璃奈板，但是俄罗斯方块

作者：王翊霏，李欣宸

摘要：本作品是两个反差很大（~~毫不相关~~）的创意的结合。第一部分是运用microbit的5\*5点阵进行抽象的表情随机显示及音乐播放（毕竟只有25个格点）。第二部分是一个经典小游戏俄罗斯方块的简化+改编版（毕竟只有25个格点），游戏结束后配有创意彩蛋。当玩家成功填满一定数量的格子时，会触发动画和语音功能。当然有某个数量的格子和其他数量的格子不太一样，它与pi相乘并取整后恰好是50！然后会播放大家熟悉的音乐和话语！（猜一猜）下面会详细介绍创意、实现方案等。

1. 选题及创意介绍

由于我们小组选择无实物创作，因此想把单片机的功能运用的全面一些。第一位同学主要运用图像和音乐，做了一个简单版“璃奈板”用于随机显示及切换不同表情（用点来表示）。

第二位同学主要运用按钮，并做了她最爱玩的俄罗斯方块的改编版。最初的想法是复现俄罗斯方块，但由于板的空间比较小，消除与旋转实现起来较为复杂，她最终选择了减少每个俄罗斯方块的组成块的个数，并不允许旋转与消除，游戏结束条件即为不能再放置俄罗斯方块。

她和第一位同学在游戏结束后进行了再创意。通过结算已经填充的方格的个数count结算money，money=[count\*pi]，然后显示图案，播放语音“crazy Thursday V me {money}”。细心的同学可能发现，当count==16时，money=50，此情形当然是不平凡的，因此我们要播放98k音乐使你如同置身KFC，并自信的说出“crazy Thursday V me fifty”。

1. 设计方案和硬件连接

第一个创意的设计方案：预先写入不同点阵表示的表情，运用random进行随机抽取表情播放，同样运用random确定每个表情显示的时长。关于音乐播放部分，用列表写入一定格式表示的音符即可。

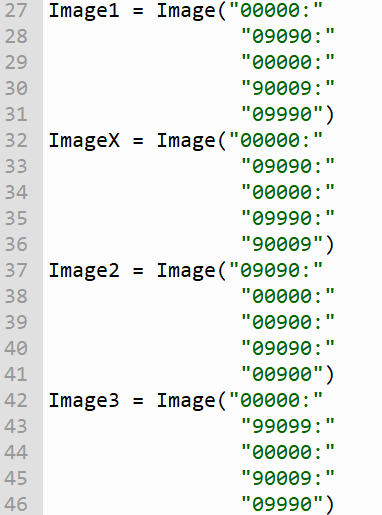
第二个创意的设计方案：通过改变坐标来控制俄罗斯方块的左移、右移、下移，用嵌套列表记录每个点是否被点亮，在进行移动操作时先检验该移动是否合法。初始时每个方块从顶部落下，当方块不能再下落时，更改记录每个点是否被点亮的嵌套列表，并点亮该方块的最终位置。在下落的过程中，监测A、B两个按钮是否被按下，如果A被按下且不发生冲突，方块向左移动，否责方块不移动。B按钮同理可知。如果方块在初始位置刚一出现就发生冲突，则游戏结束，通过记录每个点是否被点亮的嵌套列表结算被点亮的点的总个数，并计算出总钱数，如果总钱数恰好等于50，滚动“KFC!orz”并播放音乐和语音。否则仅播放语音。

一些设计上的细节：考虑到仅有25个方格，俄罗斯方块的组成块数变为1/2/3，且1/2块的俄罗斯方块生成概率较大。为使得不同块在下落摆放好后能被区分开来，在初始生成该块时，就随机选择一个亮度。在结算money的时候，由于money是一个数字，直接读会变成读电话号码，因此维护一个字典，里面是所有可能出现的数字和英文的对应，例如50：fifty。

