

糖果爭奪戰

像素風街機 * 雙人競賽

第16組 410031259 黃湘庭、
412580046 洪子晴、412580383 鍾睿芸

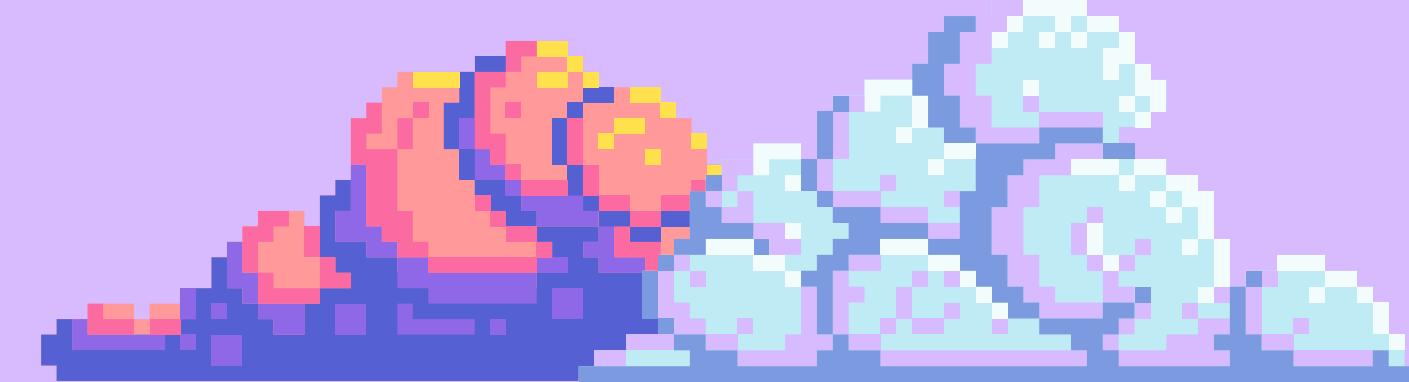
Start

OUTLINES

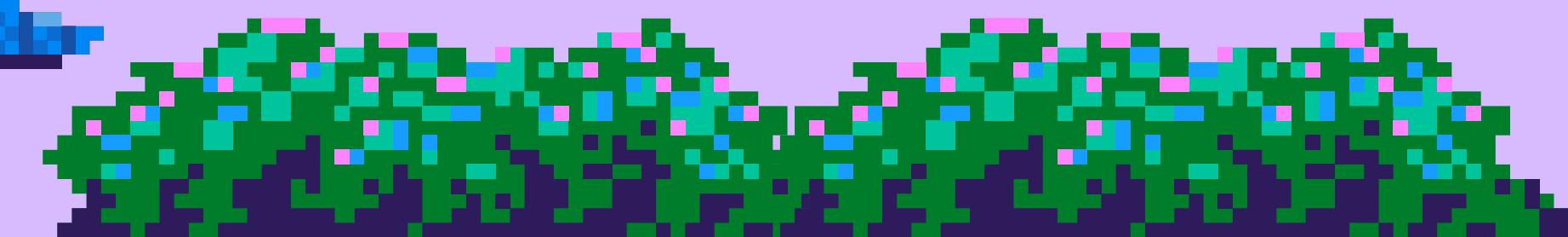
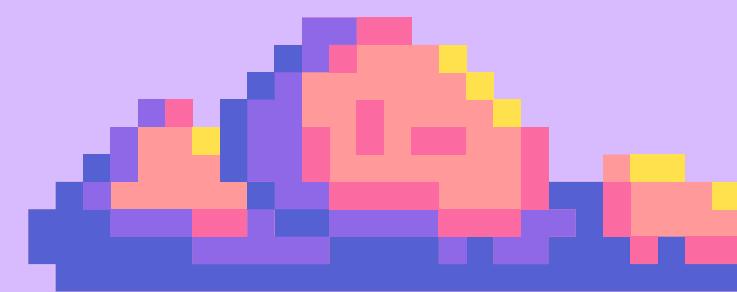
- 遊戲主題
- 遊戲規則
- 製作動機
- 使用的元素
- 用到的程式編寫技巧[課內/外]
- 展示DEMO



