

使用 arduino 实现 FlappyBird 小游戏

黄宏达 17307110019

1. 课题设计

使用 16×16 点阵，LCD1602A 和按钮实现 FlappyBird 游戏。

需求：

- (1)使用单个像素点表示鸟，使用宽度为 2 个像素点，高为 16 个像素点，中间含有 3 个像素点空缺的列表示水管。
- (2)鸟自动下降，速度随通过水管数增加以提高难度。
- (3)使用按钮控制鸟的上升，每次按动按钮可以使鸟上升一个像素点，需要连续按动以保持鸟的高度不变
- (4)鸟每过一根水管，则总得分加 1，使用 LCD1602A 实时显示分数
- (5)设计开场和结束标志，开场标志为一个小鸟图案，结束标志为 GG 图案。

实验仪器：

arduino uno 开发板， 16×16 点阵，LCD1602A，按钮一个，面包板，杜邦线若干。

实验原理：

(1) 16×16 点阵

对于点阵，当像素点对应行为 1，列为 0 时则此点为亮，否则为暗。

使用 2 个 74HC138 控制行数，原理为将四位二进制数转化为行数，例如 1001 表示第 9 行，此时第 9 行对应所有像素点的行为 1。

使用 2 个 74HC395 控制列，原理为缓存 16 位二进制数进入点阵，每一位对应一列。与 74HC138 配合控制某行的亮与暗。例如向 74HC138 传入参数 1001，向 74HC395 传入参数 1100110010000000，则第 9 行 1, 2, 5, 6, 9 列对应像素点为亮，其他为暗。

每次显示一行的像素点，通过视觉暂留实现多行的同时显示。

(2) LCD1602A

使用 I2C 协议进行控制。本项目使用 LiquidCrystal_I2C 库对 LCD1602A 进行控制，实时显示总分数。

2. 实验过程

首先，由于 16×16 点阵为商家所研制，没有对应标准库，所以根据商家所提供源码和 16×16 点阵的原理构造基本函数，使得能够根据一个 16×16 的二维数组来显示函数，便于之后的编程和实现。

接着，设计开始画面和结束画面，实现方法为根据需要显示的图案创建二维数组，数组中1对应的点为亮，其他点为暗。

然后，设计 FlappyBird 主程序：使用中断函数控制鸟的上升，使用随机函数和数组来添加水管，实现水管的移动，统计分数，计算难度，控制水管前进速度。

最后，使用 LCD1602A 实时显示分数。

3. 实验结果

- (1) 开始界面，运行界面，结束界面如图 1, 2, 3 所示，开始图案为一只像素小鸟，结束图案为 GG 图标。

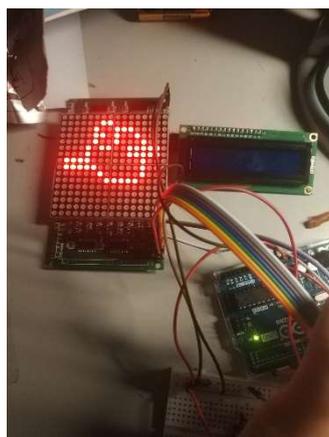


图 1 开始画面

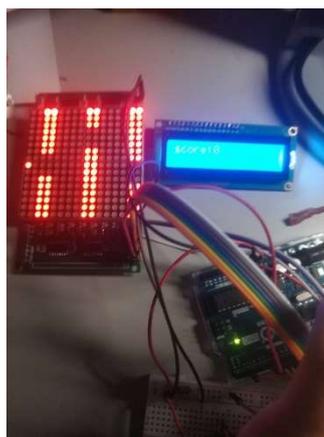


图 2 运行画面

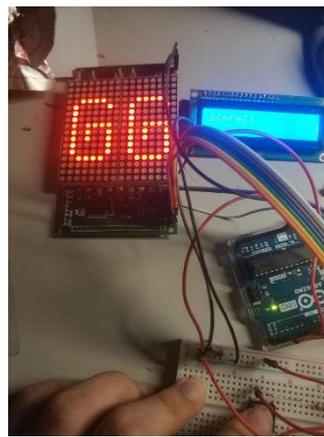


图 3 结束画面

- (2) 可以实现 FlappyBird 游戏的基本操作。鸟可以通过按钮控制上升和下降。水管自动向左移动，当一个水管到达最左端时，会有一个新的水管进入，新水管的缺口位置由随机函数生成，避免重复。每通过一个水管，则分数加 1。根据通过水管数增加难度，每 12 个水管难度加一级，表现为水管左移速度的加快。
- (3) LCD1602A 可在第一行实时显示分数。

