作品实习报告

Microbit作品：钢琴块儿

作者：徐正贤 袁帅

摘要

我们小组的作品是通过Microbit自带的5\*5 LED/光线传感器、加速度传感器与实体按键，将前几年大火的手机游戏《钢琴块儿》的玩法在轻巧便携的python开源硬件上实现。玩家可以在软件中选择不同模式与乐曲，进入游戏后通过屏幕的指示击打相应按键或进行其他操作，模拟演奏出音乐。

1. 选题及创意介绍

小组成员在对Microbit的功能与外型进行分析后，发现其适合手持的外型与早期的游戏掌机十分相像，因此小组萌生了开发移植Microbit游戏的想法。在经过挑选后，我们选择了前段时间大火的手机游戏《钢琴块儿》，并结合Microbit的特性进行改动，加入了新的功能玩法与创意。

1. 设计方案和硬件连接

当玩家按下logo键时，主题音乐响起，并进入游戏的UI。游戏系统的操作由logo键、A键、B键进行，分别代表确定，向左，向右。

首先，我们为游戏设计了模式选择的功能。模式选择有两种：休闲娱乐模式、计时挑战模式。由AB键进行模式切换，logo键进行模式选择。

1.休闲娱乐模式（classic）

顾名思义，即没有任何游玩压力，让玩家能够练习娱乐的模式，玩家在此模式可以不受时间限制地游玩曲目，如果曲目全程无失误，则显示通关。

2.计时挑战模式（speed）

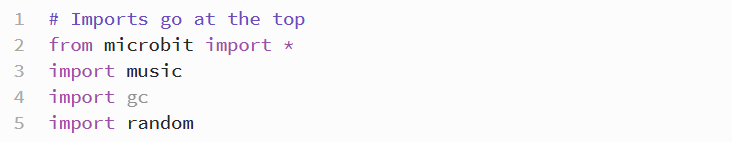
在规定时间内让玩家尽可能打击更多次数，并记录分数的模式，期间需要玩家没有失误，否则游戏也会立即结束。在游戏结束后会显示玩家的分数，因此有一定的竞技性。

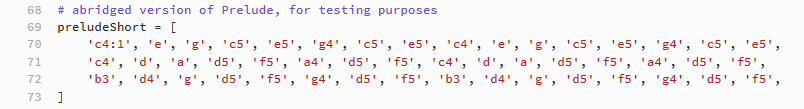
其次，我们为游戏设计了歌曲选择的功能。玩家在选择模式之后，就可以选择相应的歌曲，并进入游戏。同样的，AB键进行乐曲的切换，logo键进行乐曲的选择。玩家在游戏结束后，显示屏对结果进行显示，此时按下logo键，可以重新开始游戏。

进入游戏，画面上会显示出两个轨道，分别对应microbit的A键与B键，上面有需要击打的音符，玩家要按顺序正确的击打音符，每击打成功一个音符，便会有对应的音乐声响起（声音调用提前录入进去的乐谱）。同时游戏中也有模拟钢琴踏板的音符（一行全亮），玩家需要摇晃microbit，通过加速传感器获取摇晃的信号，实现音符的打击。

因为游戏是在Microbit上独立运行、操作、游玩的，因此没有设计额外的硬件连接方案。但我们也大致构想了游戏未来的一些外接设计，详情请看下文【四、后续工作展望】部分。

