.commandNum > 2) {
            gp.ui.commandNum = 0;
        }
    }
    if (code == KeyEvent.VK_ENTER) {
        if (gp.ui.commandNum == 0) {
            gp.ui.titleScreenState = 1; // Character class selection screen
            //gp.gameState = gp.playState;
        }
        if (gp.ui.commandNum == 1) {
            System.exit(status: 0);
        }
    }
}
```

1st title screen



2nd title screen

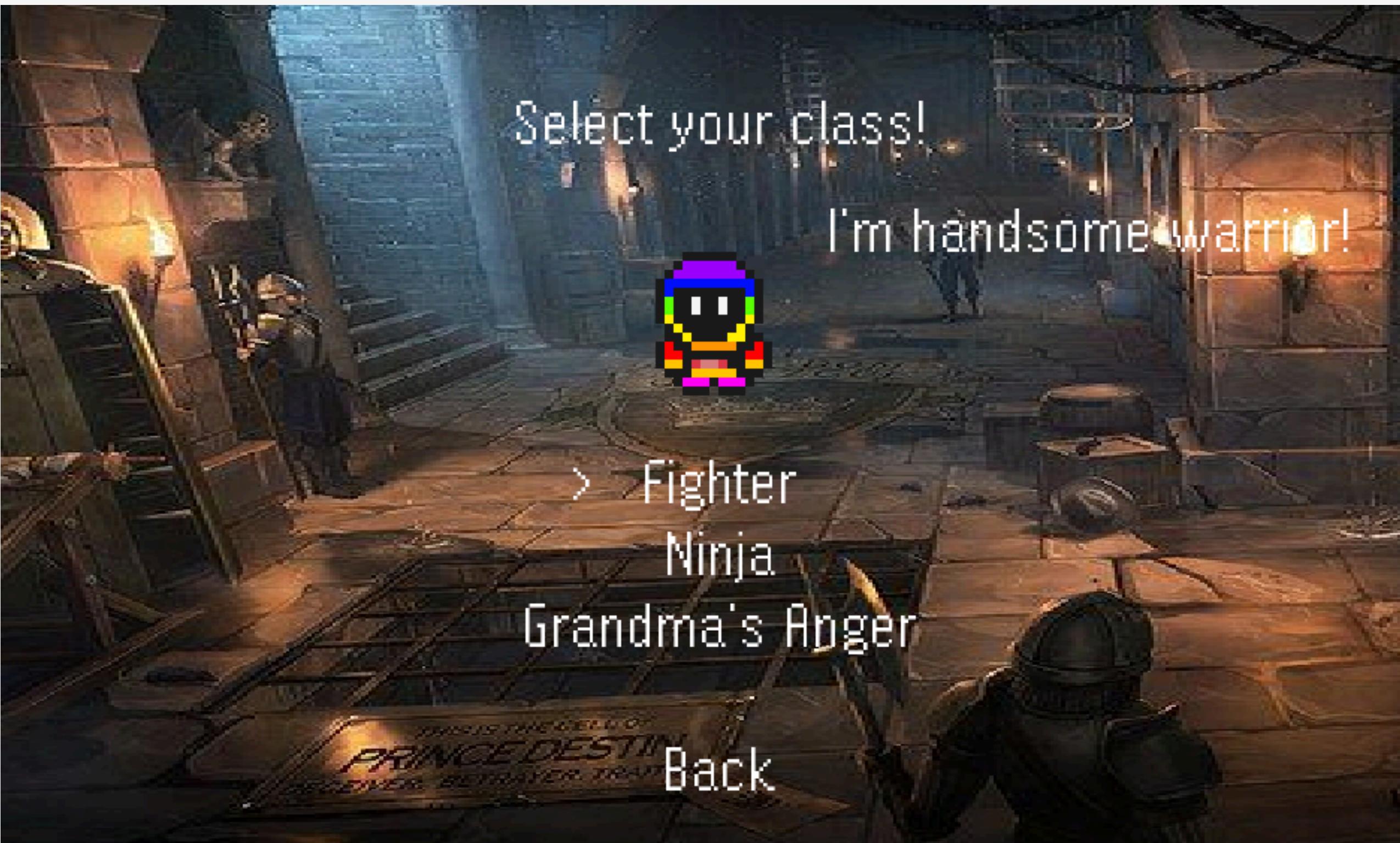
```
if (code == KeyEvent.VK_ENTER) {
    //FIGHTER
    if (gp.ui.commandNum == 0) {
        System.out.println("Do some fighter specific stuff!");
        gp.player = new Player(gp, keyH: this, characterChoice: 1);
        gp.gameState = gp.playState;
        //Khoa
        //gp.setCharacterChoice(1);
        //System.out.println("KeyHandler "+gp.setCharacterChoice(1));
        //
        //gp.gameState = gp.playState;
        gp.playMusic( i: 0);
    }
    //THIEF
    if (gp.ui.commandNum == 1) {
        System.out.println("Do some thief specific stuff!");
        gp.player = new Player(gp, keyH: this, characterChoice: 2);
        gp.gameState = gp.playState;
        //Khoa
        //gp.setCharacterChoice(2);
        //System.out.println("KeyHandler "+gp.setCharacterChoice(2));
        //
        //gp.gameState = gp.playState;
        gp.playMusic( i: 0);
    }
    //SORCERER
    if (gp.ui.commandNum == 2) {
        System.out.println("Do some sorcerer specific stuff!");
        gp.player = new Player(gp, keyH: this, characterChoice: 3);
        gp.gameState = gp.playState;
        gp.playMusic( i: 0);
    }
    //BACK
    if (gp.ui.commandNum == 3) {
        gp.ui.titleScreenState = 0;
    }
}
```