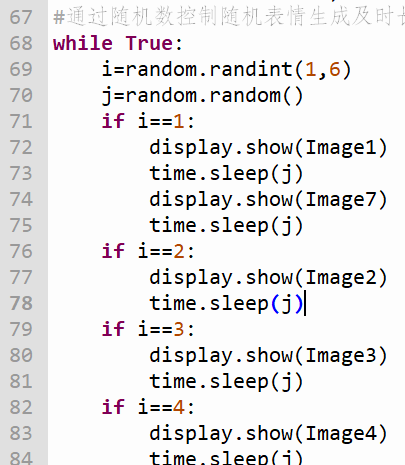
1. 实现方案和代码分析

璃奈板部分：

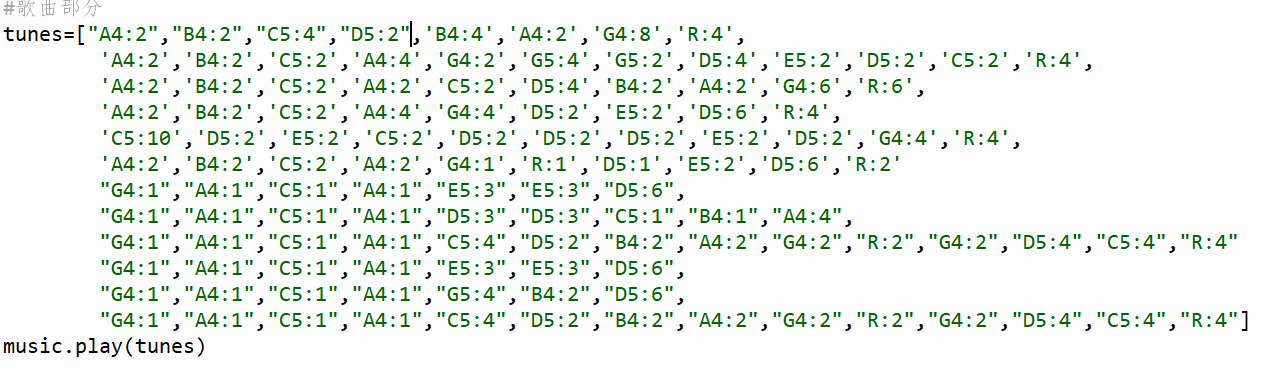
首先将表情表示出来：



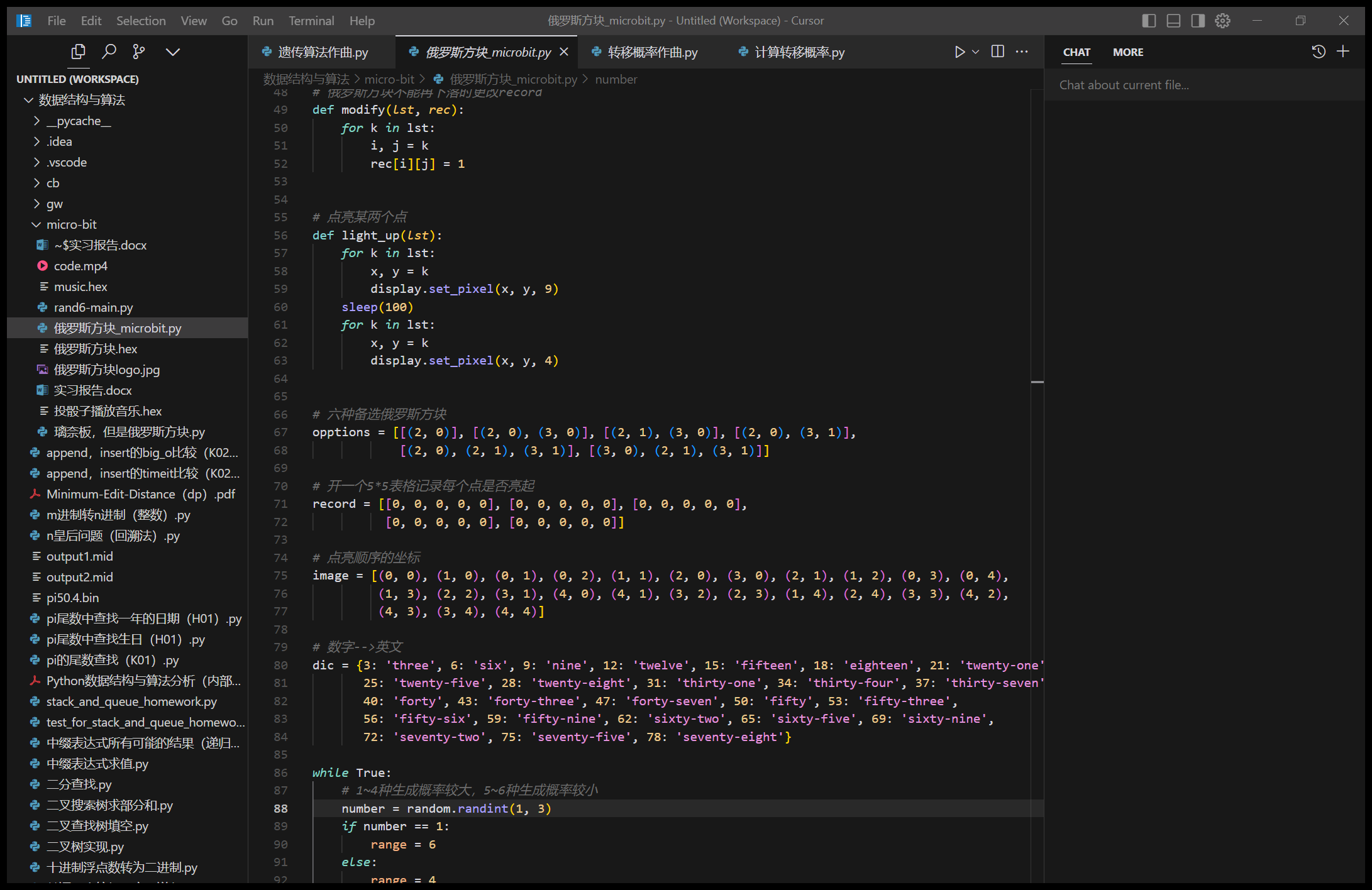
随机播放



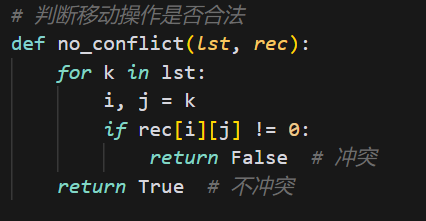
写入并播放歌曲：



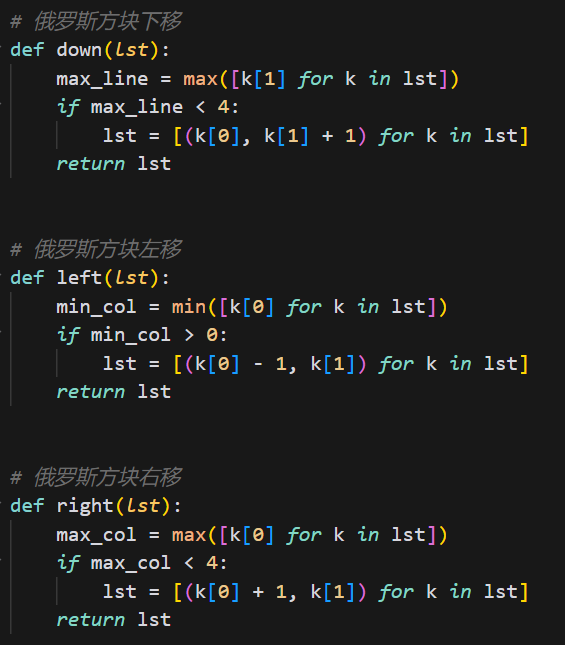
俄罗斯方块部分：

首先初始化一个二维列表options，每个元素是一个列表，存放俄罗斯方块每个小方块初始位置的坐标。再初始化一个二维列表record，记录每个点是否亮起。

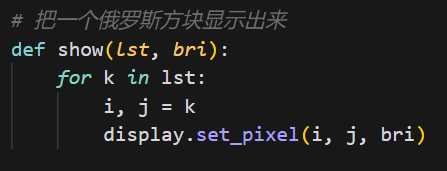
下面定义判断是否冲突的函数，参数为两个列表，一个为记录当前俄罗斯方块坐标的列表lst，另一个为记录每个点是否亮起的列表rec（即为record），对于第一个列表中的每个元素，即每个点的坐标，在rec中检索其值，判断是否已经被点亮。