遊戲主題



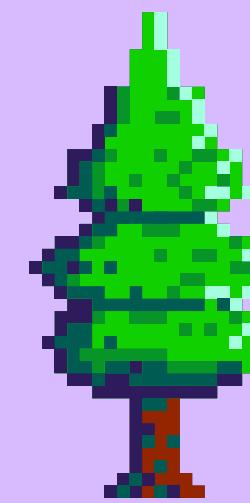
- 主題：「吃糖果 + 雙人對戰」
- 加入行進路線障礙物
- 爭奪糖果，提升腎上腺素
- 像素風格畫面，活潑有趣



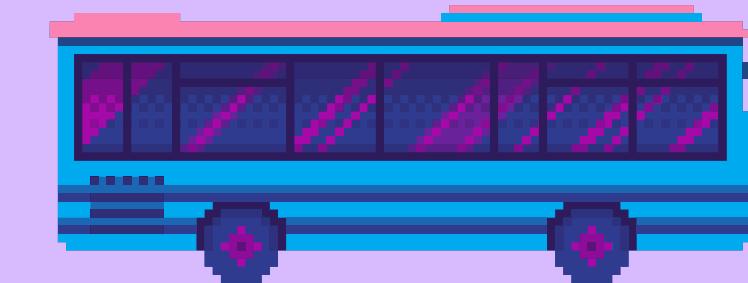
遊戲規則

- 模式：雙人對戰，1v1
- 操作方式：
 - 玩家一：WASD
 - 玩家二：↑↓←→方向鍵
- 總共2張地圖
- 倒數秒數與吃的糖果數顯示在頁面上方
- 限時30秒內，糖果吃得最多的玩家獲勝





製作動機



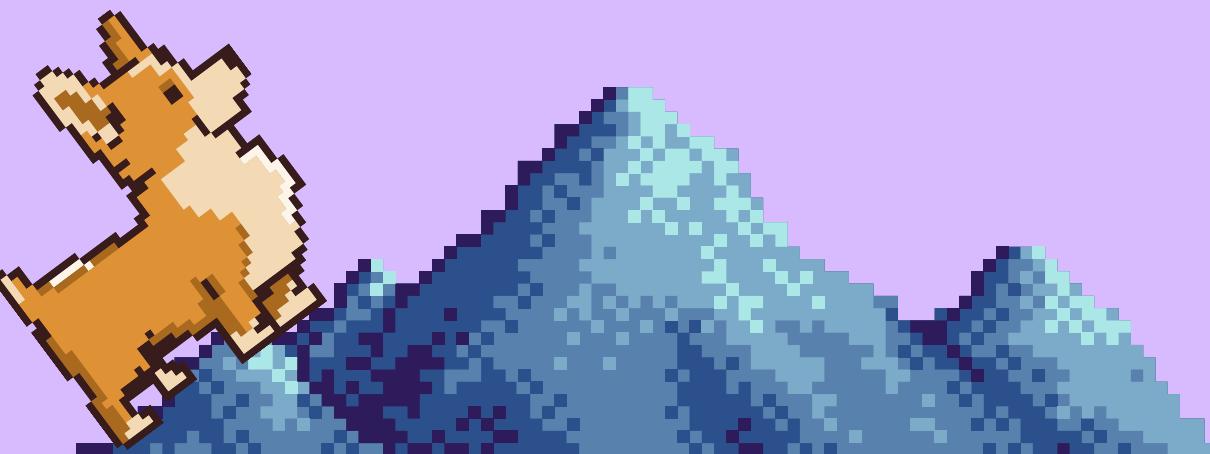
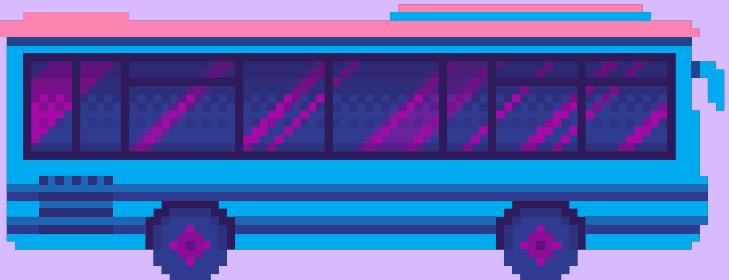
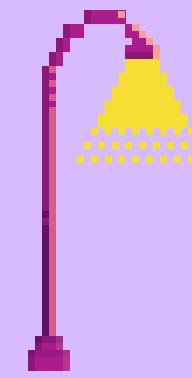
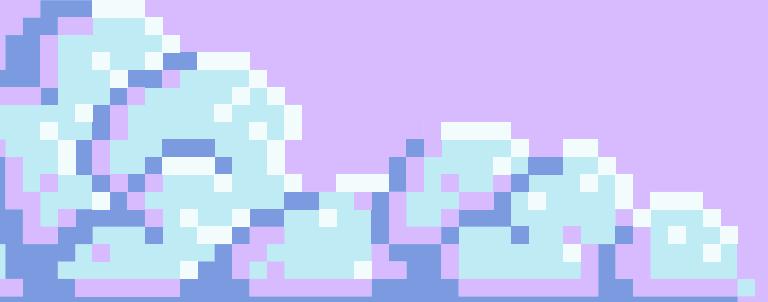
- 喜愛街機小遊戲的玩家到遊戲製作者
- 結合課堂所學，實作雙人對戰機制
- 想打造一款童趣感十足又能競爭的遊戲