```
public Player(GamePanel gp, KeyHandler keyH, int characterChoice)  5 usages  ± Khoa Tap Hat +1
{
    super(gp); // calling constructor of super class(from entity class)
    this.keyH=keyH;
    this.characterChoice = characterChoice;

    public void getImage(int characterChoice )  ± Khoa Tap Hat +1
    {
        switch(characterChoice){...}
    }

    public void playState(int code)  1 usage  ± Khoa Tap Hat +1
    {...}
    public void pauseState(int code)  1 usage  ± Khoa Tap Hat
    {...}
    public void dialogueState(int code)  1 usage  ± Khoa Tap Hat
    {...}
    public void characterState(int code)  1 usage  ± Khoa Tap Hat
    {...}
    public void optionsState(int code)  1 usage  ± Khoa Tap Hat
    {...}
    public void gameOverState(int code)  1 usage  ± Khoa Tap Hat
    {
```

2nd title screen



Game Over Screen

```
585     if(keyH.godModeOn == false)
586     {
587         if(life <= 0)
588         {
589             gp.gameState = gp.gameOverState;
590             gp.ui.commandNum -= 1; //for if you die while pressing enter
591             gp.stopMusic();
592             gp.playSE(i: 12);
593         }
594     }
```

```
public void gameOverState(int code) 1 usage ▲ Khoa Tap Hat
{
    if(code == KeyEvent.VK_W)
    {
        gp.ui.commandNum--;
        if(gp.ui.commandNum < 0)
        {
            gp.ui.commandNum = 1;
        }
        gp.playSE(i: 9);
    }
    if(code == KeyEvent.VK_S)
    {
        gp.ui.commandNum++;
        if(gp.ui.commandNum > 1)
        {
            gp.ui.commandNum = 0;
        }
        gp.playSE(i: 9);
    }
    if(code == KeyEvent.VK_ENTER)
    {
        if(gp.ui.commandNum == 0) //RETRY, reset position, life, mana, monsters, npcs...
        {
            gp.gameState = gp.playState;
            gp.resetGame(restart: false);
            gp.playMusic(i: 0);
        }
        else if(gp.ui.commandNum == 1) //QUIT, reset everything
        {
            gp.ui.titleScreenState = 0;
            gp.gameState = gp.titleState;
            gp.resetGame(restart: true);
        }
    }
}
```