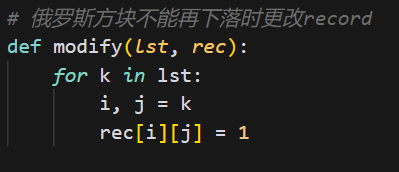
下面定义俄罗斯方块移动的函数，以下移为例。参数为记录当前俄罗斯方块坐标的列表lst，首先计算出所有坐标中的最大行，如果最大行不是最后一行，则将所有点纵坐标+1，否则不操作。



下面是把俄罗斯方块显示出来的函数，参数为记录当前俄罗斯方块坐标的列表lst和亮度bri。



下面为更改record的函数，一个参数是记录当前俄罗斯方块坐标的列表lst，另一个为记录每个点是否亮起的列表rec。

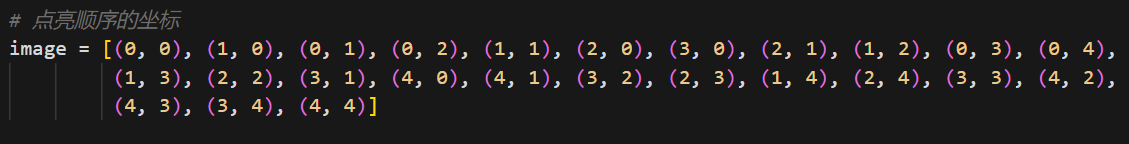


下面是实现的主体部分。随机生成一个俄罗斯方块和其颜色之后，先把它显示出来，如果不发生冲突，在俄罗斯方块每次下移操作之前，对A/B是否被按下进行五次检测，如果A被按下则执行left函数，如果B被按下则执行right函数，然后进行下移操作。如果发生冲突，则更改record，退出下移的循环。此时外层循环会产生新的俄罗斯方块，继续进行上述循环。若新的俄罗斯方块一经产生就发生了冲突，则游戏结束。将已经被点亮的点全部变成最亮（通过遍历record）实现。