使用的元素

糖果



樹



使用的元素



PLAYER 1



PLAYER 2

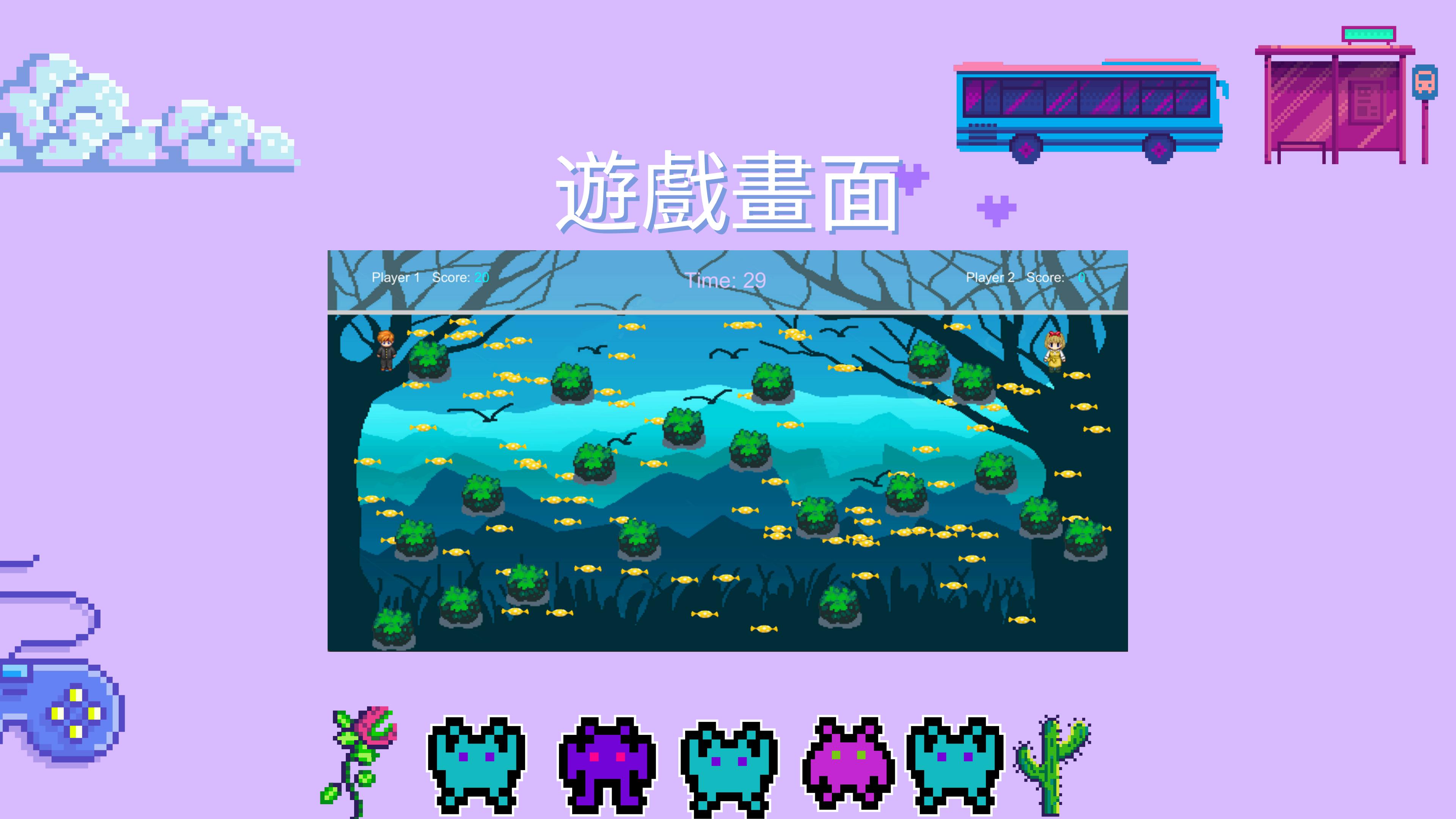


使用的元素

背景



遊戲畫面



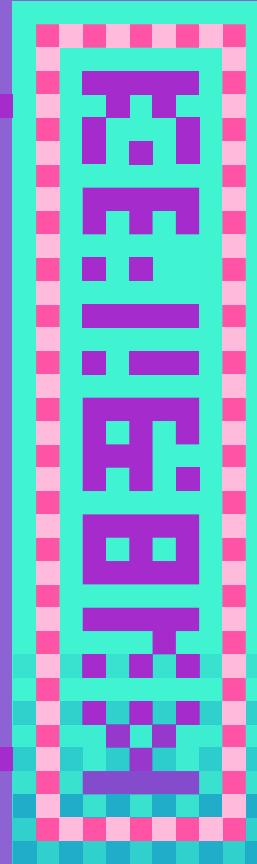
