指尖律动

一、选题及创意介绍

本次项目为“音乐跟随游戏”，通过micro:bit板块内置的声音传感器和LED灯带，实现一个有趣的音乐游戏。玩家需要在规定时间内将手指放到板子上，板子会检测声音的强度，然后让LED灯带根据声音的强度闪烁，让玩家尽可能的跟随音乐的节奏来移开手指，若未能按时移开手指则减分，分数达到一定值即可胜利。

二、设计方案和硬件连接

方案：

1.采用micro:bit板块内置的声音传感器实现声音信号的采集；

2.使用python对采集到的音频数据进行处理；

3.使用LED灯带实现多彩闪烁效果。

硬件连接：

micro:bit模块+LED灯带

1.将LED灯带的GND接地线连接至microbit GND口。

2.将LED灯带的VCC接电源使其亮起。

3.将LED灯带的DIN接至microbit pin1口。

三、实现方案及代码分析

工作原理：1.启动程序后，等待用户放置手指；

2.检测到声音强度后，根据强度控制LED灯带闪烁；

3.在规定时间内移开手指即可得分。

代码实现：

from microbit import \*

import neopixel

# 初始化LED灯

np = neopixel.NeoPixel(pin0, 8)

# 常量定义

TIME\_LIMIT = 30 # 游戏时间限制（秒）

MAX\_SCORE = 10 # 最大分数

BLINK\_DELAY = 100 # 每次闪烁延时（毫秒）

def get\_sound\_level():

"""获取声音信号强度"""

sound\_level = 0

for i in range(100):

sound\_level += microphone.sound\_level()

return sound\_level / 100

def blink\_led(score):

"""控制LED灯带闪烁"""

if score == 0:

np.clear()

else:

for i in range(score):

np[i] = (255, 0, 0)

np.show()

sleep(BLINK\_DELAY)

np.clear()

sleep(BLINK\_DELAY)

# 启动游戏

display.show('Ready')

while not button\_a.is\_pressed():

pass

score = 0

start\_time = running\_time()

while True:

sound\_level = get\_sound\_level()

if sound\_level > 50:

score += 1

if score >= MAX\_SCORE:

display.scroll('You win!')

break

else:

if score > 0:

score -= 1

cur\_time = running\_time()

elapsed\_time = (cur\_time - start\_time) // 1000

if elapsed\_time >= TIME\_LIMIT:

display.scroll('Time up! You lose!')

break

blink\_led(score)

代码说明：

1.通过 microphone.sound\_level() 获取声音信号强度；

2.使用neopixel模块控制LED灯带，根据分数控制闪烁的LED灯数量和颜色；

3.使用 running\_time() 函数计算游戏时间，规定时间内得分越多则胜利，否则失败。

四、后续工作展望

在这个项目中，我们可以进一步优化处理声音数据的算法，增加游戏的难度。也可以将游戏扩展到多人互动模式，让玩家之间进行PK，增加游戏的趣味性。

五、小组分工合作

由个人完成所以没啥合作。本次项目分为硬件和软件两个方面的深入研究设计，所以可以这样分工：一个人负责硬件连接和调试，另外一个人则负责程序编写和算法优化等。