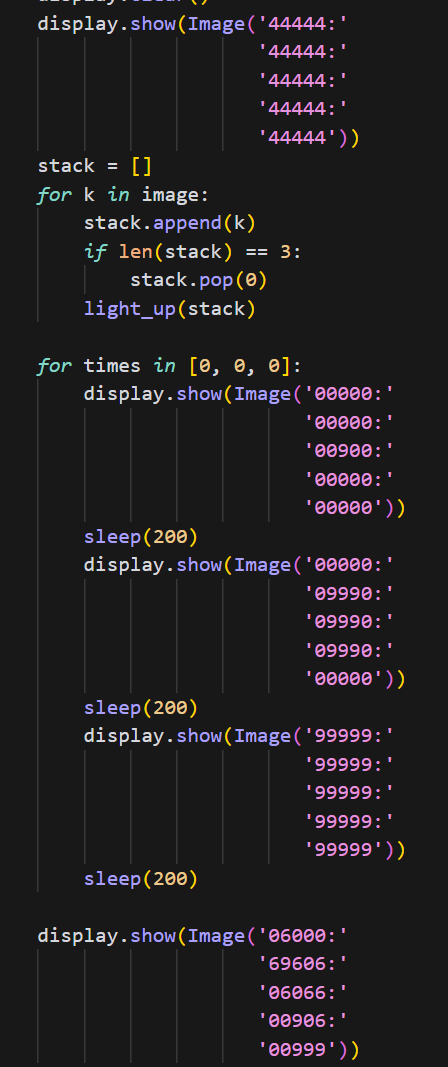




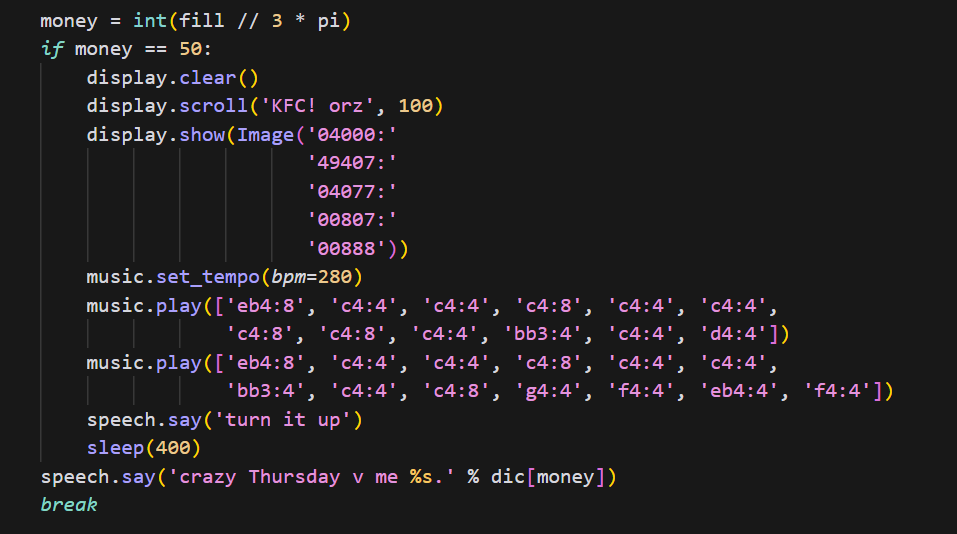
此时代码的主体部分就完成了。下面介绍一些创意图案的设计。下面是5\*5方块蛇形遍历的坐标构成的列表。首先将所有点设为较低亮度。然后遍历image，维护一个栈记录要点亮的几个点的坐标（两个，边界情况只有一个）。栈作为light\_up函数的参数，可以实现点亮0.1s在恢复原来亮度的功能。最终的视觉效果就是两个亮点在沿蛇形跑动。然后展示一些图案，和一个logo。Logo的设计理念是左上角一颗星星，右下角两个经典俄罗斯方块。



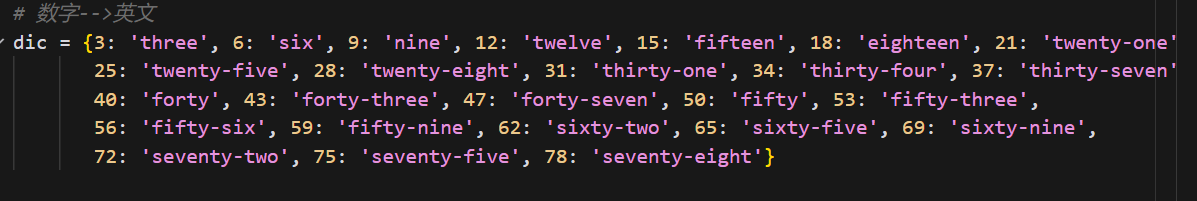




下面一部分是结算money的代码，并显示logo，播放音乐和语音。



最后的代码是之前提到的一个数字转化为英文的细节。



1. 后续工作展望

璃奈板部分：5\*5真的限制发挥（咆哮ing）。其他方面优化可以从表情切换更和音乐贴合入手。

俄罗斯方块部分：游戏还有一定的优化空间，比如俄罗斯方块的旋转功能，游戏分成几个不同难度模式，通过下落速度来区分，游戏进行过程中还可以逐渐加快下落速度，对于A/B建是否被按下的检测也可以进行优化，增强灵敏性，以及可以增加一个按钮，直接下落到底部。最后还可以尝试实现俄罗斯方块的消除功能。

五、小组分工合作

璃奈板部分：李欣宸

俄罗斯方块部分：王翊霏

视频剪辑、海报制作：李欣宸

实习报告：王翊霏、李欣宸