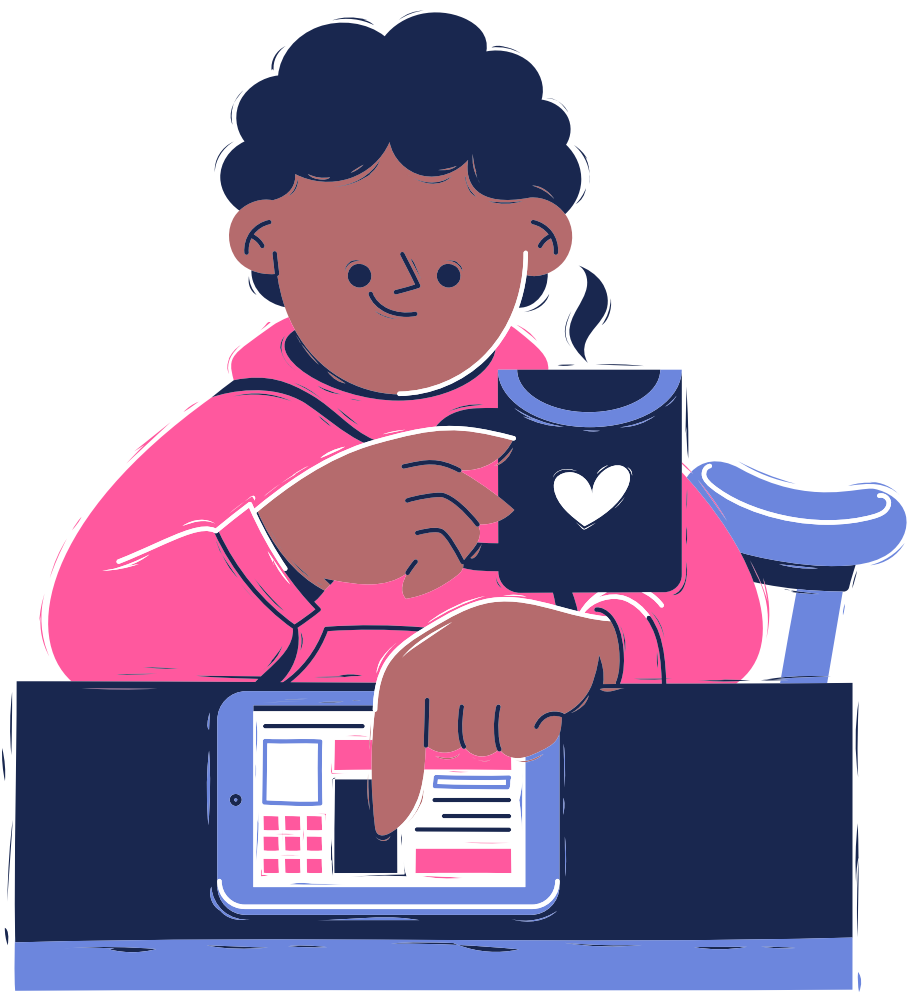


程式設計一期末

出埃及記額外功能

413570012林宜庭

功能增加



- WASD按鍵朝四個方向發射子彈
- 自主追殺摩西的ChasingSprite
- 計時器
- 拿Rifle使ChasingSprite停止移動

自主追殺摩西的 ChasingSprite



ChasingSprite Class

```
package com.company.Sprite;

> import ...

public class ChasingSprite extends Sprite { 3 usages
    private Point position; 13 usages
    private Moses moses; // 摩西對象的引用 3 usages
    public boolean active; // 標記是否有效 5 usages

    public ChasingSprite(int x, int y, Moses moses) { 1 usage
        this.position = new Point(x, y);
        this.moses = moses;
        this.active = true; // 初始狀態為有效
    }

    // 自動追蹤摩西的行為
    public void moveTowardsMoses() { 1 usage
        if (!active) return; // 如果無效，停止動作
        Point mosesPosition = moses.getRelativePosition();
```

```
        // 判斷上下左右的移動方向，優先向摩西移動
        if (mosesPosition.x > position.x) {
            position.x += 1; // 向右移動
        } else if (mosesPosition.x < position.x) {
            position.x -= 1; // 向左移動
        }

        if (mosesPosition.y > position.y) {
            position.y += 1; // 向下移動
        } else if (mosesPosition.y < position.y) {
            position.y -= 1; // 向上移動
        }
    }
}
```

