

w18: Microcontroller Experiments

📅 Dates	@January 12, 2023
☰ Topic	Final project

Problem Description

貪食蛇遊戲：

1. 產生長度大於2的蛇
2. 周圍的牆不可穿越
3. 隨機產生食物，被吃掉後會再次產生
4. 蛇吃掉食物後會計分

Code and Explanations

Final project大部分的程式碼是改寫Lab6的程式碼，因此以下只說明特別新增及更改的部分。

```
void draw()
{
    int i,j;
    Set_DisplayStartLine (0);
    Set_Yaddr (0);
    for(j=0;j<8;j++){
        Set_Xaddr (j);
        for (i=0;i<64;i++)
            Send_Data (0x00);
    }
}

void initial(){
    Set_DisplayStartLine (0);
    Set_DisplayOn (0);
    draw();
    Set_DisplayOn (1);
    draw();
}
```

在面板一通電或遊戲結束時會利用initial()清空GLCD上的圖案，而draw()是負責設置GLCD數值。

```

void
drawByteLeft(int x,int y,int bits){
    int i;
    Set_DisplayStartLine (0);
    Set_DisplayOn (1);
    Set_Xaddr(x);
    Set_Yaddr(y*8);
    for(i = y*8;i < y*8+8; i++){
        Send_Data (bits);
    }
}
void
drawByteRight(int x,int y,int bits){
    int i;
    Set_DisplayStartLine (0);
    Set_DisplayOn (0);
    Set_Xaddr(x);
    Set_Yaddr((y-8)*8);
    for(i = (y-8)*8;i < (y-8)*8+8; i++){
        Send_Data (bits);
    }
}
}

```

drawByteRight()和drawByteLeft()具有相似的功能，就是利用輸入的x、y數值和要設置的圖樣，分別以一單位為8*8的圖案畫在面板左(drawByteLeft)和右側(drawByteRight)。由於一單位是8*8所以GLCD總共會被劃分成8*16，因此 $0 \leq x \leq 7$, $0 \leq y \leq 15$ 。

```

void food(){
    fx = rand()%8+0;
    fy = rand()%16+0;
    if(fy<8) { // draw left
        drawByteLeft(fx,fy,0x3c);
    }
    if(fy>=8 && fy<16) { // draw right
        drawByteRight(fx,fy,0x3c);
    }
}
}

```

food()負責隨機產生食物，由於要和蛇產生區別，每一行的圖形我採用0x3c，相較於0xff小但是寬度一樣是8bits。

```

int snackLen = 2;
//default head & bottom
int snackX[3] = {7,7};
int snackY[3] = {15,14};
int head = 1,bottom=0;
int fx,fy;
int score = 0;
void snack(int snackDest){
    int x,y; //0 <= x <= 7 ; 0 <= y <= 15
    int i = 0;

```

```

if(snackX[button] > 0 && snackY[button] > 0){
    x = snackX[button];
    y = snackY[button];
    if(y<8) { // draw left
        drawByteLeft(x,y,0x00);
    }
    if(y>=8 && y<16) { // draw right
        drawByteRight(x,y,0x00);
    }
}
button = head;
if(snackDest == 1){ // up
    head = (head + 1)%2;
    snackX[head] = snackX[button] - 1;
    snackY[head] = snackY[button];
}
else if(snackDest == 2){ // down
    head = (head + 1)%2;
    snackX[head] = snackX[button] + 1;
    snackY[head] = snackY[button];
}
else if(snackDest == 3){ // left
    head = (head + 1)%2;
    snackX[head] = snackX[button];
    snackY[head] = snackY[button] - 1;
}
else if(snackDest == 4){ // right
    head = (head + 1)%2;
    snackX[head] = snackX[button];
    snackY[head] = snackY[button] + 1;
}
if(snackX[head] >= 0 && snackX[head] < 8 && snackY[head] >= 0 && snackY[head] < 16){
    x = snackX[head];
    y = snackY[head];
    if (x==fx && y==fy){
        score += 1;
        food();
    }
    if(y<8) { // draw left
        drawByteLeft(x,y,0xff);
    }
    if(y>=8 && y<16) { // draw right
        drawByteRight(x,y,0xff);
    }
}
else if(snackX[head] < 0 || snackX[head] >= 8 || snackY[head] < 0 || snackY[head] >= 16){
    initial();
    food();
    snackX[0] = 7;
    snackX[1] = 7;
    snackY[0] = 15;
    snackY[1] = 14;
}
}
}

```

snack()負責畫出蛇的樣貌和判斷是否撞牆。

在snackX和snackY中，[0]和[1]會輪流儲存頭的位置，並且在頭尾交換前，會先在GLCD上擦除當前尾巴位置。

snackDest是用來判斷當前的按鈕希望蛇頭往哪的方向行進，在if-else判斷式中除了更新蛇頭位置外還判斷是否撞牆了，如果撞牆較呼叫initial()清除當前畫面、重新產生食物和重設蛇的位置，代表遊戲重新開始。

```
void
main (){
    int key;
    system_init_config ();
    Shutup_WatchDog ();
    GLCD_Reset ();
    // initial canvas
    initial();
    snack(0);
    food();
    while (1){
        P4 = score;
        key = button_detect ();
        if(key == 1){ //up
            snack(1);
        }
        else if(key == 2){ //down
            snack(2);
        }
        else if(key == 3){ //left
            snack(3);
        }
        else if(key == 4){ //right
            snack(4);
        }
    }
}
} //end of function main
```

主程式，主要負責設置硬體和偵測按鈕。

Discussions

這個project因為具有很高的互動性所以製作成功後滿足感大於之前的lab很多很多，可惜最後我只剩下兩周的時間可以做，主要有大進展只花了一周的時間，但是就是因為如此我在完成這個project的時候充滿感激的心，因為我沒有遇到很多人都遇到的空間不足的問題，蛇動起來也很正常看起來沒有故障，最主要的是板子不管是LED或GLCD都非常配合，在硬體上面我真的花很少時間排除障礙，除此之外同學們都很願意幫忙解除疑惑，這個期末雖然同樣緊張，卻很滿足、開心。