使用到的程式編寫技巧(課內)

初始化背景圖、人物

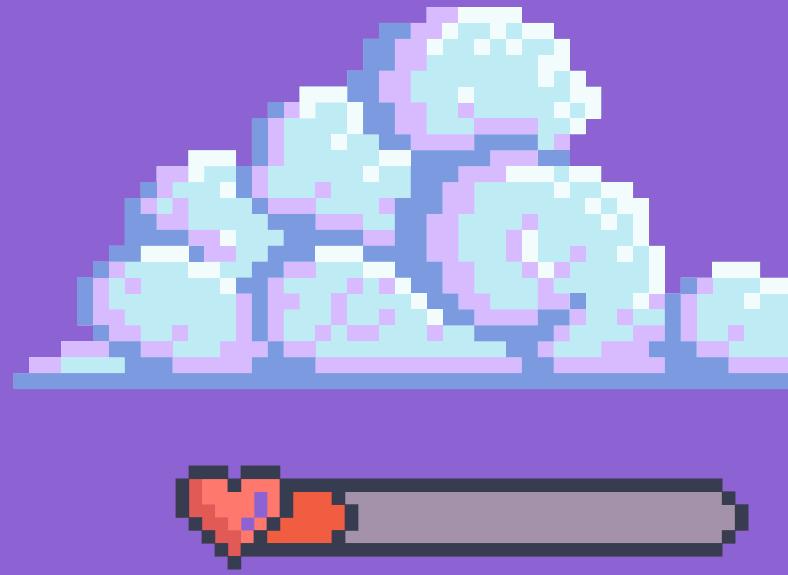
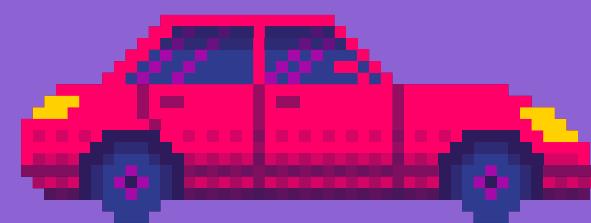
`loadImage()`