```
public void drawGameOverScreen() 1 usage ▲ Khoa Tap Hat *
{
    g2.setColor(new Color(r: 0, g: 0, b: 0, a: 150)); //Half-black
    g2.fillRect(x: 0, y: 0, gp.screenWidth, gp.screenHeight);

    int x;
    int y;
    String text;
    g2.setFont(g2.getFont().deriveFont(Font.BOLD, size: 110f));
    text = "Game Over";
    //Shadow
    g2.setColor(Color.BLACK);
    x = getXforCenteredText(text);
    y = gp.tileSize * 4;
    g2.drawString(text, x, y);
    //Text
    g2.setColor(Color.white);
    g2.drawString(text, x: x-4, y: y-4);
    //RETRY
    g2.setFont(g2.getFont().deriveFont(size: 50f));
    text = "Retry";
    x = getXforCenteredText(text);
    y += gp.tileSize * 4;
    g2.drawString(text, x, y);
    if(commandNum == 0)
    {
        g2.drawString(str: ">", x: x-40, y);
    }
    //BACK TO THE TITLE SCREEN
    text = "Quit";
    x = getXforCenteredText(text);
    y += 55;
    g2.drawString(text, x, y);
    if(commandNum == 1)
    {
        g2.drawString(str: ">", x: x-40, y);
    }
}
```

Game Over Screen



Check Event

```
public void checkEvent() 1 usage  ▲ Khoa Tap Hat
{
    //Check if the player character is more than 1 tile away from the last event
    int xDistance = Math.abs(gp.player.worldX - previousEventX); //pure distance
    int yDistance = Math.abs(gp.player.worldY - previousEventY);
    int distance = Math.max(xDistance, yDistance); //returns greater value
    if(distance > gp.tileSize)
    {
        canTouchEvent = true;
    }

    if(canTouchEvent == true)
    {
        //if(hit(0,23,12, "any") == true) {healingPool(gp.dialogueState);}
        //else if(hit(0,27,16, "right") == true) {damagePit(gp.dialogueState);}
        //else if(hit(0,10,39, "any") == true) {teleport(1,12,13, gp.indoor);} //to merchant's house
        //else if(hit(1,12,13, "any") == true) {teleport(0,10,39, gp.outside);} //to outside
        if(hit( map: 0, col: 42, row: 44, reqDirection: "any") == true) {speak(gp.npc[1][0]);} //merchant

        //else if(hit(0,12,9, "any") == true) {teleport(2,9,41, gp.dungeon);} //to the dungeon
        //else if(hit(2,9,41, "any") == true) {teleport(0,12,9, gp.outside);} //to outside
        //else if(hit(2,8,7, "any") == true) {teleport(3,26,41, gp.dungeon);} //to B2
        //else if(hit(3,26,41, "any") == true) {teleport(2,8,7, gp.dungeon);} //to B1
        else if(hit( map: 0, col: 47, row: 88, reqDirection: "any") == true) {skeletonLord();} //BOSS
        else if(hit( map: 0, col: 47, row: 89, reqDirection: "any") == true) {skeletonLord();} //BOSS
        else if(hit( map: 0, col: 47, row: 90, reqDirection: "any") == true) {skeletonLord();} //BOSS
    }
}

public boolean hit(int map, int col, int row, String reqDirection) 4 usages  ▲ Khoa Tap Hat
{...}
```

```
public void scene_skeletonLord() 1 usage  ▲ Khoa Tap Hat
{
    if(scenePhase == 0)
    {
        gp.bossBattleOn =true;

        //Shut the iron door to trap player
        for(int i = 0; i < gp.obj[1].length; i++) //Search a vacant slot for the iron door
        {
            if(gp.obj[gp.currentMap][i] == null) {...}
        }
        /* ... */

        //Reset
        sceneNum = NA;
        scenePhase = 0;
        gp.gameState = gp.playState;

