

# 基于 python 语言逻辑结构的单片机小游戏设计与流行文化的结合尝试

工学院 范文琳

**摘要：**micro:bit 是英国广播电视公司为青少年编程教育推出的一款微型单片机，具有操作简单易行、互动性强等特点。笔者从 micro:bit 的固有特点出发，考虑到流行文化在青少年群体当中的影响力，以流行文化元素为背景语境，基于 python 基本的语言逻辑结构设计出一款互动反应类小游戏，作为一次编程教育与流行文化结合的尝试，意图探索流行文化对青少年编程教育的促进作用。

## 一、 选题及创意介绍

笔者在对 micro:bit 基本操作了解的基础上，希望能够基于 micro:bit25 各 LED 显示等与声音装置开发出一款兼具互动性和观赏性的动画舞蹈小游戏。在确定该选题方向后，笔者立刻想到近年来网络上与舞蹈密切相关的热梗“坤坤唱跳 rap 篮球”，并认为这是一个对青少年群体有一定吸引力且具有可操作性的动画呈现方向。但由于硬件设备的局限性（显示分辨率、内存等）等原因，无法呈现该舞蹈动画的完整版本，只能从原视频中提取并抽离出概念形象，以抽象化的方式在作品中呈现，达到让人“会心一笑”的目的。

除舞蹈动画外，本作品还加入了闯关小游戏机制，通过完成不同难度的挑战来解锁最后的舞蹈彩蛋。同时，小游戏操作中也融入了音乐与舞蹈元素，一方面凸显主题，另一方面让挑战者在游戏过程中得到一些乐趣。