// 初始化影像

```
background = loadImage("forest.png");
bgCanvas = createGraphics(width, height);
bgCanvas.beginDraw();
bgCanvas.image(background, 0, 0, width, height);
bgCanvas.endDraw();
```



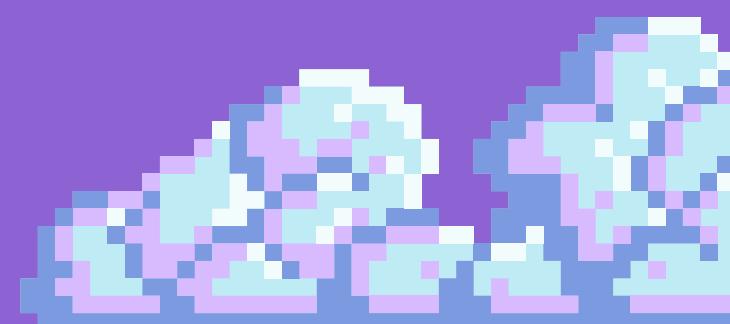
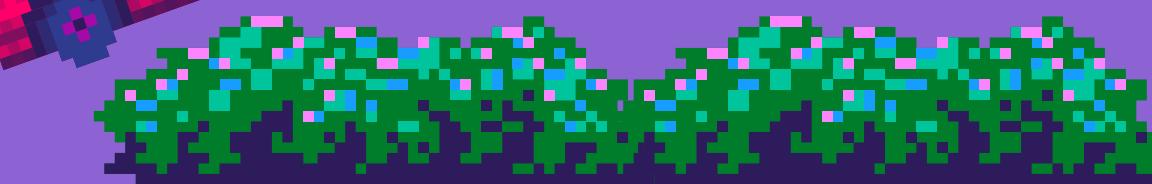
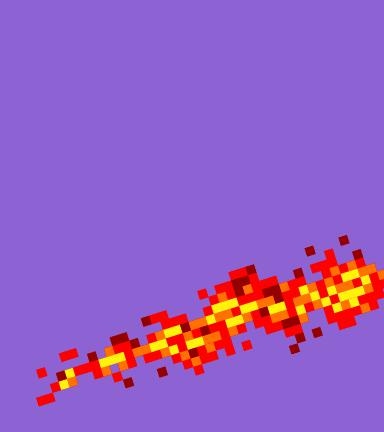
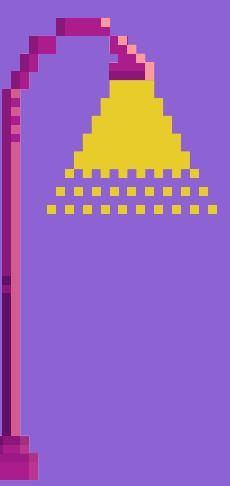
```
boy = loadImage("boy.png");
girl = loadImage("girl.png");
candy = loadImage("candy.png");
obstree = loadImage("smalltree.png");
```



使用到的程式編寫技巧(課內)

播放背景圖、人物動畫 draw()

```
void draw(){  
    image(bgCanvas, 0, 0);  
    line(0, 150, width, 150);  
  
    p1.give(boy);  
    p2.give(girl);  
}
```

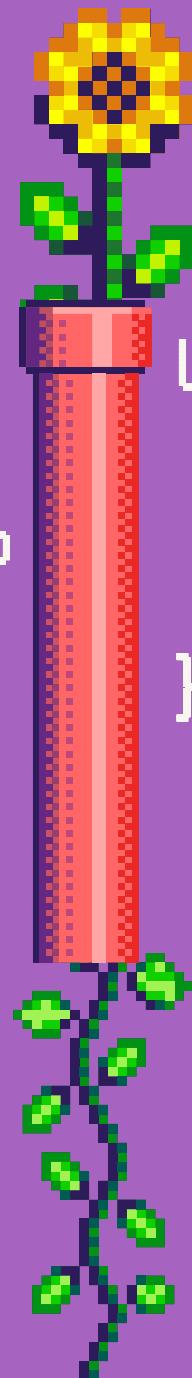


使用到的程式編寫技巧(課內)