4. 简要分析

(1) 拖影和重影的防止：对于 16×16 点阵，由于采用 74HC395 进行控制，重影问题较为方便解决，即每次更新数据时都关闭数据的显示，等到数据更新完毕后，再打开使能端，显示图案，避免图案的连续更新，由此便可以避免

```
392 void Display(const unsigned char cols[16][2])
393 {
394     for (unsigned char i = 0; i < 16; i++)
395     {
396         digitalWrite(LEDARRAY_G, HIGH); //更新数据时关闭显示，等更新完数据，再显示，防止重影
397         Send(cols[i][1]);
398         Send(cols[i][0]);
399
400         digitalWrite(LEDARRAY_LAT, HIGH);
401         delayMicroseconds(1);
402
403         digitalWrite(LEDARRAY_LAT, LOW);
404         delayMicroseconds(1);
405
406         Scan_row(i);
407
408         digitalWrite(LEDARRAY_G, LOW);
409         delayMicroseconds(300); //延时一段时间，让LED亮起来
410     }
411 }
```

图 4 点阵显示代码

重影。代码如图 4 所示。而对于 LCD1602A，由于硬件限制，没有办法完全消除拖影，由于本项目不需要过高的刷新率，因而仅当成绩发生变动时更新屏幕，可缓解拖影现象。对于使用体验没有明显影响。

(2) 水管的生成和运动：水管的运动通过改变首个水管的坐标来实现，而速度则通过循环次数来控制，速度越快，循环次数越少。水管的生成采用添加一个冗余水管数组来实现，即除了显示在点阵中的

三组水管外，还额外生成一组水管，四组水管同时左移，当一个水管到左端消失后，冗余的水管便从右端进入点阵，并且结合随机函数生成一个新的冗余水管至于点阵外。通过这一方案，可以保持点阵中始终显示 3 组水管，且水管的缺口分布随机这一需求。这一过程的示意图如图 5 所示。count 表示首位坐标。

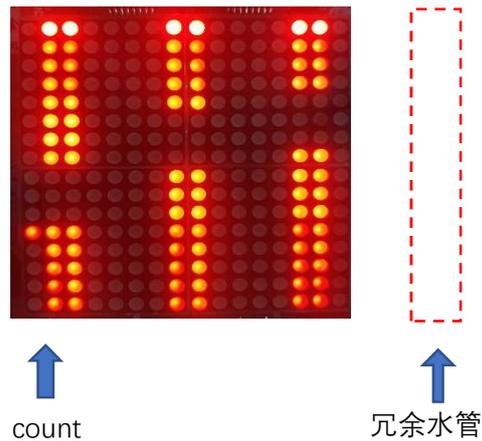


图 5 水管生成和运动过程的示意图

(3) 为了能够随时响应按钮的触发以及避免按住按钮导致鸟的连续上升，因而采用中断函数进行控制，触发条件为 RISING，即按钮端电压上升则触发一次鸟的上升，其他情况则不触发，由此实现鸟的精确控制。

5. 实验结论

本实验使用 arduino 平台实现了 FlappyBird 小游戏。通过 16×16 点阵，按钮和 LCD1602A，实现了 FlappyBird 的所有基本操作，包括鸟的上升与下降，水管的平移和加速，以及计分。通过对点阵源码的扩展，使得能够根据二维数组更

为方便的编写程序控制点阵，为以后的其他项目的设计和改进了基础。

代码上传于 <https://github.com/hhd88888/FlappyBird-by-arduino-uno>。