        //Change the music
        gp.stopMusic();
        gp.playMusic( i: 22);
    }
}
```

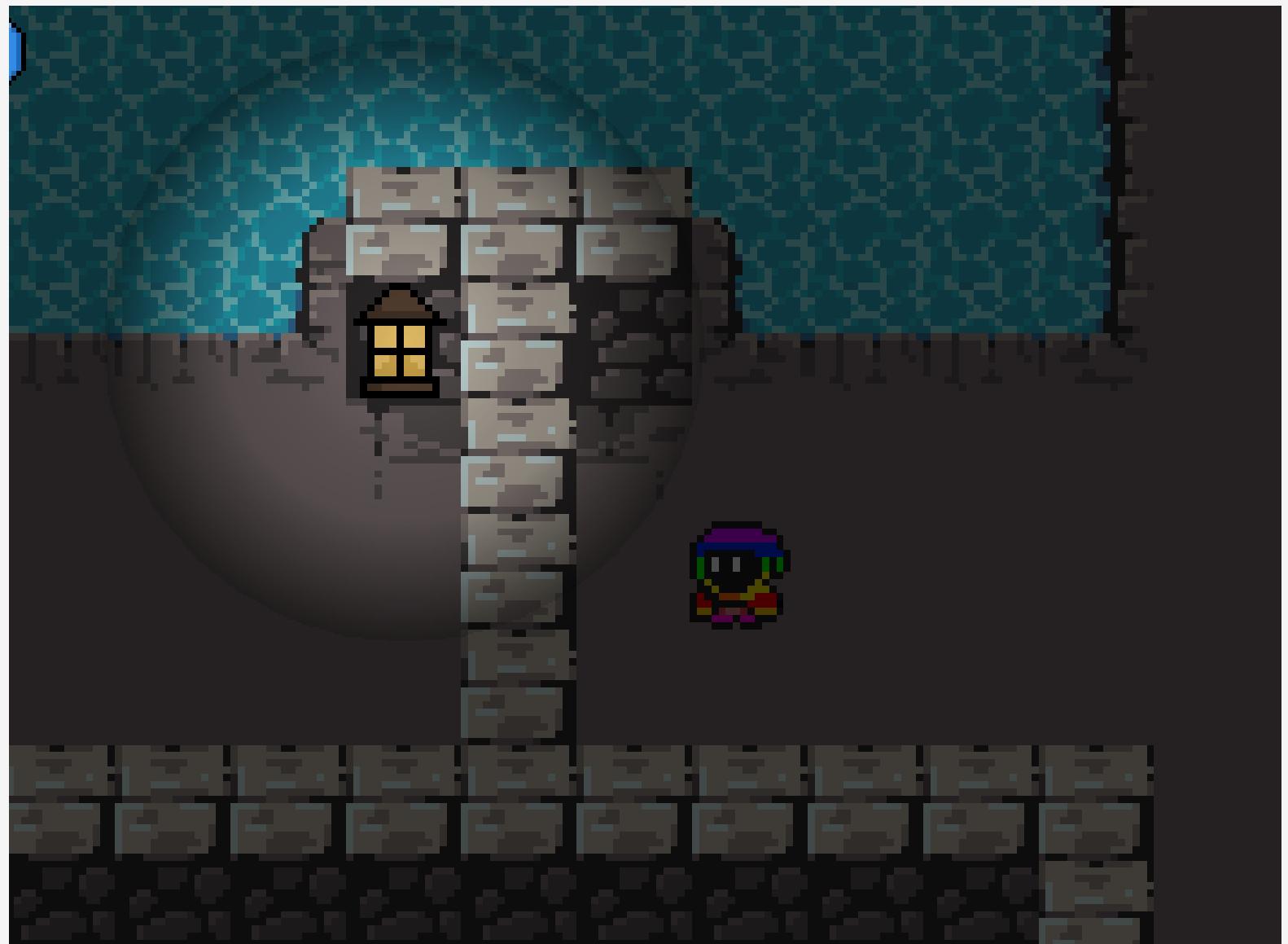
Check Event

```
public void checkDrop() 1 usage ▾ Khoa Tap Hat
{
    gp.bossBattleOn = false;
    Progress.skeletonLordDefeated = true;

    //Restore the previous music
    gp.stopMusic();
    gp.playMusic( i: 19);

    // Remove the iron doors
    for(int i = 0; i < gp.obj[1].length; i++)
    {
        if(gp.obj[gp.currentMap][i] != null && gp.obj[gp.currentMap][i].name.equals(OBJ_Door_Iron.objName))
        {
            gp.playSE( i: 21);
            gp.obj[gp.currentMap][i] = null;
        }
    }
}
```

Check Event



End game