鍵盤控制人物移動 keyPressed() & keyReleased()

```
int temp;  
void keyPressed() {  
    if(key == 's' || key == 'w' || key == 'a' || key == 'd') {  
        p1.setDirection(key);  
        temp = key;  
    }  
    else if(keyCode == LEFT || keyCode == RIGHT || keyCode == UP  
    || keyCode == DOWN) {  
        p2.setDirection(keyCode);  
        temp = keyCode;  
    }  
}
```

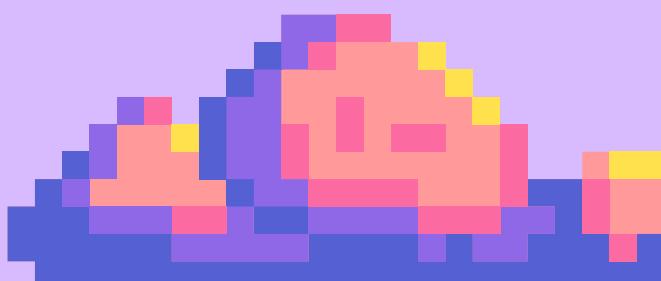
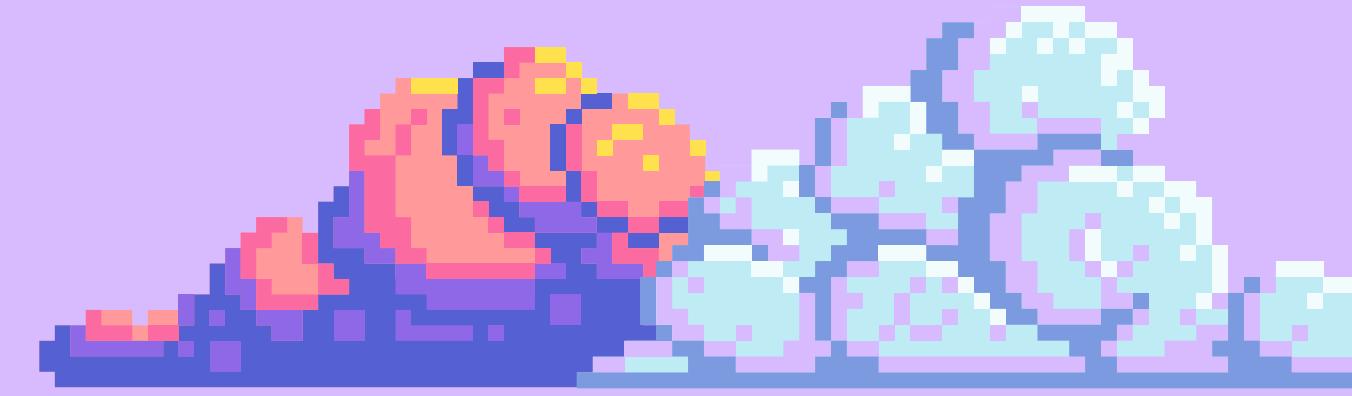
```
void keyReleased() {  
    if(key == 'a' || key == 'd' || key == 'w' || key == 's') {  
        p1.stop();  
    } else if(keyCode == LEFT || keyCode == RIGHT || keyCode ==  
    UP || keyCode == DOWN) {  
        p2.stop();  
    }  
}
```



使用到的程式編寫技巧(課內)

障礙物碰撞判定

```
boolean checkCollisionInDirection(int dir) {  
    int shrink = 20;  
    for (Obstacle r : obstacles) {  
        if (abs(r.x - x) > 200 || abs(r.y - y) > 200) continue;  
        // 若角色離障礙物過遠，跳過檢查該障礙物  
        if (dir == 1 && [y + h + speed > r.y + shrink && y < r.y + r.h - shrink] &&  
            [x + w > r.x + shrink && x < r.x + r.w - shrink]) return true;  
        if (dir == 4 && [y - speed < r.y + r.h - shrink && y > r.y + shrink] &&  
            [x + w > r.x + shrink && x < r.x + r.w - shrink]) return true;  
        if (dir == 2 && [x - speed < r.x + r.w - shrink && x > r.x + shrink] &&  
            [y + h > r.y + shrink && y < r.y + r.h - shrink]) return true;  
        if (dir == 3 && [x + w + speed > r.x + shrink && x < r.x + r.w - shrink] &&  
            [y + h > r.y + shrink && y < r.y + r.h - shrink]) return true;  
    }  
    return false;  
}
```