## 二、 方案设计和硬件连接

本作品分为两个部分，既游戏挑战环节与彩蛋放送环节，完成难度依次递增的 5 个游戏关卡即可解锁终极彩蛋。

**游戏设计方案如下：**

1. 初始设置：屏幕中央显示一个小人（坤坤），玩家可以通过按下 A、B 按

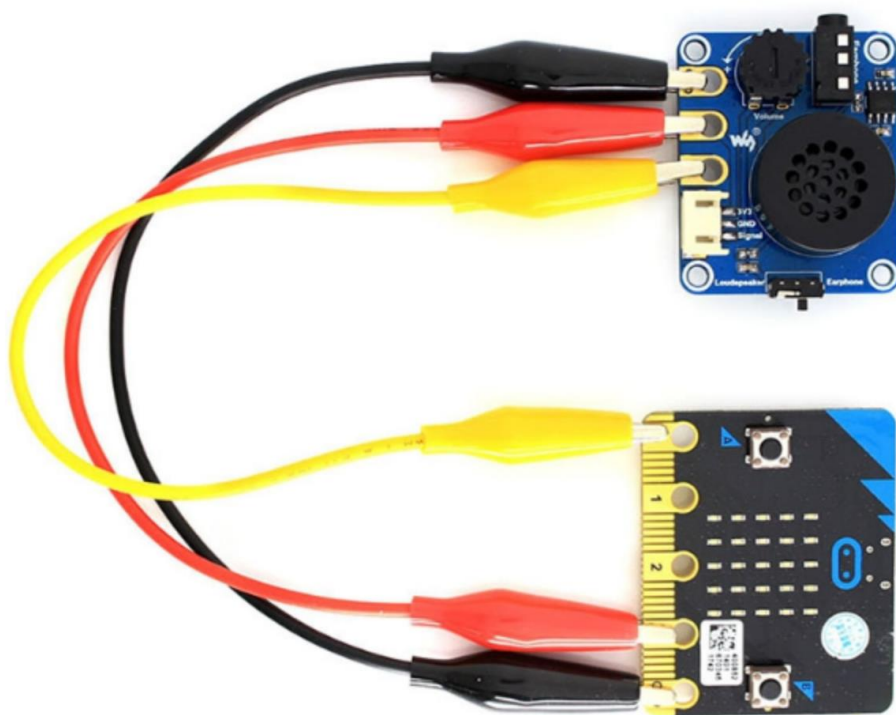
钮或触摸 pin2 引脚分别达到让小人挥舞左手、挥舞右手、跳跃三个舞步动作；

2. 游戏开始后，从屏幕四角随机刷新飞来的篮球，玩家需要通过操作小人做出相应舞步来接住篮球，汲取篮球之力。每轮篮球数量与飞行速度和当前关卡难度呈正相关。（注：由于游戏目标为“接住”篮球，故默认从上方飞下来的篮球只能通过跳跃动作接取，而下方飞来的篮球只能通过伸出对应侧手臂接取。）
3. 完成全部 5 级难度的关卡后即为游戏通关，播放彩蛋动画。

### 硬件连接方案：

由于笔者手中没有 micro:bit 实机，故以下内容均基于对 micro:bit 硬件功能的理论层面理解。

本设计需要硬件实现的部分主要为动画显示和音乐两方面。动画显示可以由 micro:bit 自带显示屏完成，而音乐设备需要外接蜂鸣器。通过鳄鱼夹将 micro:bit 引脚 0、GND 与蜂鸣器相连，连接方法如图所示。



（图片来源网络，请以实物为准）

### 三、 实现方案及代码分析

#### 实现方案:

基于 python 语言中的基本逻辑结构——循环与分支结构，笔者构建出了本作品的基本逻辑架构，即通过判定按钮/引脚是否被按下改变屏幕显示内容，通过摇晃主板进入游戏模式。在游戏过程中，通过迭代数 i 的增长情况决定篮球每一时刻的位置，并联合按钮情况检定是否接球成功。

#### 代码分析:

本作品代码分为四部分：初始设置部分、自由舞蹈部分、篮球挑战部分和彩蛋部分，下面我们对三部分分别进行分析。

##### 1. 初始设置部分

```
KUN=Image("09990:""00900:""09990:""00900:""09090")
KUNb=Image("09990:""00999:""09900:""00900:""09090")
KUNa=Image("09990:""99900:""00990:""00900:""09090")
KUNleft=Image("09990:""00900:""00090:""00900:""09090")
KUNright=Image("09990:""00090:""00009:""00090:""09090")
KUNjump=Image("09990:""09990:""00900:""99999:""00000")
tune=["A3:1","A3:1","E4:2","E4:2","A4:4","R:6","A3:1",
      "A3:1","A3:1","A3:1","E4:2","E4:2","A4:4","R:6"]
flag=0
display.show(KUN)
```

该部分主要通过编写 LED 显示灯明暗情况完成作品中小人不同状态下的动作呈现以及音乐设计。值得一提的是，该音乐简谱取材于著名偶像练习生蔡徐坤先生的成名作《只因你太美》，并基于 micro:bit 电子发声特点进行了一定程度的改编，力图在单片机的发声范围内同时保留原作品辨识度与乐曲听觉美感。

## 2. 自由舞蹈部分

```
while True:
    if button_a.is_pressed():
        display.show(KUNa)
        music.play("C4:3")
        sleep(300)
        display.show(KUN)
    if button_b.is_pressed():
        display.show(KUNb)
        music.play("D4:3")
        sleep(300)
        display.show(KUN)
    if pin2.is_touched():
        display.show(KUNjump)
        music.play("A4:3")
        sleep(300)
        display.show(KUN)
```

在本部分中，玩家可以操纵小人完成三种舞步动作，包括左抬臂，右抬臂和跳跃，同时根据做出动作播放不同乐音。

## 3. 篮球挑战部分

```
if accelerometer.was_gesture('shake'):
    display.scroll("GO")
    sleep(100)
    display.show(KUN)
    music.play(music.POWER_UP)
    sleep(100)
    while True:
        for j in range(5):
            if j==5 and flag==1:
                break
            level=j+1
            display.show(str(level))
            sleep(1000)
            display.show(KUN)
            for k in range(level+1):
                r=random.randint(1,4)
```

