数据结构与算法【C1】开源硬件创意作品技术报告

创意作品名称：斌躺梦幻季
组长：程歆乐 组员：张开

摘要：

本作品是音游在 micro:bit 上的实现，玩家通过观察音符的移动做出相应的反应。游戏拥有两个模式：一、无尽模式（zen mode）：音符的类型和生成位置随机，玩家保持专注，一旦玩家没有及时做出正确的反应，游戏即告结束，否则音符不断生成；二、挑战模式（challenge mode）：音符按照预先设定的方式运行，这个模式可以自定义，玩家对所有的音符做出正确反应后游戏结束。

一、选题及创意介绍

我组作品的创意源自时下一种热门的游戏类型——音游（音乐游戏）。我们注意到身边有很多同学热衷于通过游玩音游来释放压力、放松心情，在讨论之初，我们就把制作音游作为备选创意之一。一般来说，音乐游戏以一首节奏明快的歌曲为核心，伴随歌曲的播放，屏幕上不断出现各种按键（note），玩家根据歌曲的节奏和屏幕显示的内容通过键盘、手指点按或身体动作的方式进行响应，游玩音游的过程类似于奏乐。音游考验的是玩家对节奏和时机的把握，需要玩家眼疾手快同时熟悉游戏内容。可以看出，音游的核心玩法在于“响应”，多样化的响应方式是音游的挑战和乐趣之所在。得益于 micro:bit 内置的多个按键（两个实体的按键以及一个电容触摸键）、不同类型的传感器（加速度计和声音传感器），我们可以在 micro:bit 上构造出多种响应方式，例如：点按单个按键，同时点按，摇晃等等。我们小组在讨论过程中注意到了 micro:bit 具有上述优势，进而坚定了制作音游的决心。

二、设计方案和硬件连接

方案概述：

我们的作品主要需要实现“显示”和“检测”两个方面，通过音符不同的生成方式进一步实现无尽模式和挑战模式。

设计方案：

对应显示，游戏的主 whlie 循环中，每隔一段时间清除显示内容并重新绘制新内容（用到 set_pixel 方法），这便是游戏的一帧，每帧绘制时改变音符的位置，从而实现了音符随时间流逝而移动的效果。

对应检测，每帧绘制的末尾，程序对底部一行 LED 灯珠进行检测，记录亮起灯珠的位置和数量，从而决定玩家应该做出的响应；若此帧要求玩家进行响应，程序进行一个新的 while 循环，从而检测玩家是否在一段时间内做出正确的回应。

硬件连接：

我们的硬件连接相对简单：只需使用一根课传输数据的 micro usb 线将 micro:bit 和电脑相连即可。

三、实现方案和代码分析

这里分析本作品最关键的三部分代码

1、绘制 note

```
#用来绘制一帧。
def draw_note(note):
    global x, y
    display.clear()
    r = len(note)
    c = len(note[0])
    for i in range(r):
        for j in range(c):
            if -1<x-c+1+j<5 and -1<y-r+1+i<5:
                display.set_pixel(x-c+1+j, y-r+1+i, note[i][j])
```

本作品中的 note 用一个 m*n 的方阵表示，x，y 是该方阵右下角对应的位置。draw_note 实现得是将给定 note 在给定位置实现。每次绘制前清屏，这通过调用 display.clear 方法实现。两个 for 循环旨在遍历 note 方阵，并将方阵中的各点与实际 micro:bit 上的 LED 灯珠对应起来。相应的灯珠亮度即设定为方阵对应位置的数据。这样，方阵所代表的图案就被绘制到了 micro:bit 的显示屏上。

2、检测底部

```
def check_bottom():
    bottom = [1 if display.get_pixel(i, 4) != 0 else 0 for i in range(5)]
    print(bottom)
    if bottom.count(1) == 0:
        return 0
    elif bottom.count(1) == 1:
        if bottom.index(1) < 2:
            return button_a.is_pressed
        elif bottom.index(1) == 2:
            return pin_logo.is_touched
        else:
            return button_b.is_pressed
    elif bottom.count(1) == 2:
        return both
    else:
        return shake
```