使用到的程式編寫技巧 〔課外〕



使用到的程式編寫技巧 (課外) 物件導向程式設計 OOP

```
Player p1, p2;  
ArrayList<Candy> candies = new ArrayList<>();  
Obstacle[] obstacles;  
class Obstacle {  
    int x, y, w, h;  
    Obstacle (int x, int y, int w, int h) {  
        this.x = x;  
        this.y = y;  
        this.w = w;  
        this.h = h;  
    }  
    void show() {  
        fill(128);  
        image(obstree, x, y, w, h);  
    }  
}
```

使用到的程式編寫技巧(課外)

吃糖果判定&糖果消失

```
Iterator<Candy> iter = candies.iterator();
while [iter.hasNext()] {
    Candy c = iter.next();
    c.show();
    if [p1.checkCollision(c)] {
        iter.remove();      // 吃到糖果直接移除
        player1_score += 10;
    } else if [p2.checkCollision(c)] {
        iter.remove();
        player2_score += 10;
    }
}
```

使用到的程式編寫技巧(課外)

倒數計時器

```
int elapsedTime = millis() - countdownStart;  
int remainingTime = countdownTime - elapsedTime / 1000;  
if (remainingTime > 0) {  
    fill(255, 0, 0);  
    textSize(100);  
    textAlign(CENTER, CENTER);  
  
    if (remainingTime == 6 || remainingTime == 5) {  
        // 顯示標題  
        fill(255, 215, 0); // 金黃色  
        text("糖果爭奪戰", width/2, height/2);  
    } else {  
        int elapsedTime = millis() - countdownStart;  
        int remainingTime = countdownTime - elapsedTime / 1000;
```

使用到的程式編寫技巧(課外)

倒數計時器

```
if (remainingTime > 0) {  
    fill(255, 0, 0);  
    textSize(100);  
    textAlign(CENTER, CENTER);  
  
    // 顯示倒數數字  
    text(remainingTime-1, width/2, height/2); // 顯示 3,2,1  
}  
return; // 倒數中，暫停遊戲邏輯  
}
```

使用到的程式編寫技巧 (課外)

遊戲結束視窗

```
if [gameRunning] {  
    textAlign(LEFT, TOP);  
    fill(227, 201, 255);  
    text("Time: " + remainingTime, width/2 - 100, 50);  
    if [elapsedTime >= time] {  
        gameRunning = false;  
        // 判斷勝負  
        String resultMessage;  
        int winner_score;  
        boolean player1_wins = player1_score >  
            player2_score;  
        boolean draw = player1_score == player2_score;  
        if [player1_wins] {  
            resultMessage = "玩家 1 獲勝🎉";  
            winner_score = player1_score;  
        } else if [draw] {  
            resultMessage = "平手! 😱😱😱";  
            winner_score = player1_score;  
        } else {  
            resultMessage = "玩家 2 獲勝🎉";  
            winner_score = player2_score;  
        }  
    }  
}
```

使用到的程式編寫技巧(課外)

遊戲結束視窗

// 顯示對話框

```
String message = "<html><div align='center'>" +  
    "<h2 style='font-size:25px;'>" + resultMessage + "</h2>" +  
    "<h2 style='font-size:20px;'>分數🎮</h2>" +  
    "<h2 style='font-size:20px;color:red;'>" + winner_score + "</h2>" +  
    "</div></html>";
```

```
JOptionPane.showMessageDialog(null, message, "🏆 Game Over 🏆",  
JOptionPane.PLAIN_MESSAGE);  
}
```