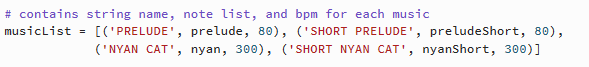
1. 实现方案及代码分析

此处为所需要使用的库。

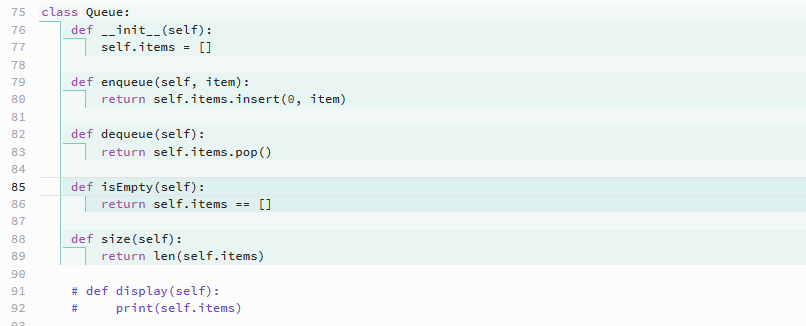


此处为所使用歌曲的乐谱，格式为microbit中music.play所能识别的列表形式。

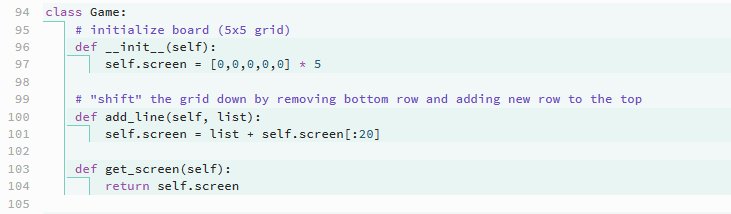
 Music.play的规则如上。



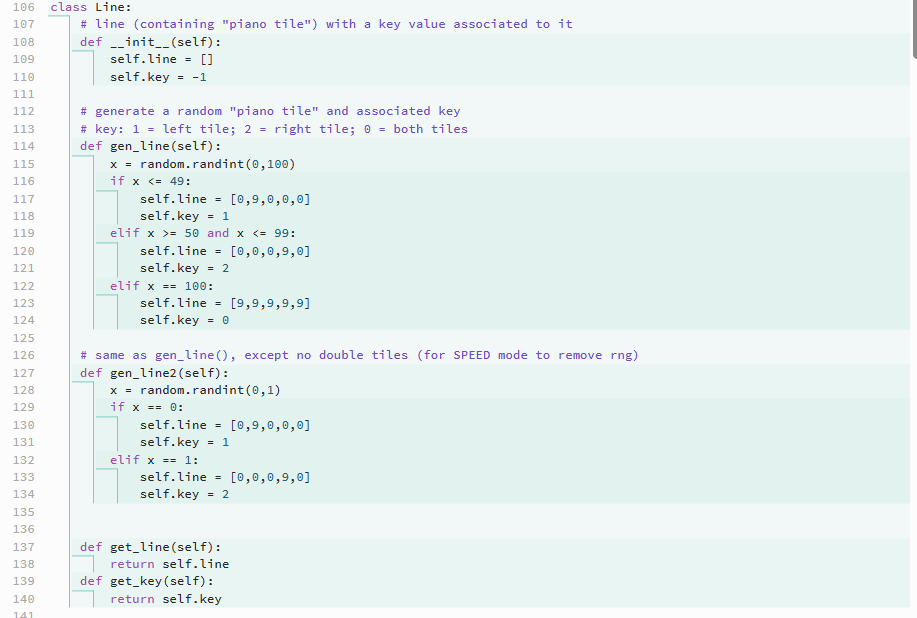
一个存放音乐相关信息的列表。方便后面选择音乐时进行调用。



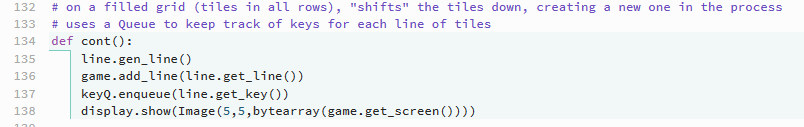
需要在后续使用的Queue类



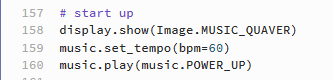
Class Game实现了游戏画面的基本运行逻辑。初始的游戏画面为5x5的空白画面。当执行add\_line的时候，则是将最下一行的五个元素删除（通过切片的方式），并添加新的一行在最上端，以此来实现‘音符’向下掉落的效果。Get\_screen返回当前的游戏画面。

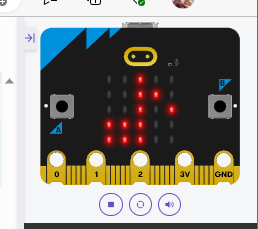


Class line 实现了以行为单位的音符的生成。一个初始的line为空列表，其对应的key为异常值-1。gen\_line生成每一行的音符，首先通过random方法随机生成0-100的自然数，并设定范围分配相应的值（1为A键，2为B键，0为加速传感器中的shake）。Getline、getkey分别返回该行音符与对应需要按的key值。分别定义了两个gen\_line，对应两个不同的模式。因为考虑到shake相对于按键更耗费时间，且具有随机性，会影响speed模式分数的公平性。因此shake按键只在classic模式生成。