```
public class OBJ_thay_nghia extends Entity { 5 usages + Khoa Tap Hat

    GamePanel gp; 5 usages
    public static final String objName = "Blue Heart";
    public OBJ_thay_nghia(GamePanel gp) 3 usages + Khoa Tap Hat
    {
        super(gp);
        this(gp);
        type = type_pickupOnly;
        name = objName;
        down1 = setup( imagePath: "/objects/thay_nghia", gp.tileSize, gp.tileSize);
        setDialogues();
    }
    public void setDialogues() 1 usage + Khoa Tap Hat
    {
        dialogues[0][0] = "You found Teacher Nghia.";
        dialogues[0][1] = "You successfully rescued him from the dungeon.";
    }
    public boolean use(Entity entity) //when pickup this method will be called + Khoa Tap Hat
    {
        gp.gameState = gp.cutsceneState;
        gp.csManager.sceneNum = gp.csManager.ending;
        return true;
    }
}
```

```
public void scene_ending()
{...}
```

After the fierce battle with the Skeleton Lord,
Teacher Nghia finally found.
To thank you for the rescue,
Teacher Nghia gave you 100 points for the project.

Mission complete...

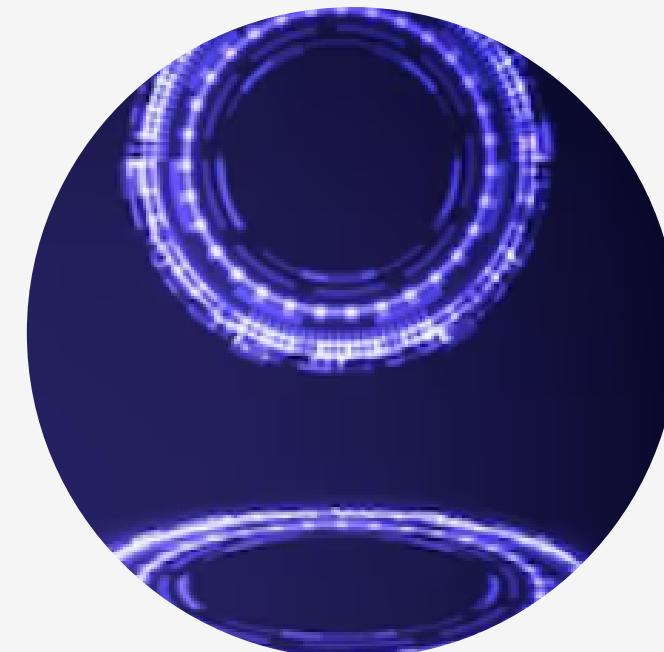
CONCLUSION AND FUTURE WORKS

06

Conclusion

- *Proficient in object-oriented programming, Java programming knowledge and the SOLID principles*
- *The ability to work in groups, search and select open sources documents is enhanced*
- *Gain more knowledge*
- *Experience the joy that game programming brings*

FUTURE ROADMAP



TELEPORT



THROWING
ITEM



Customize
characters

REFERENCES

- [1] BlueBoyAdvanture (2022).RyiSnow/blueboyadvanture[Java]. Retrieved May 10, 2024, from [here](#)
- [2] Java Swing Tutorial. Javatpoint. (n.d.). Retrieved December 11, 2023, from <https://www.javatpoint.com/java-swing>
- [3] UML Diagrams Tutorial. Javatpoint. (n.d.). Retrieved January 4, 2023, from <https://www.javatpoint.com/uml-diagrams>
- [4] Drew, O. (2022). Othneildrew/Best-README-Template. Retrieved January 4, 2023, from <https://github.com/othneildrew/Best-README-Template>
- [5] SOLID Principle in Programming: Understand With Real Life Examples. GeeksforGeeks. Retrieved December 28, 2023, from <https://www.geeksforgeeks.org/solid-principle-in-programming-understand-with-real-life-examples/>
- [*] aztharis.itch.io/tilemap-mini-dungeon, from [here](#)

**THANK YOU FOR
LISTENING!**