

boxGame_Arduino

简单方块游戏

这是我的课程期末作业，一款基于 Arduino 的简单游戏

一、游戏前准备

当玩家插上电源，设备首先会测量环境光照强度，玩家需要依据画面指示配合完成设备的光照适应调整。之后即可开始游戏。

二、游戏过程

游戏画面会显示一个方块从远处而来，当方块移动到玩家面前时（也就是当 LED 点阵上一半的灯泡都亮起的时候），玩家需要做出反应来跨越（或躲避）方块。

三、游戏规则

- 1、游戏设有 10 个关卡，每当玩家跨越一个方块，分数加 1 分，10 个关卡完成后总分为 260 分。
- 2、玩家可以选择**站立**、**跳起**或**蹲下**三种姿势，正常情况下，玩家状态为站立，如果玩家需要跳起：**玩家需要双脚离开设备上的相应的两个光敏电阻**；如果玩家需要蹲下：**玩家需要同时遮盖住三个光敏电阻，这意味着玩家必须蹲下并用手遮住第三个光敏电阻。**

3、**当有方块从上方移动而来，到达玩家跟前时，玩家需要跳起；当有方块从下方移动而来，到达玩家跟前时，玩家需要蹲下。**

4、当玩家有一只脚（当然也包括两只）踩在设备上，游戏开始；之后，只要玩家有一只脚（当然也包括两只）踩在设备上，游戏就会继续；否则，游戏结束。

四、注意事项

1、玩家应尽可能**把握准行动时机**，过早或过晚地跳起或蹲下都不会作数。

2、**玩家应确保行动后双脚仍然有一只及以上遮盖在光敏电阻上**，而不是恰好踩在光敏电阻旁边（特别是跳起落地后），这会导致设备以为玩家离开游戏，游戏结束。

3、**每一次开机都需要重新测光调整设备，较远距离的移动设备也应重启设备重新测光。**

4、游戏设有“最高分”，当玩家最终分数超过当前最高分时，就会显示破纪录动画；不过遗憾的是，这个“最高分”并不能永久地保存，当设备关闭后，所有数据都会丢失，再次游戏时，上一次的“最高分”清零。

程序相关

请参见文件：**boxgame.ino**

游戏程序包括 2.0 和 3.0 两个版本。简单来说，2.0 版本是一个半成品，不支持移位寄存器，并且没有全部的动画效果，没有音效，没有难度变化和关卡等等。

3.0 版本则是完整的游戏，鉴于硬件和本人编程技术的限制，虽然它仍然有诸多不足、游戏的体验仍然不尽人意，不过它比 2.0 更好。3.0 经过了多次的优化和调整，增加了更多的功能和交互，每一个功能都被写成了函数分成一个个模块。

playsound() 声音播放

```
void palyound(int t, int mode = 1, int del = 1000);
```

t: 表示蜂鸣器发出的频率。

mode: 可以输入 0, 1, 2, 表示 3 中声音播放模式,默认值为 1:

0: 设置 IO 口为 “输出”，并持续播放一个音调 (t);

1: 设置 IO 口为 “输出”，并播放一个音调 (t) 持续 del 毫秒，设置 IO 口为 “输入”;

2: 设置 IO 口为 “输出”，并播放从音调 (t) 开始持续上升的声音，设置 IO 口为 “输入”。

del: 表示声音播放时间，默认值为 1。

display() LED 控制 (显示画面)

```
void display(byte dat[8], int del);
```

dat[8]: 表示要显示的画面，是一个包含 8 个二进制数元素的一维数组;

del: 表示画面显示时长。

display_boxDown() LED 控制（上下颠倒显示画面）

```
void display_boxDown(byte dat[8], int del);
```

dat[8]: 表示将要显示画面的上下颠倒的画面，是一个包含 8 个二进制数元素的一维数组；

del: 表示画面显示时长。

RandomBoxComing()

```
int RandomBoxComing();
```

无输入，返回一个 int 值。

这个函数将随机生成箱子并将其显示在 LED 点阵上，并返回当前箱子的位置（即上方/下方）。

这个函数中调用了其他函数：playsound()、display()、display_boxDown()。

whatShouldPlayerDo()

```
void whatShouldPlayerDo(int left, int right);
```

left: 表示当前左脚位置光敏电阻的读数；

right: 表示当前右脚位置光敏电阻的读数。

这个函数将根据当前箱子状态，判断玩家是否做出相应的动作并决定游戏是否继续。

这个函数中调用了其他函数：playsound()、game_end()

这个函数使用了全局变量：up_or_down、score。

game_start()

```
void game_start();
```

无输入，无返回值。

这个函数将显示游戏开始动画，并标明游戏状态为“开始”而不是“无人游戏”。

这个函数中调用了其他函数：display()。

这个函数使用了全局变量：flag。

game_end()

```
void game_end();
```

无输入，无返回值。

这个函数在游戏结束时被调用，它将判断玩家是否胜利并播放相应的动画，然后调用“最高分”函数来判断是否播放最高分动画，最后显示玩家分数并重置游戏数据（不会重置“最高分”）。

这个函数中调用了其他函数：display()、bestScore()。

这个函数使用了全局变量：flag、score。

level_pass()

```
void level_pass();
```

无输入，无返回值。

这个函数将显示通关动画。

这个函数中调用了其他函数：display()。

the_game_difficulty()

```
void the_game_difficulty(int level);
```

level: 表示当前关卡 (事实上是当前分数: score)。

这个函数设置了游戏的难度变化和关卡。

这个函数中调用了其他函数: level_pass()。

这个函数使用了全局变量: speed。

light_adapt()

```
void light_adapt();
```

无输入, 无返回值。

这个函数用来测量环境光照强度, 并对一些条件数值进行合适的初始化。

这个函数中调用了其他函数: playsound()、display()

这个函数中使用了全局变量: left_light(默认值 900)、right_light(默认值 710)、hand_light(默认值 850)。

bestScore()

```
void bestScore();
```

无输入, 无返回值。

这个函数用来判断最终分数是否为“最高分”, 是则记录新纪录并播放动画, 否则不执行多余动作。

这个函数中调用了其他函数: display()。