Cont函数进行了完整的一次音符下落的操作，即让所有音符平移下来，创建一行新音符放在最上面，并显示在屏幕上。Cont2与Cont原理相同，仅是为了区分不同的mode，在此不多做赘述。





Start up开机界面生成如上的一个音符图案，并播放音乐。

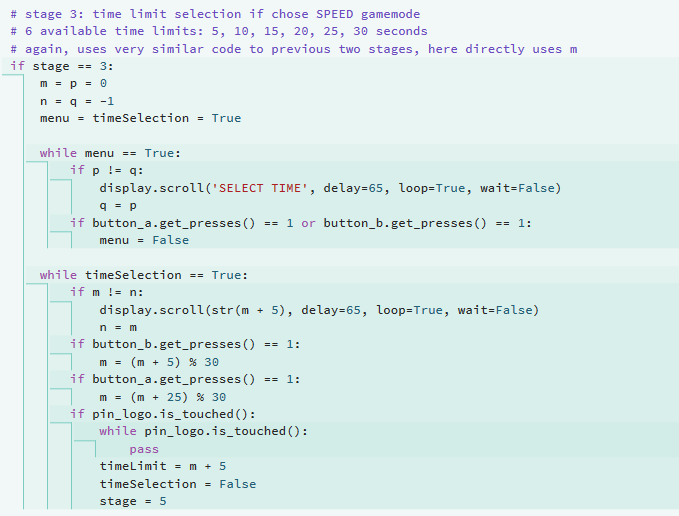
我们将这个游戏的结构分成几个stage：stage0为开机的界面，stage1为选择歌曲界面，stage2为选择模式界面。Stage3为选择speed模式后的限制时间选择界面，stage4为classic模式的游玩界面，stage5为speed模式的游玩界面。接下来我将分stage对代码进行讲解。



这个大while语句是为了进入游戏，并选择音乐。（即stage0与stage1）我们使用有关logo的按键（pin\_logo.is\_touched()）的while语句来判断是否触发，具体实现方法详见注释。当玩家触发按键，即进入stage1。

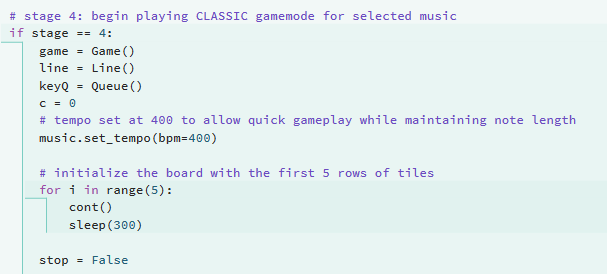
m = p = 0 n = q = -1两句是为了后续语句判断的布尔值而设置。并无实际作用（没有想到更好的方案）button get\_presses()是为了在后续重新游玩时，清除之前按键的记录。进入stage1，首先进入menu界面，滚动'SELECT MUSIC'，直到按键触发使menu的布尔值为false，进入实际歌曲的选择。实际歌曲的选择则是通过对m，n的加法，整除（循环歌曲列表）操作进行的，当玩家选定歌曲，点击logo（if pin\_logo.is\_touched()），则进入stage2。

stage2（游戏模式选择）的运作机理与stage1相同，在此不做赘述。



Stage3选择speed模式的游戏限制时间。运作机理依然很接近前两个stage。

需要注意时间选择时timelimit与m的不同：m是为了方便计算而设置的变量，并非实际的时间。最后要在m上+5，才是最终游玩时的时间（也是display上显示的时间）