ChasingSprite Class

```
37      @Override 46 usages
38      public Point getRelativePosition() {
39          return active ? position : null; // 返回位置 (如果無效返回 null)
40      }
41
42      @Override
43      public String overlap(int x,int y){return null;}
44
45      @Override 3 usages
46      public void draw(Graphics g) {
47          if (!active) return; // 如果無效，停止繪製
48          g.setColor(Color.RED); // 可以選擇不同顏色來表示敵人
49          g.fillRect(x: position.x * Main.CELL, y: position.y * Main.CELL, Main.CELL, Main.CELL);
50      }
51
52      // 判斷這個敵人是否碰到摩西
53      public boolean checkCollision() { 1 usage
54          return active && position.equals(moses.getRelativePosition());
55      }
56
57  }
```

自主追殺摩西的ChasingSprite



計時器 與 拿Rifle使 ChasingSprite停止移動



摘要

研究不僅僅是使用主題或問題來尋找參考文獻。 以下是在開始研究時可以派上用場的一些技能。

確定和使用可靠來源

總是通過審查資源類型、作者、網站連結或通用資源定位器 (URL)，以及是否最新進行快速信譽測試。

將搜尋詞與修飾符相結合

使用關鍵構想作為搜尋詞，並結合修飾符，如 以及、或和引號（“”）。

略讀和瀏覽

使用這些方法將幫助你通過更多資源，並快速確定最適合你的主題。

Moses Class

public String overlap(int x, int y)

else if(Main.gameView instanceof TenCommandmentsView)

```
        for(Rifle r : rifle) {
            if (r.getRelativePosition() != null && x == r.getRelativePosition().x && y == r.getRelativePosition().y) {
                hasRifle = true; // Moses picks up the rifle
                r.setNullPosition();
            }
        }
        return "none";
    }
    public boolean hasRifleCheck() { 1 usage
        return hasRifle;
    }
```

Main Class

```
private Timer timer; 6 usages
private long startTime; 6 usages
private ChasingSprite chasingSprite; // 新增一個追蹤摩西的敵人
```

```
public void resetGame(GameView game, boolean resetTimer) { 8 usages
    moses = new Moses(x: 1, y: 1);
    gameView = game;
    chasingSprite = new ChasingSprite(x: -13, y: 1, moses);
    if (resetTimer) {
        startTimer(); // 只有在遊戲重置時才重新啟動計時器
    }

    repaint();
}

private void startTimer() { 1 usage
    startTime = System.currentTimeMillis(); // 記錄開始時間
    if (timer != null) {
        timer.stop(); // 確保先停止舊的計時器
    }
    timer = new Timer(delay: 1000, e -> repaint()); // 每秒刷新一次畫面
    timer.start();
}
```

```
private void stopTimer() { 4 usages
    if (timer != null) {
        timer.stop(); // 停止計時
    }
}
```

```
@Override
public void paintComponent(Graphics g) {
    super.paintComponent(g); // 確保清除舊畫面
    gameView.drawView(g);
    moses.draw(g);
    chasingSprite.draw(g); // 畫出敵人
    // 顯示計時器 (整數秒數)
    g.setColor(Color.BLACK);
    g.setFont(new Font(name: "Arial", Font.BOLD, size: 16));
    long elapsedTime = (System.currentTimeMillis() - startTime) / 1000; // 直接顯示整數秒數
    g.drawString(str: "時間: " + elapsedTime + " 秒", x: 10, y: 20);
    requestFocusInWindow();
}
```

Main Class

keyPressed

```
if (result.equals("Game over")) {  
    stopTimer(); // 停止計時  
    long elapsedTime = (System.currentTimeMillis() - startTime) / 1000; // 計算花費時間  
    JOptionPane.showMessageDialog(parentComponent: this, message: "你完成了遊戲！花費時間： " + elapsedTime + " 秒");  
    return; // 防止繼續進行
```

```
if (!moses.hasRifleCheck()) {  
    //if (true) {  
  
    // 讓敵人自動追蹤摩西  
    chasingSprite.moveTowardsMoses();  
    // 檢查是否有碰到摩西  
    if (chasingSprite.checkCollision()) {  
        JOptionPane.showMessageDialog(parentComponent: this, message: "You were caught by the enemy!");  
        // 重置遊戲  
        level = 1;  
        resetGame(new DisasterView(), resetTimer: true);  
        return;  
    }  
}
```


計時器



拿Rifle使 ChasingSprite停止移動





影片連結