如图为篮球挑战部分的主干程序，当玩家摇晃主板后进入挑战模式。用 for 循环实现 5 级难度的挑战，并在每一层循环中嵌套 for 循环实现产生与挑战难度等级相对应的篮球数量。引用 random 库来产生决定篮球初始位置的随机种子。

```
if r==1:
    display.set_pixel(0,0,4)
    for i in range(4):
        sleep(1000-level*100)
        display.set_pixel(0,i,0)
        display.set_pixel(0,i+1,4)
        if i==2 and pin2.is_touched():
            display.show(KUNjump)
            music.play("A4:3")
            sleep(300)
            display.show(KUN)
            flag=1
            break
        if i==3:
            flag=2
            break
    if flag==2:
        break
```

根据随机产生的 r 大小生发出四个分支，由于各分支逻辑结构大体类似，固此处单以 r=1 的情况为例进行分析。

r=1 时，在左上角（若 r 为 2、3、4，则为左下、右上、右下）产生一个亮度为 4（即略暗于小人图案亮度以示区分）的像素点，通过循环、sleep 并设置下一个位置的像素点亮度来实现篮球随时间推移缓缓下降，下降速度与关卡难度线性相关。当篮球达到指定位置时判定为“可接球范围”，此时做出相应的动作（r=1 时为跳跃），即为接球成功，变量 flag 的值保持为 1，跳出当前循环进入下一轮。

```
if flag==2:
    display.show(Image.SAD)
    music.play(music.WAWAWAWAA)
    sleep(1000)
    break
```

如果接球失败，将变量 flag 的值设为 2，跳出当前二重循环，进入失败动画，随后自动重新开始游戏。

#### 4. 彩蛋部分

```
if flag==1:
    for i in range(1):
        display.show(KUNleft)
        music.play(tune[0])
        display.show(KUNright)
        music.play(tune[1])
        display.show(KUNleft)
        music.play(tune[2])
        display.show(KUNright)
        music.play(tune[3])
        display.show(KUNleft)
        music.play(tune[4])
    display.show(Image.HEART)
    music.play(tune[5:])
    break
```

由于 micr:bit 一次只能执行一条命令，和人类对其一心二用的期望完全背离，因此笔者灵机一动，决定利用计算机单行程序运行快的特点，用极小的时间差给人脑造成动画与音乐同时播放相互配合的印象。加上歌曲《只因你太美》节奏较快，能够很好地符合程序需求，因此这里采用了图像命令与音乐命令交替执行的方式。

## 四、 后续工作展望

本次工作中，笔者（亦即开发者）体验到了极大的发现问题与解决问题的乐趣，同时，在编程作品中融入流行文化的“整活儿”动机也在笔者陷入找 bug 的痛苦深渊时为笔者带来了不竭动力。

纵观本次作品，笔者在此提出几条本设计仍存在的缺陷以及未来可以继续开展工作的方向：

1. 可实现功能相对单一，且在挑战模式中非判定时刻按动按钮/引脚，小人不会产生动作，与一般游戏逻辑不符；
2. 接球判定以当前按钮/引脚状态为判据，不具有瞬时性，若在小球刷新后立刻按住相应按钮，可在一定程度上实现“作弊”；该问题后续可以通过对按钮指令 `is_pressed()`、`was_pressed()` 等进行修改得到解决；
3. 实现音乐动画的方法受局限于快节奏、短篇幅的乐句，无法实现“一拍两动”；

针对以上问题，笔者认为，后续可以对其他单片机小游戏程序进行学习，寻找在“一心二用”方面可供优化的算法或实现方式，实现更加“丝滑”的 `micro:bit` 动画。

此外，笔者在编程过程中感受到利用 `micro:bit` 编写游戏的互动性很大程度上受到单线程运行逻辑的局限。笔者希望能够在后续对单片机进行更加深入的了解和学习，并试图探索其实现多线程工作的方法或者模拟多线程工作的可能性。

## 五、 小组合作分工

本小组为单人小组，一切工作均由笔者（范文琳）承担。