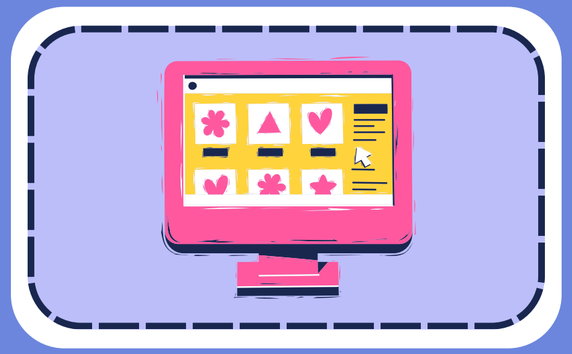

程式設計二

貪食蛇遊戲

413570012林宜庭

功能增添摘要

道具修改

- 草莓：加一分與一格長度、蛇身變色
- 蘋果：加一分
- 跑鞋：加速
- 毒蘋果：扣一格長度
- 炸彈：遊戲結束

玩法添加

- 每十秒減少蛇身長度，長度為0則遊戲結束
- 時間歸零或撞牆則遊戲結束

畫面顯示

- 新增蛇頭上的對話泡，特定道具有獨特訊息
- 左上角顯示目前分數與倒數180s
- 結束新增顯示剩餘時間

各功能程式碼

道具功能

Main Class
private void tick()

```
for (Item item : items) {
    if (newX == item.getX() && newY == item.getY()) {
        switch (item.getType()) {
            case APPLE:
                item.resetPosition(snake, fruit);
                score++;
                snakeMessage = "+1!";
                messageTime = System.currentTimeMillis();
                break;
            case POISON:
                //蛇身長度大於 1 才能扣一格蛇尾
                if (snakeBody.size() > 1) snakeBody.remove( index: snakeBody.size() - 1);
                item.resetPosition(snake, fruit);
                snakeMessage = "Ouch!";
                messageTime = System.currentTimeMillis();
                break;
            case BOMB:
                gameOver();
                return;
            case SHOES:
                tickSpeed = Math.max(50, tickSpeed - 50); //遊戲加速（最低為 50ms）
                resetGameTimer(); //重設 timer 套用新速度
                item.resetPosition(snake, fruit);
                snakeMessage = "Zoom!";
                messageTime = System.currentTimeMillis();
                break;
        }
    }
}
```

Item Class
public void resetPosition(Snake snake, Fruit fruit, ArrayList<Item> existingItems)

```
do {
    newX = rand.nextInt(Main.column) * Main.CELL_SIZE;
    newY = rand.nextInt(Main.row) * Main.CELL_SIZE;
    overlap = false;

    // 與蛇重疊
    for (Node node : snake.getSnakeBody()) {
        if (node.x == newX && node.y == newY) {
            overlap = true;
            break;
        }
    }

    // 與水果重疊
    for (Node fruitNode : fruit.getFruits()) {
        if (fruitNode.x == newX && fruitNode.y == newY) {
            overlap = true;
            break;
        }
    }

    // 與其他 item 重疊
    for (Item other : existingItems) {
        if (other.getX() == newX && other.getY() == newY) {
            overlap = true;
            break;
        }
    }

} while (overlap);
```

吃草莓蛇身變色與對話泡訊息

Main Class private void tick()

```
if (fruit.checkAndReplaceIfEaten(newX, newY, snake, items)) {
    score++;
    snakeColor = new Color(rand.nextInt( bound: 256),
            rand.nextInt( bound: 256), rand.nextInt( bound: 256)); //產生新隨機顏色並指定為目前蛇身顏色
    snakeMessage = messages[rand.nextInt(messages.length)]; //顯示一段隨機訊息
    messageTime = System.currentTimeMillis(); //控制訊息顯示時間
    grew = true;
}
```

public void paintComponent (Graphics g)

```
if (!snakeMessage.isEmpty()) { //如果目前有訊息內容（非空字串）才顯示泡泡
    //設定氣泡格式與計算氣泡出現位置
    g.setColor(new Color( r: 255,  g: 255,  b: 255,  a: 200));
    int bubbleWidth = g.getFontMetrics().stringWidth(snakeMessage) + 20;
    int bubbleHeight = 30;
    int bubbleX = snake.getSnakeBody().get(0).x + (CELL_SIZE / 2) - (bubbleWidth / 2);
    int bubbleY = snake.getSnakeBody().get(0).y - 30;
    g.fillRoundRect(bubbleX, bubbleY, bubbleWidth, bubbleHeight,  arcWidth: 10,  arcHeight: 10);
    //設定文字為黑色並畫出訊息文字
    g.setColor(Color.BLACK);
    g.drawString(snakeMessage,  x: bubbleX + 10,  y: bubbleY + (bubbleHeight / 2) + 5);
}
```

撞牆、蛇身長度0、時間歸零遊戲結束

Main Class
private void tick()

```
if (newX < 0 || newY < 0 || newX >= width || newY >= height) {
    gameOver();
    return;
}
```

private void setTimers()

```
effectTimer = new Timer();
effectTimer.scheduleAtFixedRate(new TimerTask() {
    @Override
    public void run() {
        timeLeft--;
        //每 10 秒減少一節蛇尾（如果蛇身還有）
        if (snake.getSnakeBody().size() > 0 && timeLeft % 10 == 0) {
            snake.getSnakeBody().remove( index: snake.getSnakeBody().size() - 1);
        }
        //如果蛇身變空或時間歸零，就遊戲結束
        if (snake.getSnakeBody().isEmpty() || timeLeft <= 0) {
            gameOver();
        }
    }
}, delay: 1000, period: 1000);
```

左上顯示目前分數與倒數180s
結束新增顯示剩餘時間

Main Class
private paintComponent(Graphics g)

```
g.setColor(Color.WHITE);
g.drawString( str: "Score: " + score,  x: 10,  y: 20);
g.drawString( str: "Time: " + timeLeft + "s",  x: 10,  y: 40);
```

private void gameover()

```
int response = JOptionPane.showOptionDialog( parentComponent: this,  message: "Game Over! Score: " + score +
                "\nHighest Score: " + Math.max(score, highest_score) +
                "\nTime Left: " + timeLeft +
                "\nTry again?",  title: "Game Over", JOptionPane.YES_NO_OPTION, JOptionPane.INFORMATION_MESSAGE,
        icon: null,  options: null, JOptionPane.YES_OPTION);
```

展示影片連結