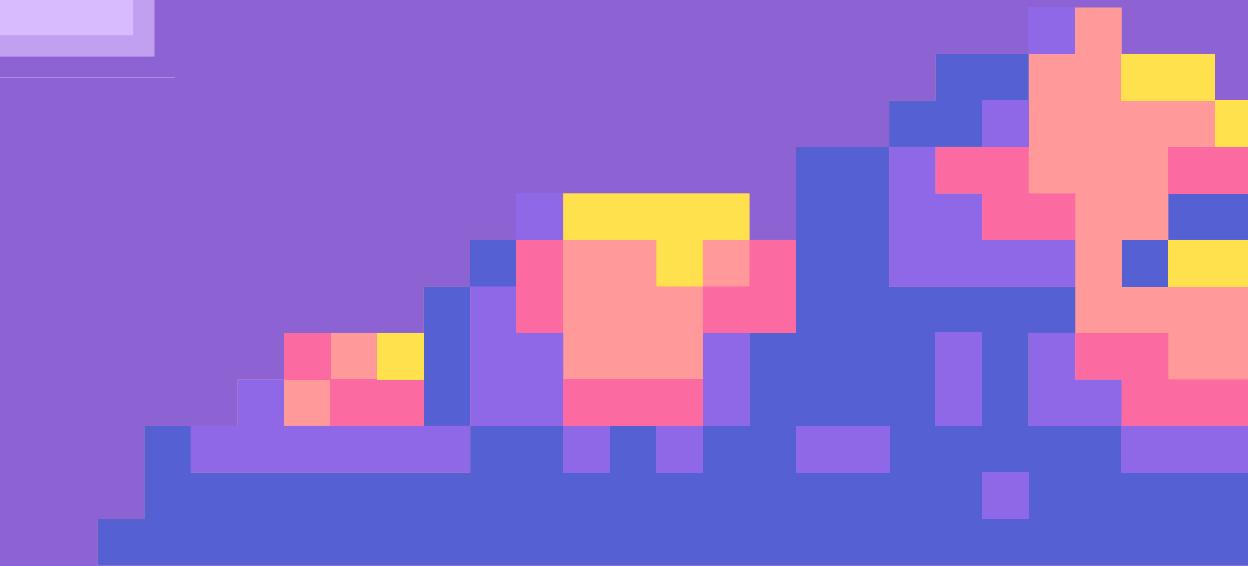
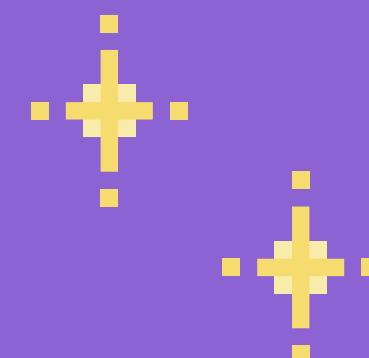
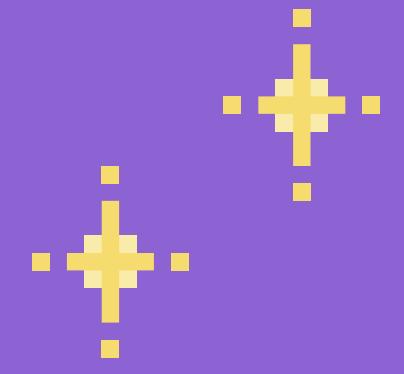
```
} else {  
    textAlign(LEFT, TOP);  
    fill(208);  
    text("Game Over !!!", width/2 - 150, 50);
```

使用到的程式編寫技巧 (課外) 遊戲結束視窗

```
// 設定文字大小  
textSize[32];  
  
// 顯示玩家 1 分數及結果  
if [player1_score > player2_score] {  
    fill[255, 255, 0]; // 黃色 [Winner]  
    text("Winner !", 390, 50);  
} else if [player1_score < player2_score] {  
    fill[255, 0, 0]; // 紅色 [Loser]  
    text("Loser !", 380, 50);  
} else {  
    fill[255]; // 白色 [平手]  
    text(" Draw !", 380, 50);  
}  
  
// 顯示玩家 2 分數及結果  
if [player2_score > player1_score] {  
    fill[255, 255, 0]; // 黃色 [Winner]  
    text("Winner !", width - 500, 50);  
} else if [player2_score < player1_score] {  
    fill[255, 0, 0]; // 紅色 [Loser]  
    text("Loser !", width - 490, 50);  
} else {  
    fill[255]; // 白色 [平手]  
    text("Draw !", width - 490, 50);  
}  
//fill[0]; // 黑色
```

DISPLAY OUR GAME

DEMO



THE END

THANKS FOR YOUR LISTENING