每帧绘制的末尾，这段代码将被调用，本质上，bottom 列表通过遍历最下面一行的五个灯珠得到，由于列表的有序性，底部亮度及亮起位置的信息就全部被收集到了 bottom 列表中。随后代码只需考察 bottom 列表，进而决定是否需要玩家操作以及需要操作时玩家的正确响应方式。其中 both 和 shake 的含义分别是同时按下 a 键和 b 键以及晃动 micro:bit（micro:bit 没有内置这两种方法）。

3、游戏的主循环

```
while gameOn:
    if zenMode:#无尽模式，随机生成，玩家若不失误则游戏一直进行
        sleep(25)#调节sleep的时间来控制移动的快慢
        t += 1
        if t == 40:
            t = 0
            if y < 5 + len(note):
                y += 1
                draw_note(note)
                check_button_and_show()

            else:
                print(str(zenScore))
                y = 0
                note = choice(note_lst)
                random_spawn(note)
                draw_note(note)

            if check_bottom() != 0:
                tp = 0
                while tp < 15:
                    sleep(25)
                    tp += 1
                    if check_bottom()():
                        zenScore += 1
                        break
                else:
                    gameOn = False
                    display.clear()
                    display.scroll("Game over! Your score is: ")
                    for i in range(5):
                        display.clear()
                        sleep(500)
                        display.show(zenScore)
                    display.clear()
```

这一部分代码就是游戏的主循环（这里以无尽模式为例），每次循环开始前的 sleep(25)旨调节循环的速度。变量 t 每次循环都加上 1，调节了两帧之间的时间间隔。每帧 y 都加上一，含义即是 note 向下方移动了一格。每一帧的绘制结束后，调用 check_bottom 方法来进行检测。一旦玩家美正确响应，变量 gameOn 成为 False，游戏循环结束。

4.串口通信部分

主要通过 pyserial 进行实现。

```
import serial
import serial.tools.list_ports as list_ports

def find_microbit_comport():
    ports = list(list_ports.comports())
    for p in ports:
        if (p.pid == 516) and (p.vid == 3368):
            return str(p.device)
def check_button_to_move_cb(s):
    if "a" in s:
        return "A"
    else:
        return "B"
def printtest():
    print("a")
if __name__ == '__main__':
    ser = serial.Serial()
    ser.baudrate = 115200
    ser.timeout = 1
    ser.port = find_microbit_comport()
    ser.open()
    #ser.write(b'testing')
```

四、后续工作展望

目前我们已经实现了编写 note 的接口，这使得我们的作品有丰富的可拓展性，目前可以做到定制 note 的形状，移动速度以及移动方向。我们后续可以根据合适的歌曲编写一系列 note，玩家们也可以根据自己喜爱的歌曲来编写一系列的 note，为了做到这一点，只需要将设计好的所有 note 形状、移动速度、生成位置分别装入代码对应位置的三个列表中，目前的工作就已经可以保证自定义的 note 正常运行。此外，micro:bit 按键和各种传感器的组合多种多样，我们目前只用到了其中的一部分，后续我们可以加入更多的相应方式，从而让游戏体验更加的丰富。

五、小组分工合作

本次创意活动分工如下：五一期间，张开完成了代码雏形的编写，初步实现了无尽模式（zen mode）以及挑战模式（challenge mode），显示与操作均在 micro:bit 上完成，受限于 micro:bit 糟糕的显示性能（只有 5*5LED 灯），游戏效果不尽如人意。随后程歆乐检查并测试了代码，发现了几处 bug，两人共同完成了 debug。为了获得更好的可视化效果，程歆乐同学导入了 pgzero 模块，并且为游戏增添了一些精致的图形和隐藏的彩蛋，实现了 micro:bit 和笔记本电脑的联动。收尾阶段，技术报告主题由张开写成，程歆乐做了补充和修改，作品的宣传 poster 由程歆乐制作完成。最后程歆乐同学完成了小组作品的运行和介绍视频。