Stage4为classic的游玩界面。因代码较长，分为几个部分讲述。

首先进入stage4，建立几个需要使用的类，并先生成五行音符（玩家需要先击打音符才能生成下一个音符，因此最开始五个音符的生成是必要的）

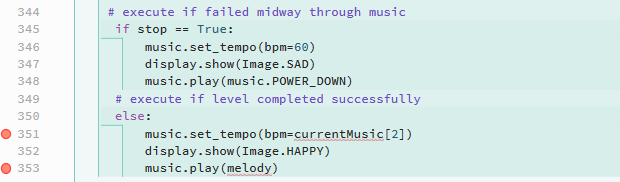


接下来迭代乐谱list中的每一个元素，生成音符。

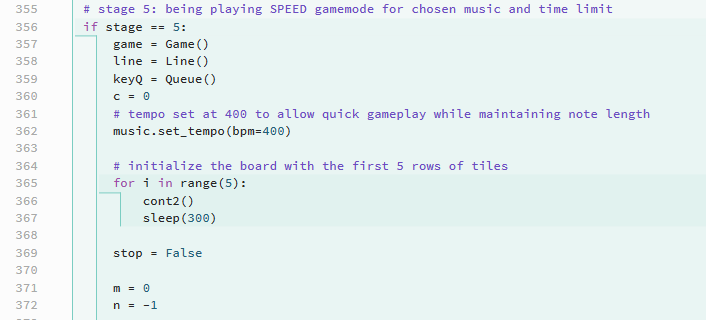
当stop布尔值为true时，通过key值获取每一个音符对应的key值，并判断按键（或shake）是否与key值相吻合，以此来达到判断玩家是否按到了正确按键的目的。当玩家按到正确按键，则消除旧音符，更新一行新的音符。

当玩家按到的按键与key值不符合，则使stop布尔值变为true，并中断接下来的音符生成。

需要注意的是因为classic模式音乐不循环，因此设置了一个判断音乐结束前音符不够生成五行音符时的特殊情况，详见代码299-304。



当游戏结束后，通过stop布尔值来判断游戏进行的情况，输出不同的结果。



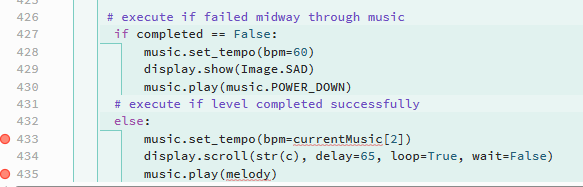
Stage5为speed模式下的游戏界面。同样，因代码较长的缘故，分段进行说明。

前面的部分与stage4相同，仅有新变量mn是为了进行时间的计算。



此处编写了speed的运作逻辑。timeElapsed表示启动定时器后经过的时间，timer表示定时器是否在运行，通过timeElapsed和start的时间记录相减（通过time.ticks\_ms()来实现计时）得到当前的游戏时间，并后续与timelimit进行比较。completed表示玩家是否在没有失败的情况下完成了游戏。

需要注意的是，因为speed模式下歌曲是无限循环的，因此不像stage4一样，需要对music len少于5的情况进行特殊的安排。



最后根据游玩的情况进行结果的输出。因为游戏结构整体采用stage的分割形式，因此玩家在游玩结束后，可以直接返回到先前的循环，再次开始新的游戏。

1. 后续工作展望

我们的作品仍然有很多值得完善的地方，比起一个完整的项目，倒不如说更像是一个demo，有大量的新功能可以添加与挖掘。小组成员后续将对项目进行以下几个方向的开发：

1. 玩法的增加

目前游戏只有两个键与一个摇晃的操作，游戏画面也基本只用到了2、4轨道，玩法相对单薄。后续可能会结合Microbit的其他功能与传感器，进行更多玩法的创新。比如同时按下两个键演奏乐曲中的chord，并结合加速度传感器的其他功能增加shake以外的音符。比如将5x5的屏幕利用起来，最左、右面的音符需要玩家手持Microbit向左、右倾斜再击打音符，甚至是通过v2新增的麦克风来实现唱歌击打音符……

1. 外部硬件连接

Microbit 支持使用Chrome 或 Edge 网络浏览器在编辑器中访问串行控制台，使用计算机的键盘将信息输入到 Python 程序中。因此后续可以添加端口，使玩家只要将microbit连接到电脑，就可以通过串行控制台将想要演奏的歌曲便捷地录入到机器中（只要遵守music.play的格式）

此外，microbit 的底部边缘有金色引脚，可用于连接触摸传感器。后续可以使用触摸传感器代替原本的按键，让玩家更有演奏乐器的代入感。

1. 联网排名

当Microbit连接电脑时，通过串行控制台输出机器内记录的最高分数，并上传到服务器。玩家可以通过服务器查看自己的排名，提高游戏的竞技性。

1. 小组分工合作

袁帅：软件编写、debug、视频拍摄、游戏设计、Poster设计

徐正贤：游戏设计、曲谱编写、软件编写、实习报告编写